3η ΥΠΟΧΡΕΩΤΙΚΗ ΕΡΓΑΣΙΑ ΣΤΟ ΜΑΘΗΜΑ «ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ – ΒΑΘΙΑ ΜΑΘΗΣΗ»

Ονοματεπώνυμο : Απόστολος Γιαννουλίδης

ΑΕΜ : 2906

Η υλοποίηση της εργασίας έγινε στην γλώσσα προγραμματισμού Python 3. Η εργασία αφορά την κατηγοροποίηση των δεδομένων από την Mnist βάση δεδομένων σε μονούς και ζυγούς αριθμούς .

Κλάση RBF (αρχείο rbf.py) :

Η κλάση RBF αναπαριστά ένα νευρωνικό δίκτυο με τρία επίπεδα , εισόδου , κρυμμένο επίπεδο με rbf νευρώνες και επίπεδο εξόδου (ένας νευρώνας) . Καλώντας την κλάση μέσω ενός αντικειμένου και προσδιορίζοντας τον τρόπο εκπαίδευσης μέσω των παραμέτρων δημιουργείται ένα νευρωνικό δίκτυο με τρία επίπεδα και γίνονται οι αρχικοποιήσεις όλων των κατάλληλων πεδίων. Έπειτα καλώντας την μέθοδο fit το νευρωνικό δίκτυο αρχικοποιεί τις τιμές των κέντρων και στην συνέχεια προπονεί το δίκτυο σύμφωνα με τις παραμέτρους που δώσαμε .

Η επιλογή των κέντρων γίνεται είτε τυχαία είτε με την εφαρμογή του kmeans αλγορίθμου .

Η ανανέωση των βαρών που συνδέουν το κρυφό επίπεδο με την έξοδο γίνεται με ADALINE.

Δοκιμές υλοποίησης :

Δοκιμές υλοποίησης όσο αναφορά το βήμα εκπαίδευσης και τον καθορισμό κέντρων με σταθερό αριθμό κρυφών νευρώνων (10) :

βήμα=0.1

```
epoch : 0   train acc: 0.7000333333333333 loss: [0.21909762]
epoch : 1   train acc: 0.73425 loss: [0.20282167]
epoch : 2   train acc: 0.7534 loss: [0.1965656]
epoch : 3   train acc: 0.7655166666666666 loss: [0.19291652]
epoch : 4   train acc: 0.7750666666666667 loss: [0.19061612]
epoch : 5   train acc: 0.7812833333333333 loss: [0.18910131]
epoch : 6   train acc: 0.7861 loss: [0.18807014]
epoch : 7   train acc: 0.7894666666666666 loss: [0.18734872]
epoch : 8   train acc: 0.7822333333333333 loss: [0.18683215]
epoch : 9   train acc: 0.7739 loss: [0.18645482]
Time :  185.25
Train results :  0.7739
Testing results :  0.7737
```

βήμα : 0.01

```
epoch : 0   train acc: 0.6285 loss: [0.2325848]
epoch : 1   train acc: 0.6412833333333333 loss: [0.22220314]
epoch : 2   train acc: 0.6499166666666667 loss: [0.21792535]
epoch : 3   train acc: 0.6562833333333333 loss: [0.21464171]
epoch : 4   train acc: 0.66195 loss: [0.21189837]
epoch : 5   train acc: 0.6667333333333333 loss: [0.20954393]
epoch : 6   train acc: 0.6710833333333334 loss: [0.20749632]
epoch : 7   train acc: 0.6747833333333333 loss: [0.2056982]
epoch : 8   train acc: 0.6777 loss: [0.2041055]
epoch : 9   train acc: 0.681 loss: [0.20268335]
Time :  175.109375
Train results :  0.681
Testing results :  0.6812
```

Δοκιμή επιλογής αλγορίθμου για τον καθορισμό των κέντρων , με σταθερό βήμα και αριθμό κρυφό νευρώνων :

Τυχαία

epoch : 0 train acc: 0.7000333333333333 loss: [0.21909762]
epoch : 1 train acc: 0.73425 loss: [0.20282167]
epoch : 2 train acc: 0.7534 loss: [0.1965656]
epoch : 3 train acc: 0.7655166666666666 loss: [0.19291652]
epoch : 4 train acc: 0.7750666666666667 loss: [0.19061612]
epoch : 5 train acc: 0.7812833333333333 loss: [0.18910131]
epoch : 6 train acc: 0.7861 loss: [0.18807014]
epoch : 7 train acc: 0.7894666666666666 loss: [0.18734872]
epoch : 8 train acc: 0.7822333333333333 loss: [0.18683215]
epoch : 9 train acc: 0.7739 loss: [0.18645482]
Time : 185.25
Train results : 0.7739
Testing results : 0.7737

Κ μέσοι

epoch : 0 train acc: 0.6202833333333333 loss: [0.25197674]
epoch : 1 train acc: 0.6426166666666666 loss: [0.2376299]
epoch : 2 train acc: 0.66035 loss: [0.23160168]
epoch : 3 train acc: 0.6735333333333333 loss: [0.22804008]
epoch : 4 train acc: 0.6829333333333333 loss: [0.22559857]
epoch : 5 train acc: 0.6897666666666666 loss: [0.22376715]
epoch : 6 train acc: 0.6956 loss: [0.22231655]
epoch : 7 train acc: 0.7001166666666667 loss: [0.22112897]
epoch : 8 train acc: 0.7045166666666667 loss: [0.22013584]
epoch : 9 train acc: 0.7076333333333333 loss: [0.21929287]
Time : 159.171875
Train results : 0.7076333333333333
Testing results : 0.708

Σύμφωνα με τις παραπάνω δοκιμές επιλέγονται η τιμή 0.1 για βήμα και καθορισμός κέντρων με τον αλγόριθμο kmeans .
Ακολουθούν δοκιμές για τον καθορισμό του αριθμού των κρυφών νευρώνων .

K=2

epoch : 0 train acc: 0.5848833333333333 loss: [0.26119756]
epoch : 1 train acc: 0.5921166666666666 loss: [0.25222856]
epoch : 2 train acc: 0.5955333333333334 loss: [0.25129013]
epoch : 3 train acc: 0.5982 loss: [0.25076594]
epoch : 4 train acc: 0.6007 loss: [0.25029803]
epoch : 5 train acc: 0.6037166666666667 loss: [0.24986763]
epoch : 6 train acc: 0.6055833333333334 loss: [0.24947136]
epoch : 7 train acc: 0.60755 loss: [0.24910653]
epoch : 8 train acc: 0.61025 loss: [0.24877067]
epoch : 9 train acc: 0.6125833333333334 loss: [0.24846146]
Time : 58.140625
Train results : 0.6125833333333334
Testing results : 0.6137

K=5

epoch : 0 train acc: 0.7901166666666667 loss: [0.19678242]
epoch : 1 train acc: 0.7645333333333333 loss: [0.18037671]
epoch : 2 train acc: 0.7570833333333333 loss: [0.1793049]
epoch : 3 train acc: 0.7549333333333333 loss: [0.17919248]
epoch : 4 train acc: 0.7543666666666666 loss: [0.17917759]
epoch : 5 train acc: 0.7540166666666667 loss: [0.17917512]
epoch : 6 train acc: 0.7539166666666667 loss: [0.17917457]
epoch : 7 train acc: 0.7539 loss: [0.17917442]
epoch : 8 train acc: 0.7539 loss: [0.17917437]
epoch : 9 train acc: 0.7539 loss: [0.17917436]
Time : 104.609375
Train results : 0.7539
Testing results : 0.7492

Παρατηρούμε ότι για K=5 στο πρώτο πέρασμα έχουμε ποσοστό υψηλότερο από κάθε άλλο μέχρι στιγμής και στην συνέχει το βλέπουμε να πέφτει . Ως εκ τούτου πραγματοποιούμε την ίδια δοκιμή με μικρότερο βήμα εκπαίδευσης και μεγαλύτερο αριθμό εποχών .
Αποτελέσματα :

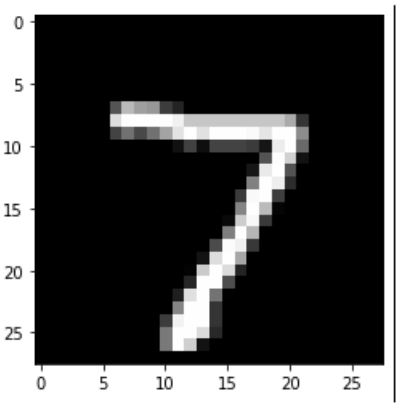
epoch : 0 train acc: 0.6694833333333333 loss: [0.23652673]
epoch : 1 train acc: 0.70025 loss: [0.20540366]
epoch : 2 train acc: 0.7211666666666666 loss: [0.19556875]
epoch : 3 train acc: 0.7378666666666667 loss: [0.18879524]

epoch : 4 train acc: 0.7503666666666666 loss: [0.18395148]
epoch : 5 train acc: 0.7603166666666666 loss: [0.18043694]
epoch : 6 train acc: 0.7682666666666667 loss: [0.17786665]
epoch : 7 train acc: 0.7747666666666667 loss: [0.17597678]
epoch : 8 train acc: 0.7797666666666667 loss: [0.17458101]
epoch : 9 train acc: 0.7838333333333334 loss: [0.17354587]
epoch : 10 train acc: 0.787 loss: [0.17277505]
epoch : 11 train acc: 0.7893833333333333 loss: [0.17219871]
epoch : 12 train acc: 0.7908166666666666 loss: [0.171766]
epoch : 13 train acc: 0.7917666666666666 loss: [0.17143978]
epoch : 14 train acc: 0.79165 loss: [0.17119282]
epoch : 15 train acc: 0.7913666666666667 loss: [0.17100508]
epoch : 16 train acc: 0.7907333333333333 loss: [0.17086177]
epoch : 17 train acc: 0.7895166666666666 loss: [0.17075192]
epoch : 18 train acc: 0.7884833333333333 loss: [0.17066738]
epoch : 19 train acc: 0.78785 loss: [0.17060206]
Time : 196.296875
Train results : 0.78785
Testing results : 0.7807

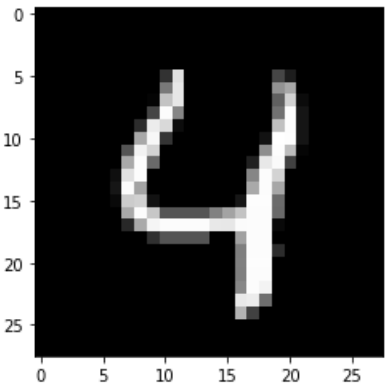
Συμπεραίνουμε πως έχουμε βρει κάποιο τοπικό ελάχιστο (εποχή 13) από το οποίο δεν μπορεί να ξεφύγει η συγκεκριμένη εκπαίδευση .

Λάθος και ορθές προβλέψεις(αρχείο predictions.py) **:**

```
Prediction :  1
Real value :  1
```



```
Prediction :  1
Real value :  0
```



Το μοντέλο πρόβλεψε σωστά πως το 7 είναι μονός αριθμός όμως δεν μπόρεσε να προβλέψει πως το 4 είναι άρτιος αριθμός.

Σύγκριση με τους κατηγοριοποιητής πλησιέστερων γειτώνων και πλησιέστερου κέντρου κλάσης:

Αποτελέσματα από την εκτέλεση του αρχείου try.py :

(Σημείωση ο αλγόριθμος KNeighborsClassifier όπως και ο NearestCentroid , ουσιαστικά δεν εκτελεί κάποιο training το οποίο σημαίνει πως ο χρόνος εκπαίδευσης του μοντέλου είναι μηδενικός , όμως αντίθετα ο χρόνος για να μετρήσουμε την απόδοση του και γενικότερα να κάνουμε πρόβλεψη σε κάποιο μεγάλο σύνολο δεδομένων είναι αρκετά μεγάλος . Για την σύγκριση θα χρησιμοποιηθεί ο χρόνος περιλαμβανομένου και τον χρόνο που χρειαζόμαστε για την αξιολόγησή του .)

RBF :
Time : 128.828125
Train results : 0.7908833333333334
Testing results : 0.7866

KNN3:
Train acc k=3 = 0.9939333333333333
Test acc k=3 = 0.9865
Time : 211.21875

KNN1:
Train acc k=1 = 1.0
Test acc k=1 = 0.9857
Time : 27.671875

NearestCentroid
Train acc = 0.8074666666666667
Test acc = 0.8018
Time : 0.09375

Συμπεραίνουμε πώς για τον καθορισμό των ψηφίων ως μονά και ζυγά είναι προτιμητέο να χρησιμοποιήσουμε τον αλγόριθμο κοντινότερων γειτόνων με 1 γείτονα από ότι το συγκεκριμένο νευρωνικό .