AlphaGo: A Quick Summary

This review gives a concise summary of the article "Mastering the game of Go with deep neural networks and tree search" written by the Google DeepMind team, which talks about the algorithm behind *AlphaGo,* a Go playing agent, and the importance of it successfully performing at the highest level.

The game of Go is seen as the most challenging game for AI agents due to its enormous search space and the difficulty of evaluating moves and positions. Historically, the strongest Go programs have been based on Monte Cristo tree searches (MCTS) enhanced by policies that emulate the style of expert human players. However, this approach has only been able to produce strong amateur level agents. Therefore, the DeepMind team decided to use a new approach that uses a combination of MCTS with value networks for evaluating the position and policy networks for move selection. The networks use an image of the board positions to construct a representation of the position using convolutional layers. These deep convolutional neural networks were trained with a combination of supervised learning (SL) from human expert games and reinforcement learning (RL) from games of self-play (p. 1).

The SL policy network was trained to predict expert moves from the current state of the board using 30 million positions from the KGS Go Server. The RL policy network is identical in structure to the SL policy network except it was trained by playing games against a randomly selected previous iteration of the network. Feedback was returned to the network through a reward function that returned 0 for non-terminal states, +1 for winning, and -1 for losing. When playing head-to-head the RL policy network won more than 80% of games against the SL policy network and 85% of games against Pachi, a state-of-the-art open-source Go program (p.2). When training the value network to predict the outcome of a game based on the board position, the DeepMind team found that using SL resulted in a network that memorized the outcomes and was unable to generalize for new positions. To remove this problem, a new data set consisting of 30 million board positions from different games was created. Each of the sampled positions

was played between itself and the RL policy network until the game ended. Lastly, the networks were combined in a MCTS algorithm that transverses the search tree and selects the best next move (p. 3).

AlphaGo was then evaluated in an internal tournament against other high-performance Go algorithms where *AlphaGo* readily beat each opponent. The result suggested that other algorithms were well below the level of *AlphaGo*. Finally, *AlphaGo* competed against Fan Hui, a professional Go player and European champion, and won the match 5 games to 0. This was the first time a computer was able to defeat a professional player, an achievement that was thought to be at least a decade away. This sheds hope on other domains of artificial intelligence that are believed to be impossible or attainable only in the distant future.