Alpha Zero Improvements

Zhongyao Chu
New York University
zc1213@nyu.edu

Tianyu Lei
New York University
tl2441@nyu.edu

Lingxiao Gong
New York University
lg2897@nyu.edu

1. Introduction

The Alpha Zero, a well-known computer program with the ability to achieve superhuman level of play in go, chess and shogi within 24 hours where only game rules are provided, is based on the Monte-Carlo tree search (MCTS) algorithm using a single Deep Residual Network. Our goal for this project is trying to improve the Alpha Zero by introducing an ensemble of assorted neural networks.

After certain researches, we propose to build a neural network ensemble consisting of two non-identical Residual Networks and one Densely Connected Convolutional Network for the project and integrate them.

2. Objectives

- Understand basic idea of Alpha Zero, algorithm and structure behind it along with its training process.
- Figure out a proper combination of neural networks to be the neural network ensemble for our project.
- Integrate the new ensemble of neural networks with the original Alpha Zero program appropriately.
- Train the entire system and adjust parameters accordingly in order to achieve better performance.
- Evaluate our model by comparing its performance before and after neural network ensemble.

3. Neural Network Ensemble

In order to combine DenseNet and ResNet under various hyper-parameters, the ensemble part of model is required to combine the outputs (p_i^k, v_i^k) of neural network $f_i^k(s_i)$ with input s_i. We define the overall output (P_i, V_i) = Ens(p_i^1...p_i^k, v_k^1...v_i^k,s_i) where Ens() is the ensemble function we need to implement.

Bootstrap Aggregation

Feed k subsets from the training set into our system to get k different outputs from k different models, then make a majority vote among them.

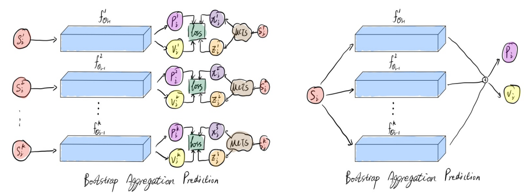

Figure 1. Bootstrap Aggregation Prediction

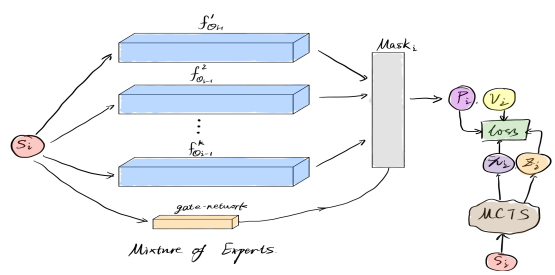

Figure 2. Bootstrap Aggregation

Mixture of Experts

A mask M = $[0, 1]^k$ generated by a gate neural network g(s_i) will be applied on the k outputs, which acts as a trainable weighted average operation.

4. Evaluation of Model

Above ensemble methods will be implemented, and we'll use Elo-rating to analyze the performance of model based on training time.

$$E_a = \frac{1}{1 + 10^{\frac{R_b - R_a}{400}}} \qquad R_a^{'} = R_a + K(S_a - E_a)$$

Elo-rating calculation is based on above equations (from the perspective of player a) where k is assigned value and $S_a \in (1.0, 0.5, 0)$. based on the output.
[5] [2] [3] [4] [1]

References

[1] G. Brown. *Diversity in neural network ensembles*. PhD thesis, University of Birmingham, 2004.

[2] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3, 2017.

[5] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.