IMDb Sentiment Analysis with RNN Architectures

Author: Athreya Sudarshan Srikanth
Coursework: NLP DATA641 – Homework 3

Abstract

This project implements and compares multiple Recurrent Neural Network (RNN) architectures for binary sentiment classification on the IMDb movie reviews dataset.
The models—RNN, LSTM, and BiLSTM—are evaluated under varying hyperparameters, including activation functions, optimizers, sequence lengths, and gradient clipping.

The results show that **LSTM with ReLU activation and Adam optimizer** consistently achieves the best performance, with an accuracy and F1-score of approximately **0.82** at a sequence length of **100 tokens**.

Dataset and Preprocessing

The dataset used is the IMDb Dataset of **50,000 labeled movie reviews** (balanced between positive and negative sentiments).
Each review was:

- Lowercased and cleaned of HTML tags and punctuation.
- Tokenized using a TextVectorization layer with a vocabulary size of **10,000** and an `<UNK>` token for out-of-vocabulary words.
- Split 50/50 into training and testing sets (25,000 each).
- Sequences were padded or truncated to fixed lengths of **25**, **50**, or **100 tokens**.

Preprocessed datasets were saved as `.npz` files (`imdb_25.npz`, `imdb_50.npz`, `imdb_100.npz`) for reproducibility.

Model Design

Three RNN variants were implemented in **PyTorch**:

- **RNN:** Basic recurrent layer with ReLU or Tanh activation.
- **LSTM:** Long Short-Term Memory with 2 layers and hidden size 64.
- **BiLSTM:** Bidirectional LSTM with the same hidden size.

Common settings:

- Embedding dimension: 100
- Dropout: 0.4
- Batch size: 32
- Loss: `BCEWithLogitsLoss`
- Optimizers tested: Adam, SGD, RMSProp
- Gradient clipping: enabled/disabled (max norm = 1.0)

All experiments were run on CPU with fixed random seeds for full reproducibility.

Experiments and Results

Each experiment trained for **5 epochs** while varying one factor at a time.
Below are summarized results from `results/metrics.csv`:

Architecture	Activation	Optimizer	Seq Length	Clip	Accuracy	F1	Time/Epoch (s)
RNN	ReLU	Adam	50	no	0.588	0.587	13.7
LSTM	ReLU	Adam	50	no	0.771	0.771	12.0
BiLSTM	ReLU	Adam	50	no	0.763	0.763	23.4
LSTM	Sigmoid	Adam	50	no	0.763	0.763	12.3
LSTM	Tanh	Adam	50	no	0.766	0.766	12.1
LSTM	ReLU	SGD	50	no	0.522	0.517	10.0
LSTM	ReLU	RMSProp	50	no	0.763	0.763	11.1
LSTM	ReLU	Adam	25	no	0.721	0.721	8.4
LSTM	ReLU	Adam	100	no	0.821	0.821	19.0
LSTM	ReLU	Adam	50	yes	0.769	0.769	12.1

Analysis and Discussion

Architecture:
LSTMs outperform basic RNNs significantly, confirming their ability to capture long-term dependencies.
The BiLSTM improves slightly on RNN but not enough to justify doubled runtime.

Activation:
ReLU and Tanh activations perform similarly, with ReLU converging faster and achieving the highest F1.
Sigmoid tends to saturate early.

Optimizer:
Adam provides the best overall stability and accuracy.
SGD fails to converge well, while RMSProp performs comparably to Adam but with slightly slower learning.

Sequence Length:
Longer sequences (100 tokens) improve accuracy up to ~0.82 but at the cost of nearly doubled training time.
Shorter sequences (<25 tokens) lose context and drop to ~0.72.

Gradient Clipping:
Clipping slightly reduces training variance and runtime but doesn’t impact accuracy significantly.
It’s recommended as a safe default for stability.

Best Configuration

LSTM + ReLU + Adam + Sequence Length = 100 + Gradient Clipping = Yes

Achieved:

- Accuracy: **0.8206**
- F1-score: **0.8205**
- Time per epoch: ~19 s

This configuration achieves the optimal trade-off between model complexity, stability, and accuracy.

Conclusion

This project demonstrated the importance of architecture and sequence length in recurrent models for sentiment analysis.
While basic RNNs struggle with long dependencies, LSTMs handle them effectively.
Using Adam optimization with ReLU activation yields robust convergence.
The **LSTM with sequence length 100** provides the highest validation accuracy (≈0.82), confirming that longer contexts improve semantic understanding, albeit with increased computation time.