**Assignment 4 Report**

**Names: Likitha Sree Yarabarla**

**: Sai mithra vaddepalli**

**SUMMARY**

Classifying movie reviews as good or bad is the aim of the binary classification problem utilizing the IMDB dataset. With training sample sizes of 100, 500, 1000, and 100,000 samples and validation on 10,000 samples, the dataset comprising 50,000 reviews is filtered such that only the top 10,000 words are taken into account. Reviews that are less than 150 words are not accepted. Following preprocessing, different assessment procedures are explored to find the best performance approach for classifying reviews as positive or negative. The data is then fed into an embedding layer and a pretrained embedding model.

**TECHNIQUES**

**Preprocessing of the dataset**:

Positive and negative sentiment is assigned to movie reviews in the IMDB dataset. Preprocessing entails turning every review into a series of word embeddings, with a fixed-size vector standing in for each word. Reviews are converted from word strings into sequences of integers, where each integer represents a distinct word, due to a 10,000 sample restriction. These integers must first be converted into tensors in order to be used as input into a neural network. In order to do this, a tensor with an integer data type and a shape of (word indices, samples) must be created from the integer list. All reviews are padded with dummy words (integers) to equal the length of the longest review in order to guarantee uniformity in input length.

**Approach:**

In my study on the IMDB dataset, I explored two methods of generating word embeddings: a pretrained word embedding layer using the GloVe model and a custom-trained embedding layer. The GloVe model is well-regarded in natural language processing for its ability to capture both syntactic and semantic word relationships, having been trained on vast text corpora such as Wikipedia and Gigaword 5, with 6 billion tokens and 400,000 words in its 6B version.

For the experiment, I implemented two embedding layers using the IMDB review dataset: one was custom-trained, while the other utilized the pretrained GloVe model. I assessed the effectiveness of these techniques by varying the training sample sizes from 100 to 10,000 and evaluating the accuracies of the resulting models.

Initially, I trained a custom embedding layer using the IMDB review dataset across different sample sizes and then evaluated its accuracy on a separate testing set. Subsequently, I tested a model with a pretrained word embedding layer, also evaluating its accuracy across various sample sizes to compare the performance of the two methods.

**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 findings, the custom-trained embedding layer achieved accuracy rates between 97.7% and 98.7%, with the highest accuracy of 98.7% achieved using a training sample size of 100. This high accuracy could be attributed to the embedding layer being specifically tailored for the IMDB review sentiment classification task, leading to more effective representations of the text data.

However, it's noteworthy that the accuracy did not increase beyond a training sample size of 100, suggesting that the technique might not derive significant benefits from larger amounts of training data.

Pretrained word embedding layer (GloVe):

Depending on the size of the training sample, the pretrained word embedding layer (GloVe) had an accuracy ranging from 81.8% to 100%. Using 100 training samples, the maximum accuracy of 100% was attained. Because they capture a large amount of the text's semantic information, pretrained embeddings can be useful even with a small amount of training data, which explains the high accuracy with a small training sample size.

Nevertheless, pretrained embeddings might find it more difficult to grasp the subtler aspects of the task as the training sample size grows, which could result in decreased accuracy. Furthermore, the model may rapidly overfit when pretrained embeddings with greater training sample sizes are used, which would reduce accuracy.

Determining the superior method depends on the specific task's needs and constraints, making it difficult to definitively declare one as the "best" option. However, in this study, the custom-trained embedding layer generally outperformed the pretrained word embedding layer, especially when working with larger training sample sizes. If computational resources are limited and a small training sample size is necessary, the pretrained word embedding layer could be a preferable choice, albeit with precautions to prevent overfitting.