基于改进分层 DQN 算法的智能体路径规划

杨尚志 张 刚* 陈跃华 何小龙
(宁波大学海运学院 浙江 宁波 315211)

摘 要 针对智能体使用 DQN(Deep Q Network)算法进行路径规划时存在收敛速度慢、Q 值难以准确描述动作好坏的问题,提出一种优化 DQN 模型结构的分层 DQN 算法。该算法建立的激励层和动作层叠加生成更为准确的 Q 值,用于选择最优动作,使整个网络的抗干扰能力更强。仿真结果表明,智能体使用分层 DQN 算法的收敛速度更快,从而验证了算法的有效性。

关键词 分层 DQN 神经网络 强化学习 路径规划

中图分类号 TP181 TP3 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2024.05.035

PATH PLANNING FOR AGENT BASED ON IMPROVED LAYERED DQN ALGORITHM

Yang Shangzhi Zhang Gang* Chen Yuehua He Xiaolong
(*Faculty of Maritime and Transportation, Ningbo University, Ningbo 315211, Zhejiang, China*)

Abstract In order to solve the problems that the convergence speed is slow and it is difficult for Q value to describe the action accurately when an agent uses DQN algorithm in the process of path planning, a layered DQN algorithm optimized by the model structure of DQN is proposed. The excitation layer and the action layer built by the algorithm were superimposed to generate a more accurate Q value, which was used to select the optimal action and make the anti-interference ability of the whole network stronger. The simulation results show that the agent using layered DQN algorithm has a faster convergence speed, thus verifying the feasibility and effectiveness of the algorithm.

Keywords Layered DQN Neural network Reinforcement learning Path planning

0 引 言

路径规划是机器人研究领域的一个关键技术,是指移动机器人在有障碍物的环境中按照某些性能指标寻找一条最优或近似最优的无碰撞路径[1-2]。按照规划的智能程度可分为传统的路径规划算法和智能算法,传统的路径规划算法包括栅格法、人工势场法和 A* 算法等。文献[3]提出一种新的栅格建模方法,用于蚁群算法的 AGV 路径规划中,提高了算法的收敛速度。文献[4]将改进的人工势场法与混沌优化算法结合,降低陷阱区域的影响,减少了局部最优解,但收敛速度较慢。文献[5]提出一种改进的 A* 算法,该算法根据目标点的距离远近调整搜索速度,提高了实时性,但在搜索空间较大时稳定性不足。

智能算法包括遗传算法、蚁群算法、粒子群算法、深度强化学习等。使用更加高效的智能算法在复杂环境下完成智能体的路径规划是当前研究的热点。文献[6]提出了自适应交叉和变异概率方法优化遗传算法,提高了算法的收敛速度,但参数难以确定。文献[7]采用自适应参数方法改进蚁群算法,使得算法在前期和后期搜索中均有较快的收敛速度,但算法存在容易陷入局部最优的问题。文献[8]将粒子群优化算法用于路径规划的全局范围内,充分提高了路径规划的速度,但是存在搜索空间过大的问题。总体而言,传统的路径规划算法计算效率、决策实时性有待进一步提高,智能算法在模型简化、降低算法的复杂度、提高收敛速度方面需要进一步优化。

收稿日期:2021-01-30。国家自然科学基金项目(51675286);浙江省重点研发项目(2018C02G2070536)。**杨尚志**,硕士生,主研领域:机器学习。**张刚**,高工。**陈跃华**,副教授。**何小龙**,硕士生。

深度强化学习(Deep Reinforcement Learning,DRL)是一种结合了深度学习的感知能力和强化学习的决策能力的方法,具有无需环境模型、鲁棒性强和自主探索的优点,学者们将其广泛应用于智能体的自主避障[9]和路径规划[10]中。文献[11]使用 Q-learning 算法实现了无人船舶的路径规划,但是存在收敛速度慢、维数灾难[12]等问题。在深度学习(Deep Learning,DL)的基础上,文献[13]提出了深度 Q 网络(DQN)算法,将 Q-learning 算法和神经网络相结合,提高了算法的收敛速度,解决了维数灾难的问题。文献[14]在 DQN 算法的基础上,使用双网络结构有效降低了预测 Q 值和目标 Q 值的相关性,进一步提高了算法的稳定性。文献[15]提出一种竞争网络模型(Dueling DQN)结构,将输出分为动作值和状态值,减小了值函数大小与动作无关的影响;文献[16]提出一种可独立训练 Dueling DQN 网络模型用于机器人避障,将模型中的两个网络分别进行训练,使动作值更加准确;文献[17]提出一种 Double DQN 算法,算法使用两个不同的网络模型参数计算目标 Q 值,解决 Q 值过大的问题。由于在不同领域中,DQN 算法针对不同目标优化的效果存在差异,特别使用 DQN 算法进行路径规划时往往缺乏泛化性、存在过拟合等现象。

因此,本文基于 DQN 神经网络模型结构,提出一种分层 DQN 算法。将神经网络的输出分为激励层和动作层两层,叠加生成的 Q 值用于智能体在下一状态中最优动作的选择。智能体在处于未知的环境下,使用随机梯度下降算法进行神经网络的训练,不断更新网络模型参数,使得生成的 Q 值向最优逼近,进而在一定程度上选择最优动作。通过仿真对比,结果表明分层 DQN 算法在仿真环境中进行路径规划时,训练时间更短、收敛速度更快、稳定性大幅度提高。

1　改进分层 DQN 算法

1.1　DQN 算法原理

DQN 算法是一种将强化学习和深度学习相结合的端对端的深度强化学习算法。深度学习对当前状态下的高维数据进行特征提取,通过神经网络输出每个动作的 Q 值;强化学习根据 Q 值进行决策,选择最佳动作。最后根据强化学习反馈的数据训练神经网络,优化损失函数中的参数。DQN 算法使用一个权重参数为 θ 的卷积神经网络(CNN)[18]作为动作值函数 $Q(s,a,\theta_i)$ 的网络模型,将输入的状态经过卷积、全连接后,最后输出该状态下所有动作的 Q 值。模型结构如图 1 所示。

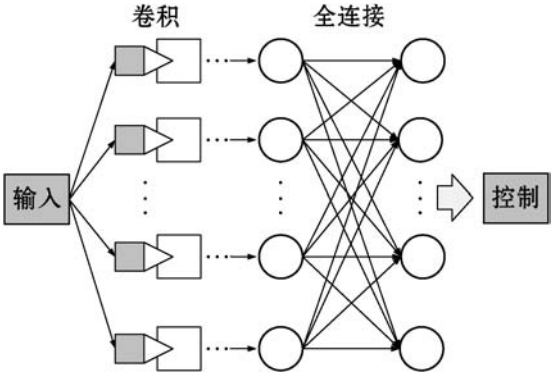


图 1　DQN 模型结构

DQN 除了使用预测网络 $Q(s,a,\theta_i)$ 来近似表示当前的动作值函数外,还另外使用目标网络 $Q(s,a,\theta_i^-)$ 来产生目标 Q 值。目标 Q 值一般用式(1)表示。

$$T_{\text{argetQ}} = r + \gamma \max_{a'}(s',a',\theta_i^-) \tag{1}$$

式中:r 表示智能体采取行动后获得的即时奖励;γ 表示衰减系数,决定智能体对未来奖励的看重程度。

随后使用均方误差(Mean-square Error)定义 DQN 的目标函数,作为神经网络的损失函数,如式(2)所示。

$$L_i(\theta_i) = E[(r + \gamma \max_{a'} Q(s',a',\theta_i^-) - Q(s,a,\theta_i))^2] \tag{2}$$

式中:参数 s' 和 a' 为下一时间步的状态和动作。

最后,使用小批量随机梯度下降算法(SGD)实现对损失函数 $L(\theta)$ 的权重参数 θ 的更新,从而得到最优动作值(Q 值),如式(3)所示。

$$\nabla_{\theta_i} L_i(\theta_i) = E[(r + \gamma \max_{a'} Q(s',a',\theta_i^-) - Q(s,a,\theta_i))\nabla_{\theta_i} Q(s,a,\theta_i)] \tag{3}$$

预测网络的参数 θ 根据式(3)的损失函数进行实时更新,每经过 N 轮迭代后,将预测网络的参数 θ 复制给目标网络中的参数 θ^-。从而在一定程度上降低了预测 Q 值和目标 Q 值的相关性,使得训练时损失值震荡发散的可能性降低,从而提高了算法的稳定性。除了双神经网络,DQN 还在训练过程中使用了经验回放机制、小批量随机采样数据进行训练,有效地去除了样本间的相关性和依赖性,使网络模型更容易收敛。

但是在一些深度强化学习任务中,某状态下无论选择何种动作,对下一状态都没有多大影响,此时智能体的动作值函数的大小是与动作无关的。在环境突然变化时,之前状态下的无效动作影响了智能体在干扰下的决策能力。最后导致 Q 值拟合性不足,难以真实描述动作的好坏。

1.2　分层 DQN 算法的模型结构

针对某些状态下,DQN 值函数受到外界干扰后区分度很低、拟合性较差、智能体难以选出最优动作,导

致算法最后收敛速度慢的问题,本文基于 DQN 算法,结合探索与利用平衡策略以及随机梯度下降法等方法,提出分层 DQN 算法。该算法优化了 DQN 的模型结构,将每个动作的 Q 值施加一个动态的激励值,使 Q 值的拟合性更好。

本文提出分层 DQN 算法,所谓分层即在神经网络隐藏层和输出层之间加入一组神经元节点,且将其分为两层,一层为激励层,用来给不同的 Q 值施加一个激励值,放大或缩小该动作的影响;另一层为动作层,用来描述不同动作的好坏程度。具体分层 DQN 的模型结构如图 2 所示。

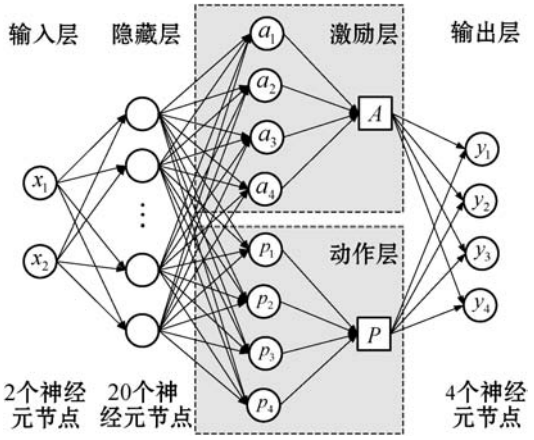


图 2 分层 DQN 的模型结构

在输入层,需要输入路径规划的状态信息,而在仿真环境下的状态是由智能体此时的位置坐标(x,y)决定,故定义 2 个神经元节点。可将智能体当前状态下的位置坐标(x,y)与目标点的位置坐标(x_0,y_0)的单位距离作为输入值,用 $\boldsymbol{s}=[x_1,x_2]$ 表示。$\boldsymbol{s}$ 为二维数组的形式,可以一次将 n 组数据作为输入值,用于神经网络的训练。

在隐藏层,神经元节点的个数理论上是没有限制的,但是一方面受限于计算机的计算能力,另一方面过多的节点会导致过拟合现象。因此在本文的网络模型中,实际上设置 20 个神经元节点,用于对输入值的特征提取。

而隐藏层节点一般是输入值与神经网络参数的线性组合,为了提升神经元模型的表达和处理能力,需要加入一个激活函数处理非线性问题。其表达式如式(4)所示。

$$\boldsymbol{y}=\mathrm{ReLU}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}+\boldsymbol{b}) \tag{4}$$

式中:$\boldsymbol{y}$ 为隐藏层节点的输出信号;激活函数为 ReLU;$\boldsymbol{W}$ 为神经网络权重矩阵;$\boldsymbol{X}$ 为状态输入矩阵;$\boldsymbol{b}$ 为偏置参数。

在隐藏层之后定义两组共 8 个神经元节点,每组分别对应到激励层和动作层中。在动作层中,用动作函数 $P(s,a)=[p_1,p_2,p_3,p_4]$ 来表示 4 个动作的好坏,每个 p 值对应 4 个动作在该状态下各自的伪价值。在激励层中,用激励函数 $A(s,a)=[a_1,a_2,a_3,a_4]$ 来表示 4 个动作的影响程度,每个 a 值分别用于修饰动作层中的 p 值。

最后,由于智能体在栅格环境下定义了上下左右 4 个动作,故输出层设置 4 个神经元节点。每个节点将激励函数值和动作函数值对应聚合到一起得到每个动作的 Q 值,智能体根据一定策略选取最优动作。

不同于 DQN 中直接用来描述动作好坏的 Q 值,它无法减小外界对动作的干扰,波动较大。分层 DQN 通过动作函数对动作的暂时评价,经过激励函数对其处理后,放大或缩小 p 值,降低了外界的干扰。基于分层网络,使得智能体能够学习到更加真实的价值,拟合性更好。

其中,激励函数 $A(s,a)$ 表达式为:

$$A(s,a)\cong A(s,a,\alpha,\mu) \tag{5}$$

式中:s 和 a 分别指输入的状态和该状态下的动作;α 为激励层中神经网络参数;μ 为激励函数中的激励系数,用于调节放大或缩小的比例。

动作函数 $P(s,a)$ 表达式为:

$$P(s,a)\cong P(s,a,\beta) \tag{6}$$

式中:s 和 a 同激励函数;β 为动作层中神经网络参数,根据输入生成伪价值。

最终的动作 Q 值表达式为:

$$Q(s,a)=A(s,a,\alpha,\mu)+P(s,a,\beta) \tag{7}$$

此时的 Q 值由激励函数和动作函数聚合而成,每个动作值均加有一个激励值用于调节动作的好坏程度。在某些状态下,智能体产生了无效动作,神经网络激励层首先初始化不同的激励值,用于优化动作函数,最后选择动作。进入到下一状态时,将此次激励层的参数 α 反馈到神经网络中,经训练后,输出的激励值将更准确。随着迭代次数的增加,激励函数可准确地修正无效动作带来的影响,小幅度修正正常状态下的动作函数。最后使生成的 $Q(s,a)$ 更具真实性,抗干扰能力更强。

在实际应用中,一般将动作函数设置为单动作函数值减去某状态下所有动作函数的平均值,如式(8)所示。

$$Q(s,a)=A(s,a,\alpha,\mu)+\left[P(s,a,\beta)-\frac{1}{|P|}\sum_{a}P(s,a,\beta)\right] \tag{8}$$

通过这样处理,能够保证缩小 Q 值范围,去除多

余的自由度进而提高算法稳定性。

1.3　分层 DQN 算法流程

在定义了分层 DQN 的模型结构后,基于 DQN 算法的技术特点,分层 DQN 算法的训练流程如图 3 所示。

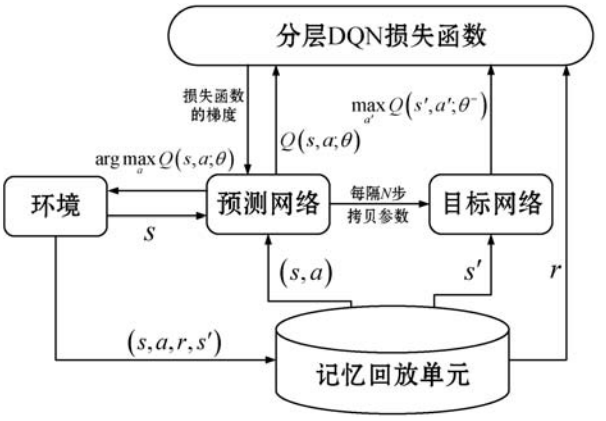


图 3　分层 DQN 算法训练流程

具体算法流程为:

1)初始化经验池 D,存储经验样本最大值为 N;

2)初始化预测网络的权重参数为 θ;

3)初始化目标网络的权重参数为 $\theta^- = \theta$;

4)开始循环 A,$i = 1$ 到 M,即最大回合数为 M;

5)初始化状态 s_1,处理输入数据;

6)开始循环 B,从 $t = 1$ 到 T,即记录每一回合的时间步 t;

7)以概率 ε 随机选择动作 a_t,否则根据 $a_t = \arg\max_a Q(s_t,a,\theta)$ 选择动作 a_t;

8)执行动作 a_t,获得奖励 r_t,进入下一状态 s_{t+1};

9)存储经验样本 (s_t,a_t,r,s_{t+1}) 到经验池 D 中;

10)从经验池中随机采样小批量地存储样本 (s_j,a_j,r_j,s_{j+1});

11)设目标值计算式为:

$$y_j = \begin{cases} r_j & \text{终止在时间步 } j+1 \\ r_j + \gamma \max\limits_{a'} \hat{Q}(s_{j+1},a',\theta^-) & \text{非终止时间步} \end{cases}$$

12)使用梯度下降算法更新损失函数 $(y_j - Q(s_j,a_j,\theta))^2$ 中的网络模型参数 θ;

13)每隔 C 步重设 $\hat{Q} = Q$;

14)退出循环 B;

15)退出循环 A。

2　实　验

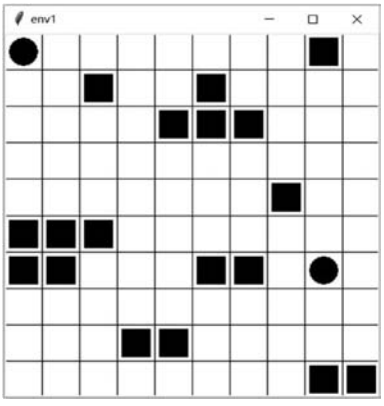
2.1　仿真环境与参数设定

本文仿真实验使用的平台如表 1 所示。

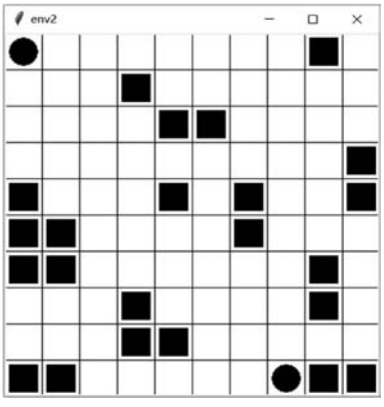
表 1　仿真实验平台

名称	版本型号
CPU	i7-4510U
RAM/GB	8
GPU	NVIDIA GeForce 840M
系统	Windows 10 专业版
TensorFlow	1.12.0
Python	3.6.2

本文使用 Python 中的标准 GUI 库 Tkinter 创建两个二维栅格环境(environment),环境一和环境二,用于模拟二维环境,如图 4 所示。环境二随机增加了一定障碍物,同时将目标点的距离调整更远。该环境的像素大小为 400 × 400,智能体的最小移动单位为 40 像素。每个栅格代表智能体的一个状态,每个状态下有上、下、左和右四个动作。图 4 中左上角的黑色圆圈代表智能体,即初始点,黑色矩形表示障碍物,右下角的黑色圆圈表示目标点,其他白色方块为安全区域。



(a)环境一



(b)环境二

图 4　不同的仿真环境

本文使用 TensorFlow 完成神经网络的搭建,使用小批量随机梯度下降算法(SGD)实现对损失函数 $L(\theta)$ 的权重参数 θ 的更新,在保证计算能力和稳定性的前提下,结合文献[14,19 - 20]的参数设置和本文仿真比较,使用的相关参数如表 2 所示。在训练过程

中,智能体的探索与利用策略使用 ε-greedy 策略,即智能体朝着奖励值最大的方向前进。该策略为:智能体以一定概率(ε)选择随机动作在环境中进行探索,以概率($1-\varepsilon$)选择价值最大的动作来尽可能地利用环境。

表 2 网络模型超参数

参数	值
学习率	0.01
衰减系数	0.9
贪婪策略概率	0.1
预测网络学习频率	5
目标网络更新频率	200
经验池的最大容量	5 000
每次训练的样本数	32

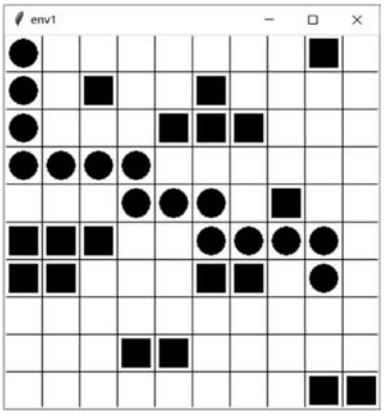
智能体在与环境交互过程中需要用奖励 r 作为对这次动作好坏的反馈,在仿真环境中设置的奖励函数如式(9)所示。

$$r=\begin{cases}1 & \text{到达目标点}\\ -1 & \text{碰到障碍物}\\ 0 & \text{在安全区域}\end{cases} \tag{9}$$

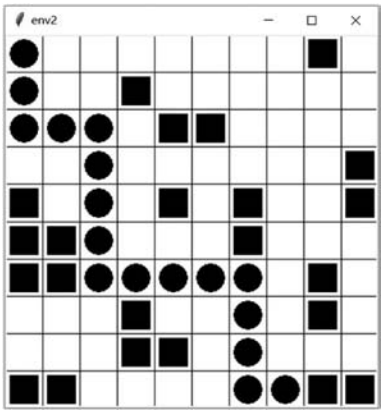
为了验证分层 DQN 在路径规划过程中的有效性和收敛速度的快慢,本文分别设置了两组在不同仿真环境下的实验。在环境一和环境二下,训练次数均设置为 5 000,分别将 DQN、Dueling DQN 和分层 DQN 三种算法到达目标点所需步长和误差曲线进行比较分析,得出结论。

2.2 实验结果与分析

智能体通过三种算法在两个仿真环境中依次经过 5 000 回合的训练后,均可得到的最优路径,如图 5 所示。该路径为智能体从初始点到目标点的最短距离。可见智能体在与环境交互过程中,使用三种 DQN 算法进行路径规划均能够达到要求。



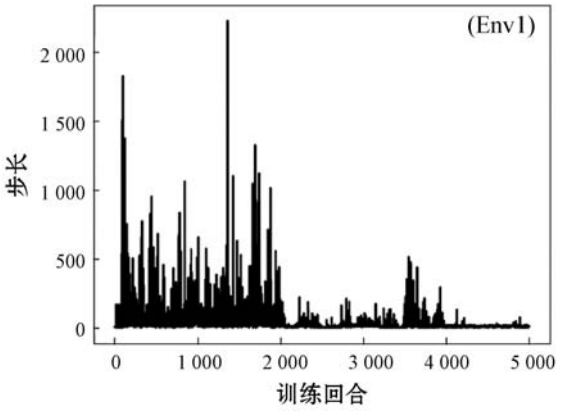
(a) 环境一下的最优路径



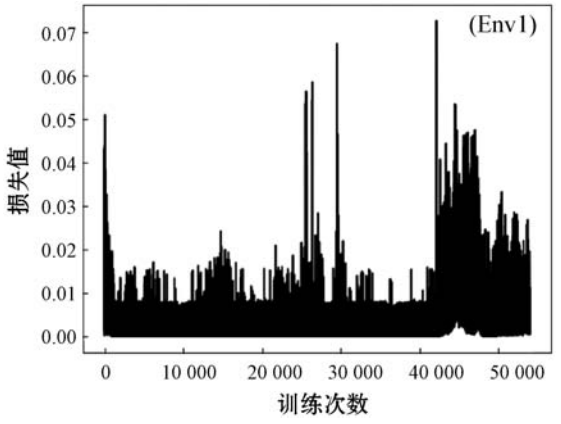
(b) 环境二下的最优路径

图 5 不同环境下的最优路径

在环境一下,图 6(a)、图 7(a)和图 8(a)分别为三种算法下智能体到达目标点所需步长与训练次数变化。可以看出,使用 DQN 算法进行路径规划收敛到最优步长时,大概需要训练 4 200 个回合左右,中间变化幅度较大,收敛速度慢,稳定性差。Dueling DQN 算法收敛时则大概训练了 3 000 个回合左右,收敛速度和稳定性均一般。而使用了分层 DQN 后,则发现训练了 1 000 个回合左右就收敛到最优步长。并且曲线呈缓慢下降状态,波动小,稳定性和收敛速度均为三种算法中最优。

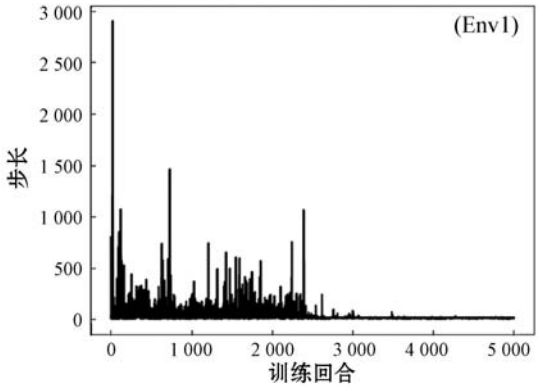


(a) DQN 的步长变化

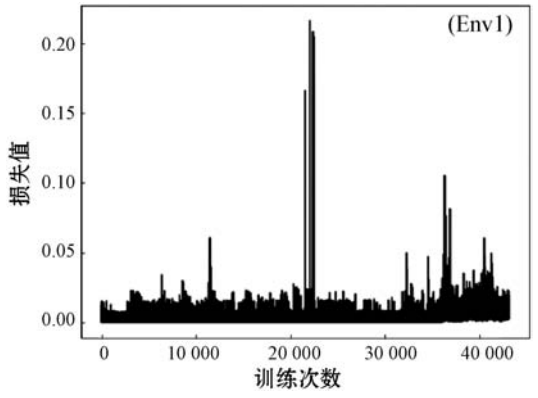


(b) DQN 的误差曲线

图 6 环境一下的 DQN 仿真分析

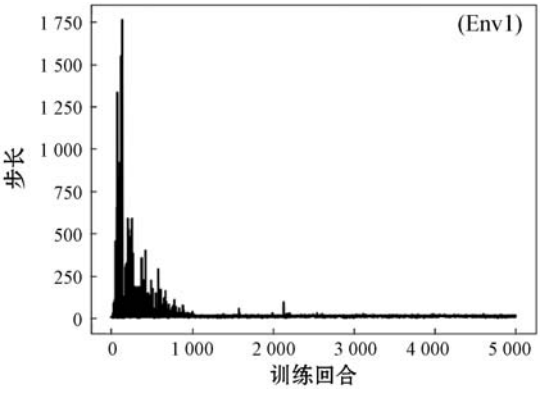


(a) Dueling DQN 的步长变化

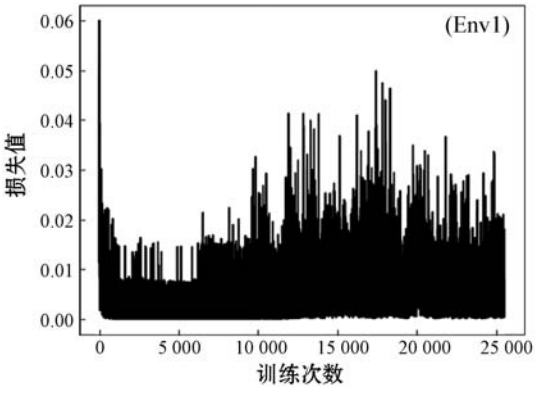


(b) Dueling DQN 的误差曲线

图 7　环境一下的 Dueling DQN 仿真分析



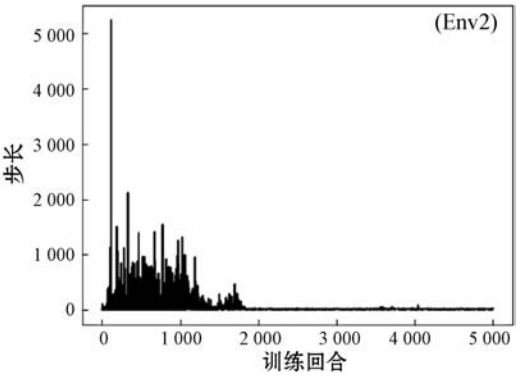
(a) 分层 DQN 的步长变化



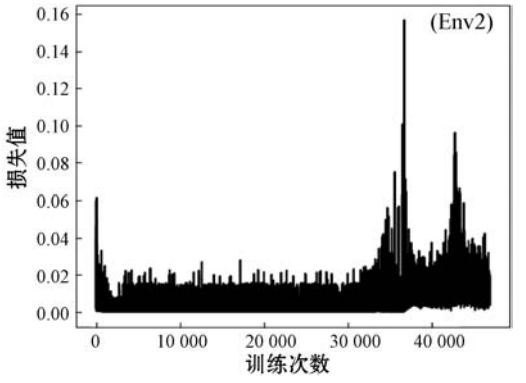
(b) 分层 DQN 的误差曲线

图 8　环境一下的分层 DQN 仿真分析

在环境二下,由图 9(a)、图 10(a)和图 11(a)可见,使用 DQN 算法大概在 1 800 回合左右收敛,Dueling DQN 大概 1 000 回合左右收敛,而分层 DQN 在 700 回合左右收敛。可见在环境和目标点改变后,分层 DQN 依然有良好的稳定性,收敛速度最快。

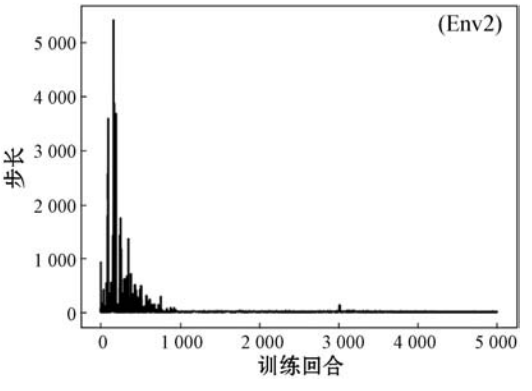


(a) DQN 的步长变化

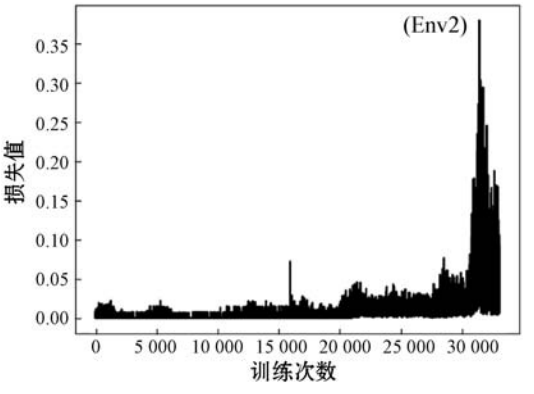


(b) DQN 的误差曲线

图 9　环境二下的 DQN 仿真分析

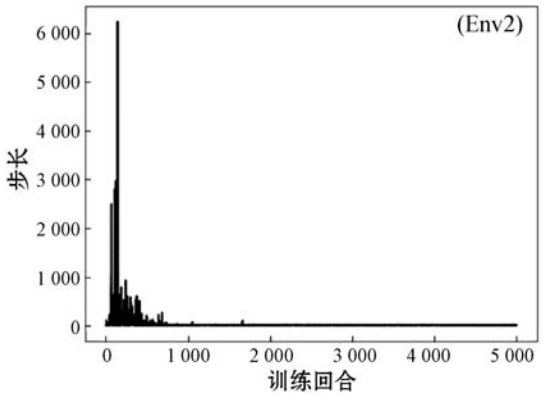


(a) Dueling DQN 的步长变化

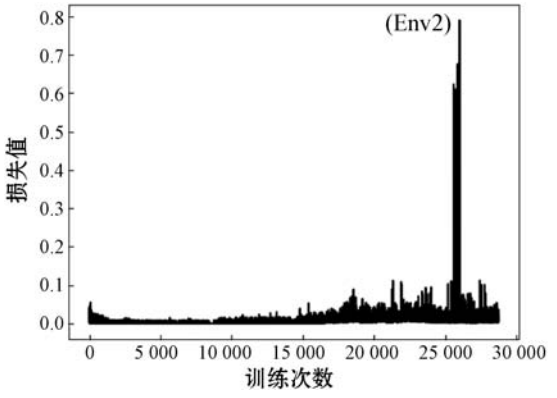


(b) Dueling DQN 的误差曲线

图 10　环境二下的 Dueling DQN 仿真分析



(a) DQN 的步长变化



(b) DQN 的误差曲线

图 11 环境二下的 DQN 仿真分析

图 6(b)、图 7(b)、图 8(b)、图 9(b)、图 10(b)和图 11(b)分别记录了两个环境下三种算法各自神经网络每次训练时产生的损失值(loss 值),即误差曲线。可见三种算法中的误差曲线不同于一般深度学习中不断下降最后趋于收敛的误差曲线,它是一个不断浮动的过程。

因为 DQN 是一种模型无关(model-free)的强化学习方法,智能体的经验样本数据是通过不断学习探索得到的,处于一个时刻更新的状态,缺乏衡量的标准。而深度学习神经网络的训练需要大量已知数据集和标签,训练结果会不断逼近标签值。因此,分层 DQN 的误差曲线会是一个不断浮动的过程。

经过三种算法的比较分析,可以发现智能体在仿真环境下进行一定回合的训练后,均能够实现自主导航,路径寻优的功能。相比较下,使用分层 DQN 算法能够更加有效地拟合 Q 值,收敛速度更快,具有更好的稳定性。

在保证存在可行路径的条件下,随机增减一定的障碍物,同时将智能体的目标点调整得更远,在一定回合的训练下,智能体使用分层 DQN 算法依然收敛速度最快、稳定性较好,进而验证了算法的有效性和可行性。

3 结 语

本文基于 DQN 算法的模型结构,提出一种分层 DQN 算法,该算法神经网络输出的 Q 值由控制动作影响幅度大小的激励函数和用来暂时评价动作好坏的动作函数聚合而成。经过神经网络的训练和参数迭代,Q 值不断逼近真实值,智能体会选择最优动作。仿真对比验证了本文算法在智能体的路径规划过程中收敛速度更快,稳定性更强,拟合效果更好。但是训练过程中,如何提高经验样本数据的利用率和设置更佳的奖励函数还有待进一步研究。

参 考 文 献

[1] 乔双虎,郑凯,陈亚博. 一种基于拓展支持向量机的无人船路径规划方法[J]. 船舶工程,2020,42(7):130 - 137.

[2] 刘建华,杨建国,刘华平,等. 基于势场蚁群算法的移动机器人全局路径规划方法[J]. 农业机械学报,2015,46(9):18 - 27.

[3] 赵江,王晓博,郝崇清,等. 栅格图特征提取下的路径规划建模与应用[J]. 计算机工程与应用,2020,56(10):254 - 260.

[4] Zhang C. Path planning for robot based on chaotic artificial potential field method[C]//IOP Conference Series: Materials Science and Engineering,2018.

[5] 王维,裴东,冯璋. 改进 A* 算法的移动机器人最短路径规划[J]. 计算机应用,2018,38(5):1523 - 1526.

[6] 王雷,李明. 改进自适应遗传算法在移动机器人路径规划中的应用[J]. 南京理工大学学报,2017,41(5):627 - 633.

[7] 孙功武,苏义鑫,顾轶超,等. 基于改进蚁群算法的水面无人艇路径规划[J]. 控制与决策,2021,36(4):847 - 856.

[8] 孙波,陈卫东,席裕庚. 基于粒子群优化算法的移动机器人全局路径规划[J]. 控制与决策,2005(9):1052 - 1055,1060.

[9] Duguleana M, Mogan G. Neural networks based reinforcement learning for mobile robots obstacle avoidance[J]. Expert Systems with Applications,2016,62:104 - 115.

[10] Xin J, Zhao H, Liu D, et al. Application of deep reinforcement learning in mobile robot path planning[C]//2017 Chinese Automation Congress,2017:7112 - 7116.

[11] 王程博,张新宇,邹志强,等. 基于 Q-Learning 的无人驾驶船舶路径规划[J]. 船海工程,2018,47(5):168 - 171.

[12] Watkins C, Dayan P. Q-learning[J]. Machine Learning, 1992,8:279 - 292.

(下转第 239 页)

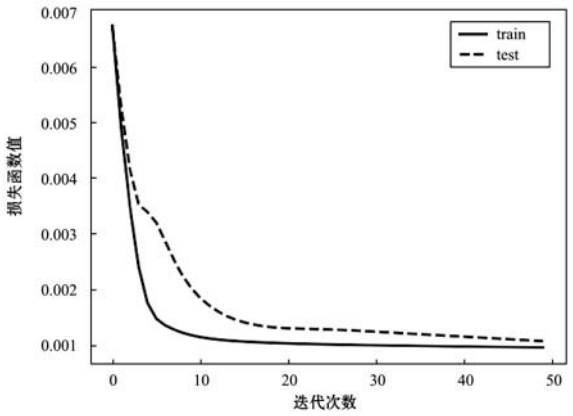


图 10　P-DTW-LSTM 的 Loss 曲线

进一步验证本文算法有效性,反归一化后 LSTM 预测模型下进行空气污染指数预测值和真实值的偏差度量,分别计算各方法的 RMSE 和 MAE,结果如图 11 所示,可知本文算法的 RMSE 值为 18.866,MAE 值为 12.839,其计算代价和其他各方法相比而言较低。

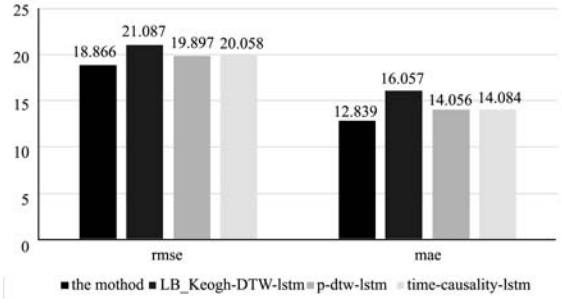


图 11　RMSE 和 MAE 标准下各方法变化柱形图

4　结　语

本文提出结合动态时间规整下界约束的过滤器因果时序预测方法,该方法通过改进 DTW 的下界约束构建的层级过滤器模型,并与格兰杰因果验证相结合以实现降维,挖掘出有效的输入特征价值信息,在 LSTM 时间序列预测模型中得到有效运用,该模型对于复杂多元的空气质量时序数据集是可行且有效的。而对于空气质量的影响因子较为复杂,同时空气质量的好坏尚有诸多其他影响因素,未来将致力于空气质量复杂网络内部运行机制及动态关系等方面的讨论研究。

参　考　文　献

[1] 杨海民,潘志松,白玮. 时间序列预测方法综述[J]. 计算机科学,2019,46(1):21 – 28.

[2] Li H, Gedikli E D, Lubbad R. Exploring time-delay-based numerical differentiation using principal component analysis [J]. Physica A: Statistical Mechanics and Its Applications, 2020,556:124839.

[3] 陶贵. DC/DC 电路的 P_DTW-LSTM 故障预测方法[J]. 电光与控制,2019,26(10):94 – 98.

[4] 史建楠,邹俊忠,张见,等. 基于 DMD-LSTM 模型的股票价格时间序列预测研究[J]. 计算机应用研究,2020,37(3):662 – 666.

[5] Hong J Y, Park S H, Baek J G. SSDTW: Shape segment dynamic time warping [J]. Expert Systems with Applications,2020,150:113291.

[6] Li T, Wu X, Zhang J. Time series clustering model based on DTW for classifying car parks[J]. Algorithms,2020,13(3):57.

[7] 任伟杰,韩敏. 多元时间序列因果关系分析研究综述[J]. 自动化学报,2021,47(1):64 – 78.

[8] 孙志伟,贾洪川,马永军. 基于 Time-Causality 模型的供热用气量预测分析[J]. 计算机应用与软件,2020,37(7):313 – 319.

[9] Varsehi H, Firoozabadi S M P. An EEG channel selection method for motor imagery based brain-computer interface and neurofeedback using Granger causality [J]. Neural Networks,2021,133:193 – 206.

[10] 周驰,李智,徐灿. 基于 DTW 算法的空间目标结构识别研究[J]. 计算机仿真,2019,36(9):98 – 102.

[11] Luo T. Research on decision-making of complex venture capital based on financial big data platform[J]. Complexity, 2018,2018:5170281.

[12] 左良利. 基于 DTW 的不确定时间序列分类方法研究[D]. 南京:南京航空航天大学,2019.

(上接第 232 页)

[13] Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with deep reinforcement learning[EB]. arXiv:1312.5602,2013.

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

[15] Wang Z, Schaul T, Hessel M, et al. Dueling network architectures for deep reinforcement learning[C]//33rd International Conference on Machine Learning,2016:1995 – 2003.

[16] 周翼,陈渤. 一种改进 dueling 网络的机器人避障方法[J]. 西安电子科技大学学报,2019,46(1):46 – 50.

[17] Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double Q-learning[C]//30th AAAI Conference on Artificial Intelligence,2016:2094 – 2100.

[18] Jain V, Seung H. Natural image denoising with convolutional networks[C]//21st International Conference on Neural Information Processing Systems,2008:769 – 776.

[19] 刘全,翟建伟,钟珊,等. 一种基于视觉注意力机制的深度循环 Q 网络模型[J]. 计算机学报,2017,40(6):1353 – 1366.

[20] 吴运雄,曾碧. 基于深度强化学习的移动机器人轨迹跟踪和动态避障[J]. 广东工业大学学报,2019,36(1):42 – 50.