Texts and Sequences

Introduction:

The IMDB dataset is widely used for natural language processing (NLP) tasks, particularly sentiment analysis. It contains movie reviews from the website IMDb, along with labels indicating whether the review is positive or negative. Each review includes a variable-length word sequence, making it an ideal dataset for challenges involving text processing and sequence modeling. The dataset is frequently used to train machine learning models that predict the sentiment (positive or negative) of a given text.

The objective of the binary classification issue on the IMDB dataset is to determine if a movie review is either positive or negative. The dataset consists of 50,000 reviews, with just the top 10,000 words evaluated, training samples limited to 100,10000, 5000, 2500, and 1000 validation based on 10,000 samples, and reviews examined after 150 words. And also we tried with the top 20000 words with the best sample size. The data is pre-processed.

The data is then given to a pre-trained embedding model and an embedding layer, where different strategies are tested to evaluate performance. The primary objective of the binary classification work on the IMDB dataset is to identify which method performed best and whether a movie review is positive or negative.

Summary:

1. ***Cutoff Reviews after 150 words***

We have first cut off the no.of words in each review to 150 words by giving length=150. So that the model will only consider the first 150 words from each review in the dataset. Then we trained the models with different training sample sizes 100, 10000, 5000, 2500, 1000 respectively, and then used pre-trained models.

2. ***Restrict training samples to 100.***

After cutting off the no. of words to 150, we trained the model with the training sample size of 100. So, that the model can randomly select 100 data points for its training.

3. ***Validate for 10000 records.***

We have trained the model with different sample sizes. but we have fit the model with the validation sample size of 10000 records for all the models. To get a better understanding of hyperparameters. validating with more records can provide a more accurate assessment of the model's performance and its ability to generalize to new data.

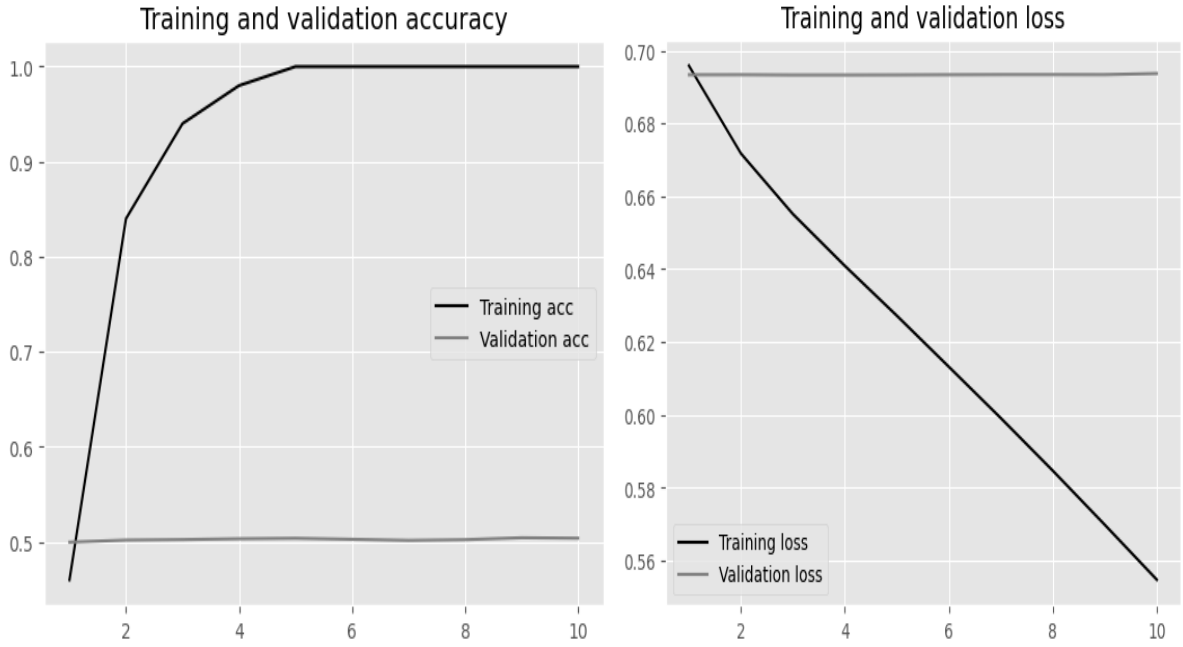
4. ***Consider only the top 10000 words.***

Out of four lakh words in English vocabulary, we have trained our model with the first 10000 words which were frequently being used by native speakers daily. As we have limited computational resources or time may limit the quantity of data that can be used in training. While training with fewer words might be helpful in certain situations, it is important to examine the potential drawbacks. A smaller dataset may not represent the entire complexity of

the problem, and the model's performance on a small dataset may not apply to a larger, more diverse dataset.

For training sample size 100, validation sample size 10000, and no. of words 10000:

model1 with 100 samples	**Training**	**Validation**	**Test**
Accuracy	100%	50.38%	50.39%
Loss	0.5548	0.6937	0.6935



As the training sample size was too small the accuracy for training and validation was too low around 50.38% and the loss was too high. As there was an overfitting due to very less data. So, lets increase the size of the training sample to 1000 by keeping the validation sample size the same.

For training sample size 1000, validation sample size 10000, and no. of words 10000:

model with 1000 samples	**Training**	**Validation**	**Test**
Accuracy	100%	69.86%	69.9%
Loss	0.0065	0.6272	0.626

As the training sample size increased from 100 to 1000, the training loss was reduced to 0.006 and the validation and test accuracies were increased to 62% and losses has been reduced and the validation sample size was kept the same. With increase in sample size, the model is likely to generalize better to new, unseen data. This is because it has been exposed to a wider range of examples during training. With more data, the model is less likely to be biased towards a particular subset of the data.

For training sample size 2500, validation sample size 10000, and no. of words 10000:

model with 2500 samples	**Training**	**Validation**	**Test**
Accuracy	100%	80.89%	80.8%
Loss	0.0041	0.488	0.496

Here the test and validation accuracies are increased to 80% and the losses has been reduced. Let's increase the sample size furthermore to 5000.

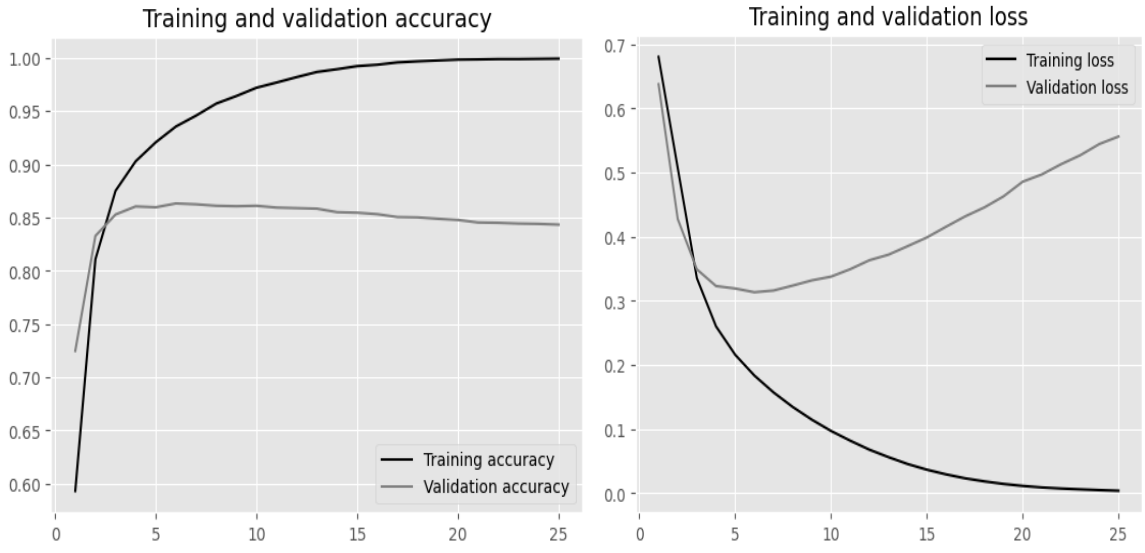
For training sample size 5000, validation sample size 10000, and no. of words 10000:

model with 5000 samples	**Training**	**Validation**	**Test**
Accuracy	99.9%	83.1%	83.1%
Loss	0.007	0.467	0.466

Here the test and validation accuracies are increased to 83% and the losses has been reduced. Let's increase the sample size furthermore to 10000.

For training sample size 10000, validation sample size 10000, and no. of words 10000:

model with 10000 samples	**Training**	**Validation**	**Test**
Accuracy	99.9%	84.34%	84.63%
Loss	0.0040	0.5562	0.5554



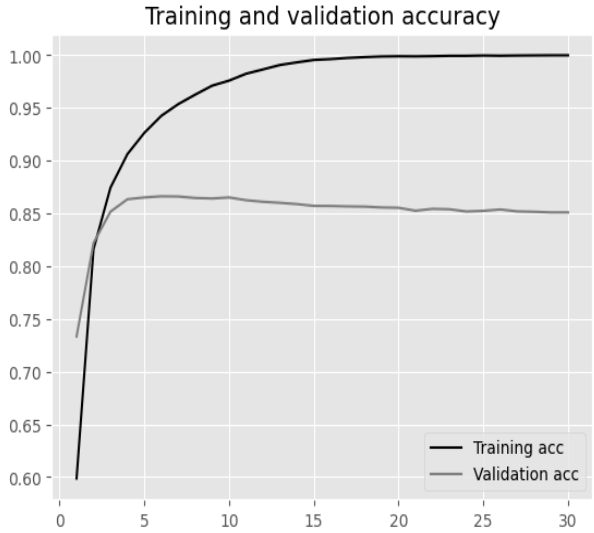
As the training sample size increased from 100 to 10000, the training loss has been reduced to 0.004 and the validation and test accuracies has been increased to 84% and the losses has been reduced and the validation sample size has been kept the same. With increase in sample size, the model is likely to generalize better to new, unseen data. This is because it has been exposed to a wider range of examples during training. With more data, the model is less likely to be

biased towards a particular subset of the data. With a larger sample size, the variability in the data is reduced. This can lead to a more stable model, as it is less likely to be influenced by random fluctuations in the data.

We have found the best model with training sample size of 10000, validation sample size of 10000, and no. of words 10000, lets increase the size of no. of words to 20000.

For training sample size 10000, validation sample size 10000, and no. of words 20000

model with 10000 samples	**Training**	**Validation**	**Test**
Accuracy	99.9%	85.09%	84.6%
Loss	0.0012	0.563	0.558



As the no. of words increases the training loss almost becomes 0, and the validation and test accuracies increased a little bit to 85%. A bigger vocabulary enables the model to possibly capture more subtle and particular textual elements, resulting in improved performance for tasks that require a thorough comprehension of the language. However, this means that the embedding matrix will have more rows, which increases the number of parameters and the danger of overfitting, particularly with minimal data. Furthermore, a bigger vocabulary necessitates additional memory and processing resources, which affects the model's scalability. As there was not so much improvement from 10000 to 20000, we can go with 10000 words.

Let's use a pre-trained word embedding model (GloVe):

Sometimes you have so little training data that you are unable to develop a suitable task-specific vocabulary embedding using just your data. In these situations, you can load embedding vectors from a precomputed embedding space that you know is highly structured and exhibits helpful properties—one that captures basic characteristics of language structure instead of learning word embeddings in conjunction with the problem you wish to address.

The reasoning behind the use of pre-trained word embeddings in natural language processing is similar to that of pre-trained convolutional networks in image classification: you expect that the features you require are fairly generic, i.e., common visual or semantic features, but

you don't have enough data available to learn truly powerful features on your own. Reusing features that were discovered on an unrelated topic makes sense in this situation.

For training sample size 100, validation sample size 10000, and no. of words 10000:

model1 with 100 samples	**Training**	**Validation**	**Test**
Accuracy	100%	57.13%	50.6%
Loss	0.0009	0.8023	0.850

On the training dataset, the pretrained model obtained a perfect accuracy of 100%, demonstrating its ability to precisely fit the training data. Nevertheless, the model's high accuracy did not translate well to the test and validation datasets, where it obtained accuracies of 50.6% and 57.13%, respectively. This shows that the model might have performed well on seen data but poorly on unseen data due to overfitting of the training set. This pattern is also shown in the loss numbers, which show that the training dataset had a very low loss of 0.0009 while the test and validation datasets had larger losses of 0.8023 and 0.850, respectively. Overall, these findings show that although the pretrained model did remarkably well on the training set, it had trouble generalizing too generalize to new, unseen data.

For training sample size 1000, validation sample size 10000, and no. of words 10000:

model1 with 1000 samples	**Training**	**Validation**	**Test**
Accuracy	99.9%	50.61%	49.3%
Loss	0.0015	2.0035	2.049

A similar trend can be seen in the updated pretrained model with 1000 samples compared to the prior model with 100 samples. A strong match to the training data is indicated by the model's high accuracy of 99.9% on the training dataset. On the validation and test datasets, however, the model's high accuracy did not translate well, yielding lower accuracy of 50.61% and 49.3%, respectively. This implies that the model might have overfit the training set, just like the prior model did. This pattern is also shown in the loss values, which show that the training dataset had a low loss of 0.0015 while the test and validation datasets had larger losses of 2.0035 and 2.049, respectively. While the model did remarkably well overall on the training set of data, it struggled to generalize to new, unseen data.

For training sample size 2500, validation sample size 10000, and no. of words 10000:

model1 with 2500 samples	**Training**	**Validation**	**Test**
Accuracy	99.32%	50.29%	50.36%
Loss	0.0164	1.7286	1.685

The model achieved a high accuracy of 99.32% on the training dataset, indicating a strong fit to the training data. However, this high accuracy did not generalize well to the validation and test datasets, where the model achieved lower accuracies of 50.29% and 50.36% respectively. This suggests that, like the previous models, the model may have overfit the training data. The loss values also reflect this trend, with a low loss of 0.0164 on the training dataset compared to higher losses of 1.7286 and 1.685 on the validation and test datasets respectively. Overall,

while the model performed exceptionally well on the training data, it struggled to generalize to new, unseen data, similar to the previous models.

For training sample size 5000, validation sample size 10000, and no. of words 10000:

model with 5000 samples	**Training**	**Validation**	**Test**
Accuracy	99.52%	49.4%	50.38%
Loss	0.0103	1.7722	1.704

The model that was trained on 5000 data displays a pattern that is somewhat better than the earlier models. With a high accuracy of 99.52% on the training dataset, the model demonstrated a good match to the training set of data. On the validation dataset, the accuracy is 49.4%, which is marginally less than the models with fewer samples. With a test dataset accuracy of 50.38%, it is marginally higher than the validation accuracy but still quite poor. This shows that although the model is operating effectively on the training set, it is having difficulty generalizing to fresh, untested data. A similar pattern can be seen in the loss values as well, with the training dataset showing a low loss of 0.0103 and greater losses of 1.7722 and 1.704 on the validation and test datasets respectively. Overall, the model's performance improves with more training samples, but it still faces challenges in generalization.

For training sample size 10000, validation sample size 10000, and no. of words 10000:

model with 5000 samples	**Training**	**Validation**	**Test**
Accuracy	99.32%	49.8%	49.9%
Loss	0.0222	3.804	3.705

The model that was trained on 10000 data displays a pattern that is somewhat better than the earlier models. With a high accuracy of 99.32% on the training dataset, the model demonstrated a good match to the training set of data. On the validation dataset, the accuracy is 49.8%, which is marginally less than the models with fewer samples. With a test dataset accuracy of 49.9%, it is marginally higher than the validation accuracy but still quite poor. This shows that although the model is operating effectively on the training set, it is having difficulty generalizing to fresh, untested data. A similar pattern can be seen in the loss values as well, with the training dataset showing a low loss of 0.0222 and greater losses of 3.804 and 3.705 on the validation and test datasets respectively.

Conclusion:

At last, we can say that the training sample size is 10000, the validation sample size is 10000, and no. of words is 10000. As the accuracy for this model in both training and validation was high about 84% and the losses were too less. There are a few important reasons why the model trained with 10,000 samples performed significantly better in terms of generalization than the earlier models. The larger dataset size, in the initial stages, gives the model a greater number of samples to work with and improves its ability to identify underlying patterns in the data. Additionally, because this larger dataset is usually closer to the reality of the data's underlying distribution, there is less chance of overfitting and the model performs better when applied to fresh, untested data. More samples also result in a dataset with less variance, which makes the model more stable and less dependent on small discrepancies in the training set. Additionally, the 10,000-sample model's model architecture may be better suited for the task, making it

capable of gaining more insight from the bigger dataset. Furthermore, the larger dataset might have offered more chances to optimize the learning rate, batch size, and regularization—three hyperparameters that could have further enhanced the model's performance. Taken together, these variables highlight how crucial model design, representation, and dataset size are to obtaining strong results in machine learning tasks.