ΚΕΣ 03 – Αναγνώριση Προτύπων και Ανάλυση Εικόνας

Μη γραμμικοί ταξινομητές – Νευρωνικά Δίκτυα

Τμήμα Επιστήμης και Τεχνολογίας Τηλεπικοινωνιών

Πανεπιστήμιο Πελοποννήσου

- Εισαγωγή
- Πολυεπίπεδες Perceptron
- Ο αλγόριθμος back-propagation
- Επιλογή παραμέτρων
- Παραδείγματα

Περιεχόμενα – Βιβλιογραφία

- Περιεχόμενα Ενότητας
  - Εισαγωγή
  - Πολυεπίπεδες Perceptron
  - Ο αλγόριθμος back-propagation
  - Επιλογή παραμέτρων στον αλγόριθμο back-propagation
  - Παραδείγματα
- Βιβλιογραφία:
  - Duda [2004]: Chapter 6
  - Theodoridis [2002]: Chapter 4
  - Bow [2002]: Chapter 8

★ Εισαγωγή
□ Πολυεπίπεδες Perceptron
□ Ο αλγόριθμος back-propagation
□ Επιλογή παραμέτρων
□ Παραδείγματα

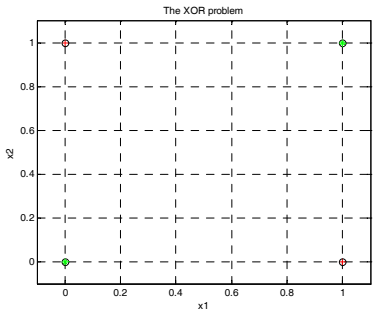
Εισαγωγή

- Μέχρι τώρα είδαμε ότι οι γραμμικοί ταξινομητές μπορούν να σχεδιαστούν με τέτοιο τρόπο ώστε να ελαχιστοποιούν το σφάλμα ταξινόμησης σε γραμμικά διαχωρίσιμα δεδομένα (αλγόριθμος Perceptron, SVM) αλλά και σε μη γραμμικά διαχωρίσιμα δεδομένα (αλγόριθμος Pocket, SVM)
  - Υπάρχουν εντούτοις προβλήματα στα οποία οι γραμμικοί ταξινομητές δεν έχουν αποδεκτή απόδοση
  - Σε τέτοιους είδους προβλήματα πολυεπίπεδα δίκτυα Perceptron σε συνδυασμό με το αλγόριθμο back-propagation αποδεικνύονται ιδιαίτερα αποτελεσματικά
  - Τα πολυεπίπεδα Perceptron αποτελούν μια υποκατηγορία μιας ευρείας κατηγορίας μη γραμμικών ταξινομητών που ονομάζονται Νευρωνικά Δίκτυα.
  - Άλλα είδη Νευρωνικών Δικτύων είναι τα δίκτυα RBF, τα δίκτυα Hopfiled, κ.λπ.

★ Εισαγωγή
□ Πολυεπίπεδες Perceptron
□ Ο αλγόριθμος back-propagation
□ Επιλογή παραμέτρων
□ Παραδείγματα

Το πρόβλημα XOR

x_1	x_2	XOR	Class
0	0	0	ω_2
0	1	1	ω_1
1	0	1	ω_1
1	1	0	ω_2



- Το απλούστερο μη γραμμικό πρόβλημα είναι το πρόβλημα XOR (βλέπε σχήμα και πίνακα αληθείας στα αριστερά).
  - Είναι εμφανές ότι δεν μπορούμε να βρούμε μια γραμμή:

$$g(\underline{x}) = \underline{w}^T \underline{x} + w_0$$

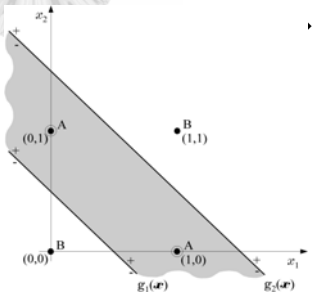
για την οποία να ισχύει:

$$g(\underline{x}) > 0, \quad \underline{x} \rightarrow \omega_1$$
$$g(\underline{x}) < 0, \quad \underline{x} \rightarrow \omega_2$$

(δηλαδή να διαχωρίζει τις κλάσεις ω_1, ω_2)

☑ Εισαγωγή
★ Πολυεπίπεδες Perceptron
☐ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

Πολυεπίπεδες Perceptron

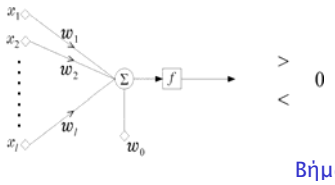


- Το πρόβλημα XOR μπορεί να λυθεί χρησιμοποιώντας δύο γραμμές.
  - Η κλάση ω_2, βρίσκεται εκτός της σκιαγραμμισμένης περιοχής και η κλάση ω_1 βρίσκεται εντός.
  - Το XOR πρόβλημα μπορεί να λυθεί σε δύο βήματα:

Βήμα 1: Σχεδιάζουμε δύο γραμμές:

$$g_1(\underline{x}) = g_2(\underline{x}) = 0$$

Καθεμία από τις γραμμές υλοποιείται με μια μηχανή Perceptron (νευρώνας). Η έξοδος τους θα είναι:



$$y_i = f(g_i(\underline{x})) = \begin{cases} 0 \\ 1 \end{cases} \quad i = 1, 2$$

Βήμα 2: Βρίσκουμε τη θέση του $\underline{x}$ σε σχέση και με τις δύο γραμμές (τιμές y_1, y_2)

☑ Εισαγωγή
★ Πολυεπίπεδες Perceptron
☐ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

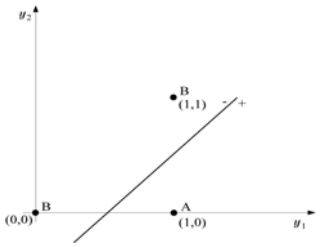
Πολυεπίπεδες Perceptron (II)

1st phase				2nd phase
x_1	x_2	y_1	y_2	
0	0	0(−)	0(−)	$\omega_2(0)$
0	1	1(+)	0(−)	$\omega_1(1)$
1	0	1(+)	0(−)	$\omega_1(1)$
1	1	1(+)	1(+)	$\omega_2(0)$

- Ισοδύναμα:
  - Οι υπολογισμοί στο πρώτο βήμα πραγματοποιούν μια αντιστοίχιση:

$$\underline{x} \rightarrow \underline{y} = [y_1, y_2]^T$$

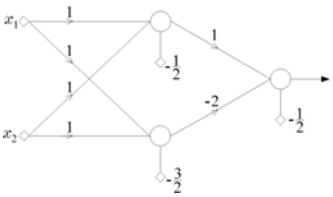
  - Η ταξινόμηση γίνεται τώρα στα μετασχηματισμένα δεδομένα y (βλέπε σχήμα).
  - Η ταξινόμηση αυτή πραγματοποιείται από μια γραμμή (υπερεπιφάνεια) και επομένως μπορεί να υλοποιηθεί μέσω μιας μηχανής Perceptron.



Οι υπολογισμοί στο πρώτο βήμα εκτελούν μια αντιστοίχιση η οποία μετασχηματίζει το μη γραμμικά διαχωρίσιμο πρόβλημα σε διαχωρίσιμο

☑ Εισαγωγή
★ Πολυεπίπεδες Perceptron
☐ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

Perceptron δύο επιπέδων



$$g_1(\underline{x}) = x_1 + x_2 - \frac{1}{2} = 0$$

$$g_2(\underline{x}) = x_1 + x_2 - \frac{3}{2} = 0$$

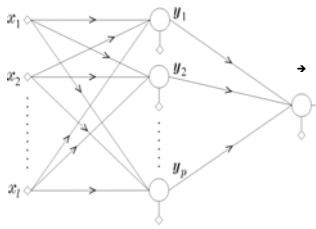
$$g(\underline{y}) = y_1 - 2y_2 - \frac{1}{2} = 0$$

- Η αρχιτεκτονική του συνολικού ταξινομητή για το πρόβλημα XOR φαίνεται στο σχήμα.
  - Η αρχιτεκτονική αυτή είναι γνωστή ως μηχανή Perceptron δύο επιπέδων με ένα κρυφό (hidden) (δύο εσωτερικοί νευρώνες) επίπεδο και ένα επίπεδο εξόδου (εξωτερικός νευρώνας)

    Η συνάρτηση ενεργοποίησης f όλων των νευρώνων δίνει έξοδο 0 ή 1. $f(\cdot) = \begin{cases} 0 \\ 1 \end{cases}$
  - Τα δεδομένα εισόδου καθορίζουν το επίπεδο εισόδου το οποίο όμως δεν πραγματοποιεί κάποια επεξεργασία.
  - Συνολικά η μηχανή Perceptron δύο επιπέδων του παραδείγματος υλοποιεί τις γραμμές (υπερεπίπεδα) που δίνονται από τις σχέσεις στα αριστερά (<=).

☑ Εισαγωγή
★ Πολυεπίπεδες Perceptron
☐ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

Ταξινόμηση από τη μηχανή Perceptron δύο επιπέδων



- Η αντιστοίχιση που πραγματοποιείται στο πρώτο επίπεδο μετασχηματίζει τα δεδομένα εισόδου στις κορυφές ενός τετραγώνου μοναδιαίας πλευράς, δηλαδή στα σημεία (0, 0), (0,1), (1,0), (1,1).
- Στη γενική περίπτωση (βλέπε σχήμα) έχουμε:

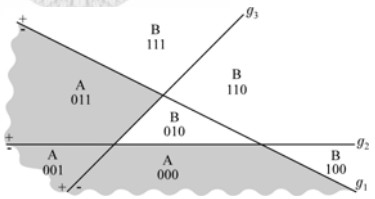
  $\underline{x} \in R^l$

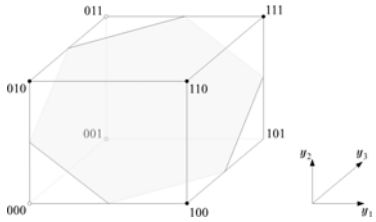
  $\underline{x} \rightarrow \underline{y} = [y_1, \ldots y_p]^T,\ y_i \in \{0,1\}\ i = 1, 2, \ldots p$

  και η αντιστοίχιση του διανύσματος $\underline{x}$ γίνεται σε μια από τις κορυφές ενός υπερκύβου H_p.
- Η αντιστοίχιση αυτή επιτυγχάνεται από τα p υπερεπίπεδα που ορίζουν οι εσωτερικοί νευρώνες. Η έξοδος καθενός από αυτούς τους νευρώνες είναι είτε μηδέν (0) είτε ένα (1) ανάλογα με τη σχετική θέση του $\underline{x}$ ως προς το υπερεπίπεδο.

☑ Εισαγωγή
★ Πολυεπίπεδες Perceptron
☐ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

Ταξινόμηση από τη μηχανή Perceptron δύο επιπέδων (II)

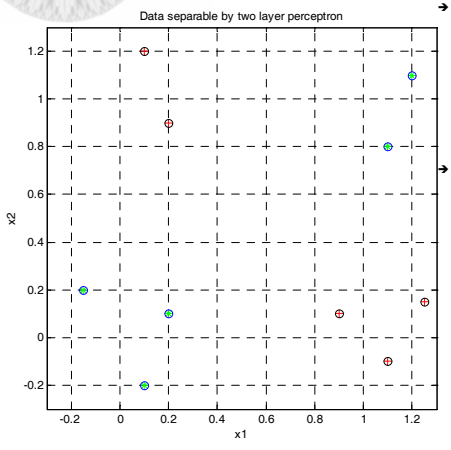




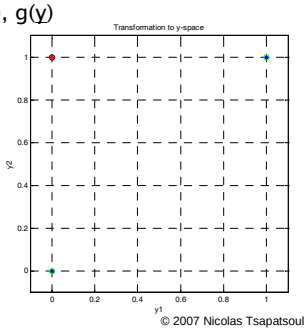
- Οι τομές των υπερεπιπέδων ορίζουν περιοχές (βλέπε σχήμα) στον /–διάστατο χώρο. Κάθε περιοχή αντιστοιχεί σε μια κορυφή του υπερκύβου.
  - Για παράδειγμα η περιοχή 001 βρίσκεται:
    στην αρνητική (–) πλευρά της $g_1(\underline{x})=0$
    στην αρνητική (–) πλευρά της $g_2(\underline{x})=0$
    στη θετική (+) πλευρά της $g_3(\underline{x})=0$
- Ο νευρώνας εξόδου υλοποιεί ένα υπερεπίπεδο στο μετασχηματισμένο χώρο $\underline{y}$, το οποίο διαχωρίζεις κάποιες κορυφές από κάποιες άλλες.
- Συγκεκριμένα μπορεί να διαχωρίσει περιοχές των οποίων η ένωση ορίζει μια συνεχή περιοχή αλλά όχι κάθε συνδυασμό περιοχών

☑ Εισαγωγή
★ Πολυεπίπεδες Perceptron
☐ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

Παράδειγμα



- Για τα δεδομένα του σχήματος στα αριστερά (κόκκινα κλάση ω_1, πράσινα κλάση ω_2) να βρείτε μια μηχανή Perceptron δύο επιπέδων που να τα ταξινομεί σωστά.
- Ζητούμενα:
  - $g_1(\underline{x})$, $g_2(\underline{x})$, $g(\underline{y})$
  - $f(\cdot)$



☑ Εισαγωγή
★ Πολυεπίπεδες Perceptron
☐ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

Παράδειγμα ΙΙ

- Οι παρακάτω γραμμές διαχωρίζουν τον διδιάστατο χώρο σε επτά περιοχές:

$$g_1(\underline{x}) = x_1 + x_2 = 0$$

$$g_2(\underline{x}) = x_2 - \frac{1}{4} = 0$$

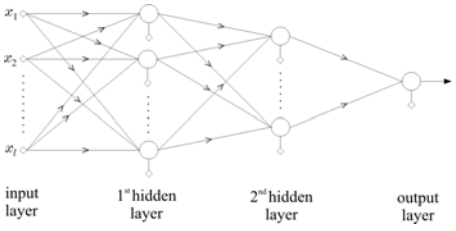
$$g_3(\underline{x}) = x_1 - x_2 = 0$$

1. Να σχηματίσετε τις ανωτέρω περιοχές
2. Να βρείτε σε ποιά κορυφή του κύβου αντιστοιχείται κάθε περιοχή
3. Αν οι περιοχές 110 και 001 σχηματίζουν την κλάση ω_1 και οι υπόλοιπες την κλάση ω_2 να ελέγξετε κατά πόσο το ανωτέρω πρόβλημα ταξινόμησης είναι επιλύσιμο με μια μηχανή perceptron δύο επιπέδων. Αν είναι βρείτε την επιφάνεια διαχωρισμού των κλάσεων (δηλαδή τη συνάρτηση g(y))

☑ Εισαγωγή
★ Πολυεπίπεδες Perceptron
☐ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

Μηχανή Perceptron τριών επιπέδων

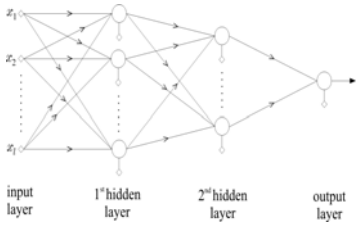
- Η αρχιτεκτονική της μηχανής Perceptron τριών επιπέδων φαίνεται στο σχήμα.
  - Η μηχανή Perceptron τριών επιπέδων μπορεί να διαχωρίσει οποιοδήποτε συνδυασμό περιοχών στο μετασχηματισμένο χώρο y.
  - Η βασική ιδέα είναι παρόμοια με αυτή του XOR προβλήματος: Χρησιμοποιούμε περισσότερα από ένα υπερεπίπεδα για να διαχωρίσουμε το χώρο $\underline{y} \in R^p$
  - Τα υπερεπίπεδα αυτά συνδυάζονται στο επίπεδο εξόδου για τον τελικό διαχωρισμό σε δύο κλάσεις.



☑ Εισαγωγή
★ Πολυεπίπεδες Perceptron
☐ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

Μηχανή Perceptron τριών επιπέδων (II)

- Η μηχανή Perceptron τριών επιπέδων μπορεί να διαχωρίσει οποιοδήποτε συνδυασμό περιοχών διότι:
  - Για κάθε κορυφή του υπερκύβου που ανήκει στην κλάση ω_1 μπορούμε να ορίσουμε ένα υπερεπίπεδο που διαχωρίζει αυτή την περιοχή (+) από όλες τις άλλες (-).
  - Το επίπεδο εξόδου υλοποιεί μια συνάρτηση OR (όλων των κορυφών που ανήκουν στην κλάση ω_1).



- Συνοπτικά:
  - Το πρώτο κρυφό επίπεδο δημιουργεί τις υπερεπιφάνειες διαχωρισμού
  - Το δεύτερο σχηματίζει τις περιοχές
  - Το επίπεδο εξόδου σχηματίζει τις κλάσεις

☑ Εισαγωγή
★ Πολυεπίπεδες Perceptron
☐ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

Παράδειγμα III

- Οι παρακάτω γραμμές διαχωρίζουν τον διδιάστατο χώρο σε επτά περιοχές:

$$g_1(\underline{x}) = x_1 + x_2 = 0$$

$$g_2(\underline{x}) = x_2 - \frac{1}{4} = 0$$

$$g_3(\underline{x}) = x_1 - x_2 = 0$$

1. Αν οι περιοχές 110 και 001 σχηματίζουν την κλάση ω_1 και οι υπόλοιπες την κλάση ω_2 να βρείτε τις υπερεπιφάνειες διαχωρισμού των κλάσεων (δηλαδή τις συνάρτησεις $q_1(\underline{y})$, $q_2(y)$) καθώς και την συνάρτηση εξόδου (OR)

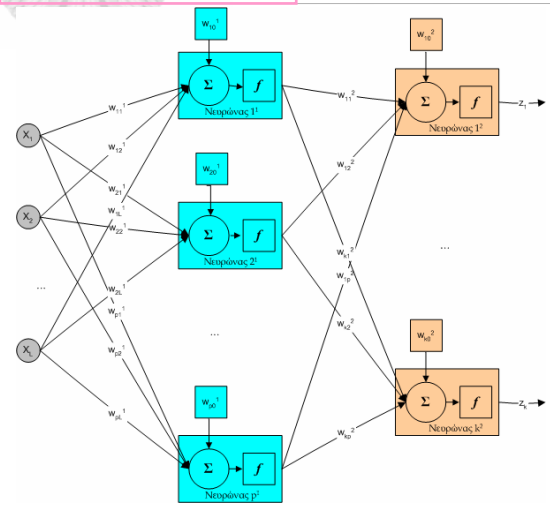
☑ Εισαγωγή
★ Πολυεπίπεδες Perceptron
☐ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

Σχεδίαση πολυεπίπεδων μηχανών Perceptron

- Με τον όρο σχεδίαση αναφερόμαστε στην εύρεση, με συστηματικό τρόπο, των συναρτήσεων αντιστοίχισης (πρώτο επίπεδο), σχηματισμού περιοχών (δεύτερο επίπεδο), και διαχωρισμού κλάσεων (επίπεδο εξόδου) από ένα σύνολο διανυσμάτων εκπαίδευσης $[\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N]$ για τα οποία είναι γνωστές οι κλάσεις στις οποίες ανήκουν.
- Υπάρχουν δύο βασικές κατευθύνσεις όσον αφορά τη σχεδίαση πολυεπίπεδων μηχανών Perceptron:
  - Κατασκευή perceptron τριών επιπέδων για ορθή ταξινόμηση όλων των διανυσμάτων εκπαίδευσης με χρήση της μεθοδολογίας που περιγράφηκε στις προηγούμενες διαφάνειες. Προφανώς σε αυτή την περίπτωση η δομή του δικτύου Perceptron δεν είναι γνωστή και πρέπει να προκύψει από τη διαδικασία μάθησης.
  - Με δεδομένη μια δομή για το δίκτυο Perceptron υπολογίζουμε τα συναπτικά βάρη w_{ij}, w_{i0} για την ελαχιστοποίηση ενός κριτηρίου κόστους. Η μεθοδολογία αυτή δεν οδηγεί πάντα σε ορθή ταξινόμηση όλων των διανυσμάτων εκπαίδευσης.

☑ Εισαγωγή
☑ Πολυεπίπεδες Perceptron
★ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

Ο αλγόριθμος back-propagation



- Ο αλγόριθμος back-propagation εφαρμόζεται σε πολυεπίπεδα δίκτυα Perceptron όπως αυτό του σχήματος
- Συμβολισμοί:
  - Είσοδος =>

    Διάνυσμα $x = [x_1\ x_2\ \ldots x_L]^T$
  - Έξοδος => $z = [z_1\ z_2\ \ldots\ z_k]^T$
  - Νευρώνας i^m => i-στός νευρώνας m-στού επιπέδου
  - Συναπτικά βάρη w_{ij}^m => Βάρος που συνδέει τον i-στο νευρώνα του m-στού επιπέδου με τον j-στο νευρώνα του αμέσως προηγούμενου επιπέδου

☑ Εισαγωγή
☑ Πολυεπίπεδες Perceptron
★ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

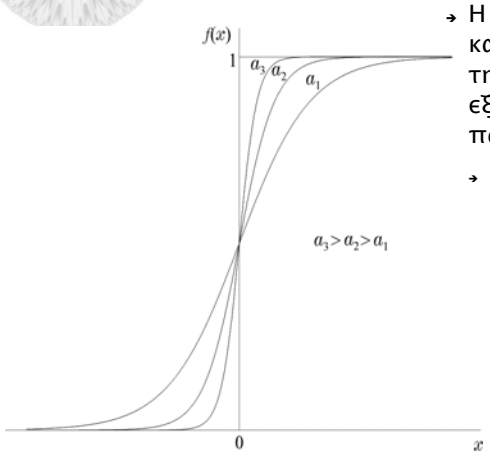
Ο αλγόριθμος back-propagation (II)

- Ο αλγόριθμος back-propagation είναι μια επαναληπτική διαδικασία υπολογισμού των συναπτικών βαρών $[w_{ik}^j\ w_{i0}^j]$ σε ένα πολυεπίπεδο δίκτυο Perceptron με τη βοήθεια των δεδομένων εκπαίδευσης $(\underline{x}_q, \underline{d}_q)$, q = 1,2, …,N, όπου:
- $\underline{x}_q = [\underline{x}_{q1}\ \underline{x}_{q2}\ \ldots \underline{x}_{qL}]^T$ το q-στο διάνυσμα εισόδου διάστασης L ($\underline{x}_q \in R^L$)
- $\underline{d}_q$: η επιθυμητή έξοδος του διανύσματος $\underline{x}_q$, μέσω της οποίας παράγεται η τελική κλάση στην οποία ανήκει (1 => κλάση ω_1, 0=> κλάση ω_2)
- Επειδή στον αλγόριθμο back-propagation βελτιστοποιείται ένα κριτήριο κόστους (και όχι στο σφάλμα ταξινόμησης) είναι επιθυμητό οι συναρτήσεις ενεργοποίησης $f(\cdot)$ των νευρώνων να έχουν συνεχή μορφή ώστε να είναι παραγωγίσιμες:
  - Η συνάρτηση ενεργοποίησης της μηχανής Perceptron δεν είναι αποδεκτή $f(x) = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \end{cases}$
  - Αντίθετα η σιγμοειδής συνάρτηση: $f(x) = \dfrac{1}{1+e^{-ax}}$

  έχει παρόμοια συμπεριφορά αλλά επιπλέον είναι και παραγωγίσιμη

☑ Εισαγωγή
☑ Πολυεπίπεδες Perceptron
★ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

Μορφή σιγμοειδούς συνάρτησης



- Η σιγμοειδής συνάρτηση με κατάλληλη τιμή του α προσεγγίζει την βηματική συνάρτηση αλλά εξακολουθεί να είναι παραγωγίσιμη.
  - Η παράγωγος της σιγμοειδούς συνάρτησης δίνεται από τη σχέση:

$$f(x) = \frac{1}{1+e^{-ax}}$$

$$f'(x) = \frac{df(x)}{dx} = \frac{ae^{-ax}}{1+e^{-ax}} = af(x)(1-f(x))$$

☑ Εισαγωγή
☑ Πολυεπίπεδες Perceptron
★ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

Υπολογισμός συναπτικών βαρών

- Βήματα:
  - Ορισμός μιας συνάρτησης κόστους J. Η συνηθέστερη επιλογή είναι η συνάρτηση σφάλματος ελαχίστων τετραγώνων (least squares error).
    - Το σφάλμα ορίζεται ως η διαφορά ανάμεσα στο επιθυμητό διάνυσμα εξόδου $\underline{d}_q$ του διανύσματος $\underline{x}_q$ και την πραγματική έξοδο $\underline{z}_q$ που δίνει το συγκεκριμένο διάνυσμα:

$$\underline{e}_q = \underline{d}_q - \underline{z}_q$$

$$J = \frac{1}{2}\sum_{q=1}^{N} \underline{e}_q^T \underline{e}_q$$

  - Ορισμός μιας επαναληπτικής διαδικασίας βελτιστοποίησης (ελαχιστοποίησης) της συνάρτησης κόστους ως προς τα συναπτικά βάρη w_{ij}^m. Η συνηθέστερη επιλογή είναι η χρήση της μεθόδου κατάβασης κατά τη μέγιστη κλίση (Gradient descent). Εναλλακτικές επιλογές:
    - Αλγόριθμος Newton
    - Συζυγής κλίση (Conjugate gradient)

☑ Εισαγωγή
☑ Πολυεπίπεδες Perceptron
★ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

Υπολογισμός συναπτικών βαρών (II)

- Η διαδικασία εύρεσης των συναπτικών βαρών είναι ένα μη γραμμικό πρόβλημα βελτιστοποίησης εξαιτίας της συνάρτησης ενεργοποίησης $f(\cdot)$ η οποία στις περισσότερες περιπτώσεις είναι μη γραμμική (π.χ. η σιγμοειδής συνάρτηση).
  - Στην περίπτωση της μεθόδου κατάβασης κατά τη μέγιστη κλίση η επαναληπτική διαδικασία υπολογισμού των συναπτικών βαρών περιγράφεται από τις σχέσεις:

$$\underline{w}_1^r(\text{new}) = \underline{w}_1^r(\text{old}) + \Delta\underline{w}_1^r$$

$$\Delta\underline{w}_1^r = -\mu\frac{\partial J}{\partial w_1^r}$$

☑ Εισαγωγή
☑ Πολυεπίπεδες Perceptron
★ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

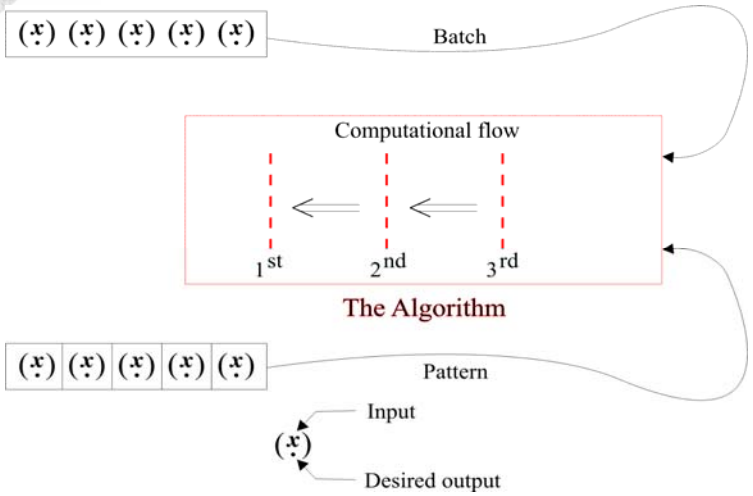
Υπολογισμός των βαρών στον back-propagation

- Διαδικασία:
  1. Αρχικοποίηση των συναπτικών βαρών με μικρές τυχαίες τιμές
  2. Υπολογισμός της κλίσης της συνάρτησης κόστους σε σχέση με τα συναπτικά βάρη από το τελευταίο επίπεδο (επίπεδο εξόδου) προς τα πίσω
  3. Διόρθωση των βαρών με μεταβολή αντίθετα προς την κλίση της συνάρτησης κόστους
  4. Επανάληψη των βημάτων (2)–(3) μέχρι την ικανοποίηση ενός κριτηρίου τερματισμού (π.χ. πολύ μικρή μεταβολή των βαρών)

- Υπάρχουν δύο διαφορετικές προσεγγίσεις ως προς την διόρθωση των βαρών
  1. Διόρθωση μετά από τον υπολογισμό της εξόδου κάθε προτύπου (Pattern mode)
  2. Διόρθωση μετά από την εφαρμογή του συνόλου των δεδομένων εκπαίδευσης – προτύπων (Batch mode)

☑ Εισαγωγή
☑ Πολυεπίπεδες Perceptron
★ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

Σχηματική αναπαράσταση του back-propagation



☑ Εισαγωγή
☑ Πολυεπίπεδες Perceptron
★ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

Εξισώσεις διόρθωσης των βαρών

- Έστω ότι έχουμε κριτήριο κόστους το τετραγωνικό σφάλμα:

$$J = \frac{1}{2}\underline{e}^T\underline{e} \qquad \underline{e} = \underline{d} - \underline{z}$$

όπου $\underline{d}$ η επιθυμητή έξοδος για το διάνυσμα εισόδου $\underline{x}$ και $\underline{z}$ η έξοδος του δικτύου υπολογισμένη με τα τρέχοντα βάρη.

- Η διόρθωση των βαρών στο δεύτερο επίπεδο δίνεται από τις σχέσεις:

$$\underline{w}_i^{(2)}(new) = \underline{w}_i^{(2)}(old) + \Delta\underline{w}_i^{(2)}$$

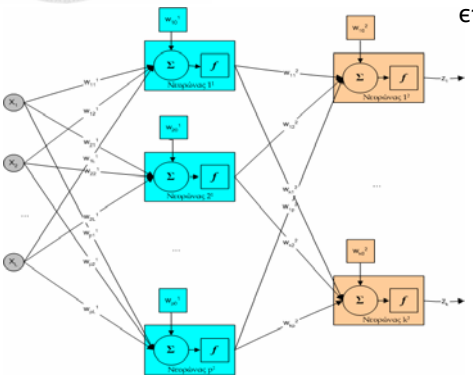
$$\Delta\underline{w}_i^{(2)} = \mu\cdot\underline{y}\cdot\delta_i, \qquad i = 1,2,\ldots,k$$

$$\delta_i = f'\left(\underline{w}_i^{(2)T}\cdot\underline{y} + \underline{w}_{i0}^{(2)}\right)\cdot(d_i - z_i)$$

όπου $0<\mu<1$ είναι το βήμα προσαρμογής, $f'(\cdot)$ είναι η παράγωγος της συνάρτησης ενεργοποίησης, $\underline{y}$ είναι το διάνυσμα των εξόδων του πρώτου επιπέδου, και $\underline{w}_i^{(2)}$ είναι το διάνυσμα συναπτικών βαρών που εισέρχονται στον i-στο νευρώνα του δευτέρου επιπέδου.

☑ Εισαγωγή
☑ Πολυεπίπεδες Perceptron
★ Ο αλγόριθμος back-propagation
☐ Επιλογή παραμέτρων
☐ Παραδείγματα

Εξισώσεις διόρθωσης των βαρών (II)



- Η διόρθωση των βαρών στο πρώτο επίπεδο δίνεται από τις σχέσεις:

$$\underline{w}_j^{(1)}(new) = \underline{w}_j^{(1)}(old) + \Delta\underline{w}_j^{(1)}, \qquad j = 1,2,\ldots,p$$

$$\Delta\underline{w}_j^{(1)} = \mu\cdot\left(\underline{u}_j^T\cdot\underline{\delta}\right)\cdot f'\left(\underline{w}_i^{(1)T}\cdot\underline{x} + \underline{w}_{j0}^{(1)}\right)\cdot\underline{x}$$

$$\underline{\delta} = \begin{bmatrix}\delta_1\\ \delta_2\\ \ldots\\ \delta_k\end{bmatrix}, \quad \underline{u}_j = \begin{bmatrix}w_{ij}^{(2)}\\ w_{2j}^{(2)}\\ \ldots\\ w_{kj}^{(2)}\end{bmatrix}$$

όπου $\underline{x}$ είναι το πρότυπο εισόδου, $\underline{w}_j^{(1)}$ είναι το διάνυσμα συναπτικών βαρών που εισέρχονται στον j-στο νευρώνα του πρώτου επιπέδου, και $\underline{u}_j$ είναι το διάνυσμα συναπτικών βαρών που εξέρχονται από τον j-στο νευρώνα του πρώτου επιπέδου

☑ Εισαγωγή
☑ Πολυεπίπεδες Perceptron
☑ Ο αλγόριθμος back-propagation
★ Επιλογή παραμέτρων
☐ Παραδείγματα

Επιλογή Παραμέτρων

- Η εφαρμογή του αλγορίθμου back-propagation απαιτεί τη ρύθμιση κάποιων παραμέτρων εξαρχής:
  - Κριτήριο κόστους
    - Μέσο τετραγωνικό σφάλμα (Least Mean Squares Error –LMS Error):

$$J = \sum_{i=1}^{N} E(i) \qquad E(i) = \sum_{m=1}^{k} e_m^2(i) = \sum_{m=1}^{k} (d_m(i) - z_m(i))^2$$
$$i = 1,2,\ldots,N$$

όπου $d_m(i)$ είναι η επιθυμητή έξοδος από τον m νευρώνα εξόδου για το πρότυπο i (μπορεί να είναι 0 ή 1) και $z_m(i)$ η πραγματική έξοδος (στον ίδιο νευρώνα και για το ίδιο πρότυπο) η οποία κυμαίνεται στο διάστημα [0 1]

    - Διασταυρούμενη εντροπία (Cross Entropy)

$$J = \sum_{i=1}^{N} E(i) \qquad E(i) = \sum_{m=1}^{k} \{d_m(i) \ln z_m(i) + (1 - d_m(i)) \ln(1 - z_m(i))\}$$

☑ Εισαγωγή
☑ Πολυεπίπεδες Perceptron
☑ Ο αλγόριθμος back-propagation
★ Επιλογή παραμέτρων
☐ Παραδείγματα

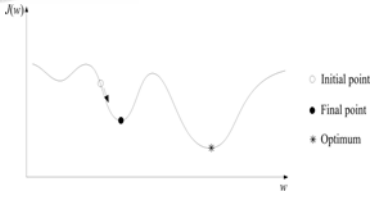
Επιλογή Παραμέτρων (ΙΙ)

- Μέγεθος δικτύου Perceptron
  - Συνήθως επιλέγεται δίκτυο με ένα μόνο κρυφό (hidden) επίπεδο. Οι νευρώνες εξόδου καθορίζονται από τον αριθμό των κλάσεων στα οποία θα ταξινομήσουμε τα δεδομένα. Επομένως η επιλογή αφορά το πλήθος των νευρώνων στο κρυφό επίπεδο (p). Υπάρχουν δύο βασικές προσεγγίσεις:
    - Μέθοδοι κλαδέματος (pruning techniques). Ξεκινάμε από ένα δίκτυο με πολλούς κόμβους και σταδιακά αφαιρούμε τους πιο ανενεργούς με χρήση διαφόρων κριτηρίων
    - Μέθοδοι σύνθεσης (constructive techniques). Ξεκινάμε από ένα δίκτυ με λίγους κόμβους και προσθέτουμε νέους με στόχο τη βελτίωση της ταξινόμησης
  - Γενικά πολλοί κόμβοι στο εσωτερικό επίπεδο οδηγούν σε εύκολη μάθηση αλλά σε πολλές περιπτώσεις φτωχή γενικευτική ικανότητα. Αντίθετα λίγοι κόμβοι μπορεί να μας οδηγήσουν σε ένα τοπικό ελάχιστο της συνάρτησης κόστους (μη επαρκής λύση). Έχουν όμως καλύτερη γενικευτική ικανότητα

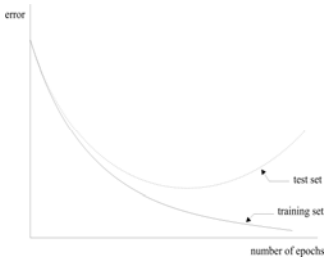
☑ Εισαγωγή
☑ Πολυεπίπεδες Perceptron
☑ Ο αλγόριθμος back-propagation
★ Επιλογή παραμέτρων
☐ Παραδείγματα

Προβλήματα του back-propagation

1. Πιθανή σύγκλιση σε τοπικό ελάχιστο:



2. Φτωχή γενίκευση



☑ Εισαγωγή
☑ Πολυεπίπεδες Perceptron
☑ Ο αλγόριθμος back-propagation
☑ Επιλογή παραμέτρων
★ Παραδείγματα

Παραδείγματα

- Κατά την εφαρμογή του αλγορίθμου back-propagation στην m-στη επανάληψη έχουμε:
  - Είσοδο το πρότυπο $\underline{x} = [2\ 3]^T$ το οποίο έχει επιθυμητή έξοδο $\underline{d} = [0\ 1]^T$
  - Πίνακα βαρών στο πρώτο επίπεδο $W_1 = [\underline{w}_1^{(1)}\ \underline{w}_2^{(1)} \ldots \underline{w}_p^{(1)}]$ (η τελευταία γραμμή του πίνακα W_1 αντιστοιχεί στα κατώφλια $w_{i0}^{(1)}$ των κόμβων):

$$W_1 = \begin{bmatrix} -0.6 & 0.1 & -0.1 & -1.3 \\ 2.2 & 1.1 & -0.8 & 0.7 \\ 0.1 & 0.1 & 0.3 & 1.6 \end{bmatrix}$$

  - Πίνακα βαρών στο δεύτερο επίπεδο $W_2 = [\underline{w}_1^{(2)}\ \underline{w}_2^{(2)} \ldots \underline{w}_k^{(2)}]$ (η τελευταία γραμμή του πίνακα W_2 αντιστοιχεί στα κατώφλια $w_{i0}^{(2)}$ των κόμβων):

$$W_2 = \begin{bmatrix} -0.7 & 0.6 \\ 0.9 & -0.4 \\ 1.3 & 0.7 \\ -1.6 & 0.8 \\ -1.4 & 0.7 \end{bmatrix}$$

☑ Εισαγωγή
☑ Πολυεπίπεδες Perceptron
☑ Ο αλγόριθμος back-propagation
☑ Επιλογή παραμέτρων
★ Παραδείγματα

Παραδείγματα (II)

- Ζητούμενα:
  1. Να σχηματίσετε το δίκτυο
  2. Να υπολογίσετε την έξοδο y του κρυφού επιπέδου για το πρότυπο x.
  3. Να υπολογίσετε την έξοδο z του δικτύου για το πρότυπο x
  4. Να εφαρμόσετε τον αλγόριθμο back-propagation για τον υπολογισμό των πινάκων W_1, W_2 στην m+1 επανάληψη (μετά την εφαρμογή της εισόδου x)

- Θεωρήστε ότι όλοι οι κόμβοι έχουν συνάρτηση ενεργοποίησης:

$$f(x) = \frac{1}{1+e^{-x}}$$

για την οποία ισχύει: $f'(x) = f(x)(1-f(x))$