Assignment 3

1. Dataset Summary

The IMDb Movie Review Dataset is used as directed. With 50,000 reviews, it is a common benchmark for binary sentiment classification. The dataset was divided into 25,000 training reviews and 25,000 testing reviews in line with the guidelines.

The tensorflow.keras.datasets.imdb.load_data utility was used for all preprocessing and data loading. The necessary preprocessing steps are effectively carried out by this function:

- **Vocabulary:** The dataset is automatically limited to the 10,000 most frequently occurring words.
- **Tokenization:** Every review is transformed into a series of integer token IDs.
- **Special Tokens:** It sets aside particular IDs for words that are out of vocabulary (2), start-of-sequence (1), and padding (0).
- **Sequencing:** To test their effects on performance, each sequence was either truncated or padded (using the padding token 0) to fixed lengths of 25, 50, and 100 words for the experiments.

2. Model Configuration

To guarantee a controlled and fair comparison, a standard set of hyperparameters was used when building each model. Only the particular variables under test (architecture, optimizer, etc.) were altered.

- **Embedding Layer:** 100 dimensions (padding_idx=0)
- **Recurrent Layers:** 2 hidden layers
- **Hidden Size:** 64
- **Dropout:** 0.4 (applied to embedding layer and between recurrent layers)
- **Batch Size:** 32
- **Output Layer:** A single fully-connected linear layer
- **Loss Function:** BCEWithLogitsLoss (which combines a Sigmoid activation and Binary Cross-Entropy loss for better numerical stability)
- **Hardware:** All experiments were executed on a single A100 GPU.

3. Comparative Analysis

To systematically evaluate the effects of various architectures, optimizers, sequence lengths, and the application of gradient clipping, a total of 72 experiments were carried out.
Note: The "sigmoid" activation for the vanilla RNN was skipped. For its nonlinearity argument, the PyTorch nn.RNN module only accepts relu and tanh. This is a design decision because

deep RNNs suffer from serious vanishing gradient issues due to the sigmoid's output range of 0 to 1. The 18 experiments involving this invalid combination were therefore not conducted. Also we don't have an activation variation for the lstm and bilstm models since they have a fixed architecture.

The full results table is in the **Appendix**. Below is a summary of the best and worst-performing models from that data.

Top 5 Performing Configurations (by Test F1-Score)

model	activation	optimizer	seq_len	clipping	test_accuracy	test_f1	avg_epoch_time_s
bilstm	n/a	rmsprop	100	True	0.804707	0.799352	9.020119
bilstm	n/a	adam	100	True	0.803828	0.798379	9.069477
bilstm	n/a	rmsprop	100	False	0.801031	0.795613	8.635333
lstm	n/a	rmsprop	100	False	0.796196	0.790172	7.597136
lstm	n/a	adam	100	True	0.793518	0.787670	7.963705

Worst 5 Performing Configurations (by Test F1-Score)

model	activation	optimizer	seq_len	clipping	test_accuracy	test_f1	avg_epoch_time_s
rnn	tanh	adam	25	False	0.532249	0.418745	6.845202
rnn	tanh	adam	100	False	0.512348	0.445906	7.352975
bilstm	n/a	sgd	100	True	0.495604	0.464657	8.812227
bilstm	n/a	sgd	25	False	0.495245	0.465578	7.745394
bilstm	n/a	sgd	25	True	0.495245	0.465578	8.043204

Plots

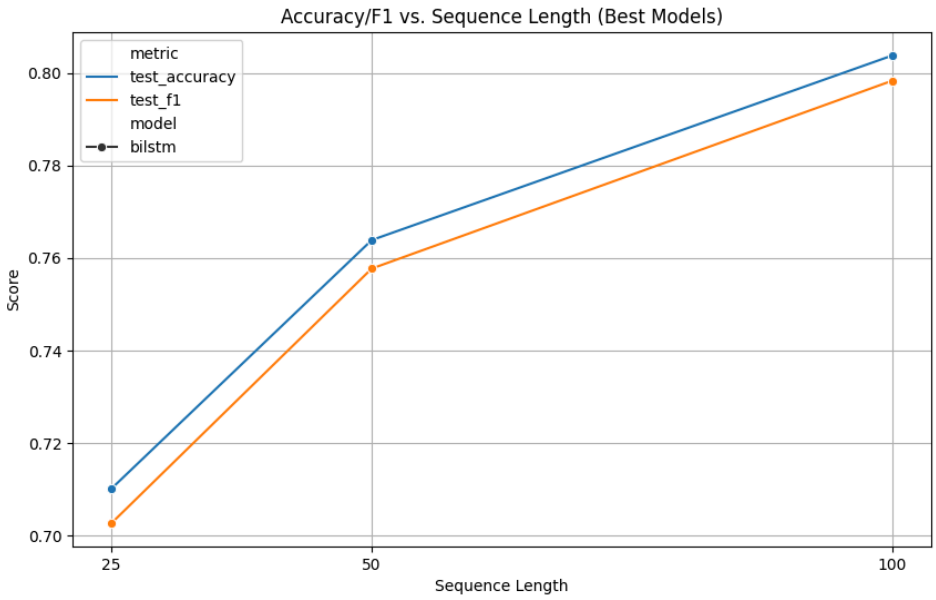

Figure 1: The effect of sequence length on the top-performing model class (BiLSTM with Adam and clipping) is displayed in this chart. Sequence length clearly and favorably correlates with test accuracy and F1-score, with 100 words outperforming 25.

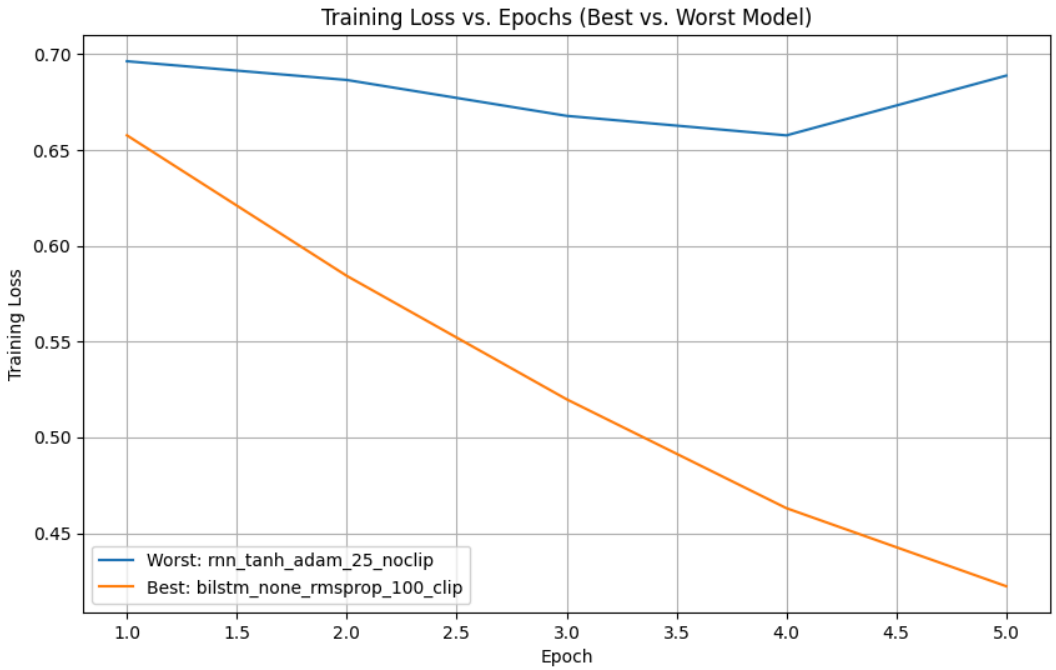

Figure 2: An analysis of the training loss curves for one of the worst-performing models (rnn_tanh_adam_25_noclip) and the best-performing model (bilstm_none_rmsprop_100_clip). The optimal model learns rapidly and steadily reduces its loss. The worst model collapses when it is unable to learn, as evidenced by its loss fluctuating and even increasing.

4. Discussion

Which configuration performed best?

A bidirectional LSTM with a sequence length of 100 that used gradient clipping and the RMSProp optimizer performed the best. The test F1-score for this model was 0.799352.
I think that this outcome makes sense because the BiLSTM architecture has a greater understanding of contextual information because it can process the sequence both forward and backward. The most information was probably contained in the 100-word length, and RMSProp was a very good optimizer for this task.

How did sequence length or optimizer affect performance?

- **Sequence Length:** Figure 1 shows how sequence length significantly improved results.

As the length increased from 25 to 50 to 100, performance steadily improved. This strongly suggests too much information was being truncated by the shorter 25- and 50-word sequences, leaving out important portions of the reviews that were necessary to assess sentiment.

- **Optimizer:** The most crucial element in determining success or failure was the optimizer.
    - SGD, was a total failure. All SGD models (RNN, LSTM, and BiLSTM) performed at the level of random guessing (~0.5 accuracy, ~0.47 F1), as can be observed in the Worst 5 table.. They were unable to converge.
    - **Adam** was mixed and appears in both the best and worst tables suggesting that it didn't have a very strong impact and other factors played a more imp role. RMSProp was quite great and is in the best-performing BiLSTM model.

How did gradient clipping impact stability?

Gradient clipping provided a clear and crucial stability boost, especially for the vanilla RNN models.

- The effect on RNNs was significant. The rnn_tanh_adam_25 model with clipping, for instance, obtained an F1-score of 0.645. A similar "worst model" example can be found in Figure 2, where the identical model without clipping collapsed and received a score of just 0.418. This illustrates how clipping effectively avoided the exploding gradients that affected the vanilla RNN.
- Since LSTMs' internal gates already aid in regulating gradient flow, the effect was less pronounced for LSTM/BiLSTM. Nonetheless, the data indicates a slight but steady advantage. Clipping improved the performance of the top two models (bilstm_rmspmrop_100 and bilstm_adam_100).

5. Conclusion

The optimal configuration identified through this systematic evaluation is:

- **Architecture:** Bidirectional LSTM (BiLSTM)
- **Optimizer:** RMSProp
- **Sequence Length:** 100
- **Stability:** Gradient Clipping Enabled

This is optimal in CPU also as it does not require any extra compute which cannot work without GPU. This setup makes sense because it uses gradient clipping to guarantee stable training, the most robust optimizer (RMSProp), the most complete data representation (100-word sequences), and the strongest architecture (BiLSTM). The experiments clearly show that a complete failure to learn results from either a stable architecture (such as an RNN without clipping) or a poor optimizer (such as using SGD).

Appendix: Full Experimental Results (72 Runs)

ID	model	activation	optimizer	seq_len	clipping	test_accuracy	test_f1	avg_epoch_time_s
70	bilstm	n/a	rmsprop	100	True	0.804707	0.799352	9.020119
58	bilstm	n/a	adam	100	True	0.803828	0.798379	9.069477
71	bilstm	n/a	rmsprop	100	False	0.801031	0.795613	8.635333
53	lstm	n/a	rmsprop	100	False	0.796196	0.790172	7.597136
40	lstm	n/a	adam	100	True	0.793518	0.787670	7.963705
59	bilstm	n/a	adam	100	False	0.786245	0.778574	8.720498
52	lstm	n/a	rmsprop	100	True	0.782769	0.774882	7.831722
41	lstm	n/a	adam	100	False	0.778812	0.768594	7.653851
57	bilstm	n/a	adam	50	False	0.765305	0.759093	8.165924
56	bilstm	n/a	adam	50	True	0.763907	0.757743	8.481594
69	bilstm	n/a	rmsprop	50	False	0.762748	0.756191	8.074755
68	bilstm	n/a	rmsprop	50	True	0.762348	0.755953	8.358449
50	lstm	n/a	rmsprop	50	True	0.755275	0.748378	7.460532
39	lstm	n/a	adam	50	False	0.752198	0.745654	7.306907
38	lstm	n/a	adam	50	True	0.752318	0.744230	7.635160
8	rnn	relu	adam	100	True	0.741848	0.734226	7.616652
51	lstm	n/a	rmsprop	50	False	0.743366	0.731502	7.221487
32	rnn	relu	rmsprop	100	True	0.729060	0.715731	7.602648
37	lstm	n/a	adam	25	False	0.712116	0.705345	7.228234
55	bilstm	n/a	adam	25	False	0.711077	0.703717	8.107926
36	lstm	n/a	adam	25	True	0.710758	0.703667	7.500350
54	bilstm	n/a	adam	25	True	0.710158	0.702675	8.456016
66	bilstm	n/a	rmsprop	25	True	0.707241	0.699819	8.309939
67	bilstm	n/a	rmsprop	25	False	0.707161	0.699555	7.940746
48	lstm	n/a	rmsprop	25	True	0.704723	0.697102	7.378399
49	lstm	n/a	rmsprop	25	False	0.704524	0.696545	7.155687
33	rnn	relu	rmsprop	100	False	0.701287	0.689039	7.199107
4	rnn	relu	adam	50	True	0.695053	0.688011	7.225582
28	rnn	relu	rmsprop	50	True	0.687540	0.680118	7.191071
0	rnn	relu	adam	25	True	0.678469	0.670246	7.149034
29	rnn	relu	rmsprop	50	False	0.677310	0.670220	6.868122
1	rnn	relu	adam	25	False	0.677190	0.669526	6.886120

24	rnn	relu	rmsprop	25	True	0.670197	0.662748	7.048573
25	rnn	relu	rmsprop	25	False	0.669158	0.661069	6.814211
6	rnn	tanh	adam	50	True	0.654811	0.646640	7.175586
2	rnn	tanh	adam	25	True	0.655291	0.645118	7.119918
5	rnn	relu	adam	50	False	0.660965	0.641271	6.972806
26	rnn	tanh	rmsprop	25	True	0.640785	0.628061	7.088592
7	rnn	tanh	adam	50	False	0.613491	0.605650	6.933697
30	rnn	tanh	rmsprop	50	True	0.621324	0.602095	7.111173
9	rnn	relu	adam	100	False	0.600464	0.592085	7.328365
31	rnn	tanh	rmsprop	50	False	0.610654	0.582051	6.886679
34	rnn	tanh	rmsprop	100	True	0.583680	0.566635	7.468638
27	rnn	tanh	rmsprop	25	False	0.577885	0.563782	6.745072
35	rnn	tanh	rmsprop	100	False	0.572770	0.562092	7.319667
10	rnn	tanh	adam	100	True	0.586277	0.539253	7.590454
43	lstm	n/a	sgd	25	False	0.506154	0.496784	6.998377
42	lstm	n/a	sgd	25	True	0.506154	0.496784	7.209654
22	rnn	tanh	sgd	100	True	0.506354	0.495029	7.330070
23	rnn	tanh	sgd	100	False	0.506314	0.494979	7.035812
45	lstm	n/a	sgd	50	False	0.503916	0.494822	7.118698
44	lstm	n/a	sgd	50	True	0.503916	0.494822	7.369469
15	rnn	tanh	sgd	25	False	0.501838	0.493886	6.594257
14	rnn	tanh	sgd	25	True	0.501838	0.493847	6.875922
12	rnn	relu	sgd	25	True	0.502038	0.491023	6.854023
13	rnn	relu	sgd	25	False	0.501798	0.490734	6.544710
20	rnn	relu	sgd	100	True	0.506074	0.490596	7.297030
21	rnn	relu	sgd	100	False	0.506114	0.490496	7.031587
18	rnn	tanh	sgd	50	True	0.498282	0.488365	7.031148
19	rnn	tanh	sgd	50	False	0.498322	0.488335	6.821027
17	rnn	relu	sgd	50	False	0.496324	0.485778	6.685876
16	rnn	relu	sgd	50	True	0.495964	0.485508	7.011561
47	lstm	n/a	sgd	100	False	0.496563	0.483667	7.491513
46	lstm	n/a	sgd	100	True	0.496563	0.483667	7.667188
63	bilstm	n/a	sgd	50	False	0.499920	0.468963	7.876525
62	bilstm	n/a	sgd	50	True	0.499920	0.468963	8.195544
61	bilstm	n/a	sgd	25	False	0.495245	0.465578	7.745394
60	bilstm	n/a	sgd	25	True	0.495245	0.465578	8.043204
65	bilstm	n/a	sgd	100	False	0.495604	0.464657	8.499188
64	bilstm	n/a	sgd	100	True	0.495604	0.464657	8.812227
11	rnn	tanh	adam	100	False	0.512348	0.445906	7.352975
3	rnn	tanh	adam	25	False	0.532249	0.418745	6.845202