AlphaGo Research Review

Isolation-Playing Agent through Adversarial Search

Introduction

The game of Go has long been viewed as the most challenging of classic games for artificial intelligence owing to its enormous search space(branching factor around 250 and search depth around 150) and the difficulty of evaluating board position and moves[1]. Google DeepMind tackles this grand challenge through an innovative combination of deep neural network(DNN) and Monte-Carlo tree search(MCTS).

Techniques

Policy Network. Three policy networks are trained to model the action distribution under given board state p(a|s). All these networks are deep convolutional networks with board state as 19-by-19 image as input, alternative convolution layer and ReLU activation layer as hidden layers and softmax layer for legal move distribution as output.

- The first policy network is a supervised learning(SL) one built from human expert move dataset of size 30 million.
- The second policy network is a linear softmax network on small feature subset.
- The third policy network is a reinforcement learning(RL) one initialized with SL one and optimized through self-play.

The first network is used for RL network initialization and final board evaluation function. The second and the third are used for final evaluation function.

Value Network. Another deep convolutional network is built for evaluating board position. To build training dataset, for each input board state the game is played under RL policy network until it ends. 30 million such input instances are collected from distinct games. The network achieves decent performance without state space search.

Final Value Function. The final evaluation function consists of two parts. The first part is merged board evaluation. This evaluation is a linear combination of value network output and terminal output played using the second policy network. The second part is random exploration score generated using the first policy network.

Final Policy Function. The final policy function is built upon MCTS using final value function. At each board state, the action with maximum expected utility is selected.

Results

AlphaGo achieved predominant winning rate of 99.8% against other Go programs and even defeated many human champions. Its grand achievement provides hope that human-level performance can now be achieved in other seemingly intractable artificial intelligence domains[1].

References

1. David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. Nature, Vol. 529, No. 7587. (27 January 2016), pp. 484-489