Satish Boggarapu

Udacity AIND – AlphaGo Research Review

The game of Go is viewed as one of the most challenging classic game for artificial intelligence to master due to its enormous search space and the difficulty of evaluation board positions and moves. A search tree for a game has approximately b^d possible sequence of moves, where b is the games breadth and d is the depth of the game. For example, a Chess game has on average breadth of 35 and depth of 80. Similarly, a Go game has a breadth of 250 and depth of 150. As a result of such a high breadth and depth values for the game of Go, it clear why its so difficult for an AI system to master the game. The DeepMind team accepted this challenge and built a AI agent that has successfully defeated the best Go player on the planet by winning 5 games out of 5 games. The AI agent used various techniques to accomplish this task.

A depth-first minimax search with alpha-beta pruning algorithm has been shown to be effective in many games like chess, checkers and Othello, but necessarily for Go. Since the minimax with alpha-beta pruning doesn't quite work well for Go, an alternative approach to minimax was used. The alternative approach was a Monte Carlo tree search (MCTS) which estimates the optimal value of interior nodes by a double approximation. The first approximation uses n Monte Carlo simulations to estimate the value function of a simulation policy P^n. The second approximation uses a simulation policy P^n in place of minimal optimal actions. AlphaGo uses a value function that's based on truncated Monte Carlo search algorithms, which terminate rollouts before the end of the game and use a value function in place of a terminal reward.

AlphaGo used an asynchronous policy and value MCTS algorithm to efficiently integrate large neural networks. They also implemented a distributed APV-MCTS algorithm. The architecture of the algorithm consist of a single master machine that executes the main search, many remote worker CPUs that execute asynchronous rollouts along with many GPUs. Along with these methods symmetries, policy network classification, reinforcement learning and value network regressions were used to successfully build a AI agent that can beat the best the Go player.