Research Review of AlphaGo

by Avery Heizler

Brief Summary:

Mastering the game of Go with deep neural networks and tree search, published in Nature, delves into the novel approach that Google's DeepMind team used to create the computer program **AlphaGo** that uses **value networks** to evaluate positions on the game board and **policy networks** to select moves for the game Go. The team also introduced a new search algorithm that combines **Monte Carlo Tree Search** with value and policy networks.

Approach:

A game of Go uses a 19x19 board that has a branching factor of ~250 with the typical number of moves being ~150. AlphaGo uses **neural networks** and **Monte Carlo Tree Search** in order to train and conquer the game of Go. Three policy neural netoworks are used for deciding which moves to evaluate and play. They were trained to identify possible beneficial moves from a 19x19 image of the Go gameboard. A fourth neural network, called a **value network**, looks at each of the images and assigns a value to the current player position. The Monte Carlo Tree Search uses the four networks to evaluate the value of each game position and to identify the most promising move at the moment.

The **Supervised Learning policy network** is a 13-layer deep convolutional neural network trained on 30 million Go game positions. With a given position, this network predicts the next most likely move. It alternates between convolutional layers and rectifier nonlinearities. A final soft-max layer outputs a probability distribution over all legal moves.

The next stage uses a **Reinforcement Learning policy network** to predict the next best move, rather than the most likely move. This network has the same architecture as the SL network as it started out as the same network. This RL network became stronger by playing against itself 1.2 million times and defeated earlier iterations of itself, keeping the network weights of the winner, thus becoming stronger.

The third policy network is called the **Fast Rollout policy network**. Similar to the SL network, it was trained to predict the next move, but was created to select an action in just 2μs, rather than 3ms for the policy network. While it only achieved an accuracy of only 24.2%, it is used to play out the rest of the game, predicting the most likely outcome.

The **Value network** estimates the probability that, given the current position, would lead to a win or loss for the current player, originally trained on the same data of the SL policy network. To avoid overfitting, and improve its ability to generalize, the team trained it on games collected during the reinforcement-learning phase (~ 30 million distinct positions).

The paper presents three sets of results for the two different implementations of AlphaGo (distributed and non-distributed):
• Both versions significantly outperformed other Go-playing AIs and defeated the best European player.
• Simply using the value network, without all the neural networks, AlphaGo outperforms other AIs.
• By utilizing more hardware (using 40 search threads on 1,202 CPUs and 176 GPUs), AlphaGo performs at its best.