Deepmind's AlphaGo paper review

Author: Kiran Ramineni

This research report is about summarizing findings from the Deepmindâ€™s paper **Mastering the game of Go with deep neural networks and tree search** 1 about AlphaGo game they developed. I will be covering how this research is related to learning we did in the class and applying the similar strategies in the agent playing code.

Problem statement

AlphaGo has state space explosion problem since the number of states since the board is *19x19* that leads to 10^171 states. Chess game has only 10^47 in *8x8* board. Also, the number of total possible moves are 150 vs 80 compared to chess.

AlphaGo has two main strategies for finding the best move.

Policy deep learning network

Policy network is based on the supervised learning where it evaluates whats the best move for the player to win. It is like the hybrid heuristic to find the best move using different pruning techniques. * It uses previous guided moves (supervised learning from human moves) to find the best moves. * Since there are a lot of moves possible, there is a need to cut down search space. Iterative deepening would time out searching through the huge space. This is done by looking at a specific section of the board (looking at the last move by the opponent and own move) and shrinking down the board to manageable size like *9x9*.

Value deep learning network

Value network is a different heuristic that keeps track of board positioning and evaluates the moves suggested by policy network to tell whether its a good or bad comparing the winning position in the board Value networks job is isolated from policy network and hence it can be run parallel.

MonteCarlo search

Given a move, we learnt there are several techniques like Depth first vs Breadth first to navigate the Search tree. The problem with AlphaGo tree is that it has so many branches and the depth is huge compared to chess game. Going depth first vs breadth first might not identify the best position in the given time while doing the iterative deepening. MonteCarlo simulation provides a good alternative since it mixes the possible moves in random fashion so we might reach the optimal route faster.

Summary

By using two different heuristics (policy for finding moves and value for evaluating moves) and adopting MonteCarlo simulation to find the deep layers faster, AlphaGo has done applying AI concepts to beat the world champion several times with 100% winning ratio.