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

This paper presents a new approach to the game GO, which is based on :

a. Deep Neural Networks trained by supervised learning (human experts games) and reinforcement learning (from games of self-learning). The training pipeline is as follows:

1. pσ: supervised learning (SL) policy network from expert human moves.
2. pπ: fast policy that samples actions during rollouts
3. pρ: reinforcement learning (RL) policy network that improves the pσ policy network by optimizing the final outcome towards the correct goal of winning games (but it does not maximize predictive accuracy)
4. vθ: value network that predicts the winner of games played by the RL policy network against itself.

b. Tree search: This search algorithms combines Neural Networks evaluation with Monte Carlo Rollouts that simulates random games of self-play.

The previous programs to the solution presented in the paper (and the ones that were more powerful) are based on MCTS (Monte Carlo Tree search). This algorithm uses Monte Carlo Rollouts for getting the state in a search tree. On the other hand the solution presented in the paper uses Monte Carlo with value and policy networks (trained with supervised and reinforcement learning) in the three search by selecting actions by lookahead search . This new approach is based on "value Networks" for evaluate board positions and "policy networks" for select moves or actions. The policy used to select actions during search is improved over time, by selecting children with higher values. This policy converges to optimal play. The program presented in the paper combines asynchronous multi-threaded search on CPUs, and computes policy and value networks in parallel on GPUs for an efficient implementation of the combination of MCTS with deep neural networks

The board position (as an image of 19x19) is passed through a convolutional network in order to get a representation of the position which is used to reduce the depth and breadth of the search trees.

the results are very good since this program won 494 out of 495 games (99.8%) against other Go programs. The distributed version of AlphaGo is stronger, since it won 77% of games against single-machine AlphaGo and 100% of its games against other Go programs

Also the network (supervised learning policies in the first stage of the pipeline) was able of predict expert moves with an accuracy of 57 %, compared to 44% of other programs. Also, the second stage of the pipeline (reinforcements learning policy networks) won more that 80 % agains the SL policy network (first stage in the pipeline)

Another very impressive result of this program was that it won 5-0 against a professional human player