Mastering the game of Go without Human Knowledge

Before we begin, we must go through a symbolic definition of a go problem. Go problem, board 19 × 19 = 361 intersections are available for fall, three states for each point, white (with 1), black (with -1), and no piece (with 0). For the time being, we can't consider the situation where we can't fall. Then we have 361 places to go. The next step is to use a 361-dimensional vector.

In the new network, a deep neural network with the parameter θ (which needs to be constantly adjusted through training) is used.

F θ . This network structure is based on the Residual Network convolutional network, which contains 20 or 40 Residual Blocks (remainder modules), and adds batch normalized batch normalization and non-linear rectifier non-linearities modules. In the neural network, the input of the system is a 19×19×17 0/1 value: Now the board state →s and 7-step history book. The last position records black and white, 0 white and 1 black. The neural network has two outputs, a roll probability, which indicates the probability of the next step at each possible position (362 output values) and an evaluation value, indicating that the player who is currently preparing the current move is entering the eight-step historical situation. The winning rate under →s (between [-1, 1]).

We know that the initial Monte Carlo tree search algorithm uses random simulations and uses the situation function assisted strategy function as a reference for the simulation in AlphaGo 1.0. In the latest model, the Monte Carlo search tree uses the output of the neural network f θ as a reference for the fall. Using each step of the MCTS, the self-taught game and the reinforcement neural network are trained in the iterative process of the learning algorithm. Initially, the use of a complete random fall training lasted for about 3 days. In the training process, there were 4.9 million self-taught games, about 1600 simulations per MCTS, and 0.4 seconds for each step. 700,000 Mini-Batches in 2048 locations were used for training. After 24 hours of learning, intensive learning programs without artificial factors defeated supervised learning methods that mimic human chess. After that, the final AlphaGo Zero used 40 residual modules and trained for nearly 40 days. In the training process, 29 million sets of self-playing games were generated, and 3.1 million Mini-Batches were used to train the neural network. Each Mini-Batch contained 2048 different states. In the end, AlphaGo Zero and AlphaGo Master had a score of 89:11, limiting one game to two hours in the game.