Research Review

By Martin Lopez

AlphaGo by the DeepMind Team

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

Goals

The DeepMind team was looking to master the game of Go with deep neural networks and tree search. To beat what was known as Artificial Intelligence's last frontier in full information games.
Go has long been seen as the most challenging of classical games for Artificial Intelligence. Having approximately 250^150 possible sequences of moves, making exhaustive search infeasible and evaluating board positions and moves was also a huge obstacle. AlphaGo had to come up with something unheard of to beat the world champion.

Techniques

AlphaGo was made with two kinds of Deep Neural Networks and Monte Carlo Tree Search(MCTS). A fast rollout policy network and a supervised learning(SL) policy network were trained to predict human expert moves. Then a reinforcement learning(RL) policy network is initialized with the SL policy network, and improved via policy gradient learning to win against previous versions of the policy network. Finally a value network is trained by regression to predict the expected outcome in game states during self play. MCTS uses Monte Carlo rollouts to estimate the value of each state in the tree, as more simulations are executed the values become more accurate. MCTS is used alongside with the policy and value networks.

Results

AlphaGo had a five second computation time per move, and a 99.8% winning rate against the other computer Go programs. DeepMind's research also revealed the level of computational power required to conquer such a task. The final version of AlphaGo used 40 search threads, 48 CPUs, and 8 GPUs. Though a distributed version with 40 search threads, 1,202 CPUs, and 176 GPUs was also implemented, the program's competitiveness in terms of Elo rating exhibited diminishing returns.