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

The game of Go was considered to be the most complicated board game as evaluating board positions (game depth) and possible moves (game breadth) were enormous compared to other board games such as Chess. Due to the enormous search space in Go, it was said that human would outperform A.I. for another decade. However, the new algorithm introduced by AlphaGo overturned this theory; AlphaGo highly outperformed other Go programs and defeated the human Go champion for the first time.

AlphaGo's techniques were using machine learning in the search algorithm, and combining this algorithm with Monte Carlo Tree Search (MCTS). As the game of Go has enormous pattern search to play, it is very critical to reduce the number of both depth and breadth of the search to win. The depth of the search is reduced by position evaluation (value network), and the breadth of the search is reduced by minimizing the possible moves (policy network). Traditional Go programs also used their human-based algorithm to reduce the search; however, Alpha Go applied the techniques of supervised learning (SL) and reinforcement learning (RL).

First, AlphaGo began by training an SL policy network to predict human expert moves by learning 30 million positions from the games played by a human and also trained a faster but less accurate rollout policy (fast rollout policy). Then, AlphaGo trained an RL policy network that improved the SL policy network by optimizing the outcome of games of self-play rather than improving the accuracy of the prediction of human exert moves. At last, AlphaGo focused on position evaluation, where AlphaGo trained a value network that predicts the winner of games played by the RL policy network against itself.

Finally, to decide each move of the game, AlphaGo combined the policy and value networks in an MCTS algorithm that selects actions by lookahead search. Each simulation of the current moves traverses the search tree starting from the root state by selecting the edge with maximum action. Then, the leaf node is expanded when the traversal reaches a leaf node. The new node is processed once by the policy network, and the output probabilities are stored as prior probabilities for each action. The leaf node is then evaluated by both the value network and running a rollout to the end of the game with the fast rollout policy. At the end of the simulation, traversed edges are updated by accumulating the visit count and mean evaluation of all simulation passing through that edge. Once the search is complete, the algorithm chooses the most visited move from the root position.

As combining MCTS with policy and value networks which used SL and RL techniques, Alpha Go has reached a level of exceeding the human performance. Not only AlphaGo has contributed a significant breakthrough in the field of A.I. but also provided a potential to solve other domains such as general game-playing, classical planning, partially observed planning, scheduling, and constraint satisfaction.