Research Review

Most of Go program use Monte Carlo tree search (MTCS) as their core algorithm. Base on MTCS, Google Deep Mind team introduce a new approach to computer Go that uses deep neural networks and tree search to improve performance to a 99.8% winning rate against other Go programs, and defeated many world-class human Go players.

With this new approach, we can develop effective move selection and position evaluation function for Go and integrate with a search algorithm that combines neural network evaluation with Monte Carlo rollouts. The basic idea is that we first train a supervised learning (SL) policy network directly from export human moves. We also train a fast policy that can rapidly sample actions during rollouts. Next, we train a reinforcement learning (RL) policy network that improves the SL policy network by optimizing the final outcome of game of self-play. Finally we train a value network that predicts the winner of games played by the RL policy against itself. In this way, AlphaGo efficiently combines the policy and value network with MCTS.

Supervised learning of policy networks

In this stage, supervised learning is used to predict human experts' moves. With 30 million positions from the KGS Go Server, 13-layor policy network is trained to be able to predict expert moves at around 55% approximately.

Reinforcement learning of policy networks

This stage aims at improving the policy network by policy gradient reinforcement learning. We played games between current policy network and a randomly selected policy previous iteration of the policy network. Weights are then updated at each time step that maximizes the accuracy of prediction probability function.

Reinforcement learning of value networks

The final stage of the training pipeline focuses on position evaluation, estimating a value function that predicts the outcome form position of games. We trained the weights of the value network by regression on state-outcome pairs, using stochastic gradient to minimize the mean squared error between the predicted value and the corresponding outcome.

Searching with policy and value networks

AlphaGo combines the policy and value networks in a MCTS algorithm.

When the traversal reaches a leaf node, the leaf node may be expanded. The leaf position is processed just once by the SL policy network. The output probabilities are stored as prior probabilities for each legal action. The leaf node is evaluated in two very different ways: first, by the value network; and second, by the outcome of a random rollout played out until terminal step using the fast rollout policy; these evaluations are combined into a leaf evaluation.

At the end of simulation, the action values and visit counts of all traversed edges are updated. Each edge accumulates the visit count and mean evaluation of all simulations passing through that edge. Once the search is complete, the algorithm chooses the most visited move from the root position.

AlphaGo performance evaluation

To evaluate AlphaGo, several Go programs are tested and Alpha performed stronger than any previous Go program by winning 494 out of 495 games (99.8%). Finally AlphaGo won the match against FanHui, a professional 2 *dan* with 5 games to 0. And as we all know, AlphaGo also defeated world-class player Lee Se-dol and world no.1 ranking holder KeJie by 2016 and 2017.

#