Research Analysis – AlphaGo Paper by Google's DeepMind Team

Google's DeepMind team set out to create a game agent capable of competing against professional Go players and winning. The challenge inherent in this problem is the intractability of searching deep and wide enough for a given board configuration to choose an optimal move. Go is played on a board of 19x19 available positions. This makes the breadth, d, of the problem approximately 250 possible moves on average with a maximum of 361. In addition to this, the depth of a typical game is approximately 150 moves to reach a goal state. This leads to a tree with 250^150 possible states.

The DeepMind team attempted to solve this by splitting the problem up into three distinct pieces. First, they used a database of expert Go player games to build a "policy network" using supervised learning. They looked at board configurations and next moves to form a probability distribution so that the game agent could quickly decide what moves to prioritize when searching. This cut the breadth requirement significantly because the agent could focus on going deep with high probability moves.

The second piece was taking the policy network from the first step and using reinforced learning to play games against the RL policy and the SL model. This was to optimize the weights for each move so that the new RL policy network could further optimize the correct moves being made towards a goal of winning games vs selecting the most common move. Doing so resulted in high win percentages against not only the SL policy network but also other AI agents for Go including Pachi and Fuego.

The third step and last step is to train a value network, ie a network that can help reduce the depth requirement of the agent, by playing it against itself to determine the result of that move for that specific state if both players were to play the rest of the game optimally. This is very effective when you must limit the depth of search in a real-world scenario because of the time required to compute the larger depths. It can cutoff the search with a reasonable estimation of the result based on the current state and the next move. The DeepMind team used a Monte Carlo tree search for this piece.

Combining all three of these techniques, the DeepMind team could compete against the other Go programs currently out and win at a rate of 99.8%. In addition, it played against the European Go champion and could win 5 out of 5 games. This was the first time a Go agent could play against a human and win in a full-sized board. Prior to AlphaGo, this was thought to be at least a decade away from being solved.