

# Research Review for Alpha Go Nature Paper

By Yutao Liu

After finishing my first game playing agent, I tried to take a roughly look at the relevant paper in this domain, and I choose to review the paper on Alpha-Go from Deep-Mind team<sup>i</sup>. In our work, we implement **Monte Carlo tree search** algorithm to build a game tree of different possible moves in Isolation Game. Then use min-max function to find the best move of the player and even the most possible move of the opponent player. But the optimal value of a search tree containing approximately  $b^d$  possible move, where  $b$  is the average of game's breadth and  $d$  is its depth. So we implement heuristic function to evaluating the score of moves and alpha-beta pruning algorithm to cut the game tree. And making it possible to get a better move in limit times and limit calculate resources. In the game of Go, there is even 250 breadths and 150 depths, which needs better techniques for AI agent to against the professional human player.

## **Paper's Goals:**

This is a paper shows how a computer program can defeated a human professional player in the full-sized game of Go, then have a introduction of Alpha-Go's techniques, last show some details of its methods.

## **System design:**

Alpha-Go uses Monte Carlo tree search(**MCTS**) , Reinforcement Learning and Deep Neural Network .etc . The MCTS is to find the as much as possible moves and use other techniques like alpha-beta pruning to cut unnecessary nodes. Reinforcement Learning technique is used to find the best policy or move in the game board. Deep Neural Network uses supervise learning to evaluate the effort of the action on the game state.

## **Supervised learning of policy networks:**

At first, we should use supervise learning to evaluating the effort of the action in the state. Then we can use this score to find which policy we should take in the different state. The supervise deep neural network is a 13-layer policy network, which called the SL policy network, from 30 million positions from the KGS Go Server.

## **Reinforcement Learning:**

As we have a evaluation of the action and state score, then make a Q table contains actions and states. Then to find the highest effort of the action in the same state.

**Searching with policy and value networks:**

Like human players, when we are playing chess, we always think if I take this move and how will my opponent take. So we use MCTS to search as much deeper as the agent can to find the best policy of in the current state. As we mentioned before, it's hard to evaluate all the possible moves in the game search tree, then we implement alpha-beta pruning to cut unnecessary nodes and make it more efficient.

**Results:**

The paper shows that based on deep neural networks that are trained by a novel combination of supervised and reinforcement learning, it's more efficient to make an AI agent to playing GO. And they introduced a new search algorithm that successfully combines neural network evaluations with Monte Carlo rollouts.

**References:**

[1] Mastering the game of Go with deep neural networks and tree search, by David Silver etc. @ <https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf>

---