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

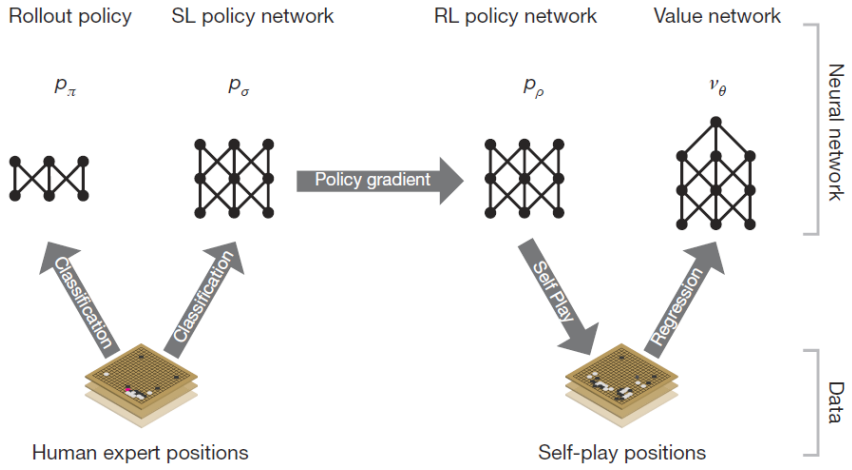
(AlphaGo article)

Goals

Perfect information games such as Chess and Checkers always have a perfect optimal value function, but depending on the game rules, the search space to find them can be intractable. While strategies to restrict search space have worked well in the past for many of those games, in Go the search space is so big that they could at best achieve amateur level. The goal of the research is to use deep convolutional neural networks to reduce the depth and breadth of the search space.

Techniques

The training pipeline is complex and composed of several layers, summarized in this picture:



1. Supervised Learning of a policy network, based on expert knowledge from human games. A neural network of 13 layers is trained against a database of 30 million moves.
2. Then they improve the policy network with Reinforcement Learning. By playing games between the current policy network and a random previous version of it, the algorithm train weights that are updated by stochastic gradient ascent.
3. The final stage focusses on position evaluation by estimating a value function that is trained on a database of 30 million distinct positions, generated by the previous RL policy network by playing against itself over and over again.

Evaluating policy and value networks like that requires a lot more computational power than the search heuristics we saw so far in the course. Thus, AlphaGo used 48 CPUs to compute playing simulation and 8 GPUs for training of these networks.

Results

AlphaGo has beaten other previous Go programs/AIs in 99.8% of all games, and was the first time a Go AI has beaten a human champion, a feat believed to still be at least a decade away.