**ASSIGNMENT 4  
Text and Sequence**

I used the IMDB dataset, which has 50,000 reviews (25,000 good and 25,000 negative), for this task. After preprocessing the dataset, which was split into 25,000 reviews for training and 25,000 reviews for testing, I used a bidirectional LSTM model to do sentiment analysis. I limited the training samples to 100 and set the maximum review length at 150 words to compare pre-trained models and embedding layers. Furthermore, I used 10,000 samples to validate the model's performance and only took the top 10,000 phrases into account. To find the most successful method, the pre-trained model's and the embedding layer's efficacy were compared. To ascertain the ideal quantity required for peak performance, I also experimented with the amount of training samples.

In conclusion, I explored sentiment analysis using pre-trained models and embedding layers in the IMDB dataset. By changing the quantity of training samples, considering the top 10,000 words, and validating on 10,000 samples, the effectiveness of each strategy was examined.

I first trained a simple sequence model on the IMDB dataset. The main conclusions from this method are as follows:

To provide a baseline for the assignment, I built a first simple sequence model and compared its results to those of other models. The results show that the model seems to have successfully learned to fit the training set. Overfitting to the validation set is a possibility, though, because the test accuracy is higher than the validation accuracy. This is a good result because it shows that the model worked well on data that was unknown before.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model-1** | **Training**  **Accuracy** | **Training**  **Loss** | **Validation**  **Accuray** | **Validation**  **Loss** | **Test**  **Accuracy** |
| First Basic Sequence  Model | 1.00 | 0.5453 | 0.50 | 0.6827 | 0.498 |

I started from scratch and built a new model that utilizes word embedding. I deliberately turned off masking to test the model's functionality. The findings demonstrate that while the second model, which incorporated word embedding, outperformed the simple sequence model in terms of training accuracy, it obtained a considerably lower total accuracy. Moreover, the second model had a higher validation loss but a lower training loss. These results suggest that the model does not generalize well to new, previously unknown data and is overfit to the training set. When masking is turned off, the model can be learning from padded values that are devoid of meaningful information, which is a common outcome. To sum up, the information suggests that

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model- 2** | **Training**  **Accuracy** | **Training**  **Loss** | **Validation**  **Accuray** | **Validation**  **Loss** | **Test**  **Accuracy** |
| Embedding layer from  scratch | 0.9827 | 0.1181 | 0.822 | 0.4019 | 0.8319 |

**CUSTOM-TRAINED EMBEDDING LAYER**

1. Custom-trained embedding layer with training sample size = 100

A graph with a line graph

Description automatically generatedA graph of training and validation loss

Description automatically generated

2. Custom-trained embedding layer with training sample size = 5000

A graph showing training and validation

Description automatically generatedA graph showing the performance of training

Description automatically generated

3. custom-trained embedding layer with training sample size = 2500

A graph showing the performance of training

Description automatically generatedA graph of training and validation

Description automatically generated

4. custom-trained embedding layer with training sample size = 10000

A graph showing the growth of training

Description automatically generatedA graph of training and validation

Description automatically generated

With the custom-trained embedding layer, the validation accuracy ranged from 50% to 82%, depending on the size of the training sample. The best accuracy was obtained with a training sample size of 5000.

**PRETRAINED WORD EMBEDDING LAYER**

1. pretrained word embedding layer with training sample size = 100

A graph showing the performance of training

Description automatically generated

A graph showing the line of training and validation

Description automatically generated

2. pretrained word embedding layer with training sample size = 5000

A graph showing the line of a training

Description automatically generated with medium confidence

A graph showing the loss of training

Description automatically generated

3. pretrained word embedding layer with training sample size = 2500

A graph showing the growth of a training

Description automatically generated

A graph showing the growth of training and validation

Description automatically generated

1. pretrained word embedding layer with training sample size = 10000

A graph with a line

Description automatically generated

A graph with a line graph

Description automatically generated

The pretrained word embedding layer (GloVe) exhibited varying degrees of accuracy, from 48%

to 54%, depending on the size of the training sample. Most accurate result was obtained with100 training samples. In addition, the model quickly overfits when using the pretrained embeddings with bigger training sample sizes, which reduces accuracy. Because it depends on the needs and constraints of the task at hand, these findings make it difficult to determine which approach is the "best" to adopt with confidence.

# Results:

|  |  |  |  |
| --- | --- | --- | --- |
| **Embedding Technique** | **Training Sample Size** | **Training Accuracy (%)** | **Test loss** |
| Custom-trained embedding layer | 100 | 100 | 0.69 |
| Custom-trained embedding layer | 5000 | 98.2 | 0.37 |
| Custom-trained embedding layer | 2500 | 98.6 | 0.50 |
| Custom-trained embedding layer | 10000 | 98.7 | 0.469 |
| Pretrained word embedding (GloVe) | 100 | 100 | 0.81 |
| Pretrained word embedding (GloVe) | 5000 | 70.7 | 0.74 |
| Pretrained word embedding (GloVe) | 2500 | 95.4 | 0.52 |
| Pretrained word embedding (GloVe) | 10000 | 87.9 | 0.49 |

**Conclusion:** In the current study, however, the custom-trained embedding layer performed better than the pretrained word embedding layer, especially when training with a higher number of training samples. If the training sample size is small and computer resources are restricted, the pretrained word embedding layer might be a "better choice" despite the risk of overfitting.