Research Review

Title: Mastering the game of Go with deep neural network and tree search

1. Research Goal:
   * Conquer the most challenging of classic game “GO” with Artificial Intelligent, which can defeat a human professional player most of the time.
   * Solve the problem of the enormous search space and the difficulty of evaluating board positions and moves in “GO”.
2. Current Problem:
   * Exhaustive search is infeasible. And, existing ways for position evaluation, like minimax, alpha-beta pruning, Monte Carlo tree search techniques, are somewhat not that powerful in playing “Go”.
3. Techniques:
   * Deep Neural Network: Value network & Policy network
     + About Policy Network, which consists of deep convolutional layers (13 layers)
       - Type: Fast Rollout (*pπ*(*a*|*s*)), Supervised Learning (SL, *pσ*(*a*|*s*)), Reinforcement Learning (RL, *pρ* (a|s))
       - Aim: Predict & Select a move
       - Input: The representation of the board state (19 x 19 image)
       - Output: A probability distribution over all legal moves
     + About Value network:
       - Aim: Evaluate the board positions
       - Input: The board state
       - Output: A single prediction instead of probability distribution
       - This network’s architecture is similar to policy network, but outputs a single prediction instead of a probability distribution, which means it memorized the game outcomes rather than generalizing to new positions
   * Monto Carlo Tree Search (MCTS)
     + AlphaGo combines the policy and value networks in MCTS algorithm, which can select actions by lookahead search.
     + The edge (state, action) of the search tree stores an action value Q(s, a), visit count N(s, a), and prior probability P(s, a).
     + The search tree is traversed by simulation (descending the tree without backup)
     + The leaf node is evalualted in 2 different ways, by value network or by the outcome of a random rollout using fast rollout policy *pπ* .
     + Each edge accumulates the visit count and mean evaluation of all simulations passing through that edge. The algorithm chooses the most visited move from the root position.
4. Results & Discussion:
   * Training results of policy network:
     + Trained with 30 million positions from KGS Go server, the accuracy of SL policy network to predict expert moves is 57.0%, and 55.7% using only raw board position and move history as inputs. (State-of-art is 44.4%). On the other hand, the accuracy of Fast Rollout policy network is 24.2%. Fast Rollout policy network spent 2 μs to select an action while SL policy network took 3 ms.
     + RL policy network won more than 80% of games against the SL policy network. It also won 85% of games against Pachi, which was the strongest open-source Go program then with no search at all.
   * AlphaGo performance
     + Single machine AlphaGo has 99.8% chance of winning against other Go program like Pachi. Even without rollouts, AlphaGo still exceeds the performance of all other Go programs, which demonstrates that value networks provide a viable alternative to Monte Carlo evaluation in Go.
     + AlphaGo with mixed approach (value network combined with Monte Carlo tree search) has more than 95% chance of winning against AlphaGo’s variants, which means that two position-evaluation mechanisms are complementary.
     + Distributed AlphaGo program defeated Fan Hui, who is a professional 2 dan, in a formal five-game match. AlphaGo won the match (5 vs. 0)