IMDb RNN Experiments — Summary

Configuration

Data: cleaned, tokenized, top 10k vocabulary from IMDb 50k with a predetermined 25k/25k split; IDs padded to {25, 50, 100}.

Model Template: Embedding (100) -> {RNN/LSTM/BiLSTM, 2 layers, hidden=64, dropout=0.5} -> mean-pool activation -> FC(1) is the model template.

Loss: BCEWithLogits; Device=CPU; Batch=32; Epochs=5; Seed=1337.

Metrics: Macro-F1, accuracy, and training time/epoch.

Results

	model	activation	optimizer	seq_len	grad_clip	epochs	final_test_loss	accuracy	f1_macro	epoch_time_s(last_train)	device	seed
0	RNN	relu	ADAM	50	off	5	0.480314	0.76792	0.767447	1.6807	cpu	1337
1	LSTM	relu	ADAM	50	off	5	0.490834	0.76404	0.764038	6.3161	cpu	1337
2	BILSTM	relu	ADAM	50	off	5	0.487694	0.76188	0.761685	13.6662	cpu	1337
3	LSTM	sigmoid	ADAM	50	off	5	0.508847	0.75428	0.753633	6.6786	cpu	1337
4	LSTM	relu	ADAM	50	off	5	0.481788	0.76440	0.763800	6.5080	cpu	1337
5	LSTM	tanh	ADAM	50	off	5	0.484845	0.76512	0.764925	6.0606	cpu	1337
6	LSTM	relu	ADAM	50	off	5	0.488589	0.75704	0.757032	6.4766	cpu	1337
7	LSTM	relu	SGD	50	off	5	0.692958	0.49916	0.334303	6.0823	cpu	1337
8	LSTM	relu	RMSPROP	50	off	5	0.486625	0.75968	0.757985	6.1206	cpu	1337
9	LSTM	relu	ADAM	25	off	5	0.559067	0.70484	0.704575	3.1427	cpu	1337
10	LSTM	relu	ADAM	50	off	5	0.484411	0.76384	0.763437	6.2108	cpu	1337
11	LSTM	relu	ADAM	100	off	5	0.396150	0.82272	0.822396	12.3214	cpu	1337
12	LSTM	relu	ADAM	50	off	5	0.501528	0.75160	0.750061	6.2651	cpu	1337
13	LSTM	relu	ADAM	50	1.0	5	0.493700	0.76196	0.760032	6.3978	cpu	1337

What worked the best

Overall best: LSTM, ReLU, Adam, seq_len=100 => Acc 0.823, F1= 0.822, approx. 12.32 s/epoch.
In **five epochs**, this configuration is the only one that pushes >0.82 on CPU.

Factor-by-factor effects (changing one variable at a time)

Architecture (no clipping, Adam, ReLU, seq_len=50)
RNN: Acc=0.768 / F1=0.767 / approx. 1.68 s/epoch (fastest).
LSTM: approx. 6.32 s, Acc 0.764, and F1=0.764.
BiLSTM: Acc=0.762, F1=0.762, approx. 13.67 s (almost 2 x slower; no gain).
Conclusion: LSTM is equivalent to RNN on short sequences (50), however BiLSTM's additional expense was ineffective.

Activation (len=50, LSTM, Adam)

ReLU: Acc=0.764/F1=0.764 (baseline).
Sigmoid: F1=0.754 (lower) and Acc=0.754.
Tanh: Acc=0.765/F1=0.765 (equivalent to ReLU).
Conclusion: Sigmoid performs poorly as a hidden activation here, although ReLU/tanh are safe.

Optimizer (len=50, LSTM, ReLU)

Adam: F1=0.764/ Acc=0.764 (steady).
RMSProp: somewhat worse Acc=0.760/F1=0.758.
SGD (default LR): Acc=0.499 / F1=0.334 (collapsed).
Conclusion: Adam is the ideal plug-and-play option. The SGD run probably utilized an LR that was too small for this configuration; it would be competitive with a tweaked LR (e.g., 0.01).

Length of sequence (LSTM, ReLU, Adam)

len=25: Acc=0.705/F1=0.705/3.14 s.
len=50: Acc=0.764 / F1=0.763 / 6.21 s.
len=100: F1=0.822 / Acc=0.823 / 12.32 s.
Conclusion: Cost scales roughly linearly with length; accuracy/F1 clearly improves with greater context.

Gradient clipping (len=50, LSTM, ReLU, Adam)

Off: F= 0.763, Acc= 0.764.
max_norm=1.0: F= 0.760, Acc=0.762.
Conclusion: In this stable regime (5 epochs, CPU), clipping did not significantly alter the results.

Observations during training

At len=50, BiLSTM ≈ 2× LSTM time.
With significant accuracy gains, doubling seq_len (50->100) almost doubles epoch time (6.2->12.3 s).
At larger lengths, accuracy lags behind LSTM, although plain RNN is the fastest.

Recommendations

i) LSTM + ReLU + Adam is the default for reports and downstream tasks.
ii) Use seq_len=100 and keep epochs small (>=5) if you want the highest CPU accuracy.
iii) Seq_len=50 is a nice balance (approx. 6 s/epoch) with approx. 0.764 F1 if you need quick iterations.
iv) Here, stay away from BiLSTM unless you can take use of the additional expense (longer training, GPU, etc.).
v) ReLU or tanh are preferable to sigmoid hidden activations.
vi) LR tweaking is necessary for SGD; it collapsed with default settings. Rerun SGD with lr equivalent to 1e-2 and momentum 0.9 if you need to compare fairly.