Google DeepMind took on a task which was considered one of the most challenging problems in the modern day AI. The goal was to teach a machine to play the game of Go better than human experts.

This was believed to a tough problems for two main reasons. First, its search space is overwhelmingly large (>>10^200, much bigger that chess). Considering that traversing the whole search space is not feasible, different techniques and approaches are employed to find a reliable proxy for the best next move at each state of the game. In addition to big search space problem, the objective of Go is more difficult to learn or hard-code compare to for example chess. In chess, the goal is " to checkmate the opponent's king by placing it under an inescapable threat of capture." [wikipedia]. The objective of Go "is to fully surround a larger total area of the board than the opponent" [wikipedia] on a board of 19x19. Therefore, variations in the end state (end of the game) is much higher for Go compare to chess.

The evaluation approach consists of two components: building a value network and a policy network. Value network simply captures the current state of the game (i.e., board positions) and policy network is used for finding the next best move (acts similar to an evaluation function). Both are evaluated using deep convolutional neural network models, which help with reducing the depth and breadth of the search space. Convolutional networks are shown to be efficient in processing images and perceiving its content. Therefore, they are a reasonable choice for Go as different states of the game can be capture in a single image. Besides, convolutional neural network makes it possible to have a more manageable number of free parameters that requires fitting. The models are first built with supervised learning by using the actual record of the games played by human experts, and then improved by using reinforcement learning. Reinforcement learning, In short, improves the agent performance by playing against itself and uses feedback (i.e., the game result) to enhance the chance of winning.

Performance of the game agent built using only value and policy network is comparable with, if not better than, the state-of-the-art methods of the time, based on Monte Carlo Tree Search (MCTS), with significantly less computational demand. MCTS basically performs many initially random games to the end and improves the performance of the subsequent simulations. Eventually, MCTS reduces the randomness and converges to an optimal solution.

However, combining these two powerful approaches lead to a 99.8% winning rates against other Go programs and the even defeated the human European Go champion by 5 games 0. This is considered a milestone in the game search algorithms, as the solution to this game believed to be another decade away.