强化学习—小老鼠走迷宫

班级：自 73

姓名：陈昱宏

学号：2017011507

一、任务要求：

在给定 6*6 大小的迷宫下，使用强化学习算法训练智能体，让智能体完成走迷宫的任务；除了基本的 6*6 底图外，尝试改变地图大小或地图内容，还有加上老鼠夹（踩到直接 game over），重新训练智能体，完成走迷宫的任务。

二、编程语言和环境配置：

本项目以 Python3+PyQt5 完成，详细环境配置见 Readme.txt。

三、算法设计与实现：

（一）类结构设计：

1.Maze 类：

2.Agent 类：

（二）算法介绍:

这次项目采用时序差分的方式进行仿真训练，具体算法使用QLearning，分别采用ϵ贪心和贪心算法进行 Q 表更新，伪代码如下:

Void Train(agent, maze, lr, γ, ϵ, trainingTime)

For i = 0,1, … , trainingTime − 1:
$\epsilon \leftarrow \epsilon - 1/(\text{trainingTime} // 2)$
S ← randint(0~agent. stateNum)
while maze. GetState(S)! = 1:
S ← randint(0~agent. stateNum)
time ← 0
while maze. GetState(state)! = 0.9:
Choose A and S by the ϵ − greedy derived from agent. qTable
Take A, observe R, S′
agent. qTable[S, A] ← agent. qTable[S, A] + $lr * (R + \gamma \cdot \max_{a \in A}$ agent. qTable[S′, a] − agent. qTable[S, A])
S ← S′
time + +
if time > 200 or maze. GetState(S) == 0.5:
break

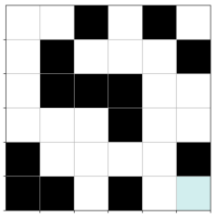
为了让智能体学习到迷宫的信息，对于每一个行为设定了惩罚，以下为各种情况的惩罚分配:

$$R = \begin{cases} -0.75, \text{撞墙或超出边界} \\ -5，\text{踩到老鼠夹} \\ -2，\text{回到起点} \\ -0.1，\text{走到正常格子} \\ 1，\text{抵达终点} \end{cases}$$

四、训练流程:

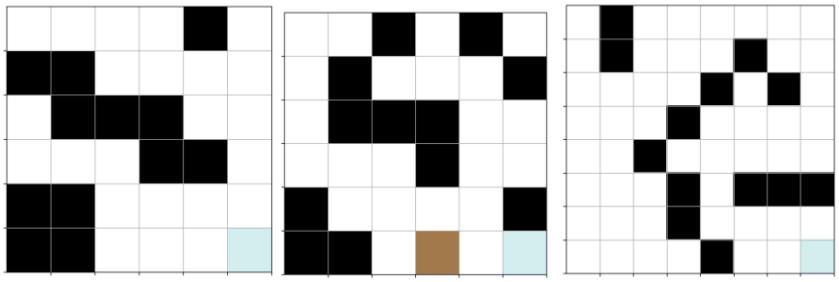
本项目提供了四个内建地图，分别是 Maze6-1.npy、Maze6-2.npy、Maze6-3.npy、Maze8.npy，各个地图如下：

Maze6-1.npy：



黑色格子代表墙，白色格子代表代表可以走的地方，咖啡色的格子代表老鼠夹，浅蓝色的格子代表终点。

Maze6-2.npy：　　Maze6-3.npy：　　Maze8.npy：



为了方便助教演示，在 Agent 文件夹中预训练了对应的智能体，其命名规则为 Agent(地图名).npy。

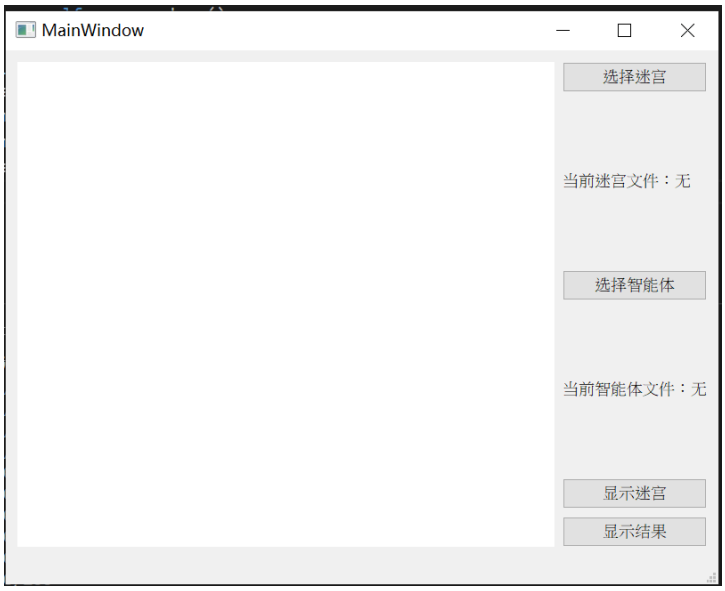
如果需要训练新的地图，请先创建一个$m \times n$的 ndarray，并在 array 中的数值，用 1 代表可以走的区域，0 代表墙体，0.9 代表终点，0.5 代表老鼠夹，最后保存成.npy 文件。

训练智能体时，请执行 Train.py 文件，并按照提示内容进行输入，输入的范本如下：

```
请输入迷宫文件：./Maze/Maze8.npy
是否使用预训练模型？（y/n）:y
请输入智能体的预训练模型文件名：./Agent/Agent(Maze8).npy
请输入训练次数：200
episodes:20/200
episodes:40/200
episodes:60/200
episodes:80/200
episodes:100/200
episodes:120/200
episodes:140/200
episodes:160/200
episodes:180/200
episodes:200/200
训练完成！
请输入训练结果保存的文件名：./Agent/Agent(Maze8).npy
```

五、界面介绍：

为了方便演示，我设计了一个界面来显示，界面基础画面如下：



点击右侧的选择迷宫和选择智能体按钮可以选择要演示的迷宫和训练好的智能体，以 Maze6-3.npy 为例进行示范，点击显示迷宫可以显示以下画面：

最后点击显示结果即可观察训练好的智能体能不能顺利到达终点。

六、实验总结:

这次实验透过老鼠走迷宫的形式，让我练习了强化学习算法的实践，在老师上课说的几种算法，我选择了 QLearning 算法进行实践，将课本上的计算实际在程序中完成，更能将理论和实践结合。