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

Summary by Egor Kraev

Paper's objective: to create an algorithm good at playing the game of Go. This is a worthwhile task as one can see it as a representative of a class of problems with huge search spaces and difficult to define evaluation functions, that were intractable before.

Methods: The paper builds on what it describes as the state of the art before it, namely Monte Carlo tree search. MCTS takes a policy for sampling possible actions, uses that to do repeated rollouts to maximum depth, and averages over their results for an approximation of the value of a move; it also updates the policy over time to select children with higher values, thus focussing the sampling on better prospective branches. However, to date such methods had used shallow policies and value functions that were simply linear in the features.

The AlphaGo approach improves upon that by training a series of neural networks for both policy specification and node value estimation. Specifically, they first train a supervised learning (SL) policy network, on a set of past human moves. They also train a simpler, fast version of that to sample actions during rollouts.
They then refine that by taking the structure and weights of the SL network, iteratively letting it play against random earlier versions of itself, and tuning the weights using reinforcement learning (RL network). To prevent overfitting, just one position is taken from each game played, and combined with the game outcome to get a training data point.
Finally, they train a value network that predicts the outcomes of RL playing against itself, to value the individual game states.

For the final algorithm, they use a weighted average of the value network output, and of rollouts using the regular and fast SL networks (not the RL network, purely because the latter performed worse in this context). For running this, they looked at two hardware configurations, a single machine with 48 CPUs and 8 GPUs, and a distributed version with 1202 CPUs and 176 GPUs.

Results: Both the RL network and the value network, each taken on its own, exceeded the performance of the strongest Go programs before it; the single-machine mixed value/rollout version was stronger still, winning 99.8% of the games against other programs (on a sample of 495 games); and the distributed version won 77% of the games against the single-machine version, and 100% against other programs.
Finally, the distributed version won 5 games to 0 against Fan Hui, the winner of 2013, 2014, and 2015 European Go championships.