Martin Přílučík – Artificial Intelligence nanodegree

Research review

Purpose

The purpose of this document is to present review of research paper Mastering the game of Go with deep neural networks and tree search as a part Game-playing Agent project submission.

Introduction

As a part of the project I was implementing an agent playing isolation game. Isolation is simple game compared to Go which has been viewed as the most challenging game for Artificial Intelligence given its huge search space and complexity in board and moves evaluation.

Goal

Goal of the research was to achieve professional human level performance in the full-sized game of Go.

Go is game of perfect information. Generally, these games may be solved by recursively computing the optimal value function by e.g. minimax or alpha-beta pruning but this is infeasible for Go where branching factor $b \approx 250$ and game length $d \approx 150$.

Techniques

To achieve the goal researches used following techniques and approaches.

AplhaGo is combination of **policy** and **value networks** with **Monte Carlo Tree Search (MCTS).** Value networks are used to evaluate board position and policy networks to select moves.

Monte Carlo rollouts are used to estimate the value in search tree.

Deep convolutional networks are used to process board position as image and construct the representation of the position. Neural networks reduce the effective depth and breadth of the search tree. Positions are evaluated using a value network and actions using a policy network.

Neural network is trained in several machine learning stages.

Supervised learning (SL) policy network from expert human moves. Input is simple representation of board state. The network has 13 layers and was trained from 30 million positions from KGS Go Server. Board representation, as input, passes through convolutional layers. Output is probability distribution over legal moves over the board map.

Reinforcement learning (RL) policy network to improve SL by self-play games between current policy network and randomly selected previous network iteration. Focus is on winning games rather than prediction accuracy. RL network has the same structure as SL network.

Value network to predict winner.

Position *s* is evaluated for both players. This is rather estimated strongest value for RL policy than optimal value under perfect play.

Searching with policy and value networks

AlphaGo combines the policy and value networks in a MCTS algorithm. Goal is to select action by lookahead search. AlphaGo used asynchronous multi-threaded search that executes simulation on CPUs and in parallel computes policy and value networks on GPUs. The final version of AlphaGo used 40 search threads, 48 CPUs, and 8 GPUs.

Result

AplhaGo is program playing Go game using deep neural networks and tree search.

To evaluate AplhaGo the researches ran an internal tournament among its variants and other Go programs. Commercial ones - Crazy Stone, Zen and open source - Pachi and Fuego. All of them are based high-performance MCTS algorithms. The results of the tournament showed that AlphaGo ran in single machine is much stronger than the other programs winning 494 out of 495 games (99.8%).

Distributed version of AlphaGo won 5 – 0 against Fan Hui, professional player and European champion. That means AlphaGo can play on level of strongest human players and has achieved one of the **“grand challenges”** of artificial intelligence that were seen a decade away.