Research Paper Review

Ilya Nikokoshev

December 17, 2017

Abstract

As part of the Project 2 in the Artificial Intelligence Nanodegree Program, we review an AlphaGo paper [SHM+16] by the DeepMind team.

1 Overview

The article describes the Monte Carlo tree search algorithm used in previously existing programs to select the move in the game. The effectiveness of this algorithm depends on heuristics for position evaluation and best move prediction.

Authors implement those functions using convolutional neural networks. Crucially, those networks can be first trained using the expert knowledge, such as existing databases of human moves and hand-crafted evaluation functions, but then fine-tuned without human intervention.

2 Training pipeline

The *policy network* that assigns best move probabilities to moves in a position p_σ is implemented with 13 convolutional layers and trained on the expert moves.

The weights σ are then refined by self-play with the stochastic gradient ascent in the direction of weights that win more games. Using only p_ρ, the algorithm is already strong enough to win over strongest open-source Go programs.

For the tree search part of the algorithm, a separate policy network p_π, that is only considering small pattern features, is implemented. It is significantly simpler and about 1000 times faster, making its use more practical for rollouts. The value prediction function is implemented with a *value network* ν^{p_ρ} similar to p_ρ.

3 Monte Carlo tree search

For a given position, the search tree is built. A policy network is used to compute prior probabilities of selecting the moves. The leafs of the current search tree are then expanded in a simulation according to this probability (plus an exploration bonus). On a leaf, the value is computed using a mix of:

1. value predicted by a value network,

2. result of a rollout using p_π.

The authors have experimentally established that

- p_σ works better than a p_ρ as a policy network for a prior probability.
- ν^{p_ρ} works better than ν^{p_σ} as a value network.
- Averaging the value heuristics is a better strategy than using only one of them.

4 Competition results

The final single-machine version of AlphaGo (using 48 CPUs and 8 GPUs) won 494 out of 495 games against other Go programs.

The distributed version of AlphaGo (using 1,202 CPUs and 176 GPUs) won 5 to 0 in a formal match against a professional 2 dan player, placing it on the level of the strongest human players.

5 Further development

The work of the authors is continued in [SSS+17], where the technical improvements allow the AlphaGo Zero program to skip the step of using the human expert knowledge, and proceed to gain mastery in the game by tabula rasa reinforcement learning.

This work is further generalized to other knowledge domains in [SHS+17].

References

[SHM+16] 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, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016.

[SHS+17] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *ArXiv e-prints*, December 2017.

[SSS+17] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 10 2017.