ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ
Ακαδημαϊκό Έτος 2023-2024
Εργαστηριακή Άσκηση
Μέρος Α΄
Ονοματεπώνυμο: Ελένη Μαράκη
ΑΜ: 1084534
Έτος: 4ο
Github Repo: https://github.com/ElinaMaraki/compintel

Α1.α.

Για την υλοποίηση του project χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python καθώς φάνηκε η πιο κατάλληλη για την υλοποίηση νευρωνικών δικτύων. Για την διαδικασία της προεπεξεργασίας των δεδομένων χρησιμοποιήθηκε η βιβλιοθήκη sklearn. Το csv αρχείο που δόθηκε περιέχει διάφορα στοιχεία για τις ελληνικές επιγραφές που περιέχονται. Το νευρωνικό δίκτυο που ζητείτε να υλοποιηθεί πρέπει να έχει την δυνατότητα να προβλέψει την ακριβής χρονολογία μίας επιγραφής βάση του κειμένου της. Γίνεται λοιπόν αντιληπτό ότι από τα δεδομένα τα οποία δίνονται τα αναγκαία για την υλοποίηση του τεχνητού νευρωνικού δικτύου είναι τα πεδία “text” το οποίο περιέχει το κείμενο της περιγραφής, καθώς και τα παιδία “date_min” και “date_max” που προσδιορίζουν το χρονικό περιθώριο στο οποίο έχει γραφθεί η επιγραφή.

Αρχικά είναι αναγκαία η προεπεξεργασία των δεδομένων. Σκοπός είναι η δημιουργεία ενός λεξικού με όλες τις μοναδικές λέξεις(“tokenization”). Έπειτα κάθε φράση(“text”) που λαμβάνει το νευρωνικό θα χαρακτηρίζεται από τις λέξεις του λεξικού τις οποίες περιέχει(“Vectorization”). Πριν την χρήση των δεδομένων για την δημιουργία του λεξικού, πρέπει να «καθαστιστούν» τα κείμενα, καθώς περιέχουν χαρακτήρες συμβόλων. Ακόμα είναι προφανώς ότι υπάρχουν πολλές λέξεις όπως οι συνδετικές, δεν προσφέρουν πληροφορία για το κείμενο καθώς χρησιμοποιούνται πολύ συχνά. Αντιστοίχως, υπάρχουν και λέξεις με πολύ μικρή συχνότητα, οι οποίες δεν προσφέρουν πληροφορία λόγω της σπανιότητάς τους. Αυτές οι λέξεις, λοιπόν, είναι αναγκαίο να αφαιρεθούν από το λεξικό. Ο tf-idf vectorizer κωδικοποιεί την κάθε λέξη με βάση την συχνότητα εμφάνισή της. Επομένως η διαδικασία που πραγματοποιείται είναι η εξής. Αρχικά, έγινε εισαγωγή και αποθήκευση του csv αρχείου και ορίστηκε ο vectorizer ο οποίο θα χρησιμοποιηθεί. , οι λέξεις που περιέχονται στα κείμενα τον επιγραφών καθαρίζονται από χαρακτήρες σύμβολα. Έπειτα εντοπίζεται κάθε μοναδική λέξη. Για κάθε μία από αυτές υπολογίζεται η συχνότητα εμφάνισής της σε όλο το dataset. Για την δημιουργία του λεξικού επιλέγονται οι λέξεις με την μεγαλύτερη tf-idf συχνότητα. Από τις συνολικά 24679 λέξεις κρατήθηκαν οι 8000 με την μεγαλύτερη tf-idf συχνότητα. (Αρχικά εφαρμόστηκε περιορισμός, εξαίροντας λέξεις που εμφανίζονται με συχνότητα μεγαλύτερη από 0,9 και λέξεις που εμφανίζονται σε λιγότερο από 2 documents. Το πλήθος των λέξεων αυτών ήταν λίγο μεγαλύτερο από 6000 και από αυτές κρατήθηκαν οι 6000 με την

μεγαλύτερη tf-idf συχνότητα. Όμως με σκοπό την υλοποίηση ενός πιο αποδοτικού νευρωνικού δικτύου κρίθηκε προτιμότερό να συμπεριληφθεί μεγαλύτερο πλήθος λέξεων.)

Με τις 8000 αυτές λέξεις, δημιουργήθηκε το λεξικό (“dicts”).

A1.β.

Το scaling με min-max scaler χρησιμοποιείται με σκοπό να μεταφέρει όλα τα δεδομένα σε κλίμακα από μηδέν μέχρι ένα, αποφεύγοντας έτσι την μεγαλύτερη του νευρωνικού από δεδομένα που βρίσκονται σε ακραίες τιμές.

Σε δεδομένα όπως ο χρόνος που το εύρος των τιμών είναι πολύ μεγάλο με ορισμένες τιμές να απέχουν πολύ μεταξύ τους η κανονικοποίηση είναι αναγκαία.

Για το χρονικό περιθώριο δημιουργήθηκαν δύο κώδικες. Ένας ο οποίος χρησιμοποιεί και τις δύο στήλες min και max με σκοπό τον υπολογισμό του σφάλματος με βάση την απόκλιση είτε από την min τιμή είτε από την max και ένας ο οποίο υπολογίζει την μέση τιμή των min και max και χρησιμοποιεί την απόκλιση από την μέση τιμή για τον υπολογισμό του σφάλματος.

Για την πρώτη περίπτωση έγινε scaling και των δύο στειλών με χρησιμοποιώντας την ελάχιστη και μέγιστη τιμή, των δύο στηλών. Αφού αποθηκεύτηκαν οι δύο στήλες από το αρχικό dataframe, υλοποιήθηκε ένας custom_scaler. Σκοπός ήταν η αντιμετώπιση του προβλήματος, σε έτοιμη συνάρτηση min-max scaling, γινόταν ανά στήλη, επομένως τιμές min και max χάνανε την μεταξύ τους σχέση αφού βρισκόντουσαν σε διαφορετικές στήλες. Η συνάρτηση που υλοποιήθηκε εντοπίζει την ελάχιστη και μέγιστη τιμή και των δύο στηλών και υλοποιέι την διαδικασία του min-max scaling με τον κλασσικό τύπο:

Για την δεύτερη περίπτωση αρχικά έγινε ο υπολογισμός των μέσων τιμών μεταξύ min-max και χρησιμοποιήθηκε έτοιμη συνάρτηση για το scaling των μέσων τιμών.

Το λεξικό που έχουμε απαρτίζεται από λέξεις οι οποίες χαρακτηρίζονται από την tf-idf συχνότητά τους. Αντιλαμβανόμαστε λοιπόν ότι οι λέξεις βρίσκονται ήδη με μία κλίμακα από μηδέν μέχρι ένα με βάση την συχνότητα τους.

Παρ,ότι λοιπόν, για τις λέξεις το min-max scaling δεν κρίνεται απαραίτητο, καθώς φαίνεται να είναι ήδη σε αυτό το εύρος, επιλέχθηκε να εφαρμοστεί min-max scaler και σε αυτές, ακόμα και αν δεν προσφέρει μεγάλη αλλαγή στην αντιμετώπισή τους από το νευρωνικό.

Προτιμήθηκε η χρήση min-max scaler και όχι normalizer καθώς σκοπός δεν ήταν η μεταβολή του σχήματος ή της κατανομής των δεδομένων, αλλά μόνο του εύρους τους.

A1.γ.

Παρότι δεν ζητήθηκε από την εκφώνηση κρίθηκε σκόπιμο να αποσπαστεί ένα τυχαίο κομμάτι των δεδομένων για τον έλεγχο του νευρωνικού στο τέλος. Κρατήθηκε λοιπόν το 20% των δεδομένων με σκοπό του testing του νευρωνικού μετά το πέρας της εκπαίδευσης σε δεδομένα τα οποία δεν έχει ξανά δει. Για τον διαχωρισμό αυτό χρησιμοποιήθηκε η συνάρτηση train_test_split η οποία παίρνει ως είσοδο τα δεδομένα εισόδου και εξόδου, το ποσοστό που θα κρατηθεί για το testing καθώς και έναν ακέραιο ο οποίο χρησιμοποιείται για το «ανακάτεμα» των δεδομένων πριν τον διαχωρισμό τους.

Το υπόλοιπο 80% των δεδομένων διαχωρίζεται σε υοσύνολα με τον παρακάτω τρόπο. Χρησιμοποιείται η συνάρτηση kf.split() της sklearn η οποία παράγει δείκτες για training και validation sets. Με βάση τους δείκτες αυτούς χωρίζονται τα δεδομένα εισόδου x και δεδομένα εξόδου y σε training (X_it_train, y_it_train) και validation(X_it_val, y_it_val) σε κάθε iteration του cross validation.

Η διαδικασία που υλοποιείται είναι η εξής. Σε κάθε επανάληψη τα σύνολα εισόδου και εξόδου χωρίζονται σε 5 επιμέρους υποσύνολα το καθένα. Σε κάθε fold τέσσερα από αυτά χρησιμοποιούνται για την εκπαίδευση και ένα από αυτά χρησιμοποιείται για την επαλήθευση. Η διαδικασία αυτή χρησιμοποιείται για την εξακρίβωση της σωστής γενικοποίησης από το νευρωνικό ανεξαρτήτως από το ποιό υποσύνολο χρησιμοποιείται για validation.

A2.

Για την υλοποίηση του τεχνητού νευρωνικού δικτύου χρησομοποιήθηκε το API του Keras tensorflow. Υλοποιήθηκε συνάρτηση με σκοπό την παραμετροποίηση της συνάρτησης Sequential με της keras με σκοπό τον καθορισμό των ζητούμενων στοιχείων του δικτύου. Όπως το πλήθος των κόμβων και των κρυφών επιπέδων. Ακόμα ορίστηκε το learning rate σε 0.001 όπως ζητείται από την εκφώνηση, καθώς και ο αριθμός των εισόδων σε 8000 και ο αριθμός των εξόδων σε 1 καθώς η επιθυμητή έξοδος είναι μία χρονολογία.

A2.α.

Ζητείται Το τετραγωνικό σφάλμα όπως αναφέρθηκε παραπάνω υλοποιήθηκε και με τους δύο τρόπους που ζητά η εκφώνηση με σκοπό να διακριθεί ποιο έχει την καλύτερη απόδοση. Αναμένεται βέβαια ότι αυτό με την καλύτερη απόδοση και το μικρότερο σφάλμα, θα είναι αυτό που υπολογίζει το σφάλμα ως το RMSE σφάλμα από το κοντινότερο άκρο όταν βρίσκεται εκτός του εύρους min-max και μηδέν όταν βρίσκεται εντός, διότι χρησιμοποιώντας μόνο την μέση τιμή του διαστήματος, γίνεται αντιληπτό ότι «αδικούνται» όλες οι τιμές που βρίσκονται εντός του εύρους min-max αλλά δεν είναι η μέση τιμή.

Και για τι δύο εκδοχές χρειάστηκε η υλοποίηση συνάρτησης για τον υπολογισμό της ρίζας του μέσου τετραγωνικού σφάλματος καθώς αυτό δεν υπήρχε στις βιβλιοθήκες που επιλέχθηκαν. Είναι σημαντικό να σημειωθεί ότι λόγω της χρήσης της βιβλιοθήκης tensorflow στην συνάρτηση οι είσοδοι που δέχεται θα είναι τενσορες για τον λόγο αυτό χρησιμοποιήθηκαν και οι ανάλογες εντολές. Η συνάρτηση υλοποιήθηκε με βάση τον τύπο του rmse:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Όπου y_i: οι πραγματικές τιμές, $\hat{y}_i$: οι προβλεπόμενες τιμές και n: το πλήθος των δειγμάτων

Αρχικά υπολογίζεται το τετράγωνο τις διαφοράς μεταξύ των προβλεπόμενων τιμών από των πραγματικών για κάθε δείγμα(tf.square(y_pred - y_true)), έπειτα υπολογίζει την μέση τιμή των διαφορών αυτών(tf.reduce_mean), αναπαριστώντας το MSE και τέλος υπολογίζει την τετραγωνική ρίζα του μέσου τετραγώνου των διαφορών(tf.sqrt).

Για τον πρώτο τρόπο όμως χρειάστηκε μία custom error συνάρτηση η οποία παίρνει ως είσοδο έναν tensor για την προβλεπόμενη τιμή και ένα tensor με δύο στήλες για την πραγματική τιμή. Καθορίζει την πρώτη στη ως true_min και την δεύτερη ως true_max, έπειτα εάν τη προβλεπόμενη τιμή είναι μεταξύ των true_min και true_max, τότε το σφάλμα είναι μηδέν, αλλιώς εάν είναι μικρότερο του true_min το σφάλμα υπολογίζεται ως η rmse μεταξύ των y_pred και true_min, ενώ εάν είναι μεγαλύτερο από το true_max,ως η rmse μεταξύ y_pred και true_max

Α2.β.

Για την συνάρτηση ενεργοποίησης των κρυφών κόμβων επιλέχθηκε η ReLU. Η ReLU είναι μη γραμμική συνάρτηση άρα μπορεί να μάθει και να αναπαραστήσει πιο πολύπλοκες συναρτήσεις από γραμμικές συναρτήσεις. Η ReLU βοηθά στην αποφυγή του προβλήματος vanishing gradients που συνήθως συμβαίνει σε συναρτήσεις ενεργοποίησης με μικρή πρώτη παράγωγο, όπως η σιγμοειδής συνάρτηση. Πιο συγκεκριμένα, όταν οι κλίσεις (gradients) που αναδίδονται προς τα πίσω κατά τη διαδικασία του backpropagation ναι πολύ μικρές, η ενημέρωση των παραμέτρων γίνεται με πολύ αργό ρυθμό, και το δίκτυο δυσκολεύεται να μάθει αποτελεσματικά. Αυτό οδηγεί σε μια καθυστερημένη ή ακόμα και σε μηδενική εκπαίδευση των πρώτων επιπέδων του δικτύου, καθιστώντας τα αναποτελεσματικά στην εξαγωγή χρήσιμων χαρακτηριστικών από τα δεδομένα εισόδου. Ακόμα ο υπολογισμός της παραγώγου της ReLU είναι εύκολα υπολογίσιμος, παρότι δεν είναι αυστηρά διαφορίσιμη σε όλο το πεδίο ορισμού της.

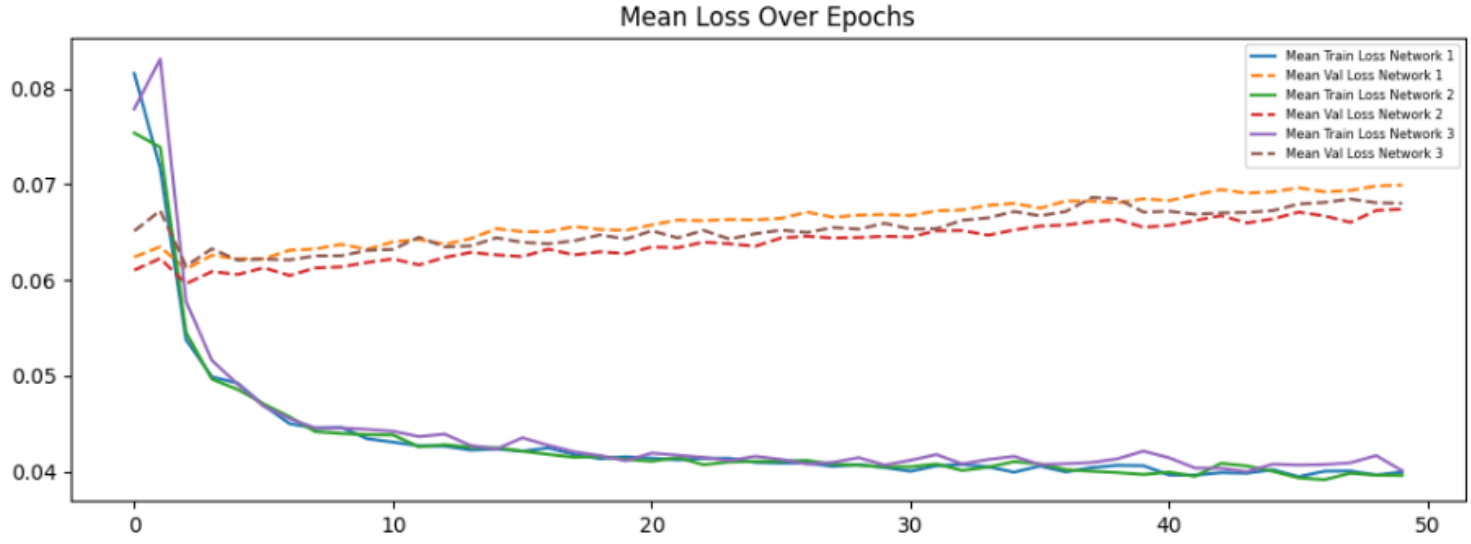
A2.γ.

Για το επίπεδο εξόδου επιλέχθηκε γραμμική συνάρτηση ενεργοποίησης. Η γραμμική συνάρτηση ενεργοποίησης είναι απλή και εύκολη στον υπολογισμό της και στην εκπαίδευση του μοντέλου. Ακόμα, επιτρέπει την έξοδο σε ένα ευρύ φάσμα τιμών, που είναι αναγκαίο λόγω του μεγάλου πλήθους χρονολογιών που θα πρέπει να μπορούν να παραχθούν στην μοναδική έξοδο.

A2.δ.

Υλοποιήθηκε το μοντέλο με ένα κρυφό επίπεδο και διαφορετικό πλήθος κόμβων 125(Network 1), 300 (Network 2) και 600 κόμβους (Network 3). Πρώτη παρατήρηση είναι ότι το validation loss είναι σχεδόν ευθεία σε κάθε περίπτωση, υποδεικνύοντας ότι ανεξαρτήτως ποιο υποσύνολο χρησιμοποιηθεί για το validation στην διαδικασία του cross-validation η γενικοποίηση είναι σωστή. Δεύτερη παρατήσει είναι ότι υπάρχει και μία μικρή άνοδος στην κλήση των validation losses, από αυτό γίνεται κατανοητό ότι οι 50 εποχές ίσως είναι υπερβολικές για συγκεκριμένο πρόβλημα και οδηγούν σε κάποιο overfitting. Παρατηρείται ακόμα, ότι με την πρώτη αύξηση των κόμβων από τους 125 στους 300 το σφάλμα φαίνεται να μειώνεται. Το ίδιο φαίνεται και στο rmse. Με την δεύτερη όμως αύξηση από 300 σε 600 δεν υφίσταται το ίδιο. Συμπεραίνεται, λοιπόν ότι οι κόμβοι που χρησιμοποιούνται στο τρίτο δίκτυο να είναι παραπάνω απότι είναι βέλτιστο, λόγω της απλότητας του προβλήματος.Για την βελτιστοποίηση λοιπόν του μοντέλου θα χρησιμοποιηθούν 30 εποχές και 300 κόμβοι.

Αριθμός νευρώνων στο κρυφό επίπεδο	RMSE Training	RMSE Validation
H1= 125	0.04330320486426353	0.06619656072556973
H1= 300	0.04321637158095837	0.06391891741752624
H1= 600	0.04411830103397371	0.06533068126440049

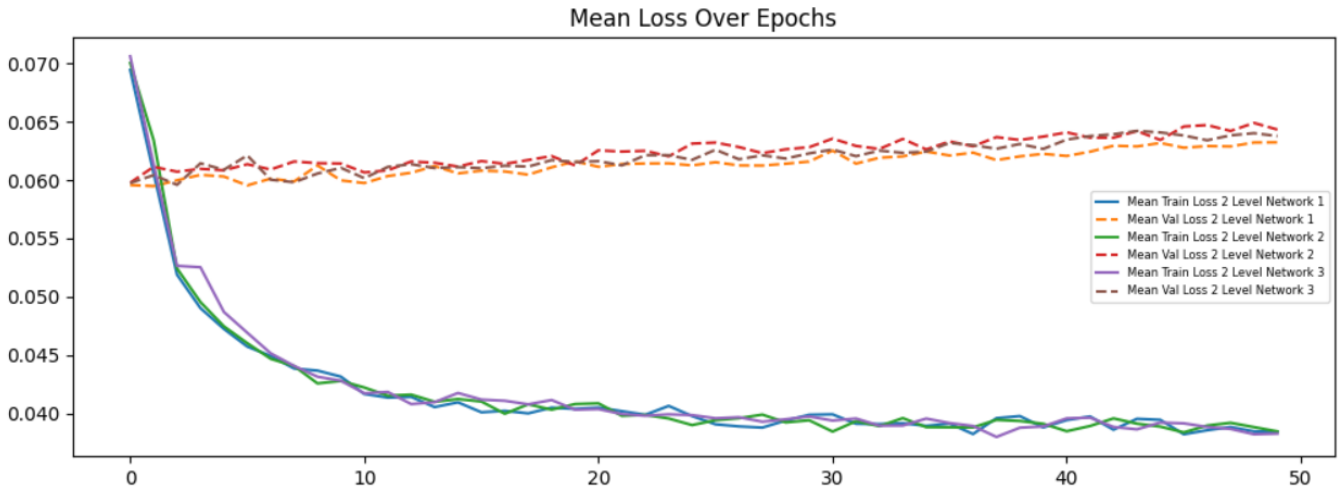


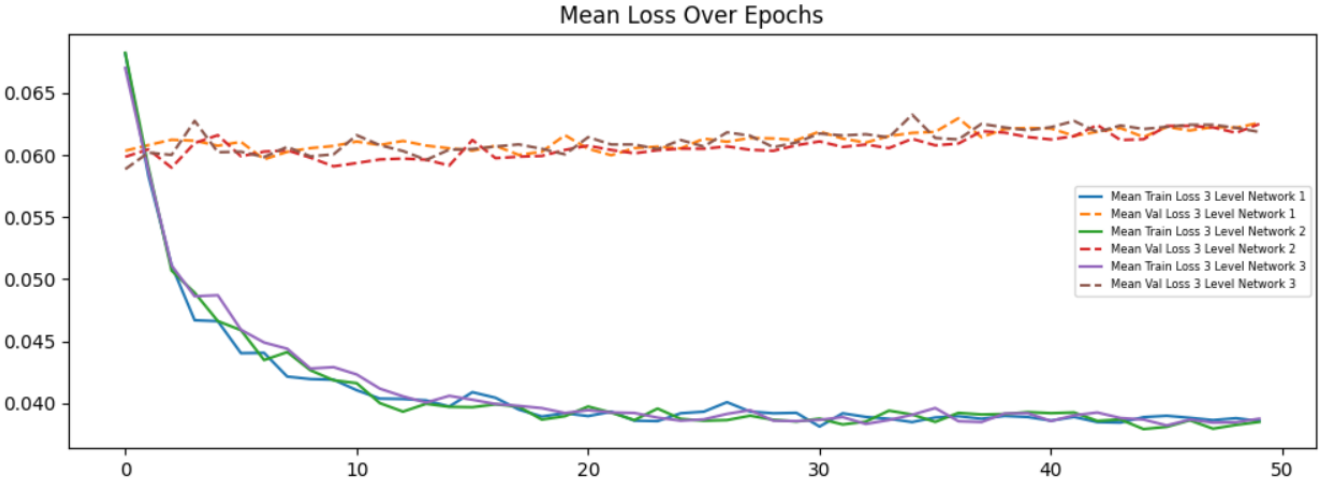
A2.ε.

Για την καλύτερη σύγκριση με τα αποτελέσματα του ενός κρυφού επιπέδου διατηρήθηκε το ίδιο πλήθος εποχών. Με την μεταβολή του πλήθους των κρυφών επιπέδων παρατηρούμε μείωση στο σφάλμα. Με την μείωση του πλήθους των κόμβων σε κάθε επίπεδο παρατηρούμε ότι το σφάλμα είναι μεγαλύτερο από αυτό με σταθερό πλήθος κόμβων. Με την αύξηση του πλήθους των κόμβων ανα επίπεδο το σφάλμα μειώνεται ακόμα παραπάνω. Επομένως κρίνεται προτιμότερο το μεγαλύτερο πλήθος επιπέδων με αυξανόμενο πλήθος κόμβων ανά επίπεδο.

(Να σημειωθεί ότι τα τελευταία δύο παραδείγματα δεν συμπεριλήφθηκαν στο plot)

Επίπεδα	**Πλήθος κόμβων 1ου επιπέδου**	**Πλήθος κόμβων 1ου επιπέδου**	**Πλήθος κόμβων 1ου επιπέδου**	**RMSE Training**	**RMSE Validation**
2	200	50	-	0.04168722939491272	0.0614177779853344
2	400	100	-	0.041785489574074734	0.062493455737829204
2	600	200	-	0.0419220385402441	0.06203466202318668
3	400	100	100	0.04105514201521874	0.06123237179219723
3	600	300	150	0.04106109683215618	0.060715800657868385
3	800	400	200	0.041284039288759226	0.061216611519455906





3	800	800	800	0.0412810878008604	0.06027791255712509
	600	700	800	0.04111090169847012	0.05993711249530316

A2.στ

Υπάρχουν διάφορα κριτήρια τερματισμού που μπορούν να χρησιμοποιηθούν όπως μέγιστος αριθμός εποχών ο οποίος όταν ξεπεραστεί, η διαδικασία της εκπαίδευσης τερματίζεται, ορισμός κατωφλίου σφάλματος και όταν το σφάλμα πέσει κάτω από αυτό τερματισμός της διαδικασίας, ή ακόμα τερματισμός με την σταθεροποίηση του validation loss. Στο early stopping η εκπαίδευση τερματίζεται όταν δεν παρατηρείται βελτίωση στο σφάλμα για έναν καθορισμένο αριθμό εποχών ή όταν η απόσταση μεταξύ των σφαλμάτων σε δύο συνεχόμενες εποχές είναι πολύ μικρή. Αυτός είναι ένας τρόπος να αποφευχθεί το overfitting και να βελτιωθεί η γενίκευσς του μοντέλου. Αναλυτικότερα, κατά τη διάρκεια της εκπαίδευσης του μοντέλου, το σύστημα παρακολουθεί την απόδοση του μοντέλου σε ένα σύνολο επικύρωσης (validation set). Αν η απόδοση στο σύνολο επικύρωσης σταματήσει να βελτιώνεται ή αρχίσει να χειροτερεύει για μια συνεχόμενη σειρά εποχών, τότε η διαδικασία εκπαίδευσης τερματίζεται.

A3.

Στο ερώτημα αυτό ζητείται να μεταβάλουμε τις τιμές του ρυθμού εκπαίδευσης και της σταθεράς ορμής. Χρησιμοποιήθηκαν 30 εποχές 3 επίπεδα με κόμβους [600,700,800]

Ο ρυθμός εκπαίδευσης καθορίζει την αλλαγή των παραμέτρων του μοντέλου κατά την διάρκεια της εκπαίδευσης. Πόσο δηλαδή μεταβάλλονται οι παράμεροι με σκοπό να ελαχιστοποιήσουν την συνάρτηση κόστους. Με πολύ αργό ρυθμό εκπαίδευσης η σύγκλιση του μοντέλου γίνεται πολύ αργή και υπάρχει η πιθανότητα εγκλωβισμού σε τοπικά ελάχιστα. Αντιθέτως με πολύ μεγάλο ρυθμό εκπαίδευσης μπορεί να οδηγήσει σε πολύ μεγάλες ανεπιθύμητες αλλαγές, που οδηγούν σε αστάθεια και δυσκολία σύγκλισης σε καλή λύση.

Η σταθερά ορμής καθορίζει το ποσοστό στο οποίο λαμβάνεται υπόψη η προηγούμενη αλλαγή των παραμέτρων κατά την ενημέρωσή τους. Όσο μεγαλύτερη, τόσο μεγαλύτερη η επιρροή των προηγούμενων ενημερώσεων, στην τωρινή. Ενώ η υψηλή σταθερά ορμής μπορεί να βοηθήσει στην προσπέραση τοπικών ελαχίστων και γρήγορη σύγκλιση σε καλύτερες λύσεις, η πολύ υψηλή σταθερά ορμής μπορεί να προκαλέσει διαταραχές με την πολύ γρήγορη αλλαγή των παραμέτρων.

Η σταθερά ορμής πρέπει να είναι κάτω της μονάδας. Αν είναι ίση ή μεγαλύτερη της μονάδας οι προηγούμενες ενημερώσεις των βαρών θα είναι πιο σημαντικές από τις τωρινές. Όσο αυξάνουμε τον ρυθμό μάθησης φαίνεται να εγκλωβίζεται το δίκτυο σε τοπικά ελάχιστα.

η	**m**	**RMSE**
0.001	**0.2**	0.06776
0.001	**0.6**	0.06486
0.05	**0.6**	0.16321
0.1	**0.6**	0.48633