人工智能基础大作业三：强化学习

自75 常成 2017010252

目录

1 任务描述

在方形迷宫中，有一只老鼠和一块蛋糕，老鼠和蛋糕的起始位置都是固定的，老鼠在迷宫中探索，最终目的为找到并吃掉蛋糕。

大作业任务如下：

1. 使用强化学习算法，对于给定的迷宫，训练老鼠在迷宫中寻找蛋糕。

2. 自行生成不同迷宫（尺寸、地图），完成前述操作。

3. 若迷宫中存在老鼠夹子，且位置固定，完成前述操作。

4. 考虑时变因素，如迷宫的某些格子会周期性产生火焰。

2 问题建模

强化学习是通过智能体与环境交互进而产生反馈，利用一系列的决策与状态转换来达到预期目标。本次大作业的老鼠寻找蛋糕的问题就是一个经典案例，老鼠在迷宫中经过多次实验，充分了解环境信息，这一阶段称为"学习"过程。在这一过程中，老鼠可能不断出错，甚至会出现撞墙、穿墙、接触火、碰鼠夹等危险行为，但是这些都会作为经验存储起来，在经过多轮次的学习之后，老鼠可以利用已经存储的经验进行探索，进行找到蛋糕所在地。

将这一问题抽象为数学模型，可以表示为：

状态表：$state = [0, 1, 2, \cdots, n-1]$　其中n代表迷宫格数

动作表：$action = [0, 1, 2, 3]$　$0, 1, 2, 3$分别代表动作$up, down, left, right$

奖励表：$rewards = [$执行动作至迷宫格i这一过程的奖励$]$　$0 \leq i \leq n-1$，根据迷宫格i是路径、墙、鼠夹、火焰分别设定不同的奖励值/惩罚值。

行动价值表：$Q[state][action]$　$state \in [0, 1, 2, \cdots, n-1]$　$action \in [0, 1, 2, 3]$

行动价值表示在当前状态，采取某个动作的行动价值，是依据贝尔曼方程更新的，如图1。在本次大作业中，如果采用离线策略学习，进行时序差分控制，主要有两种方案：

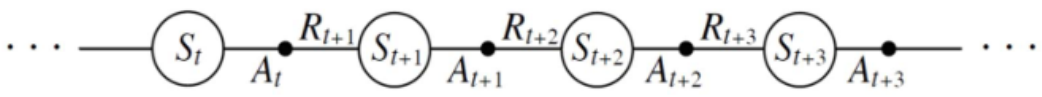

图 1: 行动价值链

1. E-SARSA

$$Q\left(S_t, A_t\right) \leftarrow Q\left(S_t, A_t\right)+\alpha\left(R_{t+1}+\gamma \sum_{a \in A} \pi\left(a \mid S_{t+1}\right) Q\left(S_{t+1}, a\right)-Q\left(S_t, A_t\right)\right)$$

其中$\pi\left(a \mid S_{t+1}\right)$表示在$S_{t+1}$状态采取动作$a$的概率，所以E-SARSA方法相当于取下一状态各方向的行动价值的平均，选取期望值进行行动价值表的更新，是"小心谨慎"的更新方案。

2. Q-learning

$$\boldsymbol{Q}\left(\boldsymbol{S}_t, \boldsymbol{A}_t\right) \leftarrow \boldsymbol{Q}\left(\boldsymbol{S}_t, \boldsymbol{A}_t\right)+\boldsymbol{\alpha}\left(\boldsymbol{R}_{t+1}+\gamma \max_{a \in A} \boldsymbol{Q}\left(\boldsymbol{S}_{t+1}, \boldsymbol{a}\right)-\boldsymbol{Q}\left(\boldsymbol{S}_t, \boldsymbol{A}_t\right)\right)$$

Q-learning方法中选取下一状态的各方向行动价值的最大值，是具有"冒险精神"的更新方案。算法表示如Algotithm 2

Algorithm 1 Q-learning

Input: step size $\alpha \in (0,1]$,small $\epsilon > 0$
Output: Initialize Q(s,a),for $s \in S$,$a \in A(s)$,arbitrarily except that $Q(teminal, \cdot) = 0$

```
for each episode do
    Initialize S
    for each step of episode do
        Choose A from S using policy derived from Q(e.g.,ε − greedy)
        Take action A,observe R,S'
        Q(S_t, A_t) ← Q(S_t, A_t) + α(R_{t+1} + γ max_{a∈A} Q(S_{t+1}, a) − Q(S_t, A_t))
        S ← S'
        Until S is terminal
    end for
end for
```

在本次大作业项目中，经过两种更新方案的尝试，可以明显发现采用Q-learning的收敛速度更快，学习轮次更少，并且在最终寻找过程中，表现为小鼠四处游荡的现象大大减少，所以Q-learning在此问题背景下显然是更适合的。

3 必做解决方案

在必做问题中，引入墙壁、鼠夹两种障碍，规定墙壁的rewards值为-10，鼠夹的rewards值为-30，蛋糕的rewards值为10，其余正常路径设为0。

首先随机生成迷宫，利用宽度优先搜索算法，判断所有相通的结点，如果迷宫不可解，即老鼠或蛋糕周围被墙壁或鼠夹包围，系统给出提示，可以重新生成随机迷宫。

规定老鼠在边界时的运动不可越出边界，其余位置的运动按照ϵ-greedy策略进行选择：

$$\pi(a|s) = \begin{cases} 1-\varepsilon+\varepsilon/m & \text{if } a = \arg\max Q(s,a) \\ \varepsilon/m & \text{otherwise} \end{cases}$$

在学习过程中，设定ϵ为0.2，即老鼠有20%的概率随机选择动作，其余80%的情况是选择行动价值表中可以产生最大行动价值的动作。这样设置的目的是让老鼠既可以"学习"，又可以加入"探索"的环节，充分熟悉环境信息，否则老鼠有可能陷入当前局部最优路径，而无法学习到全局最优路径。每进行一次动作，更新行动价值表并进行状态转换，当老鼠吃到一次蛋糕，认为结束一轮学习。

在经过一定轮次的学习后，当行动价值表(Q表)接近收敛时，老鼠便可以开始正式的寻找过程了，在寻找过程中，老鼠按照行动价值表采用绝对贪婪策略$a = \arg\max Q(s,a)$选择动作，如果有多个最大值，则随机选择一个，并进行状态转换，直至达到终点。

在必做问题的界面更新过程中，采用了多线程的方法，原因在于程序运行过程中需要一直动态刷新界面，如果都在主线程中执行，界面会出现假死现象，所以需要另开线程，在子线程中执行状态转换和行动价值表更新等操作，每当出现一次状态转换，由子线程通过PYQT的信号槽机制，向主线程发送信号，主线程收到信号后更新界面，这样便实现了界面的动态显示。

4 选做解决方案

当考虑时变因素后，使迷宫的某些格子产生周期性火焰，我们假设火焰的周期是已知的，并根据此周期设计学习训练方案，在本次大作业项目中，我主要尝试了如下三种时变系统的训练方案：

4.1 增加火焰行动价值表

在原来的行动价值表(A)不变的基础上，我们考虑增加另一份火焰存在时的行动价值表(B)，并在新的行动价值表B更新过程中，设定火焰的rewards值为-10，原有行动价值表的rewards不变，即和正常路径相同为0。

这样在学习过程中，我们可以按照两种环境分别训练，当训练达到一定轮次后，两份行动价值表均已迭代至收敛，老鼠便可以开始正常寻找过程，设定老鼠移动的速度为0.5s一步，火焰周期为2s(1s有火，1s无火)，这样记录老鼠移动的步数，如果移动步数被4除的余数小于2(此时环境无火焰)，老鼠便按照A表中最大贪婪地选择自己的动作，而如果移动步数被4除的余数不小于2(此时环境有火焰)，老鼠按照B表选择。这样在移动时不断切换行动价值表，直至到达终点。

这种方法虽然可行，但是存在明显的局限性。可以想到，如果老鼠附近无火焰，它按照A表选择下一步的路径，当老鼠移动到那个位置，假设此时附近环境出现火焰，老鼠切换行动状态表，它按照B表选择下一步的路径，则必然会远离火焰(火焰的rewands值很低)，则有很大的概率回到原来的位置，而当火焰消失后，老鼠按照A表又会靠近过来，这便形成了振荡现象。

在程序的测试过程中，振荡现象十分常见，事实上我们将火焰周期调为小鼠单步移动时间的4倍，已经一定程度上缓解了该问题，老鼠在多次游荡后有一定概率跳出振荡；如果将火焰周期设定为1s，小鼠每走一步都要更换行动价值表，那么振荡现象会更加严重，可能占所有迷宫情况40%以上。不管怎样，振荡问题在这种方法下是可能出现的问题，导致老鼠不一定每次都会成功吃到蛋糕，我们还需要考虑其他方法。

4.2 增加停等动作

在已知火焰周期的情况下，设定老鼠移动的速度为0.5s一步，火焰周期为2s(1s有火，1s无火)，我们可以为小鼠增加停等动作来适应有火焰的环境，即仍然是一份行动价值表，小鼠按照无火焰的环境正常去学习，不断进行状态转换和行动价值表的更新，待迭代至接近收敛，小鼠按照行动价值表最大贪婪地选取动作，并记录时间数(总时间/0.5)，如4.1部分所述，根据时间数被4除的余数判断当前环境是否有火焰，如果在S状态下选择动作a，到达状态是S'，而S'位置此时有火焰，小鼠便会在原地停留等待，不采取任何移动动作，待过一段时间判断无火焰后，再按照之前的选择移动。

这种解决方案确实可以有效解决振荡问题，并且因为训练过程中没有加入火焰相关的信息，小鼠每次的寻找路径都十分接近最优路径，只是在最优路径

上存在等待动作。

但是这似乎违背了强化学习的本意，相当于小鼠未经学习便拥有了"上帝视角"，可以知晓环境中的火焰信息，因此虽然这种方法可行，但不属于强化学习的一种方法。

4.3 扩展行动价值表

经过前面的考虑之后，我们可以考虑将上述两种方法结合起来，形成一种新的有效的强化学习解决方案，设定火焰周期为小鼠单步移动时间的2倍，即设定老鼠移动的速度为0.5s一步，火焰周期为1s(0.5s有火，0.5s无火)。

首先扩展动作表为：

$action = [0, 1, 2, 3, 4]$　$0, 1, 2, 3, 4$分别代表动作$up, down, left, right, wait$

再扩展状态表：

$state = [0, 1, 2, \cdots, 2n-1]$　其中n代表迷宫格数

再扩展行动价值表为：

$Q[state][action]$　$state \in [0, 1, 2, \cdots, 2n-1]$　$action \in [0, 1, 2, 3, 4]$

注意到此时我们将状态数进行翻倍处理，即状态i　$i \in [0, 1, 2, \cdots, n-1]$代表无火焰各迷宫方格状态，状态$i$　$i \in [n, n+1, n+2, \cdots, 2n-1]$代表有火焰时各迷宫方格状态，这样在学习过程中便可以正常训练，更新迭代$(2n-1) \times 5$的行动状态表，而火焰的rewards仍设定为-10，且只在状态$[n, n+1, n+2, \cdots, 2n-1]$为-10，在状态$[0, 1, 2, \cdots, n-1]$视为无火焰，原火焰位置的rewards仍设为0。

举例说明这种训练方法，若迷宫阶数为6×6，小鼠起始位置状态为0，此时环境中有火焰，如果小鼠选择向右移动，则达到的状态为$1 + 6 \times 6 = 37$，这时环境变为无火焰，若小鼠选择向下移动，则达到的状态为$37 + 6 - 6 \times 6 = 7$，小鼠的状态就这样在$[0, 1, 2, \cdots, n-1]$和$[n, n+1, n+2, \cdots, 2n-1]$之间来回切换，并不断根据rewards值更新迭代行动价值表。

这样在小鼠在正式寻找过程中，便可以按照行动价值表最大贪婪地选择动作，躲避火焰的同时，容易学习到最优路径。

值得说明的是，如果训练过程中不加入停等动作，即action表中没有4选项，那么小鼠仍然可能出现振荡现象，但这种振荡相对于第一种方法已经很少见，在扩展动作表后，振荡现象基本可以完全消除。

这种方法是上述三种方案中最有效且最符合要求的解决方案，使得小鼠在环境中学习到各种障碍的信息，并作出最优决策。

5 UI界面设计及操作

5.1 程序开发

本程序采用Python 3.6语言进行开发，界面框架采用PYQT5，如需编译源码，需配置好PYQT5，pyqt5-tools，threading等包，生成的exe文件在dist文件夹中，由pyinstaller打包生成，文件夹已含有程序运行依赖的图片资源文件，文件夹中有$Search, Search_wait, Search_extend$三个exe文件，分别对应着选做解决方案的三种方法，可分别运行查看效果。

注:效果最好的为$Search_extend.exe$，对应选做4.3方案。

5.2 模式选择

程序运行后，可选择难度模式，困难模式意味着障碍增多，用户可选择迷宫阶数(6-10阶)，加入鼠夹以及加入火焰，点击生成迷宫即可随机生成，用户需要设定小鼠的学习轮次(建议100+)。

5.3 运行样例

必做部分的界面如图2所示

若选择10阶迷宫，困难模式，加入鼠夹及加入火焰，则如图3，点击开始学习，并勾选学习动画，即可看到小鼠在迷宫中训练学习的场景，为加快显示，这里设置小鼠移动速度为0.01s每步，界面上方会显示学习的轮次，用户可随时取消勾选，小鼠会自动学习完毕。

学习完毕后，界面上方显示learned，这时点击开始寻找，如图4

注：寻找过程中，如果在无火焰时，小鼠穿过原火焰位置，界面可能会出现该步显示不出来。

待小鼠吃到蛋糕，界面上方提示Congratualations，如图5

5.4 UI界面容错机制

当迷宫不可解时，利用宽度优先搜索，系统判断后给出提示，如图6

当正在演示寻找动画或学习动画时，用户点击其他按键，系统会给出提

图 2: 必做部分界面

图 3: 选做部分学习

图 4: 开始寻找

图 5: 完成任务

图 6: 迷宫不可解

示，如图7 8

6 项目总结

本次大作业让我进一步理解了强化学习算法，更加熟悉PYQT界面框架的应用，对于多线程刷新界面和QT信号槽传递消息的机制也有了更好的掌握，其中选做部分的方法思考也锻炼了我全面考虑问题的能力，是一次收获满满的大作业。

感谢老师和助教们一学期来的辛苦付出！

图 7: 学习容错

图 8: 寻找容错