第8次作业

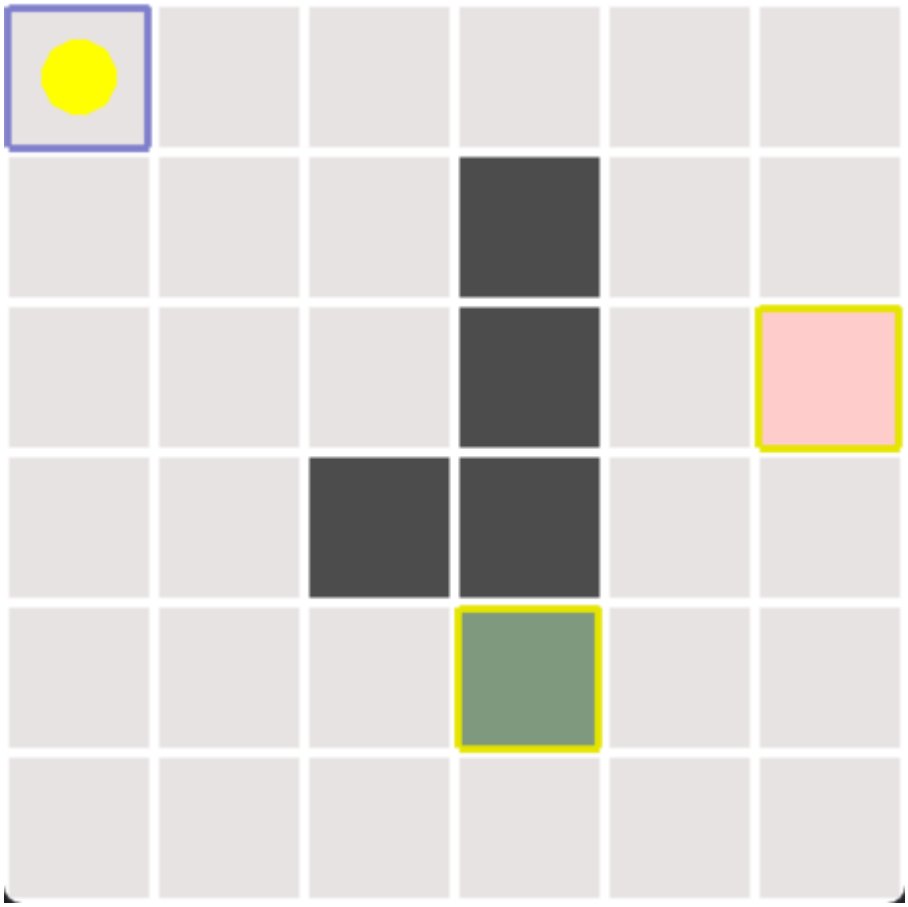
1.强化学习: 策略迭代与值迭代算法

在网格环境 `MiniWorld`上实现策略迭代或者值迭代算法之一. 该环境由一个 6×6 网格组成，其中黄色圆圈为智能体出发点，黑色格子为无法通过的墙壁，若智能体向着墙壁方向移动，则会停留在原地，地图边界的移动同理。带有黄色边框的格子为终止状态，即若智能体行动至此状态则整个 episode 结束。红色和绿色表示当前该状态的奖励值，奖励越高则绿色越深，奖励越低则红色越深。

状态空间：36（`0-35` 这 36 个整数，左下角为 0，向右 +1，向上 +6）

动作空间：4（`0-3` 这 4 个整数分别代表左、右、上、下）

奖励函数：普通格子 -0.1，到达终点(s=9) +1.0，掉入陷阱(s=23) -1.0。



要求如下:

- 将所有方法集成到类 `DPAgent`中. 该类的构造函数以强化学习环境 `env`作为输入.
- 在 `iteration()`方法上实现策略迭代算法或者值迭代算法, 以迭代误差界限 `threshold`参数作为输入(即策略/值误差小于这个值时结束迭代), 返回迭代后的值函数 `values`与策略 `policy`
- 返回的值函数 `values`与策略 `policy`的数据类型均为numpy数组. `values[s]`为状态 `s`下的V值, `policy[s][a]`为状态 `s`下采取动作 `a`的概率.
- 进行$\pi(s) \leftarrow \arg\max_a \cdots$操作时, 如果取值最大的动作不唯一, 则策略修改为等概率采取这些动作. 例如动作1,3均达到最大值, 则 `policy[s]=[0,0.5,0,0.5]`. 若只有唯一的动作1达到最大值, 则 `policy[s]=[0,1,0,0]`, 以此类推.

提示

- 状态 `s`为整数而不是网格坐标. 以6*6网格环境为例, `0<=s<36`, `s=0`对应网格(0,0), `s=1`对应网格(0,1), 以此类推
- `env.R`给出了奖励函数, `R[s]`为转移到状态 `s`后获得的奖励值
- `env._state_to_xy()`和 `env._xy_to_state()`给出了状态 `s`与格子坐标 `x,y`转换的方法
- `env.action_meaning`给出了动作 `a=0,1,2,3`对应的含义为 `"^", ">", "v", "<"`.
- `env.blocks`给出了墙壁(障碍物)的坐标位置, 需根据该信息和 `action_meaning`自行实现转移概率$P(s' \mid s, a)$的计算
- 环境集成了 `show_values()`和 `show_policy()`, 方便可视化结果.

实验环境安装指引参考

本次实验环境需要用到旧版本的gym库, 可能与其他作业环境不兼容. 因此为了顺利完成作业, 请按照以下指引安装实验环境:

1. 打开Anaconda Prompt终端, 新建python=3.7版本的conda环境(请不要在之前作业的环境上安装), 命名为 `hw8` (或者其他命名也可以)

   ```
   conda create -n hw8 python=3.7
   ```

2. 激活conda环境, 确保当前的环境是刚刚新建的环境(终端命令行最前面为 `(hw8)`)

```
conda activate hw8
```

3. 用以下命令安装依赖包(注意版本号, 若版本不同可能无法正常运行代码)

```
python -m pip install numpy==1.21.2 gym==0.10.0 pyglet==1.2.4 scipy==1.7.3
```