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

```
N_STATES = 6 #寻宝路径的长度为6
ACTIONS = ['left','right'] #可用动作
EPSILON = 0.9 #贪婪度
ALPHA = 0.1 #学习率
GAMMA = 0.9 #奖励递减值
MAX_EPISODES = 13 #最大回合数
FRESH_TIME = 0.3 #移动时间间隔
```

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

```
Q-table:

        left     right
0   0.000000  0.004320
1   0.000000  0.025005
2   0.000030  0.111241
3   0.000000  0.368750
4   0.027621  0.745813
5   0.000000  0.000000
```

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：

In [31]:
```
q_table = rl()
print('\r\nQ-table:\n')
print(q_table)
```

```
Q-table:

       left     right
0  0.000000  0.004320
1  0.000000  0.025005
2  0.000030  0.111241
3  0.000000  0.368750
4  0.027621  0.745813
5  0.000000  0.000000
```

可以看见，向右的奖励值在逐渐增大