Natural Language Inference on SNLI Dataset

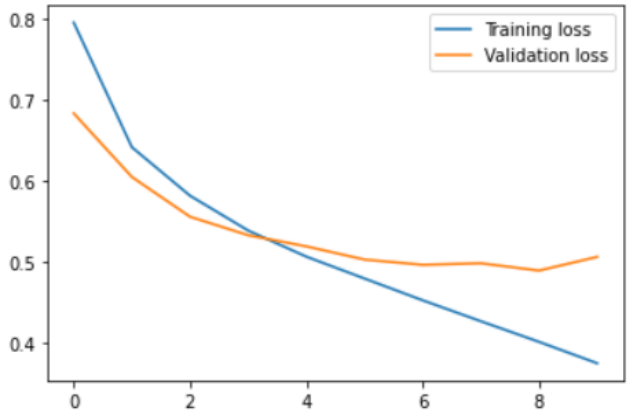
I decided to go with a LSTM model which incorporates GloVe embeddings rather than set up a tokenizer and train a new embedding layer. Expectedly the model with the pretrained embedding not only gave a higher accuracy but also reduced the training time per epoch by more than 3 times.

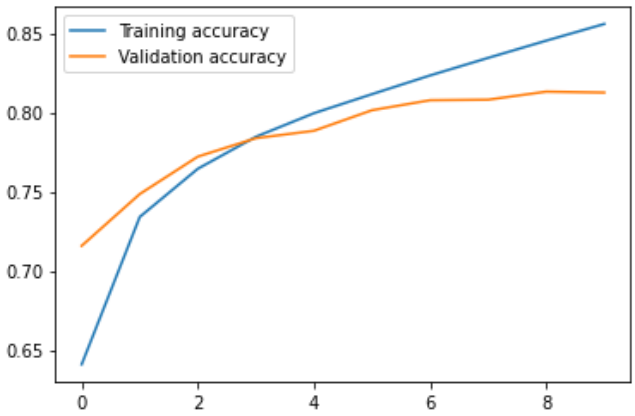
	Validation accuracy	Training time
LSTM from scratch	69.02	~ 10 m 12s
LSTM with GloVe	81.30	~ 3m 20s

The model was based of the GloVe Embedding implementation by GitHub user bentrevett[1]. It consists of a frozen embedding layer, a Dense layer and a 2 layered Bidirectional LSTM model through which the premise and hypothesis is passed, then after concatenating the two it is passed through a classifier consisting of 5 dense layers with dropout.

Since I used GloVe, the vocabulary has to be constructed according to the pre-trained vectorization and the embedding layer weights had to be frozen before starting to train the model.

The model only had to be trained for 10 epochs after which it started to overfit/plateau.





I also tried to carry out fine tuning by unfreezing the embedding layer and training the model with a very small learning rate, however that led to overfitting on the training set and a drop in validation and test accuracy.

Reference

1. PyTorch Natural Language Inference
https://github.com/bentrevett/pytorch-nli