Εργασία Υπολογιστική Νοημοσύνη Μέρος Β

Γρηγόρης Καπαδούκας (ΑΜ: 1072484)

19 Μαΐου 2023

0 Περιβάλλον Εργασίας - Σύνδεσμος GitHub με Κώδικα

Για την διεκπεραίωση αυτής της εργασίας έχω επιλέξει να χρησιμοποιήσω γλώσσα προγραμματισμού Python μαζί τις βιβλιοθήκες PyGAD για την υλοποίηση του γενετικού αλγορίθμου και TensorFlow (κυρίως το API της, το Keras) για τον σχεδιασμό και την εκπαίδευση του νευρωνικού δικτύου από το μέρος Α. Επίσης χρησιμοποιώ Pandas, Numpy και Scikit-Learn με σκοπό τον χειρισμό του CSV αρχείου και της προεπεξεργασίας.

Ο κώδικας που γράφτηκε για την εργασία βρίσκεται στο repository στον παρακάτω σύνδεσμο:

https://github.com/GregKapadoukas/University-Computational-Intelligence-Project-B

Στο repository αυτό συμπεριλαμβάνω και το αρχείο 'environment.yml' το οποίο μπορεί να χρησιμοποιηθεί με χρήση του εργαλείου conda για να δημιουργηθεί πανομοιότυπο Python virtual environment με αυτό που χρησιμοποίησα για τη συγγραφή και εκτέλεση του κώδικα. Βέβαια για χρήση της βιβλιοθήκης TensorFlow απαιτούνται επιπλέον configuration βήματα, που αναλύονται στην εξής σελίδα: https://www.tensorflow.org/install/pip

1 Σχεδιασμός ΓΑ

1.a Κωδικοποίηση

Παρατηρώντας τα δεδομένα του dataset βλέπουμε ότι για οι αισθητήρες είναι συνολικά 4, και τα δεδομένα για κάθε αισθητήρα είναι 3 σε κάθε περίπτωση, δίνοντας συνολικά 12 πραγματικές τιμές στο εύρος [-702, 533] (υποθέτοντας πως οι αισθητήρες παράγουν τιμές στο ίδιο εύρος και πως έχουν παρατηρηθεί τα άκρα των πιθανών τιμών στις μετρήσεις του dataset. Στον κώδικα χρησιμοποιώντας τη συνάρτηση .describe() του Pandas εμφανίζουμε τη μέγιστη και ελάχιστη τιμή για κάθε αισθητήρα που παρατηρήθηκε στο dataset).

Λόγω του συνδυασμού του μεγάλου πλήθους τιμών που χρησιμοποιούνται για την αναπαράσταση της κλάσης κίνησης και του μεγάλου εύρους στο οποίο ανήκουν οι τιμές αυτές, αποφάσισα να χρησιμοποιήσω real-valued encoding, σε συνδυασμό με MinMax κωδικοποίηση στο εύρος [0,1].

Άρα η λύση που προσεγγίζουμε καθώς και κάθε χρωμόσωμα, θα αποτελούνται από ένα πίνακα πραγματικών αριθμών στο εύρος [0,1] μεγέθους 12. Εννοείται πως όταν βρεθεί η βέλτιστη λύση θα κάνουμε inverse MinMax scaling ώστε να επαναφερθούν οι τιμές στο εύρος [-702, 533].

2 Προεπεξεργασία και Προετοιμασία Δεδομένων

2.a Κωδικοποίηση και προεπεξεργασία δεδομένων

2.a.1 Διάβασμα του CSV αρχείου και μετατροπή κατηγορικών δεδομένων σε αριθμητικά

Αρχικά, με σκοπό το διάβασμα του dataset στη μορφή του CSV αρχείου χρησιμοποιώ εντολές της βιβλιοθήκης Pandas για φορτώσω τα δεδομένα στη μορφή ενός DataFrame. Έπειτα χωρίζω το αρχικό DataFrame σε δύο, από τα οποία το πρώτο περιέχει τα δεδομένα των αισθητήρων και τα στοιχεία του ατόμου πάνω στο οποίο έγιναν οι μετρήσεις και το δεύτερο περιέχει την κλάση δραστηριότητας στην οποία ανήκει το άτομο.

Έπειτα μετατρέπω τα κατηγορικά δεδομένα των κλάσεων του δεύτερου DataFrame σε αριθμητικά δεδομένα, τα οποία όμως είναι one-hot encoded διανύσματα μεγέθους $\mathbb{R}^{1\times5}$. Άρα οι τιμές μετατρέπονται σε διανύσματα με την εξής αντιστοίχηση:

- ’sitting’: [1 0 0 0 0]
- ’sitting-down’: [0 1 0 0 0]
- ’standing’: [0 0 1 0 0]
- ’standing-up’: [0 0 0 1 0]
- ’walking’: [0 0 0 0 1]

Επέλεξα την παραπάνω one-hot encoded προσέγγιση αντί για την αριθμητική 1-5 όπως προτείνεται στην εκφώνηση, επειδή σκοπεύω να έχω 5 εξόδους στο νευρωνικό δίκτυο και να χρησιμοποιήσω μετρική σφάλματος Categorical Cross-Entropy, όπως θα εξηγήσω στο ερώτημα 3.

Έπειτα μετατρέπω και τα κατηγορικά δεδομένα του δεύτερου DataFrame, δηλαδή τα series ’Name’ και ’Gender’ σε αριθμητικά δεδομένα, με βάση την εξής αντιστοίχηση.

Για τα ονόματα:

- ’debora’: 1
- ’katia’: 2
- ’wallace’: 3
- ’jose_carlos’: 4

Για το φύλλο:

- ’Man’: 1
- ’Woman’: 2

Έτσι πλέον έχω μετατρέψει όλα τα κατηγορικά δεδομένα σε αριθμητικά, το οποίο είναι αναγκαίο για να μπορέσει να γίνει η μετέπειτα προεπεξεργασία των δεδομένων και η χρήση τους για την εκπαίδευση του νευρωνικού δικτύου.

2.a.2 Εξάλειψη Ενδεχόμενης Πόλωσης

Από τις μεθόδους που αναφέρονται στην εκφώνηση έχουμε δύο επιλογές διαδικασιών με σκοπό την εξάλειψη ενδεχόμενης πόλωσης.

Η πρώτη επιλογή είναι να κάνουμε αρχικά κεντράρισμα και έπειτα κανονικοποίηση των δεδομένων. Με αυτή τη διαδικασία καταλήγουμε με δεδομένα στο εύρος που επιλέγουμε για την κανονικοποίηση και ταυτόχρονα τα δεδομένα σε κάθε Series έχουν μέση τιμή το μέσο του εύρους που επιλέξαμε.

Η δεύτερη επιλογή είναι να κάνουμε τυποποίηση στα δεδομένα που έχει αποτέλεσμα τη μεταφορά των δεδομένων για κάθε Series σε γκαουσιανή κατανομή με μέση τιμή 0 και διακύμανση 1. Σε αυτή τη περίπτωση δεν υπάρχει ανάγκη για κεντράρισμα ή τυποποίηση εφόσον τα δεδομένα κεντράρονται μέσω της διαδικασίας (μέση τιμή 0), και η κανονικοποίηση θα επηρέαζε τη διακύμανση, το οποίο δεν είναι επιθυμητό χαρακτηριστικό σε αυτή τη περίπτωση.

Σχετικά με τις δύο προσεγγίσεις σημειώνω ότι η τυποποίηση φέρει καλύτερα αποτελέσματα σε περιπτώσεις όπου τα δεδομένα ακολουθούν ήδη γκαουσιανή κατανομή, με διαφορετικές βέβαια μέσες τιμές και διακυμάνσεις ή σε περιπτώσεις που έχουμε outliers. Στην περίπτωση που η κατανομή δεν είναι γκαουσιανή και δεν έχουμε outliers, υπάρχει πιθανότητα το κεντράρισμα με μετέπειτα κανονικοποίηση να φέρει καλύτερο αποτέλεσμα.

Σε αυτή τη περίπτωση θέλουμε να επεξεργαστούμε τα δεδομένα του DataFrame που περιέχει τις μετρήσεις των αισθητήρων και τα στοιχεία των ατόμων, και όχι το DataFrame με τις κλάσεις δραστηριοτήτων, εφόσον αυτό έχει ήδη μετατραπεί στη μορφή που χρειάζεται μέσω του one-hot encoding. Άρα παρατηρούμε (διαισθητικά και μέσω της συνάρτησης plot.density() του Pandas) ότι τα δεδομένα σε όλες τις κλάσεις ακολουθούν κανονική κατανομή εκτός από τις κλάσεις 'User', 'Gender' και 'Age'.

Οπότε εφόσον έχουμε συνδυασμό κανονικών και διαφορετικών κατανομών στα δεδομένα, αποφασίζω να δοκιμάσω και τις δύο διαδικασίες στο νευρωνικό δίκτυο του ερωτήματος 4 (με $\eta = 0.001$ και $m = 0.6$) και παρατηρώ ότι η μετρική της ακρίβειας στη περίπτωση της τυποποίησης μόνο είναι καλύτερη (0.9851 vs 0.8529) για τη σύγκριση των μέσων όρων των τιμών που προέκυψαν από το evaluation με το evaluation σετ σε κάθε fold, σε σύγκριση με τον συνδυασμό κεντραρίσματος και κανονικοποίησης (η κανονικοποίηση έγινε σε εύρος [-1,1], επειδή χρησιμοποιήθηκε ReLU συνάρτηση ενεργοποίησης, οπότε το εύρος [0,1] ήταν λιγότερο αποδοτικό).

Άρα συμπεραίνω πως επειδή τα δεδομένα ακολουθούν επί το πλείστον κανονικές κατανομές (15/18 Series), η συνολική ακρίβεια του μοντέλου με τυποποίηση είναι υψηλότερη από ότι με κεντράρισμα και κανονικοποίηση. Μάλιστα τα δεδομένα των μη κανονικών κατανομών (3/18 Series) επιδέχονται επίσης κανονικοποίηση ως αποτέλεσμα της τυποποίησης, οπότε δεν θεώρησα αναγκαίο να χρησιμοποιήσω διαφορετική τεχνική προεπεξεργασίας μόνο σε αυτά.

2.b Διασταυρούμενη Επικύρωση (cross-validation)

Αρχικά πριν τον διαχωρισμό των δεδομένων για το 5-fold CV, θα ενώσω τα δύο DataFrame που διαχώρισα πριν για να κάνω την προεπεξεργασία, με χρήση της εντολής concat του Pandas. Άρα πλέον έχουμε πάλι ένα DataFrame με όλα τα δεδομένα, αυτή τη φορά όμως προεπεξεργασμένα.

Με σκοπό τον διαχωρισμό των δεδομένων στα 5 σύνολα και τη μετέπειτα χρήση 5-fold CV, θα χρησιμοποιήσω ένα object τύπου KFold της βιβλιοθήκης SKLearn. Έτσι θα αρχικοποιήσω το object χρησιμοποιώντας παραμέτρους 'nsplits = 5' και 'shuffle = true' με 'random_state = 2'. Με την παράμετρο nsplits ορίζω ότι θέλω να χωρίσω τα δεδομένα μου σε 5 μέρη, με την παράμετρο shuffle ορίζω ότι θέλω το κάθε fold να είναι ισορροπημένο ως προς τον αριθμό των δειγμάτων και με το random_state ορίζω ένα seed που χρησιμοποιείται για την τυχαιότητα για την τελική σειρά των χωρισμένων συνόλων (ορίζει έμμεσα την τυχαιότητα ως προς ποιο σύνολο επιλέγεται ως σύνολο εκπαίδευσης).

Έπειτα για να διαχωρίσω τα δεδομένα με βάση το KFold που χρησιμοποιήθηκε, χρησιμοποιώ τον βρόχο "for result in kf.split(preprocessed_df):", ο οποίος αποθηκεύει στον πίνακα result σε κάθε iteration στη πρώτη θέση του τα 4 από τα 5 τμήματα, που αποτελούν το train set και στη δεύτερη θέση το τελευταίο τμήμα που αποτελεί το validation set. Οπότε το περιεχόμενο της for θα εκτελειστεί 5 φορές, μια φορά για κάθε fold. Τέλος αποθηκεύω στο περιεχόμενο της for αντίστοιχα τις μεταβλητές train_set, train_classes, validation_set, validation_classes, που περιέχουν τα δεδομένα του test και validation set κάθε φορά, πάλι διαχωρισμένα στα δεδομένα ατόμων / αισθητήρων και στις κλάσεις.

Τέλος σημειώνω ότι σε αυτή τη περίπτωση δεν έχω φτιάξει ξεχωριστό test set, για να ελέγξω την τελική απόδοση του μοντέλου σε καινούργια δεδομένα, αλλά χρησιμοποιώ όλα τα δεδομένα για το 5-fold CV και στο τέλος υπολογίζω τον μέσο όρο των μετρικών που προκύπτουν από το evaluation με το validation set για όλα τα folds. Με αυτό το τρόπο μπορώ πάλι να καταλήξω σε συμπεράσματα ως προς την αναμενόμενη συμπεριφορά του μοντέλου σε νέα δεδομένα και τη δυνατότητά του για γενίκευση.

3 Επιλογή Αρχιτεκτονικής

Αρχικά με σκοπό την δημιουργία του ΤΝΔ χρησιμοποιώ το sequential API του Keras, πιο συγκεκριμένα την εντολή ”model = keras.Sequential([...])”. Έτσι με αντίστοιχες εισόδους στον πίνακα ορίζω ότι το επίπεδο εισόδου περιέχει 18 νευρώνες (όσες οι συνολικές στήλες των δεδομένων αφού αφαιρεθούν οι τελικές κλάσεις δραστηριοτήτων. Έπειτα ορίζω το ένα κρυφό επίπεδο, με τον αριθμό νευρώνων που περιέχει σε κάθε δοκιμή, μαζί με την συνάρτηση ενεργοποίησης. Τέλος ορίζω τους νευρώνες στο επίπεδο εξόδου, οι οποίοι είναι 5, λόγω της επιλογής να χρησιμοποιήσουμε one-hot encoding στις κλάσεις δραστηριοτήτων. Η συνάρτηση ενεργοποίησης στους νευρώνες εξόδου ορίζεται επίσης εδώ.

Επίσης στην εντολή ”model.compile(...)” ορίζεται ο optimizer, η συνάρτηση σφάλματος, οι μετρικές αξιολόγησης και ο ρυθμός μάθησης που χρησιμοποιείται κάθε φορά.

3.a

3.a.1 Ανάλυση Μετρικών

Cross-Entropy (CE):

Το cross-entropy σαν μετρική σφάλματος θεωρείται καλό για την επίλυση προβλημάτων κατηγοριοποίησης. Σαν διαδικασία υπολογίζει μια προσέγγιση του διανύσματος της κατανομής πυκνότητας πιθανότητας για να ανήκει η είσοδος σε καθεμία από τις διαθέσιμες κλάσεις δραστηριοτήτων. Έπειτα μέσω της μάθησης οι προσεγγίσεις της πυκνότητας πιθανότητας βελτιώνονται ως έμμεσο αποτέλεσμα του back-propagation, με σκοπό την ελαχιστοποίηση της εντροπίας, δηλαδή της μέσης αβεβαιότητας για τις τιμές του διανύσματος πιθανοτήτων. Με αυτόν τον τρόπο μέσω της μάθησης, το ΤΝΔ γίνεται όλο και πιο ”βέβαιο” για τα αποτελέσματα που παράγει.

Μέσο Τετραγωνικό Σφάλμα (MSE):

Το μέσο τετραγωνικό σφάλμα σαν μετρική σφάλματος θεωρείται καλό για την επίλυση προβλημάτων παλινδρόμησης. Σαν διαδικασία υπολογίζει τη μέση διαφορά μεταξύ του γνωστού σωστού αποτελέσματος και της προσέγγισης στην έξοδο του νευρωνικού δικτύου, τετραγωνισμένη. Οπότε μέσω της ελαχιστοποίησης του μέσου τετραγωνικού σφάλματος το ΤΝΔ φτάνει όλο και πιο κοντά στις προσεγγίσεις του στο σωστό αποτέλεσμα. Το μέσο τετραγωνικό σφάλμα σαν μετρική σφάλματος έχει επίσης την ιδιότητα ότι οδηγεί τη διαδικασία της μάθησης σε γρηγορότερη σύγκλιση σε σχέση με μετρικές όπως το μέσο απόλυτο σφάλμα, γιατί όταν μεγαλώνει η διαφορά μεταξύ του σωστού αποτελέσματος και της προσέγγισης, το σφάλμα αυξάνεται εκθετικά, οδηγώντας το ΤΝΔ σε πιο δραστικές αλλαγές στα βάρη και στο κατώφλι, μέσω της διαδικασίας του back-propagation.

Ακρίβεια Ταξινόμησης (Accuracy):

Η ακρίβεια ταξινόμησης σαν μετρική χρησιμοποιείται συνήθως για την αξιολόγηση μοντέλων, επειδή προσφέρει μια ευνόητη μετρική της απόδοσης ενός ΤΝΔ. Δεν χρησιμοποιείται σαν μετρική σφάλματος στη διαδικασία του back-propagation, επειδή δεν είναι διαφορίσιμη και επίσης δεν προσφέρει πληροφορία σχετικά με το πόσο μακριά ήταν η προσεγγισμένο αποτέλεσμα από το σωστό, για να γίνει αντίστοιχα μεγάλη αλλαγή στα βάρη και στο κατώφλι.

3.a.2 Επιλογή προτιμότερης μετρικής σφάλματος για το ΤΝΔ της άσκησης

Το ΤΝΔ της άσκησης έχει σκοπό την επίλυση προβλήματος κατηγοριοποίησης πολλαπλών κλάσεων, οπότε από την προηγούμενη ανάλυση που έχουμε κάνει καταλήγουμε ότι η προτιμότερη μετρική είναι το categorical cross-entropy. Επίσης ο λόγος που δεν χρησιμοποιούμε sparse categorical cross-entropy είναι επειδή έχουμε αποφασίσει να έχουμε one-hot encoded έξοδο για τις κατηγορίες και όχι ακέραιες τιμές, οπότε η κλασσική μέθοδος Categorical CE είναι η επιθυμητή στην περίπτωση αυτή.

3.b

Εφόσον έχουμε επιλέξει παραπάνω να κάνουμε αναπαράσταση των κλάσεων δραστηριοτήτων που ανήκουν τα δεδομένα με one-hot encoding, και ο συνολικός αριθμός κλάσεων είναι 5, θα χρειαστούμε 5 νευρώνες στο επίπεδο εξόδου. Έτσι κάθε νευρώνας εξόδου του νευρωνικού δικτύου αντιπροσωπεύει μια τιμή του one-hot encoded διανύσματος αποτελέσματος που δηλώνει την κλάση που θεώρησε το νευρωνικό δίκτυο ότι ανήκουν τα δεδομένα εισόδου.

Εδώ σημειώνω επίσης ότι θα μπορούσαν οι κλάσεις να είχαν αναπαρασταθεί με τιμές integer και να χρησιμοποιηθεί τη μετρική σφάλματος sparse categorical cross-entropy, αλλά θα χρειαζόταν να έχω 6 νευρώνες εξόδου αντί για 5, καθώς και ο χρόνος σύγκλισης του κλασσικού binary cross-entropy είναι καλύτερος.

3.c

Η συνάρτηση ενεργοποίησης που επέλεξα για τους κρυφούς κόμβους είναι η ReLU. Οι λόγοι που την επέλεξα είναι οι εξής:

- Η συνάρτηση ReLU είναι απλή και γρήγορη να υπολογιστεί από έναν υπολογιστή.
- Η ReLU έχει την ιδιότητα να δημιουργεί αραιά νευρωνικά δίκτυα, δηλαδή νευρωνικά όπου οι νευρώνες που δεν επηρεάζουν την έξοδο έχουν μηδενικές τιμές στα βάρη και στα κατώφλια. Αυτή η ιδιότητα προκύπτει λόγω της μηδενικής τιμής y της ReLU για τιμές x ≤0. Αυτό είναι σε αντίθεση με άλλες συναρτήσεις ενεργοποίησης, όπως η σιγμοειδή, όπου οι νευρώνες που δεν επηρεάζουν την έξοδο έχουν μικρές μη μηδενικές τιμές στα βάρη και στα κατώφλια, οδηγώντας σε ανάγκη για περισσότερους υπολογισμούς και καθυστερώντας τη διαδικασία της μάθησης.
- Η ReLU βοηθάει στην αντιμετώπιση του προβλήματος του vanishing gradient, το οποίο συμβαίνει όταν η κλίση γίνεται πολύ μικρή ή μηδενίζεται καθώς γίνεται το back-propagation στους νευρώνες. Αυτό οδηγεί στην ανάγκη για πολλές ανανεώσεις των βαρών ή το ”κόλλημα” των βαρών, και καθυστερεί ή σταματά την διαδικασία της μάθησης. Η συνάρτηση ReLU βοηθάει στην εξάλειψη του προβλήματος αυτού επειδή έχει κλίση ίση με ένα και μηδενίζεται μόνο όταν η τιμή x είναι μικρότερη του μηδενός.

3.d

Η συνάρτηση που επέλεξα για το επίπεδο εξόδου είναι η Softmax. Ο λόγος για αυτό είναι επειδή η Softmax οδηγεί την κάθε έξοδο στη πιθανότητα που ορίζει το ΤΝΔ η είσοδος να ανήκει στην κλάση αυτή (άρα έχει και εύρος 0 έως 1). Έτσι το σύνολο των 5 νευρώνων εξόδου αναπαριστούν την συνάρτηση πυκνότητας πιθανότητας για όλες τις πιθανές εξόδους, και μετά τη διαδικασία της μάθησης, εφόσον η αβεβαιότητα ελαχιστοποιείται (back-propagation με Categorical CE), το αποτέλεσμα στους νευρώνες εξόδου είναι η one-hot encoded αναπαράσταση της κατηγορίας στην οποία ανήκει η έξοδος.

Η σιγμοειδή σε αντίθεση για παράδειγμα, επειδή αναφέρεται στην εκφώνηση, έχει εύρος από -1 έως 1, οπότε από εκεί και μόνο δεν πληρεί τα χαρακτηριστικά που επιθυμώ για το ΤΝΔ της άσκησης, έτσι όπως το έχω σχεδιάσει.

3.e

Σημειώσεις:

- Επειδή στην εκφώνηση ζητείται να διατυπώσουμε συμπεράσματα ως προς τη συνάρτηση κόστους, έχω κάνει τις μετρήσεις δύο φορές, μια φορά με συνάρτηση κόστους Categorical CE και μια φορά για συνάρτηση κόστους MSE. Σημειώνω εδώ ότι η μετρική Accuracy δεν μπορεί να χρησιμοποιηθεί ως συνάρτηση κόστους, επειδή δεν είναι διαφορίσιμη, όπως αναφέρθηκε και παραπάνω στο ερώτημα 3.a.1. Για αυτό το λόγο χρησιμοποιείται μόνο ως μετρική αξιολόγησης. Παρόλα αυτά και στις δύο περιπτώσεις συναρτήσεων σφάλματος δείχνω όλες τις μετρικές σφάλματος στους πίνακες, με σκοπό να παρουσιαστούν ως μετρικές απόδοσης.
- Στην εκφώνηση ζητείται να συμπεριλάβουμε τις γραφικές παραστάσεις σύγκλισης (Μ.Ο.) ανά κύκλο εκπαίδευσης. Επειδή όμως για 5 folds, δύο συναρτήσεις σφάλματος, τρεις διαφορετικούς αριθμούς νευρώνων ανά κρυφό επίπεδο και τρεις μετρικές αξιολόγησης, οι γραφικές είναι πολλές, αποφάσισα να συνδυάσω τα γραφήματα ώστε να παρουσιάζονται τα αποτελέσματα για εκπαίδευση με τις δύο διαφορετικές συναρτήσεις σφάλματος και τα αποτελέσματα για κάθε fold σε ένα γράφημα. Άρα έχω εννιά διαφορετικά γραφήματα, τρία για κάθε αριθμό νευρώνων στο κάθε κρυφό επίπεδο, από τα οποία το κάθε ένα δείχνει μια από τις τρεις μετρικές που ζητείται, και όλα τα δεδομένα που ζητούνται φαίνονται πάνω σε αυτά. Με αυτόν τον τρόπο μπορώ να συγκρίνω ευκολότερα την απόδοση του Categorical CE σε σύγκριση με το MSE ως συνάρτηση σφάλματος, σε σχέση με κάθε μετρική αξιολόγησης, σε κάθε fold.
- Έχω επιλέξει να μην δεσμεύσω ξεχωριστά δεδομένα για να δημιουργήσω test set, αλλά ακολουθώ τον ορισμό του k-fold Cross Validation και για να κρίνω την συνολική απόδοση του μοντέλου στο τέλος χρησιμοποιώ το μέσο όρο των k validation στα validation sets κάθε φορά, όπως αναφέρθηκε και παραπάνω. Οπότε τα στοιχεία που εισάγω στους πίνακες είναι ο μέσος όρος των μετρικών σφαλμάτων και accuracy που προέκυψαν από το evaluation με το validation set κάθε φορά για κάθε fold.
- Όλες οι παρακάτω μετρήσεις έχουν γίνει για αριθμό εποχών ίσο με 10 για κάθε fold, χωρίς early stopping αφού αυτό εισάγεται στο επόμενο ερώτημα.

3.e.1 Πίνακες μετρικών αξιολόγησης

Παρακάτω φαίνεται ο πίνακας με τις μετρικές αξιολόγησης που προέκυψαν από την διαδικασία μάθησης του νευρωνικού με συνάρτηση σφάλματος Categorical CE. Όπως ανέφερα και παραπάνω, στον πίνακα δείχνω μόνο τον μέσο όρο των τιμών των μετρικών που προέκυψαν από το evaluation με το validation set σε κάθε fold.

Αριθμός νευρώνων κρυφό επίπεδο	CE Loss	MSE	Acc
5	0.2471	0.0234	0.9216
5 + 18 / 2 ≈ 12	0.1181	0.0102	0.9677
5 + 18 = 23	0.0728	0.0063	0.9802

Σχήμα 1: Πίνακας μετρικών αξιολόγησης για συνάρτηση σφάλματος Categorical CE

Επίσης δείχνω τον αντίστοιχο πίνακα μετρικών αξιολόγησης για νευρωνικό δίκτυο που εκπαιδεύτηκε με συνάρτηση σφάλματος MSE. Στον πίνακα δείχνω πάλι μόνο τις τιμές των μετρικών που προέκυψαν μετά από το evaluation με το validation set.

Αριθμός νευρώνων κρυφό επίπεδο	CE Loss	MSE	Acc
5	0.3960	0.0274	0.9321
5 + 18 / 2 ≈ 12	0.2966	0.0121	0.9680
5 + 18 = 23	0.1428	0.0074	0.9821

Σχήμα 2: Πίνακας μετρικών αξιολόγησης για συνάρτηση σφάλματος MSE

3.e.2 Γραφικές παραστάσεις σύγκλισης

Παρακάτω φαίνονται οι γραφικές παραστάσεις σύγκλισης για χρήση Categorical CE συνάρτηση σφάλματος σε σύγκριση με MSE συνάρτηση σφάλματος για τις διαφορετικές μετρικές που ζητούνται, για τα διαφορετικά ποσά νευρώνων κρυφού επιπέδου που ζητούνται και για όλα τα folds:

Με 5 νευρώνες στο κρυφό επίπεδο:

Παρακάτω δίνονται και γραφικές παραστάσεις σύγκλισης για τις ίδιες μετρικές και περιπτώσεις όπως πάνω, με τη διαφορά ότι δεν εμφανίζουμε κάθε fold αναλυτικά, παρά υπολογίζω τους μέσους όρους των μετρικών για το training όλων των fold. Αυτό το κάνω για να είναι ευκρινές ποιο μοντέλο είναι πιο αποδοτικό σε κάθε περίπτωση.

Με 5 νευρώνες στο κρυφό επίπεδο:

3.e.3 Συμπεράσματα

1. Σχετικά με τον αριθμό των κρυφών κόμβων παρατηρώ πως ανεξάρτητα από τη συνάρτηση κόστους που χρησιμοποιείται στην εκπαίδευση, ο μεγαλύτερος αριθμός κόμβων οδηγεί σε μικρότερες τιμές των μετρικών σφαλμάτων και μεγαλύτερη τιμή της μετρικής Accuracy. Ά-ρα με περισσότερους κρυφούς κόμβους η απόδοση του ΤΝΔ είναι καλύτερη.

2. Σχετικά με τη συνάρτηση κόστους παρατηρώ ότι η εκπαίδευση μέσω Categorical CE φέρει καλύτερα αποτελέσματα για τις μετρικές Categorical Cross-Entropy error και MSE όπως ήταν αναμενόμενο για τους λόγους που αναφέρθηκαν στο κεφάλαιο 3.a.1.

   Παρόλα αυτά για πρόσεξα ότι τα μοντέλα που εκπαιδεύτηκαν με MSE συνάρτηση κόστους κατά μέσο όρο είχαν οριακά καλύτερο Accuracy από τα μοντέλα που εκπαιδεύτηκαν με Categorical CE. Οπότε ενώ το σφάλμα Categorical CE και MSE ήταν μεγαλύτερο σε εκπαίδευση με MSE, το Accuracy ήταν οριακά καλύτερο, που με οδηγεί στο να πιστέψω ότι μερικές φορές το αυξημένο σφάλμα σχετικά με το Categorical CE και το MSE οδήγησε τις τελικές τιμές σε πιο σωστή κατηγοριοποίηση, εφόσον χρησιμοποιείται μετρική Categorical Accuracy (σύμφωνα με τον ορισμό του Accuracy που δόθηκε στην εκφώνηση).

   Αυτό το συμβάν πιθανότατα να είναι τυχαίο (ειδικά αν λάβουμε υπόψη πόσο μικρή ήταν η διαφορά) και να προκύπτει μόνο στα συγκεκριμένα δεδομένα. Πιθανότατα επίσης, αν είχα μεγαλύτερο dataset, να μην παρουσιαζόταν το φαινόμενο αυτό τόσο έντονα.

   Οπότε θα συνεχίσω να προτιμάω την εκπαίδευση με Categorical CE συνάρτηση κόστους, επειδή είχε τις μικρότερες μετρικής σφάλματος και θεωρείται γενικότερα καλύτερη για προβλήματα multiclass classification.

3. Σχετικά με την ταχύτητα σύγκλισης παρατηρώ πως όσο μεγαλώνει ο αριθμός των εποχών τόσο μικρότερη γίνεται η κλήση μείωσης του σφάλματος και αύξησης του Accuracy. Αυτή η συμπεριφορά προκύπτει από τον αλγόριθμο Adam που χρησιμοποιούμε για την προσέγγιση των βέλτιστων βαρών, δηλαδή αυτά που επιτελούν το ελάχιστο κόστος (και με έμμεσο τρόπο τη μέγιστη ακρίβεια). Έτσι όσο ο αλγόριθμος Adam συγκλίνει προς ένα ελάχιστο (τοπικό ή μέγιστο) ελαττώ-

νει το ”βήμα” που κάνει με στόχο να μην ”προσπεράσει” το ελάχιστο, έτσι οδηγώντας και στη σταδιακή μείωση της ταχύτητας σύγκλισης.

Επίσης σημειώνω επειδή ο αριθμός των εποχών που κάνω είναι σταθερός, υπάρχει πιθανότητα αν ο αριθμός των εποχών είναι μεγάλος, ο αλγόριθμος Adam να βρει το ελάχιστο και να το ”περάσει”, έτσι ώστε το μοντέλο με τα νέα βάρη να οδηγούν σε μεγαλύτερο σφάλμα στα validation sets από ότι προηγουμένως. Το φαινόμενο αυτό ονομάζεται overfitting και αντιμετωπίζεται με χρήση κατάλληλου κριτηρίου τερματισμού της εκπαίδευσης, με ομαλοποίηση και με άλλες τεχνικές. Επιλογή κριτηρίου τερματισμού και ομαλοποίηση κάνω στα επόμενα ερωτήματα.

3.f

3.f.1 Μπορεί να χρησιμοποιηθεί η τεχνική του πρόωρου σταματήματος (early stopping);

Η τεχνική του πρόωρου σταματήματος μπορεί να χρησιμοποιηθεί, σε περιπτώσεις όπου έχει επιλεγεί κατάλληλος ρυθμός μάθησης, έτσι ώστε η συνάρτηση σφάλματος ως προς τον αριθμό των εποχών να είναι φθίνουσα για όλο το χρονικό διάστημα που υπάρχει βελτίωση από εποχή σε εποχή. Άρα ο ρυθμός μάθησης δεν πρέπει να είναι πολύ μεγάλος, ταυτόχρονα όμως δεν πρέπει να είναι πολύ μικρός, επειδή τότε, σε περίπτωση που δεν υπάρχει εναλλακτική συνθήκη τερματισμού όπως μέγιστος αριθμός εποχών, η διαδικασία της μάθησης μπορεί να απαιτήσει πάρα πολύ χρόνο.

Σε αυτή τη περίπτωση ο ρυθμός μάθησης είναι κατάλληλος ώστε να μπορεί να χρησιμοποιηθεί τεχνική πρόωρου σταματήματος. Ιδανικά σε συνδυασμό βέβαια με καθορισμό ενός, σχετικά μεγάλου, αριθμού μέγιστων εποχών.

3.f.2 Επιλογή κατάλληλου κριτηρίου τερματισμού

Αν και η τεχνική πρόωρου σταματήματος μπορεί να χρησιμοποιηθεί για το μοντέλο αυτό (κάνουμε εκπαίδευση με συνάρτηση σφάλματος Categorical CE), το κριτήριο τερματισμού που προτιμάω σε αυτή τη περίπτωση είναι συνδυασμός πρόωρου σταματήματος με ένα μεγάλο σχετικά μέγιστο αριθμό εποχών (επέλεγα την τιμή 100).

Επίσης χρησιμοποιώ τιμή patience = 0, που είναι ο αριθμός των έξτρα εποχών που θα κάνει το πρόγραμμα αφού το κριτήριο τερματισμού γίνει αληθής, μήπως καταφέρει να επιτελέσει παραπάνω μείωση του σφάλματος. Άρα εγώ σταματάω την εκπαίδευση απευθείας μόλις ανιχνευθεί ότι δεν υπάρχει άλλη βελτίωση στο σφάλμα.

Επίσης χρησιμοποιώ τιμή min_delta = 0, που ορίζει τιμή διαφοράς μεταξύ τιμών του validation σφάλματος από εποχή σε εποχή που είναι αμελητέα και πυροδοτείται και πάλι το πρόωρο σταμάτημα. Άρα με τιμή μηδέν απενεργοποιώ αυτή τη λειτουργία.

Στα σχήματα παρακάτω φαίνεται παράδειγμα εκπαίδευσης του μοντέλου, όπου δείχνω τις τιμές του Categorical CE σφάλματος του training set και validation set ανά εποχή μέχρι τον τερματισμό, από πρόωρο σταμάτημα ή μέγιστο αριθμό εποχών:

4 Μεταβολές στον ρυθμό εκπαίδευσης και σταθεράς ορμής

Παρατηρήσεις:

- Ο αλγόριθμος βελτιστοποίησης Adam που διάλεξα, στη βιβλιοθήκη Keras που χρησιμοποιώ έχει την υπερπαράμετρο learning_rate, η οποία είναι η υπερπαράμετρος ρυθμού εκπαίδευσης η της εκφώνησης.
- Επίσης έχει την υπερπαράμετρο beta_1, η οποία ορίζει το decay rate για την πρώτη εκτίμηση κίνησης, δηλαδή αποτελεί την σταθερά της ορμής m, και την υπερπαράμετρο beta_2, η οποία ορίζει το decay rate για την δεύτερη εκτίμηση κίνησης, δηλαδή αποτελεί την σταθερά της ταχύτητας.
- Την παράμετρο beta_2, εφόσον δεν δίνονται δοκιμαστικές τιμές στην εκφώνηση για την σταθερά της ταχύτητας, θα την κρατήσω στην default τιμή της, δηλαδή 0.999.

Παρακάτω φαίνεται ο πίνακας με τις μετρικές αξιολόγησης που προέκυψαν από την διαδικασία μάθησης του νευρωνικού με συνάρτηση σφάλματος Categorical CE και χρήση μεθόδου βελτιστοποίησης Adam, με Learning Rate (η) και Beta 1 (m) τιμές αυτές που ζητούνται κάθε φορά. Στον πίνακα δείχνω μόνο τον μέσο όρο των τιμών των μετρικών που προέκυψαν από το evaluation με το validation set σε κάθε fold.

η	m	CE Loss	MSE	Acc
0.001	0.2	0.0600	0.0049	0.9842
0.001	0.6	0.0555	0.0046	0.9851
0.05	0.6	0.2593	0.0160	0.9460
0.1	0.6	0.4418	0.0338	0.8731

Σχήμα 3: Πίνακας μετρικών αξιολόγησης για βελτιστοποιητή Adam και τις διαφορετικές τιμές η και m

Από τις τιμές του παραπάνω πίνακα και από τις γραφικές παραστάσεις σύγκλισης, βγάζω τα εξής συμπεράσματα:

Ρυθμός εκπαίδευσης:

- Όσο μεγαλύτερος είναι ο ρυθμός εκπαίδευσης (learning rate) τόσο μεγαλύτερα βήματα κάνει ο αλγόριθμος βελτιστοποίησης με σκοπό να βρει το τοπικό ή ολικό ελάχιστο της συνάρτησης σφάλματος. Οπότε όσο μεγαλύτερος ο ρυθμός εκπαίδευσης τόσο λιγότερες εποχές χρειάζονται συνολικά μέχρι το κριτήριο του πρόωρου σταματήματος να σταματήσει τη διαδικασία της μάθησης. Αυτό φαίνεται και από τον αριθμό τον άξονα x των γραφικών παραστάσεων, που δείχνει τον αριθμό των εποχών σε κάθε περίπτωση.
- Ταυτόχρονα όμως αν ο ρυθμός εκπαίδευσης είναι πολύ μεγάλος, υπάρχει μεγάλη πιθανότητα η συνάρτηση βελτιστοποίησης να κάνει πολύ μεγάλο βήμα και να ”χάσει” το ελάχιστο, επειδή η επόμενη τιμή που εξετάζει έχει μεγαλύτερο σφάλμα, γίνεται το πρόωρο σταμάτημα και ο αλγόριθμος βρίσκει ένα ελάχιστο που μπορεί να αποκλίνει πολύ από το πραγματικό. Για αυτό η απόδοση παραπάνω για μεγάλο ρυθμό μάθησης είναι χειρότερη από τις περιπτώσεις με μικρότερο ρυθμό μάθησης.

Σταθερά ορμής:

- Όσο μεγαλύτερη είναι η σταθερά ορμής, τόσο λιγότερο λαμβάνονται υπόψη οι προηγούμενες τιμές στον υπολογισμό του σφάλματος. Άρα παρατηρούμε ότι όσο μικρότερο είναι το beta 1, τόσο πιο πολύ ”ταλαντώνεται” η γραφική παράσταση σύγκλισης, το οποίο συμβαίνει επειδή λαμβάνονται πολύ υπόψη οι νέες τιμές και πολύ λίγο οι παλιές, άρα η γραφική είναι πιο επιρρεπής σε μεγάλες μεταβολές. Βέβαια στις περιπτώσεις του learning rate = 0.001, η διαφορά για τις δύο τιμές του beta 1 είναι πολύ μικρές, επειδή το learning rate είναι τόσο μικρό, οπότε οι διαδοχικές τιμές διαφέρουν λιγότερο μεταξύ τους εξαρχής. Σε άλλες περιπτώσεις, όπως για μικρότερο learning rate, το φαινόμενο αυτό θα ήταν πιο εμφανές.

- Παρατηρούμε επίσης πως για τις μεγαλύτερες τιμές beta 1 τα σφάλματα και το accuracy είναι (λίγο) καλύτερα, για το ίδιο learning rate βέβαια, το οποίο συμβαίνει επειδή με μικρό beta 1, η διαδικασία early stopping είναι πιθανό να γίνει νωρίτερα και να μην βρεθεί το ίδιο καλό ελάχιστο σφάλμα, όπως θα βρισκόταν με μεγαλύτερο beta 1. Παρόλα αυτά η διαφορά μεταξύ beta 1 ίσο με 0.6 και 0.2 σε αυτή τη περίπτωση ήταν μηδαμινή, λόγω του μεγάλου learning rate.

5 Ομαλοποίηση

5.a Επιλογή Μεθόδου Ομαλοποίησης

5.a.1 Σύγκριση δύο μεθόδων

Αρχικά αναφέρω ότι η μέθοδος ομαλοποίησης L1 είναι καλύτερη όταν έχουμε πολλές εισόδους στο νευρωνικό δίκτυο ή πολλούς κόμβους στο νευρωνικό που δεν συμβάλλουν στην εξαγωγή των εξόδων, δηλαδή οι έξοδοι δεν εξαρτώνται από αυτά. Αυτό συμβαίνει επειδή έχει τη δυνατότητα να μηδενίσει τα βάρη τους, με αποτέλεσμα το μοντέλο να γίνεται ποιο απλό και τα πιο απλά μοντέλα έχουν μικρότερη πιθανότητα για overfitting. Αυτός ο μηδενισμός δεν είναι εφικτός με τη μέθοδο ομαλοποίησης L2, που χρησιμοποιεί την L2 νόρμα αντί για την L1 νόρμα.

Αντίθετα, η μέθοδος ομαλοποίησης L2 είναι καλύτερη όταν οι περισσότερες είσοδοι στο νευρωνικό και οι περισσότεροι κόμβοι του συμβάλλουν στη δημιουργία των εξόδων. Αυτό συμβαίνει επειδή λόγω του τετραγωνισμού στην L2 νόρμα, οι μεγάλες τιμές συμβάλλουν στη μεγαλύτερη ομαλοποίηση από την ομαλοποίηση που θα γίνονταν στη περίπτωση της L1 μεθόδου ομαλοποίησης.

5.a.2 Επιλογή και εφαρμογή στο νευρωνικό δίκτυο

Αρχικά θα πρέπει να επιλέξω ποια μέθοδο ομαλοποίησης. Επειδή τα περισσότερα δεδομένα στην περίπτωση αυτή και οι περισσότεροι νευρώνες είναι χρήσιμοι, με βάση τα παραπάνω θα προτιμήσω την L2 μέθοδο.

Επίσης πρέπει να αποφασίσω αν θα εφαρμόσω την ομαλοποίηση μόνο στους νευρώνες κρυφού επιπέδου ή και στους νευρώνες εξόδου. Επειδή όμως επηρεάζοντας τα βάρη για τους νευρώνες εξόδου, επηρεάζουμε τα τελικά ποσοστά (προκύπτει από την one-hot encoded έξοδο με χρήση Categorical CE συνάρτηση σφάλματος), η τελική επιλογή της κατηγορίας θα επηρεαστεί με αρνητικό τρόπο, οπότε επιλέγω να κάνω ομαλοποίηση μόνο στους νευρώνες κρυφού επιπέδου.

5.b Πίνακας και γραφικές παραστάσεις σύγκλισης

Παρακάτω φαίνεται ο πίνακας με τις μετρικές αξιολόγησης που προέκυψαν από την διαδικασία μάθησης του νευρωνικού με συνάρτηση σφάλματος Categorical CE, χρήση μεθόδου βελτιστοποίησης Adam με Learning Rate ίσο με 0.001 και Beta 1 ίσο με 0.6, και L2 κανονικοποίηση με την τιμή r που ζητείται κάθε φορά. Στον πίνακα δείχνω μόνο τον μέσο όρο των τιμών των μετρικών που προέκυψαν από το evaluation με το validation set σε κάθε fold.

Συντελεστής r	**CE Loss**	**MSE**	**Acc**
0.1	0.1326	0.0120	0.9614
0.5	0.1909	0.0175	0.9440
0.9	0.2359	0.0227	0.9256

Σχήμα 4: Πίνακας μετρικών αξιολόγησης για μέθοδο ομαλοποίησης L2 και τις διαφορετικές τιμές r

5.c Συμπεράσματα

Παρατηρώ ότι όσο μεγαλύτερος είναι ο συντελεστής r τόσο μεγαλύτερο είναι το σφάλμα και τόσο μικρότερο είναι το accuracy στο μέσο όρο των τιμών των που προέκυψαν από το evaluation στο validation set σε κάθε fold.

Αυτό πιστεύω συμβαίνει επειδή το μοντέλο μου δεν ήταν overfitted εξαρχής. Δηλαδή στην περίπτωση αυτή ο τυχαίος διαχωρισμός στα 5 folds και το cross-validation, η εφαρμογή early stopping και τα χαρακτηριστικά των δεδομένων του dataset είχαν ως αποτέλεσμα ένα μοντέλο το οποίο μπορούσε να γενικεύσει εξαρχής και σε νέα δεδομένα.

Επίσης παρατηρώ ότι όσο μεγαλύτερος ο συντελεστής r τόσο λιγότερο τέμνονται οι γραφικές παραστάσεις σύγκλισης στο validation set για κάθε fold. Αυτό με κάνει ακόμα πιο σίγουρο για το γεγονός ότι το μοντέλο μου δεν ήταν overfitted εξαρχής, γιατί τότε θα τέμνονταν παραπάνω οι γραφικές παραστάσεις για μεγαλύτερα r και δεν θα τέμνονταν τόσο για μικρά r.

Οπότε συμπεραίνω ότι με την εφαρμογή ομαλοποίησης στη περίπτωση αυτή απλά επηρεάζονται οι τιμές των βαρών με αποτέλεσμα το μοντέλο να καταλήγει να είναι underfitted. Παρόλα αυτά θα μπορούσα να εφαρμόσω μια μικρή τιμή r (πολύ μικρότερη από το 0.1 που έχω ήδη ελέγξει) με σκοπό να αντιμετωπίσω τυχόν overfitting που θα μπορούσε να προκύψει, αλλά για τόσο μικρή τιμή r η βελτίωση θα παρουσιαστεί θα είναι πολύ μικρή, εφόσον το συγκεκριμένο μοντέλο μαζί με το early stopping θα ήταν ελάχιστα overfitted εξαρχής.

6 Βαθύ Νευρωνικό Δίκτυο

6.a Περιγραφή λογικής στοίχισης κρυφών επιπέδων

Αρχικά αναφέρω ότι επέλεξα να φτιάξω ένα νευρωνικό δίκτυο με τρία κρυφά επίπεδα για το ερώτημα αυτό.

Απαντώντας τα ερωτήματα της εκφώνησης, δεν είναι καλό τα κρυφά επίπεδα να έχουν ίδιο αριθμό κόμβων, ούτε αυξανόμενο. Η ιδανική στοίχιση των κρυφών επιπέδων είναι με φθίνων αριθμό νευρώνων από κρυφό επίπεδο σε κρυφό επίπεδο, πηγαίνοντας προς την είσοδο στην έξοδο με το επίπεδο εισόδου να έχει τον μεγαλύτερο αριθμό νευρώνων και το επίπεδο εξόδου να έχει το μικρότερο αριθμό νευρώνων.

Ο λόγος για αυτό είναι επειδή πρέπει το νευρωνικό δίκτυο από επίπεδο σε επίπεδο να φτάνει όλο και πιο κοντά στη λύση, αναγνωρίζοντας χαρακτηριστικά σε κάθε επίπεδο στα δεδομένα. Παρόλα αυτά ο κανόνας αυτός δεν οδηγεί πάντα στις βέλτιστες λύσεις, όπως είδαμε και στο ερώτημα A2, όπου με 23 νευρώνες είχαμε την καλύτερη απόδοση (ενώ η είσοδος αποτελούταν μόνο από 18 νευρώνες). Οπότε για αυτό το λόγο η τελευταία δοκιμή που θα κάνω θα έχει παραπάνω νευρώνες στο πρώτο κρυφό επίπεδο από ότι νευρώνες εισόδου.

Παρόλα αυτά, ο παραπάνω κανόνας είναι καλός κανόνας να ακολουθεί κανείς για να μην έχει πολλούς ανενεργούς κόμβους (παραπάνω κόμβοι) και το αποτέλεσμα μη αποδοτικό (λιγότεροι κόμβοι) ή για να μην είναι πολύ αργή η εκπαίδευση.

6.b Δοκιμές με διαφορετικούς αριθμούς κόμβων και γραφικές παραστάσεις σύγκλισης

Σημειώσεις: Αναφέρω εδώ ότι σχετικά με τα χαρακτηριστικά του νευρωνικού δικτύου, συνεχίζω να χρησιμοποιώ συνάρτηση σφάλματος Categorical CE, βελτιστοποιητή Adam με learning rate = 0.001 και beta 1 = 0.6 και χωρίς να κάνω ομαλοποίηση, αφού αυτή χειροτέρευε τα αποτελέσματα στα προηγούμενα ερωτήματα. Επίσης στου νευρώνες των κρυφών επιπέδων χρησιμοποιώ παντού συνάρτηση ενεργοποίησης ReLU και στους νευρώνες εξόδου χρησιμοποιώ Softmax.

Παρακάτω φαίνεται ο πίνακας με τις μετρικές αξιολόγησης που προέκυψαν από την διαδικασία μάθησης του νευρωνικού δικτύου. Στον πίνακα δείχνω μόνο τον μέσο όρο των τιμών των μετρικών που προέκυψαν από το evaluation με το validation set σε κάθε fold.

KE 1	**KE 2**	**KE 3**	**CE Loss**	**MSE**	**Acc**
12	8	5	0.0826	0.0071	0.9773
18	15	12	0.0444	0.0039	0.9872
23	20	18	0.0351	0.0031	0.9900

Σχήμα 5: Πίνακας μετρικών αξιολόγησης για συνάρτηση σφάλματος MSE

6.c Συμπεράσματα

- Από τις παραπάνω μετρήσεις και συναρτήσεις σύγκλισης συμπεραίνω ότι η περίπτωση με τους 23 νευρώνες στο πρώτο κρυφό επίπεδο, 20 στο δεύτερο και 18 στο τρίτο ήταν η πιο αποδοτική, παρόλο που είχε παραπάνω νευρώνες στο πρώτο κρυφό επίπεδο από ότι νευρώνες εισόδου. Παρόλα αυτά η διαφορά με τη περίπτωση των 18 νευρώνων στο πρώτο κρυφό επίπεδο, 15 στο δεύτερο και 12 στο τρίτο ήταν μηδαμινή και θα μπορούσε να είναι και τυχαία.
- Οπότε συμπεραίνω πως η λογική στοίχισης που ανέφερα παραπάνω, αν και όχι πάντα η ιδανική, για αριθμούς νευρώνων κοντά στο υψηλότερο δυνατό με βάση τη λογική, είναι μια πολύ καλή αρχική προσέγγιση για τον σχεδιασμό βαθιών νευρωνικών δικτύων, που φτάνει κοντά στη μέγιστη δυνατή απόδοση.
- Ακόμα συμπεραίνω πως ο μεγαλύτερος αριθμός νευρώνων στα κρυφά επίπεδα έχει μια τάση συνήθως να γίνει καλύτερα αποτελέσματα,

αν και δεν γίνεται να αυξήσουμε τον αριθμό αυθαίρετα για λόγους ταχύτητας μάθησης και απαίτησης σε υπολογιστικούς πόρους.

- Επίσης το βαθύ νευρωνικό δίκτυο σε αυτή τη περίπτωση είχε καλύτερη απόδοση από τα νευρωνικά δίκτυα με ένα κρυφό επίπεδο που σχεδιάσαμε προηγουμένως, ειδικά για τις περιπτώσεις των 23 νευρώνων στο πρώτο κρυφό επίπεδο, 20 νευρώνων στο δεύτερο, 18 στο τρίτο και 18 στο πρώτο, 15 στο δεύτερο και 12 στο τρίτο. Μάλιστα τα νευρωνικά δίκτυα με ένα κρυφό επίπεδο δεν έφτασαν ποτέ στην απόδοση αυτών των δύο βαθιών μοντέλων.