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

Context

Go is a very difficult AI problem because it branches widely, and games last for a very long time. The paper estimated a branching factor of 250 and game length of 150 moves. That's enormous.

Prior techniques have included using a policy to select next moves, rather than selecting all of them. As I could tell, the policy is a function that selects a subset of the possible next moves. Using this policy limits the branching factor, making search more feasible.

The achieve greater search depth, evaluation of a possible move is done by using the policy to probabilistically pick next moves beginning with a candidate position until the game is finished, then choose the move that resulted in the best total number of positive outcomes. This is a monte carlo simulation using the policy.

Techniques

AlphaGo's primary contribution is using a convolutional neural network (aka deep learning) to simulate the policy and value functions. The board was read in as an image and trained using history from prior expert games. Then to extend it's learning they had the system play itself (reinforcement learning) to educate itself on which moves resulted in wins. They did this both for the policy function, as well as the valuation function. To prevent over-fitting for the policy function (that is that the system didn't learn only how to play and counterplay one corner tactic) they had each iteration play a random prior policy. To prevent over-fitting for the value function they generated new games using the policy function and rolled out to the end of the game to determine the supervised value.

They then combine all of these techniques together to do some search, and some policy roll-out. Their value function of leaf nodes used a weighted combination of the valuation function learned above, and monte carlo simulation using a faster policy function than the CNN technique (fewer layers) for roll out. Then then threw a TON of compute power at the problem.

Results

It crushed other computer AIs. The single computer system beat other agents in 99.9% of games. To better evaluate things they then gave the other systems a 4 move handicapped, in which case the single machine still performed between 80% to 99% wins. The multi-computer system won 100% of its games against other systems. even when the other systems were given a handicap. It beat the reigning European Go champion 5-0.