TEXT ANALYSIS ON IMDB REVIEWS

Vinay Kumar Kaja (811313881)

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, and only the top 10,000 words are considered. The training samples are restricted to 100, 500, 1000, and 10,000, and 10,000 samples are used for validation. After preparing the data, it is fed into both a pretrained embedding model and a custom embedding layer. Various strategies are then evaluated to measure model performance.

Data Preprocessing:

The dataset preparation process involves transforming each review into word embeddings, where every word is mapped to a fixed-size vector. The dataset is limited to 10,000 samples. Instead of using raw text, each word in a review is converted into a unique integer. While these integer lists can be generated, they are not directly suitable as input for a neural network. Therefore, it's necessary to convert them into tensors with a consistent shape—specifically, arrays of integers in the form (samples, word indices). To achieve this, padding with placeholder values (dummy words or numbers) is applied so that all reviews have the same length.

Procedure:

In this study, I looked into two distinct methods for creating word embeddings for this IMDB dataset:
1. An embedding layer with custom training
2. GloVe model-based pre-trained word embedding layer.

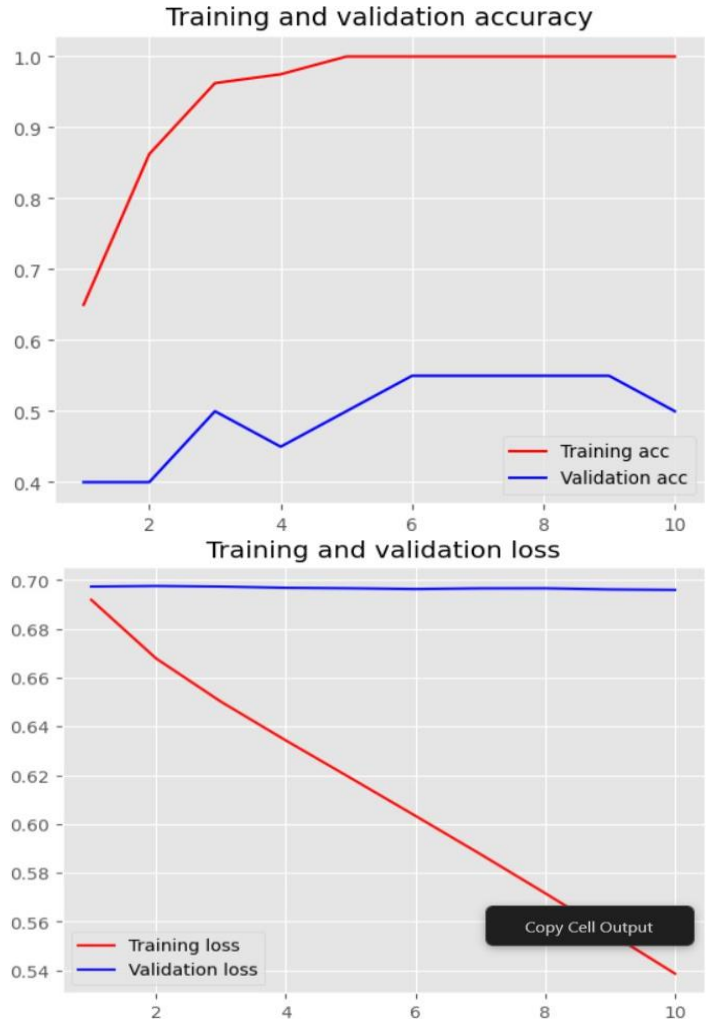
Large volumes of textual data are used to train the popular pretrained word embedding model GloVe, which we used in our work.

• To evaluate the effectiveness of various embedding techniques, I employed two distinct embedding layers—one with a pre-trained word embedding layer and the other with a custom-trained layer—using the IMDB review dataset. The accuracy of the two models was evaluated using training sample sizes of 100, 5000, 1000, and 10,000.

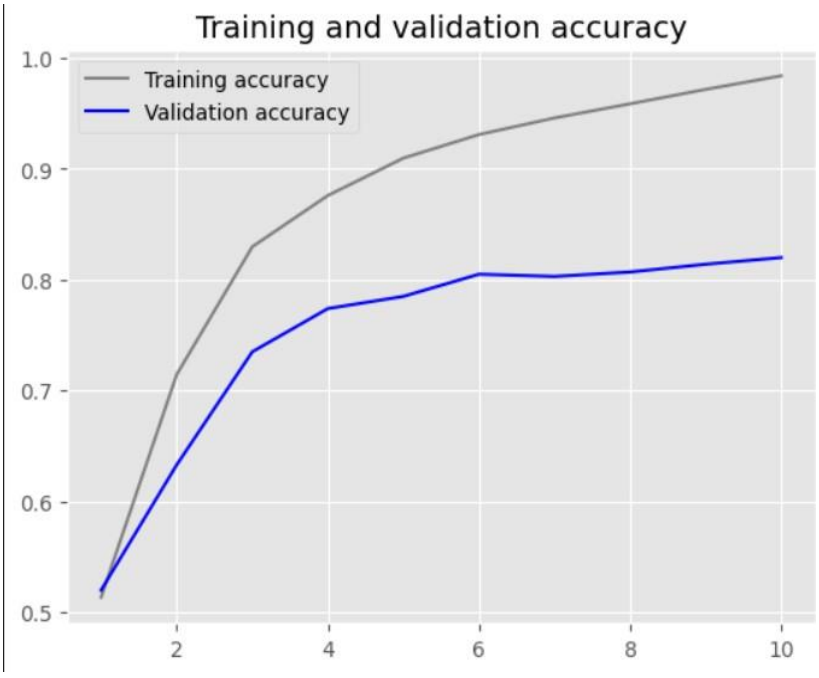
Initially, we used the IMDB review dataset to build a custom-trained embedding layer. After training the model on different sample sizes, we evaluated its accuracy using a test set. We then compared these results with those from a model that used a pretrained word embedding layer, which had also been tested across the same sample sizes.

CUSTOM-TRAINED EMBEDDING LAYER

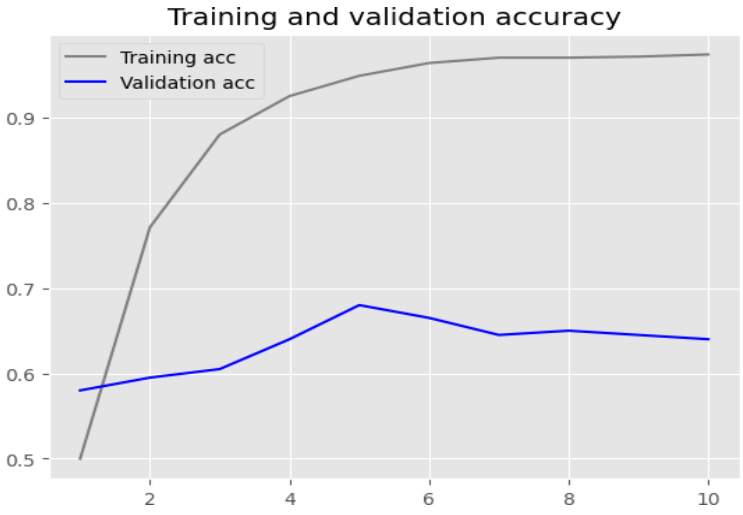
1. Custom-trained embedding layer with training sample size = 100

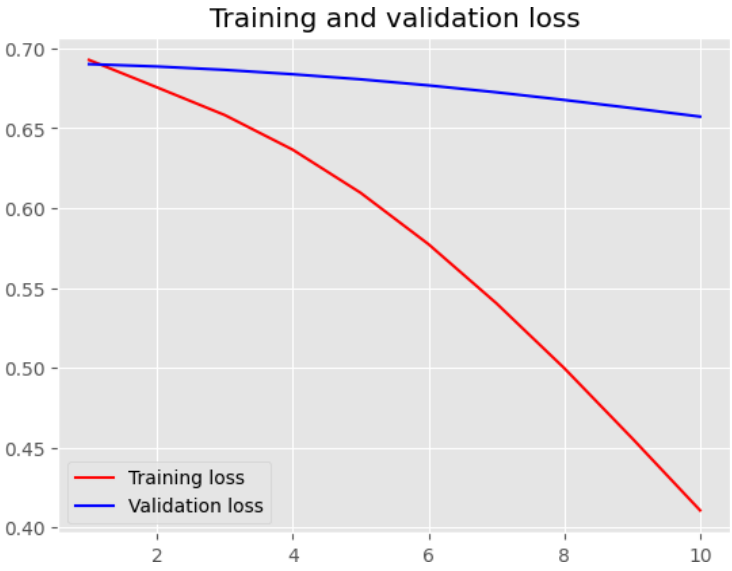


2. Custom-trained embedding layer with training sample size = 5000

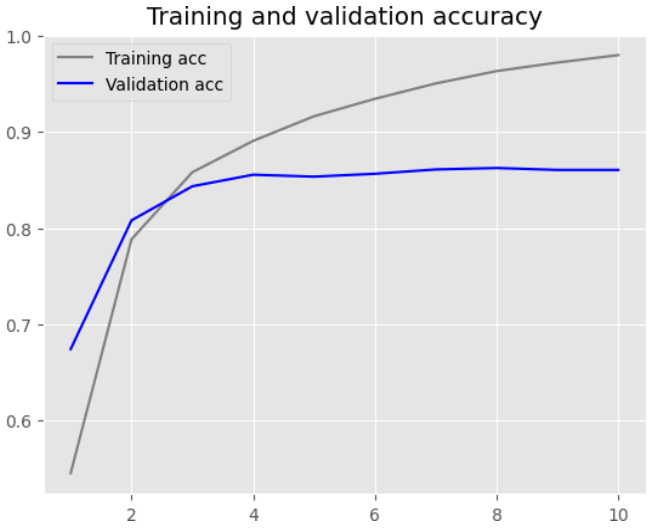


3. Custom-trained embedding layer with training sample size = 1000





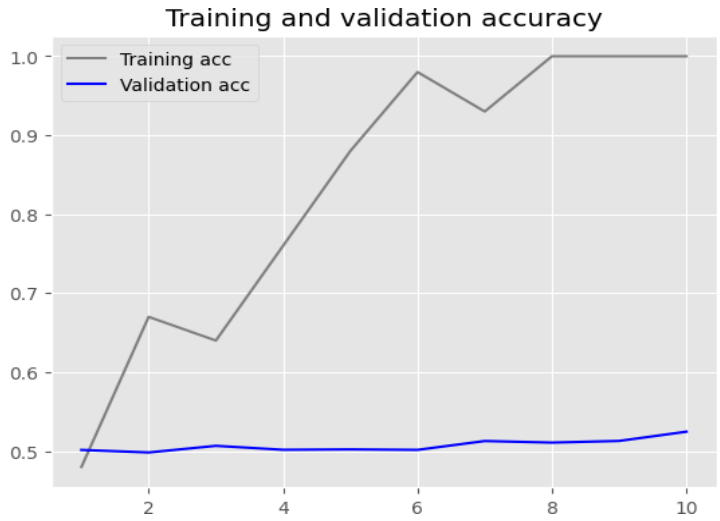
4. Custom-trained embedding layer with training sample size = 10000

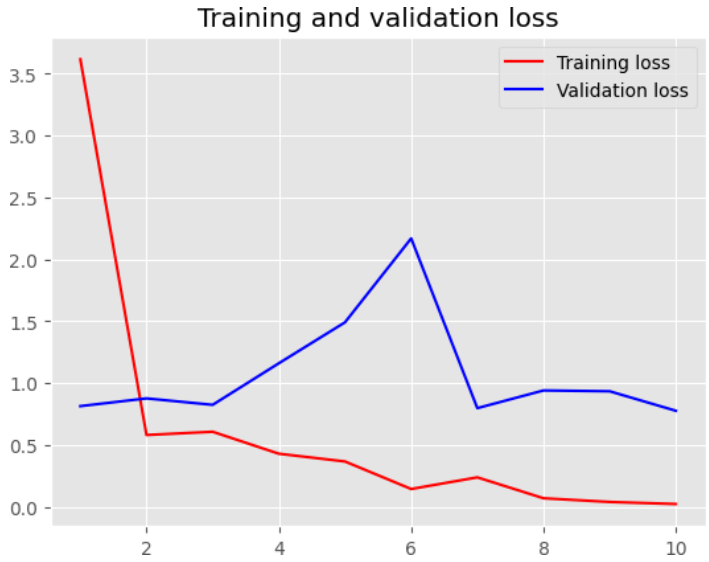


The custom-trained embedding layer achieved accuracy between 97.3% and 100%, varying with the size of the training set. The highest accuracy was recorded when using 100 training samples.

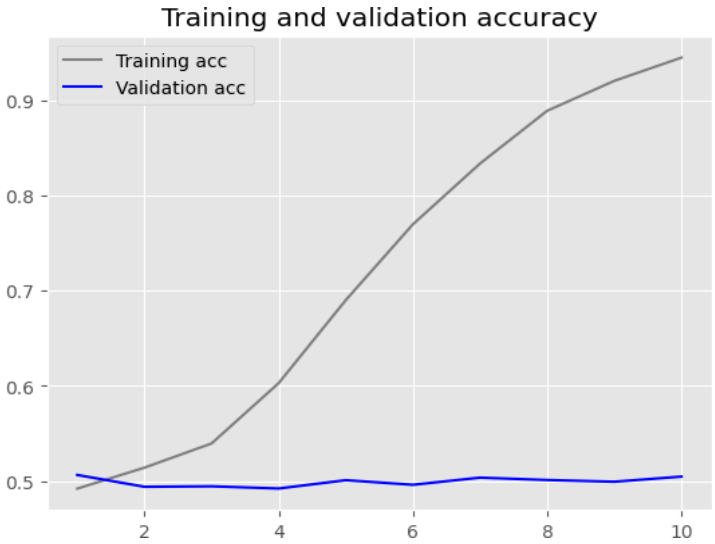
PRETRAINED WORD EMBEDDING LAYER

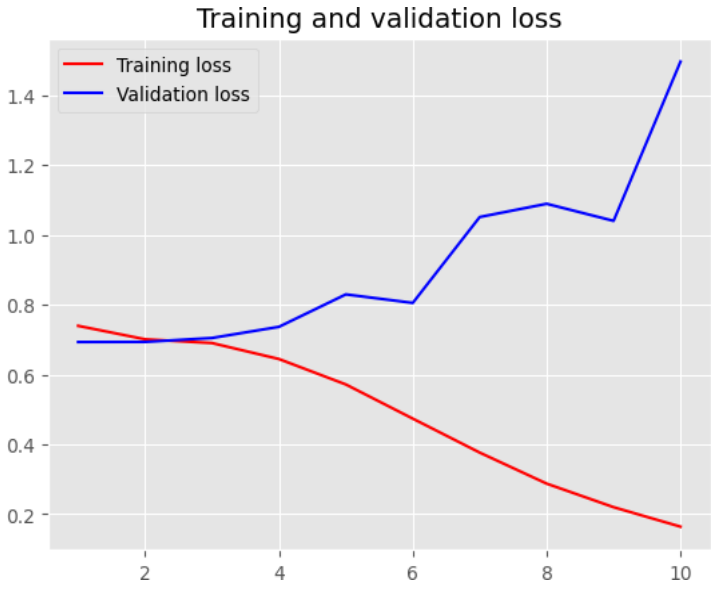
1. pretrained word embedding layer with training sample size = 100



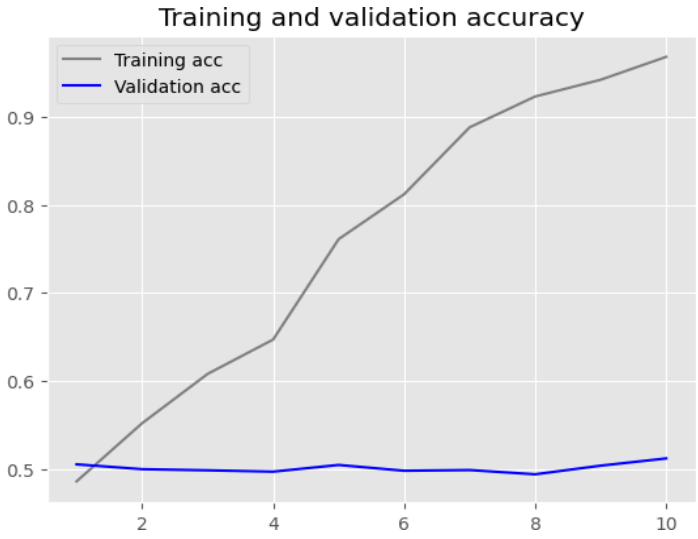


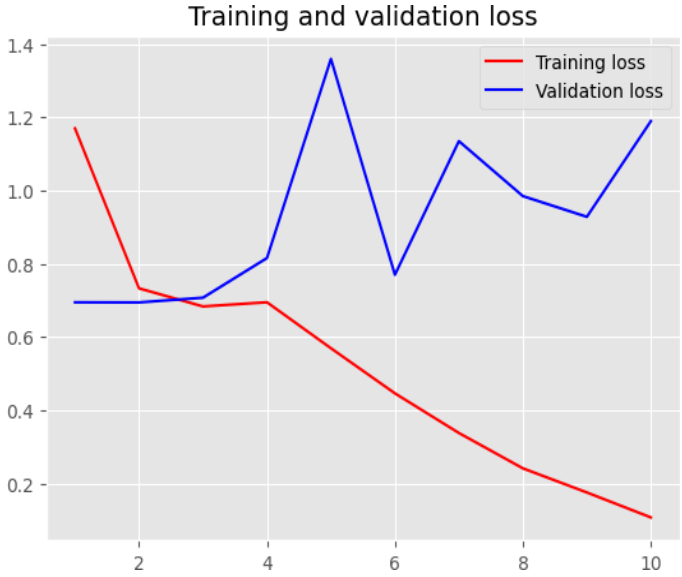
2. pretrained word embedding layer with training sample size = 500



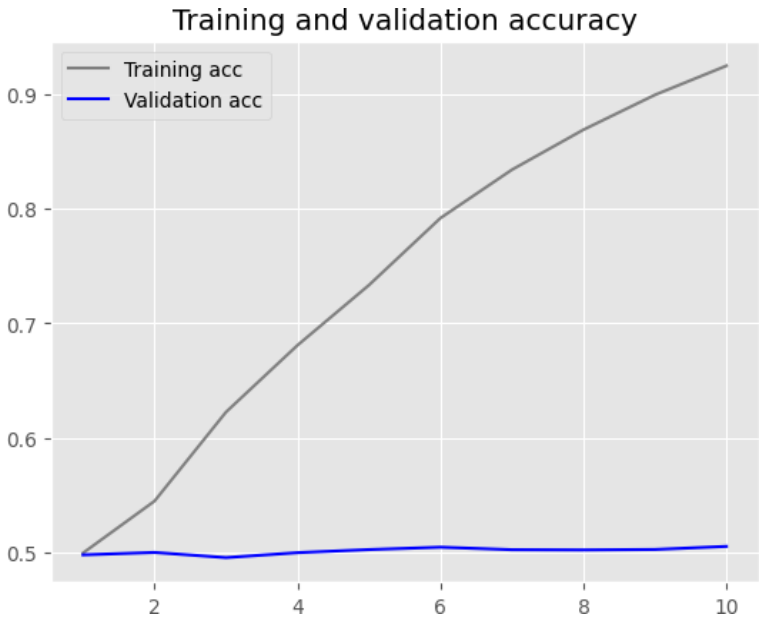


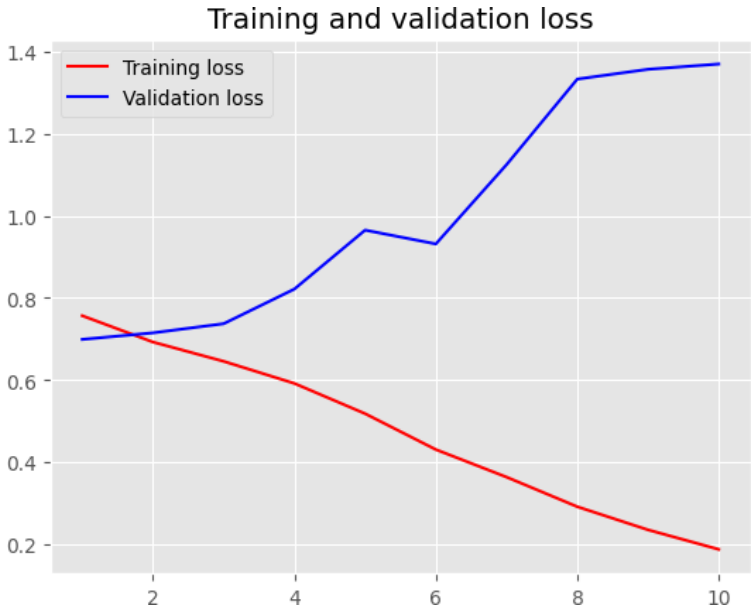
3. pretrained word embedding layer with training sample size = 1000





4. pretrained word embedding layer with training sample size = 10000





The pretrained word embedding layer (GloVe) showed accuracy between 92% and 100%, depending on the size of the training data. The highest accuracy was observed with just 100 training samples. However, increasing the training size led to overfitting, which reduced the model's performance. Choosing the most effective approach can be difficult, as it often depends on the specific requirements and limitations of the task.

RESULTS:

Embedding Technique	Training Sample Size	Training Accuracy (%)	Test loss
Custom-trained embedding layer	100	100	0.69
Custom-trained embedding layer	5000	98.40	0.37
Custom-trained embedding layer	1000	97.3	0.68
Custom-trained embedding layer	10000	98	0.34
Pretrained word embedding (GloVe)	100	100	0.79
Pretrained word embedding (GloVe)	5000	94.48	1.43
Pretrained word embedding (GloVe)	1000	96.80	1.10
Pretrained word embedding (GloVe)	10000	92.48	1.41

CONCLUSION:

In this experiment, the custom-trained embedding layer generally outperformed the pretrained word embedding layer, particularly with larger training sample sizes. However, if computational resources are limited and only a small training set is available, the pretrained embedding layer could be a more practical option—even though it may lead to overfitting.