(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 118428444 A
(43)申请公布日 2024.08.02

(21)申请号 202410679202.7

(22)申请日 2024.05.29

(71)申请人 北京航空航天大学
地址 100191 北京市海淀区学院路37号

(72)发明人 石荣晔 王力 吴文峻 于鑫 周号益

(74)专利代理机构 北京永创新实专利事务所 11121
专利代理师 易卜

(51)Int.Cl.
G06N 3/092(2023.01)
G06N 3/042(2023.01)
G06N 3/048(2023.01)
G06F 18/213(2023.01)

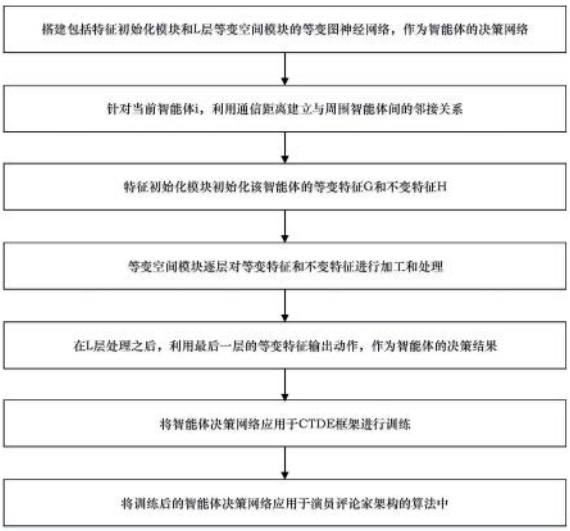
权利要求书2页 说明书10页 附图3页

(54)发明名称

一种基于等变图神经网络的多智能体强化学习方法

(57)摘要

本发明公开了一种基于等变图神经网络的多智能体强化学习方法,属于多智能体强化学习领域;首先,搭建包括特征初始化模块和等变空间模块的等变图神经网络,作为智能体的决策网络;针对当前智能体i,建立与周围智能体间的邻接关系;然后,初始化该智能体的等变特征和不变特征;逐层进行不变特征的提取,更新智能体的不变特征并从中提取智能体的深层不变信息;其次进行等变特征的更新,提取深层几何信息并保持等变特性。同时设计了具有等变特性的激活函数以增强模型的表征能力。最后,输出动作为智能体的决策结果,将该智能体决策网络应用于演员评论家架构的算法中;本发明加快了收敛速度和收敛性能,促进解决复杂多变问题的效果的意义。



CN 118428444 A

1.一种基于等变图神经网络的多智能体强化学习方法,其特征在于,具体步骤如下:

首先,搭建包括特征初始化模块和L层等变空间模块的等变图神经网络,作为智能体的决策网络;

然后,针对当前智能体i,利用通信距离建立与周围智能体间的邻接关系;特征初始化模块初始化该智能体的等变特征G和不变特征H;等变空间模块逐层对等变特征和不变特征进行加工和处理;在L层处理之后,利用最后一层的等变特征输出动作,作为智能体的决策结果;

具体为:

步骤301、利用不变特征更新模块提取智能体的深层不变特征$H_i^{(l)}$,并进行更新;

针对智能体i第l层的不变特征$H_i^{(l)} \in R^V$和等变特征$G_i^{(l)} \in R^U$,通过下面公式更新下一层不变特征$H_i^{(l+1)}$:

$$\mathbf{m}_{ij} = f_g\left(H_i^{(l)}, H_j^{(l)}, \| G_i^{(l)} - G_j^{(l)} \|^2, e_{ij}\right)$$

$$H_i^{(l+1)} = f_h\left(H_i^{(l)}, \sum_{j \in comm_n(i)} m_{ij}\right)$$

其中m_{ij}表示智能体i和智能体j之间传递的消息向量,函数$f_g(.)$和$f_h(.)$采用多层感知机实现,$||.||^2$表示二范数,(.,.,.)表示拼接操作符,comm_n(i)表示与智能体i在通信范围内的邻居集合,e_{ij}表示智能体i和智能体j之间的标量边特征;

步骤302、将等变特征和更新后的不变特征,更新等变非线性激活函数的系数;

激活函数更新的公式如下:

$$\left(\hat{H}, \hat{G}\right) = T_{\text{act}}(H, G) = (f_1(H), f_2(H) \odot G)$$

其中,T_{act}表示等变激活函数类的通用表示,$\hat{H}$表示变换后的不变特征,G是变换后的等变特征,⊙表示逐元素乘法;f_1是对不变特征所采用的非线性激活函数;f_2利用不变特征计算缩放系数并对等变特征进行缩放;

步骤303、利用等变特征更新模块提取智能体的空间信息,并通过更新后的等变非线性激活函数,对等变特征$G_i^{(l)}$进行缩放,生成最终的决策动作,同时确保等变性;

针对智能体i第l层的不变特征$H_i^{(l)} \in R^V$和等变特征$G_i^{(l)} \in R^U$,通过下面公式更新下一层等变特征$G_i^{(l+1)}$:

$$G_i^{(l+1)} = f_2\left(H_i^{(l)}\right) G_i^{(l)} + D \sum_{j \in comm_n(i)} \left(G_i^{(l)} - G_j^{(l)}\right) f_m(m_{ij})$$

其中,$\mathbf{G}_i^{(l+1)}$表示更新后的等变特征向量,$f_2\left(H_i^{(l)}\right)$表示经过不变特征计算的等变激活函数的输出,加权函数f_m采用多层感知机实现;$\mathbf{G}_i^{(l)} - \mathbf{G}_j^{(l)}$项感知智能体特征与邻居智能体特征的差异,D为加权系数;

最后,将智能体决策网络应用于CTDE框架进行训练,并将训练后的智能体决策网络应用于演员评论家架构的算法中。

2.如权利要求1所述的一种基于等变图神经网络的多智能体强化学习方法,其特征在于,所述邻接关系中,在通信距离内的智能体视作相邻,通信矩阵用e表示。

3.如权利要求1所述的一种基于等变图神经网络的多智能体强化学习方法,其特征在于,所述等变特征指会随旋转发生改变的特征,包括场景中智能体的绝对坐标和绝对速度;不变特征指不随场景旋转改变的特征,包括智能体之间的距离和相对方位,速度的大小。

4.如权利要求1所述的一种基于等变图神经网络的多智能体强化学习方法,其特征在于,所述训练阶段,利用完全图信息,在所有智能体之间进行消息聚合和传递;在分布式执行阶段,仅在每个智能体的邻居范围内进行消息聚合和传递。

5.如权利要求1所述的一种基于等变图神经网络的多智能体强化学习方法,其特征在于,所述演员评论家架构的算法中,策略网络使用等变特征来控制动作,而值网络则使用不变特征来对状态价值进行估计。

一种基于等变图神经网络的多智能体强化学习方法

技术领域

[0001] 本发明属于多智能体强化学习领域,涉及一种基于等变图神经网络的多智能体强化学习方法。

背景技术

[0002] 多智能体强化学习(Multi-Agent Reinforcement Learning,MARL)作为一种前沿的机器学习技术,是实现群体智能的重要途径之一,在模拟复杂交互环境中的决策过程方面显示出极大潜力。近年来,MARL在众多领域中的应用日益增加,包括实时战略游戏、纸牌游戏、自动驾驶、多机器人系统和智能电网管理等,而且在大多数应用上都能取得较好的效果。

[0003] 但是,MARL在实际应用中仍面临一系列挑战,最重要的是低样本效率问题。在许多实际情况下,强化学习的算法框架要求智能体与环境进行大量交互来学习有效的策略,而智能体却不利用先前的知识。这种大量的试错学习方式主要依赖暴力计算,不仅耗时且成本高昂,有时难以收敛,而且在许多实际情况下是不可行的。

[0004] 此外,现实世界的任务通常呈现出部分可观察性的特征,增加了智能体学习到有效策略所需样本的数量。尽管多智能体强化学习被证明是一种有潜力的方法,但在物理世界应用多智能体强化学习面临巨大的挑战,探索现实世界时产生样本的效率低下,且存在破坏硬件的风险。在真实场景的应用中,通常需要首先在仿真环境训练智能体,再将训练好的智能体迁移到真实物理世界中并针对真实环境与仿真环境的差异进行修正和调整。尽管这样的方式可能实现较好的性能,但不可避免地需要与环境进行大量交互,真实环境可能也需要大量的交互,因此不论是虚拟环境中还是真实环境中,样本效率低下是一个MARL中普遍存在的问题。因此,如何提高MARL系统与环境交互的样本效率,即如何在较少的交互次数下快速学习到有效的策略,成为了一个亟待解决的问题。

[0005] 提升多智能体强化学习(MARL)的样本效率,不仅能降低学习过程中的资源消耗,其更深远的意义在于增强MARL在现实世界应用的可行性与实用性。在自动驾驶、多机器人系统导航等关键技术领域,样本效率的提高直接关联到整个系统的安全性,可行性与运行效率,是推动这些技术商业化和广泛应用的核心驱动力。因此,深入探究如何通过促进智能体间的有效交互、整合历史学习经验、以及创新高效学习算法等策略来提升MARL的样本效率,具有不可忽视的理论价值和广泛的应用前景。

[0006] 在单智能体强化学习(Single-Agent Reinforcement Learning)中,B.Ravindran等人最初提出利用对称性来减少状态和动作空间中的冗余。近年来,在深度强化学习中利用对称性的方法大致可分为两类:数据增强和网络架构设计;利用对称性进行数据扩充的方式主要适用于基于图像的任务,需要在模型的训练阶段通过图像变换来生成额外数据进行训练。受到计算机视觉中数据增强任务的启发,Michael Laskin等人提出了Reinforcement Learning with Augmented Data(RAD)方法,将数据增强技术融合到强化学习中,在智能体训练的缓冲区中随机抽取数据并采用随机裁剪,随机旋转等方式在保留

时序性的同时对数据进行变换增强,然后再将变换后的数据用于强化学习的训练,能有效提高数据效率和泛化能力。在此基础上,Ilya Kostrikov等人提出了Data-regularized Q(DrQ)方法,在数据增强的基础上对多个变换数据的状态价值函数进行平均,并将平均值作为估计,以降低状态价值函数估计的方差,并可灵活地选择性应用各种变换,以提高采样的效率和学习的稳定性。Lin等人提出一个用于深度强化学习中数据增强的通用框架,它扩展了传统观念的经验回放缓冲器,提出不变变换经验重放模块,利用反射对称性和特殊任务对于目标位置有一定的误差容忍范围的特点进行数据增强,有一定的创新性。Fabio Amadio等人提出一种在双手协调任务中利用对称性的方法,将一个手臂的动作通过对称性映射到另一个手臂,使用相对熵搜索算法从用户提供的示范中进行学习,能减少参数空间的维度,加快学习速度。

[0007] 数据扩充也可以与损失函数结合使用,以更好地约束对称性。Mondal等人通过引入动作等变转移损失,群等变转移损失和动作不变奖励损失,三个损失函数来编码等变性或不变性的限制,能在隐层状态表示中保留更多关于奖励函数分布的信息,有效地改善和提升算法性能。

[0008] 尽管以上方法都能在一定程度上利用对称性提升样本效率,但本质上都是利用对称性进行样本扩充或加上对称性约束,期望模型能学到对称性的知识,达到的都不是完美的对称,且实施起来较为繁琐。

[0009] 网络设计方法包括创建隐式嵌入任务相关先验知识的特定体系结构。例如,联合状态动作空间中的对称性可以通过策略网络的实现来表达。Elise van der Pol及合作者开发了一种针对马尔可夫决策过程(Markov decision process,MDP)的同态网络,旨在通过利用状态-动作空间的等变性原理来缩小解空间,从而提升训练效率。该网络基于MDP的一致状态-动作空间具有的等变性特征,利用线性变换和施密特正交化构造具有等变性的网络并将等变性作为先验知识集成到策略和价值网络中,有效地减少了模型需要探索的解空间大小,为解决复杂决策问题提供了一种高效的框架。Dian Wang等人在强化学习两种经典算法Deep Q-Network(DQN)和Soft Actor-Critic(SAC)算法中利用具有等变结构的网络层搭建神经网络,使网络嵌入了对称性知识,提高了样本效率。Hai Nguyen等人将MDP的框架扩展到了部分可观测的设置中,利用等变卷积神经网络模块用于特征提取和输出,并设计了处理时间依赖性的等变循环神经网络模块,探索了部分可观测设置下强化学习智能体的学习效果。采用网络结构设计融入对称性的方法能达到较为完美的对称性,但单智能体领域的方法未建模智能体之间的交互关系,在扩展到多智能体领域有一定的限制。

[0010] 在多智能体强化学习(MARL)中,对称性的有效利用,特别是排列不变性和旋转对称性是至关重要的。该领域的许多研究强调了置换不变性的优势,如图1所示,其中改变智能体考虑的顺序不会影响系统行为,鼓励广泛的参数共享和提高数据效率。在解决MARL中状态空间随智能体数量指数增长的问题上,Jianye Hao及其团队提出了一种依赖于排列不变性(Permutation Invariance,PI)和排列等变性(Permutation Equivariance,PE)的框架用于简化多智能体状态空间的复杂性。设计了Dynamic Permutation Network(DPN)和Hyper Policy Network(HPN),通过构建专门针对实体级别的PI输入和PE输出网络模块,实现了对状态空间与动作空间的分解和端到端连接。DPN通过模块选择网络使得输入顺序改变时其权重矩阵也随之改变,而加权求和后的结果不变,从而可以实现PE或PI。HPN采用超

网络技术,直接生成权重矩阵作为模块选择网络,进一步增强了模型的表达能力。此框架在多个任务上展现了优异的性能,但其利用的是置换不变性而非对称不变性,且此方法难以扩展到对称不变性中。

[0011] 通过将对称性变换应用于经验回放池的样本,从而增加样本数量,也是一种利用对称性的方式。受到多智能体领域中排列不变性的启发,Zhenhui Ye等人提出了一种名为“经验增强”的新型数据增强方法,通过对多智能体系统中智能体的状态数据进行排列变换来增加经验回放池中的样本数量,从而生成大量的额外数据。此方法旨在克服现有单智能体强化学习领域中,数据增强技术难以直接应用于多智能体场景的挑战,同时解决了当前多智能体增强技术不适用于异构场景的问题。但同样利用数据增强的方法帮助模型学到对称性先验知识达到的也是非完美对称,只是对于对称性在一定程度上的利用。

[0012] 此外,多智能体MDP同态网络对旋转对称性的探索在提高数据效率方面也显示出了前景。Elise van der Pol等人提出一种分解方法将全局对称性分解为局部对称性,在分布式执行时通过交换局部信息来共享全局信息,并采用等变策略网络使得智能体具有等变性,且比较了其相较于非等变方法的优势。Xin Yu提出了Symmetry Prior Exploitation (ESP)方法,通过数据增强并在损失函数中添加等变的正则化约束项,从而在策略网络中嵌入对称性,并在虚拟仿真场景和真实场景中都验证了ESP方法的有效性。

[0013] 以上方法更多的是在某种程度上利用对称性来扩增样本或作为网络的约束,而并未保持完美的对称性,若能利用网络结构本身的特性在网络结构中嵌入对称性,有望取得更好的效果,提高样本效率,使MARL具有更加广泛的应用前景。

发明内容

[0014] 针对上述问题,本发明提出了一种基于等变图神经网络的多智能体强化学习方法,采用了图神经网络和多智能体强化学习技术,通过利用内嵌对称性的等变图神经网络EGNN作为多智能体强化学习的策略网络和值网络,使策略网络具有等变特性,值网络具有不变特性,同时设计了具有等变特性的激活函数以增强模型的表征能力,达到了将对称性先验知识嵌入多智能体强化学习智能体的技术效果,具有提升样本效率,加快多智能体强化学习的收敛性能的优势,和解决多智能体强化学习暴力试错带来的样本效率低下的问题,加快收敛速度和收敛性能,促进解决复杂多变问题的效果的意义。

[0015] 所述基于等变图神经网络的多智能体强化学习算法,具体步骤如下:

[0016] 步骤一,搭建包括特征初始化模块和L层等变空间模块的等变图神经网络,作为智能体的决策网络;

[0017] 步骤二、针对当前智能体i,利用通信距离建立与周围智能体间的邻接关系;

[0018] 在通信距离内的智能体视作相邻,通信矩阵用e表示。

[0019] 步骤三、特征初始化模块初始化该智能体的等变特征G和不变特征H;

[0020] 其中等变特征指会随旋转发生改变的特征,包括场景中智能体的绝对坐标和绝对速度;不变特征指不随场景旋转改变的特征,包括智能体之间的距离和相对方位,速度的大小等。

[0021] 步骤四、等变空间模块逐层对等变特征和不变特征进行加工和处理;

[0022] 智能体i的初始不变特征为$H_i^{(0)}$和初始等变特征为$G_i^{(0)}$;

[0023] 具体为：

[0024] 步骤301、利用不变特征更新模块提取智能体的深层不变特征$H_i^{(l)}$，并进行更新。

[0025] 针对智能体i第l层的不变特征$H_i^{(l)} \in R^V$和等变特征$G_i^{(l)} \in R^U$，通过下面公式更新下一层不变特征$H_i^{(l+1)}$：

[0026]
$$\mathbf{m}_{ij} = f_g\left(H_i^{(l)}, H_j^{(l)}, \| G_i^{(l)} - G_j^{(l)} \|^2, e_{ij}\right)$$

$$H_i^{(l+1)} = f_h\left(H_i^{(l)}, \sum_{j \in comm_n(i)} m_{ij}\right)$$

[0027] 其中m_{ij}表示智能体i和智能体j之间传递的消息向量，函数$f_g(.)$和$f_h(.)$采用多层感知机实现，$||.||^2$表示二范数，(.,.,.)表示拼接操作符，comm_n(i)表示与智能体i在通信范围内的邻居集合，e_{ij}表示智能体i和智能体j之间的标量边特征。

[0028] 步骤302、将等变特征和更新后的不变特征，更新等变非线性激活函数的系数；

[0029] 激活函数更新的公式如下：

[0030]
$$\left(\hat{H}, \hat{G}\right) = T_{\text{act}}(H, G) = (f_1(H), f_2(H) \odot G)$$

[0031] 其中，T_{act}表示等变激活函数类的通用表示，$\hat{H}$表示变换后的不变特征，G是变换后的等变特征，⊙表示逐元素乘法。f_1是对不变特征所采用的非线性激活函数；f_2利用不变特征计算缩放系数并对等变特征进行缩放。

[0032] 步骤303、利用等变特征更新模块提取智能体的空间信息，并通过更新后的等变非线性激活函数，对等变特征$G_i^{(l)}$进行缩放，生成最终的决策动作，同时确保等变性。

[0033] 针对智能体i第l层的不变特征$H_i^{(l)} \in R^V$和等变特征$G_i^{(l)} \in R^U$，通过下面公式更新下一层等变特征$G_i^{(l+1)}$：

[0034]
$$G_i^{(l+1)} = f_2\left(H_i^{(l)}\right) G_i^{(l)} + D \sum_{j \in comm_n(i)} \left(G_i^{(l)} - G_j^{(l)}\right) f_m\left(m_{ij}\right)$$

[0035] 其中，$\mathbf{G}_i^{(l+1)}$表示更新后的等变特征向量，$f_2\left(H_i^{(l)}\right)$表示经过不变特征计算的等变激活函数的输出，加权函数f_m采用多层感知机实现；$\mathbf{G}_i^{(l)} - \mathbf{G}_j^{(l)}$项感知智能体特征与邻居智能体特征的差异，D为加权系数。

[0036] 步骤五、在L层处理之后，利用最后一层的等变特征输出动作，作为智能体的决策结果。

[0037] 步骤六、将智能体决策网络应用于CTDE框架进行训练；

[0038] 在训练阶段，利用完全图信息，在所有智能体之间进行消息聚合和传递；

[0039] 而在分布式执行阶段，仅在每个智能体的邻居范围内进行消息聚合和传递。

[0040] 步骤七、将训练后的智能体决策网络应用于演员评论家架构的算法中；

[0041] 策略网络使用等变特征来控制动作，而值网络则使用不变特征来对状态价值进行估计。

[0042] 本发明的优点在于：

[0043] 1)、一种基于等变图神经网络的多智能体强化学习方法，利用图神经网络到多智能体强化学习领域，能有效建模多智能体之间的交互关系，利用消息传递和消息聚合完成多智能体强化学习任务。

[0044] 2)、一种基于等变图神经网络的多智能体强化学习方法，迁移等变图神经网络EGNN到多智能体强化学习领域作为策略网络和值网络，使得策略网络具有等变特性，值网络具有不变特性，从而提高样本效率，提升多智能体强化学习的收敛速度和收敛性能。

[0045] 3)、一种基于等变图神经网络的多智能体强化学习方法，设计了能保持等变特性的激活函数，能增强模型表征能力，提升模型性能。

附图说明

[0046] 图1为现有技术中改变智能体顺序排列不变性的示意图；

[0047] 图2为本发明一种基于等变图神经网络的多智能体强化学习方法的流程图；

[0048] 图3为本发明采用的等变图神经网络总体架构图；

[0049] 图4为本发明聚集任务对比实验图；

[0050] 图5为本发明导航任务对比实验图；

[0051] 图6为本发明追逐任务对比实验图。

具体实施方式

[0052] 为了便于本领域普通技术人员理解和实施本发明，下面结合附图和实施例对本发明作进一步的详细描述。显然，所描述的实施例仅仅是本发明部分的实施例，而不是全部的实施例，基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都应当属于本发明保护的范围。

[0053] 本发明涉及一种能保持等变特性的多智能体强化学习算法，特殊的图神经网络和多智能体强化学习；具体涉及一种利用具有等变特性的图神经网络EGNN作为多智能体强化学习的策略网络和值网络，从而对智能体嵌入先验知识以提升样本效率，加快多智能体强化学习训练和收敛的方法。

[0054] 所述基于等变图神经网络的多智能体强化学习算法，如图2所示，具体步骤如下：

[0055] 步骤一，搭建包括特征初始化模块和L层等变空间模块的等变图神经网络，作为智能体的决策网络；

[0056] 为了提高多智能体强化学习算法的性能，采用如图3所示的网络结构作为智能体的决策网络。该网络结构由两个模块组成:特征初始化模块和等变空间模块。

[0057] 特征初始化模块负责初始化智能体的等变特征和不变特征，并建立智能体之间的邻接关系，可以方便的建模和表示多智能体强化学习场景中的环境信息以及智能体之间的交互关系；等变空间模块由多层组成，逐层对等变特征和不变特征进行加工和处理。经过多层的处理后利用最后一层的等变特征输出动作作为智能的决策结果。

[0058] 经过多层的更新可以较为全面的处理和提取等变和不变信息；利用等变特征输出动作保留了动作的等变性，从而使策略网络的输出保留了等变特性，嵌入先验信息，能有效提高采样效率。

[0059] 步骤二、针对当前智能体i,利用通信距离建立与周围智能体间的邻接关系;

[0060] 在通信距离内的智能体视作相邻,通信矩阵用e表示。

[0061] 步骤三、特征初始化模块初始化该智能体的等变特征G和不变特征H;

[0062] 其中等变特征指会随地图旋转发生改变的特征,如场景中智能体的绝对坐标和绝对速度;不变特征指不随场景旋转改变的特征,包括智能体之间的距离和相对方位,速度的大小,编号等。

[0063] 步骤四、等变空间模块逐层对等变特征和不变特征进行加工和处理;

[0064] 智能体i的初始不变特征为$H_i^{(0)}$和初始等变特征为$G_i^{(0)}$;

[0065] 首先进行不变特征$H_i^{(l)}$的提取,更新智能体的不变特征$H_i^{(l)}$并从中提取智能体的深层不变信息;其次进行等变特征$G_i^{(l)}$的更新,提取深层几何信息并保持等变特性。经过多层的更新能有效提取和加工信息,便于智能体更好的理解当前的环境信息,从而更好的进行决策。

[0066] 不变特征更新模块被设计用于提取不变特征$H_i^{(l)}$,在多智能体任务中,由于存在复杂的交互过程,通过消息聚合和传递来增强相关等变特征$G_i^{(l)}$。不变信息更新模块能够高效捕捉智能体之间的不变关系,从而增强表征能力。

[0067] 在具体技术方案中,消息聚合和传递过程中除了需要不变特征$H_i^{(l)}$外,还应该补充智能体之间的等变特征$G_i^{(l)}$,以帮助智能体通过消息聚合和传递过程更全面地捕捉当前状态下的所有重要信息,更好地完成设定的任务。等变特征更新模块用于建模智能体之间的空间交互过程,通过多层更新逐层聚合和处理几何特征,更全面地理解当前状态下的几何结构和空间信息。

[0068] 具体为:

[0069] 步骤301、利用不变特征更新模块提取智能体的深层不变特征$H_i^{(l)}$,并进行更新。

[0070] 给定智能体i的不变特征$H_i^{(l)} \in R^V$和等变特征$G_i^{(l)} \in R^U$,通过下面公式更新下一层不变特征$H_i^{(l+1)}$:

[0071]
$$\mathbf{m}_{ij} = f_g\left(H_i^{(l)}, H_j^{(l)}, \| G_i^{(l)} - G_j^{(l)} \|^2, e_{ij}\right)$$

$$H_i^{(l+1)} = f_h\left(H_i^{(l)}, \sum_{j \in comm_n(i)} m_{ij}\right)$$

[0072] 其中m_{ij}表示智能体i和智能体j之间传递的消息向量,$||.||^2$表示二范数,(.,.,.)表示拼接操作符,函数$f_g(.)$和$f_h(.)$采用多层感知机实现,l表示第l层等变特征更新模块,comm_n(i)表示与智能体i在通信范围内的邻居集合,e_{ij}表示智能体i和智能体j之间的标量边特征。

[0073] 步骤302、将等变特征和更新后的不变特征,经过等变非线性激活函数更新该激活函数的系数,以增强模型的表征能力。

[0074] 这确保了整个等变空间处理模块具有等变特性和较强的表达能力,等变非线性激活函数的作用在于在保持等变性的同时增强网络的表征能力,其核心在于使用不变特征经过处理得到一个系数,然后对等变特征进行缩放,从而增强模型的表征能力。

[0075] 激活函数更新的公式如下:

[0076] $$\left(\widehat{H},\widehat{G}\right)=T_{\mathrm{act}}(H,G)=(f_1(H),f_2(H)\odot G)$$

[0077] 其中,T_{act}表示提出的等变激活函数类的通用表示,$\widehat{H}$表示变换后的不变特征,G是变换后的等变特征,⊙表示逐元素乘法。f_1是对不变特征所采用的非线性激活函数,不会改变模型等变性;f_2利用不变特征计算缩放系数并对等变特征进行缩放,也不会改变模型等变性。f_1和f_2均可选择常规非线性激活函数,如SiLU,Mish,tanh,ReLU等,在本专利的具体实现中,f_1选择SiLU函数,f_2选择tanh函数。

[0078] 步骤303、利用等变特征更新模块提取智能体的空间信息,并通过更新后的等变非线性激活函数,逐层聚合和传递来增强智能体之间相关的等变特征$G_i^{(l)}$,生成最终的决策动作,同时确保等变性。

[0079] 给定智能体i的不变特征$H_i^{(l)}\in R^V$和等变特征$G_i^{(l)}\in R^U$,通过下面公式更新下一层等变特征$G_i^{(l+1)}$:

[0080] $$G_i^{(l+1)}=f_2\left(H_i^{(l)}\right)G_i^{(l)}+D\sum_{j\in comm_n(i)}\left(G_i^{(l)}-G_j^{(l)}\right)f_m(m_{ij})$$

[0081] 其中,$\mathbf{G}_i^{(l+1)}$表示更新后的等变特征,第一项中$f_2\left(H_i^{(l)}\right)$表示经过不变特征计算的等变激活函数的输出,用于对$\mathbf{G}_i^{(l)}$进行缩放。第二项在与智能体i通信距离内的邻居集合comm_n(i)范围内进行消息传递,加权函数f_m采用多层感知机实现,用于对消息m_{ij}进行加权;$\mathbf{G}_i^{(l)}-\mathbf{G}_j^{(l)}$项感知智能体特征与邻居智能体特征的差异,通过消息传递机制对所有消息进行消息聚合,以在局部观测的设定下扩大通信范围。两项的加权系数D用于平衡两项的大小,帮助智能体捕获交互系统中周围环境的动态变化并感知全局信息。

[0082] 步骤五、在L层处理之后,利用最后一层的等变特征输出动作,作为智能体的决策结果。

[0083] 步骤六、将智能体决策网络应用于CTDE框架进行训练,并将训练后的智能体决策网络应用于演员评论家架构的算法中;

[0084] 智能体决策网络适用于MARL的中心式训练分布式执行(Centralized Training with Decentralized Execution,CTDE)框架,并能够方便地适用于全局或局部信息。在训练阶段,利用完全图信息,在所有智能体之间进行消息聚合和传递;而在分布式执行阶段,仅在每个智能体的邻居范围内进行消息聚合和传递。

[0085] 这种设计使得网络能够灵活地在全局信息和局部信息之间进行调整,高效地适用于MARL的各种训练范式。在具体的网络应用中,将上述网络结构运用于演员评论家架构的算法中。策略网络使用等变特征来控制动作,而值网络则使用不变特征来对状态价值进行估计。通过将演员和评论家应用于CTDE范式中,并利用MARL的损失函数进行训练。

[0086] 将本发明所提出的方法(用EGNN表示)在多智能体强化学习常见的三个大规模的集体运动任务中进行了实验,以验证有效性:1)聚集,其中智能体基于观察自主聚集在一起,没有预定义的目标;2)追逐,几个智能体试图追赶移动速度更快的智能体;以及3)导航,其中多个智能体朝向目标导航,同时避免彼此碰撞和与障碍物碰撞。

[0087] 本EGNN方法专注于利用多智能体系统中的对称性。在这一领域,相关的基线包括利用对称先验(ESP)和部分对称利用(PSE),在完全对称的情况下,PSE退化为ESP。将ESP作为比较的关键基线,因为它在多智能体系统中利用对称性的背景下具有相关性。此外,本方法与递归神经网络(RNN)、常用的多层感知器(MLP)和图神经网络GraphSAGE进行了全面比较。实验数据使用5个不同的随机种子来评估每个算法的性能。最后的实验结果表明,当MARL算法与本EGNN网络集成时,与原始版本相比,在收敛速度和收敛回报方面都有显著改进。EGNN通过嵌入对称性先验知识明显提高了算法在多智能体任务中的有效性和效率。

[0088] 在如图4所示的聚集任务中,本EGNN方法在收敛速度和最终收敛有效性方面明显优于其他算法。一个值得注意的观察结果是GraphSAGE的性能,通过考虑图结构,它在收敛速度方面比传统的MLP和RNN具有优势。然而,ESP相对于标准MLP的改进虽然存在,但并不特别明显。这表明ESP方法在受试者人数较多的情况下可能没有那么有效。这一发现表明,数据扩充方法在具有大量智能体的场景中是无效的。

[0089] 在如图5所示的导航任务中,本EGNN方法表现出了显著的进步,有效地将多个智能体引导到各自的目标,同时最大限度地减少碰撞。这种性能超过了传统的算法,如多层感知器(MLP)和递归神经网络(RNN),以及具有嵌入对称性的ESP方法。此外,与GraphSAGE相比,EGNN在收敛速度和最终结果方面都有显著改进。实验结果还表明,尽管ESP比标准MLP有一些改进,特别是在减少碰撞方面,但其在朝向目标导航方面的总体增强并不明显。这一观察结果表明,与聚集场景类似,ESP方法在更复杂的多智能体导航任务中可能面临限制。

[0090] 在如图6所示的追逐场景时,本发明利用Voronoi策略来制定躲避者的策略。包括训练特工追捕沃罗诺伊逃亡者。这里的转变是RNN的性能,它超过了MLP。这强调了将时间信息纳入算法的重要性,尤其是在追踪等动态场景中。相反,ESP的性能低于MLP,这表明智能体数量的增加会显著影响数据扩充策略。此外,虽然GraphSAGE比MLP和RNN实现了更快的收敛,但其最终收敛效果不如EGNN。虽然集成图结构有助于提高速度,但忽略内在对称性可能会影响收敛过程的整体有效性。本EGNN方法在更快的收敛和更有效的最终结果方面展示了其优势。

[0091] 实验细节说明

[0092] (1)场景动力学

[0093] 在实验中,智能体的动力学是通过控制x和y方向上的速度来决定的。具体地说,智能体的状态表示为$s=[x,y]\in S=\{[x,y]\in R^2: 0\leqslant x\leqslant x_{max}, 0\leqslant y\leqslant y_{max}\}$,其中$x_{max}$和$y_{max}$分别在x和y维度上划分环境的边界。智能体的控制输入是x和y方向上的位移,表示为△x和△y。系统的运动学模型由以下方程描述:

[0094]
$$\dot{x} = v_x$$
$$\dot{y} = v_y$$

[0095] 这里,$\dot{x}$和$\dot{y}$分别表示智能体在x和y方向上的位置变化率。

[0096] 每个智能体的动作空间允许控制其在x和y方向上的速度。智能体能够调整速度。智能体i的速度可以表征为：

[0097] (v_x, v_y)

[0098] 其中v_x和v_y分别是沿x轴和y轴的速度分量。

[0099] (2)任务详细信息

[0100] 介绍了三项任务的观察空间和奖励函数。智能体i的状态可以表示为$\tau_i = \left(g_i^{(0)}, h_i^{(0)}, \mathcal{E}\right)$，ε表示智能体之间的邻接关系。

[0101] (3)观测空间

[0102] 聚集任务的观测空间：智能体i的聚集任务观测空间可以表示为：

[0103] $g = \{(x_i, y_i, v_{x_i}, v_{y_i})$

[0104] $h = \{(d_{ij}, \sin(\theta_{ij}), \cos(\theta_{ij})) \mid j \neq i\}$

[0105] 其中，(x_i, y_i)表示智能体i的坐标，(v_{x_i}, v_{y_i})是智能体i的速度分量，$(d_{ij}, \sin(\theta_{ij}), \cos(\theta_{ij}))$分别是到另一个智能体j的距离以及相对角度的正弦和余弦。

[0106] 导航任务的观测空间：在导航任务中，每个智能体i都拥有一个观测空间，该空间包括其自身的状态和与其与指定目标点的关系相关的信息。

[0107] 具体地说，智能体i的观测可以表示为：

[0108] $g = \{(x_i, y_i, v_{x_i}, v_{y_i}, x_g, y_g)$

[0109] $h = \{(d_{ij}, \sin(\theta_{ij}), \cos(\theta_{ij})) \mid j \neq i\}$

[0110] 这里，x_g和y_g是智能体i沿x轴和y轴到其目标点的距离。h的部分与聚集任务观测空间相同。

[0111] 追逐任务的观测空间：在追逐任务中，每个追击者智能体i都有一个观察空间，该空间包括关于其自身状态以及相对于躲避者的相对位置和方向的信息。智能体i的观察结果可以表示为：

[0112] $g = \{(x_i, y_i, v_{x_i}, v_{y_i}, x_e, y_e)$

[0113] $h = \{(d_{ij}, \sin(\theta_{ij}), \cos(\theta_{ij})) \mid j \neq i\}$

[0114] 这里，x_e和y_e是智能体i沿着x轴和y轴到躲避点的距离。h的部分与聚集任务观测空间相同。

[0115] (4)奖励函数

[0116] 聚集任务的奖励函数：聚集任务的奖励功能旨在鼓励智能体尽量减少彼此之间的距离，同时惩罚高动作输出。其定义如下：

[0117] $$R(s,a) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} min\left(d^{i,j}, d_c\right) + \| \beta \|$$

[0118] 其中，$d^{i,j}$表示智能体间距离的集合。截断距离d_c设置为$\max(x_{max}, y_{max})$。β项$= -1 * 10^{-3}$用于惩罚高动作输出，促进智能体节约能量的行为。

[0119] 追逐任务的奖励函数：追逐任务的奖励函数是为了表达最小化躲避者和最近追捕者之间距离的目标而定制的。其公式为：

[0120] $$R(s,a)=-\frac{1}{d_o}min(d_{min},d_o)$$

[0121] 其中$d_{min}=\min(d^{1,e},\cdots,d^{N,e})$表示到最近的追赶者的距离。$d_o$表示到躲避者的最大可能距离。该公式确保了奖励函数激励追赶者减少与躲避者的距离。

[0122] 导航任务的奖励函数:导航任务的奖励函数旨在鼓励智能体靠近其指定的目标点,并避免与其他智能体发生碰撞。它可以公式化为:

[0123] $$R(s,a)=-\gamma\sum_{i=1}^{N}min(d_{ik},d_t)-\lambda I$$

[0124] d_{ik}表示智能体i与其目标点k的距离,d_t是阈值距离,鼓励智能体在接近其目标时精确定位自己。I是一个碰撞指示器,如果发生碰撞,则假设值为1,否则为0,以确保智能体因碰撞而受到惩罚。

[0125] (5)参数设置

[0126] 在场景中,超参数都使用了mappo中的默认参数。表1描述了MAPPO的超参数。当存在同质智能体时,使用参数共享,因为这在过去的大多数MARL工作中都是公认的做法。本EGNN方法也可以用于异构任务。计算实验是在带有i9-12900KF CPU的Linux服务器上进行的。使用的Python版本是3.9.12,使用的PyTorch版本是1.13.0。

[0127] 表1

[0128]

超参数	值
梯度裁剪范数	10.0
GAE 参数	0. 95
折扣率λ	0.99
价值损失	huber 损失
批量大小	缓冲池大小*智能体数量
最小批量尺寸	批量化大小/最小批量

[0129]

优化器	Adam
优化器小量	1e-5
权重衰减	0
网络初始化	正交初始化
使用奖励正则化	真
使用特征正则化	真

[0130] 以上所述,仅为本发明较佳的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明结论的技术范围内,根据本发明的技术方案及其发明构思加以等同替换或改变,都应涵盖在本发明的保护范围内。

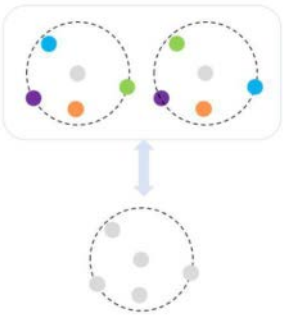


图1

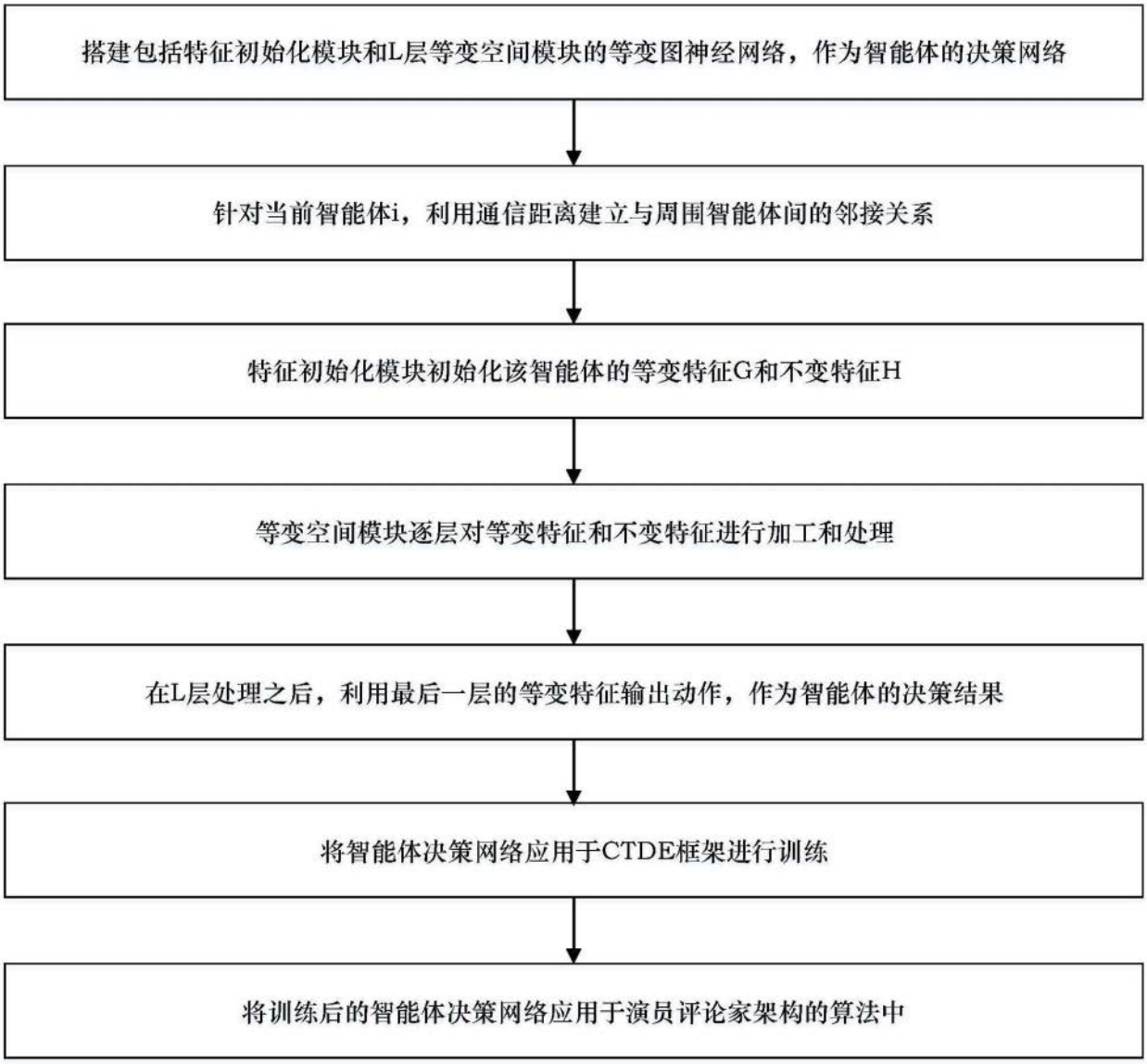

搭建包括特征初始化模块和L层等变空间模块的等变图神经网络，作为智能体的决策网络
针对当前智能体i，利用通信距离建立与周围智能体间的邻接关系
特征初始化模块初始化该智能体的等变特征G和不变特征H
等变空间模块逐层对等变特征和不变特征进行加工和处理
在L层处理之后，利用最后一层的等变特征输出动作，作为智能体的决策结果
将智能体决策网络应用于CTDE框架进行训练
将训练后的智能体决策网络应用于演员评论家架构的算法中

图2

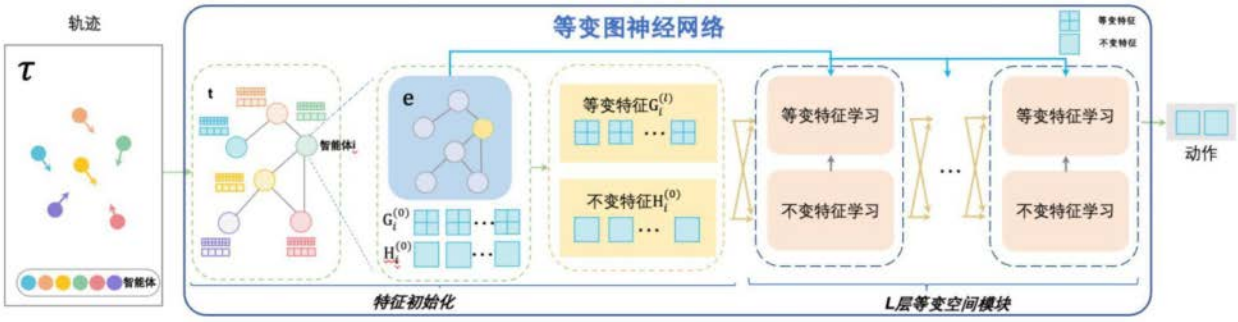

轨迹
τ
智能体
等变图神经网络
等变特征
不变特征
t
智能体i
e
等变特征$G_i^{(l)}$
不变特征$H_i^{(0)}$
$G_i^{(0)}$
$H_i^{(0)}$
等变特征学习
不变特征学习
等变特征学习
不变特征学习
动作
特征初始化
L层等变空间模块

图3

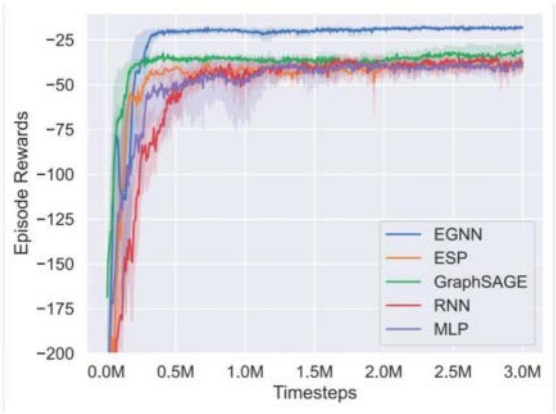

Episode Rewards
-25
-50
-75
-100
-125
-150
-175
-200
0.0M
0.5M
1.0M
1.5M
2.0M
2.5M
3.0M
Timesteps
EGNN
ESP
GraphSAGE
RNN
MLP

图4

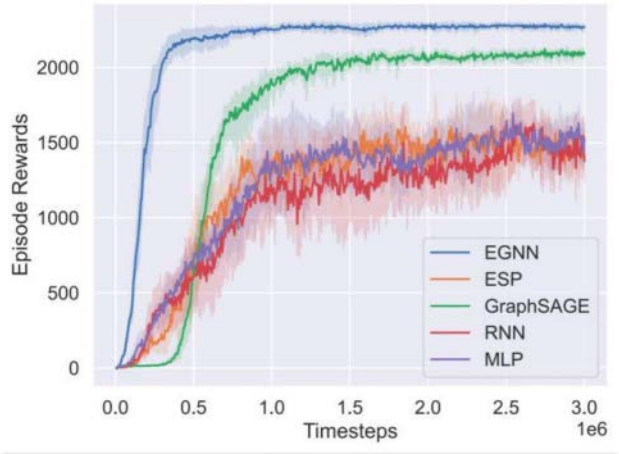

Episode Rewards
2000
1500
1000
500
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1e6
Timesteps
EGNN
ESP
GraphSAGE
RNN
MLP

图5

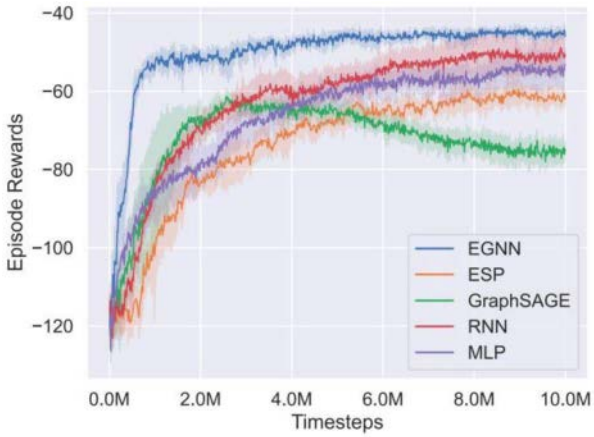

-40
-60
-80
-100
-120
Episode Rewards
0.0M
2.0M
4.0M
6.0M
8.0M
10.0M
Timesteps
EGNN
ESP
GraphSAGE
RNN
MLP

图6