H-Go

A Go AI build with PyTorch.

91d906h4 2024/04/14

Introduction

Go, known for its intricate rules and vast search space, has long stood as a formidable challenge for AI systems in the past. And H-Go leverages machine learning algorithms and simple decision-making strategies to achieve human performance levels.

Driven by a combination of Convolutional Neural Networks (CNN) and Monte Carlo Tree Search (MCTS), H-Go can play the game of Go with a remarkable degree of accuracy. Through iterative self-improvement and training on vast datasets, H-Go continuously refines its gameplay strategies, adapting to diverse playing styles and evolving board states.

Dataset

We leverage the extensive Fox Go Dataset as the foundational supervised learning dataset for H-Go. Comprising a 21.1 million 19x19 Go games gathered from diverse ranks on the Fox Go server from 2021 to 2019.

To ensure robust model training while maintaining computational efficiency, we judiciously select a representative subset of the Fox Go Dataset. Specifically, we utilize approximately 5% of the dataset, equating to approximately 100,000 games, to train our H-Go model.

Pre-Processing

In our approach, we employ a game-centric methodology to process the dataset. Each game within the dataset serves as a fundamental unit for data extraction and analysis.

In a typical Go game, players alternate between black and white stones, with the black side traditionally making the first move. To capture the sequential nature of gameplay and facilitate feature extraction, we employ a sliding window approach with a window size of 7, this window traverses the entirety of each game. The result is an array with a length equal to the total number of steps of this game, and the elements are arrays of length 7, representing the previous 7 step of the current move (label).

Furthermore, we adopt a predictive modeling framework wherein the subsequent move (i.e., the next step) following each sliding window serves

as the label for the corresponding input.

Model

H-Go utilizes a convolutional neural network (CNN) architecture augmented with residual structures as its underlying model. This architecture is designed to effectively capture the intricate patterns and features inherent in the game of Go.

The model outputs two key components: policy and value. The policy component provides insights into the most probable next move, guiding H-Go in its decision-making process. Meanwhile, the value component offers an estimation of the likelihood of winning based on the current game state.

The optimizer is Radam with default learning rate 1e-3, the loss function for policy and value network are CrossEntropyLoss and BCELoss, respectively.

HTS

HTS, a variant of Monte Carlo Tree Search (MCTS), is employed to determine the optimal move in a game scenario. Initially, HTS identifies the most probable moves based on the current game state, considering them as the children of the root node. These move probabilities are generated using a policy network. Subsequently, each of these selected moves is expanded by simulating subsequent moves until a total of 256 leaf nodes are reached. The values of all nodes within these expanded branches are then aggregated. Finally, the move with the highest aggregated value is selected as the next move.

References

[1] D. Silver et al., "Mastering the game of Go with deep neural networks and tree search," Nature, vol. 529, pp. 484–489, Jan. 2016.
[2] D. Silver et al., "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm," arXiv:1712.01815 [cs.AI], Dec. 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1712.01815

[3] Klein, "Alphago についてのまとめ," kleinblog, 06 February 2021. [Online]. Available: https://kleinblog.net/alphago

[4] T. Yamaoka, "AlphaGo Zero の論文を読む その 2(ネットワーク構成)," TadaoYamaoka の開発日記, 20 October 2017. [Online]. Available: https://tadaoyamaoka.hatenablog.com/entry/2017/10/20/221030

[5] T. Yamaoka, "将棋でディープラーニングする その 33(マルチタスク学習)," TadaoYamaoka の開発日記, 08 June 2017. [Online]. Available: https://tadaoyamaoka.hatenablog.com/entry/2017/06/08/000040
[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," arXiv:1512.03385 [cs.CV], Dec. 2015. [Online]. Available: https://arxiv.org/pdf/1512.03385.pdf
[7] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," arXiv:1603.05027 [cs.CV], Mar. 2016. [Online]. Available: https://arxiv.org/abs/1603.05027
[8] Deep Learning Training Challenges and Solutions: A Comprehensive Summary (OpenAI's ChatGPT, private communication, 10 April 2024).
[9] Featurecat, "Fox Go Dataset," GitHub. [Online]. Available: https://github.com/featurecat/go-dataset

[10] DOORS 編集部, "強化学習入門 Part3 － AlphaGoZero でも重要な技術要素！ モンテカルロ木探索の入門 －," Brainpad, Nov. 1, 2020. [Online].Available: https://www.brainpad.co.jp/doors/contents/01_tech_2018-04-05-163000/