ASSIGNMENT 4  
Text and Sequence

➢ For this assignment, I utilized the IMDB dataset, which comprises 50,000 reviews (25,000 positive and 25,000 negative). The dataset was divided into 25,000 reviews for training and 25,000 for testing, and I performed sentiment analysis using a bidirectional LSTM model after preprocessing the data.

➢ To conduct a comparison between pre-trained models and embedding layers, I set the maximum review length at 150 words and limited the training samples to 100. Additionally, I validated the model's performance on 10,000 samples and considered only the top 10,000 words.

➢ The effectiveness of the pre-trained model and embedding layer was evaluated to determine which approach worked best. I also varied the number of training samples to determine the optimum number needed for optimal performance.

In summary, I explored the use of pre-trained models and embedding layers for sentiment analysis in the IMDB dataset. The performance of each approach was analyzed by varying the number of training samples, considering the top 10,000 words, and validating on 10,000 samples.

Top of Form

Regenerate response

Bottom of Form

➢ To start with, I trained the IMDB dataset on a basic sequence model. Here are the key findings from this approach:

➢ I constructed a first simple sequence model and compared its findings to those of other models to set a baseline performance for the assignment. According to the findings, the model appears to have effectively trained to suit the training data. However, because the test accuracy is greater than the validation accuracy, there is a risk of overfitting to the validation set. This indicates that the model performed well on previously unknown data, which is a desirable outcome.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model-1 | Training  Accuracy | Training  Loss | Validation  Accuray | Validation  Loss | Test  Accuracy |
| First Basic Sequence  Model | 0.9728 | 0.1098 | 0.8086 | 0.4934 | 0.848 |

➢ I created a new model from scratch that makes use of word embedding. I purposefully disabled masking to see how the model would function. The results show that the second model, which included word embedding, performed better in terms of training accuracy but obtained somewhat lower total accuracy than the simple sequence model. Furthermore, the second model had a reduced training loss but a greater validation loss. These findings imply that the model is overfitting to the training data and does not generalize well to new, previously unknown data. This is a common result when masking is disabled, as the model may be learning from padded values that contain no significant information. In conclusion, the data indicate that

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model- 2 | Training  Accuracy | Training  Loss | Validation  Accuray | Validation  Loss | Test  Accuracy |
| Embedding layer from  scratch | 0.9899 | 0.0478 | 0.7780 | 0.7052 | 0.799 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model-3 | Training Accuracy | Training Loss | Validation Accuracy | Validation Loss | Test Accuracy |
| Embedding layer from scratch with  Masking enabled | 0.9890 | 0.0497 | 0.7999 | 0.5629 | 0.799 |

➢ I developed a new model from scratch, similar to the second model, but with one crucial addition: masking is enabled. After evaluation, the findings indicate that this model performs better than the previous one due to the addition of masking. Masking enables the model to ignore padded zeros and solely focus on the relevant input data. Although the training accuracy is slightly lower than that of the second model, the validation accuracy has significantly improved. This suggests that the model can handle sequences of varying lengths more effectively than the previous model. Overall, these results demonstrate that masking is a crucial feature to consider when using word emb

➢ In this model, we evaluated the influence of pre-trained word embedding, especially the GloVe word embedding, on performance. The findings of the examination indicate that employing pre-trained word embedding, such as GloVe, may not necessarily result in superior results. This model's training accuracy was worse than that of all prior models, suggesting that the model did not properly learn the data's characteristics. This might be because the pre-trained model was trained on a different corpus and did not catch the intricacies and context of the dataset's individual language. While pre-trained embedding can be useful in some situations, it is critical to experiment with different embeddings or fine-tune the embedding to increase performance.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Training  Accuracy | Training  Loss | Validation  Accuracy | Validation  Loss | Test  Accuracy |
| Model 4– Pretrained Word  Embedding | 0.8216 | 0.4096 | 0.7781 | 0.4815 | 0.7796 |

.

➢ I conducted an experiment to evaluate the impact of different training samples on the embedding layer's performance. After evaluating various training samples, the results indicate that increasing the training samples does not always lead to improved model performance. For instance, the model trained with 5000 samples demonstrated lower validation accuracy and higher validation loss compared to the model trained with only 1000 samples. This outcome may be attributed to overfitting or noise present in the additional training data.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Training Accuracy | Training Loss | Validation Accuracy | Validation Loss | Test  Accuracy |
| Training Samples  1000 | 0.9886 | 0.0346 | 0.8124 | 0.6398 | 0.803 |
| Training Samples  5000 | 0.9916 | 0.0372 | 0.7215 | .309865 | 0.808 |
| Training Samples  10000 | 0.9819 | 0.0316 | 0.8203 | 0.7756 | 0.860 |
| Training Samples  15000 | 0.9928 | 0.0267 | 0.8092 | 0.7832 | 0.802 |
| Training Samples  20000 | 0.9899 | 0.0338 | 0.8087 | 0.6522 | 0.800 |
| Training Samples  25000 | 0.9935 | 0.0212 | 0.8125 | 0.76865 | 0.803 |

.

To provide a comprehensive overview, I compiled a table summarizing the findings of all models. The table is as follows: -

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Training  Accuracy | Training Loss | Validation  Accuracy | Validation Loss | Test Accuracy |
| 1Model First Basic Sequence  Model | 0.9726 | 0.1097 | 0.8080 | 0.4928 | 0.820 |
| 2Model Embedding layer from  scratch | 0.9889 | 0.0459 | 0.7760 | 0.7038 | 0.798 |
| 3 Model Embedding layer from scratch with  Masking enabled | 0.9887 | 0.0488 | 0.7998 | 0.5622 | 0.796 |
| 4 Model Pretrained Word  Embedding | 0.8223 | 0.4098 | 0.7785 | 0.4828 | 0.774 |
| 5Model  Embedding layer with  Training Samples 1000 | 0.9881 | 0.0335 | 0.8138 | 0.6398 | 0.802 |
| Embedding layer with  Training Samples 5000 | 0.9924 | 0.0369 | 0.7212 | 0.30941 | 0.803 |
| Embedding layer with  Training Samples 10000 | 0.9897 | 0.0314 | 0.8208 | 0.77442 | 0.825 |
| Embedding layer with  Training Samples 15000 | 0.9947 | 0.2637 | 0.8087 | 0.7831 | 0.812 |
| Embedding layer with  Training Samples 20000 | 0.9889 | 0.0333 | 0.8099 | 0.6519 | 0.8000 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Embedding layer with  Training Samples 25000 | 0.9936 | 0.0218 | 0.8137 | 0.7682 | 0.812 |

➢Evaluation of Sequence Models for Natural Language Processing

In the context of Natural Language Processing, sequence models are essential in understanding and processing sequential data. In this study, several sequence models were evaluated to determine their performance in processing text data. The following are the key findings:

➢ Good Generalization of Basic Sequence Model

The initial basic sequence model produced satisfactory results on the test data, indicating that the model was capable of generalizing well to unseen data.

➢ Importance of Proper Regularization Techniques

Training a model with word embedding from scratch but without masking led to overfitting on the training data. This emphasizes the importance of implementing proper regularization techniques, such as masking.

➢ Improved Performance with Masking

The model with word embedding from scratch with masking performed better than the one without, highlighting the significance of considering masking while embedding.

➢ Better Performance with Scratch Word Embedding and Masking

Surprisingly, the model with word embedding from scratch with masking outperformed the pre-trained word embedding, suggesting that pre-trained word embedding does not necessarily result in superior performance.

No Guarantee of Better Performance with Increased Training Samples

Increasing the number of training samples did not necessarily guarantee better performance, as demonstrated by the model trained with 5000 samples.

➢ Learning Improvement with Increased Training Samples

However, the training loss decreased with an increase in the number of training samples, indicating that the model was able to learn better with more data.

Challenges of Generalization with Increased Training Samples

On the other hand, the validation loss increased with an increase in training samples, which suggests that the model was unable to generalize well to unseen data.

Importance of Proper Regularization Techniques and Experimentation

Proper regularization techniques, such as masking, and experimenting with different embeddings or fine-tuning the embedding can improve the performance of the model.

Optimal Number of Training Samples

The model with 15000 training samples performed the best, and thus, this can be considered the optimal number of training samples.

➢Conclusion: No One-Size-Fits-All Approach

Finally, the tested models performed similarly, indicating that there is no one-size-fits-all method to embedding layers. To find the best model for a particular dataset, many models should be tried. Regularization techniques, experimenting with alternative embedding sizes and LSTM layers, training pre-trained embeddings on the provided dataset, and increasing the training sample to a particular point can all be used to improve performance.