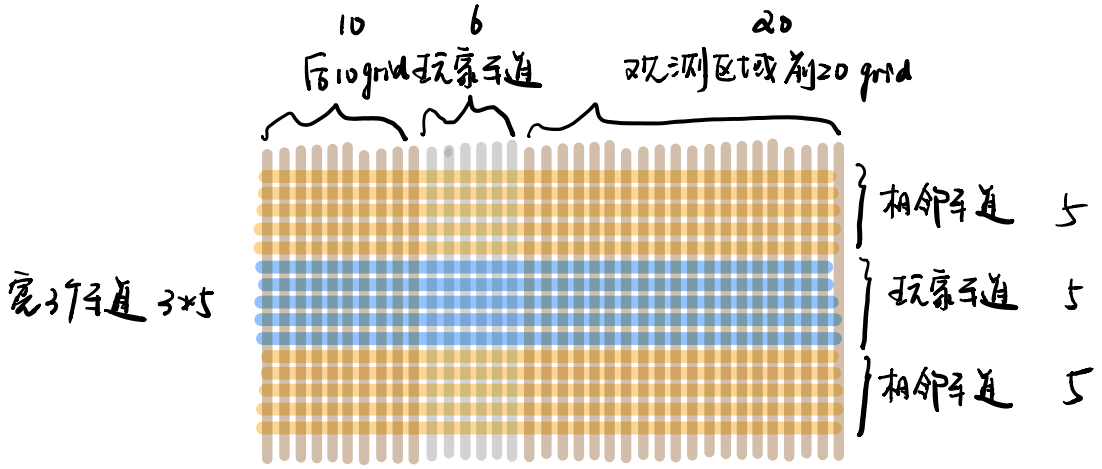
Deep traffic 项目平台.

1. 局部可观状态

栅格化表示

车辆：3*6 grid { 玩家车： 2 2 2 2 2 / 2 2 2 2 2 / 2 2 2 2 2 ； 障碍车： 1 1 1 1 1 / 1 1 1 1 1 / 1 1 1 1 1 ； 其他区域 0 表示

每个车道宽 5 个 grid，车辆部分可观区域



Oi 区域越大，检测效果越好但计算量越大

2. 网络输入

O_1 O_2 O_3 O_4 过去4个观测窗口

帧数 车道 栅格 前 后

4 * 3 * 5 * (20 + 10) 1800 Pixels

车辆入状态为 1800

3、Action

5个动作空间.

0	1	2	3	4
上改道	保持	下改道	加速	减速

4. reward

目标：玩家车辆不断超车.

加速　超车 { 变道 / 减速 }　不同情况下 决策的结果不同

Action	Reward
左改道	0
右改道	0
加速	0.1 * (Current_velocity - Min_velocity)
减速	0.05 * (Current_velocity - Min_velocity)
保持	0.1
Done 撞车	-3

5、实验

1) DQN

使用 Experience Replay 和 Double DQN 技巧

以像素为输入，似类于"游戏"DQN 操作

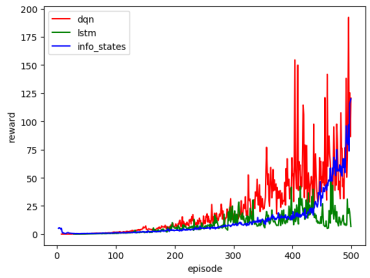
实际中获取车辆的位置信息为输入，因此作了改动.

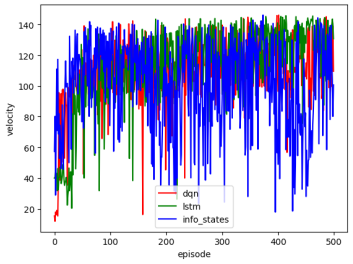
2) LSTM

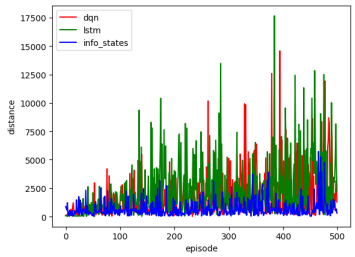
连续动作，自然想到了用 LSTM 网络处理.

3) 输入 相邻车辆的位置和速度. DQN

Input 维度从 1800 下降 帧数 * (位置 + 速度) * 7 = 112







记录 3 种方法 Reward, Velocity, distance 更改数据.

① 3 种方法都可以有效控制 Play Car 避障运行

② dqn 方法 得分较高, 运得距离并不长, 以加速回馈得分

③ lstm 能让车辆以较为平稳的较长时间运行.

④ 输入状态改变 (维度 1800 → 112), 也能得到较好的控制策略.

6、 Multi-agent:

Deep traffic 中 20 agent 控制其中 1 个

DQN 算法 Copy 给其 车辆上 没有协同控制

使用 MARL 方法 协同控制车辆, 提速.