Go:

Goal: As of today, 3rd June, 2017, google's alpha go has retired since it achieved it's goal to defeat the best human player in the game of Go, which is a chinese board game.

It's one of the challenging games, the number of combination of the moves makes it a difficult problem to solve.

Normally game requires b^d, where b is the breadth of the board and d is the depth, that makes exhaustive search infeasible. To solve this, depth is reduced by two principle:
1. Evaluate the board positions
2. Select the moves policty

Position evaluation by truncating the search tree at state s and replacing the subtree s by an approximation value function that predicts the outcome from state s. Secondly, the policy to select the moves, the breadth first search may be reduced by sampling actions from a policy p(a|s) that is a probability distribution over possible moves a in position s.

Monte Carlo tree search(MCTS), search to maximum depth without branching at all, by sampling long sequences of actions for both players from a policy p. Initial training by the Go team, similar to the deep convolutional networks, uses convolutional layers to construct a representation of the position (board position as 19*19 image).

There flow is:
1. Supervised learning policy network directly from expert human moves, this provides fase, efficient learning updates with immediate feedback and high-quality gradients. Along with a fase policy that can rapidly sample actions during rollouts.
    The SL policy network p(a|s) alternates between convolutional layers with weights and rectifier nonlinearities. A final softmax layer outputs a probability distribution over all legal moves a. A faster but less accurate rollout policy using a linear softmax of small pattern features with weights.
2. Reinforcement Learning policy network that improves the SL policy network by optimizing the final outcome of games of self-play. It is similar to the SL policy network and its weights p are initialized to same values. Playing the games between the current policy network and a randomly selected previous iteration of the policy network stablizes training by preventing overfitting.
3. Train a value network that predicts the winner of games played by the RL policy network against itself. This neural network has a similar architecture to the policy network, but outputs simgle prediction instead of a probability distribution. The naive approace of predicting game outcomes from data consisting of complete games leads to overfitting. The problem is that successive positions are strongly correlated, differing by just one stone, but the regression target is shared for the entire game. It is solved by using the KGS data set.

Alpha Go combines the policy and value networks in an MCTS algorithm that selects actions by lookahead search. It is worth noting that the SL policy network perfomed better in alphaGo than the stronger RL policy network presumably because humans select a diverse beam of promising moves whereas RL optimizes for the single best move.

Conclusion:
    alphaGo is a combination of deep neural networks and tree search, that plays at the level of the strongest human players, thereby achieving one of artificial intelligence’s grand challenges. It should be noted to use the neural network evaluation with Monte Carlo rollouts.