计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(12):43−51 [doi: 10.15888/j.cnki.csa.009315]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

融合两级注意力的多机器人强化学习导航①

张耀丹[1], 况立群[1,2,3], 焦世超[1,2,3], 韩慧妍[1,2,3], 薛红新[1,2,3]

[1](中北大学 计算机科学与技术学院, 太原 030051)
[2](机器视觉与虚拟现实山西省重点实验室 (中北大学), 太原 030051)
[3](山西省视觉信息处理及智能机器人工程研究中心, 太原 030051)
通信作者: 况立群, E-mail: kuang@nuc.edu.cn

摘　要: 针对多智能体强化学习中因智能体之间的复杂关系所导致的学习效率低及收敛速度慢的问题, 提出基于两级注意力机制的方法 MADDPG-Attention, 在 MADDPG 算法的 Critic 网络中增加了软硬两级注意力机制, 通过注意力机制学习智能体之间的可借鉴经验, 提升智能体之间的相互学习效率. 由于单层的软注意力机制会给完全不相关的智能体也赋予学习权重, 因此采用硬注意力判断两个智能体之间学习的必要性, 裁减无关信息的智能体, 再用软注意力判断两个智能体间学习的重要性, 按重要性分布来分配学习权重, 据此向有可用经验的智能体学习. 在多智能体粒子的合作导航环境上进行测试, 实验结果表明, MADDPG-Attention 算法对复杂关系的理解更为清晰, 在 3 种环境的导航成功率都达到了 90% 以上, 有效提高了学习效率, 加快了收敛速度.

关键词: 多智能体强化学习; 导航; MADDPG; 硬注意力; 软注意力

引用格式: 张耀丹,况立群,焦世超,韩慧妍,薛红新.融合两级注意力的多机器人强化学习导航.计算机系统应用,2023,32(12):43–51. http://www.c-s-a.org.cn/1003-3254/9315.html

Multi-robot Reinforcement Learning Navigation Incorporating Two Levels of Attention

ZHANG Yao-Dan[1], KUANG Li-Qun[1,2,3], JIAO Shi-Chao[1,2,3], HAN Hui-Yan[1,2,3], XUE Hong-Xin[1,2,3]

[1](School of Computer Science and Technology, North University of China, Taiyuan 030051, China)
[2](Shanxi Key Laboratory of Machine Vision and Virtual Reality (North University of China), Taiyuan 030051, China)
[3](Shanxi Province's Vision Information Processing and Intelligent Robot Engineering Research Center, Taiyuan 030051, China)

Abstract: To solve the low learning efficiency and slow convergence due to the complex relationship among intelligent agents in multi-agent reinforcement learning, this study proposes a two-level attention mechanism based on MADDPG-Attention. The mechanism adds soft and hard attention mechanisms to the Critic network of the MADDPG algorithm and learns the learnable experience among intelligent agents through the attention mechanism to improve the mutual learning efficiency of the agents. Since the single-level soft attention mechanism assigns learning weights to completely irrelevant intelligent agents, hard attention is employed to determine the necessity of learning between two intelligent agents, and the agents with irrelevant information are cut. Then soft attention is adopted to determine the importance of learning between two intelligent agents, and the learning weights are assigned according to the importance distribution to learn from the agents with available experience. Meanwhile, tests on a collaborative navigation environment with multi-agent particles show that the MADDPG-Attention algorithm has a clearer understanding of complex relationships and achieves a success rate of more than 90% in all three environments, which improves the learning efficiency and accelerates the convergence rate.

Key words: multi-agent reinforcement learning; navigation; MADDPG; hard attention; soft attention

① 基金项目: 国家自然科学基金 (62272426, 62106238); 山西省科技重大专项计划 (202201150401021); 山西省科技成果转化引导专项 (202104021301055); 山西省回国留学人员科研资助项目 (2020-113); 山西省基础研究计划 (202203021222027)
收稿时间: 2023-05-25; 修改时间: 2023-06-26; 采用时间: 2023-07-03; csa 在线出版时间: 2023-09-19
CNKI 网络首发时间: 2023-09-21

强化学习[1–3]在未知环境中的自适应性与自学习能力在导航问题[4–6]中表现出了较大的潜力, 但现有的算法大都应用于单智能体环境, 在多智能体环境中仍存在很多困难. 单智能体路径规划中智能体只用学习自己的策略, 所在的环境是稳定不变的, 奖励函数只需要考虑到目标点的距离以及与环境的碰撞问题. 而多智能体路径导航[7–11]中, 每个智能体的策略不断改变, 导致环境变得不稳定, 智能体间的交互也变得更加复杂, 智能体除了考虑与环境的碰撞问题, 还要考虑智能体之间的碰撞问题, 奖励函数设定也需要考虑各个智能体联合动作的影响, 大大增加了多智能体路径导航学习的难度, 出现效率低, 收敛速度慢的问题.

早期的工作尝试采用注意力机制[12]模拟智能体之间的关系, 学习可用的经验来加速多智能体强化学习过程. 文献 [13]提出的 ATOC 算法[13]采用一个双向的 LSTM 网络作为通信群组之间的通信信道, 通过整合共享信息以进行合作决策, 依据注意力模块来决定是否进行通信. 文献 [14]提出的 MAAC 算法[14]使用多头注意力机制减少对集中式 Critic 网络的输入, 适应了因智能体数量增加造成的输入空间的暴涨. 文献 [15]提出的 EPC 算法[15]在 Actor 和 Critic 网络都使用注意力机制解决网络输入维度增大的问题, 以获得更好的性能表现. 文献 [16]提出的 Qatten 算法[16]用多头注意力机制分解联合 Q 值, 度量每个智能体对全局系统的重要程度. 文献 [17]提出的 TarMAC 算法[17]用基于签名的软注意力机制 (soft attention) 建立有针对性的通信, 适应动态的智能体数量. 这些算法的主要思想都是利用软注意机制, 通过 Softmax 函数学习智能体之间的关注程度. 然而, Softmax 函数的输出值是一个相对值, 会将非零的小概率分配给不相关的智能体, 削弱了对相关智能体所给予的关注, 不能真正模拟智能体之间的关系.

针对软注意力机制分配权重不合理的问题, 本文用硬注意力机制 (hard attention) 弥补软注意力机制的缺陷, 提出了基于两级注意力机制的 MADDPG-Attention 算法. 在训练期间用硬注意力机制筛选需要关注的智能体, 对需要关注的智能体赋予概率值 1, 对不需要关注的智能体赋予概率值 0. 同时根据需要关注的程度利用软注意力机制分配权重, 建立智能体之间的学习关系, 加快多智能体强化学习的速度.

1 背景知识

1.1 马尔可夫博弈与纳什均衡

在强化学习领域, 待解决的问题通常被描述为马尔可夫决策过程 (MDP). 马尔可夫博弈[18]是马尔可夫决策过程在多智能体环境下的扩展. 纳什均衡是马尔科夫博弈问题的目标. 多智能体强化学习借助马尔科夫博弈将问题进行建模, 以纳什均衡为目标找到求解问题的方法.

马尔可夫博弈由多元组$(I,S,A,\Omega,T,R,\gamma)$构成, 其中$I=\{1,\cdots,n\}$为 n 个智能体的集合, 其中S为状态集合; $A=A_1\times A_2\times\cdots\times A_n$是联合动作空间, 其中$A_i=[a_i^1,\cdots,a_i^n]$; T为联合状态转移函数, 表示智能体执行联合动作A从当前状态到下一个状态转换函数的几率分布; $\Omega=\{O_1,\cdots,O_n\}$是联合部分观测的集合; $R=[R_1,\cdots,R_n]$为所有智能体联合奖励空间, 其中$S\times A_i\to R_i$; $\gamma\in[0,1]$为折扣因子, 代表智能体长期回报的重视程度. 智能体 i 收到部分观测O_i并根据随机策略π_{θ_i}生成下一个动作A_i, 根据状态转移函数T得到下一个状态S_{t+1}, 其中θ是智能体 i 的策略参数. 环境据此反馈一个全局奖励信号R_i. 每个智能体旨在最大化自己的总预期回报$R_i=\sum_{t=0}^{T}\gamma^t r_t^i$, t 是时间范围.

纳什均衡是在多个智能体中达成的一个不动点, 对于其中任意一个智能体来说, 无法通过采取其他的策略来获得更高的累积回报. 纳什均衡不一定是全局最优, 但它是在概率上最容易产生的结果, 是在学习时较容易收敛到的状态.

1.2 MADDPG 算法

MADDPG 算法 (multi-agent deep deterministic policy gradient)[19]是解决多智能体连续行为的确定性策略梯度算法. 它是 DDPG 算法[20–22]针对多智能体环境下的改进, 在 Critic 网络中加入了其他智能体的信息, 采用集中式训练分布式执行的架构, 在训练过程中由 Critic 网络集中式共享全局信息, 在执行过程中由 Actor 分布式获取局部信息, 即当前智能体的观测信息. 由于每个智能体都具有 Critic 网络, 因此 MADDPG 算法可以具有不同奖励函数的智能体, 在完全合作、完全竞争和混合关系的问题中都能取得较好效果.

MADDPG 最初是由解决连续行为的策略梯度 (PG) 算法[23]演化而来, PG 算法随着动作维度的增加,

计算性能消耗过大, 因此 DeepMind 提出了确定性策略梯度 (DPG) 算法[24]. 后来人们意识到了可以使用神经网络去拟合强化学习算法中的价值函数和策略函数, DeepMind 又提出了深度 Q 网络算法 (DQN)[25], 但不能解决连续空间的问题. DDPG 算法是 DPG 算法与 DQN 算法的结合, 将 DQN 中用来拟合 Q 函数的神经网络用在了 DPG 框架中, 实现了基于深度神经网络的 DPG 算法, 可以很好地适应连续空间的问题. 由于多智能体环境的不稳定性, DDPG 直接用于多智能体环境的表现不佳, 因此将 DDPG 算法扩展到多智能体深度确定性策略算法 MADDPG.

2 MADDPG-Attention 方法

2.1 两级注意力机制

随着深度强化学习的发展, 注意力机制在多智能体强化学习领域的应用越来越广, 它对重要信息的合理利用, 在一定程度上促进了智能体的学习效果, 特别是硬注意力机制和软注意力机制的作用非常显著.

硬注意力机制是从输入的所有元素中选择一个子集, 迫使模型只能关注重要元素, 完全忽略其他元素. 虽然这恰好可以帮助智能体筛选与自己相关的信息, 但硬注意力机制对相关信息的学习是同等概率, 缺乏侧重性. 软注意力机制通过计算元素的重要性分布来分配学习权重, 这对学习其他智能体的经验提供了方案. 然而软注意力机制通常将非零的小概率赋给不相关的元素, 削弱了对真正重要元素的关注.

基于以上工作的启发, 本文提出基于两级注意力机制的多智能体强化学习方法 MADDPG-Attention, 使用一层硬注意力机制将当前智能体与其他智能体按关联性区分, 再通过软注意力机制按重要性分布来分配学习权重, 达到对有用信息的合理利用, 从而提升 MADDPG 算法的性能.

硬注意力机制的结构如图 1 所示, 智能体的状态和动作进行拼接后经过一层全连接网络 MLP 和 ReLU 激活函数进行解码得到e. 然后经过一层门控循环神经网络 GRU 将e编码为一个特征向量 G, 将每个智能体的特征G_i与其他智能体的特征G_j分别拼接, 经过一层双向 GRU 得到每个智能体之间的关系. 最后经过一层 MLP 对智能体之间的关系进行分析得到智能体之间的权重, 经过 Gumble-Softmax 函数得到硬注意力权重 hard-weight.

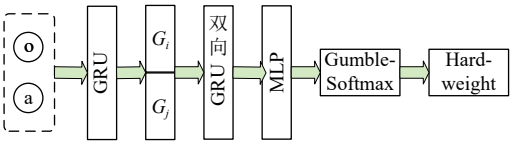


图 1 Hard-Attention 结构框架图

软注意力机制的结构如图 2 所示, 将每个智能体的特征G_i作为键 q, 其他智能体的特征G_j作为值 k, q 与 k 进行缩放点积得到智能体之间的相似值, 将相似值进行 Softmax 处理得到智能体之间的软注意力权重 soft-weight.

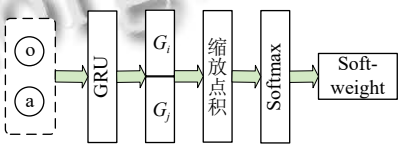


图 2 Soft-Attention 结构框架图

2.2 MADDPG-Attention 网络结构

MADDPG 算法中的 Critic 网络无差别的输入所有智能体的状态和动作, 学习没有重点, 针对该问题, 本文提出一种基于两级注意力机制的 MADDPG 方法. 该方法仍采用 MADDPG 算法集中训练分散执行的思想, 采用双重神经网络以及经验回放的结构, 具有训练稳定, 收敛速度快的特点.

Actor 网络根据当前状态$o_t=(o_1,\cdots,o_n)$选择当前动作$a_t=(a_1,\cdots,a_n)$, 将当前动作输入环境交互, 生成新状态o_{t+1}和奖励$r_t=(r_1,\cdots,r_n)$, 将(o_t,a_t,r_t,o_{t+1})作为样本放入经验回放池.

Attention_Critic 网络在训练阶段接收所有智能体的动作a_t和观测信息o_t, 输出值函数$Q_i^{\mu}(o_t,a_t)$, 其中$o_t=(o_1,\cdots,o_n)$, $a_t=(a_1,\cdots,a_n)$, μ为确定性策略, n为环境中智能体个数. 网络结构如图 3 所示.

Attention_Critic 网络将动作和观测信息进行拼接得到e. 将e送入硬注意力网络, 通过硬注意力机制判断其他智能体与当前智能体是否有相关性, 得到硬注意力权重 hard-weight. 得到硬注意力权重之后, 将e送入软注意力网络, 计算智能体之间的重要性分布来分配学习权重, 得到软注意力权重 soft-weights. 然后 soft-weights 连同 values (V) 和 hard-weights 进行点积得到其他智能体对第 i 个智能体的贡献, 最后经过一层 MLP 进行解码得到 Actor 网络的评价 Q 值.

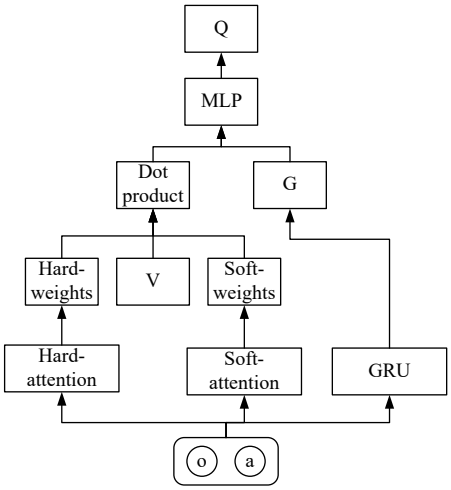


图 3 Attention_Critic 网络结构

Target_Actor 网络根据经验回放池中采样的下一状态o_{t+1}选择下一动作a_{t+1}. Actor 网络的更新公式如式 (4) 所示:

$$\nabla_{\theta_i} J(\mu_i) = E_{o,a\sim D}\left[\nabla_{\theta_i}\mu_i(a_i \mid o_i)\nabla_{a_i} Q_i^{\mu}(o,a)\big|_{a_i=\mu_i(o_i)}\right] \quad (1)$$

其中, μ_i为智能体i的确定性策略, θ为 μ的参数, D=$(o_1,\cdots,o_n,a_1,\cdots,a_n,r_1,\cdots,r_n)$是经验池, 包含了所有智能体的经验轨迹.

Target_Attention_Critic 网络接收经验池中采样的下一状态o_{t+1}和 Target_Actor 网络输出的动作a_{t+1}计算目标 Q 值. 通过 Critic 网络与 Target_Critic 网络的均方误差更新 Critic 网络. 其中 Critic 网络的更新公式计算如式 (5) 所示:

$$L(\theta_i) = E_{x,a,r,x'}\left[\left(Q_i^{\mu}(x,a_1,\cdots,a_n) - y\right)^2\right],$$
$$\text{where } y = r_i + \gamma \bar{Q}_i^{\mu'}\left(x',a_1',\cdots,a_n'\right)\big|_{a_j'=\mu_j'(o_j)} \quad (2)$$

其中, $\bar{Q}_i^{\mu'}$ 表示目标网络, $\mu' = \left[\mu_1',\cdots,\mu_n'\right]$为目标策略具有滞后更新的参数$\theta_j'$, $x = (o_1,\cdots,o_n)$为所有智能体的观测.

2.3 观测空间设计

多智能体强化学习中智能体通过与环境交互和试错来学习知识. 交互过程中智能体从环境中获得的信息称为观测空间, 依据观测空间智能体通过策略做出动作对环境进行回应. 合适的观测空间的设计对智能体的学习至关重要. 因此, 根据导航任务, 智能体的观测空间设计如表 1 所示. 实验环境是 2 维空间, 智能体和地标的个数都为 k. 每个智能体的观测空间包括当前智能体的速度信息、位置信息、距离目标地标的距离、距离其他地标的距离和距离其他智能体的距离. 第 i 个智能体的速度信息由 x 轴的速度和 y 轴的速度(v_{ix},v_{iy})表示, 位置信息由 x 轴的位置和 y 轴的位置(p_{ix},p_{iy})表示. 第 i 个智能体目标地标的位置由(g_{ix},g_{iy})表示, 距离目标地标的距离由$(g_{ix}-p_{ix},g_{iy}-p_{iy})$表示.

表 1 观测空间

观测信息	变量名	维度
第i个智能体的速度信息	(v_{ix},v_{iy})	2维
第i个智能体的位置信息	(p_{ix},p_{iy})	2维
第i个智能体距离目标地标的距离	$(g_{ix}-p_{ix},g_{iy}-p_{iy})$	2维

2.4 奖励函数设计

奖励是智能体在采取了特定动作之后所得到的及时反馈, 智能体根据反馈去学习达到目标. 根据最终目标合理地设置奖励对智能体的学习非常重要. 依据奖励密度, 奖励分为稀疏奖励和稠密奖励. 稀疏奖励指的是只有在智能体达到目标状态或者完成任务才会获得奖励, 会导致智能体采取大量的随机动作, 忽略强化学习中非常重要的环境信息, 导致学习速度变慢. 而稠密奖励指的是智能体行动的每一个步骤都有奖励或惩罚, 促使智能体很快地学习到环境的信息, 起到良好的引导作用, 但过多的步骤奖励设置可能导致智能体专注于基本动作的实现, 达到局部最优解. 因此本文在稀疏奖励和稠密奖励之间寻找平衡, 设置成功奖励 O、距离奖励 D 和碰撞惩罚 C, 既保存了稀疏奖励的必要性, 又没有过分增加步骤奖励设置. 奖励函数R设置如式 (3) 所示:

$$R = -\sum_{i=1}^{n} D_i + C + O \quad (3)$$

其中, 距离奖励D_i设置如式 (4) 所示, D_i为第 i 个智能体与目标地标的欧氏距离.

$$D_i = \sqrt{(g_{ix}-p_{ix})^2 + (g_{iy}-p_{iy})^2} \quad (4)$$

碰撞惩罚C设置如式 (5) 所示, C 为智能体之间相互碰撞时受到的惩罚.

$$C = \begin{cases} -1, & \text{发生碰撞} \\ 0, & \text{无碰撞} \end{cases} \quad (5)$$

成功奖励O设置如式 (6) 所示, O为智能体到达目标地标时给予的奖励.

$$O = \begin{cases} 1, & \text{覆盖地标} \\ 0, & \text{未覆盖地标} \end{cases} \quad (6)$$

3 实验结果

在多智能体强化学习领域中广泛使用的多智能体粒子环境上进行了仿真实验, 验证并对比了 MADDPG-Attention 算法、MADDPG 算法、MAAC 算法和 DDMA 算法的实验结果. 以测试阶段每 100 回合单个智能体的平均奖励、成功率和碰撞率为指标对算法性能进行评判. 其中 MADDPG 算法增加了高斯噪声探索, MAAC 算法 Critic 网络使用了多头注意力增加算法的可扩展性, DDMA 算法采用单智能体算法预学习多智能体策略, 并引入交互检测器提高学习效率.

3.1 多智能体粒子环境

目前常采用多智能体粒子环境来对多智能体算法进行验证, 本文以其中的合作导航为实验环境, 通过控制二维空间中的粒子, 实现对于各类多智能体强化学习算法的验证及对比分析. 实验中, 智能体是运动的, 地标是静态的, 每回合结束后智能体位置和地标位置都会随机设置, 每个智能体要学会在避免碰撞的同时覆盖地标.

为了使实验环境更加贴近真实环境, 将合作导航环境转换为小车导航, 小车作为智能体, 需要在不碰撞的情况下合理规划路线以到达目标.

实验环境如图 4 所示, 采用 3 种不同的环境进行训练和测试. 3 对 3 环境如图 4(a) 所示, 3 个小车合作导航到 3 个地标 (图标为房子), 训练开始之前为每个小车随机分配一个目标, 每个小车之间碰撞会受到惩罚, 到达目标则给予奖励. 4 对 2 环境如图 4(b) 所示, 4 个小车合作导航到两个地标, 每两个小车为一组, 随机为每组小车分配一个地标, 相同地标的小车碰撞不惩罚, 不同地标的小车碰撞会惩罚, 每个小车到达地标会给予奖励. 6 对 3 环境如图 4(c) 所示, 6 个小车合作导航到 3 个地标, 每两个小车为一组, 为每组小车分配一个地标, 相同地标的小车碰撞不惩罚, 不同地标的小车碰撞会惩罚, 每个小车到达地标会给予奖励.

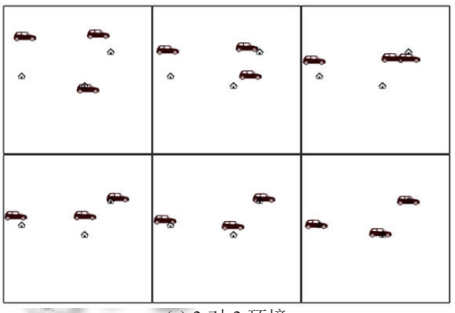
3.2 实验参数设置

MADDPG-Attention 算法的实验参数如表 2 所示.

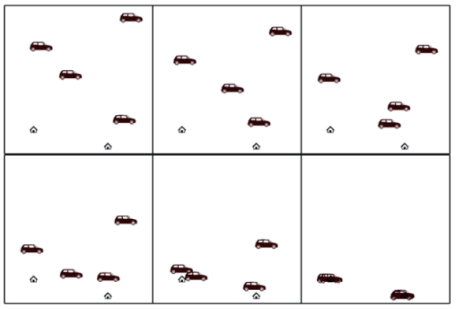
3.3 结果分析

MADDPG-Attention、MADDPG、MAAC 和 DDMA 这 4 种算法在 3 对 3 环境、4 对 2 环境和 6 对 3 环境中训练 100 000 回合, 每 1 000 回合进行一次测试. 测试 1 000 回合, 每 100 回合的进行一次统计, 取 10 次统计的平均值作为实验结果.

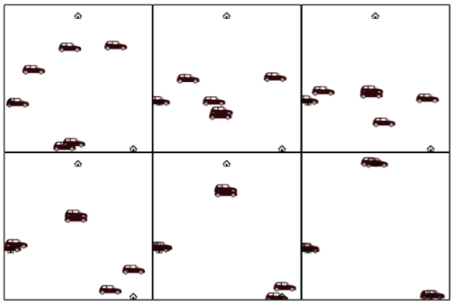
为了评估 4 种方法在测试环境中的性能, 选取成功率, 即 100 回合平均覆盖地标的次数进行测试. 3 对 3 环境的测试结果如图 5(a) 所示, 4 对 2 环境的测试结果如图 5(b) 所示, 6 对 3 环境的测试结果如图 5(c) 所示.



(a) 3 对 3 环境



(b) 4 对 2 环境

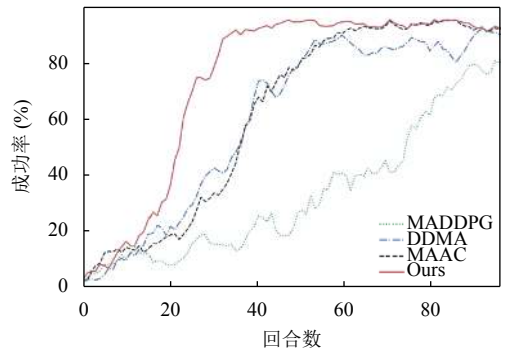


(c) 6 对 3 环境

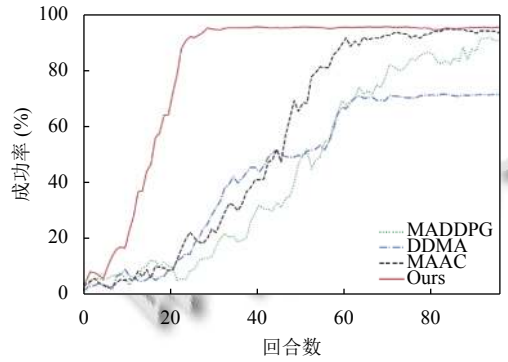
图 4 合作导航

表 2 实验参数

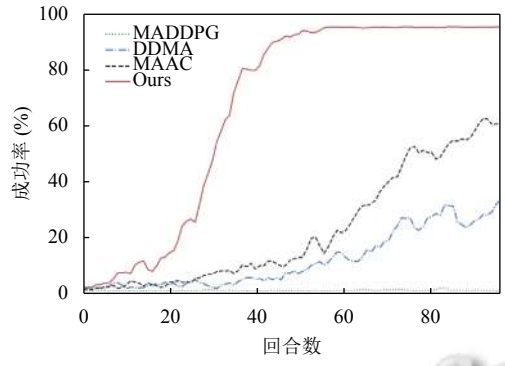
参数名	参数值
Actor网络学习率α	0.000 1
Critic网络学习率α	0.001
网络更新率τ	0.01
衰减度γ	0.95
探索率ε	0.1
噪声	0.1
样本池容量	500 000
最小取样样本数	128
随机种子数	678
训练回合时间步长	25
总训练回合数	100 000



(a) 3 对 3 环境



(b) 4 对 2 环境



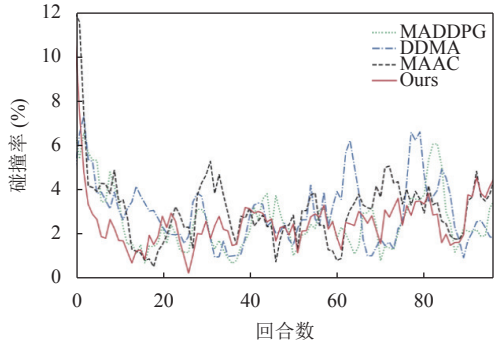
(c) 6 对 3 环境

图 5 成功率

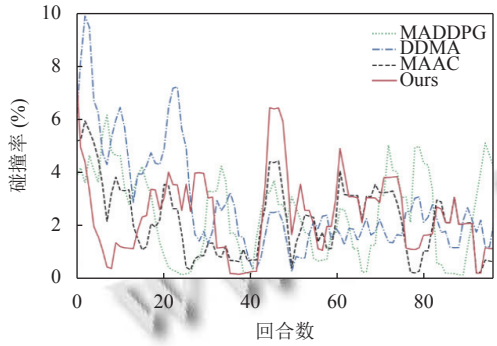
从图 5(a) 可以看出, MADDPG 算法学习速度较慢, 最终成功率也只能达到 80% 左右, 效果略差. MAAC 算法和 DDMA 算法 60 回合后成功率都可以到达 90% 以上, 可以较好地完成覆盖地标的任务. MADDPG-Attention 算法 25 回合左右成功率就收敛到了 90% 以上, 学习速度最快. 从图 5(b) 可以看出 MADDPG 算法学习速度稳步上升, 在 100 回合后成功率收敛到 93% 左右, 学习效果较好. MAAC 相对环境 1 来说, 学习速度有所下降, 60 回合后也能收敛到 95% 左右. DDMA 成功率出现明显下降, 60 回合后收敛到 72% 左右. MADDPG-Attention 算法收敛速度仍旧保持最快, 25 回合左右就收敛到了 96% 左右. 从图 5(c) 可以看出 MADDPG 算法完全无法完成学习导航的任务, 成功率一直为 0% 左右. MAAC 算法和 DDMA 算法 60 回合后成功率才有明显的上升, MAAC算法 100 000 回合后成功率达到 62% 左右, 效果略好, DDMA 算法 100 000 回合后成功率达到 30% 左右, 效果较差. MADDPG-Attention 算法收敛速度保持最快, 50 回合左右收敛到了 96% 左右. 总体来说, MADDPG-Attention 算法对于不同环境的迁移性很好, 随着智能体数目的增多仍能保持良好的表现.

使用平均碰撞次数评估 4 种算法的性能, 3 对 3 环境的测试结果如图 6(a) 所示, 4 对 2 环境的测试结果如图 6(b) 所示, 6 对 3 环境的测试结果如图 6(c) 所示.

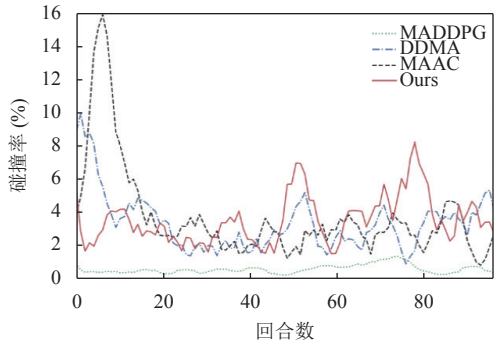
从图 6(a) 可以看出, MADDPG 算法碰撞率初期处于较低的状态, 但 80 回合后出现了较高的起伏, 碰撞率略高, 结合成功率来看, MADDPG 算法前期探索性不强, 80 回合后才有明显的学习效果. MAAC 算法处一开始碰撞率高, 且波动一直较大, 结合成功率来看, MAAC 算法通过大量的碰撞去增加探索性来学习导航的目标, 学习的观念比较激进. DDMA 算法碰撞率虽然一开始处于较低状态, 但 60 回合后碰撞率出现明显增加, 且持续时间较长, 结合成功率来看, DDMA 算法在 60–80 回合之间成功率出现了下降, 说明 DDMA 算法的稳定性较差. MADDPG-Attention 算法的碰撞率一直处于较低的水平, 没有明显波动, 结合成功率来看, 学习速度也一直保持稳定上升, 说明 MADDPG-Attention 算法的稳定性以及学习效率都较好. 从图 6(b) 可以看出, 4 种算法因为智能体有同一目标的设定碰撞率有不同程度的增加, MADDPG 算法碰撞率一直处于稳定的波动, 没有很好的学习到不碰撞的任务. MAAC 算法、DDMA 算法和 MADDPG-Attention 算法虽然前期碰撞率高, 但 80 回合后都保持在 2% 以下, 对不碰撞任务的学习效果较好. 图 6(c) 可以看出, MADDPG 算法虽然碰撞率在 1% 左右, 但结合成功率来看, 没有学习到任何任务. MAAC 算法仍旧保持激进的学习观念, 初期的碰撞率非常高, 且 80 回合后波动较大. DDMA 算法一直保持较大的波动学习, 学习效果不稳定. MADDPG-Attention 算法虽然前期碰撞率较低, 但 45 回合后出现了较大的波动, 结合成功率来看, 45 回合后成功率逐渐趋于收敛, 说明 MADDPG-Attention 算法以牺牲一定的碰撞率来学习导航的任务.



(a) 3 对 3 环境



(b) 4 对 2 环境



(c) 6 对 3 环境

图 6 碰撞率

3 种实验环境中的实验结果显示, 随着环境复杂度的变化和智能体数目的增加, MADDPG 算法逐渐失去了学习能力, 环境迁移性较差. MAAC 算法以碰撞去学习探索, 学习观念较为激进, 学习效果较好. DDMA 算法学习效果不稳定, 学习效果较差. MADDPG-Attention 算法在 3 种环境中都保持了良好的学习效果, 环境的迁移性较好.

将 3 种环境的成功率和碰撞率结果汇总在表 3 中, 其中最优结果用粗体表示, 结果保留两位小数.

实验结果表示, 在 3 对 3 环境中, 本文算法碰撞率虽然较高, 但是成功率也最高, 相比 MADDPG、MAAC 和 DDMA 算法, 本文算法的成功率分别提高了 10.38%、0.95% 和 2.11%. 在 4 对 2 环境中, 本文算法碰撞率较低, 成功率保持最高, 相比 MADDPG、MAAC 和 DDMA 算法, 本文算法的成功率分别提高了 3.68%、1.37% 和 24.4%. 在 6 对 3 环境中, 本文算法碰撞率略高, 成功率仍然保持最高, 相比 MADDPG、MAAC 和 DDMA 算法, 本文算法的成功率分别提高了 96.32%、34.33% 和 65.23%.

表 3 多个环境下的导航测试结果

测试环境	测试算法	成功率 (%)	碰撞率 (%)
3对3	MADDPG	82.04	2.42
	MAAC	91.47	3.87
	DDMA	90.31	**1.89**
	Ours	**92.42**	3.86
4对2	MADDPG	92.6	3.13
	MAAC	94.91	**1.35**
	DDMA	71.88	2.16
	Ours	**96.28**	1.59
6对3	MADDPG	0.07	**0.27**
	MAAC	62.06	2.66
	DDMA	31.16	4.17
	Ours	**96.39**	2.74

总体来看, 在 3 对 3 环境中, 4 种算法都有差不多的效果, 但在 4 对 2 环境和 6 对 3 环境中存在具有相同目标的智能体, MADDPG 和 DDMA 算法对每个智能体给予同等的关注, 智能体数目增多后智能体之间的关系变得复杂, 无法学习到导航任务, 效果出现明显下降, 使用软注意力的 MAAC 算法虽然可以学习其他智能体的一些经验, 但随着智能体数目的增多, 不同目标的智能体会占据越来越多的不必要的注意力, 因此效果出现一定下降, 而 MADDPG-Attention 算法利用两级注意力机制只关注同一目标的智能体, 因而可以保持优越的性能. 综上所述, MADDPG-Attention 算法的整体表现优于 MADDPG、MAAC 和 DDMA 算法.

3.4 实验探究

为了进一步研究两级注意力机制对算法性能的影响, 对两级注意力的组合方式和单级注意力对 MADDPG 算法的表现进行实验分析.

3.4.1 注意力组合探究

为了探索更优秀的注意力组合方式, 选取硬注意力与软注意力的乘积和硬注意力与软注意力的线性加权两种方式在 6 对 3 环境进行实验. 其中线性加权方式软注意力与硬注意力线性权值之和为 1, 软注意力权值选取 (0.1, 0.3, 0.5, 0.7, 0.9). 实验结果如图 7 所示.

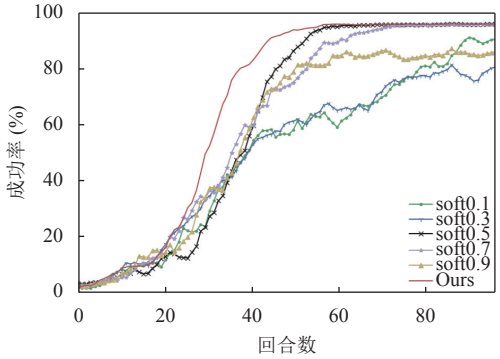


图 7 组合探究成功率

实验结果显示, 本文硬注意力与软注意力乘积的方式优于所有线性加权的方式. 两种注意力的乘积可以把不相关智能体的权重赋为零, 而线性加权的方式仍会保留不相关智能体的权重, 只是会进一步缩小不相关智能体的权重, 虽然对软注意力有一定的帮助, 但选用两种注意力采取相乘的方式更为合适.

3.4.2 消融实验

为了验证硬注意机制和软注意力机制对 MADDPG 算法性能的影响, 在 6 对 3 的环境中进行消融实验对比, 其中包括的算法有 3 种: 仅使用硬注意力机制的 MADDPG+Hard 算法、仅使用软注意力机制的 MADDPG+Soft 算法和本文基于两级注意力机制的 MADDPG 算法. 消融实验结果如图 8 所示.

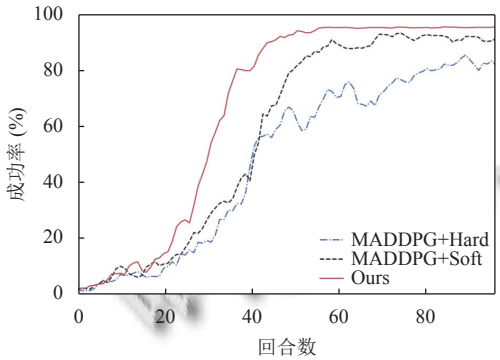


图 8 消融成功率

实验结果显示, MADDPG+Hard 算法的成功率在 80 回合后收敛到 80% 左右, 学习速度最慢; MADDPG+Soft 算法的成功率在 60 回合后收敛到 90% 左右, 学习速度有一定的提高; 本文基于两级注意力的 MADDPG 算法相较于单级注意力的算法无论是成功率还是学习速度均为最佳. 硬注意力弥补了软注意力权重分配不合理的缺陷, 两者相互耦合, 取得了更加优秀的性能.

4 结论与展望

本文提出了一种基于两级注意力的 MADDPG 算法. 该算法通过在 Critic 网络增加硬注意力和软注意力机制实现了对其他智能体的有选择性的学习, 并在合作导航环境中在扩展性以及成功率等方面表现出了优秀的性能. 下一步研究将考虑在 Actor 网络中使用长短期记忆方法, 通过参考前序信息, 做到有预测的导航.

参考文献

1 李茹杨, 彭慧民, 李仁刚, 等. 强化学习算法与应用综述. 计算机系统应用, 2020, 29(12): 13–25. [doi: 10.15888/j.cnki.csa.007701]

2 Lyu XG, Baisero A, Xiao YC, *et al*. A deeper understanding of state-based critics in multi-agent reinforcement learning. Proceedings of the 37th AAAI Conference on Artificial Intelligence. Washington: AAAI Press, 2022. 9396–9404.

3 Li PY, Tang HY, Yang TP, *et al*. PMIC: Improving multi-agent reinforcement learning with progressive mutual information collaboration. Proceedings of the 2022 International Conference on Machine Learning. Baltimore: PMLR, 2022. 12979–12997.

4 Papoudakis G, Christianos F, Schäfer L, *et al*. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. arXiv:2006.07869, 2021.

5 Skrynnik A, Andreychuk A, Yakovlev K, *et al*. POGEMA: Partially observable grid environment for multiple agents. arXiv:2206.10944, 2022.

6 Christianos F, Schäfer L, Albrecht S V. Shared experience actor-critic for multi-agent reinforcement learning. Proceedings of the 34th International Conference on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 898.

7 项宇, 秦进, 袁琳琳. 结合向前状态预测和隐空间约束的强化学习表示算法. 计算机系统应用, 2022, 31(11): 148–156. [doi: 10.15888/j.cnki.csa.008801]

8 Arpino CPD, Liu C, Goebel P, *et al*. Robot navigation in constrained pedestrian environments using reinforcement learning. Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA). Xi ’an: IEEE, 2021. 1140–1146.

9 Marchesini E, Farinelli A. Discrete deep reinforcement learning for mapless navigation. Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA). Paris: IEEE, 2020. 10688–10694.

10 Liu L, Dugas D, Cesari G, *et al*. Robot navigation in crowded

environments using deep reinforcement learning. Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Las Vegas: IEEE, 2020. 5671–5677.

11 Koh S, Zhou B, Fang H, *et al*. Real-time deep reinforcement learning based vehicle navigation. Applied Soft Computing, 2020, 96: 106694. [doi: 10.1016/j.asoc.2020.106694]

12 Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6000–6010.

13 Jiang JC, Lu ZQ. Learning attentional communication for multi-agent cooperation. Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montréal: Curran Associates Inc., 2018. 7265–7275.

14 Iqbal S, Sha F. Actor-attention-critic for multi-agent reinforcement learning. Proceedings of the 36th International Conference on Machine Learning. Long Beach: PMLR, 2019. 2961–2970.

15 Long Q, Zhou ZH, Gupta A, *et al*. Evolutionary population curriculum for scaling multi-agent reinforcement learning. Proceedings of the 8th International Conference on Learning Representations. Addis Ababa: OpenReview.net, 2020.

16 Yang YD, Hao JY, Liao B, *et al*. Qatten: A general framework for cooperative multiagent reinforcement learning. arXiv:2002.03939, 2020.

17 Das A, Gervet T, Romoff J, *et al*. TarMAC: Targeted multi-agent communication. Proceedings of the 36th International Conference on Machine Learning. Long Beach: PMLR, 2019. 1538–1546.

18 Littman ML. Markov games as a framework for multi-agent reinforcement learning. Proceedings of the 11th International Conference on International Conference on Machine Learning. New Brunswick: Morgan Kaufmann Publishers Inc., 1994. 157–163.

19 Lowe R, Wu Y, Tamar A, *et al*. Multi-agent actor-critic for mixed cooperative-competitive environments. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6382–6393.

20 殷雨竹, 陈建平, 傅启明, 等. 基于自监督网络的 DDPG 算法的建筑能耗控制. 计算机系统应用, 2022, 31(2): 161–167. [doi: 10.15888/j.cnki.csa.008365]

21 李卓远, 张德平. 基于 BN-DDPG 轻量级强化学习算法的智能兵棋推演. 计算机系统应用, 2023, 32(4): 293–299. [doi: 10.15888/j.cnki.csa.009015]

22 Lillicrap TP, Hunt JJ, Pritzel A, *et al*. Continuous control with deep reinforcement learning. Proceedings of the 4th International Conference on Learning Representations. San Juan: ICLR, 2016.

23 Sutton RS, McAllester D, Singh S, *et al*. Policy gradient methods for reinforcement learning with function approximation. Proceedings of the 12th International Conference on Neural Information Processing Systems. Denver: MIT Press, 1999. 1057–1063.

24 Silver D, Lever G, Heess N, *et al*. Deterministic policy gradient algorithms. Proceedings of the 31st International Conference on International Conference on Machine Learning. Beijing: JMLR.org, 2014: I-387–I-392.

25 Mnih V, Kavukcuoglu K, Silver D, *et al*. Playing atari with deep reinforcement learning. arXiv:1312.5602, 2013.

(校对责编: 牛欣悦)