



Προγραμματιστική άσκηση 1 – Backpropagation XOR

Όνομα: Μιχαήλ-Παναγιώτης Μπόφος

Ταυτότητα: ΑΟ082689

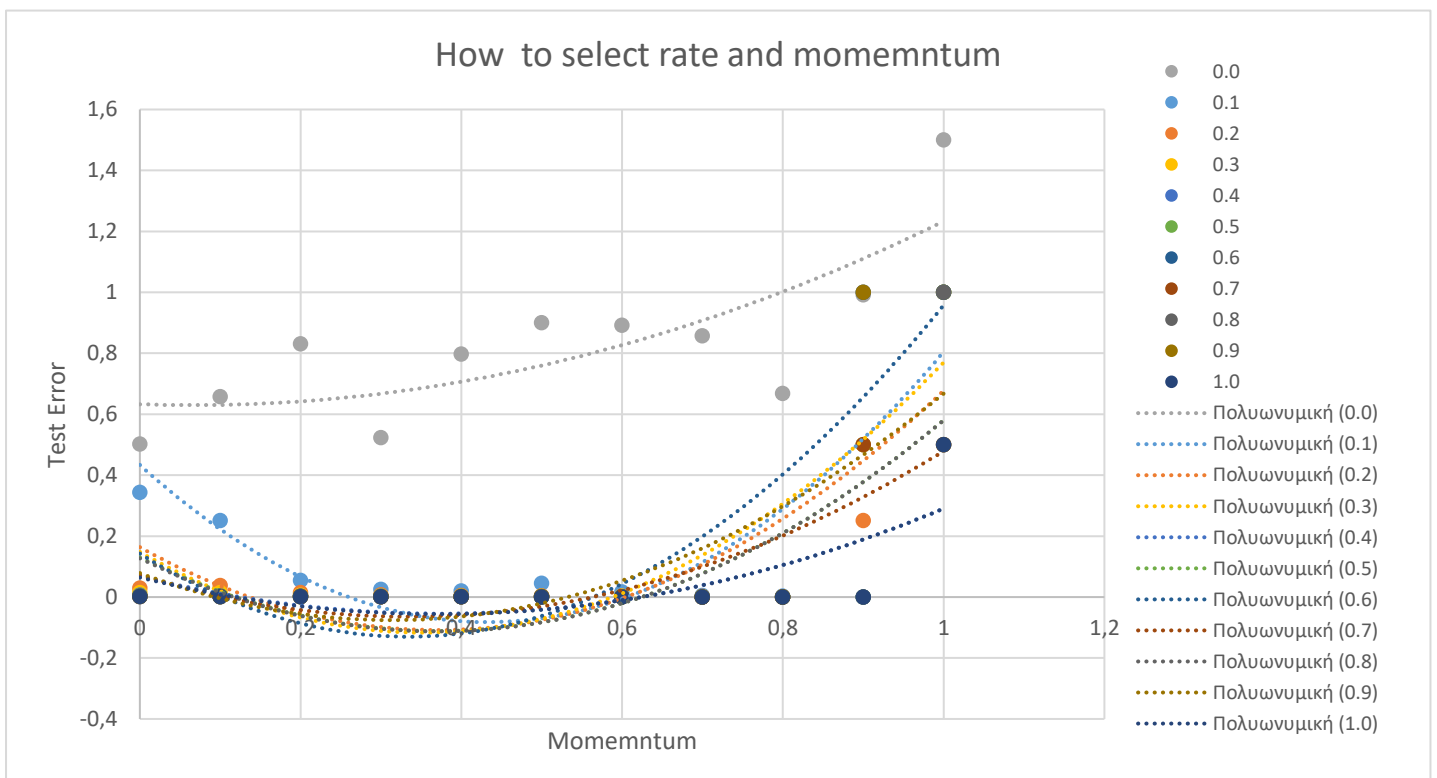
Ανάλυση επιλογών για το Νευρωνικό Δίκτυο

Τοπολογία Δικτύου

Καθώς το πρόβλημα του XOR gate είναι αρκετά απλό, αποφάσισα να δημιουργήσω ένα απλό νευρωνικό δίκτυο. Το δίκτυό μας αποτελείται από ένα επίπεδο εισόδου, ένα κρυφό επίπεδο και ένα επίπεδο εξόδου και η συνδεσιμότητά του είναι πλήρης. Όπως και μια πύλη XOR έτσι και το δίκτυο μας έχει δύο inputs. Για το κρυφό επίπεδο θεωρητικά αρκούν δύο κόμβοι (άρα και δύο γραμμές απόφασης) που θα κάλυπταν τα επιθυμητά σημεία, ωστόσο επέλεξα να χρησιμοποιήσω έξι νευρώνες που σύμφωνα με τις δοκιμές μου δίνουν πιο σταθερά αποτελέσματα (λογικό αν σκεφτούμε το πρόβλημα των τοπικών ελαχίστων). Επίσης, επειδή η έξοδος της XOR είναι ένα bit, αρκεί μόνο ένας νευρώνας εξόδου. Οι αρχικές τιμές των βαρών κυμαίνονται από το πλην ένα μέχρι το ένα, το input των bias είναι ίσο με το ένα. Η υλοποίηση του backpropagation που επέλεξα ήταν αυτή με το δεύτερο forward pass για την αλλαγή των βαρών. Τέλος, οι επιλογές του learning rate και του momentum θα ερευνηθούν πιο κάτω.

Επιλογή ρυθμού μάθησης και αδράνειας

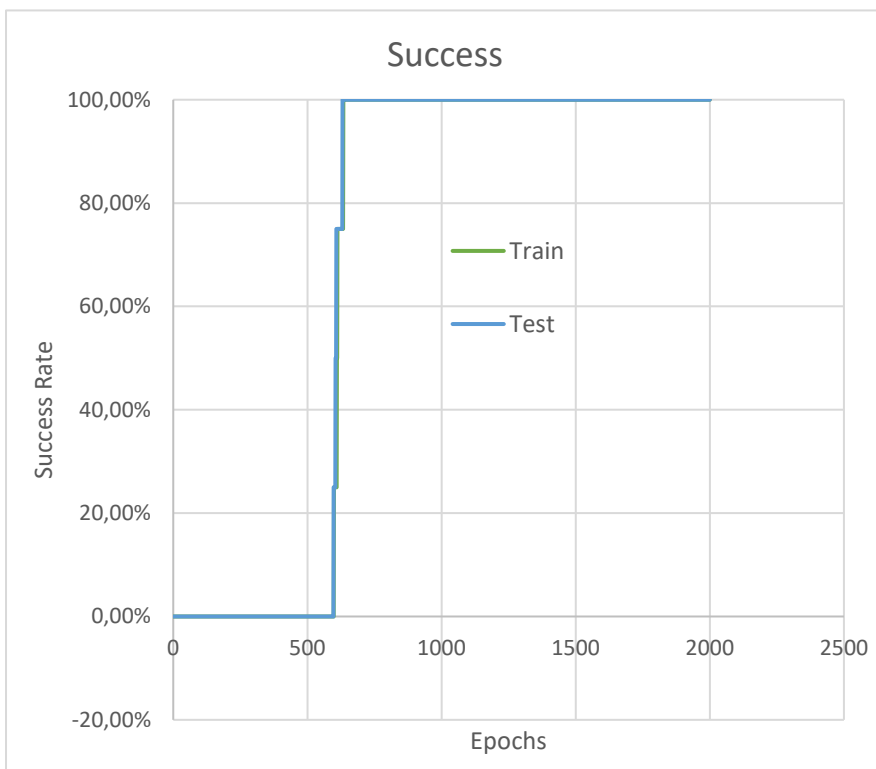
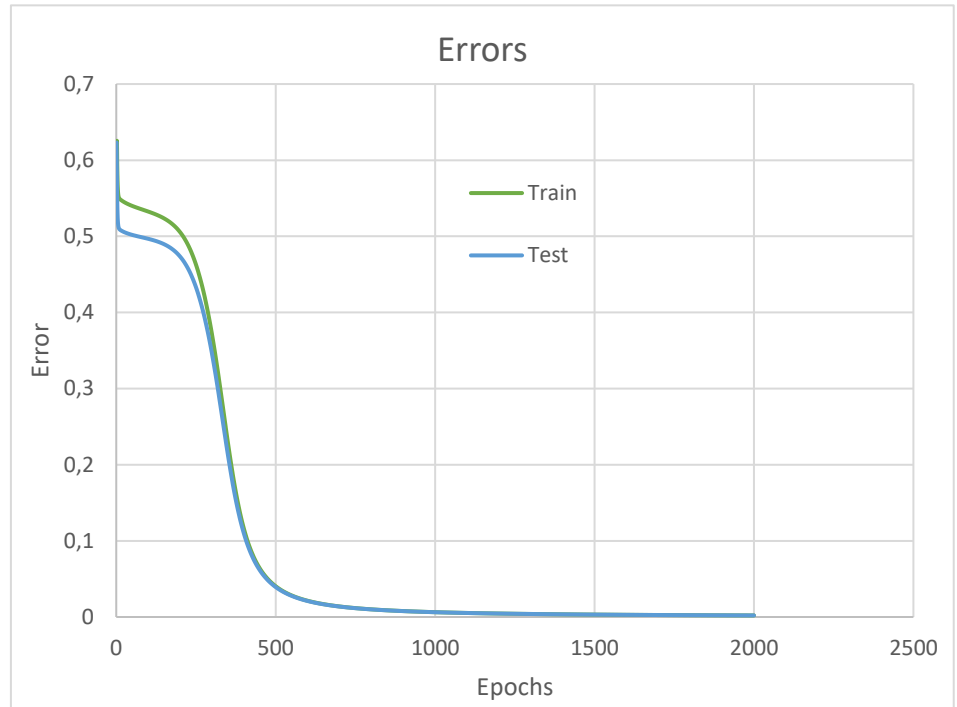
Από την παρακάτω γραφική βλέπουμε πως η ιδανική τιμή αδράνειας κυμαίνεται από 0,3 με 0,5 ενώ για τον ρυθμό μάθησης όλες οι τιμές μεγαλύτερες του μηδενός λειτουργούν σωστά, νοουμένου ότι η αδράνεια βρίσκεται ανάμεσα του 0,3 και του 0,5, τις δύο μεταβλητές καθώς ελαχιστοποιείται το σφάλμα. Η αρνητική τιμή που φαίνεται στο γράφημα είναι αποτέλεσμα της εκτιμώμενης καμπύλης (οι πραγματικές τιμές είναι θετικές) της οποίας στην περίπτωσή μας μας ενδιαφέρει μόνο η συμπεριφορά της και όχι η τιμή αυτή καθ' αυτή.



Επιλογή αριθμού εποχών

Στην εξής γραφική βλέπουμε ότι θεωρητικά το σφάλμα μας ελαχιστοποιείται κοντά στις χίλιες εποχές. Ωστόσο παρατήρησα ότι οι δύο χιλιάδες εποχές δίνουν μια σταθερότητα τόσο στο σφάλμα όσο και στο ποσοστό επιτυχίας, με αποτέλεσμα να επιλέξω τις δύο χιλιάδες εποχές.

Σημείωση: Το σφάλμα μας συνήθως φτάνει την ελάχιστη τιμή του και με κάτω από χίλιες εποχές αλλά ούτε είναι σταθερό ούτε το τελικό του αποτέλεσμα είναι πάντα ορθό.



Στη γραφική παράσταση της επιτυχίας παρατηρούμε ότι υπάρχει σταθερή άνοδος και ότι αν και φτάνουμε νωρίς στο 100% στην αντίστοιχη εποχή (~600^η) το σφάλμα μας δεν έχει φτάσει στο ελάχιστο. Το συμπέρασμα στο οποίο καταλήγουμε είναι ότι το ποσοστό επιτυχίας από μόνο του δεν αρκεί για να κρίνουμε το δίκτυό μας στο πρόβλημα του XOR gate.

Σημείωση: Για να καθορίσουμε αν η έξοδος μας είναι μηδέν ή ένα χρησιμοποιήσαμε το κανόνα: αν η αναμενόμενη έξοδος είναι μηδέν οι αποδεκτές τιμές είναι κάτω από το 0,1, αντίστοιχα αν η αναμενόμενη τιμή είναι ένα, οι αποδεκτές τιμές είναι πάνω από 0,9.

Ανάλυση αποτελεσμάτων

Μετά από πολλές δοκιμές συνδυασμών τα συμπεράσματα μας είναι κάπως πιο ξεκάθαρα. Ενώ όπως εξηγήθηκε πιο πάνω μια μετριοπαθής επιλογή είναι πιο ασφαλής, παρατηρήσαμε πως αν ο ρυθμός μάθησης είναι μεγάλος και η αδράνεια μικρή, το δίκτυο ελαχιστοποιεί το σφάλμα πολύ πιο γρήγορα από τις δύο χιλιάδες εποχές (Παρ' όλα αυτά με δύο χιλιάδες εποχές το σφάλμα παραμένει μικρότερο αλλά για πολύ μικρή διαφορά). Ως γενικό σχόλιο, παρατηρούμε ότι στη training phase το σφάλμα είναι υψηλότερο και η επιτυχία είναι μικρότερη από τη testing phase, φαινόμενο λογικό καθώς μετά την εκπαίδευση το δίκτυο μας είναι ένα βήμα πιο κοντά στην «ιδανική» του κατάσταση.

Input layer: 2 Hidden layer: 6 Output layer: 1 Max epochs: 700 Learning rate: 0.9 Momenmtum: 0.2 Train file: training.txt Test file: test.txt			
INPUT 1	INPUT 2	TARGET	REAL
0.0	0.0	0.0	0.0
0.0	1.0	1.0	1.0
1.0	0.0	1.0	1.0
1.0	1.0	0.0	0.0

Input layer: 2 Hidden layer: 6 Output layer: 1 Max epochs: 2000 Learning rate: 0.4 Momenmtum: 0.4 Train file: training.txt Test file: test.txt			
INPUT 1	INPUT 2	TARGET	REAL
0.0	0.0	0.0	0.0
0.0	1.0	1.0	1.0
1.0	0.0	1.0	1.0
1.0	1.0	0.0	0.0

