Summary of “Mastering the game of Go with deep neural networks and tree search” by DeepMind

Jumping on the wagon of deep-learning-driven artificial intelligence, AlphaGo uses supervised and reinforcement networks as heuristics in a Monte Carlo tree search (MCTS). Sheerly without lookahead, AlphaGo performs “at the level of state-of-the-art Monte Carlo tree search programs…” which find the best move from a large number of simulations. Combining Monte Carlo tree search with value and policy networks led to AlphaGo’s landmark defeat of Fan Hui, the European Go champion.

AlphaGo’s heuristics were motivated by the need to have an agent that both chooses actions like a professional player and actively seeks to win. If the agent was only trained to behave like a professional player, it wouldn’t have a model for game states that are uncommon amongst professional players. And if the agent only seeks to win, moves won’t be evaluated efficiently. Therefore, search is optimized in two stages, with a value network and a policy network.

The policy network is a deep neural network that predicts moves that are likely to win. A supervised learning policy network was trained on 30 million expert moves until it could predict moves with 57% accuracy. Of course, emulating player behavior isn’t good enough. In order for the policy network to beat expert players, a reinforcement learning policy network is subsequently trained to improve the SL policy network by playing it against itself and rewarding optimal behavior. For each iteration, the current policy is played against a randomly chosen previous iteration. The SL network creates a framework for choosing human-like moves, while RL encourages winning. The policy network in turn can make winning moves with human efficiency.

The value network is an additional deep neural network trained to predict the winner of games the RL policy plays against itself. While the policy network narrows search breadth by only considering winning moves, this network reduces plies of search (depth) by estimating the winner in every position to effectively avoid searching to the end of the game.

By using deep learning as a search heuristic, AlphaGo’s raw neural network can beat MCTS-based Go programs. Therefore, the combination of policy and value networks with MCTS elevated AlphaGo to human-level performance. Of course, evaluating policy and value networks comes at a computational cost, meaning deep learning is several orders of magnitude more costly than conventional heuristics. The coincidental advent of GPUs specifically for matrix-intensive neural networks enabled parallel value and policy network computation to be done in acceptable time. A reasonable 8 GPUs are used in the final, asynchronous version of AlphaGo.

Go has been a benchmark of game play reserved for humans. A computer-based artificial agent surpassing human performance encourages this technology to be used for important real-world problems of optimization.