控制与决策
Control and Decision

MADDPG算法经验优先抽取机制

何明， 张斌， 柳强， 陈希亮， 杨钺

引用本文:
何明, 张斌, 柳强, 等. MADDPG算法经验优先抽取机制[J]. 控制与决策, 2021, 36(1): 68−74.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2019.0834

MADDPG算法经验优先抽取机制

何 明[1], 张 斌[1†], 柳 强[2], 陈希亮[1], 杨 钺[1]

(1. 中国人民解放军陆军工程大学 指挥控制工程学院，南京 210007；2. 海军指挥学院，南京 210000)

摘 要: 针对多智能体深度确定性策略梯度算法(MADDPG)学习训练效率低、收敛速度慢的问题,研究MADDPG算法经验优先抽取机制,提出PES-MADDPG算法. 首先,分析MADDPG算法的模型和训练方法;然后,改进多智能体经验缓存池,以策略评估函数误差和经验抽取训练频率为依据,设计优先级评估函数,以优先级作为抽取概率获取学习样本训练神经网络;最后,在合作导航和竞争对抗2类环境中进行6组对比实验,实验结果表明,经验优先抽取机制可提高MADDPG算法的训练速度,学习后的智能体具有更好的表现,同时对深度确定性策略梯度算法(DDPG)控制的多智能体训练具有一定的适用性.

关键词: 多智能体；深度强化学习；MADDPG；经验优先抽取

中图分类号: TP273 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2019.0834 **开放科学(资源服务)标识码(OSID):**

引用格式: 何明,张斌,柳强,等. MADDPG算法经验优先抽取机制[J]. 控制与决策, 2021, 36(1): 68-74.

Multi-agent deep deterministic policy gradient algorithm via prioritized experience selected method

HE Ming[1], *ZHANG Bin*[1†], *LIU Qiang*[2], *CHEN Xi-liang*[1], *YANG Cheng*[1]

(1. College of Command and Control Engineering，The Army Engineering University of PLA，Nanjing 210007，China；
2. Naval Command College，Nanjing 210000，China)

Abstract: In order to mitigate the problem of low efficiency and slow convergence of the multi-agent deep deterministic policy gradient (MADDPG) algorithm, the prioritized experience selection mechanism of MADDPG algorithm is studied and PES-MADDPG algorithm is proposed. Firstly, the model and the training method of the MADDPG algorithm are analyzed, the multi-agent experience buffer pool is ameliorated, and the priority evaluation function is designed based on the error of critic function and the training frequency of experience. The priority is treated as the selection probability to obtain the learning sample for training neural network. Finally, six groups of comparative experiments are conducted in both cooperative navigation and competitive environment. The experiments results show that the prioritized experience selection mechanism improves the training speed of the MADDPG algorithm, and the trained agents have better performance. The prioritized experience selection mechanism also has certain applicability to the training of multi-agents controlled by the deep detcrministic policy gradient (DDPG) algorithm.

Keywords: multi-agent；deep reinforcement learning；MADDPG；prioritized experience selected method

0 引 言

近年来,深度强化学习(DRL)算法的研究取得了较大的进展[1-5],并应用在自动驾驶策略学习[6-7]、机器人控制[8]和游戏竞赛[9-11]等领域. 相对于单智能体,多智能体能够互相通信、协作,共同完成目标任务,具有更广阔的应用场景. 但是,多智能体强化学习(MADRL)也面临很多问题和挑战[12]: 1)环境变化的不确定性,每个智能体不仅要考虑环境的变化,还要考虑其他智能体所采取策略对自身策略的影响; 2)智能体之间的协同通信对学习策略具有较大影响; 3)智能体数量多,状态空间大,训练速度慢甚至难以收敛. 当前的研究主要集中在智能体学习行为涌现、智能体通信、相互协作和智能体行为建模4个方面[13],并取得了一定的突破. 多智能体协同控制的强化学习算法主要有Lenient-DQN[14]、Hysteretic-DRQN[15]、WDDQN[16]、FTW[17]、VDN[18]、QMIX[19]、

收稿日期: 2019-06-11；**修回日期:** 2019-08-12.

基金项目: 国家重点研发计划项目(2018YFC0806900, 2016YFC0800606, 2016YFC0800310)；江苏省自然科学基金项目(BK20161469)；江苏省重点研发计划项目(BE2016904, BE2017616, BE2018754)；中国博士后基金项目(2018M633757).

†通讯作者. E-mail: qdjmzb@qq.com.

COMA[20]、MADDPG[21] 等. 相对于其他算法, 多智能体深度确定性策略梯度算法 (MADDPG) 既能应用于多智能体协同合作、又能应用于竞争对抗场景, 同时, 在集中训练时可以利用其他智能体的观察信息和策略加速训练过程, 并应用策略推断和策略集合机制增强算法的鲁棒性, 具有更广阔的应用场景.

智能体与环境交互产生的经验进入缓存池, 而后抽取经验进行训练. 由于数据的不均衡性, 随机抽取经验的质量参差不齐, 导致学习效率低、算法收敛速度慢. 在单智能体深度强化学习上, 针对经验的优先选择机制主要集中在两个方面: 1) 决定哪些高质量的经验进入缓存池; 2) 如何从缓存池中抽取高质量的经验. 前者又称为优选进入, 后者又称为优先选择. 优先选择机制的代表算法是文献 [22] 提出的 PER (prioritized experience replay), 该方法结合 DQN(deep Q_network) 算法的特点, 使用 TD(temporal-difference) 误差衡量经验的重要性, 优先选择误差较大的经验进行训练, 在 Atari 游戏中获得了较好的效果. 优先进入机制的代表算法是文献 [23] 提出的重抽样优选缓存经验回放机制方法, 该方法根据经验的 TD 误差决定是否进入经验缓存池, 优先级较小的经验以一定的概率直接舍弃, 从而避免了多次计算 TD 误差权衡经验重要性带来的算法时间复杂度. 后续研究在上述方法的基础上, 作了进一步改进, 文献 [24] 提出了 PSER(prioritized sequence experience replay) 方法, 该方法以 DQN 算法为基础, 利用前后经验状态转化的相关性, 对稀疏奖励的强化学习方法具有较好的适用性. 文献 [25] 将 PER 方法应用于 DDPG(deep deterministic policy gradient) 算法, 提高了采样效率.

上述方法仅在单智能体强化学习算法上进行了改进和实验, 在多智能体强化学习中, 每个智能体拥有单独的策略网络、评估网络和经验缓存池, 难以找到统一的指标衡量每次交互的经验. PER[22] 机制通过二叉树对 TD 误差排序, 将算法复杂度排序降低到 $O\log(n)$, PSER 方法在单智能体的 DQN 算法上, 具有较好的效果, 而多智能体环境中每个智能体拥有独立经验缓存池, 各自排序打乱了经验集中式训练的关联性, 无法完成训练. 重抽样优选缓存经验回放机制的方法[23] 通过经验的 TD 误差决定是否进入经验缓存池, 不同智能体对相同的经验有不同 TD 误差, 难以决定哪个经验进入缓存池, 所以难以应用于多智能体连续动作控制. 本文借鉴了上述 2 种方法, 以 MADDPG 算法为基础, 首先分析算法的模型训练框架, 而后提出了经验优先抽取的多智能体深度确定性策略梯度算法 (prioritized experience selected multi-agent deep deterministic policy gradient, PES-MADDPG). 优先经验抽取机制有以下特点:

1) 每个智能体拥有单独的经验缓存池, 记忆自身的观察信息, 各自计算经验的优先级.

2) 所有智能体缓存池中经验不排序, 且保持初始的进入顺序, 所以优先抽取的算法复杂度相对于缓存池大小为 $O(1)$.

3) 仅在进入缓存池时计算经验的策略损失误差, 降低算法的时间复杂度.

1　多智能体深度确定策略算法模型及训练方法

1.1　基本假设

设环境 E 中有 N 个智能体, 其策略的集合为 $\pi=\{\pi_1,\pi_2,\ldots,\pi_N\}$, 各个策略由神经网络表示, 其参数的集合 $\theta=\{\theta_1,\theta_2,\ldots,\theta_N\}$, 环境系统满足以下假设.

假设 1　每个智能体的策略仅取决于其自身观察到的状态, 而与其他智能体观察到的状态无关, 即 $a_i=\pi_i(o_i)$.

假设 2　环境是未知和无模型的, 每个智能体的奖励以及采取动作后的下一状态是不可预料的, 奖励来源于环境的反馈, 自身动作仅取决于策略.

假设 3　在训练时, 各智能体之间的通信不作设定, 即相互之间不通信, 或通信的内容作为其观察值的一个分量.

1.2　模型训练框架

MADDPG 算法训练框架如图 1 所示, 环境中所有的智能体均由 actor 网络、critic 网络、target actor 网络和 target critic 网络组成, 为便于画图展示, 图中以智能体 i 为例进行展开, 其他智能体以方框代表.

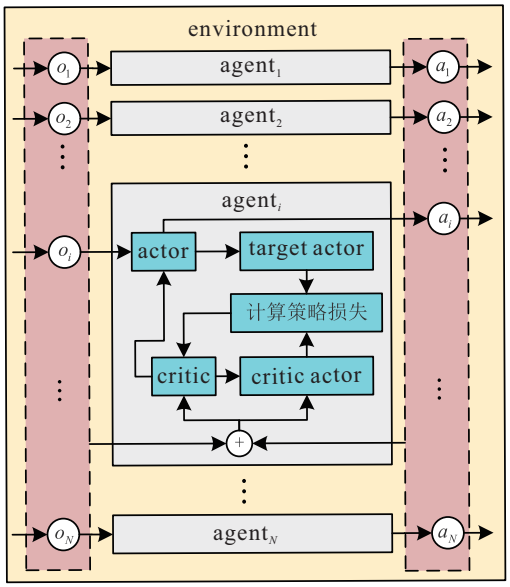


图 1　MADDPG 算法训练框架图

训练过程采取分散执行、集中训练的方式,即每个智能体根据自身策略得到当前状态执行的动作$a_i^j = \pi_i^j(o_i^j)$,并与环境交互得到的经验$(o_i^j, a_i^j, o_i^{j+1}, r_i^j)$一起存入自身的经验缓存池. 待所有智能体与环境交互后,每个智能体从经验池中随机抽取经验,训练各自的神经网络. 为加速智能体的学习过程,critic网络的输入要包括其他智能体的观察状态和采取的动作,即$Q = Q(s_j, a_1, a_2, \dots, a_N, \theta^Q)$,其中$s_j = (o_1^j, o_2^j, \dots, o_N^j)$. 通过最小化损失以更新critic网络参数,策略损失的计算公式为

$$L = \frac{1}{K}\sum_{j=1}^{K}(y_j - Q(s_j, a_1, a_2, \dots, a_N, \theta^Q))^2. \quad (1)$$

通过梯度下降法计算更新动作网络的参数,梯度计算公式为

$$\nabla_{\theta^\pi} J = \frac{1}{K}\sum_{j=1}^{K}\nabla_{\theta^\pi}\pi(o, \theta^\pi)\nabla_a Q(s, a_1, a_2, \dots, a_N, \theta^Q). \quad (2)$$

1.3 算法适用性拓展

在部分训练场景中,智能体难以得知其他智能体的策略来加速训练过程. 为解决这个问题,可以采用策略估计的方法,智能体i对智能体j的策略估计用$\hat{u}_i^j$来表示,$\hat{u}_i^j$为神经网络,通过最大化智能体j的动作概率对数和正则化熵来更新网络参数,这样便不用输入其他智能体的策略,而采用自身估计的策略进行学习,具体公式如下所示:

$$\boldsymbol{L} = -E_{o_j, a_j}[\log \hat{u}_i^j(a_j|o_j) + \lambda H(\hat{u}_i^j)]. \quad (3)$$

2 经验优先抽取的多智能体深度确定策略梯度算法

在MADDPG算法中,每个智能体各自抽取自身的经验,训练自身的神经网络. 为了便于抽取质量更高的经验,在经验缓存池存取的经验内容,除当前状态、采取动作、下一状态、奖励回报外,还保存目标评估网络损失Loss、经验抽取训练次数T和当前经验的优先级Pr等数据. 优先级Pr是衡量经验重要性的唯一指标,是抽取的衡量依据. 其中

$$\text{Loss} = (y - Q^\pi(s, a_1, a_2, \dots, a_N))^2, \quad (4)$$

$$y = r + \gamma Q^{\pi'}(s', a_1', a_2', \dots, a_N')|_{a_i' = \pi_i'(o_i)}. \quad (5)$$

Loss越大,说明对此次经验而言,目标网络的评估值和实际值差别越大,需要提高采样频率,以便尽快更新目标网络和评估网络的值,达到最优的训练效果. 最简单的方式如下:

$$\text{Pr}(i) = \frac{\text{Loss}(i)}{\sum_{i=1}^{M}\text{Loss}(i)}. \quad (6)$$

不同经验的目标网络损失Loss数值差异较大,单纯以数值计算会导致部分经验的Pr值较小而无法抽取进行训练,而以rank(Loss(i))作为无量纲的排序量,可以较好地衡量经验的重要性,其中rank(Loss(i))为Loss在递增排序中的位置.

经验取样次数T为经验被抽取后进行训练的次数. 仅仅以Loss作为评价经验的重要性,并选取Loss大的经验进行训练而舍弃较小的经验,虽然会加速训练的过程,但是由于经验抽取的不均衡,部分经验由于Loss较小而被抽取训练的次数较少,甚至始终无法训练,导致神经网络的过拟合或者陷入局部最优. 因此,在衡量经验的优先级时,要综合考量Loss和被抽取训练的次数. 在之前训练中,被抽取的次数越多,其概率优先级Pr越小,被抽取的次数越小,其概率优先级Pr越大. Pr的计算公式为

$$\text{Pr}(i) = \frac{p^\alpha(i)}{\sum_{j=1}^{M}p^\alpha(j)} + \beta. \quad (7)$$

其中

$$p(i) = \text{rank}(\text{rank}(\text{Loss}(i)) + \text{rank}_{\text{reverse}}(T)), \quad (8)$$

$\text{rank}_{\text{reverse}}(T)$为抽取次数$T$在递减排序中的位置,$T$越小,递减排序次数越大;$\alpha$为优先级的放大次数,$\alpha$越大,表明越依赖$p(i)$的大小抽取经验;$\beta \in (0,1)$为概率的偏移量,防止由于$p(i)$过小而抽中经验的概率较低.

Loss计算的时机:智能体与环境交互后,将所得单次经验存入经验缓存池,此时经验的Loss、Pr为空,而取样次数为0. 每探索K步,批量计算这K步经验的损失并写入经验,保证在抽样训练之前,所有的经验都已计算出损失Loss.

经验优先抽取算法的流程如图2所示. 从经验缓存池中抽取M个经验,计算每个经验的优先级Pr,并以概率Pr将经验放入$N_{\text{minibatch}}$中,重复抽取M个经验,直到$N_{\text{minibatch}}$到达指定大小.

在训练后,actor网络、critic网络以及对应的目标网络,其网络参数都已发生变化,考虑到算法复杂度和计算量问题,只在训练中更新抽中的$N_{\text{minibatch}}$经验中的Loss,经验缓存池中其他经验的Loss不再重新计算.

经验优先抽取的多智能体深度确定性策略梯度

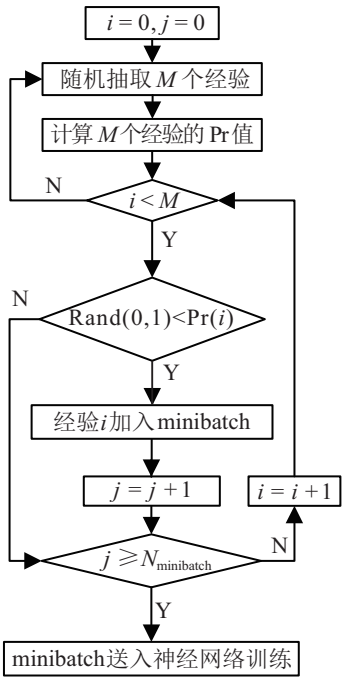


图 2　经验优先抽取机制算法流程

算法(PES-MADDPG)伪代码如下所示.

step 1: 初始化每个智能体的策略网络 $\pi_i(o,\theta^{\pi_i})$ 和评估网络 $Q_i(s,a_1,a_2,\ldots,a_N,\theta^{Q_i})$, 以及网络参数 θ^{π_i} 和 θ^{Q_i}.

step 2: 初始化每个智能体的目标策略网络 $\pi_i'(o,\theta^{\pi_i'})$ 和目标评估网络 $Q'_i(s,a_1,a_2,\ldots,a_N,\theta^{Q_i'})$ 以及网络参数 $\theta^{\pi_i'}$ 和 $\theta^{Q_i'}$.

step 3: 初始化每个智能体的经验缓存池 R_i 和动作探索噪声 $\aleph_a$.

step 4: 对于每个回合(episode), 循环以下步骤.

step 4.1: 初始化环境和所有智能体状态集合 s_1.

step 4.2: 在回合中的每一步(step), 对环境中的每个智能体 i, 执行以下步骤:

1) 根据当前智能体的观察状态、策略网络和噪声探索策略选择动作 $a_i^j=\pi_i(o_i^j,\theta^{\pi_i})+\aleph_t$;

2) 智能体 i 执行当前动作 a_i^j 得到下一状态 o_i^{j+1} 和奖励 r_i^j, 并将经验 $(o_i^j,a_i^j,o_i^{j+1},r_i^j)$ 存入经验缓存池.

step 4.3: 每执行 M 步, 对每个智能体, 按照以下步骤训练神经网络:

1) 根据式(4)和(5)计算当前智能体近 M 步经验的策略损失;

2) 根据式(7)计算当前经验的优先级;

3) 根据图 2 算法, 抽取 minibatch 的经验;

4) 通过目标评估网络计算每个经验动作期望回报 $y_j=r_j+\gamma Q'(s_{j+1},a'_1,a'_2,\ldots,a'_N,\theta^{Q'})$;

5) 最小化损失以更新评估网络参数

$$L=\frac{1}{K}\sum_{j=1}^{K}(y_j-Q(s_j,a_1,a_2,\ldots,a_N,\theta^Q))^2;$$

6) 通过以下梯度更新当前智能体的策略网络参数:

$$\nabla_{\theta^\pi}J=$$

$$\frac{1}{K}\sum_{j=1}^{K}\nabla_{\theta^\pi}\pi(o,\theta^\pi)\nabla_a Q(s,a_1,a_2,\ldots,a_N,\theta^Q).$$

step 4.4: 每执行一定的步数, 根据下式更新所有智能体的目标策略网络和目标评估网络参数

$$\theta^{Q'}=\tau\theta^Q+(1-\tau)\theta^{Q'},$$

$$\theta^{\pi'}=\tau\theta^\pi+(1-\tau)\theta^{\pi'}.$$

step 4.5: 结束步(step)循环.

step 5: 结束回合(episode)循环.

3　实验数据与分析

3.1　实验环境

软件环境为 ubantu16.04+TensorFlow+gym, 硬件为英特尔至强 E52628v3*2+GeForce GTX 1080TI*3+64 G 内存, 测试环境为 DeepMind multi-agent actor-critic for mixed cooperative-competitive environments, actor、critic 网络和对应的目标神经网络都采用 2 层隐藏层的全连接神经网络, 隐藏单元数为 64.

与深度学习等其他机器学习方法不同, 强化学习没有数据集. 衡量强化学习算法的优劣主要有两个方面: 一是奖励随训练轮数变化的曲线, 在相同训练轮数下, 奖励越大, 训练效率越高, 奖励曲线越快趋于平稳, 收敛速度越快; 二是训练一定轮数后, 智能体在环境下的表现, 智能体表现越好, 算法训练效率越高. 在以下实验中, 分别对比各算法的奖励曲线以及智能体的实际表现.

3.2　合作导航实验

在坐标 $[-1,1]$ 二维平面中有 N 个智能体和 N 个目标点, 智能体的学习目标是每个智能体以最短的时间(步数)到达一个目标点, 而避免与其他智能体相撞. 为尽快达到总体目标, 各智能体不仅考虑与最近目标的距离, 同时要考虑其他智能体与目标的相对位置, 以免陷入局部最优策略, 延长整体的导航时间.

每个智能体的奖励取决于距离最近的目标距离以及是否与其他智能体发生碰撞, 距离越近, 奖励越大, 且对碰撞的智能体施加惩罚. 碰撞奖励定义为

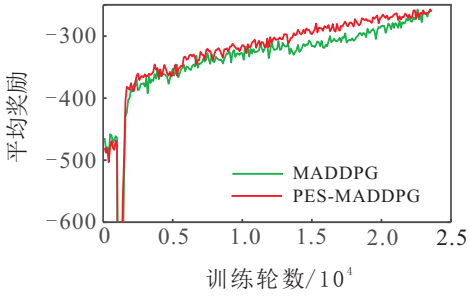
$$C=\begin{cases}-1, & \text{如果碰撞;}\\ 0, & \text{如果未碰撞.}\end{cases}\tag{9}$$

设智能体 i 与目标 j 的距离为 $D(i,j)$, 则智能体 i

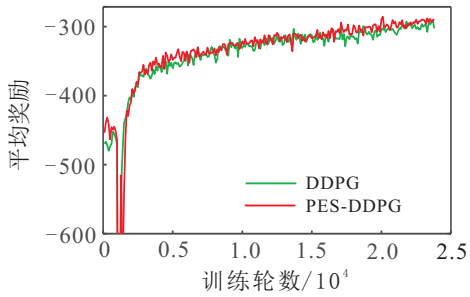
的奖励为

$$r_i = -\min_{j\leqslant N}(D(i,j)) + C. \tag{10}$$

在$N = 3$的环境中,多智能体学习策略分别采取MADDPG算法和DDPG算法,经过25 000轮的训练后,其平均奖励曲线如图3所示. 由图3可知,MADDPG算法在采用经验优先抽取机制时,奖励得到较大的提升,DDPG算法在采用经验优先抽取机制时奖励有较小的提升.



(a) 智能体采取MADDPG算法



(b) 智能体采取DDPG算法

图3 合作导航实验奖励曲线图

表1为经过24 000轮训练后,智能体学习到的策略经过随机1 000轮、50 000步的实验,得到每一步的平均碰撞次数、智能体与最近目标的平均最小距离以及智能体平均占领目标的次数. 对于MADDPG算法控制的多智能体,3个指标分别提高了1.8%、11.8%和33.9%,对于DDPG算法控制的多智能体,提高幅度较小,分别为无明显提高、5.6%和36.2%.

表1 平均每步碰撞次数、最近距离和占领次数

实验算法	经验优先抽取机制	平均碰撞次数	平均最近距离	平均占领目标次数
MADDPG	否	0.530	0.649	0.991
	是	0.521	0.573	1.327
DDPG	否	0.524	0.898	0.368
	是	0.524	0.847	0.501

鉴于实验中,平均碰撞次数、智能体与最近目标的平均最小距离两个指标提升幅度较小,对1 000轮随机实验的上述两个指标进行假设检验. 假设H_0:采用经验优先抽取机制对上述两个指标提升不明显. 表2为两个指标的均值、方差、对应的t双尾临界和T统计量. 由表2可知,T统计量远大于t双尾临界,因此拒绝H_0,采取经验优先抽取机制对于平均碰撞次数、智能体与最近目标的平均最小距离两个指标有明显的提高.

表2 MADDPG算法平均碰撞次数、最近距离假设检验

检验指标	经验优先抽取机制	均值	方差	t双尾临界	T统计量
平均碰撞次数	否	0.530	0.003 1	1.96	4.48
	是	0.521	0.001 6		
平均最近距离	否	0.649	0.044 1	1.96	8.79
	是	0.573	0.033 7		

3.3 竞争对抗实验

在相互合作的学习环境中,随着训练时间的延长,策略会更加成熟完善,多智能体的奖励也会更高. 而在竞争对抗环境中,多智能体的奖励不仅取决于自身策略,同时取决于对手学习到的对抗策略,两者的策略学习速度未必同步,导致其奖励未必会持续升高,甚至出现波动和震荡,采用经验优先抽取机制算法会相对提高奖励.

环境为经典的捕食者与被捕食环境[21],在坐标[0,1]的二维平面中有N个速度相对较慢的捕食者,相互合作共同追捕1个速度较快的被捕食者. 环境中有L个相对较大的障碍物,可以阻挡智能体的运动和观察. 捕食者的奖励取决于其和被捕食者之间的距离以及是否发生碰撞,距离越小,其奖励越大. 设$D(i,j)$为捕食者i与被捕食者j之间的距离,有

$$D(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{11}$$

当捕食者碰撞被捕食者(即捕获成功)时,捕食者会收获较大的奖励,同时,被捕食者收获较大的惩罚(负值奖励). 碰撞奖励

$$C = \begin{cases} 10, \text{ 如果碰撞;} \\ 0, \text{如果未碰撞.} \end{cases} \tag{12}$$

为维持环境的正常运行,防止因智能体逃出边界而难以学习到较好的策略,对逃出边界的智能体施加较大的惩罚. 惩罚大小取决于远离边界的程度,即边界奖励

$$C = \begin{cases} 0, \ \max(x_i, y_i) < 0.9; \\ 200(\max(x_i, y_i) - 0.9), \ \text{otherwise.} \end{cases} \tag{13}$$

各捕食者相互协作,共同完成捕食任务,良好的捕食策略并不是每个智能体贪心策略的集合,部分智能体为完成整体任务而牺牲个人的瞬时较高奖励,所以在定义奖励时采取的距离为众多智能体的最小距离,则捕食者i的奖励为

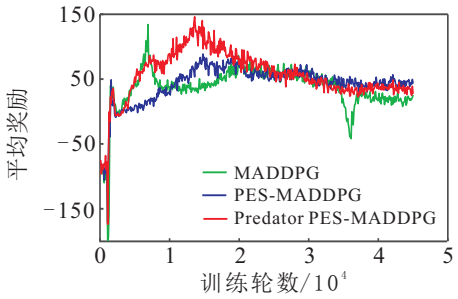
$$r_i = -0.1\min_{i\leqslant N}(D(i,j)) + C + B. \tag{14}$$

被捕食者为今后的逃脱预留空间,且运动的速度较快,所以不能考虑当前距捕食者的最小距离,而是要考虑与环境中众多被捕食者的距离之和,被捕食者 j 的奖励为

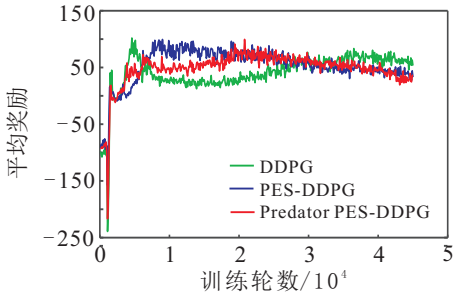
$$r_j = 0.1\sum_{i=1}^{N}(D(i,j)) - C + B. \tag{15}$$

对 3 个智能体的环境进行 45 000 步的训练,得到捕食者的平均奖励与训练轮数之间的关系. 为验证优先经验抽取算法的适用性,捕食者分别采取 MADDPG 算法和 DDPG 算法学习控制策略.

图 4(a) 为捕食者采取 MADDPG 算法的奖励曲线,对比智能体随机抽取经验、所有智能体均采用经验优先抽取和仅捕食者采用经验优先抽取算法 3 种策略. 随着训练的进行,3 种策略在达到奖励峰值后均有下降的趋势,并在 35 000 轮以后趋于稳定. 采用经验优先抽取算法的策略奖励值总体高于原始算法,且仅捕食者采取经验优先抽取时,奖励值明显高于其他两种策略.



(a)　捕食者采取 MADDPG 算法

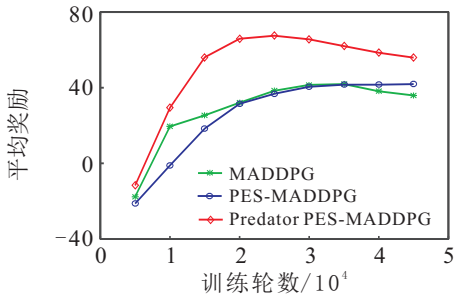


(b)　捕食者采取 DDPG 算法

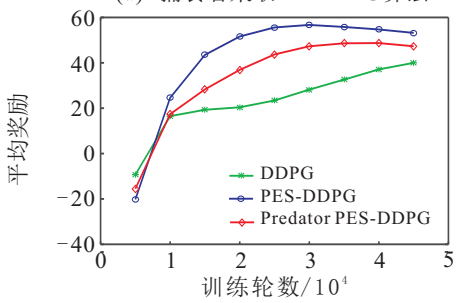
图 4　捕食者奖励曲线

图 4(b) 为捕食者采取 DDPG 算法的奖励曲线,对比智能体随机抽取经验、所有智能体均采用经验优先抽取和仅捕食者采用经验优先抽取算法 3 种策略. 采用 DDPG 算法的多捕食者系统内,所有捕食者采用一个 critic 网络来评估在当前状态下采取某个动作的优劣值. 3 种策略在训练后,均能在较高的平均奖励值处达到收敛,且采用经验优先抽取算法的策略在前 30 000 轮训练时,平均策略均较高,能保持较好的稳定性. 尽管随着训练的进行,模型出现了退化现象,但是并未采取经验优先抽取的原始算法在 40 000 轮以后也出现了退化,但采用经验优先抽取算法的策略总体上奖励值较高.

图 5 为捕食者采取 MADDPG 算法和 DDPG 算法学习控制策略时,3 种策略的前 N 轮平均奖励曲线图. 由图 5 可知,采用经验优先抽取算法的策略前 N 轮平均奖励值明显高于原始算法,且捕食者采取该机制,对提高奖励、加速训练有明显的提升,验证了算法的优越性.



(a)　捕食者采取 MADDPG 算法



(b)　捕食者采取 DDPG 算法

图 5　捕食者前 N 轮平均奖励曲线

4　结　论

为解决多智能体深度确定性策略梯度算法训练效率低的问题,本文利用多智能体缓存经验的策略评估网络误差和训练次数来评估经验的优先级,设计了经验优先抽取机制. 将该机制与 MADDPG 算法结合,提出了 PES-MADDPG 算法,并通过仿真实验验证了算法的可用性、优越性和适用性,而且在 DDPG 控制算法中依然可用. 随着智能体数量增多,需要训练的网络数量呈线性增长,同时,由于神经网络输入节点的线性增长,网络中的参数也呈线性增长,训练的复杂度增大. 降低算法复杂度的难点在于状态空间的增大以及神经网络训练复杂度的增加,可以采用的方法包括压缩状态空间,去除不必要的信息,以及智能体之间共享网络参数等. 下一步,将研究随着智能体数量增多如何降低 MADDPG 算法的复杂度.

参考文献(References)

[1]　Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. 2015, arXiv: 1509.02971.

[2]　Mnih V, Badia A P, Mirza M, et al. Asynchronous

methods for deep reinforcement learning[C]. International Conference on Machine Learning, New York, 2016: 1928-1937.
[3] Schulman J, Levine S, Abbeel P, et al. Trust region policy optimization[C]. Proceeding of the 32nd International Conference on Machine Learning. Lille, 2015: 1889-1897.
[4] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J]. 2017, arXiv: 1707.06347.
[5] Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms[C]. Proceeding of the International Conference on Machine Learning. Detroit, 2014: 387-395.
[6] 夏伟, 李慧云. 基于深度强化学习的自动驾驶策略学习方法[J]. 集成技术, 2017, 6(3): 29-40.
(Xia W, Li H Y. Training method of automatic driving strategy based on deep reinforcement learning[J]. Journal of Integration Technology, 2017, 6(3): 29-40.)
[7] 张斌, 何明, 陈希亮, 等. 改进DDPG算法在自动驾驶中的应用[J]. 计算机工程与应用, 2019, 55(10): 264-270.
(Zhang B, He M, Chen X L, et al. Self-driving via improved DDPG algorithm[J]. Computer Engineering and Applications, 2019, 55(10): 264-270.)
[8] Levine S, Finn C, Darrell T, et al. End-to-end training of deep visuomotor policies[J]. The Journal of Machine Learning Research, 2016, 17(1): 1334-1373.
[9] Lee D, Tang H, Zhang J O, et al. Modular architecture for starCraft II with deep reinforcement learning[C]. Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference. Edmonton, 2018: 187-193.
[10] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[J]. 2013, arXiv: 1312.5602.
[11] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529.
[12] Hernandez-Leal P, Kaisers M, Baarslag T, et al. A survey of learning in multiagent environments: Dealing with non-stationarity[J]. 2017, arXiv:1707.09183.
[13] Hernandez-Leal P, Kartal B, Taylor M E. Is multiagent deep reinforcement learning the answer or the question? A brief survey[J]. 2018, arXiv: 1810.05587.
[14] Palmer G, Tuyls K, Bloembergen D, et al. Lenient multi-agent deep reinforcement learning[C]. Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. Richland, 2018: 443-451.
[15] Omidshafiei S, Pazis J, Amato C, et al. Deep decentralized multi-task multi-agent reinforcement learning under partial observability[C]. Proceedings of the 34th International Conference on Machine Learning. Sydney, 2017: 2681-2690.
[16] Zheng Y, Meng Z, Hao J, et al. Weighted double deep multiagent reinforcement learning in stochastic cooperative environments[C]. Pacific Rim International Conference on Artificial Intelligence. Nanjing, 2018: 421-429.
[17] Jaderberg M, Czarnecki W M, Dunning I, et al. Human-level performance in first-person multiplayer games with population-based deep reinforcement learning[J]. Sience, 2019, 364(6443): 859-865.
[18] Sunehag P, Lever G, Gruslys A, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward[C]. Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. Richland, 2018: 2085-2087.
[19] Rashid T, Samvelyan M, De Witt C S, et al. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning[J]. The 35th International Conference on Machine Learning. Stockholm, 2018: 6846-6859.
[20] Foerster J N, Farquhar G, Afouras T, et al. Counterfactual multi-agent policy gradients[C]. The 32nd AAAI Conference on Artificial Intelligence. New Orleans, 2018: 2974-2982.
[21] Lowe R, Wu Y, Tamar A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[C]. Advances in Neural Information Processing Systems. Morgan Koufmann, 2017: 6379-6390.
[22] Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay[J]. 2015, arXiv: 1511.05952.
[23] 陈希亮, 曹雷, 李晨溪, 等. 基于重抽样优选缓存经验回放机制的深度强化学习方法[J]. 控制与决策, 2018, 33(4): 600-606.
(Chen X L, Cao L, Li C X, et al. Deep reinforcement learning via good choice resampling experience replay memory[J]. Control and Decision, 2018, 33(4): 600-606.)
[24] Brittain M, Bertram J, Yang X, et al. Prioritized sequence experience replay[J]. 2019, arXiv: 1905.12726.
[25] Zhang H, Xiong K, Bai J. Improved deep deterministic policy gradient algorithm based on prioritized sampling[C]. Proceedings of 2018 Chinese Intelligent Systems Conference. Singapore, 2019: 205-215.

作者简介

何明 (1978−), 男, 教授, 博士生导师, 从事智联网与无人化控制等研究, E-mail: paper_review@126.com;

张斌 (1990−), 男, 硕士生, 从事强化学习、人工智能的研究, E-mail: qdjmzb@qq.com;

柳强 (1983−), 男, 博士, 从事强化学习、智能控制理论的研究, E-mail: citizenliuqiang@163.com;

陈希亮 (1985−), 男, 副教授, 博士, 从事机器学习、决策支持理论与技术等研究, E-mail: 383618393@qq.com;

杨铖 (1990−), 男, 硕士生, 从事机器学习的研究, E-mail: 978008436@qq.com.

(责任编辑：孙艺红)