

NLP Assignment 2

Timur Yakupov

March 2017

1 Introduction

Two tasks were given. I've chosen to implement a sentiment classifier using LSTM.

2 Implementation

2.1 Dataset

I decided to use an IMDB dataset. The data for IMDB dataset was collected by Stanford researchers and was used in a paper[1] where a split of 50/50 of the data was used for training and test. It is provided with keras.

2.2 LSTM

LSTM implementation from keras library for python was used. Keras has a good documentation and examples that I used.

2.3 Extensions

I decided to use a callback library from keras. I used `ModelCheckpoint()` in order to save best results while training. It is made in order to avoid overtraining and to ensure the best result. The model is saved into .hdf5 file

2.4 Structure

Implementation consists of one method called `lstm_nlp()` which takes two parameters:

- `mcw` - amount of most common words among the dataset that are taken into account.
- `review_len` - amount of words from each review taken among most frequent.

The default values are 7500 for `mcw` and 250 for `review_len`.

3 Results

First of all, I'd like to mention that my computer is weak and one epoch may take more than half an hour depending on parameters. So I tried to make it as fast as it can be.

```
25000/25000 [=====] - 2121s - loss: 0.1003 - acc: 0.9640 - val_loss: 0.4544 - val_acc: 0.8599
```

So, my first results. It is for 7500 most frequent words taken into account and 200 words from each review taken among most frequent. I also didn't use checkpoints to ensure best results here.

```
Epoch 13/15  
25000/25000 [=====] - 566s - loss: 0.0421 - acc: 0.9860 - val_loss: 0.6286 - val_acc: 0.8557  
Epoch 14/15  
25000/25000 [=====] - 565s - loss: 0.0329 - acc: 0.9886 - val_loss: 0.6548 - val_acc: 0.8539  
Epoch 15/15  
25000/25000 [=====] - 571s - loss: 0.0283 - acc: 0.9906 - val_loss: 0.7047 - val_acc: 0.8561  
24992/25000 [=====>.] - ETA: 0sTest score: 0.704691961455  
Test accuracy: 0.85608
```

Another try, the result for 7500 most frequent words and 250 words from each review is 0.85404. But the last checkpoint is at the point where result is:

```
Epoch 6/15  
24992/25000 [=====>.] - ETA: 0s - loss: 0.1792 - acc: 0.9306Epoch 00005: val_loss improved from 0.35848 to 0.35739, saving model to /tmp/weights  
.hdf5  
25000/25000 [=====] - 671s - loss: 0.1793 - acc: 0.9306 - val_loss: 0.3574 - val_acc: 0.8650
```

4 Conclusion

I think that it is possible to get a better result by just changing the parameters, but it takes too much time to train, so my highest result so far is 86.5%.

References

- [1] Andrew L Maas et al. "Learning word vectors for sentiment analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics. 2011, pp. 142–150.