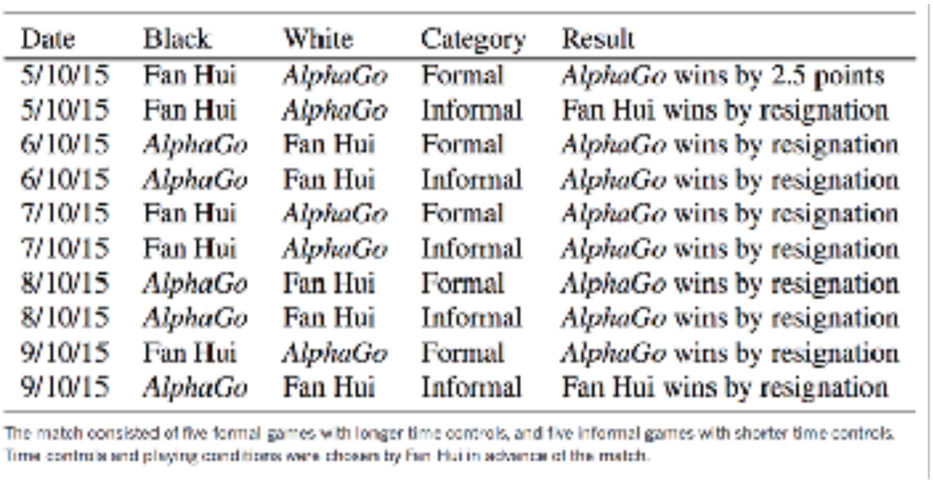
This research paper explained the techniques behind the famous Go game agent, AlphaGo, which is developed by Google DeepMind, and how it is different from the prior work. Previous, the game agent of Go often was based on the algorithm, called Monte Carlo tree search (MCTS). Monte Carlo tree search is a heuristic search algorithm, and its focus is on the analysis of the most promising moves, expanding the search tree based on random sampling of the search space. DeepMind team introduced the use of neural network to its logic. There are two types of network used in the game play. Value network is to evaluate board positions, which is similar to the scoring heuristic we have in the isolation game agent, and policy network is to select the best move in the current game.

DeepMind developed a pipeline to train these network, and these networks are later combined with MCTS to generate the moves. The first stage of the pipeline focus on training the network so it can effectively predict expert moves using supervised learning. The second stage of the training pipeline aims at improving the policy network by policy gradient reinforcement learning. The final stage of the training pipeline focuses on position evaluation, estimating a value function that predicts the outcome from position of games played by using policy for both players. During the play, AlphaGo combines the policy and value networks in an MCTS algorithm that selects actions by lookahead search. Since evaluating policy and value networks requires several orders of magnitude more computation than traditional search heuristics, the hardware to support these techniques are also very crucial. The team use 40 asynchronous search threads, 48 CPUs, and 8 GPUs in its final version of AlphaGo.

To evaluate AlphaGo, the team hosted an internal tournament among variants of AlphaGo and several other Go programs, similar to the isolation project tournament.py. The opponents include commercial programs, Crazy Stone and Zen, and the open source programs Pachi and Fuego. The final results indicate that single-machine AlphaGo is many dan ranks stronger than any previous Go program, and it has 99.8% winning rate against other game agents in the total of 495 game. To understand how well AlphaGo is performing against other programs, AlphaGo also played with four handicap stones with the opponents, and the result was amazing. AlphaGo won 77%, 86%, and 99% of handicap games against Crazy Stone, Zen and Pachi, respectively. Also, the distributed version of AlphaGo was significantly stronger than the single-machine version. Finally, the distributed version of AlphaGo played against Fan Hui, a professional 2 dan, and the winner of the 2013, 2014 and 2015 European Go championships. The result is attached below.



Date	Black	White	Category	Result
5/10/15	Fan Hui	*AlphaGo*	Formal	*AlphaGo* wins by 2.5 points
5/10/15	Fan Hui	*AlphaGo*	Informal	Fan Hui wins by resignation
6/10/15	*AlphaGo*	Fan Hui	Formal	*AlphaGo* wins by resignation
6/10/15	*AlphaGo*	Fan Hui	Informal	*AlphaGo* wins by resignation
7/10/15	Fan Hui	*AlphaGo*	Formal	*AlphaGo* wins by resignation
7/10/15	Fan Hui	*AlphaGo*	Informal	*AlphaGo* wins by resignation
8/10/15	*AlphaGo*	Fan Hui	Formal	*AlphaGo* wins by resignation
8/10/15	*AlphaGo*	Fan Hui	Informal	*AlphaGo* wins by resignation
9/10/15	Fan Hui	*AlphaGo*	Formal	*AlphaGo* wins by resignation
9/10/15	*AlphaGo*	Fan Hui	Informal	Fan Hui wins by resignation

The match consisted of five formal games with longer time controls, and five informal games with shorter time controls. Time controls and playing conditions were chosen by Fan Hui in advance of the match.