Assignment-4

Esha Reddy Emani

Introduction:

This task examines how RNNs and Transformers can be used with text and sequence data to enhance model performance when faced with limited data. In the IMDB sentiment analysis example from Chapter 6, you will try out several changes such as limiting reviews to 150 words, using only 100 training samples, validating on 10,000 samples, and focusing on the top 10,000 words. In addition, you will evaluate the effectiveness of a trainable embedding layer compared to pretrained word embeddings and analyze how different training sample sizes affect performance. These activities offer useful information on improving NLP models, handling data constraints, and selecting suitable embedding techniques for better predictions.

Problem Statement:

The task aims to explore and evaluate the application of advanced neural network architectures, such as Recurrent Neural Networks (RNNs) and Transformers, for text classification tasks, specifically sentiment analysis on the IMDB dataset. The study investigates the challenges of working with limited data and explores strategies to improve model performance. Key aspects include evaluating the impact of truncating input sequences, restricting training sample sizes, and limiting vocabulary size. Furthermore, the report compares the performance of trainable embedding layers against pretrained word embeddings, examining the conditions under which each approach is more effective. The ultimate goal is to identify optimal methods for handling sequence data and achieving robust performance under varying data constraints.

Methodology:

Preparation of data for analysis:

The IMDB dataset contains a vocabulary that only includes the 10,000 most common words. All reviews are shortened to 150 words, and padding is added to maintain consistent input length.

Setup for training and validation:

The tests are done with different sizes of training samples, beginning with 100 samples and then gradually going up to 10,000 and 15,000 samples. A set of 10,000 samples is consistently used for validation purposes.

Creating a model with adjustable embeddings:

A neural network is created in sequence with an embedding layer to produce word vectors, an LSTM layer for temporal relationships, and a dense output layer that utilizes a sigmoid activation function for binary categorization. The model can learn word representations through training because the embedding layer is trainable.

Integration of pretrained embeddings:

The pretrained GloVe word embeddings are used instead of the embedding layer for comparison.

Pretrained embeddings are uploaded, and the embedding layer is set up with these preexisting word vectors, utilizing external linguistic knowledge to enhance model performance.

Development, assessment, and examination:

The RMSprop optimizer and binary cross-entropy loss function are used to train and validate the models for 10 epochs. Analysis is done on performance patterns in trainable and pretrained embeddings, with the impact of different training sample sizes on accuracy and loss shown visually to identify the best strategy.

Results:

Method	Training Size	Training Accuracy	Validation Accuracy	Test Accuracy
Using Embedding Layer	100	58	50.27	50
Using Embedding Layer	10000	93.01	85.64	85.7
Using Embedding Layer	17485	92.69	84.29	84.42
Using Embedding Layer	25000	95.36	84.40	85
Pre-trained	100	100	55.30	52.61
Pre-trained	10000	98.9	57.23	57
Using Embedding Layer	17000	95.81	85.16	86

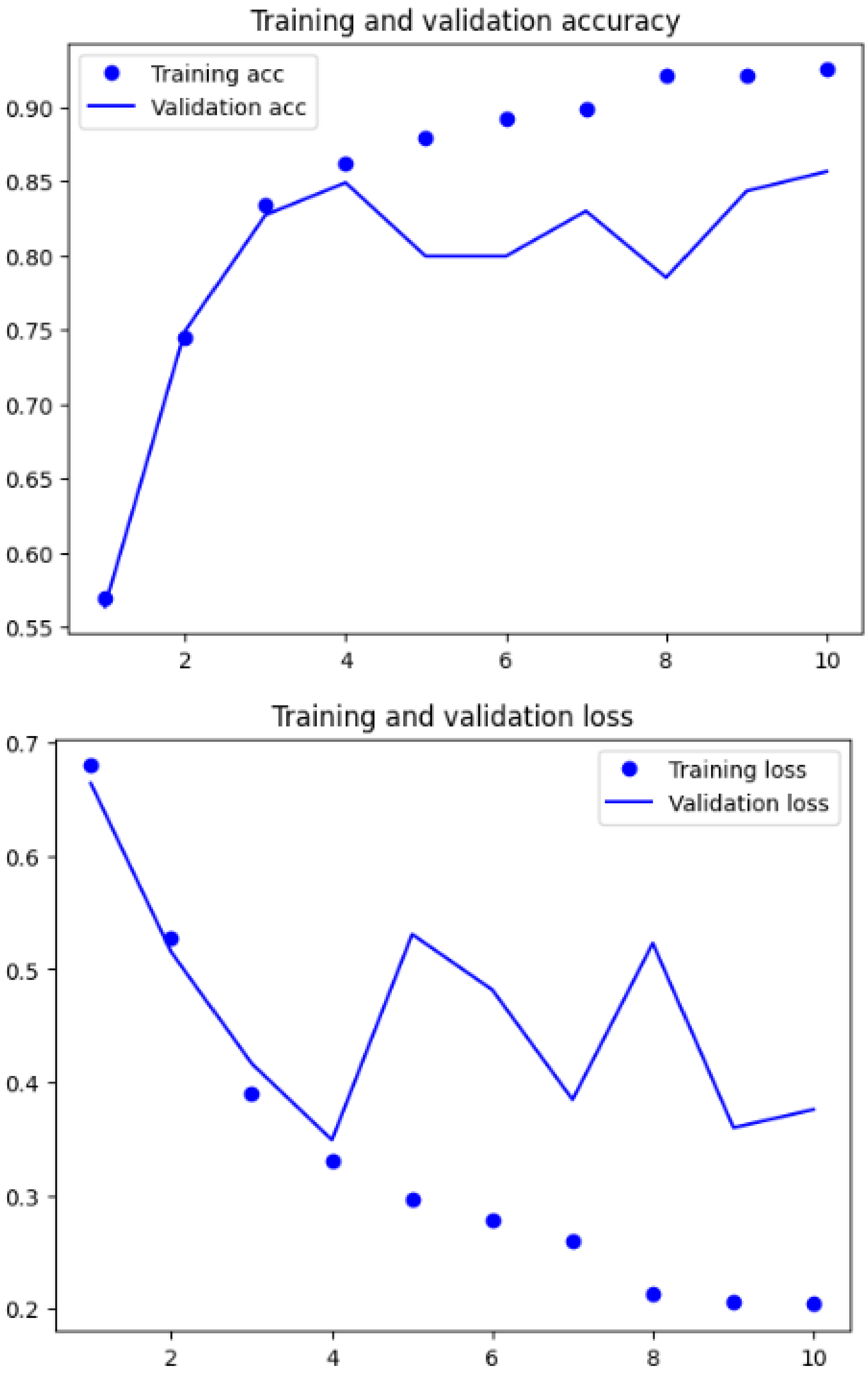

Fig1. Accuracy and loss for embedding layer of 10000 training samples

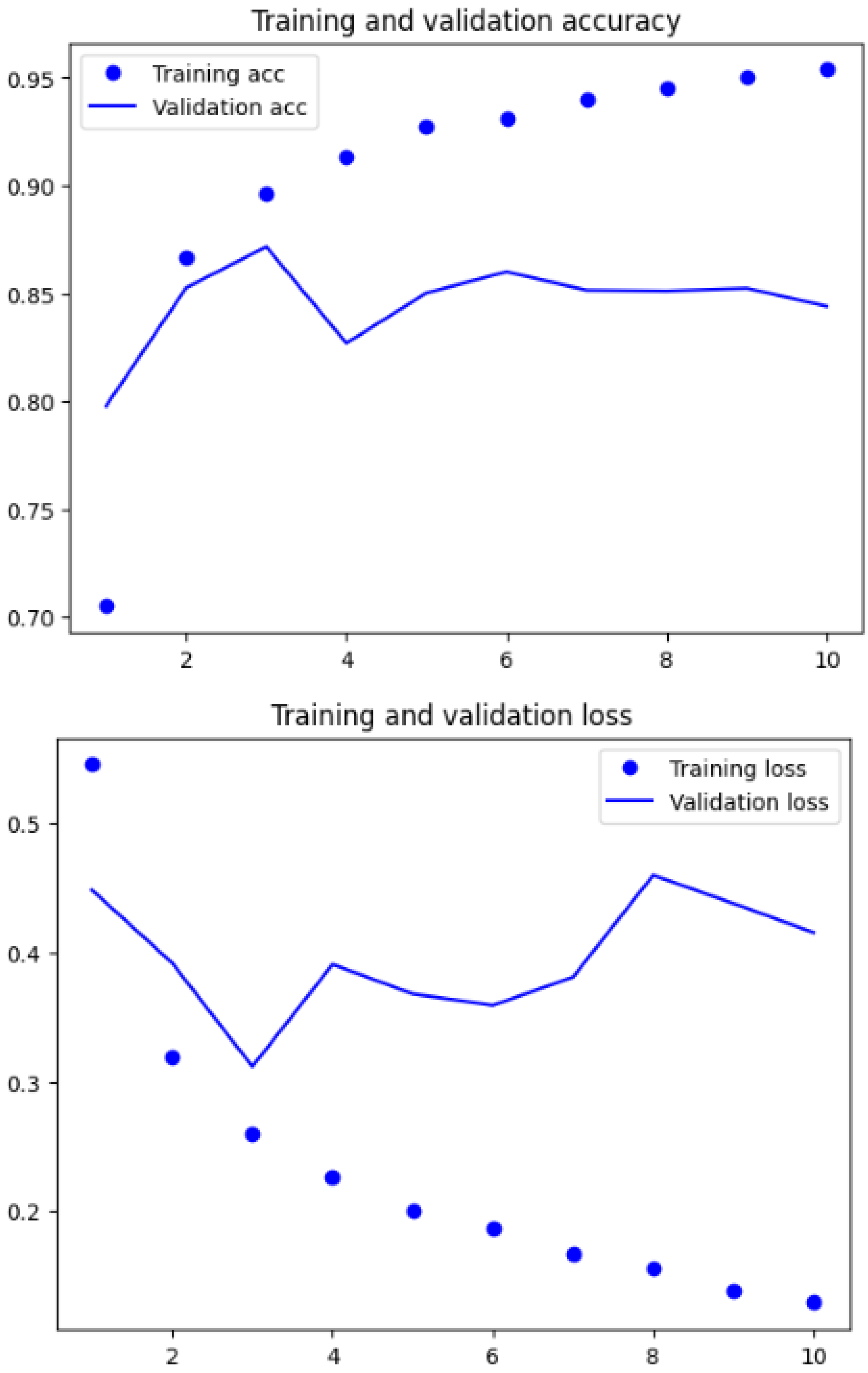

Fig2. Accuracy and loss for embedding layer of 25000 training samples

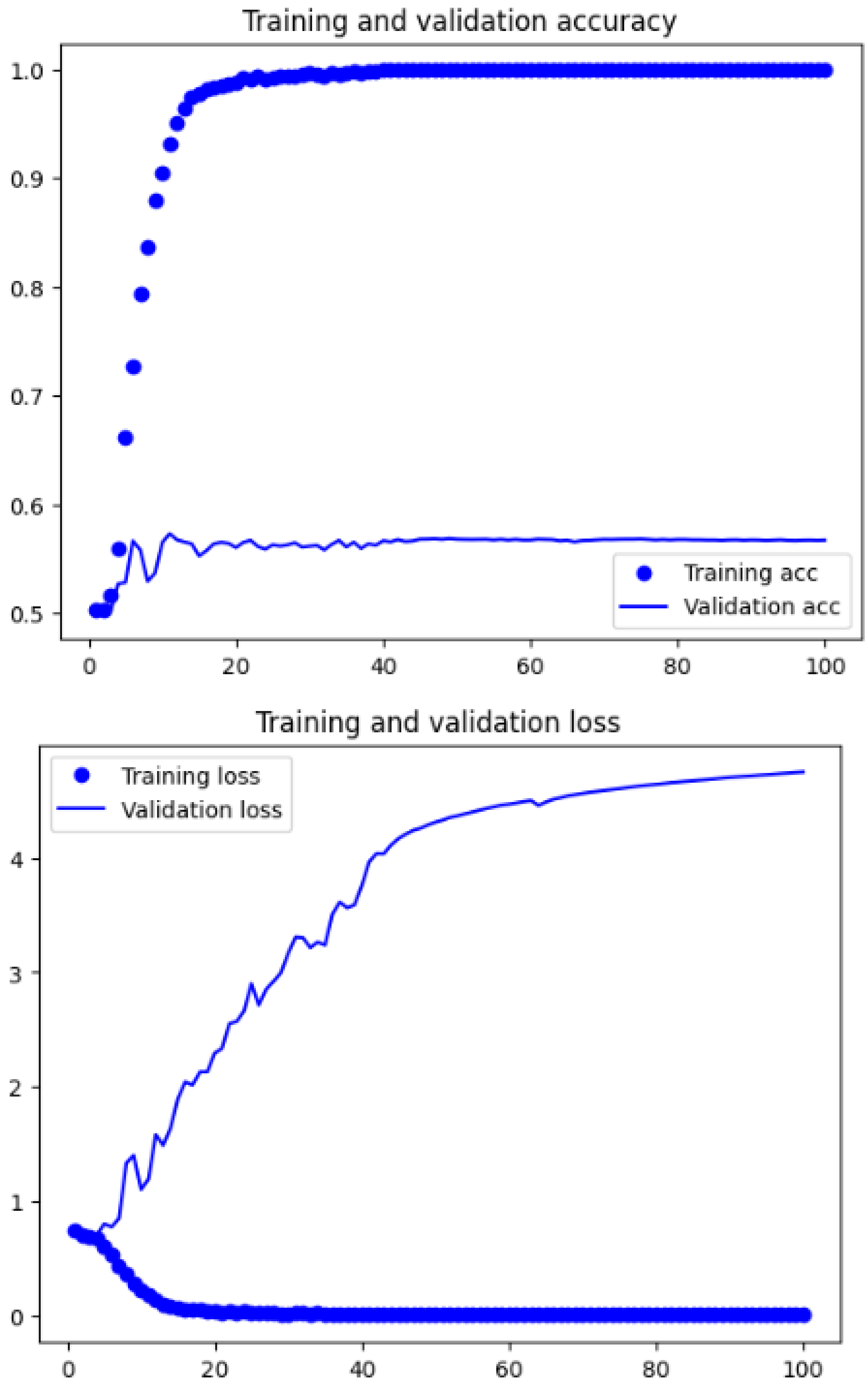

Fig3. Accuracy and loss for pretrained word embedding of 10000 training samples

Conclusion:

The findings emphasize how the size of training data and the strategies used for embedding are crucial in obtaining the best performance for text classification tasks. Utilizing a trainable embedding layer consistently showed better performance than using pretrained embeddings, especially with larger training sets. Using a small dataset of 100 samples, pretrained embeddings showed a slight increase in validation accuracy compared to trainable embeddings. However, they did not generalize effectively when tested. With the increase in training size, trainable embeddings showed a notable improvement in accuracy, reaching a peak test accuracy of 86% when 17,000 samples were used. This shows that trainable embeddings are more appropriate for tasks with ample labeled data, enabling the model to acquire domain-specific representations. Pretrained embedding is beneficial in situations with limited data, but they need fine-tuning or larger datasets to achieve competitive performance.