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

Research review by **Marco Zorzi**

The Paper, Mastering the game of Go with deep neural networks and tree search, introduce the computer program AlphaGo that uses “value networks” to evaluate board positions and ‘policy networks’ to select moves. These deep neural networks are trained by a novel combination of supervised learning from human expert games, and reinforcement learning from games of self-play. Without any lookahead search, the neural networks play Go at the level of state-of-the-art Monte Carlo tree search programs simulate thousands of random games of self-play. AlphaGo uses a new search algorithm that combines Monte Carlo simulation with value and policy networks. Using this search algorithm AlphaGo achieved a 99.8% winning rate against other Go programs and defeated the Fan Hui, a professional 2 dan, and the winner of the 2013, 2014 and 2015 European Go championships. In October 2015 AlphaGo and Fan Hui competed in a formal five-game match. AlphaGo won the match 5 games to 0. This is the first time that a computer Go program has defeated a human professional player in the full game of Go—a feat that was previously believed to be at least a decade away.

Details on implementation

1- **Supervised learning (SL) of policy networks**

With a 13-layer policy network, called SL policy network, is trained on randomly sample state-action pairs from 30 million positions from the KGS Go Server. The neural network takes input features from the board position and outputs the probability of each move on the board being the actual next move.

2- **Reinforcement learning (RL)**

The RL policy network is identical in structure to the SL policy network, and its weights are initialized to the same values. They play games between the current policy network and a randomly selected previous iteration of the policy network. Randomizing from a pool of opponents in this way stabilizes training by preventing overfitting to the current policy. Weights are then updated at each time by stochastic gradient ascent in the direction that maximizes expected outcome.

3- **Reinforcement learning of policy networks**

The training pipeline focuses on position evaluation, estimating a value function that predicts the outcome from positions of games played by using policy for both players. This neural network has a similar architecture to the policy network, but outputs a single prediction instead of a probability distribution. They generated a new self-play data set consisting of 30 million distinct positions, each sampled from a separate game. Each game was played between the RL policy network and itself until the game terminated.

4- **Searching with policy and value networks**

AlphaGo combines the policy and value networks in an MCTS algorithm that selects actions by lookahead search. Each edge of the search tree stores an action value, visit count, and prior probability. The tree is traversed by simulation, starting from the root state. Once the search is complete, the algorithm chooses the most visited move from the root position.