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

Adelene Sim

5th June 2017

Executive summary: Go is a game that has large branching factor and depth, making it an intractable game for artificial intelligence (AI) agents that implement traditional exhaustive search. AlphaGo uses Monte Carlo simulation, together with policy and value neural networks to achieve exemplary Go playing ability without any lookahead search.

Techniques: The AlphaGo training pipeline consists of the following steps:

1. **Supervised learning of policy networks**
   A 13-layer neural network was trained using the board state representation as the input (s), with the desired outcome to learn the policy of state-action (s, a) pairs. The goal of the network was to identify human expert moves. The neural network was trained using 30 million board positions in the KGS Go Server. The training was done through stochastic sampling of (s, a) pairs, together with stochastic gradient ascent to maximize the likelihood of a selecting a human expert move for each state s. The authors show that even a small increase in training accuracy would lead to substantial improvements in AlphaGo win rate.
2. **Reinforcement learning of policy networks**
   The policy network from (1) was then refined through self-play and policy gradient reinforcement learning (RL). The difference with (1) is that the objective of the policy is to identify (s, a) pairs that lead to winning games, rather than simply mimicking human expert moves. This policy network was able to achieve an 85% win rate against Pachi -- the strongest open-source Go program. For each state, a probability distribution of its action (i.e. policy) is determined.
3. **Reinforcement learning of value networks**
   Next, RL was used to train the value function for given board position s. In practice, this estimates the optimal value function for the best policy from (2). Unlike (2), however, the value network outputs a single value for each state s. The authors highlighted the danger of overfitting arising from highly correlated consecutive positions. They circumvented the problem by generating a new self-play training set comprising of 30 million initial distinct and randomly sampled positions. While the value network slightly underperforms the supervised learning and RL policy networks, it requires 15,000 times less computation, and hence is more efficient for game play.
4. **Lookahead search**
   The policy and value networks were combined in a Monte Carlo Tree Search algorithm for lookahead search to identify the best action for a given board state. Tree exploration takes place by simulation, with the action chosen to maximize the sum of the action value and a bonus. The bonus helps to encourage tree exploration. At the end of the search, the most visited move from the tree root is returned.

A distributed AlphaGo version was also implemented.

Results: AlphaGo performed exceedingly well compared to other AI agents (such as GnuGo, Fuego, Pachi, Zen and Crazy Stone). As reported in this paper, AlphaGo beat the human European Go champion Fan Hui 5-0. More recently (after this paper was published), AlphaGo proceeded to defeat the world Go champion Ke Jie by 3-0..