Research Review

Mastering the game of Go with deep neural networks and tree search
By: Eric Iacutone

Goals

The goal of AlphaGo is to develop a technique in order to solve problems with large search trees and board position and moves. The game of Go uses a 19 by 19 board. Furthermore, Go has 361 board positions, a state-space complexity of 170 (log to base 10), a game-tree complexity of of 360 (log to base 10) and a branching factor of 250. These attributes lead to a high branching factor and large search space which contribute to a game with many board positions and moves.

Techniques

AlphaGo uses multi-layer neural network with a value network in order to evaluate board positions and a policy network to sample potential actions. AlphaGo uses Monte Carlo tree searching (MCTS) in order select good moves. MCTS is a search algorithm used to make decisions. AlphaGo searches the game tree for good moves based on the above methods. The neural networks used are both a supervised learning policy and a reinforcement learning policy.

The first stage of training pipeline is a supervised learning (SL) of policy networks trained on randomly sampled state-action pairs. The supervised learning network goes through all of the legal moves for a given state and selects the best move. The training process pulled over 30 million positions from the KGS Go Server.

The second stage of the training pipeline is called the reinforcement learning (RL) of policy networks. The goal of this layer is to improve the decisions made by the SL layer. The RL network played the SL network and adds weight to paths with maximum outcomes.

The final stage of the training pipeline is reinforcement learning of value networks. This stage is similar to the RL policy, but only returns a single prediction.

The above policy and value networks are then searched in a MCTS algorithm. The actions of the MCTS algorithm are selected by lookahead search. The algorithm chooses the most visited node from the root position based on the lookahead search.

Results

Between October 5-9 AlphaGo and Fan Hui played a five game Go match. AlhpaGo won five games to zero. This is the first time a computer program has defeated a human professional player, without handicap, in a full game of Go. This feat was previously believed to be a least a decade away due to the large search space of Go. AlphaGo uses a novel combination of MCTS

and deep neural networks. The policy and value networks are executed in parallel. MCTS uses an asynchronous multi-threaded search. In the end, AlphaGo uses 48 CPU's and 8 GPU's. A distributed version of AlphaGo was also developed. An interesting fact is AlphaGo evaluated thousands of fewer positions playing Hui than Deep Blue evaluated playing Kasparaov in chess.