AIND Project #2
- Isolation Game -
Deepmind's AlphaGo
research paper review

Goals and techniques introduced

The paper « Mastering the game of Go with deep neural networks and tree search » was published by Deepmind in January 2016.

The game of Go is highly challenging from an AI point of view, as the search space is gigantic (b^d with b ~250 and d~150) , making it impossible to search exhaustively. In such cases, the common approach is to reduce the depth search using an approximation function, and to reduce the breadth by sampling action from a policy. Until then, the best Go game implementations were heavily relying on Monte Carlo Tree Search (MCTS) algorithm (basically a type of Tree Search algorithm that uses policy functions to select and expands nodes).

In this paper Deepmind introduced a new approach combining Deep Learning and Tree search technics, more specifically:

- Value networks to evaluate board positions
- Policy networks to select moves
- MCTS algorithm

More in details, the Deep learning processing pipeline is composed of 4 parts:

- First, a « Fast Rollout Policy » and a « Supervised Learning Policy » (SL Policy) networks which are trained to predict human experts moves from a dataset of 30 million of positions.
- Then a « Reinforcement Learning Policy » (RL Policy) network is initialised from the SL Policy and is improved by self-playing against previous version of itself with the goal of wining games.
- Aside from improving the RL policy, this self-play games also generate lots of data, which are then used to train a « Value Network » which goal is to predict the outcome of a game from a position.

The policy networks are composed of « Convolutional Neural Networks » (CNN), taking at their inputs some representation of the board (at the input the network architecture is 19x19x48 : 19x19 corresponding to the board layout, and 48 planes to store the different features with one hot encoding format) and producing at their output a probability distribution of possible moves.

The value networks is also a CNN, but it outputs a single value to predict the expected outcome of the game position.

AlphaGo combines these policy and value network with a MCTS algorithm, in order to search efficiently through the tree of possible actions and select the best ones to perform.

From the architecture point of view, in order to combine efficiently the Deep Learning networks and the MCTS, Deepmind came with an asynchronous multi-threaded search approach for AlphaGo, where simulations are running on CPUs while the policy and value networks computations are executed in parallel on GPUs. (Deepmind came with 2 AlphaGo implementations: one version which uses 40 search threads, 48 CPUs and 8GPUS, while the distributed version which uses multiple computers, 40 search threads, 1202 CPUs and 176 GPUs)

Results

The Deepmind searchers have accessed the AlphaGo playing level by running tournaments between several versions of AlphaGo, and several other Go commercial and open source programs

With this new approach, AlphaGo has achieved a 99.9% winning rate against several Go programs, but was especially the first Go program to play at expert level, and to defeat a European Go champion (Fan Hui , 5-0). This is an important milestone for AI, as this novelty technic allows to « mimic » / capture some kind of intuition about the game, as a Human player has to do, considering the huge space of possible actions.