Research Review
Mastering the Game of Go with Deep Neural Networks and Tree Search
Reviewed by Jenny Hung

Problem

Like all other games with perfect information, Go also as an optimal value function, $v^*(s)$, that determines the outcome of the game from all possible board position and assuming perfect play of all players. Theoretically, games of this type can all be solved by recursively computing the optimal value function in a search tree. However, this theoretical solution considers b^d possible sequences of moves, where b is the game's breadth (number of moves per position), and d is its depth (game length). Improvement to the theoretical solution, such as truncating the search depth, does not help Go although it can dramatically improve the outcome of Chess. Other improvement, such as sampling actions from a policy $p(a|s)$ that is a probability distribution over possible moves a in position s, helps other types of games such Scrabble but not Go.

Solution Approach

Prior to Silver et al proposed the new solution, the best approach for solving Go uses Monte Carlo tree search. As simulations are executed, the search tree grows larger and deeper and the relevant values become more accurate. Further, the policy used to select actions during search is also improved over time, by selecting children with higher values. This policy converges to an optimal play asymptotically. However, prior work using this approach has been limited to shallow policies.

In 2016, Silver et al proposed a new hybrid approach that had shown strong promise in solving Go. The proposed approach combines deep convolutional neural networks as well as the Monte Carlo tree search, where the former reduces the effective breadth and depth of the search tree, and the latter helps to increases the accuracy of the model. More concretely, the neural networks were trained using a pipeline consisting of several stages of machine learning. First, a policy network was training using supervised learning using human expert moves. Then a policy network was trained using reinforcement learning, which modifies the goal to win games, rather than maximizing predictive accuracy. Subsequently, a value network is trained to predict the winner of games as predicted by the policy network generated from the reinforcement learning.

Result

To evaluate the new solution approach, an internal tournament was run to compare AlphaGo, whose solution engine is based on the approach proposed by Silver et al, as well as other existing Go solution algorithm. AlphaGo has 99.8% winning rate against other algorithms. In addition, another tournament was run between AlphaGo and Fan Hui, a professional Go player. In that tournament, AlphaGo played a full game without handicap, and defeated the professional player. This was indeed a milestone for artificial intelligence.