Artificial intelligence nanodegree research review: AlphaGo

Reviewer: Naren Karthik

The game of Go is considered one of the most challenging game for an AI to win. This is because, unlike chess which has approximate breadth (b) as 35 and depth (d) as 80, Go has approximate breadth around 250 and depth around 150. As we already know to calculate end state of isolation games an AI needs to do b^d calculations recursively. In the case of Go the search tree is 250^{150} which is exponential large compared to game of chess.

An algorithm called Monte Carlo tree search (MCTS) is the widely used algorithm to solve this problem. In fact the strongest Go programs before AlphaGo was mainly built using MCTS.

Deepmind's novel approach:

Deepmind's AlphaGo is no exception in using MCTS to solve Go. However AlphaGo uses a recent advancement in AI called deep convolutional neural network (CNN) in combination with MCTS to solve the game of Go.

AlphaGo address the problem of enormous search space 250^{150} (as we discussed above) by reducing the breadth and depth of the search space. This is no different than any other prior approaches but how AlphaGo reduces the breadth such that only highly probable and high impact nodes are selected for evaluation and how AlphaGo reduces the depth by building an efficient evaluation of the nodes have what made AlphaGo the best AI available for Go today which has won against world human champions.

The overview of this novel approach is explained below.

The Go board is passed as a 19X19 image to the CNN. First a supervised learning (SL) policy network is trained on expert human moves. By doing so the network learns the impactful expert moves based on the current board position. Then a reinforcement learning (RL) is trained based on the game play by the policy gradient network which is essentially the AI playing against itself based past gameplay. This improves the AI to make more winning moves based on the board position fed to it. Since this network has got better at choosing winning moves the nodes it has to traverse gets for a given board position gets narrowed there by reducing the breadth of the search space.

The value network is also a CNN which is trained by regression to output a single value. Value network can be considered as equivalent of evaluation function from the game playing agent discussed in the AIND classes. The value network reduces the depth of search by providing a better scoring on the outcome of the game based on the move selected by the policy networks. To understand this let us assume the same game we have in our AIND isolation project and let's consider a bad evaluation function and a good evaluation function. In this scenario the bad evaluation function has to traverse to the end of game to find who will win the game, where as a good evaluation function can do the same with far less node traversals based on how good the evaluation function is. Now consider the value network as a bad evaluation function at the beginning and as we train the network the bad evaluation function encoded in the network starts to become more efficient and becomes far better at evaluation (thereby reducing the search depth needed to do an effective scoring) than human built evaluation functions.

Using the above mentioned Policy and Value network AlphaGo is able to better estimate & predict the best possible move and approximate the winning end states thereby giving it the power to master one of the complex game which was not expected to me mastered by an AI at least for another decade.