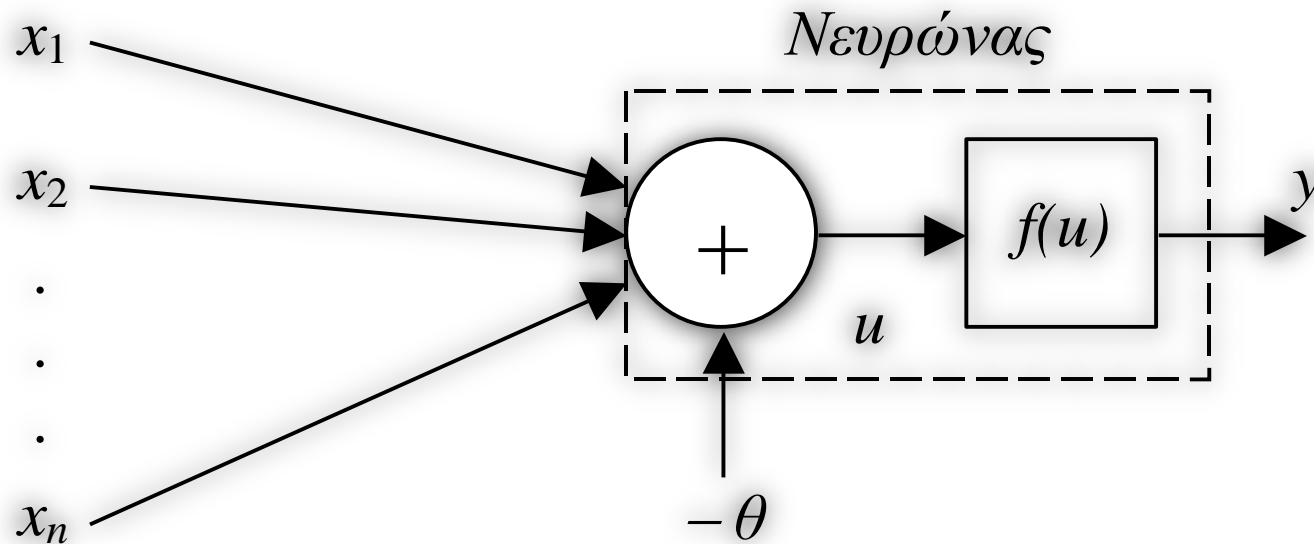


# Τα Δίκτυα Perceptron και ADALINE

Κώστας Διαμαντάρας  
Τμήμα Πληροφορικής  
ΤΕΙ Θεσσαλονίκης

# Ο Τεχνητός Νευρώνας

- Το μοντέλο McCulloch-Pitts



# Το μοντέλο Perceptron

- Ένας μόνο νευρώνας McCulloch-Pitts
- Εκπαίδευση με επίβλεψη δηλ. με στόχους
- Ανάκληση: Είσοδοι  $\rightarrow$  Έξοδος

$$u = w_1x_1 + w_2x_2 + \dots + w_nx_n - \theta$$

$$y = f(u)$$

- Συνάρτηση ενεργοποίησης:

$$f = \text{βηματική (είτε 0/1 είτε -1/1)}$$

# Κατώφλι = συναπτικό βάρος

- Αν θέσουμε

$$w_0 = \theta, \quad x_0 = -1$$

- Τότε

$$u = w_1 x_1 + w_2 x_2 + \dots + w_n x_n + w_0 x_0$$

ή

$$u = \mathbf{w}^T \mathbf{x}$$

# Κατώφλι = συναπτικό βάρος (2)

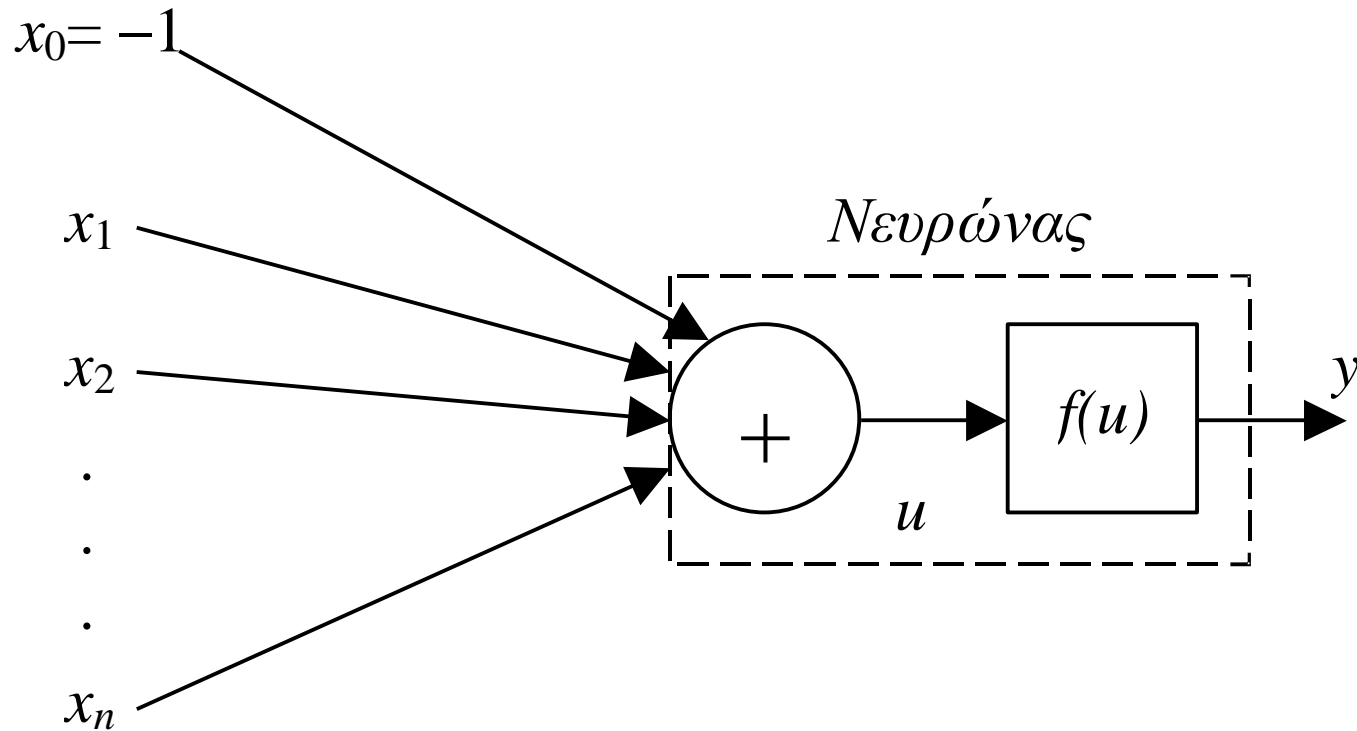
Επαυξημένο διάνυσμα βαρών

$$\mathbf{w} = [w_0, w_1, w_2, \dots, w_n]^T$$

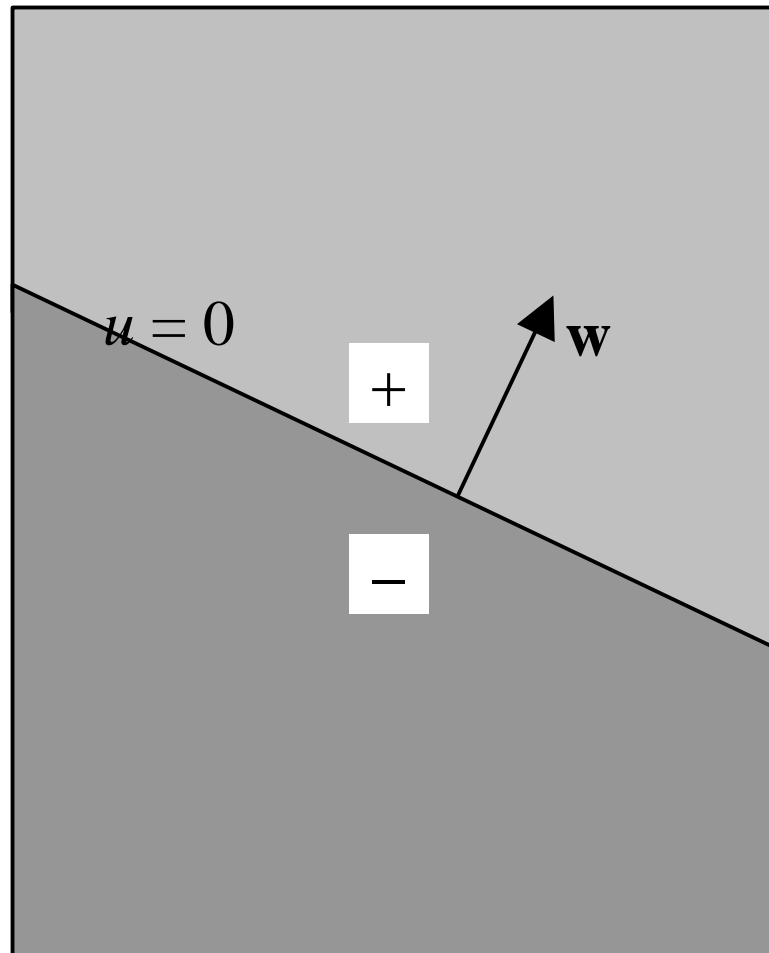
Επαυξημένο διάνυσμα εισόδου

$$\mathbf{x} = [-1, x_1, x_2, \dots, x_n]^T$$

# Κατώφλι = συναπτικό βάρος (3)



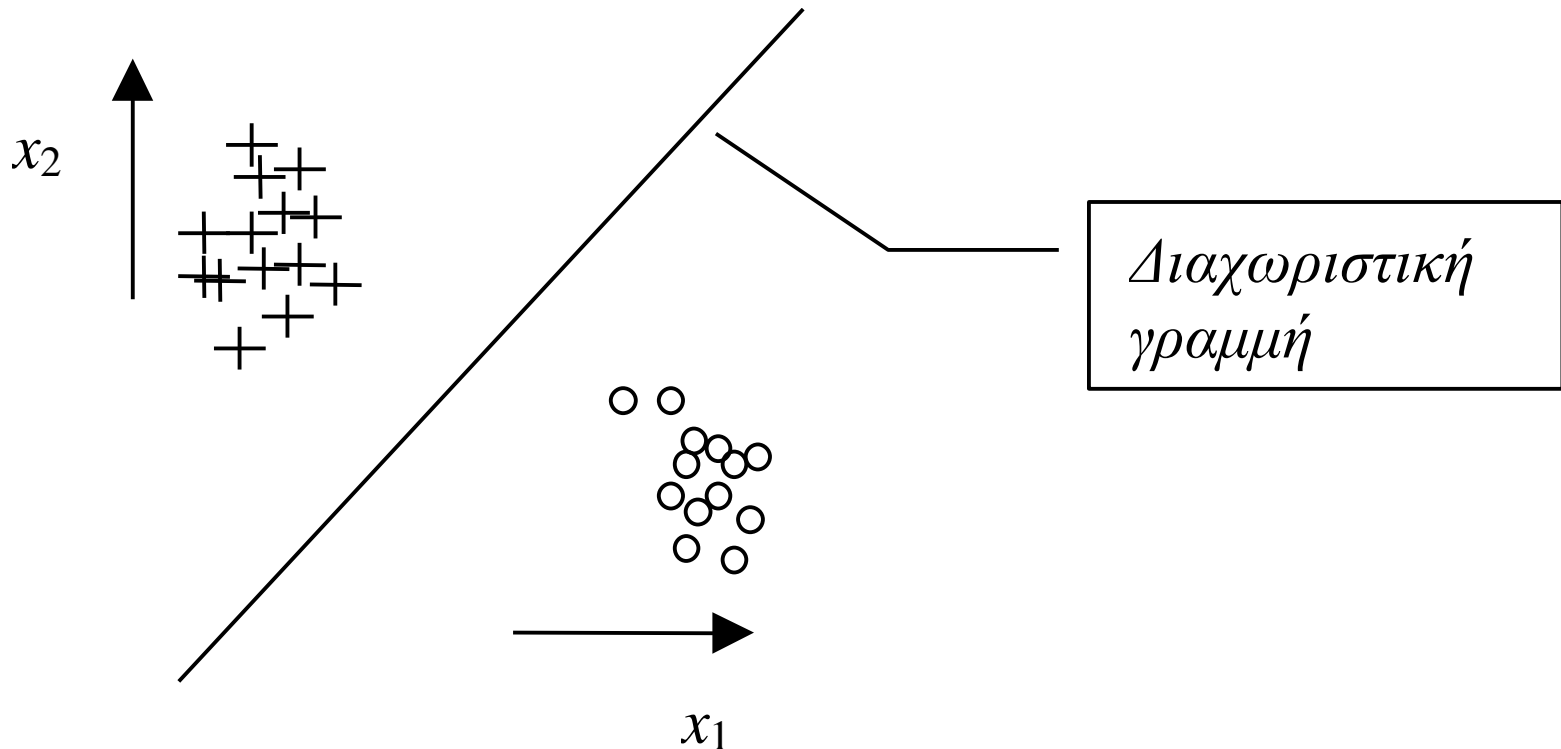
# Δυνατότητες Perceptron



# Γραμμικά διαχωρίσιμο πρόβλημα

ο = άλογα

+ = γαϊδούρια



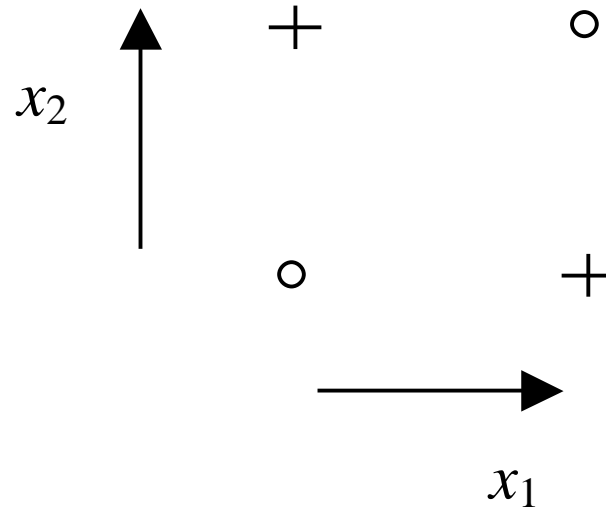


# Μη γραμμικά διαχωρίσιμο πρόβλημα

- Πχ. XOR:

ο = κλάση 0

+ = κλάση 1



# Εκπαίδευση Perceptron

- $P$  Επαυξημένα Πρότυπα Εισόδου  
 $\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(P)$
- Στόχοι (1 για κάθε πρότυπο εισόδου)  
 $d(1), d(2), \dots, d(P)$
- Έξοδος (διαφορετική για κάθε πρότυπο)  
 $y(1), y(2), \dots, y(P)$

# Κανόνας Εκπαίδευσης Perceptron

Εισήγαγε τα πρότυπα με τη σειρά. Όταν τελειώσουν ξανάρχισε πάλι από την αρχή.

*Εποχή* = μια κυκλική επανάληψη όλων των προτύπων

Για κάθε πρότυπο  $k$

$$y(k) = f(\mathbf{w}(k)^T \mathbf{x}(k))$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \beta (d(k) - y(k)) \mathbf{x}(k)$$

- *Αν στόχος = έξοδος τότε δεν γίνεται καμία διόρθωση. Διόρθωση μόνο σε περίπτωση λάθους*

# Τερματισμός αλγορίθμου

- Ο αλγόριθμος τερματίζεται όταν δεν γίνεται πλέον καμία διόρθωση σε κανένα πρότυπο. Αυτό σημαίνει ότι ΟΛΟΙ οι στόχοι είναι ίσοι με ΟΛΕΣ τις εξόδους

$$d(1) = y(1)$$

$$d(2) = y(2)$$

...

$$d(P) = y(P)$$

# Η παράμετρος $\beta =$ βήμα εκπαίδευσης

$\beta$  = βήμα εκπαίδευσης  
= θετική (μικρή) σταθερά

Διόρθωση βαρών ανάλογη του  $\beta$ .

Μεγάλο  $\beta \rightarrow$  κίνδυνος ταλάντωσης

Μικρό  $\beta \rightarrow$  αργή σύγκλιση

# Ιδιότητες κανόνα Perceptron

- Αν το πρόβλημα δεν είναι γραμμικά διαχωρίσιμο το Perceptron δεν συγκλίνει ποτέ!
- Θεώρημα:  
Αν το πρόβλημα είναι γραμμικά διαχωρίσιμο τότε συγκλίνει σε πεπερασμένο (αλλά άγνωστο) αριθμό επαναλήψεων

# Το δίκτυο ADALINE

- *Ομοιότητα με Perceptron:*
  1. Ένας μόνο νευρώνας McCulloch-Pitts
  2. Έπαρξη στόχων (εκπαίδευση με επίβλεψη)
- *Διαφορά με Perceptron:*
  1. Οι στόχοι συγκρίνονται με την διέγερση  $u$  και όχι με την έξοδο  $y$
  2. Κριτήριο μέσο τετραγωνικό σφάλμα

# Μέσο Τετραγωνικό Σφάλμα

Mean Square Error (MSE)

$$J_{MSE} = E\{(u - d)^2\}$$

$E$  = Μέση τιμή

$$u = \mathbf{w}^T \mathbf{x}$$

$d$  = στόχος

Πρακτικά:

$$J_{MSE} \sim 1/P \sum_k (u(k) - d(k))^2$$



# Κανόνας ADALINE (Widrow-Hoff ή delta rule)

Για κάθε πρότυπο  $k$

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \beta(k) (d(k) - u(k)) \mathbf{x}(k)$$

- *Διόρθωση σε κάθε περίπτωση*
- Τερματισμός όταν  $J < \text{κάποιο όριο}$
- Πρόβλημα ορισμού στόχων
  - Στο *Perceptron* στόχοι σαφείς (0/1 ή -1/1)
  - Στο *ADALINE*  $-\infty < u < \infty$ ,  $d = ?$  Συνήθως βάζουμε  $d=1$  αν το πρότυπο ανήκει στην κλάση 1 ή  $d = -1$  αν το πρότυπο ανήκει στην κλάση 0.

# Παράδειγμα ADALINE (1)

## Γραμμικά Διαχωρίσιμο

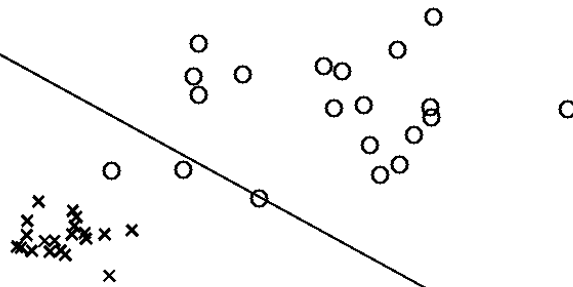
Το ADALINE διαχωρίζει τις κλάσεις με επιτυχία



# Παράδειγμα ADALINE (2)

## Γραμμικά Διαχωρίσιμο

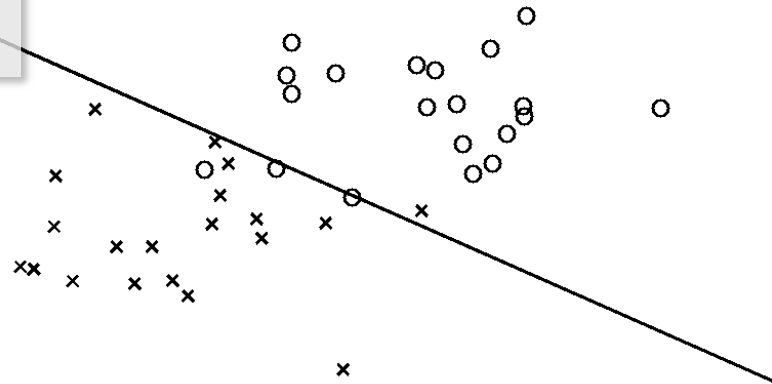
Το ADALINE δεν διαχωρίζει τις κλάσεις σωστά, αν και είναι γραμμικά διαχωρίσιμες!



# Παράδειγμα ADALINE (3)

## Μη Γραμμικά Διαχωρίσιμο

Το ADALINE κάνει καλή δουλειά, αν και είναι μη γραμμικά διαχωρίσιμες



# Σύγκριση Perceptron - ADALINE

- Και οι δύο κανόνες είναι αυτοπροσαρμοστικοί.
- Πλεονέκτημα ADALINE: συγκλίνει ακόμη κι αν το πρόβλημα δεν είναι γραμμικά διαχωρίσιμο. Στην περίπτωση αυτή το Perceptron ταλαντεύεται επ' άπειρον.
- Μειονέκτημα ADALINE: δεν εγγυάται το διαχωρισμό των κλάσεων όταν το πρόβλημα είναι γραμμικά διαχωρίσιμο. Στον αλγόριθμο Perceptron τέτοιο πρόβλημα δεν υφίσταται.