基于强化学习和注意力机制的车辆换道研究

沈炼成，林镇阳，韩立君

***Abstract*— 这里是华丽丽的摘要**

I. 简介

II. 任务描述与分析

A. 任务描述

整体任务为根据输入的车辆周围情况，借助深度强化学习算法得到高层的指令规划。再借助仿真器内部的运动规划器，将高层指令转化为具体的轨迹让底层控制器有效跟踪。最终实现汽车的超车换道。评价标准如下:

- 尽可能少碰撞
- 尽量保持车道线内平稳行驶
- 速度越快越好
- 有前车阻挡能尝试安全换道

B. 仿真环境描述

整体实验基于 *highway_env* 开发，具有较强的灵活性。下面分别针对状态空间、动作空间等进行叙述。

1) 状态空间: 在环境中，状态空间可以选择底层的低维向量输入，也可以选择高维的图像输入和占用格作为输入。下面重点叙述使用低维向量输入和图片输入的基本情况。

当使用低维输入时，传入 15 辆车的坐标、速度、倾斜角度等信息表示出来，包括 $x, y, vx, vy, cos_h, sin_h$。传入一个大小为 $[15, 7]$ 的数组。方便后面网络进行处理，其中第一行表示的是本车的的各种信息。

使用图片输入时，传入大小为 $[600, 150]$ 的图片，同时将图片转化为灰度图。由于一张图片难以获得车辆的完整信息（如速度、运动方向等难以推测），因此将过去 4 帧叠放在一起作为状态观测输入，便于从观测推出状态。

后续实验证明，向量输入的归一化以及将两者信息整合同时输入可以显著提升换道效果。

2) 动作空间: 仿真环境可以将动作设置为连续的油门和方向盘角度，但连续空间训练收敛难度会增大。考虑到算力限制，因此使用离散动作进行控制，再将离散动作送入规划器中生成连续动作。在仿真器中，可以分为 5 个离散动作——左转、保持车道、右转、加速、减速。分别用动作 a_0 到 a_4 表征。这五个动作共同构成了仿真环境的动作空间，下层规划器根据决策指令对车辆动作进行规划控制。

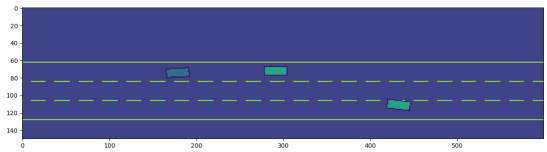

Fig. 1. 图像输入

3) 奖励设置: 环境中奖励设置较为灵活，主要分为三个部分，碰撞惩罚、车速奖励、靠右奖励。系统默认碰撞惩罚为-1，无碰撞为 0。车速奖励为 $r_v = 0.4\frac{(v_t - v_{min})}{(v_{max} - v_{min})}$ 其中，$v_{max} = 30m/s, v_{min} = 20m/s$。此外还有靠右奖励，奖励系数为 0.1。后面我们会讨论调整奖励系数对车辆的影响，实验发现，删除靠右奖励会导致车辆换道次数显著增加。

C. 任务分析

根据任务需要我们设计强化学习算法，将低维或高维输入转化为离散的汽车换道决策指令。即首先需要对观测量进行提取推测，得到状态量。再将状态量作为输入得到动作。而近来深度强化学习的兴起让这两个环节可以直接联合训练，提升了学习效率。

在学习算法方面，我们采用了 Dueling DDQN 技术加入了一些训练技巧，能够有效提升收敛速度和训练稳定性。具体方法我们会在下节进行详细叙述，同时与之前的 Dueling DQN，DQN 进行比较，证明我们方法的有效性。同时针对输入，我们分别探讨了对于向量输入、图片输入和混合输入的效果。此外，针对不同奖励设置进行了消融实验。

III. 强化学习算法

可以将超车换道问题转化为一个马尔可夫问题，由 $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$ 来表示，分别代表了状态空间、动作空

间、转移概率、奖励函数、折扣因子。

根据前一节的分析，状态空间、动作空间、奖励函数均已得到。折扣因子 γ 设为 0.99。不妨设 t 时刻的回报 G_t 为:

$$G_t = \sum_{i=0}^{\infty} \gamma^i r_{t+k}$$

G_t 表示了从 t 时刻到结束的总的奖励，整个强化学习过程的目标即为最大化回报。但转移概率未知，因此采用无模型强化学习的方法，常用的无模型方法为 Q 学习算法。针对某个策略，定义 Q^π 为状态动作对好坏的评价标准，可以借助前面的回报定义为:

$$Q^\pi = \mathbb{E}_\pi[G_t | s_t = s, a_t = a]$$

同时使用经验回放池，使得训练数据满足独立同分布假设，以及充分利用过往探索经验，方便进行深度学习。使用目标网络，防止计算 TD 误差时目标变动导致更新困难。使用 $\epsilon - greedy$ 算法，ϵ 值逐渐衰减，来平衡整个利用与探索的过程。整个 DQN 算法流程如下所示:

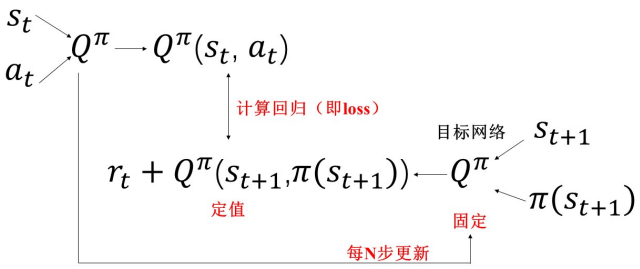

Fig. 2. DQN 算法框架

但是传统的 DQN 算法会存在 Q 值过高估计的问题，因此引入了 Double DQN, 将决策动作生成网络和状态动作对估值网络分开，可以一定程度缓解过估计问题。此外在训练数据中，很多状态并不需要估计每个动作的值，使用 Q 函数可能引入较多噪声。因此将 Q 函数分解为状态函数 V 和优势函数 A 最终 Q 值为两个函数的叠加。此外由于一个 Q 值可以有无数种分解方式，如果直接估计会导致难以收敛。因此对优势函数去均值保证分解的唯一性。最终 Q 值可以表示为:

$$Q(s,a) = V(s) + (A(s,a) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a'))$$

此外我们经过研究发现，获得终止信号有两种情况，撞车或达到仿真时间（50 步）。但这两种情况实际上是不一样的。撞车后，我们不学习如何从撞车后恢复（相当于车辆已死亡），此时设为终止没有问题。但是只是到达了仿真步数，此时车辆其实是还可以正常行驶的，如果直接将该状态存为终止状态，会导致 Q 值的估计震荡不容易收敛。因此我们将原有算法进行了修改，加入了死亡判断。即每次动作得到终止信号时，会判断信号来源于撞车还是到达仿真步数，如果是来源于撞车，则存入经验池的 $done = True$，否则仍设为 $False$。具体算法如算法 1 所示。

算法 1 加入死亡判断的 Dueling DDQN 算法

初始化大小为 N 的经验回放池 D，初始化随机选择概率 ϵ
使用随机参数 θ 初始化动作价值函数 Q
使用参数 θ^- 初始化目标动作价值函数 $\hat{Q}$, 其中 $\theta^- = \theta$
for episode=1,episode<=M,episode++ **do**
    初始化状态序列 $s_1 = x_1$
    for t=1,t<=T,t++ **do**
        以 ϵ 的概率进行随机动作选择 a_t，否则选择 $a_t = argmax_a \hat{Q}(s_t, a; \theta)$
        执行动作 a_t 得到奖励 r_t 和下一步的状态 s_{t+1}，以及是否结束 d_t
        if $d_t == True$ 且发生了撞车 **then**
            $\hat{d}_t = True$
        else
            $\hat{d}_t = False$
        end if
        将数据 $(s_t, a_t, r_t, s_{t+1}, \hat{d}_t)$ 存入经验回放池 D 中
        随机在经验回放池中选取一个 $batch$ 的 $(s_j, a_j, r_j, s_{j+1}, , \hat{d}_j)$
        令 $y_j = \begin{cases} r_j & \hat{d}_j = True \\ r_j + \gamma \max_{a'} \hat{Q}(s_{j+1}, a'; \theta^-) & \text{其余情况} \end{cases}$
        设定损失函数为 $(y_j - Q(s_j, a_j; \theta))^2$，采用随机梯度下降的方法更新参数 θ
        更新 ϵ
        每 C 步令 $\hat{Q} = Q$
    end for
end for

IV. 注意力机制

V. 实验分析

致谢