

Research Review of DeepMind's AlphaGo Paper

Why Go is a challenge?

The ancient game of Go is one of perfect information. As such it has an optimal value function $v^*(s)$ that can determine the outcome of the game for any state s . Furthermore, Go is really complex. The number of possible states is estimated to be 250^{150} which is roughly equal to 10^{761} . The game of chess has 'only' 35^{80} states or roughly 10^{120} . It was widely accepted that Go is intractable for machine players.

New techniques

The previously strongest machine players at Go were based on Monte Carlo Tree Search (MCTS) techniques, which basically try to reduce the complexity of the search tree via sampling/making simulations.

AlphaGo uses a somewhat novel approach to estimate the value of the current state. To do that, it uses a tree search procedure and convolutional neural networks to 'guide' the search.

Three different neural networks are trained, of two different kinds: two policy and one value network. Both types receive as input the current game state. The output of the value network is the probability of winning. The policy networks provide the probability of winning if each possible move is chosen.

Training

A policy neural network was trained on games played by human experts. Its accuracy on validation set was around 57%. Additionally, Deep Reinforcement Learning was used to improve the performance further through self-play.

Evaluation

Initially, AlphaGo was opposed to the strongest commercial and open source Go programs. All programs were allowed 5s of computational time. AlphaGo won 494 out of 495 (99.8%) of the played games. This result suggests that DeepMind's agent is many dan ranks above its opponents.

Most importantly, AlphaGo defeated the European Go champion (Fan Hui) 5-0. This is the first reported result of computer program winning over a professional human player in full-size Go game.