Research review of the paper:

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

Fangjun Shi

Goals and techniques introduced

The paper introduces a new approach to computer Go that uses ‘value networks’ to evaluate board positions and ‘policy networks’ to select moves based on deep neural networks and reinforcement learning. With thousands of random games of self-play, the deep neural networks play Go at the level of state-of-the-art Monte Carlo tree search. Monte Carlo tree search uses Monte Carlo rollouts to estimate the value of each state in a search tree. They also use a new search algorithm that combines Monte Carlo simulation with value and policy networks.

Pass in the board position as a 19 × 19 image and use convolutional layers in convolutional neural network to construct a representation of the position and reduce the effective depth and breadth of the search tree.

AlphaGo efficiently combines the policy and value networks with Monte Carlo tree search. They train the neural networks by using a pipeline consisting of several stages of machine learning. Here are three stage of the training pipeline :

1. Supervised learning of policy networks: Build on prior work on predicting expert moves in the game of Go using supervised learning
2. Reinforcement learning of policy networks: improving the policy network by policy gradient reinforcement learning
3. Reinforcement learning of value networks: focuses on position evaluation

AlphaGo combines the policy and value networks in a Monte Carlo tree search algorithm that selects actions by lookahead search. Evaluating policy and value networks requires several orders of magnitude more computation than traditional search heuristics. To efficiently combine Monte Carlo tree search 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.

The final version of AlphaGo used 40 search threads, 48 CPUs, and 8 GPUs. The distributed version of AlphaGo that exploited multiple machines, 40 search threads, 1,202 CPUs and 176 GPUs.

Key results

Single-machine AlphaGo is many dan ranks stronger than any previous Go program, winning 494 out of 495 games (99.8%) against other Go program.AlphaGo also played games with four handicap stones, AlphaGo won 77%, 86%, and 99% of handicap games against Crazy Stone, Zen and Pachi, respectively. The distributed version of AlphaGo was significantly stronger, winning 77% of games against single-machine AlphaGo and 100% of its games against other programs.

In other variants of AlphaGo, even without rollouts AlphaGo exceeded the performance of all other Go programs, demonstrating that value networks provide a viable alternative to Monte Carlo evaluation in Go. However, the mixed evaluation performed best, winning ≥95% of games against other variants.