Το Δίκτυο Multi-Layer Perceptron και ο Κανόνας Back-Propagation

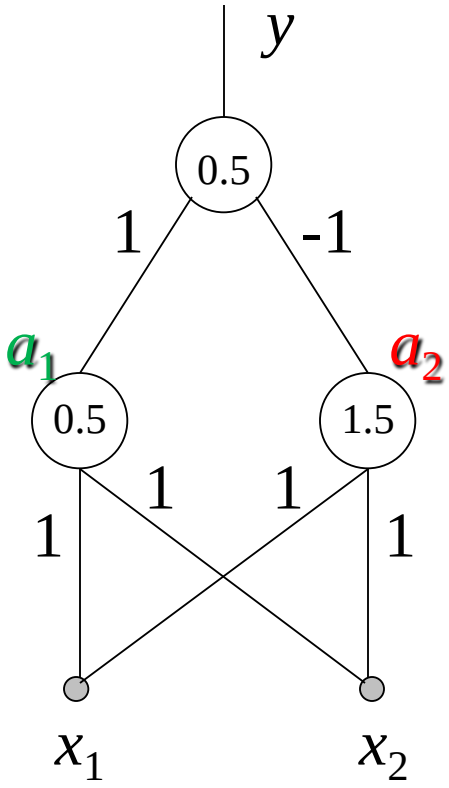
Κώστας Διαμαντάρας
Τμήμα Πληροφορικής
ΤΕΙ Θεσσαλονίκης

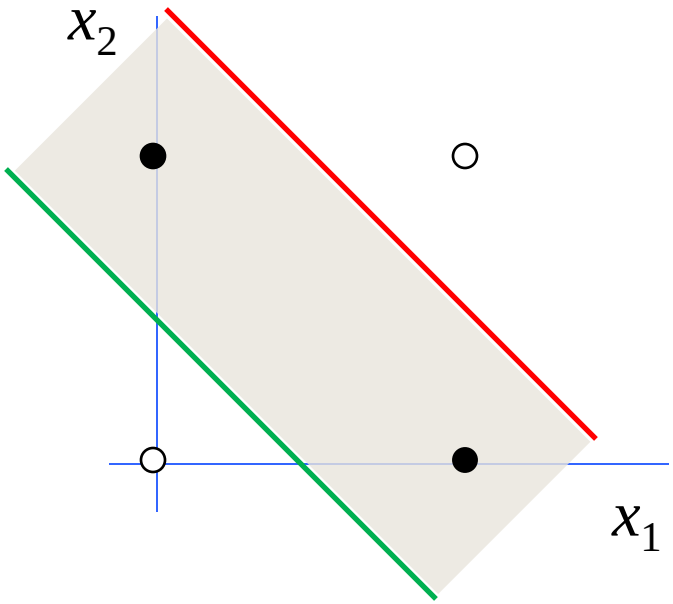
Το Πρόβλημα XOR

- Περιορισμένες δυνατότητες Perceptron (=1 νευρώνας). Πχ. Αδυναμία λύσης απλού προβλήματος XOR:

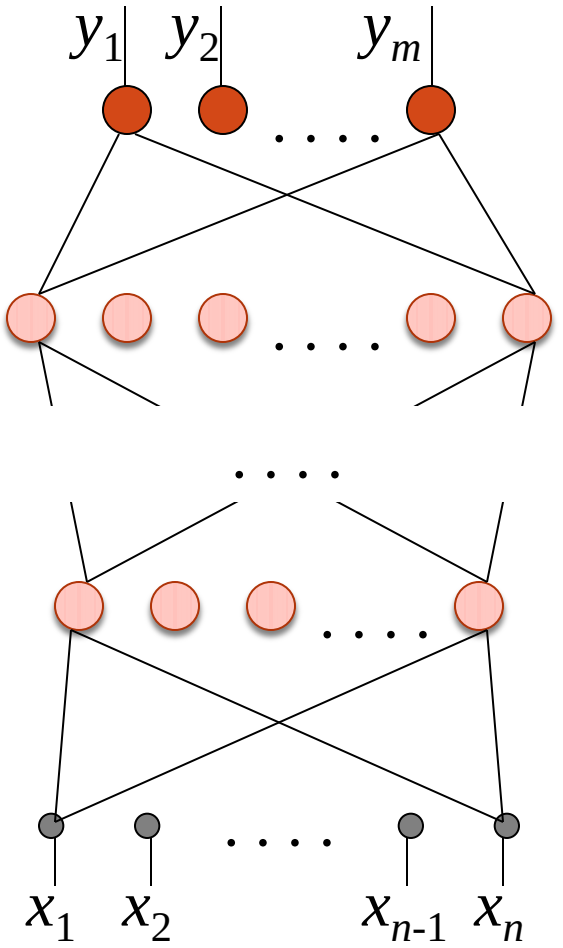
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Λύση του XOR με MLP





Γενική Τοπολογία Δικτύου MLP



Στρώμα Εξόδου

Κρυφό Στρώμα N

. . . .

Κρυφό Στρώμα 1

Στρώμα Εισόδου

Συνάρτηση Ενεργοποίησης

Δυαδική

Μη παραγωγίσιμη

$f(u)=1/(1+e^{-u})$

Συνεχείς τιμές

Παραγωγίσιμη

Δυνατότητες δικτύων MLP (1)

- **Θεώρημα**

Έστω f = σιγμοειδής και $g(x_1, x_2, \ldots, x_p)$ *οποιαδήποτε συνεχής συνάρτηση p μεταβλητών ορισμένη στον μοναδιαίο κύβο* $I_p = [0,1]^p$. *Τότε υπάρχει ακέραιος M και κάποιες τιμές των παραμέτρων* α_i, w_{ij}, ϑ_i, *έτσι ώστε η συνάρτηση*

$$\phi(x_1, x_2, \cdots, x_p) = \sum_{i=1}^{M} \alpha_i f\left(\sum_{j=1}^{p} w_{ij} x_j + \theta_i\right)$$

προσεγγίζει την $g(x_1, x_2, \ldots, x_p)$ *με σφάλμα μικρότερο του* ε για όλες τις τιμές $\{x_1, x_2, \ldots, x_p\} \in I_p$ *και για οποιοδήποτε* $\varepsilon > 0$.

Δυνατότητες Δικτύων MLP (2)

Με απλά λόγια: (Ιδιότητα *Universal Approximator*)

- Ένα δίκτυο δύο στρωμάτων μπορεί να προσεγγίσει όσο καλά επιθυμούμε οποιαδήποτε συνεχή συνάρτηση

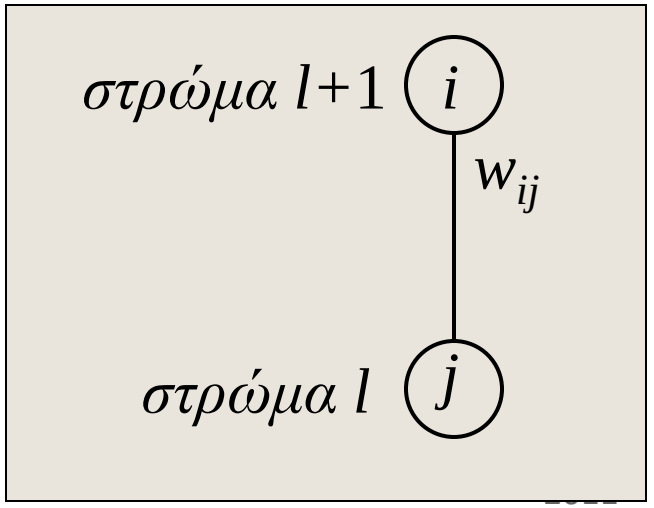
  - Οι νευρώνες του κρυφού στρώματος έχουν τη σιγμοειδή συνάρτηση ενεργοποίησης
  - Ο νευρώνας εξόδου έχει τη γραμμική συνάρτηση ενεργοποίησης

Δυνατότητες Δικτύων MLP (3)

- Όσο πιο πολύπλοκη είναι η συνάρτηση που επιθυμούμε να προσεγγίσουμε τόσο περισσότερους κρυφούς νευρώνες θέλουμε
- Δύο στρώματα αρκούν
- Universal Approximator : ισχύει και για συναρτήσεις πολλών εξόδων

Συμβολισμοί

- $N(l)$: πλήθος νευρώνων στο στρώμα l
- $a_i(l)$: έξοδος του νευρώνα i στρώμα l
- x_i : είσοδοι του δικτύου (μηδενικό στρώμα)
- y_i : έξοδοι του τελευταίου στρώματος
- w_{ij} : συναπτικό βάρος από νευρώνα j σε i



Ανάκληση

Είσοδοι: $a_1(0) = x_1, a_2(0) = x_2, \ldots, a_{N(0)}(0) = x_{N(0)}$

/* 0 = στρώμα εισόδου, L = στρώμα εξόδου */

Έξοδοι: $y_1 = a_1(L), y_2 = a_2(L), \ldots, y_{N(L)} = a_{N(L)}(L)$

Για κάθε στρώμα $l = 1, \ldots, L$ {

Για κάθε νευρώνα $i = 1, \ldots, N(l)$ {

$$a_i(l) = f\left(\sum_{j=1}^{N(l-1)} w_{ij}(l)a_j(l-1) + w_{i0}(l)\right)$$

}

}

Εκπαίδευση

Κανόνας Back-Propagation

- Μάθηση με επίβλεψη
- $p = 1, \ldots, P$ ζεύγη {εισόδων/στόχων}
- Διάνυσμα στόχων $\mathbf{d}^{(p)} = [d_1^{(p)}, \cdots, d_m^{(p)}]^T$
- Διάνυσμα εισόδου $\mathbf{x}^{(p)} = [x_1^{(p)}, \cdots, x_n^{(p)}]^T$
- Διάνυσμα εξόδου $\mathbf{y}^{(p)} = [y_1^{(p)}, \cdots, y_m^{(p)}]^T$

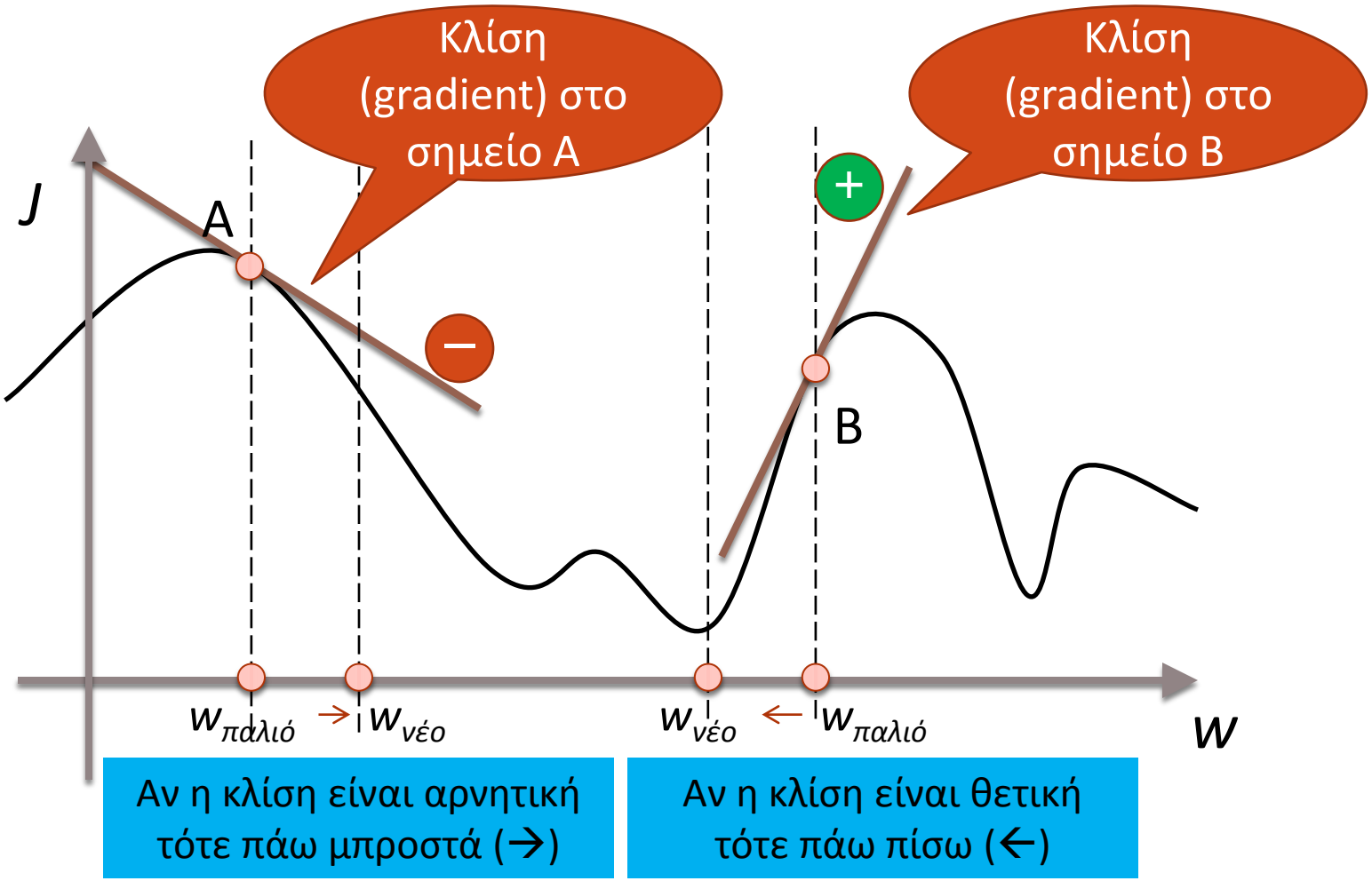
Back-propagation

- Κριτήριο μάθησης: ελαχιστοποίηση του μέσου τετραγωνικού σφάλματος για όλα τα πρότυπα (P το πλήθος)

$$J = \frac{1}{P}\sum_{p=1}^{P}\left\|\mathbf{d}^{(p)} - \mathbf{y}^{(p)}\right\|^2 = \frac{1}{P}\sum_{p=1}^{P}\sum_{i=1}^{m}[d_i^{(p)} - y_i^{(p)}]^2$$

- Μέθοδος βελτιστοποίησης: Κατάβαση Δυναμικού

Κατάβαση Δυναμικού (1)



Κατάβαση Δυναμικού (2)

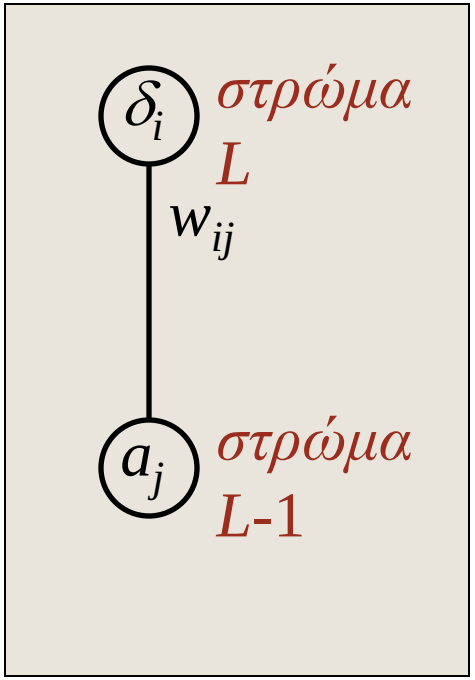
- Κινούμαι αντίθετα απ' ότι λέει η παράγωγος: αν η παράγωγος είναι θετική τότε μειώνω το *w*, αν η παράγωγος είναι αρνητική τότε αυξάνω το *w*.
- Κάνω μικρά βήματα χρησιμοποιώντας το β = βήμα εκπαίδευσης (= μικρή θετική τιμή)
- Παραγώγιση του κόστους ως προς τα συναπτικά βάρη

$$w_{ij}(k+1) - w_{ij}(k) = -\beta \frac{\partial J}{\partial w_{ij}(k)}$$

$$w_{ij}(k+1) = w_{ij}(k) - \beta \frac{\partial J}{\partial w_{ij}(k)}$$

Ο κανόνας Back-Propagation (BP)

- Έχει υπολογιστεί η παράγωγος και ο κανόνας κατάβασης δυναμικού για ένα MLP πολλών στρωμάτων [Paul Werbos 1974]
- Ισχύουν διαφορετικοί τύποι για το εξωτερικό στρώμα *L* και για τα υπόλοιπα στρώματα. Για το εξωτερικό στρώμα *L:*



$$w_{ij}(k+1) = w_{ij}(k) + \beta\, \delta_i^{(k)}\, a_j^{(k)}$$

$$i = 1, \ldots, N(L)$$

$$j = 0, 1, \ldots, N(L-1)$$

$$\delta_i^{(k)} = (d_i^{(k)} - y_i^{(k)})\, f'(u_i^{(k)})$$

Ο κανόνας BP (2)

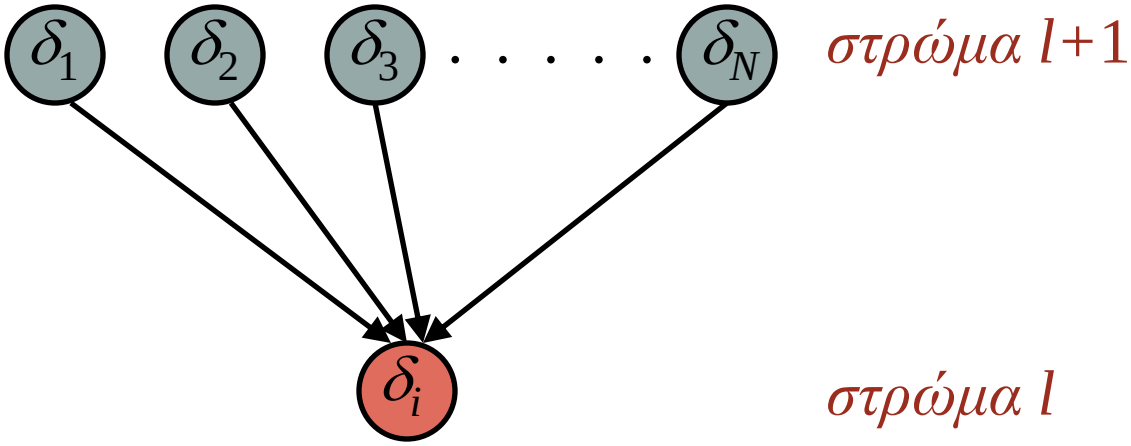
Το σφάλμα δ:

- Αποτελείται από το γινόμενο δύο όρων

1. Το σφάλμα $(d_i - y_i)$ και
2. Την παράγωγο $f'(u_i)$ της συνάρτησης ενεργοποίησης f των νευρώνων του στρώματος L. Ευτυχώς η παράγωγος υπολογίζεται εύκολα για τις σημαντικότερες συναρτήσεις όπως:
   - f = σιγμοειδής → $f'(u_i) = f(u_i)[1-f(u_i)] = a_i[1-a_i]$
   - f = tanh → $f'(u_i) = = 1 - f(u_i)^2 = 1 - a_i^2$
   - f = γραμμική → $f'(u_i) = 1$

Ο κανόνας BP (3)

- Για οποιοδήποτε εσωτερικό στρώμα $l < L$:
  - *Υπολογισμός* $\delta_i(l)$ → προώθηση προς τα πίσω (backward phase). Υπολογίζω τα $\delta_i(l)$ χρησιμοποιώντας τα $\delta_i(l+1)$ του πιο πάνω στρώματος.



Ο κανόνας BP (4)

- Για το εσωτερικό στρώμα $l < L$ (*συνέχεια*)

$$w_{ij}(k+1) = w_{ij}(k) + \beta\, \delta_i^{(k)}\, a_j^{(k)}$$

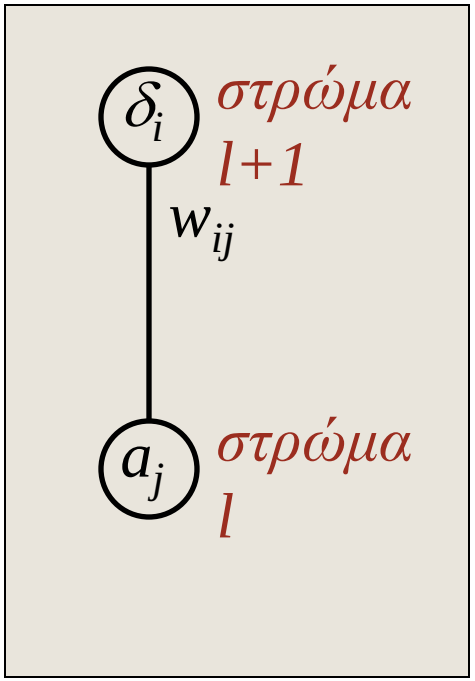
$$i = 1, \ldots, N(l)$$

$$j = 0, 1, \ldots, N(l-1)$$

$$\delta_i^{(k)}(l) = f'(u_i^{(k)}(l)) \sum_{\mu=1}^{N(l+1)} w_{\mu i}(l+1)\delta_\mu(l+1)$$

- Ο τύπος είναι ίδιος με το στρώμα L.

Το μόνο που αλλάζει είναι ο υπολογισμός του δ.



Ο κανόνας BP (5)

Αρχικοποίησε τα βάρη $w_{ij}(l)$ σε μικρές τυχαίες τιμές

Για κάθε εποχή = 1, 2, …, MAXepoch {

Για κάθε πρότυπο p = 1, …, P { */* Εκπαίδευση */*

/ Φάση ανάκλησης = Forward phase */*

/ Φάση υπολογισμού δ = Backward phase */*

/ Φάση ενημέρωσης βαρών = Update phase */*

}

J=0;

Για κάθε πρότυπο p = 1, …, P { */* Υπολογισμός J */*

/ Φάση ανάκλησης = Forward phase */*

/ Άθροισε το τετραγωνικό σφάλμα στο J */*

}

} *Μέχρι* J μικρότερο από κάποιο κατώφλι MINJ

Εφαρμογή και Προβλήματα

- Δυνατότητα επίλυσης πολύπλοκων προβλημάτων. Καλύτερος ταξινομητής από το Perceptron και το ADALINE. Universal Approximator.
- Κατάβαση Δυναμικού → Πρόβλημα τοπικών ελαχίστων: Μπορεί να «κολλήσει» σε μια λύση που δεν είναι ιδιαίτερα καλή
- Αργή σύγκλιση
- Ποιο β να επιλέξω; Δεν υπάρχει εύκολη και σαφής απάντηση. Συνήθως βάζω τυχαία μικρό β και δοκιμάζω.

Παραλλαγές BP

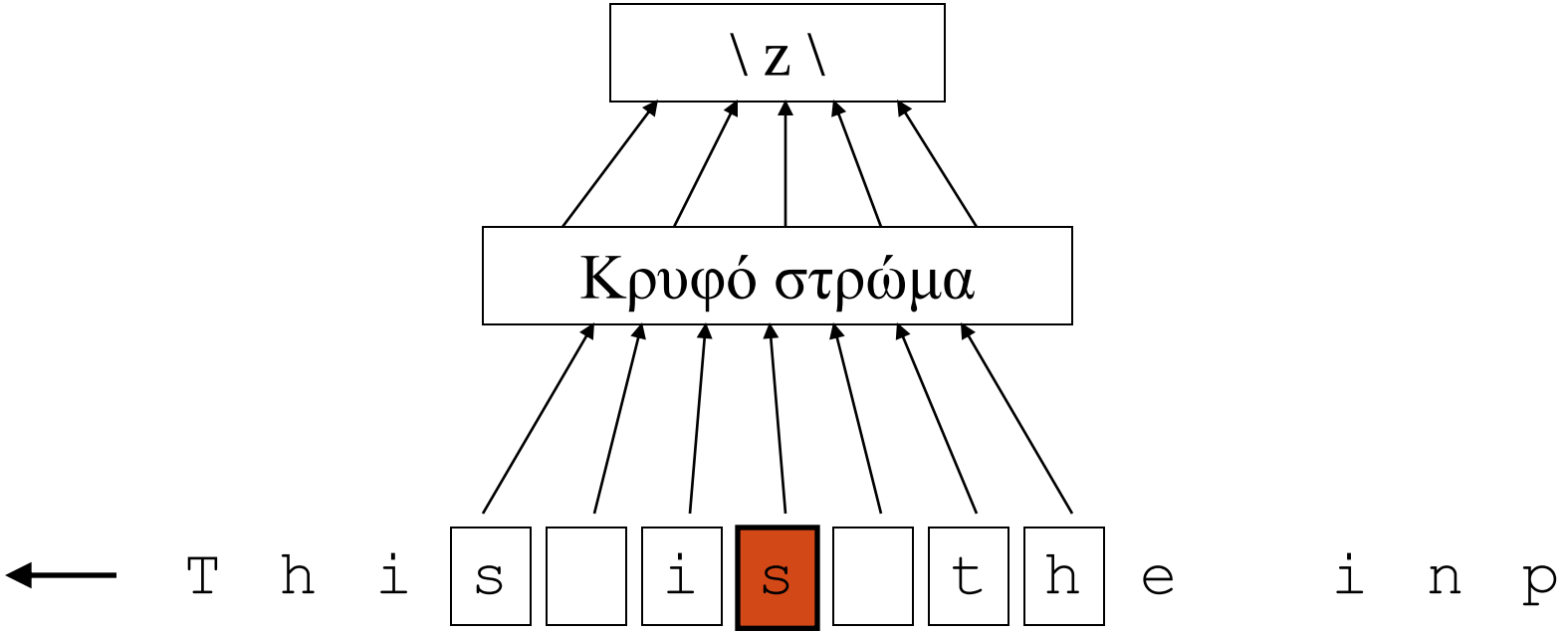
- Χρήση ορμής (momentum): Φυλάω την προηγούμενη διόρθωση $\Delta w_{ij}(k)$ και την προθέτω στην τωρινή

$$\Delta w_{ij}(k+1) = -\beta \frac{\partial J}{\partial w_{ij}(k)} + \mu \, \Delta w_{ij}(k)$$

- Ο συντελεστής μ πρέπει να είναι μικρότερος από το 1 αλλά κοντά στο 1 (π.χ. 0.95). Όσο πιο κοντά, τόσο πιο γρήγορα τρέχει, αλλά αν είναι πολύ κοντά μπορεί να οδηγήσει σε απόκλιση.
- Αποτέλεσμα: Σε περιοχές όπου το *J* μειώνεται αργά, η ορμή επιταχύνει τον αλγόριθμο. Σε περιοχές όπου το *w* ταλαντώνεται επιβάλλει εξομάλυνση (είτε τείνει να κινείται προς μια κατεύθυνση είτε μειώνει τις ταλαντώσεις).
- Άλλες παραλλαγές: Χρήση line search, Χρήση παραγώγων δευτέρου βαθμού (Hessian)

Εφαρμογές

- Text-To-Speech (NetTalk)



Εφαρμογές (2)
NetTalk

- 7x29 είσοδοι, κωδικοποιούν 7 διαδοχικούς χαρακτήρες + σημεία στίξης
- 80 κρυφοί νευρώνες
- 26 έξοδοι κωδικοποιούν φωνήματα
- Εκπαίδευση 1024 λέξεις
- 95% ακρίβεια μετά από 50 εποχές
- Γενίκευση 78% άγνωστο κείμενο

Εφαρμογές (3)
Αναγνώριση χειρόγραφων χαρακτήρων

Αναγνώριση ταχ. Κωδικών ΗΠΑ (zip codes)

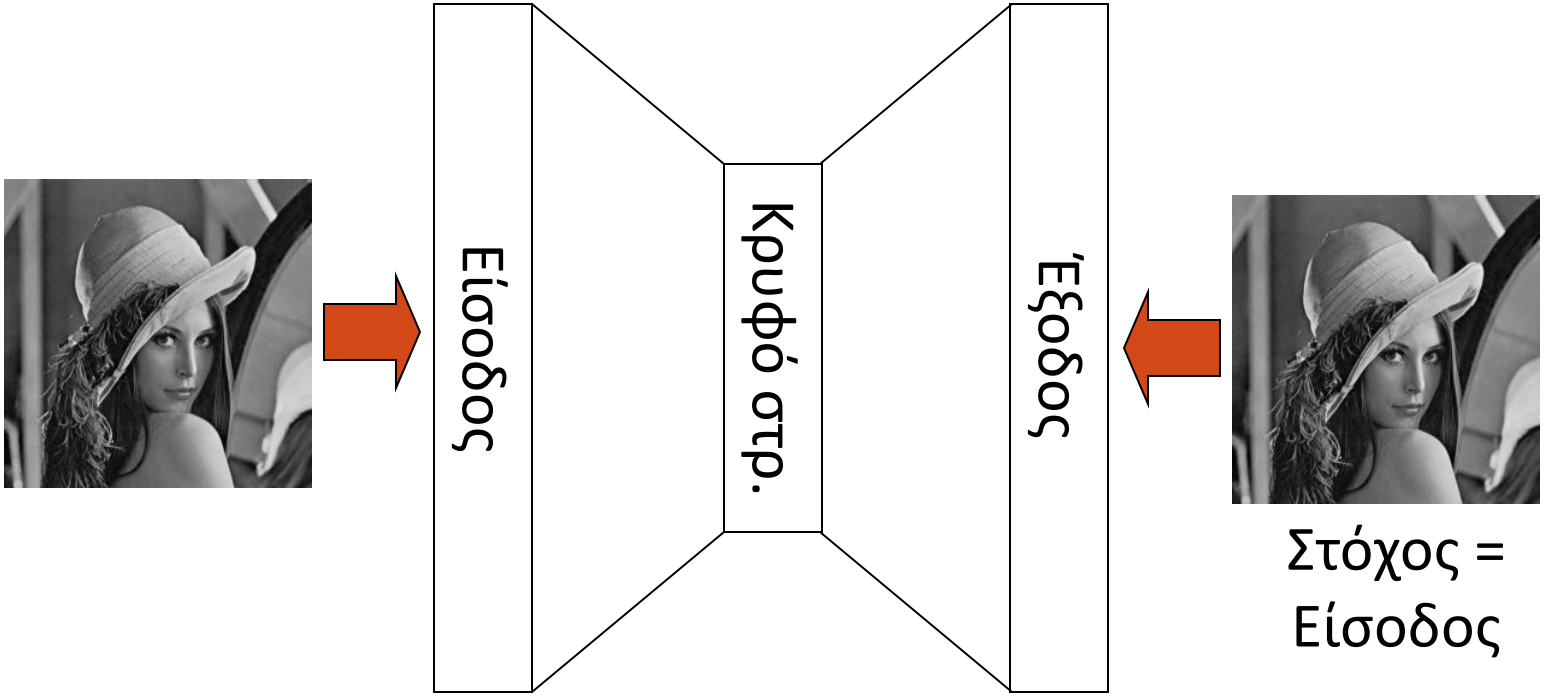
- 4 στρώματα + στρώμα εισόδου
- Είσοδος = εικόνα 16x16
- Στρώμα 1+2 = feature extraction
- Στρώμα 3 = 30 νευρώνες πλήρως συνδεδ.
- Στρώμα 4 = 10 νευρώνες εξόδου

Εφαρμογές (4)
Αναγνώριση ZIP codes

- Σύνολο 1256 νευρώνες, 9760 παράμετροι προς εκπαίδευση.
- Εκπαίδευση με BP.
- Πρότυπα εκπαίδευσης = 7300 χειρόγραφα ψηφία, σφάλμα 1%
- Πρότυπα ελέγχου = 2000 χειρόγραφα ψηφία, σφάλμα 5%

Εφαρμογές (5)

- Συμπίεση εικόνων



Εφαρμογές (6)

- Τάβλι
- Πλοήγηση οχήματος
- Αναγνώριση ομιλίας
- Συλλαβισμός λέξεων
- Αναγνώριση πλαστών συνδιαλλαγών
- Εκτίμηση τιμών μετοχών
- κλπ