Summary: AlphaGo

Goals and techniques introduced

Google's AlphaGo has succeeded in making a great progress in the game of Go, defeating the human European Go champion. It had been said that it would take at least another decade in order for AI to beat a human professional player in the full-sized game of Go. Compared with other board games such as Chess, Go requires AI enormous amount of computation which had made AI unable to find the best results from calculation to beat a human professional. The volume of computation is measured by the game's breadth(number of legal moves per position) and depth(game length). For example, while Chess requires b(breadth) of 35 and d(depth) of 80, Go requires b of 250 and d of 150.

To solve this, Google found the way to reduce the volume of computation and search optimal moves. As already mentioned, AI needs to compute in two directions of breadth and depth. Google succeeded in reducing the volume in both directions by making value function to reduce the depth and policy to reduce breadth. In details, using the method of deep learning, convolutional neural network(CNN), Google found out where valuable positions are in a certain state of the game regarding them as images that CNN can learn, and made policies in terms of which moves have higher probabilities of winning.

Techniques and results

Google took the following steps for this achievement.

1. Supervised learning of policy networks(SL)

   Using database of games played by human professionals, SL policy, which can output the highest probabilities, pθ(a | s), for the next move from a certain state was created and trained by the method of gradient descend. This policy can predict the best move with a high accuracy of 57.0% whereas that of other research groups stayed with an accuracy of 44.4%. In addition, Google created another policy which has a less accuracy of 24.2% but more than 1,000 times faster than convolutional methods.

2. Reinforcement learning of policy networks(RL)

   In this step, they improved the policy network by policy gradient reinforcement learning(RL). In fact, they made the policy network play against the policy network itself. They started with SL Policy network, and then made them play each other and reinforce the network. As a result, RL policy network become to win 85% of games against Pachi which was one of the strongest algorithms.

3. Reinforcement learning of value networks

   From games of RL policy, they created value networks to evaluate positions and then a value function which can output a single value. By this function, AlphaGo can see where in a board are valuable to consider as a next move. This value function is consistently accurate and uses only 15,000 times less computation compared with Monte Carlo rollouts with RL policy network.

4. Searching with policy and value networks

   Finally, AlphaGo combines the policy and value networks in an MCTS(Monte Carlo tree search) algorithm. Since value networks reduce depth and policy networks reduces breadth, MTCS can effectively search optimal moves, which has made AlphaGo much stronger than ever.