With AlphaGo, the DeepMind Team tackles the challenge of playing the game of Go at a professional level or higher. Go is an ideal choice as an AI problem because of its incredibly large search space, and optimal solution thought to be too complex to simply approximate. In other words, creating an evaluation function for intermediate states of the game is a very non-trivial task. To overcome these difficulties, AlphaGo combines Monte Carlo tree search with deep neural networks to evaluate and select board states and actions.

Two types of neural networks are trained for AlphaGo: policy networks to sample actions to be further evaluated and searched, and evaluation networks to estimate the outcomes of board states. During play, a deep neural network optimized through reinforcement learning and a softmax classifier are used, called RL Policy and Fast Policy hereafter. The RL Policy network is trained in a pipeline that combined supervised learning with reinforcement learning.

In the first step, a deep neural network is trained on state-action pairs randomly sampled from expert games, to predict the action corresponding to a state. Next, the RL Policy network has its parameters initialized to those learned through this supervised learning step. These parameters are then optimized through policy gradient reinforcement learning in games of self-play between the current RL policy network and randomly selected previous iterations of itself. The randomness prevents overfitting to the current policy. The reward function is 1 for a win, -1 for a loss, and 0 at non-terminal states.

Once the RL Policy network is trained, the Value Network is trained using supervised learning with state-outcome pairs. Data from expert games is augmented by results of self-play, in order to improve generalization of the value network.

Monte Carlo Tree Search, which was a recent breakthrough in Go playing systems at the time of AlphaGo, is combined with the neural networks to explore and select moves. In MCTS, the search tree is traversed by simulation, completing full games before backing up. The number of games simulated is not exhaustive, and therefore must be randomly sampled, hence the name. When evaluating certain candidate states, results of full rollouts using the Fast Policy network are combined with the output of the Value Network. To facilitate these massive computational requirements, asynchronous multi-threaded search is used for MCTS, while the deep neural networks compute in parallel using GPUs.

AlphaGo achieved a 99.8% win rate against other programs. More significantly, it defeated the European Go Champion, thus outperforming a human professional for the first time in history. It did this while evaluating thousands of times fewer positions than Deep Blue did in its own seminal chess match, and did not rely on handcrafted evaluation functions.

Regarding AlphaGo's technical and algorithmic innovations, it's SL policy network predicted expert moves with 57% accuracy, compared to the previous state of the art of 44.4%. Small improvements in this accuracy were shown to translate to large differences in playing strength. The RL Policy was also shown to win against the SL policy network 80% of the time, and 85% of the time against a then state-of-the-art program without using search. The value network also had positive results, approaching the accuracy of MCTS using the slow RL Policy results, using 15,000 times less resources, proving itself to be a viable alternative. The fact that a combination was used showed that the value network and MCTS were complementary mechanisms in evaluating board states.