Mastering the Game of Go with Deep Neural Networks and Tree Search

by Artem Odintsov

Humans have always been interested in games. We have created many amazing games such as Go, Chess, and checkers. But now we are not only interested in making games, but also in teaching computer to play those games. One of the most striking games that we have come up with is Go, which contains hundreds of thousands of possible moves. For a long time it was hard to imagine that somebody could develop an algorithm that will allow the computer to learn this game, to play it and to win. Recently Google announced it's child, AlphaGo, a super computer that can not only play Go, but defeat all existing Go programs and even the best human minds.

The article "Mastering the Game of Go with Deep Neural Networks and Tree Search" describes the basic principles and ideas that the team from Google used to develop and implement AlphaGo. Needless to say, Go is one of the most complex known games due to its enormous search space and complexity of evaluating board positions and moves. Maybe because of these characteristics, the Google team came up with a new approach by not only using the well-known Monte Carlo search algorithm, but by combining it with new, modern technology such as Neural Networks. They were used in two different parts; firstly "value networks" were used to evaluate board positions, and secondly, "policy networks" were used for selecting moves. It's worth mentioning that Neural Networks require training. To teach these networks, the team used human experts of Go, as well as getting NN's to play a lot with themselves heavily decreasing the winning chances, and decreasing the chances of making a mistake. Thus the combination of the Monte Carlo search algorithm, values NN's, and policy nets allowed this supercomputer to get a 99.8% winning streak against other Go programs, and allowed it to defeat the European Go champion 5 - 0.

Nowadays convolution NN are extremely popular and they are found applicable in different areas. Due to the fact that Go has an extremely huge branching factor, convolutional NN has significantly helped to reduce the depth and breadth of a search tree: evaluating a position using value NN, and sampling action using policy networks. In addition, the whole training of NN had three stages: 1) Supervised training for policy networks from human experts, which provided fast and efficient learning updates. 2) Reinforcement learning (RL) of policy NN, which has improved the final outcome of games of self-playing. 3) Training value NN that predicts the winner of games played by RL policy network.

Unquestionably, this new approach has shown a great result. The supervised learning improved results from the previous record of 44% to 57% on hold out test set, and to 55.7% on raw data. Reinforcement learning allowed AlphaGo to win 85% of the time against Rachi, one of the most complex programs of Go. And finally

training value NN allowed AlphaGo to reach MSE of .226 and .234 on training and testing set respectively.

In summary, the new approach used by Googled has shown us that something we believed would only be available in 10 years' time, is in fact available now.