AI agents are designed to play games by exploring the possible states for the next move in a way to maximize agent’s own utility from the move and minimize the opponent’s utility. The complexity of search problem is determined based on the branching factor i.e. the number of possible immediate next moves and the depth of the search graph i.e. levels to look ahead in the game with each level alternating between agent and its opponent. The complexity of a game increases exponentially even with a linear increase of branching factor and depth.

The game of Go is incredibly complex because the number of possible states which increases the search space to identify the best next move for an agent. The researchers at Google developed an AI agent for Go that combined deep neural network and tree search capabilities to intelligently select the most efficient next move. The deep neural networks were trained by a combination of supervised learnings from human experts and reinforcement learnings from the self-play. Prior to Alpha Go implementation, AI agents for the game of GO were based on Monte Carlo tree search (MCTS). MCTS technique uses simulations to predict the value of each state in a search tree.

Convolutional neural networks work by many layers with each layer focusing on a localized representation of the problem space. Alpha Go uses convolutional neural networks by representing the board position as a 19 X 19 image.

Alpha Go program defeated a professional human player by 5-0. This is the first time a computer program defeated a human player beginning a new era for the field of artificial intelligence.