**ΤΡΙΤΗ ΕΡΓΑΣΙΑ: Radial Basis Function Network on MNIST**

**ΜΑΘΗΜΑ: ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ - ΒΑΘΙΑ ΜΑΘΗΣΗ**

**ΓΙΑΝΝΑΚΟΠΟΥΛΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ , ΑΕΜ: 16051** (τμήμα Μαθηματικών)

**ΕΙΣΑΓΩΓΗ**

**Βάση Δεδομένων:**

Για την παρούσα εργασία, καθώς και για την επικείμενη 1η Εργασία, χρησιμοποιήθηκε η βάση δεδομένων MNIST, μια πολύ γνώστη βάση δεδομένων, υποσύνολο της μεγαλύτερης NIST. Περιέχει χειρόγραφα ψηφία αριθμών 0,1,…,9. Στην ιστοσελίδα http://yann.lecun.com/exdb/mnist/ υπάρχουν τέσσερα αρχεία διαθέσιμα: ένα training set 60,000 παρατηρήσεων και τα αντίστοιχα labels, καθώς και ένα test set 10,000 παρατηρήσεων και τα αντίστοιχα labels. Οι εικόνες που περιέχονται σε αυτά είναι διαστάσεων 28x28 σε επίπεδα του γκρι και έχει κανονικοποιηθεί το μέγεθος τους αλλά και έχουν κεντροποιηθεί.

**Γλώσσα Προγραμματισμού:**

Χρησιμοποιήθηκε η -ανοικτού κώδικα- γλώσσα προγραμματισμού R, που χρησιμοποιείται ευρέως στον τομέα της υπολογιστικής στατιστικής. Με τις κατάλληλες βιβλιοθήκες και με τη νεοσύστατη βιβλιοθήκη Keras, δίνεται η δυνατότητα υλοποίησης πολλών αλγορίθμων, όπως o kNN και ο nearest Centroid που χρησιμοποιήθηκαν εδώ, αλλά και άλλων αλγορίθμων στατιστικής μάθησης αλλά και αρχιτεκτονικών βαθιάς μάθησης.

**Preprocessing**

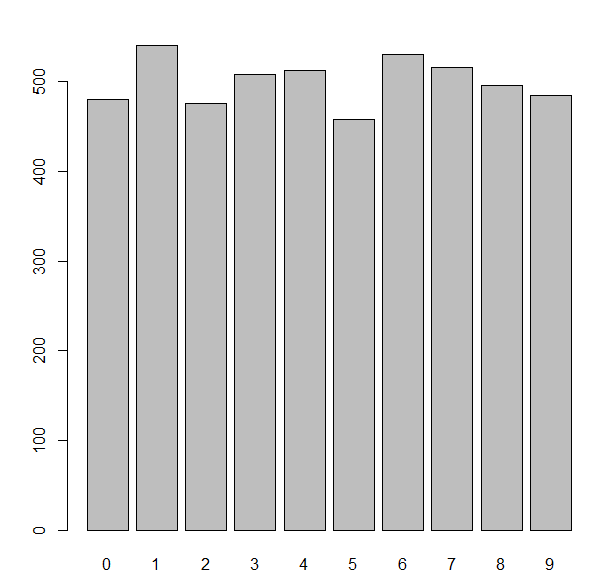
Στη βιβλιοθήκη Keras της R, περιέχονται μερικά από τα πιο γνωστά dataset, όπως η MNIST, Cifar100, Cifar10 κλπ. Αφού, λοιπόν, φορτώθηκαν τα δεδομένα και ονομάστηκαν ως:

train: 60,000 x (784+1)

test: 10,000 x (784+1)

Στη συνέχεια, έγινε scaling στο [0,1] και για να μειωθεί το υπολογιστικό κόστος επιλέχθηκε τυχαίο subset 5,000 παρατηρήσεων από το train για την εκπαίδευση του δικτύου.

*(Σημείωση: Λόγω της μη εξοικείωσης με την PCA, προτίμησα να μην συμπεριλάβω κάτι που δεν κατέχω στο 100%)*



*Τα labels του τυχαίου subset*

**RBF:**

Το πρώτο επίπεδο του δικτύου θα αποτελείται από συναρτήσεις ακτινικής βάσης, συγκεκριμένα Γκαουσιανές. Τα κέντρα τους, επιλέγονται με χρήση του αλγορίθμου k-means. O αριθμός των κέντρων, δηλαδή των νευρώνων του πρώτου επιπέδου επιλέγεται να είναι 30, 70 αρχικά και στη συνέχεια 100. Ο k-means εφαρμόζεται στο σύνολο του dataset και όχι σε κάθε κλάση. Τέλος, το gamma που επιλέγεται, δηλαδή το "πλάτος" της κάθε ακτινικής συνάρτησης είναι σταθερό σε όλους τους νευρώνες 0.24, δηλαδή σχεδόν ίσο με τη μέγιστη απόσταση δυο κέντρων προς την τετραγωνική ρίζα του διπλάσιου αριθμού τους (για καλύτερη σύγκριση το gamma παρέμεινε σταθερό στο 0.24, ανεξάρτητα του αριθμού των κέντρων).

Στη συνέχεια, υπολογίζεται ο πίνακας Φ, με τις ομοιότητες ως εξής:

Δηλαδή, η εκθετική συνάρτηση εις την -0.24 επί την ευκλείδεια νόρμα της διαφοράς κάθε δείγματος από κάθε κέντρο που έχει επιλεχθεί, εις το τετράγωνο.

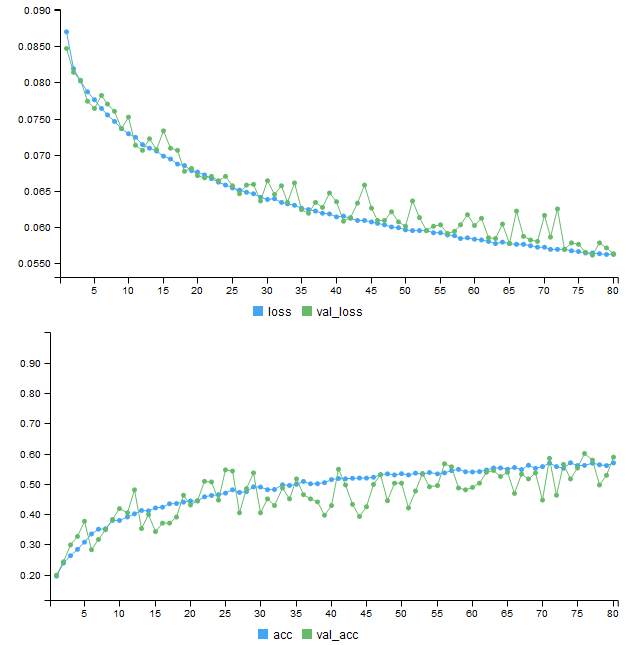
**K- means:** Τυχαία επιλογή k κέντρων από το dataset. Στη συνέχεια υπολογίζεται η ευκλείδεια απόσταση από κάθε κέντρο για κάθε παρατήρηση και τοποθετείτε στο cluster του κέντρου με την ελάχιστη απόσταση. Έπειτα γίνεται εκ νέου υπολογισμός των κέντρων παίρνοντας τη μέση τιμή των στοιχείων του cluster. Η διαδικασία συνεχίζεται για πεπερασμένο αριθμό βημάτων, ώστε να ελαχιστοποιηθεί η διακύμανση από το κέντρο, σε κάθε cluster.

(Σημείωση: Μπορεί να γίνει k- means σε κάθε κλάση των labels ξεχωριστά και επίσης να δοθούν πολλές αρχικοποιήσεις για τα κέντρα και να βρεθεί η καλύτερη)

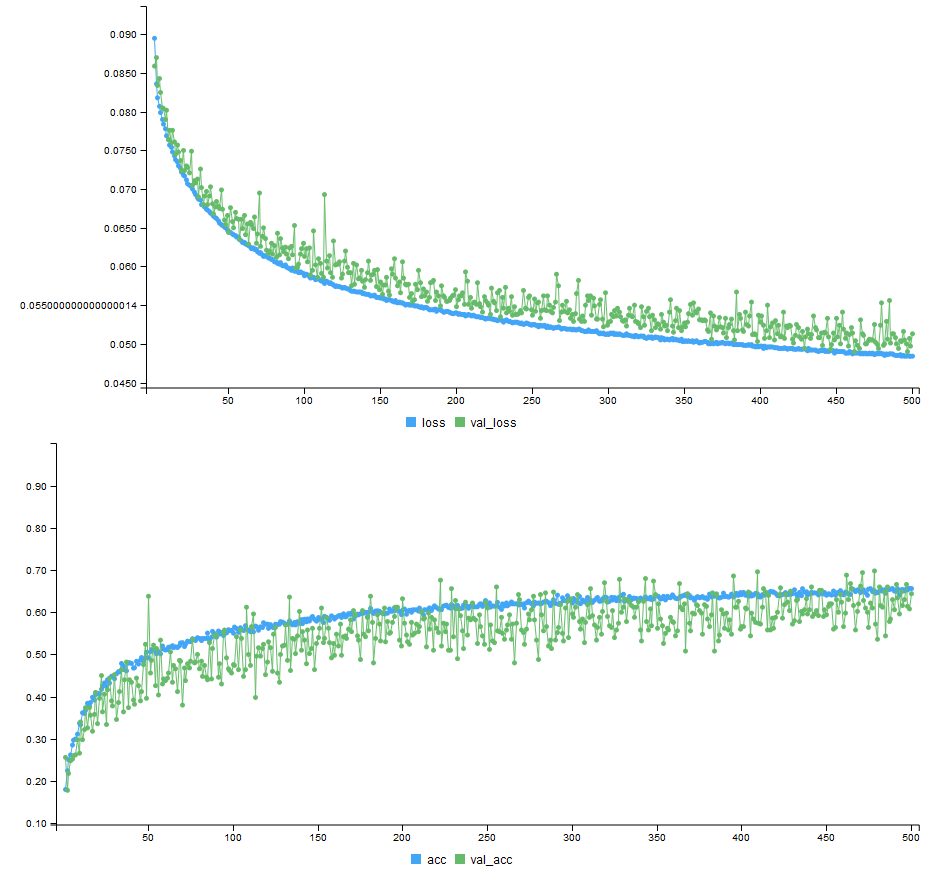
**Perceptron:**

Εν συνεχεία, οι ομοιότητες του πίνακα Φ, περνάνε στο δεύτερο κρυφό επίπεδο, όπου ένας οι perceptron κάνουν classification στις 10 κλάσεις.

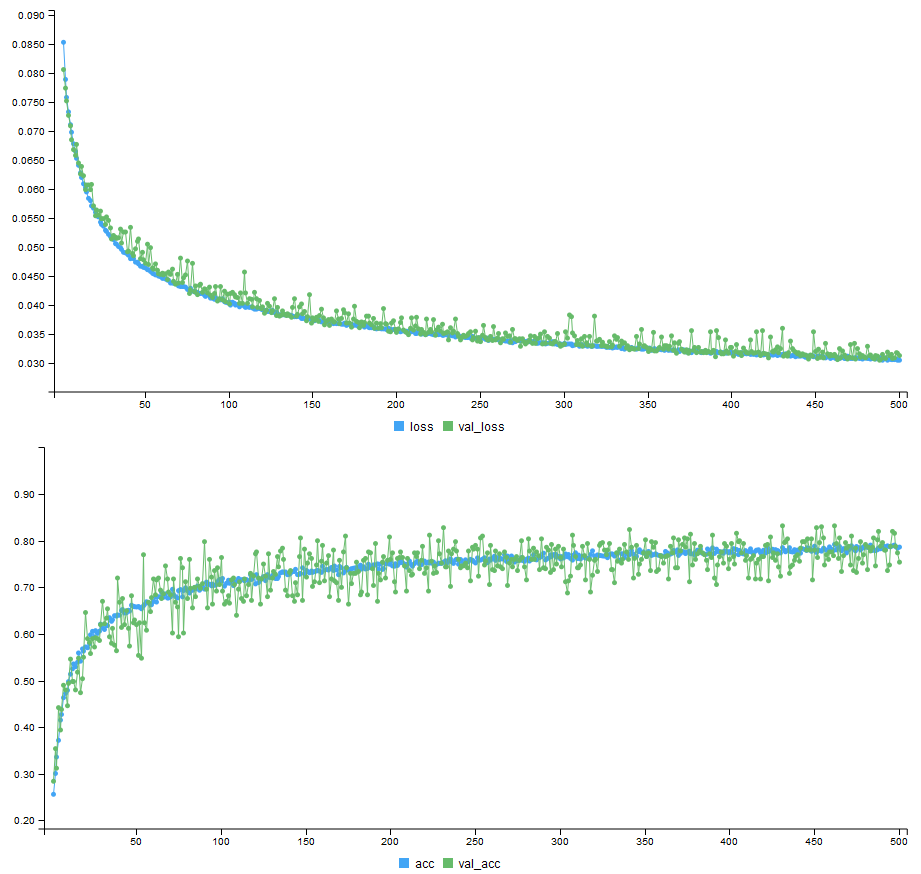
* Για την περίπτωση των 30 κέντρων και τις παραμέτρους του δεύτερου κρυφού επιπέδου δοκιμάστηκαν πολλές τιμές:

Ως συνάρτηση ενεργοποίησης τη λογιστική, συνάρτηση απώλειας το μέσο τετραγωνικό σφάλμα και μετρική το accuracy. Ως optimizer το rmsprop, με learning rate από 0.001 έως 1, με πολύ καλά αποτελέσματα στο 0.4, batch size από 32 έως 128 με καλύτερα αποτελέσματα στο 64, αριθμό εποχών από 30 έως 500 και 10-fold Cross Validation. Ο χρόνος που χρειάστηκε κάθε εποχή ήταν λιγότερο από 1s.

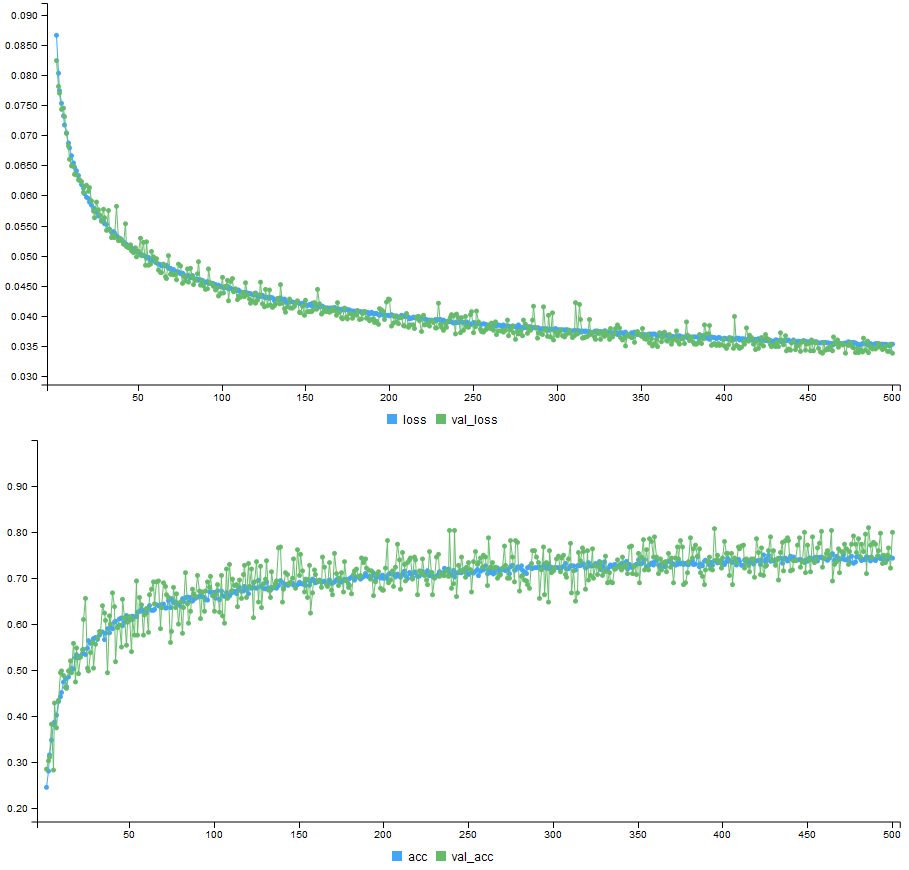
*Ένα παράδειγμα με 80 εποχές learning rate 0.5, batch size 64.  
Το validation accuracy είναι μεταξύ 0.5 και 0.6*



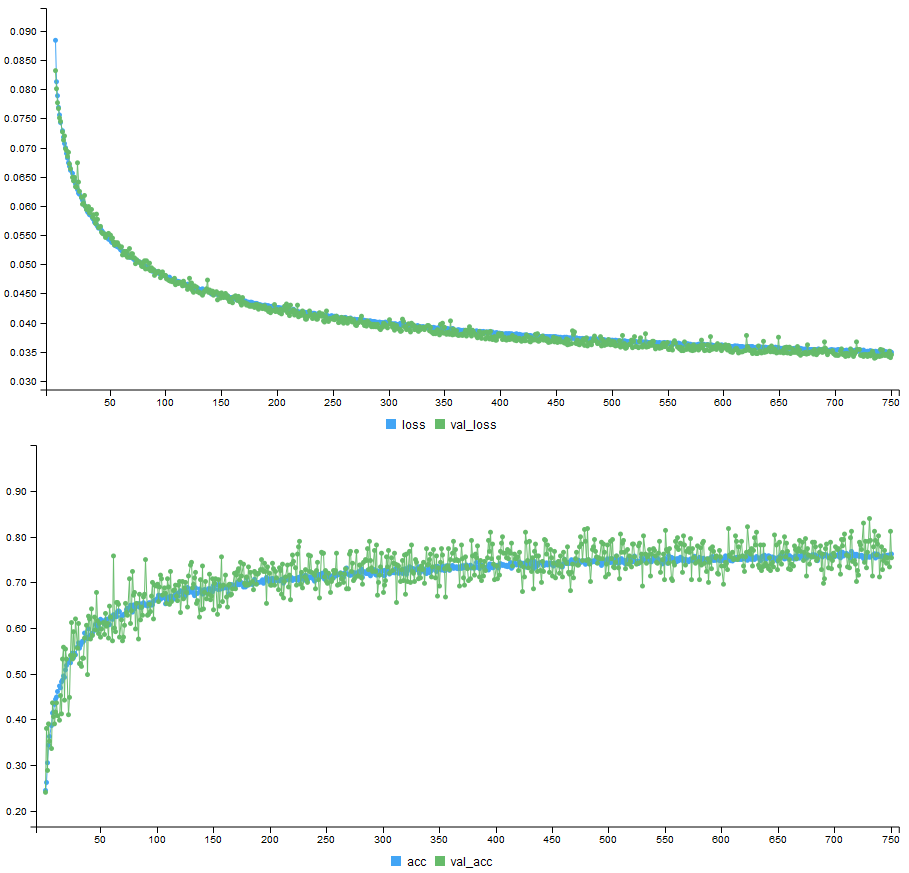
*Από τις καλύτερες επιδόσεις στο validation, σταθερά μεταξύ 0.65 και 0.7.  
Στις 500 εποχές, batch size 64, learning rate 0.4*

* Mια δοκιμή έγινε για 100 κέντρα στο πρώτο κρυφό επίπεδο. Στο δεύτερο κρυφό επίπεδο έχουμε λογιστική συνάρτηση ενεργοποίησης, learning rate 0.75, 500 εποχές και batch size 90, τα οποία φαίνεται να έδωσαν καλύτερα αποτελέσματα.
* Για την περίπτωση των 70 κέντρων και τις παραμέτρους του δεύτερου κρυφού επιπέδου δοκιμάστηκαν πολλές τιμές:

Ως συνάρτηση ενεργοποίησης τη λογιστική, συνάρτηση απώλειας το μέσο τετραγωνικό σφάλμα και μετρική το accuracy. Ως optimizer το rmsprop, με learning rate από 0.001 έως 1, με πολύ καλά αποτελέσματα στο 0.8, batch size από 32 έως 128 με καλύτερα αποτελέσματα στο 90, αριθμό εποχών από 30 έως 750 και 10-fold Cross Validation. Ο χρόνος που χρειάστηκε κάθε εποχή ήταν λιγότερο από 1s.



*Σε αυτό το παράδειγμα έχουμε 500 εποχές, learning rate 0.8, batch size 90. To Validation accuracy είναι με συνέπεια μεταξύ 0.75 και 0.81.*

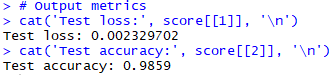


*Εδώ έχουμε 750 εποχές, learning rate 0.5 και batch size 90. Το validation accuracy είναι μεταξύ 0.71 και 0.82.*

Τέλος, με 70 κέντρα στο πρώτο κρυφό επίπεδο και στη συνέχεια στο δεύτερο κρυφό επίπεδο Perceptrons με παραμέτρους:

* Συνάρτηση απώλειας: Μέσο τετραγωνικό σφάλμα
* Μετρική: Accuracy
* Optimizer: rmsprop με learning rate 0.5
* batch size: 90
* epochs: 500

και παίρνοντας 1,000 παρατηρήσεις από το test set, έχουμε accuracy 98.59%.



**Σύγκριση**

Σε αυτό το σημείο μπορούμε να συγκρίνουμε με τα ανάλογα αποτελέσματα των αλγορίθμων k nearest neighbours και nearest centroid που έδωσαν αποτελέσματα 96.91% (k=1), 97.08% (k=3) και 82.03% στα ίδια δεδομένα.

(Εδώ να σημειωθεί ότι είχε χρησιμοποιηθεί ολόκληρο το dataset σε αυτούς τους αλγορίθμους)