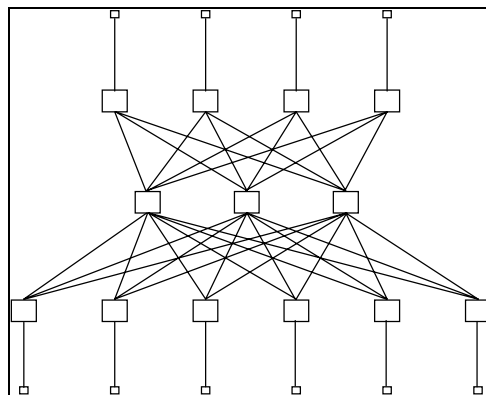


Νικόλαος Γλυνός  
Επικ. Καθηγητής  
Τμήμα Μαθηματικών

Γεωργίου Β. Χάρης  
Μελισσόβας Β. Σπύρος  
Παπαδόπουλος Σ. Δημήτρης

## Μέθοδοι Εκπαίδευσης και Μοντέλα Τεχνητών Νευρωνικών Δικτύων



Πανεπιστήμιο Ιωαννίνων  
Τμήμα Πληροφορικής

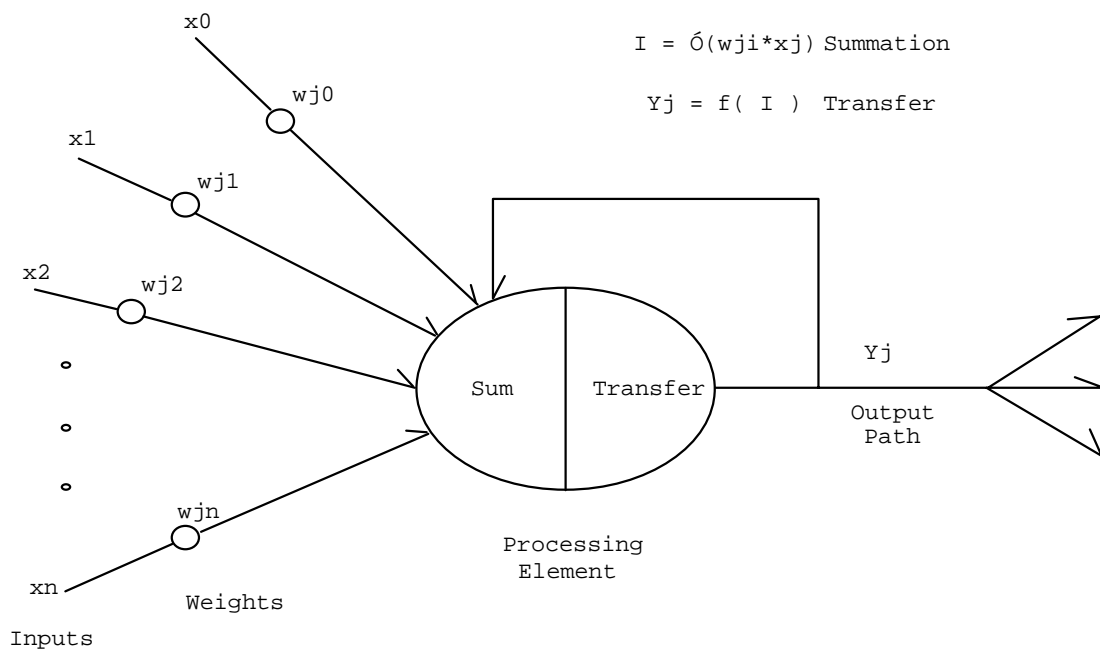
*Ιωάννινα - 1995*

# Μέθοδοι Εκπαίδευσης και Μοντέλα Τεχνητών Νευρωνικών Δικτύων

## Γενικά

Ο όρος νευρωνικά δίκτυα χρησιμοποιείται για να περιγράψει ένα σύνολο διαφορετικών υπολογιστικών μοντέλων που εξομοιώνουν κάποιες από τις λειτουργίες του ανθρώπινου εγκεφάλου, χρησιμοποιώντας ορισμένες βασικές δομές.

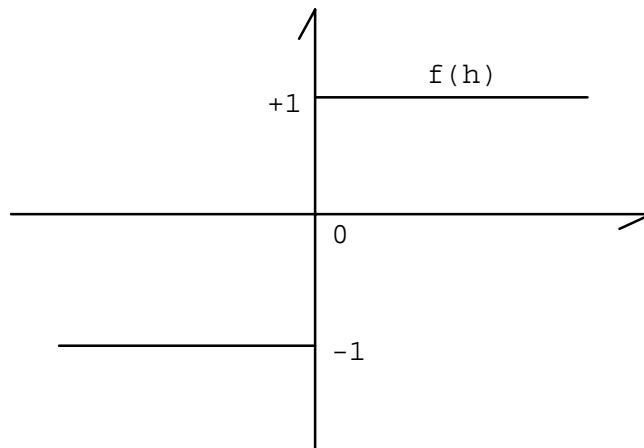
Ένα Τεχνητό Νευρωνικό Δίκτυο (Τ.Ν.Δ.) είναι μια οργανωμένη δομή μονάδων επεξεργασίας, οι οποίες συνδέονται μεταξύ τους. Το δίκτυο περιλαμβάνει μια σειρά εισόδων και μια σειρά εξόδων. Μια επεξεργαστική μονάδα είναι ένας τεχνητός νευρώνας με περιορισμένη μνήμη και επεξεργαστική ισχύ. Δέχεται εισόδους από νευρώνες με τους οποίους συνδέεται και υπολογίζει μια τιμή εξόδου σαν συνάρτηση των εισόδων του, την οποία διοχετεύει με τη σειρά του σε άλλους νευρώνες με τους οποίους επικοινωνεί. Οι υπολογισμοί που πραγματοποιούνται από κάθε μονάδα είναι μη-γραμμικοί. Υπάρχουν τρεις κατηγορίες μονάδων: μονάδες "εισόδου" οι οποίες λαμβάνουν τα δεδομένα εισόδου από εξωτερικές πηγές, μονάδες "εξόδου" οι οποίες στέλνουν τα αποτελέσματα εκτός συστήματος, και "κρυμμένες" μονάδες.



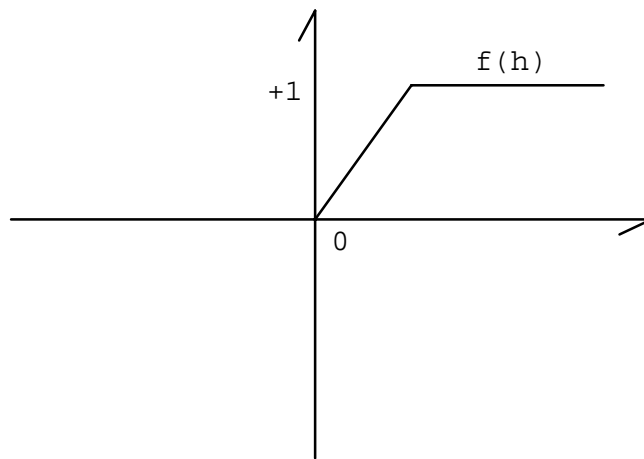
**Σχήμα 1 - Μονάδα Επεξεργασίας**

Η ενεργοποίηση  $a_j(t)$  κάποιας μονάδας  $u_j$  αντιπροσωπεύει την κατάσταση της στη χρονική στιγμή  $t$ . Οι τιμές ενεργοποίησης μπορεί να είναι συνεχείς ή διακριτές. Αν είναι συνεχείς, τότε μπορεί να είναι φραγμένες ή μη-φραγμένες. Αν είναι διακριτές, τότε μπορεί να είναι δυαδικές ή να παίρνουν τιμές από ένα μικρό σύνολο τιμών. Διαφορετικές υποθέσεις για τις τιμές ενεργοποίησης οδηγούν σε μοντέλα με λίγο διαφορετικά χαρακτηριστικά.

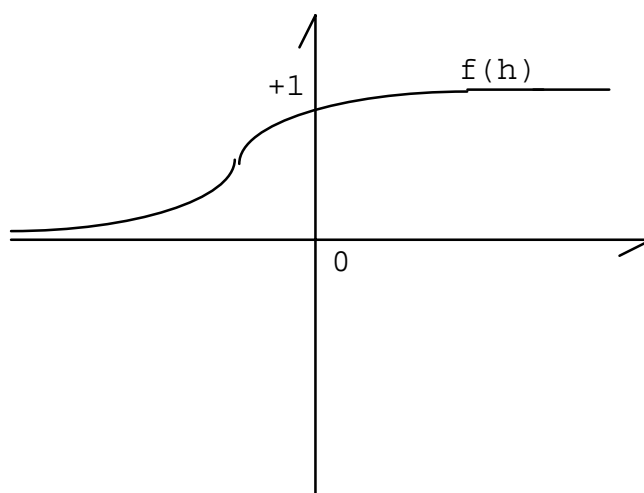
Σε κάθε μονάδα  $u_j$  αντιστοιχεί μια συνάρτηση εξόδου  $f_j$ , η οποία μετασχηματίζει την παρούσα κατάσταση  $a_j(t)$  της μονάδας σε ένα σήμα εξόδου  $o_j(t) = f_j(a_j(t))$ . Στις περισσότερες περιπτώσεις, η  $f(x)$  είναι κάποια οριακή συνάρτηση (threshold function). Έτσι μία μονάδα επηρεάζει μια γειτονική της, μόνο όταν η τιμή ενεργοποίησής της είναι μεγαλύτερη από μια προκαθορισμένη τιμή.



Hard Limiter



Threshold Logic



Sigmoid

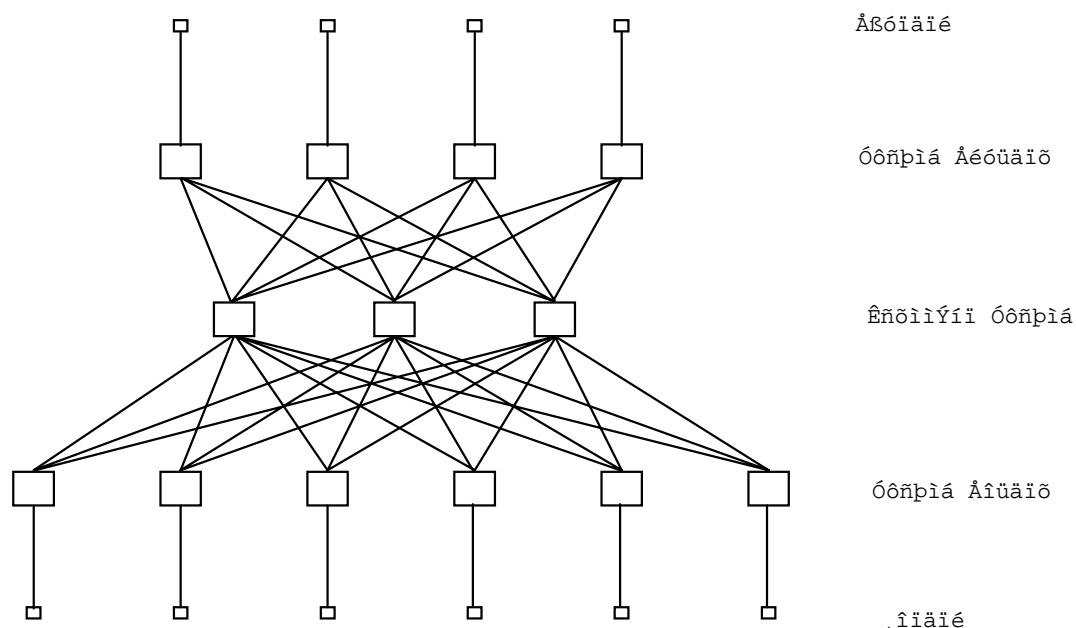
**Σχήμα 2 - Οριακές Συναρτήσεις Εξόδου**

Η κατάσταση ενεργοποίησης σε μια μονάδα δίνεται από την

$$\text{εξίσωση: } a_j(t) = \sum_{k=1}^i w_{jk} o_k(t) + \theta_j$$

όπου  $w_{jk}$  είναι οι συντελεστές βαρών των εισόδων της μονάδας  $u_j$  και  $\theta_j$  είναι ένα εσωτερικό όριο που χαρακτηρίζει την ίδια τη μονάδα. Σε μια τέτοια περίπτωση, η αναπαράσταση του προτύπου σύνδεσης καθορίζεται από τον προσδιορισμό του συντελεστή βάρους κάθε σύνδεσης του δικτύου. Ο συντελεστής  $w_{jk}$  είναι θετικός, εάν η μονάδα  $u_k$  διεγείρει τη μονάδα  $u_j$ , αρνητικός εάν την αποδιεγείρει και μηδέν, αν η μονάδα  $u_k$  δε συνδέεται με τη μονάδα  $u_j$ . Η απόλυτη τιμή του συντελεστή βάρους  $w_{jk}$  καθορίζει την ισχύ της σύνδεσης.

Συνήθως οι μονάδες του δικτύου διατάσσονται σε ξεχωριστές δομές, οι οποίες καλούνται "στρώματα" (layers). Τα πρωταρχικά μοντέλα αποτελούνται από ένα απλό στρώμα, όπου κάθε είσοδος του δικτύου συνδέεται με όλες τις μονάδες του στρώματος αυτού. Οι πληροφορίες ρέουν μέσω του στρώματος από τις εισόδους στις εξόδους, χωρίς ανάδραση των εξόδων. Τα μοντέλα αυτής της κλάσης ονομάζονται feed-forward μοντέλα. Άλλα μοντέλα έχουν επεκτείνει αυτή την ιδέα σε δομές πολλαπλών feed-forward στρωμάτων. Επίσης έχουν εισαχθεί μοντέλα τα οποία χρησιμοποιούν την έννοια των feed-backward συνδέσεων. Τα δίκτυα αυτά (δίκτυα ανάδρασης) βελτιώνουν τη διαδικασία εκπαίδευσης και μάθησης του δικτύου τροφοδοτώντας προς τα πίσω αποτελέσματα (σφάλματα) από ένα επόμενο στρώμα σε ένα προηγούμενο.



**Σχήμα 3 - Γενικό Μοντελο Δομής ΤΝΔ**

Ο κανόνας μάθησης είναι ο τρόπος εύρεσης των σωστών συντελεστών βαρών, έτσι ώστε οι σωστές μορφές ενεργοποίησης να δημιουργηθούν κάτω από κατάλληλες συνθήκες. Για να αλλάξουμε τη δομή επεξεργασίας ή της γνώσης σε ένα ΤΝΔ, χρειάζεται να τροποποιήσουμε το πρότυπο συνδέσεων. Οι αλλαγές αυτές μπορούν να γίνουν με τους εξής τρόπους:

1. Ανάπτυξη καινούριων συνδέσμων.
2. Αφαίρεση παλιών συνδέσμων.
3. Τροποποίηση συντελεστών βάρους.

Οι μέθοδοι 1 και 2 χρησιμοποιούνται πιο σπάνια, διότι μπορούν να θεωρηθούν ως ειδικές περιπτώσεις της μεθόδου 3. Η αλλαγή του συντελεστή βάρους μιας σύνδεσης από 0 σε κάποια θετική ή αρνητική τιμή έχει το ίδιο αποτέλεσμα όπως και η δημιουργία μιας νέας σύνδεσης. Η αλλαγή ενός συντελεστή βάρους σε 0 έχει το ίδιο αποτέλεσμα όπως και η αφαίρεση μιας υπάρχουσας σύνδεσης.

## Μορφές μάθησης των ΤΝΔ

Μεγάλες δυσκολίες συναντώνται στην προσπάθεια καθορισμού κατάλληλων αλγορίθμων μάθησης για την εκπαίδευση των ΤΝΔ. Η προσέγγιση, η οποία χρησιμοποιείται για την ανάπτυξη αλγορίθμων μάθησης ΤΝΔ μοντέλων είναι τελείως διαφορετική από την προσέγγιση των κλασικών αλγορίθμων μηχανικής μάθησης (machine learning).

Πρώτον, υποθέτουμε ότι ο σκοπός της εκπαίδευσης δεν είναι ο σχηματισμός σαφών κανόνων, αλλά η εύρεση κατάλληλων συντελεστών βάρους, οι οποίοι θα επιτρέπουν στο δίκτυο να ενεργεί σαν να γνώριζε τους κανόνες.

Δεύτερον, δεν αποδίδουμε ισχυρές υπολογιστικές ικανότητες στο μηχανισμό μάθησης, αλλά χρησιμοποιούμε απλούς μηχανισμούς διαμόρφωσης συντελεστών βάρους, οι οποίοι αναπροσαρμόζουν την ισχύ των συνδέσεων μεταξύ των μονάδων, βασισμένοι μόνο σε τοπικά διαθέσιμες πληροφορίες στο δίκτυο.

Οι μεθοδολογίες που χρησιμοποιούνται για την εκπαίδευση ΤΝΔ μπορούν να διακριθούν στις: μάθηση με επίβλεψη και μάθηση χωρίς επίβλεψη.

Η μάθηση με επίβλεψη (supervised learning) απαιτεί ένα σύνολο εκπαιδευτικών προτύπων γνωστής ταξινόμησης και μία εξωτερική διαδικασία διδασκαλίας. Η διαδικασία διδασκαλίας χρησιμοποιείται για την αναπροσαρμογή των συντελεστών βάρους του δικτύου, σύμφωνα με την απόκριση του δικτύου στα εκπαιδευτικά πρότυπα. Τις περισσότερες φορές η αναπροσαρμογή αυτή είναι ανάλογη του σφάλματος που προκύπτει κατά την ταξινόμηση των προτύπων εισόδου. Η χρήση μάθησης με επίβλεψη χωρίζεται σε δύο φάσεις: τη φάση μάθησης (learning phase) και τη φάση έρευνας (search phase). Κατά τη διάρκεια της φάσης μάθησης δημιουργείται ένα σύνολο εκπαιδευτικών προτύπων, το οποίο προέρχεται από αντιπροσωπευτικά δείγματα του περιβάλλοντος στο οποίο πρόκειται να λειτουργήσει το ΤΝΔ. Αυτό το σύνολο πρέπει να περιλαμβάνει δείγματα από όλες τις δυνατές κλάσεις προτύπων. Κατόπιν, τα εκπαιδευτικά πρότυπα εισάγονται στο δίκτυο και το σύστημα τροποποιείται μέσω ενός εκπαιδευτικού αλγορίθμου. Όταν τα αποτελέσματα της φάσης μάθησης είναι ικανοποιητικά, δηλαδή όταν όλα τα εκπαιδευτικά πρότυπα ταξινομούνται σωστά, το δίκτυο μπορεί να χρησιμοποιηθεί για τη φάση έρευνας. Κατά τη φάση έρευνας, άγνωστα πρότυπα εισάγονται στο δίκτυο.

Η μάθηση χωρίς επίβλεψη (unsupervised learning) χρησιμοποιεί εκπαιδευτικά πρότυπα άγνωστης ταξινόμησης, χωρίς καμία εξωτερική διαδικασία διδασκαλίας. Μια εσωτερική διαδικασία διδασκαλίας καθορίζει τον τρόπο αναπροσαρμογής των παραμέτρων του δικτύου, βασιζόμενη στη φύση των προτύπων εισόδου. Στην περίπτωση αυτή, η διαδικασία διδασκαλίας έχει ως αποτέλεσμα την εσωτερική ταξινόμηση των προτύπων εκπαίδευσης, σύμφωνα με κάποιο μέτρο ομοιότητας μεταξύ των προτύπων. Δηλαδή, παρόμοια εκπαιδευτικά πρότυπα ομοδοποιούνται από τον αλγόριθμο εκπαίδευσης. Αυτές οι ομάδες ή κατηγορίες αποτελούν τις κλάσεις προτύπων, στις οποίες άγνωστα πρότυπα εισόδου ταξινομούνται. Αυτή η προσέγγιση έχει χρησιμοποιηθεί για την ανάπτυξη ΤΝΔ μοντέλων, τα οποία έχουν την ικανότητα να ανακαλύπτουν μοναδικά χαρακτηριστικά σε ένα σύνολο προτύπων.



## Κανόνες εκπαίδευσης ΤΝΔ

Όλοι οι αλγόριθμοι εκπαίδευσης βασίζονται σε γενικούς "κανόνες" μάθησης, εκ των οποίων άλλοι έχουν ως πρότυπό τους το μοντέλο της βιολογικής μάθησης και άλλοι αποτελούν υλοποίηση μαθηματικών μοντέλων. Πάντως, ο μηχανισμός της μάθησης είναι σίγουρα πιο πολύπλοκος από τις απλοποιήσεις που εμπεριέχουν οι κανόνες που έχουν αναπτυχθεί.

Οι κυριότερες τεχνικές μάθησης παρουσιάζονται παρακάτω.

- Κανόνας Hebb

Ο κανόνας αυτός παρουσιάστηκε από τον Hebb το 1949. Η βασική του ιδέα είναι ότι αν μια επεξεργαστική μονάδα δέχεται είσοδο από μια άλλη και εάν και οι δύο έχουν την ίδια μορφή ενεργοποίησης, ο συντελεστής βάρους μεταξύ τους ενισχύεται.

- Κανόνας Δέλτα

Είναι αλλιώς γνωστός ως Windrow-Hoff κανόνας μάθησης και βασίζεται στην ιδέα της συνεχούς μεταβολής των συντελεστών βάρους. Έτσι ώστε να ελαχιστοποιείται η διαφορά ( $\Delta$ ) μεταξύ των επιθυμητών εξόδων και των εκάστοτε εξόδων του δικτύου. Ο κανόνας αυτός χρησιμοποιείται στο μοντέλο Adaline και τον LMS (Least Mean Square) αλγόριθμο εκπαίδευσής του.

- Gradient Descent

Είναι μια μαθηματική προσέγγιση του προβλήματος ελαχιστοποίησης του σφάλματος μεταξύ των επιθυμητών και των πραγματικών εξόδων. Η ποσοτική αναπροσαρμογή των συντελεστών βάρους είναι ανάλογη της πρώτης παραγώγου του σφάλματος. Ο κανόνας αυτός, αν και συγκλίνει σε μια κατάσταση ισορροπίας πολύ αργά, χρησιμοποιείται συχνά.

Ο κανόνας Δέλτα είναι ένα παράδειγμα του γενικότερου Gradient Descent κανόνα.

- Κανόνας Kohonen

Ο κανόνας αυτός προέρχεται από τον Τευνο Kohonen και είναι εμπνευσμένος από τη μάθηση σε βιολογικά συστήματα. Χρησιμοποιείται μονάχα σε μη-επιβλεπόμενη μάθηση. Η μονάδα επεξεργασίας με τη μεγαλύτερη τιμή εξόδου έχει τη δυνατότητα απαγόρευσης στις υπόλοιπες μονάδες, εκτός από συγκεκριμένο αριθμό γειτονικών, να μεταβάλλουν τους συντελεστές βάρους τους. Επιπλέον, το πλήθος των γειτονικών μονάδων μπορεί να μεταβάλλεται χρονικά κατά τη διαδικασία της εκπαίδευσης (συνήθως φθίνει).

- Back-Propagation

Η τεχνική αυτή βασίζεται στη διάδοση του σφάλματος "προς τα πίσω" (back-propagation) και είναι η πιο συχνά χρησιμοποιούμενη γενίκευση του κανόνα Δέλτα. Η διαδικασία αποτελείται από δύο φάσεις: τη φάση ανάκλησης και τη φάση μάθησης.

Γενικά, η μέθοδος αυτή εφαρμόζεται σε ιεραρχικά δίκτυα, όπου έχουμε κρυμμένα στρώματα μονάδων επεξεργασίας. Γενικά είναι πολύ αργή, μερικές φορές ασταθής, και συχνά συγκλίνει σε τοπικά ελάχιστα. Για το λόγο αυτό έχουν αναπτυχθεί αρκετές βελτιωμένες εκδόσεις. Ο κανόνας αυτός εφαρμόζεται κυρίως σε Multi-Layer Perceptron ΤΝΔ μοντέλα.

- Κανόνας Grossberg

Σύμφωνα με τον κανόνα αυτό, το ΤΝΔ αποτελείται από εσωτερικές και εξωτερικές μονάδες. Εσωτερικές είναι οι μονάδες που δέχονται πολλές εισόδους, ενώ εξωτερικές οι μονάδες που παράγουν πολλές εξόδους. Οφείλεται σε μελέτες του Steven Grossberg πάνω στον κανόνα του Hebb.

Σύμφωνα με τον κανόνα αυτό, η αναπροσαρμογή των συντελεστών βάρους είναι ανάλογη τόσο με τις καταστάσεις εισόδου όσο και με τις καταστάσεις εξόδου. Σημαντικές παράμετροι είναι η τιμή κατωφλίου (threshold) και ο χρόνος.

- Drive-Reinforcement Theory

Οφείλεται στον Harry Klopf. Το DRT είναι περισσότερο ένα μοντέλο μάθησης, παρά TND μοντέλο, αλλά έχει αρκετές ομοιότητες με τον κανόνα Grossberg.

Εδώ, σε αντίθεση με τον κανόνα Hebb, οι μονάδες επεξεργασίας μεταβάλλουν τους συντελεστές βάρους αναλογικά με το γινόμενο των καταστάσεων εισόδου και εξόδου. Επιπλέον, η τρέχουσα κατάσταση εξόδου εξαρτάται περισσότερο από την προηγούμενη είσοδο, παρά την τρέχουσα. Έτσι, η διάταξη των προτύπων στην είσοδο του δικτύου είναι πλέον σημαντική.

Η ισχύς κάθε σύνδεσης μετριέται με βάση την συχνότητα ενεργοποίησής της από την μονάδα επεξεργασίας. Αυτή είναι και η κύρια διαφορά με τους υπόλοιπους κανόνες μάθησης, και η βασική ομοιότητα με πραγματικά βιολογικά συστήματα.

Προς το παρόν, το DRT μοντέλο χρησιμοποιείται ως εργαλείο έρευνας.

- Boltzmann

Μια άλλη τεχνική μάθησης είναι αυτή που εφαρμόζεται στις αντίστοιχες "μηχανές" (Boltzmann). Υλοποιεί ένα στοχαστικό μοντέλο μετάβασης των μονάδων επεξεργασίας, αφού ο συσχετισμός των καταστάσεων εισόδου με τις αντίστοιχες καταστάσεις εξόδου βασίζεται σε υπό συνθήκη πιθανότητες. Αυτή είναι και η βασική διαφορά με τις υπόλοιπες τεχνικές, αφού για τις ίδιες εισόδους, το δίκτυο μπορεί να παράγει διαφορετικές εξόδους.

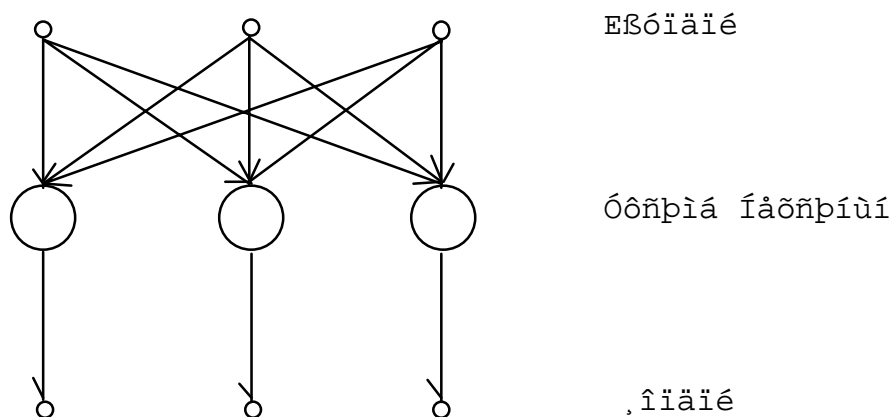
## Μοντέλα ΤΝΔ & Αλγόριθμοι

Παρακάτω παρουσιάζονται τα σημαντικότερα ΤΝΔ μοντέλα και οι αλγόριθμοι που χρησιμοποιούνται για την εκπαίδευσή τους.

- Perceptron

Κεντρική ιδέα του μοντέλου Perceptron είναι η ενσωμάτωση ενός μηχανισμού μάθησης στο αρχαιότερο μοντέλο Τ.Ν.Δ. που εισήγαγαν οι McCulloch-Pitts περίπου το 1943 και βασίζεται σε εργασίες του Rosenblatt. Ο όρος Perceptron αναφέρεται σε μια μεγάλη κλάση αλγορίθμων μάθησης για Νευρωνικά Μοντέλα.. Οι αλγόριθμοι οι οποίοι μελετήθηκαν και αναπτύχθηκαν από τον Rosenblatt είναι πάρα πολλοί και ποικίλοι. Ο σπουδαιότερος από αυτούς λέγεται Perceptron Αλγόριθμος Σύγκλισης (Perceptron Convergence Algorithm) και εξακολουθεί να βρίσκει εφαρμογές ακόμα και σήμερα.

Το μοντέλο Perceptron αποτελείται από ένα στρώμα  $N$  επεξεργαστικών μονάδων όπου κάθε μονάδα αντιστοιχεί σε μια είσοδο. Τιμές εισόδου είναι συνεχείς συναρτήσεις και η μάθηση γίνεται με επίβλεψη.



**Σχήμα 4** - Δομή Perceptron μοντέλου ΤΝΔ

Οι συντελεστές βάρους  $w_{ij}$  αναπροσαρμόζονται σύμφωνα με το βαθμό ομοιότητας της πραγματικής εξόδου  $t_i^m$ . Η διαφορά μεταξύ της πραγματικής και της επιθυμητής εξόδου ορίζεται ως το σφάλμα μάθησης, το οποίο τροφοδοτείται προς τα πίσω, για την

αναπροσαρμογή των συντελεστών βάρους, σύμφωνα με τον κανόνα μάθησης. Ο αλγόριθμος μάθησης περιγράφεται παρακάτω:

Βήμα 1:

Αρχικά τα βάρη  $w_{ij}$  και τα εσωτερικά όρια  $\theta_i$  παίρνουν μικρές τυχαίες τιμές.

Βήμα 2:

Παρουσιάζεται κάποιο πρότυπο  $m$  ( $x_0, x_1, \dots, x_{n-1}$ ) στην είσοδο του δικτύου, το οποίο επιλέγεται τυχαία από το σύνολο προτύπων εκπαίδευσης.

Βήμα 3:

Υπολογίζεται το αποτέλεσμα εξόδου

$$o_i(t) = f_h \left( \sum_{j=0}^{N-1} w_{ij}(t) x_j(t) + \theta_i \right)$$

Βήμα 4:

Οι συντελεστές βάρους των συνδέσμων αναπροσαρμόζονται σύμφωνα με τον παρακάτω κανόνα:

$$w_{ij}(t+1) = w_{ij}(t) + \varepsilon [t_i^m - o_i(t)] x_j(t)$$

Βήμα 5:

Επανάληψη, ξεκινώντας από το βήμα 2 μέχρι τη μάθηση όλων των προτύπων εκπαίδευσης.

$i$  =αριθμός μονάδας,  $i = 0, 1, \dots, N-1$

$j$  =αριθμός εισόδου,  $j = 0, 1, \dots, N-1$

$t$  =χρονική στιγμή

$w_{ij}(t)$  =συντελεστής βάρους του συνδέσμου μεταξύ της εισόδου  $j$  και της μονάδας  $i$ , τη χρονική στιγμή  $t$

$x_j(t)$  =τιμή της εισόδου  $j$  του προτύπου  $m$  τη χρονική στιγμή  $t$

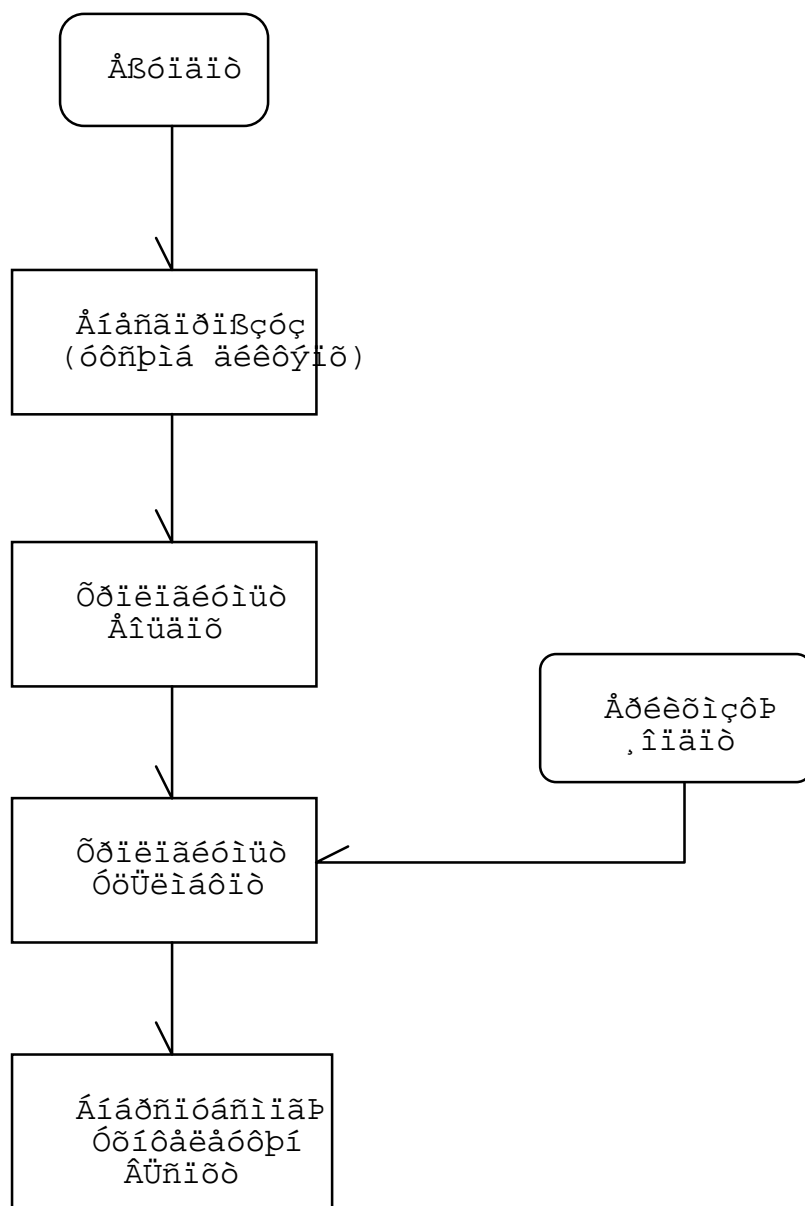
$\theta_i$  =εσωτερικό όριο της μονάδας  $i$

$t_i^m$  =επιθυμητή έξοδος της μονάδας  $i$  στο πρότυπο  $m$

$\varepsilon$  =ρυθμός μάθησης. Είναι θετικός αριθμός μεταξύ 0 και 1 και κανονίζει το ρυθμό αναπροσαρμογής των συντελεστών βάρους.

$f_h$  =οριακή συνάρτηση (hardlimiter, threshold logic)

Παρατηρούμε ότι για τα βάρη της κάθε μονάδας έχουμε ένα πίνακα μεγέθους  $N$ , που αντιστοιχεί στις  $N$  εισόδους του δικτύου. Επίσης έχουμε ένα στρώμα  $N$  μονάδων στο δίκτυο.



**Σχήμα 5** - Εκπαίδευση Perceptron ΤΝΔ

Σε κάθε επανάληψη του αλγορίθμου για τον υπολογισμό των εξόδων της κάθε μονάδας έχουμε για κάθε κόμβο ένα άθροισμα  $N$  όρων, καθένας από τους οποίους απαιτεί ένα άθροισμα και ένα γινόμενο, και συνολικά  $N$  κόμβους στο δίκτυο.

$(N \text{ μονάδες}) \cdot ((N \text{ όροι αθροίσματος}) \cdot (\text{γινόμενο} + \text{άθροισμα})) \Rightarrow O(N^2)$

Επίσης, για την αναπροσαρμογή των  $N$  συντελεστών βάρους για όλες τις  $N$  επεξεργαστικές μονάδες, έχουμε  $O(N^2)$  πολυπλοκότητα. Συνεπώς, η πολυπλοκότητα για κάθε επανάληψη είναι  $O(N^2)$ .

Τέλος, για  $M$  πλήθος προτύπων εκπαίδευσης, η συνολική πολυπλοκότητα του αλγορίθμου είναι  $O(MN^2)$ .

Σημείωση: Η πολυπλοκότητα αυτή προκύπτει επειδή έχουμε  $N$  πλήθος εισόδων και όλες οι εισοδοί συνδέονται με όλες τις μονάδες, ενώ το πλήθος των επαναλήψεων είναι  $M$ .

Ο Rosenblatt έχει αποδείξει ότι αν οι κλάσεις των προτύπων είναι γραμμικά διαχωρίσιμες (δηλαδή, υπάρχουν υπερεπίπεδα τα οποία μπορούν να διαχωρίσουν όλα τα ζεύγη των κλάσεων προτύπων), τότε ο παραπάνω αλγόριθμος συγκλίνει σε ένα πεπερασμένο αριθμό επαναλήψεων. Διαφορετικά, σε μη-διαχωρίσιμες περιπτώσεις, οι τιμές των συντελεστών βάρους μπορεί να μεταβάλλονται καθ' όλη τη διάρκεια της εκτέλεσης του αλγορίθμου.

Συγκεκριμένα, θεωρώντας  $K$  το πλήθος των διαφορετικών καταστάσεων στις οποίες μεταβαίνει το δίκτυο κατά την εκπαίδευση, τότε:

$$\frac{T}{\ln T} < \frac{N}{2\delta (K+\delta)}$$

όπου  $T$  είναι ο αριθμός επαναλήψεων που απαιτούνται για την πλήρη εκπαίδευση του δικτύου,  $N$  ο αριθμός των επεξεργαστικών μονάδων και  $\delta$  ένας συντελεστής που εξαρτάται από το  $K$ . Τόσο το  $K$ , όσο και το  $\delta$ , ορίζονται από το νόμο του Gardner σχετικά με τις μεταβατικές φάσεις των διανυσμάτων των συντελεστών βάρους στο δίκτυο και τη μεταβολή τους σε κάθε επανάληψη. Ο πίνακας των διανυσμάτων των συντελεστών βάρους των μονάδων που τελικά προκύπτει είναι συνήθως μη-συμμετρικός.

Αποδεικνύεται ότι για  $K \rightarrow 0$ , ο νόμος του Gardner δείχνει ότι για τυχαίο σύνολο προτύπων, το μοντέλο συγκλίνει για  $\delta \cong N^{-\frac{1}{2}}$ , και μάλιστα ο χρόνος σύγκλισης είναι γραμμικός ως προς το πλήθος  $N$  των μονάδων στο δίκτυο (πολυωνυμικός σε κάθε περίπτωση).

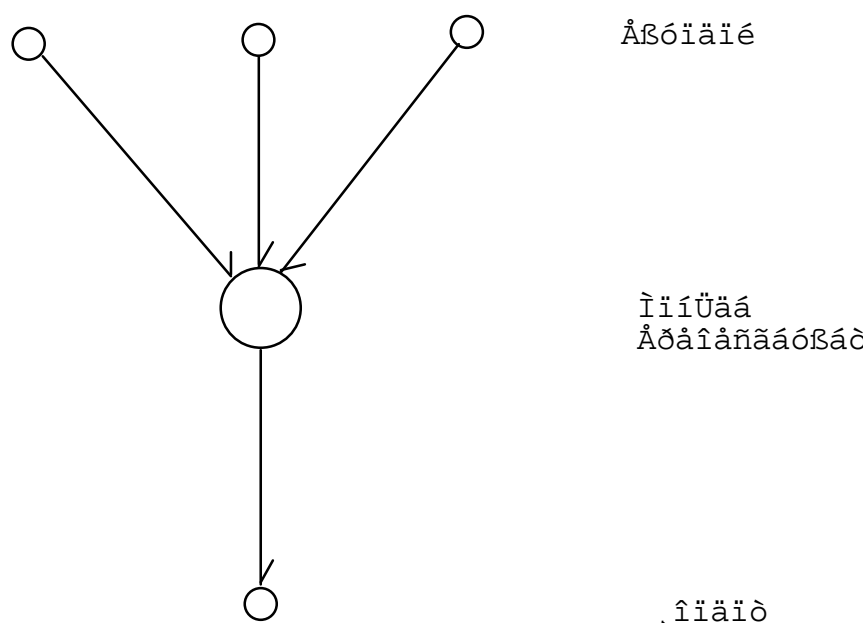
Για  $K > 0$ , το άνω όριο του  $T$  υπολογίζεται από την παραπάνω ανισότητα. Επίσης παρατηρούμε ότι για  $\delta \rightarrow 0$  δεν υπάρχει κανένα όριο στον αριθμό επαναλήψεων. Η παραπάνω σχέση δείχνει ότι το  $T$  είναι ανάλογο του πλήθους των μονάδων  $N$ . Τελικά, αποδεικνύεται ότι ο απαιτούμενος χρόνος εκπαίδευσης είναι γραμμικός ως προς  $N$ , γιατί η εκπαίδευση συντελείται ξεχωριστά σε κάθε μονάδα και μπορεί να γίνει παράλληλα, και ο τελικός

πίνακας διανυσμάτων των συντελεστών βάρους είναι συνήθως μη-συμμετρικός.

Τα Perceptron ΤΝΔ βρίσκουν εφαρμογή στην κατηγοριοποίηση σχημάτων, στην αναγνώριση χαρακτήρων και στην ρομποτική (robot vision systems).

- Adaline-Madaline

Το Adaline μοντέλο αναπτύχθηκε λίγο μετά τη διατύπωση του Perceptron ΤΝΔ και οφείλεται στον Windrow και στους συνεργάτες του. Το μοντέλο αυτό αποτελείται από ένα στρώμα, το οποίο περιέχει μια μόνο μονάδα. Κάθε είσοδος του δικτύου συνδέεται με τη μονάδα αυτή. Οι τιμές εισόδου είναι συνεχείς, και η μάθηση γίνεται με επίβλεψη.



**Σχήμα 6** - Δομή ADALINE μοντέλου ΤΝΔ

Μια επέκταση του Adaline μοντέλου είναι το Madaline, το οποίο περιέχει πολλές μονάδες στο στρώμα του δικτύου. Ο αλγόριθμος μάθησης του Adaline-Madaline μοντέλου ονομάζεται Least Mean Square (L.M.S.) και αναπτύχθηκε από τους Windrow και Hoff. Όπως δηλώνει και το όνομά του, η μάθηση επιτυγχάνεται ελαχιστοποιώντας το μέσο τετραγωνικό σφάλμα μεταξύ της πραγματικής και επιθυμητής κατάστασης ενεργοποίησης. Η



ελαχιστοποίηση αυτή πραγματοποιείται χρησιμοποιώντας μεθόδους καθόδου μέγιστης κλίσης (steepest descent) σε μια επιφάνεια η οποία αντιπροσωπεύει το χώρο των συντελεστών βάρους. Σε κάθε επανάληψη του LMS αλγορίθμου, οι συντελεστές βάρους τροποποιούνται προς την κατεύθυνση που καθορίζεται από το διάνυσμα κλίσης (gradient vector). Τα στάδια του LMS αλγορίθμου μάθησης είναι τα ακόλουθα:

Βήμα 1:

Αρχικά, τα βάρη  $w_{ij}$  παίρνουν μικρές τυχαίες τιμές, και τα εσωτερικά όρια  $\theta_i$  παίρνουν την τιμή +1

Βήμα 2:

Παρουσιάζεται κάποιο πρότυπο  $m(x_0, x_1, \dots, x_{N-1})$ , από το σύνολο προτύπων εκπαίδευσης στην είσοδο του δικτύου.

Βήμα 3:

Υπολογίζεται η κατάσταση ενεργοποίησης της μονάδας

$$a_i(t) = \sum_{j=0}^{N-1} w_{ij}(t)x_j(t) + \theta_i$$

Βήμα 4:

(α) Υπολογίζεται το δυαδικό αποτέλεσμα εξόδου  $o_i(t)$

$$o_i(t) = f_h(a_i(t))$$

(β) Οι συντελεστές βάρους των συνδέσμων αναπροσαρμόζονται σύμφωνα με τον παρακάτω κανόνα:

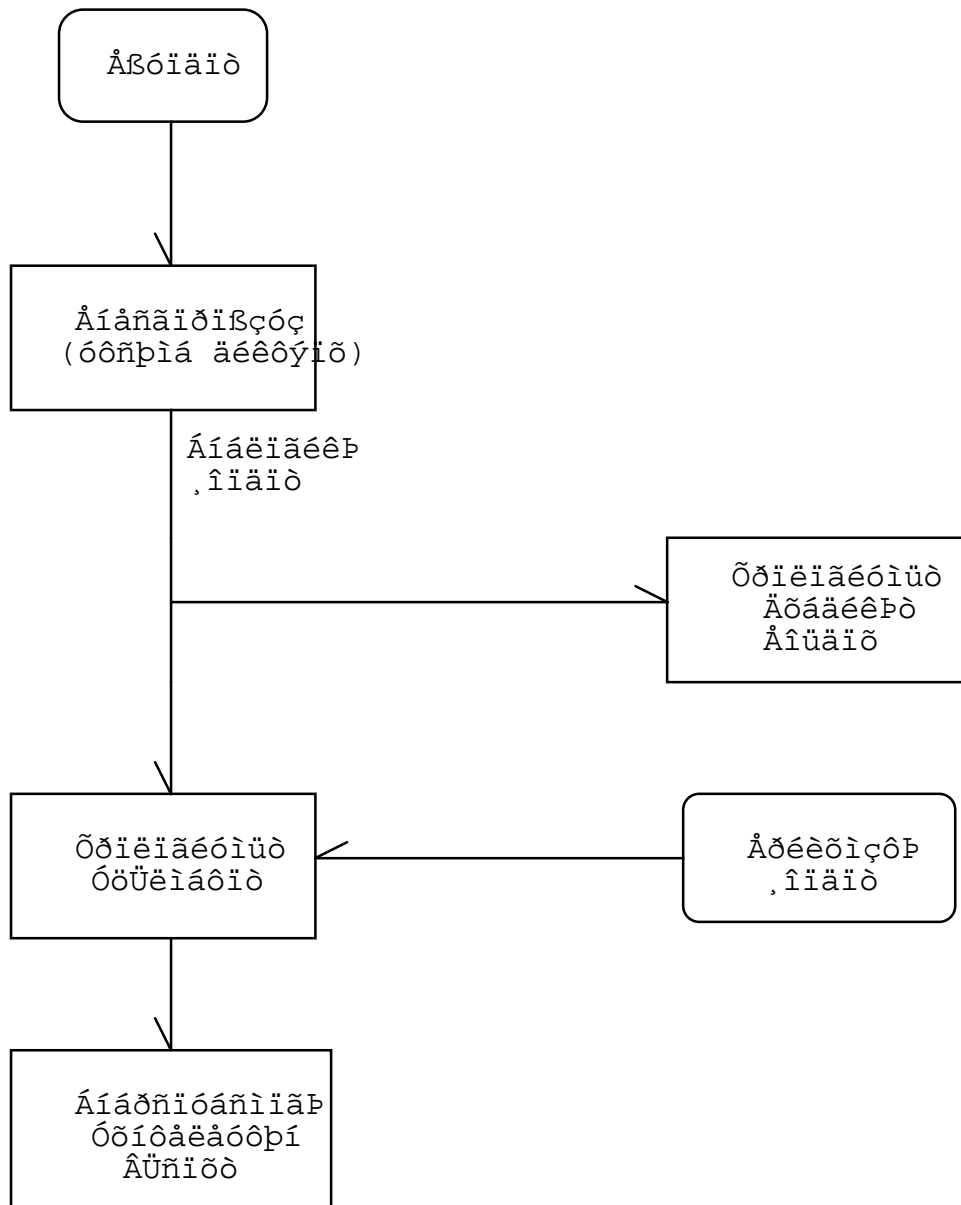
$$w_{ij}(t+1) = w_{ij}(t) + \varepsilon \frac{(t_i^m - a_i(t))x_j(t)}{\sum_{\rho=0}^{n-1} [x_{\rho}(t)]^2}$$

Βήμα 5:

Επανάληψη πηγαίνοντας στο βήμα 3, μέχρι σύγκλισης των συντελεστών βάρους  $w_{ij}$ , για το τρέχον πρότυπο.

Βήμα 6:

Επανάληψη, πηγαίνοντας στο βήμα 2, μέχρι μάθησης όλων των προτύπων εκπαίδευσης.



**Σχήμα 7** - Εκπαίδευση ADALINE TNA

Η κύρια διαφορά μεταξύ του Adaline-Madaline μοντέλου και του Perceptron είναι ότι στον υπολογισμό του σφάλματος, το Adaline-Madaline μοντέλο συγκρίνει την κατάσταση ενεργοποίησης της μονάδας με την επιθυμητή έξοδο. Αυτή η σύγκριση δίνει μια πιο ακριβή ένδειξη σφάλματος από ότι το Perceptron μοντέλο, το οποίο συγκρίνει τη δυαδική έξοδο με την επιθυμητή έξοδο. Επίσης, υπάρχουν και διαφορές στην αναπροσαρμογή των συντελεστών βάρους. Συγκεκριμένα, ο LMS αλγόριθμος, προκειμένου να επηρεάσει το μέγεθος της αλλαγής των συντελεστών βάρους, διαιρεί το σφάλμα με το εσωτερικό γινόμενο των εισόδων. Αυτή η αναπροσαρμογή, εξασφαλίζει ότι η επιθυμητή διόρθωση σφάλματος

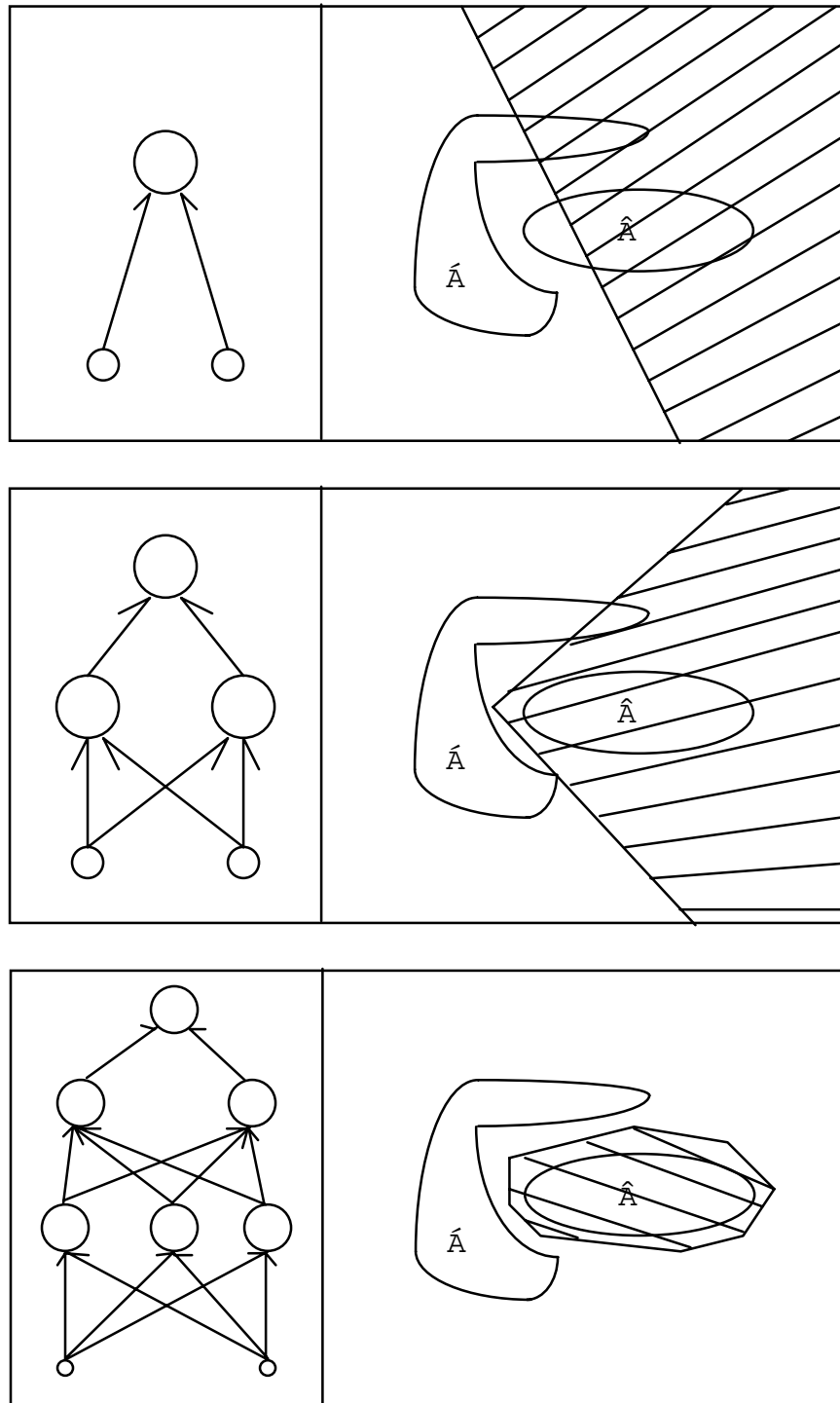
επιτυγχάνεται με τη μικρότερη αλλαγή τιμής στους συντελεστές βάρους. Επιπλέον, οι συντελεστές βάρους αναπροσαρμόζονται μέχρι σύγκλισης κρατώντας σταθερό κάποιο πρότυπο εκπαίδευσης. Πάντως, επειδή η αναπροσαρμογή του διανύσματος των συντελεστών βάρους είναι ανάλογη προς την πρώτη παράγωγο της συνάρτησης του σφάλματος, η σύγκλιση είναι σχετικά αργή όταν το δίκτυο βρίσκεται σε (σχεδόν) σταθερή κατάσταση. Τελικά, οι πληροφορίες που έχουν αποθηκευτεί στο δίκτυο από την εκπαίδευση προηγούμενων προτύπων αλλοιώνονται ελάχιστα.

Τα Adaline-Madaline ΤΝΔ εφαρμόζονται στον έλεγχο θορύβου (noise cancelling, jamming signals cancelling in antenna arrays), στη μελέτη μοντέλων καιρού (weather modelling, weather forecasting), αναγνώριση φωνής (speech recognition), αναγνώριση σεισμικών κυματομορφών (seismic signal pattern recognition), σε φίλτρα (filtering), σε ιατρικές εφαρμογές (blood pressure regulation), σε προβλήματα εξισορρόπησης (balancing examples), κ.ά.

- Multi-Layered Perceptron (MLP)

Τα Perceptron μοντέλα με ένα στρώμα είναι κατάλληλα για την ταξινόμηση προτύπων, των οποίων οι κλάσεις είναι γραμμικά διαχωρίσιμες. Όταν οι κλάσεις προτύπων δεν είναι γραμμικά διαχωρίσιμες, απαιτείται ένα πιο πολύπλοκο δίκτυο. Μια επέκταση του Perceptron μοντέλου με ένα στρώμα είναι το Perceptron μοντέλο με πολλά στρώματα (Multi-Layered Perceptron).

Το Multi-Layered Perceptron μοντέλο είναι ένα feed-forward δίκτυο που έχει τη δυνατότητα να αντιμετωπίζει τους περιορισμούς του Perceptron μοντέλου με ένα στρώμα, χρησιμοποιώντας επιπλέον ένα ή περισσότερα στρώματα μονάδων. Ενώ το Perceptron ενός στρώματος μπορεί να διαχωρίσει τις κατανομές δύο διαφορετικών κλάσεων δεδομένων γραμμικά, ένα Multi-Layered Perceptron με δύο στρώματα μονάδων μπορεί να τις διαχωρίσει δημιουργώντας κυρτές και πολυγωνικές περιοχές. Επιπλέον, μπορεί να αποδειχτεί ότι ένα Perceptron μοντέλο με τρία στρώματα είναι ικανό να δημιουργήσει μη-γραμμικά όρια οποιασδήποτε πολυπλοκότητας, μεταξύ κλάσεων προτύπων.



**Σχήμα 8** - Παράδειγμα περιοχών απόφασης MLP μοντέλου ΤΝΔ, για ένα, δύο και τρία στρώματα (layers) μονάδων, αντίστοιχα.

Οι ικανότητες των δικτύων αυτών προκύπτουν όταν οι συναρτήσεις εξόδου των μονάδων είναι μη-γραμμικές. Εάν χρησιμοποιηθούν γραμμικές συναρτήσεις εξόδου, τότε ένα κατάλληλα επιλεγμένο ΤΝΔ με ένα στρώμα μπορεί να πραγματοποιήσει τους υπολογισμούς οποιουδήποτε ΤΝΔ μοντέλου

με πολλά στρώματα. Σε ένα ΤΝΔ με πολλά στρώματα, κάθε μονάδα ξεχωριστά έχει τη δυνατότητα να πραγματοποιεί μία μόνο γραμμικά διαχωρίσιμη συνάρτηση. Ο συνδυασμός των χαμηλότερων επιπέδων εξόδου (κρυμμένα στρώματα) επιτρέπει στο δίκτυο να υπολογίζει μη-γραμμικές λογικές συναρτήσεις.

Οι τιμές εισόδου στο μοντέλο αυτό είναι συνεχείς, και η μάθηση γίνεται με επίβλεψη. Οι αλγόριθμοι εκπαίδευσης για ΤΝΔ με πολλά στρώματα (Multi-Layered Perceptrons) διαιρούνται σε δύο φάσεις:

- (α) Τη φάση της ανάκλησης (Retrieving)
- (β) Τη φάση μάθησης (Learning).

Στη φάση ανάκλησης του αλγορίθμου, οι πληροφορίες ρέουν από την είσοδο του δικτύου στην έξοδο του διαμέσου των κρυμμένων στρωμάτων. Σε αυτή τη φάση, οι μονάδες υπολογίζουν τις νέες καταστάσεις ενεργοποίησης και τις τιμές εξόδου τους. Στη φάση μάθησης, πραγματοποιείται η αναπροσαρμογή των συντελεστών βάρους. Οι φάσεις ανάκλησης και μάθησης λειτουργούν διαδοχικά, η μια μετά την άλλη, έως ότου οι συντελεστες βάρους αποκτήσουν τιμές που θα επιτρέπουν στο δίκτυο να πραγματοποιεί την επιθυμητή ταξινόμηση.

Στο παρελθόν, τέτοια μοντέλα είχαν ελάχιστη πρακτική αξία, λόγω έλλειψης κατάλληλων αλγορίθμων εκπαίδευσης. Πρόσφατα, έχουν αναπτυχθεί αρκετοί επιτυχείς αυτόματοι αλγόριθμοι εκπαίδευσης, μεταξύ των οποίων ο πιο διαδεδομένος είναι ο Back-Propagation.

#### • Back-Propagation

Ο Back-Propagation αλγόριθμος εκπαίδευσης επεκτείνει την ιδέα της ελαχιστοποίησης του σφάλματος του LMS αλγορίθμου σε δίκτυα πολλαπλών στρωμάτων. Όταν χρησιμοποιείται για την εκπαίδευση κάποιου ΤΝΔ, τότε οι συναρτήσεις εξόδου των μονάδων του ΤΝΔ πρέπει να είναι συνεχείς και διαφορίσιμες μη-γραμμικές. Η διαφορισιμότητα των συναρτήσεων εξόδου επιτρέπει τον υπολογισμό των μερικών παραγώγων, που απαιτούνται για τον προσδιορισμό των σφαλμάτων κατά την αναπροσαρμογή των συντελεστών βάρους. Ο αλγόριθμος εκπαίδευσης περιγράφεται παρακάτω:

##### Βήμα 1:

Τα βάρη  $w_{ij}$  και τα εσωτερικά όρια  $\theta_i$  παίρνουν μικρές τυχαίες τιμές.

##### Βήμα 2:

Παρουσιάζεται κάποιο πρότυπο  $m(x_0, x_1, \dots, x_{N-1})$  από το σύνολο προτύπων εκπαίδευσης.

**Βήμα 3 (Φάση Ανάκλησης - Retrieving):**

Υπολογίζονται τα αποτελέσματα εξόδου σε κάθε στρώμα

$$a_i^{(1)}(t) = \sum_{j=1}^{N_0} w_{ij}^{(1)}(t) o_j^{(0)}(t) + \theta_i^{(1)}, \quad o_i^{(1)} = f_h(a_i^{(1)})(t), \quad 1 \leq i \leq N_1$$

$$a_i^{(2)}(t) = \sum_{j=1}^{N_1} w_{ij}^{(2)}(t) o_j^{(1)}(t) + \theta_i^{(2)}, \quad o_i^{(2)} = f_h(a_i^{(2)})(t), \quad 1 \leq i \leq N_2$$

.....

$$a_i^{(L)}(t) = \sum_{j=1}^{N_{L-1}} w_{ij}^{(L)}(t) o_j^{(L-1)}(t) + \theta_i^{(L)}, \quad o_i^{(L)} = f_h(a_i^{(L)})(t), \quad 1 \leq i \leq N_L$$

**Βήμα 4 (Φάση Μάθησης - Learning):**

Οι συντελεστές βάρους των συνδέσμων μεταξύ δύο γειτονικών στρωμάτων  $(l, l-1)$  αναπροσαρμόζονται, ξεκινώντας από το στρώμα εξόδου, έως το στρώμα εισόδου (δηλαδή:  $l = L, L-1, \dots, 1$ ), όπως παρακάτω:

$$w_{ij}^{(l)}(t+1) = w_{ij}^{(l)}(t) + \varepsilon \delta_i^{(l)}(t) o_j^{(l-1)}(t)$$

όπου  $\delta_i^{(l)}(t)$  είναι το σήμα σφάλματος και προσδιορίζεται ως εξής:

$$\delta_i^{(L)}(t) = (t_i^m - o_i^{(L)}(t)) f_h'(a_i^{(L)}(t))$$

$$\delta_i^{(l)}(t) = f_h'(a_i^{(l)}(t)) \sum_j \delta_j^{(l+1)}(t) w_{ij}^{(l+1)}(t)$$

**Βήμα 5:**

Επανάληψη πηγαίνοντας στο Βήμα 2, μέχρι μάθηση όλων των προτύπων εκπαίδευσης.

$f_h$  : η μη-γραμμική συνάρτηση εξόδου

$o_i^{(l)}(t)$  : ( $l = 1, 2, \dots, L$ ) αποτέλεσμα εξόδου της μονάδας  $i$  του στρώματος  $l$

$o_i^{(0)}(t) = x_i$  : τρέχον πρότυπο εισόδου

$a_i^{(l)}(t)$  : κατάσταση ενεργοποίησης της μονάδας  $i$  του στρώματος  $l$

$w_{ij}^{(l)}(t)$  : συντελεστής βάρους μεταξύ της μονάδας  $j$  του στρώματος  $l-1$  και της μονάδας  $i$  του στρώματος  $l$

$\varepsilon$  : ρυθμός μάθησης,  $0 < \varepsilon < 1$

$t_i^{(m)}$  : επιθυμητή έξοδος της μονάδας  $i$  όταν το πρότυπο  $m$  παρουσιάζεται στο δίκτυο

Η επιλογή της συνάρτησης εξόδου παίζει σπουδαίο ρόλο στον Back-Propagation αλγόριθμο εκπαίδευσης. Συνήθως, η συνάρτηση εξόδου είναι η sigmoid οριακή συνάρτηση, δηλαδή

$$o_i^{(l)}(t) = f_h(a_i^{(l)}(t)) = \frac{1}{1+e^{-a_i^{(l)}(t)}}$$

Τότε, τα σήματα σφάλματος  $\delta_i^{(l)}(t)$  παράγονται ως εξής:

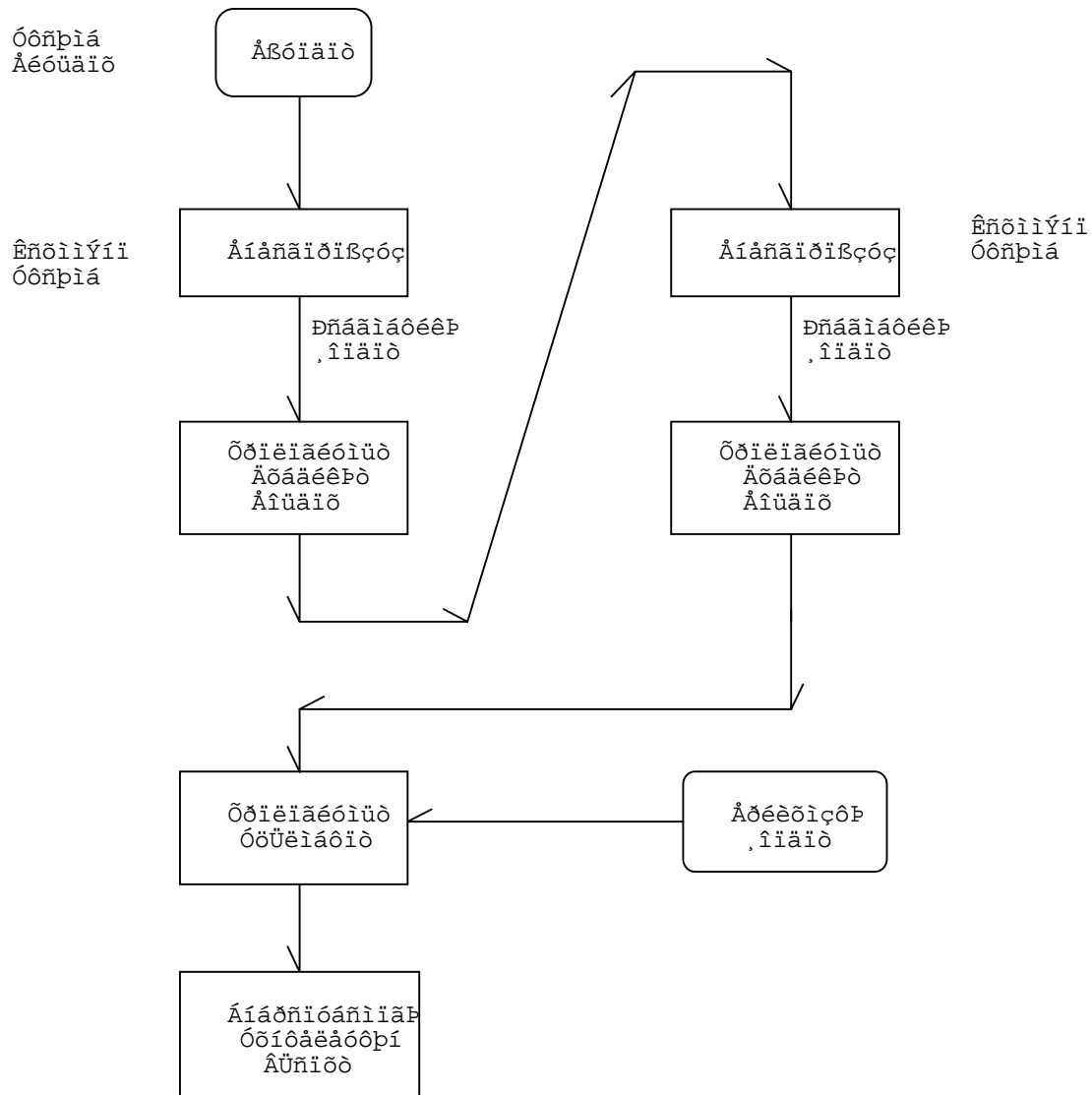
$$\delta_i^{(L)}(t) = (t_i^m - o_i^{(L)}(t))o_i^{(L)}(t)(1 - o_i^{(L)}(t))$$

$$\delta_i^{(l)}(t) = o_i^{(l)}(t)(1 - o_i^{(l)}(t)) \sum_j \delta_j^{(l+1)}(t) w_{ji}^{(l+1)}(t)$$

Τα εσωτερικά όρια  $\theta_i^{(l)}$  μπορούν να αναπροσαρμοστούν παρόμοια με του συντελεστές βάρους. Μπορούμε να θεωρήσουμε απλά, ότι τα  $\theta_i^{(l)}$  είναι ο συντελεστής βάρους μίας μονάδας η οποία έχει πάντα σταθερή τιμή εισόδου. Είναι εύκολο να αποδειχθεί ότι τα εσωτερικά όρια μπορούν να αναπροσαρμοστούν σύμφωνα με την παρακάτω εξίσωση:

$$\theta_i^{(l)}(t+1) = \theta_i^{(l)}(t) + \varepsilon \delta_i^{(l)}(t)$$





**Σχήμα 10** - Εφαρμογή του Back-Propagation αλγορίθμου σε MLP ΤΝΔ

Για το πρώτο βήμα του αλγορίθμου εκπαίδευσης, έχουμε αρχικοποίηση  $W = \sum_{i=0}^{L-1} (N_i \cdot N_{i+1})$  συντελεστών βάρους και  $\sum_{i=0}^{L-1} N_i$  εσωτερικών ορίων των μονάδων. Το  $W$  υπολογίζεται ως εξής: Θεωρώντας ότι το πλήθος των μονάδων  $N_i$  ανά στρώμα είναι διακριτή τυχαία μεταβλητή, ανεξάρτητη από τα υπόλοιπα  $N_j$ , τότε :

$$E(N_i N_{i+1}) = E(N_i) E(N_{i+1}) = E^2(N_i) = \bar{N}^2$$

Συνεπώς  $W = L \cdot \bar{N}^2$  και έχουμε  $O(W) = O(L \cdot \bar{N}^2)$ . Για το δεύτερο βήμα έχουμε  $O(1)$ . Στο τρίτο βήμα έχουμε υπολογισμό  $N_i$  εξόδων για κάθε ένα από τα  $L$  στρώματα του δικτύου. Έτσι, όπως και στο βήμα 1, έχουμε  $O(W) = O(L \cdot \bar{N}^2)$ . Στο βήμα 4 έχουμε υπολογισμό

$N_i \cdot N_{i+1}$  συντελεστών βάρους για κάθε ένα από τα  $L$  στρώματα του δικτύου, ενώ συγχρόνως απαιτείται ο υπολογισμός του συντελεστή  $\delta_i$  για κάθε μια μονάδα, που είναι άθροισμα  $N_i$  ορων, δηλαδή συνολικά

$$R = \sum_{i=0}^{L-1} ((N_i \cdot N_{i+1}) \cdot N_i)$$

Όπως και στο βήμα 1, θεωρώντας τα  $N_i$  (διακριτές) ανεξάρτητες τυχαίες μεταβλητές, προκύπτει ότι:

$$R = \sum_{i=0}^{L-1} (N_i^2 N_{i+1}) = \sum_{i=0}^{L-1} \bar{N}^3 = L\bar{N}^3$$

και η πολυπλοκότητα είναι  $O(R) = O(L\bar{N}^3)$ , που είναι και η μέγιστη πολυπλοκότητα για κάθε επανάληψη. Η διαδικασία επαναλαμβάνεται για κάθε ένα από τα  $M$  πρότυπα εισόδου, ενώ θεωρώντας  $\tau$  το μέσο αριθμό εκπαιδευτικών κύκλων που απαιτούνται μέχρι τη σύγκλιση, η συνολική πολυπλοκότητα του αλγορίθμου εκπαίδευσης είναι το πολύ  $O(\tau LM\bar{N}^3)$

Συνήθως, απαιτείται ένας μεγάλος αριθμός επαναλήψεων για τη σύγκλιση των συντελεστών βάρους. Πειραματικά αποτελέσματα έχουν δείξει ότι ο απαιτούμενος αριθμός επαναλήψεων συνδέεται άμεσα με το αριθμό των κρυμμένων στρωμάτων και είναι αντιστρόφως ανάλογος με τον αριθμό των κρυμμένων μονάδων ανά στρώμα.

Ο Back-Propagation αλγόριθμος είναι εξαιρετικά χρονοβόρος. Έχει αποδειχθεί ότι ο χρόνος εκπαίδευσης είναι περίπου  $O(W^3)$  σε ακολουθιακές μηχανές (serial machines), ενώ σε παράλληλες μηχανές (parallel machines) είναι  $O(W^2)$ , με την προϋπόθεση ότι για κάθε σύνδεση χρησιμοποιείται ξεχωριστός επεξεργαστής, όπου  $W$  είναι ο συνολικός αριθμός βαρών που υπάρχουν στο δίκτυο. Ο αριθμός των προτύπων εκπαίδευσης είναι  $O(W)$ . Τέλος, οι περισσότερες προσομοιώσεις έχουν δείξει ότι απαιτούνται 1000 με 10000 επαναλήψεις του συνόλου εκπαίδευσης (κύκλοι) για σύγκλιση. Αξίζει να σημειωθεί ότι στις περισσότερες εφαρμογές χρησιμοποιούνται MLP ΤΝΔ με τρία στρώματα.

Σύμφωνα με τα παραπάνω, στις περισσότερες εφαρμογές που χρησιμοποιούν Back-Propagation αλγορίθμους εκπαίδευσης, η πραγματική πολυπλοκότητα (ως προς τον αριθμό των μονάδων) του δικτύου είναι  $O(c\bar{N}^5)$ , όπου  $c \approx \tau L^2$ .

- Βελτιωμένοι Back-Propagation αλγόριθμοι

Το υψηλό χρονοβόρο κόστος εκπαίδευσης μεγάλων δικτύων χρησιμοποιώντας τον Back-Propagation, θέτει ένα πρακτικό όριο στο μέγεθος του δικτύου. Συγκεκριμένα, πρόσφατη έρευνα έχει δείξει ότι το γενικό πρόβλημα εκμάθησης της αντιστοιχίας εισόδου-εξόδου μιας λογικής (boolean) συνάρτησης, χρησιμοποιώντας τον Back-Propagation, είναι NP-Complete πρόβλημα και δεν υπάρχει γνωστή αποδοτική λύση.

Έχουν αναπτυχθεί δύο γενικοί τρόποι προσέγγισης για την βελτίωση της αποδοτικότητας.

Η πρώτη χρησιμοποιεί μη-γραμμικούς όρους στο υπολογισμό της κατάστασης ενεργοποίησης μιας μονάδας. Δηλαδή, ο υπολογισμός της κατάστασης ενεργοποίησης πραγματοποιείται, όχι ως άθροισμα πρώτης τάξεως γινομένων, αλλά χρησιμοποιούνται και υψηλότερου βαθμού γινόμενα. Μια τέτοια προσέγγιση επιτρέπει σε μία μονάδα να υπολογίζει μη-γραμμικές περιοχές απόφασης, ώστε να μην υπάρχει ανάγκη για τη χρησιμοποίηση δικτύων με πολλαπλά στρώματα. Παρ' όλα αυτά, ΤΝΔ μοντέλα με μονάδες αυτής της μορφής χρησιμοποιούν αρχιτεκτονικές δικτύων με πολλαπλά στρώματα.

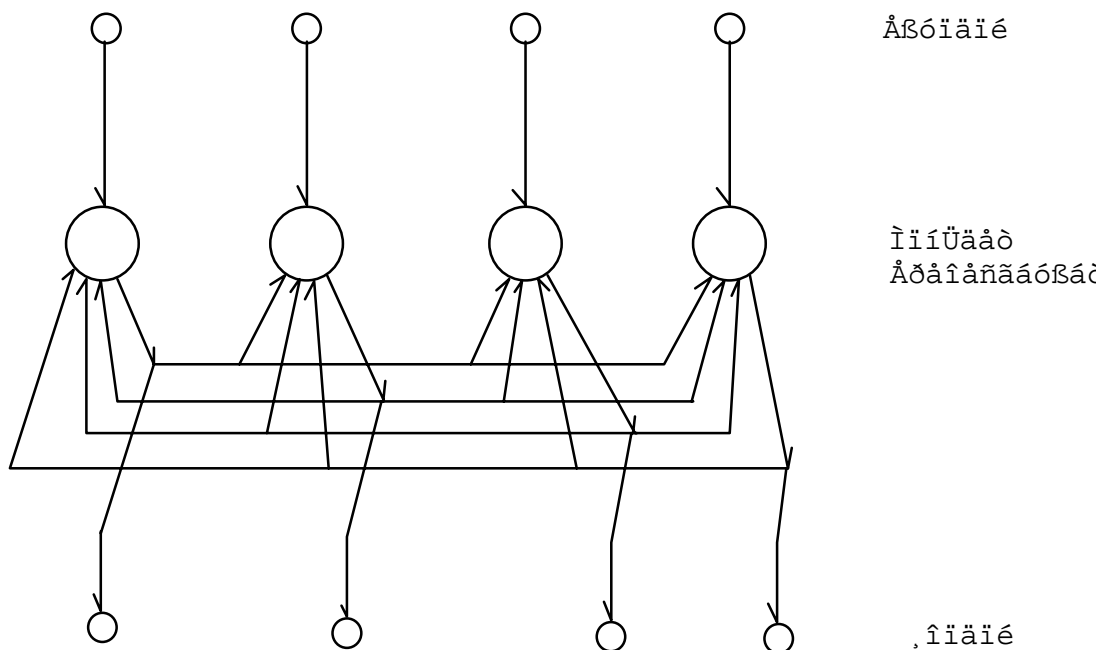
Η δεύτερη προσέγγιση αποτελείται από μεθόδους που βελτιώνουν την αποδοτικότητα του Back-Propagation, αλλάζοντας τον ίδιο τον αλγόριθμο και διατηρώντας την γραμμικότητα του υπολογισμού της κατάστασης ενεργοποίησης των μονάδων του δικτύου. Σε τέτοιες περιπτώσεις επιπρόσθετες πληροφορίες χρησιμοποιούνται από τους βελτιωμένους αλγορίθμους, για την αποφυγή μη αποδοτικών ερευνών στο χώρο των συντελεστών βάρους για την έρευνα ολικού ελαχίστου της συνάρτησης σφάλματος. Αυτό σημαίνει ότι το ολικό ελάχιστο μπορεί να προσεγγισθεί ταχύτερα. Πρέπει να σημειωθεί ότι οι τεχνικές που χρησιμοποιούνται σε αυτούς τους αλγορίθμους είναι γενικά "ευρηστικές".

Τα Multi-Layered Perceptron μοντέλα με Back-Propagation αλγορίθμους εκπαίδευσης, είναι αρκετά χρήσιμα σε εφαρμογές σχετικά με την εκτίμηση του ρίσκου δανειοδότησης, σύνθεση ομιλίας από κείμενο, στη ρομποτική (βραχίονες), επεξεργασία εικόνας, στην πρόβλεψη (οικονομικά και τακτικά-στρατιωτικά μοντέλα), κ.ά.

- Hopfield

Τα είδη των ΤΝΔ που αναφέρθηκαν μέχρι τώρα αποτελούνται από στρώματα μονάδων, όπου μονάδες από κάποιο στρώμα συνδέονται μόνο με μονάδες σε επόμενα στρώματα. Πιο πολύπλοκα δίκτυα περιλαμβάνουν συνδέσμους μεταξύ των μονάδων στο ίδιο στρώμα, καθώς επίσης και σε προηγούμενα στρώματα. Η ανάλυση της συμπεριφοράς τέτοιων Τ.Ν.Δ. είναι πολύ δύσκολη.

Μια ενδιαφέρουσα ειδική περίπτωση των μοντέλων αυτών είναι το μοντέλο Hopfield. Το δίκτυο αυτό αποτελείται από ένα στρώμα εξόδου, που είναι ταυτόχρονα και στρώμα εξόδου. Κάθε έξοδος των κόμβων συνδέεται με όλους τους υπόλοιπους κόμβους. Αυτή η "αναδρομικότητα" προσδίδει στο μοντέλο μη-γραμμικότητα. Το Hopfield ΤΝΔ χρησιμοποιείται συνήθως με δυαδικά δεδομένα, αλλά μπορούν να χρησιμοποιηθούν και συνεχή δεδομένα, με τη μετατροπή τους σε δυαδικά, μέσω κάποιας κατάλληλης αναπαράστασης.



**Σχήμα 11** - Δομή Hopfield μοντέλου ΤΝΔ

Η εκπαίδευση των ΤΝΔ μοντέλων που έχουμε εξετάσει μέχρι τώρα, πραγματοποιείται με την αναπροσαρμογή των συντελεστών βάρους. Αντίθετα, στο μοντέλο του Hopfield, οι συντελεστές βάρους καθορίζονται αυτόματα. Έτσι, το μοντέλο αυτό δεν έχει την ικανότητα εκπαίδευσης και χρησιμοποιεί αυτο-association μάθηση. Το αποτέλεσμα εξόδου του δικτύου καθορίζεται μετά από ένα

πεπερασμένα αριθμό επαναλήψεων. Η λειτουργία του βασίζεται στον παρακάτω αλγόριθμο:

Βήμα 1:

Καθορίζονται οι συντελεστές βάρους στους συνδέσμους:

$$w_{ij} = \sum_{m=0}^{M-1} x_i^m x_j^m, \text{ αν } i \neq j$$

$$w_{ij} = 0, \text{ αν } i = j$$

όπου:  $N$ =αριθμός των κόμβων,  $M$ =αριθμός των προτύπων,  $0 \leq i, j \leq N-1$  και  $x_i^m$ = $i$ -οστό στοιχείο του  $m$  προτύπου.

Βήμα 2:

Εισαγωγή προτύπου  $m$  ( $x_0, x_1, \dots, x_{n-1}$ ) στους κόμβους του δικτύου  $o_i(0) = x_i$ , για  $0 \leq i \leq N-1$

Βήμα 3:

Υπολογίζεται επαναληπτικά το αποτέλεσμα εξόδου σε κάθε κόμβο

$$o_j(t) = \sum_{i=0}^{N-1} w_{ij} o_i(t)$$

$$o_j(t+1) = f_h(o_j(t))$$

μέχρι  $o_j(t+1) = o_j(t)$ .

Η  $f_h$  συμβολίζει τη hardlimiter οριακή συνάρτηση.

Σε κάθε επανάληψη, το αποτέλεσμα εξόδου κάθε κόμβου γίνεται είσοδος σε όλους τους υπόλοιπους κόμβους του δικτύου. Οι επαναλήψεις συνεχίζονται, εως ότου το αποτέλεσμα εξόδου τη χρονική στιγμή  $t+1$  είναι το ίδιο με το αποτέλεσμα εξόδου τη στιγμή  $t$  (σύγκλιση).

Παρατηρούμε ότι στο πρώτο βήμα έχουμε υπολογισμό  $N$  συντελεστών βάρους για κάθε μια από τις  $N$  μονάδες, χρησιμοποιώντας όλα τα  $M$  πρότυπα. Έτσι, κατά τη φάση αυτή, η πολυπλοκότητα είναι  $O(MN^2)$ . Κατά το δεύτερο βήμα δεν έχουμε επανάληψη, άρα  $O(1)$ . Στο τρίτο βήμα θεωρούμε  $\tau$  τον αριθμό επαναλήψεων μέχρι τη σταθεροποίηση των εξόδων, και έχουμε  $N$  υπολογισμούς τιμών για επανατροφοδότηση (feedback) για κάθε έναν από τους  $N$  κόμβους,  $\tau$  φορές. Συνεπώς η συνολική πολυπλοκότητα του αλγορίθμου είναι  $O(RN^2)$ , όπου  $R = \max\{M, \tau\}$ .

Ο Hopfield και άλλοι έχουν αποδείξει ότι αυτός ο αλγόριθμος συγκλίνει όταν οι συντελεστές βάρους είναι συμμετρικοί ( $w_{ij} = w_{ji}$ ) και οι έξοδοι των κόμβων αναπροσαρμόζονται ασύγχρονα χρησιμοποιώντας τις εξισώσεις του βήματος 3. Η επαναληπτική διαδικασία του βήματος 3 μπορεί να παρουσιαστεί ως η ελαχιστοποίηση της "συνάρτησης ενέργειας" του συστήματος, χρησιμοποιώντας τη μέθοδο καθόδου μέγιστης κλίσης. Συγκεκριμένα, ο Hopfield έχει δείξει ότι αν  $w_{ij} = w_{ji}$  και  $w_{ii} = 0$ , τότε το δίκτυο αλλάζει πάντα τις καταστάσεις ενέργειάς του κατά τέτοιο τρόπο ώστε να ελαχιστοποιεί μια συνάρτηση ενέργειας η οποία ορίζεται ως εξής:

$$E(t) = -\frac{1}{2} \sum_{i=0}^{N-1} \sum_{j=1}^{N-1} w_{ij} o_i(t) o_j(t) - \sum_{i=0}^{N-1} \theta_i o_i(t)$$

όπου  $\theta_i$  το εσωτερικό όριο της μονάδας  $i$

Τα  $\theta_i$  είναι μηδέν στην μορφή του Hopfield αλγορίθμου που περιγράφεται εδώ, αλλά γενικά σε άλλες μορφές του αλγορίθμου είναι μη-μηδενικά και έχουν αρκετή σημασία.

Το Hopfield TND χρησιμοποιείται συνήθως σαν μνήμη συμπραζομένων, για την ταξινόμηση δεδομένων και για την λύση προβλημάτων βελτιστοποίησης. Χρησιμοποιούμενο σαν μνήμη συμπραζομένων, τα δεδομένα, τα οποία αποθηκεύονται στο δίκτυο, αντιστοιχούν σε τοπικά ελάχιστα της συνάρτησης ενέργειας. Κάθε δεδομένο εισόδου αντιστοιχεί σε κάποιο σημείο της συνάρτησης αυτής. Το δίκτυο συγκλίνει στο ελάχιστο της συνάρτησης ενέργειας που βρίσκεται πιο κοντά στο σημείο αυτό. Ο αριθμός των δεδομένων που έχει την ικανότητα να αποθηκεύσει το δίκτυο περιορίζεται από τον αριθμό των κόμβων του, και μάλιστα αν  $D$  είναι το πλήθος των δεδομένων και  $N$  ο αριθμός των μονάδων του δικτύου, τότε ισχύει:  $D_{\max} = 0.146N$ , με την προϋπόθεση ότι τα δεδομένα που αποθηκεύονται δεν έχουν πολλά σημεία. Αυτή η προϋπόθεση μπορεί να ικανοποιηθεί πολλές φορές χρησιμοποιώντας τεχνικές ορθο-κανονικοποίησης.

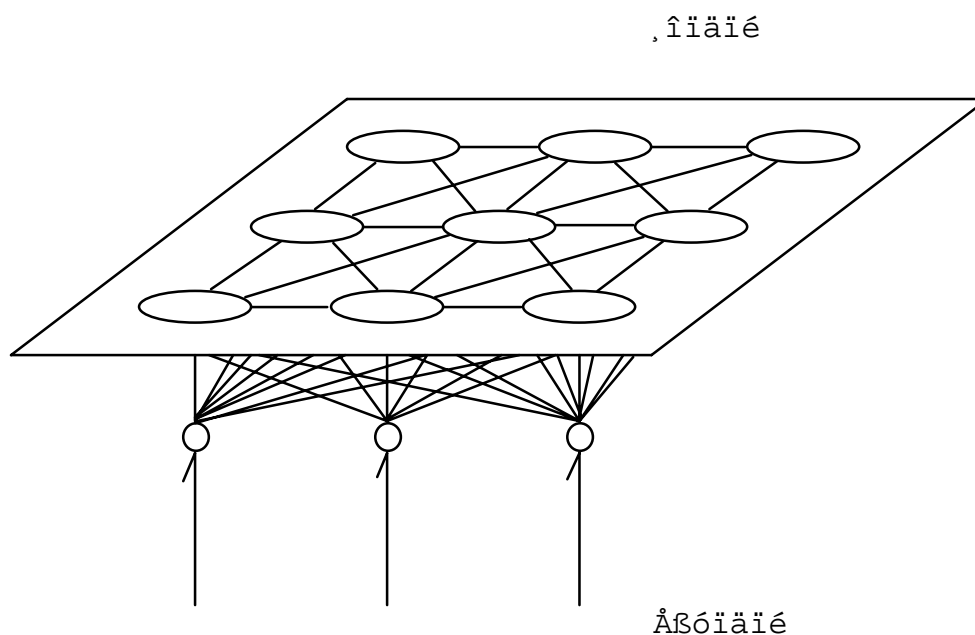
Για την λύση προβλημάτων βελτιστοποίησης (optimization) σκοπός είναι η έκφραση της συνάρτησης ελαχιστοποίησης του προβλήματος στη μορφή της συνάρτησης ενέργειας του δικτύου. Η βέλτιστη λύση του προβλήματος αντιστοιχεί στο ολικό ελάχιστο της συνάρτησης ενέργειας του δικτύου.

Άλλες πιθανές εφαρμογές του Hopfield TND είναι στη ρομποτική (robotic control systems), στις στρατιωτικές εφαρμογές (target/friend-or-foe identification), στην πλήρη ανάκτηση

δεδομένων από δείγμα και στην μελέτη NP-Complete προβλημάτων, όπως το γνωστό TSP.

- Kohonen

Αυτός ο ΤΝΔ αλγόριθμος αρχικά είχε προταθεί για την παραγωγή αυτοπροσαρμοζόμενων χαρτών χαρακτηριστικών (self-organizing feature maps), παρόμοιων μ'αυτούς που εμφανίζονται στον εγκέφαλο. Το Kohonen μοντέλο, χρησιμοποιούμενο ως αλγόριθμος ταξινόμησης, λειτουργεί παρόμοια με τον κλασικό αλγόριθμο ταξινόμησης K-Means. Αυτό το ΤΝΔ αποτελείται από ένα στρώμα, όπου οι μονάδες εξόδου είναι διατεταγμένες συνήθως σε ένα δισδιάστατο πλέγμα. Κάθε είσοδος συνδέεται με όλες τις μονάδες εξόδου. Ο αριθμός των εισόδων καθορίζεται από τη διάσταση των προτύπων εκπαίδευσης, ενώ ο αριθμός των εξόδων από τον επιθυμητό αριθμό των κλάσεων. Η εκπαίδευση γίνεται χωρίς επίβλεψη.



**Σχήμα 12** - Δομή Kohonen μοντέλου ΤΝΔ

Πρότυπα εκπαίδευσης, τα οποία μπορεί να περιέχουν συνεχείς τιμές, παρουσιάζονται τυχαία στις εισόδους του δικτύου, χωρίς να καθορίζεται η επιθυμητή έξοδος. Αφού έχουν παρουσιαστεί αρκετά πρότυπα εκπαίδευσης, οι συντελεστές βάρους των συνδέσμων καθορίζουν τα κέντρα των κλάσεων που σχηματίζουν τα δεδομένα εισόδου. Με άλλα λόγια, μετά την εκπαίδευση του δικτύου, τα αντιπροσωπευτικά παραδείγματα κάθε κλάσης αποθηκεύονται στις συνάψεις εισόδου-εξόδου.



Ακολουθεί η περιγραφή του αλγορίθμου:

Βήμα 1:

Τα βάρη  $w_{ij}$  παίρνουν μικρές τυχαίες τιμές. Επίσης, καθορίζονται οι τιμές της ακτίνας γειννίασης, καθώς και του συντελεστή μάθησης  $\varepsilon$ .

Βήμα 2:

Παρουσιάζεται τυχαία κάποιο πρότυπο  $(x_0, x_1, \dots, x_{n-1})$  από το σύνολο προτύπων εκπαίδευσης στην είσοδο του δικτύου.

$$o_i(0) = x_i, \text{ για } 0 \leq i \leq N-1$$

Βήμα 3:

Υπολογίζονται οι αποστάσεις  $d_j$  μεταξύ των εισόδων και κάθε κόμβου εισόδου

$$d_j = \sum_{i=0}^{N-1} (o_i(t) - w_{ij}(t))^2$$

Βήμα 4:

Επιλέγεται ο κόμβος εξόδου  $j^*$  με τη μικρότερη απόσταση  $d_j$ .

Βήμα 5:

Αναπροσαρμόζονται οι συντελεστές βάρους του κόμβου  $j^*$ , καθώς και των κόμβων που βρίσκονται μέσα στη γειτονιά  $N_{j^*}$  του κόμβου  $j^*$  που ορίζεται από την ακτίνα γειννίασης

$$w_{ij}(t+1) = w_{ij}(t) + \varepsilon (o_i(t) - w_{ij}(t))$$

για όλα τα  $j$  που ανήκουν στη  $N_{j^*}$  και για  $0 \leq i \leq N-1$ .

Βήμα 6:

Επανάληψη, πηγαίνοντας στο βήμα 2, μέχρι τη σύγκλιση.

Στα δύο πρώτα βήματα του αλγορίθμου έχουμε πολυπλοκότητα  $O(1)$ . Στο τρίτο βήμα, έχουμε υπολογισμό  $N$  αποστάσεων, κάθε μια από τις οποίες απαιτεί άθροισμα  $N$  όρων, συνεπώς έχουμε πολυπλοκότητα  $O(N^2)$ . Το τέταρτο βήμα έχει πολυπλοκότητα  $O(N)$ , επειδή η τυχαία κατανομή των αποστάσεων απαιτεί αναγκαστικά γραμμική αναζήτηση για την ελάχιστη τιμή. Στο βήμα 5,

θεωρώντας  $R$  το μέσο αριθμό μονάδων σε κάθε περιοχή γειτνίασης, η πολυπλοκότητα είναι ανάλογη του γινομένου  $R$  κόμβων επί  $N$  συντελεστές βάρους που αναπροσαρμόζονται, και επιπλέον το χρόνο εύρεσης των μονάδων που βρίσκονται μέσα στην περιοχή γειτνίασης. Για το πρώτο μέρος έχουμε πολυπλοκότητα  $O(RN)$ , ενώ για το δεύτερο, ο αλγόριθμος εύρεσης των μονάδων απαιτεί στην καλύτερη περίπτωση  $O(R)$ . Έτσι για το βήμα αυτό η συνολική πολυπλοκότητα είναι  $O(RN)$ . Τέλος, ο αριθμός επαναλήψεων  $L$  ορίζεται ως  $L = \max\{M, \tau\}$ , όπου  $M$  είναι το συνολικό πλήθος των προτύπων και  $\tau$  ο απαιτούμενος αριθμός των επαναλήψεων μέχρι τη σύγκλιση. Παρατηρώντας, τέλος, ότι το  $R$  είναι συνάρτηση της ακτίνας γειτνίασης και  $R \leq N$  σε κάθε περίπτωση, συμπεραίνουμε ότι η συνολική πολυπλοκότητα του αλγορίθμου είναι  $O(LN^2)$ .

Το κυριότερο χαρακτηριστικό του Kohonen T.N.Δ. είναι ότι ο αριθμός των επιθυμητών κλάσεων καθορίζεται από την αρχή της διαδικασίας εκπαίδευσης. Επιπλέον, η εκπαίδευση αυτού του μοντέλου δεν απαιτεί να είναι γνωστή η σωστή απάντηση για ένα πρότυπο εισόδου και προσομοιάζει αρκετά τα πραγματικά νευροβιολογικά συστήματα. Ο αλγόριθμος αυτός εξαρτάται από δύο παραμέτρους, την ακτίνα γειτνίασης  $N_j$  και το συντελεστή μάθησης  $\varepsilon$ , οι οποίοι συνήθως καθορίζονται εμπειρικά. Η ακτίνα γειτνίασης καθορίζει σε ποιούς από τους κόμβους εξόδου, οι οποίοι βρίσκονται γύρω από τον κόμβο με την ελάχιστη τιμή, θα γίνει η αναπροσαρμογή των συντελεστών βάρους. Γενικά, οι παράμετροι αυτές δεν παραμένουν σταθερές κατά τη διαδικασία εκπαίδευσης. Όπως έχει δείχθει μετά από πολλά πειράματα, η αποδοτικότητα του αλγορίθμου βελτιώνεται αν αυτές οι παράμετροι ελαττώνονται με αργό ρυθμό στο χρόνο. Μια μέθοδος για τον καθορισμό των παραμέτρων αυτών μπορεί να επιτευχθεί, διαιρώντας το πρόβλημα σε δύο μέρη. Υποθέτοντας ότι όλη η διαδικασία περιλαμβάνει  $t_2$  βήματα, πρέπει να καθορίσουμε τον αριθμό των επαναλήψεων  $t_1$  στο πρώτο μέρος. Διαλέγοντας την αρχική τιμή της ακτίνας γειτνίασης, η ελάττωσή της μπορεί να γίνει γραμμικά, έτσι ώστε μετά από  $t_1$  βήματα να φτάσει στο 0.

Στις επόμενες  $t_2 - t_1$  επαναλήψεις, η αναπροσαρμογή των συντελεστών βάρους πραγματοποιείται μόνο στους συντελεστές βάρους των συνάψεων που ενώνουν τις εισόδους με τον  $j^*$  κόμβο. Ένας πρακτικός τρόπος για την αναπροσαρμογή του συντελεστή μάθησης  $\varepsilon(t)$  είναι:  $\varepsilon(t) = k_1(1 - \frac{t}{t_1})$ , για  $0 \leq t \leq t_1$ , και

$\varepsilon(t) = k_2(1 - \frac{t}{t_2})$  , για  $t_1 \leq t \leq t_2$  , όπου  $k_1$  και  $k_2$  είναι δύο σταθερές. Γενικά, η απόδοση του αλγορίθμου αυτού εξαρτάται από έναν αριθμό παραμέτρων και υποπαραμέτρων, των οποίων οι τιμές καθορίζονται μετά από έναν μεγάλο αριθμό προσομοιώσεων. Σημαντική βελτίωση αποτελεί η χρήση της sigmoid οριακής συνάρτησης, αντί των hardlimiter και threshold, σύμφωνα με εργασίες του Peretto.

Μαθηματικά, ο αλγόριθμος αναπαριστά μια διαδικασία Markov.

Οι μεταβάσεις του συστήματος συνεχίζονται μέχρι οι παράμετροι  $N_j(t)$  και  $\varepsilon(t)$  να πάρουν τέτοιες τιμές (μετά από πεπερασμένο αριθμό επαναλήψεων), ώστε το δίκτυο να βρεθεί σε σταθερή κατάσταση.

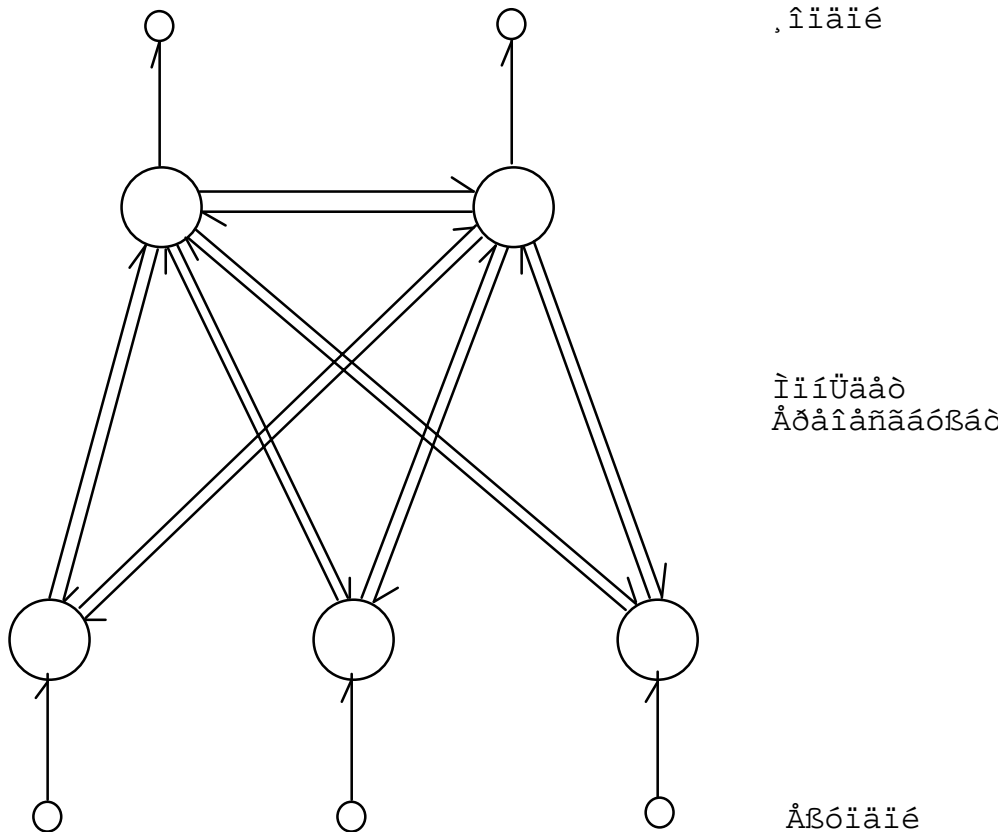
Το κύριο πλεονέκτημα του Kohonen TND είναι η υψηλή ταχύτητα επεξεργασίας, και για το λόγο αυτό βρίσκει εφαρμογή σε real-time συστήματα. Επίσης, η εκπαίδευση είναι συνεχής, δηλαδή το δίκτυο αναπροσαρμόζεται σε μεταβολές, συναρτήσεως του χρόνου. Τέλος, επειδή η σύνδεση των συντελεστών βάρους σε ένα Kohonen δίκτυο μεταβάλεται συναρτήσεως του χρόνου, ώστε να προσεγγίζουν την πυκνότητα των προτύπων εισόδου, το δίκτυο αυτό θα μπορούσε να σχεδιάζει τις εισόδους του σε ένα δισδιάστατο επίπεδο (grid), δημιουργώντας έτσι μια γραφική απεικόνιση της κατανομής των εισόδων αυτών.

Γενικά, τα Kohonen μοντέλα βρίσκουν εφαρμογή κυρίως σε στατιστικά και τοπολογικά προβλήματα (statistical & topological modelling), π.χ. είσοδοι θα μπορούσαν να είναι συχνότητες ή άλλα στατιστικά δεδομένα.

- ART1

Αναπτύχθηκε από τους Gai Carpenter και Stephen Grossberg και πραγματοποιεί ταξινόμηση σε δυαδικά δεδομένα. Η λέξη ART προέρχεται από τα αρχικά των λέξεων Adaptive Resonance Theory. Οι Carpenter και Grossberg έχουν ερευνήσει τις ιδιότητες ορισμένων δικτύων, των οποίων η συμπεριφορά περιγράφεται με διαφορικές εξισώσεις. Το ART1 αποτελείται από την ένωση πολλών τέτοιων δικτύων, προκειμένου να δημιουργηθεί ένα πολύπλοκο σύστημα με περισσότερο ικανές ιδιότητες. Αν και τα δίκτυα αυτά υλοποιούνται χρησιμοποιώντας μονάδες συνεχούς χρόνου, οι οποίες περιγράφονται με διαφορικές εξισώσεις, η θεωρία του ART1 εμπεριέχει ορισμένες απλοποιημένες υποθέσεις, οι οποίες επιτρέπουν η συμπεριφορά του ART1 T.N.Δ. να περιγραφεί σαν

ένας αλγόριθμος ταξινόμησης διακριτού χρόνου. Το ART1 αποτελείται από ένα στρώμα εισόδου και ένα στρώμα εξόδου. Κάθε κόμβος εισόδου συνδέεται με όλους τους κόμβους εξόδου, και οι συνδεσμοί αυτοί ονομάζονται bottom-up. Επιπλέον, κάθε κόμβος εξόδου συνδέεται με όλους τους κόμβους εισόδου και οι σύνδεσμοι αυτοί ονομάζονται top-down.



**Σχήμα 13** - Δομή ART1 μοντέλου ΤΝΔ

Σε μια εφαρμογή ταξινόμησης δεδομένων, οι κόμβοι εισόδου παριστάνουν τα δεδομένα εισόδου, και οι κόμβοι εξόδου παριστάνουν τις κλάσεις. Τα αντιπροσωπευτικά παραδείγματα κάθε κλάσης αποθηκεύονται στους top-down συνδέσμους. Η εκπαίδευση του αλγορίθμου γίνεται χωρίς επίβλεψη. Ο αλγόριθμος εκπαίδευσης είναι ο εξής:

**Βήμα 1:**

Δίνονται αρχικές τιμές στους bottom-up  $w_{ij}$  και top-down  $t_{ij}$  συντελεστές βάρους.

$$w_{ij}(0) = \frac{1}{1+N} , t_{ij}(0) = 1$$

όπου  $0 \leq i \leq N-1$  και  $0 \leq j \leq M-1$  . Ο αριθμός των κόμβων εισόδου είναι  $N$  και των κόμβων εξόδου είναι  $M$  . Επίσης καθορίζεται η τιμή του vigilance threshold μεταξύ 0 και 1 .

Βήμα 2:

Εισάγεται τυχαία κάποιο πρότυπο  $(x_0, x_1, \dots, x_{N-1})$ .

Βήμα 3:

Υπολογίζονται τα αποτελέσματα εξόδου

$$o_j = \sum_{i=0}^{N-1} w_{ij}(t)x_i$$

όπου  $0 \leq j \leq M-1$ .

Βήμα 4:

Επιλέγεται ο κόμβος με τη μεγαλύτερη τιμή εξόδου  $o_j^* = \max_j o_j$ .

Βήμα 5:

Συγκρίνεται το πρότυπο εκπαίδευσης με το αντιπροσωπευτικό παράδειγμα του κόμβου εξόδου που επιλέχτηκε στο προηγούμενο

βήμα. Συγκεκριμένα, υπολογίζεται η ποσότητα:  $\frac{\|T.X\|}{\|X\|}$ , όπου

$$\|T.X\| = \sum_{i=0}^{N-1} t_{ij}x_i$$

$$\|X\| = \sum_{i=0}^{N-1} x_i$$

Στη συνέχεια, η ποσότητα αυτή συγκρίνεται με το vigilance threshold.

Αν  $\frac{\|T.X\|}{\|X\|} > p$ , δηλαδή το πρότυπο εκπαίδευσης είναι "πολύ όμοιο" με το παράδειγμα αυτό, τότε ταξινομείται στην ίδια κλάση και οι bottom-up και top-down συντελεστές βάρους από και προς τον κόμβο αυτό αναπροσαρμόζονται:

$$t_{ij}(t+1) = t_{ij}(t)x_i$$

$$w_{ij}(t+1) = \frac{t_{ij}(t)x_i}{0.5 + \sum_{i=0}^{N-1} t_{ij}(t)x_i}$$

Στην αντίθετη περίπτωση, αν  $\frac{\|T.X\|}{\|X\|} < p$ , απενεργοποιούνται οι top-down συνδεσμοί που ξεκινούν από τον κόμβο που επιλέχτηκε στο βήμα 4 και συνεχίζουμε πηγαίνοντας στο βήμα 3. Αν έχουν απενεργοποιηθεί όλοι οι κόμβοι εξόδου, που αντιστοιχούν στις κλάσεις που έχουν δημιουργηθεί μέχρι αυτή τη στιγμή, τότε το δεδομένο εισόδου ανήκει σε μια καινούρια κλάση και ένας καινούριος κόμβος εξόδου επιλέγεται για να αντιπροσωπεύσει την

κλάση αυτή. Στη συνέχεια, οι συντελεστές βάρους των top-down και bottom-up συνδέσμων, οι οποίοι ξεκινούν και καταλήγουν σε αυτό τον κόμβο, αναπροσαρμόζονται κατά τον ίδιο τρόπο σύμφωνα με τις παραπάνω εξισώσεις.

Βήμα 6:

Ενεργοποιούνται ξανά οι κόμβοι που απενεργοποιήθηκαν στο βήμα 5.

Επανάληψη, πηγαίνοντας στο βήμα 2, μέχρι ταξινόμησης όλων των προτύπων.

Για την αρχικοποίηση των  $NM$  συντελεστών βάρους στο βήμα 1, έχουμε πολυπλοκότητα  $O(NM)$ . Για το βήμα 2, έχουμε  $O(1)$ , ενώ στο βήμα 3 απαιτείται υπολογισμός  $M$  εξόδων η καθεμία από τις οποίες είναι άθροισμα  $N$  όρων, συνεπώς η πολυπλοκότητα είναι  $O(NM)$ . Στο βήμα 4, επειδή έχουμε τυχαία κατανομή των τιμών εξόδου, απαιτείται αναγκαστικά γραμμική αναζήτηση πολυπλοκότητας  $O(M)$ . Στο βήμα 5, ο υπολογισμός των  $\|X\|$  και  $\|T.X\|$  γίνεται με άθροισμα  $N$  όρων για το καθένα, άρα  $O(N)$ . Στην

περίπτωση που  $\frac{\|T.X\|}{\|X\|} > P$ , τότε ο υπολογισμός των top-down ( $t_{ij}$ ) και bottom-up ( $w_{ij}$ ) είναι πολυπλοκότητας  $O(NM)$  και

$O(MN^2)$  αντίστοιχα. Αντίθετα, στην περίπτωση που  $\frac{\|T.X\|}{\|X\|} < P$ , έχουμε προηγουμένως την απενεργοποίηση των top-down συνδέσμων του επιλεγόμενου κόμβου, πολυπλοκότητας  $O(NM)$ , επιλογή νέου κόμβου εξόδου αν έχουμε νέα κλάση, πολυπλοκότητας  $O(1)$ , και στη συνέχεια αναπροσαρμογή των  $t_{ij}$  και  $w_{ij}$  όπως παραπάνω. Συνολικά, η πολυπλοκότητα για το βήμα 5 είναι  $O(MN^2)$ . Στο βήμα 6, για την ενεργοποίηση των απενεργοποιημένων (από το βήμα 5) κόμβων έχουμε πολυπλοκότητα  $O(NM)$ . Τέλος, ο αλγόριθμος επαναλαμβάνεται για όλα τα L σε πλήθος πρότυπα εισόδου.

Από τα παραπάνω συμπεραίνουμε ότι η πολυπλοκότητα του αλγορίθμου είναι  $O(LMN^2)$ .

Το ART1 T.N.Δ. λειτουργεί παρόμοια με τον κλασικό Simple Cluster Seeking αλγόριθμο ταξινόμησης. Ο αριθμός των κλάσεων δεν καθορίζεται εξ αρχής, αλλά εξαρτάται από την παράμετρο  $p$  που ονομάζεται vigilance threshold. Το vigilance threshold καθορίζει τη

μέγιστη διαφορά μεταξύ δύο προτύπων, τα οποία ανήκουν στην ίδια κλάση. Μεγαλύτερο  $\rho$  αντιστοιχεί σε μικρότερη απόκλιση των στοιχείων μιας κλάσης, το οποίο σημαίνει ότι ένας μεγαλύτερος αριθμός κλάσεων απαιτείται για την ταξινόμηση όλων των δεδομένων. Ουσιαστικά, το vigilance threshold αντιστοιχεί στην "επαγρύπνηση" του δικτύου να αυτοπροσαρμόζεται και να αναδιοργανώνει τα δεδομένα που είναι αποθηκευμένα στο δίκτυο σε νέες κλάσεις. Το κυριότερο χαρακτηριστικό του ART1 ΤΝΔ είναι ότι μαθαίνει παρουσιάζοντας τα πρότυπα εκπαίδευσης στο δίκτυο μόνο μια φορά. Αξίζει όμως να σημειωθεί ότι οι κλάσεις που σχηματίζονται εξαρτώνται από τη σειρά με την οποία τα πρότυπα εισόδου παρουσιάζονται στο δίκτυο.

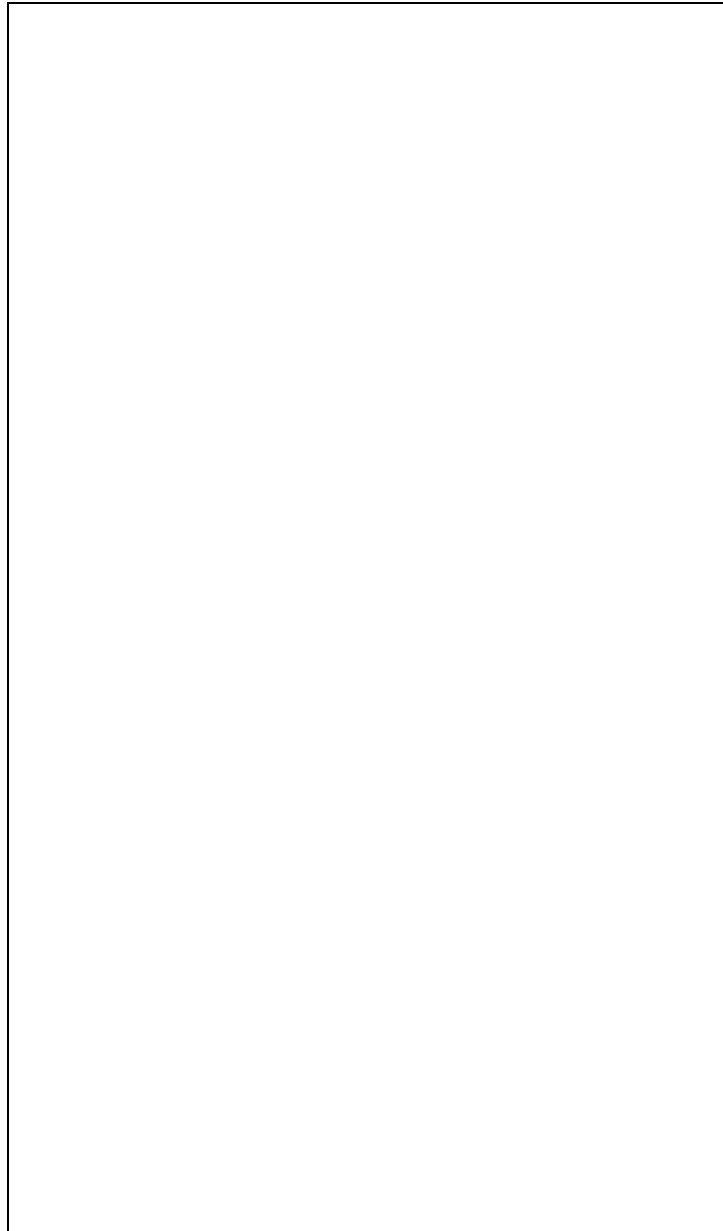
- ALN - Adaptive Logic Networks

Τα ALNs (Adaptive Logic Networks) είναι ένας τύπος νευρωνικών δικτύων που αναπτύχθηκε στο Πανεπιστήμιο της Alberta του Καναδά, συνεχίζοντας προηγούμενη δουλειά των Bell Telephone Laboratories και του Πανεπιστημίου του Montreal. Αυτό το μοντέλο Τ.Ν.Δ. είναι μια ειδική περίπτωση του γνωστού MLP (Multi-Layered Perceptron) μοντέλου. Τα ALNs έχουν σχεδιαστεί για την εκτέλεση βασικών υπολογισμών ταξινόμησης προτύπων με πολύ υψηλή ταχύτητα, μεγαλύτερη από τα περισσότερα ψηφιακά συστήματα. Προσφέρουν ταχύτητα πολλές τάξεις μεγέθους μεγαλύτερη από κάθε άλλη γνωστή τεχνολογία, χωρίς να χάνουν σε ευελιξία και αντοχή στο θόρυβο.

Τα δεδομένα εισόδου στο σύστημα εκπαίδευσης πρέπει να είναι λογικές (boolean) μεταβλητές. Μη λογικά δεδομένα κωδικοποιούνται σε λογικά διανύσματα, πριν μπουν στη φάση εκπαίδευσης. Πολλές λογικές έξοδοι μπορεί να χρησιμοποιηθούν για να συνθέτουν εξόδους που είναι πιο πολύπλοκες από τις λογικές, όπως είναι οι συνεχείς τιμές.

Οι διασυνδέσεις των κόμβων έχουν συνήθως τη μορφή δυαδικών δένδρων. Υπάρχουν επίσης πολλές είσοδοι στους κόμβους - φύλλα του δένδρου. Στην πράξη δεν υπάρχει όριο στον αριθμό των ενδιάμεσων (κρυφών) επιπέδων. Τυπικό μέγεθος είναι 10 με 15 επίπεδα.





**Σχήμα 14** - Λειτουργία ALN μοντέλων ΤΝΔ

Οι κόμβοι έχουν δύο εισόδους. Τα σήματα εισόδου  $x_1$ ,  $x_2$  είναι λογικές μεταβλητές (0 ή 1). Κάθε βάρος στις εισόδους καθορίζεται από την τιμή ενός bit. Η μη γραμμικότητα (*squashing function*) επιτυγχάνεται με σύγκριση με μια σταθερά κατωφλίου. Ειδικότερα, αν  $b_1$  και  $b_2$  είναι λογικές μεταβλητές, τότε το δίκτυο υπολογίζει μια λογική τιμή, η οποία είναι 1, αν και μόνο αν

$$(b_1+1)*x_1+(b_2+1)*x_2 \geq 2$$

Οι τέσσερις δυνατοί συνδυασμοί των  $b_1$  και  $b_2$  (00, 11, 10, 01) δημιουργούν 4 λογικές συναρτήσεις δύο μεταβλητών: AND, OR, LEFT και RIGHT, όπου  $LEFT(x_1, x_2)=x_1$  και  $RIGHT(x_1, x_2)=x_2$ . Οι συναρτήσεις LEFT και RIGHT χρησιμοποιούνται για να παρέχουν ευελιξία στις διασυνδέσεις και καταργούνται μετά την εκπαίδευση.

Ένα δέντρο που αποτελείται από αυτά τα στοιχεία είναι τυχαία συνδεδεμένο με μερικές λογικές μεταβλητές εισόδου και τα συμπληρώματά τους. Οι εισοδοί τροφοδοτούνται στα φύλλα του δέντρου και η έξοδος είναι στη ρίζα του. Οι μεταβλητές εισόδου είναι μέρη ενός διανύσματος που παριστάνει ένα πρότυπο (pattern), όπως είναι, για παράδειγμα, ένας χειρόγραφος χαρακτήρας. Η εκπαίδευση για ένα σύνολο διανυσμάτων εισόδου και ο υπολογισμός των επιθυμητών λογικών εξόδων περιλαμβάνει την τροφοδότηση των διανυσμάτων εισόδου με τυχαία σειρά και των επιθυμητών αποτελεσμάτων. Έτσι, κάθε κόμβος αποκτά μια συνάρτηση, που επιτρέπει στο δέντρο να προσεγγίζει τις επιθυμητές εξόδους, όπως έχουν οριστεί από τα δεδομένα εκπαίδευσης.

Οι μεταβολές που γίνονται κατά την εκπαίδευση στηρίζονται στη βασική ιδέα ότι, αν μια είσοδος σε μια πύλη AND (OR) είναι 0 (1), τότε οι κόμβοι του υποδένδρου που συνδέονται με την άλλη είσοδο είναι αδιάφοροι. Σύμφωνα με αυτόν τον απλό κανόνα, ένας κόμβος είναι υπεύθυνος, αν μια αλλαγή στην έξοδό του μπορεί να αλλάξει την έξοδο του δικτύου. Αυτό μοιάζει με τον αλγόριθμο Back Propagation, με τη διαφορά ότι η ποσότητα που μεταφέρεται προς τα πίσω είναι 0 ή 1.

Μετά την εκπαίδευση το ALN έχει ενσωματωμένη δυνατότητα γενίκευσης. Το σύνολο που χρησιμοποιείται για την εκπαίδευση μπορεί να περιέχει αντιφατικά δεδομένα. Τα δείγματα αυτά λαμβάνονται υπόψη κατά την εκπαίδευση, για τη βελτίωση του τελικού αποτελέσματος. Η επιτυχημένη γενίκευση στηρίζεται στο γεγονός ότι δένδρα με εισόδους δύο από τις λογικές πύλες AND, OR, LEFT και RIGHT είναι εξαιρετικά ανθεκτικά στις μεταβολές των εισόδων τους.

Η εκπαίδευση των ALNs καταλήγει σε ένα δένδρο από τελεστές (λογικές πύλες) AND και OR. Το δένδρο δέχεται σαν είσοδο ένα πρότυπο που αποτελείται από 0 και 1. Κάποια από αυτά τα σήματα αντιστρέφονται, πριν περάσουν στο δένδρο, αφού η αντιστροφή είναι απαραίτητη για τη δημιουργία της επιθυμητής σχέσης ανάμεσα στις εισόδους και στις εξόδους του δικτύου. Μια πύλη στο δένδρο δέχεται σαν είσοδο σήματα 0 ή 1. Μια πύλη OR παράγει έξοδο 1, αν οποιαδήποτε από τις εισόδους είναι 1, και μια πύλη AND παράγει έξοδο 1, αν όλες οι εισοδοί είναι 1. Όπως είναι φανερό, οι πύλες OR

επιτρέπουν στο δίκτυο να αντιμετωπίσει εναλλακτικές γεωμετρικές μορφές για μια δεδομένη είσοδο.

Ένα άλλο χαρακτηριστικό των ALNs είναι η εξαιρετική ανθεκτικότητά τους στην επίδραση θορύβου στα δεδομένα εισόδου, και μάλιστα χωρίς σημαντική επιβάρυνση στον χρόνο επεξεργασίας.

Αποδεικνύεται ότι στα ALNs η πολυπλοκότητα του χρόνου επεξεργασίας είναι λογαριθμική σε σχέση με το μέγεθος του προβλήματος. Έτσι, ο υπολογιστικός χρόνος αυξάνει με πολύ μικρότερο ρυθμό σε σχέση με το μέγεθος του προβλήματος ως προς τα άλλα μοντέλα δικτύων.

Τα ιδιαίτερα χαρακτηριστικά των ALNs τα καθιστούν χρήσιμα για εφαρμογές όπως ο αυτοματισμός εργοστασίων, η αναγνώριση φωνής και εικόνας, οπτική αναγνώριση χαρακτήρων, η ανάλυση δεδομένων, διάκριση μορίων, συστήματα ελέγχου στη ιατρική, μελέτη σύνθεσης εδάφους, κ.τ.λ.

- BSB - Brain-State-in-a-Box

Το όνομά του προέρχεται από τα αρχικά των λέξεων Brain-State-in-a-Box, και οφείλεται στη σταθεροποίηση του δικτύου μέσα σε ένα "κουτί" ορίων ή αλλιώς "υπερκύβων" (hypercubes). Είναι αποτέλεσμα μελέτης του James Anderson.

Όπως και στο Adaline, έτσι και το BSB μοντέλο χρησιμοποιεί ανάδραση (feedback) με διόρθωση λάθους. Το δίκτυο χαρακτηρίζεται ως auto-associative και δεν είναι πλήρως συνδεδεμένο. Κάθε πρότυπο εισόδου δίνεται με τη μορφή διανύσματος που αναπαριστά τα μερικώς ορισμένα δεδομένα. Η έξοδος περιορίζεται από τη συνάρτηση εξόδου που μπορεί να είναι η threshold οριακή συνάρτηση. Αυτό εξασφαλίζει τη σύγκλιση σε μια γωνία του "κουτιού" και ελαχιστοποιεί το σφάλμα σε σχέση με το σύνολο προτύπων εκπαίδευσης. Το BSB μοντέλο είναι ανθεκτικό σε λάθη και σε θόρυβο, αλλά απαιτεί χρόνο και επίβλεψη για την κωδικοποίηση. Γενικά το μοντέλο αυτό επεκτείνει την κατανόηση από εμάς του εγκεφάλου και του τρόπου λειτουργίας του.

Οι εφαρμογές του είναι περισσότερο πειράματα, παρά προϊόντα. Το μοντέλο μπορεί να εφαρμοστεί στην ανάκτηση γνώσης από βάσεις δεδομένων ("bird" classification database), συστήματα ιατρικής διάγνωσης, κ.ά.

- Boltzmann

Ονομάζεται αλλιώς και Boltzmann Machine. Είναι αποτέλεσμα μελέτης των Hinton και Sejnowski και βασίζεται στο στοχαστικό Hopfield T.N.Δ. μοντέλο, με την προσθήκη "κρυμμένων" μονάδων. Τέτοια δίκτυα σχηματίζουν γενικευμένες υπολογιστικές μηχανές βασισμένες σε στοχαστικούς υπολογιστικούς κανόνες και ονομάζονται Boltzmann Machines, σε αντίθεση με τους ηλεκτρονικούς υπολογιστές που βασίζονται σε Van Neuman αρχιτεκτονικές.

Γενικά, ο back-propagation αλγόριθμος δε μπορεί να εφαρμοστεί άμεσα σ' αυτό το μοντέλο, γιατί οι σύνδεσμοι είναι συμμετρικοί και όχι προσανατολισμένοι προς τα εμπρός. Οι διακριτές μονάδες σ' ένα τέτοιο δίκτυο μπορούν να πάρουν μια από τις δύο διακριτές τιμές  $s_i = \pm 1$ , υποθέτοντας τη θετική τιμή με πιθανότητα  $f(h_i)$ , όπου

$$h_i = \sum_k w_{ik} s_k$$

και  $f(h)$  Fermi (sigmoid) συνάρτηση.

Η δυναμική ενός τέτοιου δικτύου μπορεί να περιγραφεί από τη συνάρτηση "ενέργειας":

$$E(s) = -\frac{1}{2} \sum_{ik} w_{ik} s_i s_k$$

Το ελάχιστο αυτής της συνάρτησης αντιστοιχεί σε σταθερές καταστάσεις του δικτύου. Πολλά χαρακτηριστικά του δικτύου μπορούν επίσης να μελετηθούν με μεθόδους στατιστικής φυσικής. Ο αλγόριθμος εκπαίδευσης βασίζεται στην έννοια της εντροπίας.

Αρχικά, διαχωρίζουμε τις μονάδες σε "κρυφές" (εσωτερικές) και σε "περιφερειακές" (εξωτερικές). Διαχωρίζουμε τις καταστάσεις των μονάδων εξόδου με ένα δείκτη  $\alpha$ , των μονάδων εισόδου με ένα δείκτη  $\beta$  και των κρυμμένων μονάδων με ένα δείκτη  $\gamma$ . Η κατάσταση ολόκληρου του δικτύου ορίζεται τότε από το συνδυασμό των δεικτών, δηλαδή  $\alpha\beta\gamma$ .

Για ένα συγκεκριμένο σετ προτύπων εκπαίδευσης, η κατανομή της πιθανότητας να έχουμε μια συγκεκριμένη διαμόρφωση εισόδου  $\beta$ , μπορεί να περιγραφεί ως  $Q_\beta$ . Η κατανομή αυτή παρέχεται εξωτερικά ανεξάρτητα από το δυναμικό του δικτύου. Αν η υπό συνθήκη πιθανότητα να έχουμε μια συγκεκριμένη διαμόρφωση εξόδου  $\alpha$  για τη δεδομένη διαμόρφωση εισόδου  $\beta$ , δίνεται ως

$P_{\alpha|\beta}$  και η πραγματική κατανομή των καταστάσεων εξόδου δίνεται ως  $P_{\alpha} = \sum_{\beta} P_{\alpha|\beta} Q_{\beta}$  .

Παρόμοιες εξισώσεις μπορούν να εφαρμοστούν αν συμπεριληφθούν και οι κρυμμένες μονάδες. Οι δείκτες δεξιά της καθέτου στον ορισμό των πιθανοτήτων αντιστοιχεί πάντα στη σταθερή διαμόρφωση μιας υπό συνθήκη πιθανότητας. Επίσης, όλες οι κατανομές πιθανοτήτων θεωρούνται κανονικοποιημένες ως προς τη μονάδα, δηλαδή

$$\sum_{\alpha} P_{\alpha|\beta} = \sum_{\alpha\gamma} P_{\alpha\gamma|\beta} = 1$$

Η υπό συνθήκη πιθανότητα  $P_{\alpha|\beta}$  καθορίζεται από τους συντελεστές βάρους  $w_{ik}$  του δικτύου. Γενικά, η πιθανότητα

$$P[s] = Z^{-1} e^{-\beta E[s]}$$

$$Z = \sum_{[s]} e^{-\beta E[s]}$$

Έτσι έχουμε

$$P_{\alpha|\beta} = \sum_{\gamma} P_{\alpha\gamma|\beta} = Z_{|\beta}^{-1} \sum_{\gamma} \exp\left(-\frac{E_{\alpha\gamma\beta}}{T}\right)$$

όπου  $E_{\alpha\gamma\beta}$  είναι η συνάρτηση ενέργειας του δικτύου και  $Z_{|\beta}$  είναι η συνάρτηση διαμέρισης για τη δεδομένη διαμόρφωση εισόδου  $\beta$  :  $Z_{|\beta} = \sum_{\alpha\gamma} \exp\left(-\frac{E_{\alpha\gamma\beta}}{T}\right)$  . Η διαδικασία εκπαίδευσης στοχεύει στην

κατάλληλη επιλογή συνδέσμων κατά τέτοιο τρόπο, ώστε η υπό συνθήκη πιθανότητα  $P_{\alpha|\beta}$  να πάρει την επιθυμητή τιμή, την οποία εμείς ορίζουμε με την  $Q_{\alpha|\beta}$  . Αν η μοναδική αντίδραση  $\alpha_0(\beta)$  πρέπει να συσχετιστεί με κάθε είσοδο  $\beta$  , η κατανομή αυτή δίνεται ως  $Q_{\alpha|\beta} = \delta_{\alpha\alpha_0}$  .

Για την επίλυση του προβλήματος απαιτείται ο καθορισμός μιας κατάλληλης συνάρτησης  $D$  που περιγράφει την απόκλιση της επιθυμητής κατανομής απόκρισης  $Q_{\alpha|\beta}$  και της πραγματικής κατανομής  $P_{\alpha|\beta}$  για τις δεδομένες συνδέσεις του δικτύου.

Με βάση τα παραπάνω και γνωρίζοντας την υπο συνθήκη πιθανότητα  $P_{\alpha}$  , καθορίζεται ο τρόπος μεταβολής των συντελεστών

βάρους  $w_{ik}$ , γνωστός και ως ο κανόνας εκπαίδευσης Boltzmann (Boltzmann learning rule):

$$\begin{aligned}\delta_{w_{ik}} &= \frac{\varepsilon}{T} \sum_{\alpha\beta} Q_{\alpha|\beta} Q_{\beta} \left( \sum_{\gamma} \frac{P_{\alpha\gamma\beta}}{P_{\alpha\beta}} [s_i s_k]_{\alpha\gamma\beta} - \langle s_i s_k \rangle_{|\beta} \right) \\ &= \frac{\varepsilon}{T} \sum_{\beta} Q_{\beta} \left( \sum_{\alpha} Q_{\alpha|\beta} \langle s_i s_k \rangle_{\alpha\beta} - \langle s_i s_k \rangle_{|\beta} \right)\end{aligned}$$

όπου

$$\begin{aligned}\langle s_i s_k \rangle_{|\beta} &= \sum_{\alpha\gamma} P_{\alpha\gamma|\beta} [s_i s_k]_{\alpha\gamma\beta} \\ &= Z_{|\beta}^{-1} \sum_{\alpha\gamma} \exp\left(-\frac{E_{\alpha\gamma\beta}}{T}\right) [s_i s_k]_{\alpha\gamma\beta}\end{aligned}$$

και  $[s_i s_k]_{\alpha\gamma\beta}$  δηλώνει τις καταστάσεις των δύο μονάδων που εξετάζουμε, δεδομένου ότι η καθολική κατάσταση του δικτύου δίνεται ως  $\alpha\gamma\beta$ . Το σύμβολο  $|\beta$  υποδεικνύει ότι ο μέσος όρος υπολογίζεται παρουσία της διαμόρφωσης εισόδου  $\beta$ .

Με αυτόν τον τρόπο, η διαμόρφωση των συνδέσμων μπορεί να γίνει με βάση τοπικές παρατηρήσεις, γεγονός που απλοποιεί σημαντικά την αρχιτεκτονική του δικτύου, ειδικά όταν το δίκτυο είναι μεγάλο.

Η διαδικασία εκπαίδευσης μιας μηχανής Boltzmann αποτελείται από μια σειρά εναλλασσόμενα βήματα. Για κάθε πρότυπο εισόδου στο πρώτο βήμα οι μονάδες εισόδου και εξόδου συνδυάζονται και η απόκριση του δικτύου μετρείται. Στη συνέχεια, η ίδια διαδικασία επαναλαμβάνεται, καθώς οι μονάδες εξόδου αφήνονται να "εξελιχθούν" ελεύθερα. Σύμφωνα με τον κανόνα των ανταγωνιστικών Hebb συνάψεων (contrastive Hebbian synapses), οι συντελεστές βάρους  $w_{ik}$  αναπροσαρμόζονται χρησιμοποιώντας την διαφορά και των δύο μετρήσεων. Η διαδικασία αυτή πρέπει να επαναληφθεί για όλα τα πρότυπα εισόδου  $\beta$  και συνήθως πολλοί τέτοιοι κύκλοι εκπαίδευσης είναι απαραίτητοι. Αυτό μπορεί να είναι υπολογιστικά δαπανηρό, αφού οι συσχετισμοί των στοχαστικά μεταβαλλόμενων ποσοτήτων  $\langle s_i s_k \rangle$  πρέπει να μετρηθούν.

Μία πιθανώς γρηγορότερη ντετερμινιστική μηχανή Boltzmann θα μπορούσε να κατασκευαστεί βασιζόμενη στην μέθοδο Mean Field Approximation. Με αυτό τον τρόπο οι στοχαστικές μονάδες  $s_i$  αντικαθίστανται με αναλογικές ντετερμινιστικές μονάδες  $\bar{s}_i$ , με

ενεργοποίηση στην περιοχή:  $-1 \leq \bar{s}_i \leq +1$  . Αυτές καταλήγουν σε ένα συγκεκριμένο σημείο σταθεροποίησης του συστήματος των εξισώσεων  $\bar{s}_i = \tanh\left(\frac{1}{T} \sum_j w_{ij} \bar{s}_j\right)$  , το οποίο ελαχιστοποιεί την

"ελεύθερη" ενέργεια:  $F = E - TS$  . Με αυτή την προσέγγιση οι συσχετισμοί γίνονται απλά γινόμενα:  $\langle s_i s_k \rangle \rightarrow \bar{s}_i \bar{s}_k$  .

Ο Sejnowski έχει μελετήσει την επίδοση των μηχανών Boltzmann, δηλαδή των συμμετρικά συνδεδεμένων δικτύων με κρυφά στρώματα.

Γενικά, λόγω της στοχαστικής λειτουργίας του, το δίκτυο δεν συγκλίνει αναγκαστικά στην ίδια λύση σε κάθε δοκιμή, αν υπάρχουν περισσότερες από μία δεκτές λύσεις του προβλήματος. Χαρακτηριστικό παράδειγμα είναι το πρόβλημα αναγνώρισης συμμετριών.

- KRN - Knowledge Representation Network

Ένα ακόμα ΤΝΔ μοντέλο είναι το KRN (Knowledge Representation Network), αποτέλεσμα εργασιών του Claude Cruz. Το δίκτυο αυτό, όπως δηλώνει άλλωστε και το όνομά του, είναι ένα μοντέλο αναπαράστασης γνώσης και ένα "κέλυφος" επεξεργασίας. Έτσι, δε συνδέεται άμεσα με κανένα συγκεκριμένο πρόβλημα επεξεργασίας γνώσης. Το KRN είναι μια προσπάθεια να εμφανιστεί "έξυπνη" συμπεριφορά από ένα συνήθως μεγάλο αριθμό από ομοιόμορφα και απλά στοιχεία. Ενώ έχει τη βάση του στα νευρωνικά δίκτυα, δεν έχει στενή σχέση με καμία οργάνωση ΤΝΔ μοντέλων, ούτε και με κάποιο στοιχείο τους.

Το KRN μοντέλο δε βασίζεται σε τόσο αυστηρούς κανόνες οργάνωσης και λειτουργίας, όσο τα άλλα μοντέλα. Σχεδιάστηκε έχοντας υπόψη τα εξής:

- Η σημασία των KRN γνωστικών οντοτήτων είναι να αναπαρασταθούν σε γνωστικές δομές περισσότερο, παρά σε εξωτερική (ανθρώπινη) ερμηνεία των δομών αυτών.

Η συμπεριφορά του συστήματος πρέπει να είναι μεταβλητή από το ίδιο το σύστημα με βάση την "εμπειρία" του. Παρ' όλα αυτά, αυτό δεν αποκλείει τη χρήση προκαθορισμένης γνώσης επίσης.

- Μια ενδιαμέση προσέγγιση είναι να χρησιμοποιηθεί για την υλοποίηση πλήθους συνεργαζόμενων μηχανισμών ελέγχου.

- Για λόγους αποδοτικότητας, θα πρέπει να χρησιμοποιηθεί παράλληλη επεξεργασία (όπου αυτό είναι δυνατό) στην αναπαράσταση και τη διαχείριση της γνώσης.

Η αρχιτεκτονική του KRN ΤΝΔ περιλαμβάνει αρκετούς κανόνες:

- Η αναγνώριση γεγονότων και οντοτήτων γίνεται με βάση μια ιεραρχία, που παίρνει δεδομένα από διάφορες εισόδους ("input channels"). Με τον τρόπο αυτό κωδικοποιείται ο καθορισμός της γνώσης (declarative knowledge) στο δίκτυο.

- Η διεργασία ενεργοποίησης πραγματοποιείται σύμφωνα με μια συγκεκριμένη "ιεραρχία", η οποία συντονίζει τις "χαμηλού επιπέδου" λειτουργίες, σύμφωνα με εντολές του υψηλού επιπέδου.



Με τον τρόπο αυτό κωδικοποιείται η διαδικαστική γνώση (procedural knowledge).

- Διασταύρωση μεταξύ αυτών των ιεραρχιών δημιουργεί μια συμπεριφορά με τη μορφή κανόνων. Η έξυπνη λήψη απόφασης γίνεται μέσα από προσαρμόσιμη σχεδίαση, χρησιμοποιώντας την επίκτητη ή εξαρχής καθορισμένη γνώση.

- Σε κάθε χρονική στιγμή, πολλές ταυτόχρονες λειτουργίες μπορούν να συμβούν στο KRN δίκτυο. Αυτό είναι εφικτό επειδή το ίδιο το δίκτυο αποτελείται από παράλληλο μηχανισμό χαμηλού επιπέδου (νευρωνικό δίκτυο).

- Ταυτόχρονα με την δυνατότητα παράλληλης επεξεργασίας, η προσαρμοστικότητα του KRN μοντέλου διατηρείται από έναν τοπικό μηχανισμό εκπαίδευσης. Αυτός ο τοπικός μηχανισμός μπορεί να προέρχεται από τα υψηλότερα επίπεδα οργάνωσης του δικτύου.

Το KRN μοντέλο είναι ιδιαίτερα ενδιαφέρον γιατί είναι σχεδιασμένο αποκλειστικά για αποτελεσματική λειτουργία σε "πραγματικό" περιβάλλον. Αντίθετα από τα υπάρχοντα συστήματα τεχνητής νοημοσύνης, το KRN μοντέλο βασίζεται στην επεξεργασία σήματος περισσότερο παρά στη συμβολική επεξεργασία. Η εφαρμογή KRN συστημάτων μέσω ΤΝΔ ως χαμηλού επιπέδου μοντέλα υλοποίησης προσφέρουν υψηλής απόδοσης παράλληλη επεξεργασία. Επιπλέον η προσαρμοστικότητα των νευρωνικών δικτύων ικανοποιεί την απαίτηση των KRN δικτύων να μεταβάλλουν τα περιεχόμενά τους, με βάση την αλληλεπίδραση με το περιβάλλον τους.

Η πρακτική εφαρμογή αυτών των συστημάτων στοχεύει στη δημιουργία μεγάλου εύρους "έξυπνων" συστημάτων, ενώ η κατανεμημένη αρχιτεκτονική τους είναι κατάλληλη για εφαρμογές πραγματικού χρόνου.

## Επίλογος

Τα ΤΝΔ αποτελούν μια θεμελιώδη νέα προσέγγιση στην επεξεργασία πληροφοριών και αποτελούν την πρώτη πρακτική εναλλακτική λύση, σε σχέση με τις κλασικές προγραμματιστικές μεθόδους, οι οποίες χρησιμοποιούνται κυρίως σε συστήματα αυτόματου ελέγχου, αναγνώριση χαρακτήρων, ανάλυση και επεξεργασία εικόνων, αναγνώριση και σύνθεση ομιλίας.

Η σημερινή τεχνολογία επιτρέπει την υλοποίηση ΤΝΔ, τόσο σε software, όσο και σε hardware. Ήδη είναι διαθέσιμες "νευρωνικές" μονάδες CPU και ειδικά προσαρμοστικά κυκλώματα, που υλοποιούν σε επίπεδο hardware μοντέλα ΤΝΔ όπως το Hopfield. Σε συνδυασμό με ειδικό software, τα υπολογιστικά αυτά συστήματα επιτυγχάνουν επιδόσεις που αγγίζουν τα 1000 MFLOPS και ενημέρωση 500 εκατομμυρίων συνάψεων ανά δευτερόλεπτο.

Γενικά, πιστεύεται ότι τα ΤΝΔ δε θα αντικαταστήσουν παραδοσιακές μεθόδους υπολογισμού, κυρίως αυτών που σχετίζονται με υψηλής ταχύτητας αριθμητικές επεξεργασίας, αλλά θα συμπληρώσουν και θα αυξήσουν την αποδοτικότητά τους. Ο συνδυασμός του παραδοσιακών υπολογιστών και η μοναδική ισχύς των ΤΝΔ, μπορούν να λύσουν προβλήματα, τα οποία κάτω από άλλες συνθήκες θα παρέμεναν άλυτα.

## Περιεχόμενα:

▪ Γενικά	
..... 1	
▪ Μορφές Μάθησης ΤΝΔ	6
▪ Κανόνες εκπαίδευσης	8
Κανόνας Hebb	8
Κανόνας Δέλτα	8
Gradient Descent	8
Κανόνας Kohonen	9
Back-Propagation	9
Κανόνας Drive-Reinforcement	9
Boltzmann	10
..... 10	
▪ Μοντέλα ΤΝΔ & Αλγόριθμοι	11
Perceptron	11
..... 11	
Adaline-Madaline	15
Multi-Layered Perceptron (MLP)	18
Back-Propagation	20
Βελτιωμένοι Back-Propagation αλγόριθμοι	24
Hopfield	26
Kohonen	29
ART1	32
..... 32	
ALN - Adaptive Logic Network	36

BSB	-	Brain-State-in-a-Box	39
.....			
Boltzmann			
.....			
... 40			
KRN	-	Knowledge Representation Network	43
.....			
■		Επίλογος	
.....			
..... 45			
Περιεχόμενα			
Βιβλιογραφία			

## Βιβλιογραφία:

- {01} "3ο Πανελλήνιο Συνέδριο Πληροφορικής"  
Πρακτικά Συνεδρίου, Αθήνα, Μάιος 1990  
Ελληνική Εταιρεία Επιστήμης Η/Υ & Πληροφορικής
- {02} "Modelling brain function-  
The world of attractor neural networks"  
Daniel J. Amit (Racah Institute of physics)  
Cambridge University Press (89)
- {03} "Neural and Concurrent Real-Time Systems- The Sixth  
Generation"  
Branko Soucek  
(Department of Mathematics - University of Zagreb)  
WILEY (89)
- {04} "Third Italian Workshop : Parallel Architectures and Neural  
Networks".  
(Vietri sul Mare, Salerno, 15-18 May 1990)  
E. R. Caianiello  
(Universita di Salerno)  
World Scientific Publishing (90)
- {05} "The Neurobiology of Neyral Networks"  
Daniel Garder - Douglas A. Baxter - John H. Byrne  
Massachusetts Institute of Technology (93)
- {06} "A Practical Guide to Neural Nets"  
Marilyn McCord Nelson - W. T. Illingworth  
(Texas Instruments)  
Addison - Wesley Publishing Company, Inc. (91)
- {07} "Neural Computation and Self-Organizing Maps:  
An Introduction"  
Helge Ritter - Thomas Martinetz - Klaus Schulten  
(Universities of Bielefield and Illinois - Beckman)  
Addison - Wesley Publishing Company, Inc (92)
- {08} "Neural Networks : An Introduction (Physics of Neural  
Networks".  
B. Muller - J. Reinhardt  
Springer - Verlag (90)
- {09} "Understanding Neural Networks : A Primer /2ed."  
Claude Cruz  
Cutter Information Corporation (91)

- {10} "Cybernetics"  
Norbert Wiener  
Hermann (1958)
- {11} Περιοδικό : "Computer για όλους" - τ.116, Σεπτ. 1993  
Άρθρο : "Νευρωνικά Δίκτυα"  
Τ. Μποσδογιάννη
- {12} Περιοδικό : "Computer & Software" - τ. Αυγ./Σεπτ. 1990  
Άρθρο : "Νευρωνικά Δίκτυα"  
Κ.Πάστρας - Β. Κρικέτος
- {13} Περιοδικό : "Αεροναυτική & Άμυνα" - τ. Μάιος 1990  
Άρθρο : "Νευρωνικοί Υπολογιστές:  
Αρχές και Αμυντικές Εφαρμογές"  
Αλέξανδρος Λούφας

**Software:**

- {I} N. N. GUIDE : Πρακτικός οδηγός για τα N. N.  
(Μαζί με το {06})
- {II} NERVES : Ένα "πραγματικό" παράδειγμα εκπαίδευσης και  
λειτουργίας N. N.  
(downloaded from : {oak.oakland.edu} ftp site)
- {III} LFEDIT & OCR : Εξομοίωση & OCR εφαρμογή που  
χρησιμοποιεί ΤΝΔ.  
(Μαζί με το {11})
- {IV} BRAIN v1.2 : Παράδειγμα εφαρμογής του Back-Propagation  
αλγόριθμου εκπαίδευσης (D. Perkonich).