# Research Review on AlphaGo

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

The first news about Deep Blue has been two decades. I was quite curious on why I never heard any breakthrough on the Go game where I was young. I got the answer when I started to learn computer science during my undergraduate study. The theoretical complexity of the game state is 3^361, the number is just gigantic, it’s even greater than the number of particle in the universe. And I dare not thinking we would have a breakthrough soon.

The latest news on the Go game is our AlphaGo Master defeated the top human player Ke Jie.

The minimax and alphabeta pruning techniques taught in the AIND are very effective and easy to implement in the game board with relatively small branch factor and depth. When we put the techniques into the Go game with typical breadth around 250 and depth of around 150, the effectiveness and efficiency drop drastically.

In AlphaGo, a different search algorithm, named Monte Carlo Tree Search is implemented. Instead of expanding the nodes and running an evaluation function at the terminal node in minimax, MCTS expands the search tree by playing a random playout until reach the leaf node. The evaluation function in the minimax relies heavily on the understanding of the game rule and sometimes not easy to construct such a formulation, while MCTS doesn’t have such needs. But we almost come back to square root of 1 that it’s still a huge search space to exhaust the random moves and find the optimal one. That’s how this AlphaGo algorithm make a huge difference and championed all the Go AI agents.

The AlphaGo starts with a pipeline of several neural networks. The purpose is to predict the moves like a professional human player to build a probability distribution set of moves. This will dramatically shrink the search space. Then MCTS will be kicked in to perform the search. And in the pipeline, below neural networks and search techniques are included.

1. Supervised Learning (SL) of Policy Networks: a data set of 30 million positions taken as the input. The network is capable to predict the expert move with an accuracy of 57%. A larger network usually achieves better accuracy but with more computation cost. Therefore, a faster but less accurate Rollout Policy is also generated here. The output is a probability distributions data set.
2. Reinforcement Learning (RL) of Policy Networks: once the SL policy network is ready, RL is built on top of it in such a way that it competes against its randomly selected previous iteration to keep evolving and growing itself.
3. Reinforcement Learning of Value Networks: unlike the policy networks to output a probability distribution of moves, the value networks take the board position as the input but predict the outcome and output a scalar value.
4. MCTS with Policy Networks and Value Networks: when the MCTS is running, it takes the input from both policy networks and value networks for evaluation and then combine them.

Results of the end tournament is remarkable. A single-machine AlphaGo is much stronger than any previous Go programs. And based on the latest news on AlphaGo master, we, human beings lost the Go game forever!