任务

任务

项目报告

除了你修改过的 `agent.py` 的文件，你还需要提交一个项目报告。报告中需要详尽全面的回答下面斜体部分的问题。

实现基本的驾驶智能体

第一步，你的任务是让这个智能出租车在环境中动起来。此时，你不需要考虑任何行驶优化问题。注意这个智能车在每个路口都会或者下列信息。

- 下一路径点位置（相对于当前位置和前进方向）；
- 十字路口状态（红绿灯和是否有车）；
- 当前最后期限值（剩余的时间步）；

要完成这个任务，要你的智能车在每个路口从下面这几个可能的动作中随机选择一个。 (`None`, `forward`, `left`, `right`) 忽略上面提到的输入信息。设置模拟环境中运行截止时间，即 `enforce_deadline` 设置为 `False`，并观察智能车的驾驶行为。

问题： 在你的报告中观察并记录智能车在采取随机动作时的行为。它最终到达目标位置了吗？还有什么其他有趣的现象值得记录下来？

训练智能车

现在你的智能车已经可以在环境中移动，你接下来的任务是确定一组合适的状态对智能出租车与它的环境建模。状态变量的主要来源是当前路口的当前输入（`inputs`），但并非所有都需要被表示出来。你可以显式地定义状态，或者用一些输入的组合作为一个隐式的状态。在每一时间节点中，你需要处理输入，用`self.state`更新智能车的当前状态。依然把 `enforce_deadline` 设置为 `False` 观察你的智能车在随着模拟的进行，如何汇报状态的改变。

问题： 你认为哪些状态适合对智能出租车和环境建模？为什么你认为在这个题目中这些状态是合适的？

可选题目： 在这个环境中，智能出租车总共可能有多少状态？这个数字足够让我们的智能车做 *Q-Learning*，使得它在每个状态可以做出基于训练的决策吗？如果是，为什么？如果不是，也说一下原因。

实现 Q-Learning 智能车

当你的智能车能够理解输入信息，并且有一个环境状态的映射。你下一步的任务是为智能车实现 Q-Learning 算法，使得它可以在每一个时间节点，基于当前状态和动作的Q值做最佳动作选择。智能出租车所做的每个动作，都会产生一个基于环境状态的奖励。Q-Learning 智能车需要在更新Q值時，考虑这些奖励。实现完成，把模拟环境中的 `enforce_deadline` 设置为 `True`。运行模拟器，观察环境中的智能出租车在每一轮测试中的移动情况。

能出租车在每一轮测试中的移动情况。

更新 Q-values 的公式可以在这个视频 任务 (https://classroom.udacity.com/nanodegrees/nd009/parts/0091345409/modules/e64f9a65-fdb5-4e60-81a9-72813beebb7e/lessons/5446820041/concepts/6348990570923)中找到。

问题： 与一直选择随机动作相比，你发现智能车的行为有了什么样的变化？为什么会发生这种变化？

提高 Q-Learning 智能车

在这个项目中，你最终的任务是强化你的智能车，让它在经过足够的训练之后，能够在规定的时间内安全高效的抵达目的地。Q-Learning 算法中的参数，例如学习率（`alpha`），折扣因子（`gamma`）以及探索率（`epsilon`）这些都对智能车能否在每个状态学习出最佳动作选择有影响。调整他们，要提高你智能出租车的成功率：

- 把测试次数，`n_trials`，在模拟器中设置成100。
- 在 `enforce_deadline` 设置为 `True` 的情况下运行模拟器。（你还需要减小更新延迟 `update_delay` 并且把 `display` 参数设置为 `False`）。
- 观察智能车的学习过程和成功率，特别是在后期的测试中的表现。
- 调整上述参数中的一个或多个参数，迭代这个过程。

完成这个任务，你需要找到你认为对智能车能够成功学习的最佳参数组合。

问题： 把你实现基本*Q-Learning*时的参数调节过程记下来。哪个参数组合智能车表现最好？它最终的表现有多好？

问题： 你觉得你的智能车已经几乎找到了最佳策略吗？例如，能够在最短时间内到达目的地，不遇到任何惩罚。在这个问题中，你觉得应该怎样定义最佳策略？