Angelos Kamaris sdi1900070
AI 2 - project 2

January 27, 2023

Καμάρης Άγγελος sdi1900070

1 Πηγές

Αξιοποίησα τον κώδικα του φροντιστηρίου κατά κύριο λόγο, χρησιμοποιώντας επίσης εντολές από το sklearn, καθώς επίσης αξιοποίησα σαν σκελετό την προηγούμενη εργασία μου, κρατόντας το preprocessing των δεδομένων μου καθώς και τον χωρισμό των δεδομένων, την εισαγωγή τεστ και την εμφάνιση αποτελεσμάτων και το glove .

2 Επεξήγηση Κώδικα

Χρησιμοποίησα τις στήλες rating και review σαν Υ και X αντίστοιχα, όπου επεξεργάστηκα τα rating, έτσι ώστε τα "κακά" (0-4) να έχουν την τιμή 0 και τα "καλά" (7-10) να έχουν την τιμή 1.

Επεξεργάστηκα τα δεδομένα από τα reviews κάνοντας τα κεφαλαία μικρά, αυτή την φορά κράτησα τα σημεία στίξης, έβγαλα τις λέξεις που είχαν λιγότερα από 2 γράμματα, βρίσκοντας τις πιο πολυχρησιμοποιημένες λέξεις και αφαιρώντας τις πιο σπάνιες, και τέλος έκανα lematize , χωρίς stematize φυσικά για να αναγνωρίζονται οι λέξεις.

Χώρισα τα δεδομένα μου έτσι ώστε 30% εξ αυτών να γίνονται validate και τα υπόλοιπα να χρησιμοποιούνται για το training και για την εισαγωγή των δεδομένων στο νευρονικό δίκτυο, τα πέρασα από το glove2word2vec βάζοντας σαν input το glove.6B.300d.txt καθώς με αυτό έχω την μεγαλύτερη ακρίβεια (από 76% σε 84%).

Δοκίμασα και LSTM καθώς και GRU και αν και γενικά τα αποτελέσματα ήταν αρκετά κοντά, προτίμησα το δεύτερο, παρόλου που θέλει περισσότερο χρονο. Του δίνω input size = 300, hidden size = 300, layers size = 1, output size = 2, clip=10 και p=0.5, καθώς αυτά μου έδωσαν τα καλύτερα αποτελέσματα ενώ για το LSTM

έχω : input size = 300, hidden size = 100, layers size = 1, output size = 2, clip=1000 και p=0.6, . Θα παραθέσω μέσα στον φάκελο και τους φακέλους που έχω τα αποτελέσματα από τις συγκρίσεις μου, για τις οποίες θα μιλήσω σύντομα.

Συγκεκριμένα, φορτώνω τα δεδομένα που ανέφερα πριν στον classifier μου, καλώ Linear με output = hidden size*3 καθώς δίνω 300 σαν hidden size . καλώ επίσης clip, για gradient clipping, Dropout για dropout probability καθώς επίσης χρησιμοποιώ σαν attention weight το $nn.Parameter(torch.randn(hiddensize * 2))$. στο forward καλώ το κελί που έχω ορίσει, εκτελώ dropout,mechanism of attention, dropout , με αυτή την σειρά (βρήκα τυχαία όταν είχα βάλει λάθος στο μοντέλο μου, ότι έτσι αποφεύγουμε περισσότερο το οερφιττινγ και το κράτησα), Linear και βγάζω έναν πίνακα. Ενώ στο backwards εκτελώ clip grad norm . Επειδή το αποτέλεσμα που θα μου δωθεί μετά το Linear έχει έχει διάσταση τον αριθμό των inputs επί 2, αφού τελειώσει το μοντέλο πρέπει να κρατήσουμε μόνο την μεγαλύτερη εκ των 2 τιμών.

Για την εισαγωγή ενός test, αρκεί να εισάγετε τα δεδομένα στην μεταβλητή test df , όπως το παράδειγμα στα σχόλια.

3 Παρατηρήσεις

Την μεγαλύτερη αλλαγή την παρατήρησα, στην χρήση layers, όταν το μοντέλο δεν έκανε overfit ή underfit καθώς παρατηρούνται αλλαγές της τάξης $0.02 - 0.04$, αναλόγως και τα υπόλοιπα δεδομένα, καθώς το μοντέλο μου δεν φαίνεται να επιθυμεί μεγάλο αριθμό, κάτι το οποίο θα το επιβράδυνε αν ήταν μεγαλύτερο του 1.

Το p φαίνεται να επηρεάζει σε μεγάλο βαθμό το overfitting , καθώς όσο μικρότερο είναι τόσο μεγαλύτερο overfitting θα κάνουμε αλλά για μεγαλύτερο, μικραίνει η ακρίβεια. Το hidden size φαίνεται να επηρεάζουν επίσης πολύ τα αποτελέσματα, αναλόγως το κελί που χρησιμοποιούμε, καθώς το LSTM προτιμά μικρό που το κάνει και ταχύτερο ενώ το GRU προτιμά μεγάλο. Τέλος το clip αν και μετά από ένα σημείο δεν υπάρχει μεγάλυ αλλαγή, είναι προτιμότερο να είναι σχετικά μεγάλος αριθμός.

Σε σχέση με τα μοντέλα από τις προηγούμενες εργασίες, αυτό το μοντέλο είναι πιο αποδοτικό, αν και θέλει πολύ περισσότερο χρόνο. Συγκεκριμένα, αποφεύγει περισσότερο το ωορφιττινγ, ενώ παράλληλα δίνει καλύτερα αποτελέσματα ως προς την ακρίβεια.

4 Παραδοχές

Έχω στείλει 2 αρχεία κώδικα, ένα για κάθε κελί, καθώς επίσης και 2 .txt αρχεία με τα αποτελέσματα από διάφορες δοκιμές που έκανα. Μέσα υπάρχει επίσης και το μοντέλο του lstm , καθώς και του gru , τα οποία τα κατέβασα όπως είχαν δείξει στο φροντιστήριο.

5 Αποτελέσματα

Τα αποτελέσματα που έχει το πρόγραμμά μου για τα δεδομένα που ανέφερα είναι:
LSTM
F1 Score (train): 0.8710447206123905 F1 Score (validation): 0.8549215406562054
Recall Score (train): 0.8953236493374108 Recall Score (validation): 0.8797709923664122
Precision Score (train): 0.8480477943395088 Precision Score (validation): 0.8314372918978913
GRU
F1 Score (train): 0.8789439793947199 F1 Score (validation): 0.8523263654753878
Recall Score (train): 0.8674928503336511 Recall Score (validation): 0.83980510851912
Precision Score (train): 0.8907014681892332 Precision Score (validation): 0.8652266504411318