Research Review of AlphaGo

Guo Qiang

August 9, 2017

1 Goals and Techniques

This paper introduces the techniques AlphaGo applies and how it achieves the high-scored performance. The objective of game Go is to occupy as much territories as possible to get highest points to win. But for Go, the search space is huge: the search breadth and search depth are around 250 and 150 respectively, usually b^d (b is the breadth and d is the depth), so it is always a challenge for AI because exhaustion is always inevitable.

AlphaGo truncates the depth of the tree by evaluating the state and replacing the state's subtree, and reduces the breadth of the search tree by sampling actions from a policy that is from a well trained model. In technical part, a policy network and a value network are trained in supervised learning and reinforcement learning and Monte Carlo Tree Search (MCTS) is applied to combine these networks together to enhance the performance. It finally wins all open source Go programs and professional Go experts.

1.1 Supervised learning of policy networks

This is the 1st stage of training. To train the neural network, the input data are paired state-action (s, a) and the outcome is the probability array of all possible actions. On the test stage, the accuracy of SL policy network is around 50% as it varies with the training data sets. And a faster but less accurate (accuracy 24.2%) rollout policy using a linear softmax is also introduced which reduced action selecting time from 3 ms to 2μs.

1.2 Reinforcement learning of policy networks

Reinforcement learning is at the second stage of training and is to improve the policy network. It has identical structure with SL policy network and its weight values are initialized to the same values. After reinforcement learning, AlphaGo wins Pachi and Feugo(two open source Go program) by 11% and 12% respectively.

1.3 Reinforcement learning of value networks

This is the final stage of training. This neural network has a similar architecture to the policy network, but outputs a single prediction other than a probability distribution. This part focuses position evaluation s of games played by using policy p for both players. Problems of overfitting is mitigated by generating a new self-play data set.

1.4 Searching with policy and value networks

Monte Carlo Tree Search (MCTS) combines value network and policy network together. Thus Monte Carlo rollouts search to maximum depth without branching to improve the calculating ability. First, an action a is selected at time step t from state is targeted at maximizing action value plus a bonus function. Second, the leaf node may be expanded when traversal reaches that one and value network and rollout policy network are combined to evaluate that node. In the end, the action values and visit counts of all traversed edges are updated during backup process. MSTC chooses the most visited move from the root position after search is completed.

2 Conclusion

Monte Carlo Tree Search combines policy network and value network together to perform the AlphaGo strategy. CNN(Convolutional neural network) is built in the construction of both networks and supervised learning and reinforcement learning are applied during training process. Finally, AlphaGo is implemented in both single and distributed machine and both reach professional level and both version win Professional players and current best open source Go programs.