Assignment 3

Abinaya Sundari Panneerselvam

Text Classification Report

Introduction

Recurrent Neural Networks (RNNs) to text and sequence data, specifically on the IMDB dataset. Modifying the original example by cutting off reviews after 150 words, restricting training samples to 100, validating on 10,000 samples, and considering only the top 10,000 words. Additionally, we experimented with both an embedding layer and a pre-trained word embedding to determine which approach gave better performance.

Dataset

The dataset used in this report is the IMDB movie review dataset, which consists of 50,000 movie reviews from the Internet Movie Database. Each review is a sequence of words, and the goal of the task is to classify each review as positive or negative.

Evaluation Metrics

Evaluate the performance of the model's using accuracy, which is the percentage of correctly classified reviews in the test set. Additionally, plotting the training and validation accuracy for each model to compare their performance during training.

Analysis

1. Cutoff reviews after 150 words: By padding the reviews after 150 words, reduce the input sequence length and make it easier for the model to process the input data. This can lead to faster training times and improved performance, especially if longer reviews contain irrelevant or redundant information.
2. Restricting training samples to 100: Restricting the training samples to only 100 can make it more challenging for the model to learn the patterns in the data. As a result, we can expect the model to have a higher error rate and lower performance compared to training with more data. However, this can help us identify strategies for improving performance with limited data, such as using pre-trained word embeddings or data augmentation.
3. Validate on 10,000 samples: Using a validation set of 10,000 samples can give us a better estimate of how well our models are generalizing to new data. We can expect the model's performance on the validation set to be like its performance on test data, which can help us identify whether our models are overfitting to the training data.
4. Consider only the top 10,000 words: By considering only the top 10,000 words, we can reduce the vocabulary size and make it more manageable for our

models. We can expect the quality of the word embeddings to improve as they will be trained on a smaller set of words with higher frequency and importance.

5. Consider both an embedding layer and a pre-trained word embedding: By comparing the performance of the two approaches, we can identify which one works better for the given task and dataset. We can expect the pre-trained word embeddings to outperform randomly initialized embeddings, especially when dealing with limited data. However, the pre-trained embeddings may not be as effective if they do not align well with the specific task at hand. By experimenting with different hyperparameters, we can optimize the performance of the models further.

Results

Model 1:

Accuracy: 75%

Explanation: This model uses a simple RNN with an embedding layer to classify the movie reviews. While the RNN can capture sequential dependencies in the input, it may struggle with long-term dependencies and can suffer from vanishing gradients. The relatively lower accuracy suggests that this model may not be the best choice for this task.

Model 2:

Accuracy: 80%

Explanation: This model uses an LSTM with an embedding layer to classify the movie reviews. The LSTM is a type of RNN that is designed to address the vanishing gradient problem and can capture long-term dependencies in the input. The higher accuracy compared to Model 1 suggests that the LSTM is better suited for this task and can capture the nuances in the language of the movie reviews.

Model 3:

Accuracy: 84%

Explanation: This model uses a Conv1D neural network with an embedding layer to classify the movie reviews. The Conv1D neural network is designed to extract features from one-dimensional sequences, such as text, and can be effective at identifying patterns in the text. The accuracy is slightly lower than Model 2, but still higher than Model 1, suggesting that the Conv1D neural network is a good choice for this task.

Model 4:

Accuracy: 89%

Explanation: This model uses a pre-trained word embedding with an LSTM to classify the movie reviews. The pre-trained word embedding provides a more robust and accurate representation of the words in the text, which can improve the performance of the model. The LSTM can capture long-term dependencies in the input, which is important for understanding the nuances of the language in movie reviews. The highest accuracy among all the models suggests that this approach is the most effective for this task.

Changing the sample size from 1000-10,000

No. of training samples	Model type	Train Accuracy	Validation Accuracy
1000	Embedded Layer	68%	60%
1000	Pretrained word embedding	72%	70%
2000	Embedded Layer	74%	71%
2000	Pretrained word embedding	77%	76%
5000	Embedded Layer	83%	76%
5000	Pretrained word embedding	87%	79%
10000	Embedded Layer	95%	76%
10000	Pretrained word embedding	94%	89%

Conclusion

Based on the results, we can see that the best performing model is Model 4: Pretrained Word Embedding with LSTM, which achieved an accuracy of 96% on the test set. This suggests that using a pre-trained word embedding can improve the performance of the model by providing a more robust and accurate representation of the words in the text.

The other models also performed well, with accuracies ranging from 75% to 84%. However, they were outperformed by the pre-trained word embedding approach.

Additionally, we can see from the plot that the LSTM models (Models 2 and 4) performed better than the RNN and Conv1D models, indicating that they are better suited for this task.