

Step 1: Preprocessing

Αρχικά έχοντας διαβάσει τα datasets κάνω μια διαδικασία καθαρισμού των tweet , αφαιρώ δηλαδή ειδικούς χαρακτήρες , links tags και mentions τα οποία δεν βοηθούν το νευρωνικό. Στην συνέχεια κάνω tokenize με την βοήθεια του Torchtext όπου χρησιμοποιώ ως tokenizer τον Spacy αφού φαίνεται να δίνει μέσω πειραμάτων ελαφρώς καλύτερα αποτελέσματα από τον word_tokenize που χρησιμοποιούσα στην δεύτερη εργασία. Έχοντας λοιπόν φτιάξει ένα text_field και ένα label_field , χρησιμοποιώ την συνάρτηση TabularDataset.splits όπου δίνοντας τα δυο csv αρχεία επιστρέφει το train και val set με τα δεδομένα tokenized όπως όρισα προηγουμένως.

Χρησιμοποιώντας τα pretrained του glove (100d) κατασκευάζω τα λεξικά που χρειάζομαι για τα embeddings και τέλος φτιάχνω iterators για το training και validation μέσω της συνάρτησης BucketIterator.splits .

Στο πρώτο αυτό step όσο αναφορά τον πειραματισμό πέρα από τους tokenizers που αναφέρθηκαν πολύ σημαντικό

ήταν το batch_sizes όρισμα της BucketIterator.splits στο οποίο δοκίμασα τιμές από 16 μέχρι και 2048 αλλά κατέληξα σε 32 για το train και 128 για το val.

Step 2: Bidirectional stacked RNN with LSTM/GRU model.

Στο σημείο αυτό υπήρξε αρκετός πειραματισμός και ήταν με τις εξής υπερπαραμετρους :

- 1) Τα αποτελέσματα μου ήταν καλύτερα με όσο δυνατόν λιγότερη πολυπλοκότητα για τον λόγω αυτό επέλεξα 3 layers embedding-LSTM/GRU-linear και ανάμεσα dropout.
- 2) Dropout, οι τιμές από 0.40 - 0.60 φαίνεται να δουλεύουν καλύτερα μειώνοντας το overfit , μετά από πειραματισμούς κατέληξα στο 0.45
- 3) Number of recurrent layers, δοκίμασα από 2-5 και τα καλύτερα αποτελέσματα έδωσε το 2 λόγω μικρότερης πολυπλοκότητας καθώς τα δεδομένα που δόθηκαν είναι σχετικά λίγα για LSTM/GRU.
- 4) The number of features in the hidden state, επίσης πολλές δοκιμές η καλύτερη με 256
- 5) To learning rate στον optimizer δοκιμαστήκαν τιμές 0.0001 - 0,001 με την καλύτερη να είναι 0.0003
- 6) Ο optimizer Adam και το CrossEntropyLoss χρησιμοποιήθηκαν χωρίς άλλα πειράματα καθώς έτσι ζητήθηκε.
- 7) Skip connections , δεν κατάφερα κάποια βελτίωση για τον λόγω αυτό και δεν τα έβαλα στο τελικό μοντέλο
- 8) Attention δεν δοκιμάστηκε καθόλου.

Step 3: Training and evaluation.

Εδώ έγινε πειραματισμός με :

- 1) Gradient clipping , δοκιμάστηκε από 2-5 με το καλύτερο αποτέλεσμα να δίνει η τιμή 2
- 2) Εποχές, πάνω από τις 10 εποχές δεν υπήρχε νόημα καθώς παρατηρείτε overfit

Step 4: Roc curve.

Στην καμπύλη αυτή μπορεί εύκολα κάποιος να συμπεράνει την ανωτερότητα του μοντέλου αυτού σε σχέση

με της προηγούμενης εργασίας , καθώς φαίνεται σημαντική διάφορα όχι μόνο στην κλάση με τα θετικά

tweet αλλά κυρίως σε αυτή των αρνητικών. Εάν και πάλι είναι η κλάση με το χαμηλότερο score τα αποτελέσματα

είναι πολύ καλύτερα από αυτά του προηγούμενου γραμμικού μοντέλου.

Step 5: Σύγκριση μοντέλων και συμπεράσματα.

Αρχικά είναι φανερό από τα classification reports ότι το μοντέλο της τρίτης εργασίας είναι πολύ καλύτερο από των άλλων 2 εργασιών . Οσοσω όσο αναφορά τα αποτελέσματα μεταξύ GRU και LSTM δεν υπάρχουν σημαντικές διαφορές, για τον λόγο αυτό δόθηκε το μοντέλο LSTM καθώς έχει γίνει περισσότερος πειραματισμός πάνω σε αυτο.Αν παρατηρήθηκε κάτι στην μικρή διαφορά τους είναι ότι το μοντέλο με LSTM είχε πιο σταθερά αποτελέσματα αλλά το μοντέλο GRU είχε το υψηλότερο score .Το τελευταίο μάλλον είναι λόγω της απλότητας του προβλήματος καθώς ο LSTM είναι πιο περίπλοκος. Τέλος ένα από τα προβλήματα της τρίτης εργασίας ήταν το overfit όπου όπως φαίνεται από τα learning curves δεν κατάφερα να εξαλείψω εντελώς.