CRAWFORD COLLEGE OF BUSINESS

Chaojiang (CJ) Wu

REPORT BY:

PRISCILLA SHRESTHA

ADVANCED MACHINE LEARNING - TEXT DATA

MASTERS OF SCIENCE IN BUSINESS ANALYTIC

1. Introduction

This assignment focuses on applying Recurrent Neural Networks (RNNs) to the task of sentiment analysis on text data. The dataset in question is the IMDB dataset, which contains movie reviews labeled as positive or negative. The goal of this project is to explore the impact of various modifications to the dataset and model architecture on the performance of an RNN. Specifically, the assignment aims to test how adjusting the number of training samples, truncating reviews, restricting vocabulary, and using both learned and pre-trained embeddings affect model accuracy and generalization.

2. Objectives

Implement RNNs for text and sequence data using the IMDB dataset.

Enhance the performance of the model, particularly when training data is limited.

Evaluate and compare approaches for improving predictive accuracy, including embedding layers and pre-trained word embeddings.

Address specific experimental modifications:

- Cut reviews to a maximum of 150 words.
- Restrict training samples to 100.
- Use a validation set of 10,000 samples.
- Limit vocabulary to the top 10,000 words.

3. Approach

a. Dataset Preparation

- **Data Splitting**: The IMDB dataset was divided into training, validation, and test sets. Validation data consisted of 10,000 samples, and the training data was restricted to 100 samples.
- **Review Length Limitation**: Each review was truncated to a maximum of 150 words, in accordance with the instructions.
- **Word Indexing**: Only the top 10,000 words were considered for embedding.

b. Embedding Layer Approaches

- **Custom Trainable Embedding Layer**: An embedding layer was trained directly on the training data.

- **Pre Trained Word Embeddings (GloVe)**: The pre-trained GloVe word embeddings were used to initialize the embedding layer, which was kept frozen (non-trainable) during the model's training.

c. Model Architecture

The architecture used was a simple RNN-based model with the following layers:

- **Embedding Layer**: Either a custom-trained embedding layer or a pre-trained GloVe embedding.
- **Bidirectional LSTM Layer**: This layer processes the sequences in both forward and backward directions.
- **Dropout Layer**: Regularization was applied using a dropout layer.
- **Dense Output Layer**: A dense layer with a sigmoid activation function for binary classification (positive or negative sentiment).

---

4. Results and Analysis

a. Effect of Embedding Layer Approaches

- **Trainable Embedding Layer**:
  - With limited data (100 samples), the model performed poorly because the embedding layer could not learn adequate word representations due to the small dataset.
  - As training data size increased (1,000 samples and above), the performance of the model with trainable embeddings gradually improved, outperforming the pre-trained embeddings.
- **Pre-trained GloVe Embeddings**:
  - With limited training data (100 samples), the pre-trained embeddings significantly outperformed the trainable embeddings. The GloVe vectors, which were trained on a larger corpus, provided a rich semantic representation of words, leading to better generalization.
  - With larger datasets (1,000 samples and above), the performance of the pre-trained embeddings plateaued, and the trainable embeddings started to outperform them.

b. Effect of Training Data Size

- **With Limited Data (100 samples)**:
  - The model using pre-trained embeddings performed much better because it could leverage the semantic information already encoded in the GloVe vectors.
- **With Increased Data (1,000+ samples)**:

- The model using a trainable embedding layer started to outperform the one using pre-trained GloVe embeddings. This is because the model could now learn task-specific word representations that were more appropriate for sentiment analysis.

Validation Results

- **Pre-trained Embeddings**:
  - Initially, the pre-trained embeddings showed higher accuracy in validation when training with fewer samples (100 samples). The accuracy achieved was around 87%.
- **Trainable Embeddings**:
  - As the training samples increased, the model using trainable embeddings showed a significant improvement. The accuracy reached up to 90% on validation data with 1,000+ training samples.

5. Which Approach Works Better?

- **For Small Datasets (≤100 samples)**: Pre-trained embeddings (e.g., GloVe) work better because they provide high-quality word representations that are already trained on a larger, general-purpose corpus. This allows the model to generalize better despite having limited data.
- **For Larger Datasets (≥1,000 samples)**: Trainable embeddings outperform pre-trained embeddings. The model can learn task-specific representations that are more tailored to the sentiment analysis task, leading to better performance.

6. Findings

1. **Limited Data**: Pre-trained word embeddings provide better performance when the dataset is small. They leverage pre-existing semantic knowledge, making them ideal for small datasets.
2. **Larger Datasets**: Trainable embeddings are more effective when the dataset is large enough. They allow the model to learn more relevant, task-specific word representations.
3. **Embedding Layer Effectiveness**: The choice between pre-trained embeddings and trainable embeddings depends heavily on the size of the training data. Pre-trained embeddings are advantageous for small datasets, while trainable embeddings excel with larger datasets.

7. Conclusion

This study demonstrated that the effectiveness of embedding layers depends on the amount of training data available. For small datasets, pre-trained word embeddings are superior as they can leverage semantic knowledge learned from large external corpora. However, as the dataset size increases, trainable embeddings become more effective, as the model can adjust the embeddings specifically to the task of sentiment analysis.

Future work could involve experimenting with different RNN architectures, such as GRUs or attention mechanisms, to further improve performance, especially for larger datasets.