Research Review

Author: Alexis Alulema

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

---

Introduction

Go has been for long time a big challenge for Artificial Intelligence, because of its complexity and the huge number of possible moves, rising the number of possible moves to 250^{150}, which makes it not feasible to solve by mapping the entire tree of moves.

The most amazing fact about this paper is the fantastic way that AlphaGo combinates several fields from Game Theory, Machine Learning and Computer Science. They have used the usual methods that we have been studying: minimax and alpha-beta pruning in order to find the optimal value function, but these methods have not been effective when applied on Go. Because of this, the authors have find a good performance when use Supervised Learning (SL) and Reinforcement Learning (RL) techniques to find the value function.

State-of-art

The elements saw in the implementation of AlphaGo have been the following:

1. Value networks :: to evaluate positions
2. Policy networks :: to select moves
3. Supervised Learning :: to learn from human expert games
4. Reinforcement Learning :: to learn from games of self-play
5. Monte Carlo Search Tree :: to estimate the value of each state in a search tree
6. Neural Networks :: to play Go on MC tree search simulating random self-play games

AlphaGo have used a 13-layer network to be trained using image representations of the board, with moves taken from the KGS Go server (30 million samples). The first neural network performs a classification which produces a policy gradient, and outputs a probability distribution over all legal moves.

Reinforcement Learning has been used here to use a probability distribution of moves coming from the previous NN, and apply with Monte-Carlo tree search (MCTS), which uses random sampling of the tree evaluating each game tree branch. MCTS determines the better moves. The value for each explored node is the mean of an evaluation function over the number of visits on that node.

Conclusions

In the past, I've used Convolutional Neural Networks (CNNs) to perform image classification, working in an excellent way. In this time, the authors have adapted the board to behave like an image and learn and construct representations of parts of the board. As I understand, the CNN will learn from expert moves and it will recognize learned moves (evaluate positions) in order to pick the best move (policy networks).

This idea, under my point of view, is the most fascinating part of this paper because it indicates me that we could adapt not only images on CNNs, but we can apply CNNs on other elements looking like a 2-D matrix, like game boards, maps, differential equations represented as matrices, etc. Over this idea, AlphaGo will apply different techniques to evaluate the value function and decide the better moves.

As conclusion, AlphaGo demonstrate that value functions not always have to be designed by “hand”, but, in the case of very complex scenarios, we can use powerful tools like Neural Networks to abstract this function and keep using known methods to adapt and solve the problem.