Research Review for AIND

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

Summary of Paper's Goals and Techniques

Goal

The Uber goal of the AlphaGo project was build an AI agent that can consistently defeat the professional Go player in the world.

Problem

When building an AI game playing agent for Go the first thing is the really large nature of the search tree (b≈250, d≈150).
Defining a positional evaluation function for a board state, is also very difficult because of the game complexity.
Prior agents resorted to techniques like MCTS but achieved only weak amateur rating.

Techniques

Evaluation Function

The team at AlphaGo decided to mix rollout strategy like MCTS along with multiple neural networks in an approach to solve the problem of defining an evaluation function.

Instead of mathematically modelling an evaluation function they used NN to define it based on weights and biases.
The evaluation function (A set of Models) was generated with the following pipeline -

1- Supervised Learning Components – The program was fed with game play from actual masters and it generated
   a. Rollout Policy – It can rapidly evaluate actions during rollout based on rollout strategy the network has learned from the game data. (similar to MCTS)
   b. SL Policy network – A model of CNN generated on game play data.

2- Reinforced Learning Components – The RL components seek to improve the policy network.
   a. RL Policy Network- The RL Policy network is similar to SL Policy network and the model is generated based on SL Policy network. And it is trained again and again by playing against itself and reinforcing the actions that end in win.
   b. Value Network – This network allows to predict outcomes of the game based on action from both players. It takes in complete games of master and even when the RL policy network plays itself.

AlphaGo selects the next action based on look ahead search.
It chooses the best action determining the outcome of the game based on 2 methods

1- Rollout Policy – Fast but has less accuracy.

2- Value Network - slower but has greater accuracy.

Depending on the time at hand to evaluate a board state the best move is chosen.

Apart from master game play data AlphaGo was trained and evaluated against Pachi, Feugo, Fan Hui, GnuGo and itself.

Even after solving for the evaluation function there was need for parallel computing. The AlphaGo used about 40 search threads, 48 CPUs and 8 GPU, to search through the search tree.

Summary of Paper’s Results

The AlphaGo has had tremendous success as a game playing agent. It has consistently defeated other Go playing Agents like Pachi and even the Fan hui 5-0, the first time a AI agent has defeated a human player without handicap.

The other result worth mentioning are when we have an evaluation function which is mix of rollout policy and value network performed the best wining > 95% of the games.