Udacity Artificial Intelligence Nanodegree

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

Research Review

Abstract

In this paper, the authors describe the method they used to build an algorithm that was capable to achieve human-level performance in the game of Go. The method is based on the combination of Monte Carlo Tree Search and Deep Neural Networks.

Technique

The high-level description of the architecture used in AlphaGo is as follows:

1. **Policy Network** - Trained to predict the likelihood of every move for a given board state.
2. **Value Network** - Estimate the value of each move for a given board state.
3. **Monte Carlo Tree Search** - Used to traverse the game tree to find the most optimal move at any given board state.

Neural networks are used reduce the effective depth and breadth of the search tree using a value network, and sampling actions using a policy network.

Policy Network Training

The policy network is a **Convolutional Neural Network** architecture similar to ones recently used for image classification. The network is trained in two stages:

The first stage is a **Supervised Learning** stage where the model is trained directly on expert human moves. Then at the second stage, the model is trained using **Reinforcement Learning** to optimize the final outcome of games through self-play. When played head to head, the **RL** model won 80% of games against the **SL** model.

Value Network Training

This network focuses on position evaluation. The Value Network is also a **Convolutional Neural Network.** This model is trained using **Reinforcement Learning**. The model is trained on a newly generated self-play dataset that was sampled from seperate games to avoid overfitting e.g. memorising the sequences of moves from existing games.

Searching the tree with policy and value networks

Once the two models are trained they can now be used to efficiently search the game tree. AlphaGo combines the policy and value networks in a Monte Carlo Tree Search algorithm that selects actions by lookahead search.

The authors do note that using this method requires orders of magnitude more computation compared to earlier methods that do not use neural networks to evaluate the gamestate.

Results

AlphaGo achieved a 99.8% winning rate against other Go programs, and defeated the human European Go champion by 5 games to 0. This is the first time that a computer program has defeated a human professional player in the full-sized game of Go, a feat previously thought to be at least a decade away.