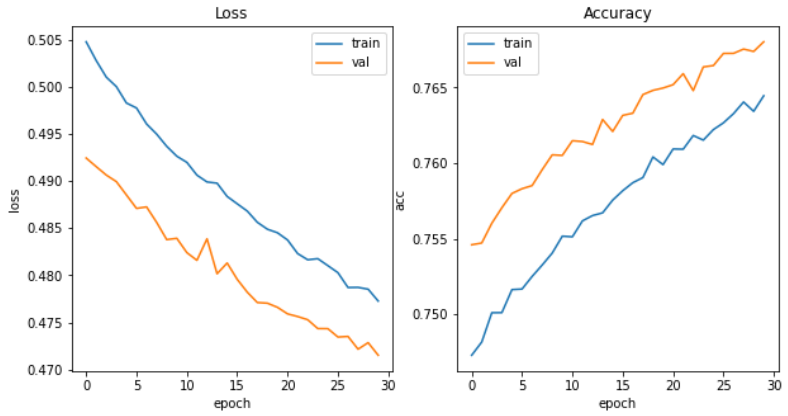
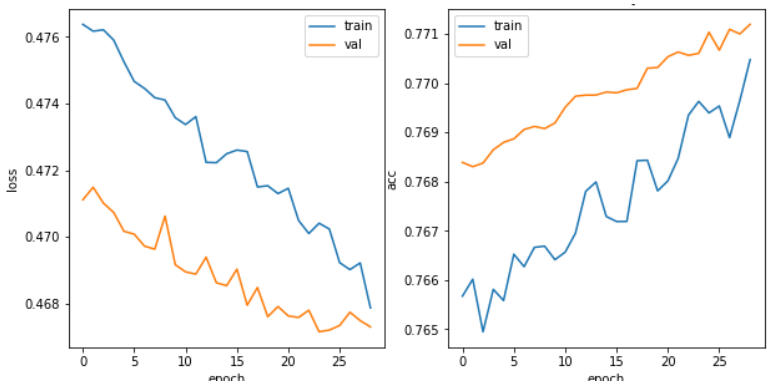
LAB - Text classification

(Please note that classification report and confusion matrix have only been obtained for experiments where the accuracy was touching 80% i.e experiment 6, 7, 8)

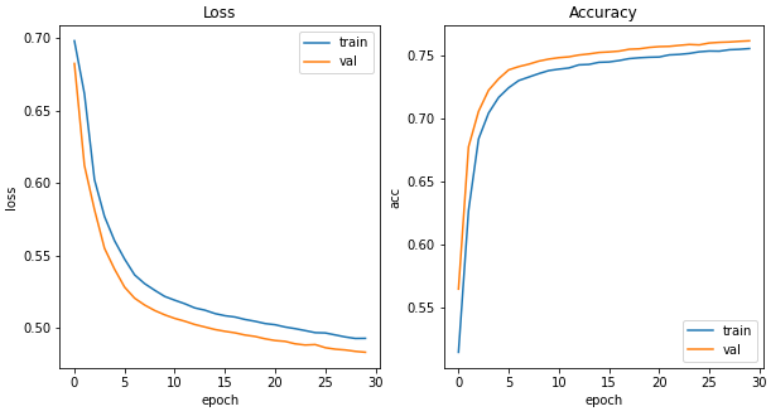
Experiment 1:
Embedding = Word2Vec (generated using gensim)
Preprocessing => removal of punctuations, numbers and lemmatization
Window for word2Vec = 10
Embedding Dimension = 100
Batch Size = 8192
LSTM units = 64
Dense layer = [1 (Sigmoid activation)]
Epochs = 30
Learning rate = 0.001



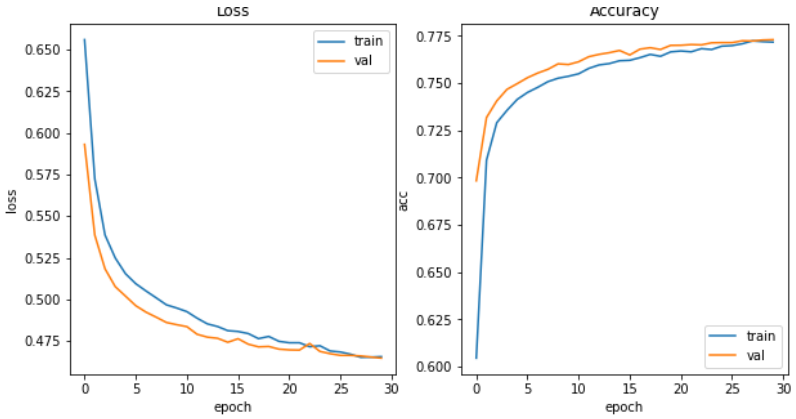
Experiment 2:
The model was training very slowly so to speed up the process the batch size was increased to 16384. The epochs were also increased to 50 but the early stopping callback with patience of 5 stopped the training at 29 epochs. Nevertheless, the model accuracy increased by the same amount.



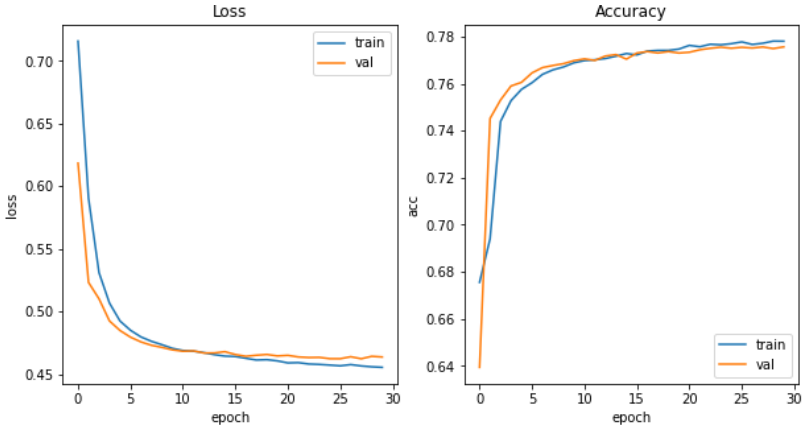
Experiment 3:
Similar to experiment 1 just another dense layer with 32 neurons was added. Batch size was again 16384. The graphs look smoother now but that's because the initial accuracy was lower and the matplotlib adjusted the scale of the graph to fit in the initial points. The accuracy after 30 epochs is lower than what was achieved without using the additional dense layer.



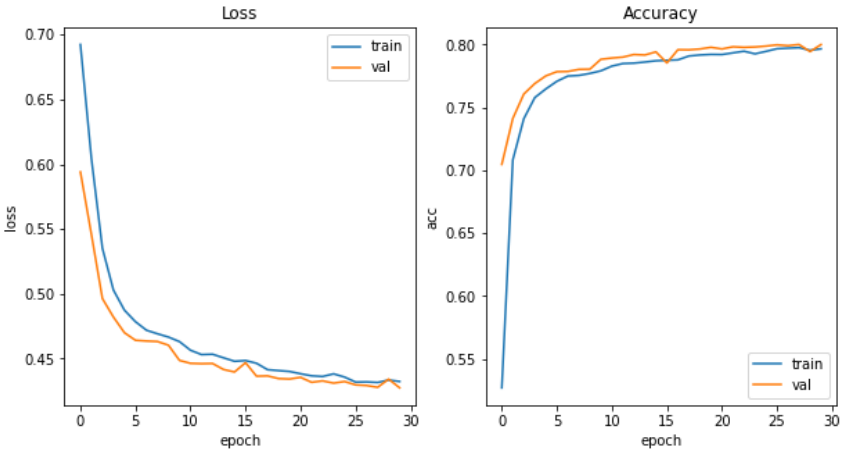
Experiment 4:
With increasing the learning rate from 0.001 to 0.005 much better accuracy was achieved in 30 epochs



Experiment 5:
Since the model achieved much better results with a learning rate of 0.05 so I increased the learning rate again to 0.01 and slightly better results were achieved.



Experiment 6:
I added another LSTM layer with 32 units and return_sequences = True on top of the LSTM layer used in the previous models and also increased the number of neurons in the Dense layer to 128. This lead to improvement in accuracy to peak over 80%



Here is the classification report for this experiment:

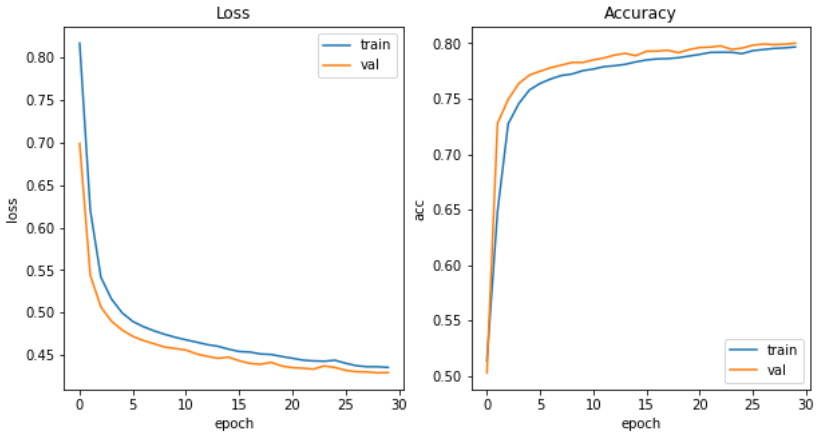
	precision	recall	f1-score	support
0	0.788	0.824	0.806	33600.000
1	0.815	0.779	0.797	33600.000
accuracy	0.801	0.801	0.801	0.801
macro avg	0.802	0.801	0.801	67200.000
weighted avg	0.802	0.801	0.801	67200.000

Here is the confusion matrix:

	0	1
0	27671	5929
1	7428	26172

I allowed another 30 epochs but the model accuracy did not go much above 80% so I interrupted the training.

Experiment 7:
I added a Conv1D with 64 filters and kernel size 5 to emphasize more on the short-term features. Along with the convolutional layer I also added a max-pooling and flatten layer to get data in appropriate dimensions. Similar accuracy of around 80% was achieved.



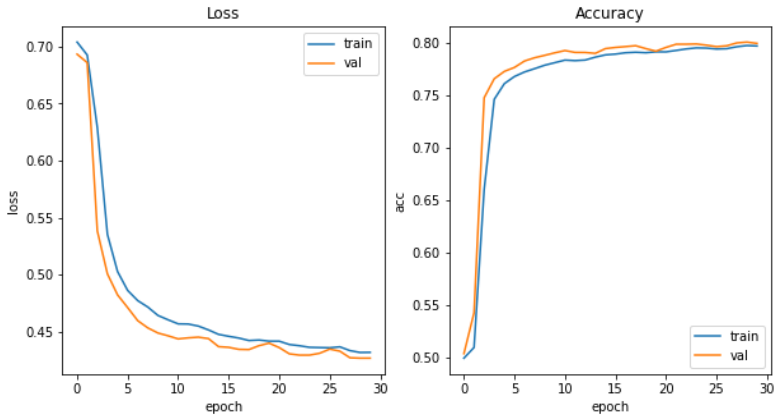
Here is the classification report for this experiment:

	precision	recall	f1-score	support
0	0.813	0.781	0.797	33600.000
1	0.789	0.821	0.805	33600.000
accuracy	0.801	0.801	0.801	0.801
macro avg	0.801	0.801	0.801	67200.000
weighted avg	0.801	0.801	0.801	67200.000

Here is the confusion matrix:

	0	1
0	26245	7355
1	6022	27578

Experiment 8:
Having done with LSTM I moved on to GRU. This was similar to experiment 6 but LSTMs were replaced with GRUs



Here is the classification report for this experiment:

	precision	recall	f1-score	support
0	0.792	0.811	0.801	33600.000
1	0.806	0.787	0.797	33600.000
accuracy	0.799	0.799	0.799	0.799
macro avg	0.799	0.799	0.799	67200.000
weighted avg	0.799	0.799	0.799	67200.000

Here is the confusion matrix:

	0	1
0	27253	6347
1	7158	26442