**Assignment 4**

**Text And Sequence Data**

**Summary:**

The basic idea of the binary classification project with the help of IMDB dataset is to always predict whether the given movie will have a positive review or not. The dataset we have considered contains 50000 reviews There are 10000 most frequent words selected. Training samples of 100, 500, 1000, and 100000 are trained and validated of 10000 samples using the cutoff of 150 words. The data is preprocessed. Then for the next step include data fed into the pre-trained embedding model and the embedding layer and contrast the results from different methods for measuring performance.

The given RNN model is then trained with the IMDB dataset which consist of only reviews and at last binary decision is made to accept which either positive or negative reviews. This is the system of a sequential neural network method that works and operates in only one direction. RNN has the recurrent connection which is useful to handle with usage data.

**Word Embedding:**

Contrary to one-hot encodings, word embeddings comprise a way of representing words as vectors in enormously high-dimensional space. These vectors make machine algorithms to understand and interpret the concept of words and sentences of them more effectively as representing semantic relations between words.

**Pre trained word embedding:**

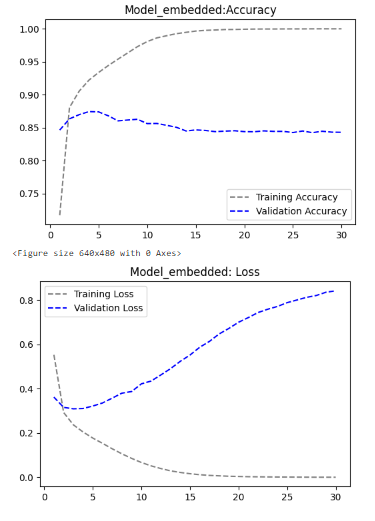
Like these models of neural networks, there are word embeddings that have been trained on a vast amount of unannotated text. These method can be trained using Words2Vec, GloVe and FastText, all of which are models capable of processing large text data. First, in various kinds of natural language processing tasks, pre-trained embeddings can be served as the set of features without training the model from the small scratch on the same dataset costs both time and computation resources.

**Results:**

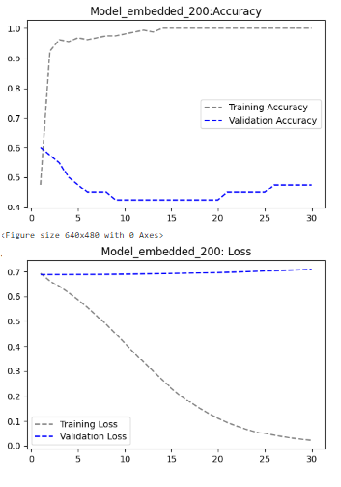
Initially using only, the embedded layer with different training sizes:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Review Words | Max Length | Training Size | Accuracy | Loss |
| 150 | 10000 | Minimum | 84.28 | 84.17 |
| 150 | 10000 | 200 | 47.50 | 70.96 |
| 150 | 10000 | 500 | 57.00 | 83.12 |
| 150 | 10000 | 1000 | 61.00 | 80.69 |
| 150 | 10000 | 2000 | 71.50 | 68.18 |

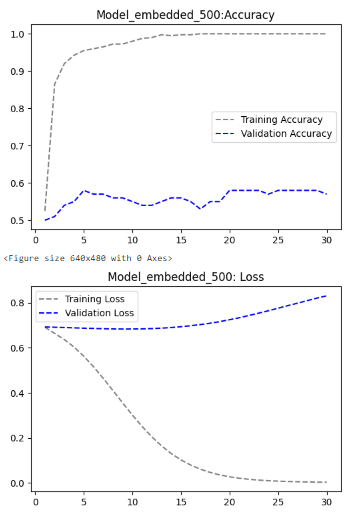
**A single embedded layer:**



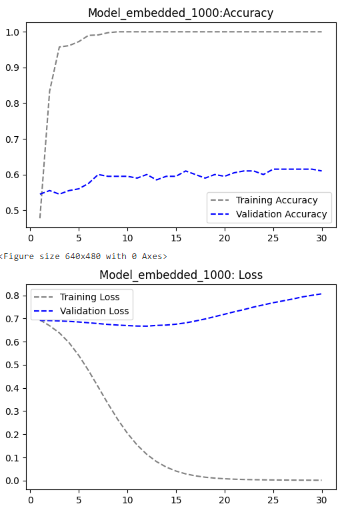
**A single embedded layer with 200 training size:**



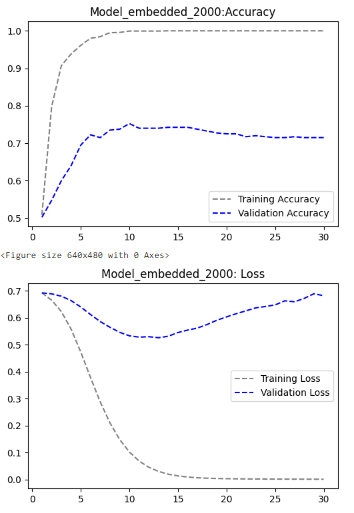
**A single embedded layer with 500 training size:**



**A single embedded layer with 1000 training size:**



**A single embedded layer with 2000 training size:**

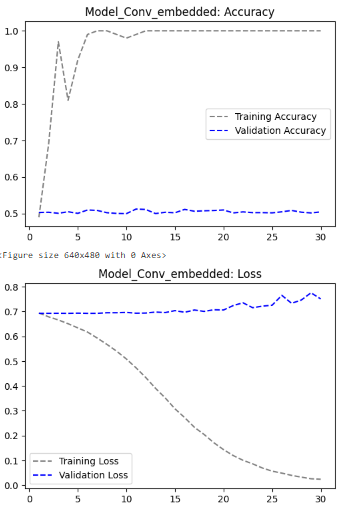


During training, high precision may be obtained quite fast which marks overtraining, particularly where the sample sizes are much smaller for the only embedded layers models, as with the model with 500 in training area having slower yet stable convergence rate.

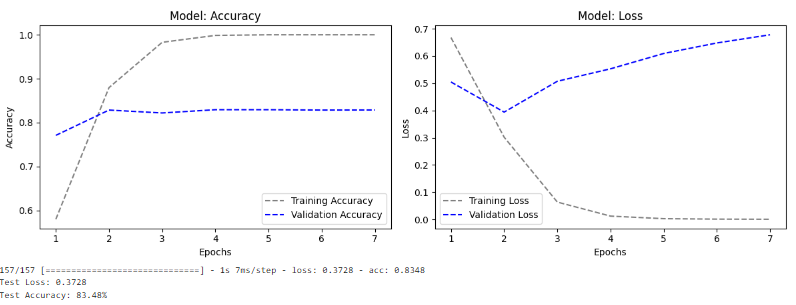
Now we will be using pretrained word embedding; that is using an embedded layer and a convolution 1D with different embedding dimensions:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Review Words | Max Length | Embedding Dimension | Accuracy | Loss |
| 150 | 10000 | 10 | 50.91 | 75.31 |
| 150 | 10000 | 50 | 52.08 | 69.24 |
| 150 | 10000 | 10000 | 93.14 | 23.56 |

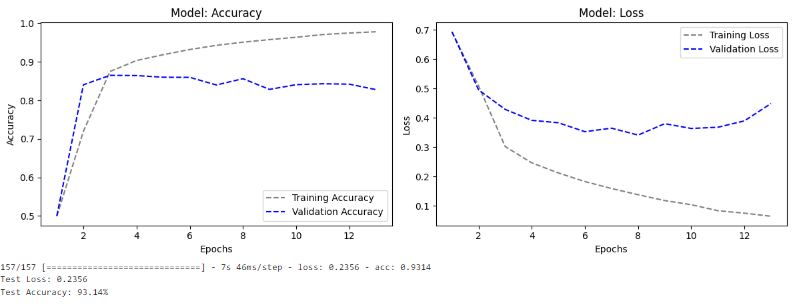
**Pretrained model with embedded layer and convolution 1D:**



**Pretrained model with 50 embedded dimensions:**



**Pretrained model with 10000 embedded dimensions:**



It looks that using a pretrained word embedding with both Embedding and Conv1D layers we are facing an overfit, this might caused by a complexity of the network and the size of our dataset.

Later for embedding dimension we used simple architectural designs or use the dropout approach to increase regularization or to reduce overfitting and increase in accuracy other than feature selection. But yet, with all these scenarios, we did not get the best accuracy with the least loss. Because the loss should be least mean and accuracy should be maximum for a best perform all models.

An embedding dimension of 10,000 is used into the W2V for word representation. To handle the overfitting issue dropout is employed after every convolution operation and convolution layers are then followed by pooling layers in this case MaxPooling1D is used to down sample on spatial dimensions. The model is trained using sixty percent of the dataset with an approximate average training accuracy of 97.80 % and the model validation accuracy is approximately 82.79 %. In particular, the test accuracy is 93.14% with the least loss 23.56%. This high accuracy level for the both training and the validation datasets can suggest a reasonable balance of the model’s complexity and its abilities in generalization. The accuracy of 93.14% enhances the implication of the model on unseen data in model generalization.

We also tried two different models:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Review Words | Max Length | Model | Accuracy | Loss |
| 150 | 10000 | RNN | 81.75 | 72.02 |
| 150 | 10000 | LSTM | 82.16 | 52.14 |

In general, RNN learns the series of word vectors and finds out features important for sentiment analysis. It has 64 RNN units that allow it to understand 64 features of sequential data present. The model can be greatly used to tag IMDB movie reviews as positive or negative. However, in our dataset Simple RNN fails to learn long sequences, therefore we include LSTM and GRU layers to solve the vanishing gradient.

As we can see by using LSTM model, there is a slight increase in accuracy and decrease in loss.

**Conclusion:**

• The high accuracy on the training set and the validation set also illustrate the appropriate level of model complexity as well as the model’s ability of generalization.. The pre-trained model accuracy of 93.14% with containing dimensions of 10000 also reveals that even the unseen data summarizing and regularization capability of the model can be made better.

• There values were continually improving as the training size increased and the actual loss was decreasing indicating that as the training size increases the model learns..

• The training loss of the model decreased as more training samples were used as evidence that the model can learn with more samples..

• Extant remedial approaches such as masking and the attempt of various dimensionality of the embeddings with the incorporation of dropout or with the help of fine-tuning the model would improve the model significantly..

• The tested models of two have the similar results, means there is not one method which works for all the models of embedding layer.. To select the most suitable model for a given data set it is customary to consider a large variety of models. When Regularization is in use, when trying out different sizes for embedding, when building LSTM layers, when the pre-trained embedding is trained using a given provided dataset and when the training sample is increased to a certain level, then performance can improve.