**ADVANCED MACHINE LEARNING ASSIGNMENT 4**

**Text Data - Summary Report**

**Summary:**

This research examines how effective several techniques are in identifying the sentiment (positive/negative) of movie reviews. It uses a big IMDB dataset, limiting itself to the 10,000 most frequently occurring words. For the experiment, the sentiment classification model is trained on different data sizes (100, 500, 1,000, and 100,000 samples) and validated on a separate set of 10,000 reviews. Data is processed step by step before sending it to a pre-trained word embedding model. Through evaluation of these various approaches, the aim is to identify the most optimal method for sentiment analysis in movie reviews.

**Methods:**

**Data Preprocessing:**

The IMDB dataset labels movie reviews as positive or negative. The preprocessing step turns every review into a sequence of word embeddings, which are constant-length vector representations of prototypical words. Although the dataset is only 10,000 samples, each review is converted from text to integers (where each integer represents a word in the sequence) and this creates a sequence that is not appropriate for a neural network. Converting these integer lists is necessary because networks need tensors. A tensor with an integer data type and the shape of (samples, word indices) is made to accomplish this. Nonetheless, it is essential to make sure that every review is the same length. To provide consistency, shorter reviews are padded with fictitious integer values.

**Approach:**

To generate word embeddings for the IMDB sentiment analysis task, this study examined two methods:

Word Embedding Layer (GloVe) pre-trained: A GloVe model that had already been trained was used. By learning word representations from large text datasets, this popular approach successfully captures the syntactic and semantic links between words. It is a well-liked option for natural language processing tasks because of its effectiveness.

Custom-trained Embedding Layer: In this method, an embedding layer is specially learned for the IMDB dataset. This enables the model to perhaps pick up word representations specific to the film review domain.

In this study, the efficacy of various word embedding methods for sentiment analysis on film reviews was compared. Gigaword 5 and Wikipedia were used to train a pre-trained GloVe model (6B tokens, 400k words) to generate word embeddings. To evaluate these methods, the model was built with two distinct embedding layers: the pre-trained GloVe layer and a custom-trained layer tailored to the IMDB data. From 100 to 10,000 samples, different training data sizes were used to assess the models' performance.

I used the IMDB dataset to train a custom embedding layer and compare its accuracy on different sample sizes to a model that used a pre-trained embedding layer evaluated across different amounts of training data to examine the efficacy of word embeddings.

**Custom-trained embedding layer**

Using training sample 100, a specially trained embedding layer

A graph with blue dots

Description automatically generatedA graph with red dots

Description automatically generated

Layer of custom-trained embeddings using 500 training samples

A graph with blue dots

Description automatically generated A graph of training and validation loss

Description automatically generated

An embedding layer that has been specially trained using 1,000 training samples

A graph with blue dots

Description automatically generated A graph with red dots

Description automatically generated

Custom-trained embedding layer using 10,000 training samples

A graph with a line graph

Description automatically generated A graph with red dots

Description automatically generated

**Pretrained word embedding layer (GloVe)**

Pretrained word embedding layer (GloVe) with training sample 100

A graph with blue lines

Description automatically generatedA graph with red lines and dots

Description automatically generated

Pretrained word embedding layer (GloVe) with training sample 500

A graph with blue dots

Description automatically generated A graph with red dots and a line

Description automatically generated

Pretrained word embedding layer (GloVe) with training sample 1000

A graph with blue dots

Description automatically generated A graph with red lines and numbers

Description automatically generated

Pretrained word embedding layer (GloVe) with training sample 10000

A graph with blue dots

Description automatically generated A graph with red dots

Description automatically generated

**Results:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **Embedding Technique** | **Training sample size** | **Loss and Accuracy on Test** | **Accuracy** |
| **1** | Custom-trained embedding layer | 100 | **Loss: 0.694**  **Accuracy:0.498** | **98.7** |
| **2** | Custom-trained embedding layer | 500 | **Loss:0.691**  **Accuracy: 0.525** | **97.0** |
| **3** | Custom-trained embedding layer | 1000 | **Loss:0.679**  **Accuracy:0.568** | **97.5** |
| **4** | Custom-trained embedding layer | 10000 | **Loss:0.340**  **Accuracy:0.568** | **97.6** |
| **5** | Pretrained word embedding layer (GloVe) | 100 | **Loss:0.938**  **Accuracy:0.495** | **100** |
| **6** | Pretrained word embedding layer (GloVe) | 500 | **Loss: 0.905**  **Accuracy:0.501** | **99.2** |
| **7** | Pretrained word embedding layer (GloVe) | 1000 | **Loss:0.989**  **Accuracy:0.501** | **96.2** |
| **8** | Pretrained word embedding layer (GloVe) | 10000 | **Loss:1.305**  **Accuracy:0.501** | **81.9** |

**Custom-trained embedding layer:**

Our analysis showed that the performance of the custom-trained embedding layer peaked at a training sample size of just 100, and it attained remarkable accuracy ranging from 97.0% to 98.7%. This better performance is probably due to the layer's capacity to pick up domain-specific word representations that are extremely pertinent to the sentiment classification task of IMDB reviews. Remarkably, the accuracy reached a peak at 100 samples, indicating that a large volume of training data might not be necessary for the custom layer to function at its best.

**Pretrained word embedding layer (GloVe):**

Our analysis showed that even with a small amount of training data, pre-trained GloVe embeddings performed well (81.9% to 100% accuracy), reaching maximal accuracy with just 100 samples. Their capacity to capture the rich semantic information included in language is probably the cause of this. The pre-trained embeddings could not have been able to capture the subtleties unique to the sentiment analysis task of movie reviews, though, as the size of the training data increased. This could have resulted in overfitting with very big datasets and a decline in performance.

The results of the study of the "optimal" embedding method depend on the circumstance. Although the custom-trained embedding layer performed better overall, especially when dealing with bigger datasets, the pre-trained GloVe embeddings are preferred due to limited training data and resource constraints. To avoid overfitting, however, cautious mitigation techniques are required when using pre-trained embeddings with bigger datasets.

**Conclusion:**

Text categorization neural network models require careful assessment of pretrained networks, embeddings, and training set size to achieve outstanding performance. The useful information provided by these variables can be used to improve and modify models in various situations. Pretrained networks—particularly those trained on large datasets—allow the model to grasp complex linguistic patterns and semantic representations. Having been extracted from large amounts of textual input, these learned representations are useful for tasks such as text categorization. To provide context and word meaning to machine learning models, embeddings are required. Through the facilitation of a more intricate comprehension of the connections among words, they improve the model's text classification skills. The training set's size has a significant impact on a model's ability to generalize.

The results demonstrated that the simple embedding layer model performed better than the pre-trained model, even though pre-trained embeddings are believed to enhance model performance. Remembering that the pre-trained model in this instance is not optimal for the task at hand and did not improve the embeddings during training is frequently very important. Performance may be enhanced primarily by optimizing the embeddings.

In conclusion, because these results are based on a small number of training samples and a restricted set of hyperparameters, we should exercise caution when extrapolating them. Altering the hyperparameters or adding more training data may provide different outcomes.