Understanding AlphaGo

Scale Challenge and Proposed Solution

Go is a popular two players board game in Asia, with the goal of capturing more territory than the opponent. Like Chess, Go is a game of perfect information in which each player, when making any decision, is perfectly informed of all the events that have previously occurred. Game of perfect information may be solved by evaluating each possible move of the game tree containing b^d sequences of moves, where b is the game's breadth (number of legal moves per position), and d is it depth (game length). Unfortunately, exhaustive search of each possible move of the game tree is infeasible because for Go, b is around 250, and d is around 150.

To make the problem more tractable, AlphaGo uses deep neural networks to reduce the effective search space in two aspects. First, it applies two *policy networks* to reduce the breadth of the search by finding the best probable moves. Another *value network* is used to reduce the depth of the search tree by making it unnecessary to search to the end of the tree. The game board is represented as a 19×19 image and convolutional layers are used to construct the game state that becomes the input to these networks.

Policy Network

The policy network takes the current state of the game to decide the best next move to make. The output is a probability distribution over legal moves, in which the best move has the highest probability to win. At first training stage, a *Supervised-Learning (SL)* policy network was trained on 30 million positions from games played by human experts, available at the KGS Go server. The trained network is able to predict expert moves with an accuracy of 57%. Second stage of the training is *Reinforcement Learning (RL)* that involves playing games between the policy network and randomly selected previous iterations of itself. The objective is to teach the network which next moves are likely to lead to winning the game. The trained RL network won more than 80% of games against SL policy network, and 85% games against Pachi — the strongest open-source Go program based on 100,000 simulations of *Monte Carlo Tree Search (MCTS)* at each turn.

Value Network

Final stage of training is focused on estimating the chance of each player winning the game, given the current game-state. A value network was trained on 30 million game positions, selected randomly from simulated games between two RL policy networks. A single evaluation of the trained value function had the similar accuracy of Monte-Carlo rollouts using the RL policy network, but using 15,000 times less computation.

Tree Searching with Policy and Value Networks

Finally, AlphaGo combines the policy and value networks with MCTS to construct the final gameplay engine. It uses the mixture of the output of the value network and the result of a self-play simulation of the policy network to guide the search tree in selecting which variations to be explored, and how deeply to explore them.

Results

The performance of AlphaGo is tremendous. It achieved 99.8% winning rate against the other Go programs like Crazy Stone, Zen, and Pachi in tournament evaluation. At the time the paper was written, AlphaGo had defeated European Go champion Fan Hui. On 27 May 2017, AlphaGo beat Ke Jie, the highest ranking professional Go player by 3 games to 0. The win over Ke, essentially confirms that AlphaGo has surpassed human Go ability and could be regarded as a major milestone in the field of artificial intelligence.

Reference

Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." *Nature* 529.7587 (2016): 484-489.