Text And Sequence Assignment

**Objective:**

Developing a model for predicting whether a review is negative or positive based on the IMDb reviews dataset and learning the reviews' features.

**By using different sample sizes:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model number | Training | Validation | Test | Performance on test set (Loss, Accuracy) |
| Model 2 | 100 | 10000 | 5000 | 0.692,0.515 |
| Model 3 | 1000 | 10000 | 5000 | 0.566,0.729 |
| Model 4 | 25000 | 10000 | 5000 | 0.394,0.891 |
| Model 5 | 35000 | 10000 | 5000 | 0.206,0.943 |

**Observations:**

1. In the initial 2 models, Models 2 and 3, the network architecture used to train these models was quite simple with just an embedding layer allowing them to be tested on very small amounts of data. We cannot even say that the model's performance was exceedingly good, but it wasn't bad either.
2. Our analysis of Model 4 and Model 5 results indicates that the models' performance changed when we experimented with Convolution 1D, Embedding Layers, and Dense Layers:

* Providing more samples to the model allows it to learn more about the data, such as in this case, learning the positive and negative elements of a review, which then helped it generalize well to test data.
* It is possible to generalize well from one sample to another by providing more samples to the model, such as learning the positive and negative elements of a review in this case.
* A more robust model is achieved by using Conv1D in conjunction with an embedded layer. Conv1D works through filters, and as the filters move along, the model tends to gain a better understanding of the review, which results in a more accurate prediction when applied.
* In the model-building phase, controlling hyper-parameters is crucial, as it is important to observe the previous model's performance to make changes to improve the model's performance.
* There have been several key hyperparameters that have been fine-tuned along the way to increase the generalization ability of the model, including embedding vector dimensions, learning rate, convolution 1D layers, dense layers, dropout rate, and nodes.

1. To determine when a model has begun to overfit, it is useful to plot the training and validation loss / accuracy.
2. The most effective way to achieve better generalization on the test set is to build a complex network architecture and then let the model overfit.
3. In the end, whenever your model tends to underperform, always provide more data for it to learn from.

**Summary:**

Model 5, with a training set of 35,000 samples, showcased exceptional performance compared to the other models. This model achieved notably higher accuracy (0.943) and lower loss (0.206), showcasing its superior ability to generalize and predict sentiment in IMDB reviews. The strategic use of a complex architecture, along with a large dataset, facilitated improved generalization and minimized underperformance, making it the most effective among the experimented models.

The following graphs of the observations of total models:

A graph with red dots

Description automatically generated

A graph with blue dots

Description automatically generated

**Pre-Trained Network:**

**Objective:**

In these models, LSTM layers were built using the vector dimension of the GloVe network.

**By trying the different sample sizes:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model number | Training | Validation | Test | Performance on test set (Loss, Accuracy) |
| Model 2 | 1000 | 10000 | 5000 | 0.690,0.730 |
| Model 3 | 10000 | 10000 | 5000 | 0.691,0.530 |
| Model 4 | 15000 | 10000 | 5000 | 0.693,0.493 |
| Model 5 | 15000  (reducing complexity) | 10000 | 5000 | 0.432,0.810 |

**Observations:**

1. This model has been designed to learn from underfitting and overfitting issues. A model that tends to underfit indicates that it is not doing well on the training data itself, which in turn signifies that it will do even worse on the unseen data.
2. Underfit models 3 and 4 do not achieve higher performance on the training set, and their performance on the test set is even worse than their performance on the training set.
3. The best way to build a better model may not always be to choose a complex architecture with layers and nodes, but rather to choose a simpler architecture with fewer input layers and nodes.
4. Model 5 was constructed with an easy network architecture, so the model had good performance on the training set and did not underfit. Eventually, some test sets performed better than other models.
5. In conclusion, the 5th model was the most successful out of all the five models it was built with 15000 training samples, 1000 validation samples, and 5000 test samples.

As a result of evaluating pre-trained models, we learned that complex networks are not always the best option for improving generalization abilities. Sometimes simple models or weak learners can outperform others, as was likely the case in our model performance evaluation. Moreover, we must always check whether the model is overfitting or underfitting and adjust the parameters accordingly.

**Summary:**

Model 5, built on 15,000 training samples, employed a simpler LSTM architecture with GloVe vectors, achieving a notable test accuracy of 0.810 and a loss of 0.432. Its success emphasized the power of simplicity over complexity in model performance, highlighting the need for ongoing parameter adjustments to counter underfitting or overfitting during development of models.

The following shows the graph for these results:

A graph with red dots

Description automatically generated

A graph with blue dots

Description automatically generated