Graph Neural Networks for Chess

Saleh Alwer

Supervised by:
Aske Plaat
Walter Kosters

Leiden University, The Netherlands

Abstract. This research introduces an application of Graph Neural Networks (GNNs) in chess modelling. The GNN model receives as input a graph representation of the chessboard, which depicts chess squares as nodes and legal moves as edges. This approach significantly outperforms traditional array-based Residual Networks in learning chess positions' policy and value. Policy prediction is particularly superior to traditional methods. This is attributed to the model's ability to learn policy along the edges, capturing the relational dynamics of chess positions, and learning over a significantly reduced action space. This stands in stark contrast to traditional methods that learn over the full 4672-move chess action space.

Keywords: Graph Neural Networks · Chess AI · Deep Learning

1 Introduction

This paper explores the use of Graph Neural Networks (GNNs), particularly the Graph Attention Network (GAT) [3] variant, for modeling chess. This domain is traditionally served by array-based neural networks as in AlphaZero [2] and Leela Chess Zero [1] utilizing Residual Networks (ResNets). The study aims to discern critical elements for the graph representation, propose an effective GNN architecture for chess analysis and contrast GNN performance to ResNets.

2 Methods

We utilize an encoding depicting the chessboard as a graph, with squares as nodes and potential moves as edges. Each node and edge encompass a feature vector detailing game specifics. We employ a dual-headed model (*GNN-graph*, detailed in Figure 1) that comprises of GATs to generate two distinct node embeddings. The first is aggregated and mapped to estimate the board state's value (value head). The second set of node embeddings is leveraged to form move embeddings through concatenation of source and target embeddings. Self-attention is applied to a combined global move representation. This is mapped down to a probability distribution over legal moves (policy head).

For comparison, we establish two other models: *ResNet*, a baseline model that maps an array-based chessboard representation to game policy and value

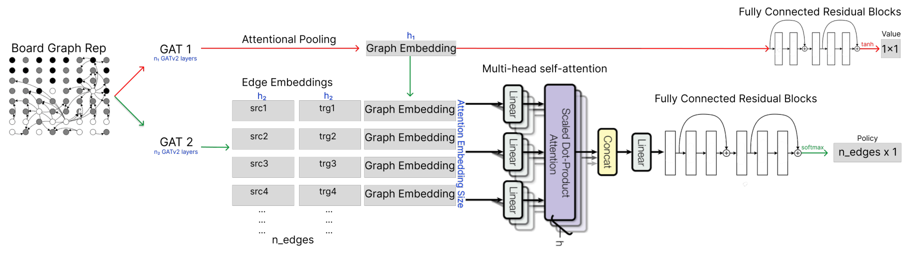

Fig. 1. This GNN model processes a graph representation of a chessboard through to GATs with hidden sizes h_1 and h_2 and number of layers n_1 and n_2. Edge-based move representations are formed by concatenating source (*src*) and target (*trg*) node embeddings. The final output is a distribution over legal moves.

in an AlphaZero-like architecture, and *GNN-array*, a model that uses a GAT to embed the board and mirrors *ResNet's* value and policy heads.

3 Results and Discussion

We conducted Hyperparameter Optimization which revealed an effective GNN configuration for chess modelling with hidden dimensions and attention embedding sizes being significant contributors to performance. Training utilized a dataset of 135K grandmaster games and 11M randomly generated positions. True policies and values are retrieved from the Stockfish chess engine. Throughout the training process, *GNN-graph* demonstrated the most stable learning trajectory. Test results, seen in Table 1, indicated superior performance of *GNN-graph* over alternative models.

Other notable findings include that *GNN-graph* outperforms the other models in head-to-head gameplay and successfully adapts to small, player-specific datasets through fine-tuning. Our results indicate that the proposed strategy employed for policy construction primarily accounts for the superior performance of the GNN-based model over the ResNet model.

Table 1. Comparison of test performance across the *GNN-graph*, *GNN-array*, and *ResNet* models. *GNN-graph* performs best in all metrics. Best Acc and Acc quantify the fractions of instances where the model's top prediction matches the top move from the true policy, and where the highest predicted move aligns with any non-zero move in the true policy, respectively.

Model	**MSE**	**Acc**	**Best Acc**
Resnet	0.32	0.27	0.11
GNN-graph	**0.20**	**0.51**	**0.42**
GNN-array	0.21	0.16	0.07

References

1. Pascutto, G.-C., Linscott, G.: Leela Chess Zero (2020), http://lczero.org/, [Online; accessed 7-June-2023]
2. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., Hassabis, D.: Mastering the game of Go without human knowledge. Nature, 550(7676), 354–359 (2017), https://www.nature.com/articles/nature24270
3. Brody, S., Alon, U., Yahav, E.: How Attentive are Graph Attention Networks? In: Tenth International Conference on Learning Representations (ICLR 2022), https://openreview.net/forum?id=F72ximsx7C1