Mastering the Go Game with Deep Neural Networks and Tree Research

Objectives and techniques used

The game of Go has been seen as the most challenging of the classic games for artificial intelligence due to its huge search space and the difficulty of assessing positions and movements of the board. the purpose of this paper is to find a new approach when calculating Go using "value networks" to evaluate the positions of the board and the "policy networks" to select movements. These deep neural networks are trained a new technique that combines supervised learning, in which games of human experts were used to train the network and reinforcement learning from self-play games. We also present a new search algorithm that combines Monte Carlo simulation with networks of values and policies. Since the search space is very large, it is necessary to use some technique to reduce it using two general principles:
First, the depth of the search can be reduced by position evaluation: by reducing the search tree in the state by replacing the subtree below s by a function of approximate value $v(s) \approx v^*(s)$ that predicts the outcome of the game in state s. Secondly, the breadth of the research can be reduced by means of sampling actions through a policy $p(a \mid s)$ which is a probability distribution on the possible movements in the s position, so we can obtain an evaluation of the position obtaining superhuman performance
As more simulations are performed, the search tree grows and the relevant values become more accurate. The policy used to select actions during the search is also improved over time by selecting child nodes with higher values. Asymptotically, this policy converges to the ideal game and the ratings converge to the optimal value function.
Supervised learning of policy networks is used for the first stage of the training pipeline, a policy network SL was constructed $p\sigma(a \mid s)$ that alternates between convolutional layers with weights σ and nonlinearities of rectifiers. A final layer of softmax produces a probability distribution over all legal moves a. The s entry for the policy network is a simple representation of the state of the board. The policy network is randomly trained by sampling pairs of action states (s, a), using stochastic gradient ascent to maximize the probability that human movement is selected in state s.
The Policy Network Enhancement Learning is used for the second stage of the training pipeline aims to improve the policy network through learning policy gradient reinforcement (RL). The policy network RL $p\rho$ is identical in structure to the policy network SL and its weights ρ are initialized to the same values, $\rho = \sigma$. We use a reward function $r(s)$ which is zero for all non-temporal time steps $t < T$. The result $zt = \pm r(sT)$ is the reward of the terminal at the end of the game from the perspective of the current player. The weights are then updated at each step of time by ascending the stochastic gradient in the direction that maximizes the expected result

Results obtained

Using this search algorithm, the AlphaGo program achieved a 99.8% success rate over other Go programs and defeated the European Go European champion by 5 games to 0. This is the first time a computer program has defeated a professional human player in full play Go, a feat previously thought to be at least a decade ahead.