

# 机器学习: 编程作业 #3

Due on Jan. 30, 2023 at 23:59 p.m.

80230973-0 理科楼 A112 1-2

李文昊 无 92 2019011612  
li-wh19@mails.tsinghua.edu.cn

## 作业 3.2

使用 RL 实现猫捉老鼠

## 1 平台介绍

强化学习由于没有统一的 Python 库, 因此使用基础的库来实现。整体的代码环境为 Python3.11.1, 所用的库依赖为 numpy1.24.1, tkinter 为 Python3 自带库。

## 2 算法简介

本次编程作业选取 Q-Learning 算法来学习猫捉老鼠的策略, 核心思想是建立 Q 表来记录每个状态和动作所获得的奖励, 通过不断更新 Q 表的值来达到学习策略的目的, 其中 Q 表的更新算法如下所示

---

### Algorithm 1: Q Learning

---

```
Initialize Q(s,a) arbitrarily;  
foreach episode do  
    Initialize s;  
    foreach step do  
        action  $\leftarrow$  greedy(Q, s,  $\epsilon$ );  
        reward, s'  $\leftarrow$  observe(Q, s, action);  
        Q(s,a)  $\leftarrow$  Q(s,a) +  $\eta$ [reward +  $\gamma$ maxs' - Q(s,a)];  
        s  $\leftarrow$  s';  
    end  
end
```

---

其中  $\epsilon$  表示随机选取行动的概率,  $\eta$  表示学习率,  $\gamma$  是下一状态的影响因素, 在测试中, 分别设置这些参数为  $\epsilon = 0.9, \eta = 0.1, \gamma = 0.9$ 。

### 3 效果演示

为了简单起见，用红色方块表示猫，绿色圆形表示老鼠，黑色方块表示障碍，障碍的位置为随机生成。在 `visual.py` 中顶部可以修改参数改变格子的数目和障碍个数，在同文件的 `render` 函数中注释掉 `_mouse_wander` 函数可以取消老鼠的随机游走。动态的演示效果见 `cat_mouse.gif`

#### 3.1 老鼠固定

固定老鼠时，强化学习如图1所示。此时可见，在 2、30 次训练之后，猫已经能够比较稳定地抓住老鼠了，赢多输少，且花费的步骤也较少，此时还有输的情况是因为先前设置参数时有一定概率随机选取动作，而不是始终选取最优的动作，从而导致碰到障碍。

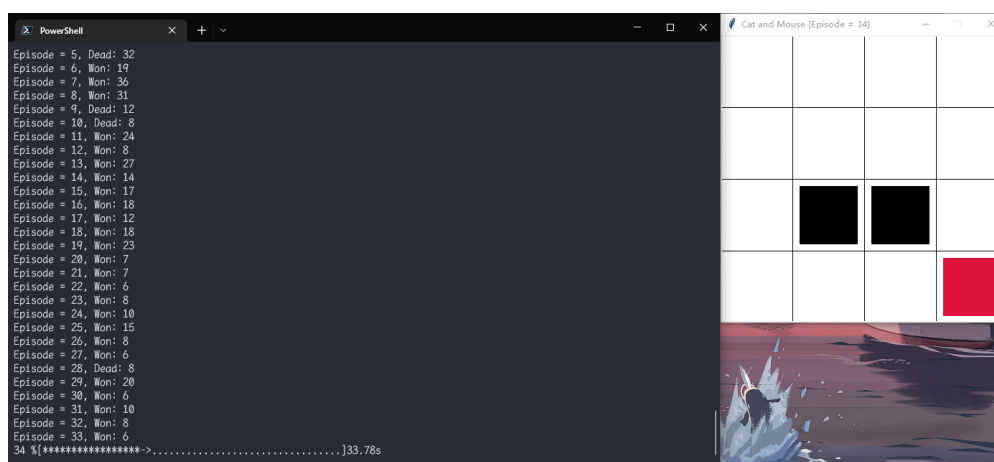


图 1: 老鼠固定

#### 3.2 老鼠游走

老鼠游走时，强化学习如图2所示。此时，由于老鼠在随机游走，学习速度编码，在 6、70 次训练之后，猫才能够比较稳定地抓住老鼠。

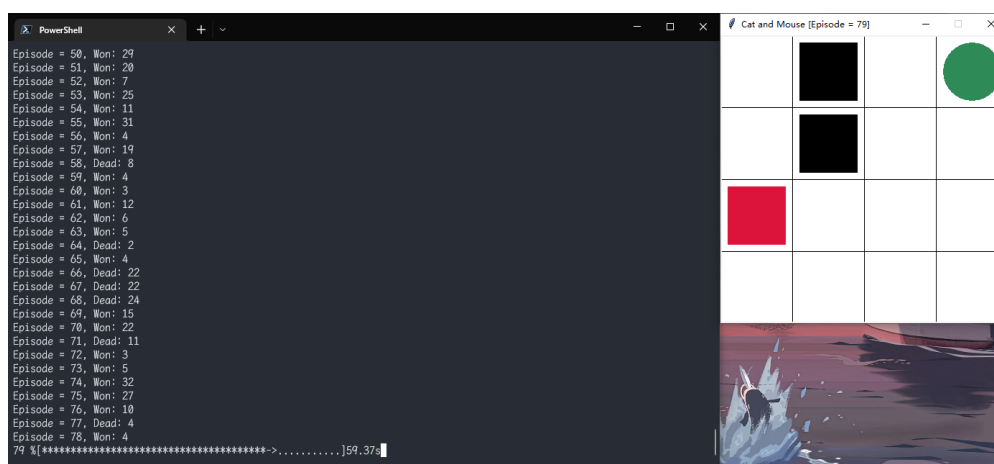


图 2: 老鼠游走

### 3.3 增加格子

增加格子到  $50 \times 50$ ，同时障碍为 40 时，强化学习如图3所示。此时，由于棋盘过大，需要更多的迭代次数才能够更新 Q 表，时间过长，但是整体的逻辑是一样的。

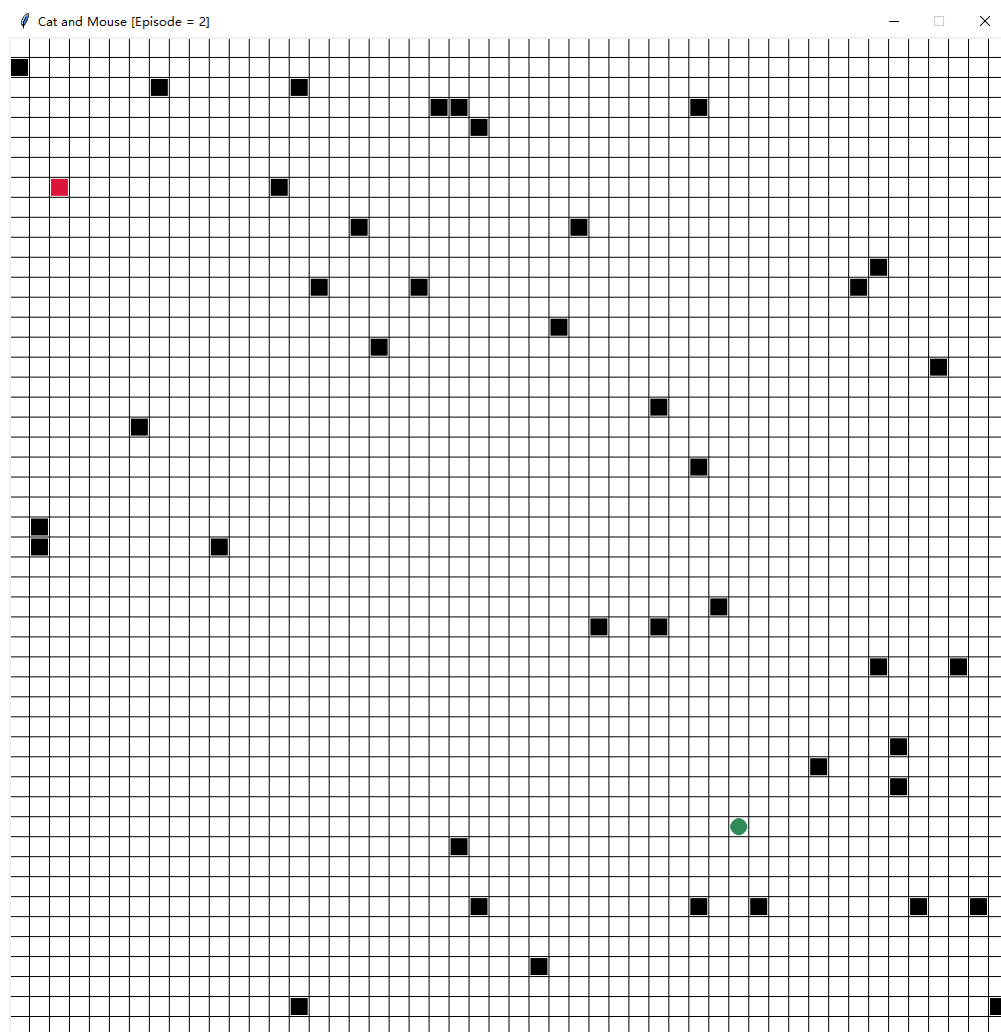


图 3: 增加格子

## 4 总结

通过这次强化学习的练习，我熟悉了 Q Learning 学习和更新的策略，同时体验了强化学习的趣味所在，收获颇大。最后感谢老师和助教的辛勤付出！