Research review

This paper contains short review of the “Mastering the game of Go with deep neural networks and tree search” article. Comparing to the other fully observable games like tic-tac-toe, elimination, checkers and chess, Go is considered much more difficult for computers to win because of much larger branching factor, which makes traditional AI search methods such as minimax, alpha-beta pruning and heuristic search infeasible. Taking into account inefficiency of the mentioned above algorithms for implementing Go AI agent, a new approach based on deep neural networks in combination with Monte Carlo tree search (MCTS) is used.

Training neural networks

Recently, deep convolutional neural networks achieved outstanding performance in image processing and this architecture was employed in order to reduce the effective depth and breadth of the search tree by evaluating positions using a value network, and sampling actions using a policy network. The neural network is trained using a pipeline consisting of several stages of machine learning. First, a 13-layer policy network is trained from 30 million positions of expert human moves by applying supervised learning machine learning task. This initial network predicted expert moves with an accuracy of 57.0%. Second, the policy network is improved by applying reinforcement learning machine learning task. With improved policy network, the agent won more than 80% of the games against the supervised learning policy network. Furthermore, using no search at all, the policy network created by reinforcement learning won 85% of games against Pachi, the strongest open-source Go program. In comparison, the initial policy network created using supervised learning won only 11% of games against Pachi, and 12% against a slightly weaker program, Fuego. On third and the last stage of the training pipeline, a value network was built using reinforcement learning. The value network is designed to predict the outcome from position S of games played by using policy P for both players.

Searching with trained policy and value networks

AlphaGo combines the policy and value networks in an MCTS algorithm. MCTS is a heuristic search algorithm for some kinds of decision processes, which is focused on the analysis of the most promising moves, expanding the search tree based on random sampling of the search space. In AlphaGo the tree is traversed by simulation, starting from the root state. Once the search is complete, the algorithm chooses the most visited move from the root position. To efficiently combine MCTS with deep neural networks, AlphaGo uses an asynchronous multi-threaded search that executes simulations on CPUs, and computes policy and value networks in parallel on GPUs.

Results

By running a tournament among variants of AlphaGo and several other Go programs, AlphaGo achieved dan rank which is many ranks stronger than any previous Go program, winning 494 out of 495 games (99.8%). Furthermore, Fan Hui, a professional 2 dan player and the winner of the 2013, 2014 and 2015 European Go championships, was defeated by AlphaGo in October 2015 with score 0 to 5 for AlphaGo. By combining MCTS with deep neural networks, AlphaGo has finally reached a professional level in Go, providing hope that human-level performance can now be achieved in other seemingly intractable artificial intelligence domains.