Andy Miller
AlphaGo - Review

Traditionally, computer game solving agents have followed a strategy of solving games that involved recursively searching a decision tree to approximate or see all possible moves within a game. The agent would then pick the following move that would maximize its position within the game. This strategy is optimal for zero-sum games so long as the resources are available to adequately approximate or "see" all possible moves. The game of go has an extremely large search space. Like chess, it is infeasible to compute every possible move in the search space. The search space must be reduced and the optimal move for the agent must be optimized.

The AlphaGo system deals with the large search space in many ways. It also deals with the large search space is by utilizing powerful state of the art hardware. The distributed AlphaGo system is large and compex with a high level of computing power. The actual distributed system employs 40 search threads, thousands of CPU's for general processing and hundreds of GPUs for parallel processing (this is taken directly from the Deep Mind AlphaGo paper). Another way the AlphaGo system predicts its best move using several different novel software techniques. It uses two main algorithmic networks for deciding its best move. One network defines the decision making policy for each move (reducing search space) and the other network defines the predicted value for each move (maximizing the agent's outcome). These networks are combined with a Monte Carlo Tree Search algorithm so that the system contains one trainable system. The novelist aspect of the system is its optimization. The network is optimized by deep learning techniques that strengthen / weaken connections between the nodes. This optimizes the importance of the edges between value and policy nodes so that stronger connections are created towards winning strategies and weaker connections are forced upon losing strategies. The specific training that was implemented upon the AlphaGo system can be summarized as three steps. First the system was trained by watching and predicting the moves of expert players. This is the supervised learning aspect. It should be noted that supervised learning was not done on the value networks (they're it's easily defined) but reinforcement training was implemented on the value networks to coincide with the supervised learning and reinforcement learning that is done on the policy networks. The system was then tested against other leading competitive Go programs. Once AlphaGo was considered "the best" it is then trained against itself through reinforcement learning and tests are done intermittently so that overfitting is avoided. After that the program evaluation was completed and AlphaGo was essentially considered ready for competing against human players.