Research review

Paper: **Mastering the game of Go with deep neural networks and tree search**
(https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf)

The paper introduces a novel approach for Go gaming by combining MonteCarlo Tree Search (MCTS) together with Deep Learning. Strongest Go programs prior AlphaGo were based only on MCTS and therefore have been outperformed in 494 out of 495 games by AlphaGo. Moreover, AlphaGo could win games with handicap opponent stones with minimum winrate of 77%.

AlphaGo uses pipeline of policy functions `P` and value function `V`. Policy functions are used to predict next move based on current board state. Value function is used as heuristic and gives a score of current board state.

The pipeline is described as following:

- Supervised Deep Learning classifier `Psigma` based on Convlutional Neural Network with ReLU activations and Softmax probabilities is trained for move likelihood prediction.
- Another shallow (and fast) classifier `Ppi` trained using just softmax over small pattern features. This one is used for rapid action sampling during MonteCarlo rollouts.
- Reinforcement learning (RL) network `Pro` with same structure and initial weights as `Psigma` is then trained by playing against previous iterations of itself.
- Train RL-based value network `V` is then trained based on `Pro` policy.
- Combine policy and value network with MCTS -- this is AlphaGo itself.

Most notable result is, of course, outstanding winning rate in comparison with other programs and, especially, first time that a Go program has defeated a professinal human player .

But in addition, several intermediate results are also very interesting and promising:

- Small improvements of supervised deep learning training of `Psigma` can significantly improve strenght of further Go-playing program.
- Only `Pro` policy network can defean many other programs and is ranked at 2 amateur dan.
- State evaluation function (value network `V`) approaches the accuracy of MonteCarlo rollouts with **15000 less** computations.
- Having that Go game has significantly more possible states because of its breadth and branching factors, AlphaGo evaluated thousands of times fewer position than Deep Blue did in its famous chess match.

Deep Neural Network allow to create much more intelligent AI software without handcrafting

of heuristics. The cost of this is computational resourses needed for runtime execution and, most important, for training. AlphaGo used an asynchronous multi-thread search that used 48 CPUs and 8 GPUs. Distributed version of AlphaGo exploited multiple machines with total 1202 CPUs and 176 GPUs.