

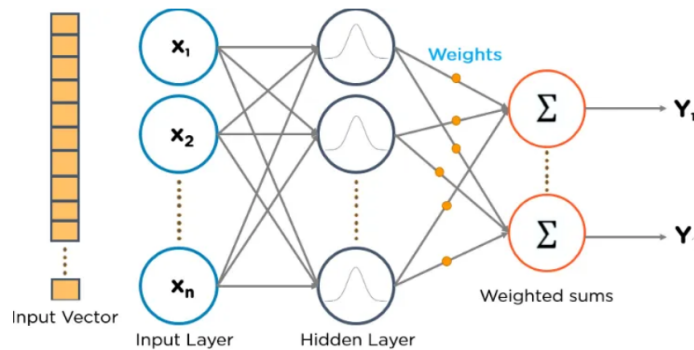
Νευρωνικά Δίκτυα  
Εργασία 3  
RBF neural network

Ηλιάνα Κόγια  
(AEM: 10090)  
ilianakogia@ece.auth.gr

# 1 RBFNN - 10 class classification

## 1.1 Ανάλυση

Δίνεται, αρχικά, το σχήμα του RBF νευρωνικού δικτύου



Η συνάρτηση ενεργοποίησης του κρυφού στρώματος είναι η RBF function:

$$R_i = e^{-\gamma \|x - c_i\|^2}$$

Δύο στάδια εκπαίδευσης:

i) Unsupervised Learning (input layer  $\rightarrow$  hidden layer):

για τον καθορισμό των κέντρων με τη χρήση του αλγορίθμου k-Means. Οι νευρώνες του κρυφού στρώματος είναι ίσοι με τον αριθμό των clusters (= πλήθος κέντρων) στον αλγόριθμο k-Means (ή με τυχαία επιλογή κέντρων).

παράμετροι:  $\gamma$ , κέντρα  $c$

Ομαδοποίηση (clustering)

Μέθοδος k-μέσων

1. Κάθε πρότυπο ανήκει στην ομάδα (cluster) του κέντρου που βρίσκεται πιο κοντά σ' αυτό

2. Κάθε κέντρο είναι ο μέσος όρος των προτύπων που ανήκουν στην ομάδα του

ii) Supervised learning (hidden layer  $\rightarrow$  output layer):

για τον καθορισμό των βαρών, χρήση του κανόνα δέλτα:

$$w(k+1) = w(k) + \beta(d - y) \text{activation}[k]$$

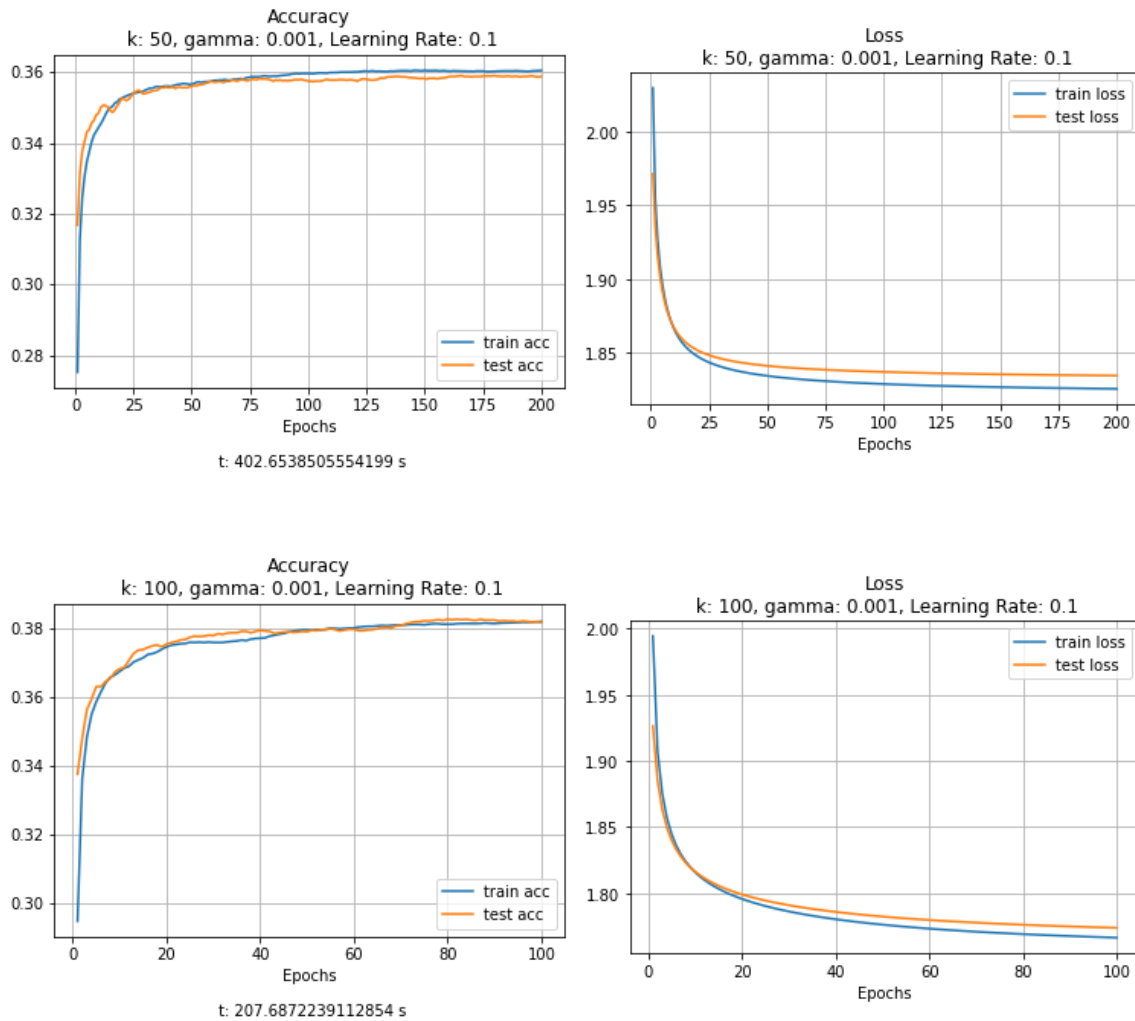
παράμετροι: learning rate, epochs

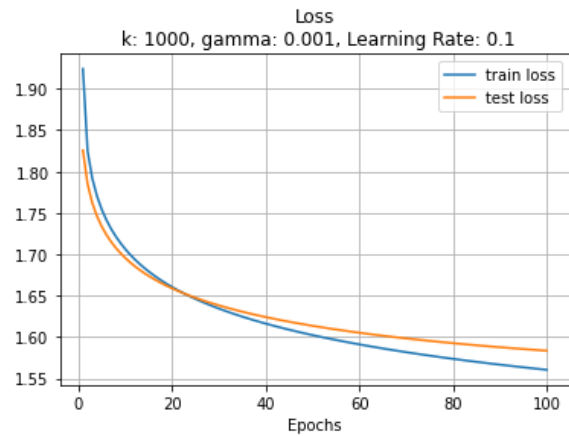
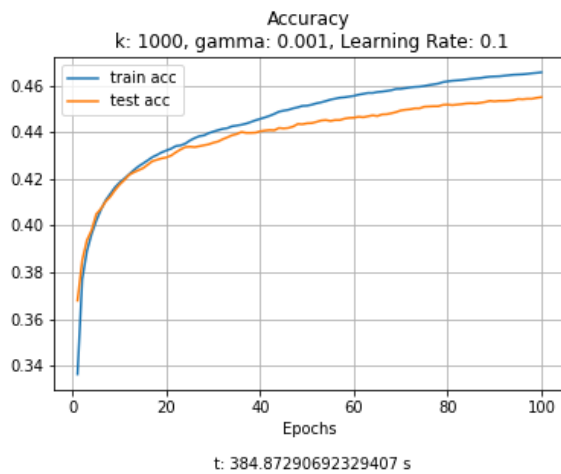
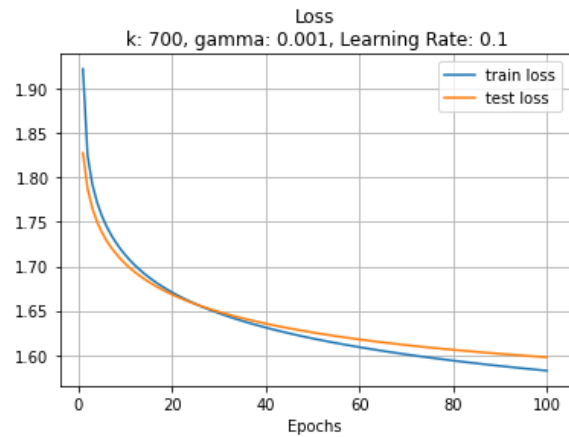
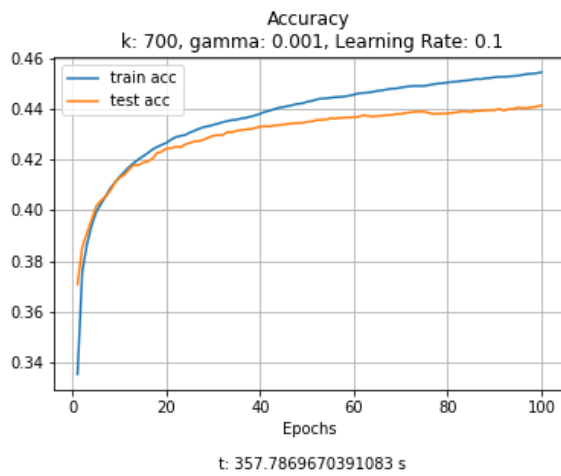
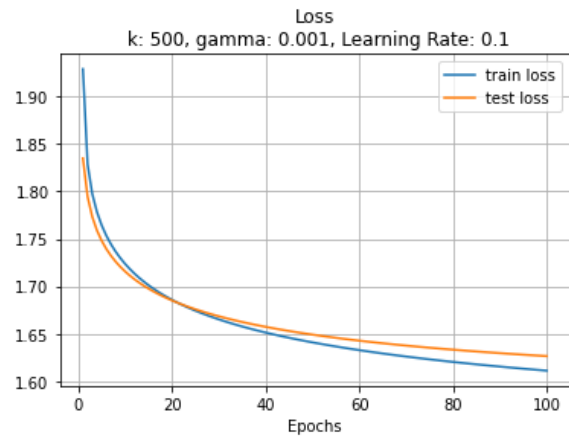
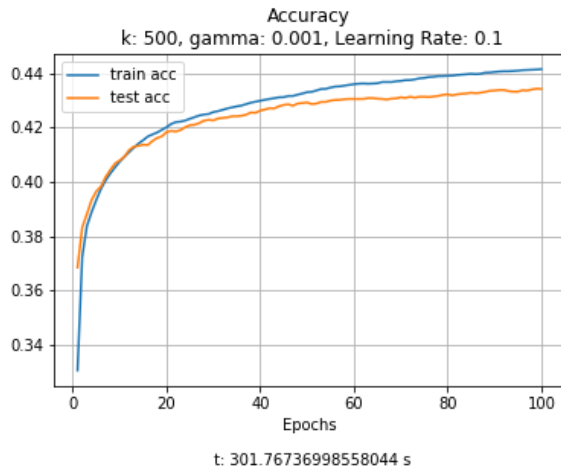
\* Στην υλοποίηση του αλγορίθμου χρησιμοποιήσαμε έναν πίνακα βαρών  $W$  διάστασης: (αριθμός κλάσεων  $\times$  πλήθος hidden neurons), ώστε να εργαστούμε με πράξεις πινάκων. Χρησιμοποιήθηκε η συνάρτηση ενεργοποίησης softmax  $\text{softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$  για το output layer και αντίστοιχα το cross entropy loss.

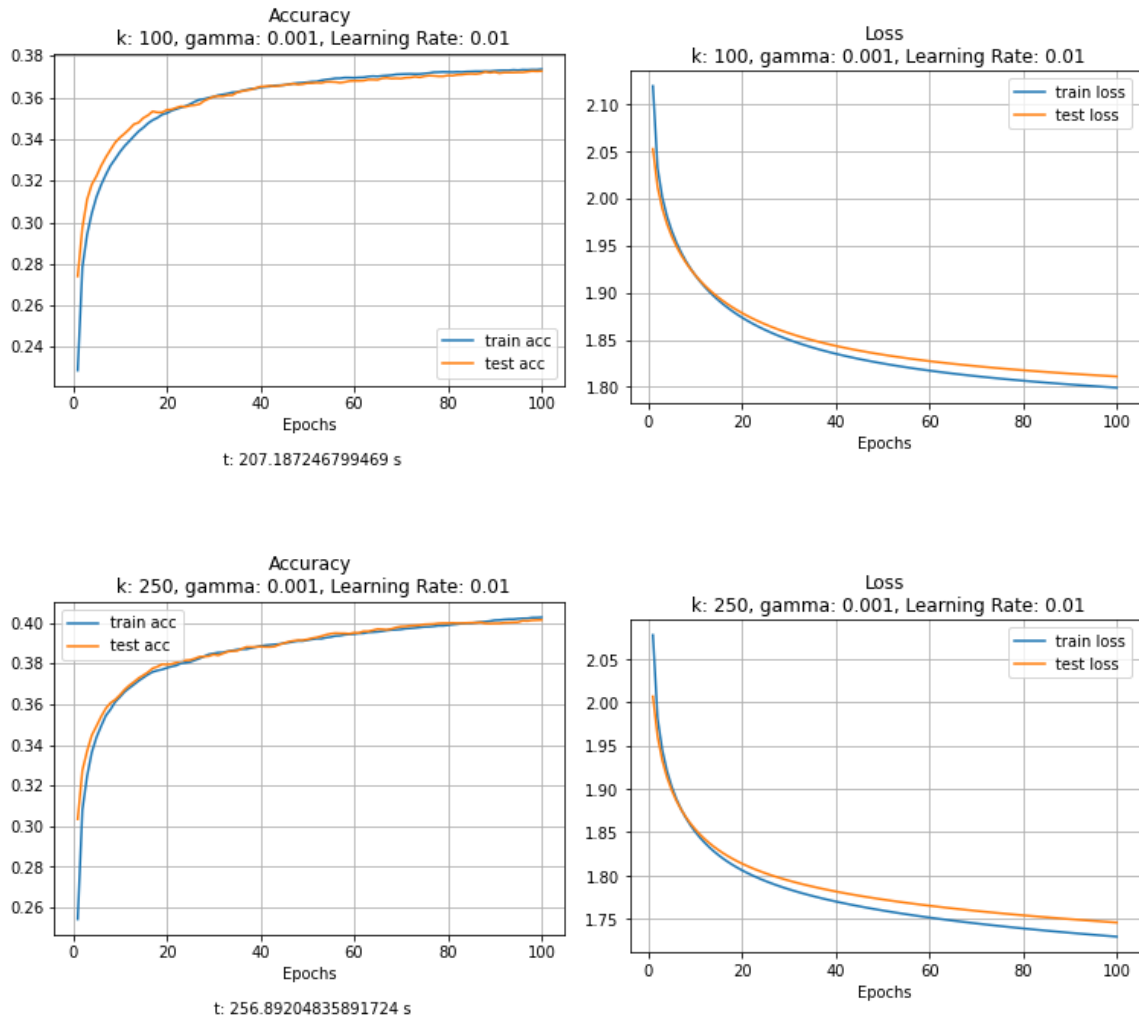
\* Επίσης, χρησιμοποιείται για preprocessing: StandardScaler().

## 1.2 Αποτελέσματα για ταξινόμηση των 10 κλάσεων της **Cifar - 10**

Δοκιμάστηκαν διάφορες τιμές των παραμέτρων  $k$ ,  $\gamma$ , learning rate. Παρατίθενται τα διαγράμματα:



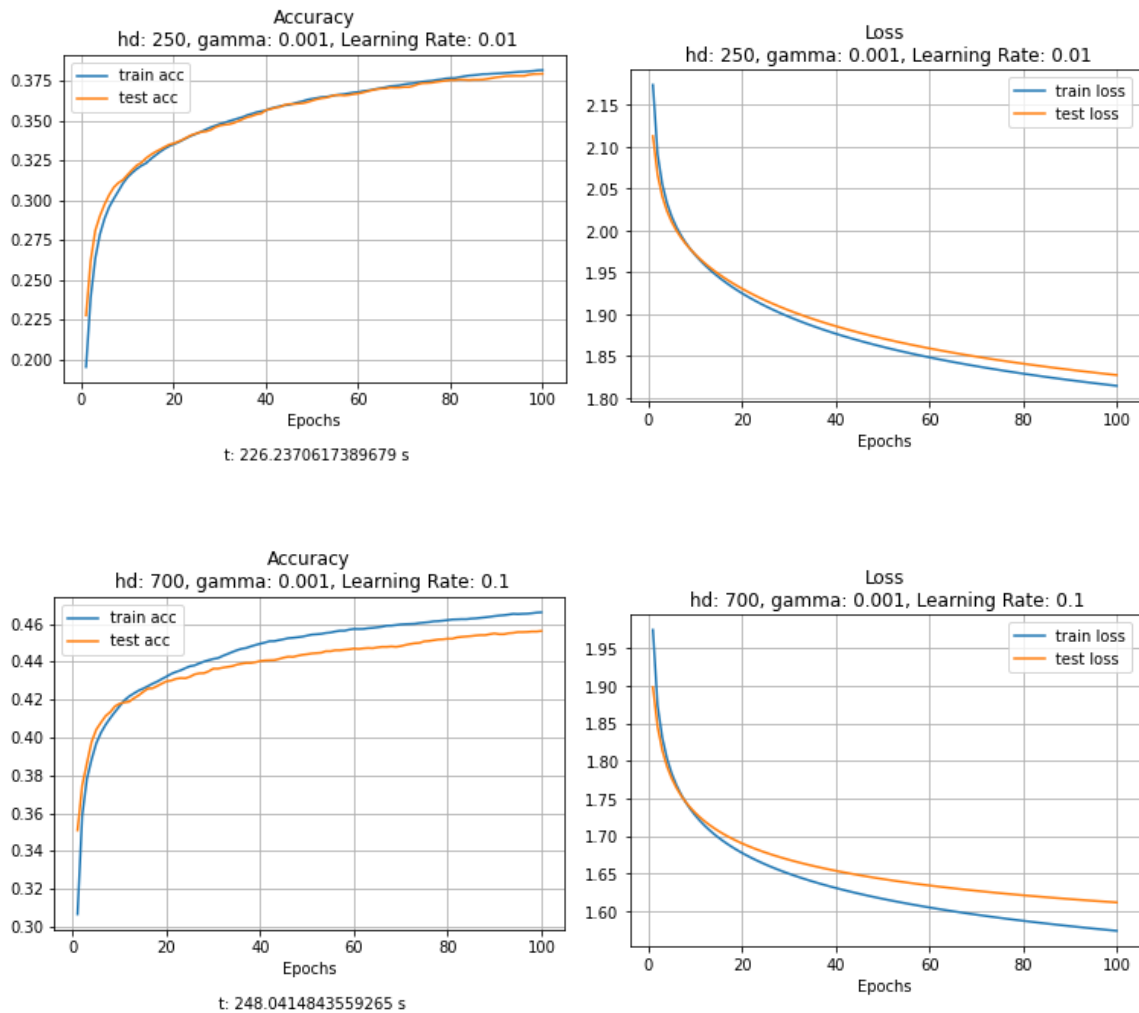




Παρατηρούμε ότι για ίδιες τιμές των  $\gamma$  και learning rate, παρατηρείται καλύτερο accuracy όσο αυξάνεται ο αριθμός των cluster  $k$ , με το καλύτερο να είναι 45% για  $k = 1000$ . Το πιο αργό μέρος είναι το πρώτο στάδιο επιλογής των κέντρων μέσω του  $k$ -Means και γι αυτό διαπιστώνουμε ότι όσο αυξάνεται η τιμή του  $k$ , αυξάνεται αισθητά και ο συνολικός χρόνος (για τον ίδιο αριθμό εποχών στο δεύτερο στάδιο του supervised learning).

Για  $k$  μικρότερο από 50 προέκυπταν χαμηλές τιμές ακρίβειας.

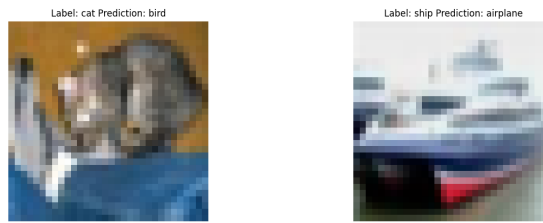
Για τυχαία επιλογή κέντρων μεταξύ των προτύπων:



Συμπεραίνουμε ότι μπορούμε να επιτύχουμε παρόμοια τιμή accuracy σε σύγκριση με τον αλγόριθμο **kMeans**, για την επιλογή των κέντρων, για μεγάλο πλήθος  $k$ , σε λιγότερο χρόνο για ίδιο πλήθος νευρώνων στο κρυφό στρώμα.

Παραδείγματα ορθής και εσφαλμένης ταξινόμησης:



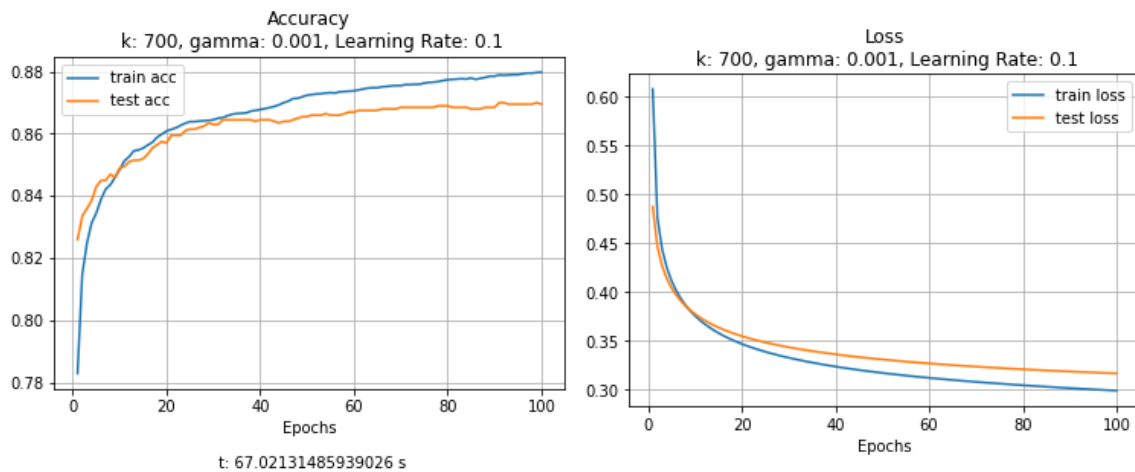


### 1.3 Αποτελέσματα για ταξινόμηση των 2 πρώτων κλάσεων της **Cifar - 10**

Παρατηρούμε ότι για 2 κλάσεις της **Cifar-10** μπορούμε να επιτύχουμε διπλάσιο περίπου accuracy 88%.

(Διαπιστώνουμε ότι με **RBFNN** επιτυγχάνουμε παρόμοια τιμή accuracy με τα νευρωνικά **SVM** για τις δύο αυτές κλάσεις).

Παρατίθενται ενδεικτικά τα παρακάτω:



## 2 Σύγκριση με Nearest Centroid classifier και kNN

Nearest Centroid και k-NN για 10 κλάσεις:

accuracy	
<b>k = 1 nn</b>	<b>35.39%</b>
<b>k = 3 nn</b>	<b>33.03%</b>
<b>nc</b>	<b>27.74%</b>

RBFNN για 10 κλάσεις: **45 %** (hidden neurons = 1000)

Παρατηρούμε ότι επιτυγχάνουμε μεγαλύτερη τιμή accuracy σε σχέση με τους classifiers nearest-centroid και k-NN για τον διαχωρισμό των 10 κλάσεων της Cifar 10.