ASSIGNMENT 3: CONVOLUTIONAL NEURAL NETWORK (BA_64061_001)

Prachi Agrawal

Overview and Objective

The task of binary classification on the IMDB dataset aims to categorize movie reviews into positive or negative sentiments. The dataset consists of 50,000 reviews, with a focus on the top 10,000 most frequent words. The project involves training models on different subsets of the data (100, 1000, 3000, 5000, 10000, 20000 samples) and validating the results using a fixed set of 10,000 samples. After preprocessing, the data is fed into a pretrained embedding model and evaluated using different strategies to compare performance.

Dataset Overview

- **Reviews**: The dataset contains 50,000 movie reviews.
- **Vocabulary**: The model only considers the top 10,000 most frequent words in the reviews.
- **Data Split**: The data is divided into training sets of varying sizes (100, 1000, 3000, 5000, 10000, 20000), with **10,000 samples** allocated for validation.
- **Text Length**: Reviews are truncated to **150 words**.

Approach

1. **Data Preprocessing**:
    - **Limiting training samples**: The training set size is constrained to specific values.
    - **Truncating Reviews**: Each review is stopped after 150 words to ensure consistency.
    - **Top 10,000 Vocabulary**: Only the 10,000 most frequent words are retained in the dataset, reducing the size of the vocabulary.
2. **Word Embedding**:
    - Word embeddings represent words as vectors in a high-dimensional space. Unlike one-hot encoding, word embeddings capture semantic relationships between words and enable the model to better understand the meaning of sentences.

- **Pretrained Word Embeddings**: Pretrained embeddings like **GloVe**, **Word2Vec**, and **FastText** are used to provide semantic representations for words that have been trained on vast amounts of unannotated text. These embeddings can serve as features in downstream tasks, saving both time and computational resources compared to training embeddings from scratch.

3. **Model Architecture**:
    - **Recurrent Neural Network (RNN)**: The model architecture uses an RNN with an embedding layer to process sequential data, where each review is treated as a sequence of words.
    - **Embedding Layer**: The RNN model integrates an embedding layer that maps each word to a dense vector. This layer is either pretrained (with embeddings like GloVe) or trained along with the model.
    - **Hyperparameters**:
        - The model includes a specified number of **RNN units** and **embedding dimensions**, influencing the model's capacity to learn complex patterns.
4. **Training and Validation**:
    - The model is trained with the IMDB dataset, with different training sample sizes (100, 1000, 3000, 5000, 10000, 20000 reviews). The model's performance is validated on a fixed set of 10,000 samples.
    - Performance metrics (such as **accuracy** and **loss**) are recorded for comparison across different methods and training sample sizes.

Methodology

1. Baseline Model

- **Model Description**: The baseline model is a simple RNN integrated with an embedding layer that learns embeddings directly from the data.
- **Model Architecture**:
    - The RNN processes the text data sequentially, while the embedding layer converts words into dense vectors.

- **Hyperparameters**:
    - Number of **RNN units** and **embedding dimensions** are specified and tuned.
- **Validation and Test Results**:
    - The performance of the baseline model is evaluated by measuring **validation accuracy**, **test accuracy**, and **loss**.

2. Model with Pretrained Word Embeddings

- **Utilization of Pretrained Word Embeddings**: Instead of training embeddings from scratch, pretrained embeddings like **GloVe** are loaded into the model. This allows the model to leverage existing semantic representations, improving its understanding of textual features.
- **Process of Loading Pretrained Embeddings**:
    - The pretrained embeddings are integrated into the model, enabling it to use word vectors trained on large corpora.
- **Validation and Test Results**:
    - The model incorporating pretrained embeddings is evaluated using **validation** and **test accuracy** and **loss metrics**.

3. Varying Training Set Size

- **Training Set Size Adjustment**: The training set size is systematically altered to assess the impact of data quantity on model performance. The training sizes considered are 100, 500, 1,000, and 100,000 samples.
- **Validation and Test Results**:
    - The model's performance is tracked across different training set sizes to evaluate the influence of the amount of training data on performance.

4. Comparison of Embedding Layer vs. Pretrained Embeddings

- **Performance Analysis**:
    - The performance of models utilizing the **embedding layer** versus those using **pretrained embeddings** is compared. This comparison is made across different training set sizes to understand how the embedding strategy interacts with the volume of training data.

- The results provide insights into how pretrained embeddings may offer better semantic representations, especially when larger training datasets are used.

RESULTS:

MODEL	ACCURACY	LOSS	VALIDATION ACCURACY	VALIDATION LOSS
One Hot model	79.0%	0.460	79.4%	0.454
Trainable Embedding Layer	79.8%	0.436	80%	0.435
Masking Padded Sequences in the Embedding Layer	79.8%	0.436	79.8%	0.437
Model with Pretrained GloVe Embeddings	79.8%	0.453	79.7%	0.455

MODEL	ACCURACY	LOSS
Embedding Layer of 100 Training Samples	75.1%	0.520
Pretrained Embedding Layer of 100 Training Samples	77.2%	0.477
Embedding Layer of 1000 Training Samples	81.0%	0.436
Pretrained Embedding Layer of 1000 Training Samples	78.9%	0.448
Embedding Layer of 3000 Training Samples	79.8%	0.472
Pretrained Embedding Layer of 3000 Training Samples	79.1%	0.447
Embedding Layer of 5000 Training Samples	79.6%	0.446
Pretrained Embedding Layer of 5000 Training Samples	79.2%	0.442

Embedding Layer of 10000 Training Samples	79.4%	0.453
Pretrained Embedding Layer of 10000 Training Samples	78.3%	0.462
Embedding Layer of 20000 Training Samples	80.6%	0.488
Pretrained Embedding Layer of 20000 Training Samples	78.4%	0.459

Expected Outcome and Conclusion

For small datasets (100-1000 samples), pretrained embeddings consistently outperformed trainable embeddings, showing higher validation accuracy and lower loss. Pretrained embeddings leverage external linguistic knowledge, which is crucial when data is limited.

As the training dataset increases (3000-20000 samples), the performance gap between pretrained and trainable embeddings diminishes. Both methods achieve comparable results, suggesting that with enough data, models can learn meaningful word representations without pretrained embeddings.

The model with pretrained embeddings and 20,000 samples showed the best overall performance, but the differences between pretrained and trainable embeddings were minimal for larger datasets.

In conclusion, for limited data, pretrained embeddings are highly recommended. As the dataset grows, trainable embeddings become just as effective and offer flexibility to adapt to the specific dataset. The choice depends on the available data size and the task requirements.