YS19 - Artificial Intelligence II
Project 3

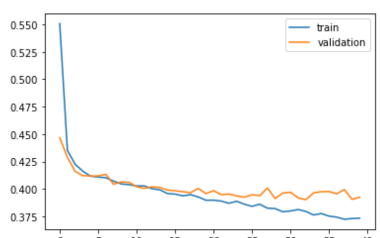
Efstathios Chatziloizos - 1115201800212

January 2023

A sentiment classifier using a bidirectional stacked RNN with LSTM/GRU cells for imdb movie reviews has been developed. The machine learning framework PyTorch was used. The neural network's input size is the same as the number of dimensions of the GloVe embeddings used. For the purposes of this assignment glove.twitter.27B.200d.txt was used, hence the input size has a value of 200.The neural network's output layer consists of one neuron, whose output is a probability p, where p $>$ 0.5 indicates a positive sentiment and p $<$ 0.5 indicates a negative sentiment.

Architecture

A wide range of architectures were considered, implemented and tested during this assignment. The most prominent ones, in accordance to the class's notes, consisted of 2 layers of LSTMs/GRUs. Architectures with 3 layers of RNNs did not seem to improve the results.

[2 layers] 

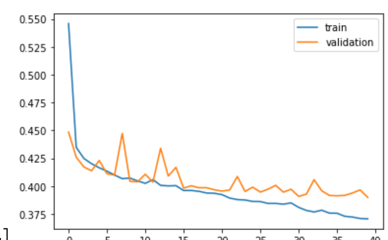
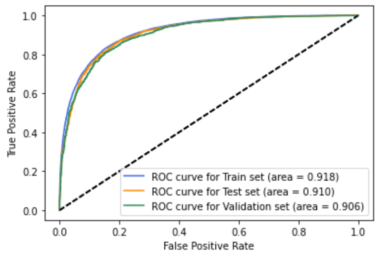
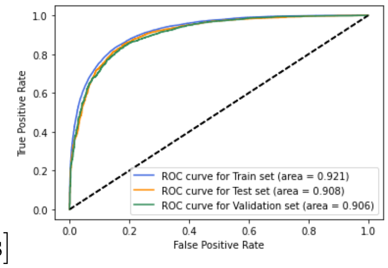
[3 layers] 

Figure 1: Learning curves of models with 2 and 3 RNN layers.



[2 layers]



[3 layers]

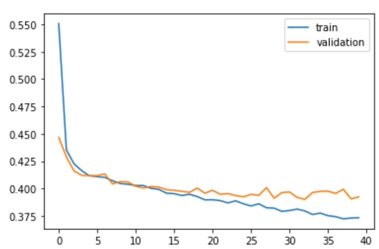
Figure 2: ROCs of models with 2 and 3 RNN layers.

Different hidden sizes were considered, namely a range of 50-500 units. The results were not much affected by this choice. 50 hidden units produced adequate results in much less time than 500 units required. The big problem with large hidden sizes was overfitting and fine tuning of the hyperparameters as well as normalization were crucial. A happy medium of 200 hidden units was chosen for the final model.

Bidirectional vs Unidirectional

Even though the scope of this assignment involved only Biderectional RNNs, some Unidirectional RNNs were also implemented. The differences between the two was minimal for this dataset. This can only mean that the model did not gain much by the wider range of context of the Bidirectional model. In antithesis, the model took more time to train due to the doubling of the total hidden units.

[Bi]



[uni]

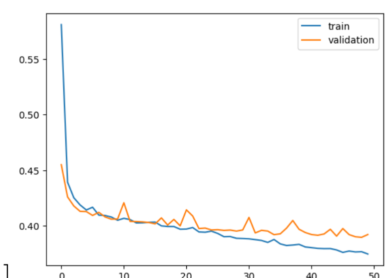
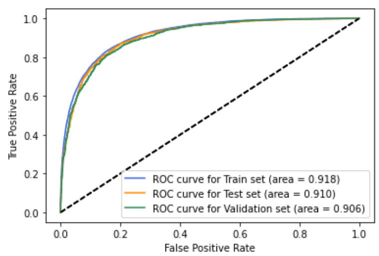


Figure 3: Learning curves of the best bidirectional and unidirectional models.

[Bi] 

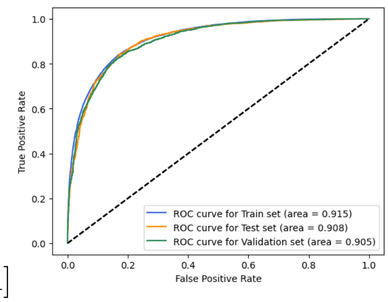
[uni] 

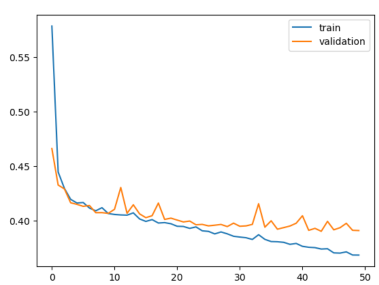
Figure 4: ROCs of the best bidirectional and unidirectional models.

Dropout

Using dropout (ranging from 10% to 50%) slightly improved the results of the model. Additionally, dropout proved to be a very useful normalization technique and it seemed to make the learning curve, and for that matter the learning process, smoother. It seemed that a small dropout yielded better results than bigger values of dropout. For that reason a 20% dropout was chosen for the final model.

Skip Connections

Skip connections marginally improved the accuracy of the models. It seemed to allow the model to retain much more information from the lower layers to the higher layers, thus aiding the learning process.

[No skip] 

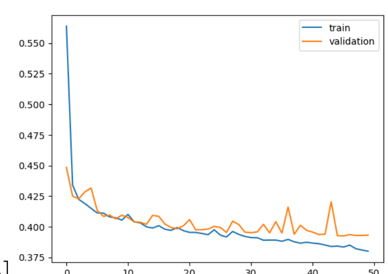
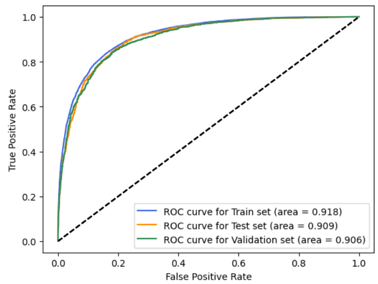
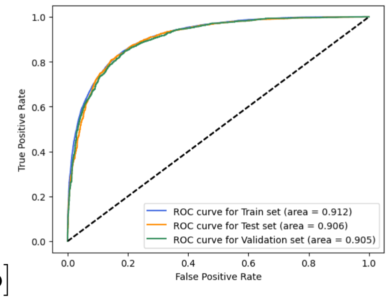
[skip] 

Figure 5: Learning curves of two models, one without skip connections and one with skip connections.



[No skip]



[skip]

Figure 6: ROCs of two models, one without skip connections and one with skip connections.

Gradient Clipping

Gradient clipping although useful in general did not prove to be much of a help in this case. This is probably due to the fact that we didn't have to deal with extremely large gradients. It seemed to have an effect in smoothing out the learning curve. In the final model a range of gradients [-20, 20] is allowed.

Type of Cells

Two type of cells were tested, namely LSTM and GRU cells. There was not much of a difference between the results of the two. After multiple runs with both the results of the LSTM cells seemed to edge out the results of the GRU cells, but the differences were not big at all. In antithesis, the GRU cells were consistently faster to train than their LSTM counterparts.

The GRU cells prooved harder to deal with and hyperparameter fine tuning was necessary. Lastly the GRU cells were prone to overfitting to the train set. As a result LSTM cells were chosen for the final model.

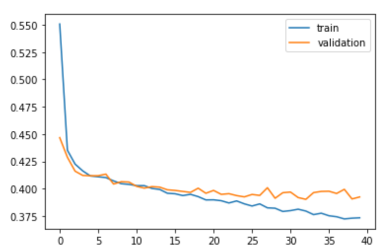
Attention

Attention was ultimately removed from the final model due to the fact that this implementation only run for unidirectional RNNs and I could not get it two work for bidirectional RNNs. Still, the source code is commented out. The classes regarding Attention, additiveAttention and multiplicativeAttention were developed by Tomek Korbak, PhD student, University of Sussex. The source code can be found on: https://tomekkorbak.com/2020/06/26/implementing-attention-in-pytorch/

Chosen Model

The chosen model is comprised of 2 layers of LSTMs. A hidden size of 200 is chosen. Skip connections, dropout of 20% and gradient clipping of 20 were chosen. The learning rate value is 0.0005 and the model is trained in 40 epochs.

The dataset is split into train set 80%, validation set 10% and test set 10%.

[Learning curve] 

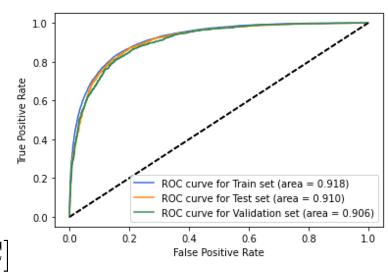
[ROC] 

Figure 7: Chosen Model's learning curve and ROC.

Train: Precision: 0.825, Recall: 0.855, F1: 0.840
Validation: Precision: 0.815, Recall: 0.844, F1: 0.829
Test: Precision: 0.832, Recall: 0.836, F1: 0.834

Extensive comments have been used throughout the whole source code, explaining every step of this implementation.

The source code was written using VSCode, although the code was also tested on https://colab.research.google.com/

Comparison to the models of the two previous assignments

The RNN model of this assignment edges out the results of the DNN created for the second assignment, but ultimately falls short to the performance of the model of the first assignment.

All three models were implemented in a way to both avoid underfitting and overfitting and thus comparing their learning curves would not be much of a use. As a result the f1 score seems as the best candidate to compare the three models. The classifier of the first assignment got a score of 0.897, the RNN got a score of 0.834 and finally the DNN got a score of 0.827.

2021 Data Mining Assignment

The 2021 Data Mining 1st Assignment also used GloVe embeddings. Snippets of code might have been used from my last year's project. The project was carried out with the help of Loukovitis Georgios - 1115201800100, who passed this class in 2021 and therefore won't be handing in any projects this year.