ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ

Ακαδημαϊκό Έτος 2021-2022

Εργαστηριακή Άσκηση

Μέρος Α’

Γεώργιος Κοντογιάννης

1070908 – Δ’ έτος

Code Repo Link:

https://github.com/gkontogiannhs/Text-Recognition-NLP

Α1. Προεπεξεργασία και Προετοιμασία δεδομένων

α) Η κωδικοποίηση των λέξεων είναι γωνστή και ανήκει στο εύρος [0, 8519]. Για το λόγο αυτό, ο CountVectorizer παραμετροποιήθηκε με αυτό το ήδη υπάρχον λεξικό. Καλώντας την μέθοδο transform, μετατράπηκαν τα δεδομένα εισόδου σε BoW και κατ' επέκταση δημιουργήθηκε το Document Term Matrix.

β) Η κλιμάκωση(scaling) των δεδομένων, είναι ένα από τα πιο σημαντικά βήματα προεπεξεργασίας δεδομένων στη μηχανική εκμάθηση. Οι αλγόριθμοι που υπολογίζουν την απόσταση μεταξύ των χαρακτηριστικών, ωθούνται προς τις αριθμητικά μεγαλύτερες τιμές (outliers) εάν τα δεδομένα δεν είναι κλιμακωμένα.

1. Το **Κεντράρισμα** (centering) αφαιρεί μία σταθερή τιμή απο κάθε μεταβλητή εισόδου. Πρακτικά, επαναπροσδιορίσει το σημείο 0 για τον προγνωστικό παράγοντα, ώστε να είναι οποιαδήποτε τιμή αφαιρέθηκε. Μετατοπίζει την κλίμακα, αλλά διατηρεί τις μονάδες.

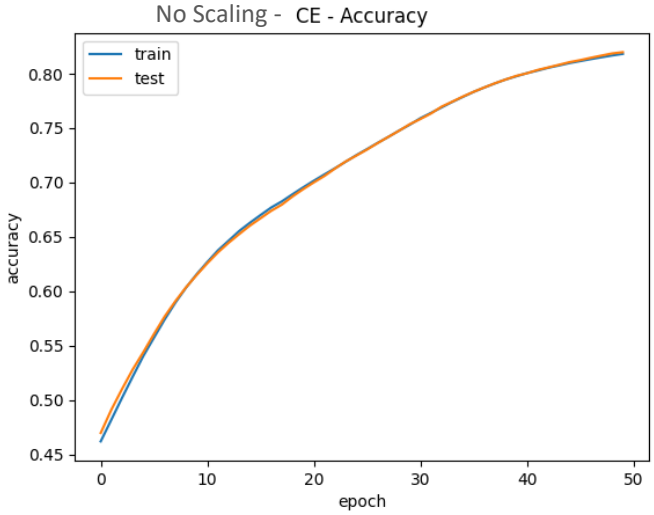
2. Η **κανονικοποίηση** (scaling) χρησιμοποιείται για τη μετατροπή των χαρακτηριστικών σε παρόμοια κλίμακα. Το νέο σημείο υπολογίζεται απο το τύπο: $X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}}$. Αυτό κλιμακώνει το εύρος σε [0, 1] ή μερικές φορές [-1, 1]. *Η κανονικοποίηση είναι χρήσιμη όταν δεν υπάρχουν ακραίες τιμές, καθώς δεν μπορεί να τις αντιμετωπίσει*.

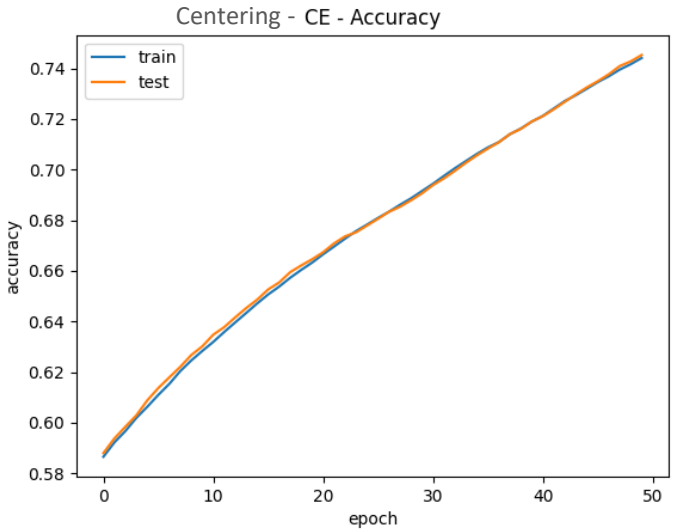
3. Η **Τυποποίηση** (standardization) είναι ο μετασχηματισμός χαρακτηριστικών με αφαίρεση τον μέσο όρο και διαίρεση με τυπική απόκλιση. $X_{new} = \frac{X - X_{mean}}{\sigma_X}$. Η τυποποίηση μπορεί να είναι χρήσιμη σε περιπτώσεις όπου τα δεδομένα ακολουθούν μια κατανομή Gauss. Ωστόσο, αυτό δεν είναι απαραίτητο να ισχύει. *Η τυποποίηση δεν επηρεάζεται από ακραίες τιμές επειδή δεν υπάρχει προκαθορισμένο εύρος μετασχηματισμένων χαρακτηριστικών.*

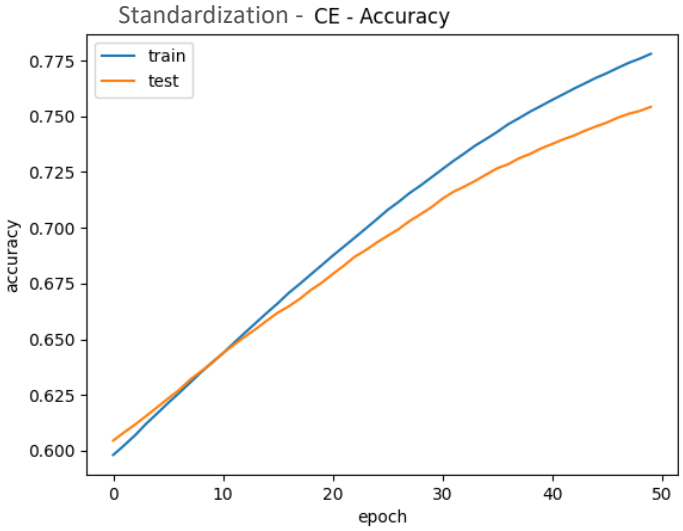
****** Πρακτικά, ο συνδυασμός του κεντραρίσματος και κανονικοποίησης έχει ώς αποτέλεσμα την τυποποίηση.***

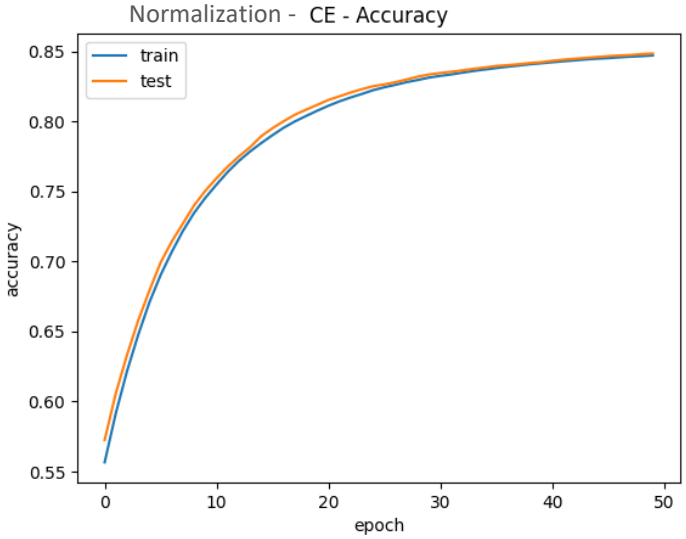
A1. Προεπεξεργασία και Προετοιμασία δεδομένων

- Τα παρακάτω πειράματα έγιναν για να δείξουν τη συμπεριφορά του αλγορίθμου ανά είδος αρχικής προεπεξεργασίας των δεδομένων εισόδου Τυπικά, τα πειράματα έγιναν με 20 νευρώνες στο κρυφό επίπεδο, 50 epoch και binary CE ως loss function.









Α1. Προεπεξεργασία και Προετοιμασία δεδομένων

Πόρισμα 1.1:

Απο το παραπάνω πείραμα εύκολα βλέπουμε οτι η **κανονικοποίηση** αποφέρει καλύτερα. Αυτό δε θα έπρεπε να μας παραξενεύει καθώς τα δεδομένα εισόδου είναι όλα θετικά, και στη περίπτωση των outliers, όπου η διαφορά με τη πληθώρα των μηδενικών γίνεται μεγάλη (τάξη του 10^2), αυτή εξαλείφεται.

Πόρισμα 1.2:

Η **τυποποίηση** και το **κεντράρισμα** εισάγουν αρντητικές τιμές εισόδων. Αυτό δεν είναι επιθυμητό για τον αλγόριθμο διότι οι αρνητικές τιμές συχνότητας επαναλήψεων δεν έχουν πρακτικό νόημα. Ωστόσο, στο σημείο αυτό να αναφέρουμε, οτι η δεύτερη καλύτερη επιλογή είναι να μην εφαρμόσουμε κάποιο μετασχηματισμό στα δεδομένα. Ούτε αυτό θα έπρεπε να μας παραξενέψει καθώς τα έγγραφα έχουν υποστεί αφαίρεση των stopwords και stemming. Μεγάλες συχνότητες των λέξεων προσδίνουν μεγάλη βαρύτητα, το οποίο είναι λογικό.

Α2. Επιλογή αρχιτεκτονικής

α) Κάθε μια απο τις μετρικές αυτές, είναι μια μέθοδος αξιολόγησης του πόσο καλά μοντελοποιεί ο αλγόριθμός το σύνολο δεδομένων. Εάν οι προβλέψεις είναι εντελώς «εκτός», η συνάρτηση κόστους θα παράγει μεγαλύτερο αριθμό. Αν είναι αρκετά καλά, θα βγάζει μικρότερο αριθμό. Αυτός ο αριθμός είναι πολύ σημαντικός γιατί μας βοηθάει να εντοπίσουμε τοπικά ελάχιστα. Στη περίπτωση του **Cross-Entropy,** έχουμε ένα προβλεπτικό **ταξινομητή**, όπου μικρές πιθανότητες πρόβλεψης τιμωρούνται περισσότερο. Στο **MSE**, έχουμε προβλεπτική **παλινδρόμηση**, όπου τα μεγαλύτερα σφάλματα είναι αυτά που «τιμωρούνται» περισσότερο λόγω του τετραγώνου. Η ταξινόμηση προβλέπει μία διακριτή ετικέτα κλάσης. Η παλινδρόμηση προβλέπει μια συνεχή ποσότητα. Ένας αλγόριθμος ταξινόμησης μπορεί να προβλέψει μια συνεχή τιμή, αλλά η συνεχής τιμή έχει τη μορφή πιθανότητας για μια ετικέτα κλάσης. Ένας αλγόριθμος παλινδρόμησης μπορεί να προβλέψει μια διακριτή τιμή, αλλά η διακριτή τιμή με τη μορφή μιας ακέραιας ποσότητας. Τέλος, ο τρόπος με τον οποίο αξιολογούμε τις προβλέψεις ταξινόμησης και παλινδρόμησης ποικίλλει και δεν επικαλύπτεται. Για παράδειγμα, προβλέψεις ταξινόμησης μπορούν να αξιολογηθούν χρησιμοποιώντας την ακρίβεια(accuracy), ενώ οι προβλέψεις παλινδρόμησης όχι. Οι προβλέψεις παλινδρόμησης μπορούν να αξιολογηθούν χρησιμοποιώντας το RMSE, ενώ οι προβλέψεις ταξινόμησης όχι.

β) Η ταξινόμηση πολλαπλών ετικετών μπορεί να υποστηριχθεί απευθείας από νευρωνικά δίκτυα, προσδιορίζοντας τον αριθμό των ετικετών-στόχων που υπάρχουν στο πρόβλημα ως τον αριθμό των κόμβων στο επίπεδο εξόδου. Στη περίπτωση μας, το πρόβλημα έχει είκοσι (20) πιθανές ετικέτες εξόδου (κλάσεις), άρα για το επίπεδο εξόδου του νευρωνικού δικτύου απαιτούνται ***20 κόμβοι/νευρώνες εξόδου***.

γ) Κάθε κόμβος στο κρυφό επίπεδο θα χρησιμοποιεί την **ReLU** συνάρτηση ενεργοποίησης. Τα χαρακτηριστικά της μας φαίνοται ιδιαίτερα χρήσιμα στο συγκεκριμένο πρόβλημα καθώς οι τιμές των διανύσματων εισόδου αποτελούνται απο διακριτές στο εύρος [0, k], με $k \in N$, δίνοντας έτσι στην εκάστοτε τιμή το βάρος που της αναλογεί, καθώς λέξη με μεγάλη συχνότητα έχει νόημα να εισαχθεί στο δίκτυο με μεγαλύτερο βάρος.

Α2. Επιλογή αρχιτεκτονικής

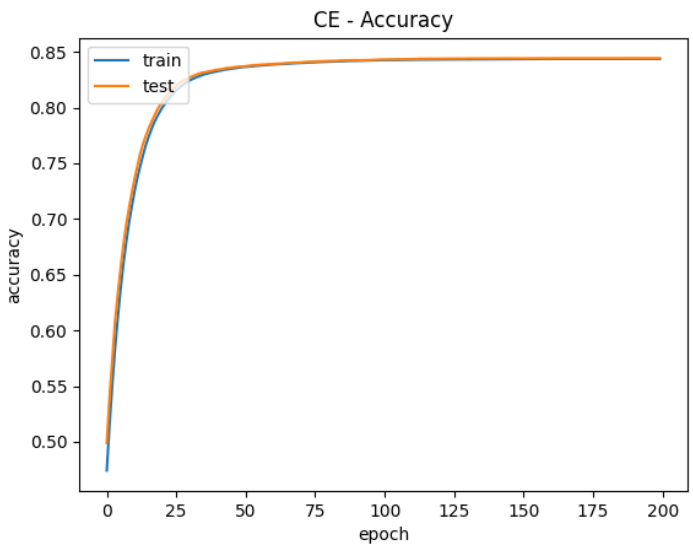
δ) Κάθε κόμβος στο επίπεδο εξόδου πρέπει να χρησιμοποιεί την **σιγμοειδή** συνάρτηση ενεργοποίησης. Ο βασικός λόγος είναι ότι οι πιθανότητες που παράγονται από τη σιγμοειδή είναι ανεξάρτητες και δεν περιορίζονται στο να αθροιστούν στο ‘1’.

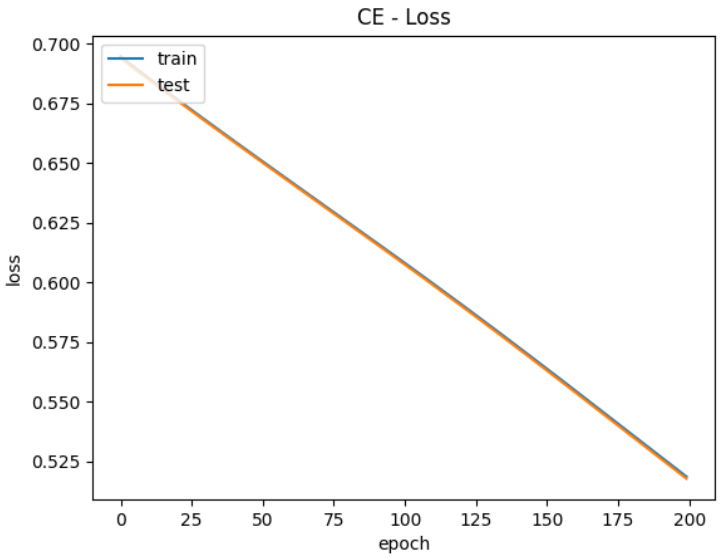
A2. Επιλογή αρχιτεκτονικής

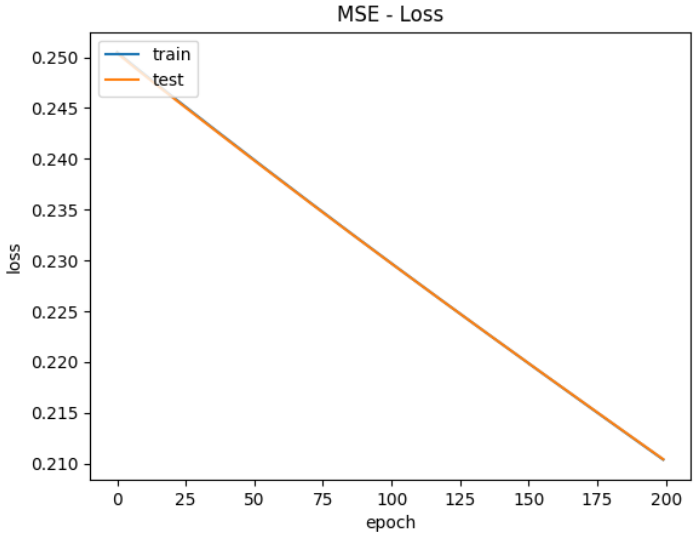
ε)

#Νευρώνων στο κρυφό επίπεδο	CE Loss	MSE	Acc
H1 = O = 20	**0.5180**	**0.2104**	**0.8540**
H1=(I+O)/2=4270	**0.5813**	**0.2214**	**0.8542**
H1 = (I+O) = 8540	**0.5704**	**0.2173**	**0.8531**

- **20 νευρώνες κρυφού επιπέδου**

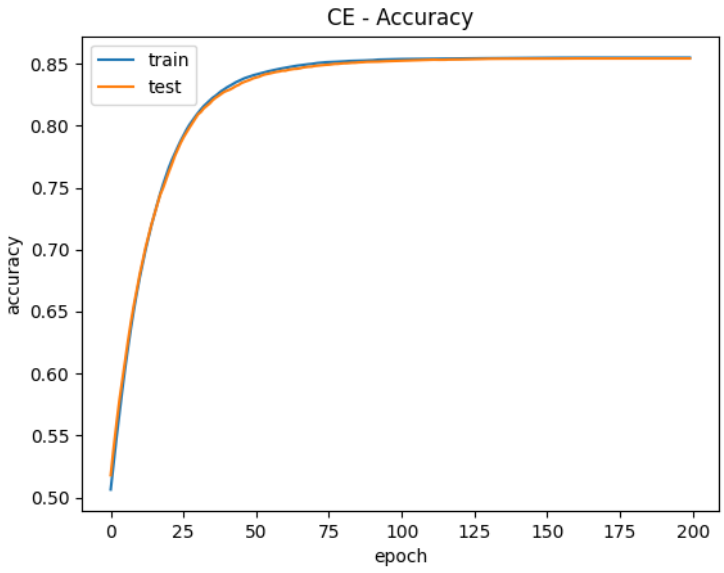


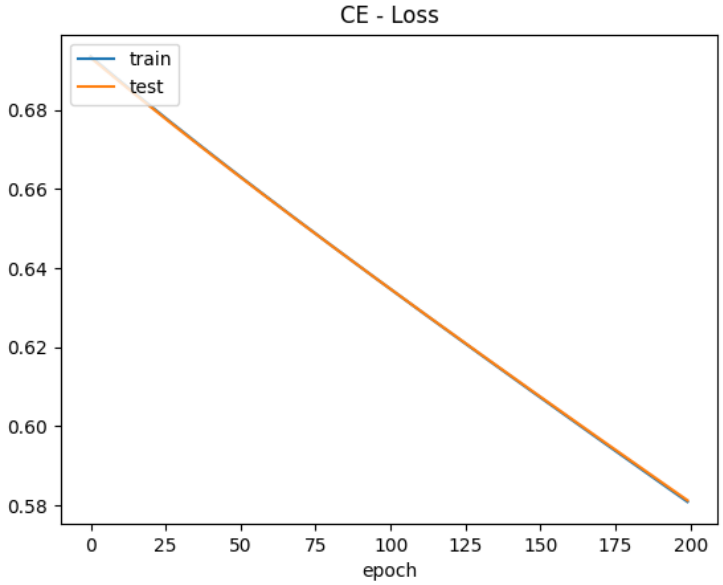


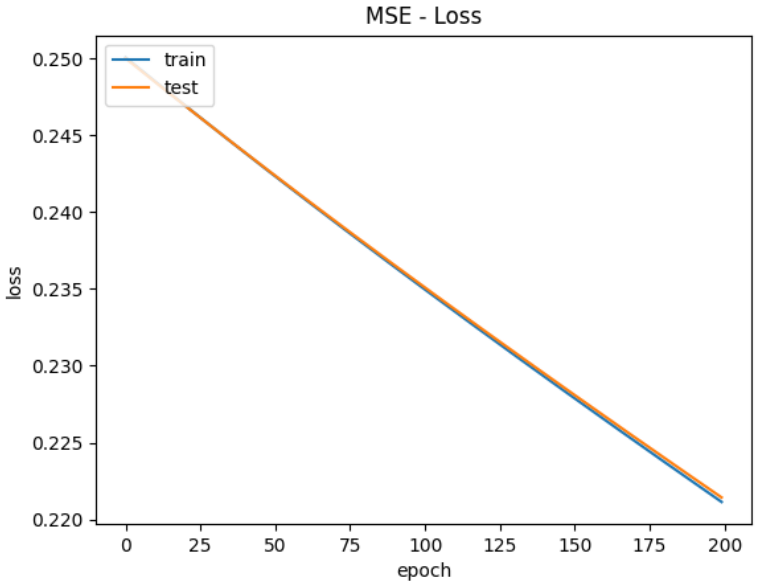


A2. Επιλογή αρχιτεκτονικής

- **4270 νευρώνες κρυφού επιπέδου**

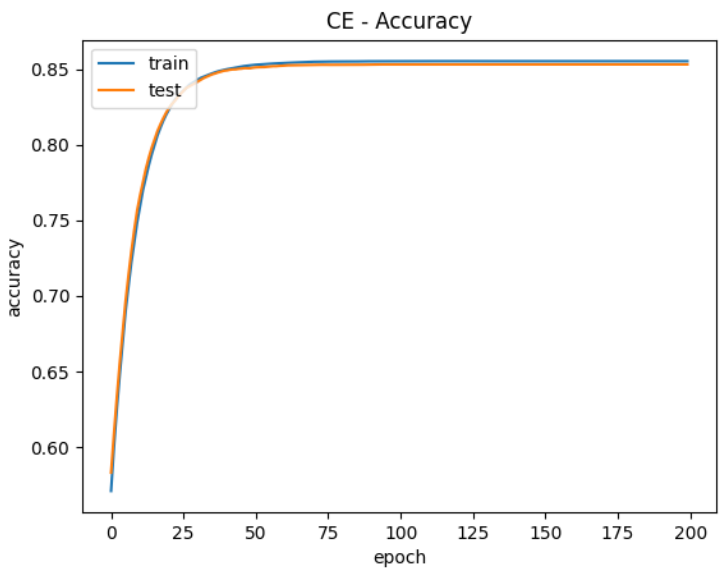


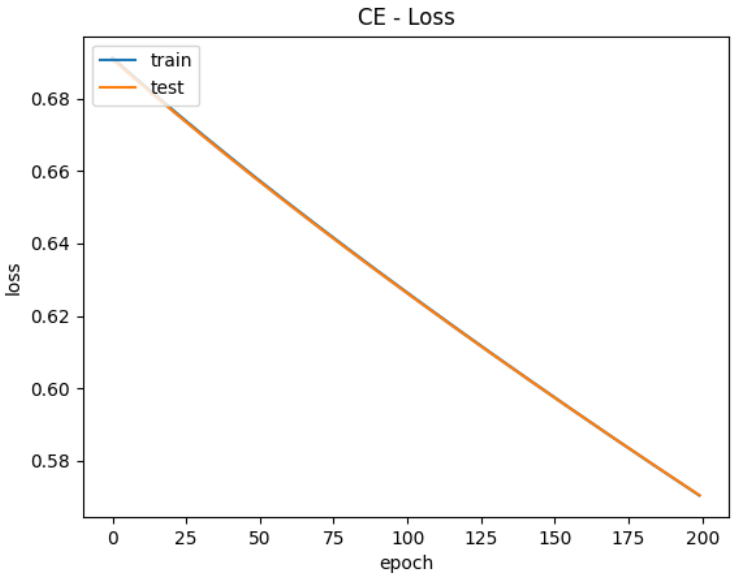


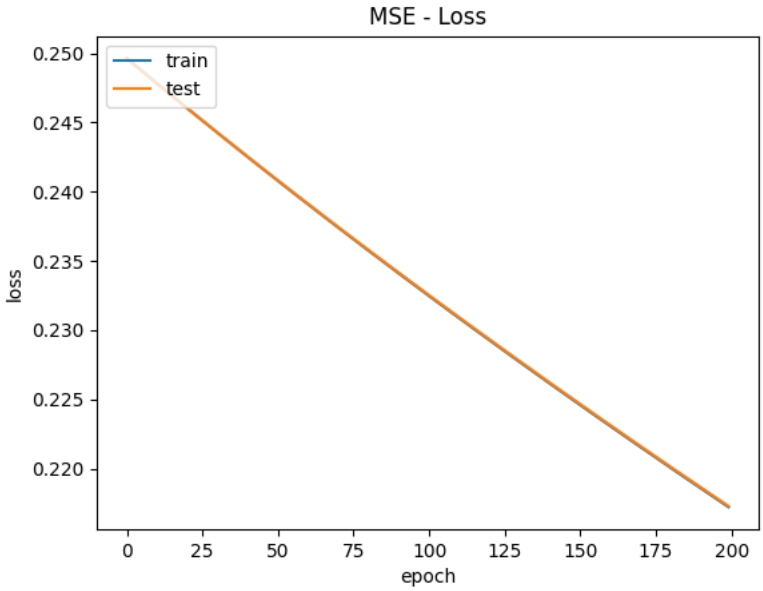


A2. Επιλογή αρχιτεκτονικής

- **8540 νευρώνες κρυφού επιπέδου**







Α2. Επιλογή αρχιτεκτονικής

Πόρισμα 2.1:

Απο το παραπάνω πείραμα, διαπιστώνεται ότι με είκοσι (20) νευρώνες κρυφου επιπέδου ο αλγόριθμος αποδίδει καλύτερα, καθώς το loss είναι μικρότερο. Πολύ κοντά στη ποσότητα αυτή βρίσκονται οι δύο άλλες τιμές. Για το accuracy, και στις τρείς περιπτώσεις του H1 οι τιμές είναι πάρα πολύ κοντά.

Πόρισμα 2.2:

Όσον αφορά τη συνάρτηση κόστους, παρατηρείται ότι η CE είναι πιο «αυστηρή» απο την MSE. Αυτο συμβαίνει γιατί οι έξοδοι που θέλουμε να εκτιμήσουμε είναι τιμές στο εύρος [0, 1], και ο λογάριθμος είναι πολύ πιο ευαίσθητος σε αυτές απο το τετράγωνο. Ωστόσο, και οι δυο συναρτήσεις loss φαίνεται να τοποθετούν μια αρνητική γραμμική κλίση loss στον αλγόριθμο.

Πόρισμα 2.3:

Παρατηρούμε ότι η σύγκλιση του αλγορίθμου με CE συνάρτηση κόστους φαίνεται να βοηθά τη σύγκλιση του αλγορίθμου πιο γρήγορα απο την MSE και στις τρείς (3) περιπτώσεις του H1, καθώς για τον ίδιο αριθμό εποχών το ποσοστό μείωσης του loss είναι μεγαλύτερο.

Πόρισμα 2.4:

Ωστόσο στο σημείο αυτό να πούμε οτι και στις τρείς περιπτώσεις ο αλγόριθμος υπερεκπεδεύεται. Αυτό το καταλαβαίνουμε απο τις γραφικές παραστάσεις του Training Loss, που φαίνεται να συνεχίζει να μειώνεται με την εμπειρία.

Α2. Επιλογή αρχιτεκτονικής

❖ **Το πείραμα αυτό θα συνεχιστεί με H1 = 20**

στ)

#Νευρώνων στο κρυφό επίπεδο	**CE Loss**	**MSE**	**Acc**
H2 = 15	**0.6270**	**0.2327**	**0.8420**
H2 = 20	**0.6015**	**0.2245**	**0.8193**
H2 = 40	**0.6054**	**0.2278**	**0.8527**

Πόρισμα 2.4:

Απο το παραπάνω πίνακα προκύπτει οτι η CE συμπεριφέρεται καλύτερα (πάλι). Όσον αφορά τον αριθμό νευρώνων του $2^{ου}$ κρυφού επιπέδου φαίνεται για H1=H2 να αποδίδει καλύτερα. Ωστόσο, το $2^{ο}$ κρυφό επίπεδο δε προσδίδει ουσιαστικό όφελος.

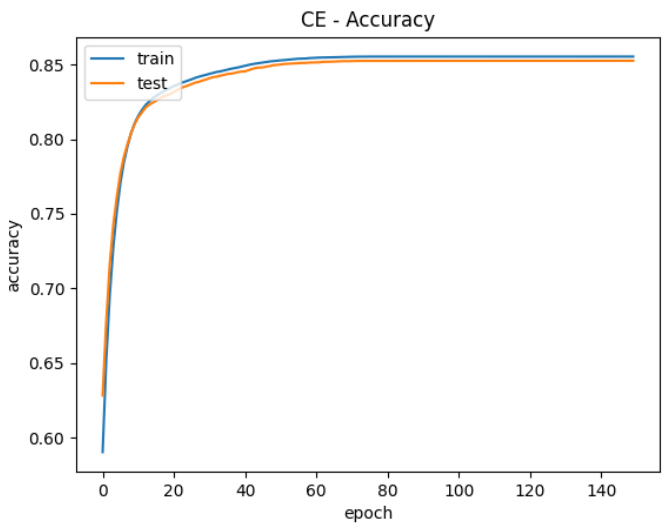
ζ) Απο τις γραφικές σύκλισης, παρατηρούμε οτι ένα καλό κριτήριο τερματισμού που θα θέσουμε σε κάθε fold είναι το πρώιμο σταμάτημα οταν το loss του validation set σταματάει να μειώνεται. Στη περίπτωση που τα train, test του κάθε fold είναι άνισα μεταξύ τους, εντοπίζεται το «μεγαλύτερο» και βάσει αυτό τα υπόλοιπα κάνουν padding με τη τελευταία τους τιμή.

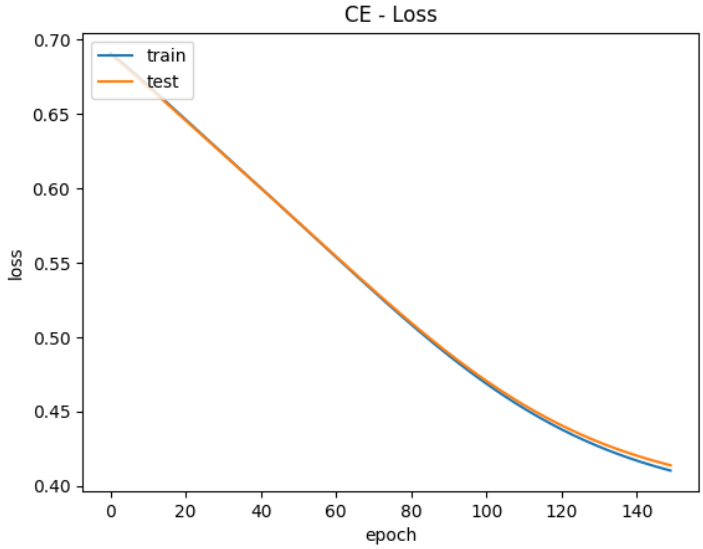
Η πρώιμη διακοπή θα μπορούσε να χρησιμοποιηθεί με διασταυρούμενη επικύρωση k-fold, αν και δεν συνιστάται. Η διαδικασία διασταυρούμενης επικύρωσης k-fold εκτιμά το σφάλμα γενίκευσης ενός μοντέλου, επανατοποθετώντας και αξιολογώντας το επανειλημμένα σε διαφορετικά υποσύνολα ενός συνόλου δεδομένων. Η πρώιμη διακοπή παρακολουθεί το σφάλμα γενίκευσης ενός μοντέλου και διακόπτει την εκπαίδευση όταν το σφάλμα γενίκευσης αρχίζει να υποβαθμίζεται. Δηλαδή, είναι σε αντίθεση επειδή η διασταυρούμενη επικύρωση προϋποθέτει ότι δεν γνωρίζουμε το σφάλμα γενίκευσης και η πρόωρη διακοπή προσπαθεί να μας δώσει το καλύτερο μοντέλο που βασίζεται στη γνώση του σφάλματος γενίκευσης.

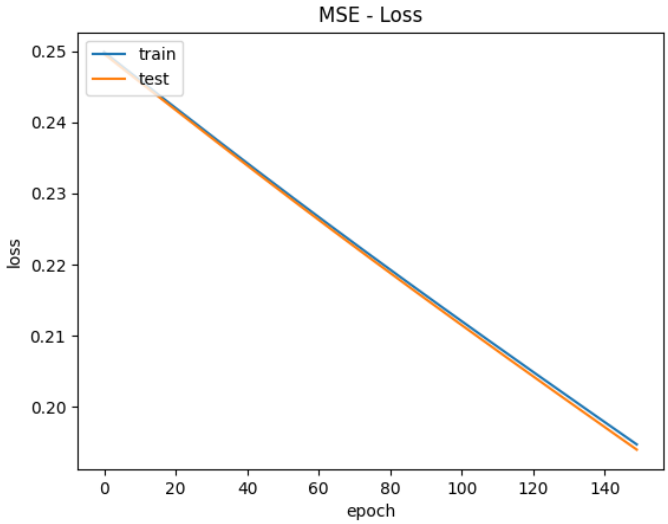
A3. Μεταβολές στον ρυθμό εκπαίδευσης και σταθεράς ορμής

η	m	CE Loss	MSE	Acc
0.001	0.2	**0.4809**	**0.1940**	**0.8543**
0.001	0.6	**0.3892**	**0.1302**	**0.8534**
0.05	0.6	**0.3172**	**0.1068**	**0.8727**
0.01	0.6	**0.3713**	**0.1165**	**0.8531**

❖ (0.001,0.2)

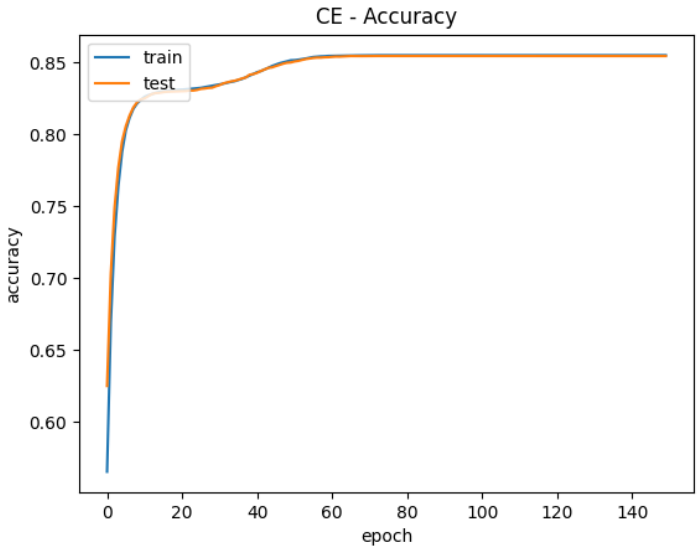


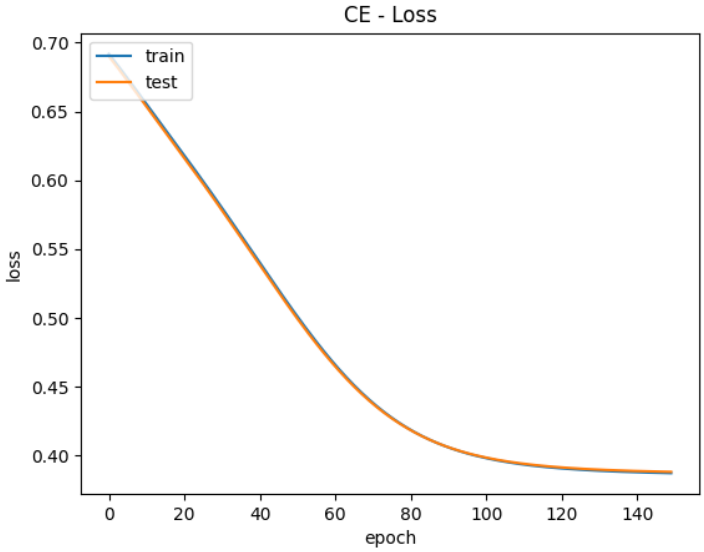


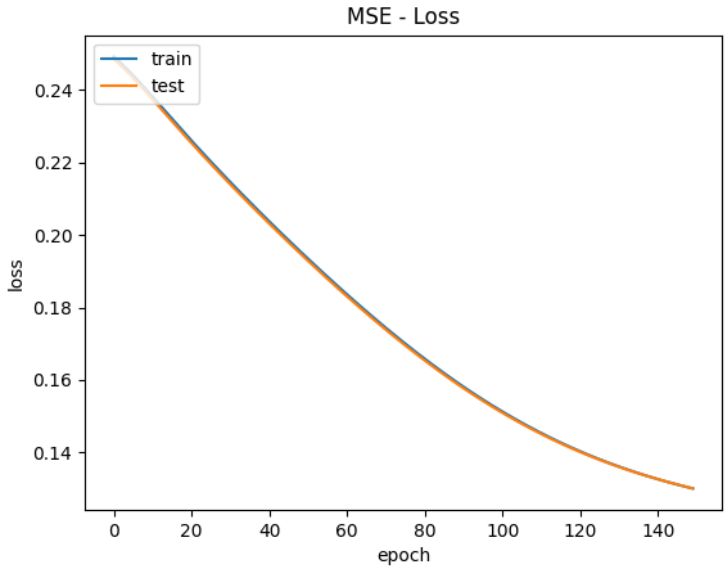


A3. Μεταβολές στον ρυθμό εκπαίδευσης και σταθεράς ορμής

- (0.001, 0.6)

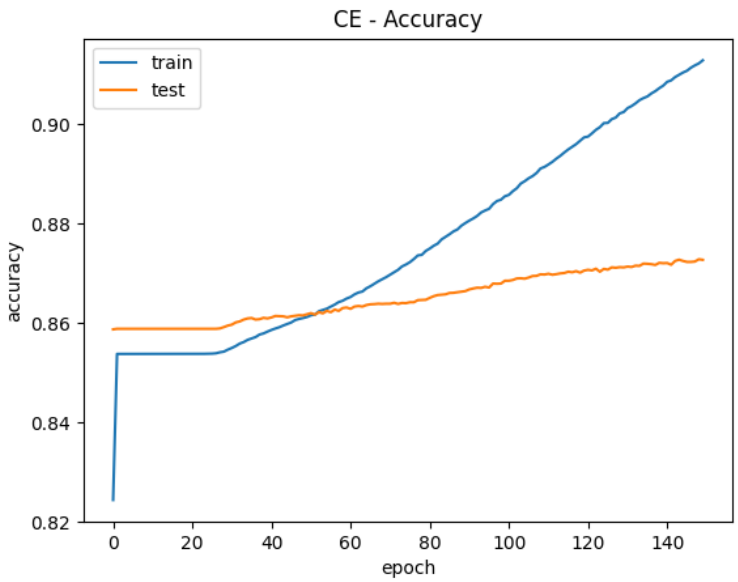


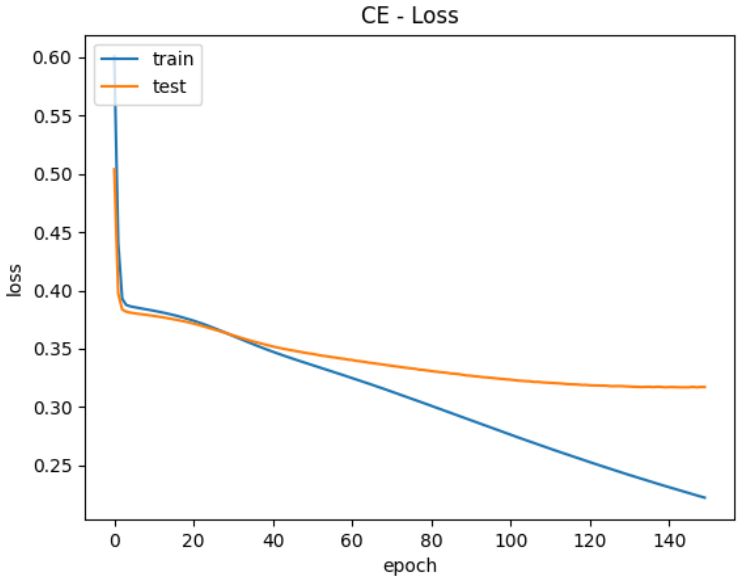


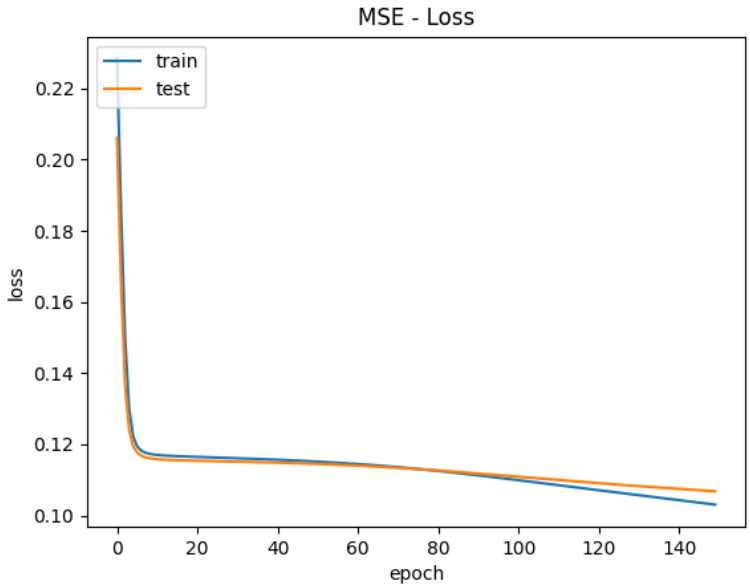


A3. Μεταβολές στον ρυθμό εκπαίδευσης και σταθεράς ορμής

❖ (0.05, 0.6)

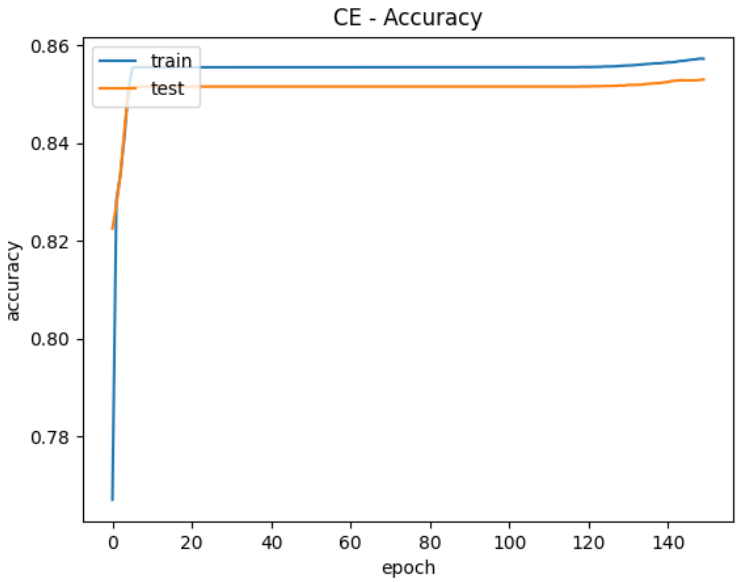


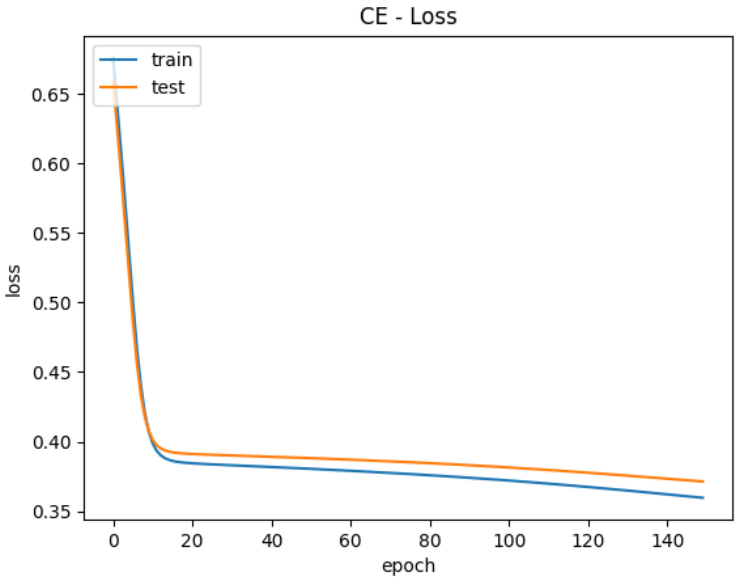


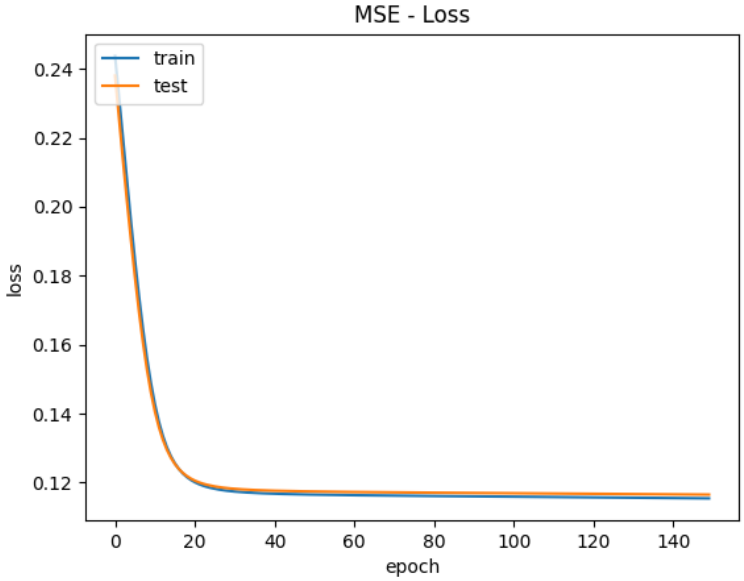


A3. Μεταβολές στον ρυθμό εκπαίδευσης και σταθεράς ορμής

❖ (0.01, 0.6)







Α3. Μεταβολές στον ρυθμό εκπαίδευσης και σταθεράς ορμής

- Εάν θέσουμε την υπερπαράμετρο ορμής ‘1’ ή έστω πολύ κοντά στο 1 (π.χ. 0,99999) όταν χρησιμοποιήσουμε σαν optimizer SGD, τότε ο αλγόριθμος πιθανότατα θα αυξήσει πολύ την ταχύτητα, ελπίζοντας ότι θα κινηθεί περίπου προς το καθολικό ελάχιστο, αλλά η ορμή του θα το μεταφέρει ακριβώς πέρα από το ελάχιστο. Μετά θα επιβραδύνει και θα επανέλθει, θα επιταχύνει ξανά, θα ξεπεράσει ξανά και ούτω καθεξής. Μπορεί να ταλαντωθεί με αυτόν τον τρόπο πολλές φορές πριν συγκλίνει, επομένως συνολικά θα χρειαστεί πολύ περισσότερος χρόνος για να συγκλίνει από ό,τι με μια μικρότερη τιμή ορμής.

Πόρισμα 3.1:

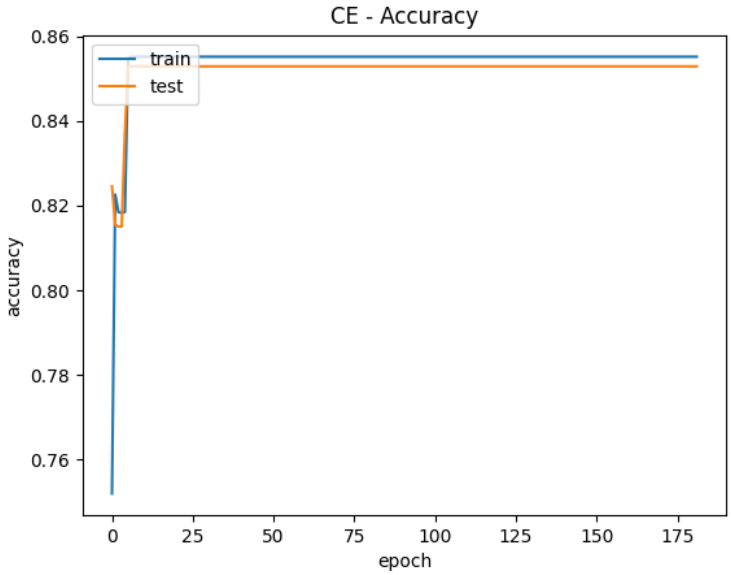
Από τα παραπάνω πειράματα, αρχικά επαληθεύεται ότι η CE συνάρτηση loss συγκλίνει ταχύτερα απο την MSE. Επίσης, οι επιλογές των υπερπαραμέτρων ρυθμού μάθησης και ορμής, αποδεικνύουν ότι ο αγλόριθμος ευνοείται κατά πολύ, καθώς βλέπουμε, συγκριτικά και με τα πειράματα του Α2, οτι για 150 εποχές, ο αλγόριθμος συγκλίνει, ενώ πριν οχι. Για (0.001, 0.2) δε παρατηρούμε ιδιαίτερη βελτίωση. Για m=0.6 η διαφορά βελτίωσης είναι εμφανής για όλες τις επιλογές του η, **με καλύτερη απόδοση να να επιτυγχάνεται για το συνδυασμό (0.01, 0.6) καθώς η καμπύλες του loss έχουν σταθεροποιηθεί με κοινές τιμές**. Στη περίπτωση (0.05, 0.6), η καθοδική πορεία μόνο του training set υποδυκνύει πώς πάλι ο αλγόριθμος υπερεκπεδεύεται.

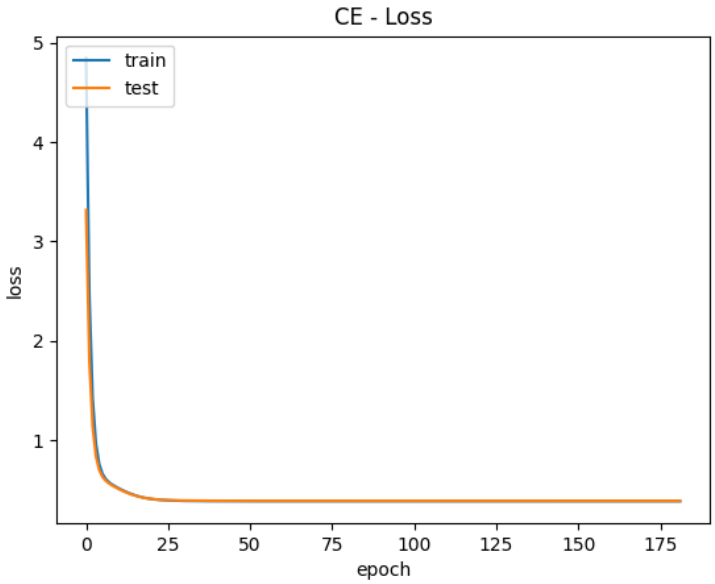
A4. Ομαλοποίηση

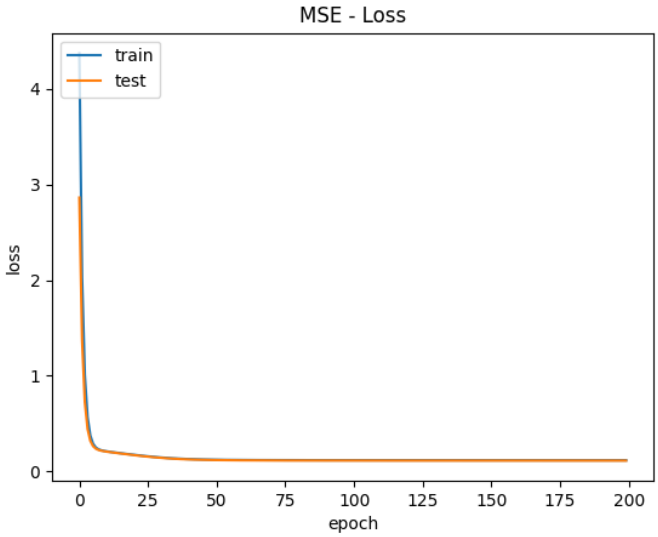
❖ **Το πείραμα αυτό θα συνεχιστεί με η=0.01 και m=0.6**

Συντελεστής Φθοράς	CE Loss	MSE	Acc (CE/MSE)
0.1	**0.3829**	**0.1155**	**0.8556**
0.5	**0.3910**	**0.1161**	**0.8531**
0.9	**0.3783**	**0.1161**	**0.8591**

- r=0.1

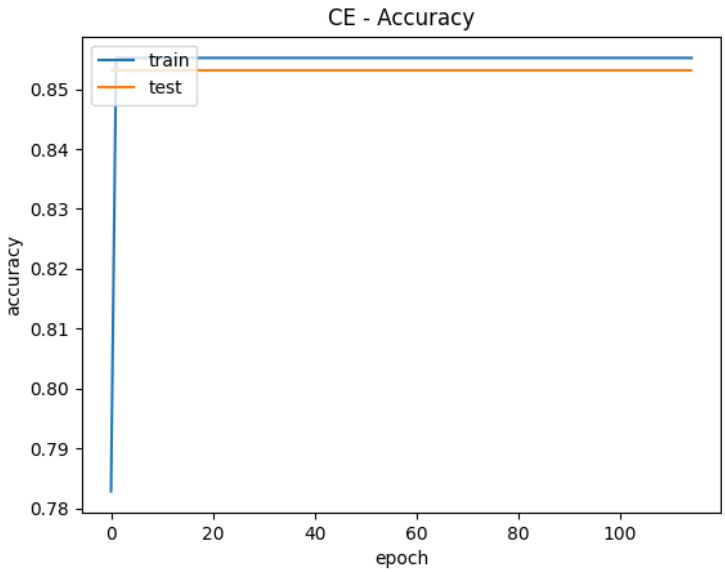


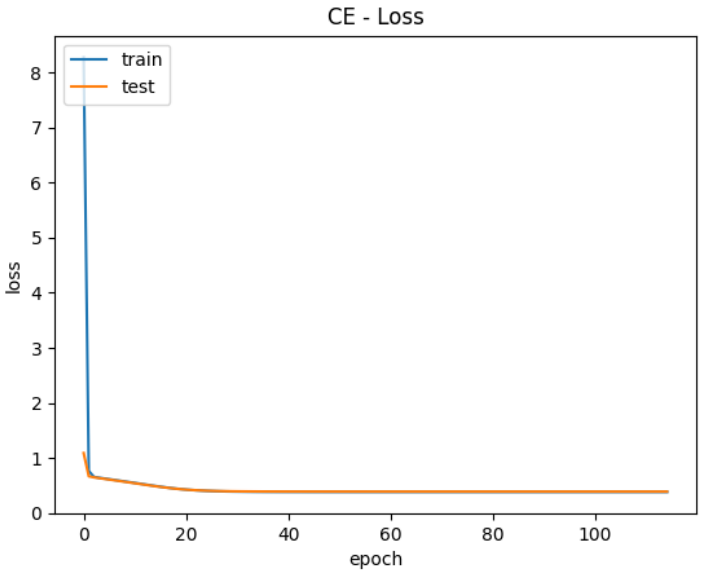


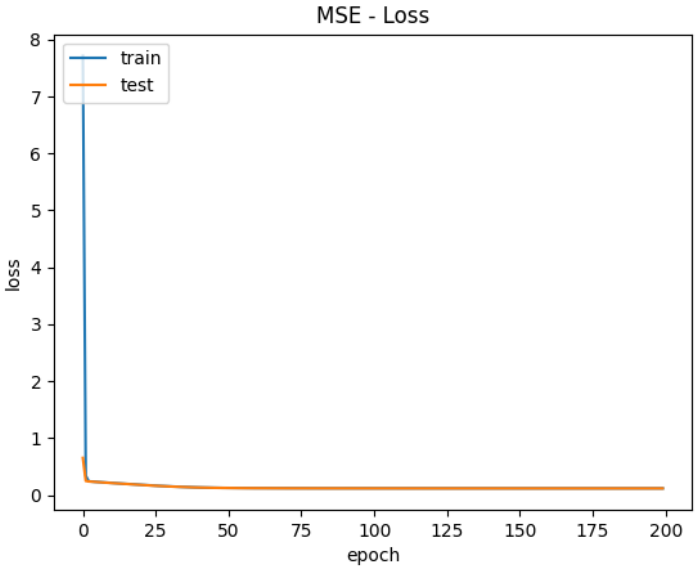


A4. Ομαλοποίηση

- r=0.5

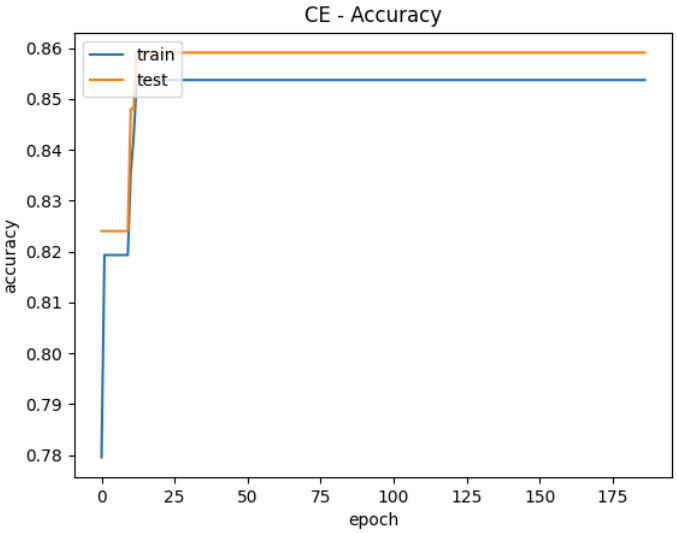


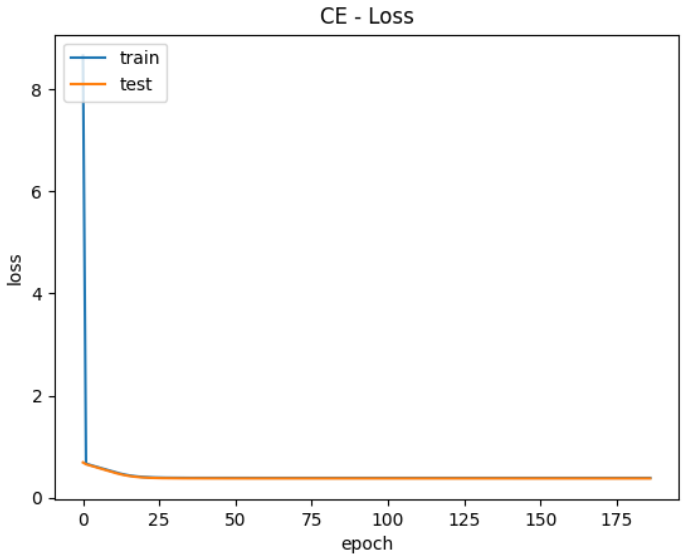


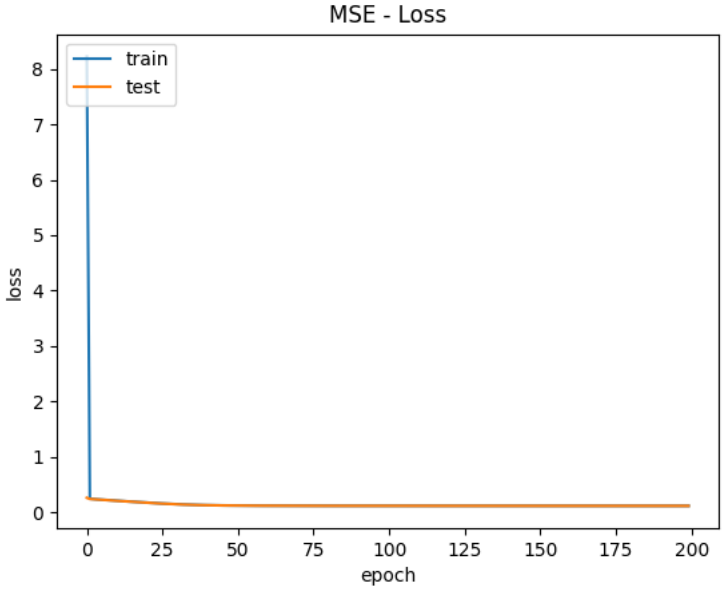


A4. Ομαλοποίηση

- r=0.9



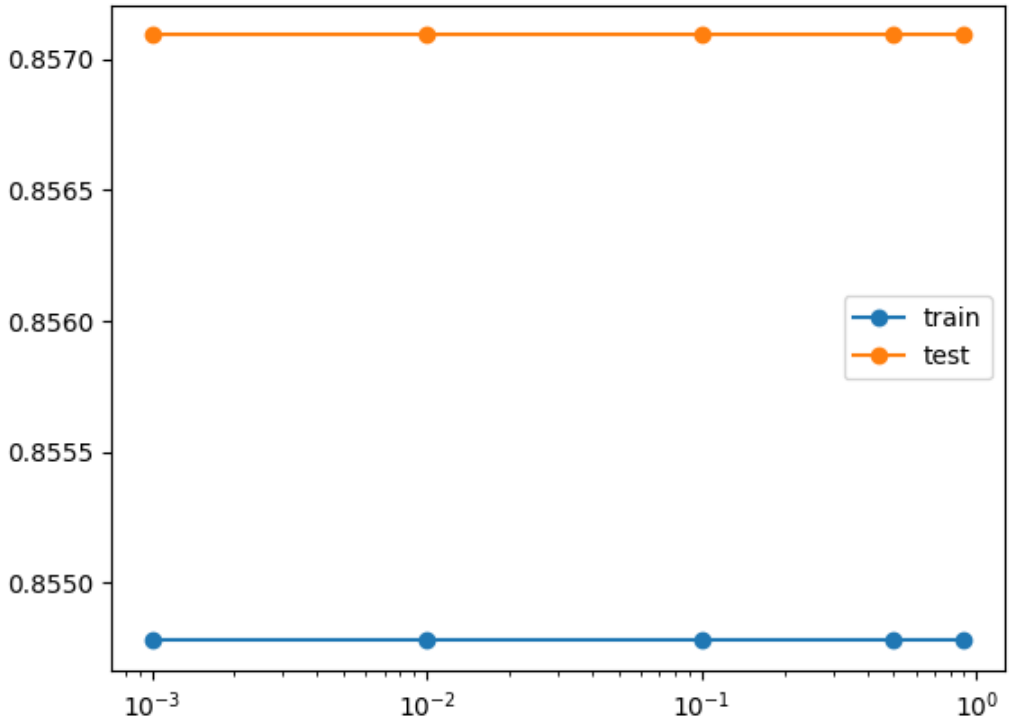




A4. Ομαλοποίηση

Πόρισμα 4.1:

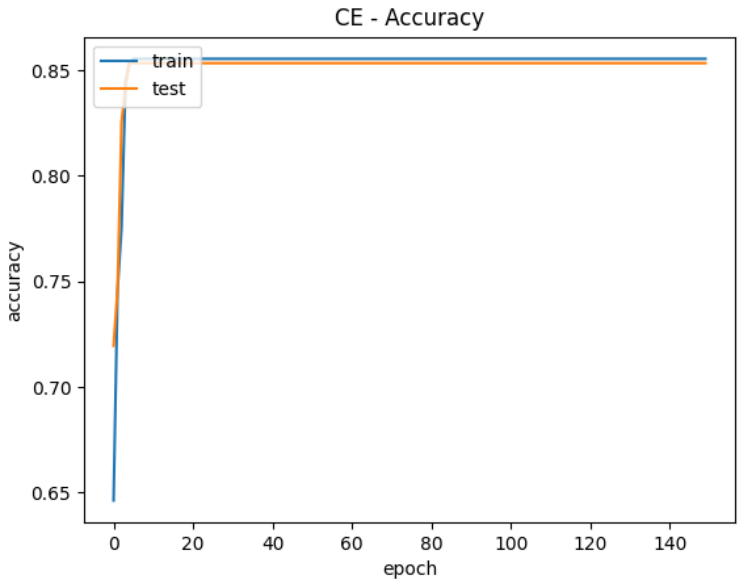
Σε όλες τις περιπτώσεις παρατηρείται οτι μετά τη προσθήκη φθοράς βαρών, στις πρώτες εποχές, το training loss είναι πάρα πολύ αυξημένο. Επίσης, απο τη καμπύλη του test loss φαίνεται οτι επιτυγχάνουμε ελαφρώς καλύτερη γενίκευση, διότι απο τα πρώτα κιόλας δείγματα το loss είναι πολύ χαμηλότερο του training και μετα απο λίγο σταθεροποιούνται απο κοινού. Τέλος, το παρακάτω γράφημα εξηγεί λίγο καλύτερα τις μετρήσεις απο το παραπάνω πινακάκι και μας επαληθεύει οτι για r=0.9 επιτυγχάνουμε, ελαφρώς, τη καλύτερη γενίκευση.

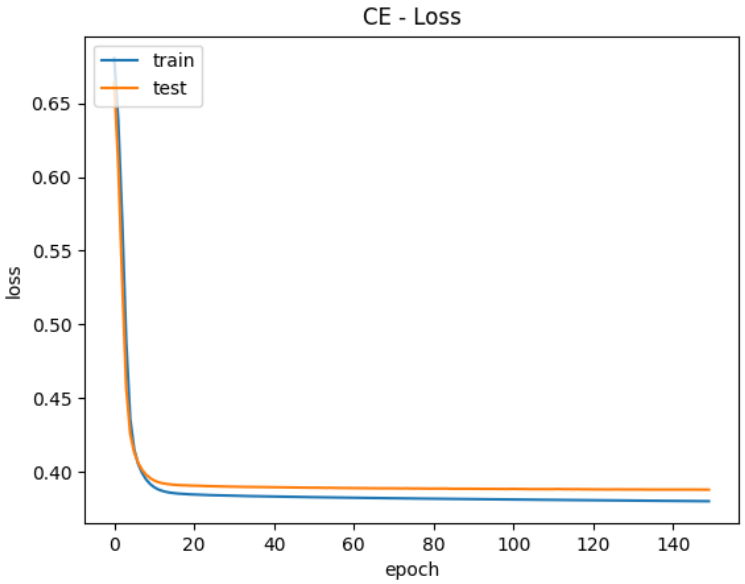


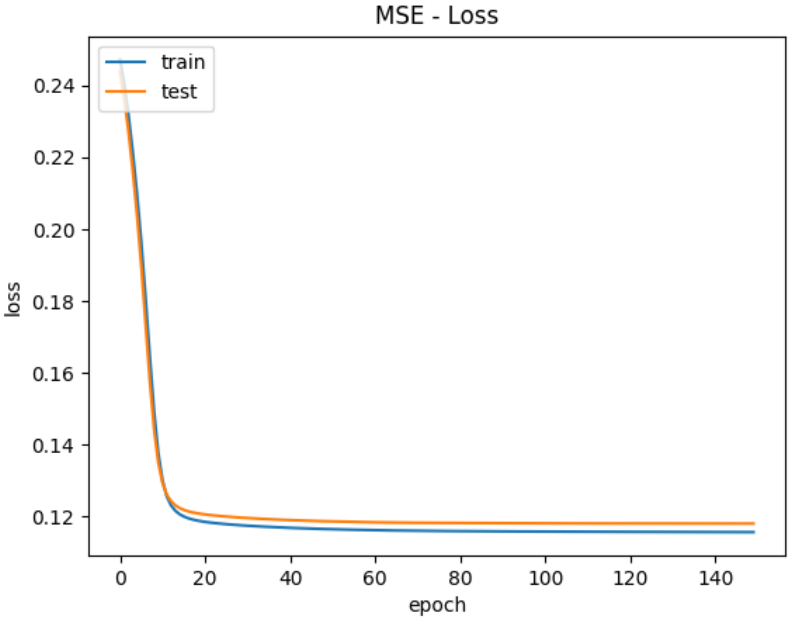
A5. Ενσωματώσεις Λέξεων *[προαιρετικό ερώτημα]*

a, b)

#Νευρώνων στο κρυφό επίπεδο	**CE Loss**	**MSE**	**Acc**
H1 = 20	**0.3270**	**0.1027**	**0.8521**







Α5. Ενσωματώσεις Λέξεων *[προαιρετικό ερώτημα]*

Πόρισμα 5.1:

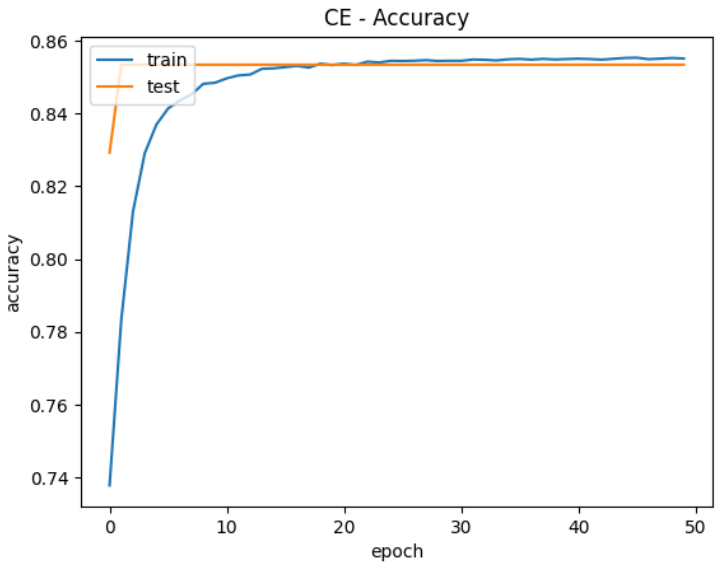
Συγκρίνοντας το Word Embedding μοντέλο με αυτό του Α.2, εύκολα παρατηρείται η συντριπτική διαφορά. Το loss σε αυτό το μοντέλο είναι ~36% χαμηλότερο σε μόλις 20 εποχές εκπαίδευσης, όταν το μοντέλο του Α.2 δε κατάφερε να συκλίνει στις 200 εποχές. Ωστόσο, απο τους πίνακες επιδόσεων και γραφικών σύγκλισης των WE, Α.4, το WE μοντέλο φαίνεται αντάξιο του Α.4 με τις βελτιστοποιημένες υπερπαραμέρους των η, m, *r*.

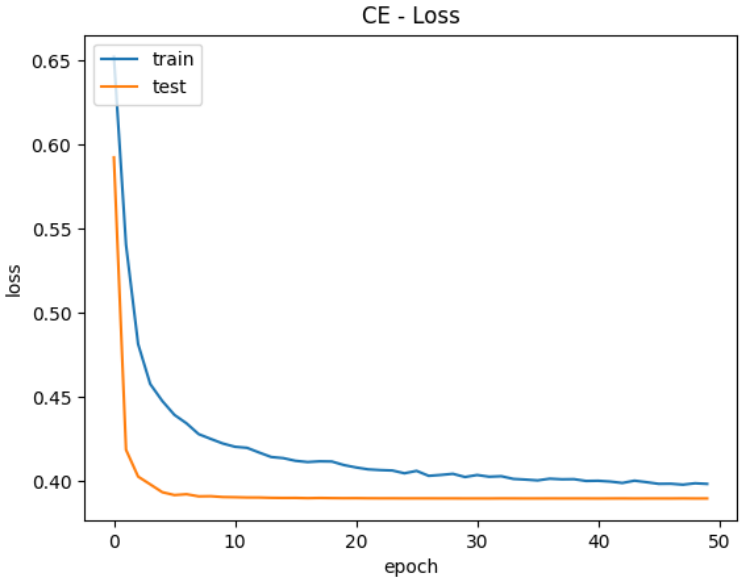
A5. Ενσωματώσεις Λέξεων *[προαιρετικό ερώτημα]*

c)

# Νευρώνων LSTM1 hidden	**# νευρώνων LSTM2 hidden**	**CE**	**MSE**	**Acc**
H1 = 32	H2 = 0	0.3390	nan	0.8570
H1 = 64	H2 = 0	0.3390	0.1159	0.8570
H1 = 32	H2 = 32	0.3390	0.1160	0.8570
H1 = 64	H2 = 64	0.3826	0.1164	0.8570

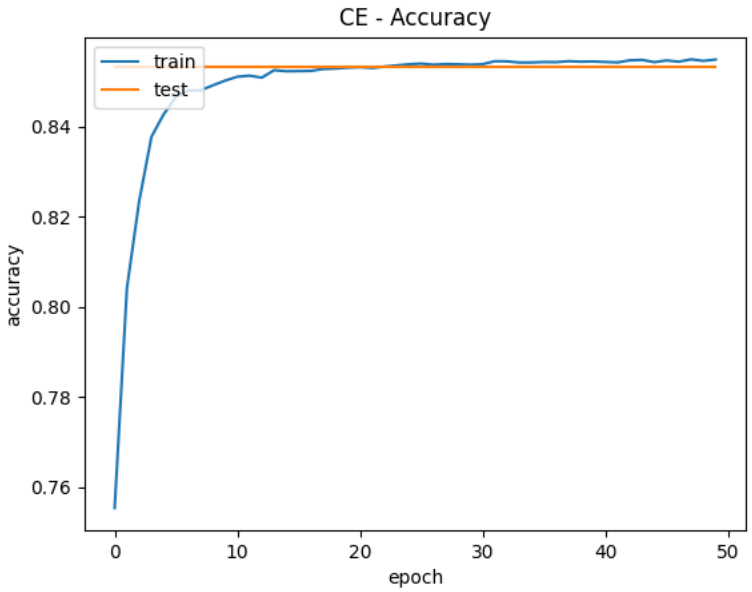
- H1 = 32, H2 = 0

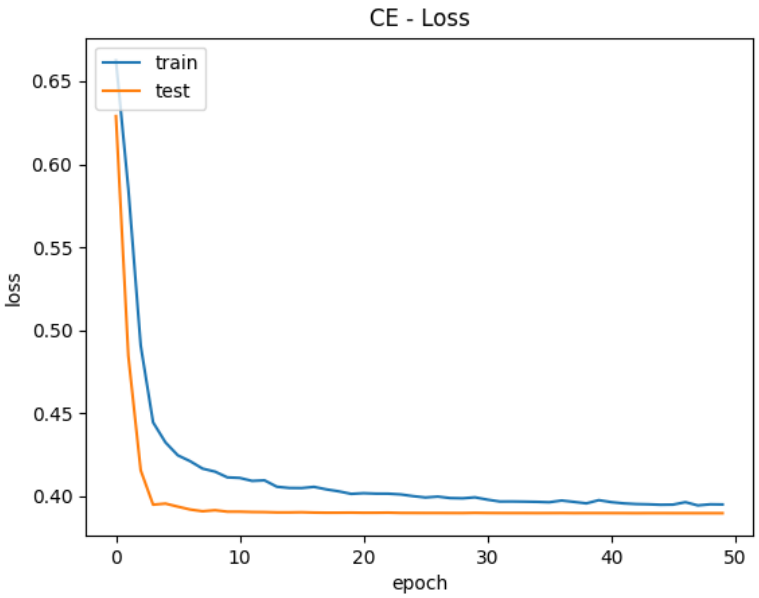


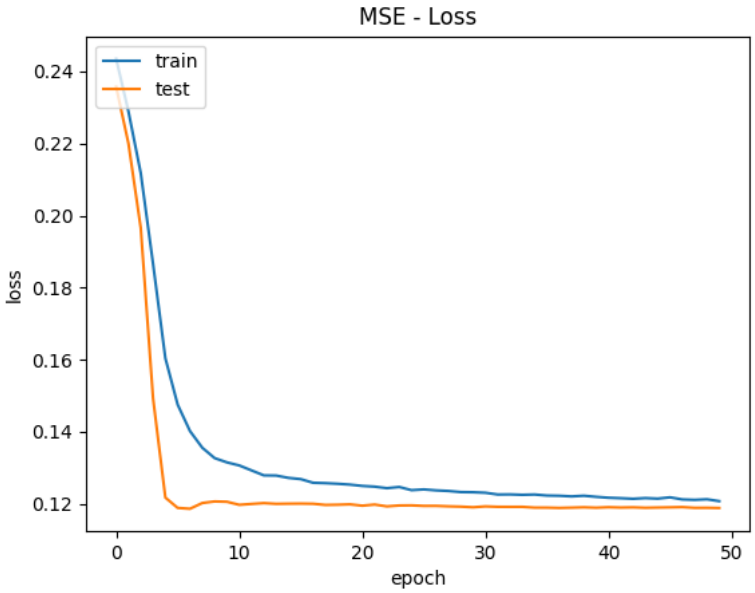


A5. Ενσωματώσεις Λέξεων *[προαιρετικό ερώτημα]*

- H1 = 64, H2 = 0

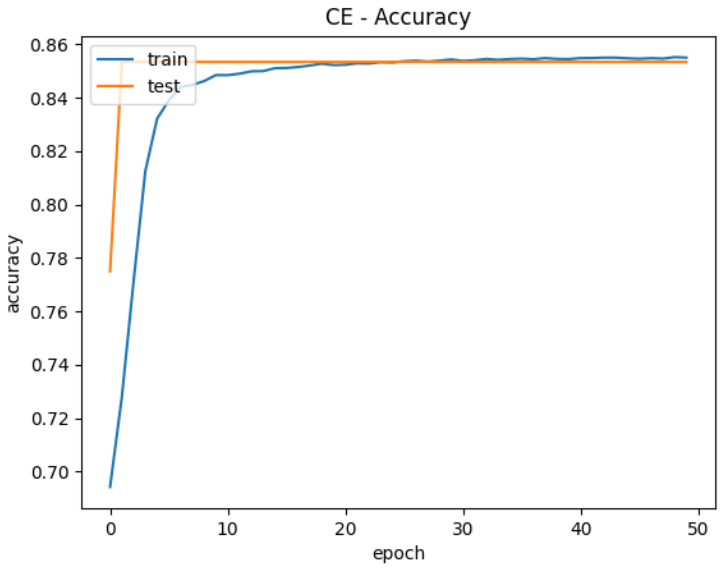


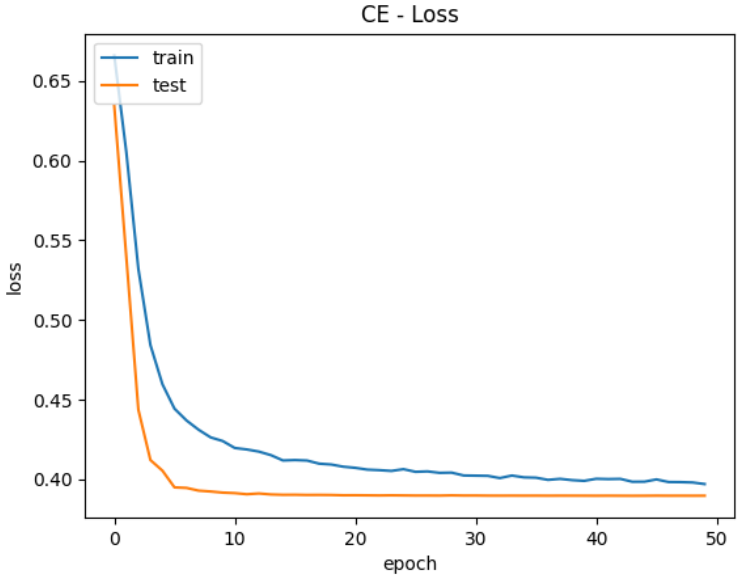


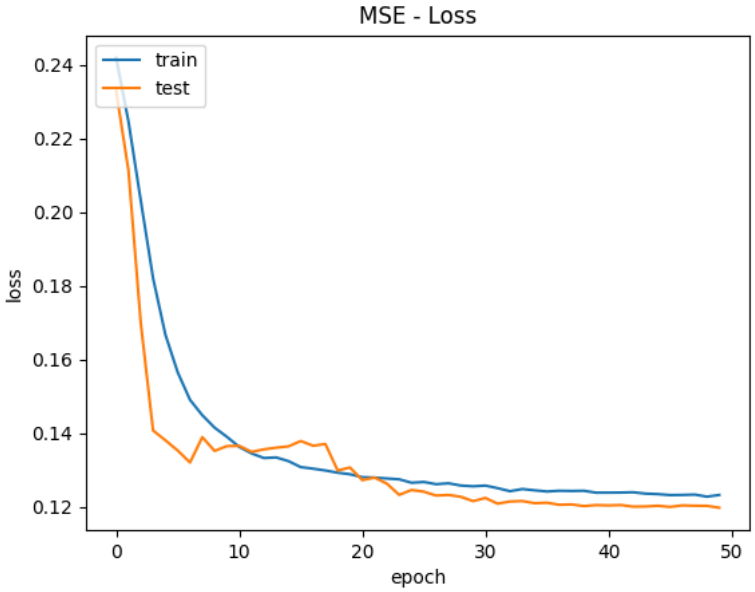


A5. Ενσωματώσεις Λέξεων *[προαιρετικό ερώτημα]*

- H1 = 32, H2 = 32

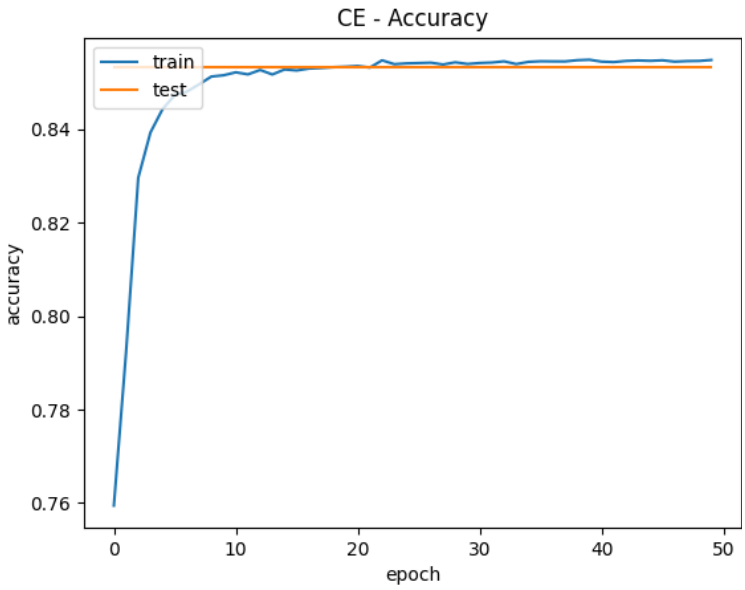


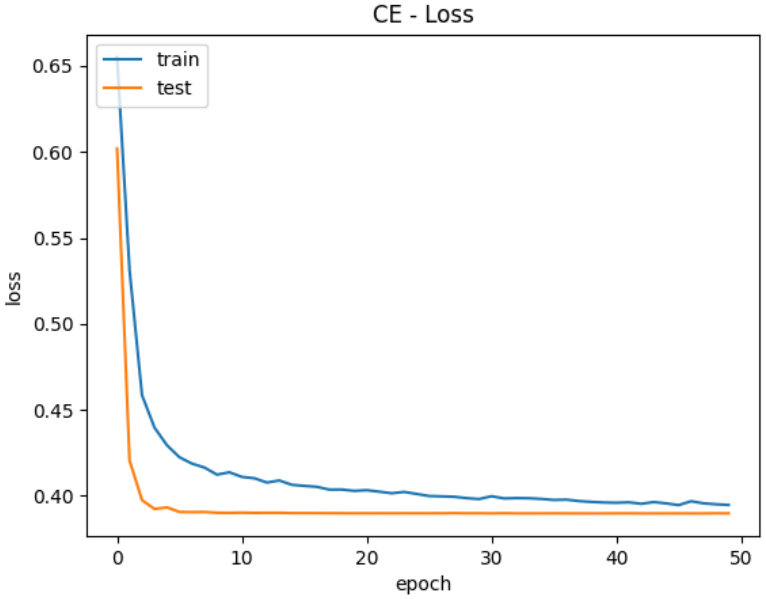


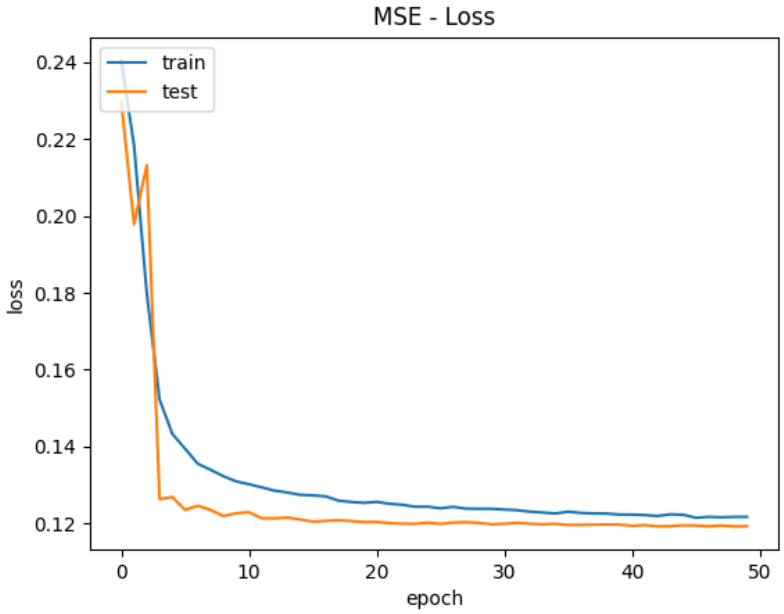


A5. Ενσωματώσεις Λέξεων *[προαιρετικό ερώτημα]*

- H1 = 64, H2 = 64







Α5. Ενσωματώσεις Λέξεων *[προαιρετικό ερώτημα]*

Πόρισμα 5.2:

Αρχικά, ο χρόνος εκμάθησης του νεωρωνικού με LSTM επίπεδα ήταν κατά πολύ πιο αργός και απο τις δυο παραπάνω αρχιτεκτονικές. Αυτό δε πρέπει να μας παραξενεύει ωστόσο καθώς το LSTM είναι τεχνική ανδρομικού νευρωνικού που απειτεί μεγάλο memory-bandwidth.

Ένα βασικό θέμα που προέκυψε στην εκμάθηση του νευρωνικού με LSTM, ειδικά στη περίπτωση του παλινδρομητή, είναι ότι οι τιμές του loss «γίνονταν» nan, μάλλον, λόγω του **exploding gradients** προβλήματος. Το πρόβλημα αυτό διορθώθηκε, πειράζοντας κατάλληλα το clipvalue του optimizer, το batch size, φθορά βαρών ή τη dropout τιμή στα LSTM επίπεδα. Ωστόσο, στη πρώτη περίπτωση του παλινδρομητή το σημείο αυτο δε βρέθηκε.

Πόρισμα 5.3:

Απο άποψη απόδοσεων, όλες οι αρχιτεκτονικές έδωσαν σχεδόν τις ίδιες τιμές με διαφοροποιήσεις στο τρόπο και ταχύτητα σύκλισης. Για παράδειγμα, στις αρχιτεκτονικές με δεύτερο κρυφό επίπεδο, η σύκλιση του loss με CE είναι πιο ομαλή και γρήγορη σε σχέση με αυτή του MSE. Στη περίπτωση τους ενός κρυφού επιπέδου οι δύο loss functions είχα παρόμοια συμπεριφορά με ένα μικρό πλεονέκτημα να τείνει πάλι προς την CE. Συμπερασματικά, μπορούμε να πούμε οτι σε αρχιτεκτονικές LSTM (ίσως γενικότερα RNN), η εκμάθηση με παλινδρομητή να μήν είναι καλή πρακτική.

Πόρισμα 5.4:

Επιπλέον, απο τις γραφικές παρατηρήθηκε ότι η απώλεια επικύρωσης είναι χαμηλότερη από την απώλεια εκπαίδευσης. Σε αυτήν την περίπτωση, υποδεικνύει ότι το σύνολο δεδομένων επικύρωσης μπορεί να είναι ευκολότερο για το μοντέλο να προβλέψει από το σύνολο δεδομένων εκπαίδευσης.

A5. Ενσωματώσεις Λέξεων *[προαιρετικό ερώτημα]*

Πόρισμα 5.5:

Τέλος, όπως ξανά αναφέρθηκε, η εκμάθηση με LSTM είναι πολύ πιο αργή απο τα Embeddings. Ωστόσο, ενώ η αρχιτεκτονική αυτή «κατάφερε» μία βελτίωση απόδοσης μόνο κατά 5e-2 μεγαλύτερη, απο τα πρώτα κιόλας samples εκμάθησης, το νευρωνικό γενικεύει πολύ πιο γρήγορα. Αυτό φαίνεται απο τις γραφικές ακρίβειας.

- ***Πηγές που χρησιμοποιήθηκαν κατά την υλοποίηση της εργασίας***

https://www.geeksforgeeks.org/normalization-vs-standardization/

https://machinelearningmastery.com/multi-label-classification-with-deep-learning/

https://towardsdatascience.com/how-to-choose-the-right-activation-function-for-neural-networks-3941ff0e6f9c

https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/

https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/

https://machinelearningmastery.com/weight-regularization-to-reduce-overfitting-of-deep-learning-models/

https://stackoverflow.com/questions/37232782/nan-loss-when-training-regression-network