Brief Review on AlphaGo

Goal of the Research

Go is an ancient game that remains very challenging for AI to beat due to the large search space and difficulty in evaluating the moves and board. Go has a branching factor of approximately 250 and depth of 150, which is impossible to perform an exhaustive search. The goal of AlphaGo is to take on this challenge, creating an AI agent that can beat the best human Go players using a combination of cutting edge AI techniques.

Techniques Introduced

There are several techniques used in AlphaGo, which will be listed and described below.

1. Deep Convolutional Neural Networks

In AlphaGo, deep neural networks are used for various purposes. First, the policy network is used for move selection. It is trained using a representation of the board positions, and output the probability distribution of the legal moves. There are two types of learning used in the policy network, supervised learning for predicting human expert moves, and reinforcement learning for improving its performance by playing against itself. Second, the value network is used to evaluate the outcome of the moves. It is trained by regression, and outputs a scalar that represents the expected outcome of the positions.

2. Reinforcement Learning

AlphaGo uses reinforcement learning in addition to supervised learning to further improve its performance. This is achieved by randomly select a previous iterations of the policy network, then play against it using the current policy network. The weights are updated at every timestep using stochastic gradient descent learning rule, in the direction that maximizes the expected outcome. Randomization of the opponent policy networks is used to reduce overfitting and improve its generalisability.

In addition, reinforcement learning is also used to train the value network. It is done by estimating the value function for the best policy, by regression using stochastic gradient descent on the outcome-state pairs to minimize the MSE of the predicted and actual outcome. Similar to the policy network, random samples of positions taken from different games are used to minimize overfitting during training.

3. Monte Carlo Tree Search

In AlphaGo, the policy and value networks are used in Monte Carlo tree search to find the best possible move for the current game state. Every edge of the tree has an action value, visit count and prior probability. From the root state, the tree is traversed in descending manner, without backtracking to the terminal of the game tree. During the simulation, an action is selected for every timestep from the state, that tries to maximize the action value proportional to the prior probability but diminish as the number of visit increases. The prior probabilities are computed by the supervised learning policy network. Each leaf node is evaluated using a combination of the value network, and the fast random rollout policy.

After every simulation, the action values and number of visits for all edges are updated and stored. The most visited move from the root position are chosen as the candidate move for the corresponding game state. The evaluations of the moves are ran on immensely powerful hardware, using (distributed) asynchronous multithreading.

Results

AlphaGo performed extremely well against other computer players. The selected programs includes Crazy Stone, Zen, Pachi and Fuego. The single machine AlphaGo won 494/495 games against all other Go programs, with a win rate of 99.8%. Even when in handicap games, AlphaGo managed to win against Crazy Stone with 77% win rate, 86% for Zen and 99% for Pachi. The distributed version of AlphaGo has even higher win rate; 77% against single machine AlphaGo, and 100% against all other Go programs.

Most importantly, AlphaGo was pitted against the consecutive European Go championship winner, Fan Hui too see if the AI player can win against professional level human player. Over the 5 matches, AlphaGo won 5-0 against Fan Hui, defeating the human champion completely. AlphaGo achieved this feat just by using general-purpose supervised and reinforced learning methods instead of relying carefully handcrafted evaluation functions.