Research article review

Article details

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

Authors: David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, Demis Hassabis

Authors' affiliation: Google and Google DeepMind

Date of publication: 28 January 2016

Publisher: Nature

Goals

The high-level goal of the research effort is to produce AI capable of playing the game of Go (a game of perfect information) at the highest professional level (capable of beating a Go world champion). Because the search space for Go is huge, growing the full game tree during the game is not feasible. Therefore the operational goal is to find a way of **reducing the breadth and the depth of the search tree**.

New techniques introduced

The innovation consists in extending the use of deep convolutional neural networks (a type of deep neural networks (NN)) from visual applications to game-playing. Concretely, several such networks are used for two purposes:

- **Sampling actions** according to their probabilities, to **reduce the breadth** of the search. Three "policy networks" were trained. A policy p(a|s) is a probability distribution over possible moves a in position s. Monte Carlo rollouts is a technique for sampling sequences of actions to be searched to maximum depth. Monte Carlo tree search (MCTS), which consists in executing many Monte Carlo rollouts, was the technique used in state-of-the-art Go-playing programs at the time the paper was written.
- **Evaluating positions** (a "value network"), to **reduce the depth** of the search. A value function in a game of perfect information such as Go determines the outcome of the game from board state s, under perfect play by all players. Because searching to endgame is impossible in games with

a large search space, the optimal value function v*(s) is estimated by an approximate value function v(s).

The networks were trained as follows:

- **The policy networks:** as a first stage, by supervised learning (SL) on expert human moves; at a second stage, by reinforcement learning (RL).
- **The value network:** by RL, on a data set generated by self-play (games played by the RL policy network against itself); the of 30 million distinct positions were each sampled from a different game, rather than using complete games as training data, to avoid overfitting.

The resulting policy and value networks were then combined within an MCTS algorithm. The computational requirements for combining deep NNs with MCTS are huge. AlphaGo used, in its final version, 40 search threads, 48 CPUs, and 8 GPUs; and in its distributed version, 40 search threads, 1202 CPUs, and 176 GPUs.

Key results

At the **outcome of the training** process:

- The SL policy network achieved a test accuracy of 55-57% correct prediction of human moves (compared with the state of the art, of only 44%). The RL policy network had a winning rate of 80% when playing against the SL policy network from the first stage, and 85% against the currently strongest open-source software (without even searching the game tree).
- The value network managed to almost entirely avoid overfitting, and approached the accuracy of Monte Carlo tree search, but with 15,000 times less computation.

In an evaluation **tournament**, AlphaGo won most of the games it played against other Go programs: 99.8% of the regular games, and between 77% and 99% of the games it played with a handicap (because the opponent had been given free moves).

Variants of AlphaGo that evaluated positions using only the value network or only rollouts also performed well, although worse than the mixed evaluation.

In a **formal five-game match**, AlphaGo defeated the European Go champion Fan Hui 5 games to 0. This was the first time a program ever defeated a human professional Go player in a full game played without a handicap for the human player. Compared with IBM's chess-playing program Deep Blue, which also defeated a human champion, AlphaGo evaluated thousands of times fewer positions, by selecting more intelligently which positions to evaluate (using the policy network), and evaluated them more precisely (using the value network).

AlphaGo is a **breakthrough in artificial intelligence**, as it has succeeded in managing very efficiently and accurately an intractable search space through an innovative search algorithm that uses deep NNs trained using a novel combination of supervised and reinforcement learning, combined with Monte Carlo rollouts.