Research Review – AlphaGo

Probably the most interesting aspect to of the work presented in the paper is the plethora of technologies and techniques used to achieve the unprecedented level of success at the game Go. The nascent of this combination of technologies was based on the challenges seen in others' research and work in the field of AI-based game playing and in particular that around Go; basically, the team was trying to avoid others' mistakes. Specifically, they wanted to avoid mistakes such as:

- Attempting to use exhaustive search which is infeasible for the game of Go which has b^d (250^150) possible sequences of moves
- Overfitting when attempting to predict outcomes from data consisting of complete games – games outcomes were being "memorized" instead of moves being generalized

To avoid these issues and to achieve the desired level of success at the game the team leveraged this combination of technologies (with a brief description of how/why it was used) :

- Convolutional neural network – pass in 19 x 19 images of the board to reduce the depth and breadth of the search tree for moves
- Supervised learning – using information based on expert human moves in the game
- Reinforcement learning – using self-play by the agent to optimize the supervised learning network to focus more on "winning games" instead of "maximizing predictive accuracy"; this also reduced computations by 15,000 times over using Monte Carlo rollouts
- Monte Carlo tree search – used to combine policy and value networks

Additionally, the team also generated a "self-play" data set of 30 million distinct positions from separate games which were played by the reinforcement learning policy network.

Also noteworthy was that, despite all of the technologies, the team found that the policy network based on humans (supervised learning) performed better than that of the reinforcement learning policy network: "*It is worth noting that the SL policy network performed better in AlphaGo than the stronger RL policy network, presumably because humans select a diverse beam of promising moves, whereas RL optimizes for the single best move*."

With all of the technologies at play, the "computational cost" was increased; a trade-off that had to be made in terms of computational intensity as the networks that were implemented required more computations than other heuristic approaches: "*AlphaGo uses an asynchronous multi-threaded search that executes simulations on CPUs, and computes policy and value networks in parallel on GPUs. The final version of AlphaGo used 40 search threads, 48 CPUs, and 8 GPUs*."

It should be noted that during this reading I encountered a number of phrases/concepts that I was not familiar with; in particular: policy network and value network. From additional research, in the context of reinforcement learning, a policy/policy network is focused on learning which rewards are received for each action and ensuring that the optimal action is chosen. This is often achieved through adjusting weights via gradient descent using feedback received from the environment. Value networks/functions enable the game agent to learn to predict how good a state or action will be for the agent to be in.