



## Εισαγωγή

Η ομάδα μας ασχολήθηκε με την προσομοίωση ενός νευρωνικού δικτύου που μαθαίνει να αναγνωρίζει χειρόγραφα αριθμητικά ψηφία. Το πρόγραμμα μας υλοποιήθηκε με γλώσσα προγραμματισμού python (έκδοση 2<sup>η</sup>) και διαθέτει ακρίβεια περίπου 94% χωρίς ανθρώπινη παρέμβαση, παρέχει δηλαδή αρκετά έγκυρα αποτελέσματα. Ο κώδικας εκπαιδεύει ένα νευρωνικό δίκτυο για την αναγνώριση χειρόγραφων αριθμών από το dataset MNIST. Για την εκπαίδευση απαιτήθηκαν 50.000 εικόνες, ενώ τέλος χρησιμοποιήσαμε την tkinter για την δημιουργία του γραφικού περιβάλλοντος χρήστη.

## Οργάνωση ομάδας

Η ομάδα μας αποτελείται από 5 άτομα. Ο Γιώργος Παχατουρίδης με τον Χρήστο Χρυσανθόπουλο ασχολήθηκαν με την εκπαίδευση του νευρωνικού δικτύου. Ο Σπύρος Νάσσου με τον Ανδρέα Ονόπα ανέλαβαν τον διαχωρισμό των εικόνων μέσα από την δημιουργία ενός κώδικα που φορτώνει και προετοιμάζει το σύνολο δεδομένων MNIST για χρήση σε νευρωνικό δίκτυο. Τέλος το γραφικό menu με tkinter έγινε συνεργατικά με κοινή συνεισφορά από όλα τα μέλη όπως αντίστοιχα η παρουσίαση και η αναφορά της εργασίας πραγματοποιήθηκε από κοινού σε συνάντηση της ομάδας. Υπήρχε συνεχής επικοινωνία μεταξύ των μελών της ομάδας κυρίως για την αντιμετώπιση "σκοτεινών σημείων" που μπορεί να εμφανίστηκαν ενώ το κλίμα που επικρατούσε ήταν εύρυθμο.

**Για την οργάνωση του κώδικα χρησιμοποιήθηκε repository στο github όπου τα μέλη ανέβαζαν τον κώδικά τους:**

**<https://github.com/GPChrisdest/Neural-Network>**

## Διαχωρισμός

Για την πλήρη υλοποίηση του προγράμματος χρειάστηκαν 60.000 εικόνες. Οι πρώτες 50.000 χρησιμοποιήθηκαν για την εκπαίδευση του νευρωνικού δικτύου ενώ οι υπόλοιπες 10.000 αξιοποιούνται κατά την εφαρμογή του. Κάθε εικόνα αποτελείται από 784 pixel (28x28). Το πρόγραμμα στην ουσία φορτώνει το σύνολο δεδομένων MNIST από ένα συμπιεσμένο αρχείο, το οποίο περιέχει χειρόγραφους αριθμούς (0-9) σε μορφή εικόνων, και το προετοιμάζει για χρήση σε ένα νευρωνικό δίκτυο. Ανοίγει το αρχείο `mnist.pkl.gz` και διαβάζει τα δεδομένα εκπαίδευσης, επικύρωσης και δοκιμής. Στη συνέχεια, μετατρέπει τις εικόνες από διανύσματα 784 διαστάσεων σε λίστες με πλειάδες (εικόνα, αποτέλεσμα) και αναδιαμορφώνει τις εικόνες σε μορφή 28x28, ενώ τα αποτελέσματα κωδικοποιούνται ως 10-διάστατα διανύσματα. Για τον διαχωρισμό των στοιχείων χρησιμοποιήθηκαν οι βιβλιοθήκες cPickle και gzip που είναι υπεύθυνες για την διαχείριση του .zip αρχείου.

## Εκπαίδευση

Για την εκπαίδευση του νευρωνικού δικτύου απαιτήθηκε τόσο συγγραφή νέου κώδικα όσο και παρέμβαση στον ήδη δοσμένο από τον καθηγητή που υπήρξε σαν βάση. Ως προς τις αλλαγές, κυρίως εστίασαμε στις συναρτήσεις `evaluate` και `feedforward` που αφορούν την έξοδο των νευρώνων του δικτύου όταν σαν είσοδος δίνεται μία από τις εικόνες MNIST, καθώς και επαλήθευση του αποτελέσματος του δικτύου. Μεγαλύτερη βαρύτητα δώσαμε στο κομμάτι της εκπαίδευσης. Πιο συγκεκριμένα θέλαμε το πρόγραμμα μετά την εκπαίδευσή του να μπορεί στο τέλος να αποθηκεύσει τα κύρια δεδομένα του δικτύου (νευρώνες, βάρη, κλίσεις, επίπεδα). Έτσι θα μπορεί μετά σε επόμενη βάση να τα ανακαλέσει/φορτώσει και να είναι έτοιμο να αναγνωρίσει τις εικόνες που θα του εισάγουμε. Τέλος χρησιμοποιήσαμε συναρτήσεις για την κάθε λειτουργία του δικτύου με στόχο μια αρτιότερη και πιο οργανωμένη ροή του προγράμματος.

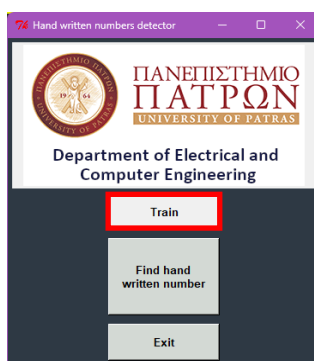
## Γραφική Διεπαφή Χρήστη

Στόχος του τμήματος αυτού ήταν η βελτίωση της εμπειρίας του χρήστη κατά την εκτέλεση του προγράμματος. Χρησιμοποιήθηκαν κυρίως οι βιβλιοθήκες `tkinter` και `matplotlib`. Η πρώτη αφορά το κυρίως `menu` και τις επιλογές του χρήστη ενώ η δεύτερη την εμφάνιση των εικόνων MNIST. Με τον τρόπο αυτό αποφεύγονται τυχόν λάθη (`NameError`, `ValueError`, `TypeError`, κλπ...) διότι ο χρήστης έχει να επιλέξει ανάμεσα σε συγκεκριμένα κουμπιά σε αντίθεση με ένα τυπικό `terminal menu` εισόδου από πληκτρολόγιο. Έτσι διασφαλίζουμε αμυντικό προγραμματισμό αποφεύγοντας τυχόν `bugs` και ο χρήστης έχει μια πιο ομαλή και διαδραστική διεπαφή.

Ως προς το τεχνικό κομμάτι αξιοποιήσαμε απλές εντολές της `tkinter`. Πιο συγκεκριμένα χρησιμοποιήσαμε `Labels` και `Buttons` που καθοδηγούν τον χρήστη σε συνδυασμό με το `terminal` της `Python` που εμφανίζονται τα αποτελέσματα του προγράμματος.

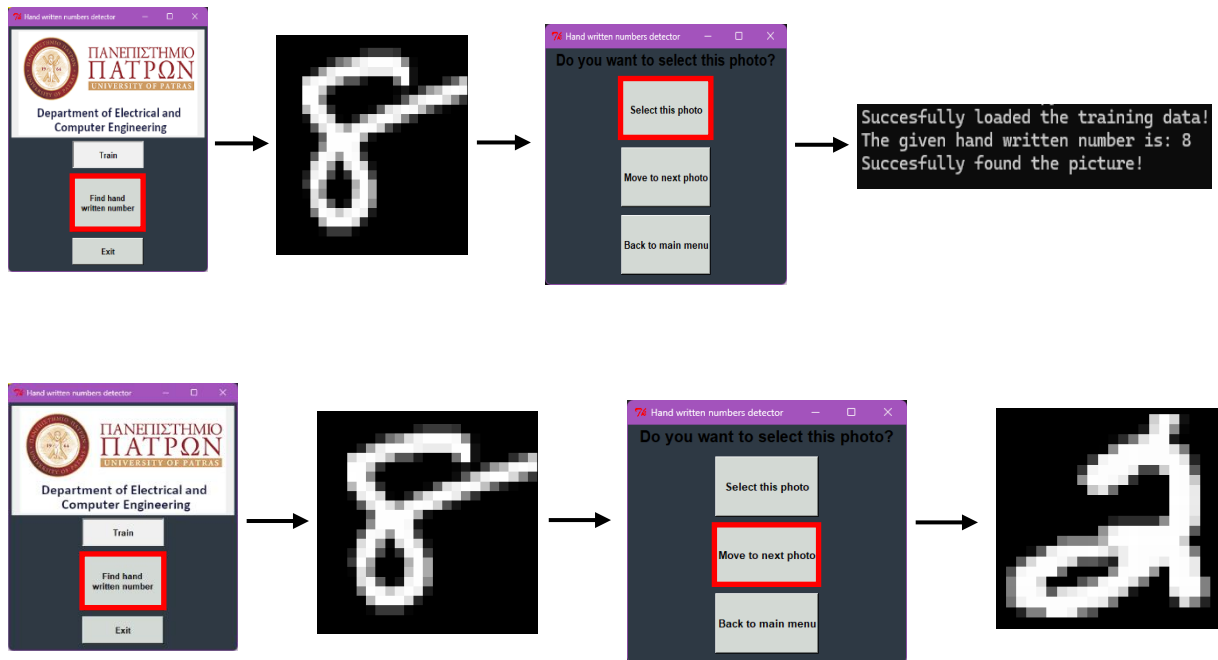
## Παράδειγμα Χρήσης

### Εκπαίδευση:



```
Started training process...
Epoch 0 complete
Epoch 1 complete
Epoch 2 complete
Epoch 3 complete
Epoch 4 complete
Epoch 5 complete
Epoch 6 complete
Epoch 7 complete
Epoch 8 complete
Epoch 9 complete
Epoch 10 complete
Epoch 11 complete
Epoch 12 complete
Epoch 13 complete
Epoch 14 complete
Epoch 15 complete
Epoch 16 complete
Epoch 17 complete
Epoch 18 complete
Epoch 19 complete
Epoch 20 complete
Epoch 21 complete
Epoch 22 complete
Epoch 23 complete
Epoch 24 complete
Epoch 25 complete
Epoch 26 complete
Epoch 27 complete
Epoch 28 complete
Epoch 29 complete
Successfully saved the training data
```

## Αναγνώριση Αριθμού:



## Οδηγίες Εγκατάστασης

- Χρειάζεται να κάνετε add την Python στο PATH του υπολογιστή
- Να υπάρχει εγκατεστημένη Python 2.7.18, pip, numpy και matplotlib  
([https://www.youtube.com/watch?v=0qwyHdA2\\_rw&t=98s](https://www.youtube.com/watch?v=0qwyHdA2_rw&t=98s)  
<https://www.youtube.com/watch?v=K87M0sMVXZE&t=179s>  
<https://www.youtube.com/watch?v=EmBntQAnWyc>)
- Στο φάκελο code πρέπει να έχετε σίγουρα μέσα το Python Terminal
- Ανοίγετε το terminal και γράφετε την εντολή: `execfile("menu.py")`