doi:10.3969/j.issn.1003-3106.2022.08.020
引用格式:刘森,李玺,黄运.基于改进 DQN 算法的 NPC 行进路线规划研究[J].无线电工程,2022,52(8):1441-1446.[LIU Sen,LI Xi,HUANG Yun.Research on Marching Route Planning of NPC Based on Improved DQN Algorithm[J].Radio Engineering,2022,52(8):1441-1446.]

基于改进 DQN 算法的 NPC 行进路线规划研究

刘　森[1],李　玺[2*],黄　运[3]
(1. 河北远东通信系统工程有限公司,河北 石家庄 050200;
2. 陆军工程大学石家庄校区,河北 石家庄 050000;
3. 中国人民解放军 32620 部队,青海 西宁 810007)

摘　要: 在军事游戏或仿真训练系统中,非玩家角色(No-player Character,NPC)的行进路线规划是物理行为建模的重要组成部分。深度 Q 网络(Deep Q-Network,DQN)算法作为深度强化学习(Deep Reinforcement Learning,DEL)的经典算法,非常适合于路线规划这类策略选择的应用研究。针对该算法在收敛性和最优路线规划上存在的问题进行了优化,提出了改进算法 PRDQN。基于 TD-error 和 SumTree 对经验回放机制进行了改进,实现了优先级经验回放;根据距离优先的原则,对奖励函数进行了重新设计,提高了对距离最近坐标点的奖励值。通过对比实验证明,该算法在收敛性和最优路线选择方面都优于传统的 DQN 算法。

关键词: 深度 Q 网络;路线规划;优先级经验回放;最优路线
中图分类号: TP391　　**文献标志码:** A　　**开放科学(资源服务)标识码(OSID):**
文 章 编 号: 1003-3106(2022)08-1441-06

Research on Marching Route Planning of NPC Based on Improved DQN Algorithm

LIU Sen[1], LI Xi[2*], HUANG Yun[3]
(1. *Hebei Far-East Communication System Engineering Co., Ltd., Shijiazhuang* 050200, *China*;
2. *Shijiazhuang Campus of Army Engineering University, Shijiazhuang* 050000, *China*;
3. *Unit* 32620, *PLA, Xining* 810007, *China*)

Abstract: In military games or simulation training systems, the marching route planning of No-player Character (NPC) is an important part of physical behavior modeling. As a classic algorithm of Deep Reinforcement Learning (DRL), Deep Q-Network (DQN) algorithm is very suitable for the application of route planning. To address the problems of the algorithm about convergence and optimal route planning, an improved algorithm, PRDQN, is proposed. Based on TD-error and SumTree, the experience replay mechanism is improved, and prioritized experience replay is realized. According to the principle of distance first, the reward function is redesigned to improve the reward value of the nearest coordinate point. The comparison experiments show that the proposed algorithm is superior to the traditional DQN algorithm in terms of convergence and optimal route selection.

Keywords: DQN; route planning; prioritized experience replay; optimal route

0　引言

在军事游戏或仿真训练系统中,非玩家角色(No-player Character,NPC)的智能化水平在很大程度上决定了模拟训练的效果。构建 NPC 的核心是进行行为建模。行为建模分为认知行为建模和物理行为建模,行进路线的规划则是 NPC 物理行为建模的重要组成部分。传统的 NPC 行进路线规划主要采用固定路线或利用有限状态机、行为树构建脚本等方式。前者相对比较原始,很难有效提升模拟训练的效果,后者具有一定的智能性,但是面对复杂多变的战场环境往往不能灵活应对。本文针对上述问题开展研究,利用强化学习(Reinforcement Learning,RL)的方法设计 NPC 行进路线规划算法,实现 NPC 行进路线的智能化选择。

近年来,深度学习(Deep Learning,DL)和 RL 相

收稿日期:2022-01-10
基金项目:国家自然科学基金(62071483);国家社会科学基金军事学资助项目(15GJ003-184)
Foundation Item: National Natural Science Foundation of China (62071483); Military Science Project of National Social Science Foundation of China (15GJ003-184)

结合产生的深度强化学习(Deep Reinforcement Learning,DRL),使传统的 RL 扩展到高维度状态空间和高维度动作空间等以前无法解决的领域[1]。特别是 Mnih 等人[2]提出的深度 Q 网络(Deep Q-Network,DQN)将 DRL 的研究推向新的高度。目前,各国学者利用 DRL 在机器人控制[3-4]、游戏[5]和无人驾驶[6-7]等领域开展了广泛而深入的研究。其中,基于 DRL 的路线规划研究[8-14]也是热点之一,同时也是多领域的基础应用研究内容。

上述研究多关注于如何进行避障,而在此基础上如何实现最优路线的选择研究较少。另外,由于 DQN 算法在经验回放时采用均匀的采样方法,不利于算法的收敛。本文针对 DQN 算法收敛性差和最优路线选取问题进行改进,提出 PRDQN 算法。该算法利用基于 SumTree 的优先经验回放方法取代了 DQN 算法的均匀采样回放机制,并重新设计了奖励函数。实验证明,该算法相对于 DQN 算法不仅提高了收敛速度,而且实现了路线的优化。

1 PRDQN 算法

1.1 Q-learning 算法

Q-learning 是 RL 的经典算法,核心是智能体通过与环境的不断交互学习,更新和完善 Q-table,以达到智能决策的目的。Q-table 的行代表状态,列代表行动,表格的数值即 Q-value 是在不同状态下采取相应行动时能够获得的最大的未来期望奖励。Q-learning 算法利用式(1)来计算 Q-value:

$$Q(s_t,a_t) = Q(s_t,a_t) + \alpha[r_t + \gamma\max Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)] \text{。} \tag{1}$$

式(1)展开可得:

$$Q(s_t,a_t) = (1-\alpha)Q(s_t,a_t) + \alpha[r_t + \gamma\max Q(s_{t+1},a_{t+1})] \text{。} \tag{2}$$

将式(2)进行迭代,得:

$$\begin{aligned} Q(s_t,a_t) &= (1-\alpha)Q(s_t,a_t) + \alpha[r_t + \gamma\max Q(s_{t+1},a_{t+1})] = \\ &(1-\alpha)\{(1-\alpha)Q(s_t,a_t) + \\ &\alpha[r_t + \gamma\max Q(s_{t+1},a_{t+1})]\} + \\ &\alpha[r_t + \gamma\max Q(s_{t+1},a_{t+1})] = \\ &(1-\alpha)^2 Q(s_t,a_t) + \\ &[1-(1-\alpha)^2][r_t + \gamma\max Q(s_{t+1},a_{t+1})] = \\ &\cdots = \\ &(1-\alpha)^n Q(s_t,a_t) + \\ &[1-(1-\alpha)^n][r_t + \gamma\max Q(s_{t+1},a_{t+1})] \end{aligned} \tag{3}$$

式中,s_t 为 t 时刻的状态;a_t 为 t 时刻采取的行动;s_{t+1} 为 $t+1$ 时刻的状态;a_{t+1} 为 $t+1$ 时刻采取的动作;$Q(s_t,a_t)$ 为 s_t 状态下采用行动 a_t 的值函数;α 为学习率;r_t 为 t 时刻已经获得的奖励;γ 为衰减因子。由于 $\alpha \in (0,1)$,因此 $0 < \alpha - 1 < 1$,当 $n \to \infty$ 时,$(1-\alpha)^n \to 0$,则式(3)可变为:

$$Q(s_t,a_t) = r_t + \gamma\max Q(s_{t+1},a_{t+1}) \text{。} \tag{4}$$

式(4)即是在各类程序中,计算 s_t 状态下采用行动 a_t 的 Q 值公式。

利用 Q-learning 算法实现 RL 简单明了,并且不存在收敛性的问题。但是,当状态空间 S 和行动空间 A 足够大时,Q-table 会变得非常大,从而导致维数灾难。因此,纯粹的 Q-learning 算法很少用于解决现实中的各类应用问题。

1.2 DQN 算法

针对 Q-learning 算法存在的维数灾难问题,DeepMind团队的 Mnih 等人将深度卷积神经网络和 Q-learning结合,利用卷积神经网络动态生成 Q-table,不仅避免了复杂空间的维数灾难,而且在一定程度上解决了非线性函数近似表示值函数的不稳定问题。

DQN 算法示意如图 1 所示,算法定义了 2 个相对独立且结构相同的网络,分别是训练网络(TrainingNet)和目标网络(TargetNet)。利用 Q-learning 算法,智能体通过与环境的交互,实现训练网络的学习。经过固定步数的训练后,将训练网络中的参数全部赋值给目标网络。设置经验回放单元的目的是减少训练样本的相关行,改善神经网络逼近强化学习的动作值函数不稳定的问题。每次训练时从经验库中均匀选取一批样本与训练样本混合在一起,破坏相邻训练样本的相关性,提高样本的利用率。

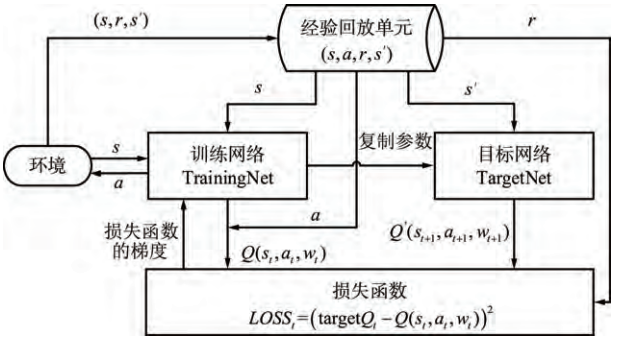


图 1 DQN 算法示意

Fig.1 Schematic diagram of DQN algorithm

DQN 的损失函数是目标网络的 Q-value 与评估网络的 Q-value 差的平方值,表示为:

$$LOSS_t = (\text{target}Q_t - Q(s_t,a_t,w_t))^2 \text{,} \tag{5}$$

式中,target$Q$ 根据式(4)可得:

$$\text{target}Q_t = r_t + \gamma\max Q'(s_{t+1},a_{t+1},w_{t+1}) \text{,} \tag{6}$$

w 为网络参数,采用梯度下降法学习:

$$w_{t+1} = w_t + E[\text{target}Q_t - Q(s_t,a_t,w_t)] \nabla Q(s_t,a_t,w_t) \text{。} \tag{7}$$

1.3 改进算法

传统的 DQN 算法利用经验回放机制，阻断了训练样本的相关行，改善了不稳定的问题，但是采用均匀采样的方式不利于算法的收敛。另外，DQN 算法奖励函数设置的比较简单，往往不能实现最优路线的规划。本文针对经验回放和奖励函数进行了改进，提出了 PRDQN 算法。

1.3.1 基于 SumTree 的优先经验回放

SumTree 采用二叉树结构，如图 2 所示。节点存储的是样本的优先级（Priority），数值越大，优先级越高，通过这种方式可以让好的样本重复利用的几率更大。SumTree 中，只有叶子节点代表具体样本，非叶子节点没有实际意义，父节点的优先级是子节点优先级的和。叶子节点的优先级通过 TD-error[15-17] 确定，TD-error 是样本在利用时序差分（Temporal Difference，TD）更新时目标网络值函数与训练网络值函数的差值，本文 TD-error 的值采用损失函数值，如式（5）所示。差值越大说明预测精度还有较大的上升空间，被训练的价值就越大，因此优先级越高。叶子节点下面的数值代表该样本对应的数值区间，优先级高的叶子节点对应的数值区间大，在均匀采样的过程中被选中的概率就高。SumTree 的算法实现如算法 1 所示。

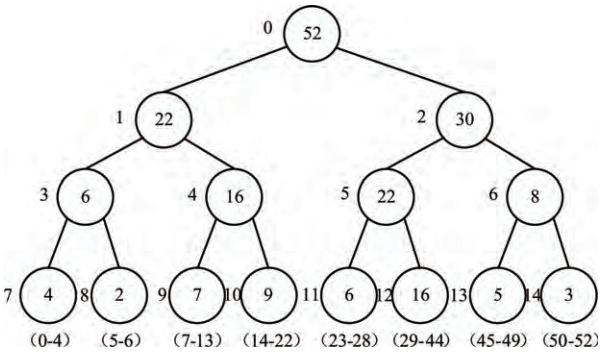


图 2 SumTree 示意

Fig.2 Schematic diagram of SumTree

算法 1：SumTree 算法

```
1)  Def SumTree(节点)
2)      if 节点==叶子节点
3)          if Priority节点<S
4)              节点=节点的兄弟节点
5)              return 节点
6)              break
7)      else
8)          if Priority节点的左子节点≥S
9)              节点=节点的左子节点
10)             SumTree(节点)
11)         else
12)             S=S- Priority当前节点的左子节点
13)             节点=节点的右子节点
14)             SumTree(节点)
```

以图 2 为例，假如从（0，52）中抽取样本 $S=25$，则从 SumTree 的根节点 0 开始，由于左子节点 1 的优先级为 22，小于 25，所以选择右子节点 2，同时 $S=25-22=3$，然后判断节点 2 的左子节点 5 的优先级值是否大于 3，依次判断下去，最后选择节点 12。

1.3.2 奖励函数的设计

在 RL 过程中，智能体在与环境的交互中能够获得奖励，驱使智能体在不断的尝试中选择获得奖励多的行为策略。传统的 DQN 算法奖励函数的设置如下：

$$r_t=\begin{cases}C,\text{到达目的地}\\-C,\text{发生碰撞},\\0,\text{其他}\end{cases}\tag{8}$$

式中，C 通常为正整数。可以看出，除去到达目的地和发生碰撞外，其他情况下智能体获得的奖励都是 0，通过大量训练虽然能够使智能体到达目的地，但是无法获取最优路线。为此，本文重新设计奖励函数，在没有到达目的地或发生碰撞时，判断下一个行动到达的点中哪一个距离终点距离更近，距离最近的点对应的行动奖励设为 βC：

$$r_t=\begin{cases}C,\text{到达目的地}\\-C,\text{发生碰撞}\\\beta C,0<\beta<1,D_a(x,y)=\min\limits_{i=1,2,3,4}[D_{a_i}(x,y)]\\0,\text{其他}\end{cases},\tag{9}$$

式中，与目的地的距离通过欧式距离公式计算得出，即：

$$D(x,y)=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\cdots+(x_n-y_n)^2}=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2},\tag{10}$$

式中，$x(x_1,x_2,\cdots,x_n)$ 是当前位置坐标；$y(y_1,y_2,\cdots,y_n)$ 是目的地坐标。

1.3.3 算法描述

本文提出的 PRDQN 算法流程如下。

算法 2：PRDQN 算法

1) 生成训练数据集
2) 初始化设置，利用 CNN 生成训练网络（TrainingNet）和目标网络（TargetNet）

```
3) 生成初始 Q-table
4) 训练网络
5)   设置 epochs,batchsize 等相关变量
6)   while count<epochs
7)     for i in range( 迭代数)
8)       选择行动策略( 采用 ε-greedy 算法,并结合奖励函数)
9)         if random( 0,1) <ε-greedy
10)          根据式( 9) 和式( 10) 计算奖励值,并选择行动
11)        else
12)          随机选择行动
13)      计算 SumTree,采用优先级回放机制
14)  运行训练网络( TrainingNet)
15)  保存训练网络模型
16)  更新目标网络( TargetNet)
17)  更新 Q-table,保存目标网络模型
18) 测试模型
19)  生成测试数据
20)  读取测试数据,读取目标网络模型
21)  迭代运行,输出测试结果
```

2 实验及结果分析

为了验证算法的有效性,本文设计了如下实验。随机生成 200 张 25 pixel×25 pixel 的图片作为 NPC 行进训练地图,每张图根据不同的起点,再生成 25×25=625 张图片,共计 125 000 张,图中随机生成若干黑色像素点代表地图中的障碍。NPC 从左上角的像素点 [0,0] 出发,到达右下角的像素点 [24,24] 则完成路线规划,中途碰到障碍则失败。实验软件环境: python 3.8,tensorflow 2.4.0,pycharm 2020.3.2。

本文设计了 2 层卷积网络加 2 层全连接网络的结构,卷积核大小为 2* 2,由于图像包含 3 个通道,因此卷积核的厚度为 3,第 1 次卷积包含 10 个卷积核,第 2 次卷积包含 20 个卷积核,第 1 层全连接层包含 100 个神经元,第 2 层全连接层包含 2 个神经元,生成的网络结构如图 3 所示。

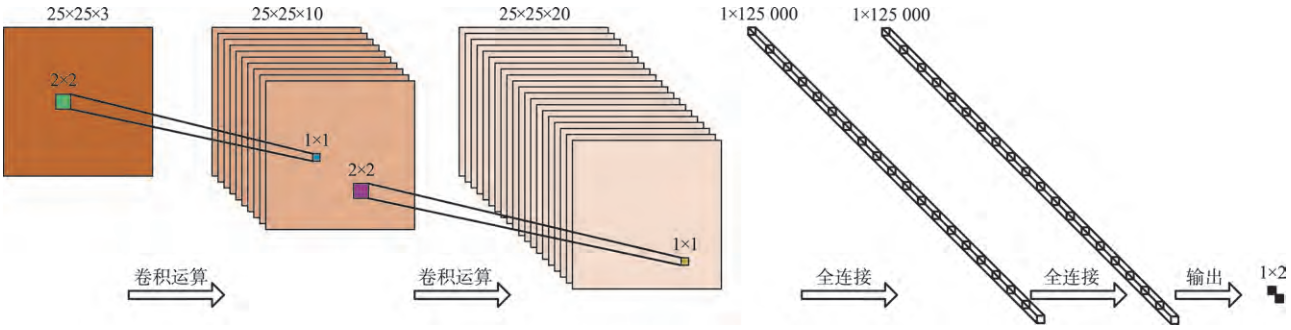


图 3 算法中深度卷积网络的处理过程

Fig.3 Processing flow of deep convolution network in the algorithm

实验程序参数设置如表 1 所示。

表 1 参数设置表

Tab.1 Parameter setting

参数	含义	数值
imageSize	样本大小	25×25×3
kernelSize	卷积核大小	2×2
α	学习率	0.001
γ	衰减系数	0.9
ε	贪婪指数	0.9
batchSize	批大小	100
epochs	训练周期	8
action	动作空间	4
β	奖励函数调整参数	0.8
C	奖励函数奖励值	100

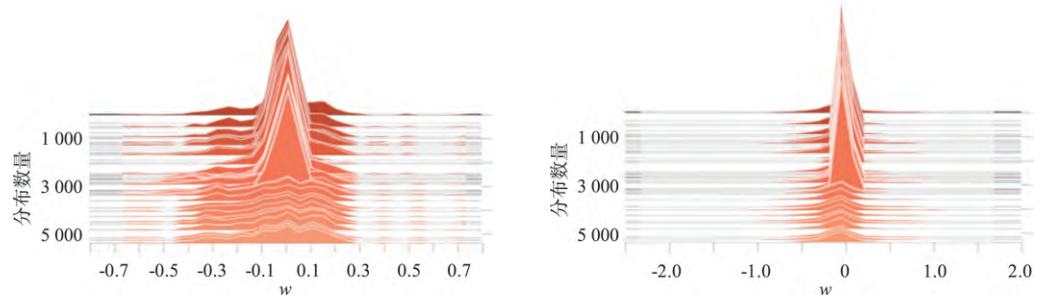
训练过程中的参数变化如图 4 所示。横轴代表 w 值,纵轴代表的是分布数量,每个切片代表在当前训练步数下 w 值的分布情况。

使用同样的训练集,分别对传统 DQN 算法和 PRDQN 算法进行训练,损失函数的变化情况如图 5 所示。

从图 5 的实验结果可以看出,本文提出的 PRDQN 算法在收敛性上要好于传统 DQN 算法,在训练步数到达 1 000 时,就可以将损失函数值稳定地控制在 30 以下。

训练完成之后,随机生成 10 张测试用图,对传统 DQN 算法和 PRDQN 算法进行测试,结果如图 6 所示。

从图 6 的实验结果可以看出,PRDQN 算法生成的规划路线基本沿图像对角线方向,从行进的角度来说,利于 NPC 在最短的时间内到达指定地点,不仅实现了路线规划,而且优化了行进路线。传统 DQN 算法虽然也能够完成路线规划,但是路线明显比较曲折。

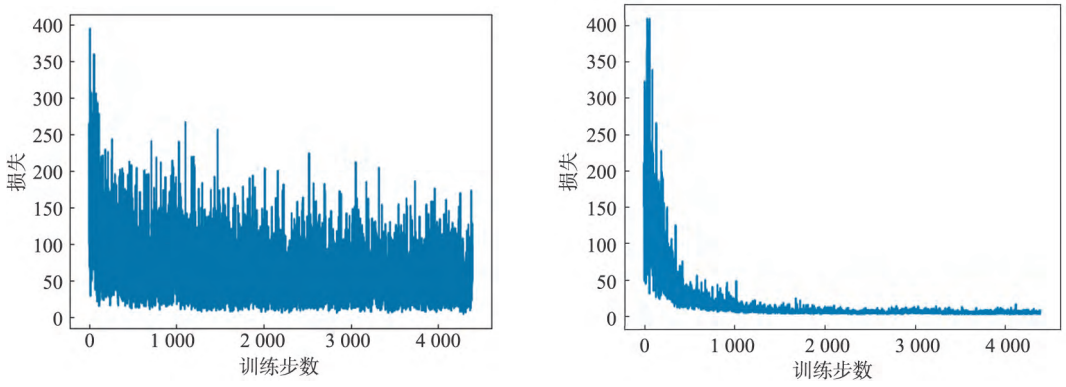


(a) 第 1 卷积层的 w 值　(b) 第 2 卷积层的 w 值

(c) 第 1 全连接层的 w 值　(d) 第 2 全连接层的 w 值

图 4　参数变化情况

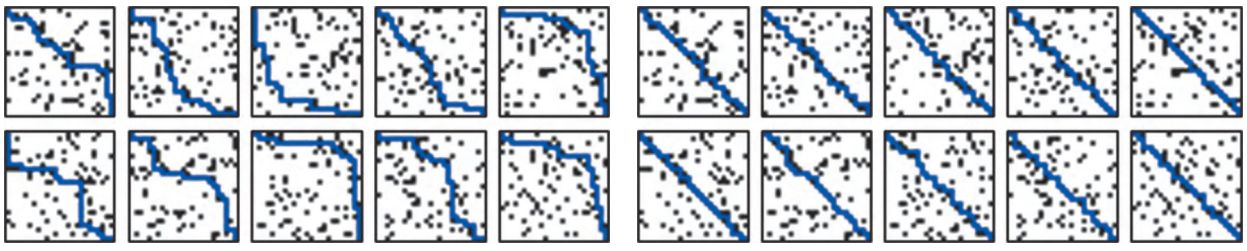
Fig.4　Parameter variation



(a) 传统 DQN 算法　(b) PRDQN 算法

图 5　损失函数值的变化情况

Fig.5　Variation of loss function value



(a) 传统 DQN 算法的测试效果　(b) PRDQN 算法的测试效果

图 6　测试结果对比

Fig.6　Comparison of test results

3 结束语

本文改进了传统 DQN 算法的经验回放机制和奖励函数，设计了 PRDQN 算法。该算法基于 TD-error 为经验回放单元中的样本设置优先级，并利用 SumTree 算法实现了优先级经验回放机制。同时，根据 NPC 行进的需求，重新设计了奖励函数。实验证明，PRDQN 算法在收敛性和最优路线规划方面都优于传统的 DQN 算法。

目前，算法应用于二维地图中，动作空间限于二维空间。在三维仿真训练系统中，行为策略将在三维空间中展开，需要针对奖励函数做进一步改进，考虑地形起伏等更多维度的因素对 NPC 行进的影响。

参考文献

[1] 刘建伟，高峰，罗雄麟.基于值函数和策略梯度的深度强化学习综述[J].计算机学报，2019，42(6)：1406-1438.

[2] MNIH V，KAVUKCUOGLU K，SILVER D，et al.Human-level Control through Deep Reinforcement Learning[J].Nature，2015，518(7540)：529-533.

[3] ZHANG M，GENG X Y，BRUCE J，et al.Deep Reinforcement Learning for Tensegrity Robot Locomotion[C]// IEEE International Conference on Robotics and Automation.Singapore：IEEE，2017：634-641.

[4] GU S X，HOLLY E，LILLICRAP T，et al.Deep Reinforcement Learning for Robotic Manipulation with Asynchronyous Off-policy Updates[C]// IEEE International Conference on Robotics and Automation.Singapore：IEEE，2017：3389-3396.

[5] MNIH V，KAVUKCUOGLU K，SILVER D，et al.Playing Atari with Deep Reinforcement Learning[J/OL].(2013-12-19)[2022-01-10].http//arxiv.org/abs/1312.5602.

[6] XIONG X，WANG J Q，ZHANG F，et al.Combining Deep Reinforcement Learning and Safety Based Control for Autonomous Driving[J/OL].(2016-12-01)[2022-01-05].https：//arxiv.org/abs/1612.00147v1.

[7] SALLAB A E L，ABDOU M，PEROT E，et al.Deep Reinforcement Learning Framework for Autonomous Driving[J].Electronic Imaging，2017，2017(19)：70-76.

[8] XIN J，ZHAO H，LIU D，et al.Application of Deep Reinforcement Learning in Mobile Robot Path Planning[C]// Chinese Automation Congress.Jinan：IEEE，2017：7112-7116.

[9] TAI L，LIU M.Towards Cognitive Exploration through Deep Reinforcement Learning for Mobile Robots[J/OL].(2016-10-06)[2022-01-06].https：//arxiv.org/abs/1610.01733.

[10] 董瑶，葛莹莹，郭鸿湧，等.基于深度强化学习的移动机器人路径规划[J].计算机工程与应用，2019，55(13)：15-19.

[11] 刘志荣，姜树海，袁雯雯，等.基于深度 Q 学习的移动机器人路径规划[J].测控技术，2019，38(7)：24-28.

[12] 楼建坤，王鸿东，王检耀，等.基于机器学习的实海域无人艇避碰算法智能演进方法[J].中国舰船研究，2021，16(1)：65-73.

[13] SHAMSOSHOARA A，KHALEDI M，AFGHA H，et al.Distributed Cooperative Spectrum Sharing in UAV Networks Using Multi-agent Reinforcement Learning[C]// 16th IEEE Annual Consumer Communications & Networking Conference (CCNC).Las Vegas：IEEE，2019：1-6.

[14] YANG Q，JANG S J，YOO S J.Q-learning-based Fuzzy Logic for Multi-objective Routing Algorithm in Flying Ad Hoc Networks[J].Wireless Personal Communications，2020，113(1)：115-138.

[15] SCHAUL T，QUAN J，ANTONOGLOU I，et al.Prioritized Experience Replay[C]// 4th Int.Conf.on Learning Representations (ICLR).New York：ICLR，2016：256-265.

[16] HARM V S，SUTTON R S.Planning by Prioritized Sweeping with Small Backups[J/OL].(2013-01-10)[2021-12-20].https：//arxiv.org/abs/1301.2343.

[17] 白辰甲，刘鹏，赵巍，等.基于 TD-error 自适应校正的深度 Q 学习主动采样方法[J].计算机研究与发展，2019，56(2)：262-280.

作者简介

刘 森 女，(1980—)，毕业于燕山大学通信与信息系统专业，硕士，高级工程师。主要研究方向：人工智能。

(*通信作者)李 玺 男，(1980—)，博士，副教授。主要研究方向：人工智能。

黄 运 男，(1979—)，工程师。主要研究方向：信息通信与网络。