doi:10.3772/j.issn.1002-0470.2024.01.001

基于持续强化学习的自动驾驶赛车决策算法研究①

牛京玉②*** 胡 瑜③*** 李 玮* 韩银和***

(*中国科学院计算技术研究所智能计算机研究中心 北京 100190)

(**中国科学院大学 北京 100049)

摘 要 赛道形状与路面材质变化对自动驾驶赛车的行为决策带来了严峻挑战。为应对道路间的动力学差异,本文提出一种基于持续强化学习(CRL)的高速赛车决策算法。该算法将不同道路看作独立任务。算法的第 1 训练阶段负责提取描述不同任务上赛车动力学的低维特征,从而计算出任务间的相似性关系。算法的第 2 训练阶段负责为策略学习过程提供 2 个持续强化学习约束:其一是权重正则化约束,策略网络中对于旧任务重要的权重将在新任务学习期间被限制更新,其限制力度由任务相似性自适应调节;其二是奖励函数约束,鼓励在新任务学习期间策略的旧任务性能不下降。设计不同任务排序下的赛车实验和持续强化学习评价指标以评估算法性能。实验结果表明,所提算法能在既不存储旧任务数据也不扩展策略网络的条件下获得比基准方法更出色的驾驶性能。

关键词 强化学习(RL);持续学习;行为决策;自动驾驶赛车;动力学特征提取

自动驾驶赛车挑战赛[1-4]的兴起反映出自动驾驶赛车已成为当下促进高速自动驾驶技术发展的一个研究热点。最近的研究进展表明,深度强化学习(deep reinforcement learning,DRL)[5]是解决自动驾驶赛车决策问题的一个潜力研究方向[6-7]。DRL 通过试错方式自主学习最优驾驶策略,令赛车在动力学模型固定的道路上实现累计奖励的最大化。这里的动力学模型由道路形状以及摩擦系数、滚动阻力系数、粗糙度等路面物理参数信息共同描述。动力学模型不同的道路被视为不同任务。当道路的形状或路面参数发生变化时,传统 DRL 策略会在继续学习新任务时遗忘旧任务,即发生灾难性遗忘,从而无法应对道路多变的实际赛车需求,比如拉力赛[8]。为此,针对赛道涵盖多种道路形状和材质的情况,本文开展面向多任务的持续强化学习[2-10](continual reinforcement learning,CRL)自动驾驶赛车决策算法研究。

CRL 的核心定义是策略在不断学习新任务的过程中不遗忘旧任务知识,即无灾难性遗忘。其还有 2 个进阶能力:积极前向迁移和后向迁移。前者指策略利用旧任务知识帮助新任务更快或更好收敛的能力。后者指策略利用新任务知识反过来帮助提高旧任务表现的能力。可见,无灾难性遗忘就是积极后向迁移的下限。

现有相关 CRL 工作可分为 3 类:经验回放方法[11-17]、参数独立方法[18-19]和权重正则化方法[20-23]。然而,前 2 类方法是通过持续保存旧任务数据或扩张策略网络规模来实现 CRL,存在数据存储压力大、可扩展性差的缺点,难以满足复杂自动驾驶任务的长期决策需求。第 3 类权重正则化方法计算策略网络权重对各个旧任务的重要性,并限制对旧任务重要的权重在新任务学习期间的更新。该类方法能

① 国家自然科学基金(62176250,62003323)和中国科学院计算技术研究所计算机体系结构国家重点实验室创新项目(CARCH5203,CARCH5406)资助。

② 女,1992 年生,博士生;研究方向:基于深度强化学习的自动驾驶决策;E-mail:hnnjy123@163.com。

③ 通信作者,E-mail:huyu@ict.ac.cn。

(收稿日期:2022-12-12)

在无需存储旧任务数据或扩张网络规模的条件下令强化学习(reinforcement learning, RL)策略避免灾难性遗忘,可满足自动驾驶应用需求。但权重正则化方法对所有旧任务施加了无差别的权重约束,忽略了任务之间的相似性关系,导致其持续学习能力受限。另外,这些方法还存在 2 点不足:未充分利用 RL 的优化要素来提升持续学习效果,以及未应用于复杂的高速驾驶任务中。

为克服上述缺点,本文提出一种融入任务相似性的两阶段 CRL 算法框架。其中包含 4 个主要创新点。(1)算法第 1 训练阶段提出了一种无监督的任务特征提取方法。该方法包含 1 个特征提取器和 1 个动力学权重生成器。两者的联合训练可获取描述任务动力学特征的低维向量。这些任务特征用于计算任务相似性关系。(2)算法第 2 阶段提出了一种融入任务相似性的权重正则化方法。该方法利用任务相似性关系来自适应地调节策略网络权重的更新约束。当新旧任务相似性低时,加强限制对旧任务重要的部分权重发生变化。反之,减弱对权重更新的限制,以提供任务间进行知识迁移的机会,获得进阶 CRL 能力。(3)算法第 2 阶段还设计了一个适用于持续学习的 RL 奖励函数,促使策略向着共同提高新任务和旧任务性能的方向优化。(4)本文利用赛车仿真平台设置了一系列测试实验,定义了一套评估 CRL 方法的性能指标。实验结果表明,在不存储旧任务数据且不扩张策略网络规模的条件下,本文算法能获得比所有基准 CRL 方法更亮眼的成绩。

1 相关工作

持续学习已经在监督学习领域,尤其是在图像分类任务中获得了较深入的研究[24-26]。最近,RL 中的持续学习问题[9-10]也受到了越来越多的关注。相关 CRL 方法主要可分为经验回放方法、参数独立方法和权重正则化方法。

经验回放方法实现 CRL 能力的思路是存储旧任务数据,并用其与新任务数据一起优化策略。其中一部分方法关注如何在维持无灾难性遗忘的同时缓解数据存储压力,比如利用不同数据抽样方法精简存储[11-13]或学习数据生成模型[14-15]。该类别中的另一部分方法关注如何获取进阶 CRL 能力。比如梯度情景记忆(gradient episodic memory,GEM)方法[16]和平均梯度情景记忆(averaged GEM,A-GEM)方法[17]约束策略在新任务上的梯度更新方向与在旧任务上的梯度方向间夹角不超过 90 °,从而与新任务相似的旧任务能在新任务学习中获得积极的后向迁移。但随着任务数的增加,这类方法的数据存储压力加剧,不适合自动驾驶赛车应用的长期决策需求。

参数独立方法保留旧任务知识的思路是为每个任务扩展一部分独立的策略网络分支,其代表性方法是渐进神经网络(progressive neural networks, PNN)[18-19]。除了避免灾难性遗忘问题外,该方法还具备积极的前向迁移能力。其得益于 PNN 中从旧任务网络到新任务网络的横向连接,令策略有选择地从相似旧任务网络中获取促进新任务收敛的有用知识。但策略网络分支彼此独立一方面导致无法发展后向迁移能力,另一方面导致网络规模随着任务数的增加而不断扩张,可扩展性差。该方法同样不满足自动驾驶赛车的应用需求。

权重正则化方法的思路是计算策略网络权重对每个旧任务的重要性,并在新任务训练时限制对旧任务重要的部分策略权重更新。其代表性方法是利用 Fisher 信息矩阵估计重要性的可塑权重巩固(elastic weight consolidation,EWC)[20-21]。在线可塑权重巩固(online EWC)方法[22-23]降低了 EWC 方法的计算成本。渐近压缩(progress & compress,P&C)方法[23]结合 PNN、online EWC 和知识蒸馏思想提出一种可扩展策略。这类方法既无需存储旧任务数据也无需扩张网络规模,该优势使其适用于自动驾驶赛车的应用需求。但由于权重正则化方法对所有旧任务的无差别约束,这类方法仅局限于解决无灾难性遗忘问题。将权重正则化方法和可获取部分进阶 CRL 能力的 GEM、A-GEM 和 PNN 方法相比较可以发现:可获得部分进阶 CRL 能力的 3 个方法的共性在于均隐含考虑了新旧任务间的关系。因此,本文提出一种融入任务相似性的权重正则化方法。该方法利用任务相似性对不同旧任务的约束力度进行自

适应调节,从而实现积极前向和后向迁移能力。为计算任务相似性,本文提出一种特征提取方法来获取描述每个任务动力学特征的低维向量。此外,本文还设计了一个适应 CRL 的奖励函数,以进一步发掘 RL 优化范式下的持续学习能力。上述 3 项创新设计共同构成了本文满足自动驾驶赛车需求的 CRL 算法。

2 方法介绍

2.1 问题定义

本文旨在提出 CRL 算法解决传统 RL 策略面对道路变化时的灾难性遗忘问题。每段动力学表现不同的道路被看作一个独立任务。赛车在每个任务上的 RL 决策过程均为一个由元组 (S,A,P,r,γ) 表示的马尔科夫决策过程。其中,S 是状态空间,A 是动作空间,P 是描述任务动力学的状态转移函数,r 是引导 RL 优化方向的奖励函数,γ 是折扣因子。不同任务具有相同的状态空间、动作空间和折扣因子。任务间的差异体现在状态转移函数和奖励函数随任务变化。设 $i \in \{1,2,\cdots,T\}$ 是任务标识,t 是时间步,状态转移函数和奖励函数分别表达为 $P^i(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t,\boldsymbol{a}_t)$ 和 $r_t^i(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{s}_{t+1})$。第 t 时间步时,赛车状态 $\boldsymbol{s}_t \in S$ 是一个 29 维多传感信息向量,包括赛车和道路轴线的夹角及距离、三维速度信息、赛车与道路边缘的 19 维测距信息、4 个轮速转速和发动机转速。赛车动作 $\boldsymbol{a}_t \in A$ 表达为由转向、加速度组成的二维向量。

2.2 算法概述

当赛车遇到一个新任务 i 时,图 1 展示了本文提出 CRL 算法的 2 个训练过程。第 1 训练阶段旨在提取可描述当前任务动力学的低维特征向量 $\boldsymbol{feature}_i$。该任务特征的提取利用了一个编码器 E 和一个权重生成器 G_{dyn} 的联合训练结构实现,过程详见 2.3 节。学到的新任务特征被用于计算新任务与前 $i-1$ 个旧任务之间的相似性关系。第 2 训练阶段旨在向着既学习新任务又不遗忘所有旧任务知识的方向优化当前策略 $\boldsymbol{a}_t^i = \pi_i(\boldsymbol{s}_t^i, \boldsymbol{feature}_i)$。策略的 CRL 能力由融入任务相似性的权重正则化方法以及鼓励持续学习的 RL 奖励函数共同实现,详见 2.4 节的介绍。

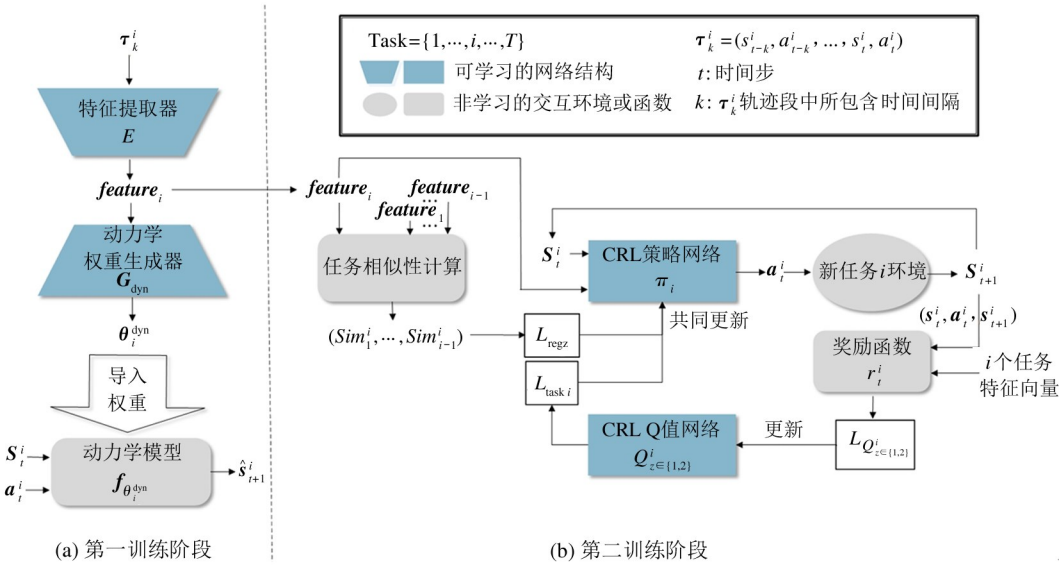

图 1 本文两阶段 CRL 算法的训练过程总览图

2.3 第 1 训练阶段:提取任务特征

本文算法的第 1 训练阶段展示在图 1(a)中。首先,算法通过随机驾驶起点和控制动作的方式收集新任务环境 i 中的轨迹数据。这些数据包含了车辆和任务 i 道路之间的动力学关系。然后,从轨迹数据集中取时间间隔为 k 的数据 $\boldsymbol{\tau}_k^i = (\boldsymbol{s}_{t-k}^i,\boldsymbol{a}_{t-k}^i,\cdots,\boldsymbol{s}_t^i,\boldsymbol{a}_t^i)$ 喂进特征提取器中输出特征向量,即 $E(\boldsymbol{\tau}_k^i) = \boldsymbol{feature}_i$。再将 $\boldsymbol{feature}_i$ 作为任务 i 的动力

学权重生成器的唯一输入,获得一组网络权重,即 $\boldsymbol{G}_{\mathrm{dyn}}(\boldsymbol{feature}_i)=\boldsymbol{\theta}_i^{\mathrm{dyn}}$。这些网络权重 $\boldsymbol{\theta}_i^{\mathrm{dyn}}$ 可导入不可学习的动力学模型 $\boldsymbol{f}_{\theta_i^{\mathrm{dyn}}}$ 中,并根据当前状态 $\boldsymbol{s}_t^i$ 和当前动作 $\boldsymbol{a}_t^i$ 预测智能体在当前任务 i 中获得的下一时间步状态 $\hat{\boldsymbol{s}}_{t+1}^i$,即 $\boldsymbol{f}_{\theta_i^{\mathrm{dyn}}}(\boldsymbol{s}_t^i,\boldsymbol{a}_t^i)=\hat{\boldsymbol{s}}_{t+1}^i$。本文采用一个动力学权重生成网络而不是动力学模型来辅助训练特征提取器的原因是:生成器把任务特征当作唯一输入,而动力学模型只将任务特征作为部分输入,前者能提供比后者更强有力的训练约束。

(1) 网络结构设计。以轨迹数据为输入的特征提取器采用了长短期记忆(long short-term memory, LSTM)结构[27],这是一种适合处理时序信息的循环神经网络。又因为轨迹数据中包含的状态信息有位姿、速度等非图像数据,所以动力学权重生成模型网络和动力学模型均采用全连接网络实现。通常,应对复杂控制的动力学模型包含多个神经网络层,且每层需要数百个神经元来获取足够的表达能力。大量模型权重直接从权重生成器输出会引发生成器网络规模和计算压力过大的问题。为了避免该问题,本文受文献[28]启发,设计了一个可利用少量网络参数实现复杂动力学的权重生成器结构。

该生成器结构细节如下:设 $\boldsymbol{\theta}_{l_j}^{\mathrm{dyn}}$ 是不可学习的任务 i 动力学模型权重 $\boldsymbol{\theta}_i^{\mathrm{dyn}}$ 中第 l_j 层网络权重。$\boldsymbol{\theta}_{l_j}^{\mathrm{dyn}}$ 包含一个权重矩阵 $\boldsymbol{\theta}_{l_j}^{\mathrm{weight}}$ 和一个偏置矩阵 $\boldsymbol{\theta}_{l_j}^{\mathrm{bias}}$。设该层网络的输入、输出尺寸分别为 $H_{l_j}^{\mathrm{input}}$ 和 $H_{l_j}^{\mathrm{output}}$,则权重矩阵 $\boldsymbol{\theta}_{l_j}^{\mathrm{weight}}$ 大小为 $H_{l_j}^{\mathrm{input}}\times H_{l_j}^{\mathrm{output}}$,偏置矩阵 $\boldsymbol{\theta}_{l_j}^{\mathrm{bias}}$ 大小为 $1\times H_{l_j}^{\mathrm{output}}$。生成器负责输出 2 个小权重矩阵 $\boldsymbol{LG}_{\mathrm{dyn}}^{l_j}$ 和 $\boldsymbol{RG}_{\mathrm{dyn}}^{l_j}$,$\boldsymbol{\theta}_{l_j}^{\mathrm{weight}}$ 由 $\boldsymbol{LG}_{\mathrm{dyn}}^{l_j}$ 和 $\boldsymbol{RG}_{\mathrm{dyn}}^{l_j}$ 进行矩阵相乘实现。设 U 是控制 $\boldsymbol{LG}_{\mathrm{dyn}}^{l_j}$ 和 $\boldsymbol{RG}_{\mathrm{dyn}}^{l_j}$ 大小的参数,$U<<H_{l_j}^{\mathrm{input}}$ 且 $U<<H_{l_j}^{\mathrm{output}}$,$\boldsymbol{LG}_{\mathrm{dyn}}^{l_j}$ 大小为 $H_{l_j}^{\mathrm{input}}\times U$,$\boldsymbol{RG}_{\mathrm{dyn}}^{l_j}$ 大小为 $U\times H_{l_j}^{\mathrm{output}}$。

因此,动力学权重生成模型利用 $[U\times(H_{l_j}^{\mathrm{input}}+H_{l_j}^{\mathrm{output}})+H_{l_j}^{\mathrm{output}}]$ 个网络权重获得了动力学模型第 l_j 层的 $(H_{l_j}^{\mathrm{input}}\times H_{l_j}^{\mathrm{output}}+H_{l_j}^{\mathrm{output}})$ 个网络权重。该设计大幅降低了需要学习的网络规模,为复杂控制动力学描述提供了一条高效的实现思路。

(2)第 1 训练阶段的损失函数。权重生成器的训练目标是从特征提取器输出的任务特征中获得一组精准的动力学模型权重。这促使特征提取器聚焦轨迹数据中的车辆-道路动力学信息。因此,特征提取器和权重发生器的网络通过最小化动力学损失函数优化。该损失函数计算了下一时间步状态真值和由动力学模型计算的下一时刻状态预测值之间的加权均方误差,如式(1)所示。

$$L_{\mathrm{dyn}}=\frac{1}{M}\sum_{m=1}^{M}\sum_{n=1}^{N}b_n(\boldsymbol{s}_{t+1}^{imn}-\boldsymbol{f}_{\theta_i^{\mathrm{dyn}}}^{mn}(\boldsymbol{s}_t^{im},\boldsymbol{a}_t^{im}))^2 \tag{1}$$

其中,M 是每次训练采样数据的批尺寸(batch size),N 是每个状态中包含的传感器类别个数,不同传感器信息的均方误差具有不同的加权参数 b_n,$\boldsymbol{s}_{t+1}^{imn}$ 和 $\boldsymbol{f}_{\theta_i^{\mathrm{dyn}}}^{mn}$ 分别代表着真实状态和预测状态中的第 n 个传感器信息。

此外,特征提取器 E 的优化还需要一项额外的损失函数 L_E 来确保 $\boldsymbol{feature}_i$ 对任务 i 来说是唯一的,如式(2)所示。

$$\begin{aligned}L_E=&\frac{1}{M}\sum_{m=1}^{M}(E(\boldsymbol{\tau}_k^{im1})-E(\boldsymbol{\tau}_k^{im2}))^2\\&+\frac{1}{M/2}\sum_{m=1}^{M/2}\sum_{q=1}^{2}(E(\boldsymbol{\tau}_k^{imq})-E(\boldsymbol{\tau}_k^{i(m+\frac{M}{2})q}))\end{aligned} \tag{2}$$

其中,每次针对任务 i 的训练需要采样 2 个批尺寸均为 M 的轨迹数据批,为保证 $M/2$ 为整数,M 设为偶数;$\boldsymbol{\tau}_k^{im1}$ 和 $\boldsymbol{\tau}_k^{im2}$ 分别表示来自这 2 个轨迹数据批中的第 m 个数据;$q\in\{1,2\}$ 代表着这 2 个轨迹数据批中的第 q 批数据。该损失函数约束了特征提取器从不同轨迹数据批中学到相同的任务特征。

2.4 第 2 训练阶段:策略持续学习

图 1(b)描述了本文算法的第 2 训练阶段。本阶段对 RL 优化损失进行了 2 项持续学习设计。这里采用基于最大熵原理的软演员-评论家(soft actor critic, SAC)算法[29-30]作为本文的 RL 实现基础。SAC 算法在最大化折扣奖励期望的同时最大化策略的熵 H,如式(3)所示,令策略具备适应环境变化的鲁棒性。

$$\boldsymbol{\pi}^*=\arg\max_{\boldsymbol{\pi}}\mathbb{E}_{(\boldsymbol{s}_t,\boldsymbol{a}_t)\sim D}[\sum_{t=0}^{\infty}\gamma^t(r_t+\alpha H(\boldsymbol{\pi}(\boldsymbol{s}_t)))] \tag{3}$$

其中,γ 是折扣因子,$\boldsymbol{\pi}^*$ 是学到的最优策略,重放

缓冲器 D 用于存放赛车与当前环境的交互数据 $(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{s}_{t+1},r_t)$,$\mathbb{E}_{(\boldsymbol{s}_t,\boldsymbol{a}_t)\sim D}$ 是求期望符号,α 是影响最优策略随机性的温度因子。

在具体实现上述目标函数时,SAC 算法采用演员-评论家(actor-critic)结构,包括 1 个策略网络(policy network)和 2 个 Q 值网络(Q-value function)。策略网络是根据当前状态 $\boldsymbol{s}_t$ 生成当前动作 $\boldsymbol{a}_t$ 的演员,即 $\boldsymbol{a}_t = \pi(\boldsymbol{s}_t)$。2 个 Q 值网络 $Q_1(\boldsymbol{s}_t,\boldsymbol{a}_t)$ 和 $Q_2(\boldsymbol{s}_t,\boldsymbol{a}_t)$ 是评估生成策略质量的评论家。Q 值网络先利用累计折扣奖励值进行训练,再来指导策略网络的优化。由于 SAC 算法有 2 个 Q 值函数,每次网络更新均采用两者的最小值参与计算。这种双值函数做法[31]缓解了由值函数过度估计偏见导致的策略性能下降。

需要注意的是,现有 CRL 策略输入往往包含当前状态 $\boldsymbol{s}_t$ 和任务标识 i 的嵌入向量(embedding vector)2 部分信息。本文将任务特征 $\boldsymbol{feature}_i$ 代替现有方法中使用的任务嵌入向量,即 $\boldsymbol{a}_t = \pi_i(\boldsymbol{s}_t^i, \boldsymbol{feature}_i)$,$Q_1(\boldsymbol{s}_t^i, \boldsymbol{feature}_i, \boldsymbol{a}_t^i)$ 和 $Q_2(\boldsymbol{s}_t^i, \boldsymbol{feature}_i, \boldsymbol{a}_t^i)$。

(1) 融入任务相似性的权重正则化损失。当学完任务 $i-1$ 后获得的最优策略 π_{i-1}^* 遇到新任务 i 时,先从本文算法第 1 训练阶段获得任务特征 $\boldsymbol{feature}_i$。接着,该特征向量参与到策略网络 π_i 在新任务 i 上的优化过程中。策略网络的损失函数如下所示。

$$L_{\pi_i} = L_{\text{task}\,i} + L_{\text{regz}} \tag{4}$$

其中,$L_{\text{task}\,i}$ 代表原始 SAC 策略损失项,如式(5)所示。其具体的推导细节可参考文献[30]。

$$L_{\text{task}\,i} = \mathbb{E}_{s_t\sim D}[\alpha\log(\pi_i(\boldsymbol{s}_t^i, \boldsymbol{feature}_i)) - Q_1^i(\boldsymbol{s}_t^i, \boldsymbol{feature}_i, \pi_i(\boldsymbol{s}_t^i, \boldsymbol{feature}_i))] \tag{5}$$

式(5)中的 L_{regz} 是本文设计的融入任务相似性的权重正则化项,其表达如式(6)所示。

$$L_{\text{regz}} = \frac{\lambda}{2}\Big[\sum_{c=1}^{i-1}(1-Sim_c^i)\cdot\boldsymbol{F}_c\Big]\cdot(\boldsymbol{\theta}_{\pi_i}-\boldsymbol{\theta}_{\pi_{i-1}}^*)^2 \tag{6}$$

其中,超参数 λ 用于平衡 $L_{\text{task}i}$ 和 L_{regz} 对策略优化的影响程度;$Sim_c^i \in [-1,1]$ 表示新任务特征向量 $\boldsymbol{feature}_i$ 和某一旧任务特征向量 $\boldsymbol{feature}_c$($c \in \{1,\cdots,i-1\}$)间的相似性;$\boldsymbol{F}_c$ 指 Fisher 信息矩阵,由刚学习完旧任务 c 时的策略 π_c^* 对每个网络权重计算梯度的平方得到[20],用于表达策略网络中的每个权重对旧任务的重要性;$\boldsymbol{\theta}_{\pi_i}$ 是任务 i 学习期间的策略网络权重;$\boldsymbol{\theta}_{\pi_{i-1}}^*$ 是学完任务 $i-1$ 时的最优策略权重。本文采用的相似性量度是广泛用于计算向量相似性关系的夹角余弦值。

权重正则化损失项 L_{regz} 受启发于 online EWC 方法[22-23]中将旧任务似然估计的高斯近似在网络最新的最大后验参数 $\boldsymbol{\theta}_{\pi_{i-1}}^*$ 处重定位的做法。这意味着策略在持续学习中只需保留最近一次的最优策略权重以及不同旧任务 Fisher 信息的累加结果。但 online EWC 只使用一个固定的降权参数来折算所有旧任务的 Fisher 信息和,这使策略更倾向于逐渐遗忘远离新任务的旧任务。与此不同,本文将任务相似性和旧任务的 Fisher 信息相结合,令其成为新任务学习期间对策略网络权重更新幅度的自适应控制器。

为了获得最优策略 π_i^*,其训练过程需要最小化 L_{π_i},从而权重正则化损失项 L_{regz} 也被期望越小越好。当一个旧任务 c 与新任务 i 相似度高时,Sim_c^i 变大,式(6)的 $[(1-Sim_c^i)\cdot\boldsymbol{F}_c]$ 变小,则限制那些对旧任务重要的网络权重参与新任务更新的约束减弱,$(\boldsymbol{\theta}_{\pi_i}-\boldsymbol{\theta}_{\pi_{i-1}}^*)^2$ 的变化范围变大,给予两任务间进行积极前向或后向迁移的机会。相反,与当前新任务 i 不相似的旧任务 c 拥有更小的 Sim_c^i,则 $[(1-Sim_c^i)\cdot\boldsymbol{F}_c]$ 变大,为了达到最小化 L_{regz} 的目的,$(\boldsymbol{\theta}_{\pi_i}-\boldsymbol{\theta}_{\pi_{i-1}}^*)^2$ 应尽量小,令对旧任务重要的部分策略权重在新任务学习期间保持不变,避免灾难性遗忘问题。

(2) 鼓励持续学习的奖励函数。奖励函数会直接影响 Q 值函数是否能公正地评价当前 RL 策略,对策略的后续收敛起到重要的指导作用。面对新任务 i 时,每一时间步 t 的奖励函数 r_t^i 包含 2 项内容:针对新任务 i 的奖励项 r_t^{new} 和针对所有旧任务的奖励项 r_t^{old},如式(7)所示,κ_{new} 和 κ_{old} 是 2 个经验参数。

$$r_t^i = \kappa_{\text{new}}\cdot r_t^{\text{new}} + \kappa_{\text{old}}\cdot r_t^{\text{old}} \tag{7}$$

为了计算式(7),这里定义一个可评估策略在单个任务上表现好坏的函数 $f_{\text{reward}}(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{s}_{t+1})$,如式(8)所示,该公式在文献[32]中被提出。

$$f_{\text{reward}}(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{s}_{t+1}) = \Delta l_t(\cos\psi_{t+1} - |\sin\psi_{t+1}| - |\Delta dis_{t+1}|) \tag{8}$$

其中,Δl_t 表示赛车在时间步 t 和 $t+1$ 之间行驶的距离;ψ_{t+1} 是 $\boldsymbol{s}_{t+1}$ 中赛车航向和道路轴线间的夹角;Δdis_{t+1} 是 $\boldsymbol{s}_{t+1}$ 中车辆位置和道路轴线的间距。

在本文提出的奖励函数 r_t^i 中,负责新任务 i 的奖励项表达为 $r_t^{\text{new}} = f_{\text{reward}}(\boldsymbol{s}_t^i,\boldsymbol{a}_t^i,\boldsymbol{s}_{t+1}^i)$,负责旧任务的奖励项 r_t^{old} 表达为式(9)。

$$r_t^{\text{old}} = \frac{1}{i-1}\sum_{c=1}^{i-1}\Delta\hat{l}_t^{i\to c}\cdot f_{\text{norm}}(f_{\text{reward}}(\boldsymbol{s}_t^i,\boldsymbol{a}_t^{i\to c},\hat{\boldsymbol{s}}_{t+1}^{i\to c}) - f_{\text{reward}}(\boldsymbol{s}_t^i,\boldsymbol{a}_t^{(i-1)^*\to c},\hat{\boldsymbol{s}}_{t+1}^{(i-1)^*\to c})) \tag{9}$$

其中,$\boldsymbol{a}_t^{i\to c} = \pi_i(\boldsymbol{s}_t^i,\boldsymbol{feature}_c)$ 是根据当前策略 π_i 计算出的适合旧任务 c 的当前动作;$\hat{\boldsymbol{s}}_{t+1}^{i\to c} = f_{\theta_c^{\text{dyn}}}(\boldsymbol{s}_t^i,\boldsymbol{a}_t^{i\to c})$ 是根据当前策略 π_i 预测到的旧任务 c 下一时间步状态;$f_{\theta_c^{\text{dyn}}}$ 是任务 c 的动力学模型;$\Delta\hat{l}_t^{i\to c}$ 是赛车从 $\boldsymbol{s}_t^i$ 到 $\hat{\boldsymbol{s}}_{t+1}^{i\to c}$ 的行驶距离;$\boldsymbol{a}_t^{(i-1)^*\to c} = \pi_{i-1}^*(\boldsymbol{s}_t^i,\boldsymbol{feature}_c)$ 是根据学习新任务 i 前的最优策略 π_{i-1}^* 计算出的适合旧任务 c 的当前动作;$\hat{\boldsymbol{s}}_{t+1}^{(i-1)^*\to c} = f_{\theta_c^{\text{dyn}}}(\boldsymbol{s}_t^i,\boldsymbol{a}_t^{(i-1)^*\to c})$ 是根据上一次最优策略 π_{i-1}^* 预测到的任务 c 下一时刻状态;归一化函数 $f_{\text{norm}}(x) = \tanh(x/\rho)$ 用于限制 π_i 和 π_{i-1}^* 在相同任务上的奖励差值范围,ρ 是一个缩放因子,在本文中经验取值为 10。式(9)中计算所有旧任务 $\Delta\hat{l}_t^{i\to c}$ 和 f_{norm} 乘积的均值是为了令 r_t^{old} 和 r_t^{new} 奖励项的大小相当。综上所述,奖励项 r_t^{old} 的含义是鼓励赛车寻找令当前策略在旧任务上的表现优于上一最优策略的优化方向。

再将获得的总奖励 r_t^i 代入到 2 个 Q 值网络的损失函数 $L_{Q_{z\in\{1,2\}}^i}$ 中,如式(10)所示。

$$L_{Q_{z\in\{1,2\}}^i} = \mathbb{E}_{(\boldsymbol{s}_t^i,\boldsymbol{a}_t^i)\sim D}\left[\frac{1}{2}\cdot(Q_{z\in\{1,2\}}^i(\boldsymbol{s}_t^i,\boldsymbol{feature}_i,\boldsymbol{a}_t^i) - y)^2\right] \tag{10}$$

式中 y 通过式(11)来表达。

$$y = \mathbb{E}_{(\boldsymbol{s}_t^i,\boldsymbol{a}_t^i,\boldsymbol{s}_{t+1}^i,r_t^i,\gamma)\sim D}[r_t^i + \gamma\cdot(\min_{z\in\{1,2\}} Q_{z_{\text{targ}}}^i(\boldsymbol{s}_{t+1}^i,\boldsymbol{feature}_i,\pi_i(\boldsymbol{s}_{t+1}^i,\boldsymbol{feature}_i)) - \alpha\log\pi_i(\boldsymbol{s}_{t+1}^i,\boldsymbol{feature}_i))] \tag{11}$$

式中,α 是 SAC 算法的温度因子;$Q_{z_{\text{targ}}}^i$ $(z\in\{1,2\})$ 是对应 2 个 Q 值网络的目标网络(target network),用于稳定 Q 值网络训练,如式(12)所示。

$$\boldsymbol{\theta}_{Q_{z_{\text{targ}}}^i} = \sigma\boldsymbol{\theta}_{Q_z^i} + (1-\sigma)\boldsymbol{\theta}_{Q_{z_{\text{targ}}}^i},\ z\in\{1,2\} \tag{12}$$

其中,缩放因子 $\sigma\in[0,1]$,$\boldsymbol{\theta}_{Q_{z_{\text{targ}}}^i}$ 和 $\boldsymbol{\theta}_{Q_z^i}$ 分别表示 Q 值函数和对应的目标函数的网络权重。

在完成每次策略和 Q 值网络训练时,温度因子 α 通过最小化式(13)来调节,H_0 为目标熵预设值。

$$L_\alpha = \mathbb{E}_{\pi_i}[-\alpha\log\pi_i(\boldsymbol{s}_t^i,\boldsymbol{feature}_i) - \alpha H_0] \tag{13}$$

最后,本文算法的伪代码总结如算法 1 所示。

算法 1　本文提出的两阶段 CRL 算法

输入:新任务环境 i,$i\geqslant 1$,收集随机轨迹数据
输出:当前最优 CRL 策略 π_i

for 当前新任务 $i = 1,2,3,\cdots$ **do**
　/* 进入算法的第 1 训练阶段:提取任务特征 */
　初始化特征提取器 E;
　初始化动力学权重生成器 G_{dyn};
　for 每个训练轮次(epoch) **do**
　　for 每轮中的迭代次数(iteraction) **do**
　　从轨迹数据中采样轨迹批;
　　根据式(1)和(2)更新 E 和 G_{dyn};
　　end for
　end for
　从 E 获得新任务特征 $\boldsymbol{feature}_i$;
　从 G_{dyn} 获得新任务动力学模型权重 $\boldsymbol{\theta}_i^{\text{dyn}}$;
　/* 进入算法的第 2 训练阶段:策略持续学习 */
　if $i = 1$ **then** $L_{\text{regz}} = 0$,$r_t^{\text{old}} = 0$;
　else 计算新、旧任务间的相似性;
　end if
　初始化重放缓冲器 D、策略网络 π_i;
　初始化 2 个 Q 值函数 Q_1^i 和 Q_2^i;
　初始化 2 个目标 Q 值函数 $Q_{1_{\text{targ}}}^i$ 和 $Q_{2_{\text{targ}}}^i$;
　for 每个 RL 训练回合(episode) **do**
　　for 每回合中的时间步 t **do**
　　$\boldsymbol{a}_t^i = \pi_i(\boldsymbol{s}_t^i,\boldsymbol{feature}_i)$;
　　在新任务 i 中执行 $\boldsymbol{a}_t^i$ 后 $\boldsymbol{s}_t^i\to\boldsymbol{s}_{t+1}^i$;
　　利用式(7)计算当前奖励 r_t^i;
　　将 $(\boldsymbol{s}_t^i,\boldsymbol{a}_t^i,\boldsymbol{s}_{t+1}^i,r_t^i)$ 存储至 D 中;
　　从 D 中采样一个数据批;
　　最小化式(10)优化 Q_1^i 和 Q_2^i;

```
        最小化式(4)优化 π_i;
        最小化式(13)调节温度参数 α;
        根据式(12)更新 Q^i_{1_targ} 和 Q^i_{2_targ};
    end for
  end for
  计算策略 π_i^* 权重对任务 i 的重要性 F_i;
  将其用于后续新任务的策略优化;
end for
```

3 实验设置及结果分析

3.1 实验设置

本文利用 3D 逼真赛车仿真平台(TORCS)[33]设计了一系列测试实验。TORCS 赛车仿真平台提供了具有多种道路形状和表面材质的道路选项。本文利用 5 个各具特色的 TORCS 道路场景参与实验,设计了 2 条不同的任务序列,如表 1 和图 2 所示。相较于道路 1 和 2,道路 3、4 和 5 难度更大。尤其是道路4和5,两者单独利用传统RL训练时,均无法像其他 3 条道路一样在有限训练回合内收敛。因此,任务序列 1 反映了一个从易到难的持续学习过程,任务序列 2 则是一个从难到易的持续学习过程。

(1)基准方法。本文选择现有 3 类 CRL 方法中具有代表性的方法作为测试基准。经验回放基准方法采用选择性经验回放(selective experience replay, SER)方法[11]、多时间尺度经验回放(multi-timescale replay, MTR)方法[13]以及 A-GEM 方法[17]。参数独立基准方法采用 PNN 方法[18-19]。权重正则化基准方法采用 EWC 方法[20]、online EWC (后续简称为 OEWC)方法[22-23]和 P&C 方法[23]。此外,本文还设计了一个精调(fine tuning, FT)基准方法,用于展示无持续学习设计的 RL 算法在顺序学习多任务时的表现。在 FT 方法中,每个新任务的训练利用前一个任务的最优策略初始化网络,再遵循原始 SAC 算法更新。为保证比较的公平性,所有方法采用相同的策略网络结构且均以 SAC 算法为 CRL 的实现基础。基准方法中输入策略的任务标识嵌入向量维度与本文算法中代替嵌入向量输入策略的任务特征维度相同。

表 1 所选道路的特征描述

道路名称	道路形状描述	长度/m	路面材质描述	
			材质	物理参数[a]
道路 1	路面平坦,多种弯道	3 260.43	柏油路	[1.200 0, 0.001 0, 0.000 0, 1.000 0]
道路 2	路面单向倾斜,单一弯道	3 703.83	混凝土路	[1.100 0, 0.001 5, 0.500 0, 1.000 0]
道路 3	路面凹凸不平,急弯,上下坡	2 205.93	泥土路	[0.850 0, 0.005 0, 0.020 0, 30.000 0]
道路 4	路面破损,部分路面倾斜,多种弯道	6 282.81	(带补丁的)开裂柏油路	[0.800 0(0.950 0), 0.003 5, 1.000 0, 5.000 0(20.000 0)][b]
道路 5	路面波浪形起伏,发夹弯,上下坡	1 005.58	沙地	[0.900 0, 0.006 0, 0.040 0, 8.000 0]

a 用于描述路面材质的物理参数有[摩擦系数,滚动阻力系数,粗糙度,粗糙度波长]

b 括号里的值对应路面补丁和主体路面不同的物理参数值

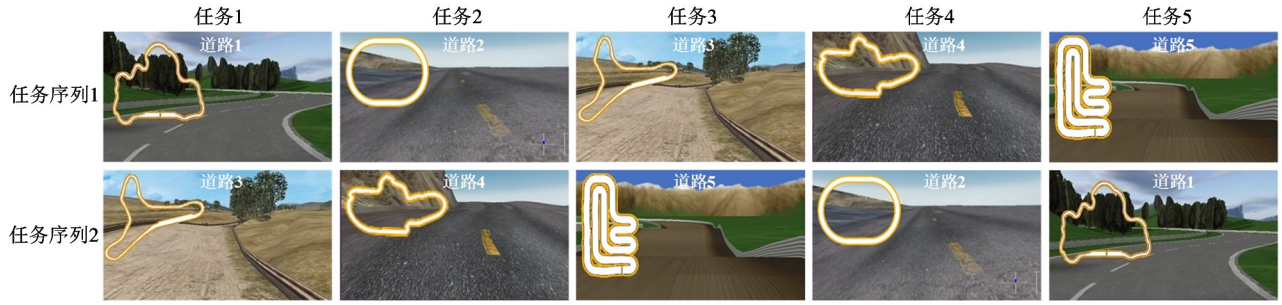

图 2 参与本文实验的 TORCS 道路环境及任务序列展示

(2)消融实验。首先,讨论第 1 训练阶段中任务特征维度 Dim 值和动力学权重生成网络的小矩阵参数 U 值的不同对动力学预测效果的影响。接着,检验第 2 训练阶段中两项持续学习设计的贡献。该阶段的消融实验利用被命名为 Oldr 和 SimEWC 的 2 个方法进行性能分析。其中,Oldr 方法指本文算法去掉权重正则化损失、保留鼓励持续学习的奖励函数。SimEWC 方法指本文算法保留权重正则化损失、去掉鼓励持续学习的奖励函数。

(3)本文算法实现细节。特征提取器的 LSTM 网络有 2 层且每层 100 个神经元,其输入 τ_k 的时间间隔 k 为 4,输出的任务特征维度 Dim 为 10。动力学权重生成网络的 U 值为 32,对应不可学习的动力学模型是一个 2 层且每层 256 个神经元的全连接网络。第 1 训练阶段采用 Adam 优化器[34],动力学损失 L_{dyn} 学习率为 0.001 0,特征提取器的额外损失 L_{E} 学习率为 0.005 0。SAC 策略网络和评论家网络均采用尺寸为(400,300) 的两层全连接网络以及学习率设为 0.000 1 的 Adam 优化器。折扣因子 γ 和缩放因子 σ 分别取 0.990 和 0.995。奖励函数的 κ_{new} 和 κ_{old} 均取 0.5。目标熵预设值 H_0 为动作维度的负值。最大训练回合数和每一个回合的最大步数分别是 3 000和 2 000。本文中所有运算均通过 NVIDIA GTX 1080Ti GPU 实现。

3.2 评价指标定义

本文定义了一系列评价 CRL 方法的性能指标。

(1)成功率:在 100 次测试回合中,最终策略通过全部任务的回合数占比。该值越高,算法越好。

(2)平均性能(average performance,AP):如式(14)所示,Y 表示总任务数,$ap_{Y,y}$ 是最终策略在任务 $y(y \leqslant Y)$ 上的性能表现。本文针对自动驾驶赛车问题的性能表现具体分为驾驶速度和稳定性两部分。评估速度时,$ap_{Y,y}$ 是计算 π_Y 在任务 y 上所有成功测试回合的平均速度,越大越好。评估稳定性时,$ap_{Y,y}$ 是计算 π_Y 在任务 y 上所有成功测试回合的平均车辆-道路轴线夹角绝对值,越小越好。若策略无法在一个任务上驾驶成功,根据仿真平台设置,ap 的速度和稳定性部分被分别赋值为0 km · h^{-1}和 21 °。

$$AP = \frac{1}{Y}\sum_{y=1}^{Y} ap_{Y,y} \tag{14}$$

(3)后向迁移(backward transfer,BWT):如式(15)所示,$ap_{y,y}$ 表示刚学完任务 y 时的策略 π_y 在该任务上的性能表现。该指标评估了策略在学习新任务后对旧任务性能的影响。积极的 BWT 结果表达为速度部分大于 0,且越大越好;稳定性部分小于 0,且越小越好。明显消极的 BWT 结果表示灾难性遗忘。

$$BWT = \frac{1}{Y-1}\sum_{y=1}^{Y-1} ap_{Y,y} - ap_{y,y} \tag{15}$$

(4)前向迁移(forward transfer,FWT):如式(16)所示,$ap_{1,y}$ 是只学习一个任务 y 时的策略性能。该指标评估了策略的旧任务知识对新任务学习产生的影响。积极的 FWT 结果表达为速度部分大于 0,且越大越好;稳定性部分小于 0,且越小越好。

$$FWT = \frac{1}{Y-1}\sum_{y=2}^{Y} ap_{y,y} - ap_{1,y} \tag{16}$$

(5)归一化策略容量(normalized policy capacity, NPC):最终策略与学习首个任务时策略在网络容量方面的比值,$NPC \geqslant 1$。该值越大,网络扩张越快。

(6)归一化重放缓冲器(normalized replay buffer, NRB):CRL 方法与传统 RL 方法在重放缓冲器尺寸方面的比值,$NRB \geqslant 1$。该值越大,数据需求越大。

(7)单步平均奖励:策略先根据式(8)计算针对单一任务的每回合奖励总和,再除以回合的总步数。

(8)精度的相对倍数变化(relative fold change in accuracy, RFCA):该指标用于分析算法第 1 训练阶段的消融实验结果。如式(17)所示。RFCA 结果越高,说明对 Dim 或 U 候选值的选择越合适。

$$RFCA = \frac{\text{采用不同 } Dim \text{ 或 } U \text{ 候选值计算的动力学预测精度}}{\text{最小 } Dim \text{ 或 } U \text{ 候选值计算的动力学预测精度}} \div \left(\frac{\text{采用不同 } Dim \text{ 或 } U \text{ 候选值时的训练总次数}}{\text{最小 } Dim \text{ 或 } U \text{ 候选值时的训练总次数}}\right) \tag{17}$$

3.3 与基准方法的对比结果

表 2 和 3 分别展示了 2 个任务序列中本文算法和基准方法的测试结果。图 3 和 4 分别细致展示了 2 个任务序列中不同方法在各个任务训练期间的单步平均奖励变化。图 3 中,由于每个任务具有2 000个训

练回合,因此整个持续学习过程一共有10 000个回合。图中平行于纵坐标的4条虚线代表任务的切换。图中沿着纵坐标标注的“任务1”至“任务5”记录了各个任务在整个策略学习期间分别作为新任务和旧任务时的性能表现。所有基准方法在学习第一个任务期间没有区别,因此用一条命名为“Allbaselines”的

表2　任务序列1中本文算法和基准方法的性能比较

方法	*AP*		*BWT*		*FWT*		成功率/%	*NPC*	*NRB*
	速度/(km·h^{-1})	稳定性/°	速度/(km·h^{-1})	稳定性/°	速度/(km·h^{-1})	稳定性/°			
FT	5.42	17.49	−119.68	18.14	**41.97**	−9.58	0	1	1
SER	87.94	4.13	−7.39	0.60	31.12	−8.55	59	1	2
MTR	86.86	3.87	−7.76	0.38	30.17	−8.92	55	1	2
A-GEM	89.95	3.64	0.91	**−0.11**	27.72	−8.65	62	1	2
PNN	**99.23**	2.65	0.00	0.00	36.80	−9.21	68	5	1
EWC	91.74	3.05	−0.80	0.10	28.88	−9.01	76	1	1
OEWC	93.52	2.80	−2.07	0.19	29.34	−9.04	70	1	1
P&C	80.77	3.32	−5.44	0.36	16.51	−9.21	54	1	1
本文算法	99.09	**2.48**	**3.09**	−0.04	32.46	**−10.04**	**78**	1	1

表3　任务序列2中本文算法和基准方法的性能比较

方法	*AP*		*BWT*		*FWT*		成功率/%	*NPC*	*NRB*
	速度/(km·h^{-1})	稳定性/°	速度/(km·h^{-1})	稳定性/°	速度/(km·h^{-1})	稳定性/°			
FT	25.78	17.57	−89.17	17.63	35.09	**−13.53**	0	1	1
SER	73.22	3.79	−6.57	0.75	12.49	−8.43	56	1	2
MTR	70.82	4.55	−5.70	0.30	9.22	−7.69	68	1	2
A-GEM	90.93	3.62	0.99	**−0.14**	28.94	−8.58	71	1	2
PNN	96.95	**2.73**	0.00	0.00	34.55	−9.54	72	5	1
EWC	89.93	3.73	−1.66	0.34	28.15	−8.56	70	1	1
OEWC	86.89	3.29	−2.34	0.42	23.28	−9.02	75	1	1
P&C	79.96	3.63	−9.47	0.93	22.78	−9.47	55	1	1
本文算法	**98.64**	3.34	**1.43**	**−0.14**	**37.47**	−8.96	**80**	1	1

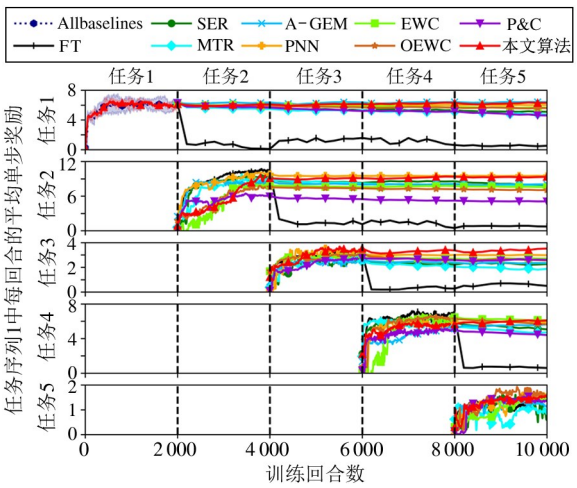

图3　本文算法和基准方法在任务序列1中的整个持续学习过程的奖励变化曲线

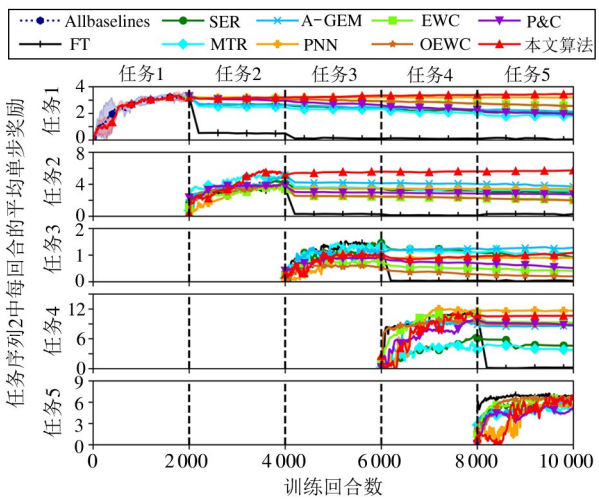

图4　本文算法和基准方法在任务序列2中的整个持续学习过程的奖励变化曲线

奖励曲线统一表示。为了清晰展示多种方法的奖励曲线变化趋势,该图只在任务 1的训练阶段展示出奖励曲线的置信区间,即 All Baseline 曲线和本文算法曲线的阴影区域。当一个任务是新任务时,所有方法的奖励曲线记录了起始2 000 个训练回合的全部奖励数据。当该任务变成旧任务时,改为每间隔 200 个训练回合测试并记录此时刻的旧任务奖励。后续奖励曲线变化图与该图的设置保持一致。根据这些图表内容,本节从以下 5 个不同角度进行性能分析。

(1)FT 方法在 2 个任务序列上成功率均为零,且 FWT 和 BWT 结果均明显消极。这验证了传统 RL 策略在面对多任务时普遍存在的灾难性遗忘问题,其凸显了 CRL 研究的必要性。

(2)图 3 和 4 中,本文算法在策略训练第一个任务时的奖励曲线比基准方法具有更窄的置信区间,即训练波动更小。由于此时训练不涉及任何旧任务,该对比结果体现了本文采用的任务特征方式比基准方法中普遍使用的嵌入向量方式提供了更丰富的任务信息,令策略收敛过程更平稳。

(3)本文算法在 2 个任务序列中均收获了较好的成绩。在成功率方面,本文算法的结果高于所有基准方法。在驾驶性能方面,本文算法在既不存储旧任务数据又不扩张网络的情况下,同时满足了所有 CRL 能力。这是其他基准方法未能做到的。从表2 和表3 中可以看出:PNN 在 AP 和 FWT 指标上的表现是基准 CRL 方法中最好的;A-GEM 方法的 BWT 结果是基准 CRL 方法中最好的。然而,PNN 方法的 $NPC=5$,即策略网络随着任务增加而扩张。AGEM 方法的 $NRB=2$,即需要额外的旧任务数据存储。本文方法能在 NPC 和 NRB 结果均保持最低的情况下,在任务序列 1 中获得与 PNN 方法最接近的平均速度;并在 AP 和 FWT 结果的稳定性部分以及 BWT 结果的速度部分,比所有基准方法表现更好。在任务序列 2 中,本文算法在 AP 和 FWT 结果的速度部分以及全部 BWT 结果上均表现最佳。

(4)与同属于权重正则化类别的 EWC、OEWC 和 P&C 方法相比,本文算法能在任务序列 1 的所有指标上均获得最好结果,在任务序列 2 的 AP 的平均速度结果、BWT 结果和 FWT 的平均速度上均表现最好。结合上述第 3 点分析可得,本文提出的融入任务相似性的权重正则化方法和鼓励旧任务性能不下降的奖励函数两项设计,确实能帮助策略发挥出更佳的 CRL 性能。同类方法中,P&C 方法表现最差,因为其利用知识蒸馏的模型压缩思想来实现策略网络的不扩张,以部分性能的下降为代价。

(5)对比 2 个任务序列中同一方法性能可知,任务排序的不同会导致最终策略的收敛差异。在任务序列 1 中,排在前面的任务令策略先掌握到速度较高、能应对相对单一道路特征的先验知识。与之相反,策略从任务序列 2 中学到速度较低、可对应相对复杂道路特征的先验知识。任务序列 1 中策略在面对道路特征更加复杂的后续任务时,更倾向获得比任务序列 2 中策略具有更高驾驶速度的收敛点。

3.4　消融实验

下面将分析本文算法在 2 个训练阶段的创新点。

(1) 第 1 训练阶段。本阶段负责提取可描述车辆-道路动力学关系的任务特征,其中起到重要作用的 2 个超参数是任务特征维度 *Dim* 值和动力学权重生成器的小矩阵参数 *U* 值。图5 和图6 分别展示了2 个超参数的不同数值选择对动力学预测效果的影响。实验结果表明,尽管随着 *Dim* 值或 *U* 值取值的增大,动力学预测的精度有所提升,但其精度的提升速度远逊于训练次数的增加速度。因此,两图中的 RFCA 折线都呈现出先升后降的趋势。本文选择两折线的顶点 $Dim = 10$ 和 $U = 32$ 作为两者的最佳选择。

(2) 第 2 训练阶段。本阶段具有融入任务相似性的权重正则化策略损失和鼓励持续学习的RL奖

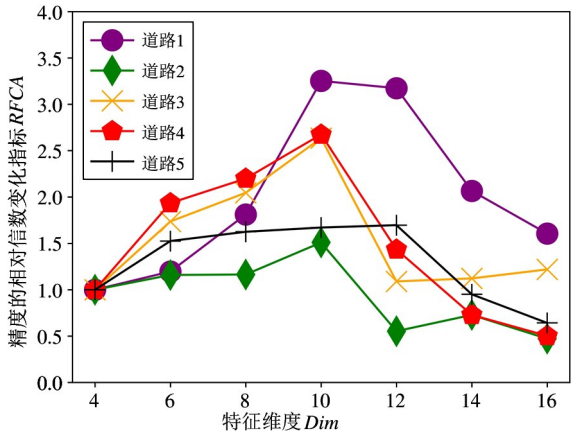

图 5　不同特征维度 *Dim* 对动力学预测效果的影响

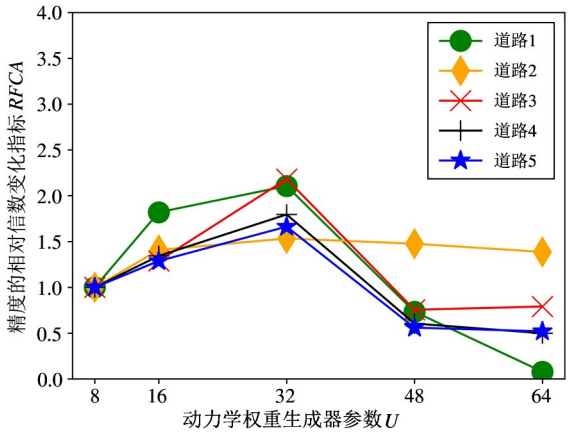

图 6　不同动力学权重生成器参数 U 对动力学预测效果的影响

励函数两项创新设计,分别通过 SimEWC 和 Oldr 方法实现。表 4 和 5 分别展示了这 2 个方法的测试结果。

首先,将表 4、5 分别与展示基准方法性能的表 2、3 进行对比。对比结果表明,在 2 个任务序列中 2 项创新设计均对本文算法的性能提升做出了积极贡献,且两者的结合具有协同作用。从 SimEWC 方法与 EWC、OEWC 基准方法的性能对比可知,SimEWC 方法在 2 个任务序列中的 FWT 和 BWT 结果均比 EWC 和 OEWC 方法更积极。这充分体现了任务相似性的融入对提升传统权重正则化方法在进阶 CRL 能力方面的帮助,使其不再局限于仅仅解决灾难性遗忘问题。

表 4　对应本文算法第 2 阶段的 2 个创新点的 Oldr 和 SimEWC 方法在任务序列 1 中的性能展示

方法	*AP*		*BWT*		*FWT*		成功率/%	*NPC*	*NRB*
	速度/(km·h^{-1})	稳定性/°	速度/(km·h^{-1})	稳定性/°	速度/(km·h^{-1})	稳定性/°			
Oldr	94.05	3.37	3.41	0.09	27.40	−9.14	66	1	1
SimEWC	93.30	2.86	0.19	−0.04	30.86	−9.60	78	1	1

表 5　对应本文算法第 2 阶段的 2 个创新点的 Oldr 和 SimEWC 方法在任务序列 2 中的性能展示

方法	*AP*		*BWT*		*FWT*		成功率/%	*NPC*	*NRB*
	速度/(km·h^{-1})	稳定性/°	速度/(km·h^{-1})	稳定性/°	速度/(km·h^{-1})	稳定性/°			
Oldr	86.25	3.61	1.48	−0.11	20.22	−8.75	70	1	1
SimEWC	92.07	2.84	1.32	−0.20	28.76	−9.51	80	1	1

其次,表 4 与表 5 中相同方法的对比展示了 Oldr 和 SimEWC 方法在任务序列 1 上均获得了比在任务序列 2 上更高的平均速度、FWT 结果以及稍低的成功率。这些结果与 3.2 节的第 5 点分析吻合,即策略在不同任务排序下的收敛存在差异。SimEWC 方法在 2 个任务序列间的平均速度变化幅度比 Oldr 方法小 84%,其平均稳定性变化幅度比 Oldr 方法小 92%。可见,SimEWC 方法对任务排序变化的敏感程度比 Oldr 方法低。再对比这 2 个方法的 BWT 结果发现,Oldr 方法在 2 个任务序列中的 BWT 速度部分结果比 SimEWC 方法表现更好,而 BWT 稳定性部分结果是 SimEWC 方法比 Oldr 方法表现更好。这些结果一方面展现了 Oldr 方法对持续学习的有效性,另一方面也说明了直接对策略网络权重进行自适应正则化的 SimEWC 方法在实现 CRL 能力方面比 Oldr 方法发挥更稳定。而利用旧任务性能升降作为奖惩信号引导 CRL 优化的 Oldr 方法更适于搭配其他持续学习约束一起使用,可促进策略在满足持续学习需要的同时达到更好的收敛点。

最后,本文利用图 7 和图 8 分别展示了本文算法、Oldr 以及 SimEWC 方法在 2 个任务序列中的整个策略训练过程。图 9 可视化了道路特征间的相似性关系,用于配合 SimEWC 训练曲线来共同分析任务相似性对权重正则化损失创新设计的作用。

当遇到与旧任务相似性低的新任务时,策略的更新过程侧重保持已有知识不被遗忘。从图 9 中可以看出,道路 1 和 2、道路 1 和 5 以及道路 2 和 5 之间的相似性得分最低。道路 1 和 2、道路 1 和 5 之间的低相似性可以对应到图 7 中 SimEWC 方法的任务 1 奖励曲线在任务2、任务5的学习期间只是努力保持不

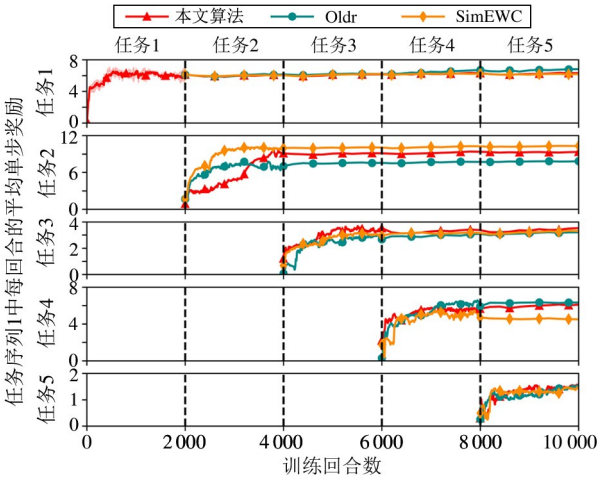

图 7　本文算法、Oldr 和 SimEWC 方法在任务序列 1 中的整个持续学习过程的奖励变化曲线

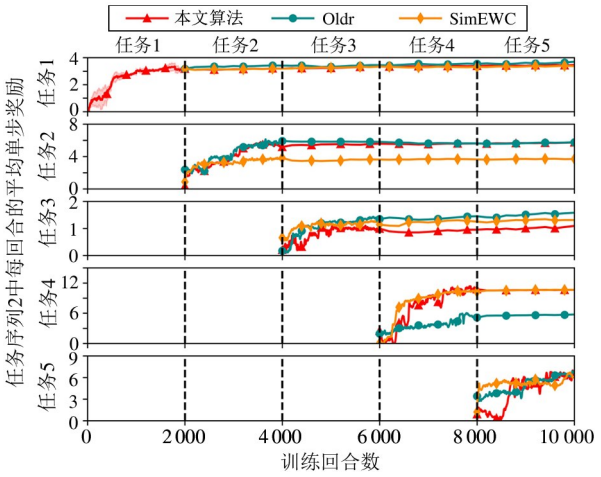

图 8　本文算法、Oldr 和 SimEWC 方法在任务序列 2 中的整个持续学习过程的奖励变化曲线

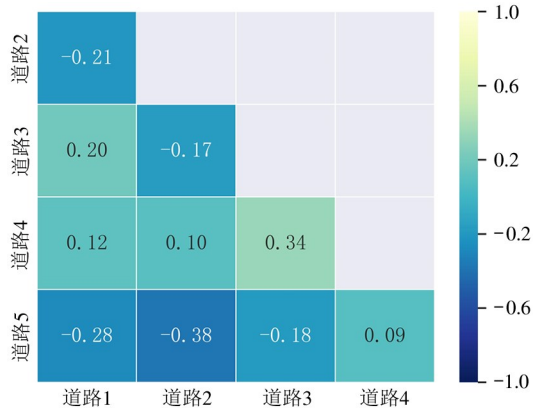

图 9　不同道路特征之间的相似性可视化展示

下降。道路 2 和 5 之间低相似性可对应到图 7 中 SimEWC 方法的任务 2 奖励曲线从任务 4 到任务 5 学习期间的明显回落趋势。上述情况也同样体现在图 8 中 SimEWC 方法的任务 3 奖励曲线在任务 4 的学习期间、任务 4 奖励曲线在任务 5 的学习期间。

当遇到与旧任务相似度高的新任务时，策略的更新过程追求在学习新任务的同时也提升了旧任务的驾驶性能。从图 9 中可以看到，道路 1、3 及 4 这三者间的特征相似性是最高的。其可对应到图 7 中 SimEWC 方法的任务 1 奖励曲线在任务 3、4 的学习期间有明显上升趋势。与此相同的积极后向迁移表现也出现在图 8 中 SimEWC 方法的任务 1 奖励曲线在任务 2 和 5 的学习期间。

4　结　论

本文提出了一个融入任务相似性的 CRL 算法来应对自动驾驶赛车在多变道路上的持续决策问题。该算法包括动力学特征提取方法、融入任务相似性的权重正则化方法以及维护旧任务性能不下降的奖励函数 3 项设计。从而，在无需存储旧任务数据且无需扩展策略网络规模的前提下，该方法显著提高了持续决策算法的前向和后向迁移能力。这是现有方法无法做到的。仿真实验结果表明，本文算法在解决 CRL 问题上比其他基准方法具有更优越的综合表现。在本文工作基础上，未来工作将进一步研究如何避免由任务排序变化引起的策略收敛差异问题。

参考文献

[1] HERMANSDORFER L, BETZ J, LIENKAMP M. Benchmarking of a software stack for autonomous racing against a professional human race driver[C]//Proceedings of International Conference on Ecological Vehicles and Renewable Energies. Monte Carlo, Monaco: IEEE, 2020:1-8.

[2] HARTMANN G, SHILLER Z, AZARIA A. Autonomous head-to-head racing in the Indy Autonomous Challenge Simulation Race[EB/OL]. (2022-10-30)[2022-12-05]. https://arxiv.org/pdf/2109.05455.pdf.

[3] O'KELLY M, SUKHIL V, ABBAS H, et al. F1/10: an open-source autonomous cyber-physical platform[EB/OL]. (2019-01-24)[2022-12-05]. https://arxiv.org/pdf/1901.08567v1.pdf.

[4] KOPPULA S. Learning a CNN-based end-to-end controller

for a formula SAE racecar [EB/OL]. (2017-07-12) [2022-12-05]. https://arxiv.org/pdf/1708.02215.pdf.

[5] SUTTON R S, BARTO A G. Reinforcement learning: an introduction[M]. 2nd edition. Cambridge: MIT Press, 2018:1-2.

[6] JARITZ M, DE CHARETTE R, TOROMANOFF M, et al. End-to-end race driving with deep reinforcement learning [C]// Proceedings of IEEE International Conference on Robotics and Automation. Brisbane, Australia: IEEE, 2018:2070-2075.

[7] ZHU Y, ZHAO D. Driving control with deep and reinforcement learning in the open racing car simulator[C]// Proceedings of International Conference on Neural Information Processing. Siem Reap, Cambodia: Springer, 2018:326-334.

[8] WRC Promoter GmbH. History of FIA world rally championship [EB/OL]. [2022-12-05]. https://www.wrc.com/en/more/wrc-history/wrc-present/.

[9] KHETARPAL K, RIEMER M, RISH I, et al. Towards continual reinforcement learning: a review and perspectives[EB/OL]. (2022-11-11)[2022-12-05]. https://arxiv.org/pdf/2012.13490v2.pdf.

[10] WOŁCZYK M, ZAJAC M, PASCANU R, et al. Continual world: a robotic benchmark for continual reinforcement learning[EB/OL]. (2021-10-28)[2022-12-05]. https://arxiv.org/pdf/2105.10919.pdf.

[11] ISELE D, COSGUN A. Selective experience replay for lifelong learning[C]// Proceedings of AAAI Conference on Artificial Intelligence. New Orleans, USA: AAAI Press, 2018:3302-3309.

[12] ROLNICK D, AHUJA A, SCHWARZ J, et al. Experience replay for continual learning[C]// Proceedings of International Conference on Neural Information Processing Systems. Vancouver, Canada: Curran Associates Inc, 2019:350-360.

[13] KAPLANIS C, CLOPATH C, SHANAHAN M. Continual reinforcement learning with multi-timescale replay [EB/OL]. (2020-04-16)[2022-12-05]. https://arxiv.org/pdf/2004.07530.pdf.

[14] ATKINSON C, MCCANE B, SZYMANSKI L, et al. Pseudo-rehearsal: achieving deep reinforcement learning without catastrophic forgetting [J]. Neurocomputing, 2021,428:291-307.

[15] HUANG Y, XIE K, BHARADHWAJ H, et al. Continual model-based reinforcement learning with hypernetworks [C]// Proceedings of IEEE International Conference on Robotics and Automation. Xi'an, China: IEEE, 2021: 799-805.

[16] LOPEZ-PAZ D, RANZATO M A. Gradient episodic memory for continual learning[C]// Proceedings of International Conference on Neural Information Processing Systems. Long Beach, USA: Curran Associates Inc, 2017: 6470-6479.

[17] CHAUDHRY A, RANZATO M A, ROHRBACH M, et al. Efficient lifelong learning with A-GEM[C]// Proceedings of International Conference on Learning Representations. New Orleans, USA: ICLR, 2019:1-20.

[18] RUSU A A, RABINOWITZ N C, DESJARDINS G, et al. Progressive neural networks [EB/OL]. (2016-09-07) [2022-12-05]. https://arxiv.org/pdf/1606.04671v3.pdf.

[19] RUSU A A, VE ČERÍK M, ROTHÖRL T, et al. Sim-to-real robot learning from pixels with progressive nets[C]// Proceedings of Annual Conference on Robot Learning. Mountain View, USA: CoRL, 2017: 262-270.

[20] KIRKPATRICK J, PASCANU R, RABINOWITZ N, et al. Overcoming catastrophic forgetting in neural networks [J]. Proceedings of the National Academy of Sciences of the United States of America, 2017, 114 (13): 3521-3526.

[21] LOMONACO V, DESAI K, CULURCIELLO E, et al. Continual reinforcement learning in 3D non-stationary environments[C]// Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. Seattle, USA: IEEE, 2020:248-249.

[22] HUSZÁR F. On quadratic penalties in elastic weight consolidation[J]. The National Academy of Sciences of the United States of America, 2018,115(11):2496-2497.

[23] SCHWARZ J, CZARNECKI W, LUKETINA J, et al. Progress & compress: a scalable framework for continual learning[C]// Proceedings of International Conference on Machine Learning. Stockholm, Sweden: IMLS, 2018: 4528-4537.

[24] DE LANGE M, ALJUNDI R, MASANA M, et al. A continual learning survey: defying forgetting in classification tasks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021,44(7):3366-3385.

[25] PARISI G I, KEMKER R, PART J L, et al. Continual lifelong learning with neural networks: a review[J]. Neural Networks, 2019,113:54-71.

[26] QU H, RAHMANI H, XU L, et al. Recent advances of continual learning in computer vision: an overview[EB/

OL]. (2021-09-24)[2022-12-05]. https://arxiv.org/pdf/2109.11369.pdf.

[27] VAN HOUDT G, MOSQUERA C, NÁPOLES G. A review on the long short-term memory model[J]. Artificial Intelligence Review, 2020,53(8):5929-5955.

[28] SINGH P, MAZUMDER P, RAI P, et al. Rectification-based knowledge retention for continual learning[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville, USA: IEEE, 2021:15282-15291.

[29] HAARNOJA T, ZHOU A, ABBEEL P, et al. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]// Proceedings of International Conference on Machine Learning. Stockholm, Sweden: ICML, 2018:1861-1870.

[30] HAARNOJA T, ZHOU A, HARTIKAINEN K, et al. Soft actor-critic algorithms and applications[EB/OL]. (2019-01-29)[2022-12-05]. https://arxiv.org/pdf/1812.05905.pdf.

[31] FUJIMOTO S, HOOF H, MEGER D. Addressing function approximation error in actor-critic methods[C]// Proceedings of International Conference on Machine Learning. Stockholm, Sweden: ICML, 2018:1587-1596.

[32] NIU J, HU Y, JIN B, et al. Two-stage safe reinforcement learning for high-speed autonomous racing[C]// Proceedings of IEEE International Conference on Systems, Man, and Cybernetics. Toronto, Canada: IEEE, 2020:3934-3941.

[33] LOIACONO D, CARDAMONE L, LANZI P L. Simulated car racing championship: competition software manual [EB/OL]. (2013-04-29)[2022-12-05]. https://arxiv.org/pdf/1304.1672.pdf.

[34] KINGMA D P, BA J. ADAM: a method for stochastic optimization[C]// Proceedings of International Conference for Learning Representations. San Diego, USA: ICLR, 2015:1-9.

Decision making based on continual reinforcement learning for autonomous racing

NIU Jingyu[*,**], HU Yu[*,**], LI Wei[*], HAN Yinhe[*,**]

([*]Research Center for Intelligent Computing Systems, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

([**]University of Chinese Academy of Sciences, Beijing 100049)

Abstract

The variety of road shapes and materials presents a serious decision-making challenge for high-speed autonomous racing. To address the issue of dynamics gap between various roads, a decision-making algorithm based on continual reinforcement learning (CRL) is proposed. These roads are considered as different tasks. The first training stage of the algorithm extracts low-dimension task features that can characterize the vehicle dynamics on different roads. These features are used to compute the task similarity. The second training stage of the algorithm provides two CRL constraints for policy learning. One is the weight regularization constraint, which restricts the updates of policy weights that are important for old tasks. This restriction is adaptively regulated by task similarity. The other is the reward constraint, which encourages no performance degradation on old tasks while the policy is learning a new task. Racing experiments with different task sequences and CRL metrics are set to evaluate the algorithm. The results show that the proposed algorithm outperforms baselines without storing old tasks' data or expanding policy network size.

Key words: reinforcement learning (RL), continual learning, decision making, autonomous racing, dynamics feature extraction