Research Review:
AlphaGo by the DeepMind Team

Summary of AlphaGo

AlphaGo is based on a combination of deep neural networks and tree search, that plays Go beyond the strongest human player level.

Game of Go has long been viewed as the most challenging of classic games for artificial intelligence for two main reasons:

1. Search space is really huge - b^d (b≈250, d≈150) where b is the game's breadth and d is its depth (game length).
2. Complexity to write evaluation function to determine who is winning.

DeepMind team get over these two problem using two trained Neural Network

1. **Policy Network:** Policy network provide the probability distribution of moves. It learns to predict - for any particular position what's the most likely moves were to be played. So instead of looking at one position of all possible legal moves, it looks for top 3 or top 5 most likely moves were taken into consideration.This reduces down the breadth of the search space. Policy network is first trained by Supervised learning using 30 million positions data from the KGS Go Server. (13 convolutional layers policy network). The second stage of policy network is trained using policy gradient reinforcement learning. This improved the policy network further.
2. **Value Network**: Value network evaluate a particular position and determines who is winning (0 - white or 1 - black). The value network is trained using reinforcement training. Initially using KGS data, the value network memorised the game outcomes rather than generalising to new positions. Achieving a minimum MSE(mean squared error) of 0.37 on the test set, compared to 0.19 on the training set. This is mainly because of the complexity of Go game - as successive positions are strongly correlated, differing by just one stone. To mitigate this problem, alpha-go is made to play against each other and generated new self-play data set consisting of 30 million distinct positions, each sampled from a separate game. Later, the best version of Alpha is trained by playing against the previous best version of AlphGo Eventually value network got better.

AlphaGo combines the policy and value networks in an MCTS (Monte Carlo tree search) algorithm that selects actions by lookahead search. Evaluating policy and value networks requires several orders of magnitude more computation than traditional search heuristics. To efficiently combine MCTS with deep neural networks, AlphaGo uses an asynchronous multi-threaded search that executes simulations on CPUs and computes policy and value networks in parallel on GPUs

Summary of Results

Notable matches of AlphaGO:

1. Mar 2016 - AlphaGo vs Fan Hui (AlphaGo won by 5-0)
2. Oct 2016 - AlphaGo vs Lee Sedol (AlphaGo won by 4-1)

AlphaGo was able to beat the strongest human player, thereby achieving one of artificial intelligence's grand challenges. It's important to note that AlphaGo evaluated thousands of times fewer positions than Deep Blue did in its chess match against Kasparov compensating by selecting those positions more intelligently, using the policy network, and evaluating them more precisely, using the value network—an approach that is perhaps closer to how humans play. Furthermore, while Deep Blue relied on a handcrafted evaluation function, the neural networks of AlphaGo are trained directly from gameplay purely through general-purpose supervised and reinforcement learning methods. Go is exemplary in many ways of the difficulties faced by artificial intelligence a challenging decision-making task, an intractable search space, and an optimal solution so complex it appears infeasible to directly approximate using a policy or value function