AML ASSIGNMENT – 4

REPORT

Shloka Sabnekar

811328298

Objective:

In the IMDB dataset, the binary classification task aims to classify movie reviews into positive and negative categories. The dataset contains 50,000 reviews; only the top 10,000 words are assessed; training samples are restricted to 100, 5000, 1000, and 10,000; and 10,000 samples are used for validation. The information has been prepared. A pretrained embedding model and the embedding layer are then used to process the data, and a number of strategies are evaluated to gauge performance.

Data Preprocessing:

- The review processing produces word embedding vectors with fixed dimension while creating the dataset. Each word used in the dataset receives a predefined vector representation. preparation procedure. A restriction exists which amounts to 10,000 samples. Steps have been taken to derive numbers from review content which serve as input data. Each word receives its individual representation instead of preserving its original word sequence. The neural Due to the unsuitable nature of the provided numbers I possess the network refuses to accept them as input.

- The construction of tensors happens through number applications. The integer list might be A tensor of integer data type can be generated through the specified form (samples. word indices). My accomplishment requires the assurance that all reviews have equal lengths. Each sample needs to have equal length so I need to verify the reviews maintain consistent lengths. The reviews need to have equal lengths through the addition of dummy words or numbers.

Procedure: This project studied two distinct methods of embedding word generation for this IMDB database.

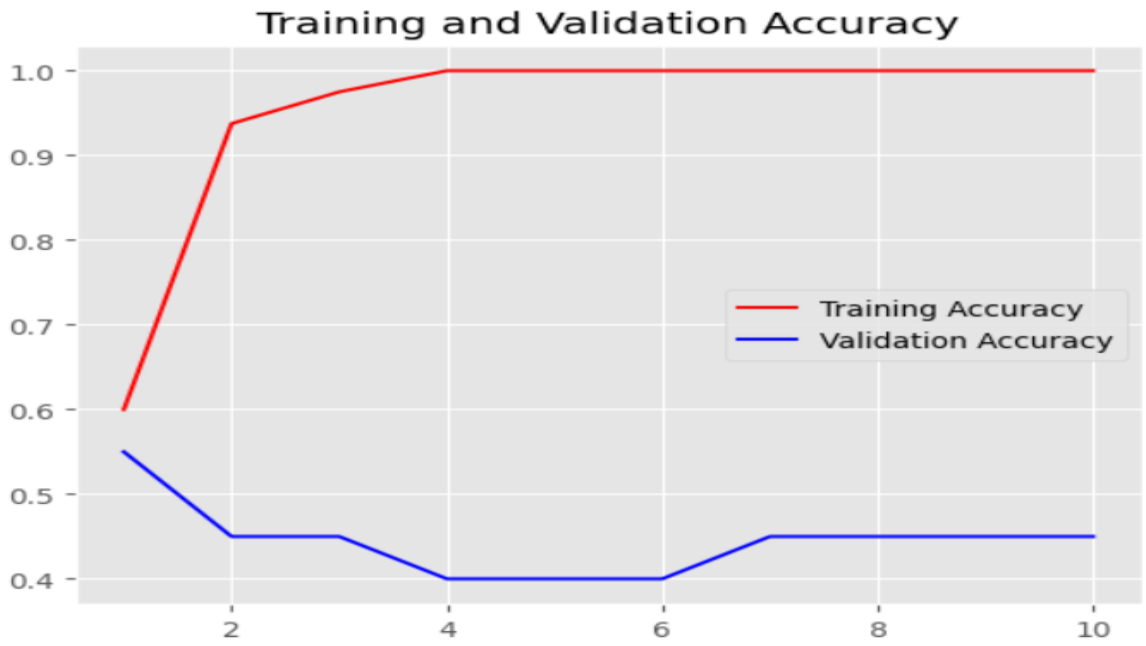
1. Custom-trained embedding layer
2. The embedding layer adopts pre-trained word representation from the GloVe model. Our work used the widely applied pretrained word embedding model

GloVe for our research. The GloVe-trained work embedding layer operates on extensive sets of textual material.

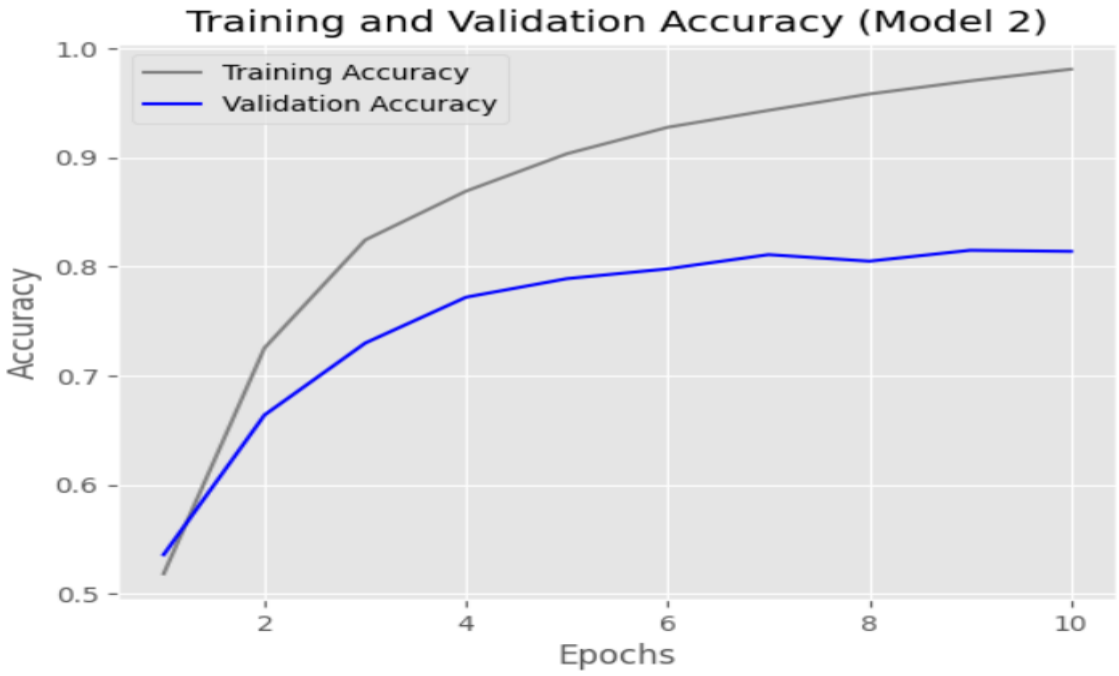
- I analyzed two embedding layers through the IMDB review dataset by deploying both custom-trained and pre-trained embedding methods which represented different embedding strategies. Two embedding layers were tested with a customized layer alongside a word embedding layer derived from pre-trained data to determine framework effectiveness for assessment purposes. The accuracy measurements of these two models received comparison. The research examined training sample sizes that included values of 100, 5000, 1000 and 10,000.
- To start, we used the IMDB review dataset to create a specially-trained embedding layer. We used a testing set to gauge each model's accuracy after it had been trained on a range of dataset samples. Next, we compared these precisions with a model that had a pre-trained word embedding layer and had already been evaluated on various sample sizes.

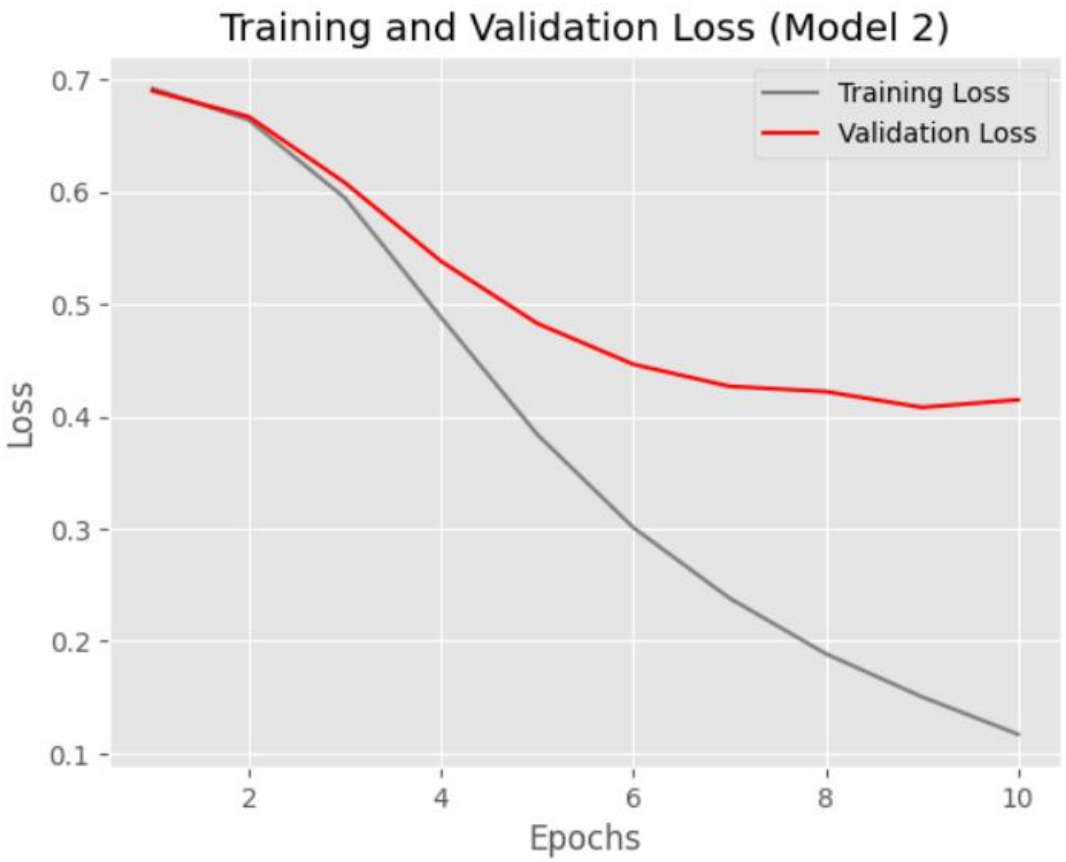
EMBEDDING LAYER WITH CUSTOM TRAINING

1. An embedding layer that has been specially trained using a training sample of 100

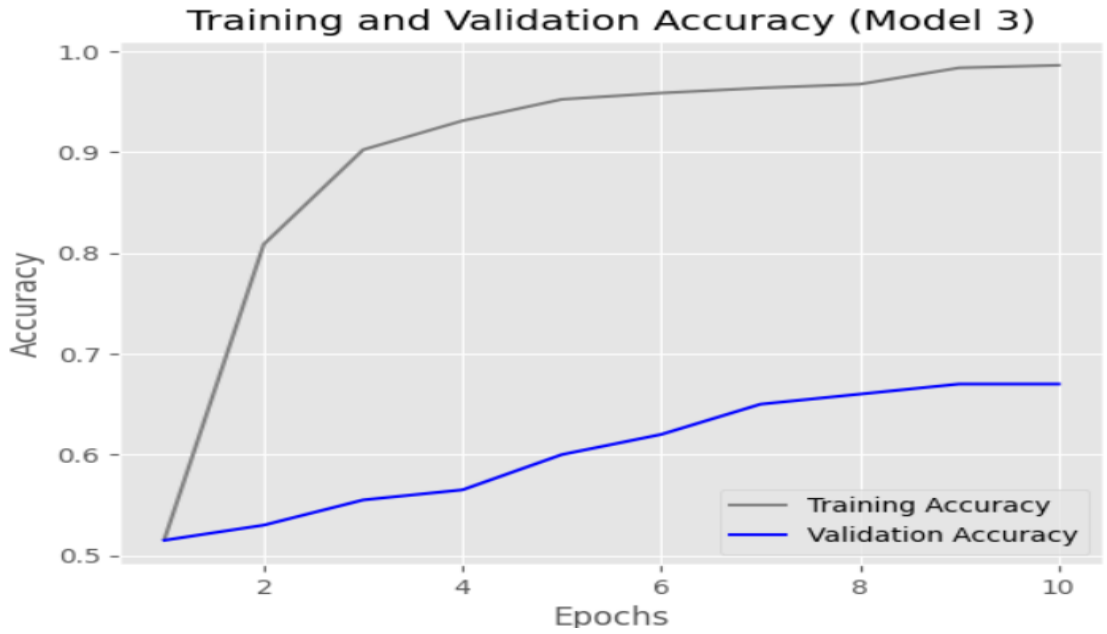


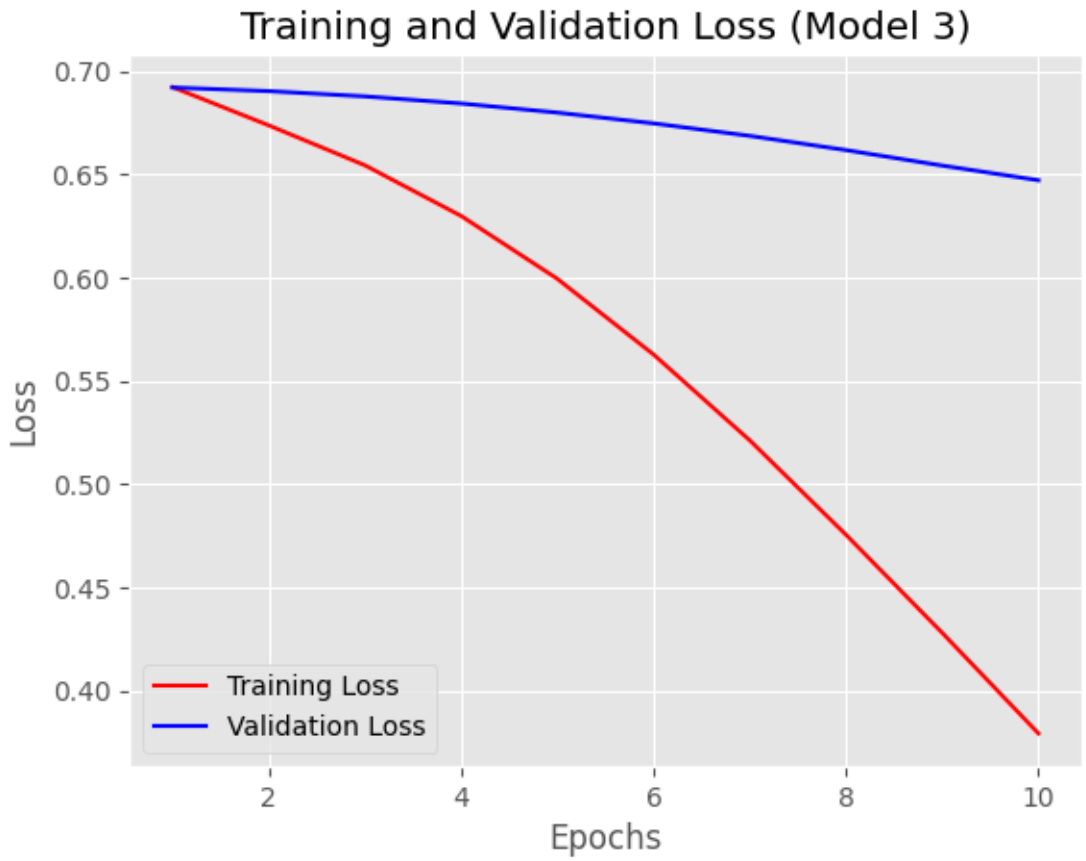
2. An embedding layer that has been specially trained with a training sample size of 5000



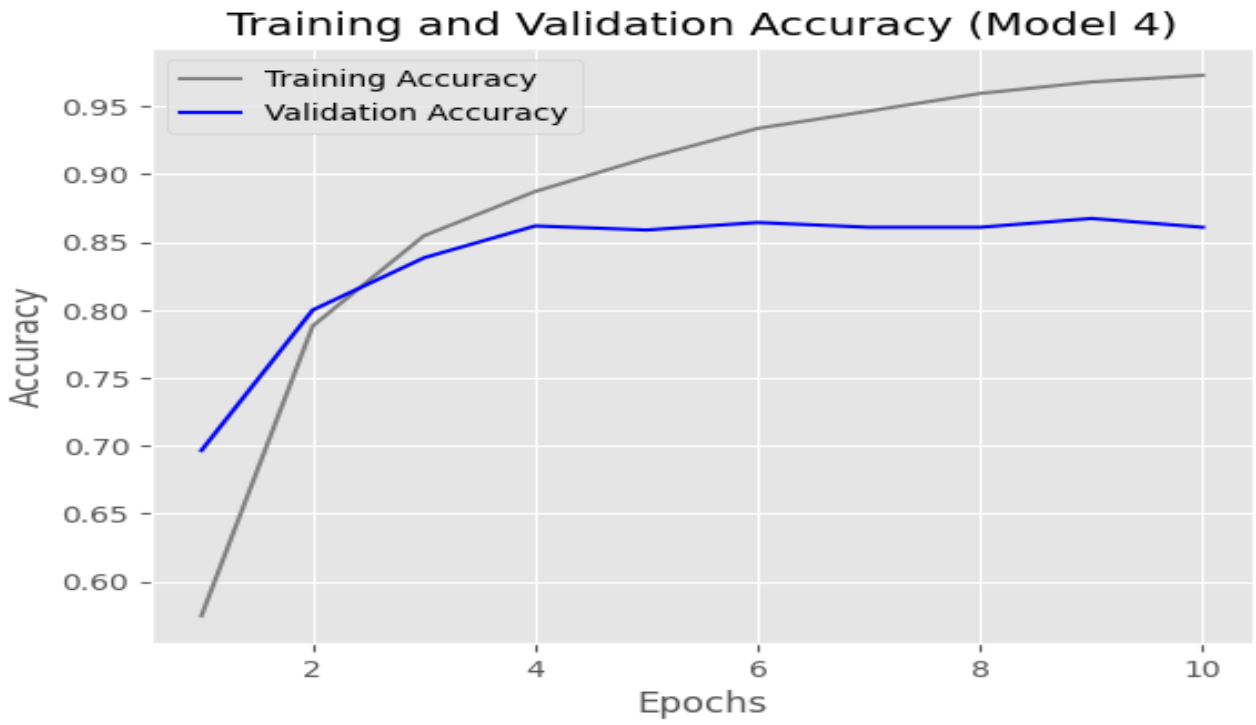


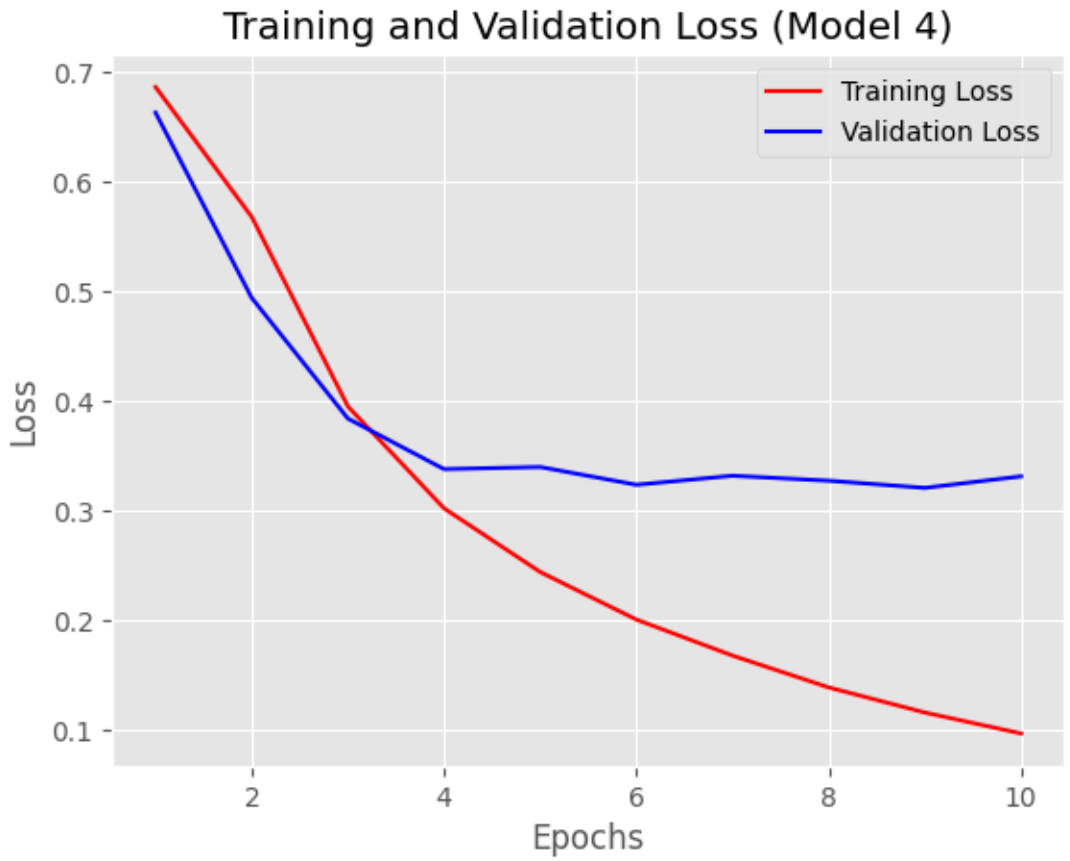
3. An embedding layer that has been specially developed with a training sample size of 1000





4. A specially trained embedding layer with a 10,000 training sample

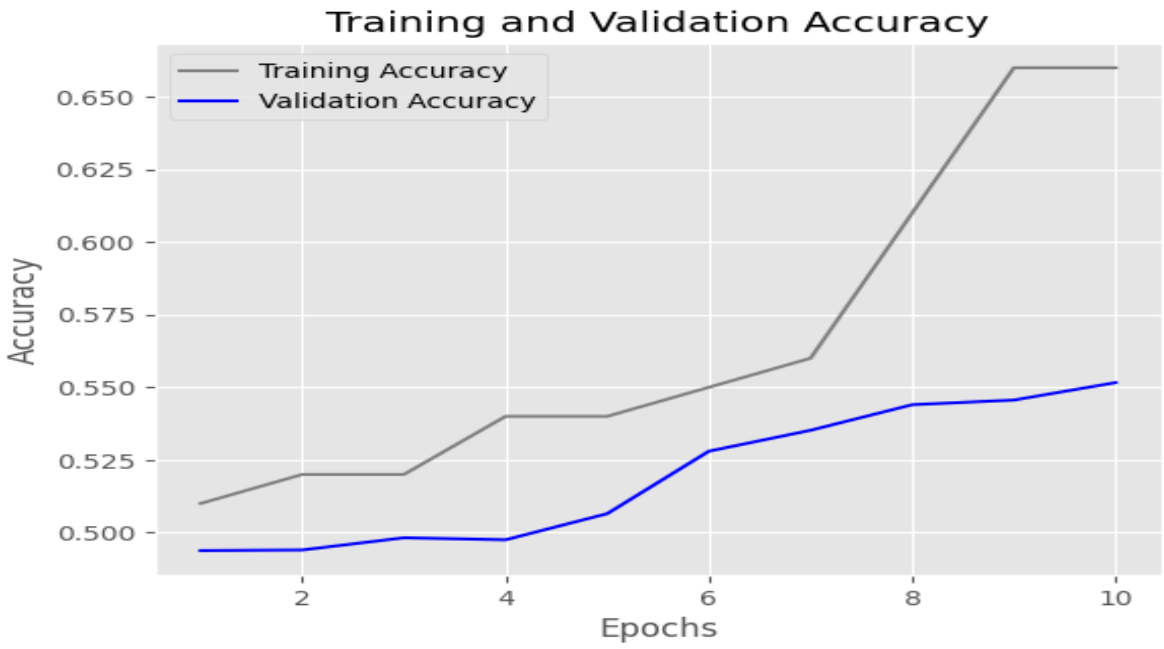


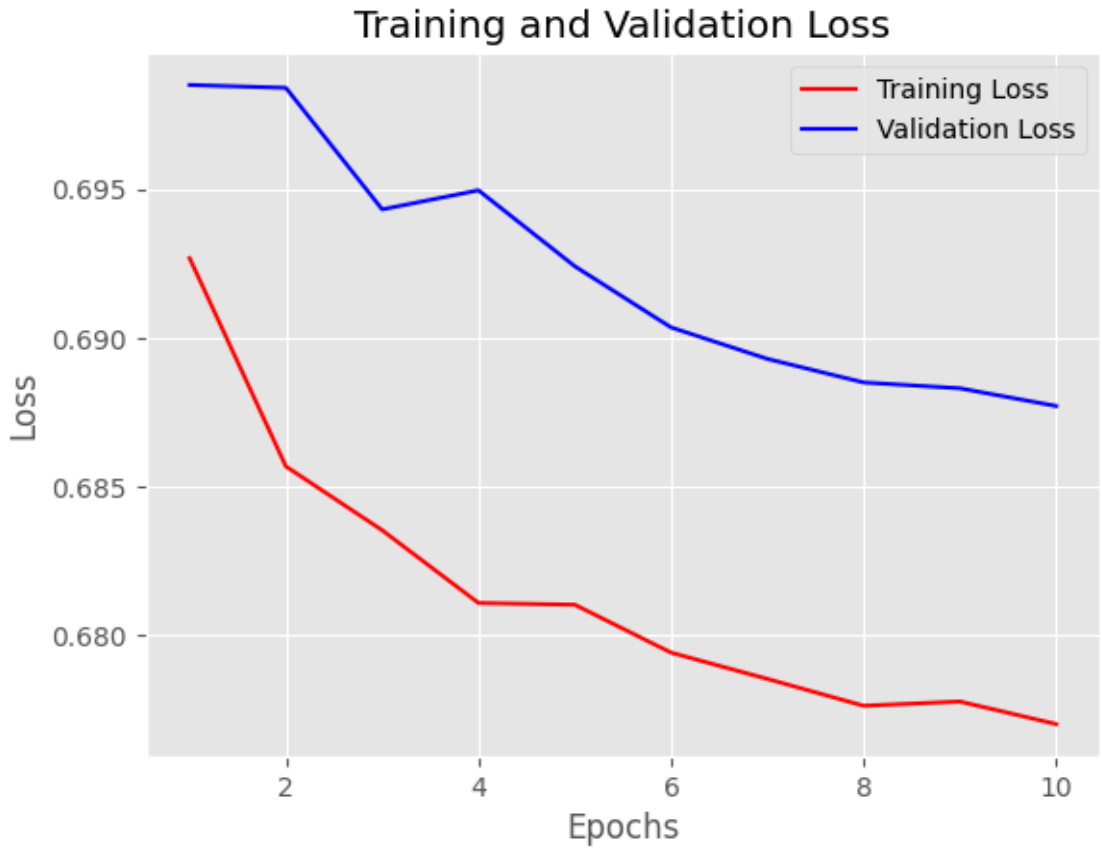


The accuracy of the custom-trained embedding layer varied according on the size of the training sample, ranging from 97.18% to 98.7%. One hundred was the training sample size that produced the highest accuracy.

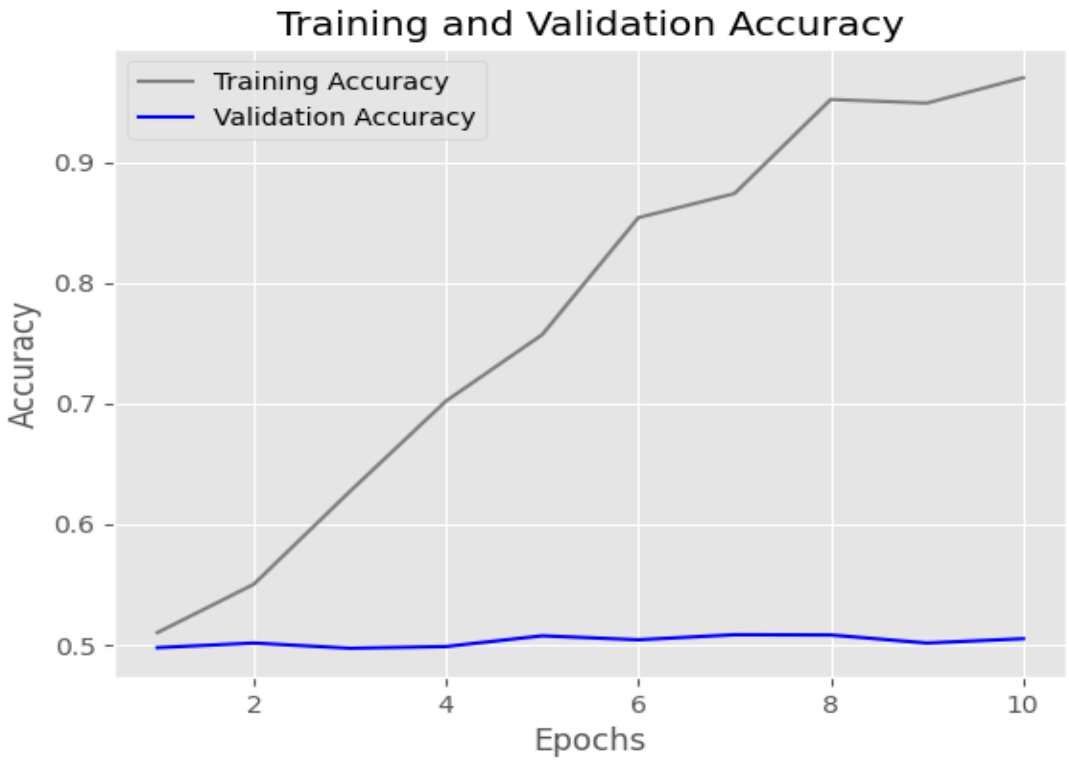
PRETRAINED LAYER FOR WORD EMBEDDING

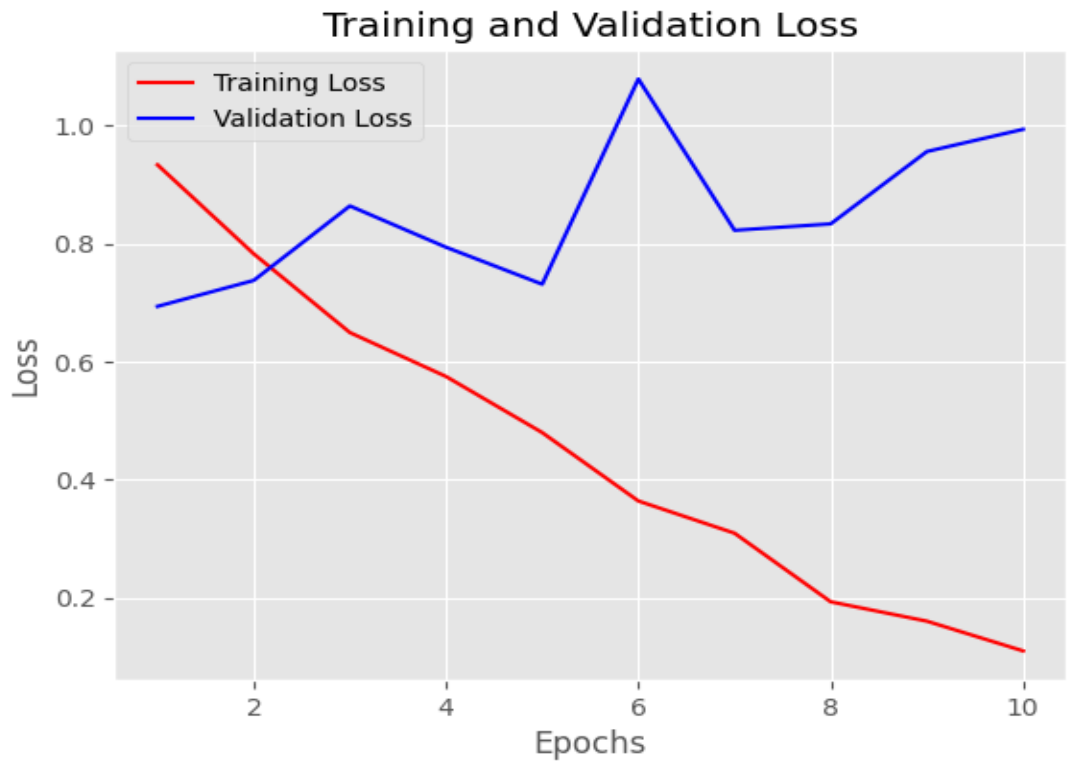
1. pretrained word embedding layer with training sample size = 100



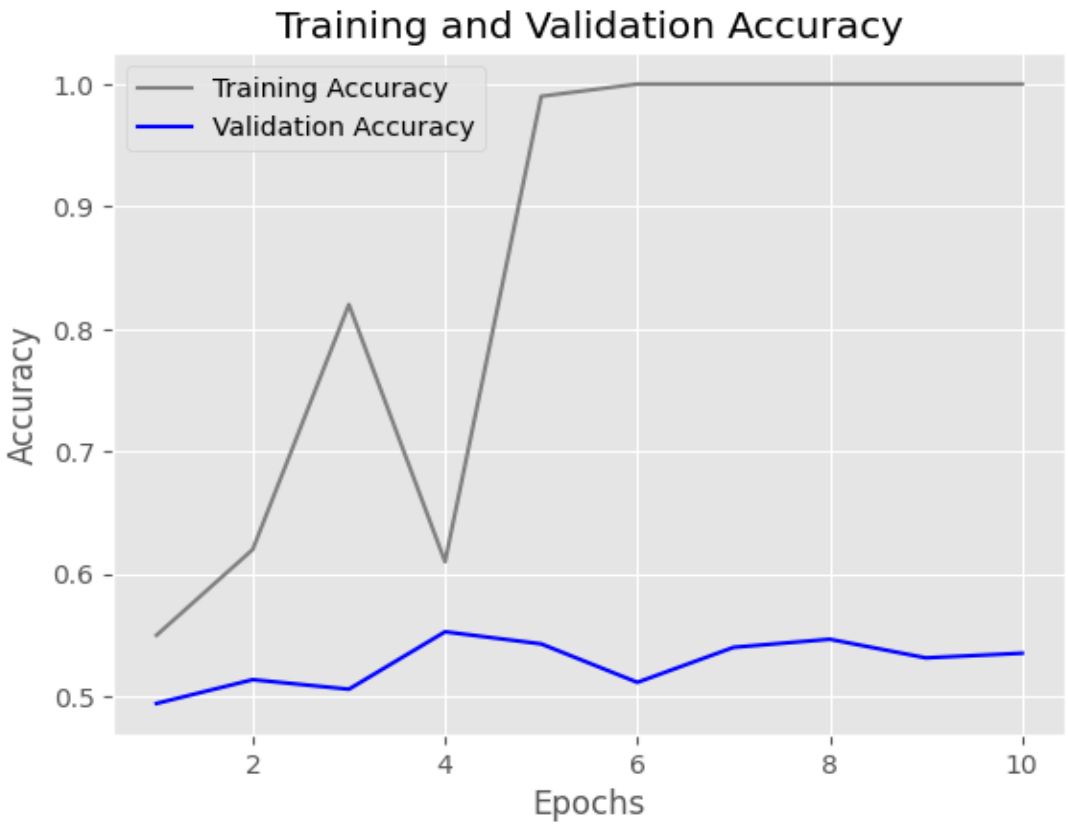


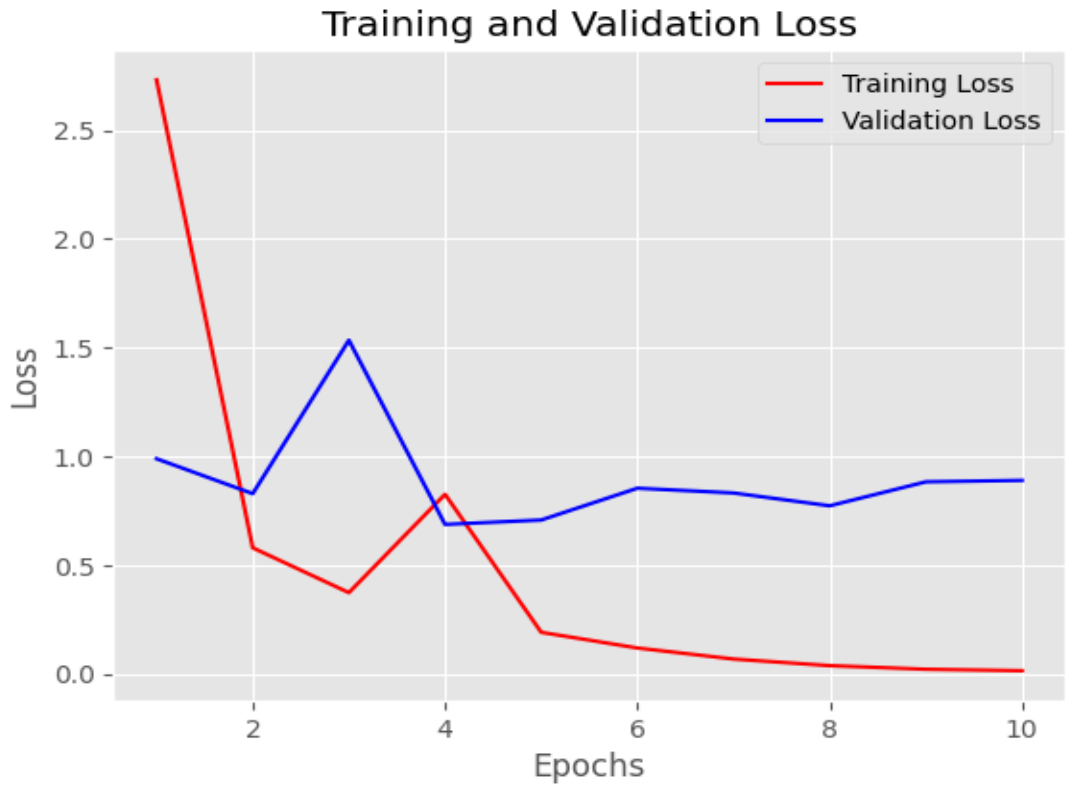
2. Using a training sample size of 5000, the pretrained word embedding layer



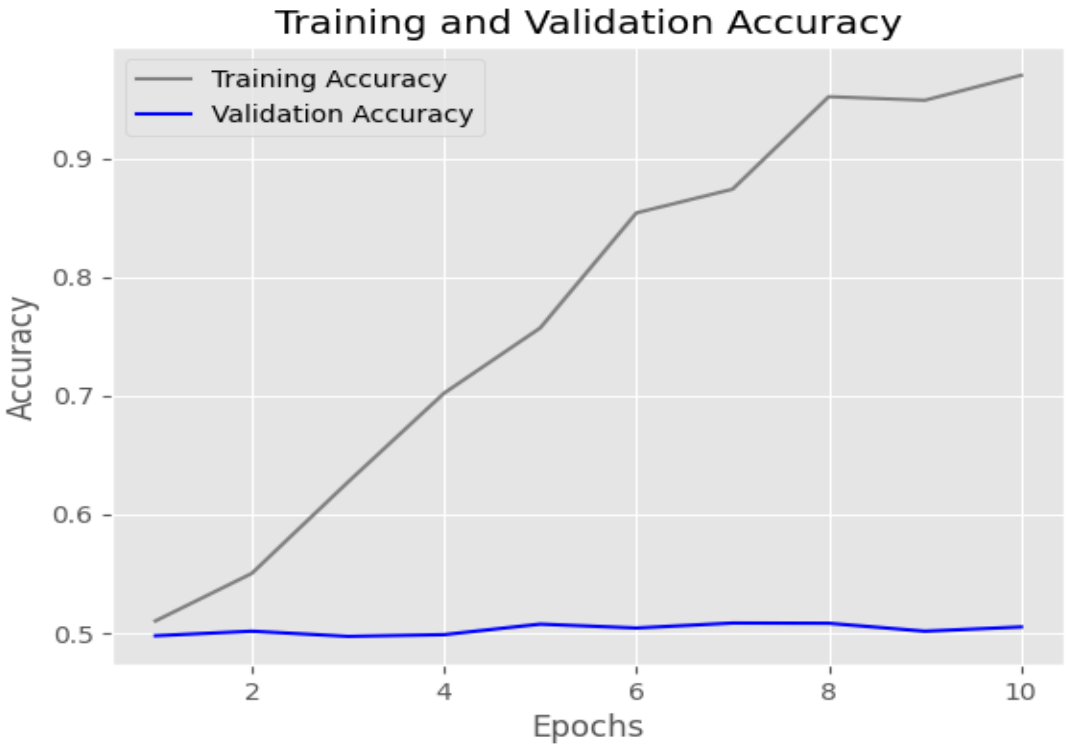


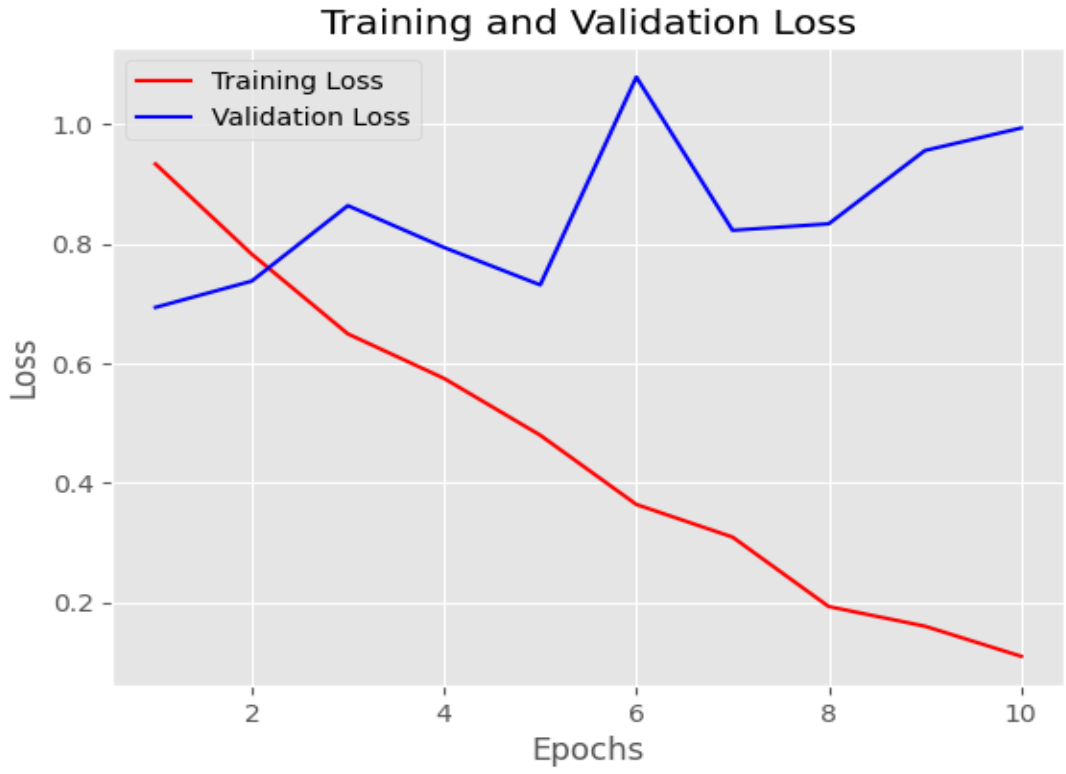
3. Word embedding layer that has been pretrained with a training sample size of 1000





4. A pre-trained word embedding layer with a 10,000 training sample





Across all training sample sizes, the custom-trained embedding layer showed consistently high accuracy, ranging from 97.18% to 98.7%. With 100 training samples,

the best accuracy was recorded at 98.7% with a test loss of 0.694. Better generalization with bigger datasets is suggested by the test loss, which dramatically improves with increasing sample size (down to 0.344 with 10,000 samples).

The pretrained word embedding layer (GloVe), on the other hand, demonstrated accuracy levels ranging from 92.48% to 100%, with the minimum training sample size of 100 yielding the greatest results. But as the size of the training sample grew, test loss increased and accuracy decreased, suggesting possible overfitting. The pretrained embeddings had the maximum test loss of 0.997 at 10,000 samples, indicating that they had trouble generalizing to bigger datasets.

These findings imply that the custom-trained embedding layer provides more consistent and robust performance, particularly when bigger training datasets are available, even if GloVe embeddings can function well with very little data. Therefore, the availability of data and the significance of generalization for the job at hand should be taken into account while choosing an embedding strategy.

Results :

Embedding Technique	Training Sample Size	Training Accuracy (%)	Test loss
Custom-trained embedding layer	100	98.7	0.694
Custom-trained embedding layer	5000	97.5	0.388
Custom-trained embedding layer	1000	97.7	0.655
Custom-trained embedding layer	10000	97.18	0.344
Pretrained word embedding (GloVe)	100	100	0.693
Pretrained word embedding (GloVe)	5000	94.48	0.820
Pretrained word embedding (GloVe)	1000	96.80	0.978
Pretrained word embedding (GloVe)	10000	92.48	0.997

Conclusion :

However, this experiment showed that the custom-trained embedding layer performed better than the pretrained word embedding layer, especially when training with a higher number of training samples. In situations where a short training sample size is required and computer resources are restricted, the pretrained word embedding layer might be a "better choice" despite the risk of overfitting.