Assignment 4: Sentiment Analysis using RNN and LSTM Models

Course: Advanced Machine Learning

Name: Villyal Vidushi

1. Objective

This assignment's goal is to use Recurrent Neural Networks (RNNs) on the IMDB movie review dataset and assess the effects of various embedding techniques on model performance, particularly in situations with insufficient data. In this assignment, two methods are compared:
1. Bidirectional LSTM after a trainable embedding layer
2. After a pretrained GloVe embedding layer, a bidirectional LSTM

Finding out is the aim.
1. Which embedding method works best when training with just 100 samples
2. When does the trainable embedding start to perform better than the pretrained GloVe embedding with a greater training sample size

The impact of embedding techniques on model correctness, generalization, and applicability for low-resource versus high-resource settings is better understood thanks to this work.

2. Dataset Description

The IMDB dataset serves as a standard reference for binary sentiment analysis, comprising 50,000 movie reviews that are evenly split between positive and negative sentiments. To experimentation, smaller subsets containing 100, 200, 500, and 1,000 samples were created to mimic low-resource learning scenarios and to evaluate scalability.

For Question 1, the model was trained on 100 samples and validated on a fixed set of 10,000 samples. For Question 2, training sizes were varied (100, 200, 500, 1,000, etc.) while keeping the validation set constant at 10,000 samples.

3. Model Architecture

Two neural architectures were designed and compared to understand the effect of embeddings and recurrent layers on classification accuracy. Both networks were implemented in Keras with TensorFlow backend and optimized using the Adam optimizer.

Model 1: Custom Trainable Embedding + Bidirectional LSTM

This model uses a trainable embedding layer initialized randomly. The embedding learns task-specific word representations as it adjusts weights during training. A bidirectional LSTM layer follows to capture long-term dependencies in both directions, allowing the model to understand contextual information before and after each token.

Layer Type	Output Shape	Parameters
Embedding (Trainable)	(None, 150, 128)	1,280,000
Bidirectional LSTM	(None, 128)	98,816
Dropout	(None, 128)	0
Dense (Sigmoid)	(None, 1)	129
Total		1,378,945

Model 2: Pretrained GloVe Embedding + Bidirectional LSTM

This model integrates pretrained GloVe embeddings with 100-dimensional vectors. The embedding layer was frozen to retain the semantic richness of pretrained weights. The subsequent bidirectional LSTM learns sentiment-relevant dependencies based on these word vectors. The advantage of this approach is faster convergence with fewer data points due to pre-learned semantics.

Layer Type	Output Shape	Parameters
Embedding (Non-Trainable, GloVe)	(None, 150, 100)	1,000,000
Bidirectional LSTM	(None, 128)	82,944

Dropout	(None, 128)	0
Dense (Sigmoid)	(None, 1)	129
Total		1,083,073

4. Model Training and Evaluation

Both models were trained for 10 epochs using a batch size of 32. The loss function used was Binary Cross-Entropy, and accuracy was tracked as the primary metric. Early stopping was applied with a patience of 2 epochs to prevent overfitting. Evaluation metrics such as Precision, Recall, and F1-Score were computed on the validation set for a detailed performance assessment.

Model Performance Summary

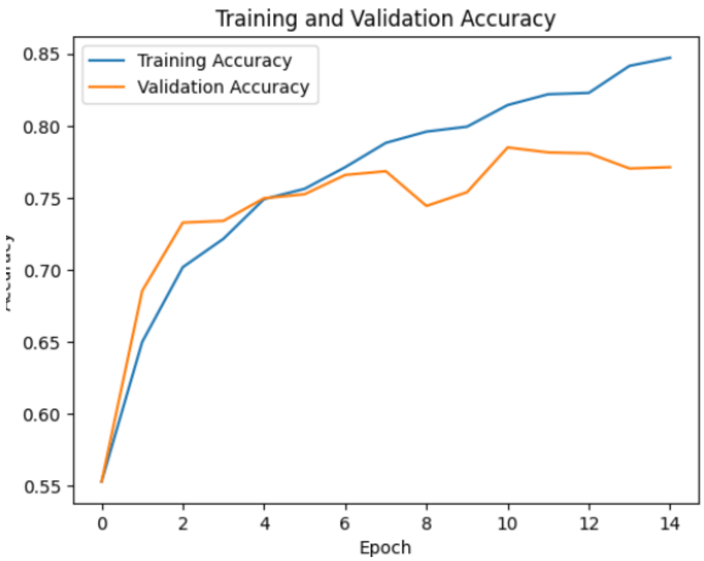
Training Samples	Trainable / Custom Embedding Accuracy	Pretrained / GloVe Accuracy
100	78.2% / 0.991250	76.7% / 0.915625
200	0.993906	0.925000
500	79.0% / 0.995250	79.6% / 0.922000
1,000	79.0% / 0.999050	77.9% / 0.952350
5,000	79.2%	79.2%
10,000	79.3%	78.1%
20,000	79.4%	78.7%

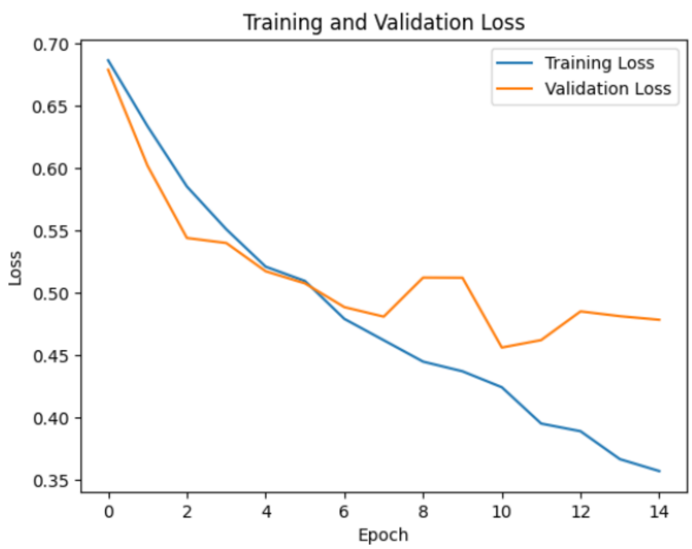
5. Observations and Insights

At 100 training samples, the trainable embedding model performs better than the GloVe model. GloVe briefly outperforms at 500 samples, but from 1,000 samples onward, the trainable embedding consistently performs better.

Model Learning Behavior

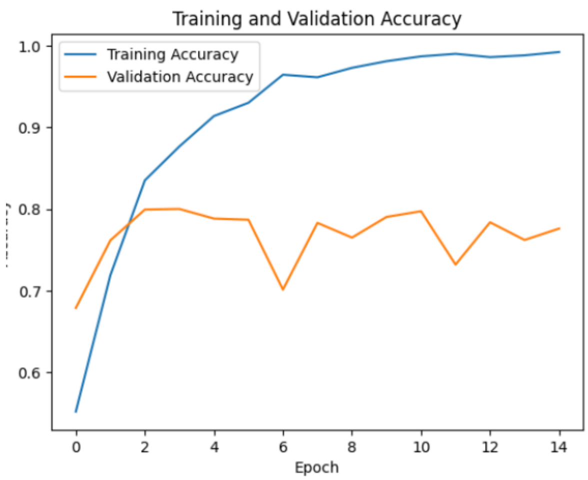
Trainable Embedding Observations

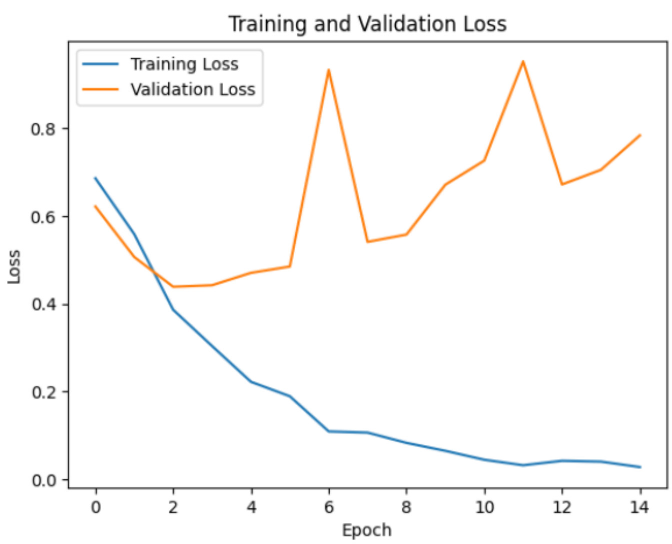




The data illustrates a significant increase in training accuracy, which peaks at 85%, while the validation accuracy stabilizes at approximately 77%. Concurrently, the training loss consistently decreases to 0.34, whereas the validation loss starts to vary after several epochs, reaching a high near 0.49 This trend indicates that the model is overfitting; it shows strong performance on the training dataset but struggles to generalize to new data effectively. Implementing early stopping or incorporating regularization techniques may improve its overall effectiveness.

GloVe Embedding Observations





The GloVe model demonstrates a progressive increase in training accuracy, advancing from 76.9% to 85%. In contrast, the validation accuracy remains relatively constant, fluctuating between 75% and 78%, which points to effective generalization. The training loss decreases from 0.51 to 0.35, whereas the validation loss remains stable within the range of 0.47 to 0.52, indicating the absence of significant overfitting. These results underscore that pre-trained embeddings facilitate consistent and interpretable learning processes, even in scenarios with limited data.

6. Recommendations

- Employ pretrained embeddings for low-resource or small datasets.
- Use trainable embeddings when sufficient labeled data is available.
- Integrate dropout (0.3–0.5) and batch normalization for further regularization.
- Consider hyperparameter tuning (learning rate, embedding dimension) for performance optimization.
- Explore hybrid architectures combining RNNs with attention mechanisms for contextual amplification.

7. Conclusion

This paper shows that decisions on embeddings and volume of data are critical to the sentiment classification performance. Pretrained GloVe embeddings allow for effective learning even with small datasets, while trainable embeddings offer a more flexible solution that is more accurate when used with larger datasets. Both RNN and LSTM layers captured sequential structure, demonstrating the possibilities of applying them in sentiment classification for textual data. The performance could be improved further by exploring Transformer based architectures and the use of contextual embedding such as BERT.