基于深度强化学习的端到端无人驾驶决策

黄志清[1,3],曲志伟[1,3],张 吉[1,3],张严心[2],田 锐[1,3]
(1. 北京工业大学信息学部,北京 100124; 2. 北京交通大学电子信息工程学院,北京 100044;
3. 北京市物联网软件与系统工程技术研究中心,北京,100124)

摘 要: 端到端的驾驶决策是无人驾驶领域的研究热点. 本文基于 DDPG(Deep Deterministic Policy Gradient) 的深度强化学习算法对连续型动作输出的端到端驾驶决策展开研究. 首先建立基于 DDPG 算法的端到端决策控制模型,模型根据连续获取的感知信息(如车辆转角,车辆速度,道路距离等) 作为输入状态,输出车辆驾驶动作(加速,刹车,转向) 的连续型控制量. 然后在 TORCS(The Open Racing Car Simulator) 平台下不同的行驶环境中进行训练并验证,结果表明该模型可以实现端到端的无人驾驶决策. 最后与离散型动作输出的 DQN(Deep Q-learning Network) 模型进行对比分析,实验结果表明 DDPG 决策模型具有更优越的决策控制效果.

关键词: 无人驾驶; 端到端决策; 深度强化学习; 深度确定性策略梯度算法
中图分类号: TP242.6 **文献标识码:** A **文章编号:** 0372-2112 (2020) 09-1711-09
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2020.09.007

End-to-End Autonomous Driving Decision Based on Deep Reinforcement Learning

HUANG Zhi-qing[1,3],QU Zhi-wei[1,3],ZHANG Ji[1,3],ZHANG Yan-xin[2],TIAN Rui[1,3]
(1. *Faculty of Information Technology,Beijing University of Technology,Beijing* 100124,*China*;
2. *School of Electronic and Information Engineering,Beijing Jiaotong University,Beijing* 100044,*China*;
3. *Beijing Engineering Research Center for IoT Software and Systems,Beijing* 100124,*China*)

Abstract: The end-to-end driving decision making is a research hotspot in the field of autonomous driving. This paper studies the end-to-end driving decision of continuous action output based on DDPG (Deep Deterministic Policy Gradient) deep reinforcement learning algorithm. First,an end-to-end decision-making control model based on DDPG algorithm is established. The model outputs the continuous control quantity of vehicle driving action (acceleration,braking,steering) according to the continuously acquired perception information (such as vehicle angle,vehicle speed,road distance,etc.) as the input state. Then,the model is trained and verified in different driving environments on the platform of TORCS (The Open Racing Car Simulator). The results show that the model can realize the end-to-end decision-making of autonomous driving. At last,it is compared with DQN(Deep Q-Learning Network) model of discrete action output. The experimental results show that DDPG model has better decision control effect.

Key words: autonomous driving; end-to-end decision-making; deep reinforcement learning; DDPG

1 引言

传统的无人驾驶系统主要基于规则进行决策[1,2]. 但在面对复杂多变的交通场景时,规则构建复杂,且难以覆盖可能出现的场景. 随着人工智能技术在无人驾驶领域的应用,将复杂的场景理解与决策均由神经网络来执行,不需要人为地制定规则,形成一种端到端的决策控制模型,即通过获取车辆及行驶环境相关信息(如车辆转角、速度,道路距离,环境图像等) 经过神经网络的处理之后直接输出车辆控制信号. 端到端决策系统简单且性能良好,受到学术界和工业界的广泛关注.

端到端无人驾驶决策目前主要采用深度学习和强化学习两种研究方法. 在深度学习方法上,文献[3] 最

收稿日期: 2018-08-18; 修回日期: 2020-05-20; 责任编辑: 梅志强
基金项目: 国家自然科学基金(No. 61502018)

早提出端到端驾驶模型,利用 3 层全连接网络通过输入图像对车辆进行跟车控制. 文献 [4] 使用 6 层卷积网络,通过双目相机 RGB 图使机器人小车在野外场景完成驾驶决策. 文献 [5,6] 通过卷积神经网络建立端到端控制模型 Pilotnet,输入车辆左前方、正前方与右前方三个方向的图像对车辆转向进行控制. 文献 [7] 利用长短期记忆(Long Short-Term Memory,LSTM)神经网络提高了跟车模型的精度. 文献 [8] 在大型视频数据集上使用全卷积-长短期记忆网络预测车辆控制决策. 文献 [9] 基于 LSTM 神经网络建立模型,同时支持跟车与变道两种驾驶行为的决策. 深度学习的端到端方法需要大量的标注数据以保证其泛化性,所以深度学习端到端的无人驾驶模型目前仍然难以训练.

在强化学习方法上,文献 [10] 通过 Q-Learning 算法,在 TORCS [11] 平台上完成超车动作. 文献 [12] 通过加入示教数据改进 Q-Learning 算法,在 TORCS 平台上使得训练时间减少 72%,稳定性提升 32%. 文献 [13] 通过深度 Q 网络(Deep Q-Learning Network,DQN) [14] 算法在 PreScan 平台下使得车辆在城市道路中学习刹车,以减少事故. 文献 [15] 利用多任务学习和强化学习方法,在 VTORCS(Visual TORCS)环境实现了对车辆的横向决策控制. 文献 [16] 使用异步优势评论家算法(Asynchronous Advantage Actor-Critic,A3C) [17] 在 WRC(World Rally Championship)平台下以图像作为输入完成驾驶决策. 文献 [18,19] 使用 DQN 算法在 TORCS 平台下完成驾驶决策. 文献 [20] 提出可控模仿强化学习(Controllable Imitative Reinforcement Learning,CIRL)方法,在 CARLA 平台下证明了其决策具有良好的泛化性. 目前强化学习方向采用离散型动作输出算法(例如将 [0,1] 区间离散为集合{0,0.25,0.5,0.75,1})的研究居多,跳跃式变化的动作取值与真实场景下的驾驶动作(如油门踏板)为无级调节的方式存在较大差异.

深度确定性策略梯度(DDPG)算法 [21] 可以根据状态信息得到连续型动作输出(例如动作取值可在区间 [0,1] 内平滑连续的变化),与真实世界中的驾驶动作更加相近. 本文基于 DDPG 算法构建端到端无人驾驶决策模型,在 TORCS 平台下输入影响车辆控制的关键信息(车辆状态、环境信息)进行训练,输出连续控制量,完成车辆行驶决策. 并与离散型动作输出的 DQN 模型对比车辆控制精度和模型泛化性.

2 DDPG 深度强化学习算法

2.1 深度强化学习

深度学习的提取特征能力与强化学习的序列决策能力相结合,使得在复杂状态空间下的决策问题得以解决,该方向已经成为人工智能领域的一个新的研究热点,并被尝试应用在无人驾驶领域 [22]. 深度强化学习通常由如下四部分构成:①交互环境 E;②智能体 Agent;③状态转移的规则 ρ;④奖励 R 的规则. 其结构如图 1 所示,左侧智能体 Agent 根据环境输出策略,深度神经网络将状态 s 编码为动作 a 输入给右侧交互环境 E,环境根据奖励函数 R 返回该动作的奖励 r 以及根据状态转移规则更换至下一状态.

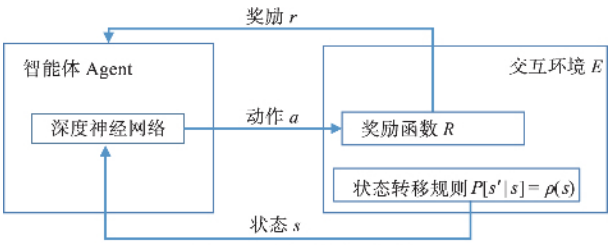

图1 深度强化学习框架

深度强化学习通常将问题建模为马尔科夫决策过程(Markov Decision Process,MDP). 通常情况下将当前的状态 s_t、动作 a_t、奖励 r_t,以及下一状态 s_{t+1} 作为一个元组 (s_t,a_t,r_t,s_{t+1}) 进行收集并构成集合 (S,A,R,S'). 优化目标为策略 $\pi: s\rightarrow a$,通过优化策略 π 使得在 s_t 时刻累计获得奖励 R 式(1)最大,其中 $\gamma\in(0,1)$ 代表衰减因子,使得 R 存在上界,代表未来的奖励对 R 影响逐渐减少.

$$R=\sum_{t'=t}^{T}\gamma^{t'-t}r_t \tag{1}$$

Q 函数被定义为 $Q^{\pi}=\mathrm{E}[R_t|s_t,a_t]$,强化学习优化目标为求得最优策略 π^* 使得式(1) R 的期望最大,即寻找出最优策略 π^* 使得式(2)成立.

$$Q^{\pi^*}(s,a)\geqslant Q^{\pi}(s,a)\quad \forall s,a\in S,A \tag{2}$$

2.2 DDPG 算法

DDPG 算法属于模型无关(Model-Free)深度强化学习方法,可以得到连续型动作输出. 算法采用行动者-评论家(Actor-Critic,AC)结构体系,网络部分由 Actor 网络 $\mu(s|\theta^{\mu})$,Critic 网络 $Q(s,a|\theta^{Q})$ 以及 Actor 网络对应的行动者目标网络(Target-Actor) $\mu(s|\theta^{\mu'})$ 和 Critic 网络对应的评论家目标网络(Target-Critic) $Q(s,a|\theta^{Q'})$ 组成. 除去网络部分还包含随机噪声 N 以增加环境探索能力和经验回放池(Replay-Buffer)以离线策略(Off-Policy)的方式供网络训练.

2.2.1 Actor 网络

Actor 网络通过一组参数 θ^{μ} 来代表当前确定性策略,通过该策略输出动作,而累计奖励 $Q^{\pi}=\mathrm{E}[R_t|s_t,a_t]$ 与动作相关,故如式(3)所示,可以通过梯度上升对 θ^{μ} 进行参数更新,使得 Q^{π} 上升.

$$\begin{aligned}\nabla_{\theta^{\mu}}J&\approx\mathrm{E}_{s_t\sim\rho^{\beta}}[\nabla_{\theta^{\mu}}Q(s,a|\theta^{Q})\,|_{s=s_t,a=\mu(s_t|\theta^{\mu})}]\\&=\mathrm{E}_{s_t\sim\rho^{\beta}}[\nabla_{a}Q(s,a|\theta^{Q})\,|_{s=s_t,a=\mu(s_t)}\\&\quad\cdot\nabla_{\theta^{\mu}}\mu(s|\theta^{\mu})\,|_{s=s_t}]\end{aligned} \tag{3}$$

2.2.2 Critic 网络

Critic 网络通过一组参数 θ^Q 来估计当前状态动作下的 Q 值，该 Q 值以链式法则的形式对式(3)产生影响，所以准确的 Q 值对网络收敛有至关重要的影响. 通过最小化损失函数式(4)对 θ^Q 进行更新，使得 Q 值更加准确.

$$L(\theta^Q)=\mathrm{E}_{s_t\sim\rho^\beta,a_t\sim\beta,r_t\sim\mathrm{E}}[Q(s_t,a_t|\theta^Q)-\hat{Q})^2] \tag{4}$$
$$\hat{Q}=r_t+\gamma Q'(s_{t+1},\mu'(s_{t+1}|\theta^{\mu'})|\theta^{Q'})$$

2.2.3 目标网络

Target-Actor 网络使用参数 $\theta^{\mu'}$ 估计目标动作，Target-Critic 网络使用参数 $\theta^{Q'}$ 估计目标 Q 值. 目标网络的参数更新如式(5)所示，采用滑动平均的方式，与真实网络存在一定的延迟. τ 为滑动平均系数，在实际操作中 τ 的数值远远小于 1，以保证真实网络与目标网络存在一定的差异，从而切断数据相关性.

$$\begin{aligned}\theta^{Q'}&\leftarrow\tau\theta^Q+(1-\tau)\theta^{Q'}\\ \theta^{\mu'}&\leftarrow\tau\theta^\mu+(1-\tau)\theta^{\mu'}\end{aligned} \tag{5}$$

2.2.4 随机噪声

DDPG 的 Actor 网络 $\mu(s|\theta^\mu)$ 输出确定性动作，在相同的状态 s 下会输出相同的结果，使得探索的样本减少. 为使得 Actor 网络进行更好的探索，将对策略 μ 增加随机噪声 N，即 $\mu'(s_t)=\mu(s_t|\theta^\mu)+N$ 使其输出的动作具有随机性，可以进行更多的探索. 在实际训练中噪声 N 随训练迭代次数进行线性衰减至 0，不再对 Actor 网络输出产生影响.

2.2.5 经验回放池

由于 DDPG 算法是离线策略算法，需要利用环境探索的历史数据进行训练，因此我们将每一次 Agent 探索得到的四元组 (s_t,a_t,r_t,s_{t+1}) 进行存储，在网络训练时从经验回放池中随机选取多组 (s_t,a_t,r_t,s_{t+1}) 进行训练，式(3)中的 ρ^β 代表从经验回放池中进行随机抽取的序列对，通过离线策略的形式对网络进行训练.

3 基于 DDPG 的端到端决策控制模型

3.1 系统模型

本文的优化目标为在单车运行环境下，如何利用车辆状态以及环境信息(车辆速度，道路距离等)直接控制车辆驾驶动作(加速，减速，转向)，并平稳、快速地到达终点. 系统总体结构如图 2 所示: Actor 网络与 Target-Actor 网络输出动作，Critic 网络与 Target-Critic 网络负责估计动作 Q 值，经验回放池负责存储探索数据，TORCS 为车辆运行交互环境，奖励函数输出动作的奖励. 整个系统运行流程分为环境探索及网络训练两部分.

3.1.1 环境探索

如图 2 所示，在任意时刻 t，Actor 网络根据车辆状态 s_t，通过神经网络 θ^μ 计算出此时车辆采取的动作 $a_t=\mu(s_t|\theta^\mu)$，将当前动作 a_t 加噪声 N 后通过共享内存的方式发送给交互环境控制车辆，得到下一状态 s_{t+1} 以及代表在 a_t 动作下环境给予的奖励 $r_t=R(s_{t+1})$. 此时车辆完成对 (s_t,a_t,r_t,s_{t+1}) 的收集，将探索到的数据对儿存入经验回放池供网络进行离线训练.

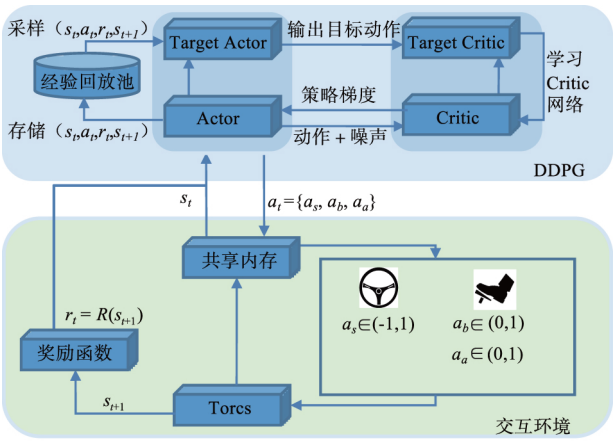

图2　系统模型

3.1.2 网络训练

首先在经验回放池中采样数据 (s_t,a_t,r_t,s_{t+1})，根据式(3)更新 Actor 网络参数 θ^μ 以优化 Q 值上升. 然后 Target-Actor 网络根据采样的 s_{t+1} 计算出估计动作 a_{t+1}，将 s_{t+1}、a_{t+1} 传给 Target-Critic 网络计算下一状态的 Q 值 $\hat{Q}=Q(s_{t+1},a_{t+1}|\theta^{Q'})$，同时 Critic 网络计算当前状态 Q 值 $Q=Q(s_t,a_t|\theta^Q)$，通过最小化损失函数式(4)更新 Critic 网络参数 $\theta^{Q'}$. 最终根据式(5)更新目标网络.

3.2 状态空间和动作空间

车辆状态空间 $s_t=\{v,w,\delta,\theta,\zeta,\varepsilon\}$ $(\dim(s_t)=29)$，具体描述如表 1 与图 3 所示. 其中速度 v 可以分解为 x、y、z 三个轴方向速度，车辆距车道中心偏移量 δ 归一化到 $(-1,1)$ 之间，车辆前方道路距离 ζ 采样角度为 $\{-45,-19,-12,-7,-4,-2.5,-1.7,-1,-0.5,0.5,1,1.7,2.5,4,7,12,19,45\}$.

表 1　状态空间表

符号定义	状态描述	取值范围
v	车辆速度($\dim(v)=3$)	(0,300)(km/h)
w	车轮角速度($\dim(w)=4$)	(0,+∞)(rad/s)
δ	车辆距中心偏移量(0 为中心，左正右负)($\dim(\delta)=1$)	(−1,1)(Normalization)
θ	车辆与道路偏离角度($\dim(\theta)=1$)	(−π,π)(rad)
ζ	车辆与前方道路距离($\dim(\zeta)=19$)	(0,200)(m)
ε	车辆引擎转速($\dim(\varepsilon)=1$)	(0,+∞)(rpm)

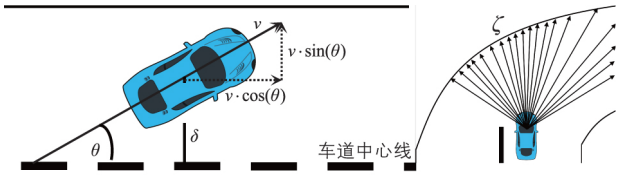

图3 驾驶模拟环境

车辆动作空间 $a_t=\{a_a,a_b,a_s\}$ ($\dim(a_t)=3$) ,定义如表 2 所示.

表 2 动作空间表

符号定义	动作描述	取值范围
a_a	油门力度(0 为无,1 为满)	(0,1)
a_b	刹车力度(0 为无,1 为满)	(0,1)
a_s	转向值(-1、+1 分别代表全右、全左)	(-1,1)

3.3 Actor 网络

Actor 网络结构如图 4 所示. Actor 网络为全连接网络,网络输入为 t 时刻状态 s_t,经过两层全连接层后输出动作 a_t. 其中各隐层之间使用 Relu 激活函数,在最终的输出层中使用 tanh 激活函数将转角动作规范在(-1,1) 之间,使用 sigmoid 激活函数将刹车动作和加速动作规范在(0,1) 之间. Target-Actor 网络结构与 Actor 网络相同.

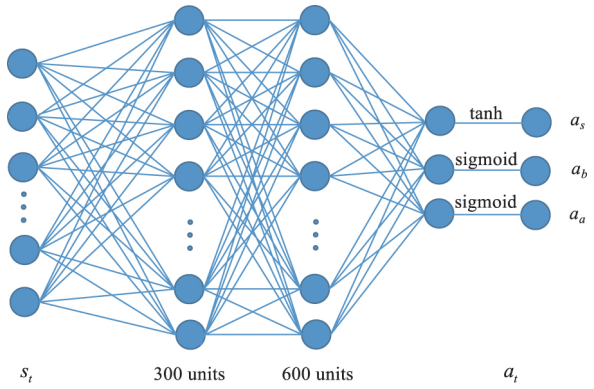

图4 Actor网络

3.4 Critic 网络

Critic 网络结构如图 5 所示. 网络输入包含 s_t 与 a_t 两部分,其中先将 s_t 作为输入,通过两层全连接层之后分别得到特征 l_s^1、l_s^2. 再将 a_t 作为输入通过一个隐层后得到特征 l_a^1. 此时 l_s^2 与 l_a^1 的维度均为 600,将 l_s^2 与 l_a^1 逐级相加融合得到 l_{sa},再通过最后全连接层得到 Q 值. l_s^1 使用 Relu 作为激活函数,l_s^2、l_a^1 均无激活函数,l_{sa} 通过全连接层后使用 Relu 作为激活函数. 最终输出 Q 值无激活函数. Target-Critic 网络与 Critic 网络相同.

3.5 奖励函数设定

奖励函数输入为{ σ,δ,θ,v} ,其中 σ 代表车辆受到的损伤,δ 代表车辆当前位置,θ 代表车辆当前与道路切向夹角,v 代表车辆当前速度. 本文目标为让车辆沿道路切向速度高并且尽可能保持在路中央,奖励函数设定如算法 1 所示. $\text{stuck}_{\text{count}}$为车辆静止计数器,在 TORCS 中异步采取,与交互过程同时运行.

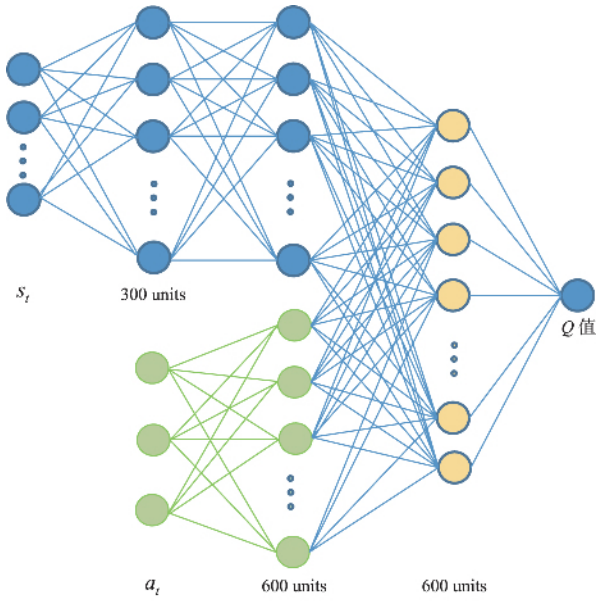

图5 Critic网络

算法 1 奖励函数

输入: σ,δ,θ,v
输出: 奖励值 r,回合结束标志 done
1: if $\text{stuck}_{\text{count}}>200$ and $\cos(\theta)\cdot v<5$ then
2: $\text{stuck}\leftarrow$True, $\text{stuck}_{\text{count}}\leftarrow 0$ ▷车辆前进速度过低
3: else
4: $\text{stuck}_{\text{count}}\leftarrow\text{stuck}_{\text{count}}+1$
5: end if
6: if $\sigma>0$ then ▷发生碰撞
7: $r\leftarrow -1$
8: else
9: $r\leftarrow\cos(\theta)\cdot v-|\sin(\theta)\cdot v|-|\delta\cdot v|$
10: end if
11: if $\text{abs}(\delta)>1$ then ▷车辆越出车道,回合结束
12: $r\leftarrow -200$, done$\leftarrow$True
13: else if $\cos(\theta)<0$ or stuck then ▷车辆逆向或陷住,回合结束
14: done$\leftarrow$True
15: end if

4 实验

4.1 实验环境与内容

本文实验平台为 ubuntu16.04,python3.5,选取 pytorch 深度学习框架,模拟环境为 TORCS-1.3.1,环境模拟频率为 3Hz. 训练车道与验证车道如图 6 所示,选取 TORCS 中 CG SpeedWay-1 与 CG Track-3 进行分阶段训

练. CG SpeedWay-1 包含长直线、连续转弯、上下坡等丰富场景,十分适合进行第一阶段的初始训练. CG Track-3 包含 U 型弯、直角弯等复杂弯道,适合第二阶段的增强泛化性训练. 验证车道选取 TORCS 中的 E-Track 5 与 Alpine 1. 实验内容上首先训练得到连续型动作输出的 DDPG 模型,使车辆能够沿车道自主行驶,然后对训练结果与验证结果进行分析,并与离散型动作输出的 DQN 模型从车辆控制精度和模型泛化性方面对比.

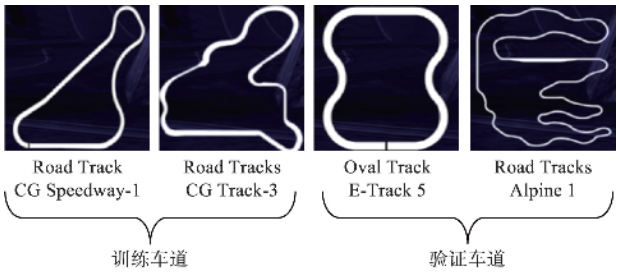

图6　训练车道与验证车道

4.2　DDPG 网络训练

4.2.1　网络结构及参数

网络结构如 3.3 与 3.4 小节所示. Actor 网络优化算法使用 Adam 算法,学习率 $\alpha = 0.0001$,Critic 网络优化器算法使用 Adam 算法,学习率 $\alpha = 0.001$,网络平滑参数 $\tau = 0.001$,衰减因子 $\lambda = 0.99$. 经验回放池大小为 100000,批训练大小为 32. 动作的噪声函数 N 使用 Ornstein-Uhlenbeck(OU) 噪声,噪声影响因数 ε 如式(6)所示:

$$\varepsilon = \max\left(0, 1 - \frac{\text{step}}{\text{exploration}}\right) \tag{6}$$

使得 OU 随机的力度随训练步数而下降,exploration 在两阶段的取值依次为 100000、50000. GPU 设备使用 GTX1070,第一阶段训练 320000 步、共计 1593 个训练回合,第二阶段训练 180000 步、共计 1310 个训练回合.

4.2.2　车辆行驶的学习过程

初始训练车道如图 7 所示,起点与终点为同一位置. 车辆从起点处开始沿逆时针方向(内侧箭头方向)驾驶通过终点为完成 1 圈驾驶. 训练过程中有代表性的四种路段: A 代表左转弯,B 代表直线,C 代表连续反向转弯,D 代表急转弯. 当车辆可以通过这四段后网络基本收敛可以完成 1 圈驾驶,但网络还未完全收敛,偶尔会冲出车道. 实验结果表明车辆连续跑完 20 圈后网络收敛.

A 段为车辆起步阶段,环境状态为左转弯,当车辆可以成功起步通过该段时车辆只学会左转,但由于此时网络刚刚开始训练,车辆还未学会如何驾驶,速度基本为 0 时奖励为 0,根据图 8 的 0 步到 80000 步可以观测到奖励大部分集中在 0 附近.

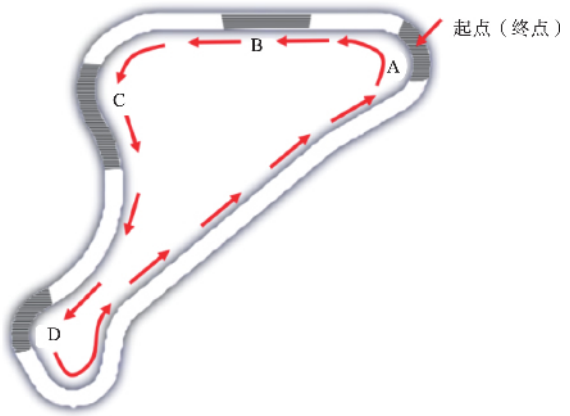

图7　学习过程分解

B 段环境状态为直线,由于通过 A 段后只学会左转,经过一段时间探索学习后可以适应直线,但此时车辆因为随机噪声仍然存在摆动现象,偶尔会冲出车道.

C 段环境状态为反向连续转弯车道且后半段为右转弯,由于此时经验回放池中数据没有反向转弯和右转弯数据,车辆会在这里进行相当长时间的环境探索学习.

D 段环境状态为急转弯,当学会 A,B,C 三个路段后车辆已经能够以较快的速度驾驶,但此时车辆不会刹车,经常冲出车道. 根据图 8 可以观察 80000 步到 160000 步虽然整体奖励分布变高,但是冲出车道次数相对于 0 步到 80000 步反而提高也可以说明此时车辆不会急转弯状态下的决策,在该段经过 10% 概率刹车探索学习后,学会采取刹车通过. 至此车辆已经可以顺利的完成一圈,但此时网络仍然不稳定,偶尔会冲出车道.

4.2.3　奖励分布

训练时的奖励值分布可以代表深度强化学习训练效果. 选取更具代表意义的第一阶段奖励值进行分析,该阶段能够体现出车辆从无到有的驾驶学习过程. 奖励分布直方图如图 8 所示,该图将训练中的 320000 步分为四部分,每 80000 步进行一次统计,横轴为奖励大小,区间大小为 10,纵轴为分布概率. 可以明显观察到在 80000 步之前车辆奖励经常为 -200,并且很大一部分集中在 0 附近,说明刚开始网络在随机探索,速度较低导致奖励集中在 0 附近,并且由于经常冲出车道,使奖励变为 -200. 在 80000 至 160000 步之间奖励的分布发生改变,由最初的以(0,100)为主体逐渐变为以(80,160)为主体,分布整体向奖励变大方向偏移,说明此时智能体已经学到了如何在车道中驾驶,但仍然会有很高的概率冲出车道,而在 160000 步之后奖励分布没有明显变化,并且基本不会冲出车道,说明此时网络已经收敛.

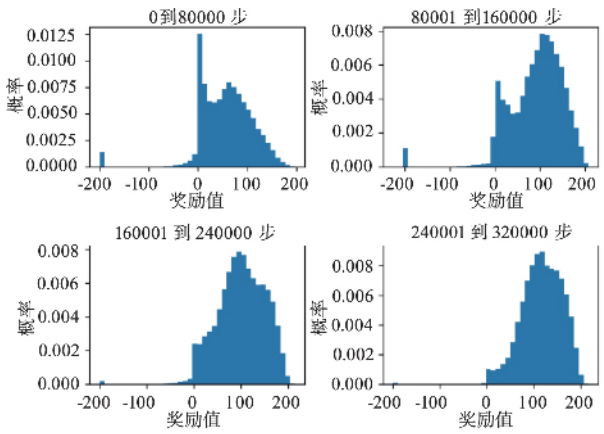

图8 奖励分布图

4.3 DQN 实验参数

DQN 实验的状态空间与 DDPG 实验一致，动作空间如表 3 所示(动作定义参考表 2)，网络结构与 DDPG 的 Actor 网络类似(输出层变更为 15 分类)，实验中的超参数等设置均参考 DDPG 实验.

表 3 DQN 动作空间表

分类数	动作取值		
	a_s	a_a	a_b
10	{0, ±0.1, ±0.3}	{0.3, 0.7}	{0}
5	{0, ±0.1, ±0.3}	{0}	{0.1}

选用典型的离散数值作为动作取值可以避免 DQN 模型由于动作组合造成的输出维度过高，而较少的动作组合又将导致模型在部分场景下所做出的动作选择均无法得到理想的奖励值，影响决策的准确性. 故本文通过在 DDPG 奖励函数的基础之上，添加了如算法 2 所示的限定条件(变量定义参考表 1、表 2)，在边界情况辅助车辆进行决策，降低车辆的学习难度，使得 DQN 模型能够完成与 DDPG 模型相同的驾驶任务.

算法 2 DQN 奖励函数限定条件

```
▷向车道中心调整时取消转角惩罚
if (δ>0.0 and a_s<0.0) or (δ<0.0 and a_s>0.0) then
    r←r + |sin(θ) · v|
end if
▷起步时刹车惩罚;限制偏离车道中心幅度
if v<50.0 and a_b>0.0 then
    r←−100
else if (δ>0.5 and a_s≥0.0) or (δ<−0.5 and a_s≤0.0) then
    r←−100
end if
```

4.4 训练结果分析

4.4.1 第一阶段训练结果

第一阶段训练结果如表 4 所示，表头从左至右分别代表当前圈数、该圈完成时间、该圈最高速度与该圈最低速度. 表 4 中还包含了我们利用离散型动作输出的 DQN 算法进行对比实验的数据.

表 4 Track CG Speedway-1 训练结果

圈数(Lap)	时间(s)	最高速度(km/h)	最低速度(km/h)
	DDPG 模型数据/DQN 模型数据		
1	53:42/1:44:02	198/96	111/46
2	47:02/1:36:87	199/98	103/48
3	47:59/1:37:25	199/97	112/55
4	47:37/1:36:78	202/97	115/53
5	46:87/1:36:92	198/96	110/54
6	47:39/1:36:46	202/98	103/56
7	47:60/1:37:90	192/98	112/52
8	47:38/1:37:26	202/97	103/55
9	47:51/1:38:07	200/97	113/54
10	47:31/1:37:03	197/96	111/54

两种模型均能够控制车辆完成 10 圈的驾驶任务，说明网络均已完全收敛，可以顺利完成驾驶. 第一阶段的训练中没有对车辆行驶的最高速度进行限定，故单圈平均速度会随着训练的推进而不断增加，直至达到完成训练任务所允许的极限. 此阶段能够充分发挥模型对车辆的控制能力，体现出连续型动作输出与离散型动作输出在控制精准度方面的差异.

此阶段训练结果表明 DDPG 模型能够在更高的车速状态下控制车辆完成驾驶任务. 在奖励函数主体相同的条件下，DDPG 实验模型在 Track CG Speedway-1 训练车道上最快速度为 202km/h，最快记录为 46.87s，明显优于 DQN 模型. 这是由于 DQN 模型所能离散的动作组合有限且固定，在高速状态下无法做出适宜的动作组合对车辆进行控制，从而导致车辆在高速状态下会冲出车道，很难学会如何在高速状态下行驶.

4.4.2 第二阶段训练结果

为消除训练模型对单一车道过拟合的影响，并增强训练模型的泛化性，我们在 Track CG Track-3 上进行了第二阶段的训练. 此阶段中通过奖励函数将车速上限设定为 100km/h(当已超速且继续加速时给予负奖励惩罚). 训练结果如表 5 所示，DDPG 模型与 DQN 模型均能够完成训练任务，并达到限定的最高车速.

此阶段训练结果表明 DDPG 模型已学会将车速控制在目标范围之内. DQN 模型的单圈完成时间略短，这是由于 DQN 模型可采取的油门、刹车动作值有限，模型无法很好的学习到主动降速，只能在超速后依靠强制降速将车速限定在目标范围之内，所以 DQN 模型的平均车速会更加接近 100km/h. 而 DDPG 模型学会了通过

自主调节油门与刹车将车速限定在目标范围之内，在最高车速 100km/h 的限定下，单圈均速约为 80km/h. 因此 DDPG 模型的车速波动范围较大、单圈完成时间略长.

表 5　Track CG Track-3 训练结果

圈数(Lap)	时间(s)	最高速度(km/h)	最低速度(km/h)
	DDPG 模型数据/DQN 模型数据		
1	2:05:23/1:49:93	107/102	54/86
2	1:54:97/1:42:18	99/102	69/84
3	1:56:50/1:41:90	98/102	62/87
4	1:55:93/1:41:89	102/102	54/85
5	1:55:94/1:42:34	99/102	65/84
6	1:55:63/1:41:79	102/102	76/96
7	1:57:41/1:41:93	98/102	46/94
8	1:56:49/1:43:09	107/102	64/54
9	1:53:77/1:41:96	109/102	77/85
10	1:56:00/1:41:77	106/102	60/93

4.5　验证结果分析

验证车道如图 6 所示，DDPG 模型在更复杂的 Alpine 1 和更简单的 E-Track 5 中均可以完成驾驶任务，而 DQN 模型只能在更简单的 E-Track 5 中完成. 验证结果如图 9 ~ 图 11 所示，横轴代表迭代步数，纵轴代表奖励数值与车辆速度，绿色实线代表奖励值，红色圆形虚线代表 x 轴分解速度，蓝色三角虚线代表 y 轴分解速度. 模型需要在验证车道中行驶 3 圈，可以看到每张图中均具有很强的周期性.

E-Track 5 中的道路平坦，弯道较缓且规律性强，DDPG 模型与 DQN 模型均可以在该车道中完成验证，x 轴分解速度 speedx 均接近最大车速 100km/h. 如图 9、图 10 所示，DQN 模型的 x 轴分解速度长期处于最高限速 100km/h 附近，说明 DQN 模型的降速控制学习效果

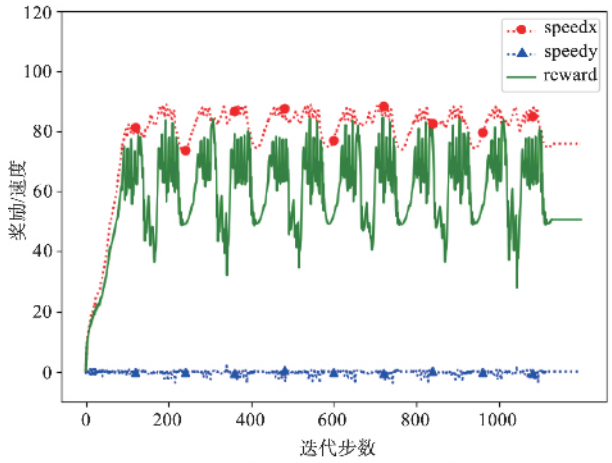

图9　DDPG模型E-Track 5验证结果

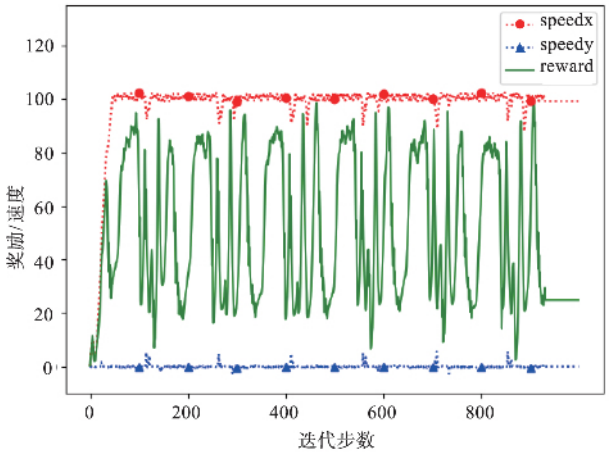

图10　DQN模型E-Track 5验证结果

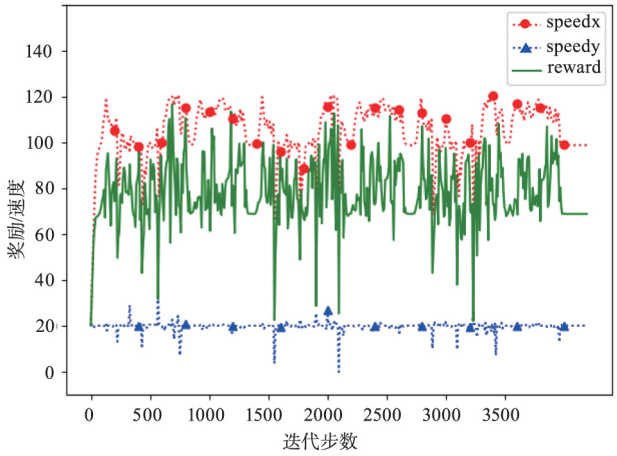

图11　DDPG模型 Alpine 1 验证结果

不佳. DDPG 模型的平均奖励值更高，说明其可以更好的贴合道路中心行驶，奖励值波动幅度更小，说明车辆在过弯时依旧不会严重偏离道路中心. 验证结果总体表明 DDPG 模型对车辆的控制更加精准.

Alpine 1 中多为连续弯道，并包含多个 U 型急弯，仅 DDPG 模型可以在该车道中完成验证，DQN 模型在该车道中会冲出道路边界、无法完成驾驶，表明 DDPG 模型相对于 DQN 模型具有更优越的泛化能力. 由于该车道的路况复杂，故图 11 中各曲线的波动幅度均有所增加，可以看到当 x 轴分解速度降低时，对应的 y 轴分解速度有所增高，说明此时车辆正在减速过弯. 当迭代步数在 200 ~ 700 区间时，对应于车道中的连续 U 型弯路，此时 x 轴分解速度波动的均值较低，而 700 ~ 1300 步迭代时对应的弯路较少，x 轴分解速度则在较高的均值处波动，表明车辆已学会合理的运用油门与刹车调节车速完成驾驶任务.

5　结论

本文建立基于 DDPG 深度强化学习算法的端到端

无人驾驶决策控制模型.在 TORCS 平台下通过输入连续的车辆行驶环境感知信息,决策控制模型直接输出加速、刹车、转向驾驶动作,实现了端到端无人驾驶决策.实验结果表明连续型动作输出的 DDPG 模型相比于离散型动作输出的 DQN 模型,在无人驾驶决策控制的精确度以及模型泛化能力方面更具优势.在未来的工作中,希望对 Actor 网络进行可视化,观察状态输入对车辆转向、加速、减速动作决策的影响.

参考文献

[1] 熊璐,康宇宸,张培志,等.无人驾驶车辆行为决策系统研究[J].汽车技术,2018,515(08):1-9.
XIONG Lu,KANG Yu-chen,ZHANG Pei-zhi,et al. Research on behavior decision-making system for unmanned vehicle [J]. Automobile Technology,2018,515(08):1-9.(in Chinese)

[2] 刘国荣,张扬名.移动机器人轨迹跟踪的模糊 PID-P 型迭代学习控制[J].电子学报,2013,41(8):1536-1541.
LIU Guo-rong,ZHANG Yang-ming. Trajectory tracking of mobile robots based on fuzzy PID-P type iterative learning control [J]. Acta Electronica Sinica,2013,41(8):1536-1541.(in Chinese)

[3] POMERLEAU D A. ALVINN: An autonomous land vehicle in a neural network [A]. Advances in Neural Information Processing Systems [C]. San Francisco,California,USA: Margan Kaufmann,1989.305-313.

[4] MULLER U,BEN J,COSATTO E,et al. Off-road obstacle avoidance through end-to-end learning [A]. Advances in Neural Information Processing Systems [C]. Cambridge,MA,USA: MIT Press,2006.739-746.

[5] BOJARSKI M,DEL TESTA D,DWORAKOWSKI D,et al. End to End Learning for Self-driving Cars [OL]. https://arxiv.org/pdf/1604.07316.pdf,2016.

[6] BOJARSKI M,YERES P,CHOROMANSKA A,et al. Explaining How a Deep Neural Network Trained with End-to-end Learning Steers a Car [OL]. https://arxiv.org/pdf/1704.07911.pdf,2017.

[7] WANG X,JIANG R,LI L,et al. Capturing car-following behaviors by deep learning [J]. IEEE Transactions on Intelligent Transportation Systems,2018,19(3):910-920.

[8] XU H Z,GAO Y,YU F,et al. End-to-end learning of driving models from large-scale video datasets [A]. 2017 IEEE Conference on Computer Vision and Pattern Recognition [C]. Honolulu,HI,USA: IEEE,2017.3530-3538.

[9] ZHANG X,SUN J,QI X,et al. Simultaneous modeling of car-following and lane-changing behaviors using deep learning [J]. Transportation Research,2019,104(JUL.):287-304.

[10] LOIACONO D,PRETE A,LANZI P L,et al. Learning to overtake in torcs using simple reinforcement learning [A]. Proceedings of the IEEE Congress on Evolutionary Computation [C]. Barcelona,Spain: IEEE,2010.1-8.

[11] WYMANN B,ESPIE E,GUIONNEAU C,et al. TORCS: The Open Racing Car Simulator [OL]. http://www.torcs.org,2014.

[12] XIA W,LI H Y,LI B. A control strategy of autonomous vehicles based on deep reinforcement learning [A]. Proceedings of the IEEE International Symposium on Computational Intelligence and Design [C]. Hangzhou,China: IEEE,2016.198-201.

[13] CHAE H,KANG C M,KIM B D,et al. Autonomous braking system via deep reinforcement learning [A]. 2017 IEEE 20th International Conference on Intelligent Transportation Systems [C]. Yokohama,Japan: IEEE,2017.1-6.

[14] MNIH V,KAVUKCUOGLU K,SILVER D,et al. Human-level control through deep reinforcement learning [J]. Nature,2015,518(7540):529-533.

[15] LI D,ZHAO D,ZHANG Q,et al. Reinforcement learning and deep learning based lateral control for autonomous driving [J]. IEEE Computational Intelligence Magazine,2019,14(2):83-98.

[16] JARITZ M,DE CHARETTE R,TOROMANOFF M,et al. End-to-end race driving with deep reinforcement learning [A]. 2018 IEEE International Conference on Robotics and Automation [C]. Brisbane,QLD,Australia: IEEE,2018.2070-2075.

[17] MNIH V,BADIA A P,MIRZA M,et al. Asynchronous methods for deep reinforcement learning [A]. International Conference on Machine Learning [C]. New York,USA: ACM,2016.1928-1937.

[18] VITELLI M,NAYEBI A. CARMA: A Deep Reinforcement Learning Approach to Autonomous Driving [OL]. https://web.stanford.edu/~anayebi/projects/CS_239_Final_Project_Writeup.pdf,2016.

[19] SALLAB A,ABDOU M,PEROT E,et al. Deep reinforcementlearning framework for autonomous driving [J]. Electronic Imaging,2017,2017(19):70-76.

[20] LIANG X,WANG T,YANG L,et al. Controllable imitative reinforcement learning for vision-based self-driving [A]. European Conference on Computer Vision [C]. Munich,Germany: Springer-Verlag,2018.604-620.

[21] LILLICRAP T P,HUNT J J,PRITZEL A,et al. Continuous control with deep reinforcement learning [A]. International Conference on Learning Representations [C]. Caribe Hilton,San Juan,Puerto Rico: ICLR,2016.1-14.

[22] 刘全,翟建伟,章宗长,等.深度强化学习综述[J].计算

机学报,2018,41(01):1 -27.
LIU Quan, QU Jian-wei, ZHANG Zhong-chang, et al. A survey on deep reinforcement learning [J]. Chinese Journal of Computers, 2018, 41(01): 1 -27. (in Chinese)

作者简介

黄志清　男,1970 年 11 月出生于四川荣县,博士. 现为北京工业大学信息学部副教授,主要研究方向为无人驾驶智能决策控制、车联网及区块链等.
E-mail: zqhuang@ bjut. edu. cn

曲志伟　男,1995 年 2 月出生于山东省烟台市. 现为北京工业大学信息学部硕士研究生,主要研究方向为无人驾驶智能决策控制与深度强化学习.
E-mail: quzhiwei@ emails. bjut. edu. cn

张　吉　男,1994 年 12 月出生于北京市. 分别于 2016 年和 2019 年在北京工业大学获得学士和硕士学位. 研究方向为强化学习和自动驾驶.
E-mail: s201625019@ emails. bjut. edu. cn

张严心　女,1976 年 2 月出生于辽宁省盘锦市,博士. 现为北京交通大学电子信息与工程学院副教授,主要研究方向是复杂大系统的智能控制、无人驾驶中的智能控制、复杂交通网络控制等.
E-mail: yxzhang@ bjtu. edu. cn

田　锐　男,1983 年 1 月出生于湖北省天门市,博士. 现为北京工业大学信息学部讲师,主要研究方向是车联网、区块链、多方安全计算等.
E-mail: rui. tian@ bjut. edu. cn