Assignment 4: Text and Sequence Data

By Jeremy Glasgow

Results

Model	Test Accuracy
1. Cutoff after 150 words	0.73
2. 100 training samples	0.63
3. 10,000 validation samples	0.88
4. Top 10,000 words	0.87
5. Embedded Layer	0.53
6. Pretrained word embedding	0.50
7. Pretrained word embedding, but with adapted training vectorization	0.50
8. Started from beginning with preprocessing and model building using pretrained word embeddding	0.64

Summary:

Starting off with Assignment 4 on text and sequence data, I had a difficult time understanding the preprocessing approach. The preprocessing from Chapter 6 did not translate well for the model building, so I explored with my own preprocessing approach at the end. To start, the first model was configured to cut off reviews after 150 words. This achieved a test accuracy of 0.73. The second model was configured to only use 100 training samples, which achieved a test accuracy of 0.63. The drop in performance was expected because there was less data available to train the model. The third model was configured to only use 10,000 validation samples, which achieved a test accuracy of 0.88. This was by far the best performing model because there were no restrictions on the volume of data used by the model. The fourth model was configured to only use the top 10,000 words from the IMDB dataset, which was the number of words already selected at the beginning of the preprocessing phase. This model achieved a test accuracy of 0.87 which was very similar to the third model attempted. The fifth model was configured to include an embedded layer, which achieved a test accuracy of 0.53. The sixth model was configured using Standford's Glove (Global Vector's for Word Embeddings) dataset to build the embedding layer. This model achieved a test accuracy of 0.5 and took quite a long time to train. The seventh model was configured to use the same pretrained word embedding, only I made sure to adapt the text vectorization to the IMDB dataset before building the model. This model performed similarly to model six, achieving a test accuracy of 0.50. The eighth and final model, I started the preprocessing process from the beginning to ensure the data was represented properly. I applied the same model building and pretrained word embedding techniques used for models six and seven, and achieved a slightly higher test accuracy of 0.64. By far, the best performing models were numbers three and four. These two models leveraged a model with an input layer, a dense layer, a dropout layer, and a final dense layer. Both generalized well and achieved similar test accuracy scores. While embedded layers and pretrained word embeddings might

result in better performance in other scenarios, I was still able to achieve very good test accuracy scores with a relatively simple model.