

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΑΝΑΓΝΩΡΙΣΗΣ ΠΡΟΤΥΠΩΝ**

**ΣΥΜΠΛΗΡΩΜΑΤΙΚΕΣ ΣΗΜΕΙΩΣΕΙΣ
ΓΙΑ ΤΟ ΜΑΘΗΜΑ
“ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ 1”**

**ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ HOPFIELD
ΚΑΙ KOHONEN**

Σπυρίδων Λυκοθανάσης
Καθηγητής,
Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Πανεπιστημίου Πατρών

ΠΑΤΡΑ 2007

5. Δίκτυα Hopfield

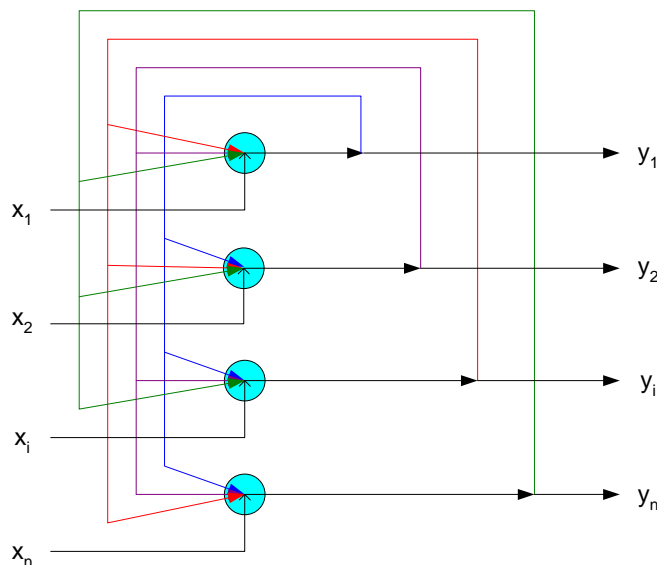
Τα δίκτυα Hopfield ανήκουν στην κατηγορία των αναδρομικών Νευρωνικά Δικτύων. Ένα αναδρομικό Νευρωνικό Δίκτυο έχει εκτός από τις προς τα εμπρός και προς τα πίσω συνδέσεις. Έχει δηλαδή βρόγχους αναδρομής από τις εξόδους του προς τις εισόδους του. Η παρουσία τέτοιων βρόγχων έχει ισχυρή επίπτωση στην ικανότητα μάθησης του δικτύου.

Ένα αναδρομικό Νευρωνικό Δίκτυο, λοιπόν, μαθαίνει ως εξής: αφού εφαρμοστεί μια είσοδος υπολογίζεται η έξοδος του δικτύου η οποία στη συνέχεια ανατροφοδοτείται ως είσοδος στο δίκτυο. Υπολογίζεται η νέα έξοδος του δικτύου και η διαδικασία επαναλαμβάνεται έως ότου η έξοδος του δικτύου γίνει σταθερή (πάψει να μεταβάλλεται).

Βέβαια δεν γίνεται πάντα η έξοδος του δικτύου σταθερή. Η παραπάνω διαδικασία που περιγράψαμε δεν εξασφαλίζει πάντα ότι σε κάθε επανάληψη οι μεταβολές στην έξοδο του δικτύου θα είναι ολοένα και μικρότερες έτσι ώστε σε κάποια χρονική στιγμή η έξοδος να πάψει να μεταβάλλεται. Αντιθέτως, είναι πολύ πιθανόν να οδηγήσει σε μια χαοτική συμπεριφορά του δικτύου. Στην περίπτωση αυτή η έξοδος του δικτύου δεν γίνεται ποτέ σταθερή και τότε λέμε ότι το δίκτυο είναι ασταθές.

Η ευστάθεια των αναδρομικών Νευρωνικών Δικτύων αποτέλεσε αντικείμενο έρευνας πολλών ερευνητών του χώρου στις δεκαετίες του 1960 και 1970. Παρόλα αυτά, κανείς δεν κατάφερε να προβλέψει πιο αναδρομικό δίκτυο θα μπορούσε να ήταν ευσταθές, με αποτέλεσμα πολλοί από αυτούς να εκφράζουν μια απαισιοδοξία για το εάν τελικά θα μπορούσε να βρεθεί μια λύση σε αυτό το πρόβλημα. Η λύση όμως ήρθε το 1982, όταν ο John Hopfield διατύπωσε τη φυσική αρχή της αποθήκευσης πληροφορίας σε ένα δυναμικά ευσταθές δίκτυο [Hopfield, 1982].

Στο σχήμα 5.1 απεικονίζεται ένα δίκτυο Hopfield ενός επιπέδου το οποίο αποτελείται από n υπολογιστικούς νευρώνες. Η έξοδος κάθε νευρώνα ανατροφοδοτείται ως είσοδος σε όλους τους υπόλοιπους νευρώνες (στα δίκτυα Hopfield δεν υπάρχει αυτό-ανατροφοδότηση). Οι υπολογιστικοί νευρώνες που χρησιμοποιούνται ακολουθούν συνήθως το μοντέλο McCulloch – Pitts με συνάρτηση ενεργοποίησης την συνάρτηση προσήμου (sign function).



Σχήμα 5.1 Ένα Δίκτυο Hopfield ενός επιπέδου με n νευρώνες

Η λειτουργία ενός νευρώνα k που χρησιμοποιεί τη συνάρτηση προσήμου περιγράφεται από τις ακόλουθες σχέσεις:

$$v_k = \sum_{j=0}^m w_{kj} x_j \quad (5.1)$$

$$y_k = \varphi(v_k) = \begin{cases} +1, & v_k > 0 \\ -1, & v_k < 0 \\ y_k, & v_k = 0 \end{cases} \quad (5.2)$$

Είσοδοι

Δηλαδή, η έξοδος του νευρώνα γίνεται +1 (ο νευρώνας μεταβαίνει σε κατάσταση +1) εάν το δυναμικό ενεργοποίησης v_k του νευρώνα είναι μεγαλύτερο του 0, -1 (μετάβαση σε κατάσταση -1) εάν το v_k είναι μικρότερο του μηδέν και παραμένει αμετάβλητη (ο νευρώνας παραμένει στην προηγούμενή του κατάσταση) εάν το v_k είναι ίσο με το μηδέν.

Μια άλλη συνάρτηση ενεργοποίησης που μπορεί να χρησιμοποιηθεί αντί της συνάρτησης προσήμου είναι η κορεσμένη γραμμική συνάρτηση (saturated linear function), η οποία συμπεριφέρεται ως καθαρά γραμμική συνάρτηση στο διάστημα (-1, +1) και ως συνάρτηση προσήμου οπουδήποτε αλλού. Στην περίπτωση αυτή η έξοδος του δικτύου δίνεται από τη σχέση:

$$y_k = \varphi(v_k) = \begin{cases} +1, & v_k \geq +1 \\ -1, & v_k \leq -1 \\ v_k, & -1 < v_k < +1 \end{cases} \quad (5.3)$$

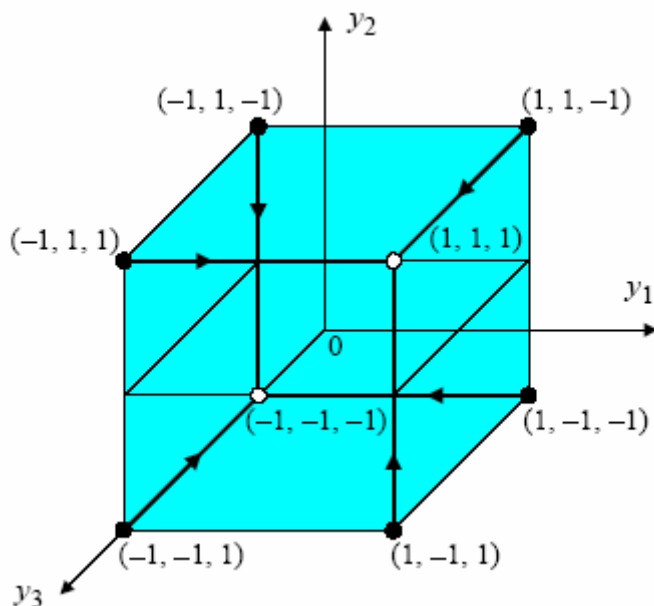
Η τρέχουσα κατάσταση του δικτύου καθορίζεται από τις τρέχουσες εξόδους όλων των νευρώνων y_1, y_2, \dots, y_n του δικτύου. Έτσι, για την περίπτωση ενός δικτύου Hopfield ενός επιπέδου που αποτελείται από n νευρώνες η κατάσταση περιγράφεται από το ακόλουθο διάνυσμα κατάστασης (state vector):

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \mathbf{M} \\ y_n \end{bmatrix} \quad (5.4)$$

Στα δίκτυα Hopfield τα βάρη μεταξύ των νευρώνων συνηθίζεται να αναπαριστώνται με τη μορφή πινάκων. Έτσι, ο πίνακας των βαρών ενός δικτύου Hopfield υπολογίζεται ως εξής:

$$\mathbf{W} = \sum_{m=1}^M \mathbf{Y}_m \times \mathbf{Y}_m^T - \mathbf{M} \times \mathbf{I}, \quad (5.5)$$

Όπου M είναι ο αριθμός των καταστάσεων που θα απομνημονευθούν από το δίκτυο, \mathbf{Y}_m είναι το n -διάστατο δυαδικό διάνυσμα και \mathbf{I} είναι ο $(n \times n)$ ταυτοτικός πίνακας (Identity Matrix).



Σχήμα 5.2 Αναπαράσταση των δυνατών καταστάσεων ενός δικτύου Hopfield με τρεις νευρώνες

Η λειτουργία ενός δικτύου Hopfield μπορεί να αναπαρασταθεί γεωμετρικά. Στο σχήμα 5.2 απεικονίζονται οι δυνατές καταστάσεις που μπορεί να πάρει ένα δίκτυο με 3 νευρώνες και οι οποίες αναπαρίστανται στον τρισδιάστατο χώρο ως κύβος. Γενικά, ένα

δίκτυο Hopfield με n νευρώνες μπορεί να βρεθεί σε 2^n δυνατές καταστάσεις και να αναπαρασταθεί με ένα n -διάστατο υπέρ-κύβο. Όπως μπορούμε να δούμε στο σχήμα 5.2 κάθε κατάσταση του δικτύου απεικονίζεται από μια κορυφή του υπέρ-κύβου. Όταν εφαρμόζεται ένα νέο διάνυσμα εισόδου, τότε το δίκτυο μετακινείται από μία κατάσταση – κορυφή σε μία άλλη έως ότου σταθεροποιηθεί.

Η σταθερή κατάσταση – κορυφή καθορίζεται από τον πίνακα των βαρών \mathbf{W} , το τρέχον διάνυσμα εισόδου \mathbf{X} και από τον πίνακα των κατωφλίων $\boldsymbol{\theta}$. Εάν στο δίκτυο εισάγουμε ένα διάνυσμα το οποίο είναι μερικώς λανθασμένο ή κατεστραμμένο τότε το δίκτυο θα συγκλίνει σε μία σταθερή κατάσταση μετά από έναν αριθμό επαναλήψεων. Οι σταθερές καταστάσεις πολλές φορές ονομάζονται και βασικές μνήμες (fundamental memories).

Αφού αποθηκεύσουμε τα διανύσματα που επιθυμούμε να απομνημονεύσει το δίκτυο, στη συνέχεια μπορούμε να το δοκιμάσουμε. Αυτό γίνεται ως εξής: Αρχικά ενεργοποιούμε το δίκτυο εισάγοντας το διάνυσμα εισόδου X . Στη συνέχεια υπολογίζουμε το διάνυσμα εξόδου Y .

$$\mathbf{Y}_m = \text{sign}(\mathbf{W} \times \mathbf{X}_m - \boldsymbol{\theta}), \quad m=1,2, \dots, M, \quad (5.6)$$

όπου $\boldsymbol{\theta}$ είναι ο πίνακας των κατωφλίων.

Τέλος συγκρίνουμε το αποτέλεσμα (δηλαδή το διάνυσμα εξόδου Y) με το διάνυσμα X στην είσοδο.

Στη συνέχεια θα παρουσιάσουμε μια σύνοψη του αλγορίθμου εκπαίδευσης των δικτύων Hopfield [Negnevitsky, 2002].

Σύνοψη του αλγορίθμου εκπαίδευση του δικτύου Hopfield

Βήμα 1^ο: Αποθήκευση

Έστω ότι έχουμε ένα δίκτυο Hopfield με n νευρώνες στο οποίο επιθυμούμε να αποθηκεύσουμε ένα σύνολο από M διανύσματα – σταθερές καταστάσεις (βασικές μνήμες): Y_1, Y_2, \dots, Y_M . Τα συναπτικά βάρη από τον νευρώνα i στον νευρώνα j υπολογίζονται ως εξής:

$$w_{ji} = \begin{cases} \sum_{m=1}^M y_{m,i} \cdot y_{m,j}, & i \neq j \\ 0, & i = j \end{cases}, \quad (5.7)$$

όπου $y_{m,i}$ και $y_{m,j}$ είναι τα i -οστό και j -οστό στοιχείο του διανύσματος Y_m , αντίστοιχα. Σε μορφή πίνακα τα συναπτικά βάρη μπορούν να αναπαρασταθούν ως εξής:

$$\mathbf{W} = \sum_{m=1}^M \mathbf{Y}_m \times \mathbf{Y}_m^T - \mathbf{M} \times \mathbf{I} \quad (5.8)$$

Το δίκτυο Hopfield μπορεί να αποθηκεύσει ένα σύνολο από βασικές μνήμες (διανύσματα) εάν ο πίνακας των βαρών του είναι συμμετρικός και όλα τα στοιχεία της κύριας διαγωνίου του είναι ίσα με το μηδέν. Δηλαδή ο πίνακας των βαρών του δικτύου Hopfield θα πρέπει να έχει τη μορφή:

$$\mathbf{W} = \begin{bmatrix} 0 & w_{12} & L & w_{1i} & L & w_{1n} \\ w_{21} & 0 & L & w_{2i} & L & w_{2n} \\ M & M & M & M & M & M \\ w_{i1} & w_{i2} & L & 0 & L & w_{in} \\ M & M & M & M & M & M \\ w_{n1} & w_{n2} & L & w_{ni} & L & 0 \end{bmatrix}, \quad (5.9)$$

όπου $w_{ji} = w_{ij}$.

Στο δίκτυο Hopfield τα βάρη υπολογίζονται μόνο μία φορά, κατά τη φάση της αποθήκευσης. Από τη στιγμή που υπολογιστούν δεν μεταβάλλονται και παραμένουν σταθερά.

Βήμα 2^ο: Δοκιμή

Αφού αποθηκεύσουμε τις βασικές μνήμες, στη συνέχεια, είναι απαραίτητο να επιβεβαιώσουμε ότι το δίκτυο Hopfield είναι ικανό να ανακαλέσει από τη «μνήμη» όλες τις βασικές μνήμες. Με άλλα λόγια, το δίκτυο θα πρέπει να είναι ικανό να ανακαλέσει κάθε διάνυσμα (βασική μνήμη) \mathbf{Y}_m , όταν αυτό παρουσιάζεται στην είσοδό του. Δηλαδή:

$$x_{m,i} = y_{m,i}, \quad i=1,2, \dots, n \text{ και } m=1,2, \dots, M$$

$$y_{m,j} = \text{sign} \left(\sum_{i=1}^n w_{ji} \cdot x_{m,i} - \theta_j \right), \quad (5.10)$$

όπου το $y_{m,j}$ είναι το j -στο στοιχείο του διανύσματος εξόδου \mathbf{Y}_m και το $x_{m,i}$ είναι το i -οστό στοιχείο του διανύσματος εισόδου \mathbf{X}_m . Οι παραπάνω σχέσεις σε μορφή πινάκων γίνονται:

$$\mathbf{X}_m = \mathbf{Y}_m, \quad m=1,2, \dots, M$$

$$\mathbf{Y}_m = \text{sign}(\mathbf{W} \times \mathbf{X}_m - \boldsymbol{\theta}) \quad (5.11)$$

Εάν όλες οι βασικές μνήμες ανακαλούνται τέλεια τότε μπορούμε να συνεχίσουμε με το επόμενο βήμα.

Βήμα 3^ο: Ανάκτηση

Παρουσιάζουμε ένα άγνωστο n -διάστατο διάνυσμα X στο δίκτυο και ανακαλούμε μια σταθερή κατάσταση. Συνήθως το διάνυσμα αυτό είναι μία «φθαρμένη» ή ατελής εκδοχή μιας βασικής μνήμης. Δηλαδή:

$$\mathbf{X} \neq \mathbf{Y}_m, m=1,2, \dots, M$$

a. Αρχικοποιούμε τον αλγόριθμο ανάκλησης του δικτύου Hopfield θέτοντας:

$$x_i(0) = x_i, i=1,2, \dots, n \quad (5.12)$$

και υπολογίζουμε την αρχική κατάσταση κάθε νευρώνα:

$$y_j(0) = \text{sign} \left(\sum_{i=1}^n w_{ji} \cdot x_i(0) - \theta_j \right), j=1,2, \dots, n \quad (5.13)$$

όπου $x_i(0)$ είναι το i -οστό στοιχείο του διανύσματος X στην επανάληψη $p=0$, και $y_j(0)$ είναι η κατάσταση του νευρώνα j στην επανάληψη $p=0$.

Σε μορφή πινάκων το διάνυσμα κατάστασης στην επανάληψη $p=0$ υπολογίζεται ως εξής:

$$\mathbf{Y}(0) = \text{sign}(\mathbf{W} \times \mathbf{X}(0) - \boldsymbol{\theta}) \quad (5.14)$$

όπου $\mathbf{X}(0)=\mathbf{X}$.

b. Ενημερώνουμε τα στοιχεία του διανύσματος κατάστασης $\mathbf{Y}(p)$, σύμφωνα με τον ακόλουθο κανόνα:

$$y_j(p+1) = \text{sign} \left(\sum_{i=1}^n w_{ji} \cdot x_i(p) - \theta_j \right), j=1,2, \dots, n \quad (5.15)$$

όπου $x_i(p) = y_i(p), i=1,2, \dots, n$

Σε μορφή πινάκων η σχέση γίνεται:

$$\mathbf{Y}(p+1) = \text{sign}(\mathbf{W} \times \mathbf{X}(p) - \boldsymbol{\theta}), \quad (5.16)$$

όπου $\mathbf{X}(p)=\mathbf{Y}(p)$

Οι νευρώνες προς ενημέρωση επιλέγονται ασύγχρονα, δηλαδή τυχαία και ένας κάθε φορά.

Η διαδικασία επαναλαμβάνεται έως ότου το διάνυσμα κατάστασης πάψει να μεταβάλλεται, δηλαδή έως ότου επιτευχθεί μια σταθερή κατάσταση. Η συνθήκη για την ισορροπία (ευστάθεια – σταθερότητα) του δικτύου μπορεί να ορισθεί ως εξής:

$$y_j(p+1) = y_j(p), j=1,2, \dots, n \quad (5.17)$$

Ή σε μορφή πίνακα ως εξής:

$$\mathbf{Y}(\mathbf{p}+1) = \mathbf{Y}(\mathbf{p}) \quad (5.18)$$

Περιορισμοί των δικτύων Hopfield

Το δίκτυο Hopfield θα συγκλίνει πάντα σε μια σταθερή κατάσταση (κατάσταση ισορροπίας) εάν η ανάκτηση γίνεται ασύγχρονα [Haykin, 1994]. Ωστόσο, τίποτα δεν εξασφαλίζει ότι αυτή η σταθερή κατάσταση θα είναι μία από τις βασικές μνήμες που έχουν αποθηκευτεί στο δίκτυο και ακόμα και αν είναι βασική μνήμη δεν θα είναι αναγκαστικά η πλησιέστερη προς το διάνυσμα εισόδου.

Παράδειγμα

Έστω ότι θέλουμε να αποθηκεύσουμε τις παρακάτω βασικές μνήμες σε ένα δίκτυο Hopfield με 5 νευρώνες.

$$X_1 = [+1 \ +1 \ +1 \ +1 \ +1]^T$$

$$X_2 = [+1 \ -1 \ +1 \ -1 \ +1]^T$$

$$X_3 = [-1 \ +1 \ -1 \ +1 \ -1]^T$$

Χρησιμοποιώντας την εξίσωση

$$\mathbf{W} = \sum_{m=1}^3 \mathbf{X}_m \times \mathbf{X}_m^T - \mathbf{M} \times \mathbf{I} = \mathbf{X}_1 \times \mathbf{X}_1^T + \mathbf{X}_2 \times \mathbf{X}_2^T + \mathbf{X}_3 \times \mathbf{X}_3^T - \mathbf{M} \times \mathbf{I}$$

υπολογίζουμε τον πίνακα των βαρών, που είναι:

$$W = \begin{bmatrix} 0 & -1 & 3 & -1 & 3 \\ -1 & 0 & -1 & 3 & -1 \\ 3 & -1 & 0 & -1 & 3 \\ -1 & 3 & -1 & 0 & -1 \\ 3 & -1 & 3 & -1 & 0 \end{bmatrix}$$

Έστω ότι στην είσοδο του δικτύου παρουσιάζεται το διάνυσμα:

$$X = [+1 \ +1 \ -1 \ +1 \ +1]^T,$$

Συγκρίνοντας το διάνυσμα X με τη βασική μνήμη X_1 παρατηρούμε ότι διαφέρουν μόλις σε ένα ψηφίο. Οπότε αναμένουμε ότι εισάγοντας το διάνυσμα X στο δίκτυο Hopfield αυτό θα συγκλίνει στην κατάσταση X_1 . Ωστόσο εφαρμόζοντας τον αλγόριθμο

εκπαίδευσης που περιγράψαμε παραπάνω παρατηρούμε ότι το δίκτυο ανακαλεί τη βασική μνήμη X_3 , η οποία διαφέρει σε 2 ψηφία από το X και όχι το X_1 όπως αναμενόταν.



Το παραπάνω παράδειγμα αποκάλυψε ένα από τα προβλήματα που είναι εγγενή στα δίκτυα Hopfield.

Ένα άλλο πρόβλημα που αντιμετωπίζουμε όταν εφαρμόζουμε τα δίκτυα Hopfield είναι η αποθηκευτική χωρητικότητα (storage capacity), ο μέγιστος, δηλαδή, αριθμός διανυσμάτων που μπορούν να αποθηκευτούν και να ανακληθούν χωρίς σφάλμα. Ο Hopfield έδειξε πειραματικά [Hopfield, 1982] ότι ο μέγιστος αριθμός βασικών μνημών M_{max} που μπορούν να αποθηκευτούν σε ένα αναδρομικό δίκτυο με n νευρώνες δίνεται από τη σχέση (δεν μας ενδιαφέρει αν στην ανάκληση έχουμε λάθη):

$$M_{max} = 0.15 \cdot n \quad (5.19)$$

Η μέγιστη αποθηκευτική χωρητικότητα των δικτύων Hopfield μπορεί να ορισθεί και στη βάση του ότι επιθυμούμε οι περισσότερες από τις βασικές μνήμες να μπορεί να ανακληθούν χωρίς σφάλμα [Amit, 1989; Negnevitsky, 2002]:

$$M_{max} = \frac{n}{2 \cdot \ln n} \quad (5.20)$$

Στην περίπτωση, τώρα, που επιθυμούμε όλες οι βασικές μνήμες, που έχουν αποθηκευτεί σε ένα δίκτυο Hopfield με n νευρώνες, να μπορούν να ανακληθούν τέλεια (χωρίς κανένα σφάλμα) τότε το M_{max} γίνεται [Amit, 1989; Negnevitsky, 2002]:

$$M_{max} = \frac{n}{4 \cdot \ln n} \quad (5.21)$$

Παρατηρούμε, λοιπόν, ότι η αποθηκευτική χωρητικότητα ενός δικτύου Hopfield πρέπει να κρατιέται αρκετά χαμηλά προκειμένου να μπορούμε να ανακαλέσουμε χωρίς σφάλμα τις βασικές μνήμες. Αυτό αποτελεί ένα πολύ σημαντικό περιορισμό των δικτύων Hopfield.

Από τη συζήτηση που κάναμε έως τώρα παρατηρούμε ότι ένα δίκτυο Hopfield αντιπροσωπεύει, βασικά, ένα τύπο αυτοσυσχετιστικής (auto-associative) μνήμης. Με άλλα λόγια, ένα δίκτυο Hopfield μπορεί να ανακαλέσει μια φθαρμένη ή ατελή μνήμη αλλά δεν μπορεί να συσχετίσει μια μνήμη με κάποια άλλη διαφορετική. Σε αντίθεση με τα δίκτυα Hopfield, η ανθρώπινη μνήμη είναι βασικά συσχετιστική: ένα πράγμα μπορεί να μας θυμίσει κάποιο άλλο, και αυτό κάποιο άλλο κ.ο.κ. Αυτή τη λειτουργία την επιτελεί μια άλλη κατηγορία αναδρομικών δικτύων που αποκαλείται Συσχετιστική

Μνήμη Διπλής Κατεύθυνσης (Bidirectional Associative Memory) [Kosko, 1987; 1988] και η οποία βασίζεται στη φιλοσοφία λειτουργίας των δικτύων Hopfield.

Άσκηση αυτοαξιολόγησης 1:

Έστω ότι θέλουμε να αποθηκεύσουμε τα ακόλουθα διανύσματα (βασικές μνήμες):

$$\mathbf{X}_1 = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix} \text{ και } \mathbf{X}_2 = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

σε ένα δίκτυο Hopfield. Θεωρείστε ότι τα κατώφλια είναι ίσα με το μηδέν.

- Ποια πρέπει να είναι η αρχιτεκτονική του δικτύου Hopfield που θα χρησιμοποιηθεί;
- Υπολογίστε τον πίνακα των βαρών αυτού του δικτύου.
- Εξακριβώστε αν τα δύο διανύσματα έχουν αποθηκευτεί σωστά.

Απάντηση

- Εφόσον τα διανύσματα που θέλουμε να αποθηκεύσουμε έχουν διάσταση 3, το δίκτυο Hopfield που θα χρησιμοποιηθεί θα πρέπει να αποτελείται από 3 νευρώνες.
- Ο πίνακας των βαρών του δικτύου υπολογίζεται ως εξής:

$$\mathbf{W} = \sum_{m=1}^2 \mathbf{X}_m \times \mathbf{X}_m^T - \mathbf{M} \times \mathbf{I} = \mathbf{X}_1 \times \mathbf{X}_1^T + \mathbf{X}_2 \times \mathbf{X}_2^T - 2 \times \mathbf{I} \Rightarrow$$

$$\mathbf{W} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \cdot [1 \ 1 \ 1] + \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} \cdot [-1 \ -1 \ -1] - 2 \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{bmatrix}$$

- Στη συνέχεια δοκιμάζουμε αν τα δύο διανύσματα έχουν αποθηκευτεί σωστά. Έτσι έχουμε:

$$\mathbf{Y}_1 = \mathbf{sign}(\mathbf{W} \times \mathbf{X}_1 - \boldsymbol{\theta}) = \mathbf{sign} \left(\begin{bmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{bmatrix} \cdot \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) = \mathbf{sign} \left(\begin{bmatrix} 4 \\ 4 \\ 4 \end{bmatrix} \right) = \begin{bmatrix} +1 \\ +1 \\ +1 \end{bmatrix} = \mathbf{X}_1$$

και

$$\mathbf{Y}_2 = \mathbf{sign}(\mathbf{W} \times \mathbf{X}_2 - \boldsymbol{\theta}) = \mathbf{sign} \left(\begin{bmatrix} 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) = \mathbf{sign} \left(\begin{bmatrix} -4 \\ -4 \\ -4 \end{bmatrix} \right) = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} = \mathbf{X}_2$$

6. Δίκτυα Kohonen

Οι αλγόριθμοι για τους οποίους έχουμε μιλήσει έως τώρα ανήκουν στην κατηγορία της επιβλεπόμενης μάθησης (supervised learning), δηλαδή της εκπαίδευσης με τη χρήση «δασκάλου» (για κάθε είσοδο στο σύνολο εκπαίδευσης γνωρίζουμε και την αντίστοιχη έξοδο). Στο κεφάλαιο αυτό θα αναφερθούμε σε μια κατηγορία δικτύων και σε έναν αλγόριθμο που ανήκουν στην κατηγορία της μη-επιβλεπόμενης μάθησης (un-supervised learning), δηλαδή της μάθησης χωρίς την παρουσία «δασκάλου». Αυτά τα δίκτυα είναι ιδιαίτερα χρήσιμα στην περίπτωση κατηγοριοποίησης προτύπων σε γενικές κατηγορίες (pattern clustering) όταν δεν γνωρίζουμε από πριν ποιες ακριβώς είναι αυτές οι κατηγορίες. Δηλαδή αφήνουμε το Νευρωνικό Δίκτυο να «αποφασίσει» ποιες είναι οι κατηγορίες και να αντιστοιχήσει τα δείγματα εισόδου σε κάθε μία από αυτές. Ειδικότερα, θα αναφερθούμε σε ένα συγκεκριμένο είδος μη-επιβλεπόμενης μάθησης, την ανταγωνιστική μάθηση (competitive learning). Στην ανταγωνιστική μάθηση οι νευρώνες ανταγωνίζονται μεταξύ τους για το ποιος θα ενεργοποιηθεί. Ενώ σε άλλα είδη μάθησης (για παράδειγμα στη μάθηση Hebb) μπορούν να ενεργοποιούνται ταυτόχρονα περισσότεροι του ενός νευρώνες, στην ανταγωνιστική μάθηση μόνο ένας νευρώνας μπορεί να είναι ενεργός κάθε φορά. Ο νευρώνας αυτός που «κερδίζει τον ανταγωνισμό» ονομάζεται «winner-takes-all neuron».

Η βασική ιδέα της ανταγωνιστικής μάθησης έχει τις «ρίζες» της στη δεκαετία του 1970, όμως για πρώτη φορά έτυχε της προσοχής της ακαδημαϊκής και ερευνητικής κοινότητας στα τέλη της δεκαετίας του 1980, όταν ο Teuvo Kohonen παρουσίασε μια ειδική τάξη τεχνητών Νευρωνικών Δικτύων τα οποία ονόμασε αυτό-οργανωμένους χάρτες χαρακτηριστικών (self-organizing feature maps) [Kohonen, 1989]. Τα δίκτυα αυτά βασίζονται στην ανταγωνιστική μάθηση.

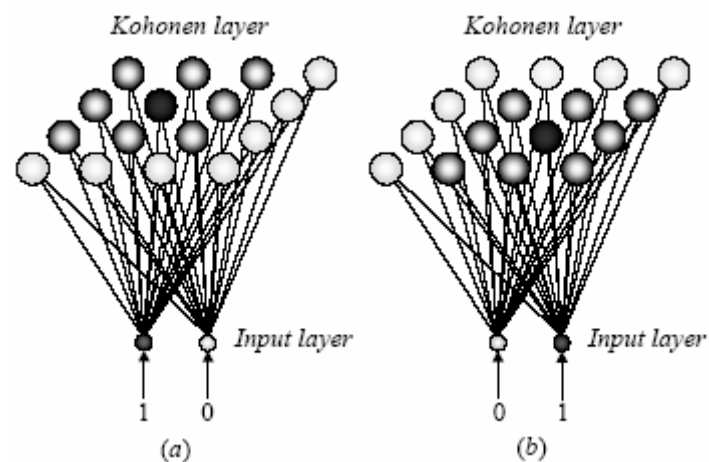
Τι είναι το Self-Organizing feature map

Ο ανθρώπινος εγκέφαλος «κυριαρχείται» από τον εγκεφαλικό φλοιό, ο οποίος είναι μια πολύ πολύπλοκη δομή από δισεκατομμύρια νευρώνες και εκατοντάδες δισεκατομμύρια συνάψεις. Ο φλοιός στην οργάνωσή του δεν παρουσιάζει ούτε ομοιομορφία ούτε και ομοιογένεια. Περιλαμβάνει περιοχές που χαρακτηρίζονται από το «πάχος» των επιπέδων τους και τον τύπο των νευρώνων που υπάρχουν σε αυτές. Κάθε μία από τις περιοχές αυτές είναι υπεύθυνη και για μια διαφορετική ανθρώπινη λειτουργία, όπως για

παράδειγμα την όραση, την κίνηση, την ακοή κ.λπ., και κάθε μια σχετίζεται και με διαφορετικά αισθητήρια όργανα. Δηλαδή, μπορούμε να πούμε ότι κάθε διαφορετικό αισθητήριο όργανο απεικονίζεται σε μια αντίστοιχη περιοχή στον εγκεφαλικό φλοιό. Με άλλα λόγια ο φλοιός είναι ένας αυτό-οργανωμένος υπολογιστικός χάρτης (self-organizing computational map) στον ανθρώπινο εγκέφαλο.

Μοντελοποίηση των αυτό-οργανωμένων χαρτών

Ο Kohonen διατύπωσε την αρχή του σχηματισμού τοπογραφικού χάρτη (principle of topographic map formation) [Kohonen, 1990]. Σύμφωνα με την αρχή αυτή η θέση ενός νευρώνα εξόδου σε ένα τοπογραφικό χάρτη αντιστοιχεί σε ένα συγκεκριμένο χαρακτηριστικό του προτύπου εισόδου. Ο Kohonen πρότεινε επίσης το μοντέλο της αντιστοίχισης των χαρακτηριστικών το οποίο φαίνεται στο σχήμα 6.1 [Kohonen, 1982]. Το μοντέλο αυτό «συλλαμβάνει» τα βασικά χαρακτηριστικά των αυτό-οργανωμένων χαρτών του εγκεφάλου, παρόλα αυτά όμως μπορεί να αναπαρασταθεί και να υλοποιηθεί εύκολα σε έναν υπολογιστή.



Σχήμα 6.1 Το δίκτυο Kohonen

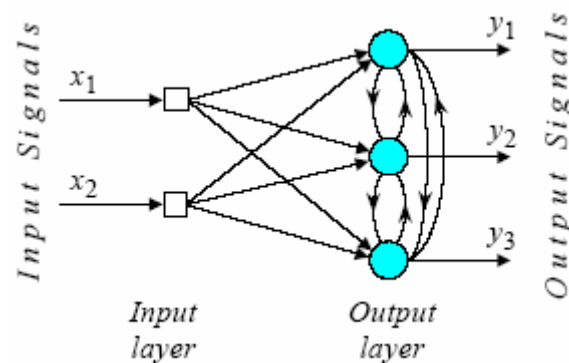
Το μοντέλο του Kohonen κάνει μια τοπογραφική απεικόνιση ενός συγκεκριμένου – σταθερού αριθμού προτύπων εισόδου (που παρουσιάζονται στο επίπεδο εισόδου) σε ένα μεγαλύτερης διάστασης επίπεδο εξόδου, το οποίο ονομάζεται και επίπεδο Kohonen. Στο σχήμα 6.1 για παράδειγμα το επίπεδο Kohonen αποτελείται από ένα πλέγμα που σχηματίζεται από 4x4 νευρώνες, με κάθε νευρώνα να έχει δύο εισόδους. Ο νικητής νευρώνας απεικονίζεται με μαύρο χρώμα ενώ οι γείτονές του με γκρι. Οι γείτονες νευρώνες του νικητή νευρώνα είναι νευρώνες που βρίσκονται σε κοντινή φυσική γειτνίαση με τον νικητή νευρώνα.

Το πόσο κοντινή είναι αυτή η φυσική γειτνίαση εξαρτάται από το σχεδιαστή του δικτύου. Έτσι η γειτονιά του νικητή νευρώνα μπορεί να περιλαμβάνει νευρώνες που βρίσκονται σε ακτίνα ενός, δύο ή τριών θέσεων από αυτόν. Στο σχήμα 6.1 για παράδειγμα η γειτονιά του νικητή νευρώνα έχει ακτίνα (ή μέγεθος) 1. Γενικά, κατά την εκπαίδευση ενός δικτύου Kohonen ξεκινάμε με μια γειτονιά μεγάλου μεγέθους (μεγάλη ακτίνα) και όσο προχωρά η διαδικασία της εκπαίδευσης το μέγεθος της γειτονιάς βαθμιαία μικραίνει. Αυτό μπορεί να γίνει με μια ποικιλία μεθόδων.

Το δίκτυο Kohonen αποτελείται από το επίπεδο εισόδου (οι νευρώνες εισόδου δεν κάνουν καμία επεξεργασία) και από ένα επίπεδο υπολογιστικών νευρώνων, το επίπεδο Kohonen. Στο δίκτυο Kohonen, επίσης, υπάρχουν δύο είδη συνάψεων – συνδέσεων:

1. Οι συνδέσεις προς τα εμπρός (forward connections) από τους νευρώνες εισόδου προς τους νευρώνες του επιπέδου Kohonen, και
2. Οι παράπλευρες συνδέσεις (*lateral connections*) μεταξύ των νευρώνων του επιπέδου εξόδου (επιπέδου Kohonen).

Αυτό απεικονίζεται στο σχήμα 6.2. Οι παράπλευρες συνδέσεις χρησιμοποιούνται για να δημιουργήσουν τον ανταγωνισμό μεταξύ των νευρώνων του επιπέδου Kohonen. Ο νευρώνας με το μεγαλύτερο δυναμικό ενεργοποίησης μεταξύ όλων των νευρώνων του επιπέδου εξόδου «στέφεται» νικητής (*winner-takes-all neuron*). Αυτό ο νευρώνας είναι ο μόνος νευρώνας που παράγει έξοδο. Η δραστηριότητα όλων των άλλων νευρώνων καταπνίγεται μέσω του ανταγωνισμού.

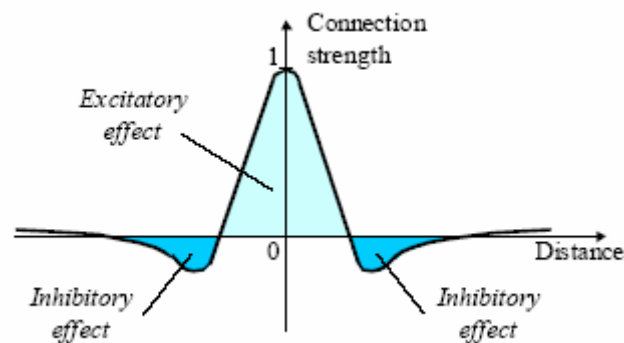


Σχήμα 6.2 Το δίκτυο Kohonen

Όταν παρουσιάζεται στις εισόδους του δικτύου ένα δείγμα, κάθε νευρώνας στο επίπεδο Kohonen λαμβάνει ένα πλήρες αντίγραφο του δείγματος, τροποποιημένο φυσικά

από τις αντίστοιχες τιμές των βαρών στις συνδέσεις μεταξύ του επιπέδου εισόδου και του επιπέδου εξόδου. Οι παράπλευρες συνδέσεις έχουν είτε διεγερτική είτε κατασταλτική δράση ανάλογα με την απόστασή τους από τον νικητή νευρώνα. Αυτό μπορεί να επιτευχθεί με τη χρήση της συνάρτησης Mexican Hat, η οποία περιγράφει τα παράπλευρα συναπτικά βάρη μεταξύ των νευρώνων στο επίπεδο Kohonen.

Η συνάρτηση Mexican Hat υπολογίζει τις τιμές των βαρών των παράπλευρων συνδέσεων συναρτήσει της απόστασης των νευρώνων από τον νικητή νευρώνα. Έτσι σύμφωνα με τη συνάρτηση αυτή η κοντινή γειτονιά γύρω από το νικητή νευρώνα έχει ισχυρή διεγερτική επίδραση, η απομακρυσμένη γειτονιά έχει μια ήπια ανασταλτική επίδραση και η πολύ απομακρυσμένη γειτονιά έχει μια αδύναμη διεγερτική επίδραση που συνήθως αγνοείται. Η συνάρτηση Mexican Hat σχηματικά αναπαρίσταται όπως φαίνεται στο σχήμα 6.3.



Σχήμα 6.3 Η συνάρτηση Mexican Hat

Στο δίκτυο Kohonen ένας νευρώνας μαθαίνει μετατοπίζοντας τα βάρη του από τις μη ενεργές συνδέσεις προς τις ενεργές. Μόνο ο νικητής νευρώνας και η γειτονιά του έχουν το δικαίωμα να μαθαίνουν. Εάν ένας νευρώνας δεν ανταποκρίνεται στο δείγμα στην είσοδο τότε δεν μπορεί να «μάθει». Η έξοδος του νικητή νευρώνα είναι ίση με το 1 και όλων των υπολοίπων με το 0.

Ο **στάνταρντ κανόνας ανταγωνιστικής μάθησης (standard competitive learning rule)** [Haykin, 1994], καθορίζει τη μεταβολή Δw_{ji} του συναπτικού βάρους w_{ji} σύμφωνα με τη σχέση:

$$\Delta w_{ji} = \begin{cases} \alpha \cdot (x_i - w_{ji}), & \text{εάν ο νευρώνας } j \text{ κερδίσει το διαγωνισμό} \\ 0, & \text{εάν ο νευρώνας } j \text{ χάσει το διαγωνισμό} \end{cases} \quad (6.1)$$

Όπου x_i είναι το σήμα στο νευρώνα εισόδου i , και το a είναι ο ρυθμός μάθησης, που παίρνει τιμές μεταξύ 0 και 1.

Η συνολική επίδραση, λοιπόν, του κανόνα της ανταγωνιστικής μάθησης είναι να «μετακινεί» το διάνυσμα των βαρών του νικητή νευρώνα w_j προς το διάνυσμα εισόδου x .

Πως όμως καθορίζεται ποιος είναι ο νικητής νευρώνας; Όταν παρουσιάζεται ένα διάνυσμα στην είσοδο του Kohonen υπολογίζεται το πόσο «ταιριάζει» το διάνυσμα των βαρών κάθε νευρώνα του επιπέδου Kohonen με το διάνυσμα αυτό. Ο νευρώνας το διάνυσμα των βαρών του οποίου «ταιριάζει» περισσότερο με το διάνυσμα εισόδου είναι ο νικητής. Η εκτίμηση του «ταιριάσματος» μπορεί να γίνει με διάφορους τρόπους, ο πιο συνηθισμένος είναι με τον υπολογισμό της *Ευκλείδειας απόστασης (Euclidean Distance)*. Στην περίπτωση αυτή δηλαδή νικητής είναι ο νευρώνας το διάνυσμα των βαρών του οποίου έχει τη μικρότερη Ευκλείδεια απόσταση από το διάνυσμα εισόδου.

Η Ευκλείδεια απόσταση μεταξύ δύο διανυσμάτων x και w_j διάστασης n δίνεται από τη σχέση:

$$d = \|x - w_j\| = \left[\sum_{i=1}^n (x_i - w_{ji})^2 \right]^{1/2} \quad (6.2)$$

όπου x_i και w_{ji} είναι τα i -οστά στοιχεία των διανυσμάτων x και w_j αντίστοιχα.

Έτσι, στην περίπτωση αυτή ο νικητής νευρώνας j_x , που ταιριάζει καλύτερα με το διάνυσμα εισόδου x είναι αυτός με τη μικρότερη Ευκλείδεια απόσταση, δηλαδή [Haykin, 1994]:

$$j_x = \min_j \|x - w_j\|, \quad j=1,2, \dots, m \quad (6.3)$$

όπου m είναι ο αριθμός των νευρώνων στο επίπεδο Kohonen (ή επίπεδο εξόδου).

Σύνοψη του αλγορίθμου εκπαίδευσης του δικτύου Kohonen

Στη συνέχεια θα παρουσιάσουμε μια σύνοψη του αλγορίθμου εκπαίδευσης του δικτύου του Kohonen [Kohonen 1989]:

Βήμα 1^ο: Αρχικοποίηση

Αρχικοποίησε τα συναπτικά βάρη δίνοντας τους μικρές τυχαίες αρχικές τιμές, για παράδειγμα στο διάστημα $[0, 1]$ (με ομοιόμορφη κατανομή). Αρχικοποίησε το ρυθμό μάθησης a δίνοντας του μια μικρή θετική τιμή.

Βήμα 2^ο: Ενεργοποίηση

Ενεργοποίησε το δίκτυο Kohonen εφαρμόζοντας στην είσοδό του ένα διάνυσμα εκπαίδευσης \mathbf{x} και βρίσκοντας το νικητή νευρώνα (winner-takes-all neuron) j_x για την επανάληψη p , χρησιμοποιώντας ως κριτήριο την ελάχιστη Ευκλείδεια απόσταση:

$$j_x(p) = \min_j \|\mathbf{x} - \mathbf{w}_j(p)\| = \left\{ \sum_{i=1}^n [x_i - w_{ji}(p)]^2 \right\}^{1/2}, \quad j=1,2, \dots, m \quad (6.4)$$

όπου n είναι ο αριθμός των νευρώνων στο επίπεδο εισόδου, και m είναι ο αριθμός των νευρώνων στο επίπεδο εξόδου ή επίπεδο Kohonen.

Βήμα 3^ο: Εκπαίδευση

Ενημέρωσε τις τιμές των βαρών σύμφωνα με τη σχέση:

$$w_{ji}(p+1) = w_{ji}(p) + \Delta w_{ji}(p) \quad (6.5)$$

όπου $\Delta w_{ji}(p)$ είναι η διόρθωση στην τιμή του βάρους στην επανάληψη p .

Η διόρθωση του λάθους $\Delta w_{ji}(p)$ καθορίζεται από τον κανόνα της ανταγωνιστικής μάθησης:

$$\Delta w_{ji}(p) = \begin{cases} \alpha \cdot [x_i - w_{ji}(p)], & j \in \Lambda_j(p) \\ 0, & j \notin \Lambda_j(p) \end{cases} \quad (6.6)$$

όπου α είναι ο ρυθμός μάθησης και $\Lambda_j(p)$ είναι η γειτονιά του νικητή νευρώνα (winner-takes-all-neuron) j_x στην επανάληψη p .

Ο ρυθμός μάθησης α μπορεί:

- Είτε να είναι σταθερός καθ' όλη τη διάρκεια της διαδικασίας εκπαίδευσης.
- Είτε να μεταβάλλεται. Σε αυτή την περίπτωση, συνήθως, ξεκινάμε με μια μεγάλη τιμή του ρυθμού μάθησης για $p=0$ και εν συνεχεία κατά τη διάρκεια της εκπαίδευσης την ελαττώνουμε με γραμμικό ή μη-γραμμικό τρόπο [Callan, 1999]. Μια σχέση που μπορούμε να χρησιμοποιήσουμε, για παράδειγμα, είναι η εξής [Αργυράκης, 2001]:

$$\alpha(p) = \alpha(0) \cdot \left(1 - \frac{p}{P}\right) \quad (6.7)$$

Όπου $a(0)$ είναι η αρχική τιμή (συνήθεις τιμές για την περίπτωση αυτή είναι στο διάστημα $(0.2, 0.5)$) του ρυθμού μάθησης, p είναι ο τρέχον κύκλος (επανάληψη) εκπαίδευσης και P είναι ο συνολικός αριθμός των κύκλων εκπαίδευσης του δικτύου Kohonen.

Η γειτονιά Λ_j του νικητή νευρώνα μπορεί:

- Να είναι σταθερή, καθ' όλη την εκπαιδευτική διαδικασία.
- Να ξεκινά η εκπαίδευση με μια μεγάλη γειτονιά γύρω από το νικητή νευρώνα η οποία στη συνέχεια να ελαττώνεται όσο προχωρούν οι κύκλοι εκπαίδευσης μέχρις ότου να πάρει τελικά την τιμή 1, να περιλαμβάνει δηλαδή μόνο το νικητή νευρώνα. Στην περίπτωση αυτή δίνουμε στη γειτονιά ένα αρχικό πλάτος $d(0)$ (συνήθως θα πρέπει να είναι περίπου το $\frac{1}{2}$ του μεγέθους του επιπέδου Kohonen) το οποίο και μειώνουμε κατά τη διάρκεια της εκπαίδευσης. Αυτή η μείωση μπορεί να γίνει με γραμμικό και μη-γραμμικό τρόπο. Μια σχέση, για παράδειγμα, που μπορούμε να χρησιμοποιήσουμε είναι η ακόλουθη [Αργυράκης, 2001]:

$$d(p) = d(0) \cdot \left(1 - \frac{p}{P}\right) \quad (6.8)$$

Όπου $d(0)$ είναι το αρχικό πλάτος (ακτίνα) της γειτονιάς, p είναι ο τρέχον κύκλος εκπαίδευσης και P είναι ο συνολικός αριθμός των κύκλων εκπαίδευσης του δικτύου Kohonen.

Όλοι οι νευρώνες που βρίσκονται μέσα στην τοπολογική γειτονιά ενεργοποιούνται ταυτόχρονα και η σχέση ανάμεσα σε αυτούς τους νευρώνες είναι ανεξάρτητη από την απόστασή τους από τον νικητή νευρώνα j_x .

Για τους νευρώνες του επιπέδου Kohonen (επίπεδο εξόδου) η έξοδος τους δίνεται από τη σχέση:

$$y_j(p) = \begin{cases} 1, & j \in \Lambda_j(p) \\ 0, & j \notin \Lambda_j(p) \end{cases} \quad (6.9)$$

Βήμα 4^ο: Επανάληψη

Αύξησε τον μετρητή επανάληψης p κατά 1, πήγαινε στο βήμα 2 και συνέχισε την εκπαίδευση μέχρις ότου να ικανοποιηθεί το κριτήριο της ελάχιστης

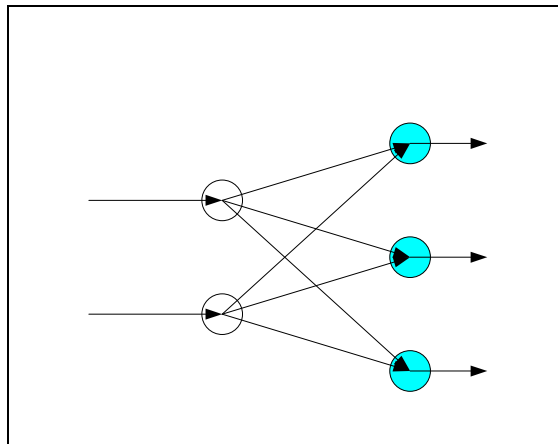
Ευκλείδεια απόστασης ή μέχρις ότου να μη συμβαίνει κάποια αξιοσημείωτη μεταβολή στον χάρτη χαρακτηριστικών (feature map).

Θα ολοκληρώσουμε τη συζήτησή μας για τα δίκτυα Kohonen με μερικές ασκήσεις όπου θα δούμε την εφαρμογή των δικτύων σε κάποια πολύ μικρά πρακτικά προβλήματα.



Άσκηση αυτοαξιολόγησης 2:

Υποθέστε ότι έχουμε ένα δίκτυο Kohonen με δύο εισόδους και τρεις νευρώνες (στο επίπεδο Kohonen), όπως φαίνεται στο παρακάτω σχήμα.



Έστω ότι τα αντίστοιχα βάρη (υποθέστε ότι δεν υπάρχουν συνδέσεις μεταξύ των κρυφών νευρώνων) που συνδέουν τους νευρώνες εισόδου με τους νευρώνες του επιπέδου Kohonen είναι:

$$\mathbf{W}_1 = \begin{bmatrix} 0.27 \\ 0.81 \end{bmatrix} \text{ και } \mathbf{W}_2 = \begin{bmatrix} 0.42 \\ 0.70 \end{bmatrix} \text{ και } \mathbf{W}_3 = \begin{bmatrix} 0.43 \\ 0.21 \end{bmatrix}$$

- a. Βρείτε το νικητή νευρώνα στην περίπτωση που στην είσοδο του δικτύου παρουσιαστεί το διάνυσμα:

$$\mathbf{X} = \begin{bmatrix} 0.52 \\ 0.12 \end{bmatrix}$$

Επίπεδο
Εισόδου

χρησιμοποιώντας το κριτήριο της ελάχιστης Ευκλείδειας απόστασης.

- b. Υπολογίστε τις μεταβολές στα συναπτικά βάρη, υποθέτοντας ότι ο ρυθμός μάθησης α έχει τιμή 0.1.
- c. Υπολογίστε τη νέα τιμή του διανύσματος των βαρών του νικητή νευρώνα.

\mathbf{x}_1

\mathbf{x}_2

Απάντηση

- a. Υπολογίζουμε τις Ευκλείδειες αποστάσεις κάθε νευρώνα από το διάνυσμα εισόδου.

Έτσι έχουμε:

$$d_1 = \sqrt{(x_1 - w_{11})^2 + (x_2 - w_{12})^2} = \sqrt{(0.52 - 0.27)^2 + (0.12 - 0.81)^2} = 0.73$$

$$d_2 = \sqrt{(x_1 - w_{21})^2 + (x_2 - w_{22})^2} = \sqrt{(0.52 - 0.42)^2 + (0.12 - 0.70)^2} = 0.59$$

$$d_3 = \sqrt{(x_1 - w_{31})^2 + (x_2 - w_{32})^2} = \sqrt{(0.52 - 0.43)^2 + (0.12 - 0.21)^2} = 0.13$$

Παρατηρούμε ότι ο νευρώνας 3 είναι αυτός με τη μικρότερη Ευκλείδεια απόσταση από το διάνυσμα εισόδου X.

- b. Σύμφωνα με τον αλγόριθμο εκπαίδευσης, θα μεταβληθούν μόνο τα συναπτικά βάρη του νικητή νευρώνα. Έτσι θα έχουμε:

$$\Delta w_{31} = \alpha \cdot (x_1 - w_{31}) = 0.1 \cdot (0.52 - 0.43) = 0.009$$

$$\Delta w_{32} = \alpha \cdot (x_2 - w_{32}) = 0.1 \cdot (0.12 - 0.21) = -0.009$$

- c. Μόνο το διάνυσμα βαρών του νικητή νευρώνα θα ενημερωθεί. Οπότε θα έχουμε:

$$\mathbf{W}_3(\mathbf{p}+1) = \mathbf{W}_3(\mathbf{p}) + \Delta \mathbf{W}_3(\mathbf{p}) = \begin{bmatrix} 0.43 \\ 0.21 \end{bmatrix} + \begin{bmatrix} 0.009 \\ -0.009 \end{bmatrix} = \begin{bmatrix} 0.439 \\ 0.201 \end{bmatrix} \cong \begin{bmatrix} 0.44 \\ 0.20 \end{bmatrix}$$

ενώ τα διανύσματα των βαρών των υπολοίπων νευρώνων θα μείνουν ως έχουν:

$$\mathbf{W}_1(\mathbf{p}+1) = \mathbf{W}_1(\mathbf{p}) = \begin{bmatrix} 0.27 \\ 0.81 \end{bmatrix} \text{ και } \mathbf{W}_2(\mathbf{p}+1) = \mathbf{W}_2(\mathbf{p}) = \begin{bmatrix} 0.42 \\ 0.70 \end{bmatrix}$$



7. Μελέτη Περίπτωσης: Επίλυση του TSP με τη χρήση δικτύου Hopfield

Στο κεφάλαιο αυτό θα ασχοληθούμε με μια εφαρμογή των Τεχνητών Νευρωνικών Δικτύων σε ένα πολύ γνωστό πρόβλημα βελτιστοποίησης, το πρόβλημα του περιοδεύοντος πωλητή (Traveling Salesman Problem – TSP). Το πρόβλημα αυτό επιλέχθηκε, αν και δεν είναι ένα από τα «κλασικά» πεδία εφαρμογής των Τεχνητών Νευρωνικών Δικτύων, επειδή αποτελεί ένα από τα πλέον διάσημα (και με πολλές εφαρμογές) προβλήματα βελτιστοποίησης το οποίο έχει αντιμετωπιστεί με μια πλειάδα μεθόδων οι περισσότερες από τις οποίες ανήκουν στη γενικότερη περιοχή της Τεχνητής Νοημοσύνης, που αποτελεί και το αντικείμενο μελέτης της θεματικής ενότητας «Τεχνητής Νοημοσύνη - Εφαρμογές» του προγράμματος σπουδών Πληροφορικής του Ελληνικού Ανοικτού Πανεπιστημίου. Έτσι για λόγους ομοιομορφίας (αντιμετώπιση του TSP με διαφορετικές μεθόδους) επιλέχθηκε το συγκεκριμένο πρόβλημα. Το TSP είναι ένα διάσημο πρόβλημα, για λόγους όμως πληρότητας της συζήτησης θα δώσουμε μια σύντομη περιγραφή του.

***Το TSP ορίζεται ως εξής:** Δεδομένων των εξόδων μεταφοράς (αυτό μπορεί να είναι είτε το κόστος μετακίνησης, είτε η απόσταση, είτε κάτι άλλο) μεταξύ διαφόρων πόλεων, το TSP συνίσταται στην προσπάθεια του πλανόδιου πωλητή να επισκεφτεί όλες τις πόλεις, μία μόνο φορά την καθεμιά, και να επιστρέψει στην πόλη από την οποία ξεκίνησε με το ελάχιστο δυνατό κόστος.*

Το TSP ανήκει στην κατηγορία των συνδυαστικών προβλημάτων βελτιστοποίησης και εμφανίζεται σε ένα πολύ μεγάλο αριθμό εφαρμογών. Υπάρχουν αρκετοί προσεγγιστικοί και ευρετικοί αλγόριθμοι που προσπαθούν να επιλύσουν το πρόβλημα. Στην ενότητα αυτή θα παρουσιάσουμε μια προσπάθεια επίλυσης του προβλήματος με τη χρήση δικτύων Hopfield. Τα δίκτυα Hopfield, στα οποία αναφερθήκαμε σε προηγούμενο κεφάλαιο, είναι μια κατηγορία αναδρομικών Τεχνητών Νευρωνικών Δικτύων που βρίσκει εφαρμογή και σε προβλήματα συνδυαστικής βελτιστοποίησης. Το TSP είναι το πρώτο πρόβλημα συνδυαστικής βελτιστοποίησης που αντιμετωπίστηκε με τη χρήση των δικτύων Hopfield. Η επίλυση του TSP με δίκτυα Hopfield παρουσιάζεται στην εργασία των Hopfield και Tank [Hopfield και Tank, 1985] και είναι αυτή που θα παρουσιάσουμε σε αυτή τη μελέτη περίπτωσης.

Έστω λοιπόν ότι έχουμε ένα δίκτυο από N πόλεις, οι οποίες συνδέονται όλες μεταξύ τους και τις οποίες θέλουμε να επισκεφτούμε (μία φορά κάθε πόλη) με το μικρότερο δυνατό κόστος (διανύοντας την μικρότερη απόσταση, για παράδειγμα). Οι δυνατές λύσεις ενός τέτοιου TSP με N πόλεις είναι $N!$. Για να επιλύσουμε αυτό το TSP με ένα δίκτυο Hopfield θα πρέπει να ακολουθήσουμε την παρακάτω διαδικασία [Hopfield και Tank, 1985]:

1. Το TSP μπορεί να αναπαρασταθεί με τη μορφή ενός πίνακα $N \times N$ όπου οι γραμμές αναπαριστούν τις πόλεις και οι στήλες τη σειρά με την οποία επισκεπτόμαστε κάθε πόλη. Για παράδειγμα, για ένα TSP με 5 πόλεις μια πιθανή αναπαράσταση του προβλήματος σε μορφή πίνακα (5×5) θα είναι:

	1	2	3	4	5
A	0	1	0	0	0
B	0	0	0	1	0
C	0	0	1	0	0
D	1	0	0	0	0
E	0	0	0	0	1

Από τον πίνακα αυτό βλέπουμε ότι η σειρά επίσκεψης των πόλεων θα είναι: $D \Rightarrow A \Rightarrow C \Rightarrow B \Rightarrow E$. Κάθε στοιχείο του πίνακα περιγράφεται από δύο συντεταγμένες, μια που καθορίζει την πόλη και μια που καθορίζει την σειρά επίσκεψης. Σύμφωνα με αυτή την αναπαράσταση σε κάθε γραμμή και σε κάθε στήλη του πίνακα θα πρέπει να υπάρχει μόνο ένα 1 και ο συνολικός αριθμός των 1 σε όλο τον πίνακα θα πρέπει να είναι ίσος με τον αριθμό των πόλεων, δηλαδή ίσος με N .

2. Θα δημιουργήσουμε ένα δίκτυο Hopfield το οποίο θα αποτελείται από N^2 νευρώνες. Το δίκτυο θα αναπαρίσταται σαν ένας πίνακας $N \times N$ όπου κάθε στοιχείο του πίνακα θα είναι και ένας νευρώνας. Η έξοδος κάθε νευρώνα θα ανατροφοδοτείται ως είσοδος σε όλους τους άλλους νευρώνες, ενώ δεν θα υπάρχει αυτό-ανατροφοδότηση. Το δίκτυο Hopfield που θα χρησιμοποιηθεί θα είναι το συνεχές και όχι το διακριτό που περιγράψαμε σε προηγούμενο κεφάλαιο. Το συνεχές Hopfield δουλεύει με τον ίδιο τρόπο με το διακριτό με μόνη διαφορά ότι χρησιμοποιεί νευρώνες συνάρτηση ενεργοποίησης που είναι συνεχής (για παράδειγμα την λογιστική ή την υπερβολική εφαπτομένη). Για την συγκεκριμένη

υλοποίηση η συνάρτηση ενεργοποίησης που θα χρησιμοποιήσουμε είναι η [Hopfield και Tank, 1985]:

$$s_{xi} = \text{sgm}_{0,\lambda}(u_{xi}) = \frac{1}{2} \left(1 + \tanh \left(\frac{u_{xi}}{\lambda} \right) \right) \quad (7.1)$$

όπου s_{xi} είναι η κατάσταση (state) το νευρώνα x_i , u_{xi} είναι το δυναμικό ενεργοποίησης του νευρώνα x_i και λ είναι μια παράμετρος που καθορίζει την κλίση της υπερβολικής εφαπτομένης (\tanh). Στο δίκτυο αυτό, κάθε νευρώνας θα χαρακτηρίζεται από δύο δείκτες όπου ο πρώτος θα καθορίζει την πόλη και ο δεύτερος τη σειρά με την οποία επισκεπτόμαστε την πόλη, έτσι για παράδειγμα ένας νευρώνας $V_{x,i}$ καθορίζει ότι η πόλη x είναι i -στη στη σειρά επίσκεψης. Θέλουμε, λοιπόν, «εκπαιδευοντας» το δίκτυο Hopfield να καταλήγουμε στο τέλος σε μια κατάσταση ισορροπίας όπου όλοι οι νευρώνες θα έχουν κατάσταση 0 εκτός από N όπου θα βρίσκονται σε μη-μηδενική κατάσταση (θα είναι ενεργοί). Οι ενεργοί νευρώνες (με τις συντεταγμένες τους) είναι αυτοί που καθορίζουν τη σειρά επίσκεψης των πόλεων. Έτσι για το παράδειγμα του TSP των 5 πόλεων και για την συγκεκριμένη λύση που είδαμε παραπάνω η κατάσταση στην οποία θα ισορροπεί το δίκτυο θα είναι η εξής:

0	1	0	0	0
0	0	0	1	0
0	0	1	0	0
1	0	0	0	0
0	0	0	0	1

Δηλαδή όλοι οι νευρώνες θα είναι σε κατάσταση 0, εκτός από τους $V_{4,1}$, $V_{1,2}$, $V_{3,3}$, $V_{2,4}$, $V_{5,5}$ που είναι ενεργοί και καθορίζουν ότι η σειρά επίσκεψης των πόλεων θα είναι: $D \Rightarrow A \Rightarrow C \Rightarrow B \Rightarrow E$.

3. Στη συνέχεια υπολογίζουμε τα συναπτικά βάρη του δικτύου Hopfield. Τα βάρη υπολογίζονται σύμφωνα με τη σχέση [Hopfield και Tank, 1985]:

$$w_{xi,yj} = -A \cdot \delta_{xy} \cdot (1 - \delta_{ij}) - B \cdot \delta_{ij} \cdot (1 - \delta_{xy}) - C - D \cdot d_{xy} \cdot (\delta_{j,i+1} + \delta_{j,i-1}), \quad (7.2)$$

$$x, i, y, j \in [1, N]$$

Όπου το:

- $w_{xi,yj}$ είναι το βάρος της σύνδεσης του νευρώνα x_i με τον νευρώνα y_j (κάθε νευρώνας, όπως είπαμε και παραπάνω, εξαιτίας του τρόπου αναπαράστασης του δικτύου καθορίζεται από δύο δείκτες).
- $\delta_{ij} = \begin{cases} 1, & \text{αν } i = j \\ 0, & \text{αν } i \neq j \end{cases}$, (Kronecker)
- d_{xy} είναι η απόσταση (το κόστος μετακίνησης) ανάμεσα στις πόλεις x και y
- A, B, C, D : είναι παράμετροι για τις οποίες θα μιλήσουμε παρακάτω.

Η σχέση αυτή για τον υπολογισμό των βαρών προκύπτει από την σχέση που υπολογίζει την ενέργεια του δικτύου. Έτσι λοιπόν, η ενέργεια του δικτύου Hopfield για την επίλυση του TSP, όπως αυτή διαμορφώνεται από τους περιορισμούς που επιβάλλει το συγκεκριμένο πρόβλημα έχει τη μορφή [Hopfield και Tank, 1985]:

$$\begin{aligned}
 E &= E_1 + E_2 + E_3 + E_4, \\
 E_1 &= \frac{A}{2} \cdot \sum_x \sum_i \sum_{j \neq i} s_{xi} \cdot s_{xj} \\
 E_2 &= \frac{B}{2} \cdot \sum_i \sum_x \sum_{y \neq x} s_{xi} \cdot s_{yi} \\
 E_3 &= \frac{C}{2} \cdot \left(\sum_x \sum_i s_{xi} - N \right)^2 \\
 E_4 &= \frac{D}{2} \cdot \sum_x \sum_{y \neq x} \sum_i d_{xy} \cdot s_{xi} \cdot (s_{y,i+1} + s_{y,i-1})
 \end{aligned} \tag{7.3}$$

Όπου:

- Το E_1 εκφράζει τον περιορισμό ότι κάθε γραμμή του πίνακα δεν πρέπει να περιέχει περισσότερους από έναν άσους (1).
- Το E_2 εκφράζει τον περιορισμό ότι κάθε στήλη δεν πρέπει να περιέχει περισσότερους από έναν άσους (1).
- Το E_3 εκφράζει τον περιορισμό ότι θα πρέπει να υπάρχουν ακριβώς N άσοι (1) σε ολόκληρο τον πίνακα (με διάσταση $N \times N$).
- Το E_4 εκφράζει τον περιορισμό ότι θα πρέπει να ευνοείται η συντομότερη (με το μικρότερο κόστος μετάβασης) διαδρομή

Συγκρίνοντας τη συνάρτηση (7.3) με την συνάρτηση:

$$E = -\frac{1}{2} \cdot \sum_i \sum_{j \neq i} s_i \cdot s_j \cdot w_{ij} - \sum_i s_i \cdot I_i \tag{7.4}$$

που δίνει γενικά την ενέργεια ενός δικτύου Hopfield προκύπτει η σχέση (7.2) για τον υπολογισμό των βαρών για την επίλυση του TSP.

Στον υπολογισμό των βαρών θα πρέπει να προσέξουμε, επειδή στα δίκτυα Hopfield δεν υπάρχει ανατροφοδότηση, το βάρος της σύνδεσης ενός νευρώνα με τον εαυτό του να είναι μηδέν, δηλαδή: $w_{xi,xi} = 0$.

4. Αφού ολοκληρώσουμε τον υπολογισμό των βαρών του δικτύου στη συνέχεια αρχικοποιούμε τις καταστάσεις των νευρώνων σε μικρές τυχαίες τιμές, που τις επιλέγουμε ομοιόμορφα από ένα διάστημα τιμών.
5. Τέλος ενημερώνουμε με τυχαίο τρόπο τις καταστάσεις των νευρώνων του δικτύου μέχρις ότου αυτό φτάσει σε μια κατάσταση ισορροπίας. Η ενημέρωση της κατάστασης κάθε νευρώνα γίνεται ως εξής:

- Υπολογίζουμε το δυναμικό ενεργοποίησης του νευρώνα [Hopfield και Tank, 1985]:

$$u_{xi} = \sum_y \sum_j w_{xi,yj} \cdot s_{yj} + I_{xi}, \quad (7.5)$$

$$I_{xi} = C \cdot N$$

όπου I_{xi} είναι η πόλωση (bias) του νευρώνα x_i .

- Υπολογίζουμε τη νέα κατάσταση του νευρώνα ως εξής [Hopfield και Tank, 1985]:

$$s_{xi} = \frac{1}{2} \left(1 + \tanh \left(\frac{u_{xi}}{\lambda} \right) \right) \quad (7.6)$$

Μετά από έναν αριθμό ενημερώσεων των καταστάσεων το δίκτυο Hopfield θα ισορροπήσει σε μία κατάσταση που θα είναι και η λύση του προβλήματος. Από τα πειράματα που έγιναν με διάφορα προβλήματα TSP προέκυψε ότι τα δίκτυα Hopfield μπορούν να δώσουν λύση σε μικρά σχετικά προβλήματα (μέχρι 30 πόλεις όπως υποστηρίζεται στο [Hopfield και Tank, 1985], ενώ άλλοι ερευνητές υποστηρίζουν ότι τα προβλήματα που μπορούν να επιλυθούν με αυτή τη μέθοδο είναι ακόμη μικρότερα). Επίσης οι λύσεις που προκύπτουν δεν είναι πάντα οι βέλτιστες, ενώ πολλές φορές το δίκτυο μπορεί να μην καταλήξει σε κάποια «νόμιμη» λύση. Αυτό συμβαίνει επειδή αυτή η μέθοδος είναι πολύ ευαίσθητη στην επιλογή των τιμών των παραμέτρων A, B, C, D. Οι παράμετροι αυτοί καθορίζουν σε πολύ μεγάλο βαθμό τη σύγκλιση της μεθόδου και έως σήμερα δεν έχει βρεθεί κάποιος συστηματικός τρόπος για την επιλογή τους (για την επιλογή των τιμών τους ακολουθείται η μέθοδος της δοκιμής και του λάθους – trial and

error method). Παρ' όλο όμως ότι η μέθοδος της επίλυσης του TSP με τη χρήση δικτύων Hopfield δεν είναι η καλύτερη για συγκεκριμένο πρόβλημα, είχαμε την ευκαιρία, μέσα από αυτή τη μελέτη περίπτωσης, να δούμε ένα ενδιαφέρον πεδίο εφαρμογής των Νευρωνικών Δικτύων, όχι και τόσο διαδεδομένο.

BIBΛΙΟΓΡΑΦΙΑ

1. Amit, D. J. (1989), "*Modeling Brain Functions: The World of Attractor Neural Networks*", Cambridge University Press, New York.
2. Callan, R. (1999), "*The essence of Neural Networks*", Prentice Hall Europe.
3. Fausett, L. (1994). "*Fundamentals of Neural Networks: Architectures, Algorithms and Applications*". Prentice Hall International.
4. Golden, R. (1996). "*Mathematical Methods for Neural Network Analysis and Design*". MIT Press.
5. Hagan, M. Demuth, H. and Beale M. (1995). "*Neural Network Design*". International Thomson Publishing Company.
6. Hassoun, M. (1995). "*Fundamentals of Artificial neural Networks*". MIT Press.
7. Haykin S. (1994). "*Neural Networks, A Comprehensive Foundation*". Macmillan College Publishing Company Inc.
8. Haykin S. (1999). "*Neural Networks, A Comprehensive Foundation. 2nd edition*" Prentice Hall international.
9. Hopfield, J. J. (1982), "Neural Networks and physical systems with emergent collective computational abilities", *Proceedings of the National Academy of Sciences of the USA*, 79, 2554-2558.
10. Hopfield, J. J. and Tank, D. W. (1985), "Neural computations of decisions in optimization problems", *Biological Cybernetics*, 52, 141.
11. Kohonen, T. (1982), "Self-Organized formation of topologically correct feature maps", *Biological Cybernetics*, 43, 59-69.
12. Kohonen, T. (1989), "*Self-Organization and Associative Memory*", 3rd edition, Springer-Verlag, Berlin, Heidelberg.
13. Kohonen, T. (1989), "The self-organizing map", *Proceedings of the IEEE*, 78, 1464-1480.
14. Kosko, B. (1987), "Adaptive Bidirectional associative memories", *Applied Optics*, 26(23), 4947-4960.
15. Kosko, B. (1988), "Bidirectional associative memories", *IEEE Transactions on Systems Man and Cybernetics*, SMC-18, 49-60.

16. Kramer, A. H. και Sangiovanni-Vincentelli, A. (1989), “Efficient parallel learning algorithms for Neural Networks”, *Advances in Neural Information Processing Systems*, vol. 1, pp. 40-48, San Mateo, CA: Morgan Kaufmann.
17. Minsky, M. L. και Papert, S. A. (1969), “*Perceptrons*”, MIT Press, Cambridge, MA.
18. Negnevitsky, M. (2002), “*Artificial Intelligence: A Guide to Intelligent Systems*”, Addison Wesley.
19. Patterson D. (1996), “*Artificial Neural Networks, Theory and Applications*”. Prentice Hall.
20. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986), “Learning Representations by Back-Propagating Errors”. *Nature* 323,533.
21. Rumelhart, D. E. and J. L. McClelland, eds (1986), “*Parallel Distributed Processing: Explorations in the Microstructure of Cognition*”, Cambridge, MA: MIT Press.
22. Αργυράκης Π. (2001). “*Νευρωνικά Δίκτυα και Εφαρμογές*” Έντυπο Εκπαιδευτικό Υλικό για τη Θεματική Ενότητα «Τεχνητή Νοημοσύνη – Εφαρμογές», Ελληνικό Ανοικτό Πανεπιστήμιο.