

The review of AlphaGO.

This paper attempts to train a program to play game GO that even can beat the best Go players of the world.

The game GO, which has a potential of 250^{150} move sequence, makes algorithm such as min-max or alpha-beta prune not workable. In this paper, two novel neural networks are mentioned: one is policy network, the other is value network.

The policy network uses deep convolutional neural network, which used widely in image recognition. The input is the history GO games, and the output is the probability distribution of Go players next movement. With policy network, program can better predict Go players' selection, and also can be used in value network.

The value network is similar with policy network, but the output is a single value, not a distribution. The value network is used by program to choose the best movement. Here, value network value = $(1 - r) * (\text{value network}) + r * (\text{the outcome in this node of random rollout played out until terminal})$, $0 \leq r \leq 1$

AlphaGo uses Monte Carlo tree search as basis. First, when program plays, it choose highest score node valued by value network. Then it predicts user's next movement, sort user's score, and expand the MCT from highest score to lowest score.

The interesting part is the number of history Go games is small, and causes the overfitting of policy network. So AlphaGo team use reinforcement learning to improve it, that is use two neural network to play together, and use win/lose as penalize.

The result is AlphaGo beats the best Go player of the world, several times.

Some personal thoughts:

1. Use convolution neural network is a good shot. Because CNN works quite well for image processing, and then the Go game can be easily map to a image.
2. Policy network is really nice. The go game player has the game pattern, and players intended to use the same pattern in similar situation.
3. While in policy network training, they find network is overfit and use two network to player with is a interesting try.