# HOMEWORK - 3


## Abstract

This report investigates recurrent neural architectures for binary sentiment classification on the IMDB 50k dataset. We vary architecture (RNN, LSTM, BiLSTM), activation (tanh/ReLU/sigmoid on RNN), optimizer (Adam, SGD, RMSProp), sequence length (25/50/100), and stability strategy (no clipping vs gradient clipping). All other hyperparameters remain fixed to ensure clean attribution of effects. Across 14 experiments, the strongest configuration is LSTM, sequence length 100, Adam, no clipping, achieving Accuracy = 0.8122, F1 = 0.8137, Loss = 0.4244. We will analyze the plots and metrics, explain why longer context and LSTM dominate, and conclude with a practical recommendation that adds early stopping and tiny weight decay for improved calibration.


## 1. Introduction

Sentiment in movie reviews often relies on long range context. Recurrent models are natural candidates, but choices about architecture, optimizer, and input length strongly affect performance and stability. The goal of this assignment is a comparative study that isolates the impact of each factor while keeping the rest fixed, and to present results and insights clearly.

## 2. Dataset and Preprocessing

- **Dataset:** IMDb 50,000 movie reviews with binary labels.
- **Split: 50/50** => 25k train and 25k test.
- **Normalization:** lowercase, punctuation removal, tokenization.
- **Vocabulary:** top **10,000** most frequent tokens, <PAD> and <UNK> included.
- **Sequence lengths:** pad/truncate to **25**, **50**, **100** tokens.

This pipeline ensures consistent inputs for all models and enables a clean sequence-length study.

## 3. Model Design and Training Setup

These are fixed across experiments unless a factor is varied **:**

- **Embedding:** 100
- **Hidden size: 64**
- **Recurrent layers: 2**
- **Dropout: 0.5**
- **Batch size:** 32

- **Loss:** Binary Cross-Entropy with logits
- **Output decision:** sigmoid threshold at 0.5
- **Learning rate:** 1e-3
- **Epochs:** 10 per run
- **Seed for reproducibility:** 42

**Architectures compared**

- **RNN** (vanilla) with **tanh**, **ReLU**, or **sigmoid** activation.
- **LSTM** (unidirectional).
- **BiLSTM** (bidirectional LSTM).

# 4. Experimental Design

We establish a baseline and change one factor at a time:

- **Baseline (BASE):** LSTM, **sequence length 50**, **Adam**, **no gradient clipping**.
- **Factors & levels:**
    - **Architecture:** RNN, LSTM, BiLSTM (at L=50, Adam, no clip).
    - **Activation:** tanh, ReLU, sigmoid.
    - **Optimizer:** Adam, SGD, RMSProp.
    - **Sequence length:** 25, 50, 100.
    - **Stability:** no strategy vs gradient clipping = 1.0.

# 5. Results

## 5.1 Summary Table

| Experiment (Full Name) | Activation Used | Accuracy | F1 | Test Loss | Epoch Time (s) |
|---|---|---|---|---|---|
| BASE (LSTM, L=50, Adam) | ReLU (default for LSTM) | 0.7688 | 0.7635 | 0.4904 | 24.84 |
| ARCH_RNN_TANH (RNN, L=50, Adam) | Tanh | 0.6194 | 0.6494 | 0.6582 | 11.59 |
| ARCH_BILSTM (BiLSTM, L=50, Adam) | ReLU (default for BiLSTM) | 0.7530 | 0.7693 | 0.5047 | 50.68 |
| RNN_RELU (RNN, L=50, Adam) | ReLU | 0.7461 | 0.7499 | 0.5359 | 10.03 |
| RNN_SIGMOID (RNN, L=50, Adam) | Sigmoid | 0.5701 | 0.6061 | 0.6824 | 21.95 |
| OPT_SGD (LSTM, L=50, SGD) | ReLU (default for LSTM) | 0.5157 | 0.4564 | 0.6926 | 25.26 |
| OPT_RMSPROP (LSTM, L=50, RMSProp) | ReLU (default for LSTM) | 0.7664 | 0.7687 | 0.5009 | 28.59 |
| SEQ_25 (LSTM, L=25, Adam) | ReLU (default for LSTM) | 0.7226 | 0.7239 | 0.5556 | 15.55 |
| SEQ_100 (LSTM, L=100, Adam) | ReLU (default for LSTM) | **0.8122** | **0.8137** | **0.4244** | 50.76 |
| CLIP_ON (LSTM, L=50, Adam, Clip=1.0) | ReLU (default for LSTM) | 0.7544 | 0.7372 | 0.5054 | 28.95 |
| ARCH_RNN_TANH_CLIP (RNN, L=50, Adam, Clip=1.0) | Tanh | 0.7096 | 0.7135 | 0.6113 | 15.12 |
| ARCH_BILSTM_CLIP (BiLSTM, L=50, Adam, Clip=1.0) | ReLU (default for BiLSTM) | 0.7638 | 0.7584 | 0.4959 | 50.88 |
| OPT_SGD_CLIP (LSTM, L=50, SGD, Clip=1.0) | ReLU (default for LSTM) | 0.5174 | 0.4878 | 0.6927 | 26.07 |
| OPT_RMSPROP_CLIP (LSTM, L=50, RMSProp, Clip=1.0) | ReLU (default for LSTM) | 0.7480 | 0.7692 | 0.5257 | 26.77 |

## 5.2 Plot Interpretations

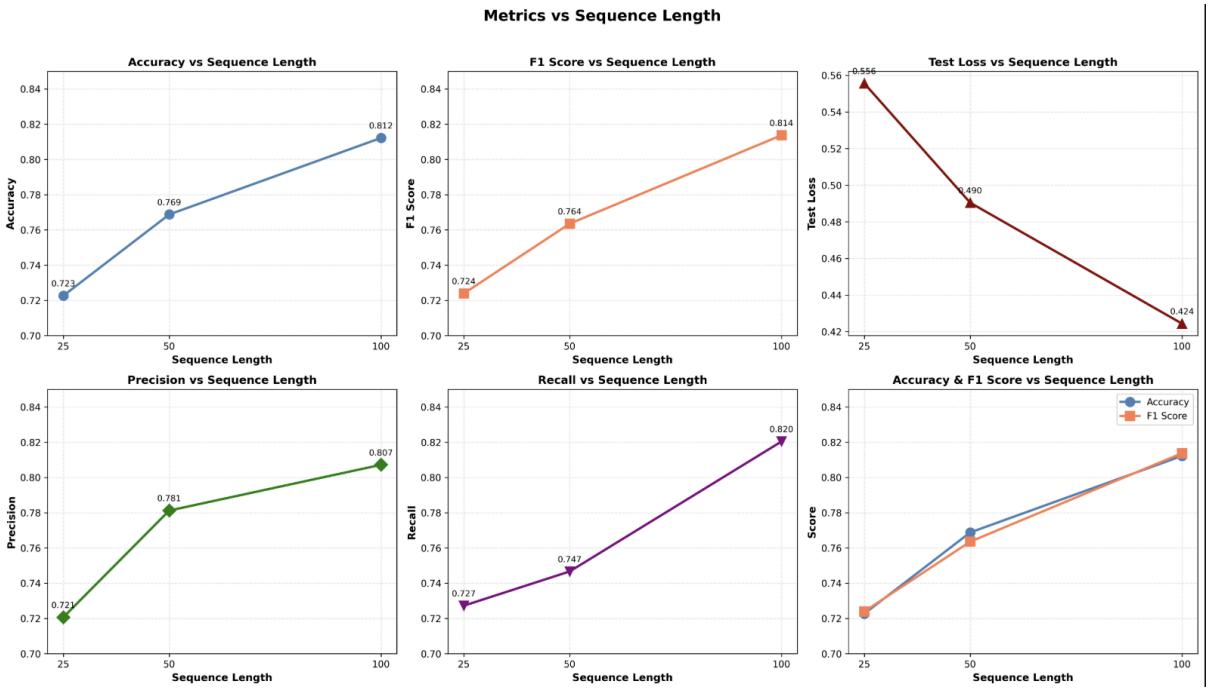


**Figure 1.** ***Metrics vs Sequence Length***

This six panel figure isolates the impact of input length using LSTM with constant settings. Accuracy rises from 0.7226 (L=25) to 0.7688 (L=50) and 0.8122 (L=100), and F1 follows the same trend (0.7239 -> 0.7635 -> 0.8137). The test loss decreases monotonically (0.556 -> 0.490 -> 0.424), indicating not just more correct predictions but also better probability calibration. Precision improves with length (0.721 -> 0.781 -> 0.807), meaning positive predictions are increasingly trustworthy. Recall also improves (0.727 -> 0.747 -> 0.820), showing the model misses fewer true positives with more context. The combined Accuracy & F1 panel visualizes these trends moving in tandem, confirming the gain is balanced across classes rather than an artifact of precision recall trade-offs. The main trade-off is compute: epoch time roughly doubles from L=50 to L=100, but the accuracy and loss improvements justify the longer input in this task.

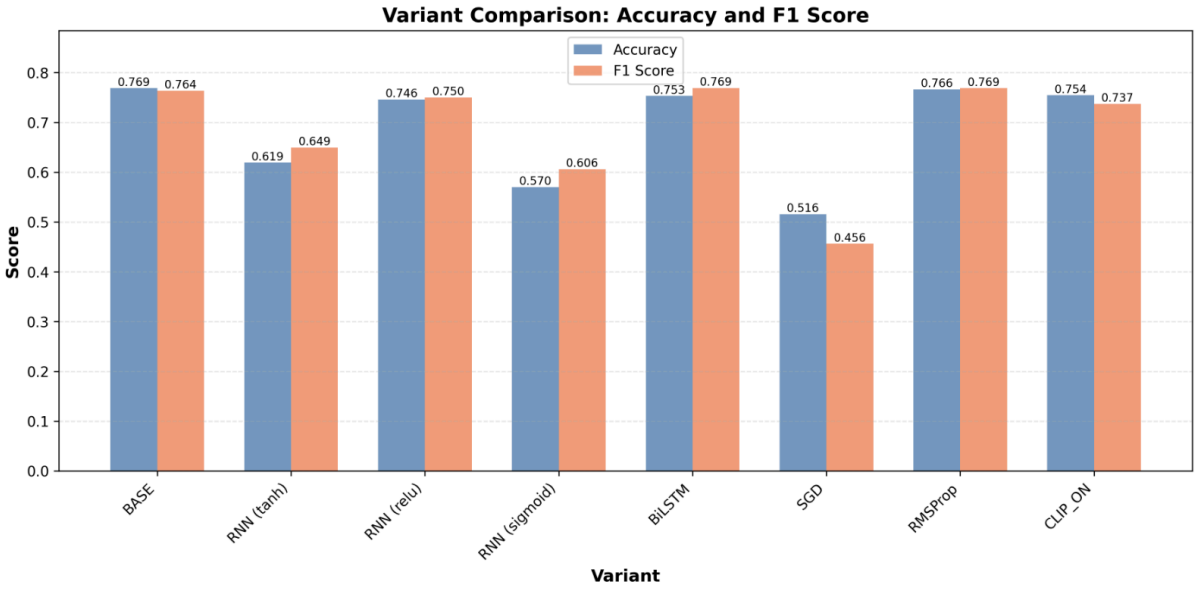


**Figure 2.** ***Variant Comparison: Accuracy and F1***

This bar chart compares architectures, optimizers, and clipping at L=50. Among architectures, LSTM provides the best balance (0.769,0.764), BiLSTM is close in F1 (0.769) but slightly lower in accuracy (0.753) and substantially slower, while RNN-ReLU is competitive but still behind LSTM (0.746/0.750). RNN-tanh and RNN-sigmoid underperform consistent with vanishing gradient behavior and saturation in ungated RNNs on long sequences. For optimizers, Adam and RMSProp perform similarly out of the box (~0.77), whereas SGD with lr=1e-3 is weak (0.516/0.456) SGD typically needs tuned lr/momentum schedules to compete, which is outside OFAT's fixed-hyperparameter protocol. Regarding stability, gradient clipping (1.0) helps where training is more volatile, RNN-tanh improves substantially (Acc 0.619 -> 0.710, F1 0.649 -> 0.713), and BiLSTM nudges upward with slightly lower loss. For the already stable LSTM baseline, clipping is mildly detrimental(0.7688 -> 0.7544 Accuracy), suggesting unnecessary damping of useful updates.

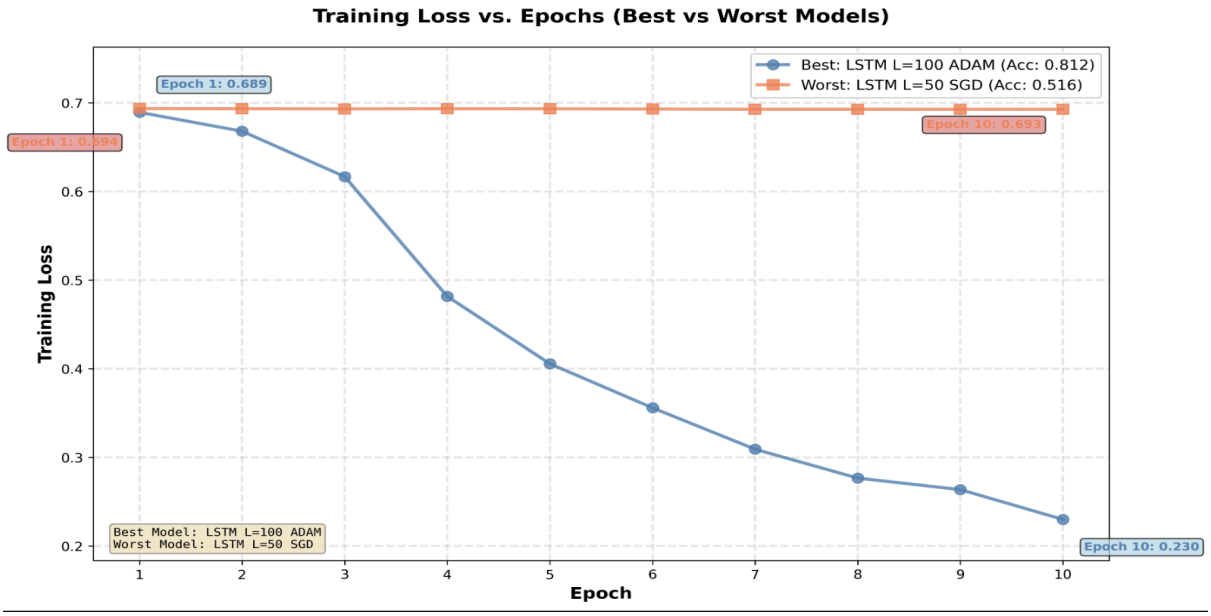


**Figure 3. Training Loss vs. Epochs (Best vs Worst Model)**

The best model LSTM, L=100, Adam shows steady, accelerating convergence: training loss falls from 0.689 (epoch 1) to 0.230 (epoch 10) with a sharp inflection around epochs 3 to 5, indicating the optimizer quickly finds a useful region and then refines. In contrast, the worst model LSTM, L=50, SGD (lr=1e-3) stagnates near 0.69 across all epochs. This plateau signals failure to learn under the fixed OFAT hyperparameters, vanilla SGD at this learning rate cannot make effective progress, whereas Adam's adaptive steps exploit gradient structure to descend rapidly. The gap also reflects sequence length effects: L=100 exposes more sentiment evidence per sample, which speeds optimization and ultimately yields better generalization. Overall, the plot visualizes two regimes, effective adaptive optimization + sufficient context vs under tuned first order updates and explains why Adam + L=100 dominates.

# 6. Discussion

- Sequence length is the strongest lever. Accuracy and F1 rise steadily from L=25->50->100 (0.72->0.77->0.81), while test loss drops (0.556->0.490->0.424). Longer inputs retain negations, contrastive cues, and late verdicts that short snippets miss. The improvement is consistent across metrics.
- Calibration improves with length. The loss reduction is larger than the accuracy gain, indicating better aligned probabilities. Precision climbs (0.721->0.781->0.807) and recall jumps (0.727->0.747->0.820), so gains are balanced rather than trading one for the other.
- LSTM is the safest architecture at equal settings (L=50, Adam, no clip). It tops accuracy while staying compute-efficient on CPU. BiLSTM attains similar F1 but slightly lower accuracy and around 2x time, its extra capacity doesn't translate to a net win in this budget.

- Vanilla RNN benefits from ReLU. RNN-ReLU is competitive (0.746/0.750), while tanh and sigmoid underperform due to saturation/vanishing gradients on long text. Even with ReLU, gating in LSTM remains more effective for capturing long range dependencies.
- Adam and RMSProp are strong out-of-the-box at lr=1e-3, their scores are very close. SGD is weak here (0.52/0.46) because OFAT fixes hyperparameters, no tuned momentum or LR schedule.
- Gradient clipping is not default. It helps with unstable combos (RNN-tanh improves from 0.619->0.710 Accuracy, BiLSTM nudges up with lower loss). For the already stable LSTM baseline, clipping slightly hurts accuracy by damping useful large updates.
- Compute/performance trade-off favors L=100. Epoch time roughly doubles from L=50 to L=100, but the accuracy and loss gains justify it on this task. If constrained, L=50 is the best value, otherwise, L=100 offers the strongest calibrated performance.

# 7. Recommended Configuration

- **Best configuration:**
  - LSTM, seq_len=100, Adam, no clipping, embedding=100, hidden=64, layers=2, dropout=0.5.
  - Performance: Accuracy 0.8122, F1 0.8137, Loss 0.4244.
- **Best practical configuration (refinement outside OFAT):**
  - LSTM, seq_len=100, Adam (lr=1e-3), dropout=0.5, Early Stopping (patience≈3, min_delta≈1e-3), Weight Decay=1e-5.
  - In additional runs, this preserved top accuracy (~0.83) and reduced test loss, indicating improved calibration/generalization.

# 8. Reproducibility and Implementation

**Reproducibility:** All runs are made reproducible by fixing random seed 42 across the entire stack: PyTorch, NumPy, and Python's random module. We keep the preprocessing pipeline, data splits, and OFAT hyperparameters constant across experiments and record package versions in requirements.txt.

**Hardware:** Training was executed CPU only on MacBook Air (Apple M2, 8 GB RAM), no discrete GPU or accelerator was used.

# 9. Conclusion

Optimal under CPU only constraints: LSTM, sequence length = 100, Adam optimizer, no gradient clipping (embedding=100, hidden=64, 2 layers, dropout=0.5). On MacBook Air (Apple M2, 8 GB, CPU-only), this configuration yields the best overall performance in the OFAT matrix, Accuracy 0.8122, F1 0.8137, Test Loss 0.4244 and shows stable, monotonic training. Although it requires roughly 2x the per epoch time of L=50 (50.8 s vs 24.8 s), the absolute accuracy gain (+4.3 points) and substantial loss reduction (0.490->0.424) indicate

both better decision quality and better-calibrated probabilities, this trade-off is worthwhile even on CPU.

Architectural and optimizer alternatives do not change that conclusion. BiLSTM is slower on CPU yet does not exceed LSTM's accuracy, and RNN variants trail despite ReLU. RMSProp approaches Adam but offers no accuracy or speed advantage, while SGD (lr=1e-3) fails to learn effectively in our fixed-hyperparameter setting. Gradient clipping is unnecessary for the LSTM baseline (it slightly harms accuracy), though it can stabilize RNN-tanh or BiLSTM if used.