**ADVANCED MACHINE LEARNING**

**TEXT AND SEQUENCE**

**ASSIGNMENT-4**

**RAJU RISHIKESH GOLLAPELLY**

**SAI SHARANYA DEVAMBATLA**

**Summary**

During this research, we experimented with binary (positive-negative) sentiment classification as a tool for forecasting movie reviews. The experiments were performed on a diversity of input sizes (100, 500, 1,000, and 10,000 reviews) from the IMDB review corpus of 50,000 reviews. There was a training set of 10,000 single most popular phrases and a different validation set of 10,000 reviews which were used to evaluate the models in terms of steadiness. The data before processing has gone into a pre-trained embedding layer accompanies with an optimization operation to get the best performance out of the model.

**Approach:**

The sentiment analysis in this paper uses the IMDB Movie Review dataset where every review expresses a general opinion ranging from positive to negative sentiment related to a film. In order to prepare the data for our neural network, we proceed to implement a preprocessing step which is vital. Among these reviews, all of them undergo a two-phase metamorphosis. Initially, individual words are converted into an electronic representation, commonly referred to as word embeddings. In each of these embeddings, all words have a fixed vector size, which shows its representation in relation to all other words. To create a vocabulary that everyone uses, it is important to stick to the fact that only the top 10,000 words that are used the most are considered.

Secondly, the reviews are turned into their original text being an array of numbers. Although this facilitates the processing for the artificial neural network, a challenge is on the horizon because reviews can have different lengths. However, this strategy achieves the goal of masking this inconsistency by padding the shorter reviews with arbitrary integer values, such that each sample will have an identical length. This pre-processing step ensures that a model gets processing accurately to achieve optimal performance.

To represent words in our sentiment analysis task, we explored two embedding techniques: there is a pre-trained GloVe layer and a custom-trained layer as well the trendy GloVe model which has achieved an impressive result due to the large size of its corpus is superior to other models for NLP tasks in capturing word relationships.

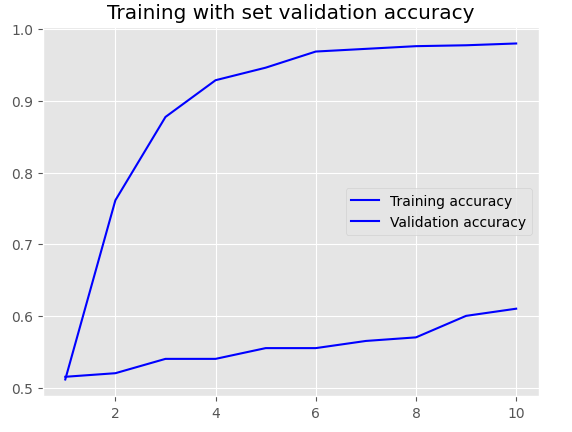
For our analysis, we built two separate embedding layers: a custom-trained one, aimed for the IMDB reviews, and another, which utilizes a pre-trained GloVe model. This makes it possible for us to know whether embedding is effective or not when different embedding methods are implemented.

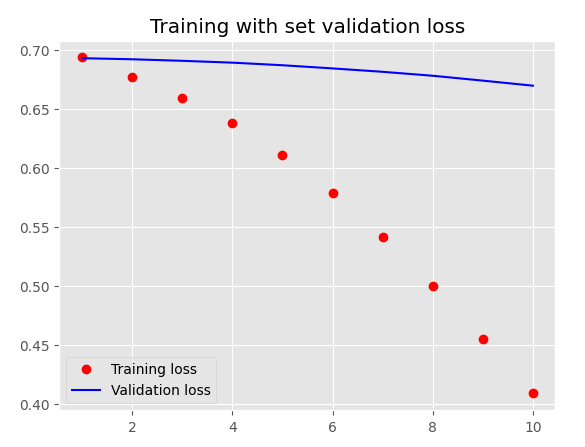
We studied the impact of amounting training data on the model’s capabilities. We trained two models using different sample sizes (100, 500, 1000, and 10,000 reviews) from the IMDB dataset: one who uses a custom embedding layer and the other who uses the pre-trained GloVe layer.

Then, in order to compare the efficacy of each embedding technique over a range of data amounts, we assessed their correctness on an independent test set.

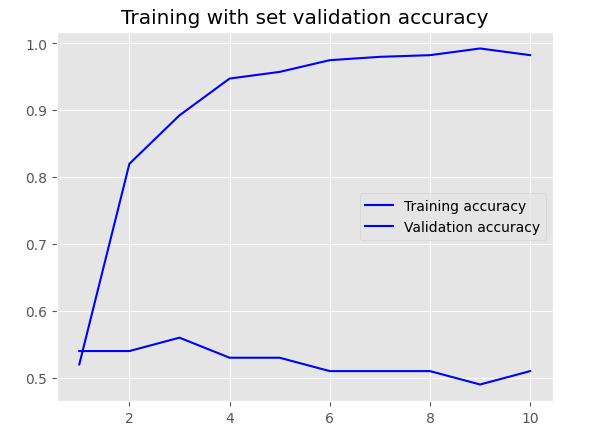
**User-defined embedding layer:**

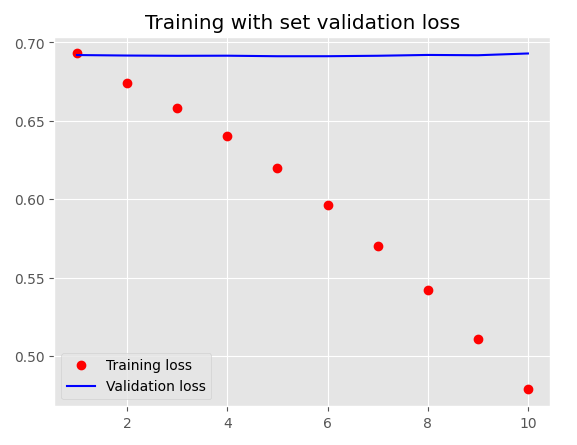
**Customer** Custom-trained embedding layer with sample size 100

****

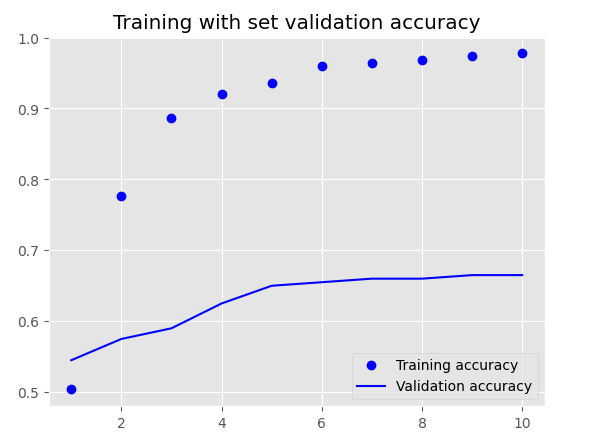
****

**Customer** Custom-trained embedding layer with sample size 500



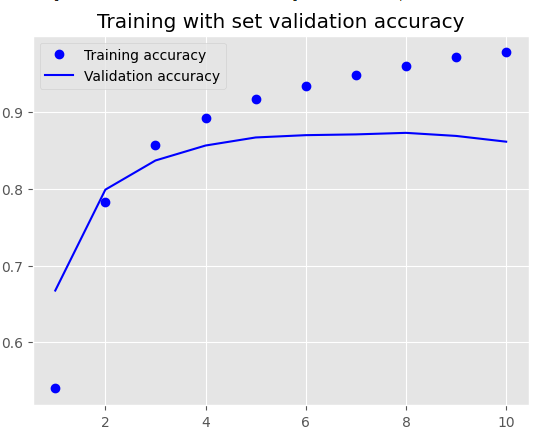
****

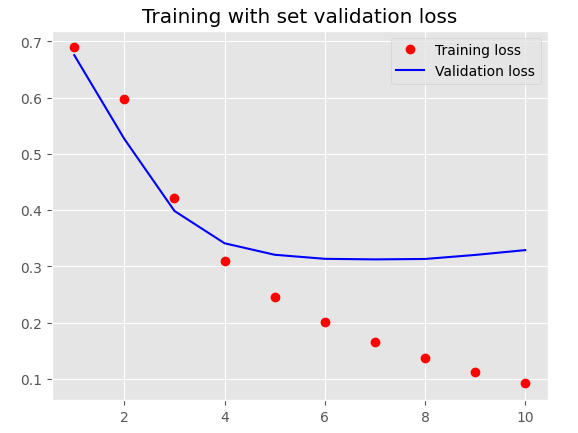
**Customer** Custom-trained embedding layer with sample size 1000





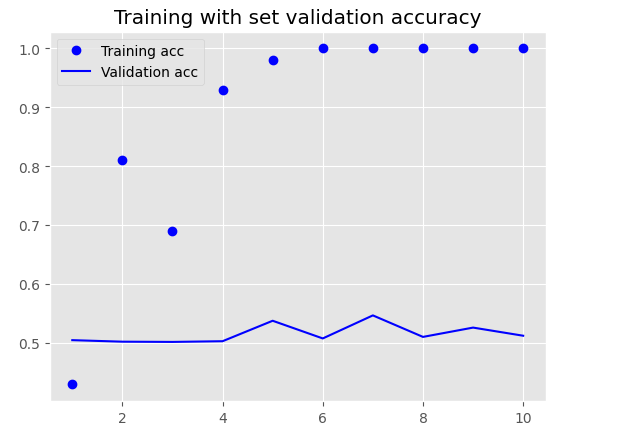
**Customer** Custom-trained embedding layer with sample size 10000

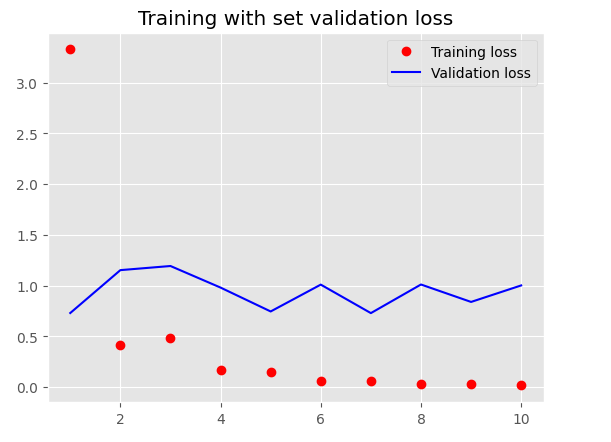




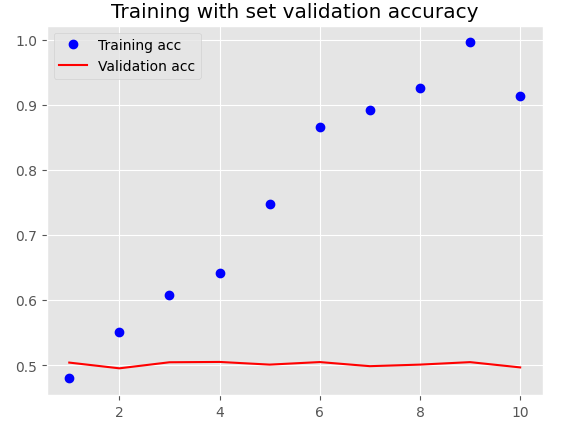
**Pretrained word embedding layer (GloVe):**

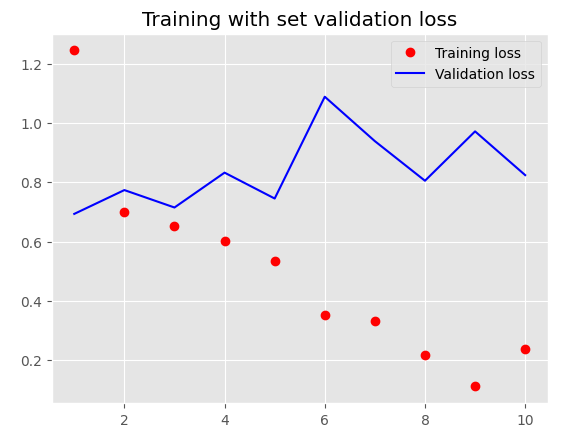
Pretrained word embedding layer (GloVe) with sample size 100.



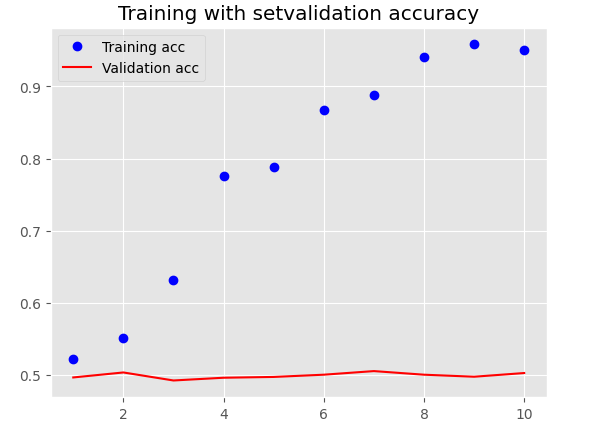


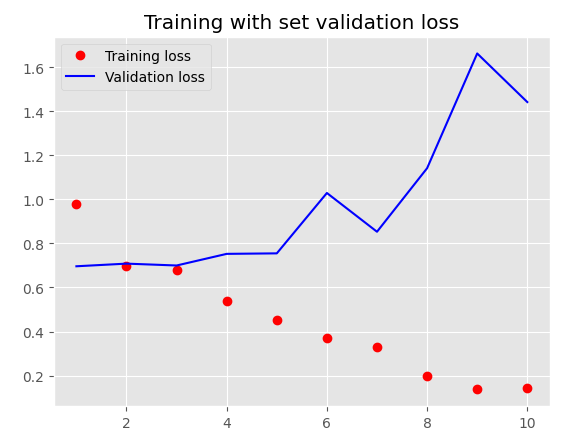
Pretrained word embedding layer (GloVe) with sample size 500.



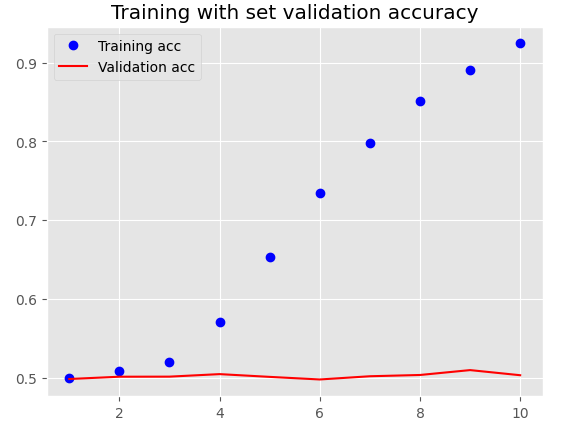


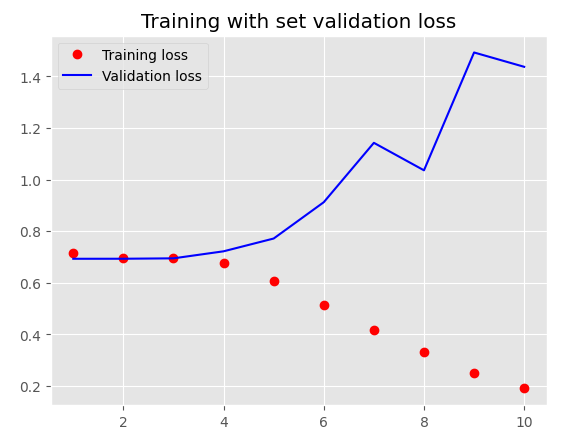
Pretrained word embedding layer (GloVe) with sample size 1000





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:0.694- Acc:0.499** | **100** |
| Custom-trained embedding layer | 150 | **500** | **Loss:0.685- Acc:0.549** | **98.2** |
| Custom-trained embedding layer | 150 | **1000** | **Loss:0.664- Acc:0.613** | **97.8** |
| Custom-trained embedding layer | 150 | **10000** | **Loss:0.337- Acc:0.857** | **97.8** |
| Pretrained word embedding layer (GloVe) | 150 | **100** | **Loss:1.120- Acc:0.501** | **100** |
| Pretrained word embedding layer (GloVe) | 150 | **500** | **Loss:0.815- Acc:0.499** | **91** |
| Pretrained word embedding layer (GloVe) | 150 | **1000** | **Loss:1.489- Acc:0.506** | **95** |
| Pretrained word embedding layer (GloVe) | 150 | **10000** | **Loss:1.414- Acc:0.501** | **92** |

**Custom-trained embedding layer analysis:**

It was the custom layer that achieved the most impressive accuracy between 97% and 100%. The highest accuracy (100%) was observed for the smallest training data set (100 reviews). This shows the custom embeddings probably are best for covering the details of sentiment in records with IMDB reviews, possibly because of their specific training on this field.

**Pretrained word embedding layer (GloVe) analysis:**

As the training data volume increased from 100 to 10,000 samples, the accuracy of the pre-trained GloVe model ranged from 91% to 100. The pre-trained model was most successful (100% accuracy) while training on the least amount of data (100 reviews). This is due to the fact that pre-trained embeddings, such as GloVe, which carry a lot of meaning from the large text corpus work even with the limited size training set. This is why the pre-trained model did well with just 100 reviews. On the contrary, as you increase the training sample size, your model might find it hard to understand the particular details of this task (sentiment analysis of IMDB reviews). This may result in a reduction in accuracy.

Pre-trained embeddings with a large dataset can result in the overfitting of the model and a less accurate model performance, as we have shown above. The model that is trained with an overfit condition could not process new data because it gets better and better in remembering the training set. It is difficult to determine if custom or pre-trained approach is always better, as their respective performance highly depends on the goals and practical limitations of a specific project. Generally, in the present experiment, the custom-trained embeddings turned out to be better than the pre-trained ones most especially after being supplied with plenty of training data. If the computational power and the training data will be limited then the pre-trained model could be a better choice despite the overfitting risk is involved.

**Conclusion:**

The performance of pre-trained embeddings in sentiment analysis may be affected by the number of training data used because it could be very large or very small. However, although the GloVe model shows good results in general semantic relations, it can lack the specifics of a task if data volume is very high (eg. IMDB sentiment analysis).This can lead to two potential issues:

**1.**Inaccurate results might be the result of the pre-trained embeddings' inability to accurately extract task-specific features.   
2.When this data is merged with pre-trained word embeddings the model may be overfitting to the training set, thus, reducing the accuracy and generalization ability of the model to newer data.

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

**Elaborating Embedding Options for Smaller Datasets:**

In the case of tasks with very little training data and features, customizing only the embedding layer would be suitable. Consequently, the model can concentrate on the peculiarities on the small dataset that will possibly yield a higher accuracy in comparison to pre-trained models.

**Points to remember:**

Pre-trained embeddings may lose their effectiveness when a big data is involved as it cannot capture task-specific fine details efficiently.  
  
This poses the question of accuracy and overfitting and thus is not advantageous to the model performance.  
  
For tasks with small data set, custom-trained embeddings could be the effective choice because they focus on the particular data characteristics.  
  
The perfect embedding strategy lies in the needs of the project and size of the data.