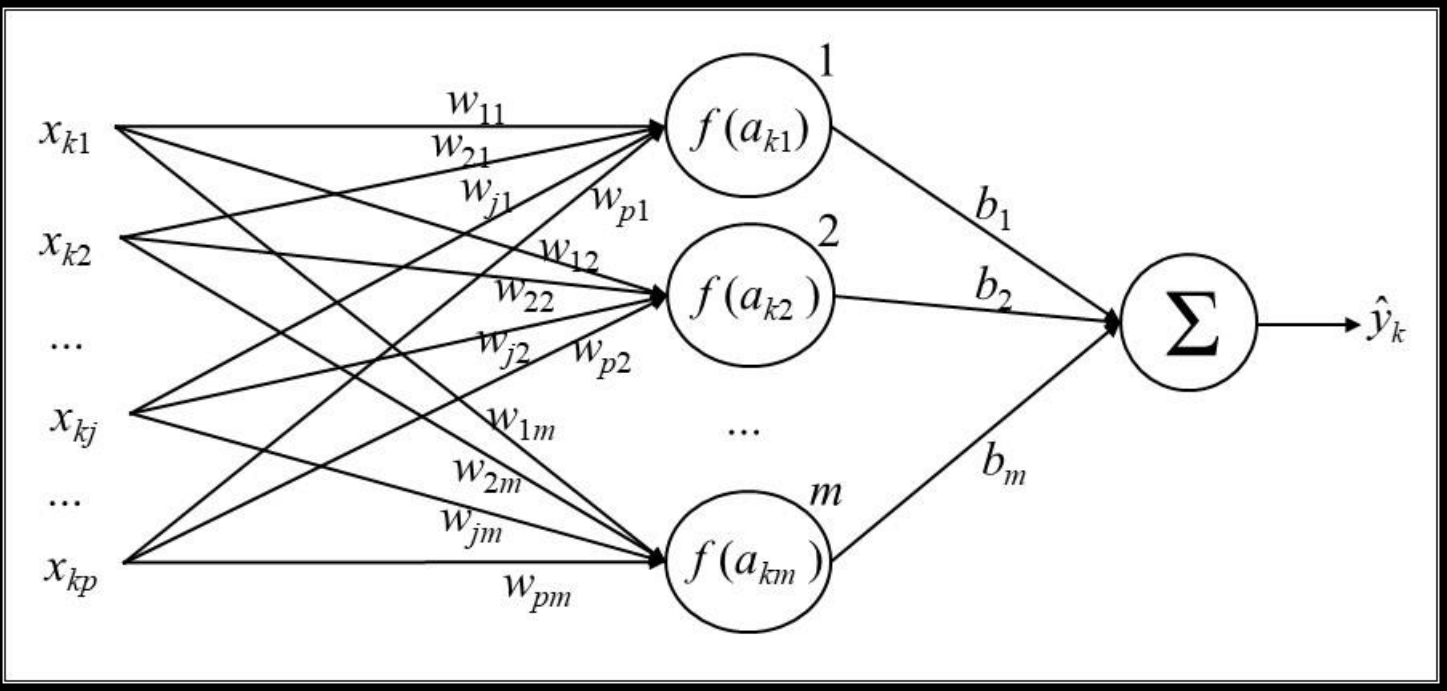


# **Μάθημα 11**

## **Εξελικτικοί Αλγόριθμοι (Evolutionary Algorithms)**

# Εκπαίδευση Νευρωνικών Δικτύων



Συνεπώς, η έξοδος του ΝΔΕΤ με  $m$  νευρώνες είναι:

$$\hat{y}_k = f(a_{k1})b_1 + f(a_{k2})b_2 + \dots + f(a_{km})b_m = \sum_{i=1}^m f(a_{ki})b_i$$



$$a_{ki} = x_{k1}w_{1i} + x_{k2}w_{2i} + \dots + x_{kj}w_{ji} + \dots + x_{kp}w_{pi} = \sum_{j=1}^p x_{kj}w_{ji}$$



$$\hat{y}_k = \sum_{i=1}^m f\left(\sum_{j=1}^p x_{kj}w_{ji}\right)b_i$$



$$f(a_{ki}) = \frac{1}{1 + e^{-a_{ki}}} = \frac{1}{1 + \exp(-a_{ki})}$$



$$\hat{y}_k = \sum_{i=1}^m \frac{1}{1 + \exp\left(-\sum_{j=1}^p x_{kj}w_{ji}\right)} b_i$$

## Εκπαίδευση Νευρωνικών Δικτύων

Έξοδος του ΝΔΕΤ με 2 νευρώνες:

$$\hat{y}_k = \sum_{i=1}^2 \frac{1}{1 + \exp\left(-\sum_{j=1}^p x_{kj} w_{ji}\right)} b_i$$

Έξοδος του ΝΔΕΤ με  $m$  νευρώνες:

$$\hat{y}_k = \sum_{i=1}^m \frac{1}{1 + \exp\left(-\sum_{j=1}^p x_{kj} w_{ji}\right)} b_i$$

Και στις δύο περιπτώσεις η συνάρτηση σφάλματος είναι:

$$E = \sum_{k=1}^N (y_k - \hat{y}_k)^2$$

Για λόγους στατιστικής ευχέρειας στο πρόγραμμα εμφανίζουμε συνήθως την ρίζα του μέσου τετραγωνικού σφάλματος  
Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{\sum_{k=1}^N (y_k - \hat{y}_k)^2}{N}}$$

# Εκπαίδευση Νευρωνικών Δικτύων

## Αντικειμενική Συνάρτηση

$$E(w_{11}, w_{12}, \dots, w_{pm}, b_1, b_2, \dots, b_m) = \sum_{k=1}^N \left( y_k - \sum_{i=1}^m \frac{1}{1 + \exp(-\sum_{j=1}^p x_{kj} w_{ji})} b_i \right)^2$$

## Μερικές Παράγωγοι

$$\frac{\partial E}{\partial w_{ji}} = (-2) \sum_{k=1}^N (y_k - \hat{y}_k) \frac{\exp(-\sum_{j=1}^p x_{kj} w_{ji})}{\left(1 + \exp(-\sum_{j=1}^p x_{kj} w_{ji})\right)^2} x_{kj} b_i$$

$$\frac{\partial E}{\partial b_i} = (-2) \sum_{k=1}^N (y_k - \hat{y}_k) \frac{1}{1 + \exp(-\sum_{j=1}^p x_{kj} w_{ji})}$$

## Κανόνες Εκμάθησης

$$w_{ji}^{new} = w_{ji}^{old} - \eta \frac{\partial E}{\partial w_{ji}^{old}} \Rightarrow w_{ji} = w_{ji} - \eta \frac{\partial E}{\partial w_{ji}}$$

$$b_i^{new} = b_i^{old} - \eta \frac{\partial E}{\partial b_i^{old}} \Rightarrow b_i = b_i - \eta \frac{\partial E}{\partial b_i}$$

# Εκπαίδευση Νευρωνικών Δικτύων

## Συμπερασματικά:

Ο στόχος της εκπαίδευσης ενός Νευρωνικού Δικτύου με  $m$  νευρώνες και  $N$  δεδομένα εκπαίδευσης, τα οποία περιγράφονται από  $p$

μεταβλητές εισόδου, είναι η εύρεση των τιμών των συναπτικών βαρών  $w_{11}, w_{12}, \dots, w_{pm}, b_1, b_2, \dots, b_m$

Οι οποίες όταν αντικατασταθούν στο συνολικό σφάλμα του δικτύου

$$E(w_{11}, w_{12}, \dots, w_{pm}, b_1, b_2, \dots, b_m) = \sum_{k=1}^N \left( y_k - \sum_{i=1}^m \frac{1}{1 + \exp(-\sum_{j=1}^p x_{kj} w_{ji})} b_i \right)^2$$

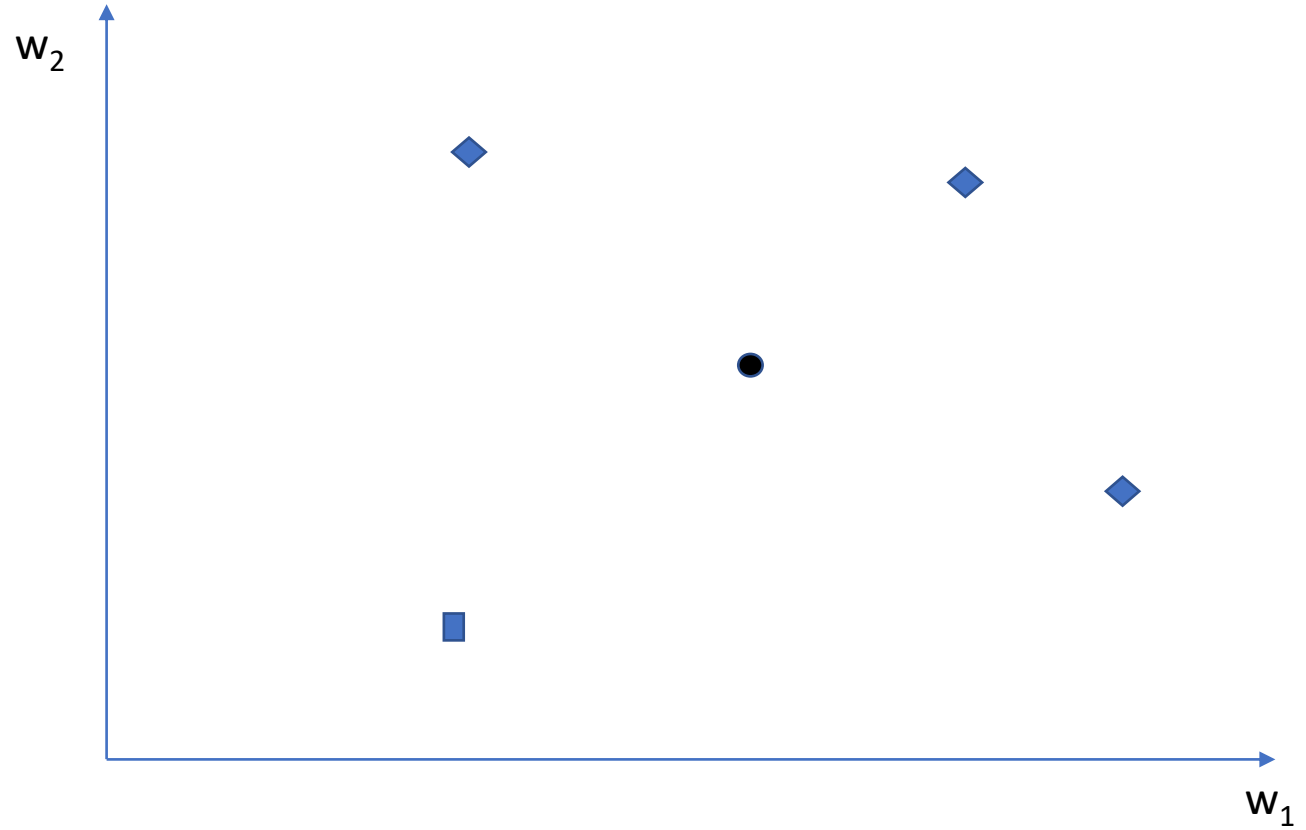
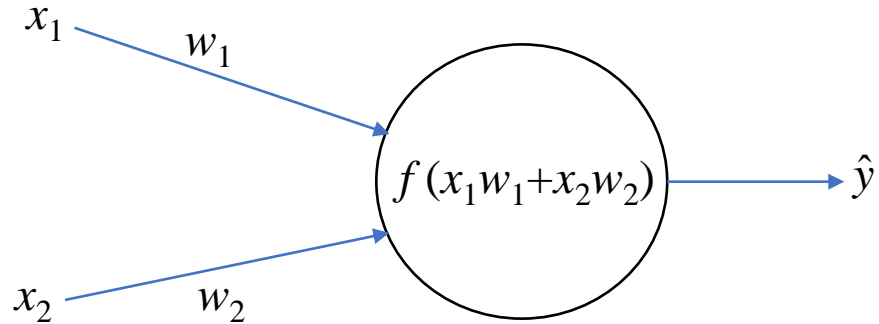
Το σφάλμα γίνεται ελάχιστο (δηλαδή παίρνει την μικρότερη τιμή από ότι παίρνει για οποιοδήποτε άλλον συνδυασμό τιμών αυτών των παραμέτρων).

Ένας τρόπος για να βρούμε αυτές τις τιμές είναι η χρήση του Κανόνα Δέλτα ή της Μεθόδου Καθοδικής Κλίσης (Gradient-Descent Method). Η τελευταία είναι η πιο αποδοτική και είναι μία επαναληπτική διαδικασία σύμφωνα με την οποία, σε κάθε επανάληψη οι τιμές των συναπτικών βαρών ανανεώνονται ως εξής:

$$w_{ji}^{new} = w_{ji}^{old} - \eta \frac{\partial E}{\partial w_{ji}^{old}} \qquad b_i^{new} = b_i^{old} - \eta \frac{\partial E}{\partial b_i^{old}}$$

Όταν τελειώσει η επαναληπτική διαδικασία θα έχουν υπολογιστεί η κατάλληλες τιμές των συναπτικών βαρών

# Gradient Descent



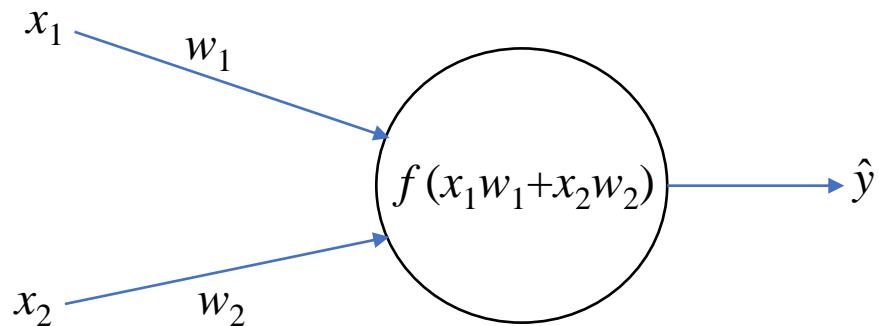
$$w_{ji}^{new} = w_{ji}^{old} - \eta \frac{\partial E}{\partial w_{ji}^{old}}$$

$$b_i^{new} = b_i^{old} - \eta \frac{\partial E}{\partial b_i^{old}}$$

$$E = \sum_{k=1}^N (y_k - \hat{y}_k)^2$$

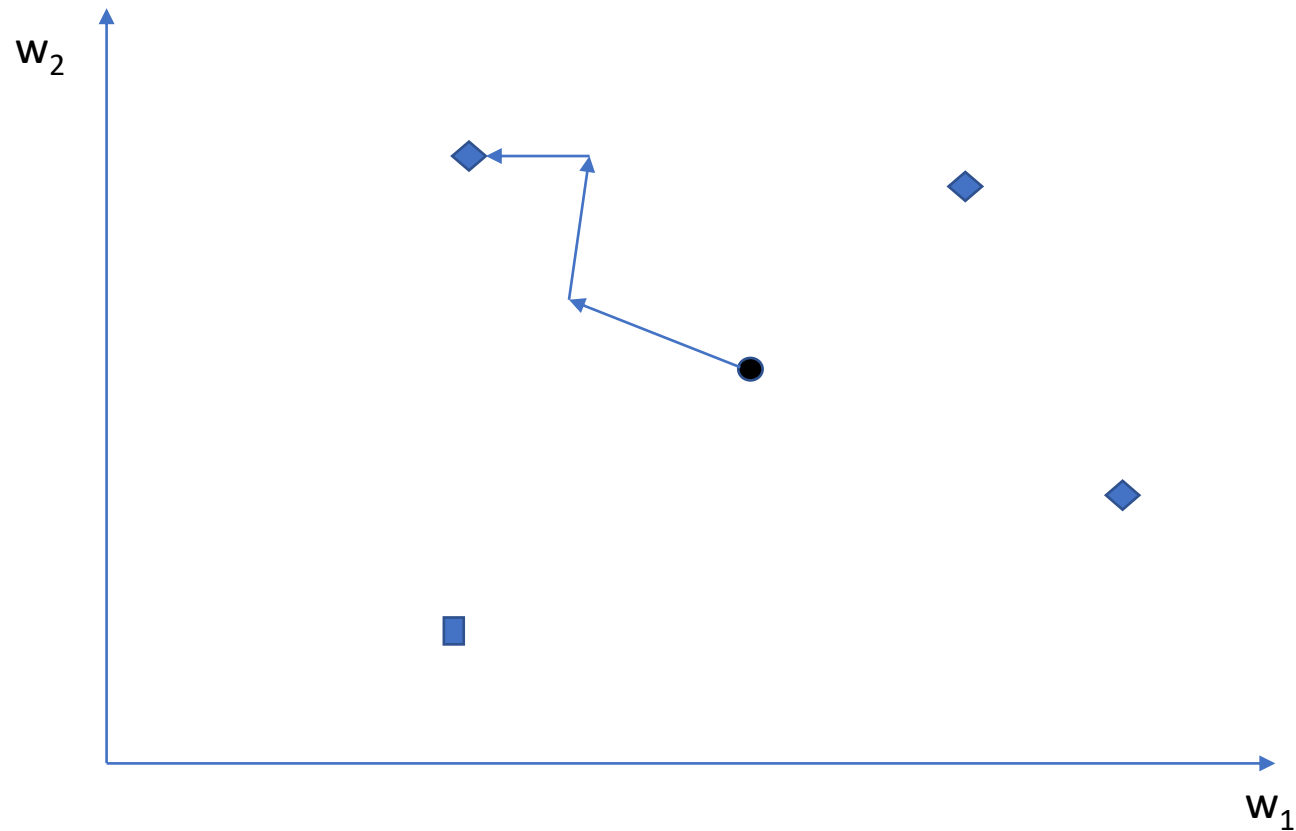
- Νευρωνικό δίκτυο
- ◆ Αποδεκτές λύσεις
- Η καλύτερη λύση

# Gradient Descent



$$w_{ji}^{new} = w_{ji}^{old} - \eta \frac{\partial E}{\partial w_{ji}^{old}}$$

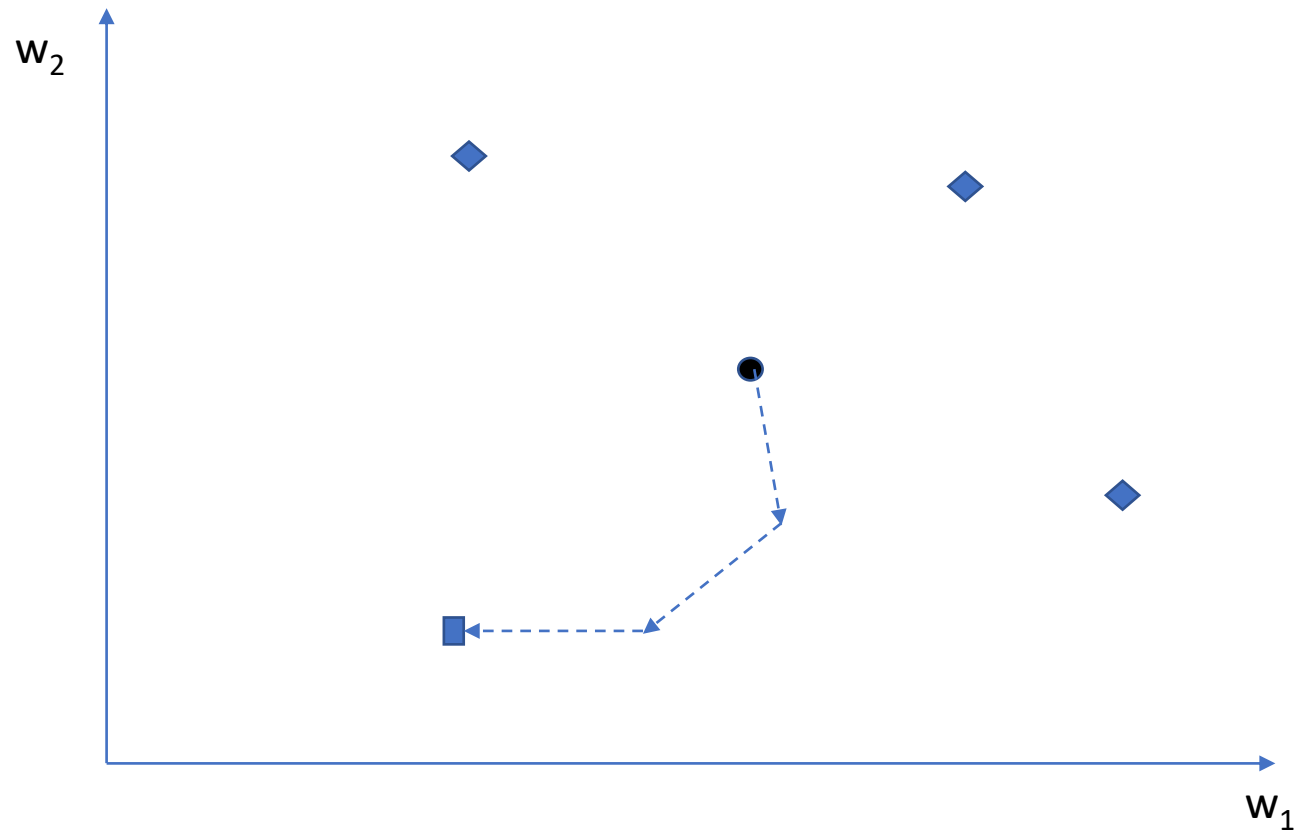
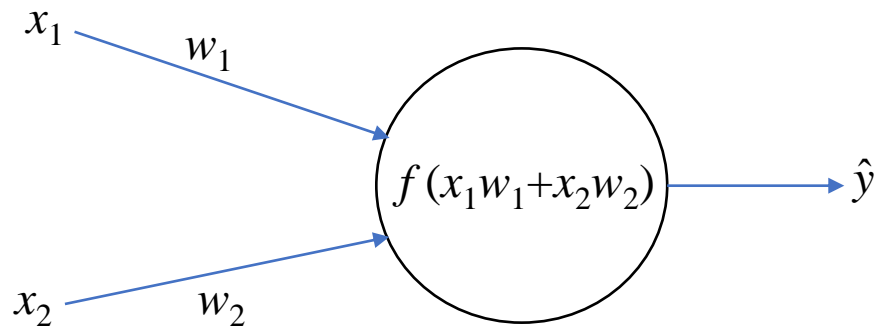
$$b_i^{new} = b_i^{old} - \eta \frac{\partial E}{\partial b_i^{old}}$$



$$E = \sum_{k=1}^N (y_k - \hat{y}_k)^2$$

- Νευρωνικό δίκτυο
- ◆ Αποδεκτές λύσεις
- Η καλύτερη λύση

# Gradient Descent



$$w_{ji}^{new} = w_{ji}^{old} - \eta \frac{\partial E}{\partial w_{ji}^{old}}$$

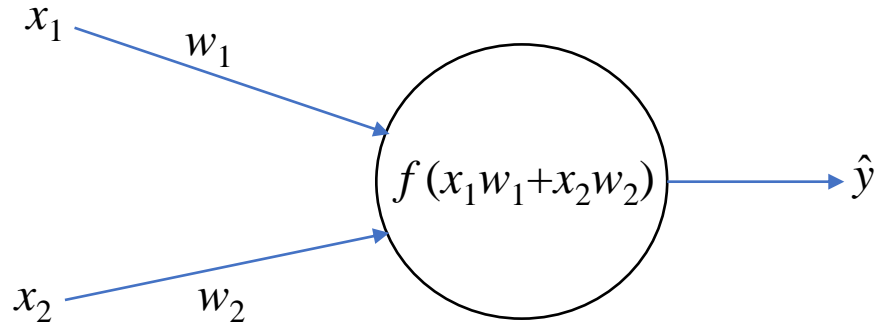
$$b_i^{new} = b_i^{old} - \eta \frac{\partial E}{\partial b_i^{old}}$$

$$E = \sum_{k=1}^N (y_k - \hat{y}_k)^2$$

- Νευρωνικό δίκτυο
- ◆ Αποδεκτές λύσεις
- Η καλύτερη λύση

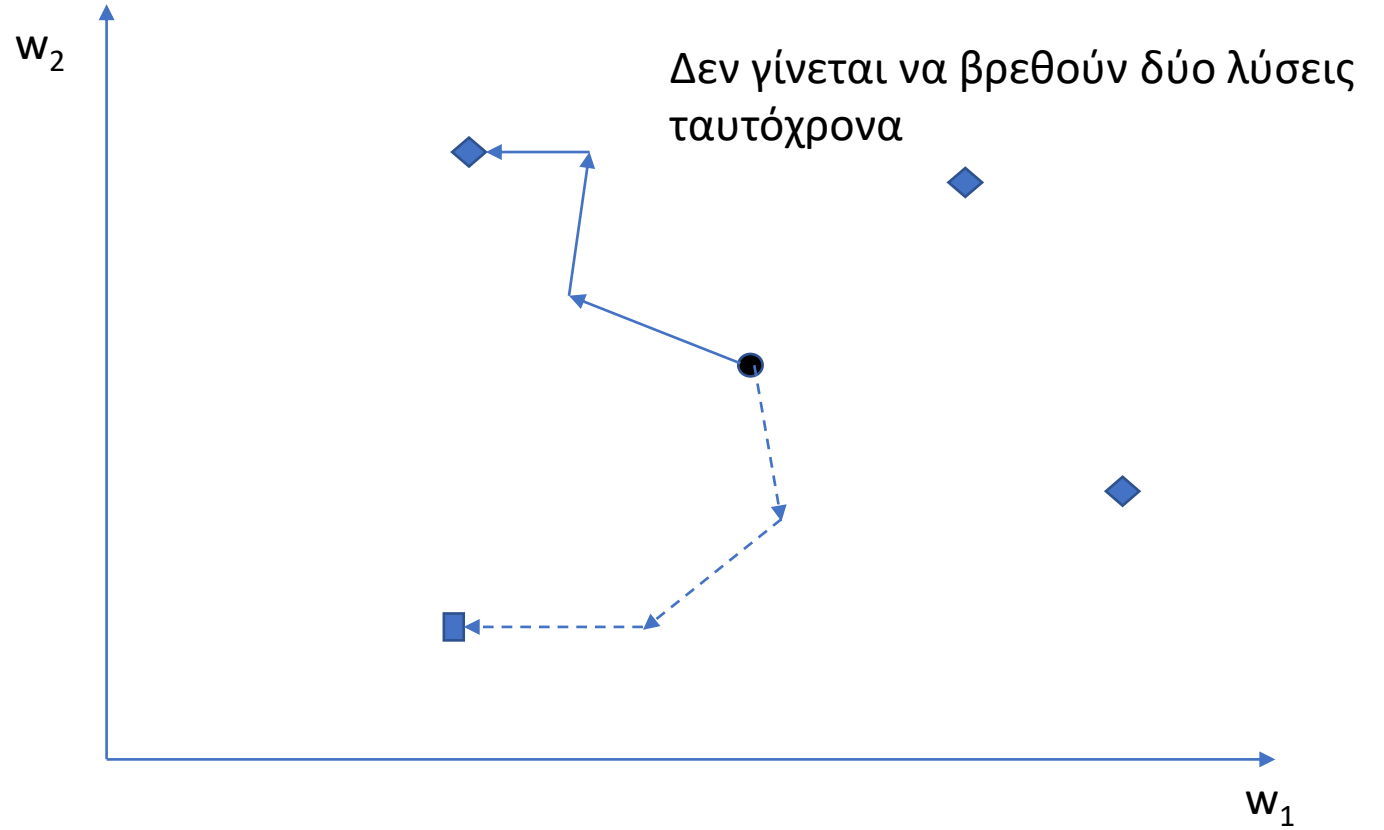


# Gradient Descent



$$w_{ji}^{new} = w_{ji}^{old} - \eta \frac{\partial E}{\partial w_{ji}^{old}}$$

$$b_i^{new} = b_i^{old} - \eta \frac{\partial E}{\partial b_i^{old}}$$



$$E = \sum_{k=1}^N (y_k - \hat{y}_k)^2$$

- Νευρωνικό δίκτυο
- ◆ Αποδεκτές λύσεις
- Η καλύτερη λύση

# Μηχανική Μάθηση

## Ερώτηση:

Υπάρχει άλλος τρόπος να εκπαιδεύσουμε ένα νευρωνικό δίκτυο; Δηλαδή να βρούμε τις τιμές των συναπτικών βαρών που ελαχιστοποιούν το σφάλμα του δικτύου με άλλον τρόπο;

**Απάντηση:** Πάρα πολλές

**Παρατήρηση:** Κάθε μέθοδος που ελαχιστοποιεί την συνάρτηση σφάλματος (και άρα εκπαιδεύει ένα σύστημα Τεχνητής Νοημοσύνης ονομάζεται μέθοδος **Μηχανικής Μάθησης (Machine Learning)**

Άρα η Μηχανική Μάθηση είναι ένα σύνολο μεθόδων που ελαχιστοποιούν αντικειμενικές συναρτήσεις σφάλματος ή και άλλες αντικειμενικές συναρτήσεις με στόχο την εκπαίδευση συστημάτων Τεχνητής Νοημοσύνης και άρα τον σχεδιασμό, την ανάπτυξη και υλοποίηση συστημάτων Τεχνητής Νοημοσύνης. Τα συστήματα αυτά ονομάζονται ΜΟΝΤΕΛΑ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ.

Τα Νευρωνικά Δίκτυα και τα Ασαφή Συστήματα ΕΙΝΑΙ Μοντέλα Τεχνητής Νοημοσύνης



# Εξελικτικοί Αλγόριθμοι

Είναι μέθοδοι Μηχανικής Μάθησης που βασίζονται σε παραδείγματα ΣΥΛΛΟΓΙΚΗΣ ΝΟΗΜΟΣΥΝΗΣ από το ζωικό βασίλειο





## Νοημοσύνη Σμήνους στο Ζωικό Βασίλειο

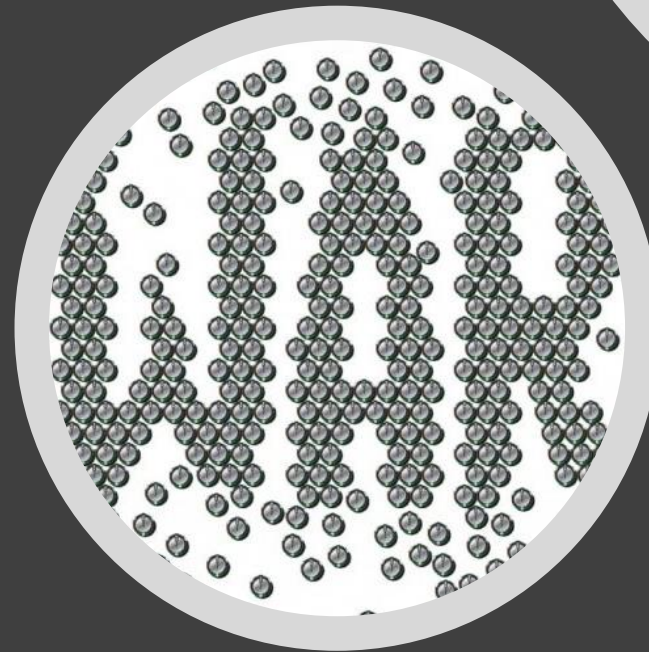
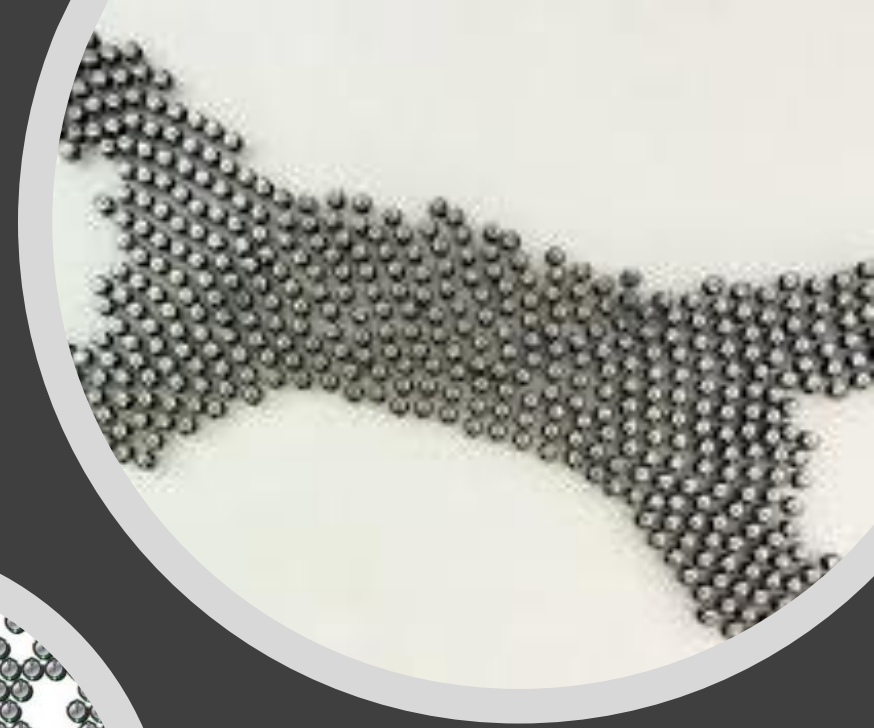
- Πολλές κοινότητες ζώων επιδεικνύουν ικανότητες συντονισμένης και συνισταμένης πλοήγησης σε πολλά γκρουπ σμήνους, τα οποία αποτελούνται από δεκάδες έως και χιλιάδες άτομα.
- Κάθε άτομο, για να πλοηγηθεί, χρησιμοποιεί τοπική πληροφορία από άλλα άτομα που βρίσκονται κοντά του.
- Σε γενικές γραμμές, δεν υπάρχει κάποιος αρχηγός στο σμήνος.
- Σε σπάνιες, περιπτώσεις μπορεί να υπάρξει κάποιος αρχηγός, αλλά και σε αυτή την περίπτωση τα άτομα παίρνουν πληροφορία από τους γείτονες τους και όχι από τον αρχηγό.





# Εξελικτικοί Αλγόριθμοι

- Οι μέθοδοι μηχανικής μάθησης που βασίζονται στην φιλοσοφία της συλλογικής νοημοσύνης ονομάζονται Εξελικτικοί Αλγόριθμοι.
- Μία υποκατηγορία Εξελικτικών Αλγόριθμων είναι οι Αλγόριθμοι Νοημοσύνης Σμήνους (Swarm Intelligence)
- Κάτι αντίστοιχο με την συλλογική νοημοσύνη στο ζωικό βασίλειο είναι η τεχνητή συλλογική νοημοσύνη που αφορά σμήνη από drones και robots



# Τεχνητή Νοημοσύνη Σμήνους (Artificial Swarm Intelligence)

Η Νοημοσύνη Σμήνους είναι μία σχετικά καινούργια ερευνητική περιοχή

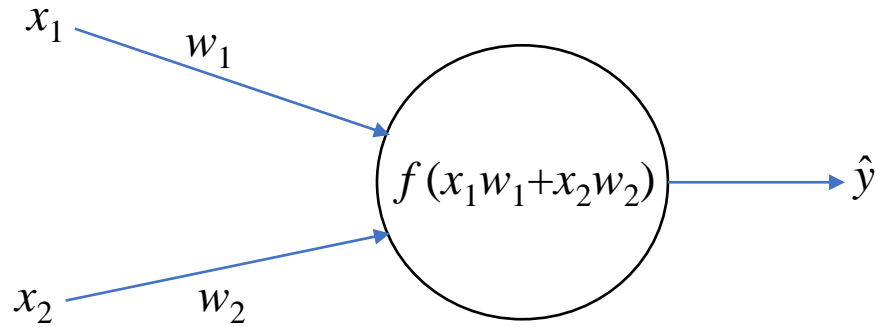
- Χρησιμοποιεί ένα σύνολο ατόμων που ονομάζονται και σωματίδια, όπου κάθε ένα από αυτά αντιστοιχεί σε ένα ολόκληρο Μοντέλο Τεχνητής Νοημοσύνης.
- Κάθε άτομο βρίσκει μία λύση. Η λύση που θα υιοθετήσει τελικά το σμήνος θα είναι η καλύτερη λύση που έχει βρεθεί από όλα τα άτομα του σμήνους.
- Πλεονεκτήματα
  - Το σμήνος είναι σε θέση να επιλύσει πολύπλοκα προβλήματα ελαχιστοποίησης σύνθετων και μεγάλης πολυπλοκότητας αντικειμενικών συναρτήσεων.
  - Σε αντίθεση με τον Κανόνα Δέλτα ή την μέθοδο Gradient-Descent, τα οποία ελέγχουν μόνο ένα μοντέλο, το σμήνος ελέγχει τόσα μοντέλα όσα είναι και τα άτομα που το απαρτίζουν. Αυτό σημαίνει ότι «σαρώνουν» όλον τον χώρο και η πιθανότητα να βρεθεί κακή λύση είναι πολύ μεγαλύτερη (οι πολλοί έχουν καλύτερη απόδοση από τον έναν).
  - Πολλά άτομα (σωματίδια) μπορεί να μην έχουν καλή απόδοση, αυτό όμως δεν επηρεάζει την συνολική απόδοση του σμήνους γιατί έστω και ένα άτομο να βρει μία καλή λύση, το σμήνος θα καταλήξει σε αυτή την λύση (δηλ. στην καλύτερη λύση που έχει βρεθεί από όλα τα άτομα)
- Μειονεκτήματα
  - Τα άτομα σε ένα σμήνος έχουν τοπική πληροφορία με βάση την οποία θα επιτελέσει απλές ενέργειες σε τοπικό επίπεδο. Δεν έχουν, δηλ., αίσθηση ούτε γνώση για την συνολική απόδοση του σμήνους
  - Έχει μεγάλο υπολογιστικό κόστος γιατί ελέγχονται πάρα πολλά μοντέλα σε κάθε επανάληψη

## Particle Swarm Algorithm

Βασίζεται σε σμήνη πουλιών και στο πως συμπεριφέρονται όταν ψάχνουν για τροφή

- Το σμήνος θέλει να βρει την περιοχή με την μεγαλύτερη πυκνότητα τροφής (δηλ. έντομα)
- Τα πουλιά δεν ξέρουν που βρίσκεται αυτή η περιοχή
- Το καθένα από αυτά επικοινωνεί με του γείτονες του και τους πληροφορεί για το αν υπάρχουν έντιμα στην θέση που βρίσκεται αυτό και πόσα
- Επίσης, το κάθε πουλί θυμάται σε ποιο σημείο (θέση/τοποθεσία) βρήκε τα πιο πολλά έντομα
- Για να βρει άφθονη τροφή το σμήνος, το κάθε πουλί εκτελεί με συγκεκριμένο τρόπο τα παρακάτω τρία βήματα (τελεστές):
  - ✓ Συνεχίζει στην ίδια κατεύθυνση μέχρι να βρει κάτι
  - ✓ Αν δεν βρει κάτι καλό σε συγκεκριμένο χρονικό διάστημα, γυρίζει πίσω στην καλύτερη θέση που βρήκε
  - ✓ Ακολουθεί μία καινούργια κατεύθυνση βασισμένο στην πληροφορία που πήρε από τα κοντινά σε αυτό πουλιά

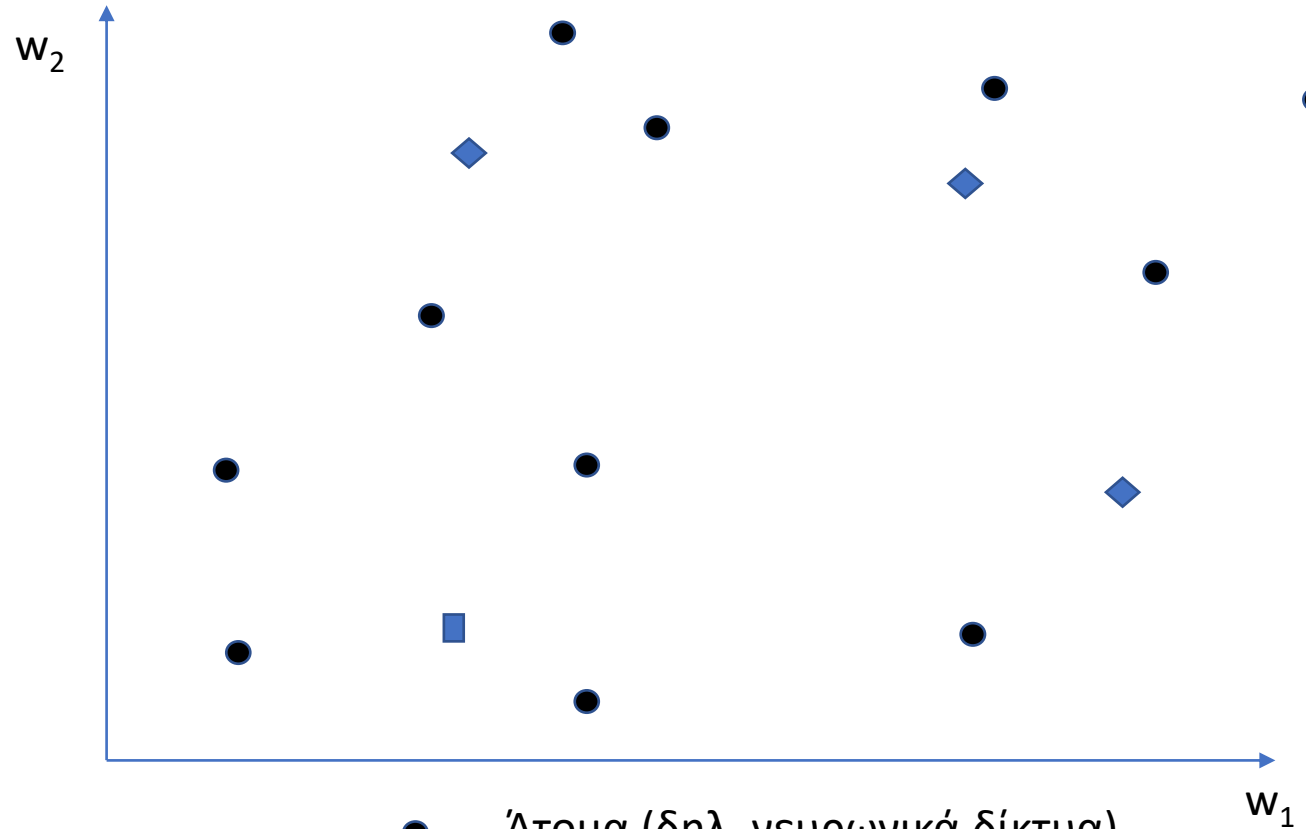
# Particle Swarm Algorithm



$$E = \sum_{k=1}^N (y_k - \hat{y}_k)^2$$

Προσδιορισμός καλύτερης λύσης κάθε σωματιδίου

Προσδιορισμός καλύτερης λύσης

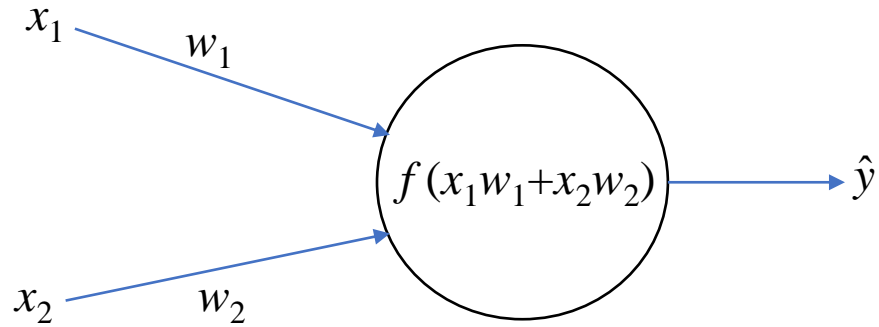


- Άτομα (δηλ. νευρωνικά δίκτυα)
- ◆ Αποδεκτές λύσεις
- Η καλύτερη λύση

Αρχικοποίηση: Τα άτομα αρχικοποιούνται τυχαία



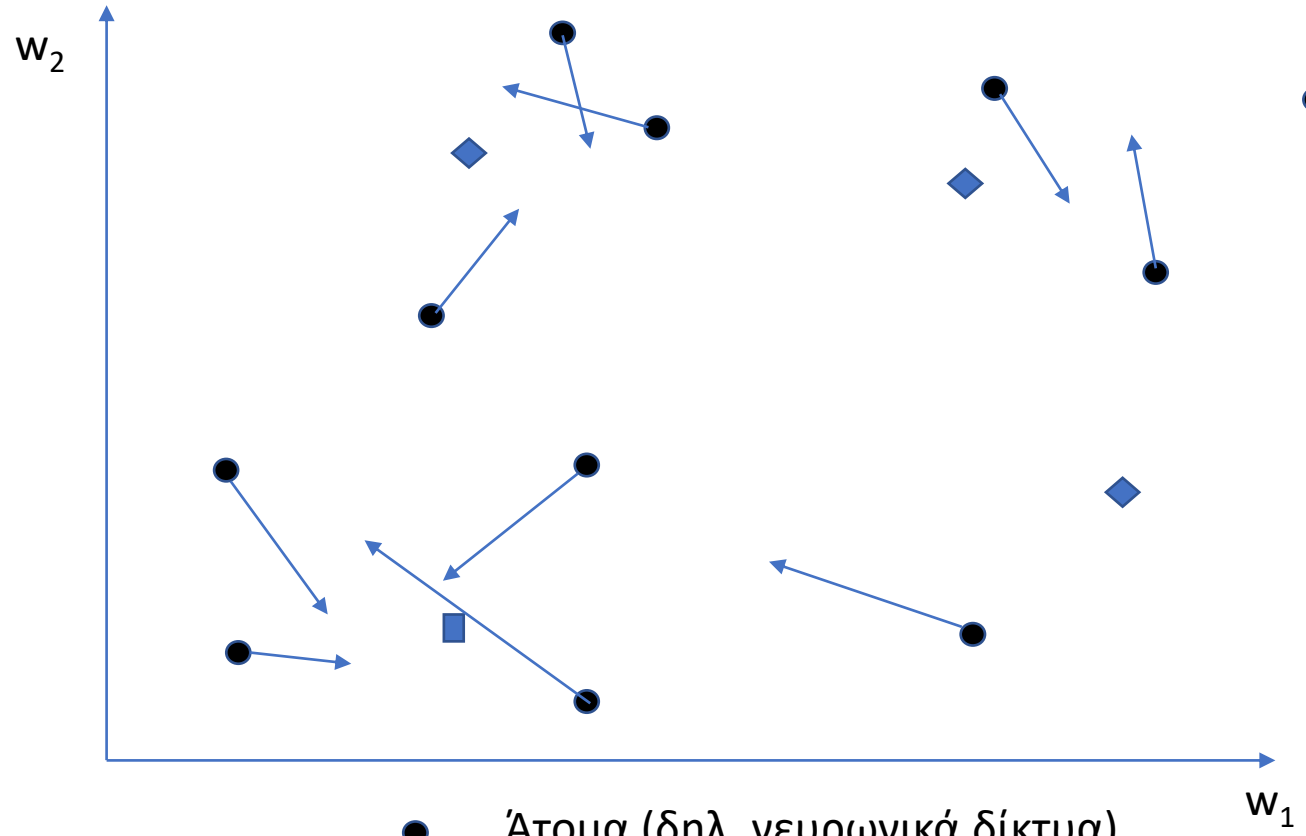
# Particle Swarm Algorithm



$$E = \sum_{k=1}^N (y_k - \hat{y}_k)^2$$

Προσδιορισμός καλύτερης λύσης κάθε σωματιδίου

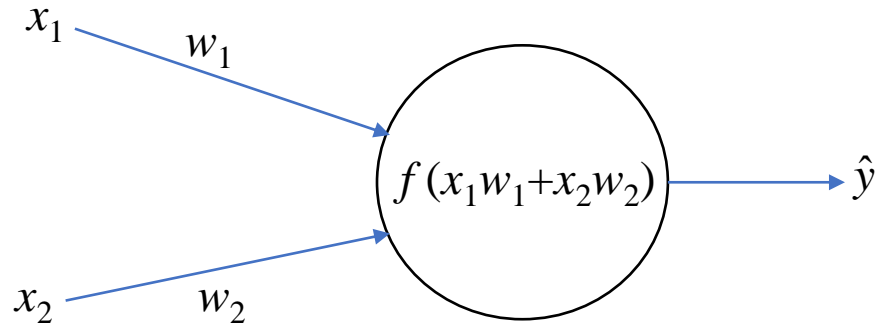
Αναθεώρηση καλύτερης λύσης



- Άτομα (δηλ. νευρωνικά δίκτυα)
- ◆ Αποδεκτές λύσεις
- Η καλύτερη λύση

Πρώτη επανάληψη: Τα άτομα μετακινούνται με βάση πληροφορία από τους γειτόνους τους

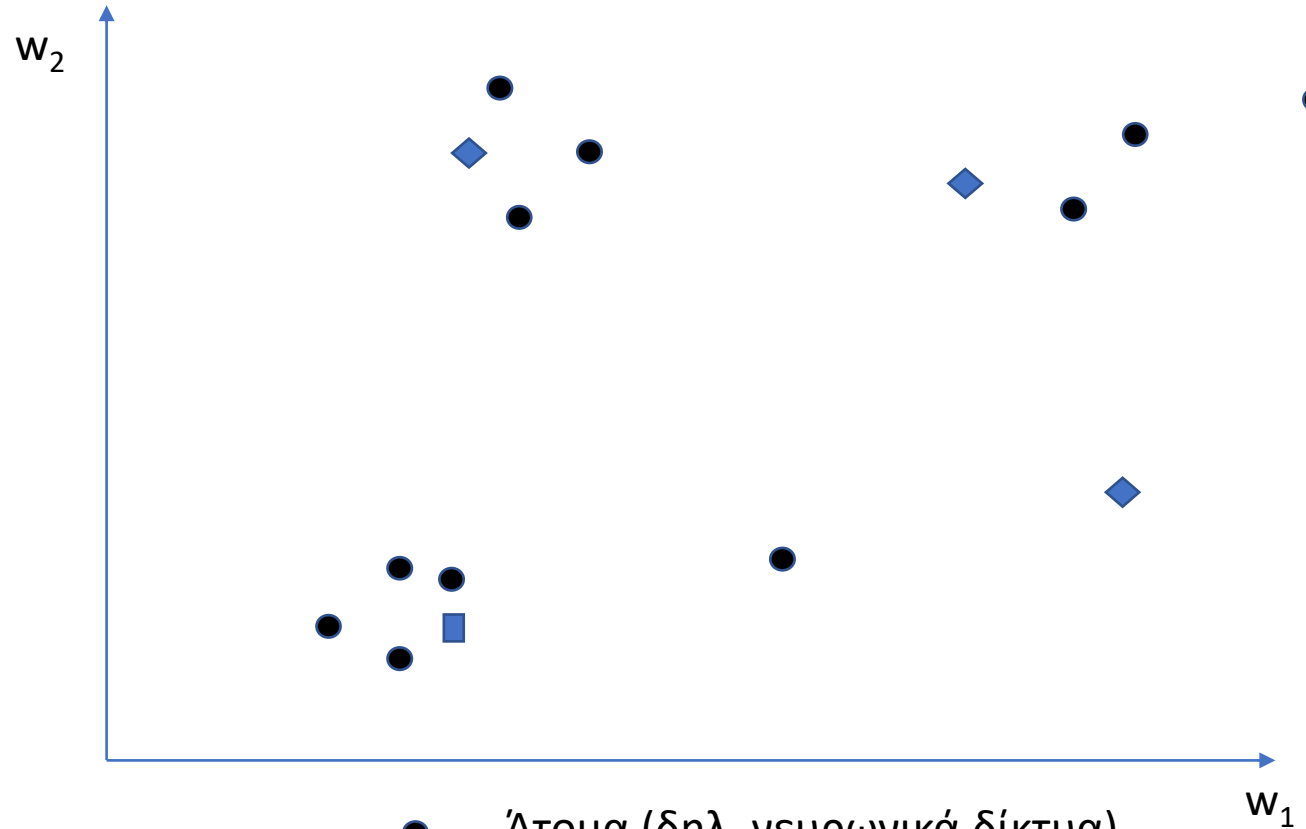
# Particle Swarm Algorithm



$$E = \sum_{k=1}^N (y_k - \hat{y}_k)^2$$

Προσδιορισμός καλύτερης λύσης κάθε σωματιδίου

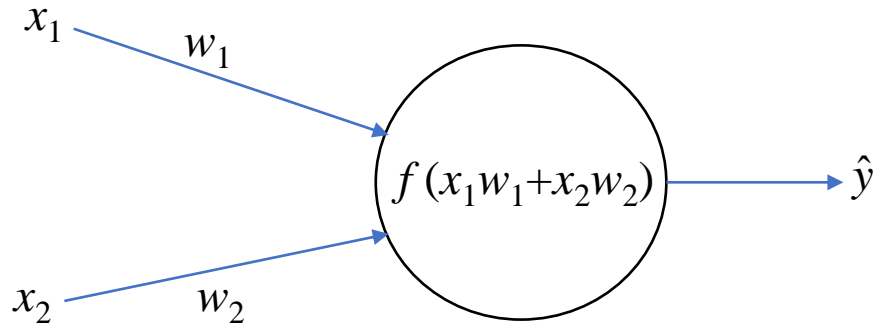
Αναθεώρηση καλύτερης λύσης



- Άτομα (δηλ. νευρωνικά δίκτυα)
- ◆ Αποδεκτές λύσεις
- Η καλύτερη λύση

Δεύτερη επανάληψη: Τα άτομα παίρνουν την πληροφορία από τους γειτόνους τους

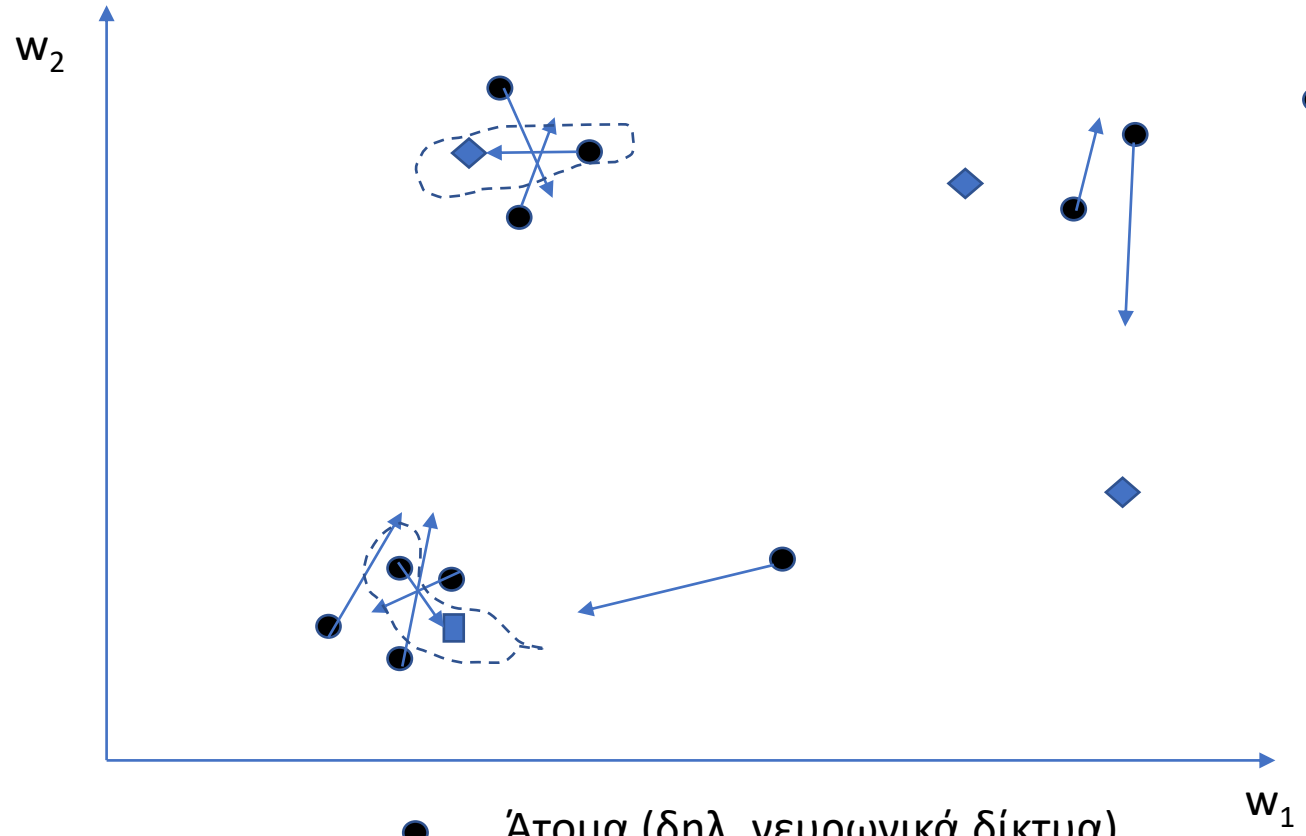
# Particle Swarm Algorithm



$$E = \sum_{k=1}^N (y_k - \hat{y}_k)^2$$

Προσδιορισμός καλύτερης λύσης κάθε σωματιδίου

Αναθεώρηση καλύτερης λύσης



- Άτομα (δηλ. νευρωνικά δίκτυα)
- ◆ Αποδεκτές λύσεις
- Η καλύτερη λύση

Τρίτη επανάληψη: Τα άτομα μετακινούνται με βάση πληροφορία από τους γείτονες τους

## Artificial Bee Colony Algorithm

- Βασίζεται στον τρόπο που συμπεριφέρονται οι μέλισσες για να βρουν τροφή
- Χρησιμοποιεί τα παρακάτω είδη μελισσών:
  - ✓ Εργάτρια μέλισσα (Employed bee): Είναι η μέλισσα που έχει επισκεφθεί θέσεις με τροφή και ξαναγυρίζει σε μία από αυτές
  - ✓ Παρατηρήτρια μέλισσα (Onlooker bee): Είναι η μέλισσα που περιμένει να δει τον χορό των εργατριών που έχουν γυρίσει στην κυψέλη ώστε να αποφασίσει που θα πάει να βρει τροφή
  - ✓ Εξερευνήτρια μέλισσα (Scout bee): Είναι η μέλισσα που πάει μόνη της να επισκεφθεί τυχαίες τοποθεσίες με την ελπίδα να βρει καινούργιες πηγές τροφής

**ΚΑΛΟ ΑΠΟΓΕΥΜΑ**