Research Review

Mastering the game of Go with deep neural networks and tree search

Goals

The AlphaGo paper introduce how to use deep learning to train a model using a value neural network and how to select branch of the game tree during play without human-designed heuristics. Those neural networks are trained by supervised learning from human expert games, and then reinforcement learning from games of self-play. The DeepMind team also uses a new search algorithm that combines Monte Carlo simulation with value and policy networks. Using this search algorithm, program ***AlphaGo*** achieved a 99.8% winning rate against other Go programs.

Methods

AlphaGo uses neural networks and Monte Carlo Tree Search (MCTS). The board position for each game state is represented as a 19x19 pixels image. The neural network takes input features from the board position and MCTS uses the networks to evaluate the value of each game position in the search tree and calculate the next promising move.

1. Supervised Learning
   The first policy is a 13-layer deep CNN trained on Go game positions by Supervised Learning. It provides fast efficient learning updates with immediate feedback.

2. Reinforcement Learning
   The next step is further trained using reinforcement learning which optimizes on the previous trained policy network. The games were calculated between the current policy network and a randomly selected previous iteration of the policy network.

Summary of result

AlphaGo was evaluated by different tournaments based on high performance MCTS algorithms like Crazy Stone, Zen, Pachi and Fuego. The results of the tournament suggest that single machine AlphaGo is many dan ranks stronger than any previous Go program, winning 494 out of 495 games (99.8%) against other Go programs. In order to have a greater challenge to AlphaGo, DeepMind also played games with four handicap stones; and then AlphaGo won 77%, 86%, and 99% of handicap games against Crazy Stone, Zen and Pachi, respectively. The final version of AlphaGo used 40 search threads, 48 CPUs, and 8 GPUs. Though a distributed version with 40 search threads, 1,202 CPUs, and 176 GPUs was also implemented, the program's competitiveness in terms of Elo rating exhibited diminishing returns.

The DeepMind's research has provided a thrilling result that human-level performance can be achieved in other seemingly artificial intelligence, and encourage more and more research in uncharted human vs computer competition domains.