基于 DPES Dueling DQN 的路径规划方法研究

武 曲 张 义 郭 坤 王 玺
(青岛理工大学信息与控制工程学院 山东 青岛 266520)

摘 要 针对传统的路径规划算法难以处理复杂环境等问题,提出分布式优先级经验置换的深度强化学习优化方法,实现复杂环境的路径规划。以样本训练时产生的 TD error 为节点权重构建小根堆的记忆库,不断地将低优先级的样本替换出记忆库保证样本的训练价值;基于堆层数进行优先级采样,解决训练被某些异常高优先级样本所诱导的问题;采用分布式的方式加速训练过程;通过进行路径规划的仿真实验证明了该方法的有效性和可行性。

关键词 优先级经验替换 深度 Q 网络 堆 深度强化学习 路径规划

中图分类号 TP391 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2023.06.023

PATH PLANNING METHOD BASED ON DPES DUELING DQN

Wu Qu Zhang Yi Guo Kun Wang Xi
(*School of Information and Control Engineering, Qingdao University of Technology, Qingdao 266520, Shandong, China*)

Abstract Aimed at the problem that the traditional path planning algorithms are difficult to deal with complex environment, a distributed priority experience substitution deep reinforcement learning optimization method is proposed to realize path planning in complex environments. The TD error generated during sample training was used as the node weight to construct the memory store of the small root heap, and the low-priority samples were constantly replaced by the memory store to ensure the training value of the samples. Priority sampling was carried out based on the layers of heaps to solve the problem that training was induced by some abnormally high priority samples. The distributed method was adopted to speed up the training process. The validity and feasibility of the proposed method were proved by the simulation of path planning.

Keywords Priority experience substitution DQN Heap Deep reinforcement learning Path planning

0 引 言

在路径规划领域,目前已经存在很多经典的算法。如迪杰斯特拉提出的 Dijkstra 算法,应用贪心的思想,通过每次在未标记的节点中选择距离源点最近的节点实现最短路径的求解[1]。在已知地图的情况下,这种算法仍可以取得很好的效果。A* 算法在 Dijkstra 算法的基础上,加入了启发式函数[2],也就是一种评估当前点到达目标的度量,用来决定下一步应该优先扩展哪个节点,这种算法在多维度规划问题上,或是在较大规模的地图上,算法复杂度很大。势场法将规划空间看作物理学中“场”的概念,将智能体看作一个粒子,障碍物会对这个粒子产生斥力,目标会对这个粒子产生引力,两者的合力即为最后智能体运动的方向[3],势场法成功的关键在于如何设计引力和斥力函数。这种方法实时性较好,同时产生的路径通常十分平滑,适合于机械臂一类的应用,缺点是在合力为 0 的位置智能体容易陷入局部最优解。

近年来,由于人工智能的兴起,很多基于人工智能的路径规划方法被提出,文献[4]提出了一种基于模糊逻辑的移动机器人路径规划算法,将状态空间与动

收稿日期:2020-09-01。山东省自然科学基金项目(ZR2017BF043)。**武曲**,副教授,主研领域:强化学习,深度强化学习。**张义**,硕士生。**郭坤**,硕士生。**王玺**,硕士生。

作空间关联起来,形成映射关系,解决了人工势场法中容易陷入局部极小的问题;文献[5]中详细介绍了遗传算法在路径规划中的研究,并提出了一种基于改进染色体编码的自适应遗传算法,使得算法能够避免过早收敛的问题;文献[6]中提出了利用双向神经网络来解决在未知环境中进行路径规划的方法。

尽管上述方法可以在各自的领域取得不错的效果,但是它们都基于已知环境这个前提,需要人工将环境与路径规划算法结合,在实际应用时具有一定的局限性。强化学习是一类应用在未知环境的机器学习方法[7],作为机器学习的三大分支之一,不同于监督学习和无监督学习,强化学习无须提供数据,所有的学习资料都将从环境中获取。智能体通过不断地探索环境,根据不同的动作产生的不同的反馈进行模型的学习,最终智能体将能以最优策略在指定环境中完成任务。

在利用强化学习进行路径规划问题方面,也已经出现了一些研究成果,文献[8]提出利用偏好评估的强化学习技术,结合降维的方法,实现了智能体在存在移动障碍物的环境中的路径规划;文献[9]将深度强化学习技术与策略梯度法结合起来,解决自动驾驶中的路径规划问题,提升了路径规划问题的效率;文献[10]将监督学习与强化学习相结合,为智能体提供规划好的路径,接下来智能体利用强化学习中函数近似的方法来进行泛化,实现在其他环境中的路径规划,具有较强的泛化能力。

将深度学习与强化结合的结合上,Mnih 等[11]构建的 DQN(Deep Q Network)无疑是一项重要的研究成果,通过经验重放的 off-policy 方式,解决了强化学习领域的数据之间的强相关性无法在深度学习算法中取得好的效果的问题。可以说,经验重放是深度学习与强化学习结合的关键所在,一些学者就此过程进行研究。Schaul 等[12]提出了一种基于优先级经验重放的(Prioritized Experience Replay,PER)的采样方式,记忆库中的数据按照被利用来进行训练时的 TD error 计算其优先级,造成的 TD error 比较大的样本说明模型对此类样本还未能很好地收敛,在再次采样时应该更多地选择此类样本,反之,造成 TD error 小的样本应该尽量少地被再次采样。这种方式被证明优于随机采样的方式,在多种 Atari 中的表现优于随机采样。陈希亮等[13]提出一种基于重抽样优选缓存经验回放的抽样机制,解决了 PER 抽样方式导致的抽样不充分的问题。何明等[14]提出一种 PES(Prioritized Experience Selected)的采样方式,根据 TD error 排序序数的倒序为样本设定优先级,解决了 PER 过程中 TD error 量级间隔过大而导致的多数样本因采样概率低而无法被采集到的问题,在 MADDPG 算法中取得了比 PER 采样过程更好的效果。

以上抽样方法都着重于从记忆库中采样的方式,在离线学习的深度强化学习的机制中,需要设定一个记忆库来存储与环境的交互数据,通常,这个数据库被设计为固定规格,当记忆库存满后,采用先进先出的替换原则用新的数据把当前记忆库中最先存入的数据替换出去,以此来为模型提供较新的数据。这种抽样的方式带来的问题是对于早期存入的数据,即使有高的优先级,也有可能被替换出去,而排在末尾的低优先级的数据,也会因为有更长的存在时间而可能被抽样。本文采用了一种基于堆结构的优先级经验置换策略,在无须严格排序的基础上实现了样本替换优先级的定义与运用,保证了记忆库中样本的高可用性。此外为了解决训练过程被某些异常高 loss 样本所诱导产生的训练不平稳的问题,本文提出使用基于层序数的优先级采样进行解决。

另外,在多智能体联动学习方面,OpenAI 团队在其提出的 MADDPG[15]算法中使用了集中式学习、分布式执行的框架,不同进程之间共享最新的参数,可以使模型更快地收敛。

Dueling DQN[16]是在 DQN 算法上的一种改进,该算法将 Q 值分为 Value 和 Advantage 两部分,经本文验证 Dueling DQN 在复杂长回合问题中具有更好的表现。

综上,本文提出一种分布式优先级经验置换 Dueling DQN(DPES Dueling DQN)的算法结构,并在较大规模的复杂环境中进行路径规划的仿真实验,验证了本文算法的可行性和高效性。

1 相关理论

1.1 DQN

Q-Learning 算法是强化学习中一种经典的基于值的算法[17],该算法维护一个状态与动作的 Q 值表格,在每一个状态下,都可以通过查询表格的方式获得各个动作所对应的 Q 值,其中一个表项值 Q_{ij} 表示在状态 s_i 下选择动作 a_j 的行为价值,通常依 $\mathrm{argmax}_a(Q_{ij})$ 的策略进行动作选择。在动作执行之后,根据从环境中获得回报 r 按式(1)对当前 $<s,a>$ 值对应的 Q 值进行更新。

$$Q(s,a)\leftarrow Q(s,a)+\gamma[r+\max_{a'}Q(s',a')-Q(s,a)] \tag{1}$$

循环该过程直至整个 Q 值表收敛。式中:γ 表示衰减

度,用来表达一个回合中较后的动作所产生的回报对较前的动作选择的影响。

Q-Learning 算法可以近乎完美地解决低维简单的强化学习问题,但是在处理多状态多动作的复杂问题时,Q-Learning 算法就会变得力不从心,复杂的状态空间和动作空间让 Q 值表变得非常巨大,两相组合更是使得 Q 值表的数量级呈指数型增长,这就导致 Q 值表的收敛变得异常困难。另外对于未参与训练的状态,Q-Learning 算法将无法为其生成动作,也就是说 Q-Learning 算法没有泛化能力。

上述限制使得强化学习在很长一段时间没有出现突破性的研究进展,一直到 2013 年,DeepMind 团队的 Mnih 等提出了 DQN 算法,这标志着 DRL(Deep Reinforcement Learning) 时代的到来,自此不断涌现出许多 DRL 的相关技术。

DQN 由两个结构相同但参数间隔更新的网络构成,可以分别定义为 Q_{target} 和 Q_{eval},其中 Q_{eval} 从记忆库中提取数据进行学习,其参数实时更新,而 Q_{target} 每隔一定步数之后同步 Q_{eval} 的参数,通过如式(2)所示的 l_{oss} 值来进行 Q_{eval} 网络的学习。

$$l_{\text{oss}} = Q_{\text{eval}}(s,a) - (r + \gamma \max_{\text{a}} Q_{\text{target}}(\text{s}',\text{a}')) \quad (2)$$

深度学习的使用通常以训练数据相互之间互不相关为前提,而在强化学习中,一个回合的前后动作之间往往存在着很强的相关性,这就为深度学习的使用带了困扰。在 DQN 中,通过离线学习的方式解决了这个问题。DQN 引入了记忆库的概念,模型会将训练过程中的所有实时产生的 $<s,a,s',r>$ 元组保存在记忆库中,并不立即用来进行模型的学习,而是通过在记忆库中随机抽样的方式选择数据进行网络的学习。这样就有效地减弱了数据之间的相关性,使得训练好的模型能够具有泛化性。

1.2　Dueling DQN

Dueling DQN 是 DQN 的一种改进,Dueling DQN 将 Q 值分成了价值函数 Value 和优势函数 Advantage 两部分,其中:Value 表示当前状态的重要程度;Advantage 则对应每个动作各有一个值代表每个动作的优势,而后通过式(3)构造最终的 Q 值。

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + (A(s,a;\theta,\alpha) - \max_{a' \in |A|} A(s,a';\theta,\alpha) \quad (3)$$

式中:θ 表示网络卷积层的参数;α 和 β 分别表示 Advantage 和 Value 函数全连接层的参数。

本文实验证明,Dueling DQN 的这种设计有利于长回合场景下的动作选择,在复杂环境的路径规划应用中有较好的表现。

2　本文算法实现

2.1　分布式执行框架

本文算法采用一种分布式执行的框架,框架结构如图 1 所示。通过多线程的方式构建多个智能体,多个智能体各自独立地进行动作选择、动作执行,在获得回报后将数据样本存入共享的记忆库。

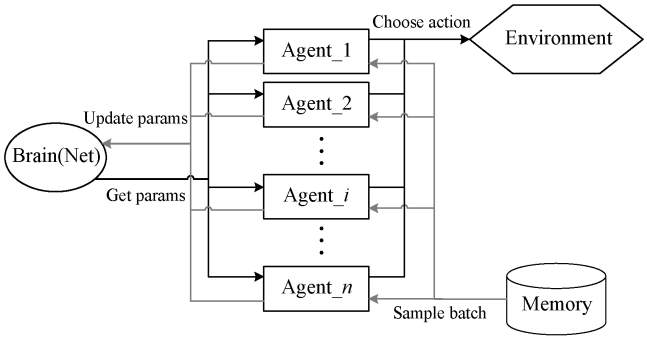

图 1　DPES Dueling DQN 结构

智能体在执行时,首先加载最新的共享全局参数,再进行动作选择。智能体在学习时,各自独立地从记忆库中进行样本抽取,对参数进行梯度更新后将智能体参数上传到全局共享参数,以保证全局参数获得实时更新,通过多智能体分布式处理的方式,可以进一步降低样本之间的相关性,减少模型收敛耗费的时间。

2.2　基于小根堆的 PES 策略

在传统的离线学习模型中,当记忆库存满时,模型便采用先进先出的机制,从索引 0 开始替换掉最先存入的数据,这样会导致高采样优先级的数据样本可能因为位于记忆库开始而被替换出去,造成有价值数据被丢弃。

本文提出的 PES 策略采用小根堆的结构实现。堆是二叉树的一种,不同于排序二叉树在节点增删时需要调整树的结构来保证树的平衡,堆结构在增删节点的同时自动保证了自身的平衡性,也就保证了在插入删除时的平均复杂度。堆有小根堆和大根堆之分,小根堆中的根节点是整个树中的最小节点,其子树中的根节点同样满足此性质。大根堆中则是根节点为最大节点。

小根堆在移动节点时的上浮和下沉两种操作定义如下:

上浮　若当前节点权重比父节点权重值小,交换当前节点与父节点的位置。

下沉　若当前节点权重比左子节点权重或右子节点权重大,交换当前节点与较小的子节点位置。

记忆库处在运行态时,在不同情境下的处理方式

如下：

1）新数据插入。

（1）堆节点数未达到上限。为新节点赋予初始权重 1，将节点插入到堆末尾。

（2）堆节点数达到上限时。替换掉堆根节点，为新节点赋予初始权重等同堆尾节点的权重，对节点进行下沉操作，移动节点到合适位置。

2）权重更新后的节点移动。改变被抽样的节点权重，根据新的权重值大小决定节点上浮或是下沉，移动节点到合适位置。

本文利用数据样本参与训练时产生的 TD-error 即损失值作为构建堆的权重，l_{oss} 越大，说明模型对此样本尚不能很好地拟合，需要保留在记忆库中继续训练。反之，l_{oss} 越小，说明模型对该样本拟合得很好，此样本就应该从记忆库中替换出去，避免该类样本过多地参与训练使模型陷入局部最优。而通过小根堆的方式，l_{oss} 最小的样本将会始终被放置在根节点的位置，可以以 $O(1)$ 的时间复杂度拿到并完成样本替换。而之所以在新样本存入初始化其权重与堆尾节点相同，是为了保证新写入数据的采样优先级。

2.3 基于堆层序数的优先级采样

在采样环节，本文提出一种基于堆层序数的优先级采样方法。在 Schaul 等提出的 PER 中，根据式(4)构建的优先级进行采样。

$$P(i) = \frac{p_i^{\alpha}}{\sum_k p_k^{\alpha}} \tag{4}$$

式中：p_i 表示需要为 i 的样本参与训练时产生的损失值。本文通过实验发现，模型在训练前期所产生的误差之间差异较大，依此方式产生的样本抽样概率将会更加悬殊，这并不利于模型的平滑收敛。

本文提出的基于小根堆层序数的优先级采样方式减弱了这种现象的影响。基于小根堆层序数的优先级采样并不严格依赖损失值的完整排序。在小根堆中的数据并不像二叉排序树那样满足严格的排序关系，堆中的层级间满足如下偏序关系：

$$L_i \leqslant L_j \quad i < j \tag{5}$$

式中：L_i 表是第 i 层的数据。依照此偏序关系构建每层的采样优先级，既可以保证高损失的样本具有较高的采样优先级，又不至于采样被限制在某些异常高的 loss 值上。基于层序数的优先级采样具体实现方式为首先令 $p_i = i, i = 1, 2, \cdots, \lceil \log_2(n+1) \rceil$，其中：$i$ 为堆的层序数；n 为堆中节点的总个数。将序列 p 代入式(4)中获得堆中各层的采样概率。在选中抽样层后，层内采用随机抽样的方式进行采样。本文方法的优势在于实现代价低，无须对序列进行排序，且能保证按优先级进行采样。此外，本文方法的采样效率比较高，可以直接通过索引定位数据，时间复杂度为 $O(1)$，相较于 PES 中通过 SumTree 的 $O(\log_2 n)$ 的时间复杂度有了较大的提升。

2.4 模型核心网络

DPES Dueling DQN 的网络如图 2 所示，其中包含三个全连接的隐藏层，每层设置 300 个节点，以 ReLU 作为激活函数。第 4 层采用 Dueling 的设计方式，分为 Value 和 Advantage 两部分，输出层即为 Q 值，由第四层中的两部分相加而得。

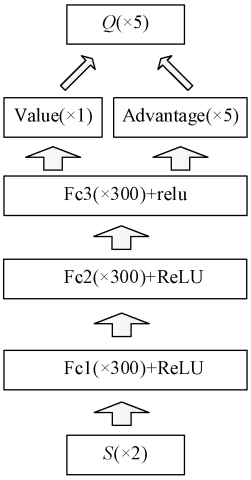

图 2 DPES Dueling DQN 核心网络结构

2.5 DPES Dueling DQN 算法步骤

DPES Dueling DQN 的算法伪代码如算法 1 所示。

算法 1 DPES Dueling DQN 算法

Initialize Agent_Ps, Heap, Learn_point, Global_θ

To every Agent_P:

Repeat max_loop:

while True:

load θ from Global_θ

with probability ϵ select a random action a_i otherwise select $a_i = \text{argmax}_a Q(\phi(s_i), a; \theta)$

take action a_i, return r, s'

replace the root element in heap with the new data $< s_t, a_t, s_{t+1}, r_t >$ and sink it

if Counter_Memory > Memory_size:

if Counter_Learn% Replace_point:

update θ from Q_{eval} to Q_{target}

end if

sample Layer_i from Heap with $j \sim P(j) = p_j^{\alpha} / \sum_i p_i^{\alpha}$ and sample random from ith layer

do model_P learn

update θ to global_θ

end if

if $s' \in S_{\text{target}}$ or $s' \in S_{\text{danger}}$:
end while
end if

3　强化学习环境搭建

本文的环境借助 OpenAI 团队构筑的 gym 环境框架搭建而成,环境以某建筑其中一层平面构建模拟环境,其可视化效果如图 3 所示。

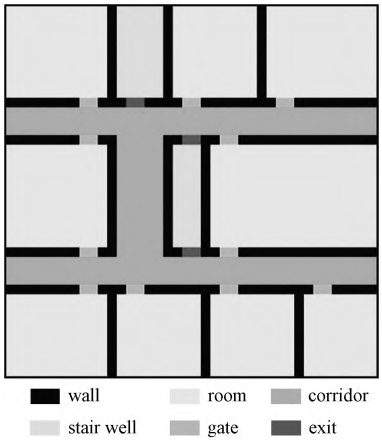

图 3　环境仿真

模拟环境由 40 ×40 的格点区域组成,仿真地图区域主要包含房间、楼道、楼梯井三部分。为验证算法处理复杂环境的能力,本文在实验时除了普通障碍(即环境中的“wall”)外另添加了一种“危险区域”(即环境中的“danger”)。加入该场景后的环境如图 4 所示。

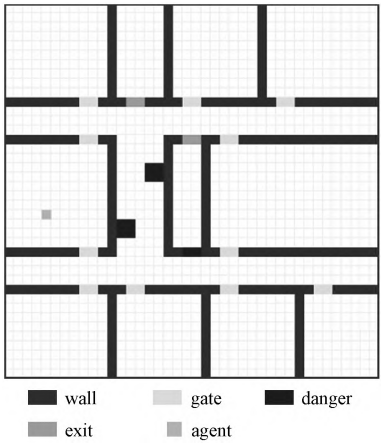

图 4　发生险情的环境仿真

假定发生险情后,智能体可能会分布在地图中的任意位置,要求强化学习模型可以为智能体规划最短最安全的逃生路径。智能体到达安全出口(图中的 exit 区域)视为逃生路径规划成功。在地图中间区域附近出现两处危险点(图中的 danger 区域),若智能体不慎步入其中,将随即死亡,本回合路径规划失败。另外,环境中原本的一处安全出口因为险情无法通过,也变为危险区域。

3.1　状态空间构建

环境中的状态由网格点的二维坐标表示,状态空间为平面中智能体所有可能处于的位置。即去除墙体和楼梯井之外的所有网格点。

3.2　动作空间构建

本文设定的动作空间为离散空间,包括 5 个动作,分别为原地不动、上、下、左、右,分别以整型数 0、1、2、3、4 表示。

3.3　环境回报构建

强化学习主要依赖环境的回报优化动作选择策略以完成任务,所以环境的回报对于任务的成功与否具有决定作用,本文基于先验知识和实验经验进行了下述的回报设定。

(1) 单步回报。因为环境中发生了险情,对于智能体来说,每多走一步,就会增加一分危险,因此设定 $r_{\text{step}} = -1$。这样的设定也会使得智能体会选择多条可行路径中最短的一条路径。

(2) 越界、碰壁回报。如果智能体在墙体边缘选择了“撞墙”的动作,这是一步无意义的动作,因此应当为此类动作设定一个负值回报 $r_{\text{wall}} = -1$。

(3) 险地回报。智能体踏入险地即死亡,回合结束,因此险地的回报应该为全局最小值。同时为了保证智能体能通过险地之间的过道,险地的设定值不应该太小,经过多次实验,最终设定 $r_{\text{danger}} = -50$。

(4) 安全出口回报。安全出口处是路径规划任务的最终目标,因此应给予全局最大的正值回报。安全出口的回报应该能保证即使长路程的规划路径回合的总回报大于短路程的死亡回合的总回报,在本实验中,设定其回报为 $r_{\text{target}} = 200$。

综上,智能体获得的回报定义如式(6)所示。

$$R_{\text{eward}} = \begin{cases} r_{\text{target}} & s \in S_{\text{target}} \\ r_{\text{danger}} & s \in S_{\text{danger}} \\ r_{\text{step}} + r_{\text{wall}} & s' \in S_{\text{wall}} \\ r_{\text{step}} & \text{其他} \end{cases} \tag{6}$$

4　仿真实验

4.1　实验参数设置

对于模型的核心网络,设计的层数、节点越少,则网络无法完成对复杂环境的全局收敛;设计的层数、节点过多,则可能会产生过拟合,且十分耗费计算资源。经过多次实验测试,最终设定网络结构如图 2 所示,为 3 ×300 节点的全连接层,以 ReLU 作为激活函数。设

定学习率为 10^{-4}，采用批量梯度下降的方式进行学习，设定 batch_size 为 256，Q_{target} 每 2 000 步与 Q_{eval} 同步参数。

此外，为了处理探索与利用的矛盾问题，采用动态 ϵ 的机制处理训练过程，设定初始 $\epsilon_0=1$，而后按照每学习一次 10^{-6} 的步进逐渐减小 ϵ，并设置下限 0.1，即 $\epsilon=\max(\epsilon-10^{-6}t,0.1)$，其中 t 为学习的步数。

设定记忆库的规模 memory_size 为 50 000，记忆库中存储数据到达 10 000 条时开始进行模型的学习。

在强化学习部分，ϵ 采用动态设计，设定初值 $\epsilon_0=0.1$，在模型开始学习后以 10^{-6} 的步进开始增加，至达到上限 0.9 时截止。设定衰减率 $\gamma=0.99$。

软件环境为 Ubuntu18.04，内存 24 GB，显卡为 GTX1060，显存 6 GB，采用 Pytorch 的深度学习框架。

4.2 实验结果分析

为了验证本文方法，还同时进行与 DQN、Dueling DQN、DPER Dueling DQN 算法的对比实验，表 1 对比 DQN、Dueling DQN、DPER Dueling DQN、DPES Dueling DQN 在各自在训练的不同阶段的模型效果。

表 1 测试效果对比表

模型	回合							
	2×10^4		5×10^4		1×10^5		2×10^5	
	完成率/%	回报值	完成率/%	回报值	完成率/%	回报值	完成率/%	回报值
DQN	52	14.2	61	35.7	73	114.2	79	109.8
Dueling DQN	73	113.4	80	122.9	93	154.4	95	160.1
DPER Dueling DQN	79	124.6	95	159.6	99	167.3	100	168.5
DPES Dueling DQN	87	132.4	99	168.2	100	168.5	100	168.3

表 1 中的完成率指标是加载当时的训练模型进行 100 次随机初始起点的模拟逃生路径规划成功次数所占的比例。其中的回报值为这 100 次路径规划的回合回报的均值，为了避免智能体在环境中徘徊，设置单回合最大步数为 200，超过此限制则认为路径规划任务失败。

从表 1 的数据可以看出保证了采样空间中的数据的高价值性、并通过优先级进行数据采样的 PES 策略的表现最佳，可以使训练产生的模型具有更好的全局可用性。

用三种算法分别训练 200 000 个回合，得到损失变化如图 5 所示。

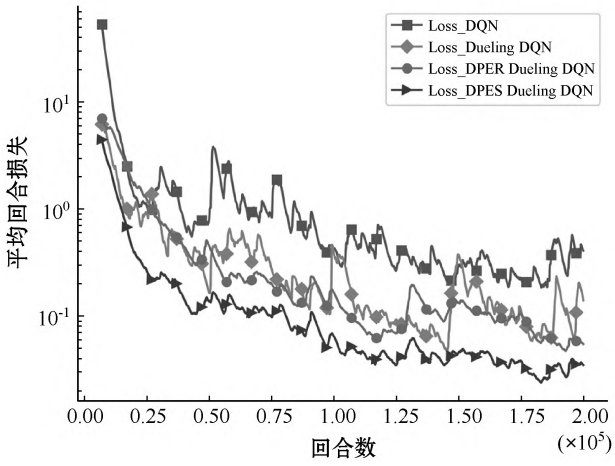

图 5 loss 变化

可以看出，Dueling 的结构在较大规模复杂环境中有较好的表现，体现在训练之初，通过代表状态重要性的 Value 值更能确定准确的方向选择，体现在 loss 图中为在训练开始时可以更快地向着拟合模型，降低 loss。而 PER 和 PES 策略能在处理训练后期尚未收敛的个别数据时发挥作用，PER 策略可以提高这些个别数据的抽样优先级，而本文的 PES 策略在保证其优先级的同时，又能确保这些数据能够保持在记忆库中不被替换出去，在 loss 图中可以看出 PER 策略几乎把 loss 降低了一个量级，而本文的 PES 策略又进一步降低了 loss。而且本文方法得到的曲线更加平滑，这也印证了本文方法不会被个别异常数据所左右的观点。结合表 1 的数据也可以看出本文提出的 PES 策略具有更好的全局收敛效果。

图 6 所示为随着训练进行，平均回合回报的变化趋势，通过对比平均回合回报的变化趋势也可以得到与上文同样的结论。本文方法可以更快地完成收敛，Dueling 的结构可以更好地帮助智能体找寻前进方向，能尽快地完成收敛。PES 的采样方式则可以使模型尽快适应某些尚未收敛的格点，更快地达到在全局任意位置都能安全逃生的路径规划效果。

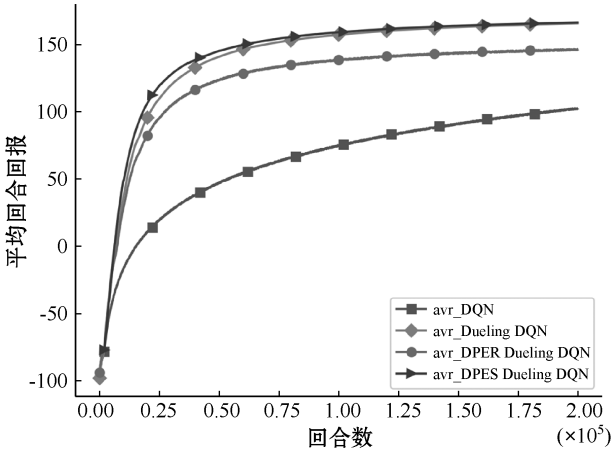

图 6 平均回合回报

训练到 20 000 轮的参数进行测试效果如图 7 所示,其中每个格点上的小三角形指示了智能体位于该位置时应该选择的动作方向。可以看到对于地图上的绝大部分区域,智能体都能找到安全逃生路径。

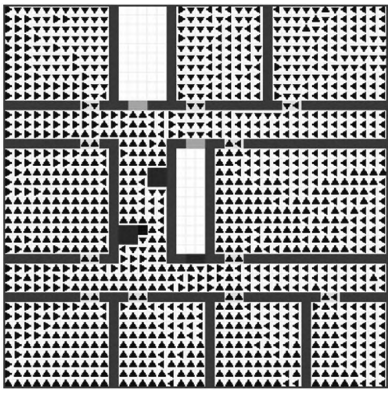

图 7　第 20 000 轮的路径规划效果

取第 110 000 轮的参数进行测试,效果如图 8 所示,此时无论在地图上的任意位置,智能体都能完成安全逃生的路径规划。

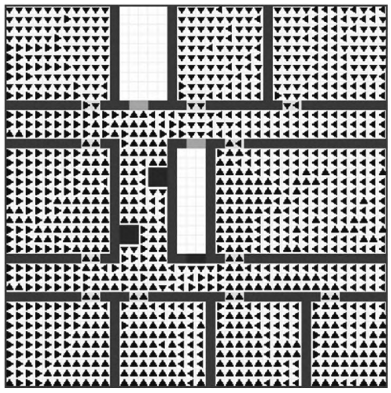

图 8　第 110 000 轮的路径规划效果

另外,本文还利用 DPES Dueling DQN 算法进行如图 9 和图 10 场景下的测试。图 9 是在未发生险情的安全路径规划场景,可以看到在地图中的任何位置,智能体都能按照指示方向到达安全出口,且所选路径为最短路径。图 10 所示环境中,一处险情阻塞了主要通道,可以看到模型在进行路径规划时会选择穿过其他房间到达安全出口。

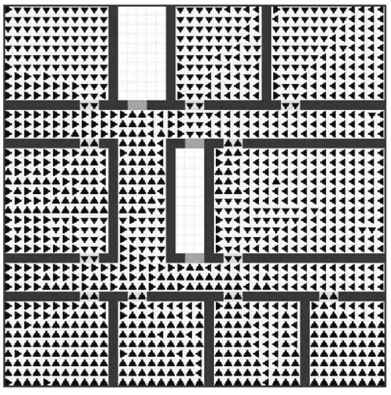

图 9　无险情发生时路径规划效果

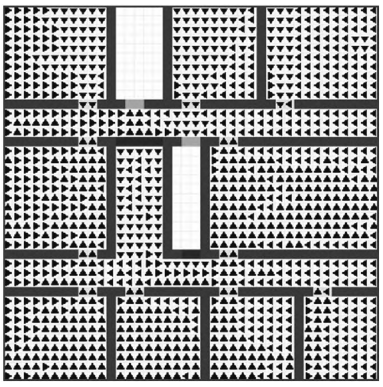

图 10　险情阻塞主干道路径规划效果

综合上述的实验可以看出,本文提出的 PES 策略可以在深度强化学习算法的训练过程中取得较好的加速表现,记忆库中样本质量的提高有助于模型更快、更稳定地收敛。此外,结合 Dueling DQN 提出的 DPES Dueling DQN 算法应用在路径规划场景中很好地完成了路径规划任务,通过不同实验场景的训练,本文算法的泛化性也得到了证明。

5　结　语

本文将深度强化学习应用在路径规划领域,提出使用 DPES Dueling DQN 算法进行复杂环境下的路径规划。采用 PES 策略将欠拟合的样本数据保留在记忆库中,使记忆库中的样本对于模型的全局收敛而言是高收益的。采用分布式的方式既有利于收集全局样本,也提高了模型收敛的速度及学习效率。又结合了在较大环境中表现更佳 Dueling DQN 算法进行最优路径规划。最终通过实验与 DQN、Dueling DQN、DPER Dueling DQN 进行对比,验证了 DPES Dueling DQN 方法进行路径规划的高效性和泛化能力。

参　考　文　献

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

[2] Yang D, Xu B, Rao K, et al. Passive infrared(PIR)-based indoor position tracking for smart homes using accessibility maps and a-star algorithm[J]. Sensors,2018,18(2):332.

[3] Orozco-Rosas U, Montiel O, Sepúlveda R. Mobile robot path planning using membrane evolutionary artificial potential field[J]. Applied Soft Computing,2019,77:236 – 251.

[4] 毕盛,朱金辉,闵华清,等. 基于模糊逻辑的机器人路径规划[J]. 机电产品开发与创新,2006,19(1):21 – 22.

[5] 刘传领. 基于势场法和遗传算法的机器人路径规划技术研究[D]. 南京:南京理工大学,2012.

(下转第 233 页)

提升算法效率。

4 结　语

本文提出一种基于实值 Root-MUSIC 算法与 MP 算法结合的间谐波检测方法。该方法利用 MUSIC 算法不受非同步采样影响的特性,精确估计出信号所含间谐波的频率,进而通过 MP 算法估计信号的幅值与相位参数。通过构建间谐波信号模型对算法进行仿真,由仿真结果以及与其他文献的算法进行比较可以得出:本文算法在检测间谐波信号时有着很好的精确度与抗噪能力,与传统的 MUSIC 检测方法相比,在同样的条件下有着更高的检测精度。

参 考 文 献

[1] 郝江涛,刘念,幸晋渝,等. 电力系统间谐波分析[J]. 电力自动化设备,2004,24(12):36 – 39.

[2] 张伏生,耿中行,葛耀中. 电力系统谐波分析的高精度 FFT 算法[J]. 中国电机工程学报,1999,19(3):63 – 66.

[3] 赵航宇,毛王清. 基于 FFT 的电力系统谐波分析[J]. 电工技术,2018,475(13):5 – 8.

[4] 郝柱,顾伟,褚建新,等. 基于四谱线插值 FFT 的电网谐波检测方法[J]. 电力系统保护与控制,2014,42(19):107 – 113.

[5] 许鸿飞,张姣姣,庞思睿,等. 基于汉宁双窗 apFFT 单谱线插值的电力谐波和间谐波检测算法[J]. 电测与仪表,2017,54(10):87 – 93.

[6] 丁铭,陈红卫. 基于 APFFT 和快速 TLS-ESPRIT 的间谐波检测方法[J]. 电测与仪表,2019,56(17):121 – 127.

[7] 李圣清,王飞刚,朱晓青. 基于改进型小波神经网络的谐波检测方法[J]. 电测与仪表,2019,56(10):118 – 121.

[8] Lobos T, Leonowicz Z, Rezmer J, et al. High-resolution spectrum-estimation methods for signal analysis in power systems[J]. IEEE Transactions on Instrumentation and Measurement,2006,55(1):219 – 225.

[9] Rao B D, Hari K V S. Performance analysis of Root-MUSIC [J]. IEEE Transactions on Acoustics, Speech, and Signal Processing,1989,37(12):1939 – 1949.

[10] Zoltowski M D, Kautz G M, Silverstein S D. Simultaneous sector processing via Root-MUSIC for large sensor arrays [C]//5th ASSP Workshop on Spectrum Estimation and Modeling,1990:372 – 376.

[11] 贾清泉,姚蕊,王宁,等. 一种应用原子分解和加窗频移算法分析频率相近谐波/间谐波的方法[J]. 中国电机工程学报,2014,34(27):4605 – 4612.

[12] 李惠章,李承,王臻. 基于混沌与 Pseudo-Newton 法组合优化的直接衰减正弦原子库分解方法在低频振荡分析中的应用[J]. 中国电机工程学报,2018,38(1):148 – 157,351.

[13] 蔡涛,段善旭,刘方锐. 基于实值 MUSIC 算法的电力谐波分析方法[J]. 电工技术学报,2009,24(12):149 – 155.

[14] 石敏,吴正国,尹为民. 基于多信号分类法和普罗尼法的间谐波参数估计[J]. 电网技术,2005,29(15):81 – 84.

[15] 高培生,谷湘文,吴为麟. 基于求根多重信号分类和遗传算法的谐波间谐波频谱估计[J]. 电工技术学报,2008,23(6):109 – 113.

(上接第 153 页)

[6] Chen Y, Chiu W. Optimal robot path planning system by using a neural network-based approach[C]//2015 International Automatic Control Conference,2015:85 – 90.

[7] 王毅然,经小川,田涛,等. 基于强化学习的多 Agent 路径规划方法研究[J]. 计算机应用与软件,2019,36(8):165 – 171.

[8] Faust A, Chiang H, Rackley N, et al. Avoiding moving obstacles with stochastic hybrid dynamics using PEARL: Preference appraisal reinforcement learning[C]//2016 IEEE International Conference on Robotics and Automation, 2016: 484 – 490.

[9] Shalev-Shwartz S, Shammah S, Shashua A. Safe, multi-agent, reinforcement learning for autonomous driving[EB]. arXiv:1610.03295,2016.

[10] Kulkarni T, Narasimhan K, Saeedi A, et al. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation[C]//30th International Conference on Neural Information Processing Systems, 2016: 3682 – 3690.

[11] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature,2015, 518:529 – 533.

[12] Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay[EB]. arXiv:1511.05952,2015.

[13] 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.

[14] 何明,张斌,柳强,等. MADDPG 算法经验优先抽取机制研究[J]. 控制与决策,2021,36(1):68 – 74.

[15] Lowe R, Wu Y, Tamar A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[C]//31st International Conference on Neural Information Processing Systems,2017:6382 – 6393.

[16] Wang Z, Schaul T, Hessel M, et al. Dueling network architectures for deep reinforcement learning[EB]. arXiv:1511.06581,2015.

[17] 黄铎,应娜,蔡哲栋. 基于强化学习的多人姿态检测算法优化[J]. 计算机应用与软件,2019,36(4):186 – 191.