1η ΥΠΟΧΡΕΩΤΙΚΗ ΕΡΓΑΣΙΑ
ΣΤΟ ΜΑΘΗΜΑ «ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ – ΒΑΘΙΑ ΜΑΘΗΣΗ»

Χρήστος Χριστίδης
ΑΕΜ 3350
christpc@csd.auth.gr
28/11/2021
Τμήμα Πληροφορικής ΑΠΘ

Εισαγωγή

Για την εκπόνηση της εργασίας χρησιμοποιήθηκε η γλώσσα Python (version 3.0) και συγκεκριμένα το framework PyTorch στο περιβάλλον Anaconda. 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 όταν τρέχει στη CPU και όχι στη GPU. Ο σκοπός του νευρωνικού δικτύου είναι η αναγνώριση hand drawn 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 randomize για να υπάρχει καλύτερη κατανομή των δεδομένων.

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 κατά την εκπαίδευση και τον έλεγχο, το ποσοστό επιτυχίας training και testing καθώς και θα δοθούν παραδείγματα ορθής και εσφαλμένης κατηγοριοποίησης. Κάθε παραλλαγή του δικτύου (διαφορετικός αριθμός νευρώνων στο κρυφό στρώμα, διαφορετικό learning rate, διαφορετικός αριθμός κρυφών layers, διαφορετικός αριθμός εποχών κλπ.) θα συνοδεύεται από τα αντίστοιχα δεδομένα και στατιστικά που την αφορούν. Τέλος, γίνεται σύγκριση του δικτύου που υλοποιήθηκε με τον αλγόριθμο K-Nearest Neighbor.

Σημείωση προς τον καθηγητή: Σε περίπτωση που κάποιο/α από τα αρχεία είναι corrupt ή δεν ανοίγουν, μπορείτε να βρείτε το αρχείο κώδικα και τη γραπτή αναφορά της εργασίας στο GitHub repository https://github.com/FSBgr/neural1

Περιγραφή του δικτύου και του αντίστοιχου κώδικα

Αρχικά να σημειωθεί ότι το δίκτυο που υλοποιήθηκε είναι πλήρως συνδεδεμένο (όχι συνελλικτικό ή μεικτό) και ότι ελέγχθηκαν 6 παραλλαγές του.

Αρχικός Σχεδισμός ή ΑΣ (MyNeural.py): Δίκτυο με συνολικά 4 στρώματα (δύο hidden layers). Το αρχικό layer έχει **είσοδο 784 νευρώνες** (διάνυσμα 784 στοιχείων, ο αριθμός προκύπτει από τα 28x28 pixel της εικόνας) και έχει **έξοδο 64 νευρώνες**. Τα δύο ενδιάμεσα κρυφά στρώματα έχουν **είσοδο και έξοδο 64 νευρώνες**. 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 στρώμα (εξόδου) χρησιμοποιήθηκε η συνάρτηση **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=0.001** ο οποίος αυτόματα αρχικοποιεί τα βάρη (μέσω της μεθόδου zero_grad) και διαχειρίζεται την ενημέρωση των βαρών κατά τη διάρκεια του back propagation, που επίσης έγινε μέσω της ενσωματωμένης μεθόδου backward του πακέτου. Η εκπαίδευση του μοντέλου διαρκεί **3 εποχές.**

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

2η παραλλαγή: Ίδιος σχεδιασμός με ΑΣ και διαφορετικός αριθμός κρυφών επιπέδων.. **Α)** 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 στο αρχικό και στα κρυφά στρώματα (συγκεκριμένα softplus αντί για ReLU).

4η παραλλαγή: Ίδιος σχεδιασμός με τον ΑΣ με διαφορετικό learning rate **Α)** 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. **Α)** 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 τον μέσο όρο 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.

	Χρόνος Εκπαίδευσης	Χρόνος Testing	Training Accuracy	Testing Accuracy
Αρχικός Σχεδιασμός	53.27s	8.07s	95.10%	97.70%
1η Παραλλαγή Α	61.8s	8.02s	95.70%	98.30%
1η Παραλλαγή Β	50.80s	8.36s	93.50%	97.00%
2η Παραλλαγή Α	81.2s	8.9s	92.70%	97.10%
2η Παραλλαγή Β	53.6s	8.6s	95.20%	97.70%
3η Παραλλαγή	57.38s	9.4s	94.40%	97.70%
4η Παραλλαγή Α	62s	8.91s	91.80%	94%
4η Παραλλαγή Β	54.31s	8.42s	94.90%	98.10%
5η Παραλλαγή Α	107s	8.48s	96.50%	98.50%
5η Παραλλαγή Β	37.35s	9.14s	94.10%	97.10%

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

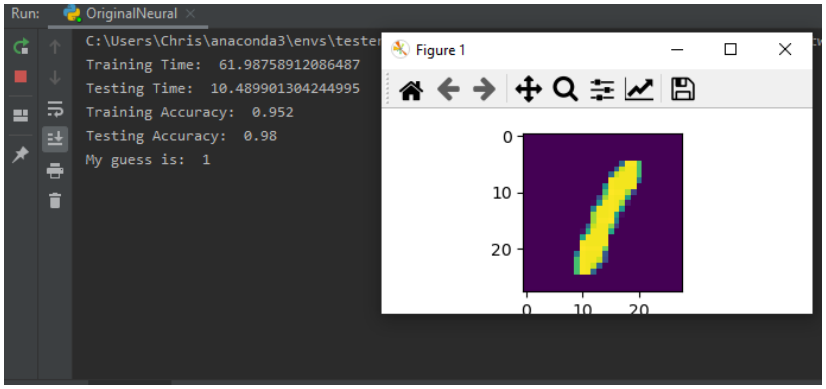
Φαίνεται από τα δεδομένα ότι όσο περισσότερες είναι οι εποχές εκπαίδευσης, τόσο περισσότερο διαρκεί ο χρόνος εκπαίδευσης. Επίσης, όσο μικρότερο είναι το learning rate, τόσο περισσότερο χρόνο χρειάζεται το δίκτυο για να εκπαιδευτεί. Η τυπική απόκλιση των χρόνων εκπαίδευσης είναι **σ=18.313s.** Αντίθετα, ο χρόνος testing δε διαφέρει δραματικά σε καμία παραλλαγή του δικτύου, με τυπική απόκλιση χρόνων **σ=0.427s.**

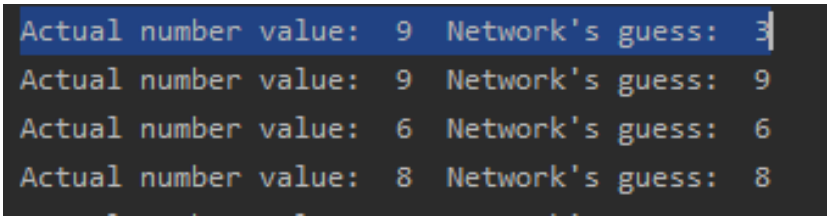
Όσο αφορά τα επίπεδα επιτυχίας, πάλι φαίνεται ότι ο μεγαλύτερος αριθμός εποχών εκπαίδευσης μεγιστοποιεί την επιτυχία εκπαίδευσης αλλά και ελέγχου ενώ ο μικρός αριθμός κρυφών επιπέδων αλλά κυρίως το μεγαλύτερο learning rate τείνουν να μειώνουν την αξιοπιστία του δικτύου κατά την εκπαίδευση και το testing.

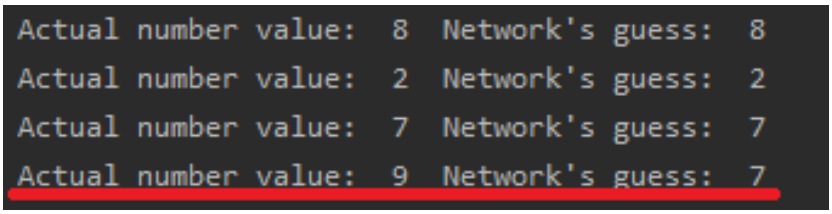
Η τυπική απόκλιση του ποσοστού επιτυχίας εκπαίδευσης είναι **σ= 1.343s**, ενώ του ποσοστού επιτυχίας testing είναι **σ= 1.201s.**

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 του δικτύου με τον αλγόριθμο K-Nearest Neighbors (KNN)

Πριν γίνει η σύγκριση των δύο μοντέλων, πρέπει να γίνει η παραδοχή ότι για την κατηγοριοποίηση του dataset MNIST μέσω KNN χρησιμοποιήθηκαν μέθοδοι από τη βιβλιοθήκη Sklearn. Η σύγκριση θα γίνει μεταξύ του KNN και του ΑΣ του

νευρωνικού δικτύου ως προς την ακρίβεια αναγνώρισης ενός ψηφίου από το testing set.

Επιλέχθηκε να γίνει κατηγοριοποίηση για **k=1,** όπου μετά από δοκιμές φαίνεται να δίνει την μέγιστη ακρίβεια πρόβλεψης ψηφίων από το testing set. Στη συνέχεια μέσω της βιβλιοθήκης Sklearn τρέχει ο αλγόριθμος KNN πάνω στο training set που εξήχθη από το MNIST για την εκπαίδευση του μοντέλου. Στη συνέχεια γίνεται η πρόβλεψη των ψηφίων από το testing set μέσω της μεθόδου exampleModel.predict() και εκτυπώνονται τα αποτελέσματα μέσω της built-in μεθόδου classification_report. Τα στατιστικά πάνω στα οποία θα γίνει η σύγκριση με το νευρωνικό δίκτυο παρατίθενται παρακάτω:

KNN	
	Precision
0	1
1	0.95
2	1
3	0.98
4	0.98
5	0.98
6	1
7	1
8	0.97
9	0.96
avg/total	0.982

Από τον παραπάνω πίνακα προκύπτει ότι ο αλγόριθμος έχει ακρίβεια **98.2%** ως προς την κατηγοριοποίηση των ψηφίων 0-9 ενώ μάλιστα είναι σε θέση να προβλέψει σωστά τα ψηφία 0,2,6,7 στο **100%** των περιπτώσεων. Στις παραλλαγές του νευρωνικού που παρατέθηκαν εντοπίζεται ποσοστό ακρίβειας μεγαλύτερο του 98.2%, συγκεκριμένα στην **1Α με 98.3%** και στην **5Α με 98.5%** κατά τη διάρκεια του testing. Επομένως υπάρχουν περιπτώσεις όπου ο KNN δεν είναι πιο αξιόπιστος από το νευρωνικό δίκτυο που υλοποιήθηκε παραπάνω.

Το πιο σημαντικό στοιχείο ωστόσο είναι ότι ο αλγόριθμος τελειώνει την κατηγοριοποίηση σε περίπου **0.02s** (ο χρόνος αυτός μάλιστα συμπεριλαμβάνει και το training και το testing), κάνοντάς τον έτσι πολλές φορές πιο γρήγορο από το νευρωνικό δίκτυο, το οποίο στην γρηγορότερη παραλλαγή του ολοκληρώνει την εκπαίδευση σε **37.35s** και στη συνέχεια το testing σε **9.14s.**

Ωστόσο είναι σημαντικό να σημειωθεί ότι για διαφορετικές τιμές k, ο αλγόριθμος ΚΝΝ έχει διαφορετικά ποσοστά ακρίβειας εκπαίδευσης. Για παράδειγμα, για **k=23** το ποσοστό επιτυχίας εκπαίδευσης του μοντέλου πέφτει στο **97.04%** που και πάλι είναι μεγαλύτερο από την πιο αποτελεσματική παραλλαγή του νευρωνικού δικτύου.