Το Πολυεπίπεδο Perceptron

(Διασκευή διαφανειών από ΕΑΠ-ΠΛΗ31)

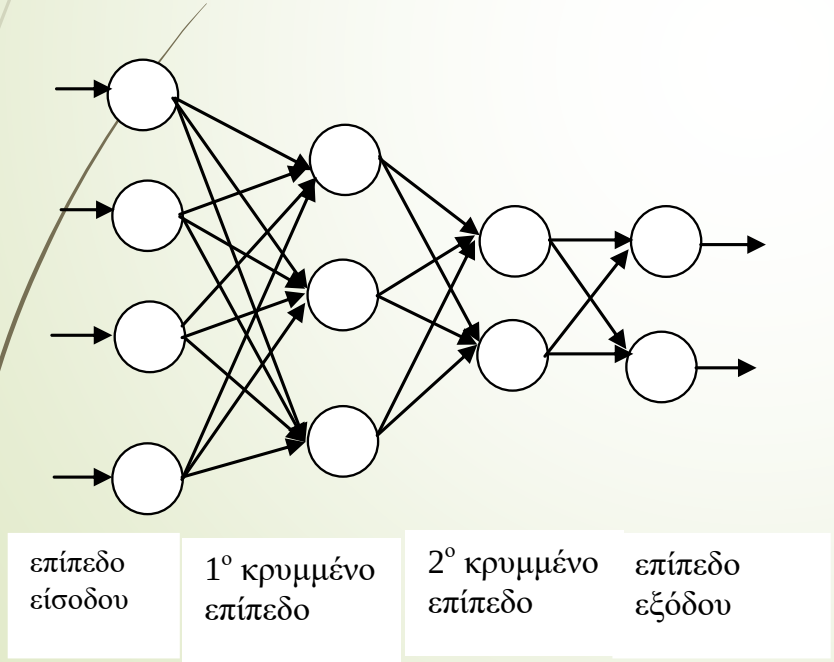
Διδάσκων:

Ι. ΧΑΤΖΗΛΥΓΕΡΟΥΔΗΣ

Πανεπιστήμιο Πατρών, Τμήμα Μηχ/κών Η/Υ και Πληροφορικής

Το Πολυεπίπεδο Perceptron (MultiLayer Perceptron-MLP)

- Δίκτυο πρόσθιας τροφοδότησης (feedforward).
- Οι νευρώνες οργανωμένοι σε **επίπεδα ή στρώματα (layers):** Επίπεδο εισόδου, ένα ή περισσότερα κρυμμένα επίπεδα μη γραμμικών νευρώνων εσωτερικού γινομένου, επίπεδο εξόδου.



- Πλήρης διασύνδεση μεταξύ των νευρώνων δύο διαδοχικών επιπέδων. **Συνήθως δεν** επιτρέπονται συνδέσεις μεταξύ νευρώνων που ανήκουν σε επίπεδα που δεν είναι διαδοχικά.

Το Πολυεπίπεδο Perceptron

- Συμβολισμός i^{ℓ} : νευρώνας i του επιπέδου ℓ .
- $\mathrm{u}_{\mathrm{i}}^{(\ell)}$: τη συνολική είσοδο στο νευρώνα
- $y_{\mathrm{i}}^{(\ell)}$: την έξοδο του νευρώνα
- $\delta_{\mathrm{i}}^{(\ell)}$: το σφάλμα του νευρώνα
- $w_{\mathrm{i}0}^{(\ell)}$: την πόλωση του νευρώνα (ή $b_{\mathrm{i}}^{(\ell)}$)
- g_{ℓ} : τη συνάρτηση ενεργοποίησης των νευρώνων στο επίπεδο ℓ
- d_{ℓ} : τον αριθμό των νευρώνων στο επίπεδο ℓ
- $w_{\mathrm{ij}}^{(\ell)}$: βάρος της σύνδεσης από το νευρώνα i^{ℓ} στο νευρώνα $j^{\ell-1}$

Το Πολυεπίπεδο Perceptron

- Έστω ένα MLP με **d** εισόδους, **p** εξόδους και **H** κρυμμένα επίπεδα. Το επίπεδο εισόδου είναι το μηδέν το επίπεδο εξόδου το H+1. (d_0 = d, d_{H+1} = p)
- **Ευθύ πέρασμα (forward pass)** (δοθέντος του διανύσματος εισόδου υπολογίζεται το διάνυσμα εξόδου):
- Επίπεδο εισόδου: $y_i^{(0)}=x_i$, $y_0^{(0)}=x_0=1$
- Κρυμμένα επίπεδα και επίπεδο εξόδου: Για h=1,…,H+1

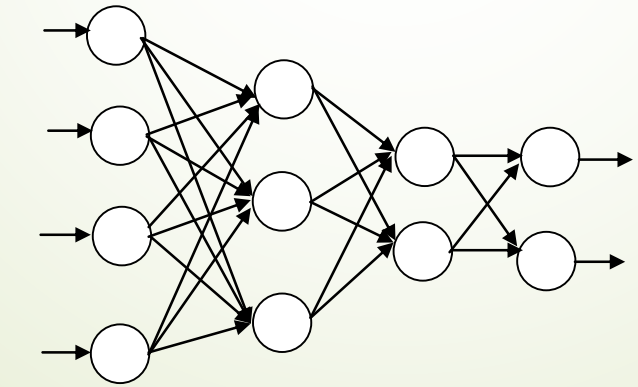
$$u_i^{(h)}=\sum_{j=0}^{d_{h-1}} w_{ij}^{(h)} y_j^{(h-1)} \quad <==> \quad u_i^{(h)}=\sum_{j=1}^{d_{h-1}} w_{ij}^{(h)} y_j^{(h-1)} + w_{i0}^{(h)}, \quad i=1,\ldots d_h$$

$$y_i^{(h)}=g_h(u_i^{(h)}) \quad i=1,\ldots,d_h, \quad y_0^{(h)}=1$$

- Έξοδος του δικτύου: $o_i=y_i^{(H+1)} \quad i=1,\ldots,p$

Το Πολυεπίπεδο Perceptron

- Η συνάρτηση ενεργοποίησης των **κρυμμένων νευρώνων είναι μη γραμμική** (συνήθως **λογιστική**: σ(u)=1/(1+exp(-u))).
- Στο **επίπεδο εξόδου** η συνάρτηση ενεργοποίησης είναι συνήθως **γραμμική ή λογιστική** ανάλογα με το προς επίλυση πρόβλημα.
- Για προβλήματα ταξινόμησης προτιμάται η λογιστική και για προβλήματα συναρτησιακής προσέγγισης η γραμμική.



Υπολογιστικές Δυνατότητες του MLP

- Το MLP υλοποιεί συναρτησιακή προσέγγιση (απεικόνιση) από το χώρο των εισόδων στο χώρο των εξόδων.
- Η απεικόνιση που επιθυμούμε να υλοποιήσουμε καθορίζεται από τα παραδείγματα εκπαίδευσης.
- Το MLP χαρακτηρίζεται από την ιδιότητα της **παγκόσμιας προσέγγισης** (universal approximation): **ένα MLP** με τουλάχιστον ένα κρυμμένο επίπεδο με **μη γραμμικούς νευρώνες μπορεί να προσεγγίσει οποιαδήποτε συνάρτηση με οποιαδήποτε ακρίβεια**, αυξάνοντας επαρκώς τον αριθμό των κρυμμένων νευρώνων.
- Η ιδιότητα αυτή είναι θεωρητικά μόνο σημαντική, αλλά δεν είναι πρακτικά χρήσιμη.

Υπολογιστικές Δυνατότητες του MLP

- Η ύπαρξη των **μη γραμμικών κρυμμένων νευρώνων** προσδίδει στο MLP τις αυξημένες υπολογιστικές δυνατότητες.
- Το MLP **μπορεί** να επιλύσει προβλήματα ταξινόμησης που είναι **μη γραμμικά** διαχωρίσιμα.
- **Θεωρητικά** μπορεί να υλοποιήσει οποιαδήποτε επιφάνεια διαχωρισμού όσο πολύπλοκη και εάν είναι.
- Συνήθως βάζουμε 1 ή 2 κρυμμένα επίπεδα – τα τελευταία χρόνια χρησιμοποιούνται και περισσότερα (deep neural networks)

Εκπαίδευση του MLP

- Έστω **σύνολο εκπαίδευσης D={(x^n,t^n)}**, n=1,…,N.
- $x^n=(x_{n1},\ldots,x_{nd})^T$, $t^n=(t_{n1},\ldots,t_{np})^T$ (πρόβλημα συναρτησιακής προσέγγισης).
- Θα πρέπει το MLP να έχει d νευρώνες στο επίπεδο εισόδου και p νευρώνες στο επίπεδο εξόδου.
- Θα πρέπει ο χρήστης να καθοριστεί η υπόλοιπη αρχιτεκτονική: κρυμμένα επίπεδα, αριθμός κρυμμένων νευρώνων ανά επίπεδο, είδος συναρτήσεων ενεργοποίησης.
- o(x^n; w): το διάνυσμα εξόδου του MLP όταν το διάνυσμα εισόδου είναι το x^n, και $\mathbf{w}=(w_1,w_2,\ldots,w_L)^T$ είναι ένα διάνυσμα στο οποίο συγκεντρώνουμε όλα τα βάρη και τις πολώσεις.
- Εκπαίδευση: καθορισμός του διανύσματος **w**.

Εκπαίδευση του MLP

- Στην περίπτωση που για κάποιο διάνυσμα βαρών w η εκπαίδευση είναι τέλεια θα ισχύει ότι (διανυσματική ισότητα):

  $o(x^n; w)=t^n$ για κάθε n=1,…,N

- ή ισοδύναμα

  $o_m(x^n; w)=t_{nm}$ για κάθε n=1,…,N, m=1,…,p

- Σε αναλογία με τον απλό νευρώνα ορίζουμε την **τετραγωνική συνάρτηση σφάλματος**

$$E(w)=\frac{1}{2}\sum_{n=1}^{N}\|t^n-o(x^n;w)\|^2=\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{p}(t_{nm}-o_m(x^n;w))^2$$

$$\Updownarrow$$

$$E(w)=\sum_{n=1}^{N}E^n(w),\quad E^n(w)=\frac{1}{2}\|t^n-o(x^n;w)\|^2=\frac{1}{2}\sum_{m=1}^{p}(t_{nm}-o_m(x^n;w))^2$$

Εκπαίδευση του MLP

$$E(w)=\sum_{n=1}^{N} E^n(w), \quad E^n(w)=\frac{1}{2}\|t^n-o(x^n;w)\|^2=\frac{1}{2}\sum_{m=1}^{p}(t_{nm}-o_m(x^n;w))^2$$

- Το E(w) ως άθροισμα τετραγώνων των σφαλμάτων ανά παράδειγμα (x^n, t^n) έχει κάτω φράγμα την τιμή μηδέν η οποία προκύπτει όταν έχουμε τέλεια εκπαίδευση.
- Εκπαίδευση του MLP: **ενημέρωση του διανύσματος των βαρών w με σκοπό την ελαχιστοποίηση του τετραγωνικού σφάλματος E(w).**
- Όπως και στον απλό νευρώνα η μέθοδος ελαχιστοποίησης που έχει ευρύτερα χρησιμοποιηθεί είναι η **μέθοδος της gradient descent**.
- Χρειάζεται ο υπολογισμός των **μερικών παραγώγων** του σφάλματος E^n ως προς τα βάρη w_i :

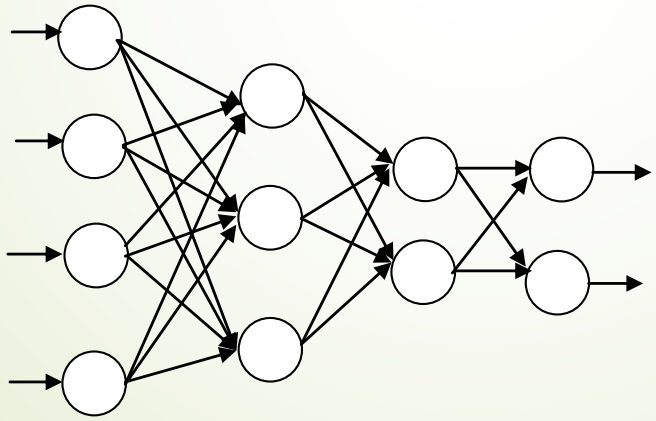
κανόνας οπισθοδιάδοσης σφάλματος (error backpropagation)

Μέθοδος backpropagation

- **Τεχνική υπολογισμού των μερικών παραγώγων** του σφάλματος για ένα παράδειγμα (x,t) ως προς τα βάρη σε ένα δίκτυο πρόσθιας τροφοδότησης με νευρώνες εσωτερικού γινομένου και παραγωγίσιμες συναρτήσεις ενεργοποίησης (MLP).
- Πήρε το όνομά της από το γεγονός ότι βασίζεται στην **προς τα πίσω διάδοση διαμέσου του δικτύου των σφαλμάτων που προκύπτουν στις εξόδους του δικτύου.**
- Για τον υπολογισμό των σφαλμάτων η ροή των υπολογισμών **είναι από την έξοδο προς την είσοδο.**
- **Υπολογίζονται επιμέρους τιμές σφαλμάτων για τους κρυμμένους νευρώνες του δικτύου.**

Μέθοδος backpropagation

- Έστω το παράδειγμα (x^n, t^n) και θέλουμε να υπολογίσουμε τις μερικές παραγώγους του σφάλματος E^n ως προς τα βάρη του MLP.
- Αλγόριθμος back-propagation: δύο περάσματα κατά την εκτέλεση των υπολογισμών: **ευθύ πέρασμα (forward pass)**, και το **αντίστροφο πέρασμα (reverse pass)**.



Μέθοδος backpropagation

- **Ευθύ πέρασμα**: Για διάνυσμα εισόδου x^n υπολογίζεται η έξοδος y κάθε νευρώνα του δικτύου.
- **Αντίστροφο πέρασμα (υπολογισμός σφάλματος δ κάθε νευρώνα)**
  - ✓ αρχίζει από το επίπεδο εξόδου (H+1), όπου συγκρίνονται οι τελικές έξοδοι o_i του δικτύου με τις επιθυμητές t_{ni} παράγοντας **το σφάλμα στις εξόδους** του MLP.
  - ✓ Κατόπιν **τα σήματα σφάλματος διαδίδονται προς τα πίσω στο δίκτυο** και υπολογίζεται σταδιακά **το σφάλμα για τους νευρώνες κάθε επιπέδου** από το τελευταίο κρυμμένο επίπεδο έως το πρώτο κρυμμένο επίπεδο.
- **Μερική παράγωγος βάρους σύνδεσης:**
  - ✓ **σφάλμα προορισμού * έξοδος πηγής**

Μέθοδος backpropagation

- **Υπολογισμός σφαλμάτων (αντίστροφο πέρασμα)**
  - ✓ Νευρώνες εξόδου (επίπεδο H+1) (συν. ενεργοποίησης g_{H+1})

$$\delta_i^{(H+1)} = g'_{H+1}(u_i^{(H+1)})(o_i - t_{ni}),\ i=1,...,p$$

$$\delta_i^{(H+1)} = (o_i - t_{ni}),\ i=1,...,p \text{ (γραμμική συν. ενεργοποίησης)}$$

$$\delta_i^{(H+1)} = o_i(1-o_i)(o_i - t_{ni}),\ i=1,...,p \text{ (λογιστική συν. ενεργοποίησης)}$$

  - ✓ Νευρώνες κρυμμένων επιπέδων: για επίπεδο h=H,…,1 (συν. ενεργοποίησης g_h)

$$\delta_i^{(h)} = g'_h(u_i^{(h)})\sum_{j=1}^{d_{h+1}} w_{ji}^{(h+1)}\delta_j^{(h+1)},\quad i=1,...,d_h$$

$$\delta_i^{(h)} = y_i^{(h)}(1-y_i^{(h)})\sum_{j=1}^{d_{h+1}} w_{ji}^{(h+1)}\delta_j^{(h+1)},\quad i=1,...,d_h \text{ (λογιστική συν. ενεργοποίησης)}$$

Μέθοδος backpropagation

- Μερική παράγωγος βάρους σύνδεσης:

$$\frac{\partial E^n}{\partial w_{ij}^{(h)}} = \delta_i^{(h)} y_j^{(h-1)}$$

- Μερική παράγωγος πόλωσης = σφάλμα του νευρώνα

$$\frac{\partial E^n}{\partial w_{i0}^{(h)}} = \delta_i^{(h)}$$

Εκπαίδευση MLP με gradient descent (ομαδική ενημέρωση)

1. Αρχικοποίηση: Θέτουμε t:=0, αρχικοποίηση βαρών w(0) (τυχαίες τιμές στο διάστημα (-1,1)) και ρυθμού μάθησης η.
2. Σε κάθε επανάληψη t (εποχή), έστω w(t) το διάνυσμα των βαρών

   2.1 Αρχικοποιούμε: $\frac{\partial E}{\partial w_i} = 0,\ i=1,...,L$

   2.2 Για n=1,…,N

      2.2.1 εφαρμογή του κανόνα backpropagation για το (x^n,t^n) και υπολογισμός των $\frac{\partial E^n}{\partial w_i},\ i=1,...,L$

      2.2.2 ενημέρωση του μερικού αθροίσματος: $\frac{\partial E}{\partial w_i} := \frac{\partial E}{\partial w_i} + \frac{\partial E^n}{\partial w_i}$

   2.3 Ενημέρωση των βαρών: $w_i(t+1)=w_i(t)-\eta \frac{\partial E}{\partial w_i},\ i=1,..,L$

   2.4 Έλεγχος τερματισμού. Αν όχι, t:=t+1, goto 2

Εκπαίδευση MLP με gradient descent (σειριακή ενημέρωση)

1. Θέτουμε t:=0, αρχικοποίηση βαρών w(0) (τυχαίες τιμές στο διάστημα (-1,1)) και ρυθμού μάθησης n. Αρχικοποίηση του μετρητή επαναλήψεων (τ:=0) και του μετρητή εποχών (t:=0).
2. Στην αρχή κάθε επανάληψης t (εποχή), έστω w(τ) το διάνυσμα των βαρών

   2.1 Έναρξη εποχής t, αποθήκευση του τρέχοντος διανύσματος βαρών w_{old}=w(τ). Για n=1,…,N

      2.1.1 εφαρμογή του κανόνα backpropagation για το (x^n,t^n) και υπολογισμός των $\frac{\partial E^n}{\partial w_i}$, $i=1,...,L$

      2.1.2 Ενημέρωση των βαρών: $w_i(\tau+1)=w_i(\tau)-n\frac{\partial E^n}{\partial w_i}$, $i=1,..,L$

      2.1.3 $\tau:=\tau+1$

   2.2 Τέλος εποχής t, έλεγχος τερματισμού. Αν όχι, t:=t+1, goto 2

Εκπαίδευση MLP με gradient descent

- Κριτήρια τερματισμού της εκπαίδευσης (έλεγχος στο τέλος κάθε εποχής)
  - ✓ Μικρή διαφορά στην τιμή του διανύσματος βαρών μεταξύ δύο εποχών.
  - ✓ **Μικρή διαφορά στην τιμή του ολικού σφάλματος E(w) μεταξύ δύο εποχών.**
  - ✓ Μείωση της τιμής του ολικού σφάλματος E(w) κάτω από μια επιθυμητή τιμή.
  - ✓ **Πρόωρο σταμάτημα** (early stopping): χρήση συνόλου επικύρωσης.

MLP για προβλήματα ταξινόμησης (κωδικοποίηση κατηγοριών)

- **Κωδικοποίηση των κατηγοριών**: μετατροπή του προβλήματος ταξινόμησης σε πρόβλημα συναρτησιακής προσέγγισης, μέσω της αντιστοίχισης κάθε κατηγορίας σε κάποιο διάνυσμα (ή τιμή) εξόδου.
- Το αρχικό σύνολο εκπαίδευσης με ζεύγη της μορφής **(δεδομένο, κατηγορία)** μετασχηματίζεται ώστε να περιέχει ζεύγη της μορφής **(δεδομένο, διάνυσμα στόχος)**.
- **Κωδικοποίηση 1-από-p (1-out of-p)** για πρόβλημα p κατηγοριών $C_1,\ldots,C_p$

  Κάθε διάνυσμα-στόχος έχει p συνιστώσες $(t_1,\ldots,t_p)$ και η κατηγορία C_k κωδικοποιείται θέτοντας $t_k=1$ και $t_i=0$ για $i\neq k$.

MLP για προβλήματα ταξινόμησης (κωδικοποίηση κατηγοριών)

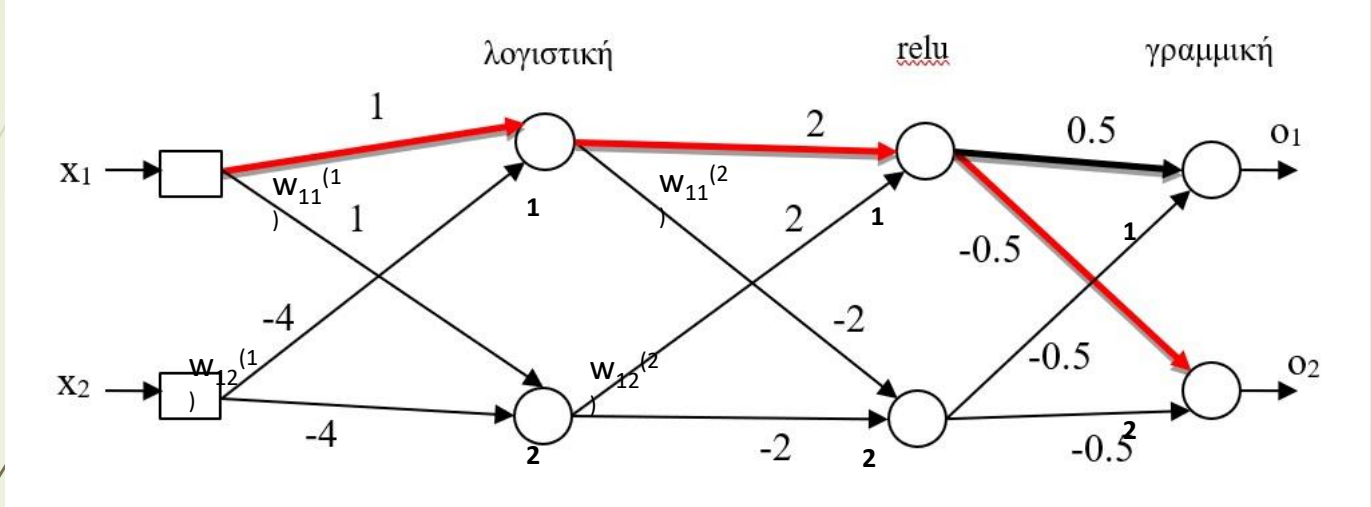
- Παράδειγμα: σε ένα πρόβλημα με τρεις κατηγορίες, τα αντίστοιχα τρία διανύσματα εξόδου είναι τα (1,0,0), (0,1,0), (0,0,1)
  - ✓ Απαιτείται ένα MLP με τρεις εξόδους.
- Η ταξινόμηση ενός προτύπου γίνεται εφαρμόζοντας το πρότυπο ως είσοδο στο δίκτυο και **επιλέγοντας την κατηγορία που αντιστοιχεί στην έξοδο με τη μεγαλύτερη τιμή**.
  - ✓ Όσο πιο κοντά στο 1 είναι αυτή η έξοδος και κοντά στο μηδέν οι υπόλοιπες έξοδοι, τόσο πιο αξιόπιστη είναι η ταξινόμηση.

MLP για προβλήματα ταξινόμησης (κωδικοποίηση κατηγοριών)

- Ειδικά **για την περίπτωση δύο κατηγοριών**, μπορεί να χρησιμοποιηθεί και κωδικοποίηση με μία έξοδο:
  - ✓ αντιστοιχίζουμε την έξοδο t=1 στην μια κατηγορία (C_1)
  - ✓ και την έξοδο t=0 στην άλλη κατηγορία (C_2).
- Στην περίπτωση αυτή, η ταξινόμηση ενός δεδομένου εισόδου γίνεται ως εξής:
  - ✓ αν η έξοδος είναι μεγαλύτερη του 0.5 τότε το πρότυπο ταξινομείται στην κατηγορία C_1, αλλιώς στην κατηγορία C_2.

Παράδειγμα Εκπαίδευσης MLP

Παράδειγμα εκπαίδευσης MLP (1)



Επιπ. 1 Επιπ. 2 Επιπ. 3

$w_{11}^{(1)} = w_{21}^{(1)}=1$ $w_{12}^{(1)}= w_{22}^{(1)}=-4$ $w_{10}^{(1)}= w_{20}^{(1)}=0$

$w_{11}^{(2)} = w_{12}^{(2)}=2$ $w_{21}^{(2)}=w_{22}^{(2)}= -2$ $w_{10}^{(2)}= w_{20}^{(2)}=0$

$w_{11}^{(3)}=0.5$ $w_{21}^{(3)}=-0.5$ $w_{12}^{(3)}= w_{22}^{(3)}=-0.5$ $w_{10}^{(3)}= w_{20}^{(3)}=0$

Παράδειγμα εκπαίδευσης MLP (2)

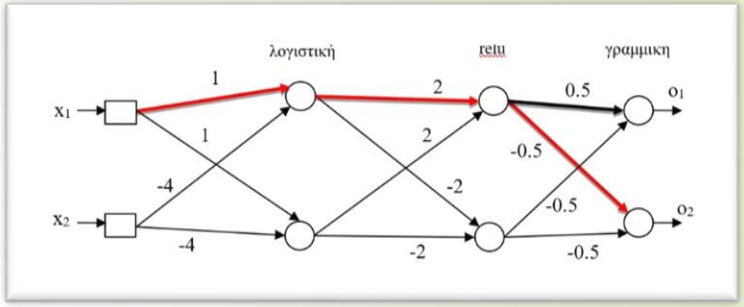
Είσοδος	Έξοδος
x1 = 1	t1 = 0.5
x2 = 0.25	t2 = -0.5

$\eta = 0.2$

Να υπολογιστούν τα ανανεωμένα βάρη της κόκκινης διαδρομής μετά από ένα κύκλο backpropagation (μπρος και πίσω πέρασμα).

Παράδειγμα εκπαίδευσης MLP (3)

Εμπρός πέρασμα



Νευρώνες πρώτου κρυμμένου επιπέδου:

$u_i^{(1)} = 1*1 + (-4)*0.25+0=0$ $\quad y_i^{(1)}=\sigma(0)=0.5$ $\quad i=1,2$

$$u_i^{(h)} = \sum_{j=1}^{d_{h-1}} w_{ij}^{(h)} y_j^{(h-1)} + w_{i0}^{(h)}$$

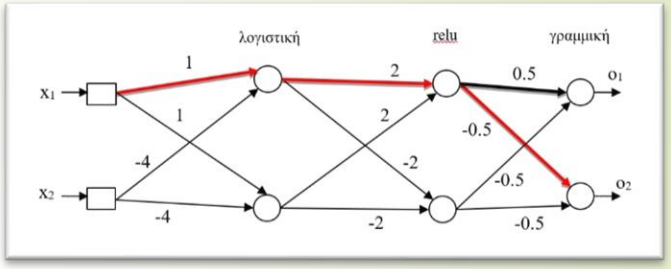
Νευρώνες δεύτερου κρυμμένου επιπέδου:

$u_1^{(2)} = 2*0.5 +2*0.5 + 0 =2$ $\quad y_1^{(2)}=relu(2)=2$

$u_2^{(2)} = (-2) *0.5 + (-2)* 0.5 + 0 =-2$ $\quad y_2^{(2)}=relu(-2)=0$

Είσοδος	Έξοδος
x1 = 1	t1 = 0.5
x2 = 0.25	t2 = -0.5

Παράδειγμα εκπαίδευσης MLP (4)



Νευρώνες εξόδου:

$u_1^{(3)}$=0.5*2 +(-0.5)*0 + 0 =1 $y_1^{(3)}$=linear(1)=1

$u_2^{(3)}$=(-0.5)*2+(-0.5)*0 + 0 =-1 $y_2^{(3)}$=linear(-1)=-1

Άρα για είσοδο x=(1,0.25) οι έξοδοι του δικτύου είναι o=(1,-1)

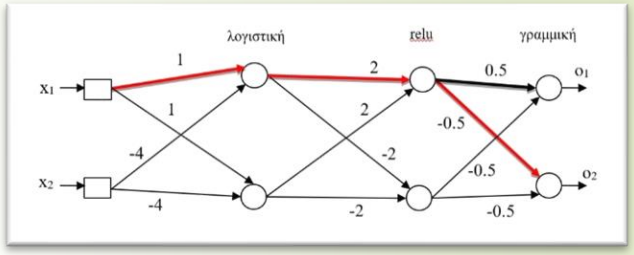
Το **τετραγωνικό σφάλμα** εκπαίδευσης είναι

$e(x,t)= ((0.5-1)^2 + ((-0.5)-(-1))^2) / 2= 0.25.$

$$E(w)= \frac{1}{2}\sum_{m=1}^{p}(t_{nm}-o_m(x^n;w))^2$$

Παράδειγμα εκπαίδευσης MLP (5)

Προς τα πίσω πέρασμα



Η μερική παράγωγος του σφάλματος ως προς κάποιο βάρος σύνδεσης w_{ij} (από τον κόμβο j στον κόμβο i) είναι ίση με το γινόμενο:

(έξοδος του j) x (σφάλμα του i)

Υπολογίζουμε τα σφάλματα $\delta_2^{(3)}$, $\delta_1^{(2)}$ και $\delta_1^{(1)}$ που μας ενδιαφέρουν (κόκκινες συνδέσεις)

<u>Νευρώνες εξόδου</u>: Λόγω γραμμικής συνάρτησης ενεργοποίησης (παράγωγος = 1):

$\delta_1^{(3)} = (o_1 - t_1) = 0.5$, $\delta_2^{(3)} = (o_2 - t_2) = -0.5$

$$\delta_i^{(H+1)} = (o_i - t_{ni})$$

Παράδειγμα εκπαίδευσης MLP (6)

Νευρώνες δεύτερου κρυμμένου επιπέδου: Από τον ορισμό της relu(u), για τον πρώτο νευρώνα του επιπέδου αυτού relu'(2)=1 ενώ για τον δεύτερο νευρώνα relu'(-2)=0. Επομένως

$$\delta_i^{(h)} = g_h'(u_i^{(h)}) \sum_{j=1}^{d_{h+1}} w_{ji}^{(h+1)} \delta_j^{(h+1)}$$

$\delta_1^{(2)}=1*[0.5*0.5+ (-0.5)*(-0.5)] =0.5, \quad \delta_2^{(2)}=0$

Νευρώνες πρώτου κρυμμένου επιπέδου: Οι νευρώνες αυτού του επιπέδου έχουν λογιστική συνάρτηση ενεργοποίησης. Επομένως:

$$\delta_i^{(h)} = y_i^{(h)}(1 - y_i^{(h)}) \sum_{j=1}^{d_{h+1}} w_{ji}^{(h+1)} \delta_j^{(h+1)}$$

$\delta_1^{(1)}= 0.5*(1-0.5)*[2*0.5 + (-2)*0] =0.25$

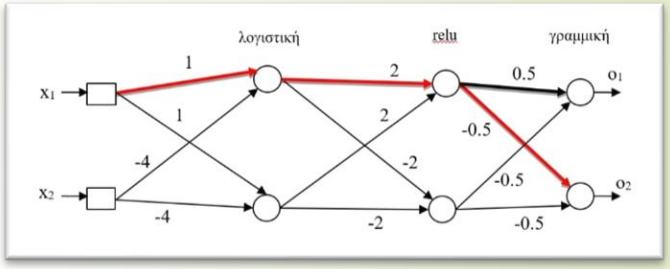
Παράδειγμα εκπαίδευσης MLP (6)

Οι ζητούμενες **μερικές παράγωγοι** είναι:

$\partial e/\partial w_{21}^{(3)}=(-0.5)*2 = -1$

$\partial e/\partial w_{11}^{(2)}= 0.5*0.5=0.25$

$\partial e/\partial w_{11}^{(1)}= 0.25*1 = 0.25$



$$\frac{\partial E^{n}}{\partial w_{ij}^{(h)}} = \delta_{i}^{(h)} y_{j}^{(h-1)}$$

Παράδειγμα εκπαίδευσης MLP (7)

Για ρυθμό μάθησης η=0.2, οι **νέες τιμές βαρ**ών (για τις κόκκινες συνδέσεις) που προκύπτουν από την εξίσωση ενημέρωσης gradientdescent
είναι:

$$w_{21}^{(3)}=-0.5-0.2*(-1)=-0.3$$
$$w_{11}^{(2)}= 2-0.2*0.25=1.95$$
$$w_{11}^{(1)}=1-0.2*0.25=0.95$$

$$w_i(\tau+1)=w_i(\tau)-n\frac{\partial E^n}{\partial w_i}$$

Τα υπόλοιπα βάρη και πολώσεις δεν αλλάζουν.

Παράδειγμα εκπαίδευσης MLP (8)

Εφαρμόζοντας το x=(1, 0.25) σαν είσοδο στο **νέο δίκτυο**, βρίσκουμε έξοδο o= (0.9753, -0.5852) και **νέο τετραγωνικό σφάλμα** e=0.1166. Παρατηρούμε λοιπόν ότι, ακόμα και εάν ενημερώνονται μερικά μόνο από τα βάρη, το τετραγωνικό σφάλμα μειώνεται (από 0.25 σε ~0.12).

Μάθηση και Γενίκευση

(Διασκευή διαφανειών από ΕΑΠ-ΠΛΗ31)

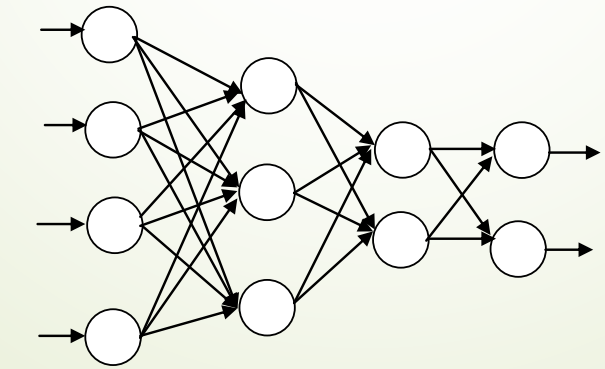
Διδάσκων:

Ι. ΧΑΤΖΗΛΥΓΕΡΟΥΔΗΣ

Πανεπιστήμιο Πατρών, Τμήμα Μηχ/κών Η/Υ και Πληροφορικής

Το Πολυεπίπεδο Perceptron (MultiLayer Perceptron-MLP)

- Έστω **σύνολο εκπαίδευσης D={(x^n,t^n)}**, n=1,…,N.
- $x^n=(x_{n1,...,}x_{nd})^T$, $t^n=(t_{n1,...,}t_{np})^T$
- Θα πρέπει το MLP να έχει d νευρώνες στο επίπεδο εισόδου και p νευρώνες στο επίπεδο εξόδου.
- Ο χρήστης καθορίζει: κρυμμένα επίπεδα, αριθμός κρυμμένων νευρώνων ανά επίπεδο, είδος συναρτήσεων ενεργοποίησης.

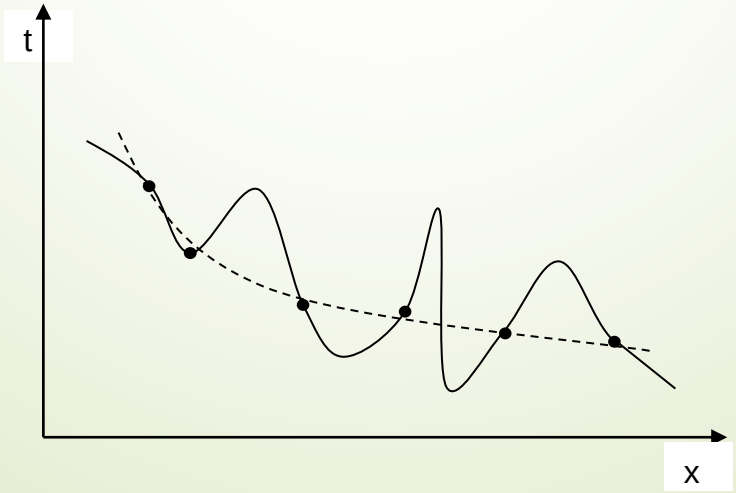


Ικανότητα Γενίκευσης

- Απώτερο στόχος της εκπαίδευσης είναι η η κατασκευή συστημάτων που να παρέχουν σωστές αποφάσεις για παραδείγματα που δεν έχουν χρησιμοποιηθεί κατά την εκπαίδευση: **ικανότητα γενίκευσης** (generalization).
- **Επιλογή αρχιτεκτονικής** στο MLP: με μεγάλο αριθμό κρυμμένων νευρώνων, ένα MLP μπορεί να εκπαιδευτεί ώστε να απεικονίζει με μεγάλη ακρίβεια όλα τα παραδείγματα του συνόλου εκπαίδευσης.
- 'Μεγάλο' MLP → συνήθως μικρή ικανότητα γενίκευσης: `απομνημονεύει' τα δεδομένα εκπαίδευσης και δεν παρουσιάζει καλές επιδόσεις σε νέα δεδομένα διότι, λόγω της μεγάλης 'ευελιξίας' του, δημιουργεί απεικονίσεις οι οποίες είναι συνήθως περισσότερο 'πολύπλοκες' απ' ότι χρειάζεται.

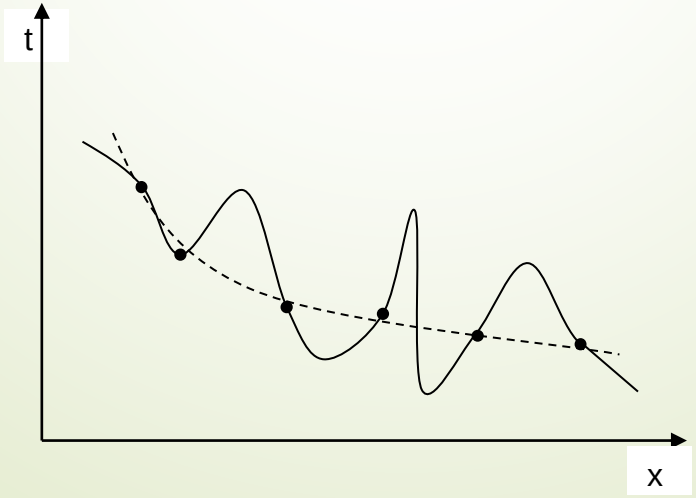
Ικανότητα Γενίκευσης (Παράδειγμα)

- Μονοδιάστατο πρόβλημα απεικόνισης: τα δεδομένα εκπαίδευσης αναπαρίστανται με τις μαύρες κουκίδες.
- Η συνάρτηση που αναπαρίσταται με συνεχή γραμμή, παρότι έχει μηδενικό σφάλμα εκπαίδευσης, είναι περισσότερο πολύπλοκη απ' ότι χρειάζεται (**υπερεκπαίδευση**).
- Η συνάρτηση που αναπαρίσταται με διακεκομμένη γραμμή είναι πιο ομαλή και προτιμότερη ως λύση.



Ικανότητα Γενίκευσης (Παράδειγμα)

- Η πραγματική λύση από την οποία προέκυψαν τα δεδομένα εκπαίδευσης θα μπορούσε να είναι και η πολύπλοκη συνάρτηση. Αν ίσχυε κάτι τέτοιο τα παραδείγματα εκπαίδευσης που έχουμε στη διάθεσή μας δεν είναι αντιπροσωπευτικά.
- Για το συγκεκριμένο παράδειγμα, αφού και οι δύο συναρτήσεις ταιριάζουν επαρκώς στα δεδομένα, **η προτιμότερη λύση είναι η ομαλότερη συνάρτηση** (διακεκομμένη γραμμή).



Occam's razor

- Ενα δίκτυο MLP με **λίγους κρυμμένους** νευρώνες πιθανόν να μην **έχει την απαιτούμενη 'ευελιξία'** ώστε να μπορεί να ορίσει πολύπλοκες περιοχές απόφασης ή να προσεγγίσει συναρτήσεις με πολύπλοκη γραφική παράσταση (**υποεκπαίδευση**).
- Στην περίπτωση που η αρχιτεκτονική του MLP είναι **μεγαλύτερη από τη απαιτούμενη (πιο ευέλικτο δίκτυο) μπορεί να εμφανιστεί υπερεκπαίδευση**.
- **Βασική εμπειρική αρχή μηχανικής μάθησης (occam's razor)**
  - ✓ Προτιμούμε το **απλούστερο δίκτυο** που μπορεί να **μάθει επαρκώς** τα παραδείγματα εκπαίδευσης.
  - ✓ Βασικό ερώτημα: Πώς θα βρούμε το κατάλληλο δίκτυο;

Εκτίμηση της Ικανότητας Γενίκευσης

- Δεν έχει αντιμετωπιστεί επαρκώς με τη χρήση μαθηματικών μεθόδων. Καταφεύγουμε σε **εμπειρικές προσεγγίσεις:** χρήση **συνόλου παραδειγμάτων ελέγχου (test set).**
- Σύνολο ελέγχου: υποσύνολο των παραδειγμάτων που έχουμε στη διάθεσή μας, τα οποία **δεν τα χρησιμοποιούμε** κατά την εκπαίδευση του ΤΝΔ, η οποία γίνεται χρησιμοποιώντας τα υπόλοιπα παραδείγματα.
- Μετά την εκπαίδευση, εφαρμόζουμε τα παραδείγματα του συνόλου ελέγχου ως εισόδους στο ΤΝΔ και υπολογίζουμε τα αντίστοιχα σφάλματα στις εξόδους του.
- **Σφάλμα γενίκευσης: Η μέση τιμή (ή το ποσοστό) των σφαλμάτων ενός ΤΝΔ για τα παραδείγματα του συνόλου ελέγχου.**

Εκτίμηση της Ικανότητας Γενίκευσης

- Μικρό σφάλμα γενίκευσης συνεπάγεται υψηλή ικανότητα γενίκευσης και αντίστροφα.
- Για την αξιολόγηση της ικανότητας γενίκευσης απαιτείται ο **χωρισμός** του συνόλου των διαθέσιμων παραδειγμάτων σε δύο (ξένα μεταξύ τους) υποσύνολα:
  - ✓ το **σύνολο εκπαίδευσης (training set)** που το χρησιμοποιούμε για τον καθορισμό των βαρών του ΤΝΔ
  - ✓ το **σύνολο ελέγχου (test set)** που χρησιμοποιείται για τον υπολογισμό του σφάλματος γενίκευσης του δικτύου που προκύπτει από την εκπαίδευση.
- **Πώς θα γίνει ο χωρισμός**; Ποια παραδείγματα θα χρησιμοποιηθούν για εκπαίδευση και ποια για έλεγχο;

Hold-out

- Εάν τα παραδείγματα **είναι πολλά** δεν έχουμε ιδιαίτερο πρόβλημα (π.χ. τα χωρίζουμε τυχαία σε ποσοστό 70-30%) (μέθοδος **hold-out**).
- Εάν τα παραδείγματα **δεν είναι πολλά** χρειάζονται πιο πολύπλοκες προσεγγίσεις.
- Πολλαπλό hold-out:
  - ✓ Μπορούμε να επαναλάβουμε αρκετές φορές τη διαδικασία hold-out: τυχαία διάσπαση σε σύνολα εκπαίδευσης και ελέγχου, εκπαίδευση του ΤΝΔ και υπολογισμός του σφάλματος γενίκευσης.
  - ✓ Η τελική εκτίμηση για το σφάλμα γενίκευσης προκύπτει ως ο μέσος όρος των επιμέρους σφαλμάτων που υπολογίσαμε.

Cross-Validation

- **Διασταυρωμένη επικύρωση Κ-τμημάτων (K-fold cross-validation (**K-CV):
  - ✓ διαίρεση του συνόλου παραδειγμάτων D σε K *ξένα* μεταξύ τους υποσύνολα (folds) D_1 ,…, D_K (συνήθως K=10).
  - ✓ Για κάθε υποσύνολο D_i (i=1 ,…, K), εκπαιδεύουμε ένα ΤΝΔ θεωρώντας ως σύνολο εκπαίδευσης τα παραδείγματα των υπολοίπων K-1 υποσυνόλων ($D-D_i$) και υπολογίζουμε το σφάλμα γενίκευσης ge_i χρησιμοποιώντας ως σύνολο ελέγχου τα παραδείγματα του υποσυνόλου D_i.
  - ✓ Εκτιμούμε το σφάλμα γενίκευσης (ge) ως το μέσο όρο των επιμέρους σφαλμάτων ge_i
- Είναι πιο συστηματική, χρησιμοποιείται πολύ συχνά.

Leave-one-out

- **Ένα παράδειγμα ελέγχου κάθε φορά (Leave-one-out) (LOT)** **Ειδική περίπτωση** της διασταυρωμένης επικύρωσης K τμημάτων όταν θέσουμε **K=N**, όπου N ο αριθμός όλων των παραδειγμάτων του συνόλου D που έχουμε στη διάθεσή μας.
- Για κάθε (x^i,t^i) του συνόλου D κατασκευάζουμε ένα ΤΝΔ θεωρώντας ως σύνολο εκπαίδευσης ολόκληρο το D εκτός από το συγκεκριμένο παράδειγμα. Στη συνέχεια εκτιμούμε το σφάλμα γενίκευσης ge_i υπολογίζοντας το σφάλμα του ΤΝΔ για το συγκεκριμένο παράδειγμα που αγνοήσαμε κατά την εκπαίδευση.
- Επαναλαμβάνοντας τη διαδικασία για όλα τα (x^i,t^i), $(i=1,\ldots,N)$ εκτιμούμε το σφάλμα γενίκευσης ως το μέσο όρο των ge_i.
- Πιο αξιόπιστη (δεν έχει τυχαιότητα), αλλά αυξημένη πολυπλοκότητα.

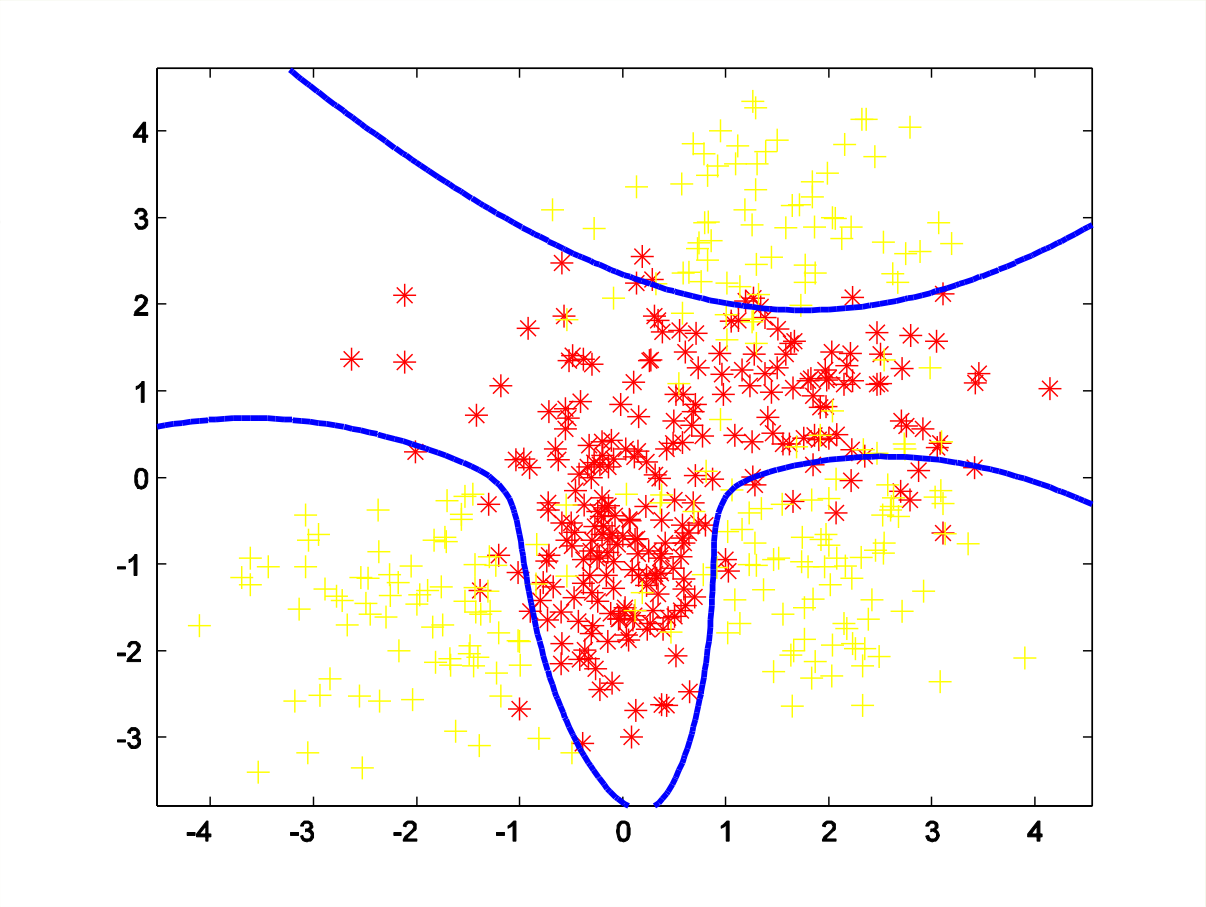
Επιλογή MLP με cross-validation (K-CV) (ένα κρυμμένο επίπεδο με M νευρώνες)

1. Καθορισμός αρχικού M (π.χ. M=2), M_{max}, αριθμού folds K (π.χ. K=10) και παραμέτρων εκπαίδευσης (π.χ. ρυθμός μάθησης).
2. Διαμερισμός του συνόλου παραδειγμάτων D σε υποσύνολα D_1, …, D_K για την εφαρμογή της τεχνικής K-CV.
3. Υπολογισμός με (K-CV) του σφάλματος γενίκευσης ge(M) για M κρυμμένους νευρώνες.
4. Αύξηση του αριθμού των κρυμμένων νευρώνων, π.χ. M:=M+1 και επιστροφή στο βήμα 3 εάν $M \leq M_{max}$.
5. Επιλογή ως βέλτιστης αρχιτεκτονικής εκείνης με το μικρότερο σφάλμα γενίκευσης: $ge(M^*) \leq ge(M)$
6. Εκπαίδευση του MLP με M^* κρυμμένους νευρώνες σε όλο το σύνολο παραδειγμάτων και εύρεση της τελικής λύσης.

Παράδειγμα Εκπαίδευσης

Σύνολο εκπαίδευσης (τεχνητά δεδομένα με θόρυβο)

(μπλε γραμμή: πραγματικό όριο απόφασης)

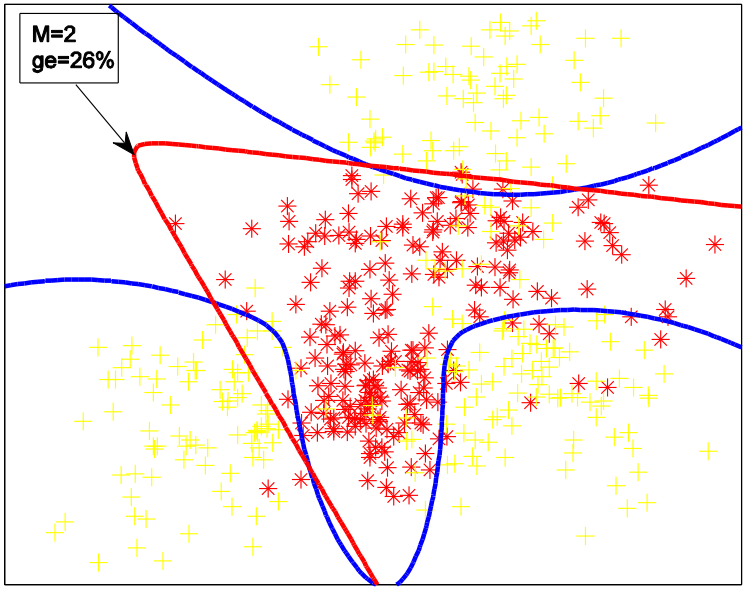


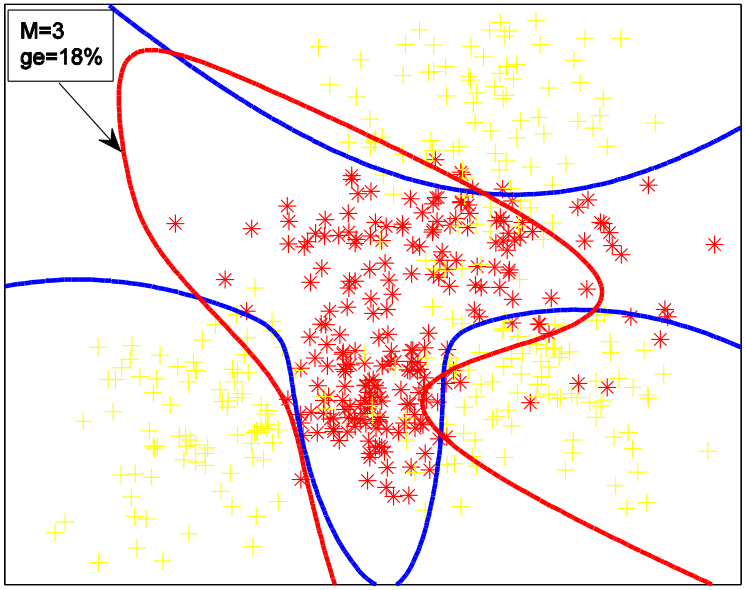
Παράδειγμα Εκπαίδευσης

Εκπαιδεύουμε MLP με ένα κρυμμένο επίπεδο με M νευρώνες

M	Σφάλμα Γενίκευσης (10-CV)	Ικανότητα Γενίκευσης (10-CV)
2	28%	72%
3	18%	82%
4	13%	87%
5	**12%**	**88%**
6	15%	85%
7	15%	85%

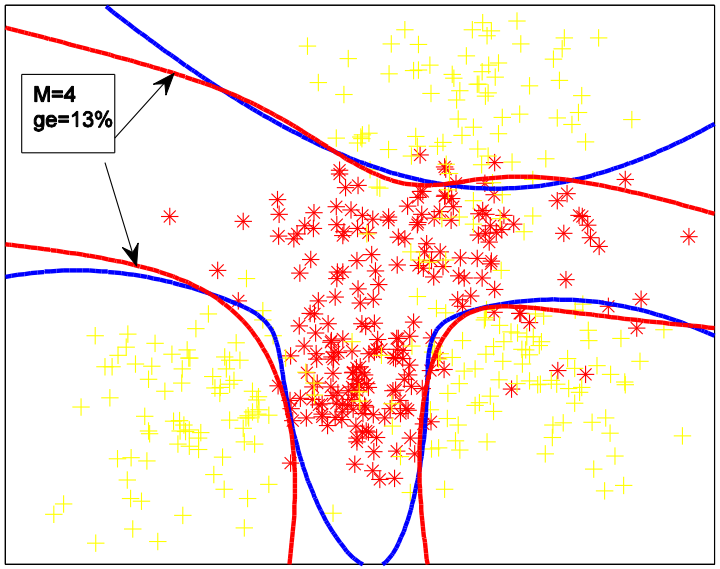
Παράδειγμα Εκπαίδευσης

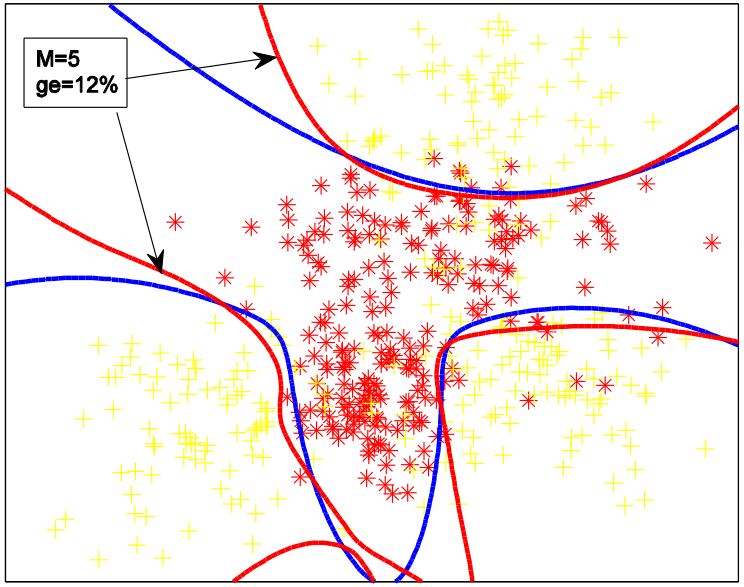




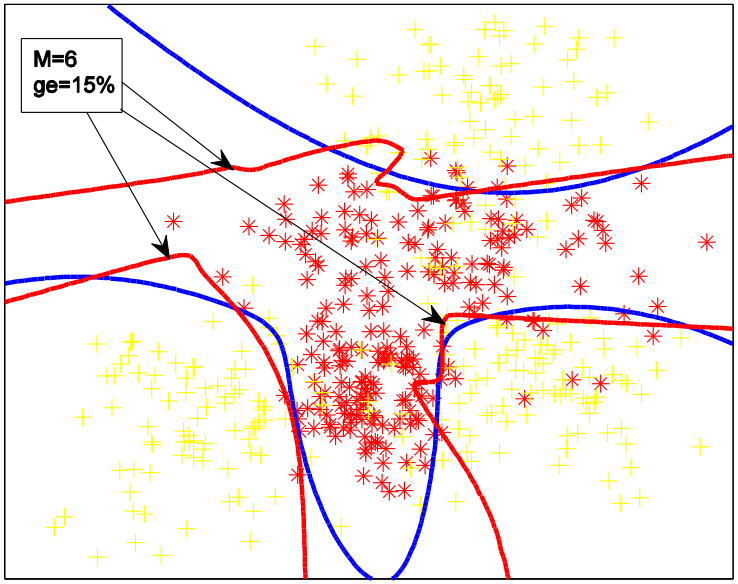
κόκκινη γραμμή: όριο απόφασης MLP

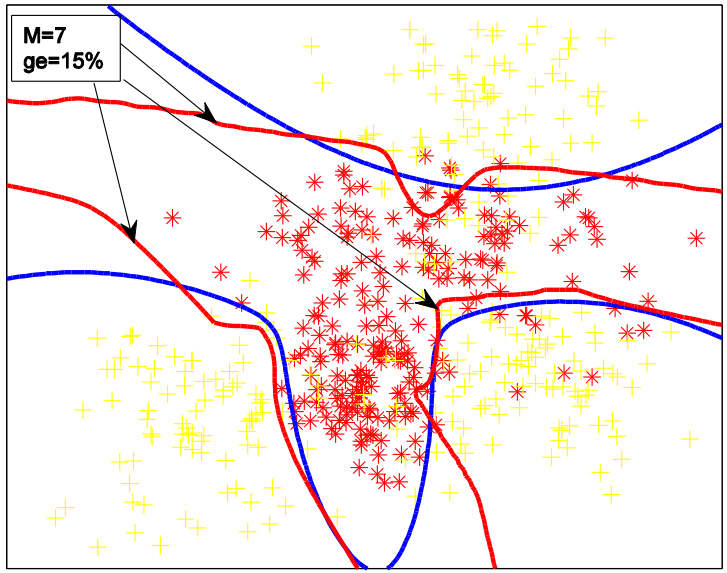
Παράδειγμα Εκπαίδευσης





Παράδειγμα Εκπαίδευσης





Εκτίμηση της Γενικευτικής Ικανότητας

- Δύο ερωτήματα: αν εκπαιδεύσουμε πολλά ΤΝΔ (π.χ. 10-fold CV) για να εκτιμήσουμε την ικανότητα γενίκευσης και να επιλέξουμε τη βέλτιστη αρχιτεκτονική δικτύου:
  - ✓ α) πώς θα κατασκευάσουμε **το τελικό ΤΝΔ** που θα αποτελεί τη λύση στο πρόβλημά μας;
  - ✓ β) ποια θα είναι η ικανότητα γενίκευσης αυτού του τελικού δικτύου;
- Απαντήσεις: α) κατασκευάζουμε το τελικό ΤΝΔ χρησιμοποιώντας την βέλτιστη αρχιτεκτονική που έχουμε βρεί και **όλα τα διαθέσιμα** παραδείγματα εκπαίδευσης.
- β) Η ικανότητα γενίκευσης του τελικού ΤΝΔ έχει ήδη υπολογιστεί από την μέθοδο εκτίμησης της ικανότητας γενίκευσης για τη βέλτιστη αρχιτεκτονική.

Αποφυγή υπερεκπαίδευσης: η μέθοδος της φθοράς των βαρών

- Ο προφανής τρόπος για να περιορίσουμε την ‘ευελιξία’ ενός MLP είναι περιορίζοντας την αρχιτεκτονική του, δηλαδή ουσιαστικά των αριθμό των βαρών του δικτύου.
- Ένας εναλλακτικός τρόπος περιορισμού της ευελιξίας ενός MLP είναι **περιορίζοντας τις τιμές** που μπορούν να πάρουν τα βάρη κατά τη διάρκεια της εκπαίδευσης. Η ιδέα αυτή ονομάζεται **κανονικοποίηση (regularization)**.
- Ο πιο απλός τρόπος για να επιτύχουμε κανονικοποίηση βασίζεται στην προσθήκη ενός **όρου τιμωρίας (penalty term)** στη συνάρτηση τετραγωνικού σφάλματος που ελαχιστοποιούμε κατά την εκπαίδευση του δικτύου.

Η μέθοδος της φθοράς των βαρών

- Πιο συγκεκριμένα, ένας **όρος κανονικοποίησης** που χρησιμοποιείται συχνότερα είναι **το άθροισμα των τετραγώνων των τιμών των βαρών** (όπου L ο αριθμός των βαρών)

$$R(w)=\sum_{i=1}^{L} w_i^2$$

- Η συνάρτηση που ελαχιστοποιείται κατά την εκπαίδευση γίνεται:

$$E_R(w)=E(w)+rR(w)=E(w)+r\sum_{i=1}^{L} w_i^2$$

- E(w) είναι η συνάρτηση τετραγωνικού σφάλματος εκπαίδευσης.
- Η παράμετρος r καθορίζει το σχετικό βάρος των δύο στόχων της εκπαίδευσης: αφενός ελαχιστοποίηση του E(w), αφετέρου διατήρηση μικρών απόλυτων τιμών των βαρών του δικτύου.

Η μέθοδος της φθοράς των βαρών

- Η προσθήκη του όρου κανονικοποίησης στην ουσία παρεμποδίζει τα βάρη να λάβουν υψηλές (κατ' απόλυτη τιμή) τιμές κατά την εκπαίδευση.
- Μερικές φορές οδηγεί κάποιες τιμές των βαρών να γίνουν σχεδόν μηδέν, δηλαδή στην ουσία είναι σαν οι αντίστοιχες συνδέσεις να αφαιρούνται από το δίκτυο.
- Μπορούμε δηλαδή να θεωρήσουμε ότι οι τιμές των βαρών 'φθείρονται' κατά τη διάρκεια της εκπαίδευσης, για το λόγο αυτό η μέθοδος ονομάζεται **εκπαίδευση με φθορά βαρών (weight decay)** .
- Ενημέρωση των βαρών:

$$w_i(t+1)=w_i(t)-\eta\frac{\partial E_R}{\partial w_i}$$

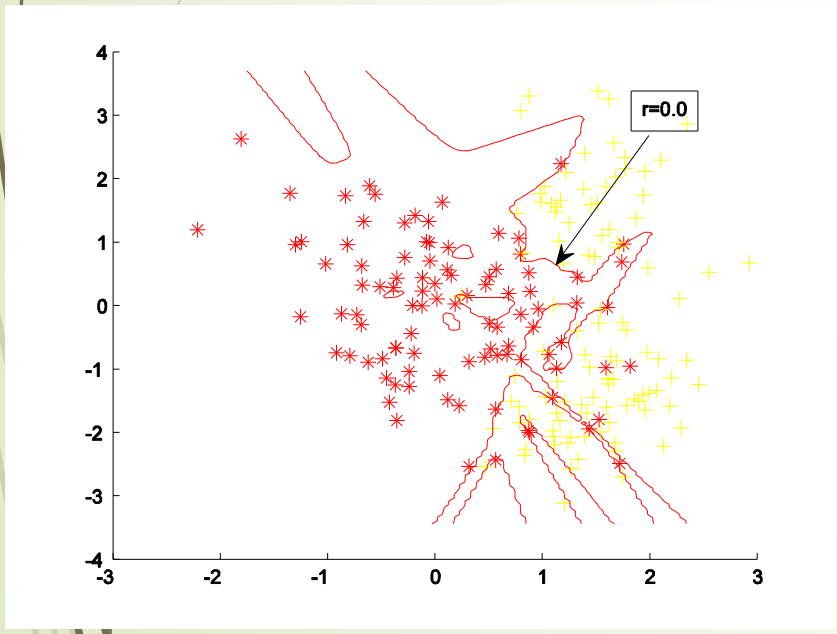
$$w_i(t+1)=w_i(t)-\eta\left(\frac{\partial E}{\partial w_i}+2rw_i(t)\right)$$

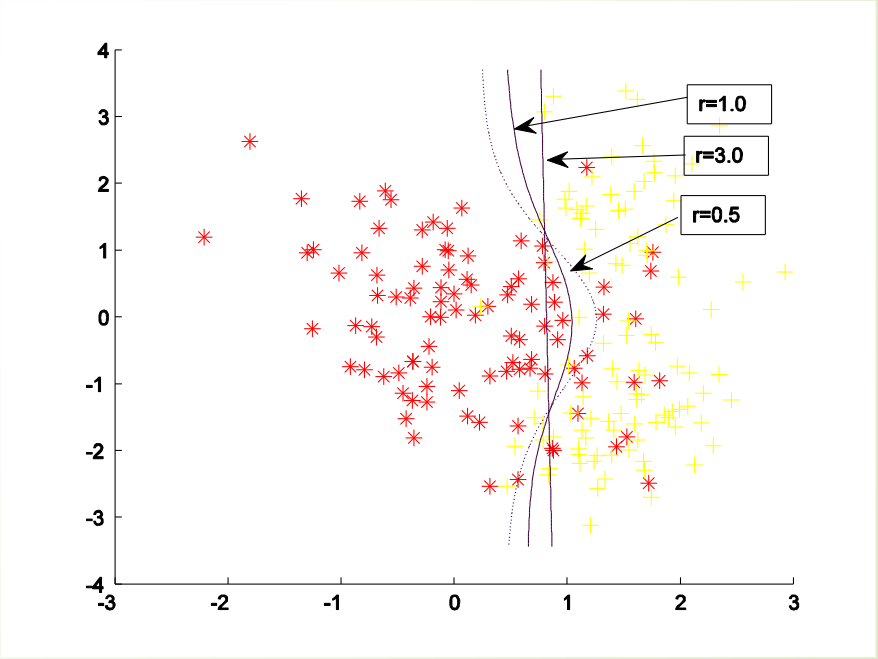
Η μέθοδος της φθοράς των βαρών

- Εάν η παράμετρος r έχει καθοριστεί σωστά και το μέγεθος του δικτύου είναι μεγαλύτερο απ’ ότι απαιτείται, στο τέλος της εκπαίδευσης προκύπτουν συνήθως δίκτυα με καλύτερες δυνατότητες γενίκευσης.
- Εάν η παράμετρος r είναι μεγάλη τότε παρεμποδίζεται η προσαρμογή του δικτύου στα παραδείγματα εκπαίδευσης.
- Εάν η παράμετρος r τείνει στο μηδέν τότε είναι σαν να εκπαιδεύουμε το δίκτυο χωρίς κανονικοποίηση.
- Η σωστή ρύθμιση της παραμέτρου r αποτελεί το βασικό πρόβλημα αυτής της μεθόδου.

Η μέθοδος της φθοράς των βαρών

- MLP με 1 κρυμμένο επίπεδο με 20 νευρώνες





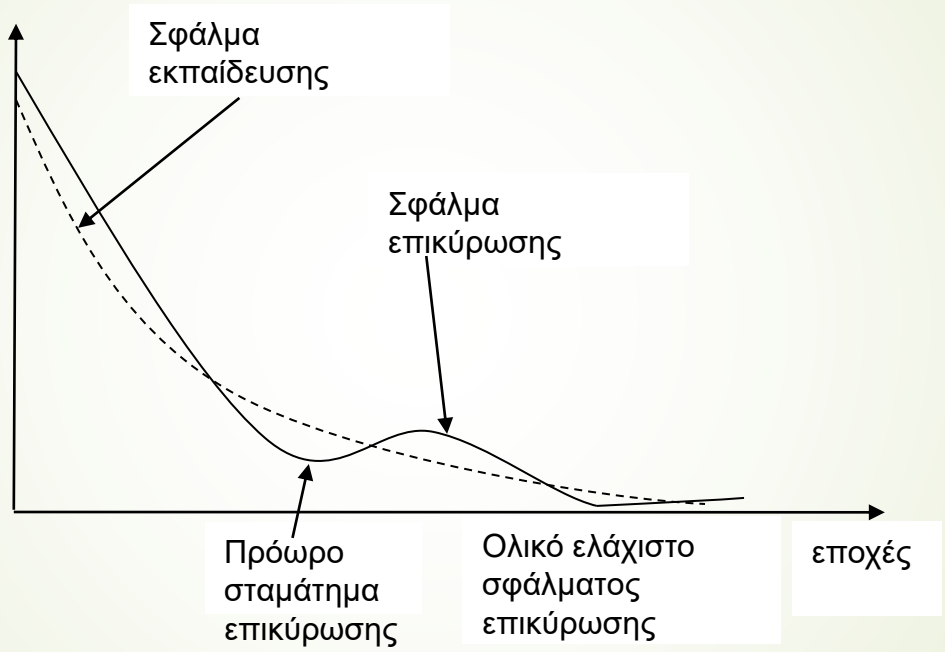
Αποφυγή υπερεκπαίδευσης: πρόωρο σταμάτημα (early stopping)

- Εκπαιδεύουμε το MLP (ενημερώνουμε τα βάρη του) μέσω της ελαχιστοποίησης του σφάλματος εκπαίδευσης.
- Σε τακτά χρονικά διαστήματα (π.χ. κάθε 10 εποχές) **'παγώνουμε' τη διαδικασία εκπαίδευσης και** με τις τρέχουσες τιμές των βαρών **υπολογίζουμε μια εκτίμηση του σφάλματος γενίκευσης** σε ένα ανεξάρτητο σύνολο παραδειγμάτων (διαφορετικό από το σύνολο εκπαίδευσης και το σύνολο ελέγχου).
- Το τρίτο αυτό σύνολο παραδειγμάτων που χρησιμοποιούμε ονομάζεται **σύνολο επικύρωσης (validation set)** και το αντίστοιχο σφάλμα ονομάζεται **σφάλμα επικύρωσης**.
- Κατόπιν συνεχίζουμε τη διαδικασία εκπαίδευσης και της ενημέρωσης των βαρών μέχρι το επόμενο χρονικό σημείο υπολογισμού του σφάλματος επικύρωσης.

Πρόωρο Σταμάτημα (early stopping)

- Στις αρχικές επαναλήψεις της εκπαίδευσης και όσο προχωρεί η εκπαίδευση, μειώνεται το σφάλμα εκπαίδευσης και συγχρόνως μειώνεται και το σφάλμα επικύρωσης.
- Υπάρχει συνήθως ένα **χρονικό σημείο (ειδικά στις περιπτώσεις μεγάλων δικτύων) πέρα από το οποίο περαιτέρω μείωση του σφάλματος εκπαίδευσης οδηγεί σε αύξηση του σφάλματος επικύρωσης**, διότι αρχίζει να εμφανίζεται το φαινόμενο της υπερεκπαίδευσης.
- Στο σημείο αυτό μπορούμε να σταματήσουμε την εκπαίδευση του δικτύου (**πρόωρο σταμάτημα**).

Πρόωρο Σταμάτημα (early stopping)



Πρόωρο Σταμάτημα (early stopping)

- Εναλλακτικά, μπορούμε, αντί να σταματήσουμε πρόωρα, να εκτελέσουμε τον αλγόριθμο εκπαίδευσης μέχρι να τερματίσουμε σε τοπικό ελάχιστο, φροντίζοντας όμως να **αποθηκεύουμε κάθε φορά το διάνυσμα βαρών w_{val} που παρέχει το μικρότερο σφάλμα επικύρωσης που έχουμε υπολογίσει μέχρι στιγμής** κατά τη διάρκεια της εκπαίδευσης.
  - ✓ Η τιμή των βαρών w_{val} στο τέλος της εκπαίδευσης αποτελεί και το τελικό διάνυσμα βαρών για το MLP, διότι παρέχει την ελάχιστη τιμή του σφάλματος επικύρωσης.

Πρόωρο Σταμάτημα (early stopping)

- Συνοψίζοντας, στη μέθοδο του πρόωρου σταματήματος:
  - ✓ α) Το MLP πρέπει να είναι σχετικά μεγάλο.
  - ✓ β) ενημερώνουμε τα βάρη χρησιμοποιώντας τα παραδείγματα του συνόλου εκπαίδευσης
  - ✓ γ) επιλέγουμε ως τελική λύση για τα βάρη αυτή με την μικρότερη τιμή του σφάλματος που υπολογίζουμε χρησιμοποιώντας τα παραδείγματα του συνόλου επικύρωσης.
- **Τίμημα:** θα πρέπει να αφαιρέσουμε ένα ποσοστό των παραδειγμάτων από το σύνολο εκπαίδευσης και να τα βάλουμε στο σύνολο επικύρωσης. Πρόβλημα εάν τα παραδείγματα είναι λίγα. Εξάρτηση από τον διαμερισμό.
- Δεν επιτρέπεται τα σύνολα εκπαίδευσης, επικύρωσης και ελέγχου να έχουν κοινά παραδείγματα.