AlphaGo

Research Review

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

Krishna Kumar | Udacity: AI Engineering Nano Degree | Part - I

This paper aims to review the efforts behind the DeepMind team's successful and ground-breaking work in the Alpha-Go Game Playing AI, their implementation of Deep Neural Networks that makes it capable of achieving superhuman performance in the game of GO, a game where the branching factor is ~250 and the typical length of moves is ~150, and requires much more severe pruning.

Design

Goals

To design a system effective at minimizing the number of optimal states, in a 2-player zero-sum deterministic game with full observability, where an exhaustive search would be prohibitively expensive, both in terms of computational time and resources.

System Design

The AlphaGo approach involves using an asynchronous policy and value MCTS algorithm, with asynchronously executed evaluations (Deep-CNN), using remote CPUs and GPUs connected to a main master machine which stores the entire search, and executes the first in-tree phase of each simulation, and offloads the rest of the search onto its worker machines.

Policy Networks

The Alpha-Go agent uses three different policy networks, the first two for supervised(SL) and reinforcement(RL) learning and one for fast rollouts(FR).

Both the SL and RL networks are 13-layer deep CNN's. The SL network is trained on a sample of 30 million game positions, and given a game position, it predicts the *most likely next move, (*i.e. *what moves are possible)*. The RL network, starts as a copy of the SL network and predicts the *best next move, (*i.e. *what moves are optimal?)* by iteratively learning through 1.2 million games played against previous iterations of itself and selectively preserving the network weights of winners across games.

The third policy network(FR), predicts a *likely next move* like the SL network, but it's much faster and while not as accurate, its utility lies in quickly playing out the game up to the end-game state and predicting a likely outcome from the predicted next move.

Value Networks

The Value networks are designed to estimate if a current position will lead to a win or a loss for the current player. The value network trained on SL policy network had problems with overfitting because of strongly correlated positions within the training set, so a new dataset of uncorrelated self-play positions was constructed for the RL Value Network.

Search

The search occurs in 4 phases, starting with the selection phase where the action/edge with highest action value Q is selected. This is followed by the expansion phase where the selected branch is processed once by the slow policy network, to come up with a strong move. The third evaluation phase consists of the simultaneously running the value networks once to evaluate the new position and using the FR network to playout from that position to the end of the game, trying as many runs as it can, within the specified time limit. In the final backup phase, all the information collected from the all the lower level runs are propagated up the tree, the Q values are updated, and the agent chooses the move that yielded the highest Q value.

RESULTS

The paper shows that the AlphaGo AI achieved superhuman performance with far fewer searches compared to it contemporaries all the while using standardized evaluation functions like stochastic gradient ascent and Bayesian logistic regression.

References.

1. Mastering the game of Go with deep neural networks and tree search; David Siler et al; 2016, Nature.