State-Space Models for Long-Range Dependency Tasks: Sentiment Analysis

Pallekonda Naveen Kumar
Indian Institute of Science
SR No: 22915
kpnaveen@iisc.ac.in

Abstract

State-Space Models (SSMs) have emerged as a promising alternative to traditional sequence modelling architectures particularly for long-range dependency tasks. This report explores the application of SSMs to *Sentiment Analysis*, benchmarking their performance against standard baselines focusing on Long Short-Term Memory (LSTM)[4], Convolutional Neural Networks (CNN), Bidirectional Encoder Representations from Transformers (BERT)[1], Generative Pre-trained Transformer (GPT)[6], and the innovative Mamba[3] model. Their performance was compared on the Twitter US airline dataset[2] and the IMDB movie review dataset[5]. Through this comparative analysis, we aim to highlight the strengths and limitations of each model, particularly in their ability to efficiently handle complex language patterns and varying sequence lengths. The code is available at GitHub Link.

1 Introduction

Sentiment analysis, Various models have been developed, including traditional machine learning approaches like Naive Bayes and SVMs, and advanced deep learning techniques such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM), transformers, and the innovative Mamba model. LSTMs are effective for short to medium-length sequences, but struggle with long sequences, while CNNs excel at identifying local patterns but are limited in capturing long-range dependencies. Transformers represent a significant advancement, addressing many shortcomings of LSTMs and CNNs, but require substantial memory and computational resources. Mamba, a novel State Space Model, offers greater flexibility and efficiency, particularly for long-sequence tasks, and is a promising alternative to traditional models.

2 Problem Statement

Large sequence modelling poses significant challenges for common architectures:

- **RNNs**: Struggle with vanishing gradients and sequential processing, with time complexity $O(n)$. The hidden state update $h_t = f(W_h h_{t-1} + W_x x_t + b)$, fails to preserve long-term dependencies as $|W_h| < 1$.
- **CNNs**: Limited receptive field for long sequences, requiring deeper stacks of layers. For a sequence of length n, the receptive field grows as Receptive Field $\propto k \cdot d$ where k is the kernel size and d is depth.
- **Transformers**: Effective for long-range dependencies but scale poorly due to quadratic complexity $O(n^2)$ in self-attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V.$$

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

Mamba's selective scan mechanism enables context-dependent reasoning, and its hardware-aware optimizations exploit GPU parallelization to handle sequences of varying lengths without padding.

- **State Equation:**
$$\frac{d\mathbf{h}(t)}{dt} = A\mathbf{h}(t) + B\mathbf{u}(t),$$
where $\mathbf{h}(t)$ is the hidden state, A governs dynamics, and B maps the input $\mathbf{u}(t)$.
- **Output Equation:**
$$\mathbf{y}(t) = C\mathbf{h}(t) + D\mathbf{u}(t).$$
- **Efficiency:** mitigates the vanishing gradient problem through orthogonal matrix constraints and scales linearly with sequence length, making it particularly suitable for long-sequence tasks:
$$\mathbf{y}(t) = \int_0^t K(t-\tau)\mathbf{u}(\tau)d\tau,$$
where $K(t)$ is precomputed.

The objective is to investigate their effectiveness in handling long sequences, computational efficiency, and generalization capabilities across datasets of different complexities, such as the Twitter US Airline and IMDB Movie Review datasets.

3 Experiement Setup

3.1 Datasets

Twitter US Airline Dataset[2]: containing 14,640 samples categorised as 3,099 neutral, 2,363 positive, and 9,178 negative sentiments. It was split into 70% training, 15% validation, and 15% testing(sequence lengths were up to 45).

IMDB Movie Review Dataset[5]: Designed for binary sentiment classification, it consists of 50,000 reviews (25,000 for training and 25,000 for testing). Reviews vary significantly in length(up to 1400 sequence length), posing realistic challenges for NLP models.

3.2 Experiment Settings

- All sequences were padded/truncated to a maximum length of 512 tokens.
- **Models:** LSTM and CNN trained from scratch; BERT-base, GPT-2-small, and Mamba-130M initialized with pre-trained weights (From HuggingFace).
- **Optimizer:** AdamW with a weight decay of 0.01 and a cosine annealing learning rate scheduler.
- **Epochs:** Trained for 50 epochs with early stopping applied if validation loss did not improve for 5 epochs.
- **Dropout:** 0.1 before the final classification layer.

3.3 Evaluation Metrics:

Accuracy, Precision, Recall, F1-score,Training and evaluation time.

3.4 Cross Entropy Loss Function:

where y_{true} is the true label and y_{pred} is the predicted probability. Loss = $-\sum(y_{true} * \log(y_{pred}))$

3.5 Fine Tuning and Model Hyperparameters

Pretrianed LLMs (Mamba, BERT, GPT): Initial learning rate = 3×10^{-6}

Experimented with many learning rates on each model with a different dataset.

Batch size determination: Tested powers of 2 (1, 2, 4, 8, ...) until GPU memory limits were reached

Model	Parameters	Layers	Dimension
Mamba-130M	129M	24	768
GPT-2-small	124M	12	768
BERT-base	109M	12	768
LSTM	33M	2	768
CNN	24M	3	768

Table 1: Model parameter settings.

4 Results

A batch size of 8 was used to ensure a fair comparison between models. The performance metrics for the five models—Mamba, GPT, BERT, LSTM, and CNN—are summarized in Table 2, 3.

Metric	Mamba	GPT	BERT	LSTM	CNN
Test Accuracy (%)	82.3	82.0	83.9	76.2	78.6
F1-Score (%)	82.2	81.8	83.8	75.6	77.5
Recall (%)	82.3	82.0	83.9	76.2	78.6
Precision (%)	82.1	81.8	83.8	75.3	78.6
Learning Rate	3×10^{-6}	3×10^{-6}	9×10^{-7}	3×10^{-5}	3×10^{-5}
Training Time per Epoch (min)	1.14	1.94	0.68	0.121	0.075
Evaluation Time per Epoch (min)	1.12	0.07	0.07	0.015	0.015
Total Training Time (min)	10.24	11.65	5.46	1.57	0.91

Table 2: Model Performance on the Twitter US Airline Dataset.

Metric	Mamba	GPT	BERT	LSTM	CNN
Test Accuracy (%)	93.6	92.8	93.9	86.3	89.9
F1-Score (%)	93.6	92.8	93.9	86.3	89.9
Precision (%)	93.6	92.8	93.9	86.3	89.9
Recall (%)	93.6	92.8	93.9	86.3	89.9
Learning Rate	5×10^{-5}	3×10^{-7}	2×10^{-5}	1×10^{-5}	1×10^{-5}
Training Time per Epoch (mins)	15.39	24.61	20.4	3.53	1.14
Evaluation Time per Epoch (mins)	4.63	6.22	5.34	1.35	0.36
Total Training Time (mins)	354.08	418.48	163.2	35.34	25.48

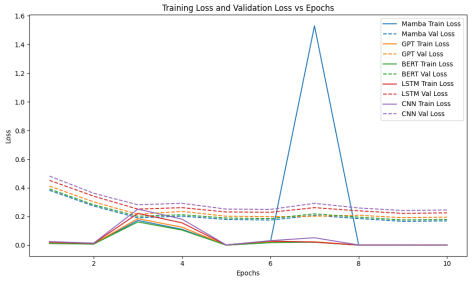
Table 3: Model Performances on IMDB Movie Review Dataset.

5 Analysis and Discussions of Results:

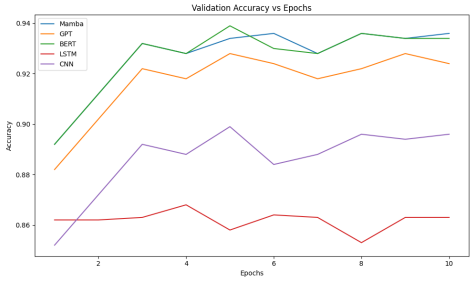
5.1 Scaling: Context Length

Investigate the scaling properties of models with respect to sequence length. We only compare the LSTM, CNN, BERT, GPT and Mamba models, as quadratic attention becomes prohibitively expensive at longer sequence lengths. We pretrain models on sequence lengths ranging from 1 to 1400. we fix the 129M model and try different embed dimensions, learning rates, and batch sizes.(More plots in the github)

Results. Tables 2, 3 shows that **Mamba is able to make use of a longer context even up to extremely long sequences**, and its testing accuracies increase as the context increases(difference with BERT decreases). On the other hand, the LSTM, CNN model gets worse with sequence length(difference to BERT). This is intuitive from the discussion in Section 2 on properties of the selection mechanism. In particular, from a convolutional perspective, a very long convolution kernel aggregates all information across a long sequence which may be very noisy.



(a) Training and Validation loss vs Epochs



(b) Validation Accuracy vs Epochs

Figure 1: Loss and Accuracy(taken upto 10 epochs irrespective of optimality of epochs for comparision

5.2 Fast Training and Inference

Even though the training of RNNs, CNNs are fast but have less test accuracies. The total time of Mamba is high because the optimality of epochs is higher but as the sequence increases each epoch training is lesser than the other LLMs and accuracies are almost high. since they require a pass over the full context for every new sample(also for shorter sequences). As a stateful model, SSMs automatically have this ability. By switching to its recurrent representation, It requires constant memory and computation per time step – in contrast to standard autoregressive models which scale in the context length.

5.3 Trade-Offs in Speed, Accuracy, and Resource Efficiency

- **Training and Evaluation Speed:** In comparison, Mamba, GPT, and BERT required significantly longer training times per epoch (15.39, 24.61 and 20.4 minutes, respectively). However, Mamba's efficient training for long-sequence tasks demonstrated an advantage over GPT, which required the highest total training time (418.48 minutes).
- **Accuracy and Generalization:** While CNN and LSTM were computationally efficient, they showed limited ability to generalize to complex datasets, as evidenced by their higher test losses and lower F1-scores. In contrast, BERT and Mamba demonstrated superior generalization and accuracy, making them ideal choices for tasks requiring high precision and recall.
- **Scalability and Flexibility:** Mamba's linear complexity and ability to handle sequences of varying lengths without padding make it a practical alternative to transformer-based models in memory-constrained environments. BERT and GPT, constrained by maximum input sequence lengths (512 and 1024 tokens, respectively), are less flexible in tasks involving highly variable input lengths.

6 Conclusion

The Mamba model, excelling in handling long sequences with linear complexity while maintaining competitive accuracy and generalization. Its efficient training and inference for variable-length sequences make it an ideal choice for long-context NLP tasks, particularly in resource-constrained environments.

Despite its slower total training time due to higher optimal epoch counts, Mamba outperformed traditional models like LSTM and CNN and demonstrated comparable performance to transformers. This positions Mamba as a scalable and flexible solution for long-sequence tasks.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186, 2018.

[2] Alec Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision, 2009. CS224N Project Report, Stanford University.

[3] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[5] Andrew Maas et al. Imdb dataset for binary sentiment classification, 2011. Stanford Sentiment Treebank.

[6] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. In *OpenAI Blog*, 2018.