Reinforcement Learning Task3

姚冠宇

2020211260070

先展示最终结果（程序可以直接运行）：

Sarsa和Q-learning各自迭代了500轮

Sara

最优路线：

```
['^^', 'vv', '<<', '>>']
```

分别对应上、下、左、右

最优路线的步数：

最终得到每个状态的动作表格（一个位置可能存在多个策略故采用这种方式打印）：

Q-learning;

最优路线：

```
['^^', 'vv', '<<', '>>']
```

分别对应上、下、左、右

```
Q-learning算法最终收敛得到的路径为：
oo oo oo oo oo oo oo oo oo oo oo oo
oo oo oo oo oo oo oo oo oo oo oo oo
>> >> >> >> >> >> >> >> >> >> >> vv
^^ ** ** ** ** ** ** ** ** ** ** EE
```

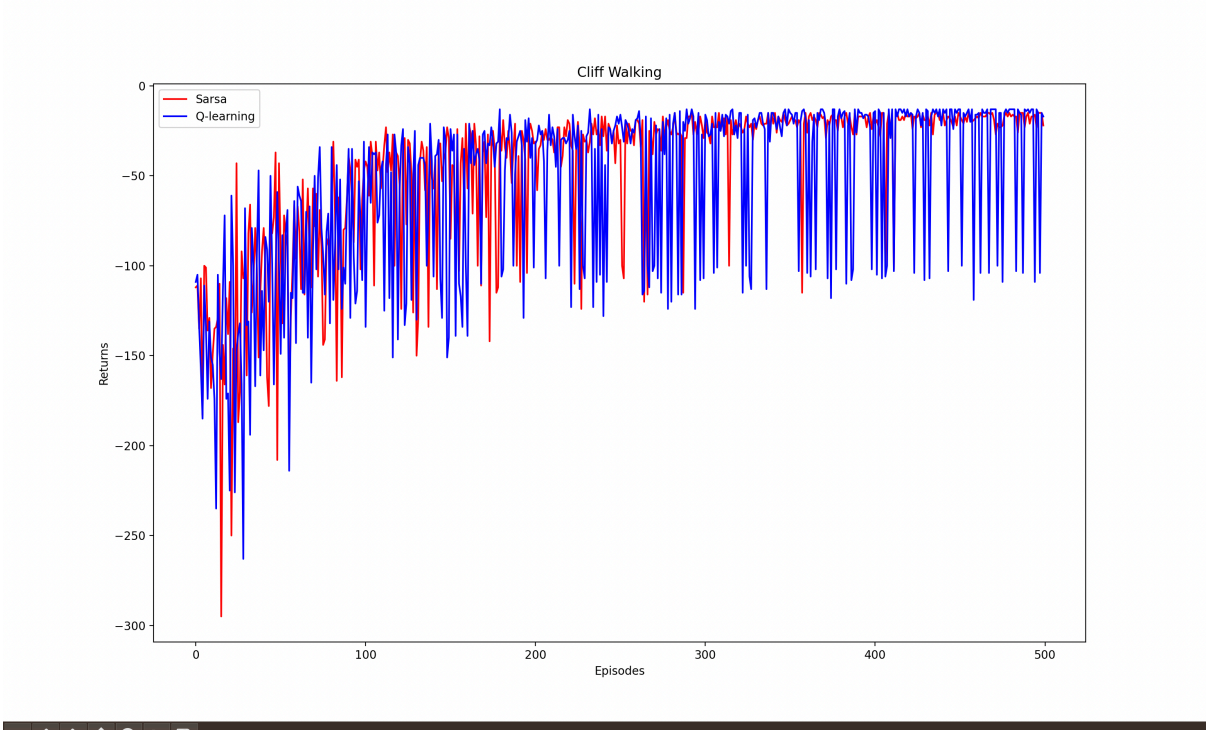
最优路线步数：

```
Q-learning算法最终收敛得到的步数为： 13
```

最终得到每个状态的动作表格（一个位置可能存在多个策略故采用这种方式打印）：

```
oo<o oo<o oo<o ovoo ooo> ooo> ooo> ^ooo ooo> ovoo ovoo ovoo
^ooo ooo> ooo> ^ooo ooo> ooo> ooo> ovoo ooo> ooo> ooo> ovoo
ooo> ooo> ooo> ooo> ooo> ooo> ooo> ooo> ooo> ooo> ooo> ovoo
^ooo **** **** **** **** **** **** **** **** **** **** EEEE
```

两种算法每个epoch的回报



问答题

$\pi_*(Q)$和$\pi_*(S)$是否相同？

答：不同，我们可以用价值迭代的思想来理解 Q-learning，即 Q-learning 是直接在估计Q*，因为动作价值函数的贝尔曼最优方程是

$$Q*(s,a) = r(s,a) + \gamma \sum_{s' \epsilon S} P(s'|s,a) max_{a'} Q*(s',a')$$

而Sarsa估计的是ϵ-贪婪策略的价值函数。这也导致Q-learning的结果一般比Sarsa算法要好。

我们把目标策略的行为打印出来后，发现Q-leraning更偏向于走在悬崖边上，这与 Sarsa 算法得到的比较保守的策略相比是更优的。 但是仔细观察 Sarsa 和 Q-learning 在训练过程中的回报曲线图，我们可以发现，在一个序列中 Sarsa 获得的期望回报是高于 Q-learning 的。这是因为在训练过程中智能体采取基于当前函数的-贪婪策略来平衡探索与利用，Q-learning 算法由于沿着悬崖边走，会以一定概率探索“掉入悬崖”这一动作，而 Sarsa 相对保守的路线使智能体几乎不可能掉入悬崖。

Sarsa 和 Q-learning 在训练过程中的回报曲线图在上一页已经附上，文件中也附上了清晰的大图。

同时可以注意到，Q-learning 的更新并非必须使用当前贪心策略$argmax_a Q(s,a)$得到的数据，因为给定任意(s,a,r,s')都可以直接根据更新公式来更新Q。Sarsa 必须使用当前ϵ-贪婪策略采样得到的数据，因为它的更新中用到的Q(s',a')的a'是当前策略在s'下的动作。因此，Sarsa是在线策略，Q-learning是离线策略。

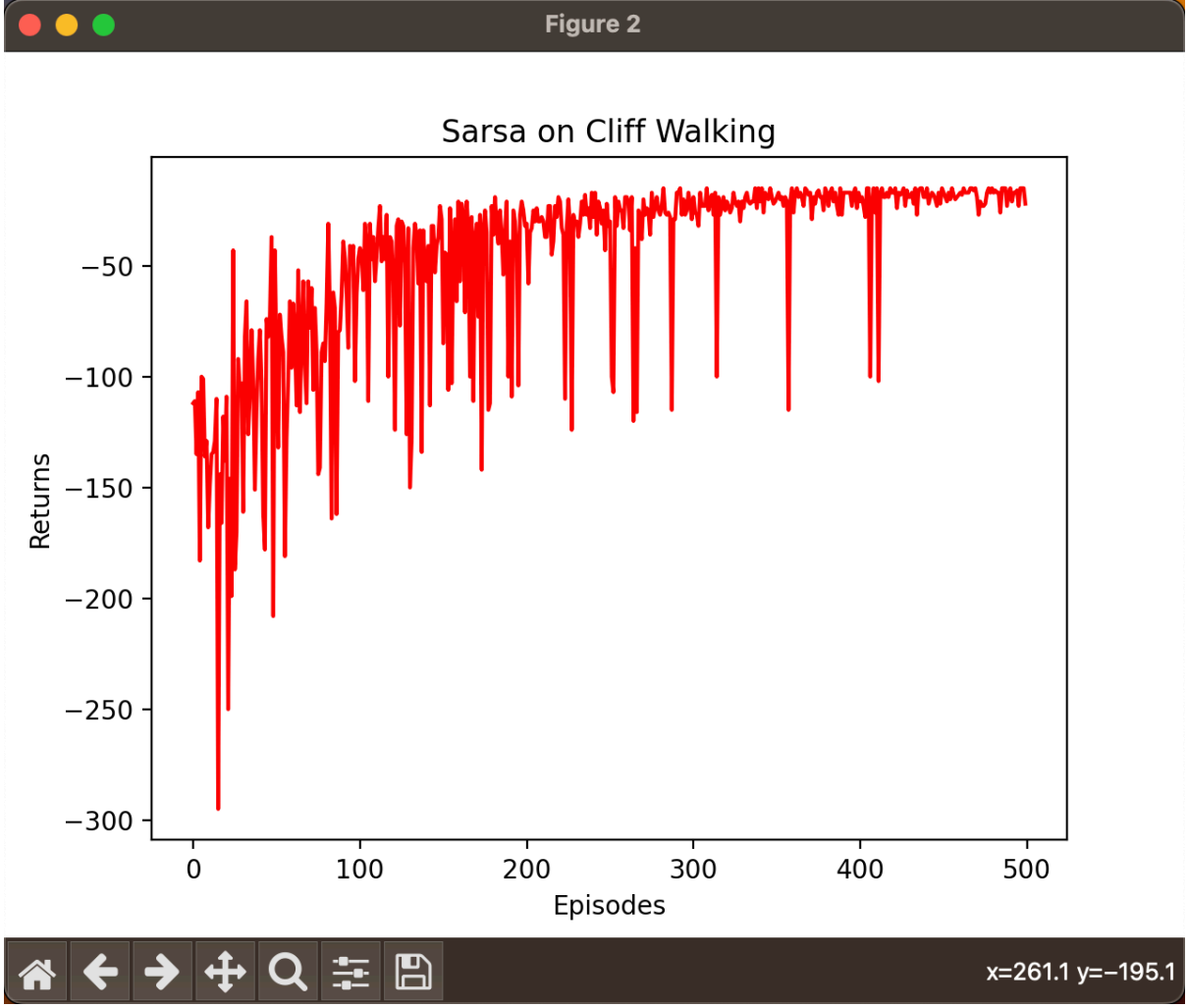


Figure 2
Sarsa on Cliff Walking
Returns
−50
−100
−150
−200
−250
−300
0
100
200
300
400
500
Episodes
x=261.1 y=−195.1

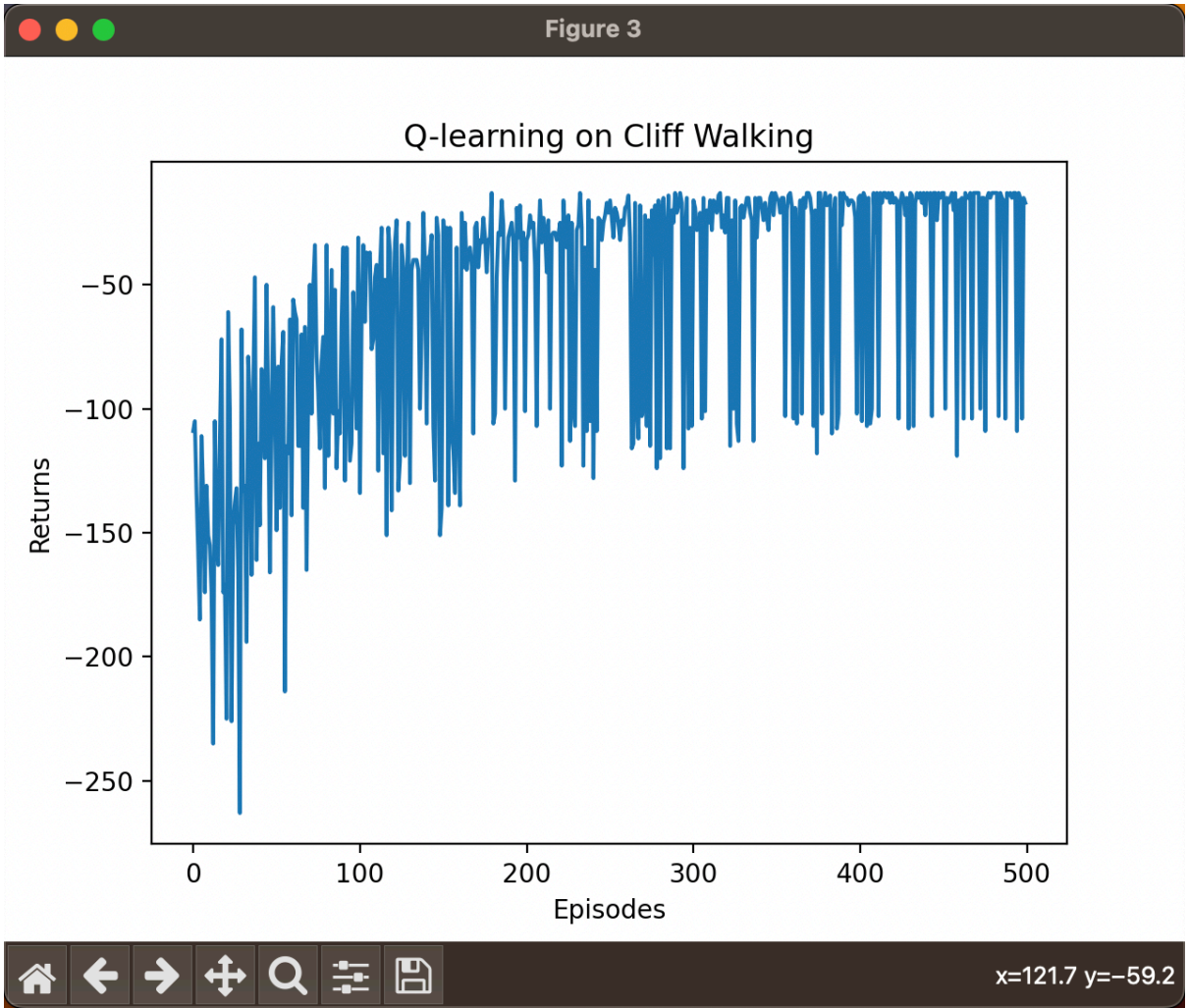


Figure 3
Q-learning on Cliff Walking
Returns
−50
−100
−150
−200
−250
0
100
200
300
400
500
Episodes
x=121.7 y=−59.2

大致讲解一下代码结构，代码里有详细的注释

CliffWalkingEnv主要是环境，输入智能体的动作给出下一个状态，reward和是否到达终态的done

Sarsa类和Q-learning是智能体，主要维护一个nrow*ncol行，n_action列的Q-table

我们初始化学习率alpha=0.1，折扣因子gamma=1，epsilon-贪婪中的epsilon=0.1。每一步选择动作价值最大的动作或随机选择动作。同时定义best action函数用来打印策略。定义update方法实现sarsa和Q-learning的q_table更新。

主函数进行了num_episodes=500轮次训练，每10条序列打印一下这10条序列的平均回报。训练过程中需要注意的是，Sarsa的输入为(s,a,r,s',a')，Q- learning的输入为(s,a,r,s')。因此不能直接套用Sarsa的训练代码用于Q-learning，下面讲到的print_route()函数亦是如此。

Sarsa算法的更新公式为

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_t + \gamma Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)]$$

Q-learning算法的更新公式为：

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[R_t + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)]$$

这是Sara和Q-learning最大的不同

程序中也根据这个公式来更新q_table.

最后定义了两个打印函数print_agent()和print_route()。第一个用于打印智能

体在每个位置会选择的动作（即打印策略），第二个用于打印智能体从开始位置出发会走的路线（即打印路径）。