Assignment – 4

Text Data

-Teja Tarapatla

Objective:

Classifying film reviews as either favorable or negative is the aim of the binary classification job for the IMDB dataset. Out of the 50,000 reviews in the dataset, the 10,000 most frequently occurring words are considered. Various sample sizes are used for training (100, 1,000, 5,000, and 100,000 samples), while 10,000 samples are used for validation. After the data has been preprocessed, it is fed into a pretrained embedding model together with the embedding layer, where performance is assessed using a variety of techniques.

Data:

Each review undergoes a series of word embeddings as part of the dataset preparation process, where each word is represented by a fixed-length vector.

▪ As a result, 10,000 samples are the limit. Furthermore, the reviews are used to generate a series of numbers that correlate to individual words rather than a string of words. The list of numbers is in my possession, but it cannot be used as input by the neural network directly.

▪ It is necessary to use numerical values when building tensors. Making a tensor with samples and word indices in integer format is one method of handling the integer list.

▪ To do this, I must make sure that every sample is the same length, which entails consistent sample size by padding reviews with dummy words or numbers.

Method Used:

I discovered two unique techniques for creating word embeddings for the IMDB dataset:

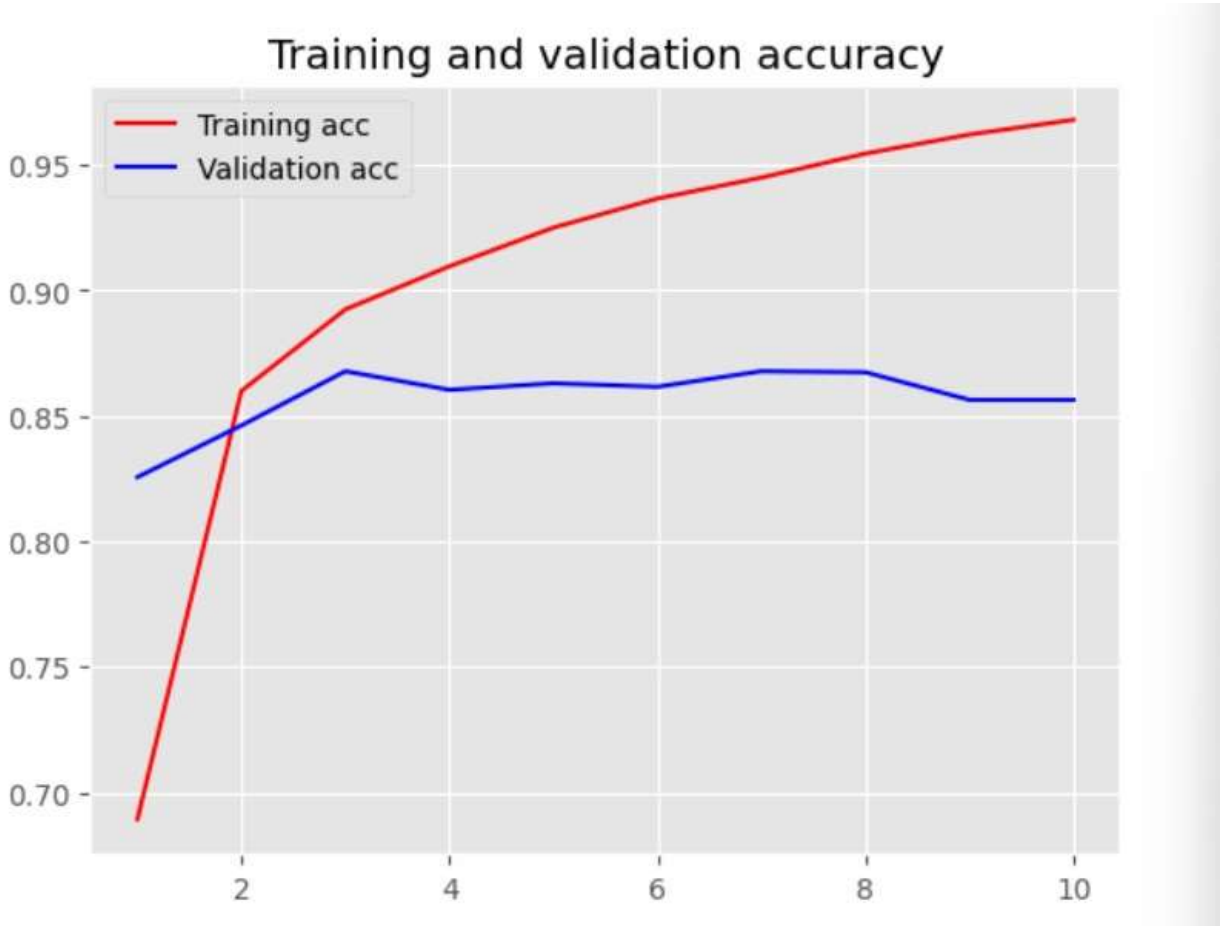
1. Custom-trained embedding layer

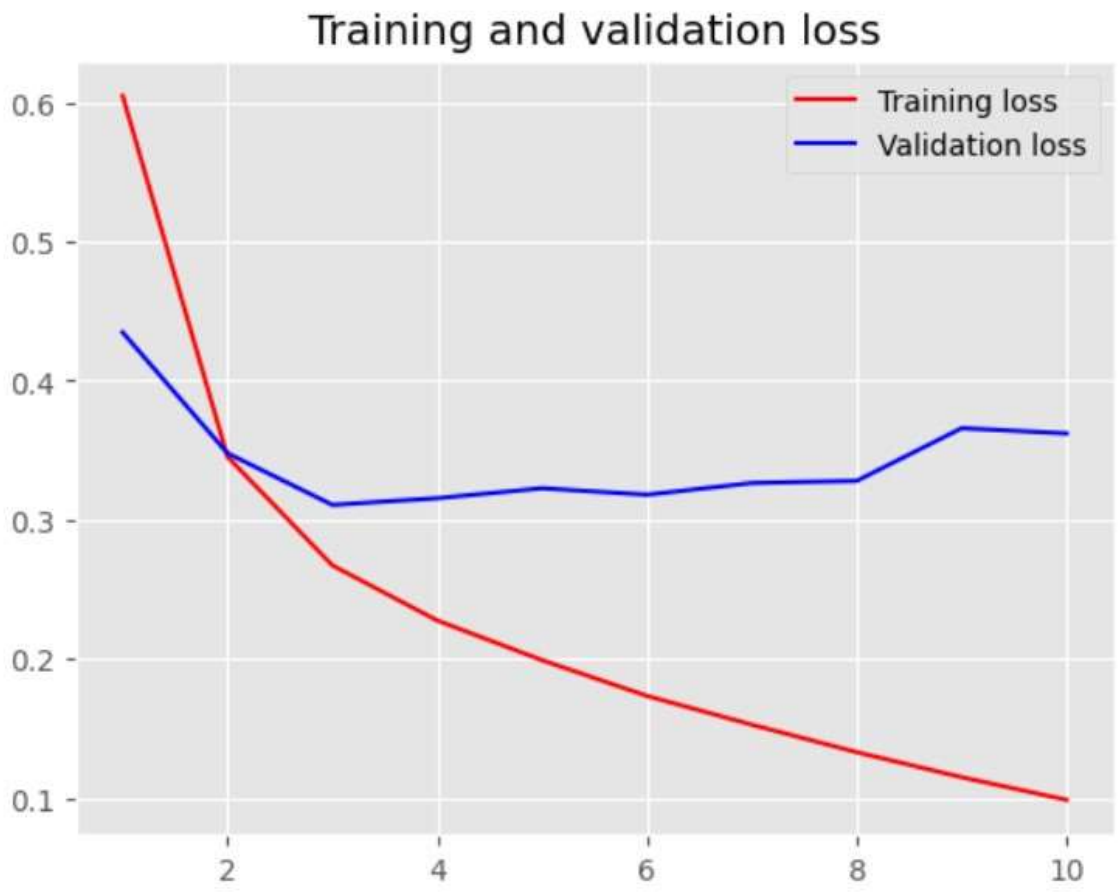
2. GloVe model-based pretrained word embedding layer.

Using a huge corpus of text as training data, we applied the popular GloVe pretrained word embedding model in this work. Using the IMDB dataset, we compared custom-trained and pretrained embedding layers to assess accuracy across 10k, 1,000, 5,000, and 10,000 sample sizes. The models were evaluated using IMDB reviews with different sample sizes, and their accuracy was evaluated using either pretrained or custom-trained embeddings.

CUSTOM-TRAINED EMBEDDING LAYER:

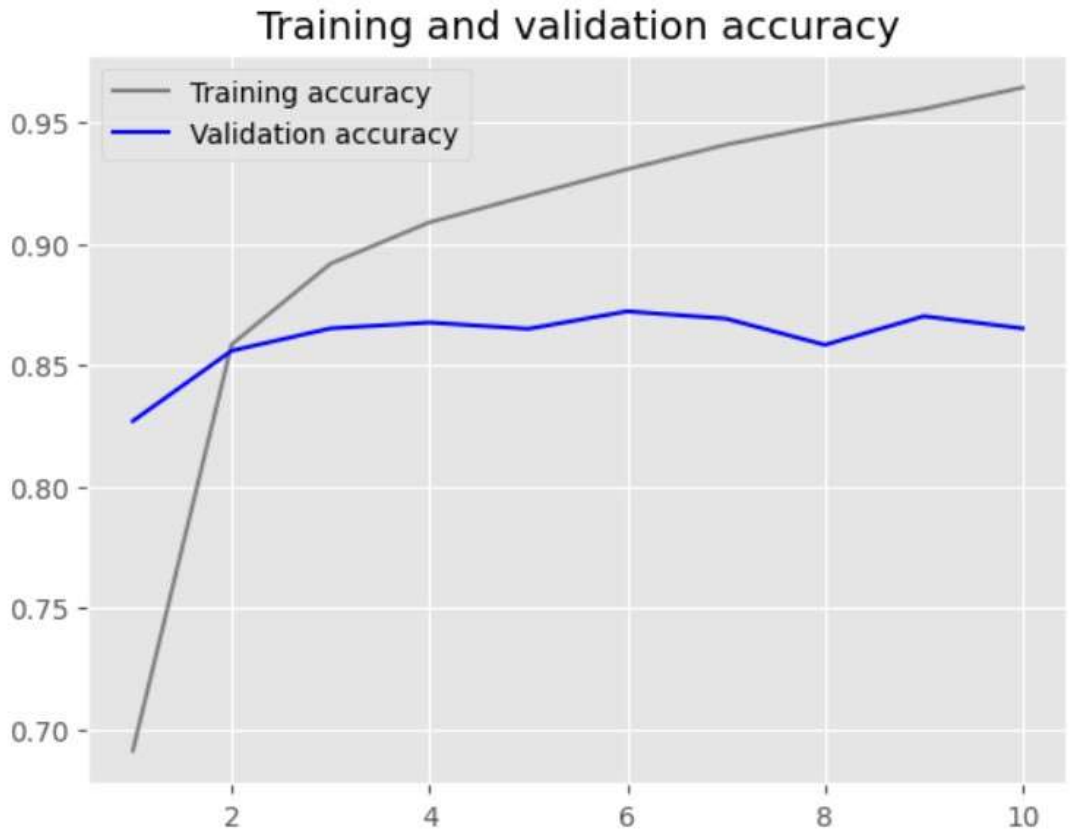
Training Dataset with 100 samples





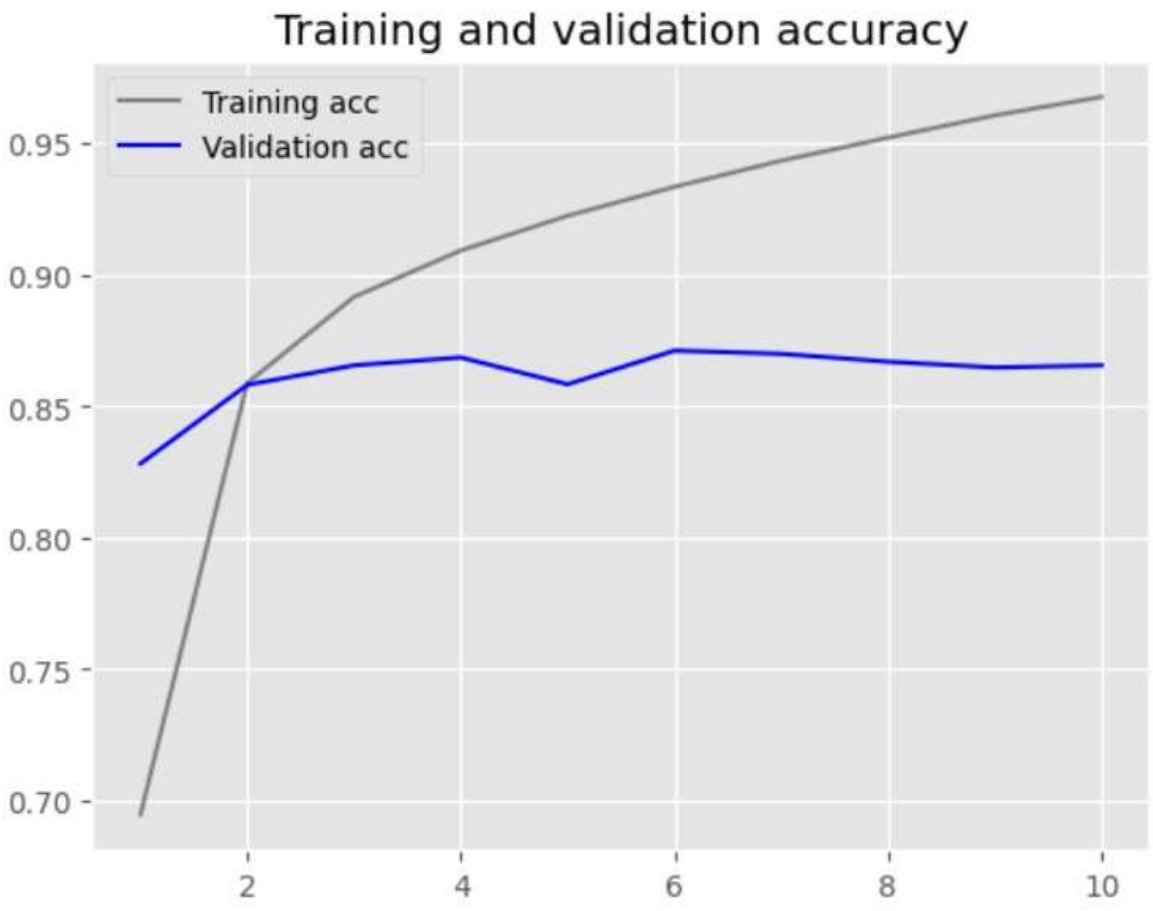
The training accuracy of the model increased steadily over the course of ten epochs, from 60.43% in the first epoch to 97.14% in the tenth. The training loss showed good convergence, dropping from 0.6665 to 0.0932. Starting at 82.56%, the accuracy on the validation set varied a little until reaching a peak of 86.78% by the seventh epoch and then settling at 85.64% in the last epoch. The validation loss decreased from 0.4347 to 0.3621, following a similar pattern. Even though the validation accuracy varied slightly, the model demonstrated good generalization when tested on the test set, achieving a test accuracy of 85.72% with a test loss of 0.3629.

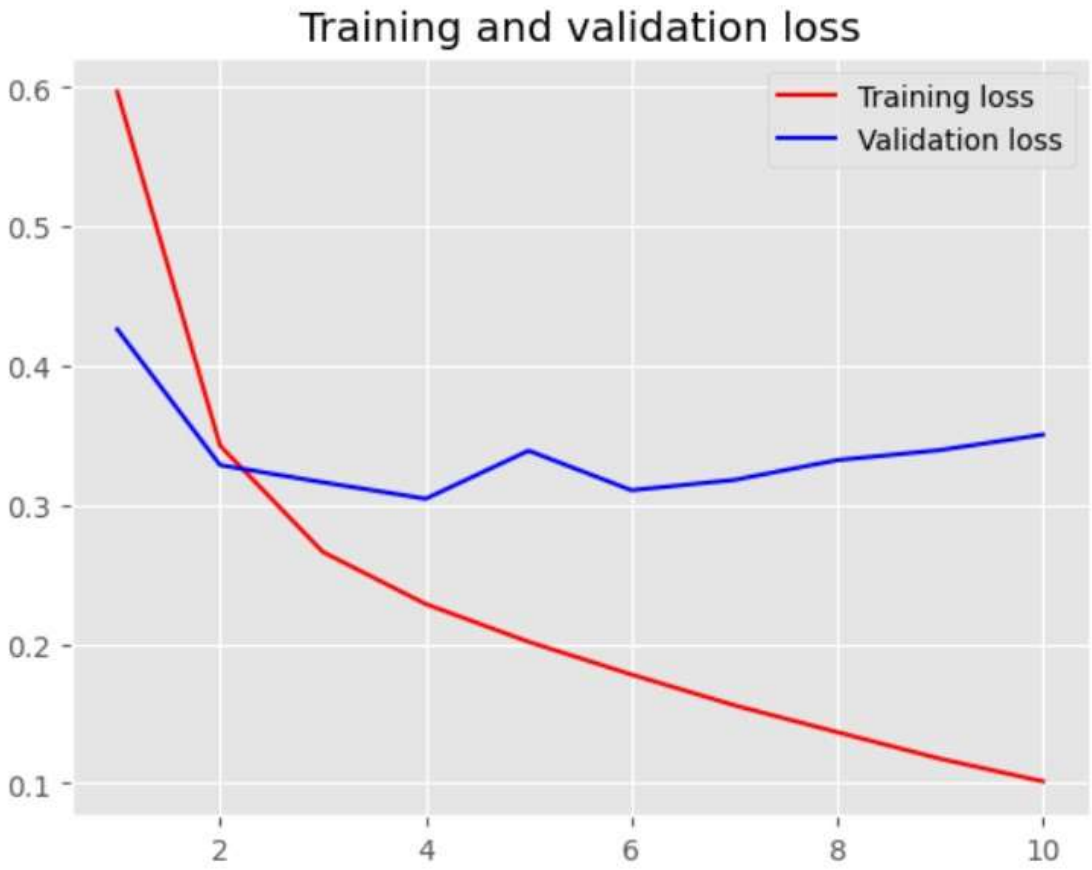
Training Dataset with 5000 samples:



The training accuracy increased from 59.01% in the first epoch to 96.55% by the tenth epoch after the model was trained for ten epochs. Training loss dropped from 0.6649 to 0.1077, demonstrating successful learning. Prior to stabilizing at 86.52% by the last epoch, accuracy on the validation set began at 82.70% and fluctuated, reaching a peak of 87.22% in the sixth epoch. The validation loss decreased from 0.4246 to 0.3479 in a similar manner. The model had good generalization performance despite the changes in the validation accuracy, achieving a test accuracy of 86.30% with a test loss of 0.3515 when assessed on the test set.

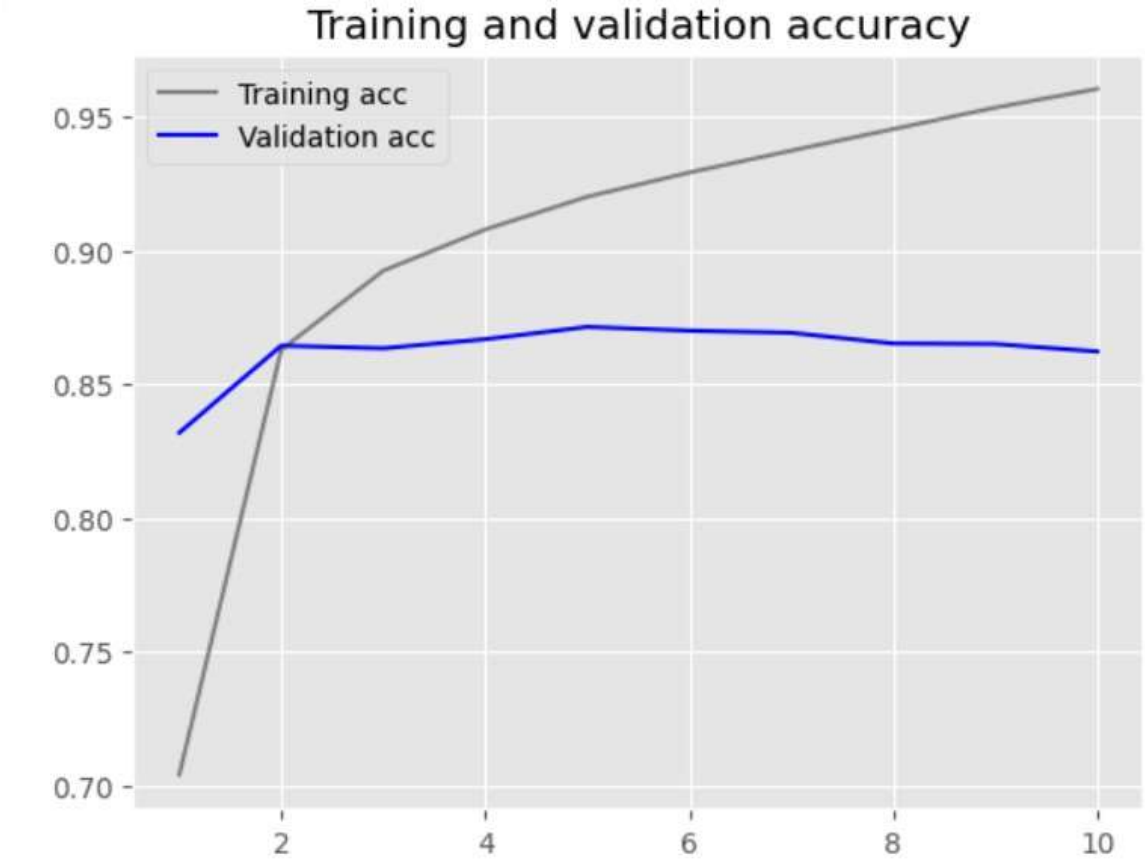
Training Dataset with 1000 samples:

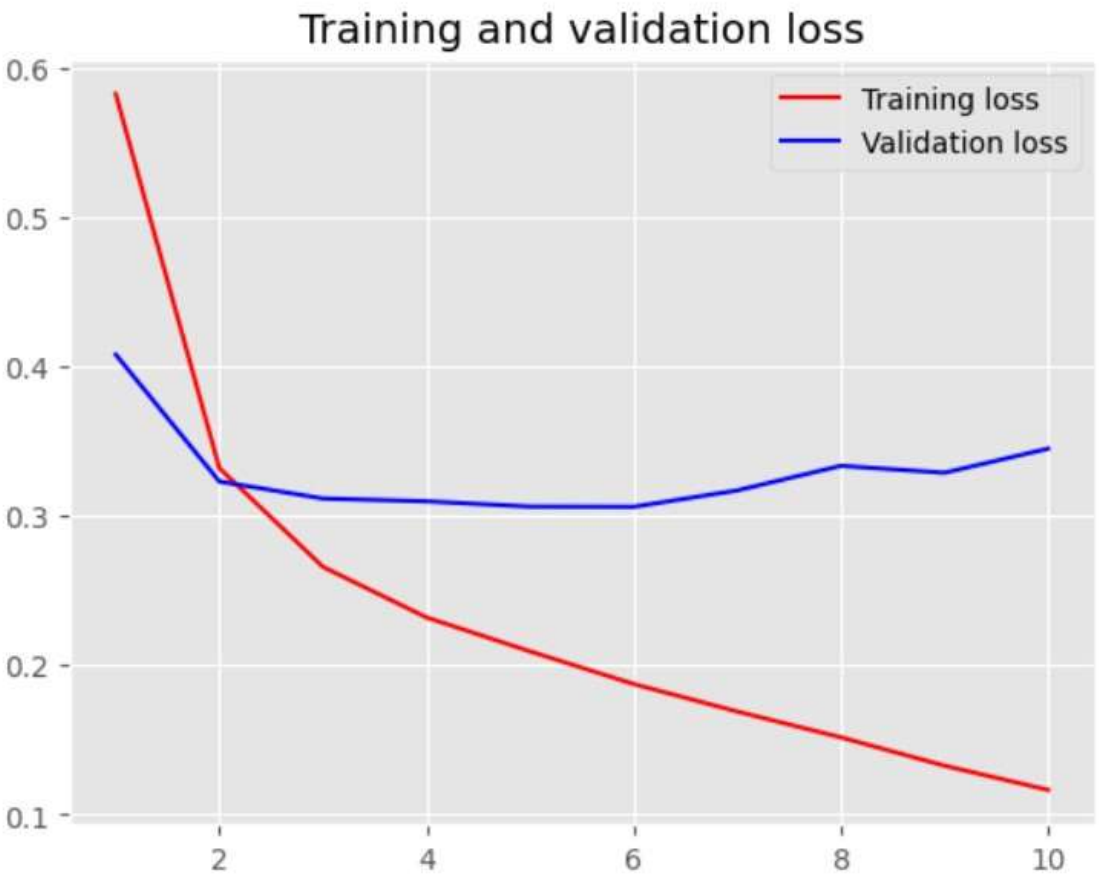




In the first epoch, the model's training accuracy was 59.99%; by the tenth epoch, it had increased to 96.88%, and the training loss had decreased from 0.6639 to 0.0995. After beginning at 82.82%, validation accuracy steadily increased, reaching a peak of 87.12% in the sixth epoch before leveling off at 86.56% at the conclusion of training. From 0.4262 to 0.3504, the validation loss dropped. The model scored a test accuracy of 86.41% with a test loss of 0.3487 when tested on the test set, indicating high performance and generalization despite validation accuracy variations.

Custom-trained embedding layer with training sample size = 10000





Over the course of ten epochs, the model's training accuracy increased from 60.49% in the first epoch to 96.23% by the tenth, while the associated training loss decreased from 0.6575 to 0.1136. The validation accuracy began at 83.18% and steadily increased, peaking at 87.14% in the fifth epoch before progressively dropping to 86.22% at the conclusion of training. With a test accuracy of 86%, the validation loss dropped from 0.4079 to 0.3449, suggesting a generally successful match despite some variations in validation performance.

Highest Test Accuracy:

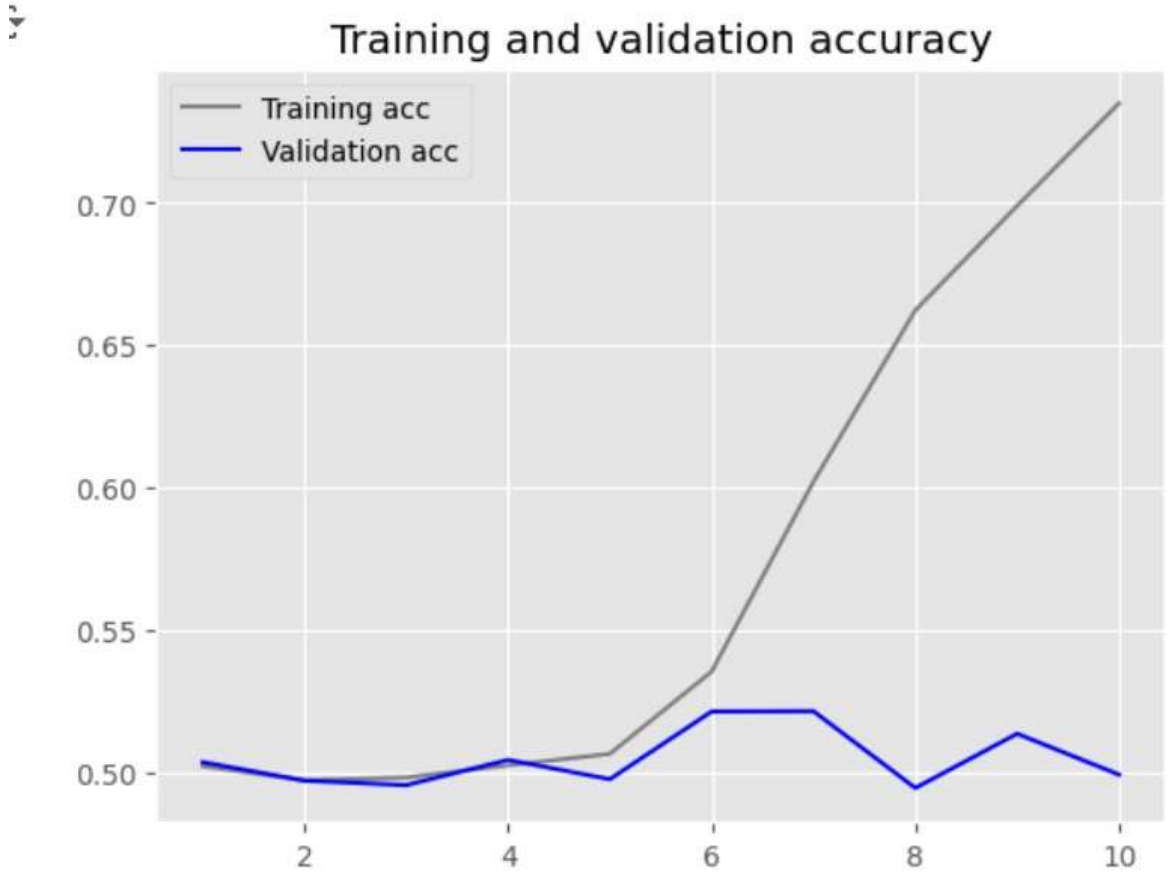
• The maximum test accuracy of 86.58% was attained with a sample size of 1000.

Precision Variation:

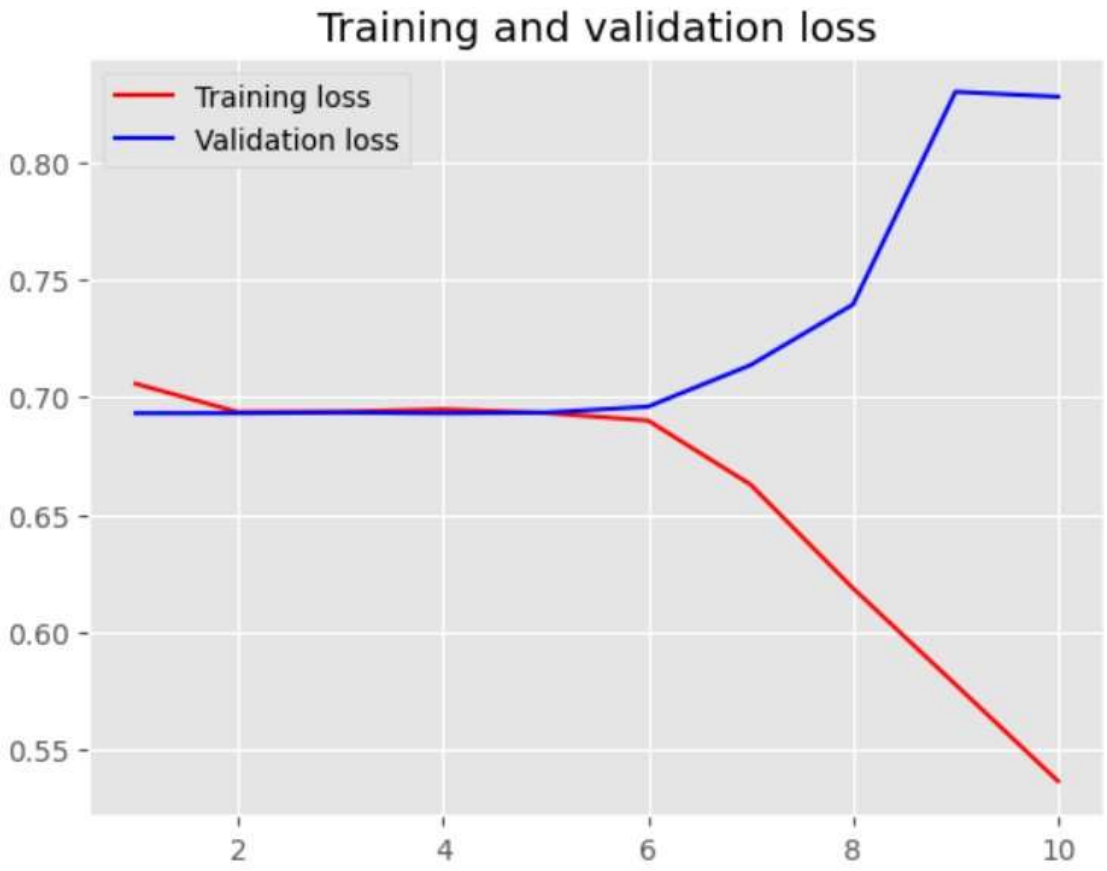
- The difference between sample sizes is negligible, but the precision (test accuracy) seems to rise slightly with increasing sample size.
- The accuracy ranges from 85.72% (sample size 100) to 86.58% (sample size 1000), increasing by 0.86% from the lowest to the greatest.

PRETRAINED WORD EMBEDDING LAYER:

Training Sample Size 100

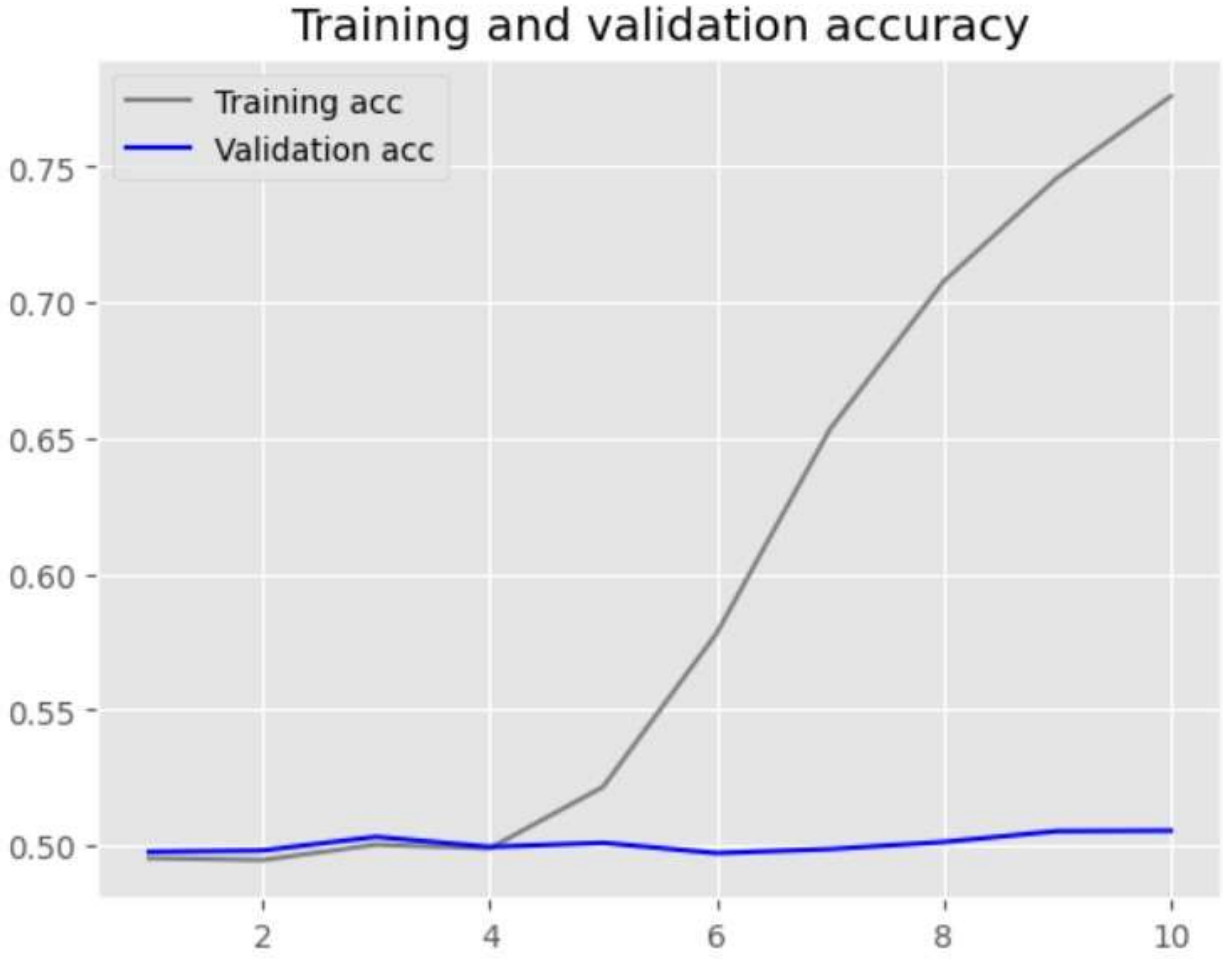


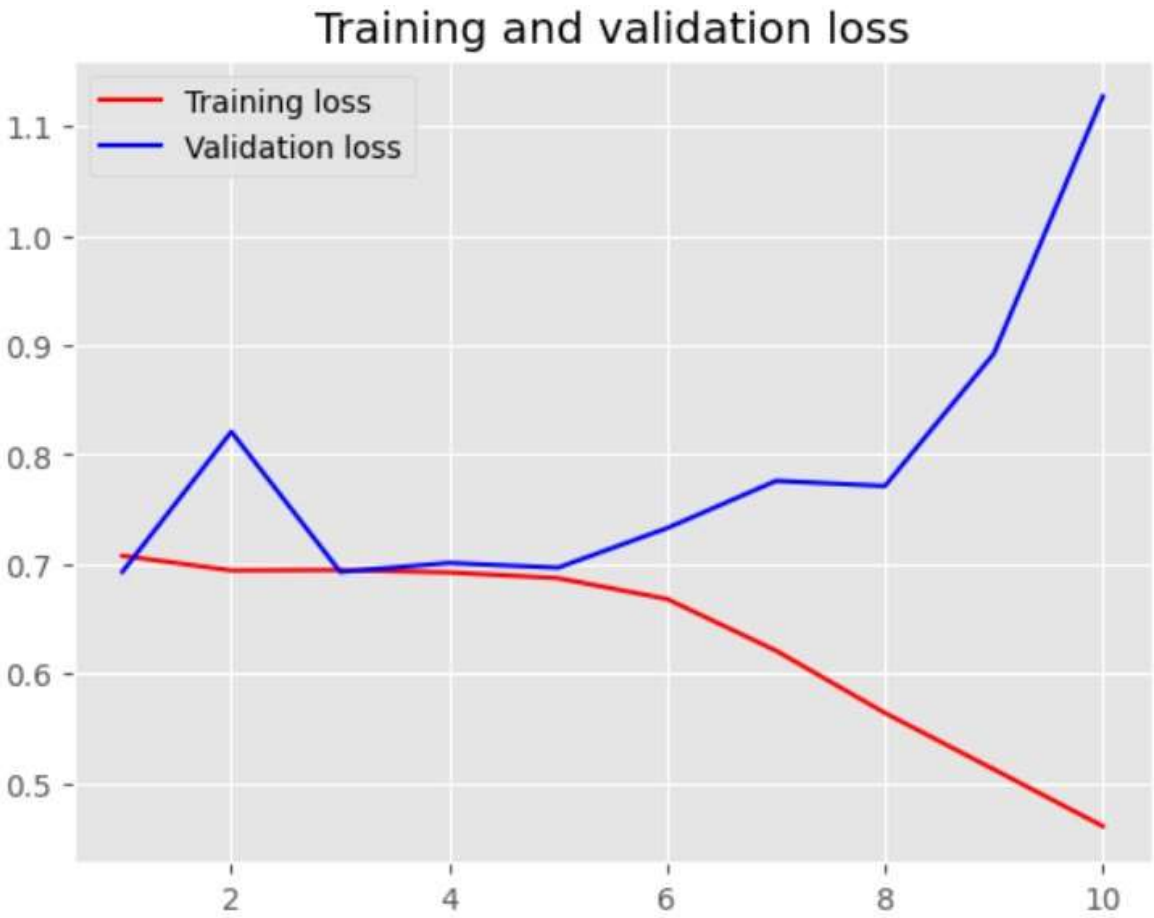
The model's validation accuracy stayed between 49.43% and 52.12%, showing weak generalization, whereas its training accuracy increased consistently across 10 epochs from 50.26% to 74.21%. Overfitting was suggested by fluctuating validation accuracy, even with the increasing training performance. Despite being marginally higher than the validation performance, the test accuracy of 56.51% is still quite low, with a test loss of 0.7322, which illustrates how poorly the model generalizes to unknown data. These findings imply that the model has a limited capacity for generalization and could use more modifications, like regularization strategies or additional data.



The model's validation accuracy stayed between 49.43% and 52.12%, showing weak generalization, whereas its training accuracy increased consistently across 10 epochs from 50.26% to 74.21%. Overfitting was suggested by the fluctuating validation accuracy, even with the increasing training performance. Despite being marginally higher than the validation performance, the test accuracy of 56.51% is still quite low, with a test loss of 0.7322, which illustrates how poorly the model generalizes to unknown data. These findings imply that the model has a limited capacity for generalization and could use more modifications, like regularization strategies or additional data.

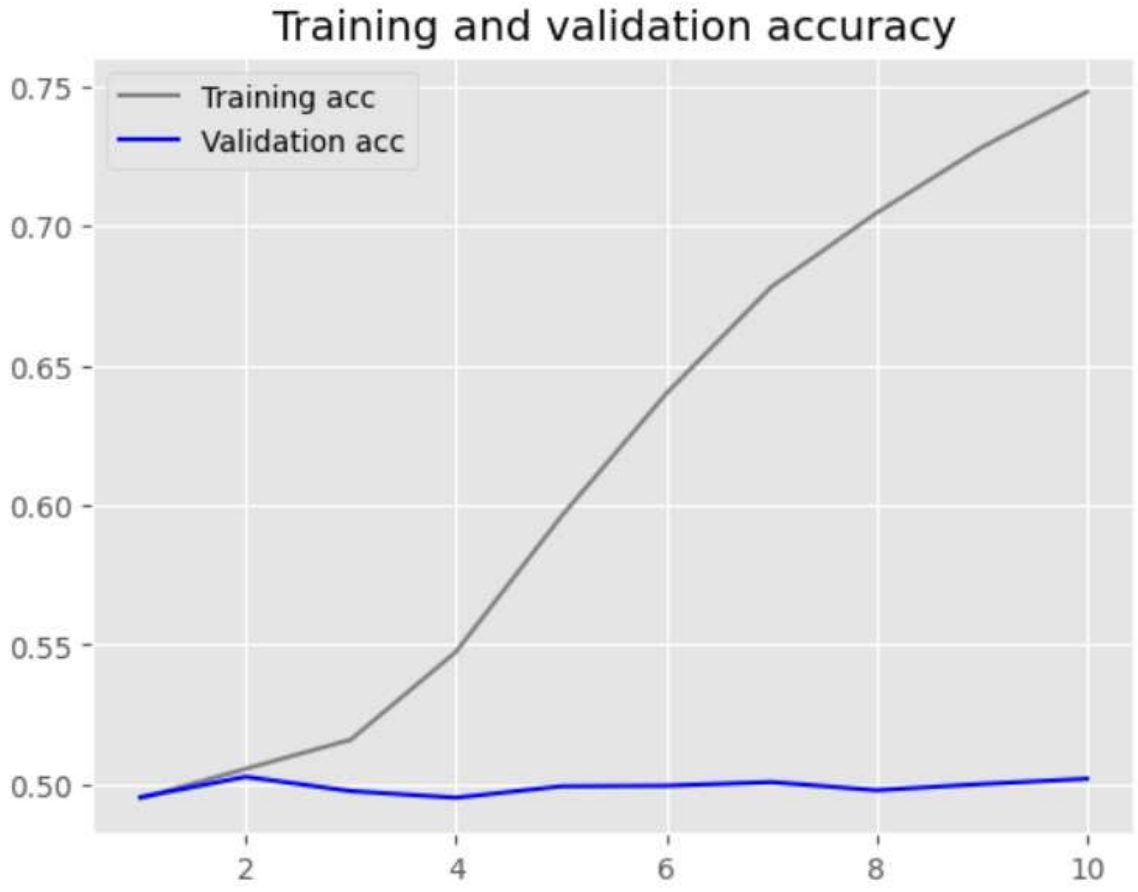
Training Sample 5000:

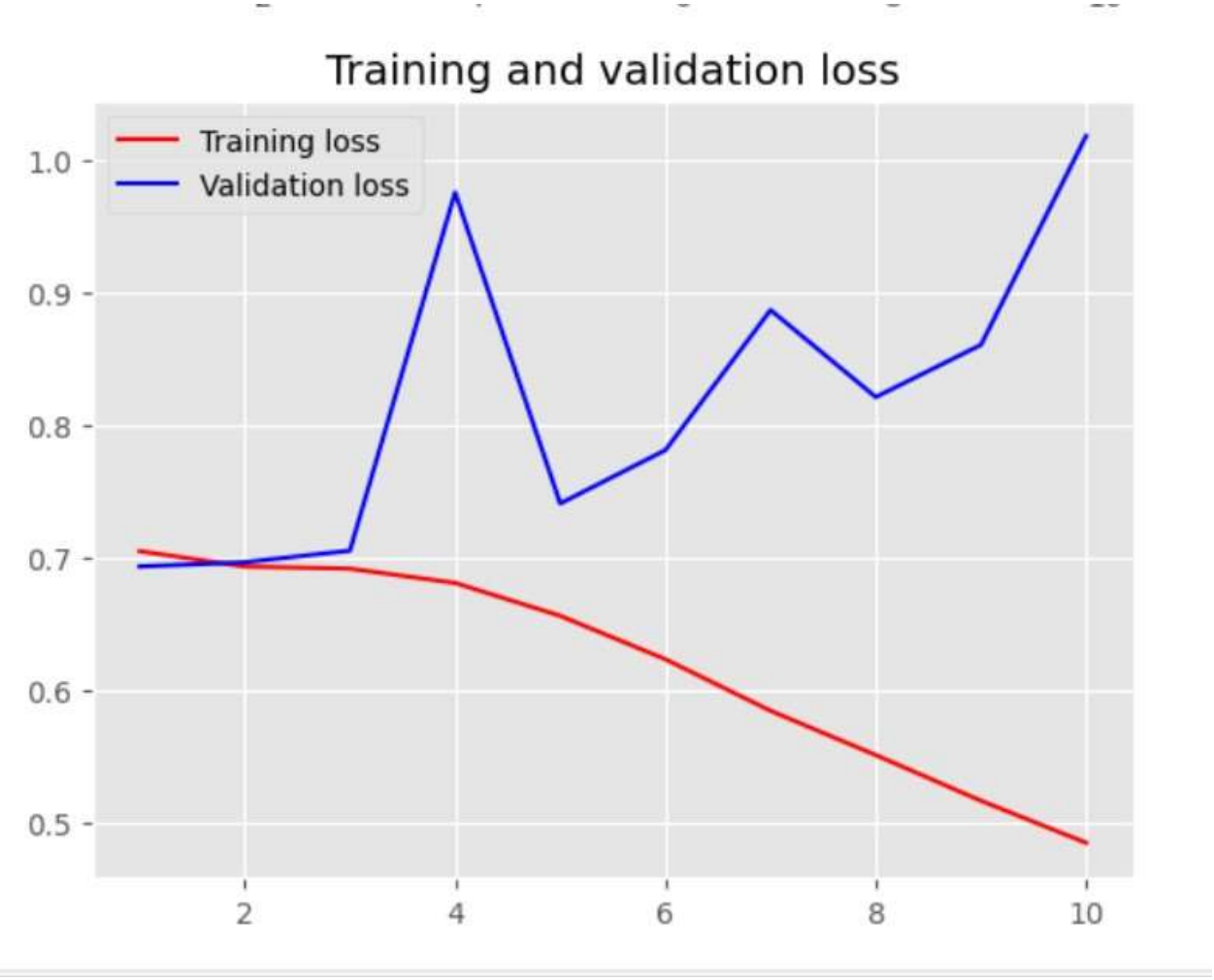




As the model learned the task, its training accuracy increased across 10 epochs, from 49.31% to 78.29%. With a range of 49.43% to 50.56%, the validation accuracy was continuously poor, indicating that the model had trouble generalizing to the validation set. In addition, the model's poor test accuracy of 56.73% and test loss of 0.9146 demonstrate its incapacity to generalize well to new data. Increased training accuracy combined with stagnating test and validation performance suggests possible overfitting, and other enhancements, like regularization or data augmentation methods, may be required.

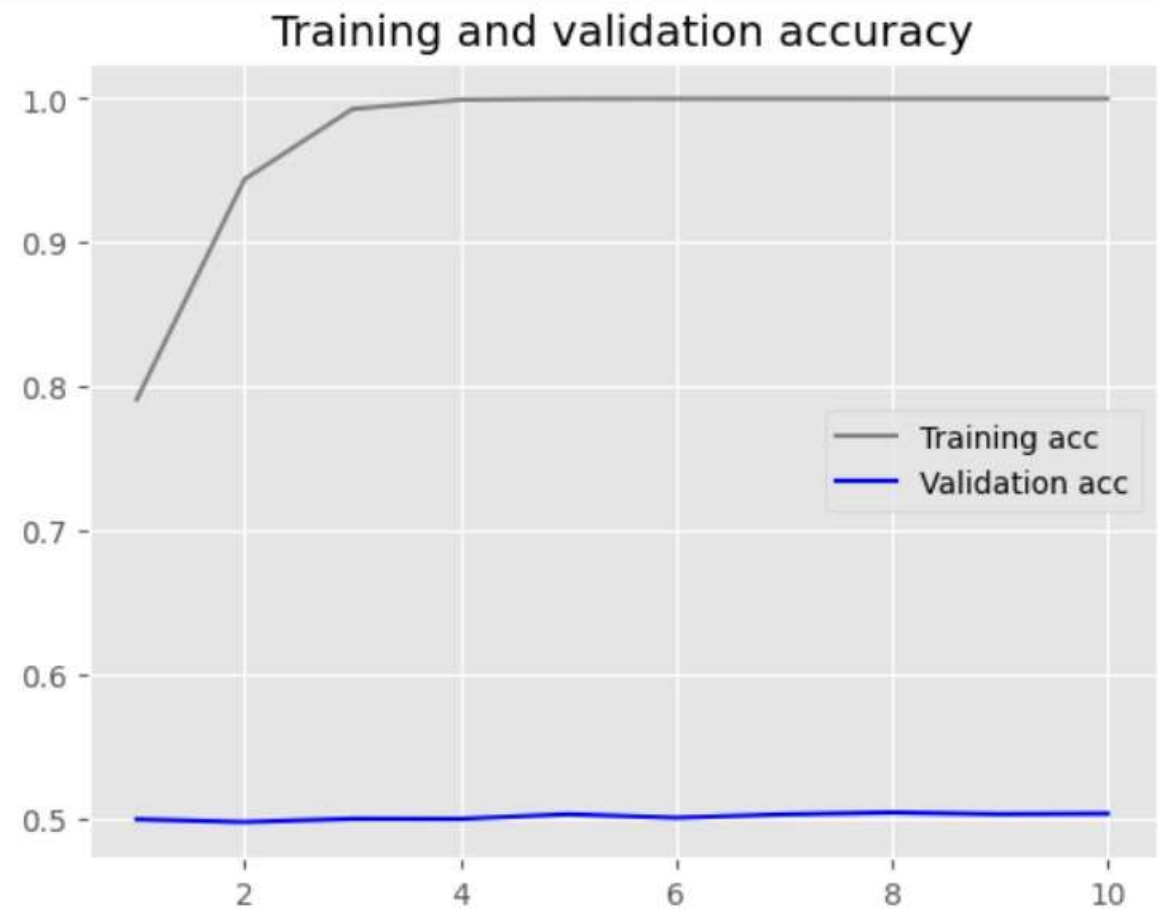
Training Sample 1000:

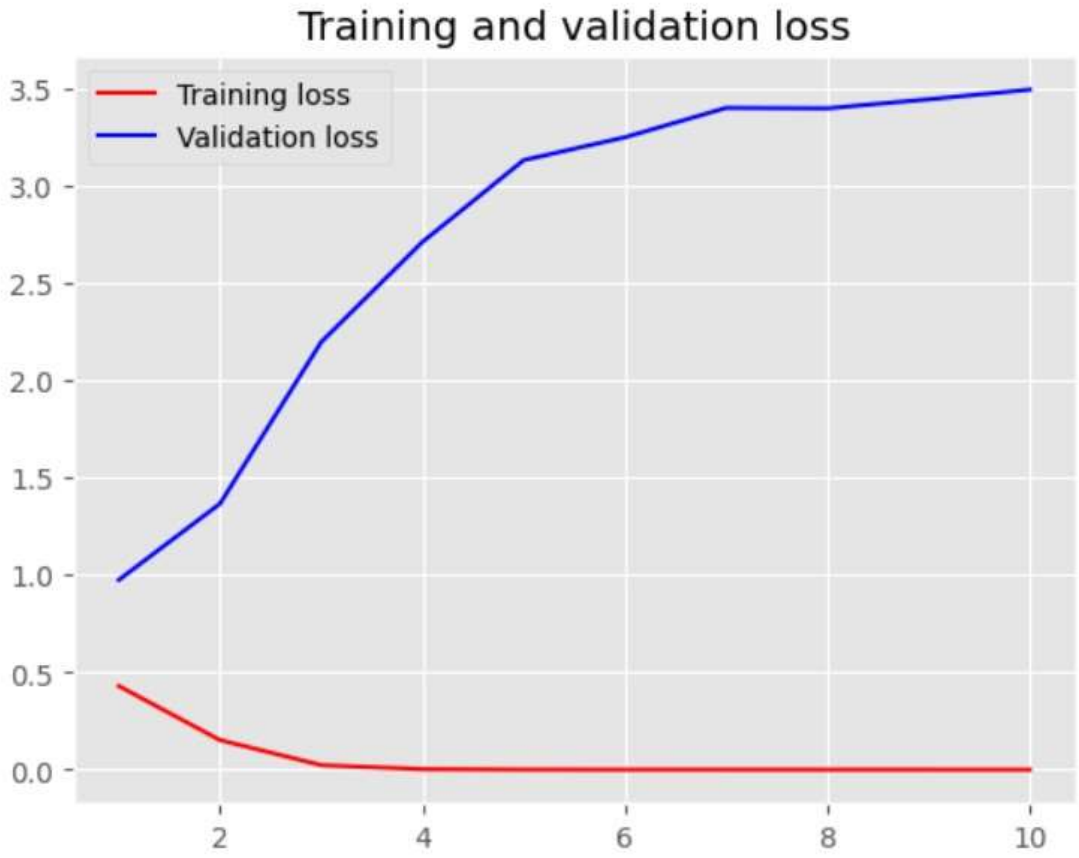




The training accuracy of the model steadily increased throughout 10 epochs, going from 49.46% to 75.61%, indicating that it was learning the task. Nonetheless, the validation accuracy ranged between 49.52% and 50.20%, remaining low and comparatively stable, indicating that the model had trouble generalizing to the validation set. The validation loss was likewise substantial, suggesting possible problems like overfitting or a model architecture that wasn't suitable for this specific dataset. The discrepancy between training and validation accuracy raises the possibility that to improve generalization, additional model performance enhancements like regularization strategies or data augmentation may be required.

Training Sample 10000:





During training, the model showed remarkable performance, increasing accuracy from 71.31% to 100% across 10 epochs and reducing loss from 0.5297 to almost zero by the end of the epoch. However, the validation loss grew dramatically, indicating that the model overfitted the training data, while the validation accuracy stayed low and reasonably stable, ranging between 49.4% and 50.34%. This shows that the model had trouble generalizing to the validation data even if it was able to memorize the training set flawlessly. The accuracy of 83.09% in the test set results suggested a possible overfitting problem while also showing some degree of generalization.

Highest Test Accuracy:

With 10,000 samples, the test accuracy reached its maximum of 0.8309.

Why did the 10000 samples performed best:

- A larger sample size usually yields a more varied set of instances, improving the model's ability to generalize. The model performed well on the test set even though it overfitted on the training set (achieving 100% accuracy).
- Due to their reliance on capturing global word-to-word relationships, GloVe embeddings perform best in large datasets. This is because larger corpora are more reliable.

Smaller datasets, on the other hand, may result in overfitting or inadequate training if the model lacks the diversity necessary to identify intricate patterns. The model with 10,000 samples achieved the best test accuracy because it was more likely to generalize effectively to new data.

Results:

S No	Technique Used	Training Sample Size	Training Accuracy (%)	Test Loss
1	Custom Trained embedding layer	100	97.1	0.36
2	Custom Trained embedding layer	5000	96.5	0.35
3	Custom Trained embedding layer	1000	96.8	0.34
4	Custom Trained embedding layer	10000	96.2	0.34
5	Pretrained word embedding (GloVe)	100	74.2	0.73
6	Pretrained word embedding (GloVe)	5000	78.2	0.91
7	Pretrained word embedding (GloVe)	1000	75.61	0.89
8	Pretrained word embedding (GloVe)	10000	100	0.95