AlphaGo by the DeepMind Team

1.Goals and Techniques

In this paper, Deepmind, a British artificial intelligence company founded in September 2010, introduced a new approach to conquer Go which is an abstract strategy board game and very complex to the computer as its enormous amount of searching space.

As the author mentioned in the paper, the strategies can be generally described as three part: **supervised learning (SL)**, **reinforcement learning (RL)** and **Monte-Carlo tree search (MCTS)**.

At the SL stage, a **supervised learning (SL) policy network** trained with human expert moves, it outputs a final soft-max layer outputs a probability distribution over all legal moves . More concretely, a 13-layer network is trained using image representations of the board, with moves taken from the KGS Go server (30 million samples). This is a effectively end-to-end approach like the image classification using convolutional neural networks. At the same time, a **fast policy network** trained that can rapidly sample actions during rollouts.

At the RL stage, a **reinforcement learning (RL) policy network** that evaluates self-play outcomes of the current state of the game and then **reinforcement learning (RL)of value network** predicts the winner of games using the data of SL policy network and RL policy network produced. More concretely, in **RL policy network**, it will copy the SL policy network as the initial newwork and then play games between the current policy network and a randomly selected previous iteration of the policy network. According to the reuslt it will use reinforce algorithm to update parameters for maximazation of probability distribution of moves. In **RL value network**, it has a similar architecture to the policy network, but outputs a single prediction instead of a probability distribution. the training pipeline focuses on position evaluation, estimating a value function that predicts the outcome from positions of games played by using SL policy network and RL policy network for both players.

At the **MCTS** stage, AlphaGo combines the policy and value networks in an MCTS algo-rithm that selects actions by lookahead search and utilizes random sampling of the tree with evaluation of each game tree branch. As the paper described, it has **4 steps**. **(a)** Each simulation traverses the tree by selecting the edge with maximum action value, plus a bonus that depends on a stored prior probability for that edge. **(b)** The leaf node may be expanded; the new node is processed once by the **SL policy network** and the output probabilities are stored as prior probabilities for each action. **(c)** At the end of a simulation, the leaf node is evaluated in two ways: using the **RL value network** ; and by running a rollout to the end of the game with the fast rollout **fast policy network** , then computing the winner with a evaluation function. **(d)** Action values are updated to track the mean value of all evaluations in the subtree below that action. Once the search is complete, the algorithm chooses the most visited move from the root position. The policy network helps the MCTS focus on paths that are of high likelihood of actually occurring. The value network helps the MCTS focus on moves that would likely be of high value.

2. Results

Using a combination of deep neural networks and tree search, AlphaGo achieved a 99.8% winning rate against other Go programs, and defeated the human European Go champion by 5 games to 0. At the same time, the game agent has evaluated far fewer moves than DeepBlue did in its chess match. Especially, AlphaGo creates new stategeis of Go that currently professional player can't understand, which is so amazing and unbelievable.