Technical Summary

Mastering the game of Go with deep neural networks and tree search

The main goal for the paper was to introduced a game playing agent for Go that is able to play at the level of the strongest human players. The main contributions of the paper expands on the Monte-Carlo Tree Search (MCTS) methodology by supplementing the agent with effective move selection and positional evaluation functions through the use of deep neural networks trained through supervised learning and reinforcement learning.

AlphaGo provides a new level of game-playing for Go through two main novel approches; the use of a policy network for move intuition and value networks for board evaluation. Essentially, policy networks provide AlphaGo with a probability distribution of possible legal moves which favor a game win while value networks replace the traditional heuristic evaluation function.

The policy networks for AlphaGo are constructed in two different ways. Firstly, a 13-layer policy network, known as the SL policy network, is constructed through supervised learning from 30 million positions from the KGS Go server. These are known moves made by human players from all over the world. This is done by representing the Go board as an image and using a deep neural network to train the model. Such a large network achieved an accuracy over 50% but was slow at 3ms per move. A secondary network, which achieved an accuracy of 24.2%, was trained and managed to achieve a speed of 2μs. Secondly, AlphaGo uses a reinforcement learning approach to train a second policy network known as the RL policy network. Games were played between current policy networks and randomly selected previous iterations. This achieved a win-rate of more tha 80% against the SL policy network and beat Pachi, an open-sourced Monte Carlo search Go program, 85% of the time.

The value networks for AlphaGo are different from the traditional approach of game-playing AI. Instead of handcrafted heuristic evaluation functions that mimic human game-playing strategies, AlphaGo learns via reinforcement learning to evaluate the game state. Essentially, the value network is similar in structure to the policy network, with the difference being that instead of a probability distribution output over all legal moves, a single overall value score is returned. The network is trained via stochastic gradient descent, minimizing the mean square error between the predicted value and the final outcome of the time.

Finally, AlphaGo combines the policy network and value network using the MCTS algorithm. The tree is traversed via simulation starting from the root node and nodes that are frequently visited are penalized, encouraging exploration of new moves. Leaf nodes are evaluated using both the value network as well as a policy rollout which simulates the game till termination from that current state. At the end of the simulation, all action values and visit counts are updated and the nodes are scored accordingly.