TEXT AND SEQUENCE

ASSIGNMENT-4

Summary:

In this study, we looked at the viability of using binary classification to forecast sentiment (positive or negative) expressed in movie reviews. We trained many models on a variety of sample sizes (100, 500, 1,000, and 10,000 reviews) from the 50,000 reviews in the IMDB review corpus. The 10,000 most common phrases were used for training and a different validation set of 10,000 reviews to guarantee uniformity in model assessment. The data underwent pre-processing prior to being fed into a pre-trained embedding layer, followed by an optimization process to achieve optimal model performance.

Technique:

The sentiment analysis in this study utilizes the IMDB movie review dataset, where each review expresses an overall positive or negative opinion about a film. To prepare the data for our neural network, a crucial preprocessing step is employed. Here, each review undergoes a two-fold transformation. First, individual words are converted into numerical representations known as word embeddings. Each word is given a fixed size vector in these embeddings, which represent its meaning in relation to other words. To establish a consistent vocabulary, it's crucial to remember that just the top 10,000 most often occurring terms are considered.

Secondly, the reviews are transformed from their original text format into a sequence of integers. While this simplifies processing for the neural network, a challenge arises because reviews can vary in length. To address this inconsistency, shorter reviews are padded with additional dummy integer values, ensuring all samples have a uniform length. This preprocessing step guarantees the model receives consistent data for optimal performance.

Approach:

To represent words in our sentiment analysis task, we explored two embedding techniques: a pre-trained GloVe layer and a custom-trained layer. The popular GloVe model, trained on vast amounts of text, excels at capturing word relationships for NLP tasks.

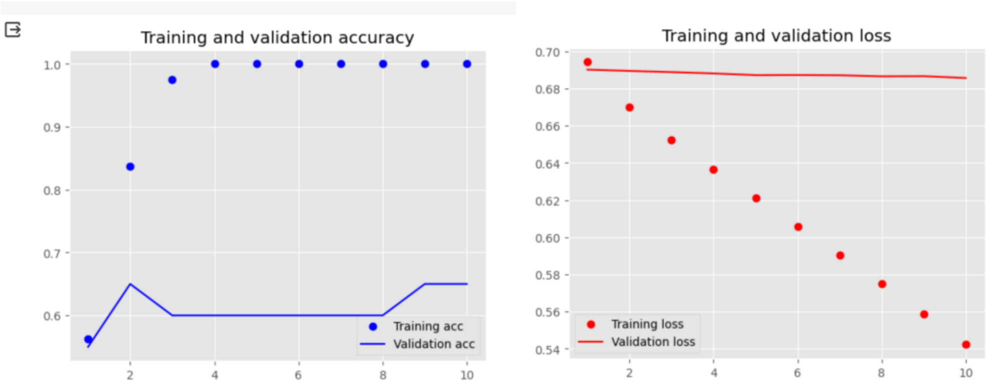
For our analysis, we built two separate embedding layers: a custom-trained one specific to the IMDB reviews, and another leveraging a pre-trained GloVe model. This allows us to compare the effectiveness of different embedding approaches.

We examined how the size of training data affected the performance of the model. We trained two models using different sample sizes (100, 500, 1000, and 10,000 reviews) from the IMDB dataset: one with a custom embedding layer and the other with a pre-trained GloVe layer.

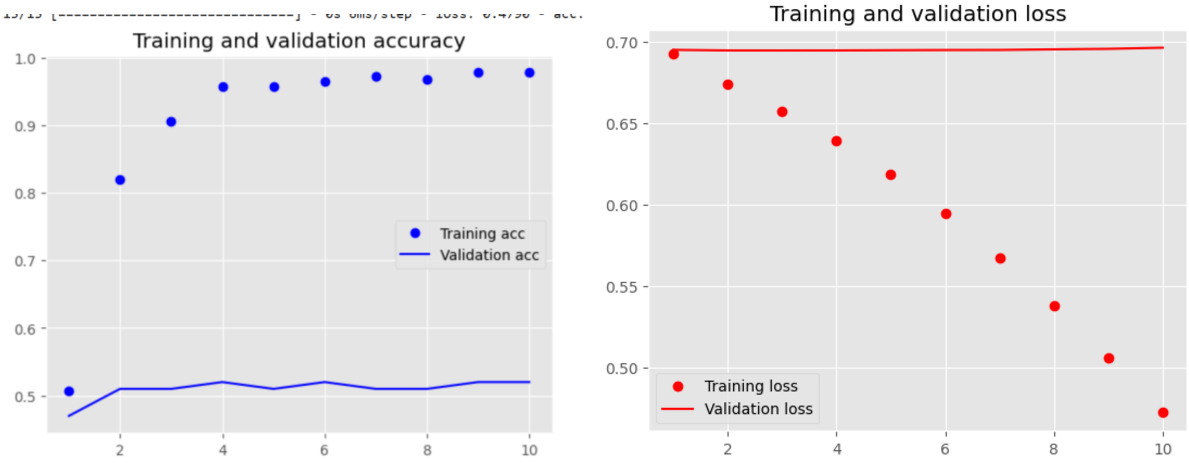
We then evaluated their accuracy on a separate test set to compare the effectiveness of each embedding approach across different data quantities.

Custom-trained embedding layer

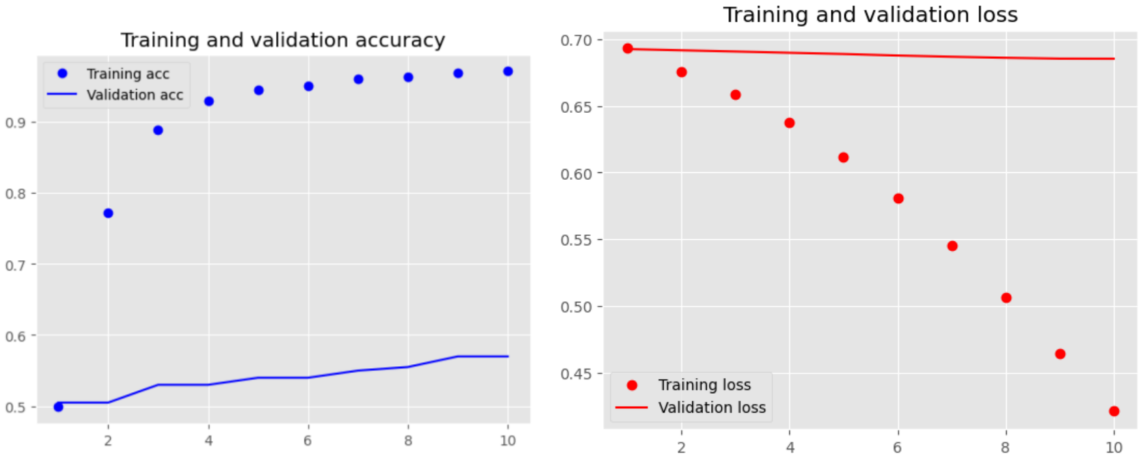
Customer Custom-trained embedding layer with sample size 100



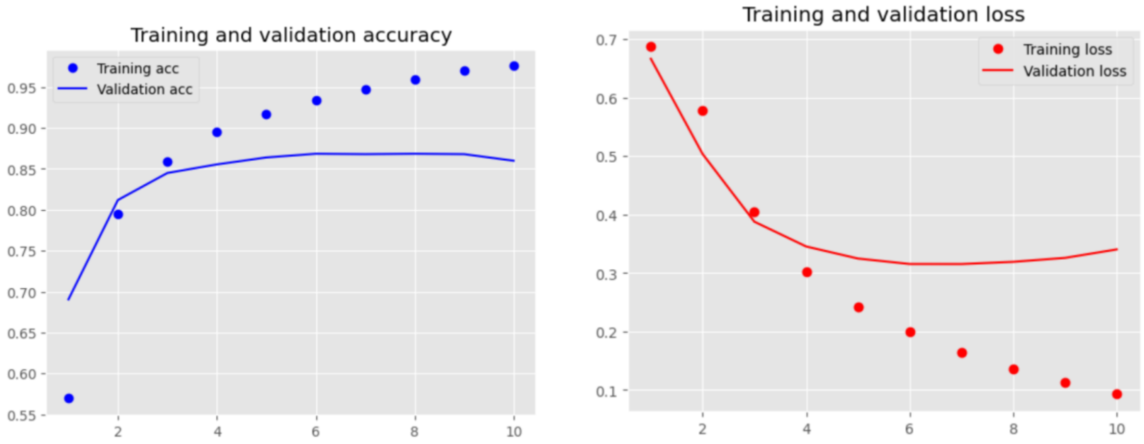
Customer Custom-trained embedding layer with sample size 500


```
13/13 [==============================] - 0s 6ms/step - loss: 0.4796 - acc:
```

Customer Custom-trained embedding layer with sample size 1000

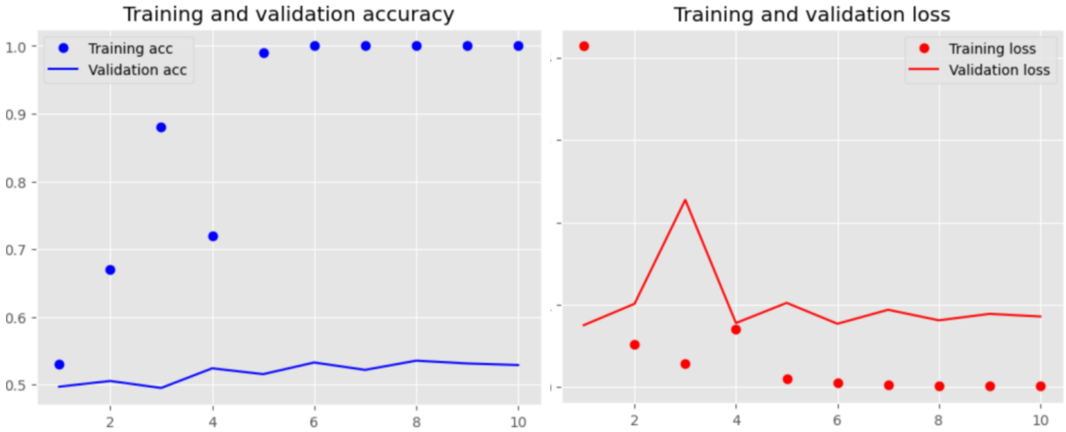


Customer Custom-trained embedding layer with sample size 10000

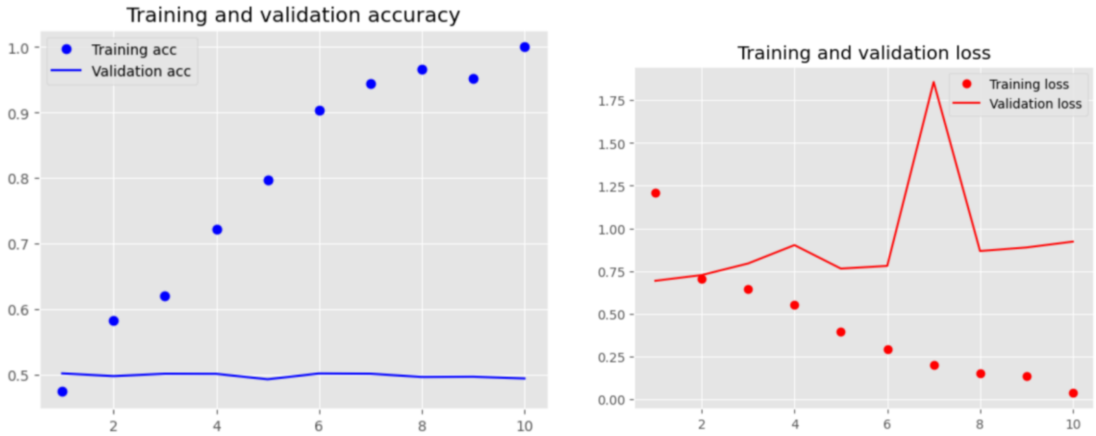


Pretrained word embedding layer (GloVe):

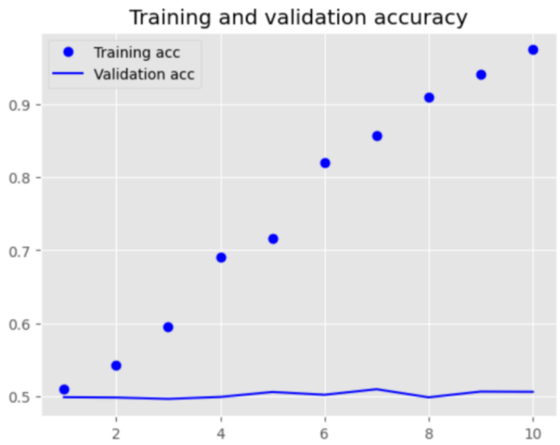
Pretrained word embedding layer (GloVe) with sample size 100.

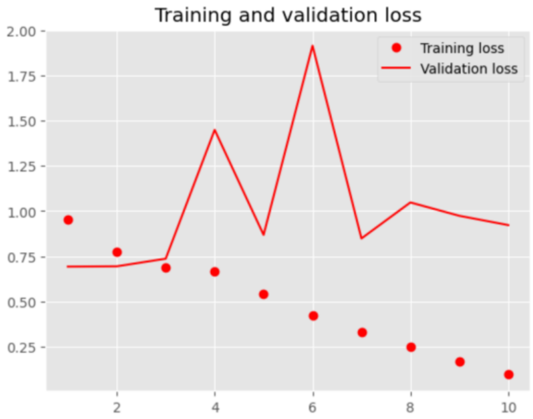


Pretrained word embedding layer (GloVe) with sample size 500.

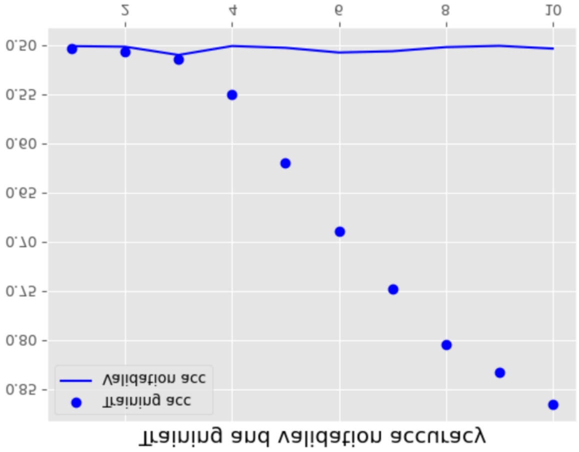


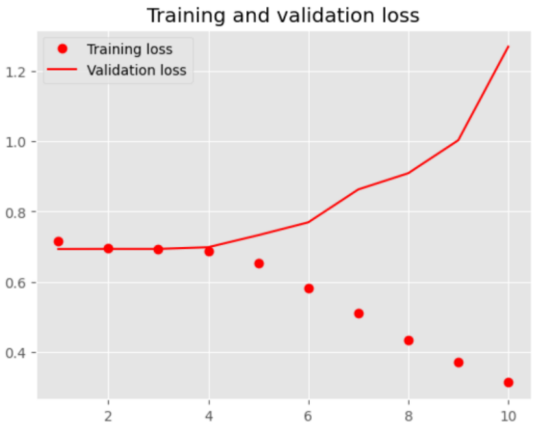
Pretrained word embedding layer (GloVe) with sample size 1000





Pretrained word embedding layer (GloVe) with sample size 10000





Results:

Embedding Technique	Maxlen	Training sample size	Loss and Accuracy on Test	Accuracy(%)
Custom-trained embedding layer	150	100	Loss:0.693- Acc:0.500	100
Custom-trained embedding layer	150	500	Loss:0.688- Acc:0.538	98.5
Custom-trained embedding layer	150	1000	Loss:0.690- Acc:0.537	97.6
Custom-trained embedding layer	150	10000	Loss:0.3434- Acc:0.8524	97.8

Pretrained word embedding layer (GloVe)	150	100	Loss:0.88- Acc:0.49	100
Pretrained word embedding layer (GloVe)	150	500	Loss:0.91- Acc:0.50	96.6
Pretrained word embedding layer (GloVe)	150	1000	Loss:0.933- Acc:0.49	93.5
Pretrained word embedding layer (GloVe)	150	10000	Loss:1.287- Acc:0.50	90

Custom-trained embedding layer:

The custom-trained embedding layer achieved impressive accuracy, ranging from 98.5% to 100%, with the highest performance (100%) observed for the smallest training set (100 reviews). This suggests the custom embeddings might be particularly well-suited to capturing the nuances of sentiment in IMDB reviews, potentially due to their specific training on this domain.

Pretrained word embedding layer (GloVe):

The accuracy of the pre-trained GloVe model changed depending on how much training data it had (100 to 10,000 samples), ranging from 90% to perfect accuracy (100%). The pre-trained model did best (100% accuracy) when trained on the smallest amount of data (100 reviews). Because pre-trained embeddings like GloVe already capture a lot of meaning from a vast amount of text, they work well even with limited training data. This explains why the pre-trained model did well with just 100 reviews. However, as you give the pre-trained model more data (increasing training sample size), it might struggle to grasp the specific details of this task (sentiment analysis of IMDB reviews). This could lead to lower accuracy.

Using pre-trained embeddings with a large amount of training data can lead to overfitting and decreased accuracy in the model, as demonstrated earlier. An overfitting model is unable to process new data because it becomes too good at recalling the training set. It is difficult to say for sure whether approach custom-trained or pre-trained is always better because it depends on the objectives and limitations of a given project. Overall, in this experiment, the **custom-trained embeddings outperformed the pre-trained ones**, especially after receiving more training data. If you have limited computing power and minimal training data, the pre-trained model could be a better option even if it has an overfitting risk.

Conclusion:

The impact of pre-trained embeddings on sentiment analysis performance might be influenced by the size of the training data. While pre-trained models like GloVe excel at capturing general semantic relationships, they may struggle with the specific nuances of a given task (e.g., IMDB sentiment analysis) as the training data volume increases. This can lead to two potential issues:

Inaccuracy: Inaccurate results might be the consequence of the pre-trained embeddings' inability to accurately collect task specific features.
Overfitting: datasets when combined with pre-trained embeddings might cause the model to become overfit to the training set, hence decreasing its accuracy and restricting its ability to generalize to new data.

Therefore, the requirements and constraints of the project will determine which embedding strategy is best.

Exploring Embedding Options for Smaller Datasets:

For tasks with limited training data, employing a custom-trained embedding layer can be more effective. This allows the model to focus on the unique characteristics of the smaller dataset, potentially leading to improved accuracy compared to pre-trained models.

key points:

Pre-trained embeddings might be less effective with larger training datasets due to limitations in capturing task-specific nuances.

This can lead to inaccuracy and overfitting, reducing model performance.

Custom-trained embeddings might be a better option for tasks with limited data as they can focus on the specific data characteristics.

The best embedding approach depends on the project's needs and data size.

Recommendations:

- When employing pre-trained networks and suitable embedding techniques, you may get good results even with a small quantity of training data.
- The model's ability to generalize is improved by using pre-trained networks and embeddings.
- Use data augmentation techniques, which involve making modifications to existing data to create new samples for training. This improves the model's capacity for generalization, particularly when dealing with limited data, and diversifies the training set.