Text and Sequence

Assignment 4

Report

Introduction:

This assignment focuses on applying Recurrent Neural Networks (RNNs) and Transformers to analyze text and sequence data, specifically using the IMDB movie review dataset. The main objectives include evaluating model performance on text data, exploring ways to improve performance on sparse datasets, and identifying the best approaches to enhance predictive accuracy.

Data Preparation:

The IMDB dataset underwent preprocessing steps, which included:

- Restricting the training dataset to 100 samples.
- Limiting each review to the first 150 words.
- Validating on a set of 10,000 samples.
- Considering only the top 10,000 words from the vocabulary.

Methodology:

1. **Baseline Model:**
    - *Description:* The baseline model used was an RNN with an embedding layer.
    - *Architecture and Hyperparameters:* Text data was processed sequentially with an RNN, starting with an embedding layer to convert words into dense vectors. The model included specific settings for the number of RNN units and embedding dimensions.
    - *Results:* The baseline model's performance was evaluated by recording validation and test accuracy/loss.
2. **The model with Pre-trained Word Embeddings:**
    - *Description:* Pretrained word embeddings, such as GloVe, were integrated to provide the model with semantic word representations.
    - *Integration Process:* These pre-trained embeddings were loaded into the model to enhance its ability to capture textual features.
    - *Results:* The model's performance with pre-trained embeddings was measured using validation and test accuracy/loss.
3. **Varying Training Set Size:**

- *Description:* The training set size was adjusted by changing the number of samples used in training.
- *Results:* Validation and test accuracy/loss were tracked for different training set sizes to observe the effect of data quantity on model performance.

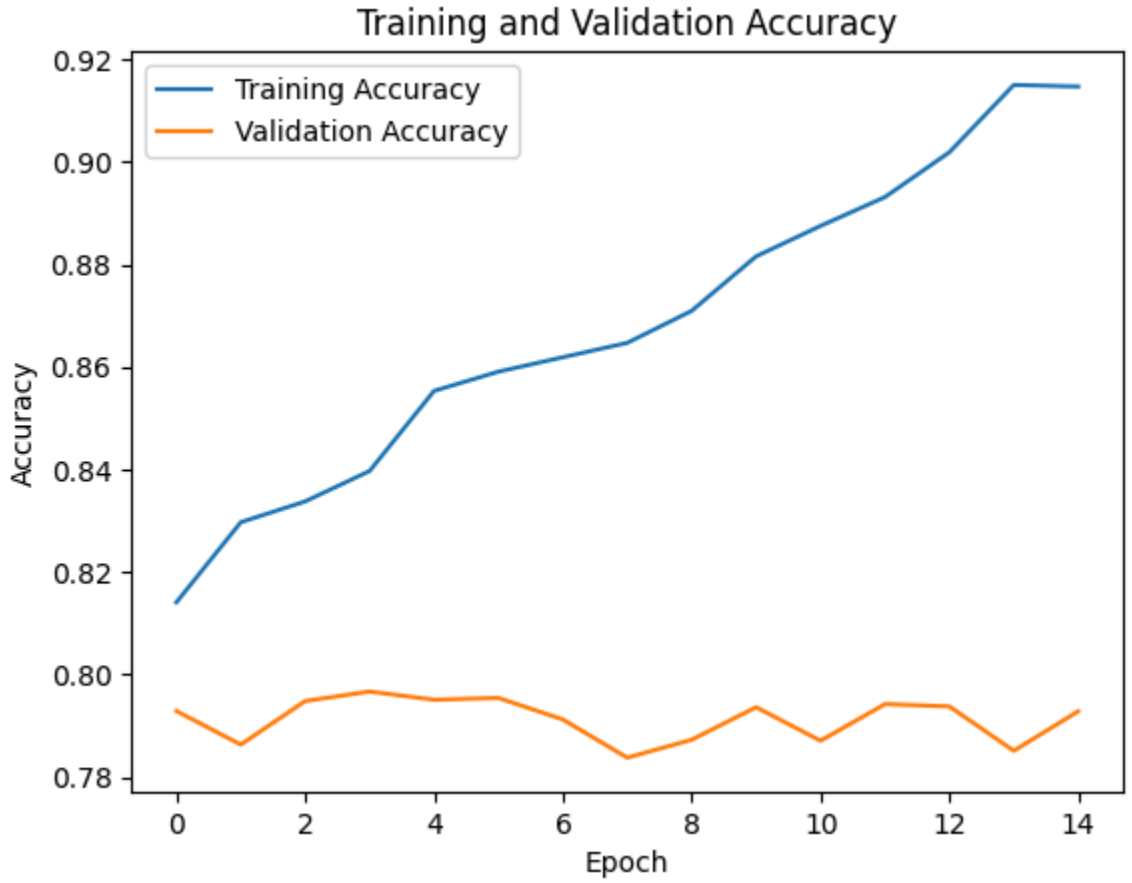
Performance Comparison:

An analysis was conducted to compare the effectiveness of models using the embedding layer against those with pre-trained embeddings across varying training set sizes. This comparison aimed to reveal how different data volumes influence the performance of these embedding strategies.

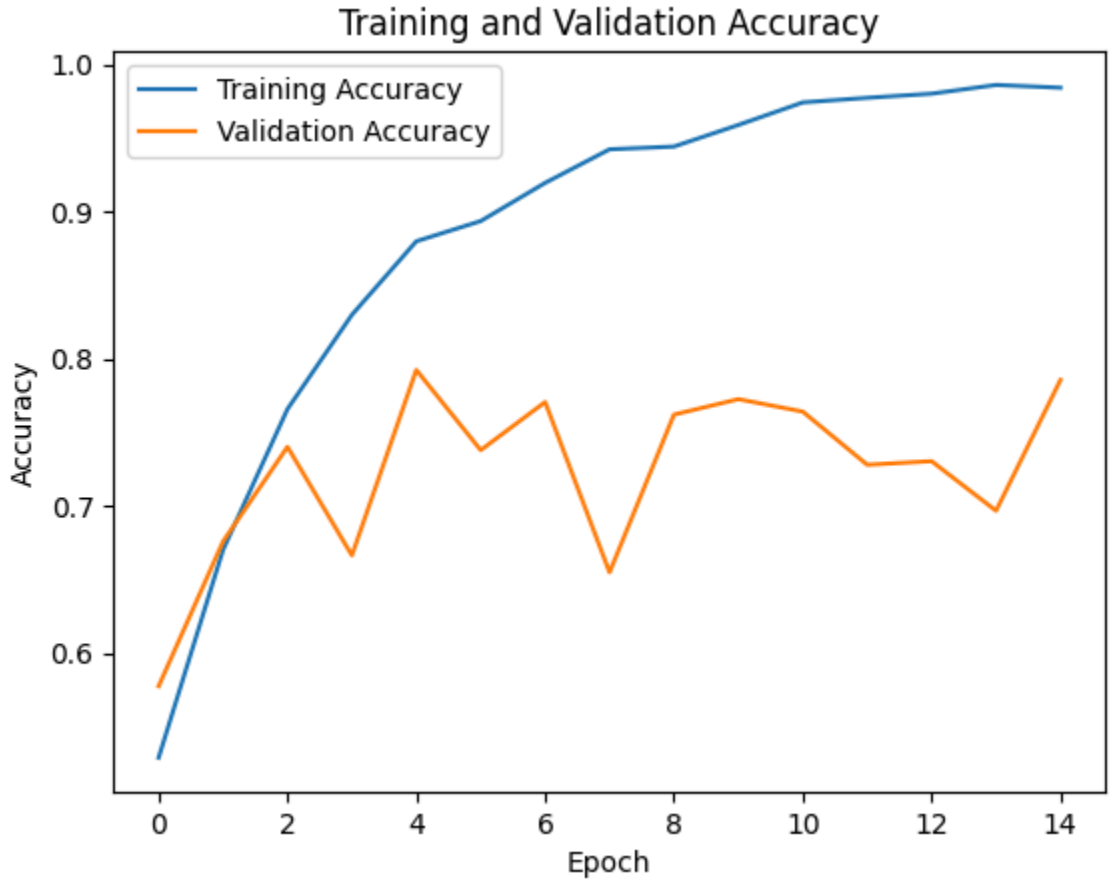
Results:

One Hot Model:

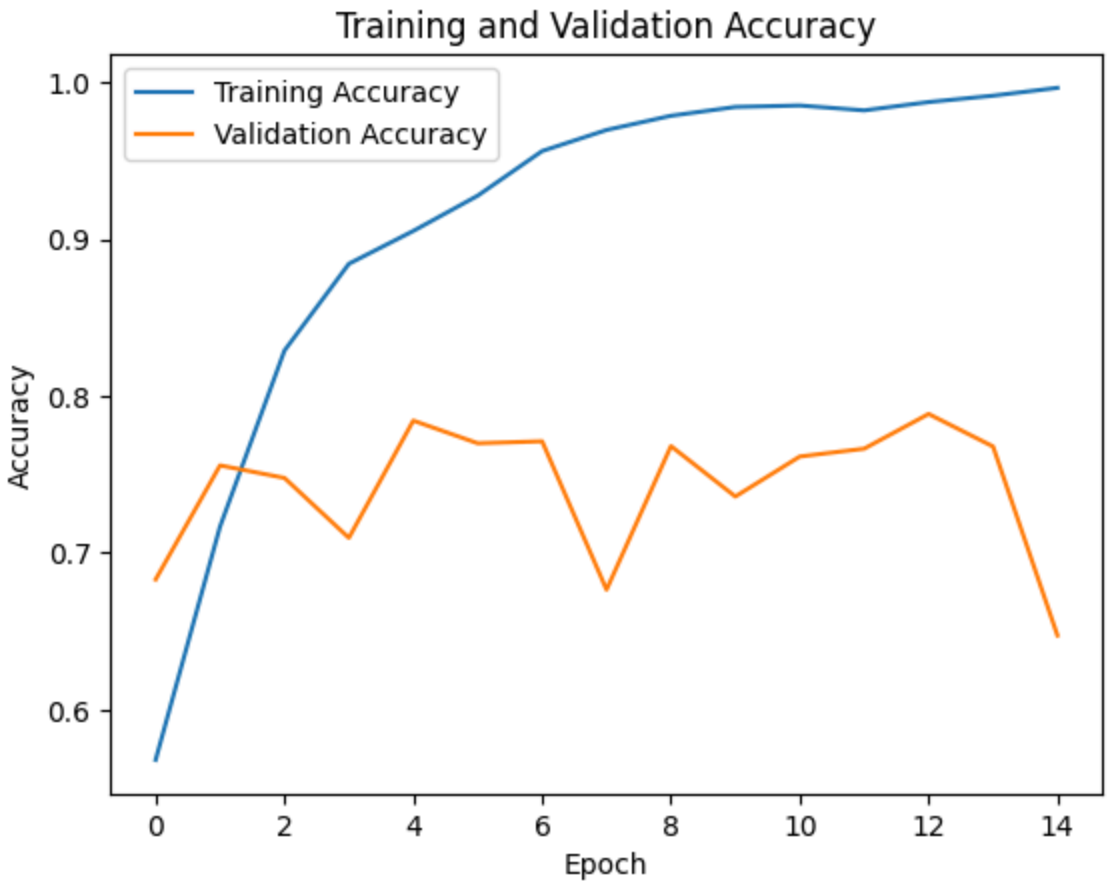
The One-Hot model attained an accuracy of 0.78 with a loss value 0.44.



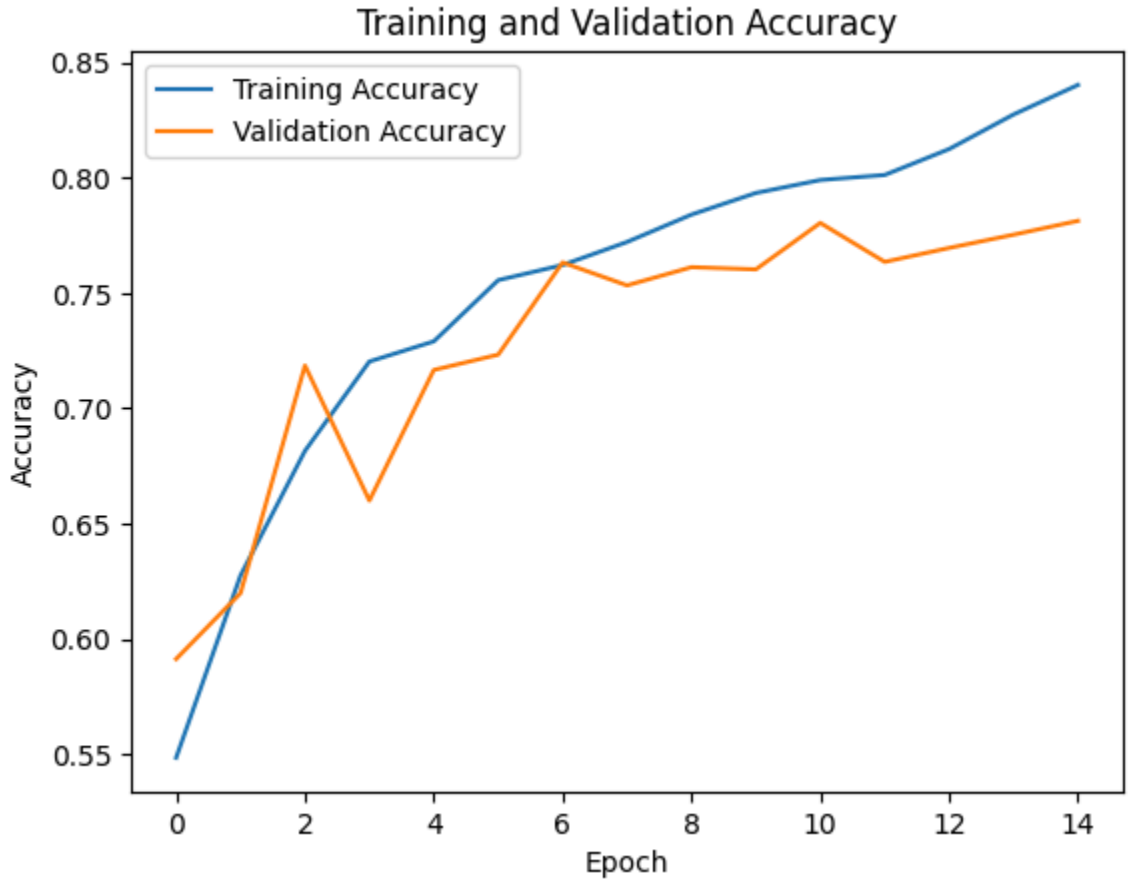
Trainable Embedding Layer:
The model with a trainable embedding layer achieved an accuracy of 0.78 and a loss of 0.49.



Masking Padded Sequences in the Embedding Layer:
Applying masking to padded sequences in the embedding layer resulted in a validation accuracy of 0.77 and a loss of 0.51.

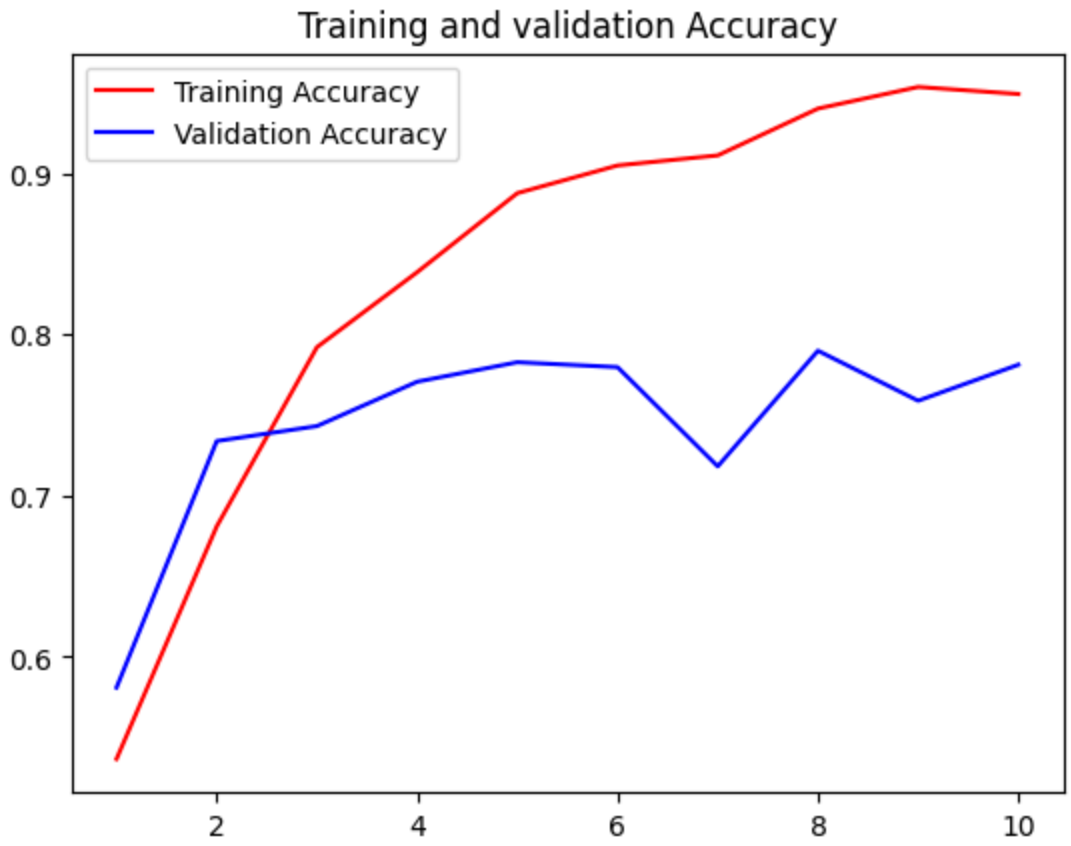

Training and Validation Accuracy
Training Accuracy
Validation Accuracy
Accuracy
Epoch
1.0
0.9
0.8
0.7
0.6
0
2
4
6
8
10
12
14

Model with pretrained GloVe Embeddings:
The model using pre-trained GloVe embeddings achieved an accuracy of 0.76 with a loss of 0.48.

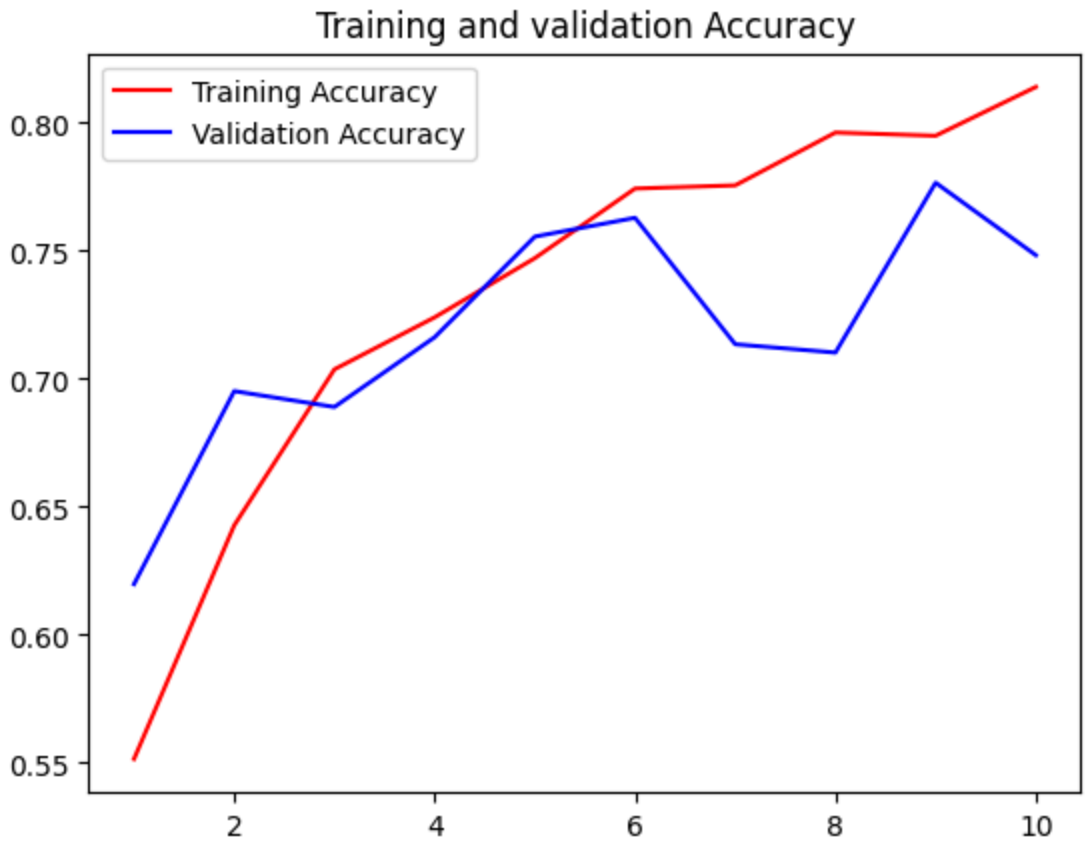


Comparing Model Performance with Different Training Set Sizes

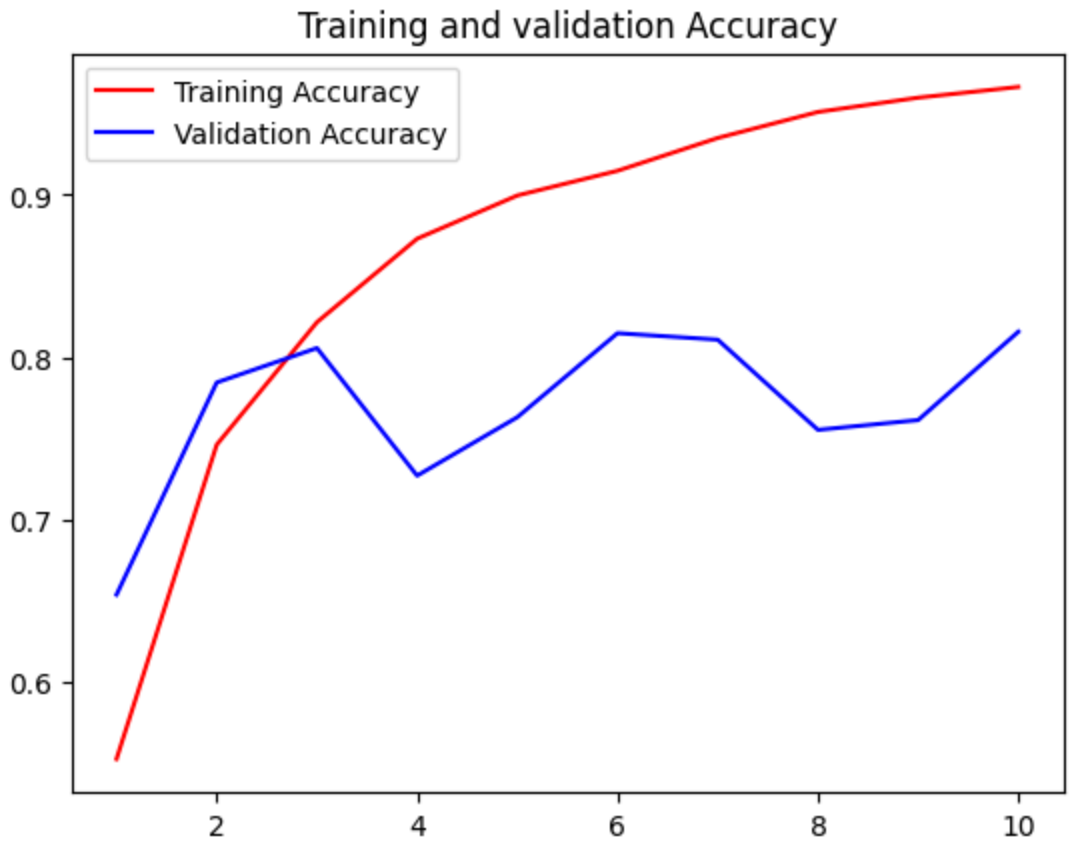
Embedding Layer with 100 Training Samples:
An embedding layer trained on 100 samples achieved an accuracy of 0.76 and a loss of 0.50.



Pretrained Embedding Layer with 100 Training Samples:
Using a pre-trained embedding layer with 100 training samples resulted in an accuracy of 0.77 and a loss of 0.47.

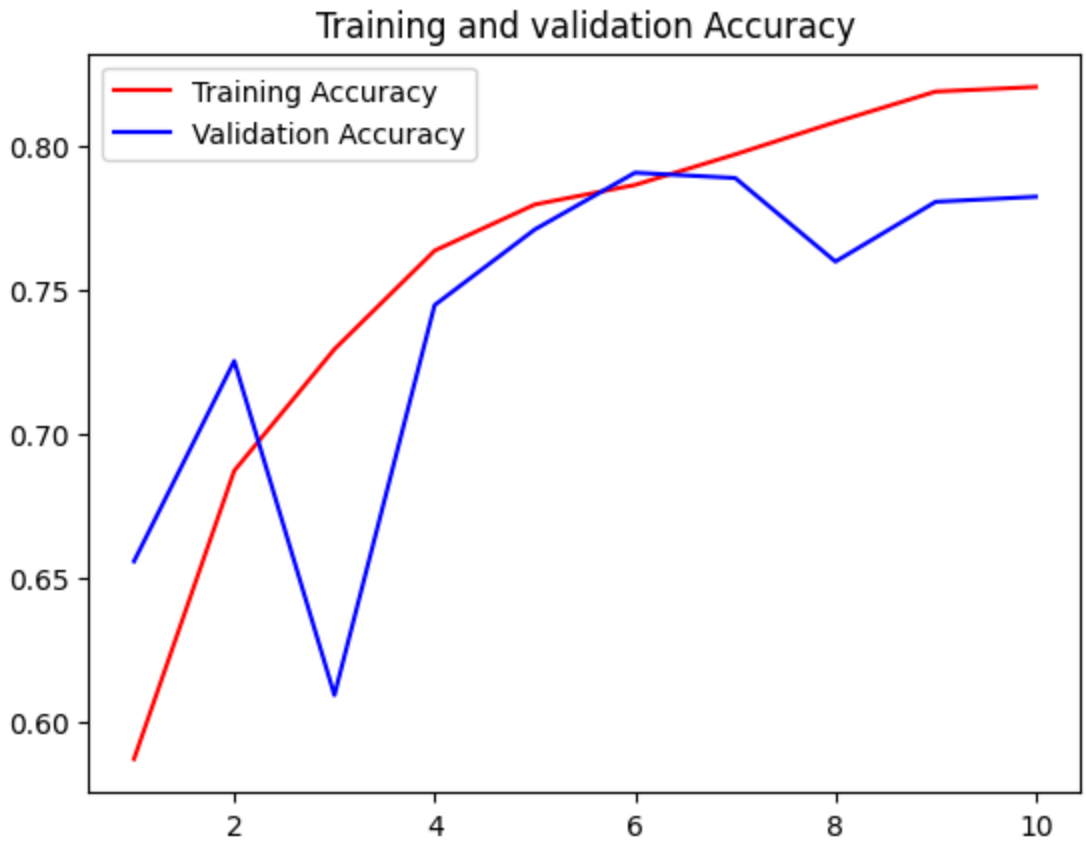


Embedding Layer with 500 Training Samples:
An embedding layer trained on 500 samples achieved an accuracy of 0.80 and a loss of 0.44.



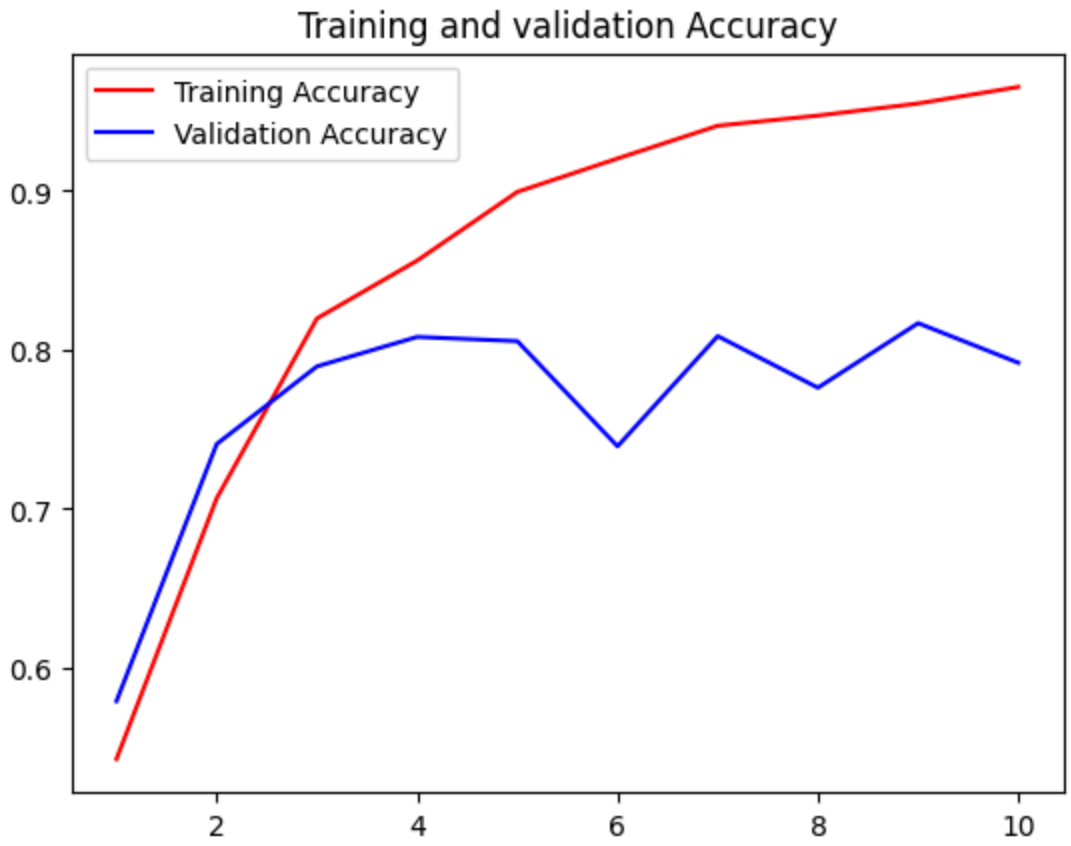
Pretrained Embedding Layer with 500 Training Samples:

With 500 training samples, the pre-trained embedding layer achieved an accuracy of 0.78 and a loss of 0.45.



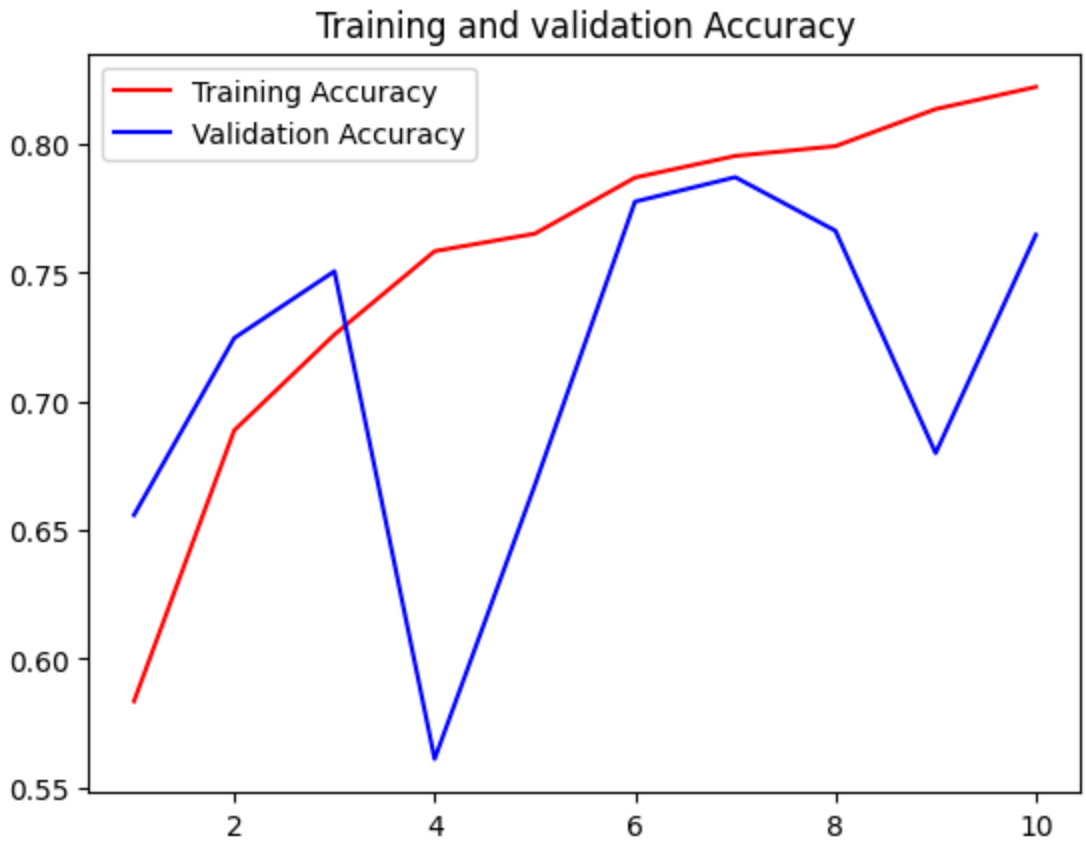
Embedding Layer with 1000 Training Samples:

An embedding layer trained on 1000 samples reached an accuracy of 0.80 and a loss of 0.43.



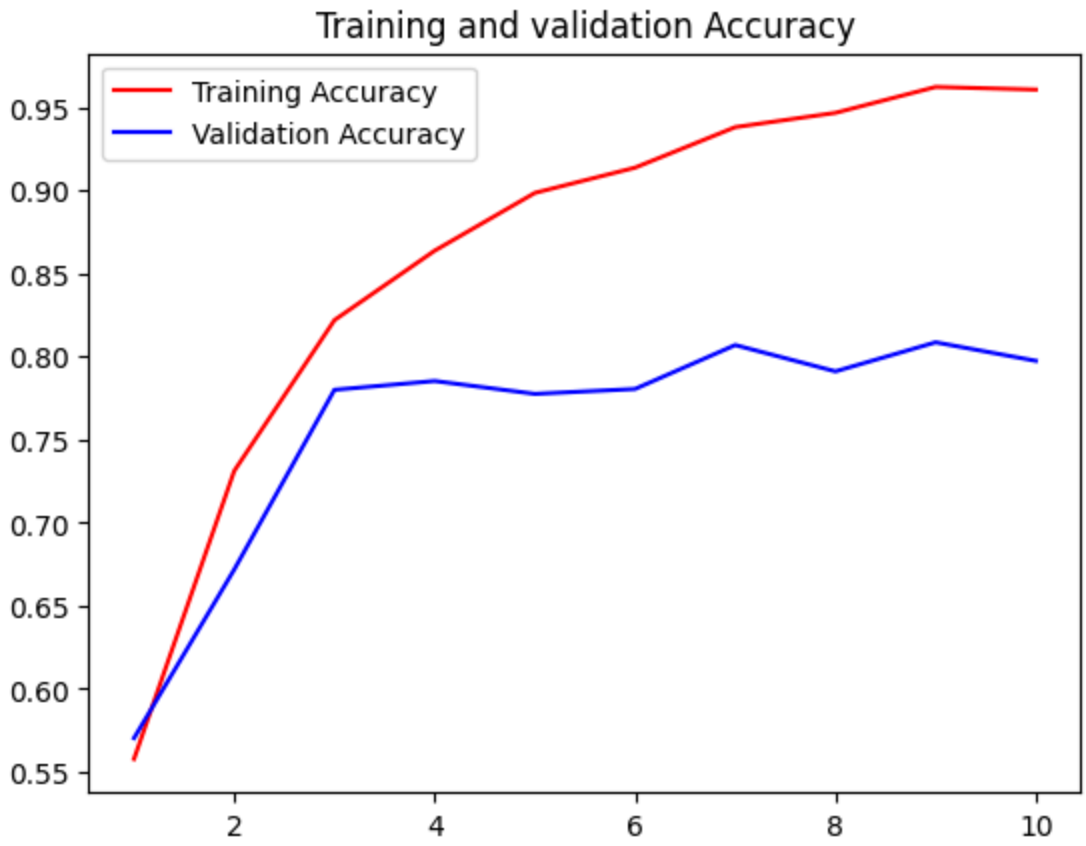
Pretrained Embedding Layer 1000 Training Samples:

A Pretrained Embedding Layer with 1000 training samples results in an accuracy of 0.78 and a loss value of 0.46.



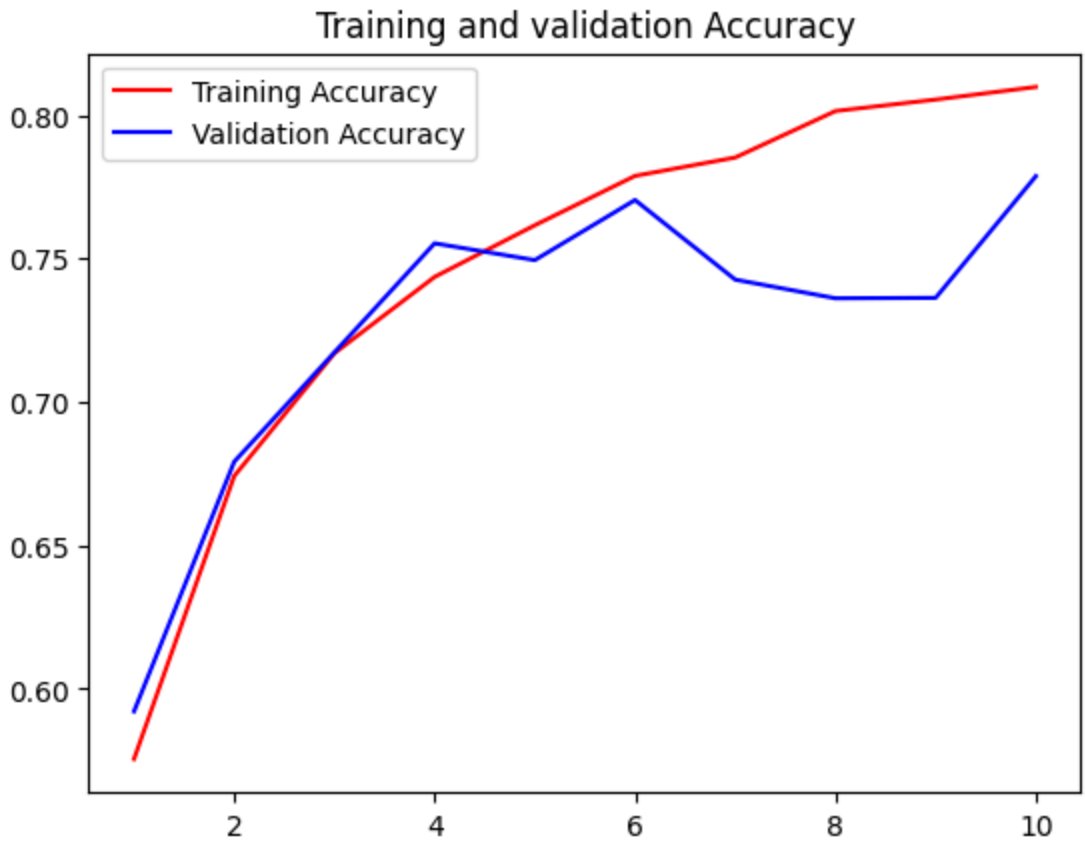
Embedding Layer 5000 Training Samples:

An Embedding Layer with 5000 training samples results in an accuracy of 0.77 and a loss of 0.47.



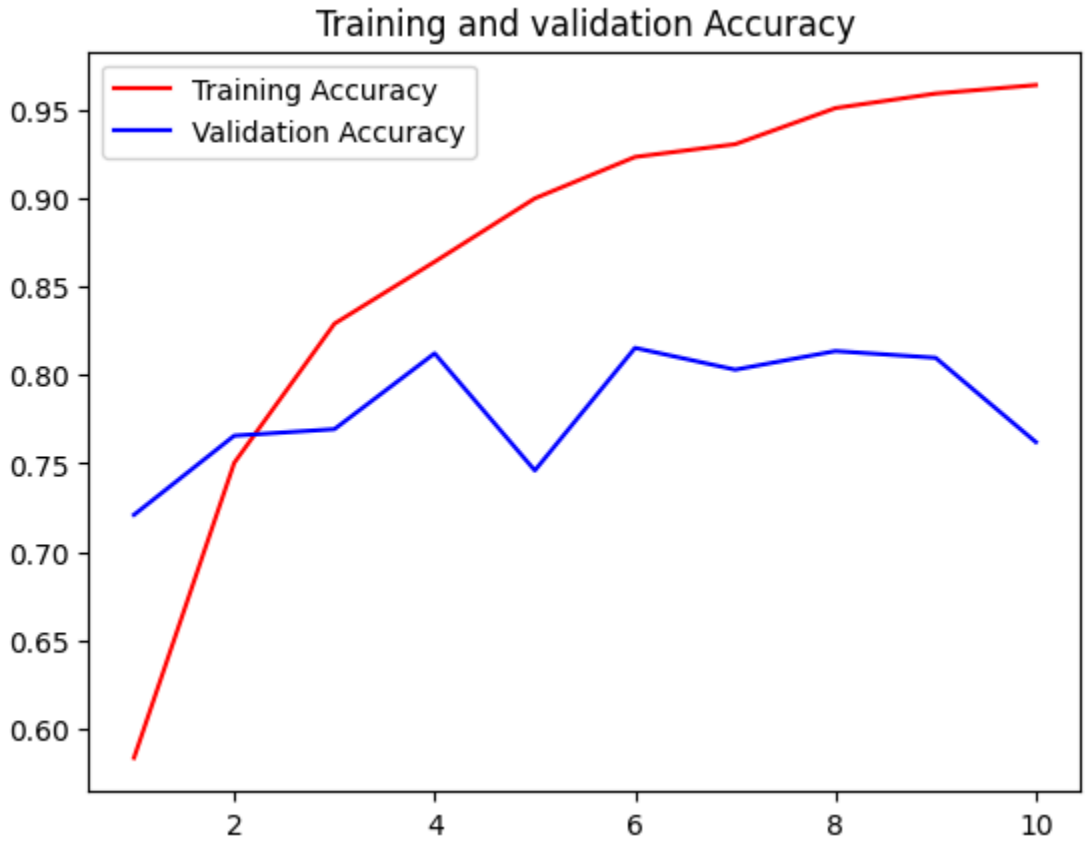
Pretrained Embedding Layer 5000 Training Samples:

A pretrained Embedding Layer with 5000 training samples results in an accuracy of 0.77 and a loss of 0.48.


Training and validation Accuracy
Training Accuracy
Validation Accuracy
0.80
0.75
0.70
0.65
0.60
2
4
6
8
10

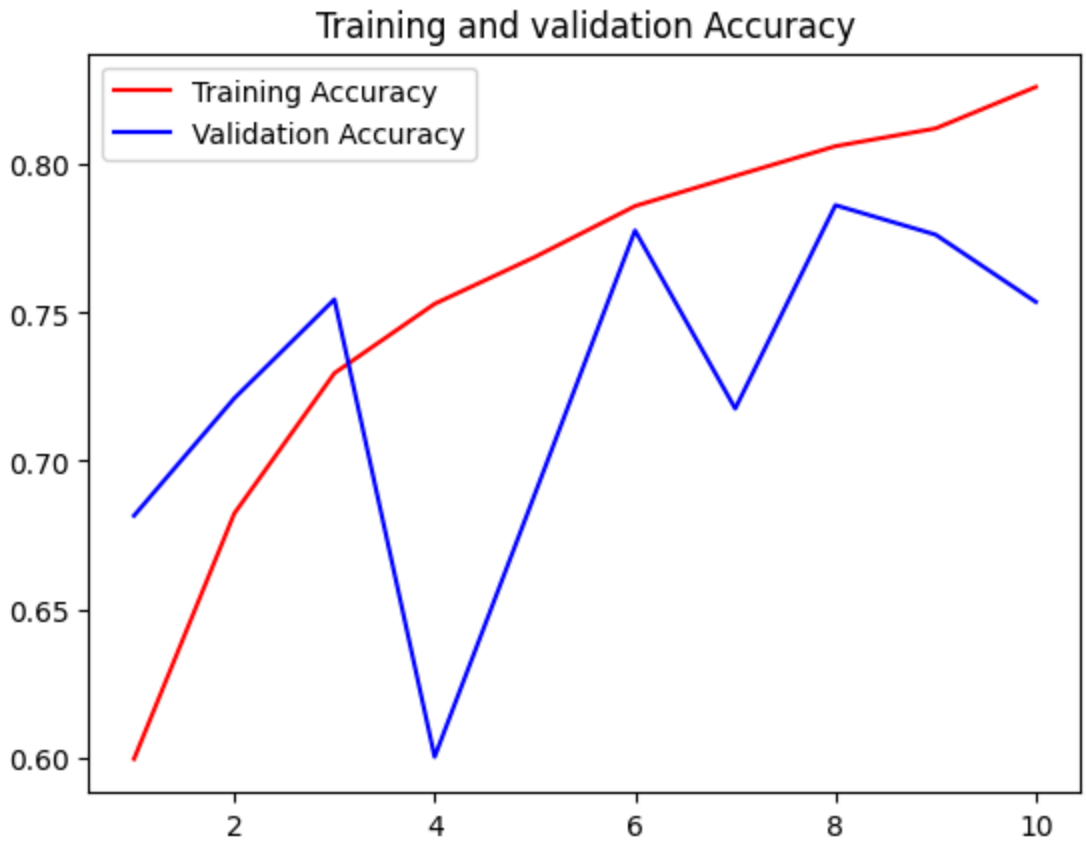
Embedding Layer 10000 Training Samples:

An Embedding Layer with 10,000 training samples achieves an accuracy of 0.80 and a loss of 0.46.



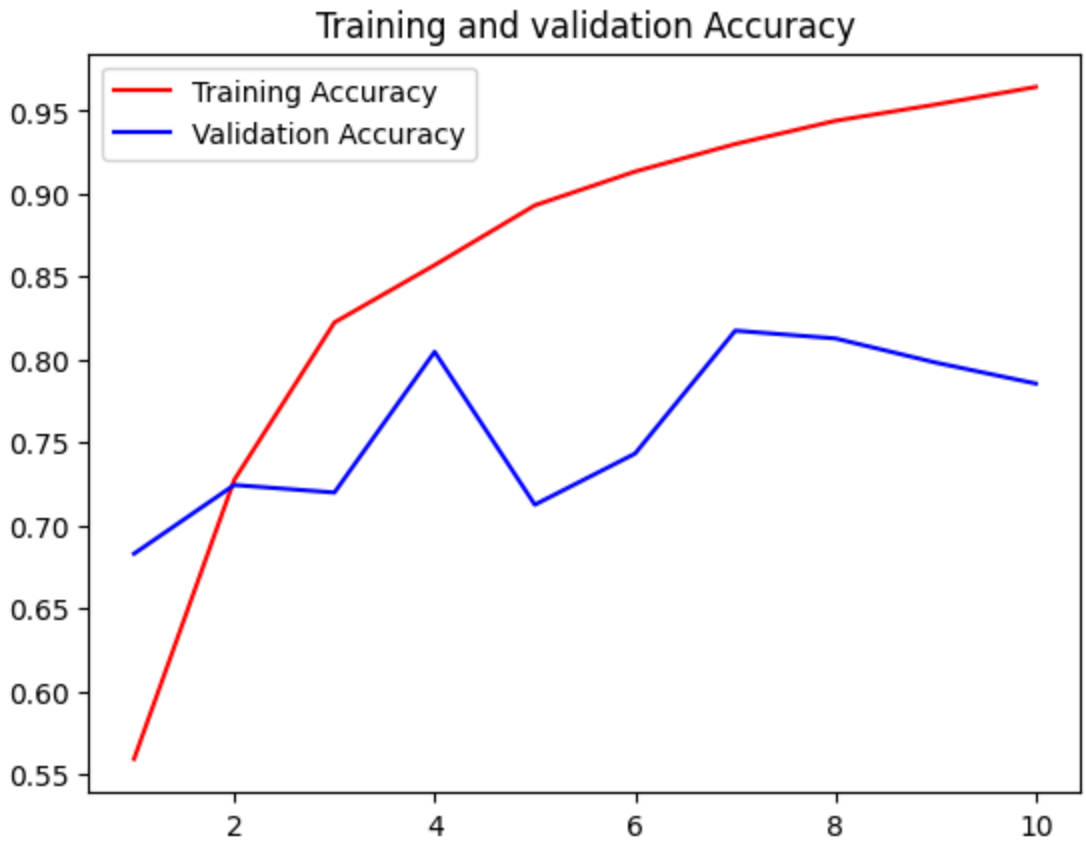
Pretrained Embedding Layer 10000 Training Samples:

A Pretrained Embedding Layer with 10,000 training samples results in an accuracy of 0.78 and a loss of 0.46.



Embedding Layer 20000 Training Samples:

An Embedding Layer with 20000 Training Samples achieves an Accuracy of 0.80 and a loss of 0.44.



Pretrained Embedding Layer 20000 Training Samples:

A Pretrained Embedding Layer with 20000 Training Samples achieves an Accuracy of 0.78 and a loss of 0.45.

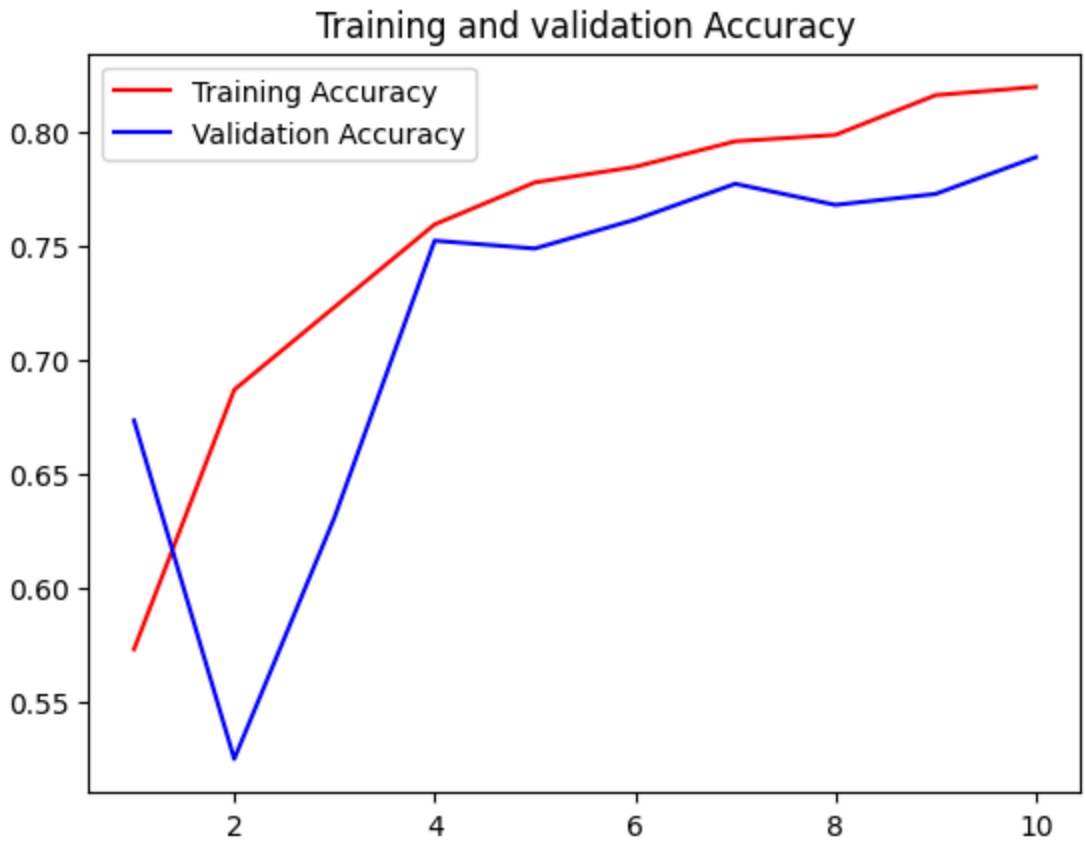


Table Results:

Model	**Accuracy**	**Loss**
One Hot model	0.78	0.44
Trainable Embedding Layer	0.78	0.49
Masking Padded Sequences in the Embedding Layer	0.77	0.51
Model with Pretrained GloVe Embeddings	0.76	0.48
Embedding Layer of 100 Training Samples	0.76	0.50
Pretrained Embedding Layer of 100 Training Samples	0.77	0.47

Embedding Layer of 500 Training Samples	0.80	0.44
Pretrained Embedding Layer of 500 Training Samples	0.78	0.45
Embedding Layer of 1000 Training Samples	0.80	0.43
Pretrained Embedding Layer of 1000 Training Samples	0.78	0.46
Embedding Layer of 5000 Training Samples	0.777	0.47
Pretrained Embedding Layer of 5000 Training Samples	0.775	0.48
Embedding Layer of 10000 Training Samples	0.80	0.46
Pretrained Embedding Layer of 10000 Training Samples	0.78	0.46
Embedding Layer of 20000 Training Samples	0.80	0.44
Pretrained Embedding Layer of 20000 Training Samples	0.78	0.45

Conclusion:

The results underscore the benefits of using pre-trained word embeddings, particularly in scenarios with limited training data. It can be concluded that pre-trained embeddings, like GloVe, provide a significant advantage when dealing with smaller datasets. Models that incorporated pre-trained embeddings consistently outperformed those using trainable embeddings in cases where the training set was between 100 and 500 samples. Pretrained embeddings serve as a strong foundation for word representations, resulting in higher validation accuracy and lower loss compared to models trained from scratch.

However, as the training dataset grew to 1,000 samples or more, the performance gap between pre-trained embeddings and trainable layers narrowed. Both approaches yielded similar validation accuracy and loss, indicating that with sufficient data, models can learn meaningful

word representations without relying on pre-trained embeddings. Trainable layers were able to capture the specific nuances of the dataset when provided with enough data.

Notably, the pre-trained embedding model with 20,000 training samples showed the best overall performance, with a validation accuracy of 0.79 and a loss of 0.45. Yet, the performance differences among models with larger datasets were relatively small, suggesting that after a certain point, increasing the dataset size on performance becomes less significant.

In summary, the choice between pre-trained and trainable embeddings should be guided by the amount of available training data. For smaller datasets, pre-trained embeddings are highly recommended for better results. As the dataset grows, the advantages of pre-trained embeddings lessen, and trainable embedding layers become a more viable option. Ultimately, the decision should be based on the task, available computational resources, and the desired balance between model complexity and performance.