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

## Background

Go is one of the most challenging classic game for artificial intelligence (AI). It is a perfect information game with 250 breadths and 150 depths. So, It is not practical to perform exhaustive search. In the past, AI was unable to defeat expert player of Go. And in the paper, neural networks and Monte Carlo tree search (MCTS) are suggested to solve this problem.

## Technique

4 neural networks and MCTS are used to develop the game agent, AlphaGo.

**Rollout policy network** is a neural network for classification purpose. The training dataset is come from 8 million positions from human games. It outputs the probability distribution of next legal moves. This network is fast, but less accurate. It just uses 2μs to obtain the output.

**Supervised learning (SL) policy network** is a deep convolutional neural network (ConvNet) for classification. It is a 13-layer network and trained by 30 million positions from human games. The output of this network is the probability distribution of next legal moves. This network has better accuracy, but requires larger computation power. It needs 3ms to calculate the output. But the accuracy of this network is 57%, which is much higher than rollout policy network (24.2%)

The network structure of **reinforcement learning (RL) policy network** is same as SL policy network. It is trained by self-play of SL policy network. The output of this network is still the probability distribution of next legal moves. But instead of predicting next human move, this policy network is concerned about winning the game. It is focused on long term benefit instead of short term reward. RL policy network performs quite well. And the winning rate against SL policy network is 80%.

The network structure of **value network** is similar to SL policy network. But it is for regression purpose. The training dataset is generated from self-play with 30 million distinct positions. It outputs a scalar that represent the chance to win in given game state.

**MCTS** is a tree search algorithm. AlphaGo combines above networks with MCTS to determine the future move. The tree is traversed by simulation. Firstly, in each simulation, it selects the edge with maximum action value plus bonus value. Then, the leaf node may be expanded. After that, the leaf node is evaluated in following two different ways to find the winner,

1. By value network
2. By rapid playing with rollout policy network

Finally, the action value of the edges are updated. And after search is completed, MCTS will choose the most visited move from the root position.

## Result

AlphaGo achieved 99.8% winning rate against other Go programs. The distributed version of AlphaGo was even stronger. It achieved 100% winning rate against other programs.
Also, in October 2015, AlphaGo defeated Fan Hui, who is a human European Go champion, by 5 games to 0. This is the first time a computer program defeated a professional human player in this game.