|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.no** | **Training Set** | **Validation Set** | **Test Set** | **Embedding layer** | **Pretrained Network** | **Loss and Accuracy on Test** |
| 1. | 100 | 10000 | 5000 | Yes | Yes | loss: 0.7597 - acc: 0.5907 |
| 2. | 1000 | 10000 | 5000 | Yes | Yes | loss: 2.4553 - acc: 0.4680 |
| 3. | 3000 | 10000 | 5000 | Yes | Yes | loss: 1.2912 - acc: 0.7457 |
| 4. | 8000 | 10000 | 5000 | Yes | Yes | loss: 0.9858 - acc: 0.8003 |
| 5. | 10000 | 10000 | 5000 | Yes | Yes | loss: 0.7565 - acc: 0.8598 |

**FINDINGS:**

* All the above models have been run with the following parameters to compile the model: optimizer**=**'adam', loss**=**'binary\_crossentropy,’ metrics**=**['acc']).
* A pre-trained network is trained on a vast dataset, usually an extensive image classification dataset. When the original dataset is sizable and diverse, the knowledge acquired by the network can be applied to other image classification tasks, even if these tasks are distinct from the original dataset. A pre-trained network can serve as a foundation for various image classification challenges.
* The base model considered all the samples from the IMDB dataset, with maximum words restricted = 150 and only the top 10000 samples. The below plots show that validation loss peaked at 0.2979 and validation accuracy at 0.8742 on the 4th epoch.

A picture containing text, screenshot, line, plot

Description automatically generated A picture containing text, screenshot, line, plot

Description automatically generated

* The test results on the same dataset showed a loss of 0.7067 accuracy of 0.8428.
* For the above models, the max\_words and max\_features were set to 10,000 and 150, respectively.
* The model has no additional parameters to find the optimal training sample size for the current dataset. The base models in training samples of 100, 1000, 3000, 8000, and 10000 showed accuracies being increased as the training samples increased.
* The base model of 100 training samples below results in the validation loss increased gradually from 0.70 to 0.80. The validation accuracy was very at 0.56. This shows that more than training samples were needed to train the model. The test accuracy, too, showed a very low of 0.5907.

A picture containing text, screenshot, line, plot

Description automatically generatedA picture containing text, screenshot, plot, line

Description automatically generated

* To verify if the training samples increase will yield better results, the samples were changed in increments of 1000, 3000, 8000, and 10000. The results of the validation accuracy and loss are as follows:
  + 1000 samples- Validation loss: 1.41; Validation accuracy: 0.50
  + 3000 samples- Validation loss: 0.69; Validation accuracy: 0.79
  + 8000 samples- Validation loss: 0.70; Validation accuracy: 0.84
  + 10000 samples- Validation loss: 0.78; Validation accuracy: 0.85
* Finally, the training samples of the 10000 yielded a test loss of 0.7565 and a test accuracy of 0.8598.

A picture containing text, screenshot, line, plot

Description automatically generatedA picture containing text, screenshot, line, plot

Description automatically generated

**Conclusion:**

The effectiveness of an embedding layer versus a pre-trained word embedding for text and sequence tasks depends on several factors, such as the size of the training data, domain-specific vocabulary, and the nature of the task involved. Therefore, assessing each approach's strengths and limitations is crucial before deciding which is best suited for a particular natural language processing task. The purpose of training samples in deep learning for texts and sequences is to teach the neural network to recognize patterns and relationships within the data. In supervised learning, the training samples consist of input data (e.g., text sequences) and corresponding target labels (e.g., sentiment labels for sentiment analysis). The neural network is trained on these examples, adjusting the model's parameters to minimize the difference between predicted and target outputs. By providing the model with a large and diverse set of training samples, the neural network can learn to identify important features and patterns within the data, making accurate predictions on new, unseen data. In addition, using training samples can help prevent overfitting, a common problem in deep learning where the model becomes too specialized to the training data and performs poorly on new data. Overall, the quality and quantity of training samples are critical factors in the success of a deep learning model for texts and sequences. The more high-quality training data available, the better the model can learn to generalize and make accurate predictions on new data.