

作业内容

实现Dagger算法

作业描述

环境描述

本次作业的环境为gym库中的[蒙特祖马的复仇](#)，玩家可以通过选择动作来移动人物，拿到钥匙并走到出口。

###任务描述

请完成：

1. 与环境交互，在环境中采样并记录轨迹。
2. 依据[源代码](#)中的动作说明，了解相关动作的含义并对轨迹中的状态标记上正确的动作。
3. 依据正确标记的（状态，动作），用Dagger算法学习一个策略模型，可以基于输入的状态输出动作，使得玩家可以获得尽可能高的累计奖励。**禁止使用奖励信息（reward）来学习策略。**
4. 利用代码中内置的plot函数，绘制Dagger算法的性能图。

代码描述

代码文件夹code由'mian.py', 'arguments.py', 'Dagger.py' 组成。

'main.py': 包含了代码的主要结构，包括环境初始化、如何与环境交互的样例、算法性能展示方式等等。**你需要在其中实现Dagger算法的相关部分**，并用你的策略(agent.select_action)来玩游戏并进行性能测试。

'arguments.py': 包含了默认的参数，可以修改。

'Dagger.py': 包含了待填充的Dagger算法DaggerAgent，**请继承其中的DaggerAgent来实现你自己的算法。**

运行代码前请安装：

numpy、argparse、pickle、gym、matplotlib、PIL

提交方式

完成的作业请在课程页面的上传文件接口处上传提交。上传的格式为一份压缩文件，命名为'学号+姓名'的格式，例如'MG20370001张三.zip'。文件中需包含 'mian.py', 'arguments.py', 'Dagger.py', 'performance.png'（由main.py中的plot函数自动实现）和'Document.pdf'（一份pdf格式的说明文档），文档内容至少需要包含：

1. 实验效果说明。
2. 如何复现实验效果。
3. Dagger算法的实现说明。
4. 如果有相关的改进，也请在其中说明。

文档模板参见'Document2.tex'和'Document2.pdf'。

Tips

考虑后续的作业，建议大家提前安装并调试深度学习库，例如pytorch或tensorflow。

你可以先尝试玩一下蒙特祖马的复仇这个游戏。通过运行

```
python main.py --play-game True
```

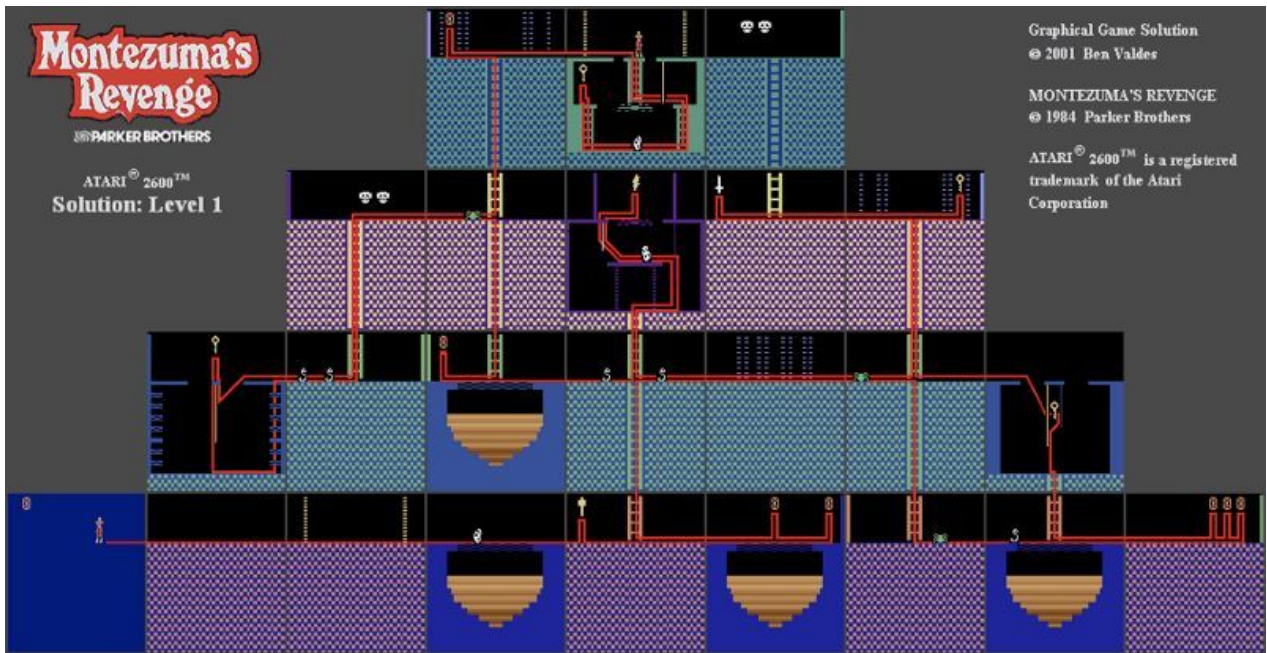
你可以在/code/imgs/screen.jpeg中看到你的当前游戏画面，并通过输入数字来选择你想执行的动作。

参考[源代码](#)，可以得到各个动作对应的含义：

ACTION_MEANING = {	
原地不动	0: "NOOP",
跳跃	1: "FIRE",
向上	2: "UP",
向右	3: "RIGHT",
向左	4: "LEFT",
向下	5: "DOWN",
	6: "UPRIGHT",
	7: "UPLEFT",
	8: "DOWNRIGHT",
	9: "DOWNLEFT",
	10: "UPFIRE",
向右跳	11: "RIGHTFIRE",
向左跳	12: "LEFTFIRE",
	13: "DOWNFIRE",
	14: "UPRIGHTFIRE",
	15: "UPLEFTFIRE",
	16: "DOWNRIGHTFIRE",
	17: "DOWNLEFTFIRE",
	}

只需要考虑0、1、2、3、4、5、11、12这几个动作即可。

蒙特祖马的复仇游戏地图如下：



同时，你也可以通过输入[gym](#)库中的其他游戏名来玩其他游戏，例如：

```
python main.py --play-game True --env-name AlienNoFrameskip-v0
```

思考题

在玩游戏的过程中标注数据与Dagger算法中的标注数据方式有何不同？这个不同会带来哪些影响？