

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ
ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΡΓΑΣΤΗΡΙΟ
ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΜΕΓΑΛΗΣ ΚΛΙΜΑΚΑΣ
ΚΑΙ ΜΗΧΑΝΙΚΗΣ ΔΕΔΟΜΕΝΩΝ ΝΕΦΟΥΣ

Υπολογιστική νοημοσύνη και βαθιά μάθηση

Σ. ΛΥΚΟΘΑΝΑΣΗΣ
Δ. ΚΟΥΤΣΟΜΗΤΡΟΠΟΥΛΟΣ



ΣΠΥΡΙΔΩΝ ΛΥΚΟΘΑΝΑΣΗΣ
Καθηγητής Πανεπιστημίου Πατρών

ΔΗΜΗΤΡΙΟΣ ΚΟΥΤΣΟΜΗΤΡΟΠΟΥΛΟΣ
Δρ. Ε.ΔΙ.Π. Πανεπιστημίου Πατρών

*Υπολογιστική νοημοσύνη
και βαθιά μάθηση*



Υπολογιστική νοημοσύνη και βαθιά μάθηση

Συγγραφή

Σπυρίδων Λυκοθανάσης

Δημήτριος Κουτσομητρόπουλος

Συντελεστές έκδοσης

Γλωσσική Επιμέλεια: Βασιλική Κανάρη

Γραφιστική Επιμέλεια: Ευαγγελία Μουστάνη

Κεντρική Ομάδα Υποστήριξης

Γλωσσικός Έλεγχος: Γεωργία Τριανταφυλλίδου

Γραφιστικός Έλεγχος: Χρήστος Κεντρωτής

Βιβλιοθηκονομική Επεξεργασία: Ευδοξία Καρλή

Copyright © 2023, ΚΑΛΛΙΠΟΣ, ΑΝΟΙΚΤΕΣ ΑΚΑΔΗΜΑΪΚΕΣ ΕΚΔΟΣΕΙΣ
(ΣΕΑΒ + ΕΛΚΕ-ΕΜΠ)



Το παρόν έργο αδειοδοτείται υπό τους όρους της άδειας Creative Commons Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Παρόμοια Διανομή 4.0. Για να δείτε ένα αντίγραφο της άδειας αυτής επισκεφτείτε τον ιστότοπο <https://creativecommons.org/licenses/by-nc-sa/4.0/deed.el>

Αν τυχόν κάποιο τμήμα του έργου διατίθεται με διαφορετικό καθεστώς αδειοδότησης, αυτό αναφέρεται ρητά και ειδικώς στην οικεία θέση.

ΚΑΛΛΙΠΟΣ

Εθνικό Μετσόβιο Πολυτεχνείο

Ηρώων Πολυτεχνείου 9, 15780 Ζωγράφου

www.kallipos.gr

ISBN: 978-618-5726-47-8

Βιβλιογραφική Αναφορά: Λυκοθανάσης, Σ., & Κουτσομητρόπουλος, Δ. (2023). *Υπολογιστική νοημοσύνη και βαθιά μάθηση* [Προπτυχιακό εγχειρίδιο]. Κάλλιπος, Ανοικτές Ακαδημαϊκές Εκδόσεις. <http://dx.doi.org/10.57713/kallipos-168>

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΙΝΑΚΑΣ ΑΚΡΩΝΥΜΙΩΝ.....	11
ΠΡΟΛΟΓΟΣ.....	13
Κεφάλαιο 1 Δυνατότητες και Εφαρμογές Νευρωνικών Δικτύων και Γενετικών Αλγορίθμων	15
1.1 Εισαγωγικές Παρατηρήσεις	16
1.2 Εισαγωγή στην Τεχνητή Νοημοσύνη.....	16
1.2.1 Νευρωνικά Δίκτυα	17
1.2.2 Γενετικοί Αλγόριθμοι.....	17
1.2.3 Έμπειρα Συστήματα.....	18
1.2.4 Ασαφή Συστήματα.....	18
1.3 Στοιχεία της Θεωρίας Μάθησης.....	18
1.4 Μέθοδοι Αναζήτησης και Βελτιστοποίησης.....	20
1.5 Εφαρμογές των ΝΔ	21
1.6 Η Έμπνευση από τα Βιολογικά Συστήματα.....	22
1.6.1 Από τα Βιολογικά Νευρωνικά Δίκτυα στα Τεχνητά.....	22
1.6.2 Πλεονεκτήματα των Νευρωνικών Δικτύων	23
1.6.3 Η θεωρία της Εξέλιξης των Ειδών.....	24
1.6.4 Πλεονεκτήματα ΓΑ.....	25
1.7 Σύνοψη κεφαλαίου	27
Βιβλιογραφία.....	28
Απαντήσεις ασκήσεων	29
Κεφάλαιο 2 Εισαγωγή στα Τεχνητά Νευρωνικά Δίκτυα	31
2.1 Εισαγωγικές Παρατηρήσεις	31
2.2 Ιστορική αναδρομή.....	32
2.3 Το μοντέλο του τεχνητού νευρώνα	34
2.3.1 Τύποι συναρτήσεων ενεργοποίησης	37
2.3.2 Θεωρώντας τα ΝΔ ως κατευθυνόμενους γράφους	39
2.4 Αρχιτεκτονικές των Νευρωνικών Δικτύων.....	41
2.5 Σύνοψη κεφαλαίου	44
Βιβλιογραφία.....	45
Απαντήσεις ασκήσεων	46
Κεφάλαιο 3 Αλγόριθμοι Μάθησης.....	49
3.1 Εισαγωγικές Παρατηρήσεις	50
3.2 Ο αλγόριθμος μάθησης του Perceptron (Αισθητήρα)	50
3.2.1 Το θεώρημα σύγκλισης του Perceptron	52
3.2.2 Ανακεφαλαίωση.....	53
3.3 Ο αλγόριθμος Πίσω Διάδοσης (ΠΔ) του λάθους	54
3.3.1 Παραγωγή του αλγορίθμου Πίσω Διάδοσης.....	58
3.3.2 Διαδικασία Μάθησης.....	59
3.3.3 Τα δύο περάσματα του υπολογισμού.....	64
3.3.4 Σιγμοειδής μη-γραμμικότητα.....	65
3.3.5 Ρυθμός μάθησης.....	66
3.3.6 Τρόποι εκπαίδευσης του δικτύου.....	67

3.3.7 Κριτήρια τερματισμού	69
3.3.8 Σύνοψη του αλγορίθμου Πίσω Διάδοσης	69
3.4 Σύνοψη κεφαλαίου	72
Βιβλιογραφία	73
Απαντήσεις ασκήσεων	74
Κεφάλαιο 4 Αρχές και Περιορισμοί Σχεδιασμού Τεχνητών Νευρωνικών Δικτύων	79
4.1 Εισαγωγικές Παρατηρήσεις	79
4.2 Αρχικοποίηση	80
4.3 Το πρόβλημα της XOR	81
4.4 Μερικοί τρόποι αποτελεσματικότερης εκτέλεσης του αλγορίθμου	84
4.5 Γενίκευση	86
4.6 Επικύρωση εκπαιδευμένου νευρωνικού δικτύου	88
4.6.1 Επιλογή καλύτερου δικτύου με διασταυρωμένη επικύρωση	89
4.6.2 Βέλτιστες σταθερές μάθησης και ορμής	91
4.7 Κατάλληλο μέγεθος εκπαιδευτικού συνόλου για έγκυρη γενίκευση	92
4.8 Σύνοψη κεφαλαίου	93
Βιβλιογραφία	95
Απαντήσεις ασκήσεων	96
Κεφάλαιο 5 Δίκτυα Hopfield και Δίκτυα Kohonen	101
5.1 Εισαγωγικές Παρατηρήσεις	102
5.2 Εκπαίδευση Δικτύου Hopfield	103
5.3 Δίκτυα Kohonen	109
5.3.1 Εκθετική μείωση του εύρους της γειτονιάς	114
Βιβλιογραφία	117
Απαντήσεις ασκήσεων	118
Κεφάλαιο 6 Βελτιστοποιήσεις για την Εκπαίδευση Βαθέων Μοντέλων	129
6.1 Η συνάρτηση κόστους διεντροπίας	129
6.1.1 Εισαγωγή της συνάρτησης κόστους διεντροπίας	132
6.1.2 Τι σημαίνει διεντροπία; Από πού προέρχεται;	135
6.1.3 Softmax	136
6.2 Υπερεκπαίδευση και ομαλοποίηση	138
6.2.1 Ομαλοποίηση	139
6.2.2 Άλλες τεχνικές ομαλοποίησης	141
6.3 Αρχικοποίηση βαρών	146
6.4 Επιλογή υπερπαραμέτρων νευρωνικών δικτύων	148
6.4.1 Ρυθμός μάθησης	148
6.4.2 Χρήση πρόωρου σταματήματος για προσδιορισμό του αριθμού των εποχών	150
6.4.3 Πρόγραμμα ρυθμού μάθησης	151
6.4.4 Η παράμετρος ομαλοποίησης, λ	151
6.4.5 Μέγεθος Mini-batch	151
Βιβλιογραφία	154
Απαντήσεις ασκήσεων	155
Κεφάλαιο 7 Μοντέλα Νευρωνικών Δικτύων Βαθιάς Μάθησης	159
7.1 Εισαγωγή στα συνελκτικά δίκτυα	160

7.1.1 Τοπικά πεδία πρόσληψης (local receptive fields)	161
7.1.2 Κοινόχρηστα βάρη και πολώσεις.....	163
7.1.3 Επίπεδα ομαδοποίησης (pooling layers).....	165
7.1.4 Πλήρες συνελκτικό δίκτυο.....	166
7.2 Συνελκτικά νευρωνικά δίκτυα στην πράξη.....	167
7.2.1 Χρήση ανορθωμένων γραμμικών μονάδων (ReLUs).....	168
7.2.2 Επέκταση του συνόλου εκπαίδευσης.....	169
7.2.3 Εισαγωγή επιπλέον πλήρως συνδεδεμένου επιπέδου	169
7.2.4 Χρησιμοποιώντας ένα σύνολο δικτύων (ensemble)	170
7.3 Άλλες προσεγγίσεις στα βαθιά νευρωνικά δίκτυα	171
7.3.1 Επαναλαμβανόμενα νευρωνικά δίκτυα (RNNs).....	171
7.3.2 Μονάδες μακράς βραχυπρόθεσμης μνήμης (LSTMs).....	172
7.3.3 Βαθιά δίκτυα πεποιθήσεων, γενετικά μοντέλα και μηχανές Boltzmann	173
7.3.4 Άλλες ιδέες.....	173
Απαντήσεις ασκήσεων	175
Βιβλιογραφία.....	177
Κεφάλαιο 8 Εισαγωγή στους Γενετικούς Αλγορίθμους	179
8.1 Εισαγωγή.....	180
8.1.1 Πλεονεκτήματα των Γενετικών Αλγορίθμων	181
8.1.2 Τι Προκαλεί Δυσπιστία.....	183
8.1.3 Βασικά Χαρακτηριστικά Γενετικών Αλγορίθμων	184
8.2 Βασικά Στοιχεία Γενετικού Αλγορίθμου	186
8.2.1 Εφαρμογή – μεγιστοποίηση συνάρτησης	188
8.3 Θεωρητική Ανάλυση των Γενετικών Αλγορίθμων	196
8.4 Συμπεράσματα.....	204
Βιβλιογραφία.....	205
Απαντήσεις ασκήσεων	206
Κεφάλαιο 9 Μελέτη Περίπτωσης – Εξελκτικά Νευρωνικά Δίκτυα	213
9.1 Εισαγωγή.....	213
9.2 Το μοντέλο του νευρώνα.....	215
9.3 Ο Αλγόριθμος.....	216
9.4 Εξομοιώσεις	218
9.5 Ανασκόπηση – Συμπεράσματα	221
Βιβλιογραφία.....	222
Απαντήσεις ασκήσεων	223
Κεφάλαιο 10 Γενετικός Προγραμματισμός (ΓΠ).....	227
10.1 Εισαγωγικές Παρατηρήσεις	228
10.2 Εισαγωγή στον Γενετικό Προγραμματισμό	229
10.2.1 Το σύνολο των τερματικών.....	233
10.2.2 Το σύνολο των συναρτήσεων	234
10.2.3 Δομές εκτελέσιμων προγραμμάτων	234
10.2.4 Αρχικοποίηση του πληθυσμού.....	235
10.2.5 Διασταύρωση	236
10.2.6 Μετάλλαξη.....	237
10.2.7 Αναπαραγωγή	238

10.2.8 Αντικειμενική συνάρτηση.....	238
10.2.9 Επιλογή	239
10.2.10 Ο βασικός αλγόριθμος.....	239
10.3 Λειτουργίες Μεταβολής της Αρχιτεκτονικής του Προγράμματος.....	241
10.4 Βασικά ζητήματα της υλοποίησης του ΓΠ.....	243
10.4.1 Ανασκόπηση του διαγράμματος ροής του ΓΠ.....	243
10.4.2 Παραγωγή δένδρων συντακτικής ανάλυσης προγράμματος.....	245
10.5 Εφαρμογές του ΓΠ	248
10.5.1 Μελέτη περίπτωσης: Ευφυής στρατηγική για ένα τεχνητό μυρμήγκι (artificial ant)	248
Βιβλιογραφία	251
Απαντήσεις ασκήσεων	252

ΠΙΝΑΚΑΣ ΑΚΡΩΝΥΜΙΩΝ

Back Propagation	(BP)
Convolutional Neural Networks	(CNNs)
Cross Validation	(CV)
Deep Belief Networks	(DBNs)
Error Back Propagation	(EBP)
Exclusive OR	(XOR)
k-fold Cross Validation	(k-CV)
Least Mean Squares	(LMS)
Long-Short Term Memory	(LSTMs)
Multi Layer Perceptrons	(MLPs)
Neural Turing Machine	(NTM)
Processing Units	(PUs)
Rectified Linear Units	(ReLU)
Recurrent Neural Networks	(RNNs)
Self-Organizing Maps	(SOMs)
Stochastic Gradient Descent	(SGD)
Ασαφή Συστήματα	(ΑΣ)
Γενετικοί/Εξελικτικοί Αλγόριθμοι	(Γ/ΕΑ)
Γενετικοί Αλγόριθμοι	(ΓΑ)
Γενετικός Προγραμματισμός	(ΓΠ)
Ελάχιστου Μέσου Τετραγωνικού	(EMT)
Έμπειρα Συστήματα	(ΕΣ)
Εξελικτικοί Αλγόριθμοι	(ΕΑ)
Ευφυΐα Σμήνους	(ΕΣ)
Μέσου Τετραγωνικού Λάθους	(ΕΜΛ)
Νευρωνικά Δίκτυα	(ΝΔ)
Πίσω Διάδοσης	(ΠΔ)
Τεχνητά Νευρωνικά Δίκτυα	(ΤΝΔ)
Τεχνητή Νοημοσύνη	(ΤΝ)

ΠΡΟΛΟΓΟΣ

Σκοπός αυτού του συγγράμματος είναι να αποτελέσει βασικό εκπαιδευτικό υλικό στην «Υπολογιστική Νοημοσύνη και Βαθιά Μάθηση». Απευθύνεται σε αποφοίτους των Τμημάτων Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής, Ηλεκτρολόγων Μηχανικών, Επιστήμης της Πληροφορικής και συναφών Τμημάτων. Η Υπολογιστική Νοημοσύνη αποτελεί έναν υποκλάδο της Τεχνητής Νοημοσύνης (Engelbrecht, 2007), η οποία περιλαμβάνει τον σχεδιασμό και την ανάπτυξη θεωριών και μεθόδων που έχουν εμπνευστεί από τη φύση, καθώς και την εφαρμογή τους στην επίλυση προβλημάτων του πραγματικού κόσμου. Σύμφωνα με έναν άλλον ορισμό (Bezdek, 1994), ένα σύστημα είναι υπολογιστικά ευφυές όταν αυτό ασχολείται μόνο με αριθμητικά (χαμηλού επιπέδου) δεδομένα, έχει στοιχεία αναγνώρισης προτύπων, δεν χρησιμοποιεί γνώση με την έννοια της Τεχνητής Νοημοσύνης και, επιπλέον, όταν αυτό εμφανίζει: 1) υπολογιστική προσαρμοσιμότητα, 2) υπολογιστική ανοχή λαθών, 3) ταχύτητα που πλησιάζει την ανθρώπινη ευστροφία και 4) αναλογία λαθών τα οποία προσεγγίζουν την ανθρώπινη επίδοση.

Οι μέθοδοι της Υπολογιστικής Νοημοσύνης, που βασίζονται σε παραδείγματα από τη φύση, έχουν οδηγήσει στην ανάπτυξη μιας μεγάλης ποικιλίας αλγορίθμων και μεθόδων, όπως είναι τα Τεχνητά Νευρωνικά Δίκτυα, οι Γενετικοί Αλγόριθμοι (ΓΑ) και γενικότερα οι Εξελικτικοί Αλγόριθμοι (ΕΑ), τα Ασαφή Συστήματα (ΑΣ), η Ευφυΐα Σμήνους (ΕΣ) κλπ. Αυτές οι μέθοδοι είναι συνήθως υπολογιστικά αποδοτικές σε χρόνο και οδηγούν σε ικανοποιητικές λύσεις και —υπό προϋποθέσεις— σε βέλτιστες λύσεις. Την τελευταία εικοσαετία έχουν αναπτυχθεί υβριδικές μέθοδοι οι οποίες αξιοποιούν τα πλεονεκτήματα δύο ή περισσότερων από τις παραπάνω μεθόδους, εξαλείφοντας έτσι τα μειονεκτήματα που έχει η καθεμία ξεχωριστά.

Οι πρόσφατες εξελίξεις στην άμεση διαθεσιμότητα μεγάλου όγκου πληροφοριών, στη μείωση του κόστους του αποθηκευτικού χώρου και στην αύξηση της υπολογιστικής ισχύος έχουν οδηγήσει στην εκ νέου άνθιση και ευρεία εφαρμογή των τεχνικών και των μεθόδων της Υπολογιστικής Νοημοσύνης. Η επανάκαμψη αυτών των τεχνικών, που μπορεί να αποφεύγονταν στο παρελθόν, οφείλεται επομένως και στην έλευση των λεγόμενων Μεγάλων Δεδομένων (Big Data) και βρίσκει ιδιαίτερη έκφραση στη Βαθιά Μάθηση (Deep Learning): στον σχεδιασμό και στην εφαρμογή Νευρωνικών Δικτύων με πολλαπλά (περισσότερα του ενός) κρυφά επίπεδα μεταξύ του επιπέδου εξόδου και του επιπέδου εισόδου (Bengio, 2018) και αντίστοιχες βελτιστοποιήσεις στον *de facto* αλγόριθμο εκπαίδευσής τους, αυτόν της πίσω διάδοσης του λάθους.

Στο παρόν εκπαιδευτικό υλικό αρχικά παρουσιάζονται οι βασικές αρχές της τεχνολογίας, των ΤΝΔ και των ΓΑ. Γίνεται η σύνδεση με την Τεχνητή Νοημοσύνη (ΤΝ) και τις κλασικές μεθόδους αναζήτησης και βελτιστοποίησης, καθώς και με τα βιολογικά πρότυπα από τα οποία τα έχουν εμπνευστεί. Δίνονται τα βασικά στοιχεία της θεωρίας μάθησης και τα δύο παραδείγματα μάθησης (επιβλεπόμενη και μη επιβλεπόμενη μάθηση). Ακολουθεί η παρουσίαση των βασικών αλγορίθμων εκπαίδευσης των ΤΝΔ, με έμφαση στον πιο διαδεδομένο αλγόριθμο, της πίσω διάδοσης του λάθους (Error Back Propagation – EBP). Επίσης, συζητούνται θέματα αποδοτικής υλοποίησης των ΤΝΔ. Εδώ ολοκληρώνεται η εκπαίδευση με επιβλεπόμενη μάθηση. Ακολουθούν δύο παραδείγματα μη επιβλεπόμενης μάθησης. Παρουσιάζονται δύο αρχιτεκτονικές ΝΔ, τα δίκτυα Hopfield (ένα είδος αυτοσυσχετιστικής μνήμης) και τα δίκτυα Kohonen (ένα είδος αυτο-οργανούμενων χαρτών για ομαδοποίηση δεδομένων), καθώς και οι αλγόριθμοι εκπαίδευσής τους. Τα δίκτυα βαθιάς μάθησης (Deep Learning), εκπαίδευση και μοντέλα, ολοκληρώνουν την παρουσίαση των ΤΝΔ. Στη συνέχεια παρουσιάζονται οι Γενετικοί/Εξελικτικοί Αλγόριθμοι ΓΑ/ΕΑ, αλλά και ο συνδυασμός τους σε έναν υβριδικό αλγόριθμο, ένα Εξελικτικό Νευρωνικό Δίκτυο και η εισαγωγή στον Γενετικό Προγραμματισμό (ΓΠ), μια παραλλαγή των Εξελικτικών Αλγορίθμων, η οποία στηρίζεται στη βασική θεωρία της εξέλιξης του Δαρβίνου. Σε όλα τα κεφάλαια τονίζονται οι στόχοι και τα μαθησιακά αποτελέσματα και υπάρχει ικανοποιητικός αριθμός παραδειγμάτων, ασκήσεων και δραστηριοτήτων για την καλύτερη εμπέδωση της ύλης. Η κατανόηση της ύλης προϋποθέτει βασική γνώση διακριτών μαθηματικών, γραμμικής άλγεβρας, συνδυαστικής ανάλυσης και γνώση προγραμματισμού υπολογιστών. Όσοι θέλουν να ασχοληθούν με την επίλυση προβλημάτων με χρήση των παραπάνω τεχνολογιών πρέπει να φροντίσουν να αποκτήσουν αυτές τις δεξιότητες, διαφορετικά θα έχουν δυσκολίες και θα χρειαστεί να καταβάλουν μεγάλη προσπάθεια για την επιτυχή μελέτη του υλικού.

Όταν ο αναγνώστης ολοκληρώσει τη μελέτη αυτού του υλικού, δεν θα έχει γίνει προφανώς ειδικός στον τομέα της Υπολογιστικής Νοημοσύνης. Θα έχει αποκτήσει όμως όλες τις βασικές γνώσεις στο αντικείμενο και κυρίως, μέσα από τα παραδείγματα και τις εφαρμογές, θα έχει πάρει πολλές χρήσιμες ιδέες για το πώς μπορούν να χρησιμοποιηθούν αυτές οι τεχνολογίες στην επίλυση πραγματικών προβλημάτων του κόσμου. Είναι πολύ βασικό ο νέος επιστήμονας της Πληροφορικής να κατανοήσει έγκαιρα ότι πρέπει να ασχοληθεί συστηματικά με την επίλυση προβλημάτων. Όταν γίνει καλός σε αυτό, τότε μπορεί να γίνει πολύ καλός προγραμματιστής, πολύ καλός μηχανικός και πολύ καλός επιχειρηματίας, κάτι που συνιστά τον τελικό στόχο, που έδωσε το ερέθισμα για τη συγγραφή του συγκεκριμένου υλικού.

Τέλος, πρέπει να αναφέρουμε ότι σημαντικό τμήμα αυτού του υλικού βασίζεται στη μετάφραση από αγγλικό κείμενο, επομένως αποτέλεσε πρόκληση η ορθή απόδοση των αγγλικών όρων στην ελληνική γλώσσα. Έχουν αποδοθεί όλοι οι όροι και όπου δεν υπήρχε δόκιμη απόδοση στη ελληνική, δίνεται σε παρένθεση και ο αγγλικός όρος.

Σπύρος Λυκοθανάσης,
Δημήτρης Κουτσομητρόπουλος

Κεφάλαιο 1

Δυνατότητες και Εφαρμογές Νευρωνικών Δικτύων και Γενετικών Αλγορίθμων

Σύνοψη

Σε αυτό το κεφάλαιο, αφού γίνει πρώτα μία σύντομη αναφορά στις τεχνολογίες της Τεχνητής Νοημοσύνης (TN), θα ακολουθήσει η εισαγωγή στα Νευρωνικά Δίκτυα (ΝΔ) και στους Γενετικούς Αλγορίθμους (ΓΑ). Στόχος της ενότητας είναι η σύνδεση των ΝΔ όσο και των ΓΑ με την TN από την οποία προήλθαν.

Τα ΝΔ προσπαθούν να μιμηθούν τη λειτουργία του ανθρώπινου εγκέφαλου, όσον αφορά τον τρόπο μάθησης. Η παρουσίαση μερικών βασικών στοιχείων της λειτουργίας του εγκέφαλου είναι πολύ χρήσιμη για την κατανόηση των βασικών αρχών λειτουργίας των ΝΔ. Για να γίνει κατανοητή η λειτουργία ενός Γενετικού Αλγορίθμου είναι σκόπιμο να δοθούν ορισμένα στοιχεία από τη Θεωρία της Βιολογικής Γενετικής Εξέλιξης, από την οποία δανείζονται στοιχεία οι αλγόριθμοι αυτοί, και στη συνέχεια θα δούμε πώς στοιχεία της μπορούν να χρησιμοποιηθούν για να δομηθεί ένας αλγόριθμος αναζήτησης και βελτιστοποίησης. Επίσης, θα παρουσιαστούν τα βασικά πλεονεκτήματα αυτών των δύο κατηγοριών αλγορίθμων.

Μελετώντας το συγκεκριμένο κεφάλαιο ο αναγνώστης θα έχει εξοικειωθεί με τις έννοιες της Τεχνητής Νοημοσύνης, της Μηχανικής Μάθησης και της Αναζήτησης και Βελτιστοποίησης. Κυρίως, όμως, αυτό το κεφάλαιο θα τον εισαγάγει σε θέματα της θεωρίας της μάθησης και της γενετικής αναζήτησης, δηλαδή στο πώς ιδέες που προέρχονται από τη μάθηση και την εξέλιξη των βιολογικών ειδών μπορούν να χρησιμοποιηθούν σε πραγματικά προβλήματα μάθησης και αναζήτησης του βέλτιστου και να οδηγήσουν στη λύση τους γρήγορα και αποδοτικά. Επίσης, θα αναφερθούν οι περιοχές στις οποίες βρίσκουν εφαρμογή αυτοί οι αλγόριθμοι για την επίλυση πρακτικών προβλημάτων. Τέλος, θα γίνει μία σύντομη παρουσίαση των βιολογικών συστημάτων από τα οποία οι ερευνητές εμπνεύστηκαν τόσο τα ΝΔ όσο και τους ΓΑ.

Προσδοκώμενα αποτελέσματα:

Όταν θα έχετε τελειώσει τη μελέτη αυτού του κεφαλαίου, θα μπορείτε να:

- περιγράψετε τις τεχνολογίες της TN,
- περιγράψετε ποιοτικά τι είναι ένα νευρωνικό δίκτυο,
- περιγράψετε ποιοτικά τι είναι ένας γενετικός αλγόριθμος,
- περιγράψετε τις ομοιότητες και τις διαφορές με τα ανάλογα βιολογικά συστήματα,
- απαριθμήσετε μερικές περιοχές όπου βρίσκεται εφαρμογή η YN.

Λέξεις-Κλειδιά:

- Τεχνητά νευρωνικά δίκτυα
- Βιολογικά νευρωνικά δίκτυα
- Νευρώνας
- Συναπτικό βάρος ή σύναψη
- Κανόνας μάθησης
- Γενετικός αλγόριθμος
- Πληθυσμός
- Γενιά
- Χρωμόσωμα
- Γενετικοί τελεστές
- Διαδικασία επιλογής

Προαπαιτούμενη γνώση

Δεν απαιτούνται εξειδικευμένες γνώσεις.

1.1 Εισαγωγικές Παρατηρήσεις

Κατά τη διάρκεια των τελευταίων δεκαετιών υπάρχει ένα αυξανόμενο ενδιαφέρον σε αλγόριθμους οι οποίοι βασίζονται σε αναλογίες με τις φυσικές διαδικασίες. Η εμφάνιση των μαζικά παράλληλων ηλεκτρονικών υπολογιστών έκανε αυτούς τους αλγόριθμους να αποκτήσουν πρακτικό ενδιαφέρον. Στους πιο γνωστούς αλγόριθμους αυτής της κατηγορίας περιλαμβάνονται τα νευρωνικά δίκτυα, οι γενετικοί αλγόριθμοι, ο εξελικτικός προγραμματισμός, οι εξελικτικές στρατηγικές, οι μέθοδοι τοπικής αναζήτησης και τα συστήματα ταξινόμησης. Οι παραπάνω αλγόριθμοι, αν και έγιναν γνωστοί ως τεχνολογίες που χρησιμοποιούνται στην Τεχνητή Νοημοσύνη (Artificial Intelligence) και πιο συγκεκριμένα στη Μάθηση Μηχανής (Machine Learning), πρόσφατα αναφέρονται στη διεθνή βιβλιογραφία ως αλγόριθμοι Υπολογιστικής Νοημοσύνης (Computational Intelligence).

Αυτό το εκπαιδευτικό υλικό πραγματεύεται δύο κλάσεις αυτών των αλγορίθμων. Πρώτα θα αναφερθούμε σε τεχνικές που βασίζονται στη θεωρία της μάθησης, δηλαδή τα Νευρωνικά Δίκτυα (Ν.Δ.). Τα Ν.Δ. εκπαιδεύονται με τη βοήθεια παραδειγμάτων, έτσι ώστε να μαθαίνουν το περιβάλλον τους. Ένα παράδειγμα περιλαμβάνει την είσοδο και την επιθυμητή έξοδο σε αυτή. Το σύνολο των παραδειγμάτων αποτελεί το εκπαιδευτικό σύνολο. Για την εκπαίδευση χρησιμοποιούνται κανόνες, οι οποίοι βασίζονται στην ελαχιστοποίηση του λάθους στην έξοδο του δικτύου. Ακολουθεί η γενίκευση, δηλαδή τα Ν.Δ. μαθαίνουν παραδείγματα για τα οποία δεν έχουν εκπαιδευτεί. Υπάρχουν πολλές κατηγορίες Ν.Δ., ανάλογα με την αρχιτεκτονική τους και τον τρόπο εκπαίδευσής τους. Επίσης, υπάρχει μεγάλη ποικιλία αλγορίθμων εκπαίδευσης, ανάλογα με τον κανόνα μάθησης και τον αλγόριθμο ελαχιστοποίησης που χρησιμοποιείται.

Στη συνέχεια θα αναφερθούμε σε αλγόριθμους που βασίζονται στην αρχή της εξέλιξης (επιβίωση του καταλληλότερου). Σε αυτούς τους αλγόριθμους ένας πληθυσμός από άτομα (υποψήφιες λύσεις) υποβάλλεται σε μια ακολουθία από μοναδιαίους (τύπου μετάλλαξης) και υψηλότερης τάξης (τύπου διασταύρωσης) μετασχηματισμούς. Αυτά τα άτομα ανταγωνίζονται για επιβίωση: ένα σχήμα επιλογής προκατειλημμένο υπέρ των καταλληλότερων ατόμων, επιλέγει την επόμενη γενιά. Μετά από έναν αριθμό επαναλήψεων, το πρόγραμμα συγκλίνει, ελπίζοντας ότι το καλύτερο άτομο αναπαριστά τη βέλτιστη λύση. Υπάρχουν πολλοί διαφορετικοί αλγόριθμοι σε αυτήν την κατηγορία. Για να υπογραμμίσουμε τις ομοιότητές τους χρησιμοποιούμε τον κοινό όρο «εξελικτικά προγράμματα». Τα εξελικτικά προγράμματα μπορούν να θεωρηθούν σαν γενίκευση των Γενετικών Αλγορίθμων. Οι κλασικοί Γενετικοί Αλγόριθμοι λειτουργούν με σταθερού μήκους δυαδικές συμβολοσειρές, οι οποίες δεν είναι απαραίτητες για τα εξελικτικά προγράμματα. Επίσης, τα εξελικτικά προγράμματα συνήθως ενσωματώνουν μία ποικιλία από «γενετικούς» τελεστές, ενώ οι Γενετικοί Αλγόριθμοι χρησιμοποιούν δυαδική κωδικοποίηση, δηλαδή συμβολοσειρές δυαδικών ψηφίων στις οποίες εφαρμόζονται οι κατάλληλοι τελεστές για τη διασταύρωση και τη μετάλλαξη των ατόμων του πληθυσμού.

1.2 Εισαγωγή στην Τεχνητή Νοημοσύνη

Η γεφύρωση του χάσματος ανάμεσα στον άνθρωπο και στις μηχανές αποτελεί τον μεγαλύτερο στόχο της Επιστήμης των Υπολογιστών. Όμως, ακόμη και σήμερα, αν και η επιστήμη αυτή έχει εμφανίσει αλματώδη ανάπτυξη, δεν έχει καταφέρει να παρουσιάσει ηλεκτρονικούς υπολογιστές απαλλαγμένους από τα κλασικά μειονεκτήματα μιας μηχανής. Έτσι, εκτός από ελάχιστες εξαιρέσεις, δεν υπάρχουν σήμερα μηχανές που να είναι σε θέση να επικοινωνήσουν με τον άνθρωπο σε φυσική γλώσσα, να απαντούν σε ερωτήσεις για διάφορα συγκεκριμένα προβλήματα (εκτός και εάν είναι εφοδιασμένοι με μία κατάλληλη μέθοδο επίλυσης) ή να αποκτούν εμπειρίες και να μαθαίνουν από τις αποτυχίες και τα λάθη τους. Η περιοχή έρευνας που ασχολείται με αυτού του είδους τα προβλήματα καλείται Τεχνητή Νοημοσύνη.

Τεχνητή Νοημοσύνη (ΤΝ) ονομάζεται η μελέτη των τεχνικών και των διεργασιών που δίνουν σε έναν υπολογιστή τη δυνατότητα να αποκτά διανοητικές ικανότητες, ανάλογες με αυτές που διαθέτει ο ανθρώπινος εγκέφαλος. Η ΤΝ έχει ως κύριο στόχο της να γίνει ο υπολογιστής πιο έξυπνος και κατ' επέκταση πιο χρήσιμος, αφού θα είναι σε θέση να ανταποκρίνεται πολύ καλύτερα στις ανάγκες και τις επιθυμίες του ανθρώπου.

Σήμερα ο όρος ΤΝ χρησιμοποιείται τόσο για τη μελέτη της ανθρώπινης νοημοσύνης, όσο και για τη βελτίωση των δυνατοτήτων των υπολογιστών. Οι κυριότερες εφαρμογές της είναι οι εξής:

- 1) Ο προγραμματισμός μιας ακολουθίας ενεργειών για την επίτευξη ενός στόχου (planning).
- 2) Η εξαγωγή συμπερασμάτων (inference) μέσα από αλληλοσυσχετιζόμενα γεγονότα και η λήψη αποφάσεων (decision making).

- 3) Η παροχή συμβουλών και συμπερασμάτων μέσα από σύνθετες δομές κανόνων και γεγονότων (expert systems).
- 4) Η εκπαίδευση των υπολογιστών για επικοινωνία με τους ανθρώπους μέσω φυσικών γλωσσών. Αυτό περιλαμβάνει μια ποικιλία εφαρμογών, όπως αναγνώριση φωνής, παραγωγή φωνής, κατανόηση κειμένου κλπ.
- 5) Η αυτόνομη κίνηση των υπολογιστών και η μετακίνηση από αυτούς αντικειμένων μέσα στον χώρο (robotics).
- 6) Η αναγνώριση αντικειμένων μέσω κάμερας (vision).

Όπως είναι φυσικό, η πρόοδος στον τομέα της ΤΝ σημειώνεται με αργούς ρυθμούς, διότι απαιτεί πολυετή και επίπονη έρευνα. Είναι επίσης φανερό ότι η κατασκευή μιας μηχανής με ανθρώπινες ικανότητες και ιδιότητες είναι μια ιδιαίτερα δύσκολη εργασία. Τα μέχρι τώρα επιτεύγματα, όμως, αν και είναι μακριά από την τελειοποίησή τους, μπορούν να χαρακτηριστούν αρκετά ικανοποιητικά.

Οι περιοχές έρευνας της ΤΝ που συγκεντρώνουν το μεγαλύτερο ενδιαφέρον σήμερα είναι οι εξής:

- Νευρωνικά Δίκτυα (Neural Networks)
- Γενετικοί Αλγόριθμοι (Genetic Algorithms)
- Έμπειρα Συστήματα (Expert Systems)
- Ασαφή Συστήματα (Fuzzy Systems).

Άσκηση 1.2/1: Ποιες είναι οι κυριότερες εφαρμογές της ΤΝ;

1.2.1 Νευρωνικά Δίκτυα

Τα Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ) ή απλά Νευρωνικά Δίκτυα (ΝΔ) αποτελούν μια προσπάθεια προσέγγισης της λειτουργίας του ανθρώπινου εγκεφάλου από μια μηχανή. Έχουν την ικανότητα να εκτελούν υπολογισμούς με μαζικό παράλληλο τρόπο. Η αρχιτεκτονική τους βασίζεται στην αρχιτεκτονική των Βιολογικών Νευρωνικών Δικτύων. Τα ΤΝΔ είναι μια συλλογή από νευρώνες (Processing Units-PU) που συνδέονται μεταξύ τους. Κάθε PU έχει πολλές εισόδους αλλά μόνο μία έξοδο, η οποία με τη σειρά της μπορεί να αποτελέσει είσοδο για άλλες PUs. Οι συνδέσεις μεταξύ των PUs διαφέρουν ως προς τη σημαντικότητά τους, η οποία και προσδιορίζεται από τον συντελεστή βάρους (σύναψη). Η επεξεργασία κάθε PU καθορίζεται από τη συνάρτηση μεταφοράς που καθορίζει την κάθε έξοδο σε σχέση με τις εισόδους και τους συντελεστές βάρους.

Για να χρησιμοποιηθεί ένα ΤΝΔ πρέπει πρώτα να εκπαιδευτεί για να μάθει το μοντέλο που παράγει τα δεδομένα. Η μάθηση συνίσταται στον προσδιορισμό των κατάλληλων συντελεστών βάρους, ώστε το ΤΝΔ να εκτελεί τους επιθυμητούς υπολογισμούς, και πραγματοποιείται με τη βοήθεια αλγορίθμων που είναι γνωστοί ως κανόνες μάθησης. Ο ρόλος των συντελεστών βάρους μπορεί να ερμηνευτεί ως αποθήκευση γνώσης, η οποία παρέχεται μέσω παραδειγμάτων. Με αυτόν τον τρόπο τα ΝΔ μαθαίνουν το περιβάλλον τους, δηλαδή το φυσικό μοντέλο που παρέχει τα δεδομένα.

Γνωστά ΤΝΔ είναι τα ΤΝΔ χωρίς ανατροφοδότηση, π.χ. τα εμπρός τροφοδότησης ΝΔ πολλών επιπέδων (feedforward multilayer Neural Networks), με ανατροφοδότηση, π.χ. το μοντέλο Hopfield, τα Cellular neural networks κ.ά.

1.2.2 Γενετικοί Αλγόριθμοι

Τα τελευταία τριάντα χρόνια έχει παρατηρηθεί ένα συνεχώς αυξανόμενο ενδιαφέρον για ανάπτυξη συστημάτων επίλυσης προβλημάτων βασισμένων στις αρχές της *Γενετικής Εξέλιξης* και της *Κληρονομικότητας*. Τα μειονεκτήματα των κλασικών μεθόδων αναζήτησης και βελτιστοποίησης, καθώς και η διαρκώς αυξανόμενη ανάγκη για παραγωγή λογισμικού που να μπορεί να εκμεταλλεύεται πιο αποδοτικά τις τεράστιες δυνατότητες του υλικού, ήταν η βασική αιτία που ώθησε τους επιστήμονες σε αυτήν την αναζήτηση. Αυτού του είδους τα συστήματα λειτουργούν διατηρώντας έναν πληθυσμό κωδικοποιημένων πιθανών λύσεων και εφαρμόζοντας πάνω του διάφορες διαδικασίες επιλογής του καλύτερου, καθώς και διάφορους *γενετικούς τελεστές*. Οι συγκεκριμένοι τελεστές αντιγράφουν τον τρόπο με τον οποίο αναπαράγονται και μεταλλάσσονται τα χρωμοσώματα των κυττάρων των ζωντανών οργανισμών. Έτσι, περνώντας από γενιά σε γενιά, τα συστήματα αυτά δημιουργούν συνεχώς νέους πληθυσμούς πιθανών λύσεων χρησιμοποιώντας τόσο κομμάτια και στοιχεία

από την προηγούμενη γενιά, όσο και εντελώς καινούρια κομμάτια που δοκιμάζονται για τυχόν καλή απόδοσή τους. Με αυτόν τον τρόπο αξιοποιούν τις πληροφορίες που τους παρέχει το περιβάλλον τους.

Επανειλημμένες δοκιμές και πειράματα έχουν δείξει ότι μια «φυσική» αναπαράσταση των πιθανών λύσεων για ένα δεδομένο πρόβλημα σε συνδυασμό με την εφαρμογή σε αυτή μιας οικογένειας γενετικών τελεστών αποτελεί εξαιρετικά χρήσιμο εργαλείο στην προσπάθεια προσέγγισης των πραγματικών λύσεων σε μια πολύ μεγάλη ποικιλία προβλημάτων και εφαρμογών. Αυτό το γεγονός μετατρέπει την προσέγγιση «φυσικού μοντέλου» σε μια πολλά υποσχόμενη κατεύθυνση όσον αφορά την επίλυση προβλημάτων γενικότερα.

1.2.3 Έμπειρα Συστήματα

Τα Έμπειρα Συστήματα (ΕΣ) αποσκοπούν στην πραγματοποίηση συστημάτων υπολογιστών με δυνατότητες αυτόματης επεξεργασίας των δεδομένων ενός προβλήματος και την εξαγωγή συγκεκριμένων συμπερασμάτων. Χρησιμοποιούν προγραμματιστικές μεθόδους της ΤΝ σε συνδυασμό με τις γνώσεις εμπειρογνομόνων σε ειδικά θέματα. Ένα ΕΣ αποτελείται από τη βάση γνώσης (knowledge base) και τον μηχανισμό συμπεράσματος (inference engine), ο οποίος μπορεί να έχει διάφορες μορφές ανάλογα με την εφαρμογή.

1.2.4 Ασαφή Συστήματα

Η γνώση που λαμβάνει ένας υπολογιστής για να συμπεριφερθεί έξυπνα αναπαρίσταται με τη μορφή κανόνων (rules) και γεγονότων (facts). Στην πράξη όμως οι κανόνες και τα γεγονότα δεν παίρνουν πάντα την τιμή 0 ή 1, αλλά ισχύουν με πιθανότητες. Το γεγονός αυτό οδήγησε στην ανάπτυξη μιας σύγχρονης μαθηματικής λογικής που αποτελεί επέκταση της απλής άλγεβρας Boole και ονομάζεται Ασαφής Λογική (Fuzzy Logic). Η Ασαφής Λογική εισάγει στον λογικό προγραμματισμό τις μη ακέραιες λογικές τιμές που ανήκουν στο διάστημα $[0,1]$ και ορίζει τελεστές για τον συνδυασμό τους.

Τα συστήματα που αναπαριστούν τη γνώση και την ανθρώπινη λογική με βάση την Ασαφή Λογική, ονομάζονται Ασαφή Συστήματα (Fuzzy Systems).

Άσκηση 1.2/2: Ένα νευρωνικό δίκτυο μπορεί να θεωρηθεί ως έμπειρο σύστημα;

Άσκηση 1.2/3: Στις σελ. 16-17 είδαμε μερικές εφαρμογές της ΤΝ. Σε ποιες από αυτές μπορούν να εφαρμοστούν τα ΝΔ και σε ποιες οι ΓΑ;

1.3 Στοιχεία της Θεωρίας Μάθησης

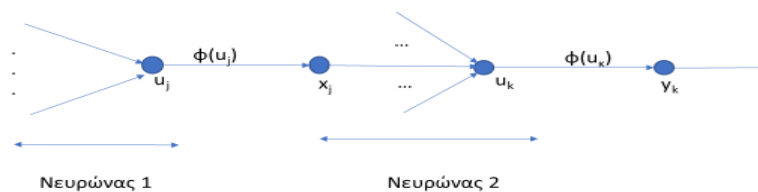
Ανάμεσα στις πολλές ενδιαφέρουσες ιδιότητες ενός νευρωνικού δικτύου αυτή με τη μεγαλύτερη σπουδαιότητα είναι η ικανότητά του να μαθαίνει από το περιβάλλον του και έτσι να βελτιώνει την απόδοσή του μέσω της μάθησης. Η βελτίωση αυτή γίνεται σταδιακά, με τον χρόνο, σύμφωνα με κάποιο καθορισμένο μέτρο. Η μάθηση επιτυγχάνεται μέσω μιας επαναληπτικής διαδικασίας ρυθμίσεων της τιμής των συναπτικών βαρών και των κατωφλίων. Θεωρητικά το δίκτυο αποκτά περισσότερη γνώση για το περιβάλλον του μετά από κάθε επανάληψη της διαδικασίας μάθησης.

Ο ορισμός μια έννοιας, όπως αυτή της μάθησης εξαρτάται από τη σκοπιά που θα την εξετάσουμε. Εμείς —με σημείο αναφοράς τα νευρωνικά δίκτυα— χρησιμοποιούμε τον ορισμό των Mendel και McClaren (1970) (Haykin, 1999):

Μάθηση είναι μια διαδικασία με την οποία προσαρμόζονται οι ελεύθερες παράμετροι ενός νευρωνικού δικτύου μέσω μίας συνεχούς διαδικασίας διέγερσης από το περιβάλλον στο οποίο βρίσκεται το δίκτυο. Το είδος της μάθησης καθορίζεται από τον τρόπο με τον οποίο πραγματοποιούνται οι αλλαγές των παραμέτρων.

Ο παραπάνω ορισμός της διαδικασίας μάθησης υπονοεί την ακόλουθη σειρά βημάτων:

1. Το νευρωνικό δίκτυο «διεγείρεται» από ένα περιβάλλον.
2. Το νευρωνικό δίκτυο υφίσταται αλλαγές ως συνέπεια αυτής της διέγερσης.
3. Το νευρωνικό δίκτυο «απαντά» με έναν καινούριο τρόπο στο περιβάλλον, λόγω των αλλαγών που συνέβησαν στην εσωτερική του δομή.



Σχήμα 1.1 Ζεύγος κομβικών σημάτων.

Πιο συγκεκριμένα, ας αναλύσουμε την παραπάνω περιγραφή με ένα παράδειγμα. Θεωρούμε ένα ζεύγος κομβικά σήματα x_j και u_k που συνδέονται με βάρος σύνδεσης w_{kj} , όπως απεικονίζεται στο σχήμα 1.1. Το x_j αναπαριστά την έξοδο του νεύρωνα j, ενώ το u_k την εσωτερική λειτουργία του νεύρωνα k. Με βάση το βάρος σύνδεσης w_{kj} , τα σήματα x_j και u_k αναφέρονται συχνά ως προσυναπτική και μετασυναπτική λειτουργία αντίστοιχα. Ας συμβολίσουμε με $w_{kj}(n)$ την τιμή του βάρους σύνδεσης w_{kj} τη χρονική στιγμή n. Τότε γίνεται μια ρύθμιση (διόρθωση) $\Delta W_{kj}(n)$ στο βάρος $w_{kj}(n)$ και παράγεται η νέα ενημερωμένη τιμή $w_{kj}(n+1)$. Έτσι προκύπτει η εξίσωση:

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \quad (1.1)$$

όπου $w_{kj}(n)$ και $w_{kj}(n+1)$, η παλιά και η νέα τιμή του βάρους σύνδεσης w_{kj} αντίστοιχα. Η εξίσωση (1.1) συγκεντρώνει τη συνολική επίδραση των βημάτων 1 και 2 που προκύπτουν από τον ορισμό της διαδικασίας μάθησης και αναφέρθηκαν παραπάνω. Πιο συγκεκριμένα η ρύθμιση $\Delta W_{kj}(n)$ υπολογίζεται ως αποτέλεσμα της διέγερσης από το περιβάλλον (βήμα 1) και η ενημερωμένη τιμή $w_{kj}(n+1)$ εκφράζει την αλλαγή που συνέβη στο δίκτυο ως αποτέλεσμα αυτής της διέγερσης (βήμα 2). Το βήμα 3 πραγματοποιείται όταν υπολογίζεται η απάντηση του νέου δικτύου που λειτουργεί με το ενημερωμένο σύνολο παραμέτρων $\{w_{kj}(n+1)\}$.

Ένα καθορισμένο σύνολο από καλά ορισμένους κανόνες για τη λύση ενός προβλήματος μάθησης καλείται αλγόριθμος μάθησης (learning algorithm). Όπως είναι φανερό, δεν υπάρχει ένας μοναδικός τέτοιος αλγόριθμος για τον σχεδιασμό νευρωνικών δικτύων. Αντίθετα, υπάρχει ένα σύνολο από εργαλεία που αναπαρίστανται από μια μεγάλη ποικιλία αλγορίθμων μάθησης, καθένας από τους οποίους έχει τα δικά του πλεονεκτήματα. Βασικά οι αλγόριθμοι διαφέρουν μεταξύ τους στον τρόπο που εκφράζεται η ρύθμιση Δw_{kj} στο βάρος σύνδεσης w_{kj} . Ένας ακόμη παράγοντας που πρέπει να εξετάσουμε είναι ο τρόπος που το νευρωνικό δίκτυο σχετίζεται με το περιβάλλον. Στη συνέχεια όταν αναφερόμαστε σε ένα παράδειγμα μάθησης (learning paradigm), θα εννοούμε το μοντέλο του περιβάλλοντος στο οποίο λειτουργεί το δίκτυο.

Άσκηση 1.3/4: Μάθηση είναι η διαδικασία:

- A. κατά την οποία το ΝΔ «απαντά» με έναν καινούριο τρόπο στο περιβάλλον, λόγω των αλλαγών που συνέβησαν στην εσωτερική του δομή,
- B. κατά την οποία προσαρμόζονται οι ελεύθερες παράμετροι του δικτύου,
- C. κατά την οποία το ΝΔ υφίσταται αλλαγές ως συνέπεια μιας εξωτερικής διέγερσης.

Να επιλέξετε τη σωστή απάντηση.

Άσκηση 1.3/5: Αν η ρύθμιση που υπολογίστηκε για το βάρος $w_{kj}(n)$ είναι αρνητική, τι επίδραση θα έχει αυτό στη νέα τιμή του βάρους;

Δραστηριότητα 1/1: Στο δεύτερο κεφάλαιο της αναφοράς (S. Haykin, 1999) γίνεται μία ολοκληρωμένη παρουσίαση της θεωρίας της μάθησης. Εκεί θα δείτε ότι αναφέρονται διάφοροι κανόνες για τον υπολογισμό της ρύθμισης ΔW . Να κάνετε μία σύντομη παρουσίαση αυτών των κανόνων μάθησης, σε περίπου δύο σελίδες.

1.4 Μέθοδοι Αναζήτησης και Βελτιστοποίησης

Μετά την εισαγωγή στους Γενετικούς Αλγορίθμους, κρίνουμε σκόπιμο να κάνουμε για λόγους σύγκρισης μία πολύ σύντομη παρουσίαση των κυριότερων παραδοσιακών μεθόδων αναζήτησης και βελτιστοποίησης. Οι μέθοδοι που παρουσίασαν αξιόλογα αποτελέσματα, όσον αφορά την εφαρμογή τους σε υπολογιστικές μηχανές, και κυριάρχησαν για πολλά χρόνια είναι οι εξής:

Μέθοδοι βασισμένες στον λογισμό (calculus-based methods): Έχουν γίνει αντικείμενο ευρείας μελέτης. Χωρίζονται σε δύο βασικές κατηγορίες, τις έμμεσες και τις άμεσες. Οι έμμεσες ασχολούνται με την εύρεση τοπικών ακρότατων, επιλύοντας συνήθως ένα σύνολο μη γραμμικών συναρτήσεων. Οι άμεσες, από την πλευρά τους, ψάχνουν για τοπικά ακρότατα κάνοντας μικρά άλματα στη συνάρτηση (hillclimbing). Αν και αρκετά δοκιμασμένες, και οι δύο κατηγορίες παρουσιάζουν σημαντικά μειονεκτήματα. Το βασικότερο από αυτά είναι ότι εμφανίζουν τοπικότητα στην εμβέλεια: δηλαδή το ακρότατο το οποίο βρίσκουν είναι το καλύτερο στη γειτονιά ενός σημείου.

Απαριθμητικές (enumerative) ή τυχαίες (random) μέθοδοι: Συναντώνται σε πολλές μορφές και σε διάφορα προβλήματα. Μέσα σε έναν πεπερασμένο (ή άπειρα διακριτό) χώρο αναζήτησης, αναζητούνται κάποια βέλτιστα σημεία με ψάξιμο ένα προς ένα. Παρόλο που η απλότητα εδώ είναι ελκυστική, η αποδοτικότητα είναι προφανές ότι είναι πολύ χαμηλή, κάτι που δεν τις κάνει ιδιαίτερα δημοφιλείς. Σχεδόν ποτέ δεν χρησιμοποιούνται μόνες τους, αλλά σε συνδυασμό με άλλες αποδοτικότερες μεθόδους.

Μέθοδοι επαναληπτικής αναζήτησης (iterated search): Πρόκειται για έναν παραγωγικό συνδυασμό των μεθόδων των δύο προηγούμενων κατηγοριών. Μόλις το hillclimbing εντοπίσει μια κορυφή (τοπικό μέγιστο ή ελάχιστο), επιλέγεται τυχαία ένα νέο σημείο και αρχίζει ξανά η ίδια διαδικασία για τον εντοπισμό μιας νέας κορυφής. Αυτό γίνεται αρκετές φορές κρατώντας πάντα την καλύτερη τιμή που έχει βρεθεί. Η τεχνική αυτή έχει το πλεονέκτημα της απλότητας, δεν υπάρχει περίπτωση παγίδευσης, αλλά όταν τα τοπικά μέγιστα είναι πολλά, η απόδοσή της πέφτει σημαντικά.

Προσομοιωμένη Ανόπτηση (Simulated Annealing): Αποτελεί μια τροποποιημένη έκδοση του hillclimbing. Τα μειονεκτηματά της είναι ότι ασχολείται με μόνο μια λύση σε κάθε βήμα, ενώ δεν κάνει αξιοποίηση της πληροφορίας που έχει υποστεί επεξεργασία σε προηγούμενα στάδια, μην αποκτώντας έτσι μια γενική εικόνα του χώρου αναζήτησης.

Δυναμικός προγραμματισμός (Dynamic Programming): Αποτελεί προγραμματιστική τεχνική που βρίσκει εφαρμογή σε περιορισμένη περιοχή προβλημάτων. Χρησιμοποιείται κυρίως για τη βελτιστοποίηση της λύσης ενός προβλήματος πολλαπλών φάσεων, για καθεμία από τις οποίες είναι διαθέσιμος ένας αριθμός εναλλακτικών αποφάσεων. Είναι αυτονόητο ότι δεν αποτελεί ισχυρό εργαλείο βελτιστοποίησης λόγω της υπερβολικής εξειδίκευσης για μικρό εύρος προβλημάτων.

Ευρετικές μέθοδοι (heuristic methods): Ευρετική ονομάζεται κάθε μη αλγοριθμική μέθοδος επίλυσης προβλημάτων, στην οποία η πορεία προς ένα τελικό αποδεκτό αποτέλεσμα στηρίζεται σε μια σειρά προσεγγιστικών αποτελεσμάτων. Αν και οι ευρετικές μέθοδοι δίνουν απλές και ικανοποιητικές λύσεις σε μερικά προβλήματα, τίποτα δεν εγγυάται ότι είναι οι καλύτερες δυνατές. Συνήθως δίνουν προσεγγίσεις των βέλτιστων λύσεων και κάποιες φορές προτιμούνται επειδή δίνουν αποδεκτές απαντήσεις σε σύντομο χρόνο. Συνεπώς δεν μπορούν να αποτελέσουν κύριο εργαλείο βελτιστοποίησης.

Αν και δεν έχουν εξεταστεί εξαντλητικά όλες οι μέθοδοι αναζήτησης, το συμπέρασμα που προκύπτει από την παραπάνω σύντομη παρουσίαση είναι πως οι μέθοδοι αυτές δεν έχουν την ισχύ για να αντεπεξέλθουν σε μεγάλο αριθμό προβλημάτων, χωρίς αυτό να σημαίνει ότι είναι άχρηστες. Αντιθέτως έχουν δώσει λύσεις σε πολλές περιπτώσεις μέχρι σήμερα. Καθώς όμως παρουσιάζονται ολοένα και δυσκολότερα προβλήματα, τόσο πιο επιτακτική γίνεται η ανάγκη για εύρεση νέων μεθόδων αναζήτησης και βελτιστοποίησης. Μια τέτοια κατηγορία αλγορίθμων που βασίζονται σε αναλογίες με τις φυσικές διαδικασίες της εξέλιξης είναι οι Γενετικοί Αλγόριθμοι που παρουσιάζονται στο όγδοο κεφάλαιο.

Άσκηση 1.4/6: Ποιο είναι το βασικό μειονέκτημα των κλασικών μεθόδων αναζήτησης;

1. έχουν μεγάλη πολυπλοκότητα,
2. κάνουν τυχαίο ψάξιμο,
3. κάνουν αναζήτηση σε ένα μόνο σημείο σε κάθε βήμα,
4. δεν αξιοποιούν την πληροφορία που παρέχει το περιβάλλον,
5. είναι τοπικής εμβέλειας.

Να επιλέξετε τη(ις) σωστή(ές) απάντηση(εις).

1.5 Εφαρμογές των ΝΔ

Η ευρεία εφαρμογή των νευρωνικών δικτύων σε μία ποικιλία προβλημάτων, σε πολλά διαφορετικά πεδία τα κάνει πολύ ελκυστικά. Επίσης, οι πιο γρήγοροι υπολογιστές και αλγόριθμοι έχουν κάνει δυνατή τη χρήση νευρωνικών δικτύων στην επίλυση πολύπλοκων βιομηχανικών προβλημάτων, τα οποία απαιτούσαν πάρα πολλούς υπολογισμούς. Από την πρώτη τους εφαρμογή σε προσαρμοζόμενους εξισωτές καναλιών το 1984 μέχρι σήμερα έχουν αξιοποιηθεί σε μεγάλη ποικιλία εφαρμογών όπως φαίνεται παρακάτω (Hagan et al., 1996):

- Αεροπορία: Υψηλής απόδοσης αυτόματοι πιλότοι αεροπλάνων, προσομοιωτές πτήσης, συστήματα αυτόματου ελέγχου αεροπλάνων, συστήματα ανίχνευσης βλαβών.
- Αυτοκίνηση: Αυτοκινούμενα συστήματα αυτόματης πλοήγησης.
- Τραπεζικές εφαρμογές: Αναγνώστες επιταγών και άλλων παραστατικών, συστήματα αξιολόγησης αιτήσεων δανειοδότησης.
- Άμυνα: Πλοήγηση όπλων, ανίχνευση στόχων, νέα είδη αισθητήρων, σόναρ, ραντάρ, ψηφιακή επεξεργασία σημάτων, συμπίεση δεδομένων, εξαγωγή χαρακτηριστικών, αναγνώριση σήματος/εικόνας.
- Ηλεκτρονική: Πρόβλεψη ακολουθίας κωδικών, μορφοποίηση ολοκληρωμένων κυκλωμάτων, έλεγχος διεργασιών, διάγνωση βλαβών ολοκληρωμένων κυκλωμάτων, μηχανική όραση, σύνθεση φωνής.
- Οικονομία: Οικονομική ανάλυση, πρόβλεψη τιμών συναλλάγματος.
- Κοινωνική ασφάλιση: Αξιολόγηση εφαρμοζόμενης πολιτικής, βελτιστοποίηση παραγωγής.
- Βιομηχανία: Βιομηχανικός έλεγχος διεργασιών, ανάλυση και σχεδίαση προϊόντων, συστήματα ποιοτικού ελέγχου, διάγνωση βλαβών διεργασιών και μηχανών, ανάλυση σχεδιασμού χημικών προϊόντων, δυναμικό μοντελάρισμα συστημάτων χημικών διεργασιών, σχεδιασμός και διοίκηση.
- Ιατρική: Ανάλυση καρκινικών κυττάρων, ανάλυση ηλεκτροεγκεφαλογραφήματος και ηλεκτροκαρδιογραφήματος, βελτιστοποίηση χρόνου νοσηλείας, μείωση νοσοκομειακού κόστους, βελτίωση ποιότητας νοσοκομείων.
- Γεωλογικές έρευνες: Εντοπισμός πετρελαίου και φυσικού αερίου.
- Ρομποτική: Έλεγχος τροχιάς και σύστημα όρασης ρομπότ.
- Επεξεργασία φωνής: Αναγνώριση φωνής, συμπίεση φωνής, σύνθεση φωνής από κείμενο.
- Χρηματιστηριακές εφαρμογές: Ανάλυση αγοράς, πρόβλεψη τιμών μετοχών.
- Τηλεπικοινωνίες: Συμπύεση εικόνας και δεδομένων, αυτοματοποιημένες υπηρεσίες πληροφοριών, μετάφραση πραγματικού χρόνου, συστήματα επεξεργασίας πληρωμών.
- Μεταφορές: Συστήματα διάγνωσης βλαβών φρένων, χρονοπρογραμματισμός οχημάτων, συστήματα δρομολόγησης.

Από το παραπάνω πλήθος εφαρμογών μπορούμε να συμπεράνουμε ότι τα νευρωνικά δίκτυα έχουν αναπτυχθεί ταχύτατα. Γι' αυτόν τον λόγο έχουν επενδυθεί πολλά χρήματα τόσο για την ανάπτυξη λογισμικού, όσο και για την ανάπτυξη υλικού.

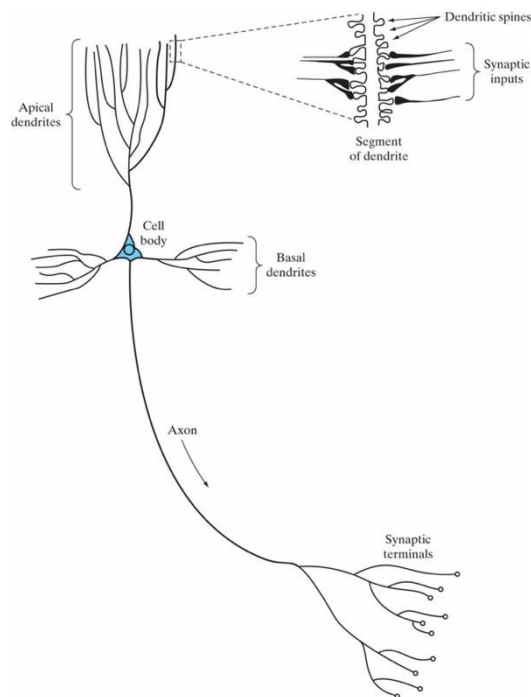
1.6 Η Έμπνευση από τα Βιολογικά Συστήματα

1.6.1 Από τα Βιολογικά Νευρωνικά Δίκτυα στα Τεχνητά

Η μελέτη των τεχνητών νευρωνικών δικτύων, κοινώς γνωστά ως «νευρωνικά δίκτυα», υποκινήθηκε άμεσα από την αναγνώριση ότι οι εγκεφαλικοί υπολογιστές είναι ένας εξολοκλήρου διαφορετικός δρόμος από τους συμβατικούς ψηφιακούς υπολογιστές. Τυπικά οι νευρώνες του εγκεφάλου είναι 5-6 τάξεις μεγέθους αργότεροι από τις λογικές πύλες πυριτίου. Ο εγκέφαλος είναι ένας εξαιρετικά πολύπλοκος, μη-γραμμικός και παράλληλος υπολογιστής. Έχει τη δυνατότητα να οργανώνει τους νευρώνες έτσι ώστε να εκτελεί συγκεκριμένους υπολογισμούς πολύ πιο γρήγορα από τους πιο γρήγορους ψηφιακούς υπολογιστές που υπάρχουν.

Πώς τα καταφέρνει όμως ο εγκέφαλος; Κατά τη γέννησή του ο εγκέφαλος έχει την ικανότητα να κατασκευάζει τους δικούς του κανόνες, κοινώς «εμπειρία», η οποία μεγαλώνει με την πάροδο του χρόνου. Κατά τα δύο πρώτα χρόνια ζωής έχουμε τη μέγιστη ανάπτυξη, όπου περίπου 1 εκατομμύριο συνάψεις (synapses) δημιουργούνται στο δευτερόλεπτο.

Οι συνάψεις είναι οι βασικές δομικές και λειτουργικές μονάδες που μεσολαβούν στην ενδοεπικοινωνία των νευρώνων, όπως φαίνεται στο σχήμα 1.2 που δείχνει τη βασική δομή ενός νευρώνα, και είναι δανεισμένο από την αναφορά (S. Haykin, 1999). Αρα ένα νευρωνικό δίκτυο είναι ένας συμπαγής παράλληλος καταναεμημένος επεξεργαστής, που έχει τη φυσική κλίση να αποθηκεύει εμπειριστατωμένη γνώση και να την κάνει διαθέσιμη για χρήση.



Σχήμα 1.2 Η βασική δομή ενός βιολογικού νευρώνα (Haykin, 1999).

Ένα ΤΝΔ μοιάζει με τον εγκέφαλο στα εξής:

- Η γνώση αποκτάται από το δίκτυο μέσα από διαδικασία μάθησης.
- Οι δυνάμεις σύνδεσης των νευρώνων, γνωστές ως συναπτικά (synaptic) βάρη, χρησιμοποιούνται για την αποθήκευση γνώσης.

Η διαδικασία για την εκμάθηση ονομάζεται «αλγόριθμος μάθησης». Ο καθορισμός του τρόπου μεταβολής των συναπτικών (synaptic) βαρών αποτελεί την παραδοσιακή μέθοδο για τον σχεδιασμό των νευρωνικών δικτύων.

1.6.2 Πλεονεκτήματα των Νευρωνικών Δικτύων

Η συμπαγής παράλληλα κατανεμημένη δομή και η ικανότητα μάθησης των νευρωνικών δικτύων κάνουν δυνατή την επίλυση πολύπλοκων προβλημάτων, τα οποία σπάνε σε μικρότερα έργα, που αναλαμβάνουν τα νευρωνικά δίκτυα ανάλογα με τις δυνατότητές τους.

Η χρήση των Νευρωνικών Δικτύων παρέχει τις ακόλουθες χρήσιμες ιδιότητες :

- 1) **Μη-γραμμικότητα.** Αυτή οφείλεται στο γεγονός ότι ένα Νευρωνικό Δίκτυο δομείται από τη σύνδεση νευρώνων, οι οποίοι είναι μη-γραμμικές συσκευές. Η μη-γραμμικότητα είναι πολύ σημαντική ιδιότητα, ιδιαίτερα αν ο φυσικός μηχανισμός για την παραγωγή των σημάτων εισόδου είναι μη-γραμμικός.
- 2) **Σχεδιασμός Εισόδου-Εξόδου.** Ένα συνηθισμένο παράδειγμα μάθησης που καλείται επιβλεπόμενη μάθηση εμπλέκει μεταβολή των συναπτικών βαρών του Νευρωνικού Δικτύου, εφαρμόζοντας ένα σύνολο παραδειγμάτων εκπαίδευσης. Κάθε παράδειγμα αποτελείται από ένα μοναδικό σήμα εισόδου και την επιθυμητή απόκριση. Η εκπαίδευση του δικτύου επαναλαμβάνεται για πολλά παραδείγματα, μέχρι το δίκτυο να φτάσει σε μια σταθερή κατάσταση, όπου πλέον δεν γίνονται αλλαγές στα βάρη. Έτσι το δίκτυο μαθαίνει από τα παραδείγματα, κατασκευάζοντας έναν σχεδιασμό εισόδου-εξόδου για το δεδομένο πρόβλημα.
- 3) **Προσαρμοστικότητα.** Τα Νευρωνικά Δίκτυα έχουν τη δυνατότητα να προσαρμόζουν τα βάρη τους στις αλλαγές του περιβάλλοντός τους. Μερικές φορές οι προσαρμογές οδηγούν σε μείωση της απόδοσης του συστήματος, γι' αυτό θα πρέπει να είναι επαρκείς, το γνωστό ως δίλημμα σταθερότητας-πλαστικότητας.
- 4) **Αποδεικτική Απόκριση.** Ένα Νευρωνικό Δίκτυο σχεδιάζεται για να παρέχει πληροφορίες όχι μόνο για το συγκεκριμένο υπόδειγμα που επιλέγεται, αλλά και για την εμπιστοσύνη στην απόφαση που παίρνεται. Αυτό βελτιώνει την απόδοση του συστήματος.
- 5) **Συναφής Πληροφορία.** Η γνώση αναπαριστάται από την πολύ δομημένη και ενεργή κατάσταση του Νευρωνικού Δικτύου.
- 6) **Αντοχή σε σφάλματα.** Ένα νευρωνικό δίκτυο υλοποιημένο σε υλικό (hardware) έχει τη σημαντική ιδιότητα να είναι «ανεκτικό σε σφάλματα», με την έννοια ότι η απόδοσή του δεν μειώνεται κάτω από αντίξοες λειτουργικές συνθήκες.
- 7) **Υλοποιησιμότητα σε VLSI.** Η συμπαγής παράλληλη φύση του Νευρωνικού Δικτύου κάνει δυνατή την υλοποίησή του σε VLSI τεχνολογία, έτσι ώστε τα νευρωνικά δίκτυα να μπορούν να χρησιμοποιηθούν σε εφαρμογές πραγματικού χρόνου.
- 8) **Ομοιομορφία Ανάλυσης και Σχεδιασμού.** Η έννοια είναι ότι ο ίδιος συμβολισμός χρησιμοποιείται σε όλα τα πεδία που περιέχουν εφαρμογή των νευρωνικών δικτύων. Αυτό το χαρακτηριστικό υποδηλώνεται με διαφορετικούς τρόπους: Οι νευρώνες παριστάνουν ένα συστατικό συνηθισμένο σε όλα τα νευρωνικά δίκτυα. Αυτή η ιδιότητα κάνει δυνατό τον διαμοιρασμό θεωριών και αλγορίθμων εκμάθησης σε διαφορετικές εφαρμογές των νευρωνικών δικτύων. Ρυθμιστικά (αναδρομικά) δίκτυα μπορούν να κατασκευαστούν μέσω μιας αμιγούς ολοκλήρωσης από modules.
- 9) **Αναλογία με Νευροβιολογία.** Ο σχεδιασμός νευρωνικών δικτύων γίνεται σε αναλογία με τον εγκέφαλο. Οι νευροβιολόγοι βλέπουν τα νευρωνικά δίκτυα σαν αντικείμενο έρευνας για την εξήγηση νευροβιολογικών φαινομένων. Ομοίως οι μηχανικοί αναζητούν στη νευροβιολογία νέες ιδέες για την επίλυση πολύπλοκων προβλημάτων.

1.6.3 Η θεωρία της Εξέλιξης των Ειδών

Η θεωρία της *Εξέλιξης των Ειδών (Evolution of Species)* που αναπτύχθηκε από τον Δαρβίνο στα μέσα του περασμένου αιώνα προκάλεσε μεγάλη αναστάτωση, αφού ερχόταν σε σύγκρουση με τις επικρατούσες θρησκευτικές αντιλήψεις περί προέλευσης της ζωής. Με την πάροδο ενός και πλέον αιώνα ο θόρυβος αυτός δεν έχει κοπάσει πλήρως, όμως η θεωρία έχει γίνει αποδεκτή από το σύνολο των επιστημόνων, γιατί κατόρθωσε να πείσει και να δώσει ικανοποιητικές απαντήσεις σε θεμελιώδη ερωτήματα. Σκοπός της είναι να δώσει μια εξήγηση για το φαινόμενο της ζωής, την προέλευσή της και τις βασικές λειτουργίες της. Τα κυριότερά της σημεία που σχετίζονται με τον τρόπο λειτουργίας των Γενετικών Αλγορίθμων και τον ερμηνεύουν είναι τα εξής:

Δεν υπάρχει αντικειμενική βάση διαχωρισμού των ζωντανών οργανισμών σε ανώτερους και κατώτερους. Σε κάθε βιολογικό είδος μερικά άτομα αφήνουν περισσότερους απογόνους σε σύγκριση με τα υπόλοιπα και έτσι τα κληροδοτούμενα χαρακτηριστικά των αναπαραγωγικά επιτυχημένων ατόμων γίνονται περισσότερα στην επόμενη γενιά. Οι δυσκολίες, τα εμπόδια και οι αντιξοότητες που παρουσιάζονται κατά τη διάρκεια της ζωής των οργανισμών είναι οι παράγοντες που καθορίζουν ποιοι από αυτούς θα κατορθώσουν να ζήσουν και να πολλαπλασιαστούν. Έτσι με την αλλαγή του περιβάλλοντος και των συνθηκών διαβίωσης αλλάζουν και τα χαρακτηριστικά τους, προσπαθώντας να προσαρμοστούν κάθε φορά, με στόχο την εξασφάλιση της επιβιώσής τους.

Αυτή η αλλαγή όμως που συμβαίνει στα χαρακτηριστικά των ατόμων είναι αλλαγή στα *χρωμοσώματά τους (chromosomes)*, τα πολύπλοκα οργανικά μόρια που κωδικοποιούν τη δομή και τα χαρακτηριστικά τους. Τα χρωμοσώματα αποτελούνται από μικρότερα μέρη, γνωστά ως *γονίδια (genes)*. Το σύνολο της γενετικής πληροφορίας που είναι κωδικοποιημένο στα γονίδια ονομάζεται *γονότυπος ή γενότυπος (genotype)*. Η δημιουργία ενός νέου οργανισμού περιλαμβάνει την αποκωδικοποίηση των χρωμοσωμάτων. Το σύνολο των «ορατών» χαρακτηριστικών του και της συμπεριφοράς του, που καθορίζονται από τις πληροφορίες των γονιδίων, συνιστούν τον *φαινότυπο (phenotype)*.

Κυρίαρχες λειτουργίες του φαινομένου της εξέλιξης είναι η *αναπαραγωγή (reproduction)* και η *μετάλλαξη (mutation)*. Κατά τη μετάλλαξη γίνεται με τυχαίο τρόπο η αλλαγή της δομής των χρωμοσωμάτων, συνήθως από λανθασμένη αντιγραφή βιολογικών μορίων ή από εξωγενείς παράγοντες (π.χ. ακτινοβολία), έχοντας ως άμεσο αποτέλεσμα αλλαγή σε κάποιο χαρακτηριστικό. Η μετάλλαξη μερικές φορές μπορεί να προκαλέσει βελτιώσεις και, χωρίς αμφιβολία, μερικά λάθη που έγιναν αποτέλεσαν σημαντικό παράγοντα για την προοδευτική εξέλιξη της ζωής.

Προϊόν της αναπαραγωγής είναι ένας νέος οργανισμός, τα χρωμοσώματα του οποίου αποτελούνται από γονίδια που προέρχονται τα μισά από τον πατέρα και τα μισά από τη μητέρα. Έτσι για κάθε χαρακτηριστικό το νέο άτομο έχει πάρει ένα γονίδιο από κάθε γονέα. Μερικές φορές τα δύο αυτά γονίδια συμφωνούν μεταξύ τους όσον αφορά την «τιμή» που θα δώσουν στο χαρακτηριστικό, π.χ. γαλάζιο χρώμα ματιών, ενώ άλλες φορές δεν συμφωνούν, π.χ. το ένα υποδεικνύει καστανό χρώμα ματιών και το άλλο γαλάζιο. Στη δεύτερη περίπτωση κυριαρχεί η «τιμή» ενός γονιδίου (π.χ. του καστανού) και αγνοείται η «τιμή» του άλλου, μολονότι το δεύτερο μπορεί να περάσει σε επόμενες γενιές. Το γονίδιο που τελικά καθορίζει το χαρακτηριστικό λέγεται *κυρίαρχο* ή *επικρατές (dominant)* και το άλλο *υπολειπόμενο (recessive)*. Γονίδια που διεκδικούν την ίδια θέση σε ένα χρωμόσωμα (δηλαδή που είναι υπεύθυνα για το ίδιο χαρακτηριστικό) λέγονται *αλληλόμορφα (alleles)*.

Όλος αυτός ο μηχανισμός της φυσικής επιλογής φάνηκε ιδιαίτερα ελκυστικός στον John Holland, πρωτοπόρο των Γενετικών Αλγορίθμων, στις αρχές της δεκαετίας του '70 (Goldberg, 1989). Ο Holland φαντάστηκε ότι κάποιες ιδέες και λειτουργίες που εφαρμόζει η φύση στα συστήματά της θα μπορούσαν να έχουν αποτελέσματα, αν ενσωματώνονταν σε αλγόριθμους για υπολογιστές, ώστε να προκύψουν αποδοτικές τεχνικές επίλυσης δύσκολων προβλημάτων. Αποτέλεσμα αυτής της εργασίας του Holland ήταν οι Γενετικοί Αλγόριθμοι, μια καινούρια εξελισσόμενη και πολλά υποσχόμενη τεχνική αναζήτησης και βελτιστοποίησης.

Η βασική ιδέα που κρύβεται πίσω από τους Γενετικούς Αλγόριθμους (ΓΑ) είναι η μίμηση των μηχανισμών της φύσης. Ας πάρουμε π.χ. τους λαγούς και πώς αναπαράγονται και εξελίσσονται από γενιά σε γενιά. Έστω ότι αρχίζουμε να παρατηρούμε έναν συγκεκριμένο πληθυσμό από λαγούς. Όπως είναι φυσικό, κάποιοι από αυτούς θα είναι πιο γρήγοροι και πιο εύστροφοι από τους άλλους. Αυτοί οι γρηγορότεροι και εξυπνότεροι λαγοί έχουν λιγότερες πιθανότητες να αποτελέσουν γέυμα κάποιας αλεπούς, άρα από τη στιγμή που καταφέρνουν να επιβιώσουν θα ασχοληθούν με την αναπαραγωγή του είδους τους. Φυσικά θα υπάρχει και ένας μικρός αριθμός αργών και λιγότερο εύστροφων λαγών που θα καταφέρουν να επιβιώσουν μόνο και μόνο επειδή στάθηκαν τυχεροί. Όλοι αυτοί οι λαγοί που έχουν καταφέρει να επιβιώσουν θα αρχίσουν την παραγωγή της επόμενης γενιάς τους, μιας γενιάς που θα συνδυάζει με διάφορους τρόπους όλα τα χαρακτηριστικά των

μελών της προηγούμενης. Έτσι μερικοί αργοί λαγοί θα αναμειχθούν με κάποιους γρήγορους, κάποιοι γρήγοροι με άλλους γρήγορους, κάποιοι εύστροφοι λαγοί με κάποιους μη εύστρους και ούτω καθεξής. Οι μικροί λαγοί της επόμενης γενιάς θα είναι κατά μέσο όρο γρηγορότεροι και έξυπνότεροι από τους προγόνους τους, αφού από την προηγούμενη γενιά επιβίωσαν περισσότεροι γρήγοροι και έξυπνοι λαγοί. Ευτυχώς για τη διατήρηση της φυσικής ισορροπίας, και οι αλεπούδες υφίστανται την ίδια διαδικασία αναπαραγωγής, διαφορετικά οι λαγοί θα γινόντουσαν υπερβολικά γρήγοροι και έξυπνοι για να μπορούν να τους πιάσουν.

1.6.4 Πλεονεκτήματα ΓΑ

Η χρήση των ΓΑ σε διάφορες εφαρμογές είναι ελκυστική για αρκετούς λόγους. Οι κυριότεροι ίσως είναι οι εξής:

- 1) **Μπορούν να λύσουν δύσκολα προβλήματα γρήγορα και αξιόπιστα.** Ένας από τους σημαντικούς λόγους χρήσης των ΓΑ είναι η μεγάλη τους αποδοτικότητα. Τόσο η θεωρία όσο και η πράξη έχουν δείξει ότι προβλήματα που έχουν πολλές, δύσκολα προσδιορισμένες λύσεις μπορούν να αντιμετωπιστούν καλύτερα από ΓΑ. Είναι δε αξιοσημείωτο ότι συναρτήσεις που παρουσιάζουν μεγάλες διακυμάνσεις και καθιστούν ανεπαρκείς άλλες μεθόδους στην εύρεση των ακρότατών τους, για τους ΓΑ αυτές οι διακυμάνσεις δεν αποτελούν σημεία δυσχέρειας.
- 2) **Μπορούν εύκολα να συνεργαστούν με τα υπάρχοντα μοντέλα και συστήματα.** Οι ΓΑ προσφέρουν το σημαντικό πλεονέκτημα της χρήσης τους με προσθετικό τρόπο στα μοντέλα που χρησιμοποιούνται σήμερα, μην απαιτώντας την επανασχεδιάσή τους. Μπορούν εύκολα να συνεργαστούν με τον υπάρχοντα κώδικα, χωρίς μεγάλο κόπο. Αυτό συμβαίνει διότι χρησιμοποιούν μόνο πληροφορίες της διαδικασίας ή συνάρτησης που πρόκειται να βελτιστοποιήσουν, δίχως να ενδιαφέρει άμεσα ο ρόλος της μέσα στο σύστημα ή η όλη δομή του συστήματος.
- 3) **Είναι εύκολα επεκτάσιμοι και εξελίξιμοι.** Όπως θα γίνει σαφές στα επόμενα κεφάλαια, οι ΓΑ δεν αντιστέκονται σε αλλαγές, επεκτάσεις και μετεξελίξεις, ανάλογα με την κρίση του σχεδιαστή. Σε πολλές εφαρμογές έχουν αναφερθεί λειτουργίες των ΓΑ που δεν είναι δανεισμένες από τη φύση ή που έχουν υποστεί σημαντικές αλλαγές, πάντα προς όφελος της απόδοσης. Παραλλαγές στο βασικό σχήμα δεν είναι απλά ανεκτές, αλλά σε ορισμένες περιπτώσεις επιβάλλονται.
- 4) **Μπορούν να συμμετέχουν σε υβριδικές μορφές με άλλες μεθόδους.** Αν και η ισχύς των ΓΑ είναι μεγάλη, σε μερικές ειδικές περιπτώσεις προβλημάτων, όπου άλλες μέθοδοι συμβαίνει να έχουν πολύ υψηλή αποδοτικότητα λόγω εξειδίκευσης, υπάρχει η δυνατότητα χρησιμοποίησης ενός υβριδικού σχήματος ΓΑ με άλλη μέθοδο. Αυτό είναι αποτέλεσμα της μεγάλης ευελιξίας των ΓΑ.
- 5) **Εφαρμόζονται σε πολύ περισσότερα πεδία από κάθε άλλη μέθοδο.** Το χαρακτηριστικό που τους εξασφαλίζει αυτό το πλεονέκτημα είναι η ελευθερία επιλογής των κριτηρίων που καθορίζουν την επιλογή μέσα στο τεχνικό περιβάλλον. Έτσι ΓΑ μπορούν να χρησιμοποιηθούν στην οικονομία, στον σχεδιασμό μηχανών, στην επίλυση μαθηματικών εξισώσεων, στην εκπαίδευση Νευρωνικών Δικτύων και σε πολλούς άλλους τομείς.
- 6) **Δεν απαιτούν περιορισμούς στις συναρτήσεις που επεξεργάζονται.** Ο κύριος λόγος που καθιστά τις παραδοσιακές μεθόδους δύσκαμπτες και ακατάλληλες για πολλά προβλήματα είναι η απαίτησή τους για ύπαρξη περιορισμών, όπως η ύπαρξη παραγώγων, η συνέχεια, οι μη «θορυβώδεις» συναρτήσεις κλπ. Τέτοιου είδους ιδιότητες είναι αδιάφορες για τους ΓΑ, στοιχείο που τους κάνει κατάλληλους για μεγάλο φάσμα προβλημάτων.
- 7) **Δεν ενδιαφέρει η σημασία της υπό εξέταση πληροφορίας.** Η μόνη «επικοινωνία» του ΓΑ με το περιβάλλον του είναι η αντικειμενική συνάρτηση. Αυτό εγγυάται την επιτυχία του ανεξάρτητα από τη σημασία του προβλήματος. Βέβαια δεν σημαίνει ότι δεν υπάρχουν άλματα προβλήματα για τους ΓΑ. Όπου όμως δεν τα καταφέρνουν, η αιτία είναι η φύση του χώρου που ερευνούν και όχι το πληροφοριακό περιεχόμενο του προβλήματος.
- 8) **Έχουν από τη φύση τους το στοιχείο του παραλληλισμού.** Οι ΓΑ σε κάθε τους βήμα επεξεργάζονται μεγάλες ποσότητες πληροφορίας, αφού κάθε άτομο θεωρείται αντιπρόσωπος πολλών άλλων. Έχει υπολογιστεί ότι η αναλογία αυτή είναι της τάξεως $O(n^3)$, δηλαδή 10 άτομα αντιπροσωπεύουν περίπου 1000. Επίσης, είναι η μοναδική μέθοδος που εξετάζει παράλληλα πολλά σημεία (όσα και τα άτομα του

πληθυσμού). Είναι λοιπόν προφανές ότι μπορούν να καλύψουν με αποδοτικό ψάξιμο μεγάλους χώρους σε μικρούς χρόνους.

9) **Είναι η μόνη μέθοδος που κάνει ταυτόχρονα εξερεύνηση του χώρου αναζήτησης και εκμετάλλευση της ήδη επεξεργασμένης πληροφορίας.** Ο συνδυασμός αυτός σπάνια συναντάται σε οποιαδήποτε άλλη μέθοδο. Με το τυχαίο ψάξιμο γίνεται καλή εξερεύνηση του χώρου, αλλά δεν γίνεται εκμετάλλευση της πληροφορίας. Αντίθετα, με την αναζήτηση με μικρά άλματα στη συνάρτηση (hillclimbing) γίνεται καλή εκμετάλλευση της πληροφορίας, αλλά όχι καλή εξερεύνηση. Συνήθως τα δύο αυτά χαρακτηριστικά είναι ανταγωνιστικά και το επιθυμητό είναι να συνυπάρχουν και τα δύο προς όφελος της όλης διαδικασίας. Οι ΓΑ επιτυγχάνουν τον βέλτιστο συνδυασμό εξερεύνησης και εκμετάλλευσης, κάτι που τους κάνει ιδιαίτερα αποδοτικούς και ελκυστικούς.

10) **Επιδέχονται παράλληλη υλοποίηση.** Οι ΓΑ μπορούν να εκμεταλλευτούν τα πλεονεκτήματα των παράλληλων μηχανών, αφού λόγω της φύσης τους εύκολα μπορούν να δεχτούν παράλληλη υλοποίηση. Το χαρακτηριστικό αυτό αυξάνει ακόμη περισσότερο την απόδοσή τους, ενώ σπάνια συναντάται σε ανταγωνιστικές μεθόδους.

Άσκηση 1.6/7: Να αναφέρετε τις ομοιότητες μεταξύ νευρωνικών δικτύων και ανθρώπινου εγκέφαλου.

Άσκηση 1.6/8: Ποια είναι τα πλεονεκτήματα των ΝΔ;

Άσκηση 1.6/9: Πού οφείλεται η μη-γραμμικότητα των νευρωνικών δικτύων;

1. στην παράλληλη κατανομημένη δομή τους,
2. στη δυνατότητα να προσαρμόζουν τα βάρη τους στις αλλαγές του περιβάλλοντός τους,
3. στο γεγονός ότι ένα ΝΔ δομείται από νευρώνες οι οποίοι είναι μη-γραμμικές συσκευές,
4. στα συναπτικά βάρη.

Να επιλέξετε τη σωστή απάντηση.

Άσκηση 1.6/10: Η εφαρμογή των ΓΑ σε μεγάλο φάσμα εφαρμογών οφείλεται:

1. στο ότι η μόνη «επικοινωνία» τους με το περιβάλλον είναι η αντικειμενική συνάρτηση,
2. στη χρησιμοποίηση μόνο πληροφοριών της συνάρτησης που πρόκειται να βελτιστοποιήσουν, δίχως να ενδιαφέρει άμεσα ο ρόλος της μέσα στο σύστημα ή η όλη δομή του συστήματος,
3. στο ότι είναι εύκολα επεκτάσιμοι και εξελίξιμοι και μπορούν να συμμετέχουν σε υβριδικές μορφές με άλλες μεθόδους,
4. στην ελευθερία επιλογής των κριτηρίων που καθορίζουν την επιλογή μέσα στο τεχνητό περιβάλλον και στη μη απαίτηση για ύπαρξη περιορισμών,
5. σε όλα τα παραπάνω.

Να επιλέξετε τη σωστή απάντηση.

Άσκηση 1.6/11: Ποια είναι τα κύρια στοιχεία της Θεωρίας της Εξέλιξης των Ειδών;

Άσκηση 1.6/12: Το σύνολο των ορατών χαρακτηριστικών και της συμπεριφοράς ενός ατόμου που καθορίζονται από τις πληροφορίες των γονιδίων αποτελούν τον γονότυπό του;

Άσκηση 1.6/13: Αλληλόμορφα λέγονται τα γονίδια που προέρχονται από τον ίδιο γονέα;

Δραστηριότητα 1/2: Σε ένα είδος ζωντανών οργανισμών ποιος θα ζήσει και ποιος θα πεθάνει; Να δώσετε την απάντησή σας σε μισή σελίδα περίπου.

1.7 Σύνοψη κεφαλαίου

Σε αυτό το κεφάλαιο, αφού έγινε πρώτα μία σύντομη αναφορά στις τεχνολογίες της Τεχνητής Νοημοσύνης (TN), ακολούθησε η εισαγωγή στα Νευρωνικά Δίκτυα (ΝΔ) και στους Γενετικούς Αλγορίθμους (ΓΑ). Στόχος ήταν η σύνδεση των ΝΔ όσο και των ΓΑ με την TN της οποίας αποτελούν υποπεριοχές.

Τα ΝΔ προσπαθούν να μιμηθούν τη λειτουργία του ανθρώπινου εγκεφάλου, όσον αφορά τον τρόπο μάθησης. Η παρουσίαση μερικών βασικών στοιχείων της λειτουργίας του εγκεφάλου ήταν πολύ χρήσιμη για την κατανόηση των βασικών αρχών λειτουργίας των ΝΔ. Για να γίνει κατανοητή η λειτουργία ενός Γενετικού Αλγορίθμου ήταν σκόπιμο να δοθούν ορισμένα στοιχεία από τη Θεωρία της Βιολογικής Γενετικής Εξέλιξης, από την οποία δανείζονται στοιχεία οι Γενετικοί Αλγόριθμοι, ενώ στη συνέχεια είδαμε πώς στοιχεία αυτής της θεωρίας μπορούν να χρησιμοποιηθούν για να δομηθεί ένας αλγόριθμος αναζήτησης και βελτιστοποίησης. Επίσης, παρουσιάστηκαν τα βασικά πλεονεκτήματα αυτών των δύο κατηγοριών αλγορίθμων.

Μελετώντας το συγκεκριμένο κεφάλαιο, ο αναγνώστης πρέπει να έχει εξοικειωθεί με τις έννοιες της Τεχνητής Νοημοσύνης, της Τεχνητής Μάθησης και της Αναζήτησης και Βελτιστοποίησης. Κυρίως όμως τον έχει εισαγάγει σε θέματα της θεωρίας της μάθησης και της γενετικής αναζήτησης, δηλαδή στο πώς ιδέες που προέρχονται από τη μάθηση και την εξέλιξη των βιολογικών ειδών μπορούν να χρησιμοποιηθούν σε πραγματικά προβλήματα μάθησης και αναζήτησης του βέλτιστου και να οδηγήσουν στη λύση τους γρήγορα και αποδοτικά. Επίσης, αναφέρθηκαν οι περιοχές στις οποίες βρίσκουν εφαρμογή αυτοί οι αλγόριθμοι για την επίλυση δύσκολων πρακτικών προβλημάτων. Τέλος, έγινε μία σύντομη παρουσίαση των βιολογικών συστημάτων από τα οποία οι ερευνητές εμπνεύστηκαν τόσο τα ΝΔ, όσο και τους ΓΑ.

Η βιβλιογραφία που ακολουθεί χρησιμοποιείται συμπληρωματικά. Η πρώτη αναφορά περιέχει μια πολύ καλά θεμελιωμένη παρουσίαση της θεωρίας της μάθησης και κυρίως των Τεχνητών Νευρωνικών Δικτύων, τόσο από την πλευρά των αλγορίθμων εκπαίδευσης όσο και από τη σκοπιά της θεωρητικής τους μελέτης. Επίσης, περιέχει μεγάλο αριθμό παραδειγμάτων και πλήθος πρακτικών εφαρμογών. Από το βιβλίο αυτό έχει χρησιμοποιηθεί σημαντικό υλικό και θα αναφέρεται συχνά στα επόμενα κεφάλαια. Στη δεύτερη αναφορά γίνεται μια πιο εφαρμοσμένη παρουσίαση των ΝΔ, με έμφαση κυρίως σε εφαρμογές. Η τρίτη αναφορά αποτελεί το πρώτο βασικό βοήθημα που εμφανίστηκε στη διεθνή βιβλιογραφία σχετικά με τους Γενετικούς Αλγορίθμους. Εκτός από την εκτεταμένη εισαγωγή στους ΓΑ δίνεται έμφαση στη χρήση τους σε προβλήματα αναζήτησης και βελτιστοποίησης. Επίσης, εκεί γίνεται για πρώτη φορά μια συστηματική προσπάθεια για τη θεωρητική τους θεμελίωση. Στο τέταρτο βιβλίο, μετά από μία σύντομη εισαγωγή στους ΓΑ με πολλά παραδείγματα, δίνεται έμφαση στη γενίκευσή τους με χρήση άλλων τελεστών εκτός από αυτούς που ήδη αναφέραμε· δηλαδή επικεντρώνεται στα Εξελικτικά Προγράμματα, τα οποία θα μας απασχολήσουν στο τελευταίο κεφάλαιο. Στο βιβλίο αυτό θα γίνεται συχνά αναφορά στα επόμενα κεφάλαια. Η τελευταία αναφορά είναι προσανατολισμένη σε εφαρμοσμένα προβλήματα που επιλύονται με ΓΑ και ενδιαφέρουν κυρίως τους μηχανικούς.

Βιβλιογραφία

- Bengio, Y. (2009). Learning Deep Architectures for AI. *Foundations and Trends in Machine Learning*, 2(1). 1–127.
- Davis, L. (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley Publishing Company, Inc.
- Hagan, M.T., Demuth, H.B., & Beal, M. (1996). *Neural Network Design*. PWS Publishing Company.
- Haykin, S. (1999). *Neural Networks, A Comprehensive Foundation* (2η έκδ.). Prentice Hall International.
- Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs* (2η έκδ.). Springer - Verlag.

Απαντήσεις ασκήσεων

Άσκηση 1.2/1: Ποιες είναι οι κυριότερες εφαρμογές της TN;

Απάντηση: Σωστά, οι πιο γνωστές είναι αυτές που αναφέρονται στη σ. 8.

Άσκηση 1.2/2: Ένα νευρωνικό δίκτυο μπορεί να θεωρηθεί ως έμπειρο σύστημα;

Απάντηση: Ένα ΝΔ εκπαιδεύεται να μάθει ένα σύνολο από ζευγάρια εισόδου-εξόδου, δηλαδή AN είσοδος $TOTE$ έξοδος. Με αυτήν την έννοια μπορεί να θεωρηθεί ως ΕΣ.

Άσκηση 1.2/3: Στη σ. 8 είδαμε μερικές εφαρμογές της TN. Σε ποιες από αυτές μπορούν να εφαρμοστούν τα ΝΔ και σε ποιες οι ΓΑ;

Απάντηση: 1. ΓΑ 2. ΝΔ ή/και ΓΑ 3. ΝΔ 4. ΝΔ 5. ΝΔ ή/και ΓΑ 6. ΝΔ

Άσκηση 1.3/4: Μάθηση είναι η διαδικασία:

- κατά την οποία το ΝΔ «απαντά» με έναν καινούριο τρόπο στο περιβάλλον, λόγω των αλλαγών που συνέβησαν στην εσωτερική του δομή,
- κατά την οποία προσαρμόζονται οι ελεύθερες παράμετροι του δικτύου,
- κατά την οποία το ΝΔ υφίσταται αλλαγές ως συνέπεια μιας εξωτερικής διέγερσης.

Να επιλέξετε τη σωστή απάντηση.

Απάντηση: Σύμφωνα με τον ορισμό που δόθηκε στη σ. 12, σωστή απάντηση είναι η Β.

Άσκηση 1.3/5: Αν η ρύθμιση που υπολογίστηκε για το βάρος $w_{kj}(n)$ είναι αρνητική, τι επίδραση θα έχει αυτό στη νέα τιμή του βάρους;

Απάντηση: Από τη σχέση (1) προκύπτει αμέσως ότι η νέα τιμή του βάρους είναι μικρότερη σε σχέση με την προηγούμενη.

Άσκηση 1.4/6: Ποιο είναι το βασικό μειονέκτημα των κλασικών μεθόδων αναζήτησης;

- έχουν μεγάλη πολυπλοκότητα,
- κάνουν τυχαίο ψάξιμο,
- κάνουν αναζήτηση σε ένα μόνο σημείο σε κάθε βήμα,
- δεν αξιοποιούν την πληροφορία που παρέχει το περιβάλλον,
- είναι τοπικής εμβέλειας.

Να επιλέξετε τη(ις) σωστή(ές) απάντηση(εις).

Απάντηση: Τα βασικά μειονεκτήματα των κλασικών μεθόδων είναι τα 3 και 4.

Άσκηση 1.6/7: Να αναφέρετε τις ομοιότητες μεταξύ νευρωνικών δικτύων και ανθρώπινου εγκεφάλου.

Απάντηση: Η απάντηση βρίσκεται στο τέλος της υποενότητας 1.6.1.

Άσκηση 1.6/8: Ποια είναι τα πλεονεκτήματα των ΝΔ;

Απάντηση: Η απάντηση βρίσκεται στην υποενότητα 1.6.4.

Άσκηση 1.6/9: Πού οφείλεται η μη-γραμμικότητα των νευρωνικών δικτύων;

- στην παράλληλη κατανομημένη δομή τους,
- στη δυνατότητα να προσαρμόζουν τα βάρη τους στις αλλαγές του περιβάλλοντός τους,
- στο γεγονός ότι ένα ΝΔ δομείται από νευρώνες οι οποίοι είναι μη-γραμμικές συσκευές,

4. στα συναπτικά βάρη.

Να επιλέξετε τη σωστή απάντηση.

Απάντηση: Η σωστή απάντηση είναι η 3.

Άσκηση 1.6/10: Η εφαρμογή των ΓΑ σε μεγάλο φάσμα εφαρμογών οφείλεται:

1. στο ότι η μόνη «επικοινωνία» τους με το περιβάλλον είναι η αντικειμενική συνάρτηση,
2. στη χρησιμοποίηση μόνο πληροφοριών της συνάρτησης που πρόκειται να βελτιστοποιήσουν, δίχως να ενδιαφέρει άμεσα ο ρόλος της μέσα στο σύστημα ή η όλη δομή του συστήματος,
3. στο ότι είναι εύκολα επεκτάσιμοι και εξελίξιμοι και μπορούν να συμμετέχουν σε υβριδικές μορφές με άλλες μεθόδους,
4. στην ελευθερία επιλογής των κριτηρίων που καθορίζουν την επιλογή μέσα στο τεχνητό περιβάλλον και στη μη απαίτηση για ύπαρξη περιορισμών,
5. σε όλα τα παραπάνω.

Να επιλέξετε τη σωστή απάντηση.

Απάντηση: Η σωστή απάντηση είναι η 5.

Άσκηση 1.6/11: Ποια είναι τα κύρια στοιχεία της Θεωρίας της Εξέλιξης των Ειδών;

Απάντηση: Τα βασικά στοιχεία της Θεωρίας της Εξέλιξης των Ειδών είναι τα εξής:

- Δεν υπάρχει αντικειμενική βάση διαχωρισμού των ζωντανών οργανισμών σε ανώτερους και κατώτερους.
- Η αλλαγή που συμβαίνει στα χαρακτηριστικά των ατόμων είναι αλλαγή στα *χρωμοσώματά* τους (*chromosomes*), που είναι πολύπλοκα οργανικά μόρια τα οποία κωδικοποιούν τη δομή και τα χαρακτηριστικά τους.
- Κυρίαρχες λειτουργίες του φαινομένου της εξέλιξης είναι η *αναπαραγωγή* (*reproduction*) και η *μετάλλαξη* (*mutation*).
- Προϊόν της αναπαραγωγής είναι ένας νέος οργανισμός, τα χρωμοσώματα του οποίου αποτελούνται από γονίδια που προέρχονται τα μισά από τον πατέρα και τα μισά από τη μητέρα.

Άσκηση 1.6/12: Το σύνολο των ορατών χαρακτηριστικών και της συμπεριφοράς ενός ατόμου που καθορίζονται από τις πληροφορίες των γονιδίων αποτελούν τον γονότυπό του;

Απάντηση: Όχι, αυτό που είναι ορατό στο άτομο είναι ο φαινότυπος, που αντιπροσωπεύει το σύνολο των «ορατών» χαρακτηριστικών και της συμπεριφοράς του ατόμου. Ο γονότυπος είναι το σύνολο όλης της γενετικής πληροφορίας που είναι κωδικοποιημένη στα γονίδια του ατόμου.

Άσκηση 1.6/13: Αλληλόμορφα λέγονται τα γονίδια που προέρχονται από τον ίδιο γονέα;

Απάντηση: Όχι, αλληλόμορφα λέγονται τα γονίδια που διεκδικούν την ίδια θέση σε ένα χρωμόσωμα, δηλαδή αυτά που είναι υπεύθυνα για το ίδιο χαρακτηριστικό, π.χ. το χρώμα των ματιών.

Κεφάλαιο 2

Εισαγωγή στα Τεχνητά Νευρωνικά Δίκτυα

Σύνοψη

Στο πρώτο κεφάλαιο έγινε αρχικά μία σύντομη αναφορά στις τεχνολογίες της Τεχνητής Νοημοσύνης (TN) και ακολούθησε σύντομη εισαγωγή στα Νευρωνικά Δίκτυα (ΝΔ) και στους Γενετικούς Αλγορίθμους (ΓΑ). Στόχος αυτού του κεφαλαίου είναι η σύνδεση των ΝΔ και των ΓΑ με την TN από την οποία προήλθαν.

Εδώ λοιπόν θα ασχοληθούμε με την εξέλιξη των ΝΔ και τις γνωστές αρχιτεκτονικές που χρησιμοποιούνται. Πρώτα θα γίνει μια σύντομη ιστορική αναδρομή και θα ακολουθήσει η παρουσίαση του μαθηματικού μοντέλου του τεχνητού νευρώνα. Στη συνέχεια θα παρουσιαστούν οι διάφορες κατηγορίες νευρωνικών δικτύων σε σχέση με την αρχιτεκτονική τους και τη λειτουργία που εκτελούν.

Σκοπός αυτού του κεφαλαίου είναι να παράσχει όλη την απαραίτητη πληροφορία, έτσι ώστε ο αναγνώστης να είναι έτοιμος για τη μελέτη των αλγορίθμων εκπαίδευσης των Νευρωνικών Δικτύων.

Προσδοκώμενα αποτελέσματα:

Όταν θα έχετε τελειώσει τη μελέτη αυτού του κεφαλαίου, θα μπορείτε να:

- κάνετε μια σύντομη ιστορική αναδρομή της εξέλιξης των ΝΔ,
- δώσετε το μοντέλο του τεχνητού νευρώνα,
- παρουσιάσετε τις βασικές αρχιτεκτονικές των ΝΔ.

Λέξεις-Κλειδιά:

- Μοντέλο τεχνητού νευρώνα
- Συνάρτηση ενεργοποίησης
- Αρχιτεκτονική ΝΔ
- Δίκτυα ενός επιπέδου
- Δίκτυα πολλών επιπέδων
- Αναδρομικά δίκτυα
- Δίκτυα Hopfield

Προαπαιτούμενη γνώση

Κεφάλαιο 1.

2.1 Εισαγωγικές Παρατηρήσεις

Όπως έχουμε ήδη αναφέρει, τα Νευρωνικά Δίκτυα (ΝΔ) αποτελούν μια προσπάθεια προσέγγισης της λειτουργίας του ανθρώπινου εγκεφάλου από μια μηχανή. Έχουν την ικανότητα να εκτελούν υπολογισμούς με μαζικό παράλληλο τρόπο. Η αρχιτεκτονική τους βασίζεται στην αρχιτεκτονική των Βιολογικών Νευρωνικών Δικτύων. Αυτές οι ιδέες, αν και ήταν γνωστές από την αρχή του αιώνα, άρχισαν να υλοποιούνται στα μέσα της δεκαετίας του 1940, όταν πρωτοπαρουσιάστηκε το μαθηματικό μοντέλο του νευρώνα το οποίο εξομοίωνε τη λειτουργία του βιολογικού νευρώνα. Στη συνέχεια με την εμφάνιση των πρώτων υπολογιστών άρχισε η προσπάθεια υλοποίησής τους. Μεγάλη ώθηση της εφαρμογής τους έδωσε η εμφάνιση των ψηφιακών υπολογιστών. Τότε άρχισε η ευρεία υλοποίησή τους, τόσο με λογισμικό όσο και με υλικό. Έτσι τα απλά δίκτυα ενός επιπέδου εξελίχθηκαν σε δίκτυα πολλών επιπέδων μέχρι τα δίκτυα βαθιάς μάθησης. Παράλληλα παρουσιάστηκε στη διεθνή βιβλιογραφία μια μεγάλη ποικιλία αλγορίθμων εκπαίδευσης και εφαρμογών τους στην επίλυση πρακτικών προβλημάτων.

2.2 Ιστορική αναδρομή

Η μοντέρνα περίοδος των νευρωνικών δικτύων λέγεται (Haykin, 1999) ότι άρχισε με την *πρωτοποριακή δουλειά των McCulloch και Pitts (1943)*. Ο πρώτος ήταν ψυχίατρος και ο δεύτερος μεγαλοφυής μαθηματικός. Σύμφωνα με τον Rall (1990) η κλασική εργασία των McCulloch και Pitts έγινε μέσα σε μια κοινότητα που ασχολούνταν με τους νευρώνες, στο πανεπιστήμιο του Σικάγο, για πάνω από πέντε χρόνια. Αυτή η εργασία περιέγραφε τον *λογικό λογισμό των νευρωνικών δικτύων*. Είναι αξιοσημείωτο το ότι ο von Neumann χρησιμοποίησε ιδεατά στοιχεία καθυστέρησης —τα οποία είχαν υπολογιστεί από τα ιδεατά στοιχεία νευρώνων των McCulloch και Pitts— για την κατασκευή του EDVAC (Electronic Discrete Variable Automatic Computer), ο οποίος κατόρθωσε στον ENIAC, τον πρώτο γενικού σκοπού υπολογιστή.

Η επόμενη μεγάλη ανάπτυξη πάνω στα νευρωνικά δίκτυα ήρθε το 1949 με την έκδοση του βιβλίου του Hebb *“The Organization of Behavior”*, στο οποίο έγινε για πρώτη φορά μια ιδιαίτερη δήλωση ενός *φυσιολογικού κανόνα μάθησης για συναπτικές τροποποιήσεις*. Πιο συγκεκριμένα ο Hebb πρότεινε ότι η *συνδετικότητα του εγκεφάλου συνεχώς αλλάζει καθώς ο οργανισμός μαθαίνει διάφορες εργασίες και ότι οι νευρωνικοί συγκεντρωτές δημιουργούνται από τέτοιες αλλαγές*. Επίσης, πρότεινε το διάσημο *αίτημα μάθησης* σύμφωνα με το οποίο η *αποτελεσματικότητα μιας σύναψης μεταβλητής ανάμεσα σε δύο νευρώνες αυξάνεται από την επαναλαμβανόμενη ενεργοποίηση του ενός νευρώνα από τον άλλο κατά μήκος της σύναψης*.

Η αναφορά των Rochester, Holland, Habit και Duda (1956) είναι ίσως η πρώτη προσπάθεια για *χρήση υπολογιστή που χρησιμοποιεί την εξομοίωση*, για να ελεγχθεί μια καλά σχηματισμένη νευρωνική θεωρία βασισμένη στον κανόνα μάθησης του Hebb. Η εξομοίωση έδειξε ότι χρειαζόταν να προστεθεί παρεμπόδιση ώστε η θεωρία να δουλέψει πραγματικά.

Το 1952 εκδόθηκε το βιβλίο του Ashby *“Design for a brain: The Origin of Adaptive Behavior”*, το οποίο ασχολήθηκε με τη *βασική έννοια ότι η προσαρμοζόμενη συμπεριφορά δεν είναι έμφυτη αλλά μαθαίνεται*. Το 1954 ο Minsky έγραψε τη διδακτορική του διατριβή *“Theory of Neural-Analog Reinforcement Systems and Its Application to the Brain-Model Problem”* και το 1961 μια εργασία με τίτλο *“Steps Toward Artificial Intelligence”*. Επίσης, το 1954 η ιδέα των *μη γραμμικών προσαρμοζόμενων φίλτρων* προτάθηκε από τον Gabor (πρωτοπόρο της θεωρίας επικοινωνιών και εφευρέτη του *ολογραφήματος*), ο οποίος υλοποίησε μια τέτοια μηχανή στην οποία η μάθηση επιτυγχανόταν με τροφοδότηση δειγμάτων στοχαστικών διαδικασιών στη μηχανή, μαζί με τη συνάρτηση-στόχο την οποία ήταν αναμενόμενο να παράγει η μηχανή.

Ένα θέμα ιδιαίτερου ενδιαφέροντος για τα νευρωνικά δίκτυα είναι αυτό της *σχεδίασης ενός αξιόπιστου δικτύου με νευρώνες που μπορούν να θεωρηθούν ως μη αξιόπιστα στοιχεία*. Αυτό το σημαντικό πρόβλημα λύθηκε από τον von Neumann (1956) χρησιμοποιώντας την ιδέα του πλεονασμού. Δεκαπέντε χρόνια μετά την έκδοση της εργασίας των McCulloch και Pitts, μια νέα προσέγγιση πάνω στο πρόβλημα της αναγνώρισης προτύπων έγινε από τον Rosenblatt (1958) στην εργασία του πάνω στον αισθητήρα (perceptron). Το ιδιαίτερο επίτευγμά του ήταν το *αποκαλούμενο θεώρημα σύγκλισης του αισθητήρα* (perceptron convergence theorem). Το 1960 οι Widrow και Hoff πρότειναν τον *αλγόριθμο ελάχιστου μέσου τετραγώνου* (least mean-square-LMS) και τον χρησιμοποίησαν για να σχηματίσουν το *Adaline* (*adaptive linear element*). Η διαφορά ανάμεσα στον αισθητήρα και στο Adaline βρίσκεται στον τρόπο μάθησης. Ένα από τα από τα πρόσφατα εκπαιδευσιμα νευρωνικά δίκτυα με πολλαπλά στοιχεία είναι η δομή Madaline (Widrow). Κατά τη διάρκεια της κλασικής περιόδου του perceptron (δεκαετία '60) πιστευόταν ότι τα νευρωνικά δίκτυα μπορούσαν να κάνουν τα πάντα. Αλλά τότε εκδόθηκε το βιβλίο των Minsky και Papert που με τη βοήθεια των μαθηματικών απέδειξε ότι υπάρχουν όρια πάνω στο τι μπορεί να υπολογιστεί από τον αισθητήρα. Ένα σημαντικό πρόβλημα πάνω στη σχεδίαση ενός πολυεπίπεδου αισθητήρα είναι το *πρόβλημα της ανάθεσης εμπιστοσύνης* (credit assignment problem), το οποίο βρήκε τη λύση του μόλις τη δεκαετία του 1980.

Κατά τη δεκαετία του '70 λόγω των προβλημάτων εγκαταλείφθηκε το ενδιαφέρον πάνω στα νευρωνικά δίκτυα. Μια σημαντική ενέργεια τη δεκαετία αυτή ήταν οι *χάρτες αυτοοργάνωσης με ανταγωνιστική μάθηση*.

Το 1980 έγιναν πολλές εργασίες πάνω στη θεωρία αλλά και στον σχεδιασμό των νευρωνικών δικτύων. Ο Grossberg (1980) ανέπτυξε μια καινούρια αρχή αυτοοργάνωσης που συνδυάζει φίλτράρισμα από «κάτω προς τα πάνω» και αντίθετη αύξηση σε μικρή μνήμη με από «πάνω προς τα κάτω» ταίριασμα προτύπων και σταθεροποίηση του κώδικα μάθησης. Δεδομένης μιας τέτοιας ικανότητα και αν το πρότυπο εισόδου ταιριάζει με την ανάδραση μάθησης, λαμβάνει χώρα μια δυναμική κατάσταση που καλείται *προσαρμοστικός συλλογισμός* (adaptive resonance). Αυτό το φαινόμενο δίνει τη βάση για μια νέα κατηγορία νευρωνικών δικτύων γνωστών ως *adaptive resonance theory (ART)*.

Το 1982 ο Hopfield χρησιμοποίησε την ιδέα μιας συνάρτησης ενέργειας για να δημιουργήσει έναν νέο τρόπο κατανόησης του υπολογισμού που γίνεται από τα δίκτυα με συμμετρικές συναπτικές συνδέσεις.

Επιπλέον, καθιέρωσε τον *ισομορφισμό* ανάμεσα σε τέτοια *περιοδικά δίκτυα* και σε ένα *Ising μοντέλο* που χρησιμοποιείται στη στατιστική. Αυτή η αναλογία άνοιξε τον δρόμο για έναν κατακλυσμό θεωριών για τα νευρωνικά δίκτυα. Η συγκεκριμένη τάξη νευρωνικών δικτύων με ανατροφοδότηση έτυχε ιδιαίτερης προσοχής κατά τη δεκαετία του '80 και με τον χρόνο έγιναν γνωστά ως δίκτυα Hopfield.

Το 1983 οι Cohen και Grossberg έκαναν μια νέα αρχή για σχεδίαση μιας *διευθυνσιοδοτούμενης μνήμης* (content-addressable memory) που περιλαμβάνει την έκδοση συνεχούς χρόνου του δικτύου Hopfield σαν μια ιδιαίτερη περίπτωση. Μια ακόμα σημαντική ανάπτυξη το 1982 ήταν η έκδοση της εργασίας του Kohonen πάνω στους χάρτες αυτοοργάνωσης, χρησιμοποιώντας *μιας ή δύο διαστάσεων δικτυωτές δομές*. Το 1983 οι Kirkpatrick, Gallat και Vecchi περιέγραψαν μια νέα διαδικασία που λέγεται *εξομοιωμένη απόπτηση* για λύση προβλημάτων συνδυαστικής βελτιστοποίησης. Η εξομοιωμένη απόπτηση χρησιμοποιείται στη στατιστική θερμοδυναμική και βασίζεται σε μια απλή τεχνική. Την ίδια χρονιά εκδόθηκε μια εργασία από τους Burto, Sutton και Anderson πάνω στην *ενισχυτική μάθηση*, η οποία δημιούργησε μεγάλο ενδιαφέρον πάνω σε αυτήν και την εφαρμογή της. Το 1984 ο Braitenberg εξέδωσε το βιβλίο "*Vehicles: Experiments in Synthetic Psychology*", το οποίο περιγράφει διάφορες μηχανές με απλή εσωτερική αρχιτεκτονική και το οποίο ενσωματώνει μερικές σημαντικές αρχές της αυτοοργανούμενης απόδοσης. Το 1986 η ανάπτυξη του αλγορίθμου για πίσω διάδοση (back-propagation algorithm) παρουσιάστηκε για πρώτη φορά από τον Rumelhart. Αυτός ο αλγόριθμος έγινε πολύ δημοφιλής και έδωσε νέα ώθηση στις εφαρμογές των νευρωνικών δικτύων.

Το 1988 ο Linsker περιέγραψε μια νέα αρχή για την αυτοοργάνωση σε ένα δίκτυο αισθητηρίων (perceptrons). Η αρχή αυτή σχεδιάστηκε ώστε να διατηρεί τη μέγιστη πληροφορία σχετικά με τα πρότυπα ενεργειών που υπόκεινται σε περιορισμούς, όπως συναπτικές συνδέσεις και δυναμικές περιοχές σύναψης. Ο Linsker χρησιμοποίησε αφηρημένες έννοιες πάνω στη θεωρία πληροφοριών ώστε να σχηματίσει την αρχή της διατήρησης μέγιστης πληροφορίας.

Επίσης, το 1988 οι Broomhead και Lowe περιέγραψαν μια διαδικασία για τον σχεδιασμό της «προς τα εμπρός τροφοδότησης» (feedforward) δικτύων χρησιμοποιώντας *συναρτήσεις ακτινικής βάσης* (*Radial Basis Functions - RBF*), που αποτελούν μια παραλλαγή των πολυεπίπεδων αισθητηρίων. Το 1989 εκδόθηκε το βιβλίο του "Mead *VLSI and Neural Systems*, το οποίο δίνει μια ασυνήθιστη μείξη περιεχομένων από τη νευροβιολογία και την τεχνολογία VLSI. Ίσως περισσότερο από κάθε άλλη έκδοση, η εργασία του Hopfield (1982) και το δίτομο βιβλίο των Rummelhart και McLelland (1986) ήταν οι σημαίνουσες εκδόσεις, υπεύθυνες για την αναζωογόνηση του ενδιαφέροντος για τα νευρωνικά δίκτυα στη δεκαετία του '80.

Τα νευρωνικά δίκτυα έχουν σίγουρα διανύσει πολύ δρόμο από την εποχή των McCulloch και Pitts. Πραγματικά έχουν εγκαθιδρυθεί ως διεπιστημονικό αντικείμενο με βαθιές ρίζες στην επιστήμη των νευρώνων, στην ψυχολογία, στα μαθηματικά, στις φυσικές επιστήμες και στη μηχανική. Δεν είναι αναγκαίο να πούμε ότι είναι *εδώ για να μείνουν και θα συνεχίσουν να αναπτύσσονται σε θεωρία, σχεδιασμό και εφαρμογές*.

Δραστηριότητα 2/1:

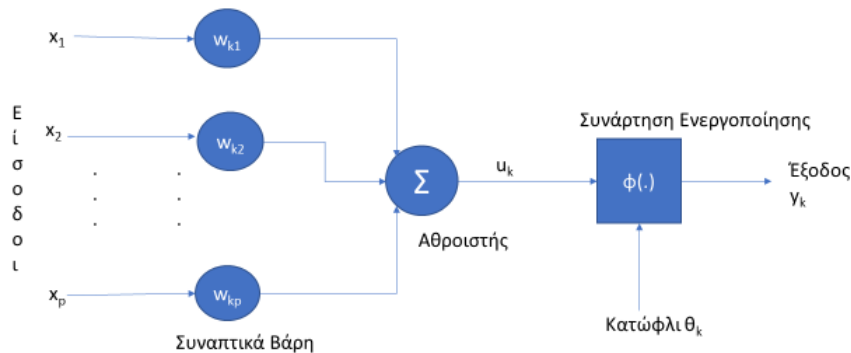
Ποια ήταν κατά τη γνώμη σας τα ορόσημα στην εξέλιξη των Τεχνητών Νευρωνικών Δικτύων; Να γράψετε μία αναφορά σε 2-3 σελίδες.

Υπόδειξη: Αφού μελετήσετε καλά την ενότητα 2.2, να ανατρέξετε στη σχετική βιβλιογραφία.

2.3 Το μοντέλο του τεχνητού νευρώνα

Ένας νευρώνας είναι μια μονάδα επεξεργασίας πληροφορίας θεμελιώδης για τη λειτουργία ενός νευρωνικού δικτύου. Το σχήμα 2.1 δείχνει το μοντέλο ενός νευρώνα. Τα τρία βασικά στοιχεία αυτού του μοντέλου είναι:

1. Ένα σύνολο από *συναψεις ή συνδετικούς κρίκους*, καθένα από τις οποίες χαρακτηρίζεται από ένα «βάρος» ή «δύναμη». Συγκεκριμένα ένα σήμα x_j στην είσοδο της συναψης j που συνδέεται στον νευρώνα k , πολλαπλασιάζεται με το συναπτικό βάρος w_{kj} . Το βάρος w_{kj} είναι θετικό αν η συναψη είναι διεγερτική (δηλαδή ωθεί τον νευρώνα να αποκριθεί στη διέγερση) και αρνητικό αν η συναψη είναι απαγορευτική (δηλαδή αποτρέπει τον νευρώνα να παράγει μια απόκριση).
2. Ένας *αθροιστής* για την πρόσθεση των σημάτων εισόδου, που παίρνουν βάρος από την αντίστοιχη συναψη. Αυτές οι λειτουργίες αποτελούν τον γραμμικό συνδυαστή.
3. Μια *συνάρτηση ενεργοποίησης* για τη μείωση του εύρους της εξόδου του νευρώνα.



Σχήμα 2.1 Το μη-γραμμικό μοντέλο ενός τεχνητού νευρώνα.

Το μοντέλο επίσης περιλαμβάνει ένα εξωτερικά εφαρμοζόμενο *κατώφλι* θ_k , που επηρεάζει την ελάττωση της εισόδου στην εφαρμοζόμενη συνάρτηση ενεργοποίησης που ακολουθεί. Στη βιβλιογραφία το κατώφλι αναφέρεται και σαν *πόλωση* (bias). Με μαθηματικούς όρους ένας νευρώνας k περιγράφεται από τις παρακάτω εξισώσεις:

$$u_k = \sum_{j=1}^p w_{kj} x_j \quad (2.1)$$

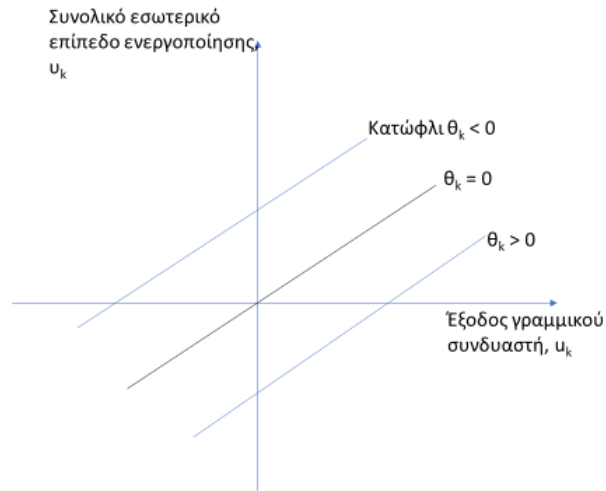
$$y_k = \phi(u_k - \theta_k) \quad (2.2)$$

όπου x_j είναι τα σήματα εισόδου και τα w_{kj} είναι τα synaptic βάρη του k .

Το u_k είναι η γραμμική συνδυαστική έξοδος του νευρώνα, θ_k είναι το κατώφλι, $\phi(\cdot)$ είναι η συνάρτηση ενεργοποίησης και y_k είναι το σήμα έξοδος του νευρώνα, που αναφέρεται και ως *πραγματική έξοδος*. Η χρήση του κατωφλίου θ_k έχει ως αποτέλεσμα την εφαρμογή ενός εγγενούς (affine) μετασχηματισμού της εξόδου u_k του γραμμικού συνδυαστή του σχήματος 2.1, όπως φαίνεται από την παρακάτω σχέση:

$$v_k = u_k - \theta_k \quad (2.3)$$

Η έξοδος u_k , ανάλογα με αν το θ_k είναι θετικό ή αρνητικό, μεταβάλλεται όπως φαίνεται στο σχήμα 2.2.



Σχήμα 2.2 Εγγενής μετασχηματισμός που παράγεται από την παρουσία κατωφλίου.

Το κατώφλι θ_k είναι εξωτερική παράμετρος του νευρώνα k . Έτσι μπορούμε να διαμορφώσουμε τον συνδυασμό των εξισώσεων (2.1), (2.2) ως εξής:

$$u_k = \sum_{j=0}^p w_{kj} x_j \quad (2.4)$$

και

$$y_k = \phi(u_k) \quad (2.5)$$

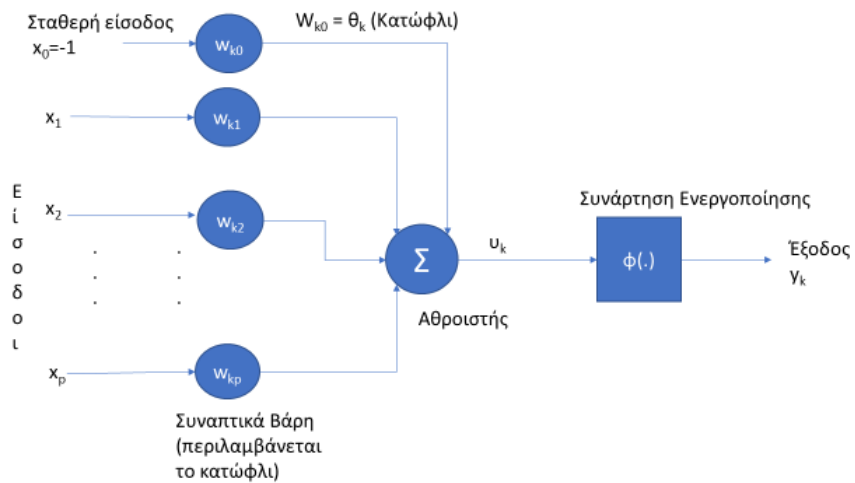
Στην εξίσωση (2.4) έχουμε προσθέσει μια νέα σύναψη της οποίας η είσοδος είναι:

$$x_0 = -1 \quad (2.6)$$

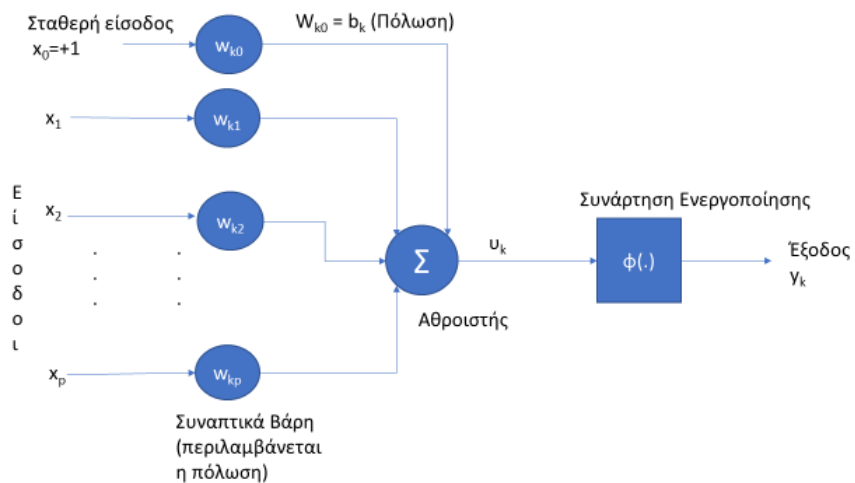
και της οποίας το βάρος είναι:

$$w_{k0} = \theta_k \quad (2.7)$$

Έτσι μπορούμε να αναδιαμορφώσουμε το μοντέλο του νευρώνα k , όπως φαίνεται στα σχήματα 2.3.α και 2.3.β που είναι ισοδύναμα με το σχήμα 2.1.



(α)



(β)

Σχήμα 2.3 Δύο άλλα μη-γραμμικά μοντέλα ενός νευρώνα.

Άσκηση 2.3/1:

Η χρήση κατωφλίου στο μοντέλο του τεχνητού νευρώνα χρησιμοποιείται για:

1. να επιβληθεί μια εξωτερική έξοδος στον νευρώνα,
2. να γραμμικοποιήσει την έξοδο του νευρώνα,
3. να ελαττώσει την είσοδο στη συνάρτηση ενεργοποίησης,
4. να μετασχηματίσει τη γραμμική έξοδο του νευρώνα,
5. τα 2 και 3,
6. τα 3 και 4.

2.3.1 Τύποι συναρτήσεων ενεργοποίησης

Η συνάρτηση ενεργοποίησης $\varphi(\cdot)$ ορίζει την έξοδο ενός νευρώνα συναρτήσει του επιπέδου ενεργοποίησης της εισόδου. Έχουμε τρεις βασικούς τύπους συναρτήσεων ενεργοποίησης:

1. Συνάρτηση Κατωφλίου

Για αυτόν τον τύπο συνάρτησης ενεργοποίησης, που περιγράφεται στο σχήμα 2.4a, έχουμε:

$$\varphi(v) = \begin{cases} 1, & v \geq 0 \\ 0, & v < 0 \end{cases} \quad (2.8)$$

Επομένως η έξοδος ενός νευρώνα k με την παραπάνω συνάρτηση ενεργοποίησης έχει τη μορφή:

$$y_k = \begin{cases} 1, & v_k \geq 0 \\ 0, & v_k < 0 \end{cases} \quad (2.9)$$

όπου v_k είναι το εσωτερικό επίπεδο ενεργοποίησης του νευρώνα και δίνεται από τη σχέση:

$$v_k = \sum_{j=0}^p w_{kj} x_j - \theta_k$$

2. Τμηματικά Γραμμική Συνάρτηση

Για την τμηματικά γραμμική συνάρτηση, έχουμε:

$$\varphi(v) = \begin{cases} 1, & v \geq 1/2 \\ v, & 1/2 > v > -1/2 \\ 0, & v \leq -1/2 \end{cases} \quad (2.10)$$

Αυτή η μορφή συνάρτησης ενεργοποίησης μπορεί να θεωρηθεί ως μια προσέγγιση ενός μη-γραμμικού ενισχυτή.

3. Σιγμοειδής

Είναι η πιο συνηθισμένη μορφή συνάρτησης ενεργοποίησης που χρησιμοποιείται στην κατασκευή τεχνητών νευρωνικών δικτύων. Ορίζεται ως αυστηρά αύξουσα συνάρτηση, η οποία παρουσιάζει εξομάλυνση και ασυμπτωτικές ιδιότητες. Ένα παράδειγμα είναι η *λογιστική συνάρτηση*, που ορίζεται από τη σχέση:

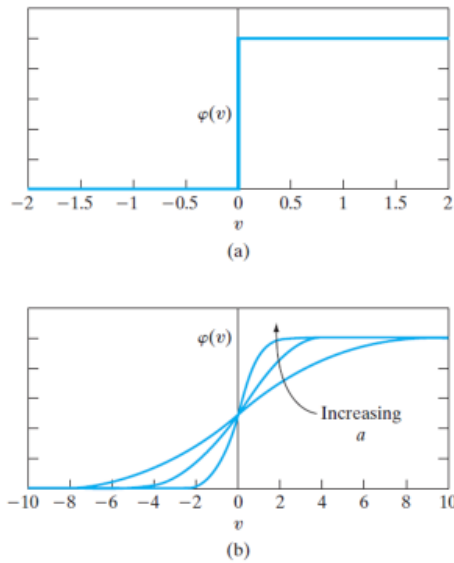
$$\phi(v) = \frac{1}{1 + \exp(-av)} \quad (2.11)$$

Μεταβάλλοντας την παράμετρο a παίρνουμε διαφορετικές συναρτήσεις, όπως φαίνεται στο σχήμα 2.4b.

Οι προηγούμενες συναρτήσεις ενεργοποίησης κυμαίνονται από 0 ως +1. Αν θέλουμε συνάρτηση που να κυμαίνεται από -1 ως +1, επαναπροσδιορίζουμε τη συνάρτηση κατωφλίου ως:

$$\phi(v) = \begin{cases} 1, & v > 0 \\ 0, & v = 0 \\ -1, & v < 0 \end{cases} \quad (2.12)$$

που ονομάζεται συνάρτηση *προσήμου* (*signum*) και συμβολίζεται ως $\text{sgn}(\cdot)$.



Σχήμα 2.4 (a) Συνάρτηση κατωφλίου, (b) Σιγμοειδής συνάρτηση (Haykin, 1999).

Άσκηση 2.3/2:

Στη λογιστική συνάρτηση η παράμετρος a χρησιμοποιείται για:

1. να κάνει τη συνάρτηση αυστηρά αύξουσα,
2. να παίρνουμε συναρτήσεις με διαφορετικές κλίσεις,
3. να εξομαλύνουμε τη συνάρτηση,
4. να δίνει ασυμπτωτικές ιδιότητες στη συνάρτηση.

Άσκηση 2.3/3:

Μια περιττή (odd) σιγμοειδής συνάρτηση είναι το αλγεβρικό σιγμοειδές, το οποίο δίνεται από τη σχέση:

$$\varphi(v) = \frac{v}{\sqrt{1+v^2}}$$

της οποίας οι οριακές τιμές είναι -1 και $+1$. Να δείξετε ότι η παράγωγος της $\varphi(v)$ ως προς v δίνεται από τη σχέση:

$$\frac{d\varphi}{dv} = \frac{\varphi^3(v)}{v^3}$$

Ποια είναι η τιμή αυτής της παραγώγου στην αρχή;

Άσκηση 2.3/4:

Ένας νευρώνας j δέχεται εισόδους από τέσσερις άλλους νευρώνες, των οποίων τα επίπεδα ενεργοποίησης είναι 10 , -20 , 4 και -2 . Τα αντίστοιχα συναπτικά βάρη του νευρώνα j είναι 0.8 , 0.2 , -1.0 και -0.9 . Να υπολογίσετε την έξοδο του νευρώνα j για τις επόμενες δύο καταστάσεις:

- (α) Ο νευρώνας είναι γραμμικός.
- (β) Ο νευρώνας παριστάνεται από το μοντέλο McCulloch-Pitts.

Να υποθέσετε ότι το κατώφλι που εφαρμόζεται στον νευρώνα είναι μηδενικό.

2.3.2 Θεωρώντας τα ΝΔ ως κατευθυνόμενους γράφους

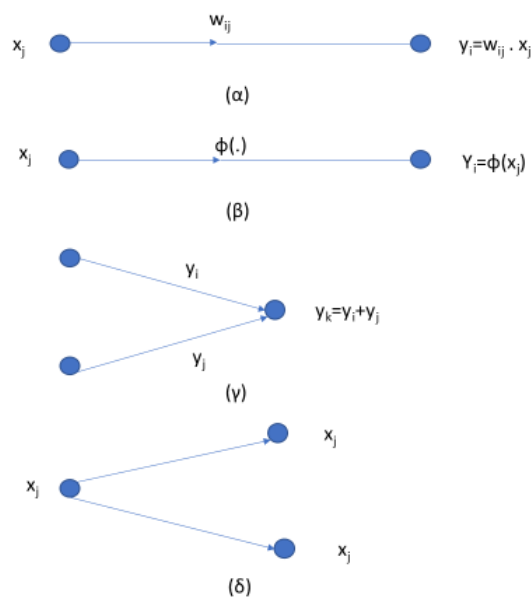
Απλοποιούμε εδώ την εμφάνιση των μοντέλων των νευρώνων, χρησιμοποιώντας την ιδέα των γράφων ροής σημάτων, χωρίς να θυσιάζουμε τίποτα από τις λειτουργικές λεπτομέρειες του μοντέλου. Ένας γράφος ροής σημάτων είναι ένα δίκτυο με κατευθυνόμενους κλάδους, που συνδέονται σε συγκεκριμένα σημεία, τους «κόμβους». Ένας τυπικός κόμβος j σχετίζεται με ένα σήμα κόμβου x_j .

Ακολουθούνται τρεις βασικοί κανόνες:

ΚΑΝΟΝΑΣ 1. Ένα σήμα ρέει κατά μήκος της σύνδεσης μόνο στην κατεύθυνση που ορίζεται από το βέλος.

Υπάρχουν δύο τύποι σύνδεσης:

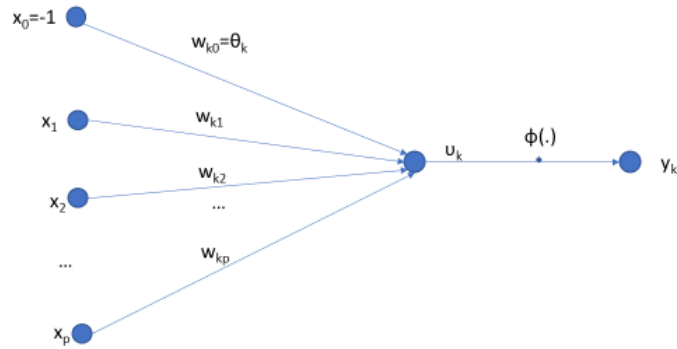
1. *Συναπτικές συνδέσεις*, όπου το σήμα x_j πολλαπλασιάζεται με το βάρος w_{kj} για να παραγάγει το σήμα y_k , όπως φαίνεται στο σχήμα 2.5α.
2. *Συνδέσεις ενεργοποίησης*, που περιγράφουν μια συνάρτηση $\phi(\cdot)$, η οποία δεν είναι γραμμική και φαίνεται στο σχήμα 2.5β.



Σχήμα 2.5 Βασικοί κανόνες για κατασκευή διαγραμμάτων ροής σήματος.

ΚΑΝΟΝΑΣ 2. Το σήμα κόμβου ισούται με το αλγεβρικό άθροισμα όλων των σημάτων που φτάνουν στον κόμβο (σχήμα 2.5γ).

ΚΑΝΟΝΑΣ 3. Το σήμα σε έναν κόμβο μεταβιβάζεται σε κάθε εξερχόμενη σύνδεση που ξεκινά από αυτόν, με τη μεταβίβαση να είναι ανεξάρτητη από τις συναρτήσεις μεταφοράς των εξερχόμενων συνδέσεων (σχήμα 5d). Στο σχήμα 2.6 περιγράφεται το διάγραμμα του σχήματος 2.3α.

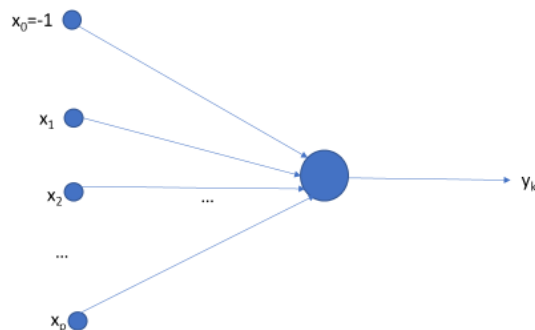


Σχήμα 2.6 Γράφος ροής σήματος ενός νευρώνα.

Μπορούμε τώρα να αναφέρουμε τους παρακάτω μαθηματικούς ορισμούς ενός νευρωνικού δικτύου: Ένα νευρωνικό δίκτυο είναι ένας κατευθυνόμενος γράφος που αποτελείται από κόμβους με συναπτικές διασυνδέσεις και συνδέσεις ενεργοποίησης και χαρακτηρίζονται από τις ακόλουθες ιδιότητες:

- Κάθε νευρώνας παριστάνεται από ένα σύνολο γραμμικών συναπτικών συνδέσεων, ένα εξωτερικά εφαρμοζόμενο κατώφλι και μια μη-γραμμική σύνδεση ενεργοποίησης. Το κατώφλι παριστάνεται από συναπτικές συνδέσεις με σήμα εισόδου σταθερής τιμής -1 .
- Οι συναπτικές συνδέσεις ενός νευρώνα ζυγίζουν τα αντίστοιχα σήματα εισόδου.
- Το άθροισμα των βαρών των σημάτων εισόδου καθορίζει το συνολικό εσωτερικό επίπεδο ενεργοποίησης του νευρώνα που ζητείται.
- Η σύνδεση ενεργοποίησης συνθλίβει (περιορίζει) το εσωτερικό επίπεδο ενεργοποίησης για την παραγωγή της εξόδου που παριστάνει την κατάσταση του νευρώνα.

Υπάρχουν όμως και μερικώς ολοκληρωμένοι κατευθυνόμενοι γράφοι που είναι γνωστοί ως «αρχιτεκτονικοί γράφοι». Ένας τέτοιος γράφος παριστάνεται στο παρακάτω σχήμα 2.7.



Σχήμα 2.7 Αρχιτεκτονικός γράφος ενός νευρώνα.

Πρέπει να σημειωθεί ότι τόσο στους κατευθυνόμενους όσο και στους αρχιτεκτονικούς γράφους οι κόμβοι που παριστάνονται με τετράγωνο δεν πραγματοποιούν καμία λειτουργία. Χρησιμοποιούνται απλώς για το πέρασμα των εισόδων στο δίκτυο.

Άσκηση 2.3/5:

Οι συναπτικές συνδέσεις διαφέρουν από τις συναρτήσεις ενεργοποίησης στο ότι:

1. δεν επιτρέπουν καμία λειτουργία,
2. ζυγίζουν τα αντίστοιχα σήματα εισόδου, ενώ οι συνδέσεις ενεργοποίησης συνθλίβουν το εσωτερικό επίπεδο ενεργοποίησης,
3. δεν παράγουν έξοδο.

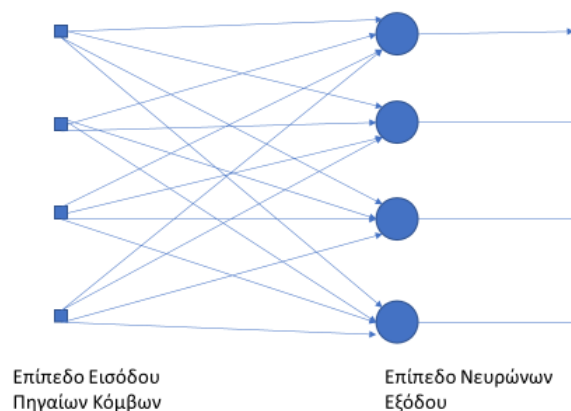
2.4 Αρχιτεκτονικές των Νευρωνικών Δικτύων

Ο τρόπος με τον οποίο οι νευρώνες ενός νευρωνικού δικτύου δομούνται είναι στενά συνδεδεμένος με τον αλγόριθμο εκμάθησης που χρησιμοποιείται για την εκπαίδευση του δικτύου. Ως εμπρός τροφοδότησης αναφέρονται τα δίκτυα στα οποία τα σήματα κατευθύνονται από την είσοδο στην έξοδο. Όταν οι εξοδοί κάποιων νευρώνων γίνονται εισοδοί σε νευρώνες προηγούμενων επιπέδων (προς το μέρος της εισόδου του δικτύου), τότε έχουμε ανάδραση.

Μπορούμε να διακρίνουμε τρεις διαφορετικές κλάσεις αρχιτεκτονικών δομών:

Ενός-επιπέδου Εμπρός-Τροφοδότησης Δίκτυα

Σε ένα τέτοιο δίκτυο οι νευρώνες είναι οργανωμένοι σε μορφή επιπέδων. Οι νευρώνες του πηγαιού επιπέδου δείχνουν στους νευρώνες του επόμενου επιπέδου αλλά όχι αντίστροφα, όπως στο Σχήμα 2.8.

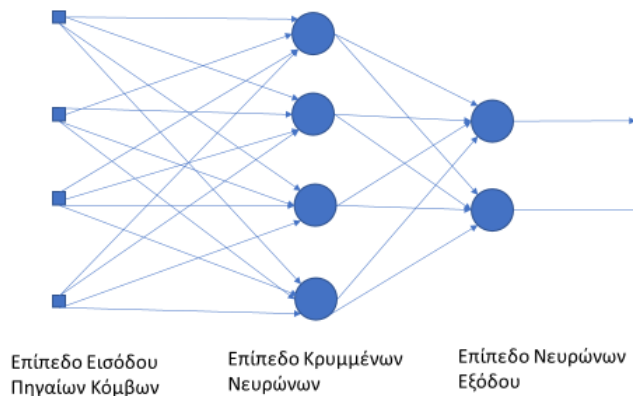


Σχήμα 2.8 Εμπρός-Τροφοδότησης δίκτυο με ένα επίπεδο νευρώνων.

Πολλαπλών-Επιπέδων Εμπρός-Τροφοδότησης Δίκτυα

Εδώ έχουμε περισσότερα του ενός κρυφά επίπεδα, των οποίων οι κόμβοι υπολογισμού ονομάζονται «κρυφοί νευρώνες». Τυπικά οι νευρώνες σε κάθε επίπεδο έχουν για εισόδους τα σήματα εξόδου του προηγούμενου μόνο επιπέδου.

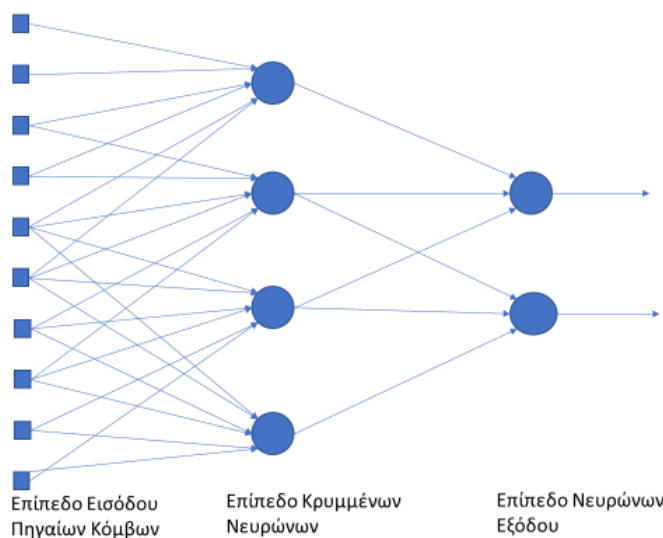
Στο σχήμα 2.9 έχουμε ένα πλήρως συνδεδεμένο νευρωνικό δίκτυο, με την έννοια ότι κάθε κόμβος συνδέεται με όλους τους κόμβους του αμέσως επόμενου επιπέδου.



Σχήμα 2.9 Πλήρως συνδεδεμένο δίκτυο εμπρός-τροφοδότησης με ένα κρυφό επίπεδο και ένα επίπεδο εξόδου.

Ένα τέτοιο δίκτυο περιγράφεται συνοπτικά με τον συμβολισμό 10-4-2. Αυτός ο συμβολισμός σημαίνει ότι το ΝΔ έχει δέκα εισόδους, ένα κρυφό επίπεδο με τέσσερις νευρώνες (κόμβους) και το επίπεδο εξόδου έχει δύο νευρώνες ή κόμβους. Γενικά ένα πολυεπίπεδο δίκτυο εμπρός τροφοδότησης με p εισόδους, m κρυφά επίπεδα με h_j , $j=1, \dots, m$ κόμβους ανά επίπεδο και n κόμβους εξόδου συμβολίζεται ως $p-h_1, h_2, \dots, h_m-n$.

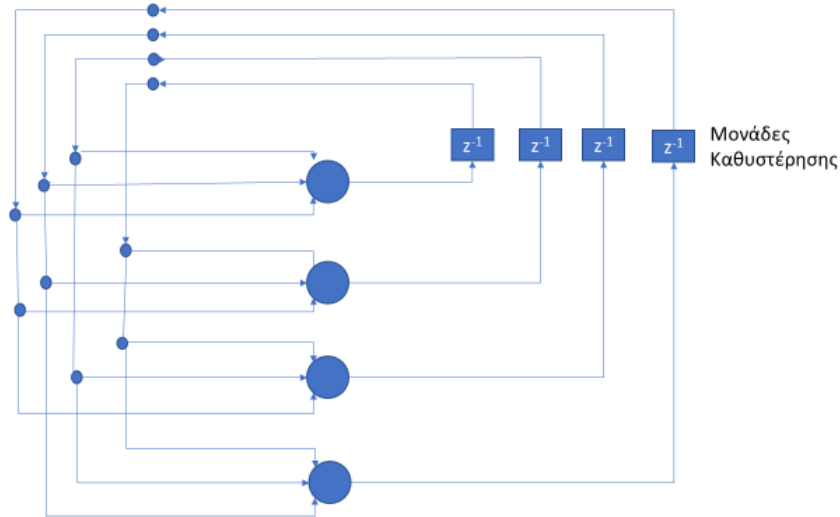
Αντίθετα, στο σχήμα 2.10 έχουμε ένα μερικώς συνδεδεμένο νευρωνικό δίκτυο. Στο παρακάτω δίκτυο υπάρχουν δέκα εισόδοι, ένα μέρος από τις οποίες τροφοδοτούνται σε κάθε νευρώνα του κρυφού επιπέδου. Επίσης, οι νευρώνες εξόδου δεν δέχονται για είσοδο όλες τις εξόδους των νευρώνων του κρυφού επιπέδου (όπως θα συνέβαινε σε ένα πλήρως διασυνδεδεμένο ΝΔ). Ένα τέτοιο δίκτυο μπορεί να χρησιμοποιηθεί π.χ. για την εκμάθηση του μοντέλου μιας χρονοσειράς, η οποία δειγματοληπτείται με δέκα δείγματα σε κάθε παράθυρο δειγματοληψίας.



Σχήμα 2.10 Μερικώς συνδεδεμένο δίκτυο εμπρός-τροφοδότησης.

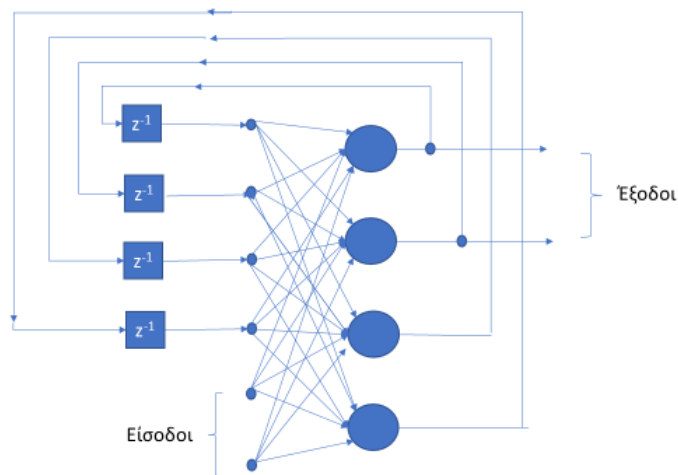
Αναδρομικά Δίκτυα

Τα Αναδρομικά Δίκτυα (αναφέρονται και ως επαναλαμβανόμενα, Recurrent Neural Networks – RNNs) διαφέρουν από τα εμπρός τροφοδότησης στο ότι έχουν έναν τουλάχιστον βρόχο ανάδρασης. Στα σχήματα 2.11 και 2.12 φαίνονται δύο αναδρομικά δίκτυα, το πρώτο χωρίς και το δεύτερο με κρυφούς νευρώνες.



Σχήμα 2.11 Αναδρομικό δίκτυο χωρίς ανατροφοδότηση και χωρίς κρυφούς νευρώνες.

Το δίκτυο του σχήματος 2.11 λέγεται και δίκτυο Hopfield. Είναι μια μη-γραμμική συσχετιστική μνήμη ή μνήμη διευθυνσιοδοτούμενη από τα περιεχόμενα. Η κύρια λειτουργία μιας τέτοιας μνήμης είναι η ανάκτηση ενός προτύπου που έχει αποθηκευτεί σε αυτήν, ένα είδος απόκρισης σε μια ελλιπή ή θορυβώδη έκδοση αυτού του προτύπου.



Σχήμα 2.12 Αναδρομικό δίκτυο με κρυφούς νευρώνες.

Άσκηση 2.4/6:

Να κατασκευάσετε τον αρχιτεκτονικό γράφο ενός πλήρως διασυνδεδεμένου νευρωνικού δικτύου εμπρός τροφοδότησης με $p=10$, $h_1=4$, $h_2=3$ και $n=1$.

Άσκηση 2.4/7:

Να θεωρήσετε ένα πολυεπίπεδο δίκτυο εμπρός τροφοδότησης, του οποίου όλοι οι νευρώνες λειτουργούν στη γραμμική τους περιοχή. Να αποδείξετε ότι ένα τέτοιο δίκτυο είναι ισοδύναμο με ένα δίκτυο εμπρός τροφοδότησης ενός επιπέδου.

Άσκηση 2.4/8:

Να κατασκευάσετε ένα πλήρως αναδρομικό νευρωνικό δίκτυο με πέντε νευρώνες, αλλά χωρίς αυτο-ανάδραση.

Δραστηριότητα 2/2:

Να ανατρέξετε στη βιβλιογραφία και να μελετήσετε τα δίκτυα Hopfield. Να ετοιμάσετε μια αναφορά, περίπου πέντε σελίδων, όπου να παρουσιάζετε τον αλγόριθμο εκπαίδευσης αυτών των δικτύων και ένα παράδειγμα.

Υπόδειξη: Σχετικό υλικό θα βρείτε στην αναφορά Haykin, 1999 στο κεφάλαιο 8.3 (σ. 289).

2.5 Σύνοψη κεφαλαίου

Σε αυτό το κεφάλαιο είδαμε ότι τα Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ.) αποτελούν μια προσπάθεια προσέγγισης της λειτουργίας του ανθρώπινου εγκεφάλου από μια μηχανή και έχουν την ικανότητα να εκτελούν υπολογισμούς με μαζικό παράλληλο τρόπο. Η αρχιτεκτονική τους βασίζεται στην αρχιτεκτονική των Βιολογικών Νευρωνικών Δικτύων. Αυτές οι ιδέες, αν και ήταν γνωστές από την αρχή του προηγούμενου αιώνα, άρχισαν να υλοποιούνται στα μέσα της δεκαετίας του 1940, όταν πρωτοπαρουσιάστηκε το μαθηματικό μοντέλο του νευρώνα το οποίο εξομοίωνε τη λειτουργία του βιολογικού νευρώνα. Στη συνέχεια, με την εμφάνιση των πρώτων υπολογιστών, έγινε η πρώτη προσπάθεια υλοποίησής τους. Μεγάλη ώθηση στην εφαρμογή τους έδωσε η εμφάνιση των ψηφιακών υπολογιστών. Τότε άρχισε η ευρεία υλοποίησή τους, τόσο με λογισμικό όσο και με υλικό. Έτσι τα απλά δίκτυα ενός επιπέδου εξελίχθηκαν σε δίκτυα πολλών επιπέδων. Παράλληλα παρουσιάστηκε στη διεθνή βιβλιογραφία μια μεγάλη ποικιλία αλγορίθμων εκπαίδευσης και εφαρμογών τους στην επίλυση πρακτικών προβλημάτων.

Μετά την ιστορική αναδρομή παρουσιάσαμε το μαθηματικό μοντέλο του τεχνητού νευρώνα. Αναφερθήκαμε τόσο στο γραμμικό όσο και στο μη-γραμμικό μοντέλο. Εκεί είχαμε την ευκαιρία να δούμε διάφορες συναρτήσεις που χρησιμοποιούνται ως συναρτήσεις ενεργοποίησης του νευρώνα. Η συνέχεια και η παραγωγισιμότητα είναι οι δύο βασικές ιδιότητες που πρέπει να έχει μια συνάρτηση για να μπορεί να χρησιμοποιηθεί ως συνάρτηση ενεργοποίησης. Επίσης, όπως θα δούμε στο επόμενο κεφάλαιο, είναι επιθυμητό το να υπολογίζεται εύκολα η παράγωγός τους. Ακόμη παρουσιάστηκαν οι έννοιες του κατευθυνόμενου και του αρχιτεκτονικού γράφου, οι οποίες βοηθούν στην απλοποίηση της αναπαράστασης των ΤΝΔ.

Τέλος, παρουσιάσαμε τις βασικές γνωστές αρχιτεκτονικές των Νευρωνικών Δικτύων. Στην αρχή ορίσαμε την έννοια της εμπρός τροφοδότησης, όπου τα σήματα κατευθύνονται από την είσοδο στην έξοδο. Στη συνέχεια ορίσαμε τα δίκτυα εμπρός τροφοδότησης ενός και πολλών επιπέδων, που είναι πλήρως ή μερικά διασυνδεδεμένα. Η παρουσίαση τελείωσε με τα αναδρομικά δίκτυα, με ή χωρίς αυτο-ανάδραση και τις δικτυωτές δομές.

Για περισσότερες λεπτομέρειες ο αναγνώστης παραπέμπεται στην αναφορά (Haykin, 1999), από όπου προέρχεται και το βασικό υλικό αυτού του κεφαλαίου.

Βιβλιογραφία

- Hagan, M.T., Demuth, H.B., & Beal, M. (1996). *Neural Network Design*. PWS Publishing Company.
- Haykin, S. (1999). *Neural Networks, A Comprehensive Foundation* (2η έκδ.). Prentice Hall International.
- Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs* (2η έκδ.). Springer - Verlag.

Απαντήσεις ασκήσεων

Άσκηση 2.3/1:

Η χρήση κατωφλίου στο μοντέλο του τεχνητού νευρώνα χρησιμοποιείται για:

1. να επιβληθεί μια εξωτερική έξοδος στον νευρώνα,
2. να γραμμικοποιήσει την έξοδο του νευρώνα,
3. να ελαττώσει την είσοδο στη συνάρτηση ενεργοποίησης,
4. να μετασχηματίσει τη γραμμική έξοδο του νευρώνα,
5. τα 2 και 3,
6. τα 3 και 4.

Απάντηση: Η σωστή απάντηση είναι η 6.

Άσκηση 2.3/2:

Στη λογιστική συνάρτηση η παράμετρος a χρησιμοποιείται για:

1. να κάνει τη συνάρτηση αυστηρά αύξουσα,
2. να παίρνουμε συναρτήσεις με διαφορετικές κλίσεις,
3. να εξομαλύνουμε τη συνάρτηση,
4. να δίνει ασυμπτωτικές ιδιότητες στη συνάρτηση.

Απάντηση: Η σωστή απάντηση είναι η 2, όπως φαίνεται και στο σχήμα 2.4b.

Άσκηση 2.3/3:

Μια περιττή (odd) σιγμοειδής συνάρτηση είναι το αλγεβρικό σιγμοειδές, το οποίο δίνεται από τη σχέση:

$$\varphi(v) = \frac{v}{\sqrt{1+v^2}}$$

της οποίας οι οριακές τιμές είναι -1 και $+1$. Να δείξετε ότι η παράγωγος της $\varphi(v)$ ως προς v δίνεται από τη σχέση:

$$\frac{d\varphi}{dv} = \frac{\varphi^3(v)}{v^3}$$

Ποια είναι η τιμή αυτής της παραγώγου στην αρχή;

Απάντηση: Για τον υπολογισμό της παραγώγου να χρησιμοποιήσετε τον κανόνα παραγώγισης συνθέτων συναρτήσεων. Επίσης, είναι $\varphi'(0)=1$.

Άσκηση 2.3/4:

Ένας νευρώνας j δέχεται εισόδους από τέσσερις άλλους νευρώνες, των οποίων τα επίπεδα ενεργοποίησης είναι 10, -20, 4 και -2. Τα αντίστοιχα συναπτικά βάρη του νευρώνα j είναι 0.8, 0.2, -1.0 και -0.9. Να υπολογίσετε την έξοδο του νευρώνα j για τις επόμενες δύο καταστάσεις:

- (α) Ο νευρώνας είναι γραμμικός.
- (β) Ο νευρώνας παριστάνεται από το μοντέλο McCulloch-Pitts.

Να υποθέσετε ότι το κατώφλι που εφαρμόζεται στον νευρώνα είναι μηδενικό.

Απάντηση: Από τη σχέση (1), για $p=4$, υπολογίζεται το επίπεδο ενεργοποίησης $v_j=1.8$.

Άρα είναι: (α) από την (4), $y_j=1.8$ και (β) από την (5), $y_j=+1$.

Άσκηση 2.3/5:

Οι συναπτικές συνδέσεις διαφέρουν από τις συναρτήσεις ενεργοποίησης στο ότι:

1. δεν επιτρέπουν καμία λειτουργία,
2. ζυγίζουν τα αντίστοιχα σήματα εισόδου, ενώ οι συνδέσεις ενεργοποίησης συνθλίβουν το εσωτερικό επίπεδο ενεργοποίησης,
3. δεν παράγουν έξοδο.

Απάντηση: Η σωστή απάντηση είναι η 2.

Άσκηση 2.4/6:

Να κατασκευάσετε τον αρχιτεκτονικό γράφο ενός πλήρως διασυνδεδεμένου νευρωνικού δικτύου εμπρός τροφοδότησης με $p=10$, $h_1=4$, $h_2=3$ και $n=1$.

Απάντηση: Θα χρησιμοποιήσετε τον γράφο του σχήματος 2.6 και θα εργασθείτε όπως στο σχήμα 2.9.

Άσκηση 2.4/7:

Να θεωρήσετε ένα πολυεπίπεδο δίκτυο εμπρός τροφοδότησης, του οποίου όλοι οι νευρώνες λειτουργούν στη γραμμική τους περιοχή. Να αποδείξετε ότι ένα τέτοιο δίκτυο είναι ισοδύναμο με ένα δίκτυο εμπρός τροφοδότησης ενός επιπέδου.

Απάντηση: Θεωρήστε για παράδειγμα ένα ΝΔ με ένα κρυφό επίπεδο και έναν νευρώνα εξόδου. Να γράψετε πρώτα τη σχέση για τον υπολογισμό της εξόδου του δικτύου και στη συνέχεια την αντίστοιχη σχέση για την έξοδο ενός νευρώνα του κρυφού επιπέδου. Στη συνέχεια συνδυάζοντας τις δύο σχέσεις προκύπτει εύκολα το ζητούμενο.

Άσκηση 2.4/8:

Να κατασκευάσετε ένα πλήρως αναδρομικό νευρωνικό δίκτυο με πέντε νευρώνες, αλλά χωρίς αυτο-ανάδραση.

Απάντηση: Μπορείτε να εργαστείτε όπως στο σχήμα 2.11.

Κεφάλαιο 3

Αλγόριθμοι Μάθησης

Σύνοψη

Στο δεύτερο κεφάλαιο αναφέραμε τα δομικά στοιχεία ενός Τεχνητού Νευρωνικού Δικτύου. Επίσης, μελετήσαμε διάφορες συναρτήσεις, οι οποίες χρησιμοποιούνται ως συναρτήσεις ενεργοποίησης των τεχνητών νευρώνων. Τέλος, παρουσιάσαμε τις γνωστές κατηγορίες και αρχιτεκτονικές των ΤΝΔ, που χρησιμοποιούνται για πρακτικές εφαρμογές.

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τους δύο βασικούς αλγόριθμους μάθησης (εκπαίδευσης) ΝΔ. Θα ξεκινήσουμε την παρουσίαση με τον αλγόριθμο εκπαίδευσης του απλού Perceptron (Αισθητήρα) και το θεώρημα της σύγκλισής του. Μετά θα παρουσιάσουμε τον βασικό αλγόριθμο εκπαίδευσης για δίκτυα εμπρός τροφοδότησης πολλών επιπέδων, που είναι γνωστά ως Perceptrons πολλών επιπέδων.

Ο αλγόριθμος εκπαίδευσης αυτών των δικτύων είναι ο πολύ δημοφιλής αλγόριθμος Πίσω Διάδοσης (ΠΔ) του λάθους. Αν και η παραγωγή του αλγόριθμου είναι αρκετά πολύπλοκη, ο ίδιος ο αλγόριθμος είναι εύκολο να υλοποιηθεί και έχει τύχει ευρείας εφαρμογής σε πολλά πρακτικά προβλήματα.

Συνοψίζοντας, μπορούμε να πούμε ότι σκοπός αυτού του κεφαλαίου είναι η παρουσίαση των βασικών αλγορίθμων εκπαίδευσης τόσο απλών όσο και πολυεπίπεδων ΤΝΔ. Έτσι ο αναγνώστης, μελετώντας τα τρία πρώτα κεφάλαια, θα έχει αποκτήσει μια γενική εικόνα για το τι είναι τα Τεχνητά Νευρωνικά Δίκτυα, που αποτελούν ένα σημαντικό τμήμα της Υπολογιστικής Νοημοσύνης, καθώς και πώς εκπαιδεύονται.

Προσδοκώμενα αποτελέσματα:

Όταν θα έχετε τελειώσει τη μελέτη αυτού του κεφαλαίου, θα μπορείτε να:

- υλοποιήσετε τον αλγόριθμο εκπαίδευσης του απλού Perceptron,
- υλοποιήσετε τον αλγόριθμο εκπαίδευσης EMT λάθους,
- υλοποιήσετε τον αλγόριθμο εκπαίδευσης ΠΔ του λάθους για Perceptrons πολλών επιπέδων,
- εξηγήσετε τη λειτουργία απλών και πολυεπίπεδων ΤΝΔ.

Λέξεις-Κλειδιά:

- Αλγόριθμοι εκπαίδευσης
- Κανόνας διόρθωσης του λάθους
- Εκπαιδευτικό σύνολο
- Γραμμικά διαχωριζόμενα πρότυπα
- Αλγόριθμος του perceptron
- Θεώρημα σύγκλισης
- Συνάρτηση κόστους
- Αλγόριθμος EMT λάθους
- Μέθοδος ταχύτερης καθόδου
- Αλγόριθμος ΠΔ του λάθους
- Λειτουργικά σήματα
- Σήματα λάθους
- Αλυσιδωτός κανόνας παραγωγής

Προαπαιτούμενη γνώση

Κεφάλαια 1, 2.

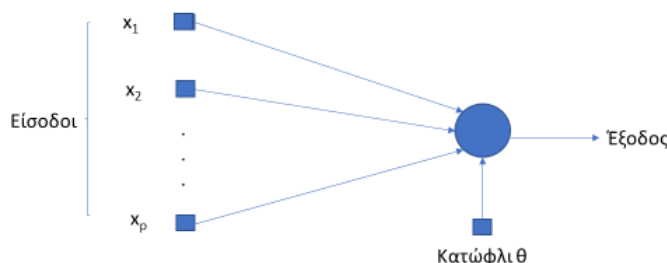
3.1 Εισαγωγικές Παρατηρήσεις

Στα δύο προηγούμενα κεφάλαια αναφέραμε ότι τα ΝΔ εκπαιδεύονται με τη βοήθεια παραδειγμάτων έτσι ώστε να μαθαίνουν το περιβάλλον τους. Ένα παράδειγμα περιλαμβάνει την είσοδο και την επιθυμητή έξοδο του δικτύου. Το σύνολο των παραδειγμάτων αποτελεί το εκπαιδευτικό σύνολο. Για την εκπαίδευση χρησιμοποιούνται κανόνες, οι οποίοι βασίζονται στην ελαχιστοποίηση του λάθους στην έξοδο του δικτύου. Ακολουθεί η γενίκευση, δηλαδή τα ΝΔ μαθαίνουν παραδείγματα για τα οποία δεν έχουν εκπαιδευτεί. Όπως υπάρχουν πολλές κατηγορίες ΝΔ, ανάλογα με την αρχιτεκτονική τους και τον τρόπο εκπαίδευσής τους, έτσι υπάρχει μεγάλη ποικιλία αλγορίθμων εκπαίδευσης, ανάλογα με τον κανόνα μάθησης και τον αλγόριθμο ελαχιστοποίησης που χρησιμοποιείται. Στις επόμενες ενότητες θα παρουσιάσουμε δύο βασικές κατηγορίες εκπαίδευσης των ΤΝΔ. Πρώτα θα παρουσιάσουμε τους δύο αλγορίθμους εκπαίδευσης απλών ΝΔ, ενός επιπέδου. Αυτά τα δίκτυα είναι κατάλληλα για την ταξινόμηση προτύπων που είναι γραμμικά διαχωριζόμενα. Στη συνέχεια θα ασχοληθούμε με δίκτυα πολλών επιπέδων που είναι γνωστά και ως Perceptrons πολλών επιπέδων· αυτά εκπαιδεύονται με τον αλγόριθμο Πίσω Διάδοσης του λάθους και είναι κατάλληλα για την ταξινόμηση προτύπων που δεν είναι γραμμικά διαχωριζόμενα. Αυτός είναι ο λόγος που η συγκεκριμένη κατηγορία ΤΝΔ έχει χρησιμοποιηθεί για την επίλυση μιας μεγάλης ποικιλίας πρακτικών προβλημάτων.

3.2 Ο αλγόριθμος μάθησης του Perceptron (Αισθητήρα)

Το Perceptron είναι η απλούστερη μορφή Νευρωνικού δικτύου, το οποίο χρησιμοποιείται για την ταξινόμηση ενός ειδικού τύπου προτύπων που είναι γραμμικά διαχωριζόμενα (δηλαδή πρότυπα που βρίσκονται στις αντίθετες πλευρές ενός υπερεπιπέδου, το οποίο ορίζει τις περιοχές απόφασης).

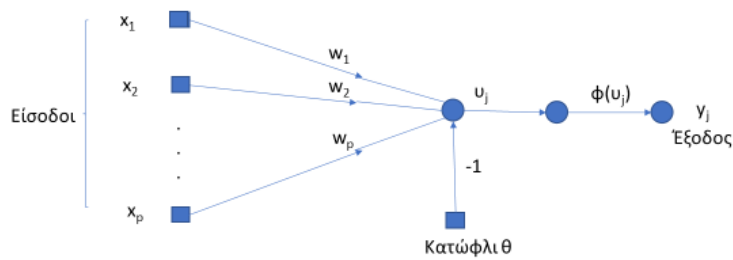
Ένα τέτοιο δίκτυο φαίνεται στο παρακάτω σχήμα 3.1:



Σχήμα 3.1 Αισθητήρας (Perceptron).

Προκειμένου να εκπαιδευτεί ένα τέτοιο Ν.Δ, ως αλγόριθμος εκπαίδευσης χρησιμοποιείται ο γνωστός κανόνας του Rosenblatt (Haykin, 1999). Αυτός ο κανόνας εφαρμόζεται στο οικείο μοντέλο Mc Culloch – Pitts για τον νευρώνα. Όπως είδαμε στο δεύτερο κεφάλαιο, αποτελείται από έναν γραμμικό συνδυαστή ακολουθούμενο από ένα στοιχείο κατωφλίου και η παραγόμενη έξοδος παίρνει τιμές ± 1 .

Θεωρούμε το διάγραμμα ροής σήματος του Perceptron, που φαίνεται στο σχήμα 2.



Σχήμα 3.2 Το διάγραμμα ροής σήματος του Perceptron.

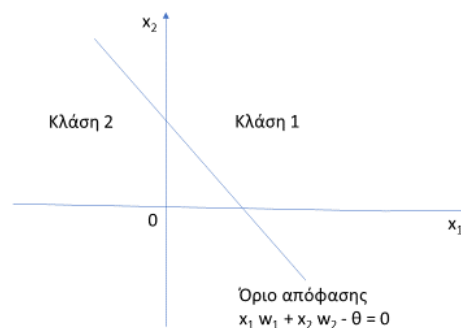
Η έξοδος του γραμμικού συνδυαστή υπολογίζεται εύκολα από το διάγραμμα του σχήματος 3.2 και είναι:

$$v = \sum_{i=1}^p w_i x_i - \theta \quad (3.1)$$

Σκοπός του Perceptron είναι να ταξινομήσει ένα σύνολο εισόδων (προτύπων) x_1, x_2, \dots, x_p σε μία από τις κλάσεις I_1 και I_2 . Ο κανόνας απόφασης για την ταξινόμηση είναι: ανάθεσε το σημείο που αναπαριστούν x_1, x_2, \dots, x_p τα στην κλάση I_1 , αν $y = +1$ και στην κλάση I_2 αν $y = -1$. Οι περιοχές απόφασης διαχωρίζονται από το υπερεπίπεδο που ορίζεται από τη σχέση:

$$v = \sum_{i=1}^p w_i x_i - \theta = 0 \quad \Leftrightarrow \quad w_1 x_1 + w_2 x_2 - \theta = 0 \quad (3.2)$$

Στο σχήμα 3.3 φαίνεται η γραμμική διαχωρισιμότητα για ένα διδιάστατο πρόβλημα ταξινόμησης με δύο κλάσεις.



Σχήμα 3.3 Το όριο και οι περιοχές απόφασης για ένα διδιάστατο πρόβλημα ταξινόμησης δύο κλάσεων.

Από το παραπάνω σχήμα φαίνεται το αποτέλεσμα της εφαρμογής του κατωφλίου, το οποίο μετατοπίζει το όριο απόφασης από την αρχή των αξόνων. Τα συναπτικά βάρη του Perceptron μπορούν να προσαρμοσθούν επαναληπτικά. Για την προσαρμογή του διανύσματος βαρών w χρησιμοποιούμε έναν κανόνα διόρθωσης λάθους, που είναι γνωστός ως κανόνας σύγκλισης του Perceptron και αναπτύσσεται στην επόμενη υποενότητα.

3.2.1 Το θεώρημα σύγκλισης του Perceptron

Για την παραγωγή του αλγορίθμου μάθησης διόρθωσης λάθους για ένα απλό Perceptron ενός επιπέδου, θα εργαστούμε με το μοντέλο ροής σήματος του σχήματος 3.4. Θεωρούμε το κατώφλι $\theta(n)$ ως ένα συναπτικό βάρος που είναι συνδεδεμένο σε μια σταθερή είσοδο -1 . Άρα το $(p + 1) \times 1$ διάνυσμα εισόδου είναι:

$$\mathbf{x}(n) = [-1, x_1(n), x_2(n), \dots, x_p(n)]^T \quad (3.3)$$

και αντίστοιχα ορίζουμε το $(p + 1) \times 1$ διάνυσμα βαρών:

$$\mathbf{w}(n) = [\theta(n), w_1(n), w_2(n), \dots, w_p(n)]^T \quad (3.4)$$

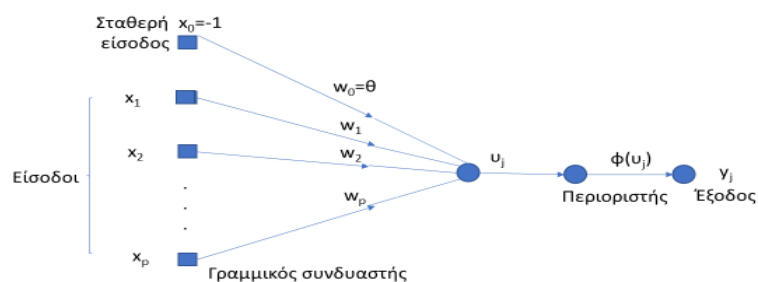
Η έξοδος του γραμμικού συνδυαστή είναι:

$$\mathbf{u}(n) = \mathbf{w}^T(n) \mathbf{x}(n) \quad (3.5)$$

Αν οι κλάσεις I_1 και I_2 είναι γραμμικά διαχωριζόμενες, τότε υπάρχει ένα διάνυσμα βαρών για το οποίο μπορούμε να ορίσουμε ότι:

$$\text{και} \quad \left. \begin{array}{l} \mathbf{w}^T \mathbf{x} \geq 0 \quad \forall \mathbf{x} \in I_1 \\ \mathbf{w}^T \mathbf{x} < 0 \quad \forall \mathbf{x} \in I_2 \end{array} \right\} \quad (3.6)$$

Το πρόβλημα για το απλό Perceptron είναι να βρούμε το διάνυσμα βαρών w , το οποίο ικανοποιεί τις ανισότητες (3.6).



Σχήμα 3.4 Ισοδύναμο διάγραμμα ροής σήματος του Perceptron.

Ο αλγόριθμος προσαρμογής των βαρών μπορεί τώρα να διατυπωθεί ως εξής:

1. Αν το n -στό μέλος του εκπαιδευτικού διανύσματος $\mathbf{x}(n)$ ταξινομείται σωστά από το διάνυσμα βαρών στην n -στή επανάληψη του αλγορίθμου, δεν γίνεται καμία διόρθωση στο $w(n)$, δηλαδή:

$$\begin{array}{l} \mathbf{w}(n + 1) = \mathbf{w}(n) \text{ αν } \mathbf{w}(n)^T \mathbf{x}(n) \geq 0 \text{ και } \mathbf{x}(n) \in I_1 \\ \text{και } \mathbf{w}(n + 1) = \mathbf{w}(n) \text{ αν } \mathbf{w}(n)^T \mathbf{x}(n) < 0 \text{ και } \mathbf{x}(n) \in I_2 \end{array} \quad (3.7)$$

2. Διαφορετικά, το διάνυσμα βαρών του Perceptron ενημερώνεται σύμφωνα με τον κανόνα:

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) - \eta(n) \mathbf{x}(n) \text{ αν } \mathbf{w}^T(n) \mathbf{x}(n) \geq 0 \text{ και } \mathbf{x}(n) \in I_2 \\ \text{και } \mathbf{w}(n+1) &= \mathbf{w}(n) + \eta(n) \mathbf{x}(n) \text{ αν } \mathbf{w}^T(n) \mathbf{x}(n) < 0 \text{ και } \mathbf{x}(n) \in I_1 \end{aligned} \quad (3.8)$$

όπου η παράμετρος ρυθμού – μάθησης $\eta(n)$ ελέγχει τις ρυθμίσεις που εφαρμόζονται στο διάνυσμα βαρών στην επανάληψη η .

Αν $\eta(n) = \eta - ct > 0$, τότε έχουμε έναν κανόνα σταθερά αυξανόμενης προσαρμογής (fixed increment adaptation rule) για το Perceptron. Για τη μελέτη της σύγκλισης αυτού του αλγορίθμου, ο αναγνώστης παραπέμπεται στην αναφορά [1, κεφάλαιο 4]. Εκεί αποδεικνύεται ότι ο κανόνας εκπαίδευσης του απλού Perceptron για γραμμικά διαχωριζόμενα πρότυπα συγκλίνει σε πεπερασμένο αριθμό επαναλήψεων.

3.2.2 Ανακεφαλαίωση

Στον πίνακα 2.1 παρουσιάζεται η ανακεφαλαίωση του αλγορίθμου σύγκλισης του Perceptron (Haykin, 1999).

Το σύμβολο $\text{sgn}(\cdot)$, που χρησιμοποιείται στο βήμα 3 του πίνακα, για τον υπολογισμό της πραγματικής απόκρισης του Perceptron, παριστάνει τη συνάρτηση προσήμου:

$$\text{sgn}(v) = \begin{cases} +1 & \text{αν } v > 0 \\ -1 & \text{αν } v < 0 \end{cases}$$

ΠΙΝΑΚΑΣ 2.1: Αλγόριθμος Σύγκλισης του Perceptron

Μεταβλητές και Παράμετροι

- $\mathbf{x}(n)$ = $(p+1) \times 1$ input vector
[-1, $x_1(n)$, $x_2(n)$, ..., $x_p(n)$]^T
- $\mathbf{w}(n)$ = $(p+1) \times 1$ weight vector
= [$\theta(n)$, $\mathbf{w}_1(n)$, $\mathbf{w}_2(n)$, ..., $\mathbf{w}_p(n)$]^T
- $\theta(n)$ = threshold (κατώφλι)
- $\mathbf{y}(n)$ = actual response (πραγματική έξοδος)
- $\mathbf{d}(n)$ = desired response (επιθυμητή έξοδος)
- η = learning - rate parameter, θετική σταθερά < 1

Step 1: Αρχικοποίηση

Θέσε $\mathbf{w}(0) = 0$. Κατόπιν κάνε τους υπολογισμούς για $\eta = 1, 2, \dots$

Step 2: Ενεργοποίηση

Στον χρόνο n ενεργοποίησε το Perceptron εφαρμόζοντας το συνεχές διάνυσμα εισόδου $\mathbf{x}(n)$ και το $\mathbf{d}(n)$.

Step 3: Υπολογισμός πραγματικής απόκρισης

Υπολόγισε την πραγματική απόκριση του Perceptron:

$$\mathbf{y}(n) = \text{sgn}[\mathbf{w}^T(n) \mathbf{x}(n)]$$

Step 4: Προσαρμογή διανύσματος βαρών

Προσάρμοσε τα βάρη του Perceptron:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta [\mathbf{d}(n) - \mathbf{y}(n)] \mathbf{x}(n) \text{ (E.C.L. rule)}$$

όπου:

$$\mathbf{d}(n) = \begin{cases} +1 & \\ -1 & \end{cases}, \text{ αν } \mathbf{x}(n) \text{ ανήκει στην κλάση } I_1 \\ \text{, αν } \mathbf{x}(n) \text{ ανήκει στην κλάση } I_2$$

Step 5: Αύξησε τον χρόνο η κατά μία μονάδα και πήγαινε στο βήμα 2.

Άσκηση 3.2/1:

Γραμμικά διαχωριζόμενα είναι τα πρότυπα

1. για τα οποία μπορούμε εύκολα να ορίσουμε περιοχές απόφασης,
2. για τα οποία οι περιοχές απόφασης είναι γραμμικές,
3. που βρίσκονται στις αντίθετες πλευρές ενός υπερεπιπέδου που ορίζει τις περιοχές απόφασης,
4. τα οποία χρησιμοποιούνται για την εκπαίδευση του Perceptron.

Άσκηση 3.2/2:

Να υποθέσετε ότι στο διάγραμμα ροής σήματος του Perceptron του σχήματος 3.4 η συνάρτηση ενεργοποίησης έχει τη μορφή:

$$\varphi(v) = \tanh\left(\frac{v}{2}\right)$$

όπου v είναι η γραμμική έξοδος του νευρώνα. Οι αποφάσεις ταξινόμησης από το Perceptron καθορίζονται από τον ακόλουθο κανόνα:

Το διάνυσμα παρατήρησης x ανήκει στην κλάση I_1 αν $y > \theta$.
Διαφορετικά το x ανήκει στην κλάση I_2 .

Άσκηση 3.2/3:

- (α) Το Perceptron μπορεί να χρησιμοποιηθεί για να μάθει διάφορες λογικές συναρτήσεις. Να δείξετε την υλοποίηση των δυαδικών λογικών συναρτήσεων AND, OR και COMPLEMENT.
- (β) Ένα βασικό μειονέκτημα του Perceptron είναι ότι δεν μπορεί να υλοποιήσει τη συνάρτηση EXCLUSIVE OR. Να εξηγήσετε τον λόγο για αυτόν τον περιορισμό.

3.3 Ο αλγόριθμος Πίσω Διάδοσης (ΠΔ) του λάθους

Σε αυτήν την ενότητα θα μελετήσουμε μία σπουδαία κλάση νευρωνικών δικτύων, τα δίκτυα εμπρός τροφοδότησης πολλών επιπέδων. Τυπικά ένα τέτοιο δίκτυο αποτελείται από ένα σύνολο αισθητήρων (πηγαιόι κόμβοι), που αποτελούν το επίπεδο εισόδου, ένα ή περισσότερα κρυφά επίπεδα (hidden layers) υπολογιστικών κόμβων και ένα επίπεδο υπολογιστικών κόμβων εξόδου. Το σήμα εισόδου διαδίδεται μέσα στο δίκτυο σε μία προς τα εμπρός κατεύθυνση, από επίπεδο σε επίπεδο. Αυτά τα νευρωνικά δίκτυα αναφέρονται ως Perceptrons πολλών επιπέδων (Multi Layer Perceptrons – MLPs), τα οποία είναι μια γενίκευση του απλού Perceptron.

Τα MLPs έχουν εφαρμοστεί με επιτυχία στην επίλυση δύσκολων και ποικίλων προβλημάτων, αφού εκπαιδευτούν με έναν επιβλεπόμενο τρόπο (supervised manner), με έναν πολύ δημοφιλή αλγόριθμο γνωστό ως αλγόριθμο πίσω διάδοσης του λάθους (error Back Propagation algorithm – BP). Αυτός ο αλγόριθμος βασίζεται στον κανόνα μάθησης διόρθωσης του λάθους (error correction learning rule).

Βασικά η διαδικασία της πίσω διάδοσης του λάθους αποτελείται από δύο περάσματα μέσω των διαφορετικών επιπέδων του δικτύου, ένα προς τα εμπρός πέραςμα (forward pass) και ένα προς τα πίσω πέραςμα (backward pass).

- Στο εμπρός πέραςμα ένα διάνυσμα εισόδου (input vector) εφαρμόζεται στους νευρώνες εισόδου του δικτύου και η επίδρασή του διαδίδεται μέσα στο δίκτυο από επίπεδο σε επίπεδο (layer by layer). Τελικά, ένα σύνολο από εξόδους παράγεται ως πραγματική απόκριση του δικτύου. Κατά τη διάρκεια του εμπρός περάσματος τα βάρη του δικτύου είναι σταθερά.
- Από την άλλη μεριά, κατά τη διάρκεια της πίσω διάδοσης τα βάρη προσαρμόζονται σε συμφωνία με τον κανόνα διόρθωσης λάθους.

Πιο συγκεκριμένα, η πραγματική απόκριση του δικτύου αφαιρείται από την επιθυμητή απόκριση για την παραγωγή ενός σήματος λάθους, που διαδίδεται προς τα πίσω στο δίκτυο, αντίθετα από την κατεύθυνση των συνδέσεων, από όπου προκύπτει και το όνομα της πίσω διάδοσης του λάθους.

Τα συναπτικά βάρη προσαρμόζονται έτσι ώστε να κάνουν την πραγματική απόκριση του δικτύου να πλησιάζει την επιθυμητή απόκριση.

Στη βιβλιογραφία ο αλγόριθμος πίσω διάδοσης του λάθους συχνά αναφέρεται και ως αλγόριθμος πίσω διάδοσης (Back Propagation Algorithm) ή πιο απλά Back Prop. Από εδώ και στο εξής θα αναφερόμαστε σε

αυτόν ως αλγόριθμος πίσω διάδοσης ή ΠΔ. Η διαδικασία μάθησης που εκτελείται με αυτόν τον αλγόριθμο ονομάζεται μάθηση πίσω διάδοσης.

Ένα Perceptron πολλών επιπέδων έχει τρία διακριτικά χαρακτηριστικά:

1. Το μοντέλο κάθε νευρώνα στο δίκτυο περιλαμβάνει μια μη γραμμικότητα στην έξοδο. Ένα σημαντικό σημείο στο οποίο πρέπει να δώσουμε έμφαση είναι ότι η μη γραμμικότητα είναι «εξομαλισμένη» (smooth), δηλαδή είναι παντού παραγωγίσιμη. Μία συνηθισμένη μορφή μη γραμμικότητας που ικανοποιεί αυτήν την προϋπόθεση είναι μια σιγμοειδής μη γραμμικότητα (sigmoidal nonlinearity) που ορίζεται από την παρακάτω λογιστική συνάρτηση:

$$y_j = \frac{1}{1 + \exp(-v_j)} \quad (3.9)$$

όπου v_j : η τιμή ενεργοποίησης του νευρώνα j

και y_j : η έξοδος του νευρώνα j

Η παρουσία μη γραμμικοτήτων είναι σημαντική, διότι διαφορετικά η σχέση εισόδου-εξόδου του δικτύου θα ελαττωνόταν σε αυτήν του perceptron ενός επιπέδου. Επιπλέον, η χρήση της λογιστικής συνάρτησης έχει βιολογικά κίνητρα, μιας και προσπαθεί να δικαιολογήσει την επίμονη φάση των πραγματικών νευρώνων (χαρακτηριστικό των πραγματικών βιολογικών νευρώνων είναι ότι δεν έχουν δυαδικές εξόδους, αλλά η έξοδός τους έχει συνεχώς κάποια τιμή).

2. Το δίκτυο περιέχει ένα ή περισσότερα κρυφά επίπεδα από νευρώνες τα οποία δεν είναι τμήμα της εισόδου ή της εξόδου του δικτύου. Αυτοί οι κρυφοί νευρώνες τού δίνουν τη δυνατότητα να μάθει πολύπλοκες εργασίες με το να εξάγουν προοδευτικά τα πιο σημαντικά χαρακτηριστικά από τα διανύσματα εισόδου.
3. Το δίκτυο επιδεικνύει έναν υψηλό βαθμό διασύνδεσης (connectivity) που καθορίζεται από τις συνδέσεις (συνάψεις) του δικτύου. Μία αλλαγή στον τρόπο διασύνδεσης του δικτύου απαιτεί αλλαγή στον πληθυσμό των συνδέσεων ή στα βάρη τους.

Πράγματι το Perceptron πολλών επιπέδων αντλεί την υπολογιστική του ισχύ από τον συνδυασμό αυτών των χαρακτηριστικών, μαζί με την ικανότητα να μαθαίνει από την εμπειρία μέσω της εκπαίδευσης. Αυτά τα ιδιοχαρακτηριστικά όμως είναι επίσης υπεύθυνα για τις ελλείψεις στην παρούσα κατάσταση της γνώσης μας πάνω στη συμπεριφορά του δικτύου.

- Πρώτον η παρουσία μιας κατανομημένης μορφής μη γραμμικότητας και η υψηλή διασύνδεση του δικτύου κάνουν τη θεωρητική ανάλυση ενός Perceptron πολλών επιπέδων πολύ δύσκολο να επιχειρηθεί.
- Δεύτερον η χρήση κρυφών νευρώνων κάνει τη διαδικασία μάθησης πιο δύσκολη στο να κατανοηθεί. Κατά μία έννοια, η διαδικασία μάθησης πρέπει να αποφασίσει ποια χαρακτηριστικά των διανυσμάτων εισόδου πρέπει να παρασταθούν από τους κρυφούς νευρώνες. Επομένως η διαδικασία μάθησης γίνεται πιο δύσκολη, επειδή η έρευνα πρέπει να διεξαχθεί σε έναν πολύ μεγαλύτερο χώρο από πιθανές συναρτήσεις και πρέπει να γίνει μια επιλογή μεταξύ εναλλακτικών αναπαραστάσεων του διανύσματος εισόδου.

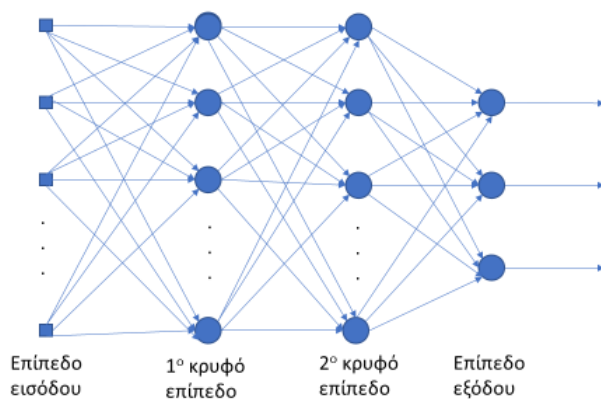
Η ανάπτυξη του αλγόριθμου πίσω διάδοσης αποτελεί σταθμό στα νευρωνικά δίκτυα, γιατί παρέχει μια υπολογιστικά αποδοτική μέθοδο για την εκπαίδευση πολυεπίπεδων Perceptrons. Αν και δεν μπορεί να παράσχει λύσεις για όλα τα επιλύσιμα προβλήματα, είναι δίκαιο να πούμε ότι έβαλε στην άκρη την αρνητική προκατάληψη για τη μάθηση σε πολυεπίπεδες μηχανές που μπορεί να είχε συναχθεί από το βιβλίο των Minsky και Papert (1969).

Στο σχήμα 3.5 φαίνεται η γραφική αναπαράσταση ενός πολυεπίπεδου Perceptron με δύο κρυφά επίπεδα (hidden layers). Το δίκτυο που φαίνεται εδώ είναι πλήρως διασυνδεδεμένο (fully connected), πράγμα που σημαίνει ότι ένας νευρώνας οποιουδήποτε επιπέδου είναι συνδεδεμένος με όλους τους νευρώνες του

προηγούμενου επιπέδου. Η ροή του σήματος στο δίκτυο προχωρά σε μια προς τα εμπρός κατεύθυνση, από τα αριστερά προς τα δεξιά από επίπεδο σε επίπεδο.

Στο σχήμα 3.6 (δανεισμένο από τον Haykin, 1999) απεικονίζεται ένα τμήμα ενός MLP. Σε αυτό το δίκτυο αναγνωρίζονται δύο είδη σημάτων.

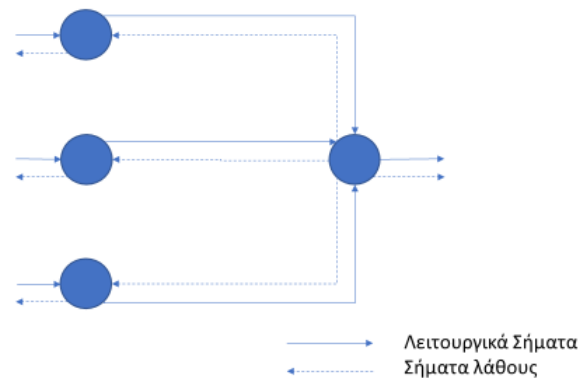
1. **Λειτουργικά σήματα:** Ένα λειτουργικό σήμα (function signal) είναι ένα σήμα εισόδου (ερέθισμα) που εισέρχεται από την απόληξη εισόδου του δικτύου, διαδίδεται προς τα εμπρός μέσω του δικτύου και εξέρχεται από την έξοδό του ως ένα σήμα εξόδου. Το ονομάζουμε “function signal” για δύο λόγους:



Σχήμα 3.5 Ο αρχιτεκτονικός γράφος ενός πολυεπίπεδου perceptron με δύο κρυφά επίπεδα.

- Πρώτον γιατί υποτίθεται ότι επιτελεί μια χρήσιμη συνάρτηση στην έξοδο του δικτύου.
 - Δεύτερον γιατί σε κάθε νευρώνα του δικτύου, μέσω του οποίου περνά ένα λειτουργικό σήμα, το σήμα υπολογίζεται ως συνάρτηση των εισόδων και των συσχετιζόμενων βαρών που εφαρμόζονται στον νευρώνα.
2. **Σήμα λάθους:** Ένα σήμα λάθους (error signal) δημιουργείται σε έναν νευρώνα εξόδου του δικτύου και διαδίδεται προς τα πίσω (layer by layer) μέσω του δικτύου. Αναφερόμαστε σε αυτό ως “error signal”, επειδή ο υπολογισμός του από κάθε νευρώνα του δικτύου εμπεριέχει μια συνάρτηση εξαρτώμενη από το λάθος στη μια ή στην άλλη μορφή.

Οι νευρώνες εξόδου αποτελούν το επίπεδο εξόδου του δικτύου. Οι υπόλοιποι σχηματίζουν τα κρυφά επίπεδά του. Οι κρυφές μονάδες δεν ανήκουν στο επίπεδο εισόδου ή εξόδου του δικτύου, γι’ αυτό ονομάζονται και κρυφές (hidden). Το πρώτο κρυφό επίπεδο τροφοδοτείται από το επίπεδο εισόδου που αποτελείται από τις αισθητήριες μονάδες, ενώ οι εξόδοι που προκύπτουν από το πρώτο κρυφό επίπεδο εφαρμόζονται με τη σειρά τους στο επόμενο κρυφό επίπεδο και ούτω καθεξής για το υπόλοιπο του δικτύου.



Σχήμα 3.6 Απεικόνιση των διευθύνσεων των δύο βασικών σημάτων ροής σε ένα πολυεπίπεδο Perceptron.

Κάθε κρυφός νευρώνας ή νευρώνας εξόδου του πολυεπίπεδου Perceptron σχεδιάζεται έτσι ώστε να επιτελούνται δύο υπολογισμοί:

1. Ο υπολογισμός του λειτουργικού σήματος που εμφανίζεται στην έξοδο ενός νευρώνα, το οποίο εκφράζεται ως μια συνεχής μη γραμμική συνάρτηση των σημάτων εισόδου και των συναπτικών βαρών που σχετίζονται με τον νευρώνα.
2. Ο υπολογισμός μιας στιγμιαίας εκτίμησης του διανύσματος κλίσης, ο οποίος χρειάζεται για την πίσω διάδοση μέσω του δικτύου.

Η παραγωγή του αλγόριθμου πίσω διάδοσης είναι πολύπλοκη. Για να διευκολύνουμε τη μαθηματική επιβάρυνση που εμπεριέχεται σε αυτήν τη διαδικασία παρουσιάζουμε μια σύνοψη από συμβολισμούς που χρησιμοποιούνται στη συγκεκριμένη παραγωγή.

- Τα i , j και k αντιστοιχούν σε διαφορετικούς νευρώνες, με τα σήματα να διαδίδονται μέσα από το δίκτυο από τα αριστερά προς τα δεξιά· ο νευρώνας j βρίσκεται ένα επίπεδο αριστερά από τον νευρώνα i και ο νευρώνας k ένα επίπεδο αριστερά από τον νευρώνα j , όταν ο j είναι μια κρυφή μονάδα.
- Η επανάληψη n αντιστοιχεί στο n -οστό διάνυσμα εκπαίδευσης που δόθηκε ως είσοδος στο δίκτυο.
- Το σύμβολο $\mathbf{E}(n)$ είναι το στιγμιαίο άθροισμα των τετραγωνικών λαθών στην επανάληψη n . Ο μέσος όρος του $\mathbf{E}(n)$ όλων των τιμών του n είναι το μέσο τετραγωνικό λάθος \mathbf{E}_{av} .
- Το σύμβολο $e_j(n)$ αντιστοιχεί στο σήμα λάθους στην έξοδο του νευρώνα j για την επανάληψη n .
- Το σύμβολο $d_j(n)$ αντιστοιχεί στην επιθυμητή απόκριση για τον νευρώνα j και χρησιμοποιείται στον υπολογισμό του $e_j(n)$.
- Το σύμβολο $y_j(n)$ αντιστοιχεί στο λειτουργικό σήμα στην έξοδο του νευρώνα j για την επανάληψη n .
- Το σύμβολο $w_{ij}(n)$ είναι το συναπτικό βάρος που συνδέει τον νευρώνα i με τον νευρώνα j κατά τη διάρκεια της επανάληψης n . Η ποσότητα κατά την οποία διορθώνεται το βάρος της σύναψης στη επανάληψη n συμβολίζεται με $\Delta w_{ij}(n)$.
- Η τιμή ενεργοποίησης του νευρώνα j στην επανάληψη n συμβολίζεται με $v_j(n)$.
- Η συνάρτηση ενεργοποίησης του νευρώνα j συμβολίζεται με $\phi_j(\cdot)$.

- Το κατώφλι το οποίο εφαρμόζεται στον νευρώνα j συμβολίζεται με $\vartheta_j(n)$. Συνήθως αναπαρίσταται με μια σύναψη βάρους $W_{j0} = \vartheta_j$ και είναι συνδεδεμένο σε μια σταθερή είσοδο που ισούται με -1 .
- Το i -οστό στοιχείο του διανύσματος εισόδου συμβολίζεται με $x_i(n)$.
- Το k -οστό στοιχείο του συνολικού διανύσματος εξόδου συμβολίζεται με $O_k(n)$.
- Η παράμετρος μάθησης συμβολίζεται με η .

3.3.1 Παραγωγή του αλγορίθμου Πίσω Διάδοσης

Το σήμα λάθους στην έξοδο του νευρώνα j στην επανάληψη n ορίζεται από τη σχέση:

$$e_j = d_j - y_j \quad \text{όπου ο νευρώνας } j \text{ είναι κόμβος εξόδου} \quad (3.10)$$

Ορίζοντας τη στιγμιαία τιμή του τετραγωνικού λάθους για τον νευρώνα j : $1/2 e_j^2(n)$, το στιγμιαίο άθροισμα των τετραγωνικών λαθών του δικτύου γράφεται ως εξής:

$$E(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad (3.11)$$

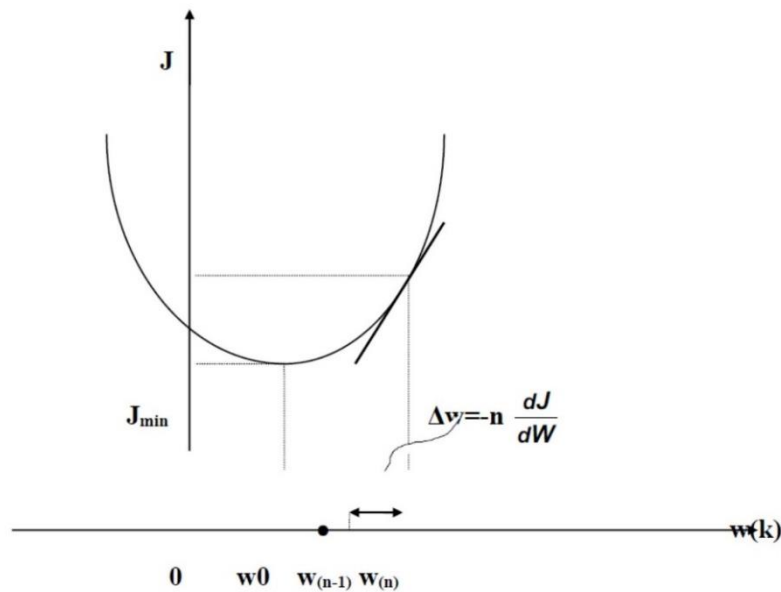
όπου το σύνολο C περιλαμβάνει όλους τους νευρώνες του επιπέδου εξόδου του δικτύου. Έστω N ο συνολικός αριθμός διανυσμάτων στο σύνολο εκπαίδευσης. Το μέσο τετραγωνικό λάθος για όλο το σύνολο εκπαίδευσης είναι:

$$E_{av} = \frac{1}{N} \sum_{n=1}^N E(n) \quad (3.12)$$

Το στιγμιαίο άθροισμα των τετραγωνικών λαθών E και κατά συνέπεια και το μέσο τετραγωνικό λάθος E_{av} είναι μια συνάρτηση όλων των ελεύθερων παραμέτρων (π.χ. συναπτικά βάρη και κατώφλια) του δικτύου. Για ένα δοσμένο εκπαιδευτικό σύνολο (training set) η E_{av} αντιπροσωπεύει τη συνάρτηση κόστους (cost function) ως το μέτρο για την απόδοση μάθησης του συνόλου εκπαίδευσης. Ο αντικειμενικός σκοπός της διαδικασίας μάθησης είναι να προσαρμόσει τις ελεύθερες παραμέτρους του δικτύου έτσι ώστε να ελαχιστοποιήσει το E_{av} . Όπως θα δούμε στη συνέχεια, για να βρεθεί το ελάχιστο της συνάρτησης κόστους, χρησιμοποιείται η μέθοδος Απότομης Κλίσης (Steepest Descent).

Σύμφωνα με αυτήν τη μέθοδο υποθέτουμε ότι τα βάρη του δικτύου είναι χρονικά μεταβαλλόμενα και ότι οι τιμές τους διορθώνονται με έναν επαναληπτικό τρόπο κατά μήκος της επιφάνειας λάθους, μετακινώντας τα προοδευτικά προς τη βέλτιστη λύση. Η μέθοδος ταχύτερης καθόδου έχει στόχο της τη συνεχή αναζήτηση βέλτιστης λύσης.

Όπως φαίνεται και στο σχήμα 3.7, για να είναι επιτυχείς οι διορθώσεις πρέπει να γίνονται σε κατεύθυνση αντίθετη προς το διάνυσμα κλίσης, του οποίου τα στοιχεία καθορίζονται από την κλίση $\nabla_{w_k J}$, για $k=1, 2, \dots, p$, όπου J η συνάρτηση κόστους.



Σχήμα 3.7 Η μέθοδος *steepest descent* για την προσαρμογή ενός βάρους w .

Επειδή ο υπολογισμός της κλίσης γίνεται πάνω σε όλα τα δείγματα (training set), χρειάζεται τεράστιος αριθμός υπολογισμών. Έτσι αν ο αριθμός των εισόδων ή των δειγμάτων αυξηθεί πολύ, τότε ο αλγόριθμος γίνεται πολύ αργός.

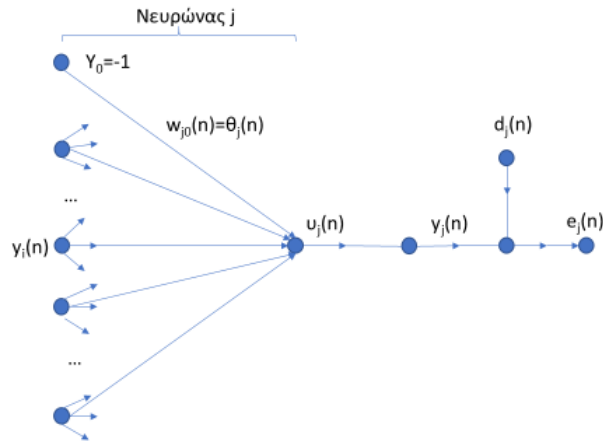
Για την περίπτωση μεγάλου όγκου δεδομένων, προκειμένου να βελτιωθεί το πρόβλημα της αργής σύγκλισης, χρησιμοποιείται ο αλγόριθμος Στοχαστικής Καθόδου Κλίσης (Stochastic Gradient Descent SGD). Ο αλγόριθμος SGD επιλέγει τυχαία σε κάθε επανάληψη ένα δεδομένο από το σύνολο των δειγμάτων, με στόχο να μειώσει σημαντικά το υπολογιστικό κόστος. Στην πράξη συνηθίζεται να δειγματοληπτείται ένα μικρό υποσύνολο από τα δείγματα σε κάθε επανάληψη. Αυτή η τεχνική ονομάζεται mini-batch GD και προσπαθεί να φέρει μια ισορροπία μεταξύ Steepest Descent και του SGD.

3.3.2 Διαδικασία Μάθησης

Θεωρούμε μια απλή μέθοδο εκπαίδευσης στην οποία τα βάρη ενημερώνονται σε μια πρότυπο προς πρότυπο (pattern by pattern) βάση. Οι προσαρμογές (ρυθμίσεις) των βαρών γίνονται σε συμφωνία με τα αντίστοιχα λάθη που υπολογίζονται για κάθε πρότυπο το οποίο παρουσιάζεται στην είσοδο.

Ο αριθμητικός μέσος όρος αυτών των ατομικών αλλαγών στα βάρη πάνω στο σύνολο εκπαίδευσης είναι λοιπόν μια εκτίμηση της πραγματικής αλλαγής που θα συνέβαινε στα βάρη από τη μεταβολή (ρύθμισή) τους για την ελαχιστοποίηση της συνάρτησης κόστους. Εάν πάνω στο συνολικό σύνολο εκπαίδευσης.

Στο σχήμα 3.8 απεικονίζεται ο νευρώνας j ο οποίος δέχεται ένα σύνολο από λειτουργικά σήματα που παράγονται από το επίπεδο στα αριστερά του.



Σχήμα 3.8 Γράφος που δείχνει με λεπτομέρεια τη ροή των σημάτων στον νευρώνα εξόδου j .

Για τον νευρώνα j έχουμε:

$$v_j(n) = \sum_{i=1}^p w_{ji}(n) y_i(n) \quad (3.13)$$

όπου p είναι ο συνολικός αριθμός εισόδων (εξαιρούμε το κατώφλι) που εφαρμόζονται στον νευρώνα j . Επίσης, θέτουμε $w_{j0} = \theta_j$.

Επομένως το λειτουργικό σήμα $y_j(n)$ στην έξοδο του νευρώνα j θα είναι

$$y_j(n) = \varphi_j(v_j(n)) \quad (3.14)$$

Με τρόπο παρόμοιο με τον LMS αλγόριθμο ο αλγόριθμος πίσω διάδοσης εφαρμόζει μια διόρθωση $\Delta w_{ji}(n)$ στο συναπτικό βάρος $w_{ji}(n)$, η οποία είναι ανάλογη της στιγμιαίας κλίσης $\frac{\partial E(n)}{\partial w_{ji}(n)}$. Σύμφωνα με τον αλυσιδωτό κανόνα (Chain Rule) μπορούμε να εκφράσουμε την κλίση ως εξής:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \cdot \frac{\partial e_j(n)}{\partial y_j(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} \cdot \frac{\partial v_j(n)}{\partial w_{ji}(n)} \quad (3.15)$$

Η κλίση $\frac{\partial E(n)}{\partial w_{ji}(n)}$ αντιπροσωπεύει έναν παράγοντα ευαισθησίας, καθορίζοντας την κατεύθυνση έρευνας στον χώρο των βαρών για το συναπτικό βάρος w_{ji} . Παραγωγίζοντας και τις δύο πλευρές της εξίσωσης (3.11) ως προς $e_j(n)$ παίρνουμε:

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n) \quad (3.16)$$

Παραγωγίζοντας τώρα και τις δυο πλευρές της εξίσωσης (3.10) με το $y_j(n)$ παίρνουμε:

j

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \quad (3.17)$$

Τελικά, παραγωγίζοντας την (3.14) και (3.13) ως προς $v_j(n)$ και $w_{ji}(n)$ αντίστοιχα, παίρνουμε:

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \phi'_j(v_j(n)) \quad (3.18)$$

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = -1 \quad (3.19)$$

Αντικαθιστώντας τις εξισώσεις (3.16), (3.17), (3.18) και (3.19) στην (3.15) παίρνουμε:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -e_j(n) \cdot \phi'_j(v_j(n)) \cdot y_j(n) \quad (3.20)$$

Η διόρθωση $\Delta w_{ji}(n)$ που εφαρμόζεται στο $w_{ji}(n)$ καθορίζεται από τον δέλτα κανόνα:

$$\Delta w_{ji}(n) = -\eta \cdot \frac{\partial E(n)}{\partial w_{ji}(n)} \quad (3.21)$$

όπου το η ονομάζεται παράμετρος ρυθμού μάθησης (learning rate parameter). Η χρήση του αρνητικού συμβόλου (-) στην (3.21) ερμηνεύεται ως πτώση της κλίσης στον χώρο των βαρών. Από τις (3.20) και (3.21) παίρνουμε:

$$\Delta w_{ji}(n) = \eta \cdot \delta_j(n) \cdot y_j(n) \quad (3.22)$$

όπου η τοπική κλίση $\delta_j(n)$ ορίζεται από τη σχέση:

$$\delta_j(n) = - \frac{\partial E(n)}{\partial e_j(n)} \cdot \frac{\partial e_j(n)}{\partial y_j(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} = e_j(n) \cdot \phi'_j(v_j(n)) \quad (3.23)$$

Η τοπική κλίση δείχνει τις απαιτούμενες αλλαγές στα βάρη. Σύμφωνα με την εξίσωση (3.23) η τοπική κλίση $\delta_j(n)$ για τον νευρώνα εξόδου j είναι ίση με το γινόμενο του αντίστοιχου σήματος λάθους $e_j(n)$ και της παραγώγου $\phi'_j(v_j(n))$ της συνάρτησης ενεργοποίησής του. Από τις εξισώσεις (3.22) και (3.23) φαίνεται ότι ένας παράγοντας κλειδί που εμπλέκεται στον υπολογισμό της προσαρμογής (του βάρους) $\Delta w_{ji}(n)$ είναι το σήμα στην έξοδο του νευρώνα j . Στο σημείο αυτό μπορούμε να διακρίνουμε δύο περιπτώσεις, ανάλογα με το πού είναι τοποθετημένος ο νευρώνας j στο δίκτυο. Στην περίπτωση (I) ο j είναι ένας νευρώνας εξόδου. Την περίπτωση αυτή είναι απλό να τη χειριστούμε, διότι κάθε νευρώνας εξόδου στο δίκτυο εφοδιάζεται με την επιθυμητή απόκρισή του, κάνοντας εύκολη υπόθεση τον υπολογισμό του αντίστοιχου σήματος λάθους. Στην περίπτωση (II) ο νευρώνας j είναι ένας κρυφός νευρώνας.

Αν και οι κρυφοί νευρώνες δεν είναι άμεσα προσπελάσιμοι, μοιράζονται ευθύνη για κάθε λάθος που συμβαίνει στην έξοδο του δικτύου. Το ζήτημα όμως είναι να γνωρίζουμε πώς να επιβάλλουμε ποινή (penalize) ή να επιβραβεύσουμε (reward) κρυφούς νευρώνες για το μερίδιο της ευθύνης τους. Αυτό συνιστά το πρόβλημα της επιβράβευσης (Credit-Assignment) (Haykin, 1999). Όπως θα δούμε στη συνέχεια, λύνεται με έναν κομψό τρόπο, με την πίσω διάδοση των σημάτων λάθους στο δίκτυο.

Στη συνέχεια θεωρούμε τις περιπτώσεις I και II.

Περίπτωση I: Ο νευρώνας j είναι ένας κόμβος εξόδου.

Όταν ο νευρώνας j βρίσκεται στο επίπεδο εξόδου του δικτύου, τροφοδοτείται με την επιθυμητή του έξοδο. Επομένως, μπορούμε να χρησιμοποιήσουμε την εξίσωση (3.10) για να υπολογίσουμε το σήμα λάθους $e_j(n)$ που σχετίζεται με τον νευρώνα. Αφού έχουμε καθορίσει το σήμα λάθους, ακολουθεί μια απλή διαδικασία προκειμένου να υπολογίσουμε την τοπική κλίση $\delta_j(n)$ χρησιμοποιώντας την εξίσωση (3.23).

Περίπτωση II: Ο νευρώνας j είναι ένας κρυφός κόμβος.

Όταν ένας νευρώνας j βρίσκεται σε ένα κρυφό επίπεδο του δικτύου, δεν υπάρχει κάποια καθορισμένη επιθυμητή απόκριση γι' αυτόν. Σε αναλογία το σήμα λάθους για έναν κρυφό νευρώνα θα έπρεπε να καθοριστεί επαναληπτικά σε όρους των σημάτων λάθους από όλους τους νευρώνες με τους οποίους αυτός συνδέεται άμεσα. Σε αυτό το σημείο είναι που περιπλέκεται η ανάπτυξη του αλγορίθμου πίσω διάδοσης. Θεωρούμε την περίπτωση, όπου ο νευρώνας j είναι ο κρυφός νευρώνας του δικτύου που συνδέεται με τον νευρώνα k (με τον 2ο προς την πλευρά της εξόδου).

Σύμφωνα με την εξίσωση (3.23) μπορούμε να ορίσουμε πάλι την τοπική κλίση $\delta_j(n)$ για κρυφό νευρώνα j ως εξής:

$$\delta_j(n) = \frac{\partial E(n)}{\partial y_j(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} = -\frac{\partial E(n)}{\partial y_j(n)} \cdot \varphi'_j(v_j(n)) \quad (3.24)$$

όπου ο j είναι κρυφός νευρώνας.

Για να υπολογίσουμε τη μερική παράγωγο $\frac{\partial E(n)}{\partial y_j(n)}$ μπορούμε να προχωρήσουμε ως εξής:

Το στιγμιαίο άθροισμα των τετραγώνων του λάθους στην επανάληψη n δίνεται από τη σχέση:

$$E(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n) \quad , \text{ όπου ο νευρώνας } k \text{ είναι ένας κόμβος εξόδου.} \quad (3.25)$$

Ας θυμηθούμε εδώ ότι η στιγμιαία τιμή του τετραγωνικού λάθους για τον νευρώνα k ορίζεται $1/2 e_k^2(n)$ και η $e(n)$ προκύπτει αθροίζοντας τα $1/2 e_k^2(n)$ για όλους τους νευρώνες εξόδου. Άρα από την εξίσωση (3.25) έχουμε:

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial y_j(n)} \quad (3.26)$$

Στη συνέχεια χρησιμοποιούμε τον αλυσιδωτό κανόνα για τον υπολογισμό της μερικής παράγωγου· δηλαδή η $\partial e_k(n) / \partial y_j(n)$ και η σχέση (3.26) ξαναγράφεται ως εξής:

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)} \quad (3.27)$$

Αλλά από την εξίσωση (3.10) έχουμε ότι:

$$\begin{aligned} e_k(n) &= d_k(n) - y_k(n) \quad \text{ή} \\ e_k(n) &= d_k(n) - \varphi_k(v_k(n)), \text{ αν ο νευρώνας } k \text{ είναι κόμβος εξόδου} \end{aligned} \quad (3.28)$$

$$\text{επομένως: } \frac{\partial e_k(n)}{\partial v_k(n)} = -\varphi'(v_k(n)) \quad (3.29)$$

Επίσης, για τον νευρώνα k το εσωτερικό επίπεδο ενεργοποίησης του δικτύου είναι:

$$v_k(n) = \sum_{j=0}^q w_{kj}(n) y_j(n) \quad (3.30)$$

όπου q είναι ο συνολικός αριθμός εισόδων (εξαιρούμε το κατώφλι) που εφαρμόζονται στον νευρώνα k . Και εδώ επίσης το $w^{k0}(n)$ ισούται με το $\theta^k(n)$ που εφαρμόζεται στον νευρώνα k και η αντίστοιχη είσοδος y^0 έχει σταθερή τιμή -1 .

$$\text{Από την εξίσωση (3.13): } \frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n) \quad (3.31)$$

Έτσι χρησιμοποιώντας τις εξισώσεις (3.29) και (3.31) στην (3.27) παίρνουμε την επιθυμητή μερική παράγωγο.

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_k(n) \varphi'_k(v_k(n)) w_{kj}(n) = \sum_k \delta_k(n) w_{kj}(n) \quad (3.32)$$

Εδώ έχουμε χρησιμοποιήσει τη σχέση $\delta_k(n) = e_k(n) \varphi'_k(v_k(n))$ (δηλαδή τη σχέση (3.23)), όπου ο k είναι νευρώνας εξόδου.

Τελικά χρησιμοποιώντας την εξίσωση (3.32) στην (3.24), παίρνουμε την τοπική κλίση $\delta_j(n)$ για έναν κρυφό νευρώνα j , αφού επαναδιατάξουμε τους όρους ως εξής:

$$\delta_j(n) = \varphi'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) \quad \text{όπου ο νευρώνας } j \text{ είναι κρυφός.} \quad (3.33)$$

Ο παράγοντας $\varphi'_j(v_j(n))$ στην (3.33) εξαρτάται αποκλειστικά από τη συνάρτηση ενεργοποίησης που σχετίζεται με τον κρυφό νευρώνα j .

Ο παράγοντας που απομένει στον υπολογισμό του $\delta_j(n)$, δηλαδή ο $\sum_k \delta_k(n) w_{kj}(n)$, —η πρόσθεση για όλα τα k — εξαρτάται από δύο σύνολα όρων:

- Το πρώτο σύνολο όρων, το $\delta_k(n)$, απαιτεί γνώση των σημάτων λάθους $e_k(n)$ για όλους εκείνους τους νευρώνες που βρίσκονται στο αμέσως δεξιό επίπεδο από τον κρυφό νευρώνα j , οι οποίοι είναι άμεσα συνδεδεμένοι με αυτόν.
- Το δεύτερο σύνολο όρων, το $w_{kj}(n)$, αποτελείται από τα συναπτικά βάρη που σχετίζονται με αυτές τις συνδέσεις.

Μπορούμε τώρα να συνοψίσουμε τις σχέσεις που έχουμε παραγάγει για τον αλγόριθμο Πίσω Διάδοσης.

Πρώτον, η διόρθωση $\Delta w_{ji}(n)$ που εφαρμόζεται στο συναπτικό βάρος, το οποίο συνδέει τον νευρώνα i στον νευρώνα j καθορίζεται από τον δέλτα κανόνα:

$$\left\{ \begin{array}{l} \text{Weight} \\ \text{correction} \\ \Delta w_{ji}(n) \end{array} \right\} = \left\{ \begin{array}{l} \text{Learning} \\ \text{parameter} \\ n \end{array} \right\} \cdot \left\{ \begin{array}{l} \text{Local} \\ \text{gradient} \\ \delta_j(n) \end{array} \right\} \cdot \left\{ \begin{array}{l} \text{Input signal} \\ \text{of neuron } j \\ y_i(n) \end{array} \right\} \quad (3.34)$$

$$\text{ή} \quad \Delta w_{ji}(n) = \eta \delta_j(n) \cdot y_i(n)$$

Δεύτερον, η τοπική κλίση $\delta_j(n)$ εξαρτάται από το εάν ο νευρώνας j είναι ένας κόμβος εξόδου ή ένας κρυφός κόμβος.

1. Εάν ο νευρώνας j είναι ένας κόμβος εξόδου, $\delta_j(n)$ ισούται με το γινόμενο της παραγωγού $\varphi'_j(v_j(n))$ και του σήματος λάθους $e_j(n)$ και τα δύο εκ των οποίων σχετίζονται με τον νευρώνα (βλέπε εξίσωση (3.24)).

$$\text{Δηλαδή αν ο } j \text{ είναι κόμβος εξόδου, τότε: } \delta_j(n) = e_j(n) \varphi'_j(v_j(n))$$

2. Εάν ο νευρώνας j είναι ένας κρυφός νευρώνας το $\delta_j(n)$ ισούται με το γινόμενο της σχετιζόμενης παραγωγού $\varphi'_j(v_j(n))$ και του ζυγισμένου αθροίσματος (weighted sum) των δ_k , που υπολογίζονται για τους νευρώνες, στο επόμενο κρυφό ή επίπεδο εξόδου, που είναι συνδεδεμένοι στον j (βλέπε εξίσωση (3.33)).

$$\text{Δηλαδή αν ο νευρώνας } j \text{ είναι κρυφός κόμβος, τότε:}$$

$$\delta_j(n) = \phi'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n)$$

3.3.3 Τα δύο περάσματα του υπολογισμού

Στην εφαρμογή του αλγόριθμου BP μπορούμε να διακρίνουμε δύο ξεχωριστά περάσματα του υπολογισμού. Το πρώτο περάσμα αναφέρεται ως προς τα εμπρός (forward pass) και το δεύτερο ως προς τα πίσω (backward pass).

Στο **forward pass** (προς τα εμπρός πέρασμα) τα συναπτικά βάρη παραμένουν αμετάβλητα μέσα στο δίκτυο και τα λειτουργικά σήματα του δικτύου υπολογίζονται σε μια νευρώνα προς νευρώνα βάση.

Συγκεκριμένα το λειτουργικό σήμα που εμφανίζεται στην έξοδο του νευρώνα j υπολογίζεται ως εξής:

$$y_j(n) = \phi(v_j(n)) \quad (3.35)$$

όπου $v_j(n)$ είναι το εσωτερικό επίπεδο ενεργοποίησης του νευρώνα j , που ορίζεται από τη σχέση:

$$v_j(n) = \sum_{i=1}^p w_{ji}(n) y_i(n) \quad (3.36)$$

όπου p είναι ο συνολικός αριθμός εισόδων (εξαιρουμένου του κατωφλίου ενεργοποίησης), οι οποίες εφαρμόζονται στον νευρώνα j , $w_{ji}(n)$ είναι το συναπτικό βάρος (της σύνδεσης) που συνδέει τον νευρώνα i στον νευρώνα j και $y_i(n)$ είναι το σήμα εξόδου του νευρώνα i ή ισοδύναμα το λειτουργικό σήμα που εμφανίζεται στην έξοδο του νευρώνα i .

Αν ο νευρώνας j βρίσκεται στο πρώτο κρυφό επίπεδο του δικτύου, τότε ο δείκτης i αναφέρεται στο i -οστό άκρο εισόδου του δικτύου, για το οποίο γράφουμε:

$$y_i(n) = x_i(n) \quad (3.37)$$

όπου $x_i(n)$ είναι το i -οστό στοιχείο του διανύσματος εισόδου (input pattern).

Αν, από την άλλη μεριά, ο j βρίσκεται στο επίπεδο εξόδου του δικτύου, ο δείκτης j αναφέρεται στο j -στό άκρο της εξόδου του δικτύου, για το οποίο γράφουμε:

$$y_j(n) = o_j(n) \quad (3.38)$$

όπου $o_j(n)$ είναι το j -οστό στοιχείο του διανύσματος εξόδου (pattern).

Αυτή η έξοδος συγκρίνεται με την επιθυμητή απόκριση $d_j(n)$, παρέχοντας το σήμα λάθους $e_j(n)$ για το j -στό νευρώνα. Έτσι η **προς τα εμπρός φάση** του υπολογισμού ξεκινά στο πρώτο κρυφό επίπεδο προσφέροντάς του το διάνυσμα εξόδου και τερματίζει στο επίπεδο εξόδου, υπολογίζοντας το σήμα λάθους για κάθε νευρώνα αυτού του επιπέδου. Το **προς τα πίσω πέρασμα**, από την άλλη, ξεκινά στο επίπεδο εξόδου, περνώντας τα σήματα λάθους προς τα αριστερά μέσω του δικτύου, επίπεδο προς επίπεδο και υπολογίζοντας το δ (δηλαδή την τοπική κλίση) επαναληπτικά, για κάθε νευρώνα. Αυτή η επαναληπτική διαδικασία επιτρέπει στα συναπτικά βάρη του δικτύου να υφίστανται αλλαγές (μεταβολές) σύμφωνα με τον κανόνα Δέλτα, δηλαδή την εξίσωση (3.34).

Για έναν νευρώνα τοποθετημένο στο επίπεδο εξόδου, το δ είναι απλά ίσο με το σήμα λάθους γι' αυτόν τον νευρώνα πολλαπλασιασμένο με την πρώτη παράγωγο της μη-γραμμικότητάς του. Επομένως, χρησιμοποιούμε την εξίσωση (3.34) για να υπολογίσουμε τις αλλαγές στα βάρη όλων των συνδέσεων που τροφοδοτούν το (καταλήγουν στο) επίπεδο εξόδου.

Αφού έχουμε υπολογίσει τα δ για τους νευρώνες του επιπέδου εξόδου, στη συνέχεια χρησιμοποιούμε την εξίσωση (3.35) για να υπολογίσουμε τα δ για όλους τους νευρώνες στο προτελευταίο επίπεδο και αντίστοιχα τις αλλαγές στα βάρη για όλες τις συνδέσεις που καταλήγουν σε αυτό. Ο επαναληπτικός υπολογισμός συνεχίζεται, επίπεδο προς επίπεδο, διαδίδοντας τις αλλαγές που έγιναν σε όλα τα συναπτικά βάρη.

Σημειώστε ότι για την παρουσίαση κάθε εκπαιδευτικού παραδείγματος (training example), το πρότυπο εισόδου είναι σταθερό καθ' όλη τη διάρκεια της διαδικασίας ταξιδιού μετ' επιστροφής, συμπεριλαμβάνοντας το προς τα εμπρός πέρασμα ακολουθούμενο από το προς τα πίσω πέρασμα.

3.3.4 Σιγμοειδής μη-γραμμικότητα

Ο υπολογισμός του δ για κάθε νευρώνα του perceptron πολλών επιπέδων απαιτεί τη γνώση της παραγώγου της συνάρτησης ενεργοποίησης του αντίστοιχου νευρώνα. Για να υπάρχει αυτή η παράγωγος απαιτείται η συνάρτηση να είναι συνεχής. Σε βασικές γραμμές η παραγωγισιμότητα είναι η μόνη συνθήκη που πρέπει να ικανοποιεί μια συνάρτηση ενεργοποίησης. Ένα παράδειγμα παραγωγίσιμων συνεχών μη γραμμικών συναρτήσεων ενεργοποίησης, οι οποίες χρησιμοποιούνται συχνά στα πολυεπίπεδα perceptrons, είναι οι σιγμοειδείς μη γραμμικές συναρτήσεις. Μια τέτοια συνάρτηση, η οποία έχει οριστεί εδώ για τον νευρώνα j , είναι η λογιστική συνάρτηση (logistic function):

$$y_j(n) = \varphi_j(v_j(n)) = \frac{1}{1+e^{(-v_j(n))}} \quad -\infty < v_j(n) < \infty \quad (3.39)$$

όπου $v_j(n)$ είναι η τιμή ενεργοποίησης του νευρώνα j . Εδώ η τιμή της εξόδου βρίσκεται στη περιοχή $0 < y_j < 1$. Μια άλλη τυπική σιγμοειδής συνάρτηση είναι η υπερβολική εφαπτομένη της οποίας οι τιμές εξόδου βρίσκονται στη περιοχή $-1 < y_j < 1$. Παραγωγίζοντας τις δύο πλευρές της εξίσωσης με το $v_j(n)$ έχουμε:

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \varphi'(v_j(n)) = \frac{e^{(-v_j(n))}}{[1+e^{(-v_j(n))}]^2} \quad (3.40)$$

Χρησιμοποιώντας την εξίσωση (39) για να απαλείψουμε τον ενθετικό όρο $\exp(-v_j(n))$ από τη σχέση (40), μπορούμε να εκφράσουμε την παράγωγο $\varphi'_j(v_j(n))$ ως:

$$\varphi'_j(v_j(n)) = y_j(n)[1 - y_j(n)] \quad (3.41)$$

Για κάθε νευρώνα j ο οποίος ανήκει στο επίπεδο εξόδου ισχύει ότι:

$$y_j(n) = o_j(n) \quad (3.42)$$

Επομένως μπορούμε να εκφράσουμε την τοπική κλίση για τον νευρώνα j ως :

$$\begin{aligned} \delta_j(n) &= e_j(n) \varphi'_j(v_j(n)) \\ &= [d_j(n) - o_j(n)] o_j(n) [1 - y_j(n)] \quad \text{ο νευρώνας } j \text{ ανήκει στο επίπεδο εξόδου} \end{aligned} \quad (3.43)$$

όπου $o_j(n)$ είναι το λειτουργικό σήμα στην έξοδο του νευρώνα j και $d_j(n)$ η επιθυμητή απόκριση· ενώ για έναν νευρώνα j ο οποίος βρίσκεται σε ένα κρυφό επίπεδο μπορούμε να εκφράσουμε την τοπική κλίση ως εξής:

$$\delta_j(n) = \varphi'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) = y_j(n) [1 - y_j(n)] \sum_k \delta_k(n) w_{kj}(n) \quad (3.44)$$

όταν ο νευρώνας j ανήκει στο κρυφό επίπεδο.

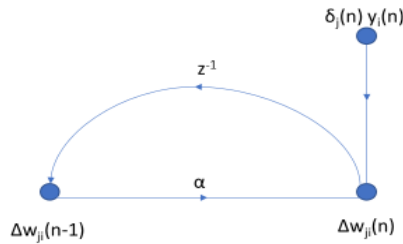
Σημειώστε ότι στην εξίσωση (3.41) η παράγωγος $\varphi'_j(n)$ έχει μέγιστη τιμή για $y_j(n)=0.5$ και την ελάχιστη (μηδέν) για $y_j(n)=1.0$. Αφού η αλλαγή των συναπτικών βαρών του δικτύου είναι ανάλογη της παραγώγου $\varphi'_j(v_j(n))$, συμπεραίνουμε ότι για μια σιγμοειδή συνάρτηση ενεργοποίησης τα συναπτικά βάρη θα τροποποιηθούν πιο πολύ γι' αυτούς τους νευρώνες, των οποίων τα λειτουργικά σήματα βρίσκονται στη μέση τιμή τους. Σύμφωνα με τον Rummelhatt (Haykin, 1999) είναι αυτή η ιδιότητα της μάθησης πίσω διάδοσης η οποία συνεισφέρει στη σταθερότητά του στον αλγόριθμο μάθησης.

3.3.5 Ρυθμός μάθησης

Ο αλγόριθμος πίσω διάδοσης δίνει μια προσέγγιση της φθίνουσας καμπύλης προς το ελάχιστο στο υπερεπίπεδο του χώρου των βαρών, χρησιμοποιώντας τη μέθοδο της απότομης μεταβολής. Όσο πιο μικρή είναι η παράμετρος μάθησης, τόσο πιο μικρές θα είναι οι αλλαγές στα βάρη του δικτύου σε κάθε επανάληψη του συνόλου εκπαίδευσης και τόσο πιο ομαλή θα είναι η φθίνουσα καμπύλη στο υπερεπίπεδο, δηλαδή έχουμε πιο σταθερή σύγκλιση. Όμως όλες αυτές οι βελτιώσεις έχουν για αντίτιμο πιο αργό ρυθμό μάθησης. Αν λοιπόν η παράμετρος μάθησης γίνει πολύ μεγάλη, έτσι ώστε να επιταχύνει τον ρυθμό μάθησης, οι μεγάλες αλλαγές που θα υπάρξουν στα βάρη μπορεί να δημιουργήσουν ένα ασταθές δίκτυο. Μια απλή μέθοδος για να αυξήσουμε τον ρυθμό μάθησης και ταυτόχρονα να αποφύγουμε τον κίνδυνο της αστάθειας είναι να τροποποιήσουμε τον δέλτα κανόνα της εξίσωσης (3.21) συμπεριλαμβάνοντας έναν όρο ορμής (momentum term):

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n) \quad (3.45)$$

όπου α είναι συνήθως ένας θετικός αριθμός που ονομάζεται σταθερά ορμής (momentum constant) και καθορίζει πόσο μεγάλη είναι η αλλαγή του βάρους στον επόμενο υπολογισμό, όπως φαίνεται στο σχήμα (3.9), όπου z^{-1} είναι ένα μοναδιαίο στοιχείο καθυστέρησης.



Σχήμα 3.9 Γραφική παράσταση που δείχνει την επίδραση της σταθεράς ορμής α .

Η εξίσωση (3.45) ονομάζεται **Γενικευμένος Δέλτα Κανόνας (Generalized Delta Rule)**, από όπου ο Δέλτα Κανόνας (Delta Rule) προκύπτει ως ειδική περίπτωση με $\alpha=0$.

Για να δούμε την επίδραση των ακολουθιών από παρουσιάσεις προτύπων πάνω στα βάρη μέσω της σταθεράς α , θα εκφράσουμε τη σχέση (3.45) ως χρονοσειρά με δείκτη το t . Ο δείκτης t παίρνει τιμές από 0 έως η . Τη σχέση (3.45) μπορούμε να τη δούμε ως διαφορική εξίσωση πρώτης τάξης του παράγοντα διόρθωσης βάρους $\Delta w_{jt}(n)$, άρα έχουμε:

$$\Delta w_{jt}(n) = -n \sum_{t=0}^n a^{n-t} \delta_j(t) y_t(t) \quad (3.46)$$

η οποία είναι μια σειρά μεγέθους $n+1$. Από τις σχέσεις (3.21) και (3.23) συμπεραίνουμε ότι το γινόμενο $\delta_j(n) y_i(n)$ ισούται με $-\partial E(n)/\partial w_{ji}(n)$. Επομένως, μπορούμε να ξαναγράψουμε τη σχέση ως εξής:

$$\Delta w_{jt}(n) = -n \sum_{t=0}^n a^{n-t} \frac{\partial E(t)}{\partial w_{jt}(t)} \quad (3.47)$$

Βασισμένοι σε αυτήν μπορούμε να κάνουμε τις εξής παρατηρήσεις:

1. Ο παράγοντας διόρθωσης βάρους $\Delta w_{ji}(n)$ αποτελείται από άθροισμα εκθετικών όρων. Για να συγκλίνει αυτή η σειρά πρέπει $0 < |a| < 1$. Όταν $a=0$, τότε ο αλγόριθμος πίσω διάδοσης δεν χρησιμοποιεί τη σταθερά ορμής. Η σταθερά a μπορεί να πάρει και αρνητικές τιμές, όμως είναι απίθανο να χρησιμοποιηθεί στην πράξη.
2. Όταν η μερική παράγωγος $\partial E(t)/\partial w_{ji}(t)$ έχει το ίδιο πρόσημο σε συνεχόμενες επαναλήψεις, το εκθετικό άθροισμα $\Delta w_{ji}(n)$ αυξάνει και έτσι το βάρος $w_{ji}(n)$ τροποποιείται κατά μεγάλο ποσό. Συνεπώς η συμπερίληψη της ορμής στον αλγόριθμο πίσω διάδοσης επιταχύνει σε περιπτώσεις όπου υπάρχουν σταθερές φθίνουσες κατευθύνσεις.
3. Όταν η μερική παράγωγος $\partial E(t)/\partial w_{ji}(t)$ έχει αντίθετο πρόσημο σε διαδοχικές επαναλήψεις, το εκθετικό άθροισμα $\Delta w_{ji}(n)$ ελαττώνεται και έτσι το βάρος $w_{ji}(n)$ τροποποιείται κατά ένα μικρό ποσό. Συνεπώς η συμπερίληψη της ορμής στον αλγόριθμο πίσω διάδοσης σταθεροποιεί σε περιπτώσεις όπου υπάρχουν ταλαντώσεις πρόσημων.

Η συμπερίληψη της ορμής στον αλγόριθμο πίσω διάδοσης αποτελεί μια μικρή αλλαγή όσον αφορά την τροποποίηση των βαρών, όμως έχει πολλές θετικές επιδράσεις στη συμπεριφορά μάθησης του αλγορίθμου. Επίσης, μπορεί να εμποδίσει τον τερματισμό της διαδικασίας σε ένα τοπικό ελάχιστο, το οποίο δεν είναι το ολικό ελάχιστο.

Πρόσθετες παρατηρήσεις.

Αναλύοντας τον αλγόριθμο πίσω διάδοσης υποθέσαμε ότι η παράμετρος μάθησης είναι σταθερά και την ονομάσαμε η . Στην πραγματικότητα θα έπρεπε να οριστεί ως η_i , δηλαδή η παράμετρος μάθησης θα έπρεπε να είναι διαφορετική ανάλογα με τη σύνδεση. Πράγματι πολλές ενδιαφέρουσες εξελίξεις θα μπορούσαν να προκύψουν κάνοντας την παράμετρο μάθησης διαφορετική για διάφορα μέρη του δικτύου.

Είναι ακόμα αξιοσημείωτο ότι στον αλγόριθμο πίσω διάδοσης μπορούμε να ορίσουμε όλα τα βάρη να τροποποιούνται ή μερικά από αυτά να παραμένουν σταθερά κατά τη διάρκεια της πίσω διάδοσης. Σε αυτήν την περίπτωση τα σήματα λάθους διαδίδονται πίσω στο δίκτυο, αλλά τα βάρη δεν τροποποιούνται. Αυτό μπορεί να επιτευχθεί θέτοντας την παράμετρο μάθησης η_i για το σημαντικό βάρος w_{ji} ίση με μηδέν.

Άλλο ένα ενδιαφέρον σημείο είναι ο τρόπος με το οποίο τα διάφορα επίπεδα του δικτύου είναι συνδεδεμένα. Στην ανάπτυξη του αλγορίθμου πίσω διάδοσης ο οποίος παρουσιάστηκε εδώ βασιστήκαμε στην υπόθεση ότι κάθε νευρώνας ενός επιπέδου δέχεται εισόδους από νευρώνες του προηγούμενου επιπέδου, όπως φαίνεται στο σχήμα 3.5. Όμως δεν υπάρχει λόγος ένας νευρώνας να μην δέχεται εισόδους από μονάδες άλλων, προηγούμενων επιπέδων. Στον χειρισμό ενός τέτοιου νευρώνα πρέπει να λάβουμε υπόψη δύο σήματα λάθους: (1) ένα σήμα λάθους το οποίο προκύπτει από τη σύγκρουση του σήματος εξόδου με την επιθυμητή απόκριση και (2) ένα σήμα λάθους το οποίο διαδίδεται μέσα από άλλες μονάδες στις οποίες επιδρά. Σε αυτήν την περίπτωση απλώς προσθέτουμε τις αλλαγές στα βάρη που πρέπει να γίνουν σύμφωνα με την απευθείας σύγκριση με όσες διαδίδονται πίσω από τις άλλες μονάδες.

3.3.6 Τρόποι εκπαίδευσης του δικτύου

Στον αλγόριθμο πίσω διάδοσης η μάθηση επιτυγχάνεται εφαρμόζοντας ένα σύνολο από διανύσματα εκπαίδευσης ως είσοδο στο πολυεπίπεδο perceptron. **Η προβολή όλων των διανυσμάτων εκπαίδευσης στο δίκτυο λέγεται «κύκλος» (epoch).** Η διαδικασία μάθησης προχωράει από epoch σε epoch μέχρι να σταθεροποιηθούν τα βάρη και τα κατώφλια του δικτύου και το μέσο τετραγωνικό λάθος όλων των διανυσμάτων εκπαίδευσης να τείνει σε κάποια ελάχιστη τιμή. Είναι καλή πρακτική να θέτουμε τα διανύσματα εκπαίδευσης σε μια τυχαία σειρά από το ένα epoch στο άλλο. Αυτή η τυχαιότητα τείνει να τρέψει το ψάξιμο στον χώρο των βαρών εντός κάθε κύκλου μάθησης σε στοχαστική διαδικασία, αποφεύγοντας έτσι τον κίνδυνο να γίνουν λιγότεροι κύκλοι από όσους πρέπει.

Για ένα δεδομένο σύνολο διάδοσης μπορούμε να ακολουθήσουμε έναν από τους εξής δύο τρόπους:

1. **Τρόπος Προτύπων (Pattern Mode).** Στο pattern mode η τροποποίηση των βαρών γίνεται με την προβολή κάθε διανύσματος του συνόλου εκπαίδευσης. Γι' αυτόν τον τρόπο λειτουργίας αναλύσαμε τον αλγόριθμο πίσω διάδοσης που παρουσιάσαμε εδώ. Συγκεκριμένα κάθε epoch αποτελείται από N διανύσματα (patterns) τοποθετημένα στη σειρά $[x(1), d(1), \dots, x(n), d(n)]$. Το πρώτο διάνυσμα $[x(1),$

$d(1)$] σε ένα epoch προβάλλεται στο δίκτυο και εκτελείται η ακολουθία των προς στα μπροστά (forward) και προς τα πίσω (backward) υπολογισμών που παρουσιάστηκαν στα προηγούμενα κεφάλαια, τροποποιώντας τα βάρη και τα κατώφλια του δικτύου. Όταν το δεύτερο διάνυσμα $[x(2), d(2)]$ σε ένα epoch προβάλλεται στο δίκτυο, τότε όλη η ακολουθία επαναλαμβάνεται και έχει για αποτέλεσμα την περαιτέρω τροποποίηση των βαρών και κατωφλίων. Αυτή η διαδικασία συνεχίζεται μέχρι να δοθεί το διάνυσμα $[x(n), d(n)]$. Το $\Delta w_{ji}(n)$ είναι η τροποποίηση του βάρους w_{ji} μετά την προβολή του διανύσματος n . Τότε η μέση αλλαγή του βάρους $\Delta \tilde{W}$ πάνω από όλα τα διανύσματα δίνεται από τη σχέση:

$$\begin{aligned}\Delta \tilde{W} &= \frac{1}{N} \sum_{n=1}^N \Delta w_{ji}(n) \\ &= -\frac{n}{N} \sum_{n=1}^N \frac{\partial E(n)}{\partial w_{ji}(n)} \\ &= -\frac{n}{N} \sum_{n=1}^N e_j(n) \frac{\partial e_j(n)}{\partial w_{ji}(n)}\end{aligned}\tag{3.48}$$

όπου στη δεύτερη και τρίτη γραμμή κάναμε χρήση των εξισώσεων (3.21) και (3.11).

2. **Σωρηδόν Τρόπος (Batch Mode).** Στο batch mode η τροποποίηση γίνεται μετά την προβολή όλων των διανυσμάτων εκπαίδευσης που αποτελούν ένα epoch στο δίκτυο. Για ένα epoch ορίζουμε τη συνάρτηση κόστους (cost function) ως το μέσο τετραγωνικό λάθος των σχέσεων (3.1) και (3.12), έχοντας πάλι αντικαταστήσει την (3.35) στην (3.36):

$$E_{av} = \frac{1}{2N} \sum_{n=1}^N \sum_{j \in C} e_j^2(n)\tag{3.49}$$

όπου το σήμα λάθους $e_j(n)$ αντιστοιχεί στην έξοδο του νευρώνα j για το διάνυσμα εκπαίδευσης n και ορίζεται από τη σχέση (3.10). Το $e_j(n)$ ισούται με τη διαφορά μεταξύ των τιμών $d_j(n)$ και $y_j(n)$, οι οποίες αναπαριστούν το j -οστό στοιχείο του επιθυμητού διανύσματος $d(n)$ και την αντίστοιχη τιμή του διανύσματος εξόδου. Στην (3.49) το εσωτερικό άθροισμα σε σχέση με το j συμπεριλαμβάνει όλα τα διανύσματα εκπαίδευσης σε ένα epoch. Έχοντας για παράμετρο μάθησης το η , η τροποποίηση στο βάρος w_{ji} , που συνδέει τον νευρώνα i με τον νευρώνα j , ορίζεται από τον δέλτα κανόνα και ισούται με:

$$\Delta w_{ji} = -\eta \frac{\partial E_{av}}{\partial w_{ji}} = -\frac{\eta}{N} \sum_{n=1}^N e_j(n) \frac{\partial e_j(n)}{\partial w_{ji}(n)}\tag{3.50}$$

Για να υπολογιστεί η μερική παράγωγος $\partial e_j(n)/\partial w_{ji}$ προχωράμε όπως προηγουμένως. Σύμφωνα με την (3.50) στο batch mode η τροποποίηση Δw_{ji} γίνεται μόνο αφού έχει γίνει προβολή όλων των διανυσμάτων εκπαίδευσης στο δίκτυο. Συγκρίνοντας τις (3.48) και (3.50) βλέπουμε καθαρά ότι ο μέσος όρος τροποποίησης βαρών $\Delta \tilde{W}$ στο pattern mode είναι διαφορετικός από την αντίστοιχη τιμή Δw_{ji} στο batch mode. Πράγματι το $\Delta \tilde{W}$ του pattern mode είναι μια εκτίμηση του Δw_{ji} του batch mode.

Από μια “on-line” (σε πραγματικό χρόνο) οπτική γωνία το pattern mode είναι προτιμότερο ενάντια στο batch mode, επειδή χρειάζεται λιγότερη τοπική μνήμη για κάθε σύναψη. Ακόμα, έχοντας ως δεδομένο ότι τα διανύσματα προβάλλονται στο δίκτυο με μια τυχαία σειρά, η χρήση της πρότυπο προς πρότυπο τροποποίησης, όπως γίνεται στο pattern mode, μετατρέπει το ψάξιμο στον χώρο των βαρών σε στοχαστική διαδικασία, η οποία δεν επιτρέπει εύκολα την παγίδευση του αλγορίθμου σε τοπικά ελάχιστα. Όμως η χρήση του batch mode δίνει μια πιο σωστή εκτίμηση του διανύσματος κλίσης. Σε τελική ανάλυση η αποδοτικότητα των δύο τρόπων εκπαίδευσης εξαρτάται από το πρόβλημα.

3.3.7 Κριτήρια τερματισμού

Ο αλγόριθμος πίσω διάδοσης σε γενικές γραμμές ούτε συγκλίνει ούτε υπάρχουν σαφώς ορισμένα κριτήρια για να σταματούν τη λειτουργία του, αλλά έχει λογικά κριτήρια που μπορούν να χρησιμοποιηθούν για να τερματίσουν τις ρυθμίσεις των βαρών. Για να διαμορφώσουμε ένα τέτοιο κριτήριο το λογικό είναι να σκεφτούμε σε σχέση με τις μοναδικές ιδιότητες του τοπικού ή ολικού ελαχίστου της επιφάνειας λάθους. Έστω ότι το διάνυσμα βαρών \mathbf{w}^* δηλώνει ένα ελάχιστο, τοπικό ή ολικό. Για να είναι το \mathbf{w}^* ένα ελάχιστο πρέπει το διάνυσμα κλίσης $\mathbf{g}(\mathbf{w})$ της επιφάνειας λάθους σε σχέση με το διάνυσμα βαρών \mathbf{w} να είναι μηδέν όταν $\mathbf{w} = \mathbf{w}^*$. Αντίστοιχα, μπορούμε να διατυπώσουμε ένα λογικό κριτήριο σύγκλισης για τη μάθηση πίσω διάδοσης, όπως παρακάτω (Kramer and Sangiovanni-Vincentelli, 1989 [Haykin, 1999]):

- Ο αλγόριθμος πίσω διάδοσης συγκλίνει όταν η ευκλείδεια νόρμα του διανύσματος κλίσης φτάσει σε ένα αρκετά μικρό κατώφλι κλίσης.

Το μειονέκτημα από αυτό το κριτήριο σύγκλισης είναι ότι ο χρόνος μάθησης μπορεί να είναι μεγάλος και χρειάζεται ο υπολογισμός του διανύσματος κλίσης $\mathbf{g}(\mathbf{w})$.

Άλλη μοναδική ιδιότητα ενός ελαχίστου είναι ότι η συνάρτηση κόστους ή μέτρο λάθους $\mathbf{E}_{av}(\mathbf{w})$ είναι στάσιμα στο σημείο $\mathbf{w}=\mathbf{w}^*$ και μπορούμε ως εκ τούτου να προτείνουμε ένα διαφορετικό κριτήριο σύγκλισης:

- Ο αλγόριθμος πίσω διάδοσης συγκλίνει όταν ο απόλυτος ρυθμός μεταβολής στο μέσο τετραγωνικό λάθος ανά κύκλο είναι αρκετά μικρός.

Τυπικά ο ρυθμός της μεταβολής στο μέσο τετραγωνικό λάθος θεωρείται αρκετά μικρός εάν βρίσκεται στο διάστημα 0.1 έως 1 % ανά κύκλο εκπαίδευσης, ενώ μερικές φορές χρησιμοποιείται μια τιμή που είναι αρκετή μικρή, έως 0.01 % ανά κύκλο.

Ένα άλλο κριτήριο σύγκλισης του αλγορίθμου, παραλλαγή του προηγούμενου, είναι να απαιτούμε η μέγιστη τιμή του μέσου τετραγωνικού λάθους $\mathbf{E}_{av}(\mathbf{w})$ να είναι ίση ή μικρότερη από ένα αρκετά μικρό κατώφλι. Οι Kramer και Sangiovanni-Vincentelli (1989) πρότειναν ένα υβριδικό κριτήριο σύγκλισης που συνίσταται σε αυτό το τελευταίο κατώφλι και ένα κατώφλι κλίσης, όπως δηλώνεται παρακάτω:

- Ο αλγόριθμος πίσω διάδοσης τερματίζεται στο διάνυσμα βαρών \mathbf{w}_{final} όταν $\|\mathbf{g}(\mathbf{w}_{final})\| \leq \epsilon$, όπου ϵ είναι ένα αρκετά μικρό κατώφλι κλίσης ή όταν $\mathbf{E}_{av}(\mathbf{w}_{final}) \leq \tau$, όπου τ είναι ένα αρκετά μικρό κατώφλι ενέργειας λάθους.

Άλλο χρήσιμο κριτήριο σύγκλισης είναι το εξής: μετά την κάθε επανάληψη μάθησης το δίκτυο δοκιμάζεται για τη γενικευτική του απόδοση· αν αυτή είναι αρκετή ή έχει κορυφωθεί, τότε σταματάμε τη διαδικασία μάθησης (βλ. και ενότητα 6.2).

3.3.8 Σύνοψη του αλγορίθμου Πίσω Διάδοσης

Έχουμε αναφέρει ότι η pattern by pattern μέθοδος ενημέρωσης των βαρών προτιμάται για on-line υλοποίηση του αλγορίθμου πίσω διάδοσης. Γι' αυτόν τον τρόπο λειτουργίας, ο αλγόριθμος κάνει κύκλους μέσω των δεδομένων εκπαίδευσης $\{[\mathbf{x}(n), \mathbf{d}(n)]; n = 1, 2, \dots, N\}$ όπως παρακάτω:

1. *Αρχικοποίηση.* Ξεκίνα με μία λογική διαμόρφωση του δικτύου και θέσε σε όλα τα συναπτικά βάρη και τα επίπεδα κατωφλίου του δικτύου μικρούς τυχαίους αριθμούς που είναι ομοιόμορφα κατανεμημένοι.
2. *Παρουσίαση Παραδειγμάτων Εκπαίδευσης.* Παρουσίασε στο δίκτυο έναν κύκλο από παραδείγματα εκπαίδευσης. Για κάθε παράδειγμα ταξινομημένο με κάποιο τρόπο μέσα στο σύνολο, εκτέλεσε την παρακάτω σειρά από μπρος και πίσω υπολογισμούς του 3 και 4 αντίστοιχα.
3. *Εμπρός Υπολογισμός.* Έστω ότι ένα παράδειγμα εκπαίδευσης του κύκλου δηλώνεται από το $[\mathbf{x}(n), \mathbf{d}(n)]$, με το διάνυσμα εισόδου $\mathbf{x}(n)$ να εφαρμόζεται στο επίπεδο εισόδου των αισθητήρων κόμβων και το επιθυμητό

διάνυσμα απόκρισης $\mathbf{d}(n)$ να παρουσιάζεται στο επίπεδο εξόδου των κόμβων υπολογισμού. Υπολόγισε τα επίπεδα ενεργοποίησης και τα λειτουργικά σήματα του δικτύου, προχωρώντας προς τα εμπρός μέσω του δικτύου, επίπεδο ανά επίπεδο. Το εσωτερικό επίπεδο ενεργοποίησης του δικτύου $\mathbf{u}^{(l)}$ για τον νευρώνα j στο επίπεδο l είναι:

$$u_j^{(l)}(n) = \sum_{i=0}^p w_{ji}^{(l)}(n) y_i^{(l-1)}$$

όπου $y_i^{(l-1)}(n)$ είναι το λειτουργικό σήμα του νευρώνα i στο προηγούμενο επίπεδο $l-1$ στην επανάληψη n και το $w_{ji}^{(l)}$ είναι το συναπτικό βάρος του νευρώνα j στο επίπεδο l που τροφοδοτείται από τον νευρώνα i στο επίπεδο $l-1$. Για $i = 0$ έχουμε $y_0^{(l-1)}(n) = -1$ και $w_{j0}^{(l)}(n) = \theta_j^{(l)}(n)$, όπου το $\theta_j^{(l)}$ είναι το κατώφλι που εφαρμόζεται στον νευρώνα j στο επίπεδο l . Υποθέτοντας τη χρησιμοποίηση μιας λογιστικής συνάρτησης για τη σιγμοειδή μη γραμμικότητα, το λειτουργικό σήμα (έξοδος) από τον νευρώνα j στο επίπεδο l είναι:

$$y_j^{(l)}(n) = \frac{1}{1 + e^{-u_j^{(l)}(n)}}$$

Αν ο νευρώνας j είναι στο πρώτο κρυμμένο επίπεδο (δηλαδή $l=1$), τότε:

$$y_j^{(0)}(n) = x_j(n)$$

όπου $x_j(n)$ είναι το j -στό στοιχείο του διανύσματος εισόδου $\mathbf{x}(n)$. Αν ο νευρώνας j είναι στο επίπεδο εξόδου, τότε:

$$y_j^{(L)}(n) = o_j(n)$$

Στη συνέχεια υπολογίζεται το σήμα λάθους:

$$e_j(n) = d_j(n) - o_j(n)$$

όπου $d_j(n)$ είναι το j -στό στοιχείο του επιθυμητού διανύσματος απόκρισης $\mathbf{d}(n)$.

4. Πίσω Υπολογισμός. Υπολόγισε τα δ (δηλαδή τις τοπικές κλίσεις) του δικτύου, προχωρώντας προς τα πίσω, επίπεδο ανά επίπεδο:

$$\delta_j^{(L)}(n) = e_j^{(L)}(n) o_j(n) [1 - o_j(n)] \text{ για τον νευρώνα } j \text{ στο επίπεδο εξόδου } L$$

$$\delta_j^{(l)}(n) = y_j^{(l)}(n) [1 - y_j^{(l)}(n)] \sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) \text{ για τον νευρώνα } j \text{ στο κρυφό επίπεδο } l$$

Έτσι ρύθμισε τα συναπτικά βάρη του δικτύου στο επίπεδο l με βάση τον γενικό κανόνα δέλτα:

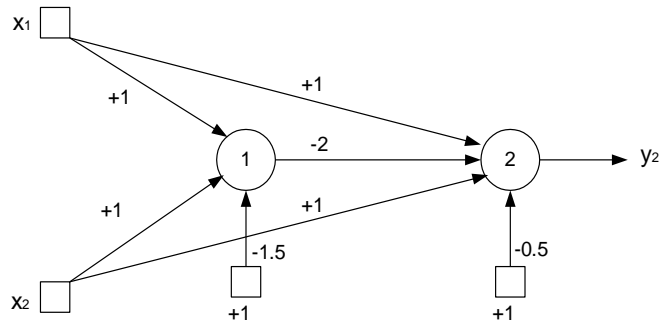
$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + a [w_{ji}^{(l)}(n) - w_{ji}^{(l)}(n-1)] + n \delta_j^{(l)}(n) y_i^{(l-1)}(n)$$

όπου η είναι ο ρυθμός μάθησης και a είναι η σταθερά ορμής.

5. *Επανάληψη*. Να επαναλάβεις τον υπολογισμό παρουσιάζοντας νέους κύκλους παραδειγμάτων εκπαίδευσης μέχρι που οι ελεύθερες παράμετροι του δικτύου σταθεροποιηθούν και το μέσο τετραγωνικό λάθος $E_{av}(w)$ που υπολογίσαμε πάνω στο συνολικό σύνολο εκπαίδευσης να είναι σε ένα ελάχιστο ή σε μια αποδεκτά μικρή τιμή. Η σειρά παρουσίασης των παραδειγμάτων εκπαίδευσης θα πρέπει να είναι τυχαία από κύκλο σε κύκλο. Καθώς αυξάνεται ο αριθμός των επαναλήψεων μάθησης, η ορμή και ο ρυθμός μάθησης μεταβάλλονται τυπικά (και συνήθως μειώνονται).

Άσκηση 3.3/4:

Στο παρακάτω σχήμα φαίνεται ένα νευρωνικό δίκτυο για την επίλυση του XOR προβλήματος. Δείξτε ότι όντως το δίκτυο επιλύει το XOR πρόβλημα κατασκευάζοντας: (α) τις περιοχές απόφασης και (β) έναν πίνακα αλήθειας του δικτύου.



Άσκηση 3.3/5:

Χρησιμοποιήστε τον Αλγόριθμο Πίσω Διάδοσης του Λάθους για να υπολογίσετε ένα σύνολο από πολώσεις (bias) και βάρη για ένα δίκτυο με δύο εισόδους, δύο κρυφούς νευρώνες και μία έξοδο, προκειμένου να επιλύει το XOR πρόβλημα. Για συνάρτηση ενεργοποίησης χρησιμοποιήστε τη λογιστική απεικόνιση.

Άσκηση 3.3/6:

Στην παράμετρο ορμής δίνουμε συνήθως θετικές τιμές στο διάστημα [0, 1). Ερευνήστε τι επίδραση θα είχε στη συμπεριφορά της εξίσωσης:

$$\Delta w_{jt}(n) = -n \sum_{t=0}^n a^{n-t} \frac{\partial E(t)}{\partial w_{jt}(t)}$$

η χρήση αρνητικών τιμών για το momentum a από το διάστημα (-1, 0].

Άσκηση 3.3/7:

Θεωρήστε ένα απλό παράδειγμα δικτύου με ένα βάρη και συνάρτηση κόστους:

$$E(w) = k_1(w-w_0)^2 + k_2$$

όπου τα w_0 , k_1 και k_2 είναι σταθερές. Για την ελαχιστοποίηση του $E(w)$ χρησιμοποιούμε τον αλγόριθμο Πίσω Διάδοσης του Λάθους με παράμετρο ορμής. Ερευνήστε πώς η χρήση ορμής επηρεάζει τη διαδικασία μάθησης με ιδιαίτερη αναφορά στον αριθμό των κύκλων (epochs) που απαιτούνται για σύγκλιση συναρτήσει του αλγορίθμου.

3.4 Σύνοψη κεφαλαίου

Για να χρησιμοποιηθεί ένα ΤΝΔ πρέπει πρώτα να εκπαιδευτεί για να μάθει. Η μάθηση συνίσταται στον προσδιορισμό των κατάλληλων συντελεστών βάρους, ώστε το ΤΝΔ να εκτελεί τους επιθυμητούς υπολογισμούς, και πραγματοποιείται με τη βοήθεια αλγορίθμων που είναι γνωστοί ως κανόνες μάθησης ή εκπαίδευσης. Ο ρόλος των συντελεστών βάρους μπορεί να ερμηνευτεί ως αποθήκευση γνώσης, η οποία παρέχεται μέσω παραδειγμάτων. Με αυτόν τον τρόπο τα ΝΔ μαθαίνουν το περιβάλλον τους, δηλαδή το φυσικό μοντέλο που παρέχει τα δεδομένα.

Σε αυτό το κεφάλαιο παρουσιάσαμε τους τρεις βασικούς αλγορίθμους μάθησης (εκπαίδευσης) ΝΔ. Αρχίσαμε την παρουσίαση με τον αλγόριθμο εκπαίδευσης του απλού Perceptron (Αισθητήρα) και το θεώρημα της σύγκλισής του. Μετά παρουσιάσαμε τον βασικό αλγόριθμο εκπαίδευσης για δίκτυα εμπρός τροφοδότησης πολλών επιπέδων, που είναι γνωστά ως Perceptrons πολλών επιπέδων.

Ο αλγόριθμος εκπαίδευσης αυτών των δικτύων είναι ο πολύ δημοφιλής αλγόριθμος Πίσω Διάδοσης (ΠΔ) του λάθους, ο οποίος βασίζεται στον υπολογισμό της κλίσης της συνάρτησης λάθους στην έξοδο του δικτύου και τον προς τα πίσω υπολογισμό των τοπικών κλίσεων. Κατά την παραγωγή του αλγορίθμου διακρίναμε δύο κατηγορίες νευρώνων, όσους είναι στο επίπεδο εξόδου και όσους είναι στα κρυφά επίπεδα. Αν και η παραγωγή του αλγορίθμου είναι αρκετά πολύπλοκη, ο ίδιος ο αλγόριθμος είναι εύκολο να υλοποιηθεί και έχει τύχει ευρείας εφαρμογής σε πολλά πρακτικά προβλήματα. Έγινε επίσης η παρουσίαση μιας τροποποίησης του αλγορίθμου ΠΔ, που είναι γνωστή ως Γενικευμένος Δέλτα κανόνας. Με αυτόν τον αλγόριθμο επιτυγχάνεται η αύξηση του ρυθμού μάθησης, ενώ εξασφαλίζεται η σύγκλιση του αλγορίθμου. Η ενότητα ολοκληρώθηκε με την παρουσίαση των δύο τρόπων υπολογισμού, δηλαδή την κατεργασία ανά πρότυπο και τη σωρηδόν κατεργασία. Για κάθε τρόπο λειτουργίας του αλγορίθμου είδαμε τα πλεονεκτήματα και τα μειονεκτήματά του. Το κεφάλαιο ολοκληρώθηκε με τη συζήτηση των κριτηρίων τερματισμού της εκτέλεσης του αλγορίθμου. Για περισσότερες λεπτομέρειες σχετικά με τα παραπάνω θέματα ο αναγνώστης παραπέμπεται στην αναφορά Haykin, 1999. Για πρακτική εξάσκηση στους αλγορίθμους εκπαίδευσης μπορεί να χρησιμοποιηθεί το δεύτερο βιβλίο το οποίο περιέχει εκτελέσιμο κώδικα για τους αλγορίθμους που παρουσιάσαμε στο παρόν.

Συνοψίζοντας, μπορούμε να πούμε ότι σκοπός αυτού του κεφαλαίου ήταν η παρουσίαση των βασικών αλγορίθμων εκπαίδευσης τόσο απλών όσο και πολυεπίπεδων ΤΝΔ. Εδώ πρέπει να αναφέρουμε ότι η τιμή της παραμέτρου μάθησης έχει ιδιαίτερη σημασία για τη σύγκλιση του αλγορίθμου. Επίσης, ο αλγόριθμος ΠΔ του λάθους απαιτεί να είναι εκ των προτέρων γνωστή η τοπολογία του δικτύου. Αυτό όμως αποτελεί ένα βασικό πρόβλημα των ΤΝΔ, για το οποίο δεν έχει ακόμη δοθεί μια γενική λύση. Τα συγκεκριμένα θέματα θα τα συζητήσουμε αναλυτικά στο επόμενο κεφάλαιο στο πλαίσιο μίας εφαρμογής.

Βιβλιογραφία

Hagan, M.T., Demuth, H.B., & Beal, M. (1996). *Neural Network Design*. PWS Publishing Company.

Haykin, S. (1999). *Neural Networks, A Comprehensive Foundation* (2η έκδ.). Prentice Hall International.

Απαντήσεις ασκήσεων

Άσκηση 3.2/1:

Γραμμικά διαχωριζόμενα είναι τα πρότυπα

1. για τα οποία μπορούμε εύκολα να ορίσουμε περιοχές απόφασης,
2. για τα οποία οι περιοχές απόφασης είναι γραμμικές,
3. που βρίσκονται στις αντίθετες πλευρές ενός υπερεπιπέδου που ορίζει τις περιοχές απόφασης,
4. τα οποία χρησιμοποιούνται για την εκπαίδευση του Perceptron.

Απάντηση: Σύμφωνα με τον ορισμό που δίνεται στην ενότητα 3.2 η σωστή απάντηση είναι η 3.

Άσκηση 3.2/2:

Να υποθέσετε ότι στο διάγραμμα ροής σήματος του Perceptron του σχήματος 4 η συνάρτηση ενεργοποίησης έχει τη μορφή:

$$\varphi(v) = \tanh\left(\frac{v}{2}\right)$$

όπου v είναι η γραμμική έξοδος του νευρώνα. Οι αποφάσεις ταξινόμησης από το Perceptron καθορίζονται από τον ακόλουθο κανόνα:

Το διάνυσμα παρατήρησης x ανήκει στην κλάση 11 αν $y > \theta$.
Διαφορετικά το x ανήκει στην κλάση 12.

$$y = \tanh\left(\frac{v}{2}\right) = \tanh\left(\frac{\theta}{2} + \frac{1}{2} \sum_i w_i x_i\right)$$

Απάντηση: Το σήμα εξόδου υπολογίζεται από τη σχέση
Ισοδύναμα μπορούμε να γράψουμε:

$$\theta + \sum_i w_i x_i = y' \quad (1)$$

όπου :

$$y' = 2 \tanh^{-1}(y)$$

Η εξίσωση (1) είναι η εξίσωση ενός υπερεπιπέδου.

Άσκηση 3.2/3:

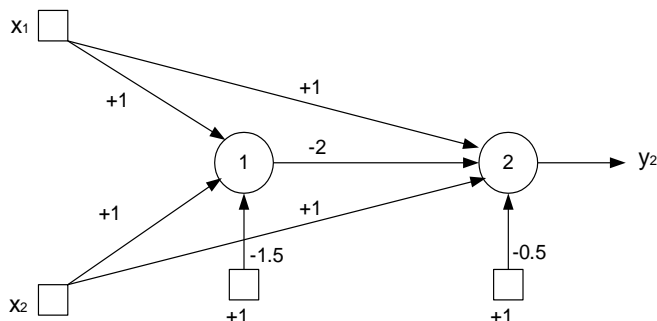
- (α) Το Perceptron μπορεί να χρησιμοποιηθεί για να μάθει διάφορες λογικές συναρτήσεις. Να δείξετε την υλοποίηση των δυαδικών λογικών συναρτήσεων AND, OR και COMPLEMENT.
- (β) Ένα βασικό μειονέκτημα του Perceptron είναι ότι δεν μπορεί να υλοποιήσει τη συνάρτηση EXCLUSIVE OR. Να εξηγήσετε τον λόγο για αυτόν τον περιορισμό.

Απάντηση:

- (α) Να κατασκευάσετε τον πίνακα αλήθειας για καθεμία συνάρτηση. Στη συνέχεια να υλοποιήσετε ένα Perceptron δύο εισόδων με μοναδιαία βάρη και κατώφλια -1.5 , -0.5 και $+0.5$ αντίστοιχα. Να επαληθεύσετε ότι ικανοποιούνται οι αντίστοιχοι πίνακες.
- (β) Από τον πίνακα αλήθειας της συνάρτησης προκύπτει εύκολα ότι δεν μπορούμε να κατασκευάσουμε ένα γραμμικό όριο απόφασης (όπως στο σχήμα 3).

Άσκηση 3.3/4:

Στο παρακάτω σχήμα φαίνεται ένα νευρωνικό δίκτυο για την επίλυση του XOR προβλήματος. Δείξτε ότι όντως το δίκτυο επιλύει το XOR πρόβλημα κατασκευάζοντας: (α) τις περιοχές απόφασης και (β) έναν πίνακα αλήθειας του δικτύου.



Απάντηση:

Υποθέτουμε ότι κάθε νευρώνας ακολουθεί το μοντέλο McCulloch – Pitts.

(α) Για να κατασκευάσουμε τις περιοχές απόφασης εργαζόμαστε όπως στην παράγραφο που περιγράφει το XOR πρόβλημα. Έτσι κατασκευάζουμε πρώτα τις περιοχές απόφασης που σχηματίζει ο νευρώνας 1 και στη συνέχεια τις περιοχές που σχηματίζει ο νευρώνας 2, οι οποίες είναι και οι περιοχές απόφασης του δικτύου.

(β) Για τον νευρώνα 1 έχουμε:

$$u_1 = x_1 + x_2 - 1.5$$

Έτσι μπορούμε να κατασκευάσουμε τον ακόλουθο πίνακα:

x1	0	0	1	1
x2	0	1	0	1
u1	-1.5	-0.5	-0.5	0.5
y1	0	0	0	1

Για τον νευρώνα 2 έχουμε:

$$u_2 = x_1 + x_2 - 2 \cdot y_1 - 0.5$$

x1	0	0	1	1
x2	0	1	0	1
y1	0	0	0	1
u2	-0.5	0.5	0.5	-0.5
y2	0	1	1	0

Από τον πίνακα αυτόν παρατηρούμε ότι η έξοδος του δικτύου y_2 είναι 0 εάν x_1 και x_2 είναι 0 ή 1, και 1 εάν κάποιο από τα δύο είναι 0 και το άλλο 1. Άρα το δίκτυο έχει μάθει τη συνάρτηση XOR.

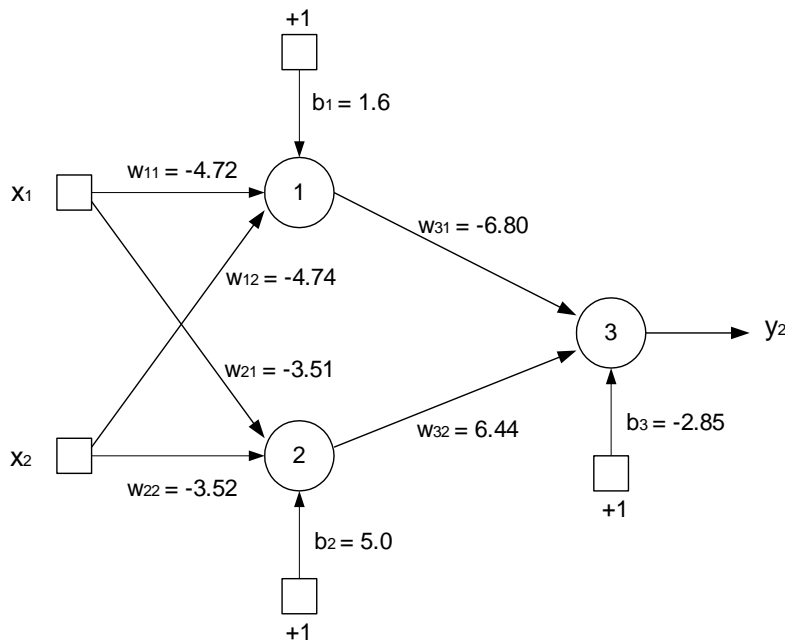
Άσκηση 3.3/5:

Χρησιμοποιήστε τον Αλγόριθμο Πίσω Διάδοσης του Λάθους για να υπολογίσετε ένα σύνολο από πολώσεις (bias) και βάρη για ένα δίκτυο με δύο εισόδους, δύο κρυφούς νευρώνες και μία έξοδο, προκειμένου να επιλύει το XOR πρόβλημα. Για συνάρτηση ενεργοποίησης χρησιμοποιήστε τη λογιστική απεικόνιση.

Απάντηση:

Επειδή γενικά η λογιστική συνάρτηση παίρνει τιμές από 0.1 έως 0.9 ως επιθυμητή απόκριση του δικτύου, θα χρησιμοποιήσουμε την 0.1 για την περίπτωση του 0 και την 0.9 για την περίπτωση του 1. Ως αρχικές τιμές στα βάρη και στις πολώσεις δίνουμε τιμές στο διάστημα (-1.0, 1.0).

Τρέχοντας τον Αλγόριθμο Πίσω Διάδοσης του Λάθους για 2500 επαναλήψεις προκύπτει το παρακάτω δίκτυο που έχει μάθει τη συνάρτηση XOR. Εδώ θα πρέπει να σημειώσουμε ότι οι τιμές των ελευθέρων παραμέτρων (βάρη και πολώσεις) του δικτύου που προέκυψε είναι ενδεικτικές και δεν σημαίνει πως οι δικές σας θα είναι αναγκαία ίδιες με αυτές.



Άσκηση 3.3/6:

Στην παράμετρο ορμής δίνουμε συνήθως θετικές τιμές στο διάστημα [0, 1). Ερευνήστε τι επίδραση θα είχε στη συμπεριφορά της εξίσωσης

$$\Delta w_{ji}(n) = -\eta \sum_{t=0}^n a^{n-t} \frac{\partial E(t)}{\partial w_{ji}(t)}$$

η χρήση αρνητικών τιμών για το momentum a από το διάστημα (-1, 0].

Απάντηση:

Αν το momentum a πάρει αρνητικές τιμές, θα έχουμε:

$$\Delta w_{ji}(n) = -\eta \sum_{t=0}^n a^{n-t} \cdot \frac{\partial E(t)}{\partial w_{ji}(t)}$$

$$-\eta \sum_{t=0}^n (-1)^{n-t} \cdot |a|^{n-t} \cdot \frac{\partial E(t)}{\partial w_{ji}(t)}$$

Από την παραπάνω εξίσωση μπορούμε να δούμε ότι αν το $\frac{\partial E(t)}{\partial w_{ji}(t)}$ έχει το ίδιο πρόσημο σε συνεχόμενες επαναλήψεις του αλγορίθμου, το μέγεθος του $\Delta w_{ji}(n)$ ελαττώνεται. Το αντίθετο ισχύει εάν το $\frac{\partial E(t)}{\partial w_{ji}(t)}$ αλλάζει πρόσημο σε διαδοχικές επαναλήψεις του αλγορίθμου. Άρα η επίδραση του momentum α είναι η αντίθετη από αυτήν που έχει όταν παίρνει θετικές τιμές.

Άσκηση 3.3/7:

Θεωρήστε ένα απλό παράδειγμα δικτύου με ένα βάρος και συνάρτηση κόστους:

$$E(w) = k_1(w-w_0)^2 + k_2$$

όπου τα w_0 , k_1 και k_2 είναι σταθερές. Για την ελαχιστοποίηση του $E(w)$ χρησιμοποιούμε τον αλγόριθμο Πίσω Διάδοσης του Λάθους με παράμετρο ορμής. Ερευνήστε πώς η χρήση ορμής επηρεάζει τη διαδικασία μάθησης, με ιδιαίτερη αναφορά στον αριθμό των κύκλων (epochs) που απαιτούνται για σύγκλιση συναρτήσε του αλγορίθμου.

Απάντηση:

Από τη θεωρία γνωρίζουμε ότι:

$$\Delta w_{ji}(n) = -\eta \sum_{t=1}^n a^{n-t} \frac{\partial E(t)}{\partial w_{ji}(t)} \quad (1)$$

Στην περίπτωση που έχουμε ένα μόνο βάρος η συνάρτηση κόστους είναι:

$$E(w) = k_1(w-w_0)^2 + k_2 \quad (2)$$

Οπότε, η εξίσωση (1) λόγω της (2) γίνεται:

$$\Delta w(n) = -2 \cdot k_1 \cdot \eta \sum_{t=1}^n a^{n-t} \cdot (w(t) - w_0) \quad (3)$$

Στην περίπτωση αυτή η μερική παράγωγος $\frac{\partial E(t)}{\partial w(t)}$ έχει το ίδιο πρόσημο σε διαδοχικές επαναλήψεις. Επομένως, για $0 \leq \alpha < 1$ το $\Delta w(n)$ αυξάνει σε μέγεθος. Αυτό σημαίνει ότι το βάρος διορθώνεται με μεγάλες ποσότητες. Άρα η χρήση του momentum α στην εξίσωση έχει ως συνέπεια την επιτάχυνση της «καθόδου» προς το βέλτιστο.

Κεφάλαιο 4

Αρχές και Περιορισμοί Σχεδιασμού Τεχνητών Νευρωνικών Δικτύων

Σύνοψη

Στο τρίτο κεφάλαιο είδαμε τους δύο βασικούς αλγόριθμους εκπαίδευσης Τεχνητών Νευρωνικών Δικτύων. Συγκεκριμένα, πρώτα είδαμε δύο αλγόριθμους για εκπαίδευση απλών ΝΔ ενός επιπέδου, δηλαδή τον κανόνα του perceptron. Για την περίπτωση των ΝΔ πολλών επιπέδων παρουσιάσαμε τον αλγόριθμο Πίσω Διάδοσης του λάθους και μια γενίκευσή του, τον Γενικευμένο Δέλτα κανόνα. Το κύριο βάρος στην παρουσίαση αυτών των αλγορίθμων δόθηκε στην παραγωγή τους και σε άλλα θεωρητικά θέματα. Οι παραπάνω αλγόριθμοι εργάζονται για προκαθορισμένη διαμόρφωση (τοπολογία) του δικτύου.

Σε αυτό το κεφάλαιο θα ασχοληθούμε με θέματα που αφορούν την υλοποίηση ΤΝΔ πολλών επιπέδων. Θα συζητήσουμε πρώτα το πρόβλημα της αρχικοποίησης, δηλαδή μερικούς τρόπους αποτελεσματικότερης εκτέλεσης των αλγορίθμων. Όμως το βασικό πρόβλημα στην υλοποίηση ενός ΝΔ για την επίλυση ενός πραγματικού προβλήματος είναι ότι δεν είναι εκ των προτέρων γνωστή η διαμόρφωση του δικτύου, δηλαδή δεν γνωρίζουμε τον ακριβή αριθμό κρυφών επιπέδων ούτε και τον αριθμό των κόμβων ανά κρυφό επίπεδο. Δεν υπάρχει κάποια γενική θεωρία που επιλύει αυτό το πρόβλημα. Σε αυτό το κεφάλαιο θα προσπαθήσουμε να δώσουμε μερικές ιδέες για το πώς αντιμετωπίζεται ο καθορισμός του βέλτιστου δικτύου, τόσο με τη βοήθεια προσομοίωσης σε ηλεκτρονικό υπολογιστή, όσο και με τη βοήθεια κάποιων θεωρητικών αποτελεσμάτων από τη βιβλιογραφία.

Συνοψίζοντας αναφέρουμε ότι σκοπός του συγκεκριμένου κεφαλαίου είναι να βοηθήσει τον αναγνώστη στη σχεδίαση και υλοποίηση ΤΝΔ, δίνοντάς του ιδέες για τον τρόπο αντιμετώπισης των πρακτικών προβλημάτων που προκύπτουν.

Προσδοκώμενα αποτελέσματα:

Όταν θα έχετε τελειώσει τη μελέτη αυτού του κεφαλαίου, θα μπορείτε να:

- σχεδιάσετε ένα ΤΝΔ κατάλληλο για την επίλυση ενός πρακτικού προβλήματος,
- υλοποιήσετε ένα τέτοιο δίκτυο,
- αντιμετωπίσετε τα επιμέρους προβλήματα που παρουσιάζονται,
- βελτιστοποιήσετε τη δομή ενός δικτύου.

Λέξεις-Κλειδιά:

- Αρχικοποίηση
- Αναπαράσταση εξόδου
- Βέλτιστη δομή
- Γενίκευση

Προαπαιτούμενη γνώση

Κεφάλαια 1, 2, 3.

4.1 Εισαγωγικές Παρατηρήσεις

Το βασικό πλεονέκτημα των perceptrons πολλών επιπέδων είναι ότι μπορούν να διαχωρίσουν μη-γραμμικά διαχωριζόμενα πρότυπα. Θεωρητικά λοιπόν μπορούν να αντιμετωπίσουν οποιοδήποτε πρόβλημα ταξινόμησης προτύπων. Δυστυχώς όμως στην πράξη τα πράγματα δεν είναι τόσο απλά. Όπως ήδη αναφέραμε, το βασικό μειονέκτημα ενός τέτοιου δικτύου είναι ο καθορισμός της βέλτιστης και καταλληλότερης δομής για την επίλυση ενός συγκεκριμένου προβλήματος. Έχοντας επιλέξει σωστά το εκπαιδευτικό σύνολο, αυτόματα έχουμε καθορίσει τον αριθμό των εισόδων και των εξόδων του δικτύου. Δεν έχουμε ωστόσο καμία εκ των προτέρων γνώση για την εσωτερική αρχιτεκτονική του δικτύου (αριθμός κρυφών επιπέδων και αριθμός κόμβων ανά κρυφό επίπεδο). Ένας προφανής τρόπος για να αντιμετωπίσουμε αυτό το πρόβλημα είναι με τη μέθοδο δοκιμής και λάθους (trial and error). Με βάση κάποιους εμπειρικούς κανόνες υλοποιούμε μια συγκεκριμένη αρχιτεκτονική δικτύου και το εκπαιδεύουμε, ελπίζοντας σε καλή γενίκευση. Αν, μετά την εκπαίδευση, το λάθος στην έξοδο του δικτύου είναι σχετικά μεγάλο, τροποποιούμε τη δομή του (προσθέτοντας ή αφαιρώντας κρυφά επίπεδα ή/και κόμβους) και το εκπαιδεύουμε πάλι. Έτσι, με διαδοχικές δοκιμές, ελπίζουμε να επιτύχουμε το

(σχεδόν) βέλτιστο δίκτυο. Είναι προφανές ότι αυτή η διαδικασία είναι χρονοβόρα και επίπονη. Γι' αυτόν τον λόγο λέγεται ότι μαθαίνει κανείς μαζί με το δίκτυο.

Τα τελευταία χρόνια έχουν εμφανιστεί στη βιβλιογραφία πιο έξυπνες μέθοδοι για την επίλυση αυτού του προβλήματος. Τέτοιες είναι η μέθοδος του κλαδέματος (pruning) (Haykin, 1999), της ανάπτυξης (growing) (Haykin, 1999) και της εξέλιξης ΝΔ (Goldberg, 1989 και Michalewicz, 1992). Η παρουσίαση αυτών των αλγορίθμων βρίσκεται πέρα από τους σκοπούς του συγκεκριμένου μαθήματος και ο αναγνώστης παραπέμπεται στη σχετική βιβλιογραφία.

Στις ενότητες που ακολουθούν θα παρουσιάσουμε τις πιο γνωστές ευριστικές μεθόδους με τις οποίες μπορεί κανείς να αντιμετωπίσει τα παραπάνω προβλήματα.

4.2 Αρχικοποίηση

Το πρώτο βήμα της μάθησης πίσω διάδοσης είναι να αρχικοποιήσουμε το δίκτυο. Μια καλή επιλογή για τις αρχικές τιμές των ελεύθερων παραμέτρων του δικτύου θα βοηθήσει στην επιτυχία του σχεδιασμού και της λειτουργίας του. Σε περιπτώσεις που έχουμε προηγούμενες πληροφορίες θα ήταν καλύτερα να τις χρησιμοποιήσουμε για να μαντεύσουμε τις αρχικές τιμές των ελεύθερων παραμέτρων. Πώς όμως μπορούμε να αρχικοποιήσουμε το δίκτυο αν δεν έχουμε προηγούμενη πληροφορία; Η συνηθισμένη πρακτική είναι να θέσουμε στις ελεύθερους παραμέτρους του δικτύου τυχαίους αριθμούς που είναι ομοιόμορφα κατανομημένοι μέσα σε μία μικρή περιοχή τιμών (π.χ. -0.1 μέχρι $+0.1$).

Η λανθασμένη επιλογή των αρχικών βαρών μπορεί να οδηγήσει σε ένα φαινόμενο γνωστό ως «πρόωρο κορεσμό» (Lee et al., 1991·Haykin, 1999), το οποίο αναφέρεται σε μία κατάσταση όπου το στιγμιαίο άθροισμα των τετραγωνικών λαθών $E(\mathbf{n})$ παραμένει σχεδόν σταθερό για μία περίοδο χρόνου κατά τη διάρκεια της μάθησης. Ένα τέτοιο φαινόμενο δεν μπορεί να θεωρηθεί τοπικό ελάχιστο, γιατί το τετραγωνικό λάθος συνεχίζει να μειώνεται στο τέλος αυτής της περιόδου.

Όταν εφαρμόζεται ένα δείγμα μάθησης στο επίπεδο εισόδου ενός πολυεπίπεδου perceptron, οι τιμές εξόδου του δικτύου υπολογίζονται μέσω μίας σειράς από εμπρός υπολογισμούς που περιλαμβάνει εσωτερικά γινόμενα και σιγμοειδείς μετατροπές. Αυτή ακολουθείται από μία σειρά από πίσω υπολογισμούς των σημάτων λάθους και της σχετικής κλίσης της σιγμοειδούς συνάρτησης ενεργοποίησης και καταλήγει στις ρυθμίσεις των συναπτικών βαρών. Ας υποθέσουμε ότι για ένα συγκεκριμένο δείγμα μάθησης υπολογίστηκε το εσωτερικό επίπεδο ενεργοποίησης του δικτύου για έναν νευρώνα εξόδου να έχει μεγάλο μέγεθος. Τότε, θεωρώντας ότι η σιγμοειδής συνάρτηση ενεργοποίησης του νευρώνα έχει τις περιορισμένες τιμές -1 και $+1$, βρίσκουμε ότι η αντίστοιχη κλίση της συνάρτησης ενεργοποίησης γι' αυτόν θα είναι πολύ μικρή και η τιμή εξόδου του θα πλησιάζει το -1 ή $+1$. Σε αυτήν την περίπτωση λέμε ότι ο νευρώνας είναι σε κόρο. Αν η τιμή εξόδου πλησιάζει το $+1$ όταν η επιθυμητή απόκριση είναι -1 ή αντίστροφα, τότε λέμε ότι ο νευρώνας είναι κατά λάθος κορεσμένος. Όταν συμβαίνει αυτό, οι μεταβολές που γίνονται πάνω στα συναπτικά βάρη του νευρώνα θα είναι μικρές (ας είναι μεγάλο το μέγεθος του σχετικού σήματος λάθους) και το δίκτυο θα αργήσει να ξεφύγει από αυτήν την περίπτωση (Lee et al., 1991·Haykin, 1999).

Στην αρχική φάση της πίσω διάδοσης μάθησης, ανάλογα με τις επικρατούσες συνθήκες, νευρώνες που είναι ή δεν είναι κορεσμένοι μπορούν να υπάρχουν στο επίπεδο εξόδου του δικτύου. Καθώς η διαδικασία μάθησης συνεχίζεται, τα συναπτικά βάρη που σχετίζονται με τους μη κορεσμένους νευρώνες εξόδου αλλάζουν γρήγορα, γιατί τα αντίστοιχα σήματα λάθους και κλίσης έχουν σχετικά μεγάλο μέγεθος· γι' αυτόν τον λόγο προκαλούν μία ελάττωση του στιγμιαίου αθροίσματος των τετραγωνικών λαθών $E(\mathbf{n})$. Αν ωστόσο σε αυτό το σημείο οι κατά λάθος κορεσμένοι νευρώνες εξόδου μείνουν κορεσμένοι για μερικά δείγματα μάθησης, τότε το φαινόμενο του πρόωρου κορεσμού μπορεί να εμφανιστεί με το $E(\mathbf{n})$ να παραμένει σταθερό.

Στην αναφορά Lee et al. (1991) μία φόρμουλα για την πιθανότητα του πρόωρου κορεσμού στη μάθηση πίσω διάδοσης έχει παραχθεί για τον σωρηδόν τρόπο ενημέρωσης και έχει επαληθευτεί χρησιμοποιώντας εξομοίωση με υπολογιστή. Η ουσία αυτού του τύπου συνοψίζεται παρακάτω:

1. Αν επιλέξουμε τις αρχικές τιμές για τα συναπτικά βάρη και για τα επίπεδα κατωφλίων του δικτύου να είναι ομοιόμορφα κατανομημένες μέσα σε μία μικρή περιοχή τιμών, τότε αποφεύγεται ο κατά λάθος κορεσμός.
2. Ο κατά λάθος κορεσμός έχει μικρότερη πιθανότητα να συμβεί όταν έχουμε μικρό αριθμό από κρυμμένους νευρώνες και ικανοποιητική λειτουργία του δικτύου.

3. Ο κατά λάθος κορεσμός σπάνια συμβαίνει όταν οι νευρώνες του δικτύου λειτουργούν σε γραμμικές περιοχές.

Για τον τρόπο ενημέρωσης των βαρών πρότυπο προς πρότυπο τα αποτελέσματα εξομοιώσεων δείχνουν παρόμοιες τάσεις με τη σωρηδόν λειτουργία που αναφέρθηκε μέχρι εδώ. Οι Russo (1991) και Haykin (1999) συνιστούν έναν εμπειρικό τύπο για τις αρχικές τιμές των βαρών για να αποφεύγεται ο κορεσμός των νευρώνων. Αυτόν τον τύπο τον περιγράφουμε στην ενότητα 4.4.

Άσκηση 4.2/1:

Ο πρόωρος κορεσμός είναι η κατάσταση όπου στο ΝΔ:

1. το στιγμιαίο άθροισμα των τετραγωνικών λαθών παραμένει σταθερό για μια χρονική περίοδο και η κλίση της συνάρτησης ενεργοποίησης του νευρώνα είναι πολύ μικρή,
2. η κλίση της συνάρτησης ενεργοποίησης του νευρώνα είναι πολύ μεγάλη,
3. τα σήματα λάθους έχουν μεγάλο μέγεθος,
4. το μέσο τετραγωνικό λάθος παραμένει σταθερό.

Άσκηση 4.2/2:

Ποιες είναι οι βασικές αρχές για να αποφύγουμε τον πρόωρο κορεσμό;

4.3 Το πρόβλημα της XOR

Στο απλό perceptron δεν υπάρχουν κρυφοί νευρώνες. Κατά συνέπεια δεν μπορεί να ταξινομήσει πρότυπα εισόδου τα οποία δεν είναι γραμμικά διαχωρίσιμα. Όμως μη-γραμμικά διαχωρίσιμα πρότυπα εμφανίζονται συνήθως στα προβλήματα. Για παράδειγμα, αυτό εμφανίζεται στο πρόβλημα Exclusive OR (XOR), το οποίο μπορεί να θεωρηθεί ως μια ειδική περίπτωση ενός πιο γενικού προβλήματος, αυτού της ταξινόμησης σημείων στον μοναδιαίο υπερκύβο. Κάθε σημείο στον υπερκύβο είναι είτε στην τάξη 0 είτε στην τάξη 1. Ωστόσο, στην ειδική περίπτωση του προβλήματος XOR, χρειάζεται να λάβουμε υπόψη μας μόνο τις τέσσερις γωνίες (άκρα) του μοναδιαίου τετραγώνου, οι οποίες αντιστοιχούν στα πρότυπα εισόδου (0,0), (0,1), (1,1) και (1,0). Το πρώτο και τρίτο πρότυπο εισόδου είναι στην τάξη 0, όπως φαίνεται από την:

$$0 \text{ XOR } 0 = 0$$

και:

$$1 \text{ XOR } 1 = 0$$

Τα πρότυπα εισόδου (0,0) και (1,1) είναι σε αντίθετες γωνίες του μοναδιαίου τετραγώνου και παρ' όλα αυτά παράγουν την ταυτόσημη έξοδο 0. Από την άλλη μεριά, τα πρότυπα εισόδου (0,1) και (1,0) είναι επίσης σε αντίθετες γωνίες του τετραγώνου, αλλά είναι στην τάξη 1, όπως φαίνεται από την:

$$0 \text{ XOR } 1 = 1$$

και:

$$\text{XOR } 0 = 1$$

Αρχικά αναγνωρίζουμε ότι η χρήση ενός απλού νευρώνα με δύο εισόδους έχει ως αποτέλεσμα μια ευθεία γραμμή για όριο απόφασης στον χώρο εισόδου. Για όλα τα σημεία στη μια πλευρά αυτής της γραμμής ο νευρώνας δίνει έξοδο 1, ενώ για όλα τα σημεία στην άλλη πλευρά της γραμμής δίνει έξοδο 0. Η θέση και ο προσανατολισμός της γραμμής στον χώρο εισόδου καθορίζονται από τα συναπτικά βάρη του νευρώνα με τα οποία είναι συνδεδεμένος με τους κόμβους εισόδου και το κατώφλι που εφαρμόζεται στον νευρώνα. Με τα πρότυπα εισόδου (0,0) και (1,1) τοποθετημένα σε αντίθετες γωνίες του μοναδιαίου τετραγώνου και παρομοίως για τα άλλα δύο πρότυπα εισόδου (0,1) και (1,0) είναι προφανές ότι δεν μπορούμε να κατασκευάσουμε μια ευθεία γραμμή για ένα όριο απόφασης έτσι ώστε (0,0) και (1,1) να βρίσκονται στη μια περιοχή απόφασης και (0,1) και (1,0) να βρίσκονται στην άλλη περιοχή απόφασης. Με άλλα λόγια, ένα στοιχειώδες perceptron δεν μπορεί να επιλύσει το πρόβλημα της XOR.

Μπορεί να επιλύσουμε το πρόβλημα της XOR χρησιμοποιώντας ένα απλό κρυφό επίπεδο με δύο νευρώνες, όπως στο σχήμα 4.1.a (Touretzky & Pomerleau, 1989· Haykin, 1999). Το διάγραμμα ροής σήματος του δικτύου απεικονίζεται στο σχήμα 4.1.b. Οι υποθέσεις που έγιναν εδώ είναι οι ακόλουθες:

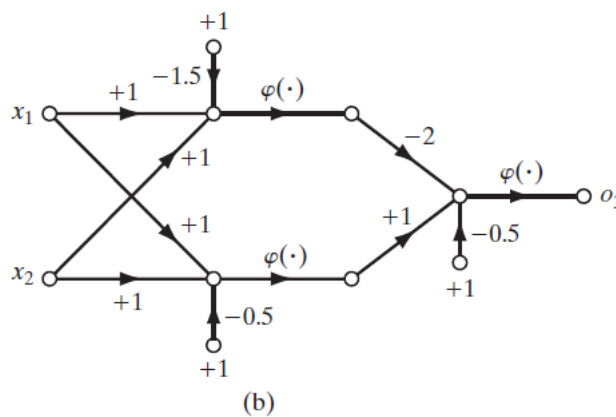
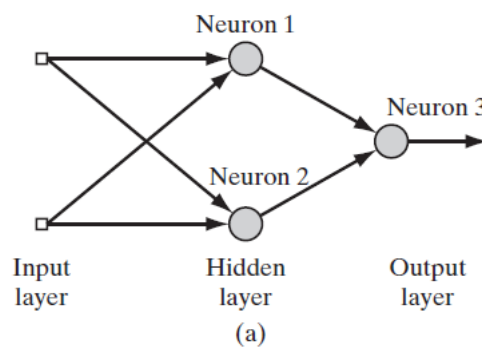
- Κάθε νευρώνας αναπαρίσταται από ένα μοντέλο McCulloch – Pitts.
- Τα ψηφία 0 και 1 αναπαρίστανται από τα επίπεδα 0 και +1 αντίστοιχα.

Ο νευρώνας κορυφής, που επιγράφεται ως 1 στο κρυφό επίπεδο, χαρακτηρίζεται από τις ακόλουθες σχέσεις:

$$w_{11} = w_{12} = +1$$

$$\theta_1 = +3/2$$

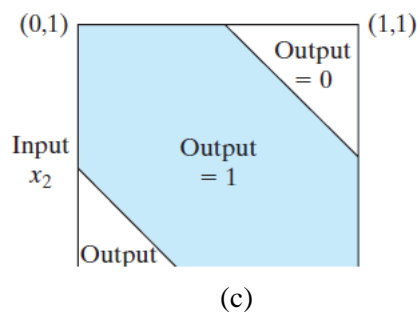
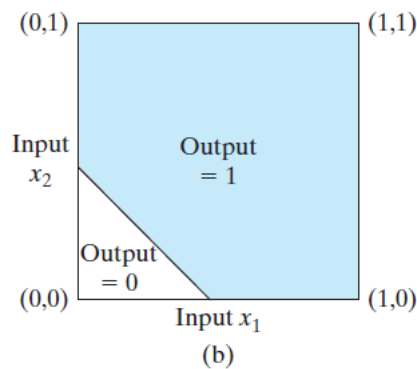
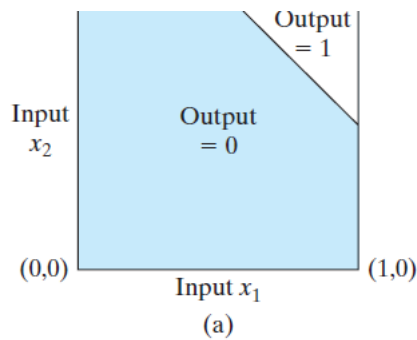
Η κλίση του ορίου απόφασης που κατασκευάζεται από αυτόν τον κρυφό νευρώνα ισούται με -1 και τοποθετείται όπως στο σχήμα 4.2.b (είναι δανεισμένο από την αναφορά Haykin, 1999). Ο νευρώνας πυθμένα, που επιγράφεται ως 2 στο κρυφό επίπεδο, χαρακτηρίζεται από τις ακόλουθες σχέσεις:



Σχήμα 4.1: (α) Ο αρχιτεκτονικός γράφος δικτύου για την επίλυση του XOR προβλήματος. (β) Το διάγραμμα ροής σημάτων του δικτύου (Haykin, 1999).

$$w_{21} = w_{22} = +1$$

$$\theta_2 = +1/2$$



Σχήμα 4.2 (α) Όριο απόφασης που κατασκευάστηκε από τον κρυμμένο νευρώνα 1 του δικτύου στο σχήμα 4.1. (β) Όριο απόφασης που κατασκευάστηκε από τον κρυμμένο νευρώνα 2 του δικτύου. (γ) Όρια απόφασης που κατασκευάστηκαν από το συνολικό δίκτυο (Haykin, 1996).

Ο προσανατολισμός και η θέση του ορίου απόφασης που κατασκευάζεται από αυτό το δεύτερο κρυφό νευρώνα είναι όπως φαίνεται στο σχήμα 4.2.b.

Ο νευρώνας εξόδου, που επιγράφεται ως 3 στο σχήμα 4.1.a, χαρακτηρίζεται από τις ακόλουθες σχέσεις:

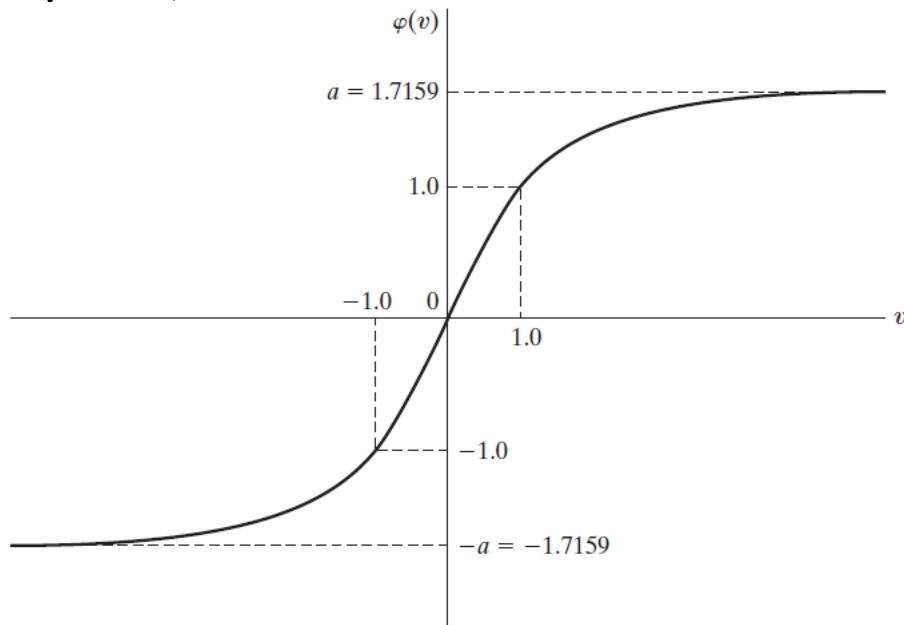
$$\begin{aligned} w_{31} &= -2 \\ w_{32} &= +1 \\ \theta_3 &= +1/2 \end{aligned}$$

Η λειτουργία του νευρώνα εξόδου είναι να κατασκευάζει έναν γραμμικό συνδυασμό των ορίων απόφασης που διαμορφώνονται από τους δύο κρυφούς νευρώνες. Το αποτέλεσμα αυτού του υπολογισμού φαίνεται στο σχήμα 4.2.c. Ο κρυφός νευρώνας στο κάτω μέρος έχει μια διεγερτική (θετική) σύνδεση με τον νευρώνα εξόδου ενώ ο νευρώνας στο πάνω μέρος έχει μια δυνατότερη κωλυτική (αρνητική) σύνδεση με τον νευρώνα εξόδου. Όταν και οι δύο κρυφοί νευρώνες είναι off, το οποίο συμβαίνει όταν το πρότυπο εισόδου είναι (0,0), ο νευρώνας εξόδου παραμένει off. Όταν και οι δύο κρυφοί νευρώνες είναι on, το οποίο συμβαίνει όταν το πρότυπο εισόδου είναι (1,1), ο νευρώνας εξόδου γίνεται πάλι off, επειδή η κωλυτική επίδραση του μεγαλύτερου αρνητικού βάρους, που είναι συνδεδεμένο με τον κρυφό νευρώνα κορυφής υπερκαλύπτει τη διεγερτική επίδραση του θετικού βάρους που είναι συνδεδεμένο με τον κρυφό νευρώνα στον πυθμένα. Όταν ο κρυφός νευρώνας κορυφής είναι off και ο κρυφός κάτω νευρώνας είναι on, το οποίο συμβαίνει όταν το πρότυπο εισόδου είναι (0,1) ή (1,0), ο νευρώνας εξόδου τίθεται on εξαιτίας της διεγερτικής επίδρασης του θετικού

βάρους που είναι συνδεδεμένο με τον κάτω κρυφό νευρώνα. Επομένως το δίκτυο του σχήματος 4.1.a επιλύει πραγματικά το πρόβλημα της XOR.

4.4 Μερικοί τρόποι αποτελεσματικότερης εκτέλεσης του αλγορίθμου

Πολλές φορές λέγεται πως ο σχεδιασμός ενός νευρωνικού δικτύου χρησιμοποιώντας τον αλγόριθμο πίσω διάδοσης είναι περισσότερο τέχνη παρά επιστήμη, με την έννοια πως πολλοί από τους πολυάριθμους παράγοντες που περιλαμβάνονται στον σχεδιασμό είναι πράγματι το αποτέλεσμα της προσωπικής πείρας του καθενός μας. Πράγματι υπάρχει κάποια αλήθεια στην προηγούμενη πρόταση. Εντούτοις υπάρχουν τρόποι με τους οποίους είναι δυνατόν να κάνουμε τον αλγόριθμο πίσω διάδοσης να δουλεύει καλύτερα (Russo, 1991· Guyon, 1991· Haykin, 1999).



Σχήμα 4.3 Ασυμμετρική συνάρτηση ενεργοποίησης (υπερβολική).

1. Ένα πολυεπίπεδο perceptron εκπαιδευμένο με τον αλγόριθμο πίσω διάδοσης θα μπορούσε γενικά να μαθαίνει γρηγορότερα (με την έννοια του αριθμού των επαναλήψεων που χρειάζονται για μάθηση) όταν η σιγμοειδής συνάρτηση ενεργοποίησης που έχει υλοποιηθεί στο μοντέλο του νευρώνα του δικτύου είναι ασυμμετρική (asymmetric) παρά όταν είναι μη συμμετρική. Λέμε πως μια συνάρτηση ενεργοποίησης $\phi(v)$ είναι ασυμμετρική όταν:

$$\phi(-v) = -\phi(v)$$

όπως απεικονίζεται στο σχήμα 4.3. Αυτή η συνθήκη δεν ικανοποιείται από τη μη συμμετρική λογιστική συνάρτηση.

Ένα γνωστό παράδειγμα μιας ασυμμετρικής συνάρτησης ενεργοποίησης είναι η σιγμοειδής μη γραμμική με τη μορφή της υπερβολικής εφαπτομένης, που ορίζεται ως

$$\phi(v) = a \tanh(bv)$$

όπου τα a και b είναι σταθερές. Σημειώνουμε πως η υπερβολική εφαπτομένη είναι απλά η λογιστική συνάρτηση πολωμένη, όπως δείχνουμε παρακάτω:

$$a \tanh(bv) = a \left[\frac{1 - \exp(-bv)}{1 + \exp(-bv)} \right] = \frac{2a}{1 + \exp(-bv)} - a \quad (4.1)$$

Αντίστοιχα οι αλλαγές που γίνονται στη διατύπωση του αλγορίθμου πίσω διάδοσης χρησιμοποιώντας αυτήν τη μορφή της σιγμοειδούς μη γραμμικότητας είναι μικρής σημασίας.

Κατάλληλες τιμές για τις σταθερές a και b είναι (Guyon 1991) (Haykin, 1999):

$$a = 1.716 \quad \text{και} \quad b = 2/3$$

2. Είναι σημαντικό οι τιμές στόχος (επιθυμητή απόκριση) να επιλεγούν μέσα στο διάστημα της σιγμοειδούς συνάρτησης ενεργοποίησης. Ειδικότερα η επιθυμητή απόκριση d_j για τον νευρώνα j στο επίπεδο εξόδου του πολυεπίπεδου perceptron θα πρέπει να μετατοπιστεί κατά κάποια ποσότητα ε μακριά από την περιοριστική τιμή της σιγμοειδούς συνάρτησης ενεργοποίησης, εξαρτώμενη από το εάν η περιοριστική τιμή είναι θετική ή αρνητική. Στην αντίθετη περίπτωση, ο αλγόριθμος πίσω διάδοσης έχει την τάση να οδηγήσει τις ελεύθερες παραμέτρους του δικτύου στο άπειρο και ως εκ τούτου επιβραδύνει τη διαδικασία μάθησης κατά τάξεις μεγέθους.

Πιο συγκεκριμένα θεωρήστε την ασυμμετρική συνάρτηση ενεργοποίησης του σχήματος 4.3. Για την πεπερασμένη τιμή $+a$ θέτουμε $d_j = a - \varepsilon$, όπου ε είναι μια κατάλληλη θετική σταθερά. Για την επιλογή $a = 1.716$ που αναφέρθηκε νωρίτερα μπορούμε να θέσουμε $\varepsilon = 0.716$. Σε αυτήν την περίπτωση η επιθυμητή τιμή d_j μπορεί να επιλεγεί κατάλληλα να είναι ± 1 .

3. Η αρχικοποίηση των συναπτικών βαρών και των επιπέδων κατωφλίου του δικτύου θα πρέπει να είναι ομοιόμορφα κατανομημένα σε ένα μικρό διάστημα. Ο λόγος που μικραίνουμε το διάστημα είναι για να μειώσουμε την πιθανότητα των νευρώνων στο δίκτυο να κορεστούν και να παραγάγουν μικρές κλίσεις λάθους. Όμως το διάστημα δεν θα πρέπει να γίνει πολύ μικρό, διότι μπορεί να έχει ως αποτέλεσμα οι κλίσεις λάθους να είναι πολύ μικρές και γι' αυτό η μάθηση αρχικά να είναι πολύ αργή. Για μια ασυμμετρική συνάρτηση ενεργοποίησης με τη μορφή της υπερβολικής εφαπτομένης, για την οποία οι σταθερές a και b προσδιορίστηκαν παραπάνω, μια πιθανή δυνατότητα αρχικοποίησης είναι να πάρουμε τυχαία τιμές για τα συναπτικά βάρη και επίπεδα κατωφλίου, οι οποίες είναι ομοιόμορφα κατανομημένες στο διάστημα:

$$\left(-\frac{2.4}{F_i}, +\frac{2.4}{F_i} \right) \quad (4.2)$$

όπου το F_i είναι το fan-in (δηλαδή ο συνολικός αριθμός των εισόδων) του νευρώνα i μέσα στο δίκτυο· με άλλα λόγια η αρχικοποίηση βαρών γίνεται από νευρώνα σε νευρώνα.

4. Όλοι οι νευρώνες του πολυεπίπεδου perceptron είναι επιθυμητό να μαθαίνουν με την ίδια ταχύτητα. Χαρακτηριστικά τα τελευταία επίπεδα τείνουν να έχουν μεγαλύτερες τοπικές κλίσεις σε σχέση με τα επίπεδα στην αρχή του δικτύου. Γι' αυτόν τον λόγο στην παράμετρο η του ρυθμού μάθησης θα πρέπει να εκχωρηθεί μια μικρότερη τιμή στα τελευταία επίπεδα σε σχέση με τα εμπρός επίπεδα. Νευρώνες με πολλές εισόδους θα πρέπει να έχουν μικρότερη παράμετρο ρυθμού μάθησης σε σχέση με νευρώνες που έχουν λίγες εισόδους.

5. Για on-line λειτουργία η ενημέρωση πρότυπο προς πρότυπο είναι προτιμότερη σε σχέση με σωρηδόν ενημέρωση για τη ρύθμιση των βαρών. Για προβλήματα ταξινόμησης δειγμάτων που περιλαμβάνουν μια μεγάλη και πλεονάζουσα βάση δεδομένων η πρότυπο προς πρότυπο ενημέρωση τείνει να είναι τάξεις μεγέθους γρηγορότερη σε σχέση με τη σωρηδόν ενημέρωση. Ωστόσο, η πρότυπο προς πρότυπο ενημέρωση είναι πιο δύσκολο να εκτελεστεί παράλληλα.

6. Η σειρά με την οποία εμφανίζονται τα παραδείγματα εκπαίδευσης στο δίκτυο θα πρέπει να καθορίζεται τυχαία από κύκλο σε κύκλο. Η μορφή της τυχαιοποίησης είναι κρίσιμη για την παραπέρα βελτίωση στην ταχύτητα της σύγκλισης. Επίσης, η χρήση ουρών μπορεί να βελτιώσει αποδοτικά την εκπαίδευση (Baum, 1991· Haykin, 1999).

7. Η εκμάθηση από ένα σύνολο δειγμάτων εκπαίδευσης σχετίζεται με μια άγνωστη συνάρτηση $f(\cdot)$ η οποία αντιστοιχίζει την είσοδο στην έξοδο. Ως εκ τούτου η διαδικασία μάθησης εκμεταλλεύεται την πληροφορία που

περιλαμβάνεται στα παραδείγματα σχετικά με τη συνάρτηση $f(\cdot)$ και εξάγει μια προσεγγιστική υλοποίησή της. Η διαδικασία μάθησης από παραδείγματα μπορεί να γενικευτεί, ώστε να συμπεριλάβει μάθηση από υπαινιγμούς, η οποία επιτυγχάνεται επιτρέποντας στην πληροφορία που εκ των προτέρων έχουμε για τη συνάρτηση $f(\cdot)$ να συμπεριληφθεί στη διαδικασία μάθησης (Abu-Mostafa, 1990· Al-Mashouq & Reed, 1991· Haykin, 1999).

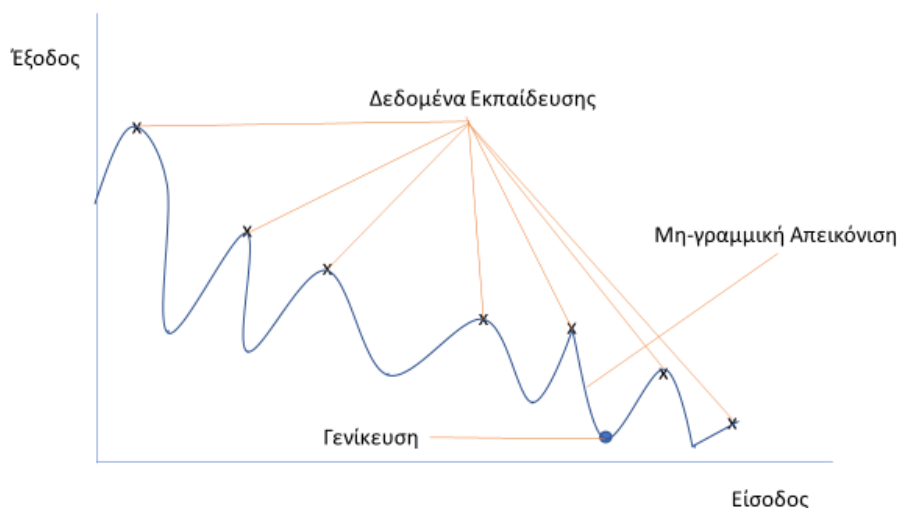
Άσκηση 4.4/3:

Να αναφέρετε τους βασικούς τρόπους για την αποτελεσματικότερη εκτέλεση του αλγορίθμου. Ποιοι από αυτούς είναι οι πιο σημαντικοί και γιατί;

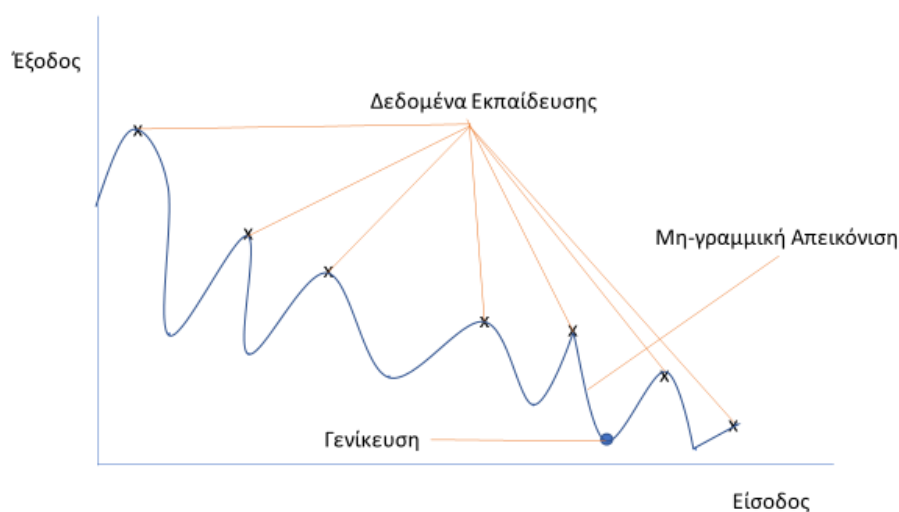
4.5 Γενίκευση

Στη μάθηση πίσω διάδοσης ξεκινάμε με ένα εκπαιδευτικό σύνολο και χρησιμοποιούμε τον αλγόριθμο πίσω διάδοσης για να υπολογίσουμε τα συναπτικά βάρη ενός πολυεπίπεδου perceptron φορτώνοντας (κωδικοποιώντας) όσα περισσότερα από τα εκπαιδευτικά παραδείγματα είναι δυνατόν μέσα στο δίκτυο. Η επιθυμία είναι το νευρωνικό δίκτυο έτσι σχεδιασμένο να γενικεύει. Ένα δίκτυο λέγεται ότι γενικεύει καλά όταν η σχέση εισόδου-εξόδου υπολογιζόμενη από το δίκτυο είναι σωστή (ή σχεδόν σωστή) για δείγματα εισόδου-εξόδου (δεδομένα ελέγχου) που ποτέ δεν χρησιμοποιήθηκαν στη δημιουργία ή στην εκπαίδευση του δικτύου. Ο όρος «γενίκευση» είναι δανεισμένος από την ψυχολογία. Εδώ βέβαια υποτίθεται ότι τα δεδομένα ελέγχου προέρχονται από τον ίδιο πληθυσμό που χρησιμοποιήθηκε για να παραγάγει τα εκπαιδευτικά δεδομένα.

Η διαδικασία μάθησης (εκπαίδευσης ενός νευρωνικού δικτύου) μπορεί να θεωρηθεί ως ένα πρόβλημα προσαρμογής καμπύλης. Το δίκτυο το ίδιο μπορεί να θεωρηθεί απλά μία μη-γραμμική αντιστοίχιση εισόδων σε εξόδους. Μια τέτοια άποψη μάς επιτρέπει να δούμε τη γενίκευση όχι σαν μία μυστηριώδη ιδιότητα των νευρωνικών δικτύων, αλλά απλούστερα ως την επίδραση μίας καλής μη-γραμμικής παρεμβολής των δεδομένων εισόδου. Το δίκτυο εκτελεί χρήσιμη παρεμβολή κυρίως επειδή πολυεπίπεδα perceptrons με συνεχείς συναρτήσεις ενεργοποίησης οδηγούν σε εξωτερικές συναρτήσεις που είναι επίσης συνεχείς.



(α)



(β)

Σχήμα 4.4 (α) Κατάλληλα ταιριασμένα δεδομένα (καλή γενίκευση) (β) Υπερβολικά ταιριασμένα δεδομένα (φτωχή γενίκευση) (Haykin, 1999).

Το σχήμα 4.4α επεξηγεί πώς η γενίκευση μπορεί να συμβεί σε ένα υποθετικό δίκτυο. Η μη-γραμμική αντιστοίχιση εισόδων σε εξόδους που παριστάνεται από την καμπύλη που απεικονίζεται σε αυτό το σχήμα υπολογίζεται από το δίκτυο ως αποτέλεσμα της εκμάθησης των σημείων με ετικέτα «εκπαιδευτικά δεδομένα». Το σημείο που είναι μαρκαρισμένο στην καμπύλη ως «γενίκευση» είναι αυτό που φαίνεται απλώς ως το αποτέλεσμα της παρεμβολής που εκτελείται από το δίκτυο.

Ένα νευρωνικό δίκτυο που σχεδιάστηκε να γενικεύει καλά θα παράγει μία σωστή αντιστοίχιση εισόδων σε εξόδους, ακόμα και όταν η είσοδος είναι λίγο διαφορετική από τα παραδείγματα που χρησιμοποιήθηκαν για να εκπαιδευτεί, όπως επεξηγείται στο σχήμα 4.4α. Όταν όμως ένα νευρωνικό δίκτυο μαθαίνει αρκετές ειδικές σχέσεις εισόδου-εξόδου (υπερεκπαιδεύεται), τότε μπορεί να κρατάει στη μνήμη του τα δεδομένα εκπαίδευσης και συνεπώς να είναι λιγότερο ικανό να γενικεύει μεταξύ παρόμοιων δειγμάτων εισόδου-εξόδου. Δεδομένα λοιπόν που φορτώνονται σε ένα πολυεπίπεδο perceptron με τέτοιο τρόπο απαιτούν τη χρησιμοποίηση περισσότερων κρυφών νευρώνων από όσους πραγματικά είναι απαραίτητο, με αποτέλεσμα ανεπιθύμητες καμπύλες στον χώρο του προβλήματος να αποθηκεύονται στα συναπτικά βάρη του δικτύου. Ένα παράδειγμα για το πώς μπορεί να προκύψει φτωχή γενίκευση οφειλόμενη σε «μνημοποίηση» σε ένα νευρωνικό δίκτυο επεξηγείται στο σχήμα 4.4β για τα ίδια δεδομένα που απεικονίζονται στο σχήμα 4.4α. Η μνημοποίηση είναι βασικά ένας «πίνακας ψαξίματος» (“look-up table”) που υπονοεί ότι η αντιστοίχιση εισόδων σε εξόδους που υπολογίζεται από το νευρωνικό δίκτυο δεν είναι «λεία».

Άσκηση 4.5/4:

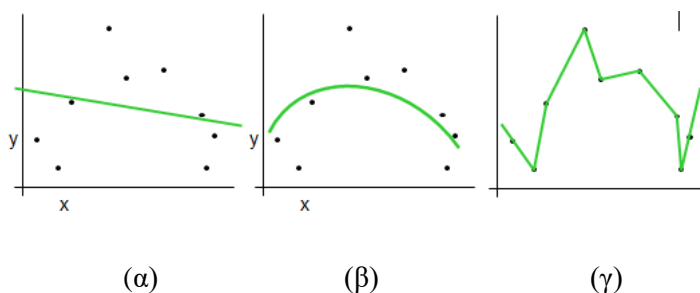
Για την κατασκευή ενός καλού ΝΔ το οποίο μαθαίνει καλά τα δεδομένα εκπαίδευσης, τι μας ενδιαφέρει περισσότερο, να επιτύχουμε πολύ μικρό λάθος εκπαίδευσης ή όσο το δυνατόν μικρότερο λάθος γενίκευσης;

4.6 Επικύρωση εκπαιδευμένου νευρωνικού δικτύου

Όπως ήδη αναφέρθηκε, η εκπαίδευση του ΝΔ ολοκληρώνεται όταν ικανοποιηθεί ένα από τα κριτήρια τερματισμού του αλγορίθμου. Στη συνέχεια όμως το ΝΔ πρέπει να αποτιμηθεί, ώστε να επικυρωθεί η αξιοπιστία της λειτουργίας του. Με άλλα λόγια δεν αρκεί να πάρουμε ένα μικρό λάθος στην έξοδο κατά την εκπαίδευση του ΝΔ, αλλά πρέπει να ελεγχθεί η σωστή λειτουργία του, δηλαδή αν απαντά σωστά και για πρότυπα τα οποία δεν έχουν χρησιμοποιηθεί κατά την εκπαίδευση του δικτύου. Αυτό λέγεται ικανότητα γενίκευσης. Η επικύρωση είναι σημαντική όχι μόνο για να μετρήσουμε την απόδοση του δικτύου, αλλά και για να αναγνωρίσουμε τη δυνατότητα βελτίωσης της απόδοσης, σχεδιάζοντας έτσι διαφορετικά το ΝΔ.

Εκτός από την επιλογή των παραμέτρων που αναφέρθηκαν παραπάνω, κρίσιμη για τη σωστή εκπαίδευση του δικτύου είναι η επιλογή της κατάλληλης αρχιτεκτονικής. Ένα πολύ μεγάλο δίκτυο (με πολλά κρυφά επίπεδα και πολλούς νευρώνες ανά επίπεδο) μπορεί να μάθει με μεγάλη ακρίβεια το εκπαιδευτικό σύνολο, αλλά συνήθως τέτοια δίκτυα έχουν μικρή ικανότητα γενίκευσης· μαθαίνει καλά τα δεδομένα εκπαίδευσης, αλλά δεν παρουσιάζει καλές επιδόσεις σε νέα δεδομένα. Αυτό οφείλεται στο ότι το δίκτυο, λόγω της μεγάλης ευελιξίας του μαθαίνει πολύπλοκες αναπαραστάσεις τις οποίες δεν χρειάζεται και δεν μαθαίνει το μοντέλο που παράγει τα δεδομένα. Αυτή η κατάσταση λέγεται υπερεκπαίδευση. Στην αντίθετη περίπτωση, που το ΝΔ είναι πολύ μικρό ή τα δεδομένα εκπαίδευσης δεν είναι επαρκή, τότε το δίκτυο δεν μπορεί να μάθει πολύπλοκες αναπαραστάσεις (και να διαχωρίσει τις περιοχές απόφασης), οπότε έχουμε υποεκπαίδευση.

Στο σχήμα 4.5 απεικονίζονται οι τρεις παραπάνω περιπτώσεις. Στο σχήμα 4.5.α το δίκτυο έχει υποεκπαιδευτεί, ενώ στο σχήμα 4.5.γ το δίκτυο έχει υπερεκπαιδευτεί. Στο σχήμα 4.5.β φαίνεται η γενίκευση του δικτύου. Ο έλεγχος της γενικευτικής ικανότητας μπορεί να γίνει με τους παρακάτω τρόπους:



Σχήμα 4.5 (α) Υποεκπαίδευση (β) Γενίκευση (γ) Υπερεκπαίδευση.

1. Η μέθοδος του συνόλου ελέγχου. Το σύνολο εκπαίδευσης διαιρείται σε δύο μέρη, σε αναλογία 70/30 (ή 80/20 κλπ.). Το πρώτο μέρος, το 70%, χρησιμοποιείται για την εκπαίδευση του δικτύου, ενώ το υπόλοιπο 30% για τον έλεγχο της απόδοσης του δικτύου. Η εκτίμηση της απόδοσης στο σύνολο ελέγχου γίνεται με τον υπολογισμό του Μέσου Τετραγωνικού Λάθους (ΜΤΛ). Ένα σοβαρό μειονέκτημα της μεθόδου είναι ότι ένα σημαντικό μέρος των δειγμάτων μένει εκτός εκπαίδευσης του δικτύου. Αν το σύνολο εκπαίδευσης είναι μικρό, αυτό μπορεί να έχει για αποτέλεσμα την υποεκπαίδευση του δικτύου.
2. Διασταυρωμένη επικύρωση άφησε – ένα – έξω (leave - one - out cross validation). Αν το σύνολο εκπαίδευσης περιλαμβάνει N δείγματα, αφαιρούμε το ένα δείγμα και κάνουμε την εκπαίδευση με τα $N-1$ δείγματα. Μετά χρησιμοποιούμε το ένα δείγμα για έλεγχο και σημειώνουμε το λάθος. Επαναλαμβάνουμε τη διαδικασία (συνολικά N φορές) και στο τέλος υπολογίζουμε το σφάλμα γενίκευσης ως τον μέσο όρο των επιμέρους σφαλμάτων. Το κέρδος από αυτήν τη διαδικασία είναι ότι όλα τα πρότυπα χρησιμοποιούνται για την εκπαίδευση, αλλά χρειάζεται να επαναληφθεί N φορές.
3. Διασταυρωμένη επικύρωση k -τιμημάτων (k -fold cross-validation, k -CV). Διαιρούμε τυχαία το σύνολο εκπαίδευσης σε k μη-επικαλυπτόμενα υποσύνολα (π.χ. $k=10$). Χρησιμοποιούμε τα $k-1$ υποσύνολα για την εκπαίδευση και το k -οστό υποσύνολο για τον έλεγχο και υπολογίζουμε το σφάλμα γενίκευσης. Επαναλαμβάνουμε τη διαδικασία $k-1$ φορές (αφήνοντας έξω ένα υποσύνολο κάθε φορά και εκπαιδεύοντας το δίκτυο με τα υπόλοιπα) και υπολογίζουμε τα σφάλματα γενίκευσης. Στο τέλος υπολογίζουμε το σφάλμα γενίκευσης ως τον μέσο όρο των επιμέρους σφαλμάτων.

Άσκηση 4.5/5: Να συγκρίνετε τις μεθόδους διασταυρωμένης επικύρωσης και να συμπληρώσετε τον παρακάτω πίνακα:

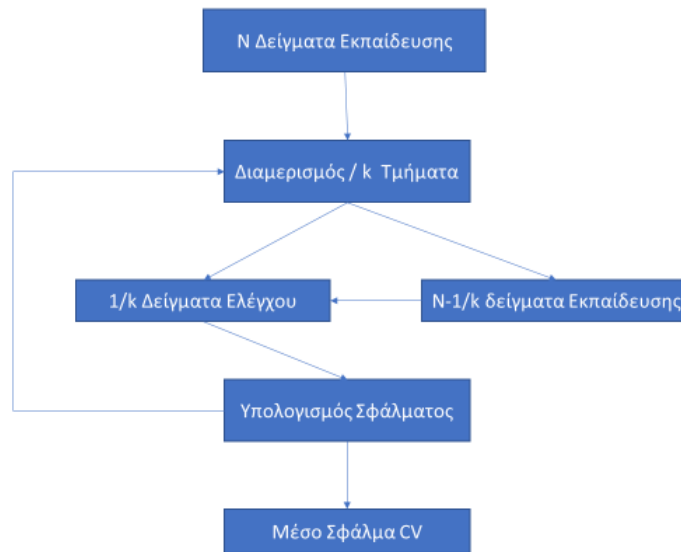
	Μειονεκτήματα	Πλεονεκτήματα
Σύνολο Ελέγχου		
Leave-one-out		
10-fold		
3-fold		
K-fold		

4.6.1 Επιλογή καλύτερου δικτύου με διασταυρωμένη επικύρωση

Η μέθοδος διασταυρωμένης επικύρωσης μπορεί να χρησιμοποιηθεί για την επιλογή του καλύτερου δικτύου, με την έννοια της καλύτερης ικανότητας γενίκευσης. Αρχικά θα περιγράψουμε τη μέθοδο για ένα MLP, με ένα κρυφό επίπεδο και M νευρώνες· στόχος είναι να επιλεγεί ο «βέλτιστος» αριθμός κρυφών νευρώνων. Η μέθοδος μπορεί να γενικευτεί για περισσότερα του ενός κρυφά επίπεδα, με M_1, M_2, \dots κρυφούς νευρώνες. Ακολουθεί μια σύντομη περιγραφή της μεθόδου.

1. Καθορισμός του αρχικού αριθμού M (π.χ. $M=2$) των κρυφών νευρώνων, του μέγιστου αριθμού M_{\max} , κρυφών νευρώνων, του αριθμού τμημάτων k (π.χ. $k=10$) και των παραμέτρων εκπαίδευσης (π.χ. ρυθμός μάθησης).
2. Διαμερισμός του εκπαιδευτικού συνόλου σε k υποσύνολα για την εφαρμογή της τεχνικής k -CV.
3. Υπολογισμός με (k -CV) του σφάλματος γενίκευσης για M κρυφούς νευρώνες. Αύξηση του αριθμού των κρυφών νευρώνων, π.χ. $M:=M+1$ και επιστροφή στο βήμα 3, εάν $M \leq M_{\max}$.
4. Επιλογή ως βέλτιστης αρχιτεκτονικής εκείνης με το μικρότερο σφάλμα γενίκευσης. Εκπαίδευση του MLP με M^* κρυφούς νευρώνες σε όλο το σύνολο παραδειγμάτων και εύρεση της τελικής λύσης.

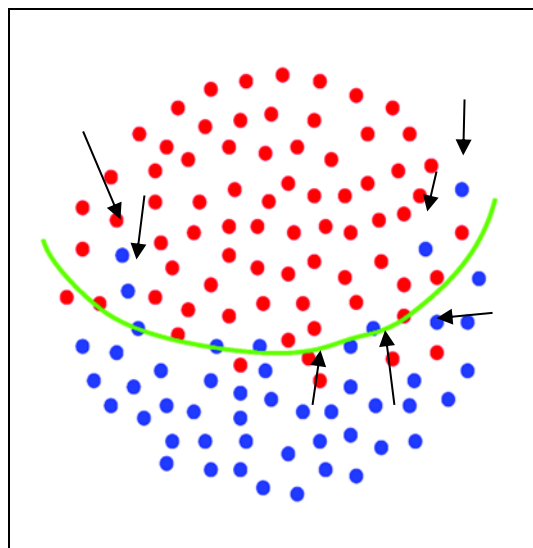
Στο παρακάτω σχήμα φαίνεται το διάγραμμα ροής της διαδικασίας της διασταυρωμένης επικύρωσης (CV).



Σχήμα 4.6 Διάγραμμα Ροής Διασταυρωμένης Επικύρωσης.

Άσκηση 4.6/6

Έστω ένα σύνολο από διδιάστατα παραδείγματα δύο κατηγοριών που απεικονίζονται στο παρακάτω σχήμα με μπλε και κόκκινες κουκίδες αντίστοιχα. Η πράσινη γραμμή δείχνει το πραγματικό όριο απόφασης μεταξύ των κατηγοριών. Με βέλη υποδεικνύονται κάποια «παραδείγματα», τα οποία έχουν καταταχθεί σε λάθος κατηγορία.



Το παραπάνω σύνολο παραδειγμάτων χρησιμοποιήθηκε για την εκπαίδευση δύο δικτύων (MLP1 και MLP2) με δύο εισόδους, ένα κρυμμένο επίπεδο και μία έξοδο. Η μόνη διαφορά στην αρχιτεκτονική των δύο δικτύων βρίσκεται στον αριθμό των κρυμμένων νευρώνων. Το MLP1 έχει $M1$ κρυμμένους νευρώνες και το MLP2 έχει $M2$ κρυμμένους νευρώνες και ισχύει ότι $M1 < M2$.

Καθένα από τα δύο δίκτυα εκπαιδεύτηκε με τον ίδιο ακριβώς αλγόριθμο (π.χ. EBP) με ελαχιστοποίηση του τετραγωνικού σφάλματος σε ολόκληρο το σύνολο παραδειγμάτων. Θεωρούμε ότι το σφάλμα εκπαίδευσης ελαχιστοποιήθηκε όσο το δυνατόν καλύτερα, δηλαδή για κάθε δίκτυο βρήκαμε το ολικό ελάχιστο ($E1$ και $E2$ αντίστοιχα) του τετραγωνικού σφάλματος εκπαίδευσης.

Α. Ποια είναι σχέση μεταξύ $E1$ και $E2$ (π.χ. $E1 < E2$) και γιατί;

- Β.** Εάν το MLP2 έχει **υποεκπαιδευτεί** (δηλαδή δεν έχουμε καταφέρει να μειώσουμε το συνολικό σφάλμα εκπαίδευσής του κάτω από ένα αποδεκτό όριο), η γενικευτική ικανότητα του MLP1 θα είναι χαμηλή ή υψηλή και γιατί;
- Γ.** Εάν το MLP1 έχει **υπερεκπαιδευτεί**, η γενικευτική ικανότητα του MLP2 θα είναι χαμηλή ή υψηλή και γιατί;
- Δ.** Εάν τα E1 και E2 είναι τοπικά αλλά όχι ολικά ελάχιστα της συνάρτησης σφάλματος κάθε δικτύου, ισχύουν οι απαντήσεις των ερωτημάτων Α, Β, Γ και γιατί;

4.6.2 Βέλτιστες σταθερές μάθησης και ορμής

Για τις βέλτιστες τιμές του ρυθμού μάθησης η και της σταθεράς ορμής α μπορούμε να χρησιμοποιήσουμε οποιονδήποτε από τους παρακάτω ορισμούς:

1. Τα η και α τα οποία, στη μέση περίπτωση, επιφέρουν σύγκλιση σε ένα τοπικό ελάχιστο της επιφάνειας λάθους του δικτύου με τον μικρότερο αριθμό κύκλων.
2. Τα η και α τα οποία, στη μέση ή τη χειρότερη περίπτωση, επιφέρουν σύγκλιση στο ολικό ελάχιστο της επιφάνειας λάθους με τον μικρότερο αριθμό κύκλων.
3. Τα η και α τα οποία, στη μέση περίπτωση, επιφέρουν σύγκλιση σε εκείνη τη διαμόρφωση του δικτύου η οποία έχει την καλύτερη γενίκευση, πάνω στο συνολικό πεδίο εισόδων, με τον μικρότερο αριθμό κύκλων.

Η μέση και χειρότερη περίπτωση αναφέρονται στην κατανομή των ζευγών εισόδου-εξόδου της εκπαίδευσης. Ο ορισμός 3 είναι ο ιδανικός. Στην πράξη όμως είναι δύσκολο να εφαρμοστεί, καθώς η ελαχιστοποίηση του μέσου τετραγωνικού λάθους είναι συνήθως το μαθηματικό κριτήριο για βελτιστοποίηση κατά τη διάρκεια της εκπαίδευσης του δικτύου και, όπως σημειώθηκε πριν από λίγο, ένα μικρότερο τετραγωνικό λάθος πάνω στο σύνολο εκπαίδευσης δεν συνεπάγεται απαραίτητα και καλή γενίκευση. Από ερευνητική άποψη ο ορισμός 2 έχει προκαλέσει μεγαλύτερο ενδιαφέρον από τον ορισμό 1. Για παράδειγμα στον Luo (1991) (Haykin) 1999) παρουσιάζονται αυστηρά αποτελέσματα για τη βέλτιστη προσαρμογή του ρυθμού μάθησης η , έτσι ώστε να απαιτείται ο μικρότερος αριθμός κύκλων για το MLP ώστε να προσεγγίσει τον συνολικά βέλτιστο πίνακα συναπτικών βαρών σε μια επιθυμητή ακρίβεια, εκτός από την ειδική περίπτωση των γραμμικών νευρώνων. Γενικά ωστόσο, ευριστικές και πειραματικές διαδικασίες χρησιμοποιούν τελευταία τον ορισμό 1 για τη βέλτιστη επιλογή των η και α .

Στην αναφορά (Haykin, 1999) παρουσιάζεται ένα πείραμα, στο οποίο χρησιμοποιείται ο ορισμός 1. Χρησιμοποιώντας ένα MLP με δύο κρυμμένους νευρώνες, διάφοροι συνδυασμοί των $\eta \in \{0,01, 0,1, 0,5, 0,9\}$ και $\alpha \in \{0,0, 0,1, 0,5, 0,9\}$ εξομοιώνονται για να παρατηρηθούν τα αποτελέσματά τους στη σύγκλιση του δικτύου. Κάθε συνδυασμός εκπαιδεύεται με το ίδιο σύνολο αρχικών τυχαίων βαρών και το ίδιο σύνολο από 500 δείγματα εισόδου-εξόδου, έτσι ώστε τα αποτελέσματα του πειράματος να μπορούν να συγκριθούν άμεσα. Όπως ειπώθηκε προηγουμένως, ένα δίκτυο θεωρείται ότι έχει συγκλίνει όταν ο απόλυτος ρυθμός αλλαγής του μέσου τετραγωνικού λάθους ανά κύκλο είναι επαρκώς μικρός. Σε αυτό το πείραμα επιλέχθηκε ο συγκεκριμένος ρυθμός να είναι μικρότερος από 0,01 %.

Τα αποτελέσματα της προσομοίωσης από τις πειραματικές καμπύλες μάθησης οδηγούν στα παρακάτω συμπεράσματα:

- Καθώς σε γενικές γραμμές μικρότερος ρυθμός μάθησης η έχει ως αποτέλεσμα πιο αργή σύγκλιση, μπορεί να εντοπίσει «βαθύτερο» τοπικό ελάχιστο στην επιφάνεια λάθους από ό,τι ένα μεγαλύτερο η . Αυτό είναι και διαισθητικά σωστό, καθώς μικρότερο η συνεπάγεται ότι η αναζήτηση για ελάχιστο θα καλύψει μεγαλύτερο μέρος της επιφάνειας λάθους από την περίπτωση ενός μεγαλύτερου η .
- Για $\eta \rightarrow 0$ η χρήση $\alpha \rightarrow 1$ προκαλεί αύξηση στην ταχύτητα σύγκλισης. Από την άλλη, για $\eta \rightarrow 1$ η χρήση $\alpha \rightarrow 0$ απαιτείται για να σιγουρέψει τη σταθερότητα μάθησης.

- Η χρήση των σταθερών $\eta = \{0,5, 0,9\}$ και $\alpha = 0,9$ προκαλεί ταλαντώσεις στο μέσο τετραγωνικό λάθος κατά τη διάρκεια μάθησης και μια υψηλότερη τιμή για το τελικό μέσο τετραγωνικό λάθος της σύγκλισης, δύο αποτελέσματα τα οποία είναι ανεπιθύμητα.

Αυτά τα αποτελέσματα ενισχύουν την αντίληψη ότι ένα χαμηλότερο μέσο τετραγωνικό σφάλμα κατά μήκος του εκπαιδευτικού συνόλου δεν είναι από μόνο του μία ικανή συνθήκη για μία καλύτερη απόδοση ταξινόμησης. Πράγματι, η μελέτη συσχέτισης της πειραματικά καθοριζόμενης πιθανότητας της σωστής ταξινόμησης έναντι του τελικού μέσου τετραγωνικού σφάλματος παρουσιάζει μόνο έναν αδύνατο αρνητικό συσχετισμό -0.1220 ανάμεσα στα δύο μέτρα απόδοσης του δικτύου.

Άσκηση 4.6/7:

Αφού ο ο ορισμός 3 είναι ο ιδανικός για τον εντοπισμό των βέλτιστων τιμών του n και α , γιατί στην πράξη χρησιμοποιείται τελικά ο ορισμός 1 για την επιλογή των n και α ;

4.7 Κατάλληλο μέγεθος εκπαιδευτικού συνόλου για έγκυρη γενίκευση

Η γενίκευση επηρεάζεται από τρεις παράγοντες: το μέγεθος και την αποτελεσματικότητα του εκπαιδευτικού συνόλου, την αρχιτεκτονική του δικτύου και τη φυσική πολυπλοκότητα του ίδιου του προβλήματος. Είναι φανερό ότι δεν έχουμε κανέναν έλεγχο γύρω από τον τελευταίο παράγοντα. Σχετικά με τους άλλους δύο μπορούμε να δούμε το ζήτημα της γενίκευσης από δύο διαφορετικές σκοπιές (Hush & Horne, 1993) (Haykin, 1999):

- Η αρχιτεκτονική του δικτύου είναι φιξαρισμένη (ευτυχώς και σε αντιστοίχιση με τη φυσική πολυπλοκότητα του δεδομένου προβλήματος) και το ζήτημα που πρέπει να επιλυθεί είναι αυτό του καθορισμού του μεγέθους του εκπαιδευτικού συνόλου που απαιτείται για να προκύψει μία καλή γενίκευση.
- Το μέγεθος του εκπαιδευτικού συνόλου είναι φιξαρισμένο και το ζήτημα που μας ενδιαφέρει είναι ο καθορισμός της καλύτερης αρχιτεκτονικής του δικτύου για την επίτευξη μίας καλής γενίκευσης.

Παρόλο που και οι δύο αυτές απόψεις είναι έγκυρες με τον δικό τους ξεχωριστό τρόπο, η πρώτη είναι αυτή που αντιμετωπίζουμε πιο συχνά στην πράξη. Γι' αυτό θα συγκεντρωθούμε σε αυτή στο εξής.

Πράγματι η καταλληλότητα του μεγέθους του εκπαιδευτικού συνόλου είναι ένα θεωρητικό ζήτημα που έχει προσελκύσει ένα μεγάλο ποσό προσοχής στη βιβλιογραφία και συνεχίζει να το κάνει. Σε αυτήν την υποενότητα θα περιγράψουμε περιληπτικά ένα χρήσιμο αποτέλεσμα προερχόμενο από τους Baum και Haussler (1989) (Haykin, 1999) για την περίπτωση ενός νευρωνικού δικτύου που περιέχει ένα μόνο κρυφό επίπεδο και χρησιμοποιείται ως δυαδικός ταξινομητής. Για να προετοιμάσουμε το έδαφος για μία πρόταση του μαθηματικού τύπου τους, πρώτα παρουσιάζουμε κάποιους ορισμούς.

Ένα παράδειγμα ορίζεται ως ζεύγος $\{x, d\}$ όπου το διάνυσμα εισόδου $x \in \mathbb{R}^p$ και η επιθυμητή έξοδος $d \in [-1, 1]$ με άλλα λόγια το δίκτυο δρα ως δυαδικός ταξινομητής. Ένας κύκλος ορίζεται ως μία σειρά από παραδείγματα που λαμβάνονται ανεξάρτητα και κατά τυχαίο τρόπο από κάποια κατανομή D . Έστω f μία συνάρτηση από το διάστημα \mathbb{R}^p στο $[-1, 1]$, με $d=f(x)$. Ένα λάθος της συνάρτησης f , με αναφορά στην κατανομή D ορίζεται ως η πιθανότητα η έξοδος y να είναι διαφορετική από το d για ένα ζεύγος (x, d) που επιλέχθηκε κατά τυχαίο τρόπο. Έστω M ο συνολικός αριθμός των κρυφών υπολογιστικών κόμβων· έστω W ο συνολικός αριθμός των συναπτικών βαρών στο δίκτυο· έστω N ο αριθμός των τυχαίων παραδειγμάτων που χρησιμοποιούνται για να εκπαιδεύσουν το δίκτυο· έστω ϵ το φράγμα των λαθών που επιτρέπονται στο τεστ. Τότε, βάσει των Baum και Haussler, το δίκτυο σχεδόν βέβαια παρέχει γενίκευση, αν και μόνο αν ικανοποιούνται οι δύο παρακάτω συνθήκες:

1. Το φράγμα των λαθών που δημιουργήθηκαν από το εκπαιδευτικό σύνολο είναι μικρότερο από $\epsilon/2$.
2. Ο αριθμός των παραδειγμάτων N , που χρησιμοποιήθηκαν στην εκπαίδευση είναι:

$$N \geq \frac{32W}{\epsilon} \ln \left(\frac{32M}{\epsilon} \right) \quad (4.3)$$

όπου \ln υποδηλώνει τον φυσικό λογάριθμο.

Η σχέση (4.3) παρέχει έναν ελεύθερο από κατανομές και χειρότερης περίπτωσης τύπο για την εκτίμηση του μεγέθους του εκπαιδευτικού συνόλου για ένα νευρωνικό δίκτυο ενός επιπέδου που είναι κατάλληλο για μία καλή γενίκευση. Λέμε «χειρότερης περίπτωσης» επειδή στην πράξη μπορεί να υπάρχει ένα τεράστιο αριθμητικό κενό ανάμεσα στο πραγματικό μέγεθος του εκπαιδευτικού συνόλου που απαιτείται και σε αυτό που προβλέπεται από το κριτήριο της σχέσης (4.3). Πρέπει όμως να υπογραμμιστεί ότι αυτό το κενό είναι απλώς μία αντανάκλαση της φύσης της χειρότερης περίπτωσης του κριτηρίου.

Αγνοώντας τον λογαριθμικό παράγοντα στη σχέση (4.3), βλέπουμε ότι ο κατάλληλος αριθμός από εκπαιδευτικά παραδείγματα είναι, σε μία πρώτη προσέγγιση, ανάλογος του αριθμού των βαρών στο δίκτυο και αντιστρόφως ανάλογος της παραμέτρου ακριβείας ϵ . Πράγματι φαίνεται ότι στην πράξη αυτό που εντέλει χρειαζόμαστε για μία καλή γενίκευση είναι να ικανοποιείται η συνθήκη:

$$N \geq \frac{W}{\epsilon} \quad (4.4)$$

Έτσι με ένα λάθος της τάξης του 10% λέμε ότι ο αριθμός των εκπαιδευτικών παραδειγμάτων πρέπει να είναι προσεγγιστικά 10 φορές ο αριθμός των συναπτικών βαρών στο δίκτυο.

Άσκηση 4.7/8:

Ποια είναι η επίδραση του μεγέθους του εκπαιδευτικού συνόλου στην εκπαίδευση του νευρωνικού δικτύου;

Άσκηση 4.7/9:

Ποιο είναι το αποτέλεσμα της αύξησης του αριθμού των κρυφών νευρώνων;

Άσκηση 4.7/10:

Ποια είναι τα βασικά συμπεράσματα σχετικά με τη βέλτιστη τιμή των παραμέτρων η και α ;

Άσκηση 4.7/11:

Να αναφέρετε ένα κριτήριο σύγκλισης ενός ΝΔ, δηλαδή πότε τερματίζει η εκτέλεση του αλγορίθμου.

Άσκηση 4.7/12:

Ποιες συνθήκες και με ποιους περιορισμούς καθορίζουν το κατάλληλο μέγεθος εκπαιδευτικού συνόλου για έγκυρη γενίκευση;

4.8 Σύνοψη κεφαλαίου

Σε αυτό το κεφάλαιο ασχοληθήκαμε με θέματα που αφορούν την υλοποίηση ΤΝΔ πολλών επιπέδων. Πρώτα συζητήσαμε το πρόβλημα της αρχικοποίησης και μερικούς τρόπους αποτελεσματικότερης εκτέλεσης των αλγορίθμων. Όμως το βασικό πρόβλημα στην υλοποίηση ενός ΝΔ για την επίλυση ενός πραγματικού προβλήματος είναι ότι δεν είναι εκ των προτέρων γνωστή η διαμόρφωση του δικτύου· δηλαδή δεν γνωρίζουμε τον ακριβή αριθμό κρυφών επιπέδων ούτε και τον αριθμό των κόμβων ανά κρυφό επίπεδο. Επειδή δεν υπάρχει κάποια γενική θεωρία που επιλύει αυτό το πρόβλημα, προσπαθήσαμε να δώσουμε μερικές ιδέες για το πώς αντιμετωπίζεται ο καθορισμός του βέλτιστου δικτύου.

Το βασικό πλεονέκτημα των perceptrons πολλών επιπέδων είναι ότι μπορούν να διαχωρίσουν μη γραμμικά διαχωριζόμενα πρότυπα. Θεωρητικά λοιπόν μπορούν να αντιμετωπίσουν οποιοδήποτε πρόβλημα ταξινόμησης προτύπων. Δυστυχώς όμως στην πράξη τα πράγματα δεν είναι τόσο απλά. Όπως ήδη αναφέραμε, το βασικό πρόβλημα ενός τέτοιου δικτύου είναι ο καθορισμός της βέλτιστης και καταλληλότερης δομής για την επίλυση ενός συγκεκριμένου προβλήματος. Έχοντας επιλέξει σωστά το εκπαιδευτικό σύνολο, αυτόματα έχουμε καθορίσει τον αριθμό των εισόδων και των εξόδων του δικτύου. Δεν έχουμε όμως καμία εκ των προτέρων γνώση για την εσωτερική αρχιτεκτονική του δικτύου (αριθμός κρυφών επιπέδων και αριθμός κόμβων ανά κρυφό επίπεδο). Ένας προφανής τρόπος για να αντιμετωπίσουμε αυτό το πρόβλημα είναι με τη μέθοδο δοκιμής και λάθους (trial and error). Με βάση κάποιους εμπειρικούς κανόνες υλοποιούμε μια συγκεκριμένη αρχιτεκτονική δικτύου και το εκπαιδεύουμε, ελπίζοντας σε καλή γενίκευση. Αν μετά την εκπαίδευση το λάθος

στην έξοδο του δικτύου είναι σχετικά μεγάλο, τροποποιούμε τη δομή του (προσθέτοντας ή αφαιρώντας κρυφά επίπεδα ή/και κόμβους) και το εκπαιδεύουμε πάλι. Έτσι, με διαδοχικές δοκιμές, ελπίζουμε να επιτύχουμε το (σχεδόν) βέλτιστο δίκτυο. Είναι προφανές ότι αυτή η διαδικασία είναι χρονοβόρα και επίπονη. Γι' αυτόν τον λόγο λέγεται ότι μαθαίνει κανείς μαζί με το δίκτυο.

Το πρόβλημα του πειραματικού προσδιορισμού του βέλτιστου πολυεπίπεδου perceptron αντιμετωπίστηκε σαν ένα σύνολο από υποπροβλήματα. Αυτά ήταν ο καθορισμός του βέλτιστου αριθμού κρυφών νευρώνων, των βέλτιστων σταθερών μάθησης και ορμής και του βέλτιστου σχεδιασμού του δικτύου. Τέλος, ασχοληθήκαμε με το πρόβλημα της γενίκευσης και του υπολογισμού του κατάλληλου μεγέθους του εκπαιδευτικού συνόλου για έγκυρη γενίκευση.

Βιβλιογραφία

Davis, L. (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley Publishing Company, Inc.

Hagan, M.T., Demuth, H.B., & Beal, M. (1996). *Neural Network Design*. PWS Publishing Company.

Haykin, S. (1999). *Neural Networks, A Comprehensive Foundation* (2η έκδ.). Prentice Hall International.

Απαντήσεις ασκήσεων

Άσκηση 4.2/1:

Ο πρόωρος κορεσμός είναι η κατάσταση όπου στο ΝΔ:

1. το στιγμιαίο άθροισμα των τετραγωνικών λαθών παραμένει σταθερό για μια χρονική περίοδο και η κλίση της συνάρτησης ενεργοποίησης του νευρώνα είναι πολύ μικρή,
2. η κλίση της συνάρτησης ενεργοποίησης του νευρώνα είναι πολύ μεγάλη,
3. τα σήματα λάθους έχουν μεγάλο μέγεθος,
4. το μέσο τετραγωνικό λάθος παραμένει σταθερό.

Απάντηση: Η σωστή απάντηση είναι η 1.

Άσκηση 4.2/2:

Ποιες είναι οι βασικές αρχές, για να αποφύγουμε τον πρόωρο κορεσμό;

Απάντηση: Είναι οι τρεις αρχές που αναφέρονται στο τέλος της ενότητας 4.2.

Άσκηση 4.4/3:

Να αναφέρετε τους βασικούς τρόπους για την αποτελεσματικότερη εκτέλεση του αλγορίθμου. Ποιοι από αυτούς είναι οι πιο σημαντικοί και γιατί;

Απάντηση: Οι βασικοί τρόποι αναφέρονται σε αυτήν την ενότητα. Οι πιο σημαντικοί είναι η επιλογή της συνάρτησης ενεργοποίησης, η αρχικοποίηση των βαρών και οι κατάλληλες τιμές των παραμέτρων μάθησης και ορμής.

Άσκηση 4.5/4:

Για την κατασκευή ενός καλού ΝΔ, το οποίο μαθαίνει καλά τα δεδομένα εκπαίδευσης, τι μας ενδιαφέρει περισσότερο, να επιτύχουμε πολύ μικρό λάθος εκπαίδευσης ή όσο το δυνατόν μικρότερο λάθος γενίκευσης;

Απάντηση:

Ένα πολύ μικρό λάθος εκπαίδευσης πιθανόν να σημαίνει ότι το δίκτυο έχει υπερεκπαιδευτεί, άρα δεν θα έχει καλή ικανότητα γενίκευσης. Αντίθετα, ένα δίκτυο με μικρό λάθος γενίκευσης είναι προτιμότερο, ακόμη κι αν το λάθος εκπαίδευσης είναι μεγαλύτερο.

Άσκηση 4.5/5: Να συγκρίνετε τις μεθόδους διασταυρωμένης επικύρωσης και να συμπληρώσετε τον παρακάτω πίνακα:

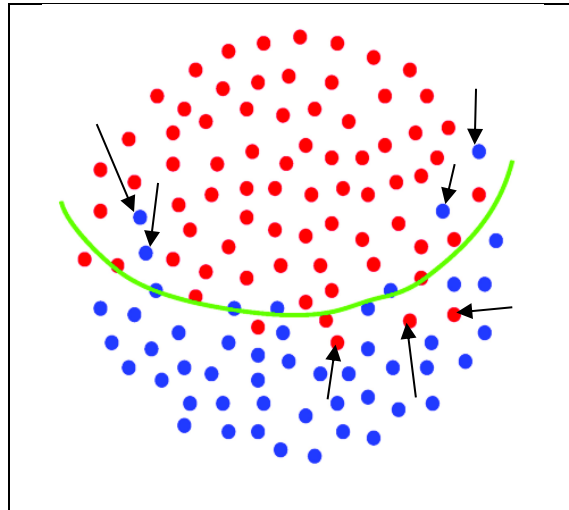
	Μειονεκτήματα	Πλεονεκτήματα
Σύνολο Ελέγχου		
Leave-one-out		
10-fold		
3-fold		
K-fold		

Απάντηση:

	Μειονεκτήματα	Πλεονεκτήματα
Σύνολο Ελέγχου	Μη αξιόπιστη εκτίμηση της γενίκευσης.	Φθηνή, αλλά σπαταλά δεδομένα εκπαίδευσης.
Leave-one-out	Ακριβή. Παρουσιάζει κάποια περίεργη συμπεριφορά.	Δεν σπαταλά δεδομένα.
10-fold	Σπαταλά 10% των δεδομένων. 10 φορές πιο ακριβή από το σύνολο ελέγχου.	Σπαταλά μόνο το 10% των δεδομένων. Μόνο 10 φορές πιο ακριβή αντί για K φορές.
3-fold	Πιο σπάταλη από το 10-fold. Πιο ακριβή από το σύνολο ελέγχου.	Λίγο καλύτερη από το σύνολο ελέγχου.
K-fold	Πανομοιότητα με leave-one-out	

Άσκηση 4.6/6

Έστω ένα σύνολο από διδιάστατα παραδείγματα δύο κατηγοριών που απεικονίζονται στο παρακάτω σχήμα με μπλε και κόκκινες κουκίδες αντίστοιχα. Η πράσινη γραμμή δείχνει το πραγματικό όριο απόφασης μεταξύ των κατηγοριών. Με βέλη υποδεικνύονται κάποια «παραδείγματα», τα οποία έχουν καταταχθεί σε λάθος κατηγορία.



Το παραπάνω σύνολο παραδειγμάτων χρησιμοποιήθηκε για την εκπαίδευση δύο δικτύων (MLP1 και MLP2) με δύο εισόδους, ένα κρυμμένο επίπεδο και μία έξοδο. Η μόνη διαφορά στην αρχιτεκτονική των δύο δικτύων βρίσκεται στον αριθμό των κρυμμένων νευρώνων. Το MLP1 έχει M1 κρυμμένους νευρώνες και το MLP2 έχει M2 κρυμμένους νευρώνες και ισχύει ότι $M1 < M2$.

Καθένα από τα δύο δίκτυα εκπαιδεύτηκε με τον ίδιο ακριβώς αλγόριθμο (π.χ. EBP) με ελαχιστοποίηση του τετραγωνικού σφάλματος σε ολόκληρο το σύνολο παραδειγμάτων. Θεωρούμε ότι το σφάλμα εκπαίδευσης ελαχιστοποιήθηκε όσο το δυνατόν καλύτερα, δηλαδή για κάθε δίκτυο βρήκαμε το ολικό ελάχιστο (E1 και E2 αντίστοιχα) του τετραγωνικού σφάλματος εκπαίδευσης.

A. Ποια είναι σχέση μεταξύ E1 και E2 (π.χ. $E1 < E2$) και γιατί;

Αφού το δίκτυο M2 έχει περισσότερους νευρώνες, μπορεί να ταιριάζει καλύτερα στα παραδείγματα του συνόλου εκπαίδευσης σε σχέση με το M1. Άρα το ολικό ελάχιστο της E2 της συνάρτησης σφάλματος του M2 θα είναι μικρότερο ή ίσο από αυτό του M1. Επομένως, $E2 \leq E1$.

Η ισότητα ισχύει π.χ. στην περίπτωση όπου και τα δύο δίκτυα έχουν την απαιτούμενη ευελιξία ώστε να μάθουν επακριβώς τα παραδείγματα του συνόλου εκπαίδευσης.

B. Εάν το MLP2 έχει **υποεκπαιδευτεί** (δηλαδή δεν έχουμε καταφέρει να μειώσουμε το συνολικό σφάλμα εκπαίδευσής του κάτω από ένα αποδεκτό όριο), η γενικευτική ικανότητα του MLP1 θα είναι χαμηλή ή υψηλή και γιατί;

Εάν το μεγαλύτερο δίκτυο (MLP2) έχει υποεκπαιδευτεί, αυτό υποδηλώνει ότι οι M2 νευρώνες δεν επαρκούν για την ικανοποιητική μείωση του σφάλματος εκπαίδευσης. Άρα και το δίκτυο με λιγότερους νευρώνες (MLP1) θα έχει κι αυτό υποεκπαιδευτεί και συνεπώς θα έχει χαμηλή γενικευτική ικανότητα.

Γ. Εάν το MLP1 έχει **υπερεκπαιδευτεί**, η γενικευτική ικανότητα του MLP2 θα είναι χαμηλή ή υψηλή και γιατί;

Εάν το μικρότερο δίκτυο (MLP1) έχει υπερεκπαιδευτεί, αυτό υποδηλώνει ότι οι M1 νευρώνες είναι περισσότεροι από όσους χρειάζονται και το δίκτυο έχει μάθει και τον θόρυβο που υπάρχει στο σύνολο εκπαίδευσης. Αφού το δίκτυο MLP2 έχει περισσότερους νευρώνες από το MLP1, θα έχει κι αυτό υπερεκπαιδευτεί και συνεπώς θα έχει χαμηλή γενικευτική ικανότητα.

Δ. Εάν τα E1 και E2 είναι τοπικά αλλά όχι ολικά ελάχιστα της συνάρτησης σφάλματος κάθε δικτύου, ισχύουν οι απαντήσεις των ερωτημάτων A, B, Γ και γιατί;

Σε περίπτωση που η εκπαίδευση δεν έχει καταλήξει στο ολικό ελάχιστο της συνάρτησης του σφάλματος εκπαίδευσης, τα ανωτέρω συμπεράσματα μπορεί να μην ισχύουν. Για παράδειγμα, μπορεί ένα μεγάλο δίκτυο να έχει υποεκπαίδευτεί, διότι η εκπαίδευσή του εγκλωβίστηκε σε ρηχό τοπικό ελάχιστο της συνάρτησης σφάλματος. Επίσης, για τον ίδιο λόγο μπορεί ένα μεγάλο δίκτυο να αποφυγεί την υπερεκπαίδευση, διότι η εκπαίδευση θα σταματήσει σε υψηλότερη τιμή του σφάλματος και θα αποφευχθεί η μάθηση των λανθασμένων παραδειγμάτων· ανάλογα δηλαδή με το ελάχιστο στο οποίο θα καταλήξει η εκπαίδευση ενός μεγάλου δικτύου μπορεί να έχουμε είτε υποεκπαίδευση είτε ικανοποιητική εκπαίδευση είτε υπερεκπαίδευση. Το πιθανότερο όμως είναι ότι η εκπαίδευση ενός μεγάλου δικτύου θα καταλήξει σε χαμηλή τιμή σφάλματος, οπότε θα έχουμε υπερεκπαίδευση. Για τον λόγο αυτόν χρησιμοποιούνται τεχνικές όπως το πρόωρο σταμάτημα και η φθορά βαρών για την αποφυγή της υπερεκπαίδευσης (Ενότητα 6.2).

Άσκηση 4.6/7:

Αφού ο ορισμός 3 είναι ο ιδανικός για τον εντοπισμό των βέλτιστων τιμών του n και α , γιατί στην πράξη χρησιμοποιείται τελικά ο ορισμός 1 για την επιλογή των n και α ;

Απάντηση:

Ο ορισμός 3, αν και είναι ο ιδανικός, στην πράξη δεν μπορεί να εφαρμοστεί, καθώς η ελαχιστοποίηση του ΜΤΛ είναι το μαθηματικό κριτήριο για βελτιστοποίηση κατά τη διαδικασία της εκπαίδευσης. Αλλά, όπως είδαμε, ένα μικρότερο ΜΤΛ πάνω στο σύνολο εκπαίδευσης δεν συνεπάγεται απαραίτητα και καλή γενίκευση. Οπότε πειραματικά ο ορισμός 1 είναι προτιμότερος, γιατί βρίσκει τα n και α που επιφέρουν σύγκλιση σε ένα τοπικό ελάχιστο με τον μικρότερο αριθμό επαναλήψεων.

Άσκηση 4.7/8:

Ποια είναι η επίδραση του μεγέθους του εκπαιδευτικού συνόλου στην εκπαίδευση του νευρωνικού δικτύου;

Απάντηση:

Όσο μεγαλύτερο είναι το μέγεθος, τόσο μικρότερος είναι ο αριθμός των κύκλων και το μέσο τετραγωνικό λάθος.

Άσκηση 4.7/9:

Ποιο είναι το αποτέλεσμα της αύξησης του αριθμού των κρυφών νευρώνων;

Απάντηση:

Βελτιώνεται ελαφρώς το μέσο τετραγωνικό λάθος, αλλά δεν βελτιώνεται ο σωστός διαχωρισμός των προτύπων.

Άσκηση 4.7/10:

Ποια είναι τα βασικά συμπεράσματα σχετικά με τη βέλτιστη τιμή των παραμέτρων n και α ;

Απάντηση:

Είναι τρία και προκύπτουν από τα αποτελέσματα των προσομοιώσεων που παρουσιάζονται στην ενότητα 4.7.

Άσκηση 4.7/11:

Να αναφέρετε ένα κριτήριο σύγκλισης ενός ΝΔ, δηλαδή πότε τερματίζει η εκτέλεση του αλγορίθμου.

Απάντηση:

Σε κάθε επανάληψη πρέπει να ισχύει η συνθήκη: $E(n+1) - E(n) \leq \epsilon$, όπου ϵ οσοδήποτε μικρό, π.χ. $\epsilon=10^{-4}$.

Άσκηση 4.7/12:

Ποιες συνθήκες και με ποιους περιορισμούς καθορίζουν το κατάλληλο μέγεθος εκπαιδευτικού συνόλου για έγκυρη γενίκευση;

Απάντηση:

Αφού καθοριστεί το φράγμα λάθους ϵ , η απάντηση προκύπτει από τη σχέση (3).

Κεφάλαιο 5

Δίκτυα Hopfield και Δίκτυα Kohonen

Σύνοψη

Στα προηγούμενα κεφάλαια είδαμε μοντέλα ΝΔ τα οποία έχουμε εμπνευστεί από φυσικές διαδικασίες, προσομοιώνουν δηλαδή τη διαδικασία μάθησης των βιολογικών συστημάτων. Αυτά τα μοντέλα, κατά τη διαδικασία μάθησης, επιβλέπονται από έναν «δάσκαλο» ο οποίος τροφοδοτεί το δίκτυο με παραδείγματα της μορφής AN είσοδος – ΤΟΤΕ έξοδος. Αυτό είναι ένα παράδειγμα επιβλεπόμενης μάθησης. Όμως υπάρχει και το παράδειγμα μη επιβλεπόμενης μάθησης. Σε αυτήν την περίπτωση, κατά την εκπαίδευση, δεν δίνεται στο δίκτυο η επιθυμητή έξοδος, αλλά ένα σύνολο από παραδείγματα εισόδου.

Σκοπός του παρόντος κεφαλαίου είναι να κάνει μια εισαγωγή σε αυτά τα μοντέλα νευρωνικών δικτύων. Πρώτα θα ασχοληθούμε με τα λεγόμενα αναδρομικά δίκτυα (έχουν —εκτός από τις προς τα εμπρός— και προς τα πίσω συνδέσεις, δηλαδή βρόχους αναδρομής από τις εξόδους τους προς τις εισόδους τους). Αφού τροφοδοτηθεί η είσοδος, υπολογίζεται η έξοδος, η οποία ανατροφοδοτείται στην είσοδο, μέχρι η έξοδος να γίνει σταθερή. Αυτά τα δίκτυα λέγονται και δίκτυα αυτο-συσχετιστικής μνήμης (*auto-associative memory*) και τα πρώτα που εμφανίστηκαν είναι τα δίκτυα Hopfield, που παρουσιάζονται παρακάτω.

Μια δεύτερη πολύ ενδιαφέρουσα κατηγορία είναι οι χάρτες αυτοοργάνωσης (*Self-Organizing Maps SOMs*). Ένας χάρτης αυτοοργάνωσης χαρακτηρίζεται από έναν τοπογραφικό χάρτη των νευρώνων εισόδου, στον οποίο οι τοπικές θέσεις (δηλαδή οι συντεταγμένες) των νευρώνων στο πλέγμα είναι ενδεικτικές των εγγενών (φυσικών) στατιστικών χαρακτηριστικών που περιέχονται στα πρότυπα εισόδου. Το κίνητρο για την ανάπτυξη αυτού του μοντέλου οφείλεται στην ύπαρξη τοπολογικά διαταγμένων υπολογιστικών χαρτών στον εγκέφαλο. Ένας υπολογιστικός χάρτης καθορίζεται από μια διάταξη νευρώνων που αναπαριστούν ελαφρά διαφορετικούς συντονισμένους επεξεργαστές, που λειτουργούν παράλληλα στα σήματα εισόδου. Ως επακόλουθο αυτού οι νευρώνες μετασχηματίζουν τα σήματα εισόδου σε μια τοπικά κωδικοποιημένη πιθανοτική κατανομή, η οποία αναπαριστά τις υπολογισθείσες τιμές των παραμέτρων με την έννοια της μέγιστης σχετικής δραστηριότητας μέσα στον χάρτη. Με άλλα λόγια οι νευρώνες ανταγωνίζονται μεταξύ τους (διαδικασία γνωστή ως ανταγωνιστική μάθηση) ποιος θα επικρατήσει και θα δώσει την έξοδό του για απάντηση στην είσοδο. Αυτή η διαδικασία είναι γνωστή ως *winner-takes-all* (ο νικητής τα παίρνει όλα), επειδή οι τιμές των βαρών του νικητή νευρώνα είναι διάφορες του μηδενός, ενώ των υπολοίπων παίρνουν την τιμή μηδέν.

Υπάρχουν δύο διαφορετικά μοντέλα χαρτών αυτοοργάνωσης:

- το μοντέλο Willshaw von der Malsburg και
- το μοντέλο Kohonen.

Και στα δύο οι νευρώνες εξόδου τοποθετούνται σε ένα 2-διάστατο πλέγμα. Η διαφορά τους είναι ότι στο πρώτο μοντέλο και η είσοδος είναι ένα 2-διάστατο πλέγμα, ενώ στο δεύτερο η είσοδος είναι ένα διάνυσμα.

Επειδή το μοντέλο Kohonen είναι μια πιο γενική έκδοση του μοντέλου Willshaw von der Malsburg, στη συνέχεια θα ασχοληθούμε μόνο με αυτό.

Ολοκληρώνοντας αυτό το κεφάλαιο θα μπορείτε να σχεδιάσετε και να υλοποιήσετε δίκτυα μη επιβλεπόμενης μάθησης για επίλυση πρακτικών προβλημάτων.

Προσδοκώμενα αποτελέσματα

Όταν ολοκληρωθεί η μελέτη αυτού του κεφαλαίου, θα μπορείτε να:

- διακρίνετε τη διαφορά μεταξύ επιβλεπόμενης και μη επιβλεπόμενης μάθησης,
- σχεδιάσετε, υλοποιήσετε και εκπαιδεύσετε δίκτυα αυτοσυσχετιστικής μνήμης,
- σχεδιάσετε και υλοποιήσετε και εκπαιδεύσετε SOMs,
- χρησιμοποιήσετε τα παραπάνω δίκτυα σε επίλυση πρακτικών προβλημάτων.

Λέξεις-Κλειδιά:

- Μη επιβλεπόμενη μάθηση
- Αναδρομικά δίκτυα
- Αυτοσυσχετιστική μνήμη
- Αυτοοργάνωση
- Ανταγωνιστική μάθηση

5.1 Εισαγωγικές Παρατηρήσεις

Τα δίκτυα Hopfield ανήκουν στην κατηγορία των αναδρομικών Νευρωνικών Δικτύων. Ένα αναδρομικό Νευρωνικό Δίκτυο έχει —εκτός από τις προς τα εμπρός— και προς τα πίσω συνδέσεις· έχει δηλαδή βρόχους αναδρομής από τις εξόδους του προς τις εισόδους του. Η παρουσία τέτοιων βρόχων έχει ισχυρή επίπτωση στην ικανότητα μάθησης του δικτύου.

Ένα αναδρομικό Νευρωνικό Δίκτυο λοιπόν μαθαίνει ως εξής: αφού εφαρμοστεί μια είσοδος, υπολογίζεται η έξοδος του δικτύου, η οποία στη συνέχεια ανατροφοδοτείται ως είσοδος στο δίκτυο. Υπολογίζεται η νέα έξοδος του δικτύου και η διαδικασία επαναλαμβάνεται έως ότου η έξοδος του δικτύου γίνει σταθερή (πάψει να μεταβάλεται).

Βέβαια δεν γίνεται πάντα η έξοδος του δικτύου σταθερή. Η διαδικασία που περιγράψαμε παραπάνω δεν εξασφαλίζει πάντα ότι σε κάθε επανάληψη οι μεταβολές στην έξοδο του δικτύου θα είναι ολοένα και μικρότερες, έτσι ώστε σε κάποια χρονική στιγμή η έξοδος να πάψει να μεταβάλεται. Αντιθέτως, είναι πολύ πιθανόν να οδηγήσει σε μια χαοτική συμπεριφορά του δικτύου. Στην περίπτωση αυτή η έξοδος του δικτύου δεν γίνεται ποτέ σταθερή και τότε λέμε ότι το δίκτυο είναι ασταθές.

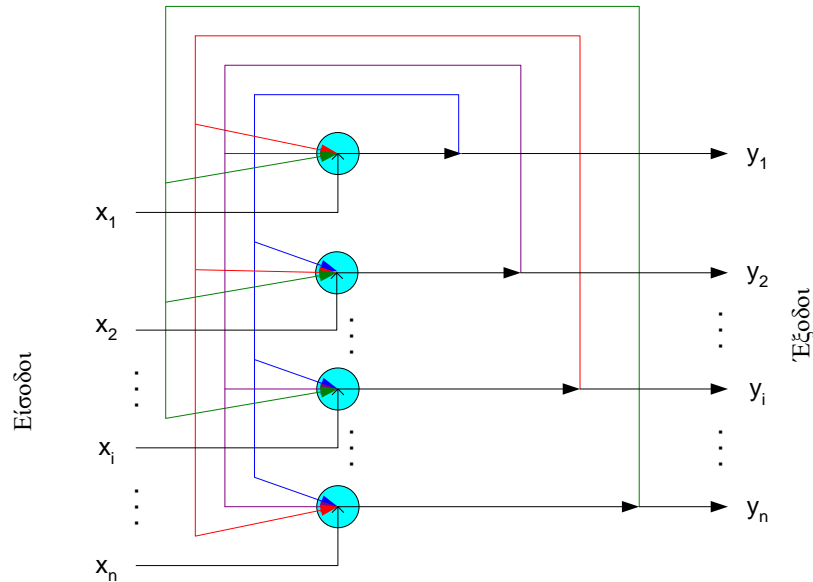
Η ευστάθεια των αναδρομικών Νευρωνικών Δικτύων αποτέλεσε αντικείμενο έρευνας πολλών ερευνητών του χώρου στις δεκαετίες του 1960 και 1970. Παρ' όλα αυτά, κανείς δεν κατάφερε να προβλέψει ποιο αναδρομικό δίκτυο θα μπορούσε να είναι ευσταθές, με αποτέλεσμα πολλοί να εκφράζουν μια απαισιοδοξία για το εάν τελικά θα μπορούσε να βρεθεί μια λύση σε αυτό το πρόβλημα. Η λύση όμως ήρθε το 1982, όταν ο John Hopfield διατύπωσε τη φυσική αρχή της αποθήκευσης πληροφορίας σε ένα δυναμικά ευσταθές δίκτυο (Hopfield, 1982).

Στη συνέχεια θα αναφερθούμε σε μια κατηγορία δικτύων, τα δίκτυα Kohonen, και σε έναν αλγόριθμο της κατηγορίας της μη επιβλεπόμενης μάθησης (un-supervised learning), δηλαδή της μάθησης χωρίς την παρουσία «δασκάλου». Αυτά τα δίκτυα είναι ιδιαίτερα χρήσιμα στην περίπτωση κατηγοριοποίησης προτύπων σε γενικές κατηγορίες (pattern clustering), όταν δεν γνωρίζουμε από πριν ποιες ακριβώς είναι αυτές οι κατηγορίες· δηλαδή αφήνουμε το Νευρωνικό Δίκτυο να «αποφασίσει» ποιες είναι οι κατηγορίες και να αντιστοιχίσει τα δείγματα εισόδου σε καθεμία από αυτές. Ειδικότερα θα αναφερθούμε σε ένα συγκεκριμένο είδος μη επιβλεπόμενης μάθησης, την ανταγωνιστική μάθηση (competitive learning), όπου οι νευρώνες ανταγωνίζονται μεταξύ τους για το ποιος θα ενεργοποιηθεί. Ενώ σε άλλα είδη μάθησης (για παράδειγμα στη μάθηση Hebb) μπορούν να ενεργοποιούνται ταυτόχρονα περισσότεροι του ενός νευρώνες, στην ανταγωνιστική μάθηση μόνο ένας μπορεί να είναι ενεργός κάθε φορά. Ο νευρώνας αυτός που «κερδίζει τον ανταγωνισμό» ονομάζεται “winner-takes-all neuron”.

Η βασική ιδέα της ανταγωνιστικής μάθησης έχει της «ρίζες» της στη δεκαετία του 1970, όμως για πρώτη φορά έτυχε της προσοχής της ακαδημαϊκής και ερευνητικής κοινότητας στα τέλη της δεκαετίας του 1980, όταν ο Teuvo Kohonen παρουσίασε μια ειδική τάξη τεχνητών Νευρωνικών Δικτύων, τα οποία ονόμασε αυτο-οργανωμένους χάρτες χαρακτηριστικών (self-organizing feature maps) (Kohonen, 1989). Τα δίκτυα αυτά βασίζονται στην ανταγωνιστική μάθηση.

5.2 Εκπαίδευση Δικτύου Hopfield

Στο σχήμα 5.1 απεικονίζεται ένα δίκτυο Hopfield ενός επιπέδου, το οποίο αποτελείται από n υπολογιστικούς νευρώνες. Η έξοδος κάθε νευρώνα ανατροφοδοτείται ως είσοδος σε όλους τους υπόλοιπους νευρώνες (στα δίκτυα Hopfield δεν υπάρχει αυτο-ανατροφοδότηση). Οι υπολογιστικοί νευρώνες που χρησιμοποιούνται ακολουθούν συνήθως το μοντέλο McCulloch – Pitts με συνάρτηση ενεργοποίησης τη συνάρτηση προσήμου (sign function).



Σχήμα 5.1 Ένα Δίκτυο Hopfield ενός επιπέδου με n νευρώνες.

Η λειτουργία ενός νευρώνα k που χρησιμοποιεί τη συνάρτηση προσήμου περιγράφεται από τις ακόλουθες σχέσεις:

$$v_k = \sum_{j=0}^m w_{kj} x_j \quad (5.1)$$

$$y_k = \varphi(v_k) = \begin{cases} +1, & v_k > 0 \\ -1, & v_k < 0, \\ y_k, & v_k = 0 \end{cases} \quad (5.2)$$

Δηλαδή η έξοδος του νευρώνα γίνεται +1 (ο νευρώνας μεταβαίνει σε κατάσταση +1) εάν το δυναμικό ενεργοποίησης v_k του νευρώνα είναι μεγαλύτερο του 0, -1 (μετάβαση σε κατάσταση -1) εάν το v_k είναι μικρότερο του μηδέν και παραμένει αμετάβλητη (ο νευρώνας παραμένει στην προηγούμενη του κατάσταση) εάν το v_k είναι ίσο με το μηδέν.

Μια άλλη συνάρτηση ενεργοποίησης που μπορεί να χρησιμοποιηθεί αντί της συνάρτησης προσήμου είναι η κορεσμένη γραμμική συνάρτηση (saturated linear function), η οποία συμπεριφέρεται ως καθαρά γραμμική συνάρτηση στο διάστημα $(-1, +1)$ και ως συνάρτηση προσήμου οπουδήποτε αλλού. Στην περίπτωση αυτή η έξοδος του δικτύου δίνεται από τη σχέση:

$$y_k = \varphi(v_k) = \begin{cases} +1, & v_k \geq +1 \\ -1, & v_k \leq -1 \\ v_k, & -1 < v_k < +1 \end{cases} \quad (5.3)$$

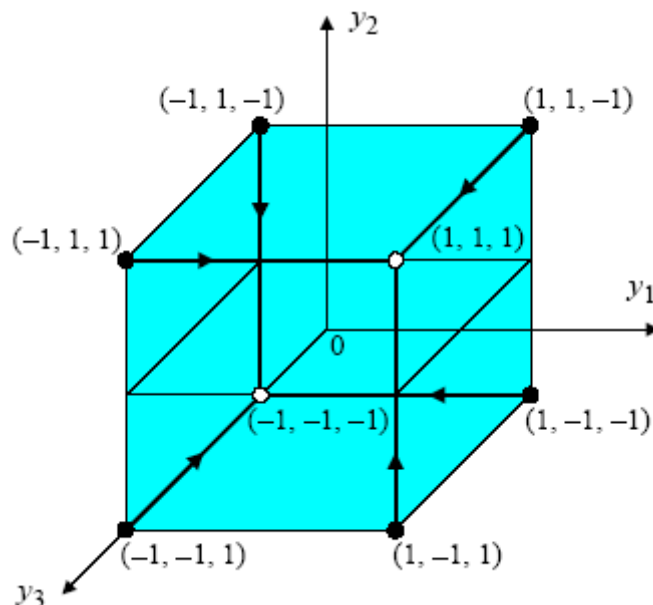
Η τρέχουσα κατάσταση του δικτύου καθορίζεται από τις τρέχουσες εξόδους όλων των νευρώνων y_1, y_2, \dots, y_n του δικτύου. Έτσι, για την περίπτωση ενός δικτύου Hopfield ενός επιπέδου που αποτελείται από n νευρώνες, η κατάσταση περιγράφεται από το ακόλουθο διάνυσμα κατάστασης (state vector):

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (5.4)$$

Στα δίκτυα Hopfield τα βάρη μεταξύ των νευρώνων συνηθίζεται να αναπαρίστανται με τη μορφή πινάκων. Έτσι ο πίνακας των βαρών ενός δικτύου Hopfield υπολογίζεται ως εξής:

$$W = \sum_{m=1}^M Y_m \times Y_m^T - M \times I, \quad (5.5)$$

όπου M είναι ο αριθμός των καταστάσεων που θα απομνημονευθούν από το δίκτυο, Y_m είναι το n -διάστατο δυαδικό διάνυσμα και I είναι ο $(n \times n)$ ταυτοτικός πίνακας (Identity Matrix).



Σχήμα 5.2 Αναπαράσταση των δυνατών καταστάσεων ενός δικτύου Hopfield με τρεις νευρώνες.

Η λειτουργία ενός δικτύου Hopfield μπορεί να αναπαρασταθεί γεωμετρικά. Στο σχήμα 5.2 απεικονίζονται οι δυνατές καταστάσεις που μπορεί να πάρει ένα δίκτυο με 3 νευρώνες και οι οποίες αναπαρίστανται στον τριδιάστατο χώρο ως κύβος. Γενικά ένα δίκτυο Hopfield με n νευρώνες μπορεί να βρεθεί σε 2^n δυνατές καταστάσεις και να αναπαρασταθεί με ένα n -διάστατο υπερ-κύβο. Όπως μπορούμε να δούμε στο σχήμα 5.2, κάθε κατάσταση του δικτύου απεικονίζεται από μια κορυφή του υπερ-κύβου. Όταν εφαρμόζεται ένα νέο διάνυσμα εισόδου, τότε το δίκτυο μετακινείται από μία κατάσταση – κορυφή σε μία άλλη έως ότου σταθεροποιηθεί.

Η σταθερή κατάσταση – κορυφή καθορίζεται από τον πίνακα των βαρών W , το τρέχον διάνυσμα εισόδου X και από τον πίνακα των κατωφλίων θ . Εάν στο δίκτυο εισαγάγουμε ένα διάνυσμα το οποίο είναι μερικώς λανθασμένο ή κατεστραμμένο, τότε το δίκτυο θα συγκλίνει σε μία σταθερή κατάσταση μετά από έναν αριθμό επαναλήψεων. Οι σταθερές καταστάσεις πολλές φορές ονομάζονται και βασικές μνήμες (fundamental memories).

Αφού αποθηκεύσουμε τα διανύσματα που επιθυμούμε να απομνημονεύσει το δίκτυο, στη συνέχεια μπορούμε να το δοκιμάσουμε. Αυτό γίνεται ως εξής: αρχικά ενεργοποιούμε το δίκτυο εισάγοντας το διάνυσμα εισόδου X . Στη συνέχεια υπολογίζουμε το διάνυσμα εξόδου Y .

$$Y_m = \text{sign}(W \times X_m - \theta), \quad m=1,2, \dots, M, \quad (5.6)$$

όπου θ είναι ο πίνακας των κατωφλίων. Τέλος, συγκρίνουμε το αποτέλεσμα (δηλαδή το διάνυσμα εξόδου Y) με το διάνυσμα X στην είσοδο.

Στη συνέχεια θα παρουσιάσουμε μια σύνοψη του αλγορίθμου εκπαίδευσης των δικτύων Hopfield (Negnevitsky, 2002).

Σύνοψη του αλγορίθμου εκπαίδευσης του δικτύου Hopfield

Βήμα 1ο: Αποθήκευση

Έστω ότι έχουμε ένα δίκτυο Hopfield με n νευρώνες στο οποίο επιθυμούμε να αποθηκεύσουμε ένα σύνολο από M διανύσματα – σταθερές καταστάσεις (βασικές μνήμες): Y_1, Y_2, \dots, Y_M . Τα συναπτικά βάρη από τον νευρώνα i στον νευρώνα j υπολογίζονται ως εξής:

$$w_{ji} = \begin{cases} \sum_{m=1}^M y_{m,i} \cdot y_{m,j}, & i \neq j \\ 0, & i = j \end{cases}, \quad (5.7)$$

όπου $y_{m,i}$ και $y_{m,j}$ είναι τα i -οστό και j -οστό στοιχείο του διανύσματος Y_m , αντίστοιχα. Σε μορφή πίνακα τα συναπτικά βάρη μπορούν να αναπαρασταθούν ως εξής:

$$W = \sum_{m=1}^M Y_m \times Y_m^T - M \times I \quad (5.8)$$

Το δίκτυο Hopfield μπορεί να αποθηκεύσει ένα σύνολο από βασικές μνήμες (διανύσματα) εάν ο πίνακας των βαρών του είναι συμμετρικός και όλα τα στοιχεία της κύριας διαγωνίου του είναι ίσα με το μηδέν· δηλαδή ο πίνακας των βαρών του δικτύου Hopfield θα πρέπει να έχει τη μορφή:

$$W = \begin{bmatrix} 0 & w_{12} & \cdots & w_{1i} & \cdots & w_{1n} \\ w_{21} & 0 & \cdots & w_{2i} & \cdots & w_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{i1} & w_{i2} & \cdots & 0 & \cdots & w_{in} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{ni} & \cdots & 0 \end{bmatrix}, \quad (5.9)$$

όπου $w_{ji} = w_{ij}$.

Στο δίκτυο Hopfield τα βάρη υπολογίζονται μόνο μία φορά, κατά τη φάση της αποθήκευσης. Από τη στιγμή που υπολογιστούν δεν μεταβάλλονται και παραμένουν σταθερά.

Βήμα 2ο: Δοκιμή

Αφού αποθηκεύσουμε τις βασικές μνήμες, στη συνέχεια είναι απαραίτητο να επιβεβαιώσουμε ότι το δίκτυο Hopfield είναι ικανό να ανακαλέσει από τη «μνήμη» όλες τις βασικές μνήμες. Με άλλα λόγια, το δίκτυο θα πρέπει να είναι ικανό να ανακαλέσει κάθε διάνυσμα (βασική μνήμη) Y_m , όταν αυτό παρουσιάζεται στην είσοδό του· δηλαδή:

$$x_{m,i} = y_{m,i}, \quad i=1,2, \dots, n \text{ και } m=1,2, \dots, M$$

$$y_{m,j} = \text{sign} \left(\sum_{i=1}^n w_{ji} \cdot x_{m,i} - \theta_j \right), \quad (5.10)$$

όπου το $y_{m,j}$ είναι το j -στό στοιχείο του διανύσματος εξόδου Y_m και το $x_{m,i}$ είναι το i -οστό στοιχείο του διανύσματος εισόδου X_m . Οι παραπάνω σχέσεις σε μορφή πινάκων γίνονται:

$$\mathbf{X}_m = \mathbf{Y}_m, \quad m=1,2, \dots, M$$

$$\mathbf{Y}_m = \text{sign}(\mathbf{W} \times \mathbf{X}_m - \boldsymbol{\theta}) \quad (5.11)$$

Εάν όλες οι βασικές μνήμες ανακαλούνται τέλεια, τότε μπορούμε να συνεχίσουμε με το επόμενο βήμα.

Βήμα 3ο: Ανάκτηση

Παρουσιάζουμε ένα άγνωστο n -διάστατο διάνυσμα X στο δίκτυο και ανακαλούμε μια σταθερή κατάσταση. Συνήθως το διάνυσμα αυτό είναι μία «φθαρμένη» ή ατελής εκδοχή μιας βασικής μνήμης· δηλαδή:

$$\mathbf{X} \neq \mathbf{Y}_m, \quad m=1,2, \dots, M$$

A. Αρχικοποιούμε τον αλγόριθμο ανάκλησης του δικτύου Hopfield θέτοντας:

$$x_i(0) = x_i, \quad i=1,2, \dots, n \quad (5.12)$$

και υπολογίζουμε την αρχική κατάσταση κάθε νευρώνα:

$$y_j(0) = \text{sign} \left(\sum_{i=1}^n w_{ji} \cdot x_i(0) - \theta_j \right), \quad j=1,2, \dots, n \quad (5.13)$$

όπου $x_i(0)$ είναι το i -οστό στοιχείο του διανύσματος X στην επανάληψη $p=0$, και $y_j(0)$ είναι η κατάσταση του νευρώνα j στην επανάληψη $p=0$.

Σε μορφή πινάκων το διάνυσμα κατάστασης στην επανάληψη $p=0$ υπολογίζεται ως εξής:

$$\mathbf{Y}(0) = \text{sign}(\mathbf{W} \times \mathbf{X}(0) - \boldsymbol{\theta}) \quad (5.14)$$

όπου $\mathbf{X}(0)=\mathbf{X}$.

B. Ενημερώνουμε τα στοιχεία του διανύσματος κατάστασης $Y(p)$, σύμφωνα με τον ακόλουθο κανόνα:

$$y_j(p+1) = \text{sign} \left(\sum_{i=1}^n w_{ji} \cdot x_i(p) - \theta_j \right), \quad j=1,2, \dots, n \quad (5.15)$$

όπου $x_i(p) = y_i(p)$, $i=1,2, \dots, n$

Σε μορφή πινάκων η σχέση γίνεται:

$$\mathbf{Y}(p+1) = \text{sign}(\mathbf{W} \times \mathbf{X}(p) - \boldsymbol{\theta}), \quad (5.16)$$

όπου $\mathbf{X}(p)=\mathbf{Y}(p)$

Οι νευρώνες προς ενημέρωση επιλέγονται ασύγχρονα, δηλαδή τυχαία και ένας κάθε φορά.

Η διαδικασία επαναλαμβάνεται έως ότου το διάνυσμα κατάστασης πάψει να μεταβάλλεται, δηλαδή έως ότου επιτευχθεί μια σταθερή κατάσταση. Η συνθήκη για την ισορροπία (ευστάθεια – σταθερότητα) του δικτύου μπορεί να ορισθεί ως εξής:

$$y_j(p+1) = y_j(p), j=1,2, \dots, n \quad (5.17)$$

ή σε μορφή πίνακα ως εξής:

$$\mathbf{Y}(\mathbf{p} + \mathbf{1}) = \mathbf{Y}(\mathbf{p}) \quad (5.18)$$

Περιορισμοί των δικτύων Hopfield

Το δίκτυο Hopfield θα συγκλίνει πάντα σε μια σταθερή κατάσταση (κατάσταση ισορροπίας) εάν η ανάκτηση γίνεται ασύγχρονα (Haykin, 1994). Ωστόσο, τίποτα δεν εξασφαλίζει ότι αυτή η σταθερή κατάσταση θα είναι μία από τις βασικές μνήμες που έχουν αποθηκευτεί στο δίκτυο και ότι —ακόμα και αν είναι— θα είναι αναγκαστικά η πλησιέστερη προς το διάνυσμα εισόδου.

Παράδειγμα

Έστω ότι θέλουμε να αποθηκεύσουμε τις παρακάτω βασικές μνήμες σε ένα δίκτυο Hopfield με 5 νευρώνες.

$$X_1 = [+1 \quad +1 \quad +1 \quad +1 \quad +1]^T$$

$$X_2 = [+1 \quad -1 \quad +1 \quad -1 \quad +1]^T$$

$$X_3 = [-1 \quad +1 \quad -1 \quad +1 \quad -1]^T$$

Χρησιμοποιώντας την εξίσωση

$$\mathbf{W} = \sum_{m=1}^3 \mathbf{X}_m \times \mathbf{X}_m^T - \mathbf{M} \times \mathbf{I} = \mathbf{X}_1 \times \mathbf{X}_1^T + \mathbf{X}_2 \times \mathbf{X}_2^T + \mathbf{X}_3 \times \mathbf{X}_3^T - \mathbf{M} \times \mathbf{I}$$

υπολογίζουμε τον πίνακα των βαρών που είναι:

$$W = \begin{bmatrix} 0 & -1 & 3 & -1 & 3 \\ -1 & 0 & -1 & 3 & -1 \\ 3 & -1 & 0 & -1 & 3 \\ -1 & 3 & -1 & 0 & -1 \\ 3 & -1 & 3 & -1 & 0 \end{bmatrix}$$

Έστω ότι στην είσοδο του δικτύου παρουσιάζεται το διάνυσμα:

$$X = [+1 \quad +1 \quad -1 \quad +1 \quad +1]^T$$

Συγκρίνοντας το διάνυσμα X με τη βασική μνήμη X₁ παρατηρούμε ότι διαφέρουν μόλις σε ένα ψηφίο. Οπότε αναμένουμε ότι εισάγοντας το διάνυσμα X στο δίκτυο Hopfield, αυτό θα συγκλίνει στην κατάσταση X₁. Ωστόσο, εφαρμόζοντας τον αλγόριθμο εκπαίδευσης που περιγράψαμε παραπάνω, παρατηρούμε ότι το δίκτυο ανακαλεί τη βασική μνήμη X₃, η οποία διαφέρει σε 2 ψηφία από το X και όχι το X₁ όπως αναμενόταν.

Το παραπάνω παράδειγμα αποκάλυψε ένα από τα προβλήματα που είναι εγγενή στα δίκτυα Hopfield.

Ένα άλλο πρόβλημα που αντιμετωπίζουμε όταν εφαρμόζουμε τα δίκτυα Hopfield είναι η αποθηκευτική χωρητικότητα (storage capacity), ο μέγιστος δηλαδή αριθμός διανυσμάτων που μπορούν να αποθηκευτούν και να ανακληθούν χωρίς σφάλμα. Ο Hopfield έδειξε πειραματικά [Hopfield, 1982] ότι ο μέγιστος αριθμός βασικών μνημών M_{\max} που μπορούν να αποθηκευτούν σε ένα αναδρομικό δίκτυο με n νευρώνες δίνεται από τη σχέση (δεν μας ενδιαφέρει αν στην ανάκληση έχουμε λάθη):

$$M_{\max} = 0.15 \cdot n \quad (5.19)$$

Η μέγιστη αποθηκευτική χωρητικότητα των δικτύων Hopfield μπορεί να ορισθεί και στη βάση της επιθυμίας οι περισσότερες από τις βασικές μνήμες να μπορούν να ανακληθούν χωρίς σφάλμα (Amit, 1989; Negnevitsky, 2002):

$$M_{\max} = \frac{n}{2 \cdot \ln n} \quad (5.20)$$

Στην περίπτωση ωστόσο που επιθυμούμε όλες οι βασικές μνήμες, που έχουν αποθηκευτεί σε ένα δίκτυο Hopfield με n νευρώνες, να μπορούν να ανακληθούν τέλεια (χωρίς κανένα σφάλμα), τότε το M_{\max} γίνεται (Amit, 1989, Negnevitsky, 2002):

$$M_{\max} = \frac{n}{4 \cdot \ln n} \quad (5.21)$$

Παρατηρούμε λοιπόν ότι η αποθηκευτική χωρητικότητα ενός δικτύου Hopfield πρέπει να κρατιέται αρκετά χαμηλά προκειμένου να μπορούμε να ανακαλέσουμε χωρίς σφάλμα τις βασικές μνήμες. Αυτό αποτελεί έναν πολύ σημαντικό περιορισμό των δικτύων Hopfield.

Από τη συζήτηση που κάναμε έως τώρα παρατηρούμε ότι ένα δίκτυο Hopfield αντιπροσωπεύει βασικά έναν τύπο αυτοσυσχετιστικής (auto-associative) μνήμης: με άλλα λόγια ένα δίκτυο Hopfield μπορεί να ανακαλέσει μια φθαρμένη ή ατελή μνήμη, αλλά δεν μπορεί να συσχετίσει μια μνήμη με κάποια άλλη διαφορετική. Σε αντίθεση με τα δίκτυα Hopfield η ανθρώπινη μνήμη είναι βασικά συσχετιστική: ένα πράγμα μπορεί να μας θυμίσει κάποιο άλλο και αυτό κάποιο άλλο κ.ο.κ. Αυτή τη λειτουργία την επιτελεί μια άλλη κατηγορία αναδρομικών δικτύων που αποκαλείται Συσχετιστική Μνήμη Διπλής Κατεύθυνσης (Bidirectional Associative Memory) (Kosko, 1987, 1988) και η οποία βασίζεται στη φιλοσοφία λειτουργίας των δικτύων Hopfield.

Άσκηση 5.1:

Έστω ότι θέλουμε να αποθηκεύσουμε τα ακόλουθα διανύσματα (βασικές μνήμες):

$$\mathbf{X}_1 = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix} \text{ και } \mathbf{X}_2 = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

σε ένα δίκτυο Hopfield. Θεωρήστε ότι τα κατώφλια είναι ίσα με το μηδέν.

- Ποια πρέπει να είναι η αρχιτεκτονική του δικτύου Hopfield που θα χρησιμοποιηθεί;
- Υπολογίστε τον πίνακα των βαρών αυτού του δικτύου.
- Εξακριβώστε αν τα δύο διανύσματα έχουν αποθηκευτεί σωστά.

Άσκηση 5.2:

Θεωρήστε και πάλι το δίκτυο Hopfield της προηγούμενης άσκησης όπου έχουν αποθηκευτεί τα δύο διανύσματα (βασικές μνήμες):

$$\mathbf{X}_1 = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix} \text{ και } \mathbf{X}_2 = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

Όπως γνωρίζουμε ο συνολικός αριθμός διανυσμάτων τριών ψηφίων είναι $2^3=8$. Τροφοδοτήστε λοιπόν το δίκτυο με το σύνολο των διανυσμάτων τριών ψηφίων και υπολογίστε την κατάσταση του δικτύου για καθένα από αυτά. Τι παρατηρείτε;

Άσκηση 5.3:

Θεωρήστε και πάλι το παράδειγμα που περιγράψαμε συνοπτικά παραπάνω· δηλαδή έστω ότι θέλουμε να αποθηκεύσουμε τις παρακάτω βασικές μνήμες σε ένα δίκτυο Hopfield με 5 νευρώνες.

$$X_1 = [+1 \ +1 \ +1 \ +1 \ +1]^T$$

$$X_2 = [+1 \ -1 \ +1 \ -1 \ +1]^T$$

$$X_3 = [-1 \ +1 \ -1 \ +1 \ -1]^T$$

Θεωρήστε ότι τα κατώφλια είναι ίσα με το μηδέν.

- Εφαρμόστε τα βήματα του αλγορίθμου εκπαίδευσης του δικτύου Hopfield προκειμένου να αποθηκεύσετε τα παραπάνω διανύσματα.
- Προσπαθήστε να ανακαλέσετε τη βασική μνήμη που αντιστοιχεί στο φθαρμένο διάνυσμα $X = [+1 \ +1 \ -1 \ +1 \ +1]^T$ και επαληθεύστε το αποτέλεσμα του παραδείγματος.
- Ποιος είναι ο μέγιστος αριθμός διανυσμάτων που μπορούμε να αποθηκεύσουμε σε ένα τέτοιο δίκτυο Hopfield με 5 νευρώνες (δεν ενδιαφερόμαστε για την ικανότητα ανάκλασης);
- Ποιος είναι ο μέγιστος αριθμός διανυσμάτων που μπορούμε να αποθηκεύσουμε στο δίκτυο αυτό, εάν επιθυμούμε τα αποθηκευμένα διανύσματα (βασικές μνήμες) να ανακαλούνται τέλεια — χωρίς κανένα σφάλμα;

5.3 Δίκτυα Kohonen

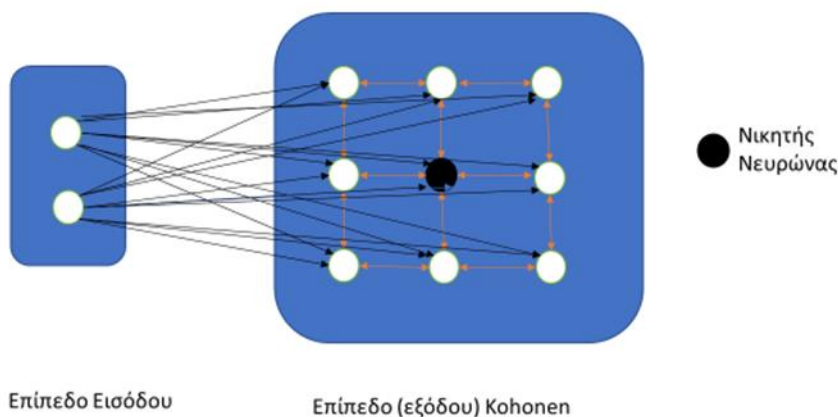
Τι είναι ο Χάρτης Αυτοοργάνωσης Χαρακτηριστικών (Self-Organizing Feature Map);

Ο ανθρώπινος εγκέφαλος «κυριαρχείται» από τον εγκεφαλικό φλοιό, ο οποίος είναι μια εξαιρετικά πολύπλοκη δομή από δισεκατομμύρια νευρώνες και εκατοντάδες δισεκατομμύρια συνάψεις. Ο φλοιός στην οργάνωσή του δεν παρουσιάζει ούτε ομοιομορφία ούτε και ομοιογένεια. Περιλαμβάνει περιοχές που χαρακτηρίζονται από το «πάχος» των επιπέδων τους και τον τύπο των νευρώνων που υπάρχουν σε αυτές. Καθεμία από τις περιοχές αυτές είναι υπεύθυνη και για μια διαφορετική ανθρώπινη λειτουργία, όπως π.χ. την όραση, την κίνηση, την ακοή κλπ. και καθεμία σχετίζεται και με διαφορετικά αισθητήρια όργανα· μπορούμε να πούμε λοιπόν ότι κάθε αισθητήριο όργανο απεικονίζεται σε μια αντίστοιχη περιοχή στον εγκεφαλικό φλοιό. Με άλλα λόγια ο φλοιός είναι ένας αυτο-οργανωμένος υπολογιστικός χάρτης (self-organizing computational map) στον ανθρώπινο εγκέφαλο.

Μοντελοποίηση των Χαρτών Αυτοοργάνωσης

Ο Kohonen διατύπωσε την αρχή του σχηματισμού τοπογραφικού χάρτη (principle of topographic map formation) (Kohonen, 1990). Σύμφωνα με την αρχή αυτή, η θέση ενός νευρώνα εξόδου σε έναν τοπογραφικό χάρτη αντιστοιχεί σε ένα συγκεκριμένο χαρακτηριστικό του προτύπου εισόδου. Ο Kohonen πρότεινε επίσης το μοντέλο της αντιστοίχισης των χαρακτηριστικών, το οποίο φαίνεται στο σχήμα 6.1 (Kohonen, 1982). Το

μοντέλο αυτό «συλλαμβάνει» τα βασικά χαρακτηριστικά των αυτο-οργανωμένων χαρτών του εγκεφάλου, παρ' όλα αυτά μπορεί εύκολα να αναπαρασταθεί και να υλοποιηθεί και σε έναν υπολογιστή.



Σχήμα 5.3 Το δίκτυο Kohonen.

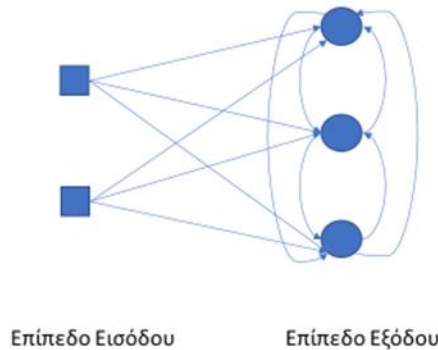
Το μοντέλο του Kohonen κάνει μια τοπογραφική απεικόνιση ενός συγκεκριμένου – σταθερού αριθμού προτύπων εισόδου (που παρουσιάζονται στο επίπεδο εισόδου) σε ένα μεγαλύτερης διάστασης επίπεδο εξόδου, το οποίο ονομάζεται και επίπεδο Kohonen. Π.χ. στο σχήμα 5.3 το επίπεδο Kohonen αποτελείται από ένα πλέγμα που σχηματίζεται από 4x4 νευρώνες, με κάθε νευρώνα να έχει δύο εισόδους. Ο νικητής νευρώνας απεικονίζεται με μαύρο χρώμα, ενώ οι γείτονές του με γκρι. Οι γείτονες νευρώνες του νικητή νευρώνα είναι όσοι βρίσκονται σε κοντινή φυσική γειτνίαση με αυτόν.

Το πόσο κοντινή είναι αυτή η φυσική γειτνίαση εξαρτάται από τον σχεδιαστή του δικτύου. Έτσι η γειτονιά του νικητή νευρώνα μπορεί να περιλαμβάνει νευρώνες που βρίσκονται σε ακτίνα ενός, δύο ή τριών θέσεων από αυτόν. Π.χ. στο σχήμα 5.3 η γειτονιά του νικητή νευρώνα έχει ακτίνα (ή μέγεθος) 1. Γενικά κατά την εκπαίδευση ενός δικτύου Kohonen ξεκινάμε με μια γειτονιά μεγάλου μεγέθους (μεγάλη ακτίνα) και όσο προχωρά η διαδικασία της εκπαίδευσης, το μέγεθος της γειτονιάς βαθμιαία μικραίνει. Αυτό μπορεί να γίνει με μια ποικιλία μεθόδων.

Το δίκτυο Kohonen αποτελείται από το επίπεδο εισόδου (οι νευρώνες εισόδου δεν κάνουν καμία επεξεργασία) και από ένα επίπεδο υπολογιστικών νευρώνων, το επίπεδο Kohonen. Στο δίκτυο Kohonen επιπλέον υπάρχουν δύο είδη συνάψεων – συνδέσεων:

1. οι συνδέσεις προς τα εμπρός (forward connections) από τους νευρώνες εισόδου προς τους νευρώνες του επιπέδου Kohonen και
2. οι παράπλευρες συνδέσεις (lateral connections) μεταξύ των νευρώνων του επιπέδου εξόδου (επιπέδου Kohonen).

Αυτό απεικονίζεται στο σχήμα 5.4. Οι παράπλευρες συνδέσεις χρησιμοποιούνται για να δημιουργήσουν τον ανταγωνισμό μεταξύ των νευρώνων του επιπέδου Kohonen. Ο νευρώνας με το μεγαλύτερο δυναμικό ενεργοποίησης μεταξύ όλων των νευρώνων του επιπέδου εξόδου «στέφεται» νικητής (winner-takes-all neuron). Αυτός είναι ο μόνος που παράγει έξοδο. Η δραστηριότητα όλων των άλλων νευρώνων καταπνίγεται μέσω του ανταγωνισμού.



Σχήμα 5.4 Το δίκτυο Kohonen.

Όταν παρουσιάζεται στις εισόδους του δικτύου ένα δείγμα, κάθε νευρώνας στο επίπεδο Kohonen λαμβάνει ένα πλήρες αντίγραφο του, τροποποιημένο φυσικά από τις αντίστοιχες τιμές των βαρών στις συνδέσεις μεταξύ του επιπέδου εισόδου και του επιπέδου εξόδου. Οι παράπλευρες συνδέσεις έχουν είτε διεγερτική είτε κατασταλτική δράση ανάλογα με την απόστασή τους από τον νικητή νευρώνα. Αυτό μπορεί να επιτευχθεί με τη χρήση της συνάρτησης Mexican Hat, η οποία περιγράφει τα παράπλευρα συναπτικά βάρη μεταξύ των νευρώνων στο επίπεδο Kohonen.

Η συνάρτηση Mexican Hat (Amit, 1989) υπολογίζει τις τιμές των βαρών των παράπλευρων συνδέσεων συναρτήσει της απόστασης των νευρώνων από τον νικητή νευρώνα. Έτσι, σύμφωνα με τη συνάρτηση αυτή, η κοντινή γειτονιά του νικητή νευρώνα έχει ισχυρή διεγερτική επίδραση, η απομακρυσμένη γειτονιά έχει μια ήπια ανασταλτική επίδραση και η πολύ απομακρυσμένη γειτονιά έχει μια αδύναμη διεγερτική επίδραση που συνήθως αγνοείται.

Στο δίκτυο Kohonen ένας νευρώνας μαθαίνει μετατοπίζοντας τα βάρη του από τις μη ενεργές συνδέσεις προς τις ενεργές. Μόνο ο νικητής νευρώνας και η γειτονιά του έχουν το δικαίωμα να μαθαίνουν. Εάν ένας νευρώνας δεν ανταποκρίνεται στο δείγμα στην είσοδο, τότε δεν μπορεί να «μάθει». Η έξοδος του νικητή νευρώνα είναι ίση με το 1 και όλων των υπολοίπων με το 0.

Ο **στάνταρντ κανόνας ανταγωνιστικής μάθησης (standard competitive learning rule)** (Haykin, 1994) καθορίζει τη μεταβολή Δw_{ij} του συναπτικού βάρους w_{ij} σύμφωνα με τη σχέση:

$$\Delta w_{ji} = \begin{cases} \alpha \cdot (x_i - w_{ji}), & \text{εάν ο νευρώνας } j \text{ κερδίσει το διαγωνισμό} \\ 0, & \text{εάν ο νευρώνας } j \text{ χάσει το διαγωνισμό} \end{cases} \quad (5.22)$$

όπου x_i είναι το σήμα στον νευρώνα εισόδου i και το α είναι ο ρυθμός μάθησης που παίρνει τιμές μεταξύ 0 και 1.

Η συνολική επίδραση λοιπόν του κανόνα της ανταγωνιστικής μάθησης είναι να «μετακινεί» το διάνυσμα των βαρών του νικητή νευρώνα \mathbf{w}_j προς το διάνυσμα εισόδου \mathbf{x} .

Πώς όμως καθορίζεται ποιος είναι ο νικητής νευρώνας; Όταν παρουσιάζεται ένα διάνυσμα στην είσοδο του Kohonen, υπολογίζεται πόσο «ταιριάζει» το διάνυσμα των βαρών κάθε νευρώνα του επιπέδου Kohonen με το διάνυσμα αυτό. Ο νευρώνας του οποίου το διάνυσμα των βαρών «ταιριάζει» περισσότερο με το διάνυσμα εισόδου είναι ο νικητής. Η εκτίμηση του «ταιριάσματος» μπορεί να γίνει με διάφορους τρόπους, ο πιο

συνηθισμένος είναι με τον υπολογισμό της Ευκλείδειας απόστασης (Euclidean Distance)· στην περίπτωση αυτή δηλαδή νικητής είναι ο νευρώνας του οποίου το διάνυσμα των βαρών έχει τη μικρότερη Ευκλείδεια απόσταση από το διάνυσμα εισόδου.

Η Ευκλείδεια απόσταση μεταξύ δύο διανυσμάτων \mathbf{x} και \mathbf{w}_j διάστασης n δίνεται από τη σχέση:

$$d = \|\mathbf{x} - \mathbf{w}_j\| = \left[\sum_{i=1}^n (x_i - w_{ji})^2 \right]^{1/2} \quad (5.23)$$

όπου x_i και w_{ji} είναι τα i -οστά στοιχεία των διανυσμάτων \mathbf{x} και \mathbf{w}_j αντίστοιχα.

Έτσι, στην περίπτωση αυτή, ο νικητής νευρώνας j_x που ταιριάζει καλύτερα με το διάνυσμα εισόδου \mathbf{x} είναι αυτός με τη μικρότερη Ευκλείδεια απόσταση, δηλαδή (Haykin, 1994):

$$j_x = \min_j \|\mathbf{x} - \mathbf{w}_j\|, \quad j=1,2, \dots, m \quad (5.24)$$

όπου m είναι ο αριθμός των νευρώνων στο επίπεδο Kohonen (ή επίπεδο εξόδου).

Σύνοψη του αλγορίθμου εκπαίδευσης του δικτύου Kohonen

Στη συνέχεια θα παρουσιάσουμε μια σύνοψη του αλγορίθμου εκπαίδευσης του δικτύου του Kohonen [Kohonen 1989]:

Βήμα 1ο: Αρχικοποίηση

Αρχικοποίησε τα συναπτικά βάρη δίνοντάς τους μικρές τυχαίες αρχικές τιμές, για παράδειγμα στο διάστημα $[0, 1]$ (με ομοιόμορφη κατανομή). Αρχικοποίησε τον ρυθμό μάθησης α δίνοντάς του μια μικρή θετική τιμή.

Βήμα 2ο: Ενεργοποίηση

Ενεργοποίησε το δίκτυο Kohonen εφαρμόζοντας στην είσοδό του ένα διάνυσμα εκπαίδευσης \mathbf{x} και βρίσκοντας τον νικητή νευρώνα (winner-takes-all neuron) j_x για την επανάληψη p , χρησιμοποιώντας ως κριτήριο την ελάχιστη Ευκλείδεια απόσταση:

$$j_x(p) = \min_j \|\mathbf{x} - \mathbf{w}_j(p)\| = \left\{ \sum_{i=1}^n [x_i - w_{ji}(p)]^2 \right\}^{1/2}, \quad j=1,2, \dots, m \quad (5.25)$$

όπου n είναι ο αριθμός των νευρώνων στο επίπεδο εισόδου και m είναι ο αριθμός των νευρώνων στο επίπεδο εξόδου ή επίπεδο Kohonen.

Βήμα 3ο: Εκπαίδευση

Ενημέρωσε τις τιμές των βαρών σύμφωνα με τη σχέση:

$$w_{ji}(p+1) = w_{ji}(p) + \Delta w_{ji}(p) \quad (5.26)$$

όπου $\Delta w_{ji}(p)$ είναι η διόρθωση στην τιμή του βάρους στην επανάληψη p .

Η διόρθωση του λάθους $\Delta w_{ji}(p)$ καθορίζεται από τον κανόνα της ανταγωνιστικής μάθησης:

$$\Delta w_{ji}(p) = \begin{cases} \alpha \cdot [x_i - w_{ji}(p)], & j \in \Lambda_j(p) \\ 0, & j \notin \Lambda_j(p) \end{cases} \quad (5.27)$$

όπου a είναι ο ρυθμός μάθησης και $\Lambda_j(p)$ είναι η γειτονιά του νικητή νευρώνα (winner-takes-all-neuron) j_x στην επανάληψη p .

Ο ρυθμός μάθησης a μπορεί:

- είτε να είναι σταθερός καθ' όλη τη διάρκεια της διαδικασίας εκπαίδευσης
- είτε να μεταβάλλεται. Σε αυτήν την περίπτωση συνήθως ξεκινάμε με μια μεγάλη τιμή του ρυθμού μάθησης για $p=0$ και εν συνεχεία, κατά τη διάρκεια της εκπαίδευσης, την ελαττώνουμε με γραμμικό ή μη-γραμμικό τρόπο (Callan, 1999). Μια σχέση που μπορούμε να χρησιμοποιήσουμε, για παράδειγμα, είναι η εξής (Αργυράκης, 2001):

$$\alpha(p) = \alpha(0) \cdot \left(1 - \frac{p}{P}\right) \quad (5.28)$$

όπου $\alpha(0)$ είναι η αρχική τιμή (συνήθεις τιμές για την περίπτωση αυτή είναι στο διάστημα (0.2, 0.5)) του ρυθμού μάθησης, p είναι ο τρέχον κύκλος (επανάληψη) εκπαίδευσης και P είναι ο συνολικός αριθμός των κύκλων εκπαίδευσης του δικτύου Kohonen.

Η γειτονιά Λ_j του νικητή νευρώνα μπορεί:

- να είναι σταθερή καθ' όλη την εκπαιδευτική διαδικασία,
- να ξεκινά η εκπαίδευση με μια μεγάλη γειτονιά γύρω από τον νικητή νευρώνα, η οποία στη συνέχεια να ελαττώνεται όσο προχωρούν οι κύκλοι εκπαίδευσης μέχρις ότου να πάρει τελικά την τιμή 1, να περιλαμβάνει δηλαδή μόνο τον νικητή νευρώνα. Στην περίπτωση αυτή δίνουμε στη γειτονιά ένα αρχικό πλάτος $d(0)$ (συνήθως θα πρέπει να είναι περίπου το $\frac{1}{2}$ του μεγέθους του επιπέδου Kohonen), το οποίο και μειώνουμε κατά τη διάρκεια της εκπαίδευσης. Αυτή η μείωση μπορεί να γίνει με γραμμικό και μη-γραμμικό τρόπο. Μια σχέση, για παράδειγμα, που μπορούμε να χρησιμοποιήσουμε είναι η ακόλουθη (Αργυράκης, 2001):

$$d(p) = d(0) \cdot \left(1 - \frac{p}{P}\right) \quad (5.29)$$

όπου $d(0)$ είναι το αρχικό πλάτος (ακτίνα) της γειτονιάς, p είναι ο τρέχον κύκλος εκπαίδευσης και P είναι ο συνολικός αριθμός των κύκλων εκπαίδευσης του δικτύου Kohonen.

Όλοι οι νευρώνες που βρίσκονται μέσα στην τοπολογική γειτονιά ενεργοποιούνται ταυτόχρονα και η σχέση ανάμεσά τους είναι ανεξάρτητη από την απόστασή τους από τον νικητή νευρώνα j_x .

Για τους νευρώνες του επιπέδου Kohonen (επίπεδο εξόδου) η έξοδός τους δίνεται από τη σχέση:

$$y_j(p) = \begin{cases} 1, & j \in \Lambda_j(p) \\ 0, & j \notin \Lambda_j(p) \end{cases} \quad (5.30)$$

Βήμα 4ο: Επανάληψη

Αύξησε τον μετρητή επανάληψης p κατά 1, πήγαινε στο βήμα 2 και συνέχισε την εκπαίδευση μέχρις ότου να ικανοποιηθεί το κριτήριο της ελάχιστης Ευκλείδεια απόστασης ή μέχρις ότου να μη συμβαίνει κάποια αξιοσημείωτη μεταβολή στον χάρτη χαρακτηριστικών (feature map).

5.3.1 Εκθετική μείωση του εύρους της γειτονιάς

Μια παραλλαγή του παραπάνω αλγορίθμου περιγράφεται στη συνέχεια. Όπως είδαμε από τη σχέση (5.29), το εύρος της γειτονιάς μειώνεται γραμμικά, ως συνάρτηση του αριθμού των επαναλήψεων. Αν d_{ij} είναι η πλευρική απόσταση μεταξύ των νευρώνων i και j και υποθέσουμε ότι ο νευρώνας i είναι ο νικητής και βρίσκεται στο κέντρο της γειτονιάς, αν με Λ_{ij} συμβολίζουμε την τοπολογική γειτονιά γύρω από τον νευρώνα i τότε το Λ_{ij} είναι μια μονότονη συνάρτηση της απόστασης d_{ij} .

Μια τυπική επιλογή της Λ_{ij} είναι η Γκαουσιανή συνάρτηση, η οποία είναι ανεξάρτητη από τη θέση του νικητή νευρώνα και δίνεται από τη σχέση:

$$\Lambda_{ij} = \exp\left(-\frac{d_{ij}^2}{2\sigma^2}\right) \quad (5.31)$$

Η παράμετρος σ είναι το ενεργό εύρος της γειτονιάς. Αυτή η παράμετρος μετρά σε ποιον βαθμό οι διεγερμένοι νευρώνες στη γειτονιά του νικητή νευρώνα συμμετέχουν στη διαδικασία μάθησης. Η απόσταση d_{ij} μεταξύ των νευρώνων είναι η Ευκλείδεια απόσταση.

Ένα άλλο χαρακτηριστικό του χάρτη αυτοοργάνωσης (SOM), όπως προαναφέραμε, είναι ότι το μέγεθος της γειτονιάς συρρικνώνεται με την πάροδο του χρόνου. Μια προφανής επιλογή εκθετικής μείωσης του εύρους είναι η σχέση:

$$\sigma(p) = \sigma_0 \exp\left(\frac{p}{\tau_1}\right) \quad p = 0, 1, 2, \dots \quad (5.32)$$

όπου σ_0 είναι η αρχική τιμή του σ , κατά την αρχικοποίηση του SOM, και τ_1 είναι μια σταθερά χρόνου. Αντικαθιστώντας την (5.32) στην (5.31) προκύπτει ο υπολογισμός του Λ_{ij} ως συνάρτηση του p , δηλαδή:

$$\Lambda_{ij}(p) = \exp\left(-\frac{d_{ij}^2}{2\sigma(p)^2}\right) \quad p = 0, 1, 2, \dots \quad (5.33)$$

Από την παραπάνω σχέση προκύπτει ότι καθώς αυξάνει ο χρόνος (δηλαδή ο αριθμός επαναλήψεων), το εύρος μειώνεται εκθετικά και η γειτονιά συρρικνώνεται ανάλογα. Η εξίσωση (5.27) ενημέρωσης των βαρών μετασχηματίζεται στην παρακάτω εξίσωση:

$$\Delta W_j = \alpha \Lambda_{ij}(p)(X - W_j) \quad (5.34)$$

και οι ενημερωμένες τιμές των βαρών από τη σχέση:

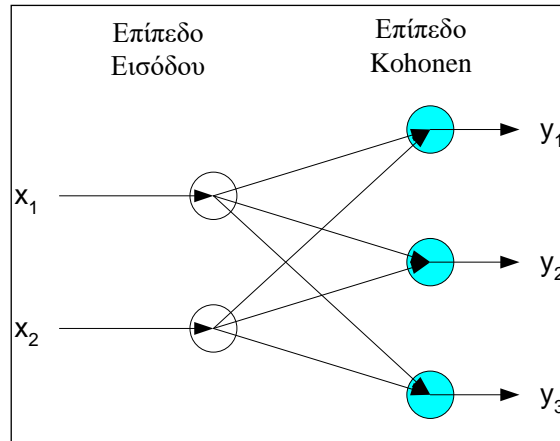
$$W_j(p+1) = W_j(p) + \alpha \Lambda_{ij}(p)(X - W_j(p)) \quad (5.35)$$

Η παραπάνω εξίσωση μετακινεί το διάνυσμα βαρών του νικητή νευρώνα (και των νευρώνων της γειτονιάς) πλησιέστερα στο διάνυσμα εισόδου X , αλλά και οι υπόλοιποι νευρώνες παίρνουν ένα κλάσμα της διόρθωσης.

Θα ολοκληρώσουμε τη συζήτησή μας για τα δίκτυα Kohonen με μερικές ασκήσεις όπου θα δούμε την εφαρμογή των δικτύων σε κάποια πολύ μικρά πρακτικά προβλήματα.

Άσκηση 5.4:

Υποθέστε ότι έχουμε ένα δίκτυο Kohonen με δύο εισόδους και τρεις νευρώνες (στο επίπεδο Kohonen), όπως φαίνεται στο παρακάτω σχήμα.



Σχήμα 5.5 Το δίκτυο Kohonen.

Έστω ότι τα αντίστοιχα βάρη (υποθέστε ότι δεν υπάρχουν συνδέσεις μεταξύ των κρυφών νευρώνων) που συνδέουν τους νευρώνες εισόδου με τους νευρώνες του επιπέδου Kohonen είναι:

$$\mathbf{W}_1 = \begin{bmatrix} 0.27 \\ 0.81 \end{bmatrix} \text{ και } \mathbf{W}_2 = \begin{bmatrix} 0.42 \\ 0.70 \end{bmatrix} \text{ και } \mathbf{W}_3 = \begin{bmatrix} 0.43 \\ 0.21 \end{bmatrix}$$

a. Βρείτε τον νικητή νευρώνα στην περίπτωση που στην είσοδο του δικτύου παρουσιαστεί το διάνυσμα:

$$\mathbf{X} = \begin{bmatrix} 0.52 \\ 0.12 \end{bmatrix}$$

χρησιμοποιώντας το κριτήριο της ελάχιστης Ευκλείδειας απόστασης.

- b. Υπολογίστε τις μεταβολές στα συναπτικά βάρη, υποθέτοντας ότι ο ρυθμός μάθησης α έχει τιμή 0.1.
 c. Υπολογίστε τη νέα τιμή του διανύσματος των βαρών του νικητή νευρώνα.

Άσκηση 5.5:

Θεωρήστε ένα δίκτυο Kohonen με τρεις εισόδους και δύο νευρώνες Kohonen. Θεωρήστε ότι υπάρχουν συνδέσεις μόνο μεταξύ των εισόδων και των νευρώνων Kohonen. Η δομή του δικτύου αυτού απεικονίζεται στο παρακάτω σχήμα (5.6).

Έστω ότι θέλουμε να εκπαιδύσουμε το δίκτυο χρησιμοποιώντας τα ακόλουθα τέσσερα διανύσματα εκπαίδευσης:

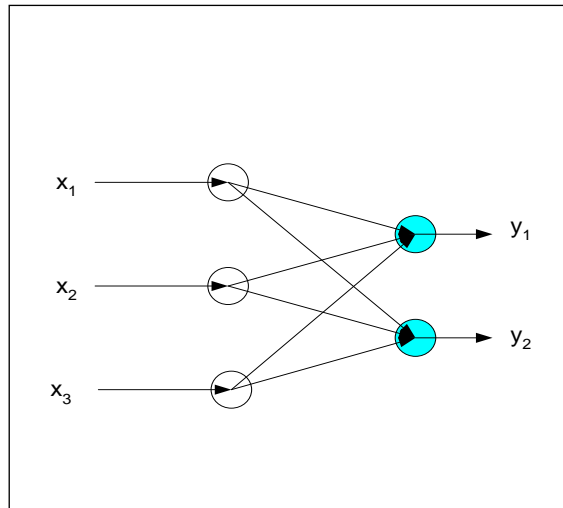
$$\mathbf{X}_1 = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.4 \end{bmatrix}, \mathbf{X}_2 = \begin{bmatrix} 0.6 \\ 0.9 \\ 0.9 \end{bmatrix}, \mathbf{X}_3 = \begin{bmatrix} 0.3 \\ 0.4 \\ 0.1 \end{bmatrix}, \mathbf{X}_4 = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.3 \end{bmatrix}$$

Έστω επίσης ότι τα αρχικά βάρη που ενώνουν τους νευρώνες εισόδου με τους νευρώνες Kohonen είναι:

$$\mathbf{W}_1 = \begin{bmatrix} 0.5 \\ 0.6 \\ 0.8 \end{bmatrix} \text{ και } \mathbf{W}_2 = \begin{bmatrix} 0.4 \\ 0.2 \\ 0.5 \end{bmatrix}$$

Θεωρήστε επίσης ότι η ακτίνα της γειτονιάς του νικητή νευρώνα είναι 0 (η γειτονιά περιλαμβάνει μόνο τον νικητή νευρώνα) και ο ρυθμός μάθησης είναι 0.5.

Υπολογίστε τις μεταβολές και τις τιμές των συναπτικών βαρών κατά τη διάρκεια του πρώτου κύκλου εκπαίδευσης χρησιμοποιώντας τα διανύσματα εισόδου με τη σειρά με την οποία δίνονται.



Σχήμα 5.6 Το δίκτυο Kohonen

Άσκηση 5.6:

Έστω ότι διαθέτουμε δεδομένα για τη συμπεριφορά οχτώ διαφορετικών χρηστών σε ένα σύστημα ηλεκτρονικής μάθησης. Για κάθε χρήστη έχουμε κρατήσει τον μέσο όρο του χρόνου που περνάει συνδεδεμένος στο σύστημα καθημερινά και τον μέσο όρο των μηνυμάτων που ανταλλάσσει με τους άλλους χρήστες. Οι τιμές των δύο αυτών χαρακτηριστικών έχουν κανονικοποιηθεί στο διάστημα $[0-10]$. Θέλουμε με βάση αυτά τα δύο χαρακτηριστικά να ομαδοποιήσουμε τους χρήστες σε τρεις κατηγορίες. Για τα οχτώ προφίλ χρηστών μετά την κανονικοποίηση έχουμε τις εξής τιμές για τα δύο χαρακτηριστικά τους:

$$A1=(2,10), A2=(2,5), A3(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9)$$

Θέλουμε να επιλύσουμε αυτό το πρόβλημα με ένα δίκτυο Kohonen. Τα αρχικά βάρη των νευρώνων Kohonen που θα χρησιμοποιήσετε είναι $W1=A1, W2=A4, W3=A7$.

Θεωρήστε ότι:

- Υπάρχουν συνδέσεις μόνο μεταξύ των εισόδων και των νευρώνων Kohonen.
- Η ακτίνα της γειτονιάς του νικητή νευρώνα είναι πάντα 0 (η γειτονιά περιλαμβάνει δηλαδή μόνο τον νικητή νευρώνα).
- Ο ρυθμός εκπαίδευσης α είναι ίσος με $\alpha=0.5$.
- Κάθε διάνυσμα εισόδου αντιστοιχεί σε έναν κύκλο εκπαίδευσης.

- Πόσοι είσοδοι και πόσοι ανταγωνιστικοί νευρώνες υπάρχουν στο δίκτυο; Σχεδιάστε το δίκτυο με τις αρχικές του τιμές.
- Σχεδιάστε ένα διάγραμμα με άξονες τα δύο χαρακτηριστικά (μέσος όρος σύνδεσης, μέσος όρος μηνυμάτων) στο οποίο να φαίνονται τα σημεία που αντιστοιχούν στον κάθε χρήστη. Με κύκλο να φαίνονται τα σημεία που αντιστοιχούν στα αρχικά βάρη.
- Να εκπαιδεύσετε το δίκτυο για τα διανύσματα εισόδου $A2$ και $A3$ παρατηρώντας κάθε φορά την αλλαγή των βαρών του δικτύου; Τι συμπεραίνετε;
- Με βάση το εκπαιδευμένο δίκτυο σε ποια από τις τρεις ομάδες (ομάδα 1, 2 ή 3 σε αντιστοιχία με τους νευρώνες Kohonen 1, 2 και 3) θα αντιστοιχούσατε τον χρήστη $A5$;

Βιβλιογραφία

- Amit, D.J. (1989). *Modeling Brain Functions: The World of Attractor Neural Networks*. Cambridge University Press.
- Αργυράκης, Π. (2001). *Νευρωνικά Δίκτυα και Εφαρμογές*. Έντυπο Εκπαιδευτικό Υλικό για τη Θεματική Ενότητα «Τεχνητή Νοημοσύνη – Εφαρμογές», Ελληνικό Ανοικτό Πανεπιστήμιο.
- Callan, R. (1999). *The essence of Neural Networks*. Prentice Hall Europe.
- Fausett, L. (1994). *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*. Prentice Hall International.
- Golden, R. (1996). *Mathematical Methods for Neural Network Analysis and Design*. MIT Press.
- Hagan, M., Demuth, H., & Beale M. (1995). *Neural Network Design*. International Thomson Publishing Company.
- Hassoun, M. (1995). *Fundamentals of Artificial neural Networks*. MIT Press.
- Haykin S., (1994). *Neural Networks, A Comprehensive Foundation*. Macmillan College Publishing Company Inc.
- Haykin, S. (1999). *Neural Networks, A Comprehensive Foundation* (2η έκδ.). Prentice Hall International.
- Hopfield, J.J. (1982). Neural Networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the USA*, 79, 2554-2558.
- Hopfield, J.J., & Tank, D.W. (1985). Neural computations of decisions in optimization problems. *Biological Cybernetics*, 52, 141.
- Kohonen, T. (1982). Self-Organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, 59-69.
- Kohonen, T. (1989). *Self-Organization and Associative Memory* (3η έκδ.). Springer-Verlag.
- Kohonen, T. (1989). The self-organizing map. *Proceedings of the IEEE*, 78, 1464-1480.
- Kosko, B. (1987). Adaptive Bidirectional associative memories. *Applied Optics*, 26 (23), 4947-4960.
- Kosko, B. (1988). Bidirectional associative memories. *IEEE Transactions on Systems Man and Cybernetics*, SMC-18, 49-60.
- Kramer, A.H., & Sangiovanni-Vincentelli, A. (1989). Efficient parallel learning algorithms for Neural Networks. Στο *Advances in Neural Information Processing Systems*, τόμ. 1 (σ. 40-48). Morgan Kaufmann.
- Minsky, M.L., & Papert, S.A. (1969). *Perceptrons*. MIT Press.
- Negnevitsky, M. (2002). *Artificial Intelligence: A Guide to Intelligent Systems*. Addison Wesley.
- Patterson, D. (1996). *Artificial Neural Networks, Theory and Applications*. Prentice Hall.
- Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning Representations by Back-Propagating Errors. *Nature*, 323, 533.
- Rumelhart, D.E., & McClelland, J.L. (επιμ.) (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press.

Απαντήσεις ασκήσεων

Άσκηση 5.1:

Έστω ότι θέλουμε να αποθηκεύσουμε τα ακόλουθα διανύσματα (βασικές μνήμες):

$$\mathbf{X}_1 = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix} \text{ και } \mathbf{X}_2 = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

σε ένα δίκτυο Hopfield. Θεωρήστε ότι τα κατώφλια είναι ίσα με το μηδέν.

- Ποια πρέπει να είναι η αρχιτεκτονική του δικτύου Hopfield που θα χρησιμοποιηθεί;
- Υπολογίστε τον πίνακα των βαρών αυτού του δικτύου.
- Εξακριβώστε αν τα δύο διανύσματα έχουν αποθηκευτεί σωστά.

Απάντηση

- Εφόσον τα διανύσματα που θέλουμε να αποθηκεύσουμε έχουν διάσταση 3, το δίκτυο Hopfield που θα χρησιμοποιηθεί θα πρέπει να αποτελείται από 3 νευρώνες.
- Ο πίνακας των βαρών του δικτύου υπολογίζεται ως εξής:

$$\mathbf{W} = \sum_{m=1}^2 \mathbf{X}_m \times \mathbf{X}_m^T - \mathbf{M} \times \mathbf{I} = \mathbf{X}_1 \times \mathbf{X}_1^T + \mathbf{X}_2 \times \mathbf{X}_2^T - 2 \times \mathbf{I} \Rightarrow$$

$$\mathbf{W} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} \cdot \begin{bmatrix} -1 & -1 & -1 \end{bmatrix} - 2 \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{bmatrix}$$

- Στη συνέχεια δοκιμάζουμε αν τα δύο διανύσματα έχουν αποθηκευτεί σωστά. Έτσι έχουμε:

$$\mathbf{Y}_1 = \mathbf{sign}(\mathbf{W} \times \mathbf{X}_1 - \boldsymbol{\theta}) = \mathbf{sign}\left(\begin{bmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{bmatrix} \cdot \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}\right) = \mathbf{sign}\left(\begin{bmatrix} 4 \\ 4 \\ 4 \end{bmatrix}\right) = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix} = \mathbf{X}_1$$

και

$$\mathbf{Y}_2 = \mathbf{sign}(\mathbf{W} \times \mathbf{X}_2 - \boldsymbol{\theta}) = \mathbf{sign}\left(\begin{bmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}\right) = \mathbf{sign}\left(\begin{bmatrix} -4 \\ -4 \\ -4 \end{bmatrix}\right) = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} = \mathbf{X}_2$$

Άσκηση 5.2:

Θεωρήστε και πάλι το δίκτυο Hopfield της προηγούμενης άσκησης όπου έχουν αποθηκευτεί τα δύο διανύσματα (βασικές μνήμες):

$$\mathbf{X}_1 = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix} \text{ και } \mathbf{X}_2 = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

Όπως γνωρίζουμε ο συνολικός αριθμός διανυσμάτων τριών ψηφίων είναι $2^3=8$. Τροφοδοτήστε λοιπόν το δίκτυο με το σύνολο των διανυσμάτων τριών ψηφίων και υπολογίστε την κατάσταση του δικτύου για καθένα από αυτά. Τι παρατηρείτε;

Απάντηση

Στον παρακάτω πίνακα απεικονίζονται τα αποτελέσματα της τροφοδότησης του δικτύου με τα 8 διανύσματα. Για την παραγωγή των αποτελεσμάτων τα διανύσματα δόθηκαν ως είσοδοι και το δίκτυο αφέθηκε να τρέχει μέχρις ότου καταλήξει σε μια σταθερή κατάσταση.

Από τα αποτελέσματα παρατηρούμε ότι οι βασικές μνήμες λειτουργούν ως ελκυστές και προσελκύουν τα διανύσματα τα οποία είναι πιο κοντινά σε αυτές (διαφέρουν κατά ένα ψηφίο). Έτσι η βασική μνήμη $[+1 +1 +1]^T$ προσελκύει τα διανύσματα $[-1 +1 +1]^T$, $[+1 -1 +1]^T$, $[+1 +1 -1]^T$ και η βασική μνήμη $[-1 -1 -1]^T$ προσελκύει τα διανύσματα $[-1 -1 +1]^T$, $[-1 +1 -1]^T$, $[+1 -1 -1]^T$. Βλέπουμε λοιπόν ότι το δίκτυο Hopfield λειτουργεί ως δίκτυο διόρθωσης λάθους (error correction network).

Κατάσταση	Αριθμός Επαναλήψεων	Είσοδοι			Έξοδοι			Βασική Μνήμη
		x1	x2	x3	y1	y2	y3	
$[+1 +1 +1]^T$	0	+1	+1	+1	+1	+1	+1	$[+1 +1 +1]^T$
$[-1 +1 +1]^T$	0	-1	+1	+1	+1	+1	+1	$[+1 +1 +1]^T$
	1	+1	+1	+1	+1	+1	+1	
$[+1 -1 +1]^T$	0	+1	-1	+1	+1	+1	+1	$[+1 +1 +1]^T$
	1	+1	+1	+1	+1	+1	+1	
$[+1 +1 -1]^T$	0	+1	+1	-1	+1	+1	+1	$[+1 +1 +1]^T$
	1	+1	+1	+1	+1	+1	+1	
$[-1 -1 -1]^T$	0	-1	-1	-1	-1	-1	-1	$[-1 -1 -1]^T$
$[-1 -1 +1]^T$	0	-1	-1	+1	-1	-1	-1	$[-1 -1 -1]^T$
	1	-1	-1	-1	-1	-1	-1	
$[-1 +1 -1]^T$	0	-1	+1	-1	-1	-1	-1	$[-1 -1 -1]^T$
	1	-1	-1	-1	-1	-1	-1	
$[+1 -1 -1]^T$	0	+1	-1	-1	-1	-1	-1	$[-1 -1 -1]^T$
	1	-1	-1	-1	-1	-1	-1	

Άσκηση 5.3:

Θεωρήστε και πάλι το παράδειγμα που περιγράψαμε συνοπτικά παραπάνω· δηλαδή έστω ότι θέλουμε να αποθηκεύσουμε τις παρακάτω βασικές μνήμες σε ένα δίκτυο Hopfield με 5 νευρώνες.

$$X_1 = [+1 \ +1 \ +1 \ +1 \ +1]^T$$

$$X_2 = [+1 \ -1 \ +1 \ -1 \ +1]^T$$

$$X_3 = [-1 \ +1 \ -1 \ +1 \ -1]^T$$

Θεωρήστε ότι τα κατώφλια είναι ίσα με το μηδέν.

- Εφαρμόστε τα βήματα του αλγορίθμου εκπαίδευσης του δικτύου Hopfield προκειμένου να αποθηκεύσετε τα παραπάνω διανύσματα.
- Προσπαθήστε να ανακαλέσετε τη βασική μνήμη που αντιστοιχεί στο φθαρμένο διάνυσμα $X = [+1 \ +1 \ -1 \ +1 \ +1]^T$ και επαληθεύστε το αποτέλεσμα του παραδείγματος.
- Ποιος είναι ο μέγιστος αριθμός διανυσμάτων που μπορούμε να αποθηκεύσουμε σε ένα τέτοιο δίκτυο Hopfield με 5 νευρώνες (δεν ενδιαφερόμαστε για την ικανότητα ανάκλασης);
- Ποιος είναι ο μέγιστος αριθμός διανυσμάτων που μπορούμε να αποθηκεύσουμε στο δίκτυο αυτό εάν επιθυμούμε τα αποθηκευμένα διανύσματα (βασικές μνήμες) να ανακαλούνται τέλεια – χωρίς κανένα σφάλμα;

Απάντηση

α. Αρχικά υπολογίζουμε τον πίνακα των βαρών του δικτύου Hopfield:

$$\mathbf{W} = \sum_{m=1}^3 \mathbf{X}_m \times \mathbf{X}_m^T - \mathbf{M} \times \mathbf{I} = \mathbf{X}_1 \times \mathbf{X}_1^T + \mathbf{X}_2 \times \mathbf{X}_2^T + \mathbf{X}_3 \times \mathbf{X}_3^T - 3 \times \mathbf{I} \Rightarrow$$

$$\mathbf{W} = \begin{bmatrix} +1 \\ +1 \\ +1 \\ +1 \\ +1 \end{bmatrix} \cdot [+1 \ +1 \ +1 \ +1 \ +1] + \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \\ +1 \end{bmatrix} \cdot [+1 \ -1 \ +1 \ -1 \ +1] +$$

$$+ \begin{bmatrix} -1 \\ +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} \cdot [-1 \ +1 \ -1 \ +1 \ -1] - 3 \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow$$

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & +3 & -1 & +3 \\ -1 & 0 & -1 & +3 & -1 \\ +3 & -1 & 0 & -1 & +3 \\ -1 & +3 & -1 & 0 & -1 \\ +3 & -1 & +3 & -1 & 0 \end{bmatrix}$$

Εν συνεχεία ελέγχουμε αν τα τρία διανύσματα αποθηκεύτηκαν σωστά:

Εισάγουμε το διάνυσμα \mathbf{X}_1 :

1η Επανάληψη:

$$\mathbf{Y}_1(0) = \text{sign}(\mathbf{W} \times \mathbf{X}_1(0) - \boldsymbol{\theta}) = \text{sign} \left(\begin{bmatrix} 0 & -1 & +3 & -1 & +3 \\ -1 & 0 & -1 & +3 & -1 \\ +3 & -1 & 0 & -1 & +3 \\ -1 & +3 & -1 & 0 & -1 \\ +3 & -1 & +3 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} +1 \\ +1 \\ +1 \\ +1 \\ +1 \end{bmatrix} \right) = \text{sign} \left(\begin{bmatrix} +4 \\ 0 \\ +4 \\ 0 \\ +4 \end{bmatrix} \right) = \begin{bmatrix} +1 \\ 0 \\ +1 \\ 0 \\ +1 \end{bmatrix}$$

2η επανάληψη:

$$\mathbf{Y}_1(1) = \text{sign}(\mathbf{W} \times \mathbf{X}_1(1) - \boldsymbol{\theta}) = \text{sign} \left(\begin{bmatrix} 0 & -1 & +3 & -1 & +3 \\ -1 & 0 & -1 & +3 & -1 \\ +3 & -1 & 0 & -1 & +3 \\ -1 & +3 & -1 & 0 & -1 \\ +3 & -1 & +3 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} +1 \\ 0 \\ +1 \\ 0 \\ +1 \end{bmatrix} \right) = \text{sign} \left(\begin{bmatrix} +6 \\ -3 \\ +6 \\ -3 \\ +6 \end{bmatrix} \right) = \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \\ +1 \end{bmatrix}$$

Εισάγουμε το διάνυσμα \mathbf{X}_2 :

1η επανάληψη:

$$\mathbf{Y}_2(0) = \text{sign}(\mathbf{W} \times \mathbf{X}_2(0) - \boldsymbol{\theta}) = \text{sign}\left(\begin{bmatrix} 0 & -1 & +3 & -1 & +3 \\ -1 & 0 & -1 & +3 & -1 \\ +3 & -1 & 0 & -1 & +3 \\ -1 & +3 & -1 & 0 & -1 \\ +3 & -1 & +3 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \\ +1 \end{bmatrix} \right) = \text{sign}\left(\begin{bmatrix} +8 \\ -6 \\ +8 \\ -6 \\ +8 \end{bmatrix} \right) = \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \\ +1 \end{bmatrix}$$

Εισάγουμε το διάνυσμα \mathbf{X}_3 :

1η επανάληψη:

$$\mathbf{Y}_3(0) = \text{sign}(\mathbf{W} \times \mathbf{X}_3(0) - \boldsymbol{\theta}) = \text{sign}\left(\begin{bmatrix} 0 & -1 & +3 & -1 & +3 \\ -1 & 0 & -1 & +3 & -1 \\ +3 & -1 & 0 & -1 & +3 \\ -1 & +3 & -1 & 0 & -1 \\ +3 & -1 & +3 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} \right) = \text{sign}\left(\begin{bmatrix} -8 \\ +6 \\ -8 \\ +6 \\ -8 \end{bmatrix} \right) = \begin{bmatrix} -1 \\ +1 \\ -1 \\ +1 \\ -1 \end{bmatrix}$$

Παρατηρούμε ότι το X_1 δεν αποθηκεύτηκε σωστά, αφού βάζοντάς το ως είσοδο παράγεται το X_2 . Αντίθετα, τα X_2 και X_3 αποθηκεύτηκαν σωστά.

b. Εισάγουμε στο δίκτυο το $\mathbf{X} = \mathbf{X}(0) = \begin{bmatrix} +1 \\ +1 \\ -1 \\ +1 \\ +1 \end{bmatrix}$ και υπολογίζουμε τη βασική μνήμη στην οποία καταλήγει το

δίκτυο.

$$\mathbf{Y}(0) = \text{sign}(\mathbf{W} \times \mathbf{X}(0) - \boldsymbol{\theta}) = \text{sign}\left(\begin{bmatrix} 0 & -1 & +3 & -1 & +3 \\ -1 & 0 & -1 & +3 & -1 \\ +3 & -1 & 0 & -1 & +3 \\ -1 & +3 & -1 & 0 & -1 \\ +3 & -1 & +3 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} +1 \\ +1 \\ -1 \\ +1 \\ +1 \end{bmatrix} \right) = \text{sign}\left(\begin{bmatrix} -2 \\ +2 \\ +4 \\ +2 \\ -2 \end{bmatrix} \right) = \begin{bmatrix} -1 \\ +1 \\ +1 \\ +1 \\ -1 \end{bmatrix}$$

$$\mathbf{Y}(1) = \text{sign}(\mathbf{W} \times \mathbf{X}(1) - \boldsymbol{\theta}) = \text{sign}\left(\begin{bmatrix} 0 & -1 & +3 & -1 & +3 \\ -1 & 0 & -1 & +3 & -1 \\ +3 & -1 & 0 & -1 & +3 \\ -1 & +3 & -1 & 0 & -1 \\ +3 & -1 & +3 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ +1 \\ +1 \\ +1 \\ -1 \end{bmatrix} \right) = \text{sign}\left(\begin{bmatrix} -2 \\ +4 \\ -8 \\ +4 \\ -2 \end{bmatrix} \right) = \begin{bmatrix} -1 \\ +1 \\ -1 \\ +1 \\ -1 \end{bmatrix}$$

Παρατηρούμε ότι το δίκτυο ισορροπεί στην κατάσταση \mathbf{X}_3 και όχι στη \mathbf{X}_1 όπως αναμενόταν. Όπως όμως είδαμε και στο προηγούμενο ερώτημα, το διάνυσμα \mathbf{X}_1 δεν αποθηκεύτηκε σωστά.

c. Ο μέγιστος αριθμός διανυσμάτων (βασικές μνήμες) που μπορούμε να αποθηκεύσουμε σε ένα δίκτυο Hopfield 5 νευρώνων χωρίς να ενδιαφερόμαστε για την ικανότητα ανάκλησης (αν έχουμε λάθη ή όχι) είναι:

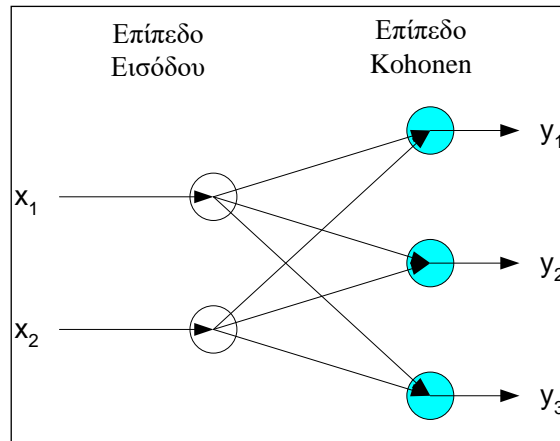
$$M_{\max} = 0.15 \cdot n^{n=5} = 0.15 \cdot 5 = 0.75$$

d. Ο μέγιστος αριθμός διανυσμάτων (βασικές μνήμες) που μπορούμε να αποθηκεύσουμε σε ένα δίκτυο Hopfield 5 νευρώνων εξασφαλίζοντας τέλεια ανάκληση (χωρίς κανένα λάθος) είναι:

$$M_{\max} = \frac{n}{4 \cdot \ln n} = \frac{5}{4 \cdot \ln 5} = 0.7767$$

Άσκηση 5.4:

Υποθέστε ότι έχουμε ένα δίκτυο Kohonen με δύο εισόδους και τρεις νευρώνες (στο επίπεδο Kohonen), όπως φαίνεται στο παρακάτω σχήμα.



Σχήμα 5.7 Το δίκτυο Kohonen.

Έστω ότι τα αντίστοιχα βάρη (υποθέστε ότι δεν υπάρχουν συνδέσεις μεταξύ των κρυφών νευρώνων) που συνδέουν τους νευρώνες εισόδου με τους νευρώνες του επιπέδου Kohonen είναι:

$$\mathbf{W}_1 = \begin{bmatrix} 0.27 \\ 0.81 \end{bmatrix} \text{ και } \mathbf{W}_2 = \begin{bmatrix} 0.42 \\ 0.70 \end{bmatrix} \text{ και } \mathbf{W}_3 = \begin{bmatrix} 0.43 \\ 0.21 \end{bmatrix}$$

a. Βρείτε τον νικητή νευρώνα στην περίπτωση που στην είσοδο του δικτύου παρουσιαστεί το διάνυσμα:

$$\mathbf{X} = \begin{bmatrix} 0.52 \\ 0.12 \end{bmatrix}$$

b. χρησιμοποιώντας το κριτήριο της ελάχιστης Ευκλείδειας απόστασης.

c. Υπολογίστε τις μεταβολές στα συναπτικά βάρη, υποθέτοντας ότι ο ρυθμός μάθησης α έχει τιμή 0.1.

d. Υπολογίστε τη νέα τιμή του διανύσματος των βαρών του νικητή νευρώνα.

Απάντηση

a. Υπολογίζουμε τις Ευκλείδειες αποστάσεις κάθε νευρώνα από το διάνυσμα εισόδου. Έτσι έχουμε:

$$d_1 = \sqrt{(x_1 - w_{11})^2 + (x_2 - w_{12})^2} = \sqrt{(0.52 - 0.27)^2 + (0.12 - 0.81)^2} = 0.73$$

$$d_2 = \sqrt{(x_1 - w_{21})^2 + (x_2 - w_{22})^2} = \sqrt{(0.52 - 0.42)^2 + (0.12 - 0.70)^2} = 0.59$$

$$d_3 = \sqrt{(x_1 - w_{31})^2 + (x_2 - w_{32})^2} = \sqrt{(0.52 - 0.43)^2 + (0.12 - 0.21)^2} = 0.13$$

Παρατηρούμε ότι ο νευρώνας 3 είναι αυτός με τη μικρότερη Ευκλείδεια απόσταση από το διάνυσμα εισόδου \mathbf{X} .

b. Σύμφωνα με τον αλγόριθμο εκπαίδευσης θα μεταβληθούν μόνο τα συναπτικά βάρη του νικητή νευρώνα. Έτσι θα έχουμε:

$$\Delta w_{31} = \alpha \cdot (x_1 - w_{31}) = 0.1 \cdot (0.52 - 0.43) = 0.009$$

$$\Delta w_{32} = \alpha \cdot (x_2 - w_{32}) = 0.1 \cdot (0.12 - 0.21) = -0.009$$

c. Μόνο το διάνυσμα βαρών του νικητή νευρώνα θα ενημερωθεί. Οπότε θα έχουμε:

$$\mathbf{W}_3(\mathbf{p}+1) = \mathbf{W}_3(\mathbf{p}) + \Delta \mathbf{W}_3(\mathbf{p}) = \begin{bmatrix} 0.43 \\ 0.21 \end{bmatrix} + \begin{bmatrix} 0.009 \\ -0.009 \end{bmatrix} = \begin{bmatrix} 0.439 \\ 0.201 \end{bmatrix} \cong \begin{bmatrix} 0.44 \\ 0.20 \end{bmatrix}$$

ενώ τα διανύσματα των βαρών των υπολοίπων νευρώνων θα μείνουν ως έχουν:

$$\mathbf{W}_1(\mathbf{p}+1) = \mathbf{W}_1(\mathbf{p}) = \begin{bmatrix} 0.27 \\ 0.81 \end{bmatrix} \text{ και } \mathbf{W}_2(\mathbf{p}+1) = \mathbf{W}_2(\mathbf{p}) = \begin{bmatrix} 0.42 \\ 0.70 \end{bmatrix}$$

Άσκηση 5.5:

Θεωρήστε ένα δίκτυο Kohonen με τρεις εισόδους και δύο νευρώνες Kohonen. Θεωρήστε ότι υπάρχουν συνδέσεις μόνο μεταξύ των εισόδων και των νευρώνων Kohonen. Η δομή του δικτύου αυτού απεικονίζεται στο παρακάτω σχήμα (5.8).

Έστω ότι θέλουμε να εκπαιδύσουμε το δίκτυο χρησιμοποιώντας τα ακόλουθα τέσσερα διανύσματα εκπαίδευσης:

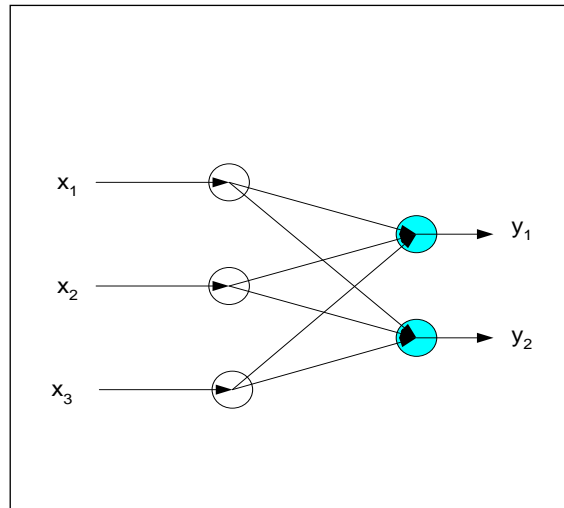
$$\mathbf{X}_1 = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.4 \end{bmatrix}, \mathbf{X}_2 = \begin{bmatrix} 0.6 \\ 0.9 \\ 0.9 \end{bmatrix}, \mathbf{X}_3 = \begin{bmatrix} 0.3 \\ 0.4 \\ 0.1 \end{bmatrix}, \mathbf{X}_4 = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.3 \end{bmatrix}$$

Έστω επίσης ότι τα αρχικά βάρη που ενώνουν τους νευρώνες εισόδου με τους νευρώνες Kohonen είναι:

$$\mathbf{W}_1 = \begin{bmatrix} 0.5 \\ 0.6 \\ 0.8 \end{bmatrix} \text{ και } \mathbf{W}_2 = \begin{bmatrix} 0.4 \\ 0.2 \\ 0.5 \end{bmatrix}$$

Θεωρήστε επίσης ότι η ακτίνα της γειτονιάς του νικητή νευρώνα είναι 0 (η γειτονιά περιλαμβάνει μόνο τον νικητή νευρώνα) και ο ρυθμός μάθησης είναι 0.5.

Υπολογίστε τις μεταβολές και τις τιμές των συναπτικών βαρών κατά τη διάρκεια του πρώτου κύκλου εκπαίδευσης χρησιμοποιώντας τα διανύσματα εισόδου με τη σειρά με την οποία δίνονται.



Σχήμα 5.8 Το δίκτυο Kohonen.

Απάντηση

Για το διάνυσμα X_1 υπολογίζουμε τις Ευκλείδειες αποστάσεις.

$$\text{Για τον 1ο νευρώνα Kohonen: } d_1 = \sqrt{(0.5 - 0.8)^2 + (0.6 - 0.7)^2 + (0.8 - 0.4)^2} = 0.51$$

$$\text{Για τον 2ο νευρώνα Kohonen: } d_2 = \sqrt{(0.4 - 0.8)^2 + (0.2 - 0.7)^2 + (0.5 - 0.4)^2} = 0.65$$

Παρατηρούμε ότι ο νευρώνας 1 είναι πλησιέστερα (με βάση την Ευκλείδεια απόσταση) στο διάνυσμα X_1 , οπότε αυτός είναι και ο νικητής νευρώνας σε αυτήν την πρώτη φάση. Άρα:

$$\begin{aligned} \mathbf{W}_1(\mathbf{1}) &= \mathbf{W}_1(0) + 0.5 \cdot (\mathbf{X}_1 - \mathbf{W}_1(0)) = \begin{bmatrix} 0.5 \\ 0.6 \\ 0.8 \end{bmatrix} + 0.5 \cdot \left(\begin{bmatrix} 0.8 \\ 0.7 \\ 0.4 \end{bmatrix} - \begin{bmatrix} 0.5 \\ 0.6 \\ 0.8 \end{bmatrix} \right) = \\ &= \begin{bmatrix} 0.5 \\ 0.6 \\ 0.8 \end{bmatrix} + \begin{bmatrix} 0.15 \\ 0.05 \\ -0.2 \end{bmatrix} = \begin{bmatrix} 0.65 \\ 0.65 \\ 0.60 \end{bmatrix} \end{aligned}$$

Επομένως μετά την παρουσίαση του διανύσματος X_1 τα συναπτικά βάρη του δικτύου έχουν ως εξής:

$$\mathbf{W}_1(1) = \begin{bmatrix} 0.65 \\ 0.65 \\ 0.60 \end{bmatrix} \text{ και } \mathbf{W}_2(1) = \begin{bmatrix} 0.4 \\ 0.2 \\ 0.5 \end{bmatrix}$$

Παρουσιάζουμε το διάνυσμα X_2 και υπολογίζουμε τις Ευκλείδειες αποστάσεις.

$$\text{Για τον 1ο νευρώνα Kohonen: } d_1 = \sqrt{(0.65 - 0.6)^2 + (0.65 - 0.9)^2 + (0.60 - 0.9)^2} = 0.39$$

$$\text{Για τον 2ο νευρώνα Kohonen: } d_2 = \sqrt{(0.4 - 0.6)^2 + (0.2 - 0.9)^2 + (0.5 - 0.9)^2} = 0.83$$

Παρατηρούμε ότι ο νευρώνας 1 είναι πλησιέστερα (με βάση την Ευκλείδεια απόσταση) στο διάνυσμα X_2 , οπότε αυτός είναι και ο νικητής νευρώνας σε αυτήν τη δεύτερη φάση. Άρα:

$$\begin{aligned} \mathbf{W}_1(2) &= \mathbf{W}_1(1) + 0.5 \cdot (\mathbf{X}_2 - \mathbf{W}_1(1)) = \begin{bmatrix} 0.65 \\ 0.65 \\ 0.60 \end{bmatrix} + 0.5 \cdot \left(\begin{bmatrix} 0.6 \\ 0.9 \\ 0.9 \end{bmatrix} - \begin{bmatrix} 0.65 \\ 0.65 \\ 0.60 \end{bmatrix} \right) = \\ &= \begin{bmatrix} 0.65 \\ 0.65 \\ 0.60 \end{bmatrix} + \begin{bmatrix} -0.025 \\ 0.125 \\ 0.15 \end{bmatrix} = \begin{bmatrix} 0.625 \\ 0.775 \\ 0.750 \end{bmatrix} \end{aligned}$$

Επομένως μετά την παρουσίαση και του διανύσματος X_2 τα συναπτικά βάρη του δικτύου έχουν ως εξής:

$$\mathbf{W}_1(2) = \begin{bmatrix} 0.625 \\ 0.775 \\ 0.750 \end{bmatrix} \text{ και } \mathbf{W}_2(2) = \begin{bmatrix} 0.4 \\ 0.2 \\ 0.5 \end{bmatrix}$$

Παρουσιάζουμε το διάνυσμα X_3 και υπολογίζουμε τις Ευκλείδειες αποστάσεις.

$$\text{Για τον 1ο νευρώνα Kohonen: } d_1 = \sqrt{(0.625 - 0.3)^2 + (0.775 - 0.4)^2 + (0.75 - 0.1)^2} = 0.82$$

$$\text{Για τον 2ο νευρώνα Kohonen: } d_2 = \sqrt{(0.4 - 0.3)^2 + (0.2 - 0.4)^2 + (0.5 - 0.1)^2} = 0.46$$

Παρατηρούμε ότι αυτήν τη φορά ο νευρώνας 2 είναι αυτός που είναι πλησιέστερα (με βάση την Ευκλείδεια απόσταση) στο διάνυσμα X_3 , οπότε αυτός είναι και ο νικητής νευρώνας σε αυτήν την τρίτη φάση. Άρα:

$$\begin{aligned} \mathbf{W}_2(3) &= \mathbf{W}_2(2) + 0.5 \cdot (\mathbf{X}_3 - \mathbf{W}_2(2)) = \begin{bmatrix} 0.4 \\ 0.2 \\ 0.5 \end{bmatrix} + 0.5 \cdot \left(\begin{bmatrix} 0.3 \\ 0.4 \\ 0.1 \end{bmatrix} - \begin{bmatrix} 0.4 \\ 0.2 \\ 0.5 \end{bmatrix} \right) = \\ &= \begin{bmatrix} 0.4 \\ 0.2 \\ 0.5 \end{bmatrix} + \begin{bmatrix} -0.05 \\ 0.1 \\ -0.2 \end{bmatrix} = \begin{bmatrix} 0.35 \\ 0.30 \\ 0.30 \end{bmatrix} \end{aligned}$$

Επομένως μετά την παρουσίαση και του διανύσματος X_3 τα συναπτικά βάρη του δικτύου έχουν ως εξής:

$$\mathbf{W}_1(3) = \begin{bmatrix} 0.625 \\ 0.775 \\ 0.750 \end{bmatrix} \text{ και } \mathbf{W}_2(3) = \begin{bmatrix} 0.35 \\ 0.30 \\ 0.30 \end{bmatrix}$$

Παρουσιάζουμε το διάνυσμα X_4 και υπολογίζουμε τις Ευκλείδειες αποστάσεις.

$$\text{Για τον 1ο νευρώνα Kohonen: } d_1 = \sqrt{(0.625 - 0.1)^2 + (0.775 - 0.1)^2 + (0.75 - 0.3)^2} = 0.96$$

$$\text{Για τον 2ο νευρώνα Kohonen: } d_2 = \sqrt{(0.35 - 0.1)^2 + (0.3 - 0.1)^2 + (0.3 - 0.3)^2} = 0.32$$

Παρατηρούμε ότι ο νευρώνας 2 είναι αυτός που είναι πλησιέστερα (με βάση την Ευκλείδεια απόσταση) στο διάνυσμα X_4 , οπότε αυτός είναι και ο νικητής νευρώνας σε αυτήν την τέταρτη φάση. Άρα:

$$\begin{aligned} \mathbf{W}_2(4) &= \mathbf{W}_2(3) + 0.5 \cdot (\mathbf{X}_4 - \mathbf{W}_2(3)) = \begin{bmatrix} 0.35 \\ 0.30 \\ 0.30 \end{bmatrix} + 0.5 \cdot \left(\begin{bmatrix} 0.1 \\ 0.1 \\ 0.3 \end{bmatrix} - \begin{bmatrix} 0.35 \\ 0.30 \\ 0.30 \end{bmatrix} \right) = \\ &= \begin{bmatrix} 0.35 \\ 0.30 \\ 0.30 \end{bmatrix} + \begin{bmatrix} -0.125 \\ -0.1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.225 \\ 0.200 \\ 0.300 \end{bmatrix} \end{aligned}$$

Επομένως μετά την παρουσίαση και του διανύσματος X_4 (οπότε και έχουμε συμπληρώσει έναν κύκλο εκπαίδευσης) τα συναπτικά βάρη του δικτύου έχουν ως εξής:

$$\mathbf{W}_1(4) = \begin{bmatrix} 0.625 \\ 0.775 \\ 0.750 \end{bmatrix} \text{ και } \mathbf{W}_2(4) = \begin{bmatrix} 0.225 \\ 0.200 \\ 0.300 \end{bmatrix}$$

Άσκηση 5.6:

Έστω ότι διαθέτουμε δεδομένα για τη συμπεριφορά οχτώ διαφορετικών χρηστών σε ένα σύστημα ηλεκτρονικής μάθησης. Για κάθε χρήστη έχουμε κρατήσει τον μέσο όρο του χρόνου που περνάει συνδεδεμένος στο σύστημα καθημερινά και τον μέσο όρο των μηνυμάτων που ανταλλάσσει με τους άλλους χρήστες. Οι τιμές των δύο αυτών χαρακτηριστικών έχουν κανονικοποιηθεί στο διάστημα $[0-10]$. Θέλουμε με βάση αυτά τα δύο χαρακτηριστικά να ομαδοποιήσουμε τους χρήστες σε τρεις κατηγορίες. Για τα οχτώ προφίλ χρηστών μετά την κανονικοποίηση έχουμε τις εξής τιμές για τα δύο χαρακτηριστικά τους:

$$A1=(2,10), A2=(2,5), A3(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9)$$

Θέλουμε να επιλύσουμε αυτό το πρόβλημα με ένα δίκτυο Kohonen. Τα αρχικά βάρη των νευρώνων Kohonen που θα χρησιμοποιήσετε είναι $W1=A1$, $W2=A4$, $W3=A7$.

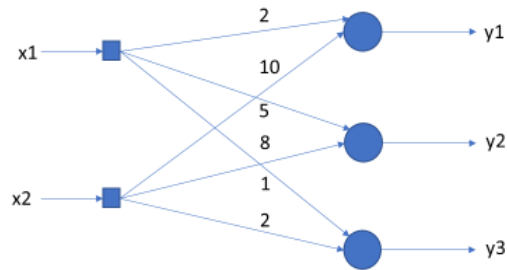
Θεωρήστε ότι:

- Υπάρχουν συνδέσεις μόνο μεταξύ των εισόδων και των νευρώνων Kohonen.
- Η ακτίνα της γειτονιάς του νικητή νευρώνα είναι πάντα 0 (η γειτονιά περιλαμβάνει δηλαδή μόνο τον νικητή νευρώνα).
- Ο ρυθμός εκπαίδευσης α είναι ίσος με $\alpha=0.5$.
- Κάθε διάνυσμα εισόδου αντιστοιχεί σε έναν κύκλο εκπαίδευσης.

α) Πόσοι εισόδοι και πόσοι ανταγωνιστικοί νευρώνες υπάρχουν στο δίκτυο; Σχεδιάστε το δίκτυο με τις αρχικές του τιμές.

Απάντηση:

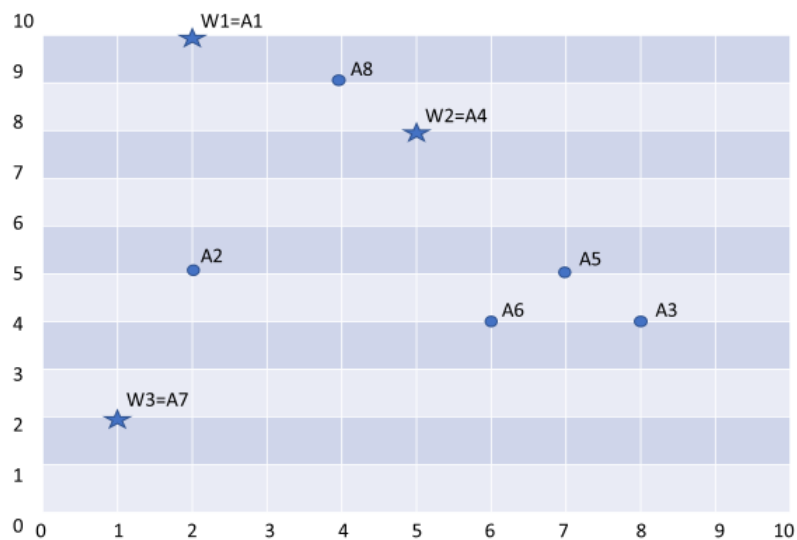
Υπάρχουν 2 εισόδοι και 3 ανταγωνιστικοί νευρώνες (νευρώνες Kohonen). Το δίκτυο φαίνεται στο παρακάτω σχήμα:



Σχήμα 5.9 Το δίκτυο Kohonen.

β) Σχεδιάστε ένα διάγραμμα με άξονες τα δύο χαρακτηριστικά (μέσος όρος σύνδεσης, μέσος όρος μηνυμάτων) στο οποίο να φαίνονται τα σημεία που αντιστοιχούν στον κάθε χρήστη. Με star να φαίνονται τα σημεία που αντιστοιχούν στα αρχικά βάρη.

Απάντηση:



Σχήμα 5.10 Διάγραμμα χαρακτηριστικών.

γ) Να εκπαιδεύσετε το δίκτυο για τα διανύσματα εισόδου A2 και A3 παρατηρώντας κάθε φορά την αλλαγή των βαρών του δικτύου; Τι συμπεραίνετε;

Απάντηση:

$$A2=(2,5)$$

$$d1=\text{sqrt}(25), d2=\text{sqrt}(18), d3=\text{sqrt}(10)$$

Ο νικητής νευρώνας είναι ο τρίτος και το βάρος του θα ανανεωθεί ως εξής:

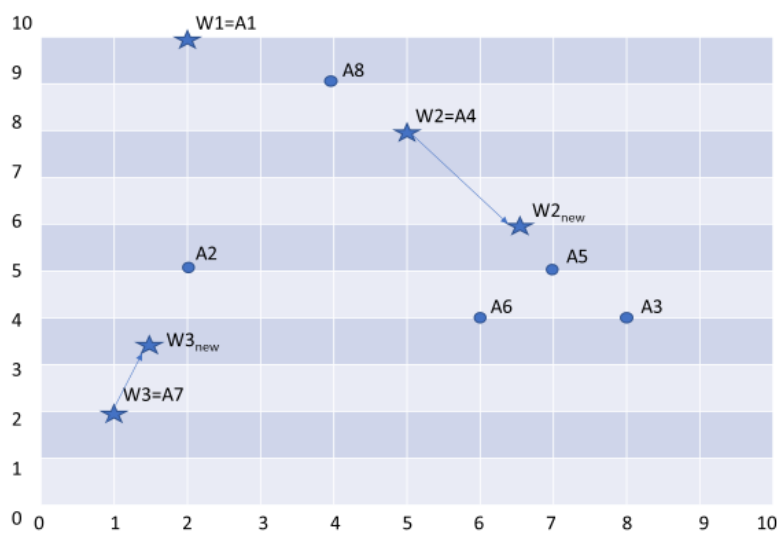
$$W3=W3+0.5*(A2-W3)=[1,2]+0.5*[1,3]=[1.5,3.5]$$

$$A3=(8,4)$$

$$d1=\text{sqrt}(36), d2=\text{sqrt}(25), d3=\text{sqrt}((8-1.5)^2+(4-3.5)^2)=\text{sqrt}(42.25+0.25)=\text{sqrt}(42.5)$$

Ο νικητής νευρώνας είναι ο δεύτερος και το βάρος του θα ανανεωθεί ως εξής:

$$W2=W2+0.5*(A3-W2)=[5,8]+0.5*[3,-4]=[6.5,6]$$



Σχήμα 5.11 Διάγραμμα χαρακτηριστικών κατά την εκπαίδευση.

Όπως παρατηρούμε από το παραπάνω σχήμα, μετά από κάθε επανάληψη εκπαίδευσης ο νικητής νευρώνας μετακινείται προς την κατεύθυνση του διανύσματος εκπαίδευσης.

δ) Με βάση το εκπαιδευμένο δίκτυο σε ποια από τις τρεις ομάδες (ομάδα 1, 2 ή 3 σε αντιστοιχία με τους νευρώνες Kohonen 1, 2 και 3) θα αντιστοιχούσατε τον χρήστη A5;

Απάντηση:

Τα βάρη του εκπαιδευμένου δικτύου είναι $W1=(2,10)$, $W2=[6.5,6]$ και $W3=[1.5,3.5]$

$$A5=(7,5) \text{ άρα}$$

$$d1=\text{sqrt}((2-7)^2+(5-10)^2)=\text{sqrt}(50)$$

$$d2=\text{sqrt}((6.5-7)^2+(6-5)^2)=\text{sqrt}(1.25)$$

$$d3=\text{sqrt}((1.5-7)^2+(5-3.5)^2)=\text{sqrt}(30.25+2.25)=\text{sqrt}(32.5)$$

Επομένως νικητής νευρώνας είναι ο 2 και ο χρήστης A5 ομαδοποιείται στην ομάδα 2.

Κεφάλαιο 6

Βελτιστοποιήσεις για την Εκπαίδευση Βαθέων Μοντέλων

Σύνοψη

Όταν κάποιος μαθαίνει για πρώτη φορά να παίζει γκολφ, συνήθως ξοδεύει το μεγαλύτερο μέρος του χρόνου του αναπτύσσοντας μια βασική ταλάντευση. Μόνο σταδιακά αναπτύσσει και άλλες βολές, μαθαίνοντας το κοντινό χτύπημα, το τράβηγμα και το ξεθώριασμα, χτίζοντας και τροποποιώντας τη βασική του κίνηση. Με παρόμοιο τρόπο μέχρι τώρα έχουμε επικεντρωθεί στην κατανόηση του αλγόριθμου πίσω διάδοσης. Είναι η «βασική μας κίνηση», το θεμέλιο για μάθηση στις περισσότερες εργασίες στα νευρωνικά δίκτυα. Σε αυτό το κεφάλαιο εξηγείται μια σειρά τεχνικών που μπορούν να χρησιμοποιηθούν για τη βελτίωση της εφαρμογής της κλασικής μεθόδου πίσω διάδοσης προκειμένου να εξελίξουν τον τρόπο με τον οποίο μαθαίνουν τα δίκτυά μας.

Οι τεχνικές που θα αναπτύξουμε σε αυτό το κεφάλαιο περιλαμβάνουν: μια καλύτερη επιλογή της λειτουργίας κόστους, γνωστή ως λειτουργία κόστους διεντροπίας (*cross-entropy*), τέσσερις λεγόμενες μεθόδους ομαλοποίησης (*regularization*) ($L1$ και $L2$ ομαλοποίηση, *dropout* και τεχνητή επέκταση του συνόλου εκπαίδευσης), οι οποίες κάνουν τα δίκτυά μας καλύτερα στη γενίκευση πέρα από τα δεδομένα κατάρτισης, μια καλύτερη μέθοδο για την αρχικοποίηση των βαρών στο δίκτυο και ένα σύνολο ευρετικών για την επιλογή καλών υπερ-παραμέτρων για το δίκτυο. Θα εξεταστούν και κάποιες ακόμη τεχνικές, αλλά σε λιγότερο βάθος.

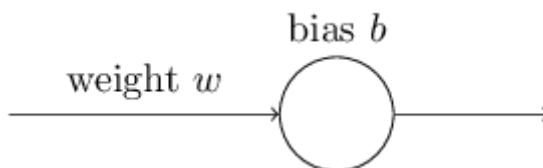
Φυσικά καλύπτουμε μόνο μερικές από τις πολλές τεχνικές που έχουν αναπτυχθεί για χρήση σε νευρωνικά δίκτυα. Η φιλοσοφία είναι ότι το καλύτερο σημείο εισόδου στην πληθώρα των διαθέσιμων τεχνικών είναι η εις βάθος μελέτη μερικών από τις πιο σημαντικές. Η γνώση αυτών των σημαντικών τεχνικών δεν είναι μόνο χρήσιμη από μόνη της, αλλά θα εμβαθύνει επιπλέον την κατανόηση για το ποια προβλήματα μπορεί να προκύψουν όταν χρησιμοποιούνται νευρωνικά δίκτυα. Αυτό επιτρέπει επομένως την κατανόηση και πρόσθετων τεχνικών, καθώς προκύπτει η αντίστοιχη ανάγκη.

Προαπαιτούμενη γνώση

Είναι καλό ο αναγνώστης να έχει μελετήσει προηγουμένως τα κεφάλαια 1, 2, 3, 4. Το μεγαλύτερο μέρος του κεφαλαίου αυτού βασίζεται στα Nielsen, 2019 και Goodfellow et al, 2016.

6.1 Η συνάρτηση κόστους διεντροπίας

Οι περισσότεροι από εμάς βρίσκουμε δυσάρεστο το να κάνουμε λάθος. Ωστόσο, ενώ είναι δυσάρεστο, μαθαίνουμε γρηγορότερα όταν κάνουμε αποφασιστικά λάθος. Αντίθετα, μαθαίνουμε πιο αργά όταν τα λάθη μας είναι λιγότερο καλά καθορισμένα. Στην ιδανική περίπτωση ελπίζουμε και αναμένουμε ότι τα νευρωνικά μας δίκτυα θα μάθουν γρήγορα από τα λάθη τους. Αυτό συμβαίνει στην πράξη; Για να απαντήσουμε σε αυτήν την ερώτηση, ας δούμε ένα απλό παράδειγμα. Το παράδειγμα περιλαμβάνει έναν νευρώνα με μία μόνο είσοδο:

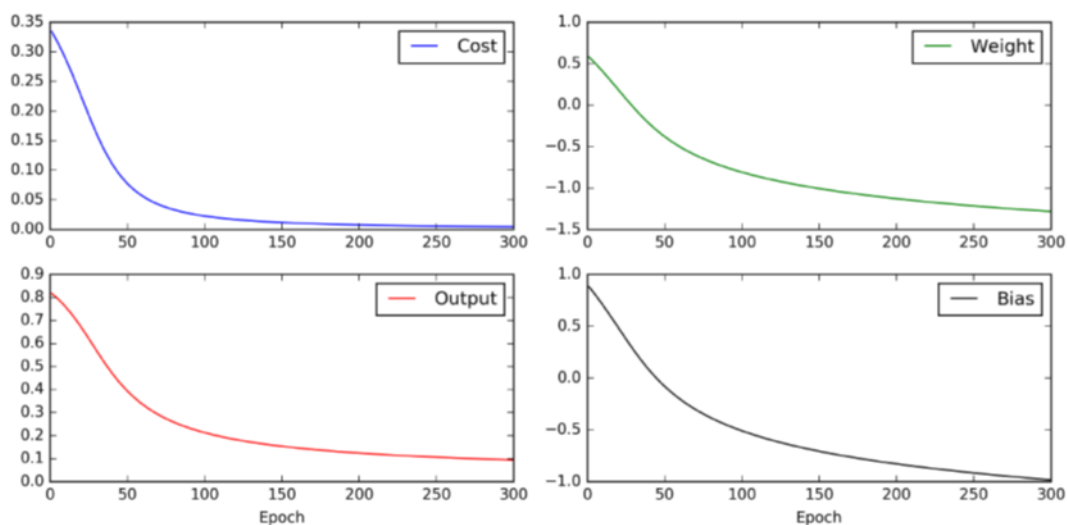


Σχήμα 6.1 Ένας απλός νευρώνας.

Θα εκπαιδύσουμε αυτόν τον νευρώνα να κάνει κάτι πολύ εύκολο: να παίρνει είσοδο 1 και να δίνει έξοδο 0. Φυσικά πρόκειται για κάτι τόσο τετριμμένο που θα μπορούσαμε εύκολα να βρούμε ένα κατάλληλο βάρος και την πόλωση με το χέρι, χωρίς να χρησιμοποιήσουμε έναν αλγόριθμο μάθησης. Ωστόσο, αποδεικνύεται ότι είναι διαφωτιστικό να χρησιμοποιήσει κανείς τη μέθοδο κατάβασης κλίσης για να προσπαθήσει να μάθει ένα βάρος και την πόλωση. Ας ρίξουμε μια ματιά στο πώς μαθαίνει ο νευρώνας.

Αναλυτικά: ας επιλέξουμε αρχικό βάρος 0,6 και αρχική πόλωση 0,9. Αυτές είναι γενικές επιλογές που χρησιμοποιούνται για την έναρξη της μάθησης και έχουν επιλεγεί τυχαία. Η αρχική έξοδος από τον νευρώνα είναι 0,82, οπότε θα χρειαστεί αρκετή μάθηση πριν ο νευρώνας μας πλησιάσει την επιθυμητή έξοδο, 0,0. Ο

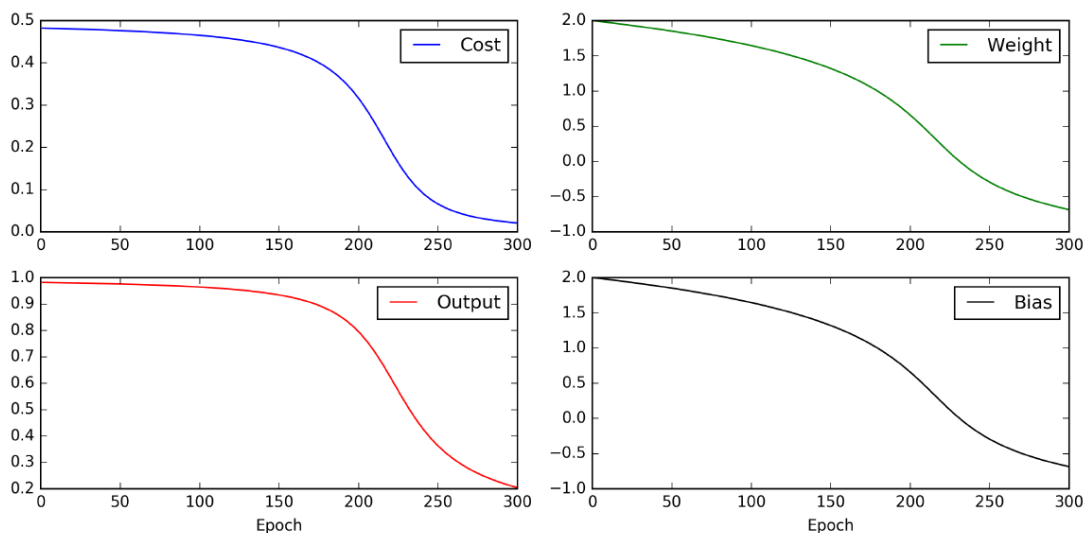
ρυθμός μάθησης είναι $\eta=0,15$, το οποίο αποδεικνύεται αρκετά αργό ώστε να μπορούμε να ακολουθήσουμε αυτό που συμβαίνει, αλλά και αρκετά γρήγορο ώστε να μπορέσουμε να αποκτήσουμε ουσιαστική μάθηση σε λίγα δευτερόλεπτα. Το κόστος είναι η τετραγωνική συνάρτηση κόστους, C .



Σχήμα 6.2 Καμπύλες μάθησης του νευρώνα.

Όπως φαίνεται στην εικόνα, ο νευρώνας μαθαίνει γρήγορα ένα βάρος και την πόλωση που μειώνει το κόστος και δίνει μια έξοδο από τον νευρώνα περίπου 0,09. Αυτό δεν είναι ακριβώς το επιθυμητό αποτέλεσμα, 0,0, αλλά είναι αρκετά καλό.

Ας υποθέσουμε ωστόσο ότι αντ' αυτού επιλέγουμε τόσο το αρχικό βάρος όσο και την αρχική πόλωση να είναι 2,0. Σε αυτήν την περίπτωση η αρχική έξοδος είναι 0,98, η οποία είναι εξαιρετικά λανθασμένη. Ας δούμε πώς ο νευρώνας μαθαίνει να δίνει έξοδο 0 σε αυτήν την περίπτωση.



Σχήμα 6.3 Καμπύλες μάθησης του νευρώνα με νέες αρχικές τιμές βάρους και πόλωσης.

Αν και αυτό το παράδειγμα χρησιμοποιεί τον ίδιο ρυθμό μάθησης ($\eta=0,15$), μπορούμε να δούμε ότι η μάθηση ξεκινά πολύ πιο αργά. Πράγματι για τις πρώτες 150 περίπου εποχές μάθησης, τα βάρη και οι πολώσεις δεν αλλάζουν καθόλου. Στη συνέχεια η μάθηση ξεκινά και, όπως και στο πρώτο μας παράδειγμα, η έξοδος του νευρώνα γρήγορα κινείται πιο κοντά στο 0,0.

Η συγκεκριμένη συμπεριφορά είναι παράξενη συγκρινόμενη με την ανθρώπινη μάθηση. Όπως αναφέρθηκε στην αρχή του κεφαλαίου, συχνά μαθαίνουμε ταχύτερα όταν σφάλουμε πολύ για κάτι. Αλλά εδώ παρατηρούμε ότι ο τεχνητός νευρώνας μας δυσκολεύεται πολύ να μάθει όταν σφάλει πολύ — πρόκειται για πολύ μεγαλύτερη δυσκολία από ό,τι όταν το λάθος είναι μικρό. Επιπλέον, αποδεικνύεται ότι αυτή η συμπεριφορά συμβαίνει όχι μόνο σε αυτό το απλοϊκό μοντέλο, αλλά και σε πιο γενικά δίκτυα. Γιατί η μάθηση είναι τόσο αργή; Μπορούμε να βρούμε έναν τρόπο να αποφύγουμε αυτήν την επιβράδυνση;

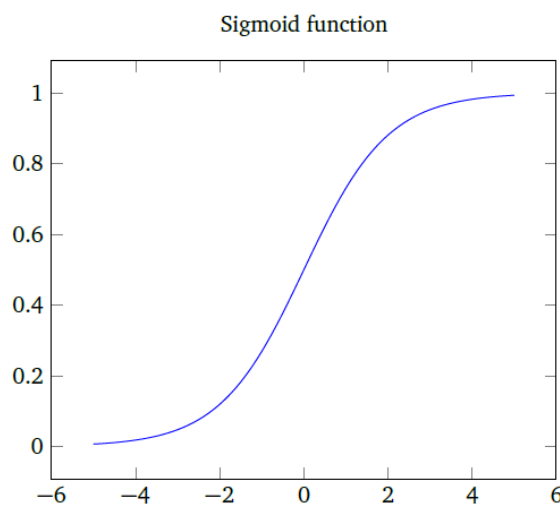
Για να κατανοήσουμε την προέλευση του προβλήματος, ας σκεφτούμε ότι ο νευρώνας μας μαθαίνει αλλάζοντας το βάρος και την πόλωση με ρυθμό που καθορίζεται από τις μερικές παραγώγους της συνάρτησης κόστους, $\partial C / \partial w$ και $\partial C / \partial b$. Έτσι το να λέμε ότι «η μάθηση είναι αργή» ταυτίζεται με το να πούμε ότι αυτές οι μερικές παράγωγοι είναι μικρές. Η πρόκληση είναι να καταλάβουμε γιατί είναι μικρές. Για να το καταλάβουμε λοιπόν, ας υπολογίσουμε τις μερικές παραγώγους. Χρησιμοποιούμε την τετραγωνική συνάρτηση κόστους, η οποία είναι:

$$C = \frac{(y - a)^2}{2} \quad (6.1)$$

όπου a είναι η έξοδος του νευρώνα όταν χρησιμοποιείται η είσοδος $x=1$ και το $y=0$ είναι η αντίστοιχη επιθυμητή έξοδος. Επειδή $a=\sigma(z)$, όπου σ η λογιστική συνάρτηση και $z=wx + b$, χρησιμοποιώντας τον κανόνα της αλυσίδας για να παραγωγίσουμε ως προς το βάρος και την πόλωση, λαμβάνουμε:

$$\begin{aligned} \frac{\partial C}{\partial w} &= (a - y)\sigma'(z)x = a\sigma'(z) \\ \frac{\partial C}{\partial b} &= (a - y)\sigma'(z) = a\sigma'(z) \end{aligned} \quad (6.2), (6.3)$$

όπου έχουμε αντικαταστήσει το $x=1$ και $y=0$. Για να κατανοήσουμε τη συμπεριφορά αυτών των εκφράσεων, ας εξετάσουμε πιο προσεκτικά τον όρο $\sigma'(z)$ στη δεξιά πλευρά. Θυμηθείτε το σχήμα της συνάρτησης σ :

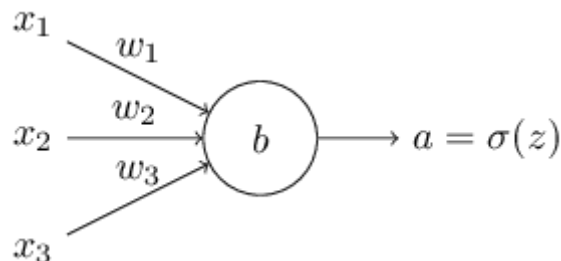


Σχήμα 6.4 Σιγμοειδής συνάρτηση.

Μπορούμε να δούμε από αυτό το γράφημα ότι όταν η έξοδος του νευρώνα είναι κοντά στο 1, η καμπύλη γίνεται πολυεπίπεδη και έτσι το $\sigma'(z)$ γίνεται πολύ μικρό. Οι εξισώσεις (6.2) και (6.3) συνεπώς μας λένε ότι οι $\partial C / \partial w$ και $\partial C / \partial b$ γίνονται πολύ μικρές. Αυτή είναι η προέλευση της επιβράδυνσης της μάθησης. Επιπλέον, όπως θα δούμε λίγο αργότερα, η επιβράδυνση της μάθησης συμβαίνει —για τον ίδιο ουσιαστικά λόγο— και σε γενικότερα νευρωνικά δίκτυα, όχι μόνο στο παραπάνω απλοϊκό παράδειγμα.

6.1.1 Εισαγωγή της συνάρτησης κόστους διεντροπίας

Πώς μπορούμε να αντιμετωπίσουμε την επιβράδυνση της μάθησης; Αποδεικνύεται ότι μπορούμε να λύσουμε το πρόβλημα αντικαθιστώντας το τετραγωνικό κόστος με μια διαφορετική συνάρτηση κόστους, γνωστή ως διεντροπία. Για να κατανοήσουμε τη διεντροπία, ας απομακρυνθούμε λίγο από το απλοϊκό μας μοντέλο. Θα υποθέσουμε αντ' αυτού ότι προσπαθούμε να εκπαιδεύσουμε έναν νευρώνα με πολλές μεταβλητές εισόδου, x_1, x_2, \dots , αντίστοιχα βάρη w_1, w_2, \dots , και μια πόλωση b :



Σχήμα 6.5 Νευρώνας πολλαπλών εισόδων.

Η έξοδος του νευρώνα είναι φυσικά $a = \sigma(z)$, όπου $z = \sum_j w_j x_j + b$ είναι το σταθμισμένο άθροισμα των εισόδων. Ορίζουμε τη συνάρτηση κόστους διεντροπίας γι' αυτόν τον νευρώνα ως:

$$C = -\frac{1}{n} \sum_x [y \ln a + (1 - y) \ln(1 - a)] \quad (6.4)$$

όπου n είναι ο συνολικός αριθμός των στοιχείων των δεδομένων εκπαίδευσης, το άθροισμα είναι πάνω από όλες τις εισόδους εκπαίδευσης, x είναι η είσοδος και y είναι η αντίστοιχη επιθυμητή έξοδος.

Δεν είναι προφανές ότι η έκφραση (6.4) διορθώνει το πρόβλημα επιβράδυνσης της μάθησης. Στην πραγματικότητα δεν είναι καν προφανές ότι είναι λογικό να την ονομάσουμε συνάρτηση κόστους! Πριν αντιμετωπίσουμε την επιβράδυνση της μάθησης, ας δούμε με ποια έννοια η διεντροπία μπορεί να ερμηνευθεί ως συνάρτηση κόστους.

Ειδικότερα δύο ιδιότητες καθιστούν εύλογη την ερμηνεία της διεντροπίας ως συνάρτησης κόστους. Πρώτον είναι μη αρνητική, δηλαδή $C > 0$. Για να το αντιληφθείτε, παρατηρήστε ότι: (α) όλοι οι επιμέρους όροι στο άθροισμα είναι αρνητικοί, καθώς και οι δύο λογάριθμοι αφορούν αριθμούς στην περιοχή 0 έως 1 και (β) υπάρχει ένα σύμβολο μείον μπροστά από το άθροισμα.

Δεύτερον, εάν η πραγματική έξοδος του νευρώνα είναι κοντά στην επιθυμητή έξοδο για όλες τις εισόδους, x , τότε η διεντροπία θα είναι κοντά στο μηδέν¹. Για να το κατανοήσετε, ας υποθέσουμε π.χ. ότι $y=0$ και $a \approx 0$ για κάποια είσοδο x . Αυτό συμβαίνει όταν ο νευρώνας κάνει καλή δουλειά γι' αυτήν την είσοδο. Βλέπουμε ότι ο πρώτος όρος στην έκφραση (6.4) για το κόστος απαλείφεται, αφού $y=0$, ενώ ο δεύτερος όρος είναι μόνο $-\ln(1-a) \approx 0$. Παρόμοια ανάλυση ισχύει και όταν $y=1$ και $a \approx 1$. Έτσι η συνεισφορά στο κόστος θα είναι χαμηλή δεδομένου ότι η πραγματική έξοδος είναι κοντά στην επιθυμητή.

Συνοψίζοντας η διεντροπία είναι θετική και τείνει προς το μηδέν, καθώς ο νευρώνας βελτιώνεται στον υπολογισμό της επιθυμητής εξόδου, y , για όλες τις εισόδους, x . Αυτές είναι δύο ιδιότητες που θα περιμέναμε διαισθητικά από μια συνάρτηση κόστους. Πράγματι και οι δύο ιδιότητες ικανοποιούνται επίσης από το τετραγωνικό κόστος. Αλλά η διεντροπία έχει το πλεονέκτημα ότι, σε αντίθεση με το τετραγωνικό κόστος, αποφεύγει το πρόβλημα της επιβράδυνσης της μάθησης. Για να το δούμε αυτό, ας υπολογίσουμε τη μερική παράγωγο του κόστους διεντροπίας σε σχέση με τα βάρη. Αντικαθιστούμε το $a = \sigma(z)$ στην (6.4) και εφαρμόζουμε τον κανόνα της αλυσίδας δύο φορές παίρνοντας:

¹ Υποθέτουμε ότι οι επιθυμητές έξοδοι είναι όλες είτε 0 είτε 1. Αυτό συνήθως συμβαίνει σε προβλήματα ταξινόμησης ή κατά τον υπολογισμό λογικών συναρτήσεων. Σε διαφορετικές περιπτώσεις δείτε τις ασκήσεις στο τέλος της ενότητας.

$$\begin{aligned} \frac{\partial C}{\partial w_j} &= -\frac{1}{n} \sum_x \left(\frac{y}{\sigma(z)} - \frac{(1-y)}{1-\sigma(z)} \right) \frac{\partial \sigma}{\partial w_j} \\ &= -\frac{1}{n} \sum_x \left(\frac{y}{\sigma(z)} - \frac{(1-y)}{1-\sigma(z)} \right) \sigma'(z) x_j \end{aligned} \quad (6.5)$$

Θέτοντας κοινό παρανομαστή και απλοποιώντας προκύπτει:

$$\frac{\partial C}{\partial w_j} = \frac{1}{n} \sum_x \frac{\sigma'(z) x_j}{\sigma(z)(1-\sigma(z))} (\sigma(z) - y) \quad (6.6)$$

Όμως $\sigma'(z) = \sigma(z)(1-\sigma(z))$. Επομένως αυτοί οι όροι απαλείφονται στην παραπάνω εξίσωση και γίνεται:

$$\frac{\partial C}{\partial w_j} = \frac{1}{n} \sum_x x_j (\sigma(z) - y) \quad (6.7)$$

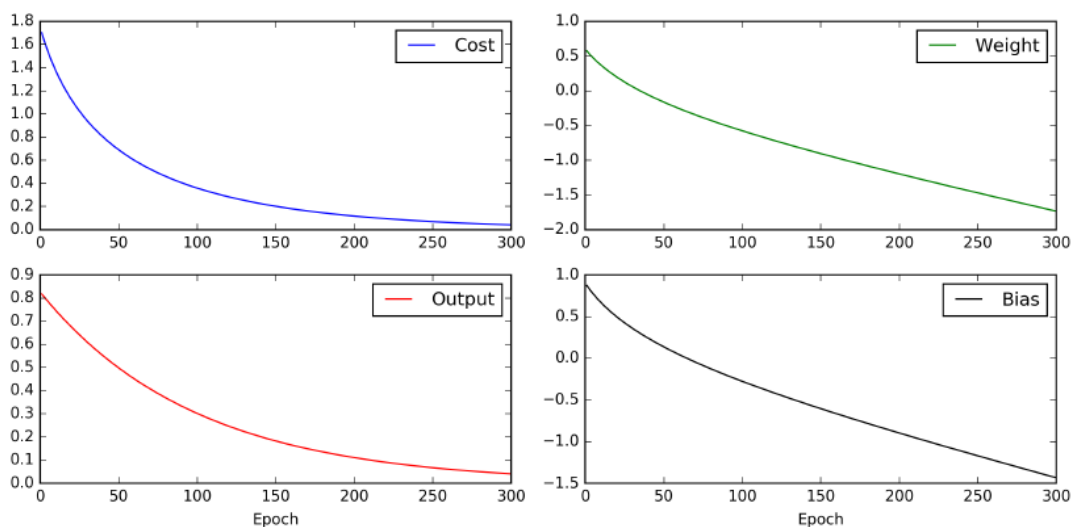
Η έκφραση αυτή μας λέει ότι ο ρυθμός με τον οποίο το βάρος μαθαίνεται, ελέγχεται από την ποσότητα $\sigma(z) - y$, δηλαδή από το σφάλμα στην έξοδο. Όσο μεγαλύτερο είναι το λάθος, τόσο πιο γρήγορα θα μάθει ο νευρώνας. Αυτό ακριβώς θα περιμέναμε διαισθητικά. Ειδικότερα αποφεύγει την επιβράδυνση της μάθησης που προκαλείται από τον όρο $\sigma'(z)$ στην αντίστοιχη εξίσωση για το τετραγωνικό κόστος (6.2). Όταν χρησιμοποιούμε τη διεντροπία, ο όρος $\sigma'(z)$ απαλείφεται και δεν χρειάζεται πλέον να ανησυχούμε ότι είναι μικρός. Αυτή η απαλοιφή είναι ακριβώς η ιδιότητα που εξασφαλίζεται από τη συνάρτηση κόστους διεντροπίας. Δεν πρόκειται για τυχαία ιδιότητα, καθώς η διεντροπία επιλέχθηκε ειδικά για τον λόγο αυτό.

Ανάλογα μπορούμε να υπολογίσουμε τη μερική παράγωγο για την πόλωση:

$$\frac{\partial C}{\partial b} = \frac{1}{n} \sum_x (\sigma(z) - y) \quad (6.8)$$

Και πάλι φαίνεται ότι αποφεύγεται η επιβράδυνση που προκαλείται από τον όρο $\sigma'(z)$ στην αντίστοιχη εξίσωση του τετραγωνικού κόστους (6.3).

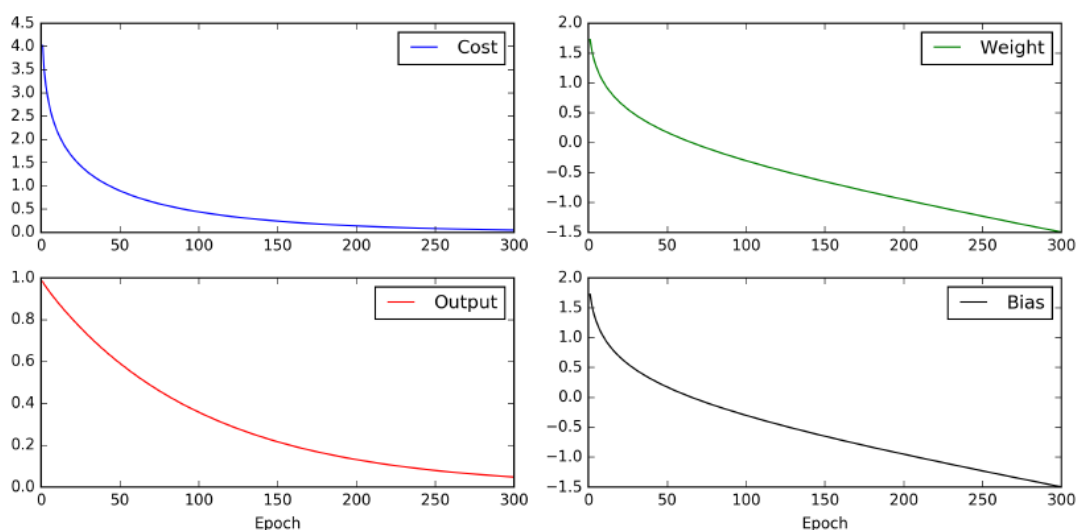
Ας επιστρέψουμε στο προηγούμενο παράδειγμα με τον έναν νευρώνα και ας διερευνήσουμε τι συμβαίνει όταν χρησιμοποιούμε τη διεντροπία αντί για το τετραγωνικό κόστος. Θα ξεκινήσουμε με την περίπτωση όπου το τετραγωνικό κόστος τα πήγε μια χαρά, με αρχικό βάρος 0,6 και αρχική πόλωση 0,9:



Σχήμα 6.6 Καμπύλες μάθησης με διεντροπία.

Όπως ήταν αναμενόμενο, ο νευρώνας μαθαίνει πολύ καλά σε αυτήν την περίπτωση, όπως και νωρίτερα.

Και τώρα ας δούμε την περίπτωση όπου ο νευρώνας μας κόλλησε πριν, με το βάρος και την πόλωση να ξεκινούν από 2,0:



Σχήμα 6.7 Καμπύλες μάθησης με διεντροπία και νέες αρχικές τιμές.

Αυτήν τη φορά ο νευρώνας έμαθε γρήγορα, όπως ελπίζαμε. Εάν παρατηρήσετε προσεκτικά, μπορείτε να δείτε ότι η κλίση της καμπύλης κόστους ήταν πολύ πιο απότομη στην αρχή της σε σχέση με την επίπεδη περιοχή στην αντίστοιχη καμπύλη για το τετραγωνικό κόστος. Είναι αυτή η απότομη αλλαγή που μας εξασφαλίζει η διεντροπία, εμποδίζοντας να κολλήσουμε, ακριβώς τη στιγμή που θα περιμέναμε ο νευρώνας μας να μάθει πιο γρήγορα, δηλαδή όταν ο νευρώνας ξεκινήσει πολύ λανθασμένα.

Μελετάμε τη διεντροπία για έναν μόνο νευρώνα. Ωστόσο, είναι εύκολο να γενικεύσουμε τη διεντροπία σε δίκτυα πολλαπλών επιπέδων πολλών νευρώνων. Συγκεκριμένα, ας υποθέσουμε $y=y_1, y_2, \dots$, είναι οι επιθυμητές τιμές στους νευρώνες εξόδου, δηλαδή οι νευρώνες στο τελευταίο επίπεδο, ενώ a_1^L, a_2^L, \dots , είναι οι πραγματικές τιμές εξόδου. Στη συνέχεια ορίζουμε τη διεντροπία ως:

$$C = -\frac{1}{n} \sum_x \sum_j [y_j \ln a_j^L + (1 - y_j) \ln(1 - a_j^L)] \quad (6.9)$$

Αυτό είναι το ίδιο με την προηγούμενη έκφραση (6.4), με τη διαφορά ότι τώρα έχουμε το \sum_j που αθροίζει όλους τους νευρώνες εξόδου. Είναι εμφανές ότι η χρήση της έκφρασης (6.9) αποφεύγει την επιβράδυνση της μάθησης σε δίκτυα πολλών νευρώνων και η αντίστοιχη παραγωγή δίνεται ως πρόβλημα παρακάτω.

Η χρήση του όρου διεντροπία μέχρι τώρα μπορεί να φαίνεται ότι έρχεται σε αντίθεση με ορισμένες πηγές. Ειδικότερα, είναι συνήθης ο ορισμός της διεντροπίας για δύο κατανομές πιθανοτήτων, p_j και q_j , ως $\sum_j p_j \ln q_j$. Αυτός ο ορισμός μπορεί να συνδεθεί με την (6.4), εάν θεωρήσουμε ότι ο ένας σιγμοειδής νευρώνας στην έξοδο δίνει μια κατανομή πιθανότητας που αποτελείται από την ενεργοποίηση του νευρώνα a και το συμπλήρωμά της $1-a$.

Ωστόσο, όταν έχουμε πολλούς σιγμοειδείς νευρώνες στην έξοδο, το διάνυσμα a_j^L των ενεργοποιήσεων δεν σχηματίζει συνήθως πιθανοτική κατανομή. Επομένως, ο ορισμός $\sum_j p_j \ln q_j$ δεν βγάζει νόημα, αφού δεν εργαζόμαστε με κατανομές πιθανότητας. Αντίθετα μπορούμε να δούμε την (6.9) ως το άθροισμα των διεντροπιών ανά νευρώνα και η ενεργοποίηση του καθενός μπορεί να ερμηνευθεί ως μέρος μιας κατανομής πιθανότητας δύο στοιχείων. Υπό αυτήν την έννοια η (6.9) αποτελεί γενίκευση της διεντροπίας για πιθανοτικές κατανομές.

Πότε πρέπει να χρησιμοποιήσουμε τη διεντροπία αντί για το τετραγωνικό κόστος; Στην πραγματικότητα η διεντροπία είναι σχεδόν πάντα η καλύτερη επιλογή, υπό την προϋπόθεση ότι οι νευρώνες εξόδου είναι σιγμοειδείς. Για να κατανοήσετε τον λόγο, σκεφτείτε ότι όταν δημιουργούμε το δίκτυο συνήθως αρχικοποιούμε τα βάρη και τις πόλωσης χρησιμοποιώντας κάποιο είδος τυχαιοποίησης. Μπορεί να προκύψει αυτές οι αρχικές

επιλογές να έχουν ως αποτέλεσμα το δίκτυο να είναι αποφασιστικά λάθος για κάποια είσοδο· δηλαδή ένας νευρώνας εξόδου θα έχει κορεστεί κοντά στο 1, όταν θα πρέπει να είναι 0 ή αντίστροφα. Αν χρησιμοποιήσουμε το τετραγωνικό κόστος, αυτό θα επιβραδύνει τη μάθηση. Δεν θα σταματήσει να μαθαίνει εντελώς, καθώς τα βάρη θα συνεχίσουν να μαθαίνουν από τις άλλες εισόδους, αλλά είναι προφανώς κάτι μη επιθυμητό.

Άσκηση 6.1.1. Διεντροπία για $0 < y < 1$.

Να δείξετε ότι η διεντροπία είναι επίσης καλή συνάρτηση κόστους όταν $0 < y < 1$.

Άσκηση 6.1.2. Δίκτυα πολλών νευρώνων πολλαπλών επιπέδων

Να δείξετε ότι για το τετραγωνικό κόστος η μερική παράγωγος ως προς τα βάρη στο επίπεδο εξόδου είναι $\frac{\partial C}{\partial w_{jk}^L} = \frac{1}{n} \sum_x a_k^{L-1} (a_j^L - y_j) \sigma'(z_j^L)$. Ο παράγοντας $\sigma'(z_j^L)$ προκαλεί επιβράδυνση της μάθησης όταν ένας νευρώνας εξόδου φτάνει σε κορεσμό στη λάθος τιμή. Να δείξετε ότι με συνάρτηση κόστους διεντροπίας η μερική παράγωγος ως προς τα βάρη γίνεται $\frac{\partial C}{\partial w_{jk}^L} = \frac{1}{n} \sum_x a_k^{L-1} (a_j^L - y_j)$ και ο όρος $\sigma'(z_j^L)$ έχει εξαφανιστεί, όπως ακριβώς και για την περίπτωση του απλού νευρώνα. Παρόμοια ανάλυση ισχύει φυσικά και για τις πολώσεις του δικτύου.

Άσκηση 6.1.3. Χρήση τετραγωνικής συνάρτησης κόστους όταν έχουμε γραμμικούς νευρώνες στην έξοδο.

Όταν οι νευρώνες εξόδου σε ένα πολυεπίπεδο ΤΝΔ είναι γραμμικοί, δηλαδή δεν εφαρμόζεται η σιγμοειδής συνάρτηση ενεργοποίησης, να δείξετε ότι η μερική παράγωγος της τετραγωνικής συνάρτησης κόστους ως προς τα βάρη και τις πολώσεις του επιπέδου εξόδου είναι

$$\begin{aligned} \frac{\partial C}{\partial w_{jk}^L} &= \frac{1}{n} \sum_x a_k^{L-1} (a_j^L - y_j) \\ \frac{\partial C}{\partial b_j^L} &= \frac{1}{n} \sum_x (a_j^L - y_j). \end{aligned}$$

Επομένως, στην περίπτωση γραμμικών εξόδων, το τετραγωνικό κόστος αποτελεί την κατάλληλη επιλογή συνάρτησης κόστους.

6.1.2 Τι σημαίνει διεντροπία; Από πού προέρχεται;

Η συζήτησή μας για τη διεντροπία επικεντρώθηκε στην αλγεβρική ανάλυση και την πρακτική εφαρμογή. Αυτό είναι χρήσιμο, αλλά αφήνει αναπάντητες ευρύτερες εννοιολογικές ερωτήσεις, όπως: Τι σημαίνει διεντροπία; Υπάρχει κάποιος διαισθητικός τρόπος σκέψης για τη διεντροπία; Και πώς θα μπορούσαμε να είχαμε φανταστεί τη συνάρτηση αυτή εξαρχής;

Ας ξεκινήσουμε με την τελευταία από αυτές τις ερωτήσεις: Τι θα μπορούσε να μας παρακινήσει να σκεφτούμε τη διεντροπία εξαρχής; Ας υποθέσουμε ότι είχαμε ανακαλύψει την επιβράδυνση μάθησης που περιγράφηκε νωρίτερα και καταλαβαίναμε ότι η προέλευση ήταν ο όρος $\sigma'(z)$ στις εξισώσεις (6.2) και (6.3). Αφού εξετάσουμε αυτές τις εξισώσεις για λίγο, μπορεί να αναρωτηθούμε αν είναι δυνατόν να επιλέξουμε μια συνάρτηση κόστους έτσι ώστε ο όρος $\sigma'(z)$ να εξαφανιστεί. Σε αυτήν την περίπτωση το κόστος $C=Cx$ για ένα μόνο δείγμα εκπαίδευσης x θα ικανοποιούσε

$$\begin{aligned} \frac{\partial C}{\partial w_j} &= x_j(a - y) \\ \frac{\partial C}{\partial b} &= (a - y). \end{aligned} \tag{6.10}, (6.11)$$

Αν μπορούσαμε να επιλέξουμε τη συνάρτηση κόστους για να κάνουμε αυτές τις εξισώσεις αληθείς, τότε θα αποτύπωναν με απλό τρόπο τη διαίσθηση ότι όσο μεγαλύτερο είναι το αρχικό σφάλμα, τόσο πιο γρήγορα μαθαίνει ο νευρώνας. Θα εξάλειφαν επίσης το πρόβλημα της επιβράδυνσης της μάθησης. Στην

πραγματικότητα, ξεκινώντας από αυτές τις εξισώσεις, θα δείξουμε τώρα ότι είναι δυνατόν να παράξουμε τον τύπο της διεντροπίας με μαθηματικό τρόπο. Από τον κανόνα της αλυσίδας έχουμε:

$$\frac{\partial C}{\partial b} = \frac{\partial C}{\partial a} \sigma'(z). \quad (6.12)$$

Αντικαθιστώντας $\sigma'(z) = \sigma(z)(1 - \sigma(z)) = a(1 - a)$ η προηγούμενη εξίσωση γίνεται

$$\frac{\partial C}{\partial b} = \frac{\partial C}{\partial a} a(1 - a). \quad (6.13)$$

Συγκρίνοντας με την (6.11) προκύπτει

$$\frac{\partial C}{\partial a} = \frac{a - y}{a(1 - a)}. \quad (6.14)$$

Το ολοκλήρωμα της έκφρασης αυτής ως προς το a είναι:

$$C = -[y \ln a + (1 - y) \ln(1 - a)] + \text{σταθερά}, \quad (6.15)$$

για κάποια σταθερά ολοκλήρωσης. Αυτή είναι η συνεισφορά στο κόστος ενός δείγματος εκπαίδευσης x . Για την πλήρη συνάρτηση κόστους χρειαζόμαστε τον μέσο όρο από όλα τα δείγματα, δηλαδή:

$$C = -\frac{1}{n} \sum_x [y \ln a + (1 - y) \ln(1 - a)] + \text{σταθερά}, \quad (6.16)$$

όπου η εδώ σταθερά είναι ο μέσος όρος των επιμέρους σταθερών για κάθε δείγμα. Επόμενως οι εξισώσεις (6.10) και (6.11) καθορίζουν μοναδικά τη μορφή της διεντροπίας μέχρι έναν συνολικό σταθερό όρο. Επομένως, η διεντροπία δεν είναι κάτι που προέκυψε ως εκ θαύματος, αλλά θα μπορούσαμε να την είχαμε ανακαλύψει με απλό και φυσικό τρόπο.

Τι γίνεται με το διαισθητικό νόημα της διεντροπίας; Αξίζει να σημειωθεί ότι υπάρχει ένας τυπικός τρόπος ερμηνείας της που προέρχεται από τον τομέα της θεωρίας της πληροφορίας. Σε γενικές γραμμές η ιδέα είναι ότι η διεντροπία είναι ένα μέτρο *έκπληξης*. Συγκεκριμένα, ο νευρώνας μας προσπαθεί να υπολογίσει τη συνάρτηση $x \rightarrow y = y(x)$, αλλά αντί αυτής υπολογίζει τη συνάρτηση $x \rightarrow a = a(x)$. Ας υποθέσουμε ότι θεωρούμε ως a την πιθανότητα εκτίμησης από την πλευρά του νευρώνα μας ότι το y είναι 1, και $1 - a$ την εκτιμώμενη πιθανότητα η σωστή τιμή για το y να είναι 0. Στη συνέχεια η διεντροπία μετράει πόσο «έκπληκτοι» είμαστε κατά μέσο όρο, όταν μαθαίνουμε την πραγματική τιμή για το y . Έχουμε χαμηλή έκπληξη εάν η έξοδος είναι αυτό που περιμένουμε και υψηλή έκπληξη εάν η έξοδος είναι απροσδόκητη. Τι ακριβώς όμως σημαίνει «έκπληξη»; Φαίνεται ότι υπάρχει ένας ακριβής τρόπος από τη θεωρία της πληροφορίας που ορίζει σαφώς αυτήν την ποσότητα, όπως φαίνεται και στο αντίστοιχο άρθρο της [Wikipedia](#).

6.1.3 Softmax

Σε αυτό το κεφάλαιο θα χρησιμοποιήσουμε κυρίως το κόστος της διεντροπίας για να αντιμετωπίσουμε το πρόβλημα της επιβράδυνσης της μάθησης. Μια διαφορετική προσέγγιση στο πρόβλημα αυτό βασίζεται στη χρήση των λεγόμενων επιπέδων εξόδου *softmax*. Θα χρησιμοποιήσουμε επίπεδα softmax στο κεφάλαιο 7, κατά τη συζήτηση των βαθέων νευρωνικών δικτύων.

Η ιδέα της softmax είναι να καθορίσει ένα νέο είδος επιπέδου εξόδου για τα νευρωνικά μας δίκτυα. Ξεκινάει με τον ίδιο τρόπο όπως με ένα σιγμοειδές επίπεδο, σχηματίζοντας τον γραμμικό συνδυασμό των εισόδων στον νευρώνα j του τελευταίου επιπέδου L ως $z_j^L = \sum_k w_{jk}^L a_k^{L-1} + b_j^L$. Ωστόσο, δεν εφαρμόζουμε τη σιγμοειδή συνάρτηση για να πάρουμε την έξοδο. Στη θέση αυτής σε ένα επίπεδο softmax εφαρμόζουμε τη λεγόμενη συνάρτηση softmax στο z_j^L . Σύμφωνα με αυτήν τη συνάρτηση η ενεργοποίηση a_j^L του νευρώνα εξόδου j είναι

$$a_j^L = \frac{e^{z_j^L}}{\sum_k e^{z_k^L}} \quad (6.17)$$

όπου στον παρονομαστή αθροίζουμε όλους τους νευρώνες εξόδου.

Δεν είναι προφανές γιατί επιθυμούμε τη χρήση αυτής της συνάρτησης, ούτε και το πώς βοηθά στο πρόβλημα της επιβράδυνσης της μάθησης. Το βασικό χαρακτηριστικό της softmax είναι ότι οι ενεργοποιήσεις σε όλους τους νευρώνες εξόδου θα αθροίζουν εξ ορισμού πάντα στο 1.

$$\sum_j a_j^L = \frac{\sum_j e^{z_j^L}}{\sum_k e^{z_k^L}} = 1 \quad (6.18)$$

Επομένως, όταν αυξάνεται π.χ. η a_4^L , οι άλλες ενεργοποιήσεις θα πρέπει να μειωθούν κατά την ίδια συνολική ποσότητα ώστε το άθροισμα όλων να διατηρηθεί 1.

Η εξίσωση (6.17) υποδηλώνει επίσης ότι οι ενεργοποιήσεις εξόδου είναι όλες θετικές, καθώς η εκθετική συνάρτηση είναι θετική. Συνδυάζοντας αυτό με την παρατήρηση στην προηγούμενη παράγραφο, βλέπουμε ότι η έξοδος από το επίπεδο softmax είναι ένα σύνολο θετικών αριθμών που αθροίζουν σε 1. Με άλλα λόγια η έξοδος από το επίπεδο softmax μπορεί να θεωρηθεί ως κατανομή πιθανότητας. Το γεγονός ότι ένα επίπεδο softmax παράγει μια κατανομή πιθανότητας είναι μάλλον ευχάριστο. Σε πολλά προβλήματα, όπως αυτό της ταξινόμησης *μονής ετικέτας πολλαπλών κλάσεων (single-label multi-class classification)*, είναι βολικό να μπορείτε να ερμηνεύσετε την ενεργοποίηση εξόδου a_j^L ως εκτίμηση του δικτύου για την πιθανότητα η σωστή έξοδος να είναι j . Έτσι π.χ. στο πρόβλημα ταξινόμησης MNIST μπορούμε να ερμηνεύσουμε το a_j^L ως την εκτιμώμενη πιθανότητα του δικτύου η σωστή ταξινόμηση του ψηφίου να είναι j . Αντίθετα, αν το επίπεδο εξόδου ήταν σιγμοειδές, τότε σίγουρα δεν θα μπορούσαμε να υποθέσουμε ότι οι ενεργοποιήσεις σχηματίζουν μια κατανομή πιθανοτήτων, αφού δεν είναι εγγυημένο ότι αθροίζουν πάντα στο 1. Μπορούμε να σκεφτούμε την softmax ως έναν τρόπο για αλλαγή κλίμακας στις z_j^L και κατόπιν κανονικοποίησής τους ώστε να σχηματίζουν πιθανοτική κατανομή.

Για να δούμε πώς η softmax βοηθά στο πρόβλημα της επιβράδυνσης της μάθησης, ας ορίσουμε τη συνάρτηση κόστους *λογαριθμικής πιθανοφάνειας (log-likelihood)*. Το x δηλώνει ένα δείγμα εκπαίδευσης και το y την αντίστοιχη επιθυμητή έξοδο. Το κόστος λογαριθμικής πιθανοφάνειας γι' αυτό το δείγμα θα είναι:

$$C \equiv -\ln a_y^L \quad (6.19)$$

Έτσι π.χ. εάν εκπαιδεύουμε με εικόνες MNIST και εισάγουμε μια εικόνα ενός 7, τότε το κόστος λογαριθμικής πιθανοφάνειας είναι $-\ln a_7^L$. Για να δείτε ότι αυτό ισχύει διαισθητικά, εξετάστε την περίπτωση όπου το δίκτυο κάνει καλή δουλειά, δηλαδή είναι βέβαιο ότι η είσοδος είναι 7. Σε αυτήν την περίπτωση θα υπολογίσει μια τιμή για την αντίστοιχη πιθανότητα a_7^L που είναι κοντά στο 1, άρα το κόστος $-\ln a_7^L$ θα είναι μικρό. Αντίθετα, όταν το δίκτυο δεν κάνει τόσο καλή δουλειά, η πιθανότητα a_7^L θα είναι μικρότερη και το κόστος $-\ln a_7^L$ θα είναι μεγαλύτερο. Επομένως, το κόστος λογαριθμικής πιθανοφάνειας συμπεριφέρεται όπως θα περιμέναμε να συμπεριφέρεται μια συνάρτηση κόστους.

Τι γίνεται με το πρόβλημα της επιβράδυνσης της μάθησης; Για να το αναλύσουμε αυτό, υπενθυμίζουμε ότι το κλειδί για την επιβράδυνση της μάθησης είναι η συμπεριφορά των ποσοτήτων $\partial C / \partial w_{jk}^L$ και $\partial C / \partial b_j^L$. Μπορεί ναδειχθεί ότι (εδώ το y δηλώνει το διάνυσμα των ενεργοποιήσεων εξόδου που αντιστοιχεί στον επιθυμητό αριθμό, 7, δηλαδή ένα διάνυσμα όπου όλα τα στοιχεία του είναι 0 εκτός από ένα 1 στην 7η θέση):

$$\begin{aligned} \frac{\partial C}{\partial b_j^L} &= a_j^L - y_j \\ \frac{\partial C}{\partial w_{jk}^L} &= a_k^{L-1} (a_j^L - y_j) \end{aligned} \quad (6.20), (6.21)$$

Αυτές οι εξισώσεις είναι οι ίδιες με τις ανάλογες εκφράσεις που ελήφθησαν στην προηγούμενη ανάλυση της διεντροπίας. Όπως και εκεί, αυτές οι εκφράσεις διασφαλίζουν ότι δεν θα αντιμετωπίσουμε μια επιβράδυνση της μάθησης. Μπορούμε να σκεφτούμε ένα επίπεδο εξόδου softmax με κόστος λογαριθμικής πιθανοφάνειας ως αρκετά παρόμοιο με ένα σιγμοειδές επίπεδο εξόδου με κόστος διεντροπίας. Δεδομένης αυτής της ομοιότητας θα πρέπει να χρησιμοποιούμε ένα σιγμοειδές επίπεδο εξόδου και διεντροπία ή ένα επίπεδο εξόδου softmax και λογαριθμική πιθανοφάνεια;

Στην πραγματικότητα σε πολλές περιπτώσεις και οι δύο προσεγγίσεις λειτουργούν καλά. Κατά τη διάρκεια του υπόλοιπου κεφαλαίου θα χρησιμοποιήσουμε ένα σιγμοειδές επίπεδο εξόδου με κόστος διεντροπίας. Αργότερα, στο κεφάλαιο 7, κάποιες φορές θα χρησιμοποιήσουμε ένα επίπεδο εξόδου softmax, με κόστος λογαριθμικής πιθανοφάνειας. Γενικότερα softmax και λογαριθμική πιθανοφάνεια αξίζει να χρησιμοποιούμε κάθε φορά που θέλουμε να ερμηνεύσουμε τις εξόδους ως πιθανότητες. Δεν είναι πάντα απαραίτητο, αλλά είναι χρήσιμο σε προβλήματα ταξινόμησης μονής ετικέτας, που περιλαμβάνουν ξένες μεταξύ τους κλάσεις (όπως το MNIST).

Άσκηση 6.1.4. Πίσω διάδοση με softmax και κόστος λογαριθμικής πιθανοφάνειας.

Για την εφαρμογή του αλγορίθμου πίσω διάδοσης σε ένα επίπεδο softmax θα πρέπει να καθοριστεί μια έκφραση για την τοπική κλίση του σφάλματος στους νευρώνες εξόδου $\delta_j^L \equiv \partial C / \partial z_j^L$. Να δείξετε ότι μια κατάλληλη έκφραση είναι $\delta_j^L = a_j^L - y_j$. Με την έκφραση αυτή μπορούμε να εφαρμόσουμε τον αλγόριθμο οπισθοδιάδοσης σε ένα δίκτυο που χρησιμοποιεί έξοδο softmax και κόστος λογαριθμικής πιθανοφάνειας.

6.2 Υπερεκπαίδευση και ομαλοποίηση

Όπως είδαμε στην ενότητα 4.5, η υπερεκπαίδευση (ή υπερπροσαρμογή, overfitting) αποτελεί μείζον πρόβλημα στα νευρωνικά δίκτυα. Αυτό ισχύει ιδιαίτερα στα σύγχρονα δίκτυα, τα οποία συχνά έχουν πολύ μεγάλο αριθμό βαρών και πολώσεων. Για αποτελεσματική εκπαίδευση χρειαζόμαστε έναν τρόπο να εντοπίζουμε πότε προκύπτει υπερεκπαίδευση, καθώς και τεχνικές για να μειώσουμε τα αποτελέσματα της επίδρασής της.

Γενικά ένας καλός τρόπος για μείωση της υπερεκπαίδευσης είναι η αύξηση του συνόλου εκπαίδευσης. Όταν είναι πολλά δεδομένα διαθέσιμα, είναι δύσκολο ακόμα και για ένα πολύ μεγάλο δίκτυο να υπερεκπαιδευτεί. Συχνά όμως τα δεδομένα είναι δύσκολο ή ακριβό να αποκτηθούν, οπότε η αύξηση αυτή δεν αποτελεί πάντα πρακτική λύση.

Για τον εντοπισμό της υπερεκπαίδευσης ένας προφανής τρόπος είναι να παρακολουθούμε την ακρίβεια στο σύνολο ελέγχου, όσο εκπαιδευτεί το δίκτυο. Αν η ακρίβεια στο σύνολο ελέγχου πάψει να βελτιώνεται, θα πρέπει να σταματήσουμε την εκπαίδευση. Από την άλλη κάτι τέτοιο δεν είναι απαραίτητα δείγμα υπερεκπαίδευσης: μπορεί να τύχει η ακρίβεια στα σύνολα εκπαίδευσης και ελέγχου να σταματήσει να βελτιώνεται την ίδια στιγμή. Παρ' όλα αυτά, αποτελεί μια καλή στρατηγική για την αντιμετώπιση της υπερεκπαίδευσης.

Εκτός από το σύνολο ελέγχου, συχνά χρησιμοποιείται κι ένα ξεχωριστό σύνολο δεδομένων, διαφορετικό από τα σύνολα εκπαίδευσης και ελέγχου, που ονομάζεται *σύνολο επικύρωσης* (validation set). Αν τα δεδομένα δεν είναι πολλά, τότε το αρχικό σύνολο εκπαίδευσης μπορεί να αξιοποιηθεί και ως σύνολο επικύρωσης, με τη μέθοδο της διασταυρούμενης επικύρωσης που είδαμε στην Ενότητα 4.6. Για να εντοπίσουμε πιθανή υπερεκπαίδευση, αντί για το σύνολο ελέγχου χρησιμοποιούμε το σύνολο επικύρωσης, ακολουθώντας την ίδια στρατηγική, δηλαδή υπολογίζουμε το σφάλμα ή την ακρίβεια ταξινόμησης στο σύνολο επικύρωσης μετά από κάθε εποχή. Μόλις η ακρίβεια δείχνει να έχει κορεστεί (δεν βελτιώνεται άλλο), τότε η εκπαίδευση σταματά. Η στρατηγική αυτή ονομάζεται *πρόωρο σταμάτημα* (early stopping).

Γιατί να χρησιμοποιήσουμε το σύνολο επικύρωσης αντί του συνόλου ελέγχου; Στην πραγματικότητα, αυτό είναι μέρος μιας γενικότερης στρατηγικής, η οποία είναι η χρήση του συνόλου επικύρωσης για την αξιολόγηση διαφορετικών δοκιμαστικών επιλογών υπερπαραμέτρων, όπως ο αριθμός των εποχών για τις οποίες θα γίνει εκπαίδευση, ο ρυθμός μάθησης, η καλύτερη αρχιτεκτονική δικτύου και ούτω καθεξής. Χρησιμοποιούμε τέτοιες αξιολογήσεις για να βρούμε και να ορίσουμε καλές τιμές για τις υπερπαραμέτρους.

Κατά τον καθορισμό υπερπαραμέτρων είναι πιθανό να δοκιμάσουμε πολλές διαφορετικές επιλογές για αυτές. Αν ορίσουμε τις υπερπαραμέτρους με βάση τις αξιολογήσεις του συνόλου ελέγχου, είναι πιθανό να καταλήξουμε να υπερπροσαρμόζουμε τις υπερπαραμέτρους μας στο σύνολο ελέγχου· δηλαδή μπορεί να καταλήξουμε να βρίσκουμε υπερπαραμέτρους που ταιριάζουν σε συγκεκριμένες ιδιαιτερότητες του συνόλου

ελέγχου, αλλά όπου η απόδοση του δικτύου δεν θα γενικεύει σε άλλα σύνολα δεδομένων. Προφυλάσσουμε το δίκτυο από αυτό βρίσκοντας υπερπαραμέτρους με χρήση του συνόλου επικύρωσης. Στη συνέχεια, μόλις έχουμε τις υπερπαραμέτρους που θέλουμε, κάνουμε μια τελική αξιολόγηση της ακρίβειας χρησιμοποιώντας το σύνολο ελέγχου. Αυτό μας δίνει τη βεβαιότητα ότι τα αποτελέσματά μας στο σύνολο ελέγχου είναι ένα πραγματικό μέτρο για το πόσο καλά γενικεύει το νευρωνικό μας δίκτυο. Μπορείτε να σκεφτείτε τα δεδομένα επικύρωσης ως ένα είδος δεδομένων εκπαίδευσης που μας βοηθά να μάθουμε καλές υπερπαραμέτρους.

6.2.1 Ομαλοποίηση

Η αύξηση των δεδομένων εκπαίδευσης είναι ένας τρόπος μείωσης της υπερεκπαίδευσης. Υπάρχουν άλλοι τρόποι με τους οποίους μπορούμε να μειώσουμε τον βαθμό στον οποίο συμβαίνει υπερεκπαίδευση; Μια πιθανή προσέγγιση είναι η μείωση του μεγέθους του δικτύου μας. Ωστόσο, τα μεγάλα δίκτυα έχουν τη δυνατότητα να είναι πιο ισχυρά από τα μικρά δίκτυα, έτσι αυτή είναι μια επιλογή που θα υιοθετούσαμε μόνο απρόθυμα.

Ευτυχώς υπάρχουν και άλλες τεχνικές που μπορούν να μειώσουν την υπερεκπαίδευση, ακόμη και όταν έχουμε ένα σταθερό δίκτυο και σταθερά δεδομένα εκπαίδευσης. Αυτές είναι γνωστές ως τεχνικές *ομαλοποίησης* (regularization). Στη συνέχεια περιγράφεται μία από τις πιο συχνά χρησιμοποιούμενες τεχνικές ομαλοποίησης, γνωστή ως φθορά βαρών (weight decay ή ομαλοποίηση L2). Η ιδέα της ομαλοποίησης L2 είναι να προστεθεί ένας επιπλέον όρος στη συνάρτηση κόστους, που ονομάζεται όρος ομαλοποίησης. Στη συνέχεια δίνεται ο τύπος της ομαλοποιημένης διασταυρούμενης εντροπίας:

$$C = -\frac{1}{n} \sum_{xj} [y_j \ln a_j^L + (1 - y_j) \ln(1 - a_j^L)] + \frac{\lambda}{2n} \sum_w w^2 \quad (6.22)$$

Ο πρώτος όρος είναι απλώς η συνήθης έκφραση της διεντροπίας. Αλλά έχουμε προσθέσει και έναν δεύτερο όρο, το άθροισμα των τετραγώνων όλων των βαρών του δικτύου. Ο όρος αυτός κλιμακώνεται με έναν παράγοντα $\lambda/2n$, όπου $\lambda > 0$ καλείται παράμετρος ομαλοποίησης και n το πλήθος των δειγμάτων εκπαίδευσης. Με τον ίδιο τρόπο μπορούν να ομαλοποιηθούν και άλλες συναρτήσεις κόστους, όπως το τετραγωνικό κόστος:

$$C = \frac{1}{2n} \sum_x \|y - a^L\|^2 + \frac{\lambda}{2n} \sum_w w^2 \quad (6.23)$$

Και στις δύο περιπτώσεις μπορούμε να γράψουμε την ομαλοποιημένη συνάρτηση κόστους ως:

$$C = C_0 + \frac{\lambda}{2n} \sum_w w^2 \quad (6.24)$$

όπου C_0 είναι η αρχική, μη ομαλοποιημένη συνάρτηση κόστους.

Διασθητικά η επίδραση της ομαλοποίησης είναι να αναγκάσει το δίκτυο να μάθει μικρά βάρη. Μεγάλα βάρη θα επιτραπούν μόνο αν βελτιώνουν σημαντικά τον πρώτο όρο της συνάρτησης κόστους, ενώ τα υπόλοιπα βάρη που δεν έχουν τόσο σημαντική επίδραση θα τείνουν στο μηδέν. Η ομαλοποίηση είναι επομένως ένας συμβιβασμός ανάμεσα στην εκμάθηση μικρών έως και μηδενικών βαρών από τη μία και στην ελαχιστοποίηση του κόστους από την άλλη. Η σημαντικότητα κάθε μέρους του συμβιβασμού εξαρτάται από την τιμή της παραμέτρου λ : όταν είναι μικρή, προτιμάμε να ελαχιστοποιήσουμε το αρχικό κόστος, ενώ όταν είναι μεγάλη προτιμάμε τα μικρά βάρη.

Ως όφελος της ομαλοποίησης μπορεί να θεωρηθεί η απλούστευση του δικτύου, εφόσον τα βάρη των λιγότερο σημαντικών συνάψεων θα τείνουν στο μηδέν. Παράλληλα το γεγονός ότι βάρη κρατιούνται μικρά σημαίνει ότι το δίκτυο δεν θα τροποποιείται πολύ λόγω τυχαίων αλλαγών σε μερικές εισόδους. Ένα ομαλοποιημένο δίκτυο θα δυσκολευτεί να μάθει το αποτέλεσμα από τυχόν τοπικό θόρυβο στα δεδομένα. Αντίθετα ένα δίκτυο με μεγάλα βάρη θα αλλάζει τη συμπεριφορά του σε μεγάλο βαθμό όταν γίνονται ακόμα και μικρές αλλαγές στην είσοδο. Επομένως, ένα μη ομαλοποιημένο δίκτυο μπορεί να χρησιμοποιήσει μεγάλα βάρη για να μάθει ένα περίπλοκο μοντέλο που φέρει πολλή πληροφορία για τον θόρυβο στα δεδομένα εκπαίδευσης. Ουσιαστικά τα ομαλοποιημένα δίκτυα αναγκάζονται να χτίσουν σχετικά απλά μοντέλα βάσει των μοτίβων που απαντούν στα δεδομένα εκπαίδευσης και είναι ανθεκτικά στις ιδιαιτερότητες του θορύβου στα

δεδομένα. Η προσδοκία είναι πως έτσι το δίκτυο πραγματικά θα μάθει για το εκάστοτε φαινόμενο και ότι θα γενικεύει καλύτερα από αυτά που έχει μάθει.

Ας δούμε πώς μπορεί να εφαρμοστεί η ομαλοποίηση στον αλγόριθμο κατάβασης κλίσης. Παίρνοντας τις μερικές παραγώγους των βαρών και των πολώσεων από την εξίσωση (6.24) έχουμε:

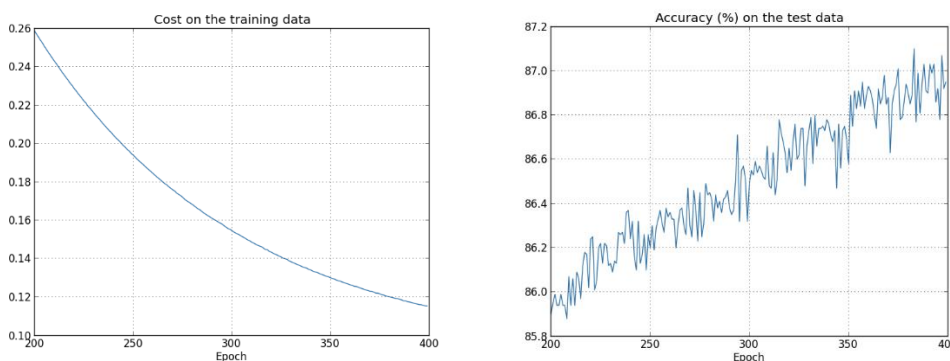
$$\begin{aligned}\frac{\partial C}{\partial w} &= \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} w \\ \frac{\partial C}{\partial b} &= \frac{\partial C_0}{\partial b}\end{aligned}$$

Οι μερικές παράγωγοι $\frac{\partial C_0}{\partial w}$ και $\frac{\partial C_0}{\partial b}$ μπορούν να υπολογιστούν με πίσω διάδοση. Επομένως, είναι εύκολο να υπολογιστεί η κλίση της ομαλοποιημένης συνάρτησης κόστους: χρησιμοποιούμε ως συνήθως πίσω διάδοση και μετά απλώς προσθέτουμε $\frac{\lambda}{n} w$ στη μερική παράγωγο για κάθε βάρος. Ο κανόνας μάθησης για τα βάρη γίνεται:

$$\begin{aligned}w &\rightarrow w - \eta \frac{\partial C_0}{\partial w} - \frac{\eta \lambda}{n} w \\ &= \left(1 - \frac{\eta \lambda}{n}\right) w - \eta \frac{\partial C_0}{\partial w}\end{aligned}\tag{6.25}$$

που είναι το ίδιο με πριν, μόνο που έχει κλιμακωθεί το βάρος w με έναν παράγοντα $\left(1 - \frac{\eta \lambda}{n}\right)$. Αυτή η κλιμάκωση είναι γνωστή ως φθορά βαρών, αφού κάνει τα βάρη μικρότερα. Μπορεί εσφαλμένα να θεωρηθεί ότι αυτή η κλιμάκωση οδηγεί όλα τα βάρη διαρκώς προς το μηδέν, ωστόσο αυτό δεν ισχύει, γιατί ο άλλος όρος μπορεί να προκαλέσει αύξηση του βάρους, αν αυτό βοηθά στη μείωση της αρχικής συνάρτησης κόστους.

Ας δούμε πώς η ομαλοποίηση αλλάζει την απόδοση του νευρικού μας δικτύου. Θα χρησιμοποιήσουμε ένα δίκτυο με 30 κρυμμένους νευρώνες, μεγέθους mini-batch 10, με ρυθμό μάθησης 0.5, και τη συνάρτηση κόστους διασταυρούμενης εντροπίας. Ωστόσο αυτήν τη φορά θα χρησιμοποιήσουμε μια παράμετρο ομαλοποίησης $\lambda=0,1$. Το κόστος στα δεδομένα εκπαίδευσης μειώνεται καθ' όλη τη διάρκεια του χρόνου, όπως και στην προηγούμενη, μη ομαλοποιημένη περίπτωση. Αλλά τώρα η ακρίβεια στο σύνολο ελέγχου συνεχίζει να αυξάνεται για όλες τις 400 εποχές:



Σχήμα 6.8 Κόστος και ακρίβεια στα σύνολα εκπαίδευσης και ελέγχου αντίστοιχα.

Είναι σαφές ότι η χρήση της ομαλοποίησης έχει καταστείλει την υπερεκπαίδευση. Επιπλέον, η ακρίβεια είναι σημαντικά υψηλότερη, με μέγιστη ακρίβεια ταξινόμησης 87.1%, σε σύγκριση με το μέγιστο 82.27% που αποκτήθηκε στη μη ομαλοποιημένη περίπτωση. Πράγματι είναι σχεδόν βέβαιο ότι θα μπορούσαμε να έχουμε σημαντικά καλύτερα αποτελέσματα συνεχίζοντας την εκπαίδευση και μετά τις 400 εποχές. Φαίνεται ότι εμπειρικά η ομαλοποίηση αναγκάζει το δίκτυό μας να γενικεύσει καλύτερα και να μειώσει σημαντικά τις επιπτώσεις της υπερεκπαίδευσης.

6.2.2 Άλλες τεχνικές ομαλοποίησης

Υπάρχουν πολλές τεχνικές ομαλοποίησης εκτός από την ομαλοποίηση L2. Σε αυτήν την ενότητα περιγράφονται εν συντομία τρεις άλλες προσεγγίσεις για τη μείωση της υπερεκπαίδευσης: η ομαλοποίηση L1, το dropout και η τεχνητή αύξηση του μεγέθους του συνόλου εκπαίδευσης. Σκοπός είναι να υπάρξει εξοικείωση με τις κύριες ιδέες και να εκτιμηθεί η ποικιλομορφία των διαθέσιμων τεχνικών ομαλοποίησης.

L1 ομαλοποίηση: Σε αυτήν την προσέγγιση τροποποιούμε τη μη ομαλοποιημένη συνάρτηση κόστους προσθέτοντας το άθροισμα των απόλυτων τιμών των βαρών:

$$C = C_0 + \frac{\lambda}{n} \sum_w |w| \quad (6.26)$$

Διαισθητικά αυτό είναι παρόμοιο με την ομαλοποίηση L2, τιμωρώντας μεγάλα βάρη και ωθώντας το δίκτυο να προτιμά μικρά βάρη. Φυσικά ο όρος ομαλοποίησης L1 δεν είναι ο ίδιος με τον όρο ομαλοποίησης L2 και έτσι δεν πρέπει να περιμένουμε να έχουμε ακριβώς την ίδια συμπεριφορά. Ας προσπαθήσουμε να κατανοήσουμε πώς η συμπεριφορά ενός δικτύου που εκπαιδεύεται με τη χρήση της ομαλοποίησης L1 διαφέρει από ένα δίκτυο που εκπαιδεύεται χρησιμοποιώντας την ομαλοποίηση L2.

Για να γίνει αυτό, θα εξετάσουμε τις μερικές παραγώγους της συνάρτησης κόστους. Διαφορίζοντας την (6.26) λαμβάνουμε:

$$\frac{\partial C}{\partial w} = \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} \text{sgn}(w), \quad (6.27)$$

όπου $\text{sgn}(w)$ είναι το πρόσημο του w , δηλαδή $+1$ αν w είναι θετικό και -1 εάν w είναι αρνητικό. Χρησιμοποιώντας αυτήν την έκφραση, μπορούμε εύκολα να τροποποιήσουμε τον αλγόριθμο πίσω διάδοσης για να κάνουμε στοχαστική κατάβαση κλίσης χρησιμοποιώντας ομαλοποίηση L1. Ο κανόνας ενημέρωσης που προκύπτει για ένα δίκτυο με L1 ομαλοποίηση είναι:

$$w \rightarrow w' = w - \frac{\eta\lambda}{n} \text{sgn}(w) - \eta \frac{\partial C_0}{\partial w} \quad (6.28)$$

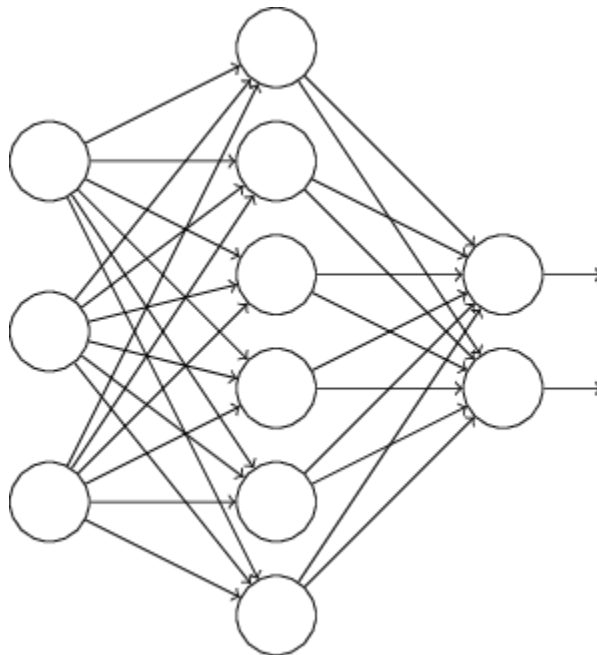
Ας τον συγκρίνουμε με τον κανόνα ενημέρωσης για την ομαλοποίηση L2:

$$w \rightarrow w' = w \left(1 - \frac{\eta\lambda}{n}\right) - \eta \frac{\partial C_0}{\partial w} \quad (6.29)$$

Και στις δύο εκφράσεις η επίδραση της ομαλοποίησης είναι να συρρικνωθούν τα βάρη. Αυτό συνάδει με τη διαίσθησή μας ότι και τα δύο είδη ομαλοποίησης τιμωρούν μεγάλα βάρη. Αλλά ο τρόπος που συρρικνώνονται τα βάρη είναι διαφορετικός. Στην ομαλοποίηση L1 τα βάρη συρρικνώνονται κατά ένα σταθερό ποσό προς το 0. Στην ομαλοποίηση L2 τα βάρη συρρικνώνονται κατά ένα ποσό που είναι ανάλογο με το w . Έτσι, όταν ένα συγκεκριμένο βάρος έχει μεγάλο απόλυτο μέγεθος, $|w|$, η L1 ομαλοποίηση συρρικνώνει το βάρος πολύ λιγότερο από ότι η L2 ομαλοποίηση. Αντίθετα, όταν το $|w|$ είναι μικρό, η ομαλοποίηση L1 συρρικνώνει το βάρος πολύ περισσότερο από την ομαλοποίηση L2. Το αποτέλεσμα είναι ότι η ομαλοποίηση L1 τείνει να συγκεντρώσει το βάρος του δικτύου σε έναν σχετικά μικρό αριθμό συνάψεων υψηλής σημασίας, ενώ τα άλλα βάρη οδηγούνται προς το μηδέν.

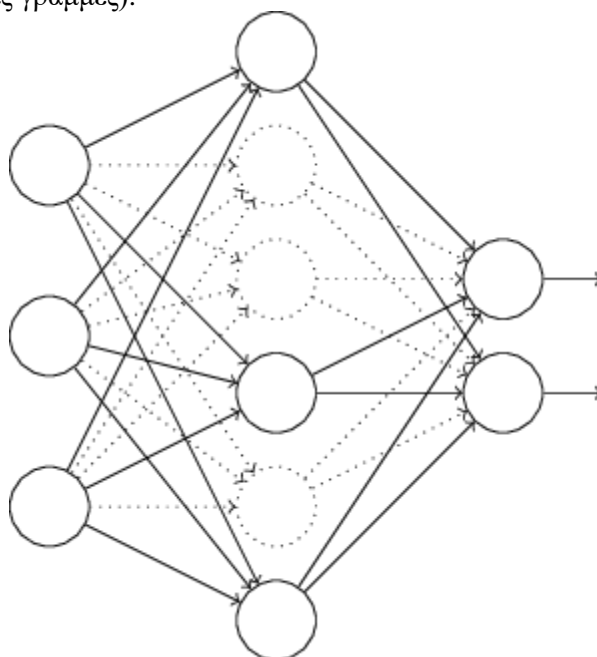
Θα πρέπει να σημειωθεί ότι στις παραπάνω εξισώσεις η μερική παράγωγος $\frac{\partial C}{\partial w}$ δεν ορίζεται όταν $w = 0$. Ο λόγος είναι ότι η συνάρτηση $|w|$ έχει μια απότομη «γωνία» στο $w = 0$ και έτσι δεν είναι διαφορίσιμη σε αυτό το σημείο. Το πρόβλημα αυτό παρακάμπτεται αν απλώς εφαρμόσουμε τον συνήθη (μη ομαλοποιημένο) κανόνα για στοχαστική κατάβαση κλίσης όταν $w = 0$. Διαισθητικά το αποτέλεσμα της ομαλοποίησης είναι να συρρικνώσει τα βάρη και προφανώς δεν μπορεί να συρρικνώσει ένα βάρος που είναι ήδη 0. Πιο συγκεκριμένα, θα χρησιμοποιήσουμε τις εξισώσεις (6.27) και (6.28) με τη σύμβαση ότι $\text{sgn}(0) = 0$.

Dropout: Είναι μια ριζικά διαφορετική τεχνική για την ομαλοποίηση. Σε αντίθεση με την ομαλοποίηση L1 και L2, το dropout δεν βασίζεται στην τροποποίηση της συνάρτησης κόστους. Αντ' αυτού τροποποιούμε το ίδιο το δίκτυο. Για να καταλάβουμε πώς λειτουργεί το dropout ας υποθέσουμε ότι έχουμε να εκπαιδεύσουμε ένα δίκτυο:



Σχήμα 6.9 Νευρωνικό δίκτυο με κρυφό επίπεδο.

Συγκεκριμένα ας υποθέσουμε ότι έχουμε ένα δείγμα εκπαίδευσης x και την αντίστοιχη επιθυμητή έξοδο y . Κανονικά η εκπαίδευση θα γινόταν με την εμπρός διάδοση του x μέσω του δικτύου και στη συνέχεια με πίσω διάδοση για να καθοριστεί η συμβολή του στην κλίση. Με το dropout αυτή η διαδικασία τροποποιείται. Ξεκινάμε διαγράφοντας τυχαία (και προσωρινά) τους μισούς κρυμμένους νευρώνες στο δίκτυο, αφήνοντας τους νευρώνες εισόδου και εξόδου ανέγγιχτους. Μετά από αυτό θα καταλήξουμε με ένα δίκτυο όπως παρακάτω. Σημειώστε ότι οι νευρώνες dropout, δηλαδή οι νευρώνες που έχουν διαγραφεί προσωρινά, εξακολουθούν να φαίνονται (με διακεκομμένες γραμμές):



Σχήμα 6.10 Εφαρμογή dropout.

Η είσοδος x τροφοδοτείται στο τροποποιημένο δίκτυο και στη συνέχεια γίνεται πίσω διάδοση του αποτελέσματος. Αφού γίνει αυτό για μια δεσμίδα δειγμάτων, ενημερώνουμε τα κατάλληλα βάρη και τις πολώσεις. Η διαδικασία επαναλαμβάνεται, πρώτα επαναφέροντας τους εγκαταλειμμένους νευρώνες, μετά διαλέγοντας ένα νέο τυχαίο υποσύνολο κρυφών νευρώνων για διαγραφή και στη συνέχεια υπολογίζοντας την κλίση για μια διαφορετική δεσμίδα δειγμάτων, οπότε και ενημερώνονται τα βάρη και οι πολώσεις του δικτύου.

Επαναλαμβάνοντας αυτήν τη διαδικασία ξανά και ξανά, το δίκτυό μας θα μάθει ένα σύνολο βαρών και πολώσεων. Φυσικά αυτά τα βάρη και οι πολώσεις θα έχουν προκύψει υπό συνθήκες στις οποίες οι μισοί κρυμμένοι νευρώνες αποσύρθηκαν. Όταν στην πραγματικότητα τρέχουμε το πλήρες δίκτυο, αυτό σημαίνει ότι διπλάσιοι κρυφοί νευρώνες θα είναι ενεργοί. Για να το αντισταθμίσουμε, μειώνουμε κατά το ήμισυ τα βάρη που προέρχονται από τους κρυμμένους νευρώνες.

Αυτή η διαδικασία dropout μπορεί να φαίνεται παράξενη και ad hoc. Γιατί να περιμένουμε να βοηθήσει στην ομαλοποίηση; Φανταστείτε ότι εκπαιδεύουμε πολλά διαφορετικά νευρωνικά δίκτυα, χρησιμοποιώντας σε όλα τα ίδια δεδομένα εκπαίδευσης. Φυσικά τα δίκτυα μπορεί να μην ξεκινούν πανομοιότυπα και ως αποτέλεσμα μετά την εκπαίδευση μπορεί μερικές φορές να δώσουν διαφορετικά αποτελέσματα. Όταν συμβεί αυτό, θα μπορούσαμε να χρησιμοποιήσουμε κάποιο είδος σχήματος μέσου όρου ή ψηφοφορίας για να αποφασίσουμε ποια έξοδο θα δεχτούμε. Π.χ. αν έχουμε εκπαιδέψει πέντε δίκτυα και τρία από αυτά ταξινομούν ένα ψηφίο ως «3», τότε πιθανότατα είναι πραγματικά ένα «3». Τα άλλα δύο δίκτυα μάλλον κάνουν λάθος. Αυτό το είδος σχήματος μέσου όρου συχνά θεωρείται ένας ισχυρός (αν και ακριβός) τρόπος μείωσης της υπερπροσφοράς. Ο λόγος είναι ότι τα διαφορετικά δίκτυα μπορεί να παρουσιάσουν υπερεκπαίδευση με διαφορετικούς τρόπους και ο μέσος όρος μπορεί να βοηθήσει στην εξάλειψη αυτού του είδους υπερεκπαίδευσης.

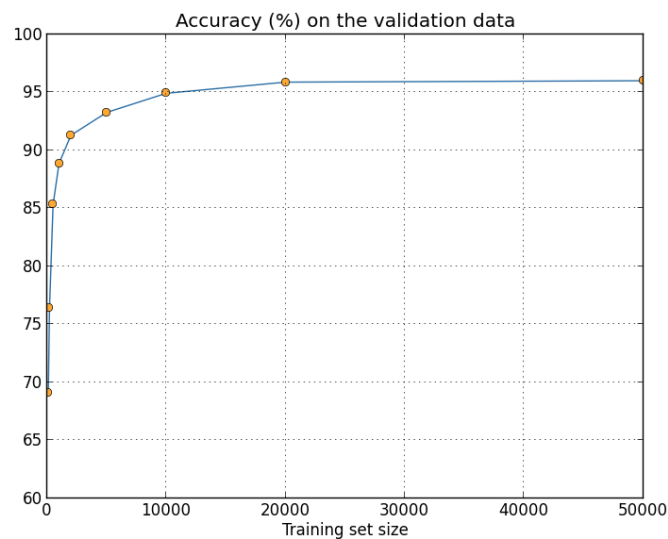
Τι σχέση έχει αυτό με το dropout; Ευρετικά, όταν διαγράφουμε διαφορετικά σύνολα νευρώνων, είναι μάλλον σαν να εκπαιδεύουμε διαφορετικά νευρωνικά δίκτυα. Άρα η διαδικασία dropout είναι σαν τον μέσο όρο των επιπτώσεων ενός πολύ μεγάλου αριθμού διαφορετικών δικτύων. Τα διάφορα δίκτυα θα υπερεκπαιδευτούν με διαφορετικούς τρόπους και έτσι —ελπίζουμε— το καθαρό αποτέλεσμα του dropout να είναι η μείωση της υπερεκπαίδευσης.

Μια σχετική ευρετική εξήγηση για το dropout δίνεται σε μία από τις πρώτες δημοσιεύσεις που χρησιμοποιούν την τεχνική (Krizhevsky et al, 2012): «Αυτή η τεχνική μειώνει τις σύνθετες συν-προσαρμογές των νευρώνων, καθώς ένας νευρώνας δεν μπορεί να βασιστεί στην παρουσία συγκεκριμένων άλλων νευρώνων. Είναι επομένως αναγκασμένος να μάθει πιο ισχυρά χαρακτηριστικά που είναι χρήσιμα σε συνδυασμό με πολλά διαφορετικά τυχαία υποσύνολα των άλλων νευρώνων». Με άλλα λόγια, αν σκεφτούμε το δίκτυό μας ως ένα μοντέλο που κάνει προβλέψεις, τότε μπορούμε να σκεφτούμε το dropout ως έναν τρόπο να διασφαλίσουμε ότι το μοντέλο είναι ισχυρό στην απώλεια οποιουδήποτε μεμονωμένου αποδεικτικού στοιχείου. Σε αυτό είναι κάπως παρόμοιο με την ομαλοποίηση L1 και L2, οι οποίες τείνουν να μειώνουν τα βάρη και έτσι να κάνουν το δίκτυο πιο ισχυρό στην απώλεια μιας μεμονωμένης σύνδεσης στο δίκτυο.

Φυσικά το πραγματικό μέτρο του dropout είναι ότι έχει υπάρξει πολύ επιτυχές στη βελτίωση της απόδοσης των νευρωνικών δικτύων. Η αρχική εργασία (Hinton et al, 2012) που εισήγαγε την τεχνική την αξιοποίησε σε πολλές διαφορετικές εφαρμογές. Έχει ιδιαίτερο ενδιαφέρον ότι εφαρμόστηκε dropout στην ταξινόμηση ψηφίων MNIST, χρησιμοποιώντας ένα κλασικό νευρωνικό δίκτυο όπως αυτά που εξετάζουμε. Η εργασία επεσήμανε ότι το καλύτερο αποτέλεσμα που είχε επιτύχει κανείς μέχρι εκείνο το σημείο, χρησιμοποιώντας μια τέτοια αρχιτεκτονική, ήταν 98,4% ακρίβεια ταξινόμησης στο σύνολο ελέγχου. Αυτό βελτιώθηκε σε 98,7% χρησιμοποιώντας έναν συνδυασμό dropout με μια τροποποιημένη μορφή ομαλοποίησης L2. Ομοίως εντυπωσιακά αποτελέσματα έχουν επιτευχθεί για πολλές άλλες εφαρμογές, συμπεριλαμβανομένων προβλημάτων στην αναγνώριση εικόνας και ομιλίας και στην επεξεργασία φυσικής γλώσσας. Το dropout είναι ιδιαίτερα χρήσιμο στην εκπαίδευση μεγάλων, βαθιών δικτύων, όπου το πρόβλημα της υπερεκπαίδευσης είναι συχνά έντονο.

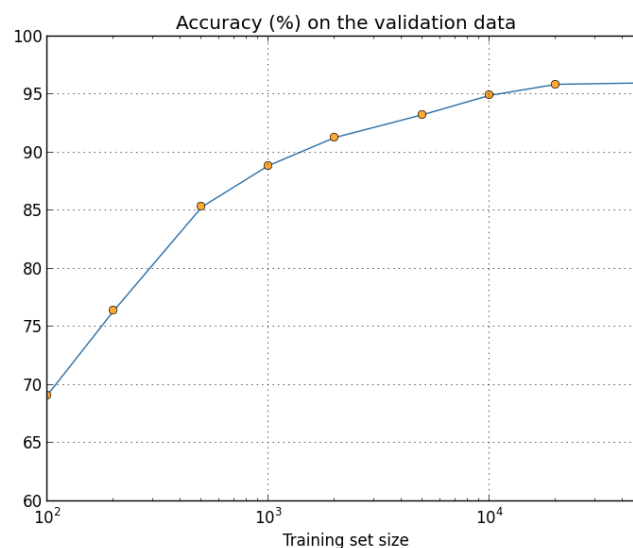
Τεχνητή επαύξηση των δεδομένων εκπαίδευσης: Είδαμε νωρίτερα ότι η ακρίβεια ταξινόμησης MNIST μειώθηκε στην περιοχή του 80% όταν χρησιμοποιήσαμε μόνο 1.000 εικόνες εκπαίδευσης. Δεν προκαλεί έκπληξη το γεγονός ότι αυτό συμβαίνει, καθώς λιγότερα δεδομένα εκπαίδευσης σημαίνουν ότι το δίκτυό μας θα εκτεθεί σε λιγότερες παραλλαγές του τρόπου με τον οποίο οι άνθρωποι γράφουν ψηφία. Ας δοκιμάσουμε να εκπαιδεύσουμε το δίκτυο 30 κρυφών νευρώνων μας με μια ποικιλία διαφορετικών μεγεθών συνόλων εκπαίδευσης για να δούμε πώς ποικίλλει η απόδοση. Εκπαιδεύουμε χρησιμοποιώντας ένα μέγεθος μίνι παρτίδας 10, έναν ρυθμό μάθησης $\eta = 0.5$, μια παράμετρο κανονικοποίησης $\lambda = 5.0$ και τη συνάρτηση κόστους διεντροπίας. Θα εκπαιδεύσουμε για 30 εποχές όταν χρησιμοποιείται το πλήρες σύνολο δεδομένων εκπαίδευσης και θα κλιμακώσουμε τον αριθμό των εποχών αναλογικά όταν χρησιμοποιούνται μικρότερα σύνολα

εκπαίδευσης. Για να διασφαλίσουμε ότι ο συντελεστής φθοράς παραμένει ο ίδιος σε όλα τα σύνολα εκπαίδευσης, θα χρησιμοποιήσουμε μια παράμετρο κανονικοποίησης $\lambda = 5.0$ όταν χρησιμοποιείται το πλήρες σύνολο δεδομένων εκπαίδευσης και θα μειώσουμε αναλογικά το λ όταν χρησιμοποιούνται μικρότερα σύνολα εκπαίδευσης.



Σχήμα 6.11 Ακρίβεια στα δεδομένα επικύρωσης.

Όπως μπορείτε να δείτε, οι ακρίβειες ταξινόμησης βελτιώνονται σημαντικά καθώς χρησιμοποιούμε περισσότερα δεδομένα εκπαίδευσης. Πιθανώς αυτή η βελτίωση θα συνεχιζόταν ακόμη περισσότερο εάν υπήρχαν περισσότερα διαθέσιμα στοιχεία. Φυσικά κοιτάζοντας το παραπάνω γράφημα φαίνεται ότι πλησιάζουμε τον κορεσμό. Ας υποθέσουμε ωστόσο ότι ξαναφτιάξαμε το γράφημα με το μέγεθος συνόλου εκπαίδευσης σχεδιασμένο λογαριθμικά:



Σχήμα 6.12 Ακρίβεια στα δεδομένα επικύρωσης με λογαριθμικό μέγεθος συνόλου εκπαίδευσης.

Φαίνεται σαφώς ότι το γράφημα εξακολουθεί να ανεβαίνει προς το τέλος. Αυτό υποδηλώνει ότι αν χρησιμοποιούσαμε πολύ περισσότερα δεδομένα εκπαίδευσης —ας πούμε εκατομμύρια ή ακόμη και δισεκατομμύρια δείγματα γραφής, αντί για μόλις 50.000— τότε πιθανότατα θα είχαμε σημαντικά καλύτερη απόδοση, ακόμη και από αυτό το πολύ μικρό δίκτυο.

Η απόκτηση περισσότερων δεδομένων εκπαίδευσης είναι μια εξαιρετική ιδέα. Δυστυχώς μπορεί να είναι ακριβή, γι' αυτόν τον λόγο δεν είναι πάντα δυνατή στην πράξη. Ωστόσο, υπάρχει μια άλλη ιδέα που μπορεί να

λειτουργήσει σχεδόν εξίσου καλά και αυτή είναι η τεχνητή επέκταση των δεδομένων εκπαίδευσης. Ας υποθέσουμε, π.χ. ότι παίρνουμε μια εικόνα εκπαίδευσης MNIST ενός πέντε:



Σχήμα 6.13 Χειρόγραφο ψηφίο.

και την περιστρέφουμε κατά ένα μικρό ποσό, ας πούμε 15 μοίρες:



Σχήμα 6.14 Χειρόγραφο ψηφίο με περιστροφή.

Είναι ακόμα αναγνωρίσιμα το ίδιο ψηφίο· όμως σε επίπεδο pixel είναι αρκετά διαφορετικό από οποιαδήποτε εικόνα αυτήν τη στιγμή στα δεδομένα εκπαίδευσης MNIST. Είναι πιθανό η προσθήκη αυτής της εικόνας στα δεδομένα εκπαίδευσης να μπορεί να βοηθήσει το δίκτυό μας να μάθει περισσότερα σχετικά με τον τρόπο ταξινόμησης των ψηφίων. Επιπλέον προφανώς δεν περιοριζόμαστε στο να προσθέσουμε μόνο αυτήν την εικόνα. Μπορούμε να επεκτείνουμε τα δεδομένα εκπαίδευσής μας κάνοντας πολλές μικρές περιστροφές όλων των εικόνων εκπαίδευσης MNIST και στη συνέχεια, χρησιμοποιώντας τα διευρυμένα δεδομένα εκπαίδευσης, να βελτιώσουμε την απόδοση του δικτύου μας.

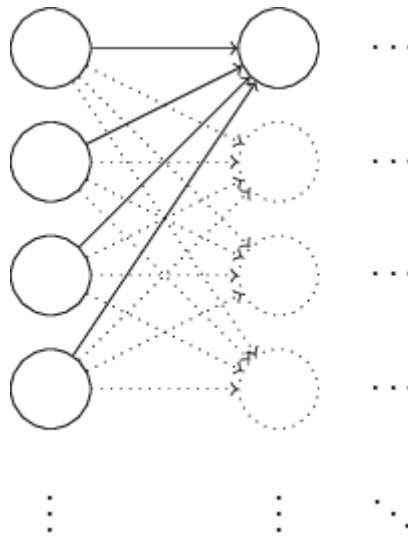
Αυτή η ιδέα είναι πολύ ισχυρή και έχει χρησιμοποιηθεί ευρέως. Στη βιβλιογραφία έχουν εφαρμοστεί διάφορες παραλλαγές της ιδέας στο MNIST. Μία από τις αρχιτεκτονικές του νευρωνικού δικτύου που εξετάστηκε ήταν παρόμοια με αυτήν που χρησιμοποιήσαμε, ένα δίκτυο με 800 κρυμμένους νευρώνες και χρήση της συνάρτησης κόστους διεντροπίας. Η εκπαίδευση του δικτύου με τα τυποποιημένα σύνολα εκπαίδευσης MNIST πέτυχε ακρίβεια ταξινόμησης 98,4% στο σύνολο ελέγχου, αλλά στη συνέχεια επέκτειναν τα δεδομένα εκπαίδευσης, χρησιμοποιώντας όχι μόνο περιστροφές, αλλά και παραλλάσσοντας και παραμορφώνοντας τις εικόνες. Με την εκπαίδευση στο διευρυμένο σύνολο δεδομένων αύξησαν την ακρίβεια του δικτύου τους στο 98,9%. Πειραματίστηκαν επίσης με αυτό που ονόμασαν «ελαστικές παραμορφώσεις», έναν ειδικό τύπο παραμόρφωσης εικόνας που προορίζεται να μιμηθεί τις τυχαίες ταλαντώσεις που βρίσκονται στους μύες των χεριών. Χρησιμοποιώντας τις ελαστικές παραμορφώσεις για να επεκτείνουν τα δεδομένα πέτυχαν ακόμα μεγαλύτερη ακρίβεια, 99,3%. Στην πραγματικότητα διέυρυναν την εμπειρία του δικτύου τους εκθέτοντάς το στο είδος των παραλλαγών που βρίσκονται στον πραγματικό γραφικό χαρακτήρα.

Οι παραλλαγές αυτής της ιδέας μπορούν να χρησιμοποιηθούν για τη βελτίωση της απόδοσης σε πολλές μαθησιακές εργασίες, όχι μόνο στην αναγνώριση χειρογράφου. Η γενική αρχή είναι να επεκταθούν τα δεδομένα εκπαίδευσης εφαρμόζοντας λειτουργίες που αντικατοπτρίζουν τις πραγματικές διακυμάνσεις. Δεν είναι δύσκολο να σκεφτεί κανείς τρόπους να το κάνει αυτό. Ας υποθέσουμε π.χ. ότι χτίζουμε ένα νευρωνικό δίκτυο για να κάνουμε αναγνώριση ομιλίας. Εμείς οι άνθρωποι μπορούμε να αναγνωρίσουμε την ομιλία ακόμα και παρουσία παραμορφώσεων, όπως ο θόρυβος στο παρασκήνιο. Ως αποτέλεσμα μπορείτε να επεκτείνετε τα δεδομένα σας προσθέτοντας θόρυβο παρασκήνιο. Μπορούμε επίσης να αναγνωρίσουμε την ομιλία ακόμη και αν επιταχύνεται ή επιβραδύνεται. Αυτός λοιπόν είναι ένας άλλος τρόπος για να επεκτείνουμε τα δεδομένα εκπαίδευσης. Οι παραπάνω τεχνικές δεν χρησιμοποιούνται πάντα — π.χ. αντί να επεκτείνουν τα δεδομένα κατάρτισης προσθέτοντας θόρυβο, μπορεί να είναι πιο αποτελεσματικό να καθαριστεί η είσοδος στο δίκτυο εφαρμόζοντας πρώτα ένα φίλτρο μείωσης θορύβου. Ωστόσο, αξίζει να έχετε κατά νου την ιδέα της επέκτασης των δεδομένων εκπαίδευσης και να αναζητάτε ευκαιρίες για την εφαρμογή τους.

6.3 Αρχικοποίηση βαρών

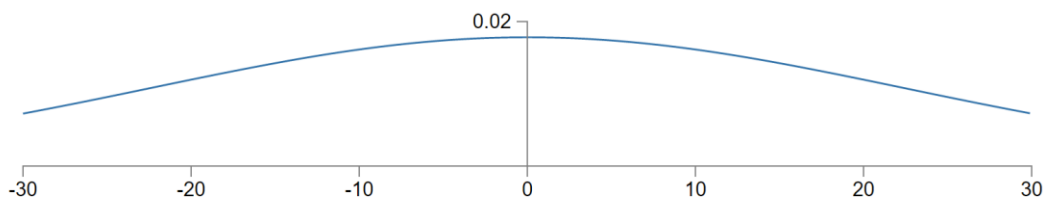
Όταν δημιουργούμε τα νευρωνικά μας δίκτυα, πρέπει να κάνουμε επιλογές για τα αρχικά βάρη και τις πολώσεις. Μέχρι τώρα τα επιλέγαμε σύμφωνα με τις προδιαγραφές που συζητήθηκαν στην ενότητα 4.2. Θυμίζουμε ότι οι οδηγίες ήταν να επιλέξετε τόσο τα βάρη όσο και τις πολώσεις χρησιμοποιώντας ανεξάρτητες γκαουσιανές τυχαίες μεταβλητές, κανονικοποιημένες για να έχουν μέση τιμή 0 και τυπική απόκλιση 1. Ενώ αυτή η προσέγγιση λειτούργησε καλά, ήταν αρκετά ad hoc και αξίζει να την επανεξετάσουμε για να δούμε αν μπορούμε να βρούμε έναν καλύτερο τρόπο να ορίσουμε τα αρχικά βάρη και τις πολώσεις μας, ώστε να βοηθήσουμε τα νευρωνικά μας δίκτυα να μάθουν ίσως γρηγορότερα.

Αποδεικνύεται ότι μπορούμε να κάνουμε κάτι καλύτερο από αυτήν την αρχικοποίηση. Για να δούμε γιατί, ας υποθέσουμε ότι εργαζόμαστε με ένα δίκτυο με μεγάλο αριθμό —ας πούμε 1.000— νευρώνων εισόδου. Και ας υποθέσουμε ότι χρησιμοποιήσαμε κανονικοποιημένες γκαουσιανές μεταβλητές για να αρχικοποιήσουμε τα βάρη που συνδέονται με το πρώτο κρυφό επίπεδο. Προς το παρόν θα επικεντρωθούμε ειδικά στα βάρη που συνδέουν τους νευρώνες εισόδου με τον πρώτο νευρώνα στο κρυφό επίπεδο αγνοώντας το υπόλοιπο δίκτυο:



Σχήμα 6.15 Επίπεδο εισόδου και 1ο κρυφό επίπεδο.

Θα υποθέσουμε για απλότητα ότι προσπαθούμε να εκπαιδεύσουμε χρησιμοποιώντας μια είσοδο εκπαίδευσης x στην οποία οι μισοί νευρώνες εισόδου είναι ενεργοί, δηλαδή έχουν οριστεί σε 1 και οι μισοί νευρώνες εισόδου είναι απενεργοποιημένοι, δηλαδή έχουν οριστεί σε 0. Το επιχείρημα που ακολουθεί ισχύει γενικότερα, αλλά θα κρατήσετε την ουσία από αυτήν την ειδική περίπτωση. Ας εξετάσουμε το σταθμισμένο άθροισμα $z = \sum_j w_j x_j + b$ των εισόδων στον κρυμμένο νευρώνα μας. 500 όροι σε αυτό το άθροισμα εξαφανίζονται, επειδή η αντίστοιχη είσοδος είναι μηδέν. Έτσι το z είναι ένα άθροισμα από 501 κανονικοποιημένες γκαουσιανές τυχαίες μεταβλητές, που αντιπροσωπεύουν τα 500 βάρη και τη μία πόλωση. Ως αποτέλεσμα και το z θα ακολουθεί γκαουσιανή κατανομή με μέση τιμή μηδέν και τυπική απόκλιση $\sqrt{501} \approx 22.4$. δηλαδή το z έχει μια πολύ ευρεία κατανομή, που δεν κορυφώνεται καθόλου απότομα:

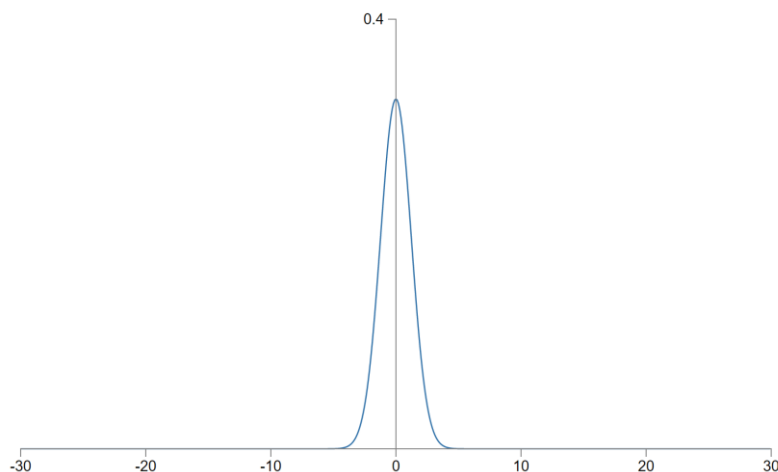


Σχήμα 6.16 Η κατανομή του z .

Ειδικότερα μπορούμε να δούμε από το παραπάνω γράφημα ότι είναι πολύ πιθανό το $|z|$ να είναι αρκετά μεγάλο, δηλαδή είτε $z \gg 1$ είτε $z \ll -1$. Εάν συμβαίνει αυτό, τότε η έξοδος $\sigma(z)$ από τον κρυμμένο νευρώνα θα είναι πολύ κοντά είτε σε 1 ή σε 0. Κάτι τέτοιο σημαίνει ότι ο κρυμμένος νευρώνας μας θα έχει κορεστεί. Και όταν συμβεί αυτό, όπως γνωρίζουμε, μικρές αλλαγές στα βάρη θα έχουν μόνο απολύτως μικροσκοπικές επιπτώσεις στην ενεργοποίηση του κρυμμένου νευρώνα μας. Η συγκεκριμένη μικροσκοπική αλλαγή στην ενεργοποίηση του κρυμμένου νευρώνα, με τη σειρά της, δύσκολα θα επηρεάσει τους υπόλοιπους νευρώνες του δικτύου, οπότε θα δούμε μια αντίστοιχα μικροσκοπική αλλαγή στη συνάρτηση κόστους. Ως αποτέλεσμα αυτά τα βάρη θα μάθουν πολύ αργά όταν χρησιμοποιούμε κατάβαση κλίσης. Είναι παρόμοιο με το πρόβλημα που συζητήσαμε νωρίτερα στο ίδιο κεφάλαιο, στο οποίο οι νευρώνες εξόδου που είναι κορεσμένοι σε λάθος τιμή προκάλεσαν την επιβράδυνση της μάθησης. Το αντιμετωπίσαμε με μια έξυπνη επιλογή συνάρτησης κόστους. Δυστυχώς, ενώ αυτό βοήθησε με τους κορεσμένους νευρώνες εξόδου, δεν προσφέρει τίποτα για το πρόβλημα με τους κορεσμένους κρυμμένους νευρώνες.

Φυσικά παρόμοια επιχειρήματα ισχύουν και για μεταγενέστερα κρυφά επίπεδα: εάν τα βάρη σε μεταγενέστερα κρυφά επίπεδα αρχικοποιούνται χρησιμοποιώντας κανονικοποιημένες γκαουσιανές κατανομές, τότε οι ενεργοποιήσεις συχνά θα είναι πολύ κοντά σε 0 ή 1 και η μάθηση θα προχωρήσει πολύ αργά.

Υπάρχει κάποιος τρόπος να επιλέξουμε καλύτερες αρχικοποιήσεις για τα βάρη και τις πολώσεις, έτσι ώστε να μην έχουμε αυτό το είδος κορεσμού και να αποφύγουμε μια επιβράδυνση μάθησης; Ας υποθέσουμε ότι έχουμε έναν νευρώνα με n_{in} πλήθος βαρών εισόδου. Στη συνέχεια θα αρχικοποιήσουμε αυτά τα βάρη ως τυχαίες μεταβλητές gaussian με μέση τιμή 0 και τυπική απόκλιση $1/\sqrt{n_{in}}$. δηλαδή θα συμπυκνώσουμε τις κατανομές, καθιστώντας λιγότερο πιθανό ότι ο νευρώνας μας θα κορεστεί. Θα συνεχίσουμε να επιλέγουμε την πόλωση ως γκαουσιανή με μέση τιμή 0 και τυπική απόκλιση 1 για λόγους που θα δούμε παρακάτω. Με αυτές τις επιλογές το σταθμισμένο άθροισμα $z = \sum_j w_j x_j + b$ θα είναι και πάλι μια τυχαία μεταβλητή gaussian με μέση τιμή 0, αλλά θα είναι πολύ πιο έντονα κορυφωμένη από ό,τι ήταν πριν. Ας υποθέσουμε, όπως κάναμε νωρίτερα, ότι 500 εισοδοί είναι μηδέν και 500 είναι 1. Στη συνέχεια είναι εύκολο να δείξετε (δείτε την παρακάτω άσκηση) ότι το z έχει μια κατανομή με μέση τιμή 0 και τυπική απόκλιση $\sqrt{3/2} = 1.22 \dots$. Αυτό είναι πολύ πιο έντονα κορυφωμένο από πριν, τόσο πολύ ώστε ακόμη και το παρακάτω γράφημα υποτιμά την κατάσταση, καθώς έπρεπε να επανατοποθετηθεί ο κατακόρυφος άξονας σε σύγκριση με το προηγούμενο γράφημα:



Σχήμα 6.17 Η νέα κατανομή του z .

Ένας τέτοιος νευρώνας είναι πολύ λιγότερο πιθανό να κορεστεί και αντίστοιχα πολύ λιγότερο πιθανό να έχει προβλήματα με μια επιβράδυνση μάθησης.

Αναφέρθηκε παραπάνω ότι θα συνεχίσουμε να αρχικοποιούμε τις πολώσεις όπως πριν, ως τυχαίες μεταβλητές gaussian με μέσο όρο 0 και τυπική απόκλιση 1. Αυτό μπορεί να γίνει, επειδή δεν κάνει πολύ πιο πιθανό οι νευρώνες μας να κορεστούν. Στην πραγματικότητα δεν έχει μεγάλη σημασία πώς θα αρχικοποιήσουμε τις πολώσεις, υπό την προϋπόθεση ότι θα αποφύγουμε το πρόβλημα με τον κορεσμό. Ορισμένοι φτάνουν στο σημείο να αρχικοποιήσουν όλες τις πολώσεις σε 0 και βασίζονται στην κατάβαση κλίσης για την εκμάθηση

κατάλληλων πολώσεων. Δεδομένου ότι είναι απίθανο να κάνει μεγάλη διαφορά, θα συνεχίσουμε με την ίδια διαδικασία αρχικοποίησης όπως και πριν.

Η προσέγγιση $1/\sqrt{n_{in}}$ στην αρχικοποίηση βάρους βοηθά στη βελτίωση του τρόπου με τον οποίο μαθαίνουν τα νευρωνικά μας δίκτυα. Έχουν επίσης προταθεί και άλλες τεχνικές για την αρχικοποίηση βαρών, πολλές από τις οποίες βασίζονται σε αυτήν τη βασική ιδέα (βλ. και Goodfellow et al., 2016).

Άσκηση 6.3.1.

Να δείξετε ότι η τυπική απόκλιση του αθροίσματος $z = \sum_j w_j x_j + b$ είναι $\sqrt{3/2}$, δεδομένου ότι: α) η διακύμανση ενός αθροίσματος ανεξάρτητων τυχαίων μεταβλητών είναι το άθροισμα των διακυμάνσεων των επί μέρους μεταβλητών και β) η διακύμανση είναι το τετράγωνο της απόκλισης.

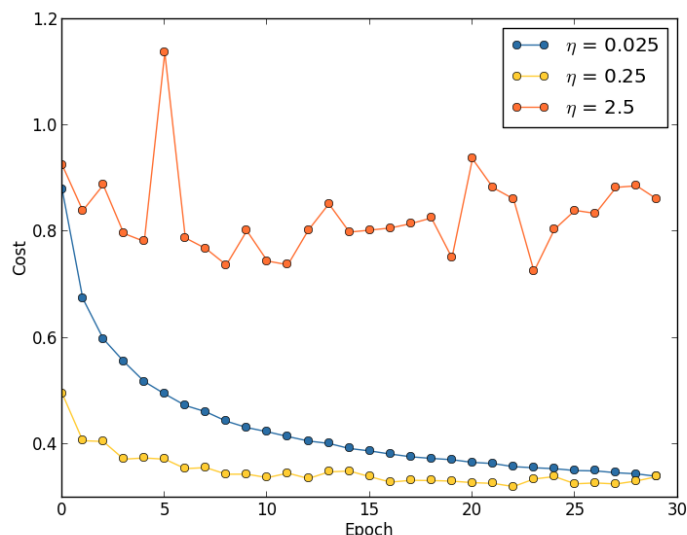
6.4 Επιλογή υπερπαραμέτρων νευρωνικών δικτύων

Σε αυτήν την ενότητα εξηγούνται ορισμένες ευρετικές που μπορούν να χρησιμοποιηθούν για τη ρύθμιση των υπερπαραμέτρων σε ένα νευρωνικό δίκτυο. Ο στόχος είναι να σας βοηθήσουμε να αναπτύξετε μια ροή εργασίας που σας επιτρέπει να κάνετε πολύ καλή δουλειά ρυθμίζοντας υπερπαραμέτρους. Φυσικά δεν μπορούν να καλυφθούν τα πάντα σχετικά με τη βελτιστοποίηση υπερπαραμέτρων. Πρόκειται για ένα τεράστιο θέμα και δεν είναι σε καμία περίπτωση ένα πρόβλημα που έχει επιλυθεί ποτέ πλήρως, ούτε υπάρχει καθολική συμφωνία μεταξύ των ερευνητών σχετικά με τις σωστές στρατηγικές που πρέπει να χρησιμοποιηθούν.

Θα εστιάσουμε στον ρυθμό μάθησης, η , την παράμετρο ομαλοποίησης L2, λ και το μέγεθος mini-batch. Ωστόσο, πολλές από τις παρατηρήσεις ισχύουν και για άλλες υπερπαραμέτρους, συμπεριλαμβανομένων εκείνων που σχετίζονται με την αρχιτεκτονική του δικτύου, καθώς και άλλες μορφές ομαλοποίησης.

6.4.1 Ρυθμός μάθησης

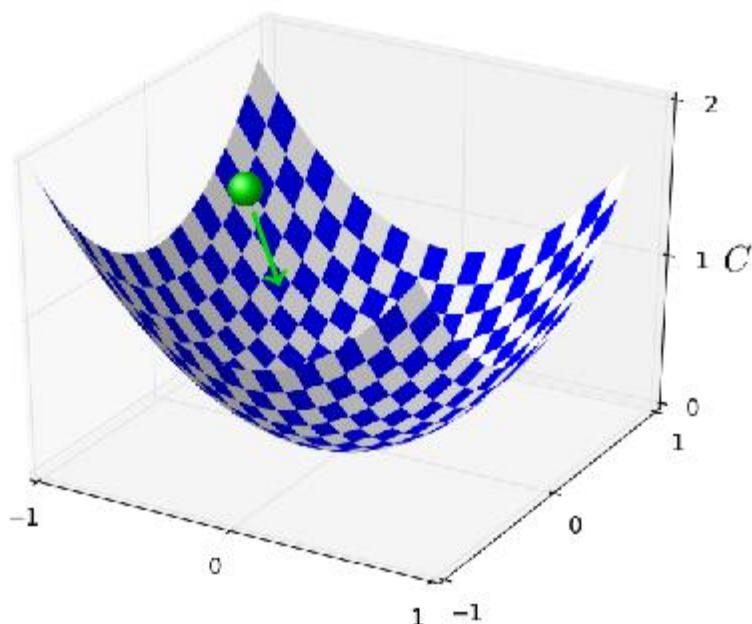
Ας υποθέσουμε ότι τρέχουμε τρία δίκτυα MNIST με τρεις διαφορετικούς ρυθμούς μάθησης, $\eta=0,025$, $\eta=0,25$ και $\eta=2,5$ αντίστοιχα. Θα ορίσουμε τις άλλες υπερπαραμέτρους, όπως στα πειράματα σε προηγούμενες ενότητες, τρέχοντας για 30 εποχές, με μέγεθος mini-batch 10 και με $\lambda=5,0$. Θα επιστρέψουμε επίσης στη χρήση ολόκληρου του συνόλου εκπαίδευσης των 50.000 εικόνων. Ακολουθεί ένα γράφημα που δείχνει τη συμπεριφορά του κόστους εκπαίδευσης καθώς το δίκτυο εκπαιδεύεται:



Σχήμα 6.18 Κόστος για διαφορετικούς ρυθμούς μάθησης.

Με $\eta=0,025$ το κόστος μειώνεται ομαλά μέχρι την τελική εποχή. Με $\eta=0,25$ το κόστος αρχικά μειώνεται, αλλά μετά από περίπου 20 εποχές είναι κοντά στον κορεσμό, ενώ στη συνέχεια οι περισσότερες από τις αλλαγές είναι απλώς μικρές και προφανώς τυχαίες ταλαντώσεις. Τέλος με $\eta=2,5$ το κόστος κάνει μεγάλες ταλαντώσεις

από την αρχή. Για να κατανοήσουμε τον λόγο των ταλαντώσεων, ας θυμηθούμε ότι η στοχαστική κατάβαση κλίσης υποτίθεται ότι μας κατεβάζει σταδιακά σε μια κοιλάδα της συνάρτησης κόστους:



Σχήμα 6.19 Κατάβαση κλίσης και σε κυρτή συνάρτηση κόστους.

Ωστόσο, εάν το η είναι πολύ μεγάλο, τότε τα βήματα θα είναι τόσο μεγάλα που μπορεί στην πραγματικότητα να ξεπεράσουν το ελάχιστο, προκαλώντας τον αλγόριθμο να ανέβει έξω από την κοιλάδα. Αυτός είναι πιθανώς ο λόγος που προκαλεί την ταλάντωση του κόστους όταν $\eta=2,5^2$. Όταν επιλέγουμε $\eta=0,25$, τα αρχικά βήματα μάς οδηγούν προς ένα ελάχιστο της συνάρτησης κόστους και μόνο όταν φτάσουμε κοντά σε αυτό το ελάχιστο αρχίζουμε να υποφέρουμε από το πρόβλημα υπέρβασης. Όταν επιλέγουμε $\eta=0,025$, δεν υποφέρουμε καθόλου από αυτό το πρόβλημα κατά τη διάρκεια των πρώτων 30 εποχών. Βέβαια η επιλογή τόσο μικρού η δημιουργεί ένα άλλο πρόβλημα, δηλαδή την επιβράδυνση της στοχαστικής κατάβασης κλίσης. Μια ακόμη καλύτερη προσέγγιση θα ήταν να ξεκινήσουμε να εκπαιδεύσουμε για 20 εποχές με $\eta=0,25$ και στη συνέχεια να μεταβούμε σε $\eta=0,025$. Θα συζητήσουμε αργότερα τέτοια προγράμματα μεταβλητού ρυθμού μάθησης. Προς το παρόν όμως ας επικεντρωθούμε στο πώς να βρούμε μια σταθερή καλή τιμή για τον ρυθμό μάθησης, η .

Πρώτον εκτιμούμε την οριακή τιμή του η για την οποία το κόστος στα δεδομένα εκπαίδευσης αρχίζει αμέσως να μειώνεται, αντί να ταλαντεύεται ή να αυξάνεται. Αυτή η εκτίμηση δεν χρειάζεται να είναι πολύ ακριβής. Μπορείτε να υπολογίσετε την τάξη μεγέθους ξεκινώντας με $\eta=0,01$. Εάν το κόστος μειωθεί κατά τη διάρκεια των πρώτων λίγων εποχών, τότε θα πρέπει να προσπαθήσετε διαδοχικά $\eta=0,1, 1,0, \dots$ μέχρι να βρείτε μια τιμή όπου το κόστος ταλαντεύεται ή αυξάνεται κατά τη διάρκεια των πρώτων λίγων εποχών. Εναλλακτικά, εάν το κόστος ταλαντεύεται ή αυξάνεται κατά τη διάρκεια των πρώτων λίγων εποχών όταν $\eta=0,01$, τότε δοκιμάστε $\eta=0,001, 0,0001, \dots$ μέχρι να βρείτε μια τιμή όπου το κόστος μειώνεται κατά τη διάρκεια των πρώτων εποχών. Ακολουθώντας αυτήν τη διαδικασία θα έχουμε μια εκτίμηση τάξης μεγέθους για την οριακή τιμή του η . Μπορείτε προαιρετικά να βελτιώσετε την εκτίμησή σας, για να επιλέξετε τη μεγαλύτερη τιμή η στην οποία

² Μια πιο ολοκληρωμένη εξήγηση είναι εν συντομία η εξής: η κατάβαση κλίσης χρησιμοποιεί μια προσέγγιση πρώτης τάξης στη συνάρτηση κόστους ως οδηγό για το πώς να μειωθεί το κόστος. Για τα μεγάλα η , οι όροι υψηλότερης τάξης στη συνάρτηση κόστους γίνονται πιο σημαντικοί και μπορεί να κυριαρχήσουν στη συμπεριφορά, προκαλώντας την κατάρρευση της κατάβασης κλίσης. Αυτό είναι ιδιαίτερα πιθανό καθώς πλησιάζουμε τα ελάχιστα και τα σχεδόν ελάχιστα της συνάρτησης κόστους, καθώς κοντά σε τέτοια σημεία η κλίση γίνεται μικρή, καθιστώντας ευκολότερο για τους όρους υψηλότερης τάξης να κυριαρχήσουν στη συμπεριφορά.

μειώνεται το κόστος κατά τη διάρκεια των πρώτων λίγων εποχών, ας πούμε $\eta=0,5$ ή $\eta=0,2$ (δεν χρειάζεται αυτό να είναι εξαιρετικά ακριβές). Αυτό μας δίνει μια εκτίμηση για την οριακή τιμή του η .

Προφανώς η πραγματική τιμή του η που χρησιμοποιείτε δεν πρέπει να είναι μεγαλύτερη από την τιμή κατωφλίου. Στην πραγματικότητα, αν η τιμή του η πρόκειται να παραμείνει σε χρήση για πολλές εποχές, τότε πιθανότατα να χρειάζεται μια τιμή του η που είναι μικρότερη, ας πούμε ένας παράγοντας 2 κάτω από το όριο. Μια τέτοια επιλογή θα επιτρέψει συνήθως την εκπαίδευση για πολλές εποχές, χωρίς να προκληθεί υπερβολική επιβράδυνση στη μάθηση.

Στην περίπτωση των δεδομένων MNIST, ακολουθώντας αυτήν τη στρατηγική, οδηγούμαστε σε εκτίμηση 0,1 για την τάξη μεγέθους της οριακής τιμής η . Με περαιτέρω εκτέλεση λαμβάνουμε μια τιμή κατωφλίου $\eta=0,5$. Ακολουθώντας την παραπάνω συνταγή υποδηλώνεται η χρήση $\eta=0,25$ ως τιμή για τον ρυθμό μάθησης. Στην πράξη η χρήση $\eta=0,5$ λειτούργησε αρκετά καλά για 30 εποχές χωρίς επιβράδυνση στη μάθηση.

Όλα αυτά φαίνονται αρκετά απλά. Ωστόσο, η χρήση του κόστους εκπαίδευσης για την επιλογή του η φαίνεται να έρχεται σε αντίθεση με αυτό που έχει ειπωθεί νωρίτερα: δηλαδή ότι θα επιλέγαμε υπερπαραμέτρους αξιολογώντας την απόδοση στα δεδομένα επικύρωσης που έχουμε κρατήσει. Στην πραγματικότητα θα χρησιμοποιήσουμε την ακρίβεια επικύρωσης για να επιλέξουμε την υπερπαραμέτρο ομαλοποίησης, το μέγεθος mini-batch και παραμέτρους δικτύου, όπως ο αριθμός των επιπέδων και των κρυφών νευρώνων κοκ. Γιατί να ενεργήσουμε διαφορετικά για τον ρυθμό μάθησης; Το σκεπτικό είναι ότι οι άλλες υπερπαραμέτροι προορίζονται να βελτιώσουν την ακρίβεια της τελικής ταξινόμησης στο σύνολο ελέγχου και έτσι είναι λογικό να τις επιλέξουμε με βάση την ακρίβεια επικύρωσης. Ωστόσο, ο ρυθμός μάθησης προορίζεται μόνο να επηρεάσει παρεμπιπτόντως την ακρίβεια της τελικής ταξινόμησης. Ο πρωταρχικός σκοπός του είναι πραγματικά να ελέγξει το μέγεθος του βήματος στην κατάβαση κλίσης και η παρακολούθηση του κόστους εκπαίδευσης είναι ο καλύτερος τρόπος προκειμένου να ανιχνεύσετε εάν το μέγεθος του βήματος είναι πολύ μεγάλο. Εντούτοις, αυτό αποτελεί μια υποκειμενική προτίμηση. Στην αρχή της εκπαίδευσης το κόστος συνήθως μόνο μειώνεται, ενόσω η ακρίβεια επικύρωσης βελτιώνεται, επομένως στην πράξη είναι απίθανο να κάνει μεγάλη διαφορά ποιο κριτήριο χρησιμοποιείτε.

6.4.2 Χρήση πρόωρου σταματήματος για προσδιορισμό του αριθμού των εποχών

Όπως συζητήσαμε νωρίτερα στο κεφάλαιο, το πρόωρο σταμάτημα σημαίνει ότι στο τέλος κάθε εποχής θα πρέπει να υπολογίσουμε την ακρίβεια ταξινόμησης στα δεδομένα επικύρωσης. Όταν αυτή σταματήσει να βελτιώνεται, η εκπαίδευση τερματίζεται. Επομένως, ο καθορισμός του αριθμού των εποχών καθίσταται πολύ απλός. Συγκεκριμένα δεν χρειάζεται να ανησυχούμε για να καταλάβουμε ρητά πώς ο αριθμός των εποχών εξαρτάται από τις άλλες υπερπαραμέτρους, αφού αυτό φροντίζεται αυτόματα. Επιπλέον, το πρόωρο σταμάτημα προλαμβάνει επίσης αυτόματα την υπερεκπαίδευση. Αυτό είναι φυσικά καλό, αν και στα πρώτα στάδια του πειραματισμού μπορεί να είναι χρήσιμο να απενεργοποιήσετε το πρόωρο σταμάτημα, ώστε να μπορείτε να δείτε τυχόν σημάδια υπερεκπαίδευσης και να τα χρησιμοποιήσετε για να ενημερώσετε την προσέγγισή σας για την ομαλοποίηση.

Για να εφαρμόσουμε το πρόωρο σταμάτημα, πρέπει να πούμε με μεγαλύτερη ακρίβεια τι σημαίνει ότι η ακρίβεια της ταξινόμησης έχει σταματήσει να βελτιώνεται. Όπως έχουμε δει, η ακρίβεια μπορεί να ταλαντώνεται αρκετά, ακόμη και όταν η συνολική τάση είναι βελτιωτική. Εάν σταματήσουμε την πρώτη φορά που μειώνεται η ακρίβεια, τότε σχεδόν σίγουρα θα έχουμε σταματήσει ενώ υπάρχουν περισσότερες βελτιώσεις που θα μπορούσαν να γίνουν. Ένας καλύτερος κανόνας είναι ο τερματισμός όταν η καλύτερη ακρίβεια ταξινόμησης δεν βελτιώνεται για αρκετό καιρό. Ας υποθέσουμε π.χ. ότι κάνουμε MNIST. Στη συνέχεια θα μπορούσαμε να επιλέξουμε να τερματίσουμε εάν η ακρίβεια ταξινόμησης δεν έχει βελτιωθεί κατά τη διάρκεια των τελευταίων δέκα εποχών. Αυτό διασφαλίζει ότι δεν θα σταματήσουμε πολύ νωρίς, αντιδρώντας σε μια ατυχία κατά την εκπαίδευση, αλλά και ότι δεν θα περιμένουμε για πάντα για μια βελτίωση που δεν έρχεται ποτέ.

Ο συγκεκριμένος κανόνας μη βελτίωσης σε δέκα εποχές είναι καλός για την αρχική εξερεύνηση του MNIST. Ωστόσο, τα δίκτυα μπορούν μερικές φορές να βρεθούν σε επίπεδες περιοχές κοντά σε μια συγκεκριμένη ακρίβεια ταξινόμησης για αρκετό καιρό, μόνο για να αρχίσουν στη συνέχεια να βελτιώνονται και πάλι. Εάν προσπαθείτε να έχετε πραγματικά καλή απόδοση, ο κανόνας χωρίς βελτίωση σε δέκα εποχές μπορεί να είναι πολύ αυστηρός. Σε αυτήν την περίπτωση προτείνεται να χρησιμοποιήσετε τον κανόνα της μη βελτίωσης σε δέκα για τον αρχικό πειραματισμό και να υιοθετήσετε σταδιακά πιο επιεικείς κανόνες, καθώς κατανοείτε καλύτερα τον τρόπο με τον οποίο το δίκτυό σας εκπαιδεύεται: καμία βελτίωση σε είκοσι, καμία βελτίωση σε πενήντα και ούτω καθεξής. Φυσικά αυτό εισάγει μια νέα υπερπαραμέτρο για βελτιστοποίηση!

Στην πράξη ωστόσο είναι συνήθως εύκολο να την ορίσετε για να έχετε αρκετά καλά αποτελέσματα. Ομοίως, για άλλα προβλήματα εκτός από το MNIST, ο κανόνας της μη βελτίωσης σε δέκα μπορεί να είναι υπερβολικά αυστηρός ή όχι αρκετά αυστηρός, ανάλογα με τις λεπτομέρειες του προβλήματος. Όμως με λίγο πειραματισμό είναι συνήθως εύκολο να βρείτε μια αρκετά καλή στρατηγική για πρόωρο σταμάτημα.

6.4.3 Πρόγραμμα ρυθμού μάθησης

Κρατάμε σταθερό τον ρυθμό μάθησης η . Ωστόσο, είναι συχνά επωφελές αυτός να μεταβάλεται. Νωρίς κατά τη διάρκεια της μαθησιακής διαδικασίας είναι πιθανό τα βάρη να είναι αρκετά λάθος. Έτσι είναι καλύτερο να χρησιμοποιηθεί ένας μεγάλος ρυθμός μάθησης που προκαλεί γρήγορη αλλαγή τους. Αργότερα μπορούμε να μειώσουμε τον ρυθμό μάθησης, καθώς κάνουμε πιο λεπτομερείς προσαρμογές στα βάρη μας.

Πώς πρέπει να ορίσουμε το πρόγραμμα ρυθμού μάθησης; Πολλές προσεγγίσεις είναι εφικτές. Μια φυσική προσέγγιση είναι να χρησιμοποιήσετε την ίδια βασική ιδέα με πρόωρο σταμάτημα. Η ιδέα είναι να διατηρηθεί σταθερός ο ρυθμός μάθησης έως ότου η ακρίβεια επικύρωσης αρχίσει να επιδεινώνεται. Στη συνέχεια μειώστε τον ρυθμό μάθησης κατά κάποιο ποσό, ας πούμε έναν παράγοντα δύο ή δέκα. Το επαναλαμβάνουμε πολλές φορές, έως ότου το ποσοστό μάθησης είναι —ενδεικτικά— ένας συντελεστής 1.024 (ή 1.000) φορές χαμηλότερος από την αρχική τιμή. Μετά τερματίζουμε.

Ένα μεταβλητό πρόγραμμα μάθησης μπορεί να βελτιώσει την απόδοση, αλλά ανοίγει επίσης έναν κόσμο πιθανών επιλογών για το πρόγραμμα μάθησης. Αυτές οι επιλογές μπορεί να είναι πονοκέφαλος — μπορείτε να περάσει πολύς χρόνος προσπαθώντας να βελτιστοποιήσετε το πρόγραμμα μάθησής σας. Για τα πρώτα πειράματα η πρόταση είναι να χρησιμοποιήσετε μια ενιαία, σταθερή τιμή για τον ρυθμό μάθησης. Κάτι τέτοιο θα σας δώσει μια καλή πρώτη προσέγγιση. Αργότερα, αν θέλετε να αποκτήσετε την καλύτερη απόδοση από το δίκτυό σας, αξίζει να πειραματιστείτε με ένα πρόγραμμα μάθησης σύμφωνα με αυτά που αναφέρθηκαν.

6.4.4 Η παράμετρος ομαλοποίησης, λ

Προτείνεται να ξεκινήσετε αρχικά χωρίς ομαλοποίηση ($\lambda=0,0$) και να καθορίσετε μια τιμή για το η , όπως παραπάνω. Χρησιμοποιώντας αυτήν την επιλογή η , μπορούμε στη συνέχεια να χρησιμοποιήσουμε τα δεδομένα επικύρωσης για να επιλέξουμε μια καλή τιμή για το λ . Ξεκινήστε δοκιμάζοντας $\lambda=1.0$ και στη συνέχεια να αυξηθεί ή να μειωθεί κατά παράγοντες του 10, όπως απαιτείται για τη βελτίωση της απόδοσης στα δεδομένα επικύρωσης. Μόλις βρείτε μια καλή τάξη μεγέθους, μπορείτε να ρυθμίσετε με ακρίβεια την τιμή του λ . Κατόπιν θα πρέπει να επιστρέψετε και να βελτιστοποιήσετε ξανά την η .

6.4.5 Μέγεθος Mini-batch

Πώς πρέπει να ορίσουμε το μέγεθος mini-batch; Για να απαντήσουμε σε αυτήν την ερώτηση, ας υποθέσουμε πρώτα ότι κάνουμε online μάθηση, δηλαδή ότι χρησιμοποιούμε ένα μέγεθος μίνι παρτίδας 1.

Η προφανής ανησυχία με την online μάθηση είναι ότι η χρήση mini-batch που περιέχουν μόνο ένα παράδειγμα εκπαίδευσης θα προκαλέσει σημαντικά σφάλματα στην εκτίμησή μας για την κλίση. Στην πραγματικότητα όμως αποδεικνύεται ότι τα λάθη δεν αποτελούν τέτοιο πρόβλημα. Ο λόγος είναι ότι οι μεμονωμένες εκτιμήσεις κλίσης δεν χρειάζεται να είναι εξαιρετικά ακριβείς. Το μόνο που χρειαζόμαστε είναι μια εκτίμηση αρκετά ακριβή, ώστε η συνάρτηση κόστους μας να τείνει να μειώνεται συνεχώς. Είναι σαν να προσπαθείτε να φτάσετε στον Βόρειο Μαγνητικό Πόλο με μια χαλασμένη πυξίδα που είναι 10-20 μοίρες λάθος κάθε φορά που την κοιτάζετε. Υπό την προϋπόθεση ότι σταματάτε για να την ελέγχετε συχνά και η πυξίδα παίρνει κατά μέσο όρο τη σωστή κατεύθυνση, θα καταλήξετε στον Βόρειο Μαγνητικό Πόλο μια χαρά.

Με βάση αυτό το επιχείρημα ακούγεται σαν να πρέπει να χρησιμοποιήσουμε online μάθηση. Στην πραγματικότητα η κατάσταση αποδεικνύεται πιο περίπλοκη. Έχει επισημανθεί ότι είναι δυνατό να χρησιμοποιηθούν τεχνικές πινάκων για τον υπολογισμό της ενημέρωσης κλίσης για όλα τα παραδείγματα σε ένα mini-batch ταυτόχρονα, αντί να γίνεται επανάληψη για ένα-ένα. Ανάλογα με τις λεπτομέρειες του υλικού σας και της βιβλιοθήκης γραμμικής άλγεβρας, αυτό μπορεί να κάνει αρκετά πιο γρήγορο τον υπολογισμό της εκτίμησης κλίσης για ένα mini-batch μεγέθους π.χ. 100, αντί να υπολογίσετε την εκτίμηση κλίσης mini-batch κάνοντας βρόχο πάνω από τα παραδείγματα εκπαίδευσης 100 φορές ξεχωριστά. Μπορεί να χρειαστεί π.χ. ο χρόνος που αντιστοιχεί μόνο σε 50 φορές, αντί σε 100.

Αρχικά όμως φαίνεται ότι κάτι τέτοιο δεν μας βοηθά τόσο πολύ. Με mini-batch μεγέθους 100 ο κανόνας μάθησης για τα βάρη μοιάζει με:

$$w \rightarrow w' = w - \eta \frac{1}{100} \sum_x \nabla C_x, \quad (6.30)$$

όπου το άθροισμα αφορά όλα τα δείγματα του mini-batch. Αντίθετα για online μάθηση ισχύει:

$$w \rightarrow w' = w - \eta \nabla C_x \quad (6.31)$$

Ακόμα κι αν χρειάζονται μόνο 50 φορές για να γίνει η ενημέρωση mini-batch, φαίνεται να είναι καλύτερο να κάνουμε online μάθηση, επειδή θα ενημερώνουμε πολύ πιο συχνά. Ας υποθέσουμε ωστόσο ότι στην περίπτωση mini-batch αυξάνουμε τον ρυθμό μάθησης κατά έναν παράγοντα 100, οπότε ο κανόνας ενημέρωσης γίνεται:

$$w \rightarrow w' = w - \eta \sum_x \nabla C_x. \quad (6.32)$$

Το αποτέλεσμα μοιάζει πολύ με το να γίνονται 100 ξεχωριστές επαναλήψεις online μάθησης με ρυθμό μάθησης η , αλλά χρειάζεται χρόνος που αντιστοιχεί μόνο σε 50 φορές. Φυσικά δεν είναι πραγματικά το ίδιο με 100 επαναλήψεις online μάθησης, καθώς στο mini-batch τα ∇C_x υπολογίζονται όλα για το ίδιο σύνολο βαρών, σε αντίθεση με τη σωρευτική μάθηση που συμβαίνει στην online περίπτωση. Ωστόσο, φαίνεται σαφώς πιθανό η χρήση του μεγαλύτερου mini-batch να επιταχύνει τα πράγματα.

Έχοντας αυτούς τους παράγοντες κατά νου, η επιλογή του καλύτερου μεγέθους mini-batch είναι ένας συμβιβασμός: πολύ μικρό σημαίνει ότι δεν μπορείτε να επωφεληθείτε πλήρως από τα οφέλη των καλών βιβλιοθηκών πινάκων που έχουν βελτιστοποιηθεί για γρήγορο υλικό· πολύ μεγάλο σημαίνει ότι απλώς δεν ενημερώνετε τα βάρη σας αρκετά συχνά. Αυτό που χρειάζεστε είναι να επιλέξετε μια συμβιβαστική τιμή που μεγιστοποιεί την ταχύτητα της μάθησης. Ευτυχώς η επιλογή του μεγέθους mini-batch με την οποία μεγιστοποιείται η ταχύτητα είναι σχετικά ανεξάρτητη από τις άλλες υπερπαραμέτρους (εκτός από τη συνολική αρχιτεκτονική), οπότε δεν χρειάζεται να τις έχετε βελτιστοποιήσει για να βρείτε ένα καλό μέγεθος mini-batch. Ο τρόπος να προχωρήσουμε είναι επομένως να χρησιμοποιήσουμε κάποιες αποδεκτές (αλλά όχι απαραίτητα τις βέλτιστες) τιμές για τις άλλες υπερπαραμέτρους και στη συνέχεια να δοκιμάσουμε έναν αριθμό διαφορετικών μεγεθών mini-batch, κλικακώνοντας το η όπως παραπάνω. Σχεδιάστε την ακρίβεια επικύρωσης σε σχέση με τον χρόνο (πραγματικός χρόνος που έχει παρέλθει, όχι εποχή!) και επιλέξτε όποιο μέγεθος mini-batch σας δίνει την πιο γρήγορη βελτίωση της απόδοσης. Με το επιλεγμένο μέγεθος mini-batch μπορείτε στη συνέχεια να προχωρήσετε στη βελτιστοποίηση των άλλων υπερπαραμέτρων.

Οι παραπάνω ευρετικές έχουν περιγραφεί σαν να βελτιστοποιείτε τις υπερπαραμέτρους σας με το χέρι. Αυτή η προσέγγιση είναι ένας καλός τρόπος για να δημιουργήσετε μια αίσθηση για το πώς συμπεριφέρονται τα νευρωνικά δίκτυα. Ωστόσο, και όπως ήταν αναμενόμενο, έχει γίνει πολλή δουλειά για την αυτοματοποίηση της διαδικασίας. Μια κοινή τεχνική είναι η *αναζήτηση πλέγματος* (grid-search), η οποία αναζητά συστηματικά μέσω ενός πλέγματος σε χώρο υπερπαραμέτρων. Μια ανασκόπηση τόσο των επιτευγμάτων, όσο και των περιορισμών της αναζήτησης πλέγματος (με προτάσεις για εύκολα υλοποιημένες εναλλακτικές λύσεις) μπορεί να βρεθεί στο Bergstra & Bengio, 2012. Έχουν επίσης προταθεί πολλές πιο εξελιγμένες προσεγγίσεις, όπως βελτιστοποίηση Bayes, γενετικοί αλγόριθμοι και άλλες μεθευρετικές τεχνικές εμπνευσμένες από τη φύση.

Ακολουθώντας τους κανόνες αυτούς δεν θα πετύχετε τα απολύτως καλύτερα δυνατά αποτελέσματα από το νευρωνικό σας δίκτυο, αλλά πιθανότατα θα δοθεί μια καλή αρχή και μια βάση για περαιτέρω βελτιώσεις. Υπάρχουν επίσης συσχετίσεις μεταξύ των υπερπαραμέτρων. Μπορείς να πειραματιστείς με το η , να νιώσεις ότι το έχεις κάνει σωστά και μετά να αρχίσεις να βελτιστοποιείς το λ , μόνο και μόνο για να διαπιστώσεις ότι επηρεάζει τη βελτιστοποίησή σου για το η . Στην πράξη βοηθά η μετάβαση μπρος-πίσω, προσεγγίζοντας σταδιακά καλές τιμές. Πάνω από όλα λάβετε υπόψη ότι οι ευρετικές αυτές είναι κυρίως εμπειρικοί κανόνες. Επομένως, θα πρέπει κανείς να είναι προσεκτικός για σημάδια ότι τα πράγματα δεν λειτουργούν και να είναι πρόθυμος να πειραματιστεί.

Η πρόκληση της ρύθμισης υπερπαραμέτρων έχει οδηγήσει μερικούς να παραπονεθούν ότι τα νευρωνικά δίκτυα απαιτούν πολλή δουλειά σε σύγκριση με άλλες τεχνικές μηχανικής μάθησης. Για παράδειγμα διατυπώνεται το ακόλουθο παράπονο: «Ναι, ένα καλά συντονισμένο νευρωνικό δίκτυο μπορεί να έχει την

καλύτερη απόδοση στο πρόβλημα. Από την άλλη μπορώ να δοκιμάσω ένα random forest ή SVM ή ... [εισάγετε τη δική σας αγαπημένη τεχνική] και απλά λειτουργεί. Δεν έχω χρόνο να προσδιορίσω το σωστό νευρωνικό δίκτυο». Φυσικά, από πρακτική άποψη, είναι καλό να έχουμε εύκολες στην εφαρμογή τεχνικές. Αυτό ισχύει ιδιαίτερα όταν μόλις ξεκινάτε με ένα πρόβλημα και μπορεί να μην είναι προφανές εάν η μηχανική μάθηση μπορεί να βοηθήσει στην επίλυσή του. Από την άλλη, εάν η επίτευξη βέλτιστης απόδοσης είναι σημαντική, τότε ίσως χρειαστεί να δοκιμάσετε προσεγγίσεις που απαιτούν πιο εξειδικευμένες γνώσεις. Ενώ θα ήταν ωραίο η μηχανική μάθηση να ήταν πάντα εύκολη, δεν υπάρχει *a priori* λόγος που θα πρέπει να είναι τετριμμένα απλή.

Βιβλιογραφία

- Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2).
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R.R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.
- Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097-1105.
- Nielsen, M.A. (2019). *Neural networks and deep learning*. Determination press. <http://neuralnetworksanddeeplearning.com>

Απαντήσεις ασκήσεων

Άσκηση 6.1.1. Διεντροπία για $0 < y < 1$.

Να δείξετε ότι η διεντροπία είναι επίσης καλή συνάρτηση κόστους όταν $0 < y < 1$.

Απάντηση

Χρειάζεται να δείξουμε ότι η συνάρτηση κόστους διεντροπίας $C(a) = -(y \ln a + (1 - y) \ln(1 - a))$ ελαχιστοποιείται όταν $a=y$.

Διαφορίζοντας τη C : $C'(a) = -\frac{y}{a} + \frac{1-y}{1-a}$

Ψάχνουμε τοπικό ακρότατο λύνοντας $C'(a)=0$:

$$\begin{aligned} -\frac{y}{a} + \frac{1-y}{1-a} = 0 & \Leftrightarrow \frac{y}{a} = \frac{1-y}{1-a} \\ & \Leftrightarrow y - ay = a - ay \\ & \Leftrightarrow a = y \end{aligned}$$

Έχουμε ολικό ακρότατο στο $a=y$. Για να καθορίσουμε αν είναι μέγιστο ή ελάχιστο, υπολογίζουμε τη 2^η παράγωγο: $C''(a) = \frac{y}{a^2} + \frac{1-y}{(1-a)^2}$

Αφού $0 < y < 1$, και ως συνήθως $0 < a < 1$, έχουμε για όλα τα $0 < a < 1$: $C''(a) \geq 0$ (η συνάρτηση είναι κυρτή). Επομένως, η συνάρτηση ελαχιστοποιείται όταν $a=y$.

Άσκηση 6.1.2. Δίκτυα πολλών νευρώνων πολλαπλών επιπέδων

Να δείξετε ότι για το τετραγωνικό κόστος η μερική παράγωγος ως προς τα βάρη στο επίπεδο εξόδου είναι $\frac{\partial C}{\partial w_{jk}^L} = \frac{1}{n} \sum_x a_k^{L-1} (a_j^L - y_j) \sigma'(z_j^L)$. Να δείξετε ότι με συνάρτηση κόστους διεντροπίας η μερική παράγωγος ως προς τα βάρη γίνεται $\frac{\partial C}{\partial w_{jk}^L} = \frac{1}{n} \sum_x a_k^{L-1} (a_j^L - y_j)$ και ο όρος $\sigma'(z_j^L)$ έχει εξαφανιστεί, όπως ακριβώς και για την περίπτωση του απλού νευρώνα.

Απάντηση

Για ένα δείγμα εκπαίδευσης x έχουμε για το τετραγωνικό κόστος:

$$\begin{aligned} \frac{\partial C}{\partial w_{jk}^L} &= a_k^{L-1} \delta_j^L \\ &= a_k^{L-1} \frac{\partial C}{\partial a_j^L} \sigma'(z_j^L) \end{aligned}$$

Η συνάρτηση κόστους για ένα δείγμα εκπαίδευσης είναι $C = \frac{1}{2} \sum_i (a_i^L - y_i)^2$, επομένως έχουμε $\frac{\partial C}{\partial a_j^L} = a_j^L - y_j$. Αυτό μας δίνει: $\frac{\partial C}{\partial w_{jk}^L} = a_k^{L-1} (a_j^L - y_j) \sigma'(z_j^L)$

Λαμβάνοντας υπόψη όλα τα δείγματα: $\frac{\partial C}{\partial w_{jk}^L} = \frac{1}{n} \sum_x a_k^{L-1} (a_j^L - y_j) \sigma'(z_j^L)$

Για τη διεντροπία υπολογίζουμε πρώτα το δ^L για ένα δείγμα εκπαίδευσης x : Για όλους τους νευρώνες j στο επίπεδο L

$$\begin{aligned}
\delta_j^L &= \frac{\partial C}{\partial a_j^L} \sigma'(z_j^L) \\
&= -\left(\frac{y_j}{a_j^L} - \frac{1-y_j}{1-a_j^L}\right) \sigma'(z_j^L) \\
&= -\left(\frac{y_j}{\sigma(z_j^L)} - \frac{1-y_j}{1-\sigma(z_j^L)}\right) \sigma(z_j^L)(1-\sigma(z_j^L)) \\
&= -(y_j(1-\sigma(z_j^L)) - (1-y_j)\sigma(z_j^L)) \\
&= \sigma(z_j^L) - y_j \\
&= a_j^L - y_j
\end{aligned}$$

Άρα $\delta^L = a^L - y$. Επομένως:

$$\begin{aligned}
\frac{\partial C}{\partial w_{jk}^L} &= a_k^{L-1} \delta_j^L \\
&= a_k^{L-1} (a_j^L - y_j)
\end{aligned}$$

Λαμβάνοντας υπόψη όλα τα δείγματα εκπαίδευσης:

$$\frac{\partial C}{\partial w_{jk}^L} = \frac{1}{n} \sum_x a_k^{L-1} (a_j^L - y_j)$$

Άσκηση 6.1.3. Χρήση τετραγωνικής συνάρτησης κόστους όταν έχουμε γραμμικούς νευρώνες στην έξοδο

Όταν οι νευρώνες εξόδου σε ένα πολυεπίπεδο ΤΝΔ είναι γραμμικοί, δηλαδή δεν εφαρμόζεται η σιγμοειδής συνάρτηση ενεργοποίησης, να δείξετε ότι η μερική παράγωγος της τετραγωνικής συνάρτησης κόστους ως προς τα βάρη και τις πολώσεις του επιπέδου εξόδου είναι

$$\begin{aligned}
\frac{\partial C}{\partial w_{jk}^L} &= \frac{1}{n} \sum_x a_k^{L-1} (a_j^L - y_j) \\
\frac{\partial C}{\partial b_j^L} &= \frac{1}{n} \sum_x (a_j^L - y_j).
\end{aligned}$$

Απάντηση

Χρησιμοποιούμε τετραγωνικό κόστος και τη συνάρτηση ενεργοποίησης $f(x) = x$ στο επίπεδο εξόδου. Έχουμε για ένα δείγμα εκπαίδευσης x και για όλους τους νευρώνες j στο επίπεδο L :

$$\delta_j^L = \frac{\partial C}{\partial a_j^L} f'(z_j^L)$$

Το κόστος για ένα δείγμα εκπαίδευσης είναι $C = \frac{1}{2} \sum_i (a_i^L - y_i)^2$, άρα $\frac{\partial C}{\partial a_j^L} = a_j^L - y_j$ και $\forall x \in \mathbb{R}, f'(x)=1$,
 οπότε $\frac{\partial C}{\partial a_j^L} = a_j^L - y_j$ και σε μορφή διανύσματος $\delta^L = a^L - y$.

Επομένως για ένα δείγμα εκπαίδευσης:

$$\begin{aligned}
\frac{\partial C}{\partial w_{jk}^L} &= a_k^{L-1} \delta_j^L \\
&= a_k^{L-1} (a_j^L - y_j)
\end{aligned}$$

Για τις πολώσεις:

$$\begin{aligned}\frac{\partial C}{\partial b_j^L} &= \delta_j^L \\ &= (a_j^L - y_j)\end{aligned}$$

Λαμβάνοντας υπόψη όλα τα δείγματα εκπαίδευσης:

$$\frac{\partial C}{\partial w_{jk}^L} = \frac{1}{n} \sum_x a_k^{L-1} (a_j^L - y_j)$$

Και

$$\frac{\partial C}{\partial b_j^L} = \frac{1}{n} \sum_x (a_j^L - y_j)$$

Άσκηση 6.1.4. Πίσω διάδοση με softmax και κόστος λογαριθμικής πιθανοφάνειας

Για την εφαρμογή του αλγορίθμου πίσω διάδοσης σε ένα επίπεδο softmax θα πρέπει να καθοριστεί μια έκφραση για την τοπική κλίση του σφάλματος στους νευρώνες εξόδου $\delta_j^L \equiv \partial C / \partial z_j^L$. Να δείξετε ότι μια κατάλληλη έκφραση είναι $\delta_j^L = a_j^L - y_j$. Με την έκφραση αυτή μπορούμε να εφαρμόσουμε τον αλγόριθμο οπισθοδιάδοσης σε ένα δίκτυο που χρησιμοποιεί έξοδο softmax και κόστος λογαριθμικής πιθανοφάνειας.

Απάντηση

Έστω d η επιθυμητή έξοδος (το ψηφίο που αναζητούμε στο MNIST). Θα είναι:

$$\begin{aligned}\delta_j^L &= \frac{\partial C}{\partial z_j^L} \\ &= \frac{\partial C}{\partial a_d^L} \frac{\partial a_d^L}{\partial z_j^L}, \text{ αφού } C = -\ln a_d^L \\ &= -\frac{1}{a_d^L} \frac{\partial a_d^L}{\partial z_j^L}\end{aligned}$$

Το a_j^L μπορεί να γραφεί ως:

$$\begin{aligned}a_j^L &= \frac{e^{z_j^L}}{e^{z_j^L} + \sum_{k \neq j} e^{z_k^L}} \\ &= \frac{1}{1 + e^{-z_j^L} \sum_{k \neq j} e^{z_k^L}} \\ &= (1 + e^{-z_j^L} \sum_{k \neq j} e^{z_k^L})^{-1}\end{aligned}$$

Έστω ότι ο j είναι ο νευρώνας εξόδου που αντιπροσωπεύει το ψηφίο που αναζητούμε, δηλαδή $j = d$. Αντικαθιστώντας την παραπάνω έκφραση για το a_j^L και παραγωγίζοντας:

$$\begin{aligned}
\delta_j^L &= -\frac{1}{a_j^L} (e^{-z_j^L} \sum_{k \neq j} e^{z_k^L}) (a_j^L)^2 \\
&= -a_j^L (1 + e^{-z_j^L} \sum_{k \neq j} e^{z_k^L} - 1) \\
&= -a_j^L \left(\frac{1}{a_j^L} - 1 \right) \\
&= a_j^L - 1 \\
&= a_j^L - y_j, \text{ αφού } y_j = 1
\end{aligned}$$

Αν ωστόσο ο j δεν αντιστοιχεί στο ψηφίο, δηλαδή $j \neq d$:

$$\begin{aligned}
\delta_j^L &= -\frac{1}{a_d^L} (-e^{-z_d^L} e^{z_j^L}) (a_d^L)^2 \\
&= a_d^L e^{-z_d^L} e^{z_j^L} \\
&= \frac{e^{z_d^L}}{\sum_k e^{z_k^L}} e^{-z_d^L} e^{z_j^L} \\
&= \frac{e^{z_j^L}}{\sum_k e^{z_k^L}} \\
&= a_j^L \\
&= a_j^L - y_j, \text{ αφού } y_j = 0
\end{aligned}$$

Άσκηση 6.3.1.

Να δείξετε ότι η τυπική απόκλιση του αθροίσματος $z = \sum_j w_j x_j + b$ είναι $\sqrt{3/2}$.

Απάντηση

Έστω $J = \{j \mid x_j = 1\}$ το σύνολο των μη μηδενικών εισόδων. Το πλήθος τους θα είναι $\frac{n_{in}}{2}$ ή σε αυτήν την περίπτωση 500. Η διακύμανση του z είναι:

$$\begin{aligned}
V(z) &= \sum_{z \in J} V(w_j) + V(b) \\
&= \sum_{z \in J} \frac{1}{n_{in}} + 1 \\
&= \frac{n_{in}}{2} \frac{1}{n_{in}} + 1 \\
&= \frac{3}{2}
\end{aligned}$$

Επομένως η τυπική απόκλιση του z θα είναι:

$$s(z) = \sqrt{\frac{3}{2}}$$

Κεφάλαιο 7

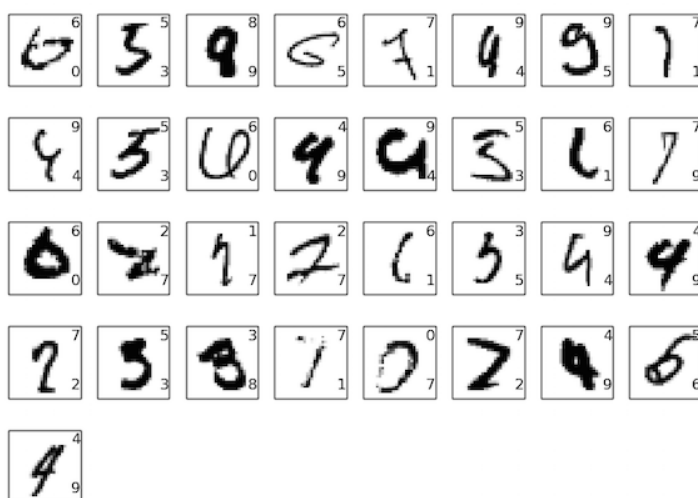
Μοντέλα Νευρωνικών Δικτύων Βαθιάς Μάθησης

Σύνοψη

Το κύριο μέρος αυτού του κεφαλαίου είναι μια εισαγωγή σε έναν από τους πιο ευρέως χρησιμοποιούμενους τύπους βαθέων δικτύων: τα βαθιά συνελκτικά δίκτυα. Θα επεξεργαστούμε ένα λεπτομερές παράδειγμα — τη χρήση συνελκτικών δικτύων για την επίλυση του προβλήματος της ταξινόμησης χειρόγραφων ψηφίων από το σύνολο δεδομένων MNIST:

504192

Θα ξεκινήσουμε την ανάλυση των συνελκτικών δικτύων με τα ρηγά δίκτυα που χρησιμοποιήθηκαν για να επιτεθούν σε αυτό το πρόβλημα στο προηγούμενο κεφάλαιο. Μέσα από πολλές επαναλήψεις θα δημιουργούμε όλο και πιο ισχυρά δίκτυα. Καθώς προχωράμε, θα διερευνήσουμε πολλές ισχυρές τεχνικές: συνελίξεις (convolutions), ομαδοποίηση (pooling), χρήση GPU για να κάνουμε πολύ περισσότερη εκπαίδευση από ό,τι κάναμε με τα ρηγά δίκτυά μας, την αλγοριθμική επέκταση των δεδομένων εκπαίδευσής μας (για να μειώσουμε την υπερεκπαίδευση), τη χρήση της τεχνικής dropout (επίσης για τη μείωση της υπερεκπαίδευσης), τη χρήση συνόλων δικτύων (ensembles) και άλλων. Το αποτέλεσμα θα είναι ένα σύστημα που προσφέρει σχεδόν ανθρώπινη απόδοση. Από τις 10.000 δοκιμαστικές εικόνες MNIST —εικόνες που δεν φαίνονται κατά τη διάρκεια της εκπαίδευσης!— το σύστημά μας θα ταξινομήσει σωστά 9.967. Ακολουθούν οι 33 εικόνες που είναι εσφαλμένες. Σημειώστε ότι η σωστή ταξινόμηση βρίσκεται πάνω δεξιά. Η ταξινόμηση του προγράμματός μας είναι κάτω δεξιά:



Σχήμα 7.1 Εικόνες από το MNIST που ταξινομούνται λανθασμένα.

Πολλά από αυτά είναι δύσκολο ακόμη και για έναν άνθρωπο να τα ταξινομήσει. Εξετάστε π.χ. την τρίτη εικόνα στην επάνω σειρά. Μοιάζει περισσότερο με «9» παρά με «8» που είναι η επίσημη ταξινόμηση. Το δίκτυό μας πιστεύει επίσης ότι είναι «9». Αυτό το είδος «σφάλματος» είναι τουλάχιστον κατανοητό, ίσως ακόμη και αξιόπαινο.

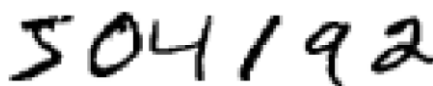
Το υπόλοιπο κεφάλαιο συζητά τη βαθιά μάθηση από μια ευρύτερη και λιγότερο λεπτομερή σκοπιά. Θα ερευνήσουμε εν συντομία άλλα μοντέλα νευρωνικών δικτύων, όπως τα επαναλαμβανόμενα νευρωνικά δίκτυα (RNNs) και τις μονάδες μακράς βραχυπρόθεσμης μνήμης (LSTMs) και πώς τέτοια μοντέλα μπορούν να εφαρμοστούν σε προβλήματα αναγνώρισης ομιλίας, επεξεργασίας φυσικής γλώσσας και άλλων περιοχών.

Προαπαιτούμενη γνώση

Το κεφάλαιο βασίζεται σε προηγούμενα κεφάλαια του βιβλίου, κάνοντας χρήση και ενσωματώνοντας ιδέες που παρουσιάστηκαν νωρίτερα, όπως η πίσω διάδοση, η ομαλοποίηση, η συνάρτηση softmax κ.ο.κ. Το μεγαλύτερο μέρος του κεφαλαίου προέρχεται από τα Nielsen, 2019 και Goodfellow et al, 2016.

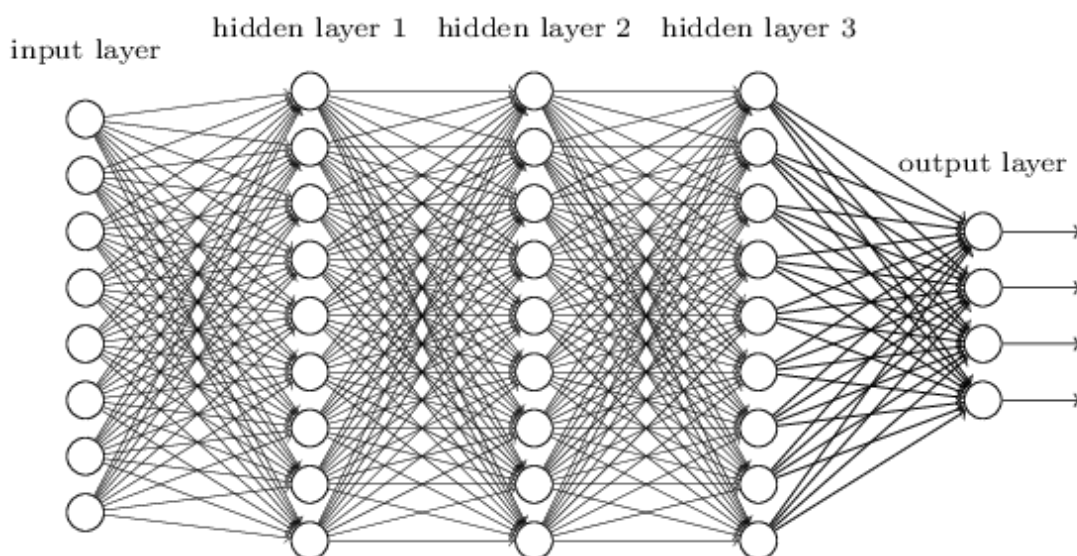
7.1 Εισαγωγή στα συνελκτικικά δίκτυα

Σε προηγούμενα κεφάλαια διδάξαμε στα νευρωνικά μας δίκτυα να κάνουν πολύ καλή δουλειά αναγνωρίζοντας εικόνες χειρόγραφων ψηφίων:



Σχήμα 7.2 Χειρόγραφο ψηφία.

Το κάναμε αυτό χρησιμοποιώντας δίκτυα στα οποία τα γειτονικά επίπεδα δικτύου είναι πλήρως συνδεδεμένα μεταξύ τους· δηλαδή κάθε νευρώνας στο δίκτυο συνδέεται με κάθε νευρώνα σε γειτονικά επίπεδα:



Σχήμα 7.3 Πλήρως συνδεδεμένο TNA.

Συγκεκριμένα, για κάθε εικονοστοιχείο στην εικόνα εισόδου, κωδικοποιήσαμε την ένταση του εικονοστοιχείου ως τιμή για έναν αντίστοιχο νευρώνα στο επίπεδο εισόδου. Για τις εικόνες 28×28 pixel που χρησιμοποιούμε, αυτό σημαίνει ότι το δίκτυό μας έχει $784784 (=28 \times 28)$ νευρώνες εισόδου. Στη συνέχεια εκπαιδεύσαμε τα βάρη και τις πολώσεις του δικτύου, έτσι ώστε η έξοδος του δικτύου να προσδιορίζει —ελπίζουμε!— ορθά την εικόνα εισόδου: «0», «1», «2», ..., «8» ή «9».

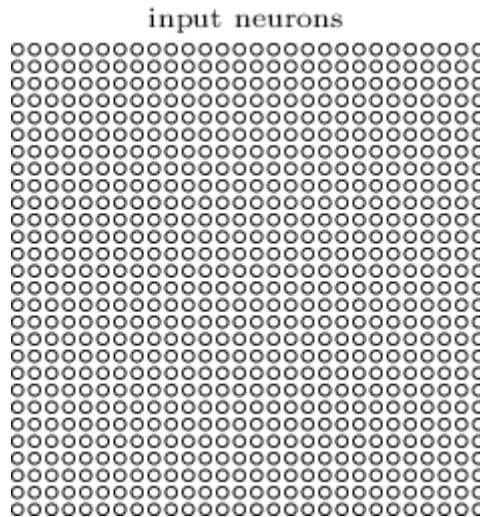
Τα προηγούμενα δίκτυά μας λειτουργούν αρκετά καλά: έχουμε αποκτήσει ακρίβεια ταξινόμησης καλύτερη από 98%, χρησιμοποιώντας δεδομένα εκπαίδευσης και ελέγχου από το σύνολο δεδομένων χειρόγραφων ψηφίων MNIST. Αλλά μετά από σκέψη αντιλαμβανόμαστε πως είναι περίεργο να χρησιμοποιούμε δίκτυα με πλήρως συνδεδεμένα επίπεδα για να ταξινομήσουμε εικόνες. Ο λόγος είναι ότι μια τέτοια αρχιτεκτονική δικτύου δεν λαμβάνει υπόψη τη χωρική δομή των εικόνων. Π.χ. αντιμετωπίζει εικονοστοιχεία εισόδου που είναι πολύ μακριά ή πολύ κοντά μεταξύ τους ακριβώς στην ίδια βάση. Επιπλέον, έννοιες όπως η χωρική δομή πρέπει να συναχθούν από τα δεδομένα εκπαίδευσης. Αλλά τι γίνεται αν, αντί να ξεκινήσουμε με μια αρχιτεκτονική δικτύου που είναι *tabula rasa*, χρησιμοποιήσουμε μια αρχιτεκτονική που προσπαθεί να εκμεταλλευτεί τη χωρική δομή; Σε αυτήν την ενότητα περιγράφονται τα συνελκτικά νευρωνικά δίκτυα (convolutional neural networks, CNNs).

Αυτά τα δίκτυα χρησιμοποιούν μια ειδική αρχιτεκτονική που είναι ιδιαίτερα καλά προσαρμοσμένη για την ταξινόμηση των εικόνων. Η χρήση της καθιστά τα συνελκτικά δίκτυα γρήγορα στην εκπαίδευση. Κάτι τέτοιο με τη σειρά του μας βοηθά να εκπαιδεύσουμε βαθιά δίκτυα πολλών επιπέδων, τα οποία είναι πολύ καλά στην ταξινόμηση εικόνων. Σήμερα βαθιά συνελκτικά δίκτυα ή κάποια στενή παραλλαγή τους χρησιμοποιούνται στα περισσότερα νευρωνικά δίκτυα για αναγνώριση εικόνας.

Τα συνελκτικά νευρωνικά δίκτυα χρησιμοποιούν τρεις βασικές ιδέες: *τοπικά πεδία πρόσληψης (local receptive fields)*, *κοινόχρηστα βάρη (shared weights)* και *ομαδοποίηση (pooling)*. Ας δούμε καθεμία από αυτές τις ιδέες με τη σειρά.

7.1.1 Τοπικά πεδία πρόσληψης (local receptive fields)

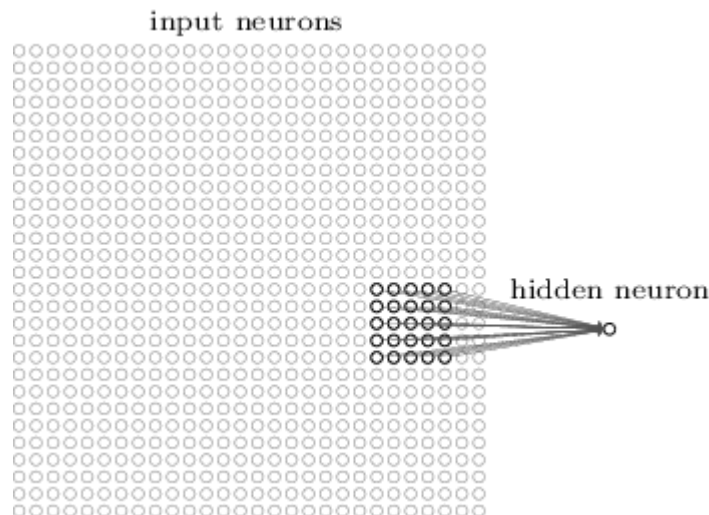
Στα πλήρως συνδεδεμένα επίπεδα που παρουσιάστηκαν νωρίτερα οι εισοδοί απεικονίστηκαν ως κάθετη γραμμή νευρώνων. Σε ένα περίπλοκο δίκτυο είναι καλύτερα να σκεφτούμε τις εισόδους ως ένα τετράγωνο 28×28 νευρώνων, των οποίων οι τιμές αντιστοιχούν στις 28×28 εντάσεις pixel που χρησιμοποιούμε ως εισόδους:



Σχήμα 7.4 Νευρώνες εισόδου 28×28 .

Ως συνήθως, θα συνδέσουμε τα εικονοστοιχεία εισόδου σε ένα επίπεδο κρυφών νευρώνων. Ωστόσο, δεν θα συνδέσουμε κάθε εικονοστοιχείο εισόδου σε όλους τους κρυφούς νευρώνες. Αντίθετα κάνουμε συνδέσεις μόνο σε μικρές, τοπικές περιοχές της εικόνας εισόδου.

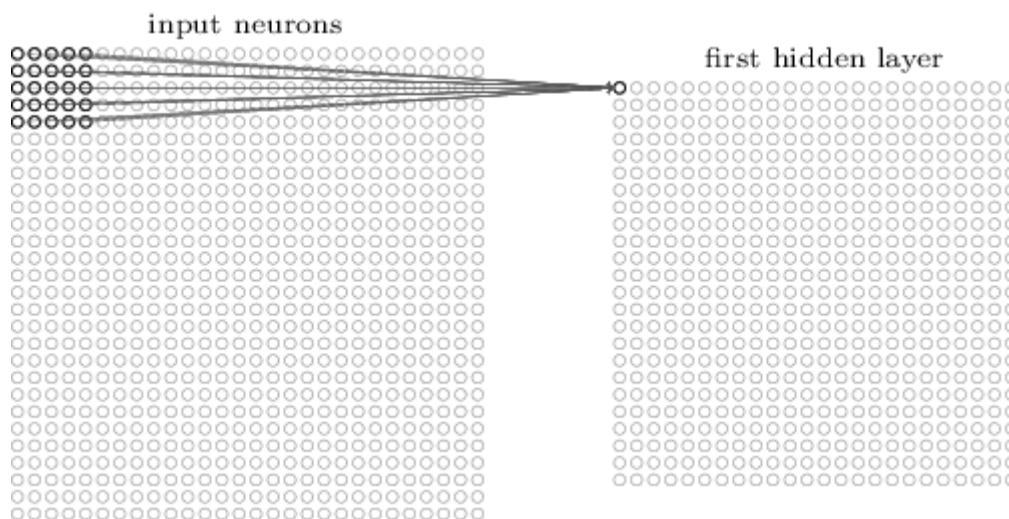
Για να είμαστε πιο ακριβείς, κάθε νευρώνας στο πρώτο κρυφό επίπεδο θα συνδεθεί με μια μικρή περιοχή των νευρώνων εισόδου· ας πούμε μια περιοχή 5×5 , που αντιστοιχεί σε 25 εικονοστοιχεία εισόδου. Έτσι, για έναν συγκεκριμένο κρυμμένο νευρώνα, μπορεί να έχουμε συνδέσεις που μοιάζουν με αυτό:



Σχήμα 7.5 Σύνδεση εισόδων με κρυφό νευρώνα.

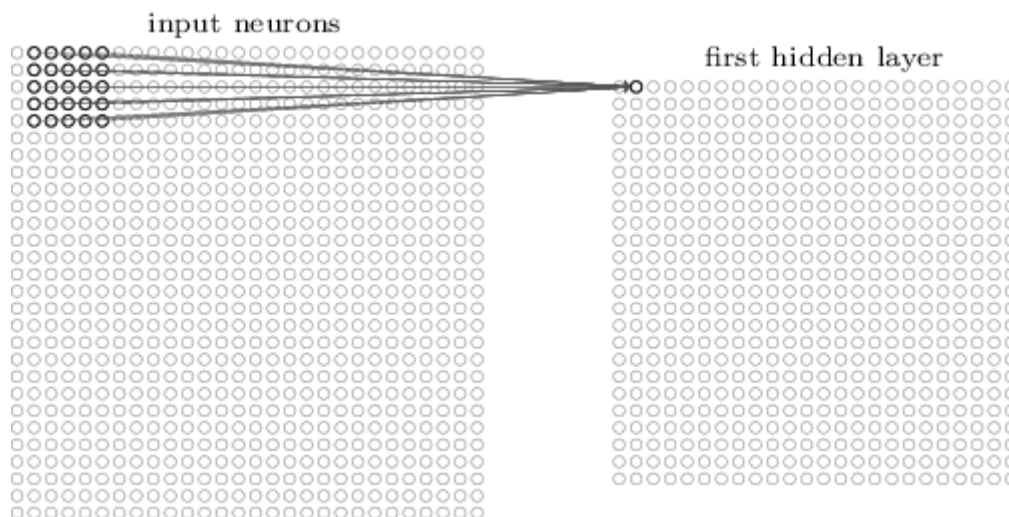
Αυτή η περιοχή στην εικόνα εισόδου ονομάζεται *τοπικό πεδίο πρόσληψης* για τον κρυφό νευρώνα. Είναι ένα μικρό παράθυρο στα εικονοστοιχεία εισόδου. Κάθε σύνδεση μαθαίνει ένα βάρος. Ο κρυφός νευρώνας μαθαίνει επίσης μια συνολική πόλωση. Μπορείτε να σκεφτείτε ότι ο συγκεκριμένος κρυφός νευρώνας μαθαίνει να αναλύει το δικό του, συγκεκριμένο τοπικό πεδίο πρόσληψης.

Στη συνέχεια μετακινούμε το τοπικό πεδίο πρόσληψης σε ολόκληρη την εικόνα εισόδου. Για κάθε τοπικό πεδίο υπάρχει ένας διαφορετικός κρυφός νευρώνας στο πρώτο κρυφό επίπεδο. Για να το απεικονίσουμε, ας ξεκινήσουμε με ένα τοπικό πεδίο στην επάνω αριστερή γωνία:



Σχήμα 7.6 Τοπικό πεδίο πρόσληψης στην επάνω αριστερή γωνία των εικονοστοιχείων/εισόδων.

Έπειτα μετατοπίζουμε το τοπικό πεδίο πρόσληψης κατά ένα pixel προς τα δεξιά (δηλαδή κατά έναν νευρώνα), για να συνδεθούμε με έναν δεύτερο κρυφό νευρώνα κοκ, χτίζοντας έτσι το πρώτο κρυφό επίπεδο. Σημειώνεται ότι έχουμε ως είσοδο μια εικόνα 28x28 και τοπικά πεδία πρόσληψης 5x5, άρα θα υπάρχουν 24x24 νευρώνες στο κρυφό επίπεδο, διότι μπορούμε να μετακινήσουμε το πεδίο μόνο κατά 23 νευρώνες προς τα δεξιά (ή προς τα κάτω), προτού συμπέσουμε με τη δεξιά (ή την κάτω) πλευρά της εικόνας εισόδου.



Σχήμα 7.7 Μετατόπιση του τοπικού πεδίου πρόσληψης κατά ένα pixel.

Ορισμένες φορές χρησιμοποιείται και διαφορετικό μέγεθος βήματος αντί για έναν νευρώνα π.χ. θα μπορούσαμε να μετακινήσουμε το τοπικό πεδίο 22 pixel στα δεξιά (ή κάτω). Σε αυτήν την περίπτωση θα λέγαμε ότι χρησιμοποιείται ένα μέγεθος βήματος 2. Στο παρόν κεφάλαιο θα παραμείνουμε ως επί το πλείστον σε μέγεθος βήματος 1, αλλά αξίζει να γνωρίζουμε ότι μερικές φορές γίνονται πειραματισμοί με διαφορετικά μεγέθη βήματος. Π.χ. μπορεί να χρησιμοποιηθεί διασταυρούμενη επικύρωση για να διαπιστωθεί ποιο είναι το καλύτερο μήκος βήματος για ένα πρόβλημα. Αντίστοιχα η ίδια προσέγγιση μπορεί να χρησιμοποιηθεί για να καθοριστεί το μέγεθος του τοπικού πεδίου πρόσληψης. Γενικά μεγάλα τοπικά πεδία πρόσληψης τείνουν να βοηθούν όταν οι εικόνες εισόδου είναι σημαντικά μεγαλύτερες από τις 28x28 εικόνες του MNIST.

7.1.2 Κοινόχρηστα βάρη και πολώσεις

Κάθε κρυφός νευρώνας έχει μια πόλωση και 5x5 βάρη που συνδέονται με το δικό του τοπικό πεδίο πρόσληψης. Ωστόσο, τα βάρη και οι πολώσεις για καθέναν από τους 24x24 κρυφούς νευρώνες είναι τα ίδια με άλλα λόγια για έναν νευρώνα στη θέση (j, k) η έξοδος θα είναι:

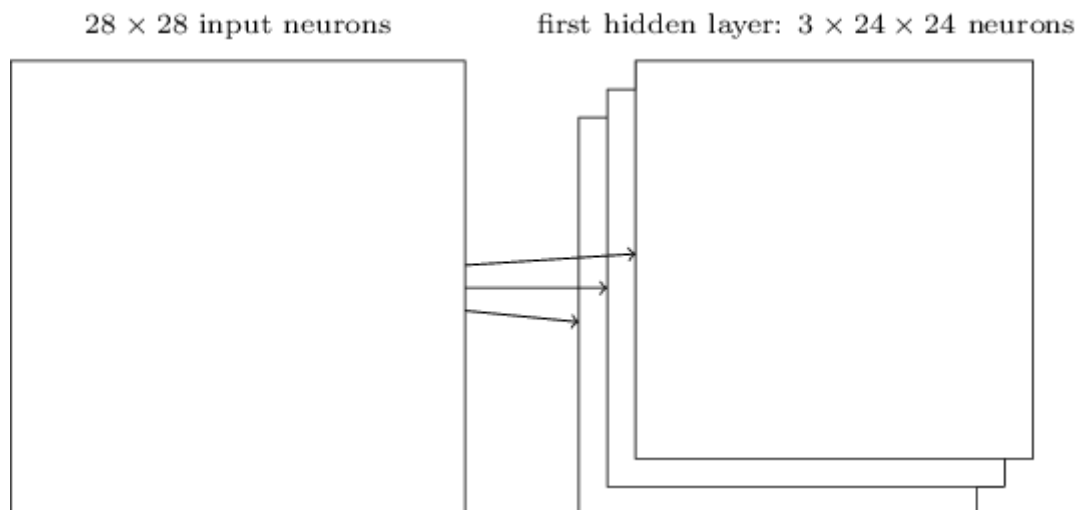
$$\sigma \left(b + \sum_{l=0}^4 \sum_{m=0}^4 w_{l,m} a_{j+l, k+m} \right). \quad (7.1)$$

Εδώ σ είναι η συνάρτηση ενεργοποίησης — πιθανώς η λογιστική συνάρτηση, b είναι η κοινή τιμή για την πόλωση, $w_{l,m}$ είναι ένας 5x5 πίνακας κοινόχρηστων βαρών και τέλος $a_{x,y}$ είναι η τιμή της εισόδου στη θέση (x, y) .

Αυτό σημαίνει ότι όλοι οι νευρώνες στο 1ο κρυφό επίπεδο εντοπίζουν ακριβώς το ίδιο χαρακτηριστικό, απλώς σε διαφορετικές περιοχές της εικόνας εισόδου. Ένα χαρακτηριστικό μπορεί να είναι ένα ερέθισμα που θα προκαλέσει την ενεργοποίηση του νευρώνα. Μπορεί να είναι μια ακμή σε μια εικόνα ή κάποιο άλλο είδος σχήματος. Π.χ. ας υποθέσουμε ότι τα βάρη και η πόλωση είναι τέτοια ώστε ο κρυφός νευρώνας να καταλαβαίνει μια κάθετη ακμή στο τοπικό πεδίο πρόσληψης. Η ικανότητα αυτή πιθανότατα είναι χρήσιμη και για άλλες περιοχές της εικόνας. Επομένως, είναι χρήσιμο να εφαρμόσουμε τον ίδιο ανιχνευτή χαρακτηριστικών σε όλη την εικόνα. Τα συνελκτικά δίκτυα είναι καλά προσαρμοσμένα να ανθίστανται σε πιθανή μεταβλητότητα μετασχηματισμού (translation invariance): Αν μετακινήσουμε την εικόνα μιας γάτας λίγο προς τα δεξιά, εξακολουθεί να είναι εικόνα γάτας.

Για τον λόγο αυτό μερικές φορές καλούμε την αντιστοίχιση από το επίπεδο εισόδου στο κρυφό επίπεδο *χάρτη χαρακτηριστικών (feature map)*. Ονομάζουμε τα βάρη που καθορίζουν τον χάρτη χαρακτηριστικών *κοινά βάρη* και ονομάζουμε την πόλωση που καθορίζει τον χάρτη χαρακτηριστικών με αυτόν τον τρόπο *κοινή πόλωση*. Τα κοινόχρηστα βάρη και πολώσεις συχνά λέγεται ότι καθορίζουν έναν *πυρήνα* ή ένα *φίλτρο*. Στη βιβλιογραφία μερικές φορές χρησιμοποιούνται αυτοί οι όροι και με ελαφρώς διαφορετικούς τρόπους, γι' αυτό θα παρουσιαστούν παρακάτω συγκεκριμένα παραδείγματα.

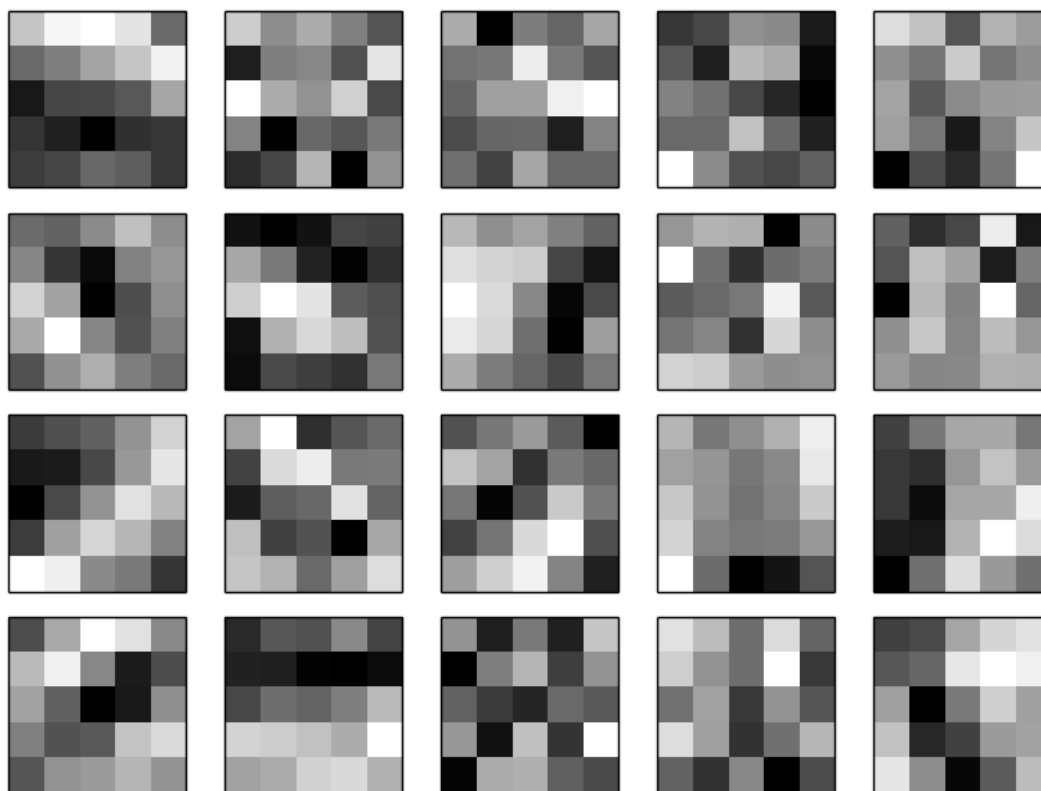
Η δομή δικτύου που έχει περιγραφεί μέχρι στιγμής μπορεί να ανιχνεύσει μόνο ένα είδος τοπικού χαρακτηριστικού. Για να κάνουμε αναγνώριση εικόνας θα χρειαστούμε περισσότερους από έναν χάρτες χαρακτηριστικών. Έτσι ένα πλήρες συνελκτικό επίπεδο αποτελείται από διάφορους χάρτες χαρακτηριστικών:



Σχήμα 7.8 Συνελκτικό επίπεδο για αναγνώριση εικόνας.

Στο παράδειγμα που παρουσιάζεται υπάρχουν τρεις χάρτες χαρακτηριστικών. Κάθε χάρτης ορίζεται από ένα σύνολο 5x5 κοινόχρηστων βαρών και μία κοινή πόλωση. Το αποτέλεσμα είναι ότι το δίκτυο μπορεί να ανιχνεύσει τρία διαφορετικά είδη χαρακτηριστικών, με κάθε χαρακτηριστικό να είναι ανιχνεύσιμο σε ολόκληρη την εικόνα.

Έχουμε δείξει μόνο τρεις χάρτες χαρακτηριστικών, για να κρατήσουμε το διάγραμμα απλό. Ωστόσο, στην πράξη τα συνελκτικά δίκτυα μπορούν να χρησιμοποιούν περισσότερους (και ίσως πολλούς περισσότερους) χάρτες χαρακτηριστικών. Ένα από τα πρώτα συνελκτικά δίκτυα, το LeNet-5, χρησιμοποίησε έξι χάρτες χαρακτηριστικών, ο καθένας συνδεδεμένος με ένα τοπικό πεδίο πρόσληψης 5×5 , για να αναγνωρίσει τα ψηφία MNIST. Έτσι το παράδειγμα που απεικονίζεται παραπάνω είναι στην πραγματικότητα αρκετά κοντά στο LeNet-5. Στα παραδείγματα που αναπτύσσουμε αργότερα στο κεφάλαιο θα χρησιμοποιήσουμε συνελκτικά επίπεδα με 20 και 40 χάρτες χαρακτηριστικών. Ας ρίξουμε μια γρήγορη ματιά σε μερικά από τα χαρακτηριστικά που μαθαίνονται:



Σχήμα 7.9 Ανίχνευση χαρακτηριστικών.

Οι 20 εικόνες αντιστοιχούν σε 20 διαφορετικούς χάρτες χαρακτηριστικών (ή φίλτρα ή πυρήνες). Κάθε χάρτης αντιπροσωπεύεται ως εικόνα 5×5 μπλοκ, καθένα από τα οποία αντιστοιχεί στα 5×5 βάρη του 5×5 τοπικού πεδίου πρόσληψης. Τα πιο λευκά μπλοκ σημαίνουν μικρότερο (συνήθως πιο αρνητικό) βάρος, οπότε ο χάρτης δυνατοτήτων ανταποκρίνεται λιγότερο στα αντίστοιχα εικονοστοιχεία εισόδου. Τα πιο σκούρα μπλοκ σημαίνουν μεγαλύτερο βάρος, οπότε ο χάρτης δυνατοτήτων ανταποκρίνεται περισσότερο στα αντίστοιχα εικονοστοιχεία εισόδου. Σε γενικές γραμμές μιλώντας, οι παραπάνω εικόνες δείχνουν τον τύπο των χαρακτηριστικών στα οποία ανταποκρίνεται το συνελκτικό επίπεδο.

Είναι σαφές ότι υπάρχει χωροταξική δομή που δεν είναι τυχαία: πολλά από τα χαρακτηριστικά έχουν σαφείς υποπεριφέρειες φωτός και σκότους. Αυτό δείχνει ότι το δίκτυό μας μαθαίνει πραγματικά στοιχεία που σχετίζονται με τη χωρική δομή. Ωστόσο, πέρα από αυτό, είναι δύσκολο να δούμε τι μαθαίνουν αυτοί οι ανιχνευτές χαρακτηριστικών. Σίγουρα δεν μαθαίνουν (ας πούμε) τα [φίλτρα Gabor](#) που έχουν χρησιμοποιηθεί σε πολλές παραδοσιακές προσεγγίσεις στην αναγνώριση εικόνων. Στην πραγματικότητα υπάρχει τώρα αρκετή έρευνα για την καλύτερη κατανόηση των χαρακτηριστικών που μαθαίνονται από τα συνελκτικά δίκτυα (Zeiler & Fergus, 2013).

Ένα μεγάλο πλεονέκτημα των κοινόχρηστων βαρών και πολώσεων είναι ότι μειώνουν σημαντικά τον αριθμό των παραμέτρων που εμπλέκονται σε ένα συνελκτικό δίκτυο. Για κάθε χάρτη δυνατοτήτων χρειαζόμαστε $25 = 5 \times 5$ κοινόχρηστα βάρη, καθώς και μία κοινή πόλωση. Έτσι κάθε χάρτης χαρακτηριστικών απαιτεί 26 παραμέτρους. Εάν έχουμε 20 χάρτες χαρακτηριστικών, αυτοί θα αντιστοιχούν σε $20 \times 26 = 520$ παραμέτρους που καθορίζουν το συνελκτικό επίπεδο. Συγκριτικά, ας υποθέσουμε ότι είχαμε ένα πλήρως

συνδεδεμένο 1ο επίπεδο, με $784 = 28 \times 28$ νευρώνες εισόδου και ένα σχετικά μέτριο πλήθος 30 κρυφών νευρώνων, όπως χρησιμοποιήσαμε σε πολλά από τα παραδείγματα νωρίτερα στο βιβλίο. Αυτό ισοδυναμεί με 784×30 βάρη, συν επιπλέον 30 πολώσεις, δίνοντας συνολικά 23.550 παραμέτρους. Με άλλα λόγια, το πλήρως συνδεδεμένο επίπεδο θα έχει περισσότερες από 40 φορές περισσότερες παραμέτρους από το συνελκτικό επίπεδο.

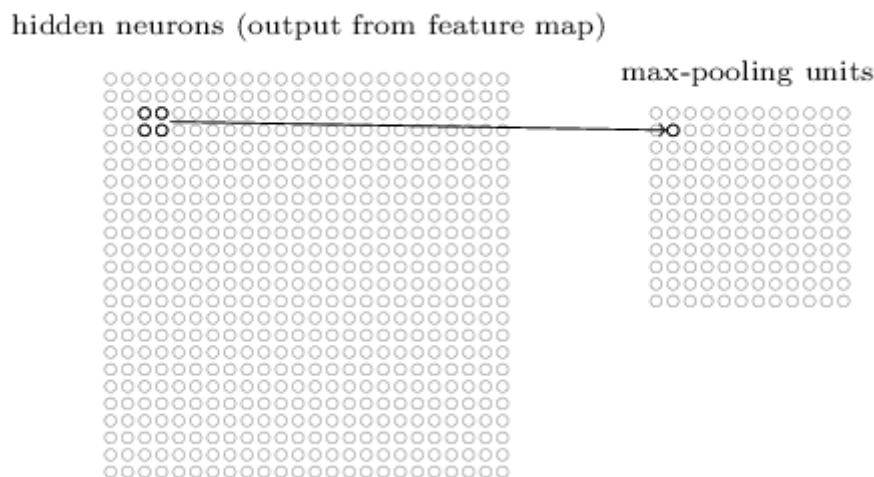
Φυσικά δεν μπορούμε πραγματικά να κάνουμε μια άμεση σύγκριση μεταξύ του αριθμού των παραμέτρων, καθώς τα δύο μοντέλα είναι διαφορετικά με ουσιαστικούς τρόπους. Αλλά, διαισθητικά, φαίνεται πιθανό ότι η αξιοποίηση της μεταφραστικής αμεταβλητότητας από το συνελκτικό επίπεδο θα μειώσει τον αριθμό των παραμέτρων που χρειάζονται για να έχει την ίδια απόδοση με το πλήρως συνδεδεμένο μοντέλο. Αυτό με τη σειρά του θα οδηγήσει σε ταχύτερη εκπαίδευση για το συνελκτικό μοντέλο και τελικά θα μας βοηθήσει να οικοδομήσουμε βαθιά δίκτυα χρησιμοποιώντας συνελκτικά επίπεδα.

Το όνομα *συνελκτικά* προέρχεται από το γεγονός ότι η εξίσωση (7.1) αναφέρεται ενίοτε ως *συνέλιξη*. Πιο συγκεκριμένα, η εξίσωση μπορεί να γραφεί ως $a^1 = \sigma(b + w * a^0)$, όπου το a^1 δηλώνει το σύνολο των ενεργοποιήσεων εξόδου για έναν χάρτη χαρακτηριστικών, το a^0 είναι το σύνολο των ενεργοποιήσεων εισόδου και το $*$ καλείται τελεστής συνέλιξης.

7.1.3 Επίπεδα ομαδοποίησης (pooling layers)

Εκτός από τα συνελκτικά επίπεδα που μόλις περιγράφηκαν, τα συνελκτικά δίκτυα περιέχουν επίσης *επίπεδα ομαδοποίησης*. Τα επίπεδα ομαδοποίησης χρησιμοποιούνται συνήθως αμέσως μετά τα συνελκτικά επίπεδα: αυτό που κάνουν είναι να απλοποιούν τις πληροφορίες στην έξοδο από το συνελκτικό επίπεδο.

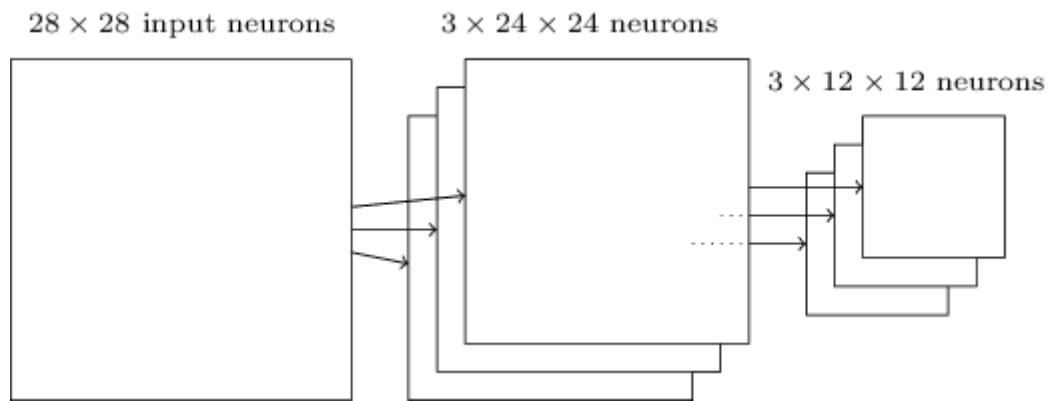
Αναλυτικά ένα επίπεδο ομαδοποίησης παίρνει την έξοδο από κάθε χάρτη χαρακτηριστικών του επιπέδου συνέλιξης και ετοιμάζει έναν συμπυκνωμένο χάρτη χαρακτηριστικών. Π.χ. κάθε μονάδα στο επίπεδο ομαδοποίησης μπορεί να συνοψίσει μια περιοχή (ας πούμε) 2×2 νευρώνων από το προηγούμενο επίπεδο. Ως συγκεκριμένο παράδειγμα δίνεται μια κοινή διαδικασία για την ομαδοποίηση, γνωστή ως *max-pooling*: μια μονάδα ομαδοποίησης απλώς δίνει ως έξοδο τη μέγιστη ενεργοποίηση στην περιοχή εισόδου 2×2 , όπως απεικονίζεται στο ακόλουθο διάγραμμα:



Σχήμα 7.10 Max-pooling.

Εφόσον έχουμε 24×24 νευρώνες εξόδου από το συνελκτικό επίπεδο, μετά την ομαδοποίηση θα έχουμε 12×12 νευρώνες.

Όπως αναφέρθηκε παραπάνω, το συνελκτικό επίπεδο συνήθως περιλαμβάνει περισσότερους από έναν χάρτες χαρακτηριστικών. Εφαρμόζουμε max-pooling σε κάθε χάρτη χαρακτηριστικών ξεχωριστά. Έτσι, αν υπήρχαν τρεις χάρτες χαρακτηριστικών, τα συνδυασμένα συνελκτικά και max-pooling επίπεδα θα έμοιαζαν με:



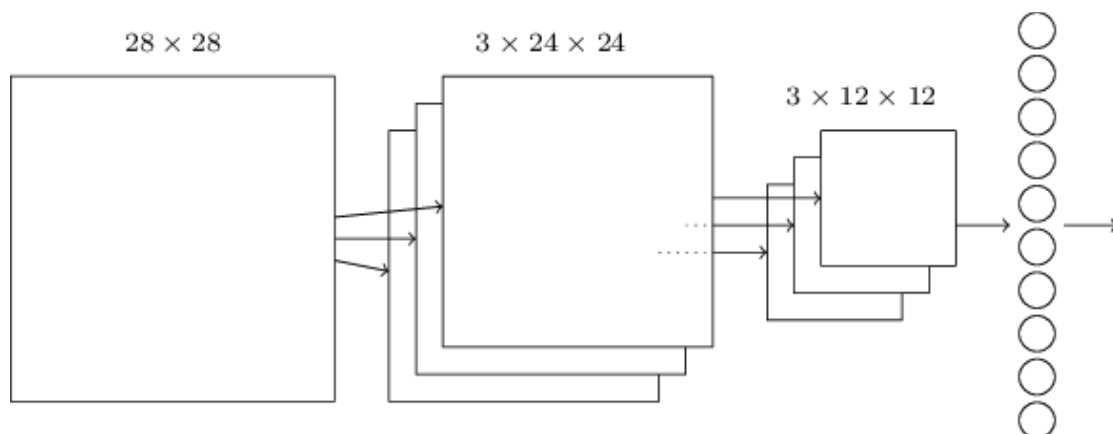
Σχήμα 7.11 Συνδυασμός συνελκτικών επιπέδων και επιπέδων ομαδοποίησης.

Μπορούμε να σκεφτούμε το max-pooling ως έναν τρόπο που έχει το δίκτυο να ρωτήσει αν ένα συγκεκριμένο χαρακτηριστικό βρίσκεται οπουδήποτε σε μια περιοχή της εικόνας. Στη συνέχεια απορρίπτει τις ακριβείς πληροφορίες θέσης. Η διαίσθηση είναι ότι μόλις βρεθεί ένα χαρακτηριστικό, η ακριβής τοποθεσία του δεν είναι τόσο σημαντική όσο η θέση του σε σχέση με άλλα χαρακτηριστικά. Ένα μεγάλο πλεονέκτημα είναι ότι υπάρχουν πολύ λιγότερα ομαδοποιημένα χαρακτηριστικά και έτσι ενισχύεται η μείωση του αριθμού των παραμέτρων που απαιτούνται σε μεταγενέστερα επίπεδα.

Το max-pooling δεν είναι η μόνη τεχνική που χρησιμοποιείται για την ομαδοποίηση. Μια άλλη κοινή προσέγγιση είναι γνωστή ως *L2 pooling*. Εδώ, αντί να λάβουμε τη μέγιστη ενεργοποίηση μιας περιοχής νευρώνων 2×2 , παίρνουμε την τετραγωνική ρίζα του αθροίσματος των τετραγώνων των ενεργοποιήσεων στην περιοχή 2×2 . Ενώ οι λεπτομέρειες είναι διαφορετικές, η διαίσθηση είναι παρόμοια με το max-pooling: το L2 pooling είναι ένας τρόπος συμπίκνωσης πληροφοριών από το συνελκτικό επίπεδο. Και οι δύο τεχνικές έχουν χρησιμοποιηθεί ευρέως στην πράξη. Μερικές φορές χρησιμοποιούνται και άλλοι τύποι ομαδοποίησης. Σε κάθε περίπτωση μπορούν να χρησιμοποιηθούν δεδομένα επικύρωσης για τη σύγκριση πολλών διαφορετικών προσεγγίσεων ομαδοποίησης και στη συνέχεια να επιλεγεί η προσέγγιση που λειτουργεί καλύτερα.

7.1.4 Πλήρες συνελκτικό δίκτυο

Μπορούμε τώρα να συγκεντρώσουμε όλες αυτές τις ιδέες για να σχηματίσουμε ένα πλήρες συνελκτικό νευρωνικό δίκτυο. Είναι παρόμοιο με την αρχιτεκτονική που μόλις εξετάσαμε, αλλά έχει την προσθήκη ενός στρώματος 10 νευρώνων εξόδου, που αντιστοιχεί στις 10 πιθανές τιμές για τα ψηφία MNIST («0», «1», «2» κλπ.):



Σχήμα 7.12 Πλήρες συνελκτικό δίκτυο.

Το δίκτυο ξεκινά με νευρώνες εισόδου 28×28 , οι οποίοι χρησιμοποιούνται για την κωδικοποίηση των εντάσεων pixel για την εικόνα MNIST. Στη συνέχεια ακολουθεί ένα συνελκτικό επίπεδο χρησιμοποιώντας ένα τοπικό πεδίο πρόσληψης 5×5 και 3 χάρτες χαρακτηριστικών. Το αποτέλεσμα είναι ένα επίπεδο $3 \times 24 \times 24$ κρυφών νευρώνων χαρακτηριστικών. Το επόμενο βήμα είναι ένα επίπεδο max-pooling που εφαρμόζεται σε

περιοχές 2×2 , σε καθέναν από τους 3 χάρτες χαρακτηριστικών. Το αποτέλεσμα είναι ένα στρώμα $3 \times 12 \times 12$ κρυφών νευρώνων χαρακτηριστικών.

Το τελικό επίπεδο συνδέσεων στο δίκτυο είναι ένα πλήρως συνδεδεμένο επίπεδο· δηλαδή το συγκεκριμένο επίπεδο συνδέει *κάθε* νευρώνα από το max-pooling επίπεδο σε καθέναν από τους 10 νευρώνες εξόδου. Αυτή η πλήρως συνδεδεμένη αρχιτεκτονική είναι η ίδια με αυτή που χρησιμοποιήσαμε σε προηγούμενα κεφάλαια. Σημειώστε ωστόσο ότι στο παραπάνω διάγραμμα έχει χρησιμοποιηθεί ένα μόνο βέλος, για απλότητα, αντί να φαίνονται όλες οι συνδέσεις.

Η παραπάνω συνελκτική αρχιτεκτονική είναι αρκετά διαφορετική από τις αρχιτεκτονικές που χρησιμοποιούνται σε προηγούμενα κεφάλαια. Αλλά η συνολική εικόνα είναι παρόμοια: ένα δίκτυο κατασκευασμένο από πολλές απλές μονάδες, των οποίων οι συμπεριφορές καθορίζονται από τα βάρη και τις πολώσεις τους. Ο γενικός στόχος εξακολουθεί να είναι ο ίδιος: να χρησιμοποιηθούν δεδομένα εκπαίδευσης για την εκμάθηση των βαρών και των πολώσεων του δικτύου, έτσι ώστε το δίκτυο να κάνει καλή δουλειά ταξινομώντας τα ψηφία εισόδου.

Συγκεκριμένα, όπως και νωρίτερα στο βιβλίο, θα εκπαιδεύσουμε το δίκτυό μας χρησιμοποιώντας στοχαστική κατάβαση κλίσης και πίσω διάδοση του σφάλματος. Αυτό προχωρά ως επί το πλείστον ακριβώς με τον ίδιο τρόπο, όπως και στα προηγούμενα κεφάλαια. Ωστόσο, πρέπει να προβούμε σε ορισμένες τροποποιήσεις στη διαδικασία πίσω διάδοσης. Ο λόγος είναι ότι η προηγούμενη παραγωγή του αλγορίθμου πίσω διάδοσης ήταν για δίκτυα με πλήρως συνδεδεμένα επίπεδα, εντούτοις είναι σχετικά απλό να τροποποιηθεί η παραγωγή για συνελκτικά και max-pooling επίπεδα.

7.2 Συνελκτικά νευρωνικά δίκτυα στην πράξη

Είδαμε τις βασικές ιδέες πίσω από τα συνελκτικά νευρωνικά δίκτυα. Ας δούμε πώς λειτουργούν στην πράξη, υλοποιώντας ορισμένα συνελκτικά δίκτυα και εφαρμόζοντάς τα στο πρόβλημα ταξινόμησης ψηφίων MNIST. Στην ενότητα αυτή παρουσιάζονται και συγκρίνονται τα αποτελέσματα από διάφορες δοκιμές και πειράματα για την επίλυση του συγκεκριμένου προβλήματος. Ο βασικός τύπος συνελκτικού δικτύου είναι εμπνευσμένος από το LeCun et al, 1998. Ο κώδικας που χρησιμοποιήθηκε είναι διαθέσιμος στο [GitHub](#).

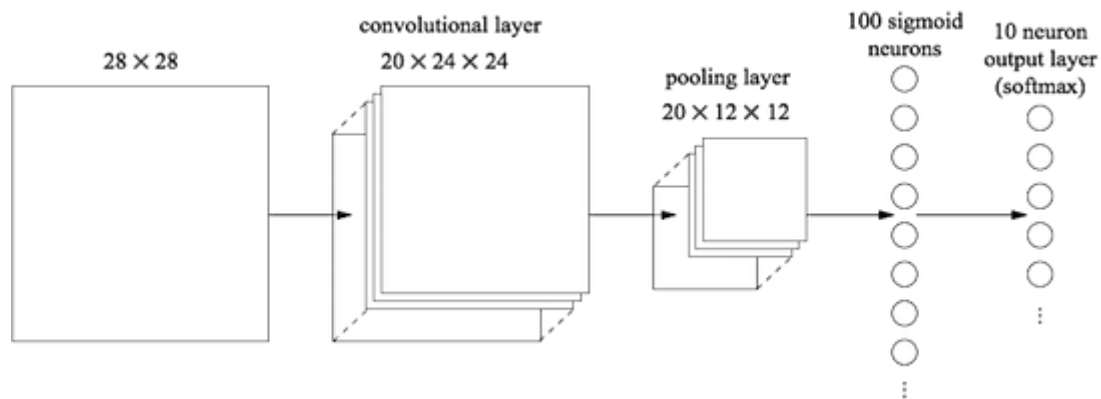
Ως βάση σύγκρισης θα ξεκινήσουμε με μια ρηχή αρχιτεκτονική, χρησιμοποιώντας μόνο ένα κρυφό επίπεδο που περιέχει 100 νευρώνες. Θα εκπαιδεύσουμε για 60 εποχές, με ρυθμό μάθησης $\eta=0,1$, μέγεθος mini-batch 10 και καθόλου ομαλοποίηση.

Να σημειωθεί ότι στην ενότητα προσδιορίζεται ακριβώς ο αριθμός των εποχών. Στην πράξη αξίζει να χρησιμοποιηθεί πρόωρο σταμάτημα (Ενότητα 6.2): δηλαδή να καταγράφεται η ακρίβεια στο σύνολο επικύρωσης και η εκπαίδευση να σταματά όταν η ακρίβεια επικύρωσης έχει σταματήσει να βελτιώνεται.

Η καλύτερη ακρίβεια ταξινόμησης που επιτεύχθηκε είναι 97,80%. Πρόκειται για την ακρίβεια ταξινόμησης στα δεδομένα ελέγχου, η οποία μετρήθηκε σε εκείνη την εποχή εκπαίδευσης που δίνει την καλύτερη ακρίβεια στα δεδομένα επικύρωσης. Η χρήση των δεδομένων επικύρωσης για να αποφασιστεί πότε θα μετρηθεί το σύνολο ελέγχου βοηθά στην αποφυγή της υπερεκπαίδευσης στα δεδομένα ελέγχου. Επίσης, το δίκτυο χρησιμοποιεί ένα επίπεδο εξόδου softmax και συνάρτηση κόστους λογαριθμικής πιθανοφάνειας (log-likelihood, βλ. Κεφάλαιο 6).

Μπορούμε να πετύχουμε καλύτερα από αυτά τα αποτελέσματα χρησιμοποιώντας μια βαθύτερη αρχιτεκτονική δικτύου;

Ας ξεκινήσουμε εισάγοντας ένα συνελκτικό επίπεδο, ακριβώς στην αρχή του δικτύου. Θα χρησιμοποιήσουμε 55 επί 55 τοπικά πεδία πρόσληψης, μήκος βήματος 11 και 20 χάρτες χαρακτηριστικών. Θα εισαγάγουμε επίσης ένα επίπεδο max-pooling, το οποίο συνδυάζει τα χαρακτηριστικά χρησιμοποιώντας παράθυρα ομαδοποίησης 2×2 . Έτσι η συνολική αρχιτεκτονική δικτύου μοιάζει πολύ με την αρχιτεκτονική που συζητήθηκε στην τελευταία ενότητα, αλλά με ένα επιπλέον πλήρως συνδεδεμένο επίπεδο:



Σχήμα 7.13 Εισαγωγή ενός συνελκτικού επιπέδου στην αρχή του δικτύου.

Σε αυτήν την αρχιτεκτονική μπορούμε να σκεφτούμε τα συνελκτικά επίπεδα και τα επίπεδα ομαδοποίησης ως μάθηση για την τοπική χωρική δομή στην εικόνα εισόδου, ενώ το μεταγενέστερο, πλήρως συνδεδεμένο επίπεδο μαθαίνει σε ένα πιο αφηρημένο επίπεδο, ενσωματώνοντας καθολικές πληροφορίες από ολόκληρη την εικόνα. Αυτό είναι ένα κοινό μοτίβο στα συνελκτικά νευρωνικά δίκτυα. Εκπαιδεύοντας ένα τέτοιο δίκτυο πετυχαίνουμε 98,78% ακρίβεια που αποτελεί σημαντική βελτίωση σε σχέση με τα προηγούμενα αποτελέσματα.

Ας δοκιμάσουμε να εισαγάγουμε ένα δεύτερο συνελκτικό επίπεδο και επίπεδο ομαδοποίησης αναμέσα στο υπάρχον συνελκτικό επίπεδο – επίπεδο ομαδοποίησης και στο πλήρως συνδεδεμένο κρυφό επίπεδο. Χρησιμοποιούμε ξανά τοπικό πεδίο πρόσληψης 5X5 και κάνουμε ομαδοποίηση σε περιοχές 2X2. Με αυτήν την αλλαγή και παρόμοιες υπερ-παραμέτρους εκπαίδευσης πετυχαίνουμε ακρίβεια 99,06%!

Το πρώτο ερώτημα που προκύπτει είναι τι ακριβώς σημαίνει η προσθήκη και ενός δεύτερου συνελκτικού επιπέδου. Μπορούμε να σκεφτούμε ότι το δεύτερο συνελκτικό επίπεδο έχει ως εισόδους «εικόνες» 12X12, των οποίων τα “pixels” αναπαριστούν την παρουσία (ή μη) συγκεκριμένων τοπικών χαρακτηριστικών στην αρχική εικόνα. Επομένως, αυτό το επίπεδο μπορεί να θεωρηθεί ότι δέχεται ως είσοδο μια εκδοχή της αρχικής εικόνας. Η συγκεκριμένη εκδοχή είναι πιο αφηρημένη και συμπυκνωμένη, αλλά εξακολουθεί να έχει αρκετή χωρική δομή και άρα έχει νόημα να χρησιμοποιήσουμε ένα δεύτερο συνελκτικό επίπεδο.

Η έξοδος από το προηγούμενο επίπεδο περιλαμβάνει 20 χωριστούς χάρτες χαρακτηριστικών και επομένως υπάρχουν 20X12X12 εισόδοι στο δεύτερο συνελκτικό επίπεδο· είναι σαν να έχουμε 20 χωριστές εικόνες ως είσοδο και όχι μία, όπως στο πρώτο συνελκτικό επίπεδο. Πώς θα πρέπει να αποκριθούν οι νευρώνες στο δεύτερο συνελκτικό επίπεδο σε αυτές τις πολλαπλές εικόνες εισόδου; Στην πραγματικότητα θα επιτρέψουμε σε κάθε νευρώνα σε αυτό το επίπεδο να μάθει από *όλους τους* 20x5x5 νευρώνες εισόδου στο τοπικό του πεδίο πρόσληψης. Οι ανιχνευτές χαρακτηριστικών στο δεύτερο συνελκτικό επίπεδο έχουν πρόσβαση σε *όλα τα* χαρακτηριστικά από το προηγούμενο, αλλά μόνο εντός του συγκεκριμένου τοπικού τους πεδίου πρόσληψης. Το ζήτημα θα μπορούσε να είχε προκύψει και στο πρώτο επίπεδο εάν οι εικόνες εισόδου ήταν έγχρωμες. Σε αυτήν την περίπτωση θα είχαμε 3 δυνατότητες εισόδου για κάθε εικονοστοιχείο, που αντιστοιχούν σε κόκκινα, πράσινα και μπλε κανάλια στην εικόνα εισόδου. Έτσι θα επιτρέπαμε στους ανιχνευτές χαρακτηριστικών να έχουν πρόσβαση σε όλες τις πληροφορίες χρώματος, αλλά μόνο μέσα σε ένα δεδομένο τοπικό πεδίο πρόσληψης.

7.2.1 Χρήση ανορθωμένων γραμμικών μονάδων (ReLU)

Ας αλλάξουμε τους νευρώνες ώστε να χρησιμοποιούν συνάρτηση ενεργοποίησης ReLU, δηλαδή $f(z) = \max(0, z)$. Θα εκπαιδεύσουμε για 60 εποχές, με παράμετρο μάθησης $\eta = 0.03$. Επίσης, θα εφαρμόσουμε L2 ομαλοποίηση με παράμετρο ομαλοποίησης $\lambda = 0.1$. Αυτό πετυχαίνει ακρίβεια ταξινόμησης 99,23%. Είναι μια μέτρια βελτίωση σε σχέση με τα σιγμοειδή αποτελέσματα (99,06%). Ωστόσο, σε όλα τα πειράματα διαπιστώθηκε ότι τα δίκτυα που βασίζονται σε ανορθωμένες γραμμικές μονάδες υπερτερούν σταθερά των δικτύων που βασίζονται σε σιγμοειδείς συναρτήσεις ενεργοποίησης. Φαίνεται να υπάρχει ένα πραγματικό κέρδος στη μετάβαση σε ανορθωμένες γραμμικές μονάδες γι’ αυτό το πρόβλημα.

Τι κάνει τη ReLU καλύτερη από τις συναρτήσεις sigmoid ή tanh; Επί του παρόντος έχουμε μικρή κατανόηση της απάντησης σε αυτήν την ερώτηση. Πράγματι οι ανορθωμένες γραμμικές μονάδες έχουν αρχίσει να χρησιμοποιούνται ευρέως μόνο τα τελευταία χρόνια. Ο λόγος αυτής της πρόσφατης υιοθέτησης είναι εμπειρικός: μερικοί δοκίμασαν ανορθωμένες γραμμικές μονάδες, συχνά με βάση ευρετικά επιχειρήματα: μια κοινή δικαιολογία είναι ότι το $\max(0,z)$ δεν υπόκειται σε κορεσμό στο όριο ενός μεγάλου z , σε αντίθεση με τους σιγμοειδείς νευρώνες, και αυτό βοηθά τις ανορθωμένες γραμμικές μονάδες να συνεχίσουν να μαθαίνουν. Πήραν καλά αποτελέσματα ταξινομώντας τα σύνολα δεδομένων αναφοράς και η πρακτική έχει εξαπλωθεί. Σε έναν ιδανικό κόσμο θα είχαμε μια θεωρία που θα μας έλεγε ποια συνάρτηση ενεργοποίησης να επιλέξουμε για ποια εφαρμογή. Αλλά προς το παρόν είμαστε πολύ μακριά από έναν τέτοιο κόσμο. Δεν θα πρέπει καθόλου να εκπλαγούμε αν μπορούν να επιτευχθούν περαιτέρω σημαντικές βελτιώσεις μέσω μιας ακόμη καλύτερης επιλογής της συνάρτησης ενεργοποίησης. Μέχρι στιγμής βασιζόμαστε σε εμπειρικούς κανόνες και δοκιμές ανάλογα με το πρόβλημα.

7.2.2 Επέκταση του συνόλου εκπαίδευσης

Ένας άλλος τρόπος με τον οποίο μπορούμε να ελπίζουμε να βελτιώσουμε τα αποτελέσματά μας είναι με την αλγοριθμική επέκταση των δεδομένων εκπαίδευσης. Ένας απλός τρόπος επέκτασης των δεδομένων είναι να μετατοπιστεί κάθε εικόνα κατά ένα μόνο εικονοστοιχείο, είτε προς τα πάνω, είτε προς τα κάτω, είτε προς τα αριστερά ή προς τα δεξιά, κατά ένα εικονοστοιχείο. Εφαρμόζοντας στο σύνολο MNIST των 50.000 εικόνων παίρνουμε 250.000 εικόνες εκπαίδευσης. Μπορούμε να χρησιμοποιήσουμε το ίδιο δίκτυο με πριν, με ανορθωμένες γραμμικές μονάδες. Αυτή η τετριμμένη επέκταση μάς δίνει σημαντική βελτίωση στην ακρίβεια ταξινόμησης, 99,37%. Στο Simard, Steinkraus and Platt, 2003 αναφέρεται ακρίβεια 99,69% με ένα παρόμοιο δίκτυο, όπου το σύνολο εκπαίδευσης επεκτάθηκε χρησιμοποιώντας επίσης περιστροφή, μετάφραση, λοξότητα και διατάραξη των εικόνων.

7.2.3 Εισαγωγή επιπλέον πλήρως συνδεδεμένου επιπέδου

Ας προσπαθήσουμε να εισαγάγουμε ένα επιπλέον πλήρως συνδεδεμένο επίπεδο, έτσι ώστε να έχουμε δύο πλήρως συνδεδεμένα επίπεδα νευρώνα 100 κρυμμένων νευρώνων. Με αυτόν τον τρόπο πετυχαίνουμε ακρίβεια 99,43%. Αυξάνοντας τους νευρώνες στα κρυφά επίπεδα σε 300 και 1.000 παίρνουμε αποτελέσματα 99,48% και 99,47% αντίστοιχα, τα οποία είναι ενθαρρυντικά, αλλά δεν δικαιολογούν σημαντικό όφελος.

Τι συμβαίνει ακριβώς; Μήπως τα εκτεταμένα ή επιπλέον κρυφά επίπεδα δεν βοηθούν με το MNIST; Ή μήπως το δίκτυο έχει τη δυνατότητα να βελτιωθεί, αλλά μαθαίνει με λάθος τρόπο; Π.χ. μπορεί η χρήση ισχυρότερων τεχνικών ομαλοποίησης να μειώσει την τάση για υπερεκπαίδευση. Μια πιθανότητα είναι να χρησιμοποιήσουμε dropout, όπως συζητήθηκε στο προηγούμενο κεφάλαιο. Η βασική ιδέα του dropout είναι να αφαιρεθούν τυχαία ορισμένες ενεργοποιήσεις κατά την εκπαίδευση, ώστε το μοντέλο να γίνει πιο εύρωστο στην απώλεια συγκεκριμένων τεκμηρίων και επομένως λιγότερο εξαρτώμενο από τις συγκεκριμένες ιδιοσυγκρασίες του συνόλου εκπαίδευσης. Με χρήση dropout στα πλήρως συνδεδεμένα επίπεδα πετυχαίνουμε ακρίβεια 99,60%, που αποτελεί σημαντική βελτίωση σε σχέση με τα προηγούμενα, ιδίως μάλιστα με το βασικό μας δίκτυο αναφοράς των 100 κρυφών νευρώνων με ακρίβεια 99,37%.

Δύο αλλαγές λοιπόν αξίζει να σημειωθούν: πρώτον μειώθηκε ο αριθμός των εποχών εκπαίδευσης σε 40. Το dropout περιόρισε την υπερεκπαίδευση και έτσι το δίκτυο έμαθε ταχύτερα. Δεύτερον τα πλήρως συνδεδεμένα κρυφά επίπεδα έχουν 1.000 νευρώνες και όχι 100. Το dropout ουσιαστικά παραλείπει πολλούς νευρώνες κατά την εκπαίδευση, επομένως κάποια επέκταση είναι αναμενόμενη. Δοκιμή με 300 νευρώνες είχε ελαφρώς χειρότερη επίδοση επικύρωσης σε σχέση με τους 1.000.

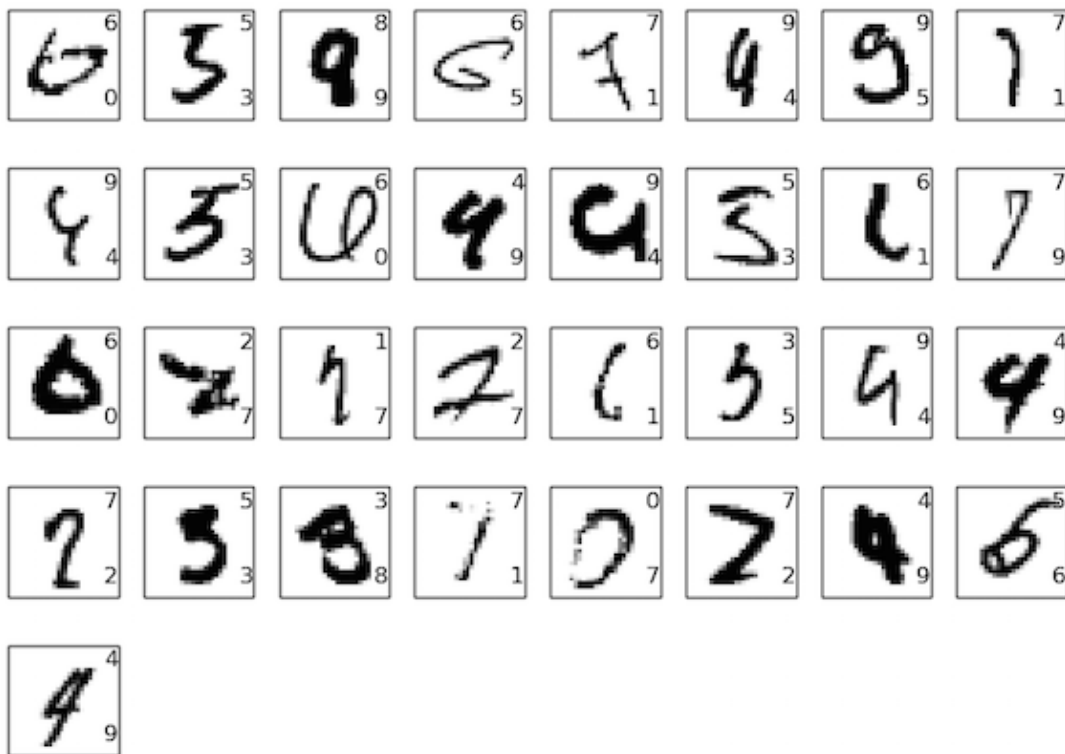
Θα μπορούσαμε να εφαρμόσουμε dropout και στα συνελκτικά επίπεδα, αλλά στην πραγματικότητα δεν υπάρχει λόγος: τα συνελκτικά επίπεδα έχουν σημαντική ενσωματωμένη αντίσταση στην υπερεκπαίδευση. Ο λόγος είναι ότι τα κοινά βάρη σημαίνουν ότι τα συνελκτικά φίλτρα αναγκάζονται να μάθουν από ολόκληρη την εικόνα. Αυτό καθιστά λιγότερο πιθανό να επηρεαστούν από τις τοπικές ιδιοσυγκρασίες στα δεδομένα εκπαίδευσης. Έτσι υπάρχει λιγότερη ανάγκη να εφαρμοστούν άλλες ομαλοποιήσεις, όπως το dropout.

7.2.4 Χρησιμοποιώντας ένα σύνολο δικτύων (ensemble)

Ένας εύκολος τρόπος για να βελτιώσουμε ακόμη περισσότερο την απόδοση είναι να δημιουργήσουμε πολλά νευρωνικά δίκτυα και στη συνέχεια να τα κάνουμε να ψηφίσουν για να καθορίσουν την καλύτερη ταξινόμηση. Ας υποθέσουμε π.χ. ότι εκπαιδεύσαμε 5 διαφορετικά νευρωνικά δίκτυα χρησιμοποιώντας την παραπάνω συνταγή, με το καθένα να επιτυγχάνει ακρίβεια κοντά στο 99.6%. Παρόλο που όλα τα δίκτυα θα είχαν παρόμοιες ακρίβειες, θα μπορούσαν κάλλιστα να κάνουν διαφορετικά λάθη λόγω των διαφορετικών τυχαίων αρχικοποιήσεων. Είναι εύλογο ότι η ψηφοφορία μεταξύ των 5 δικτύων μας μπορεί να αποδώσει μια ταξινόμηση καλύτερη από οποιοδήποτε μεμονωμένο δίκτυο.

Αυτό ακούγεται πολύ καλό για να είναι αληθινό, αλλά το συγκεκριμένο είδος ensembling είναι ένα κοινό τέχνασμα τόσο με τα νευρωνικά δίκτυα όσο και με άλλες τεχνικές μηχανικής μάθησης: στην πραγματικότητα μάλιστα αποδίδει περαιτέρω βελτιώσεις: καταλήγουμε με ακρίβεια 99.67%. Με άλλα λόγια, το σύνολο των δικτύων μας ταξινομεί σωστά όλες, εκτός από 33 από τις 10.000 εικόνες του συνόλου ελέγχου.

Τα υπόλοιπα σφάλματα στο σύνολο ελέγχου εμφανίζονται παρακάτω. Η ετικέτα πάνω δεξιά είναι η σωστή ταξινόμηση, σύμφωνα με τα δεδομένα του MNIST, ενώ κάτω δεξιά είναι η ετικέτα εξόδου από το σύνολο των δικτύων μας:



Σχήμα 7.14 Σφάλματα ταξινόμησης στο σύνολο ελέγχου MNIST.

Αξίζει να τα δούμε λεπτομερώς. Τα δύο πρώτα ψηφία, ένα 6 και ένα 5, είναι γνήσια λάθη από το σύνολό μας. Ωστόσο, είναι επίσης κατανοητά λάθη, το είδος που ένας άνθρωπος θα μπορούσε εύλογα να κάνει. Αυτό το 6 μοιάζει πραγματικά πολύ με ένα 0 και το 5 μοιάζει πολύ με ένα 3. Η τρίτη εικόνα, που υποτίθεται ότι είναι 8, στην πραγματικότητα φαίνεται περισσότερο σαν 9. Έτσι το σύνολο των δικτύων είναι εδώ δικαιολογημένο: φαίνεται να έχει κάνει καλύτερη δουλειά από ό,τι όποιος σχεδίασε αρχικά το ψηφίο. Από την άλλη η τέταρτη εικόνα, η 6, φαίνεται πραγματικά να ταξινομείται άσχημα από τα δίκτυά μας κ.ο.κ.

Στις περισσότερες περιπτώσεις οι επιλογές των δικτύων μας φαίνονται τουλάχιστον αληθοφανείς και σε ορισμένες περιπτώσεις έχουν κάνει καλύτερη δουλειά στην ταξινόμηση από ό,τι το αρχικό άτομο που έγραψε το ψηφίο. Συνολικά τα δίκτυά μας προσφέρουν εξαιρετική απόδοση, ειδικά αν ληφθεί υπόψη ότι έχουν ταξινομήσει σωστά 9.967 εικόνες που δεν εμφανίζονται. Σε αυτό το πλαίσιο τα λίγα σαφή σφάλματα εδώ φαίνονται αρκετά κατανοητά. Ακόμα και ένας προσεκτικός άνθρωπος κάνει περιστασιακά λάθος. Αναμένεται

λοιπόν ότι μόνο ένα εξαιρετικά προσεκτικό και μεθοδικό άτομο θα τα πήγαινε πολύ καλύτερα. Το δίκτυό μας πλησιάζει την ανθρώπινη απόδοση.

Άσκηση 7.2.1. Χρήση της συνάρτησης ενεργοποίησης tanh

Δοκιμάστε να εκπαιδεύσετε το δίκτυο με ενεργοποιήσεις tanh στα συνελκτικά και πλήρως συνδεδεμένα επίπεδα. Ξεκινήστε με τις ίδιες υπερπαραμέτρους όπως στο σιγμοειδές δίκτυο, αλλά εκπαιδεύστε για 20 εποχές αντί για 60. Πόσο καλά αποδίδει το δίκτυο; Τι συμβαίνει αν συνεχίσετε για 60 εποχές; Δώστε τις γραφικές παραστάσεις της ακρίβειας επικύρωσης ανά εποχή και για τις δύο περιπτώσεις (σιγμοειδής, tanh). Θα διαπιστώσετε ότι τα δίκτυα tanh εκπαιδεύονται λίγο πιο γρήγορα, αλλά η τελική ακρίβεια είναι παρόμοια. Εξηγήστε γιατί το tanh δίκτυο μπορεί να εκπαιδεύεται ταχύτερα. Προσπαθήστε να πετύχετε παρόμοια ταχύτητα εκπαίδευσης για τη σιγμοειδή, ενδεχομένως αλλάζοντας τον ρυθμό μάθησης ή κάνοντας κάποια κλιμάκωση (θυμηθείτε ότι $\sigma(z) = (1 + \tanh(z/2))/2$). Δοκιμάστε μερικές επαναλήψεις με διαφορετικές υπερπαραμέτρους ή αρχιτεκτονικές, αναζητώντας τρόπους που η tanh μπορεί να υπερτερεί.

Άσκηση 7.2.2. Επίδραση της επέκτασης των δεδομένων εκπαίδευσης

Η ιδέα των συνελκτικών επιπέδων είναι να συμπεριφέρονται με αναλλοίωτο τρόπο στις εικόνες. Μπορεί να φαίνεται εκπληκτικό λοιπόν ότι το δίκτυό μας μπορεί να μάθει περισσότερα όταν το μόνο που έχουμε κάνει είναι να μεταφράσουμε τα δεδομένα εισόδου. Μπορείτε να εξηγήσετε γιατί αυτό είναι αρκετά λογικό;

7.3 Άλλες προσεγγίσεις στα βαθιά νευρωνικά δίκτυα

Η ταξινόμηση ψηφίων MNIST είναι ένα σημαντικό πρόβλημα που μας βοηθά να κατανοήσουμε πολλές ισχυρές ιδέες, όπως η στοχαστική κατάβαση κλίσης, η πίσω διάδοση, τα συνελκτικά δίκτυα, η ομαλοποίηση κ.ά. Ωστόσο, αποτελεί περιορισμένο πρόβλημα: στη βιβλιογραφία των νευρωνικών δικτύων θα συναντήσετε πολλές ιδέες που δεν έχουμε συζητήσει: επαναλαμβανόμενα νευρωνικά δίκτυα, μηχανές Boltzmann, γενετικά μοντέλα, μάθηση μεταφοράς, ενισχυτική μάθηση και ούτω καθεξής. Τα νευρωνικά δίκτυα είναι ένα τεράστιο πεδίο. Ωστόσο, πολλές σημαντικές ιδέες είναι παραλλαγές σε ιδέες που έχουμε ήδη συζητήσει και μπορούν να γίνουν κατανοητές με λίγη προσπάθεια. Στην παρούσα ενότητα παρέχεται μια ματιά σε αυτές τις αθέατες ακόμη απόψεις. Η συζήτηση δεν είναι λεπτομερής, ούτε περιεκτική. Αντίθετα, είναι μια προσπάθεια να αναδειχθεί ο εννοιολογικός πλούτος του πεδίου και να γίνει συσχέτιση με μερικά από όσα έχουμε ήδη δει.

7.3.1 Επαναλαμβανόμενα νευρωνικά δίκτυα (RNNs)

Στα εμπρός τροφοδότησης δίκτυα που χρησιμοποιούμε υπάρχει μια ενιαία είσοδος που καθορίζει πλήρως τις ενεργοποιήσεις όλων των νευρώνων μέσω των υπόλοιπων επιπέδων. Είναι μια πολύ στατική εικόνα: τα πάντα στο δίκτυο είναι σταθερά, με μια παγωμένη, κρυσταλλική ποιότητα. Αλλά ας υποθέσουμε ότι επιτρέπουμε στα στοιχεία του δικτύου να συνεχίσουν να αλλάζουν με δυναμικό τρόπο. Π.χ. η συμπεριφορά των κρυφών νευρώνων μπορεί να μην καθορίζεται μόνο από τις ενεργοποιήσεις σε προηγούμενα κρυφά επίπεδα, αλλά και από τις ενεργοποιήσεις σε παλαιότερες χρονικές στιγμές. Πράγματι η ενεργοποίηση ενός νευρώνα μπορεί να προσδιοριστεί εν μέρει από τη δική του ενεργοποίηση σε προγενέστερο χρόνο. Αυτό σίγουρα δεν συμβαίνει σε ένα δίκτυο εμπρός τροφοδότησης. Ή ίσως οι ενεργοποιήσεις των κρυφών και εξερχόμενων νευρώνων δεν θα καθοριστούν μόνο από την τρέχουσα είσοδο στο δίκτυο, αλλά και από προηγούμενες εισόδους.

Τα νευρωνικά δίκτυα με αυτό το είδος χρονικά μεταβαλλόμενης συμπεριφοράς είναι γνωστά ως επαναλαμβανόμενα νευρωνικά δίκτυα ή RNNs. Υπάρχουν πολλοί διαφορετικοί τρόποι μαθηματικής τυποποίησης της άτυπης περιγραφής των επαναλαμβανόμενων δικτύων που δίνεται στην προηγούμενη παράγραφο. Μπορείτε να πάρετε τη γεύση ορισμένων από αυτά τα μαθηματικά μοντέλα ρίχνοντας μια ματιά στο [άρθρο της Wikipedia για τα RNNs](#). Αυτήν τη στιγμή, αυτή η σελίδα απαριθμεί όχι λιγότερα από 13 διαφορετικά μοντέλα. Αλλά πέρα από τις μαθηματικές λεπτομέρειες η γενική ιδέα είναι ότι τα RNNs είναι νευρωνικά δίκτυα στα οποία υπάρχει κάποια έννοια δυναμικής αλλαγής με την πάροδο του χρόνου. Δεν αποτελεί έκπληξη το γεγονός ότι είναι ιδιαίτερα χρήσιμα στην ανάλυση δεδομένων ή διαδικασιών που αλλάζουν με την πάροδο του χρόνου. Τέτοια δεδομένα και διαδικασίες προκύπτουν φυσικά σε προβλήματα όπως η ομιλία ή η φυσική γλώσσα.

Ένας τρόπος με τον οποίο χρησιμοποιούνται σήμερα τα RNNs είναι να συνδέουν τα νευρωνικά δίκτυα πιο στενά με παραδοσιακούς τρόπους σκέψης σχετικά με τους αλγόριθμους, τους τρόπους σκέψης που βασίζονται σε έννοιες όπως οι μηχανές Turing και οι (συμβατικές) γλώσσες προγραμματισμού. [Μια εργασία του 2014](#) ανέπτυξε ένα RNN που θα μπορούσε να λάβει ως είσοδο μια περιγραφή ανά χαρακτήρα ενός (πολύ, πολύ απλού) Python προγράμματος, και την χρησιμοποίησε για να προβλέψει την έξοδο. Ουσιαστικά το δίκτυο μαθαίνει να «κατανοεί» ορισμένα προγράμματα Python. [Μια δεύτερη εργασία, επίσης από το 2014](#), χρησιμοποίησε τα RNNs ως σημείο εκκίνησης για να αναπτύξει αυτό που ονόμασαν νευρωνική μηχανή Turing (NTM). Πρόκειται για έναν καθολικό υπολογιστή του οποίου ολόκληρη η δομή μπορεί να εκπαιδευτεί χρησιμοποιώντας κατάβαση κλίσης. Εκπαίδευσαν το NTM τους να συμπεραίνει αλγόριθμους για πολλά απλά προβλήματα, όπως η ταξινόμηση και η αντιγραφή.

Προς το παρόν αυτά είναι εξαιρετικά απλά μοντέλα. Το να μάθει ένα δίκτυο να εκτελεί το πρόγραμμα Python print(398345+42598) δεν το καθιστά έναν πλήρη διερμηνέα Python! Δεν είναι σαφές πόσο πιο μακριά θα είναι δυνατόν να προωθηθούν οι ιδέες αυτές. Ωστόσο, τα αποτελέσματα είναι ενδιαφέροντα. Ιστορικά τα νευρωνικά δίκτυα τα έχουν πάει καλά σε προβλήματα αναγνώρισης προτύπων όπου οι συμβατικές αλγοριθμικές προσεγγίσεις έχουν πρόβλημα. Αντίστροφα, οι συμβατικές αλγοριθμικές προσεγγίσεις είναι καλές στην επίλυση προβλημάτων στα οποία τα νευρωνικά δίκτυα δεν είναι τόσο καλά. Κανείς σήμερα δεν υλοποιεί έναν διακομιστή ιστού ή ένα πρόγραμμα βάσης δεδομένων χρησιμοποιώντας ένα νευρωνικό δίκτυο! Θα ήταν χρήσιμο να αναπτυχθούν ενοποιημένα μοντέλα που ενσωματώνουν τα πλεονεκτήματα τόσο των νευρωνικών δικτύων, όσο και των πιο παραδοσιακών προσεγγίσεων στους αλγόριθμους. Τα RNNs και οι ιδέες που εμπνέονται από τα RNNs μπορεί να βοηθήσουν προς αυτήν την κατεύθυνση.

Τα RNNs έχουν επίσης χρησιμοποιηθεί τα τελευταία χρόνια για να αντιμετωπίσουν πολλά άλλα προβλήματα. Είναι ιδιαίτερα χρήσιμα στην αναγνώριση ομιλίας. Οι προσεγγίσεις που βασίζονται σε RNN έχουν π.χ. [θέσει ρεκόρ για την ακρίβεια της αναγνώρισης φωνημάτων](#). Έχουν επίσης χρησιμοποιηθεί για την ανάπτυξη [βελτιωμένων μοντέλων της γλώσσας που χρησιμοποιούν οι άνθρωποι ενώ μιλούν](#). Τα καλύτερα γλωσσικά μοντέλα βοηθούν στην αποσαφήνιση των εκφράσεων που ακούγονται όμοιες. Ένα καλό γλωσσικό μοντέλο π.χ. θα μας πει ότι η αγγλική φράση “to infinity and beyond” είναι πολύ πιο πιθανή από την “two infinity and beyond”, παρά το γεγονός ότι οι φράσεις είναι ομόηχες. Τα RNNs έχουν χρησιμοποιηθεί για τον καθορισμό νέων ρεκόρ για ορισμένα γλωσσικά προβλήματα αναφοράς.

Η εργασία αυτή είναι παρεμπιπτόντως μέρος μιας ευρύτερης χρήσης βαθέων νευρωνικών δικτύων όλων των τύπων, όχι μόνο RNNs, στην αναγνώριση ομιλίας. Π.χ. μια προσέγγιση που βασίζεται σε βαθιά δίκτυα έχει επιτύχει [εξαιρετικά αποτελέσματα στη συνεχή αναγνώριση ομιλίας μεγάλου λεξιλογίου](#). Ένα άλλο σύστημα που βασίζεται σε βαθιά δίκτυα έχει αναπτυχθεί στο [λειτουργικό σύστημα Android της Google](#).

Πολλές από τις ιδέες που χρησιμοποιούνται στα δίκτυα εμπρός τροφοδότησης μπορούν επίσης να χρησιμοποιηθούν σε RNNs. Συγκεκριμένα μπορούμε να εκπαιδύσουμε RNNs χρησιμοποιώντας απλές τροποποιήσεις στην κατάβαση κλίσης και την πίσω διάδοση. Πολλές άλλες ιδέες που χρησιμοποιούνται στα εμπρόσθια δίκτυα, από τις τεχνικές ομαλοποίησης και τις συνελίξεις έως τις συναρτήσεις ενεργοποίησης και κόστους που χρησιμοποιούνται, είναι επιπλέον χρήσιμες σε επαναλαμβανόμενα δίκτυα.

7.3.2 Μονάδες μακράς βραχυπρόθεσμης μνήμης (LSTMs)

Μια πρόκληση που επηρεάζει τα RNN είναι ότι τα πρώτα μοντέλα αποδείχθηκε πολύ δύσκολο να εκπαιδευτούν, πιο δύσκολο ακόμη και από τα βαθιά δίκτυα εμπρόσθιας τροφοδότησης. Ο λόγος είναι το πρόβλημα της ασταθούς κλίσης. Η συνήθης εκδήλωση αυτού του προβλήματος είναι ότι η κλίση γίνεται όλο και μικρότερη καθώς διαδίδεται προς τα πίσω μέσω των επιπέδων. Αυτό καθιστά τη μάθηση στα πρώτα επίπεδα εξαιρετικά αργή. Το πρόβλημα στην πραγματικότητα επιδεινώνεται στα RNN, καθώς οι κλίσεις δεν διαδίδονται μόνο προς τα πίσω μέσω των επιπέδων, αλλά διαδίδονται προς τα πίσω και μέσω του χρόνου. Εάν το δίκτυο λειτουργεί για μεγάλο χρονικό διάστημα, μπορεί να κάνει την κλίση εξαιρετικά ασταθή και δύσκολο να αξιοποιηθεί για μάθηση. Ευτυχώς είναι δυνατό να ενσωματωθεί μια ιδέα γνωστή ως μονάδες μακράς βραχυπρόθεσμης μνήμης (LSTMs) σε RNNs. Οι μονάδες εισήχθησαν από τους [Hochreiter και Schmidhuber το 1997](#) με ρητό σκοπό να βοηθήσουν στην αντιμετώπιση του προβλήματος της ασταθούς κλίσης. Τα LSTMs καθιστούν πολύ πιο εύκολο να υπάρχουν καλά αποτελέσματα κατά την εκπαίδευση των RNN και πολλές πρόσφατες εργασίες (συμπεριλαμβανομένων πολλών που αναφέρθηκαν παραπάνω) χρησιμοποιούν LSTMs ή σχετικές ιδέες.

7.3.3 Βαθιά δίκτυα πεποιθήσεων, γενετικά μοντέλα και μηχανές Boltzmann

Το σύγχρονο ενδιαφέρον για τη βαθιά μάθηση ξεκίνησε το 2006, με εργασίες που εξηγούν πώς να εκπαιδευτεί ένας τύπος νευρωνικού δικτύου γνωστός ως *βαθύ δίκτυο πεποιθήσεων* (Deep Belief Net, DBN) (Hinton, Osindero & Teh, 2006 και Hinton & Salakhutdinov, 2006). Τα DBN είχαν επιρροή για αρκετά χρόνια, αλλά έκτοτε έχει μειωθεί η δημοτικότητά τους, ενώ μοντέλα όπως τα δίκτυα εμπρός τροφοδότησης και τα επαναλαμβανόμενα νευρωνικά δίκτυα έχουν γίνει πιο επίκαιρα. Παρ' όλα αυτά, τα DBN έχουν πολλές ιδιότητες που τα καθιστούν ενδιαφέροντα.

Ένας λόγος για τον οποίο τα DBN είναι ενδιαφέροντα είναι ότι αποτελούν ένα παράδειγμα αυτού που ονομάζεται *γενετικό μοντέλο*. Σε ένα δίκτυο εμπρός τροφοδότησης καθορίζουμε τις ενεργοποιήσεις εισόδου και αυτές με τη σειρά τους καθορίζουν τις ενεργοποιήσεις των νευρώνων χαρακτηριστικών αργότερα στο δίκτυο. Ένα γενετικό μοντέλο, όπως ένα DBN, μπορεί να χρησιμοποιηθεί με παρόμοιο τρόπο, αλλά είναι επίσης δυνατό να καθοριστούν οι τιμές ορισμένων από τους νευρώνες χαρακτηριστικών και στη συνέχεια να «εκτελέσετε το δίκτυο προς τα πίσω», δημιουργώντας τιμές για τις ενεργοποιήσεις εισόδου. Πιο συγκεκριμένα ένα DBN εκπαιδευμένο σε εικόνες χειρόγραφων ψηφίων μπορεί (δυστυχώς και με κάποια προσοχή) να χρησιμοποιηθεί επίσης για τη δημιουργία εικόνων που μοιάζουν με χειρόγραφα ψηφία: με άλλα λόγια το DBN θα μάθαινε κατά κάποιον τρόπο να γράφει. Ως προς αυτό ένα γενετικό μοντέλο μοιάζει πολύ με τον ανθρώπινο εγκέφαλο: όχι μόνο μπορεί να διαβάσει ψηφία, αλλά μπορεί επίσης να τα γράψει. Σύμφωνα με τον Geoffrey Hinton, [για να αναγνωρίσετε σχήματα, μάθετε πρώτα να δημιουργείτε εικόνες](#).

Ένας δεύτερος λόγος για τον οποίο τα DBN είναι ενδιαφέροντα είναι ότι μπορούν να κάνουν μη επιβλεπόμενη και ημι-επιβλεπόμενη μάθηση. Π.χ. όταν εκπαιδούνται με δεδομένα εικόνες, τα DBN μπορούν να μάθουν χρήσιμες λειτουργίες για την κατανόηση άλλων εικόνων, ακόμη και αν οι εικόνες εκπαίδευσης δεν έχουν ετικέτες. Η ικανότητα να κάνουμε μη επιβλεπόμενη μάθηση είναι εξαιρετικά ενδιαφέροντα τόσο για θεμελιώδεις επιστημονικούς λόγους, όσο και —αν μπορεί να λειτουργήσει αρκετά καλά— για πρακτικές εφαρμογές.

Δεδομένων αυτών των ελκυστικών χαρακτηριστικών, γιατί τα DBN έχουν χάσει σε δημοτικότητα ως μοντέλα για βαθιά μάθηση; Εν μέρει οφείλεται στο ότι μοντέλα, όπως τα εμπρόσθια τροφοδότησης και τα επαναλαμβανόμενα δίκτυα, έχουν επιτύχει πολλά θεαματικά αποτελέσματα, π.χ. οι επιτυχίες στην αναγνώριση εικόνων και ομιλίας. Δεν προκαλεί έκπληξη —και εύλογα— το γεγονός ότι τώρα δίνεται μεγάλη προσοχή σε αυτά τα μοντέλα. Υπάρχει όμως ένα ατυχές επακόλουθο. Η αγορά των ιδεών συχνά λειτουργεί με τρόπο ώστε ο νικητής να τα παίρνει όλα: σχεδόν όλη η προσοχή πηγαίνει στην τρέχουσα μόδα της στιγμής σε κάθε δεδομένο τομέα. Μπορεί να γίνει εξαιρετικά δύσκολο για τους ανθρώπους να εργαστούν σε —κατά ώρας— ανεπίκαιρες ιδέες, ακόμη και όταν αυτές έχουν προφανώς πραγματικό μακροπρόθεσμο ενδιαφέρον. Τα DBN και άλλα γενετικά μοντέλα πιθανότατα αξίζουν περισσότερη προσοχή από ό,τι λαμβάνουν επί του παρόντος. Δεν θα αποτελέσει έκπληξη αν τα DBN ή ένα σχετικό μοντέλο μια μέρα ξεπεράσουν τα σημερινά μοντέρνα μοντέλα. Για μια εισαγωγή στα DBN ανατρέξτε [σε αυτήν την επισκόπηση](#) και [αυτό το άρθρο](#). Δεν αφορά καθαυτά τα βαθιά δίκτυα πεποιθήσεων, αλλά περιέχει πολλές χρήσιμες πληροφορίες σχετικά με τις περιορισμένες μηχανές Boltzmann, οι οποίες αποτελούν βασικό συστατικό των DBN.

7.3.4 Άλλες ιδέες

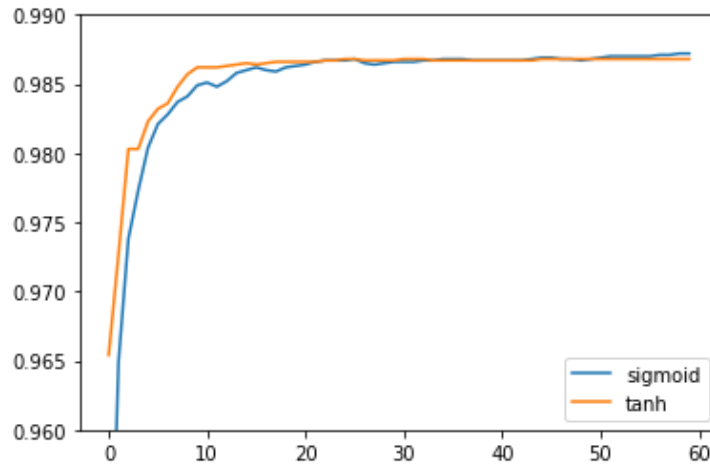
Τι άλλο συμβαίνει στα νευρωνικά δίκτυα και τη βαθιά μάθηση; Υπάρχει ένας τεράστιος αριθμός άλλων συναρπαστικών εργασιών. Οι ενεργοί τομείς έρευνας περιλαμβάνουν τη χρήση νευρωνικών δικτύων για την [επεξεργασία φυσικής γλώσσας](#), [τη μηχανική μετάφραση](#), καθώς και ίσως πιο εκπληκτικές εφαρμογές, όπως η [μουσική πληροφορική](#). Υπάρχουν φυσικά και πολλοί άλλοι τομείς. Σε πολλές περιπτώσεις, έχοντας διαβάσει αυτό το βιβλίο θα πρέπει να είστε σε θέση να αρχίσετε να ακολουθείτε πρόσφατες εργασίες, αν και (φυσικά) θα πρέπει να συμπληρώσετε κενά στις προαπαιτούμενες βασικές γνώσεις.

Ολοκληρώνοντας το κεφάλαιο, μπορεί να αναφερθεί μια ιδιαίτερα διασκεδαστική εργασία. Συνδυάζει βαθιά συνελκτικά δίκτυα με μια τεχνική γνωστή ως ενισχυτική μάθηση, προκειμένου να μάθει να [παίζει καλά βιντεοπαιχνίδια](#) (δείτε επίσης [αυτήν τη συνέχεια](#)). Η ιδέα είναι να χρησιμοποιήσετε το συνελκτικό δίκτυο για να απλοποιήσετε τα δεδομένα εικονοστοιχείων από την οθόνη του παιχνιδιού, μετατρέποντάς το σε ένα απλούστερο σύνολο χαρακτηριστικών, τα οποία μπορούν να χρησιμοποιηθούν για να αποφασιστεί ποια ενέργεια πρέπει να γίνει: «πηγαίνετε αριστερά», «κατεβείτε», «φωτιά» κ.ο.κ. Αυτό που είναι ιδιαίτερα ενδιαφέρον είναι ότι ένα μόνο δίκτυο έμαθε να παίζει επτά διαφορετικά κλασικά βιντεοπαιχνίδια αρκετά καλά, ξεπερνώντας τους ανθρώπινους ειδικούς σε τρία από τα παιχνίδια.

Όλα αυτά μπορεί να ακούγονται σαν παιχνίδι και δεν υπάρχει αμφιβολία ότι η εργασία ήταν αρκετά ελκυστική έχοντας τον τίτλο «Παίζοντας Atari με ενισχυτική μάθηση». Αλλά κοιτάζοντας πέρα από την επιφανειακή γυαλάδα, σκεφτείτε ότι αυτό το σύστημα λαμβάνει ακατέργαστα δεδομένα rixel —δεν γνωρίζει καν τους κανόνες του παιχνιδιού!— και από τα συγκεκριμένα δεδομένα μαθαίνει να κάνει λήψη αποφάσεων υψηλής ποιότητας σε πολλά πολύ διαφορετικά και υψηλής αντιπαλότητας περιβάλλοντα, το καθένα με το δικό του πολύπλοκο σύνολο κανόνων.

Απαντήσεις ασκήσεων

Άσκηση 7.2.1. Χρήση της συνάρτησης ενεργοποίησης tanh

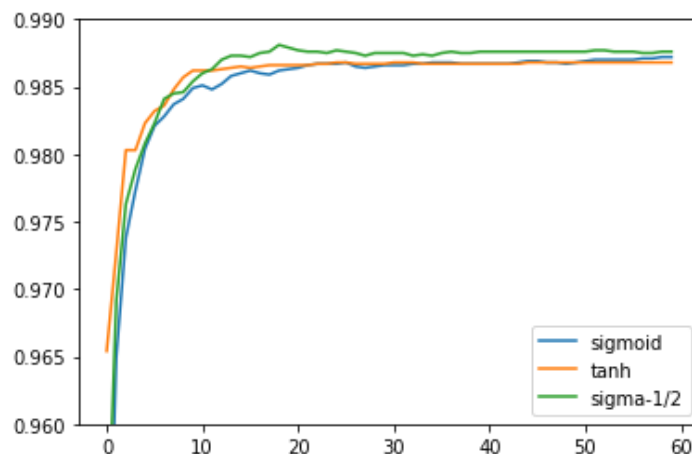


Σχήμα 7.15 Σύγκριση εκπαίδευσης με λογιστική και tanh συνάρτηση ενεργοποίησης.

Όπως φαίνεται στις παραπάνω γραφικές παραστάσεις, πράγματι με τη χρήση υπερβολικής εφαπτομένης το δίκτυο μπορεί να μάθει λίγο πιο γρήγορα στην αρχή, αλλά στο τέλος οι ακρίβειες είναι πολύ κοντά.

Ένας πιθανός λόγος είναι ότι αφού $\tanh'(z) = 4\sigma'(2z)$, όταν το z είναι μικρό, η παράγωγος της tanh είναι πιο κοντά στο 1 από ότι η σ' λόγω του παράγοντα 4. Αλλά όταν το z είναι μεγάλο, εξαιτίας του εκθετικού, το $4\sigma'(2z)$ γίνεται πολύ μικρότερο από το $\sigma'(z)$. Θυμηθείτε επίσης ότι οι μικρές τιμές των παραγώγων ήταν η αιτία του προβλήματος της εξαφανιζόμενης κλίσης (vanishing gradient) που επιβράδυνε τη μάθηση. Επομένως η συνάρτηση tanh φαίνεται καλύτερη για μικρές τιμές του z και χειρότερη για μεγάλες τιμές. Το όφελος από τις μικρές τιμές του z (που μπορεί να είναι αρκετά συχνό, ιδίως αν γίνει κάποια ομαλοποίηση) μπορεί να ξεπεράσει το κόστος από τις μεγάλες τιμές.

Άλλος ένας λόγος είναι ότι η tanh είναι συμμετρική ως προς την αρχή των αξόνων, ενώ η σ είναι πάντα θετική. Αυτό σημαίνει ότι η μεταβολή στα βάρη ενός νευρώνα που δέχεται ως είσοδο εξόδους σιγμοειδών νευρώνων θα έχει το ίδιο πρόσημο: τα βάρη είτε θα αυξάνονται είτε θα μειώνονται όλα μαζί, ανάλογα με την τοπική κλίση του νευρώνα, κάτι που μπορεί να οδηγήσει σε αστάθεια και ταλαντώσεις κατά την κατάβαση κλίσης. Μπορούμε να επιτύχουμε παρόμοιο αποτέλεσμα μετατοπίζοντας τη λογιστική συνάρτηση, χρησιμοποιώντας π.χ. τη $\sigma(z) - 1/2$ ως συνάρτηση ενεργοποίησης. Παρατηρούμε ότι το δίκτυο μαθαίνει ταχύτερα από ό,τι με τη λογιστική συνάρτηση, αλλά πιο αργά σε σχέση με την tanh.



Σχήμα 7.16 Αποτελέσματα μετά από μετατόπιση της λογιστικής συνάρτησης.

Άσκηση 7.2.2. Επίδραση της επέκτασης των δεδομένων εκπαίδευσης

Μετατοπίζοντας κάθε εικόνα εκπαίδευσης κατά ένα μόνο pixel (πάνω, κάτω, αριστερά ή δεξιά), τα συνελκτικά επίπεδα μετατοπίζονται μόνο κατά ένα pixel επίσης. Αλλά τότε το επίπεδο max-pooling μπορεί να είναι αρκετά διαφορετικό, καθώς για καθέναν από τους νευρώνες του, οι 4 ενεργοποιήσεις που λαμβάνονται υπόψη για τον υπολογισμό του μέγιστου θα είναι διαφορετικές (πιο συγκεκριμένα 2 θα παραμείνουν, αλλά 2 θα είναι νέες).

Βιβλιογραφία

- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- Hinton, G.E., & Salakhutdinov, R.R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504-507.
- Hinton, G.E., Osindero, S., & Teh, Y.W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527-1554.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Nielsen, M.A. (2019). *Neural networks and deep learning*. Determination press. <http://neuralnetworksanddeeplearning.com>
- Simard, P.Y., Steinkraus, D., & Platt, J.C. (2003). Best practices for convolutional neural networks applied to visual document analysis. *Icdar*, 3, 2003.
- Zeiler, M.D., & Fergus, R. (2013). Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*.

Κεφάλαιο 8

Εισαγωγή στους Γενετικούς Αλγορίθμους

Σύνοψη

Το κεφάλαιο αυτό αποτελεί ουσιαστικά μια εκτεταμένη εισαγωγή στους Γενετικούς Αλγορίθμους. Επαναλαμβάνοντας κάποιες από τις εισαγωγικές έννοιες του πρώτου κεφαλαίου, συνεχίζει με θέματα που έχουν να κάνουν με την περιγραφή του τρόπου λειτουργίας των Γενετικών Αλγορίθμων και τελειώνει με μια θεωρητική τεκμηρίωση της λειτουργίας τους.

Αναλυτικότερα η πρώτη ενότητα αποτελεί μια γενική εισαγωγή στους Γενετικούς Αλγορίθμους. Αρχικά δίνει κάποια στοιχεία για την ιστορία και τη χρονική εξέλιξή τους, όπως και στοιχεία γενικότερα των μεθόδων που βασίζονται στη βιολογική εξελικτική διαδικασία, καθώς και τη βασική ορολογία που χρησιμοποιείται σε αυτού του είδους τους αλγορίθμους. Ακολουθεί η παρουσίαση των πλεονεκτημάτων των ΓΑ, ενώ στη συνέχεια αναφέρονται δύο προβλήματα που δημιουργούν δυσπιστία, σχετικά με τη χρησιμότητά τους και τα αντίστοιχα αντεπιχειρήματά τους. Η ενότητα τελειώνει με τα χαρακτηριστικά που κάνουν τους Γενετικούς Αλγορίθμους πολύ αποτελεσματικούς στην επίλυση μιας μεγάλης γκάμας προβλημάτων. Στη δεύτερη ενότητα περιγράφεται ο τρόπος λειτουργίας των Γενετικών Αλγορίθμων και αναλύονται οι βασικοί γενετικοί τελεστές, μέσα από ένα παράδειγμα βελτιστοποίησης συνάρτησης πολλών μεταβλητών. Ακολουθεί η αναλυτική παρουσίαση ενός τέτοιου παραδείγματος που καθιστά αντιληπτή τη θεωρία. Στην τρίτη ενότητα παρουσιάζονται ορισμένα βασικά θέματα της θεωρητικής θεμελίωσης των Γενετικών Αλγορίθμων. Το κεφάλαιο τελειώνει με τα συμπεράσματα και την παράθεση της βασικής βιβλιογραφίας. Στο τέλος κάθε ενότητας δίνεται ικανός αριθμός ασκήσεων αυτοαξιολόγησης, οι οποίες έχουν σχεδιαστεί έτσι ώστε αφενός να βοηθούν στην καλύτερη κατανόηση και εμπέδωση της ύλης που παρουσιάστηκε στην ενότητα, αφετέρου να τη συμπληρώνουν.

Προσδοκώμενα Αποτελέσματα

Όταν θα έχετε τελειώσει τη μελέτη αυτού του κεφαλαίου, θα μπορείτε να:

- περιγράψετε τι είναι ένας Γενετικός Αλγόριθμος,
- απαριθμήσετε τα πλεονεκτήματα των Γενετικών Αλγορίθμων,
- περιγράψετε τα κύρια χαρακτηριστικά ενός Γενετικού Αλγορίθμου,
- περιγράψετε τα βασικά δομικά στοιχεία ενός Γενετικού Αλγορίθμου,
- περιγράψετε τους βασικούς γενετικούς τελεστές,
- περιγράψετε τα βήματα κωδικοποίησης ενός προβλήματος βελτιστοποίησης, με σκοπό την επίλυσή του με Γενετικό Αλγόριθμο,
- ορίσετε τα σχήματα ομοιότητας και να διατυπώσετε το θεώρημα των σχημάτων,
- ορίσετε ποσοτικά την επίδραση της επιλογής στην επιβίωση των σχημάτων,
- ορίσετε ποσοτικά την επίδραση των γενετικών τελεστών στην επιβίωση των σχημάτων,
- ορίσετε την υπόθεση δομικών στοιχείων.

Λέξεις-Κλειδιά:

- Γενετικοί Αλγόριθμοι
- Εξελικτικοί Αλγόριθμοι
- Γενετικοί Τελεστές
- Εξέλιξη
- Αρχικοποίηση
- Επιλογή
- Αναπαραγωγή
- Μετάλλαξη
- Χρωμόσωμα
- Αντικειμενική Συνάρτηση
- Βελτιστοποίηση
- Σχήμα
- Θεώρημα σχημάτων

Προαπαιτούμενη γνώση

Κεφάλαιο 1.

8.1 Εισαγωγή

Τα τελευταία τριάντα χρόνια παρατηρείται ένα συνεχώς αυξανόμενο ενδιαφέρον για την ανάπτυξη συστημάτων επίλυσης προβλημάτων, τα οποία βασίζονται στις αρχές της Φυσικής Εξέλιξης. Τα συστήματα αυτού του είδους λειτουργούν διατηρώντας έναν πληθυσμό κωδικοποιημένων πιθανών λύσεων του προβλήματος που προσπαθούμε να επιλύσουμε και εφαρμόζοντας πάνω σε αυτόν διάφορες διαδικασίες εμπνευσμένες από τη βιολογική εξέλιξη. Έτσι, περνώντας από γενιά σε γενιά, τα συστήματα αυτά δημιουργούν συνεχώς νέους πληθυσμούς πιθανών λύσεων, εξελίσσοντάς τους προηγούμενους.

Οι Γενετικοί Αλγόριθμοι (Genetic Algorithms) είναι ένα παράδειγμα τέτοιου συστήματος που μαζί με τον Εξελικτικό Προγραμματισμό (Evolutionary Programming), τις Στρατηγικές Εξέλιξης (Evolution Strategies), τα Συστήματα Ταξινόμησης (Classifier Systems) και τον Γενετικό Προγραμματισμό (Genetic Programming) αποτελούν μια κατηγορία συστημάτων επίλυσης προβλημάτων ευρύτερα γνωστή με τον όρο Εξελικτικοί Αλγόριθμοι (Evolutionary Algorithms).

Η πρώτη εμφάνιση των Γενετικών Αλγόριθμων (ΓΑ) χρονολογείται στις αρχές του 1950, όταν διάφοροι επιστήμονες από τον χώρο της βιολογίας αποφάσισαν να χρησιμοποιήσουν υπολογιστές στην προσπάθειά τους να προσομοιώσουν πολύπλοκα βιολογικά συστήματα. Η συστηματική τους ανάπτυξη όμως, που οδήγησε στη μορφή με την οποία είναι γνωστοί σήμερα, πραγματοποιήθηκε στις αρχές του 1970 από τον John Holland (Holland 1975) και τους συνεργάτες του στο Πανεπιστήμιο του Michigan.

Η βασική ιδέα που κρύβεται πίσω από τους ΓΑ είναι η μίμηση των μηχανισμών της βιολογικής εξέλιξης που απαντώνται στη φύση. Ας πάρουμε π.χ. τους λαγούς και πώς αναπαράγονται και εξελίσσονται από γενιά σε γενιά (Michalewicz, 1996). Έστω ότι αρχίζουμε να παρατηρούμε έναν συγκεκριμένο πληθυσμό από λαγούς σε ένα οικοσύστημα. Όπως είναι φυσικό, κάποιοι από αυτούς θα είναι πιο γρήγοροι και πιο εύστροφοι από άλλους. Αυτοί οι λαγοί έχουν περισσότερες πιθανότητες να επιβιώσουν στο φυσικό τους περιβάλλον (δηλαδή να εξασφαλίζουν τροφή και να ξεφεύγουν από τα διάφορα αρπακτικά που τους καταδιώκουν, όπως π.χ. τις αλεπούδες) από ό,τι κάποιοι πιο αργοί ή λιγότερο έξυπνοι λαγοί. Φυσικά δεν είναι λίγοι οι αργοί ή λιγότερο έξυπνοι λαγοί που καταφέρνουν να επιβιώνουν εξαιτίας της τύχης ή άλλων παραγόντων. Όλοι αυτοί οι λαγοί, που καταφέρνουν να επιβιώσουν, θα αρχίσουν την παραγωγή της επόμενης γενιάς τους, μιας γενιάς που θα συνδυάζει με διάφορους τρόπους όλα τα χαρακτηριστικά των μελών της προηγούμενης. Έτσι μερικοί αργοί λαγοί θα αναμειχθούν με κάποιους γρήγορους, κάποιοι γρήγοροι με άλλους γρήγορους, κάποιοι εύστροφοι λαγοί με κάποιους μη εύστροφους και ούτω καθεξής, δημιουργώντας σταδιακά έναν πληθυσμό που απαρτίζεται από λαγούς που κατά μέσο όρο είναι έξυπνότεροι και ταχύτεροι από τους προγόνους τους. Ευτυχώς όμως, για τη διατήρηση της φυσικής ισορροπίας, και τα αρπακτικά υφίστανται την ίδια διαδικασία εξέλιξης από γενιά σε γενιά, διαφορετικά οι λαγοί θα γινόντουσαν υπερβολικά γρήγοροι και έξυπνοι για να μπορούν να τους πιάσουν.

Οι ΓΑ χρησιμοποιούν ορολογία δανεισμένη από τον χώρο της Φυσικής Γενετικής. Αναφέρονται σε άτομα (individuals) ή γενότυπους (genotypes) μέσα σε έναν πληθυσμό. Κάθε άτομο ή γενότυπος αποτελείται από χρωμοσώματα (chromosomes). Στους ΓΑ αναφερόμαστε συνήθως σε άτομα με ένα μόνο χρωμόσωμα. Τα χρωμοσώματα αποτελούνται από γονίδια (genes) που είναι διατεταγμένα σε γραμμική ακολουθία. Κάθε γονίδιο επηρεάζει την κληρονομικότητα ενός ή περισσότερων χαρακτηριστικών. Τα γονίδια που επηρεάζουν συγκεκριμένα χαρακτηριστικά γνωρίσματα του ατόμου βρίσκονται σε συγκεκριμένες θέσεις του χρωματοσώματος, οι οποίες καλούνται loci. Κάθε χαρακτηριστικό γνώρισμα του ατόμου (όπως για παράδειγμα το χρώμα μαλλιών) έχει τη δυνατότητα να εμφανιστεί με διάφορες μορφές, ανάλογα με την κατάσταση στην οποία βρίσκεται το αντίστοιχο γονίδιο που το επηρεάζει. Οι διαφορετικές αυτές καταστάσεις που μπορεί να πάρει το γονίδιο καλούνται alleles (τιμές χαρακτηριστικού γνωρίσματος).

Κάθε γενότυπος αναπαριστά μια πιθανή λύση σε ένα πρόβλημα. Το «αποκωδικοποιημένο» περιεχόμενο ενός συγκεκριμένου χρωμοσώματος καλείται φαινότυπος (phenotype) (π.χ. ένας ζωντανός οργανισμός είναι ο φαινότυπος των χρωμοσωμάτων του). Μια διαδικασία εξέλιξης που εφαρμόζεται πάνω σε έναν πληθυσμό αντιστοιχεί σε ένα εκτενές ψάξιμο στον χώρο των πιθανών λύσεων. Απαραίτητη προϋπόθεση για την επιτυχημένη έκβαση ενός τέτοιου ψαξίματος αποτελεί η εξισορρόπηση δύο διαδικασιών που είναι προφανώς αντικρουόμενες, της εκμετάλλευσης και διατήρησης των καλύτερων λύσεων, και της όσο το δυνατόν καλύτερης εξερεύνησης όλου του διαστήματος.

Οι ΓΑ διατηρούν έναν πληθυσμό πιθανών λύσεων, του προβλήματος που μας ενδιαφέρει, πάνω στον οποίο δουλεύουν, σε αντίθεση με άλλες μεθόδους αναζήτησης που επεξεργάζονται ένα μόνο σημείο του διαστήματος αναζήτησης. Έτσι ένας ΓΑ πραγματοποιεί αναζήτηση σε πολλές κατευθύνσεις και υποστηρίζει καταγραφή και ανταλλαγή πληροφοριών μεταξύ αυτών. Ο πληθυσμός υφίσταται μια προσομοιωμένη γενετική εξέλιξη. Σε κάθε γενιά οι σχετικά «καλές» λύσεις αναπαράγονται, ενώ οι σχετικά «κακές» απομακρύνονται. Ο

διαχωρισμός και η αποτίμηση των διαφόρων λύσεων γίνεται με τη βοήθεια μιας αντικειμενικής συνάρτησης (*objective ή fitness function*), η οποία παίζει τον ρόλο του περιβάλλοντος μέσα στο οποίο εξελίσσεται ο πληθυσμός.

Η δομή ενός απλού ΓΑ έχει σε γενικές γραμμές ως εξής (Michalewicz, 1996): Κατά τη διάρκεια της γενιάς t ο ΓΑ διατηρεί έναν πληθυσμό $P(t)$ από n πιθανές λύσεις (individuals): $P(t) = \{x_1^t, \dots, x_n^t\}$. Κάθε individual x_i^t αποτιμάται και δίνει ένα μέτρο της καταλληλότητας και ορθότητάς του. Αφού ολοκληρωθεί η αποτίμηση όλων των μελών του πληθυσμού, δημιουργείται ένας νέος πληθυσμός (γενιά $t+1$) που προκύπτει από την επιλογή των πλέον κατάλληλων στοιχείων του πληθυσμού της προηγούμενης γενιάς. Μερικά μέλη από τον καινούριο αυτόν πληθυσμό υφίστανται αλλαγές με τη βοήθεια των γενετικών διαδικασιών της *διασταύρωσης* και της *μετάλλαξης*, σχηματίζοντας νέες πιθανές λύσεις. Η διασταύρωση συνδυάζει τα στοιχεία των χρωμοσωμάτων δύο γονέων για να δημιουργήσει δύο νέους απογόνους, ανταλλάσσοντας κομμάτια από τους γονείς. Π.χ. έστω ότι δύο γονείς αναπαριστώνται με χρωματοσώματα πέντε γονιδίων $(a_1, b_1, c_1, d_1, e_1)$ και $(a_2, b_2, c_2, d_2, e_2)$ αντίστοιχα, τότε οι απόγονοι που θα προκύψουν από διασταύρωση με σημείο διασταύρωσης (*crossover point*) το σημείο 2 είναι οι $(a_1, b_1, c_2, d_2, e_2)$ και $(a_2, b_2, c_1, d_1, e_1)$. Διαισθητικά μπορούμε να πούμε ότι η διασταύρωση εξυπηρετεί την ανταλλαγή πληροφοριών μεταξύ διαφορετικών πιθανών λύσεων. Η διαδικασία της μετάλλαξης αλλάζει αυθαίρετα ένα ή περισσότερα γονίδια ενός συγκεκριμένου χρωμοσώματος. Πραγματοποιείται με τυχαία αλλαγή γονιδίων με πιθανότητα ίση με τον *ρυθμό μετάλλαξης* (*mutation rate*). Π.χ. έστω ότι ένας individual αναπαρίσταται με το διάνυσμα πέντε διαστάσεων $(a_1, b_1, c_1, d_1, e_1)$, τότε ο individual που θα προκύψει με μετάλλαξη στη δεύτερη και στην τέταρτη διάσταση είναι ο $(a_1, b_1^*, c_1, d_1^*, e_1)$. Διαισθητικά μπορούμε να πούμε ότι η μετάλλαξη εξυπηρετεί την εισαγωγή νέων πιθανών λύσεων, διαφορετικών από τις υπάρχουσες, στον ήδη υπάρχοντα πληθυσμό.

Συνοψίζοντας μπορούμε να πούμε ότι ένας ΓΑ για ένα συγκεκριμένο πρόβλημα πρέπει να αποτελείται από τα παρακάτω πέντε συστατικά:

1. Μια γενετική αναπαράσταση των πιθανών λύσεων του προβλήματος.
2. Έναν τρόπο δημιουργίας ενός αρχικού πληθυσμού από πιθανές λύσεις (αρχικοποίηση).
3. Μια αντικειμενική συνάρτηση αξιολόγησης των μελών του πληθυσμού, που παίζει τον ρόλο του περιβάλλοντος.
4. Γενετικούς τελεστές για τη δημιουργία νέων μελών (λύσεων).
5. Τιμές για τις διάφορες παραμέτρους που χρησιμοποιεί ο ΓΑ (μέγεθος πληθυσμού, πιθανότητες εφαρμογής των γενετικών τελεστών κτλ.).

8.1.1 Πλεονεκτήματα των Γενετικών Αλγορίθμων

Μερικά από τα σημαντικότερα πλεονεκτήματα που έχει η χρήση ΓΑ για την επίλυση προβλημάτων είναι τα εξής:

- 1) Μπορούν να επιλύουν δύσκολα προβλήματα γρήγορα και αξιόπιστα. Ένας από τους σημαντικούς λόγους χρήσης των ΓΑ είναι η μεγάλη τους αποδοτικότητα. Τόσο η θεωρία, όσο και η πράξη έχουν δείξει ότι προβλήματα που έχουν πολλές, δύσκολα προσδιορισμένες λύσεις μπορούν να αντιμετωπιστούν καλύτερα από ΓΑ. Είναι δε αξιοσημείωτο ότι συναρτήσεις που παρουσιάζουν μεγάλες διακυμάνσεις και καθιστούν ανεπαρκείς άλλες μεθόδους στην εύρεση των ακρότατών τους, για τους ΓΑ δεν συνιστούν σημεία δυσχέρειας.
- 2) Μπορούν εύκολα να συνεργαστούν με τα υπάρχοντα μοντέλα και συστήματα. Οι ΓΑ προσφέρουν το σημαντικό πλεονέκτημα της χρήσης τους με προσθετικό τρόπο στα μοντέλα που χρησιμοποιούνται σήμερα, μην απαιτώντας την επανασχεδιάσή τους. Μπορούν εύκολα να συνεργαστούν με τον υπάρχοντα κώδικα, χωρίς μεγάλο κόπο. Αυτό συμβαίνει, διότι χρησιμοποιούν μόνο πληροφορίες της διαδικασίας ή συνάρτησης που πρόκειται να βελτιστοποιήσουν, δίχως να ενδιαφέρει άμεσα ο ρόλος της μέσα στο σύστημα ή η όλη δομή του συστήματος.
- 3) Είναι εύκολα επεκτάσιμοι και εξελίξιμοι. Όπως θα γίνει σαφές στα επόμενα κεφάλαια, οι ΓΑ δεν αντιστέκονται σε αλλαγές, επεκτάσεις και μετεξελίξεις, ανάλογα με την κρίση του σχεδιαστή. Σε πολλές

εφαρμογές έχουν αναφερθεί λειτουργίες των ΓΑ που δεν είναι δανεισμένες από τη φύση ή που έχουν υποστεί σημαντικές αλλαγές, πάντα προς όφελος της απόδοσης. Παραλλαγές στο βασικό σχήμα δεν είναι απλά αναγκαίες, αλλά σε ορισμένες περιπτώσεις επιβάλλονται.

- 4) Μπορούν να συμμετέχουν σε υβριδικές μορφές με άλλες μεθόδους. Αν και η ισχύς των ΓΑ είναι μεγάλη, σε μερικές ειδικές περιπτώσεις προβλημάτων, όπου άλλες μέθοδοι συμβαίνει να έχουν πολύ υψηλή αποδοτικότητα, λόγω εξειδίκευσης, υπάρχει η δυνατότητα χρησιμοποίησης ενός υβριδικού σχήματος ΓΑ με άλλη μέθοδο. Αυτό είναι αποτέλεσμα της μεγάλης ευελιξίας τους.
- 5) Εφαρμόζονται σε πολύ περισσότερα πεδία από κάθε άλλη μέθοδο. Το χαρακτηριστικό που τους εξασφαλίζει αυτό το πλεονέκτημα είναι η ελευθερία επιλογής των κριτηρίων που καθορίζουν την επιλογή μέσα στο τεχνικό περιβάλλον. Έτσι ΓΑ μπορούν να χρησιμοποιηθούν στην οικονομία, στον σχεδιασμό μηχανών, στην επίλυση μαθηματικών εξισώσεων, στην εκπαίδευση Νευρωνικών Δικτύων και σε πολλούς άλλους τομείς.
- 6) Δεν απαιτούν περιορισμούς στις συναρτήσεις που επεξεργάζονται. Ο κύριος λόγος που καθιστά τις παραδοσιακές μεθόδους δύσκαμπτες και ακατάλληλες για πολλά προβλήματα είναι η απαίτησή τους για ύπαρξη περιορισμών, όπως ύπαρξη παραγώγων, συνέχεια, όχι «θορυβώδεις» συναρτήσεις κτλ. Τέτοιου είδους ιδιότητες είναι αδιάφορες για τους ΓΑ, κάτι που τους κάνει κατάλληλους για μεγάλο φάσμα προβλημάτων.
- 7) Δεν ενδιαφέρει η σημασία της υπό εξέταση πληροφορίας. Η μόνη «επικοινωνία» του ΓΑ με το περιβάλλον του είναι η αντικειμενική συνάρτηση. Αυτό εγγυάται την επιτυχία του ανεξάρτητα από τη σημασία του προβλήματος. Βέβαια δεν σημαίνει ότι δεν υπάρχουν άλυτα προβλήματα για τους ΓΑ. Όπου όμως δεν τα καταφέρνουν, η αιτία είναι η φύση του χώρου που ερευνούν και όχι το πληροφοριακό περιεχόμενο του προβλήματος.
- 8) Έχουν από τη φύση τους το στοιχείο του παραλληλισμού. Οι ΓΑ σε κάθε τους βήμα επεξεργάζονται μεγάλες ποσότητες πληροφορίας, αφού κάθε άτομο θεωρείται αντιπρόσωπος πολλών άλλων. Έχει υπολογιστεί ότι η αναλογία αυτή είναι της τάξεως $O(n^3)$, δηλαδή 10 άτομα αντιπροσωπεύουν περίπου 1000. Είναι λοιπόν προφανές ότι μπορούν να καλύψουν με αποδοτικό ψάξιμο μεγάλους χώρους σε μικρούς χρόνους.
- 9) Είναι μία μέθοδος που κάνει ταυτόχρονα εξερεύνηση του χώρου αναζήτησης και εκμετάλλευση της ήδη επεξεργασμένης πληροφορίας. Ο συνδυασμός αυτός σπάνια συναντάται σε οποιαδήποτε άλλη μέθοδο. Με το τυχαίο ψάξιμο γίνεται καλή εξερεύνηση του χώρου, αλλά δεν γίνεται εκμετάλλευση της πληροφορίας. Αντίθετα, με το hill-climbing γίνεται καλή εκμετάλλευση της πληροφορίας, αλλά όχι καλή εξερεύνηση. Συνήθως τα δύο αυτά χαρακτηριστικά είναι ανταγωνιστικά και το επιθυμητό είναι να συνυπάρχουν και τα δύο προς όφελος της όλης διαδικασίας. Οι ΓΑ επιτυγχάνουν τον βέλτιστο συνδυασμό εξερεύνησης και εκμετάλλευσης, στοιχείο που τους κάνει ιδιαίτερα αποδοτικούς και ελκυστικούς.
- 10) Επιδέχονται παράλληλη υλοποίηση. Οι ΓΑ μπορούν να εκμεταλλευτούν τα πλεονεκτήματα των παράλληλων μηχανών, αφού λόγω της φύσης τους εύκολα μπορούν να δεχτούν παράλληλη υλοποίηση. Το χαρακτηριστικό αυτό αυξάνει ακόμη περισσότερο την απόδοσή τους, ενώ σπάνια συναντάται σε ανταγωνιστικές μεθόδους.

8.1.2 Τι Προκαλεί Δυσπιστία

Η τεχνολογία των ΓΑ, αν και δεν αποτελεί πρόσφατη ανακάλυψη, άρχισε ουσιαστικά να εφαρμόζεται τα τελευταία χρόνια. Η δυσπιστία με την οποία αντιμετώπιζαν οι επιστήμονες το όλο θέμα έχει αρχίσει πλέον να υποχωρεί. Ποιοι είναι όμως οι κυριότεροι λόγοι που ίσως θα μπορούσαν να σταθούν εμπόδιο στην εξάπλωση αυτής της τεχνολογίας;

- 1) Προβλήματα εξοικείωσης με τη Γενετική. Για τους περισσότερους που ασχολούνται με την Επιστήμη των Υπολογιστών, οι έννοιες της Εξέλιξης και της Φυσικής Επιλογής μπορεί να μην ηχούν παράξενα, αλλά δεν είναι και από τις πιο οικείες. Η Βιολογία δεν έχει άμεση σχέση με τους υπολογιστές, γι' αυτό και οι γνώσεις σχεδόν όλων είναι σε πολύ γενικό επίπεδο. Παρ' όλα αυτά, δεν απαιτούνται γνώσεις Γενετικής και Βιολογίας. Εκείνο που συμβαίνει με τους ΓΑ είναι ότι μιμούνται με αφαιρετικό τρόπο κάποιες διαδικασίες που παρατηρούνται στη φύση, χωρίς να ενδιαφέρει σε μεγάλο βαθμό λεπτομέρειας η λειτουργία τους και χωρίς να είναι απαραίτητο το γνωστικό υπόβαθρο που έχουν οι βιολόγοι για να μελετήσουν αυτά τα φαινόμενα. Οι όροι είναι δανεισμένοι από τη βιολογία με σκοπό την καλύτερη εισαγωγή και κατανόηση του θέματος και όχι την παραπομπή του μελετητή στα άγνωστα πεδία μιας ξένης επιστήμης και τελικά τη σύγχυσή του. Θα μπορούσε ίσως να παραλειφθεί η αναφορά στη Γενετική και να παρουσιαστούν οι ΓΑ ως «προσωπικές διαδικασίες για αναζήτηση και βελτιστοποίηση», αυτό όμως μάλλον θα έκανε τα πράγματα δυσκολότερα. Εξάλλου είναι συνηθισμένο το φαινόμενο θεωρίες που είναι δανεισμένες από άλλες επιστήμες να διατηρούν την αυθεντική τους ορολογία (π.χ. στα Νευρωνικά Δίκτυα: νευρώνες, συνάψεις κτλ.). Επιπλέον, το μέλλον και η εξέλιξη των ΓΑ δεν εξαρτώνται σε καμία περίπτωση από τις αντίστοιχες θεωρίες της Βιολογίας. Το αρχικό μοντέλο είναι δανεισμένο από εκεί, όμως η εφαρμογή του στα Τεχνητά Συστήματα έγινε με πλήθος διαφοροποιήσεων, προσαρμογών και «παραεκτροπών» με στόχο πάντα τη βελτίωση της απόδοσης. Πλέον μπορούμε να μιλάμε για εξέλιξη και απογόνους των πρώτων ΓΑ και για μια πορεία τους στον χρόνο που είναι πλήρως ανεξάρτητη και αυτοδύναμη.
- 2) Το πρόβλημα του χρόνου. Στη φύση —όπως είναι γνωστό— η εξέλιξη λειτουργεί με ρυθμούς πολύ αργούς. Χρειάζεται να περάσουν χιλιάδες γενιές, άρα και αρκετός χρόνος, για να αλλάξουν τα χαρακτηριστικά των ειδών και να διαφοροποιηθούν οι ικανότητες και η συμπεριφορά τους. Θέτουν έτσι ορισμένοι το ερώτημα: Πώς είναι δυνατό ένα μοντέλο αναζήτησης λύσεων να έχει καλές επιδόσεις χρόνου, όταν είναι εμπνευσμένο από μια φυσική διαδικασία που εξελίσσεται με ρυθμούς απίστευτα αργούς; Η απάντηση εδώ είναι απλή. Καταρχήν, ακόμη και στη φύση, η εξέλιξη δεν είναι από μόνη της μια αργή διαδικασία. Εξέλιξη των ειδών συμβαίνει όταν αλλάζει τα περιβάλλον τους και πρέπει να προσαρμοστούν στα καινούρια δεδομένα, ώστε να επιβιώσουν. Αλλαγές όμως του περιβάλλοντος γίνονται με πολύ αργούς ρυθμούς· κατά συνέπεια και η εξέλιξη ακολουθεί αυτούς τους ρυθμούς. Αν οι αλλαγές του περιβάλλοντος γίνονται με γρηγορότερο τρόπο, τότε επιταχύνεται και η εξέλιξη. Αυτό άλλωστε παρατηρείται και στα βιολογικά εργαστήρια, όπου μικροοργανισμοί αλλάζουν τη συμπεριφορά τους αμέσως, όταν τοποθετούνται σε νέες συνθήκες. Επιπλέον στο πεδίο των υπολογιστών τα άτομα κωδικοποιούνται συνήθως ως συμβολοσειρές και οι συνθήκες του περιβάλλοντος μοντελοποιούνται με απλές μαθηματικές σχέσεις. Έτσι το μοντέλο με το οποίο δουλεύει ο υπολογιστής δεν παρουσιάζει ιδιαίτερο υπολογιστικό φόρτο, συγκρινόμενο πάντα με αντίστοιχες μεθόδους. Το πλήθος των ατόμων που κάθε φορά εξετάζεται ποικίλλει από λίγες δεκάδες έως μερικές χιλιάδες, δηλαδή αρκετές τάξεις μεγέθους κάτω από το πλήθος των γονιδίων των χρωμοσωμάτων μιας έμβιας οντότητας. Ο ρυθμός που μπορούν να ζευγαρώνουν τα άτομα στους πιο γρήγορους υπολογιστές μπορεί να φτάσει το ένα εκατομμύριο ανά δευτερόλεπτο. Όσο μεγάλος και αν είναι ο χώρος που καλείται ο αλγόριθμος να ψάξει, η επεξεργασία μερικών μόνο ατόμων αρκεί, γιατί, όπως θα αναπτυχθεί και παρακάτω, τα άτομα αυτά θεωρούνται αντιπρόσωποι ολόκληρων κλάσεων. Έτσι λοιπόν οι ταχύτητες που μπορούν να επιτύχουν οι ΓΑ είναι πολύ υψηλές.

8.1.3 Βασικά Χαρακτηριστικά Γενετικών Αλγορίθμων

Όπως αναφέρθηκε, οι ΓΑ πλεονεκτούν αισθητά στη λύση προβλημάτων αναζήτησης και βελτιστοποίησης από τις παραδοσιακές μεθόδους. Αυτό συμβαίνει, διότι διαφέρουν θεμελιωδώς από αυτές. Τα κυριότερα νέα χαρακτηριστικά που τους διαφοροποιούν, αλλά και τους δίνουν υπεροχή, σύμφωνα με τον D. Goldberg (Goldberg, 1989), είναι τα εξής:

- 1) Οι ΓΑ δουλεύουν με μια κωδικοποίηση του συνόλου τιμών που μπορούν να λάβουν οι μεταβλητές και όχι με τις ίδιες τις μεταβλητές του προβλήματος. Π.χ. αναφέρεται το εξής πρόβλημα βελτιστοποίησης: έστω ένα μαύρο κουτί με πέντε δυαδικούς διακόπτες (on-off). Για κάθε συνδυασμό s των διακοπών παράγεται μία έξοδος $f(s)$. Ζητείται ο συνδυασμός των διακοπών που τη μεγιστοποιεί. Με τις παραδοσιακές μεθόδους το μέγιστο θα εντοπιζόταν με «παιξίμο» των διακοπών, πηγαίνοντας από συνδυασμό σε συνδυασμό με ψάξιμο στα τυφλά, αφού δεν είναι γνωστός ο τύπος της συνάρτησης. Σε ένα ΓΑ όμως η πρώτη ενέργεια είναι η κωδικοποίηση των διακοπών ως συμβολοσειρές πεπερασμένου μήκους. Μια απλή κωδικοποίηση θα μπορούσε να γίνει θεωρώντας μια δυαδική συμβολοσειρά μήκους πέντε, όπου η κάθε θέση αναπαριστά έναν διακόπτη. Το 0 αντιστοιχεί στη θέση off και το 1 στη θέση on· δηλαδή η συμβολοσειρά 11110 κωδικοποιεί τον συνδυασμό κατά τον οποίο οι πρώτοι τέσσερις διακόπτες είναι on και ο τελευταίος off. Η κωδικοποίηση δεν είναι απαραίτητο να είναι πάντα δυαδική. Όπως θα φανεί και αργότερα, μπορεί να γίνει με πολλούς τρόπους, αρκετοί από τους οποίους ίσως και να μην είναι προφανείς. Το στοιχείο της κωδικοποίησης, όπως εξηγείται παρακάτω, είναι εκείνο που επιτρέπει στους ΓΑ να κάνουν παράλληλη επεξεργασία δεδομένων.
- 2) Οι ΓΑ κάνουν αναζήτηση σε πολλά σημεία ταυτόχρονα και όχι μόνο σε ένα. Σε πολλές μεθόδους βελτιστοποίησης η επεξεργασία γίνεται βήμα προς βήμα, πηγαίνοντας προσεκτικά από σημείο σε σημείο του πεδίου ορισμού του προβλήματος. Αυτό το βήμα προς βήμα ενέχει αρκετούς κινδύνους, ο κυριότερος από τους οποίους είναι να περιοριστεί η αναζήτηση σε μια περιοχή τοπικού ακρότατου, που δεν είναι ολικό. Οι ΓΑ εξαλείφουν αυτόν τον κίνδυνο ενεργώντας ταυτόχρονα πάνω σε ένα ευρύ σύνολο σημείων (σύνολο από συμβολοσειρές). Έτσι μπορούν να «ανεβαίνουν» πολλούς λόφους (hill-climbing) την ίδια στιγμή, ελαχιστοποιώντας την πιθανότητα να βρουν μια λάθος κορυφή. Γυρίζοντας στο παράδειγμα με το μαύρο κουτί, οι κλασικές μέθοδοι θα ξεκινούσαν το ψάξιμο από έναν συνδυασμό των διακοπών και στη συνέχεια, εφαρμόζοντας κάποιον κανόνα μετάβασης, θα δοκίμαζαν τον επόμενο (ψάξιμο δηλαδή σημείο προς σημείο). Αντιθέτως ένας ΓΑ αρχίζει το ψάξιμό του από έναν πληθυσμό συνδυασμών συμβολοσειρών και κατόπιν παράγει διαδοχικά καινούριους. Ένας αρχικός πληθυσμός θα μπορούσε να είναι π.χ. 01101, 11000, 01000 και 10011. Έπειτα, «τρέχοντας» ο αλγόριθμος, δημιουργεί νέους πληθυσμούς που συγκλίνουν προς την επιθυμητή λύση. Διαλέγοντας έναν πληθυσμό που να καλύπτει αντιπροσωπευτικά ένα μεγάλο εύρος τιμών μπορούν να προκύψουν ικανοποιητικά αποτελέσματα.
- 3) Οι ΓΑ χρησιμοποιούν μόνο την αντικειμενική συνάρτηση και καμία επιπρόσθετη πληροφορία: Πολλές μέθοδοι αναζήτησης απαιτούν αρκετές βοηθητικές πληροφορίες για τη συνάρτηση που επεξεργάζονται. Τέτοιου είδους πληροφορίες δεν προαπαιτούνται από τους ΓΑ. Το ψάξιμό τους είναι κατά κάποιον τρόπο «τυφλό». Αξιοποιούν μόνο όση πληροφορία περιέχεται στην αντικειμενική συνάρτηση, κάτι που τους προσδίδει μεγάλη ευελιξία. Έτσι όμως προκύπτει το ερώτημα αν συμφέρει να αγνοούνται οι βοηθητικές πληροφορίες. Γι' αυτόν ακριβώς τον λόγο έχουν αναπτυχθεί μορφές ΓΑ που αξιοποιούν και τέτοιου είδους πληροφορίες (Knowledge-Based Genetic Algorithms).
- 4) Οι ΓΑ χρησιμοποιούν πιθανοθεωρητικούς κανόνες μετάβασης και όχι ντετερμινιστικούς: Η χρήση πιθανοτικών κανόνων μετάβασης είναι κυρίαρχο γνώρισμά τους, χωρίς αυτό να σημαίνει ότι η όλη διαδικασία βαδίζει στην τύχη. Δεν λαμβάνονται αποφάσεις με το «στρίψιμο ενός νομίσματος». Το στοιχείο της τύχης χρησιμοποιείται ως οδηγός για αναζήτηση σε περιοχές που αναμένεται να δώσουν καλά αποτελέσματα.

Τα τέσσερα προαναφερθέντα χαρακτηριστικά συμβάλλουν αποφασιστικά ώστε να έχουν οι ΓΑ την πολυπόθητη ιδιότητα της ευρωστίας.

Ασκήσεις

Λύνοντας αυτές τις ασκήσεις θα είστε σε θέση να γνωρίζετε αν έχετε κατανοήσει τις έννοιες που παρουσιάστηκαν στην ενότητα 1. Όλες οι ασκήσεις έχουν ως στόχο την ανακεφαλαίωση όσων παρουσιάστηκαν εκεί.

Άσκηση 8.1.1 (Επίπεδο δυσκολίας 1): Δώστε με τη μορφή διαγράμματος ροής τη μορφή ενός απλού Γενετικού Αλγορίθμου.

Άσκηση 8.1.2 (Επίπεδο δυσκολίας 1): Ποια είναι τα πέντε βασικά συστατικά ενός Γενετικού Αλγορίθμου;

Άσκηση 8.1.3 (Επίπεδο δυσκολίας 1): Αναφέρετε μερικά από τα πλεονεκτήματα των Γενετικών Αλγορίθμων που κάνουν ιδιαίτερα ελκυστική τη χρήση τους σε διάφορα προβλήματα.

Άσκηση 8.1.4 (Επίπεδο δυσκολίας 1): Ποια είναι τα βασικά επιχειρήματα των πολέμιων των Γενετικών Αλγορίθμων;

Άσκηση 8.1.5 (Επίπεδο δυσκολίας 1): Ποια είναι τα κυριότερα χαρακτηριστικά που διαφοροποιούν τους Γενετικούς Αλγορίθμους από τις παραδοσιακές μεθόδους βελτιστοποίησης;

8.2 Βασικά Στοιχεία Γενετικού Αλγορίθμου

Σε αυτήν την ενότητα θα παρουσιάσουμε τα βασικά χαρακτηριστικά ενός απλού Γενετικού Αλγορίθμου μέσα από την επίλυση ενός προβλήματος βελτιστοποίησης, που είναι δανεισμένο από τον Goldberg, 1989. Έστω ότι το πρόβλημα βελτιστοποίησης που θέλουμε να επιλύσουμε είναι ένα πρόβλημα μεγιστοποίησης μιας συνάρτησης f — στην περίπτωση τώρα που αντιμετωπίζουμε ένα πρόβλημα ελαχιστοποίησης μιας συνάρτησης f , αυτό μπορούμε να το μετασχηματίσουμε σε πρόβλημα μεγιστοποίησης της συνάρτησης $g = -f$. Επιπλέον, υποθέτουμε ότι η αντικειμενική συνάρτηση f παίρνει μόνο θετικές τιμές — διαφορετικά μπορούμε να εισάγουμε μια θετική σταθερά C , έτσι ώστε

$$\max g(x) = \max \{f(x) + C\}$$

Έστω λοιπόν ότι θέλουμε να μεγιστοποιήσουμε μια συνάρτηση k μεταβλητών, $f(x_1, \dots, x_k): \mathbb{R}^k \rightarrow \mathbb{R}$. Κάθε μεταβλητή x_i παίρνει τιμές στο διάστημα $D_i = [a_i, b_i] \subseteq \mathbb{R}$ και $f(x_1, \dots, x_k) > 0$, $\forall x_i \in D_i$, $i=1, \dots, k$. Επιθυμούμε να βελτιστοποιήσουμε την f με κάποια απαιτούμενη ακρίβεια q δεκαδικών ψηφίων για κάθε μεταβλητή.

Ένα από τα βασικά χαρακτηριστικά του γενετικού αλγορίθμου είναι η **γενετική αναπαράσταση** των υποψήφιων λύσεων, η οποία στο συγκεκριμένο πρόβλημα θα είναι η δυαδική. Για να επιτευχθεί η ζητούμενη ακρίβεια, θα πρέπει κάθε διάστημα τιμών $D_i = [a_i, b_i]$ να διαχωριστεί σε $(b_i - a_i) \times 10^q$ ίσα υποδιαστήματα. Έστω m_i ο μικρότερος ακέραιος για τον οποίο ισχύει $(b_i - a_i) \cdot 10^q \leq 2^{m_i} - 1$. Τότε η αναπαράσταση των μεταβλητών με δυαδικές συμβολοσειρές μήκους m_i ικανοποιεί την απαίτηση για ακρίβεια q δεκαδικών ψηφίων. Η ακόλουθη σχέση μετατρέπει κάθε τέτοια δυαδική συμβολοσειρά `bin_str` στον αντίστοιχο πραγματικό αριθμό:

$$x_i = a_i + \text{decimal}(\text{bin_str}) \cdot \frac{b_i - a_i}{2^{m_i} - 1},$$

όπου `decimal(bin_str)` επιστρέφει την αντίστοιχη δεκαδική τιμή για τον δυαδικό αριθμό που περιέχει η `bin_str`.

Κατ' αυτόν τον τρόπο κάθε χρωμόσωμα αναπαρίσταται από μια δυαδική συμβολοσειρά μήκους $m = \sum_{i=1}^k m_i$.

Τα πρώτα m_1 δυαδικά ψηφία κωδικοποιούν τη μεταβλητή x_1 , δηλαδή το διάστημα $[a_1, b_1]$, τα επόμενα m_2 κωδικοποιούν τη x_2 στο διάστημα $[a_2, b_2]$, κ.ο.κ.

Τα βασικά βήματα του απλού Γενετικού Αλγορίθμου, που επιλύει το παραπάνω πρόβλημα μεγιστοποίησης, είναι τα εξής:

1. Δημιουργία, με τυχαίο τρόπο, ενός αρχικού πληθυσμού δυνατών λύσεων (*Αρχικοποίηση - Initialization*).
2. Αξιολόγηση κάθε λύσης χρησιμοποιώντας τη συνάρτηση f ως αντικειμενική συνάρτηση.
3. Επιλογή ενός νέου πληθυσμού με βάση την απόδοση κάθε μέλους (δυνατής λύσης) του προηγούμενου πληθυσμού (*Επιλογή - Selection*).
4. Εφαρμογή στον πληθυσμό που προκύπτει μετά τη διαδικασία της επιλογής των γενετικών τελεστών της διασταύρωσης (*Crossover*) και της μετάλλαξης (*Mutation*).
5. Με την ολοκλήρωση του προηγούμενου βήματος έχει δημιουργηθεί η επόμενη γενιά, οπότε επιστρέφουμε στο βήμα 2.
6. Μετά από κάποιον αριθμό γενιών, και αφού καμιά βελτίωση δεν παρατηρείται πλέον, ο Γενετικός Αλγόριθμος τερματίζεται.

Το καλύτερο χρωμόσωμα αντιστοιχεί σε μια βέλτιστη λύση (πιθανώς καθολικά βέλτιστη).

Στη συνέχεια θα περιγράψουμε πιο αναλυτικά τα βήματα του αλγορίθμου.

Αρχικοποίηση

Στη φάση της αρχικοποίησης δημιουργούμε έναν αρχικό πληθυσμό από δυνατές λύσεις. Αυτό γίνεται παράγοντας τυχαία ($pop_size \times m$) δυαδικά ψηφία, όπου pop_size είναι το μέγεθος του πληθυσμού που θα επεξεργαστεί ο ΓΑ. Το μέγεθος του πληθυσμού παραμένει σταθερό καθόλη τη διάρκεια λειτουργίας του ΓΑ.

Επιλογή

Για τη διαδικασία επιλογής ενός νέου πληθυσμού χρησιμοποιείται μια ρουλέτα με σχισμές (*slotted roulette wheel*). Η επιλογή γίνεται με βάση την απόδοση κάθε μέλους (ατόμου - δυνατής λύσης) του πληθυσμού· έτσι όσο καλύτερο είναι κάποιο μέλος, τόσο μεγαλύτερη πιθανότητα έχει να επιλεγεί και να περάσει στην επόμενη γενεά. Τα διάφορα μέλη του πληθυσμού τοποθετούνται στη ρουλέτα ανάλογα με την απόδοσή τους (μέλη με μεγάλη απόδοση κατέχουν μεγαλύτερες σχισμές της ρουλέτας).

Η κατασκευή μιας τέτοιας ρουλέτας γίνεται ως εξής:

- Υπολογίζουμε την απόδοση $eval(v_i)$ κάθε μέλους v_i , $i=1, \dots, pop_size$ του πληθυσμού.
- Υπολογίζουμε τη συνολική απόδοση του πληθυσμού $F = \sum_{i=1}^{pop_size} eval(v_i)$.
- Υπολογίζουμε την πιθανότητα επιλογής p_i κάθε μέλους v_i , $i=1, \dots, pop_size$: $p_i = eval(v_i) / F$.
- Τέλος, υπολογίζουμε την αθροιστική (cumulative) πιθανότητα q_i κάθε μέλους v_i , $i=1, \dots, pop_size$:

$$q_i = \sum_{j=1}^i p_j.$$

Για την επιλογή των μελών του νέου πληθυσμού εκτελούμε pop_size περιστροφές της ρουλέτας. Αυτό γίνεται ως εξής:

1. Επιλέγουμε τυχαία έναν πραγματικό αριθμό r μεταξύ 0 και 1.
2. Αν $r < q_1$, τότε επιλέγουμε το πρώτο χρωμόσωμα v_1 , διαφορετικά επιλέγουμε το v_i ($2 \leq i \leq pop_size$), έτσι ώστε $(q_{i-1} < r \leq q_i)$.

Προφανώς με αυτήν τη μέθοδο επιλογής είναι δυνατόν κάποια μέλη του πληθυσμού να επιλεγθούν περισσότερες από μία φορές, με αυτά που είχαν την καλύτερη απόδοση στην προηγούμενη γενιά να έχουν τις περισσότερες πιθανότητες γι' αυτό.

Διασταύρωση

Στη συνέχεια εφαρμόζεται ο τελεστής της διασταύρωσης —ο οποίος στο συγκεκριμένο ΓΑ είναι διασταύρωση ενός σημείου— στον νέο πληθυσμό. Θεωρούμε ότι η πιθανότητα κάθε μέλους του πληθυσμού να επιλεγεί για διασταύρωση είναι p_c . Για κάθε μέλος του πληθυσμού κάνουμε τα εξής:

- Επιλέγουμε τυχαία έναν πραγματικό αριθμό r μεταξύ 0 και 1.
- Αν $r < p_c$, επιλέγουμε το τρέχον μέλος του πληθυσμού για διασταύρωση.

Μετά την επιλογή μελών του πληθυσμού για διασταύρωση (ο αναμενόμενος αριθμός αυτών των μελών είναι $p_c \times pop_size$) σχηματίζουμε ζευγάρια από μέλη και για κάθε ζευγάρι επιλέγεται τυχαία ένας ακέραιος αριθμός pos στο διάστημα $[1, m-1]$, όπου m είναι το μήκος σε δυαδικά ψηφία του (μοναδικού στην περίπτωση μας) χρωμοσώματος κάθε μέλους. Ο αριθμός pos προσδιορίζει το σημείο διασταύρωσης. Τα επιλεγμένα ζευγάρια διασταυρώνονται και τη θέση τους στον πληθυσμό την παίρνουν οι απόγονοί τους. Έτσι η διασταύρωση των δύο παρακάτω ατόμων:

$$\begin{aligned} & (b_1 b_2 \dots b_{pos} b_{pos+1} \dots b_m) \\ & (c_1 c_2 \dots c_{pos} c_{pos+1} \dots c_m) \end{aligned}$$

θα δώσει το ακόλουθο ζευγάρι απογόνων:

$$(b_1 b_2 \dots b_{pos} c_{pos+1} \dots c_m)$$

$$(c_1 c_2 \dots c_{pos} b_{pos+1} \dots b_m) \cdot$$

Πρόκειται για απογόνους οι οποίοι θα αντικαταστήσουν τους «γονείς» τους στον πληθυσμό.

Μετάλλαξη

Η μετάλλαξη επιλέγει με τυχαίο τρόπο γονίδια από τα χρωμοσώματα των μελών του πληθυσμού και μεταβάλλει την τιμή τους. Επειδή στον συγκεκριμένο ΓΑ τα γονίδια παίρνουν μόνο δυαδικές τιμές, ο τελεστής της μετάλλαξης απλώς τα αντιστρέφει. Έτσι, στον ΓΑ που παρουσιάζουμε, η μετάλλαξη αντιμετωπίζει τον πληθυσμό των ατόμων ως μια ακολουθία από δυαδικά ψηφία. Κάθε δυαδικό ψηφίο έχει την ίδια πιθανότητα να επιλεγεί προκειμένου να μεταλλαχθεί. Η πιθανότητα αυτή ισούται με την πιθανότητα μετάλλαξης p_m . Ο αναμενόμενος αριθμός των ανεστραμμένων ψηφίων μετά τη διαδικασία της μετάλλαξης θα είναι ($p_m \times m \times pop_size$). Η διαδικασία έχει ως εξής:

Για κάθε χρωμόσωμα μέλους και κάθε γονίδιο μέσα στο χρωμόσωμα:

- Επιλέγουμε τυχαία έναν πραγματικό αριθμό r μεταξύ 0 και 1.
- Αν $r < p_m$, τότε μεταλλάζουμε το γονίδιο (αντιστρέφουμε το δυαδικό ψηφίο).

8.2.1 Εφαρμογή – μεγιστοποίηση συνάρτησης

Στην ενότητα αυτή θα εφαρμόσουμε τον ΓΑ, που περιγράψαμε παραπάνω, για τη μεγιστοποίηση της συνάρτησης (Goldberg, 1989):

$$f(x_1, x_2) = 21.5 + x_1 \cdot \sin(4 \cdot \pi \cdot x_1) + x_2 \cdot \sin(20 \cdot \pi \cdot x_2),$$

με $-3.0 \leq x_1 \leq 12.1$ και $4.1 \leq x_2 \leq 5.8$.

Υποθέτουμε ότι το μέγεθος του πληθυσμού που θα επεξεργαστεί ο ΓΑ είναι $pop_size=20$ και οι πιθανότητες διασταύρωσης και μετάλλαξης είναι $p_c=0.25$ και $p_m=0.01$ αντίστοιχα.

Έστω ότι η επιθυμητή ακρίβεια για κάθε μεταβλητή είναι τέσσερα δεκαδικά ψηφία. Το διάστημα τιμών της μεταβλητής x_1 έχει μήκος $15.1 (=12.1 - (-3.0))$, οπότε το διάστημα $[-3.0, 12.1]$ θα πρέπει να διαχωριστεί σε 15.1×10000 ίσα υποδιαστήματα. Αυτό σημαίνει ότι απαιτούνται 18 δυαδικά ψηφία για τη δυαδική αναπαράσταση της x_1 (και τα οποία θα αποτελούν το πρώτο τμήμα του χρωμοσώματος κάθε ατόμου), αφού:

$$2^{17} < 151000 \leq 2^{18}$$

Με παρόμοιο τρόπο υπολογίζουμε ότι για τη δυαδική αναπαράσταση της x_2 απαιτούνται 15 δυαδικά ψηφία (επαληθεύστε το). Επομένως το συνολικό μήκος του μοναδικού χρωμοσώματος κάθε μέλους του πληθυσμού θα είναι $m = 18 + 15 = 33$ δυαδικά ψηφία.

Έστω λοιπόν το ακόλουθο άτομο:

$$(01000100101101000011110010100010)$$

Τα πρώτα 18 δυαδικά ψηφία (010001001011010000) αναπαριστούν τον δεκαδικό αριθμό:

$$x_1 = -3.0 + 70352 \cdot \frac{15.1}{262143} = -3.0 + 4.05242 = 1.05242,$$

και τα επόμενα 15 δυαδικά ψηφία (11110010100010) τον αριθμό $x_2=5.75533$ (επαληθεύστε το)· δηλαδή το άτομο αυτό αντιστοιχεί στο ζεύγος $\langle x_1, x_2 \rangle = \langle 1.05242, 5.75533 \rangle$.

Η απόδοση αυτού του ατόμου είναι $f(1.05242, 5.75533) = 20.25264$ (επαληθεύστε το).

Εδώ θα πρέπει να σημειώσουμε ότι μιας και τα άτομα που επεξεργάζεται ο ΓΑ αποτελούνται από ένα μόνο χρωμόσωμα, στη συνέχεια όταν αναφερόμαστε σε άτομα ή χρωμοσώματα, θα εννοούμε το ίδιο πράγμα.

Αρχικοποίηση

Υποθέτουμε ότι μετά την αρχικοποίηση προκύπτει ο ακόλουθος πληθυσμός:

v_1	=	(100110100000001111111010011011111)
v_2	=	(111000100100110111001010100011010)
v_3	=	(000010000011001000001010111011101)
v_4	=	(100011000101101001111000001110010)
v_5	=	(000111011001010011010111111000101)
v_6	=	(000101000010010101001010111111011)
v_7	=	(001000100000110101111011011111011)
v_8	=	(100001100001110100010110101100111)
v_9	=	(010000000101100010110000001111100)
v_{10}	=	(000001111000110000011010000111011)
v_{11}	=	(011001111110110101100001101111000)
v_{12}	=	(110100010111101101000101010000000)
v_{13}	=	(111011111010001000110000001000110)
v_{14}	=	(010010011000001010100111100101001)
v_{15}	=	(111011101101110000100011111011110)
v_{16}	=	(110011110000011111100001101001011)
v_{17}	=	(011010111111001111010001101111101)
v_{18}	=	(011101000000001110100111110101101)
v_{19}	=	(000101010011111111110000110001100)
v_{20}	=	(101110010110011110011000101111110)

Επιλογή

Αρχικά αποκωδικοποιούμε κάθε χρωμόσωμα και υπολογίζουμε την απόδοσή του. Έτσι έχουμε:

$eval(v_1)$	=	$f(6.084492, 5.652242)$	=	26.019600
$eval(v_2)$	=	$f(10.348434, 4.380264)$	=	7.580015
$eval(v_3)$	=	$f(-2.516603, 4.390381)$	=	19.526329
$eval(v_4)$	=	$f(5.278638, 5.593460)$	=	17.406725
$eval(v_5)$	=	$f(-1.255173, 4.734458)$	=	25.341160
$eval(v_6)$	=	$f(-1.811725, 4.391937)$	=	18.100417
$eval(v_7)$	=	$f(-0.991471, 5.680258)$	=	16.020812
$eval(v_8)$	=	$f(4.910618, 4.703018)$	=	17.959701
$eval(v_9)$	=	$f(0.795406, 5.381472)$	=	16.127799
$eval(v_{10})$	=	$f(-2.554851, 4.793707)$	=	21.278435
$eval(v_{11})$	=	$f(3.130078, 4.996097)$	=	23.410669
$eval(v_{12})$	=	$f(9.356179, 4.239457)$	=	15.011619
$eval(v_{13})$	=	$f(11.134646, 5.378671)$	=	27.316702

$eval(v_{14})$	$= f(1.335944, 5.151378)$	$= 19.876294$
$eval(v_{15})$	$= f(11.089025, 5.054515)$	$= 30.060205$
$eval(v_{16})$	$= f(9.211598, 4.993762)$	$= 23.867227$
$eval(v_{17})$	$= f(3.367514, 4.571343)$	$= 13.696165$
$eval(v_{18})$	$= f(3.843020, 5.158226)$	$= 15.414128$
$eval(v_{19})$	$= f(-1.746635, 5.395584)$	$= 20.095903$
$eval(v_{20})$	$= f(7.935998, 4.757338)$	$= 13.666916$

Είναι εμφανές ότι το άτομο v_{15} είναι αυτό με την καλύτερη απόδοση και το άτομο v_2 αυτό με τη χειρότερη απόδοση στον αρχικό πληθυσμό.

Στη συνέχεια κατασκευάζουμε μια ρουλέτα (roulette wheel). Η συνολική απόδοση (fitness) του πληθυσμού είναι:

$$F = \sum_{i=1}^{20} eval(v_i) = 387.776822$$

Η πιθανότητα επιλογής p_i κάθε μέλους του πληθυσμού v_i , $i=1, \dots, 20$, είναι:

$p_1 = eval(v_1)/F = 0.067099$	$p_{11} = eval(v_{11})/F = 0.060372$
$p_2 = eval(v_2)/F = 0.019547$	$p_{12} = eval(v_{12})/F = 0.038712$
$p_3 = eval(v_3)/F = 0.050355$	$p_{13} = eval(v_{13})/F = 0.070444$
$p_4 = eval(v_4)/F = 0.044889$	$p_{14} = eval(v_{14})/F = 0.051257$
$p_5 = eval(v_5)/F = 0.065350$	$p_{15} = eval(v_{15})/F = 0.077519$
$p_6 = eval(v_6)/F = 0.046677$	$p_{16} = eval(v_{16})/F = 0.061549$
$p_7 = eval(v_7)/F = 0.041315$	$p_{17} = eval(v_{17})/F = 0.035320$
$p_8 = eval(v_8)/F = 0.046315$	$p_{18} = eval(v_{18})/F = 0.039750$
$p_9 = eval(v_9)/F = 0.041590$	$p_{19} = eval(v_{19})/F = 0.051823$
$p_{10} = eval(v_{10})/F = 0.054873$	$p_{20} = eval(v_{20})/F = 0.035244$

Οι αθροιστικές πιθανότητες (cumulative probabilities) q_i για κάθε άτομο v_i , $i=1, \dots, 20$ του πληθυσμού είναι:

$q_1 = 0.067099$	$q_6 = 0.293917$	$q_{11} = 0.538381$	$q_{16} = 0.837863$
$q_2 = 0.086647$	$q_7 = 0.335232$	$q_{12} = 0.577093$	$q_{17} = 0.873182$
$q_3 = 0.137001$	$q_8 = 0.381546$	$q_{13} = 0.647537$	$q_{18} = 0.912932$
$q_4 = 0.181890$	$q_9 = 0.423137$	$q_{14} = 0.698794$	$q_{19} = 0.964756$
$q_5 = 0.247240$	$q_{10} = 0.478009$	$q_{15} = 0.776314$	$q_{20} = 1.000000$

Τώρα είμαστε έτοιμοι να περιστρέψουμε τη ρουλέτα 20 φορές; σε κάθε περιστροφή επιλέγουμε και ένα άτομο για τον νέο πληθυσμό. Υποθέτουμε ότι έχουμε παραγάγει την εξής ακολουθία 20 τυχαίων αριθμών στο διάστημα $[0, 1]$:

0.513870	0.175741	0.308652	0.534534	0.947628
0.171736	0.702231	0.226431	0.494773	0.424720
0.703899	0.389647	0.277226	0.368071	0.983437
0.005398	0.765682	0.646473	0.767139	0.780237

Ο πρώτος αριθμός $r = 0.513870$ είναι μεγαλύτερος του q_{10} και μικρότερος του q_{11} , γεγονός που σημαίνει ότι το άτομο v_{11} επιλέγεται για να «περάσει» στον νέο πληθυσμό. Ο δεύτερος αριθμός $r = 0.175741$ είναι μεγαλύτερος του q_3 και μικρότερος του q_4 , οπότε το άτομο v_4 επιλέγεται για τον νέο πληθυσμό. Συνεχίζοντας με τον ίδιο τρόπο κατασκευάζουμε τον νέο πληθυσμό:

v_1^*	=	(011001111110110101100001101111000)	(v_{11})
v_2^*	=	(100011000101101001111000001110010)	(v_4)
v_3^*	=	(00100010000011010111101101111011)	(v_7)
v_4^*	=	(011001111110110101100001101111000)	(v_{11})
v_5^*	=	(000101010011111111110000110001100)	(v_{19})
v_6^*	=	(100011000101101001111000001110010)	(v_4)
v_7^*	=	(111011101101110000100011111011110)	(v_{15})
v_8^*	=	(000111011001010011010111111000101)	(v_5)
v_9^*	=	(011001111110110101100001101111000)	(v_{11})
v_{10}^*	=	(000010000011001000001010111011101)	(v_3)
v_{11}^*	=	(111011101101110000100011111011110)	(v_{15})
v_{12}^*	=	(010000000101100010110000001111100)	(v_9)
v_{13}^*	=	(00010100001001010100101011111011)	(v_6)
v_{14}^*	=	(100001100001110100010110101100111)	(v_8)
v_{15}^*	=	(101110010110011110011000101111110)	(v_{20})
v_{16}^*	=	(100110100000001111111010011011111)	(v_1)
v_{17}^*	=	(000001111000110000011010000111011)	(v_{10})
v_{18}^*	=	(111011111010001000110000001000110)	(v_{13})
v_{19}^*	=	(111011101101110000100011111011110)	(v_{15})
v_{20}^*	=	(110011110000011111100001101001011)	(v_{16})

Διασταύρωση

Στον νέο πληθυσμό που έχει προκύψει εφαρμόζουμε τον τελεστή της διασταύρωσης. Έχουμε αρχικά υποθέσει ότι η πιθανότητα διασταύρωσης είναι $pc=0.25$, οπότε περιμένουμε ότι κατά μέσο όρο το 25% των μελών του πληθυσμού μας θα επιλεγούν για τη διαδικασία της διασταύρωσης. Για το συγκεκριμένο μέγεθος πληθυσμού (20) περιμένουμε να επιλεγούν κατά μέσο όρο 5 άτομα. Έτσι, για κάθε μέλος του πληθυσμού, διαλέγουμε έναν τυχαίο πραγματικό αριθμό r στο διάστημα $[0,1]$. Αν $r < 0.25$, τότε επιλέγουμε το συγκεκριμένο άτομο. Ας υποθέσουμε λοιπόν ότι παράχθηκε η παρακάτω ακολουθία τυχαίων αριθμών:

0.822951	0.151932	0.625477	0.314685	0.346901
0.911720	0.519760	0.401154	0.606758	0.785402
0.031523	0.869921	0.166525	0.674520	0.758400
0.581893	0.389248	0.200232	0.355635	0.826927

Άρα τα άτομα v_2^* , v_{11}^* , v_{13}^* και v_{18}^* επιλέγονται για διασταύρωση. Στο σημείο αυτό σταθήκαμε τυχεροί, αφού ο αριθμός των ατόμων που επιλέχθηκαν είναι άρτιος, οπότε το ζευγάρωμά τους είναι εύκολο. Αν ο αριθμός τους ήταν περιττός, θα έπρεπε είτε να επιλέξουμε (τυχαία πάντα) ένα άτομο ακόμα από τον πληθυσμό είτε να απορρίψουμε κάποιο από τα ήδη επιλεγμένα. Στη συνέχεια ζευγαρώνουμε τα επιλεγμένα άτομα τυχαία: έστω λοιπόν ότι ζευγαρώνουν το v_2^* με το v_{11}^* και το v_{13}^* με το v_{18}^* . Για καθένα από τα δύο ζευγάρια παράγουμε τυχαία έναν ακέραιο αριθμό pos στο διάστημα $[1, 32]$ (το συνολικό μήκος, δηλαδή ο συνολικός αριθμός bits, κάθε χρωματοσώματος είναι 33). Ο αριθμός pos καθορίζει το σημείο διασταύρωσης (crossing point) του χρωματοσώματος.

Έστω λοιπόν ότι για το πρώτο ζευγάρι (v_2^* , v_{11}^*) επιλέχθηκε σημείο διασταύρωσης $pos = 9$:

$$\begin{aligned} v_2^* &= (100011000 \mid 101101001111000001110010) \\ v_{11}^* &= (111011101 \mid 101110000100011111011110) \end{aligned}$$

Αυτά τα χρωμοσώματα θα «κοπούν» μετά το 9ο bit και θα αντικατασταθούν από το ακόλουθο ζεύγος απογόνων:

$$\begin{aligned} v_2^{**} &= (100011000 \mid 101110000100011111011110) \\ v_{11}^{**} &= (111011101 \mid 101101001111000001110010) \end{aligned}$$

Έστω τώρα ότι για το δεύτερο ζευγάρι (v_{13}^* , v_{18}^*) επιλέχθηκε σημείο διασταύρωσης, $pos = 20$:

$$\begin{aligned} v_{13}^* &= (00010100001001010100 \mid 1010111111011) \\ v_{18}^* &= (11101111101000100011 \mid 0000001000110) \end{aligned}$$

Αυτά τα χρωμοσώματα θα «κοπούν» μετά το 20ό bit και θα αντικατασταθούν από το ακόλουθο ζεύγος απογόνων:

$$\begin{aligned} v_{13}^{**} &= (00010100001001010100 \mid 0000001000110) \\ v_{18}^{**} &= (11101111101000100011 \mid 1010111111011) \end{aligned}$$

Οπότε η τρέχουσα μορφή του πληθυσμού θα είναι ως εξής:

$$\begin{aligned} v_1^* &= (011001111110110101100001101111000) \\ v_2^{**} &= (100011000101110000100011111011110) \\ v_3^* &= (00100010000011010111101101111011) \\ v_4^* &= (011001111110110101100001101111000) \\ v_5^* &= (00010101001111111110000110001100) \\ v_6^* &= (100011000101101001111000001110010) \\ v_7^* &= (111011101101110000100011111011110) \\ v_8^* &= (00011101100101001101011111000101) \\ v_9^* &= (011001111110110101100001101111000) \\ v_{10}^* &= (000010000011001000001010111011101) \\ v_{11}^{**} &= (111011101101101001111000001110010) \\ v_{12}^* &= (010000000101100010110000001111100) \\ v_{13}^{**} &= (000101000010010101000000001000110) \\ v_{14}^* &= (100001100001110100010110101100111) \\ v_{15}^* &= (101110010110011110011000101111110) \\ v_{16}^* &= (100110100000001111111010011011111) \\ v_{17}^* &= (000001111000110000011010000111011) \\ v_{18}^{**} &= (11101111101000100011101011111011) \\ v_{19}^* &= (111011101101110000100011111011110) \\ v_{20}^* &= (110011110000011111100001101001011) \end{aligned}$$

Μετάλλαξη

Η μετάλλαξη πραγματοποιείται bit-by-bit: δηλαδή ο τελεστής της μετάλλαξης αντιμετωπίζει ολόκληρο τον πληθυσμό ως έναν συρμό από δυαδικά ψηφία, όπου κάθε ψηφίο έχει την ίδια πιθανότητα να μεταλλαχθεί. Έτσι —με δεδομένο ότι η πιθανότητα μετάλλαξης είναι $p_m = 0.01$ — αναμένουμε ότι κατά μέσο όρο το 1% όλων των binary bits του πληθυσμού θα αντιστραφούν. Ο πληθυσμός μας αποτελείται από $m \times pop_size = 33 \times 20 = 660$ δυαδικά ψηφία, άρα αναμένουμε κατά μέσο όρο 6.6 μεταλλάξεις σε κάθε γενιά. Για κάθε δυαδικό ψηφίο παράγουμε έναν τυχαίο πραγματικό αριθμό r στο διάστημα $[0,1]$. Εάν $r < 0.01$, τότε αντιστρέφουμε το δυαδικό ψηφίο.

Έτσι πρέπει να παράγουμε συνολικά 660 τυχαίους αριθμούς στο διάστημα $[0, 1]$. Σε ένα δοκιμαστικό τρέξιμο παράχθηκαν 5 αριθμοί μικρότεροι από 0.01. Οι αριθμοί αυτοί, οι θέσεις στον πληθυσμό των δυαδικών ψηφίων που επιλέχθηκαν, ο αριθμός του ατόμου που αντιστοιχούν, καθώς και η θέση τους στα αντίστοιχα άτομα φαίνονται στον παρακάτω πίνακα:

Τυχαίος Αριθμός	Θέση Ψηφίου στον Πληθυσμό	Αριθμός ατόμου (χρωμοσώματος)	Θέση Ψηφίου στο Χρωμόσωμα
0.000213	112	4	13
0.009945	349	11	19
0.008809	418	13	22
0.005425	429	13	33
0.002836	602	19	8

Αυτό σημαίνει ότι τέσσερα άτομα του πληθυσμού θα υποστούν μετάλλαξη — στο 13ο άτομο μεταλλάσσονται δύο ψηφία. Οπότε ο πληθυσμός που προκύπτει μετά και τη διαδικασία της μετάλλαξης είναι ο ακόλουθος (τα ψηφία που έχουν μεταλλαχθεί είναι bold):

v_1	=	(011001111110110101100001101111000)
v_2	=	(100011000101110000100011111011110)
v_3	=	(00100010000011010111101101111011)
v_4	=	(011001111110 0 10101100001101111000)
v_5	=	(000101010011111111110000110001100)
v_6	=	(100011000101101001111000001110010)
v_7	=	(111011101101110000100011111011110)
v_8	=	(00011101100101001101011111000101)
v_9	=	(011001111110110101100001101111000)
v_{10}	=	(000010000011001000001010111011101)
v_{11}	=	(111011101101101001 0 11000001110010)
v_{12}	=	(010000000101100010110000001111100)
v_{13}	=	(000101000010010101000 1 000010001 11)
v_{14}	=	(100001100001110100010110101100111)
v_{15}	=	(101110010110011110011000101111110)
v_{16}	=	(100110100000001111111010011011111)
v_{17}	=	(000001111000110000011010000111011)
v_{18}	=	(11101111101000100011101011111011)
v_{19}	=	(11101110 0 101110000100011111011110)
v_{20}	=	(110011110000011111100001101001011)

Μόλις τελειώσαμε λοιπόν την πρώτη γενεά του ΓΑ. Είναι ενδιαφέρον να δούμε την απόδοση των μελών του πληθυσμού που προέκυψε στο τέλος αυτής της πρώτης γενεάς.

eval(v ₁)	=	f(3.130078, 4.996097)	=	23.410669
eval(v ₂)	=	f(5.279042, 5.054515)	=	18.201083
eval(v ₃)	=	f(-0.991471, 5.680258)	=	16.020812
eval(v ₄)	=	f(3.128235, 4.996097)	=	23.412613
eval(v ₅)	=	f(-1.746635, 5.395584)	=	20.095903
eval(v ₆)	=	f(5.278638, 5.593460)	=	17.406725
eval(v ₇)	=	f(11.089025, 5.054515)	=	30.060205
eval(v ₈)	=	f(-1.255173, 4.734458)	=	25.341160
eval(v ₉)	=	f(3.130078, 4.996097)	=	23.410669
eval(v ₁₀)	=	f(-2.516603, 4.390381)	=	19.526329
eval(v ₁₁)	=	f(11.088621, 4.743434)	=	33.351874
eval(v ₁₂)	=	f(0.795406, 5.381472)	=	16.127799
eval(v ₁₃)	=	f(-1.811725, 4.209937)	=	22.692462
eval(v ₁₄)	=	f(4.910618, 4.703018)	=	17.959701
eval(v ₁₅)	=	f(7.935998, 4.757338)	=	13.666916
eval(v ₁₆)	=	f(6.084492, 5.652242)	=	26.019600
eval(v ₁₇)	=	f(-2.554851, 4.793707)	=	21.278435
eval(v ₁₈)	=	f(11.134646, 5.666976)	=	27.591064
eval(v ₁₉)	=	f(11.059532, 5.054515)	=	27.608441
eval(v ₂₀)	=	f(9.211598, 4.993762)	=	23.867227

Παρατηρούμε ότι η συνολική απόδοση F του νέου πληθυσμού είναι 447.049688, πολύ μεγαλύτερη από τη συνολική απόδοση του αρχικού πληθυσμού που ήταν 387.776822. Επίσης, το χρωμόσωμα (v₁₁) με την καλύτερη απόδοση (33.351874) στον τρέχοντα πληθυσμό έχει υψηλότερη απόδοση από το χρωμόσωμα (v₁₅) με την καλύτερη απόδοση (30.060205) στον αρχικό πληθυσμό.

Ο ΓΑ συνεχίζει με τη φάση της επιλογής και τα διάφορα βήματα (επιλογή-διασταύρωση-μετάλλαξη) επαναλαμβάνονται κυκλικά έως ότου ικανοποιηθεί το κριτήριο τερματισμού του αλγορίθμου. Συνήθως το κριτήριο τερματισμού ενός ΓΑ είναι είτε ένας συγκεκριμένος (μέγιστος) αριθμός γενεών είτε ένα συγκεκριμένο ποσοστό βελτίωσης του καλύτερου ατόμου ή του συνολικού πληθυσμού σε σχέση με κάποιον αριθμό προηγούμενων γενεών. Έτσι στον συγκεκριμένο ΓΑ, αν θεωρήσουμε ως συνθήκη τερματισμού τις 1000 γενεές, ο τελικός πληθυσμός (ο πληθυσμός στο τέλος της 1000 της γενεάς) θα έχει ως εξής:

v ₁	=	(111011110110011011100101010111011)
v ₂	=	(111001100110000100010101010111000)
v ₃	=	(111011110111011011100101010111011)
v ₄	=	(111001100010000110000101010111001)
v ₅	=	(111011110111011011100101010111011)
v ₆	=	(111001100110000100000100010100001)
v ₇	=	(110101100010010010001100010110000)
v ₈	=	(111101100010001010001101010010001)
v ₉	=	(111001100010010010001100010110001)
v ₁₀	=	(111011110111011011100101010111011)

v ₁₁	=	(110101100000010010001100010110000)
v ₁₂	=	(110101100010010010001100010110001)
v ₁₃	=	(111011110111011011100101010111011)
v ₁₄	=	(111001100110000100000101010111011)
v ₁₅	=	(1110011010101111001010100110110001)
v ₁₆	=	(111001100110000101000100010100001)
v ₁₇	=	(111001100110000100000101010111011)
v ₁₈	=	(111001100110000100000101010111001)
v ₁₉	=	(111101100010001010001110000010001)
v ₂₀	=	(111001100110000100000101010111001)

Οι αντίστοιχες αποδόσεις είναι:

eval(v ₁)	=	f(11.120940, 5.092514)	=	30.298543
eval(v ₂)	=	f(10.588756, 4.667358)	=	26.869724
eval(v ₃)	=	f(11.124627, 5.092514)	=	30.316575
eval(v ₄)	=	f(10.574125, 4.242410)	=	31.933120
eval(v ₅)	=	f(11.124627, 5.092514)	=	30.316575
eval(v ₆)	=	f(10.588756, 4.214603)	=	34.356125
eval(v ₇)	=	f(9.631066, 4.427881)	=	35.458636
eval(v ₈)	=	f(11.518106, 4.452835)	=	23.309078
eval(v ₉)	=	f(10.574816, 4.427933)	=	34.393820
eval(v ₁₀)	=	f(11.124627, 5.092514)	=	30.316575
eval(v ₁₁)	=	f(9.623693, 4.427881)	=	35.477938
eval(v ₁₂)	=	f(9.631066, 4.427933)	=	35.456066
eval(v ₁₃)	=	f(11.124627, 5.092514)	=	30.316575
eval(v ₁₄)	=	f(10.588756, 4.242514)	=	32.932098
eval(v ₁₅)	=	f(10.606555, 4.653714)	=	30.746768
eval(v ₁₆)	=	f(10.588814, 4.214603)	=	34.359545
eval(v ₁₇)	=	f(10.588756, 4.242514)	=	32.932098
eval(v ₁₈)	=	f(10.588756, 4.242410)	=	32.956664
eval(v ₁₉)	=	f(11.518106, 4.472757)	=	19.669670
eval(v ₂₀)	=	f(10.588756, 4.242410)	=	32.956664

Η συνολική απόδοση F του τελικού πληθυσμού είναι 625.372857, πολύ υψηλότερη από την απόδοση του πληθυσμού στο τέλος της πρώτης γενιάς (447.049688). Η απόδοση του καλύτερου μέλους του τελικού πληθυσμού είναι 35.477938.

Ασκήσεις

Απαντώντας στις παρακάτω ασκήσεις θα είστε σε θέση να γνωρίζετε αν έχετε εμπεδώσει τις έννοιες που παρουσιάστηκαν στην ενότητα 2. Στόχος τους είναι η εξοικείωση με την εφαρμογή των ΓΑ σε προβλήματα βελτιστοποίησης.

Άσκηση 8.2.1 (Επίπεδο Δυσκολίας 1)

Η τιμή της ακέραιας παραμέτρου ρ ορίζει το σημείο διασταύρωσης σε ένα χρωμόσωμα. Πώς επιλέγεται αυτή η τιμή;

Άσκηση 8.2.2 (Επίπεδο Δυσκολίας 1)

Για έναν πληθυσμό με μέγεθος 100 και μήκος χρωμοσώματος 50 ποιος είναι αναμενόμενος αριθμός των ψηφίων που θα μεταλλαχθούν, αν η πιθανότητα μετάλλαξης παίρνει τις τιμές 0.001, 0.01 και 0.1;

Άσκηση 8.2.3 (Επίπεδο Δυσκολίας 1)

Αν στο παράδειγμα που αναλύσαμε η επιθυμητή ακρίβεια ήταν 2 δεκαδικά ψηφία, να υπολογίσετε τον συνολικό αριθμό δυαδικών ψηφίων που θα έχει το χρωμόσωμα.

Άσκηση 8.2.4 (Επίπεδο Δυσκολίας 1)

Αν η πιθανότητα διασταύρωσης είναι 0.2 και το μέγεθος του πληθυσμού είναι 100, ποιος είναι ο αναμενόμενος αριθμός ατόμων που θα επιλεγούν για διασταύρωση; Τι πρέπει να κάνουμε αν αυτός ο αριθμός είναι περιττός;

Άσκηση 8.2.5 (Επίπεδο Δυσκολίας 1)

Αν κατά το τρέξιμο του αλγορίθμου, σε κάποια γενιά τα καλύτερα άτομα που θα προκύψουν δίνουν χειρότερες τιμές για την αντικειμενική συνάρτηση, σε σχέση με τις προηγούμενες, πρέπει να διακοπεί η εκτέλεση του αλγορίθμου;

Άσκηση 8.2.6 (Επίπεδο Δυσκολίας 1)

Όταν τελειώσει η εκτέλεση του αλγορίθμου (αφού ικανοποιηθεί η συνθήκη τερματισμού), πώς θα εξασφαλίσουμε ότι το καλύτερο χρωμόσωμα της τελευταίας γενιάς είναι το καλύτερο όλου του πληθυσμού;

8.3 Θεωρητική Ανάλυση των Γενετικών Αλγορίθμων

Η θεωρητική θεμελίωση των ΓΑ βασίζεται στην αναπαράσταση των δυνατών λύσεων ως δυαδικές συμβολοσειρές, καθώς και στην έννοια του *σχήματος (schema)* — μιας φόρμας (template) που επιτρέπει τον προσδιορισμό της ομοιότητας μεταξύ των χρωμοσωμάτων. Ένα σχήμα κατασκευάζεται εισάγοντας το λεγόμενο *αδιάφορο σύμβολο (don't care symbol)* * στο αλφάβητο $\Sigma (= \{0,1\})$ των γονιδίων. Ένα σχήμα αναπαριστά όλες τις συμβολοσειρές (ένα υπερεπίπεδο ή άλλο υποσύνολο του χώρου αναζήτησης), οι οποίες ταιριάζουν σε όλες τις θέσεις εκτός από αυτές με το αδιάφορο σύμβολο *.

Ας θεωρήσουμε π.χ. τις συμβολοσειρές και τα σχήματα μήκους 10. Στο σχήμα (*111100100) ταιριάζουν οι δύο συμβολοσειρές:

{0111100100, 1111100100}

ενώ στο σχήμα (*1*1100100) ταιριάζουν οι τέσσερις συμβολοσειρές:

{0101100100, 0111100100, 1101100100, 1111100100}

Φυσικά το σχήμα (1001110001) αναπαριστά μία μόνο συμβολοσειρά, την (1001110001), ενώ το σχήμα (******) αναπαριστά όλες τις συμβολοσειρές μήκους 10. Είναι προφανές ότι κάθε σχήμα αναπαριστά 2^r συμβολοσειρές, όπου r είναι ο αριθμός των αδιάφορων συμβόλων * στο σχήμα (Γιατί;). Από την άλλη πλευρά, κάθε συμβολοσειρά μήκους m ταιριάζει σε 2^m διαφορετικά σχήματα (Γιατί;). Π.χ. ας θεωρήσουμε τη συμβολοσειρά (1001110001). Η συμβολοσειρά αυτή ταιριάζει στα ακόλουθα 2^{10} σχήματα:

(1001110001)
(*001110001)
(1*01110001)
(10*1110001)
•
•
•

(100111000*)
(**01110001)
(*0*1110001)
(10**110001)
•
•
•
(10011100**)
(***1110001)
•
•
•
(***** ***)

Διαφορετικά σχήματα έχουν και διαφορετικά χαρακτηριστικά. Θα πρέπει να έχει ήδη γίνει σαφές ότι ο αριθμός των αδιάφορων συμβόλων * σε ένα σχήμα καθορίζει τον αριθμό των συμβολοσειρών που ταιριάζουν σε αυτό. Υπάρχουν δύο σημαντικά μεγέθη που χαρακτηρίζουν τα σχήματα: η τάξη (order) και το ορίζον μήκος (defining length). Το Αποτέλεσμα των Σχημάτων (Schema Result) θα διατυπωθεί με βάση τα μεγέθη αυτά.

Η τάξη ενός σχήματος S, η οποία συμβολίζεται ο(S), είναι ο αριθμός των σταθερών θέσεων του σχήματος, δηλαδή των θέσεων που δεν περιέχουν το αδιάφορο σύμβολο * — θέσεις με 0 ή 1· με άλλα λόγια είναι το μήκος του σχήματος μείον τον αριθμό των αδιάφορων συμβόλων *. Η τάξη προσδιορίζει την ειδικότητα (specialty) ενός σχήματος, δηλαδή το πόσο ειδικό είναι το συγκεκριμένο σχήμα. Π.χ. τα ακόλουθα τρία σχήματα, όλα μήκους 10,

$$S_1 = (**001*110),$$

$$S_2 = (****00**0*),$$

$$S_3 = (11101**001),$$

έχουν τις ακόλουθες τάξεις:

$$o(S_1) = 6, o(S_2) = 3 \text{ και } o(S_3) = 8,$$

και το σχήμα S₃ είναι το πιο συγκεκριμένο (ειδικό-special) ή, με άλλα λόγια, το λιγότερο γενικό, αφού αναπαριστά μόνο τέσσερις συμβολοσειρές, σε αντίθεση με τα S₁ και S₂ που αναπαριστούν 16 και 128 συμβολοσειρές αντίστοιχα. Η έννοια της τάξης ενός σχήματος είναι χρήσιμη στον υπολογισμό της πιθανότητας επιβίωσης του σχήματος κατά τη διαδικασία της μετάλλαξης.

Το ορίζον μήκος ενός σχήματος S (συμβολίζεται δ(S)) είναι η απόσταση μεταξύ της πρώτης και της τελευταίας σταθερής θέσης. Προσδιορίζει την πυκνότητα (compactness) της πληροφορίας που περιέχεται στο σχήμα. Έτσι για τα προηγούμενα σχήματα έχουμε:

$$\delta(S_1) = 10 - 4 = 6, \delta(S_2) = 9 - 5 = 4 \text{ και } \delta(S_3) = 10 - 1 = 9.$$

Ένα σχήμα με μια μοναδική σταθερή θέση έχει ορίζον μήκος μηδέν. Η έννοια του ορίζοντος μήκους ενός σχήματος είναι χρήσιμη στον υπολογισμό της πιθανότητας επιβίωσής του κατά τη διαδικασία της διασταύρωσης.

Ως γνωστόν η διαδικασία εξέλιξης ενός ΓΑ αποτελείται από τέσσερα επαναλαμβανόμενα βήματα:

- 1) $t \leftarrow t+1$
- 2) επέλεξε νέο (προσωρινό) πληθυσμό P(t) από τον P(t-1)
- 3) ανασυνδύασε τον P(t)
- 4) εκτίμησε τον P(t)

Το πρώτο βήμα ($t \leftarrow t+1$) απλά αυξάνει το «ρολόι» της διαδικασίας κατά ένα (δηλαδή η διαδικασία προχωρά στην επόμενη γενιά). Στο τελευταίο βήμα (εκτίμησε τον $P(t)$) γίνεται η εκτίμηση του τρέχοντος πληθυσμού. Τα σημαντικότερα βήματα της εξελικτικής διαδικασίας είναι τα υπόλοιπα δύο: επιλογή και ανασυνδυασμός. Ακολουθεί μια συζήτηση σχετικά με τις επιδράσεις των δύο αυτών βημάτων στον αριθμό και το είδος των σχημάτων που περιέχονται στον πληθυσμό. Η συζήτηση θα γίνει με βάση ένα παράδειγμα.

Ας υποθέσουμε ότι ο πληθυσμός έχει μέγεθος $\text{pop_size} = 20$ και ότι το μήκος της συμβολοσειράς (επομένως και το μήκος του σχήματος) είναι $m = 33$ (όπως και στο παράδειγμα του προηγούμενου κεφαλαίου). Ακόμη ας υποθέσουμε ότι τη στιγμή (ή βήμα ή γενιά) t ο πληθυσμός αποτελείται από τις ακόλουθες συμβολοσειρές:

```

v1 = (100110100000001111111010011011111)
v2 = (111000100100110111001010100011010)
v3 = (000010000011001000001010111011101)
v4 = (100011000101101001111000001110010)
v5 = (000111011001010011010111111000101)
v6 = (00010100001001010100101011111011)
v7 = (00100010000011010111101101111011)
v8 = (100001100001110100010110101100111)
v9 = (010000000101100010110000001111100)
v10 = (000001111000110000011010000111011)
v11 = (011001111110110101100001101111000)
v12 = (110100010111101101000101010000000)
v13 = (111011111010001000110000001000110)
v14 = (010010011000001010100111100101001)
v15 = (111011101101110000100011111011110)
v16 = (110011110000011111100001101001011)
v17 = (011010111111001111010001101111101)
v18 = (011101000000001110100111110101101)
v19 = (000101010011111111110000110001100)
v20 = (101110010110011110011000101111110)

```

Έστω $\xi(S,t)$ είναι ο αριθμός των συμβολοσειρών στον πληθυσμό τη στιγμή t που ταιριάζουν στο σχήμα S . Π.χ. για το σχήμα

$$S_0 = (****111*****),$$

είναι $\xi(S_0,t) = 3$, αφού υπάρχουν τρεις συμβολοσειρές (οι v_{13}, v_{15} και v_{16}), οι οποίες ταιριάζουν με το σχήμα S_0 . Η τάξη του S_0 είναι $o(S_0) = 3$ και το ορίζον μήκος του είναι $\delta(S_0) = 7 - 5 = 2$.

Μια άλλη ιδιότητα ενός σχήματος είναι η απόδοση (fitness) τη στιγμή t ή $\text{eval}(S,t)$ — ορίζεται ως η μέση απόδοση όλων των συμβολοσειρών του πληθυσμού τη στιγμή t που ταιριάζουν με το σχήμα S . Έστω ότι υπάρχουν p συμβολοσειρές $\{v_{i1}, \dots, v_{ip}\}$ στον πληθυσμό που ταιριάζουν τη στιγμή t με το σχήμα S . Τότε

$$\text{eval}(S,t) = \left(\sum_{j=1}^p \text{eval}(v_{ij}) \right) / p.$$

Κατά τη διάρκεια της επιλογής δημιουργείται ένας προσωρινός πληθυσμός. Κάθε συμβολοσειρά αντιγράφεται μηδέν, μία ή περισσότερες φορές, σύμφωνα με την απόδοσή της. Όπως είδαμε στην προηγούμενη ενότητα, σε μια απλή επιλογή συμβολοσειράς, η συμβολοσειρά v_i επιλέγεται με πιθανότητα $p_i = \text{eval}(v_i)/F(t)$, όπου $F(t)$ είναι το άθροισμα των αποδόσεων ολόκληρου του πληθυσμού.

Μετά το βήμα της επιλογής αναμένεται ότι $\xi(S,t+1)$ συμβολοσειρές θα ταιριάζουν με το σχήμα S. Επειδή:

- 1) για μια συμβολοσειρά που ταιριάζει με το σχήμα S, η πιθανότητα επιλογής της είναι $eval(S,t)/F(t)$,
- 2) ο αριθμός των συμβολοσειρών που ταιριάζουν με το σχήμα S είναι $\xi(S,t)$ και
- 3) ο αριθμός των επιλογών σε κάθε βήμα είναι pop_size ,

θα πρέπει να είναι σαφές ότι:

$$\xi(S,t+1) = \xi(S,t) \cdot pop_size \cdot eval(S,t) / F(t)$$

Αν λάβουμε υπόψη ότι η μέση απόδοση του πληθυσμού είναι $\overline{F(t)} = F(t) / pop_size$, η παραπάνω σχέση ισοδυναμεί με την ακόλουθη:

$$\xi(S,t+1) = \xi(S,t) \cdot eval(S,t) / \overline{F(t)}$$

Με άλλα λόγια, ο αριθμός των συμβολοσειρών στον πληθυσμό αυξάνεται ανάλογα με τον λόγο της απόδοσης του αντίστοιχου σχήματος προς τη μέση απόδοση του πληθυσμού. Αυτό σημαίνει ότι ένα σχήμα που βρίσκεται πάνω από τον μέσο όρο, όσον αφορά την απόδοση, αποκτά μεγαλύτερο αριθμό συμβολοσειρών που ταιριάζουν με αυτό στην επόμενη γενιά. Αντίθετα, ένα σχήμα που βρίσκεται κάτω από τον μέσο όρο αναπαριστά λιγότερες συμβολοσειρές στην επόμενη γενιά.

Η μακροπρόθεσμη επίδραση της παραπάνω διαπίστωσης είναι η εξής: Αν υποθέσουμε ότι ένα σχήμα S βρίσκεται πάνω από τον μέσο όρο κατά $\varepsilon\%$ (δηλαδή $eval(S,t) = \overline{F(t)} + \varepsilon \cdot \overline{F(t)}$), τότε:

$$\xi(S,t) = \xi(S,0) \cdot (1 + \varepsilon)^t \text{ και}$$

$$\varepsilon = (eval(S,t) - \overline{F(t)}) / \overline{F(t)}$$

με $\varepsilon > 0$ για σχήματα πάνω από τον μέσο όρο και $\varepsilon < 0$ για σχήματα κάτω από τον μέσο όρο.

Η παραπάνω σχέση είναι μια εξίσωση γεωμετρικής προόδου. Επομένως, ένα σχήμα πάνω από τον μέσο όρο όχι μόνο αναπαριστά περισσότερες συμβολοσειρές στην επόμενη γενιά, αλλά επιπλέον ο αριθμός αυτός αυξάνεται εκθετικά.

Ας επιστρέψουμε στο παράδειγμα και συγκεκριμένα στο σχήμα S_0 . Αφού υπάρχουν τρεις συμβολοσειρές τη στιγμή t που ταιριάζουν με το σχήμα S_0 , η απόδοση του σχήματος αυτού είναι:

$$eval(S_0,t) = (27.316702 + 30.060205 + 23.867227) / 3 = 27.081378.$$

Την ίδια στιγμή η μέση απόδοση του πληθυσμού είναι:

$$\overline{F(t)} = \left(\sum_{i=1}^{20} eval(v_i) \right) / pop_size = 387.776822 / 20 = 19.388841$$

και ο λόγος της απόδοσης του S_0 προς τη μέση απόδοση του πληθυσμού είναι:

$$eval(S_0,t) / \overline{F(t)} = 1.396751.$$

Παρατηρούμε ότι το σχήμα S_0 βρίσκεται πάνω από τον μέσο όρο όσον αφορά την απόδοση και στις επόμενες γενιές αναπαριστά ένα εκθετικά αυξανόμενο αριθμό από συμβολοσειρές. Πιο συγκεκριμένα, αν τη στιγμή t το σχήμα S_0 βρίσκεται πάνω από τον μέσο όρο κατά έναν συντελεστή 1.1396751, τότε τη στιγμή t+1 αναμένουμε το σχήμα να αναπαριστά $3 \times 1.396751 = 4.19$ συμβολοσειρές (πιθανότατα 4 ή 5), τη στιγμή t+2: $3 \times 1.396751^2 = 5.85$ συμβολοσειρές (πιθανότατα 5 ή 6) κοκ.

Διαισθητικά το σχήμα S_0 αποτελεί ένα υποσχόμενο τμήμα του χώρου αναζήτησης και, για τον λόγο αυτό, δειγματοληπτείται με εκθετικά αυξανόμενο τρόπο.

Ας επιστρέψουμε και πάλι στο παράδειγμά μας. Τη στιγμή t το σχήμα S_0 αναπαριστά τρεις συμβολοσειρές. Στην προηγούμενη ενότητα η εξομοίωση της εξελικτικής διαδικασίας με τον ίδιο πληθυσμό οδήγησε στον ακόλουθο πληθυσμό:

v_1^*	=	(011001111110110101100001101111000)	(v_{11})
v_2^*	=	(100011000101101001111000001110010)	(v_4)
v_3^*	=	(00100010000011010111101101111011)	(v_7)
v_4^*	=	(011001111110110101100001101111000)	(v_{11})
v_5^*	=	(000101010011111111110000110001100)	(v_{19})
v_6^*	=	(100011000101101001111000001110010)	(v_4)
v_7^*	=	(111011101101110000100011111011110)	(v_{15})
v_8^*	=	(000111011001010011010111111000101)	(v_5)
v_9^*	=	(011001111110110101100001101111000)	(v_{11})
v_{10}^*	=	(000010000011001000001010111011101)	(v_3)
v_{11}^*	=	(111011101101110000100011111011110)	(v_{15})
v_{12}^*	=	(010000000101100010110000001111100)	(v_9)
v_{13}^*	=	(00010100001001010100101011111011)	(v_6)
v_{14}^*	=	(100001100001110100010110101100111)	(v_8)
v_{15}^*	=	(101110010110011110011000101111110)	(v_{20})
v_{16}^*	=	(100110100000001111111010011011111)	(v_1)
v_{17}^*	=	(000001111000110000011010000111011)	(v_{10})
v_{18}^*	=	(111011111010001000110000001000110)	(v_{13})
v_{19}^*	=	(111011101101110000100011111011110)	(v_{15})
v_{20}^*	=	(110011110000011111100001101001011)	(v_{16})

Πράγματι το σχήμα S_0 αναπαριστά πέντε συμβολοσειρές στον νέο πληθυσμό: v_7^* , v_{11}^* , v_{18}^* , v_{19}^* και v_{20}^* .

Παρ' όλα αυτά, η διαδικασία της επιλογής από μόνη της δεν εισάγει νέα σημεία του χώρου των λύσεων (νέες πιθανές λύσεις) στον πληθυσμό. Απλά αντιγράφει κάποιες συμβολοσειρές για τον σχηματισμό ενός προσωρινού πληθυσμού. Το δεύτερο βήμα του κύκλου εξέλιξης, ο ανασυνδυασμός, είναι υπεύθυνο για την εισαγωγή νέων ατόμων στον πληθυσμό. Αυτό γίνεται με τη χρήση των γενετικών τελεστών, τη διασταύρωση και τη μετάλλαξη. Στη συνέχεια θα μελετήσουμε χωριστά την επίδραση των δύο αυτών τελεστών στον αναμενόμενο αριθμό των σχημάτων στον πληθυσμό.

Μια οποιαδήποτε συμβολοσειρά του πληθυσμού, π.χ. η v_{18}^* :

$$(111011111010001000110000001000110)$$

ταιριάζει σε 2^{33} διαφορετικά σχήματα. Έστω τα ακόλουθα δύο σχήματα, στα οποία ταιριάζει η συγκεκριμένη συμβολοσειρά:

$$S_0 = (****111*****)$$

$$S_1 = (111*****10)$$

Ας υποθέσουμε ότι η συμβολοσειρά αυτή επιλέχθηκε για διασταύρωση (όπως συνέβη στην προηγούμενη ενότητα). Ας υποθέσουμε επίσης ότι το σημείο διασταύρωσης είναι $pos = 20$. Είναι σαφές ότι το σχήμα S_0 επιβιώνει από αυτήν τη διασταύρωση, δηλαδή ένας από τους απογόνους ταιριάζει στο S_0 . Αυτό το σημείο διασταύρωσης διατηρεί την ακολουθία 111 στην πέμπτη, έκτη και έβδομη θέση σε ένα από παιδιά, π.χ. το ζευγάρι:

$$v_{13}^* = (00010100001001010100 \mid 101011111011)$$

$$v_{18}^* = (1110\mathbf{111}1101000100011 \mid 0000001000110)$$

θα έδινε:

$$v_{13}^{**} = (00010100001001010100 \mid 0000001000110)$$

$$v_{18}^{**} = (1110\mathbf{111}1101000100011 \mid 1010111111011)$$

Αντίθετα το σχήμα S_1 καταστρέφεται, αφού κανείς από τους απογόνους δεν ταιριάζει με αυτό. Ο λόγος είναι ότι η ακολουθία 111 στην αρχή και η ακολουθία 10 στο τέλος του σχήματος τοποθετούνται σε διαφορετικούς απογόνους.

Από την παραπάνω συζήτηση θα πρέπει να έχει γίνει σαφές ότι το ορίζον μήκος ενός σχήματος παίζει καθοριστικό ρόλο για την επιβίωση ή την καταστροφή του. Στο παραπάνω παράδειγμα το ορίζον μήκος του σχήματος S_0 είναι $\delta(S_0) = 2$, ενώ του S_1 είναι $\delta(S_1) = 32$.

Γενικά το σημείο διασταύρωσης επιλέγεται ομοιόμορφα (uniformly) από $m-1$ πιθανά σημεία. Αυτό σημαίνει ότι η πιθανότητα καταστροφής ενός σχήματος S είναι:

$$p_d(S) = \frac{\delta(S)}{m-1}$$

και συνεπώς η πιθανότητα επιβίωσής του είναι:

$$p_s(S) = 1 - \frac{\delta(S)}{m-1}$$

Στο παράδειγμά μας οι συγκεκριμένες πιθανότητες για τα σχήματα S_0 και S_1 είναι:

$$p_d(S_0) = 2/32, p_s(S_0) = 30/32, p_d(S_1) = 0, p_s(S_1) = 32/32,$$

οπότε το αποτέλεσμα της διασταύρωσης ήταν αναμενόμενο.

Είναι σημαντικό να κατανοηθεί ότι μόνο μερικά χρωμοσώματα επιλέγονται για διασταύρωση, αφού η διασταύρωση έχει μια πιθανότητα p_c να εκτελεστεί. Άρα η πιθανότητα επιβίωσης ενός σχήματος είναι στην πραγματικότητα:

$$p_s(S) = 1 - p_c \cdot \frac{\delta(S)}{m-1}$$

Επιστρέφοντας στο παράδειγμά μας ισχύει ($p_c = 0.25$):

$$p_s(S_0) = 1 - 0.25 \cdot (2/32) = 63/64 = 0.984375.$$

Θα πρέπει επίσης να σημειωθεί ότι, έστω και αν το σημείο διασταύρωσης επιλεγεί ανάμεσα σε σταθερές θέσεις σε ένα σχήμα, υπάρχει ακόμα πιθανότητα για το σχήμα να επιβιώσει. Π.χ. αν και οι δύο συμβολοσειρές v_{13}^* και v_{18}^* άρχιζαν με 111 και τελείωναν με 10, το σχήμα S_1 θα επιβίωνε. Επομένως, η πιθανότητα επιβίωσης ενός σχήματος είναι:

$$p_s(S) \geq 1 - p_c \cdot \frac{\delta(S)}{m-1}$$

Συνεπώς η επίδραση της επιλογής και της διασταύρωσης στην αύξηση του αριθμού των συμβολοσειρών που ταιριάζουν σε ένα σχήμα είναι:

$$\xi(S, t+1) \geq \xi(S, t) \cdot \text{eval}(S, t) / \overline{F(t)} \cdot \left[1 - p_c \cdot \frac{\delta(S)}{m-1} \right]$$

Η παραπάνω σχέση προσδιορίζει τον αναμενόμενο αριθμό των συμβολοσειρών που θα ταιριάζουν με ένα σχήμα στην επόμενη γενιά συναρτήσεως του τρέχοντα αριθμού των συμβολοσειρών που ταιριάζουν με το σχήμα,

τη σχετική απόδοσή του και το ορισμένο μήκος του. Όπως φαίνεται, τα άνω του μέσου όρου σχήματα με μικρό ορισμένο μήκος θα δειγματοληπτούνται με εκθετικά αυξανόμενους ρυθμούς στις επόμενες γενιές.

Για το σχήμα S_0 ισχύει:

$$eval(S_0, t) / \overline{F(t)} \cdot \left[1 - p_c \cdot \frac{\delta(S_0)}{m-1} \right] = 1.396751 \cdot 0.984375 = 1.374927$$

δηλαδή το —άνω του μέσου όρου και με μικρό ορισμένο μήκος— σχήμα S_0 θα αποκτήσει εκθετικά αυξανόμενο αριθμό συμβολοσειρών στις επόμενες γενιές. Στη γενιά $t+1$ αναμένουμε $3 \times 1.374927 = 4.12$ συμβολοσειρές και στη γενιά $t+2$, $3 \times 1.374927^2 = 5.67$ συμβολοσειρές.

Ο τελεστής μετάλλαξης αντιστρέφει ένα δυαδικό ψηφίο σε κάποια τυχαία θέση με πιθανότητα p_m . Είναι φανερό ότι για να επιβιώσει κάποιο σχήμα θα πρέπει να παραμείνουν αμετάβλητες οι σταθερές θέσεις του μετά τη μετάλλαξη. Ας πάρουμε π.χ. τη συμβολοσειρά v_{19}^* :

$$(111011101101110000100011111011110)$$

και το σχήμα S_0 :

$$S_0 = (****111*****).$$

Ας υποθέσουμε ακόμα ότι η συμβολοσειρά v_{19}^* υπόκειται σε μετάλλαξη. Στο παράδειγμα της προηγούμενης ενότητας, η v_{19}^* μεταλλάχθηκε στην ένατη θέση και προέκυψε η:

$$v_{19}^{**} = (111011100101110000100011111011110),$$

η οποία ταιριάζει με το σχήμα S_0 . Εάν είχε επιλεγεί κάποια θέση στο διάστημα 1 έως 4 ή 8 έως 33, ο απόγονος που θα προέκυπτε θα ταίριαζε επίσης με το S_0 . Μόνο 3 δυαδικά ψηφία (οι σταθερές θέσεις — πέμπτη, έκτη και έβδομη) είναι «σημαντικά»: μετάλλαξη σε ένα τουλάχιστον από αυτά θα κατέστρεφε το σχήμα. Ο αριθμός αυτών των «σημαντικών» ψηφίων είναι, όπως είπαμε, η τάξη του σχήματος.

Αφού η πιθανότητα αντιστροφής ενός δυαδικού ψηφίου είναι p_m , η πιθανότητα μη αλλαγής του είναι $1 - p_m$. Οι μεταλλάξεις είναι ανεξάρτητες μεταξύ τους, οπότε η πιθανότητα επιβίωσης ενός σχήματος κατά την όλη διαδικασία της μετάλλαξης (ακολουθία μεταλλάξεων δυαδικών ψηφίων) είναι:

$$p_s(S) = (1 - p_m)^{o(S)}$$

Επειδή όμως $p_m \ll 1$, η πιθανότητα αυτή προσεγγίζεται από την:

$$p_s(S) \approx 1 - o(S) \cdot p_m$$

Αναφερόμενοι και πάλι στο παράδειγμά μας με το σχήμα S_0 και θεωρώντας $p_m = 0.01$, έχουμε:

$$p_s(S_0) \approx 1 - 3 \cdot 0.01 = 0.97$$

Επομένως, ο συνδυασμός των αποτελεσμάτων μας για την επιλογή, τη διασταύρωση και τη μετάλλαξη οδηγούν στην ακόλουθη σχέση:

$$\xi(S, t+1) \geq \xi(S, t) \cdot eval(S, t) / \overline{F(t)} \left[1 - p_c \cdot \frac{\delta(S)}{m-1} - o(S) \cdot p_m \right]$$

Η σχέση αυτή περιγράφει την εκθετική αύξηση στις επόμενες γενιές των συμβολοσειρών που αντιστοιχούν σε κάποιο άνω του μέσου όρου (από πλευράς απόδοσης) σχήμα, με μικρό ορισμένο μήκος και μικρή τάξη.

Για το σχήμα S_0 ισχύει:

$$eval(S_0, t) / \overline{F(t)} \left[1 - p_c \cdot \frac{\delta(S)}{m-1} - o(S) \cdot p_m \right] = 1.396751 - 0.954375 = 1.333024$$

δηλαδή το σχήμα S_0 (το οποίο όπως έχουμε πει είναι πάνω από τον μέσο όρο απόδοσης, με μικρό ορισμένο μήκος και μικρή τάξη) θα λάβει εκθετικά περισσότερες συμβολοσειρές στις επόμενες γενιές: στη γενιά $t+1$ αναμένουμε $3 \times 1.333024 = 4.00$ συμβολοσειρές να ταιριάζουν με το S_0 , ενώ στη γενιά $t+2$ αναμένουμε $3 \times 1.333024^2 = 5.33$ τέτοιες συμβολοσειρές.

Η παραπάνω ανάλυση και το αποτέλεσμα που περιγράφεται από τη σχέση:

$$\xi(S, t+1) \geq \xi(S, t) \cdot eval(S, t) / \overline{F(t)} \left[1 - p_c \cdot \frac{\delta(S)}{m-1} - o(S) \cdot p_m \right]$$

μπορεί να διατυπωθεί από το ακόλουθο θεώρημα (γνωστό ως *Θεώρημα Σχημάτων*):

Θεώρημα Σχημάτων (Schema Theorem): Σχήματα με απόδοση άνω του μέσου όρου και με μικρό ορισμένο μήκος και μικρή τάξη λαμβάνουν εκθετικά αυξανόμενες συμβολοσειρές σε διαδοχικές γενιές ενός Γενετικού Αλγορίθμου.

Η παραπάνω θεωρία είναι καθαρά εμπειρική και δεν ενέχει κάποια φορμαλιστική μαθηματική ανάλυση της συμπεριφοράς των ΓΑ. Στην πραγματικότητα οι ΓΑ δεν έχουν ακόμη αναλυθεί μαθηματικά και αυτό είναι το μεγαλύτερό τους μειονέκτημα. Παρουσιάζουν υψηλή αποδοτικότητα σε ένα ευρύ φάσμα προβλημάτων, αλλά η έλλειψη πλήρους μαθηματικής επεξήγησης των λειτουργιών τους συνεπάγεται την αδυναμία επεξήγησης πολλών στοιχείων της συμπεριφοράς τους και, πιθανώς, την ανικανότητα βελτιστοποίησής τους.

Ασκήσεις

Απαντώντας στις παρακάτω ασκήσεις θα είστε σε θέση να γνωρίζετε αν έχετε εμπεδώσει τις έννοιες που παρουσιάστηκαν στην ενότητα 3. Στόχος τους είναι η εξοικείωση με τις έννοιες του σχήματος, της τάξης σχήματος, του ορίζοντος μήκους σχήματος και της πιθανότητας επιβίωσης ενός σχήματος μετά την εφαρμογή των γενετικών τελεστών, καθώς και με το θεώρημα σχημάτων.

Άσκηση 8.3.1 (Επίπεδο Δυσκολίας 2)

Να αποδείξετε ότι οποιαδήποτε δυαδική συμβολοσειρά μήκους m είναι στιγμιότυπο από $2m$ διαφορετικά σχήματα.

Άσκηση 8.3.2 (Επίπεδο Δυσκολίας 2)

Να αποδείξετε ότι για ένα σχήμα με $\varepsilon > 0$, ο αριθμός των συμβολοσειρών που αναπαριστούν στις επόμενες γενιές αυξάνεται εκθετικά. Το αντίθετο συμβαίνει όταν $\varepsilon < 0$.

Άσκηση 8.3.3 (Επίπεδο Δυσκολίας 2)

Να ορίσετε την αντικειμενική συνάρτηση της δυαδικής συμβολοσειράς x έτσι ώστε να ισούται με τον αριθμό των άσων που περιέχονται στη x . Να δώσετε τη σχέση για τη μέση απόδοση ενός σχήματος S , το οποίο έχει k ορισμένα δυαδικά ψηφία όλα ίσα με 1, σε σχέση με το μήκος της δυαδικής συμβολοσειράς ℓ και το k .

Άσκηση 8.3.4 (Επίπεδο Δυσκολίας 1)

Θεωρήστε τις τρεις δυαδικές συμβολοσειρές $A_1=11101111$, $A_2=00010100$ και $A_3=01000011$ και τα έξι σχήματα $H_1=1*****$, $H_2=0*****$, $H_3=*****11$, $H_4=***0*00*$, $H_5=1*****1*$, και $H_6=1110**1*$.

- Ποιες συμβολοσειρές ταιριάζουν σε κάθε σχήμα;
- Βρείτε την τάξη και το καθορισμένο μήκος κάθε σχήματος.
- Υπολογίστε την πιθανότητα επιβίωσης κάθε σχήματος, όταν υποβάλλεται σε μετάλλαξη και η πιθανότητα μιας απλής μετάλλαξης είναι $p_m=0.001$.
- Υπολογίστε την πιθανότητα επιβίωσης κάθε σχήματος, όταν υποβάλλεται σε διασταύρωση και η πιθανότητα διασταύρωσης είναι $p_c=0.85$.

Άσκηση 8.3.5 (Επίπεδο Δυσκολίας 1)

Ένας πληθυσμός περιλαμβάνει στη γενιά 0 τα ακόλουθα χρωμοσώματα και τις αντίστοιχες αποδόσεις τους:

A/A	Δυαδική συμβολοσειρά	Απόδοση
1	10001	20
2	11100	10
3	00011	5
4	01110	15

Η πιθανότητα μετάλλαξης είναι $p_m=0.01$ και η πιθανότητα διασταύρωσης είναι $p_c=1.0$.

- α) Υπολογίστε τον αναμενόμενο αριθμό σχημάτων της μορφής $S_1=(1****)$ που θα υπάρχουν στη γενιά 1.
β) Υπολογίστε τον αναμενόμενο αριθμό σχημάτων της μορφής $S_2=(0**1*)$ που θα υπάρχουν στη γενιά 1.

8.4 Συμπεράσματα

Στο κεφάλαιο αυτό έγινε μια ολοκληρωμένη εισαγωγική παρουσίαση στους Γενετικούς Αλγορίθμους. Ο αναγνώστης έχει πλέον τις βάσεις για να μελετήσει οποιαδήποτε εργασία σχετική με τους ΓΑ. Στις προηγούμενες σελίδες αναπτύχθηκαν εισαγωγικά θέματα που έχουν να κάνουν με την περιγραφή ενός απλού αλλά πολύ αποτελεσματικού ΓΑ, την εφαρμογή του σε ένα πρόβλημα βελτιστοποίησης, καθώς επίσης δόθηκαν και κάποιες θεωρητικές θεμελιώσεις των ΓΑ.

Φυσικά όσα ειπώθηκαν έως τώρα είναι βασικές εισαγωγικές έννοιες. Ο χώρος των ΓΑ και γενικότερα του Εξελικτικού Προγραμματισμού αναπτύσσεται με πολύ γοργούς ρυθμούς και προκειμένου ο ενδιαφερόμενος αναγνώστης να εμβαθύνει σε αυτόν θα πρέπει να καταφύγει στη διεθνή βιβλιογραφία, η οποία τα τελευταία χρόνια είναι ιδιαίτερα πλούσια. Μερικά πολύ καλά συγγράμματα για περαιτέρω μελέτη προτείνονται στο τέλος του κεφαλαίου.

Βιβλιογραφία

- Davis, L. (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley Publishing Company, Inc.
- Holland, J. (1992). *Adaptation in Natural and Artificial Systems*. MIT Press.
- Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs* (2η έκδ.). Springer - Verlag.
- Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. MIT Press.

Απαντήσεις ασκήσεων

Άσκηση 8.1.1 (Επίπεδο δυσκολίας 1): Δώστε με τη μορφή διαγράμματος ροής τη μορφή ενός απλού Γενετικού Αλγορίθμου.

Απάντηση:

Από την ενότητα 1 διαβάστε την περιγραφή ενός απλού Γενετικού Αλγορίθμου και αποδώστε τη διαγραμματικά.

Άσκηση 8.1.2 (Επίπεδο δυσκολίας 1): Ποια είναι τα πέντε βασικά συστατικά ενός Γενετικού Αλγορίθμου;

Απάντηση:

1. Μια γενετική αναπαράσταση των πιθανών λύσεων του προβλήματος.
2. Έναν τρόπο δημιουργίας ενός αρχικού πληθυσμού από πιθανές λύσεις (αρχικοποίηση).
3. Μια αντικειμενική συνάρτηση αξιολόγησης των μελών του πληθυσμού, που παίζει τον ρόλο του περιβάλλοντος.
4. Γενετικούς τελεστές για τη δημιουργία νέων μελών (λύσεων).
5. Τιμές για τις διάφορες παραμέτρους που χρησιμοποιεί ο ΓΑ (μέγεθος πληθυσμού, πιθανότητες εφαρμογής των γενετικών τελεστών κτλ).

Άσκηση 8.1.3 (Επίπεδο δυσκολίας 1): Αναφέρετε μερικά από τα πλεονεκτήματα των Γενετικών Αλγορίθμων που κάνουν ιδιαίτερα ελκυστική τη χρήση τους σε διάφορα προβλήματα.

Απάντηση:

1. Μπορούν να επιλύουν δύσκολα προβλήματα γρήγορα και αξιόπιστα.
2. Μπορούν εύκολα να συνεργαστούν με τα υπάρχοντα μοντέλα και συστήματα.
3. Είναι εύκολα επεκτάσιμοι και εξελίξιμοι.
4. Μπορούν να συμμετέχουν σε υβριδικές μορφές με άλλες μεθόδους.
5. Εφαρμόζονται σε πολύ περισσότερα πεδία από κάθε άλλη μέθοδο.
6. Δεν απαιτούν περιορισμούς στις συναρτήσεις που επεξεργάζονται.
7. Δεν ενδιαφέρει η σημασία της υπό εξέταση πληροφορίας.
8. Έχουν από τη φύση τους το στοιχείο του παραλληλισμού.
9. Είναι μία μέθοδος που κάνει ταυτόχρονα εξερεύνηση του χώρου αναζήτησης και εκμετάλλευση της ήδη επεξεργασμένης πληροφορίας.
10. Επιδέχονται παράλληλη υλοποίηση.

Άσκηση 8.1.4 (Επίπεδο δυσκολίας 1): Ποια είναι τα βασικά επιχειρήματα των πολέμιων των Γενετικών Αλγορίθμων;

Απάντηση:

1. Προβλήματα εξοικείωσης με τη Γενετική.
2. Το πρόβλημα του χρόνου.

Άσκηση 8.1.5 (Επίπεδο δυσκολίας 1): Ποια είναι τα κυριότερα χαρακτηριστικά που διαφοροποιούν τους Γενετικούς Αλγορίθμους από τις παραδοσιακές μεθόδους βελτιστοποίησης;

Απάντηση:

1. Οι ΓΑ δουλεύουν με μια κωδικοποίηση του συνόλου τιμών που μπορούν να λάβουν οι μεταβλητές και όχι με τις ίδιες τις μεταβλητές του προβλήματος.
2. Οι ΓΑ κάνουν αναζήτηση σε πολλά σημεία ταυτόχρονα και όχι μόνο σε ένα.
3. Οι ΓΑ χρησιμοποιούν μόνο την αντικειμενική συνάρτηση και καμία επιπρόσθετη πληροφορία.
4. Οι ΓΑ χρησιμοποιούν πιθανοθεωρητικούς κανόνες μετάβασης και όχι ντετερμινιστικούς.

Άσκηση 8.2.1 (Επίπεδο Δυσκολίας 1)

Η τιμή της ακέραιας παραμέτρου που ορίζει το σημείο διασταύρωσης σε ένα χρωμόσωμα. Πώς επιλέγεται αυτή η τιμή;

Απάντηση:

Το που επιλέγεται τυχαία με ομοιόμορφη κατανομή στο διάστημα $[1, 1-1]$, όπου 1 είναι το μήκος του χρωμοσώματος.

Άσκηση 8.2.2 (Επίπεδο Δυσκολίας 1)

Για έναν πληθυσμό με μέγεθος 100 και μήκος χρωμοσώματος 50 ποιος είναι αναμενόμενος αριθμός των ψηφίων που θα μεταλλαχθούν, αν η πιθανότητα μετάλλαξης παίρνει τις τιμές 0.001, 0.01 και 0.1;

Απάντηση:

Ο αναμενόμενος αριθμός ψηφίων που θα υποστούν μετάλλαξη υπολογίζεται από τη σχέση: *Μέγεθος πληθυσμού x μέγεθος χρωμοσώματος x πιθανότητα*. Άρα ο αναμενόμενος αριθμός μεταλλάξεων θα είναι αντίστοιχα:

$$100 \times 50 \times 0.001 = 5.0$$

$$100 \times 50 \times 0.01 = 50.0$$

$$100 \times 50 \times 0.1 = 500$$

Άσκηση 8.2.3 (Επίπεδο Δυσκολίας 1)

Αν στο παράδειγμα που αναλύσαμε η επιθυμητή ακρίβεια ήταν 2 δεκαδικά ψηφία, να υπολογίσετε τον συνολικό αριθμό δυαδικών ψηφίων που θα έχει το χρωμόσωμα.

Απάντηση:

Για την πρώτη μεταβλητή έχουμε: $2^9 < 1510 < 2^{10}$, άρα $m_1 = 10$ και $2^6 < 170 < 2^7$, άρα $m_2 = 7$.

Τελικά, το μήκος της συμβολοσειράς θα είναι $m = m_1 + m_2 = 10 + 7 = 17$.

Άσκηση 8.2.4 (Επίπεδο Δυσκολίας 1)

Αν η πιθανότητα διασταύρωσης είναι 0.2 και το μέγεθος του πληθυσμού είναι 100, ποιος είναι ο αναμενόμενος αριθμός ατόμων που θα επιλεγούν για διασταύρωση; Τι πρέπει να κάνουμε αν αυτός ο αριθμός είναι περιττός;

Απάντηση:

Ο αναμενόμενος αριθμός ατόμων που θα υποστούν διασταύρωση είναι $100 \times 0.2 = 20$. Αν ο αριθμός είναι περιττός, τότε είτε ένα άτομο δεν θα χρησιμοποιηθεί στη διαδικασία της διασταύρωσης ή θα επιλεγεί τυχαία επιπλέον ένα από τον πληθυσμό.

Άσκηση 8.2.5 (Επίπεδο Δυσκολίας 1)

Αν κατά το τρέξιμο του αλγορίθμου, σε κάποια γενιά τα καλύτερα άτομα που θα προκύψουν δίνουν χειρότερες τιμές για την αντικειμενική συνάρτηση, σε σχέση με τις προηγούμενες, πρέπει να διακοπεί η εκτέλεση του αλγορίθμου;

Απάντηση:

Δεν πρέπει να διακόψουμε την εκτέλεση του αλγορίθμου, γιατί στην επόμενη γενιά ή τις επόμενες γενιές είναι πολύ πιθανό να προκύψουν καλύτερα άτομα.

Άσκηση 8.2.6 (Επίπεδο Δυσκολίας 1)

Όταν τελειώσει η εκτέλεση του αλγορίθμου (αφού ικανοποιηθεί η συνθήκη τερματισμού), πώς θα εξασφαλίσουμε ότι το καλύτερο χρωμόσωμα της τελευταίας γενιάς είναι το καλύτερο όλου του πληθυσμού;

Απάντηση:

Στον απλό Γενετικό Αλγόριθμο που περιγράψαμε αυτό δεν είναι δυνατόν. Μια τροποποίηση που μπορεί να γίνει είναι να αποθηκεύουμε το «μέχρι τώρα καλύτερο» άτομο σε μια ξεχωριστή θέση, συνήθως έξω από τον πληθυσμό. Έτσι στο τέλος θα είμαστε σε θέση να γνωρίζουμε την καλύτερη τιμή που βρέθηκε κατά τη διάρκεια όλης της διαδικασίας.

Άσκηση 8.3.1 (Επίπεδο Δυσκολίας 2)

Να αποδείξετε ότι οποιαδήποτε δυαδική συμβολοσειρά μήκους m είναι στιγμότυπο από 2^m διαφορετικά σχήματα.

Απάντηση:

Γενικά αν m είναι το μήκος της συμβολοσειράς και $j = 1, \dots, m$ είναι το πλήθος των αδιάφορων συμβόλων, τότε το σύνολο από όλα τα στιγμότυπα θα είναι το άθροισμα των συνδυασμών m ανά j , άρα θα είναι:

$$\sum_{j=0}^m \binom{m}{j} = (1)^j (1)^{m-j} \binom{m}{j}$$

Από το θεώρημα του διωνύμου έχουμε ότι:

$$(x + y)^m = \sum_{j=0}^m \binom{m}{j} (x)^j (y)^{m-j}$$

Άρα αν θεωρήσουμε $x = y = 1$, προκύπτει εύκολα ότι:

$$\sum_{j=0}^m \binom{m}{j} = 2^m$$

Άσκηση 8.3.2 (Επίπεδο Δυσκολίας 2)

Να αποδείξετε ότι για ένα σχήμα με $\varepsilon > 0$, ο αριθμός των συμβολοσειρών που αναπαριστούν στις επόμενες γενιές αυξάνεται εκθετικά. Το αντίθετο συμβαίνει όταν $\varepsilon < 0$.

Απάντηση:

Αν υποθέσουμε ότι ένα σχήμα S βρίσκεται πάνω από τον μέσο όρο απόδοσης του πληθυσμού κατά $\varepsilon\%$ (δηλαδή $eval(S, t) = \overline{F(t)} + \varepsilon \cdot \overline{F(t)}$), τότε:

$$\xi(S, t) = \xi(S, 0) \cdot (1 + \varepsilon)^t, \text{ και}$$

$$\varepsilon = (eval(S, t) - \overline{F(t)}) / \overline{F(t)}$$

με $\varepsilon > 0$ για σχήματα πάνω από τον μέσο όρο και $\varepsilon < 0$ για σχήματα κάτω από τον μέσο όρο.

Η παραπάνω σχέση είναι μια εξίσωση γεωμετρικής προόδου. Επομένως, ένα σχήμα πάνω από τον μέσο όρο όχι μόνο αναπαριστά περισσότερες συμβολοσειρές στην επόμενη γενιά, αλλά επιπλέον ο αριθμός αυτός αυξάνεται εκθετικά. Όταν $\varepsilon < 0$, τότε αυτός ο αριθμός θα ελαττώνεται.

Άσκηση 8.3.3 (Επίπεδο Δυσκολίας 2)

Να ορίσετε την αντικειμενική συνάρτηση της δυαδικής συμβολοσειράς x έτσι ώστε να ισούται με τον αριθμό των άσων που περιέχονται στη x . Να δώσετε τη σχέση για τη μέση απόδοση ενός σχήματος S , το οποίο έχει k ορισμένα δυαδικά ψηφία όλα ίσα με 1, σε σχέση με το μήκος της δυαδικής συμβολοσειράς ℓ και το k .

Απάντηση:

Η αντικειμενική συνάρτηση f έχει την παρακάτω μορφή:

$$f = b_0 + b_1 + \dots + b_\ell, \text{ με } b_i = 0 \text{ ή } 1 \text{ και } i = 0, 1, \dots, \ell.$$

Στο σχήμα S ανήκουν $2^{\ell-k}$ δυαδικές συμβολοσειρές των οποίων η απόδοση ξεκινάει από k (δυαδική συμβολοσειρά με k άσους και $\ell - k$ μηδενικά) και φτάνει έως ℓ (δυαδική συμβολοσειρά με ℓ άσους).

Υπάρχει $\binom{\ell-k}{0} = 1$ συμβολοσειρά με απόδοση k . Επίσης, υπάρχουν $\binom{\ell-k}{1} = \ell - k$ συμβολοσειρές με απόδοση $k+1$, και $\binom{\ell-k}{2}$ συμβολοσειρές με απόδοση $k+2$, κοκ.

Επομένως, η μέση απόδοση του σχήματος S ισούται με:

$$\overline{f(S)} = \frac{1}{2^{\ell-k}} \sum_{j=0}^{\ell-k} (k+j) \binom{\ell-k}{j} = \frac{1}{2^{\ell-k}} \left(k \sum_{j=0}^{\ell-k} \binom{\ell-k}{j} + \sum_{j=0}^{\ell-k} j \binom{\ell-k}{j} \right) = \frac{1}{2^{\ell-k}} (k 2^{\ell-k} + (\ell-k) 2^{\ell-k-1}) = \frac{\ell+k}{2}$$

Παράδειγμα: Για $\ell=7$ και $k=3$, υπάρχουν συνολικά $2^{\ell-k}=2^4=16$ συμβολοσειρές που ταιριάζουν στο σχήμα, 1 με απόδοση 3, 4 με απόδοση 4, 6 με απόδοση 5, 4 με απόδοση 6 και 1 με απόδοση 7, οπότε η μέση απόδοση του σχήματος είναι:

$$\frac{1.3 + 4.4 + 6.5 + 4.6 + 1.7}{16} = \frac{80}{16} = 5$$

Από τον παραπάνω τύπο έχουμε:

$$\frac{\ell+k}{2} = \frac{7+3}{2} = 5$$

αρα τα αποτελέσματα συμφωνούν.

Άσκηση 8.3.4 (Επίπεδο Δυσκολίας 1)

Θεωρήστε τις τρεις δυαδικές συμβολοσειρές $A_1=11101111$, $A_2=00010100$ και $A_3=01000011$ και τα έξι σχήματα $H_1=1*****$, $H_2=0*****$, $H_3=*****11$, $H_4=***0*00*$, $H_5=1*****1*$, και $H_6=1110**1*$.

- Ποιες συμβολοσειρές ταιριάζουν σε κάθε σχήμα;
- Βρείτε την τάξη και το καθορισμένο μήκος κάθε σχήματος.
- Υπολογίστε την πιθανότητα επιβίωσης κάθε σχήματος, όταν υποβάλλεται σε μετάλλαξη και η πιθανότητα μιας απλής μετάλλαξης είναι $p_m=0.001$.
- Υπολογίστε την πιθανότητα επιβίωσης κάθε σχήματος, όταν υποβάλλεται σε διασταύρωση και η πιθανότητα διασταύρωσης είναι $p_c=0.85$.

Απάντηση:

- Η συμβολοσειρά A_1 ανήκει στα σχήματα H_1 , H_3 , H_5 και H_6 .
Η συμβολοσειρά A_2 ανήκει στο σχήμα H_2 .
Η συμβολοσειρά A_3 ανήκει στα σχήματα H_2 και H_3 .

β) Η τάξη ενός σχήματος S , η οποία συμβολίζεται $o(S)$, είναι ο αριθμός των θέσεων με 0 και 1. Το καθορισμένο μήκος ενός σχήματος S (συμβολίζεται $\delta(S)$) είναι η απόσταση μεταξύ της πρώτης και της τελευταίας σταθερής θέσης.

$$\begin{aligned} o(H_1) &= 1 & \delta(H_1) &= 1-1=0 \\ o(H_2) &= 1 & \delta(H_2) &= 1-1=0 \\ o(H_3) &= 2 & \delta(H_3) &= 8-7=1 \\ o(H_4) &= 3 & \delta(H_4) &= 7-4=3 \\ o(H_5) &= 2 & \delta(H_5) &= 7-1=6 \\ o(H_6) &= 5 & \delta(H_6) &= 7-1=6 \end{aligned}$$

γ) Η πιθανότητα επιβίωσης ενός σχήματος S όταν υποβάλλεται σε μετάλλαξη με πιθανότητα μετάλλαξης p_m ισούται με:

$$p_s(S) = (1 - p_m)^{o(S)} \approx 1 - o(S) \cdot p_m$$

Επομένως έχουμε:

$$\begin{aligned} p_s(H_1) &= 1 - 1 \cdot 0.001 = 0.999 \\ p_s(H_2) &= 1 - 1 \cdot 0.001 = 0.999 \\ p_s(H_3) &= 1 - 2 \cdot 0.001 = 0.998 \\ p_s(H_4) &= 1 - 3 \cdot 0.001 = 0.997 \\ p_s(H_5) &= 1 - 2 \cdot 0.001 = 0.998 \\ p_s(H_6) &= 1 - 5 \cdot 0.001 = 0.995 \end{aligned}$$

Η πιθανότητα επιβίωσης ενός σχήματος S όταν υποβάλλεται σε διασταύρωση με πιθανότητα διασταύρωσης p_c ισούται με:

$$p_s(S) \geq 1 - p_c \cdot \frac{\delta(S)}{m-1}$$

όπου m είναι το μήκος των δυαδικών συμβολοσειρών του σχήματος.

Επομένως έχουμε:

$$\begin{aligned} p_s(H_1) &= 1 - 0.85 \cdot 0/7 = 1 \\ p_s(H_2) &= 1 - 0.85 \cdot 0/7 = 1 \\ p_s(H_3) &= 1 - 0.85 \cdot 1/7 = 0.878571 \\ p_s(H_4) &= 1 - 0.85 \cdot 3/7 = 0.635714 \\ p_s(H_5) &= 1 - 0.85 \cdot 6/7 = 0.271429 \\ p_s(H_6) &= 1 - 0.85 \cdot 6/7 = 0.271429 \end{aligned}$$

Άσκηση 8.3.5 (Επίπεδο Δυσκολίας 1)

Ένας πληθυσμός περιλαμβάνει στη γενιά 0 τα ακόλουθα χρωμοσώματα και τις αντίστοιχες αποδόσεις τους:

A/A	Δυαδική συμβολοσειρά	Απόδοση
1	10001	20
2	11100	10
3	00011	5
4	01110	15

Η πιθανότητα μετάλλαξης είναι $p_m=0.01$ και η πιθανότητα διασταύρωσης είναι $p_c=1.0$.

- Υπολογίστε τον αναμενόμενο αριθμό σχημάτων της μορφής $S_1=(1****)$ που θα υπάρχουν στη γενιά 1.
- Υπολογίστε τον αναμενόμενο αριθμό σχημάτων της μορφής $S_2=(0**1*)$ που θα υπάρχουν στη γενιά 1.

Απάντηση:

Η εξίσωση που δίνει τον αναμενόμενο, στην επόμενη γενιά, αριθμό δυαδικών συμβολοσειρών που ανήκουν σε ένα σχήμα (θεωρώντας ότι χρησιμοποιούμε τη μέση απόδοση κάθε σχήματος για την επιλογή του) ισούται με:

$$\xi(S, t+1) \geq \xi(S, t) \cdot eval(S, t) / \overline{F(t)}$$

όπου $\xi(S, t+1)$ είναι ο αναμενόμενος αριθμός, $\xi(S, t)$ είναι ο αριθμός στην τρέχουσα γενιά, $eval(S, t)$ είναι η απόδοση του σχήματος S στην τρέχουσα γενιά και $\overline{F(t)}$ είναι η μέση απόδοση του πληθυσμού στην τρέχουσα γενιά.

Επομένως έχουμε:

$$\begin{aligned} \delta(S_1) &= 1 - 1 = 0 & o(S_1) &= 1 \\ \delta(S_2) &= 4 - 1 = 3 & o(S_2) &= 2 \end{aligned}$$

$$\overline{F(t=0)} = \sum_{i=1}^4 eval(v_i) / 4 = (20 + 10 + 5 + 15) / 4 = 12.5$$

$$eval(S_1, t=0) = (20 + 10) / 2 = 15$$

$$eval(S_2, t=0) = (5 + 15) / 2 = 10$$

δηλαδή έχουμε:

$$\alpha) \xi(S_1, t=1) = 2 \cdot \frac{15}{12.5} \left(1 - 1 \cdot \frac{0}{4} - 1 \cdot 0.01 \right) = 2.376$$

$$\beta) \xi(S_2, t=1) = 2 \cdot \frac{10}{12.5} \left(1 - 1 \cdot \frac{3}{4} - 2 \cdot 0.01 \right) = 0.368$$

Θα υπάρχουν επομένως τρεις συμβολοσειρές από το σχήμα S_1 και μία συμβολοσειρά από το σχήμα S_2 στον πληθυσμό στη γενιά 1.

Κεφάλαιο 9

Μελέτη Περίπτωσης – Εξελικτικά Νευρωνικά Δίκτυα

Σύνοψη

Σε αυτό το κεφάλαιο θα παρουσιάσουμε έναν τροποποιημένο Γενετικό Αλγόριθμο ή Εξελικτικό Πρόγραμμα, όπως αλλιώς ονομάζεται, για την ταυτόχρονη εκπαίδευση και βελτιστοποίηση της δομής Νευρωνικού Δικτύου. Το πρόβλημα που θα προσπαθήσουμε να επιλύσουμε είναι το XOR· δηλαδή θα προσπαθήσουμε να παραγάγουμε ένα δίκτυο που έχει μάθει τη συνάρτηση XOR.

Το κεφάλαιο είναι δομημένο ως εξής: Αρχικά γίνεται μια εισαγωγή στο πρόβλημα της εύρεσης της βέλτιστης αρχιτεκτονικής ενός Νευρωνικού Δικτύου και παρουσιάζονται διάφορες προσεγγίσεις του προβλήματος. Στο υπόλοιπο του κεφαλαίου γίνεται μια παρουσίαση ενός απλού Γενετικού Αλγορίθμου για την ταυτόχρονη βελτιστοποίηση της δομής και την εκπαίδευση Νευρωνικού Δικτύου. Αφού γίνει μια παρουσίαση του μοντέλου του δικτύου που θα επεξεργαστεί ο Γενετικός Αλγόριθμος και των φάσεων του αλγορίθμου, προχωράμε σε εφαρμογή του αλγορίθμου για την επίλυση του XOR προβλήματος και παρουσιάζουμε τα αποτελέσματα. Το κεφάλαιο ολοκληρώνεται με την παρουσίαση των συμπερασμάτων και της βιβλιογραφίας.

Βασικός στόχος του κεφαλαίου είναι να περιγράψει έναν τροποποιημένο Γενετικό Αλγόριθμο ή Εξελικτικό Πρόγραμμα για την ταυτόχρονη εκπαίδευση και βελτιστοποίηση της δομής Νευρωνικού Δικτύου. Με αυτόν τον τρόπο συνδέουμε τα Νευρωνικά Δίκτυα που παρουσιάστηκαν στο πρώτο μέρος του συγγράμματος με τους Γενετικούς Αλγορίθμους που αποτέλεσαν το αντικείμενο του δεύτερου μέρους. Επίσης, δίνεται η δυνατότητα στον αναγνώστη να δει μια εφαρμογή των Γενετικών Αλγορίθμων σε ένα πραγματικό πρόβλημα.

Προσδοκώμενα Αποτελέσματα

Όταν θα έχετε τελειώσει τη μελέτη αυτού του κεφαλαίου, θα μπορείτε να σχεδιάσετε:

- έναν εξελικτικό αλγόριθμο για την εκπαίδευση Νευρωνικών Δικτύων και
- έναν εξελικτικό αλγόριθμο για τη βελτιστοποίηση της δομής (αρχιτεκτονικής) ενός Νευρωνικού Δικτύου.

Λέξεις-Κλειδιά:

- Γενετικοί Αλγόριθμοι
- Εξελικτικοί Αλγόριθμοι
- Γενετικοί Τελεστές
- Νευρωνικά Δίκτυα
- Βελτιστοποίηση Αρχιτεκτονικής
- Βελτιστοποίηση Δομής
- Εκπαίδευση

Προαπαιτούμενη γνώση

Κεφάλαια 1, 2, 3, 8.

9.1 Εισαγωγή

Ένα από τα σημαντικότερα ζητήματα που αντιμετωπίζουμε κατά την εφαρμογή των Νευρωνικών Δικτύων σε πραγματικά προβλήματα είναι η επιλογή της κατάλληλης αρχιτεκτονικής (τοπολογίας) που πρέπει να έχει το δίκτυο, η οποία παίζει πολύ σημαντικό ρόλο στην ικανότητά του τόσο να μαθαίνει όσο και να γενικεύει. Το πρόβλημα αυτό, γνωστό ως πρόβλημα εύρεσης της βέλτιστης αρχιτεκτονικής, έως σήμερα παραμένει άλυτο. Βέβαια έχουν γίνει πολλές προσπάθειες για την επίλυσή του. Η πλέον συνηθισμένη τεχνική για την εύρεση μίας καλής τοπολογίας Νευρωνικού Δικτύου για ένα συγκεκριμένο πρόβλημα είναι αυτή της δοκιμής και του λάθους, γνωστή παγκοσμίως με τον όρο “trial and error”. Σύμφωνα με αυτήν, δοκιμάζουμε κάποια αρχιτεκτονική, βλέπουμε πώς δουλεύει και αν δεν μας ικανοποιεί, δοκιμάζουμε κάποια άλλη· η όλη διαδικασία επαναλαμβάνεται έως ότου να βρούμε μια ικανοποιητική τοπολογία.

Την τελευταία δεκαετία έχουν γίνει πολλές προσπάθειες για τη δημιουργία αλγορίθμων οι οποίοι αναζητούν τη βέλτιστη αρχιτεκτονική ενός Νευρωνικού Δικτύου για την επίλυση ενός συγκεκριμένου προβλήματος. Τους αλγορίθμους αυτούς μπορούμε να τους διακρίνουμε σε διάφορες κατηγορίες.

Μια κατηγορία τέτοιων μεθόδων είναι οι *κατασκευαστικοί (constructive)* και οι *καταστροφικοί (pruning)* αλγόριθμοι (Reed, 1993· Yao & Liu, 1997). Ένας *κατασκευαστικός αλγόριθμος*, γενικά, ξεκινά με ένα ελάχιστο δίκτυο (ένα δίκτυο με τον ελάχιστο αριθμό κρυφών επιπέδων, νευρώνων και συνδέσεων) και προσθέτει συνδέσεις, νευρώνες και κρυφά επίπεδα κατά τη φάση της εκπαίδευσης εάν το «κρίνει» απαραίτητο. Αντίθετα, ένας *καταστροφικός αλγόριθμος* κάνει το αντίθετο, γενικά ξεκινά με ένα μέγιστο δίκτυο και «κόβει» τις συνδέσεις, τους νευρώνες και τα κρυφά επίπεδα που «θεωρεί» μη απαραίτητα κατά τη φάση της εκπαίδευσης.

Μια άλλη, πιο πρόσφατη προσέγγιση στο πρόβλημα της εύρεσης της βέλτιστης αρχιτεκτονικής είναι με τη χρήση Γενετικών Αλγορίθμων και γενικότερα Εξελικτικού Προγραμματισμού (Holland, 1992· Goldberg, 1989· Michalewicz, 1996· Mitchell, 1995). Όπως είναι γνωστό, οι Γενετικοί Αλγόριθμοι είναι μια τάξη μεθόδων βελτιστοποίησης που διακρίνονται για την ικανότητά τους να εξερευνούν μεγάλους και πολύπλοκους χώρους με πολύ έξυπνο και αποτελεσματικό τρόπο προκειμένου να βρουν λύσεις κοντά στην ολικά βέλτιστη. Το πρόβλημα της εύρεσης της βέλτιστης αρχιτεκτονικής μπορούμε πολύ εύκολα να το σχηματοποιήσουμε ως ένα πρόβλημα αναζήτησης στον χώρο των αρχιτεκτονικών, όπου κάθε σημείο του χώρου αντιπροσωπεύει και μια αρχιτεκτονική. Με παρόμοιο τρόπο το πρόβλημα της εκπαίδευσης ενός Νευρωνικού Δικτύου μπορεί να διατυπωθεί ως πρόβλημα αναζήτησης στον χώρο των βαρών. Στο τέλος περίπου της προηγούμενης δεκαετίας ξεκίνησαν διάφορες προσπάθειες για τον συνδυασμό των τεχνολογιών των Νευρωνικών Δικτύων και των Γενετικών Αλγορίθμων. Δοθέντων κάποιων κριτηρίων απόδοσης, όπως π.χ. το λάθος, η ικανότητα γενίκευσης, ο χρόνος για την εκπαίδευση, η πολυπλοκότητα της αρχιτεκτονικής κ.ά. για την αρχιτεκτονική του δικτύου, σχηματίζεται μια επιφάνεια στον χώρο των αρχιτεκτονικών. Έτσι η βέλτιστη αρχιτεκτονική αντιστοιχεί στην εύρεση του μέγιστου αυτής της επιφάνειας.

Οι πρώτες προσπάθειες (Davis, 1988· Montana & Davis, 1989· Whitley, 1988· Whitley & Hanson, 1989) για τον συνδυασμό Νευρωνικών Δικτύων και Γενετικών Αλγορίθμων επικεντρώθηκαν στην εκπαίδευση ενός δικτύου και όχι στον σχεδιασμό μιας σχεδόν βέλτιστης τοπολογίας. Έτσι σε δίκτυα με καθορισμένη, σταθερή, αρχιτεκτονική χρησιμοποίησαν Γενετικό Αλγόριθμο προκειμένου να ψάξει στον χώρο των βαρών για ένα σχεδόν βέλτιστο διάνυσμα βαρών που να επιλύει το πρόβλημα της εκπαίδευσης του δικτύου· με άλλα λόγια χρησιμοποίησαν τους Γενετικούς Αλγορίθμους αντί κάποιων κλασικών αλγορίθμων εκπαίδευσης, όπως π.χ. του αλγορίθμου Πίσω Διάδοσης του Λάθους (Error Back-Propagation). Κάποιες από αυτές τις προσπάθειες αποδείχτηκαν πολύ επιτυχημένες και μάλιστα, σε κάποια δύσκολα πραγματικά προβλήματα, πέτυχαν καλύτερα αποτελέσματα από πιο κλασικές μεθόδους, όπως τον αλγόριθμο Πίσω Διάδοσης του Λάθους.

Γρήγορα όμως το ενδιαφέρον της επιστημονικής κοινότητας μετακινήθηκε από την εκπαίδευση ενός Νευρωνικού Δικτύου στην αναζήτηση της βέλτιστης αρχιτεκτονικής του, μιας και το συγκεκριμένο πρόβλημα μπορεί να αντιμετωπιστεί πολύ αποτελεσματικά από τους Εξελικτικούς Αλγορίθμους. Τα τελευταία χρόνια λοιπόν έχει υπάρξει ένας μεγάλος αριθμός από εργασίες, στη διεθνή βιβλιογραφία, για την αντιμετώπισή του. Περισσότερες πληροφορίες πάνω στην εφαρμογή των Εξελικτικών Αλγορίθμων και στη βελτιστοποίηση της δομής Νευρωνικών Δικτύων ο ενδιαφερόμενος αναγνώστης μπορεί να αναζητήσει στις ακόλουθες εργασίες (Yao & Liu, 1997· Happel et. al., 1994· Miller et. al., 1989· Whitley & Bogart, 1990· Billings & Zheng, 1995· Bornholdt & Graudenz, 1992· Adamopoulos et. al., 1998· Likothanassis et. al., 1998· Likothanassis et. al., 1997· Adamopoulos et. al., 1997· Andreou et. al., 1997· Andreou et. al., 1998· Adamopoulos et. al., 1999).

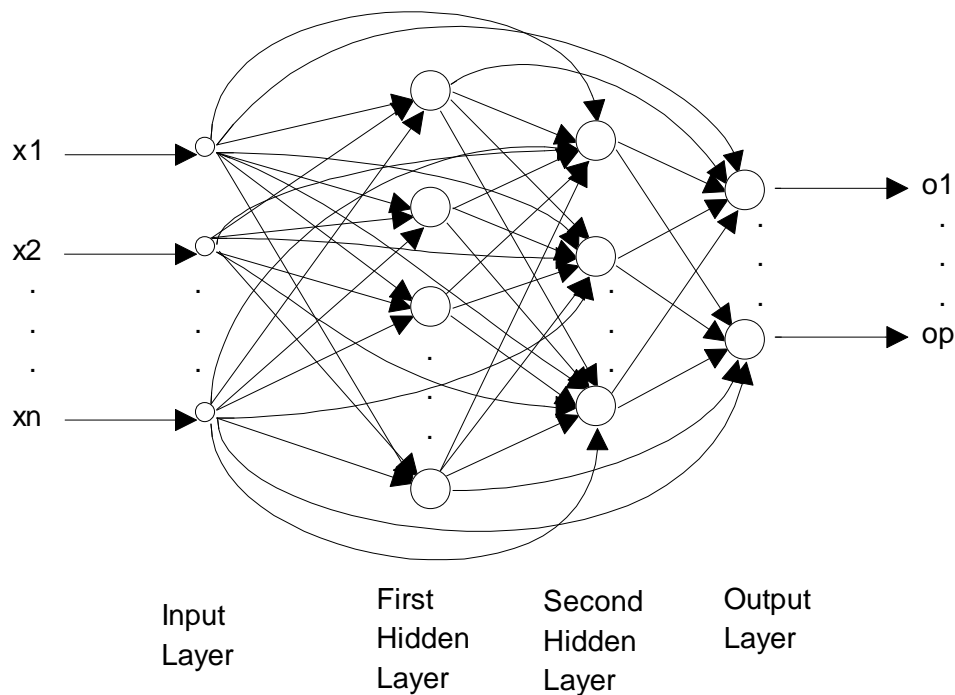
Σε αυτό το κεφάλαιο θα παρουσιάσουμε έναν τροποποιημένο Γενετικό Αλγόριθμο ή, αλλιώς, Εξελικτικό Πρόγραμμα για την ταυτόχρονη εκπαίδευση και βελτιστοποίηση της δομής Νευρωνικού Δικτύου. Το πρόβλημα που θα προσπαθήσουμε να επιλύσουμε είναι το XOR· δηλαδή θα προσπαθήσουμε να παραγάγουμε ένα δίκτυο που έχει μάθει τη συνάρτηση XOR. Ο αλγόριθμος που θα παρουσιάσουμε δημοσιεύτηκε για πρώτη φορά στο Likothanassis et. al., 1997.

9.2 Το μοντέλο του νευρώνα

Το μοντέλο του νευρωνικού δικτύου που θα επεξεργαστεί ο γενετικός αλγόριθμος επιλέχθηκε να είναι όσο πιο γενικό γίνεται. Γι' αυτό χρησιμοποιήσαμε πολυεπίπεδα δίκτυα εμπρόσθιας τροφοδότησης. Ένα χαρακτηριστικό δίκτυο αυτού του είδους αποτελείται από ένα σύνολο αισθητήριων νευρώνων που συγκροτούν το επίπεδο εισόδου, ένα ή περισσότερα κρυμμένα επίπεδα υπολογιστικών νευρώνων και ένα επίπεδο εξόδου από υπολογιστικούς νευρώνες. Στο μοντέλο του νευρωνικού δικτύου που υλοποιήσαμε έγιναν οι ακόλουθες υποθέσεις:

- Κάθε (υπολογιστικός) νευρώνας στο δίκτυο παριστάνεται από το μοντέλο McCulloch-Pitts.
- Τα bits 0 και 1 αναπαρίστανται από τα επίπεδα 0 και +1 αντίστοιχα.

Το μοντέλο του δικτύου είναι πλήρως συνδεδεμένο με feedforward συνδέσεις. Έτσι κάθε νευρώνας στο επίπεδο εισόδου συνδέεται με κάθε νευρώνα στα κρυμμένα επίπεδα, με τον ίδιο τρόπο που κάθε νευρώνας σε ένα κρυμμένο επίπεδο συνδέεται με κάθε νευρώνα στα επίπεδα που ακολουθούν. Το ίδιο ισχύει για τις συνδέσεις ανάμεσα στους νευρώνες των κρυμμένων επιπέδων και τους νευρώνες του επιπέδου εξόδου. Επιπλέον, εισάγουμε απευθείας συνδέσεις τροφοδοτούμενες προς τα εμπρός ανάμεσα στους νευρώνες εισόδου και εξόδου. Το σχήμα 9.1 απεικονίζει ένα δείγμα του γενικού μοντέλου του νευρωνικού δικτύου που εξελίσσει ο τροποποιημένος Γενετικός Αλγόριθμος.



Σχήμα 9.1 Γράφος αρχιτεκτονικής ενός γενικού μοντέλου δικτύου, με δύο κρυφά επίπεδα (Likothanassis et. al., 1997).

Ο στόχος της προσέγγισής μας ήταν να υλοποιήσουμε ένα νευρωνικό δίκτυο που να μπορεί να μεταβάλλει τη δομή του δυναμικά. Για τον λόγο αυτό χρησιμοποιήσαμε, για την υλοποίησή του, διασυνδεδεμένες λίστες. Το συνολικό νευρωνικό δίκτυο κατασκευάζεται ως μια διασυνδεδεμένη λίστα επιπέδων. Κάθε επίπεδο οργανώνεται ως μια διασυνδεδεμένη λίστα από νευρώνες και κάθε νευρώνας ως μια διασυνδεδεμένη λίστα από συνάψεις (συνδέσεις) που καταλήγουν σε αυτόν, ενώ κάθε σύναψη προσδιορίζεται από τον νευρώνα από τον οποίο ξεκινά και από την ισχύ της (βάρος). Με αυτού του είδους την υλοποίηση έχουμε τη δυνατότητα να προσθέτουμε ή να αφαιρούμε δομικά συστατικά (επίπεδα, νευρώνες, συνδέσεις) του δικτύου δυναμικά προκειμένου να μεταβάλλουμε την αρχιτεκτονική του.

9.3 Ο Αλγόριθμος

Στη συνέχεια θα περιγράψουμε τους γενετικούς τελεστές που χρησιμοποιούνται και θα παρουσιάσουμε τα βήματα του αλγορίθμου.

Αρχικοποίηση:

Στο βήμα αυτό δημιουργούμε έναν αρχικό πληθυσμό από τυχαία παραγόμενα νευρωνικά δίκτυα. Κάθε δίκτυο δημιουργείται έτσι ώστε ο αριθμός των κρυφών επιπέδων, καθώς και ο αριθμός των νευρώνων σε κάθε κρυφό επίπεδο να επιλέγεται τυχαία (χρησιμοποιώντας ομοιόμορφη κατανομή) με τη χρήση μιας γεννήτριας τυχαίων αριθμών. Τα βάρη των νευρωνικών δικτύων που προκύπτουν επιλέγονται τυχαία (με ομοιόμορφη κατανομή) στο διάστημα [ελάχιστη τιμή, μέγιστη τιμή] (η ελάχιστη και μέγιστη τιμή των βαρών μπορούν να περαστούν ως παράμετροι του αλγορίθμου). Με αυτόν τον τρόπο στο τέλος της διαδικασίας αρχικοποίησης έχουμε έναν πληθυσμό νευρωνικών δικτύων με τυχαία τοπολογία.

Αξιολόγηση:

Στο βήμα αυτό γίνεται η αξιολόγηση των διαφόρων δικτύων που απαρτίζουν τον πληθυσμό. Για κάθε πρότυπο εκπαίδευσης υπολογίζουμε την έξοδο κάθε νευρώνα εξόδου του δικτύου και τη συγκρίνουμε με την επιθυμητή. Με αυτόν τον τρόπο μπορούμε να υπολογίσουμε τον αριθμό των σωστών bits που έχει μάθει το δίκτυο. Έτσι, αν η συνάρτηση που έχουμε να εκπαιδεύσουμε στο δίκτυο έχει n bits εισόδου και r bits εξόδου, έχουμε να ελέγξουμε 2^n ενδεχόμενα, που έχουν ως αποτέλεσμα $2^n r$ bits να μπορούν να είναι αληθή (true) ή ψευδή (false). Στη συνέχεια εκτιμούμε την απόδοση κάθε νευρωνικού δικτύου στον πληθυσμό, χρησιμοποιώντας μια πολύ απλή συνάρτηση καταλληλότητας που ισοδυναμεί με τον αριθμό των σωστών bits στην έξοδο κάθε δικτύου.

Επιλογή:

Για τη διαδικασία της επιλογής μπορούμε να χρησιμοποιήσουμε οποιαδήποτε από τις γνωστές τεχνικές που υπάρχουν στη διεθνή βιβλιογραφία. Στη συγκεκριμένη εφαρμογή και για λόγους απλότητας επιλέξαμε να χρησιμοποιήσουμε την elitism roulette wheel selection (Michalewicz, 1996). Η τεχνική αυτή μοιάζει με την κλασική roulette wheel selection που περιγράψαμε στο προηγούμενο κεφάλαιο, με μόνη διαφορά ότι εδώ εξασφαλίζουμε πως το καλύτερο μέλος του πληθυσμού θα περάσει σε ένα τουλάχιστον αντίγραφο στον επόμενο πληθυσμό. Χρησιμοποιώντας λοιπόν ως μέτρο τις fitness functions των δικτύων που υπολογίστηκαν στο βήμα της αξιολόγησης, επιλέγουμε τα δίκτυα που θα «περάσουν» στον νέο πληθυσμό. Εδώ θα πρέπει να σημειώσουμε ότι κατά τη διαδικασία επιλογής, μεταξύ δικτύων με το ίδιο fitness, ευνοούμε εκείνα με το μικρότερο μέγεθος. Μια άλλη μέθοδος για να «πιέσουμε» τον πληθυσμό μας σε μικρότερα δίκτυα είναι να χρησιμοποιήσουμε μια fitness function που να περιλαμβάνει και την αρχιτεκτονική (το μέγεθος) του δικτύου. Για λόγους ευκολίας και απλότητας, όμως, αποφασίσαμε να μην το βάλουμε στον αλγόριθμο που παρουσιάζουμε.

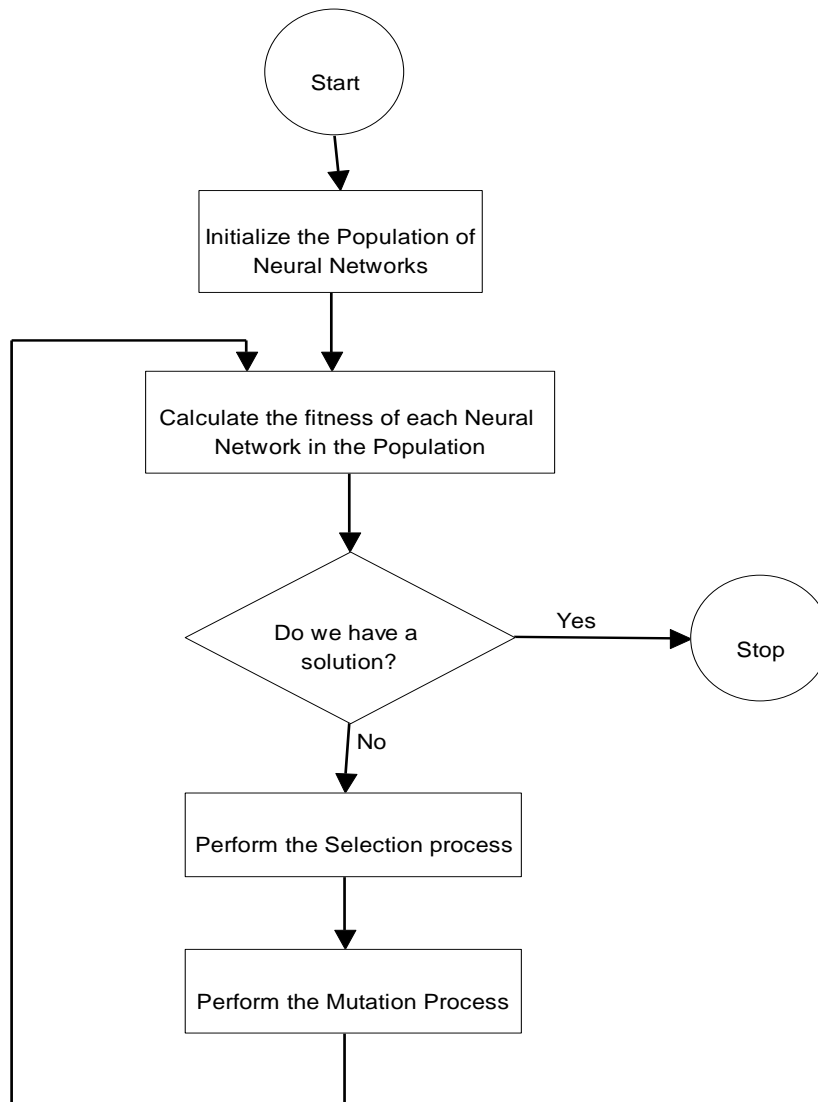
Μετάλλαξη:

Ο τελεστής της μετάλλαξης είναι αυτός που ουσιαστικά δημιουργεί τα καινούρια δίκτυα, μιας και στον συγκεκριμένο αλγόριθμο δεν χρησιμοποιούμε διασταύρωση, με αποτέλεσμα να είναι εξαιρετικά σημαντικός για την απόδοση του αλγορίθμου. Με μια ορισμένη πιθανότητα (η οποία δίνεται ως παράμετρος στον αλγόριθμο) επιλέγουμε τα δίκτυα που θα μεταλλαχθούν. Από αυτά τα δίκτυα αφαιρούμε ή προσθέτουμε έναν τυχαίο αριθμό νευρώνων από τα κρυφά επίπεδα. Τα βάρη των νέων συνδέσεων κατανέμονται τυχαία στο διάστημα [ελάχιστη τιμή, μέγιστη τιμή] (ίδια με αυτά που χρησιμοποιήσαμε στη φάση της αρχικοποίησης). Αυτό το είδος της μεταβολής δεν είναι το καλύτερο δυνατόν και πράγματι στις Adamopoulos et. al., 1998· Likothanassis et. al., 1998· Likothanassis et. al., 1997· Adamopoulos et. al., 1997· Andreou et. al., 1997· Andreou et. al., 1998 και Adamopoulos et. al., 1999 έχουν αναπτυχθεί πολύ πιο αποτελεσματικοί τελεστές, αλλά ταυτόχρονα και αρκετά πιο πολύπλοκοι. Το σημαντικό όμως είναι ότι ακόμα και αυτή η απλή ιδέα μετάλλαξης φαίνεται να εργάζεται καλά, χωρίς να υπάρχει ανάγκη να καθορίσουμε μια πολύπλοκη δομή πληροφορίας ή έναν καθολικό κανόνα μάθησης.

Όπως αναφέραμε και προηγουμένως, στον συγκεκριμένο αλγόριθμο που παρουσιάζουμε δεν χρησιμοποιούμε αναπαραγωγή. Αυτό έγινε για λόγους ευκολίας στην ανάπτυξή του, μιας και η κατασκευή ενός «καλού» τελεστή αναπαραγωγής για δυναμικά μεταβαλλόμενα νευρωνικά δίκτυα είναι πολύ δύσκολη υπόθεση και

αποτελεί ακόμα ανοιχτό αντικείμενο έρευνας. Η κατασκευή ενός απλού τελεστή αναπαραγωγής είναι σχετικά εύκολη, όμως δεν προσφέρει σχεδόν τίποτα στην απόδοση του αλγορίθμου.

Στη συνέχεια και στο σχήμα 9.2 παρουσιάζουμε το διάγραμμα ροής του εξελικτικού αλγορίθμου.



Σχήμα 9.2 Διάγραμμα ροής του εξελικτικού αλγορίθμου (Likothanassis et. al., 1997).

9.4 Εξομοιώσεις

Σε αυτήν την ενότητα θα χρησιμοποιήσουμε τον εξελικτικό αλγόριθμο που παρουσιάσαμε για να επιλύσουμε το XOR πρόβλημα — δηλαδή να παραγάγουμε ένα δίκτυο που να έχει μάθει τη συνάρτηση XOR. Το XOR είναι ένα πρόβλημα ορόσημο στην ιστορία των νευρωνικών δικτύων και παραμένει ένα πολύ αγαπημένο πρόβλημα-«παιχνίδι» στον χώρο. Στον πίνακα 9.1 μπορούμε να δούμε μερικά παραδείγματα εκπαίδευσης της απλής XOR συνάρτησης. Ο πίνακας 9.2 παρουσιάζει τον μέσο αριθμό γενεών που χρειάζονται για να λυθεί το πρόβλημα XOR και τον μέσο αριθμό νευρώνων στα κρυφά επίπεδα του καλύτερου δικτύου που προκύπτει στο τέλος της εκτέλεσης του αλγορίθμου.

Πίνακας 9.1 Τα πρώτα πέντε τρεξίματα με μέγεθος πληθυσμού 10 δίκτυα.

Αριθμός Γενεών	Αριθμός Κρυφών Νευρώνων
15	2
0	2
8	11
102	6
123	3

Πίνακας 9.2 Ο μέσος αριθμός γενεών που απαιτούνται για το πρόβλημα XOR, και ο μέσος αριθμός κρυφών νευρώνων στο καλύτερο δίκτυο για 40 τρεξίματα.

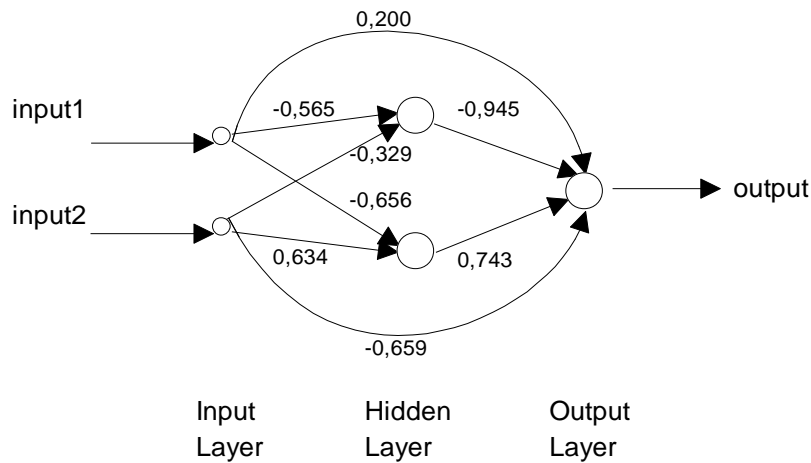
Μέγεθος Πληθυσμού	Μέσος Αριθμός Γενεών	Μέσος Αριθμός Κρυφών Νευρώνων
10	52.45	7,38
20	30.38	7,30

Στον πίνακα 9.3 παρουσιάζουμε μερικά παραδείγματα εκπαίδευσης της XOR συνάρτησης σε ένα γενετικά εκπαιδευόμενο νευρωνικό δίκτυο, από την προσέγγιση που αναφέρεται στο Bornholdt & Graudenz, 1992. Σε αυτήν την εργασία χρησιμοποιείται ένα αραιωμένο (diluted) βιολογικό νευρωνικό δίκτυο. Από τη σύγκριση των πινάκων 9.2 και 9.3 προκύπτει ότι ο αλγόριθμος που παρουσιάσαμε παράγει μικρότερα δίκτυα σε λιγότερο αριθμό γενεών.

Πίνακας 9.3 Αποτελέσματα από την εκπαίδευση της XOR συνάρτησης από την εργασία των Bornholdt & Graudenz, 1992.

Μέγεθος Πληθυσμού	Αριθμός Γενεών	Αριθμός Κρυφών Νευρώνων
10	1200	22
10	2400	7
10	5000	8
10	5500	13
10	6000	12

Στο σχήμα 9.3 μπορούμε να δούμε ένα παράδειγμα νευρωνικού δικτύου που επιλύει το XOR πρόβλημα και παράγεται από τον εξελικτικό αλγόριθμο, που παρουσιάσαμε, στο τέλος της εξελικτικής διαδικασίας.

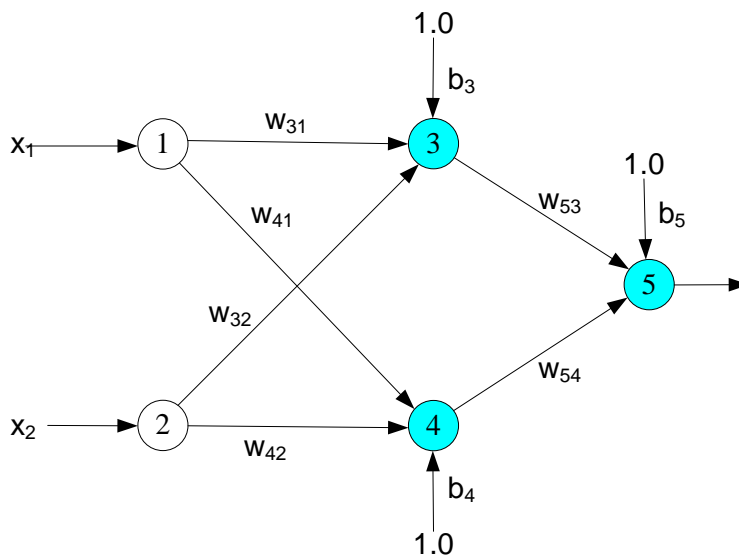


Σχήμα 9.3 Ένα γενετικά παραγόμενο νευρωνικό δίκτυο που επιλύει το XOR πρόβλημα (Likothanassis et. al., 1997).

Λύνοντας τις παρακάτω ασκήσεις θα είστε σε θέση να γνωρίζετε αν έχετε κατανοήσει τις έννοιες αλλά και τις τεχνικές που παρουσιάστηκαν παραπάνω. Όλες οι ασκήσεις έχουν ως στόχο την ανακεφαλαίωση όσων παρουσιάστηκαν στο κεφάλαιο 8.

Άσκηση 9/1:

Δίνεται το παρακάτω Τεχνητό Νευρωνικό Δίκτυο:



Έστω ότι θέλουμε να χρησιμοποιήσουμε Γενετικό Αλγόριθμο για να εκπαιδεύσουμε το συγκεκριμένο Νευρωνικό Δίκτυο — θέλουμε δηλαδή να χρησιμοποιήσουμε Γενετικό Αλγόριθμο προκειμένου να βρούμε ένα βέλτιστο ή σχεδόν βέλτιστο διάνυσμα βαρών και πολώσεων του παραπάνω δικτύου.

1. Προτείνετε μία αναπαράσταση (κωδικοποίηση των δυνατών λύσεων) για τον Γενετικό Αλγόριθμο που θα χρησιμοποιήσετε. Δώστε ένα παράδειγμα της συγκεκριμένης αναπαράστασης.

2. Πώς θα δημιουργηθεί ο αρχικός πληθυσμός; Προτείνετε μία μέθοδο αρχικοποίησης του πληθυσμού.

3. Προτείνετε μία συνάρτηση αξιολόγησης (fitness function) για τον αλγόριθμο που θα χρησιμοποιήσετε.

4. Περιγράψτε συνοπτικά τον γενετικό τελεστή της διασταύρωσης που θα χρησιμοποιήσετε.

5. Περιγράψτε συνοπτικά τον γενετικό τελεστή της μετάλλαξης που θα χρησιμοποιήσετε.

Άσκηση 9/2:

A. Τα Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ) με βηματικούς νευρώνες και ακέραια βάρη έχουν ιδιαίτερο ενδιαφέρον. Να αναφέρετε ποια θεωρείτε πως είναι μερικά πλεονεκτήματά τους σε σχέση με τα συνήθη ΤΝΔ.

B. Για να εκπαιδεύσουμε ένα ΤΝΔ με βηματικούς νευρώνες και ακέραια βάρη, θα προτιμήσουμε τη μέθοδο της οπισθοδιάδοσης των βαρών ή τη χρήση Γενετικών Αλγορίθμων;

Γ. Θέλουμε με χρήση ΓΑ να εκπαιδεύσουμε ένα ΤΝΔ για το πρόβλημα του XOR, με βηματικούς νευρώνες, ακέραια βάρη, και 2-2-1 αρχιτεκτονική. Επιπρόσθετα, περιορίζουμε τις δυνατές τιμές των βαρών στο σύνολο $B = \{-2, 2\}$ και των πολώσεων στο σύνολο $\Pi = \{-3, -2, -1, 0\}$. Να προτείνετε μια πιθανή κωδικοποίηση (να δώσετε και ένα παράδειγμα), τη συνάρτηση αξιολόγησης, τους τελεστές διασταύρωσης και μετάλλαξης, καθώς και τη συνθήκη τερματισμού.

Δραστηριότητες

Οι δραστηριότητες που ακολουθούν θα σας δώσουν τη δυνατότητα να κατανοήσετε καλύτερα την πρακτική εφαρμογή των Γενετικών Αλγορίθμων, να τους δείτε πλέον από την πλευρά της υλοποίησης και θα σας οδηγήσουν από τον χώρο της θεωρίας στον χώρο της πράξης. Ολοκληρώνοντάς τες θα είστε σε θέση να υλοποιήσετε έναν εξελικτικό αλγόριθμο και για διάφορα άλλα προβλήματα πέρα από αυτό που περιγράψαμε.

Δραστηριότητα 1

Υλοποιήστε σε κάποια γλώσσα προγραμματισμού το μοντέλο ενός νευρωνικού δικτύου που να έχει τη δυνατότητα να μεταβάλει δυναμικά τη δομή του, όπως αυτό περιγράφηκε προηγουμένως.

Δραστηριότητα 2

Υλοποιήστε σε κάποια γλώσσα προγραμματισμού τους γενετικούς τελεστές που περιγράψαμε σε αυτό το κεφάλαιο.

Δραστηριότητα 3

Υλοποιήστε σε κάποια γλώσσα προγραμματισμού τον «συνολικό» εξελικτικό αλγόριθμο που περιγράψαμε, ο οποίος απεικονίζεται στο σχήμα 8.2.

Δραστηριότητα 4

Πειραματιστείτε με το πρόβλημα XOR για διάφορες τιμές των παραμέτρων του εξελικτικού αλγορίθμου.

Δραστηριότητα 5

Πειραματιστείτε με διάφορα άλλα προβλήματα είτε δυαδικά, όπως το AND ή το OR, είτε πραγματικά. Τέτοια προβλήματα μπορείτε να «κατεβάσετε» από τη διεύθυνση (Murphy, P.M. & Aha, D.W.): <http://www.ics.uci.edu/~mlern/MLRepository.html>

9.5 Ανασκόπηση – Συμπεράσματα

Σε αυτό το κεφάλαιο παρουσιάσαμε έναν απλό τροποποιημένο Γενετικό Αλγόριθμο ή Εξελικτικό Πρόγραμμα, όπως αλλιώς το ονομάσαμε, για τη βελτιστοποίηση της αρχιτεκτονικής και την εκπαίδευση νευρωνικού δικτύου. Στόχος μας ήταν να παρουσιάσουμε ένα πεδίο εφαρμογής των Γενετικών Αλγορίθμων που συζητήσαμε στο προηγούμενο κεφάλαιο, το οποίο να συνδυάζεται με τα νευρωνικά δίκτυα που παρουσιάστηκαν στην αρχή του συγγράμματος. Προσπαθήσαμε ο αλγόριθμος που περιγράψαμε να είναι όσο πιο απλός και κατανοητός γίνεται. Παρ' όλη την απλότητά του έγινε φανερή από τις εξομοιώσεις η δύναμη των Γενετικών Αλγορίθμων στην επίλυση δύσκολων προβλημάτων, όπως είναι και το πρόβλημα της εκπαίδευσης και της εύρεσης της βέλτιστης αρχιτεκτονικής Νευρωνικών Δικτύων.

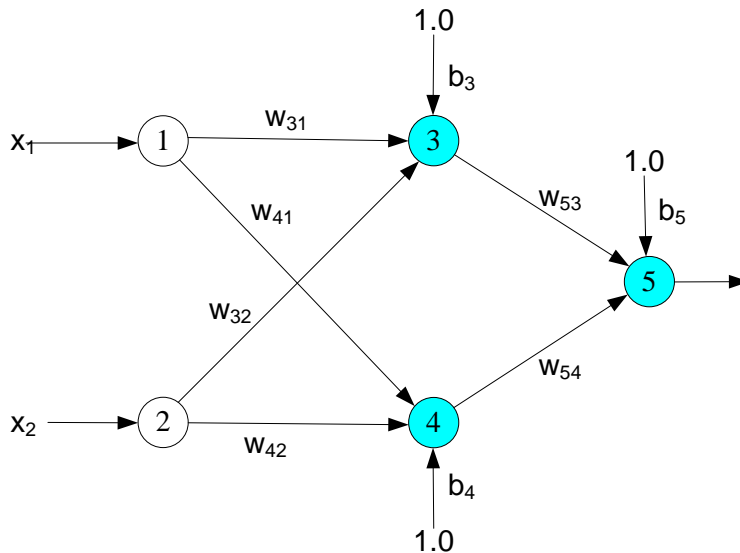
Βιβλιογραφία

- Andreou, A., Georgopoulos, E., Zombanakis, G., & Likothanassis, S. (1998). Testing Currency Predictability Using An Evolutionary Neural Network Model. Proceedings of the *fifth International Conference on Forecasting Financial Markets*, Banque Nationale de Paris and Imperial College, London.
- Adamopoulos, A., Georgopoulos, E., Manioudakis, G., & Likothanassis, S. (1998). An Evolutionary Method for System Structure Identification Using Neural Networks. *Neural Computation* '98.
- Adamopoulos, A.V., Georgopoulos, E.F., Likothanassis, S.D., & Anninos, P.A. (1999). Forecasting the MagnetoEncephaloGram (MEG) of Epileptic Patients Using Genetically Optimized Neural Networks. *GECCO 1999*.
- Billings, S.A., & Zheng, G.L. (1995). Radial basis function network configuration using genetic algorithms. *Neural Networks*, 8, 877-890.
- Bornholdt, S., & Graudenz, D. (1992). General asymmetric neural networks and structure design by genetic algorithms. *Neural Networks*, 5, 327-334.
- Davis, L. (1988). Mapping classifier systems into neural networks. Στο *Proceedings of the 1988 Conference on Neural Information Processing Systems*. Morgan Kaufmann.
- Goldberg, D. (1989). *Genetic Algorithms in Search Optimization & Machine Learning*. Addison-Wesley.
- Happel, B. et al. (1994). Design and evolution of modular neural network architectures. *Neural Networks*, 7, 985-1004.
- Holland, J. (1992). *Adaptation in Natural and Artificial Systems*. MIT Press.
- Likothanassis, S.D., Georgopoulos, E.F., & Manioudakis, G. (1998, Σεπτέμβριος). Currency Forecasting Using Genetically Optimized Neural Networks. *HERCMA* Αθήνα.
- Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag.
- Miller, G. et al. (1989). Designing neural networks using genetic algorithms. Στο *Proceedings of the 3rd International Conference on Genetic Algorithms*. Morgan Kaufmann.
- Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. MIT Press. Montana, D., & Davis, L. (1989). Training feedforward neural networks using genetic algorithms. *BBN Systems and Technologies*. Cambridge, MA.
- Murphy, P.M., & Aha, D.W. (1994). UCI Repository of Machine Learning Databases – <http://www.ics.uci.edu/~mllearn/MLRepository.html> University of California, Department of Information and Computer Science.
- Reed, R. (1993). Pruning Algorithms – a survey. *IEEE Trans. On Neural Networks*, 4(5), 740-747.
- Yao, X., & Liu, Y. (1997). A New Evolutionary System for Evolving Artificial Neural Networks. *IEEE Transactions on Neural Networks*, 8(3).
- Whitley, D. (1988). Applying genetic algorithms to neural network problems. *International Neural Networks Society*, 230.
- Whitley, D., & Hanson, T. (1989). Optimizing neural networks using faster, more accurate genetic search. Στο *3rd Intern. Conference on Genetic Algorithms, Washington D.C.* (σ. 391-396), Morgan Kaufmann.
- Whitley, D. & Bogart, C. (1990). The evolution of connectivity: Pruning neural networks using genetic algorithms. *International Joint Conference on Neural Networks, Washington D.C.*, 1 (σ. 134-137). Lawpence Erlbaum.

Απαντήσεις ασκήσεων

Άσκηση 9/1:

Δίνεται το παρακάτω Τεχνητό Νευρωνικό Δίκτυο:



Έστω ότι θέλουμε να χρησιμοποιήσουμε Γενετικό Αλγόριθμο για να εκπαιδεύσουμε το συγκεκριμένο Νευρωνικό Δίκτυο — θέλουμε δηλαδή να χρησιμοποιήσουμε Γενετικό Αλγόριθμο προκειμένου να βρούμε ένα βέλτιστο ή σχεδόν βέλτιστο διάνυσμα βαρών και πολώσεων του παραπάνω δικτύου.

1. Προτείνετε μία αναπαράσταση (κωδικοποίηση των δυνατών λύσεων) για τον Γενετικό Αλγόριθμο που θα χρησιμοποιήσετε. Δώστε ένα παράδειγμα της συγκεκριμένης αναπαράστασης.
2. Πώς θα δημιουργηθεί ο αρχικός πληθυσμός; Προτείνετε μία μέθοδο αρχικοποίησης του πληθυσμού.
3. Προτείνετε μία συνάρτηση αξιολόγησης (fitness function) για τον αλγόριθμο που θα χρησιμοποιήσετε.
4. Περιγράψτε συνοπτικά τον γενετικό τελεστή της διασταύρωσης που θα χρησιμοποιήσετε.
5. Περιγράψτε συνοπτικά τον γενετικό τελεστή της μετάλλαξης που θα χρησιμοποιήσετε.

Απάντηση:

1. Κάθε ΤΝΔ θα αναπαρασταθεί ως ένα διάνυσμα (χρωμόσωμα) με πραγματικές τιμές, όπου κάθε γονίδιο του χρωμοσώματος θα αναπαριστά (θα αντιστοιχεί) και σε ένα βάρος ή πόλωση του ΝΔ. Π.χ. μια αναπαράσταση που μπορεί να χρησιμοποιηθεί είναι η $(w_{31}, w_{32}, b_3, w_{41}, w_{42}, b_4, w_{53}, w_{54}, b_5)$.

(Ένα ενδιαφέρον σημείο όπου θα μπορούσε να επεκταθεί κανείς είναι το να περιγράψει τι θα άλλαζε αν είχαμε ακέραιες τιμές στα βάρη.)

Επειδή τα βάρη του ΤΝΔ παίρνουν πραγματικές τιμές, ενώ δεν υπάρχει κάποιο συγκεκριμένο διάστημα μέσα στο οποίο «κινούνται», για την αναπαράσταση των πιθανών λύσεων δεν θα χρησιμοποιηθεί δυαδική αναπαράσταση, καθώς:

- εξαιτίας του γεγονότος ότι δεν υπάρχει κάποιο συγκεκριμένο διάστημα μέσα στο οποίο να παίρνουν τιμές οι ελεύθεροι παράμετροι, δεν είναι δυνατόν να τις αναπαραστήσουμε με δυαδικούς αριθμούς,

- ακόμα και αν ορίζαμε ένα τέτοιο διάστημα τιμών και κάποια επιθυμητή ακρίβεια, η δυαδική αναπαράσταση σε ένα μεγάλο ΤΝΔ θα οδηγούσε σε τεράστια σε μήκος χρωμοσώματα.

2. Αρχικοποίηση: Κάθε μέλος του αρχικού πληθυσμού θα δημιουργηθεί τυχαία. Θα ορίσουμε ένα συγκεκριμένο διάστημα, π.χ. $[-0.5, 0.5]$, μέσα από το οποίο με τυχαίο τρόπο θα αρχικοποιήσουμε τις τιμές των ελεύθερων παραμέτρων του ΤΝΔ για κάθε μέλος του αρχικού πληθυσμού.

3. Συνάρτηση Αξιολόγησης: Μπορεί να χρησιμοποιηθεί οποιοδήποτε μέτρο του λάθους του ΤΝΔ που αναπαριστά κάθε μέλος του πληθυσμού. Μία καλή τέτοια συνάρτηση είναι κάποια θετική συνάρτηση που να είναι αντιστρόφως ανάλογη του MSE. Το λάθος μπορεί να υπολογιστεί δεδομένης της ύπαρξης ενός συνόλου δεδομένων εκπαίδευσης της μορφής (x,t) , όπου $x \in \mathbf{R}^2$ είναι η είσοδος του ΤΝΔ και όπου $t \in \mathbf{R}$ η επιθυμητή έξοδος του ΤΝΔ.

4. Διασταύρωση: Για τη συγκεκριμένη αναπαράσταση που έχουμε επιλέξει μπορούμε να χρησιμοποιήσουμε οποιοδήποτε τελεστή διασταύρωσης, όπως π.χ. τη διασταύρωση ενός σημείου ή τη διασταύρωση δύο σημείων ή τη διασταύρωση με μάσκα κλπ.

5. Μετάλλαξη: Για τη μετάλλαξη μπορούμε να επιλέγουμε με τυχαίο τρόπο έναν αριθμό (όπως αυτός καθορίζεται από την pm) από γονίδια (ελεύθερες παραμέτρους του ΤΝΔ) και να τα τροποποιούμε είτε προσθέτοντας στην υπάρχουσα τιμή μία μικρή τυχαία τιμή, είτε παράγοντας μία τυχαία τιμή σε ένα συγκεκριμένο διάστημα τιμών και αντικαθιστώντας την υπάρχουσα.

Άσκηση 9/2:

A. Τα Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ) με βηματικούς νευρώνες και ακέραια βάρη έχουν ιδιαίτερο ενδιαφέρον. Να αναφέρετε ποια θεωρείτε πως είναι μερικά πλεονεκτήματά τους σε σχέση με τα συνήθη ΤΝΔ.

Απάντηση:

Τα ΤΝΔ μπορούν να προγραμματιστούν με τη χρήση Η/Υ, όμως για να μπορέσουν να χρησιμοποιηθούν σε πραγματικές εφαρμογές, όπου απαιτείται πολύ μικρός χρόνος εκτέλεσης, καθώς και σε εφαρμογές πραγματικού χρόνου, πολλές φορές είναι απαραίτητη η υλοποίηση σε υλικό (hardware).

Το πρόβλημα είναι ότι οι περισσότεροι αλγόριθμοι εκπαίδευσης ΤΝΔ παράγουν δίκτυα που έχουν βάρη πραγματικούς αριθμούς. Η υλοποίηση σε υλικό τέτοιων δικτύων είναι πιο δύσκολη και ακριβή. Ένα δεύτερο πρόβλημα είναι η αποθήκευση των πραγματικών βαρών, αφού για να μπορεί το ΤΝΔ να λειτουργήσει σωστά απαιτείται αποθήκευση πολλών δεκαδικών ψηφίων τους.

Αντιθέτως τα ΤΝΔ που έχουν εκπαιδευτεί με ακέραια βάρη είναι πιο εύκολο να υλοποιηθούν, τόσο με ηλεκτρονικά, όσο και με οπτικά κυκλώματα, και η αποθήκευση των ακέραιων βαρών είναι πολύ πιο απλή και γρήγορη. Τέλος γίνονται μόνο πράξεις μεταξύ ακεραίων και απλή κατωφλίωση (ο υπολογισμός της σιγμοειδούς στο υλικό έχει πολύ μεγαλύτερο υπολογιστικό κόστος).

B. Για να εκπαιδύσουμε ένα ΤΝΔ με βηματικούς νευρώνες και ακέραια βάρη θα προτιμήσουμε τη μέθοδο της οπισθοδιάδοσης των βαρών ή τη χρήση Γενετικών Αλγορίθμων;

Απάντηση:

Η μέθοδος οπισθοδιάδοσης των βαρών απαιτεί την παράγωγο της συνάρτησης ενεργοποίησης. Όμως η βηματική συνάρτηση δεν είναι παραγωγίσιμη στο 0 (και είναι 0 στο υπόλοιπο πεδίο ορισμού της) και συνεπώς η μέθοδος δεν εφαρμόζεται. Επιπρόσθετα, τα ΤΝΔ που δημιουργεί έχουν βάρη πραγματικούς αριθμούς. Αντιθέτως οι ΓΑ δεν απαιτούν την παράγωγο της συνάρτησης ενεργοποίησης και εγγενώς μπορούν να βελτιστοποιήσουν ακέραιες αντικειμενικές συναρτήσεις. Συνεπώς θα επιλέξουμε τους ΓΑ για την εκπαίδευση του ΤΝΔ.

Γ. Θέλουμε με χρήση ΓΑ να εκπαιδύσουμε ένα ΤΝΔ για το πρόβλημα του XOR, με βηματικούς νευρώνες, ακέραια βάρη, και 2-2-1 αρχιτεκτονική. Επιπρόσθετα, περιορίζουμε τις δυνατές τιμές των βαρών στο σύνολο $B = \{-2, 2\}$ και των πολώσεων στο σύνολο $\Pi = \{-3, -2, -1, 0\}$. Να προτείνετε μια πιθανή κωδικοποίηση (να δώσετε

και ένα παράδειγμα), τη συνάρτηση αξιολόγησης, τους τελεστές διασταύρωσης και μετάλλαξης, καθώς και τη συνθήκη τερματισμού.

Απάντηση:

Αφού έχουμε 2-2-1 αρχιτεκτονική, το ΤΝΔ έχει 6 βάρη και 3 πολώσεις. Για την κωδικοποίηση κάθε βάρους απαιτεί ένα bit και κάθε πολώση δύο bits. Συνεπώς, αρκεί να δημιουργήσουμε χρωμοσώματα με 12 bits (δυναδική αναπαράσταση).

Ένα παράδειγμα είναι το παρακάτω: Θα κάνουμε τις εξής αντιστοιχίσεις: $-2 \rightarrow 0$, $2 \rightarrow 1$, $0 \rightarrow 00$, $-1 \rightarrow 01$, $-2 \rightarrow 10$, $-3 \rightarrow 11$. Έστω ότι σε ένα άτομο του πληθυσμού το διάνυσμα βαρών και κατωφλίων (φαινότυπος) είναι:

$$B1 = [2, -2, -2, -2, 2, 2, -3, -1, -2].$$

Τότε ο αντίστοιχος γενότυπος (χρωμόσωμα) θα είναι:

$$B1 = [1, 0, 0, 0, 1, 1, (1, 1), (0, 1), (1, 0)].$$

Ως συνάρτηση καταλληλότητας μπορεί να χρησιμοποιηθεί ο αριθμός των σωστών εξόδων του δικτύου (δηλαδή αν υλοποιεί την πύλη XOR) που αντιστοιχεί σε κάθε χρωμόσωμα. Οι συναρτήσεις διασταύρωσης και μετάλλαξης δεν απαιτούν τροποποίηση. Τέλος, επειδή γνωρίζουμε την καταλληλότητα της βέλτιστης λύσης (στο πρόβλημά μας ίση με 4), μπορούμε να τη χρησιμοποιήσουμε ως συνθήκη τερματισμού.

Κεφάλαιο 10

Γενετικός Προγραμματισμός (ΓΠ)

Σύνοψη

Στα προηγούμενα κεφάλαια είδαμε ότι οι Γενετικοί Αλγόριθμοι, οι οποίοι είναι εμπνευσμένοι από τη διαδικασία της φυσικής εξέλιξης, αντιμετωπίζουν με επιτυχία μια μεγάλη ποικιλία προβλημάτων. Αυτό οφείλεται στο γεγονός ότι δίνουν τη δυνατότητα στον σχεδιαστή να δημιουργήσει δικές του κωδικοποιήσεις και νέους γενετικούς τελεστές, που είναι προσαρμοσμένοι στις απαιτήσεις ενός συγκεκριμένου προβλήματος· πρέπει όμως να τηρούν και τις βασικές αρχές των Γενετικών Αλγορίθμων.

Σκοπός του συγκεκριμένου κεφαλαίου είναι να δείξει ότι αυτή η επέκταση μπορεί να προχωρήσει ένα βήμα παραπέρα, δηλαδή στην (εμπνευσμένη από τη φύση) εξέλιξη προγραμμάτων ηλεκτρονικών υπολογιστών, τα οποία εκτελούν ένα έργο καθορισμένο από τον χρήστη. Η αυτοματοποιημένη μεθοδολογία εύρεσης προγραμμάτων υπολογιστών (κώδικα) που βασίζεται στις αρχές της βιολογικής εξέλιξης ονομάζεται Γενετικός Προγραμματισμός (ΓΠ)· με άλλα λόγια πρόκειται για μια τεχνική Μηχανικής Μάθησης, η οποία χρησιμοποιεί έναν Εξελικτικό Αλγόριθμο για να βελτιστοποιήσει έναν πληθυσμό από προγράμματα υπολογιστή σύμφωνα με μια συνάρτηση καταλληλότητας που καθορίζεται από την ικανότητα του προγράμματος να εκτελέσει ένα δοσμένο υπολογιστικό έργο. Η παρουσίαση κατάλληλων γενετικών λειτουργιών (όπως οι τελεστές αναπαραγωγής, διασταύρωσης και μετάλλαξης), αλλά και λειτουργιών μεταβολής της αρχιτεκτονικής προγράμματος (δηλαδή αλλαγές στη δομή του κώδικα, όπως η διαγραφή ή η προσθήκη ορίσματος ή υπορουτίνας και άλλων λειτουργιών) που έχουν εμπνευστεί από τον πολλαπλασιασμό και την εξάλειψη γονιδίων στη φύση και η εφαρμογή τους στην ανάπτυξη μεθόδων ΕΠ, αποτελούν τους βασικούς στόχους αυτού του κεφαλαίου. Ιδιαίτερη έμφαση θα δοθεί στις λειτουργίες μεταβολής της αρχιτεκτονικής προγράμματος, επειδή αποτελούν ένα νέο ιδιαίτερο χαρακτηριστικό του ΓΠ, το οποίο δεν χρησιμοποιούν οι άλλες κατηγορίες Εξελικτικών Αλγορίθμων.

Προσδοκώμενα Αποτελέσματα

Όταν ολοκληρωθεί η μελέτη αυτού του κεφαλαίου, θα μπορείτε να:

- διακρίνετε τη διαφορά μεταξύ των Γενετικών Αλγορίθμων και του Γενετικού Προγραμματισμού,
- ορίσετε τερματικά, συναρτήσεις και τις κατάλληλες παραμέτρους,
- σχεδιάσετε/προσαρμόσετε τους γενετικούς τελεστές που χρειάζονται για το πρόβλημα,
- χρησιμοποιήσετε τις λειτουργίες μεταβολής της αρχιτεκτονικής ενός προγράμματος,
- σχεδιάσετε την επίλυση ενός προβλήματος προγραμματισμού με τη βοήθεια ΓΠ,
- υλοποιήσετε την επίλυση σε Η/Υ.

Λέξεις-Κλειδιά:

- Σύνολο τερματικών
- Συναρτήσεις μηδενικού ορίσματος
- Μετατροπή αρχιτεκτονικής προγράμματος
- Εξέλιξη προγραμμάτων
- Δένδρο αναπαραστάσης

Προαπαιτούμενη γνώση

Κεφάλαιο 8, 9.

10.1 Εισαγωγικές Παρατηρήσεις

Μία από τις κορυφαίες προκλήσεις της επιστήμης των υπολογιστών είναι να βρει έναν υπολογιστή που να εκτελεί ό,τι χρειάζεται να γίνει, χωρίς να του λείπει κάποιος πώς να το κάνει. Ο Γενετικός Προγραμματισμός αντιμετωπίζει αυτήν την πρόκληση παρέχοντας μία μέθοδο για αυτόματη λειτουργία λειτουργικών προγραμμάτων υπολογιστή από έναν υψηλού επιπέδου ορισμό του προβλήματος. Ο ΓΠ επιτυγχάνει τον συγκεκριμένο στόχο αυτόματου προγραμματισμού (ο οποίος μερικές φορές ονομάζεται σύνθεση προγράμματος ή εισαγωγή προγράμματος) εξελίσσοντας γενετικά έναν πληθυσμό από προγράμματα υπολογιστή, χρησιμοποιώντας τις αρχές της φυσικής επιλογής της Δαρβινικής Θεωρίας της Εξέλιξης και λειτουργίες εμπνευσμένες από τη Βιολογία. Αυτές οι λειτουργίες περιλαμβάνουν αναπαραγωγή, διασταύρωση, μετάλλαξη και λειτουργίες τροποποίησης της αρχιτεκτονικής που έχουν αντιγραφεί από τις αντίστοιχες λειτουργίες των γονιδίων.

Ο ΓΠ είναι μία μέθοδος ανεξάρτητη από το πεδίο ορισμού του προβλήματος, η οποία αναπαράγει γενετικά έναν πληθυσμό από προγράμματα υπολογιστή για να λύσει ένα πρόβλημα. Πιο συγκεκριμένα, ο ΓΠ μετασχηματίζει επαναληπτικά έναν πληθυσμό από προγράμματα υπολογιστή σε μια νέα γενιά προγραμμάτων εφαρμόζοντας λειτουργίες ανάλογες με τις γενετικές λειτουργίες που συμβαίνουν στη φύση. Εδώ πρέπει να τονίσουμε ότι τόσο ο ΓΠ όσο και οι ΓΑ είναι δύο διαφορετικές κατηγορίες Εξελικτικών Αλγορίθμων. Οι ΓΑ επεξεργάζονται κωδικοποιημένες συμβολοσειρές, οι οποίες αντιπροσωπεύουν περιστασιακές (particular) λύσεις του προβλήματος. Αυτές οι κωδικοποιημένες συμβολοσειρές τρέχουν σε έναν προσομοιωτή και οι καλύτερες συμβολοσειρές αναμειγνύονται για να σχηματίσουν μια νέα γενιά. Ο ΓΠ ακολουθεί μια διαφορετική προσέγγιση. Αντί να κωδικοποιεί την αναπαράσταση μιας λύσης, εξελίσσει εκτελέσιμα προγράμματα υπολογιστή.

Στον ΓΠ απαιτούνται δύο είδη βημάτων, τα προκαταρκτικά και τα εκτελεστικά. Ο χρήστης εισάγει τον υψηλού επιπέδου ορισμό του προβλήματος στο σύστημα ΓΠ, εκτελώντας κάποια καλά ορισμένα προκαταρκτικά βήματα. Τα πέντε κύρια προκαταρκτικά βήματα για τη βασική έκδοση του ΓΠ που απαιτείται να καθορίσει ο χρήστης είναι:

1. το σύνολο των τερματικών (π.χ. οι ανεξάρτητες μεταβλητές του προβλήματος, οι σταθερές και συναρτήσεις μηδενικού ορίσματος [zero-argument]) για κάθε κλάδο του προς εξέλιξη προγράμματος,
2. το σύνολο των αρχικών (στοιχειωδών) συναρτήσεων για κάθε κλάδο του προς εξέλιξη προγράμματος,
3. το μέτρο καταλληλότητας (για άμεση ή έμμεση μέτρηση της καταλληλότητας των ατόμων του πληθυσμού),
4. κάποιες παραμέτρους για τον έλεγχο του τρεξίματος,
5. το κριτήριο τερματισμού και τη μέθοδο προσδιορισμού του αποτελέσματος του τρεξίματος.

Ο ΓΠ αρχίζει τυπικά με έναν πληθυσμό από τυχαία παραγόμενα προγράμματα υπολογιστή, συγκροτημένα από τα διαθέσιμα προγραμματιστικά στοιχεία. Όπως έχουμε ήδη αναφέρει, ο ΓΠ μετατρέπει επαναληπτικά έναν πληθυσμό από προγράμματα υπολογιστή σε μία νέα γενιά του πληθυσμού εφαρμόζοντας γενετικούς τελεστές ανάλογους με αυτούς που συμβαίνουν στη φύση. Αυτοί οι τελεστές εφαρμόζονται σε άτομα που επιλέγονται από τον πληθυσμό. Τα άτομα επιλέγονται για να συμμετέχουν στις γενετικές λειτουργίες με πιθανοτικό τρόπο, βασιζόμενο στην καταλληλότητά τους (όπως έχει μετρηθεί από το μέτρο καταλληλότητας που παρέχεται από τον χρήστη στο τρίτο προκαταρκτικό βήμα). Ο επαναληπτικός μετασχηματισμός του πληθυσμού εκτελείται μέσα στον κύριο γενικευμένο βρόχο τρεξίματος του ΓΠ.

Τα εκτελεστικά βήματα του ΓΠ (με άλλα λόγια το διάγραμμα ροής του ΓΠ) είναι τα ακόλουθα:

1. Δημιουργήστε τυχαία έναν αρχικό πληθυσμό (γενιά 0) από άτομα (προγράμματα υπολογιστή που σχηματίζονται από τις διαθέσιμες συναρτήσεις και τερματικά).
2. Εκτελέστε επαναληπτικά τα παρακάτω επιμέρους βήματα:
 - α. Εκτελέστε κάθε πρόγραμμα του πληθυσμού και εξακριβώστε την καταλληλότητά του (άμεσα ή έμμεσα) χρησιμοποιώντας το μέτρο καταλληλότητας (δηλαδή την αντικειμενική συνάρτηση) του προβλήματος.
 - β. Επιλέξτε ένα ή δύο άτομα-προγράμματα από τον πληθυσμό με μια πιθανότητα που βασίζεται στην καταλληλότητα (η επανεπιλογή επιτρέπεται) για να συμμετάσχουν στο επόμενο βήμα.

γ. Δημιουργήστε νέα άτομα-προγράμματα για τον πληθυσμό εφαρμόζοντας τους ακόλουθους γενετικούς τελεστές, με προκαθορισμένες τιμές πιθανότητας:

- **Αναπαραγωγή:** Αντιγράψτε τα επιλεγμένα άτομα-προγράμματα στον νέο πληθυσμό.
- **Διασταύρωση:** Δημιουργήστε νέους απογόνους-προγράμματα για τον νέο πληθυσμό, ανασυνδυάζοντας τυχαία επιλεγμένα τμήματα από τα δύο επιλεγμένα προγράμματα.
- **Μετάλλαξη:** Δημιουργήστε έναν νέο απόγονο-πρόγραμμα για τον νέο πληθυσμό, μεταλλάσσοντας τυχαία ένα τυχαία επιλεγμένο τμήμα ενός από τα προγράμματα που επιλέξατε.
- **Λειτουργίες μετατροπής αρχιτεκτονικής:** Επιλέξτε μία λειτουργία μετατροπής αρχιτεκτονικής από τη διαθέσιμη συλλογή τέτοιων λειτουργιών και δημιουργήστε έναν νέο απόγονο-πρόγραμμα για τον νέο πληθυσμό, εφαρμόζοντας την επιλεγμένη μέθοδο μετατροπής σε ένα επιλεγμένο πρόγραμμα.

3. Όταν ικανοποιηθεί το κριτήριο τερματισμού, το μοναδικό καλύτερο πρόγραμμα στον πληθυσμό που παράγεται μετά την εκτέλεση (το καλύτερο μέχρι τώρα άτομο) συλλέγεται και χαρακτηρίζεται ως το αποτέλεσμα της εκτέλεσης. Αν η εκτέλεση είναι επιτυχής, το αποτέλεσμα μπορεί να είναι μια λύση (ή μια προσέγγιση της λύσης) του προβλήματος.

10.2 Εισαγωγή στον Γενετικό Προγραμματισμό

Σκοπός

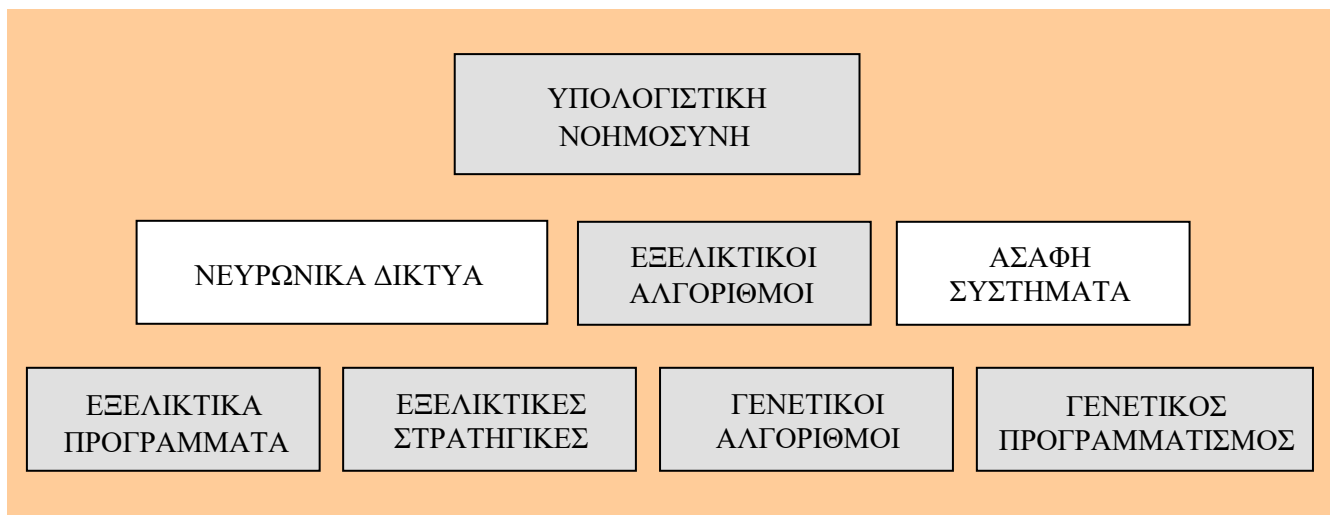
Σκοπός αυτής της ενότητας είναι αφενός να κάνει τη σύνδεση του ΓΠ με τη Μηχανική Μάθηση και αφετέρου να παρουσιάσει τα δομικά στοιχεία ενός Αλγορίθμου ΓΠ· πιο συγκεκριμένα να γίνει αναλυτική παρουσίαση του συνόλου των τερματικών, των συναρτήσεων και των δομών των εκτελέσιμων προγραμμάτων. Στη συνέχεια θα αναφερθούμε στην αρχικοποίηση του πληθυσμού και τη μελέτη ειδικών γενετικών τελεστών που χρησιμοποιούνται στον ΓΠ. Τέλος θα γίνει η παρουσίαση του βασικού αλγορίθμου ΕΠ. Με τη βοήθεια παραδειγμάτων, ασκήσεων αυτοαξιολόγησης και δραστηριοτήτων ο αναγνώστης θα αποκτήσει τις απαραίτητες δεξιότητες για τον σχεδιασμό και την υλοποίηση απλών εξελικτικών προγραμμάτων.

Προσδοκώμενα Αποτελέσματα

Όταν θα έχετε ολοκληρώσει τη μελέτη αυτής της ενότητας, θα μπορείτε να:

- ορίσετε τα τερματικά και τις συναρτήσεις για ένα πρόβλημα, προκειμένου να επιλυθεί με ΓΠ,
- σχεδιάσετε τους γενετικούς τελεστές για ένα αντίστοιχο πρόβλημα,
- υλοποιήσετε τους παραπάνω γενετικούς τελεστές,
- υλοποιήσετε τον βασικό αλγόριθμο ΓΠ,
- αξιολογήσετε τα αποτελέσματα.

Η διαδικασία ανάπτυξης προγραμμάτων για την επίλυση προβλημάτων παραμένει μια δύσκολη και χρονοβόρα διαδικασία, ακόμα και σήμερα. Εξαιτίας αυτού γίνεται μεγάλη έρευνα να ανακαλυφθούν τρόποι αυτοπρογραμματισμού του υπολογιστή· δηλαδή, αντίθετα με την κλασική προσέγγιση, όπου περιγράφουμε λεπτομερώς πώς θα επιλυθεί κάποιο πρόβλημα, γίνεται προσπάθεια να αποκτήσουν οι υπολογιστές την ικανότητα να μαθαίνουν χρησιμοποιώντας την εμπειρία και τα παραδείγματα. Τυπικά μηχανική μάθηση «είναι η μελέτη αλγορίθμων, οι οποίοι βελτιώνονται αυτόματα, μέσω της εμπειρίας». Από τη δεκαετία του 1960 έχουν αναπτυχθεί αρκετές μέθοδοι για την επίτευξη αυτού του σκοπού. Οι μέθοδοι που είναι εμπνευσμένες από τη φύση, όπως έχει ήδη αναφερθεί, εντάσσονται στην περιοχή της Υπολογιστικής Νοημοσύνης και μία από αυτές είναι ο ΓΠ. Μία κατηγοριοποίηση των συγκεκριμένων τεχνικών φαίνεται στο παρακάτω σχήμα.



Σχήμα 10.1 Κατηγοριοποίηση των τεχνικών Υπολογιστικής Νοημοσύνης.

Όλα τα διαφορετικά συστήματα μηχανικής μάθησης ακολουθούν γενικά μία κοινή διαδικασία. Καταρχήν επιλέγονται τα χαρακτηριστικά του πεδίου μάθησης, τα οποία θα είναι η είσοδος του αλγορίθμου και ορίζεται η επιθυμητή έξοδος του. Π.χ. αν το πεδίο εφαρμογής είναι το χρηματιστήριο, οι τιμές μιας μετοχής τις τελευταίες 30 ημέρες μπορεί να είναι η είσοδος. Αντίστοιχα η πρόβλεψη της αυριανής τιμής της ίδιας μετοχής μπορεί να είναι η έξοδος. Κάθε σύστημα μηχανικής μάθησης χρησιμοποιεί μια αναπαράσταση της λύσης στο πρόβλημα. Η αναπαράσταση αυτή ορίζει τον χώρο των πιθανών λύσεων. Έπειτα χρησιμοποιείται κάποια μέθοδος αναζήτησης σε αυτόν τον χώρο, έτσι ώστε να είναι δυνατή η εύρεση μιας καλής λύσης στο πρόβλημα. Στο παραπάνω παράδειγμα η λύση θα μπορούσε να αναπαρασταθεί ως μια συνάρτηση της μορφής $f(x) = ax^2 + bx + c$. Άρα ο χώρος των λύσεων είναι όλα τα πολυώνυμα δευτέρου βαθμού. Οπότε η αναζήτηση θα γίνει για να βρεθούν αυτές οι τιμές των a , b , c , για τις οποίες είναι αποδεκτή η πρόβλεψη. Στις επόμενες παραγράφους παρουσιάζεται ο τρόπος που υλοποιεί ο ΓΠ αυτήν τη διαδικασία.

Το 1992 ο John Koza παρουσίασε μια μέθοδο αυτόματης παραγωγής προγραμμάτων η οποία βασίζεται στους γενετικούς αλγορίθμους και την ονόμασε Γενετικό Προγραμματισμό. Τα βασικά στοιχεία αυτής της μεθόδου, όπως παρουσιάζονται στο Koza, 1992, είναι:

- **Στοχαστική λήψη αποφάσεων.** Ο ΓΠ χρησιμοποιεί ψευδο-τυχαίους αριθμούς για να μιμηθεί την τυχειότητα της διαδικασίας της φυσικής εξέλιξης. Ως αποτέλεσμα ο ΓΠ χρησιμοποιεί στοχαστικές διαδικασίες και πιθανοθεωρητική λήψη αποφάσεων σε διάφορα στάδια της εξέλιξης προγραμμάτων.
- **Δομή προγραμμάτων.** Ο ΓΠ συναρμολογεί μεταβλητού μήκους δομές προγραμμάτων από βασικές μονάδες, οι οποίες ονομάζονται *συναρτήσεις* και *τερματικά*. Οι συναρτήσεις εκτελούν διεργασίες στις εισόδους τους, οι οποίες είναι τερματικά ή έξοδοι άλλων συναρτήσεων. Η πραγματική συναρμολόγηση των προγραμμάτων από συναρτήσεις και τερματικά γίνεται στο στάδιο της αρχικοποίησης του πληθυσμού.
- **Γενετικοί τελεστές.** Ο ΓΠ μετασχηματίζει τα αρχικά προγράμματα, χρησιμοποιώντας τους γενετικούς τελεστές. Η διασταύρωση μεταξύ δύο ατόμων είναι ο βασικός τελεστής στον ΓΠ. Άλλοι σημαντικοί τελεστές είναι η αναπαραγωγή και η μετάλλαξη.
- **Προσομοιωμένη εξέλιξη ενός πληθυσμού, μέσω της επιλογής βάσει της ικανότητας.** Ο ΓΠ εξελίσσει έναν πληθυσμό από προγράμματα παράλληλα. Η κινητήριος δύναμη της προσομοιωμένης εξέλιξης είναι κάποιου είδους επιλογή, η οποία βασίζεται στην ικανότητα. Αυτή η διαδικασία ορίζει ποια άτομα επιλέγονται για περαιτέρω βελτιώσεις.

Ο ΓΠ αρχίζει από έναν υψηλού επιπέδου ορισμό των απαιτήσεων του προβλήματος και προσπαθεί να παραγάγει ένα πρόγραμμα υπολογιστή που λύνει το πρόβλημα. Ο χρήστης τροφοδοτεί τον υψηλού επιπέδου ορισμό του προβλήματος στο σύστημα ΓΠ, εκτελώντας τα πέντε καλά ορισμένα προπαρασκευαστικά βήματα

που αναφέρθηκαν στις εισαγωγικές παρατηρήσεις. Τα σύνολα των τερματικών και των αρχικών συναρτήσεων αποτελούν τα συστατικά τα οποία είναι διαθέσιμα για να κατασκευαστεί ένα πρόγραμμα υπολογιστή. Ένα τρέξιμο του ΓΠ συνιστά μια ανταγωνιστική αναζήτηση μεταξύ διαφορετικών προγραμμάτων που είναι μέλη ενός πληθυσμού, ο οποίος συντίθενται από διαθέσιμες συναρτήσεις και τερματικά.

Ο προσδιορισμός του συνόλου των συναρτήσεων και των τερματικών για ένα ειδικό πρόβλημα (ή μια κατηγορία προβλημάτων) είναι συνήθως μια απευθείας διαδικασία. Για κάποια προβλήματα το σύνολο των συναρτήσεων μπορεί να αποτελείται τόσο από αριθμητικές συναρτήσεις (πρόσθεση, αφαίρεση, πολλαπλασιασμού και διαίρεσης), όσο και από τελεστή υπό συνθήκη διακλάδωσης. Το σύνολο τερματικών μπορεί να αποτελείται από τις εξωτερικές εισόδους του προγράμματος (ανεξάρτητες μεταβλητές) και αριθμητικές σταθερές. Αυτό το σύνολο τερματικών και συναρτήσεων είναι χρήσιμο για ευρεία ποικιλία προβλημάτων — και αντιστοιχεί στις βασικές λειτουργίες που συναντάμε εικονικά σε κάθε γενικού σκοπού ψηφιακό υπολογιστή. Για πολλά άλλα προβλήματα τα συστατικά περιλαμβάνουν εξειδικευμένες συναρτήσεις και τερματικά, όπως φαίνεται στο παρακάτω παράδειγμα.

Παράδειγμα 10.1

Έστω ότι ο στόχος είναι να χρησιμοποιήσουμε Γενετικό Προγραμματισμό για να προγραμματίσουμε αυτόματα ένα ρομπότ να σφουγγαρίσει όλο το πάτωμα ενός δωματίου γεμάτου με εμπόδια. Ο χρήστης πρέπει να πει στον ΓΠ τι είναι ικανό να κάνει το ρομπότ. Π.χ. το ρομπότ μπορεί να είναι ικανό να εκτελεί λειτουργίες κίνησης, στροφής και χρήσης της σφουγγαρίστρας. Αν ο στόχος είναι να φτιάξουμε έναν ελεγκτή, το σύνολο συναρτήσεων μπορεί να αποτελείται από συναρτήσεις επεξεργασίας σήματος, που λειτουργούν σε σήματα στο πεδίο του χρόνου, περιλαμβάνοντας στοιχεία ολοκλήρωσης, διαφορίσης, καθυστέρησης, κέρδους, πρόσθεσης και αφαίρεσης. Το σύνολο των τερματικών μπορεί να αποτελείται από σήματα, όπως το σήμα αναφοράς και η έξοδος του συστήματος. Όταν ο χρήστης καθορίσει τα αρχικά συστατικά για ένα πρόβλημα σύνθεσης ελεγκτή, το ίδιο σύνολο τερματικών και συναρτήσεων μπορεί να χρησιμοποιηθεί για την αυτόματη σύνθεση μιας μεγάλης ποικιλίας διαφορετικών ελεγκτών.

Άσκηση 10.1

Έστω ότι θέλετε να σχεδιάσετε αυτόματα ένα ηλεκτρικό κύκλωμα με τη βοήθεια ΓΠ. Να ορίσετε το σύνολο των τερματικών και των συναρτήσεων που καθιστούν ικανό τον ΓΠ να σχεδιάσει αυτόματα το πρόγραμμα. Να αναφέρετε πέντε άλλες εφαρμογές, στις οποίες μπορεί να χρησιμοποιηθεί το σύνολο τερματικών και συναρτήσεων που ορίσατε.

Θα ολοκληρώσουμε την εισαγωγή στον ΓΠ με το πρόβλημα της αναπαράστασης. Όπως έχουμε ήδη αναφέρει, στους ΓΑ, ΕΣ και ΕΠ η αναπαράσταση των ατόμων του πληθυσμού, δηλαδή τα χρωμοσώματα, είναι γραμμικές δομές (ψηφιοσειρές, ακολουθίες ακέραιων αριθμών, διανύσματα πραγματικών αριθμών, μεταθέσεις). Στον ΓΠ τα χρωμοσώματα είναι σε μορφή δένδρου, δηλαδή μη-γραμμική δομή. Στους ΓΑ, ΕΣ, ΕΠ το μέγεθος του χρωμοσώματος είναι (συνήθως) σταθερό (fixed). Τα δένδρα στο ΓΠ μπορεί να ποικίλουν σε βάθος και πλάτος — κάτι που συνήθως ισχύει.

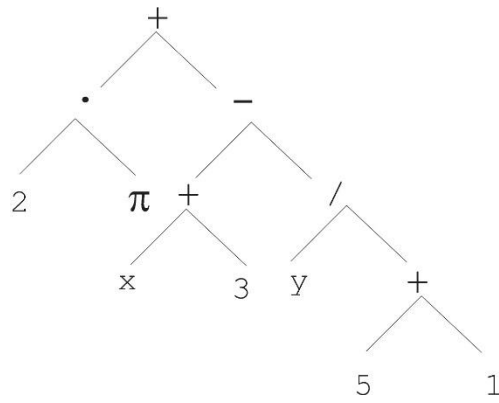
Τα δένδρα που είναι η βασική μορφή αναπαράστασης στον ΓΠ μπορεί να αναπαριστούν μία αριθμητική εξίσωση, έναν λογικό τύπο ή ένα πρόγραμμα και ουσιαστικά αναπαριστούν τη συντακτική τους ανάλυση.

Παράδειγμα 10.2

1. Η αναπαράσταση σε μορφή δένδρου της αριθμητικής παράστασης:

$$2 \cdot \pi + \left((x + 3) - \frac{y}{5 + 1} \right)$$

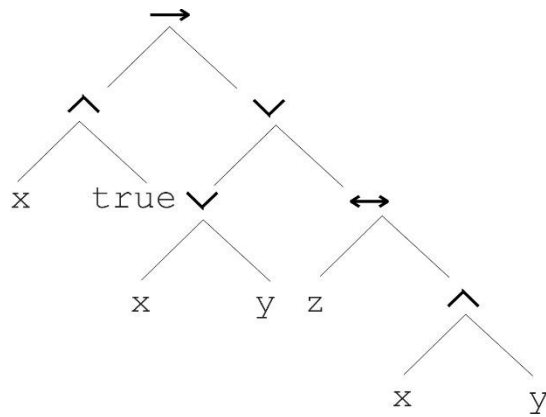
είναι η παρακάτω.



2. Η αναπαράσταση σε μορφή δένδρου του λογικού τύπου:

$$(x \wedge \text{true}) \rightarrow ((x \vee y) \vee (z \leftrightarrow (x \wedge y)))$$

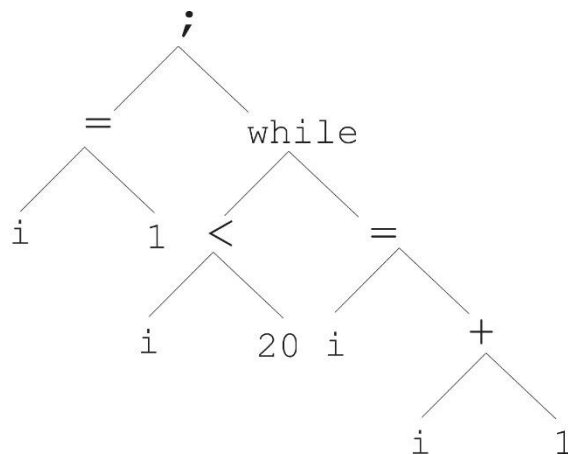
είναι η παρακάτω.



3. Η αναπαράσταση σε μορφή δένδρου του προγράμματος:

```
i=1;
while (i < 20)
{
    i=i+1
}
```

είναι η παρακάτω.



Συνοψίζοντας οι συμβολικές εκφράσεις καθορίζονται από:

- ένα σύνολο από Τερματικά (Terminal set) T ,
- ένα σύνολο από συναρτήσεις ή μη-τερματικά (Function or Non-Terminals set) F (κάθε συνάρτηση μπορεί να έχει διαφορετικό αριθμό ορισμάτων).

Γενικά ακολουθούνται οι εξής αναδρομικοί ορισμοί:

- Κάθε $t \in T$ είναι μια σωστή έκφραση.
- Η $f(e_1, \dots, e_n)$ είναι σωστή έκφραση εάν $f \in F$, $\text{arity}(f)=n$ και οι e_1, \dots, e_n είναι σωστές εκφράσεις, όπου $\text{arity}(f)$ είναι ο πληθάρηθος της f .
- Δεν υπάρχουν άλλοι τύποι σωστών εκφράσεων.

Άσκηση 10.2

Να αναπαραστήσετε σε μορφή δένδρου τις παρακάτω εκφράσεις:

α. $3(x+6)$.

β. $(x^2+1)/(x^2-1)$

γ. $(A=2) \text{ AND } (S>100)$

δ. $\text{If (FacingFood)((\{$

$\text{Move () ;$

$\}$

else {

$\text{Right () ;$

$\text{Move () ;$

$\}$

Άσκηση 10.3

Να αναφέρετε δέκα τρόπους αναπαράστασης, εκτός από τα δένδρα συντακτικής ανάλυσης, που αναφέρθηκαν στην ενότητα 10.1.

10.2.1 Το σύνολο των τερματικών

Το σύνολο των τερματικών αποτελείται από τις εισόδους στα προγράμματα του ΓΠ, δηλαδή τις μεταβλητές, τις σταθερές που μπορούν αυτά να χρησιμοποιήσουν και τις συναρτήσεις με μηδενικό αριθμό εισόδων. Γενικά τα τερματικά παρέχουν τιμές στα προγράμματα και οι συναρτήσεις επεξεργάζονται αυτές τις τιμές, για να δώσουν αποτέλεσμα. Ο όρος τερματικό χρησιμοποιείται γιατί, στον ΓΠ που εξελίσσει δενδρικές δομές, τα τερματικά είναι τα φύλλα κάθε δένδρου. Αυτό θα γίνει περισσότερο κατανοητό στην ενότητα για τη δομή των προγραμμάτων.

Είναι γνωστό ότι όλα τα συστήματα μηχανικής μάθησης χρησιμοποιούν εισόδους (χαρακτηριστικά) από το πεδίο μάθησης. Έτσι και στον ΓΠ οι εισοδοί του *συνόλου εκπαίδευσης* είναι μέρος του συνόλου των τερματικών· δηλαδή τα δεδομένα πάνω στα οποία ο αλγόριθμος εκπαιδεύεται. Η διαφορά με τις άλλες μεθόδους μηχανικής μάθησης είναι στην αναπαράσταση αυτών των χαρακτηριστικών. Ο ΓΠ χρησιμοποιεί αυτά τα χαρακτηριστικά ως δομικές μονάδες για τη δημιουργία προγραμμάτων.

Το σύνολο των τερματικών περιέχει επίσης και σταθερές. Στον τυπικό ΓΠ ένα σύνολο από πραγματικούς αριθμούς επιλέγεται εξ αρχής για όλον τον πληθυσμό. Οι τιμές αυτές δεν μεταβάλλονται κατά τη διάρκεια της εκτέλεσης του αλγορίθμου. Ονομάζονται *τυχαίες εφήμερες σταθερές* και συνήθως συμβολίζονται με το γράμμα R . Για τη δημιουργία νέων σταθερών μέσα στα προγράμματα συνδυάζονται οι τυχαίες εφήμερες σταθερές με αριθμητικές συναρτήσεις.

10.2.2 Το σύνολο των συναρτήσεων

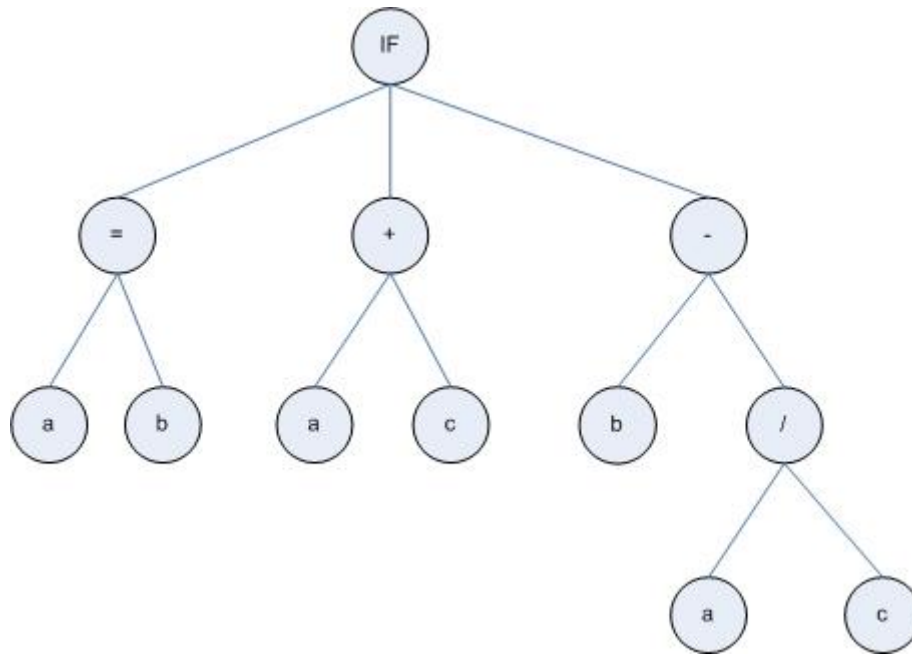
Το σύνολο των συναρτήσεων αποτελείται από τις εντολές, τους τελεστές και τις συναρτήσεις που μπορεί να χρησιμοποιήσει ο αλγόριθμος για την κατασκευή προγραμμάτων. Το εύρος των πιθανών συναρτήσεων είναι πάρα πολύ μεγάλο. Μπορούν να χρησιμοποιηθούν συναρτήσεις φτιαγμένες για το συγκεκριμένο πρόβλημα, αλλά και οποιαδήποτε δομή που μπορεί να κατασκευαστεί από μια γλώσσα προγραμματισμού. Ακολουθούν κάποια παραδείγματα:

- Λογικές συναρτήσεις
Π.χ. AND, OR, NOT, XOR.
- Αριθμητικές συναρτήσεις
Π.χ. +, -, × (ή *), ÷.
- Συναρτήσεις ανάθεσης μεταβλητών
Π.χ. ASSIGN(a, 1), όπου a είναι όνομα μεταβλητής.
- Εντολές ελέγχου ροής
Π.χ. IF, THEN, ELSE, CASE.
- Εντολές μεταφοράς ελέγχου
Π.χ. GOTO, CALL, JUMP.
- Εντολές επανάληψης
Π.χ. WHILE...DO, REPEAT...UNTIL, FOR.
- Υπορουτίνες

Το εύρος των συναρτήσεων είναι πολύ μεγαλύτερο από αυτό που παρουσιάστηκε στα προηγούμενα παραδείγματα. Οποιαδήποτε υπορουτίνα μπορεί να δημιουργηθεί από έναν προγραμματιστή έχει τη δυνατότητα να γίνει μέλος του συνόλου συναρτήσεων του ΓΠ. Π.χ. για τη λύση ενός προβλήματος ελέγχου κάποιου οχήματος είναι δυνατό να έχουν αναπτυχθεί οι συναρτήσεις «στρίψε δεξιά» ή «στρίψε αριστερά».

10.2.3 Δομές εκτελέσιμων προγραμμάτων

Τα τερματικά και οι συναρτήσεις πρέπει να συναρμολογηθούν σε μια δομή για να δημιουργηθεί ένα πρόγραμμα. Η πιο συνηθισμένη δομή που χρησιμοποιείται είναι το (δυαδικό) δένδρο. Στο σχήμα 10.3 που ακολουθεί παρουσιάζεται ένα παράδειγμα προγράμματος με τη δομή δένδρου. Τέτοια δένδρα, όπως ήδη έχουμε αναφέρει, αναπαριστούν τη συντακτική ανάλυση ενός προγράμματος, μιας αριθμητικής ή μιας λογικής έκφρασης.



Σχήμα 10.3 Δενδρική δομή εκτελέσιμου προγράμματος.

Το παραπάνω δένδρο θα μπορούσε να είναι η αναπαράσταση του παρακάτω προγράμματος:

```

IF a = b THEN
    RETURN a + c
ELSE
    RETURN b - (a / c)
  
```

Όπως αναφέρθηκε και στην ενότητα 10.1.1, κάθε μη τερματικός κόμβος είναι μια συνάρτηση και κάθε φύλλο του δένδρου είναι ένα τερματικό. Η σειρά εκτέλεσης των εντολών εξαρτάται από την εκάστοτε υλοποίηση. Άλλες μορφές δομών που χρησιμοποιούνται είναι οι γραμμικές δομές και τα γραφήματα.

10.2.4 Αρχικοποίηση του πληθυσμού

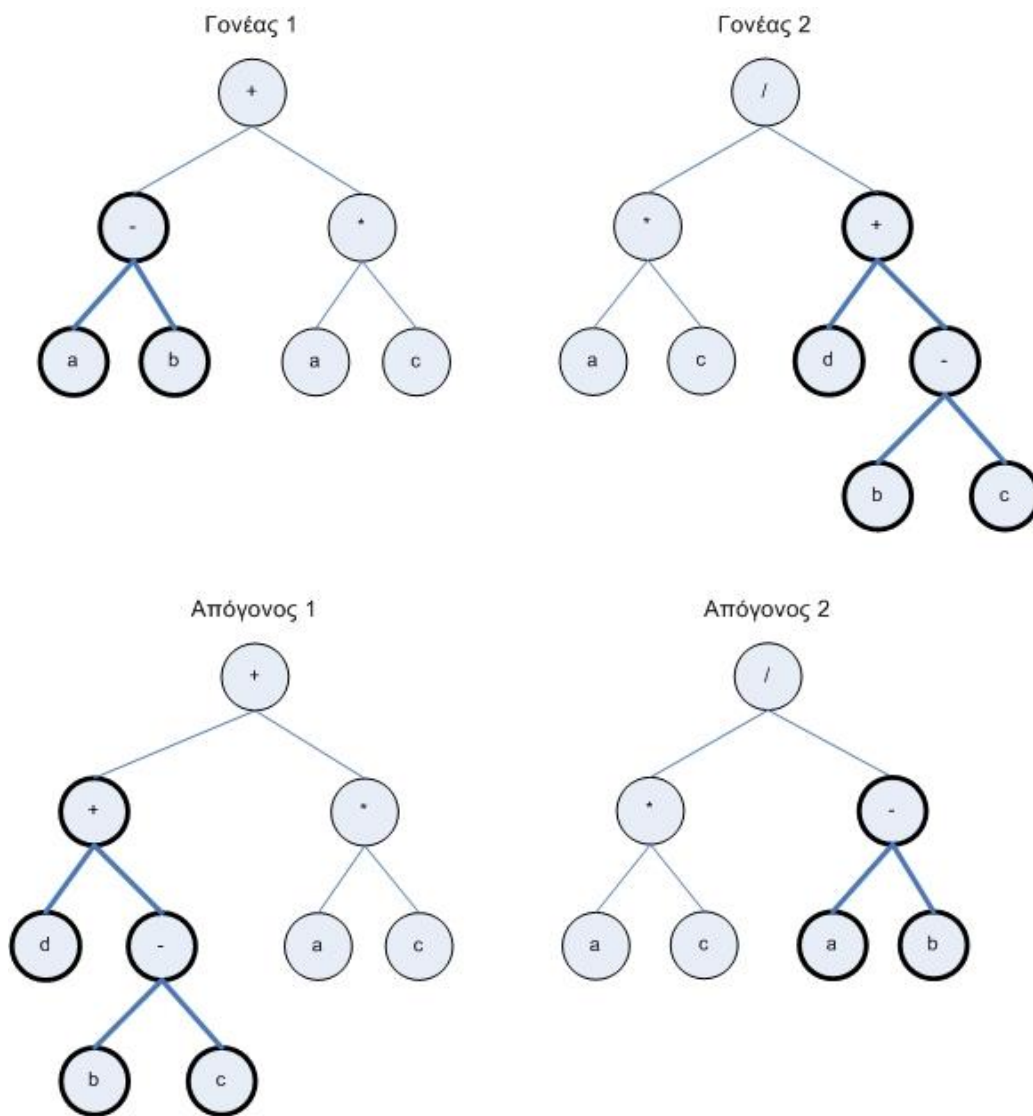
Το πρώτο βήμα της εκτέλεσης του αλγορίθμου του ΓΠ είναι η αρχικοποίηση του πληθυσμού· δηλαδή η δημιουργία μιας πληθώρας διαφορετικών δομών προγραμμάτων για περαιτέρω εξέλιξη. Μια παράμετρος στην εκτέλεση του ΓΠ είναι το μέγιστο μέγεθος κάθε ατόμου. Στην περίπτωση όπου χρησιμοποιείται δενδρική δομή, η παράμετρος αυτή εκφράζεται ως το μέγιστο βάθος κάθε κόμβου ή το μέγιστο πλήθος των κόμβων. Αφού οριστούν τα σύνολα των τερματικών και των συναρτήσεων, υπάρχουν δύο βασικές μέθοδοι για την αρχικοποίηση.

Στην πρώτη περίπτωση κάθε κόμβος επιλέγεται τυχαία, από την ένωση των συνόλων των τερματικών και των συναρτήσεων. Εξαιρέση αποτελεί η ρίζα, η οποία επιλέγεται τυχαία μόνο από το σύνολο των συναρτήσεων. Όταν επιλεγεί ένα τερματικό, το αντίστοιχο κλαδί του δένδρου τερματίζει, ανεξάρτητα από το μέγιστο μέγεθος του δένδρου. Αυτή η μέθοδος ονομάζεται *επαύξηση*. Εξαιτίας της τυχαιότητας του είδους κάθε κόμβου τα δένδρα που προκύπτουν συνήθως δεν έχουν κάποια κανονική μορφή.

Αντί για το τυχαίο είδος κάθε κόμβου, στην *πλήρη* μέθοδο επιλέγονται τυχαία στοιχεία από το σύνολο των συναρτήσεων, μέχρι το δένδρο να φτάσει το μέγιστο βάθος. Τότε επιλέγονται τυχαία μόνο τερματικά. Αποτέλεσμα αυτής της μεθόδου είναι ένας πληθυσμός ο οποίος αποτελείται από πλήρη δένδρα, ίδιου (του μέγιστου) μεγέθους.

10.2.5 Διασταύρωση

Όπως είδαμε στις προηγούμενες ενότητες, ο ΓΠ χρησιμοποιεί τους γενετικούς τελεστές για να μετασχηματίσει τις πιθανές λύσεις στο πρόβλημα. Από τη σκοπιά της μηχανικής μάθησης αυτοί είναι οι τελεστές αναζήτησης. Ο πιο σημαντικός ανάμεσά τους είναι η διασταύρωση. Ο συγκεκριμένος τελεστής συνδυάζει το γενετικό υλικό δύο γονέων, αντιμεταθέτοντας ένα μέρος του πρώτου με ένα μέρος του δεύτερου. Η γραφική αναπαράσταση της λειτουργίας του τελεστή παρουσιάζεται στο παρακάτω σχήμα 10.4. Με έντονη γραμμή φαίνονται τα υποδένδρα κάθε δένδρου που αποκόπτονται και αντιμετατίθενται από το ένα δένδρου (γονέα) στο άλλο (γονέα) για να δημιουργήσουν τους δύο απογόνους.



Σχήμα 10.4 Ο τελεστής της διασταύρωσης.

Αναλυτικά τα βήματα της εφαρμογής του τελεστή είναι:

- Επιλογή δύο ατόμων, σύμφωνα με τον τελεστή επιλογής γονέων, π.χ. ανάλογα με την απόδοσή τους. Αυτός είναι ο γνωστός τελεστής επιλογής της εξαναγκασμένης ρουλέτας (roulette wheel selection), που παρουσιάστηκε στο Κεφάλαιο 2.
- Επιλογή τυχαίου υποδένδρου σε κάθε γονέα. Τα υποδένδρα που επιλέχθηκαν στο παράδειγμα του σχήματος είναι τονισμένα. Εδώ μπορεί να γίνει μια παραλλαγή και τα υποδένδρα που περιέχουν μόνο τερματικούς κόμβους να είναι λιγότερο πιθανό να επιλεγούν.

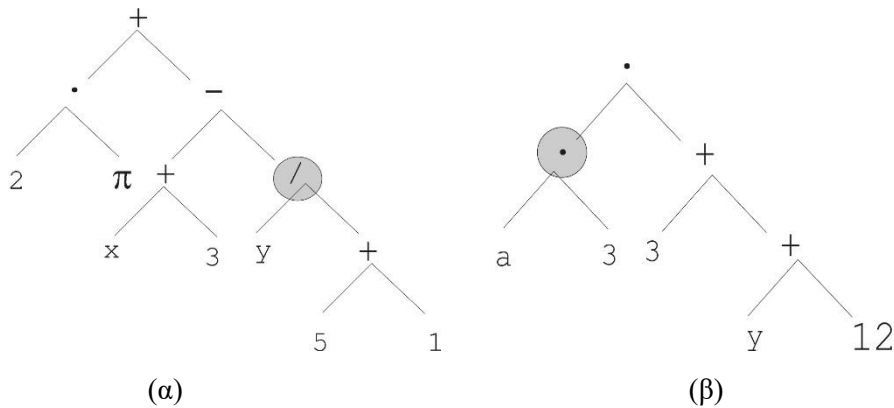
- Ανταλλαγή των επιλεγμένων υποδένδρων μεταξύ των δύο γονέων. Τα άτομα που προκύπτουν είναι οι απόγονοι.

Άσκηση 10.4

Να δώσετε σε μορφή ψευδοκώδικα τον αλγόριθμο του τελεστή διασταύρωσης.

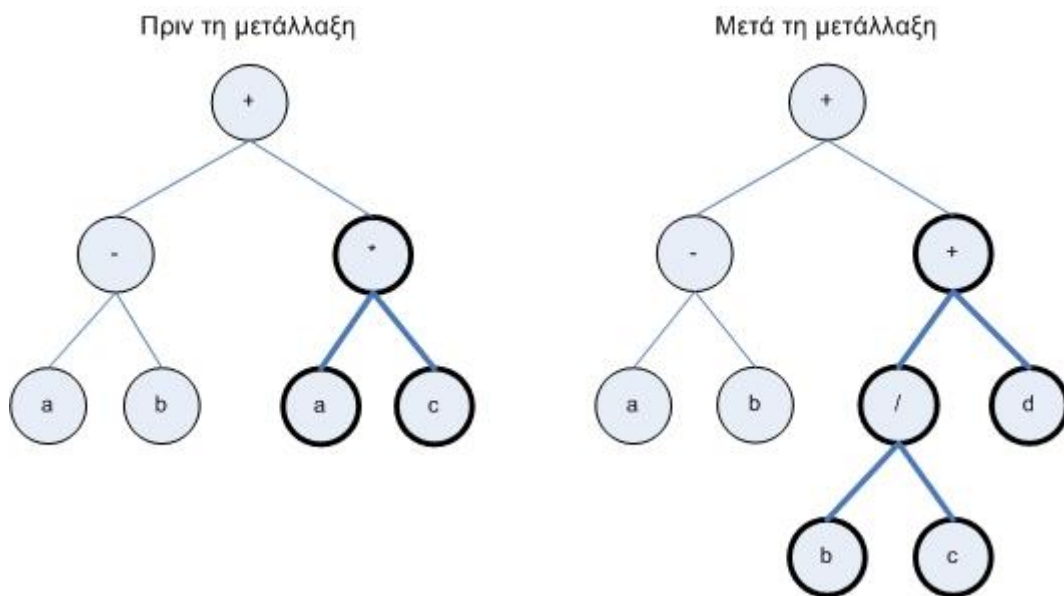
Άσκηση 10.5

Δίνονται δύο άτομα (τα α και β που φαίνονται στο παρακάτω σχήμα), τα οποία έχουν επιλεγεί από έναν πληθυσμό για διασταύρωση. Ποιοι θα είναι οι απόγονοι, αν τα σημεία διασταύρωσης είναι οι κόμβοι που είναι σκιασμένοι;



10.2.6 Μετάλλαξη

Η μετάλλαξη ενεργεί σε ένα μόνο άτομο. Συνήθως, μετά την ολοκλήρωση της διασταύρωσης, κάθε απόγονος που προκύπτει υφίσταται κατά μικρή πιθανότητα μετάλλαξη· η συγκεκριμένη πιθανότητα είναι μια παράμετρος της εκτέλεσης του αλγορίθμου. Όταν ένα άτομο επιλεγεί για μετάλλαξη, επιλέγεται ένα τυχαίο υποδένδρο του, όπως και στη διασταύρωση. Αυτό το υποδένδρο αντικαθίσταται από ένα τυχαίο υποδένδρο, το οποίο παράγεται με την ίδια διαδικασία, όπως και κατά την αρχικοποίηση του πληθυσμού. Ένα παράδειγμα μετάλλαξης παρουσιάζεται στο επόμενο σχήμα.



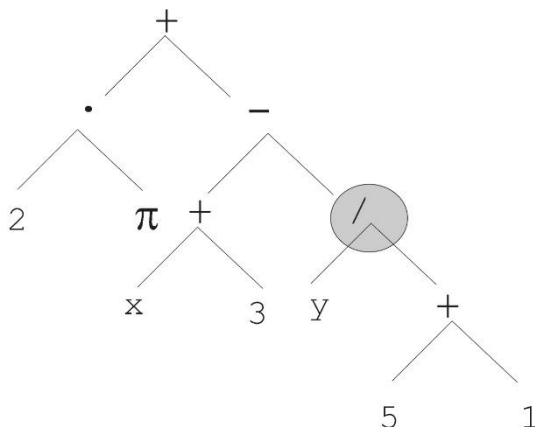
Σχήμα 10.5 Ο τελεστής της μετάλλαξης.

Άσκηση 10.6

Να δώσετε σε μορφή ψευδοκώδικα τον αλγόριθμο του τελεστή μετάλλαξης.

Άσκηση 10.7

Δίνεται ένα άτομο (που φαίνεται στο παρακάτω σχήμα), το οποίο έχει επιλεγεί από έναν πληθυσμό προκειμένου να υποστεί μετάλλαξη. Ποιος θα είναι μεταλλαγμένος απόγονος, αν το σημείο μετάλλαξης είναι ο κόμβος που είναι σκιασμένος;



10.2.7 Αναπαραγωγή

Ο τελεστής της αναπαραγωγής είναι πολύ απλός. Ένα άτομο επιλέγεται πιθανοτικά με βάση την καταλληλότητά του και αντιγράφεται. Το αντίγραφο του εισάγεται στον πληθυσμό· δηλαδή στο τέλος υπάρχουν δύο ίδια άτομα.

10.2.8 Αντικειμενική συνάρτηση

Όπως αναφέρθηκε νωρίτερα, ένα πολύ σημαντικό σημείο του αλγορίθμου είναι η αξιολόγηση των πιθανών λύσεων. Η αντικειμενική συνάρτηση είναι το μέτρο που χρησιμοποιεί ο ΓΠ κατά τη διάρκεια της εξέλιξης, το κατά πόσο ένα δεδομένο πρόγραμμα έχει μάθει να προβλέπει τις εξόδους από τις εισόδους· με άλλα λόγια τα χαρακτηριστικά του πεδίου μάθησης. Αυτή η συνάρτηση είναι σημαντική, γιατί δίνει ανάδραση αναφορικά με το ποια άτομα πρέπει να αναπαραχθούν και ποια άτομα πρέπει να εκλείψουν από τον πληθυσμό. Η συγκεκριμένη συνάρτηση υπολογίζεται με βάση το σύνολο εκπαίδευσης και πρέπει να δίνει βαθμωτή και συνεχή ανάδραση. Μία απλή και συνεχής αντικειμενική συνάρτηση είναι το άθροισμα των απόλυτων τιμών του λάθους. Τυπικά αν η πραγματική έξοδος για το i παράδειγμα είναι o_i και η έξοδος του p ατόμου είναι p_i , η απόδοση του f_p είναι:

$$f_p = \sum_{i=1}^n |p_i - o_i| \quad (10.1)$$

Σε πολλές περιπτώσεις αν χρησιμοποιηθεί το τετραγωνικό λάθος έχει καλύτερα αποτελέσματα:

$$f_p = \sum_{i=1}^n (p_i - o_i)^2 \quad (10.2)$$

Αυτό οφείλεται στο γεγονός ότι έτσι εξασφαλίζεται καλύτερη κλιμάκωση των τιμών της καταλληλότητας των ατόμων του πληθυσμού, διευκολύνοντας με αυτόν τον τρόπο την επιλογή.

10.2.9 Επιλογή

Αφού η ποιότητα μιας πιθανής λύσης έχει καθοριστεί, πρέπει να αποφασιστεί εάν θα εφαρμοστούν οι γενετικοί τελεστές σε αυτό το άτομο. Επίσης από την απόδοσή του εξαρτάται η διατήρησή του στον πληθυσμό ή η αντικατάστασή του από κάποια άλλη λύση. Γι' αυτήν τη διαδικασία είναι υπεύθυνος ο τελεστής της επιλογής. Υπάρχουν αρκετοί τέτοιοι τελεστές και η απόφαση για το ποιος θα χρησιμοποιηθεί επηρεάζει σημαντικά την ταχύτητα της εξέλιξης. Τα σημαντικότερα είδη τελεστών επιλογής είναι δύο:

Επιλογή ανάλογα με την απόδοση:

Αυτή η μέθοδος χρησιμοποιείται στους γενετικούς αλγορίθμους και είναι η πιο συνηθισμένη. Σύμφωνα με αυτήν, η πιθανότητα p_i να αφήσει απογόνους το άτομο i στην επόμενη γενιά είναι:

$$p_i = f_i / \sum_{j=1}^n f_j \quad (10.3)$$

Επιλογή με διαγωνισμό:

Αυτού του είδους η επιλογή βασίζεται στον διαγωνισμό με συγκεκριμένο αριθμό ατόμων και όχι με όλον τον πληθυσμό, όπως παραπάνω. Οι νικητές του διαγωνισμού αναπαράγονται χρησιμοποιώντας τους προηγούμενους τελεστές και οι απόγονοί τους αντικαθιστούν τους ηττημένους, στον πληθυσμό. Η συγκεκριμένη μορφή της επιλογής κερδίζει όλο και περισσότερο έδαφος, γιατί δεν απαιτεί κεντρικό υπολογισμό της απόδοσης κάθε ατόμου και επιταχύνει την εξέλιξη. Επίσης, είναι πολύ χρήσιμη στην παράλληλη επεξεργασία.

10.2.10 Ο βασικός αλγόριθμος

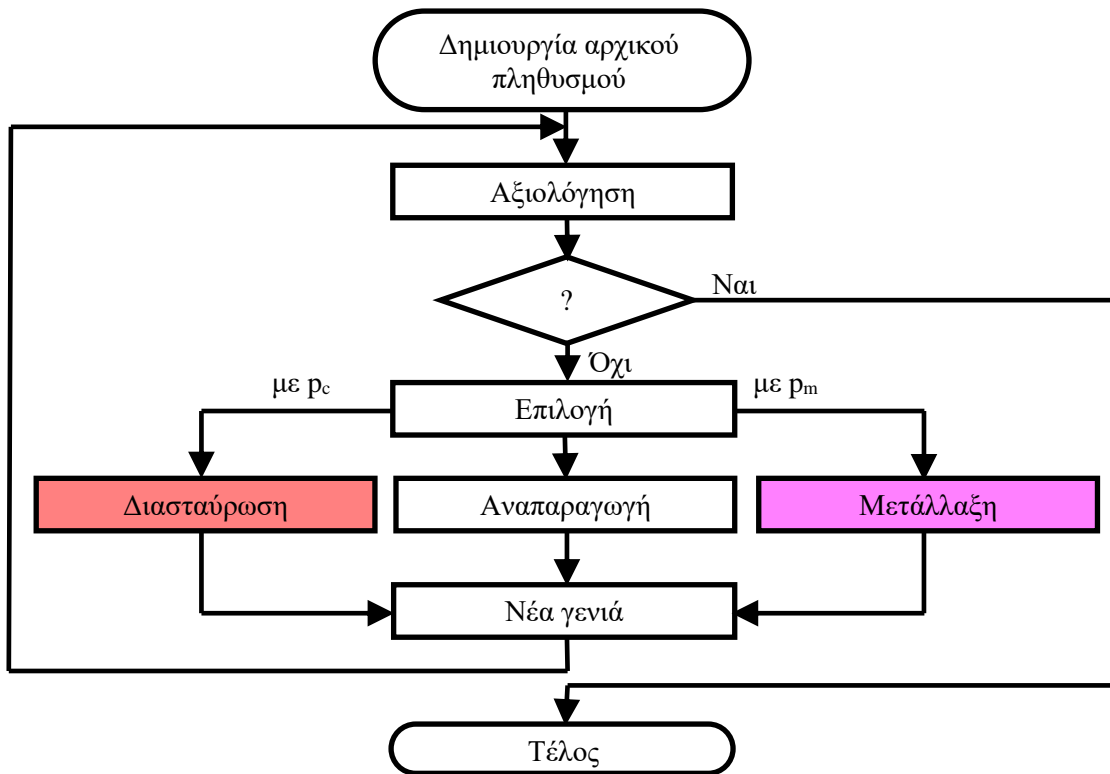
Συνδυάζοντας όσα παρουσιάστηκαν στις προηγούμενες ενότητες, καταλήγουμε σε δύο μορφές του βασικού αλγορίθμου του ΓΠ.

1. Ο αλγόριθμος που χρησιμοποιεί γενιές.

Παραδοσιακά ο ΓΠ χρησιμοποιεί διακριτές γενιές για να επιτύχει την εξέλιξη. Κάθε γενιά αποτελείται από τον ίδιο αριθμό ατόμων. Η επόμενη γενιά δημιουργείται μετασχηματίζοντας τα άτομα της προηγούμενης. Αναλυτικά τα βήματα του αλγορίθμου είναι:

1. Αρχικοποίηση του πληθυσμού.
2. Αξιολόγηση του πληθυσμού: δηλαδή ανάθεση σε κάθε άτομο μιας αριθμητικής απόδοσης.
3. Μέχρι η νέα γενιά να ολοκληρωθεί, επανάλαβε:
 - επιλογή ενός ατόμου σύμφωνα με τον αντίστοιχο τελεστή,
 - εφαρμογή των γενετικών τελεστών στο επιλεγμένο άτομο,
 - εισαγωγή του αποτελέσματος στη νέα γενιά.
4. Αν δεν πληρούνται το κριτήριο τερματισμού, αντικατάστησε τον πληθυσμό της προηγούμενης γενιάς με αυτόν της νέας και επιστροφή στο βήμα 2.
5. Παρουσίασε το καλύτερο άτομο της γενιάς ως το αποτέλεσμα του αλγορίθμου.

Η βασική φιλοσοφία του παραπάνω αλγορίθμου περιγράφεται σχηματικά από το παρακάτω διάγραμμα ροής:



Σχήμα 10.6 Διάγραμμα ροής ΕΠ.

Το σύμβολο ? στο παραπάνω διάγραμμα ροής υπονοεί τον έλεγχο της συνθήκης τερματισμού. Πρέπει να σημειώσουμε ότι το διάγραμμα ροής του σχήματος 10.6 δεν περιλαμβάνει τον τελεστή μετατροπής αρχιτεκτονικής του προγράμματος. Ο τελεστής αυτός, που πραγματοποιεί λειτουργίες μετατροπής της αρχιτεκτονικής, θα παρουσιαστεί σε επόμενη παράγραφο.

2. Αλγόριθμος σταθερής κατάστασης

Σε αυτήν την περίπτωση χρησιμοποιείται η επιλογή με διαγωνισμό, οπότε δεν υπάρχουν διακριτές γενιές. Τα βήματα αυτής της παραλλαγής είναι:

1. Αρχικοποίηση πληθυσμού.
2. Τυχαία επέλεξε ένα υποσύνολο του πληθυσμού για να λάβει μέρος στο διαγωνισμό.
3. Αξιολόγηση κάθε διαγωνιζομένου.
4. Επέλεξε τους κερδισμένους ανάλογα με την απόδοσή τους.
5. Εφάρμοσε τους γενετικούς τελεστές στους νικητές.
6. Αντικατάστησε τους ηττημένους με τους απογόνους των νικητών.
7. Εάν δεν πληρούται το κριτήριο τερματισμού, επιστροφή στο 2.
8. Παρουσίασε το καλύτερο άτομο του πληθυσμού ως το αποτέλεσμα του αλγορίθμου.

Άσκηση 10.8

Η συμβολική παλινδρόμηση είναι η διαδικασία εύρεσης τόσο της μορφής, όσο και των συντελεστών μιας συνάρτησης-στόχου, ή τουλάχιστον μια προσέγγιση αυτών. Αυτό διαφέρει από άλλους τύπους παλινδρόμησης, όπως η πολυωνυμική παλινδρόμηση όπου προσδιορίζονται οι συντελεστές ενός πολυωνύμου προκαθορισμένου βαθμού. Όλα τα προβλήματα ΓΠ είναι κατά κάποια έννοια προβλήματα συμβολικής παλινδρόμησης, αλλά ο όρος συνήθως χρησιμοποιείται για την εύρεση συναρτήσεων με πραγματικές τιμές, όπως σε αυτήν την άσκηση. Θεωρούμε τα παρακάτω ζεύγη τιμών για τις μεταβλητές x και $y=f(x)$:

X	y=f(x)
-1.00	1.00
-0.80	0.84
-0.60	0.76
-0.40	0.76
-0.20	0.84
0.00	1.00
0.20	1.24
0.40	1.56
0.60	1.96
0.80	2.44
1.00	3.00

Να βρεθεί ένα πρόγραμμα υπολογιστή με τη βοήθεια ΓΠ, το οποίο δέχεται ως είσοδο την τιμή της ανεξάρτητης μεταβλητής x και δίνει ως έξοδο την αντίστοιχη τιμή της μεταβλητής y , στο διάστημα $[-1, +1]$.

10.3 Λειτουργίες Μεταβολής της Αρχιτεκτονικής του Προγράμματος

Σκοπός

Σκοπός αυτής της ενότητας είναι να παρουσιάσει μια ξεχωριστή λειτουργία του ΓΠ, τη λειτουργία μεταβολής της αρχιτεκτονικής ενός προγράμματος, η οποία δεν χρησιμοποιείται στους άλλους ΕΑ. Η συγκεκριμένη λειτουργία χρησιμοποιείται από τον αλγόριθμο ΓΠ —που μιμείται αντίστοιχες λειτουργίες της φύσης— για να καθορίσει αυτόματα την αρχιτεκτονική ενός προγράμματος. Αυτό επιτυγχάνεται δυναμικά, κατά τη διάρκεια εκτέλεσης του αλγορίθμου, με την αντιγραφή ενός ορίσματος ή μιας υπορουτίνας, τη δημιουργία νέας υπορουτίνας, τη διαγραφή ορίσματος ή υπορουτίνας και άλλες λειτουργίες μεταβολής, όπως αύξηση ή ελάττωση του αριθμού επαναλήψεων, προσθήκη ή απάλειψη βρόχων ή αναδρομών και τέλος συνδυασμούς των παραπάνω.

Προσδοκώμενα αποτελέσματα

Όταν θα έχετε ολοκληρώσει τη μελέτη αυτής της ενότητας, θα μπορείτε να:

- αναφέρετε τουλάχιστον πέντε λειτουργίες μετατροπής,
- κάνετε συνδυασμό των παραπάνω βασικών λειτουργιών, δημιουργώντας νέες,
- σχεδιάσετε έναν ολοκληρωμένο αλγόριθμο ΓΠ.

Τα απλά προγράμματα ηλεκτρονικών υπολογιστών αποτελούνται από ένα κυρίως πρόγραμμα (που ονομάζεται κλάδος παραγωγής του αποτελέσματος). Βέβαια πιο πολύπλοκα προγράμματα περιέχουν υπορουτίνες (που επίσης καλούνται αυτόματα οριζόμενες συναρτήσεις ή κλάδοι ορισμού συναρτήσεων), επαναλήψεις (αυτόματα οριζόμενες επαναλήψεις), βρόχους (αυτόματα οριζόμενους βρόχους), αναδρομές (αυτόματα οριζόμενες αναδρομές) και μνήμη ποικίλης διάστασης και μεγέθους (αυτόματα οριζόμενες μνήμες). Εάν ένας χρήστης προσπαθεί να λύσει ένα πρόβλημα μηχανικής, ίσως επιλέξει απλώς να προκαθορίσει μια λογική σταθερή αρχιτεκτονική διάταξη για όλα τα προγράμματα του πληθυσμού (π.χ. ο αριθμός και οι τύποι των κλάδων και ο αριθμός των ορισμάτων τον οποίο κάθε κλάδος κατέχει). Ο ΓΠ μπορεί τότε να χρησιμοποιηθεί για να εξελίξει την ακριβή ακολουθία από πρωτόγονα βήματα που επιτελούν την επιθυμητή εργασία σε κάθε κλάδο.

Ωστόσο, μερικές φορές το μέγεθος και η μορφή της λύσης είναι το πρόβλημα (ή τουλάχιστον το κύριο μέρος αυτού). Ο ΓΠ είναι ικανός να πάρει όλες τις αρχιτεκτονικές αποφάσεις δυναμικά, ενόσω τρέχει.

Χρησιμοποιεί λειτουργίες τροποποίησης αρχιτεκτονικής για να καθορίσει αυτόματα την αρχιτεκτονική του προγράμματος κατά έναν τρόπο ο οποίος παραλληλίζει την αντιγραφή των γονιδίων και τη σχετική διαδικασία διαγραφής γονιδίων στη φύση. Οι λειτουργίες μεταβολής της αρχιτεκτονικής παρέχουν έναν τρόπο, δυναμικό όπως επώθηκε, για την προσθήκη και τη διαγραφή υπορουτινών και άλλων τύπων κλάδων σε ατομικά προγράμματα, να προσθέτουν και να αφαιρούν ορίσματα που κατέχονται από τις υπορουτίνες και άλλους τύπους κλάδων. Αυτή η λειτουργία τροποποίησης αρχιτεκτονικής δημιουργεί γρήγορα έναν πληθυσμό από διαφορετικά άτομα που περιέχει προγράμματα με διαφορετικούς αριθμούς υπορουτινών, ορισμάτων, επαναλήψεων, βρόχων, αναδρομών και μνήμης και επίσης διαφορετικές ιεραρχικές διατάξεις αυτών. Προγράμματα με αρχιτεκτονική που ταιριάζει καλά στο συγκεκριμένο πρόβλημα θα τείνουν να αυξηθούν και ευδοκούν στην ανταγωνιστική εξελικτική διαδικασία, ενώ προγράμματα με ακατάλληλες αρχιτεκτονικές θα εξασθενούν κάτω από την αμείλικτη πίεση επιλογής από τη συνάρτηση καταλληλότητας του προβλήματος. Έτσι οι λειτουργίες τροποποίησης της αρχιτεκτονικής απαλλάσσουν τον χρήστη από το επίπονο έργο του προκαθορισμού της αρχιτεκτονικής του προγράμματος.

Υπάρχουν μερικές διαφορετικές λειτουργίες τροποποίησης της αρχιτεκτονικής προγραμμάτων, που περιγράφονται στη συνέχεια. Καθεμία από αυτές εφαρμόζεται με φειδώ κατά τη διάρκεια του τρεξίματος (έστω με πιθανότητα 0.5% ή 1% σε κάθε γενιά).

- **Η λειτουργία αντιγραφής υπορουτίνας**, η οποία αντιγράφει μια προϋπάρχουσα υπορουτίνα σε ένα ατομικό πρόγραμμα, δίνει ένα νέο όνομα στο αντίγραφο και τυχαία διαιρεί τις προϋπάρχουσες κλήσεις στην παλιά υπορουτίνα μεταξύ των δύο (την παλιά και το αντίγραφο). Αυτή η λειτουργία αλλάζει την αρχιτεκτονική του προγράμματος διευρύνοντας την ιεραρχία των υπορουτινών σε ολόκληρο το πρόγραμμα. Όπως και στην αντιγραφή των γονιδίων στη φύση, η παραπάνω λειτουργία διατηρεί τη σημασιολογία, όταν συμβαίνει για πρώτη φορά. Οι δύο υπορουτίνες τυπικά αποκλίνουν αργότερα, μερικές φορές αποδίδοντας εξειδίκευση.
- **Η λειτουργία αντιγραφής ορίσματος** αντιγράφει ένα όρισμα μιας υπορουτίνας, τυχαία διαιρεί τις εσωτερικές αναφορές σε αυτό και διατηρεί τη σημασιολογία σε ολόκληρο το πρόγραμμα, προσαρμόζοντας όλες τις κλήσεις στην υπορουτίνα. Αυτή η λειτουργία αυξάνει τη διάσταση του υποχώρου στον οποίο λειτουργεί η υπορουτίνα.
- **Η λειτουργία δημιουργίας υπορουτίνας** μπορεί να δημιουργήσει μια νέα υπορουτίνα από ένα τμήμα ενός κλάδου που παράγει αποτέλεσμα με αυτόν τον τρόπο, εμβαθύνοντας την ιεραρχία των αναφορών σε ολόκληρο το πρόγραμμα. Δημιουργεί λοιπόν μια ιεραρχική αναφορά μεταξύ μιας προϋπάρχουσας αναφοράς ανάμεσα στο κυρίως πρόγραμμα και τη νέα υπορουτίνα. Η λειτουργία δημιουργίας υπορουτίνας μπορεί επίσης να δημιουργήσει μια νέα υπορουτίνα από ένα τμήμα μίας υπάρχουσας, εμβαθύνοντας παραπάνω την ιεραρχία των αναφορών και δημιουργώντας με αυτόν τον τρόπο μια ιεραρχική αναφορά μεταξύ μιας προϋπάρχουσας υπορουτίνας και μιας νέας σε μια πιο βαθιά και πιο σύνθετη συνολική ιεραρχία.
- **Η λειτουργία διαγραφής υπορουτίνας** διαγράφει μια υπορουτίνα από ένα πρόγραμμα, κάνοντας με αυτόν τον τρόπο την ιεραρχία των υπορουτινών είτε στενότερη είτε πιο ρηχή.
- **Η λειτουργία διαγραφής ορίσματος** διαγράφει ένα όρισμα από μια υπορουτίνα, ελαττώνοντας με αυτόν τον τρόπο το διαθέσιμο ποσό πληροφορίας στην υπορουτίνα· πρόκειται για μια διαδικασία που μπορεί να θεωρηθεί ως γενίκευση.
- **Άλλες λειτουργίες μεταβολής αρχιτεκτονικής** προσθέτουν και αφαιρούν αυτόματα καθορισμένες επαναλήψεις, αυτόματα καθορισμένους βρόχους, αυτόματα καθορισμένες αναδρομές και αυτόματα καθορισμένες μνήμες.

Υπάρχουν τρεις τρόποι με τους οποίους ο ΓΠ μπορεί να προσεγγίσει την αρχιτεκτονική τού προς εξέλιξη προγράμματος υπολογιστή:

- Ο χρήστης μπορεί να προκαθορίσει την αρχιτεκτονική ολόκληρου του προγράμματος (π.χ. να κάνει ένα επιπλέον προπαρασκευαστικό βήμα ορισμού της αρχιτεκτονικής). Έτσι ο αριθμός των προπαρασκευαστικών βημάτων αυξάνεται σε έξι.
- Ο χρήστης μπορεί να χρησιμοποιήσει εξελικτική επιλογή της αρχιτεκτονικής, όπως περιγράφεται στο Koza, 1994, καθιστώντας με αυτό τον τρόπο ικανή την αρχιτεκτονική ολόκληρου του προγράμματος να αναδυθεί από μια ανταγωνιστική διαδικασία κατά τη διάρκεια τρεξίματος του προγράμματος. Όταν χρησιμοποιείται αυτή η προσέγγιση, ο αριθμός των προπαρασκευαστικών βημάτων παραμένει στα πέντε που ορίστηκαν αρχικά.
- Το τρέξιμο μπορεί να χρησιμοποιήσει τις λειτουργίες μεταβολής αρχιτεκτονικής (Koza et al., 1999), καθιστώντας έτσι ικανό τον ΓΠ να δημιουργήσει αυτόματα την αρχιτεκτονική όλου του προγράμματος δυναμικά κατά τη διάρκεια του τρεξίματος. Όταν χρησιμοποιείται αυτή η προσέγγιση, ο αριθμός των προπαρασκευαστικών βημάτων παραμένει στα πέντε που ορίστηκαν αρχικά.

Άσκηση 10.9

Να ορίσετε τα τέσσερα δομικά στοιχεία της αρχιτεκτονικής ενός προγράμματος.

10.4 Βασικά ζητήματα της υλοποίησης του ΓΠ

10.4.1 Ανασκόπηση του διαγράμματος ροής του ΓΠ

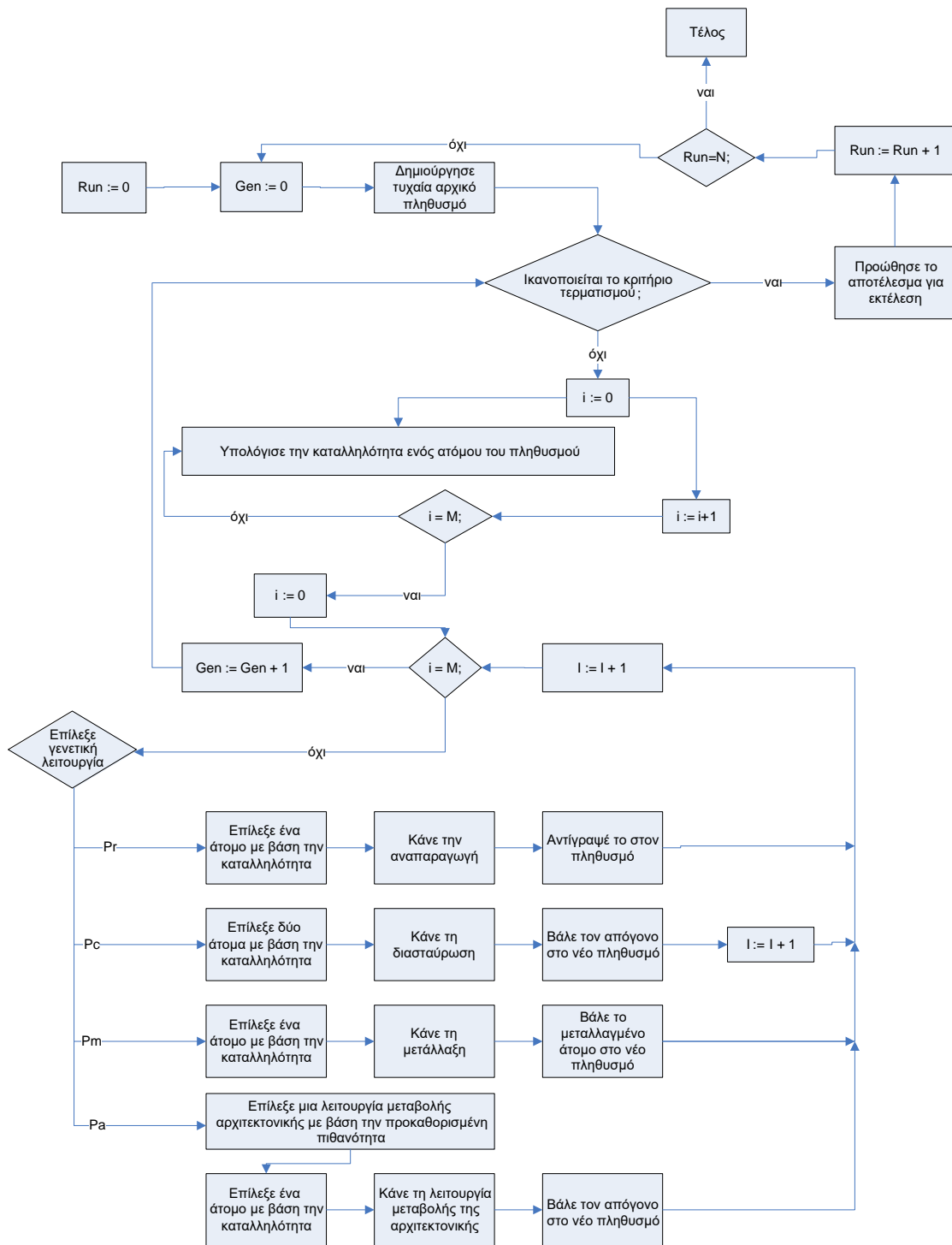
Στην προηγούμενη ενότητα ολοκληρώθηκε η παρουσίαση των γενετικών τελεστών που χρησιμοποιούνται στον ΓΠ, με τις λειτουργίες μετατροπής της αρχιτεκτονικής του προγράμματος. Μπορούμε τώρα να παρουσιάσουμε το γενικευμένο διάγραμμα ροής και να κάνουμε μια σύντομη ανασκόπηση.

Ο ΓΠ είναι ανεξάρτητος από το πρόβλημα, με την έννοια ότι το διάγραμμα ροής που προσδιορίζει την ακολουθία των εκτελεστικών βημάτων δεν τροποποιείται για κάθε νέα εκτέλεση ή κάθε νέο πρόβλημα.

Συνήθως δεν υπάρχει καμία προαιρετική ανθρώπινη μεσολάβηση ή αλληλεπίδραση κατά τη διάρκεια τρεξίματος του ΓΠ (παρόλο που ο χρήστης μπορεί να κρίνει πότε θα σταματήσει η εκτέλεση).

Το διάγραμμα ροής στο παρακάτω σχήμα δείχνει τα εκτελεστικά βήματα ενός τρεξίματος του ΓΠ. Δείχνει τόσο τους γενετικούς τελεστές της αναπαραγωγής, της διασταύρωσης και της μετάλλαξης, όσο και τις λειτουργίες μετατροπής της αρχιτεκτονικής του προγράμματος. Επίσης, δείχνει μια έκδοση του τελεστή διασταύρωσης που δίνει δύο απογόνους.

Πριν προχωρήσουμε στην ανάλυση του διαγράμματος, θα ορίσουμε πρώτα ορισμένες μεταβλητές που χρησιμοποιούνται σε αυτό. Η Run είναι η μεταβλητή στην οποία καταχωρείται ο αριθμός των εκτελέσεων του αλγορίθμου. Η Gen είναι η μεταβλητή καταχώρησης της τρέχουσας γενιάς, ενώ η σταθερά M είναι ο μέγιστος αριθμός γενεών. Τέλος, οι σταθερές Pr, Pc, Pm και Pa είναι αντίστοιχα οι πιθανότητες αναπαραγωγής, διασταύρωσης, μετάλλαξης και μετατροπής προγράμματος.



Σχήμα 10.7 Το πλήρες διάγραμμα ροής του ΓΠ.

Όπως φαίνεται από το παραπάνω σχήμα, ο ΓΠ ξεκινά με έναν αρχικό πληθυσμό από προγράμματα υπολογιστή, που συντίθενται από τερματικά και συναρτήσεις κατάλληλα για το πρόβλημα. Τα άτομα-προγράμματα στον αρχικό πληθυσμό δημιουργούνται τυπικά παράγοντας αναδρομικά ένα πρόγραμμα ρίζα δένδρου με σημειωτικές, από τυχαία επιλογή πρωτογενών συναρτήσεων και τερματικών (που παρέχονται από τον χρήστη και αποτελούν το πρώτο και το δεύτερο μέρος των προπαρασκευαστικών βημάτων τρεξίματος του ΓΠ). Τα αρχικά άτομα συνήθως παράγονται με βάση το προκαθορισμένο μέγεθος (καθορίζεται από τον χρήστη ως δευτερεύουσα παράμετρος στο τέταρτο προπαρασκευαστικό βήμα). Γενικά τα προγράμματα στον πληθυσμό έχουν διαφορετικό μέγεθος (όσον αφορά τον αριθμό συναρτήσεων και τερματικών) και διαφορετικό σχήμα (την ειδική γραφική τοποθέτηση των συναρτήσεων και τερματικών στο δένδρο του προγράμματος).

Κάθε άτομο-πρόγραμμα στον πληθυσμό εκτελείται. Κατόπιν είτε αξιολογείται είτε συγκρίνεται για το πόσο καλά εκτελεί το ζητούμενο έργο (χρησιμοποιώντας το μέτρο καταλληλότητας που παρέχεται από το 3^ο προκαταρκτικό βήμα). Για πολλά προβλήματα αυτό το μέτρο αποδίδει μια απλή αριθμητική τιμή που καλείται καταλληλότητα. Η καταλληλότητα ενός προγράμματος μπορεί να μετρηθεί με διάφορους τρόπους, περιλαμβάνοντας π.χ. το μέτρο του λάθους μεταξύ της εξόδου του και της επιθυμητής εξόδου, το ποσό χρόνου (καυσίμου, χρημάτων κλπ.) που απαιτείται για να φθάσει το σύστημα σε μια επιθυμητή κατάσταση-στόχο, την ακρίβεια του προγράμματος στην αναγνώριση προτύπων ή στην ταξινόμηση αντικειμένων σε κλάσεις, τον λογαριασμό (κέρδος) τον οποίο παράγει ένα πρόγραμμα που παίζει ένα παίγνιο ή τη συμμόρφωση μιας σύνθετης κατασκευής (όπως μια κεραία, ένα κύκλωμα ή ένας ελεγκτής) με τα σχεδιαστικά κριτήρια που βάζει ο χρήστης. Η εκτέλεση του προγράμματος μερικές φορές επιστρέφει μία ή περισσότερες ακριβείς τιμές. Διαφορετικά η εκτέλεση ενός προγράμματος μπορεί να αποτελείται μόνο από παρενέργειες στην κατάσταση του κόσμου (π.χ. η κίνηση ενός ρομπότ). Η εκτέλεση ενός προγράμματος λοιπόν μπορεί να παράγει τόσο τιμές όσο και παρενέργειες.

Το μέτρο καταλληλότητας είναι σε πολλά πρακτικά προβλήματα πολυμεταβλητό (multiobjective), με την έννοια ότι συνδυάζει δύο ή περισσότερα διαφορετικά στοιχεία. Τα διαφορετικά στοιχεία του μέτρου καταλληλότητας βρίσκονται συνήθως σε κάποιον βαθμό σε ανταγωνισμό το ένα με το άλλο.

Για πολλά προβλήματα κάθε πρόγραμμα στον πληθυσμό εκτελείται πάνω σε ένα δείγμα διαφορετικών περιπτώσεων καταλληλότητας. Αυτές οι περιπτώσεις μπορεί να αναπαριστούν διαφορετικές τιμές της (των) εισόδου (εισόδων), διαφορετικές αρχικές συνθήκες ενός συστήματος ή διαφορετικά περιβάλλοντα. Μερικές φορές οι περιπτώσεις καταλληλότητας κατασκευάζονται πιθανοτικά.

Η κατασκευή του τυχαίου αρχικού πληθυσμού είναι στην πραγματικότητα μια τυφλή τυχαία αναζήτηση του χώρου αναζήτησης του προβλήματος. Παρέχει μια γραμμή στοίχισης (baseline) για να κρίνουμε μελλοντικές προσπάθειες αναζήτησης. Τυπικά τα ατομικά προγράμματα στη γενιά 0 έχουν όλα υπερβολικά χαμηλή απόδοση. Παρ' όλα αυτά, μερικά άτομα στον πληθυσμό είναι (συνήθως) πιο αποδοτικά από άλλα. Οι διαφορές στην καταλληλότητα είναι εκμεταλλεύσιμες από τον ΓΠ. Ο ΓΠ εφαρμόζει φυσική επιλογή και γενετικές λειτουργίες για να κατασκευάσει μια νέα γενιά από προγράμματα απογόνους από τον τρέχοντα πληθυσμό.

10.4.2 Παραγωγή δένδρων συντακτικής ανάλυσης προγράμματος

Ο ΓΠ ξεκινά με έναν αρχικό πληθυσμό από προγράμματα υπολογιστή, που συντίθεται από συναρτήσεις και τερματικά κατάλληλα για το πρόβλημα. Τα ατομικά προγράμματα στον αρχικό πληθυσμό τυπικά παράγονται αναδρομικά από ένα ριζωμένο με σημεία-ετικέτες πρόγραμμα δένδρου, που συντίθεται από τυχαίες επιλογές των αρχικών συναρτήσεων και τερματικών (που παρέχονται από τον χρήστη στο πρώτο και δεύτερο προπαρασκευαστικό βήμα). Τα αρχικά άτομα συνήθως παράγονται υποκείμενα σε προκαθορισμένο μέγιστο μέγεθος (που καθορίζεται από τον χρήστη ως δευτερεύουσα παράμετρος, μέρος του τέταρτου προπαρασκευαστικού βήματος). Π.χ. στην «Πλήρη» (“Full”) μέθοδο αρχικοποίησης λαμβάνονται κόμβοι από το σύνολο συναρτήσεων μέχρι να φθάσει σε ένα μέγιστο βάθος δένδρου. Πέραν αυτού του βάθους μόνο τερματικά μπορούν να επιλεγούν. Μια παραλλαγή αυτής της μεθόδου, η μέθοδος «Ανάπτυξης» (“Grow”) αρχικοποίησης, επιτρέπει την επιλογή κόμβων από όλο το σύνολο (συναρτήσεων και τερματικών) μέχρι να φθάσει το όριο βάθους. Μετά από αυτό συμπεριφέρεται όπως η Πλήρης μέθοδος.

Στον επόμενο πίνακα δίνεται ο ψευδοκώδικας για μια επαναληπτική υλοποίηση των δύο μεθόδων αρχικοποίησης. Ο κώδικας υποθέτει ότι τα προγράμματα αναπαρίστανται ως εκφράσεις με σημείωση προθέματος.

```

Procedure: gen_rnd_expr
arguments:
  func_set      /* Σύνολο συναρτήσεων */
  term_set      /* Σύνολο τερματικών */
  max_d         /* Μέγιστο βάθος εκφράσεων */
  method        /* Είτε "Full" είτε "Grow" */
results:
  expr          /* Έκφραση με σημείωση προθέματος */
if max_d = 0 or method = "Grow" and random digit = 1 then
  expr = choose_random_element (term_set )
else
  func = choose_random_elelent ( func_set )
  for i = 1 to arity ( func );
    arg_i = gen_rnd.expr ( func_set, term_set, max_d – 1, method );
    expt ( func, arg_1, arg_2, ... );
  endif
end

```

Πίνακας 10.1 Ψευδοκώδικας για επαναληπτική παραγωγή προγράμματος με τις μεθόδους "Full" και "Grow" (Kozs, 1994).

Γενικά μετά τη φάση αρχικοποίησης τα προγράμματα στον πληθυσμό είναι διαφορετικού μεγέθους (σε αριθμό τερματικών και συναρτήσεων) και διαφορετικής δομής (η γραφική διευθέτηση των τερματικών και των συναρτήσεων στο δένδρο του προγράμματος).

Κάθε ατομικό πρόγραμμα στον πληθυσμό είτε αποτιμάται είτε συγκρίνεται για τον βαθμό που εκτελεί το συγκεκριμένο έργο (χρησιμοποιώντας το μέτρο καταλληλότητας που παρέχεται από το τρίτο προπαρασκευαστικό βήμα). Για πολλά προβλήματα αυτή η μέτρηση παίρνει μια απλή αριθμητική τιμή, την καταλληλότητα. Κανονικά η αξιολόγηση της καταλληλότητας απαιτεί εκτέλεση των προγραμμάτων του πληθυσμού, πολλές φορές μέσα στο σύστημα ΓΠ. Υπάρχει μια ποικιλία στρατηγικών εκτέλεσης, που περιλαμβάνει (τη σχετικά ασυνήθιστη) off-line ή on-line μετάφραση (compilation) και διασύνδεση και (τη σχετικά συνηθισμένη) μετάφραση κώδικα-ιδεατής-μηχανής (virtual-machine-code) και διερμηνεία.

Η διερμηνεία ενός δένδρου προγράμματος σημαίνει την εκτέλεση των κόμβων στο δένδρο, με σειρά που εγγυάται ότι οι κόμβοι δεν εκτελούνται πριν πάρουν τιμή τα ορίσματά της (αν υπάρχουν). Αυτό συνήθως γίνεται διασχίζοντας το δένδρο με επαναληπτικό τρόπο, αρχίζοντας από τον κόμβο-ρίζα και μεταθέτοντας την αξιολόγηση κάθε κόμβου μέχρι να πάρει τιμή το παιδί (όρισμά) του. Ο πίνακας 10.2 δίνει την υλοποίηση σε ψευδοκώδικα της διαδικασίας διερμηνείας. Ο κώδικας υποθέτει ότι τα προγράμματα αναπαρίστανται ως εκφράσεις με σημείωση προθέματος και ότι τέτοιες εκφράσεις μπορούν να χρησιμοποιηθούν ως λίστες από συνιστώσες (όπου μια κατασκευή σαν την `expr(i)` μπορεί να χρησιμοποιηθεί για να διαβάσει ή να δώσει τιμή στη συνιστώσα i της έκφρασης `expr`).

```

Procedure: eval
arguments:
  expr          /* Μια έκφραση με συμβολισμό-προθέματος */
results:
  value        /* Ένας αριθμός */
begin
  if expr is a list then /* Δεν υπάρχει τερματικό */
    proc = expr (1)
    value = proc (eval (2)), eval (3)), ...)
  else /* Τερματικό */
    if expr is a variable or a constant then
      value = expr
    else /* συνάρτηση με μηδενικό πληθάρημο */
      value = expr ( )
    endif
  endif
endif

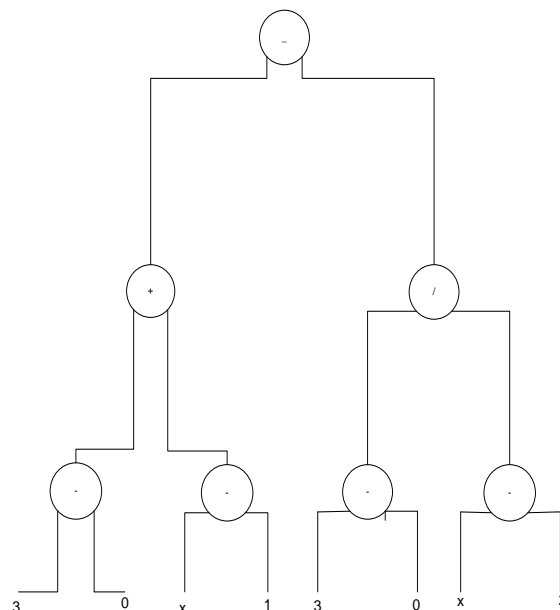
```

Πίνακας 10.2 Τυπικός διερμηνευτής για Γενετικό Προγραμματισμό (Κοζα, 1994).

Άσχετα από τη στρατηγική εκτέλεσης που θα υιοθετηθεί, η καταλληλότητα του προγράμματος μπορεί να αποτιμηθεί με πολλούς διαφορετικούς τρόπους, περιλαμβάνοντας π.χ. το μέγεθος του λάθους μεταξύ της εξόδου του και της επιθυμητής εξόδου, το ποσό του χρόνου (καυσίμου, χρημάτων κλπ.) που απαιτείται για να φέρει ένα σύστημα σε μια επιθυμητή κατάσταση-στόχο, την ακρίβεια του προγράμματος στην αναγνώριση προτύπων ή ταξινόμηση αντικειμένων. Η εκτέλεση του προγράμματος μερικές φορές επιστρέφει μία ή περισσότερες ακριβείς τιμές. Εναλλακτικά, η εκτέλεση ενός προγράμματος μπορεί να είναι μόνο μία παρενέργεια (side effect) στην κατάσταση του κόσμου (π.χ. η κίνηση ενός ρομπότ). Τέλος, η εκτέλεση ενός προγράμματος μπορεί να έχει για αποτέλεσμα και τα δύο, δηλαδή να επιστρέφει τιμές και να έχει παρενέργειες.

Άσκηση 10.10

Στο παρακάτω δένδρο συντακτικής ανάλυσης η ανεξάρτητη μεταβλητή X παίρνει την τιμή -1 . Να κάνετε την εκτέλεση των συναρτήσεων των κόμβων στο δένδρο με τη σειρά που εγγυάται ότι οι κόμβοι δεν εκτελούνται, αν δεν πάρουν τιμές τα ορίσματά τους. Να δείξετε τη φορά της εκτέλεσης (δηλαδή τη διαπέραση των κόμβων) πάνω στο σχήμα και να βάλετε στα δεξιά των κόμβων την τιμή της αποτίμησης των υποδένδρων που ριζώνουν σε αυτά τα σημεία.



10.5 Εφαρμογές του ΓΠ

Από το 1992 που παρουσιάστηκε ο ΓΠ το ενδιαφέρον γι' αυτόν ανεβαίνει με μεγάλο ρυθμό. Η γενικότητα της ιδέας και η έλλειψη συγκεκριμένων απαιτήσεων για την εφαρμογή του είναι οι βασικοί λόγοι που συνέβαλαν σε αυτό. Στον σύνδεσμο http://en.wikipedia.org/wiki/List_of_genetic_algorithm_applications υπάρχει μια εκτενής λίστα από εφαρμογές, σε αρκετούς τομείς της επιστήμης και της μηχανικής. Ενδεικτικά αναφέρονται τα εξής θεματικά πεδία:

- Βιοτεχνολογία
- Γραφικά
- Έλεγχος
- Εξόρυξη δεδομένων
- Ηλεκτρική μηχανική
- Οικονομικά
- Επεξεργασία εικόνας
- Μοντελοποίηση
- Φυσικές γλώσσες
- Βελτιστοποίηση
- Αναγνώριση προτύπων
- Επεξεργασία σημάτων

Για περισσότερες πληροφορίες σχετικά με αυτές ο αναγνώστης παραπέμπεται στην παραπάνω αναφορά και στους παρακάτω συνδέσμους:

<http://www.genetic-programming.com/>

<http://brainz.org/15-real-world-applications-genetic-algorithms/>

<http://sci2s.ugr.es/publications/ficheros/2010-Espejo-IEEETSMCC.pdf>

10.5.1 Μελέτη περίπτωσης: Ευφυής στρατηγική για ένα τεχνητό μυρμήγκι (artificial ant)

Το παράδειγμα στο οποίο θα αναφερθούμε αφορά την ανάπτυξη μιας «ευφούς» στρατηγικής, η οποία μπορεί να χρησιμοποιηθεί από ένα πολύ απλό τεχνητό μυρμήγκι. Το μυρμήγκι κινείται σε ένα πλέγμα 32x32· αν βγει από τη μια μεριά του πλέγματος, θα μπει από την αντίθετη (το πλέγμα δηλαδή δεν έχει τοιχώματα). Το πλέγμα περιέχει ένα μονοπάτι, το οποίο δεν είναι μια απλή ευθεία γραμμή, με τροφή πάνω σε αυτό. Το μονοπάτι με την τροφή θεωρούμε ότι ξεκινά στο σημείο [0, 0] του πλέγματος, σημείο από το οποίο ξεκινά και το μυρμήγκι. Κάθε θέση στο πλέγμα μπορεί να είναι άδεια, να περιέχει φαγητό ή να περιέχει μια ένδειξη ότι είναι κενό μεταξύ θέσεων με φαγητό (σε αυτήν την περίπτωση θα λέμε ότι περιέχει κενά). Το μονοπάτι πρέπει να είναι συνεχόμενο και να μην περιέχει άδειες θέσεις. Όλο το φαγητό θα πρέπει να βρίσκεται σε θέσεις που γειτνιάζουν με θέσεις που περιέχουν φαγητό ή θέσεις που περιέχουν ένδειξη ότι είναι κενό μεταξύ θέσεων με φαγητό.

Ένα απλό τεχνητό μυρμήγκι έχει δύο στοιχεία τα οποία περιγράφουν πλήρως την κατάστασή του: το πρώτο είναι η θέση του στο πλέγμα και το δεύτερο είναι η κατεύθυνση προς την οποία «κοιτά». Το απλό τεχνητό μυρμήγκι μπορεί επιπλέον να κάνει τρία διαφορετικά πράγματα: να στρίψει αριστερά, να στρίψει δεξιά ή να κινηθεί προς το τετράγωνο το οποίο «κοιτά». Το μυρμήγκι έχει μόνο έναν αισθητήρα ο οποίος το βοηθά να κινηθεί στο περιβάλλον του: μια συνάρτηση (την οποία θα την αποκαλούμε FacingFood) η οποία του λέει εάν υπάρχει φαγητό στο τετράγωνο το οποίο κοιτά ή όχι.

Το πρόβλημα είναι να δημιουργηθεί ένα απλό τεχνητό μυρμήγκι το οποίο θα μπορεί να κάνει μια σειρά από ευφυείς κινήσεις και στροφές, βασισμένο μόνο στη συνάρτηση FacingFood. Ένα παράδειγμα τέτοιου απλού τεχνητού μυρμηγκιού απεικονίζεται στον πίνακα 10.3.


```

if (FacingFood()) {
    Move();
}
else {
    TurnRight();
    Move();
}

```

Πίνακας 10.3 Ένα απλό τεχνητό μυρμήγκι.

Το μυρμήγκι του πίνακα 10.3 θα τα πήγαινε μια χαρά σε δεξιόστροφα μονοπάτια, αλλά όχι και σε μονοπάτια που περιέχουν και αριστερές στροφές. Το πρακτικό πρόβλημα λοιπόν είναι να χρησιμοποιήσουμε Γενετικό Προγραμματισμό για να δημιουργήσουμε ένα πρόγραμμα (το απλό τεχνητό μυρμήγκι) το οποίο θα διαπερνά ένα μονοπάτι φαγητού και κενών κατά τυχαία μορφή με όσο το δυνατόν πιο ευφυή τρόπο.

Υλοποίηση με Γενετικό Προγραμματισμό

Στο κλασικό βιβλίο του ο John Koza παρουσιάζει έναν πολύ μεγάλο αριθμό από προβλήματα που μπορούν να επιλυθούν με τη χρήση Γενετικού Προγραμματισμού. Ανέπτυξε έναν συστηματικό τρόπο με τον οποίο κάποιος μπορεί να συσχετίσει συγκεκριμένα προβλήματα με σύνολα σχεδιαστικών αποφάσεων που μπορεί να πάρει προκειμένου να μοντελοποιήσει ένα πρόβλημα.

Ως γνωστόν το πρώτο βήμα για τη μοντελοποίηση ενός προβλήματος με τη χρήση Γενετικού Προγραμματισμού είναι η επιλογή των τερματικών (terminals) και των συναρτήσεων (functions). Το τεχνητό μυρμήγκι θα ελέγχεται από εκφράσεις συμπεριφοράς, οι οποίες είναι δενδρικές δομές από τερματικά και συναρτήσεις. Όπως ήδη έχουμε πει, τα τερματικά είναι λειτουργίες ή literals τα οποία δεν μπορούν να αποδομηθούν περισσότερο και οι συναρτήσεις είναι υψηλότερου επιπέδου λειτουργίες οι οποίες χρησιμοποιούν άλλες συναρτήσεις ή τερματικά. Στην περίπτωση μας το τεχνητό μυρμήγκι έχει τα τερματικά: Left (αριστερά), Right (δεξιά) και Move (κινήσου). Ένα μέλος του συνόλου των συναρτήσεων είναι η συνάρτηση FacingFood, η οποία παίρνει ως ορίσματα δύο τερματικά ή συναρτήσεις και επιστρέφει μια Boolean τιμή. Η έκφραση “if FacingFood, then Left else Right” είναι ένα παράδειγμα μιας έκφρασης που χρησιμοποιεί τη συνάρτηση FacingFood και τα δύο τερματικά Left και Right.

Ο Γενετικός Αλγόριθμος ξεκινά δημιουργώντας έναν πληθυσμό από τυχαίες εκφράσεις. Αυτές στη συνέχεια εκτελούνται και αξιολογούνται, χρησιμοποιώντας ένα μέτρο απόδοσης. Η επιλογή ενός μέτρου απόδοσης είναι ένα από τα πλέον σημαντικά βήματα στην εφαρμογή του Γενετικού Προγραμματισμού. Επιθυμούμε να δημιουργούμε ικανά μυρμήγκια, τέτοια ώστε να συλλέγουν πολύ φαγητό σε όσο το δυνατόν λιγότερα βήματα. Έτσι το μέτρο απόδοσης θα πρέπει να λαμβάνει υπόψη τόσο το πλήθος του φαγητού που συλλέχθηκε, όσο και το πλήθος των βημάτων που χρειάστηκε να γίνουν. Ένα μέτρο απόδοσης λοιπόν είναι να διαιρούμε το πλήθος των βημάτων με το πλήθος του φαγητού που συγκεντρώθηκε και στη συνέχεια να προσθέτουμε έναν σταθερό αριθμό (π.χ. 1000) ώστε να εξασφαλίζουμε ότι η απόδοση θα είναι πάντα θετική. Επομένως ένα μυρμήγκι με υψηλότερη απόδοση θα είναι πάντα πιο αποτελεσματικό από ένα μυρμήγκι με χαμηλότερη.

Στη συνέχεια ακολουθούν οι κλασικές γενετικές διαδικασίες (γενετικοί τελεστές) που περιγράψαμε στις προηγούμενες ενότητες: επιλογή, διασταύρωση και μετάλλαξη. Κατά τη διαδικασία της επιλογής μόνο το 10% των καλύτερων μελών του πληθυσμού επιλέγεται για να περάσει στον επόμενο, προσωρινό πληθυσμό και να αποτελέσει τη «δεξαμενή» για την επιλογή των γονέων στη διαδικασία της διασταύρωσης. Κατά τη συγκεκριμένη διαδικασία επιλέγονται ζευγάρια από τα μέλη (το 10% που έχει «επιβιώσει») σύμφωνα με την απόδοσή τους (έτσι τα πιο ικανά μέλη έχουν μεγαλύτερη πιθανότητα να διασταυρωθούν). Τα ζευγάρια αυτά διασταυρώνονται με τον «κλασικό τρόπο», δηλαδή επιλέγεται ένα τυχαίο σημείο διασταύρωσης σε καθέναν από τους δύο γονείς και ανταλλάσσονται τα υποδένδρα που σχηματίζονται, παράγοντας με αυτόν τον τρόπο δύο απογόνους οι οποίοι μπαίνουν στον νέο πληθυσμό. Η διαδικασία της διασταύρωσης συνεχίζεται έως ότου ο πληθυσμός συμπληρωθεί με τους απογόνους από το 10% των μελών που είχαν επιβιώσει κατά τη διαδικασία της επιλογής. Στη συνέχεια ακολουθεί η διαδικασία της μετάλλαξης, όπου επιλέγεται με πιθανότητα 2% ένα τερματικό και αντικαθίσταται από ένα άλλο τυχαία επιλεγμένο.

Ο προτεινόμενος αλγόριθμος Γενετικού Προγραμματισμού, παρότι είναι αρκετά απλός, φαίνεται να αποδίδει πολύ καλά στην πράξη και να καταφέρνει να παραγάγει απλά τεχνητά μυρμήγκια με «ευφυή» στρατηγική μετακίνησης σε πλέγμα. Περισσότερες λεπτομέρειες για την υλοποίηση του αλγορίθμου που παρουσιάσαμε παραπάνω σε περιβάλλον Microsoft Visual Studio.Net και σε γλώσσα προγραμματισμού C#, καθώς και τον κώδικα υλοποίησης μπορείτε να βρείτε στη διεύθυνση:
<http://msdn.microsoft.com/msdnmag/issues/04/08/GeneticAlgorithms/>

Βιβλιογραφία

- Koza, J.R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press.
- Koza, J.R. (1994). *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press.
- Koza, J.R., Bennett III, F.H., Andre, D., & Keane, M.A. (1999). *Genetic Programming III: Darwinian Invention and Problem Solving*, Morgan Kaufmann.

Απαντήσεις ασκήσεων

Άσκηση 10.1

Έστω ότι θέλετε να σχεδιάσετε αυτόματα ένα ηλεκτρικό κύκλωμα με τη βοήθεια ΓΠ. Να ορίσετε το σύνολο των τερματικών και των συναρτήσεων που καθιστούν ικανό τον ΓΠ να σχεδιάσει αυτόματα το πρόγραμμα. Να αναφέρετε πέντε άλλες εφαρμογές στις οποίες μπορεί να χρησιμοποιηθεί το σύνολο τερματικών και συναρτήσεων που ορίσατε.

Απάντηση:

Για την αντιμετώπιση του προβλήματος με χρήση ΓΠ θα πρέπει πρώτα να οριστούν όλες οι συναρτήσεις που θα χρησιμοποιηθούν, καθώς και τα τερματικά σύμβολα (μεταβλητές, σταθερές και συναρτήσεις μηδενικού ορίσματος). Στη συνέχεια ο αλγόριθμος ΓΠ θα εκτελέσει μια εξελικτική αναζήτηση στον χώρο των προγραμμάτων που μπορούν να κατασκευαστούν με τη χρήση των συναρτήσεων και των τερματικών που ορίστηκαν.

Ο ΓΠ θα κατασκευάσει το κύκλωμα από στοιχεία όπως τρανζίστορς, διόδους, πυκνωτές και αντιστάσεις. Αυτή η διαδικασία κατασκευής (ανάπτυξης) τυπικά αρχίζει με μια εμβρυϊκή δομή, όπως ένα απλό μετατρέψιμο καλώδιο. Αν επιπλέον ένα τέτοιο σύνολο συναρτήσεων είναι τοπολογικά ενημερωμένο, η τοποθέτηση των στοιχείων και η δρομολόγηση μπορούν να συντεθούν παράλληλα με την τοπολογία και τη διαστασιολόγηση του κυκλώματος.

Όταν ο χρήστης έχει καθορίσει τα στοιχειώδη συστατικά για το πρόβλημα της σύνθεσης κυκλώματος, το ίδιο σύνολο τερματικών και συναρτήσεων μπορεί να χρησιμοποιηθεί για την αυτόματη σύνθεση ενός ενισχυτή, ενός υπολογιστικού κυκλώματος, ενός ενεργού φίλτρου, ενός κυκλώματος αναφοράς τάσης και οποιουδήποτε κυκλώματος που μπορεί να συντεθεί από αυτά τα στοιχεία.

Άσκηση 10.2

Να αναπαραστήσετε σε μορφή δένδρου τις παρακάτω εκφράσεις:

α. $3(x+6)$.

β. $(x^2+1)/(x^2-1)$

γ. $(A=2) \text{ AND } (S>100)$

δ. `If (FacingFood(() {`

`Move () ;`

`}`

`else {`

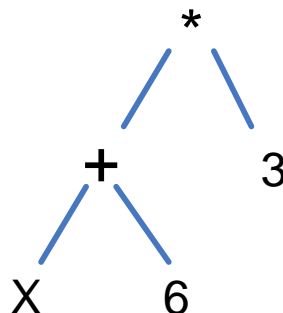
`Right () ;`

`Move () ;`

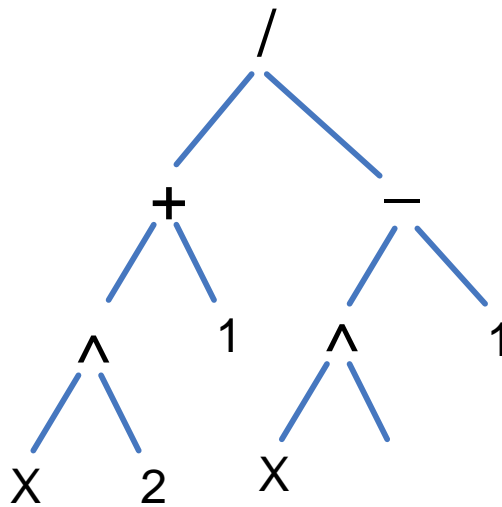
`}`

Απάντηση

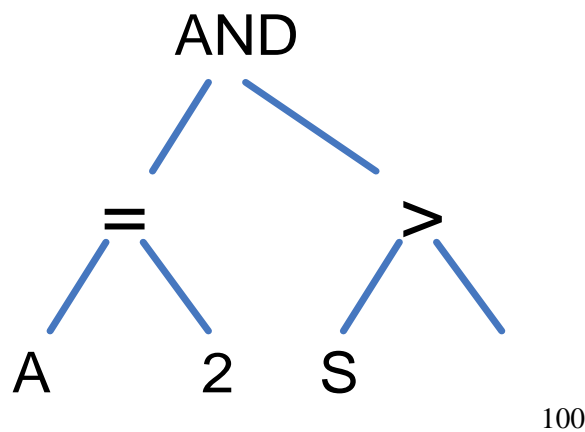
α. Το σύνολο τερματικών είναι $[x, 6, 3]$ και το σύνολο συναρτήσεων $[+, *]$. Άρα έχουμε το παρακάτω δένδρο:



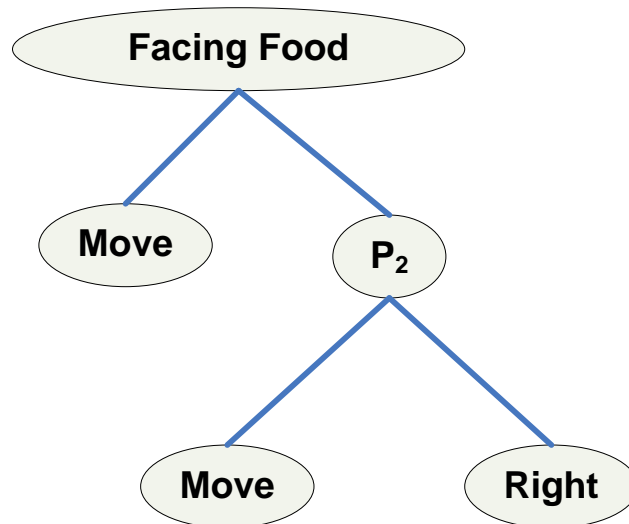
β. Το σύνολο τερματικών είναι $[x, 2, 1]$ και το σύνολο συναρτήσεων $[+, -, /, ^]$. Άρα έχουμε το παρακάτω δένδρο:



γ. Το σύνολο τερματικών είναι $[A, 2, S, 100]$ και το σύνολο συναρτήσεων $[=, >, \text{AND}]$. Άρα έχουμε το παρακάτω δένδρο:



δ. Ο κώδικας περιέχει μία συνάρτηση FacingFood και δύο τερματικά Move και Right. Αυτά παράγουν άλλη μία συνάρτηση, την P2, η οποία έχει δύο προτάσεις (clauses), που μπορεί να είναι είτε τερματικά είτε συναρτήσεις. Εδώ έχουμε τα τερματικά “Right, Move”. Άρα η αναπαράσταση με δένδρο συντακτικής ανάλυσης είναι η παρακάτω:



Άσκηση 10.3

Να αναφέρετε δέκα τρόπους αναπαράστασης, εκτός από τα δένδρα συντακτικής ανάλυσης, που αναφέρθηκαν στην ενότητα 10.1.

Απάντηση:

Δέκα τρόποι αναπαράστασης είναι οι παρακάτω:

1. IF-THEN κανόνες παραγωγής
2. Horn Clauses
3. Bayesian Δίκτυα
4. Νευρωνικά Δίκτυα
5. Πλαίσια
6. Προτασιακή Λογική
7. Δυαδικά Διαγράμματα Απόφασης
8. Τυπικές Γραμματικές
9. Συντελεστές για Πολυώνυμα
10. Συστήματα Ταξινόμησης.

Αν έχετε βρει τουλάχιστον πέντε, μπράβο σας! Έχετε κατανοήσει βασικές έννοιες της TN. Αν όχι, μην απογοητεύεστε, γιατί είναι νέες έννοιες και δεν έχετε ακόμη αποκτήσει άνεση με αυτές.

Άσκηση 10.4

Να δώσετε σε μορφή ψευδοκώδικα τον αλγόριθμο του τελεστή διασταύρωσης.

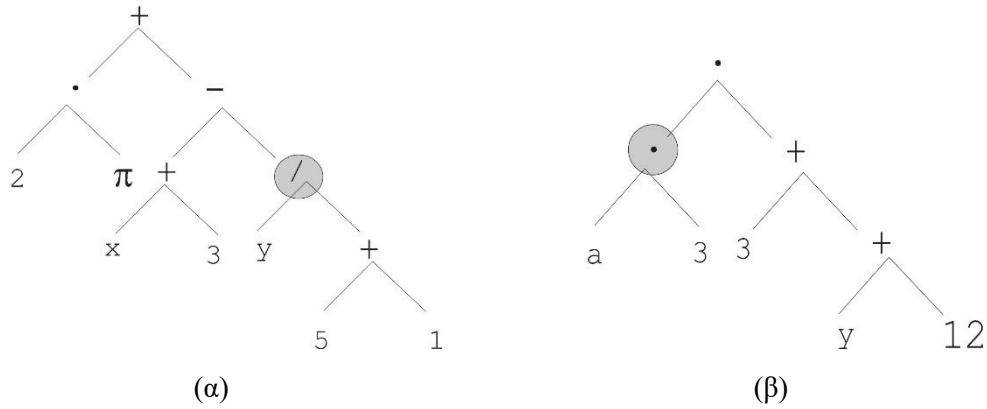
Απάντηση:

Πρέπει πρώτα να ορίσουμε τα πιθανά σημεία διασταύρωσης, έστω Number-of-Points.

1. Επίλεξε 2 γονείς (πιθανοτικά με βάση την καταλληλότητα – βλέπε ενότητα 3.1).
2. Επίλεξε τυχαία έναν αριθμό από 1 μέχρι Number-of-Points.
3. Επίλεξε ανεξάρτητα έναν τυχαίο αριθμό για τον 2ο γονέα.
4. Προσδιόρισε τα δύο υποδένδρα τα οποία φωλιάζουν στα επιλεγέντα σημεία.
5. Αντιμετάθεσε τα δύο υποδένδρα.
6. Έλεγε αν οι απόγονοι είναι συντακτικά ορθά προγράμματα.
7. Βάλε τους δύο απογόνους στην επόμενη γενιά του πληθυσμού.

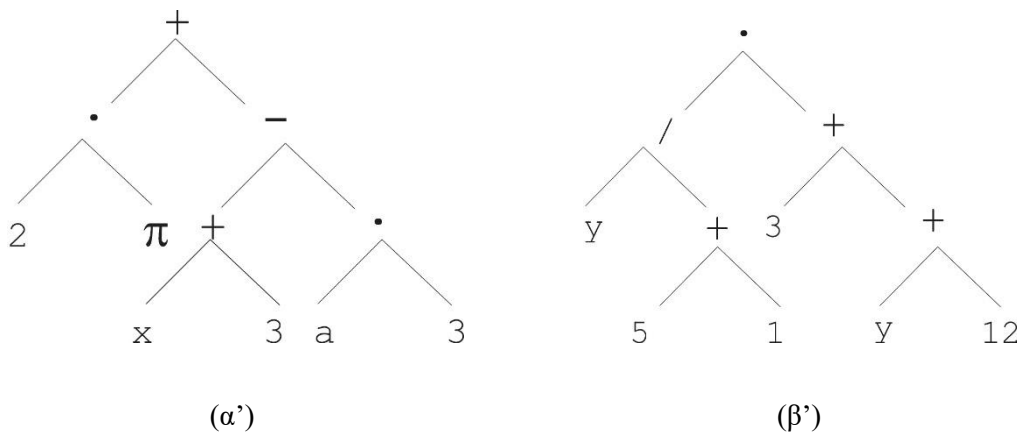
Άσκηση 10.5

Δίνονται δύο άτομα (τα α και β που φαίνονται στο παρακάτω σχήμα), τα οποία έχουν επιλεγεί από έναν πληθυσμό για διασταύρωση. Ποιοι θα είναι οι απόγονοι, αν τα σημεία διασταύρωσης είναι οι κόμβοι που είναι σκιασμένοι;



Απάντηση:

Οι δύο απόγονοι, σύμφωνα με τον τελεστή διασταύρωσης, θα προκύψουν αν αντιμεταθέσουμε τα δύο υποδένδρα, τα οποία έχουν για ρίζα τους σκιασμένους κόμβους. Άρα θα είναι:



Άσκηση 10.6

Να δώσετε σε μορφή ψευδοκώδικα τον αλγόριθμο του τελεστή μετάλλαξης.

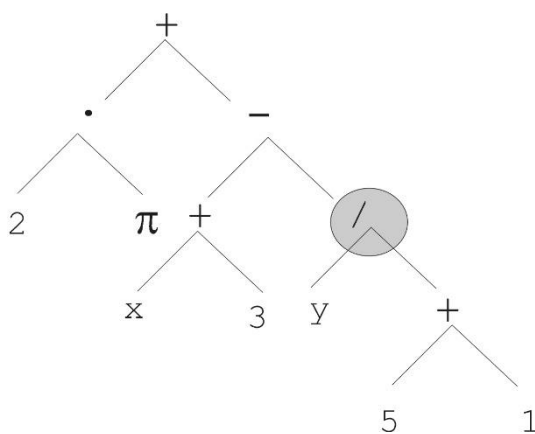
Απάντηση:

Πρέπει πρώτα να ορίσουμε τα πιθανά σημεία μετάλλαξης, έστω Number-of-Points.

1. Επίλεξε 1 γονέα (πιθανοτικά με βάση την καταλληλότητα – βλέπε ενότητα 3.1).
2. Επίλεξε τυχαία έναν αριθμό από 1 μέχρι Number-of-Points.
3. Προσδιόρισε το υποδένδρο το οποίο φωλιάζει στο επιλεγμένο σημείο.
4. Διάγραψε το υποδένδρο του βήματος 3.
5. Δημιούργησε νέο υποδένδρο στο σημείο μετάλλαξης (με τον ίδιο τρόπο που παράγονται δένδρα στον αρχικό πληθυσμό, τη γενιά 0).
6. Έλεγξε αν το αποτέλεσμα είναι συντακτικά ορθό πρόγραμμα.
7. Βάλε τον απόγονο στην επόμενη γενιά του πληθυσμού.

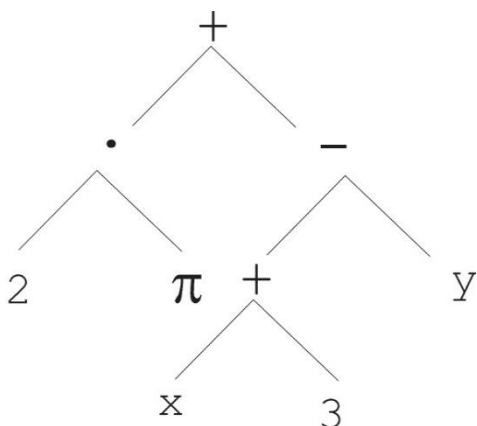
Άσκηση 10.7

Δίνεται ένα άτομο (που φαίνεται στο παρακάτω σχήμα), το οποίο έχει επιλεγεί από έναν πληθυσμό προκειμένου να υποστεί μετάλλαξη. Ποιος θα είναι μεταλλαγμένος απόγονος, αν το σημείο μετάλλαξης είναι ο κόμβος που είναι σκιασμένος;



Απάντηση:

Η πιο συνηθισμένη μορφή μετάλλαξης είναι η αντικατάσταση ενός τυχαία επιλεγμένου υποδένδρου από ένα τυχαία παραγόμενο υποδένδρο, το οποίο παράγεται επιλέγοντας τερματικά και συναρτήσεις από τα διαθέσιμα σύνολα. Το επιλεγμένο υποδένδρο μπορεί να αντικατασταθεί απλά από ένα τερματικό y (όπως φαίνεται στο παρακάτω σχήμα). Φυσικά μπορεί να αντικατασταθεί από οποιοδήποτε άλλο υποδένδρο, που παράγεται από το διαθέσιμο σύνολο τερματικών και συναρτήσεων.



Άσκηση 10.8

Η συμβολική παλινδρόμηση είναι η διαδικασία εύρεσης τόσο της μορφής, όσο και των συντελεστών μιας συνάρτησης-στόχου ή τουλάχιστον μια προσέγγιση αυτών. Αυτό διαφέρει από άλλους τύπους παλινδρόμησης, όπως η πολυωνυμική παλινδρόμηση όπου προσδιορίζονται οι συντελεστές ενός πολυωνύμου προκαθορισμένου βαθμού. Όλα τα προβλήματα ΓΠ είναι κατά κάποια έννοια προβλήματα συμβολικής παλινδρόμησης, αλλά ο όρος συνήθως χρησιμοποιείται για την εύρεση συναρτήσεων με πραγματικές τιμές, όπως σε αυτήν την άσκηση. Θεωρούμε τα παρακάτω ζεύγη τιμών για τις μεταβλητές x και $y=f(x)$:

X	y=f(x)
-1.00	1.00
-0.80	0.84

-0.60	0.76
-0.40	0.76
-0.20	0.84
0.00	1.00
0.20	1.24
0.40	1.56
0.60	1.96
0.80	2.44
1.00	3.00

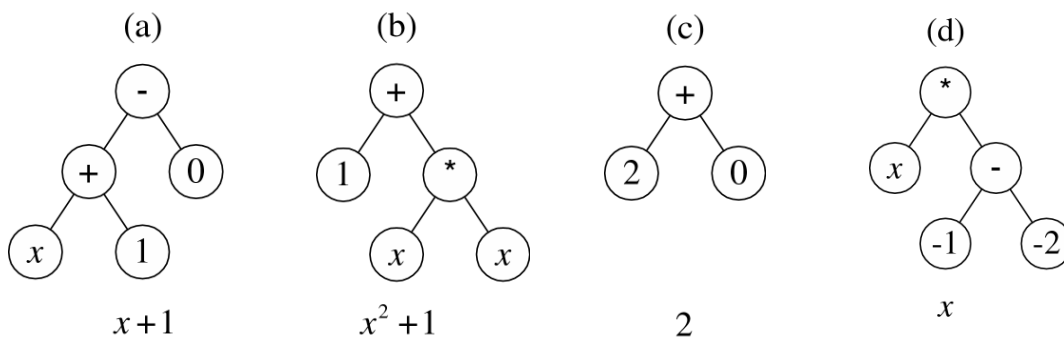
Να βρεθεί ένα πρόγραμμα υπολογιστή με τη βοήθεια ΓΠ, το οποίο δέχεται ως είσοδο την τιμή της ανεξάρτητης μεταβλητής x και δίνει ως έξοδο την αντίστοιχη τιμή της μεταβλητής y , στο διάστημα $[-1, +1]$.

Απάντηση:

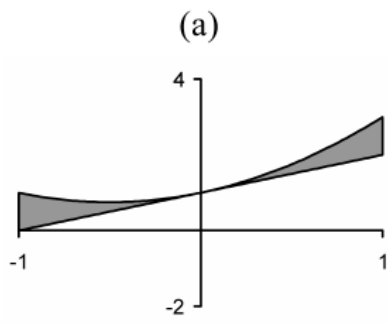
Πρέπει πρώτα να καθορίσουμε τα πέντε προπαρασκευαστικά βήματα, τα οποία φαίνονται στον παρακάτω πίνακα:

1	Σύνολο τερματικών:	$T = \{X, \text{Τυχαίες σταθερές}\}$
2	Σύνολο συναρτήσεων:	$F = \{+, -, *, \%\}$
3	Καταλληλότητα:	Το άθροισμα της απόλυτης τιμής των διαφορών μεταξύ της εξόδου του υποψήφιου προγράμματος και των δοσμένων δεδομένων (υπολογισμένες πάνω σε πολυάριθμες τιμές της ανεξάρτητης μεταβλητής x από το -1.0 μέχρι το $+1.0$)
4	Παράμετροι:	Μέγεθος πληθυσμού $M = 4$
5	Τερματισμός:	Όταν εμφανίζεται ένα άτομο του οποίου το άθροισμα των απόλυτων λαθών είναι μικρότερο από 0.1

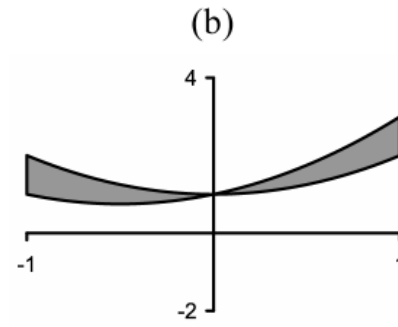
Αρχικός Πληθυσμός για 4 τυχαία δημιουργούμενα άτομα της γενιάς 0.



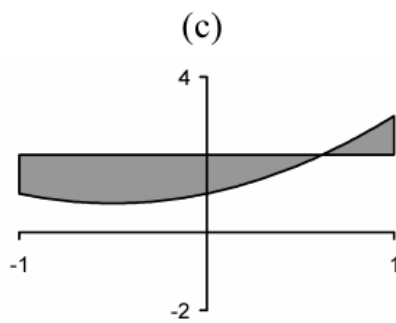
Υπολογίζουμε την απόδοση των ατόμων της γενιάς 0, με χρήση της συνάρτησης καταλληλότητας.



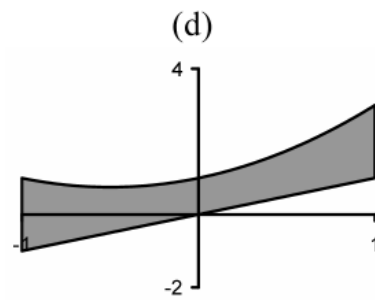
$$F_a=0.67$$



$$F_b=1.00$$

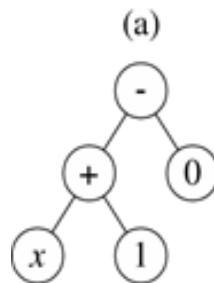


$$F_c=1.70$$



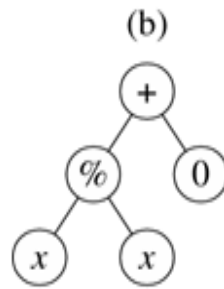
$$F_d=2.67$$

Επιλέγουμε το καλύτερο άτομο και το αντιγράφουμε στην επόμενη γενιά.



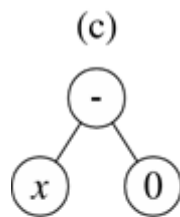
$$x+1$$

Μεταλλάσσουμε το άτομο c, επιλέγοντας το 2 ως σημείο μετάλλαξης και προσθέτουμε ένα τυχαίο υποδένδρο από τις αποδεκτές συναρτήσεις.

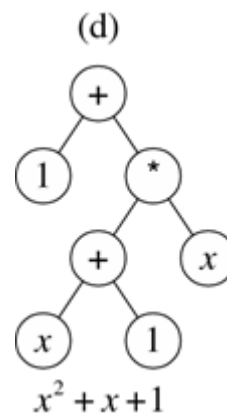


1

Στη συνέχεια διασταυρώνουμε τα άτομα a και b παίρνοντας το + από τον πρώτο γονέα και το πιο αριστερό x από τον δεύτερο γονέα ως σημεία διασταύρωσης. Έτσι δημιουργούνται οι παρακάτω απόγονοι:



x



Άσκηση 10.9

Να ορίσετε τα τέσσερα δομικά στοιχεία της αρχιτεκτονικής ενός προγράμματος.

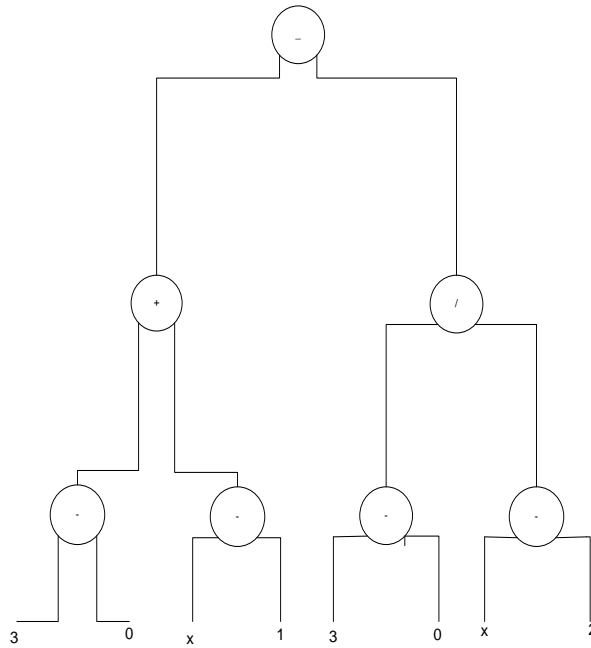
Απάντηση:

Η αρχιτεκτονική ενός προγράμματος αποτελείται από:

- τον συνολικό αριθμό των κλάδων,
- τον τύπο κάθε κλάδου (π.χ. κλάδος παραγωγής αποτελέσματος, αυτόματα παραγόμενη συνάρτηση, αυτόματα οριζόμενη επανάληψη, αυτόματα οριζόμενος βρόχος, αυτόματα οριζόμενη αναδρομή ή αυτόματα οριζόμενη αποθήκευση),
- τον αριθμό των ορισμάτων (αν υπάρχουν) που κατέχονται από κάθε κλάδο,
- τη φύση των ιεραρχικών αναφορών μεταξύ των κλάδων, αν υπάρχουν περισσότεροι από έναν κλάδοι.

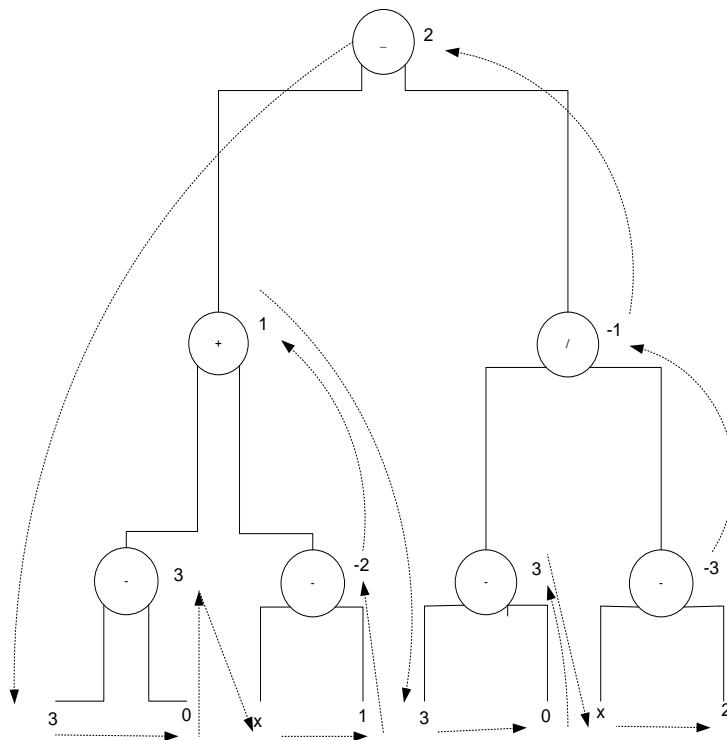
Άσκηση 10.10

Στο παρακάτω δένδρο συντακτικής ανάλυσης, η ανεξάρτητη μεταβλητή X παίρνει την τιμή -1. Να κάνετε την εκτέλεση των συναρτήσεων των κόμβων στο δένδρο με τη σειρά που εγγυάται ότι οι κόμβοι δεν εκτελούνται, αν δεν πάρουν τιμές τα ορίσματά τους. Να δείξετε τη φορά της εκτέλεσης (δηλαδή τη διαπέραση των κόμβων) πάνω στο σχήμα και να βάλετε στα δεξιά των κόμβων την τιμή της αποτίμησης των υποδένδρων που ριζώνουν σε αυτά τα σημεία.



Απάντηση:

Εργαζόμαστε ως εξής: Αρχίζοντας από τη ρίζα, διαπερνάμε τον κόμβο (+) και τον κόμβο (-), τον πρώτο από αριστερά. Επειδή τα φύλλα αυτού του κόμβου είναι σταθερές, υπολογίζεται η τιμή του (=3). Στη συνέχεια παίρνει τιμή η μεταβλητή X (=1) και υπολογίζεται η τιμή του δεύτερου από αριστερά κόμβου (-), που είναι -2. Μπορεί τώρα να εκτελεστεί ο κόμβος (+), που παίρνει τιμή 1. Με τον ίδιο τρόπο διαπερνάμε και το υπόλοιπο δένδρο, όπως φαίνεται στο παρακάτω σχήμα:



Σκοπός αυτού του συγγράμματος είναι να αποτελέσει βασικό εκπαιδευτικό υλικό στην «Υπολογιστική Νοημοσύνη και Βαθιά Μάθηση». Αρχικά παρουσιάζονται οι βασικές αρχές της τεχνολογίας, των ΤΝΔ και των ΓΑ. Γίνεται η σύνδεση με την Τεχνητή Νοημοσύνη (ΤΝ) και τις κλασικές μεθόδους αναζήτησης και βελτιστοποίησης, καθώς και με τα βιολογικά πρότυπα από τα οποία τα έχουν εμπνευστεί. Δίνονται τα βασικά στοιχεία της θεωρίας μάθησης και τα δύο παραδείγματα μάθησης (επιβλεπόμενη και μη επιβλεπόμενη μάθηση). Ακολουθεί η παρουσίαση των βασικών αλγορίθμων εκπαίδευσης των ΤΝΔ, με έμφαση στον πιο διαδεδομένο αλγόριθμο, της πίσω διάδοσης του λάθους (Error Back Propagation – EBP). Τα δίκτυα βαθιάς μάθησης (Deep Learning), εκπαίδευση και μοντέλα, ολοκληρώνουν την παρουσίαση των ΤΝΔ εμπρόσθιας τροφοδότησης. Στη συνέχεια, παρουσιάζονται οι Γενετικοί/Εξελικτικοί Αλγόριθμοι ΓΑ/ΕΑ, αλλά και ο συνδυασμός τους σε έναν υβριδικό αλγόριθμο, ένα Εξελικτικό Νευρωνικό Δίκτυο και η εισαγωγή στον Γενετικό Προγραμματισμό (ΓΠ), μια παραλλαγή των Εξελικτικών Αλγορίθμων, η οποία στηρίζεται στη βασική θεωρία της εξέλιξης του Δαρβίνου. Τέλος, παρουσιάζονται δύο αρχιτεκτονικές ΝΔ, τα δίκτυα Hopfield (ένα είδος αυτοσυσχετιστικής μνήμης) και τα δίκτυα Kohonen (ένα είδος αυτο-οργανούμενων χαρτών για ομαδοποίηση δεδομένων), καθώς και οι αλγόριθμοι εκπαίδευσής τους. Σε όλα τα κεφάλαια τονίζονται οι στόχοι και τα μαθησιακά αποτελέσματα και υπάρχει ικανοποιητικός αριθμός παραδειγμάτων, ασκήσεων και δραστηριοτήτων για την καλύτερη εμπέδωση της ύλης. Η κατανόηση της ύλης προϋποθέτει βασική γνώση διακριτών μαθηματικών, γραμμικής άλγεβρας και συνδυαστικής ανάλυσης, καθώς και γνώση προγραμματισμού υπολογιστών. Όσοι θέλουν να ασχοληθούν με την επίλυση προβλημάτων με χρήση των παραπάνω τεχνολογιών πρέπει να φροντίσουν να αποκτήσουν αυτές τις δεξιότητες, διαφορετικά θα έχουν δυσκολίες και θα χρειαστεί να καταβάλουν μεγάλη προσπάθεια για την επιτυχή μελέτη του υλικού.

Το παρόν σύγγραμμα δημιουργήθηκε στο πλαίσιο του Έργου ΚΑΛΛΙΠΟΣ+	
Χρηματοδότης	Υπουργείο Παιδείας και Θρησκευμάτων, Προγράμματα ΠΔΕ, ΕΠΑ 2020-2025
Φορέας υλοποίησης	ΕΛΚΕ ΕΜΠ
Φορέας λειτουργίας	ΣΕΑΒ/Παράρτημα ΕΜΠ/Μονάδα Εκδόσεων
Διάρκεια 2ης Φάσης	2020-2023
Σκοπός	Η δημιουργία ακαδημαϊκών ψηφιακών συγγραμμάτων ανοικτής πρόσβασης (περισσότερων από 700) <ul style="list-style-type: none">• Προπτυχιακών και μεταπτυχιακών εγχειριδίων• Μονογραφιών• Μεταφράσεων ανοικτών textbooks• Βιβλιογραφικών Οδηγιών
Επιστημονικά Υπεύθυνος	Νικόλαος Μήτρου, Καθηγητής ΣΗΜΜΥ ΕΜΠ
ISBN: 978-618-5726-47-8	DOI: http://dx.doi.org/10.57713/kallipos-168

Το παρόν σύγγραμμα χρηματοδοτήθηκε από το Πρόγραμμα Δημοσίων Επενδύσεων του Υπουργείου Παιδείας