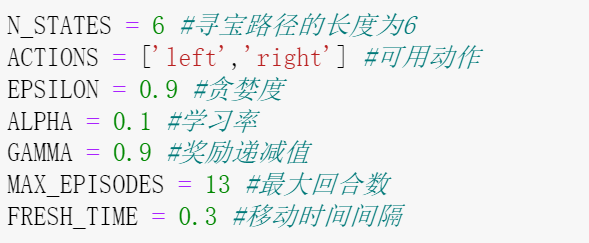
强化学习实验报告

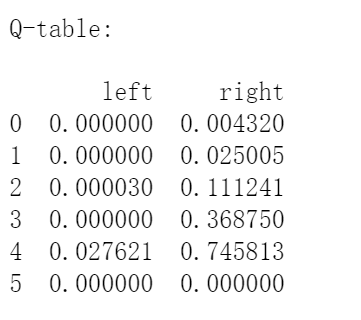
191250111 裴为东

**环境设计：**

首先对学习的环境参数进行定义



然后是q表的形式，q\_table的index对应探索者的位置state，column对应探索者的行为action，值为对应的行为值value



**伪代码描述：**

1.初始化q表

def build\_q\_table(n\_states,actions):

建立一个格式为DataFrame的q\_table，行数为状态数，列数为探索者可选动作数

2.在某个状态state，选择行为

def choose\_action(state,q\_table):

从q-table拿到这个状态state所有的行为值value

If 随机数大于贪婪度epsilon 或者 这个state还未探索过：

return 随机选择一个action到达的state

else：

return 选择行为值value更大的action执行到达的state

3.环境反馈

def get\_env\_feedback(S,A):

if A为向右行动：

if S为终点state的前一个state：

S\_为终点state

奖励值R为1

else：

S\_为S向右移动一位的state

奖励值R为0

else：

R为0

If S为起点state：

S\_仍为起点state

else：

S\_为S向左移动一位的state

return S的下一个状态S\_，奖励值R

4.强化学习过程

def rl():

用build\_q\_table函数初始化一个q-table

for 循环MAX\_EPISODES次：

设定行动次数值step\_counter为0

设定每回合初始位置S为0

设定是否到达终点的状态值为is\_terminated为False

用update\_env函数更新环境

while not 到达重点：

行为值A为调用choose\_action函数返回的行为

下一个状态S\_,奖励值R为调用get\_env\_feedback函数返回的值

估算值q\_predict为q-table中对应位置的值

If S\_不为终点：

实际值q\_target=奖励值R + 奖励递减值GAMMA \* q-table对应状态的值

else：

q\_target为R

is\_terminated为True

更新q-table中的值，更新方式加上 学习率ALPHA \*（实际值-估算值）

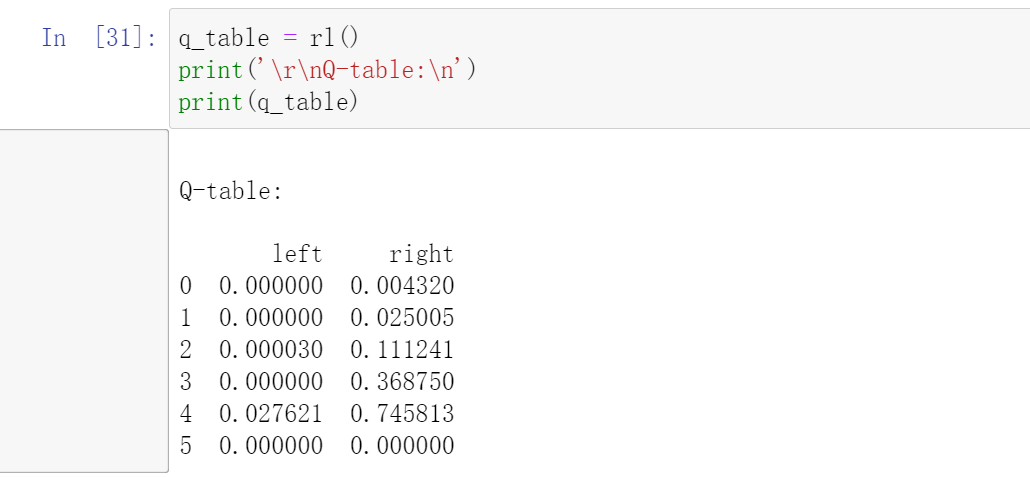
S赋值为S\_

调用update\_env函数更新环境

行动次数+1

return q-table

**实验结果：**



可以看见，向右的奖励值在逐渐增大