Itsuo Okamoto
May 29, 2017

RESEARCH REVIEW OF ALPHAGO

The game of Go has long been viewed as the most challenging of classic games for artificial intelligence owing to its enormous search space and the difficulty of evaluating board positions and moves. The game of Go may be solved by recursively computing the optimal value function in a search tree. But exhaustive search is infeasible due to large possible sequences of moves.

AlphaGo introduces a new approach by using '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.

For the first stage of the training pipeline, AlphaGo builds on prior work on predicting expert moves in the game of Go using supervised learning (SL). The SL policy network predicted expert moves on a held out test set with the accuracy of 55.7% using only raw board position and move history as inputs while other solutions predicted with the accuracy of 44.4%.

The second stage of the training pipeline aims at improving the policy network by policy gradient reinforcement learning (RL). The RL policy network is identical in structure to the SL policy network. It uses a reward function. Weights are updated at each time step by stochastic gradient ascent in the direction that maximizes expected outcome. When it played head-to-head, the RL policy network won more than 80% of games against the SL policy network.

The final stage of the training pipeline focuses on position evaluation, estimating a value function that predicts the outcome from position 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. This could result in overfitting with KGS data set. They mitigated this issue by generating a new self-play data set. As a result, compared to Monte Carlo rollouts using the fast rollout policy; the value function was consistently more accurate. A single evaluation also approached the accuracy of Monte Carlo rollouts using the RL policy network, but using 15,000 times less computation.

Finally AlphaGo combines the policy and value networks in an Monte Carlo Search Tree (MCTS) algorithm that selects actions by lookahead search.

To evaluate AlphaGo, they ran an internal tournament among variants of AlphaGo and several other Go programs. All of other programs are based on high-performance MCTS algorithms. The results of the tournament suggest that single machine AlphaGo is many dan ranks stronger than any previous Go program, winning 494 out of 495 games (99.8%) against other Go programs. Finally, they evaluated the distributed version of AlphaGo against Fan Hui, a professional 2 dan, and the winner of the 2013, 2014 and 2015 European Go championships. Over 5–9 October 2015 AlphaGo and Fan Hui competed in a formal five-game match. AlphaGo won the match 5 games to 0.