Games of perfect information and Go

Games of perfect information are games that may be solved exploring possible states using a search tree. The complexity of a tree can be approximately calculated by b^d where b is the breadth or number of legals moves per position and d is the depth or game length.

There are large games where is not feasible to explore the whole tree due to it's huge size. For exemple chess has around 10^{135} possible moves ($b \approx 35$, $d \approx 80$). Between this large games Go was supposed to be one of the most complex ones with approximately 10^{230} possible moves ($b \approx 250$, $d \approx 150$). And prior to this paper it was thought that computers will need at least a decade to be able to beat humans.

Prior Go AI

The best AI that played Go before Alpha Go used Monte Carlo tree search (MCTS).

Monte Carlo tree search (MCTS) is a heuristic algorithm that focus on searching the most promising moves. It expands the search tree basen on random sampling of the search space. It has 4 phases for expanding the tree (Figure 1).

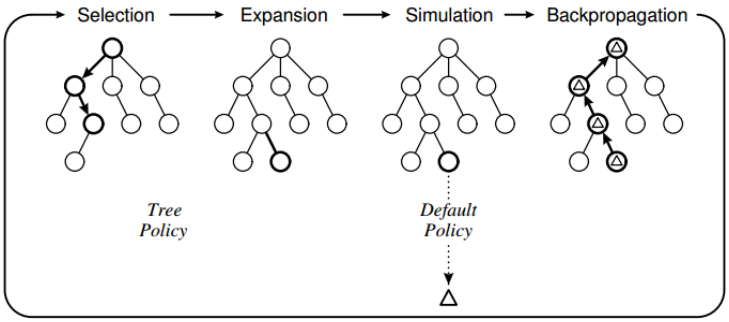

Figure 1: MCTS phases

Alpha Go

Alpha Go is the AI created by Google DeepMind team to play Go. The new idea that Alpha Go uses is to combine MCTS with policy networks. Alpa Go uses neural networks to improve the performance of MCTS. The goals of those neural networks used in AlphaGo are: effective position evaluation (value network) and actions sampling (policy network).

Policy network

This is the part where the AI choose which node should be explored in the tree. For this part they used 2 common Machine learning techniques:

- Supervised Learning (SL)
- Reinforcement Learning (RL)

Supervised learning of policy networks

For the first stage of the training pipeline they build a Convolutional Neural Network (CNN) that learned to predict Go movements based on moves of expert Go players. They also trained a faster rollout policy that complemented the CNN and helps to evaluate nodes in MCTS.

Reinforcement learning of policy networks

While the SL helps to predict the more likely moves this stage is used to predict the best possible moves. At this stage Alpha Go played against previous iterations of itself, which helped to avoid overfitting.

Value network

This is used by the AI to determinate the probability of the move to leads to a win. They started training with the KGS dataset but it lead to overfitting. So they generated a new dataset from self playing in order to prevent that.

Results

In order to test the results of Alpha Go they make it play against different versions of itself and against the best commercial and open source programs (like Crazy Stone, Pachi or Fuego). Alpha Go proved to be the strongest one winning 494 out of 495 games (99.8%). It even won more than 77% of the games when the opponent was given free moves.

But what really made Alpha Go famous was being the first AI to beat a professional Go player.