ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΕΙΡΑΙΑ ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ ΤΜΗΜΑ Η.Υ.Σ

Ανάλυση των SVM (Support Vector Machines) νευρωνικών δικτύων και λογισμικού εφαρμογών

Πτυχιακή Εργασία του Κανελλόπουλου Κλεισθένη

Επιβλέπων Καθηγητής: Βελώνη Αναστασία

ΝΙΚΑΙΑ, 8/28/2013

Περίληψη :

Στην παρούσα πτυχιακή εργασία αναπτύσσονται τα νευρωνικά δίκτυα και συγκεκριμένα οι αλγόριθμοι εκμάθησης Support Vector Machine. Οι Μηχανές Διανυσμάτων Υποστήριξης (SVM) αποτελούν μία σύγχρονη αποτελεσματική προσέγγιση της επίλυσης ζητημάτων κατηγοριοποίησης. Με κατάλληλες διαφοροποιήσεις και επεκτάσεις της βασικής μεθοδολογίας κατηγοριοποίησης σε δύο κλάσεις μπορούν να επιλυθούν προβλήματα περισσότερων κλάσεων, παλινδρόμησης (regression) και αναγνώρισης προτύπων. Η μεθοδολογία αυτή προέκυψε από τη βαθύτερη ανάλυση της στατιστικής θεωρίας μάθησης (statistical learning theory). Αναλύουμε τον τρόπο λειτουργίας τους καθώς και το λόγο που είναι τόσο αξιόπιστα και δημοφιλή στις ακαδημαϊκές και επιστημονικές κοινότητες. Χρησιμοποιήθηκε το πειραματικό πρόγραμμα του Πανεπιστημίου της Ταϊβάν LibSVM διότι αυτό αποτελεί μια από τις λίγες και επικρατέστερες επιλογές απόδοσης τεχνικών νευρωνικών δικτύων με αλγόριθμους SVM, ψηφιακά. Τα παραδείγματα που χρησιμοποιήθηκαν στην εκπόνηση αυτής της εργασίας αποτελούν πραγματικά δεδομένα. Καθότι είναι αρκετά δύσκολο να συγκεντρωθούν τα δεδομένα αυτά εκ νέου, χρησιμοποιήσαμε την ιστοσελίδα του UCI(Machine Learning Repository) λόγω του όγκου των πληροφοριών που περιείχε. Ο όγκος αυτό είναι απαραίτητος για ένα τέτοιο σύστημα κι αυτό γιατί όσα περισσότερα στοιχεία έχουμε για κάθε παράδειγμα τόσο πιο ακριβής είναι η πρόβλεψη.

Πίνακας περιεχομένων

Εισαγωγή:

Τα Support Vector Machine είναι ένα σχετικά νέο επιστημονικό πεδίο στον τομέα της πρόβλεψης-κατηγοριοποίησης με εξαιρετικές δυνατότητες και ευελιξία. Για τον λόγο αυτό αξίζει πραγματικά να τα μελετήσουμε καθώς έχουν εφαρμογή σε πολλά προβλήματα με τεράστιο εύρος θεματολογίας.

Ονομάζουμε τη μεθοδολογία, μεθοδολογία της εκμάθησης (learning methodology) ακριβώς επειδή σκοπός μας είναι να «μάθουμε» το σύστημά μας να σκέφτεται, δηλαδή να ξεχωρίζει, να αναγνωρίζει, να ομαδοποιεί, να κατηγοριοποιεί δεδομένα. Αυτό θα υλοποιηθεί μέσω συγκεκριμένων αλγορίθμων. Είναι πολύ σημαντικό ότι σχεδιάζουμε αλγορίθμους με γενικότητα
που να μπορούν να εφαρμοστούν σε πληθώρα προβλημάτων και έτσι δεν χρειάζεται να αναζητούμε νέους για κάθε νέο πρόβλημα. Εξασφαλίζουμε έτσι ευκαμψία και ελαστικότητα στις διάφορες απαιτήσεις που μπορεί να έχει το σύστημά μας. Επιπλέον ένας γενικότερος αλγόριθμος παρουσιάζει σημαντικά πλεονεκτήματα. Εξασφαλίζει υπολογιστική αποδοτικότητα και στατιστική σταθερότητα. Μπορεί δηλαδή να διαχειρίζεται πολυώνυμα κάθε πιθανού βαθμού ακόμη και όταν συνυπάρχει θόρυβος ή μη ακριβή δεδομένα λόγω ανθρώπινου λάθους. Επίσης οδηγεί πάντα, σχεδόν στα ίδια αποτελέσματα ακόμη και όταν τα δείγματα εισόδου είναι κάθε φορά τυχαία και διαφορετικά, αρκεί να προέρχονται από την ίδια πηγή. Άρα θα πρέπει τα δεδομένα να μετασχηματίζονται κάθε φορά έτσι ώστε να ικανοποιούν τον γενικότερο αλγόριθμο. Στα παραδείγματά μας ο απαιτούμενος μετασχηματισμός επιβάλλει την «μεταφορά» των δεδομένων σε ένα νέο χώρο, από αυτόν που αρχικά δίνονται, όπου θα είναι δυνατός ο γραμμικός διαχωρισμός τους. Η μεταφορά των δεδομένων γίνεται μέσω της συνάρτησης kernel (kernel function) για την οποία θα ακολουθήσει εκτενής αναφορά σε επόμενο κεφάλαιο.

Συνοψίζοντας τα παραπάνω καταλαβαίνουμε ότι η ανάλυση με πρότυπα (pattern analysis) όπως ονομάζεται όλη αυτή η διαδικασία ακολουθεί δύο στάδια. Πρώτον θα πρέπει μέσω μιας συνάρτησης kernel να μετασχηματίσουμε τα δεδομένα που δίνονται σε ένα νέο γραμμικώς διαχωριζόμενο χώρο και αφού γίνει αυτό, εύκολα πια μπορούμε να εφαρμόσουμε το συγκεκριμένο αλγόριθμο που αναφέρεται στο πρόβλημά μας. Κατόπιν εφαρμογής του αλγορίθμου μας θα υπολογίσουμε και την απόδοση της εφαρμογής του στα προβλήματα μας.
Αυτό που όμως έχει τη μεγαλύτερη σημασία είναι να μπορούμε να βγάλουμε συμπεράσματα όχι μόνο για το αν μπορούν να επιλύσουν τα προβλήματα αλλά και για το πόσο αποτελεσματικά τα αντιμετωπίζουν.

Θα ακολουθήσει σειρά πειραμάτων όπου δοκιμάζουμε τις μηχανές SVM για να καταλήξουμε σε συμπεράσματα για το πόσο ικανοποιητικά ανταποκρίνεται σε προβλήματα κατηγοριοποίησης.

Τα κεφάλαια αναπτύσσονται ως εξής:

Κεφάλαιο 1: αναλύουμε τα νευρωνικά δίκτυα και τις διάφορες εφαρμογές τους.

Κεφάλαιο 2: εξηγούμε τον τρόπο λειτουργίας των Support Vector Machine, τις μεθόδους που χρησιμοποιούν και τους διάφορους τύπους που υπάρχουν.

Κεφάλαιο 3: παρουσιάζουμε το ερευνητικό πρόγραμμα LibSVM και τον τρόπο λειτουργίας του.

Κεφάλαιο 4: χρησιμοποιούμε παραδείγματα με πραγματικά δεδομένα για να δοκιμάσουμε την αξιοπιστία τόσο των Support Vector Machine όσο και του προγράμματος LibSVM.

ΚΕΦΑΛΑΙΟ 1 – ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

1.1 Γιατί χρησιμοποιούμε ένα Νευρωνικό Δίκτυο

Τα Νευρωνικά Δίκτυα έχουν την ικανότητα να επεξεργάζονται και να εκτελούν ένα πολύ μεγάλο αριθμό από διάφορα καθήκοντα συγχρόνως. Αυτή η δυνατότητα των Νευρωνικών Δικτύων προέρχεται από την ιδιαίτερη δομή τους και από την ικανότητα τους να μαθαίνουν και επομένως να γενικεύουν. Μαθαίνουν ακριβώς όπως και ο ανθρώπινος εγκέφαλος ,δια της επαναλήψεως, και αφού αποθηκεύουν τις γνώσεις τους, τις γενικεύουν, ώστε ανά πάσα στιγμή να μπορούν να τις εφαρμόσουν και σε δεδομένα , που δεν ήταν τμήμα των δεδομένων με τα οποία εκπαιδεύτηκαν. Αυτό εξάλλου σημαίνει και ΓΕΝΙΚΕΥΣΗ-Η ικανότητα του εκπαιδευμένου Νευρωνικού Δικτύου να παράγει λογικές εξόδους για εισόδους που δεν συμμετείχαν στην διαδικασία εκπαιδεύσεως του.

1.1.1 Πλεονεκτήματα-Μειονεκτήματα των ΤΝΔ

Τα πλεονεκτήματα των Νευρωνικών Δικτύων στηρίζονται στις εξής ιδιότητες που έχουν :

Μη Γραμμικότητα : Ένα Νευρωνικό Δίκτυο μπορεί να είναι γραμμικό ,μπορεί να είναι και μη γραμμικό. Ένα Νευρωνικό Δίκτυο, που σχηματίζεται από ενώσεις μη γραμμικών νευρώνων , είναι από μόνο του μη γραμμικό.
Σχεδίαση Εισόδου –Εξόδου : Μία δημοφιλής μέθοδος εκμάθησης είναι η εκμάθηση με δάσκαλο ή επιβλέπων. Σε αυτή τη μέθοδο τα συναπτόμενα βάρη ενός Νευρωνικού Δικτύου αλλάζουν εφαρμόζοντας κάθε φορά ένα σύνολο από παραδείγματα εκπαίδευσης .Κάθε παράδειγμα αποτελείται από ένα μοναδικό σήμα εισόδου και μια αντίστοιχη μοναδική επιθυμητή απόκριση. Το δίκτυο παρουσιάζεται με ένα παράδειγμα που διαλέγουμε τυχαία από το σύνολο και τα συναπτόμενα βάρη(ελεύθεροι παράμετροι) του δικτύου τροποποιούνται έτσι ώστε να ελαχιστοποιήσουν την διαφορά ανάμεσα στην επιθυμητή και την πραγματική

απόκριση του δικτύου που προκάλεσε το σήμα εισόδου σύμφωνα με το κατάλληλο κριτήριο. Η εκπαίδευση του δικτύου επαναλαμβάνεται για πολλά παραδείγματα του συνόλου μέχρι το δίκτυο να φτάσει μια σταθερή κατάσταση όπου δεν θα υπάρχουν πια σημαντικές αλλαγές στις τιμές των συναπτόμενων βαρών. Τα προηγούμενα παραδείγματα εκπαίδευσης που εφαρμόστηκαν μπορεί να ξαναεφαρμοστούν κατά τη διάρκεια της εκπαίδευσης αλλά σε διαφορετική σειρά. Έτσι το δίκτυο μαθαίνει από τα παραδείγματα δημιουργώντας ένα χάρτη εισόδου-εξόδου για το συγκεκριμένο πρόβλημα.

Προσαρμοστικότητα: Τα Νευρωνικά Δίκτυα έχουν μία ικανότητα λόγω κατασκευής τους να προσαρμόζουν τα συναπτόμενα βάρη τους σε αλλαγές ,που οφείλονται στο γύρω από αυτά περιβάλλον. Συγκεκριμένα , ένα Νευρωνικό Δίκτυο εκπαιδευμένο να λειτουργεί σε ένα συγκεκριμένο περιβάλλον, μπορεί εύκολα να ξανά εκπαιδευτεί να χειρίζεται μικρές αλλαγές , που λάβανε χώρα στο περιβάλλον λειτουργίας τους. Επιπλέον , όταν ένα Νευρωνικό Δίκτυο λειτουργεί σε ένα μη-στάσιμο περιβάλλον, μπορεί να σχεδιαστεί έτσι ,ώστε να αλλάζει τα συναπτόμενα βάρη του σε πραγματικό χρόνο.

Αποδεικτική Ανταπόκριση: Κατά την ταξινόμηση των διαφόρων δειγμάτων , ένα Νευρωνικό Δίκτυο μπορεί να σχεδιαστεί έτσι , ώστε να παρέχει πληροφορίες όχι μόνο όσο αφορά ποίο δείγμα πρέπει να διαλέξουμε, αλλά και κατά πόσο σωστή ήταν η επιλογή μας αυτή. Αυτές οι πληροφορίες μπορούν να χρησιμοποιηθούν αργότερα, για να απορρίψουμε κάποια ανεπιθύμητα δείγματα και άρα να βελτιώσουμε την απόδοση του Δικτύου.

Συναφείς πληροφορίες :Κάθε νευρώνας του Δικτύου επηρεάζεται από την συνολική δραστηριότητα όλων των άλλων νευρώνων. Συνεπώς εκ φύσεως ένα Νευρωνικό Δίκτυο ασχολείται με συναφείς πληροφορίες.

Ανοχή στο λάθος . Ένα Νευρωνικό Δίκτυο έχει την δυνατότητα να είναι ανεκτικό στο λάθος και να κάνει σωστούς υπολογισμούς ,χωρίς μεγάλα σφάλματα Για παράδειγμα αν ένας νευρώνας ή ένας σύνδεσμος καταστραφεί , εξαιτίας της κατανεμημένης φύσης της πληροφορίας ,η οποία είναι αποθηκευμένη στο δίκτυο ,οι ζημιές θα πρέπει να είναι εκτεταμένες για να μειωθεί αρκετά η απόδοση του Δικτύου. Έτσι ένα Νευρωνικό Δίκτυο επιδεικνύει περισσότερο μια μείωση της απόδοσής του σε μια τέτοια ,σαν την προηγούμενη , περίπτωση παρά καταστροφή.

Υλοποιείται με VLSI :Η φύση του ,να υλοποιεί πολλούς υπολογισμούς συγχρόνως, το κάνει ένα γρήγορο και χρήσιμο εργαλείο κατά την εκτέλεση συγκεκριμένων

υπολογισμών. Αυτό του το χαρακτηριστικό μας δίνει την δυνατότητα να το χρησιμοποιούμε σε εφαρμογές , που υλοποιούνται με VLSI.

Σταθερότητα και ομοιομορφία στην ανάλυση και το σχεδιασμό του: Οι νευρώνες των Νευρωνικών Δικτύων είναι στην ουσία επεξεργαστές πληροφοριών. Οι νευρώνες με την μια μορφή τους ή την άλλη αντιπροσωπεύουν ένα κοινό συστατικό σε όλα τα Νευρωνικά Δίκτυα. Αυτή η ομοιότητα των Νευρωνικών Δικτύων όσο αφορά τους νευρώνες ,δίνει την δυνατότητα διαφορετικές εφαρμογές των Νευρωνικών Δικτύων να μοιράζονται διάφορες θεωρίες και αλγόριθμους εκμάθησης .

Μειονεκτήματα

Δεν υπάρχουν σαφής κανόνες για την ανάπτυξη ΤΝΔ για οποιαδήποτε εφαρμογή.
Δεν υπάρχει γενικός τρόπος για την ερμηνεία της εσωτερικής λειτουργίας του δικτύου.
Η εκπαίδευση μπορεί να είναι δύσκολη ή αδύνατη.
Η ικανότητα γενίκευσης είναι δύσκολα προβλέψιμη.

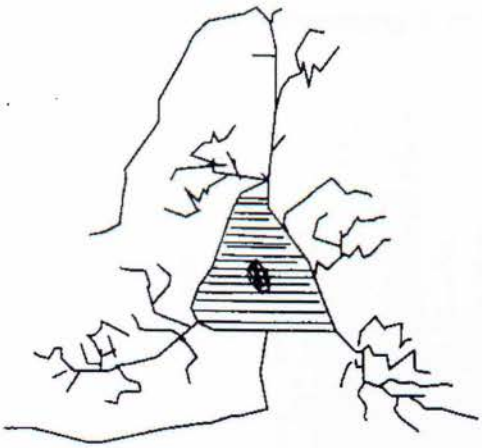
1.1.2 Λειτουργία του βιολογικού συστήματος

Επειδή η όλη λειτουργία και κατασκευή των ΤΝΔ στηρίζονται στις ιδιαίτερες υπολογιστικές ικανότητες και δυνατότητες του ανθρώπινου εγκεφάλου ,κρίναμε σκόπιμο να αναφερθούμε συνοπτικά στην λειτουργία του βιολογικού μας συστήματος.

Νευροβιολογική Αναλογία : Ο σχεδιασμός των Νευρωνικών Δικτύων βασίζεται στην δομή του εγκεφάλου. Οι νευροβιολόγοι μελετάνε τα Νευρωνικά Δίκτυα και την συμπεριφορά τους , για να ερμηνεύσουν διάφορα νευροβιολογικά φαινόμενα. Αντίστοιχα οι μηχανικοί μελετάνε την νευροβιολογία , για να βρουν νέες ιδέες στην επίλυση προβλημάτων , που είναι πιο πολύπλοκα από αυτά ,που βασίζονται σε συμβατικές τεχνικές σχεδίασης. Παρακάτω θα αναφερθούμε στη λειτουργία του βιολογικού συστήματος ώστε να έχουμε ολοκληρωμένη ιδέα για το τι προσπαθούμε να πετύχουμε με την λειτουργία των Νευρωνικών Δικτύων.

Η λειτουργία του βιολογικού συστήματος βασίζεται στη διασύνδεση εξειδικευμένων φυσικών κυττάρων που ονομάζονται νευρώνες. Οι σημαντικές

ιδιότητες των βιολογικών συστημάτων , όπως η προσαρμοστικότητα , η ικανότητα αναγνώρισης από τα συμφραζόμενα , η ανοχή στα λάθη ,η μεγάλη χωρητικότητα μνήμης , η ικανότητα επεξεργασίας βιολογικών πληροφοριών σε πραγματικό χρόνο (κυρίως από τον εγκέφαλο), μας κατευθύνουν στη μελέτη και την προσπάθεια προσομοίωσης αυτών των εναλλακτικών βιολογικών αρχιτεκτονικών. Ωστόσο, δεν είναι ακόμα επαρκώς γνωστός ο τρόπος με τον οποίο λειτουργεί ο ανθρώπινος εγκέφαλος. Επιπλέον, παρά το γεγονός ότι το βασικό στοιχείο υπολογισμού του ανθρώπινου

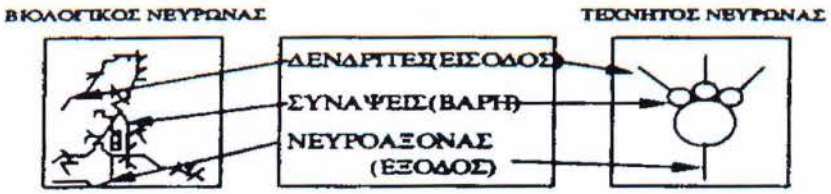


Σχήμα 1 Βιολογικός νευρώνας

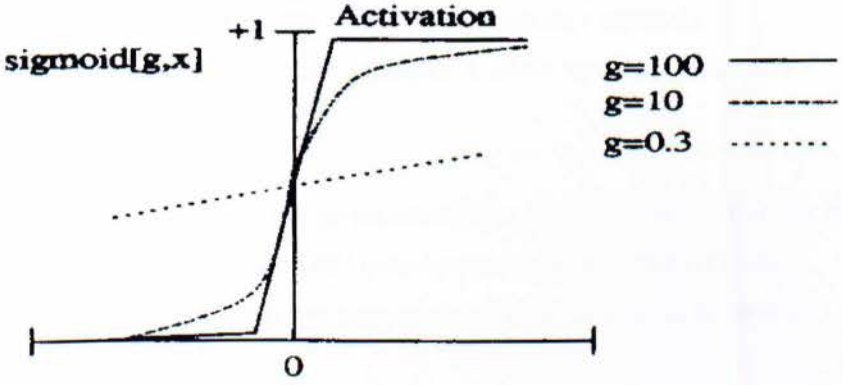
συστήματος επεξεργασίας είναι σχετικά αργό (σε σύγκριση με τα ηλεκτρονικά στοιχεία), η συνολική επεξεργασία επιτυγχάνεται σε μερικές εκατοντάδες msec. Αυτό μας οδηγεί σε συμπέρασμα ότι η βάση του βιολογικού υπολογισμού είναι ένας μικρός αριθμός ακολουθιακών βημάτων , κάθε ένα από τα οποία εκτελείται με μεγάλο παραλληλισμό. Επιπλέον, στην έμφυτη αυτή παράλληλη αρχιτεκτονική ,κάθε μονάδα επεξεργασίας είναι σχετικά απλή και τοπικά συνδεδεμένη.

Στο προηγούμενο σχήμα παρουσιάζεται η βασική δομή ενός βιολογικού νευρώνα , ο οποίος αποτελείται από το σώμα, τον άξονα, τους δενδρίτες και τις συνάψεις. Η βασική λειτουργία που επιτελεί ένας νευρώνας είναι η συσσώρευση των σημάτων που δέχεται από τους νευρώνες με τους οποίους συνδέεται η είσοδός του, το φιλτράρισμα και η ενίσχυση αυτών των σημάτων, και η παραγωγή ενός σήματος εξόδου το οποίο στη συνέχεια μεταδίδεται μέσω των συνάψεων προς τους

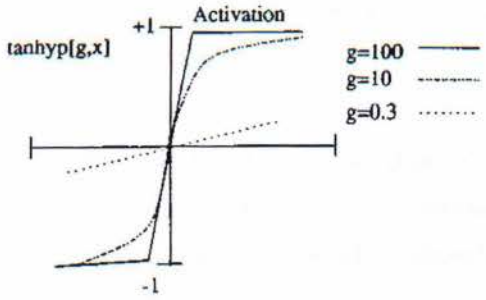
νευρώνες με τους οποίους συνδέεται η έξοδος του. Ένα πολύ σημαντικό στοιχείο είναι ότι η επίδραση ενός νευρώνα στους γειτονικούς του μπορεί να είναι είτε διεγερτική (θετική σύνδεση) είτε ανασταλτική (αρνητική σύνδεση). Σε πλήρη αντιστοιχία με το απλοποιημένο αυτό μοντέλο του βιολογικού νευρώνα αναπτύχθηκε το μοντέλο του τεχνητού νευρώνα (επόμενο σχήμα). Ας θεωρήσουμε έναν τεχνητό νευρώνα με d συνδέσεις εισόδου $x_1,\ldots..x_d$, με αντίστοιχες τιμές βαρών $w_1,\ldots.w_d$. Ο υπολογισμός που επιτελεί ένας νευρώνας διακρίνεται σε δύο στάδια: α) υπολογισμός της ενεργοποίησης $\boxed{u= \Sigma_i^d=w_ix_i + \theta}$, όπου θ η πόλωση του νευρώνα , β) ο υπολογισμός της εξόδου y του νευρώνα περνώντας την ενεργοποίηση u μέσα από μια συνάρτηση ενεργοποίησης f: y= f(u).



Σχήμα 2 : Αντιστοιχία βιολογικού-τεχνητού νευρώνα.



Σχήμα 3 : Η λογιστική συνάρτηση



Σχήμα 4 Η συνάρτηση υπερβολικής εφαπτομένης.

Η συνάρτηση ενεργοποίησης είναι συνήθως μη γραμμική και στις περισσότερες περιπτώσεις σιγμοειδής. Δύο βασικοί τύποι σιγμοειδών συναρτήσεων είναι:

Η λογιστική : f(u)= 1/ (1+ exp(-gu)), η οποία παρέχει τιμές στο (0,1) , όπου g είναι μια παράμετρος που καθορίζει την κλίση της σιγμοειδούς (Σχήμα 3)
Η υπερβολική εφαπτόμενη : f(u)= tanh(gu), η οποία παρέχει τιμές στο (-1, 1) (Σχήμα 4)

Επίσης, σε ορισμένε περιπτώσεις χρησιμοποιείται η γραμμική έξοδος (f(u) =u) και σε ορισμένες άλλες (π.χ δίκτυα RBF) η συνάρτηση καμπάνας (bell fuction) .
Αναλυτικότερα για την συνάρτηση ενεργοποίησης θα αναφερθούμε παρακάτω.

1.1.3 Νευρομορφικός Υπολογισμός

Ο όρος νευρομορφική μηχανή (neuromorphic engineering) αναφέρεται σε μια νέα τεχνολογία , βασισμένη στο σχεδιασμό και την κατασκευή τεχνητών νευρωνικών συστημάτων , των οποίων οι αρχιτεκτονικές και σχεδιαστικές αρχές ακολουθούν τις ίδιες αρχές που διέπουν και το βιολογικό νευρικό σύστημα. Η νευρομορφική μηχανή έχει ένα ευρύ φάσμα εφαρμογών από τον έλεγχο πολύπλοκων συστημάτων μέχρι το σχεδιασμό ευφυών αισθητήρων. Οι περισσότερες αρχές που ακολουθούνται σε αυτό το πεδίο της τεχνολογίας , όπως οι μέθοδοι εκπαίδευσης και η υλοποίηση υλικού παράλληλων υπολογιστών , είναι εμπνευσμένες από το βιολογικό σύστημα.

1.1.4 Ποιες είναι οι υπολογιστικές ιδιότητες του ανθρώπινου εγκεφάλου;

Ένα από τα παράδοξα θέματα που αφορούν τον ανθρώπινο εγκέφαλο είναι ότι παρά το γεγονός ότι γνωρίζουμε καλά τη φυσιολογία του νευρικού συστήματος , η ικανότητα παραγωγής υπολογισμών υψηλού επιπέδου εξακολουθεί να αποτελεί ένα μυστήριο.

Ο αριθμός των μονάδων επεξεργασίας και η πολυπλοκότητα των μεταξύ τους διασυνδέσεων στον ανθρώπινο εγκέφαλο είναι τεράστια και προέκυψαν από την ανάγκη των έμβιων όντων για επιβίωση. Μερικές από τις βασικές ικανότητες του ανθρώπινου εγκεφάλου είναι οι ακόλουθες:

Ο εγκέφαλος βιώνει και αποθηκεύει εμπειρίες. Τέτοιες εμπειρίες μπορεί να είναι η κατηγοριοποίηση ή συσχέτιση των δεδομένων εισόδου. Με αυτήν την έννοια αυτό-οργανώνει τις εμπειρίες.

Ο εγκέφαλος αποκρίνεται σε νέες εμπειρίες μέσω της συσχέτισής τους με τις αποθηκευμένες.

Ο εγκέφαλος είναι ικανός να εκτελεί προβλέψεις για νέες καταστάσεις σύμφωνα με τις εμπειρίες που έχει ήδη αποθηκευμένες ,δηλαδή χαρακτηρίζεται από ικανότητα γενίκευσης.

Ο εγκέφαλος δεν απαιτεί πλήρεις πληροφορίες για να αποφασίσει. Έχει μεγάλη ανοχή στην παραμόρφωση ,διαταραχή ή ατέλεια των δεδομένων εισόδου. Αυτό προϋποθέτει έναν ακόμα τρόπο ικανότητας γενίκευσης.

Ο εγκέφαλος αποτελεί μια μηχανή υπολογισμών με μεγάλη ανοχή στις βλάβες. Ακόμη και η απώλεια μερικών νευρώνων αντιμετωπίζεται με την κατάλληλη προσαρμογή αυτών που απομένουν και με πρόσθετη εκπαίδευση.

Ο εγκέφαλος φαίνεται να έχει διαθέσιμους νευρώνες ,πιθανώς αχρησιμοποίητους ,έτοιμους προς χρήση. Αυτό σημαίνει ότι έχει τη δυνατότητα διαρκώς να μαθαίνει.

Η μικροσκοπική ή μακροσκοπική εξέταση του εγκεφάλου δεν παρέχει αρκετές πληροφορίες για την υψηλού επιπέδου λειτουργία του. Για παράδειγμα , η ανάλυση των δράσεων του εγκεφάλου δε μας εξηγεί τον τρόπο επίλυσης προβλημάτων και σκέψης. Αυτή η αδιαφάνεια στη λειτουργία του εγκεφάλου μεταφέρεται συχνά και στα ΤΝΔ, τα οποία είναι δυνατό να παρέχουν μη εξηγήσιμες λύσεις.

Ο εγκέφαλος αν και αξιοθαύμαστος αδυνατεί να συναγωνιστεί σε πολλές λειτουργίες τους ψηφιακούς υπολογιστές που είναι σήμερα διαθέσιμοι.

1.2 Νευρωνικές προσεγγίσεις για την επίλυση προβλημάτων

1.2.1 Εκπαίδευση ή προγραμματισμός

Παραδοσιακά ο προγραμματισμός απαιτεί αυστηρή σύνταξη ,διάφορες γλώσσες προσανατολισμένες σε κάθε είδους εφαρμογή και εξειδικευμένους προγραμματιστές. Η εναλλακτική λύση προέρχεται από τα βιολογικά συστήματα και βασίζεται στην εκπαίδευση. Τα παιδιά για παράδειγμα δεν προγραμματίζονται ,μαθαίνουν μέσω της εκπαίδευσης και της προσαρμογής. Φυσικά για να υλοποιηθεί κάτι τέτοιο πρέπει τόσο η υπολογιστική μηχανή να είναι 'εκπαιδεύσιμη', όσο και τα κατάλληλα δεδομένα εκπαίδευσης να είναι διαθέσιμα.

1.2.2 Μαθηματικά μοντέλα και προσομοίωση

Η πιο πρόσφατη προσανατολισμένη στο μη βιολογικό τρόπο ,έρευνα πάνω σε νευρωνικά δίκτυα αφορά την ανάπτυξη, τον χαρακτηρισμό και την επέκταση των μαθηματικών μοντέλων νευρωνικών δικτύων. Ένα τέτοιο μοντέλο αναφέρεται συνήθως σε ένα σύνολο μη γραμμικών εξισώσεων που χαρακτηρίζουν τη συνολική λειτουργία του δικτύου, τη δομή του ,τη δυναμική του και την εκπαίδευση. Συχνά εφαρμόζονται και διαφορικές εξισώσεις.

Πρέπει να σημειωθεί ότι όλα όσα είναι μέχρι σήμερα γνωστά για την λειτουργία και την επιτυχία των νευρωνικών δικτύων, έχουν ανακαλυφθεί από προσομοιώσεις σε υπολογιστές. Η προσομοίωση συχνά απαιτεί τεράστιους υπολογιστικούς πόρους και δεν είναι σπάνιες οι φορές που αναγκαζόμαστε να τροποποιήσουμε την πραγματική υπολογιστική δομή του δικτύου. Εκτίμηση της ακριβούς συμπεριφοράς των ΤΝΔ σε πραγματικές εφαρμογές θα είναι δυνατή όταν θα γίνουν διαθέσιμα μεγάλα παράλληλα υπολογιστικά συστήματα.

1.2.3 Συνδετικά μοντέλα και υπολογισμός

Η συνδετική φιλοσοφία βασίζεται στην αντίληψη ότι πολλές ανθρώπινες υπολογιστικές διαδικασίες (οι οποίες μπορούν να εξομοιωθούν στην μηχανική ευφυία) εκτελούνται σε φυσικό παραλληλισμό αλληλεπιδρώντας μεταξύ τους. Στην ουσία , ο συνολικός υπολογισμός είναι κατανεμημένος σε έναν μεγάλο αριθμό , συχνά απλών υπολογιστικών μονάδων ,που κάθε μια έχει επιμέρους συμβολή στην όλη προσπάθεια. Ο αριθμός των μονάδων είναι μεγάλος , οι συνδέσεις μεταξύ τους αυστηρά περιορισμένες σε τοπικό επίπεδο και η πολυπλοκότητά τους μικρή. Τα νευρωνικά δίκτυα ικανοποιούν αυτές τις απαιτήσεις.

Η 'συνδετική' προσέγγιση είναι μια γενίκευση της αντίληψης των νευρωνικών δικτύων, όπου οι μεμονωμένες μονάδες επιτρέπεται να έχουν αυξημένη πολυπλοκότητα σε σύγκριση με τους νευρώνες.

1.3 Εφαρμογές Νευρωνικών Δικτύων

Η εξομοίωση των βιολογικών υπολογιστικών παραδειγμάτων που πραγματοποιείται μέσω των ΤΝΔ έχει πολύ καλά αποτελέσματα για πολλές κατηγορίες προβλημάτων. Μεταξύ αυτών είναι :

Η κατηγορία των προβλημάτων αναγνώρισης (π.χ αναγνώριση φωνής ,εικόνας κ.λ.π)
Η κατηγορία των προβλημάτων ελέγχου των οποίων τα δεδομένα είναι ελλιπή, ασαφή και στοχαστικά.
Η κατηγορία των NP-πλήρων προβλημάτων, τα οποία περιλαμβάνουν προβλήματα δρομολόγησης , αναζήτησης κ.λ.π

Εκτός από τα παραπάνω προβλήματα τα ΤΝΔ προσφέρουν και λύσεις κυρίως σε προβλήματα που σχετίζονται με τον ανθρώπινο παράγοντα (αναγνώριση ομιλίας ,εικόνας ,χειρόγραφου κειμένου κ.λ.π)

1.3.1 Παραδείγματα εφαρμογών ΤΝΔ

Επεξεργασία εικόνας και μηχανική όραση (π.χ ταίριασμα εικόνας ,προεπεξεργασία ,κατάτμηση, ανάλυση ,συμπίεση εικόνας και επεξεργασία χρονικά μεταβαλλόμενων εικόνων).
Επεξεργασία σήματος (π.χ ανάλυση και μορφολογία σεισμικού σήματος).
Αναγνώριση προτύπων (π.χ εξαγωγή χαρακτηριστικών, ανάλυση και κατηγοριοποίηση σήματος radar, αναγνώριση φωνής ,κειμένου, χειρονομιών και πιστοποίηση ταυτότητας).
Ιατρική (π.χ ανάλυση ηλεκτροκαρδιογραφήματος ,ιατρική διάγνωση και επεξεργασία ιατρικής εικόνας).
Αμυντικά συστήματα (π.χ υποβρύχια ανίχνευση ναρκών).
Οικονομία (π.χ ανάλυση αγοράς μετοχών ,ασφάλεια συναλλαγών ,εκτίμηση φερεγγυότητας δανειζόμενου πελάτη ,εκτίμηση ακίνητης περιουσίας).
Σχεδίαση ,έλεγχος και αναζήτηση (π.χ παράλληλη υλοποίηση NP-προβλημάτων ,αυτόματος έλεγχος ,ρομποτική)

Τεχνητή νοημοσύνη (π.χ υλοποίηση έμπειρων συστημάτων).
Δυναμικά εξελισσόμενα συστήματα ,πρόβλεψη χρονοσειρών π.χ εκτίμηση κατάστασης συστήματος , ανίχνευση βλαβών και ανάκαμψη).
Επικοινωνία ανθρώπου-υπολογιστή.

1.4 Τι είναι τα Νευρωνικά Δίκτυα

1.4.1 Εισαγωγή

Αφού είδαμε το λόγο που χρησιμοποιούνται σήμερα τα ΤΝΔ και τις εφαρμογές που βρίσκουν, θα προχωρήσουμε στην ανάλυση των Νευρωνικών Δικτύων ,ξεκινώντας από τον ορισμό τους.

1.4.2 Ορισμός

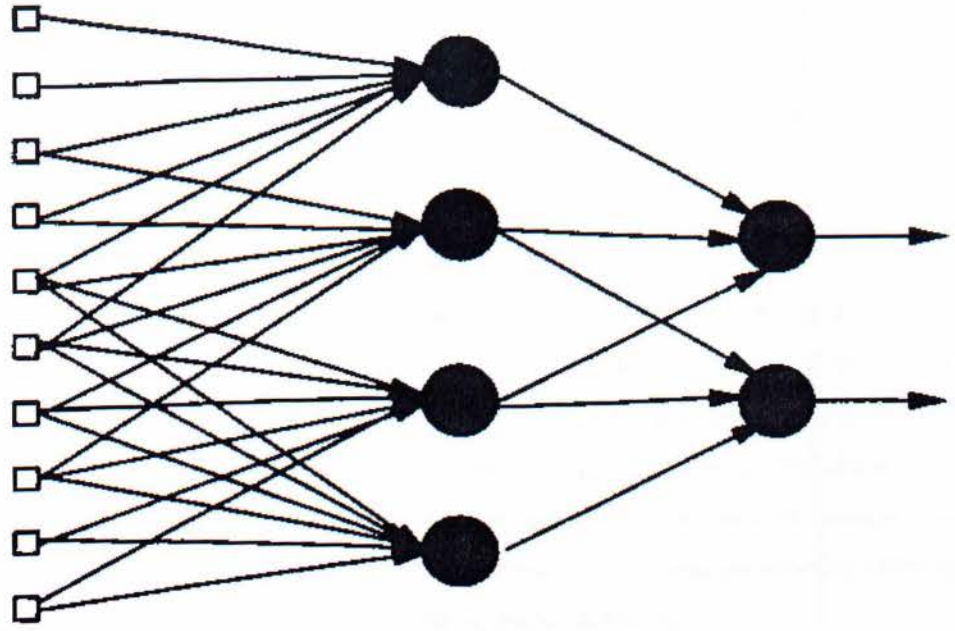
Τα Νευρωνικά Δίκτυα ορίζονται σαν ένα μεγάλο σύνολο από μονάδες , τους Νευρώνες , οι οποίοι είναι οργανωμένοι σε διαφορετικά επίπεδα-στρώματα- και τα οποία συνδέονται μεταξύ τους. Είναι μία αρχιτεκτονική δομή (δίκτυο) αποτελούμενη από ένα πλήθος διασυνδεδεμένων μονάδων (τεχνητοί νευρώνες). Κάθε μονάδα χαρακτηρίζεται από εισόδους και εξόδους και υλοποιεί τοπικά ένα απλό υπολογισμό. Κάθε σύνδεση μεταξύ δύο μονάδων χαρακτηρίζεται από μια τιμή βάρους. Οι τιμές των βαρών των συνδέσεων αποτελούν την γνώση που είναι αποθηκευμένη στο δίκτυο και καθορίζουν την λειτουργικότητά του. Η έξοδος κάθε μονάδας καθορίζεται από τον τύπο της μονάδας ,τη διασύνδεση με τις υπόλοιπες μονάδες και πιθανώς κάποιες εξωτερικές εισόδους. Πέρα από μια πιθανή δεδομένη (εκ κατασκευής)λειτουργική ικανότητα ενός δικτύου , συνήθως ένα δίκτυο αναπτύσσει μια συνολική δραστηριότητα μέσω μιας μορφής εκπαίδευσης .

Η συνολική λειτουργικότητα ενός ΤΝΔ καθορίζεται από την τοπολογία του Δικτύου, τα χαρακτηριστικά των νευρώνων ,τη μέθοδο εκπαίδευσης και από τα δεδομένα με τα οποία γίνεται η εκπαίδευση. Ο υπολογισμός που εκτελεί κάθε νευρώνας είναι απλός και κοινός για όλους τους νευρώνες. Επειδή οι νευρώνες λειτουργούν παράλληλα (ταυτόχρονα) και ο αριθμός τους μπορεί να είναι πολύ

μεγάλος , τα ΤΝΔ αποτελούν χαρακτηριστικό παράδειγμα μαζικά παράλληλου υπολογισμού.

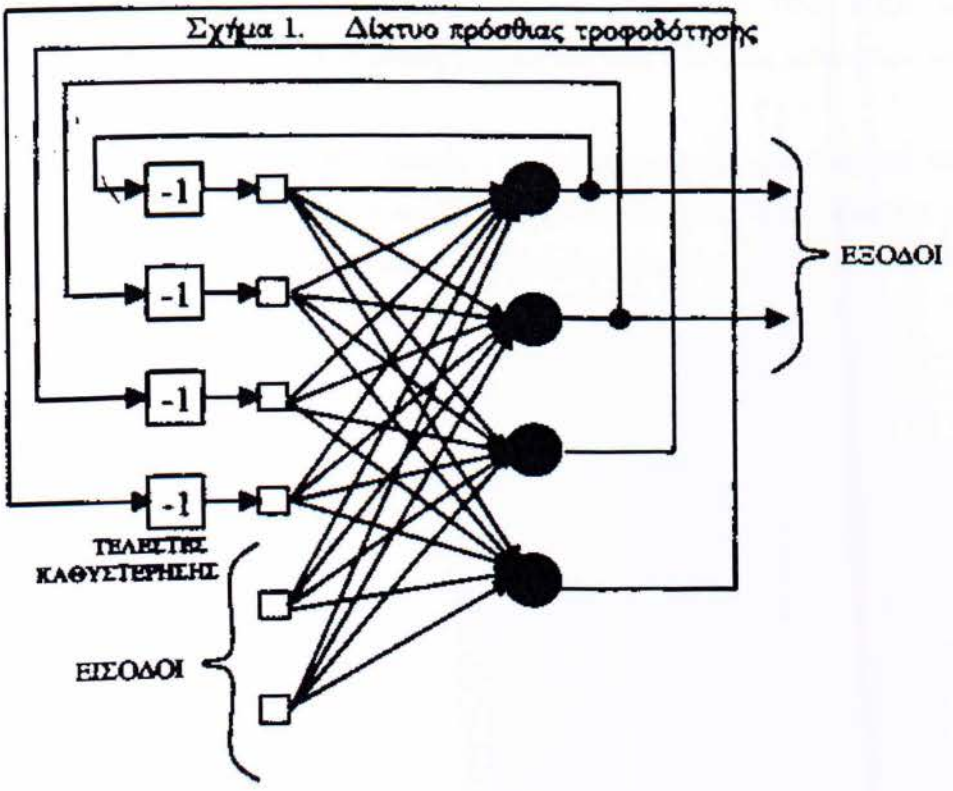
1.4.3 Βασικές Αρχές Νευρωνικών Δικτύων

Τα Νευρωνικά Δίκτυα αποτελούνται από συνδέσεις απλών στοιχείων ή μονάδων. Οι συνδέσεις αυτές χαρακτηρίζονται από βάρη. Στα επόμενα σχήματα απεικονίζονται δυο παραδείγματα δικτύων μικρής κλίμακας στα οποία οι μονάδες παρουσιάζονται με κύκλους και οι συνδέσεις με βέλη. Στο πρώτο Σχήμα απεικονίζεται ένα μη-ανατροφοδοτούμενο δίκτυο , το οποίο δηλαδή δεν περιέχει κλειστά μονοπάτια από συνδέσεις . Οι μονάδες ομαδοποιούνται σε επίπεδα (ή στρώματα-layers). Αντίθετα ,στο δεύτερο Σχήμα παρουσιάζεται η αρχιτεκτονική της ανατροφοδότησης η οποία επιτρέπει την ύπαρξη κύκλων από συνδέσεις μεταξύ μονάδων. Αυτή η δεύτερη αρχιτεκτονική προσδίδει στο δίκτυο πολύ περισσότερες δυνατότητες ,αλλά είναι πιο δύσκολο να αντιμετωπιστεί μαθηματικά. Επίσης , πρέπει να σημειωθεί ότι η τοπολογία των δικτύων μπορεί να είναι είτε στατική είτε δυναμική (μεταβαλλόμενη). Τέλος οι μονάδες που δεν έχουν συνδέσεις με τον εξωτερικό κόσμο, λέμε ότι είναι κρυμμένες (hidden) ή εσωτερικές.



ΕΠΙΠΕΔΟ ΕΙΣΟΔΟΥ | ΚΡΥΜΜΕΝΟ ΕΠΙΠΕΔΟ | ΕΠΙΠΕΔΟ ΕΞΟΔΟΥ

Σχήμα 1. Δίκτυο πρόσθιας τροφοδότησης



Σχήμα 5 : Επαναληπτικό δίκτυο

Κάθε μονάδα υλοποιεί μία συνάρτηση τοπικά και ολόκληρο το δίκτυο υλοποιεί μία συγκεκριμένη λειτουργία. Ο καθορισμός των παραμέτρων του δικτύου (τιμών των βαρών) που θα ικανοποιούν αυτές τις προδιαγραφές επιτυγχάνεται μέσω της διαδικασίας της μάθησης.

Η γνώση , η εμπειρία και η εκπαίδευση του δικτύου αποθηκεύονται στις διασυνδέσεις των μονάδων και στις τιμές των βαρών.

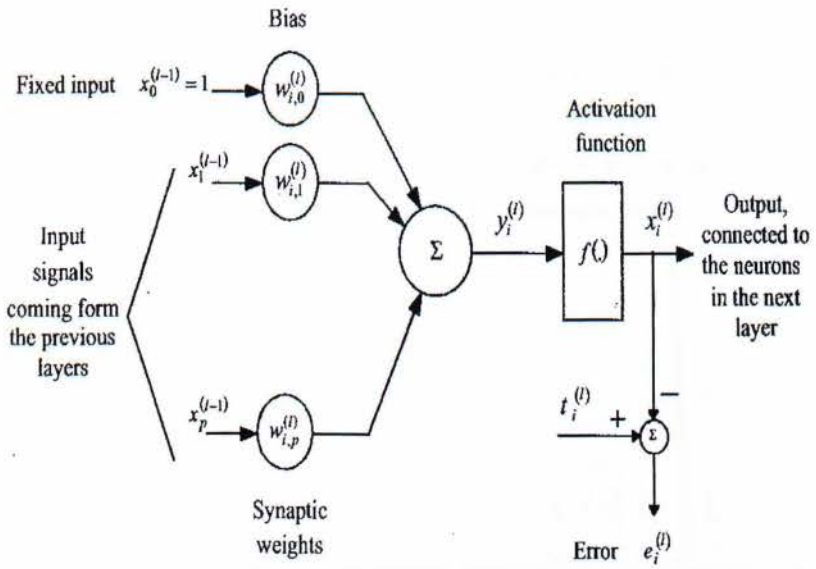
Στην πλειοψηφία τους τα ΤΝΔ εκπαιδεύονται με την ελπίδα ότι θα παρουσιάσουν καλή γενικευτική ικανότητα όταν θα τους ζητηθεί να αναγνωρίσουν ή να κατηγοριοποιήσουν καινούρια (άγνωστα) δεδομένα (πρότυπα). Αυτός είναι ο αντικειμενικός στόχος της διαδικασίας εκπαίδευσης , να αναπτύξει δηλαδή το ΤΝΔ κατάλληλη εσωτερική δομή ώστε να αναγνωρίζει πρότυπα που θα μοιάζουν με αυτά με τα οποία εκπαιδεύτηκε. Τα ΤΝΔ εκπαιδεύονται τόσο με τεχνικές μάθησης με επίβλεψη όσο και με τεχνικές μάθησης χωρίς επίβλεψη.

1.4.4 Τι είναι οι Νευρώνες

Μια και οι νευρώνες είναι το σημαντικότερο τμήμα ενός ΤΝΔ ,αξίζει να αναφερθούμε πιο αναλυτικά. Εξάλλου μελετώντας τους νευρώνες μαθαίνουμε και τα ΤΝΔ.

Οι Νευρώνες είναι απλοί επεξεργαστές , οι οποίοι χειρίζονται αποκλειστικά και μόνο τοπικά δεδομένα , τα οποία λαμβάνουν ως εισόδους μέσω των συνδέσεων.

Στο επόμενο σχήμα φαίνεται το μοντέλο ενός απλού νευρώνα.



Σχήμα 6 : μοντέλο απλού νευρώνα

Τα τρία βασικά χαρακτηριστικά ενός νευρώνα είναι :

Ένα σύνολο από συνδέσμους , ο καθένας από τους οποίους χαρακτηρίζεται από το δικό του βάρος (synaptic weights). Ένα σήμα x_j (j=1, …, m) στην είσοδο του συνδέσμου j, ο οποίος συνδέεται με τον σύνδεσμο K, πολλαπλασιάζεται με το συναπτόμενο βάρος w_{kj}.

Έναν αθροιστή , ο οποίος όπως φαίνεται και στο παρακάτω σχήμα , αθροίζει τα σήματα εισόδου , τα οποία είναι ήδη πολλαπλασιασμένα με τα αντίστοιχα συναπτόμενα βάρη των συνδέσεων. Η όλη παραπάνω λειτουργία είναι γραμμική.

Μια συνάρτηση , που περιορίζει το πλάτος της εξόδου του κάθε νευρώνα. Οι νευρώνες μπορούν να χρησιμοποιούν διαφορετικές συναρτήσεις μεταφοράς , για να δημιουργήσουν την έξοδό τους.

Αναφερόμενοι στο παραπάνω σχήμα θα λέγαμε ότι οι νευρώνες χρησιμοποιούν μη γραμμικές συναρτήσεις ενεργοποίησης που μετατρέπουν το διάνυσμα εισόδου σε μία έξοδο. Τα συναπτόμενα βάρη επιτρέπουν την σύνδεση μεταξύ διαφορετικών επιπέδων- στρωμάτων του Νευρωνικού Δικτύου . Τα βάρη αυτά είναι προσαρμοσμένα σε ένα αλγόριθμο εκμάθησης , με στόχο την εκπαίδευση του Νευρωνικού Δικτύου.

α) Τύποι της συνάρτησης f –(activation function):

Η συνάρτηση f είναι η συνάρτηση ,που χρησιμοποιεί ο κάθε νευρώνας προκειμένου να εκτελέσει την λειτουργία του και να μετατρέψει το σήμα που δέχεται σαν είσοδο σε σήμα εξόδου , το οποίο θα μεταβεί σε άλλους νευρώνες σε επόμενα στρώματα. Παρακάτω θα παρουσιάσουμε τις διάφορες μορφές που μπορεί να έχει μια τέτοια συνάρτηση.

Η συνάρτηση κατωφλίου (σχήμα 1) :

$$f(v)=\begin{cases}1 & v\geq 0\\ 0 & v<0\end{cases}$$

Αυτή η μορφή της συνάρτησης ενεργοποίησης αναφέρεται στην βιβλιογραφία και ως η Heaviside συνάρτηση και αντίστοιχα οι νευρώνες που χρησιμοποιούν αυτού του είδους την συνάρτηση αναφέρονται ως McCulloch-Pits model. Σε αυτό το μοντέλο η έξοδος του νευρώνα παίρνει την τιμή 1 ,εάν το τοπικό πεδίο του νευρώνα είναι μη αρνητικό και την τιμή 0 σε διαφορετική περίπτωση.

Η βηματική ,γραμμική συνάρτηση - The piecewise linear function :

$$f(v)=\begin{cases}1 & v\geq \frac{1}{2}\\ v & -\frac{1}{2}<v<\frac{1}{2}\\ 0 & v\leq -\frac{1}{2}\end{cases}$$

όπου ο παράγοντας ενίσχυσης στη γραμμική περιοχή λειτουργίας θεωρείται μονάδα. Οι δύο επόμενες περιπτώσεις θεωρούνται ειδικές μορφές της βηματικής – γραμμικής συνάρτησης :

Η βηματική –γραμμική συνάρτηση γίνεται συνάρτηση κατωφλίου αν ο παράγοντας ενίσχυσης της γραμμικής περιοχής γίνει απείρως μεγάλος.
Ένας γραμμικός συνδυαστής- combiner υπάρχει αν η γραμμική περιοχή λειτουργίας διατηρείται χωρίς να φτάνει στα όρια του κορεσμού.

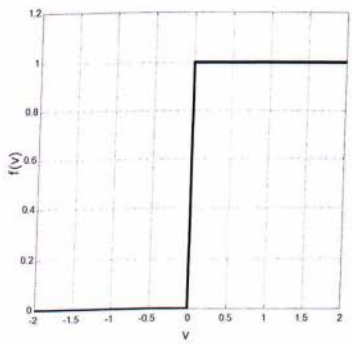
Η ημιτονοειδής συνάρτηση- sigmoid function , η οποία είναι και η πιο κοινά χρησιμοποιούμενη συνάρτηση στο σχεδιασμό των Νευρωνικών Δικτύων. Παρακάτω φαίνονται τα τρία ιδιαίτερα χαρακτηριστικά της ημιτονοειδούς συνάρτησης :

Αυστηρά αυξανόμενη συνάρτηση,
Ασυμπτωτικά περιορισμένη ,
Εξομαλυμένη

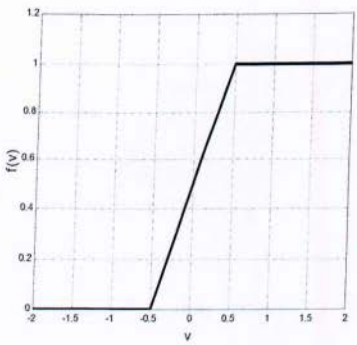
Δύο τύποι ημιτονοειδούς συνάρτησης βρίσκουν ιδιαίτερη εφαρμογή στα Νευρωνικά Δίκτυα. :
Η λογική συνάρτηση – logistic function :

$$f(v)=\frac{1}{1+\exp(-a\,v)} \quad ,$$

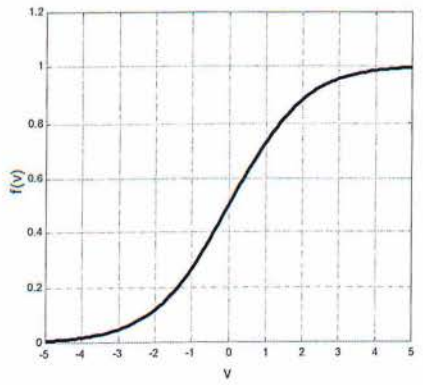
όπου το a συμβολίζει την κλίση της ημιτονοειδούς συνάρτησης. Μεταβάλλοντας την τιμή της παραμέτρου a , λαμβάνουμε ημιτονοειδείς συναρτήσεις διαφορετικής κλίσης. Η τιμή που έχει συνήθως η κλίση μιας ημιτονοειδούς συνάρτησης είναι ίση με a/4. Οριακά όμως καθώς η κλίση απειρίζεται, η ημιτονοειδής συνάρτηση μετατρέπεται σε συνάρτηση κατωφλίου. Δεδομένου ότι η συνάρτηση κατωφλίου παίρνει τιμές 0 ή 1 ,η ημιτονοειδής συνάρτηση σε αυτή την περίπτωση θα παίρνει συνεχείς τιμές από ένα διάστημα [0,1]. Μην ξεχνάμε ακόμα ότι η ημιτονοειδής συνάρτηση είναι διαφορίσιμη , ενώ η συνάρτηση κατωφλίου δεν είναι. Η εν λόγω συνάρτηση καθώς και όλες οι προηγούμενες συναρτήσεις που περιγράψαμε παραπάνω φαίνονται στα αμέσως επόμενα σχήματα.

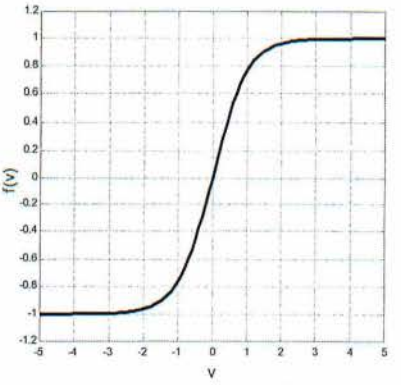


(α)



(b)





(c) (d)

Figure 7: Types of activation functions: (a) - the threshold function; (b) – the piecewise linear function; (c) - the logistic function and (d) - the hyperbolic tangent function

Οι παραπάνω συναρτήσεις ορίζονται στο διάστημα [0,1]. Μερικές φορές όμως είναι περισσότερο βολικό να ορίζουμε τις συναρτήσεις αυτές στο διάστημα [-1,+1]. Σε αυτές τις περιπτώσεις οι παραπάνω συναρτήσεις είναι περιττές συναρτήσεις του v. Η μορφή που έχει η συνάρτηση κατωφλίου , η τιμή της οποίας κυμαίνεται από –1 ως 1 , είναι γνωστή ως signum συνάρτηση.

Η υπερβολική εφαπτομενική συνάρτηση, η οποία απεικονίζεται στο παραπάνω σχήμα d,είναι η δεύτερη μετά την logistic μορφή της ημιτονοειδούς συνάρτησης και ορίζεται ως :

$$\tanh(v) = \frac{1-\exp(-v)}{1+\exp(-v)}$$

1.4.5 Διαδικασία κατασκευής ΤΝΔ

Η ανάπτυξη συστηματικής μεθοδολογίας για την κατασκευή νευρωνικών δικτύων για την επίλυση κάποιου προβλήματος δεν έχει ακόμα αναπτυχθεί. Βασικά προβλήματα που προκύπτουν είναι :

Είναι οι τεχνικές των ΤΝΔ κατάλληλες ή εφαρμόσιμες στο πρόβλημα; Το πρόβλημα έχει μία ή περισσότερες λύσεις;
Μπορούμε να τροποποιήσουμε γνωστά ΤΝΔ για να λύσουμε το πρόβλημα;
Υπάρχουν τρόποι ανάλυσης του προβλήματος (π.χ πολυπλοκότητα);

Η εφαρμογή της τεχνολογίας νευρωνικών δικτύων για την αντιμετώπιση κάποιου προβλήματος απαιτεί τον καθορισμό των στοιχείων του ΤΝΔ, όπως της αρχιτεκτονικής ,της τοπολογίας ,των παραμέτρων των μονάδων και της διαδικασίας εκπαίδευσης. Αν και φαίνεται απλό ,χρειάζεται αρκετή τεχνική κρίση. Υπάρχουν άπειροι συνδυασμοί παραμέτρων που είναι αδύνατο να δοκιμασθούν όλοι. Επιπλέον πρέπει να εξεταστεί η καταλληλότητα της νευρωνικής λύσης .
Κατά την διάρκεια της ανάπτυξης λύσεων βασισμένων σε ΤΝΔ προκύπτουν πολλά ερωτήματα, όπως:

Είναι δυνατόν το ΤΝΔ να εκπαιδευτεί ώστε να εκτελέσει την επιθυμητή λειτουργία;
Υποθέτοντας ότι υπάρχει λύση , ποιές είναι οι παράμετροι του δικτύου;
Τι υπολογιστικοί πόροι είναι απαραίτητοι;

Παρότι είναι αδύνατο να παρέχουμε έναν ολοκληρωμένο αλγόριθμο , παρακάτω παρουσιάζεται ένας βασικός σκελετός βημάτων που αντανακλά τα βασικά στάδια υλοποίησης .

Η πληθώρα των παραμέτρων σχεδίασης περιλαμβάνει :

Τοπολογία δικτύου και στρατηγική διασύνδεσης των μονάδων
Χαρακτηριστικά των μονάδων (μπορούν να διαφέρουν ανάλογα με τη θέση τους)
Διαδικασία εκπαίδευσης
Σύνολα εκπαίδευσης και ελέγχου

Αναπαράσταση εισόδου/εξόδου, προεπεξεργασία και μετεπεξεργασία.

α) Κατηγοριοποίηση των ΤΝΔ με βάση την δομή τους

Οποιαδήποτε περιγραφή ενός ΤΝΔ ξεκινάει με τον προσδιορισμό των εξής χαρακτηριστικών:
Τοπολογία δικτύου
Χαρακτηριστικά μονάδων
Λειτουργικότητα του δικτύου

β) Λειτουργίες ΤΝΔ

Μια προσέγγιση στο διαχωρισμό των ΤΝΔ προκύπτει από την επιθυμητή συμπεριφορά που θέλουμε αυτό να παρουσιάσει. Για παράδειγμα, η επιθυμητή λειτουργία ενός ΤΝΔ μπορεί να καθοριστεί με απαρίθμηση των καταστάσεων του δικτύου ή προσδιορίζοντας την επιθυμητή έξοδο βάσει των εισόδων και της τρέχουσας κατάστασης. Τα ΤΝΔ χωρίζονται στις παρακάτω κατηγορίες:

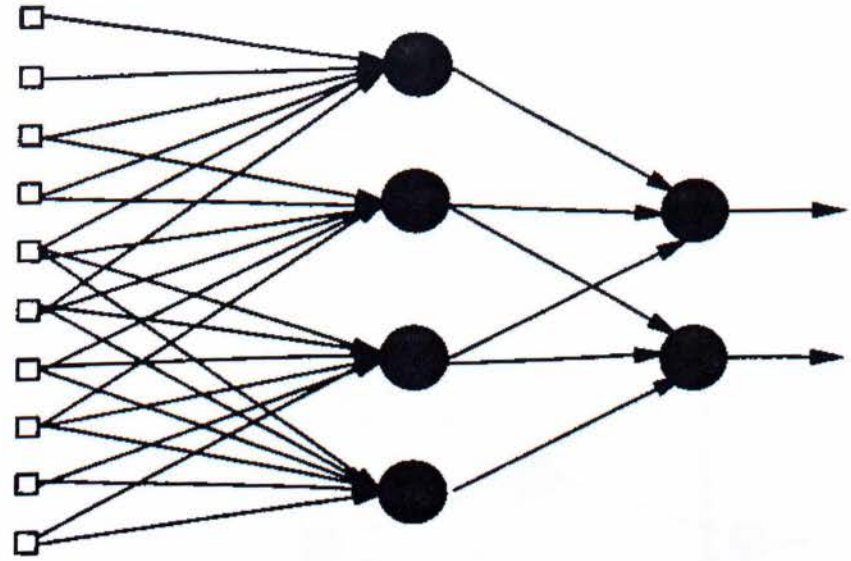
Συσχετιστές προτύπων. Η λειτουργία αυτού του δικτύου είναι η συσχέτιση προτύπων και η υλοποίηση επιθυμητών απεικονίσεων εισόδου-εξόδου. Συνήθως υλοποιούνται με δίκτυα προσωτροφοδότησης
Μοντέλο συσχετιστικής μνήμης. Αντιπροσωπευτικό παράδειγμα αποτελεί το δίκτυο Hopfield.
Αυτοοργανούμενα (self-organised) δίκτυα. Στην κατηγορία αυτή ανήκουν δίκτυα με ικανότητα μάθησης χωρίς επίβλεψη, τα οποία δηλαδή κατηγοριοποιούν την είσοδο σύμφωνα με κάποια κριτήρια ομοιότητας.

γ) Κατηγορίες και τοπολογία ΤΝΔ

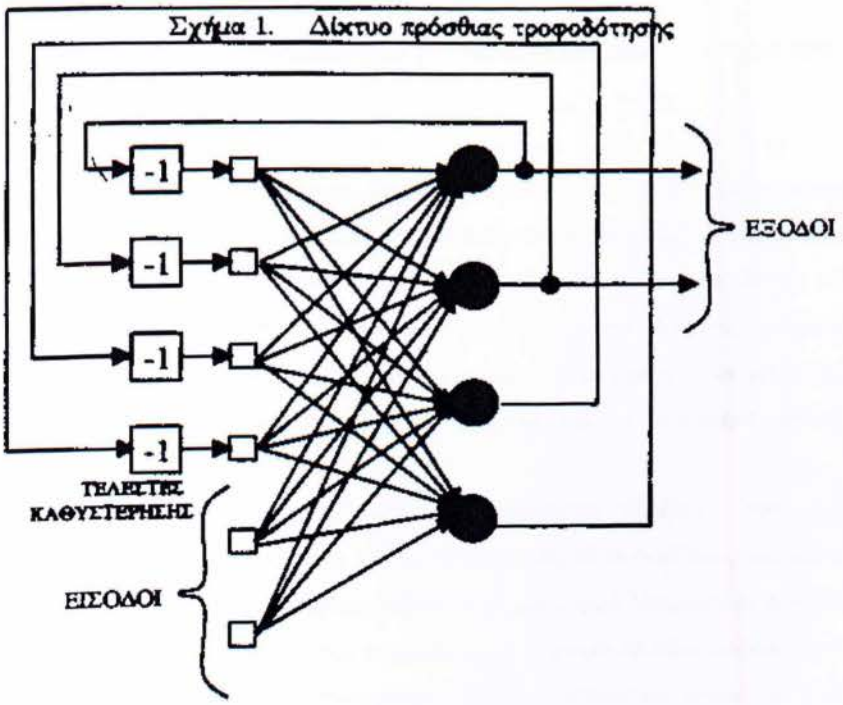
Θεωρώντας την τοπολογία και τη δομή των ΤΝΔ μπορούμε να διαχωρίσουμε τους παρακάτω τύπους:

Επαναληπτικά Δίκτυα (με συνδέσεις ανάδρασης) (Σχήμα 2)
Δίκτυα πρόσθιας τροφοδότησης (feedforward) (Σχήμα 1)
Δίκτυα με δομή επιπέδων ή ιεραρχική
Δίκτυα με ανταγωνιστικές (competitive) συνδέσεις

Μέχρι τώρα θεωρήσαμε ότι ο σχεδιαστής του δικτύου πρέπει να επιλέξει μία τοπολογία δικτύου. Πρόσφατες έρευνες προτείνουν ένα ακόμα στάδιο εκπαίδευσης το οποίο θα περιλαμβάνει συνεργασία και συνδυασμό διαφόρων τοπολογιών σε μία εφαρμογή. Παρότι η έρευνα είναι ακόμα σε αρχικό στάδιο, η ιδέα είναι η χρησιμοποίηση συνδυασμών
εκπαιδευμένων επιμέρους δικτύων (modular networks).



Σχήμα 1. Δίκτυο πρόσθιας τροφοδότησης

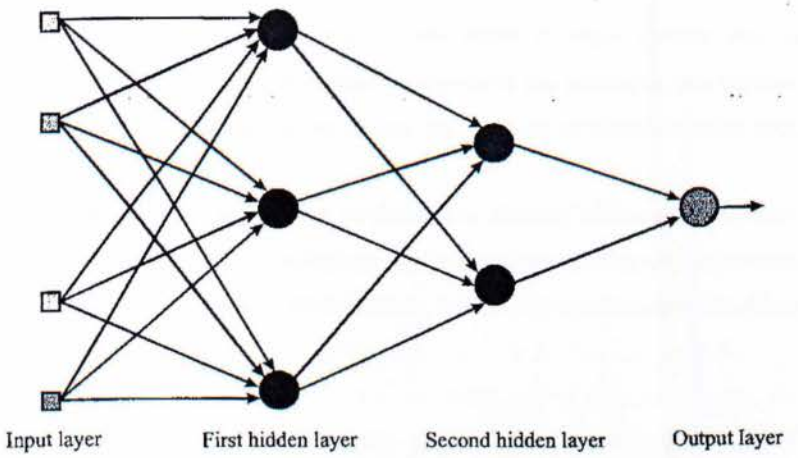


Σχήμα 8 Επαναληπτικό δίκτυο

1.4.6 Δομή Νευρωνικού Δικτύου

Στο επόμενο σχήμα φαίνεται η δομή ενός Νευρωνικού Δικτύου

Δομή ενός Νευρωνικού Δικτύου



Το Στρώμα εισόδου-Input layer- αναφέρεται στην είσοδο του Νευρωνικού Δικτύου. Από το σημείο αυτό εισέρχονται οι διάφορες μεταβλητές και παράμετροι ,βάσει των οποίων το Νευρωνικό Δίκτυο θα «αποφασίσει» και θα δείξει την απόφασή του αυτή στο στρώμα εξόδου –output layer – του Δικτύου, τι είδους διαμόρφωση έχει το προς αναγνώριση σήμα μας. Το κρυμμένο ή τα κρυμμένα στρώματα αποτελούνται και αυτά από νευρώνες και ενώνουν το στρώμα εισόδου με το στρώμα εξόδου.

Συγκεκριμένα η αρχιτεκτονική του Νευρωνικού Δικτύου , που εμάς ενδιαφέρει περισσότερο από κάθε άλλη αρχιτεκτονική, είναι αυτή ενός multilayer perceptron. Ο multilayer perceptron (MLP) είναι μια μορφή Νευρωνικού Δικτύου , που αποτελείται από ένα σύνολο νευρώνων , το στρώμα-επίπεδο εισόδου-input layer of nodes , ένα ή και περισσότερα σύνολα κρυμμένων νευρώνων , τα στρώματα-κρυμμένα επίπεδα-hidden layers of nodes και ένα σύνολο νευρώνων εξόδου, το στρώμα-επίπεδο εξόδου-output layer of nodes. Από τους νευρώνες του

στρώματος εισόδου εισέρχονται στο Νευρωνικό Δίκτυο τα προς επεξεργασία σήματά μας. Οι νευρώνες των κρυμμένων στρωμάτων επεξεργάζονται τα σήματα αυτά και μεταφέρουν τα αποτελέσματά τους στους νευρώνες του στρώματος εξόδου, από όπου και παρουσιάζονται στο χρήστη. Η εξωτερική πρόσβαση του χρήστη στα κρυμμένα στρώματα του Δικτύου είναι αδύνατη.
Από το παραπάνω σχήμα προκύπτουν δυο βασικά χαρακτηριστικά του multilayer perceptron:

Ένα multilayer perceptron είναι Δίκτυο ,στο οποίο το σήμα μπαίνει από το στρώμα εισόδου , διανύει το ή τα κρυμμένα στρώματα και καταλήγει στο στρώμα εξόδου , έχοντας διανύσει μια απόσταση από την αρχή ως το τέλος και μόνο προς μια κατεύθυνση.
Το Δίκτυο μπορεί να είναι πλήρως συνδεδεμένο. Δηλαδή κάθε νευρώνας ενός στρώματος μπορεί να είναι συνδεδεμένος με όλους τους νευρώνες του γειτονικού στρώματος. Ή να είναι μερικώς συνδεδεμένο , δηλαδή κάποια συναπτόμενα βάρη να απουσιάζουν.

Ο αριθμός των νευρώνων , που θα υπάρχει στο στρώμα εισόδου , εξαρτάται από την διάσταση των υπό αναγνώριση σημάτων. Αντίστοιχα ο αριθμός των νευρώνων στο στρώμα εξόδου εξαρτάται από την διάσταση που θα έχει το επιθυμητό σήμα εξόδου. Παρακάτω θα αναφερθούμε αναλυτικά στο MLP.
Έτσι σύμφωνα με όσα ειπώθηκαν παραπάνω ,για μας σχεδιασμός ενός ΤΝΔ σημαίνει εμπειρικά :

Ο προσδιορισμός των κρυμμένων στρωμάτων
Ο αριθμός των νευρώνων σε κάθε κρυμμένο στρώμα και,
Ο προσδιορισμός των συναπτόμενων βαρών μέσω των οποίων θα ενώνονται οι διάφοροι νευρώνες στα διάφορα στρώματα του Δικτύου.

Στην παρούσα εργασία θα χρησιμοποιηθεί ένας MLP με ένα κρυμμένο στρώμα-επίπεδο.

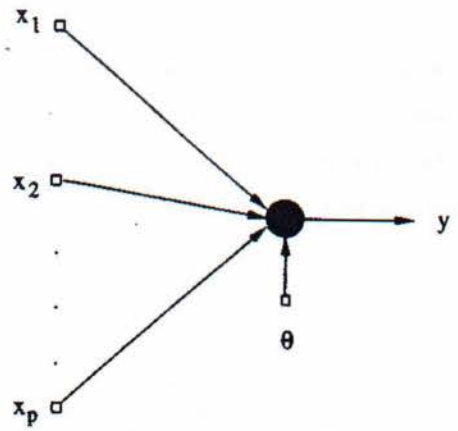
1.5 Το Perceptron

1.5.1 Εισαγωγή

Το Perceptron αποτελεί την πιο απλή μορφή νευρωνικού δικτύου που χρησιμοποιείται για την ταξινόμηση προτύπων, τα οποία πρέπει να είναι γραμμικά διαχωρισμένα: να υπάρχει ένα υπερεπίπεδο που να διαχωρίζει τα πρότυπα δύο διαφορετικών κατηγοριών.

Το perceptron αποτελείται από ένα απλό νευρώνα ο οποίος δέχεται εξωτερικές εισόδους μέσω συνδέσεων με βάρη και ,επιπλέον, διεγείρεται και από μια εξωτερική πόλωση, όπως φαίνεται στο επόμενο Σχήμα. Η πόλωση (bias) μπορεί να θεωρηθεί ως μια εξωτερικά εφαρμοζόμενη είσοδος σταθερής τιμής θ. Η τιμή του θ αποτελεί παράμετρο που ρυθμίζεται κατά την εκπαίδευση του δικτύου. Ο αλγόριθμος εκπαίδευσης που συνήθως χρησιμοποιείται για τον καθορισμό των τιμών των παραμέτρων (βάρη και πόλωση) του δικτύου αυτού αναπτύχθηκε από τον Rosenblatt (1958,1962). Ο Rosenblatt ,απέδειξε ότι αν τα πρότυπα που χρησιμοποιούνται για την εκπαίδευση του perceptron ανήκουν σε δύο γραμμικά διαχωρίσιμες κατηγορίες ,τότε ο αλγόριθμος εκπαίδευσης του δικτύου συγκλίνει σε τελικές τιμές παραμέτρων οι οποίες καθορίζουν τη θέση ενός υπερεπιπέδου που διαχωρίζει τα πρότυπα των δύο κατηγοριών. Η απόδειξη της σύγκλισης του παραπάνω αλγόριθμου εκπαίδευσης είναι γνωστή ως θεώρημα σύγκλισης του perceptron.

Ένα απλό perceptron που απεικονίζεται στο επόμενο σχήμα περιλαμβάνει έναν απλό νευρώνα . Ένα τέτοιο perceptron είναι περιορισμένο να εκτελεί ταξινομήσεις προτύπων που ανήκουν σε δύο μόνο κατηγορίες. Επεκτείνοντας το επίπεδο εξόδου του perceptron ώστε να συμπεριλάβουμε περισσότερους του ενός νευρώνες , μπορούμε αντίστοιχα να εκτελέσουμε ταξινομήσεις με περισσότερες της μία κατηγορίες. Ωστόσο, οι κατηγορίες θα πρέπει να είναι γραμμικά διαχωρίσιμες για να μπορούν να διαχωριστούν από το perceptron. Επομένως ,για την παρουσίαση της βασικής θεωρίας του perceptron , αρκεί να μελετήσουμε τη διάταξη με έναν μόνο νευρώνα, η οποία φαίνεται στο Σχήμα .



Σχήμα 9: Perceptron ενός επιπέδου

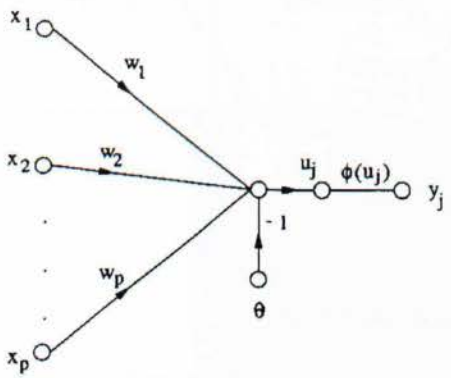
1.5.2 Βασικές θεωρήσεις

Η θεμελίωση της λειτουργίας του perceptron του Rosenblatt προέκυψε από το μοντέλο των McCulloch-Pitts για το βιολογικό νευρώνα. Το μοντέλο αυτό αποτελείται από μια μονάδα εσωτερικού γινομένου ακολουθούμενη από μια βηματική συνάρτηση κατωφλίου φ, όπως απεικονίζεται παρακάτω. Ο κόμβος άθροισης του μοντέλου υπολογίζει ένα γραμμικό συνδυασμό των εξωτερικών εισόδων με τα βάρη των αντιστοίχων συνδέσεων και επίσης προσθέτει την εξωτερικά εφαρμοζόμενη πόλωση. Το άθροισμα που προκύπτει περνά από τη βηματική συνάρτηση. Ο νευρώνας παράγει μια έξοδο που είναι 1 αν η είσοδος της βηματικής συνάρτησης είναι θετική και –1 αν αυτή είναι αρνητική.

Στο επόμενο σχήμα θεωρούμε ένα perceptron με p εισόδους το οποίο δέχεται ως είσοδο ένα διάνυσμα $x=(x_1,x_2,\ldots\ldots,x_p)^T \varepsilon R^p$. Τα βάρη των συνδέσεων δηλώνονται ως $w=(w_1,w_2,\ldots\ldots,w_p)^T$ και η εξωτερικά εφαρμοζόμενη πόλωση δηλώνεται ως θ. Από το μοντέλο του παρακάτω σχήματος προκύπτει ότι η έξοδος της μονάδας εσωτερικού γινομένου είναι

$$U=\sum w_i x_i-\theta \quad (1)$$

Σκοπός του perceptron είναι να ταξινομήσει την είσοδο $x=(x_1,x_2,\ldots\ldots,x_p)^T$ σε μία από τις δύο κατηγορίες (C_1,C_2), δοθέντος ενός συνόλου προτύπων εκπαίδευσης X το οποίο περιλαμβάνει ζεύγη της μορφής (x_i,C_i) (C_i είναι η κατηγορία του προτύπου x_i). Ο κανόνας ταξινόμησης είναι να αποδοθεί η είσοδος x στην κατηγορία C_1 αν η έξοδος y είναι +1 και στην κατηγορία C_2 αν είναι –1 .



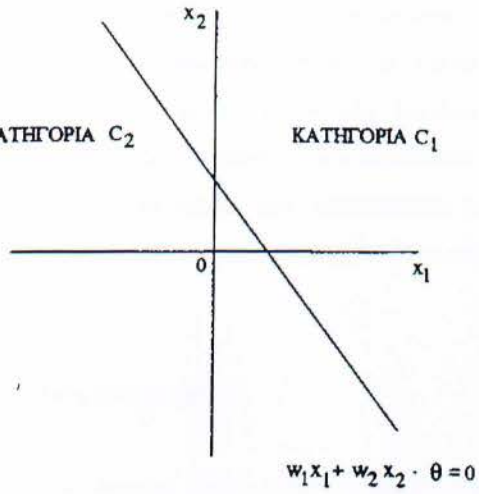
Σχήμα 10 : Γράφημα ροής σημάτων του perceptron

Για την κατανόηση της συμπεριφοράς ενός ταξινομητή προτύπων ,συνηθίζεται η γραφική αναπαράσταση των τελικών περιοχών απόφασης που προκύπτουν. Στην περίπτωση ενός στοιχειώδους perceptron, υπάρχουν δύο περιοχές απόφασης διαχωριζόμενες από ένα υπερεπίπεδο που ορίζεται από την σχέση

$$\sum w_i x_i - \theta = 0 \quad (2)$$

Αυτό απεικονίζεται επομένως στο επόμενο σχήμα για την περίπτωση δύο μεταβλητών εισόδου x_1 και x_2 , όπου το διαχωριστικό όριο παίρνει τη μορφή ευθείας γραμμής . Ένα σημείο (x_1,x_2) το οποίο βρίσκεται άνω της διαχωριστικής ευθείας ανήκει στην κατηγορία C_1, ενώ ένα σημείο(x_1,x_2) το οποίο βρίσκεται κάτω της διαχωριστικής ευθείας ανήκει στην κατηγορία C_2 .Επίσης η επίδραση της πόλωσης θ είναι η παράλληλη μερική μετατόπιση του διαχωριστικού ορίου από την αρχική του θέση. Για την εκπαίδευση του perceptron μπορούμε να

χρησιμοποιήσουμε ένα κανόνα διόρθωσης σφάλματος που είναι γνωστός ως Αλγόριθμος Σύγκλισης του perceptron.



Σχήμα 11: Γραμμική διαχωρισμότητα δυο διαστάσεων, για πρόβλημα δύο κατηγοριων

1.5.3 Συμπέρασμα

Η πρώτη σημαντική κριτική του perceptron του Rosenblatt, παρουσιάστηκε από τους Minsky και Selfridge(1961), οι οποίοι υπογράμμισαν ότι το perceptron δεν μπορεί ούτε καν να υλοποιήσει την συνάρτηση της ισοτιμίας δυο bits (πρόβλημα XOR). Σύμφωνα με τους Minsky και Selfridge(1961) το perceptron του Rosenblatt είναι ανίκανο να αντιμετωπίσει δύσκολα προβλήματα ταξινόμησης . Στο βιβλίο τους υποστηρίζουν ότι οι περιορισμοί που είχαν ανακαλύψει για το perceptron θα έπρεπε λογικά να ισχύουν και για τις πολυεπίπεδες προεκτάσεις του :
(παράθεση από την ενότητα 13.2 του βιβλίου των Minsky και Papert) :Το peceptron μας έχει προσφέρει πολύτιμη γνώση εξαιτίας των πολλών περιορισμών του . Έχει πολλά χαρακτηριστικά που είναι άξιας προσοχής: τη γραμμικότητά του ,το ενδιαφέρον θεώρημα μάθησης ,την απλότητά του ως ένα μοντέλο παράλληλου υπολογισμού .Ωστόσο ,δεν υπάρχει κανένας λόγος να πιστέψουμε ότι κάποιο από αυτά τα προτερήματά του οδηγεί σε μια πολυεπίπεδη έκδοση. Ωστόσο ,θεωρούμε ότι είναι σημαντικό ερευνητικό πρόβλημα η διαφώτιση (ή η απόρριψη)της

διαισθητικής μας κρίσης , ότι η επέκταση σε πολυεπίπεδα συστήματα θα είναι άγονη.

Η παραπάνω εικασία δημιούργησε πολλές απορίες για τις υπολογιστικές ικανότητες όχι μόνο του perceptron , αλλά και των νευρωνικών δικτύων . Εν τούτοις η ιστορία έδειξε ότι η εικασία των Minsky και Papert δεν δικαιώνεται αφού σήμερα έχουμε διάφορους τύπους νευρωνικών δικτύων οι οποίοι είναι υπολογιστικά πολύ πιο ισχυροί από το perceptron του Rosenblatt. Χαρακτηριστικά παραδείγματα ,το πολυεπίπεδο perceptron που εκπαιδεύεται με τον αλγόριθμο ανάστροφης διάδοσης σφάλματος (error backpropagation) και τα δίκτυα ακτινικών συναρτήσεων βάσης (RBF).

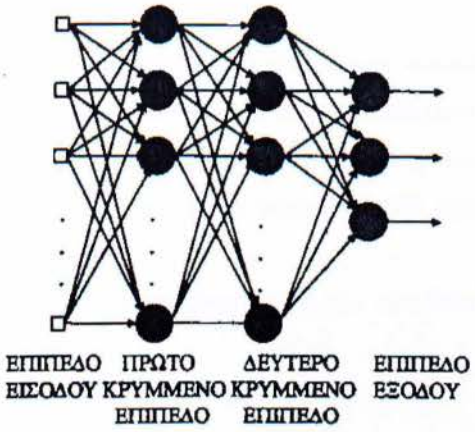
1.6 Το Πολυεπίπεδο Perceptron-MLP

Στο κεφάλαιο αυτό θα περιγράψουμε την ευρύτατα χρησιμοποιούμενη κατηγορία ΤΝΔ που είναι το πολυεπίπεδο perceptron (multiplayer perceptron-MLP). Τα δίκτυα αυτά είναι δίκτυα πρόσθιας τροφοδότησης (feedforward), αποτελούν γενίκευση του μονοστρωματικού perceptron και περιλαμβάνουν ένα επίπεδο εισόδου ,ένα ή περισσότερα κρυμμένα επίπεδα και ένα επίπεδο εξόδου.

Τα MLP έχουν εφαρμοσθεί σε πολλά και δύσκολα προβλήματα και εκπαιδεύονται με επίβλεψη με τον γνωστό αλγόριθμο της ανάστροφης διάδοσης του σφάλματος (error back-propagation). Ο κανόνας αυτός βασίζεται στο κανόνα της διόρθωσης σφάλματος (error correction). Η διαδικασία εκπαίδευσης στον αλγόριθμο back-propagation περιλαμβάνει υπολογισμούς που υλοποιούνται σε δύο περάσματα μέσω των επιπέδων (στρωμάτων) του δικτύου: ένα πέρασμα κατά την ευθεία φορά (από την είσοδο προς την έξοδο) και ένα κατά την ανάστροφη φορά (από την έξοδο προς την είσοδο). Στο ευθύ πέρασμα ,εφαρμόζεται ένα πρότυπο στις εισόδους του δικτύου, πραγματοποιούνται οι υπολογισμοί κατά την ορθή φορά και στο τέλος παράγεται ένα σύνολο από εξόδους που αποτελούν και την πραγματική έξοδο του δικτύου. Κατά την διάρκεια αυτού του περάσματος τα βάρη των συνδέσεων του δικτύου είναι σταθερά. Αντίθετα , κατά την διάρκεια του αντίστροφου περάσματος , τα βάρη προσαρμόζονται σύμφωνα με τον κανόνα διόρθωσης σφάλματος. Συγκεκριμένα ,η πραγματική τιμή εξόδου αφαιρείται από την αντίστοιχη επιθυμητή και παράγεται το σήμα σφάλματος . Αυτό το σήμα στη συνέχεια προωθείται στο δίκτυο κατά την ανάστροφη κατεύθυνση (από την έξοδο

προς την είσοδο –από όπου και το όνομα του αλγορίθμου). Στη διαδικασία αυτή τα βάρη των συνδέσεων αναπροσαρμόζονται ώστε να μετατοπισθεί η απόκριση του δικτύου πλησιέστερα στην επιθυμητή.

Ένα πολυεπίπεδο perceptron (επόμενο σχήμα) έχει τρία χαρακτηριστικά:



Σχήμα 12: Πολυεπίπεδο Perceptron

Κάθε κρυμμένος νευρώνας (νευρώνας σε κρυμμένο επίπεδο)περιέχει μία μη γραμμική συνάρτηση ενεργοποίησης (activation function). Η ύπαρξη της μη-γραμμικότητας είναι πολύ σημαντική και είναι αυτή που προσδίδει στα δίκτυα MLP τις επιθυμητές υπολογιστικές δυνατότητες.

Το δίκτυο περιέχει ένα ή περισσότερα κρυμμένα επίπεδα μη γραμμικών νευρώνων τα οποία δεν ανήκουν στα επίπεδα εισόδου ή εξόδου. Αυτοί οι νευρώνες καθιστούν το δίκτυο ικανό να μάθει πολύπλοκα πρότυπα εξάγοντας από αυτά κάποια ιδιαίτερα χαρακτηριστικά τους.

Δεν υπάρχει σύνδεση μεταξύ των νευρώνων του ίδιου επιπέδου (από τον ορισμό του επιπέδου) και συνήθως υπάρχει πλήρης διασύνδεση μεταξύ των νευρώνων δύο διαδοχικών επιπέδων. Τέλος συνήθως δεν επιτρέπονται συνδέσεις μεταξύ νευρώνων που ανήκουν σε επίπεδα μη διαδοχικά.

Η υπολογιστική ισχύς του πολυεπίπεδου perceptron προέρχεται από το συνδυασμό αυτών των τριών χαρακτηριστικών και από την ικανότητα μάθησης μέσω εκπαίδευσης. Αυτά τα χαρακτηριστικά επίσης ευθύνονται και για τις ελλείψεις που

παρουσιάζουν οι γνώσεις μας γύρω από τη συμπεριφορά αυτών των δικτύων. Η παρουσία κατανεμημένης μορφής μη-γραμμικότητας και η μεγάλη συνδεσιμότητα καθιστούν υπερβολικά δύσκολη τη θεωρητική ανάλυσή τους.

1.6.1 Εισαγωγικές Έννοιες

Στο προηγούμενο σχήμα φαίνεται η αρχιτεκτονική ενός πολυεπίπεδου perceptron με δυο κρυμμένα επίπεδα . Το δίκτυο είναι πλήρως συνδεδεμένο , που σημαίνει ότι ο κάθε νευρώνας ενός επιπέδου συνδέεται με όλους του προηγούμενου επιπέδου. Η ροή των σημάτων (υπολογισμών)γίνεται κατά την ορθή φορά ,από την είσοδο προς την έξοδο ανά επίπεδο.

Στο επόμενο σχήμα παρουσιάζεται ένα τμήμα του ίδιου δικτύου. Σε αυτό διακρίνουμε δυο ειδών σήματα :

Σήματα λειτουργίας. Τα σήματα αυτά εμφανίζονται στην είσοδο του δικτύου, ως ερέθισμα , και προωθούνται κατά την ορθή φορά ανά επίπεδο μέχρι την τελική έξοδο ,όπου παράγονται τα σήματα εξόδου. Αναφερόμαστε σε αυτά με το όνομα «σήματα λειτουργίας» για δύο λόγους. Πρώτον ,υποτίθεται ότι χρησιμοποιούνται κατά την κανονική λειτουργία του δικτύου. Δεύτερον ,κατά το πέρασμά τους από κάθε μονάδα νευρώνων ,υλοποιείται ένας υπολογισμός ως συνάρτηση των εισόδων και των αντίστοιχων βαρών.

Σήματα σφάλματος. Αυτά τα σήματα δημιουργούνται (κατά τη διάρκεια της εκπαίδευσης μόνο) στην έξοδο του δικτύου και προωθούνται κατά την ανάστροφη φορά διαμέσου του δικτύου.

Σχήμα 13 Ροή σημάτων σε ένα τμήμα ενός MLP

Κάθε νευρώνας εξόδου ή κρυμμένος είναι σχεδιασμένος να επιτελεί δύο υπολογισμούς:

Τον υπολογισμό της εξόδου του η οποία εκφράζεται κατά τα γνωστά ως μία συνεχής μη γραμμική συνάρτηση των σημάτων εισόδου και των βαρών των συνδέσεων που σχετίζονται με το συγκεκριμένο νευρώνα (δηλ. υπολογισμός εσωτερικού γινομένου διανύσματος εισόδων με διάνυσμα βαρών και πέρασμα του αποτελέσματος από την μη γραμμική συνάρτηση ενεργοποίησης).
Τον υπολογισμό της εκτίμησης του διανύσματος κλίσης , ο οποίος απαιτείται κατά το αντίστροφο πέρασμα διαμέσου του δικτύου.

1.6.2 Τρόποι εκπαίδευσης

Στην εφαρμογή του αλγορίθμου back-propagation,τον οποίο θα δούμε αναλυτικότερα παρακάτω η εκπαίδευση επιτυγχάνεται με την παρουσίαση στο δίκτυο ενός συνόλου παραδειγμάτων εκπαίδευσης. Η παρουσίαση όλων των προτύπων του συνόλου εκπαίδευσης μία φορά το καθένα ονομάζεται εποχή (epoch) αν η εκπαίδευση του δικτύου γίνεται μέσω της συνάρτησης train ή πέρασμα (pass) αν η εκπαίδευση του δικτύου γίνεται μέσω της συνάρτησης adapt. Η διαδικασία εκπαίδευσης εκτελείται σε επαναλήψεις εποχών έως ότου τα βάρη

του δικτύου σταθεροποιηθούν σε συγκεκριμένες τιμές, για τις οποίες η μέση τιμή του σφάλματος (για το σύνολο προτύπων εκπαίδευσης) συγκλίνει στην ελάχιστη τιμή της. Είναι καλό σε κάθε εποχή να παρουσιάζεται με τυχαία σειρά το σύνολο προτύπων , έτσι ώστε η διαδικασία αναζήτησης στο χώρο των βαρών να είναι περισσότερο στοχαστική.

Για ένα συγκεκριμένο σύνολο εκπαίδευσης, η διαδικασία εκπαίδευσης μπορεί να εκτελεστεί με δύο διαφορετικούς τρόπους:

Εκπαίδευση ανά πρότυπο (on line) ή incremental. Στην περίπτωση αυτή , τα βάρη ενημερώνονται έπειτα από την παρουσίαση κάθε προτύπου στο δίκτυο. Πιο συγκεκριμένα ,θεωρούμαι μια εποχή που αποτελείται από πρότυπα εκπαίδευσης τοποθετημένα στη σειρά (x(1), d(1),.....(x(N), d(N)) (στη γενική περίπτωση τα x(i), d(i) είναι διανύσματα). Το πρώτο πρότυπο (x(1),d(1)) παρουσιάζεται στο δίκτυο και εκτελούνται οι ευθείς και οι αντίστροφοι υπολογισμοί που περιγράψαμε στις προηγούμενες παραγράφους, και οι οποίοι οδηγούν στην μεταβολή των βαρών. Στη συνέχεια παρουσιάζεται το δεύτερο πρότυπο και επαναλαμβάνεται η ίδια διαδικασία που οδηγεί και πάλι στη μεταβολή των βαρών. Η εποχή τελειώνει με την παρουσίαση και του n–οστού προτύπου. Αν $\Delta w_{ji}(n)$ είναι η μεταβολή του βάρους w_{ji} μετά την παρουσίαση του προτύπου n, και $\Delta \hat{w}_{ji}(n)$είναι η μέση τιμή της μεταβολής για όλο το σύνολο προτύπων , τότε έχουμε

$$\Delta \hat{w}_{ji} = \frac{1}{N}\sum_{n=1}^{N} \Delta w_{ji}(n) = -\frac{\eta}{N}\sum_{n=1}^{N} \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = -\frac{\eta}{N}\sum_{n=1}^{N} e_j(n)\frac{\partial e_j(n)}{\partial w_{ji}(n)}$$

Εκπαίδευση ανά εποχή (off-line) ή batch. Στην περίπτωση αυτή τα βάρη ενημερώνονται έπειτα από την παρουσίαση στο δίκτυο ολόκληρο του συνόλου προτύπων της εποχής. Για μία συγκεκριμένη εποχή ορίζουμε τη μέση τιμή των τετραγωνικών σφαλμάτων από τις εξισώσεις (2) και (3) , ως εξής

$$\mathcal{G}_{av} = \frac{1}{2N} \sum_{n=1}^{N} \sum_{j \in C} e_j^2(n)$$

Όπου το σήμα σφάλματος $e_j(n)$ αναφέρεται στο νευρώνα εξόδου j για το πρότυπο εκπαίδευσης n και ορίζεται από την εξίσωση (1). Στην εξίσωση (4.39), το εσωτερικό άθροισμα εφαρμόζεται σε όλους τους νευρώνες του επιπέδου της εξόδου, ενώ το εξωτερικό άθροισμα σε όλα τα πρότυπα της εποχής. Για ρυθμό μάθησης n, η μεταβολή που εφαρμόζεται στο βάρος wji καθορίζεται από τον κανόνα δέλτα.

$$\Delta w_{ji} = -\eta \frac{\partial \mathcal{G}_{av}}{\partial w_{ji}} = -\frac{\eta}{N} \sum_{n=1}^{N} e_j(n) \frac{\partial e_j(n)}{\partial w_{ji}}$$

Για να υπολογίσουμε τη μερική παράγωγο θe_j (n)/ θw_{ji} , εργαζόμαστε όπως και πρίν. Η μεταβολή Δw_{ji} (ενημέρωση των βαρών) εφαρμόζεται μετά το τέλος της εποχής.

Συγκρίνοντας τις προηγούμενες εξισώσεις βλέπουμε ότι για την ίδια μέση τιμή σφάλματος G_{av} έχουμε διαφορετική μεταβολή των βαρών ανάλογα με τον τρόπο εκπαίδευσης. Πράγματι το $\Delta \hat{w}_{ji}$ της on line εκπαίδευσης αποτελεί μια προσέγγιση του Δw_{ji} της off-line.

Η on line είναι αυτή που συνήθως προτιμάται , καθόσον ενισχύει τη στοχαστική πλευρά της εκπαίδευσης και βοηθάει στην αποτροπή της παγίδευσης του αλγορίθμου σε τοπικά ελάχιστα. Όταν όμως τα δεδομένα εκπαίδευσης αποτελούνται από πολλά ίδια σύνολα δεδομένων ,είναι προτιμότερη η χρήση του incremental τρόπου, επειδή στον incremental τρόπο τα συναπτόμενα βάρη του Δικτύου αναπροσαρμόζονται κάθε φορά που κάποιο από τα δεδομένα εκπαίδευσης-training data set-χρησιμοποιείται για την ρύθμιση του Δικτύου και έτσι κερδίζουμε χρόνο , αφού δεν χρειάζεται να περιμένουμε να περάσουν όλα τα training data , που στην ουσία θα δώσουν το ίδιο τελικό αποτέλεσμα.

Από την μεριά της on-line εκπαίδευσης ,προτιμούμαι τον incremental τρόπο εκπαίδευσης ,επειδή απαιτεί λιγότερη χωρητικότητα μνήμης για κάθε συναπτόμενη

ένωση και ευνοεί την στοχαστική εκπαίδευση ,που είναι και αντικειμενικά πιο σωστή ,αφού έτσι αποφεύγεται κατά την εκπαίδευση η απομνημόνευση των δειγμάτων εκπαίδευσης. Κάτι τέτοιο βοηθάει τον αλγόριθμο BKP να μην παγιδεύεται σε τοπικά ελάχιστα Κατά τον ίδιο τρόπο η στοχαστική φύση του incremental τρόπου εκπαίδευσης δυσκολεύει την δημιουργία θεωρητικών συνθηκών-προυποθέσεων σύγκλισης του αλγορίθμου. Σε αντίθεση , η χρήση του batch τρόπου εκπαίδευσης μας δίνει την δυνατότητα να εκτιμήσουμε με μεγάλη ακρίβεια το σφάλμα λάθους; η κλίση σε αυτή την περίπτωση είναι εγγυημένη κάτω από απλές συνθήκες.

Γενικά η off-line εκπαίδευση παρέχει έναν ακριβή υπολογισμό του διανύσματος κλίσης. Εν τέλει, το είδος του προβλήματος είναι αυτό που καθορίζει την καταλληλότερη μορφή εκπαίδευσης.

1.6.3 Κριτήρια τερματισμού

Ο αλγόριθμος back propagation γενικά δε συγκλίνει, ούτε και υπάρχουν καλά ορισμένα κριτήρια για τον τερματισμό της λειτουργίας του. Υπάρχουν, όμως , λογικά κριτήρια που πραγματικά χρησιμοποιούνται για τον τερματισμό της ενημέρωσης των βαρών. Για τη δημιουργία τέτοιων κριτηρίων σκεφτόμαστε ιδιότητες των τοπικών και ολικών ελαχίστων της επιφάνειας σφάλματος. Ας θεωρήσουμε ότι το διάνυσμα βαρών w_* δηλώνει ένα ελάχιστο ,τοπικό ή ολικό. Απαραίτητη συνθήκη για να είναι το w_* ελάχιστο, είναι ότι το διάνυσμα κλίσης g(w), δηλαδή η πρώτη μερική παράγωγος του σφάλματος ως προς το διάνυσμα βαρών w, πρέπει να ισούται με το μηδέν για $w=w_*$. Έτσι, μπορούμε να σχηματίσουμε ένα κριτήριο τερματισμού του αλγορίθμου ως ακολούθως:

Ο αλγόριθμος back-propagation θεωρείται ότι έχει συγκλίνει όταν η ευκλείδεια νόρμα του διανύσματος κλίσης ξεπεράσει ένα ικανοποιητικά μικρό κατώφλι κλίσης.

Το μειονέκτημα αυτού του κριτηρίου είναι ότι, για επιτυχείς συγκλίσεις , ο χρόνος εκπαίδευσης μπορεί να είναι πολύ μεγάλος. Επίσης, απαιτεί τον υπολογισμό του διανύσματος κλίσης g(w).

Μία ακόμα μοναδική ιδιότητα ενός ελαχίστου είναι το γεγονός ότι η συνάρτηση $G_{αν}(w)$ στο σημείο $w=w_*$ είναι στατική. Χρησιμοποιώντας αυτή την παρατήρηση μπορούμε να ορίσουμε ένα διαφορετικό κριτήριο τερματισμού:

Ο αλγόριθμος back-propagation θεωρείται ότι έχει συγκλίνει όταν η απόλυτη τιμή του ρυθμού μεταβολής του σφάλματος ανά εποχή είναι ικανοποιητικά μικρή.

Μια παραλλαγή αυτού του κριτηρίου απαιτεί η μέση τιμή του σφάλματος $G_{αω}(w)$ να γίνει μικρότερη ένα ικανοποιητικά μικρότερο κατώφλι. Επίσης, έχει προταθεί ένα υβριδικό κριτήριο που αποτελείται από το κατώφλι που μόλις αναφέραμε και ένα κατώφλι για την κλίση όπως ορίζεται παρακάτω:

Ο αλγόριθμος back-propagation τερματίζει σε ένα διάνυσμα βαρών w_{final} όταν $||g(w_{final})|| \leq ε$, όπου ε είναι ένα ικανοποιητικά μικρό κατώφλι κλίσης ή όταν $G_{αν}(w_{final}) \leq τ$, όπου τ είναι ένα ικανοποιητικά μικρό κατώφλι για την τιμή του σφάλματος.

Τέλος, ένα ακόμα κριτήριο, τερματισμού είναι το ακόλουθο. Μετά από κάθε επανάληψη εκπαίδευσης, το δίκτυο εξετάζεται για την ικανότητα γενίκευσής του. Η διαδικασία τερματίζει , όταν η δυνατότητα γενίκευσης που απέκτησε είναι ικανοποιητική. Αυτή η μέθοδος τερματισμού που ονομάζεται πρόωρο σταμάτημα (early stopping).

Εφαρμογές του MLP:

Η βασική λειτουργία που επιτελεί ένα πολυεπίπεδο perceptron(MLP) είναι η υλοποίηση απεικόνισης (mapping) από το χώρο των εισόδων στο χώρο των εξόδων ,χρησιμοποιώντας τα ζεύγη εκπαίδευσης και τους κατάλληλους αλγόριθμους εκπαίδευσης (π.χ αλγόριθμος backpropagation).

Μάλιστα ,έχει αποδειχθεί θεωρητικά ότι MLP έχει αυξημένες δυνατότητες απεικόνισης και συγκεκριμένα χαρακτηρίζεται από την ιδιότητα της παγκόσμιας προσέγγισης (universal approximation).Παραλείποντας τον αυστηρό μαθηματικό ορισμό , η ιδιότητα αυτή με απλά λόγια μας λέει το εξής: ένα MLP με τουλάχιστον ένα κρυμμένο επίπεδο με μη γραμμικές μονάδες μπορεί να προσεγγίσει οποιαδήποτε συνάρτηση με οποιαδήποτε ακρίβεια , αυξάνοντας κατάλληλα τον αριθμό των κρυμμένων μονάδων.

Η ιδιότητα αυτή είναι θεωρητικά μόνο σημαντική , διότι μας εξασφαλίζει ότι το MLP μπορεί να υλοποιήσει οποιαδήποτε απεικόνιση ,αλλά δεν είναι πρακτικά χρήσιμη , διότι δεν μας λέει τίποτα για το πώς θα υλοποιήσουμε την απεικόνιση (πόσες κρυμμένες μονάδες πρέπει να βάλουμε , τι αλγόριθμο πρέπει να χρησιμοποιήσουμε κλπ). Το πρόβλημα του καθορισμού του αριθμού των κρυμμένων μονάδων που απαιτούνται για ένα δεδομένο σύνολο εκπαίδευσης αποτελεί σήμερα βασικό ερευνητικό ζήτημα σχετικά με το MLP.

Από τη στιγμή που ένα MLP έχει την ικανότητα να απεικονίζει ένα διάνυσμα πραγματικών εισόδων σε ένα διάνυσμα πραγματικών εξόδων , έχει χρησιμοποιηθεί με ιδιαίτερη επιτυχία για την κατασκευή συστημάτων πρόβλεψης (prediction), για την κατασκευή μοντέλων από δεδομένα (data fitting) , για τον έλεγχο συστημάτων , μέχρι και για την επίλυση μερικών διαφορικών εξιδώσεων.

Η βασικότερη όμως εφαρμογή του MLP είναι σε προβλήματα ταξινόμησης (classification). Στην περίπτωση αυτή , τα δεδομένα είναι της μορφής (πρότυπο ,κατηγορία) και προκειμένου να εκπαιδευτεί το MLP απαιτείται μια διαδικασία κωδικοποίησης των κατηγοριών. Σκοπός της διαδικασίας αυτής είναι η μετατροπή του προβλήματος ταξινόμησης σε προβλήματα απεικόνισης, μέσω της αντιστοίχισης κάθε κατηγορίας σε κάποιο διάνυσμα (ή τιμή)εξόδου. Με τον τρόπο αυτό , το αρχικό σύνολο εκπαίδευσης μετασχηματίζεται ώστε να περιέχει ζεύγη της μορφής (πρότυπο ,διάνυσμα εξόδου) και να μπορεί να χρησιμοποιηθεί το MLP για την υλοποίησης της απεικόνισης.

Ο ευρύτερα χρησιμοποιούμενος τρόπος κωδικοποίησης των κατηγοριών είναι η κωδικοποίηση 1-από-Κ για ένα πρόβλημα κατηγοριών. Στην κωδικοποίηση αυτή , το διάνυσμα εξόδου έχει συνιστώσες $(t_1,\ldots\ldots t_k)$ και η κατηγορία C_k κωδικοποιείται θέτοντας $t_k = 1$

και $t_i=0$ για $i\neq k$. Για παράδειγμα σε ένα πρόβλημα με τρεις κατηγορίες , τα αντίστοιχα τρία διανύσματα εξόδου είναι τα (1,0,0), (0,1,0), (0,0,1) και φυσικά

απαιτείται ένα MLP με τρεις εξόδους . Η ταξινόμηση ενός προτύπου γίνεται εφαρμόζοντας το πρότυπο ως είσοδο στο δίκτυο και επιλέγοντας την κατηγορία που αντιστοιχεί στην έξοδο με την μεγαλύτερη τιμή. Όσο πιο κοντά στο 1 είναι αυτή η έξοδος και κοντά στο μηδέν οι υπόλοιπες έξοδοι , τόσο πιο αξιόπιστη είναι η ταξινόμηση. Ειδικά για την περίπτωση δυο κατηγοριών , χρησιμοποιείται και η κωδικοποίηση με μία έξοδο: αντιστοιχίζουμε την έξοδο t=0 στην μια κατηγορία (C_1) και την έξοδο t=1στην άλλη κατηγορία (C_2). Στην περίπτωση αυτή , η ταξινόμηση ενός προτύπου (αφού έχει γίνει εκπαίδευση) γίνεται ως εξής: αν η έξοδος είναι μεγαλύτερη του 0.5 τότε το πρότυπο ταξινομεί στην κατηγορία C_2, αλλιώς στη κατηγορία C_1.

1.7 Αλγόριθμοι εκπαίδευσης ενός Νευρωνικού Δικτύου MLP

1.7.1 Εκπαίδευση ΤΝΔ

Η έννοια της εκπαίδευσης είναι πολύ ευρεία. Σε γενικές γραμμές, η εκπαίδευση μπορεί να οριστεί ως η κατάλληλη χρήση πληροφοριών για βελτίωση της συμπεριφοράς ενός συστήματος. Στην πιο συνηθισμένη περίπτωση των προβλημάτων απεικόνισης (συσχέτισης προτύπων εισόδου-εξόδου) η εκπαίδευση μπορεί να οριστεί ως η τροποποίηση των παραμέτρων (βαρών) του ΤΝΔ, ώστε χρησιμοποιώντας ένα σύνολο δεδομένων να πλησιάσουμε σταδιακά την επιθυμητή συμπεριφορά συγκρίνοντας την τρέχουσα απόκριση του δικτύου με την επιθυμητή απόκριση.

Όπως έχει ήδη αναφερθεί ένας MLP , είναι ένα είδος αρχιτεκτονικής ενός Νευρωνικού Δικτύου , και σαν τέτοιο για να λειτουργήσει και να αποδώσει τα αναμενόμενα αποτελέσματα θα πρέπει πρώτα να εκπαιδευτεί. Μέχρι τώρα έχουμε γνωρίσει το βασικό χαρακτηριστικό ενός Νευρωνικού Δικτύου , το νευρώνα ,καθώς και την συνάρτηση ενεργοποίησης ,που χρησιμοποιεί. Σε αυτή την παράγραφο θα γνωρίσουμε ,ποίοι τελικά είναι αυτοί οι αλγόριθμοι εκπαίδευσης , χωρίς τους οποίους το Νευρωνικό μας Δίκτυο είναι ένα απλό κουτί. Χρησιμοποιώντας αυτούς τους αλγορίθμους γίνεται δυνατή η αποτελεσματική αναπροσαρμογή των βαρών του MLP και συγχρόνως αυξάνεται το εύρος των εφαρμογών των ΤΝΔ.

Incremental και batch training: Έχουμε την δυνατότητα να χρησιμοποιήσουμε δύο διαφορετικούς τρόπους εκπαίδευσης ενός Δικτύου. Τον incremental ή on-line τρόπο, κατά τον οποίο τα συναπτόμενα βάρη του Δικτύου αναπροσαρμόζονται κάθε φορά που κάποιο από τα δεδομένα εκπαίδευσης-training data set-χρησιμοποιείται για την ρύθμιση του Δικτύου και τον batch ή off-line τρόπο, σύμφωνα με τον οποίο τα συναπτόμενα βάρη του Δικτύου αναπροσαρμόζονται αφού έχουν χρησιμοποιηθεί όλα τα training data set-τα δεδομένα εκπαίδευσης.

1.7.2 Ο backpropagation (BKP) αλγόριθμος

Ο αλγόριθμος εκπαίδευσης πραγματοποιείται σε δυο διαφορετικά βήματα. Πρώτα η παράγωγος της συνάρτησης λάθους που έχει ελαχιστοποιηθεί χρησιμοποιείται για να υπολογιστούν τα βάρη του Νευρωνικού Δικτύου. Αυτή η διαδικασία αποτελεί τη διάδοση του λάθους προς τα πίσω στο Δίκτυο και είναι ο BKP αλγόριθμος. Δηλαδή βάσει του λάθους που έχει προκύψει στην έξοδο του Δικτύου με μια προς τα πίσω διαδικασία ενημερώνονται όλα τα συναπτόμενα βάρη του Δικτύου και ανάλογα προσαρμόζονται , με σκοπό το καινούριο σφάλμα ,που θα προκύψει να είναι ακόμα μικρότερο. Στη συνέχεια αυτές οι παράγωγοι μπορούν να χρησιμοποιηθούν σε συνδυασμό με κάποιους άλλους αλγόριθμους , όπως ο gradient descent (GD), για να ενημερώσουν τα βάρη του Δικτύου. Η όλη διαδικασία της εκπαίδευσης στηρίζεται σε αυτή την epoch-by-epoch βάση και ολοκληρώνεται όταν τα συναπτόμενα βάρη και τα bias level σταθεροποιηθούν σε τέτοιες τιμές ,που το μέσο τετραγωνικό σφάλμα που προκύπτει από το σύνολο των training data ,συγκλίνει σε μια ελάχιστη τιμή.

Ας δούμε λοιπόν αναλυτικότερα τι είναι αυτός ο backpropagation (BKP) αλγόριθμος.

Υποθέτουμε έναν πλήρως συνδεδεμένο MLP , ο οποίος αποτελείται από N_K στρώματα –επίπεδα. Κάθε στρώμα-επίπεδο K αποτελείται από $N_n^{(k)}$ νευρώνες .Το σήμα που εισέρχεται στο κάθε εσωτερικό στρώμα-επίπεδο και η έξοδος του i^{th} νευρώνα στο στρώμα-επίπεδο K συμβολίζονται με $y_i^{(k)}(n)$ και $x_i^{(k)}(n)$ αντίστοιχα και υπολογίζονται σύμφωνα με τις ακόλουθες σχέσεις :

$$y_i^{(k)}(n) = \sum_{i=0}^{N_n^{(k-1)}} w_{ij}^{(k)} \cdot x_j^{(k-1)}(n)$$

$$x_i^{(k)}(n) = f\left(y_i^{(k)}(n)\right)$$

όπου $w_{ij}^{(k)}$ είναι το συναπτόμενο βάρος που συνδέει τον i^{th} νευρώνα του στρώματος- επιπέδου K στο j^{th} νευρώνα του στρώματος-επιπέδου k-1 και η f αντιπροσωπεύει την μη γραμμική συνάρτηση ενεργοποίησης του κάθε νευρώνα.

Η συνάρτηση λάθους ,που πρέπει να ελαχιστοποιηθεί συμβολίζεται με E και αποτελεί την έκφραση του αθροίσματος όλων των συναρτήσεων λάθους των δειγμάτων που έχουν χρησιμοποιηθεί για την εκπαίδευση του Δικτύου:

$$E = \sum_n E_n$$

Η συνάρτηση λάθους E_n θεωρείται διαφορίσιμη και λαμβάνει υπόψη της τις εξόδους του Δικτύου. Χρησιμοποιώντας activation functions που είναι διαφορίσιμες ,όπως sigmoid functions, εξασφαλίζουμε την προεκυψιμότητα της συνάρτησης λάθους E_n με σεβασμό στα βάρη του δικτύου. Χρησιμοποιώντας το κανόνα της αλυσίδας για τα επιμέρους παράγωγα , η παρακάτω εξίσωση μπορεί να γραφεί:

$$\frac{\delta E_n}{\delta w_{ij}^{(k)}} = \frac{\delta E_n}{\delta y_i^{(k)}} \cdot \frac{\delta y_i^{(k)}}{\delta w_{ij}^{(k)}}$$

Σημείωση:

$$\frac{\delta E_n}{\delta w_{ij}^{(k)}} = \delta_i^{(k)} \cdot x_j^{(k-1)}$$

με

$$\delta_i^{(k)} = \frac{\delta E_n}{\delta y_i^{(k)}}$$

Οι συντελεστές $\delta_i^{(k)}$ αναφέρονται ως τοπικά σφάλματα και είναι οι μόνοι παράμετροι που εκτιμώνται στο Δίκτυο με σκοπό να υπολογιστεί ολόκληρο το σύνολο των παραγώγων. Τα τοπικά λάθη ,που προκύπτουν στο στρώμα-επίπεδο εξόδου, μπορούν εύκολα να υπολογιστούν μέσω της παρακάτω φόρμουλας :

$$\delta_i^{(N_K)} = \frac{\delta E_n}{\delta y_i^{(N_K)}} = \frac{\delta E_n}{\delta x_i^{(N_K)}} \cdot f'\left(y_i^{(N_K)}\right)$$

Χρησιμοποιώντας τον κανόνα της αλυσίδας για τα επιμέρους παράγωγα του πρώτου κρυμμένου στρώματος-επιπέδου παίρνουμε:

$$\delta_i^{(k)} = \frac{\delta E_n}{\delta y_i^{(k)}} = \sum_{m=1}^{N_{k+1}} \frac{\delta E_n}{\delta y_m^{(k+1)}} \cdot \frac{\delta y_m^{(k+1)}}{\delta y_i^{(k)}}$$

το οποίο οδηγεί σε:

$$\delta_i^{(k)} = \sum_{m=1}^{N_{k+1}} \delta_m^{(k+1)} \cdot w_{mi}^{(k)} \cdot f'\left(y_i^{(k)}\right)$$

Η εξίσωση $\delta_i^{(N_K)} = \frac{\delta E_n}{\delta y_i^{(N_K)}} = \frac{\delta E_n}{\delta x_i^{(N_K)}} \cdot f'\left(y_i^{(N_K)}\right)$ μας επιτρέπει τον υπολογισμό των παραγώγων λαμβάνοντας υπό όψη όλα τα βάρη του Νευρωνικού Δικτύου ξεκινώντας από το στρώμα-επίπεδο εξόδου και συνεχίζοντας την όλη διαδικασία προς τα πίσω μέσω των κρυμμένων στρωμάτων-επιπέδων.

Ο BKP αλγόριθμος κάνει δυνατή την ελαχιστοποίηση στον υπολογισμό των παραγώγων της συνάρτησης λάθους λαμβάνοντας υπό όψη τα βάρη όλου του Δικτύου. Γνωρίζοντας τις τιμές αυτών των παραγώγων , είναι δυνατό να αναπροσαρμόσουμε τα βάρη χρησιμοποιώντας ένα απλό gradient-descent (GD) αλγόριθμο:

$$w_{ij}^{(k)}(n+1) = w_{ij}^{(k)}(n) - \mu \cdot \delta_i^{(k)}(n) \cdot x_j^{(k-1)}(n)$$

όπου μ είναι το βήμα προσαρμογής.

Στην εφαρμογή του BKP αλγορίθμου, δύο ξεχωριστά στάδια υπολογισμού διακρίνονται: το forward στάδιο και το backward στάδιο. Στο forward στάδιο τα συναπτόμενα βάρη παραμένουν ανεπηρέαστα μέσα από το Δίκτυο και the function signals (that propagate forward through the network) του Δικτύου υπολογίζονται πάνω σε μία neuron-by-neuron βάση. Με άλλα λόγια , η forward φάση υπολογισμού ξεκινάει από το πρώτο κρυμμένο στρώμα-επίπεδο με το διάνυσμα εισόδου και τερματίζει στο στρώμα-επίπεδο εξόδου , υπολογίζοντας το σήμα λάθους για κάθε νευρώνα του στρώματος. Το backward στάδιο, από την άλλη, ξεκινάει από το στρώμα-επίπεδο εξόδου και μεταδίδει το σήμα λάθους προς τα αριστερά-πίσω στρώμα με στρώμα μέσω του Δικτύου, και περιοδικά υπολογίζει την τοπική κλίση για κάθε μεταβολή. Αυτή η περιοδική διαδικασία επιτρέπει στα

συναπτόμενα βάρη του Δικτύου να μείνουν ανεπηρέαστα από αλλαγές σε συνάρτηση με την εξίσωση

$$w_{ij}^{(k)}(n+1) = w_{ij}^{(k)}(n) - \mu \cdot \delta_i^{(k)}(n) \cdot x_j^{(k-1)}(n).$$

Βελτιωμένοι Αλγόριθμοι Εκπαίδευσης ενός MLP

Ο βασικός backpropagation αλγόριθμος, είναι ένας βαθμωτός αλγόριθμος , που βασίζεται στην εκτίμηση του στιγμιαίου τετραγωνικού λάθους ,το οποίο προκύπτει στο κάθε στρώμα-επίπεδο. Η πιο απλή εφαρμογή του BKP αλγόριθμου εκμάθησης αναπροσαρμόζει τα βάρη και τις πολώσεις του Δικτύου προς εκείνη την κατεύθυνση στην οποία η συνάρτηση απόδοσης του Δικτύου μειώνεται πιο γρήγορα-αρνητική κλίση. Μία επανάληψη αυτού του αλγορίθμου μπορεί να εκφραστεί από την παρακάτω εξίσωση (βασισμένη στην εξίσωση (2.43)) (αναπροσαρμογή των βαρών):

$$\Delta W(n) = W(n+1) - W(n) = -\mu \cdot \nabla_w E(n) = -\mu \cdot g(n)$$

(2.44)

όπου W(n) είναι ένα διάνυσμα από τα τρέχοντα βάρη και πολώσεις (δείγμα n), g(n) είναι η τρέχουσα κλίση και μ είναι μια θετική σταθερά ,γνωστή ως ρυθμός μάθησης. Η απόδοση του συγκεκριμένου αλγορίθμου είναι πολύ ευαίσθητη σε συγκεκριμένους ρυθμούς εκπαίδευσης. Για παράδειγμα αν ο ρυθμός εκπαίδευσης είναι πολύ μεγάλος ,ο αλγόριθμος θα γίνει ασταθής ,ενώ αν ο ρυθμός εκπαίδευσης είναι πολύ μικρός ,ο αλγόριθμος θα χρειαστεί πάρα πολύ χρόνο για να αρχίσει να συγκλίνει.

Ένας τέτοιος αλγόριθμος είναι πολύ αργός για τρεις βασικούς λόγους, αλλά δεν αποτελούν αντικείμενο μελέτης της συγκεκριμένης εργασίας

Το μέσο τετραγωνικό σφάλμα, J(W), είναι σχετικά μια πολύπλοκη επιφάνεια στο χώρο των βαρών, πιθανώς με πολλά τοπικά ελάχιστα, επίπεδους τομείς, στενές ακανόνιστες κοιλάδες και ανυψωμένα σημεία. Η πολυπλοκότητα της επιφάνειας λάθους είναι ο βασικός λόγος που η συμπεριφορά της μεθόδου καθόδου του αλγορίθμου ελαχιστοποίησης μπορεί να είναι πολύπλοκη και να έχει ταλαντώσεις γύρω από ένα τοπικό ελάχιστο.

Οι ταχείς αλγόριθμοι εκπαίδευσης του Δικτύου χωρίζονται σε δύο κύριες κατηγορίες. Η πρώτη κατηγορία χρησιμοποιεί heuristic τεχνικές. Οι heuristic τεχνικές ,στις οποίες θα αναφερθούμε είναι: η momentum technique, η adaptive

learning rate backpropagation και η resilient backpropagation. Η δεύτερη κατηγορία χρησιμοποιεί αριθμητικές τεχνικές βελτιστοποίησης . Αναφέρουμε ονομαστικά τρεις τύπους αριθμητικών τεχνικών βελτιστοποίησης για τα Νευρωνικά Δίκτυα: η conjugate gradient, η quasi-Newton and the Levenberg-Marquardt τεχνική.

1.7.3 Heuristic βελτιώσεις του BKP αλγορίθμου

1. Ο όρος ορμής

Ο BKP αλγόριθμος παρέχει μια διαδικασία προσσέγγισης της τροχιάς του χώρου των βαρών που υπολογίζεται από τη μέθοδο της απότομης καθόδου. Όσο πιο μικρός είναι ο ρυθμός μάθησης μ, τόσο πιο μικρές θα είναι οι αλλαγές που θα υπάρχουν στα συναπτόμενα βάρη σε κάθε επανάληψη και πιο ομαλή θα είναι η τροχιά της καμπύλης των βαρών. Όμως, η βελτίωση αυτή επιτυγχάνεται με κόστος ένα πιο αργό ρυθμό μάθησης . Εάν η παράμετρος του ρυθμού μάθησης μ πάλι είναι πολύ μεγάλη με σκοπό να επιταχύνουμε το ρυθμό μάθησης του Δικτύου, οι μεγάλες αλλαγές στα συναπτόμενα βάρη που θα έχουμε ως αποτέλεσμα αυτού του ρυθμού μάθησης , οδηγούν σε τέτοια μορφή που μπορεί να κάνει το Δίκτυο ασταθές (ταλαντευόμενες). Μια απλή μέθοδος αύξησης του ρυθμού εκμάθησης αποφεύγοντας συγχρόνως τον κίνδυνο της αποσταθεροποίησης ,είναι να τροποποιήσουμε τον κανόνα Δέλτα περιλαμβάνωντας τον όρο ορμής .

Μία απλή μέθοδος να αποφύγουμε μια λάθος τροχιά στο χώρο των βαρών που ταλαντεύεται είναι να προσθέσουμε κατά την διαδικασία αναπροσαρμογής των βαρών ένα όρο ορμής (που δηλώνεται με Ω) που είναι ανάλογος της αναπροσαρμογής των βαρών που έγινε στο προηγούμενο βήμα. Ο όρος της ορμής επιτρέπει σε ένα Δίκτυο να ανταποκρίνεται όχι μόνο στις τοπικές κλίσεις ,αλλά και στις τρέχουσες τάσεις στην επιφάνεια λάθους . Λειτουργώντας ως βαθυπερατό φίλτρο , αυτού του είδους η τροποποίηση στην μέθοδο καθόδου είναι ικανή να αγνοήσει μικρά χαρακτηριστικά στην επιφάνεια λάθους . Χωρίς τον όρο της ορμής ένα Δίκτυο μπορεί να κολλήσει σε ένα ρηχό τοπικό ελάχιστο ,κάτι βέβαια που δεν είναι επιθυμητό.

$$\Delta W(n) = -\mu \cdot g(n) + \Omega \cdot \Delta W(n-1)$$

(2)

Προσαρμοστικός ρυθμός μάθησης -Adaptive learning rate

Ένας προαρμοστικός ρυθμός μάθησης κατά την διάρκεια της διαδικασίας εκπαίδευσης θα επιχειρήσει να διατηρήσει το μέγεθος του βήκματος εκμάθησης τόσο μεγάλο όσο είναι δυνατό , ώστε συγχρόνως η μάθηση –εκμάθηση να είναι σταθερή (δηλαδή να γίνεται με σταθερό ρυθμό). Μια κλασσική στρατηγική ,για να πετύχουμε το παραπάνω, βασίζεται στην απεικόνιση του ρυθμού αλλαγής του μέσου τετραγωνικού σφάλματος και μπορεί να περιγραφεί όπως φαίνεται παρακάτω:

- Εάν το μέσο τετραγωνικό σφάλμα J μειώνεται σταθερά , ώστε η κλίση του μέσου τετραγωνικού σφάλματος ∇J να είναι αρνητική για ένα προκαθορισμένο αριθμό βημάτων, τότε ο ρυθμός μάθησης αυξάνεται γραμμικά:

$$\mu(n+1) = \mu(n) + \alpha, \quad \alpha > 0$$

(3)

- Εάν το μέσο τετραγωνικό σφάλμα αυξηθεί ($\nabla J > 0$), τότε ο ρυθμός μάθησης μειώνεται εκθετικά:

$$\mu(n+1) = \beta \cdot \mu(n), \qquad 0 < \beta < 1$$

(4)

Resilient backpropagation

Τα πολυεπίπεδα νευρωνικά δίκτυα συνήθως χρησιμοποιούν sigmoid ως συνάρτηση μεταφοράς -transfer function στα κρυμμένα στρώματα. Οι ημιτονοειδείς συναρτήσεις χαρακτηρίζονται από το γεγονός ότι η κλίση τους πλησιάζει πιο πολύ το μηδέν όσο τα δεδομένα εισόδου μεγαλώνουν σε πλήθος. Αυτό μας δημιουργεί πρόβλημα όταν χρησιμοποιούμε την μέθοδο καθόδου, από τη στιγμή που η βαθμιδωτή μεταβολή σαν μέγεθος ,μπορεί να έχει ένα πολύ μικρό πλάτος και επομένως να προκαλεί μικρές αλλαγές στα βάρη και τις πολώσεις . Ο στόχος του ελαστικού -resilient backpropagation αλγορίθμου είναι να εξαφανίσει τις επιδράσεις του πλάτους στα παράγωγα. Μόνο το πρόσημο των παραγώγων

χρησιμοποιείται για να προσδιορίσει την κατεύθυνση της αναπροσαρμογής των βαρών, το πλάτος των παραγώγων δεν επιδρά καθόλου στην αναπροσαρμογή των βαρών .Η αναπροσαρμοσμένη τιμή κάθε βάρους και πόλωσης του δικτύου αυξάνεται από ένα παράγοντα γ κάθε φορά που η παράγωγος της συνάρτηση απόδοσης, λαμβάνοντας υπόψη τα βάρη έχει το ίδιο πρόσημο για δύο πετυχημένα δείγματα. Η αναπροσαρμοσμένη τιμή μειώνεται κατά ένα παράγοντα γ κάθε φορά που η παράγωγος , λαμβάνοντας υπόψη τα βάρη, αλλάζει πρόσημο σε σχέση με το προηγούμενο δείγμα Εάν η παράγωγος είναι μηδέν , δεν υπάρχουν αλλαγές στις αναπροσαρμοσμένες τιμές . Κάθε φορά που τα βάρη ταλαντεύονται η αλλαγή που θα επέλθει σε αυτά θα μειώνεται. Εάν τα βάρη συνεχίζουν να αλλάζουν προς την ίδια κατεύθυνση για πολλές επαναλήψεις ,τότε το πλάτος των αλλαγών που θα επέλθουν στα βάρη θα αυξηθεί. Γενικά ο resilient backpropagation αλγόριθμος συγκλίνει πολύ γρηγορότερα από τους προηγούμενους αλγορίθμους .

1.7.4 Σύνολα εκπαίδευσης και ελέγχου

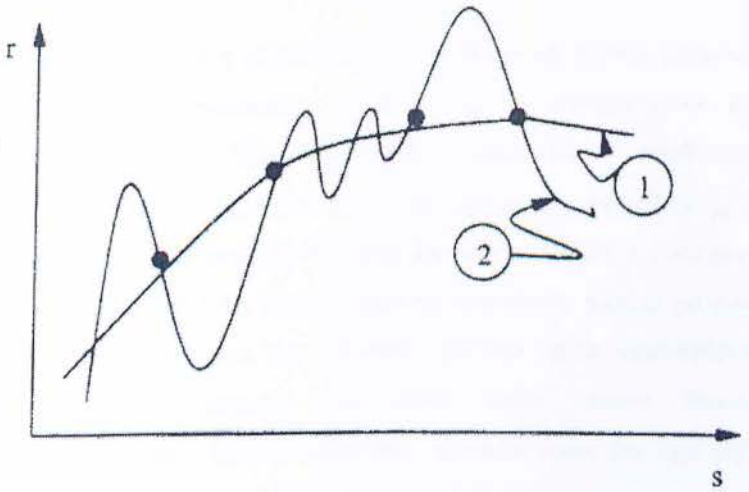
Υποθέστε ότι έχουμε ένα σύνολο δεδομένων απεικόνισης εισόδου/εξόδου ή μονάχα εισόδου τα οποία καθορίζουν την επιθυμητή συμπεριφορά του δικτύου. Το σύνολο αυτό (έστω Η) το ονομάζουμε σύνολο εκπαίδευσης. Στη μάθηση με επίβλεψη (supervised learning) το Η μας παρέχει πληροφορίες για τη συσχέτιση των εισόδων (Ι) με τις εξόδους (Ο), δηλαδή αποτελείται από ζεύγη της μορφής ($I_ι$, $O_ι$) , ι = 1 ,.....n (στη γενική περίπτωση τα I_i και O_i είναι διανύσματα). Επίσης ,πρέπει να διευκρινίσουμε ότι το σύνολο αυτό περιέχει ένα μικρό γενικά αριθμό ζευγών σε σχέση με το σύνολο των (ενδεχομένων άπειρων) πιθανών ζευγών.

Το σύνολο ελέγχου Η΄ αποτελείται επίσης από ζεύγη της μορφής (I_i,O_i). Μεταξύ των δύο συνόλων όμως δεν υπάρχει κοινή τομή. Το σύνολο αυτό χρησιμοποιείται μετά το τέλος της εκπαίδευσης για να διαπιστωθεί η ικανότητα γενίκευσης του ΤΝΔ σε δεδομένα με τα οποία δεν έχει ήδη εκπαιδευτεί.

Στη μάθηση χωρίς επίβλεψη τα δεδομένα του Η δεν είναι απεικονίσεις εισόδου/ εξόδου αλλά μόνο δεδομένα εισόδου $I_ι$. Σε αυτή την περίπτωση το δίκτυο πρέπει να καταλήξει στην εξαγωγή κάποιων βασικών ιδιοτήτων των δεδομένων του Η(π.χ εύρεση ομάδων).

1.7.5 Γενίκευση

Στην περίπτωση της μάθησης με επίβλεψη, μετά το τέλος της εκπαίδευσης , το ΤΝΔ για κάθε είσοδο θα πρέπει να παρέχει την αντίστοιχη επιθυμητή έξοδο. Το ερώτημα που προκύπτει είναι κατά πόσο αυτό επιτυγχάνεται και για εισόδους διαφορετικές από αυτές του συνόλου εκπαίδευσης. Αυτός είναι ο στόχος της γενίκευσης. Υποθέστε ότι έχουμε ένα ΤΝΔ που επιτελεί απεικόνιση μιας εισόδου σε μία έξοδο. Έχουμε 4 ζευγάρια της μορφής (χ , f(x)) για το σύνολο Η. Όπως φαίνεται και στο Σχήμα , η γενίκευση μετά την εκπαίδευση μπορεί να έχει διάφορες μορφές. Παρότι δεν υπάρχει λάθος στην εκπαίδευση , είναι δυνατόν η συνάρτηση απεικόνισης να έχει πολλές διαφορετικές μορφές. Δύο από αυτές απεικονίζονται στο επόμενο σχήμα . Οι περισσότεροι από εμάς πιθανώς να προτιμήσουν την ομαλή καμπύλη, αλλά παρόλα ταύτα το σύνολο εκπαίδευσης από μόνο του είναι πιθανό να μας οδηγήσει σε οποιαδήποτε από τις λύσεις. Σε τέτοιες περιπτώσεις , είναι απαραίτητο το σύνολο ελέγχου που θα εκτιμήσει την ικανότητα γενίκευσης και θα μας βοηθήσει στο να επιλέξουμε το καλύτερο μοντέλο.



Σχήμα 14 : Καμπύλες γενίκευσης

1.8 Διαδικασία εκμάθησης ενός Νευρωνικού Δικτύου

Τι εννοούμε με τον όρο εκμάθηση ενός Νευρωνικού Δικτύου; Ο όρος εκμάθησης στην ουσία σε αυτή την παράγραφο σημαίνει εκπαίδευση. Εκπαίδευση του Δικτύου. Όπως ήδη έχουμε προαναφέρει τα Νευρωνικά Δίκτυα έχουν την ικανότητα να εκπαιδεύονται και να μαθαίνουν από το περιβάλλον τους ,ακριβώς όπως ο ανθρώπινος εγκέφαλος, δια της επαναλήψεως.

"Learning is a process by which the free parameters of a neural network are adapted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place".
Δηλαδή: Εκπαίδευση ενός Δικτύου είναι η διαδικασία εκείνη κατά την οποία οι ελεύθερες παράμετροι του Νευρωνικού Δικτύου προσαρμόζονται στο περιβάλλον που το Δίκτυο λειτουργεί, μέσω της διαδικασίας της προσομοίωσης. Το είδος της εκπαίδευσης καθορίζεται από τον τρόπο με τον οποίο η αλλαγή των παραμέτρων λαμβάνει χώρα.

Πώς γίνεται όμως αυτή η εκπαίδευση; Τα Νευρωνικά Δίκτυα μαθαίνουν μέσω μιας αλληλεπιδρώσας διαδικασίας προσαρμογής των συναπτόμενων βαρών και των επιπέδων κλίσης- bias levels. Με απλά λόγια αποθηκεύουν και απομνημονεύουν κάποια δεδομένα κατά την διάρκεια της εκπαίδευσης και όσο η εκπαίδευση διαρκεί αυτορυθμίζονται έτσι –δηλαδή ρυθμίζουν τις πολώσεις και τα συναπτόμενα βάρη- ώστε να είναι σε θέση να διακρίνουν κάποια χαρακτηριστικά και κάποια μεγέθη ανάλογα την εργασία , για την οποία εκπαιδεύονται. Έτσι ,συγκεκριμένα στην εργασία μας ,όταν έρθει κάποιο διαμορφωμένο τηλεπικοινωνιακό σήμα προς αναγνώριση , το οποίο όμως δεν έχει λάβει μέρος στην διαδικασία εκπαίδευσης, αλλά διατηρεί κάποια παρόμοια χαρακτηριστικά με αυτά ,που είχαν τα σήματα εκπαίδευσης, τότε το Νευρωνικό Δίκτυο βάσει της όλης διαδικασίας εκμάθησης είναι σε θέση να αναγνωρίσει την διαμόρφωση αυτού του σήματος. Η όλη παραπάνω διαδικασία ονομάζεται γενίκευση-generalization.

Η Γενίκευση στηρίζεται σε τρεις βασικές παραμέτρους :

Τη διαδικασία εκπαίδευσης (τον αριθμό των δεδομένων εκπαίδευσης-training examples και στην έκταση ,στην οποία αυτά αντιπροσωπεύουν τα υπό ταξινόμηση σύνολα),
Τη δομή του Δικτύου (αριθμός κρυμμένων στρωμάτων-επιπέδων και αριθμός νευρώνων σε κάθε στρώμα-επίπεδο),
Τη πολυπλοκότητα του προς επίλυση προβλήματος.

Όσο αφορά το περιεχόμενο των δύο άλλων παραγόντων, το θέμα της γενίκευσης μπορεί να μελετηθεί κάτω από δύο διαφορετικές προοπτικές :

Η αρχιτεκτονική του Δικτύου να είναι δεδομένη (ελπίζουμε αυτή η δομή να είναι σε συμφωνία με την πολυπλοκότητα του υπό επίλυση προβλήματος) και αυτό που θα πρέπει να προσδιορίσουμε να είναι το μέγεθος-αριθμός των training data set-δεδομένων εκπαίδευσης που απαιτούνται για να πετύχουμε μια καλή γενίκευση.
το μέγεθος-αριθμός των training data set-δεδομένων εκπαίδευσης να είναι δεδομένο και αυτό που θα πρέπει να προσδιορίσουμε να είναι η καλύτερη αρχιτεκτονική του Δικτύου ,για να πετύχουμε μια καλή γενίκευση.

Στη πράξη φαίνεται ότι αυτό που χρειαζόμαστε για να πετύχουμε τελικά μια καλή γενίκευση είναι το μέγεθος-αριθμός των training data set-δεδομένων εκπαίδευσης , N, που ικανοποιεί τα δεδομένα του προβλήματος:

$$N = O\left(\frac{W}{\varepsilon}\right)$$

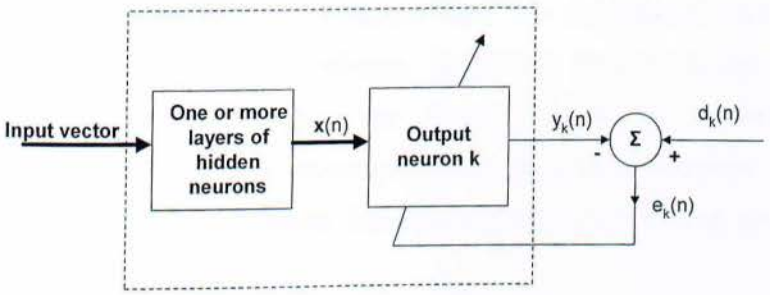
όπου W είναι ο συνολικός αριθμός των ελεύθερων παραμέτρων (οι οποίες είναι τα συναπτόμενα βάρη-synaptic weights και οι biases) του Δικτύου, ε δηλώνει ένα μέρος του λάθους ταξινόμησης κλάσμα που επιτρέπεται στα δεδομένα ελέγχου και $O(\cdot)$ δηλώνει τη σειρά της ποσότητας που εμπεριέχεται σε αυτές . Για παράδειγμα, για ένα λάθος 10 τοις εκατό ο αριθμός των δεδομένων εκπαίδευσης θα πρέπει να είναι 10 φορές ο αριθμός των ελεύθερων παραμέτρων του Δικτύου.
Η επιλογή της αρχιτεκτονικής του Δικτύου επιδρά στην διαδικασία εκπαίδευσης. Αυτό συμβαίνει λόγω της μεγάλης ελαστικότητας και του μεγάλου αριθμού

ελεύθερων παραμέτρων που έχουν τα νευρωνικά δίκτυα με τρια στρώματα, τα οποία αργότερα απαιτούν μεγάλη χρονικά εκπαίδευση προκειμένου να συγκλίνουν. Η επίδραση του αριθμού των νευρώνων του Δικτύου στην γενίκευση και η επίδραση του αριθμού των δεδομένων εκπαίδευσης είναι στενά συνδεδεμένες. Εάν το Δίκτυο είναι υπερμέγεθες , τα δεδομένα εκπαίδευσης θα απομνημονευτούν στο Δίκτυο και η διαδικασία της γενίκευσης δεν θα είναι δυνατή. Παρόλα αυτά , αν ο αριθμός των νευρώνων αυξηθεί, ο αριθμός των δεδομένων εκπαίδευσης θα αυξηθεί επίσης. Για αυτό είναι σημαντικό να διατηρούμε το μέγεθος του Δικτύου όσο το δυνατό σε χαμηλά επίπεδα με σκοπό να μειώσουμε το υψηλό κόστος εκπομπής που εισάγει ο μεγάλος αριθμός δεδομένων εκπαίδευσης.

Όλη η παραπάνω εκπαίδευση γίνεται με τους αλγόριθμους εκπαίδευσης – ένα σύνολο από καλά προσδιορισμένους κανόνες επίλυσης ενός προβλήματος. Οι αλγόριθμοι «εκπαίδευσης» διαφέρουν ο ένας από τον άλλο συνήθως στον τρόπο με τον οποίο γίνεται η προσαρμογή των συναπτόμενων βαρών των νευρώνων- δηλαδή σταδιακά ή όλη μαζί . Μία άλλη παράμετρος διαφοροποίησης των αλγορίθμων, όπως ήδη έχουμε δεί, είναι ο τρόπος με τον οποίο ένα Νευρωνικό Δίκτυο έχει κατασκευαστεί.

1.8.1 Error correction learning

Ας θεωρήσουμε την απλή περίπτωση , όπου ένας νευρώνας K αποτελεί και το μοναδικό υπολογιστικό τμήμα του τελευταίου στρώματος-επιπέδου(output layer) ενός τροφοδοτούμενου(feedforward) Νευρωνικού Δικτύου, όπως απεικονίζεται και στο παρακάτω σχήμα :



Σχήμα 15. Error-correction learning

Το υπό μελέτη σήμα μας εισέρχεται στο Δίκτυο μέσω των νευρώνων του στρώματος εισόδου –input layer. Στη συνέχεια το επεξεργασμένο πια από το πρώτα αυτό στρώμα νευρώνων περνάει από μια σειρά κρυφών στρωμάτων νευρώνων και τελικά φτάνει στο νευρώνα K ως σήμα – διάνυσμα x(n) . Η παράμετρος n σημαίνει το χρονικό βήμα από μια επαναλαμβανόμενη διαδικασία προσαρμογής των συναπτόμενων βαρών του νευρώνα K Το τελικό μας σήμα , που είναι και το σήμα εξόδου του νευρώνα K συμβολίζεται με $y_k(n)$. Αυτό το σήμα εξόδου, που αντιπροσωπεύει και την μοναδική έξοδο του Δικτύου μας, συγκρίνεται με την επιθυμητή έξοδο-με το σήμα δηλαδή που θα θέλαμε να έχουμε, που την συμβολίζουμε με $d_k(n)$. Για παράδειγμα αν εμείς δώσουμε ως είσοδο στο Δίκτυο ένα DSBSC σήμα , θα επιθυμούμε στην έξοδο το Δίκτυο να αναγνωρίσει ότι το σήμα αυτό ήταν διαμορφωμένο κατά DSBSC. Αν η διαδικασία εκπαίδευσης όμως δεν είναι η σωστή ή αν κατά την επεξεργασία του σήματος από το Δίκτυο υπάρξει κάποιο σφάλμα, τότε το Δίκτυο δεν θα δώσει την επιθυμητή έξοδο, αλλά κάποια άλλη. Αυτή η άλλη έξοδος συγκρίνεται με την επιθυμητή και προκύπτει το σφάλμα του Δικτύου , που συμβολίζεται με $e_k(n)$. Έτσι θα ισχύει:

$$e_k(n) = d_k(n) - y_k(n)$$

Το σφάλμα του Δικτύου ενεργοποιεί ένα μηχανισμό ελέγχου ,σκοπός του οποίου είναι να εφαρμόσει μια σειρά από διορθώσεις στα συναπτόμενα βάρη του νευρώνα Κ Ο μηχανισμός ελέγχου έχει σχεδιαστεί έτσι ,ώστε οι σταδιακές διορθώσεις να βελτιώνουν την έξοδο του Δικτύου, μέχρι αυτή να πλησιάσει την επιθυμητή έξοδο. Ο στόχος αυτός πετυχαίνεται ελαχιστοποιώντας την συνάρτηση, E(n), η οποία ορίζεται ως:

$$E(n)=\frac{1}{2}e_k^2(n)$$

Όπου E(n) είναι η στιγμιαία τιμή του ενεργειακού περιεχομένου της συνάρτησης λάθους. Οι σταδιακές διορθώσεις στα συναπτόμενα βάρη του νευρώνα Κ είναι συνέχεις και διαρκείς μέχρι τα συναπτόμενα βάρη σταθεροποιηθούν –και τα συναπτόμενα βάρη θα σταθεροποιηθούν όταν η έξοδος του δικτύου είναι η επιθυμητή. Στο σημείο αυτό ολοκληρώνεται και η διαδικασία της εκμάθησης.

Η διαδικασία μάθησης που περιγράφηκε εδώ είναι γνωστή ως error-correction μάθηση. Συγκεκριμένα, η ελαχιστοποίηση της συνάρτησης E(n) οδηγεί σε ένα κανόνα μάθησης γνωστό ως κανόνα δέλτα των Widrow-Hoff . Έστω ότι το $w_{kj}(n)$ δηλώνει την τιμή του συναπτόμενου βάρους w_{kj} του νευρώνα k ,που προκύπτει από το στοιχείο $x_j(n)$ του διανύσματος του σήματος x(n) την χρονική στιγμή n. Σύμφωνα με το κανόνα δέλτα, η αναπροσαρμογή $\Delta w_{kj}(n)$ που εφαρμόζεται στο συναπτόμενο βάρος w_{kj} τη χρονική στιγμή n ορίζεται ως :

$$\Delta w_{kj}(n)=\mu e_k(n)x_j(n)$$

(1)

όπου μ είναι η παράμετρος του ρυθμού μάθησης (a είναι μια θετική σταθερά που προσδιορίζει το ρυθμό μάθησης καθώς αυτή εξελίσσεται από το ένα στάδιο στο άλλο). Με απλά λόγια , ο κανόνας δέλτα μπορεί να οριστεί ως :

"Η προσαρμογή που κάνουμε στο συναπτόμενο βάρος ενός νευρώνα είναι ανάλογο του γινομένου του λάθους και του σήματος εισόδου στην είσοδο του νευρώνα".

Έχοντας υπολογίσει τις προσαρμογές $\Delta w_{kj}(n)$ που απαιτούνται στις συνάψεις του νευρώνα ,η αναπροσαρμοσμένη τιμή του συναπτόμενου βάρους w_{kj} ορίζεται από την εξίσωση

$$w_{kj}(n+1)=w_{kj}(n)+\Delta w_{kj}(n)$$

(2)

Συνεπώς , $w_{kj}(n)$ και $w_{kj}(n+1)$ μπορούν να θεωρηθούν ως η παλιά και η καινούρια τιμή του συναπτόμενου βάρους w_{kj}, αντίστοιχα.

Στην πράξη ,η παράμετρος του ρυθμού μάθησης μ παίζει ένα σημαντικό ρόλο στο προσδιορισμό της απόδοσης της διαδικασίας μάθησης error-correction learning και η επιλογή του μ επιδρά επίσης πλήρως στην ακρίβεια του αλγορίθμου μάθησης . Για αυτό απαιτείται προσοχή στην επιλογή του μ ,ώστε να εξασφαλίσουμε ότι η σταθερότητα ή η σύγκλιση της επαναληπτικής διαδικασίας μάθησης θα πετύχει.

1.8.2 Memory – based learning

Στη memory-based μάθηση, όλες (ή οι περισσότερες) από τις προηγούμενες εμπειρίες είναι αποθηκευμένες σε μια μεγάλης χωρητικότητας μνήμη από σωστά ταξινομημένα εισόδου –εξόδου παραδείγματα: $\{(\mathbf{x}_i,d_i)\}_{i=1}^{N}$, όπου x_i δηλώνει ένα διάνυσμα εισόδου και d_i δηλώνει το αντίστοιχο επιθυμητό διάνυσμα απόκρισης. Χωρίς απώλεια της γενικότητας περιορίζουμε την επιθυμητή απόκριση του Δικτύου σε αριθμό και όχι διάνυσμα. Για παράδειγμα σε ένα δυαδικό πρόβλημα ταξινόμησης υπάρχουν δύο κλάσεις ,οι οποίες συμβολίζονται με C_1 και C_2. ΣΕ αυτή την περίπτωση η επιθυμητή έξοδος ,d_i ,παίρνει την τιμή 0 (ή –1) για την κλάση C_1 και την τιμή 1 για την κλάση C_2. Όταν θελήσουμε να ταξινομήσουμε ένα σήμα –διάνυσμα x_{test} (το οποίο δεν υπήρχε κατά την εκπαίδευση), τότε ο αλγόριθμος ανταποκρίνεται με το να αναλύει και να διορθώνει τα δεδομένα της εκπαίδευσης σε μια «τοπική γειτονιά» του x_{test}.

Όλοι οι memory-based learning αλγόριθμοι έχουν δυο βασικά χαρακτηριστικά:

Χρησιμοποιούνται διάφορα κριτήρια για να ορίσουν την τοπική γειτονιά του διανύσματος x_{test}.

Ο κανόνας εκμάθησης εφαρμόζεται στα παραδείγματα εκπαίδευσης σε μια τοπική γειτονιά του x_{test}.

Ανάλογα τα δυο παραπάνω χαρακτηριστικά έχουμε και διαφορετικούς αλγορίθμους.

Σε μία απλή μορφή της μάθηση γνωστή ως ο κανόνας της κοντινής γειτονιάς , η «τοπική γειτονιά» ορίζεται ως το δεδομένο εκείνο, που προέκυψε από την

διαδικασία εκμάθησης και βρίσκεται στην άμεση «γειτονιά» του διανύσματος x_{test}.Συγκεκριμένα , το διάνυσμα

$$\mathbf{x}_N' \in \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$$

είναι ο πιο κοντινός γείτονας του x_{test} εάν

$$\min_i d(\mathbf{x}_i, \mathbf{x}_{test}) = d(\mathbf{x}_N', \mathbf{x}_{test})$$

όπου $d(x_i, x_{test})$ είναι η Ευκλείδεια απόσταση μεταξύ των διανυσμάτων x_i και x_{test}. Η τάξη που σχετίζεται με την ελάχιστη απόσταση ,η οποία είναι το διάνυσμα x_N', αναφέρεται ως τάξη του x_{test}. Αυτός ο κανόνας είναι ανεξάρτητος από την υποκείμενη διανομή, η οποία είναι υπεύθυνη για την δημιουργία των παραδειγμάτων εκπαίδευσης.

1.8.3 Hebbian learning

Για να μορφοποιήσουμε την μάθηση του Hebbian σε μαθηματικούς όρους ,θεωρώντας ένα συναπτόμενο βάρος w_{kj} του νευρώνα k ,όπου τα προσυναπτόμενα και μετασυναπτόμενα σήματα δηλώνονται με x_j και y_k αντίστοιχα. Η προσαρμογή που εφαρμόζεται στο συναπτόμενο βάρος w_{kj} τη χρονική στιγμή n εκφράζεται από το γενικό τύπο

$$\Delta w_{kj}(n) = F(y_k(n), x_j(n))$$

(2.3)

όπου $F(\cdot,\cdot)$ είναι μια συνάρτηση τόσο των προσυναπτόμενων όσο και των μετασυναπτόμενων σημάτων. Τα σήματα $x_j(n)$ και $y_k(n)$ είναι συνήθως αδιάστατα. Ακολουθούν δυο διαφορετικές μορφές της παραπάνω εξίσωσης.

α) Hebb's hypothesis

Η πιο απλή μορφή του αλγορίθμου εκμάθησης Hebbian περιγράφεται μαθηματικά από τον ακόλουθο τύπο:

$$\Delta w_{kj}(n) = \mu y_k(n) x_j(n),$$

όπου μ είναι μια σταθερά, η οποία καθορίζει τον ρυθμό εκμάθησης. Η παραπάνω εξίσωση τονίζει καθαρά την συσχετιζόμενη φύση της σύναψης του Hebbian. Η επαναλαμβανόμενη εφαρμογή του σήματος εισόδου (προσυναπτόμενη

δραστηριότητα) x_j οδηγεί σε μια αύξηση του y_k και επομένως εκθετική αύξηση που τελικά οδηγεί την συναπτόμενη σύνδεση σε κορεσμό(saturation). Σε αυτό το σημείο καμία πληροφορία δεν θα αποθηκευτεί στις συνάψεις και η επιλεκτικότητα θα χαθεί.

β) Covariance hypothesis

Ένας τρόπος για να ξεπεράσουμε τον περιορισμό της Hebb's hypothesis είναι να χρησιμοποιήσουμε την covariance hypothesis. Σε αυτή την υπόθεση τα προ –και – μετά συναπτόμενα σήματα της προηγούμενης εξίσωσης αντικαθίστανται από τμήματα των προ-και-μετά συναπτόμενων σημάτων από τις μέσες τιμές τους σε ένα χρονικό διάστημα .Σε αυτή την περίπτωση ,τα προ –και-μετα συναπτόμενα σήματα της προηγούμενης εξίσωσης αντικαθίστανται από τις μέσες τιμές των επιμέρους προ-και-μετα συναπτόμενων σημάτων ,ενός συγκεκριμένου χρονικού διαστήματος.

Ας υποθέσουμε ότι $\bar{x}$ και $\bar{y}$ είναι οι μέσοι όροι των προ συναπτόμενου σήματος x_j και του μετά συναπτόμενου σήματος y_k, αντίστοιχα. Σύμφωνα με την covariance hypothesis, προσαρμογή των συναπτόμενων βαρών w_{kj} σημαίνει σε μαθηματικούς όρους:

$$\Delta w_{kj}(n) = \mu\left(x_j - \bar{x}\right)\left(y_k - \bar{y}\right)$$

όπου μ είναι η παράμετρος του ρυθμού εκμάθησης . Αυτές οι μέσες τιμές αποτελούν για τα προ –και- μετά συναπτόμενα βάρη κατώφλια ,που προσδιορίζουν το πρόσημο της συναπτόμενης διαμόρφωσης.

Και στις δυο πάντως περιπτώσεις, Hebb's hypothesis και covariance hypothesis, η εξάρτηση του Δw_{kj} από το y_k είναι γραμμική;
Από την παραπάνω εξίσωση προκύπτουν οι επόμενες παρατηρήσεις:

Το συναπτόμενο βάρος w_{kj} επαυξάνεται εάν υπάρχουν επαρκή επίπεδα προ-και μετάσυναπτόμενης δραστηριότητας, κάτι που σημαίνει ότι πρέπει να

ικανοποιούνται οι συνθήκες : $x_j > \bar{x}$ και $y_k > \bar{y}$ συγχρόνως . Το συναπτόμενο βάρος w_{kj} καταπιέζεται εάν υπάρχει είτε

Μια προσυναπτόμενη δραστηριότητα (π.χ $x_j > \bar{x}$) απουσία επαρκούς μετασυναπτόμενης δραστηριότητας (π.χ $y_k < \bar{y}$) , είτε

Μια μετασυναπτόμενη δραστηριότητα (π.χ $y_k > \bar{y}$) απουσία επαρκούς προσυναπτόμενης δρστηριότητας (π.χ $x_j < \bar{x}$)

1.8.4 Competitive learning

Στην competitive εκμάθηση οι νευρώνες του τελευταίου στρώματος ενός Νευρωνικού Δικτύου συναγωνίζονται μεταξύ τους ,για να είναι στο τέλος ένας από αυτούς ενεργός και αυτός , ο οποίος θα δώσει το τελικό σήμα. Έτσι ενώ στην εκμάθηση μπορούν συγχρόνως να είναι πολλοί νευρώνες ενεργοί , στην competitive εκμάθηση μόνο ένας νευρώνας μπορεί να είναι ενεργός ανά πάσα στιγμή.

Υπάρχουν τρία βασικά στοιχεία σε ένα competitive learning αλγόριθμο:

Ένα σύνολο από νευρώνες που είναι όλοι οι ίδιοι εκτός από μερικά τυχαία διανεμημένα συναπτόμενα βάρη και τα οποία για αυτό το λόγο ανταποκρίνονται διαφορετικά σε ένα δεδομένο σύνολο από δείγματα.

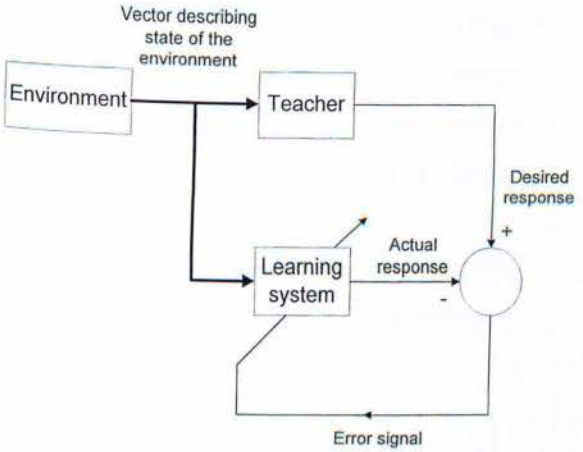
Ένα όριο που επιβάλλεται στην ''δύναμη'' του κάθε νευρώνα.

Ένα μηχανισμό που επιτρέπει στους νευρώνες να συναγωνίζονται για το δικαίωμα να ανταποκρίνονται σε ένα δεδομένο υποσύνολο από διανύσματα εισόδου, έτσι ώστε μόνο ένα νευρώνας εξόδου ή ένας νευρώνας ανά ομάδα ,να είναι ενεργός κάθε στιγμή. Ο νευρώνας νικητής ονομάζεται "ο νικητής τα παίρνει όλα " νευρώνας .

Αντιστοίχως οι ατομικοί νευρώνες του Δικτύου μαθαίνουν να ειδικεύονται σε ensembles παρόμοιων δειγμάτων; κατά αυτό τον τρόπο γίνονται μελλοντικοί detectors για διαφορετικές τάξεις δειγμάτων εισόδου.

1.8.5 Επιβλεπόμενη μάθηση

Ένα βασικό συστατικό της επιβλέπουσας μάθησης είναι η διαθεσιμότητα ενός εξωτερικού ''δασκάλου'', όπως εικονίζεται στο παρακάτω σχήμα. Το "δάσκαλο'' μπορούμε να τον φανταστούμε σαν κάποιον που έχει γνώσεις για το περιβάλλον, το οποίο αντιπροσωπεύεται από ένα σύνολο από παραδείγματα εισόδου-εξόδου. Το περιβάλλον αυτό όμως είναι άγνωστο στο νευρωνικό δίκτυο που μας ενδιαφέρει. Ας υποθέσουμε λοιπόν ότι τόσο το δίκτυο όσο και ο "δάσκαλος" εκπαιδεύονται και οι δύο από ένα διάνυσμα εκπαίδευσης ,το οποίο προέρχεται από αυτό το άγνωστο περιβάλλον. Ο "δάσκαλος" είναι τότε σε θέση να εφοδιάσει το νευρωνικό δίκτυο με μια επιθυμητή απόκριση για το συγκεκριμένο διάνυσμα εκπαίδευσης . Αυτή η επιθυμητή απόκριση αντιπροσωπεύει την απόκριση που θα είχε το νευρωνικό μας δίκτυο ,αν λειτουργούσε βέλτιστα. Οι παράμετροι του δικτύου προσαρμόζονται κάτω από την συνδυασμένη επίδραση τόσο του διανύσματος εκπαίδευσης ,όσο και του σήματος λάθους . Σαν σήμα λάθους ορίζουμε την διαφορά της επιθυμητής απόκρισης του δικτύου και της πραγματικής απόκρισης . Αυτή η προσαρμογή γίνεται σταδιακά και επαναληπτικά μέχρι να φτάσουμε όσο πιο κοντά γίνεται στην επιθυμητή έξοδο. Η γνώση του περιβάλλοντος που πλέον είναι διαθέσιμη στο "δάσκαλο", μεταβιβάζεται στο νευρωνικό δίκτυο μέσω της πλήρους εκπαίδευσης . Όταν πλέον το δίκτυο έχει εκπαιδευτεί αρκετά να αναγνωρίζει τις πληροφορίες του περιβάλλοντος ο "δάσκαλος" απομακρύνεται και αφήνει το δίκτυο να επικοινωνεί μόνο του με το περιβάλλον.



Σχήμα 16. Learning with a teacher

Η μορφή της επιβλεπόμενης μάθησης που περιγράφηκε παραπάνω είναι η error-correction μάθηση ,που συζητήσαμε προηγουμένως . Είναι ένα κλειστό πισωτροφοδοτούμενο σύστημα, όπου το άγνωστο περιβάλλον δεν περιλαμβάνεται στο βρόγχο . Σαν μέτρο απόδοσης του συστήματος μπορούμε να θεωρήσουμε το μέσο-τετραγωνικό σφάλμα ή το άθροισμα των τετραγωνικών σφαλμάτων των δειγμάτων εκπαίδευσης, και το ορίζουμε σαν μια συνάρτηση των ελεύθερων παραμέτρων του συστήματος . Αυτή η συνάρτηση μπορεί να θεωρηθεί σαν πολυδιάστατη λάθους-απόδοσης (error-performance) επιφάνεια με τις ελεύθερες παραμέτρους σαν ζεύγος συντεταγμένων. Η πραγματική επιφάνεια λάθους είναι κατά μέσο όρο πάνω από όλα τα πιθανά δείγματα εισόδου-εξόδου. Κάθε δοσμένη λειτουργία του συστήματος υπό την επίβλεψη του ''δασκάλου'' αντιπροσωπεύεται από ένα σημείο πάνω στην επιφάνεια λάθους. Για να βελτιώσουμε την απόδοση του συστήματος με το πέρασμα του χρόνου και κατά συνέπεια να μάθει το σύστημα από το ''δάσκαλο'', θα πρέπει το σημείο λειτουργίας να μετακινηθεί επιτυχώς προς τα κάτω προς ένα ελάχιστο σημείο της επιφάνειας λάθους; Το ελάχιστο σημείο μπορεί να είναι ένα τοπικό ή ένα ολικό ελάχιστο. Ένα σύστημα που δουλεύει με την μέθοδο της επιβλέπουσας μάθησης είναι σε θέση να επιτελέσει την παραπάνω διαδικασία χρησιμοποιώντας τις πληροφορίες που έχει για την μεταβολή της επιφάνειας λάθους και η οποία ανταποκρίνεται στην τρέχουσα συμπεριφορά του συστήματος .Η κλίση μιας επιφάνειας λάθους κάθε χρονική στιγμή είναι ένα διάνυσμα, το οποίο δείχνει στην κατεύθυνση , που ορίζει κάθε στιγμή η μέθοδος καθόδου. Στην πραγματικότητα, στην περίπτωση της

επιβλέπουσας μάθησης μέσω παραδειγμάτων, το σύστημα μπορεί να χρησιμοποιήσει ένα στιγμιαίο υπολογισμό του διανύσματος μεταβολής, υποθέτοντας ότι οι δείκτες των παραδειγμάτων είναι αυτοί της συγκεκριμένης χρονικής στιγμής . Παρόλα αυτά δεδομένου ενός αλγορίθμου ο οποίος ελαχιστοποιεί την συνάρτηση κόστους ,επαρκούς συνόλου δειγμάτων εισόδου-εξόδου και αρκετό διαθέσιμο χρόνο για εκπαίδευση , ένα σύστημα επιβλέπουσας μάθησης είναι συνήθως ικανό να εκτελέσει εργασίες ,όπως ταξινόμηση δειγμάτων και προσέγγιση συναρτήσεων.

1.9 Συμπεράσματα

1.9.1 Μάθηση και Γενίκευση

Στην περίπτωση που χρησιμοποιούμε τα νευρωνικά δίκτυα ταξινόμησης (MLP, RBF), υπάρχει το πρακτικό πρόβλημα της επιλογής του αριθμού M των κρυμμένων μονάδων που πρέπει να χρησιμοποιηθούν. Ο στόχος της εκπαίδευσης δεν είναι η ακριβής μάθηση ολόκληρου του συνόλου εκπαίδευσης , αλλά η εκπαίδευση του μοντέλου στην ταξινόμηση νέων δεδομένων , δηλαδή η ικανότητα γενίκευσης.

Χαρακτηριστικό παράδειγμα αποτελούν τα δίκτυα (MLP). Είναι γνωστό ότι , χρησιμοποιώντας μεγάλο αριθμό κρυμμένων μονάδων , μπορούν να μάθουν πλήρως όλα τα δεδομένα ενός συνόλου εκπαίδευσης. Ωστόσο, ένας τέτοιος ταξινομητής έχει γενικά κακές ικανότητες γενίκευσης, διότι στην ουσία « απομνημονεύει» τα δεδομένα εκπαίδευσης και δεν παρουσιάζει καλές επιδόσεις σε νέα δεδομένα. Από την άλλη πλευρά , ένα δίκτυο MLP με πολύ λίγες κρυμμένες μονάδες δεν έχει αρκετή ευελιξία ώστε να μπορεί να ορίσει πολύπλοκες περιοχές απόφασης. Βλέπουμε λοιπόν ότι γενικά υπάρχει ένας βέλτιστος αριθμός παραμέτρων (βαρών και πολώσεων) ενός δικτύου, για τον οποίο το εκπαιδευμένο μοντέλο χαρακτηρίζεται από τις καλύτερες επιδόσεις γενίκευσης.

Υπάρχει, λοιπόν , η ανάγκη τεχνικών εύρεσης της βέλτιστης πολυπλοκότητας ενός δικτύου για την οποία προκύπτει η καλύτερη γενικευτική ικανότητα. Συγκεκριμένα, ισχύει το δίλημμα πόλωσης – μεταβλητότητας (Bias – variance – dilemma) , σύμφωνα με το οποίο το σφάλμα γενίκευσης ενός ταξινομητή μπορεί να γραφτεί σαν το άθροισμα δύο παραγόντων: της πόλωσης και της μεταβλητότητας. Ένα μοντέλο που είναι πολύ απλό έχει πολύ μεγάλη πόλωση , ενώ ένα μοντέλο με πολλές παραμέτρους έχει πολύ μεγάλη μεταβλητότητα. Η καλύτερη γενίκευση προκύπτει όταν έχουμε το βέλτιστο συνδυασμό τιμών των δύο παραπάνω ποσοτήτων.

Ουσιαστικά, το δίλημμα πόλωσης –μεταβλητότητας μας λέει ότι , στην περίπτωση που προσπαθούμε να ελαττώσουμε την πόλωση ενός ταξινομητή , δηλαδή να απομνημονεύσουμε ολόκληρο το σύνολο εκπαίδευσης χρησιμοποιώντας ένα πολύ ευέλικτο μοντέλο (με μεγάλο M), το τίμημα που

πληρώνουμε είναι ότι αυξάνουμε τη μεταβλητότητα και κατά συνέπεια μειώνουμε την ικανότητα γενίκευσης του μοντέλου σε άγνωστα δεδομένα .

1.9.2 Δομική προσέγγιση

Για να πετύχουμε τη βέλτιστη ισορροπία μεταξύ πόλωσης και μεταβλητότητας, υπάρχουν δύο προσεγγίσεις που μπορούμε να ακολουθήσουμε. Η πρώτη είναι η δομική (structural) προσέγγιση κατά την οποία ξεκινάμε από ένα μικρό δίκτυο (λίγες παράμετροι) , το οποίο βαθμιαία μεγαλώνουμε (αυξάνουμε τον αριθμό των κρυμμένων μονάδων M) μέχρι να πετύχουμε βέλτιστες επιδόσεις γενίκευσης. Συγκεκριμένα θα παρατηρήσουμε ότι υπάρχει κάποια τιμή του M πέρα από την οποία το σφάλμα γενίκευσης αρχίζει να αυξάνεται. Το φαινόμενο αυτό ονομάζεται υπερεκπαίδευση (overtraining) του δικτύου και σημαίνει ότι ο ταξινομητής έχει πολύ μεγάλο αριθμό παραμέτρων και έχει στην ουσία απομνημονεύσει τα δεδομένα , μειώνοντας έτσι τη γενικευτική του ικανότητα. Από τη στιγμή που παρατηρούμε ελάττωση της γενικευτικής ικανότητας σταματάμε να αυξάνουμε τον αριθμό των παραμέτρων και θεωρούμε ότι φτάσαμε σε βέλτιστο μοντέλο.

Η παραπάνω διαδικασία μπορεί να εφαρμοστεί και με αντίστροφη φορά : εκπαιδεύουμε αρχικά ένα μοντέλο με μεγάλο αριθμό παραμέτρων. Το μοντέλο αυτό λόγω της μεγάλης μεταβλητότητας έχει μικρή πόλωση και μικρή γενικευτική ικανότητα. Σταδιακά μειώνουμε τον αριθμό των κρυμμένων μονάδων , οπότε μειώνεται το σφάλμα γενίκευσης, μέχρι να φτάσουμε σε κάποιο μέγεθος δικτύου πέρα από το οποίο το σφάλμα γενίκευσης αρχίζει να αυξάνει, οπότε σταματάμε την διαδικασία.

1.9.3 Κανονικοποίηση

Η άλλη προσέγγιση για την εύρεση του δικτύου με τον βέλτιστο αριθμό παραμέτρων βασίζεται στην έννοια της κανονικοποίησης
(regularization) . Ο πιο απλός τρόπος για να επιτύχουμε κανονικοποίηση βασίζεται στην προσθήκη ενός όρου τιμωρίας (penalty term) στην συνάρτηση μέσου τετραγωνικού σφάλματος , που ελαχιστοποιούμε κατά την εκπαίδευση του δικτύου. Ο όρος κανονικοποίησης στην ουσία εμποδίζει τις παραμέτρους να λάβουν υψηλές τιμές κατά την εκπαίδευση και οδηγεί τις τιμές των βαρών που δεν

έχουν μεγάλη σημασία να πάρουν την τιμή μηδέν , δηλαδή να αφαιρεθούν από το δίκτυο. Με τον τρόπο αυτό, στο τέλος της εκπαίδευσης προκύπτουν δίκτυα με λιγότερες παραμέτρους από ότι στην αρχή της εκπαίδευσης και φυσικά με καλύτερες δυνατότητες γενίκευσης. Αν P_i είναι οι παράμετροι του δικτύου(π.χ βάρη και πολώσεις στην περίπτωση του MLP) ένας όρος κανονικοποίησης που χρησιμοποιείται συχνά είναι το άθροισμα των τετραγώνων των τιμών των παραμέτρων $\Sigma_i p_i^2$, οπότε η ποσότητα που ελαχιστοποιείται κατά την εκπαίδευση είναι:

$$E^I = E + \lambda \Sigma_i p_i^2$$

Όπου το Ε είναι η γνωστή συνάρτηση που ορίζει το τετραγωνικό σφάλμα εκπαίδευσης. Η παράμετρος λ καθορίζει το σχετικό βάρος των δύο στόχων της εκπαίδευσης : της ελαχιστοποίησης του Ε και της επίτευξης μικρών τιμών των παραμέτρων του μοντέλου . Φυσικά στην περίπτωση αυτή οι παράγωγοι τροποποιούνται με την προσθήκη του όρου $2\lambda p_i$. Η μέθοδος αυτή, στην περίπτωση εκπαίδευσης του MLP ονομάζεται φθορά των βαρών (weight decay) και , σε πολλές περιπτώσεις, έχει οδηγήσει σε δίκτυα MLP με πολύ καλές επιδόσεις γενίκευσης.

Θα πρέπει να σημειωθεί ότι υπάρχει μεγάλη συσχέτιση μεταξύ του αριθμού των διαθέσιμων προτύπων εκπαίδευσης και του βέλτιστου μεγέθους του μοντέλου. Γενικά, λέμε ότι το μοντέλο είναι μικρό ή μεγάλο πάντα σε σχέση με τον αριθμό των προτύπων εκπαίδευσης που έχουμε στη διάθεσή μας. Αν έχουμε λίγα πρότυπα η χρήση πολύπλοκου μοντέλου θα οδηγήσει σε υπερεκπαίδευση. Αντίθετα αν ο αριθμός των προτύπων εκπαίδευσης είναι μεγάλος μπορούμε συνήθως να χρησιμοποιήσουμε μεγαλύτερα μοντέλα.

1.9.4 Ένα κριτήριο τερματισμού της εκπαίδευσης

Ένα βασικό ζήτημα το οποίο δεν διευκρινίστηκε στην παραπάνω περιγραφή σχετίζεται με τον τρόπο που υπολογίζουμε το σφάλμα γενίκευσης ενός εκπαιδευμένου μοντέλου. Σύμφωνα με την τεχνική που ακολουθείται συνήθως για την αξιολόγηση του ταξινομητή, χρησιμοποιούμε ένα ανεξάρτητο σύνολο προτύπων , που το ονομάζουμε σύνολο επικύρωσης (validation set), και , με βάση τον αριθμό των σφαλμάτων ταξινόμησης στο σύνολο αυτό , υπολογίζουμε το σφάλμα ταξινόμησης του μοντέλου.

Επομένως, αν μας δίνεται ένας αριθμός ταξινομητών που έχουν εκπαιδευτεί με βάση το ίδιο σύνολο εκπαίδευσης, για να επιλέξουμε τον καλύτερο υπολογίζουμε το σφάλμα ταξινόμησης καθενός για τα δεδομένα του συνόλου επικύρωσης και επιλέγουμε αυτόν με το μικρότερο σφάλμα επικύρωσης. Συνήθως, στην περίπτωση αυτή, για τον υπολογισμό του σφάλματος γενίκευσης του ταξινομητή που τελικά επιλέξαμε χρησιμοποιείται και ένα τρίτο σύνολο δεδομένων που ονομάζεται σύνολο ελέγχου (test set).

Η παραπάνω τεχνική μπορεί να χρησιμοποιηθεί και κατά την εκπαίδευση ενός ταξινομητή (π.χ ενός MLP) με ελαχιστοποίηση του τετραγωνικού σφάλματος ή της ποσότητας E^{l}. Στην περίπτωση αυτή, ο αλγόριθμος εκπαίδευσης λειτουργεί ενημερώνοντας τις παραμέτρους του δικτύου στην κατεύθυνση ελαχιστοποίησης του σφάλματος, αλλά ταυτόχρονα (π.χ κάθε 10 βήματα) υπολογίζουμε το σφάλμα επικύρωσης που αντιστοιχεί στις τιμές των παραμέτρων που έχουν υπολογιστεί στο συγκεκριμένο βήμα. Γενικά, όσο προχωρεί η εκπαίδευση , τόσο μειώνεται το σφάλμα εκπαίδευσης και μειώνεται και το σφάλμα επικύρωσης. Υπάρχει , όμως, συνήθως ένα όριο πέρα από το οποίο περαιτέρω μείωση του σφάλματος εκπαίδευσης οδηγεί σε αύξηση του σφάλματος επικύρωσης, διότι αρχίζει να εμφανίζεται το φαινόμενο της υπερεκπαίδευσης. Στο σημείο αυτό μπορούμε να σταματήσουμε την εκπαίδευση του μοντέλου. Η τεχνική αυτή που ονομάζεται πρόωρο σταμάτημα (early stopping) χρησιμοποιείται πολύ συχνά και παρέχει ένα πολύ πιο αποδοτικό κριτήριο τερματισμού σε σχέση με τον τερματισμό σε τοπικό ελάχιστο του σφάλματος εκπαίδευσης.

1.9.5 Τεχνικές εκτίμησης σφάλματος ταξινόμησης

Από τη στιγμή που, όπως αναφέραμε προηγουμένως , η αξιολόγηση ενός ταξινομητή θα πρέπει να γίνεται με βάση την επίδοσή του στην ταξινόμηση αγνώστων δεδομένων, η στρατηγική που χρησιμοποιούμε για την εκτίμηση του σφάλματος ταξινόμησης κάποιας μεθόδου διαιρεί το σύνολο των διαθέσιμων προτύπων σε δύο τμήματα: στο σύνολο εκπαίδευσης (training set) που χρησιμοποιείται για την κατασκευή του ταξινομητή και στο σύνολο ελέγχου (test set)που χρησιμοποιείται για τον υπολογισμό του σφάλματος γενίκευσης. Οι τεχνικές εκτίμησης σφάλματος διαφέρουν μεταξύ τους κυρίως στον τρόπο που γίνεται η διάσπαση των δεδομένων στα δύο σύνολα. Θα πρέπει να σημειωθεί ότι οι

τεχνικές αυτές δεν μπορούν να χρησιμοποιηθούν για την αξιολόγηση ενός συγκεκριμένου ταξινομητή, διότι, όπως γίνεται φανερό στη συνέχεια , βασίζονται στην κατασκευή πολλών ταξινομητών για την εκτίμηση σφάλματος. Οι λόγοι για τους οποίους χρησιμοποιούνται είναι : α) για τη σύγκριση διαφορετικών τεχνικών , π.χ RBF και MLP, β) για τη μελέτη της επίδρασης των διαφόρων χαρακτηριστικών εισόδου στο σφάλμα ταξινόμησης , π.χ σε ένα πρόβλημα με τέσσερα χαρακτηριστικά , πώς μεταβάλλεται το σφάλμα ταξινόμησης αν χρησιμοποιήσουμε μόνο το πρώτο και το τρίτο χαρακτηριστικό για την ταξινόμηση, γ) για την μελέτη της επίδρασης του αριθμού των παραμέτρων στο σφάλμα γενίκευσης, όταν χρησιμοποιείται συγκεκριμένο μοντέλο , π.χ επίδραση του αριθμού των κρυμμένων μονάδων ενός δικτύου MLP κ. τ. λ

Οι κυριότερες τεχνικές εκτίμησης σφάλματος είναι οι ακόλουθες:

Holdout: Καθορίζουμε το ποσοστό των προτύπων εκπαίδευσης και ελέγχου. Δημιουργούμε τυχαία τα σύνολα εκπαίδευσης και ελέγχου με βάση τα παραπάνω ποσοστά και κατασκευάζουμε τον ταξινομητή χρησιμοποιώντας το σύνολο εκπαίδευσης. Στη συνέχεια υπολογίζουμε το σφάλμα ταξινόμησης χρησιμοποιώντας το σύνολο ελέγχου. Επαναλαμβάνουμε αρκετές φορές την παραπάνω διαδικασία (δημιουργία συνόλων εκπαίδευσης και ελέγχου, εκπαίδευση του ταξινομητή και υπολογισμός του σφάλματος ταξινόμησης) και η τελική εκτίμηση σφάλματος προκύπτει ως ο μέσος όρος των επιμέρους σφαλμάτων ταξινόμησης που υπολογίσαμε.

Leave-one-out: Για κάθε διαθέσιμο πρότυπο κατασκευάζουμε έναν ταξινομητή . Η κατασκευή γίνεται θεωρώντας ως σύνολο εκπαίδευσης ολόκληρο το σύνολο δεδομένων εκτός από το συγκεκριμένο πρότυπο. Στη συνέχεια ελέγχουμε αν ο ταξινομητής που προκύπτει ταξινομεί σωστά το πρότυπο που αγνοήθηκε κατά την εκπαίδευση. Το σενάριο αυτό επαναλαμβάνεται για όλα τα διαθέσιμα πρότυπα και μετράμε το ποσοστό των προτύπων που ταξινομήθηκαν λάθος.

Rotation: Διαιρούμε το σύνολο προτύπων σε G υποσύνολα. Για καθένα υποσύνολο I (I= 1, ...G), κατασκευάζουμε έναν ταξινομητή θεωρώντας ως σύνολο εκπαίδευσης τα πρότυπα των υπολοίπων G-1 υποσυνόλων και υπολογίζουμε το ποσοστό e_i των σφαλμάτων ταξινόμησης χρησιμοποιώντας ως σύνολο ελέγχου τα πρότυπα του υποσυνόλου i. Επαναλαμβάνοντας τη διαδικασία G φορές (μια για κάθε υποσύνολο i) υπολογίζουμε το τεχνικό σφάλμα εκτίμησης ως το μέσο όρο των επιμέρους σφαλμάτων e_i.

Bootstrap: Έστω ότι το σύνολο δεδομένων αποτελείται από N πρότυπα. Τα σύνολα εκπαίδευσης και ελέγχου κατασκευάζονται ως εξής: επιλέγουμε με τυχαίο τρόπο N ακέραιους αριθμούς μεταξύ 1 και N. Μερικοί αριθμοί επιλέγονται περισσότερες από μία φορές, ενώ άλλοι δεν επιλέγονται καθόλου. Τα πρότυπα x^i που αντιστοιχούν στους αριθμούς I που δεν επιλέχτηκαν αποτελούν το σύνολο ελέγχου και τα υπόλοιπα το σύνολο εκπαίδευσης . Στη συνέχεια κατασκευάζουμε τον αντίστοιχο ταξινομητή και υπολογίζουμε το σφάλμα ταξινόμησης. Επαναλαμβάνουμε την παραπάνω διαδικασία αρκετές φορές και υπολογίζουμε το τελικό σφάλμα ταξινόμησης ως το μέσο όρο των επιμέρους σφαλμάτων.

1.10 Ανακεφαλαιώνοντας

α) Δίκτυα ακτινικών συναρτήσεων βάσης

Μια άλλη σημαντική κατηγορία τεχνητών νευρωνικών δικτύων πρόσθιας τροφοδότησης (feed-forward) είναι τα δίκτυα ακτινικών συναρτήσεων βάσης (radial basis function networks), τα οποία για συντομία ονομάζονται δίκτυα RBF.

Τα δίκτυα RBF έχουν τις ρίζες τους στη μαθηματική θεωρία της προσέγγισης συναρτήσεων και στην ουσία υλοποιούν μια συνάρτηση παρεμβολής (interpolation function), η οποία προσεγγίζει την τιμή μιας συνάρτησης σε κάποιο σημείο ως το μέσο όρο των τιμών της συνάρτησης σε κοντινά σημεία.

Τα δίκτυα RBF είναι νευρωνικά δίκτυα πρόσθιας τροφοδότησης (feedforward) με ένα κρυμμένο επίπεδο , του οποίου οι μονάδες j (j=1,…,M) υπολογίζουν μια ειδική συνάρτηση $h_j(\bar{x})$ του διανύσματος εισόδου.

β) Ανταγωνιστική μάθηση

Εκτός από τα προβλήματα μάθησης με επίβλεψη (π.χ ταξινόμηση) έχουν αναπτυχθεί σημαντικά μοντέλα νευρωνικών δικτύων κατάλληλα για μάθηση χωρίς επίβλεψη.

γ) Μάθηση χωρίς επίβλεψη

Στην περίπτωση της μάθησης χωρίς επίβλεψη έχουμε στη διάθεσή μας μόνο το σύνολο των δεδομένων εισόδου $\bar{x}^i$, χωρίς να υπάρχει διαθέσιμο κάποιο επιθυμητό διάνυσμα εξόδου $\bar{y}^i$, όπως συμβαίνει στην περίπτωση της μάθησης με επίβλεψη. Σκοπός της μάθησης δεν είναι πλέον η υλοποίηση κάποιας απεικόνισης από τα δεδομένα εισόδου στα δεδομένα εξόδου, αλλά η αυτοοργάνωση και η ανακάλυψη διαφόρων χαρακτηριστικών ιδιοτήτων των δεδομένων $\bar{x}^i$. Χαρακτηριστικά προβλήματα που εμπίπτουν στην κατηγορία της μάθησης χωρίς επίβλεψη είναι :

Ομαδοποίηση (clustering)

Ανάλυση βασικών συνιστωσών (Principal Component Analysis)

Μείωση της διάστασης των δεδομένων (προβολή τους σε χώρο μικρότερης διάστασης από τον αρχικό) (Dimensionality reducion).

ΚΕΦΑΛΑΙΟ – 2 SUPPORT VECTOR MACHINES

Τις Μηχανές Διανυσμάτων Υποστήριξης ή αλλιώς τα SVMs, που πρωτάρχισαν από τον Vapnik και τους συνεργάτες του το 1963, μπορούμε να τα δούμε ως ένα καινούργιο τρόπο εκπαίδευσης των feedforward νευρωνικών δικτύων των οποίων οι γνωστοί αλγόριθμοι εκπαίδευσης (όπως Radial Basis Functions) απορρέουν ως ειδικές περιπτώσεις. Συγκεκριμένα, ένα Support Vector Machine χρησιμοποιεί μη-γραμμικό μετασχηματισμό του χώρου εισόδου σε ένα πολυδιάστατο χώρο χαρακτηριστικών (feature space), στον οποίο κατασκευάζεται ένα βέλτιστο υπερεπίπεδο (hyperplane) που επιχειρεί να διαχωρίσει δύο κλάσεις. Εφόσον το βέλτιστο υπερεπίπεδο ανταποκρίνεται σε μια μη-γραμμική επιφάνεια στο χώρο εισόδου, έπεται ότι η μέθοδος, άμεσα, κατασκευάζει μια βέλτιστη διαχωριστική επιφάνεια στο χώρο εισόδου που επιχειρεί να διαχωρίσει δύο κλάσεις.

Ένα SVM είναι μια γραμμική μηχανή με μερικές πολύ καλές ιδιότητες. Όπως έχω αναφέρει πιο πάνω, η κύρια ιδέα ενός SVM είναι να δημιουργήσει ένα υπερεπίπεδο (hyperplane) ως την επιφάνεια απόφασης με τέτοιο τρόπο ώστε το περιθώριο ανάμεσα στα θετικά και αρνητικά παραδείγματα να είναι το μέγιστο. Η μηχανή επιτυγχάνει αυτό την επιθυμητή ιδιότητα, ακολουθώντας μια αρχή που έχει τις βάσεις της στην θεωρία στατιστικής μάθησης (statistical learning theory). Πιο συγκεκριμένα, τα SVMs είναι μια κατά προσέγγιση υλοποίηση της μεθόδου structural risk minimization. Αυτή η συνεπαγωγή βασίζεται στο γεγονός ότι το ποσοστό λάθους μιας μηχανής εκμάθησης στα δεδομένα ελέγχου (δηλαδή το generalization error rate) περικλείεται από το άθροισμα του ποσοστού λάθους εκπαίδευσης και ενός όρου που βασίζεται στο Vapnik- Chervonenkis (VC) dimension. Στην περίπτωση όπου έχουμε διαχωρίσιμα πρότυπα, το SVM παράγει μηδενική τιμή για τον πρώτο όρο και ελαχιστοποιεί το δεύτερο όρο. Επομένως, ένα SVM μπορεί να παράξει καλά αποτελέσματα γενίκευσης σε προβλήματα pattern classification παρά το γεγονός ότι δεν έχει ενσωματωμένη γνώση για το πρόβλημα. Αυτό το χαρακτηριστικό είναι μοναδικό στα support vector machines.

Η κεντρική ιδέα στην κατασκευή ενός αλγορίθμου SVM είναι το εσωτερικό γινόμενο μεταξύ του του "support vector" xi και του vector x από την είσοδο. Τα support vectors αποτελούνται από ένα μικρό υποσύνολο των δεδομένων

εκπαίδευσης. Εξαρτώμενοι από το πώς αυτό το εσωτερικό γινόμενο παράχθηκε, μπορούμε να κατασκευάσουμε διαφορετικές μηχανές εκμάθησης που χαρακτηρίζονται από δικές τους μη-γραμμικές επιφάνειες απόφασης. Πιο συγκεκριμένα, μπορούμε να χρησιμοποιήσουμε τον αλγόριθμο εκμάθησης των support vectors για να δημιουργήσουμε τους ακόλουθους τρεις τύπους μηχανών εκμάθησης (μεταξύ άλλων): Polynomial learning machines, Radial-basis function networks, Two-layer perceptrons (δηλαδή με ένα hidden layer). Δηλαδή, για κάθε ένα από αυτά τα feedforward δίκτυα μπορούμε να χρησιμοποιήσουμε τον αλγόριθμο εκμάθησης των support vectors για να υλοποιήσουμε μια διαδικασία εκμάθησης χρησιμοποιώντας ένα δεδομένα σύνολο από δεδομένα, αποφασίζοντας αυτόματα τον απαιτούμενο αριθμό των hidden units. Με άλλα λόγια: Ενώ ο αλγόριθμος back-propagation είναι σχεδιασμένος ειδικά για να εκπαιδεύσει ένα multilayer perceptron, ο αλγόριθμος support vector είναι πιο γενικής φύσης επειδή έχει πιο ευρεία εφαρμοσιμότητα.

2.1 Βέλτιστη υπερεπιφάνεια για γραμμικά διαχωρίσιμα πρότυπα

Θεωρούμε δεδομένο εκπαίδευσης το {(xi, di)}i=1..n, όπου το xi είναι το πρότυπο εισόδου για το i-οστό παράδειγμα και di είναι το επιθυμητό αποτέλεσμα (target output). Για αρχή, θεωρούμε ότι το πρότυπο (κλάση) που αναπαρίσταται από το υποσύνολο di = +1 και το πρότυπο που αναπαρίσταται από το υποσύνολο di = -1 είναι «γραμμικά διαχωρίσιμα». Η εξίσωση για την επιφάνεια απόφασης στη μορφή μιας υπερεπιφάνειας που κάνει το διαχωρισμό είναι:

$$(1)\quad w^T x + b = 0$$

όπου x είναι ένα διάνυσμα εισόδου (input vector), το w είναι ένα ρυθμιζόμενο διάνυσμα βαρών, και το b είναι το κατώφλι. Μπορούμε επομένως να γράψουμε:

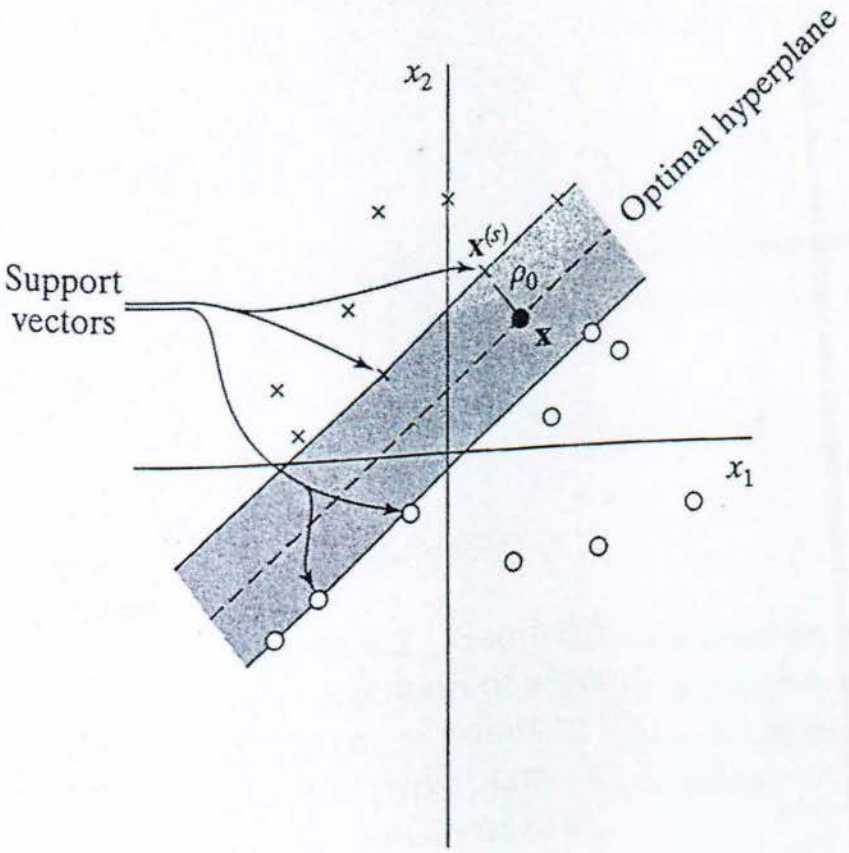
$$(2)\quad w^T x_i + b \geq 0 \text{ για } d_i = +1,$$

$$w^T x_i + b < 0 \text{ για } d_i = -1$$

Η υπόθεση για γραμμικά διαχωρίσιμα πρότυπα γίνεται εδώ για να εξηγήσουμε τη βασική ιδέα πίσω από ένα support vector machine υπό ένα πιο απλό σκηνικό.

Για δεδομένο διάνυσμα βαρών w και κατώφλι b, ο διαχωρισμός μεταξύ της υπερεπιφάνειας που ορίστηκε στην εξίσωση (1) και του κοντινότερου data point (σημείο δεδομένων) λέγεται margin of separation ή αλλιώς περιθώριο διαχωρισμού, που δηλώνεται από το ρ. Ο στόχος ενός support vector machine είναι να βρει την συγκεκριμένη υπερεπιφάνεια για την οποία το περιθώριο διαχωρισμού ρ να είναι το μέγιστο. Υπό αυτές τις συνθήκες η επιφάνεια διαχωρισμού αναφέρεται ως η *βέλτιστη υπερεπιφάνεια (optimal hyperplane)*. Το σχήμα 18

απεικονίζει τη γεωμετρική κατασκευή μιας βέλτιστης υπερεπιφάνειας για ένα δισδιάστατο χώρο εισόδου.



Σχήμα 18 Απεικόνιση μιας βέλτιστης υπερεπιφάνειας για γραμμικά διαχωρίσιμαπρότυπα

Έστω ότι τα w0 και b0 δηλώνουν τις βέλτιστες τιμές του διανύσματος βαρών και του κατωφλίου, αντίστοιχα. Παρομοίως, η βέλτιστη υπερεπιφάνεια, που αναπαριστά μια πολυδιάστατη γραμμική επιφάνεια απόφασης στο χώρο εισόδου, ορίζεται ως

$$(2)\ w^T_0 x + b_0 = 0$$

η οποία είναι επαναδιατύπωση της εξίσωσης (1). Η συνάρτηση

$$(3)\ gx = w^T_0 x + b_0$$

δίνει την αλγεβρική ποσότητα της απόστασης μεταξύ του x και της βέλτιστης υπερεπιφάνειας. Ίσως ο πιό εύκολος τρόπος για να το δούμε αυτό είναι να εκφράσουμε το x ως:

$$x = x_p + r\,(w_0 / \|w_0\|)$$

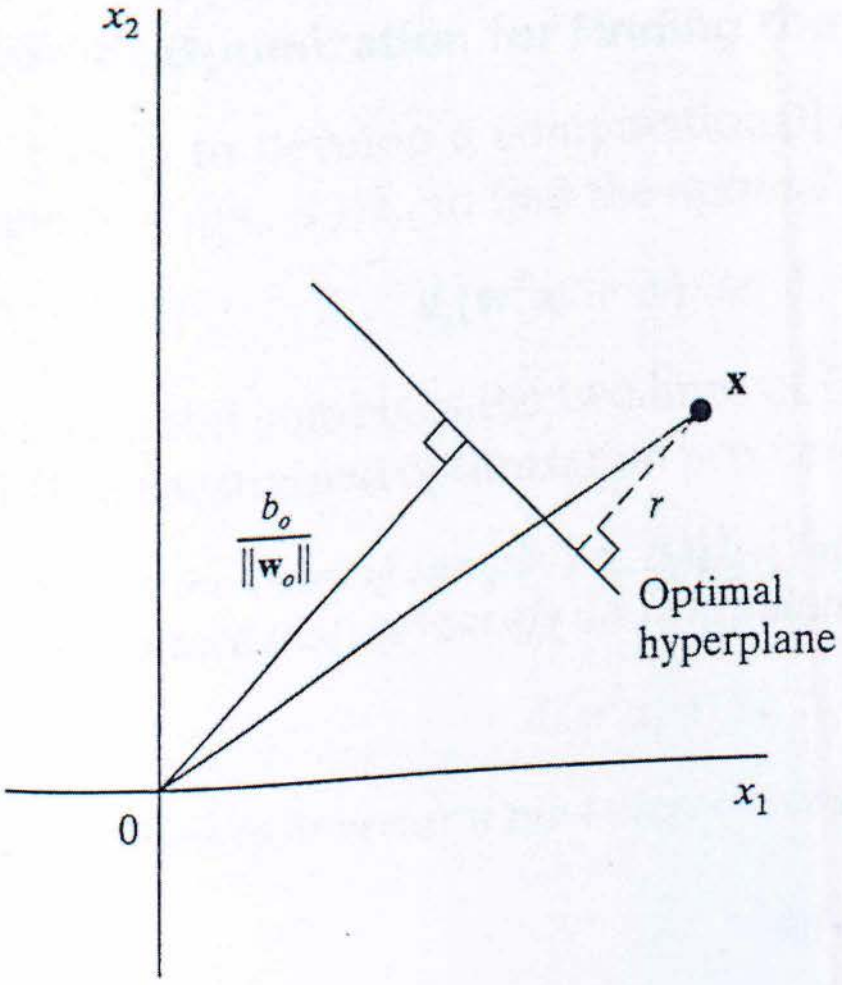
όπου το xp είναι η κανονική προβολή του x πάνω στη βέλτιστη υπερεπιφάνεια, και το r είναι η επιθυμητή αλγεβρική απόσταση (το r είναι θετικό αν το x βρίσκεται στη θετική πλευρά της βέλτιστης υπερεπιφάνειας και αρνητικό αν το x βρίσκεται στην αρνητική πλευρά). Εφόσον, εξορισμού το $g(x_p) = 0$, τότε ακολούθως

$$(5)\ g(x) = wT_0 x + b_0 = r\,\|w_0\|$$

ή

$$r = g(x) / \|w_0\|$$

Πιο συγκεκριμένα, η απόσταση από την αρχή συντεταγμένων (δηλαδή, x = 0) μέχρι τη βέλτιστη υπερεπιφάνεια δίνεται από το $b_0 / \|w_0\|$. Αν $b_0 > 0$, η αρχή συντεταγμένων βρίσκεται στη θετική πλευρά της βέλτιστης υπερεπιφάνειας, και αν $b_0 < 0$, βρίσκεται στην αρνητική πλευρά. Αν $b_0 = 0$, τότε η βέλτιστη υπερεπιφάνεια περνά διαμέσου της αρχής συντεταγμένων. Μια γεωμετρική αναπαράσταση αυτών των αλγεβρικών αποτελεσμάτων βρίσκεται στο σχήμα 19.



Σχήμα 19 Γεωμετρική αναπαράσταση των αλγεβρικών αποστάσεων μεταξύ των σημείων και της βέλτιστης υπερεπιφάνειας για δισδιάστατο χώρο.

Το ζητούμενο είναι να βρούμε τις παραμέτρους w0 και b0 της βέλτιστης υπερεπιφάνειας, δεδομένου ενός συνόλου εκπαίδευσης $J = \{(x_i, d_i)\}$. Λαμβάνοντας υπ' όψη τα αποτελέσματα του σχήματος (3), βλέπουμε ότι το ζεύγος (w_0, b_0) πρέπει να τηρεί τους περιορισμούς:

(6) $w^T_0 x_i + b_0 \geq 1$ για $d_i = +1$

$w^T_0 x_i + b_0 \leq -1$ για $d_i = -1$

Να σημειώσουμε ότι, αν η εξίσωση (2) ισχύει, δηλαδή τα πρότυπα είναι γραμμικά διαχωρίσιμα, μπορούμε πάντα να ξαναφτιάξουμε τα w_0 και b_0 σε μικρότερη κλίμακα έτσι ώστε να ισχύει και η εξίσωση (6) (αυτό αφήνει την εξίσωση (3) ανεπηρέαστη).

Τα ειδικά σημεία (x_i, d_i) για τα οποία η πρώτη ή δεύτερη γραμμή της εξίσωσης (6) ικανοποιείται με το σήμα ισότητας λέγονται διανύσματα υποστήριξης (support vectors), εξ'ού και το όνομα "support vector machine". Αυτά τα διανύσματα (vectors) παίζουν αξιοπρόσεκτο ρόλο στη λειτουργία αυτού του είδους αλγορίθμων εκμάθησης. Τα support vectors είναι εκείνα τα σημεία τα οποία βρίσκονται κοντινότερα στην επιφάνεια απόφασης και επομένως είναι και τα πιο δύσκολα για να κατηγοριοποιηθούν.

Θεωρούμε ένα support vector $x^{(s)}$ για το οποίο $d^{(s)} = +1$. Τότε εξ'ορισμού, έχουμε:

$$(7)\ g(x^{(s)}) = w^T_0\, x^{(s)} \pm b_0 = \pm 1 \text{ για } d^{(s)} = \pm 1$$

Από την εξίσωση (5) η αλγεβρική απόσταση μεταξύ του support vector x(s) και της βέλτιστης υπερεπιφάνειας είναι:

$$(8)\ r = g(x^{(s)}) / \|w_0\|$$

$$= 1 / \|w_0\| \text{ αν } d^{(s)} = +1$$

$$= - 1 / \|w_0\| \text{ αν } d^{(s)} = -1$$

όπου το θετικό πρόσημο υποδηλώνει ότι το $x^{(s)}$ βρίσκεται στη θετική πλευρά της βέλτιστης υπερεπιφάνειας και το αρνητικό πρόσημο υποδηλώνει ότι βρίσκεται στην αρνητική πλευρά. Έστω ότι το ρ δηλώνει τη μέγιστη τιμή του περιθωρίου

διαχωρισμού (margin of separation) μεταξύ δύο κλάσεων που αποτελούνται από το σύνολο εκπαίδευσης J. Τότε από την εξίσωση (8) προκύπτει ότι:

$$(9)\ \rho = 2r = 2 / \|w_0\|$$

Η εξίσωση (9) δηλώνει ότι μεγιστοποιώντας το περιθώριο διαχωρισμού μεταξύ των κλάσεων είναι ισοδύναμο με την ελαχιστοποίηση του Ευκλείδιου μέτρου (Euclidean norm) του διανύσματος βαρών w.

Περιληπτικά, η βέλτιστη υπερεπιφάνεια που ορίζεται από την εξίσωση (3) είναι μοναδική υπό την έννοια ότι το βέλτιστο διάνυσμα βαρών w0 παρέχει το μέγιστο δυνατό διαχωρισμό μεταξύ των θετικών και αρνητικών παραδειγμάτων. Αυτή η βέλτιστη κατάσταση επιτυγχάνεται με την ελαχιστοποίηση του Ευκλείδειου μέτρου του διανύσματος βαρών w.

2.2 Τετραγωνική βελτιστοποίηση για εύρεση της βέλτιστης υπερεπιφάνειας

Ο στόχος μας είναι να αναπτύξουμε μια υπολογιστικά αποδοτική διαδικασία χρησιμοποιώντας τα δεδομένα εκπαίδευσης $J = \{(x_i, d_i)\}_{i=1...n}$ για την εύρεση της βέλτιστης επιφάνειας υπό τον περιορισμό

$$(10)\ d_i(w^T x_i + b) \geq 1 \text{ για } i = 1,2,\ldots,N$$

Αυτός ο περιορισμός συνδυάζει τις δύο γραμμές της εξίσωσης (6) με το w να χρησιμοποιείται στη θέση του w0. Το περιορισμένο πρόβλημα βελτιστοποίησης (constrained optimization problem) που έχουμε να λύσουμε μπορεί να εκφραστεί ως:

Δεδομένου του δείγματος εκπαίδευσης $\{(x_i, d_i)\}_{i=1...n}$, *πρέπει να βρεθούν οι βέλτιστες τιμές του διανύσματος βαρών* w *και κατωφλίου* b *έτσι ώστε να ικανοποιείται ο περιορισμός* $d_i(w^T x_i + b) \geq 1$ *για* $i = 1,2,\ldots,N$ *και το διάνυσμα βαρών* w *να ελαχιστοποιεί τη συνάρτηση κόστους* $\Phi(w) = \frac{1}{2} w^T w$.

Ο παράγοντας ½ συμπεριλαμβάνεται εδώ για ευκολία στην παρουσίαση. Αυτό το περιορισμένο πρόβλημα βελτιστοποίησης λέγεται βασικό πρόβλημα (primal problem).

Μπορεί να χαρακτηριστεί ως ακολούθως:

- Η συνάρτηση κόστους Φ(w) είναι μια κυρτή (convex) συνάρτηση του w
- Οι περιορισμοί στο w είναι γραμμικοί

Επομένως, μπορούμε να λύσουμε το περιορισμένο πρόβλημα βελτιστοποίησης χρησιμοποιώντας τη μέθοδο των πολλαπλασιαστών Lagrange.

Πρώτα κατασκευάζουμε τη συνάρτηση Lagrange:

$$(11)\ J(w,b,\alpha) = \tfrac{1}{2}\, w^T w - \Sigma_{i=1..N}\, \alpha_i\, [d_i\, (w^T x_i + b) - 1]$$

Όπου οι βοηθητικές μη-αρνητικές τιμές αi ονομάζονται *πολλαπλασιαστές Lagrange*. Η λύση στο περιορισμένο πρόβλημα βελτιστοποίησης καθορίζεται από το σημείο J(w,b,α) της Lagrangian συνάρτησης, που πρέπει να ελαχιστοποιηθεί σε σχέση με το w και b,

επίσης πρέπει να μεγιστοποιηθεί σε σχέση με το α. Συνεπώς, διακρίνοντας το J(w,b,α) σε σχέση με τα w και b και θέτοντας τα αποτελέσματα ίσα με το μηδέν, παίρνουμε τις εξής δύο συνθήκες βελτιστοποίησης:

Συνθήκη 1: $\partial\, J(w,b,a) / \partial \mathbf{w} = 0$

Συνθήκη 2: $\partial\, J(w,b,a) / \partial \mathbf{b} = 0$

Η εφαρμογή της πρώτης συνθήκης στη συνάρτηση Lagrange της εξίσωσης (11) παράγει:

$$(12)\ w = \Sigma_{i=1..N} \alpha_i d_i x_i$$

Η εφαρμογή της δεύτερης συνθήκης στη συνάρτηση Lagrange της εξίσωσης (11) παράγει:

$$(13)\ \Sigma_{i=1..N} \alpha_i d_i = 0$$

Αξίζει να σημειώσουμε ότι στο saddle point, για κάθε πολλαπλασιαστή Lagrange αi, το γινόμενο αυτού του πολλαπλασιαστή με τον αντίστοιχο περιορισμό του εξαφανίζεται, όπως φαίνεται και πιο κάτω:

$$(14)\ \alpha_i [d_i (w^T x_i + b) - 1] = 0 \text{ για } i = 1,2,\ldots,N$$

Επομένως, μπορούμε να υποθέσουμε μη-αρνητικές τιμές μόνο στους πολλαπλασιαστές εκείνους οι οποίοι ικανοποιούν ακριβώς την εξίσωση (14). Αυτή η ιδιότητα ακολουθεί τη θεωρία βελτιστοποίησης Kuhn-Tucker.

Όπως αναφέραμε, το βασικό πρόβλημα (primal problem) έχει να κάνει με μια συνάρτηση κόστους και με γραμμικούς περιορισμούς. Δεδομένου ενός τέτοιου περιορισμένου προβλήματος βελτιστοποίησης, είναι πιθανό να κατασκευάσουμε ακόμα ένα πρόβλημα το οποίο λέγεται δυαδικό πρόβλημα (dual problem). Αυτό το δεύτερο πρόβλημα έχει την ίδια βέλτιστη τιμή όπως το primal problem, αλλά με τους πολλαπλασιαστές Lagrange να παρέχουν τη βέλτιστη λύση. Πιο

συγκεκριμένα, μπορούμε να αναφέρουμε το ακόλουθο θεώρημα δυισμού (duality theorem):

a. Αν το primal problem έχει βέλτιστη λύση, το dual problem έχει επίσης μια βέλτιστη λύση, και οι αντίστοιχες βέλτιστες τιμές είναι ίσες.
b. Για να είναι το w0 βέλτιστη primal λύση και το α0 να είναι βέλτιστη dual λύση, είναι απαραίτητο και αρκετό το w0 να είναι εφικτό για το primal problem και

$$\Phi(w0) = J(w0, b0, \alpha0) = \min_w J(w, b0, \alpha0)$$

Επεκτείνουμε την εξίσωση (11) ως ακολούθως

$$(15)\ J(w, b, \alpha) = \tfrac{1}{2} w^T w - \Sigma_{i=1..N} \alpha i\, di\, w^T xi - b \Sigma_{i=1..N} \alpha i\, di + \Sigma_{i=1..N} \alpha i$$

Ο τρίτος όρος στα δεξιά της εξίσωσης (15) είναι 0 από τη συνθήκη βελτιστοποίησης της εξίσωσης (13). Ακόμη, από την εξίσωση (12) έχουμε

$$w^T w = \Sigma_{i=1..N} \alpha i \, di \, w^T xi = \Sigma_{i=1..N} \Sigma_{j=1..N} \alpha i \, \alpha j \, di \, dj \, xi^T xj$$

Ανάλογα, θέτοντας την αντικειμενική συνάρτηση $J(w, b, \alpha) = Q(\alpha)$, μπορούμε να ανασχηματίσουμε την εξίσωση (15) ως

(16) $$Q(\alpha) = \Sigma_{i=1..N} \alpha_i - \tfrac{1}{2} \Sigma_{i=1..N} \Sigma_{j=1..N} \alpha_i \alpha_j d_i d_j x_i^T x_j$$

όπου τα α_i παίρνουν μη αρνητικές τιμές.

Μπορούμε τώρα να ορίσουμε το δυαδικό πρόβλημα:

Δεδομένου του δείγματος εκπαίδευσης $\{(x_i, d_i)\}i=1...N$, *πρέπει να βρεθούν οι πολλαπλασιαστές Lagrange* $\{\alpha_i\}i=1..N$ *οι οποίοι μεγιστοποιούν την αντικειμενική συνάρτηση* $Q(\alpha) = \Sigma_{i=1..N} \alpha i - \tfrac{1}{2} \Sigma_{i=1..N} \Sigma_{j=1..N} \alpha_i \alpha_j d_i d_j x_i T x_j$ *υπό τους περιορισμούς*

1) $\Sigma_{i=1..N} \alpha_i d_i = 0$
2) $\alpha_i \geq 0$ *για* $i = 1,2,..,N$

Το dual problem αποτιμάται πλήρως από τα δεδομένα εκπαίδευσης. Ακόμη, η συνάρτηση Q(α) για να μεγιστοποιηθεί εξαρτάται μόνο από τα πρότυπα εισόδου υπό τη μορφή ενός συνόλου από εσωτερικά γινόμενα, $\{x^T_i, x_j\}_{(i,j)} = 1...N$.

Έχοντας αποφασίσει τους βέλτιστους πολλαπλασιαστές Lagrange, που υποδηλώνονται από το $\alpha_{0,I}$,μπορούμε να υπολογίσουμε το βέλτιστο διάνυσμα βαρών w_0 χρησιμοποιώντας τη συνάρτηση (12) και επομένως να γράψουμε

$$(17)\ w_0 = \Sigma_{i=1...N}\ \alpha_{0,i}\, d_i\, x_i$$

Για να υπολογίσουμε το βέλτιστο κατώφλι b0, μπορούμε να χρησιμοποιήσουμε το w0 που βρήκαμε και να εκμεταλλευτούμε την εξίσωση (7) που αναφέρεται σε ένα θετικό support vector, και επομένως να γράφουμε

$$(18)\ b_0 = 1 - w^T_0\, x^{(s)} \text{ για } d^{(s)} = 1$$

2.3 Στατιστικές ιδιότητες της βέλτιστης υπερεπιφάνειας

Για να εφαρμόσουμε τη μέθοδο structural risk minimization χρειάζεται να κατασκευάσουμε ένα σύνολο από διαχωριστικές υπερεπιφάνειες με διαφοροποιημένο το VC dimension τέτοιο ώστε το empirical risk (δηλαδή το λάθος κατηγοριοποίησης της εκπαίδευσης) και το VC dimension να είναι ελάχιστα την ίδια στιγμή. Σε ένα support

vector machine η δομή επιβάλλεται από το σύνολο των διαχωριστικών υπερεπιφανειών με το να περιορίσουμε το Ευκλείδιο μέτρο του διανύσματος των βαρών w. Μπορούμε να ορίσουμε το ακόλουθο θεώρημα:

Έστω D η διάμετρος της μικρότερης σφαίρας που περιέχει όλα τα διανύσματα εισόδου x_1, x_2, ..., x_N. Το σύνολο από βέλτιστες υπερεπιφάνειες που περίγραφεται από την εξίσωση $w^T{}_0 x + b_0 = 0$ έχει ένα VC dimension h με ανώτατο όριο ως:

$$(19)\ h \leq \min \{\lceil D2/\rho \rceil, m_0\} + 1$$

όπου το σύμβολο οροφής $\lceil . \rceil$ σημαίνει ο μικρότερος ακέραιος αριθμός μεγαλύτερος ή ίσος με τον αριθμό που εγκλείται μέσα, ρ είναι το περιθώριο διαχωρισμού ίσο με $2/||w_0||$, και m_0 είναι η διάσταση του χώρου εισόδου.

Αυτό το θεώρημα μας λέει ότι μπορούμε να ασκήσουμε έλεγχο πάνω στο VC dimension (δηλαδή στην πολυπλοκότητα) της βέλτιστης υπερεπιφάνειας, ανεξάρτητα από την διάσταση m_0 του χώρου εισόδου, με την κατάλληλη επιλογή του περιθωρίου διαχωρισμού ρ.

Υποθέτωντας ότι έχουμε μια ένθετη δομή που περιγράφεται σε σχέση με τις διαχωριστικές υπερεπιφάνειες ως ακουλούθως:

(20) $S_k = \{w^T x + b: \|w\|^2 \leq c_k\}, k = 1,2,..$

Λόγω του άνω ορίου της VC dimension h που ορίστηκε στην εξίσωση (19), η ένθετη δομή που περιγράφηκε στην εξίσωση (20) μπορεί να αναδιαμορφωθεί σε σχέση με το περιθώριο διαχωρισμού με την μορφή:

(21) $Sk = \{\lceil r2/\rho2 \rceil + 1 : \rho2 \geq \alpha k\}, k = 1,2,..$

Τα α_k και c_k είναι σταθερές.

Από τη θεωρία στατιστικής μάθησης (statistical learning theory), για να πετύχουμε καλή ικανότητα γενίκευσης, θα έπρεπε να επιλέξουμε τη συγκεκριμένη δομή με τη μικρότερη VC dimension και μικρότερο λάθος εκπαίδευσης, σύμφωνα με την αρχή του structural risk minimization. Από τις εξισώσεις (19) και (21) βλέπουμε ότι αυτή η απαίτηση μπορεί να ικανοποιηθεί χρησιμοποιώντας τη βέλτιστη υπερεπιφάνεια (δηλαδή, τη διαχωριστική υπερεπιφάνεια με το μεγαλύτερο περιθώριο διαχωρισμού ρ). Ισοδύναμα, από την εξίσωση (9) θα μπορούσαμε να χρησιμοποιήσουμε το βέλτιστο διάνυσμα βαρών w0 που έχει το κατώτατο Ευκλίδειο μέτρο. Επομένως, η επιλογή της βέλτιστης υπερεπιφάνειας για ένα σύνολο από γραμμικά διαχωρίσιμα πρότυπα δεν είναι μόνο διαισθητικά ικανοποιητική αλλά επίσης εκπληρώνει πλήρως την αρχή του structural risk minimization για ένα support vector machine.

2.4 Βέλτιστη υπερεπιφάνεια για μη-γραμμικά διαχωρίσιμα πρότυπα

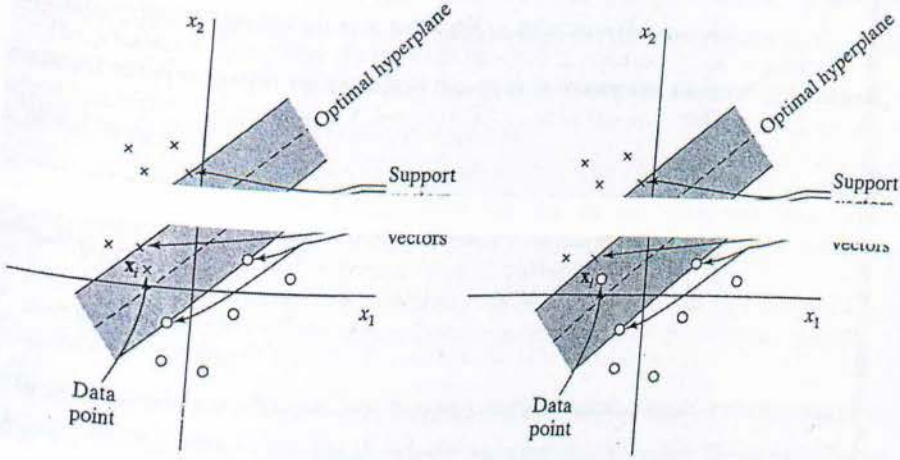
Μέχρι τώρα έχουμε αναφερθεί στα γραμμικά διαχωρίσιμα πρότυπα, τώρα θα ασχοληθούμε με την πιο δύσκολη περίπτωση όπου έχουμε μη γραμμικά διαχωρίσιμα πρότυπα. Δεδομένου ενός τέτοιου συνόλου εκπαίδευσης, είναι δύσκολο να κατασκευάσουμε μια διαχωριστική υπερεπιφάνεια χωρίς να αντιμετωπίσουμε λάθη κατηγοριοποίησης. Παρόλα αυτά, θέλουμε να βρούμε μια βέλτιστη υπερεπιφάνεια η οποία ελαχιστοποιεί την πιθανότητα λαθών στην κατηγοριοποίηση.

Το περιθώριο διαχωρισμού μεταξύ των κλάσεων λέγεται soft δηλαδή μαλακό, στην περίπτωση όπου ένα δεδομένο (xi, di) παραβιάζει την εξής συνθήκη (εξίσωση (10)) :

$$d_i(w^T x_i + b) \geq +1 \text{ για } i = 1,2,\ldots,N$$

Αυτή η παραβίαση μπορεί να προκύψει για δύο λόγους:

- Το δεδομένο (x_i, d_i) πέφτει εντός της περιοχής διαχωρισμού αλλά στη σωστή πλευρά της επιφάνειας απόφασης (σχήμα 20.α)
- Το δεδομένο (x_i, d_i) πέφτει στη λάθος πλευρά της επιφάνειας απόφασης (σχήμα 20.β)



Σχήμα 20 α. Το σημείο x_i (που ανήκει στη κλάση β_1) πέφτει εντός της περιοχής διαχωρισμού αλλά στη σωστή πλευρά της επιφάνειας απόφασης. β. Το σημείο x_i (που ανήκει στην κλάση β_2) πέφτει στη λάθος πλευρά της επιφάνειας απόφασης.

Παρατηρούμε ότι στην πρώτη περίπτωση έχουμε σωστή κατηγοριοποίηση, ενώ στη δεύτερη περίπτωση έχουμε λανθασμένη κατηγοριοποίηση.

Για το χειρισμό των μη γραμμικά διαχωρίσιμων δεδομένων εισάγουμε ένα καινούργιο σύνολο από μη αρνητικές βαθμιδωτές μεταβλητές $\{\xi i\}i=1..N$ στον ορισμό της υπερεπιφάνειας διαχωρισμού (δηλαδή επιφάνειας απόφασης) όπως φαίνεται πιο κάτω:

(22) $d_i (w^T x_i + b) \geq 1 - \xi_i$, για $i = 1,2,\ldots,N$

Η μεταβλητή ξ_i λέγεται *slack variable*, και μετρά την απόκλιση ενός δεδομένου από την ιδανική κατάσταση στο διαχωρισμό προτύπων. Για $0 \leq \xi_i \leq 1$, το δεδομένο πέφτει εντός της περιοχής διαχωρισμού αλλά στη σωστή πλευρά της επιφάνειας απόφασης, όπως φαίνεται και στο σχήμα 4.α. Για $\xi_i > 1$, πέφτει στη λάθος πλευρά της υπερεπιφάνειας διαχωρισμού, όπως φαίνεται και στο σχήμα 4.β. Τα support

vectors είναι εκείνα τα σημεία δεδομένων τα οποία ικανοποιούν ακριβώς την εξίσωση (22), ακόμα και αν $\xi_i > 0$. Σημειώνουμε ότι εάν ένα σημείο με $\xi_i > 0$ δεν συμπεριληφθεί στο σύνολο εκπαίδευσης, τότε η επιφάνεια απόφασης θα αλλάξει. Επομένως, τα support vectors ορίζονται με τον

ίδιο ακριβώς τρόπο για τα διαχωρίσιμα και για τα μη γραμμικά διαχωρίσιμα δεδομένα.

Ο στόχος μας είναι να βρούμε μια διαχωριστική υπερεπιφάνεια για την οποία το λάθος της λανθασμένης κατηγοριοποίησης να είναι το ελάχιστο. Μπορούμε να το κάνουμε αυτό με την ελαχιστοποίηση της συνάρτησης:

$$\Phi(\xi) = \Sigma_{i=1..N} I(\xi_i - 1)$$

σε σχέση με το διάνυσμα βαρών w, υπό τους περιορισμούς που περιγράφηκαν στην εξίσωση (22) και τον περιορισμό στο $\|w_2\|$. Η συνάρτηση $I(\xi)$ είναι μια συνάρτηση δείκτη (*indicator function*) που ορίζεται από το:

$$I(\xi) = 0 \text{ αν } \xi \leq 0$$

$$= 1 \text{ αν } \xi > 0$$

Δυστυχώς, η ελαχιστοποίηση του $\Phi(\xi)$ σε σχέση με το w είναι ένα nonconvex πρόβλημα βελτιστοποίησης που είναι NP-complete.

Για να κάνουμε το πρόβλημα βελτιστοποίησης μαθηματικός υπάκουο, προσεγγίζουμε τη συνάρτηση $\Phi(\xi)$ γράφοντας

$$\Phi(\xi) = \Sigma_{i=1..N} \xi_i$$

Επιπλέον, απλοποιούμε τον υπολογισμό διατυπώνοντας τη συνάρτηση που θα ελαχιστοποιηθεί σε σχέση με το διάνυσμα βαρών w ως ακολούθως:

$$(23)\ \Phi(w, \xi) = \tfrac{1}{2}\, w^T w + C \Sigma_{i=1..N} \xi_i$$

Όπως και πριν, η ελαχιστοποίηση του πρώτου όρου της εξίσωσης (23) σχετίζεται με την ελαχιστοποίηση της VC dimension του support vector machine. Όσο για τον δεύτερο όρο $\Sigma_i\ \xi_i$, είναι ένα άνω όριο στο πλήθος των test errors. Επομένως η μορφοποίηση της συνάρτησης κόστους $\Phi(w, \xi)$ στη συνάρτηση (23) συμφωνεί απόλυτα με την αρχή του structural risk minimization.

Η παράμετρος C ελέγχει την ανταλλαγή ανάμεσα στην πολυπλοκότητα της μηχανής και στο πλήθος των μη διαχωριζόμενων σημείων, μπορούμε επομένως να τη δούμε ως μια μορφή παραμέτρου κανονικοποίησης. Η παράμετρος C πρέπει να επιλεγχθεί από το χρήστη. Αυτό μπορεί να γίνει με δύο τρόπους:

- Η παράμετρος C επιλέγεται πειραματικά μέσω του κοινού τρόπου εκπαίδευσης/επαλήθευσης.
- Επιλέγεται αναλυτικά υπολογίζοντας το VC dimension μέσω της εξίσωσης (19)και χρησιμοποιώντας όρια πάνω στην απόδοση γενίκευσης της μηχανής βάση του VC dimension.

Σε οποιαδήποτε περίπτωση, η συνάρτηση $\Phi(w, \xi)$ βελτιστοποιείται σε σχέση με το w και το $\{\xi_i\}_{i=1..N}$, υπό τον περιορισμό που περιγράφηκε στην εξίσωση (22), και $\xi_i \geq 0$. Έτσι το τετραγωνικό μέτρο του w χρησιμοποιείται ως μια ποσότητα που θα ελαχιστοποιηθεί από κοινού σε σχέση με τα μη γραμμικά διαχωρίσιμα σημεία παρά ως περιορισμός που επιβάλλεται στην ελαχιστοποίηση του πλήθους των μη γραμμικά διαχωρίσιμων σημείων.

Το πρόβλημα βελτιστοποίησης για τα μη γραμμικά διαχωρίσιμα πρότυπα που μόλις δηλώσαμε, περιλαμβάνει το πρόβλημα βελτιστοποίησης για γραμμικά

διαχωρίσιμα πρότυπα ως ειδική περίπτωση. Πιο συγκεκριμένα, θέτοντας το $\xi_i = 0$ για όλα τα i στις

εξισώσεις (22) και (23) τις απλοποιεί στις αντίστοιχες μορφές για την περίπτωση όπου έχουμε γραμμικά διαχωρίσιμα πρότυπα.

Μπορούμε επίσημα να δηλώσουμε το primal πρόβλημα για την περίπτωση των μη γραμμικά διαχωρίσιμων:

Δεδομένου του συνόλου εκπαίδευσης $\{(x_i, d_i)\}$ $i=1..N$, *πρέπει να βρεθούν οι βέλτιστες τιμές του διανύσματος βαρών* w *και κατωφλίου* b *τέτοιες ώστε να ικανοποιούν τον περιορισμό*

$$d_i (w^T x_i + b) \geq 1 - \xi_i \text{ για } i = 1,2,..N$$

$$\xi_i \geq 0 \text{ για όλα τα } i$$

και τέτοια ώστε το διάνυσμα βαρών w και οι μεταβλητές slack variables ξi να ελαχιστοποιούν τη συνάρτηση κόστους

$$\Phi(w, \xi) = \frac{1}{2} w^T w + C \Sigma_{i=1..N} \xi_i$$

όπου το C είναι μια θετική παράμετρος που καθορίζεται από το χρήστη.

Χρησιμοποιώντας τη μέθοδο των πολλαπλασιαστών Lagrange μπορούμε να διαμορφώσουμε το dual problem για μη γραμμικά διαχωρίσιμα πρότυπα ως:

Δεδομένου του συνόλου εκπαίδευσης $\{(x_i, d_i)\}_{i=1..N}$, *πρέπει να βρεθούν οι πολλαπλασιαστές Lagrange* $\{\alpha_i\} i=1..N$ *οι οποίοι μεγιστοποιούν την αντικειμενική συνάρτηση*

$$Q(\alpha) = \Sigma_{i=1..N} \alpha_i - \frac{1}{2} \Sigma_{i=1..N} \Sigma_{j=1..N} \alpha_i \alpha_j d_i d_j x^T_i x_j$$

Υπό τους περιορισμούς

1) $\Sigma_{i=1..N} \alpha_i d_i = 0<$
2) $0 \leq \alpha_i \leq C$ *για* $I = 1,2,..N$

όπου το C είναι μια θετική παράμετρος που καθορίζεται από το χρήστη.

Σημειώνουμε ότι ούτε τα slack variables ξi ούτε οι πολλαπλασιαστές Lagrange εμφανίζονται στο dual problem. Το dual problem για μη γραμμικά διαχωρίσιμα πρότυπα είναι επομένως παρόμοιο με αυτό για διαχωρίσιμα πρότυπα εκτός μιας μικρής αλλά σημαντικής διαφοράς. Η αντικειμενική συνάρτηση Q(α) η οποία πρέπει να μεγιστοποιηθεί είναι η ίδια και στις δύο περιπτώσεις. Η περίπτωση μη γραμμικά διαχωρίσιμων διαφέρει από την περίπτωση διαχωρήσιμων στο ότι ο περιορισμός $\alpha_i \geq 0$ αντικαθιστάτε με τον πιό αυστηρό περιορισμό $0 \leq \alpha_i \leq C$. Εκτός από αυτή την τροποποίηση, η υπόλοιπη διαδικασία για υπολογισμό των βέλτιστων τιμών του διανύσματος βαρών και του κατωφλίου είναι η ίδια για τις δύο περιπτώσεις. Όπως επίσης και τα support vectors ορίζονται με τον ίδιο τρόπο όπως προηγουμένως.

Η βέλτιστη λύση για το διάνυσμα βαρών w δίνεται από το

$$(24)\ w_0 = \Sigma_{i=1..Ns}\ \alpha_{0,i}\, d_i\, x_i$$

όπου το Ns είναι το πλήθος των support vectors.

Για τον καθορισμό των βέλτιστων τιμών του κατωφλίου ακολουθείται μια διαδικασία παρόμοια με αυτή που περιγράφηκε προηγουμένως. Πιο συγκεκριμένα οι συνθήκες Kuhn-Tucker τώρα μπορούν να οριστούν ως

$$(25)\ \alpha_i [d_i (w^T x_i + b) - 1 + \xi_i] = 0,\ i = 1,2,..N$$

και

$$(26)\ \mu_i \xi_i = 0,\ i = 1,2,..N$$

Η εξίσωση (25) είναι επαναδιατύπωση της εξίσωσης (14) εκτός από την αντικατάσταση του όρου μονάδα (1) με τον όρο (1 - ξi). Όσο για την εξίσωση (26), το μ_i είναι πολλαπλασιαστές Lagrange που εισάχθηκαν για να ενδυναμώσουν την μη-αρνητικότητα των μεταβλητών slack variables ξi για όλα τα i. Στο σημείο saddle point το παράγωγο της συνάρτηση Lagrange για το primal problem σε σχέση με την μεταβλητή ξi είναι μηδέν, η αποτίμηση του οποίου αποδίδεται ως

$$(27)\ \alpha_i + \mu_i = C$$

Συνδυάζοντας τις εξισώσεις (26) και (27) βλέπουμε ότι

$$(28)\ \xi_i = 0 \text{ αν } \alpha_i < C$$

Μπορούμε να προσδιορίσουμε το βέλτιστο κατώφλι b_0 παίρνοντας οποιοδήποτε σημείο (x_i, d_i) από το σύνολο εκπαίδευσης για το οποίο ισχύει $0< \alpha_0,i < C$ και επομένως $\xi_i = 0$, και χρησιμοποιώντας εκείνο το σημείο στη εξίσωση (25). Από πλευράς αριθμητικής είναι καλύτερα να πάρουμε τη μέση τιμή του b_0 που απορρέει από όλα τα σημεία στο σύνολο εκπαίδευσης.

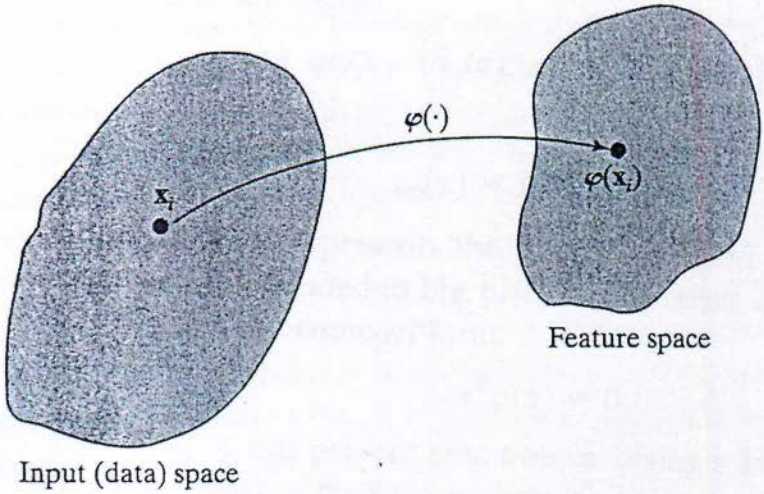
2.5 Πώς να δημιουργήσεις ένα support vector machine για αναγνώριση προτύπου

Σε αυτό το σημείο θα περιγράψουμε την κατασκευή ενός support vector machine για αναγνώριση προτύπου.

Γενικά η ιδέα ενός support vector machine βασίζεται σε δύο μαθηματικές λειτουργίες οι οποίες περιγράφονται περιληπτικά εδώ και απεικονίζονται στο σχήμα (21):

1. Μη γραμμική συσχέτιση ενός διανύσματος εισόδου σε ένα χώρο χαρακτηριστικών (feature space) ψηλότερης διάστασης που είναι κρυφός (hidden) από την είσοδο και έξοδο.
2. Δημιουργία μιας βέλτιστης υπερεπιφάνειας για διαχωρισμό των χαρακτηριστικών που βρέθηκαν στο βήμα 1.

Η λογική των πιο πάνω λειτουργιών εξηγείται πιο κάτω:



Σχήμα 21 Μη γραμμική συσχέτιση φ(.) από το χώρο εισόδου (input space) σε ένα χώρο χαρακτηριστικών (feature space)

Η πρώτη λειτουργία εκτελείται σύμφωνα με το θεώρημα του Cover για τη γραμμική διαχωρισιμότητα των προτύπων. Θεωρούμε ένα χώρο εισόδου που αποτελείται από μη γραμμικά διαχωρίσιμα πρότυπα. Το θεώρημα του Cover δηλώνει ότι ένας τέτοιος πολυδιάστατος χώρος μπορεί να μετασχηματιστεί σε ένα καινούργιο χώρο χαρακτηριστικών (feature space) όπου τα πρότυπα είναι γραμμικά διαχωρίσιμα με μεγάλη πιθανότητα όταν ικανοποιούνται δύο συνθήκες. Πρώτη, ο μετασχηματισμός είναι μη γραμμικός. Δεύτερο, η διάσταση του χώρου χαρακτηριστικών είναι αρκετά ψηλή. Αυτές οι δύο συνθήκες είναι ενσωματωμένες στη λειτουργία 1. Να σημειωθεί όμως ότι το θεώρημα του Cover δεν αναφέρεται στη βελτιστοποίηση της υπερεπιφάνειας διαχωρισμού. Μόνο με τη χρήση μιας βέλτιστης διαχωριστικής υπερεπιφάνειας η VC dimension ελαχιστοποιείται και επιτυγχάνεται γενίκευση. Και εδώ είναι που μπαίνει η δεύτερη λειτουργία. Πιο συγκεκριμένα, η λειτουργία 2 εκμεταλλεύεται την ιδέα όπου κατασκευάζεται μια βέλτιστη διαχωριστική υπερεπιφάνεια σε σχέση με τη θεωρία που εξηγήθηκε. Για μη γραμμικά διαχωρίσιμα πρότυπα, αλλά με μια βασική διαφορά: Η διαχωριστική υπερεπιφάνεια ορίζεται τώρα ως μια γραμμική συνάρτηση από διανύσματα που εξάγεται από το χώρο χαρακτηριστικών (feature space) παρά από τον αρχικό χώρο εισόδου. Πολύ σημαντικό το γεγονός ότι η κατασκευή αυτής της υπερεπιφάνειας εκτελείται σε σχέση με την αρχή του structural risk minimization που έχει τις ρίζες του στη θεωρία του VC dimension. Η κατασκευή εξαρτάται από τον υπολογισμό τον πυρήνα εσωτερικού γινομένου (inner product kernel).

2.6 Πυρήνας εσωτερικού γινομένου (inner product kernel)

Έστω ότι το x δηλώνει ένα διάνυσμα που εξάγεται από το χώρο εισόδου, υποθέτοντας ότι είναι της διάστασης m_0. Έστω ότι το $\{\varphi_j(x)\}_{j=1..m1}$ δηλώνει ένα σύνολο από μη γραμμικούς μετασχηματισμούς από το χώρο εισόδου στο χώρο χαρακτηριστικών (m_1 είναι η διάσταση του χώρου χαρακτηριστικών). Υποθέτουμε ότι το $\varphi_j(x)$ ορίζεται ως το priori για όλα τα j. Δεδομένου ενός τέτοιου συνόλου από μη γραμμικούς σχηματισμούς, μπορούμε να ορίσουμε την υπερεπιφάνεια ως την επιφάνεια απόφασης ως ακολούθως:

$$(29)\ \Sigma_{j=1..m1}\, w_j\, \varphi_j(x) + b = 0$$

Όπου το $\{w_j\}_{j=1..m1}$ δηλώνει ένα σύνολο από γραμμικά βάρη που συνδέουν το χώρο χαρακτηριστικών με το χώρο εξόδου, και b είναι το κατώφλι. Μπορούμε να το απλοποιήσουμε γράφοντας:

$$(30)\ \Sigma_{j=0..m1}\, w_j\, \varphi_j(x) = 0$$

Όπου υποθέτουμε ότι το $\varphi_0(x) = 1$ για όλα τα x, έτσι ώστε το w0 να δηλώνει το κατώφλι b. Η εξίσωση (30) ορίζει την επιφάνεια απόφασης που υπολογίζεται στο feature space σε σχέση με τα γραμμικά βάρη της μηχανής. Η ποσότητα $\varphi_j(x)$ απεικονίζει την είσοδο που προμηθεύτηκε στο βάρος w_j διαμέσου του feature space. Ορίζουμε το διάνυσμα

$$(31)\ \varphi(x) = [\varphi_0(x), \varphi_1(x), .., \varphi_{m1}(x)]^T$$

όπου εξ' ορισμού έχουμε

(32) $\varphi_0(x) = 1$ για όλα τα x

Στην ουσία, το διάνυσμα φ(x) αναπαριστά την «εικόνα» που προκλήθηκε στο χώρο χαρακτηριστικών λόγω του διανύσματος εισόδου x, όπως απεικονίζεται στο σχήμα (4). Επομένως, όσο αφορά αυτή την εικόνα μπορούμε να ορίσουμε την επιφάνεια απόφασης

στη μορφή:

(33) $w^T \varphi(x) = 0$

Προσαρμόζοντας την εξίσωση (12) στην παρούσα κατάσταση που περιλαμβάνει ένα feature space στο οποίο ψάχνουμε γραμμικό διαχωρισμό των χαρακτηριστικών, μπορούμε να γράψουμε:

(34) $w = \Sigma_{i=1..N} \alpha_i d_i \varphi(x_i)$

όπου το διάνυσμα χαρακτηριστικών φ(xi) αντιστοιχεί στο πρότυπο εισόδου xi για το i– οστο παράδειγμα. Επομένως, αντικαθιστώντας την εξίσωση (34) στην (33), μπορούμε να ορίσουμε την επιφάνεια απόφασης που υπολογίζεται στο χώρο χαρακτηριστικών ως:

(35) $\Sigma_{i=1..N} \alpha_i d_i \varphi^T(x_i) \varphi(x) = 0$

Ο όρος $\varphi^T(x_i)\varphi(x)$ αναπαριστά το εσωτερικό γινόμενο δύο διανυσμάτων που προκλήθηκαν στο χώρο χαρακτηριστικών από το διάνυσμα εισόδου x και το πρότυπο εισόδου xi που αναφέρεται στο i-οστό παράδειγμα. Μπορούμε επομένως να εισάγουμε τον πυρήνα εσωτερικού γινομένου που υποδηλώνεται από το $K(x, x_i)$ και ορίζεται από

$$(36)\ K(x, x_i) = \varphi^T(x)\,\varphi(x_i)$$

$$= \Sigma_{j=0..m1}\, \varphi_j(x)\,\varphi(x_i) \text{ για } i = 1,2,..N$$

Από αυτό τον ορισμό βλέπουμε ότι ο πυρήνας εσωτερικού γινομένου είναι μια συμμετρική συνάρτηση των παραμέτρων του, όπως φαίνεται και από:

$$(37)\ K(x, x_i) = K(x_i, x) \text{ για όλα τα } i$$

Το πιο σημαντικό, είναι ότι μπορούμε να χρησιμοποιήσουμε τον πυρήνα εσωτερικού γινομένου $K(x, x_i)$ για να κατασκευάσουμε τη βέλτιστη υπερεπιφάνεια στο χώρο χαρακτηριστικών χωρίς να χρειαστεί να μελετήσουμε άμεσα τον ίδιο το χώρο χαρακτηριστικών. Αυτό εύκολα μπορούμε να το δούμε χρησιμοποώντας της εξίσωση (36) στην (35), όπου η βέλτιστη υπερεπιφάνεια ορίζεται από

$$(38)\ \Sigma_{i=1..N}\, \alpha_i\, d_i\, K(x, x_i) = 0$$

2.7 Βέλτιστος σχεδιασμός ενός support vector machine

Η επέκταση του πυρήνα εσωτερικού γινομένου K(x, xi) στην εξίσωση (36) μας επιτρέπει να κατασκευάσουμε μια επιφάνεια απόφασης που είναι μη γραμμική στο χώρο εισόδου, αλλά η εικόνα του στο χώρο χαρακτηριστικών είναι γραμμική. Με αυτή την επέκταση, μπορούμε να ορίσουμε την dual μορφή για την περιορισμένη βελτιστοποίηση ενός support vector machine ως ακολούθως:

Δεδομένου ενός δείγματος εκπαίδευσης $\{(x_i, d_i)\}i=1..N$, *πρέπει να βρουμε τους πολλαπλασιαστές Lagrange* $\{\alpha_i\}i=1..N$ *που μεγιστοποιούν την αντικειμενική συνάρτηση*

$$(40)\ Q(\alpha) = \Sigma_{i=1..N} \alpha_i - \tfrac{1}{2} \Sigma_{i=1..N} \Sigma_{j=1..N} \alpha_i \alpha_j d_i d_j K(x_i, x_j)$$

Υπό τους περιορισμούς:

1. $\Sigma_{i=1..N} \alpha_i d_i = 0$

2. $0 \leq \alpha_i \leq C$ *για* $i = 1,2,..N$

Όπου το C είναι μια θετική παράμετρος που καθορίζεται από το χρήστη.

Να σημειώσουμε ότι ο περιορισμός (1) προκύπτει από τη βελτιστοποίηση της Lagrangian Q(α) σε σχέση με το κατώφλι $b = w_0$ για $\varphi_0(x) = 1$. Το dual πρόβλημα που μόλις αναφέραμε είναι της ίδιας μορφής οπώς εκείνο για τα μη γραμμικά διαχωρίσιμα πρότυπα, εκτός από το γεγονός ότι το εσωτερικό γινόμενο $x_i^T x_j$ που χρησιμοποιείται έχει αντικατασταθεί με τον πυρήνα εσωτερικού γινομένου $K(x_i, x_j)$. Μπορούμε να δούμε το $K(x_i, x_j)$ ως το ij-οστο στοιχείο ενός συμμετρικού N-επί-N πίνακα K, όπως φαίνεται από

(41) $K = \{K(x_i, x_j)\}_{(i,j)=1..N}$

Έχοντας βρει τις βέλτιστες τιμές των πολλαπλασιαστών Lagrange, που υποδηλώνονται από το $\alpha_{0,i}$, μπορούμε να αποφασίσουμε τις αντίστοιχες βέλτιστες τιμές του γραμμικού διανύσματος βαρών, w_0, που συνδέει το χώρο χαρακτηριστικών με το χώρο εξόδου εφαρμόζοντας τον τύπο της εξίσωσης (17) στην καινούργια κατάσταση. Πιο συγκεκριμένα, αναγνωρίζοντας ότι η εικόνα $\varphi(x_i)$ παίζει το ρόλο της εισόδου στο διάνυσμα βαρών w, μπορούμε να ορίσουμε το w_0 ως

(42) $w_0 = \Sigma_{i=1..N} \alpha_{o,i} d_i \varphi(x_i)$

όπου το $\varphi(x_i)$ είναι η εικόνα που προκαλείται στο χώρο χαρακτηριστικών λόγω του x_i.

Σημειώνουμε ότι η πρώτη συνιστώσα του w_0 αναπαριστά το βέλτιστο κατώφλι b_0.

2.8 Παραδείγματα των support vector machines

Η ανάγκη για τον πυρήνα K(x, xi) είναι για να ικανοποιήσουμε το θεώρημα του Mercer. Σε αυτή την απαίτηση υπάρχει μια ελευθερία για το πώς θα επιλεχτεί. Στον πίνακα πιο κάτω συνοψίζουμε τους πυρήνες εσωτερικού γινομένου για τρεις κοινούς τύπους support vector machines: Polynomial, Learning Machine, Radial-Basis Function Network, και Two-Layer Perceptron. Αξίζει να σημειώσουμε τα πιο κάτω σημεία:

1. Οι πυρήνες εσωτερικού γινομένου για polynomial και radial-basis function types των support vector machines πάντα ικανοποιούν το θεώρημα του Mercer. Σε αντίθεση, οι πυρήνες εσωτερικού γινομένου για το two-layer perceptron type του support vector machine είναι κατά κάποιο τρόπο περιορισμένο, όπως φαίνεται στην τελευταία γραμμή του πίνακα. Αυτό είναι μια χειροπιαστή απόδειξη στο γεγονός ότι η απόφαση για το αν ένας πυρήνας ικανοποιεί ή όχι το θεώρημα του Mercer μπορεί όντως να είναι δύσκολο ζήτημα.

2. Και για τα τρία είδη μηχανών, η διάσταση του χώρου χαρακτηριστικών καθορίζεται από το πλήθος των support vectors που εξάγχθηκαν από τα δεδομένα εκπαίδευσης μέσω της λύσης στο περιορισμένο πρόβλημα βελτιστοποίησης.

3. Η θεμελιώδης θεωρία ενός support vector machine αποφεύγει την ανάγκη για heuristics που συχνά χρησιμοποιούνται στο σχεδιασμό των συνηθισμένων radialbasis function networks και των multilayer perceptrons:

a. Στον radial basis function τυπό ενός support vector machine, το πλήθος των radial basis functions και τα κέντρα τους καθορίζονται αυτόματα από το πλήθος των support vector machines και των τιμών τους, αντίστοιχα.

b. Στον two-layer perceptron τύπο ενός support vector machine, το πλήθος των κρυφών νευρώνων και τα διανύσματα των βαρών τους καθορίζονται αυτόματα από το πλήθος των support vectors και των τιμών τους,

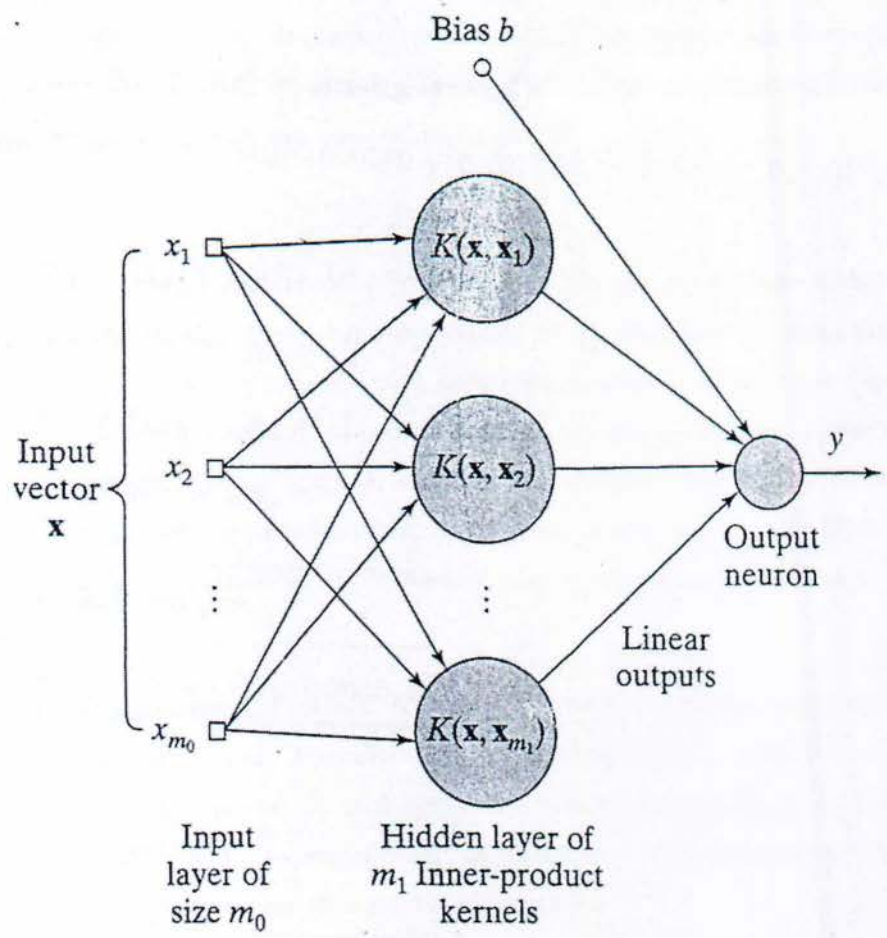
αντίστοιχα.

Περίληψη των πυρήνων εσωτερικού γινομένου

Τύπος του support vector machine	**Πυρήνας εσωτερικού γινομένου $K(x, x_i), i=1,2,..,N$**	**Σχόλια**
Polynomial Learning Machine	$(x^T x_i + 1)^p$	Η δύναμη p ορίζεται ως priori από το χρήστη
Radial-Basis function network	$\exp((- 1/(2\sigma^2)) * \|\|x - x_i\|\|^2)$	Το πλάτος σ^2, κοινό σε όλους τους πυρήνες, καθορίζεται ως priori από το χρήστη
Two-layer perceptron	$\tanch(\beta_0 x^T x_i + \beta_i)$	Το θεώρημα του Mercer ικανοποιείται μόνο για μερικές τιμές β_0 και β_i

Πίνακας 23 Πίνακας όπου περιληπτικά αναφέρονται οι πυρήνες εσωτερικού γινομένου

Στο σχήμα παρουσιάζεται η αρχιτεκτονική ενός support vector machine.



Σχήμα 24 Αρχιτεκτονική ενός support vector machine

Άσχετα από το πώς θα υλοποιηθεί ένα support vector machine, διαφέρει από την συνηθισμένη προσέγγιση στο σχεδιασμό ενός multilayer perceptron με θεμελιώδη διαφορά. Στη συνηθισμένη προσέγγιση, η πολυπλοκότητα του μοντέλου ελέγχεται κρατώντας το πλήθος των χαρακτηριστικών (δηλαδή των κρυφών νευρώνων) μικρό. Από την άλλη, το support vector machine προσφέρει μια λύση στο σχεδιασμό μιας μηχανής εκμάθησης ελέγχοντας την πολυπλοκότητα του μοντέλου ανεξάρτητα από τη διάσταση, όπως συνοψίζεται εδώ

- **Conceptual problem.** Η διάσταση του χώρου χαρακτηριστικών (hidden) είναι σκόπιμα φτιαγμένη πολύ μεγάλη για να επιτρέψει την κατασκευή μιας επιφάνειας απόφασης στη μορφή υπερεπιφάνειας σε αυτό τον χώρο. Για καλή απόδοση στη γενίκευση, η πολυπλοκότητα του μοντέλου ελέγχεται με την επιβολή κάποιων περιορισμών στην κατασκευή της διαχωριστικής υπερεπιφάνειας, που έχει ως αποτέλεσμα την εξαγωγή ενός κλάσματος (fraction) των δεδομένων εκπαίδευσης ως support vectors.

- **Computational Problem.** Αυτό το υπολογιστικό πρόβλημα αποφεύγεται χρησιμοποιώντας την ιδέα ενός πυρήνα εσωτερικού γινομένου (που ορίζεται σύμφωνα με το θεώρημα του Mercer) και επιλύοντας τη διπλή μορφή του περιορισμένου προβλήματος βελτιστοποίησης που διαμορφώθηκε στο χώρο εισόδου (δεδομένων).

ΚΕΦΑΛΑΙΟ 3 – LibSVM

Για την εργασία αυτή χρησιμοποιήθηκε το πρόγραμμα Libsvm 3.1. Το Libsvm είναι ένα απλό, εύκολο στη χρήση, και αποτελεσματικό λογισμικό για SVM ταξινόμηση και παλινδρόμηση και είναι διαθέσιμο στο http://www.csie.ntu.edu.tw/~cjlin/libsvm/ . Λύνει C-SVM ταξινόμηση, nu-SVM ταξινόμηση, one-class-SVM, epsilon-SVM παλινδρόμηση, και nu-SVM παλινδρόμησης. Παρέχει επίσης ένα εργαλείο αυτόματης επιλογής μοντέλο C-SVM ταξινόμησης. Το παρών πρόγραμμα υποστηρίχθηκε εν μέρει από το Εθνικό Επιστημονικό Συμβούλιο της Ταϊβάν, μέσω της επιχορήγησης NSC 89-2.213-E-002-106 από τους Chih-Chung Chang και Chih-Lin Jen.

3.1 Εισαγωγή Δεδομένων

Labeled μορφή του αρχείου δεδομένων.
Η μορφή των αρχείων training και testing είναι:

<label> <index1>:<value1> <index2>:<value2>

Το <label> είναι η επιθυμητή τιμή των δεδομένων εκπαίδευσης. Για κατηγοριοποίηση, μπορεί να είναι μια ακέραια τιμή η οποία αναπαριστά μια κλάση (υποστηρίζεται και multi-class κατηγοριοποίηση). Το <value> είναι ένας πραγματικός (real) αριθμός. Τα labels στο αρχείο επαλήθευσης χρησιμοποιούνται μόνο για υπολογισμό της ακρίβειας ή του λάθους. Αν η τιμή τους δεν είναι γνωστή, απλά συμπληρώνεται αυτή η στήλη με ένα αριθμό.

1	1:0.05286	2:0.0127586	3:0.07929512	4:0.044	5:0.0172
1	1:0.0611	2:0.05724	3:0.056751	4:0.0218	5:0.0088
1	1:0.0580939	2:0.008	3:0.08292199	4:0.06639	5:0.024
1	1:0.04453656	2:0.01960784	3:0.1090688	4:0.04857	5:0.0235
-1	1:0.0570191	2:0.03508756	3:0.0307649	4:0.01311	5:0.0131.
-1	1:0.0355556	2:0.0431031	3:0.084	4:0.04	5:0.0086
-1	1:0.0588247	2:0.0375	3:0.100834454	4:0.0294	5:0.0166
-1	1:0.06779915	2:0.04098377	3:0.076678	4:0.02925	5:0.0122
-1	1:0.0673023	2:0.018264	3:0.07630769	4:0.04308	5:0.0136
-1	1:0.07692769	2:0.0084384	3:0.0895897	4:0.02	5:0.0084
-1	1:0.036194	2:0.038	3:0.11301	4:0.02783	5:0.0042

Σχήμα 25 Labeled μορφή του αρχείου δεδομένων.

3.2 Χρήσιμες εφαρμογές

Υπάρχουν κάποια χρήσιμα προγράμματα σε αυτό το πακέτο.

SVM-scale:

Αυτό είναι ένα εργαλείο για την κλιμάκωση των δεδομένων των αρχείων εισόδου.

SVM-toy:

Αυτή είναι μια απλή γραφική διεπαφή που δείχνει πώς το SVM χωρίζει δεδομένα σε ένα σχεδιάγραμμα . το κουμπί "change" είναι για να επιλέξετε κατηγορία 1, 2 ή 3 (δηλαδή έως και τρεις κλάσεις υποστηρίζονται), το "load" κουμπί για να φορτώσετε τα δεδομένα από ένα αρχείο, το κουμπί "save" για να αποθηκεύσετε τα δεδομένα σε ένα αρχείο, το "run" για να δημιουργηθεί ένα αρχείο SVM model, και το "clear " κουμπί για να καθαρίσει το παράθυρο.

Μπορείτε να εισάγετε τις επιλογές στο κάτω μέρος του παραθύρου, με τη σύνταξη των επιλογών να είναι το ίδιο με αυτό του «SVM-train».

Σημειώστε ότι οι επιλογές "load" και "save " λαμβάνουν υπόψη τα στοιχεία μόνο για ταξινόμηση και όχι για παλινδρόμηση. Κάθε σημείο δεδομένων διαθέτει μία ετικέτα (το χρώμα), το οποίο πρέπει να είναι 1, 2 ή 3 και δύο ιδιότητες (άξονας x και y-άξονα τιμών) στο [0,1].

Επιλέξτε «make» για την κατασκευή τους. Τα προ-χτισμένα δυαδικά αρχεία των Windows είναι στο φάκελο «windows». Χρησιμοποιούμε Visual C + + σε μια μηχανή 32-bit, έτσι ώστε το μέγιστο μέγεθος της μνήμης einaiai cache 2GB.

3.2.1 Υλοποίηση εφαρμογών πακέτου

«SVM-train»

SVM-train [επιλογές] training_set_file [model_file] επιλογές:

-s svm_type: τύπος SVM (προεπιλεγμένη τιμή 0)

0 – C-SVC

1 - nu-SVC

2 – one-class SVM

3 - epsilon-SVR

4 - nu-SVR

-t kernel_type: τύπος του πυρήνα (προεπιλεγμένη τιμή 2)

0 - γραμμικός: v * u '

1 - πολυωνυμικός: (gamma*u'*v + coef0)^degree

2 - ακτινικών συναρτήσεων: exp (-γάμμα * | u-v | ^ 2)

3 - sigmoid: tanh(gamma*u'*v + coef0)

4 - precomputed kernel (kernel values in training_set_file)

-d degree : βαθμός λειτουργίας του πυρήνα (προεπιλογή 3)

-g gamma: γάμμα του πυρήνα (προεπιλογή 1/num_features)

-r coef0: coef0 του πυρήνα (προεπιλεγμένη τιμή 0)

-c cost: ρυθμίζει την παράμετρο C από C-SVC, Epsilon-SVR, και nu-SVR (προεπιλογή 1)

-n nu: ρυθμίστε την παράμετρο της nu nu-SVC, one-class SVM, και nu-SVR (προεπιλογή 0,5)

-p epsilon : για να ορίσετε το έψιλον στην συνάρτηση απώλειας της Epsilon-SVR (προεπιλογή 0,1)

-m CacheSize: ορίστε το μέγεθος της μνήμης cache σε MB (default 100)

-e epsilon:ορίζει την ανοχή του κριτηρίου τερματισμού (προεπιλογή 0,001)

-h shrinking : εάν θέλετε να χρησιμοποιήσετε τη συρρίκνωση heuristics, 0 ή 1 (προεπιλογή 1)

-b probability_estimates: καθορισμός προγραμματισμού ενός SVC ή SVR μοντέλου για τις εκτιμήσεις πιθανότητας, 0 ή 1 (προεπιλεγμένη τιμή 0)

-wi weight: ρυθμίστε την παράμετρο C της κατηγορίας I σε weight*C, για το C-SVC (προεπιλογή 1)

-v n: n-fold cross mode επικύρωση

-q: Λειτοθργεία χωρίς μυνήματα εξόδου.

Το k στην επιλογής-g σημαίνει τον αριθμό των χαρακτηριστικών των δεδομένων εισόδου.

«SVM-predict»

SVM-predict [επιλογές] test_file model_file output_file επιλογές:

-b probability_estimates: αν πρέπει να προβλέψουμε τις εκτιμήσεις πρόβλεψης , 0 ή 1 (προεπιλεγμένη τιμή 0) για one-class SVM μόνο 0 υποστηρίζεται model_file είναι το αρχείο μοντέλο που παράγεται από SVM-train.

test_file είναι τα δεδομένα των δοκιμών που θέλετε να προβλέψετε.

Το SVM-predict θα παράγει έξοδο στο output_file.

«SVM-scale»

SVM-scale [επιλογές] data_filename επιλογές:

-l lower: x κλιμάκωση κατώτερο όριο (προεπιλογή -1)

-u upper: x κλιμάκωση ανώτατο όριο (προεπιλογή +1)

-y y_upper y_lower: y όρια κλιμάκωσης (προεπιλογή: όχι κλιμάκωση y)

-s save_filename: για να αποθηκεύσετε τις παραμέτρους προσαύξησης στο save_filename

-r restore_filename: επαναφορά παραμέτρων κλιμάκωσης από το restore_filename

3.2.2 Παραδείγματα εντολών εκτέλεσης

> svm-scale -l -1 -u 1 -s range train > train.scale

> svm-scale -r range test > test.scale

> svm-train -s 0 -c 5 -t 2 -g 0.5 -e 0.1 data_file

> svm-train -s 3 -p 0.1 -t 0 data_file

> svm-train -s 0 -b 1 data_file

> svm-predict -b 1 test_file data_file.model output_file

3.3 Τύποι πυρήνων Kernel (Kernel Types)

Ο χρήστης έχει την επιλογή μέσω ενός ορίσματος, να διαλέξει τον τύπο του Kernel που θέλει να χρησιμοποιήσει. Οι τέσσερις πιο βασικοί πυρήνες (kernels) είναι: linear, polynomial, radial basic function (RBF) και sigmoid.

1. **Linear**: $K(x, x_i) = x^T x_i$
2. **Polynomial**: Ο πολυωνυμικός πυρήνας δύναμης d είναι της μορφής $K(x, x_i) = (x, x_i)^d$
3. **RBF**: Ο γκαουσιανός πυρήνας (Gaussian) επίσης γνωστός και ως radial basis function, είναι της μορφής $K(x, x_i) = \exp(- (||x_i - x_j||^2) / (2\sigma^2))$
4. **Sigmoid**: $K(x, x_i) = \tanh(k(x, x_i) + \theta)$

Όταν χρησιμοποιείται sigmoid kernel με το SVM μπορεί να θεωρηθεί ως twolayer network.

3.4 Τύποι μηχανών υποστήριξης διανυσμάτων (SVM Types)

Ο χρήστης έχει την επιλογή να διαλέξει τον τύπο του SVM που θέλει να χρησιμοποιήσει. Τα SVM Types που παρέχονται από τον πρόγραμμα είναι τα εξής: C-SVC, nu-SVR, one-class SVM, epsilon-SVR, nu-SVC.

3.4.1 C-SVC: C-Support Vector Classification (Binary Case)

Θεωρούμε το δεδομένο εκπαίδευσης $\{(x_i, d_i)\}_{i=1..n}$, όπου το x_i είναι το πρότυπο εισόδου για το i-οστό παράδειγμα και di είναι το επιθυμητό αποτέλεσμα (target output) με τιμές {+1, -1}.

Το support vector machine απαιτεί λύση στο ακόλουθο πρόβλημα βελτιστοποίησης:

Δεδομένου του συνόλου εκπαίδευσης $\{(x_i, d_i)\}_{i=1..N}$, πρέπει να βρεθούν οι βέλτιστες τιμές του διανύσματος βαρών w και κατωφλίου b τέτοιες ώστε να ικανοποιούν τον περιορισμό

$$d_i (w^T x_i + b) \geq 1 - \xi_i \text{ για } i = 1,2,..N$$
$$\xi_i \geq 0$$

για όλα τα i και τέτοια ώστε το διάνυσμα βαρών w και οι μεταβλητές slack variables ξ_i να ελαχιστοποιούν τη συνάρτηση κόστους

$$\Phi(w, \xi) = \tfrac{1}{2} wT\, w + C \Sigma_{i=1..N} \xi_i$$

όπου το C είναι μια θετική παράμετρος που καθορίζεται από το χρήστη.

Η συνάρτηση απόφασης είναι $sgn(\Sigma_{i=1..N} d_i \alpha_i K(x_i,x) + b)$.

3.4.2 nu-SVC: ν-Support Vector Classification (Binary Case)

Η παράμετρος $\nu \in (0,1)$ είναι άνω όριο στο κλάσμα των λαθών εκπαίδευσης και κάτω όριο στο κλάσμα των support vectors. Δεδομένου του συνόλου εκπαίδευσης $\{(xi, di)\}$ i=1..N, πρέπει να βρεθούν οι βέλτιστες τιμές του διανύσματος βαρών w και κατωφλίου b τέτοιες ώστε να ικανοποιούν τον περιορισμό:

$$d_i (wT\, x_i + b) \geq \rho - \xi_i \text{ για } i = 1,2,..N$$

$$\xi_i \geq 0, \rho \geq 0$$

για όλα τα i και τέτοια ώστε το διάνυσμα βαρών w και οι μεταβλητές slack variables ξ_i να ελαχιστοποιούν τη συνάρτηση κόστους

$$\Phi(w, \xi) = \frac{1}{2} w^T w - \theta\rho + (1/N) \Sigma_{i=1..N} \xi_i$$

Η συνάρτηση απόφασης είναι $\text{sgn}(\Sigma_{i=1..N} d_i \alpha_i K(x_i,x) + b)$.

3.4.3 One-class SVM: distribution estimation

Η διαφορά στο one-class SVM σε σχέση με τα συνηθισμένα είναι ότι τα δεδομένα εκπαίδευσης δεν είναι πανομοιότυπα κατανεμημένα με τα δεδομένα ελέγχου. Τα δεδομένα περιέχουν δύο κλάσεις: μια απο αυτές, η κλάση που στοχεύουμε (target class), ελέγχεται καλά, ενώ η άλλη κλάση είναι απούσα ή ελέγχεται αραιά λόγω των λίγων δειγμάτων. Η προσέγγιση του Scholkopf et al. [9] προτείνει να γίνεται διαχωρισμός μεταξύ της target class και του origin μέσω μιας υπερεπιφάνειας. Δεδομένου του συνόλου εκπαίδευσης $\{(x_i, d_i)\}_{i=1..N}$, πρέπει να βρεθούν οι βέλτιστες τιμές του διανύσματος βαρών w και κατωφλίου b τέτοιες ώστε να ικανοποιούν τον περιορισμό:

$$w^T x_i + b \geq \rho - \xi_i \text{ για } i = 1,2,..N$$

$$\xi_i \geq 0 \text{ για όλα τα } i$$

και τέτοια ώστε το διάνυσμα βαρών w και οι μεταβλητές slack variables ξi να ελαχιστοποιούν τη συνάρτηση κόστους

$$\Phi(w, \xi) = \frac{1}{2} w^T w - \rho + (1/N) \Sigma_{i=1..N} \xi_i.$$

Η συνάρτηση απόφασης είναι $\text{sgn}(\Sigma_{i=1..N} d_i \alpha_i K(x_i,x) + b)$.
Για τους πιο κάτω τύπους epsilon-SVR: ε-Support Vector Regression (ε SVR), nu-SVR: ν-Support Vector Regression (ν-SVR) δεν έχει γίνει περιγραφή εφόσον χρησιμοποιούνται για regression, ενώ στη δική μας περίπτωση χρησιμοποιούμε classification.

3.5 Cross Validation

Ο λόγος που χρησιμοποιείται cross validation είναι για να βρεθούν οι καλές παράμετροι έτσι ώστε ο classifier να μπορεί να προβλέψει με ακρίβεια τα άγνωστα δεδομένα [10].

Ένας κοινός τρόπος είναι να χωρίσεις τα δεδομένα εκπαίδευσης σε δύο μέρη, εκ των οποίων το ένα θεωρείται άγνωστο για την εκπαίδευση του classifier. Έτσι η ακρίβεια της πρόβλεψης σε αυτό το σύνολο μπορεί καλύτερα να αντικατοπτρίσει την απόδοση της κατηγοριοποίησης άγνωστων δεδομένων. Μια βελτιωμένη έκδοση αυτής της διαδικασίας είναι το cross-validation.

Σε μια v-fold cross-validation, το σύνολο δεδομένων χωρίζεται σε v υποσύνολα του ίδιου μεγέθους. Σειριακά, ένα υποσύνολο ελέγχεται χρησιμοποιώντας ένα classifier που είχε εκπαιδευτεί στα υπόλοιπα (v – 1) υποσύνολα. Επομένως, κάθε δείγμα (instance) από ολόκληρο το σύνολο εκπαίδευσης προβλέπεται μόνο μια φορά επομένως η ακρίβεια του cross-validation είναι το ποσοστό των δεδομένων που είχαν σωστή κατηγοριοποίηση. Η διαδικασία του cross-validation μπορεί να εμποδίσει το πρόβλημα όπου γίνεται overfitting [10].

3.6 Shrinking

Οι Chang και Lin [10] ανάφεραν ότι εφόσον για πολλά προβλήματα το πλήθος των ελεύθερων support vectors (δηλαδή $0 \leq \alpha i \leq C$) είναι μικρό, η τεχνική αυτή μειώνει το μέγεθος του εργασιακού προβλήματος χωρίς να λαμβάνει υπόψην του κάποιες περιορισμένες (bounded) μεταβλητές.

3.7 Caching

Αυτή είναι ακόμη μια τεχνική για να μειώσουμε τον υπολογιστικό χρόνο. Εφόσον ο πίνακας Q (ο Q είναι ένας NxN θετικός semi definite πίνακας, Qij = di dj K(xi, xj)) είναι πλήρως πυκνός και μπορεί να μην αποθηκευθεί στη μνήμη του υπολογιστή, τα στοιχεία Qij υπολογίζονται όπως χρειάζεται. Συνήθως μια ειδική μνήμη που χρησιμοποιεί την ιδέα της cache χρησιμοποιείται για να φυλάγονται τα Qij που χρησιμοποιήθηκαν πιο πρόσφατα [10].

ΚΕΦΑΛΑΙΟ 4 – ΥΛΟΠΟΙΗΣΕΙΣ

4.1 Τύπος του SVM (SVM Type)

Οι τύποι των SVMs είναι οι: C-SVC, nu-SVC, epsilon-SVR, nu-SVR, One-Class SVM. Για τα δικά μας παραδείγματα τα οποία είναι κατηγοριοποίησης ο κατάλληλος από τους πέντε πιο πάνω τύπους είναι ο πρώτος, δηλαδή ο C-SVC. Οι δύο τελευταίοι epsilon-SVR, nu-SVR εξαιρούνται εφόσον αυτοί αναφέρονται σε regression και όχι σε classification που δεν μας ενδιαφέρει στην περίπτωση μας. Όπως επίσης και οι τύποι nu-SVC και One-class SVM δε μπορούν να χρησιμοποιηθούν λόγω του ότι αναφέρονται σε διαφορετικό είδος από αυτό που μας απασχολεί.

Τα δεδομένα που χρησιμοποιήθηκαν είναι με χρήση και των τεσσάρων παραμέτρων

4.2 Παράδειγμα 1: Καρκίνος του μαστού

Ο καρκίνος του μαστού είναι ένα από τα πιο κοινά είδη καρκινώματος που ταλαιπωρεί χιλιάδες γυναίκες, αρκετές φορές με μοιραία αποτελέσματα, ανά τον κόσμο. Καθώς πρόκειται για μια ασθένεια όπου η πρόβλεψη της είναι κάτι που μπορεί να σώσει τον ασθενή, αν γίνει εγκαίρως , αποτελεί ένα τέλειο παράδειγμα για να δοκιμάσουμε την ικανότητα των SVM.

Χρησιμοποιήσαμε δεδομένα που λάβαμε από την επιστημονική ιστοσελίδα http://archive.ics.uci.edu. Τα δεδομένα αυτά είναι από το Διαγνωστικό Κέντρο για τον Καρκίνο του Μαστού του Wisconsin (WDBC) και αφορούν μετρήσεις για κύτταρα που έχουν παρθεί από γυναίκες υγιείς αλλά και διαγνωσμένες με καρκίνο του μαστού.

Δέκα πραγματικές τιμές-χαρακτηριστικά υπολογίζονται για κάθε πυρήνα του κυττάρου:

1) Ακτίνα του πυρήνα

2) Υφή (τυπική απόκλιση σε τιμές της γκρίζας κλίμακας)

3) Περίμετρος

4) Περιοχή

5) Ομαλότητα (τοπική παραλλαγή σε μήκη ακτίνας)

6) Συμπαγότητα (περίμετρο ^ 2 / περιοχή - 1.0)

7) Κοιλότητα (σοβαρότητα των κοίλων τμημάτων του περιγράμματος)

8) Κοίλα σημεία (αριθμός κοίλων τμημάτων του περιγράμματος)

9) Συμμετρία

10) Fractal διάσταση ("coastline approximation " - 1)

Κάθε σειρά αποτελεί μια διαφορετική κλάση με ιδιότητες που αναφέραμε πιο πάνω. Τα δεδομένα μας λοιπόν, όπως τα πήραμε από την ιστοσελίδα http://archive.ics.uci.edu έχουν αυτή την μορφή:

```
842302,M,17.99,10.38,122.8,1001,0.1184,0.2776,0.3001,0.1471,0.2419,0.07871,1.095,0.9053,8.589,153.4,0.006399,0.04904,
842517,M,20.57,17.77,132.9,1326,0.08474,0.07864,0.0869,0.07017,0.1812,0.05667,0.5435,0.7339,3.398,74.08,0.005225,0.0
84300903,M,19.69,21.25,130,1203,0.1096,0.1599,0.1974,0.1279,0.2069,0.05999,0.7456,0.7869,4.585,94.03,0.00615,0.04006,
84348301,M,11.42,20.38,77.58,386.1,0.1425,0.2839,0.2414,0.1052,0.2597,0.09744,0.4956,1.156,3.445,27.23,0.00911,0.0745
84358402,M,20.29,14.34,135.1,1297,0.1003,0.1328,0.198,0.1043,0.1809,0.05883,0.7572,0.7813,5.438,94.44,0.01149,0.0246
843786,M,12.45,15.7,82.57,477.1,0.1278,0.17,0.1578,0.08089,0.2087,0.07613,0.3345,0.8902,2.217,27.19,0.00751,0.03345,
844359,M,18.25,19.98,119.6,1040,0.09463,0.109,0.1127,0.074,0.1794,0.05742,0.4467,0.7732,3.18,53.91,0.004314,0.01382,
84458202,M,13.71,20.83,90.2,577.9,0.1189,0.1645,0.09366,0.05985,0.2196,0.07451,0.5835,1.377,3.856,50.96,0.008805,0.03
844981,M,13,21.82,87.5,519.8,0.1273,0.1932,0.1859,0.09353,0.235,0.07389,0.3063,1.002,2.406,24.32,0.005731,0.03502,0.
84501001,M,12.46,24.04,83.97,475.9,0.1186,0.2396,0.2273,0.08543,0.203,0.08243,0.2976,1.599,2.039,23.94,0.007149,0.07
845636,M,16.02,23.24,102.7,797.8,0.08206,0.06669,0.03299,0.03323,0.1528,0.05697,0.3795,1.187,2.466,40.51,0.004029,0.
84610002,M,15.78,17.89,103.6,781,0.0971,0.1292,0.09954,0.06606,0.1842,0.06082,0.5058,0.9849,3.564,54.16,0.005771,0.0
846226,M,19.17,24.8,132.4,1123,0.0974,0.2458,0.2065,0.1118,0.2397,0.078,0.9555,3.568,11.07,116.2,0.003139,0.08297,0.
846381,M,15.85,23.95,103.7,782.7,0.08401,0.1002,0.09938,0.05364,0.1847,0.05338,0.4033,1.078,2.903,36.58,0.009769,0.0
84667401,M,13.73,22.61,93.6,578.3,0.1131,0.2293,0.2128,0.08025,0.2069,0.07682,0.2121,1.169,2.061,19.21,0.006429,0.05
84799002,M,14.54,27.54,96.73,658.8,0.1139,0.1595,0.1639,0.07364,0.2303,0.07077,0.37,1.033,2.879,32.55,0.005607,0.042
848406,M,14.68,20.13,94.74,684.5,0.09867,0.072,0.07395,0.05259,0.1586,0.05922,0.4727,1.24,3.195,45.4,0.005718,0.0116
84862001,M,16.13,20.68,108.1,798.8,0.117,0.2022,0.1722,0.1028,0.2164,0.07356,0.5692,1.073,3.854,54.18,0.007026,0.025
849014,M,19.81,22.15,130,1260,0.09831,0.1027,0.1479,0.09498,0.1582,0.05395,0.7582,1.017,5.865,112.4,0.006494,0.01893,
8510426,B,13.54,14.36,87.46,566.3,0.09779,0.08129,0.06664,0.04781,0.1885,0.05766,0.2699,0.7886,2.058,23.56,0.008462,
8510653,B,13.08,15.71,85.63,520,0.1075,0.127,0.04568,0.0311,0.1967,0.06811,0.1852,0.7477,1.383,14.67,0.004097,0.0189
8510824,B,9.504,12.44,60.34,273.9,0.1024,0.06492,0.02956,0.02076,0.1815,0.06905,0.2773,0.9768,1.909,15.7,0.009606,0.
8511133,M,15.34,14.26,102.5,704.4,0.1073,0.2135,0.2077,0.09756,0.2521,0.07032,0.4388,0.7096,3.384,44.91,0.006789,0.05
```

Όπως αναφέραμε πρέπει να τα ανακατασκευάσουμε έτσι ώστε να είναι στην μορφή:

<label> <index1>:<value1> <index2>:<value2>

```
2.000000  1:1067444.000000 2:2.000000 3:1.000000 4:1.000000 5:1.000000 6:2.000000 7:1.000000 8:2.000000 9:1.000000 10:1.00
2.000000  1:1070935.000000 2:1.000000 3:1.000000 4:3.000000 5:1.000000 6:2.000000 7:1.000000 8:1.000000 9:1.000000 10:1.00
2.000000  1:1070935.000000 2:3.000000 3:1.000000 4:1.000000 5:1.000000 6:1.000000 7:1.000000 8:2.000000 9:1.000000 10:1.00
2.000000  1:1071760.000000 2:2.000000 3:1.000000 4:1.000000 5:1.000000 6:2.000000 7:1.000000 8:3.000000 9:1.000000 10:1.00
4.000000  1:1072179.000000 2:10.000000 3:7.000000 4:7.000000 5:3.000000 6:8.000000 7:5.000000 8:7.000000 9:4.000000 10:3.00
2.000000  1:1074610.000000 2:2.000000 3:1.000000 4:1.000000 5:2.000000 6:2.000000 7:1.000000 8:3.000000 9:1.000000 10:1.00
2.000000  1:1075123.000000 2:3.000000 3:1.000000 4:2.000000 5:1.000000 6:2.000000 7:1.000000 8:2.000000 9:1.000000 10:1.00
2.000000  1:1079304.000000 2:2.000000 3:1.000000 4:1.000000 5:1.000000 6:2.000000 7:1.000000 8:2.000000 9:1.000000 10:1.000000
4.000000  1:1080185.000000 2:10.000000 3:10.000000 4:10.000000 5:8.000000 6:6.000000 7:1.000000 8:8.000000 9:9.000000 10:1.0000
2.000000  1:1081791.000000 2:6.000000 3:2.000000 4:1.000000 5:1.000000 6:1.000000 7:1.000000 8:7.000000 9:1.000000 10:1.000000
4.000000  1:1084584.000000 2:5.000000 3:4.000000 4:4.000000 5:9.000000 6:2.000000 7:10.000000 8:5.000000 9:6.000000 10:1.000000
4.000000  1:1091262.000000 2:2.000000 3:5.000000 4:9.000000 5:3.000000 6:6.000000 7:7.000000 8:7.000000 9:5.000000 10:1.000000
4.000000  1:1099510.000000 2:10.000000 3:4.000000 4:3.000000 5:1.000000 6:3.000000 7:3.000000 8:6.000000 9:5.000000 10:2.000000
4.000000  1:1100524.000000 2:6.000000 3:10.000000 4:10.000000 5:2.000000 6:8.000000 7:10.000000 8:7.000000 9:3.000000 10:3.0000
4.000000  1:1102573.000000 2:5.000000 3:6.000000 4:5.000000 5:6.000000 6:10.000000 7:1.000000 8:3.000000 9:1.000000 10:1.000000
4.000000  1:1103608.000000 2:10.000000 3:10.000000 4:10.000000 5:4.000000 6:8.000000 7:1.000000 8:8.000000 9:10.000000 10:1.000
2.000000  1:1103722.000000 2:1.000000 3:1.000000 4:1.000000 5:1.000000 6:2.000000 7:1.000000 8:2.000000 9:1.000000 10:2.000000
4.000000  1:1105257.000000 2:3.000000 3:7.000000 4:7.000000 5:4.000000 6:4.000000 7:9.000000 8:4.000000 9:3.000000 10:1.000000
```

Εφόσον τα έχουμε στην κατάλληλη μορφή, τα επεξεργαζόμαστε με το εργαλείο SVM-scale για την σωστή κλιμάκωση των δεδομένων:

```
2 1:-0.860107 2:-0.111111 3:-1 4:-1 5:-1 6:-0.777778 7:-1 8:-0.555556 9:-1 10:-1
2 1:-0.859671 2:-0.111111 3:-0.333333 4:-0.333333 5:-0.111111 6:0.333333 7:1 8:-0.555556 9:-0.777778 10:-1
2 1:-0.857807 2:-0.555556 3:-1 4:-1 5:-1 6:-0.777778 7:-0.777778 8:-0.555556 9:-1 10:-1
2 1:-0.85768 2:0.111111 3:0.555556 4:0.555556 5:-1 6:-0.555556 7:-0.333333 8:-0.555556 9:0.333333 10:-1
2 1:-0.857569 2:-0.333333 3:-1 4:-1 5:-0.555556 6:-0.777778 7:-1 8:-0.555556 9:-1 10:-1
4 1:-0.857554 2:0.555556 3:1 4:1 5:0.555556 6:0.333333 7:1 8:0.777778 9:0.333333 10:-1
2 1:-0.857408 2:-1 3:-1 4:-1 5:-1 6:-0.777778 7:1 8:-0.555556 9:-1 10:-1
2 1:-0.857339 2:-0.777778 3:-1 4:-0.777778 5:-1 6:-0.777778 7:-1 8:-0.555556 9:-1 10:-1
2 1:-0.855171 2:-0.777778 3:-1 4:-1 5:-1 6:-0.777778 7:-1 8:-1 9:-1 10:-0.111111
2 1:-0.855171 2:-0.333333 3:-0.777778 4:-1 5:-1 6:-0.777778 7:-1 8:-0.777778 9:-1 10:-1
2 1:-0.854841 2:-1 3:-1 4:-1 5:-1 6:-1 7:-1 8:-0.555556 9:-1 10:-1
2 1:-0.854709 2:-0.777778 3:-1 4:-1 5:-1 6:-0.777778 7:-1 8:-0.777778 9:-1 10:-1
4 1:-0.853868 2:-0.111111 3:-0.555556 4:-0.555556 5:-0.555556 6:-0.777778 7:-0.555556 8:-0.333333 9:-0.333333 10:-1
2 1:-0.85354 2:-1 3:-1 4:-1 5:-1 6:-0.777778 7:-0.555556 8:-0.555556 9:-1 10:-1
4 1:-0.853454 2:0.555556 3:0.333333 4:-0.111111 5:1 6:0.333333 7:0.777778 8:-0.111111 9:-0.111111 10:-0.333333
4 1:-0.852997 2:0.333333 3:-0.333333 4:0.111111 5:-0.333333 6:0.111111 7:-1 8:-0.333333 9:-0.555556 10:-1
2 1:-0.852842 2:-0.333333 3:-1 4:-1 5:-1 6:-0.777778 7:-1 8:-0.777778 9:-1 10:-1
2 1:-0.852671 2:-0.333333 3:-1 4:-1 5:-1 6:-0.777778 7:-1 8:-0.555556 9:-1 10:-1
4 1:-0.852543 2:1 3:0.333333 4:0.333333 5:0.111111 6:-0.333333 7:1 8:-0.333333 9:-1 10:-0.777778
2 1:-0.852536 2:0.111111 3:-1 4:-1 5:-1 6:-0.777778 7:-1 8:-0.555556 9:-1 10:-1
4 1:-0.851958 2:0.333333 3:-0.555556 4:-0.777778 5:1 6:-0.111111 7:1 8:-0.111111 9:-0.333333 10:-0.333333
```

Πλέον έχουμε τα αρχεία μας σε σωστή μορφή, έτοιμα για την διαδικασία της εκπαίδευσης του SVM.

Ανοίγουμε την κονσόλα cmd(command prompt) και μετακινούμαστε με την εντολή cd στον κατάλογο που βρίσκετε το πρόγραμμα libsvm.

Μετά, τρέχουμε το πρόγραμμα svm-train με το αρχείο που περιέχει τα δεδομένα:

Svm-train.exe breast-cancer_scale.train

Το SVM πλέον έχει εκπαιδευτεί και μπορεί να κάνει classification βάσει των δεδομένων που δέχτηκε. Έχει επίσης δημιουργηθεί ένα αρχείο που αποτελεί το μοντέλο για την πρόβλεψη με κατάληξη .model . Για να κάνουμε μια πρόβλεψη αρκεί να τρέξουμε svm-predict με παραμέτρους ένα αρχείο με τα δεδομένα που θέλουμε να γίνουν classified , το αρχείο με κατάληξη .model και ένα αρχείο εξόδου:

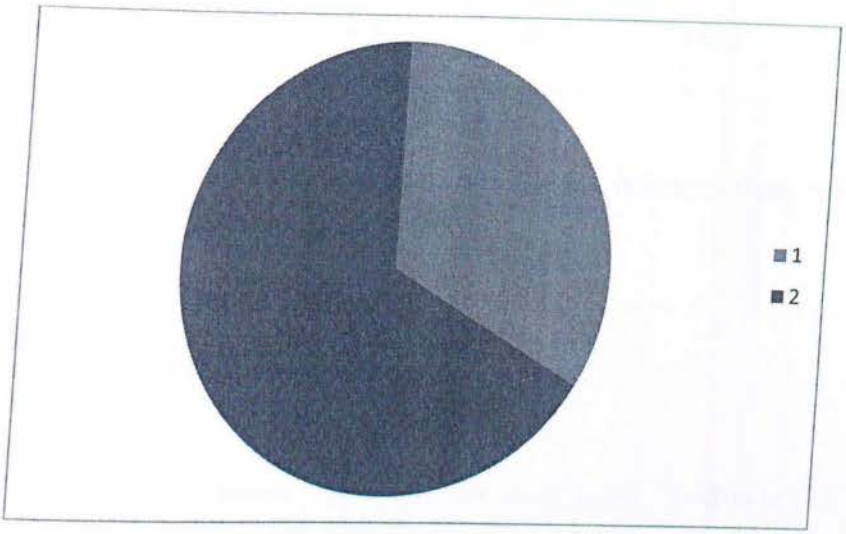
Svm-predict.exe breast-cancer.test breast-cancer_scale.train.model breast-cancer.out

Το αποτέλεσμα μας είναι το ποσοστό επιτυχίας του classification δηλαδή κατά πόσο καταφέραμε να κατηγοριοποιήσουμε τα δεδομένα που περιείχε το αρχείο breast-cancer.test .

Το αρχείο που δημιουργήθηκε (breast-cancer.out) περιέχει την κατηγοριοποίηση της κάθε κλάσης (γραμμής-ατόμου) σε 2 κατηγορίες:

1. Άτομα των οποίων τα κύτταρα μπορεί να εμφανίσουν καρκινώματα στο μέλλον.

2. Άτομα των οποίων τα κύτταρα είναι σχετικά ασφαλή από μελλοντικά καρκινώματα.



Πίνακας 4.2 Αποτελέσματα classification

A) Τύπος του SVM (SVM Type)

Όπως αναφέρθηκε ο τύπος που χρησιμοποιήθηκε για το δικό μας πρόβλημα είναι ο C-SVC που είναι ο καταλληλότερος τύπος SVM για να χρησιμοποιηθεί σε προβλήματα κατηγοριοποίησης (classification). Η αποδοτικότητα αυτού του τύπου επηρεάζεται από την παράμετρο cost. Για να ελέγξουμε αυτή τη παράμετρο έχουμε κρατήσει σταθερά τα δεδομένα εισόδου, το kernel type και τις παραμέτρους του. Πιο κάτω, ο πίνακας 4.2.1 απεικονίζει τις διάφορες τιμές που δοκιμάστηκαν για την παράμετρο cost κρατώντας τα υπόλοιπα σταθερά. Παρατηρούμε ότι το κόστος στον τύπο του SVM επηρεάζει αισθητά την απόδοση του δικτύου. Δοκιμάζοντας διάφορες τιμές του cost σε διαφορετικούς kernel types κάθε φορά έχω παρατηρήσει ότι δεν υπάρχει μια βέλτιστη τιμή του cost που να παράγει τα καλύτερα αποτελέσματα σε κάθε kernel type. Ο κάθε kernel type επηρεάζεται διαφορετικά με την τιμή του κόστους και για κάθε ένα υπάρχει διαφορετική βέλτιστη τιμή του cost.

Cost	1	5	10	20

Accuracy %	98,3	98,3	98,3	98,3

Πίνακας 4.2.1 Πίνακας με τα ποσοστά επιτυχίας για διάφορες τιμές της παραμέτρου cost για τον τύπο των SVM C-SVC

Β) Τύπος του Kernel (Kernel Type)

Δεδομένου ότι εμείς θέλαμε έναν μη γραμμικό τρόπο πρόβλεψης ο καταλληλότερος τύπος πυρήνα ήταν ο RBF.

Στον πιο κάτω πινάκα ελέγξαμε τον τύπο radial basis για πέντε διαφορετικές τιμές της παραμέτρου gamma και για SVM type cost = 5. Παρατηρούμε ότι όταν το gamma πάρει μικρή τιμή αποδίδει πιο καλά.

Gamma	0.2	1	2.5	5	10
Accuracy%	98,3	98,3	96,5	84,2	84,2

Πίνακας 4.2.2 Πίνακας με τα ποσοστά επιτυχίας για διάφορες τιμές της παραμέτρου cost για τον Τύπος του Kernel RBF

Γ) Απόδοση δικτύου με τις καλύτερες τιμές

Δεδομένου ότι έχουμε επιλέξει τον τύπο των SVM, C-SVC, και τον τύπο του Kernel , RBF, είχαμε να πειραματιστούμε μόνο με τις παραμέτρους cost και gamma. Στους παρακάτω πίνακες εμφανίζονται τα ποσοστά επιτυχίας επί τις εκατό για τις διάφορες τιμές αυτών των μεταβλητών:

	Gamma =	Gamma = 1	Gamma =	Gamma = 5	Gamma =

	0.2		2.5		10
Cost=1	97,2	84,2	84,2	84,2	84,2
Cost=5	98,3	89,5	84,2	84,2	84,2
Cost=10	97,2	89,5	84,2	84,2	84,2
Cost=20	97,3	89,5	84,2	84,2	84,2

Πίνακας 4.2.3 Πίνακας με τα ποσοστά επιτυχίας για διάφορες τιμές της παραμέτρου cost και gamma για τον Τύπος του Kernel RBF και τον τύπο των SVM, C-SVC

Δ) Συμπεράσματα

Από τους πιο πάνω πίνακες διαπιστώνουμε ότι τα καλύτερα αποτελέσματα τα παίρνουμε για cost = 5 και gamma= 0,2. Επίσης αντιλαμβανόμαστε πως ο παράγοντας κόστους δεν επηρεάζει το αποτέλεσμα τόσο πολύ όσο ο παράγοντας gamma

4.3 Παράδειγμα 2: Ηπατικές διαταραχές

Χρησιμοποιήσαμε δεδομένα που λάβαμε από την επιστημονική ιστοσελίδα http://archive.ics.uci.edu. Τα δεδομένα αυτά είναι από το BUPA Medical Research Ltd. και αφορούν μετρήσεις για άτομα που μετείχαν στην σχετική έρευνα.

Οι πρώτες 5 μεταβλητές είναι όλες εξετάσεις αίματος οι οποίες πιστεύεται ότι είναι ευαίσθητες σε διαταραχές του ήπατος που μπορεί να προκύψουν από την υπερβολική κατανάλωση αλκοόλ. Κάθε γραμμή αποτελεί την καταγραφή ενός αρσενικού ατόμου.

1. MCV Μέσος όγκος
2. Alkphos αλκαλική φωσφατάση
3. SGPT Alamine αμινοτρανσφεράση
4. SGOT αμινοτρανσφεράση
5. Gammagt γάμμα-γλουταμυλοτρανσπεπτιδάση
6. Ποτά αριθμός μισού-pint αλκοολούχων ποτών ανά ημέρα
7. Selector πεδίο που χρησιμοποιείται για να χωρίσει τα δεδομένα σε δύο σύνολα (δεν χρησιμοποιείται σαν δεδομένο εκπαίδευσης)

Κάθε σειρά αποτελεί μια διαφορετική κλάση με ιδιότητες που αναφέραμε πιο πάνω. Τα δεδομένα μας λοιπόν, όπως τα πήραμε από την ιστοσελίδα http://archive.ics.uci.edu έχουν αυτή την μορφή:

```
88,67,21,11,11,0.5,1
92,54,22,20,7,0.5,1
90,60,25,19,5,0.5,1
89,52,13,24,15,0.5,1
82,62,17,17,15,0.5,1
90,64,61,32,13,0.5,1
86,77,25,19,18,0.5,1
96,67,29,20,11,0.5,1
91,78,20,31,18,0.5,1
89,67,23,16,10,0.5,1
89,79,17,17,16,0.5,1
91,107,20,20,56,0.5,1
94,116,11,33,11,0.5,1
92,59,35,13,19,0.5,1
```

Όπως αναφέραμε πρέπει να τα ανακατασκευάσουμε έτσι ώστε να είναι στην μορφή:

<label> <index1>:<value1> <index2>:<value2>

```
1  1:85.000000 2:92.000000 3:45.000000 4:27.000000 5:31.000000 6:0.000000
2  1:85.000000 2:64.000000 3:59.000000 4:32.000000 5:23.000000 6:0.000000
2  1:86.000000 2:54.000000 3:33.000000 4:16.000000 5:54.000000 6:0.000000
2  1:91.000000 2:78.000000 3:34.000000 4:24.000000 5:36.000000 6:0.000000
2  1:87.000000 2:70.000000 3:12.000000 4:28.000000 5:10.000000 6:0.000000
2  1:98.000000 2:55.000000 3:13.000000 4:17.000000 5:17.000000 6:0.000000
1  1:88.000000 2:62.000000 3:20.000000 4:17.000000 5:9.000000 6:0.500000
1  1:88.000000 2:67.000000 3:21.000000 4:11.000000 5:11.000000 6:0.500000
1  1:92.000000 2:54.000000 3:22.000000 4:20.000000 5:7.000000 6:0.500000
1  1:90.000000 2:60.000000 3:25.000000 4:19.000000 5:5.000000 6:0.500000
1  1:89.000000 2:52.000000 3:13.000000 4:24.000000 5:15.000000 6:0.500000
1  1:82.000000 2:62.000000 3:17.000000 4:17.000000 5:15.000000 6:0.500000
1  1:90.000000 2:64.000000 3:61.000000 4:32.000000 5:13.000000 6:0.500000
1  1:86.000000 2:77.000000 3:25.000000 4:19.000000 5:18.000000 6:0.500000
1  1:96.000000 2:67.000000 3:29.000000 4:20.000000 5:11.000000 6:0.500000
1  1:91.000000 2:78.000000 3:20.000000 4:31.000000 5:18.000000 6:0.500000
1  1:89.000000 2:67.000000 3:23.000000 4:16.000000 5:10.000000 6:0.500000
1  1:89.000000 2:79.000000 3:17.000000 4:17.000000 5:16.000000 6:0.500000
1  1:91.000000 2:107.000000 3:20.000000 4:20.000000 5:56.000000 6:0.500000
```

Εφόσον τα έχουμε στην κατάλληλη μορφή, τα επεξεργαζόμαστε με το SVM-scale για την σωστή κλιμάκωση των δεδομένων:

```
1 1:0.0526316 2:0.2 3:-0.456954 4:-0.428571 5:-0.821918 6:-1
2 1:0.0526316 2:-0.286957 3:-0.271523 4:-0.298701 5:-0.876712 6:-1
2 1:0.105263 2:-0.46087 3:-0.615894 4:-0.714286 5:-0.664384 6:-1
2 1:0.368421 2:-0.0434783 3:-0.602649 4:-0.506494 5:-0.787671 6:-1
2 1:0.157895 2:-0.182609 3:-0.89404 4:-0.402597 5:-0.965753 6:-1
2 1:0.736842 2:-0.443478 3:-0.880795 4:-0.688312 5:-0.917808 6:-1
1 1:0.210526 2:-0.321739 3:-0.788079 4:-0.688312 5:-0.972603 6:-0.95
1 1:0.210526 2:-0.234783 3:-0.774834 4:-0.844156 5:-0.958904 6:-0.95
1 1:0.421053 2:-0.46087 3:-0.761589 4:-0.61039 5:-0.986301 6:-0.95
1 1:0.315789 2:-0.356522 3:-0.721854 4:-0.636364 5:-1 6:-0.95
1 1:0.263158 2:-0.495652 3:-0.880795 4:-0.506494 5:-0.931507 6:-0.95
1 1:-0.105263 2:-0.321739 3:-0.827815 4:-0.688312 5:-0.931507 6:-0.95
1 1:0.315789 2:-0.286957 3:-0.245033 4:-0.298701 5:-0.945205 6:-0.95
1 1:0.105263 2:-0.0608696 3:-0.721854 4:-0.636364 5:-0.910959 6:-0.95
1 1:0.631579 2:-0.234783 3:-0.668874 4:-0.61039 5:-0.958904 6:-0.95
1 1:0.368421 2:-0.0434783 3:-0.788079 4:-0.324675 5:-0.910959 6:-0.95
1 1:0.263158 2:-0.234783 3:-0.748344 4:-0.714286 5:-0.965753 6:-0.95
1 1:0.263158 2:-0.026087 3:-0.827815 4:-0.688312 5:-0.924658 6:-0.95
1 1:0.368421 2:0.46087 3:-0.788079 4:-0.61039 5:-0.650685 6:-0.95
1 1:0.526316 2:0.617391 3:-0.907285 4:-0.272727 5:-0.958904 6:-0.95
```

Πλέον έχουμε τα αρχεία μας σε σωστή μορφή, έτοιμα για την διαδικασία της εκπαίδευσης του SVM.

Ανοίγουμε την κονσόλα cmd(command prompt) και μετακινούμαστε με την εντολή cd στον κατάλογο που βρίσκετε το πρόγραμμα libsvm.

Μετά, τρέχουμε το πρόγραμμα svm-train με το αρχείο που περιέχει τα δεδομένα τοποθετώντας τον διακόπτη –t 3 για να sigmoid kernel στην εκπαίδευση:

Svm-train.exe –t 3 liver-disorders_scale.train

Το SVM πλέον έχει εκπαιδευτεί και μπορεί να κάνει classification βάσει των δεδομένων που δέχτηκε. Έχει επίσης δημιουργηθεί ένα αρχείο που αποτελεί το μοντέλο για την πρόβλεψη με κατάληξη .model . Για να κάνουμε μια πρόβλεψη αρκεί να τρέξουμε svm-predict με παραμέτρους ένα αρχείο με τα δεδομένα που θέλουμε να γίνουν classified , το αρχείο με κατάληξη .model και ένα αρχείο εξόδου:

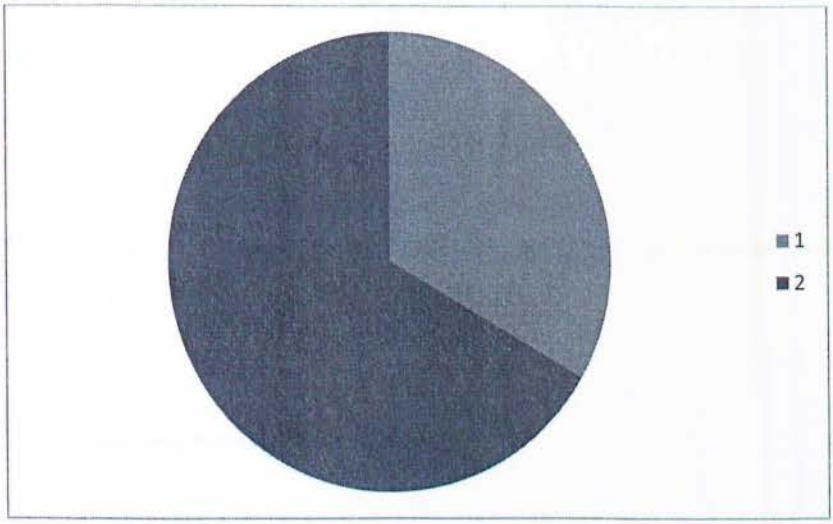
Svm-predict.exe liver-disorders.test liver-disorders_scale.train.model liver-disorders.out

Το αποτέλεσμα μας είναι το ποσοστό επιτυχίας του classification δηλαδή κατά πόσο καταφέραμε να κατηγοριοποιήσουμε τα δεδομένα που περιείχε το αρχείο liver-disorders.test .

Το αρχείο που δημιουργήθηκε (liver-disorders.out) περιέχει την κατηγοριοποίηση της κάθε κλάσης (γραμμής-ατόμου) σε 2 κατηγορίες:

1. Άτομα που παρουσιάζουν ηπατικές διαταραχές.

2. Άτομα που δεν παρουσιάζουν ηπατικές διαταραχές.



Πίνακας 4.3 Αποτελέσματα classification

A) Τύπος του SVM (SVM Type)

Όπως αναφέρθηκε ο τύπος που χρησιμοποιήθηκε για το δικό μας πρόβλημα είναι ο C-SVC που είναι ο καταλληλότερος τύπος SVM για να χρησιμοποιηθεί σε προβλήματα κατηγοριοποίησης (classification). Η αποδοτικότητα αυτού του τύπου επηρεάζεται από την παράμετρο cost. Για να ελέγξουμε αυτή τη παράμετρο έχουμε κρατήσει σταθερά τα δεδομένα εισόδου, το kernel type και τις παραμέτρους του. Πιο κάτω, ο πίνακας 4.3.1 απεικονίζει τις διάφορες τιμές που δοκιμάστηκαν για την παράμετρο cost κρατώντας τα υπόλοιπα σταθερά. Παρατηρούμε ότι το κόστος στον τύπο του SVM επηρεάζει αισθητά την απόδοση του δικτύου. Δοκιμάζοντας διάφορες τιμές του cost σε διαφορετικούς kernel types κάθε φορά έχω παρατηρήσει ότι δεν υπάρχει μια βέλτιστη τιμή του cost που να παράγει τα καλύτερα αποτελέσματα σε κάθε kernel type. Ο κάθε kernel type επηρεάζεται διαφορετικά με την τιμή του κόστους και για κάθε ένα υπάρχει διαφορετική βέλτιστη τιμή του cost.

Cost	1	5	10	20
Accuracy %	58,8	58,8	58,8	58,8

Πίνακας 4.3.1 Πίνακας με τα ποσοστά επιτυχίας για διάφορες τιμές της παραμέτρου cost για τον τύπο των SVM C-SVC

Β) Τύπος του Kernel (Kernel Type)

Δεδομένου ότι εμείς θέλαμε έναν γραμμικό τρόπο πρόβλεψης ο καταλληλότερος τύπος πυρήνα ήταν ο sigmoid.

Στον πιο κάτω πινάκα ελέγξαμε τον τύπο radial basis για πέντε διαφορετικές τιμές της παραμέτρου gamma και για SVM type cost = 5. Παρατηρούμε ότι όταν το gamma πάρει μικρή τιμή αποδίδει πιο καλά.

Gamma	0.2	1	2.5	5	10
Accuracy%	58,8	52,3	46,5	46,2	44

Πίνακας 4.3.2 Πίνακας με τα ποσοστά επιτυχίας για διάφορες τιμές της παραμέτρου cost για τον Τύπος του Kernel sigmoid.

Γ) Απόδοση δικτύου με τις καλύτερες τιμές

Δεδομένου ότι έχουμε επιλέξει τον τύπο των SVM, C-SVC, και τον τύπο του Kernel , sigmoid, είχαμε να πειραματιστούμε μόνο με τις παραμέτρους cost και gamma. Στους παρακάτω πίνακες εμφανίζονται τα ποσοστά επιτυχίας επί τις εκατό για τις διάφορες τιμές αυτών των μεταβλητών:

	Gamma = 0.2	Gamma = 1	Gamma = 2.5	Gamma = 5	Gamma = 10
Cost=1	57,2	58,1	45,2	45,2	45,2
Cost=5	58,8	58,8	45,2	45,2	45,2
Cost=10	57,2	58	45,2	45,2	45,2
Cost=20	56,3	56,4	45,2	45,2	45,2

Πίνακας 4.3.3 Πίνακας με τα ποσοστά επιτυχίας για διάφορες τιμές της παραμέτρου cost και gamma για τον Τύπος του Kernel sigmoid και τον τύπο των SVM, C-SVC

Δ) Συμπεράσματα

Από τους πιο πάνω πίνακες διαπιστώνουμε ότι τα καλύτερα αποτελέσματα τα παίρνουμε για cost = 5 και gamma= 0,2. Επίσης αντιλαμβανόμαστε πως ο παράγοντας κόστους δεν επηρεάζει το αποτέλεσμα τόσο πολύ όσο ο παράγοντας gamma

Συμπέρασμα :

Αυτή η πτυχιακή εργασία επικεντρώθηκε στη μέθοδο των Support Vector Machines με την προϋπόθεση ότι μπορούν να επιφέρουν καλύτερα αποτελέσματα έναντι άλλων τεχνικών. Αυτή η μέθοδος στηρίζεται σε πολλές παραμέτρους ανάμεσα τους και η δομή των δεδομένων. Τα SVMs μπορούν να επιφέρουν καλύτερα αποτελέσματα σε σύνολα δεδομένων που αποτελούνται από μεγάλο πλήθος δεδομένων και χαρακτηριστικών. Σε αυτή τη πτυχιακή εργασία, τα ποσοστά ακρίβειας και η σωστή κατηγοριοποίηση πέτυχαν ικανοποιητικά ποσοστά αν και μία πιθανή επανεκτέλεση της εργασίας με περισσότερα δεδομένα, ίσως επέφερε ακόμα πιο ψηλά ποσοστά επιτυχίας.

Η εκπαίδευση των SVM είναι επιβλεπόμενη (supervised learning) και γίνεται εφόσον δοθεί στον SVM όλο το σύνολο εκπαίδευσης. Αρχικά οι αλγόριθμοι SVM αναπτύχθηκαν με σκοπό των διαχωρισμό προτύπων (classification) αλλά αργότερα εφαρμόστηκαν και σε προβλήματα προσέγγισης συναρτήσεων (regression). Οι προβλέψεις που πραγματοποιούνται με τη βοήθεια αυτών καλύπτουν ένα τεράστιο εύρος, καθώς δεν είναι αναγκαίο να κατασκευάζεται κάθε φορά εκ νέου το δίκτυο όταν θέλουμε να εφαρμόσουμε διαφορετικά δεδομένα σε αυτά. Μπορούμε απλά να τοποθετήσουμε τα καινούρια μας δεδομένα χωρίς κάποια άλλη παραμετροποίηση εφόσον οι πυρήνες (kernels) που χρησιμοποιούνται έχουν εφαρμογή σε κάθε είδους πρόβλημα.

Ένα όμως από τα μειονεκτήματα του SVM είναι ο τρόπος με τον οποίο γίνεται η εκμάθηση: Αφού τελειώσει η εκπαίδευση, δεν είναι δυνατό να προστεθούν νέα σύνολα εκπαίδευσης στη μηχανή. Συνεπώς, αν επιθυμούμε να προσθέσουμε νέα γνώση στον SVM, θα πρέπει να γίνει η εκπαίδευση από την αρχή, διαδικασία η οποία είναι πολλές φορές χρονοβόρα.

Βιβλιογραφία :

1. Schölkopf, B.; Smola, A.: Learning with Kernels. MIT Press (2002)

2. Christianini N.; Shawe-Taylor, J.: An Introduction to Support Vector Machines and other kernel-based methods. Cambridge University Press (2000)

3. Trafalis, T. B.; Ince, H.: Support Vector Machine for Regression and Applications to Financial Forecasting. In: International joint conference on neutral networks, vol. 6, pp. 348--353 (2000)

4. Schölkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC: Estimating the support of a high-dimensional distribution. Neural Computation 2001, 13(7):1443-1471.

5. Chih-Chung Chang, Chih-Jen Lin: LIBSVM, a library for support vectormachines. 2001 http://www.csie.ntu.edu.tw/~cjlin/libsvm.

6. Foundamentals of Artificial Neural Networks (M.Hassoun)

7. Neural Networks: Acomprehensive Foundation (S.Haykin)

8. Introduction to the Theory of Neural Computation (J.Hertz,A.Krogh,R.Palmer)

9. Τεχνητά Νευρωνικά Δίκτυα: Θεωρία και Εφαρμογές (Γ.Ρίζος)

10. UCI, machine learning repository. Center for machine Learning and Intelligence Systems. http://archive.ics.uci.edu