Assignment 5 Report

Harsh Vora
CWID : A20445400
CS577 – S20

May 2, 2020

Problem Statement:

To train the network for Topic classification – Reuters newswire topic and tune the various model parameters.

Design Details:

I have trained my network using the below model.

I have used the Sequential model.

For classification I have used the below network layers:

1. Embedding(10000,100,100)

2. Flatten Layer

3. Dense(128,activation='relu')

4. Dense(46,activation='softmax')

5. I have used rmsprop as optimizer and loss as categorical_crossentropy.

Implementation Details:

1. I loaded the Reuters newswire dataset from keras by using function reuters.load_data().
2. Here I chose the number of words as 10000.
3. After that I split the data into training, testing and validation.
4. Then I created a Sequential model by adding an embedding layer with max_features as 10000, embedding_dimensions as 100 and maximum length as 100 along with two fully connected layers and trained the model.
5. Then I downloaded the Glove Word embedding file from *http://nlp.stanford.edu/data/glove.6B.zip* and initialized the weights fixed.
6. After that I again trained the model but this time I freezed the embedding layer weights.
7. Then I unfreezed the embedding layer weights and checked the accuracy.
8. After that I replaced the Dense Layer with LSTM layer.
9. At last I added one more LSTM layer and repeated training.

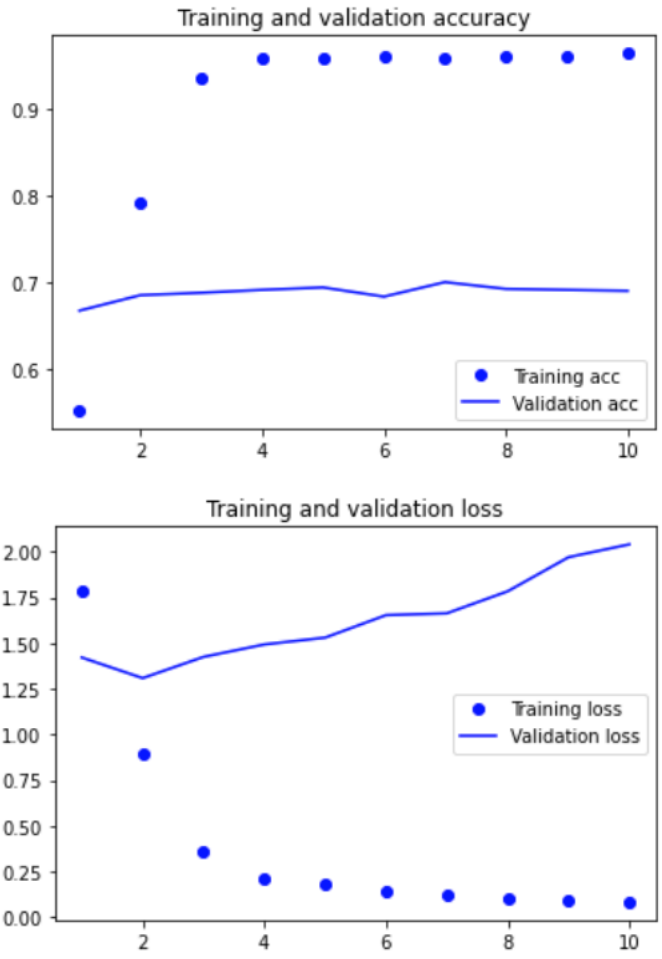
Results:

Below are my results for Reuters newswire Dataset : I have measured the Test accuracy for a trainable embedding layer, Freezed embedding layer weights, Unfreezed embedding layer weights and One and two LSTM layer with trainable embedding layer. I have trained all the models for 10 epochs.

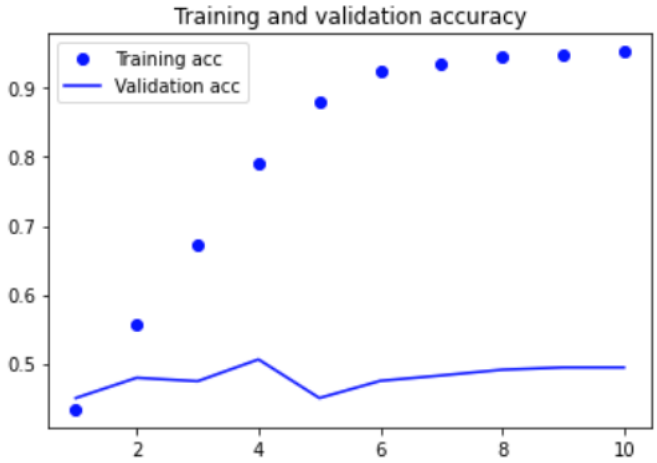
These are my results:

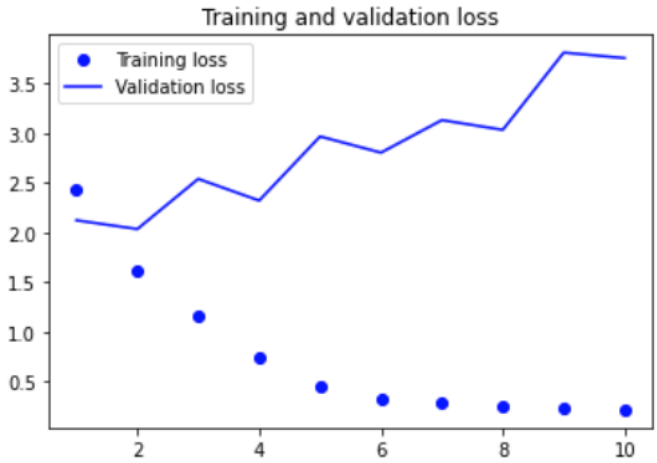
Model	Accuracy
Trainable embedding layer	67.36
Freezed embedding layer weights	49.95
Unfreezed embedding layer weights	52.14
One LSTM layer with trainable embedding layer	63.49
Two LSTM layer with trainable embedding layer	68.25

Trainable Embedding Layer:

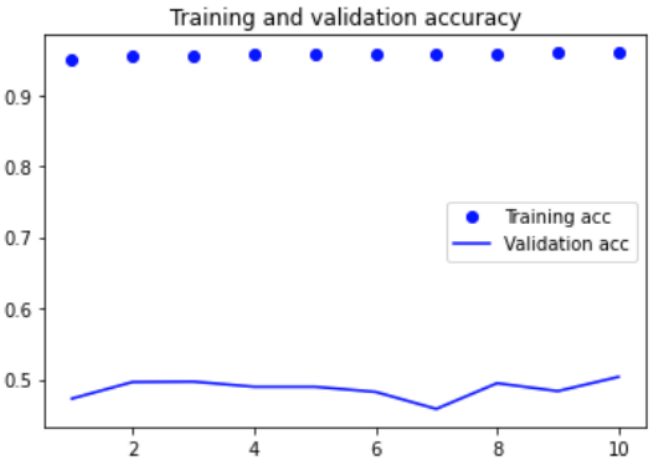


Freezed Embedding Layer:

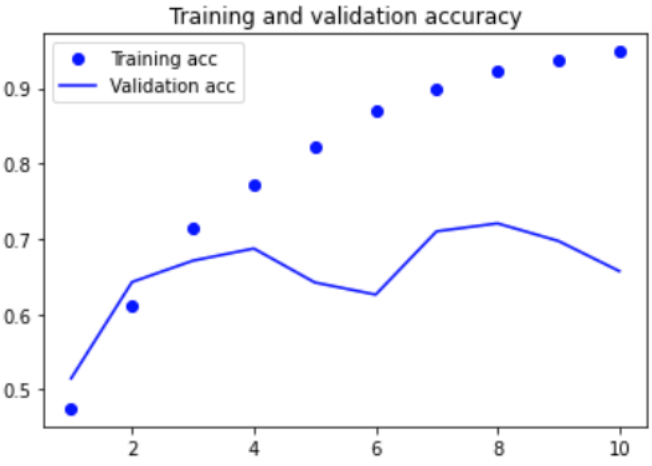




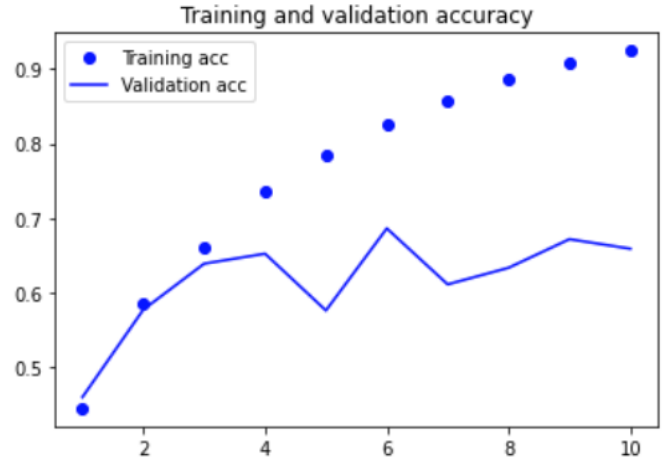
Unfreezed Embedding Layer:



One LSTM Layer with Trainable embedding layer:



Two LSTM Layer with Trainable embedding layer:



Conclusion:

I got the best results by adding two LSTM layers i.e. 68.25% whereas you can see it is almost similar to that of the normal Trainable embedding layer. Adding more LSTM layers can give better accuracy. In my case, adding a second layer improves the accuracy from 63.49% to 68.25%.