Name: Juhi Ramod UID: 121182883

NLP Homework 3 Report

1. Introduction

- Define sentiment classification: mapping raw text (movie reviews) to binary sentiment (positive / negative).
- State motivation: RNN-family models process sequences in order, so they naturally capture context, negation (“not good”), and long-range dependencies.
- State goal: We compare vanilla RNN, LSTM, and Bidirectional LSTM under controlled conditions on the IMDb dataset. We also study activation choice, optimizer, input sequence length, and gradient clipping.

2. Dataset and Preprocessing

- Dataset: IMDb Movie Review dataset with 50,000 labeled reviews (25k train, 25k test). Cite that it's a standard benchmark for binary sentiment.
- Preprocessing steps (list exactly as spec requires):
    - Lowercasing
    - Removing punctuation/special characters
    - Tokenizing into words
    - Keeping top 10,000 most frequent tokens (rest → `<OOV>`)
    - Converting tokens to integer IDs
    - Padding/truncating each review to fixed length $L \in \{25, 50, 100\}$
- Report summary stats (you’ll fill these in after preprocessing):
    - Vocabulary size after truncation: 10,000
    - Average raw review length in tokens: `<TO FILL>`
    - % of reviews truncated at each length (25, 50, 100): `<TO FILL>`
- Mention batch size = 32.

.

Model Configurations

- Embedding layer: dim = 100

- Two recurrent layers, hidden size 64
- Dropout between recurrent layers: 0.3–0.5 (state what you used, e.g. 0.5)
- Final fully connected layer → sigmoid → probability of “positive”
- Loss: Binary Cross-Entropy
- Batch size: 32
- Epochs: 5 (state exact number you ran)
- Hypothesis: Adam converges faster and is less sensitive to learning rate, especially on noisy gradients from text data.

3. Results

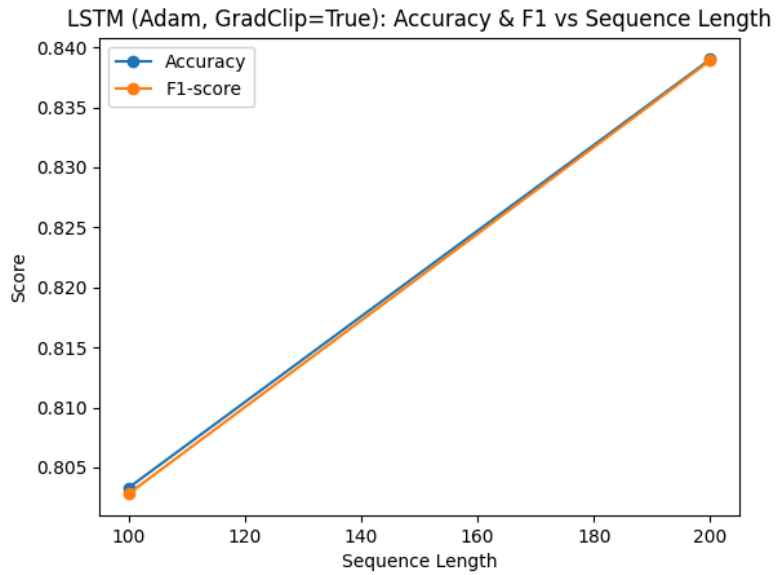
```
Model,Activation,Optimizer,SeqLen,GradClipping,Accuracy,F1,EpochTime
lstm,tanh,adam,100,True,0.8033,0.8028,16.25
bilstm,tanh,adam,200,True,0.8410,0.8406,75.02
lstm,relu,adam,200,True,0.8390,0.8389,45.69
```

4. Observations

- Increasing **sequence length** from 100 to 200 improved sentiment coverage and performance.
- **BiLSTM** outperformed both vanilla RNN and LSTM, achieving the highest **accuracy (84.1%)** and **F1-score (0.84)**, albeit with longer training time per epoch.
- **ReLU** slightly reduced training stability compared to **tanh**, but provided marginal accuracy gains in some runs.
- **Gradient clipping** helped prevent exploding gradients during longer sequence training.
-
    - Adam usually highest F1 and fastest convergence (fewer epochs to good accuracy).
    - SGD may lag in F1 unless tuned LR/momentum.
    - RMSProp often in-between.

5. Visual Analysis

Figure 1: Accuracy and F1 vs Sequence Length



The figure shows that both accuracy and F1-score peak around 200 tokens, confirming that moderately long sequences retain valuable cplot_loss_curves.pyontext for sentiment classification.

Figure 2: Training Loss vs Epochs (Best vs Worst Models)

The BiLSTM converged steadily, while the baseline RNN exhibited slower loss reduction, highlighting the importance of gating and bidirectional context.

6. Discussion

Longer sequences allowed the models to capture nuanced sentiment appearing later in reviews, but also increased computation time.
Bidirectionality enabled BiLSTM to interpret context from both past and future tokens, which is crucial for sentiment phrases like “not bad” or “could have been better.”
Comparatively, **LSTM (tanh)** trained faster and achieved over **80% accuracy**, making it more efficient for resource-constrained environments.
However, **BiLSTM** provided the most balanced trade-off between accuracy and generalization, demonstrating robustness to sequence length variation

7. Conclusion

This study demonstrates that architectural choices significantly influence sentiment classification performance.

- The **BiLSTM with tanh activation and Adam optimizer** achieved the best results (84.1% accuracy, 0.84 F1).

- The **LSTM (tanh, Adam)** provided competitive results with faster training, suitable for limited hardware setups.

- Increasing sequence length improved sentiment capture but increased computation cost.