

强化学习实验报告

四子棋

张恒硕

2212266

人工智能学院

2025 年 3 月 6 日



南开大学
Nankai University



目 录

图 片

表 格

1 实验目的

1. 以井字棋程序为例，了解基础的强化学习算法代码。
2. 将井字棋程序改写成四子棋程序，并检验不同训练轮数的智能体对弈结果。

2 代码改动

2.1 四子棋部分

将棋盘尺寸、胜利条件中的 3 改成 4，这里对修改的代码不做展示。

2.2 交互部分

将 9 键的交互部分拓展成 16 键，以适应四子棋。下图图??展示了交互界面以说明修改成果：

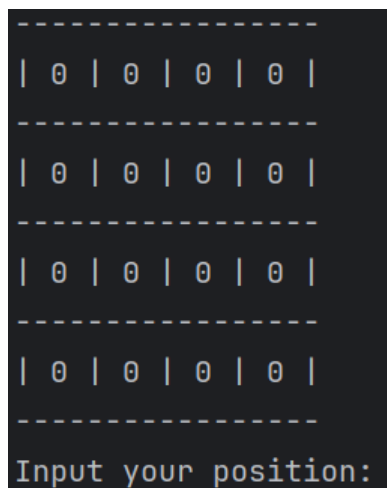


图 1: 棋盘

2.3 训练部分

源代码中，两个智能体的训练轮次是一样的，为了比较不同训练轮次下的对弈结果，对 `train` 函数进行了如下修改：

```

1 def train(n1, n2, print_every_n):
2     \
3     total_epochs = max(n1, n2)
4     for epoch in range(1, total_epochs + 1):
5         winner = judger.play(print_state=False)
6         if epoch <= n1:
7             player1.backup()
8         if epoch <= n2:
9             player2.backup()
10    \
11 if __name__ == '__main__':
12    \
13    train(n1=1000, n2=1000, print_every_n=100)
14    \

```

在每个轮次的训练中，更新未达到轮数的一方的策略，对已达到轮数的一方的策略不做变更。

2.4 简化部分

由于四子棋的可能状态较井字棋多很多，反复遍历会消耗较长时间，仅在首次运行时进行遍历，并保存结果，供后续测试时使用。为达到这一目的，增加了两个函数，并修改了主程序：

```

1 def save_all_states(all_states, filename='all_states.pkl'):
2     with open(filename, 'wb') as f:
3         pickle.dump(all_states, f)
4
5 def load_all_states(filename='all_states.pkl'):
6     if os.path.exists(filename):
7         return pickle.load(open(filename, 'rb'))
8     return None
9
10 if __name__ == '__main__':
11     all_states = load_all_states()

```

```
12     if not all_states:
13         all_states = get_all_states()
14         save_all_states(all_states)
15     \\\
```

3 实验结果与分析

为比较两个智能体在不同训练轮数下的博弈结果，经过实验得到下表表??:

表 1: 智能体博弈结果（智能体 1 胜率，智能体 2 胜率）

智能体 2\ 智能体 1	10 ³	10 ⁴	10 ⁵
10 ³	0.76,0.12	0.93,0.03	0.84,0.00
10 ⁴	0.03,0.84	0.38,0.29	0.96,0.00
10 ⁵	0.00,0.97	0.00,0.93	0.00,0.00

可以获得以下信息：

- 1. 因为平局的存在，双方胜率之和小于 1。
- 2. 四子棋是先手占优的游戏，相同训练轮次下，先手的智能体 1 占优。
- 3. 经验丰富的双方最终趋于平手。

4 实验总结

本实验作为一个强化学习的基础例子，指明了强化学习算法的实现过程，并强化了“状态”这一概念。四子棋仅仅 4 * 4 的棋盘就已经需要不小的算力进行遍历，更大的围棋棋盘将是更复杂的工程。