Jonathan S Kim
KJ1646@gmail.com
AI Nanodegree - Udacity

Summary of the Paper

"Mastering the Game of Go with Deep Neural Networks and Tree Search"

In this summary, we discuss the paper “Mastering the Game of Go with Deep Neural Networks and Tree Search” that was published in an issue of Nature. The paper describes a game playing agent named AlphaGo that is able to achieve records feats performance in the game of Go. In concise fashion, we provide a brief primer on the game of interest, a description of the AlphaGo system and the goals of the system and the techniques used to achieve said goals. We also discuss the key results of the project.

The game of Go is what is known as a zero-sum, perfect-information and deterministic game. The rules of Go are very simple and few, there only being two. Despite this, the very abstract nature of the game makes it very complex with a search space orders of magnitude larger than other comparable games, such as chess, checkers, etc. Due to this there has also been an allure to master the game, both by direct practitioners and AI researchers alike.

Originally, the best agents to ever tackle the game of Go were all are based on the Monte Carlo tree search (MCTS) heuristic search algorithm, which relied on two key components to be feasible in practice, a policy distribution to sample actions from given a state and a value function that determines the value of a given state. However, the performance of Go game playing agents have always lagged behind their human counterparts. The main shortcutting has been due to the limitations of policy and value functions used in these systems.

The goal of the AlphaGo project is to take recent advancements in deep learning and combine it with MCTS to close that gap. Specifically, the method used by AlphaGo involves using the convolutional neural networks to replace the policy distribution and value function heuristics with policy and value networks. This allowed for much more accurate policy distribution and value function than was ever achievable using conventional handcrafted heuristics. This in turn allowed for high accuracy selection of actions during rollout and high accuracy of evaluations of board state in real play.

The steps used to train AlphaGo was done in three stages. The first stage involved training a policy network using supervised learning, with the labeled data coming from 30 million positions from the KGS Go Server. In this stage, a less accurate but faster policy network was trained using a smaller feature set. The main purpose of this stage is to train the policy network to correctly predict states of the board. The second stage involved training another policy network using reinforcement learning, with this network initialized to the weights of first policy network. Playing games against previous iterations of itself, the AlphaGo agent will optimize the policy network to actually win games. The third and final stage was focused on training the value network that assignments a relevant to a given board state. The value network initially leverages the weights learned by the policy network during reinforcement training and further optimizes by learning from a board state-game outcome pair training set.

Using these deep learning techniques and combining it with MTCS, the AlphaGo achieved objectively impressive results, beating out all other state-of-the-arts Go playing agents easily, going as far as to give the other agents a handicap and still win. However, most notability, the AlphaGo agent was able to best professional human practitioners, including those with the highest professional rankings of 9 dan, in a game of Go without any handicap, a feat previously thought to be unobtainable for at least a decade.