**ASSIGNMENT 4**

**WORD EMBEDDING USING EMBEDDING LAYERS AND A PRE-TRAINED MODEL**

I have taken the IMDB dataset to perform sentiment analysis in this assignment, and I have examined how word embedding works using a pre-trained model and using embedding layers. Specifically, I have done the following: -

1. Cutoff reviews after 150 words.

2. Restrict training samples to 100.

3. Validate 10,000 samples.

4. Consider only the top 10,000 words.

5. Consider both an embedding layer and a pre-trained word embedding. Which approach did better? Now try changing the number of training samples to determine at what point the embedding layer gives better performance.

Let’s discuss each case separately.

The findings are as follows: -

1. Training on a first basic sequence model: -

In this, I have trained the dataset on a first basic sequence model. The findings are as follows: -

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Training Accuracy | Validation Accuracy | Training Loss | Validation Loss | Test Accuracy |
| Model 1 First Basic  Sequence Model | 0.7180 | 0.9588 | 0.1380 | 0.4928 | 0.811 |

The initial sequence model was developed to set a foundational performance benchmark for the task and to serve as a point of reference for comparison with other models. Based on the obtained results, it appears that the model effectively captures the patterns in the training data, but there is a possibility of overfitting to the validation set. Notably, the test accuracy surpasses the validation accuracy, suggesting that the model demonstrates strong performance on previously unseen data.

1. Training a Model that uses word embedding from scratch: -

In this model, I have implemented word embeddings without enabling masking to assess the model's performance. The outcomes of this investigation are outlined below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Training Accuracy | Validation Accuracy | Training Loss | Validation Loss | Test Accuracy |
| Model 2 –  Embedding layer from scratch | 0.9834 | 0.8135 | 0.0458 | 0.7657 | 0.791 |

The second model, utilizing word embeddings, demonstrated superior training accuracy but a slightly diminished overall accuracy compared to the initial basic sequence model. While the training loss was lower than that of the first model, the validation loss was higher, indicating a potential overfitting issue where the model struggles to generalize effectively to new data. This situation is often associated with the absence of masking, as the model may inadvertently learn from padded values that lack meaningful information. In summary, the findings suggest that building word embeddings from scratch can enhance training accuracy, but it is crucial to incorporate proper regularization techniques, such as enabling masking, to mitigate overfitting.

1. Training a Model that uses word embedding from scratch with Masking enabled: -

In this model, I have created a model that is like the 2nd model, but the only modification is that I Have enabled masking. The findings are as follows: -

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Training Accuracy | Validation Accuracy | Training Loss | Validation Loss | Test Accuracy |
| Model 3–  Embedding layer from  scratch with  Masking enabled | 0.9909 | 0.7685 | 0.0273 | 0.8431 | 0.791 |

In this model, a new component called masking has been incorporated. Masking involves the model disregarding padded zeros and concentrating solely on the actual input data. While the training accuracy is slightly lower compared to the second model, there is a notable enhancement in validation accuracy. This indicates that the model equipped with masking is more adept at handling sequences of variable lengths compared to its predecessor. The results underscore the significance of considering masking as a crucial feature when employing word embedding.

1. Using a Pre-trained word embedding: -

This model is trained using pre-trained word embedding. We have used the Glove word embedding to see how the model performs. The findings are as follows: -

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Training Accuracy | Validation Accuracy | Training Loss | Validation Loss | Test Accuracy |
| Model 4–  Pretrained  Word  Embedding | 0.8037 | 0.7665 | 0.4290 | 0.5203 | 0.756 |

Based on the findings presented above, it can be concluded that the utilization of pre-trained word embeddings, such as Glove, does not consistently lead to improved performance. The training accuracy of this model is relatively inferior compared to all preceding models, suggesting inadequate learning of the data's features. This discrepancy may arise from the pre-trained model being trained on a different corpus, potentially failing to capture the nuances and contextual intricacies of the language specific to the dataset. In summary, although pre-trained embeddings can prove beneficial in certain situations, it is crucial to explore various embeddings or fine-tune them to optimize performance.

1. Using different Training samples: -

Finally, I have tried using different training samples to determine the point where the embedding layer performs the best. The findings for different training samples are as follows: -

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Training Accuracy | Validation Accuracy | Training Loss | Validation Loss | Test  Accuracy |
| 1000 Training  Samples | 0.9865 | 0.7970 | 0.0447 | 0.9757 | 0.821 |
| 5000 Training  Samples | 0.9886 | 0.8240 | 0.0392 | 0.8162 | 0.819 |
| 10000 Training  Samples | 0.9884 | 0.8280 | 0.0425 | 0.7208 | 0.824 |
| 15000 Training  Samples | 0.9891 | 0.8050 | 0.0366 | 0.8864 | 0.821 |
| 20000 Training  Samples | 0.9919 | 0.8090 | 0.0272 | 0.7787 | 0.806 |
| 25000 Training  Samples | 0.9880 | 0.8300 | 0.0382 | 0.6963 | 0.823 |

From the above table, we can infer that the performance of the model does not necessarily improve by increasing the training samples. For example, the model trained with 5000 samples has a lower validation accuracy and higher validation loss compared to the model trained with 1000 samples. This could be due to overfitting or the noise present in the additional training data.

**Findings and Conclusion: -**

* The initial basic sequence model achieved good performance on test data, demonstrating its ability to generalize well to unseen data.
* Training a model using word embedding from scratch without masking led to overfitting to the training data, suggesting the need for proper regularization techniques like masking.
* The model with word embedding from scratch with masking outperformed the one without masking, emphasizing the importance of masking during embedding.
* The model with word embedding from scratch with masking demonstrated superior performance compared to the pre-trained word embedding model.
* Pre-trained word embedding may not always guarantee better performance.
* Increasing the training samples doesn't guarantee better performance, as evidenced by the model trained with 5000 samples.
* While the training loss decreased with increasing training samples, indicating improved model learning with more data, the validation loss increased, suggesting poor generalization on unseen data.
* Proper regularization techniques like masking and experimenting with different embeddings or fine-tuning the embedding can enhance model performance.
* Regularization techniques like masking and exploring different embeddings can improve model performance.
* The optimal number of training samples is 15000, as the model with 15000 training samples achieved the best performance.

In conclusion, we can say that despite variations in the models, their performances remained remarkably similar, indicating that no single approach is universally optimal for embedding layers. Selecting the most appropriate model for a given dataset requires a thorough evaluation of various models. To enhance performance, we can leverage regularization techniques, explore different embedding sizes and LSTM layers, adapt pre-trained embeddings to the specific dataset, and increase the training sample size to an appropriate extent.