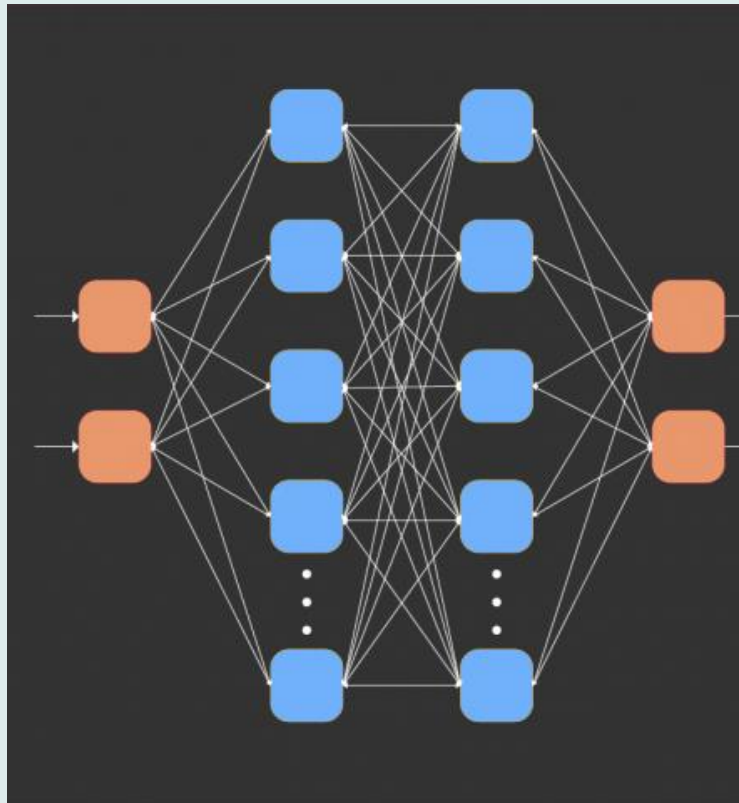




ΑΝΑΓΝΩΡΙΣΗ
ΠΡΟΤΥΠΩΝ

ΕΡΓΑΣΙΑ 7^Η

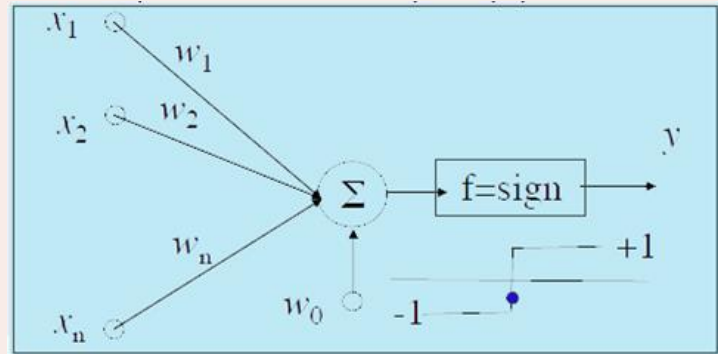
ARTIFICIAL NEURAL NETWORKS



Καραπέπερα Ελπίδα
57423

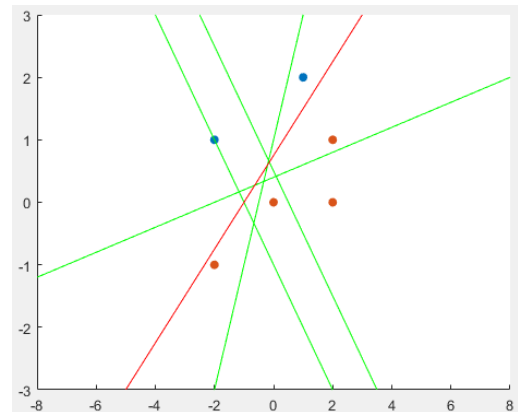
Linear Perceptron

Είναι ένα μονοστρωματικό δίκτυο που αποτελείται από N εισόδους και M νευρώνες. Κάθε νευρώνας συνδέεται με όλες τις εισόδους με συναπτικές συνδέσεις. Κάθε είσοδος που συνδέεται σε κάθε νευρώνα πολλαπλασιάζεται με ένα βάρος. Οι νευρώνες είναι τύπου McCulloch-Pitts με Hard limiter και προσαρμοσμένο κατώφλι

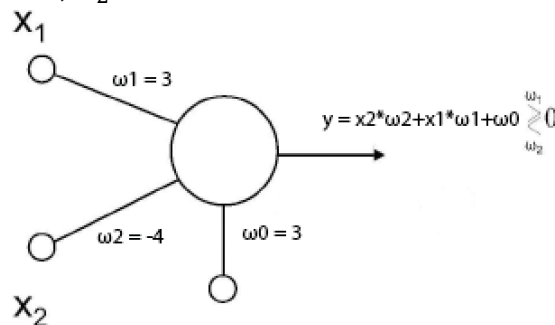


ενεργοποίησης $y_i = \text{sign}(\sum_j w_{ij}x_j + w_{0i})$. Το αποτέλεσμα κάθε εξόδου είναι ανεξάρτητο από τα αποτελέσματα όλων των υπόλοιπων. Το αποτέλεσμα μίας εξόδου αναπαρίσται στο παραπάνω σχήμα.

Για τη λύση ενός προβλήματος με την μέθοδο αυτή, θα πρέπει οι κατηγορίες στις οποίες χωρίζονται τα δεδομένα να είναι δύο και γραμμικά διαχωρίσιμες. Όπως και στις προηγούμενες μεθόδους που μελετήθηκαν, προσπαθούμε και εδώ να βρούμε ένα διάνυσμα ω , τα στοιχεία του οποίου θα μπορούν να χρησιμοποιηθούν ως συντελεστές στη συνάρτηση διαχωρισμού $g(x)$, η οποία θα περιγράφει το διαχωριστικό υπερεπίπεδο που θα χωρίζει τα δεδομένα. Για την εύρεση του διανύσματος αυτού χρησιμοποιήθηκε ο αλγόριθμος Perceptron από την εργασία 5.



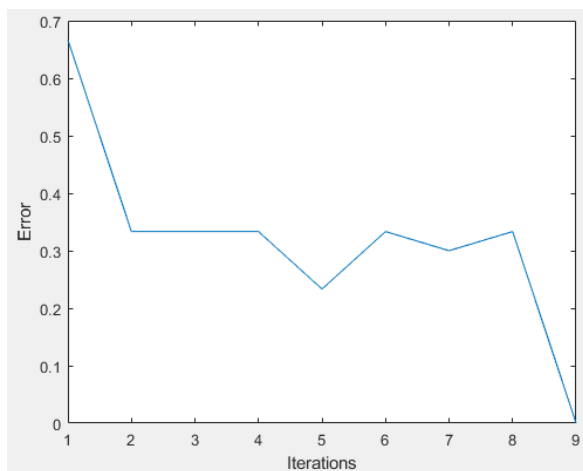
Όπως φαίνεται και στη διπλανή εικόνα, τα στοιχεία χωρίστηκαν επιτυχώς. Τα βάρη είναι τα εξής: $\omega_0 = 3, \omega_1 = 3, \omega_2 = -4$.



IRIS Data Set με Ταξινομητή Πολυστρωματικά Perceptron

A. Linear Perceptron

Χρησιμοποιώντας το Linear Perceptron για το διαχωρισμό κάθε κατηγορίας από τις άλλες (One vs all) τα errors που επιτεύχθηκαν στα validation batches σε κάθε epoch φαίνονται στις παρακάτω τρεις εικόνες:

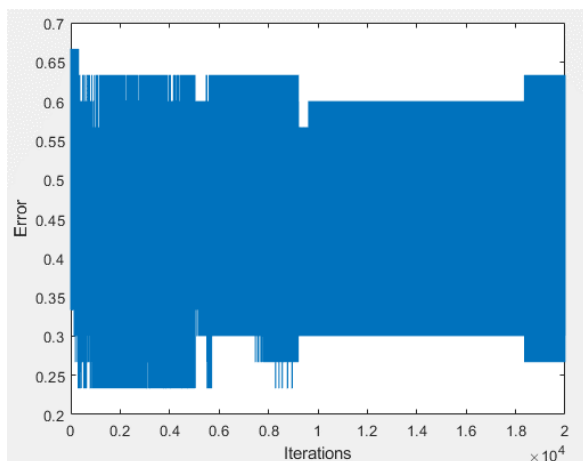


Iris Setosa

Φαίνεται ότι το Iris Setosa είναι γραμμικά διαχωρίσιμο από τα άλλα δύο, καθώς σε εννιά μόλις epochs το error είναι πλέον 0.

Ο μέσος όρος των errors στο test batch είναι επίσης 0.

Η ταξινόμηση είναι επιτυχής.

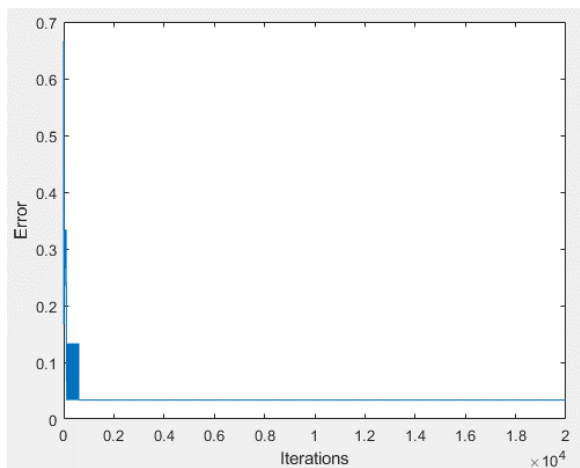


Iris Versicolor

Φαίνεται ότι το Iris Versicolor δεν είναι γραμμικά διαχωρίσιμο από τα άλλα δύο, καθώς μετά από 20,000 epochs το error είναι ακόμη πολύ μεγάλο.

Ο μέσος όρος των errors στο test batch είναι 34 %.

Η ταξινόμηση δεν είναι επιτυχής.



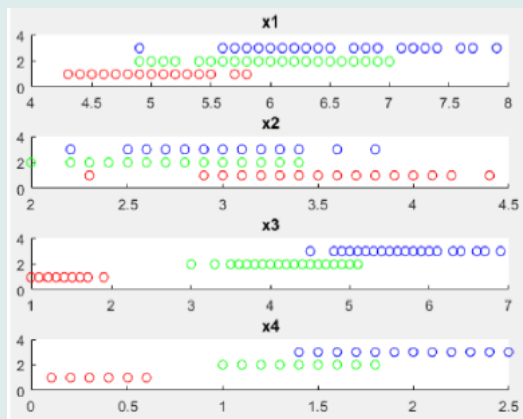
Iris Virginica

Φαίνεται ότι το Iris Virginica δεν είναι γραμμικά διαχωρίσιμο από τα άλλα δύο, καθώς μετά από 20,000 epochs το error είναι μεν μικρό, αλλά παραμένει διάφορο του 0.

Ο μέσος όρος των errors στο test batch είναι 5.334%.

Η ταξινόμηση είναι μεν αρκετά καλή, δεν είναι όμως επιτυχής, αφού δεν διαχωρίζονται εντελώς τα λουλούδια της επιθυμητής κατηγορίας.

Οι προσομοιώσεις έτρεξαν 5 φορές για κάθε ερώτημα και σαν σφάλμα χρησιμοποιήθηκε ο μέσος όρος των 5 σφαλμάτων για κάθε προσομοίωση. Αυτό έγινε γιατί τα δεδομένα του train, test και validation επιλέχθηκαν τυχαία από το σετ των data και άρα κάθε φορά που τρέχουμε την προσομοίωση έχουμε διαφορές.



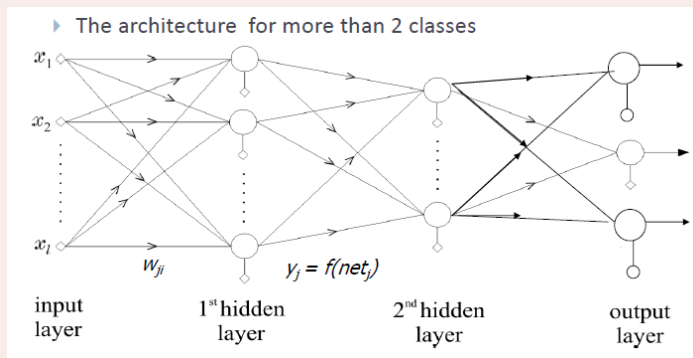
Συγκρίνοντας τις τρεις ταξινομήσεις βλέπουμε ότι η μοναδική κατηγορία που είναι γραμμικά διαχωρίσιμη είναι η Iris Setosa.

Αυτό μπορούμε να το παρατηρήσουμε και από τον διπλανό πίνακα, όπου φαίνεται ο διαχωρισμός των τριών λουλουδιών με βάση το κάθε χαρακτηριστικό ξεχωριστά. Παρατηρούμε ότι μόνο για τα χαρακτηριστικά x3 και x4 η μόνη κατηγορία που είναι γραμμικά διαχωρίσιμη είναι η κόκκινη κατηγορία (Iris Setosa).

Multi-Layer Perceptron

Ουσιαστικά, το πρόβλημα αντιμετωπίζεται σε πολλές (multi) διαδοχικές φάσεις. Στο ερώτημα Β) χρησιμοποιούνται δύο φάσεις, ενώ στο Γ) τρεις φάσεις. Έχουμε τόσες γραμμές απόφασης ($g_i(x)$) όσες και φάσεις. Στην πρώτη φάση υπολογίζω τη θέση του κάθε σημείου σε σχέση με την $g_1(x)$, στη δεύτερη συνδυάζω τα

αποτελέσματα της προηγούμενης φάσης και βρίσκω τη θέση του κάθε σημείου σε σχέση με αμφότερες τις συναρτήσεις (την $g_1(x)$ και την $g_2(x)$) και η διαδικασία επαναλαμβάνεται ίδια για κάθε $g_i(x)$. Η υλοποίηση αυτή επιτυγχάνεται με τη χρήση τόσων Perceptron όσες είναι και οι φάσεις. Οι έξοδοι που έχω είναι όσα είναι και τα Perceptron. Αυτές είναι οι σχετικές θέσεις του διανύσματος εισόδου σε σχέση με κάθε μία από τις δύο συναρτήσεις $g_i(x)$. Στη δεύτερη φάση χρησιμοποιούμε τις εξόδους και με βάση αυτές και τη χρήση ενός τρίτου Perceptron βγάζουμε ένα καινούριο συμπέρασμα. Η ίδια διαδικασία επαναλαμβάνεται για όσα επίπεδα ορίσουμε. Δύο φάσεις περιέχουν ένα hidden layer, τρεις περιέχουν δύο κλπ.



Backpropagation

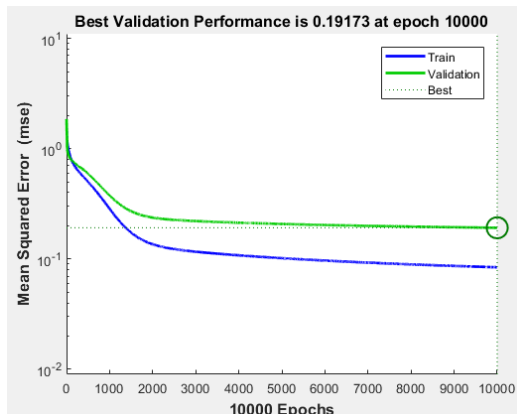
$$f(x) = \frac{1}{1 + \exp(-ax)}$$

Σε κάθε layer νευρώνων υπολογίζεται το gradient και επαναπροσδιορίζονται τα βάρη που ορίστηκαν. Χρησιμοποιούμε πλέον τη logistic function και βρίσκουμε το σφάλμα της (τη διαφορά του επιθυμητού αποτελέσματος με αυτό που πραγματικά παίρνουμε). Έχουμε λοιπόν errors τα οποία δε μπορούμε να εξαλείψουμε, προσπαθούμε όμως να ελαχιστοποιήσουμε. Κάθε φορά που φτάνουμε σε κάποιο layer υπολογίζουμε το gradient και ενημερώνουμε εκ νέου όλα τα προηγούμενα βάρη, ξεκινώντας από το κοντινότερο και προχωρώντας προς το μακρινότερο επίπεδο.

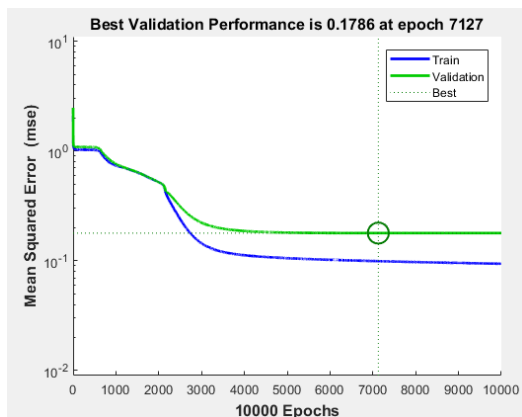
B. Multi-Layer Perceptron with backpropagation – 1 hidden layer

Αρχικά, να διευκρινιστεί ότι χρησιμοποιήθηκε στο hidden layer η συνάρτηση tansig, η οποία προσεγγίζει τη σιγμοειδή συνάρτηση, και στο τελικό layer (output layer) η απλή γραμμική (purelin).

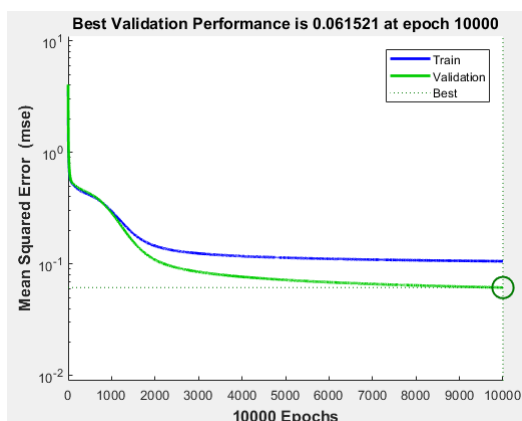
Βάζοντας 2, 5 και 10 νευρώνες στο hidden layer παρατηρήθηκαν τα εξής αποτελέσματα:



Το σφάλμα χρησιμοποιώντας 2 νευρώνες ήταν 10.002%.



Το σφάλμα χρησιμοποιώντας 5 νευρώνες ήταν 16.668%.

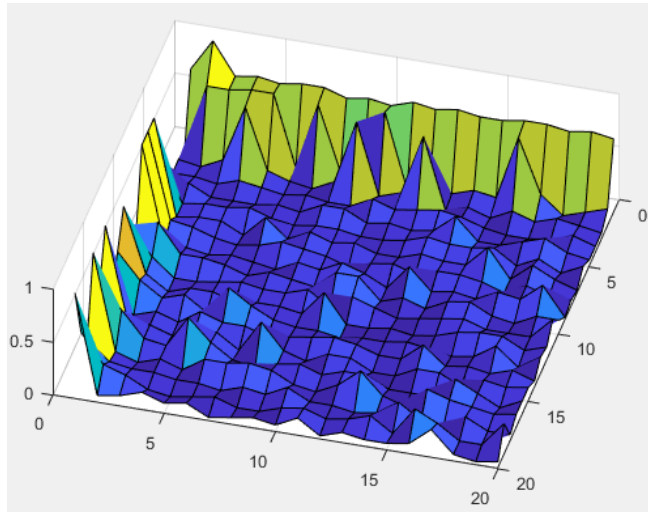


Το σφάλμα χρησιμοποιώντας 10 νευρώνες ήταν 19.334%.

Παρατηρούμε ότι αυξάνοντας τους νευρώνες αυξάνεται το σφάλμα. Αυτό συμβαίνει γιατί τα δεδομένα εφόσον μπορούν να διαχωριστούν αρκετά καλά σε ένα επίπεδο με 2 νευρώνες, προσθέτοντας επιπλέον νευρώνες απλώς

επιβαρύνουμε το network με περισσότερες πράξεις χωρίς να βοηθάμε καθόλου στο διαχωρισμό των δειγμάτων.

C. Multi-Layer Perceptron with backpropagation - 2 hidden layers



Έγιναν προσομοιώσεις για 1 έως 20 νευρώνες σε κάθε ένα από τα δύο layers. Παρατηρήθηκε από τα αποτελέσματα ότι το καλύτερο error που μπορεί να επιτευχθεί είναι με 2 νευρώνες σε κάθε hidden layer το 3.33%.

Από τα ερωτήματα A, B και C συμπεραίνουμε ότι δεν απαιτείται απαραίτητα μεγάλος αριθμός νευρώνων για την επίλυση του προβλήματος διαχωρισμού των λουλουδιών IRIS. Βοηθάει, όμως, στη λύση του προβλήματος ο διαχωρισμός τους σε επίπεδα. Όσο περισσότερα layers βάζω στο κύκλωμα, τόσο πιο επιτυχής είναι και η ταξινόμηση που πετυχαίνω. Αυτό αποδεικνύεται ως εξής: Στο ερώτημα A φαίνεται ότι ο διαχωρισμός των IRIS στις τρεις κατηγορίες έχει σφάλματα 0%, 36.67% και 5.334% για την ταξινόμηση των λουλουδιών σε κάθε μία από τις τρεις κατηγορίες. Μέσος όρος σφάλματος βγαίνει 14%. Με 1 hidden layer πήγε στο 10%, ενώ με 2 hidden layers έφτασε το 3.33%.