**AML ASSIGMENT 4**

**SENTIMENT ANALYSIS ON IMDB REVIEWS**

**SAI PRASAD BOYAPATI**

**811294674**

**Objective:**

The goal of the IMDB dataset's binary classification problem is to classify movie reviews as either positive or negative. The dataset contains 50,000 reviews, with only the top 10,000 most frequent words being considered. The training samples are limited to sizes of 100, 1,000, 5,000, and 10,000, while validation is performed on 10,000 samples. After preparing the data, it is passed through an embedding layer with a pretrained embedding model, and various strategies are tested to evaluate performance.

**Data Preprocessing:**

As part of the dataset preparation, each review is converted into a set of word embeddings, with each word represented by a fixed-size vector. There is a limit of 10,000 samples. The reviews are transformed into sequences of numbers, where each number represents a unique word, but this numerical format isn’t directly compatible with the neural network input. To address this, tensors must be created from these integers, ensuring they have the appropriate data type and shape (samples, word indices). Additionally, to ensure consistency, all reviews must be padded to the same length by adding dummy words or numbers, so they can be processed properly.

**Procedure:**

In this study, I explored two distinct methods for generating word embeddings for the IMDB dataset:

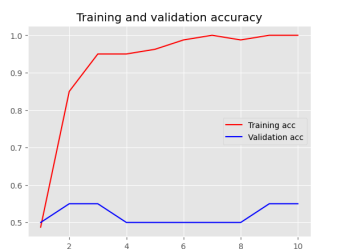
1. Custom-trained embedding layer

2. Pre-trained word embedding layer using the GloVe model

* The widely-used GloVe model, which we incorporated into our work, is a pretrained word embedding model trained on extensive textual data.
* In this study, I employed two different embedding layers using the IMDB review dataset: one with a custom-trained embedding layer and the other with a pre-trained GloVe word embedding layer. This approach allowed me to evaluate the effectiveness of each embedding strategy. I compared the accuracy of both models across different training sample sizes: 100, 1,000, 5,000, and 10,000.
* The process began by creating a custom-trained embedding layer using the IMDB dataset. After training each model on various sample sizes, I assessed their accuracy using a testing set. Subsequently, I compared the performance of these models with another model, which was also tested on different sample sizes and utilized a pre-trained word embedding layer.

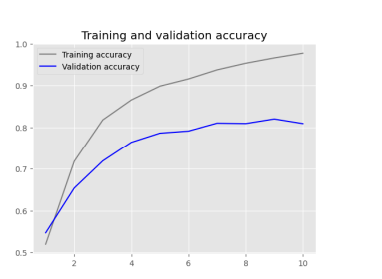
**CUSTOM-TRAINED EMBEDDING LAYER**

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

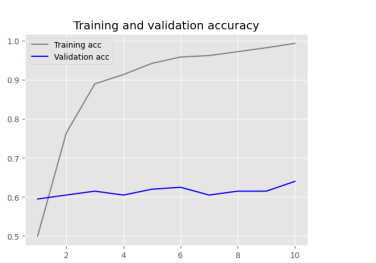




1. Custom-trained embedding layer with training sample size = 5000

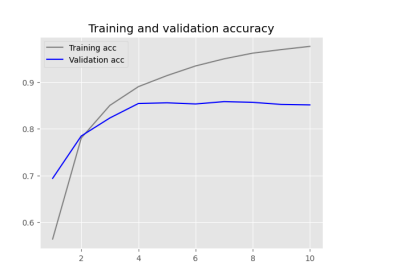
 

1. Custom-trained embedding layer with training sample size = 1000





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

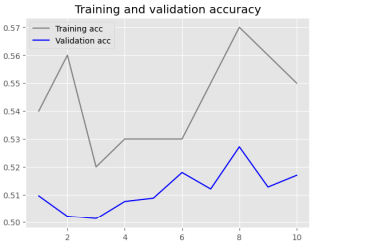
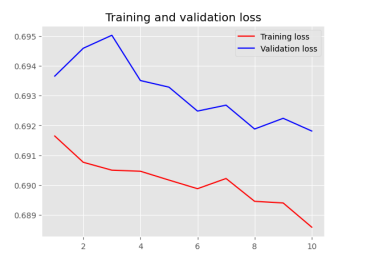




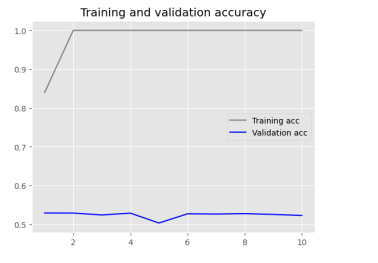
With the custom-trained embedding layer, the accuracy ranged from 97.3% to 98.48%, depending on the size of the training sample. The best accuracy was obtained with a training sample size of 100.

**PRETRAINED WORD EMBEDDING LAYER**

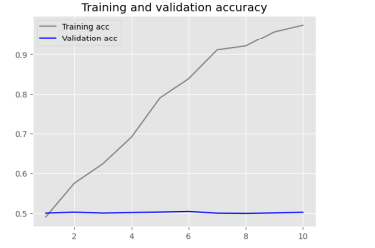
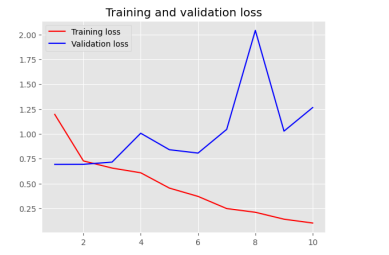
1. pretrained word embedding layer with training sample size = 100

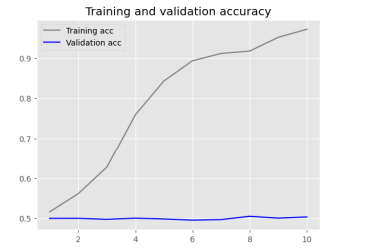
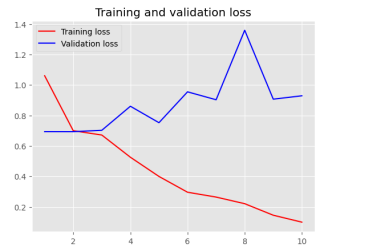
1. pretrained word embedding layer with training sample size = 5000

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) exhibited varying degrees of accuracy, from 50.5% to 92.48%, depending on the size of the training sample. Most accurate result was obtained with 100 training samples. In addition, the model quickly overfits when using the pretrained embeddings with bigger training sample sizes, which reduces accuracy. Because it depends on the needs and constraints of the task at hand, these findings make it difficult to determine which approach is the "best" to adopt with confidence.

**Results:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Embedding Technique** | **Training**  **Sample Size** | **Training Accuracy (%)** | **Test loss** |
| Custom-trained embedding  layer | 100 | 100 | 0.69 |
| Custom-trained embedding layer | 5000 | 98.48 | 0.39 |
| Custom-trained embedding  layer | 1000 | 97.3 | 0.66 |
| Custom-trained embedding  layer | 10000 | 98.4 | 0.345 |
| Pretrained word embedding  (GloVe) | 100 | 100 | 0.69 |
| Pretrained word embedding  (GloVe) | 5000 | 94.48 | 0.81 |
| Pretrained word embedding  (GloVe) | 1000 | 96.80 | 0.88 |
| Pretrained word embedding  (GloVe) | 10000 | 92.48 | 1.25 |

**Conclusion:**

In this experiment, the custom-trained embedding layer outperformed the pretrained word embedding layer, especially when using larger training sample sizes. However, the pretrained word embedding layer could be a preferable option in cases where computing resources are limited and only a small training sample size is available, despite the potential risk of overfitting.