

Review: AlphaGo <https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf>

Author: Erdem Başeğmez (eduedix@gmail.com)

Date: 15.03.2017

AlphaGo is an agent, which has defeated proficient players and other agents with a winning rate of 99.8% in game of Go. It is built by a combination of Monte Carlo Search Trees (MCTS) and deep neural networks. The difficulty of creating an agent for Go relies on the possible sequences of moves and thus exhaustive search is not viable. However, the search space can be reduced by the following technique. First, the search tree is truncated, where a state in the tree is replaced by its subtree evaluated with a value function. Second, the breadth of the search tree is reduced by sampling actions for a state according to a policy.

Prior work was limited to shallow policies and weakly designed value functions. AlphaGo takes advantage of deep neural networks to create a better policy network and value network.

Firstly, a neural network (SL) is trained for policy network p_{SL} . The input of the samples is a board state and the output is expert human move for a given board state. The output is provided as a probability distribution over legal moves. The trained SL policy network can predict with an accuracy of 57% in 3ms as compared to 44% state-of-the-art policy. The improvements in accuracy lead to large improvements in the playing strength. For AlphaGo, in addition to p_{SL} , a faster policy network p_{RO} is trained with a smaller network, which achieves an accuracy of 24% but can predict a move in 2 μ s.

Second stage of training aims to improve the p_{SL} by reinforcement learning. Another policy network p_{RL} is introduced, which has the identical structure of p_{SL} with weights initialized from p_{SL} . The policy network p_{RL} is then updated by self-playing against randomly selected previous iterations of the p_{RL} . The optimized network tries to maximize a predefined reward for a move given a state. The trained p_{RL} is able to defeat p_{SL} in 80% of games, state-of-the-art Monte Carlo search program 85% of games. In comparison previous state-of-the-art supervised deep neural networks wins 11% against the Monte Carlo search program.

The final stage is training a value network v to estimate a value function using the policy network p_{RL} , which predicts the outcome of a board state. This neural network is similar to the policy network, but outputs only a single prediction. A specific technique is followed here during training to avoid memorization of the game outcomes instead of generalizing to new board states. The generated self-play data sets sampled from separate games are used as training samples to overcome this problem. As compared to Monte Carlo rollouts with p_{RO} ,

the value network v is consistently more accurate. Moreover, v achieves the accuracy of Monte Carlo rollouts with p_{RL} using 15000 times less computation.

The policy and value networks are combined in an MCTS to select a move given a board state. Along the tree each move is evaluated with parameters action value, visit count and prior probability. Among all moves, the move with the maximum evaluation is selected. The action value is obtained by combining output of value network v and a random rollout played with policy network p_{RO} . During tree traversal the visit counts of the moves are updated. The prior probability is simply the output of p_{SL} given a board state for a move.

Evaluating the policy and value networks of AlphaGo is computation intensive and thus execution is done on multiple CPUs and networks are evaluated with GPUs in parallel. In tournaments, AlphaGo beats the strongest Go programs, all of which are based on MCTS algorithms, winning 494 of 495 games. In an experiment, it is also noticed that for a given action value without fast rollouts, AlphaGo exceeded the performance of other Go programs. It is also seen that a balanced mixed evaluation of value network v and policy network p_{RO} performs best by winning 95% of games versus other variants. Accordingly, it can be suggested that the value network v approximates the outcome of the game played by a slow and strong p_{RL} , while weaker but faster p_{RO} can precisely score and evaluate the outcome of the game.

AlphaGo defeated for the first time a human professional player. In comparison to the chess match between Deep Blue and Kasparov, AlphaGo evaluated thousands of times less states by selecting states more intelligently using the policy network and evaluating them more precisely using the value network.