**1Η ΥΠΟΧΡΕΩΤΙΚΗ ΕΡΓΑΣΙΑ**

**ΣΤΟ ΜΑΘΗΜΑ «ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ – ΒΑΘΙΑ ΜΑΘΗΣΗ»**

Χρήστος Χριστίδης

ΑΕΜ 3350

[christpc@csd.auth.gr](mailto:christpc@csd.auth.gr)

28/11/2021

Τμήμα Πληροφορικής ΑΠΘ

Εισαγωγή

Για την εκπόνηση της εργασίας χρησιμοποιήθηκε η γλώσσα Python και συγκεκριμένα το framework PyTorch. Το νευρωνικό δίκτυο και οι πράξεις που το αφορούν τρέχουν στη CPU, επομένως οι χρόνοι απόδοσης που αναφέρονται σε επόμενες ενότητες αναφέρονται στην απόδοση του νευρωνικού στη CPU και όχι στη GPU. Ο σκοπός του νευρωνικού δικτύου είναι η αναγνώριση ψηφίων 0-9 σε εικόνες 28x28 px. Για τα training και testing data sets χρησιμοποιήθηκε η βιβλιοθήκη MNIST και σε κάθε run του δικτύου τα set αυτά γίνονται shuffled.

Συμπληρωματικά χρησιμοποιήθηκαν οι βιβλιοθήκες numpy και matplotlib για την εκτέλεση πράξεων και τη σχεδίαση γραφικών παραστάσεων και σχημάτων καθώς και η βιβλιοθήκη time για την καταγραφή χρόνων εκπαίδευσης και testing.

Θα παρατεθούν δεδομένα και στατιστικά σε μορφή πινάκων. Πιο συγκεκριμένα, η χρονική απόδοση του δικτύου, το ποσοστό επιτυχίας κατά το training και testing καθώς και θα δοθούν παραδείγματα ορθής και εσφαλμένης κατηγοριοποίησης. Κάθε παραλλαγή του δικτύου (διαφορετικός αριθμός νευρώνων στο κρυφό επίπεδο, διαφορετικός optimizer, διαφορετικός αριθμός κρυφών layers) θα συνοδεύεται από ξεχωριστό πίνακα και παραδείγματα. Τέλος παρατίθεται σύγκριση του δικτύου και των διαφορετικών εκφάνσεών του με τους αλγορίθμους Nearest Neighbor και Nearest Class Centroid.

Περιγραφή του δικτύου και του αντίστοιχου κώδικα

Αρχικά να σημειωθεί ότι ελέγχθηκαν 6 παραλλαγές του νευρωνικού δικτύου.

**Αρχικός Σχεδισμός ή ΑΣ (OriginalNeural.py):** Δίκτυο με δύο hidden layers. Το αρχικό layer δέχεται ως είσοδο 784 νευρώνες (διάνυσμα 784 στοιχείων, ο αριθμός προκύπτει από τα 28x28 pixel της εικόνας) και έχει ως έξοδο 64 νευρώνες. Τα δύο ενδιάμεσα κρυφά στρώματα έχουν είσοδο και έξοδο 64 νευρώνες. Τέλος το τελικό στρώμα έχει ως είσοδο 64 νευρώνες και έξοδο 10 νευρώνες, όπου κάθε ένας αντιπροσωπεύει ένα από τα ψηφία 0-9. Ο κώδικας υλοποιεί τον παραπάνω σχεδιασμό είναι η κλάση MyNetwork.

Για τα τρία πρώτα στρώματα η συνάρτηση ενεργοποίησης (activation function) είναι η Rectified Linear Unit (ReLU) ενώ για το τελευταίο επίπεδο χρησιμοποιήθηκε η Softmax. Το κομμάτι του κώδικα που υλοποιεί το forward propagation με τις συναρτήσεις που μόλις περιεγράφηκαν είναι η μέθοδος forward της κλάσης MyNetwork.

Χρησιμοποιήθηκε ο optimizer αλγόριθμος Adam της βιβλιοθήκης torch με learning rate lr=0.01 ο οποίος αυτόματα αρχικοποιεί (μέσω της μεθόδου zero\_grad) και διαχειρίζεται την ενημέρωση των βαρών κατά τη διάρκεια της back propagation η οποία επίσης έγινε μέσω της ενσωματωμένης μεθόδου backward του πακέτου. H εκπαίδευση του μοντέλου διαρκεί 3 εποχές.

**1η παραλλαγή:** Ίδιος σχεδιασμός με ΑΣ και διαφορετικός αριθμός νευρώνων ανά κρυφό επίπεδο **A)** Μεγαλύτερος αριθμός (128 αντί για 64) **Β)** Μικρότερος αριθμός (32 αντί για 64).

**2η παραλλαγή:** Ίδιος σχεδιασμός με ΑΣ και διαφορετικός αριθμός κρυφών επιπέδων.. **A)** Μεγαλύτερος αριθμός κρυφών επιπέδων (6 αντί για 2).**Β)** Μικρότερος αριθμός κρυφών επιπέδων (1 αντί για 2).

**3η παραλλαγή:** Ίδιος σχεδιασμός με ΑΣ, διαφορετικές συναρτήσεις ενεργοποίησης (συγκεκριμένα softplus αντί για ReLU).

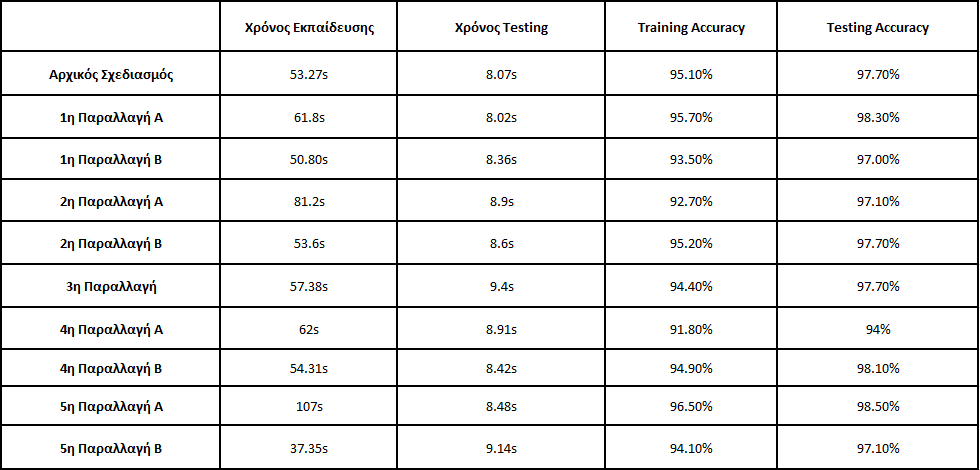
**4η παραλλαγή:** Ίδιος σχεδιασμός με τον ΑΣ με διαφορετικό learning rate **A)** Μεγαλύτερο learning rate (0.01 αντί για 0.001) **Β)** Μικρότερο learning rate (0.00095 αντί για 0.001).

**5η παραλλαγή:** Ίδιος σχεδιασμός με τον ΑΣ με διαφορετικό αριθμό εποχών. **A)** Μεγαλύτερος αριθμός εποχών (6 αντί για 3). **Β)** Μικρότερος αριθμός εποχών (2 αντί για 3).

Για την υλοποίηση κάθε παραλλαγής εντός του αρχείου OriginalNeural.py αλλάχθηκε η τιμή των μεταβλητών EPOCHS, learningRate και neurons που βρίσκονται στην αρχή του εγγράφου ή χρησιμοποιήθηκαν κάποια από τα τμήματα κώδικα που βρίσκονται σε σχόλια.

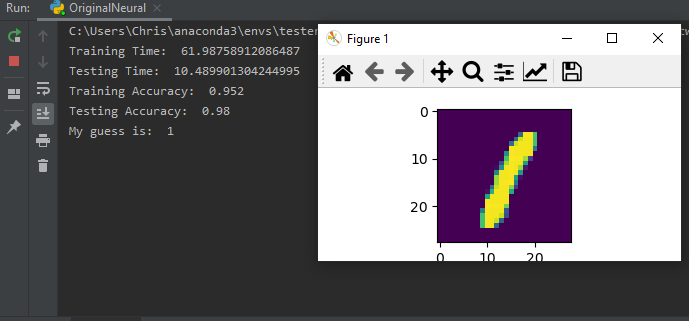
Παρουσίαση και σχολιασμός αποτελεσμάτων

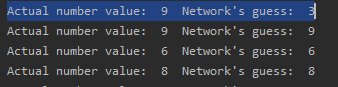
Τα στατιστικά που παρουσιάζονται παρακάτω στους πίνακες αφορούν ΜΟ για 10 manual runs της κάθε παραλλαγής του νευρωνικού δικτύου. Εντός του κώδικα υπάρχουν μετρητές χρόνου και επιτυχίας στα τμήματα που αφορούν την εκπαίδευση και το testing.

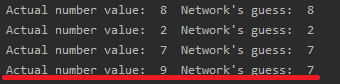


* Ο πιο αποτελεσματικός σχεδιασμός από άποψη **χρόνου εκπαίδευσης** είναι η **παραλλαγή 5Β**
* Ο πιο αποτελεσματικός σχεδιασμός από άποψη **χρόνου testing** είναι η **παραλλαγή 1A.**
* Ο πιο αποτελεσματικός σχεδιασμός από άποψη **Training Accuracy** είναι η **παραλλαγή 5A.**
* Ο πιο αποτελεσματικός σχεδιασμός από άποψη **Testing Accuracy** είναι η **παραλλαγή 5A.**

Ένα χρήσιμο συμπέρασμα που μπορεί να εξαχθεί επομένως είναι ότι ο αριθμός εποχών κατά την εκπαίδευση φαίνεται να επηρεάζει τους χρόνους και τα ποσοστά επιτυχίας εκπαίδευσης και testing περισσότερο από τις άλλες παραμέτρους.

 Στη συνέχεια ακολουθούν μερικά παραδείγματα σωστής αλλά και εσφαλμένης κατηγοριοποίησης.





Σύγκριση ΑΣ με τους Αλγορίθμους Nearest Neighbor και Nearest Class Centroid