

Report

实现基本的驾驶智能体

问题: 在你的报告中, 说明观察到的智能体的行为。它最终到达目标位置了吗? 有什么其他值得注意的地方吗?

Answer:

由于智能体的行为是随机的, 运气好的话, 它是可以到达目标位置的, 但是到达目标位置时一般都超过了deadline。如果deadline = -100时智能体还没有到达目标位置的话, 就会尝试失败。

Inform the Driving Agent

QUESTION: What states have you identified that are appropriate for modeling the smartcab and environment? Why do you believe each of these states to be appropriate for this problem?

Answer:

1. next_waypoint (下一路径点的位置)
2. light (信号灯的状态)

next_waypoint是根据目标位置得到的, 用于指示前进方向。

light是为了不违反交通规则, 不发生撞车事故。

至于input中的 三个路口车的方向, 经过测试, 如果添加, Q表格将会变得比较大, 训练100次也不能保证reward为正, 因此只使用了next_waypoint和light。

OPTIONAL: How many states in total exist for the smartcab in this environment? Does this number seem reasonable given that the goal of Q-Learning is to learn and make informed decisions about each state? Why or why not?

Answer: 在这个环境中一共有6个状态, 对于Q学习来说是合理的。如果我们将所有的状态都加进来, 那么状态的个数将达到 $3 * 2 * 4 * 4 * 4 = 384$ 种, 智能体

大概率无法在100次的有限条件下训练出完整的Q，也就是说大部分可能犯规的状态的Q仍然为0。因此为直接将状态缩小至6，虽然可能会慢一点，但是不会违反交通规则。

Implement a Q-Learning Driving Agent

QUESTION: What changes do you notice in the agent's behavior when compared to the basic driving agent when random actions were always taken? Why is this behavior occurring?

Answer:智能体完美地遵循了最短路线，并且没有任何失误情况（reward < 0）。因为Q学习算法学习到了交通规则，以及路径规划。

Improve the Q-Learning Driving Agent

QUESTION: Report the different values for the parameters tuned in your basic implementation of Q-Learning. For which set of parameters does the agent perform best? How well does the final driving agent perform?

Answer:

alpha

只要不把alpha调到大于1，或者减小到0，学习效果基本没有变化，都是完美表现。因此为设置为0.3。

gamma

情况和alpha一样，只要在(0,1)就没有问题。因此设置为0.1。

epsilon

这个参数为到策略是这样的，在80次以内时设定为0.5，80次以后设定为0，表现就是前80次，即使Q已经学习得足够好了，也一直有几率出现reward<0，后20次就时完美表现。设置0.5是因为希望它能探索更多的情况。

QUESTION: Does your agent get close to finding an optimal policy, i.e. reach the destination in the minimum possible time, and not incur any penalties? How would you describe an optimal policy for this problem?

Answer: 在程序中的智能体依据优化，已经可以以最短路径到达目的地，并且能够保证不会得到任何处罚。我认为它的策略可以这样简单表示：

```
if light == 'green':  
    self.action = self.next_waypoint  
else:  
    self.action = None
```