基于自生成专家样本的探索增强算法

刘　健†, 赵恒一

(中国矿业大学 信息与控制工程学院, 江苏 徐州 221116)

摘要: 为进一步提高深度强化学习算法在连续动作环境中的探索能力, 以获得更高水平的奖励值, 本文提出了基于自生成专家样本的探索增强算法. 首先, 为满足自生成专家样本机制以及在连续动作环境中的学习, 在双延迟深度确定性策略梯度算法的基础上, 设置了两个经验回放池结构, 搭建了确定性策略算法的总体框架. 同时提出复合策略更新方法, 在情节的内部循环中加入一种类同策略学习过程, 智能体在这个过程中完成对于参数空间的启发式探索. 然后, 提出基于自生成专家样本的演示机制, 由智能体自身筛选产生专家样本, 并根据参数的更新不断调整, 进而形成动态的筛选标准, 之后智能体将模仿这些专家样本进行学习. 在OpenAI Gym的8组虚拟环境中的仿真实验表明, 本文提出的算法能够有效提升深度强化学习的探索能力.

关键词: 深度强化学习; 探索; 专家样本; 确定性策略

引用格式: 刘健, 赵恒一. 基于自生成专家样本的探索增强算法. 控制理论与应用, 2023, 40(3): 485 – 492

DOI: 10.7641/CTA.2021.10552

Enhance exploration with self-generated expert samples

LIU Jian†, ZHAO Heng-yi

(School of Information and Control Engineering, China University of Mining and Technology, Xuzhou Jiangsu 221116, China)

Abstract: In order to further improve the exploration ability of the deep reinforcement learning algorithm in the continuous action environment, so as to obtain a higher level of reward value, an algorithm named enhance exploration with self-generated expert samples is proposed. First of all, to satisfy the self-generated expert samples mechanism and learning in the continuous action environment, on the basis of twin delayed deep deterministic policy gradient algorithm, we set up two experience replay structures and build the overall framework of the deterministic policy algorithm. Meanwhile, a combined policy update method is proposed. The approximate on-policy learning process is added to the internal loop of the episode. The agent completes the heuristic exploration of the parameter space in this process. Secondly, a demonstration mechanism based on the self-generated expert samples is proposed. Expert samples are generated by the agent's own selection, while the criteria are continuously adjusted according to the update of parameters, which could form dynamic screening criteria. After that, the agent will imitate these expert samples for learning. Simulation experiments in 8 environments in the OpenAI Gym show that the proposed algorithm can effectively improve the exploration ability of deep reinforcement learning.

Key words: deep reinforcement learning; exploration; expert sample; deterministic policy

Citation: LIU Jian, ZHAO Hengyi. Enhance exploration with self-generated expert samples. *Control Theory & Applications*, 2023, 40(3): 485 – 492

1　引言

强化学习[1–2]是一种基于估计的机器学习范式, 智能体仅能明确某个动作的奖励值, 但无法确定其是否是最优值. 正因如此, 探索对强化学习能否取得良好效果至关重要, 这决定了算法能否收敛至全局最优. 在具有高维原始数据、连续动作域的仿真环境中[3–6], 探索的重要性尤为突出.

策略梯度方法一般被用于处理连续动作环境中的控制问题, 但其本身拥有方差大、无法单步更新等缺点. 值函数方法无法直接处理连续域问题, 通常的解决方法是将连续动作离散化, 然而离散化却会造成信息损失. 将策略梯度方法与值函数方法相结合, 用深度神经网络作为函数估计器来估计动作的值函数, 可在一定程度上弥补策略梯度和值函数方

收稿日期: 2021−06−27; 录用日期: 2021−10−09.

†通信作者. E-mail: liujiansqjxt@126.com; Tel.: +86 15905216271.

本文责任编委: 苏剑波.

国家自然科学基金项目(61906198), 江苏省自然科学基金项目(BK20190622)资助.

Supported by the National Natural Science Foundation of China (61906198) and the National Natural Science Foundation of Jiangsu Province (BK20190622).

法的缺陷, 此类方法最具代表性的是Actor-Critic系列算法[7–10].

相对于随机性策略(stochastic policy)而言, 确定性策略(deterministic policy)算法的动作值输出$a=\mu_\theta(s)$是一个确定值, 具有对数据的需求量小, 算法效率高的优点, 但无法保证对环境的充分探索. Silver[11] 提出使用异策略的结构来弥补这种确定性造成的探索损失, 在行为策略中采用贪心策略来保证算法具有一定的随机性. 熵正则化方法[12–14]为目标函数添加一个正则惩罚项, 根据熵值大小调整模型在状态空间中的探索力度, 如置信域策略梯度优化算法(trust region policy optimization, TRPO)[12]采取在随机策略中增加熵的方法; Bellemare等[15]通过对状态访问进行计数, 得到访问次数$\widehat{N}(s)$, 提供额外奖励$B(\widehat{N}(s))=\sqrt{\widehat{N}(s)^{-1}}$; Pathak等[16]提出利用好奇心机制, 为奖励函数增加一个内在激励项. 此外, 一些基于后验采样的探索方法[17–18], 以及基于概率近似的方法[19], 在采样之后都会修正策略的概率分布进行探索. 上述探索方法大多是启发式的, 或对数据的需求量过大、收敛速度太慢, 同时一些对奖励函数的更改可能会导致算法无法收敛. 基于最大熵思想的自模仿学习(self-imitation learning, SIL)[20]算法在更新时只会计算优秀样本的梯度, 这样就极大的提升了探索的效率. 启发于SIL算法, 本文提出一种基于自生成专家样本的深度确定性策略算法来增强智能体在连续动作环境中的探索能力, 以获得更高的奖励.

本文在Actor-Critic结构的基础上采用确定性策略梯度[11]提出基于自生成专家样本的探索增强算法(enhance exploration with self-generated expert samples, SGES), 在一次情节中, 算法首先根据回合经验池中的样本进行一次参数更新, 并根据折扣累积奖励与值函数的对比筛选专家样本并存入专家样本池, 然后智能体在专家样本池中进行随机采样, 获得小批量样本集以进行异策略学习. 本文在OpenAI Gym中进行了8组实验, 采取深度确定性策略(deep deterministic policy gradient, DDPG)[21]和双延迟深度确定性策略梯度算法(twin delayed deep deterministic policy gradient, TD3)[22]同SGES进行对比, 实验结果表明SGES算法的效果优于对比算法.

本文的创新之处在于:

1) 构建了能够融合自生成专家样本机制和复合策略学习方式的模型结构, 增设回合经验池与专家样本经验池, 分别用于储存轨迹数据并计算折扣累积奖励和筛选出的专家经验. 这样的设计没有增加网络结构, 多出的回合经验池每个情节结束后都会清零, 不会过多增加算法的内存占用.

2) 采用复合策略更新的方式, 利用回合经验池中的样本进行类同策略学习. 类同策略学习是一种近似同策略的学习方法, 本文希望加入这种策略更新方法, 保证算法在整体的样本空间中的随机探索.

3) 利用在回合经验池中筛选出来的专家样本指导算法进行异策略学习, 提供更具有指导性的探索, 从而获取更高的奖励值.

2 Actor-Critic结构算法

强化学习问题可以建模为马尔可夫过程(Markov decision process, MDP). MDP由元组(S,A,P,r,γ)表示: S为有限的状态集; A为有限的动作集; P为状态转移概率. $P(s_{t+1}|s_t,a_t)$表示在环境处于状态s_t处采取动作a_t获得状态s_{t+1}的概率, 强化学习处理的问题需满足马尔可夫性质, 即

$$P(s_{t+1}|s_t,a_t)=P(s_{t+1}|s_1,a_1,\cdots,s_t,a_t),$$

其中: $s_1,a_1,\cdots,s_t,a_t,\cdots,s_{\rm T},a_{\rm T}$为状态–动作空间内的任意轨迹(trajectory); T为任意轨迹的时间步, t为时间步T内的第t个时刻; 奖励函数$r:S\times A\to\mathbb{R}$; γ为折扣累计因子, 用来计算累计回报. 一个随机策略记为$\pi_\theta(a_t|s_t)$, 表示在s_t处采取动作a_t的概率分布, $\theta\in\mathbb{R}^n$是一个表示参数化策略的n维向量. 智能体在当前策略的指导下, 与MDP交互获得一个轨迹的数据$H_{1:{\rm T}}=s_1,a_1,r_1,\cdots,s_t,a_t,r_t,\cdots,s_{\rm T},a_{\rm T},r_{\rm T}$. t时刻的折扣累积回报为

$$R_t=\sum_{k=0}^{\infty}\gamma^k r_{t+k+1},0<\gamma<1.$$

策略通过状态值函数$V^\pi(s_1)=E[R_t|s_1;\pi]$来衡量. $V^\pi(s_1)$表示从初始状态执行策略π得到的折扣累积回报的期望. 动作值函数$Q^\pi(s_1,a_1)=E[R_t|s_1,a_1;\pi]$表示在策略$\pi$指导下, 当前状态$s_t$采取动作$a_t$后, 所能获得的预期回报. 强化学习的目标是获得一个能够使得折扣累积回报最大的最优策略π^*, 用$J(\pi_\theta)=E[R_t;\pi_\theta]$来表示强化学习的目标函数. 策略梯度要求按照目标函数的梯度方向调整参数θ, 其基本公式为

$$\nabla_\theta J(\pi_\theta)=\int_S\rho^\pi\int_A\nabla_\theta\pi_\theta(a|s)Q^\pi(s,a){\rm d}s{\rm d}a,\quad(1)$$

其中ρ^π是执行策略π时的状态分布. 通常, $\nabla_\theta J(\pi_\theta)$很难直接通过计算得到, 可以通过采样求均值的方式来拟合策略梯度, 即

$$\nabla_\theta J(\pi_\theta)=E_{s\sim\rho^\pi,a\sim\pi_\theta}[\nabla_\theta\log\pi_\theta(a|s)Q^\pi(s,a)].\quad(2)$$

结合Q学习方法, 设定一个函数估计网络(critic), 记为$Q^\omega,\omega\in\mathbb{R}^n$. Q^ω参数的选取标准是可以使得真实值函数与估计器之间的均方误差$\varepsilon^2(\omega)=E_{s\sim\rho^\pi,a\sim\pi_\theta}[Q^\omega(s,a)-Q^\pi(s,a)]^2$最小化, 但其真实值是未知的, 可以用差分目标网络值代替. 如果利用合适的策略估计算法, Q^ω最终将近似等于真实Q值.

Actor-Critic算法策略更新公式如下:

$$\nabla_\theta J(\pi_\theta) = E_{s\sim\rho^\pi, a\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s) Q^\omega(s,a)]. \tag{3}$$

基于确定性策略的Actor-Critic方法, 采用确定性策略$\mu_\theta: S \to A$, 其梯度公式如下:

$$\nabla_\theta J(\mu_\theta) \approx E_{s\sim\rho^\mu}[\nabla_\theta \mu_\theta(s) Q^\omega(s,a)]. \tag{4}$$

值函数网络使用基于时间差分误差的增量学习方法[23]进行更新, 如

$$\delta_t = r_t + \gamma Q^\omega(s_{t+1}, \mu_\theta(s_{t+1})) - Q^\omega(s_t, a_t). \tag{5}$$

采用单步更新的方法无须计算状态的整体分布, 而环境的即时反馈r仅由环境确定, 与所使用的策略无关. 因此, 采用Q学习的一些异策略方法, 如DDP, TD3等, 在学习时也无需重要性采样. DDPG和TD3都建立在Actor-Critic的基础上, DDPG算法采用Double DQN (double deep Q network)[24–25]的思想与确定性策略梯度相结合, 建立两个Actor-Critic网络分别作为当前行为网络θ和估计网络ω以及对应的目标网络θ'和ω', 使用目标网络计算下一步动作$a_{t+1} = \mu(s_{t+1};\theta')$和差分目标$r_t + \gamma Q(s_{t+1}, a_{t+1};\omega')$. 目标网络的参数采取软更新形式从当前网络得到, 即$\theta' \leftarrow \tau\theta + (1-\tau)\theta'$和$\omega' \leftarrow \tau\omega + (1-\tau)\omega'$, 其中$\tau$为软更新系数. TD3算法大体采取了DDPG的结构, 但是其各自的估计网络是孪生的, 在更新时会选择输出值较小的网络作为目标估计网络.

3　自生成专家样本深度强化学习

3.1　策略更新

确定性策略仅在状态空间上进行运算, 较之随机策略其是在状态空间和动作空间的双重积分. 本文采用效率更高的确定性策略方法. 对Q值的过高估计是Q学习方法计算贪心目标时固有的结构性问题, 因此, 结合SGES的特点, 将SGES与TD3进行结合, 设置当前网络的ω, θ和目标网络ω', θ', 其具体作用是: θ根据当前状态s选择当前最优动作a, 同环境交互生成s'和r. 参数θ通过下式进行梯度更新:

$$\nabla_\theta J \approx N^{-1} \sum_i \nabla_{a\sim\mu(s)} Q(s,a;\omega) \nabla_\theta \mu(s;\theta). \tag{6}$$

θ'根据下一状态s'选择最优下一动作a', 网络参数定期从θ处获取. 利用ω计算当前状态–动作价值$Q(s,a;\omega)$, 其中ω的参数更新方式为

$$\omega \leftarrow \arg\min_\omega N^{-1} \sum_i (y - Q(s,a;\omega))^2. \tag{7}$$

ω'负责计算估计网络更新时的差分目标$y = r + \gamma Q(s', \mu(s';\theta');\omega')$, 其网络参数定期从$\omega$处获取. 并通过增设孪生网络, 生成当前估计网络$\omega_1$, ω_2和目标估计网络ω_1', ω_2'. 目标估计网络采用ω_1'和ω_2'的较小值作为最终的目标Q值, 分别更新对应的ω_1和ω_2, 如下式所示:

$$y_e = r + \gamma \min_{i=1,2} Q(s', \mu(s';\theta');\omega_i') - Q(s,a;\omega_i). \tag{8}$$

SGES算法筛选专家样本的标准是动态的, 尤其是在训练前期, 网络未经充分训练, 筛选标准并不成熟. 如果只集中学习当前情节数据轨迹中的专家样本, 则不可避免的会对一些有潜力的样本产生误判, 将其归类为非专家样本, 从而忽略了对于整体环境的探索. 同时, 即便从不完美的演示中学习, 同样可以增强算法的学习能力[26–29].

因此, SGES采用类同策略学习和异策略学习相结合的复合策略更新方法. 网络设置两个经验池, 即回合经验池D和专家经验池G. 同策略学习方法中, 同环境交互行为策略和需要被学习的目标策略是同一策略. 因此, 严格意义上同策略学习每获得一次样本, 策略都会进行一次参数更新. 启发于SIL算法中的同策略学习方式[20], 在单次情节中, 类同策略学习方法中的智能体首先与环境进行交互, 然后将所获得的样本全部存储入经验池D. 为了不过多的增加计算负担, 同时为仅保留其探索作用而削弱该过程对于网络参数影响, 仅对D中的所有样本进行一次参数学习. 即在同策略下进行采样, 并依据这一情节样本的平均梯度进行更新. 在类同策略学习结束后, 筛选专家样本并将这些样本存入经验池G, 结束后将经验池D清空. 然后, 在经验池G中进行多次随机采样, 每次都筛选出一定量的样本并进行参数学习. 这种学习是异策略的, 因为学习的样本完全由在经验池中随机采样得到, 并非由当前的行为策略与环境交互得出. SGES算法流程图与网络结构如图1所示.

尽管同策略确定性策略方法不能保证实际中的运用, 但是如果在动作输出时加入一定噪声则仍然会有用于探索[11]. 虽然梯度不一定会随着真实梯度方向变化, 但是这可视为沿着梯度方向的一次探索, 这正是SGES所要实现的目标.

3.2　专家样本

从演示中进行学习可以给算法提供专家系统的指导. 演示的作用体现在对专家演示样本的预训练中, 可以避免冷启动问题, 提高算法的训练效率. 基于最大熵的SQN (soft Q-learning network)[30] 算法在训练过程中最大化随机策略的熵和折扣累积奖励. 假设现已得到某情节的数据轨迹, 每个数据轨迹由一定数量的样本(s_t, a_t, R_t)组成. 折扣累积奖励R_t的计算方式为

$$R_t = \sum_{k=t+1}^{\infty} \gamma^{k-t}(r_k + \alpha H_k^\mu), \tag{9}$$

其中H_k^μ是策略熵, 其计算方式为

$$H_k^\mu = -\sum_a \mu(a|s_k)\log\mu(a|s_k). \tag{10}$$

α是熵正则化系数, 计算完成后将样本存入经验池以供采样. 在环境已知的情况下, 最优策略的回报值一定大于或等于其余策略的回报值, 所以任何策略的累积回报都可以视作最优Q值的下界[4], 对于最佳Q值$Q^*(s,a)$有: $Q^*(s,a) \geqslant Q(s,a;\omega)$. 基于下界SQN的损失函数为

$$L^{\rm sqn} = E_{s,a,R\sim\mu}[\frac{1}{2}\|(R - Q(s,a;\omega))_+\|^2], \tag{11}$$

其中$(\cdot)_+ = \max(\cdot, 0)$. 当累积奖励值大于$Q$值, 即$Q^*(s,a) \geqslant R > Q(s,a;\omega)$, 此时样本才会被纳入计算. SIL在基于下界SQN的基础上, 建立了一种可以集中学习优秀样本的算法, 然而这种方法无法完全适用于确定性策略, 且在模型的梯度计算层面只针对优秀样本进行学习的做法, 会造成模型梯度在更新时彻底忽视了非优秀样本的作用, 很大程度上缩小了智能体的视野. 尤其在经典控制环境中, 由于依赖在复杂环境中的长时间训练, 这样的方法因样本量较少而力不从心. 在OpenAI gym经典控制环境Pendulum-v0环境中将SGES与PPO2+(SIL与PPO2算法[16]的结合)进行对比, 结果如图2所示.

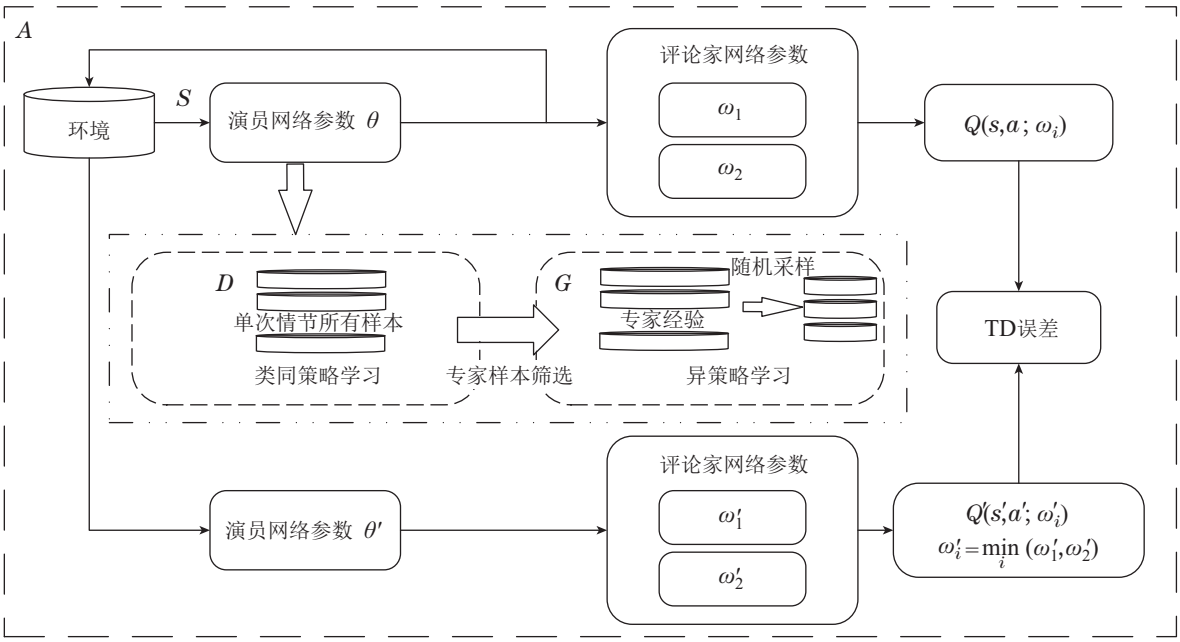


图 1 SGES算法流程图

Fig. 1 Flowchart of SGES

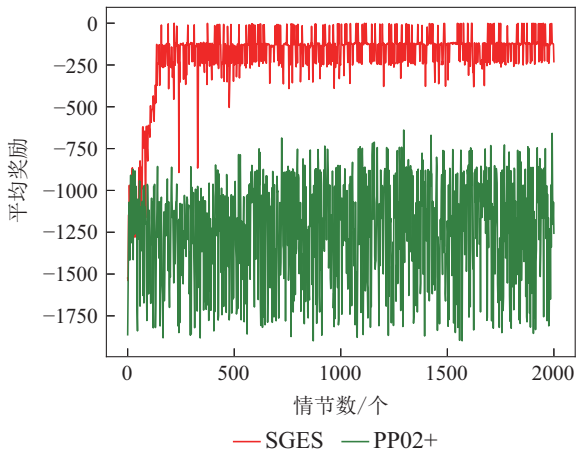


图 2 SGES与PPO2+在Pendulum-v0环境中的对比

Fig. 2 Results of SGES and PPO2+ in Pendulum-v0

SGES算法在类同策略学习过程完成后, 将对经验池D中的所有样本逐一计算奖励R_t. 从初始状态到任意时刻的累积奖励R_t代表智能体在MDP中运行到当前状态所获得的总奖励. 如果R_t大于D中所有样本的平均动作值函数$Q(s,a)$, 即$R_t \geqslant Q(s,a)$, 则认为该样本的价值高于平均水平, 所以可以在当前情节中被标记为专家样本; 反之, 则无法存入经验池G参与后续训练. 真实值函数$Q(s,a;\theta)$通过$Q(s,a;\omega)$来估计, 如果采取适当的策略, 最终将实现$Q(s,a;\omega) \approx Q(s,a;\theta)$, 筛选标准也会伴随$\theta^Q$的更新而不断改善, 动态的标准本身也增强了探索. 实际上, $Q(s,a)$即为算法中的$Q(s,a;\theta)$, 问题在于对于一个轨迹τ而言, 代表其平均奖励水平的是$V_\tau(s)$, 一个网络无法同时输出两个值函数. 参考值函数的贝尔曼方程形式为

$$Q^\mu(s,a) = E[r_t + \gamma V_\mu(s_{t+1})]. \tag{12}$$

所以可以使用$Q^\mu(s,a) - r_t$来表示$V^\mu(s')$的均值. 而且智能体希望奖励函数的增长是平稳的, 因为平滑的曲线往往会有更小的方差, 也就意味着更快

的训练速度, 所以在任意时刻s处$V^{\mu}(s)$和$V^{\mu}(s')$之间的差距都不宜过大, 所以可以使用$V^{\mu}(s')$的值来近似表示$V^{\mu}(s)$. 同时, 动作值函数$Q(s,a)$本身也代表在状态s处采取动作a所能获取奖励的平均值, 所以本文中使用$Q(s,a;\omega)$和R_t对比来选择专家样本. SGES算法伪代码如表1所示.

表 1　SGES算法伪代码

Table 1　Pseudocode of SGES

初始化θ, ω_1, ω_2, θ', ω_1', ω_2', D, G, γ, τ, ω, M, N

for each episode do:

初始化状态s和随机噪声ω

for each step do:

根据$\mu(s;\theta^{\mu})+\varepsilon$采取动作$a$, 产生$(s,a,s',r)$

end for

获得M个样本(s,a,s',r)存储到D

$l^{\text{on}}=M^{-1}\sum[r+\gamma Q(s',\mu(s;\theta);\omega_1)-Q(s,a;\omega_1)]^2$

$\omega_1\leftarrow\arg\min_{\omega_1} l^{\text{on}}$

$\nabla_{\theta}J^{\text{on}}\approx M^{-1}\sum_{t=1}^{M}[\nabla_a Q(s_t,a_t;\omega_1)\nabla_{\theta}\mu(s;\theta)]$

$\theta\leftarrow\arg\min\nabla_{\theta}J^{\text{on}}$

计算D中每个时刻的$R_t=\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}$, 将满足标准的样本存入$G$

$D\leftarrow\varnothing$, $\theta'\leftarrow\theta$, $\omega_2,\omega_1',\omega_2'\leftarrow\omega_1$

从G中随机采样N个(s,a,s',r)

$\tilde{a}=\mu(s';\theta')+\varepsilon$

$y=r+\gamma\min_{i=1,2}Q(s',\tilde{a};\omega_i')$

$l^{\text{off}}=N^{-1}\sum_{t=1}^{N}[y-Q(s_t,a_t;\omega_i)]^2$

$\omega_i\leftarrow\arg\min_{\omega_i} l^{\text{off}}$

$\nabla_{\theta}J^{\text{off}}\approx N^{-1}\sum_{t=1}^{N}[\nabla_a Q(s_t,a_t;\omega_1)\nabla_{\theta}\mu(s_t;\theta)]$

$\theta\leftarrow\arg\min\nabla_{\theta}J^{\text{off}}$

$\omega'\leftarrow\tau\omega+(1-\tau)\omega'$

$\theta'\leftarrow\tau\theta+(1-\tau)\theta'$

end for

与原始TD3算法相比, SGES在学习过程中没有改变损失函数和Q值, 因此不会改变算法的收敛性.

4　实验设置与分析

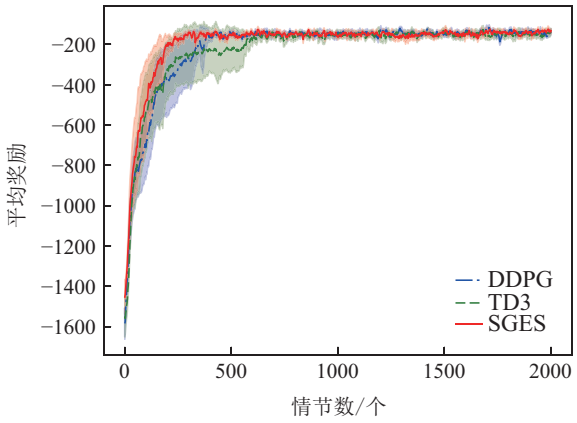
4.1　实验设置

在OpenAI Gym的一些虚拟环境中进行实验, 所有网络均采用4层全连接的神经网络结构, 具体设置参见表2. 本文一共执行了8组实验: 两组经典控制环境: Pendulum-v0 和 MountainCarContinuous-v0, 实验结果见图3; 5组MuJoCo环境: HalfCheetah-v2, Hopper-v3, Swimmer-v2, Walker2d-v2 和 Ant-v2, 以及Box2D BipedalWalker-v3 环境的实验结果见图 4. 横轴代表情节数, 纵轴代表每个情节获得的回报值. 每个实验中的每个算法都设置了10次随机数种子的重复实验, 取平均值绘制曲线, 阴影部分表示实验的标准差.

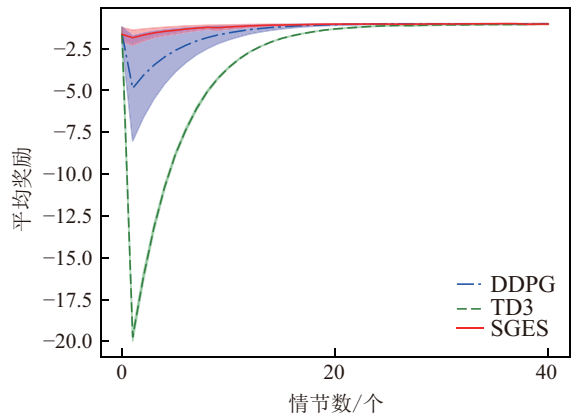
表 2　网络与超参数设置

Table 2　Settings of networks and hyper-parameters

超参数	设置
Critic-Actor学习率	0.01/0.001
Critic隐藏层神经元数	400, 300
Actor隐藏层神经元数	400, 300
Critic激活函数	ReLU, ReLU, ReLu
Actor激活函数	ReLU, ReLU, Tanh
折扣因子γ	0.99
软更新系数τ	0.005
随机噪声ω	$[-0.5, 0.5]$
每情节最大时间步M	2000
Batch-size N	100
每情节迭代次数	200
优化器	Adam[31]



(a) Pendulum-v0



(b) MountainCarContinuous-v0

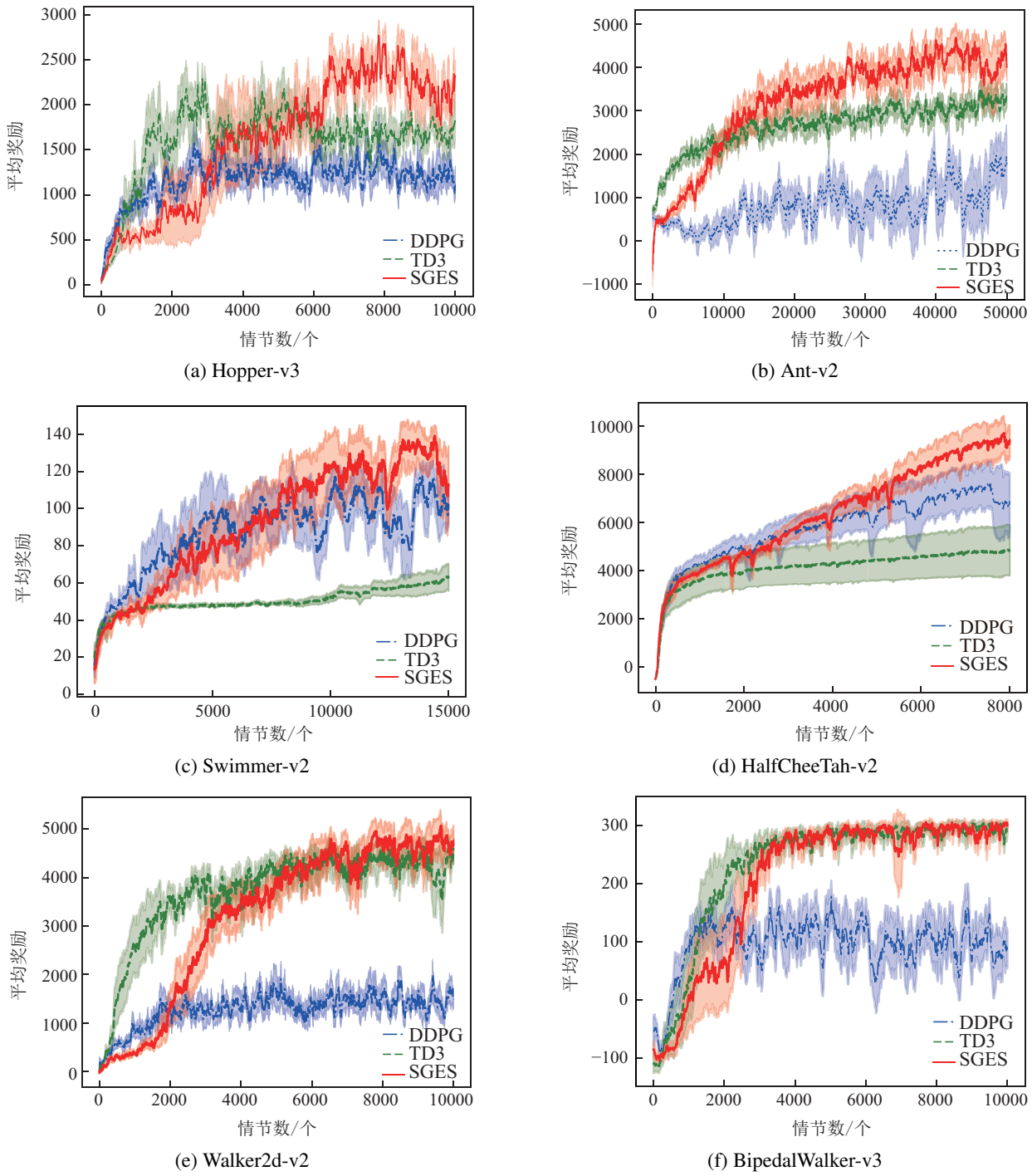
图 3　经典控制环境中的实验结果

Fig. 3　Experimental results in classical control environments

4.2　实验分析

图3中, SGES在Pendulum-v0任务中的收敛速度快于DDPG和TD3大约200个情节, 且标准差最小. 在MountainCarContinuous-v0任务中情况类似, SGES

的收敛速度和DDPG持平, 明显快于TD3. 综上, 可以认为在一些简单的、拥有密集的奖励反馈的环境中, SGES可以在保持同样奖励值的同时, 以更快的速度收敛.



(a) Hopper-v3 (b) Ant-v2

(c) Swimmer-v2 (d) HalfCheeTah-v2

(e) Walker2d-v2 (f) BipedalWalker-v3

图 4 MuJoCo和Box2D环境中的实验结果

Fig. 4 Experimental results in MuJoCo and Box2D environments

图4中, 在Hopper-v3任务上, 大约在6000个情节以后SGES的奖励值明显高于对比算法, 在8000个情节处获得最高奖励, 此后有下降的趋势; 在Ant-v2环境中, SGES大约在10000个情节后取得对TD3的优势, 且一直明显优于DDPG, 虽然SGES的回报曲线在后期产生震荡, 但仍有上升的趋势; 在Swimmer-v2中, TD3收敛在一个很低的水平, SGES同DDPG之间交替领跑, 但在后期对DDPG拉开差距; 在HalfCheeTah-v2环境中, SGES在大约3000个情节后开始显现出对DDPG的优势, 并一直保持明显的上升势头; Walker2d-v2 环境中, SGES 的前6000个, SGES 和TD3基本保持同一水平, 明显优于DDPG. 考虑到SGES提出的目的在于提升算法的探索能力, 而更加复杂的结构或许会增加算法的方差, 方差的增加确

实会牺牲算法的收敛速度, 但这一延缓在大多数环境中并不明显, 因此可认为SGES算法获得了更大的回报值, 同时并未明显降低收敛速度, 这可以证明该算法在探索中的有效性.

从实验中可以看出, TD3算法无疑拥有最好的稳定性, 但是SGES的标准差在实验中也保持着随着训练进行而缩小的趋势, 并在算法获得最高奖励值时达到最小. 这说明随着训练的推进, SGES筛选专家样本的标准变得更加合理, 经验池的样本状态分布集中在情节的后段, 而这些样本虽然具有更高的价值, 且通常很难被访问到, 这也明确显示了SGES机制起到了很大的作用.

在内存的占用方面, 首先, SGES相比于TD3并未增加网络结构; 同时, 该算法多设置的经验池D每次情节至多存储2000样本, 并在每情节结束后清空经验池; 类同策略学习的梯度更新其每次情节只进行一次. 在相同计算机中, 3种算法在各实验环境中运行时所占用的平均内存显示, SGES所占用的内存约分别为TD3和DDPG所占用内存的1.05和1.1倍. 因此, SGES算法并不会明显占用更多的内存.

5　结论

针对深度强化学习在连续动作环境中的探索问题, 本文提出了一种基于自生成专家样本(SGES)的深度确定性策略算法, 算法将模仿学习中行为克隆的思想融合进Actor-Critic框架之中, 视模型本身为一个专家演示系统来生成演示样本, 提供具有指导性的探索. 将类同策略学习与异策略学习相结合, 为算法提供一个具有启发式探索的参数更新过程. 最后, 分析了算法的收敛性. SGES在上述8个模拟环境中进行了充分的实验, 并选择DDPG和TD3作为对比, 实验结果表明, 在所有模拟环境中SGES都获得了最高的奖励值(或最高之一), 并且在大部分环境中的奖励值仍保持了上升趋势; 同时, SGES的收敛速度在复杂环境没有明显延迟, SGES的方差也保持在一个稳定的范围内, 并随着训练的进行同步缩小. 未来, 希望针对SGES的方差进行进一步研究, 致力于缩小方差以期提升算法的收敛速度.

参考文献:

[1] TANG Zhentao, SHAO Kun, ZHAO Dongbin, et al. Recent progress of deep reinforcement learning: From AlphaGo to AlphaGo Zero. *Control Theory & Applications*, 2017, 34(12): 1529 – 1546.
(唐振韬, 邵坤, 赵冬斌, 等. 深度强化学习进展: 从AlphaGo到AlphaGo Zero. 控制理论与应用, 2017, 34(12): 1529 – 1546.)

[2] WILLIAMS R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 1992, 8(34): 229 – 256.

[3] BERTSEKAS D, TSITSIKLIS J. *Neuro-dynamic Programming*. Belmont: The MIT Press, 1996.

[4] ZOU Qijie, LIU Shihui, ZHANG Yue, et al. Rapidly-exploring random tree algorithm for path re-planning based on reinforcement learning under the peculiar environment. *Control Theory & Applications*, 2020, 37(8): 1737 – 1748.
(邹启杰, 刘世慧, 张跃, 等. 基于强化学习的快速探索随机树特殊环境中路径重规划算法. 控制理论与应用, 2020, 37(8): 1737 – 1748.)

[5] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing Atari with deep reinforcement learning. *ArXiv Preprint*, 2013: ArXiv: 1312.5602.

[6] SMART, W D, KAELBLING, L P. Effective reinforcement learning for mobile robots. *Proceedings of the IEEE International Conference on Robotics and Automation*. Washington, USA: IEEE, 2002: 3404 – 3410.

[7] SCHULMAN J, LEVINE S, MORITZ P, et al. Trust region policy optimization. *Proceedings of the International Conference on Machine Learning*. Lille, France: ICML, 2015: 1889 – 1897.

[8] SUTTON R, MCALLESTER D, SINGH S, et al. Policy gradient methods for reinforcement learning with function approximation. *Proceedings of the International Conference on Neural Information Processing Systems*. Cambridge, USA: NIPS, 1999: 1057 – 1063.

[9] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms. *ArXiv Preprint*, 2017: ArXiv: 1707.06347.

[10] DEGRIS T, WHITE M, SUTTON R S. Off-policy actor-critic. *Proceedings of the International Conference on Machine Learning*. Edinburgh, UK: ICML, 2012: 179 – 186.

[11] SILVER D, LEVER G, HEESS N, et al. Deterministic policy gradient algorithms. *Proceedings of the International Conference on Machine Learning*. Beijing, China: ICML, 2014: 387 – 395.

[12] GRANDVALET Y, BENGIO Y, CHAPELLE O, et al. *Entropy Regularization*. Belmont: The MIT Press, 2006.

[13] HAARNOJA T, TANG H, ABBEEL P, et al. Reinforcement learning with deep energy-based policies. *Proceedings of the International Conference on Machine Learning*. Sydney, Australia: ICML, 2017: 1352 – 1361.

[14] GRANDVALET Y, BENGIO Y. Semi-supervised learning by entropy minimization. *Proceedings of the International Conference on Neural Information Processing Systems*. Vancouver, Canada: NIPS, 2005: 281 – 296.

[15] BELLEMARE M, SRINIVASAN S, OSTROVSKI G, et al. Unifying count-based exploration and intrinsic motivation. *Proceedings of Advances in Neural Information Processing Systems*. Barcelona, Spain: NIPS, 2016: 1471 – 1479.

[16] PATHAK D, AGRAWAL P, EFROS A A, et al. Curiosity-driven exploration by self-supervised prediction. *Proceedings of the International Conference on Machine Learning*. Sydney, Australia: ICML, 2017: 488 – 489.

[17] JAVIER R. Probability matching and reinforcement learning. *Journal of Mathematical Economics*, 2013, 49(1): 17 – 21.

[18] RUSSO D J, VAN R B, KAZEROUNI A, et al. A tutorial on Thompson sampling. *Foundations and Trends in Machine Learning*, 2018, 11(1): 1 – 96.

[19] ZHU Yuanheng, ZHAO Dongbin. Probably approximately correct reinforcement learning solving continuous-state control problem. *Control Theory & Applications*, 2016, 33(12): 1603 – 1613.
(朱圆恒, 赵冬斌. 概率近似正确的强化学习算法解决连续状态空间控制问题. 控制理论与应用, 2016, 33(12): 1603 – 1613.)

[20] OH J, GUO Y, SINGH S, et al. Self-imitation learning. *Proceedings of the International Conference on Machine Learning*. Stockholm, Sweden: ICML, 2018: 3878 – 3887.

[21] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning. *ArXiv Preprint*, 2015: ArXiv: 1509.02971.

[22] FUJIMOTO S, HOOF H, MEGER D. Addressing function approximation error in Actor-Critic methods. *Proceedings of the 35th International Conference on Machine Learning*. Stockholm, Sweden: ICML, 2018, 80: 1587 – 1596.

[23] BHATNAGAR S, GHAVAMZADEH M, LEE M, et al. Incremental natural actor-critic algorithms. *Proceedings of Advances in Neural Information Processing Systems*. Vancouver, Canada: NIPS, 2007, 20: 105 – 112.

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

[25] HASSELT H. Double Q-learning. *Proceedings of the International Conference on Neural Information Processing Systems*. Whistler, Canada: NIPS, 2010, 23: 2613 – 2621.

[26] HESTER T, VECERIK M, PIETQUIN O, et al. Deep Q-learning from demonstrations. *Proceedings of the AAAI Conference on Artificial Intelligence*. New Orleans, USA: AAAI, 2018: 2145 – 2151.

[27] NAIR A, MCGREW B, ANDRYCHOWICZ M, et al. Overcoming exploration in reinforcement learning with demonstrations. *Proceedings of the International Conference on Robotics and Automation*. Brisbane, Australia: IEEE, 2018: 6292 – 6299.

[28] ANDRYCHOWICZ M, WOLSKI F, RAY A, et al. Hindsight experience replay. *ArXiv Preprint*, 2017: ArXiv: 1707.01495.

[29] GAO Y, XU H, LIN J, et al. Reinforcement learning from imperfect demonstrations. *ArXiv Preprint*, 2018: ArXiv: 1802.05313.

[30] SCHULMAN J, CHEN X, ABBEEL P. Equivalence between policy gradients and soft Q-learning. *ArXiv Preprint*, 2018: ArXiv: 1704.06440.

[31] KINGMA D P, BA J. Adam: A method for stochastic optimization. *ArXiv Preprint*, 2014: ArXiv: 1412.6980.

作者简介:

刘 健 博士, 副教授, 目前研究方向为机器学习与智能信息处理等, E-mail: liujiansqjxt@126.com;

赵恒一 硕士研究生, 目前研究方向为深度强化学习, E-mail: zhaohysino@qq.com.