Assignment 5 Report

Harsh Vora
CWID : A20445400
CS577 – S20

May 2, 2020

Problem Statement:

To train the network for sentiment classification – IMDB movie reviews 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(64,activation='relu')

4. Dense(1,activation='sigmoid')

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

Implementation Details:

1. I loaded the IMDB data from keras by using function imdb.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. At last I replaced the fully connected layer with LSTM layer and repeated training.

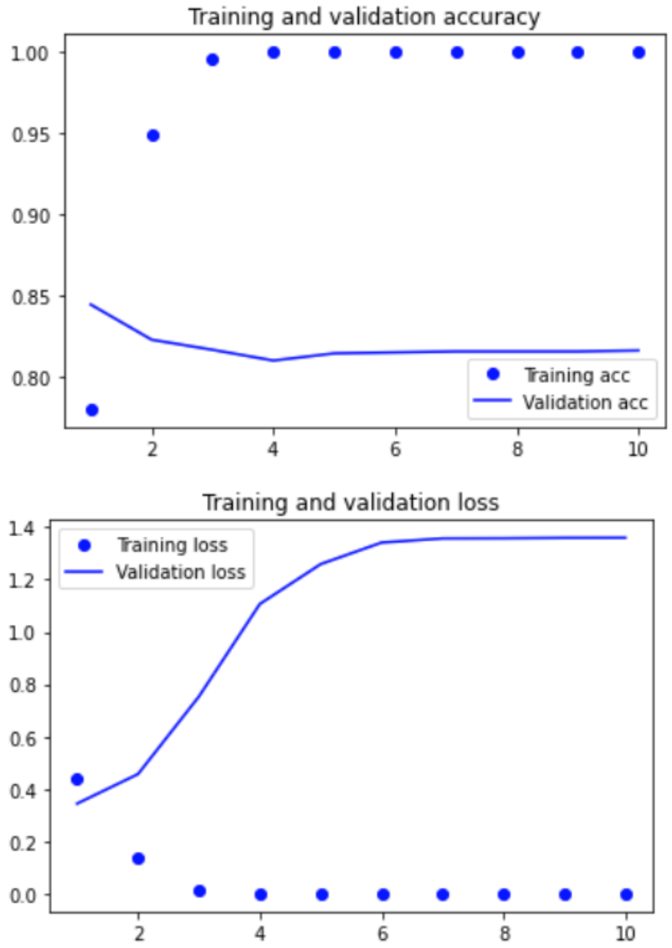
Results:

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

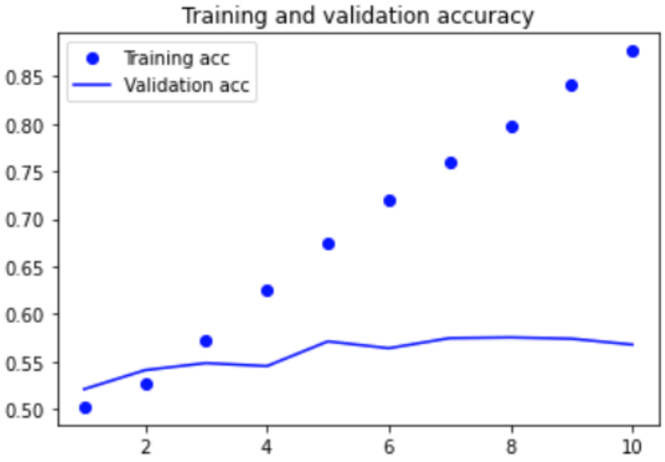
These are my results:

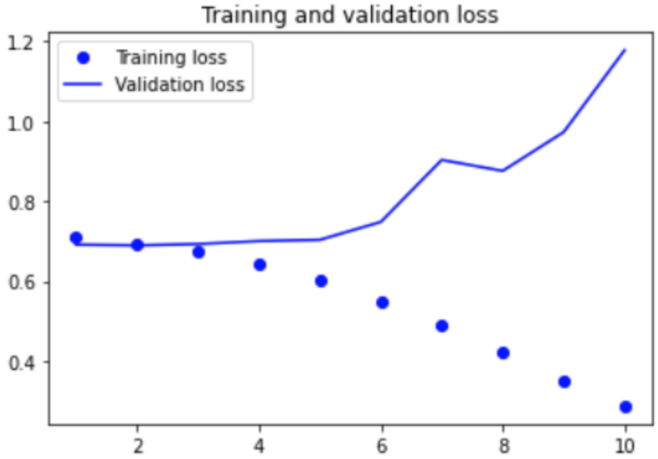
Model	**Accuracy**
Trainable embedding layer	81.53
Freezed embedding layer weights	56.78
Unfreezed embedding layer weights	72.18
LSTM layer with trainable embedding layer	82.56

Trainable Embedding Layer:

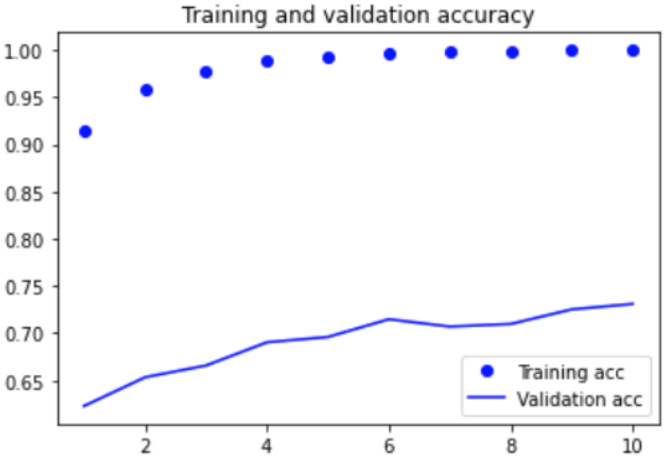


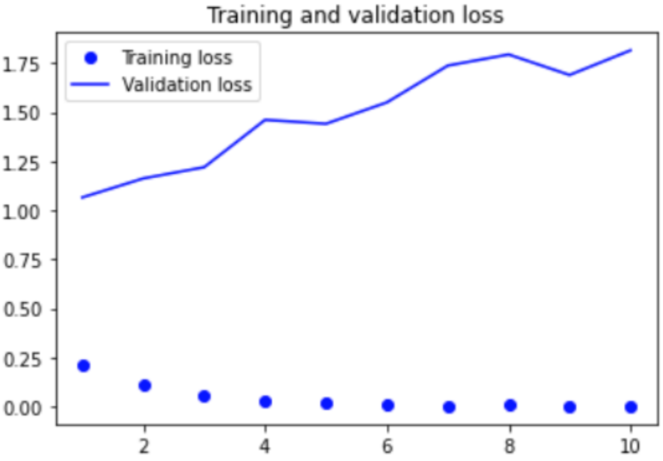
Freezed Embedding Layer:



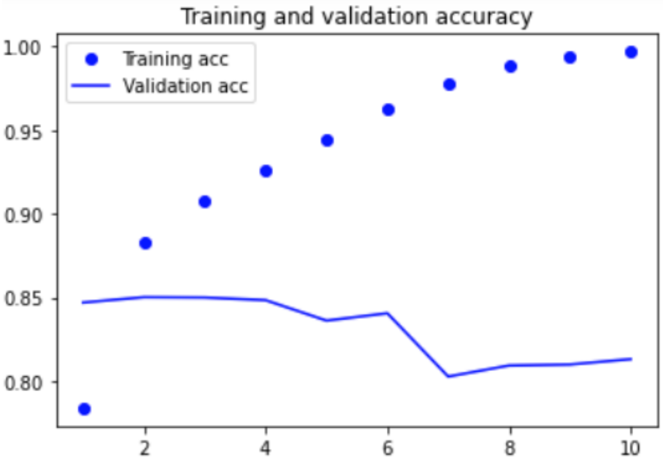


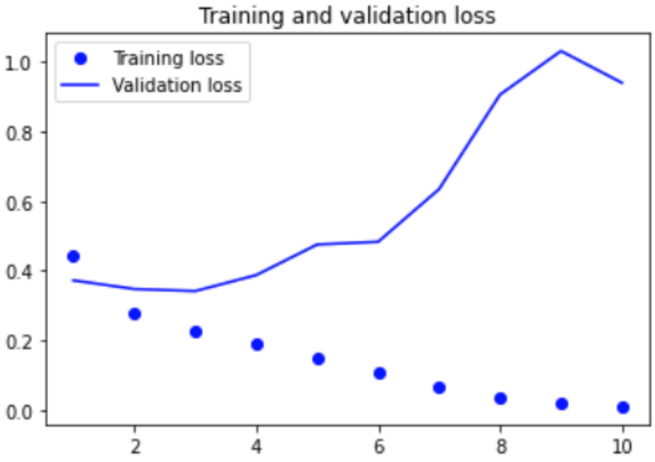
Unfreezed Embedding Layer:





LSTM Layer with Trainable embedding layer:





Conclusion:

I got the best results by adding LSTM layers i.e. 82.56%. I have trained the LSTM model on same set of hyperparameters and the model performed giving best accuracy among all models. It is because the expected output (target) of LSTM model is compared with the model's output and the weights are updated accordingly.