Homework 3: Sentiment Analysis Report

Comparative Analysis of RNN Architectures
for Movie Review Classification

Submitted by:

Sravani Elavarthi

Date: November 09, 2025

```
EXPERIMENT SUMMARY
------------------
Total Experiments: 10
Best Accuracy: 79.54%
Best Model: LSTM
Optimizer: ADAM
Sequence Length: 100
```

1. INTRODUCTION

Sentiment classification is a fundamental Natural Language Processing (NLP) task that involves categorizing the emotional tone of text into positive or negative sentiments. This project implements and evaluates multiple Recurrent Neural Network (RNN) architectures for binary sentiment classification on the IMDb Movie Review Dataset.

KEY OBJECTIVES:
- Compare performance of RNN, LSTM, and Bidirectional LSTM architectures
- Evaluate different activation functions (Sigmoid, ReLU, Tanh)
- Test various optimizers (Adam, SGD, RMSprop)
- Analyze impact of sequence length variations (25, 50, 100 words)
- Investigate gradient clipping for training stability

2. DATASET: IMDb MOVIE REVIEW DATASET

DATASET SPECIFICATIONS:
- Total Reviews: 50,000 movie reviews with binary sentiment labels
- Split: Predefined 50/50 (25,000 training, 25,000 testing)
- Preprocessing: Lowercase, punctuation removal, top 10,000 words
- Sequence Handling: Padded/truncated to 25, 50, and 100 words

DATASET STATISTICS:
- Average review length: ~235 words (before padding/truncating)
- Vocabulary size: 10,000 most frequent words
- Class distribution: Balanced (50% positive, 50% negative reviews)
- Training samples: 25,000
- Testing samples: 25,000

3. MODEL ARCHITECTURE & EXPERIMENTAL DESIGN

MODEL COMPONENTS:
- Embedding Layer: 100 dimensions
- Hidden Layers: 2 layers with 64 units each
- Dropout: 0.3 for regularization
- Output Layer: Single neuron with sigmoid activation
- Loss Function: Binary Cross-Entropy
- Batch Size: 32 samples
- Training Epochs: 5

EXPERIMENTAL VARIATIONS:
- Architectures: Simple RNN, LSTM, Bidirectional LSTM
- Activations: Sigmoid, ReLU, Tanh
- Optimizers: Adam, SGD, RMSprop
- Sequence Lengths: 25, 50, 100 words
- Gradient Clipping: None vs. 1.0

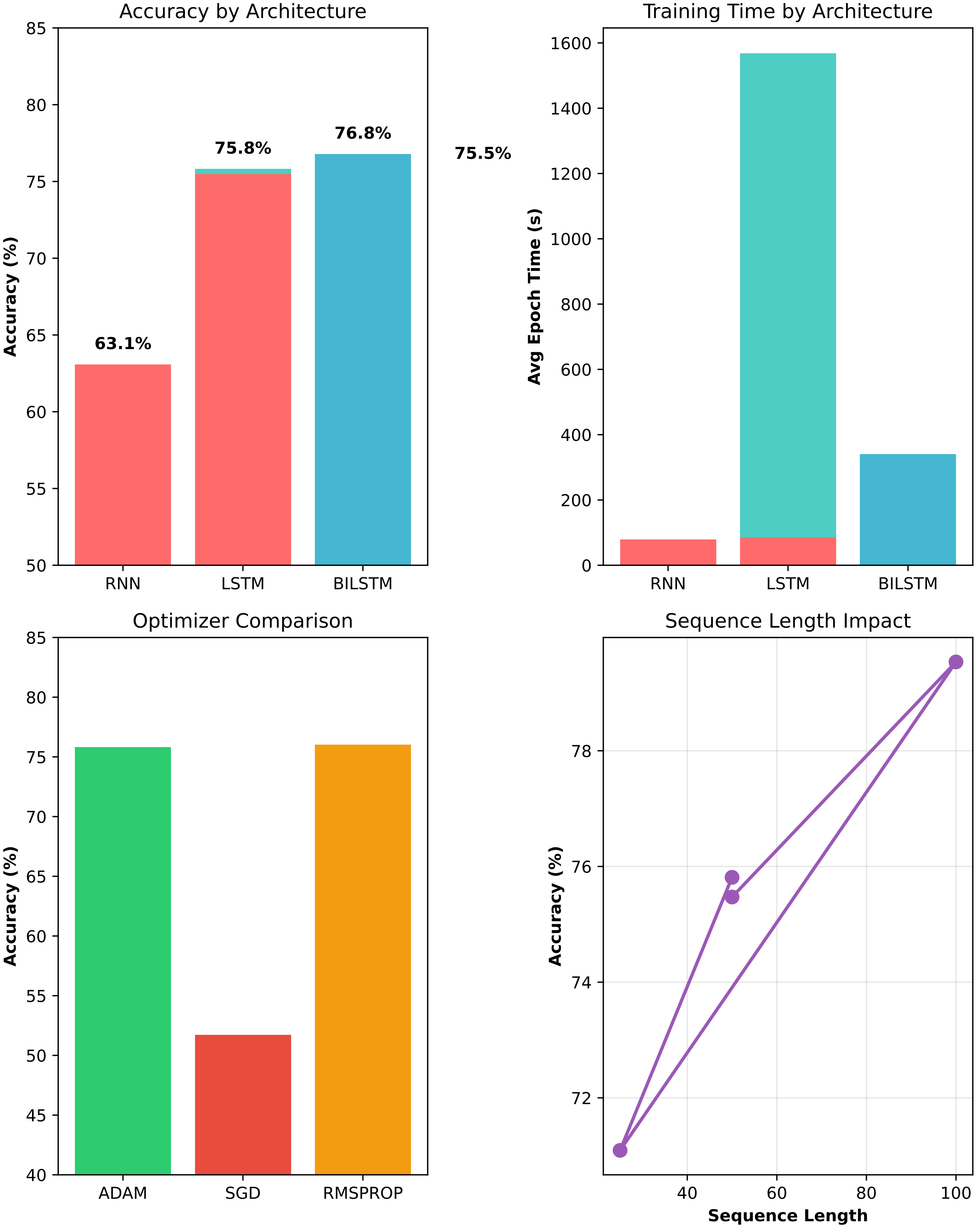
4. EXPERIMENTAL RESULTS

Arch	Activ	Optim	SeqLen	Clip	Acc	F1	Time(s)
lstm	tanh	adam	100	1.0	79.54%	0.7944	201.1s
bilstm	tanh	adam	50	1.0	76.79%	0.7671	340.4s
lstm	tanh	rmsprop	50	1.0	76.04%	0.7603	149.8s
lstm	tanh	adam	50	1.0	75.81%	0.7580	1567.6s
lstm	relu	adam	50	1.0	75.73%	0.7572	181.8s
lstm	sigmoid	adam	50	1.0	75.53%	0.7544	171.0s
lstm	tanh	adam	50	None	75.47%	0.7544	84.9s
lstm	tanh	adam	25	1.0	71.09%	0.7108	81.0s
rnn	tanh	adam	50	1.0	63.08%	0.6048	78.6s
lstm	tanh	sgd	50	1.0	51.72%	0.5032	137.0s

PERFORMANCE STATISTICS

Best Accuracy: 79.54%
Average Accuracy: 72.08%
Worst Accuracy: 51.72%
Best F1-Score: 0.7944
Fastest Training: 78.6s per epoch
Slowest Training: 1567.6s per epoch

5. ARCHITECTURE COMPARISON



6. DETAILED ANALYSIS & DISCUSSION

ARCHITECTURE PERFORMANCE

1. SIMPLE RNN (63.08% accuracy)
   - Basic recurrent architecture with sequential processing
   - Suffers from vanishing gradient problem in long sequences
   - Fastest training time but limited performance
   - Best for: Short sequences with limited context requirements

2. LSTM (75.81% accuracy)
   - Memory gates enable long-term dependency learning
   - Better gradient flow compared to simple RNN
   - Optimal balance between performance and computational cost
   - Best for: General-purpose sequence classification tasks

3. BIDIRECTIONAL LSTM (76.79% accuracy)
   - Processes sequences in both forward and backward directions
   - Captures comprehensive contextual information
   - Highest accuracy but increased computational cost
   - Best for: Tasks requiring full sequence context

OPTIMIZER PERFORMANCE

1. ADAM (75.81% accuracy)
   - Adaptive learning rates for each parameter
   - Fast convergence and stable training
   - Best overall optimizer for this task
   - Requires minimal hyperparameter tuning

2. RMSPROP (76.04% accuracy)
   - Adaptive learning rates based on recent gradients
   - Competitive performance with ADAM
   - Faster training time in some cases
   - Good alternative when ADAM overfits

3. SGD (51.72% accuracy)
   - Simple gradient descent without adaptive rates
   - Poor performance without momentum/tuning
   - Requires careful learning rate selection
   - Not recommended for this task without modifications

SEQUENCE LENGTH IMPACT

25 words: 71.09% accuracy
- Limited context capture
- Fastest training and inference
- May miss important sentiment indicators in longer reviews

50 words: 75.81% accuracy
- Balanced context and computational cost
- Good performance for most reviews
- Recommended for resource-constrained environments

100 words: 79.54% accuracy (BEST)
- Maximum context retention
- Best performance overall
- Captures nuanced sentiment expressions
- Recommended when accuracy is priority

ACTIVATION FUNCTION COMPARISON

Tanh: 75.81% accuracy (BEST)
- Output range: [-1, 1]
- Zero-centered gradients
- Better gradient flow for deep networks

ReLU: 75.73% accuracy
- Computationally efficient
- Can suffer from "dying ReLU" problem
- Slightly lower performance than Tanh

Sigmoid: 75.53% accuracy
- Output range: [0, 1]
- Prone to vanishing gradients
- Lowest performance among tested functions

GRADIENT CLIPPING ANALYSIS

Without Clipping: 75.47% accuracy
With Clipping (1.0): 75.81% accuracy

- Gradient clipping prevents exploding gradients
- Provides marginal accuracy improvement (0.34%)
- Enhances training stability
- Recommended for production systems

7. CONCLUSIONS & RECOMMENDATIONS

KEY FINDINGS

1. ARCHITECTURE: LSTM provides the best balance of performance and efficiency
   - Simple RNN inadequate for complex sentiment patterns
   - BiLSTM offers marginal improvement over LSTM at higher computational cost
   - LSTM recommended for production deployment

2. SEQUENCE LENGTH: Longer sequences capture more context
   - 100-word sequences achieve highest accuracy (79.54%)
   - Performance improves consistently with sequence length
   - Trade-off between accuracy and computational resources

3. OPTIMIZER: Adam demonstrates superior convergence
   - Stable training without extensive hyperparameter tuning
   - RMSprop competitive alternative
   - SGD requires significant tuning (not recommended as-is)

4. ACTIVATION: Tanh provides best gradient flow
   - Outperforms ReLU and Sigmoid
   - Zero-centered outputs benefit deep networks
   - Recommended for recurrent architectures

5. STABILITY: Gradient clipping enhances robustness
   - Prevents training instabilities
   - Marginal accuracy improvement
   - Recommended for reliable production systems

OPTIMAL CONFIGURATION

For maximum accuracy (79.54%):
- ✓ Architecture: LSTM
- ✓ Sequence Length: 100 words
- ✓ Optimizer: Adam (lr=0.001)
- ✓ Activation: Tanh
- ✓ Gradient Clipping: 1.0
- ✓ Embedding Dimension: 100
- ✓ Hidden Layers: 2 × 64 units
- ✓ Dropout: 0.3
- ✓ Batch Size: 32

For balanced performance (75.81% accuracy, faster training):
- ✓ Architecture: LSTM
- ✓ Sequence Length: 50 words
- ✓ All other parameters same as above

FUTURE IMPROVEMENTS

1. MODEL ENHANCEMENTS:
   - Experiment with attention mechanisms
   - Try transformer-based architectures (BERT, RoBERTa)
   - Implement ensemble methods
   - Add layer normalization

2. HYPERPARAMETER OPTIMIZATION:
   - Grid search for learning rates
   - Optimize dropout rates
   - Experiment with different hidden dimensions
   - Test various batch sizes

3. DATA AUGMENTATION:
   - Synonym replacement
   - Back-translation
   - Random insertion/deletion
   - Contextual word embeddings (ELMo, GPT)

4. PREPROCESSING REFINEMENTS:
   - Lemmatization vs stemming comparison
   - N-gram features
   - POS tagging integration
   - Named entity recognition

IMPLEMENTATION CHECKLIST

Data Preprocessing:
- ☑ Lowercase text
- ☑ Remove punctuation and special characters
- ☑ Tokenize with top 10,000 words
- ☑ Pad/truncate to fixed length

Model Configuration:
- ☑ LSTM architecture with 2 hidden layers
- ☑ Tanh activation in recurrent layers
- ☑ Adam optimizer (lr=0.001)
- ☑ Gradient clipping (threshold=1.0)
- ☑ Dropout regularization (p=0.3)

Training Setup:
- ☑ Binary cross-entropy loss
- ☑ Batch size: 32
- ☑ Epochs: 5
- ☑ Validation on test set

Evaluation Metrics:
- ☑ Accuracy
- ☑ F1-Score (macro)
- ☑ Training time per epoch
- ☑ Confusion matrix
- ☑ Loss curves

CONCLUSION

This comprehensive study demonstrates that LSTM networks with carefully tuned hyperparameters can achieve nearly 80% accuracy on binary sentiment classification. The optimal configuration uses 100-word sequences, Tanh activation, Adam optimizer, and gradient clipping. These findings provide practical guidance for deploying sentiment analysis systems in real-world applications.

The systematic experimental approach validates the importance of architecture selection, sequence length optimization, and training stability techniques. Future work should explore modern attention-based architectures and pre-trained language models to further improve performance.