AlphaGo Review

Go is challenging for AI due to extraordinarily large search space and difficulty in evaluating board positions/states and moves. In this study, convolutional neural network (CNN) 'policy networks' and 'value networks' were used to select moves and evaluate board positions, respectively. Also, Monte Carlo tree search (MCTS) algorithm combined with policy and value networks was used to select moves by look-ahead search.

Given a board state, how does the game agent sample possible actions? One way is training the agent to predict human expert moves. So first, the study trained a supervised learning (SL) policy network from 30 million expert human moves. The network takes representation of board states as input and outputs a probability distribution of possible moves. The network achieved an accuracy of 57% on predicting human expert moves.

However, the goal was not only mimicking human playing behavior but also winning games. Thus, second, the study aimed to improve policy network through reinforcement learning (RL). The weights of the SL policy network were updated by maximizing the expected outcome taking winning as reward and losing as punishment. The RL policy network won more than 80% games against SL policy network when played head-to-head.

Go is a game of perfect information. Assuming both players using the above RL policy network to play, for every board position, there is a value function predicting the outcome of the game. Finally, the value function was approximated by training a RL value network from 30 million distinct self-play positions each sampled from separate game, with the RL policy network playing each game with itself until the end of the game. The network takes representation of board states as input and outputs a prediction value, indicating the winner of the game. The network achieved a mean square error (MSE), between the predicted value and corresponding outcome, of 0.234 on the testing dataset.

Now that the board position evaluation and move selection have been solved by training CNN, the remaining problem is searching during game playing. The study used an MCTS algorithm to search and select actions. In the tree to be searched, a leaf node (board state) may be expanded by applying the policy network to get a prior probability for each legal action after it. A leaf node is evaluated by combining value network prediction and random rollout outcome. An edge (action) is chosen by maximizing the action value plus a bonus; the action value is the average of leaf node evaluations of all simulations passing through the edge; the bonus is proportional to its prior probability and decays with visit count to encourage exploration. The tree is traversed by simulation, starting from the root state. At each time step, an edge is selected from a leaf node according to the above rule. At the end of simulation, for each traversed edge, the action value and visit count are updated. Once the search is complete, the most visited move from the root position will be chosen.

Using the above algorithm, AlphaGo decisively defeated all the MCTS-based programs at that time. AlphaGo also won all the 5 matches with one of the strongest human expert players. As to

the variants of AlphaGo, the version that combines both value network prediction and random rollout outcome in evaluating leaf nodes performs better compared with versions that only use one of the two measures in evaluating leaf nodes.