**Assignment 4 Report**

**Name: Srilaya Valmeekam**

**SUMMARY**

The objective of the binary classification issue on the IMDB dataset is to determine if a movie review is either positive or negative. The dataset is made up of 50,000 reviews, of which evaluation is just the top 10,000 words, limiting training samples to 100, 500, 1000, and 100000, validating on 10,000 samples, and examining reviews after 150 words. The data undergoes pre-processing. Afterwards, the data is fed to a pretrained embedding model as well as the embedding layer, and test various strategies to evaluate performance. The main goal of the binary classification task on the IMDB dataset is to determine which method performed better and whether a movie review is positive or negative.

**TECHNIQUES**

**Preprocessing of the dataset**:

Positive and negative sentiment labels are attached to movie reviews in the IMDB dataset.

The preprocessing of the dataset involves turning every review into a series of word embeddings, where every word is represented by a fixed-size vector. 10,000 is the maximum number of samples that can be used. The reviews were also converted from a string of words to a sequence of integers, with each integer representing a unique word. Even though I have obtained a list of numbers, the input of the neural network is inappropriate for these numbers. Tensors must be created from the integers. A tensor with integer data type and shape (samples, word indices) could be created from the list of integers. To achieve that, I must ensure that every sample has the same length, which means I must fill every review with dummy words (integers) to ensure that every review is the same length.

**Approach:**

For this IMDB dataset, I explored two distinct methods for producing word embeddings in this study: Pretrained word embedding layer utilizing the GloVe model and Custom-trained embedding layer.The study employed the GloVe model, a popular pretrained word embedding model trained on extensive text data corpora. It is a good option for natural language processing tasks because of its reputation for capturing the syntactic and semantic relationships between words.

In this work, I worked with the 6B version of the GloVe model, trained on a corpus of Wikipedia data and Gigaword 5, with 6 billion tokens and 400,000 words. Using the IMDB review dataset, I implemented two different embedding layers: one with a custom-trained embedding layer and the other with a pre-trained word embedding layer to assess the efficiency of various embedding techniques. I evaluated the accuracies of the two models using training sample sizes that varied from 100 to 10,000.

First, I used the IMDB review dataset to create a specially trained embedding layer. Using various dataset samples, I trained this layer, and then I used a testing set to assess each model's accuracy. Following that, I evaluated the accuracy with a model that employed a word embedding layer that had been trained in advance and tested on a range of sample sizes.

**RESULTS:**

|  |  |  |
| --- | --- | --- |
| Embedding Technique | Training Sample Size | Accuracy (%) |
| Custom-trained embedding layer | 100 | 98.7 |
| Custom-trained embedding layer | 500 | 97.7 |
| Custom-trained embedding layer | 1000 | 98.3 |
| Custom-trained embedding layer | 10000 | 97.9 |
| Pretrained word embedding (GloVe) | 100 | 100 |
| Pretrained word embedding (GloVe) | 500 | 93.2 |
| Pretrained word embedding (GloVe) | 1000 | 81.8 |
| Pretrained word embedding (GloVe) | 10000 | 92.9 |

Custom-trained embedding layer:

Based on the results I got, depending on the size of the training sample, the custom-trained embedding layer produced accuracy ranging from 97.7% to 98.7%. An accuracy of 98.7 was attained with 100 training sample size. Since the embedding layer is specifically trained for the task at hand (IMDB review sentiment classification), it is possible that this technique's high accuracy can be attributed to more effective text data representations.

It is important though, that the accuracy did not show any further improvement after a training sample size of 100, indicating that the technique may not benefit greatly from larger amounts of training data.

Pretrained word embedding layer (GloVe):

Depending on the training sample size, the accuracy achieved with the pretrained word embedding layer (GloVe) varied from 81.8% to 100%. With 100 training samples, the highest accuracy was attained. The pretrained embeddings can be useful even with little training data because they capture a large amount of the underlying semantic information in the text, which could account for the high accuracy with a small training sample size. However, the pretrained embeddings might not be as good at capturing the small details of the task at hand as the training sample size grows, which could result in decreased accuracy. Furthermore, using the pretrained embeddings with larger training sample sizes causes the model to rapidly overfit, which lowers accuracy.

These results make it challenging to say with certainty which method is the "best" to employ because it depends on the requirements and limitations of the task at hand. But in this experiment, the custom-trained embedding layer performed better overall than the pretrained word embedding layer, especially when training with larger training sample sizes. In my opinion, the pretrained word embedding layer might be a better option if computational resources are limited and a small training sample size is required but taking care to prevent overfitting.