ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ ΙΙ
BIDIRECTIONAL STACKED RNN (LSTM/GRU)

Ονοματεπώνυμο: Βάιος Λύτρας

Γενικά:

- Οι πρώτες δοκιμές έγιναν σε διανύσματα που τους είχε εφαρμοστεί averaging όπως ακριβώς τα είχα χρησιμοποιήσει στην δεύτερη εργασία για να δω πως θα τα πήγαιναν σε σχέση με το custom input lengths αργότερα. Στο notebook υπάρχει ένα κελί που μετατρέπει το dataset σε 2-D πίνακα με averaging () και το από κάτω κελί είναι το κανονικό για τα custom input lengths.
- Όλες οι μετρήσεις που έκανα κατά την διάρκεια των πειραματισμών με τις παραμέτρους είναι γραμμένες στο excel αρχείο notes.xlsx που υπάρχει μέσα στο zip. Η τελεία '.' σημαίνει ότι η συγκεκριμένη παράμετρος δεν άλλαξε τιμή από το προηγούμενο πείραμα. Όταν κάτι αλλάζει σημειώνεται ξεκάθαρα στην αντίστοιχη στήλη. Αυτό αν θέλετε είναι προαιρετικό να το κοιτάξετε.
- Τα datasets έτσι όπως είναι αυτή τη στιγμή το notebook πρέπει να γίνουν upload σε ένα ειδικό φάκελο 'data' στο google collab για να τα φορτώσει. Διαφορετικά μπορείτε να συνδέσετε κάποιο drive ή να αλλάξετε το path.
- Το τελικό παραδοτέο περιλαμβάνει 2 διαφορετικά RNN που είναι το καλύτερο LSTM cell μοντέλο και το καλύτερο GRU cell μοντέλο μου.
- Ο κώδικας ήταν πολύ μεγάλος για να τον βάλω μέσα σε ένα κελί και το output θα ήταν ένα μπέρδεμα. Θα πρέπει να τρέξετε όλα τα κελιά διαφορετικά και με αυτόν τον τρόπο μπορείτε να διαλέξετε μεταξύ embedding averaging, custom input length.

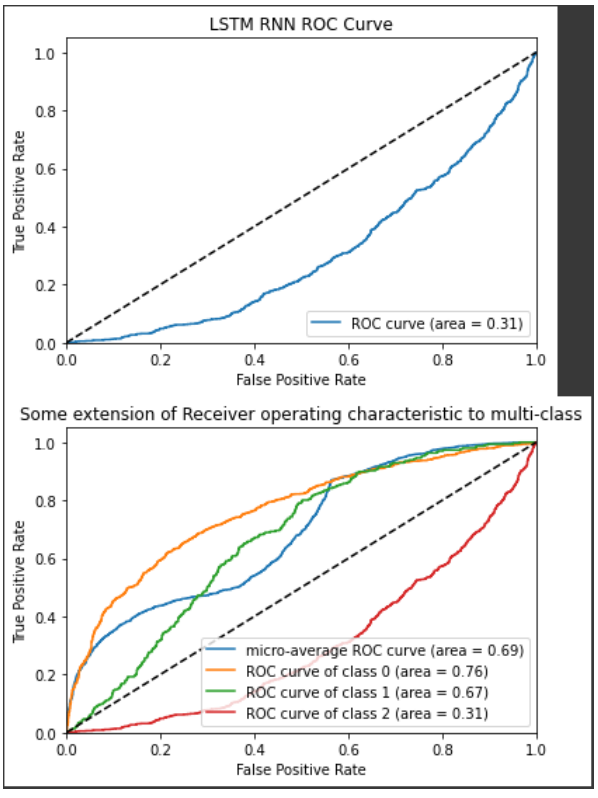
Πειραματισμός και βελτίωση των Hyperparameters

Dimension averaging:

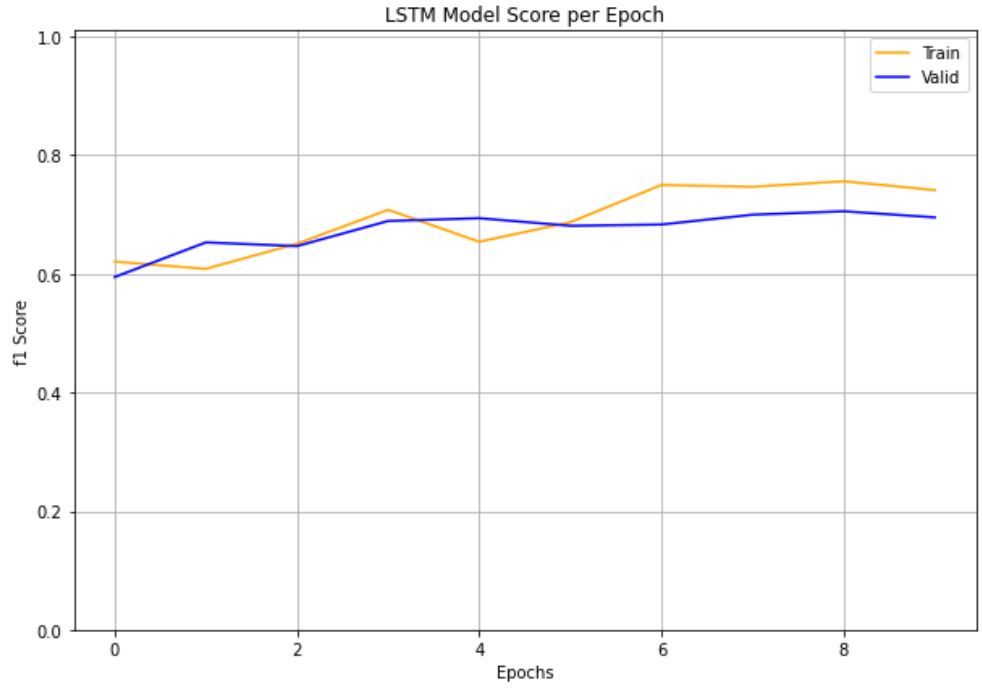
Ξεκινώντας με averaging στα διανύσματα όπως είπα και πολύ μέτριες τιμές σε όλες τις υπερπαραμέτρους δηλαδή 5 stacked layers, 20 hidden layers, 2 gradient clipping, 30% dropout, 350 batch size και 15 εποχές, το μοντέλα ξεκίνησαν με πολύ καλά f1 score δηλαδή 64% για το LSTM και 68.9% για το GRU αλλά με ένα μικρό overfit. Εδώ να σημειώσω ότι όλες οι μετρήσεις έγιναν πάνω σε LSTM και GRU RNNs ταυτόχρονα για τις ίδιες υπερπαραμέτρους. Μετά από λίγες δοκιμές παίζοντας με τον αριθμό των layers παρατήρησα ότι οι 15 εποχές ίσως τελικά να είναι πολλές για stacked RNN αφού μετά την $7^{η}$-$8^{η}$ ξεκινάει το overfit (δηλαδή παρατήρησα το test loss ανεβαίνει και το score μένει ίδιο).

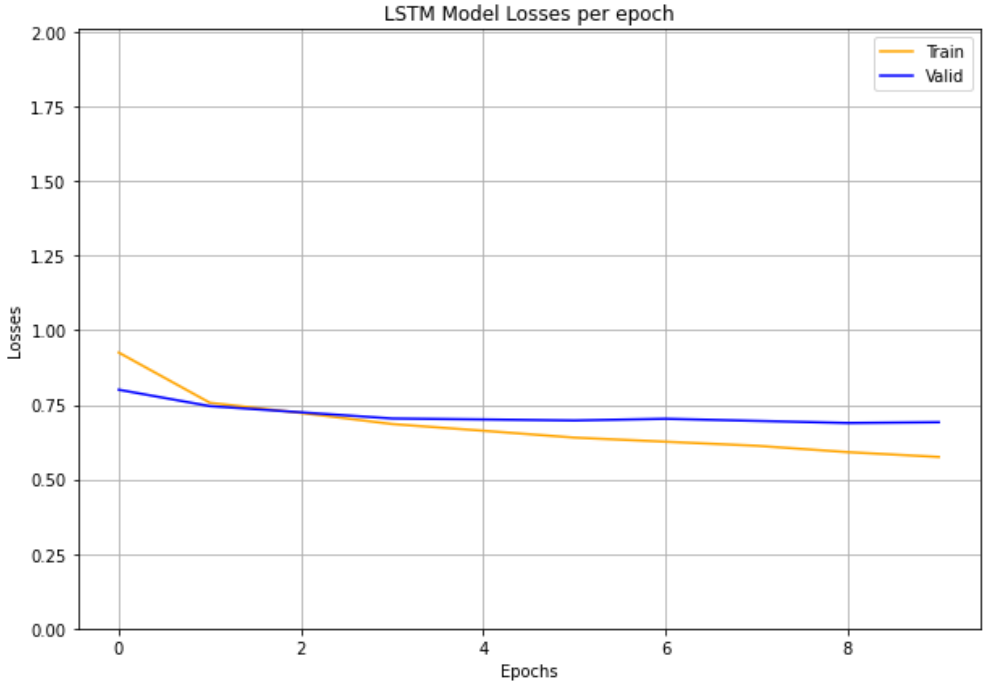
Η πρώτη αξιόλογη δυάδα παρατηρήθηκε λίγο μετά που έριξα τον αριθμό των layers στα 4, τα hidden 25, ανέβασα το gradient clipping στο 3 και έριξα το learning rate στο 5e-4. Το LSTM αυτό είχε 62% f1 και 66% το GRU αλλά δεν παρουσίαζαν overfitting.

Στη συνέχεια ανέβασα τα stacked layers σε 10 (το οποίο το δοκίμασα άλλες 2 φορές κατά την διάρκεια του project) και όλες τις φορές το μοντέλο δεν εκπαιδευόταν και ήταν ακόμα χειρότερο και από το τυχαίο μοντέλο (βλ. εικόνα). Προφανώς ένα RNN μοντέλο δεν χρειάζεται να είναι πάντα σουπερ βαθύ και περίπλοκο πρέπει απλά να βρίσκουμε το καλύτερο σημείο των παραμέτρων μας που να ταιριάζει καλύτερα στο πρόβλημά μας.

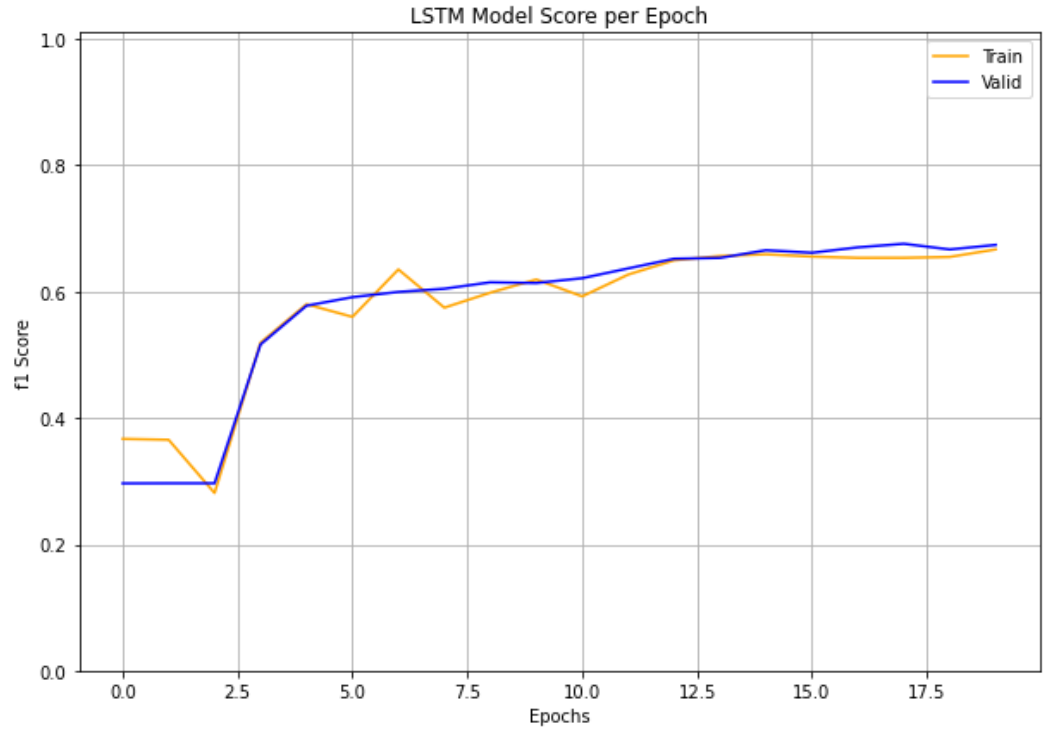


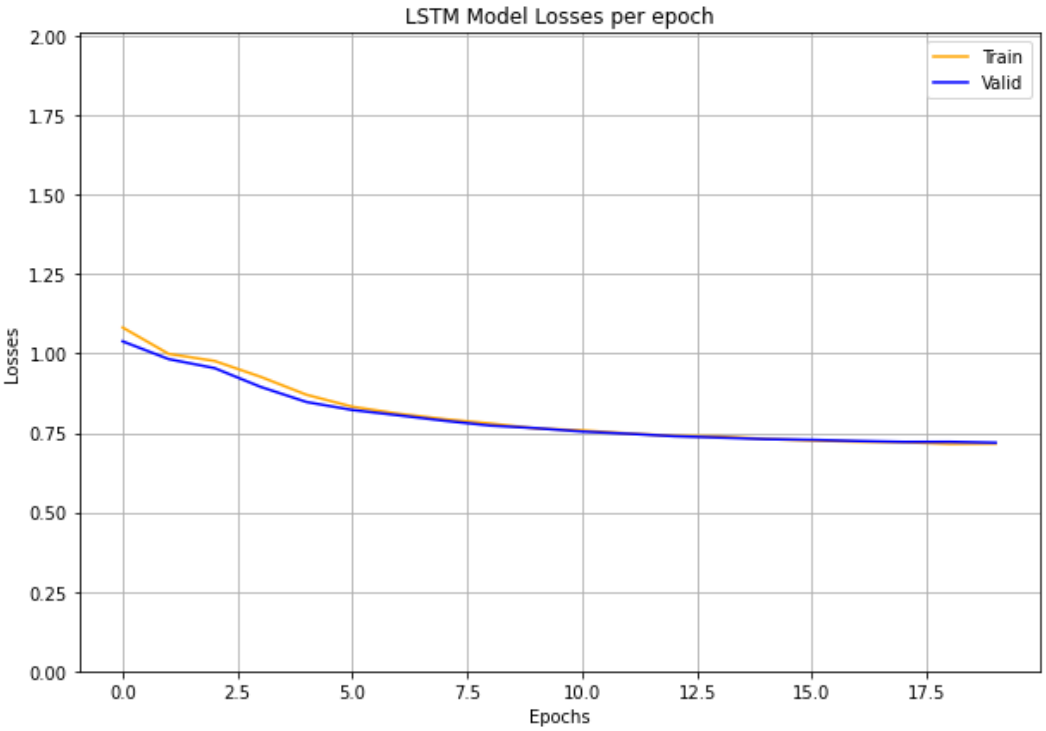
Δοκίμασα να κάνω το ίδιο αυξάνοντας αρκετά τα hidden layers (50) και συνέβει κάτι αντίστοιχο δηλαδή υπήρχε πολύ αυξημένο overfitting (βλ. εικόνα). Από την πρώτη κιόλας εποχή το μοντέλο ξεκίνησε να ανεβάζει loss και το score του μία ανέβαινε μία κατέβαινε.





Στην συνέχεια, αφαιρόντας ένα stacked layer και μειώνοντας το learning rate στο μισό το LSTM είχε 68,9% score και το GRU 69.3% χωρίς overfitting και οι καμπύλες τους ήταν οι εξής (περίπου ίδιες και για το GRU) :

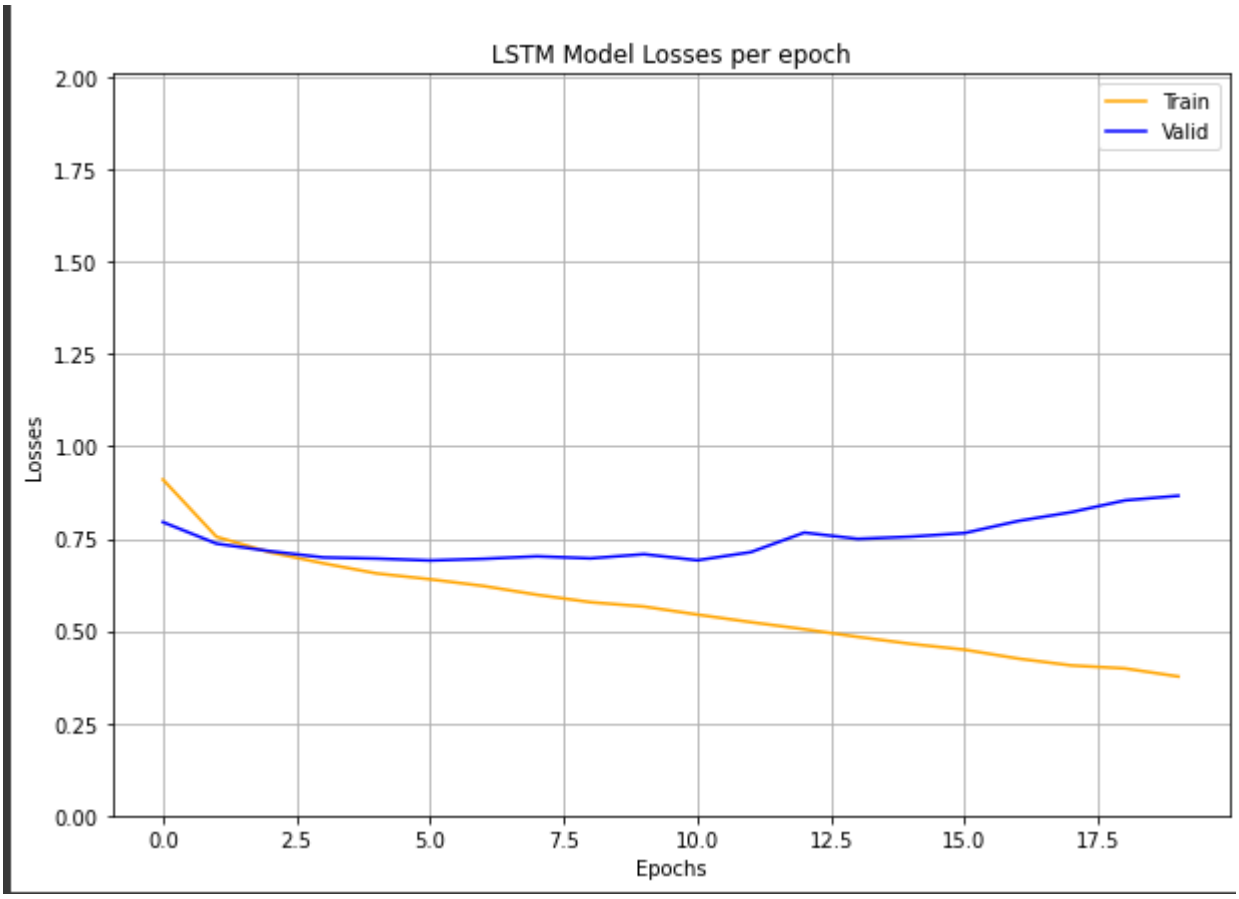




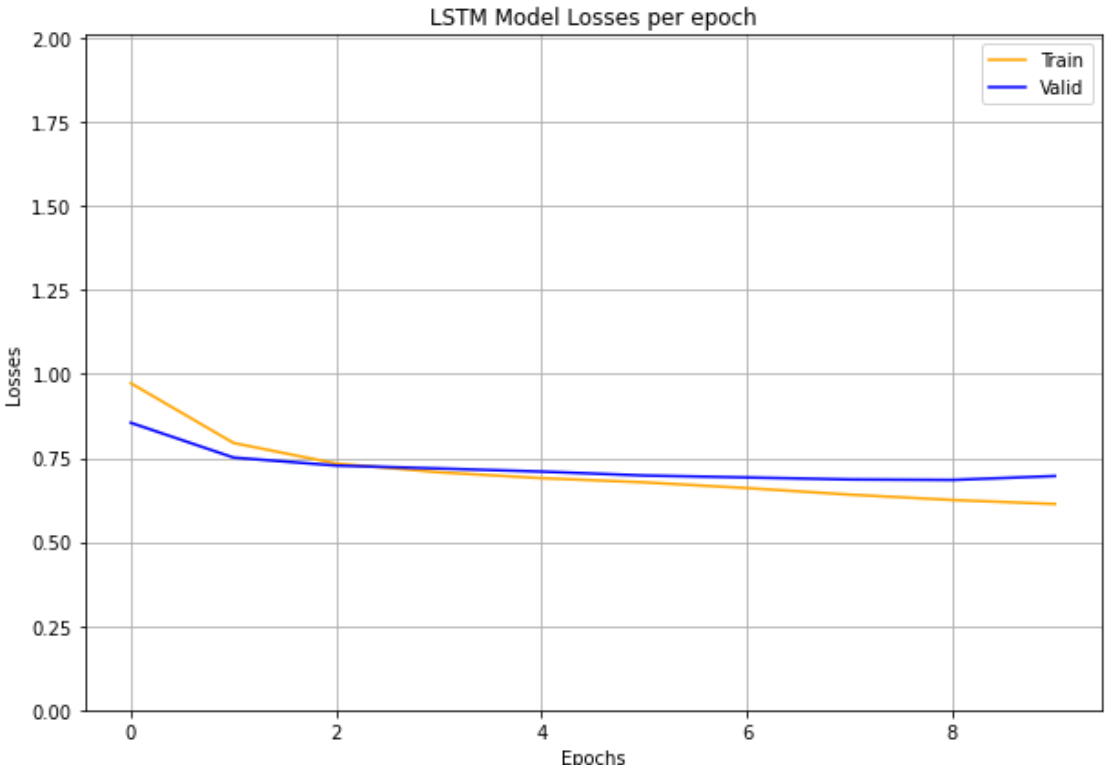
Μετά δοκίμασα να ανεβάσω το dropout στο 50% που δεν προκάλεσε overfit όπως και περιμέναμε αλλά έριξε κατά περίπου 5% τα προηγούμενα score και λογικό αφού στο τέλος ‘’πετούσε’’ τα μισά δεδομένα.

Η εκπαίδευση σε μεγάλα batches επίσης δεν βοήθησε κάπου συνέβει κάτι αντίστοιχο με το μεγάλο dropout chance και επίσης το loss έμενε σχετικά ψηλά κατά την εκπαίδευση. Για το batch size έγιναν πολλές δοκιμές και το ιδανικό συμπέρανα ότι είναι κάπου μεταξύ 200 και 400 για το vaccine sentiment.

Τέλος (μετά περνάμε στα κανονικά RNN με custom input length), δοκίμασα να μειώσω πολύ το learning rate (κοντά στο 0.01) και λογικό είναι τα μοντέλα να ήταν 10 φορές χειρότερα:



Σημείωση: Όπου αναφέρεται για μικρό overfitting παραπάνω ή παρακάτω εννοώ κάτι τέτοιο:



Custom input length:

Έχοντας συμπεράνει όλα τα παραπάνω για τις υπερπαραμέτρους, το καλύτερο GRU cell μοντέλο ήρθε κιόλας από την αρχή αρχή με το custom input length το οποίο είχε f1 score λίγο πάνω από 70%.

Οι δοκιμές με πολλά stacked layers ή hidden layers προκάλεσαν το ίδιο πρόβλημα με πριν.

Σε αυτό το σημείο έγιναν αρκετά πειράματα κάνοντας tune το gradient clipping και αυτό που παρατηρήθηκε είναι ότι το μεγάλο gradient clipping βοηθάει τα μοντέλα με GRU cells (ανεβάζει το σκορ τους) ενώ ρίχνει το score σε αυτά με LSTM. Για παράδειγμα: αν σε ένα καλό σετ υπερπαραμέτρων θέσουμε το gradient clipping=6 που είναι υπερβολικά πολύ τότε το score στα LSTM πέφτει περίπου 6% και στα GRU πέφτει περίπου 1%. Για φυσιολογικές τιμές στα LSTM είναι καλό το gradient clipping στο 2-3 για τα GRU στο 4.

Επίσης σε αυτό το μέρος επειδή οι πίνακες των δεδομένων φτιάχνονται με sequence length κλπ και είναι overall «βαρύτεροι» και περιέχουν περισσότερη πληροφορία, τα μοντέλα εκπαιδεύονταν μέσα σε το πολύ 5 εποχές και σε μερικές περιπτώσεις χρειάζονταν ακόμα λιγότερες.

Η υπόθεση για το μέγεθος του batch size δοκιμάστηκε και σε αυτό το μέρος και το συμπέρασμα για το εύρος του καλού batch size φάνηκε να είναι το ίδιο με πριν. Αυτή τη φορά δοκιμάστηκε και batch size < 100 το οποίο δεν βοήθησε και προκάλεσε μεγάλο overfitting όπως είναι και λογικό.

Μία πολύ καλή διαπίστωση που έγινε επίσης είναι για το Larning Rate. Τα GRU cells χρειάζονται σχετικά μεγαλύτερο learning rate από τα LSTM και κάνουν train σε λιγότερες εποχές κατά μέσο όρο. Για την ακρίβεια τα GRU δουλεύουν καλύτερα σε LR της τάξης των τριών δεκαδικών ψηφίων ενώ τα LSTM των τεσσάρων δεκαδικών.

Τέλος, με αυτή τη μέθοδο του custom input length και sequence length που έχουμε σχετικά μικρότερη απώλεια πληροφορίας και μικρότερο θορύβου από το averaged dataset, δεν παρατηρήθηκαν πολλά φαινόμενα overfitting (τα μισά πειράματα δεν είχαν overfit) και δεν παρατηρήθηκαν καθόλου φαινόμενα μεγάλου overfitting (όλα αυτά είναι σημειωμένα στο notes.xlsx).

Τελικές μετρήσεις και καμπύλες των καλύτερων LSTM και GRU μοντέλων

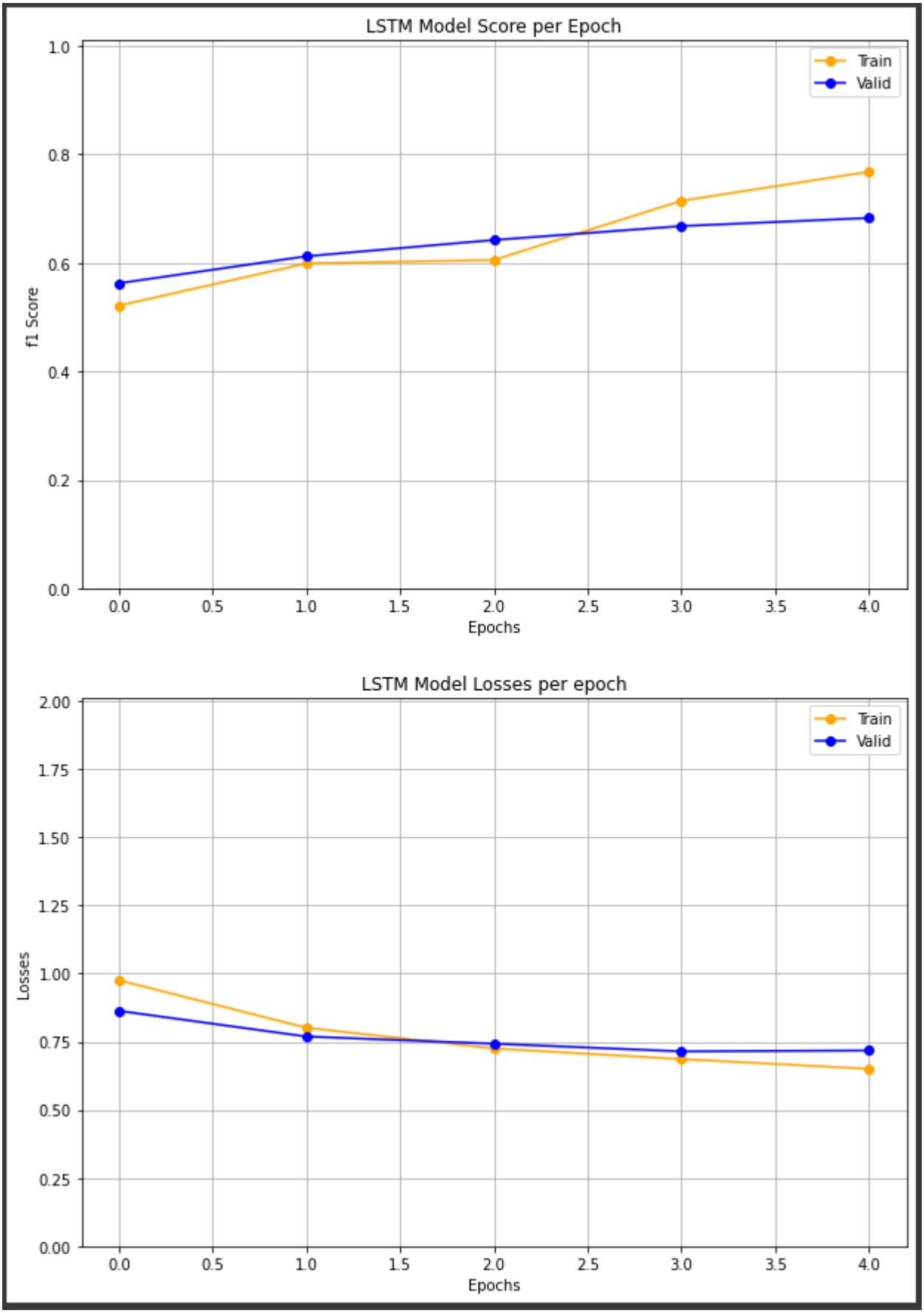
LSTM: 3 layers, 50 hidden layers, 4 gradient clipping, 25% dropout, 0.0008 LR, 400 batch size, 5 epochs

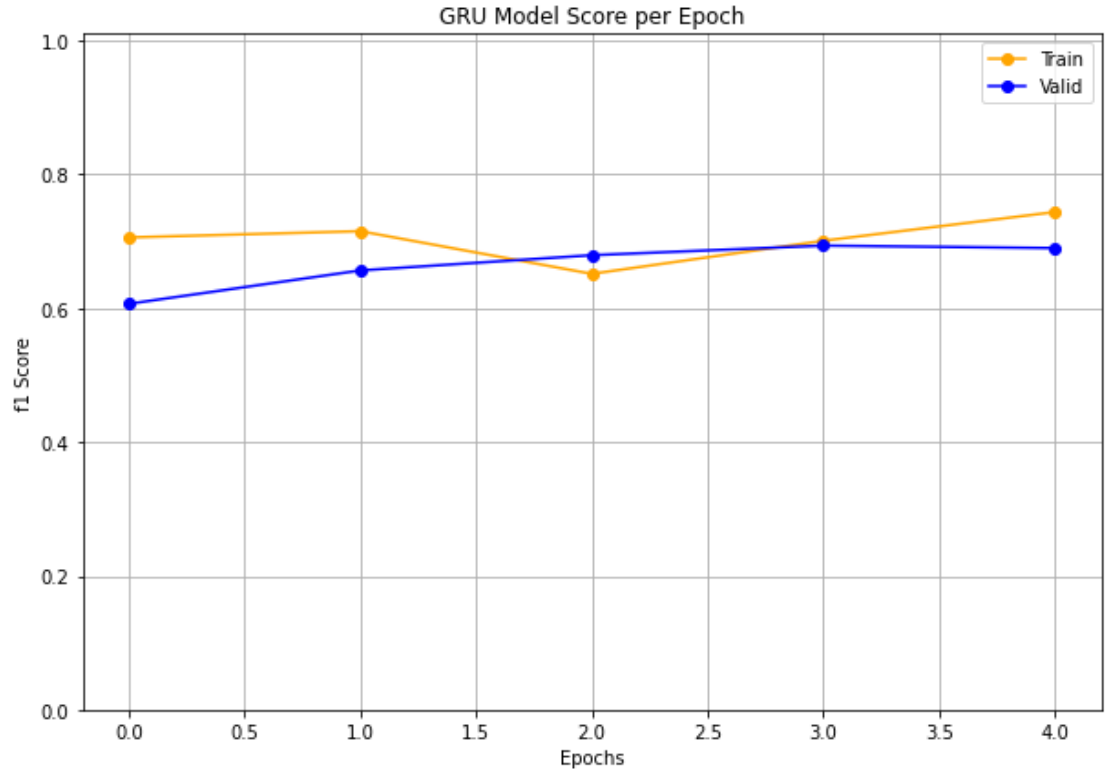
GRU: 4 layers, 30 hidden layers, 4 gradient clipping, 30% dropout, 0.001 LR, 300 batch size, 5 epochs

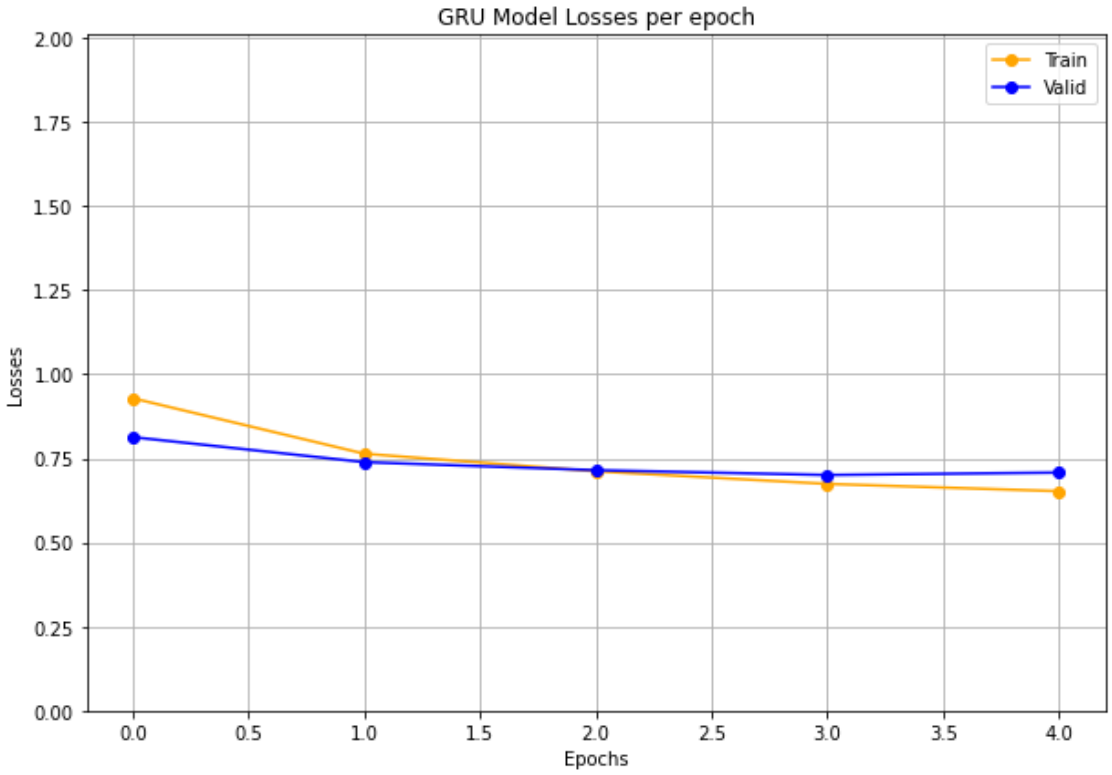
```
LSTM Model scores:
Test Accuracy Score: 69.28133216476775
Test F1 Score: 68.29802909131676
Test Recall Score: 69.28133216476775
Test Precision Score: 70.04804197040751
Test Loss: 0.7177825570106506

---------------------------------------------------------------------

GRU Model scores:
Test Accuracy Score: 68.58019281332164
Test F1 Score: 69.00472689105503
Test Recall Score: 68.58019281332164
Test Precision Score: 69.9683441122359
Test Loss: 0.7074985504150391
```


LSTM Model Score per Epoch
Train
Valid
f1 Score
Epochs
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
LSTM Model Losses per epoch
Train
Valid
Losses
Epochs
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

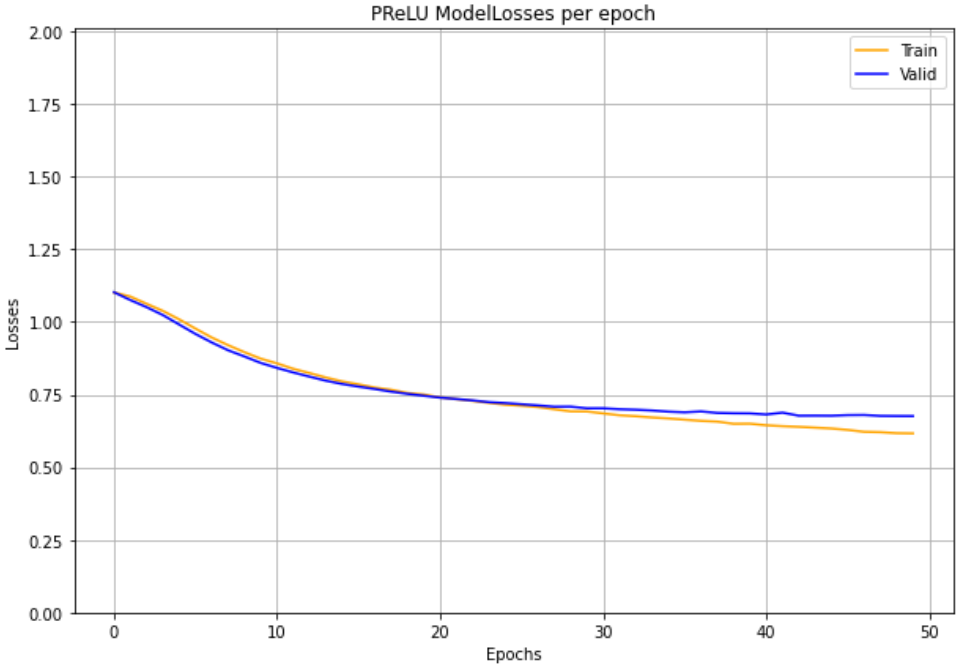




Σημείωση: Τα ROC curves μπορείτε να τα δείτε στο notebook. Είναι σχεδόν ίδια με αυτά του προηγούμενου project σε σημείο που δεν μπορεί να παρατηρηθεί κάτι.

Σύγκριση με το Feed-Forward NN του προηγούμενου Project

Τα δύο μοντέλα είναι αρκετά παρεμφερή στο θέμα των καμπύλων και του loss που έχουν. Ας θυμηθούμε την καμπύλη του PReLU μοντέλου:



Οι αρχικές και τελικές τιμές είναι σχεδόν ίδιες με το Feed-Forward NN να έχει λίγο πιο ομαλές καμπύλες λόγω των περισσότερων εποχών. Όσο για το f1 score προς έκπληξή μου το καλύτερο μοντέλο του προηγούμενου project έχει 2% παραπάνω. 0.8% παραπάνω accuracy και 1.2% παραπάνω precision. Τελικά, μπορούμε να συμπεράνουμε ότι ίσως οι απώλειες που δημιουργούνται στο dataset λόγω του padding ή clipping των προτάσεων είναι αντίστοιχο με τις απώλειες που είχαμε στο προηγούμενο project κάνουμε averaging.

Ωστόσο το σίγουρο είναι ότι τα RNN είναι πολύ πιο περίπλοκα από τα FF-NN λόγω των stacked και hidden layers και φαίνεται και στο training και την ώρα που χρειάζεται για κάθε εποχή το καθένα. Επίσης τα LSTM (ή GRU) RNN λόγω του memory χρειάζονται εκθετικά λιγότερες εποχές από τα feed-forward για να εκπαιδευτούν.

Τελικό πόρισμα:

Μετά από την μελέτη κάποιων papers έχω την εντύπωση ότι το δικό μας dataset είναι πολύ μικρό για πολύ περίπλοκα μοντέλα όπως τα LSTM-RNN, GRU-RNN τα οποία είναι stacked και bidirectional. Επίσης μετά από εμφάνιση στατιστικών στο dataset παρατηρήθηκε μεγάλο imbalance ως προς την κλάση 1 οπότε ίσως και αυτό να είναι υπαίτιο στο ότι κανένα μοντέλο δεν μπορεί να πιάσει score τάξεων 80%-95%.