Research Review

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

Written by **Federico Martini**

Goals and Techniques

Goals

The DeepMind Team at Google started this project aiming at developing an AI algorithm capable of reaching a professional level in a challenging task like any algorithm has done before. To pursue this goal, they tried to implement an Agent named AlphaGo to play a game at Go, the most challenging of classic games for Artificial Intelligence because of its enormous search space and the difficulty of evaluating board positions and moves.

Techniques

Go is a game in which the information is perfect like Chess, but the Deep Mind team went for Go because of its way more broader search space. All the games having perfect information may be solved in approximately b^d possible sequences moves, where b is the game's breadth and d is its depth. That being said, we can consider these values to figure out the difference in terms of search space between Chess and Go. In fact, in Chess $b \approx 35$ and $d \approx 80$, as in Go $b \approx 250$ and $d \approx 150$.

To deal with this challange, DeepMind figured out that the effective search space, could have been reduced by two principles: first position evaluation can reduces the depth, and second, the breadth of the search can be reduced by taking into account of the probability distribution over possible moves ***a*** in position ***s***. These two principles have been applied by taking advantage of new approaches that use *value networks* to evaluate positions and *policy networks* to select moves. These networks leverages of the cutting-edge Neural Network techniques introduced by the Deep Learning: a combination of a Supervised Learning from human expert games and Reinforcement Learning from game of self-play. Then, to further improve the win rates, the team introduced a new algorithm to combine *Monte Carlo Search Tree(MSCT)* simulation with value and policy networks.

Hardware

The hardware used to compute the results of the algorithm was configured as follows:

- 48 CPUs
- 8 GPUs

The DeepMind Team also implemented a distributed version of AlphaGo that exploited multiple machines with a complessive configuration of:

- 1022 CPUs
- 176 GPUs

Results

AlphaGo played against many of strongest Go programs available, like CrazyStone, Zen, Pachi and Fuego, all based on the high-performance MCTS algorithms. In each game, the programs were allowed 5s of computation time per move. At the end of the tournament AlphaGo reached an impressive score of 99,8% win rate (it won 494 out of 495 games). The games have been computed even leveraging of a distributed configuration, but it showed lower win rates respect to the configuration with a single machine. After that, AlphaGo played 5 games against a Fan Hui, the winner of 2013, 2014 and 2015 European Go Championship, and it won all the 5 games.
In this work, DeepMind achieved one of the Artificial Intelligence's "grand challenge", that was to play a Go game at the same level of a human player. This success provides hope that human-level performance can be achieved in other Artificial Intelligence domains.