Research Review

Jinlin Song

Udacity AIND
Build a Game-Playing Agent

Summary

Owing to the strongest Go programs based on MCTS have been limited to shallow policies or value functions based on a linear combination of input features, “Mastering the game of Go with deep neural networks and tree search” introduced a new approach ‘value networks’ to evaluate board position, and introduced ‘policy networks’ to select moves.

We use these neural networks to reduce the effective depth and breadth of the search tree: evaluating positions using a value network, and sampling actions using a policy network. It train the neural networks using a pipeline consisting of several stages of machine learning, range from a supervised learning policy network P_σ, a reinforcement learning policy networks P_ρ to a value network v_θ. ‘Value networks’ and ’policy networks’ are both trained by a combination of supervised learning, reinforcement learning. We pass in the board positions as a 19 * 19 image and use convolutional layers to construct a representation of the position. We trained a 13-layer SL policy network, from 30 minnilion positions, and the network predicted experts moves on a held out test set with an accuracy of 57.0% using all input features, compared to other research groups of 44.4% at the date of submission. We improve the policy networks by policy gradient reinforcement learning. We use a reward function $r(s)$, and the weight of policy networks are updated at each time step t by stochastic gradient ascent in the direction that maximizes expected outcome. When played head-to-head, the RL policy network won more than 80% games against the SL network. Using no search at all, the RL policy network won 85% of games against Pachi. We put forward reinforcement learning of value networks. We trained the weights of the value network by regression on state-outcome paris (s, z), using stochastic gradient descent to minimize the mean squared error between the predicted value and the corresponding outcome. In order to solve overfitting, we generated a new self-play data set consisting 30 million distinct positions, each sampled from a separate game. Training on this data set led to minimal overfitting. Compared to Monte Carlo rollouts using the fast rollout policy, It is more accurate and efficient.

Without lookahead search, the neural networks play Go at MCTS. The paper introduced a new search algorithm combining Monte Carlo simulation with value and policy networks, at scale, in a high-performance tree search engine.

Result

Under algorithm combining Monte Carlo simulation with value and policy networks, our program AlphaGo achieved a 99.8% winning rate against other Go programs, and defeated the human European Go champion by 5 games to 0. This is the first time that a computer program has defeated a human professional player in the full-sized game of Go. AlphaGo has finally reached a professional level in Go, providing hope that human-level performance can now be achieved in other seemingly intractable artificial intelligence domains.