# A Summary of AlphaGo

## Background

IBM's Deep Blue defeated Kasparov in 1997, the approach using at that time is almost the same as the minimax and alpha-beta methods we learnt in this course, with the heuristic functions designed by expert players.

However, the game of Go is much more challenging for AI owing to its enormous search space and the difficulty of evaluating board positions and moves. The search tree for Go containing approximately $b^d$ possible sequences of moves, where $b \approx 250$ and $d \approx 150$. Compare to $b \approx 35$ and $d \approx 80$ in chess, it is almost impossible to make a full search through the search tree given the computing power.

Monte Carlo tree search (MCTS) uses Monte Carlo rollouts to estimate the value of each state in a search tree. MCTS was well established and there are already some Go programs based on MCTS achieved strong amateur play. However, prior work utilized shallow or simple policies or value functions based on a linear combination of input feature.

## Techniques Introduced

This paper introduced deep convolutional neural networks (CNN) for policies and value functions. Recently, CNN have achieved great performance in image classification, face recognition and so on. And by representing the Go board as a 19x19 image, we can use the similar architecture to generate policies and value functions. Basically, policy network sampling the possible moves with some good options, so the breadth for searching is reduced. The value network evaluates the positions, and the better quality of the evaluation lead to a higher chance to win the game.

The first stage of training on the policy network is build a network to predicting expert moves. The SL policy network was trained on 30 million (state, action) pairs, which comes from history human games. In the network, all 12 layers were convolutional with ReLu activation, and the output layer was a softmax function. After training with SGD, the network predicted expert moves with an accuracy of 57% using all input features.

The second stage is to improve the previous policy network by policy gradient reinforcement learning (RL). We play games between current policy network and a random selected previous iteration of the policy network, and update weights by SGD to maximize winning possibilities. After training the RL policy network, it wins more than 80% of games against the SL policy network and 85% against Pachi (the strongest open-source Go program on KGS).

The value network's architecture was almost the same as policy network, but the output is a single value. The network is trained with data from self-play games. To prevent over-fitting, the data set is sampled from separate games.

## Results

AlphaGo wins 494 out of 495 games (99.8%) against other Go programs. AlphaGo won 77%, 86%, and 99% of handicap games against Crazy Stone, Zen and Pachi, respectively. The distributed version of AlphaGo was significantly stronger, winning 77% of games against single-machine AlphaGo and 100% of its games against other programs.

In October 2015, AlphaGo defeat European champion Fan Hui 5 to 0.

In March 2016, AlphaGo defeat 9-dan world champion Lee Sedol 4 to 1.

In January 2017, AlphaGo play games on Chinese online Go gaming platform with username "Master" and won 60 to 0 against professional Go player (including the current 1st ranked player Ke Jie).