Assignment 4

Text and Sequence Data

Summary

Goal:

Sorting film reviews into positive and negative categories is the aim of the binary classification problem for the IMDB dataset. Only the top 10,000 words out of the dataset's 50,000 reviews are taken into account. Between 100 to 100,000 samples are used for training, and 10,000 samples are used for validation. Preparing the data comes before using a pre-trained embedding model and placing the embedding layer. Various approaches are investigated to evaluate performance.

Data Preprocessing:

Each review is converted into word embeddings as part of the dataset preparation process, in which each word is represented by a fixed-size vector. There is a 10,000 sample limit on this painstaking technique. Furthermore, the reviews are used to create a numerical sequence in which individual numbers represent discrete words rather than entire word strings. Nevertheless, this sequential sequence of integers is not easily supported by the neural network's input structure.

Tensors must be created via the number sequence in order to overcome this difficulty. It may be possible to create a tensor with an integer data type and a certain structure using the list of integers, arranged as samples and word indices. To do this, though, it is necessary to make sure that every sample stays the same length. To standardize the length across all samples, techniques like padding reviews with dummy words or number placeholders must be used.

Process:

Using the IMDB dataset as a backdrop, I investigated two different approaches for creating word embeddings in this work.

- An embedding layer with custom training.
- GloVe model-based pre-trained word embedding layer.

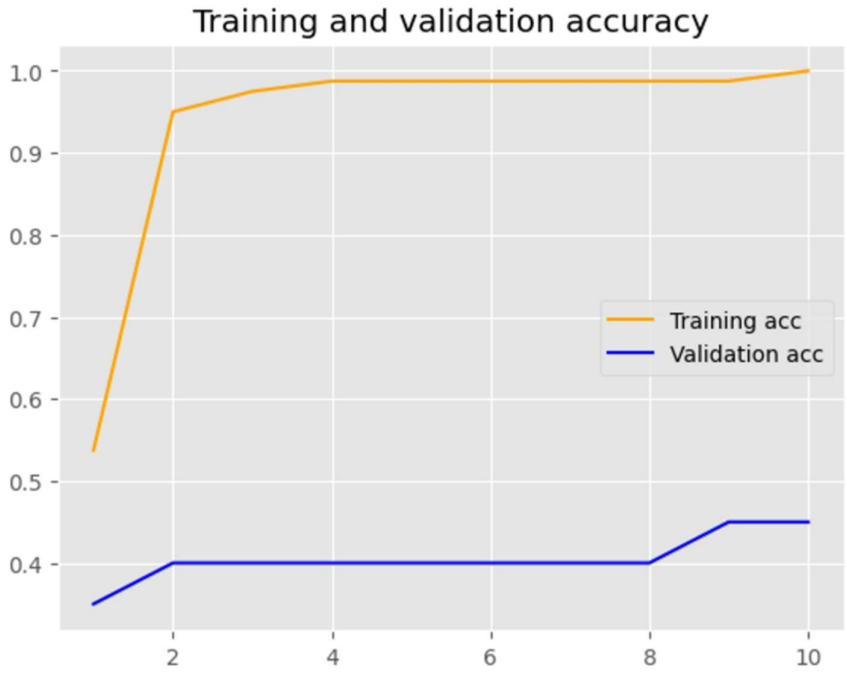
Our study used the GloVe model, a well-liked option for word embeddings that is trained on sizable textual datasets.

I used the IMDB review dataset to test two different embedding layers for various embedding methodologies. One had a pre-trained word embedding layer, and the other included a custom-trained layer. I evaluated the accuracy of these two models using training sample sizes of 100, 5000, 1000, and 10,000 in order to determine their efficacy.

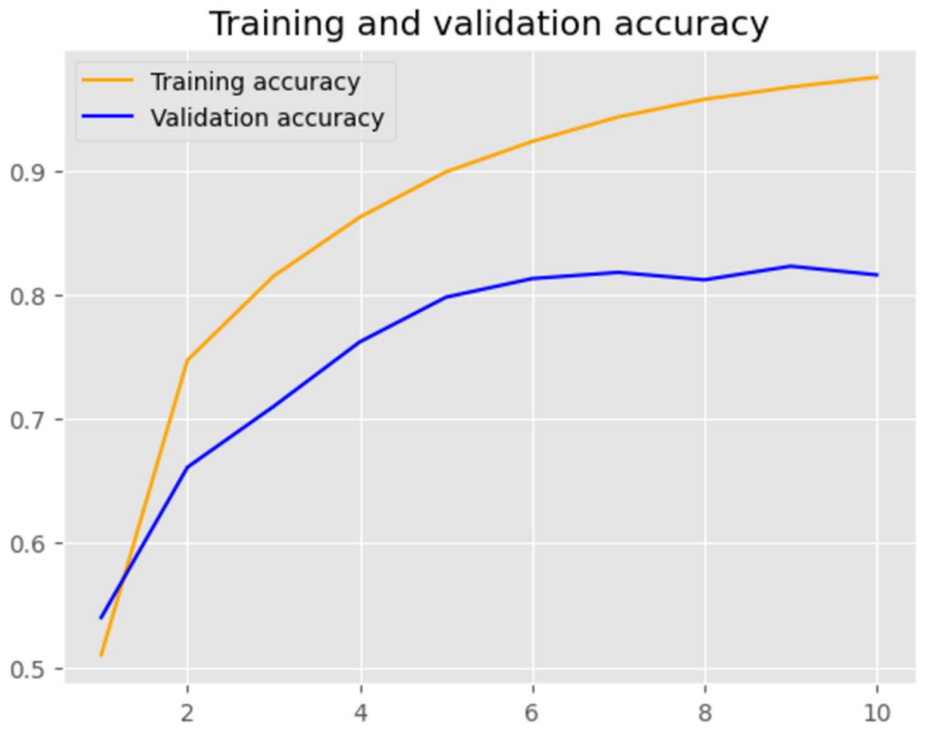
First, using the IMDB review dataset, we created a specially trained embedding layer. Each model was then trained using a variety of dataset samples, and its accuracy was evaluated using a specific testing set. After this assessment, we compared these precision outcomes with those of a model that had a pre-trained word embedding layer and was tested similarly over a range of sample sizes.

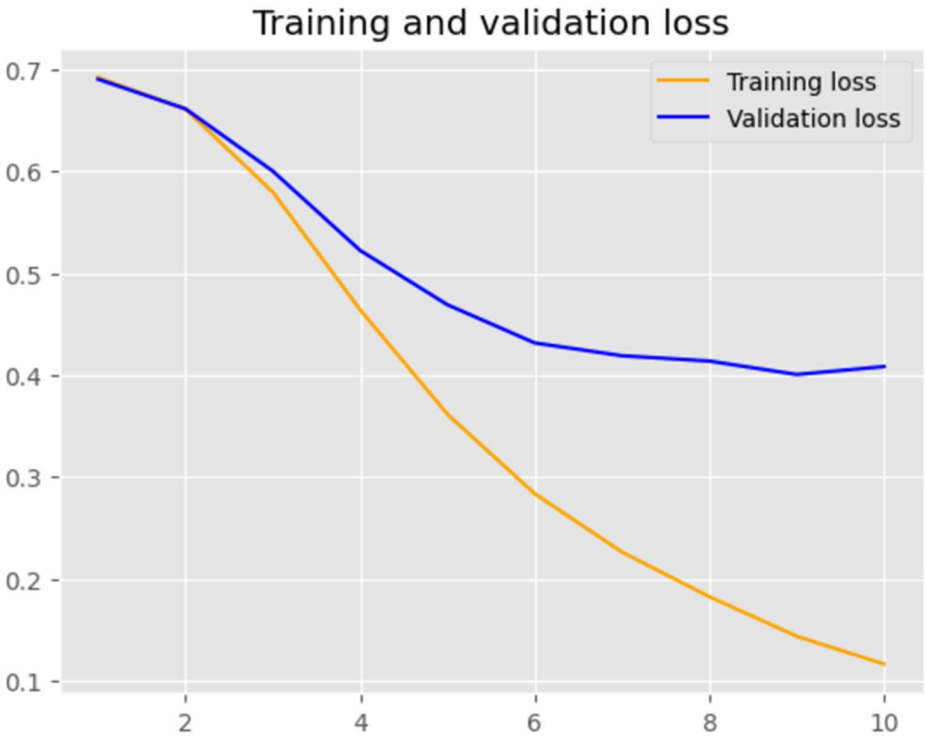
CUSTOM-TRAINED EMBEDDING LAYER

1. Custom-trained embedding layer with training sample size =100

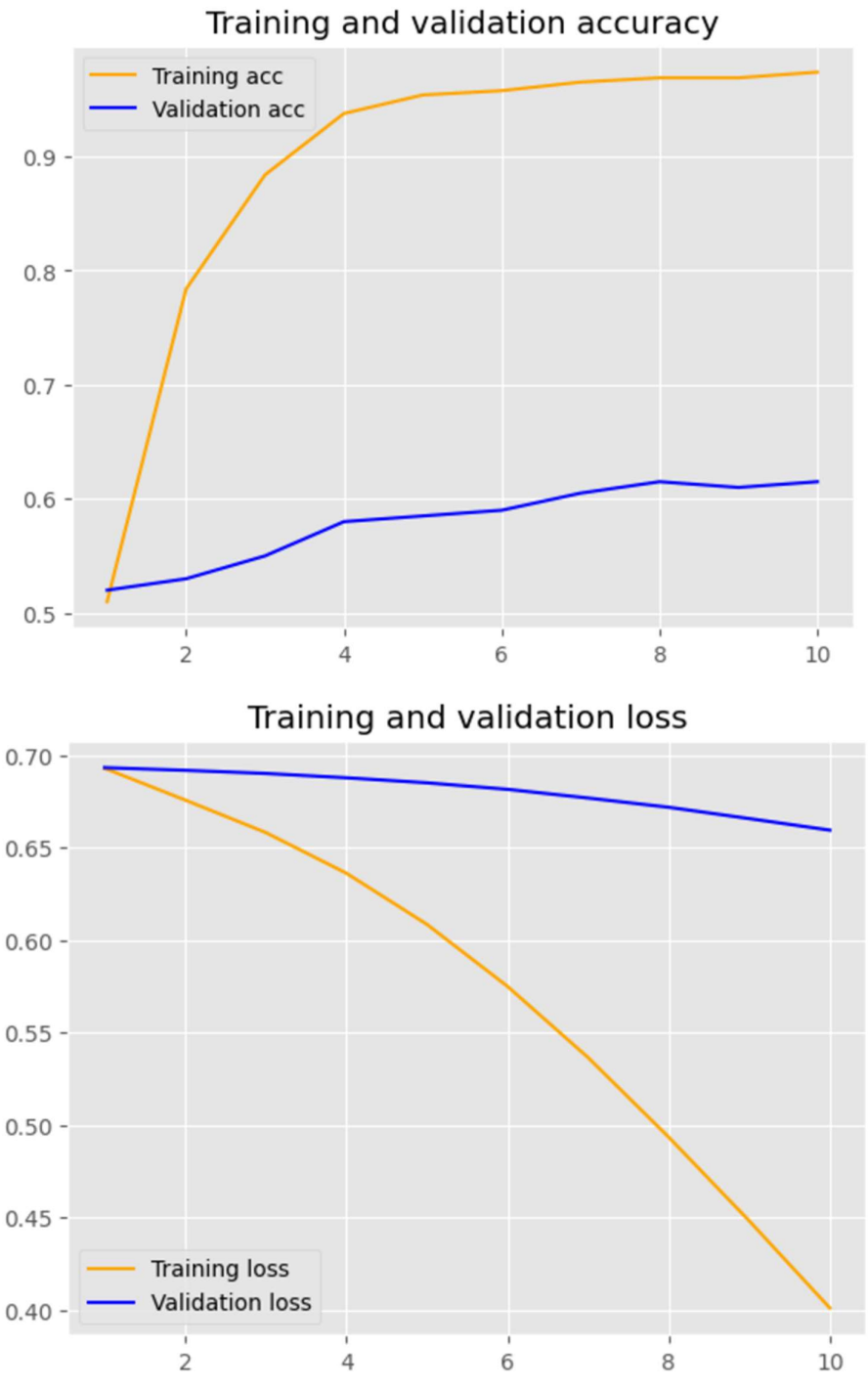


2. Custom-trained embedding layer with training sample size = 5000





3. Custom-trained embedding layer with training sample size = 1000

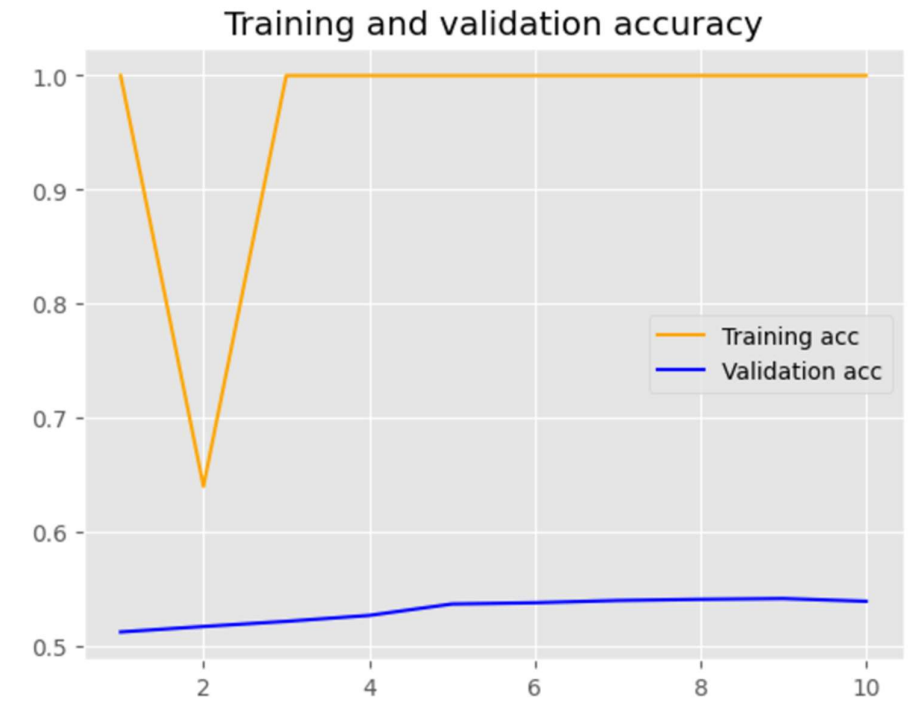


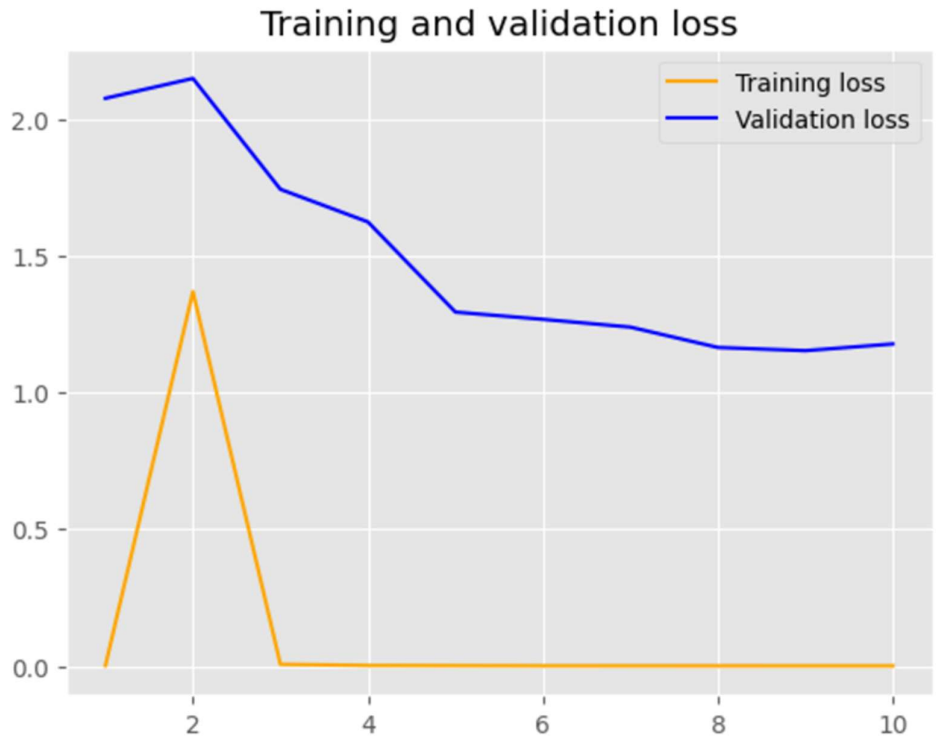
4. Custom-trained embedding layer with training sample size = 10000

The accuracy ranged from 96.92% to 100% using the custom-trained embedding layer, depending on the training sample size. With a training sample size of 100, the greatest accuracy of 100% was attained.

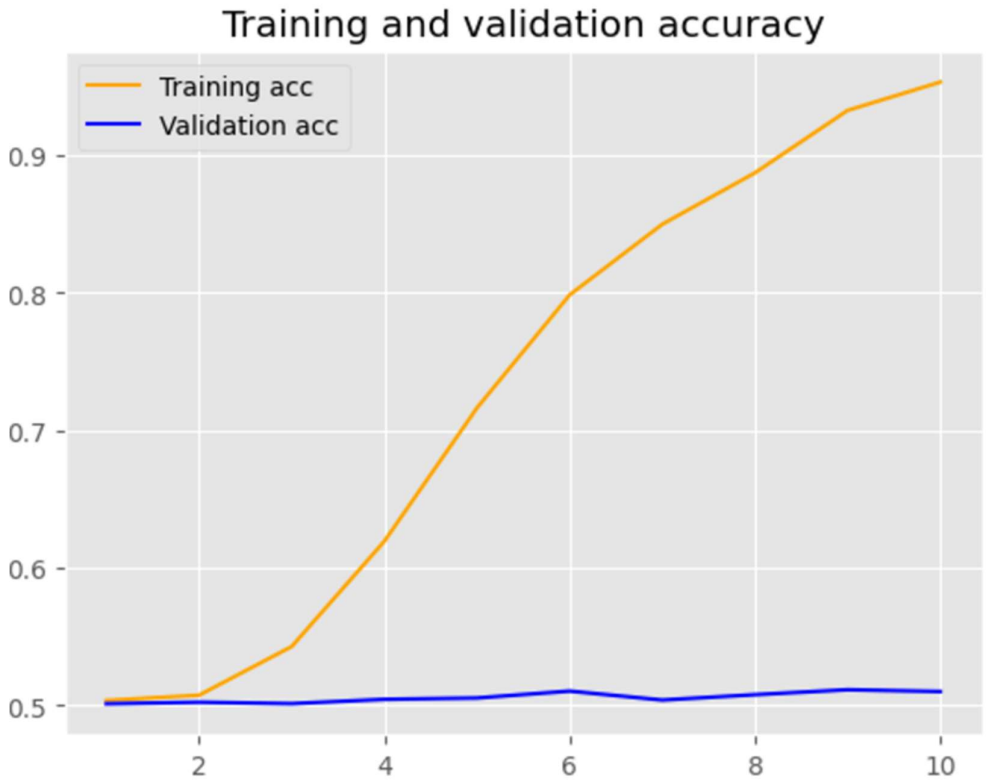
PRETRAINED WORD EMBEDDING LAYER

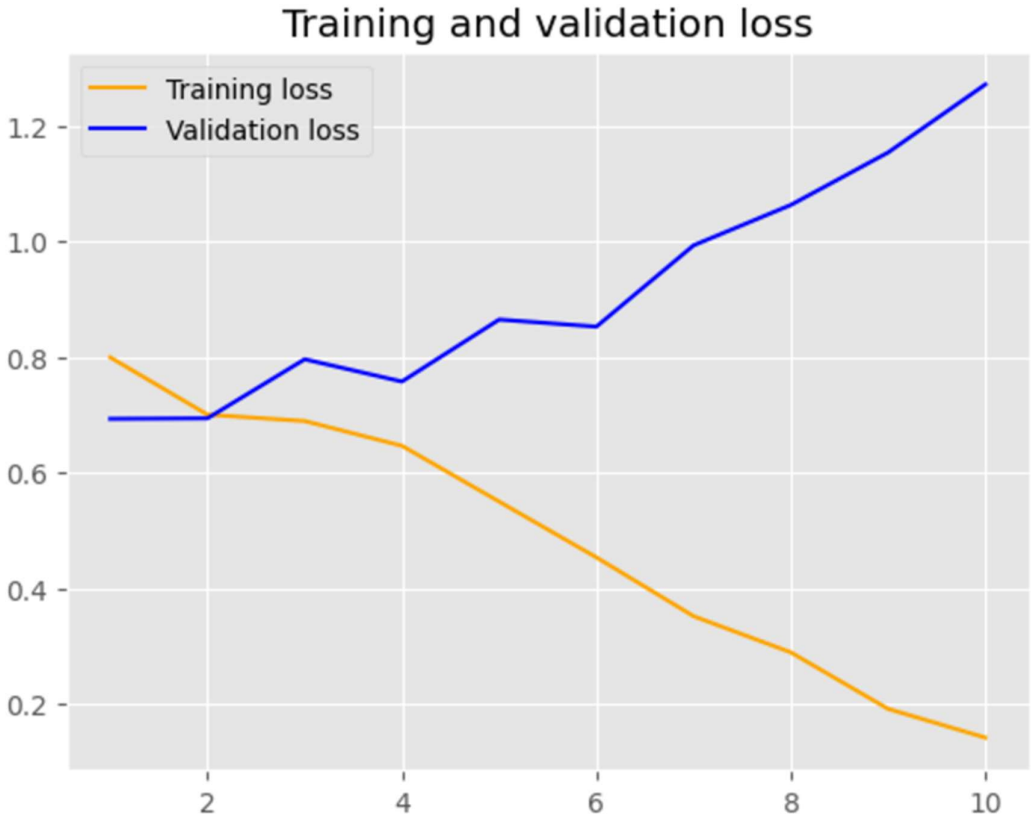
1. Pretrained word embedding layer with training sample size = 100



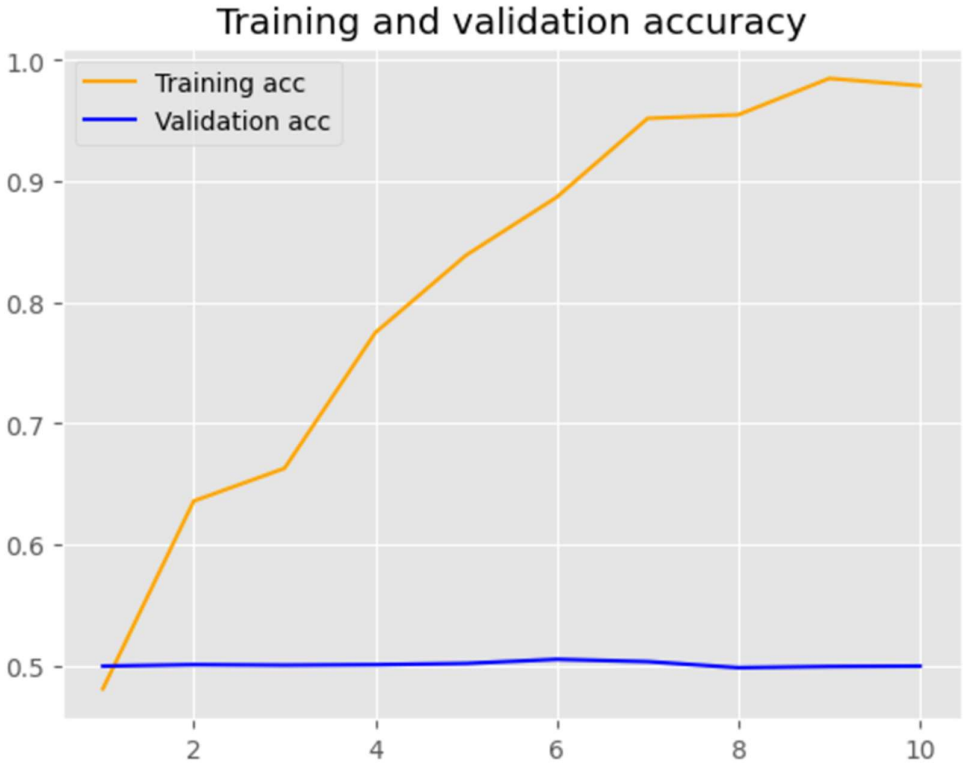


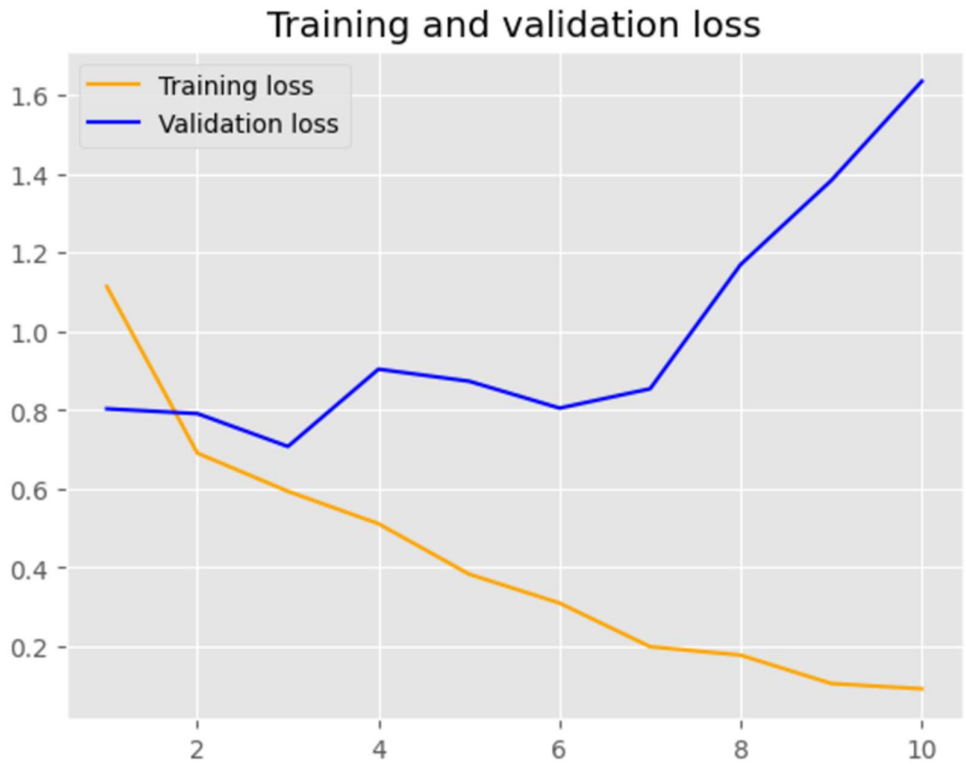
3. Pretrained word embedding layer with training sample size = 5000



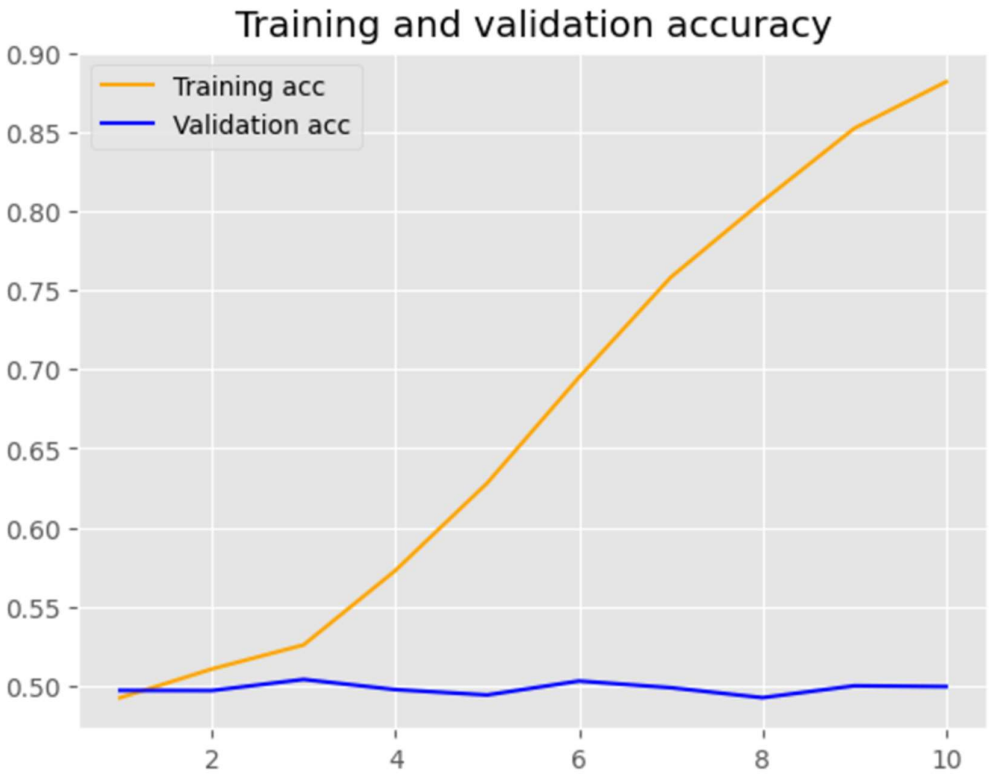


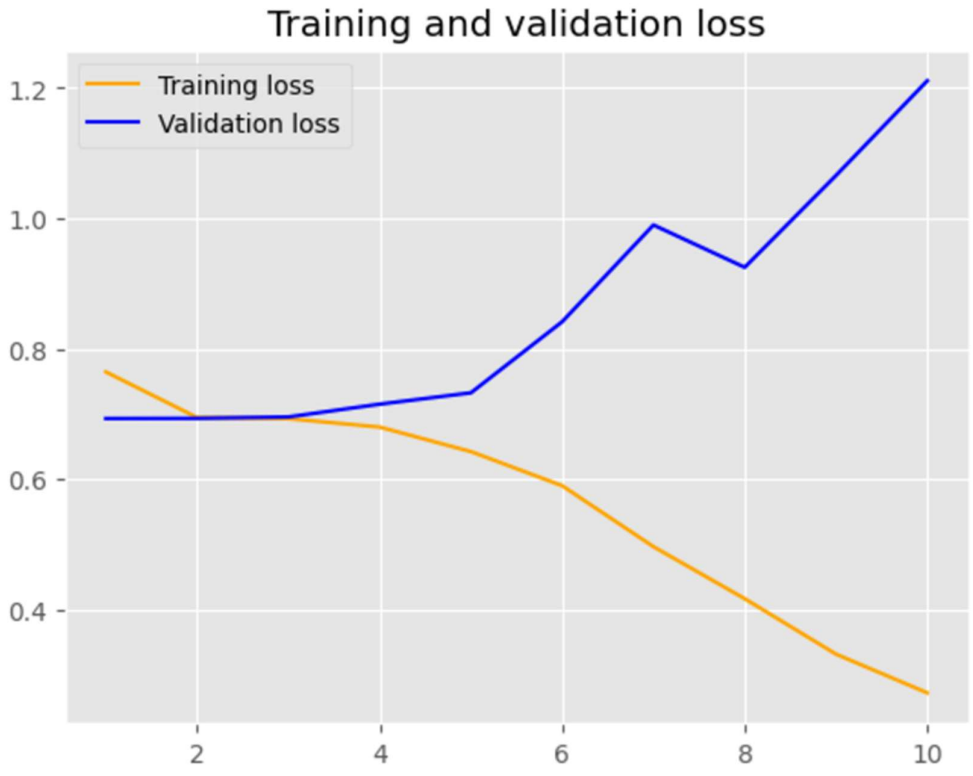
4. Pretrained word embedding layer with training sample size = 1000





5. Pretrained word embedding layer with training sample size = 10000


Training and validation accuracy
Training acc
Validation acc
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
2
4
6
8
10


Training and validation loss
Training loss
Validation loss
1.2
1.0
0.8
0.6
0.4
2
4
6
8
10

Depending on the training sample size, the Pretrained word embedding layer (GloVe) showed accuracy values between 95.48% and 100%. With a training sample size of 100, the best accuracy—100%—was recorded. However, the model using pre-trained embeddings tended to overfit more quickly and lose accuracy as the training sample sizes grew. These results highlight how difficult it is to determine the best strategy with confidence because it greatly depends on the particular needs and constraints of the work.

Results:

Embedding Technique	Training Sample Size	Training Accuracy (%)	Test loss
Custom-trained embedding layer	100	100	0.69
Custom-trained embedding layer	5000	97.90	0.40
Custom-trained embedding layer	1000	96.92	0.65
Custom-trained embedding layer	10000	97.85	0.32
Pretrained word embedding (GloVe)	100	100	1.17
Pretrained word embedding (GloVe)	5000	95.48	1.27
Pretrained word embedding (GloVe)	1000	97.86	1.63
Pretrained word embedding (GloVe)	10000	89.02	1.21

Conclusion:

When training with larger sample numbers, the custom-trained embedding layer consistently performed better than the pretrained word embedding layer in this experiment. It's important to remember that, even with the risk of overfitting, the pretrained word embedding layer might still be a "better choice" in some situations, especially when computational resources are scarce and a small training sample size is needed.