Text and sequence data assignment

Report by Perika Venkata naga sai goud

1. Introduction
   The purpose of this experiment is to evaluate two Recurrent Neural Network (RNN) models that were trained using the IMDb movie review dataset. Randomly initialized word embeddings are used in the first model, whereas the second model includes embeddings of pretrained GloVe words.

2. Dataset:
   I made use of the 50,000 movie reviews on IMDb that have been classified as either good or negative. Every review is represented by a series of integers, each of which is the dictionary index for a word. I padded or shortened the reviews to a predetermined 150-word limit.

3. Model Architecture
   Using Keras, I created two RNN models: one with pretrained embeddings and one with randomly initialized embeddings.

   RNN Model:

   - Embedding Layer: Input length of 150, embedding dimension of 32.
   - Bidirectional LSTM Layer (64 units, return sequences).
   - Dropout Layer (rate = 0.5) to prevent overfitting.
   - Batch Normalization Layer.
   - Bidirectional LSTM Layer (32 units).
   - Dropout Layer (rate = 0.5).
   - Batch Normalization Layer.
   - Dense Layer with sigmoid activation for binary classification

   Pretrained RNN Model:

   - Embedding Layer initialized with GloVe embeddings (100-dimensional).
   - Bidirectional LSTM Layer (64 units, return sequences).
   - Dropout Layer (rate = 0.5).
   - Batch Normalization Layer.
   - Bidirectional LSTM Layer (32 units).
   - Dropout Layer (rate = 0.5).
   - Batch Normalization Layer.
   - Dense Layer with sigmoid activation for binary classification

4. Model Training
   Using the RMSprop optimizer and binary cross-entropy loss function, I trained each model for ten epochs with varying sample sizes (100, 500, and 1000 training samples). A subset of the test dataset, comprising 5000 samples, is used to evaluate the models.

5. Results and Discussion

Performance of RNN Model with Random Embeddings

For 100 training samples:

- Test Loss: 0.6923078894615173
- Test Accuracy: 51.42%
- Training and validation accuracies remained almost constant throughout epochs.
- Both training and validation losses plateaued quickly, indicating limited learning,

For 500 training samples :

- Test loss: 0.8277802467346191
- Test Accuracy: 49.61%
- The model should improved performance compared to the 100-sample case.

For 1000 training samples

- Test loss: 1.0110595226287842
- Test accuracy: 67.91%
- There was a slight decrease in accuracy compared to the 500-sample case, possibly due to overfitting.

Performance of Pretrained RNN Model:

For 100 training samples:

- Test loss: 0.6975911259651184
- Test Accuracy: 51.41%
- Similar to the RNN model with random embeddings, the pretrained model should limited improvement.

For 500 training samples:

- Test loss: 0.6807900071144104
- Test accuracy: 58.02%
- The performance was similar to the model with random embeddings.

For 1000 training samples:

- Test loss: 1.566427230834961
- Test accuracy: 52.66%
- Despite using pretrained embeddings, the model's performance was comparable to that of random embeddings.

6. Conclusion:

- The experiment showed that the model performed better when the training sample size was increased.
- When compared to random embeddings, pretrained embeddings did not considerably improve the model's performance.
- Overfitting is evident in both models, particularly when training sample sizes are larger.
- s Better performance could be obtained by conducting more experiments with various hyperparameters, architectures, and embedding sizes.