SmartCab项目报告

1. 实现基本的驾驶智能体
   Qusetion：在您的报告中，说明观察到的智能体的行为。它最终到达目标位置了吗？
   Answer：
   在代码中加入

```
self.state = (self.next_waypoint, inputs)
action = random.choice(self.env.valid_actions)
```

使agent接受将next_waypoint, inputs作为当前状态state。
选取的行为从valid_actions随机选取 （即前进、左转、右转）。
经观察由于每次行为是随机的，无法对小车的行为进行预估。但是误打误撞还是能到达重点的。此时没有设置Qtable以及Reward，因此在每一步中也无法进行更新。
————————————————————————
2. 确定和更新状态
Qusetion：对您选取上述状态组的原因，以及这些状态组对模拟智能体及其环境的方式进行说明。
Answer：上题中已经说明，选择nextwaypoint, inputs作为状态组。因为上述变量可以唯一并且精确地确定一个cab所处的状态。其中包括当前状态（红绿灯，是否有来车等），在一开始的时候曾经考虑将deadline也加入state中，后来觉得如果加入了deadline变数字典索引就太麻烦了，发现在没考虑deadline情况下训练效率不错。因此最后只将next_waypoint, inputs作为状态组。
————————————————————————
3. 实现 Q 学习

```
self.Qtabel = dict()
self.discount = 0.3
self.gamma = 0.6

self.lastState = 0
self.lasAction = 0
self.lastReward = 0
```

在reset：

```
self.lastState = 0
self.lasAction = 0
self.lastReward = 0
```

只重设这三个变量。Qtable不需要重设。
在agent内定义一个Qtable作为Q学习的矩阵，将state和action生成的tuple作为键值。每次update函数执行时，重复以下过程。
获取当前状态，取得上一个状态的数据

`lastState,lastReward,lastAction`。
按照公式:

```
Q(s,a) = (1-α) * Q(s,a) + α*(R + β* max (Q(s + 1, a')))
```

对Qtable进行更新。由于Q(s+1, a')只有当执行action后才能得知，因此采取的策略是存储上一状态的数据，在下一个update更新。
发现智能体的行为具有了策略性，也就是在Qtable的指导下更加的“智能“。在pygame中可以看出来：在开始的几轮中，由于Qtable为空，小车采取随机的行为。因此开始的trial中是对小车进行训练，小车不一定能达到目的地。随着Qtable的更新，小车行为越来越有目的性。
但是经过观察。这种学习行为有一些缺陷，即如果小车在某个状态发现一个Qtable为正值，其他为0.小车根据算法会选择为正值的方向，如果在此方向上R为正值便会得到增强。小车更加容易走之前加强过的地方，对没走过的地方便不去尝试。在后期会发现小车基本沿着相同的路线在走。但是在大部分情况下可以在deadline之前到达目的地。

————————————————————————————

4. 改善驾驶智能体的表现
Question::报告您为了获得智能体的最终版本而对 Q 学习的基础实现所做的更改。它的表现如何？
智能体是否快接近找到最优策略，即在尽可能短的时间内到达目的地，同时未受到任何惩罚？

在对小车参数进行更改时，考虑了以下因素：
alpha:学习速率。由于公式中
Q(s,a) = (1- α) * Q(s,a) + α *(R* + β max (Q(s + 1, a')))
alpha越大代表新状态代表的比值越高，当alpha为1的时候完全忽略之前的状态，alpha为0的时候相当于没有进行学习。
视频上说可以讲alpha设置为1/t，但是经过实践发现效果并不是特别的理想。现在将alpha设置为常数0.3.即新的状态还是占了较大的比重，因为考虑到destination会改变，新的状态要比之前的重要。
折扣系数gamma:
具体体现在新状态占的比重中，参考论坛上老师的建议将gamma设置为0.7.
action选择方案：
根据算法，action应该选择当前状态使qtable达到最大的action。当具有多个最大值时（如刚开始Qtable初始化时），从多个x中随机选取argmax(x).
通过改变参数，经观察发现
当alpha = 0.3 gamma = 0.7时，100次训练的成功率为95%左右。但是成功率会出现不稳定的情况，即有时特别高，有时掉到70%以下。
通过改变alpha的值不能改变这种情况，我很困惑如何解决这种情况?