Assignment – 4

Text and Sequence Data

Rana

Objective:

The objective of the binary classification task for the IMDB dataset is to categorize movie reviews as either positive or negative. The dataset consists of 50,000 reviews, with the top 10,000 most frequent words being considered. Training is conducted with different sample sizes, including 100, 1,000, 5,000, and 100,000 samples, while validation is performed using 10,000 samples. The data has been preprocessed and then input into a pretrained embedding model along with the embedding layer, where various approaches are tested to evaluate performance.

Data:

The dataset preparation process converts each review into a series of word embeddings, with each word represented by a fixed-length vector.

▪ This results in a limit of 10,000 samples. Additionally, instead of using a string of words, a sequence of numbers corresponding to individual words is generated from the reviews. Although I have the list of numbers, the neural network cannot directly use it as input.

▪ Tensors must be constructed using numerical values. One way to handle the integer list is by creating a tensor containing samples and word indices in integer format.

▪ To achieve this, I need to ensure that all samples have the same length, which means padding reviews with dummy words or numbers to make them consistent in siz**e.**

Method Used:

For the IMDB dataset, I identified two distinct methods for generating word embeddings:

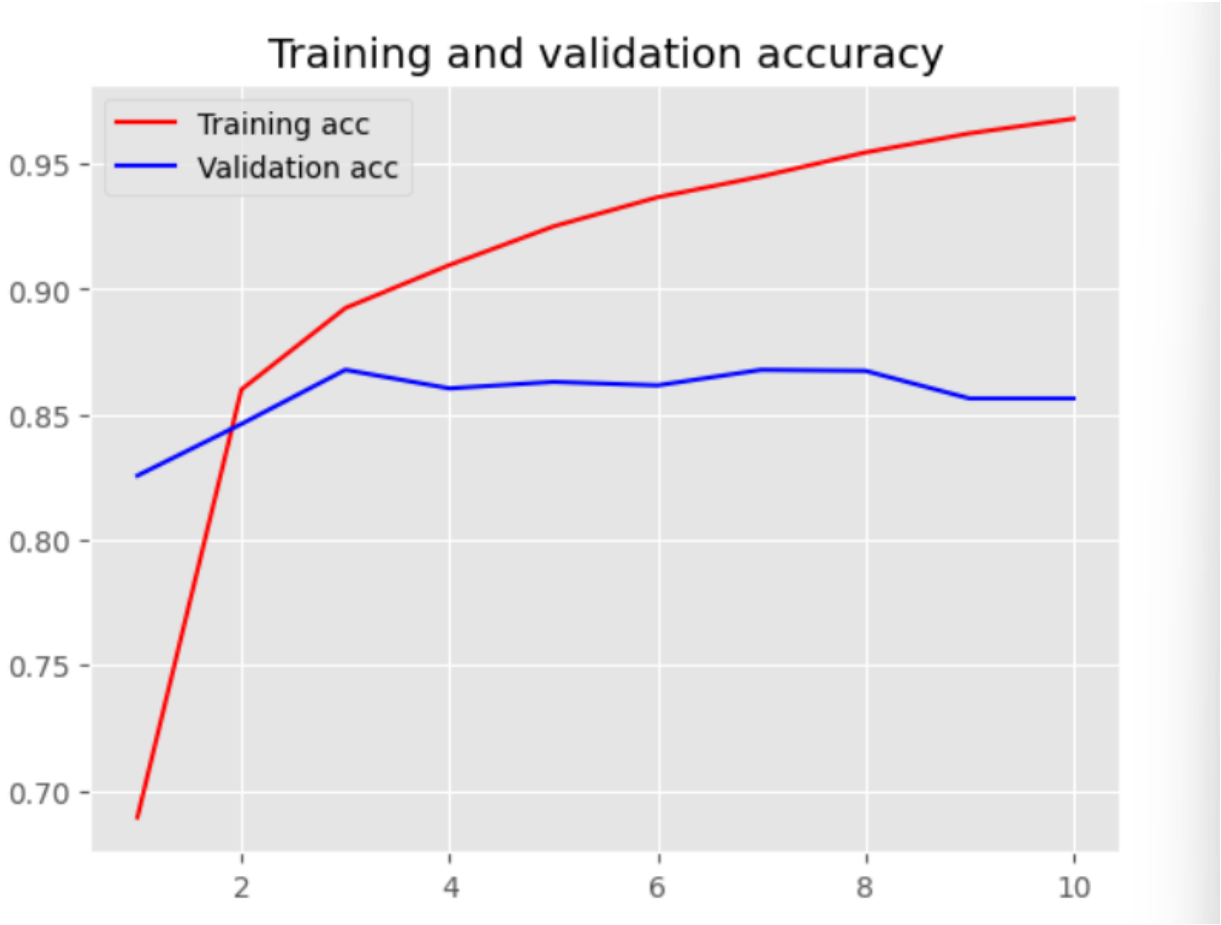
1. Custom-trained embedding layer

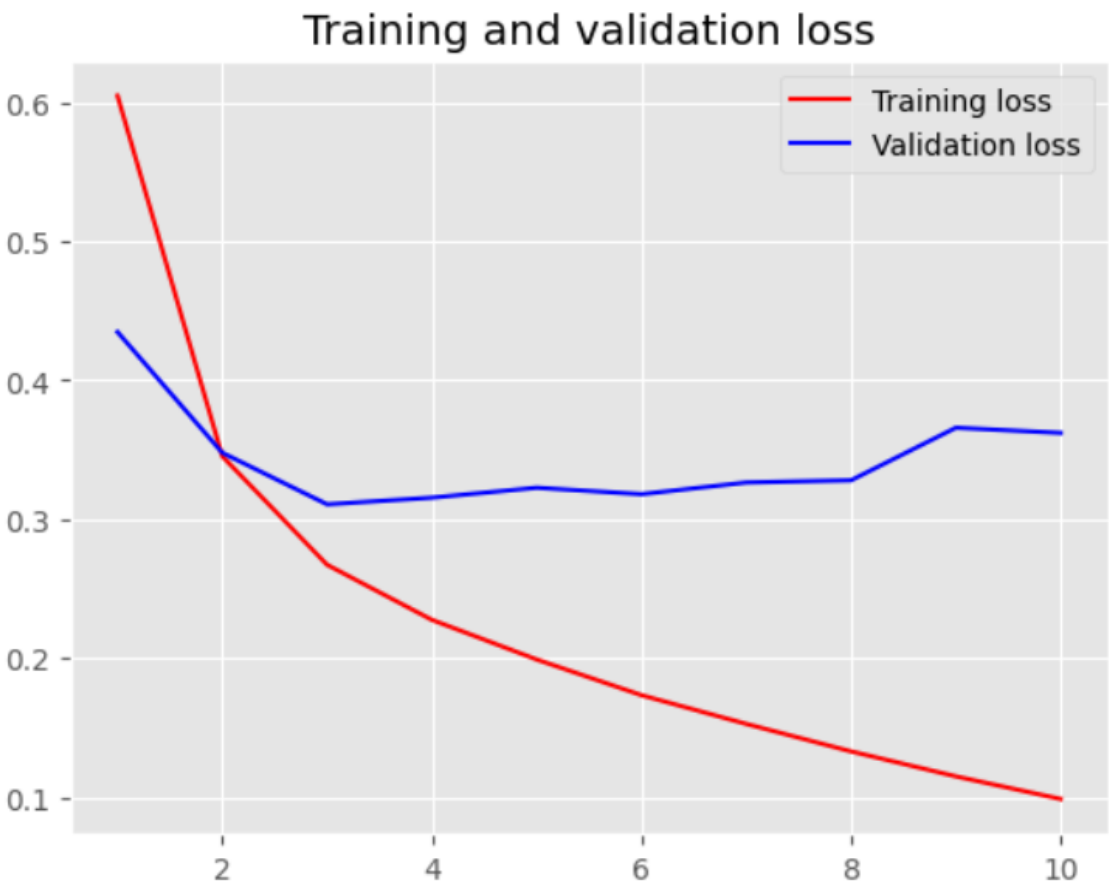
2. Pretrained word embedding layer using the GloVe model.

In this study, we utilized the widely used GloVe pretrained word embedding model, which has been trained on a large corpus of text. We evaluated accuracy across different sample sizes—100, 1,000, 5,000, and 10,000—by comparing both custom-trained and pretrained embedding layers on the IMDB dataset. The models, using either pretrained or custom-trained embeddings, were tested on IMDB reviews of varying sample sizes, with accuracy assessed on the test sets.

CUSTOM-TRAINED EMBEDDING LAYER:

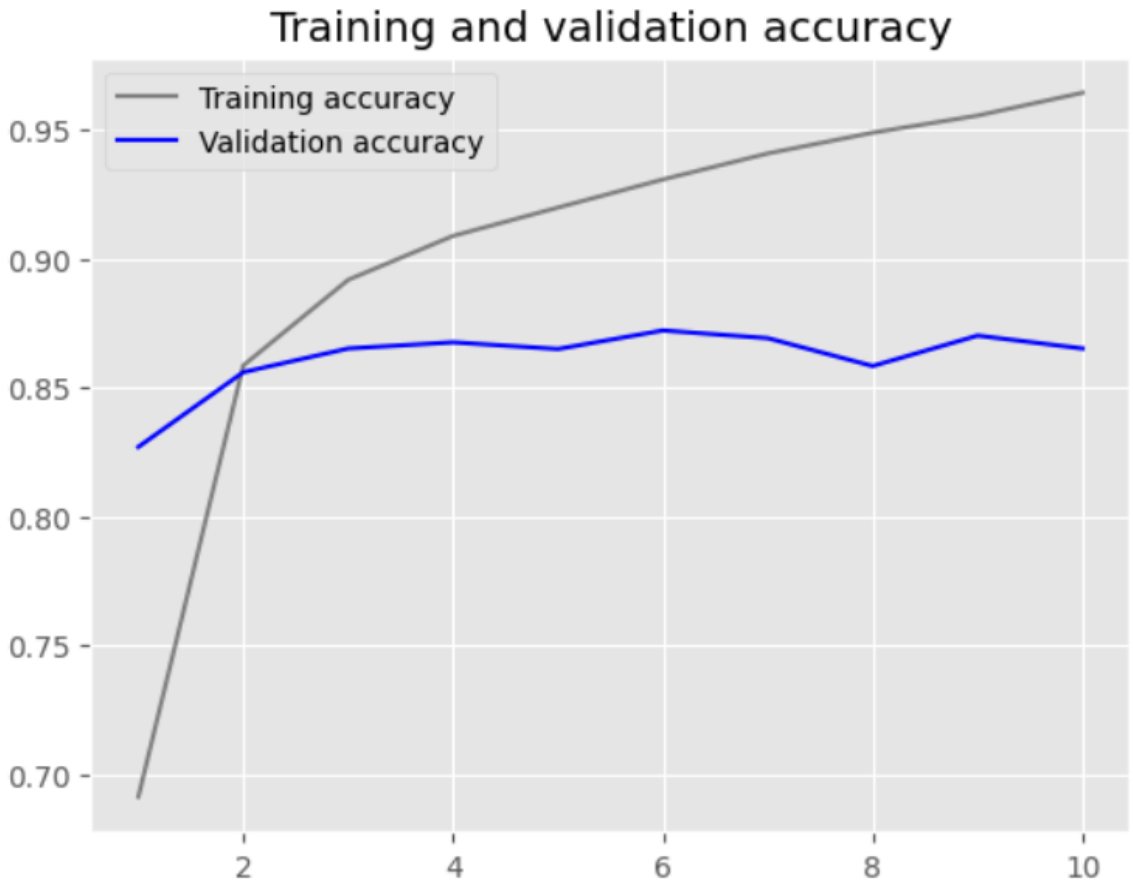
Training Dataset with 100 samples

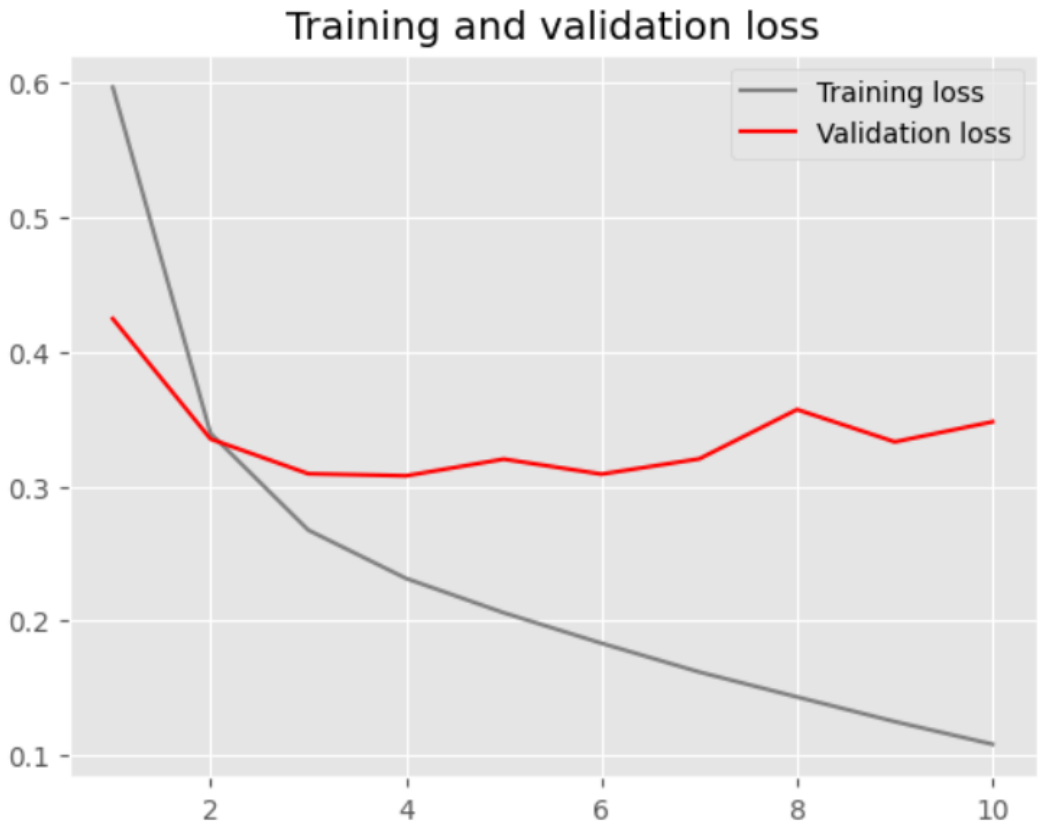

Training and validation accuracy
Training acc
Validation acc
0.95
0.90
0.85
0.80
0.75
0.70
2
4
6
8
10



The model was trained for 10 epochs, with the training accuracy steadily improving from 60.43% in the first epoch to 97.14% by the 10th epoch. The training loss decreased from 0.6665 to 0.0932, indicating good convergence. On the validation set, the accuracy started at 82.56% and fluctuated slightly, peaking at 86.78% by the 7th epoch, before settling at 85.64% in the final epoch. The validation loss showed a similar trend, decreasing from 0.4347 to 0.3621. When evaluated on the test set, the model achieved a test accuracy of 85.72% with a test loss of 0.3629, showing decent generalization despite the small variations in validation accuracy.

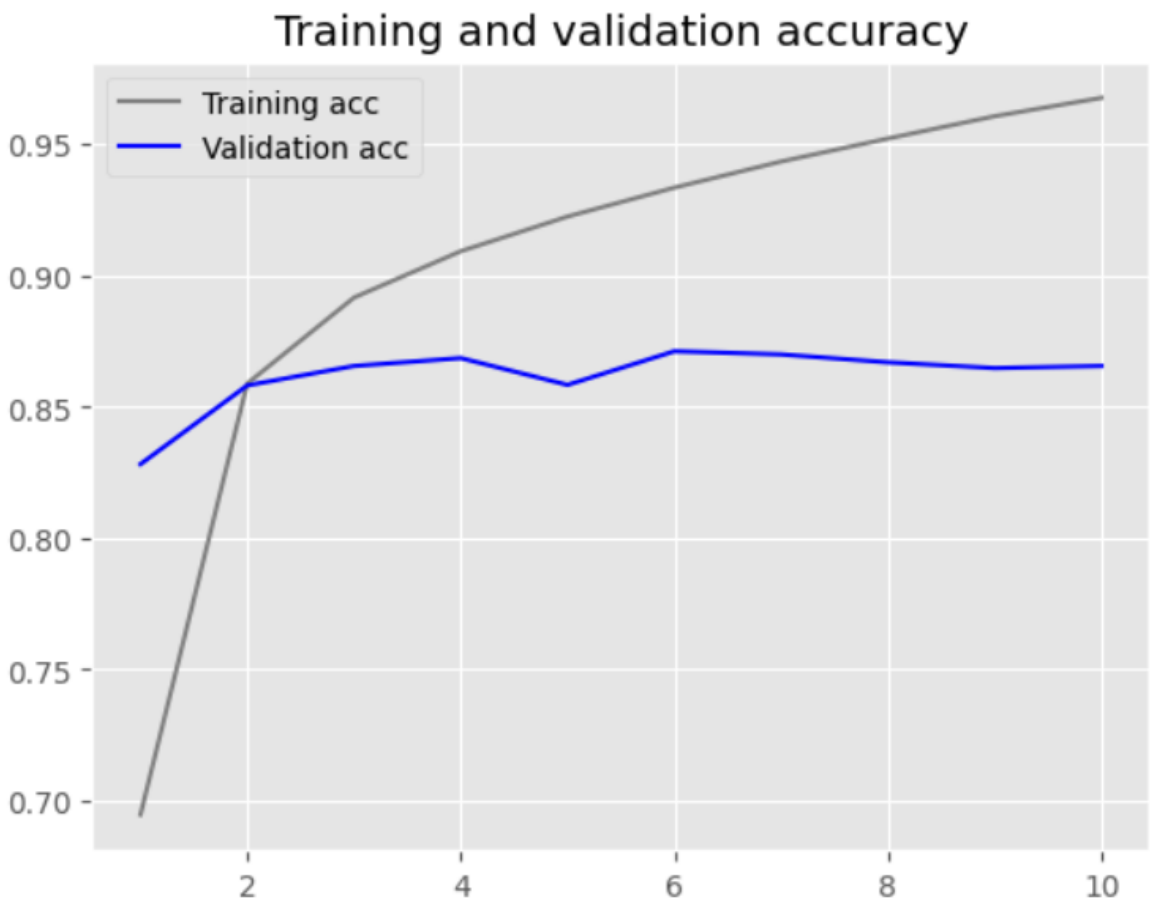
Training Dataset with 5000 samples:

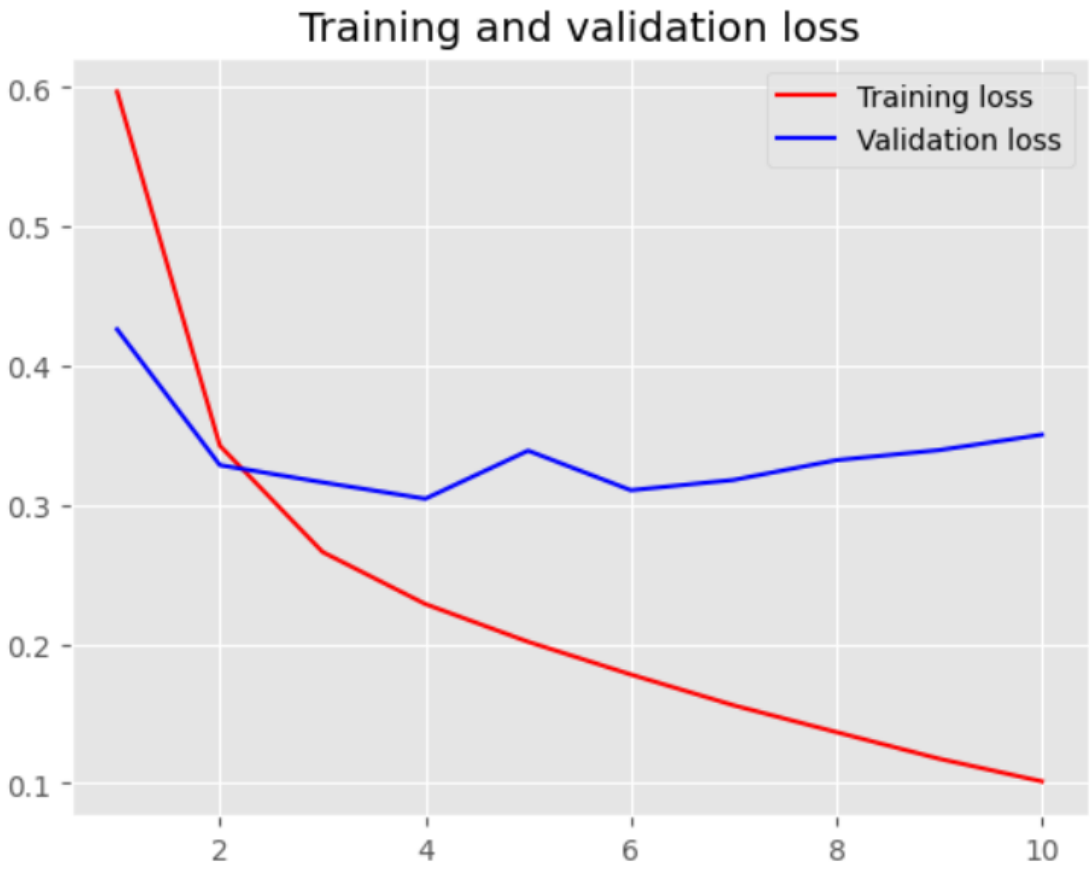

Training and validation accuracy
Training accuracy
Validation accuracy
0.95
0.90
0.85
0.80
0.75
0.70
2
4
6
8
10



The model was trained for 10 epochs, with the training accuracy improving from 59.01% in the first epoch to 96.55% by the 10th epoch. The training loss decreased from 0.6649 to 0.1077, indicating effective learning. On the validation set, accuracy started at 82.70% and showed some fluctuations, peaking at 87.22% in the 6th epoch, before stabilizing at 86.52% by the final epoch. The validation loss showed a similar pattern, decreasing from 0.4246 to 0.3479. When evaluated on the test set, the model achieved a test accuracy of 86.30% with a test loss of 0.3515, reflecting good generalization performance despite the variations observed in the validation accuracy.

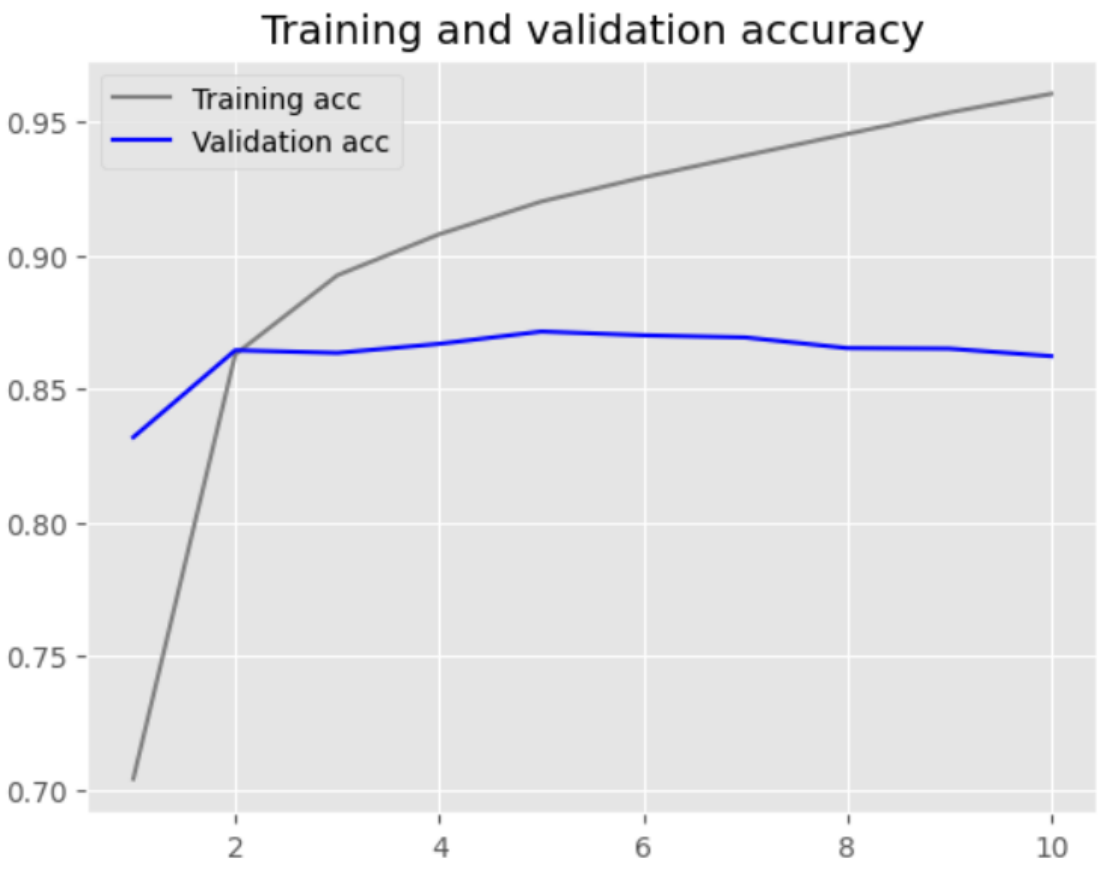
Training Dataset with 1000 samples:

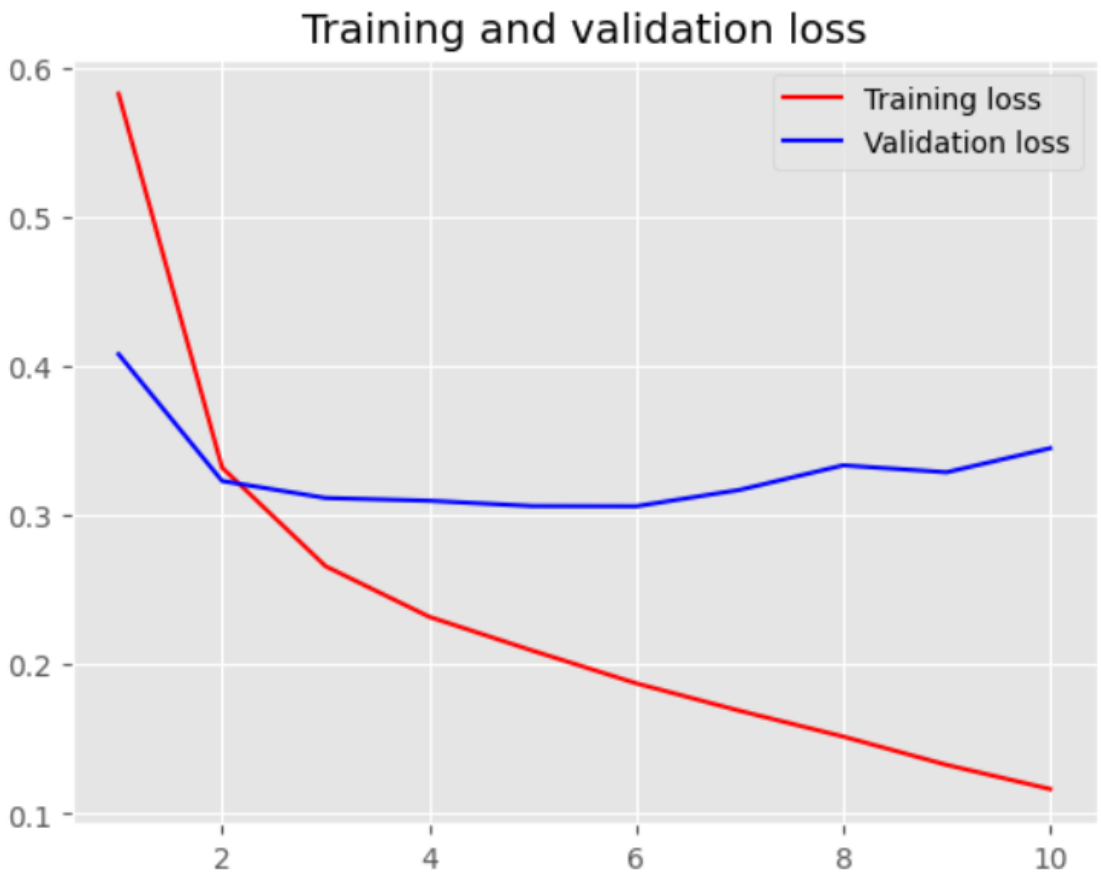

Training and validation accuracy
Training acc
Validation acc
0.95
0.90
0.85
0.80
0.75
0.70
2
4
6
8
10



The model was trained for 10 epochs, starting with a training accuracy of 59.99% in the first epoch and reaching 96.88% by the 10th epoch, with a corresponding decrease in training loss from 0.6639 to 0.0995. Validation accuracy started at 82.82% and showed gradual improvements, peaking at 87.12% in the 6th epoch, before stabilizing around 86.56% by the end of training. The validation loss decreased from 0.4262 to 0.3504. When evaluated on the test set, the model achieved a test accuracy of 86.41% with a test loss of 0.3487, demonstrating good performance and generalization despite fluctuations in validation accuracy.

Custom-trained embedding layer with training sample size = 10000


Training and validation accuracy
Training acc
Validation acc
0.95
0.90
0.85
0.80
0.75
0.70
2
4
6
8
10



The model was trained for 10 epochs, beginning with a training accuracy of 60.49% in the first epoch and increasing to 96.23% by the 10th epoch, with the corresponding training loss decreasing from 0.6575 to 0.1136. Validation accuracy started at 83.18% and showed steady improvement, reaching a peak of 87.14% in the 5th epoch, before gradually declining to 86.22% by the end of training. The validation loss decreased from 0.4079 to 0.3449, indicating a generally successful fit despite some fluctuations in validation performance with a test accuracy of 86 percent

Highest Test Accuracy:

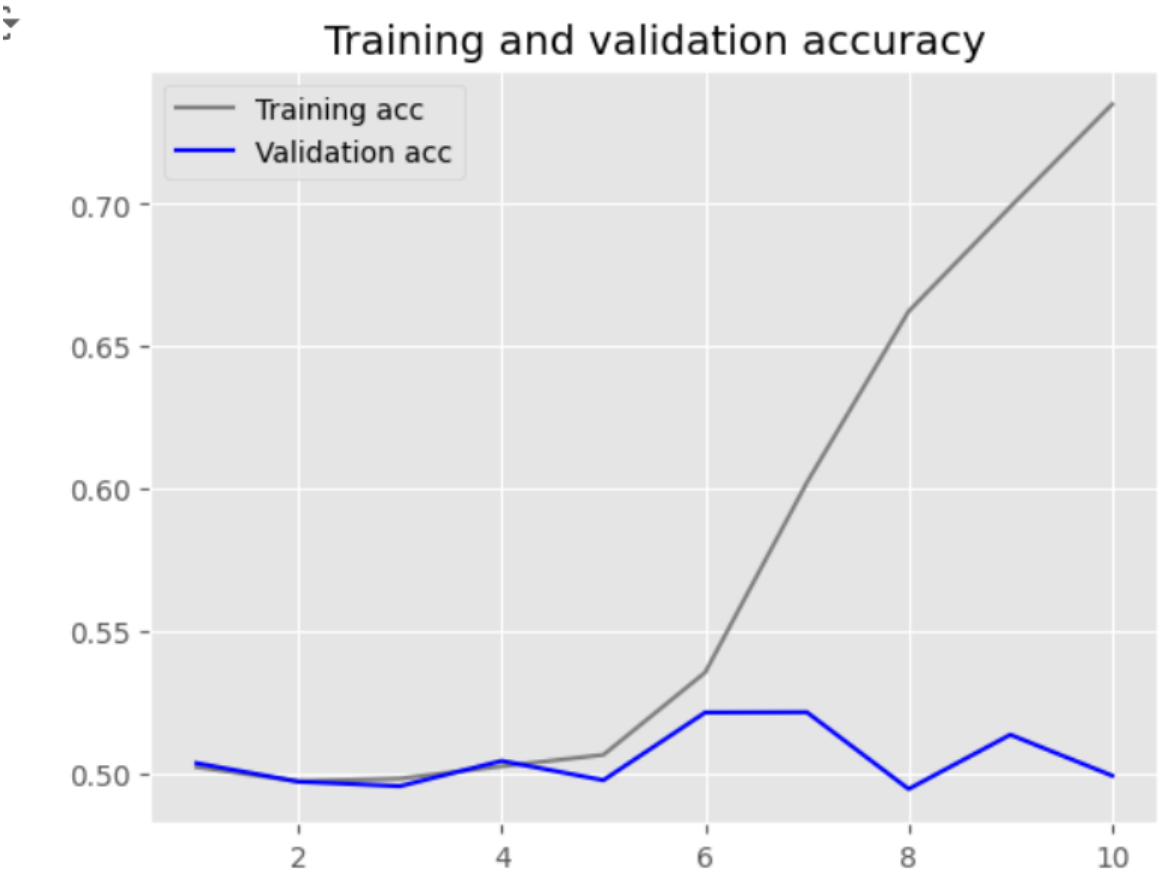
- The **highest test accuracy** is **86.58%**, which was achieved with **sample size 1000**.

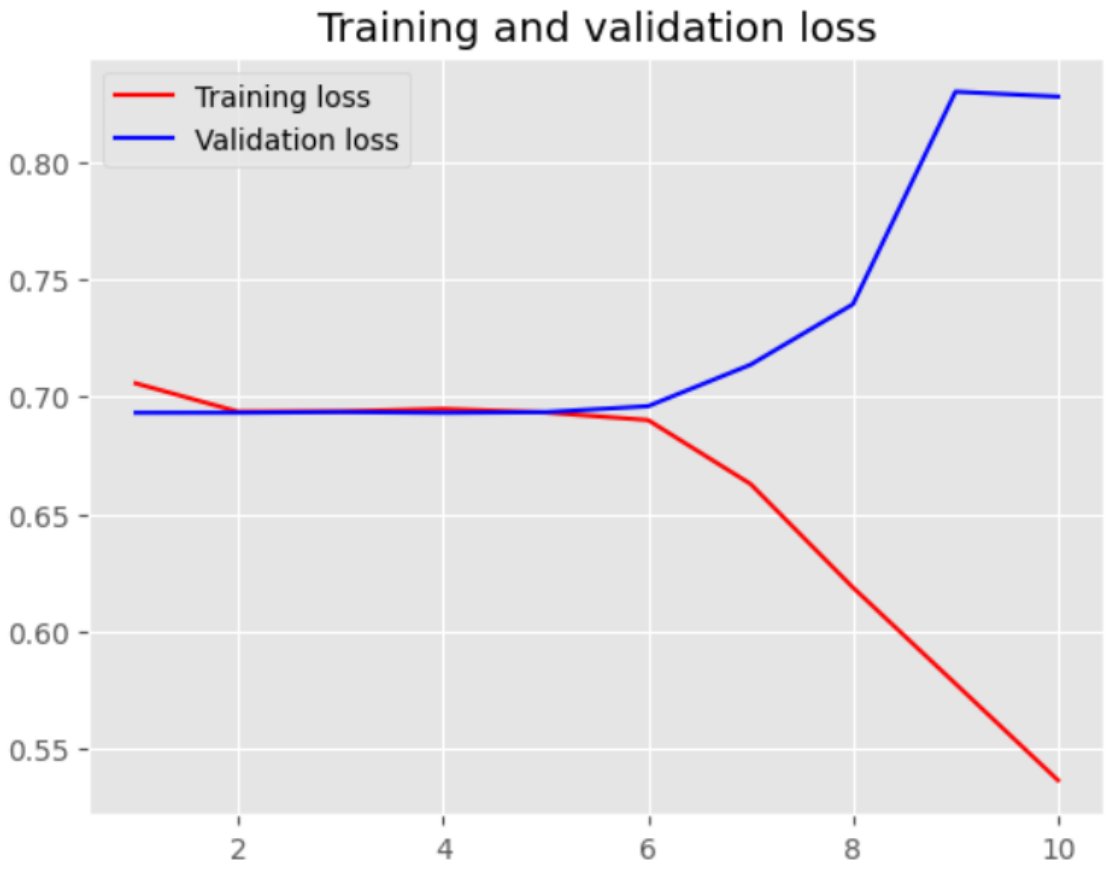
Precision Variation:

- The precision (test accuracy) appears to increase slightly as the sample size increases, but the variation between different sample sizes is minimal.
- The accuracy fluctuates between **85.72%** (sample size 100) and **86.58%** (sample size 1000), which is a **0.86% increase** from the lowest to the highest.

PRETRAINED WORD EMBEDDING LAYER:

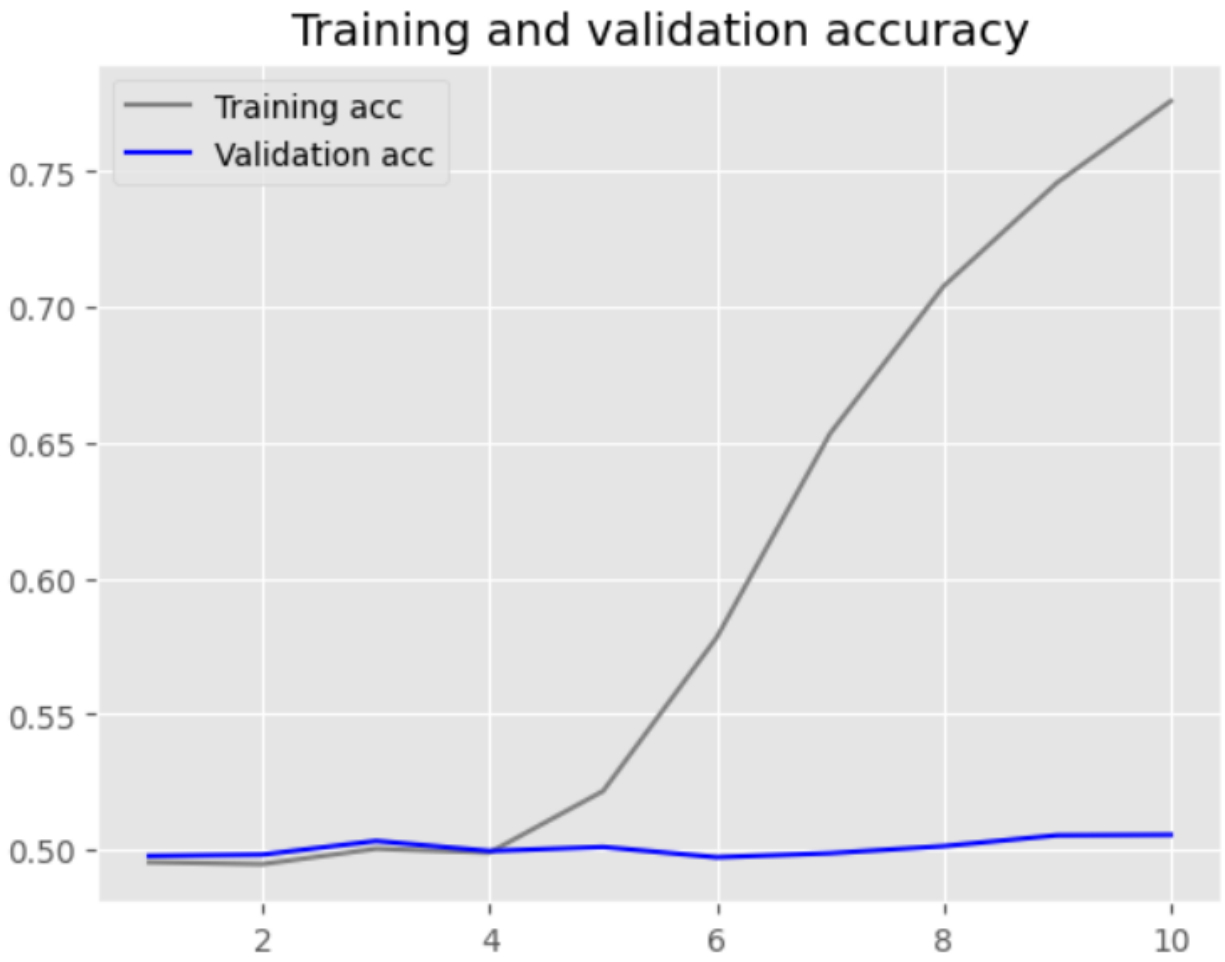
Training Sample Size 100

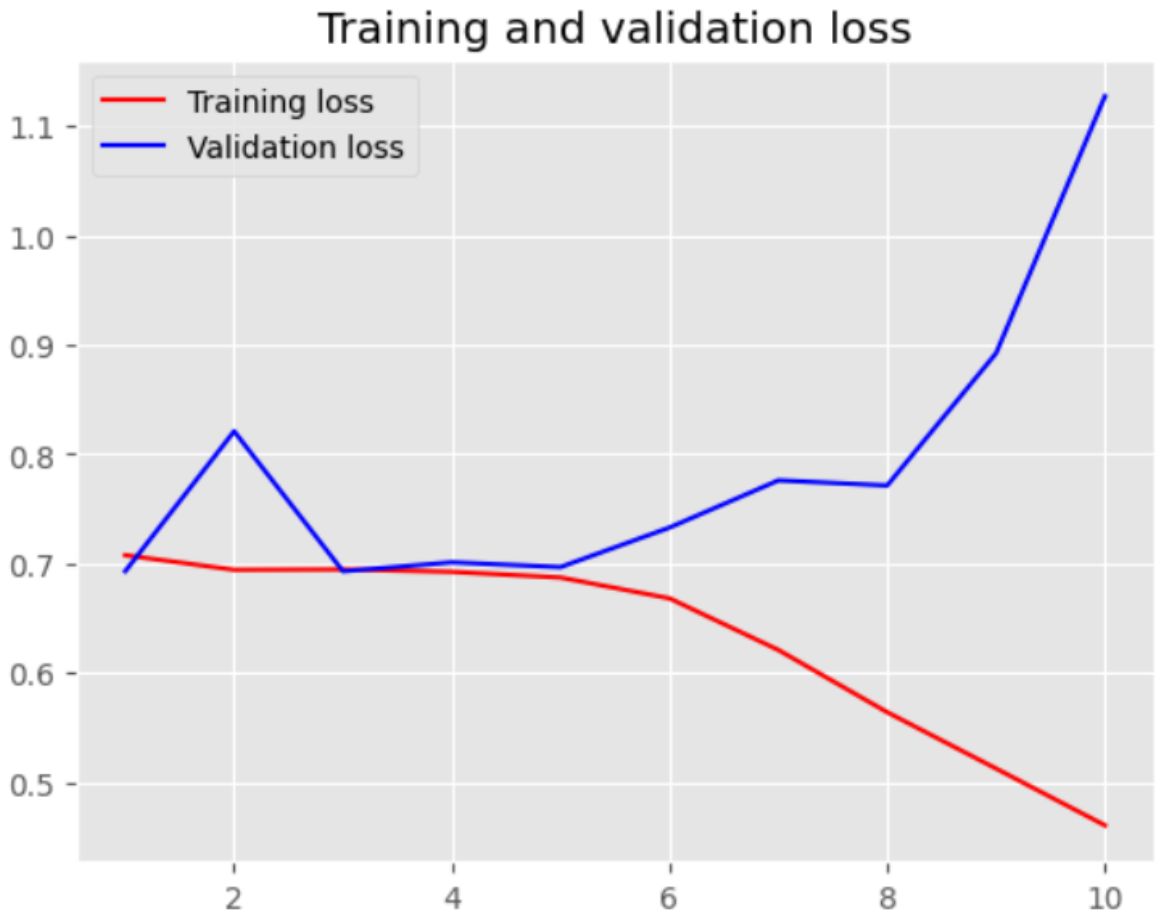




The model's training accuracy steadily improved from 50.26% to 74.21% over 10 epochs, while validation accuracy remained between 49.43% and 52.12%, indicating poor generalization. Despite the increasing training performance, the validation accuracy fluctuated, suggesting overfitting. The test accuracy of 56.51% is slightly better than the validation performance but still relatively low, with a test loss of 0.7322, reflecting the model's struggle to generalize well to unseen data. These results suggest the model's limited ability to generalize and might benefit from further adjustments, such as regularization techniques or more data.

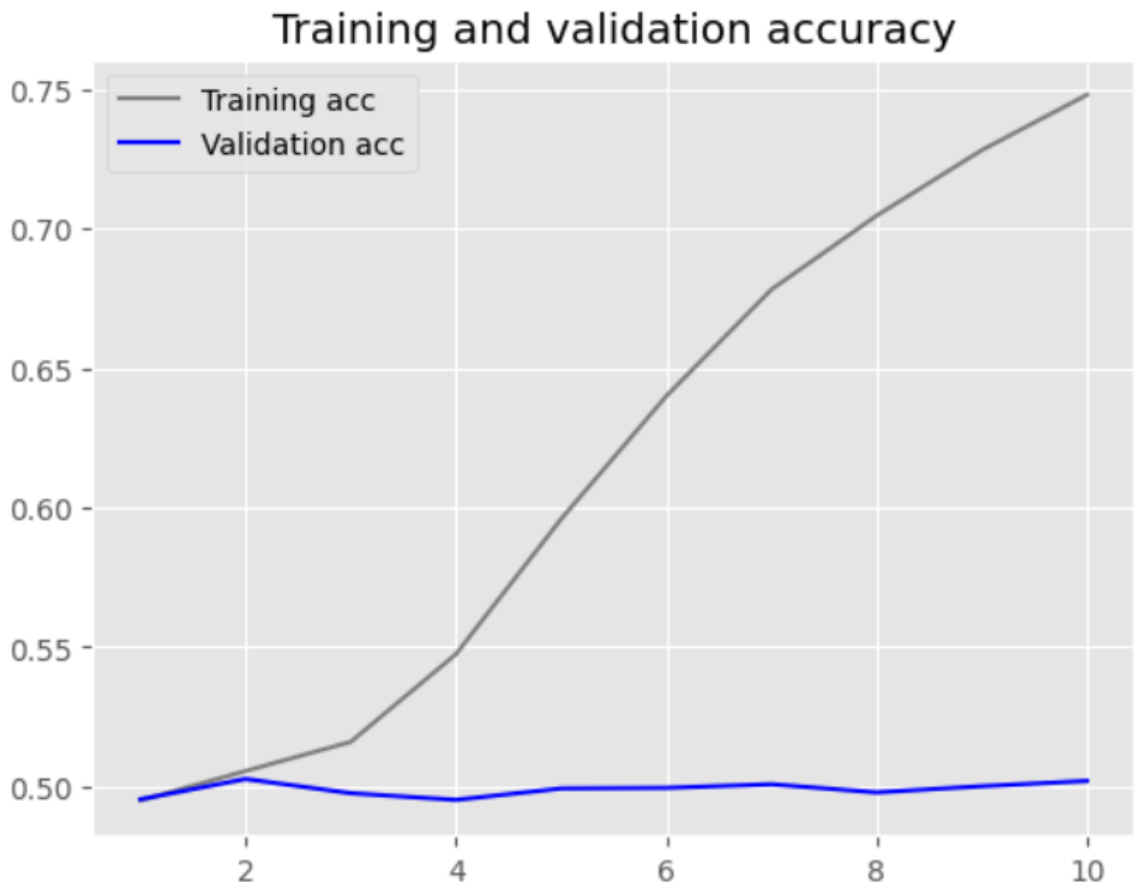
Training Sample 5000:

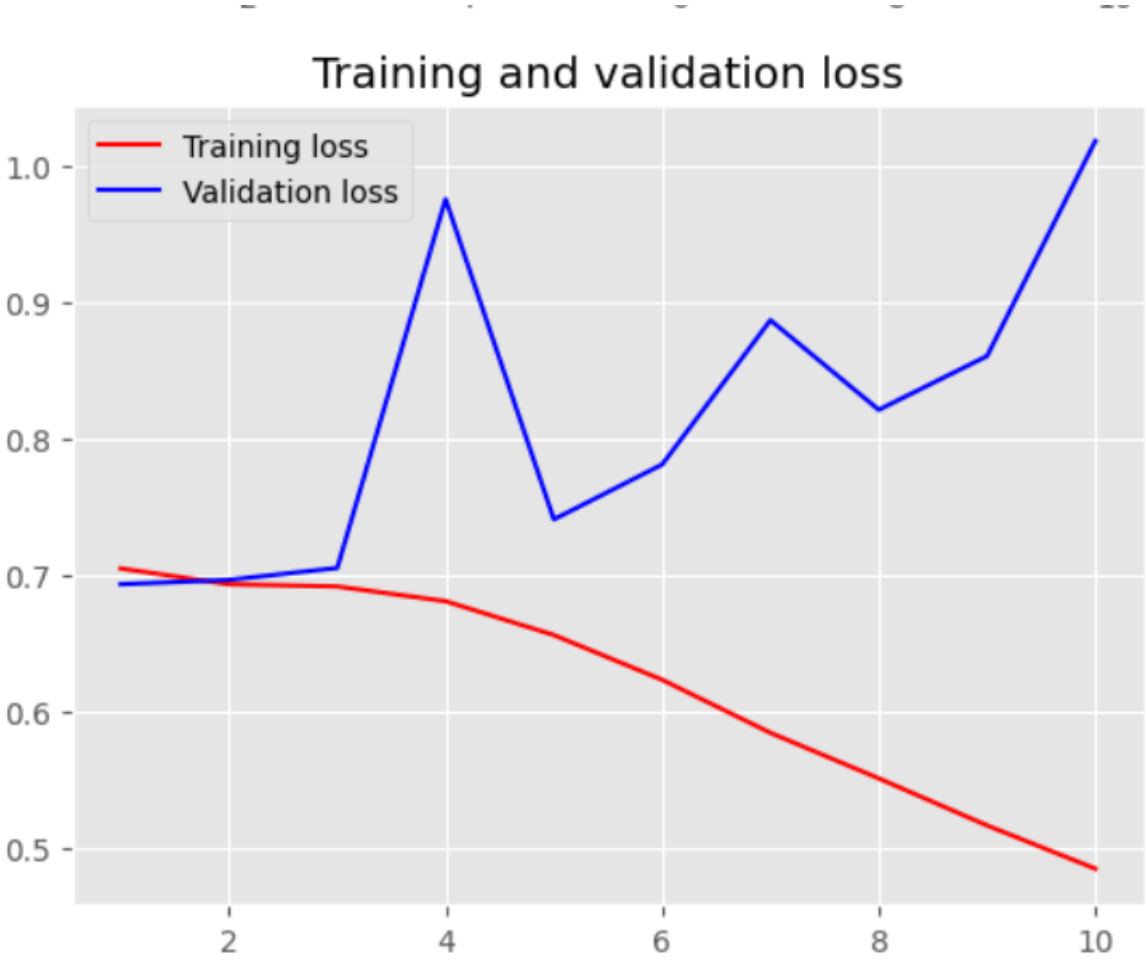

Training and validation accuracy
Training acc
Validation acc
0.75
0.70
0.65
0.60
0.55
0.50
2
4
6
8
10



The model's training accuracy improved from 49.31% to 78.29% over 10 epochs, indicating progress in learning the task. However, the validation accuracy remained consistently low, fluctuating between 49.43% and 50.56%, suggesting that the model struggled to generalize well to the validation set. The test accuracy of 56.73%, with a test loss of 0.9146, is similarly low, reflecting the model's inability to generalize effectively to unseen data. The increasing training accuracy paired with stagnant validation and test performance indicates potential overfitting, and further improvements might be necessary, such as data augmentation or regularization techniques.

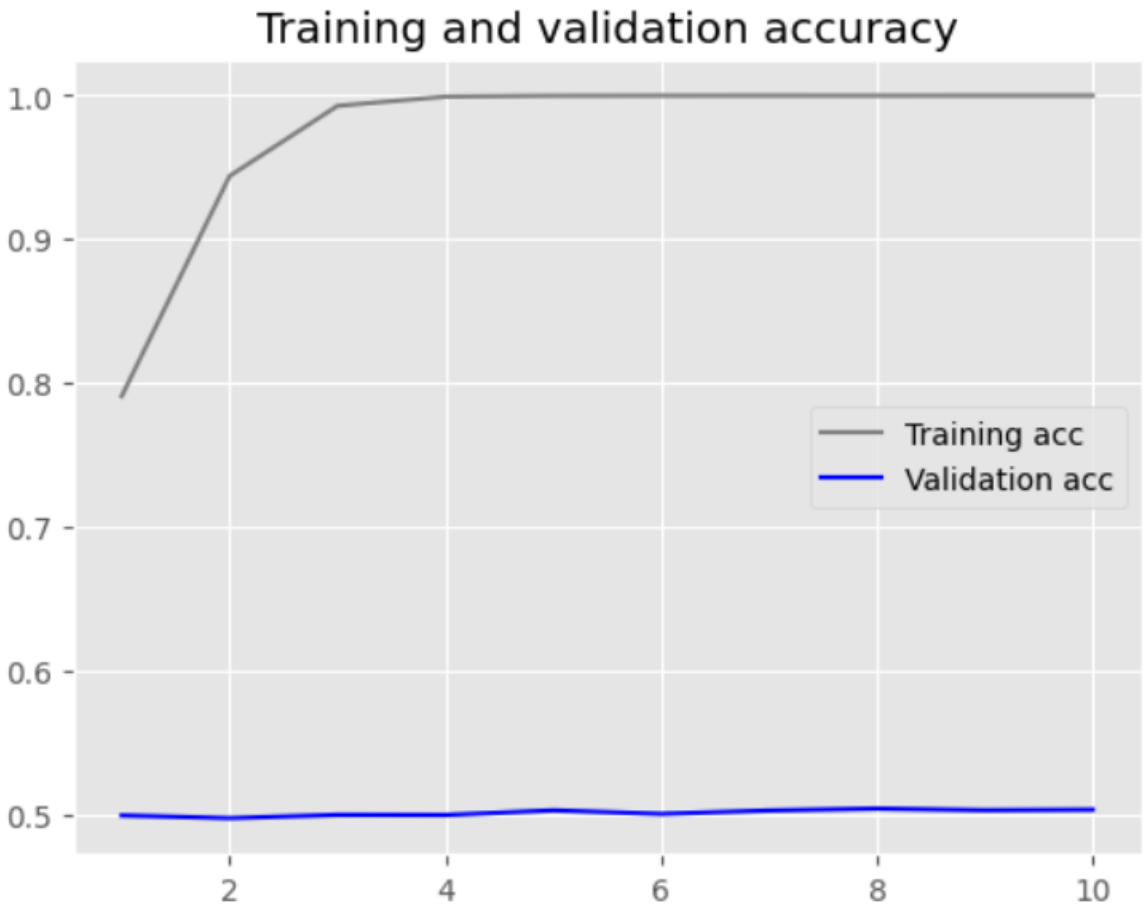
Training Sample 1000:

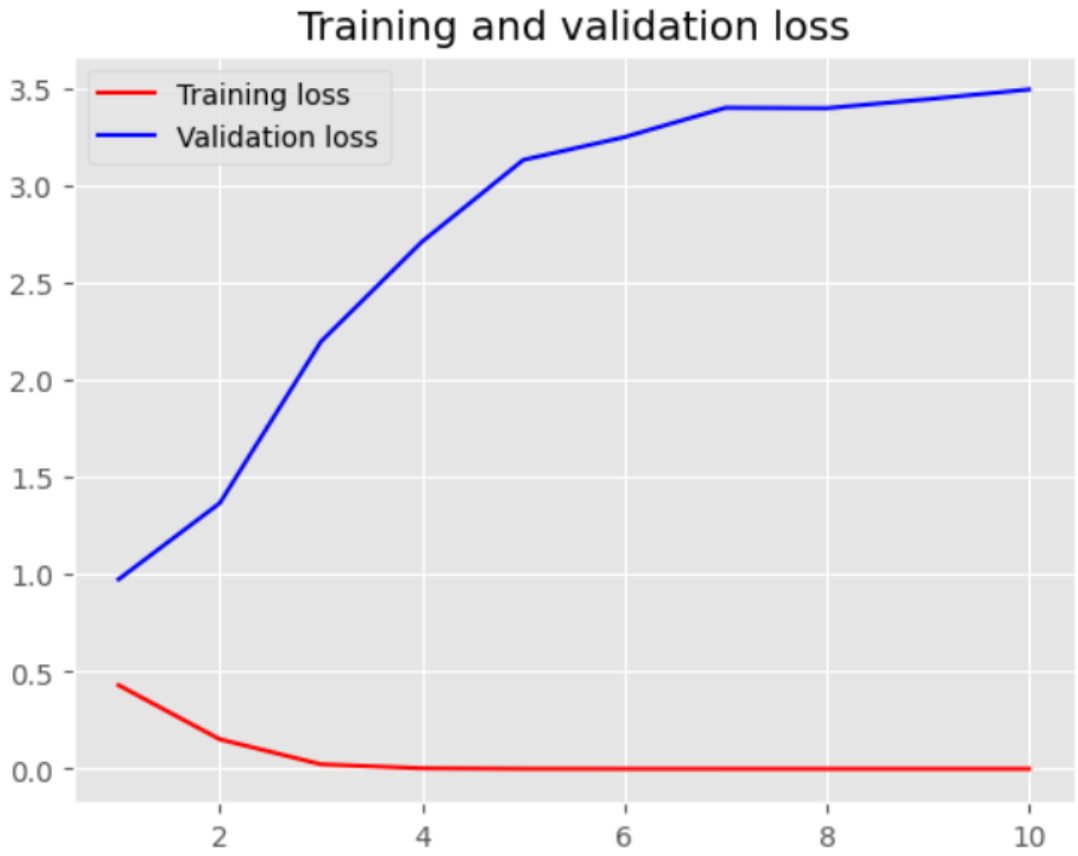

Training and validation accuracy
Training acc
Validation acc
0.75
0.70
0.65
0.60
0.55
0.50
2
4
6
8
10



The model's training accuracy gradually improved from 49.46% to 75.61% over 10 epochs, reflecting steady progress in learning the task. However, the validation accuracy remained relatively stable and low, fluctuating between 49.52% and 50.20%, suggesting that the model faced challenges in generalizing to the validation set. The loss also remained high for the validation set, indicating potential issues such as overfitting or inadequate model architecture for this particular dataset. The gap between training and validation accuracy suggests that further improvements in model performance, such as regularization techniques or data augmentation, may be necessary to enhance generalization**.**

Training Sample 10000:


Training and validation accuracy
1.0
0.9
0.8
0.7
0.6
0.5
2
4
6
8
10
Training acc
Validation acc



The model demonstrated exceptional performance during training, with accuracy rising from 71.31% to 100% over the course of 10 epochs, and the loss decreasing from 0.5297 to a near-zero value by the final epoch. Despite this, the validation accuracy remained relatively stable and low, fluctuating between 49.4% and 50.34%, and the validation loss increased significantly, suggesting the model overfitted to the training data. This indicates that although the model was able to memorize the training set perfectly, it struggled to generalize to the validation data. The test set results showed an accuracy of 83.09%, indicating some degree of generalization but still highlighting a potential overfitting issue.

Highest Test Accuracy:

The **highest test accuracy** was achieved with **10000 samples**, reaching **0.8309**.

Why the 10000 samples performed best:

- **Larger sample size** typically provides a more diverse range of examples, which allows the model to generalize better. Despite the model overfitting on the training set (reaching 100% accuracy), it was still able to perform well on the test set.

- GloVe embeddings work effectively when the dataset is large, as they are based on capturing global word-to-word relationships, which are more reliable with larger corpora.

In contrast, smaller datasets can lead to overfitting or insufficient training, as the model doesn't have enough variety to capture complex patterns. The model with **10000 samples** had a higher chance to generalize well on unseen data, leading to the highest test accuracy.

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