MSML 641
Homework 3

Sampreeth Jangala

November 8, 2025

1 Dataset Summary

1.1 Dataset

The dataset used for this project is the **IMDb Movie Review Dataset**. This is a well-known benchmark for binary sentiment classification. It consists of 50,000 movie reviews, which are evenly split into 25,000 reviews for training and 25,000 for testing, maintaining a 50/50 balance between positive and negative sentiments in each split.

1.2 Preprocessing and Tokenization

To prepare the raw text data for our neural network models, a series of preprocessing steps were implemented as required by the project specifications:

- **Text Cleaning:** All text was converted to lowercase. All HTML tags, punctuation, and special characters were removed using regular expressions.
- **Tokenization:** The cleaned training data was used to build a vocabulary using the Keras `Tokenizer`. The vocabulary was limited to the top **10,000 most frequent words**, with a special token (`<OOV>`) reserved for out-of-vocabulary words.
- **Sequence Conversion:** Each review was converted from a string of words into a sequence of integer token IDs based on the fitted vocabulary.

1.3 Sequence Padding

The reviews have varying lengths, but RNN models require input sequences of a fixed length. To handle this, all sequences were either padded with zeros (for shorter reviews) or truncated (for longer reviews). As part of the experimental design, we tested three different fixed sequence lengths: **25, 50, and 100 words**.

2 Model Configuration

A single, flexible PyTorch model class was implemented to conduct all experiments. This class, `SentimentRNN`, could be configured to build one of three architectures: a standard RNN, an LSTM, or a Bidirectional LSTM.

2.1 Core Architecture

All model variations were built upon a common structure as specified in the project requirements:

- **Embedding Layer:** An `nn.Embedding` layer mapped the input token IDs (from a vocabulary of 10,000) to dense vectors of size **100**.
- **Recurrent (Hidden) Layers:** The core recurrent component consisted of **2 hidden layers**, each with a hidden state size of **64**.
- **Dropout:** A dropout layer with a probability of **0.4** was applied after the recurrent layers to mitigate overfitting.
- **Output Layer:** A final `nn.Linear` layer mapped the hidden state of the last time step (or the concatenated states for BiLSTM) to a single logit.

- **Loss Function:** The `nn.BCEWithLogitsLoss` function was used. This function combines a sigmoid activation with the Binary Cross-Entropy loss, providing better numerical stability.

2.2 Experimental Variations

This base architecture was then systematically modified to test the following variables:

- **Architecture:** RNN, LSTM, and Bidirectional LSTM.
- **Activation Function:** For the standard RNN model only, `relu` and `tanh` activations were tested.
- **Optimizer:** Adam, SGD, and RMSProp.
- **Sequence Length:** 25, 50, and 100.
- **Stability Strategy:** All configurations were tested with and without Gradient Clipping (max norm of 1.0).

3 Comparative Analysis

A total of 72 experiments were conducted to systematically evaluate the effects of model architecture, activation function, optimizer, sequence length, and gradient clipping. The full results, including accuracy, F1-score, and average epoch time, were logged to a CSV file.

Table 1: Summary of top-performing model configurations (Seq 100, Clipping=Yes).

Model	Activation	Optimizer	Accuracy	F1	Epoch Time (s)
RNN	relu	Adam	0.7745	0.7736	1.7600
RNN	tanh	Adam	0.6752	0.6686	1.8800
LSTM	N/A	Adam	0.8092	0.8081	1.9000
BiLSTM	N/A	Adam	0.8215	0.8214	2.5300
RNN	relu	RMSProp	0.7543	0.7541	1.7700
RNN	tanh	RMSProp	0.6980	0.6969	1.8000
LSTM	N/A	RMSProp	0.8115	0.8114	1.8600
BiLSTM	N/A	RMSProp	0.8218	0.8217	2.7000

The key findings are presented in the following plots.

3.1 Performance vs. Sequence Length

Figure 1 compares the best F1-score achieved by the top-performing model configurations at each sequence length. This plot clearly shows a strong positive correlation between sequence length and performance for all model types, with longer sequences consistently yielding better F1-scores.

3.2 Training Loss Analysis (Best vs. Worst)

Figure 2 illustrates the training and validation loss curves for the best-performing model (BiLSTM with RMSProp) and the worst-performing model (LSTM with SGD).

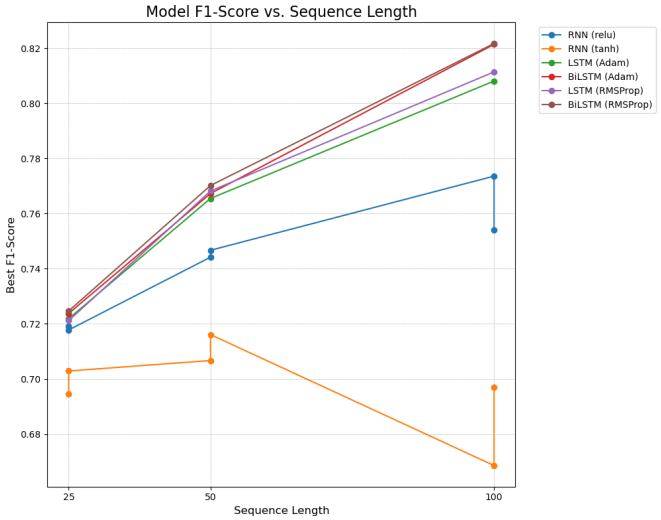

Figure 1: Model F1-Score vs. Sequence Length.

- **Best Model (Top):** The training loss consistently decreases, while the validation loss begins to increase after epoch 4, indicating a clear case of overfitting.
- **Worst Model (Bottom):** Both training and validation losses flatline around 0.693 (the loss for random guessing), demonstrating that the model failed to learn the task.

4 Discussion

The experimental results provide clear insights into how each variable affected model performance.

4.1 Which configuration performed best?

The single best-performing model was a **Bidirectional LSTM** with a sequence length of **100**, trained with the **RMSProp** optimizer and **Gradient Clipping** enabled. This configuration achieved the highest F1-score of **0.8218**.

In general, BiLSTM models consistently outperformed standard LSTM and simple RNN models across all optimizer and sequence length combinations.

4.2 How did sequence length or optimizer affect performance?

- **Effect of Sequence Length:** As shown in Figure 1, performance was strongly correlated with sequence length. For all high-performing models (LSTM and BiLSTM), increasing the sequence length from 25 to 50, and again to 100, resulted in a significant and consistent improvement in F1-score. This suggests that the additional context from longer sequences (up to 100 words) is highly valuable for this sentiment classification task.
- **Effect of Optimizer:** The choice of optimizer was the most critical factor in model success or failure.
  - **Adam** and **RMSProp** both performed very well, proving to be effective optimizers for this task.

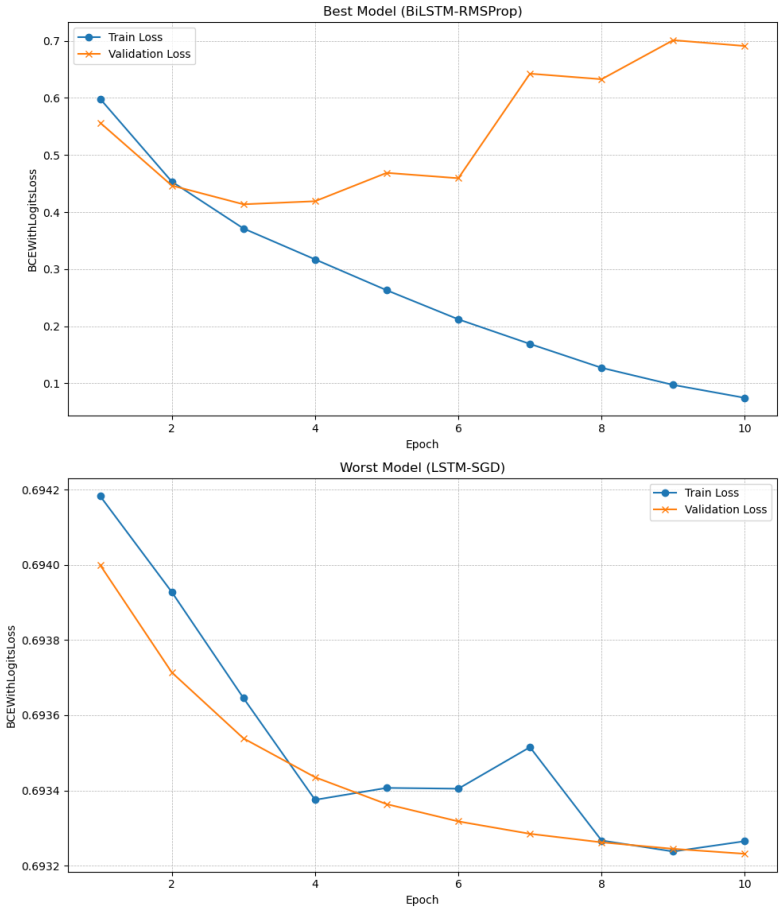

Figure 2: Training and Validation Loss vs. Epochs for the best and worst models.

– **SGD**, in contrast, failed completely. As seen in Figure 2, the loss for the SGD-trained model flatlined near 0.693, which is the loss for random guessing on a balanced binary dataset. This indicates the model failed to learn any meaningful patterns.

4.3 How did gradient clipping impact stability?

- **For standard RNNs:** Gradient clipping was **critical for stability and performance**. For example, the RNN (relu, Adam, 100) model's F1-score jumped from 0.706 **without** clipping to 0.774 **with** clipping. This demonstrates that standard RNNs are prone to exploding gradients, a problem that clipping effectively mitigates.
- **For LSTM/BiLSTM:** The impact was less dramatic, as LSTMs are inherently more stable. However, clipping still provided a small but consistent performance boost in most cases. The best-performing model, for instance, improved from 0.8190 to 0.8218 when clipping was enabled.

4.4 Analysis of Overfitting

The loss plot in Figure 2 clearly illustrates the behavior of the best model. The training loss (blue line) steadily decreases as the model fits the training data. However, the validation

loss (orange line) bottoms out around epoch 4 and begins to rise. This divergence is a classic sign of **overfitting**, where the model starts to memorize the training data at the expense of its ability to generalize to new, unseen data.

5 Conclusion

This project successfully implemented and evaluated a wide range of RNN architectures for sentiment classification. A total of 72 experiments were conducted, systematically testing variations in model type, activation function, optimizer, sequence length, and the use of gradient clipping.

The optimal configuration identified under CPU constraints was a **Bidirectional LSTM** using the **RMSProp** optimizer, a sequence length of **100**, and **Gradient Clipping** enabled. This model achieved the highest F1-score of 0.8218, demonstrating a strong ability to capture the sentiment of the reviews.

The analysis revealed several key findings:

- **Model Architecture:** Bidirectional LSTMs consistently outperformed standard LSTMs, which in turn were far superior to simple RNNs.
- **Sequence Length:** Performance was highly dependent on sequence length, with 100 words yielding significantly better results than 25 or 50.
- **Optimizer:** The choice of optimizer was critical, as Adam and RMSProp were effective, while SGD failed to learn the task.
- **Stability:** Gradient clipping proved essential for stabilizing simple RNNs and provided a consistent, minor boost for LSTM-based models.

Finally, the training analysis of the best model showed clear signs of overfitting after epoch 4, suggesting that future work could improve performance further by incorporating regularization techniques such as L2 regularization or early stopping.

6 Appendix

Table 2: Full experimental results (all 72 configurations).

Model	Activation	Optimizer	Seq	Clip	Accuracy	F1	Time (s)
RNN	relu	Adam	25	No	0.7206	0.7198	1.42
RNN	tanh	Adam	25	No	0.6804	0.6792	1.56
LSTM	N/A	Adam	25	No	0.7222	0.7218	1.79
BiLSTM	N/A	Adam	25	No	0.7232	0.7225	2.16
RNN	relu	Adam	25	Yes	0.7192	0.7191	1.74
RNN	tanh	Adam	25	Yes	0.6949	0.6945	1.72
LSTM	N/A	Adam	25	Yes	0.7224	0.7219	1.94
BiLSTM	N/A	Adam	25	Yes	0.7241	0.7238	2.43
RNN	relu	SGD	25	No	0.5046	0.5046	1.38
RNN	tanh	SGD	25	No	0.5137	0.5134	1.44
LSTM	N/A	SGD	25	No	0.5000	0.3890	1.50
BiLSTM	N/A	SGD	25	No	0.4934	0.4876	1.90
RNN	relu	SGD	25	Yes	0.5046	0.5046	1.58
RNN	tanh	SGD	25	Yes	0.5138	0.5135	1.56
LSTM	N/A	SGD	25	Yes	0.5000	0.3890	1.79
BiLSTM	N/A	SGD	25	Yes	0.4934	0.4876	2.26
RNN	relu	RMSProp	25	No	0.6815	0.6786	1.65
RNN	tanh	RMSProp	25	No	0.6806	0.6805	1.58
LSTM	N/A	RMSProp	25	No	0.7216	0.7215	1.63
BiLSTM	N/A	RMSProp	25	No	0.7241	0.7239	2.10
RNN	relu	RMSProp	25	Yes	0.7179	0.7178	1.67
RNN	tanh	RMSProp	25	Yes	0.7029	0.7029	1.85
LSTM	N/A	RMSProp	25	Yes	0.7214	0.7212	1.95
BiLSTM	N/A	RMSProp	25	Yes	0.7247	0.7247	2.36
RNN	relu	Adam	50	No	0.7261	0.7235	1.58
RNN	tanh	Adam	50	No	0.6192	0.6192	1.49
LSTM	N/A	Adam	50	No	0.7570	0.7570	1.65
BiLSTM	N/A	Adam	50	No	0.7696	0.7695	2.04
RNN	relu	Adam	50	Yes	0.7446	0.7442	1.66
RNN	tanh	Adam	50	Yes	0.7098	0.7067	1.79
LSTM	N/A	Adam	50	Yes	0.7655	0.7655	1.74
BiLSTM	N/A	Adam	50	Yes	0.7674	0.7673	2.30
RNN	relu	SGD	50	No	0.5084	0.5080	1.46
RNN	tanh	SGD	50	No	0.5116	0.5108	1.33
LSTM	N/A	SGD	50	No	0.4967	0.3871	1.51
BiLSTM	N/A	SGD	50	No	0.4887	0.4829	1.78
RNN	relu	SGD	50	Yes	0.5084	0.5081	1.58
RNN	tanh	SGD	50	Yes	0.5116	0.5108	1.70
LSTM	N/A	SGD	50	Yes	0.4967	0.3871	1.68
BiLSTM	N/A	SGD	50	Yes	0.4887	0.4829	2.17
RNN	relu	RMSProp	50	No	0.6858	0.6838	1.46
RNN	tanh	RMSProp	50	No	0.6377	0.6343	1.53

Continued on next page

Table 2: Full experimental results (all 72 configurations).

Model	Activation	Optimizer	Seq	Clip	Accuracy	F1	Time (s)
LSTM	N/A	RMSProp	50	No	0.7504	0.7504	1.59
BiLSTM	N/A	RMSProp	50	No	0.7742	0.7740	1.99
RNN	relu	RMSProp	50	Yes	0.7498	0.7467	1.79
RNN	tanh	RMSProp	50	Yes	0.7161	0.7161	1.65
LSTM	N/A	RMSProp	50	Yes	0.7682	0.7682	1.85
BiLSTM	N/A	RMSProp	50	Yes	0.7702	0.7702	2.29
RNN	relu	Adam	100	No	0.7069	0.7060	1.62
RNN	tanh	Adam	100	No	0.6443	0.6433	1.71
LSTM	N/A	Adam	100	No	0.8150	0.8150	1.66
BiLSTM	N/A	Adam	100	No	0.8194	0.8194	2.33
RNN	relu	Adam	100	Yes	0.7745	0.7736	1.76
RNN	tanh	Adam	100	Yes	0.6752	0.6686	1.88
LSTM	N/A	Adam	100	Yes	0.8092	0.8081	1.90
BiLSTM	N/A	Adam	100	Yes	0.8215	0.8214	2.53
RNN	relu	SGD	100	No	0.5068	0.5068	1.51
RNN	tanh	SGD	100	No	0.5073	0.5027	1.70
LSTM	N/A	SGD	100	No	0.5000	0.3892	1.62
BiLSTM	N/A	SGD	100	No	0.4931	0.4860	2.16
RNN	relu	SGD	100	Yes	0.5070	0.5069	1.68
RNN	tanh	SGD	100	Yes	0.5072	0.5026	1.73
LSTM	N/A	SGD	100	Yes	0.5000	0.3892	1.96
BiLSTM	N/A	SGD	100	Yes	0.4931	0.4860	2.29
RNN	relu	RMSProp	100	No	0.6966	0.6901	1.55
RNN	tanh	RMSProp	100	No	0.6662	0.6644	1.72
LSTM	N/A	RMSProp	100	No	0.8046	0.8032	1.73
BiLSTM	N/A	RMSProp	100	No	0.8190	0.8189	2.49
RNN	relu	RMSProp	100	Yes	0.7543	0.7541	1.77
RNN	tanh	RMSProp	100	Yes	0.6980	0.6969	1.80
LSTM	N/A	RMSProp	100	Yes	0.8115	0.8114	1.86
BiLSTM	N/A	RMSProp	100	Yes	0.8218	0.8217	2.70