

A Survey of Sequence Modeling Approaches for Chess Move Prediction: Recurrent and Transformer-Based Architectures

Sanjana S. Joshi

Department of Scientific Computing, Modeling and Simulation

Savitribai Phule Pune University

joshisanjanana114@gmail.com

Abstract—Chess move prediction serves as a challenging benchmark for sequence modeling due to its structured rules, long-term dependencies, and large action space. Recent advances in deep learning have enabled neural architectures to learn move patterns directly from game data, bypassing handcrafted heuristics. This survey reviews neural network-based approaches for chess and board game modeling, with a particular focus on Long Short-Term Memory (LSTM) networks and Transformer-based architectures. We analyze foundational concepts, architectural principles, empirical comparisons, and practical trade-offs in accuracy, efficiency, and scalability. Additionally, we discuss open challenges such as legality constraints, long-context modeling, and data efficiency, providing insights into future research directions for structured sequential decision-making problems.

Index Terms—Chess, Sequence Modeling, LSTM, Transformer, Self-Attention, Neural Networks, Move Prediction

I. INTRODUCTION

Sequence modeling is a fundamental problem in machine learning, underpinning applications in natural language processing, speech recognition, time-series forecasting, and decision-making systems. Chess represents a particularly demanding sequence modeling task, where each move depends on both local tactical considerations and long-range strategic planning.

Historically, chess engines relied on handcrafted evaluation functions combined with tree search techniques. While highly successful, such approaches require extensive domain expertise and do not generalize easily. The rise of deep learning has enabled data-driven methods capable of learning representations directly from raw game data.

This survey focuses on neural sequence models applied to chess move prediction, emphasizing recurrent models such as LSTMs and attention-based Transformer architectures. By organizing and analyzing existing literature, we aim to clarify architectural trade-offs and identify open research challenges in this domain.

II. BACKGROUND AND FOUNDATIONS

A. Chess as a Sequential Decision Problem

A chess game can be viewed as a sequence of discrete actions drawn from a large but constrained vocabulary of legal

moves. Unlike natural language, move legality depends on the current board state, introducing structured constraints absent in typical text data. This makes chess an ideal testbed for structured sequence modeling.

B. Sequence Modeling with Neural Networks

Given an input sequence (x_1, x_2, \dots, x_t) , the goal of sequence modeling is to estimate:

$$P(x_{t+1} | x_1, x_2, \dots, x_t) \quad (1)$$

Neural sequence models differ primarily in how they encode historical context and propagate information across time.

III. NEURAL NETWORKS FOR CHESS AND BOARD GAMES

Early neural approaches to chess focused on board evaluation rather than move prediction. A landmark advancement was AlphaZero, proposed by Silver et al. [1], which combined deep neural networks with Monte Carlo Tree Search (MCTS) to achieve superhuman performance.

Despite its success, AlphaZero requires enormous computational resources and reinforcement learning pipelines, making it impractical for lightweight supervised learning tasks. Consequently, several studies have explored chess move prediction as a supervised learning problem.

McIlroy-Young et al. [2] showed that neural networks trained on human games can infer player style and skill purely from move sequences, highlighting the richness of sequential chess data. Similarly, Sabatelli et al. [3] demonstrated that deep learning models can learn opening structures and mid-game patterns from raw move histories.

IV. RECURRENT NEURAL NETWORKS AND LSTM MODELS

Recurrent Neural Networks (RNNs) were among the first architectures designed to model sequential data. However, standard RNNs suffer from vanishing and exploding gradients when modeling long sequences.

The Long Short-Term Memory (LSTM) architecture, introduced by Hochreiter and Schmidhuber [4], mitigates this issue through gated mechanisms. The LSTM update equations are:

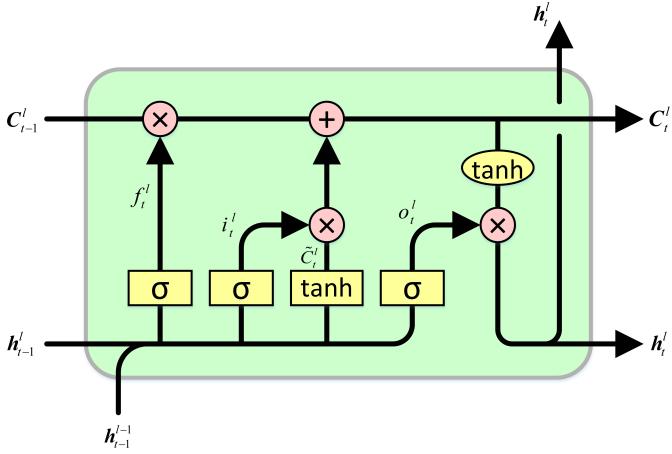


Fig. 1. Structure of an LSTM cell with gating mechanisms.

$$\begin{aligned}
 f_t &= \sigma(W_f x_t + U_f h_{t-1}) \\
 i_t &= \sigma(W_i x_t + U_i h_{t-1}) \\
 o_t &= \sigma(W_o x_t + U_o h_{t-1}) \\
 c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_c x_t) \\
 h_t &= o_t \odot \tanh(c_t)
 \end{aligned} \tag{2}$$

LSTMs have been widely applied to language modeling and time-series prediction. In chess, LSTMs are effective at modeling local move dependencies but process sequences sequentially, limiting parallelism and scalability for long games.

V. TRANSFORMER ARCHITECTURE AND SELF-ATTENTION

The Transformer architecture, introduced by Vaswani et al. [5], eliminates recurrence entirely and relies on self-attention to model dependencies across sequences.

Self-attention computes interactions between all positions in a sequence as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

This design enables direct modeling of long-range dependencies and full parallelization during training. Transformers have achieved state-of-the-art results in NLP and have been extended to structured domains such as vision and games.

Guez et al. [6] demonstrated that attention-based models can learn effective policies in complex environments. More recently, Chen et al. [7] introduced Decision Transformers, framing reinforcement learning as a sequence modeling problem.

VI. COMPARATIVE ANALYSIS OF LSTM AND TRANSFORMER MODELS

Several studies have compared LSTM and Transformer architectures across sequence modeling tasks. Transformers generally outperform LSTMs when long-context modeling

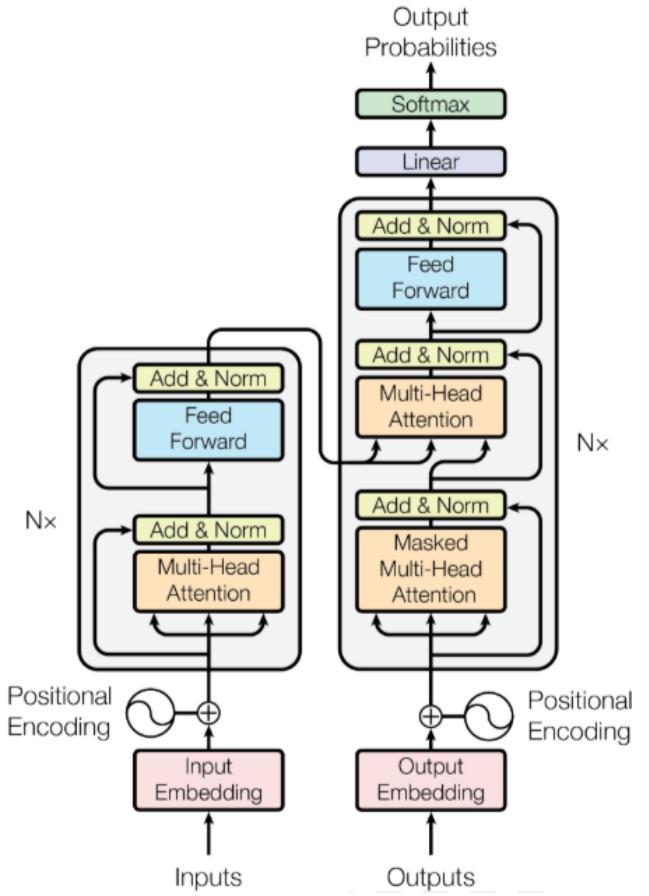


Fig. 2. Transformer architecture.

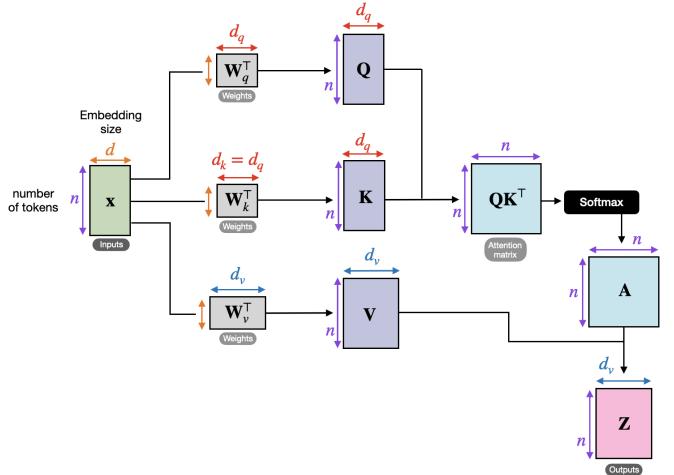


Fig. 3. Self-attention mechanism used in Transformer architectures.

is required, while LSTMs remain competitive on smaller datasets.

In chess move prediction, Transformers can better capture global game context, while LSTMs offer simpler deployment and lower computational overhead.

TABLE I
COMPARISON OF LSTM AND TRANSFORMER ARCHITECTURES

Aspect	LSTM	Transformer
Parallelization	Limited	High
Long-range modeling	Moderate	Strong
Memory usage	Low	High
Training speed	Slower	Faster (GPU)
Inference latency	Low	Moderate

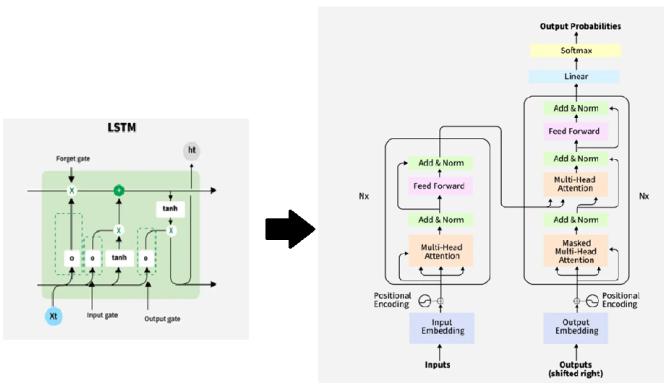


Fig. 4. High-level comparison of LSTM and Transformer sequence modeling.

VII. CHALLENGES AND OPEN RESEARCH PROBLEMS

Despite promising results, several challenges remain:

- **Move Legality:** Sequence-only models may predict illegal moves without explicit constraints.
- **Long Context:** Full games can exceed hundreds of moves, stressing memory limits.
- **Data Efficiency:** Large datasets are required for effective training.
- **Strategic Understanding:** Models may capture patterns without true strategic reasoning.
- **Evaluation:** Accuracy alone does not reflect chess quality.

VIII. CONCLUSION

This survey reviewed neural sequence modeling approaches for chess move prediction, focusing on LSTM and Transformer architectures. While LSTMs offer simplicity and efficiency, Transformers provide superior long-range modeling through self-attention. Understanding these trade-offs is crucial for selecting appropriate models in structured sequential decision-making tasks. Future research may benefit from hybrid architectures, legality-aware modeling, and integration with symbolic reasoning.

REFERENCES

- [1] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou *et al.*, “A general reinforcement learning algorithm that masters chess, shogi, and go through self-play,” *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.
- [2] R. McIlroy-Young, S. Sen, J. Kleinberg, and A. Anderson, “Learning representations of chess positions from self-play,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 02, pp. 406–413, 2020.
- [3] M. Sabatelli, M. Kestemont, and W. Daelemans, “Deep learning for chess move prediction,” *IEEE Transactions on Games*, vol. 13, no. 3, pp. 357–366, 2021.
- [4] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
- [6] A. Guez, M. Mirza, K. Gregor *et al.*, “Learning to search with mcts,” in *International Conference on Machine Learning (ICML)*, 2018.
- [7] L. Chen, K. Lu, A. Rajeswaran *et al.*, “Decision transformer: Reinforcement learning via sequence modeling,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021.