**Build a Game-Playing Agent**

**Project – Part 2: Research Review**

Student: Bruno G. Eilliar

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

1) A brief summary of the paper's goals or techniques introduced.

This paper introduces a new approach to build a agent to play the Go game using **value networks** to evaluate board positions and **policy networks** to select the next move (both are deep neural networks). The key strategy to build the agent is composed of a combination of supervised learning and reinforcement learning to train those deep neural networks.

Roughly speaking the method uses a convolutional layer to construct a representation of the positions on the 19x19 board and by evaluating positions using a **value neural network** and sampling actions using a **policy neural network** they argue that it’s possible to reduce the effective depth and breadth of the game search tree.

It’s used several stages of machine learning to train those neural networks, starting with a **supervised learning training (SL)** for the **policy network** with training data from expert human moves database. In the next stepa a **reinforcement learning (RL) policy network** is trained to improve the previous policy network (SL) by optimizing the final outcome of games of selfplay (this strategy guarantee to optimize the policy network towards the goal of winning games). The final step it’s to train a **value neural network** whose purpose is to predict the winner of games played by the RL policy neural network against itself. Finally the new Go player agent (AlphaGo) combines the traditional Monte Carlo Tree Search (MCTS) with these value and policy neural networks.

The combination of MCTS with deep neural networks requires considerable processing power, so the agent uses an asynchronous multi-threaded (40) search executing simulations on 48 CPUs and computes policy and value networks in parallel on 8 GPUs. The team also built a distribuited version of the agent with 1202 CPUs and 176 GPUs that excels the previous one in performance and seems to be pratically unbeatable on their results.

2) A brief summary of the paper's results.

Combination of Monte Carlo Tree Search (MCTS) with the proposed value and policy networks yield a 99.8% winning rate agains other know Go programs, also defeating a human professional player by 5 games to 0.

The single-machine version of the agent won 494 of 495 games against other Go programs (that includes CrazyStone, Zen, Pachi and Fuego). The team also played games with 4 handicap stones (basically they gave some advantage to the opponent), in this scenario the AlphaGo agent 77%, 86% and 99% of the games against CrazyStone, Zen and Pachi, respectively. The distributed version won 77% of the games agains single-machine version of their agent, and won 100% (unbeatable) of the games against all other Go programs (agents).

The team tested the hyperparamenters that control the combination of MCTS with the purposed value neural network and concluded that a mixed evaluation (half MCTS and half value netowrk) performed best, winning 95% of games against other agents with other hyperparamenters (MCTS and the value neural network seems to be complementary algorithms to evaluate the positions in the game).

Finally the distribuited version of their agent won 5 of 5 games against Fan Hui, a professional Go player, making history as the first computer Go player to defeat a human professional player in a full game of Go.