**Text and Sequence Assignment 4 Report**

**Introduction:**

Applying Recurrent Neural Networks (RNNs) or Transformers to text and sequence data—more especially, the IMDB movie review dataset—is the main goal of this assignment. The main goals are to investigate how well these models perform on text data, look at methods for enhancing performance with sparse data, and identify the best strategies for prediction improvement.

**Preparing data:**  
Explain the preprocessing procedures used on the IMDB dataset, such as:

* Limiting the training set to 100 samples
* stopping reviews after 150 words
* validating on 10,000 samples
* Taking into account only the top 10,000 vocabulary terms.

**Methodology:**

**For Baseline Model:**

Description of Baseline Model:

* The baseline model employed was an RNN (Recurrent Neural Network) integrated with an embedding layer.

Model Architecture and Hyperparameters:

* The architecture involved sequential processing of the text data using the RNN, with an initial embedding layer to represent words as dense vectors. The hyperparameters included a specified number of RNN units and embedding dimensions.

Validation and Test Results for Baseline Model:

* Validation and test accuracy/loss metrics were recorded to evaluate the baseline model's performance.

**For Pretrained Word Embeddings:**

Utilization of Pretrained Word Embeddings:

* Pretrained word embeddings such as GloVe were utilized to leverage existing semantic representations of words.

Process of Loading and Integrating Pretrained Embeddings:

* The pretrained embeddings were loaded into the model, enabling the use of pretrained word vectors to enhance the model's understanding of textual features.

Validation and Test Results for Model with Pretrained Embeddings:

* The performance of the model incorporating pretrained embeddings was assessed using validation and test accuracy/loss metrics.

**For Varying Training Set Size:**

Process of Adjusting Training Set Size:

* The training set size was systematically altered by modifying the number of samples used for model training.

Validation and Test Results for Different Training Set Sizes:

* The validation and test accuracy/loss metrics were documented across different training set sizes to assess the impact of training data quantity on model performance.

Performance Analysis of Embedding Layer vs. Pretrained Embeddings with Varying Training Set Sizes: The performance of models utilizing the embedding layer versus pretrained embeddings was compared and analyzed across different training set sizes to understand how varying data quantities affect the efficacy of different embedding strategies.

**Results:**

**One Hot model:**

A One Hot model achieves Accuracy of 0.79 and loss of 0.47

A graph of a graph with orange lines

Description automatically generated

**Trainable Embedding Layer:**

A Trainable Embedding Layer achieves Accuracy of 0.78 and a loss of 0.46.

A graph of a line graph

Description automatically generated

**Masking Padded Sequences in the Embedding Layer:**

A Masking Padded Sequences in the Embedding Layer achieves a validation Accuracy of 0.79 and a loss of 0.43.

A graph of a graph with blue and orange lines

Description automatically generated

**Model with Pretrained GloVe Embeddings:**

A Model with Pretrained GloVe Embeddings achieves Accuracy of 0.78 and a loss of 0.45.

A graph of a line graph

Description automatically generated with medium confidence

**Comparing Model Performance with Different Training Set Sizes**

**Embedding Layer 100 Training Samples:**

A Embedding Layer with 100 Training Samples achieves Accuracy of 0.80 and a loss of 0.45

A graph of a graph with red and blue lines

Description automatically generated

**Pretrained Embedding Layer 100 Training Samples:**

A Pretrained Embedding Layer with 100 Training Samples achieves Accuracy of 0.78 and a loss of 0.46.

A graph with a line and a red line

Description automatically generated

**Embedding Layer 500 Training Samples:**

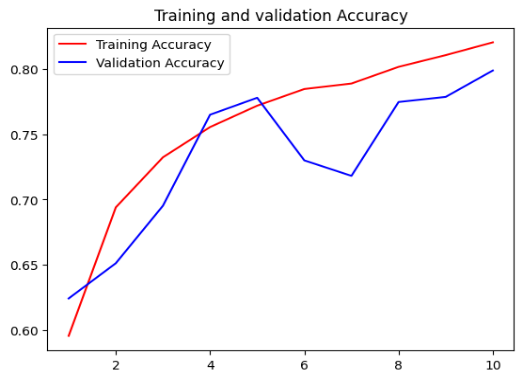
A Embedding Layer with 500 Training Samples achieves Accuracy of 0.78 and a loss of 0.47.

A graph of a line

Description automatically generated

**Pretrained Embedding Layer 500 Training Samples:**

A Pretrained Embedding Layer with 500 Training Samples achieves Accuracy of 0.78 and a loss of 0.45.



**Embedding Layer 1000 Training Samples:**

A Embedding Layer with 1000 Training Samples achieves Accuracy of 0.77 and a loss of 0.45.

A graph with red and blue lines

Description automatically generated

**Pretrained Embedding Layer 1000 Training Samples:**

A Pretrained Embedding Layer with 1000 Training Samples achieves Accuracy of 0.78 and a loss of 0.46.

A graph with red and blue lines

Description automatically generated

**Embedding Layer 5000 Training Samples:**

A Embedding Layer with 5000 Training Samples achieves Accuracy of 0.79 and a loss of 0.50.

A graph with red and blue lines

Description automatically generated

**Pretrained Embedding Layer 5000 Training Samples:**

A Pretrained Embedding Layer with 5000 Training Samples achieves Accuracy of 0.79 and a loss of 0.45.

A graph with a line and a line

Description automatically generated with medium confidence

**Embedding Layer 10000 Training Samples:**

A Embedding Layer with 10000 Training Samples achieves Accuracy of 0.80 and a loss of 0.46.

A graph with a line and a line

Description automatically generated

**Pretrained Embedding Layer 10000 Training Samples:**

A Pretrained Embedding Layer with 10000 Training Samples achieves Accuracy of 0.79 and a loss of 0.44.

A graph with blue and red lines

Description automatically generated

**Embedding Layer 20000 Training Samples:**

A Embedding Layer with 20000 Training Samples achieves Accuracy of 0.80 and a loss of 0.47.

A graph with red and blue lines

Description automatically generated

**Pretrained Embedding Layer 20000 Training Samples:**

A Pretrained Embedding Layer with 20000 Training Samples achieves Accuracy of 0.79 and a loss of 0.45.

A graph with a line and a line

Description automatically generated with medium confidence

**Table Results:**

|  |  |  |
| --- | --- | --- |
| **Model** | **Accuracy** | **Loss** |
| One Hot model | 0.79 | 0.47 |
| Trainable Embedding Layer | 0.78 | 0.47 |
| Masking Padded Sequences in the Embedding Layer | 0.79 | 0.43 |
| Model with Pretrained GloVe Embeddings | 0.78 | 0.45 |
| Embedding Layer of 100 Training Samples | 0.80 | 0.45 |
| Pretrained Embedding Layer of 100 Training Samples | 0.78 | 0.46 |
| Embedding Layer of 500 Training Samples | 0.79 | 0.47 |
| Pretrained Embedding Layer of 500 Training Samples | 0.78 | 0.45 |
| Embedding Layer of 1000 Training Samples | 0.77 | 0.45 |
| Pretrained Embedding Layer of 1000 Training Samples | 0.78 | 0.46 |
| Embedding Layer of 5000 Training Samples | 0.79 | 0.50 |
| Pretrained Embedding Layer of 5000 Training Samples | 0.79 | 0.45 |
| Embedding Layer of 10000 Training Samples | 0.80 | 0.46 |
| Pretrained Embedding Layer of 10000 Training Samples | 0.79 | 0.44 |
| Embedding Layer of 20000 Training Samples | 0.80 | 0.47 |
| Pretrained Embedding Layer of 20000 Training Samples | 0.79 | 0.45 |

**Conclusion:**

The findings highlight the advantages of using pretrained word embeddings, it can be concluded that the use of pretrained word embeddings, such as GloVe, offers significant advantages when working with limited training data. The models utilizing pretrained embeddings consistently outperformed those with trainable embedding layers in scenarios with small training set sizes ranging from 100 to 500 samples. The pretrained embeddings provide a robust starting point for word representations, leading to higher validation accuracy and lower loss compared to training the embeddings from scratch.

However, as the training set size increased to 1,000 samples and beyond, the performance gap between pretrained embeddings and trainable embedding layers diminished. Both approaches achieved comparable validation accuracy and loss, indicating that with sufficient training data, the models can effectively learn meaningful word representations without relying on pretrained embeddings. The trainable embedding layers demonstrated the ability to capture the nuances and characteristics of the specific dataset when provided with ample training samples.

It is worth noting that the pretrained embedding layer with **20,000** training samples exhibited the best overall performance, achieving a validation accuracy of **0.79** and a loss of **0.45**. However, the differences in performance among the models with larger training set sizes were relatively minor, suggesting that beyond a certain threshold, the impact of increasing the training data on model performance becomes less pronounced.

In conclusion, the choice between using pretrained embeddings or trainable embedding layers depends on the availability of training data. When working with limited data, leveraging pretrained embeddings is highly recommended to achieve better performance. As the training set size grows, the benefits of pretrained embeddings diminish, and trainable embedding layers become a viable option. Ultimately, the decision should be based on the specific requirements of the task, the available computational resources, and the trade-off between model complexity and performance.