带 Q 网络过滤的两阶段 TD3 深度强化学习方法

周娴玮,包明豪,叶 鑫,余松森
(华南师范大学 软件学院,广东 佛山 528000)

摘 要:常规的深度强化学习模型训练方式从“零”开始,其起始策略为随机初始化,这将导致智能体在训练前期阶段探索效率低、样本学习率低,网络难以收敛,该阶段也被称为冷启动过程。为解决冷启动问题,目前大多数工作使用两阶段深度强化学习训练方式;但是使用这种方式的智能体由模仿学习过渡至深度强化学习阶段后可能会出现遗忘演示动作的情况,表现为性能和回报突然性回落。因此,该文提出一种带 Q 网络过滤的两阶段 TD3 深度强化学习方法。首先,通过收集专家演示数据,使用模仿学习-行为克隆以及 TD3 模型 Q 网络更新公式分别对 Actor 网络与 Critic 网络进行预训练工作;进一步地,为避免预训练后的 Actor 网络在策略梯度更新时误选择估值过高的演示数据集之外动作,从而遗忘演示动作,提出 Q 网络过滤算法,过滤掉预训练 Critic 网络中过高估值的演示数据集之外的动作估值,保持演示动作为最高估值动作,有效缓解遗忘现象。在 Deep Mind 提供的 Mujoco 机器人仿真平台中进行实验,验证了所提算法的有效性。

关键词:两阶段深度强化学习;冷启动问题;模仿学习;预训练网络;TD3
中图分类号:TP391.9 **文献标识码**:A **文章编号**:1673-629X(2023)10-0101-08
doi:10.3969/j.issn.1673-629X.2023.10.016

Two-stage TD3 Deep Reinforcement Learning Algorithm with Q Network Filtration

ZHOU Xian-wei, BAO Ming-hao, YE Xin, YU Song-sen
(School of Software, South China Normal University, Foshan 528000, China)

Abstract: Training of conventional deep reinforcement learning model starts from “zero”, with random initialization strategy, which leads to low exploration efficiency, low sample learning rate and low network convergence of the agent in the early stage of training, which is also known as the cold start problem. To solve the problem, most of the current work use the two-stage deep reinforcement learning training mode. However, the agent using this method may forget the demonstration action after the transition from imitation learning to deep reinforcement learning, which is manifested as an abrupt decline in performance and reward. Therefore, a two-stage TD3 deep reinforcement learning method with Q network filtering is proposed. Firstly, collecting expert demonstration data, the pre-training of Actor network and Critic network is carried out respectively by using imitation learning-behavior cloning and TD3 model Q network update formula. Further, in order to avoid the pre-trained Actor network mistakenly selecting overvalued actions out of the demonstration data set while the strategy gradient update is taking place, resulting in forgetting the demonstration actions, we propose a Q network filtering algorithm to filter out the overvalued action outside the demonstration data set in the pre-trained Critic network, and keep the demonstration actions as the highest value actions, effectively alleviating the phenomenon of forgetting. Experiments were carried out on the Mujoco robot simulation platform provided by Deep Mind to verify the effectiveness of the proposed algorithm.

Key words: two-stage deep reinforcement learning; cold start; imitation learning; pretraining network; TD3

0 引 言

深度强化学习是一种解决决策性问题的算法,在自动驾驶[1]、机器人控制[2]、无人机[3]等领域应用广泛。深度强化学习以“试错”的方式与环境进行交互,智能体通过学习这些交互过程产生的经验,以最大化环境中获得的累积奖励为目标,不断优化自身策略[4]。

常规的深度强化学习模型训练方式由“零”开始训练,即智能体的起始策略为随机初始化[5]。这种方

收稿日期:2022-11-21 **修回日期**:2023-03-22
基金项目:广东省应用型科技研发重大专项(2016B020244003);广东省基础与应用基础研究基金(2020B1515120089,2020A1515110783);广东省企业科技特派员项目(GDKTP2020014000)
作者简介:周娴玮(1982-),男,博士,讲师,硕导,研究方向为机器人技术、多传感信息融合;通讯作者:余松森(1972-),男,博士,教授,硕导,研究方向为计算机视觉、物联网技术。

式会导致智能体在与环境进行交互的前期阶段过程中,出现盲目性探索环境,样本学习率低,并没有良好、稳定的表现,这种现象也被有关学者定义为冷启动(Cold Start)[6]问题。

为解决冷启动问题,近些年来,有学者提出两阶段深度强化学习训练方式[7]。具体而言,使用 A-C(Actor-Critic)演员-评论家模式的深度强化学习模型,通过采集专家演示数据[8],利用模仿学习对智能体进行预训练,而后采用深度强化学习进行下一步的训练。通过策略预训练,减少智能体前期盲目探索次数,提高学习效率,加快网络收敛速度,从而缓解训练前期的冷启动问题。

两阶段深度强化学习训练方式虽然能够缓和智能体冷启动问题,但是在智能体从模仿学习过渡至深度强化学习阶段后,可能出现专家演示动作被遗忘的问题,具体表现为性能和回报出现突然性回落的现象[2-3,9]。

造成该现象的主要原因有以下两个:

(1)若智能体在模仿学习阶段仅对 Actor 网络进行预训练,而 Critic 网络选择随机初始化[2,9]。在深度强化学习前期训练阶段,由于 Critic 网络未经过预训练,因此无法提供准确的动作估值,导致 Actor 网络进行策略梯度更新时做出错误的选择,将所学的演示动作遗忘。

(2)即使 Critic 网络经过预训练,但是由于专家演示数据集中没有提供所有动作经验,预训练时演示数据集之外的动作的估值可能被过高估计,因此演示动作不一定为最高估值动作[10]。在深度强化学习阶段进行策略梯度更新时,预训练后的 Actor 网络追求估值最高的动作,可能选择演示数据集之外的动作,进而遗忘所学的演示动作,严重时导致训练速度大大减缓。

综上所述,该文针对上述两个主要原因做出如下改进工作:

(1)提出两阶段 TD3(Twin Delayed Deep Deterministic Policy Gradient)[11]深度强化学习方法。首先,通过采集专家演示数据集采用模仿学习-行为克隆[12]方式对 Actor 网络进行预训练;其次,使用 TD3 模型 Q 网络更新公式对 Critic 网络进行预训练,避免其随机初始化。

(2)提出 Q 网络过滤算法,通过所提出的过滤函数调整预训练 Critic 网络参数权重,过滤掉网络中过高估值的演示数据集之外的动作估值,使演示动作成为估值最高的动作。目的是使预训练后的 Actor 网络在深度强化学习阶段进行策略梯度更新时,减少选择演示数据集之外的动作,尽量避免遗忘演示动作。

1 相关工作

模仿学习[13]是一种监督学习,可以在离线情况下根据数据集进行快速有效地学习,形成一个端到端的网络模型。虽然模仿学习存在分布不匹配、鲁棒性差等问题[14],但是可以被运用于智能体的预训练,而后采用深度强化学习进行改进训练,因此能够加快深度强化学习网络的收敛速度。例如,Peng 等人[7]提出一个两阶段框架,称为 IPP-RL,通过模仿学习预训练模型共享权值来初始化 DDPG(Deep Deterministic Policy Gradient)[15]模型的(Actor)行动者网络,以加快深度强化学习的训练速度。Pfeiffer 等人[2]提出增强模仿学习(R-IL)方法,结合基于专家演示的有监督的 IL,对后续的 RL 策略(Actor)网络进行预训练,比纯 RL 更容易和更快的训练。虽然上述两阶段深度强化学习方法能够缓解冷启动问题,但 Pfeiffer 与 Jing 等人[2,9]的工作表明,由于随机初始化的 Critic 网络需要在深度强化学习的前期阶段进行训练工作,在此期间无法提供准确的动作估值,可能使预训练后的 Actor 网络做出错误的更新决定,导致智能体出现性能和回报突然性回落的情况,极大地影响了网络训练速度。

为改善此情况,许多学者提出相应的 Critic 网络预训练方法。例如,Chen 等人[16]提出将 DDPG 模型中的 Actor 网络与 Critic 网络采用相同的图像提取特征 CNN 架构。首先对 Actor 网络进行预训练,随后将其卷积网络权重赋值给 Critic 网络,使两者均拥有初始能力。Ma 等人[17]提出利用先前收集的专家演示数据集通过最小化一步 TD 误差公式对 Critic 网络进行预训练;同时 Actor 网络通过复合策略梯度更新公式及行为克隆损失函数进行预训练工作。将预训练完毕后的网络权重用以初始化 DDPG 模型进行下一步的训练。Wang 等人[3]使用 ORCA(Optimal Reciprocal Collision Avoidance)作为引导策略生成演示数据,设计出一个基于 ORCA 速度障碍的损失函数来预训练 Actor 网络;同时使用 DDPG 的 Q 网络更新函数对 Critic 网络进行预训练。当智能体达到 ORCA 能力值时,采用深度强化学习进行下一步的训练。

这些工作虽然提出了相应的 Critic 网络预训练方法来改善智能体性能和回报突然性回落的情况,但是并未关注到造成该情况出现的第二个原因,即忽略了预训练后 Critic 网络中虚高的演示数据集之外的动作估值对 Actor 网络的更新影响。为此,借鉴前人经验,该文同时弥补其不足,提出相应的改进工作。

2 带 Q 网络过滤的两阶段 TD3 深度强化学习方法

该文提出的方法分为以下两大阶段:(1)预训练

阶段(Actor、Critic 网络预训练)以及 Q 网络过滤阶段,这两个小阶段使用采集得到的专家演示数据进行网络训练;(2)深度强化学习训练阶段,将上一阶段预训练得到的 Actor 网络以及 Q 网络过滤后的 Critic 网络参数权重用以初始化 TD3 深度强化学习模型,使用深度强化学习进一步训练网络。

2.1　TD3 深度强化学习模型

该文采用的 TD3 深度强化学习算法,是一种基于 A-C 模式面向连续动作空间的确定性策略深度强化学习模型。其包含主策略网络(Main Actor Network)π_φ、目标策略网络(Target Actor Network)$\pi_{\varphi'}$;与 DDPG 模型相比:

(1)TD3 模型的主评价网络(Main Critic Network)拥有 Q_{θ_1}、Q_{θ_2} 两个 Q 网络,计算目标值时取其较小值,目的在于缓解 DDPG 中存在的过高估计问题;目标评价网络(Target Critic Network)也拥有 $Q_{\theta_1'}$、$Q_{\theta_2'}$ 两个 Q 网络;

(2)增加延迟更新机制,使 Actor 网络更新频率相对于 Critic 网络更新频率要小,从而使 Actor 网络更加平稳地进行训练;

(3)添加了 Smoothing Regularization 机制,在计算目标预估值时引入随机噪声 ε,目的是使预测估值更加准确(ε 从正态分布 $\mathbb{N}(0,\tilde{\sigma})$ 中随机抽取数值,$\tilde{\sigma}$ 为标准差;同时 ε 的取值上下限为 $[-c,c]$,c 为智能体动作空间数值上限)。

TD3 模型的 Critic 网络更新公式为:

$$\tilde{a}=\pi_{\varphi'}(s')+\varepsilon,\varepsilon\sim \mathrm{clip}(\mathbb{N}(0,\tilde{\sigma}),-c,c) \tag{1}$$

$$y=r+\gamma\min_{i=1,2}Q_{\theta_i'}(s',\tilde{a}) \tag{2}$$

$$L=\frac{1}{N}\sum_{t=1}^{N}(y-Q_{\theta_i}(s,a))^2 \tag{3}$$

公式(1)中,$\tilde{a}$ 表示为 Target Actor 网络 $\pi_{\varphi'}$ 在下一状态 s' 下进行的预估动作,同时添加随机噪声 ε。式(2)中目标值 y 是由真实奖励值 r 与乘以折扣因子 γ 的预测值 $Q_{\theta_i'}(s',\tilde{a})$ 所构成,而预测值则是由 Target Critic 的两个 Q 网络中选取较小值得来。

TD3 模型使用均值平方差公式作为 Critic 网络的 Loss 损失函数,由目标值 y 与当前值 $Q_{\theta_i}(s,a)$ 之间的差值组成。其中 N 表示从经验池中随机抽取得到的经验数量,这些经验 (s,a,r,s') 包含了当前状态、动作、奖励值、下一状态。

在进行 Critic 网络更新时,不断地从经验池中随机采样 N 条经验代入公式(3)的损失函数 L,使用随机梯度下降法更新 Critic 网络参数 θ_i,以最小化目标值与当前值之间的差距。

在 Critic 网络经过 d 次更新后,同样地,需要从经验池中随机采样 N 条经验数据 (s) 代入策略梯度更新公式(4)。根据 Critic 网络中第一个 Q 网络 Q_{θ_1} 的情况对 Actor 网络 π_φ 进行训练,使用随机梯度下降法进行网络参数 φ 的更新:

$$\nabla_\varphi J(\varphi)=\frac{1}{N}\sum_{t=1}^{N}\nabla_a Q_{\theta_1}(s,a)\mid_{a=\pi_\varphi(s)}\nabla_\varphi\pi_\varphi(s) \tag{4}$$

同时,采用软更新机制,将 Actor 及 Critic 主网络参数权重 φ、θ_i 取 τ 部分复制给目标网络 φ'、θ_i'。

$$\theta_i'\leftarrow\tau\theta_i+(1-\tau)\theta_i' \tag{5}$$

$$\varphi'\leftarrow\tau\varphi+(1-\tau)\varphi' \tag{6}$$

2.2　预训练阶段

在预训练阶段,通过采集得到的专家演示数据集 $D\{s_t,a_t,r_t,s_t'\}$(演示状态、演示动作、奖励值、下一个演示状态)以模仿学习-行为克隆方式对深度强化学习 TD3 模型中 Actor 网络 π_φ 进行预训练。为避免随机初始化的 Critic 网络对后续深度强化学习阶段的影响,在预训练阶段使用 TD3 模型的 Q 网络更新公式(1)~(3)对 Critic 网络 Q_{θ_i} 进行预训练。

2.2.1　Actor 网络预训练

首先,从演示数据集 D 中随机采样 N 条数据 (s_t,a_t) 代入公式(7)中,该文使用均值平方差作为行为克隆的损失函数 L_{BC},利用梯度下降法进行 Main Actor 网络参数 φ 的更新。不断地随机采样 N 条演示数据对网络进行训练,直至网络收敛后将主网络参数权重 φ 复制给目标网络 φ',完成该阶段预训练工作。

$$L_{\mathrm{BC}}=\sum_{t=1}^{N}\|\pi_\varphi(s_t)-a_t\|^2 \tag{7}$$

$$\varphi'\leftarrow\varphi \tag{8}$$

2.2.2　Critic 网络预训练

在 Actor 网络预训练完毕后,对 Critic 网络进行预训练工作。同样地,从演示数据集 D 中随机采样 N 个数据 (s_t,a_t,r_t,s_t'),同时将预训练完毕的 Target Actor 网络参数权重 φ' 代入 Q 网络更新公式(1)~(3)中,得到新的式子:

$$a'=\pi_\varphi(s_t') \tag{9}$$

$$y=r_t+\gamma\min_{i=1,2}Q_{\theta_i'}(s_t',a') \tag{10}$$

$$L=\frac{1}{N}\sum_{t=1}^{N}(y-Q_{\theta_i}(s_t,a_t))^2 \tag{11}$$

式(9)中,a' 表示为预训练后的 Target Actor 网络在演示状态 s_t' 进行的预测动作,与原来的 Q 网络更新公式(1)区别在于去除了 ε 噪声,目的是减少噪声对专家演示动作估值的干扰。

根据式(11)对网络损失函数 L 进行梯度下降,更新 Main Critic 网络参数 θ_i,同时将主网络权重 θ_i 复制给目标网络 θ_i'。随后,不断地随机采样 N 条数据对网

络进行训练直至网络收敛。

$$\theta_i' \leftarrow \theta_i \tag{12}$$

算法 1 预训练阶段算法

输入:演示数据集 D,样本采样数量 N,奖励折扣因子 γ,训练总步骤数 T

输出:预训练后的网络参数权重 θ_i、θ_i'、φ、φ'

1. 随机初始化 Actor 网络参数 φ、φ' 及 Critic 网络参数 θ_i、θ_i'

2. For←0 to T do

3. 从演示数据集 D 中随机采样 N 条 (s_t, a_t) 数据

4. 将 (s_t, a_t) 代入行为克隆损失函数式(7)中,更新 Main Actor 网络参数 φ

5. End For

6. 将 Main Actor 网络参数权重复制给目标网络 $\varphi' \leftarrow \varphi$

7. For←0 to T do

8. 从演示数据集 D 中随机采样 N 条 (s_t, a_t, r_t, s_t') 数据

9. 将 (s_t, a_t, r_t, s_t')、预训练后的 Main Actor 网络参数权重 φ 代入式(9)~(11)中,根据损失函数式(11)更新 Main Critic 网络参数 θ_i

10. 将 Critic 主网络参数权重复制给目标网络 $\theta_i' \leftarrow \theta_i$

11. End For

2.3 Q 网络过滤阶段

2.3.1 Q 网络过滤算法原理

由于预训练后的 Critic 网络中存在过高估值的演示数据集之外的动作估值,而这些估值会影响深度强化学习阶段 Actor 网络的更新,导致智能体出现性能和回报突然性回落的情况,大大影响网络训练速度。为此,借鉴滤波算法原理,该文提出 Q 网络过滤算法,对演示数据集之外的动作过高估值进行过滤操作。

Q 网络过滤算法在智能体进入深度强化学习阶段之前,使用过滤函数调整预训练后 Critic 网络参数权重,降低网络中演示数据集之外的动作估值,使演示动作 a_t 成为估值最高的动作。

该算法原理如图 1 所示,横坐标表示在某个演示数据集状态 s_t 下的动作空间(Action space),在图中,将多维的动作空间进行一维化处理;纵坐标表示该动作的估值(Q-value)。

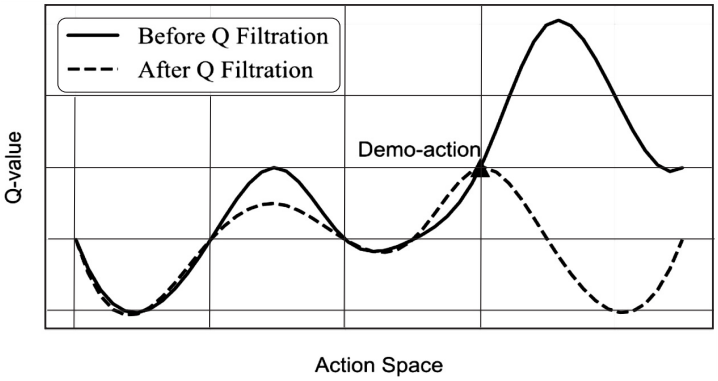


图 1 Q 网络过滤算法原理

图 1 中的实线假设为预训练后 Critic 网络中动作估值情况,此时演示动作 a_t(Demo-action 正三角标记)并非为最高估值动作。而在经历 Q 网络过滤算法后,如图 1 虚线所示,演示动作 a_t 成为最高估值动作。

这样做的目的是使 Actor 网络在过渡至深度强化学习阶段后,若是在演示数据集状态 s_t 情况下进行策略梯度更新时,所追求的最高估值动作与演示动作 a_t 重合,尽量避免误选择演示数据集之外的动作,遗忘演示动作。

2.3.2 Q 网络过滤算法步骤

步骤一 寻找估值最高动作:由于演示数据集中并未提供所有动作的经验,该文利用策略梯度更新公式寻找预训练 Critic 网络中 s_t 状态下估值最高的动作。首先从演示数据 D 中随机采样 N 条数据 (s_t, a_t),将 s_t 以及预训练后的 Actor、Critic 网络参数权重 φ、θ_i 代入至策略梯度更新公式(4)中,对 $\nabla_\varphi J(\varphi)$ 进行随机梯度下降,更新 Main Actor 网络参数 φ。将更新后的网络参数 φ 代入式(13),得到在状态 s_t 下估值最高的动作 a_m。

$$a_m = \pi_\varphi(s_t) \tag{13}$$

步骤二 降低演示数据集之外动作估值:将估值最高的动作 a_m,即演示数据集之外动作,与先前随机采样得到演示动作 a_t 及状态 s_t 代入该文提出的第一条 Q 网络滤波函数 F_1 中。

$$F_1(\theta_i) = \frac{1}{N}\sum_{t=1}^{N}\left[Q_{\theta_i}(s_t, a_m) - \lambda Q_{\theta_i}(s_t, a_t)\right]^2 \times I_{Q_{\theta_i}(s_t, a_m) \geqslant Q_{\theta_i}(s_t, a_t)} \tag{14}$$

式中,Q_{θ_i}、Q_{θ_i} 分别为预训练后的 Main、Target Critic 网络,N 为随机采样的数据数量。公式(14)中存在一个指示函数 $Q_{\theta_i}(s_t, a_m) \geqslant Q_{\theta_i}(s_t, a_t)$,该指示函数利用 Critic 网络中第一个 Q 网络来判断动作 a_m 的估值与专家演示动作 a_t 的估值的关系,若高于或等于则应当进行过滤操作,降低其估值,否则不进行过滤。

为了能够准确地对比动作估值大小,该文采用预训练后的 Target Critic 网络 Q_{θ_i} 作为基准,该网络在过滤阶段不参与网络更新,而 Main Critic 网络 Q_{θ_i} 则会进

行过滤以及网络更新。引入 λ 过滤系数则是为了将动作 a_m 估值降低至演示动作 a_t 之下,因此取值为 $0 \leqslant \lambda < 1$。

步骤三 保持演示动作估值:由于第一条 Q 网络过滤函数 $F_1(\theta_i)$ 能过滤与演示动作估值相同的其他动作,为保障在过滤期间 Main Critic 网络中演示动作估值不会被误降低,该文引入第二条 Q 网络过滤函数 $F_2(\theta_i)$。由于 Target Critic 网络 $Q_{\theta_i'}$ 并未参与到过滤阶段的网络更新操作,因此能够准确给出演示动作估值。该函数将使 Main Critic 与 Target Critic 网络中的演示动作估值保持一致。

$$F_2(\theta_i) = \frac{1}{N}\sum_{t=1}^{N}\left[Q_{\theta_i}(s_t,a_t) - Q_{\theta_i'}(s_t,a_t)\right]^2 \quad (15)$$

步骤四 网络更新:将两条过滤函数 $F_1(\theta_i)$、$F_2(\theta_i)$ 进行相加后,根据式(16)进行随机梯度下降,更新 Main Critic 网络的参数 θ_i。

$$\nabla_{\theta_i}F(\theta_i) = \nabla_{\theta_i}F_1(\theta_i) + \nabla_{\theta_i}F_2(\theta_i) \quad (16)$$

演示数据集 D 中存在多条数据,因此需要多次循环过滤。经过 Q 网络过滤后,网络中演示数据集之外动作估值比专家动作估值大、或者相等的情况会明显减少,从而达到“过滤”目的。

步骤五 网络复制:在 Q 网络过滤阶段,为了寻找最高估值动作,预训练 Main Actor 网络参数权重 φ 被策略梯度公式更新,为保持预训练阶段所学习的专家知识,将过滤阶段未参与网络更新操作的 Target Actor 网络参数权重 φ' 复制给 Main Actor 网络。

$$\varphi \leftarrow \varphi' \quad (17)$$

与此同时,由于 Main Critic 网络参数 θ_i 在过滤阶段经过了网络调整,其参数权重已经发生变化,即完成了过滤操作,因此将过滤后的主网络参数权重复制给目标网络。

$$\theta_i' \leftarrow \theta_i \quad (18)$$

算法 2　Q 网络过滤算法

输入:演示数据集 D,样本采样数量 N,训练总步骤数 T,过滤系数 λ,预训练后的 Actor 网络参数权重 φ、φ',预训练后的 Critic 网络参数权重 θ_i、θ_i'

输出:过滤后的 Critic 网络参数权重 θ_i、θ_i'

1. For←0 to T do

2. 从演示数据集 D 中随机采样 N 条 (s_t,a_t) 数据

3. 将 N 个 s_t 数据及 φ、θ_i 代入式(4),更新 Main Actor 网络参数 φ

4. 将更新后的参数 φ 代入式(13)得到估值最高动作 a_m

5. 将 N 个 (s_t,a_t,a_m) 数据、θ_i、θ_i'、λ 代入式(16),进行过滤操作,更新 Main Critic 参数 θ_i

6. End For

7. 将网络参数权重 θ_i 复制给 θ_i'

8. 将网络参数权重 φ' 复制给 φ

2.4　深度强化学习阶段

该文将预训练后的 Actor 网络以及经过 Q 网络过滤后的 Critic 网络参数权重用于初始化 TD3 深度强化学习模型,而后智能体将采用 TD3 更新公式进行自主探索环境学习[16]。

将上述智能体训练阶段绘制成总体算法流程,见图 2。图中包括预训练阶段(Actor 网络预训练,Critic 网络预训练)、Q 网络过滤阶段以及 TD3 深度强化学习阶段。

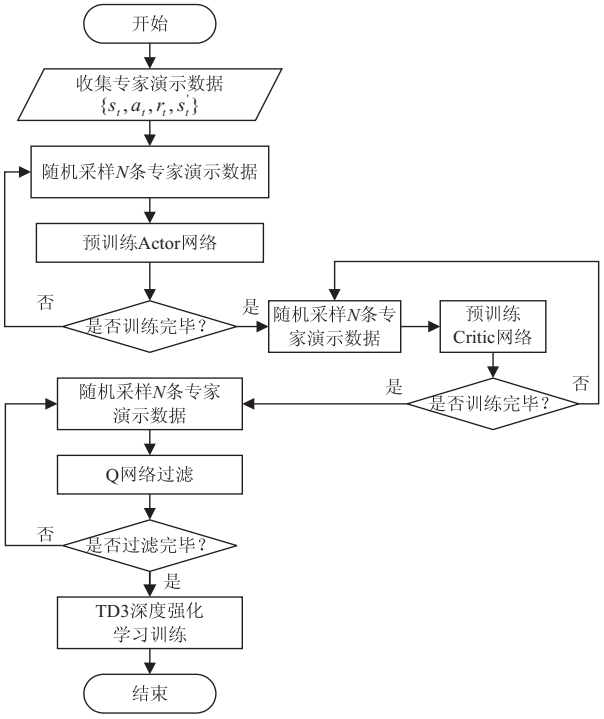


图 2　总体算法流程

3　仿真实验

该文采用 Deep Mind 提供的 Mujoco[18] 多关节仿生机器人环境进行实验,如图 3 所示,使用 3 种不同关节类型的机器人 Ant-v3(左下)、HalfCheetah-v3(左上)、Walker2d-v3(右)来验证所提出的 Q 网络过滤算法。

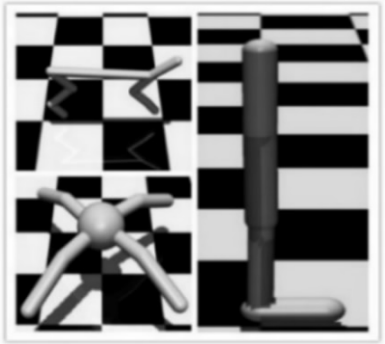


图 3　三种不同关节结构 Mujoco 机器人

仿真机器人奖励函数由三部分组成:前向速度、健

康度、动作消耗,不同的机器人中这三部分组成比例不相同;机器人所观察的状态 s 、动作 a 均为多维向量,每种机器人维度不相同。

3.1 实验设置

3.1.1 环境参数设置

在实验开始前,采用 3 个机器人未训练完毕的 TD3 模型作为“专家”生成所需的演示数据,这些“专家”并未达到深度强化学习收敛时的性能,其性能(回报)如表 1 所示。每一种机器人分别采集 20 条专家演示轨迹数据,每一条演示轨迹由 1 000 个 $(s_t,a_t,r_t,s_t^{'})$ 经验四元组构成,总共构成 20 000 经验的专家演示数据集 $D\{s_t,a_t,r_t,s_t^{'}\}$ 。利用演示数据集对 Actor、Critic 网络进行预训练工作与 Q 网络过滤操作。完成前置工作后,将网络权重赋值给 TD3 模型,进入下一步深度强化学习阶段训练。

表 1 专家演示数据

机器人类型	专家性能(回报)	轨迹个数	经验个数
Ant-v3	4 000	20	20 000
HalfCheetah-v3	9 500	20	20 000
Walker2d-v3	4 000	20	20 000

在深度强化学习阶段,每个机器人在一个随机种子数环境训练 10^6 次,总共训练 5 个随机种子数,其中机器人单次训练回合最长步数为 1 000 步;在回合中,若其健康度低至 0,则会重置机器人,重新开始下一回合训练。实验中,机器人在与环境交互 25 000 步后,开始进行网络更新工作,此后每步都进行网络更新。每隔 2 500 步进行一次测试,其中包括回报、差异度测试。将当前环境所训练的 TD3 模型转移至新的随机种子数环境中测试 10 次,取得分结果平均值作为当前阶段的回报情况;与此同时进行智能体 Actor 网络输出动作与专家演示动作之间的差异度测试,将差异值记录。

3.1.2 超参数设置

该文所使用的 TD3 模型的 Actor、Critic 网络采用三层全连接网络,其中隐藏层网络宽度为 256,输入输出维度根据机器人种类决定。折扣因子 $\gamma=0.99$,Actor、Critic 网络学习率为 $3e^{-3}$,软更新系数 $\tau=0.01$,网络延迟更新系数 d 为 2,探索噪声系数 $\varepsilon=0.1$,经验池大小为 10^6,每次采样 $N=256$ 条经验。

3.2 实验结果分析

本次实验采用以下算法进行对比:

(1)TD3 模型 Baseline 基准;

(2)经历 Actor、Critic 网络预训练但未经过 Q 网络过滤 TD3 模型,即常规的两阶段 TD3 深度强化学习方法;

(3)经历 Actor、Critic 网络预训练同时经过 Q 网络过滤 TD3 模型,即带 Q 网络过滤的两阶段 TD3 深度强化学习方法。

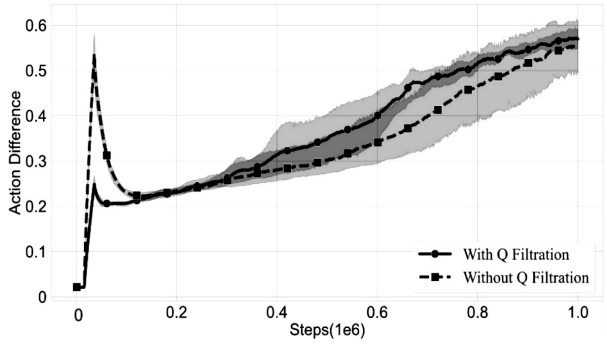
将这三种算法分别在 Ant-v3、HalfCheetah-v3、Walker2d-v3 仿真环境中进行实验,同时收集智能体产生的两种数据,分别为:

(1)深度强化学习训练阶段智能体在演示集状态 s_t 下 Actor 网络输出动作与专家动作 a_t 差异均值,用式(7) L_{BC} 作评价,其结果如图 4 所示。

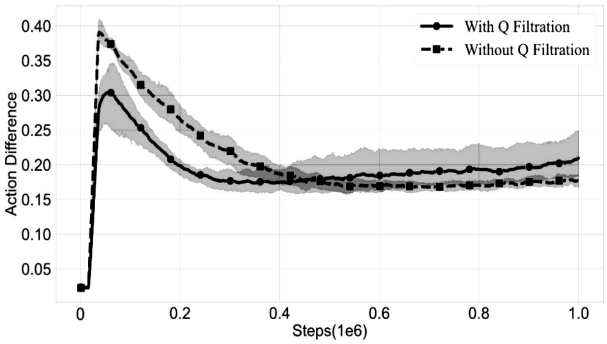
(2)在训练过程中的测试回报情况如图 5 所示,最终测试得分回报如表 2 所示。

3.2.1 动作差异分析

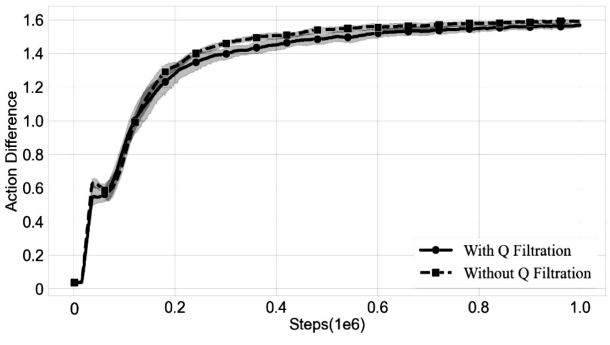
如图 4 所示,图中用圆实线(Our)表示经过 Q 网络过滤后的智能体数据,正方形虚线(Without Q Filtration)表示未经过 Q 网络过滤的智能体。图中阴影部分表示数据的 95% 置信区间。



(a) Ant-v3



(b) HalfCheetah-v3



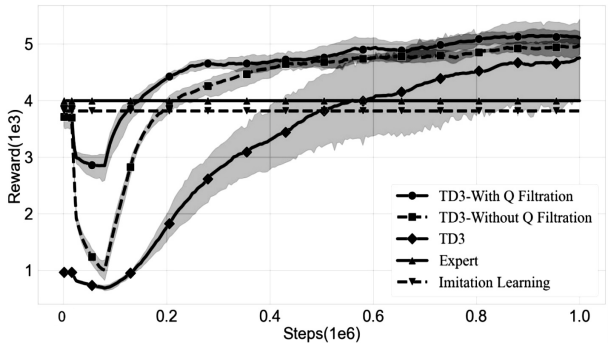
(c) Walker2d-v3

图 4 Mujoco 机器人动作差异

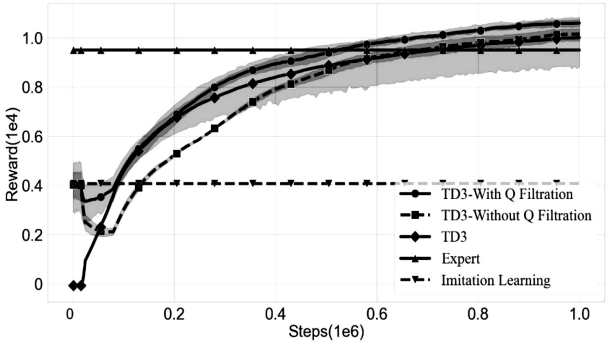
从图 4 的动作差异图中可以看出,在 Actor 网络开始进行策略梯度更新时(智能体与环境交互 25 000 步后),网络输出动作与数据集中演示动作的差异值急速上升。而经历了 Q 网络过滤的智能体,其上升幅度小于未过滤的智能体。这表明经过 Q 网络过滤后的智能体遗忘专家演示动作程度较小,保留了更多的专家演示动作,其中 Ant-v3、HalfCheetah-v3 较为明显。而在训练的后期,由于智能体探索环境,寻找到更优或者替代的动作,因此动作差异程度比较大。

3.2.2　回报情况分析

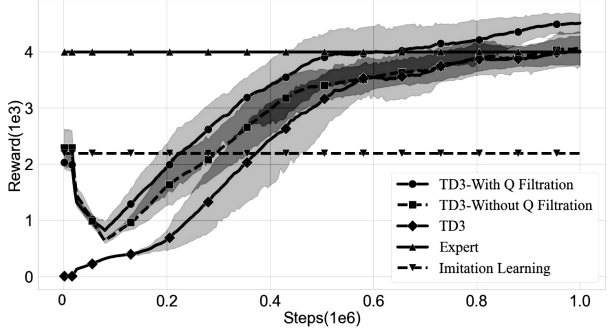
在图 5 的众多曲线中,用圆实线(Our)表示经过 Q 网络过滤后的智能体回报均值;正方形虚线(Without Q Filtration)则表示未经历过滤的智能体回报均值;使用菱形实线表示 TD3 基准回报均值。图中阴影部分表示数据的 95% 置信区间。而在直线中,使用正三角形实线(Expert)表示专家演示回报(性能),倒三角虚线(Imitation Learning)表示模仿学习回报情况。



(a) Ant-v3



(b) HalfCheetah-v3



(c) Walker2d-v3

图 5　Mujoco 机器人回报

从图 5 的回报结果中可以得知,由于智能体经历了预训练,其初始性能与模仿学习阶段相同。在与环境交互 25 000 步后,网络开始进行更新时性能和回报迅速下降,低于模仿学习阶段。但随着训练的深入,智能体能够从中恢复至模仿学习时期能力水平。而经历 Q 网络过滤的智能体能够更快恢复,同时其得分回报下降程度相对于未经历过滤的智能体更小。从实验中可得知 Q 网络过滤算法能够改善性能和回报突然性回落情况。

与 TD3 基准对比,该文提出的算法能够从一个较为良好的策略开始进行训练,缓解冷启动问题,从而加快了网络收敛速度,最终收敛结果如表 2 所示。

表 2　智能体最终回报

机器人类型	经历 Q 网络过滤最终回报(Our)	未经 Q 网络过滤最终回报(without Q Filtration)	TD3 最终回报	模仿学习回报(Imitation Learning)
Ant-v3	5 106	4 977	4 751	3 814
HalfCheetah-v3	10 595	10 149	10 006	4 082
Walker2d-v3	4 507	4 043	4 008	2 192

表 2 记录了 3 个机器人在不同算法最终收敛时的回报均值情况。从表 2 中可以看出,经过 Q 网络过滤后的智能体最终回报要高于 TD3 及未经过过滤的智能体回报。同时结合表 2 及表 1 的实验数据中得知,3 个机器人的模仿学习平均回报均没有超过专家演示,这是因为演示与测试并不在同一个随机种子数环境中,其次是专家演示数据 D 中只包含环境的一部分状态-动作分布区间,并没有包括所有未知状况,因此智能体并不能很好地处理未曾遇到过的状态。

4　结束语

针对两阶段深度强化学习训练方式中存在的遗忘演示动作问题,即智能体性能和回报突然性回落问题,提出一种带 Q 网络过滤的两阶段 TD3 深度强化学习方法。通过采集专家演示数据集对 Actor 及 Critic 网络进行预训练,同时使用 Q 网络过滤算法过滤掉预训练后 Critic 网络中过高估值的演示数据集之外的动作估值,有效缓解演示动作遗忘现象,改善了智能体性能和回报突然性回落情况。最终,通过 Mujoco 机器人仿真实验表明,该算法能够改善智能体得分回报突然性回落情况。

参考文献:

[1] YOON H S, LEE S H, SEO S W. Exploration strategy based on validity of actions in deep reinforcement learning[C]//2020 IEEE/RSJ international conference on intelligent robots and

systems (IROS). Piscataway: IEEE, 2020: 6134-6139.
[2] PFEIFFER M, SHUKLA S, TURCHETTA M, et al. Reinforced imitation: sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations [J]. IEEE Robotics and Automation Letters, 2018, 3(4): 4423-4430.
[3] WANG D, FAN T, HAN T, et al. A two-stage reinforcement learning approach for multi-UAV collision avoidance under imperfect sensing[J]. IEEE Robotics and Automation Letters, 2020, 5(2): 3098-3105.
[4] 陈佳盼,郑敏华. 基于深度强化学习的机器人操作行为研究综述[J]. 机器人,2022,44(2):236-256.
[5] SCHWARZER M, RAJKUMAR N, NOUKHOVITCH M, et al. Pretraining representations for data-efficient reinforcement learning[J]. Advances in Neural Information Processing Systems, 2021, 34: 12686-12699.
[6] XU B, HOU J, SHI J, et al. Learning time reduction using warm-start methods for a reinforcement learning-based supervisory control in hybrid electric vehicle applications[J]. IEEE Transactions on Transportation Electrification, 2020, 7(2): 626-635.
[7] PENG M, GONG Z, SUN C, et al. Imitative reinforcement learning fusing vision and pure pursuit for self-driving[C]// 2020 IEEE international conference on robotics and automation (ICRA). Piscataway: IEEE, 2020: 3298-3304.
[8] ZHU Z, LIN K, ZHOU J. Transfer learning in deep reinforcement learning: a survey[J]. arXiv: 2009.07888, 2020.
[9] JING M, MA X, HUANG W, et al. Reinforcement learning from imperfect demonstrations under soft expert guidance [C]//Proceedings of the AAAI conference on artificial intelligence. Menlo Park: AAAI, 2020: 5109-5116.
[10] GAO Y, XU H, LIN J, et al. Reinforcement learning from imperfect demonstrations[J]. arXiv: 1802.05313, 2018.
[11] FUJIMOTO S, HOOF H, MEGER D. Addressing function approximation error in actor-critic methods[C]//International conference on machine learning. Cambridge MA: JMLR, 2018: 1587-1596.
[12] LY A O, AKHLOUFI M. Learning to drive by imitation: an overview of deep behavior cloning methods[J]. IEEE Transactions on Intelligent Vehicles, 2020, 6(2): 195-209.
[13] 黄艳龙,徐 德,谭 民. 机器人运动轨迹的模仿学习综述[J]. 自动化学报,2022,48(2):315-334.
[14] CAI P, WANG H, HUANG H, et al. Vision-based autonomous car racing using deep imitative reinforcement learning[J]. IEEE Robotics and Automation Letters, 2021, 6(4): 7262-7269.
[15] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[J]. arXiv: 1509.02971, 2015.
[16] CHEN S, WANG M, SONG W, et al. Stabilization approaches for reinforcement learning-based end-to-end autonomous driving[J]. IEEE Transactions on Vehicular Technology, 2020, 69(5): 4740-4750.
[17] MA Y, XIE Y, ZHU W, et al. An efficient robot precision assembly skill learning framework based on several demonstrations[J]. IEEE Transactions on Automation Science and Engineering, 2022, 20(1): 124-136.
[18] TODOROV E, EREZ T, TASSA Y. Mujoco: aphysics engine for model-based control[C]//2012 IEEE/RSJ international conference on intelligent robots and systems. Vilamoura-Algarve: IEEE, 2012: 5026-5033.

(上接第 50 页)

[J]. International Journal of Smart Home, 2016, 10(10): 145-156.
[11] 孙爱良,王紫婷. 构建大学生学科竞赛平台培养高素质创新人才[J]. 实验室研究与探索,2012,31(6):96-98.
[12] 高云鹏,滕召胜,黎福海,等. 开放实验室与学科竞赛平台相结合的创新人才培养模式[J]. 实验技术与管理,2012,29(4):360-362.
[13] 徐 辉,郁汉琪,褚南峰. 依托校企合作共建平台提高大学生学科竞赛水平[J]. 实验室研究与探索,2010,29(12):153-155.
[14] 黎建辉,刘超良. 高校学科竞赛的管理与运行机制探讨[J]. 湖南人文科技学院学报,2010(3):119-121.
[15] 尹 仕,肖 看. 构建大学生多学科竞赛平台培养新型拔尖人才[J]. 实验技术与管理,2009,26(5):121-124.