AlphaGo is a program that uses novel artificial intelligence techniques to play the game of Go. The program makes extensive use of different neural networks along with different search techniques to achieve a very efficient Go play.

The neural networks used by the program use a representation of the board that is passed in as a 19x19 image and use convolutional layers to construct representations of the positions in the board. These networks are used to reduce the search depth and breadth of the search space.

The networks are trained through different stages, each one aimed to provide a specific outcome during each turn of the game. The first one is trained through supervised learning derived from expert human moves. A fast rollout policy is also trained to quickly sample actions. The a reinforced learning policy network is trained in order to optimize the outcomes of the supervised learning network. Finally a value network is trained to predict the winner of self play made by the reinforced learning network.

The supervised learning (SL) network is trained using randomly selected state action pairs with a carefully selected gradient to increase the likelihood of a human move. The states were taken from the KGS Go server. The network predicted 57% expert moves using all inputs and 55.7% using raw board position and move history. The results showed that small improvements in the accuracy of the network led to large improvements in playing strength. A fast rollout network was also trained achieving an accuracy of 24.2% that just took 2 microseconds to select an action, compared to 3 milliseconds for the SL policy network.

A reinforced learning (RL) network was made using the same structure as the SL network with the objective to improve the outcomes of SL network. It is used by playing games against a previous iteration of the current policy network using a reward function. Evaluating the performance of the RL network against the SL network, the RL network won 80% more than the SL network. When tested against an open source program called Pachi, the network 85% of the games without any extra search.

The final stage involved the creation of an evaluation network through reinforced learning that could predict the outcome of a certain position in the game. This network uses the same structure as the previous networks and an estimation for the ideal value function is made through the RL network of the previous stage. The outcome of the network is a single prediction instead of a probability distribution. The initial approach to the creation of the value network led to an overfitting of the network because of the strong correlation between the positions generated by the complete games of the RL network. To overcome this, a new self-play data set of 30 million new positions was used and the overfitting became minimal.

The program combines the networks described above in a Monte Carlo Tree Search algorithm. A lookahead search is performed by traversing the search tree using the neural networks mentioned above. Each edge of the tree stores an action value, visit count and prior probability. Using equations to evaluate the value of each node with all the simulations made by the algorithm, the next move is selected.

In order to provide the necessary computational power to run the program, 40 CPU threads are used in parallel for the simulations, and the valuation of the policy networks were done in 8 GPUs.

To evaluate the performance of AlphaGo, a tournament with some open source and proprietary programs was made. The results of this tournament gave a 99.8% win rate for AlphaGo without any handicap for the programs, and a 77%, 86% and 99% win rates giving the other programs a handicap of 4 stones. Finally, AlphaGo played 5 games against the renowned world champion Fan Hui, winning all of them.