

## Multilayer Perceptron:

Σε αυτό το project θα επιχειρήσουμε να εκπαιδεύσουμε ένα σύστημα νευρώνων ώστε να είναι σε θέση να αποφασίσει σε ποια κατηγορία ανήκει ένα σημείο από το καρτεσιανό τετράγωνο:  $[-1,-1]$  ,  $[1,-1]$ ,  $[1,1]$ ,  $[-1,1]$ .

Το σύστημα θα εκπαιδευτεί με 4000 σημεία, τα οποία θα χρησιμοποιήσει αρκετές φορές κατά την εκπαίδευσή του.

Θα κληθεί να αποφασίσει για 4000 καινούρια σημεία, σε ποια από τις κατηγορίες ανήκει.

Η γενική εικόνα για την λειτουργία του είναι η εξής:

A) Το σύστημα ξεκινάει ζητώντας από τον χρήστη τις βασικές παραμέτρους που θα χρειαστεί να αρχικοποιήσει, ώστε να μπορέσει να κατασκευάσει την αρχιτεκτονική του. (αριθμό κρυφών επιπέδων, αριθμό κατηγοριών, συνάρτηση ενεργοποίησης των νευρώνων , αριθμό νευρώνων στα κρυφά επίπεδα)

B) Το σύστημα παράγει αυτόματα δύο αρχεία , τα οποία περιλαμβάνουν τα σημεία εκπαίδευσης και τα σημεία ελέγχου αντίστοιχα. Κατόπιν, καλεί το πρώτο αρχείο με τα σημεία εκπαίδευσης για να ξεκινήσει την προπόνηση.

Ο αλγόριθμος προπόνησης ακολουθεί την εξής σειρά:

- 1) Παίρνει ως είσοδο ένα σημείο  $(x,y)$
- 2) Περνάει την είσοδο στα επόμενα επίπεδα (κρυφά επίπεδα) μέσω των νευρώνων - αλγόριθμος forwardPass
  - 2.1) Ο αλγόριθμος forward Pass , μέσω επεξεργασίας της ενδιάμεσης πληροφορίας, καταλήγει στην παραγωγή πληροφορίας στο επίπεδο εξόδου.

3) Εφόσον το σύστημα παρήγαγε κάποια/ες τιμή/ες εξόδου, προβαίνει στο επόμενο σημαντικό σκέλος , όπου θα ελέγξει την διαφορά των τιμών εξόδου, με τις αναμενόμενες πραγματικές τιμές.

4) Εδώ τελειώνει και η προώθηση πληροφορίας προς τα εμπρός και ξεκινάει η προώθηση πληροφορίας προς τα πίσω(backpropagation). Γνωρίζοντας το μέγεθος του σφάλματος , προωθεί την πληροφορία αυτή προς τα προηγούμενα επίπεδα και αποφασίζει για κάθε σύνδεση μεταξύ νευρώνων(βάρη) , για τι ποσοστό του τελικού σφάλματος φέρει ευθύνη. Μόλις κάνει αυτόν τον υπολογισμό για κάθε βάρος ανάμεσα στους νευρώνες, ενημερώνει την τιμή τους(gradient descend) , ώστε την επόμενη φορά το σφάλμα που θα παραχθεί από το κάθε βάρος να είναι μικρότερο.

Μετά από όλες αυτές τις διαδικασίες, το σύστημα θα τις επαναλάβει για νέο σημείο, από τα σημεία εκπαίδευσης.

Η εκπαίδευση τελειώνει, αναλόγως με τον αριθμό των εποχών (μία εποχή = εκπαίδευση του συστήματος για κάθε σημείο του συνόλου εκπαίδευσης, 1 φορά το καθένα) που έχει ορίσει ο χρήστης, ή όταν το συνολικό σφάλμα μίας εποχής έχει διαφορά με το συνολικό σφάλμα της προηγούμενης εποχής ένα κατώτατο κατώφλι που έχει ορίσει ο χρήστης.

### **ΟΛΕΣ ΟΙ ΜΕΤΡΗΣΕΙΣ ΠΑΡΑΚΑΤΩ ΕΙΝΑΙ ΓΙΑ 1000 ΕΠΟΧΕΣ**

Για το πρόβλημα Π1 (2 κρυφά επίπεδα) της άσκησης 1, το σύστημα παράγει τα εξής αποτελέσματα:

(με (\_\_,\_\_) θα συμβολίζονται ο αριθμός νευρώνων στο κρυφό επίπεδο 1 και στο κρυφό επίπεδο 2. Π.χ. (1,3) = 1 νευρώνας στο πρώτο κρυφό επίπεδο και 3 νευρώνες στο 2)

**\*\* Για συνάρτηση ενεργοποίησης TANH \*\***

Π.Ε. = Ποσοστό επιτυχίας

Για τις ακραίες τιμές  $\rightarrow (1,1)$  Π.Ε. = 0%

Για τις τιμές  $\rightarrow (1,5)$  = 0%

Για τις τιμές  $\rightarrow (3,5)$  = 20%

Για τις τιμές  $\rightarrow (5,2)$  = 0%

Για τις τιμές  $\rightarrow (5,5)$  = 23%

Για τις τιμές  $\rightarrow (7,4)$  = 50%

Για τις τιμές  $\rightarrow (9,4)$  = 35%

Για τις τιμές  $\rightarrow (7,7)$  = 68%

Για τις τιμές  $\rightarrow (9,9)$  = 66%

Για τις τιμές  $\rightarrow (6,8)$  = 46%

Για τις τιμές  $\rightarrow (6,6)$  = 59%

Για τις τιμές  $\rightarrow (10,6)$  = 82%

Για τις τιμές  $\rightarrow (10,5)$  = 71%

Για τις τιμές  $\rightarrow (10,7)$  = 83%

Για τις τιμές  $\rightarrow (10,10)$  = 85%

Φαίνεται πως κάπου εδώ (10,10) το MLP παίρνει την καλύτερη τιμή του.

**\*\* Για συνάρτηση ενεργοποίησης SIGMOID \*\***

Π.Ε. = Ποσοστό επιτυχίας

Για τις ακραίες τιμές  $\rightarrow (1,1)$  Π.Ε. = 0%

Για τις τιμές  $\rightarrow (3,5) = 37\%$

Για τις τιμές  $\rightarrow (5,2) = 13\%$

Για τις τιμές  $\rightarrow (5,5) = 64\%$

Για τις τιμές  $\rightarrow (7,4) = 78\%$

Για τις τιμές  $\rightarrow (9,4) = 74\%$

Για τις τιμές  $\rightarrow (7,7) = 82\%$

Για τις τιμές  $\rightarrow (9,9) = 86\%$

Για τις τιμές  $\rightarrow (6,8) = 71\%$

Για τις τιμές  $\rightarrow (6,6) = 73\%$

Για τις τιμές  $\rightarrow (10,6) = 88\%$

Για τις τιμές  $\rightarrow (10,5) = 88\%$

Για τις τιμές  $\rightarrow (10,7) = 89\%$

Για τις τιμές  $\rightarrow (10,8) = 90\%$

Φαίνεται πως κάπου εδώ  $(10,8)$  το σύστημα παίρνει την καλύτερη τιμή του.

Ανάμεσα στις δύο συναρτήσεις ενεργοποίησης, παρατηρούμε ότι το σύστημα μας αντιδράει καλύτερα στην sigmoid.

Η δεύτερη παρατήρηση είναι πως και στις δύο περιπτώσεις το σύστημα φαίνεται να παίρνει τις καλύτερες επιλογές όταν στο πρώτο κρυφό επίπεδο υπάρχουν 10 κρυφοί νευρώνες και στο δεύτερο από 7 έως 10.

Τώρα θα εξετάσουμε το πρόβλημα Π2 (3 κρυφά επίπεδα).

(με  $(\_,\_,\_)$  θα συμβολίζονται ο αριθμός νευρώνων στο κρυφό επίπεδο 1, στο κρυφό επίπεδο 2 και στο κρυφό επίπεδο 3. Π.χ.  $(1,3,2) = 1$  νευρώνας στο πρώτο κρυφό επίπεδο, 3 νευρώνες στο κρυφό επίπεδο 2 και 2 νευρώνες στο κρυφό επίπεδο 3)

**\*\* Για συνάρτηση ενεργοποίησης TANH \*\***

Π.Ε. = Ποσοστό επιτυχίας

Για τις ακραίες τιμές  $\rightarrow (1,1,1)$  Π.Ε. = 0%

Για τις τιμές  $\rightarrow (2,2,2)$  = 0%

Για τις τιμές  $\rightarrow (4,4,4)$  = 0%

Για τις τιμές  $\rightarrow (6,6,6)$  = 22%

Για τις τιμές  $\rightarrow (8,8,8)$  = 21%

Για τις τιμές  $\rightarrow (3,5,7)$  = 0%

Για τις τιμές  $\rightarrow (7,5,3)$  = 40%

Για τις τιμές  $\rightarrow (8,6,4)$  = 33%

Για τις τιμές  $\rightarrow (7,9,5)$  = 43%

Για τις τιμές  $\rightarrow (8,4,8)$  = 48%

Για τις τιμές  $\rightarrow (13,13,13)$  = 86%

Για τις τιμές  $\rightarrow (15,15,15)$  = 88%

Για τις τιμές  $\rightarrow (20, 20, 20)$  = 90%

Για τις τιμές  $\rightarrow (12, 6, 12)$  = 50%

Για τις τιμές  $\rightarrow (15, 9, 15)$  = 94%

Για τις τιμές  $\rightarrow (13, 9, 13)$  = 93%

**\*\* Για συνάρτηση ενεργοποίησης SIGMOID \*\***

Π.Ε. = Ποσοστό επιτυχίας

Για τις ακραίες τιμές  $\rightarrow (1,1,1)$  Π.Ε. = 0%

Για τις τιμές  $\rightarrow (2,2,2)$  = 0%

Για τις τιμές  $\rightarrow (4,4,4)$  = 70%

Για τις τιμές  $\rightarrow (6,6,6)$  = 76%

Για τις τιμές  $\rightarrow (8,8,8) = 88\%$

Για τις τιμές  $\rightarrow (3,5,7) = 5\%$

Για τις τιμές  $\rightarrow (7,5,3) = 76\%$

Για τις τιμές  $\rightarrow (8,6,4) = 87\%$

Για τις τιμές  $\rightarrow (7,9,5) = 82\%$

Για τις τιμές  $\rightarrow (8,4,8) = 84\%$

Για τις τιμές  $\rightarrow (13,13,13) = 92\%$

Για τις τιμές  $\rightarrow (15,15,15) = 95\%$

Για τις τιμές  $\rightarrow (20, 20, 20) = 92\%$

Για τις τιμές  $\rightarrow (12, 6, 12) = 89\%$

Για τις τιμές  $\rightarrow (15, 9, 15) = 92\%$

Για τις τιμές  $\rightarrow (13, 9, 13) = 93\%$

Εφόσον βρήκαμε τις πειραματικές τιμές, θα σχολιάσουμε τα αποτελέσματα.

**Παρατήρηση 1)** Φαίνεται το σύστημα να έχει μία μικρή προτίμηση στην σιγμοειδή συνάρτηση ενεργοποίησης.

**Παρατήρηση 2)** Ανάμεσα στο Perceptron με 2 κρυφά επίπεδα και στο Perceptron με 3 κρυφά επίπεδα, η διαφορά είναι μικρή για τιμές άνω του 85%. Στα 3 κρυφά επίπεδα ο αριθμός των νευρώνων αυξάνεται πάρα πολύ έως ότου το σύστημα ανταποκρίνεται με επιτυχία άνω του 90%, που σημαίνει πως το υπολογιστικό κόστος ανεβαίνει. Με βάση αυτό, για το συγκεκριμένο πρόβλημα, προτιμούμε το Perceptron με 2 κρυφά επίπεδα.

Εντολή μεταγλώττισης: `javac ./Networkneurons.java`

Εκτέλεσης: `java Networkneurons`