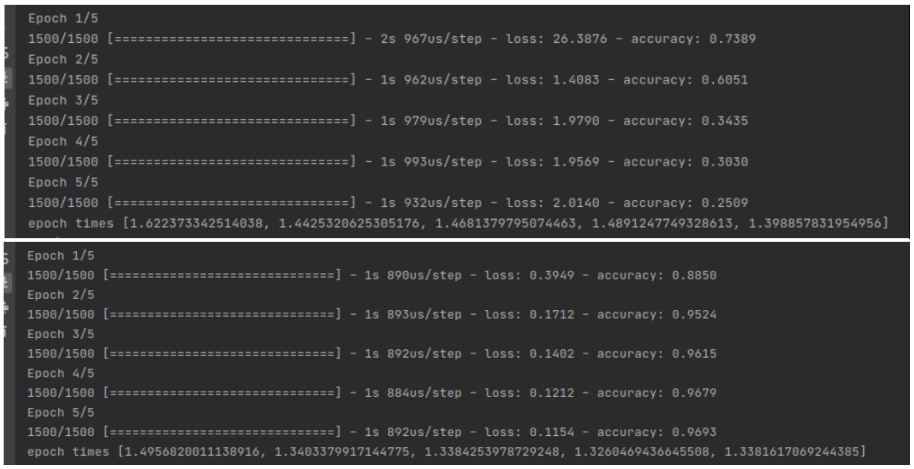
Υπολογιστική νοημοσύνη

Παρλαπάνης Αντώνιος

ΑΜ 1059709

A1. Προεπεξεργασία και Προετοιμασία δεδομένων

α)Η προεπεξεργασία και η προετοιμασία των δεδομένων είναι πολύ σημαντική για την αποτελεσματική λειτουργία των νευρωνικών δικτύων. Στο πλαίσιο αυτό , μια σύνηθη τακτική είναι να κανονικοποιούμε τα αριθμητικά δεδομένα. Στη δικιά μας περίπτωση μεταφέρουμε το έυρος τιμών των πιξελ των δειγμάτων στη κλιμακα [0-1].



Στη πρώτη εικόνα εκτελέσαμε τη μάθηση χωρίς κανονικοποίηση των δεδομένων, ενώ στη δεύτερη μεταφέραμε το ευρος των τιμών των πιξελ μεταξύ 0 και 1. Όπως παρατηρείται και η χρονική διάρκεια των εποχών αλλα και η ακρίβεια της μάθησης είναι αισθητά καλύτερα στη περίπτωση όπου κανονικοποιήσαμε τα δεδομένα. Ακόμα μετατρέψαμε τα δεδομένα ytrain και ytest σε διανύσματα one hot. Με τη τεχνική one hot encoding μπορούμε να εκφράσουμε τους αριθμούς σαν διανύσματα , καθώς απαιτείται απο τη συνάρτηση υπολογισμού σφάλματος categorical crossentropy.Τέλος μετατρέπουμε τα δεδομένα xtrain και xtest σε numpy arrays ώστε να μπορούν να χρησιμοποιηθούν απο τα μοντέλα του Keras.

β)Χρησιμοποιώντας τη κλάση Kfold του sklearn μπορούμε να υλοποιήσουμε το k-fold cross validation. Δημιουργούμε ένα αντικείμενο k-fold και δίνουμε σαν όρισμα τον αρίθμο των splits που θέλουμε και ότι θέλουμε να γίνει ανακάτεμα των δεδομένων πριν τον διαχωρισμό. Χρησιμοποιώντας τη συνάρτηση split και δίνοντας σαν όρισμα τα δεδομένα, μας επιστρέφεται σε κάθε επανάληψη του βρόγχου ένα γκρουπ των train και test δεδομένων. Το μοντέλο θα εκπαιδευτεί και έπειτα θα αξιολογηθεί πανω σε αυτά τα δεδομένα.

```
kfold = KFold(folds, shuffle=True, random_state=1)  # cross validation

for train_index, test_index in kfold.split(x_train, Y_train):
    xTrain, yTrain, xTest, yTest = x_train[train_index], Y_train[train_index], x_train[test_index], Y_train[
        test_index]  # train and test index

    history = model.fit(xTrain, yTrain, epochs=50, validation_data=(xTest, yTest),verbose=2)
    histories.append(history)
```

Α2. Επιλογή αρχιτεκτονικής

α)Με τη χρήση των παραπάνω μετρικών αξιολογούμε πόσο καλά το νευρωνικό μας δίκτυο μοντελοποιεί τα δεδομένα εισόδου. Κατά τη διάρκεια της εκπαίδευσης του νευρωνικού δικτύου , αν οι προβλέψεις αποκλίνουν πάρα πολύ από τα πραγματικά αποτελέσματα, το loss function θα μας επιστρέψει σαν loss έναν πολύ μεγάλο αριθμό. Σταδιακά με τη βοήθεια των optimization functions , το νευρωνικό μαθαίνει να μειώνει το loss στις προβλέψεις.

Ο αλγόριθμος cross-entropy χρησιμοποείται για προβλήματα ταξινόμησης , όπως την ταξινόμηση χειρόγραφων ψηφίων σε αρίθμους (0-9) την οποία προσπαθούμε να πετύχουμε.

Η έξοδος του νευρωνικού δικτύου η αλλιώς πρόβλεψη είναι ένα διάνυσμα μεγέθους 10. Κάθε θέση του διανύσματος αντιπροσωπεύει έναν αριθμό από το 0 μέχρι το 9. Οι τιμές που παίρνει κάθε θέση του διανύσματος έιναι η πιθανότητα έκεινος ο αριθμός να είναι το ψηφίο που δώθηκε σαν είσοδος στο νευρωνικό. Ολές οι τιμές του διανύσματος , δηλαδή οι πιθανότητα κάθε κλάσης ή αριθμού , έχουν άθροισμα 1.Ο αλγόριθμος cross-entropy συγκρίνει το διάνυσμα - πρόβλεψη με το διάνυσμα στόχο , το οποίο είναι η πραγματική τιμή του ψηφίου που δώθηκε σαν είσοδος . Το διάνυσμα στόχος είναι ένα one-hot διάνυσμα αφού έχει την τιμή ένα μόνο στον σωστό αριθμό και τα υπόλοιπα μηδενικά.

Το cross-entropy loss function χρησιμοποιείται για προβλήματα ταξινόμησης σε πολλαπλές κατηγορίες. Το cross-entropy θα υπολογίσει μια βαθμολογία που συνοψίζει τη μέση διαφορά μεταξύ της πραγματικής και της προβλεπόμενης κατανομής πιθανότητας για όλες τις τάξεις του προβλήματος.

β)Για το μοντέλο μας πρέπει να χρησιμοποιήσουμε σαν εισόδους διανύσματα. Έτσι τις εικόνες διαστάσεων 28*28 pixel θα τις μετατρέψουμε σε διανύσματα των 784 στοιχείων. Συνεπώς θα έχουμε 784 εισόδους.

γ)Οι δέκα αναμένομένες τιμές (0-9) από την έξοδο του νευρωνικού μας οδηγούν στον να χρησιμοποιήσουμε 10 νευρώνες στο επίπεδο εξόδου. Αυτό μας επιτρέπει σε συνδυασμό με τη συνάρτηση ενεργοποιήσης softmax να χρησιμοποιήσουμε αυτές τις δέκα τιμές εξόδου σαν πιθανότητες, και η μεγαλύτερη από αυτές να χρησιμοποιηθεί σαν πρόβλεψη.

δ)Για τους κρυφούς κόμβους ως συνάρτηση ενεργοποίησης επιλέχθηκε η Relu. Η relu είναι μια πολύ απλή συνάρτηση ενεργοποίησης. Αν η τιμή της εξόδου του νευρώνα είναι μεγαλύτερη από 0 παραμένει , αλλιώς μηδενίζεται. Λόγω της απλής μορφής της βελτιώνεται πολύ η απόδοση του νευρωνικού δικτύου , καθώς η ταχύτητα σύγκλισης είναι πολύ γρηγορότερη από τις αλλές συνήθεις συναρτήσεις ενεργοποίησης αφού απαιτείται μόνο μία τιμή κατωφλίου για να ληφθεί η τιμή ενεργοποίησης η οποία μπορεί να εξοικονομήσει πολύ χρόνο προπόνησης του gradient descent χωρίς πολλούς πολύπλοκους υπολογισμούς. Ακόμα η relu κάνει πολλούς νευρώνες να έχουν έξοδο 0 που οδηγεί σε ένα αραίο δίκτυο αφού πολλοί νευρώνες πάυουν να πυροδοτούνται. Σε αντίθεση με τη sigmoid όπου σχεδόν κάθε νευρωνας ενεργοποιέιται κάνωντας το δίκτυο πυκνό και υπολογίστικα κοστοβόρο. Τέλος με τη relu μειώνεται η αλληλοεξάρτηση των παραμέτρων και συνεπώς μικραίνει το πρόβλημα του overfitting και κάνει το μοντέλο συνολικά ποιο αποδοτικό.

δ)Απο την έξοδο του νευρωνικού δικτύου θα πάρουμε ένα διάνυσμα μεγέθους δέκα το οποίο θα περιέχει ¨ακατέργαστες τιμές¨ οι οποίες δν σημαίνουν και πολλά όταν τις βλέπεις πχ ([-0.2, 1.3, -0.7, 2.2,0.1,2,4.3,6.4,3.2,1]). Ο σκοπός μας είναι να μετατρέψουμε αυτές τις τιμές σε μια κατανοητή μορφή. Συγκεκριμένα θέλουμε να μετατρέψουμε αυτές τις τιμές σε πιθανότητες.Για να το πετύχουμε αυτό μπορούμε να χρησιμοποιήσουμε είτε την σιγμοειδή συνάρτηση ενεργοποιήσης είτε την softmax. Αυτές οι δύο έχουν μια σημαντική διαφορά. Οι πιθανότητες οι οποίες παράγονται από τη σιγμοειδή συνάρτηση είναι ανεξάρτητες μεταξύ τους, και ουσιαστικά παράγονται περισσότερες από μία σωστές απαντήσεις. Αντιθέτως η softmax παράγει πιθανότητες οι οποίες έχουν άθροισμα 1 , και βγάινει ένα σωστό τελικό αποτέλεσμα. Δηλαδή δεν γίνεται ένα ψηφίο να έχει πιθανότητα να είναι 8 κατά 80% και 6 κατά 60%.

Στη περίπτωση της αναγνώρισης χειρόγραφων ψηφίων λοιπόν θέλουμε το νευρωνικό μας δίκτυο να επιλέξει έναν αριθμό ως τελική έξοδο. Άλλωστε δν γίνεται ένας αριθμός να έχει δυο τιμές ταυτόχρονα.

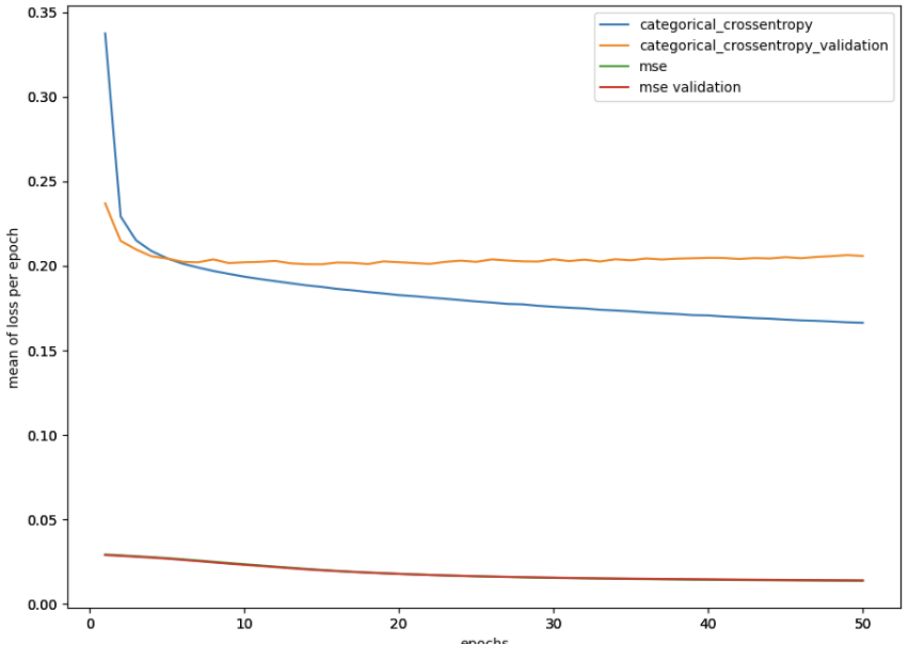
Συμπερασματικά θα χρησιμοποιήσουμε softmax αφού στην έξοδο του νευρωνικού υπάρχει μόνο μία σωστή απάντηση. Αν στο πρόβλημα μας υπηρχαν παραπάνω από μία σωστές απαντήσεις θα χρησιμοποιούμε τη σιγμοειδή.

στ)

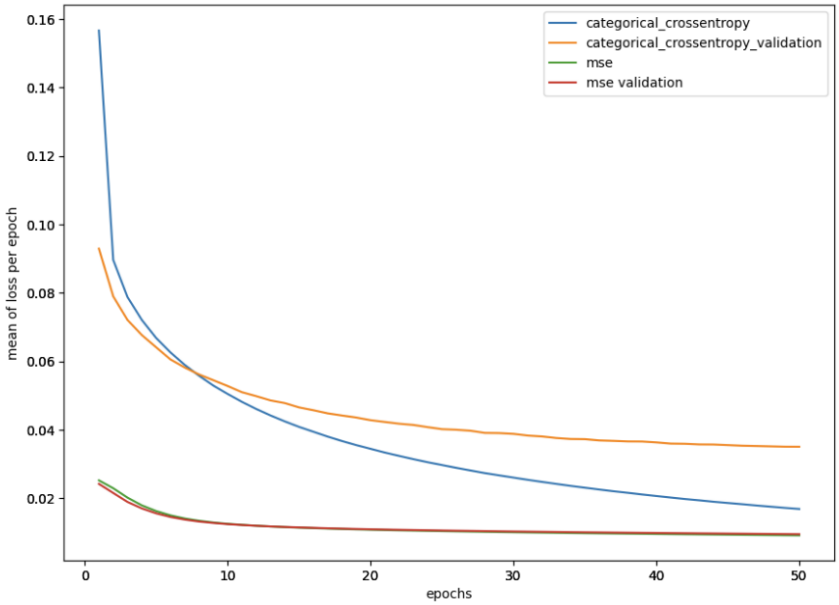
Στο παρακάτω πινακάκι βλέπουμε τον μέσο όρο των fold για το τελευταίο epoch.

Αριθμός νευρώνων στο κρυφό επίπεδο	CE validation loss	MSE validation loss
H1 = O	0.205	0.014
H1 = (I+O)/2	0.033	0.009
H1 = I+O	0.035	0.009

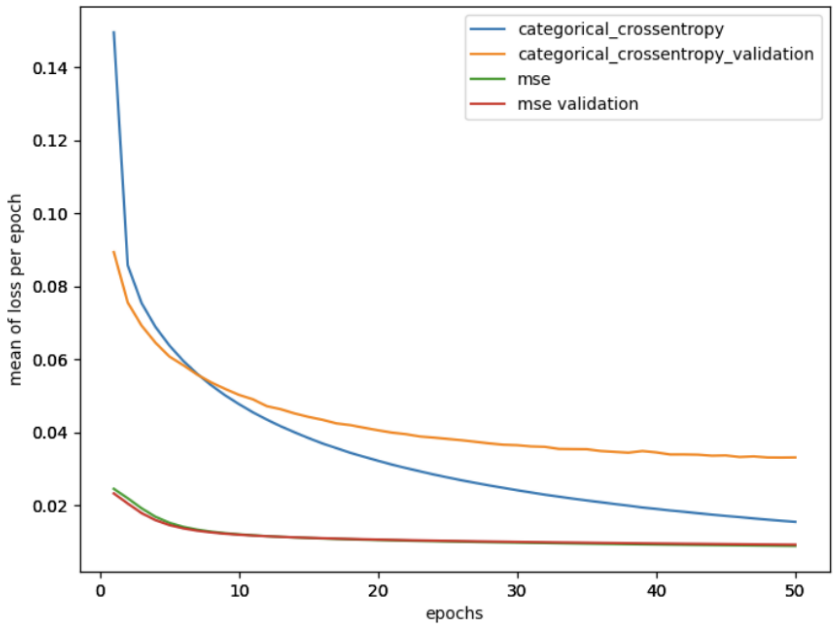
Στις παρακάτω γραφικές παραστάσεις φαίνεται ο μέσος όρος του σφάλματος των fold ανα κύκλο εκπαίδευσης (εποχή) , καθώς και ο μέσος όρος του σφάλματος επικύρωσης ανα εποχη.Το πείραμα έγινε έγινε με 50 εποχές εκπαίδευσης και default learning rate.



Σχήμα 1: CE/MSE train-loss & validation-loss για 10 κόμβους



Σχήμα 2: CE/MSE train-loss & validation-loss για 397 κόμβους



Σχήμα 3: CE/MSE train-loss & validation-loss για 10 κόμβους

α΄. Συμπεράσματα

i)Οι αλλαγές στην απόδοση είναι αισθητές καθώς αυξάνουμε τους κόμβους από 10 σε 397.Το σφάλμα και στις δυο περιπτώσεις (cross-entropy loss mse loss) είναι σημαντικά μικρότερο. Παρολό αυτά στο τελευταίο μέρος του πειράματος , όπου αυξάνουμε τους κόμβους του κρυφού επιπέδου σε 794 παρατηρούμε ότι σφάλμα μειώθηκε κατά πολύ λίγο σε σχέση με τους 397 κόμβους παρόλο που διπλασιάσαμε του κόμβους . Αυτό συμβαίνει γιατί όσο αυξάνουμε τους κόμβους σε ένα νευρωνικό δίκτυο αυτό μαθαίνει όλο και περισσότερα μοτίβα. Αυτό είναι πολύ πιθανό να οδήγησει σε overfitting ενός νευρωνικού δικτύου πάνω σε συγκεκριμένα δεδομένα. Από τα δεδομένα που προέκυψαν λοιπόν καταλήγουμε στο ότι ο ιδανικός αριθμός κόμβων για το συγκεκριμένο νευρωνικό δίκτυο βρίσκετε ανάμεσα στους 397 και 794 κόμβους.

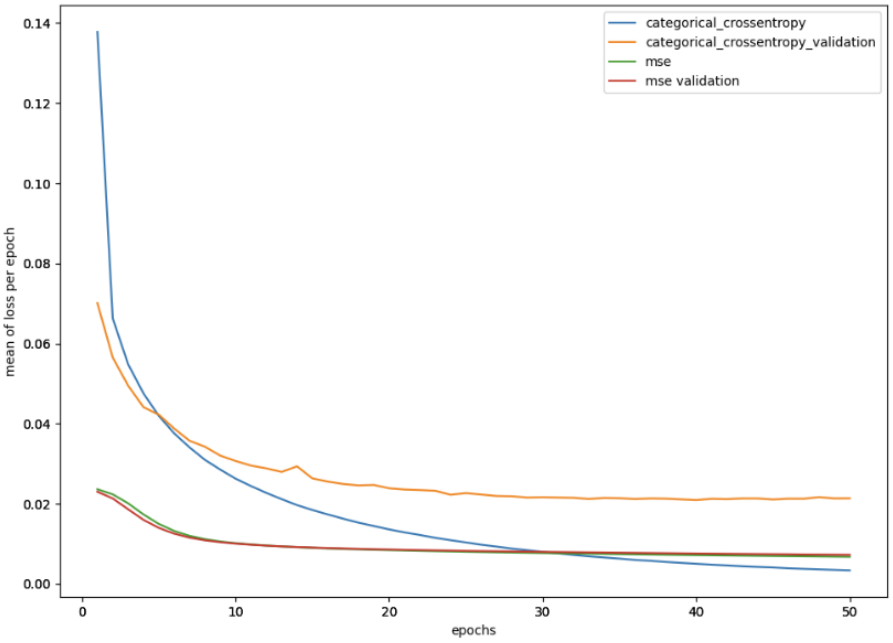
ii)Η συνάρτηση κόστους CE χρησιμοποιείται κυρίως για classification problems , δηλαδή για προβλήματα όπου η έξοδος ανήκει σε ένα σετ κλάσεων, όπως το πρόβλημα που αντιμετωπίζουμε εδώ. Παρόλο που η συνάρτηση mse δίνει συστηματικά μικρότερο loss θα επιλέξουμε να χρησιμοποιήσουμε τη συνάρτηση ΅Ε. Η συνάρτηση ΅Ε τιμώρει πολύ περισσότερο τα μέγαλα λάθη από ότι τα μικρά. Αυτό έχει σαν αποτέλεσμα να μπορούμε να διαχωρίσουμε τις κλάσεις καλύτερα μεταξύ τους, και συνεπώς να έχουμε καλύτερο αςςυραςψ κατα τη διάρκεια του αλιδατιον.

iii)Από τα διαγράμματα παρατηρούμε ότι ήδη από τις 20 εποχές το νευρωνικό είχε ήδη συγκλίνει αρκετά καλά σε όλες τις περιπτώσεις. Συμπεράνω ότι θα μπορούσα να είχα χρησιμοποιήσει λιγότερες από 50 εποχές στα πειράματα έχοντας περίπου τα ίδια αποτελέσματα αλλά με σημαντικά λιγότερο χρόνο.

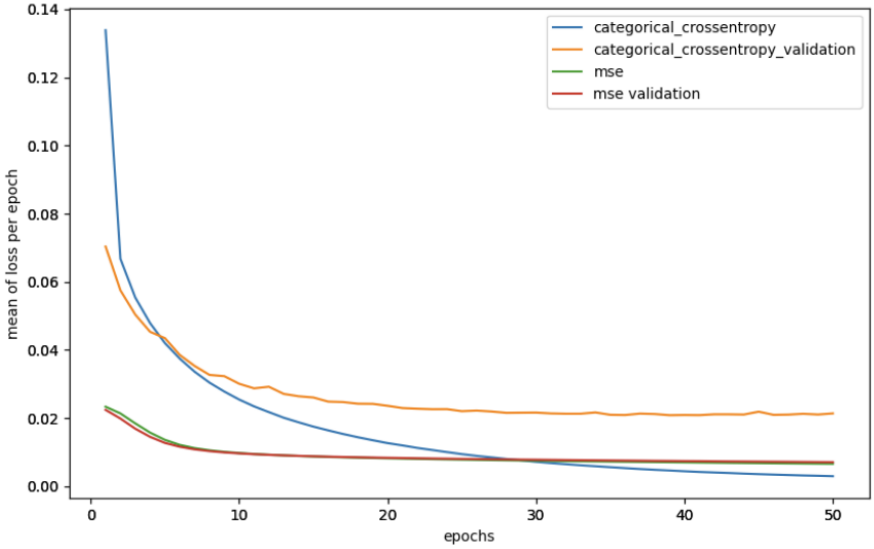
ζ

Αριθμός νευρώνων στο κρυφό επίπεδο	CE validation loss	MSE validation loss
H2 = 397	0.020	0.007
H2 = 128	0.022	0.007
H2 = 512	0.021	0.007

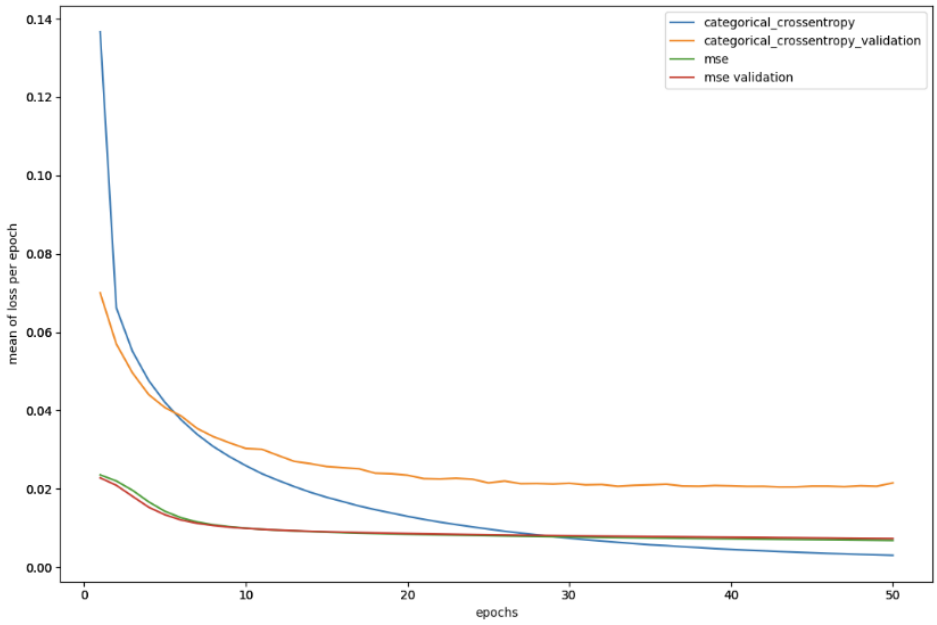
Στις παρακάτω γραφικές παραστάσεις φαίνεται ο μέσος όρος του σφάλματος των fold ανα κύκλο εκπαίδευσης (εποχή) , καθώς και ο μέσος όρος του σφάλματος επικύρωσης ανα εποχη.Το πείραμα έγινε έγινε με 50 εποχές εκπαίδευσης και default learning rate. Το πρώτο κρυφό επίπεδο του μοντέλου αποτελείται απο 397 κόμβους.



Σχήμα 4: CE/MSE train-loss & validation-loss για H2 = 397



Σχήμα 5: CE/MSE train-loss & validation-loss για H2 = 128



Σχήμα 6: CE/MSE train-loss & validation-loss για H2 = 512

Συμπεράσματα

Η λογική λέει ότι προσθέτωντας κρυφά επίπεδα θα δίναμε στο μοντέλο μας περισσότερη ευελιξία στο να κάνει προβλέψεις. Στη πραγματικότητα δεν ισχύει πάντα. Τα νευρωνικά δίκτυα μπορεί να προσαρμόσουν τις παραμέτρους τους στα δεδομένα που κάνουν τραιν τόσο πολύ που να μη μπορούν να ανταπέξελθουν σε άλλα δεδομένα εκτός του train set. Αυτό το πρόβλημα ονομάζεται overfitting.

Παρόλο αυτά βλέπουμε ότι προσθέντωντας ένα δεύτερο κρυφό επίπεδο τα σφάλματα μειώθηκαν σε όλες τις περιπτώσεις.

Ακόμα, μετά από ένα πείραμα αξιολόγησης με το test set παρατηρήθηκε ότι το αγσυραγψ είχε μια πολύ μικρή βελτίωση όταν προσθέσαμε ακόμη ένα κρυφό επίπεδο με το ίδιο πλήθος κόμβων (397). Πιο συγκεκριμένα το αγσυραγψ αυξήθηκε από 97,85% σε 98,17%. Το ίδιο βέβαια δεν ίσχυει και για το δευτερο κρυφό επίπεδο με 128 κόμβους όπου εκεί είδαμε πτώση του αγσυραγψ στο 97,78%. Τέλος σε πείραμα με δέυτερο κρυφό επίπεδο των 512 κόμβων παρατηρήθηκε ότι το αγσυραγψ είχε ελλατωθεί σε σχέση με το δεύτερο κρυφό επίπεδο των 397 κόμβων (από 98,17% σε 98,01%).

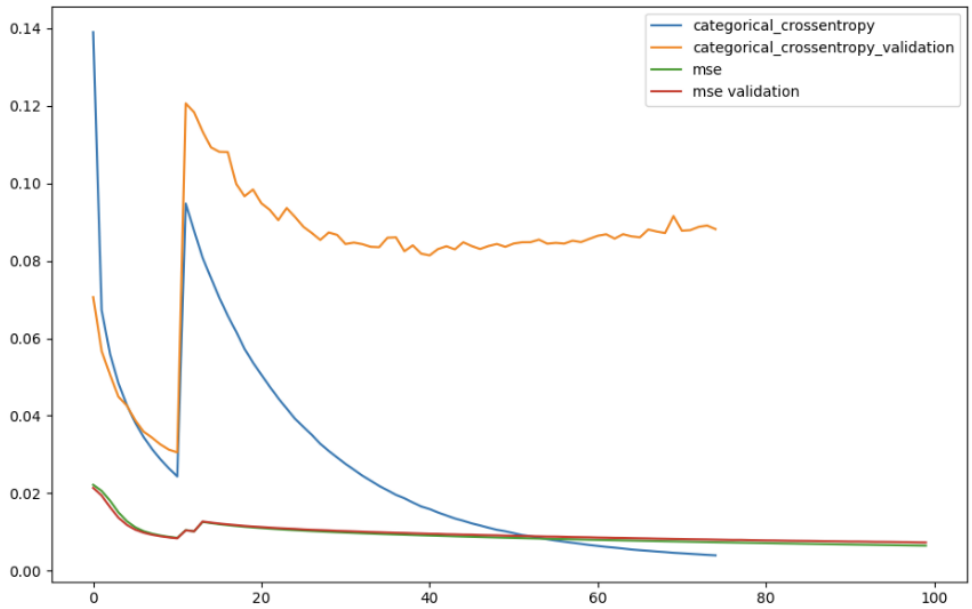
Συμπερασματικά τα επιπλέον κρυφά επίπεδα μπορούν να αυξήσουν την αποτελεσματικότητα ενός νευρωνικού δικτύου. Όμως , όσο αυξάνεται η πολυπλοκότητα των μοντέλων και η πληροφορία σχετικά με το σφάλμα πρέπει να διαθοθεί σε περισσότερα επίπεδα , τα μόντελα μπορεί μπορεί να αποτύχουν να μάθουν, κυρίως λόγω έλλειψης πληροφορίας. Ακόμα η άυξηση των κόμβων και των κρυφών επιπέδων αυξάνει σημαντικά το χρόνο που απαιτείται για την εκπαίδευση των μοντέλων.Για αυτό θα πρέπει να βρέθει το ιδανικό σημείο σε αριθμό κόμβων,εποχών, και κρυφών επιπέδων όπου δεν θα διακυβεύετε ούτε η αποτελεσματικότητα αλλά ούτε και πολύτιμος χρόνος. Για αυτούς τους λόγους επιλέγω σαν ιδανική τοπολογία τα δύο κρυφά επίπεδα με 397 κόμβους το καθένα.

η)Ένα πρόβλημα στην εκπαίδευση των νευρωνικών δικτύων είναι η επιλογή των κατάλληλων κύκλων εκπαίδευσης (εποχών). Πάρα πολλές εποχές μπορούν να οδηγήσουν σε overfitting του training data set , ενώ πολύ λίγες μπορεί να έχουν σαν αποτέλεσμα ένα μοντέλο με μη επαρκή εκπαίδευση. Για την λύση αυτού του προβλήματος μπορεί να χρησιμοποιηθεί η τεχνική του early stopping. Αυτή η τεχνική μας επιτρέπει να χρησιμοποιήσουμε έναν αυθαίρετα μεγάλο αριθμό εποχών και να σταματήσουμε την εκπαίδευση όταν η απόδοση του μοντέλου σταματήσει να βελτιώνεται σε ένα σύνολο δεδομένων.

Σαν κριτήριο τερματισμού μπορεί να χρησιμοποιηθεί είτε το loss είτε το accuracy από τα validation data.Στο συγκεκριμένο πρόβλημα που αντιμετωπίζουμε πιο σημαντικός παράγοντας είναι το accuracy. Το loss μας δείχνει πόσο σίγουρο είναι το νευρωνικό μας για μία πρόβλεψη. Δηλαδή αν έβγαλε μια πρόβλεψη με πιθανότητα 60% η 90% κτλπ.Στη περίπτωση μας , λόγω της συνάρτησης ενεργόποίησης softmax στην έξοδο , επιλέγεται μία τιμή, αυτή με τη μεγαλύτερη πιθανότητα. Ετσι δεν μας ενδιαφέρει σε ποιους αριθμούς μοίρασε τις πιθανότητες το νευρωνικό αρκει να έδωσε τη μεγαλύτερη πιθανότητα στον σωστό αριθμό. Ο αριθμός με τη μεγαλύτερη πιθανότητα θα βγεί και σαν τελικό αποτέλεσμα . Ετσι σαν καταλληλοτερο κριτήριο τερματισμού επιλέγω τη μεγιστοποίηση του validation accuracy. Τέλος άλλος ένας λόγος που χρησιμοποιούμε το validation accuracy σαν κριτήριο τερματισμού είναι ότι το λοσς λόγω του SGD δεν είναι απαραίτητο να μειώνεται σε κάθε εποχή. Ετσι είμαστε σίγουροι οτι χρησιμοποιούμε τις ελάχιστες δυνατές εποχές πετυχαίνοντας ενα ικανοποιητικό αποτέλεσμα.

Τέλος το cross validation είναι μια μέθοδος για την εκτίμηση της ακρίβειας της γενίκευσης ενός αλγορίθμου μάθησης. Από την άλλη η πρόωρη διακοπή (early stopping) είναι μια μέθοδος για την αποφυγή της υπερβολικής μάθησης και απαιτεί μια μέθοδο για την αξιολόγηση της σχέσης μεταξύ της ακρίβειας της γενίκευσης του εκπαιδευμένου μοντέλου και της ακρίβειας της εκπαίδευσης. Επομένως μπορούμε να χρησιμοποιήσουμε τη διασταυρωμένη επικύρωση (cross validation) για να αντικαταστήσουμε το σύνολο δεδομένων επικύρωσης σε ένα πλαίσιο πρόωρης διακοπής χωρίς πρόβλημα.

Υποσημείωση : Η συνάρτηση averageOfNestedListsDiffEpochs() που βρίσκετε στο αρχείο early stopping λύνει το πρόβλημα των διαφορετικών εποχών ανα fold, λόγω του early stopping, στην εύρεση του mean. Η συνάρτηση mean που χρησιμοποιήθηκε στα προηγούμενα ερώτηματα , κολλάει όταν βρεί null value σε μια θέση.



Σχήμα 7: CE/MSE train-loss & validation-loss early stopping με κριήριο validation accuracy

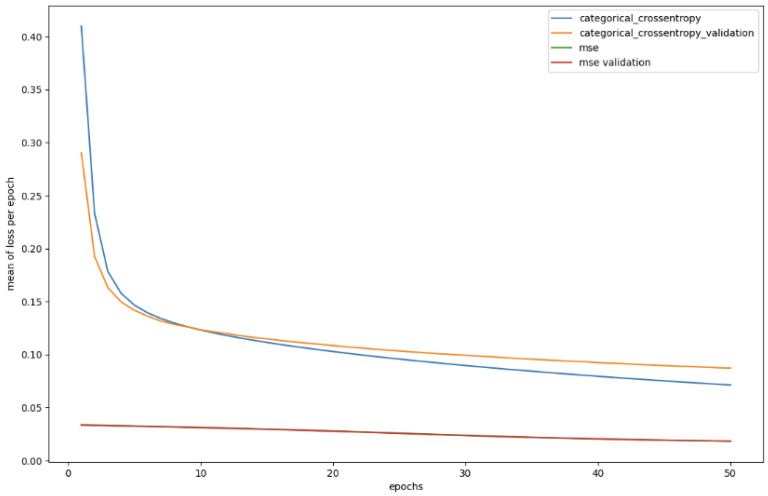
Παρόλο που το validation loss της CE είναι ψηλό σε σχέση με την αρχιτεκτονική χωρίς early stopping , το validation accuracy που χρησιμοποιώ σαν κριτήριο ήταν πάνω από 99% και το accuracy από το τεστ που πραγματοποιήθηκε με το test data set ήταν 98,1

1. Α3. Μεταβολές στον ρυθμό εκπαίδευσης και σταθεράς ορμής

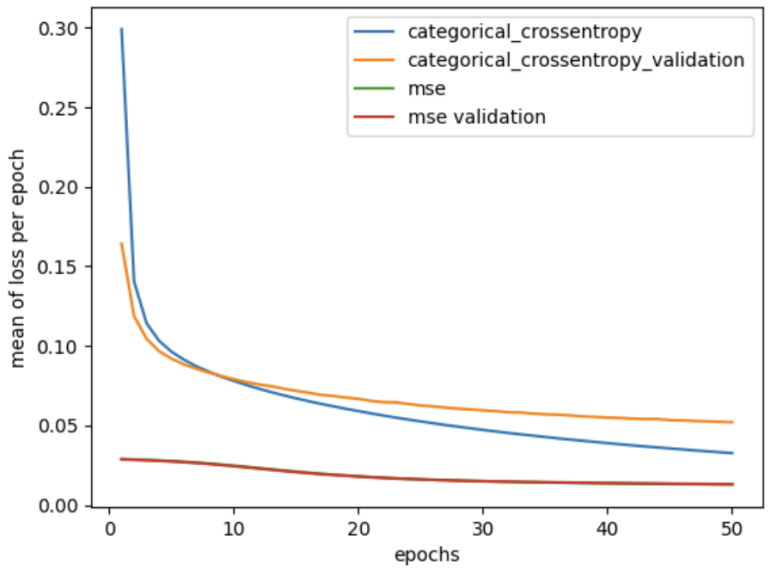
Θα χρησιμοποιηθεί η τοπολογία που δίνει τα χαμηλότερα validation loss. Έτσι θα χρησιμοποιήσω 2 κρυφά επίπεδα με 397 κόμβους το καθένα. Στα παρακάτω πειράματα δεν θα χρησιμοποιηθεί το early stopping για να δούμε το απότελεσμα των αλλαγών του learning rate και του momentum σε συνάρτηση με τα προηγούμενα ερωτήματα.

η	m	CE validation loss	MSE validation loss
0.001	0.2	0.087	0.018
0.001	0.6	0.052	0.013
0.05	0.6	0.020	0.0015
0.1	0.6	0.020	0.001

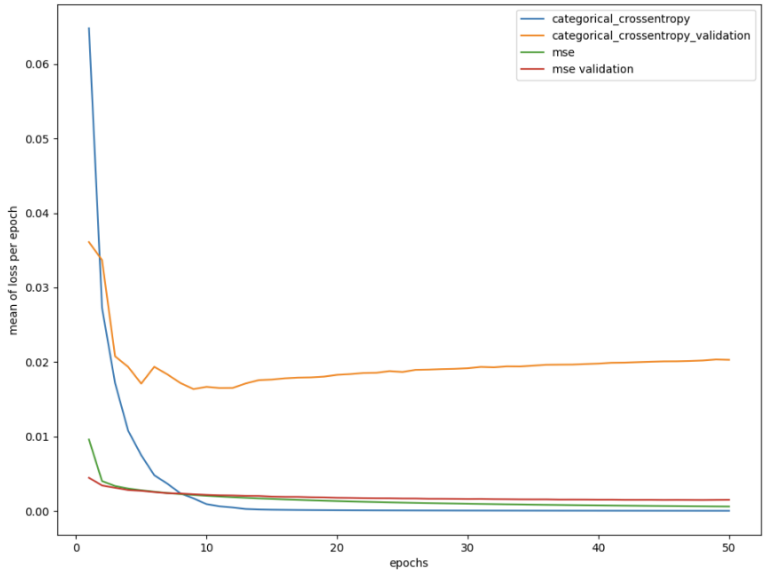
Στα παρακάτω διάγραμματα φαίνεται ο μέρος όρος του σφάλματος των fold για κάθε εποχή, καθώς και το σφάλμα από τα validation data.



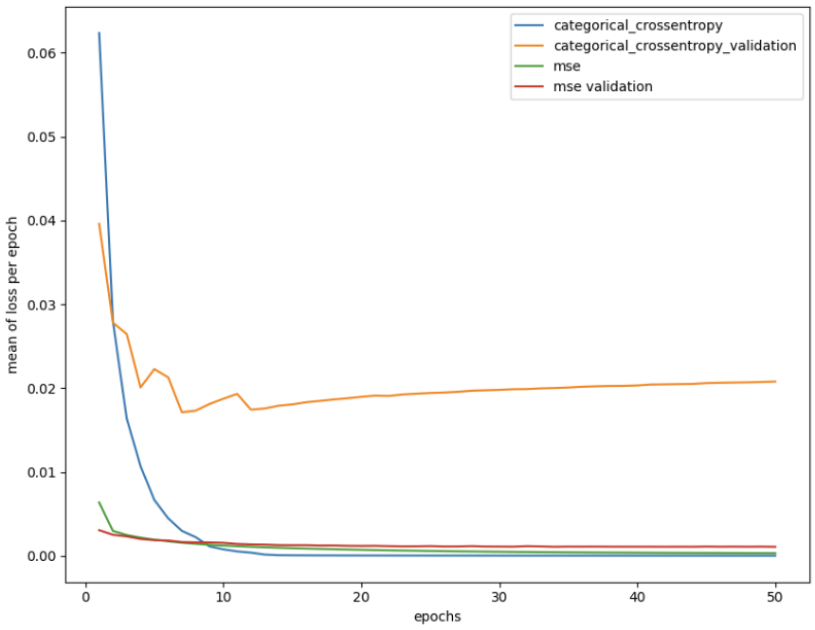
Σχήμα 8: CE/MSE train-loss & validation-loss για lr = 0.001 & m = 0.2



Σχήμα 9: CE/MSE train-loss & validation-loss για lr = 0.001 & m = 0.6



Σχήμα 10: CE/MSE train-loss & validation-loss για lr = 0.05 & m = 0.6



Σχήμα 11: CE/MSE train-loss & validation-loss για lr = 0.1 & m = 0.6

α΄. Συμπεράσματα

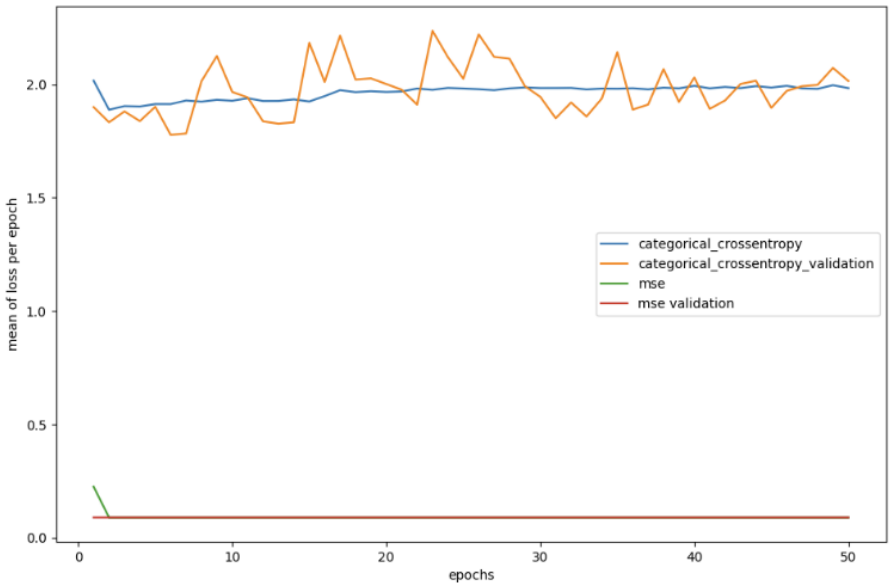
Από τα παραπάνω πειράματα βλέπουμε ότι μοντέλο ηταν ικανό να μάθει σε όλες τις τιμές του learning rate. Παρόλο αυτά η ολόκληρωση του training έπαιρνε όλο και περισσότερο χρόνο όσο μικραίναμε το λεαρνινγ ρατε ενώ ταυτόχρονα είχαμε αισθητά μεγαλύτερο validation loss. Συνεπώς καταλήγουμε στη τιμή 0.1 σαν βέλτιστο ρυθμό μάθησης. Ακόμα , από τα πρώτα δύο πειράματα παρατηρούμε ότι η προσθήκη μομεντυμ βοήθησε στην μείωση των validation loss , αλλα δεν είδαμε να βοηθάει στη γρηγορότερη σύκλιση του μοντέλου, καθώς η καμπύλη του train loss παρέμεινε σχεδόν ίδια.

Για να πετύχουμε η σύγκλιση θα πρέπει η σταθερά ορμής να είναι μικρότερη του 1 , αλλιώς το γραδιεντ του προήγουμενου βήματος θα συνέβαλε περισσότερο από το πραγματικό gradient. Ο αλγόριθμος πρέπει να εξαρτάται από τα προηγούμενα βήματα όλο και λιγότερο αντι για όλο και περισσότερο.

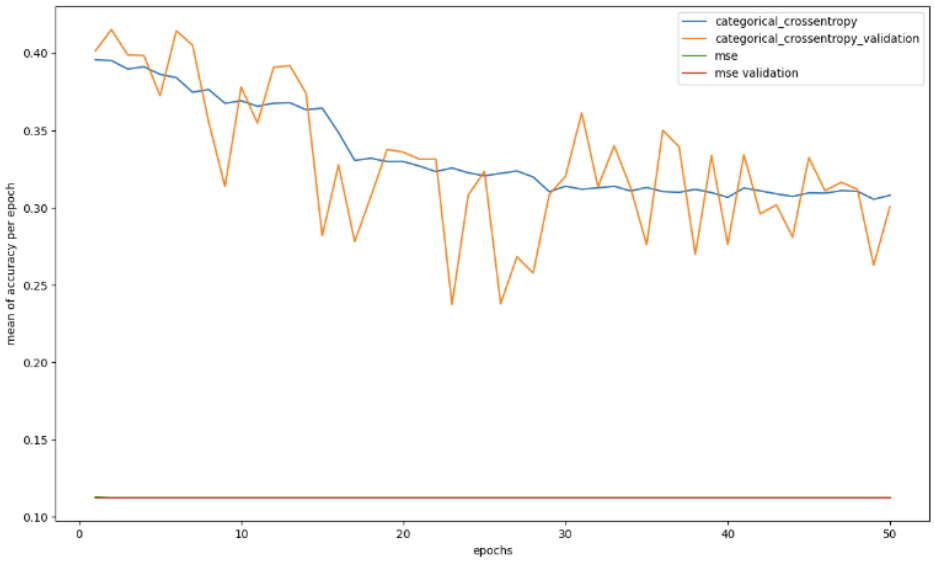
2. Α4. Ομαλοποίηση

Τα πειράματα έγιναν με 50 εποχές.

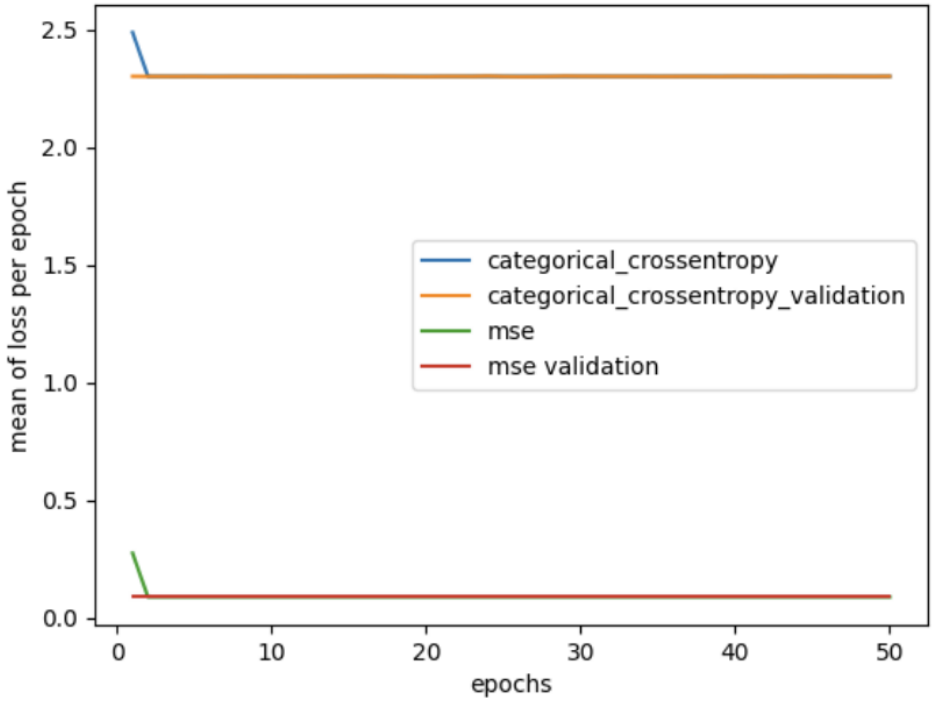
Συντελεστές φθοράς	CE validation loss	MSE validation loss
0.1	1.983	2.015
0.5	2.303	0.089
0.9	2.302	0.089



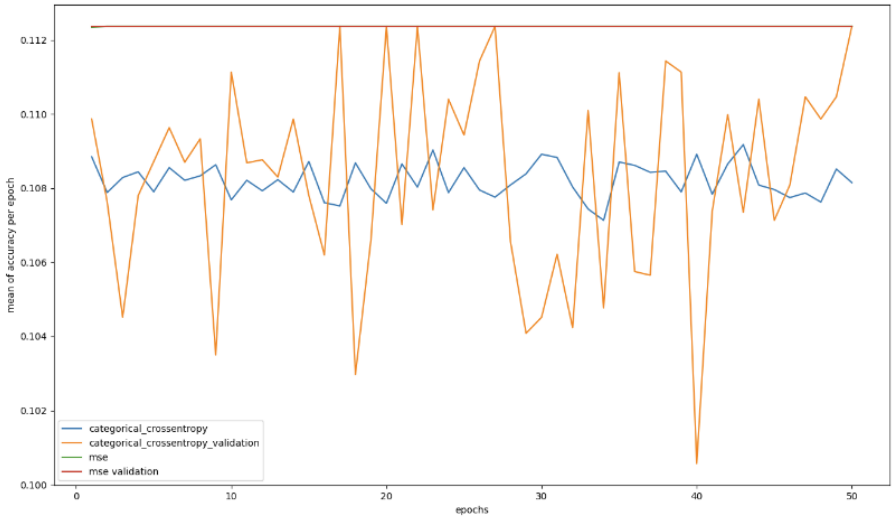
Σχήμα 12: CE/MSE train-loss & validation-loss για r = 0.1



Σχήμα 13: CE/MSE train-accuracy & validation-accuracy για r = 0.1



Σχήμα 14: CE/MSE train-loss & validation-loss για r = 0.5



Σχήμα 15: CE/MSE train-loss & validation-loss για r = 0.9

α΄. Συμπεράσματα

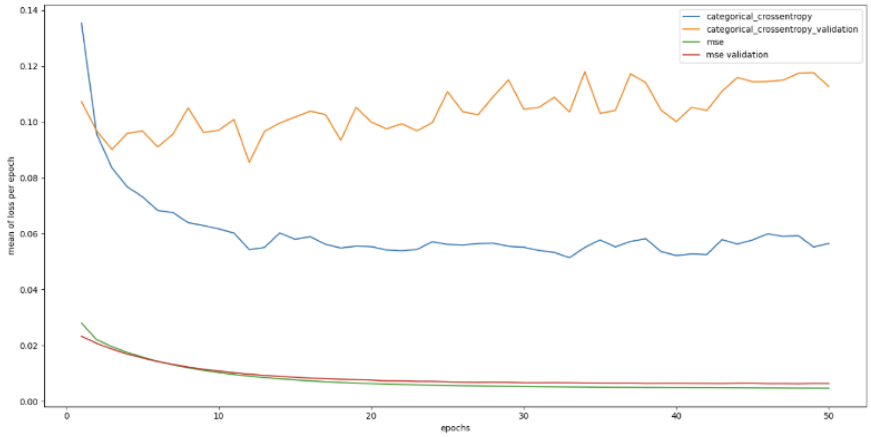
Η L2 κανονικοποίηση τιμωρεί τα βάρη με μεγάλα μεγέθη. Τα μεγάλα βάρη είναι το πιο προφανές σύμπτωμα του υπερβολικού overfitting επομένως αυτή η ομαλοποίηση θα μπορούσε να βοήθησει να αποφύγουμε αυτό το πρόβλημα.

Από τα πειράματα βλέπουμε ότι ο επιπρόσθετος θόρυβος από την ομαλοποίηση L2 είναι περισσότερος από ότι χρειάζεται και συνεπώς κάνει το δίκτυο να αγνοεί τα αδύναμα αλλα πολύτιμα μοτίβα μεταξύ των αριθμών. Έτσι έχουμε σημαντική πτώση στο accuracy του μοντέλου. Αυτό μας κάνει να αναρωτιόμαστε αν θα έπρεπε εξαρχής να μειώσουμε τα βάρη ώστε να βελτιώσουμε τα αποτέλεσματα.

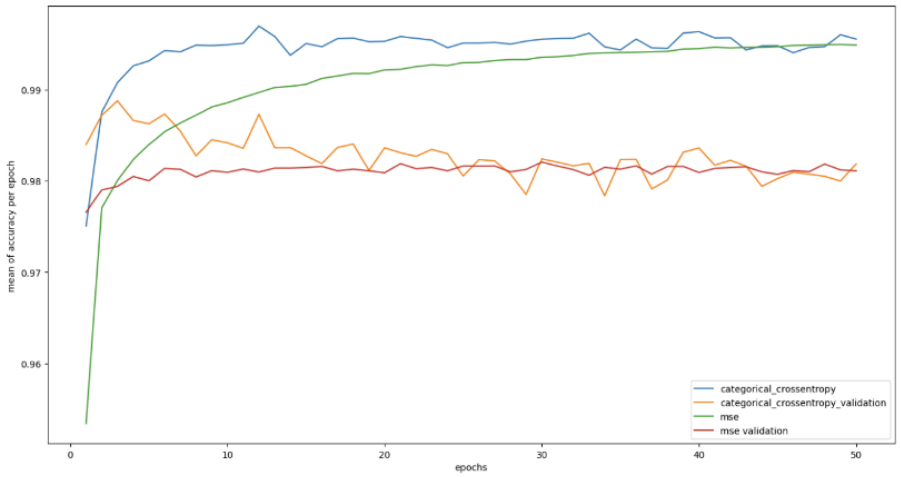
Δεδομένου ότι το overfitting δεν είναι μεγάλο πρόβλημα στη δικιά μας περίπτωση (σύμφωνα με τα προηγούμενα πειράματα), το περιθώριο για βελτιώσεις είναι παρα πολύ μικρό και επομένως το απογοητευτικά αποτελέσματα από την παραπάνω μέθοδο ομαλοποίησης με τους μεγάλους συντελεστές φθοράς ήταν αναμενόμενα.

Για να γίνουμε πιο συγκεκριμένοι , το training loss είναι δικαιολογημένα μεγαλύτερο καθώς η ομαλοποίηση προσθέτει περιορισμούς στο μοντέλο που θα προκαλέσουν αύξηση του σφάλματος στο training. Αυτό βέβαια με αντάλλαγμα να έχουμε μειωμένο σφάλμα στα δεδομένα του τεστ , αφού θα έχουμε πετύχει υποθετικά καλύτερο training. Κάτι τέτοιο βέβαια δεν συνέβη αφού και το validation loss είναι ήταν εξίσου μεγάλο. Αυτό είναι σημάδι υπερομαλοποίησης , καθώς το μοντέλο έχει περιοριστεί τόσο πολύ που δεν μπορέι πλέον να προσαρμοστεί τα δεδομένα εισόδου.

Τέλος σε ένα πείραμα που πραγματοποιήθηκε με συντελεστή φθοράς 0.0001 , το νευρωνικό τα πήγε πολύ καλύτερα. Παρόλο που το validation loss ήταν ακόμη υψηλό σε σχέση με τα πειράματα χωρίς ομαλοποίηση , το validation accuracy ηταν από τα πιο ψηλά μεταξύ όλων των πειραμάτων από όλα τα ερωτήματα.



Σχήμα 16: CE/MSE train-loss & validation-loss για r = 0.0001



Σχήμα 17: CE/MSE train-accuracy & validation-accuracy για r = 0.0001

3. A5. Convolutional Neural Network

α΄. Προετοιμασία δεδομένων

Το input shape στη περίπτωση των CNN πρέπει να είναι συγκεκριμένο και έχει διαφορετική μορφή από αυτή που χρησιμοποιήθηκε στη περίπτωση του απλού νευρωνικού δικτύου. Το Keras χρειάζεται σαν είσοδο ένα τετραδιάστατο πίνακα ενώ εμείς έχουμε 3 διάστασεις στα δεδομένα μας (batch , height , width) . Ετσι με την εντολή reshape προσθέτουμε την ιδιότητα "channels" που δηλώνει το grayscale των εικόνων. Ακόμα οσο αναφορά το preprocessing, κανονικοποιούμε τις τιμές των πίξελ και μετατρέπουμε τα labels σε one hot διανύσματα.

β΄. Περιγραφή αρχιτεκτονικής των μοντέλων

Το πρώτο layer του μοντέλου θα είναι το συνελικτικό Convo2d layer, το οποίο αποτελείται από μαθησιακά φίλτρα. Σε αυτό το layer ουαστικά γίνετε μια συνέλιξη ανάμεσα στο μητρώο της εικόνας και στο μητρώο του φίλτρου. Το αποτέλεσμα της πράξης είναι ένα χαρακτηριστικό της εικόνας.

Το δεύτερο layer του CNN μοντέλου θα είναι ένα pooling layer το MaxPool2D layer. Η δουλειά αυτού του layer είναι να μειώνει να μειώνει το μέγεθος των χαρακτηριστικών που παράχθηκαν από το συνελικτικο layer. Κοιτάει σε δυο γειτονικά πίξελ και θα διαλέγει τη μεγαλύτερη τιμή. Αυτά τα layers χρησιμοποιούνται για τη μειώση του υπολογιστικού κόστους και μέχρι ένα βαθμό για τη μείωση του overfitting.

Συνδυάζοντας συνελικτικά layers και pooling layers το νευρωνικό δημιουργεί τοπικά χαρακτηριστικά σε μία εικόνα.

Έπειτα χρησιμοποιούμε ένα flatten layer ώστε να μετατρέψουμε τα μητρώα με τα χαρακτηριστικά των εικόνων σε ένα μοναδιάστατο διανύσματα ώστε να μπορεί να χρησιμοποιηθεί από τα dense layers.

Στη συνέχεια , όπως και στη περίπτωση ενός απλού νευρωνικού δικτύου, αφου γνωρίζουμε ότι έχουμε να κάνουμε με ένα πρόβλημα ταξινόμησης θα χρειαστούμε ένα layer εξόδου με 10 κόμβους ώστε να προβλέψει τη πιθανότητα μια εικόνα να ανήκει σε μία από τις δέκα κλάσεις , σε συνδυασμό με τη συναρτήση ενεργοποίησης softmax. Πριν από αυτό το layer θα υπάρχει

βέβαια και ένα dense layer με 100 κόμβους ώστε να ερμηνέυσει τα χαρακτηριστικά των εικόνων.

Τέλος σαν optimizer θα χρησιμοποιήσουμε τον sgd με τις προκαθαρομένες παραμέτρους.

Στη δεύτερη αρχιτεκτονική θα αυξήσουμε το βάθος του ΄ΝΝ μοντέλου προσθέντωντας επιπλέον συνελικτικα και ποολινγ λαψερς με το ίδιο μέγεθος φίλτρων, για να δούμε αν μπορούμε να βελτιώσουμε περαιτέρω την αποτελεσματικότητα του νευρωνικού δικτύου.Ετσι προσθέτουμε ακόμα δυό συνελικτικά λαψερς και ακόμη ένα ποολινγ λαψερ με το ίδιο μέγεθος φίλτρου.

γ΄. Αποτελέσματα

Από τα αποτελεσματα βλέπουμε ότι σε γενικές γραμμές το μοντελο έχει κάνει πολύ καλο fit στα δεδομένα και δεν υπάρχουν σημάδια underfitting η overfitting.

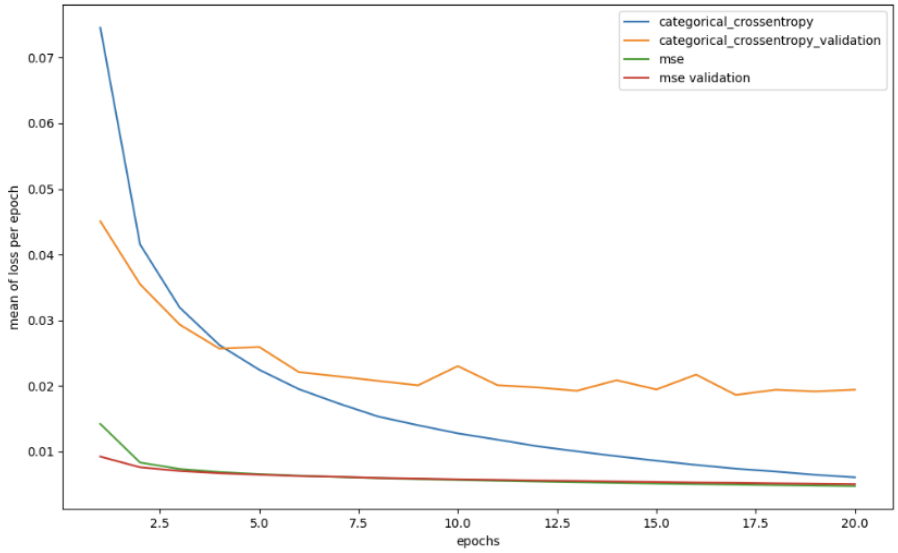
Από το διαγραμμα accuracy-epoch βλέπουμε ότι το validation-accuracy είναι πολύ κοντά στο train-accuracy πραγμα που σημαίνει ότι το μοντελο έχει πετύχει πολύ καλο train.

Τελος από το evaluate που έγινε στα test data set το μοντέλο στο καλυτερο fold πέτυχε accuracy 98.650% , το οποίο δείχνει οτι το μοντέλο τα πήγε εξίσου καλά σε δεδομένα τα οποία δεν έχει δει ποτέ.

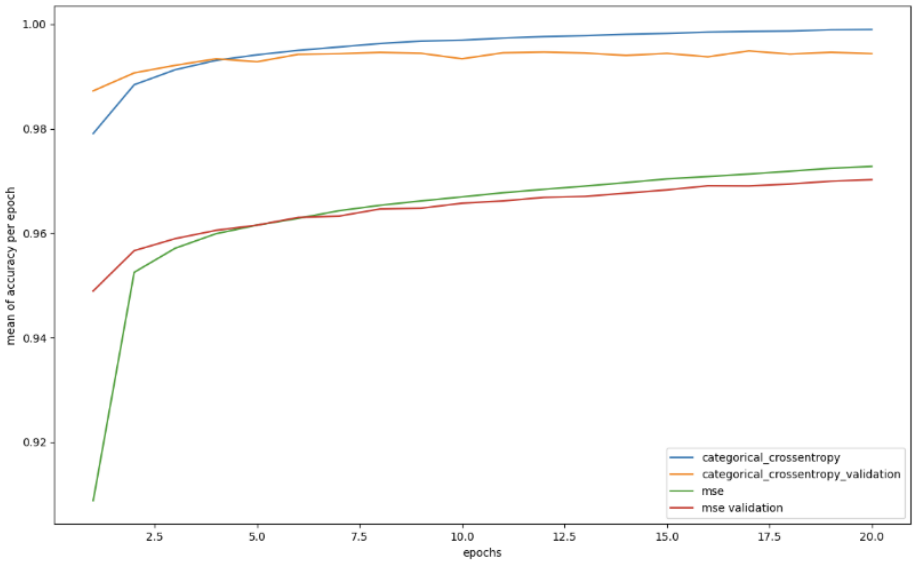
Χαρακτηριστικά CNN	CE validation loss	MSE validation loss
Αρχιτεκτονική 1	0.019	0.005
Αρχιτεκτονική 2	0.013	0.002

Χαρακτηριστικά CNN	CE validation accuracy	CE evaluate accuracy
Αρχιτεκτονική 1	99.43%	98.640%
Αρχιτεκτονική 2	99.67%	99.2%

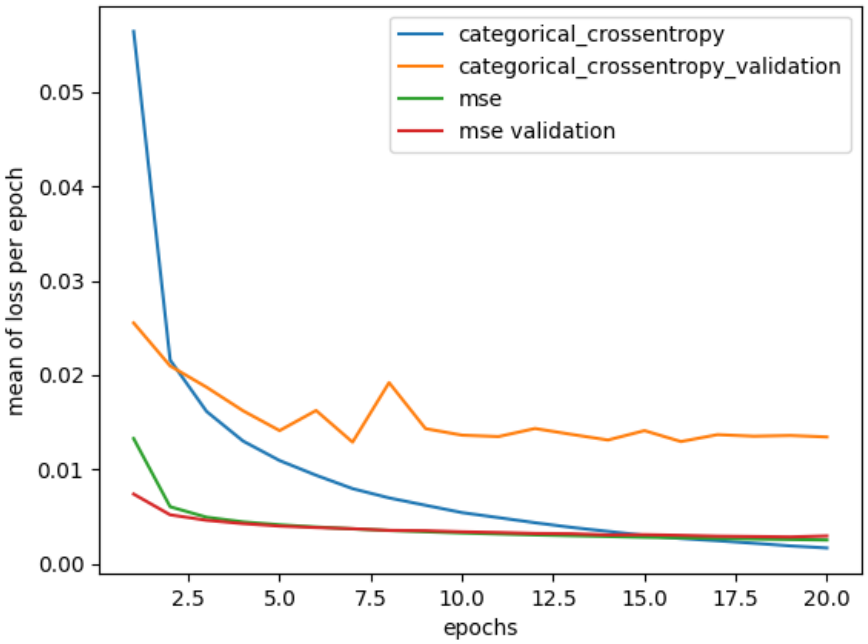
Από τα αποτελέσματα της δεύτερης αρχιτεκτονικής βλέπουμε ότι η άυξηση του βάθος του νευρωνικού δικτύου βοήθησε στη περαίτερω βελτίωση όλων των αποτελεσμάτων. Το πιο σημαντικό απ όλα είναι το accuracy από το test data set το οποίο στο καλύτερο fold με categorical crossentropy έδωσε accuracy 99.2%.



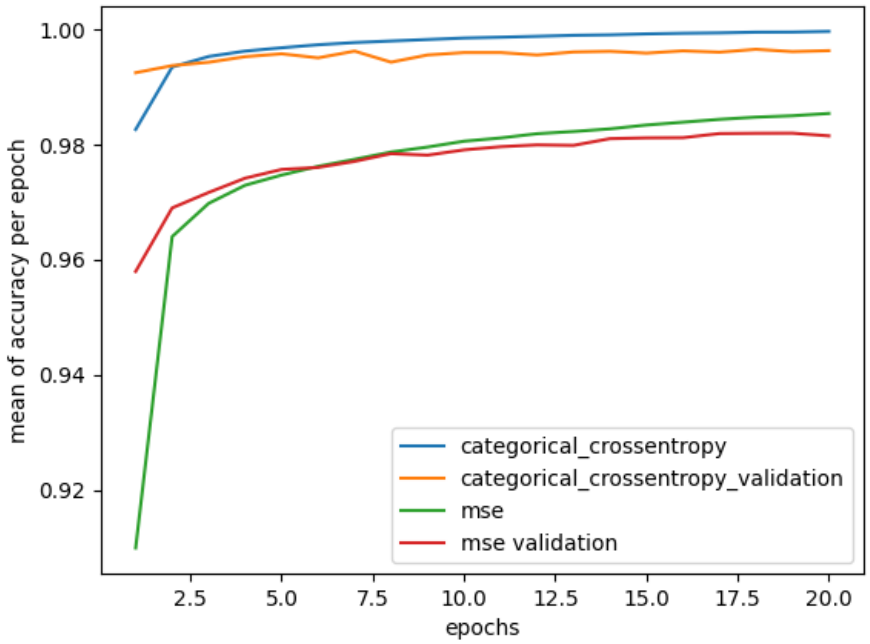
Σχήμα 18: CE/MSE train-loss & validation-loss για αρχιτεκτονική 1



Σχήμα 19: CE/MSE train-accuracy & validation-accuracy για αρχιτεκτονική 1



Σχήμα 20: CE/MSE train-loss & validation-loss για αρχιτεκτονική 2



Σχήμα 21: CE/MSE train-accuracy & validation-accuracy για αρχιτεκτονική 2