Summary of “Mastering the game of Go with deep neural networks and tree search”

Mingyi Zhang

This paper presents a new approach to computer Go ‘AlphaGo’ which can play Go at human-master-level. The authors introduced two deep convolutional neural networks, called ‘value networks’ and ‘policy networks’, to evaluate board positions and to select moves, respectively. AlphaGo, which combines Monte Carlo simulation with value and policy networks, defeated a human professional Go player.

Techniques

AlphaGo combines lots of techniques and algorithms. The most brilliant idea in this paper is to use deep convolutional neural networks (DCNN) to overcome the difficulty and complexity of position evaluation and move selection in Go. The authors evaluate positions using a value network, and sample actions using a policy network.

The policy network is a 13-layer DCNN, with a 19×19 image which represents a board position s as input, and a softmax layer that gives a probability distribution over all legal moves as output. It was trained in two stages: first supervised learning(SL) and then reinforcement learning(RL). The SL policy network is trained from 30 million positions from the KGS Go Server. Next, the author trained a RL policy network that improves the SL policy network by optimizing the final outcome of games of self-play. Using a reward function that is only non-zero for terminal steps, the RL policy network was adjusted towards the correct goal of winning games, rather than maximizing predictive accuracy.

The value network has a similar architecture to the policy network, but outputs a single prediction instead of a probability distribution. The optimal value function was estimated by using the RL policy network. The weights of the value network were trained by regression on state-outcome pairs.

AlphaGo combines the policy and value networks in an Monte Carlo Tree Search algorithm. Each simulation traverses the tree by selecting the edge with maximum action value Q, plus a bonus $u(P)$ that depends on a stored prior probability P for that edge. The nodes are processed once by the policy network and the output probabilities are stored as prior probabilities P for each action. At the end of a simulation, the leaf node is evaluated by the

value network and by running a rollout to the end of the game with the fast rollout policy, then computing the winner with reward function. The action values Q are updated to track the mean value of all evaluations in the subtree below that action.

Results

A single-machine AlphaGo is many *dan* ranks stronger than any previous Go program, winning 494 out of 495 games (99.8%) against other Go programs. The distributed version of AlphaGo was significantly stronger, winning 77% of games against single-machine AlphaGo and 100% of its games against other programs. The distributed version of AlphaGo defeated a human professional 2 *dan* player Fan Hui with 5 to 0.