Article review

"Mastering the game of Go with deep neural networks and tree search", D. Silver et al., Nature 2016

Udacity Artificial Intelligence Nanodegree – Isolation game
Pierre Oberholzer – 15.11.2017

Goals and techniques

The article presents the so-called AlphaGo algorithm used in the computer program that defeated for the first time a human professional (by 5 games to 0), while beating other Go computer programs with a success rate of 99.8 %.

Compared to its predecessors based only on the Monte Carlo tree search (MCTS), the novelty of this algorithm is to integrate into the tree search the predictions of deep neural networks representing positions and evaluating at each position the *value function* (i.e. a function returning an outcome), and the *policy* (i.e. the probability distribution of each action) starting from this position. Hence, the depth and breadth can be reduced and a computer can tackle the problem, which would otherwise be impossible to solve with an exhaustive search (250^{150} possible sequences of moves).

Precisely, three types of policies are evaluated. First a supervised learning policy network is trained on a data set including 29.4 millions of positions from 160'000 games played by human players (3 weeks training), taking a board state as an input and returning a probability of moves as an output. Second and similarly to previous algorithms, a fast policy rollout is used for the same purpose, though with higher speed but lower accuracy, so as to get rapid action sampling in absence of a search tree at given state. Third, a reinforcement learning policy network is applied to improve the supervised learning policy network obtained before by making the supervised learning network play against itself for 1.28 millions games (one day training).
Then, another neural network is used to create a value network trained on the reinforcement learning policy network when playing against itself. This network was trained on 50 millions mini-batches of 32 positions (one week training).
Finally, the selection of the best action is obtained by using the policy and value networks in a MTCS that fully traverses the game tree from the root node for each simulation. During each simulation and at each position, the action selected corresponds to the one maximizing the action value plus a bonus that favors exploration. At the leaf node, a mixed parameter is calculated from the value network and the outcome of the fast rollout policy. Once the simulation is finished, all the action values and visit counts are updated, and the algorithm choses the final move as the one featuring the most visits from the root position.

The evaluation of policy and value networks is computationally much more demanding than the tradition search heuristic. The hardware architecture reported consists of a master machine executing the main search, while remote CPUs are used for asynchronous rollouts and remote GPUs for asynchronous policy and network evaluations.

Results

The AlphaGo algorithm defeated the European Go champion Fan Hui by 5 games to 0 during formal matches, and by 3 games to 2 during informal matches. These 10 matches were played during 5 days in October 2015. The AlphaGo algorithm also defeated other commercial and open source Go computer programs, all based on the MTCS algorithm, with an overall winning rate of 99.8%. A computation time of 5s was given per move for all programs.

According to the authors, this performance achieved by AlphaGo was thought to be achievable only in about one decade, which represents a breakthrough in the field of artificial intelligence. The authors also describe their technique as more intelligent than Deep Blue in the sense that a much smaller number of positions were evaluated thanks to a more precise evaluation of each position. Also, AlphaGo was trained on real games played by experts, whereas Deep Blue was making use of handcrafted features.