AI Engineer Nanodegree
Udacity

July 11, 2017

Research Review

AlphaGo Review

- **What is Go**: The game of Go is one of the challanging classic game with high space complexity because of enormous search space ($b \sim 250, d \sim 150$) and very high time complexity because of difficulty in evaluating board position and moves.

- **What is AlphaGo**: The AlphaGo [1] program was designed to play Go at the level of professional human players. AlphaGo incorporates a number of novel techniques to overcome the difficulty in intractible search space to evaluate board position and moves. These novel techniques are development of mutiple "policy networks" to guide the search algorithm toward the best moves. These are *Supervised learning (SL) of policy networks* and *Reinforcement learning (RL) of policy networks*, *Policy and value networks* and *Monte Carlo Tree Search(MCTS).*

- **SL policy network**: The supervised learning (SL) policy network is a deep neural network trained on a large database of expert player moves. The input layer represents the current position as 19x19x48 image stack, and the final layer uses softmax to output a distribution over possible moves.

- **RL policy network**: A reinforcement learning (RL) policy network is used to correct the decision made by SL network. The SL network predict the most likely move chosen by an expert player, RL network help to pick a winning move. RL network is trained by initializing it with the weights from the SL network, then refined through self-play.

- **Policy and value networks**: The value network is a third neural network. This network provides a value function that can score any board position in one evaluation. The network structure is similar to the SL and RL networks, but the final layer is a single node that represents the probability of winning from the given board position. Training was done using reinforcement learning by selecting random board position from a game database, then playing the game to the end using the RL policy network.

- **Monte Carlo Tree Search(MCTS)**: The search algorithm in AlphaGo is also novel. It uses an aynchronous version of Monte Carlo

Tree Search. Moves are selected for expansion based on the SL policy network. Instead of simulating to end game, the resulting positions are evaluated two ways: once using the value network, and again using a random rollout using the fast rollout policy. The two values are combined using a mixing parameter.

- **CPU/GPU Parallelization**: Because the neural network evaluations are much slower than traditional heuristics, the AlphaGo uses a multithreaded, asynchronous implementation of MCTS. Policy and value network evaluations are done in parallel on GPUs, while the overall search and random rollouts are done in multiple threads on CPUs. A distributed version of AlphaGo was used for the human matches, running on 1,202 CPUs and 176 GPUs.

- **Neural Network**: Applying deep neural networks to the problems of move selection and position evaluation allowed it to succeed in a very large search space, in spite of the fact that it evaluated far fewer positions than successful Chess playing programs. The fact that the neural networks were trained directly from game play and did not rely on game-specific heuristics makes the program simpler, and also suggests that the approach will be easier to generalize to other problem domains.

References

1. D. Silver et al., Mastering the game of Go with deep neural networks and tree search, Nature, vol. 529, no. 7587, pp. 484489, 2016.