SENTIMENT ANALYSIS ON IMDB MOVIE REVIEWS USING AN EMBEDDING LAYER AND A PRE-TRAINED EMBEDDING

Spring-2023 **AML Report** **Niharika Kolliboyana**

Executive Summary

The IMDB dataset is a binary classification problem where the goal is to predict whether a movie review is positive or negative. The dataset consists of 50,000 reviews in which we cutoff reviews after 150 words,restrict training samples to 100,500,1000,10000,validate on 10000 samples,and by considering only top 10000 words. Pre-Processing is performed on the data.Later, we feed data to the embedding layer and also to a pretrained embedding model,and evaluate the performance by trying different approaches.

Problem

The IMDB dataset is a binary classification problem where the goal is to predict whether a movie review is positive or negative,and which approach showed better performance.

Technique

Dataset and Preprocessing:

The IMDB dataset is a collection of movie reviews with sentiment labels (positive or negative). The dataset is preprocessed by converting each review into a sequence of word embeddings, where each word is represented by a vector of fixed size. The maximum vocabulary size is set to 10,000. Also, the reviews, which were originally a sequence of words, were changed into a sequence of integers, where each integer represented a different word. Even though we now have a list of numbers, they are not appropriate for our neural model's input. The integers must be transformed into tensors. The list of integers could be transformed into a tensor with integer data type and shape (samples, word indices). To accomplish that, we need to make sure that each sample is the same length, thus we would need to pad each review with dummy words (integers) to make sure that they are all the same length.

Approach:

For this study, we explored two different approaches to creating word embeddings for our IMDB review dataset: a custom-trained embedding layer and a pretrained word embedding layer using the GloVe model.

The GloVe model used in our study is a widely-used pretrained word embedding model that is trained on large corpora of text data. It is known for its ability to capture semantic and syntactic relationships between words, making it a popular choice for natural language processing tasks. In our study, we used the 6B version of the GloVe model, which contains 6 billion tokens and 400,000 words, trained on a corpus of Wikipedia data and Gigaword 5.

To analyze the effectiveness of different embedding techniques, we used the IMDB review dataset and implemented two different embedding layers: one with a custom-trained embedding layer and the other with a pre-trained word embedding layer. We tested both of these models with varying training sample sizes (100, 500, 1000, 10000) and compared their respective accuracies.

Firstly, we implemented a custom-trained embedding layer for the IMDB review dataset. We trained this layer on different samples of the dataset and measured the accuracy of each model using a testing set. We then compared these accuracies with a model that used a pre-trained word embedding layer, which was also tested on varying sample sizes.

Results:

Embedding technique	Training sample size	Accuracy (%)	Embedding technique	Training sample size	Accuracy(%)
Custom-trained embedding layer	100	97.5	Pretrained word embedding layer (GloVe)	100	100
Custom-trained embedding layer	500	97.5	Pretrained word embedding layer (GloVe)	500	100
Custom-trained embedding layer	1000	98.5	Pretrained word embedding layer (GloVe)	1000	95.4
Custom-trained embedding layer	10000	97.8	Pretrained word embedding layer (GloVe)	10000	92.9

Custom-trained embedding layer:

The accuracy obtained using the custom-trained embedding layer ranged from 97.5% to 98.5%, depending on the training sample size.
The highest accuracy was obtained with a training sample size of 1000.
One possible reason for the high accuracy with this technique is that the embedding layer is trained specifically for the task at hand (IMDB review sentiment classification), which may result in more effective representations of the text data.
However, it's also worth noting that the accuracy did not continue to improve significantly beyond a training sample size of 1000, suggesting that the benefits of additional training data may be limited for this technique.

Pretrained word embedding layer (GloVe):

The accuracy obtained using the pretrained word embedding layer (GloVe) ranged from 92.9% to 100%, depending on the training sample size.
The highest accuracy was obtained with a training sample size of 100.
One possible reason for the high accuracy with a small training sample size is that the pretrained embeddings capture a lot of the underlying semantic information in the text, which can make them effective even with limited training data.

However, as the training sample size increases, the pretrained embeddings may not be as effective at capturing the nuances of the specific task at hand, which can lead to decreased accuracy.

Additionally, as noted in the prompt, the model quickly starts overfitting when using the pretrained embeddings with larger training sample sizes, leading to a decrease in accuracy.

Based on these results, it's difficult to definitively determine which technique is the "best" to use, as it depends on the specific needs and constraints of the task at hand. However, the custom-trained embedding layer generally outperformed the pretrained word embedding layer in this experiment, particularly with larger training sample sizes. If computational resources are limited and a small training sample size must be used, the pretrained word embedding layer may be a more effective option, but caution should be taken to avoid overfitting.