



# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΕΡΓΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## Επισκόπηση Νευρωνικών Δικτύων

καθ. Βασίλης Μάγκλαρης

[maglaris@netmode.ntua.gr](mailto:maglaris@netmode.ntua.gr)

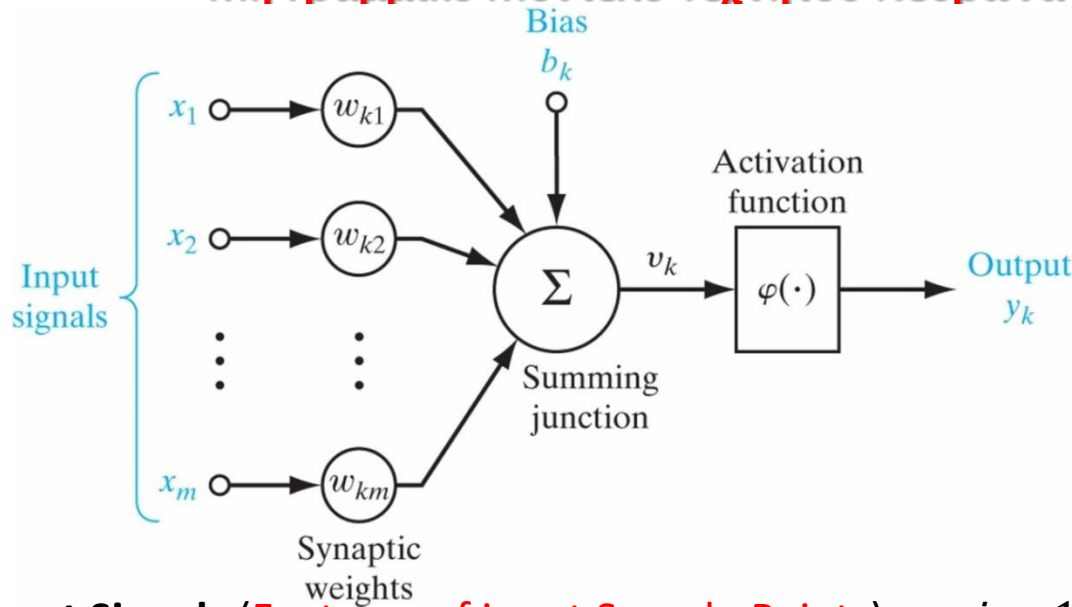
[www.netmode.ntua.gr](http://www.netmode.ntua.gr)

(μέσω Webex)

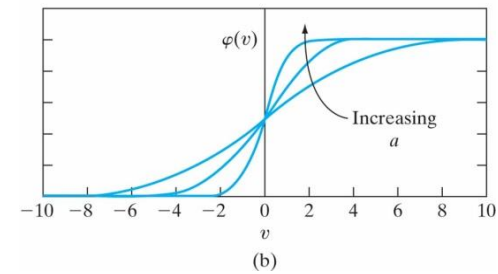
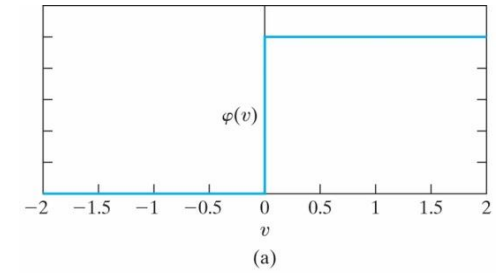
Πέμπτη 4/3/2021

# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## Μη Γραμμικό Μοντέλο Τεχνητού Νευρώνα $k$ (McCulloch-Pitts, 1943)



### Κοινές Συναρτήσεις Διέγερσης Συνάρτηση Κατωφλίου (Threshold Function)



**Input Signals (Features of input Sample Points):**  $x_j, j = 1, 2, \dots, m$

**Synaptic Weights:**  $w_{kj}, j = 1, 2, \dots, m$

**Bias  $b_k$ :**  $b_k \triangleq w_{k0}$ , **Intercept term**  $x_0 \triangleq +1$

**Induced Local Field – Activation Potential:**  $v_k = \sum_{j=0}^m w_{kj} x_j$

**Συνάρτηση Διέγερσης (Activation Function)  $\varphi(\cdot)$ :** Έξοδος  $y_k = \varphi(v_k)$

Συνήθειες Επιλογές:

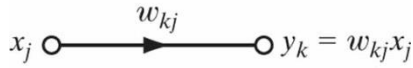
- Συνάρτηση Κατωφλίου (Threshold Function):**  $y_k = \varphi(v_k)$  **δυναμική** μεταβλητή  $\{0, 1\}$   
(McCulloch-Pitts Neural Network Model)
- Συνάρτηση Πρόσημου (Signum Function):**  $y_k = \varphi(v_k)$  **δυναμική** μεταβλητή  $\{-1, 1\}$   
(Hard Limiter)
- Σιγμοειδής Συνάρτηση (Logistic Function):**  $y_k = \varphi(v_k)$  **συνεχής** μεταβλητή  $\in (0, 1)$

**Σιγμοειδής Συνάρτηση  
(Sigmoid – Logistic Function)**

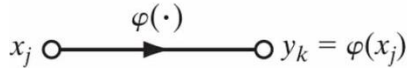
$$\varphi(v) = \frac{1}{1 + \exp(-av)}$$

# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

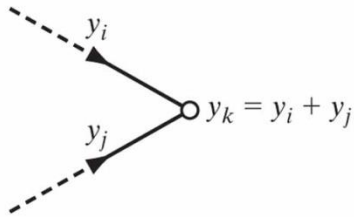
## Νευρωνικά Δίκτυα σαν Κατευθυνόμενοι Γράφοι



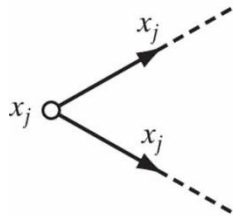
(a)



(b)

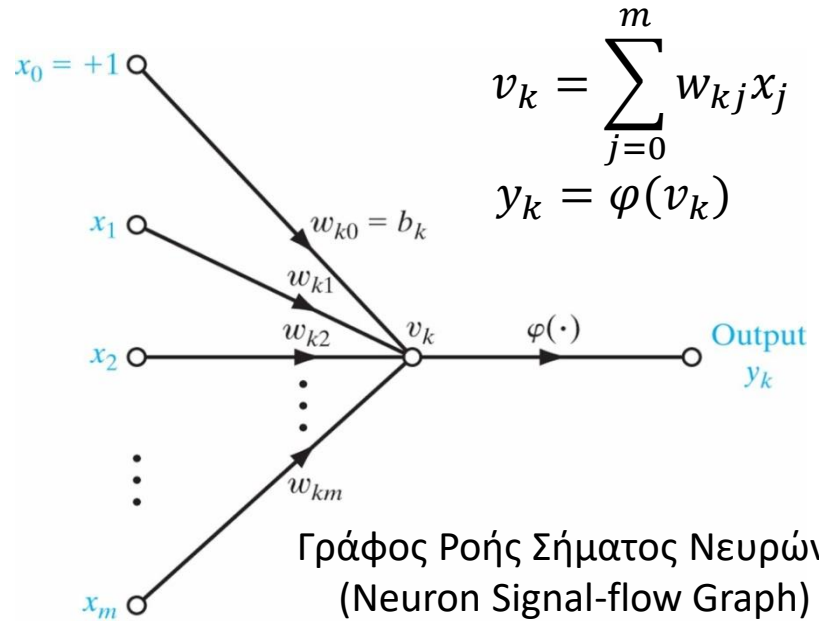


(c)

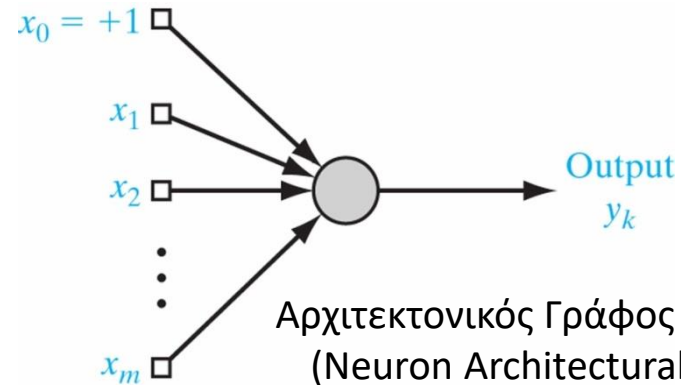


(d)

Κανόνες Γράφων Ροής Σήματος  
(Signal-flow Graph Rules)



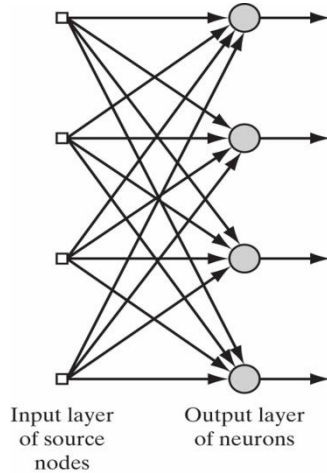
Γράφος Ροής Σήματος Νευρώνα  
(Neuron Signal-flow Graph)



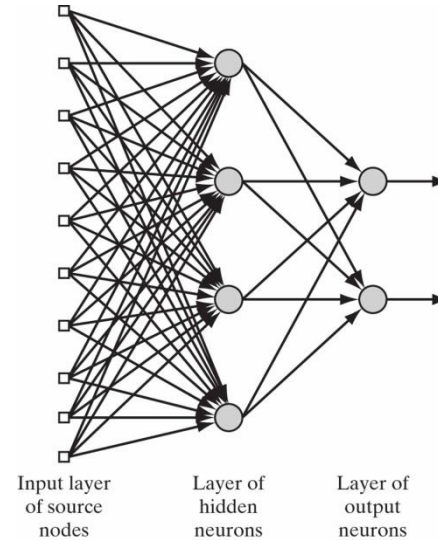
Αρχιτεκτονικός Γράφος Νευρώνα  
(Neuron Architectural Graph)

# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

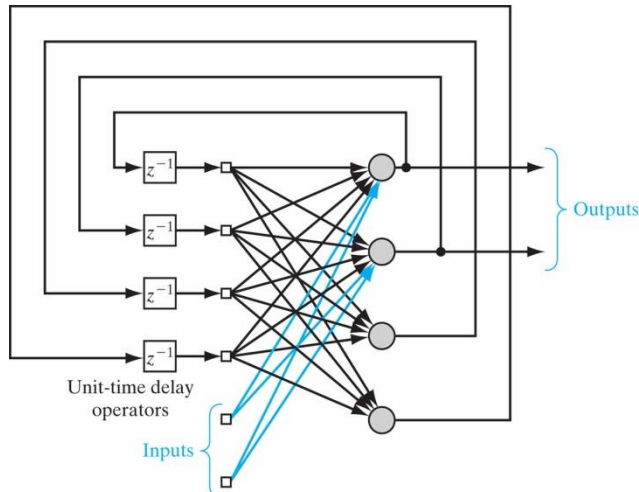
## Μοντέλα Νευρωνικών Δικτύων



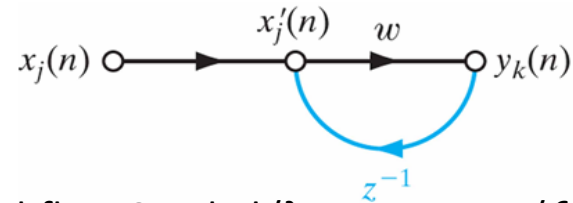
Μονοστρωματικό Δίκτυο Πρόσθιας Τροφοδότησης  
(Single-Layer Feedforward Network)



Δίκτυο Πρόσθιας Τροφοδότησης με Κρυφούς Νευρώνες  
(Multilayer Feedforward Network with Hidden Neurons)



Αναδρομικό Δίκτυο με Κρυφούς Νευρώνες  
(Recurrent Network with Hidden Neurons)



Signal-flow Graph Φίλτρου IIR 1<sup>ης</sup> Τάξης με Ανάδραση  
(Infinite Impulse Response - IIR Filter with Feedback)

$x_j(n)$ : Σήμα Εισόδου στην διακριτή στιγμή  $n$   
 $y_k(n)$ : Σήμα Εξόδου στην διακριτή στιγμή  $n$   
 $x'_j(n)$ : Εσωτερικό Σήμα στην διακριτή στιγμή  $n$

$$y_k(n) = \sum_{l=0}^{\infty} w^{l+1} x_j(n-l)$$

# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## Αναπαράσταση Γνώσης (Knowledge Representation)

Διακριτική ικανότητα (προβλέψεις, ερμηνεία και επιθυμητή απόκριση) σε εξωτερικές εισόδους του **περιβάλλοντος** που θα λειτουργήσει ένα νευρωνικό δίκτυο με:

- ❖ Πρότερη Πληροφορία (**Prior Information**)
- ❖ Μετρήσεις – Παρατηρήσεις Παραδειγμάτων (**Observations**)
  - Συνήθως εμπεριέχουν θόρυβο (noisy sensor errors)
  - Παρέχουν υποδείγματα εισόδου για **training** νευρωνικών δικτύων
    - **Labeled** (αντιστοίχιση με επιθυμητά χαρακτηριστικών εξόδου με μεσολάβηση **εκπαιδευτή**)
    - **Unlabeled** (χωρίς αντιστοίχιση χαρακτηριστικών εξόδου)

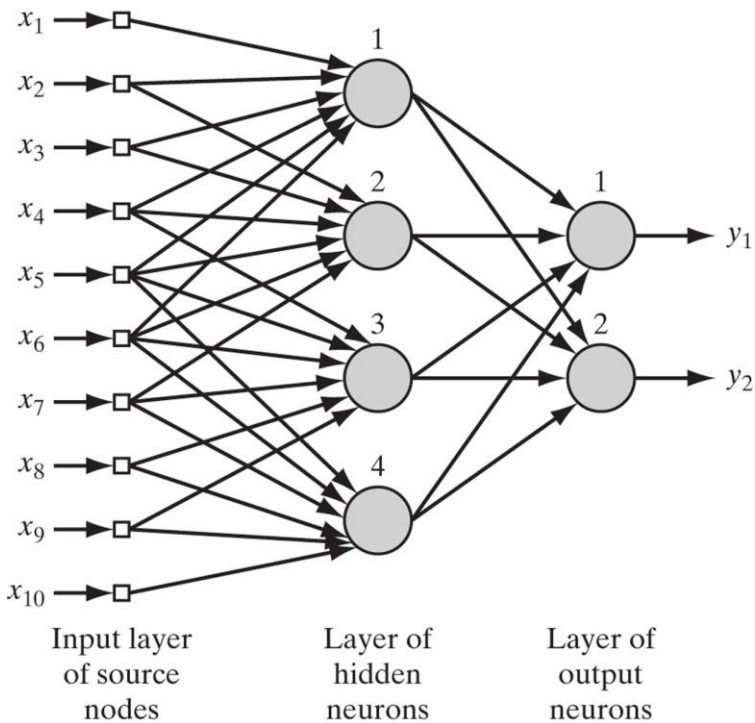
## Φάσεις Σχεδίασης Νευρωνικών Δικτύων (ΝΔ):

- ❖ Φάση Μάθησης (**Learning**) με χρήση δείγματος μάθησης δεδομένων (**training sample points**) από το περιβάλλον:
  - Επιλογή αρχιτεκτονικής (συνδεσμολογία νευρώνων, συνάρτηση διέγερσης)
  - Προσδιορισμός synaptic weights  $w_{kj}$  και biases  $b_k$
- ❖ Φάση Επικύρωσης (**Validation**): Επαλήθευση διακριτικής ικανότητας ΝΔ με χρήση επιπλέον χαρακτηρισμένων δεδομένων (**validation sample points**)
- ❖ Φάση Ελέγχου (**Testing**) επίδοσης σε νέα δεδομένα (**testing sample points**):  
Γενίκευση (**generalization**) διακριτικής ικανότητας ΝΔ

# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## Ενσωμάτωση Πρότερης Πληροφορίας στη Σχεδίαση Νευρωνικών Δικτύων (Building Prior Information into Neural Network Design)

Γενικές επιλογές αρχιτεκτονικής με οδηγό Πρότερη Πληροφορία συγκεκριμένης εμπειρίας



### Επιλογές Απλοποίησης Αρχιτεκτονικής Νευρωνικού Δικτύου:

1. Αρχιτεκτονική με ορισμό πεδίου υποδοχής (*receptive field*) ίσου αριθμού εισόδων ανά κρυφό νευρώνα
2. Διαμοιρασμός βαρών (*weight sharing*): Στο παράδειγμα με 10 κόμβους εισόδου και ένα στρώμα 4 κρυφών νευρώνων, ορίζονται κοινά βάρη  $w_i, i = 1, \dots, 6$  για συνάψεις 6 εναλλακτικών επιλογών εισόδων από τις  $x_i, i = 1, \dots, 10$  προς τους κρυφούς νευρώνες 1,2,3,4

$$v_j = \sum_{i=1}^6 w_i x_{i+j-1}, \quad j = 1, 2, 3, 4$$

Συνελικτικά αθροίσματα (*Convolutional Neural Network*)

Induced Local Fields – Activation Potential Κρυφών Νευρώνων

$$v_1 = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 + w_6 x_6$$

$$v_2 = w_1 x_2 + w_2 x_3 + w_3 x_4 + w_4 x_5 + w_5 x_6 + w_6 x_7$$

$$v_3 = w_1 x_4 + w_2 x_5 + w_3 x_6 + w_4 x_7 + w_5 x_8 + w_6 x_9$$

$$v_4 = w_1 x_5 + w_2 x_6 + w_3 x_7 + w_4 x_8 + w_5 x_9 + w_6 x_{10}$$

Ο υπολογισμός των βαρών  $w_i$  απαιτεί διαδικασία Μάθησης που θα προσδιορίσει τελική λύση

# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## Σχεδίαση Νευρωνικών Δικτύων

### Επιλογές Αρχιτεκτονικής

- Στρώματα (layers – input, output, hidden)
- Πρόσθια τροφοδότηση (feedforward)
- Ανάδραση (feedback, recurrent)

### Προσδιορισμός Παραμέτρων, Μάθηση

- Επιβλεπόμενη Μάθηση με Εκπαιδευτή - **Supervised Learning**
  - Με χρήση **Labeled Training Sample Points** (ζεύγη εισόδου – εξόδου) για ρύθμιση της αρχιτεκτονικής με κριτήριο την ελαχιστοποίηση κόστους πρόβλεψης
- Μάθηση χωρίς Εκπαιδευτή
  - Μη Επιβλεπόμενη Μάθηση - **Unsupervised Learning**
  - Ενισχυτική Μάθηση - **Reinforcement Learning**
- Κανονικοποίηση των Δεδομένων Μάθησης - **Normalization of Training Datasets**
  - Feature Scaling, **Min-max Normalization** πριν την εφαρμογή βελτιστοποίησης
- Ανανέωση των Παραμέτρων με βάση το Δείγμα Μάθησης προς αποδοτική Σύγκλιση
  - Είτε **On-line** μετά από είσοδο κάθε δειγματικού στοιχείου μάθησης (π.χ. **Stochastic Gradient Descent**) είτε μετά από είσοδο του συνόλου (**Batch**) ή υποσυνόλων (**Mini-Batches**) του δείγματος μάθησης (π.χ. **Batch Gradient Descent**)
  - Δυνατότητα επαναλήψεων κατά εποχές (**Epochs**) για το σύνολο των δεδομένων μάθησης  
<https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/>

# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## Νευρωνικό Δίκτυο (Learning System)

## Μάθηση με Εκπαιδευτή

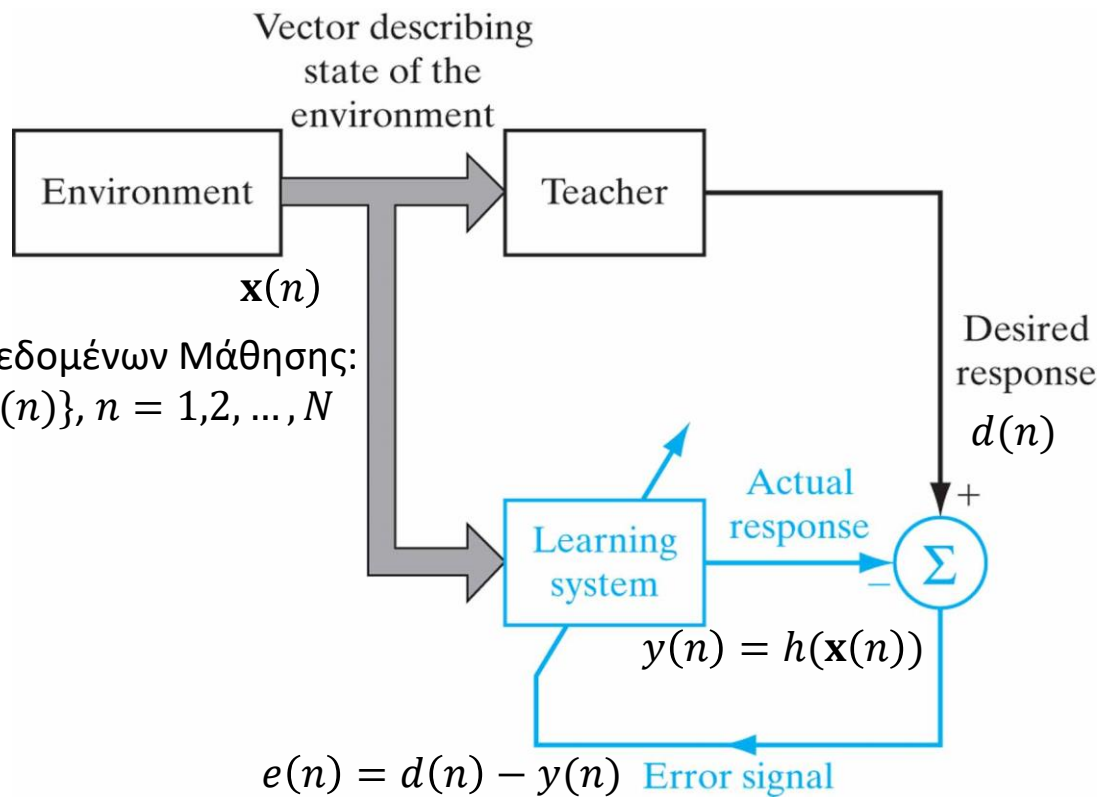
## Supervised Learning

- Εκτίμηση  $y = h(\mathbf{x})$  βαθμωτής (scalar) συνάρτησης της κατάστασης δειγματικών στοιχείων  $\mathbf{x}$  του περιβάλλοντος (**Environment State Vector**), π.χ. ταξινόμηση σε δύο κλάσεις με βάση δυαδική συνάρτησή συνιστωσών (χαρακτηριστικών) του  $\mathbf{x}$

## Φάση Μάθησης

- Δείγμα μάθησης με στοιχεία την κατάσταση του περιβάλλοντος  $\mathbf{x}(n)$  και την αντίστοιχη ζητούμενη έξοδο (desired response, label)  $d(n)$ , γνωστή στον Εκπαιδευτή (**Teacher**)
- Εκτίμηση εξόδου  $y(n) = h(\mathbf{x}(n))$  από το Νευρωνικό Δίκτυο (**Actual Response**), υπολογισμός απόκλισης  $e(n) = d(n) - y(n)$  από **Desired Response**, διόρθωση παραμέτρων συστήματος σε επαναλήψεις για μείωση του σφάλματος  $e(n)$
- Άμεση σχέση με μεθόδους **στατιστικής** εκτίμησης και **βελτιστοποίησης** με επαναληπτικούς αλγορίθμους

Ζεύγη Δεδομένων Μάθησης:  
 $\{\mathbf{x}(n), d(n)\}, n = 1, 2, \dots, N$



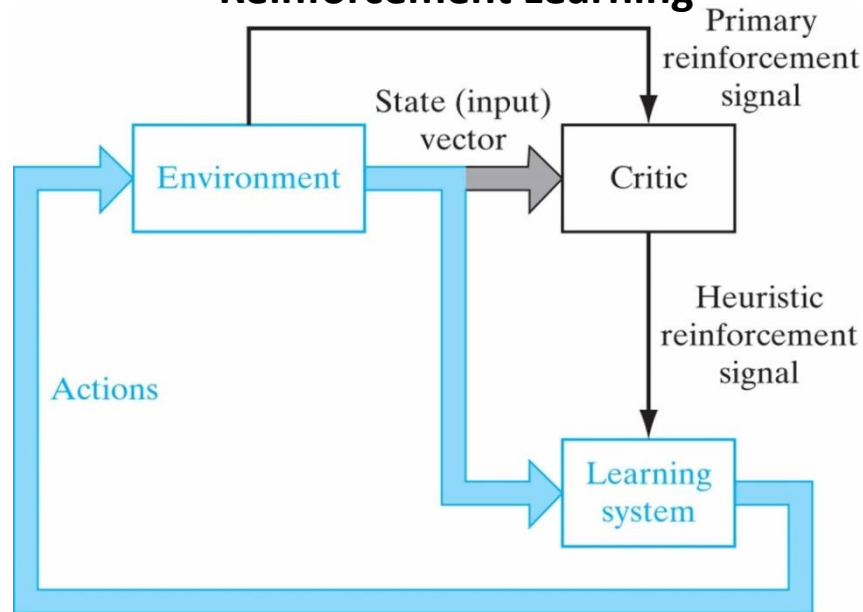
**Το Learning System** ρυθμίζει τις παραμέτρους του  $h(\cdot)$  επαναληπτικά με οδηγό την απόκλιση (σφάλμα) της εξόδου του  $y(n) = h(\mathbf{x}(n))$  από την ζητούμενη απόκριση  $d(n)$ , γνωστή στον Εκπαιδευτή, π.χ. σύμφωνα με μέθοδο **Steepest Descent** προς την κλίση (**gradient**) συναρτήσεως σφάλματος όπως αυτό εξελίσσεται σαν ακολουθία (**Στοχαστική Διαδικασία**) κατά την είσοδο των Δεδομένων Μάθησης  $n = 1, 2, \dots, N$



# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## Μάθηση χωρίς Εκπαιδευτή – Ενισχυτική Μάθηση

### Reinforcement Learning

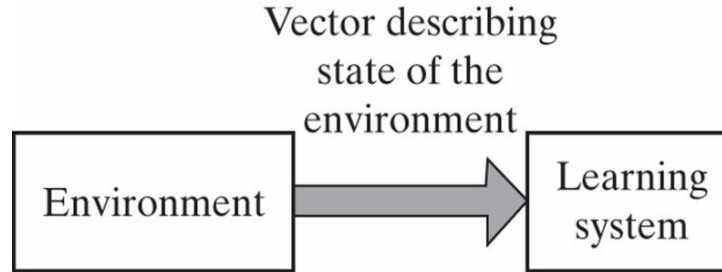


- Το Σύστημα (**Learning System**) μαθαίνει από το περιβάλλον χωρίς **labeled training data** και εκπαιδευτή. Αλλά παίρνει υπόψη του πρόσθετα ενισχυτικά σήματα από το περιβάλλον μέσω εξωτερικού κριτή (**Critic, Agent**) και επιδρά στην εξέλιξη του περιβάλλοντος μέσω ενεργειών (**Actions**)
- Το περιβάλλον υπολογίζει ένα βαθμωτό σήμα ενίσχυσης της απόδοσης του συστήματος (**Scalar Primary Reinforcement Signal**) το οποίο διαβάζει στον κριτή (**Critic, Agent**) μαζί με τη κατάσταση εισόδου (**State Vector**) σε κάθε επανάληψη κατά την εξέλιξη της κατάστασής του
- Ο εξωτερικός κριτής υπολογίζει **πολιτικές** με εκτίμηση κόστους/επιβράβευσης κατά την εξέλιξη της κατάστασης του περιβάλλοντος. Υπολογίζει ένα απλοποιημένο βαθμωτό σήμα ενίσχυσης (**Heuristic Reinforcement Signal**) για την υλοποίηση της πολιτικής και το διαβιβάζει στο Σύστημα
- Το Σύστημα ρυθμίζει τις παραμέτρους του ανάλογα με τη κατάσταση εισόδου και το απλοποιημένο σήμα ενίσχυσης. Μέσω βρόχου ανάδρασης (**Feedback**) προωθεί στο περιβάλλον ενέργειες (**Actions**) που επιδρούν στην εξέλιξη της κατάστασης για την επίτευξη μέσω-μακροπρόθεσμου στόχου (μεγιστοποίηση οφέλους ή ελαχιστοποίηση κόστους)

# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## Μάθηση χωρίς Εκπαιδευτή: Μη Επιβλεπόμενη Μάθηση

### Unsupervised Learning

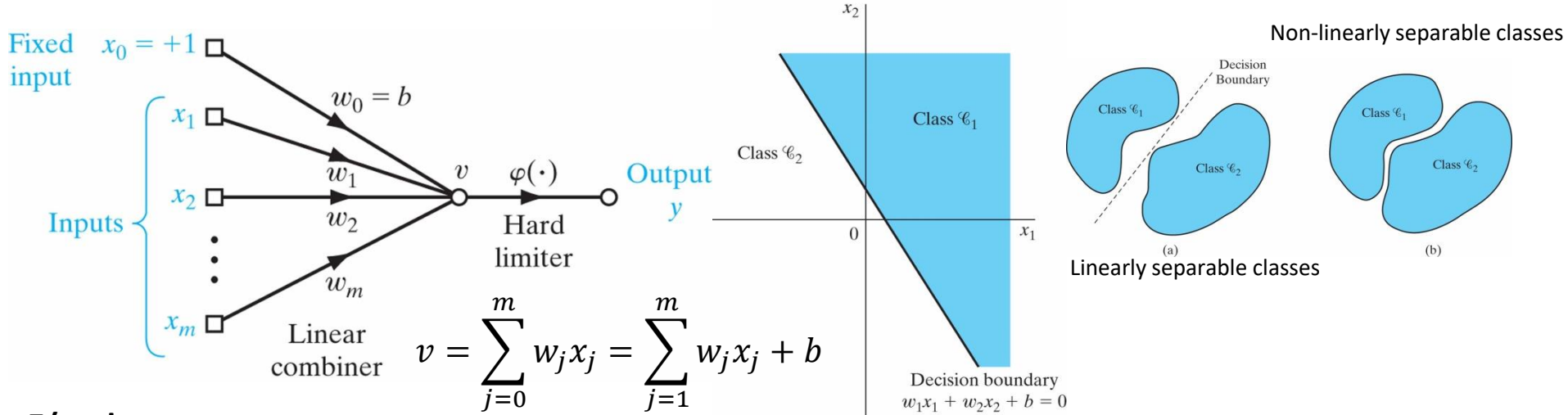


- Το σύστημα αυτορυθμίζεται ανακαλύπτοντας από μόνο του **σημαντικές** στατιστικές διακυμάνσεις του περιβάλλοντος. Υπολογίζει ενδιαφέρουσες στατιστικές δομές (**stochastic features, patterns**) σε μεγάλο όγκο μη χαρακτηρισμένων δεδομένων (**unlabeled datasets**) από τα οποία προκύπτουν μοντέλα, μέθοδοι επεξεργασίας, αποθήκευσης και ταξινόμησης, π.χ. σε ομάδες (**clusters**)
- Με βάση την διαδικασία αυτόνομης μάθησης, το σύστημα μπορεί να **δημιουργήσει** δειγματικά στοιχεία (**generated sample elements**) συμβατά με τις στατιστικές ιδιότητες του περιβάλλοντος. Αποτέλεσμα: Η ταξινόμηση και συμπλήρωση ελλειμματικών περίπλοκων δεδομένων (π.χ. για επεξεργασία εικόνων και αναγνώριση προτύπων)
- Παράδειγμα **unsupervised learning**: Νευρωνικό δίκτυο δυο επιπέδων, επίπεδο εισόδου και κρυφό επίπεδο από νευρώνες που ανταγωνίζονται για την αποθήκευση βασικών χαρακτηριστικών των δειγματικών στοιχείων εισόδου (**Competitive Learning**). Απλή υλοποίηση: Κατά την μάθηση ενεργοποιείται ο νευρώνας με τη μέγιστη τιμή διέγερσης  $v_k$  (**winner-takes-all**)

**Πρόβλημα overfitting αν το σύστημα προσπαθεί να παρακολουθεί μη σημαντικές διακυμάνσεις**

# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## Rosenblatt's Perceptron



### Σύνοψη:

- Ένας νευρώνας με γραμμικό induced local field  $v$  και συνάρτηση ενεργοποίησης  $\varphi(v)$  πρόσημου (**Hard Limiter**) για ταξινόμηση δειγματικών στοιχείων  $\mathbf{x} = [x_0 \ x_1 \ \dots \ x_m]^T$  σε δύο **γραμμικά διαχωριζόμενες** κλάσεις:  $\mathcal{C}_1$  αν  $y = \varphi(v) = 1$ ,  $\mathcal{C}_2$  αν  $y = \varphi(v) = -1$
- Τα βάρη  $\mathbf{w} = [w_0 \ w_1 \ \dots \ w_m]^T$  ρυθμίζονται on-line (stochastic iterative method) με την εφαρμογή **Error-correction Algorithm** σε δειγματικά στοιχεία μάθησης  $\{\mathbf{x}(n), d(n)\}$ ,  $n = 1, 2, \dots, N$  σε περιβάλλον **supervised learning** προς ελαχιστοποίηση σφαλμάτων  $[d(n) - y(n)]$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta[d(n) - y(n)]\mathbf{x}(n)$$

Η παράμετρος  $\eta$ ,  $0 < \eta \leq 1$  (**learning-rate parameter**) αν είναι **μικρή** οδηγεί την επαναληπτική διαδικασία μάθησης σε σύγκλιση. Αν είναι **μεγάλη** μπορεί να επιταχύνει τη σύγκλιση π.χ. σε περιβάλλοντα με μεγάλες αποκλίσεις των δεδομένων  $\mathbf{x}(n)$ , αλλά μπορεί να οδηγήσει σε αστάθειες λόγω ταλαντώσεων περί την βέλτιστη τιμή

Σε περιβάλλον δειγματικών στοιχείων  $\mathbf{x}$  κατανομής Gauss, η ταξινόμησή τους σε δύο κλάσεις  $\mathcal{C}_1$ ,  $\mathcal{C}_2$  μέσω **Bayes Classifier** (ελαχιστοποίηση ρίσκου σφάλματος με βάση a-priori πιθανότητες  $p_1, p_2$ ) ταυτίζεται με το **Rosenblatt Perceptron**

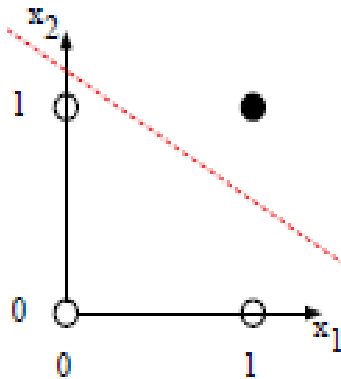
# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## Ταξινόμηση με Perceptron Διαδικιών Εισόδων: Έξοδος Συναρτήσεις AND, OR (XOR ?)

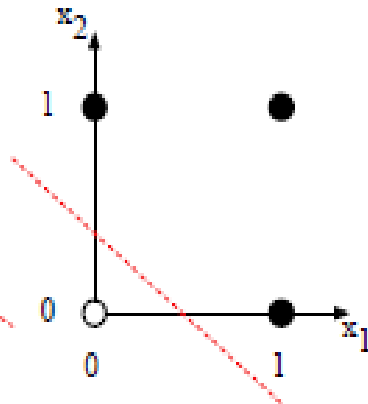
AND			OR			XOR		
x1	x2	y	x1	x2	y	x1	x2	y
0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	0	1	1
1	0	0	1	0	1	1	0	1
1	1	1	1	1	1	1	1	0

Συνάρτηση Διέγερσης:  $y = h(v) = \begin{cases} 1, & v > 0 \\ 0, & v \leq 0 \end{cases}$

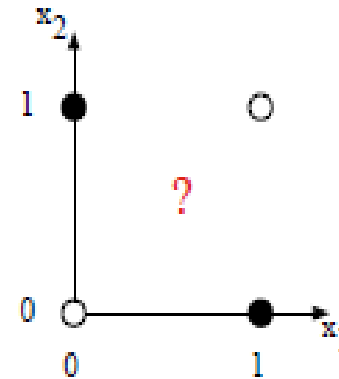
$$y = \begin{cases} 0, & w_1 \cdot x_1 + w_2 \cdot x_2 + b \leq 0 \\ 1, & w_1 \cdot x_1 + w_2 \cdot x_2 + b > 0 \end{cases}$$



a)  $x_1$  AND  $x_2$



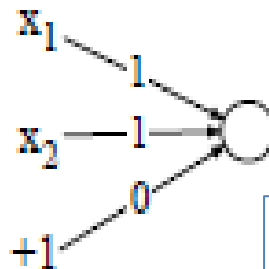
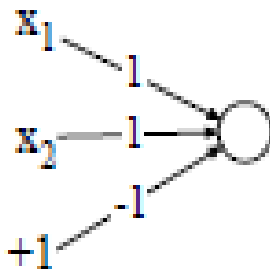
b)  $x_1$  OR  $x_2$



c)  $x_1$  XOR  $x_2$

**Γραμμικές Καμπύλες  
Διαχωρισμού**

**Μη Γραμμική Καμπύλη  
Διαχωρισμού**



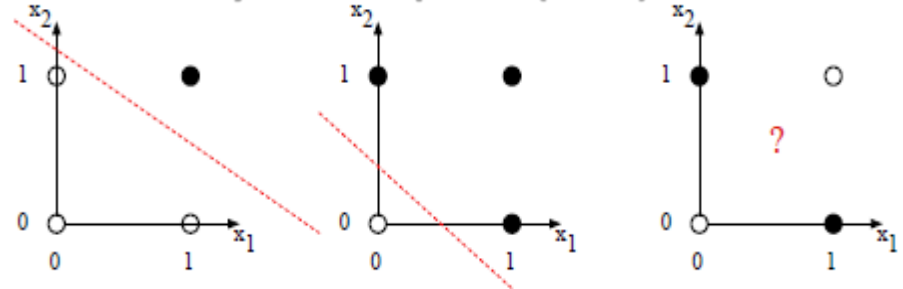
**Αδυναμία υλοποίησης με  
Perceptron ενός επιπέδου**

Βασισμένο στο Daniel Jurafsky, James H. Martin, "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition", Third Edition draft, 2018

# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

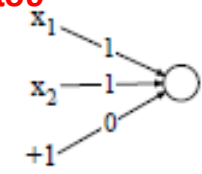
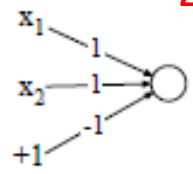
## Ταξινόμηση με Perceptron Διαδικιών Εισόδων: Έξοδος Συνάρτηση XOR Υλοποίηση με Feedforward Multi-Layer Perceptron (MLP)

AND			OR			XOR		
x1	x2	y	x1	x2	y	x1	x2	y
0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	0	1	1
1	0	0	1	0	1	1	0	1
1	1	1	1	1	1	1	1	0



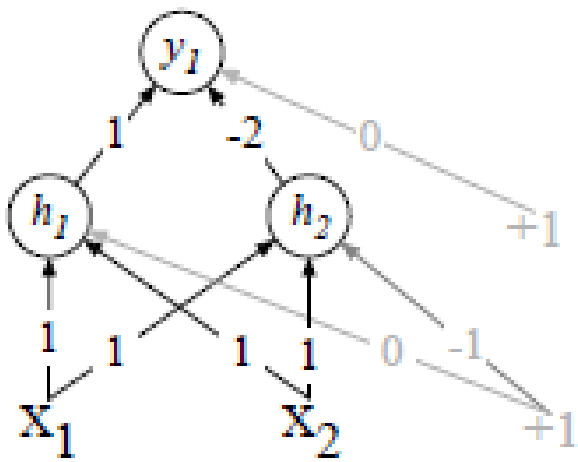
a)  $x_1$  AND  $x_2$       b)  $x_1$  OR  $x_2$       c)  $x_1$  XOR  $x_2$   
**Γραμμικές Καμπύλες Διαχωρισμού**      **Μη Γραμμική Καμπύλη Διαχωρισμού**

Συνάρτηση Διέγερσης:  $y = h(v) = \begin{cases} 1, & v > 0 \\ 0, & v \leq 0 \end{cases}$   
 $y = \begin{cases} 0, & w_1 \cdot x_1 + w_2 \cdot x_2 + b \leq 0 \\ 1, & w_1 \cdot x_1 + w_2 \cdot x_2 + b > 0 \end{cases}$



**Αδυναμία υλοποίησης με Perceptron ενός επιπέδου**

Υλοποίηση XOR: Κρυφό Επίπεδο Νευρώνων  $h_1, h_2 \rightarrow$  **Multi-Layer Perceptron, MLP**



Έξοδος Νευρώνα  $h_1$ :  $a_1 = h(w_{11} \cdot x_1 + w_{21} \cdot x_2 + b_1)$   
 Έξοδος Νευρώνα  $h_2$ :  $a_2 = h(w_{12} \cdot x_1 + w_{22} \cdot x_2 + b_2)$   
 Έξοδος Νευρώνα  $y_1$ :  $y = h(w_{13} \cdot a_1 + w_{23} \cdot a_2 + b_3)$   
 $w_{11} = w_{12} = w_{21} = w_{22} = 1, b_1 = 0, b_2 = -1$   
 $w_{13} = 1, w_{23} = -2, b_3 = 0$

$x_1$ :	0	0	1	1
$x_2$ :	0	1	0	1
$a_1$ :	0	1	1	0
$a_2$ :	0	0	1	0
$y$ :	0	1	1	0

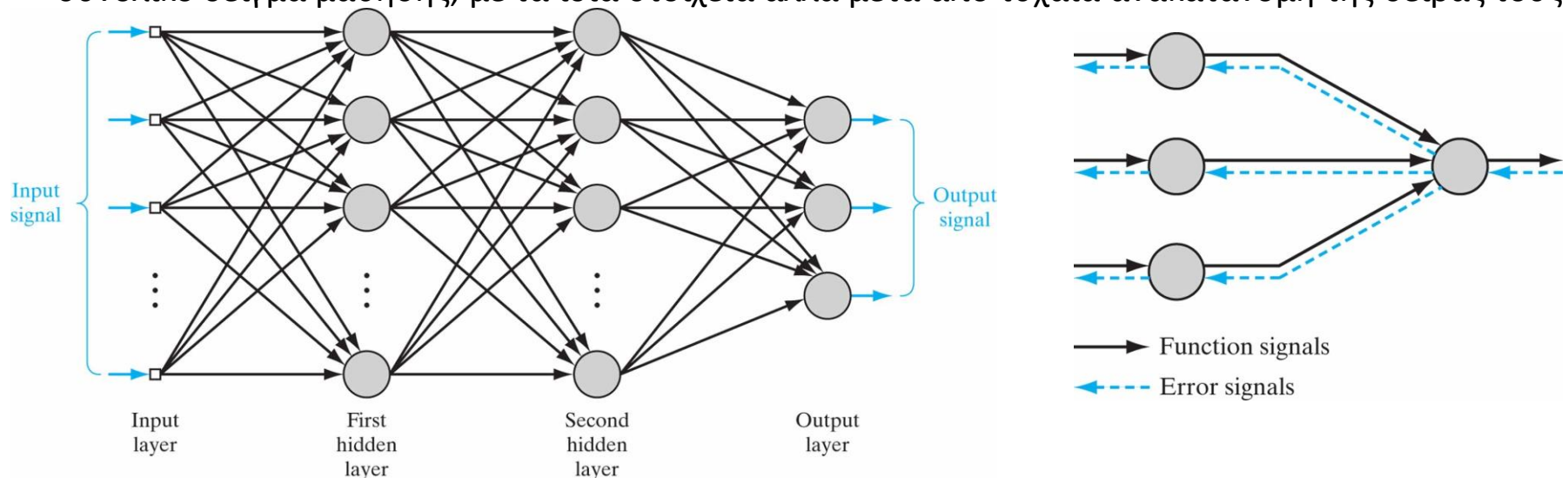
Βασισμένο στο Daniel Jurafsky, James H. Martin, "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition", Third Edition draft, 2018

# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## Multilayer Perceptrons , Back-Propagation Algorithm (1/3)

Χαρακτηριστικά:

1. Διαφορίσιμες μη γραμμικές συναρτήσεις ενεργοποίησης  $\varphi_j(v_j)$  του νευρώνα  $j$  (π.χ. Logistic Function)
2. Το δίκτυο περιλαμβάνει κρυφά επίπεδα (**hidden layers**) νευρώνων με μεγάλη συνδεσιμότητα και συναπτικά βάρη  $w_{ji}$  (από  $i \rightarrow j$ ). Οι κρυφοί νευρώνες ενισχύουν ειδοποιά χαρακτηριστικά του δείγματος εισόδου μέσω διαδικασίας επιβλεπόμενης μάθησης
3. Τα βάρη ρυθμίζονται on-line (stochastic iterative method) με την εφαρμογή **Back-Propagation Algorithm** διαδοχικά σε κάθε στοιχείο του **labeled** δείγματος μάθησης (**παράδειγμα μάθησης**)  $\{\mathbf{x}(n), d_j(n)\}, n = 1, 2, \dots, N$  σε περιβάλλον **supervised learning** με δύο φάσεις:
  - i. **Forward Phase:** Η είσοδος  $\mathbf{x}(n)$  του παραδείγματος μάθησης  $n$  διαπερνά το δίκτυο μέσω **Function Signals** με βάρη  $w_{ji}(n)$  όπως έχουν προσδιοριστεί μέχρι την εφαρμογή του παρόντος παραδείγματος και προκύπτει η έξοδος  $y_j(n)$  του νευρώνα  $j$
  - ii. **Backward Phase:** Οι αποκλίσεις  $e_j(n) = d_j(n) - y_j(n)$  διαπερνούν το δίκτυο στην αντίστροφη πορεία σαν **Error Signals** και διορθώνουν τα συναπτικά βάρη
4. Η τελική σύγκλιση ολοκληρώνεται σε πολλαπλές **εποχές** με επαναλήψεις των δύο φάσεων για το συνολικό δείγμα μάθησης, με τα ίδια στοιχεία αλλά μετά από τυχαία ανακατανομή της σειράς τους



# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## Multilayer Perceptrons , Back-Propagation Algorithm (2/3)

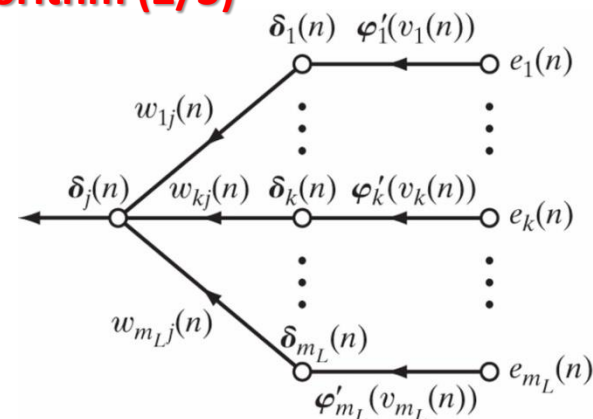
### Σύγκληση κατά Least Mean Square – LMS ανά Εποχή:

Άθροισμα των τετραγώνων των αποκλίσεων  $e_j(n) = d_j(n) - y_j(n)$  στο σύνολο των νευρώνων  $j$ ,  $\mathcal{E}(n) = \frac{1}{2} \sum_j e_j^2(n)$  όπως ορίζονται από το παράδειγμα μάθησης  $\{\mathbf{x}(n), d_j(n)\}$ , και από τα οποία προκύπτει η μέση τιμή  $\mathcal{E}_{AVG}(N)$  για όλο το δείγμα μάθησης  $n = 1, 2, \dots, N$ :

$$\mathcal{E}_{AVG}(N) = \frac{1}{N} \sum_{n=1}^N \mathcal{E}(n) = \frac{1}{N} \sum_{n=1}^N \left\{ \frac{1}{2} \sum_j e_j^2(n) \right\}$$

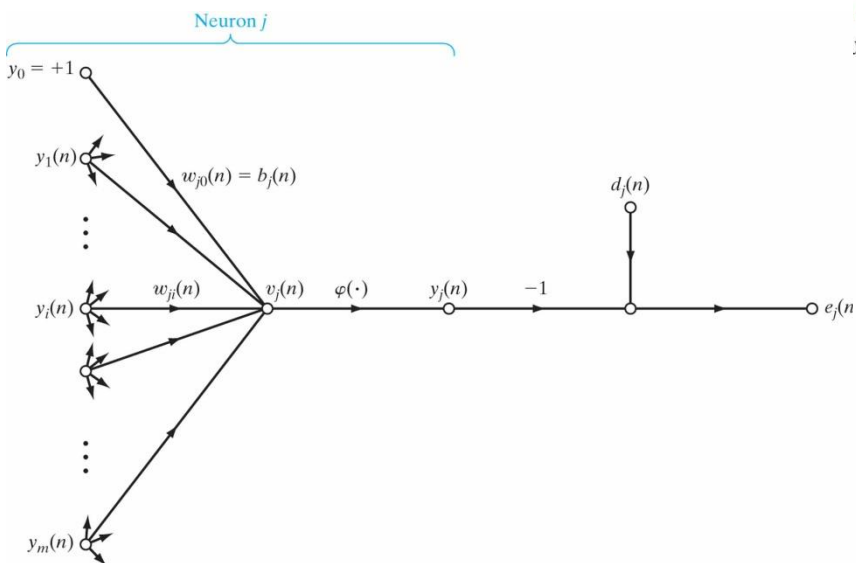
Οι επαναληπτικές διορθώσεις  $\Delta w_{ji}(n)$  στα βάρη  $w_{ji}(n)$  οδηγούν προς την ελαχιστοποίηση του  $\mathcal{E}(n)$  στη κατεύθυνση της **local gradient**

$\delta_j(n) = \frac{\partial \mathcal{E}(n)}{\partial v_j(n)}$  όπου  $v_j(n) = \sum_{i=0}^m w_{ji}(n) y_i(n)$  (**Induced Local Field** του νευρώνα  $j$ ):  $\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n)$  με βήμα την **Learning Parameter**  $\eta$

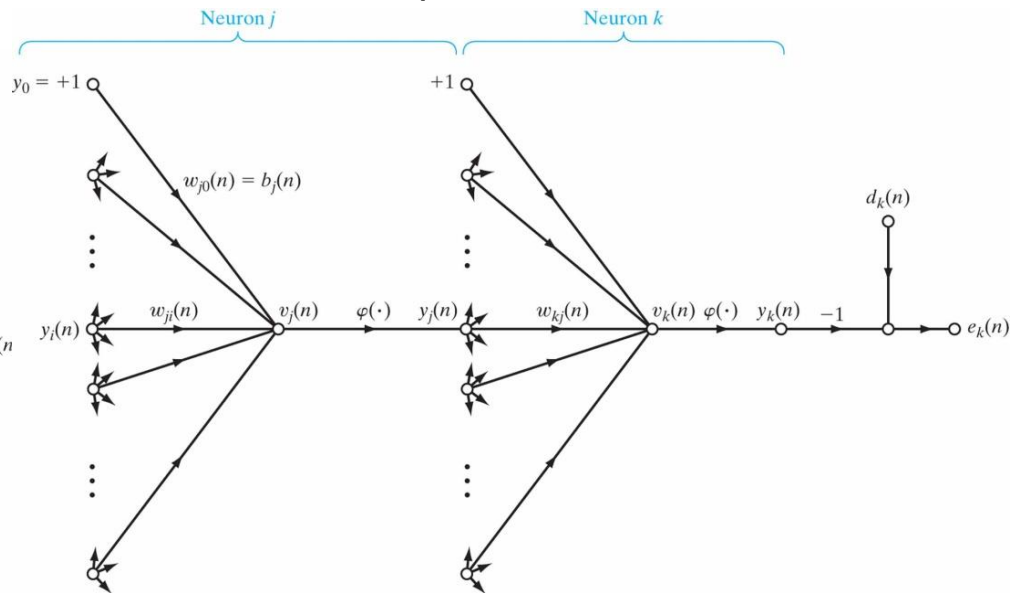


$$\delta_j'(v_j(n)) = \frac{\partial y_j(n)}{\partial v_j(n)}$$

Ροή σήματος για ανάστροφη διάδοση σφαλμάτων



Ροή σήματος σε νευρώνα εξόδου  $j$



Ροή σήματος νευρώνα εξόδου  $k$  συνδεδεμένου σε κρυφό νευρώνα  $j$

# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## Multilayer Perceptrons , Back-Propagation Algorithm (3/3)

Για το παράδειγμα μάθησης  $\{\mathbf{x}(n), d_j(n)\}, n = 1, 2, \dots, N$  με αποκρίσεις  $d_j(n)$  του νευρώνα  $j$ :

1. Αρχικοποίηση των βαρών
2. Για κάθε παράδειγμα μάθησης  $n$  υλοποιούνται οι υπολογισμοί των φάσεων forward & backward
3. Υπολογισμοί φάσης **forward**:

Για κάθε νευρώνα  $j$  και κάθε layer  $l = 1, 2, \dots, L$  το **induced local field** είναι

$v_j^{(l)}(n) = \sum_i w_{ji}^{(l)}(n) y_i^{(l-1)}(n)$ . Για  $i = 0$ ,  $y_0^{(l-1)} = +1$ ,  $w_{j0}^{(l)} = b_j^{(l)}(n)$  οι τιμές bias του συγκεκριμένου νευρώνα  $j$

Η **έξοδος** του κάθε νευρώνα είναι  $y_j^{(l)}(n) = \varphi_j(v_j^{(l)}(n))$

Για το πρώτο κρυφό επίπεδο  $y_j^{(0)}(n) = x_j(n)$

Αν το  $j$  ανήκει στο επίπεδο εξόδου  $y_j^{(L)}(n) = o_j(n)$  όπου  $o_j(n)$  είναι η τελική έξοδος  $j$  του Multilayer Perceptron. Το **σήμα σφάλματος** είναι  $e_j(n) = d_j(n) - o_j(n)$

4. Υπολογισμοί φάσης **backward**:

Υπολογισμός τοπικών μερικών παραγώγων

$$\delta_j^{(l)}(n) = \begin{cases} e_j^{(L)}(n) \varphi_j' \left( v_j^{(L)}(n) \right) & \text{αν το } j \text{ ανήκει στο επίπεδο εξόδου} \\ \varphi_j' \left( v_j^{(l)}(n) \right) \sum_k \delta_k^{(l+1)}(n) w_{ki}^{(l+1)} & \text{αν το } j \text{ ανήκει σε κρυφό επίπεδο} \end{cases}$$

Τα βάρη ρυθμίζονται με τον επαναληπτικό κανόνα με βήμα την **Learning Parameter**  $\eta$ :

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha \times w_{ji}^{(l)}(n-1) + \eta \times \delta_j^{(l)}(n) y_i^{(l-1)}(n)$$

Η σταθερά  $\alpha \geq 0$  (**momentum constant**) βοηθά στη σταθεροποίηση της σύγκλισης με συμμετοχή τιμών που προέκυψαν από προηγούμενα παραδείγματα μάθησης



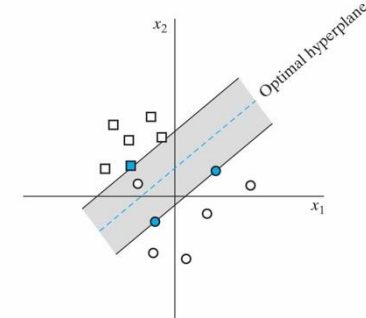
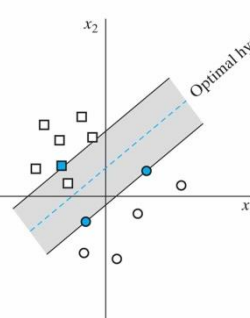
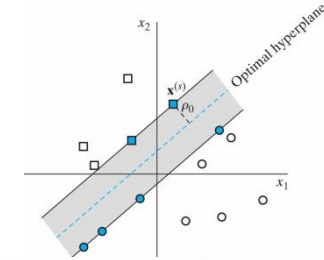
# ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ & ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

## Αναφορά σε Ειδικές Μηχανές Μάθησης - Νευρωνικά Δίκτυα

**Convolutional Neural Networks (CNN):** Εδική κατηγορία *Multilayer Perceptron* για αποδοτική κατηγοριοποίηση δυσδιάστατων δειγμάτων (π.χ. *pattern recognition* εικόνων) κυρίως με μάθηση μέσω δασκάλου (*supervised learning*). Η απλοποίηση προκύπτει με αποσύνδεση του δικτύου σε χαλαρά συνδεδεμένα επίπεδα (*layers*), κοινά χαρακτηριστικά (*receptive fields*) και συνελκτικά αθροίσματα για συναρτήσεις διέγερσης (*convolutional induced local fields*)

**K- Means Clustering:** Οργανώνει σε  $K$  συστάδες (*clusters*) παρατηρήσεις  $\mathbf{x}_i$  με βάση κοινά χαρακτηριστικά χωρίς δάσκαλο (*unsupervised learning*), π.χ. οργάνωση συστάδων με βάση την *Ευκλείδεια απόσταση*  $\|\mathbf{x}_i - \mathbf{x}_j\|^2$

**Support Vector Machines (SVM):** Ταξινόμηση δεδομένων μέσω δύο περιοχών μέγιστης γραμμικής διάκρισης με *supervised learning*. Τα όρια των περιοχών αυτών ορίζονται από σημεία υποστήριξης (*support vectors*) όπως στο δυσδιάστατο σχήμα με τα μπλε τετράγωνα και κύκλους. Αν δεν υπάρχει όριο γραμμικής διάκρισης περιοχών (*non-separable patterns*), ζητείται μηχανή που προκύπτει από τα δείγματα μάθησης με το ελάχιστο σφάλμα



**Self-Organizing Maps (SOM):** Βασίζεται σε *competitive unsupervised learning* που μειώνει τους ενεργοποιημένους νευρώνες όσο πλησιάζουν στο επίπεδο εξόδου. Οι νευρώνες τοποθετούνται στους κόμβους ενός δυσδιάστατου πλέγματος (*lattice*) και οργανώνονται σε τοπογραφικούς χάρτες μέσω παραδειγμάτων μάθησης που πυροδοτούν διαδρομές θετικής και αρνητικής ανάδρασης ώστε να προκύψει *τελικός νικητής* (*winner takes all*)