Udacity AIND
Titus Lungu - 9/19/17

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

DeepMind combined deep learning and Monte Carlo tree search in order to more intelligently search the relevant portions of the intractable search space in the game of Go. The deep learning pipeline consisted of two major parts: policy networks and value networks. Policy networks determine the type of play that should be performed, while value networks determine the value of that action. Monte Carlo simulations are then run in order to gain a better understanding of which policies return the highest values. Traditionally, policy and value functions have been relatively shallow without a good way of really fine tuning them. Deep learning allows for this to be overcome by giving the ability to expand the knowledge base in a more intelligent manner.

Two types of networks were experimented with for the policy networks: supervised learning and reinforcement learning networks. The supervised learning nets used convolutional layers to learn the best plays by training on 30 million positions from the KGS Go Server, achieving a test accuracy of 57.0%, compared to the previous state-of-the-art of 44.4%. The reinforcement learning nets played games between the current net and a random previous net, continuously learning from its mistakes and improving its performance, eventually being able to win 80% of games against the supervised learning model. These results were without yet incorporating search into the algorithms. The value network was trained similarly, using reinforcement learning and the reinforcement network as the policy net during play. In order to mitigate the high level of overfitting that was being experienced, the network played against itself to endgame, and achieved an MSE 0.226 and 0.234 on training and testing, respectively.

Monte Carlo tree search was then used to find the highest value plays. Monte Carlo tree search randomly traverses the search tree tens- or hundreds-of-thousands of times, backpropagating the results to the root node, in order to find the best move. After each game, tree nodes are weighted based on the outcome of the game in order to encourage choosing more promising nodes while still allow for tree exploration - conceptually similar to simulated annealing. Since evaluating policy and value networks takes several orders of magnitude more computation than traditional search heuristics, DeepMind used an asynchronous multi-threaded search, performing simulations on CPUs and network computations for policies and values on GPUs.

Running tournaments against all the other top performing Go AIs, AlphaGo achieved a 99.8% victory rate on a single machine, and 100% using their distributed machine implementation (which won 77% of games against the single-machine version). In October 2015, AlphaGo also beat the European Go Champion in 5 of 5 games, a feat previously thought to be at least a decade away. AlphaGo evaluated thousands of times fewer positions in these matches than DeepBlue did in it's historical chess match against Kasparov. The positions to evaluate were chosen more intelligently thanks to the policy networks, and they were evaluated more precisely using the value networks - likely more similar to how humans think.