Assignment 4
Text and Sequence

Abhinav Reddy Yelipeddi

Summary:

With an emphasis on enhancing model performance with sparse data, the goal of this project is to investigate the application of Transformer topologies and Recurrent Neural Networks (RNNs) to text and sequence data. The aim is to validate on 10,000 samples, truncate reviews to 150 words, limit training samples to 100, and take into account the top 10,000 vocabulary words using the IMDB dataset. The assignment examines how well a learned embedding layer performs in comparison to pretrained word embeddings, taking into account variations in the number of training samples. The results will shed light on the applicability of RNNs and Transformers for text categorization, the influence of embedding techniques, and methods for performance optimization in the face of limitations.

Problem:

With an emphasis on enhancing model performance with sparse data, the goal of this project is to investigate the application of Transformer topologies and Recurrent Neural Networks (RNNs) to text and sequence data. The aim is to validate on 10,000 samples, truncate reviews to 150 words, limit training samples to 100, and take into account the top 10,000 vocabulary words using the IMDB dataset. The assignment examines how well a learned embedding layer performs in comparison to pretrained word embeddings, taking into account variations in the number of training samples. The results will shed light on the applicability of RNNs and Transformers for text categorization, the influence of embedding techniques, and methods for performance optimization in the face of limitations.

Technique

Dataset Description:

The IMDB dataset contains movie reviews with sentiment classifications (positive or negative).

Preprocessing:

Word embeddings are created from each review, with a fixed-size vector representing each word. There is a 10,000 word limit on the vocabulary. Reviews are transformed into integer sequences, where each integer stands for a different word. By using padding to ensure constant length, integers are transformed into tensors to make input into the neural network easier.

Approach:

For our IMDB review dataset, we investigated two word embedding generation techniques: a custom-trained embedding layer and a pretrained word embedding layer based on the GloVe model. In our work, we used the well-known pretrained word embedding model GloVe, which is trained on large amounts of text data. Because it can capture the syntactic and semantic relationships between words, it is a popular choice for natural language processing jobs.

We used a corpus of Wikipedia data and Gigaword 5, which contains 400,000 words and 6 billion tokens, to train the 6B version of the GloVe model. To evaluate the effectiveness of different embedding techniques, we constructed two different embedding layers using the IMDB review dataset: one with a pre-trained word embedding layer and the other with a custom-trained embedding layer.

Using training sample sizes that varied—100, 5000, 1000, and 10,000—we investigated the two models' accuracy. First, we created a specially trained embedding layer using the IMDB review dataset. After training each model on different dataset samples, we evaluated its accuracy using a testing set. Then, using a model with a pre-trained word embedding layer that was also tested on different sample sizes, we contrasted these precisions.

Results:

Embedding technique	**Training sample size**	**Accuracy (%)**
Custom-trained embedding layer	**100**	100
Custom-trained embedding layer	**5000**	83.1
Custom-trained embedding layer	**1000**	56.9
Custom-trained embedding layer	**10000**	85.1
Pretrained word embedding layer (GloVe)	**100**	100
Pretrained word embedding layer (GloVe)	**5000**	50.4
Pretrained word embedding layer (GloVe)	**1000**	49.4
Pretrained word embedding layer (GloVe)	**10000**	49.9

Custom-trained embedding layer:

The accuracy attained with the custom-trained embedding layer ranged from 49.4% to 100%, depending on the size of the training sample. The maximum accuracy was obtained with a training sample size of 100. The excellent accuracy of this method may be due to more efficient text data representations because the embedding layer is specifically trained for the job at hand (sentiment categorization of IMDB reviews).

Pretrained word embedding layer (GloVe):

The accuracy attained using the pretrained word embedding layer (GloVe) ranged from 49% to 100%, depending on the size of the training sample. A maximum accuracy of 100 training samples was achieved. The high accuracy with a little training sample size may be explained by the pretrained embeddings' ability to capture a significant amount of the text's underlying semantic information, making them helpful even with little training data. Nevertheless, as the training sample size increases, the pretrained embeddings may not be as good at catching the finer points of the particular job at hand, thus leading to decreased accuracy. Additionally, as the prompt states, using the pretrained embeddings with greater training sample sizes leads to the model quickly overfitting, which reduces accuracy.

Additionally, as the prompt states, using the pretrained embeddings with greater training sample sizes leads to the model quickly overfitting, which reduces accuracy. Because it depends on the needs and constraints of the task at hand, these results make it difficult to determine with certainty which approach is the "best" to utilize. But generally, the custom-trained embedding layer outperformed the pretrained word embedding layer in this experiment, particularly when training with larger training sample sizes**. If computer resources are limited and a small training sample size is needed, the pretrained word embedding layer might be a "better choice," despite the danger of overfitting.**

THANK YOU !