The goal of AlphaGo was to create a program capable of beating a professional level human player using a combination of supervised learning from a data set containing human expert games and reinforcement learning by playing itself. It combines State-of-the-art Monte Carlo Tree Search, value networks, and policy networks. The major difficulties to overcome were the extremely large search space, a challenging decision-making process and an optimal solution that seemed impossible to predict using only a policy or value function. The overcome these difficulties, the search space was reduced in two ways. the depth of the search could be reduced by evaluating position. In other words, replacing part of the search tree below a certain state *s* with an approximate value function that predicts the outcome of said state. The breadth of the search could also be reduced by creating a probability distribution of possible moves at sate *s* and maximizing search depth according to that probability distribution. The probability distribution becomes more accurate as more simulations are performed by using Monte Carlo rollouts to estimate each state value.

The neural network training sequence contains three stages. The first stages used supervised learning of the policy network, which selects moves, by predicting professional human moves. The policy network is then trained using reinforcement learning that aims to maximize the outcome of winning games against a random instance of the policy network from a data set containing many previous instances. Finally, the third stage aims to train the value network, which evaluates board positions according to the policy, by using regression to predict whether the player wins in positions from the previously mentioned data set.

The neural networks were implemented using an Asynchronous Policy and Value Monte Carlo Tree Search algorithm (APV-MCTS). It uses edges that contains a prior probability, a Monte Carlo estimates for the value of the action and a combined mean value for the action, for the corresponding state and action. The algorithm proceeds in four stages for each simulation. The first stage consists of a search from the root to a leaf node reached at a specified time that prefers actions with high prior probability, low visit count and high action value. If the leaf node at the specified time is not present in a queue created by the value network then it is added. The second stage consists of a rollout phase from the aforementioned node to the end of the game with actions selected based on the rollout policy. The rollout policy is “a linear softmax policy based on fast, incrementally computed, local pattern-based features” meaning that it compares the best action from the search tree and the policy network to a 3x3 pattern around the current move and the previous move and if a match is found then the move is assigned a high probability and played.

An alternative distributed APV-MCTS algorithm was also created which produced best results which consists of a main machine that executes the search while remote worker CPUs and GPUs that performs rollouts and policy and value network evaluations respectively. Additionally, performance of the different algorithms increases as the number of GPU and threads increases. The AlphaGo program is capable of winning over 99% over other GO programs and the first to beat professional level human players consistently.