DOI:10.19651/j.cnki.emt.2314063

基于改进 SAC 的倒立摆控制算法研究*

张晓莉 郭仕林 刘 鼎 宋婉莹
(西安科技大学通信与信息工程学院 西安 710600)

摘 要:针对倒立摆系统控制过程中易受外界干扰和自然不稳定的特点,以及深度强化学习 SAC 算法采样数据利用率较低和随机离线策略网络收敛较慢的问题,提出了一种结合近端经验采样和优化策略网络结构的改进算法 PRER_SAC。构建神经网络拟合函数,策略网络使用性能更优的 Mish 函数作为激活函数,设置自调节温度系数以增强智能体的探索能力;设计远、近两个经验池,及一种改变数据存放频率的训练策略,提高数据样本的利用率。通过仿真实验对比,所提方法在同等训练次数下所得回报值和算法收敛速度优于 DDPG 和 SAC 算法,同传统控制方法 PID 和 LQR 相比,有更好的控制效果。最后,对训练好的智能体加入角度扰动,可在 2 s 内被消除抑制,证明提出的算法具有较强的适用性。

关键词:激活函数;神经网络;深度强化学习;倒立摆系统
中图分类号:TP18 **文献标识码**:A **国家标准学科分类代码**:510.80

Research on the control algorithm of inverted pendulum based on improved SAC

Zhang Xiaoli Guo Shilin Liu Ding Song Wanying
(College of Communication and Information Engineering, Xi′an University of Science and Technology, Xi′an 710600, China)

Abstract: In response to the characteristics of external interference and natural instability in the control process of inverted pendulum systems, and the problems of low utilization of sampling data and slow convergence of random offline strategy networks in deep reinforcement learning SAC algorithm, an improved algorithm PRER_SAC is proposed that combines recency experience sampling and optimize policy network structure. The neural network fitting function is constructed, the policy network uses the better performance Mish function as the activation function, and sets the self-adjusting temperature coefficient to enhance the exploration ability of agent. Design two experience pools, far and near, and a training strategy to change the frequency of data storage. Through simulation experiments, the return value and convergence speed of the proposed method under the same number of training times are better than DDPG and SAC algorithms, and have better control effects than the traditional control methods PID and LQR. Finally, the angle disturbance added to the trained agent can be eliminated within 2 s, which proves that the proposed algorithm has strong applicability.

Keywords: activation function; neural network; deep reinforce learning; inverted pendulum system

0 引 言

倒立摆系统是一种独立控制变量个数小于系统自由度个数的非线性系统,该系统具有不稳定、高阶次、多变量、强耦合等特点[1-2]。对倒立摆系统的研究始于 20 世纪 50 年代,在过去的几十年里,被国内外学者研究扩展,该系统已广泛应用于军工业和航天工业的控制模型基础。

针对倒立摆的稳摆控制,有诸多广泛应用的方法。PID(proportional integral derivative)控制需要依赖控制经验来调整参数,对设计人员的理论和应用能力要求较高[3];模糊控制(fuzzy control,FZ)控制器规则表的设计十分依赖专家经验,且对于连续动作控制具有局限性[4];BP(back propagation)网络中网络参数的学习和更新需要借助完整的控制模型生成数据来进行训练[5]。线性二次调节器

收稿日期:2023-07-09
* 基金项目:国家自然科学基金青年项目(61901358)、中国博士后科学基金面上项目(2019QDJ207)、陕西省教育厅一般专项(20JK0757)资助

(linear quadratic regulator，LQR)可得到状态线性反馈的最优控制规律，构成闭环最优控制，但线性控制器的局限性是不可调和的[6]；滑模变结构控制(sliding mode variable structure control，SMC)控制器虽然在理论上控制效果较好，但该控制方法复杂，不易于实时应用[7]。

深度强化学习(deep reinforcrment learning，DRL)兼具了强化学习的决策能力以及深度学习的感知能力，无需数学模型，也不需要人工提供训练数据，能通过机体自身与环境交互不断完善自身，并作出利益最大的决策，具有很强的通用性，使其在控制领域占据了重要的地位[8-10]。杨文乐[11]设计了 3 种奖励方式，使用 DQN(deep q-network)算法实现了一级倒立摆的稳摆控制实验，并改进 Q 学习(Q-learning)引入一个参数粗调的 PID 控制器以获得状态变量边缘位置的训练样本，提高了样本利用率。翁士博[12]使用柔性演员-评论家(soft actor critic，SAC)算法实现了一级倒立摆的稳摆控制实验，但算法采用了固定的熵正则化系数，策略随机性不强，且采样策略无法最大化利用所有的训练数据。王雨轩等[13]将 PG(policy gradient)算法中神经网络激活函数替换为性能更优的 Swish 函数，并添加了基线函数以提高训练效率，并应用于小车倒立摆仿真实验，证明了算法的有效性。

在训练过程中，智能体需要大量的数据用以形成最优策略，以至在面对干扰的时候能够更有效率的做出调整，使系统具有更强的鲁棒性。因此智能体如何探索到更多的有效路径，以及充分利用训练数据最大化训练效率是要解决的关键问题。

因此，本文提出一种基于柔性演员-评论家的 PRER_SAC(piecewise recency experience replay soft actor critic)控制算法。设置自调节温度系数，增强智能体策略的随机性和有效性，策略网络引入新的 Mish 激活函数，采用两个经验池以及新的经验训练策略，提高样本的利用率和算法速率。搭建仿真环境，与深度强化学习算法 SAC、DDPG(deep deterministic policy gradient)和传统控制方法 PID、LQR 方法的控制效果进行对比，分析本文所提方法的适用性。

1　问题描述

1.1　倒立摆系统数学模型

小车倒立摆系统的模型主要由小车、摆杆及导轨组成，小车通过受力左右移动并使摆杆保持竖直不倒。

该系统的简化物理模型如图 1 所示，m_c 为小车的质量；m 为摆杆的质量；l 为摆杆的长度；θ 为摆杆与竖直方向的夹角；X 为小车移动的距离；I 为摆杆的转动惯量；F 为作用于小车的推力；F_f 为与 F 相反的摩擦力，摩擦系数为 b。

利用牛顿—欧拉方法建立的倒立摆数学模型为：

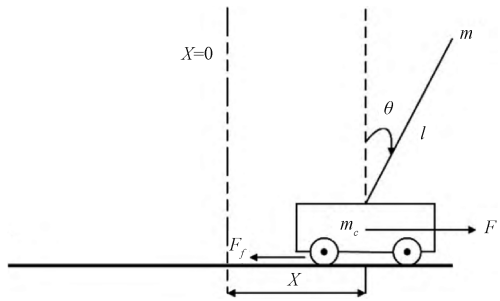


图 1　车杆模型示意图

$$\begin{cases}(m_c+m)\ddot{X}+b\dot{X}+m\dfrac{l}{2}\ddot{\theta}\cos\theta-m\dfrac{l}{2}\dot{\theta}^2\sin\theta=F\\ (I+m\dfrac{l^2}{4})\ddot{\theta}+mg\dfrac{l}{2}\sin\theta=-m\dfrac{l}{2}\ddot{X}\cos\theta\end{cases}\tag{1}$$

为了简化倒立摆系统分析，需将上述模型在平衡点进行线性化处理。需设 $\theta=\pi+\varphi$，在 θ 较小时，$\cos\theta=-1$、$\sin\theta=-\varphi$。线性化后的数学模型为：

$$\begin{cases}(m_c+m)\ddot{X}-m\dfrac{l}{2}\ddot{\phi}+b\dot{X}+m\dfrac{l}{2}\dot{\phi}^2\phi=F\\ (I+m\dfrac{l^2}{4})\ddot{\phi}-mg\dfrac{l}{2}\phi=m\dfrac{l}{2}\ddot{X}\end{cases}\tag{2}$$

1.2　算法结构

2018 年由 Haarnoja 等[14]提出的 SAC 算法，融合了随机策略和经验回放机制，并引入了最大化熵。算法在保障最大化累计奖励的同时，还要增强策略的随机性，不遗忘任何一个可能有用的动作和轨迹。熵与累计奖励的和期望最大的策略 π^* 表示为：

$$\pi^*=\arg\max_{\pi}E_{(s_t,a_t)\sim\pi}\left[\sum_{t=0}^{\infty}r(s_t,a_t)+\alpha H(\pi(\cdot\mid s_t))\right]\tag{3}$$

式中：$\sum_{t=0}^{\infty}r(s_t,a_t)$ 为 t 时刻倒立摆系统累积到的所有奖励；$H(\pi(\cdot\mid s_t))=-E_a[\log(\pi(a_t\mid s_t))]$ 为该时刻动作的熵；α 为温度系数，决定熵的权重，根据策略 π^* 来随机选择动作。

该算法共有 5 个神经网络：1 个 actor 演员策略网络 $\pi(s\mid a;\theta)$ 和 4 个 critic 评论家价值网络(包括 2 个目标 Q 网络 $\overline{Q}(s,a;\bar{\omega}_{1,2})$ 和 2 个 Q 网络 $Q(s,a;\omega_{1,2})$)，每次更新选择网络时会选择 Q 值比较小的那个，以缓解 Q 值的高估现象。

1)评论家网络

动作价值函数采用三层全连接网络拟合。智能体在 t 时刻的状态 s_t 和动作 a_t 为网络的输入，ReLU 函数 $(f(x)=\max(0,x))$ 作为激活函数，网络输出 a_t 的估计值。

Q 值网络的损失函数为 $Q(s,a)$ 函数和目标 $\bar{Q}(s,a)$ 函数的均方差：

$$L_Q(\omega)=\mathrm{E}_{(s_t,a_t)\sim D}\left[\frac{1}{2}(Q_\omega(s_t,a_t)-\bar{Q}_{\bar{\omega}}(s_t,a_t))^2\right] \tag{4}$$

其中,

$$\bar{Q}_{\bar{\omega}}(s_t,a_t)=r(s_t,a_t)+\gamma E_{(s_{t+1},a_{t+1})\sim\pi}(\bar{Q}_{\bar{\omega}}(s_{t+1},a_{t+1})-\alpha\log(\pi(a_{t+1}\mid s_{t+1})) \tag{5}$$

式中:ω 和 $\bar{\omega}$ 分别为 Q 网络和目标 Q 网络的网络参数;γ 为回报折扣因子;D 为经验缓冲池。

2)演员网络

动作策略函数采用三层全连接网络拟合。网络输入为状态 s_t,ReLU 作为激活函数。对于连续动作空间环境来说,策略网络的输出为高斯分布的均值和标准差,是不可导的,无法计算损失函数[15]。因此使用重参数化的方法,先从单位高斯分布 N 中获取采样值,将采样值乘以标准差后加上均值,可将 a_t 表示为 $a_t=f_\theta(\varepsilon_t;s_t)$,其中 ε_t 是噪声随机变量。

策略网络的损失函数由 KL 散度得到,同时考虑两个 Q 函数,重写化简后策略的损失函数为:

$$L_\pi(\theta)=\mathrm{E}_{s_t\sim D,\varepsilon_t\sim N}[\alpha\log(\pi_\theta(f_\theta(\varepsilon_t;s_t\mid s_t))-\min_{j=1,2}Q_{\omega j}(st,f_\theta(\varepsilon_t;s_t))] \tag{6}$$

3)自调节温度系数

为了提高策略的随机性,设置自调节温度系数[16],在倒立摆系统中,当最优动作未知时,增大 α 的值鼓励智能体随机探的索更多动作;当探索到最优动作时,减小 α 的值,将最优动作确定下来。因此将目标改写为一个约束问题:

$$\max_\pi \mathrm{E}_\pi\left[\sum_t r(s_t,a_t)\right]\ \ s.t.\ \mathrm{E}_{(s_t,a_t)\sim\rho\pi}[-\log(\pi_t(a_t\mid s_t))]\geqslant H_0 \tag{7}$$

设置熵的阈值 H_0,当 $H>H_0$ 时,α 减小,确定动作;反之增大 α 的值,鼓励探索。简化后自调节温度系数 α 的损失函数为:

$$L(\alpha)=\mathrm{E}_{s_t\sim D,\varepsilon_t\sim\pi(\cdot\mid s_t)}[-\alpha\log(\pi(a_t\mid s_t))-\alpha H_0] \tag{8}$$

2 算法改进

2.1 改进神经网络结构

激活函数可以增强神经网络的表示能力和学习能力,对训练效果有着至关重要的影响。目前,ReLU 函数已成为神经网络的默认激活函数,本文引入 Mish 激活函数作为 SAC 算法策略网络的激活函数进行训练,以提高神经网络体训练效率。

Mish 函数受 Swish 函数自门控性质的启发,是由 Misra[17]2019 年提出的一种自正则化的非单调函数。Mish 函数定义为:

$$f(x)=x\cdot\tanh(\ln(1+e^x)) \tag{9}$$

其中,$f(x)$ 的范围大概在 $[-0.31,\infty]$ 之间。Mish 函数是平滑的,非单调的,使得它区别于其他大多数常用的激活函数。Mish 函数的导数公式为:

$$f'(x)=\frac{e^x(4(x+1)+4e^{2x}+e^{3x}+e^x(4x+6))}{(2e^x+e^{2x}+2)^2} \tag{10}$$

Mish 函数和 Swish 函数是很相似的,又可将导数公式写为:

$$f'(x)=\mathrm{sech}^2(\mathrm{softplus}(x))\cdot x\cdot sigmoid(x)+\frac{f(x)}{x}=\Delta x\cdot swish(x)+\frac{f(x)}{x} \tag{11}$$

这里的 Δx 就是模仿了优化器的行为,给函数本身提供了一种更强的自正则化的效果,使梯度更平滑。Mish 函数的曲线和导数曲线图如图 2 所示。

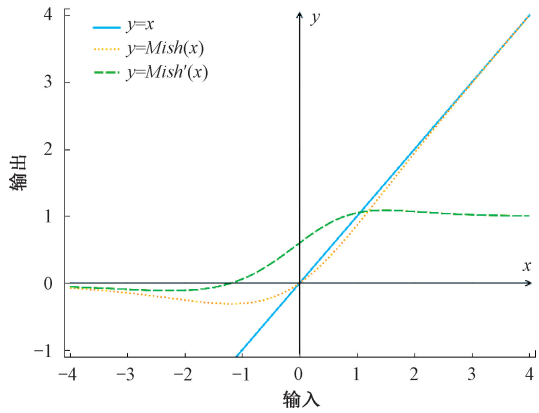


图 2 Mish 函数曲线和导数曲线图

由图 2 可知,Mish 函数是一个光滑、连续且非单调的激活函数,它在负值的时候允许比较小的负梯度流入,从而保证信息流动;其有上界无下界的特点,避免了饱和问题,不仅不会导致梯度消失,也保证了函数正则化的特性;较于 ReLU 激活函数负值区间的导数为 0,Mish 函数避免了神经元的"坏死现象"。Mish 函数又是连续可微的,避免了奇异点,在执行基于梯度的优化时避免了不必要的副作用。Mish 函数的性能相比函数更好,而且随着网络的深度加深,信息可以更深入的流动。在多数实验条件下,Mish 函数相比其他激活函数获得了更好的训练结果[18-19]。

2.2 近端经验采样策略

智能体与环境每交互一次会生产一个五元数组(s_t,a_t,r_t,s_{t+1},$done$),并存入经验缓存池中,即智能体当前所处的状态,当前执行的动作,下一时刻的状态,执行动作获得的奖励值以及当前回合是否结束。

传统 SAC 算法在更新网络时会从经验缓存池进行批量化的随机采样。这种方式是假定缓存池中的每条数据都具有同等的重要性[20],但实际情况是会有许多价值不高的数据被采样,会导致算法的效果和收敛速度受到影响。因此提出一种结合近端经验回放的策略 RER_SAC(recency experience replay soft actor critic),强调在对缓存池进行批量化随机采样的时候,着重考虑近期放进去的数据,同时也

不能忘记以前的经验[21]，该策略的经验回放过程如图 3 所示。

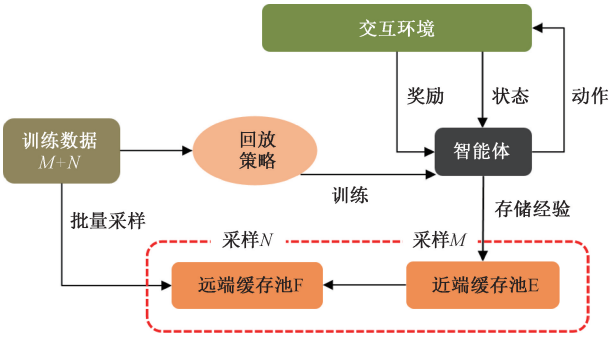


图 3　近端经验回放过程

由图 3 可知，所提的近端经验回放策略 RER_SAC 的经验缓存设计为：

1）设置近端缓存池 E 和远端缓存池 F；

2）缓存池 E 的容量为 m，缓存池 F 的容量为 n；

3）两个缓存池都为先入先出的队列；

4）智能体与境交互产生的五元组会首先存入近端缓存池 E 中；

5）当近端缓存池 E 填满之前，此时缓存池 F 是空的，训练所需批量数据从 E 中采样；

6）当近端缓存池 E 填满之后，交互产生的新数据会淘汰近端缓存池中最早的一条数据，将淘汰的数据作为新数据加入远端缓存池 F 中，并将新数据存入缓存池 E 中，并而当 F 存满之后，也将抛弃最早的数据；

7）此时分别从近端缓存池 E 中采样 M 条数据，从远端缓存池 F 中采样 N 条数据，共采样 $M+N$ 条数据送入神经网络训练更新；

8）继续下一步交互更新。

2.3　回放训练策略

随着训练的进行，缓存池里数据不断增加，但放入缓存池中数据的频次不变，一定范围内会导致缓存池中新数据的比例变小，使得一些有用数据可能没有被采样到，就被淘汰了。因此在适用上文近端经验采样策略的基础上，提出一种分段近端经验回放算法 PRER_SAC，随着池内新数据比例的下降，根据缓存池中数据的多少增加智能体的训练次数，将数据最大化利用。分别设经验缓存池 E 和 F 的最大容量分别为 C_{max_E}，C_{max_F}，且 $C_{max\_E}<<C_{max\_F}$；缓存池的最小容量都为 C_{min}。将交互过程产生的经验数据逐条放入缓存池，经验缓存池的当前数据量分别为 C_E，C_F；每次批量采样的数据量为 B；其批量数据采样策略适用图 3 的步骤，具体的训练策略为：

1）当缓存池 E 的容量满足 $C_{min}\leqslant C_E\leqslant C_{max\_E}$，且缓存池 F 的容量 $C_F\leqslant C_{min}$ 时，每增加 100 组新数据，训练智能体 10 次；

2）当缓存池 E 的容量满足 $C_E=C_{max\_E}$，且缓存池 F 的容量 $C_{min}<C_F<0.3C_{max\_F}$ 时，每增加 100 组新数据，训练智能体 20 次；

3）当缓存池 E 的容量满足 $C_E=C_{max\_E}$，且缓存池 F 的容量 $0.3C_{max\_F}\leqslant C_F<C_{max\_F}$ 时，每增加 100 组新数据，训练智能体 30 次；

4）当缓存池 E 的容量满足 $C_E=C_{max\_E}$，且 $C_F=C_{max\_F}$ 时，每增加 100 组新数据，训练智能体 40 次。

2.4　算法流程

以小车倒立摆系统为研究对象，结合上述算法设计，应用到该系统中。图 4 所示为基于该算法的倒立摆 DRL 控制过程。PRER_SAC 的算法流程如下：

1）初始化 actor 网络参数 θ，critic 网络参数 ω_1，ω_2，$\bar{\omega}_1$，$\bar{\omega}_2$ 及其他参数。

2）初始化经验缓存池 E，F；

3）每个回合(episode)循环以下步骤；

4）获取倒立摆初始状态 s_t；

5）对回合里的每一步(step)循环以下步骤；

6）根据策略选择动作 $a_t=\pi(s_t)$；

7）执行动作 a_t，获得奖励 r_t，环境状态变为 s_{t+1}；

8）将(s_t,a_t,s_{t+1},r_t)依次存入近端缓存池 E，F 中；

9）当 $C_{\min}\leqslant C_E\leqslant C_{\max\_E}$ 且 $C_F\leqslant C_{\min}$ 时，采样在 E 里进行；当存至 $C_{\min}<C_F$ 后，在 E 里采样 M，F 里采样 N，总采样 $M+N$；

10）缓存池每存入 100 组数据，进行分段训练；

11）对每组数据，根据式(5)计算目标 Q 值；

12）对两个 critic Q 网络，根据式(4)计算损失函数，更新当前 critic 网络；

13）重参数采样获取动作 a_t，根据式(6)计算损失函数，更新当前 actor 网络。

14）根据熵的阈值和式(8)更新熵的系数 α；

15）更新目标网络参数，即：$\bar{\omega}_1\leftarrow\tau\omega_1+(1-\tau)\bar{\omega}_1$ 和 $\bar{\omega}_2\leftarrow\tau\omega_2+(1-\tau)\bar{\omega}_2$；

16）结束每步(step)循环；

17）结束每回合(episode)循环。

3　仿真测试

3.1　仿真平台搭建

为了验证所提算法的有效性，进行仿真实验。本文实验的操作系统为 64 位 Window10，内存为 8 GB，处理器为 Intel(R) Core(TM) i5-8250U CPU@1.60 GHz。

定义小车倒立摆环境，当摆杆竖直向上时，摆杆角度 θ 为 0，向小车施加一个连续的[−1,1] N 的力，使小车左右摆动保持摆杆平衡。其中，摆杆初始角度在[−4°,4°]间随机取值，当摆杆角度满足 $\theta\in[-12°,12°]$时，稳摆成功。小车位置 x 的范围不能超出[−2.4,2.4]的单位长度。仿真环境示意图如图 5 所示。

智能体在 t 时间的状态 s 用向量[x，$\dot{x}$，θ，$\dot{\theta}$]表示，分

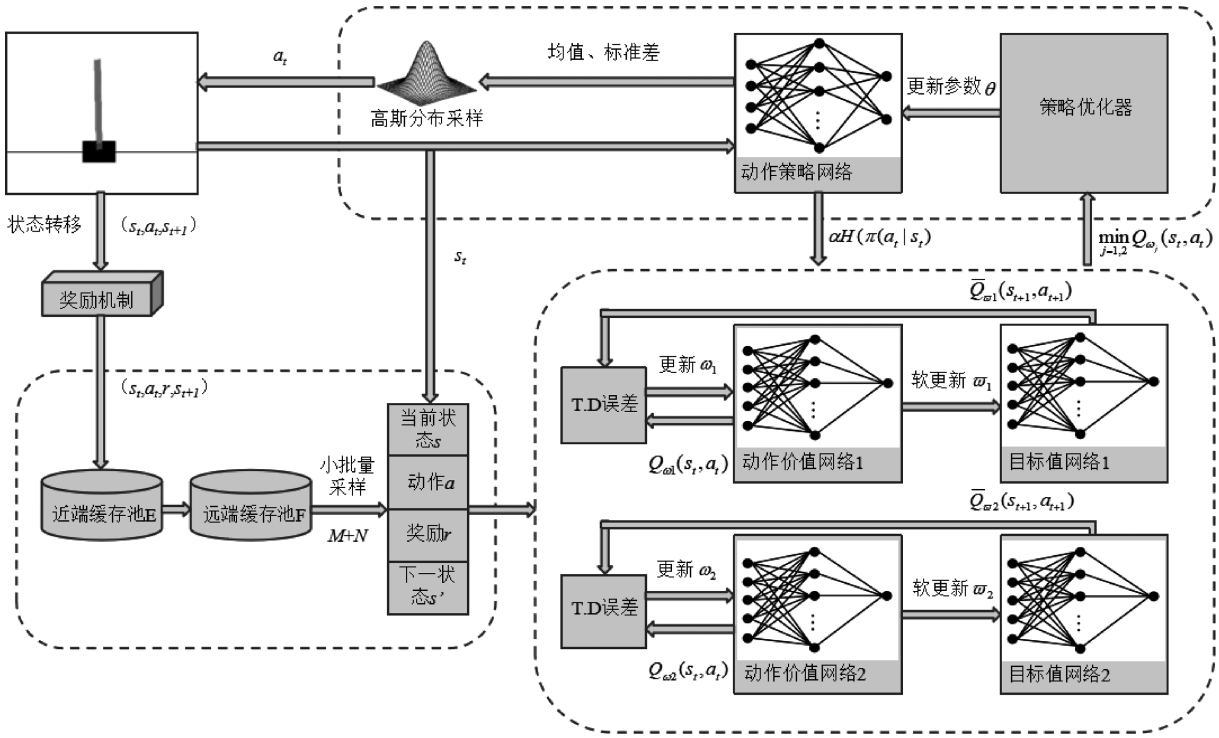


图 4 倒立摆 DRL 控制过程

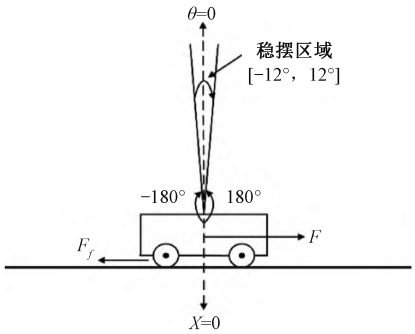


图 5 仿真环境示意图

别代表小车位置、小车速度、杆子与竖直方向的夹角和角度变化率,其中小车速度和角度变化率不设值域范围。其余主要参数如表 1。

表 1 倒立摆的主要参数

环境参数	取值
小车质量/kg	1
摆杆质量/kg	0.1
摆杆长度一半/m	0.5
每回合最大步数/step	200
摩擦系数	0.1

在训练过程中,设环境与智能体总交互 500 个回合,每个回合 200 步,每个 Episode 的结束条件为:

1) θ 超出[-12°,12°]的范围;

2) x 超出[-2.4,2.4]的单位长度;

3) 200 个 step 全部走完。

每个回合内,每保持 1 步平衡得到的 reward 为 1。若连续 100 回合奖励都保持在 200,则智能体训练完成。

3.2 实验 1:Mish 函数对训练效果的影响

为验证本文引入的 Mish 激活函数在训练中的优势,策略网络采用 3 层全连接网络,为避免深层网络结构对实验数据的影响,设置隐含层神经元个数为 16。隐藏层的激活函数分别选取 Sigmoid、ReLU、sTanh、Swish 函数以及本文引入的 Mish 激活函数。随机初始化神经网络的权重值,计算该仿真环境初始状态下 100 回合的平均回报值,结果如表 2。

表 2 不同激活函数的平均回报

激活函数	平均回报
Mish	50.13
Swish	48.47
ReLU	42.37
Tanh	39.35
Sigmoid	19.33

由表 2 可知,策略网络在采用了 Mish 激活函数后,在相同的网络结构下,在 100 回合内可以获得更高的回报值,且远高于 Sigmoid 函数,相较于常用的 ReLU 函数,性能提高了 18.4%;相比与之性质相似的 Swish 函数,性能提高了 3.5%,具有更优的算法性能。Mish 激活函数避免

了神经元的“坏死现象”，而且随着网络的深度加深，信息可以更深入的流动，训练效果会更好更明显。

3.3　经验采样策略对训练效果的影响

实验算法超参数的确定：折扣因子用来计算未来累计奖励，折扣因子越小越肯定当前的回报；神经网络学习率决定着每次权重更新的步长大小，影响神经网络的训练过程和结果；批量大小越大，训练的精度越高，但会使神经网络梯度变化减缓，从而无法走出局部最优点；软更新系数控制网络之间的权衡，较小的话会导致更新相对较快，也可能使网络不稳定；温度系数用来调节智能体不同状态下的探索策略；熵的阈值决定温度系数的自更新，一般取动作维度的相反数。通过实验 1，策略网络引入 Mish 激活函数，多次训练比对 100 回合内的平均回报值，确定出较优的算法超参数如表 3。

表 3　神经网络训练参数

参数名	取值
折扣因子	0.99
学习率	0.001
批量大小	256
软更新系数	0.05
初始温度系数	0.01
熵的阈值	−1

为验证提出的近端经验采样策略，遵从增大近期经验在样本数据中重要性的原则，该策略中近端缓存池 E 的容量大小 m 以及该缓存池中小批量采样的数据 M 的占比都对算法的性能起到至关重要的作用，因此本节将从以下两点展开实验：

1）实验 2：近端经验采样数据量 M

为了避免近端缓存池大小 m 对实验结果的影响，设置近端缓存池大小与小批量数据中近端经验数据量相同，即 $M=m$。分别取 $M=8,16,32,56,128$ 进行实验，每 10 个回合的平均回报曲线如图 6 所示。

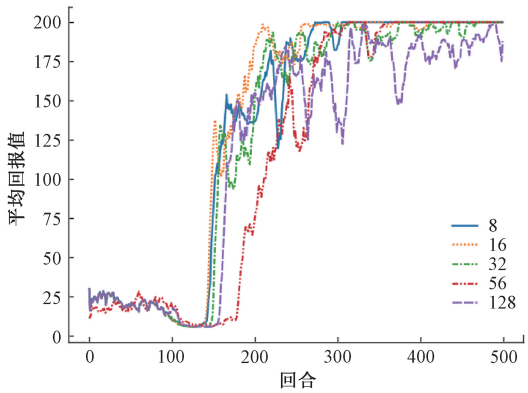


图 6　不同数据量的平均回报

由图 6 可知，M 的取值在从 128 依次减小到 16 时，M 的取值越小，算法的收敛性能越好。M 取值为 16 的时候算法性能最佳，但在 $M=8$ 的时候效果反而变差了。证明当小批量采样数据中近端数据量 M 占比较低时，算法可以更好的兼顾近端数据和远端数据的平衡。

2）实验 3：近端缓存池 E 的容量 m

由实验 2 得，在 $M=m$ 的前提下，m 此时不发挥作用，确定了在近端经验缓存池采样数 $M=16$ 的时候，算法性能最佳。因此，为了使 M 和 m 共同在实验中发挥作用，取最佳 $M=16$，分别取 $m=1\,000,1\,500,2\,500,7\,500,10\,000$ 进行实验，每 10 个回合的平均回报曲线如图 7 所示。

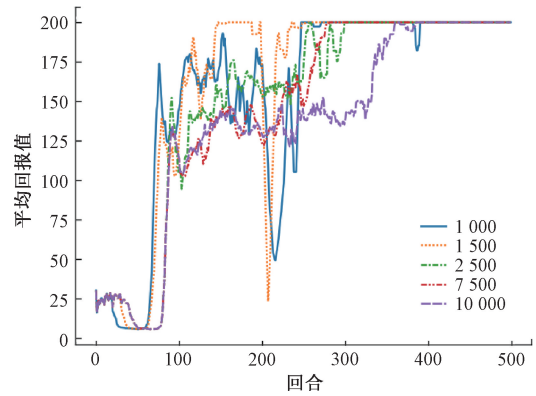
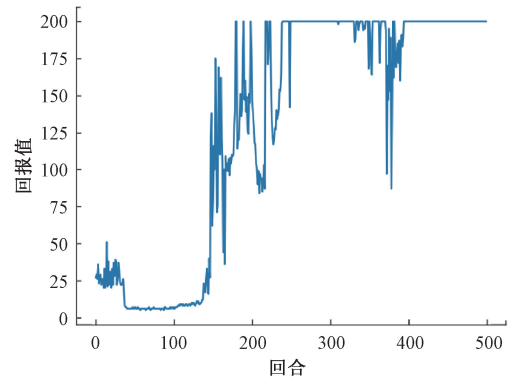


图 7　不同经验池大小的平均回报

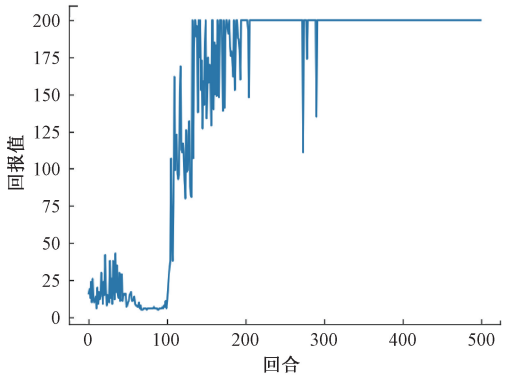
由图 7 可知，$M=16$ 的情况下 m 的取值在从 10 000 依次减小到 1 500 时，m 的取值越小，算法的收敛性能越好。m 取值为 1 500 的时候算法性能最佳，但在近 200 个回合的时候，回报值发生了大幅度降低。但在 $m\geqslant 2\,500$ 的情况下，却没有这种情况。因此可得，m 的最优取值应该在 $1\,500<m<2\,500$ 之间。

3.4　实验 4：强化学习算法性能对比

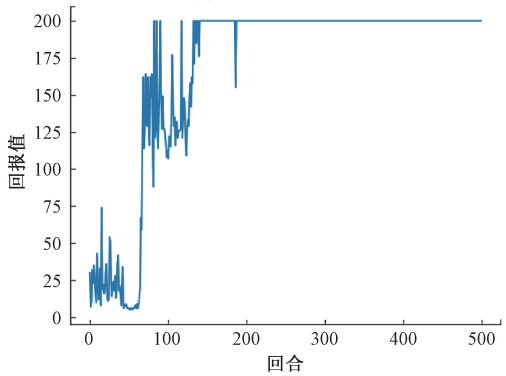
结合上述实验，取近端缓存池 E 的容量 $C_{max\_E}=m=2\,000$，批量采样数据量 $M=16$；远端缓存池 F 的容量 $C_{max\_F}=n=100\,000$，批量采样数据量 $N=256-M$。分别对比了 DDPG，SAC，RER_SAC 和 PRER_SAC 算法的训练效果，算法收敛条件为连续 100 回合回报值保持在 200，回报曲线如图 8 所示，训练数据如表 4。



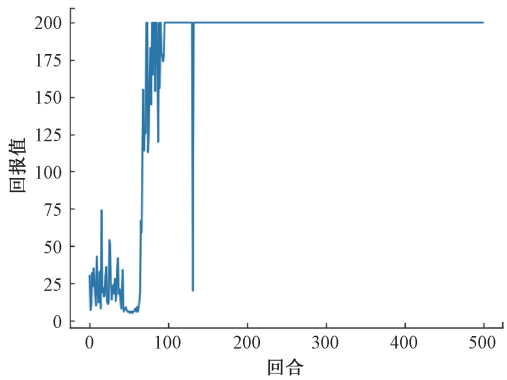
(a) DDPG算法回报曲线



(b) SAC算法回报曲线



(c) RER_SAC算法回报曲线



(d) PRER_SAC算法回报曲线

图 8 不同算法的回报曲线

表 4 不同算法的训练效率

算法名	回报首次达到 200/eposides	首次达到 200 的时间/min	收敛回合/eposides	收敛时间/min
DDPG	180	1.27	495	9.34
SAC	143	1.06	392	19.22
RER_SAC	83	0.46	285	16.03
PRER_SAC	**73**	**0.33**	**233**	**13.56**

由图 8 可知,PRER_SAC 算法在 500 个回合的训练中表现最优,其训练回报值方差明显变小,回报曲线的波动变小,且上升的更快更稳定,有更好的算法性能。

由表 4 可知,PRER_SAC 训练在 73 回合时回报值首次达到 200,历经 0.33 min;4 种算法均可在倒立摆训练中收敛,PRER_SAC 算法比 RER_SAC 算法快 52 个回合,收敛速率提高了 15.4%;比 SAC 算法快 159 个回合,收敛速率提高了 29.4%;可见,PRER_SAC 算法训练完成用时最短,明显快于其他算法。由于 SAC 算法结构比较复杂,参数也较多,所以其总体训练时间比 DDPG 算法长,但其控制效果比 DDPG 算法更稳定。

实验证明了 PRER_SAC 算法能着重考虑近期产生的数据,并根据已产生的数据量进行分段训练,能更快速有效地探索到系统最优动作,一定程度避免了对无用数据的学习,提高了数据利用率和算法的训练效率。

3.5 实验 5:小车倒立摆控制效果对比

为验证本文所提方法的控制效果,根据文献[22~25]采用传统控制方法 PID 和 LQR 与本文提出的 PRER_SAC 算法进行仿真对比。

由图 9 可知,3 种方法均可在系统初始角度偏差较大的情况下,在较短时间内进入稳摆阶段。其中,本文提出的 PRER_SAC 算法相比 PID 和 LQR 方法花费时间最短,在 6.5 s 的时刻之后,即可使摆杆稳摆角度一直保持在[−0.1°,0.1°]的偏差范围内,证明了所提算法的优越性。

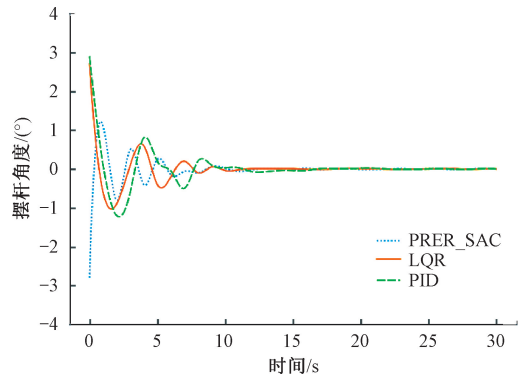
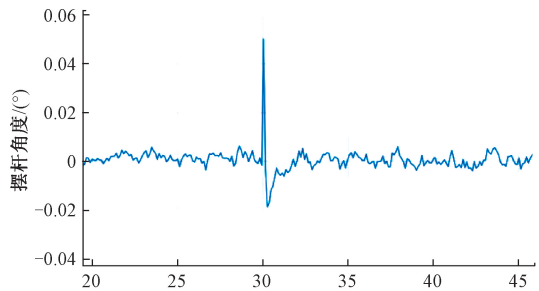


图 9 不同方法的摆杆角度曲线

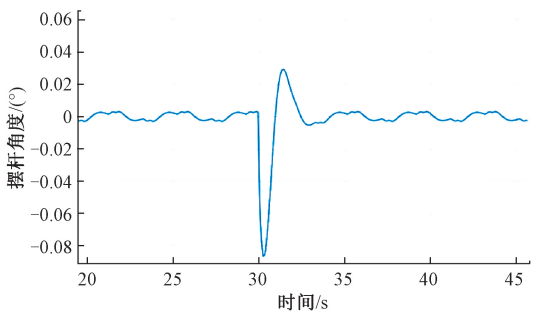
3.6 实验 6:鲁棒性测试

在通过不同算法训练好的智能体上,在稳摆的第 30 s 加入角度扰动,测试系统的鲁棒性,扰动下的小车摆杆角度曲线如图 10 所示。

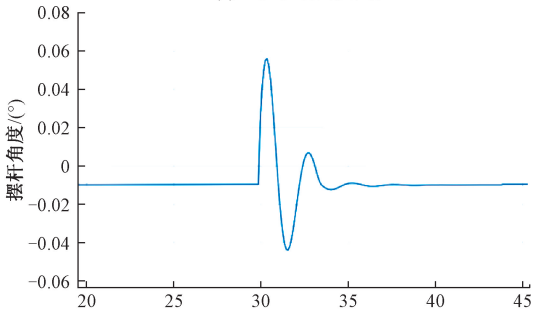
由图 10 可知,通过 PRER_SAC 算法训练好的系统,相比 PID 和 LQR 方法,系统的反应更迅速,摆杆角度的偏



(a) PRER_SAC扰动角度变化



(b) LQR扰动角度变化



(c) PID扰动角度变化

图 10　不同方法摆杆扰动角度变化

差可在 2 s 内被有效地消除和抑制，之后一直保持在[−0.005°，0.005°]的范围内，系统的稳定性更强，证明了所提方法具有较强的鲁棒性和优越性。

4　结　　论

提出一种基于柔性演员-评论家的 PRER_SAC 控制算法。以小车倒立摆为实验环境，对比仿真结果表明，所提算法相比原 SAC 算法收敛速率提高了 29.4%，提高了算法性能；对比 PID 和 LQR 传统算法，有较强的自主控制能力，仅需通过试错式的训练而不需要人为调整即可完成控制任务，且该算法具有较强的鲁棒性，可以在倒立摆稳摆控制中达到设定要求，具有较强的适用性。

参考文献

[1]　刘剑. 欠驱动非线性系统控制问题的研究[D]. 哈尔滨：哈尔滨工业大学，2017.

[2]　杜璧秀，张淑梅，高慧斌，等. 基于 T-S 模型的小车倒立摆控制[J]. 电子测量技术，2012，35(9)：56-59.

[3]　黎君，阎世梁. 一级倒立摆模糊 PID 控制器设计[J]. 国外电子测量技术，2012，31(4)：50-52.

[4]　CORREA-RAMÍREZ D V，GIRALDO-BUITRAGO D，ESCOBAR-MEJÍA A. Fuzzy control of an inverted pendulum Driven by a reaction wheel using a trajectory tracking scheme[J]. TecnoLógicas，2017，20(39).

[5]　何卫东，刘小臣，张迎辉，等. 深度强化学习 TD3 算法在倒立摆系统中的应用[J]. 大连交通大学学报，2023，44(1)：38-44.

[6]　刘微微，张静. 单级倒立摆 LQR 控制方法的鲁棒稳定性分析[J]. 黑龙江水专学报，2010，37(2)：105-108.

[7]　王珏，谢慕君，李元春，等. 基于滑模变结构的柔性倒立摆控制研究[J]. 计算机测量与控制，2015，23(12)：4045-4048.

[8]　张浩杰，苏治宝，苏波. 基于深度 Q 网络学习的机器人端到端控制方法[J]. 仪器表学报，2018，39(10)：36-43.

[9]　宁强，刘元盛，谢龙洋. 基于 SAC 的自动驾驶车辆控制方法应用[J]. 计算机工程与应用，2023，59(8)：306-314.

[10]　王军，杨云霄，李莉. 基于改进深度强化学习的移动机器人路径规划[J]. 电子测量技术，2021，44(22)：19-24.

[11]　杨文乐. 基于强化学习的倒立摆控制算法研究[D]. 西安：西安理工大学，2019.

[12]　翁士博. 深度强化学习 SAC 算法在运动控制中的应用研究[D]. 西安：西安理工大学，2020.

[13]　王雨轩，陈思溢，黄辉先. 基于改进深度强化学习的倒立摆控制器设计[J]. 控制工程，2022，29(11)：2018-2026.

[14]　HAARNOJA T，ZHOU A，ABBEEL P，et al. Soft actor-critic：Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]. International con ference on machinelearning，2018：1861-1870.

[15]　赵克刚，石翠铎，梁志豪，等. 基于柔性演员-评论家算法的自适应巡航控制研究[J]. 汽车技术，2023(3)：26-34.

[16]　HAARNOJA T，ZHOU A，HARTIKAINEN K，et al. Softator-critic algorithms and applications[J]. ArXiv Preprint，2018，ArXiv：1812.05905.

[17]　MISRA D. Mish：A self regularized non-monotonicactivation function[J]. ArXiv Preprint，2019，ArXiv：1908.08681.

[18]　唐佳强. 基于深度学习的安保机器人危险品探测与识别[D]. 天津：河北工业大学，2021.

[19]　宋倩，罗富贵. 基于手写体数字识别的激活函数对比研究[J]. 现代信息科技，2023，7(4)：95-97.

[20]　程艳. 基于深度强化学习的智能体自适应决策能力的生成[D]. 济南：山东大学，2021.

[21]　贲松. 针对异策略强化学习的优化算法研究[D]. 成都：电子科技大学，2021.

[22]　翟彦彦. 一级倒立摆模糊控制、LQR 控制和 PID 控制的比较研究[J]. 电子设计工程，2016，24(7)：116-119，124.

[23]　张凯，郁豹. 单级倒立摆的 PID 和 LQR 控制效果的比较[J]. 工业控制计算机，2017，30(8)：111-112，114.

[24]　姜海燕. 直线一级倒立摆的 PID 控制方法研究[J]. 河南科学，2019，37(6)：908-913.

[25]　程丽梅，贾文川. 连续型强化学习与 PID 控制的应用对比分析：以一阶倒立摆系统为例[J]. 工业控制计算机，2021，34(10)：20-22.

作者简介

张晓莉，硕士，副教授，主要研究方向为嵌入式应用、智能控制技术。

E-mail：zhxl_1205@163.com

郭仕林(通信作者)，硕士研究生，主要研究方向为强化学习与智能控制。

E-mail：1347696642@qq.com