Research Review (AlphaGo)

This paper reveals the algorithms used by AlphaGo that successfully defeated one of the best Go player in the world. It's being considered a major breakthrough in artificial intelligence. In this review, I will go through the major techniques mentioned in the paper, and explain how they work.

Due to the enormous search space of game Go, it's too large for exhaustive minimax tree search, and alpha-beta pruning has not been effective. So AlphaGo introduces a new search algorithm that combines Monte Carlo simulation with value and policy networks that achieved a 99.8% winning rate against other Go programs.

Supervised learning of policy networks
At this first stage, the 13-layer convolutional neural networks is trained to predict expert moves in the game of Go. It uses 30 million positions data from KGS Go Server for training and could output a probability distribution over all legal moves for given state. With the test set, it achieved an accuracy of 57.% using all input features and 55.7% using only raw board position and move history. There is another more faster and smaller version of rollout policy network with less accuracy of 24.2%, which will be used in the later stage.

Reinforcement learning of policy networks
At the second stage, the same type of convolutional neural networks is trained by using a different method. Unlike supervised learning, the goal of reinforcement learning is to improve the policy network by playing the game by itself. The RL policy network won more than 80% of games against SL policy network. And it won 85% of games against Pachi which is the strongest current Go program based on MCST in doing 100,000 simulation each turn. RL policy network doesn't use any tree search at all.

Reinforcement learning of value networks
This is the final stage of the training pipeline focuses on position evaluation. Its goal is estimating a value function that could predict the outcome of state *s* that played by policy *p*. The neural network structure is similar to the previous policy networks. And its function is quite similar to the evaluation function used in our game-play agent, but instead of handcrafted rules, this thing learns by playing against itself.

The Search
To put everything together, AlphaGo combines the policy and value networks in a MCTS algorithm that selects actions by lookahead search. Even though the game Go has a wide branching factor, but with the value network and policy networks, AlphaGo could select few most important nodes to explore based on probability distribution of the next moves, and the evaluation function is a combination of value network and simulation result.

Evaluation

The internal tournament among variants of AlphaGo and several other Go programs were performed to evaluate the performance of AlphaGo. As the result, AlphaGo won 494 out of 495 games (99.8%) against other Go programs. Finally, when AlphaGo encountered one of the best human player, it won the match 5 games to 0.