AI1603-3 人工智能编程实践 – 五子棋竞赛 – 第13组

王凯灵、卜家梓、李奇睿

2022.7.13

上海交通大学 SHANGHAI JIAO TONG UNIVERSITY

Lo_Gobang_151

Feature:
First finished on 6.30.
100% original code and scoring method.

Updated versions:

1. **Score optimization**(frequent)
2. **Double mode:**
Two scoring policies. In the beginning or when chess pieces become dense, the agent should be more likely to defend than attack. Just like bosses in RPG games will change mode under certain situations.
3. **Search and pruning(Fig. 2.):**
The agent makes decision only according to the current state, which means it lacks ability to make temporary sacrifice to get higher score in the future. I tried to make a few simulations towards future to search for the better move.

Win rate are shown in the report.

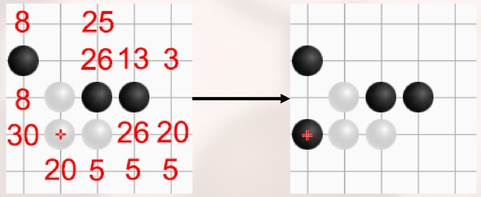

Fig. 1. Basic Logic

Basic logic(Fig. 1.):

1. Scan the map in four directions
2. Add score to each position if certain situations are found.
3. Select position with max score.

$$arg\max_{p} score(p)$$

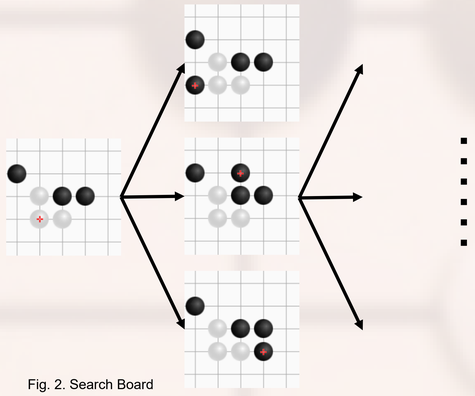

Fig. 2. Search Board

Advantages:

1. Fast and easy.
2. Easy to explain.

Disadvantage:

1. Too easy, anyone can realize.
2. **TOO MANY IF.**
3. As for the course, this method use little python coding skills.
4. Once finished, its chess skill level is fixed.
5. Time limit exceed(for the search method only)

Lo_Gobang_Zero

Feature:
Reinforcement learning. Deep Q Network.
First finished on 7.4 with pyTorch.
Designed action reward function.
Various training modes.
100% original code.
The **first** gobang AI based on TensorFlow 2.x on GitHub

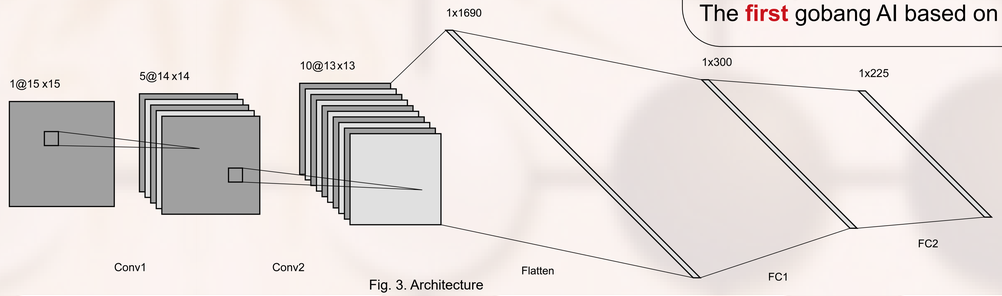

Fig. 3. Architecture

Result of the model trained with coach:
Beats random 1000:0
Can last over 20 steps playing with coach.

Chess game generation:
Use 2 Lo_Gobang_151s(151 below) to generate game experience of 6000 rounds, which is used to train a single agent. This agent is then trained with itself for 30000 rounds. The loss curve falls more steady(Fig. 4.).

About my DQN:
The network architecture is like Fig. 3. Conv is to attract feature of a chessboard block, and max pooling is avoided.
In reinforcement learning, the agent needs to explore, and the explore step is given by a 151.
The reward function is a scan-and-assess method, but it's different from the scoring method used in 151.

Self train:
In this case, two randomly initialized agents take turns to teach each other for 3000 rounds. The loss keeps low but is unstable. This model is also far better than random.

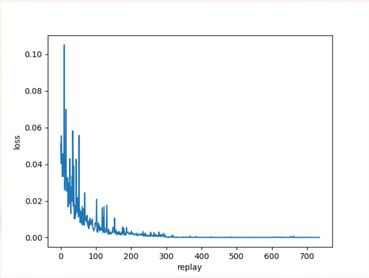

Fig. 4. Loss coach

Key parameters:
Replay size: 50; Update frequency: 300
Learning rate: 0.1; Q factor:0.5
Training platform:
Intel i9-11980HK, RTX3080laptop16gvr

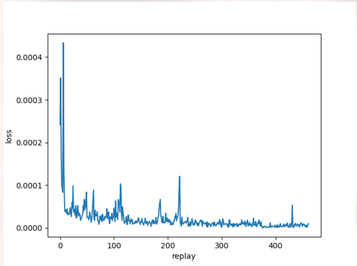

Fig. 5. Loss self

Code Repo：https://github.com/Loping151/openFiles/tree/main/2022Spring/Lo_Gobang

Email: wangkailing151@gmail.com