Hamidreza Raei

Coding Assignment 2

Final Report

Construction of the data:

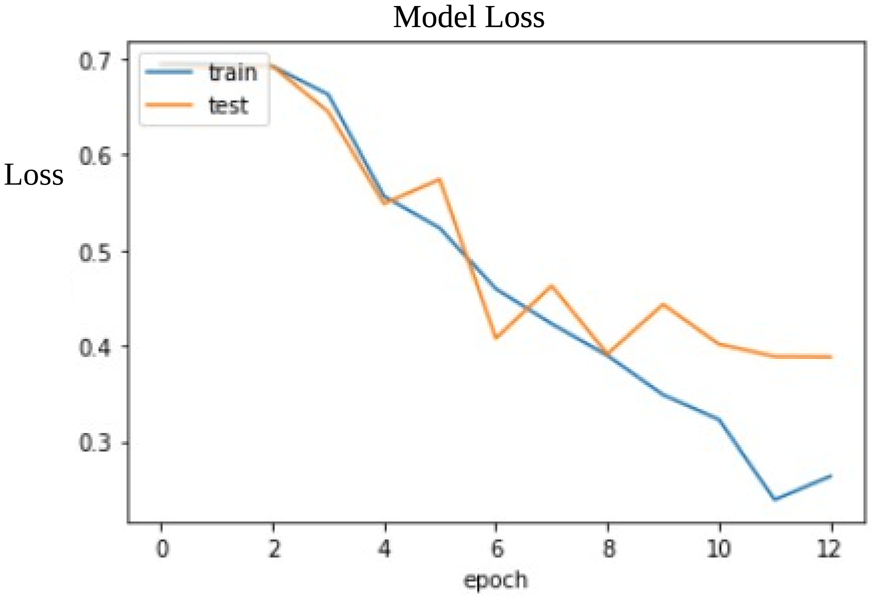
Statistics	Value
The total number of unique words in T	15480
The total number of training Ex	1600
The ratio of positive ex to negative ex	799/801 ~ 1
The average length of document in T	187.6 words
The max length of document in T	200

Performance of deep neural network classification:

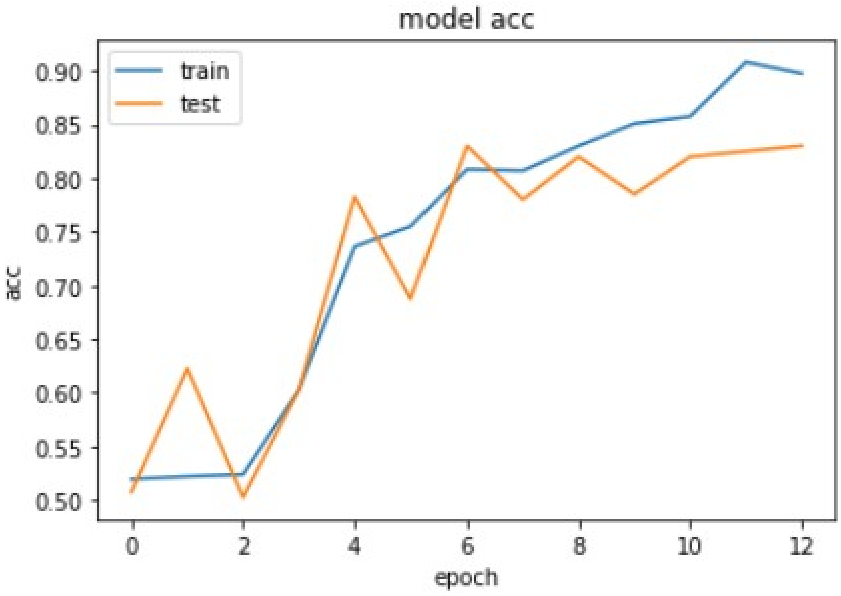
	Accuracy	Training Time
RNN w/o pretrained	0.7624	60
RNN with pretrained	0.854	50
CNN w/o pretrained	0.7825	45
CNN with pretrained	0.815	40
The max length of document in T	200	

Below you can see plots for CNN model with pretrained embedding:

Loss – Time:

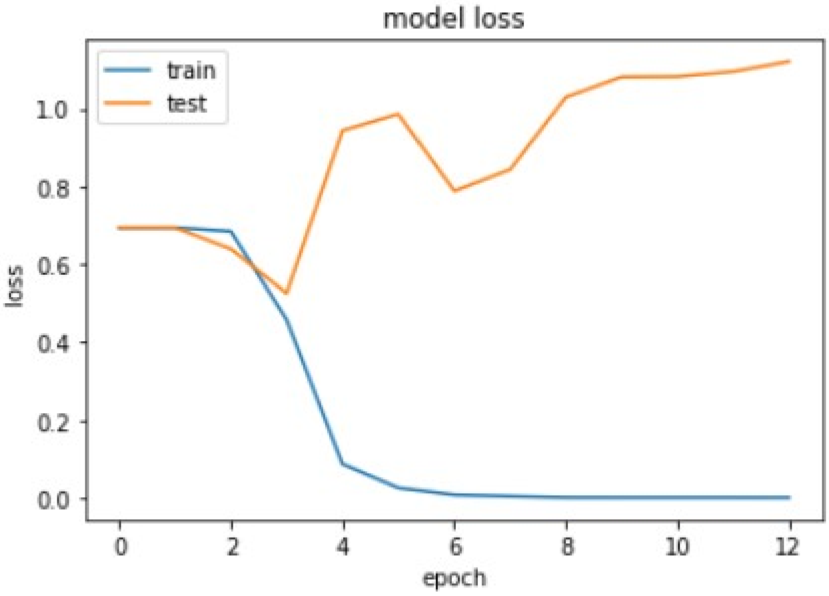


Accuracy – Time:

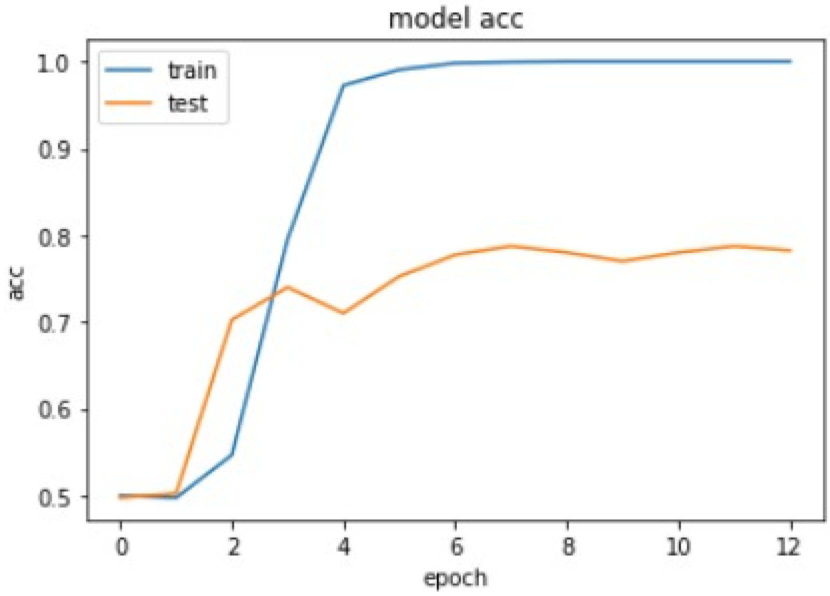


Below you can see plots for CNN model without pretrained embedding:

Loss – Time:

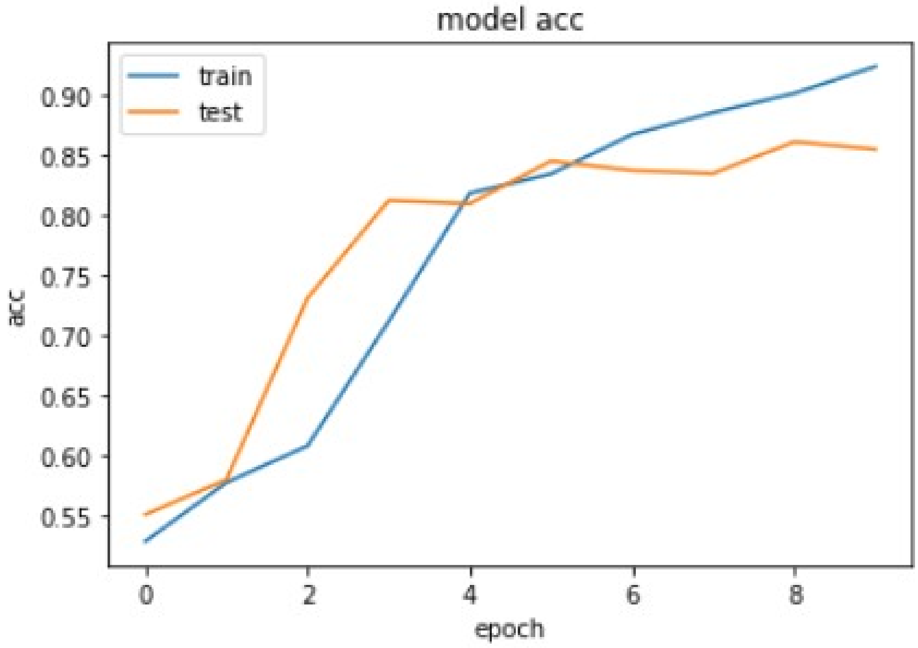


Accuracy – Time:

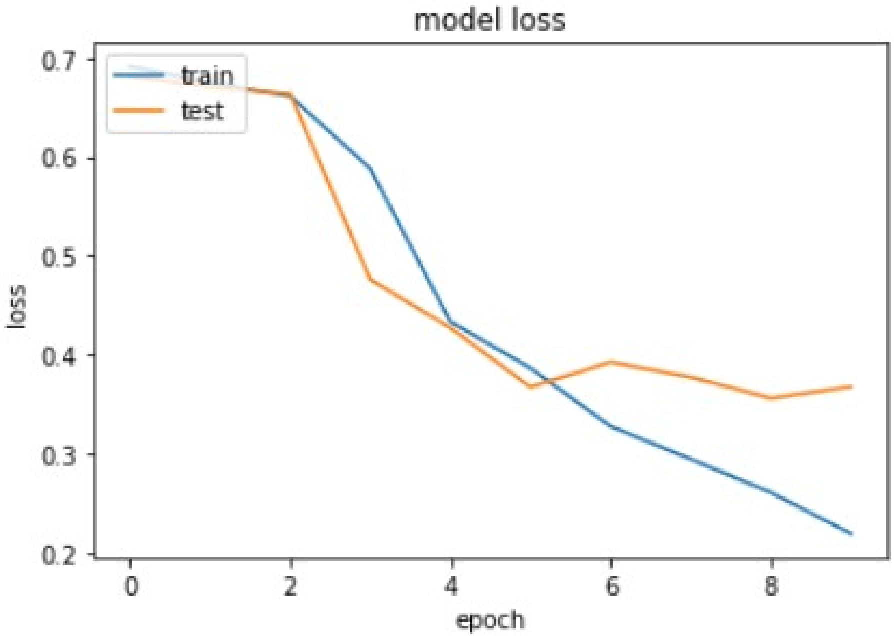


Below you can see plots for RNN model with pretrained embedding:

Loss – time:

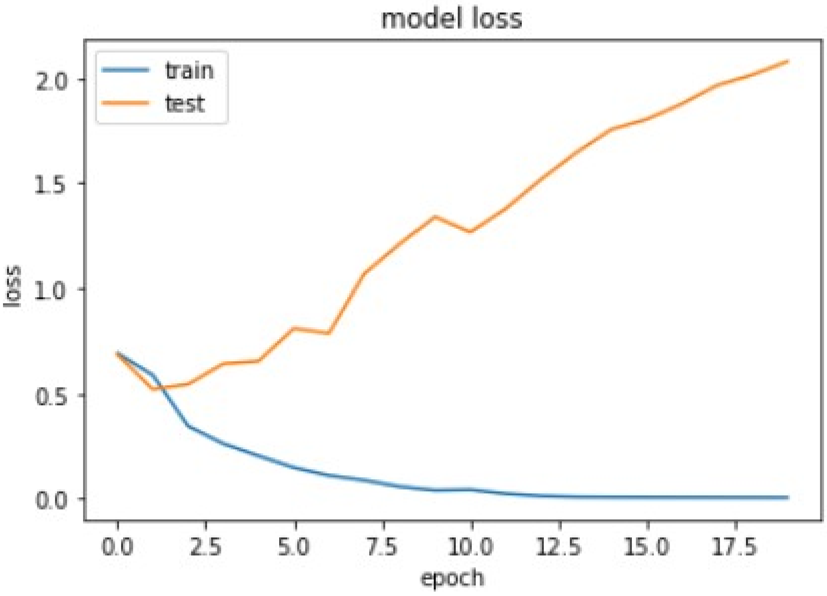


accuracy – time:

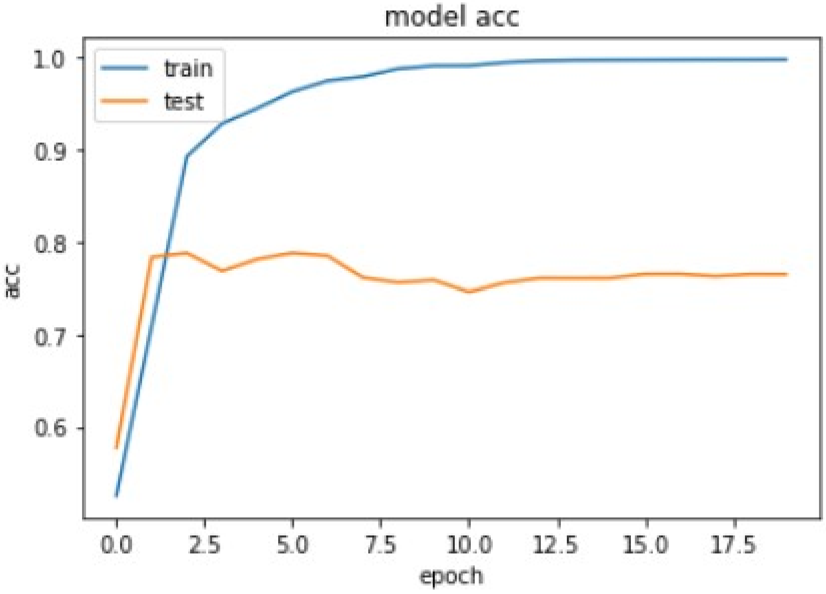


Below you can see plots for RNN model without pretrained embedding:

loss – time:

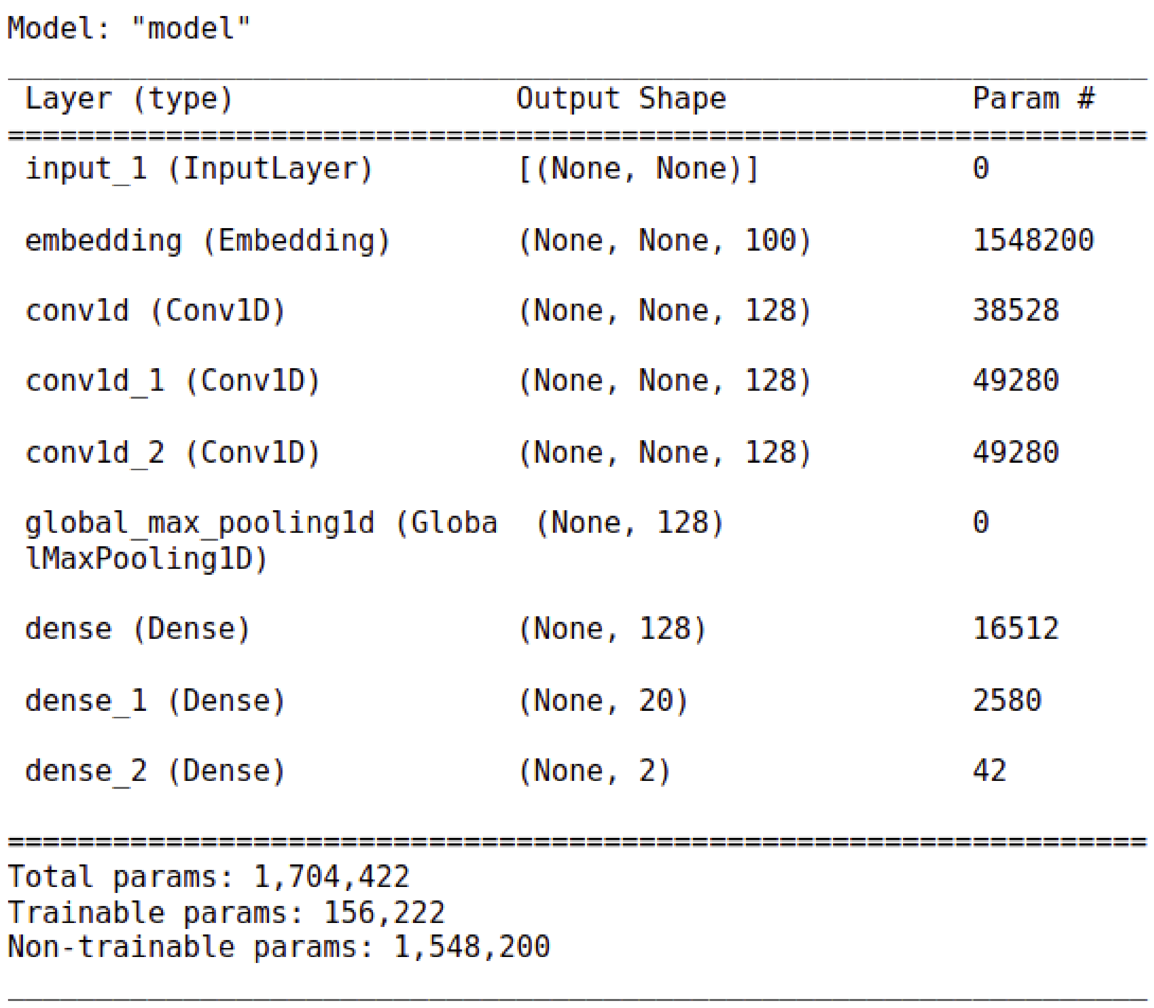


accuracy – time:



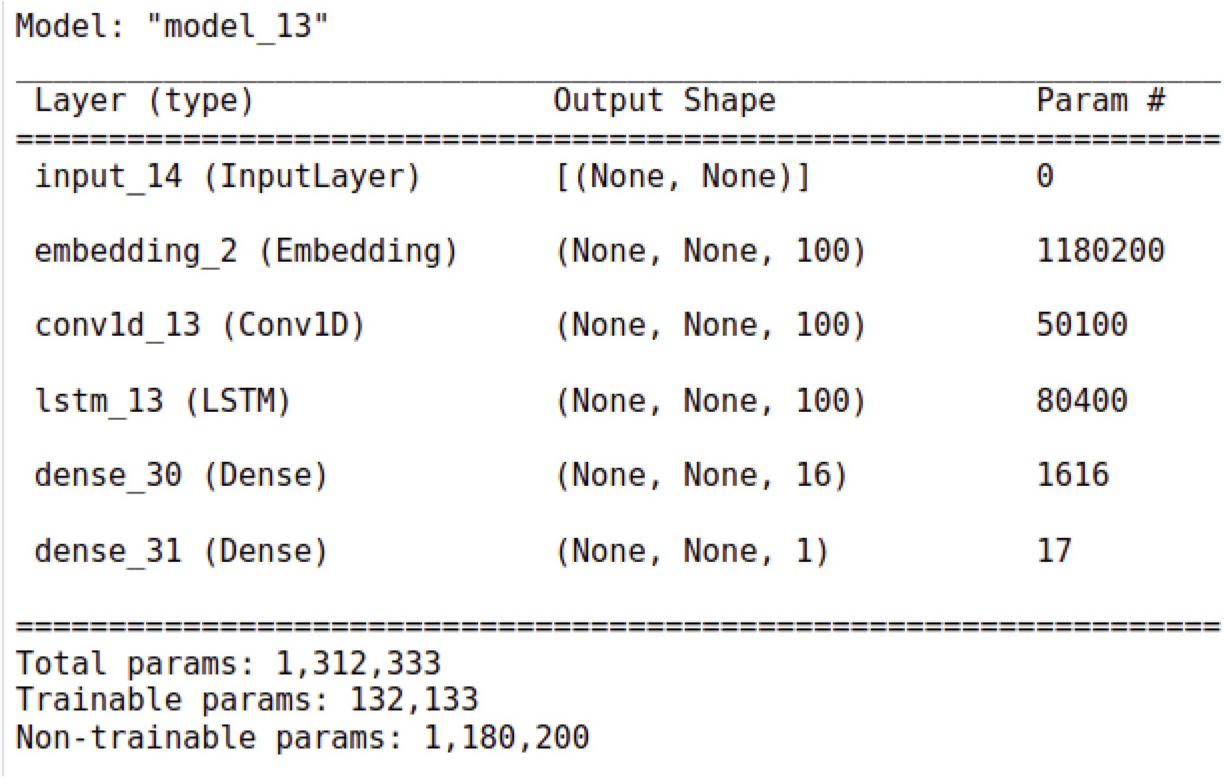
Also, you can see structure of the CNN and RNN below:

CNN:



```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, None)]            0

 embedding (Embedding)       (None, None, 100)         1548200

 conv1d (Conv1D)             (None, None, 128)         38528

 conv1d_1 (Conv1D)           (None, None, 128)         49280

 conv1d_2 (Conv1D)           (None, None, 128)         49280

 global_max_pooling1d (Globa  (None, 128)              0
 lMaxPooling1D)

 dense (Dense)               (None, 128)               16512

 dense_1 (Dense)             (None, 20)                2580

 dense_2 (Dense)             (None, 2)                 42

=================================================================
Total params: 1,704,422
Trainable params: 156,222
Non-trainable params: 1,548,200
_________________________________________________________________
```

RNN:



```
Model: "model_13"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_14 (InputLayer)       [(None, None)]            0

 embedding_2 (Embedding)     (None, None, 100)         1180200

 conv1d_13 (Conv1D)          (None, None, 100)         50100

 lstm_13 (LSTM)              (None, None, 100)         80400

 dense_30 (Dense)            (None, None, 16)          1616

 dense_31 (Dense)            (None, None, 1)           17

=================================================================
Total params: 1,312,333
Trainable params: 132,133
Non-trainable params: 1,180,200
```

Analysis of result:

It seems that when we start to do the training without using pretrained embedding the accuracy cannot increase as much as the model that was trained with pretrained embedding. Also, when we look at the loss of the model it looks really weird. Even though, the accuracy is almost not chnging after two or three epoch the validation loss starts to increase that I think this results shows that not using pretrained embedding might lead to overfitting and this is what happened to our model here. In case of pretrained embedding, over fitting can happen too, but not in second or third epoch, it usually happens after 10 epochs.

Overal, results of RNN was better than CNN for me, however, it seems when I did not use the pretrained embedding, the CNN had better result compared to the RNN, that I cannot find the reason.

The software implementation:

In this project, I used tensorflow and keras for implementation of the Neural network. Especially I believe implementation of LSTM is much easier using keras. Also, for vectorizing and shuffling the training data it had specific function which I believe made it easier for me.

For writing the prediction into the csv file I used pandas library.

I also used codecs library in python for opening the dataset file and reading the number of words.