

Homework

Q-learning实验

Name: Li Shibo Student ID: 119033910046 Email: ShiboLi@sjtu.edu.cn

1 问题说明

问题内容 用强化学习解决easy21问题。

easy21问题规则 有红和黑两种牌，都是从1到10，概率分别是1/3和2/3。

1. 刚开始的时候玩家和交易商都随机拿一张黑牌。
2. 然后玩家先开始决定要牌或不要牌，要牌的时候如果是红牌就减掉相应的数额，如果是黑牌就加相应的数额；如果最终的和超过21或小于1，就爆掉了，如果爆掉了就输了。
3. 如果玩家选择不要牌的时候，交易商就选择是否要牌。如果交易商的牌的数量大于或等于16的时候就不要牌了，否则一直要牌。如果交易商爆掉了，玩家就赢了；否则，结果就看谁的牌的和最大，如果玩家大，玩家就赢了；如果玩家小，玩家就输了；如果一样大，就算平局。

2 数据说明

实验中数据都是自动生成的。初始时，玩家和交易商分别随机生成一张底牌，之后无论玩家要牌还是交易商要牌都是由放回的，也就是说在游戏过程中，玩家或交易商获得每一种颜色每一张牌的概率都是相等的。

3 实验思路

建立模型 因为状态空间有限，总共为 21×10 个，因此可以用q-learning来解决该问题。可以使用动态规划方法来更新q矩阵。每个状态由两个动作，停牌或要牌。因此可以用 $21 \times 10 \times 2$ 的q矩阵来解决该问题。

代码见`Qlearning.py`。本次实验分别由几个函数构成。

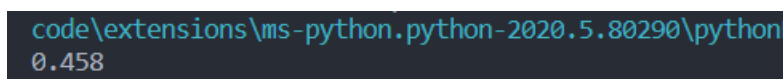
1. `initial`函数用来初始化q矩阵,初始化策略是当玩家牌小于交易商底牌时，玩家更倾向于要牌；当玩家总点数离11很近时，玩家更倾向于要牌，否则，更倾向于停牌。
2. `egreedy`函数用来实现 $\epsilon - greedy$ 探索策略的，每次以概率 $\epsilon/2 + 1 - \epsilon$ 选择给定状态下最优的动作，以 $\epsilon/2$ 的概率选择另一个动作；当给 ϵ 赋值0时代表每次获取最优动作，赋值为1时代表随机获取一个动作，用于后面和强化学习结果做对比。

3. $dealer_card$ 函数表示在玩家停牌后，交易商选择是否要牌。返回交易商最后获得的总点数。游戏规则规定交易商在点数大于等于16的时候停牌。
4. $train$ 函数是对玩家进行训练，得到q矩阵。每局玩家输则获得奖励为-1，玩家赢则获得奖励为1，不输不赢获得奖励为0。
5. $test$ 函数是对玩家学习到的q矩阵进行测试。本实验中我进行了10000次测试，每次玩家的决定都是从学习到的q矩阵中得到最优动作。
6. 剩下的函数都是画图函数，分别画平均累计奖励，q矩阵的最优值和q矩阵的最优动作。

训练模型 训练过程中分别设 $\epsilon-greedy$ 探索策略中的 ϵ 为0.2和0.1发现对胜率影响不大。设置学习率 α 为0.1, 0.8, 0.001, 0.0001发现对胜率和收敛结果影响也不是很大，只是收敛的速度有一点点影响，而且在这个实验中q矩阵很快就达到收敛，最终选择学习率为0.001。实验过程中为使未来的决策对当前影响打一些， γ 一直取0.99。在使用初始化函数之后，得到平均0.45的胜率。

实验环境和数据 实验中仅用了numpy数组，和画图的库，其余均为自己实现。

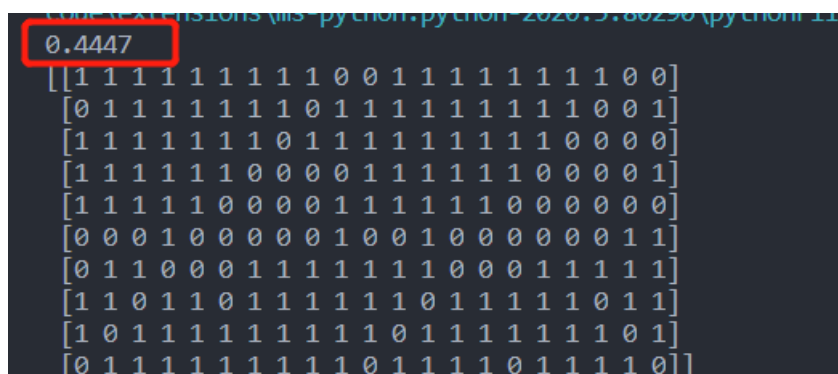
实验结果 实验过程中我分别试了不同的参数。其中是否对q矩阵初始化(是否初始化指的是是初始化为0，还是一定的对玩家有利的经验知识)对实验结果影响比较大，其余参数影响都比较小。当对q矩阵初始化之后的到的准确率在0.45左右。



```
code\extensions\ms-python.python-2020.5.80290\python
0.458
```

图 1: 对q矩阵初始化后所得到的测试准确率

而不初始化，准确率在0.44左右，相比于初始化，稍微低一些。



```
code\extensions\ms-python.python-2020.5.80290\python
0.4447
[[1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 0]
 [0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1]
 [1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0]
 [1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 1]
 [1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 1]
 [0 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1]
 [1 1 0 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1]
 [1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1]
 [0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0]]
```

图 2: 对q矩阵未初始化所得到的测试准确率，下面为q矩阵每个状态所对应的最优动作

对于初始化和不初始化所得到的q矩阵的奖励最优值和最优动作分别如下图所示。

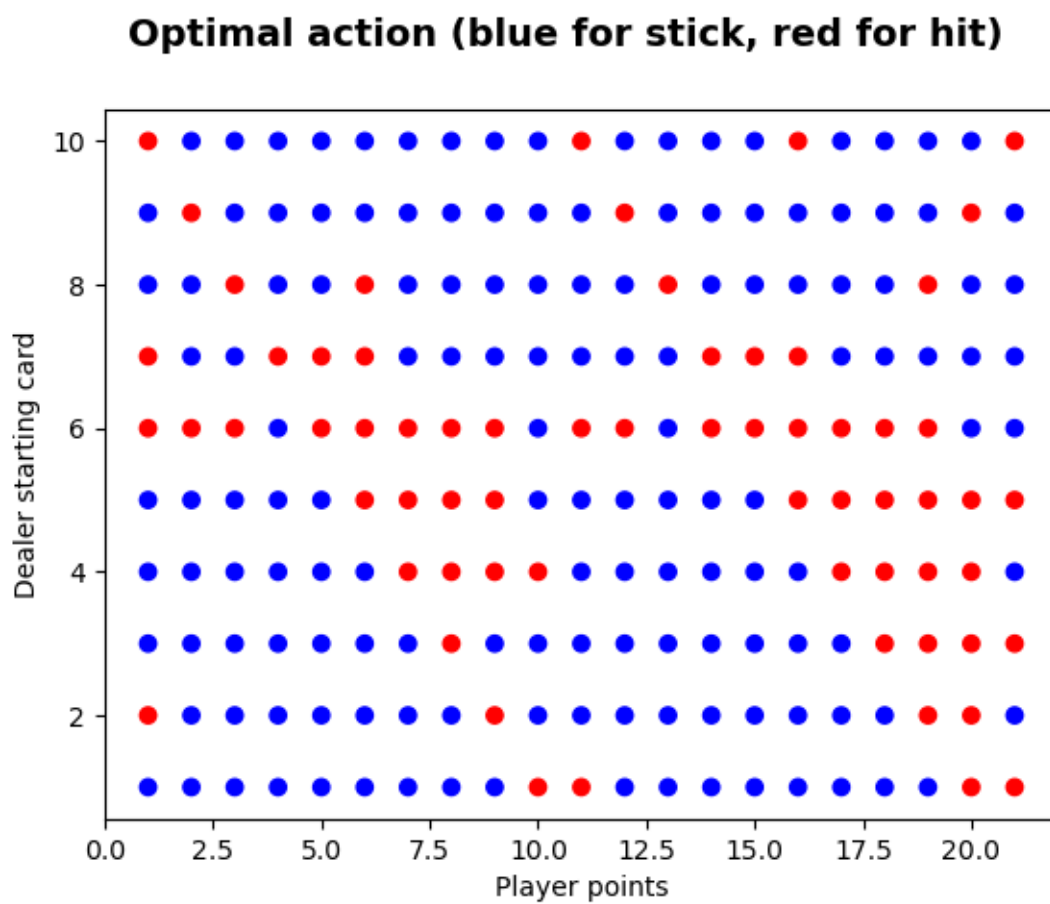


图 3: 对q矩阵未初始化所得到的每个状态所对应的最优动作

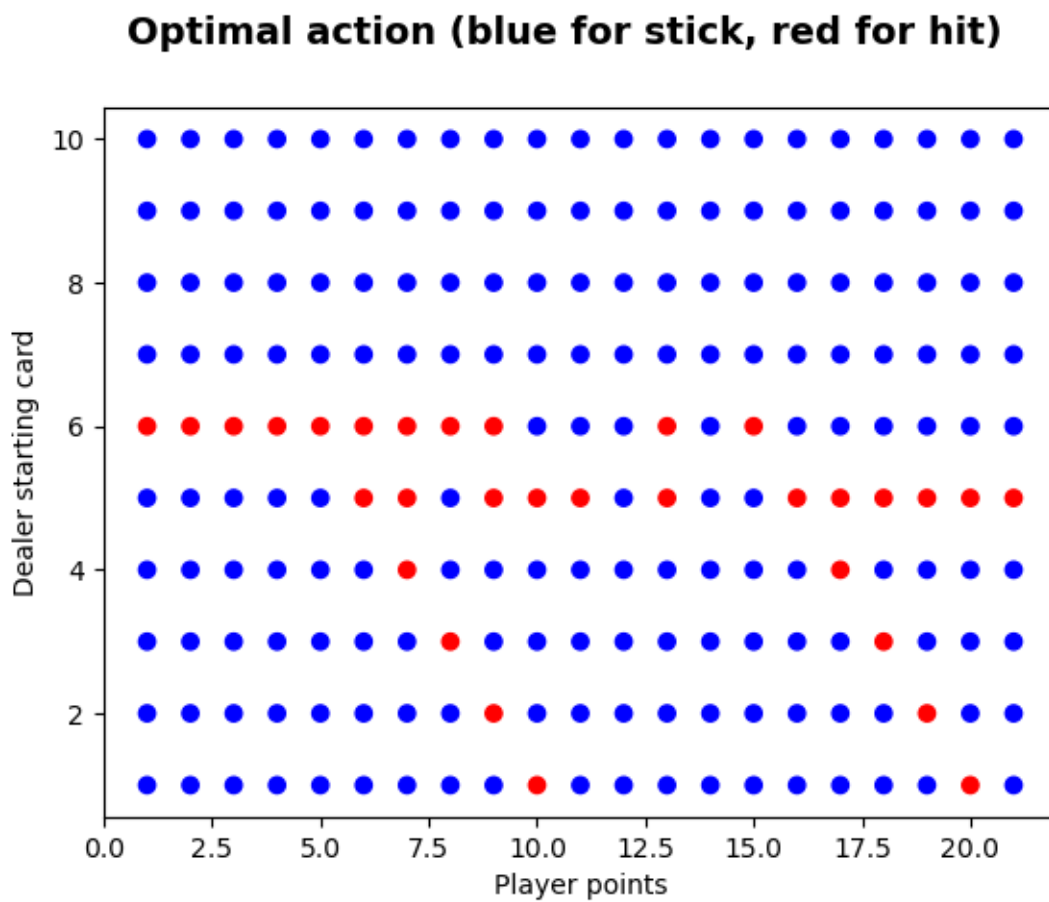


图 4: 对q矩阵初始化后所得到的每个状态所对应的最优动作

从图3和图4可以看出未初始化和初始化，玩家都趋向于停牌。但是相比于未初始化所获得的最有动作，初始化后，玩家更趋向于停牌，决策也更合理化，但是还是有些区域决策不是很合理，比如最右侧都快爆炸了依然决定要牌。

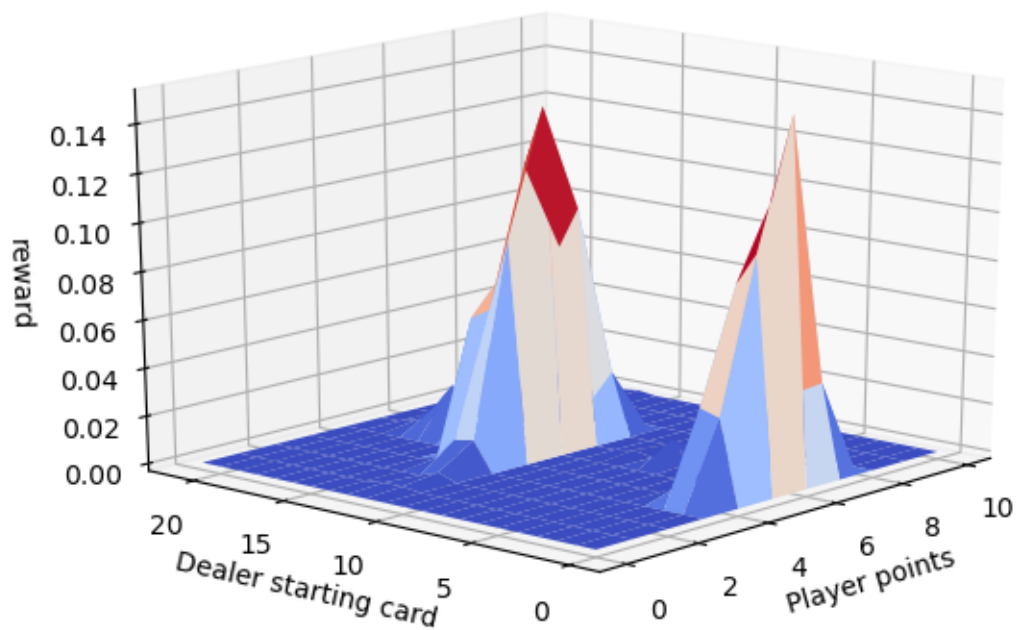


图 5: 对q矩阵未初始化时所得到的每个状态所对应的最优值

从图中可以看出，未根据先验知识初始化时，在两个区域的最优值高一些，也就是说在这两个区域玩家赢得概率大一些。

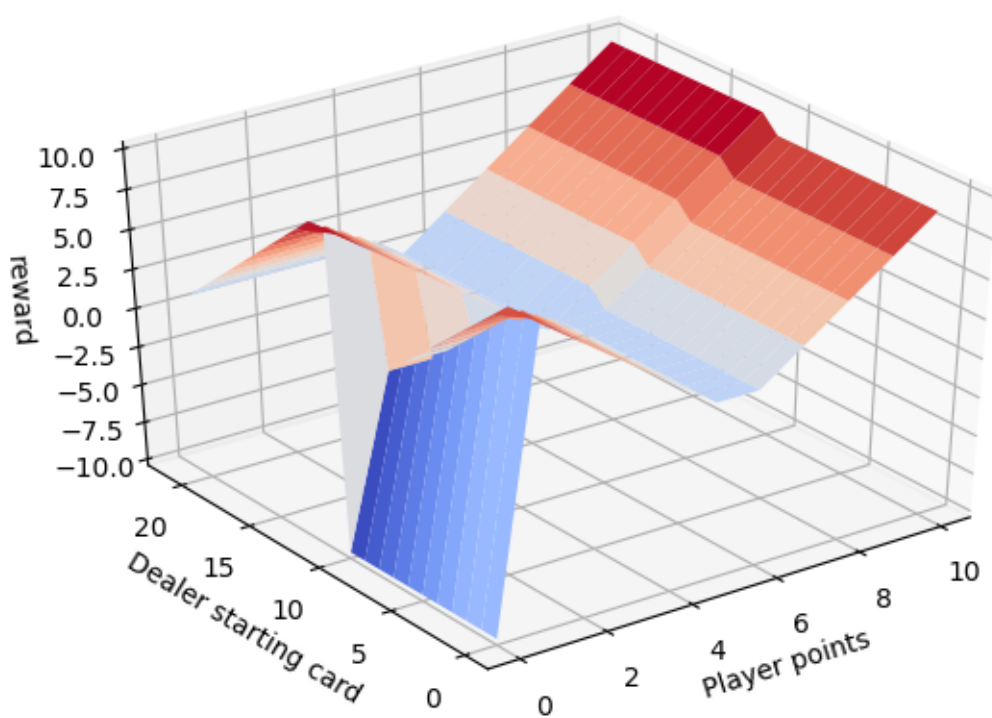


图 6: 对q矩阵初始化后所得到的每个状态所对应的最优值

从图中可以看出，在对q矩阵根据先验知识初始化后，当交易商底牌在5和6附近时玩家获胜得可能性较小，当交易商底牌接近于10或接近于0的时候，玩家获胜的可能性较大。

由于游戏过程是随机的，每次运行刚开始时获得的奖励都可能不相同，因此每次得到的累计奖励图像可能不相同，但最终都收敛到了0.1左右。

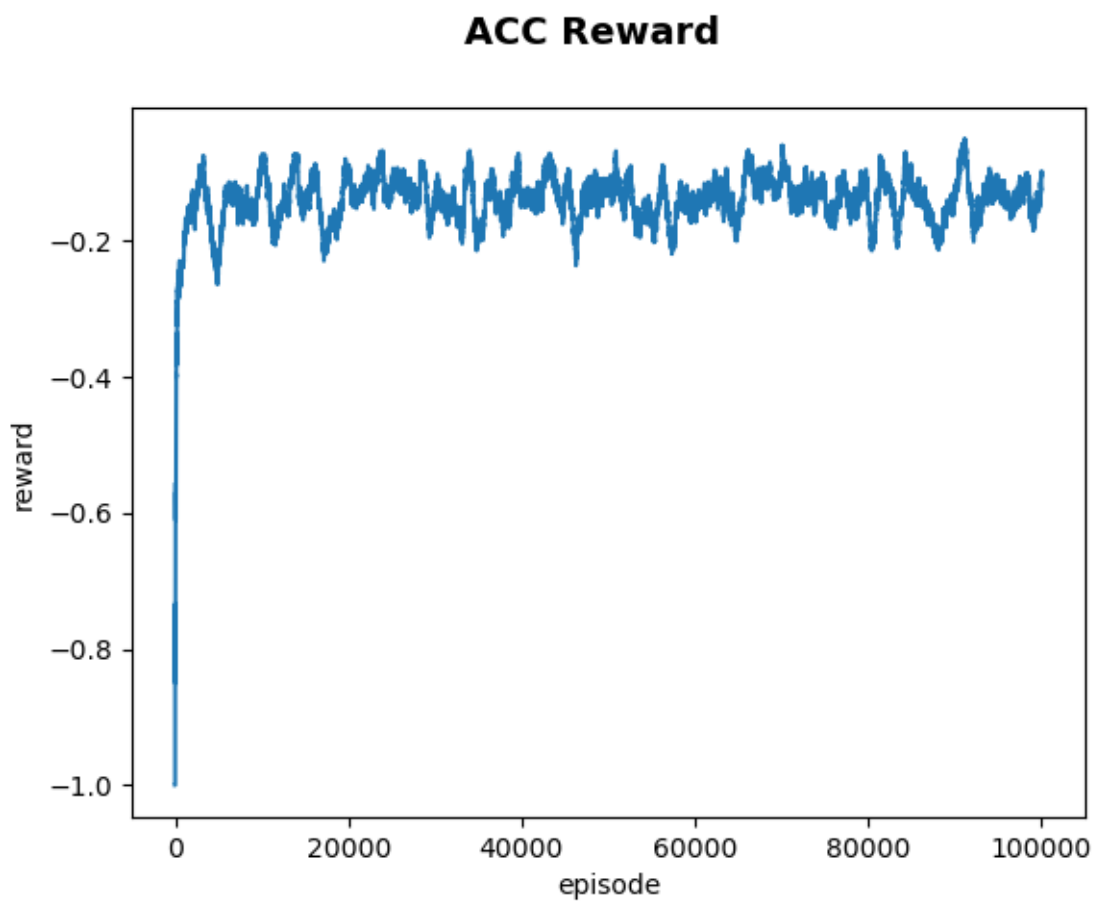


图 7: 实验训练过程中所得到的累计奖励

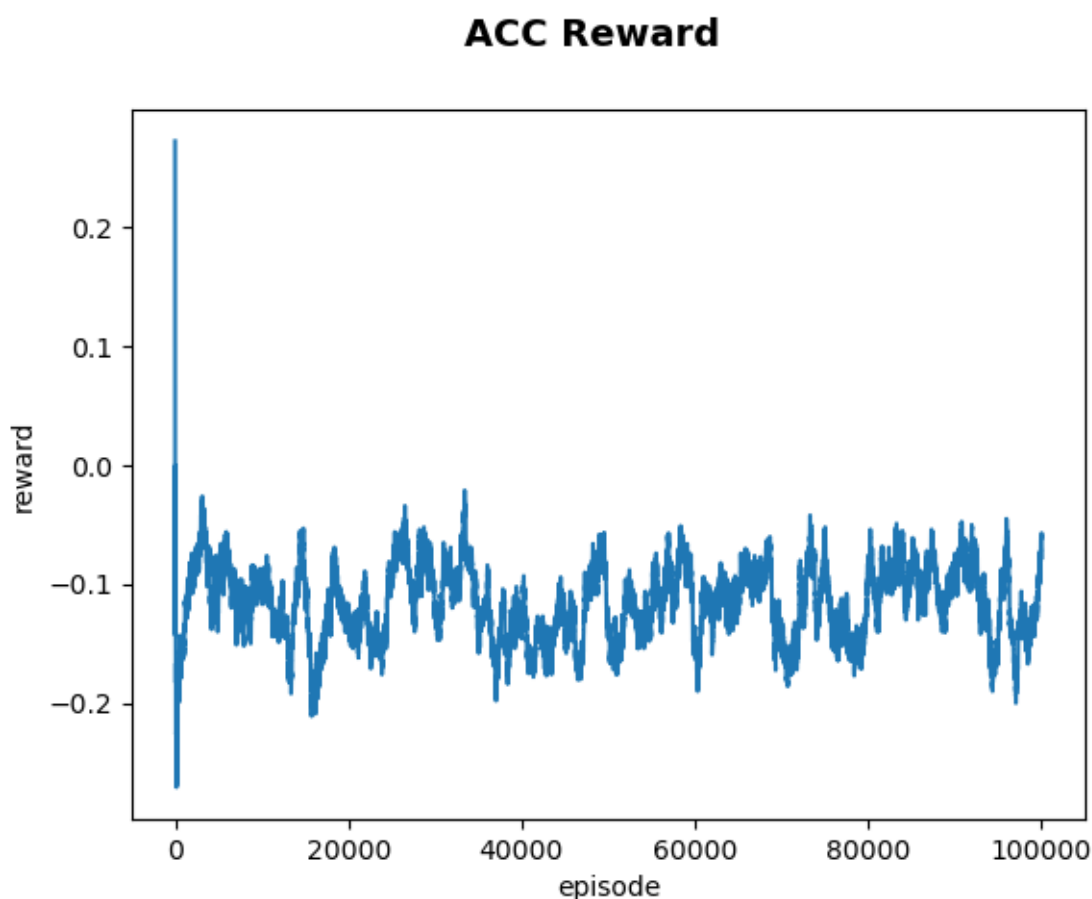


图 8: 每次运行都可能得到不同的累计奖励

上面两个图是两次不同的运行所获得的不同的累计奖励，从图中可以看出，起始位置虽然不相同，但都收敛到了0.1左右。

4 实验结论

1. 由于玩家先做决策然后爆炸之后交易商不用做决定就可以获胜，所以交易商获胜的概率较高，这是个不公平的游戏。在每个状态下用随机选择的方法测试后玩家所得到的胜率约为0.39左右，但使用q-learning之后得到胜率约为0.45左右，说明强化学习使胜率有较大幅度的增长。但即使使用q-learning对决策有所改善，但还是不可能比交易商获胜的概率高。
2. 使用先验知识对q矩阵进行初始化获得的胜率比未使用先验知识对q矩阵进行初始化获得的胜率略高，因此先验知识在强化学习中也比较重要。
3. 从最后强化学习得到的在每个状态下最优动作来看，还是有一些不合理之处，说明实验中用到的强化学习还有需要改进的地方。