ⓒ 应用数学和力学编委会, ISSN 1000-0887　　http://www.applmathmech.cn

基于输出层具有噪声的 DQN 的无人车路径规划*

李　杨[1]，　闫冬梅[2]，　刘　磊[1]

（1. 河海大学 理学院，南京 211100；
2. 南京邮电大学 现代邮政学院，南京 211100）

摘要： 在 DQN 算法的框架下，研究了无人车路径规划问题.为提高探索效率，将处理连续状态的 DQN 算法加以变化地应用到离散状态，同时为平衡探索与利用，选择仅在 DQN 网络输出层添加噪声，并设计了渐进式奖励函数，最后在 Gazebo 仿真环境中进行实验.仿真结果表明：① 该策略能快速规划出从初始点到目标点的无碰撞路线，与 *Q*-learning 算法、DQN 算法和 noisynet_DQN 算法相比，该文提出的算法收敛速度更快；② 该策略关于初始点、目标点、障碍物具有泛化能力，验证了其有效性与鲁棒性.

关　键　词： 深度强化学习；　无人车；　DQN 算法；　Gauss 噪声；　路径规划；　Gazebo 仿真
中图分类号： O29　　**文献标志码：** A　　DOI：10.21656/1000-0887.430070

UGV Path Programming Based on the DQN With Noise in the Output Layer

LI Yang[1]，　YAN Dongmei[2]，　LIU Lei[1]

（1. *College of Science*, *Hohai University*, *Nanjing* 211100, *P.R.China*;
2. *School of Modern Posts*, *Nanjing University of Posts and Telecommunications*, *Nanjing* 211100, *P.R.China*）

Abstract: The path programming of the unmanned ground vehicle (UGV) was studied under the framework of the deep *Q*-network (DQN) algorithm. To improve the exploration efficiency, the DQN algorithm was applied through discretization of the continuous state into the discrete state. To balance between exploration and exploitation, the Gaussian noise was added only in the output layer of the network, and a progressive reward function was designed. Finally, experiments were carried out in the Gazebo simulation environment. The simulation results show that, first, this strategy can quickly program a collision-free route from the initial point to the target point, and the convergence speed is significantly higher than those of the *Q*-learning algorithm, the DQN algorithm and the noisynet_DQN algorithm; second, this strategy has the generalization ability about the initial point, the target point and the obstacles, as well as verified effectiveness and robustness.

Key words: deep reinforcement learning; UGV; DQN algorithm; Gaussian noise; path programming; Gazebo simulation

* **收稿日期：** 2022-03-07；**修订日期：** 2022-12-08
基金项目： 国家自然科学基金(面上项目)(61773152)
作者简介： 李杨(1998—)，女，硕士生(E-mail：li_liiyang@163.com)；
闫冬梅(1988—)，女，讲师，博士(E-mail：ydm_1988@163.com)；
刘磊(1983—)，男，教授，博士，博士生导师(通讯作者. E-mail：liulei_hust@163.com).
引用格式： 李杨，闫冬梅，刘磊. 基于输出层具有噪声的 DQN 的无人车路径规划[J]. 应用数学和力学，2023，**44**(4)：450-460.

0 引　　言

无人车是电子计算机等最新科技成果与现代汽车工业相结合的产物,是集环境感知、规划决策、自主行驶等功能于一体的综合系统,它集中运用计算机、现代传感、信息融合、通讯、人工智能与自动控制等技术,是典型的高新技术综合体.路径规划是无人车自主导航的重要前提,也是无人车完成其他各项任务的基础.路径规划常采用 A* 算法[1]、遗传算法[2]、蚁群算法[3]、人工势场法[4]等,这些算法在无人车路径规划过程中都需要环境信息.强化学习方法[5-7]的出现以其独特的运行过程和泛化性能解决了众多问题,使得无人车进行路径规划时完全不需要环境信息,通过在训练过程中不断试错,与环境进行交互,采用延迟回报的方式寻找最优动作以获得最优决策能力[8],最终得到规划路径.

针对状态数较少的离散状态下的无人车路径规划,训练一般选取 Q-learning 算法[9-12],主要通过改进 Q-learning 算法或修改奖励函数提高算法训练效率.但当状态数较多时,使用此算法进行训练会出现维度爆炸问题,训练和收敛时间较长,成功率也较低.针对连续状态下的无人车路径规划,Q-learning 等表格型强化学习(reinforcement learning,RL)算法无法胜任,鉴于神经网络有极强的表达能力[13-14],众多学者使用神经网络代替 Q 表格,采用 DQN(deep Q-network)[15]等深度强化学习(deep reinforcement learning,DRL)算法进行训练研究.为提高算法的性能,一些新颖的训练技巧被应用到 DQN 中,例如董永峰等[16]通过动态融合 DDQN(double deep Q-network)与 averaged-DQN 的先验知识进行训练,有效降低了过估计的影响,但算法总体迭代次数较多,对算力要求较高.姜兰[17]提出了基于启发式知识的 DQN 算法,有助于加速神经网络的训练,但该算法在运用启发式知识时只考虑了避障而忽略了路径规划,致使规划路径过于冗余.丁志强[18]以 DQN 为基础,编程实现了 one-hot 编码状态映射、缩减浮点数位数、调用 SIMD 指令集等,最终在路径规划仿真中提高了算法的运行速度.总体来说,上述的多数改进算法都是基于贪婪策略进行探索,为增强 DQN 算法的探索性能,Fortunato 等[19]提出了 noisynet-DQN 算法,该算法将 DQN 网络中的线性层替换成噪声层取代传统的贪婪策略, 通过增加较少的计算成本实现探索性能的极大提升, 但是这样将导致算法的不稳定与计算能力的消耗.故本文借鉴 noisynet-DQN 的思想,在解决探索不足问题的前提下,保证算法的稳定与节省计算成本,并没有在全部全连接层添加噪声,只在输出层中加入 Gauss 噪声用于无人车避障.

此外,为了提升探索效率,本文将处理连续状态的 DQN 算法加以变化地应用到离散状态,并选择在输出层添加噪声的三层全连接层作为 Q 网络,避免了端对端处理模式对算力的依赖.考虑到有限的计算资源和算法落地的需求,本文选择具有可迁移性的 Gazebo 仿真平台对 ROS 无人车进行实验仿真.仿真结果显示此算法收敛速度、成功率与平均奖励明显高于 Q-learning 算法、DQN 算法与 noisynet_DQN 算法, 证明了在此环境下加入单层噪声的轻量级网络的有效性, 并通过测试证明了算法在起始点、目标点与障碍物方面具有泛化性能.

1 强 化 学 习

1.1 强化学习理论基础

强化学习最初是受到心理学领域关于人类和动物学习方面的影响而形成的.在强化学习中,智能体从外部环境感知状态 ($\boldsymbol{s}_t$), 随后智能体执行某动作 ($\boldsymbol{a}_t$), 该动作改变环境中原来的状态使智能体获得一个新的状态 ($\boldsymbol{s}_{t+1}$).在新的状态下,环境产生对智能体当前动作的奖励 (r_t), 此奖励是对动作好坏的评价,随后智能体根据获得的状态和奖励修正动作策略并采取下一个动作,如此反复迭代,与环境通过反馈信息进行交互以获得最大的累积奖励 ($G = r_1 + r_2 + \cdots + r_n + \cdots$).

1.2 DQN 算法

深度强化学习是在强化学习的基础上使用了神经网络,使得原始感官输入映射到原始电机输出成为可能(图 1).神经网络具有良好的拟合能力,能够通过简单函数的组成逼近任意的非线性函数.在 DQN 中,Q 网络取代了 Q-learning 的 Q 表,将环境状态利用非线性逼近映射成智能体的动作值.

网络架构、网络超参数的选择与学习都在训练阶段 (Q 网络权重的学习)中完成.在训练过程中,DQN 通过经验回放池来获取训练样本,网络的输入是智能体接收到的从环境中传来的状态,输出为智能体所有可能

动作的 Q 值,随后,采用 $\boldsymbol{\varepsilon}$-greedy 贪婪策略来选择动作并执行以平衡探索与利用之间的关系,环境根据所选动作反馈相应的奖励,定义损失函数为目标 Q 值与 Q 值之间的差:

$$\delta_{\text{loss}} = [r(\boldsymbol{s}_t,\boldsymbol{a}_t) + \gamma \max Q\_(\boldsymbol{s}_{t+1},\boldsymbol{a}_{t+1}) - Q(\boldsymbol{s}_t,\boldsymbol{a}_t)], \tag{1}$$

其中, $\max Q\_(\boldsymbol{s}_{t+1},\boldsymbol{a}_{t+1})$ 表示当状态为 $\boldsymbol{s}_{t+1}$ 时,在动作空间中选取恰当的动作得到的最大的行为值函数, $Q\_$ 表示目标 Q 网络.

对误差进行反向传播来更新 Q 网络的参数,如此不断进行训练,直到误差满足特定条件或者到达迭代次数时学习结束.具体运行流程见文献[15].

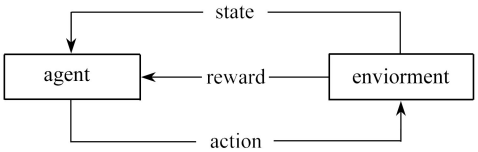


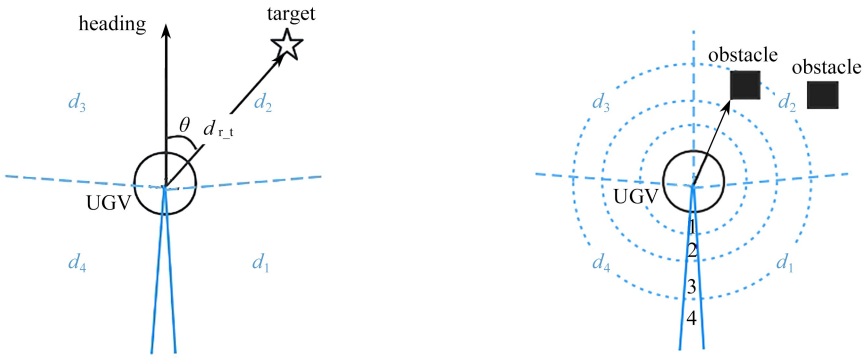
图 1 强化学习框架

Fig. 1 The reinforcement learning framework

2 算法模型设计

2.1 状态空间与动作空间设计

将图片作为输入的端对端算法需要强大的算力支持,将连续状态作为输入的算法计算时间相对较长,考虑到探索效率和算法性能,本实验采用离散化的思想,将连续状态适度离散化为离散状态,并作为算法的输入.状态的各分量划分成确定数量的非重叠区域,不同区域代表着无人车面临的不同环境信息.状态为五维向量,分别为无人车前进方向与目标点的夹角、雷达传感器在 4 个区域 d_1,d_2,d_3,d_4 中所感知到的障碍物的最近距离,即 $\boldsymbol{s} = (\theta,d_1,d_2,d_3,d_4)$, 状态示意图如图 2 所示.



(a) 状态向量各分量示意图 (b) $d_1 \sim d_4$ 区域划分图

(a) Schematic diagram of each component of the state vector (b) The zoning plan of $d_1 \sim d_4$

图 2 状态示意图

Fig. 2 The state diagram

状态分量具体划分如下所示,共有 3×4×4×4×4=768 个状态.

$$\theta = \begin{cases} 1, & 0° \leqslant \theta \leqslant 30°, \\ 2, & 30° < \theta \leqslant 90°, \\ 3, & \theta > 90°, \end{cases}$$

$$d_1 = \begin{cases} 1, & 0 \leqslant d_1 \leqslant 0.5, \\ 2, & 0.5 < d_1 \leqslant 1, \\ 3, & 1 < d_1 \leqslant 1.5, \\ 4, & d_1 > 1.5, \end{cases} \quad d_2 = \begin{cases} 1, & 0 \leqslant d_2 \leqslant 0.5, \\ 2, & 0.5 < d_2 \leqslant 1, \\ 3, & 1 < d_2 \leqslant 1.5, \\ 4, & d_2 > 1.5, \end{cases}$$

$$d_3=\begin{cases}1, & 0\leqslant d_3\leqslant 0.5,\\ 2, & 0.5<d_3\leqslant 1,\\ 3, & 1<d_3\leqslant 1.5,\\ 4, & d_3>1.5,\end{cases}\quad d_4=\begin{cases}1, & 0\leqslant d_4\leqslant 0.5,\\ 2, & 0.5<d_4\leqslant 1,\\ 3, & 1<d_4\leqslant 1.5,\\ 4, & d_4>1.5.\end{cases}$$

本实验的动作空间为 A,A 中共包含 4 个动作: 前进、后退、右转、左转, 具体为(0.5 m/s,0.0 rad/s),(−0.5 m/s,0.0 rad/s),(0.1 m/s,0.6 rad/s),(0.1 m/s,−0.6 rad/s),其中第一个分量表示线速度,第二个分量为角速度.

2.2　奖励函数的设置

在强化学习中,奖励函数是任务能否完美解决的重要因素,是无人车在某状态下所采取动作的评价.本实验的目的是使小车从初始点无碰撞且能够快速地到达目标点,故奖励函数的设置考虑到靠近目标点与远离障碍物这两个方面.

本文在文献[16]奖励函数的基础上,加入渐进式的奖励,这样可使小车能够更快地完成任务.具体为:当无人车碰到障碍物时,给予惩罚,奖励为负值;当无人车到达目标点时,给予奖励,奖励为正值.为了促使小车一直向目标点移动,当小车运行一步后与目标点的距离比上一步离目标点的距离更近时,以及当小车进入目标点的某一范围区域内时,给予正值奖励;其余情况给予−2,是为了防止小车出现循环运动获取值为 1 的奖励.具体如下所示:

$$r_t=\begin{cases}20, & d_{\mathrm{r\_t}}(t)\leqslant d_{\mathrm{t}},\\ -100, & d_{\mathrm{r\_o}}(t)\leqslant d_{\mathrm{o}},\\ 2, & d_{\mathrm{r\_t}}(t)\leqslant d_{\mathrm{n}},\\ 1, & d_{\mathrm{r\_t}}(t)<d_{\mathrm{r\_t}}(t-1),\\ -1, & d_{\mathrm{r\_t}}(t)=d_{\mathrm{r\_t}}(t-1),\\ -2, & \text{others},\end{cases}\tag{2}$$

其中, $d_{\mathrm{r\_t}}(t)$ 表示在 t 时刻无人车与目标点的距离,$d_{\mathrm{r\_o}}(t)$ 表示在 t 时刻无人车距离障碍物的最近距离,d_{o} 表示无人车撞到障碍物的阈值, d_{t} 表示实验所设定的无人车到达目标点的阈值, d_{n} 表示所设定的无人车靠近目标点的阈值.

2.3　输出层具有噪声的 DQN 算法

对无人车进行路径规划时,当所有可达状态处于可控(能够迭代)并且能存储在计算机 RAM 中时,Q-learning 算法能够很好地完成任务.然而,当环境中的状态数超过计算机容量时,Q-learning 算法中的 Q 表由于状态数过多,容易出现维度爆炸的问题.这时,一般会选取 DRL 算法来完成任务.

对于本文所设计的仅有 768 个状态的环境中,若使用 Q-learning 算法进行路径规划,虽不会出现维度爆炸问题,但状态数过大易导致实验效果较差;若使用复杂的深度强化学习算法,例如 DDPG、PPO 等会导致计算机算力成本增加.通过实验发现, 使用网络结构简单的 DQN 算法能够更快更好地完成本文的路径规划任务.

因状态数仅为 768 个,算法网络层数过多会出现训练时间长、收敛慢、浪费计算资源等问题.考虑到环境的复杂程度与计算效率,本文的算法网络只有 3 层全连接层.为提高探索性能,本文受到文献[19]的启发,在网络结构中加入 Gauss 噪声;为保证算法的稳定性和节省计算成本,仅在输出层添加分解 Gauss 噪声.所添加的噪声通过梯度下降法自动调整噪声强度,减轻了对任何超参数调优的需要,并且噪声所引发的探索程度是前后相关的,能够根据每个权重方差在不同的状态之间进行变化,此探索方法比 DQN 原本的 $\boldsymbol{\varepsilon}$-greedy 贪婪策略随机选择动作的探索性能更好,既保证了探索动作多样化,又提高了探索效率.具有 p 个输入,q 个输出的全连接层添加噪声后的具体表达式如下:

$$\boldsymbol{y}=(\boldsymbol{\mu}^w+\boldsymbol{\sigma}^w\odot\boldsymbol{\varepsilon}^w)\boldsymbol{x}+\boldsymbol{\mu}^b+(\boldsymbol{\sigma}^b\odot\boldsymbol{\varepsilon}^b),\tag{3}$$

其中, $\boldsymbol{x}\in R^p$,$\boldsymbol{y}\in R^q$,$\boldsymbol{\mu}^w\in R^{q\times p}$,$\boldsymbol{\sigma}^w\in R^{q\times p}$,$\boldsymbol{\varepsilon}^w\in R^{q\times p}$,$\boldsymbol{\mu}^b\in R^q$,$\boldsymbol{\sigma}^b\in R^q$,$\boldsymbol{\varepsilon}^b\in R^q$, $\odot$表示逐元素乘法, $\boldsymbol{\varepsilon}^w$, $\boldsymbol{\varepsilon}^b$ 为随机噪声参数.

综上所述,本文选择在输出层添加分解 Gauss 噪声的三层全连接层的 DQN 算法.具体的算法流程如图 3 所示.

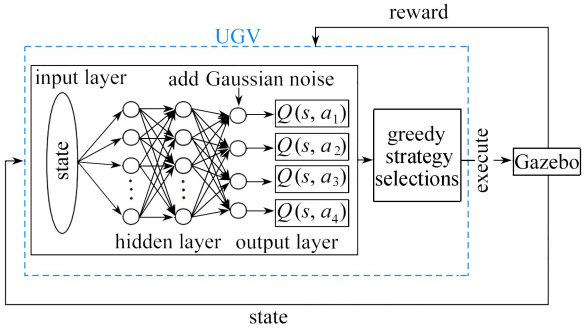


图 3 在输出层添加噪声的 DQN 算法框架

Fig. 3 The DQN algorithm framework for adding noise in the output layer

算法伪代码如下所示.

算法 1 输出层添加分解 Gauss 噪声的 DQN 算法

初始化经验回放池 D

随机初始化 Q 网络的参数 θ、目标 Q 网络的参数 θ^- 与随机噪音参数 $\boldsymbol{\varepsilon}$

for episode = 1 to maxepisode do

观测得到状态 $\boldsymbol{s}_1$

for t = 1 to T do

从 Gauss 分布中采样得到噪声 $\boldsymbol{\xi} \sim \boldsymbol{\varepsilon}$

通过贪婪策略选择动作 $\boldsymbol{a}_t = \max Q(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{\xi}; \theta)$

在环境中执行动作 $\boldsymbol{a}_t$

获得奖励 r_t 并到达下一个状态 $\boldsymbol{s}_{t+1}$

将经验 $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1})$ 存储到经验回放池 D 中

令 $\boldsymbol{s}_t = \boldsymbol{s}_{t+1}$

从经验回放池中随机采样 minibatch 个经验

for j = 1 to minibatch do

从 Gauss 分布中采样得到噪声 $\boldsymbol{\xi}' \sim \boldsymbol{\varepsilon}$

令 $y_j = \begin{cases} r_j, & \boldsymbol{s}_{j+1} \text{ is terminal}, \\ r_j + \gamma \max Q(\boldsymbol{s}_{j+1}, \boldsymbol{a}', \boldsymbol{\xi}'; \theta^-), & \text{others}. \end{cases}$

对 $(y_j - Q(\boldsymbol{s}_j, \boldsymbol{a}_j, \boldsymbol{\xi}''; \theta))^2$ 执行梯度下降策略更新 Q 网络的参数 $(\boldsymbol{\xi}'' \sim \boldsymbol{\varepsilon})$

end for

每隔固定步数对目标网络参数进行更新 $\theta^- = \theta$

end for

end for.

2.4 仿真环境

为计算方便起见,大多数文献(例如文献[12,16-18,20-21])选择在栅格地图等可视化环境中进行算法仿真,但这些仿真环境与现实环境有着较大的差别,致使实验结果难以令人信服.近年来,越来越多的学者采用 Gazebo 等逼真的仿真环境进行实验,例如文献[22-25].在栅格环境中进行实验与 Gazebo 环境进行仿真相比,环境由二维平面转换为三维空间,小车模型由一个点变为拥有动力学仿真和传感器仿真的真实模型,环境和模型的变化可使 Gazebo 平台的实验结果更加真实可信.

故本算法选择在 Gazebo 仿真环境下,使用基于机器人操作系统(ROS)搭建的差速式无人车进行实验.本实验中,在车体的前端安装了一个雷达传感器,可以检测到车体周围(-3 rad,3 rad)范围内的环境信息,雷

达传感器的检测有效范围为 0.10~30 m,精确到 0.01 m,雷达旋转一周发射 360 条射线,每一条射线进行一次测距,通过区域划分,将传感器感知到的范围划分成 4 个互不相交的区域,使用 d_1,d_2,d_3,d_4 表示无人车在 4 个区域内与障碍物的最短距离.计算无人车前进方向与目标点的夹角的绝对值的公式如下:

$$\theta = | \psi - \varphi | , \tag{4}$$

其中,φ 表示无人车前进方向与地图坐标系 x 轴正方向的夹角,ψ 表示目标点相对于无人车质心处地图坐标系 x 轴正方向的夹角.

在 Gazebo 中建立了 13×8 单位距离的仿真环境,环境四周由墙围住,中间设置 6 个障碍物,初始点为(0,0),目标点为(-6,1),Gazebo 仿真环境如图 4 所示,Rviz 仿真环境如图 5 所示.

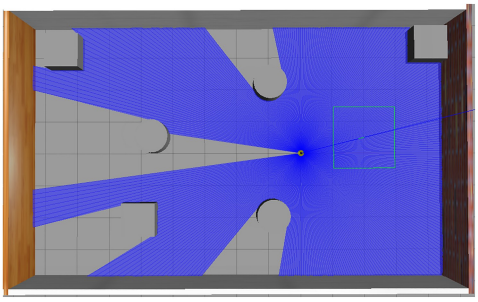


图 4　Gazebo 仿真环境

Fig. 4　The Gazebo simulation environment

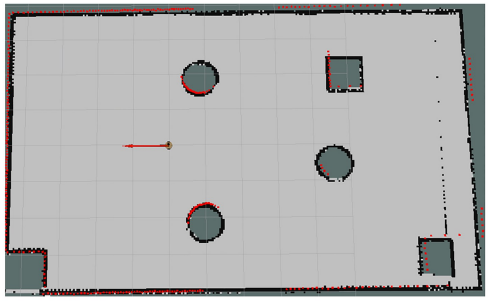


图 5　Rviz 仿真环境

Fig. 5　The Rviz simulation environment

注　为了解释图中的颜色,读者可以参考本文的电子网页版本,后同.

本实验在 UBUNTU 操作系统上运行,处理器为 Intel(R) Core(TM) i7-9700 CPU @ 3.00 GHz.无人车仿真使用 PYTHON 编程,Adam 优化器进行网络训练优化.算法的网络训练参数设置如表 1 所示.

表 1　算法训练参数

Table 1　Algorithm training parameters

parameter	meaning	value
α	learning rate	0.001
γ	discount factor	0.9
M	memory length	1 000
m	batch size during training	100
E	training number	1 000
d_t/m	target point threshold	0.25
d_o/m	obstacle threshold	0.15

本实验的任务是利用在输出层添加噪声的 DQN 算法,使无人车在未知的环境中规划出从初始点到目标点的无碰撞路径.实验共训练 1 000 个回合,智能体执行一次动作的时间为 1 s,无人车到达目标点或碰到障碍物则本回合结束.

3　实验结果分析

为了验证在输出层添加噪声的 DQN 算法在无人车路径规划中的有效性,采用 PYTHON 语言在 Gazebo 仿真环境中对 Q-learning 算法、DQN 算法、3 层全连接层都添加 Gauss 噪声的 DQN 算法与仅在输出层添加 Gauss 噪声的 DQN 算法进行 1 000 回合的实验.为区分算法的名称,用 changed_DQN 表示在输出层添加噪声的 DQN 算法,noisynet_DQN 表示 Q 网络中的 3 层全连接层都添加噪声的 DQN 算法.

图 6 为采用 Q-learning 算法、DQN 算法、changed_DQN 算法与 noisynet_DQN 算法训练 1 000 回合的成功率对比图.从图中看出,changed_DQN 算法训练效果最好,在 100 回合内成功率就开始快速提升,在 200 回合左右开始收敛,1 000 回合时成功率为 94.2%;Q-learning 算法在训练期间因状态空间略大,训练结果不稳定,成功率十分低,相比之下,3 个 DQN 算法的成功率都远高于 Q-learning 算法.依靠 ε-greedy 贪婪策略进行探

索的 DQN 算法,成功率提升速度缓慢,1 000 回合时成功率才 71.6%,明显低于另外两种 DQN 算法,这表明在网络结构中添加噪声的 DQN 算法的探索效率高于使用 $\boldsymbol{\varepsilon}$-greedy 贪婪策略的 DQN 算法;noisynet_DQN 算法在前期探索阶段成功率和 changed_DQN 相当,但后期因噪声添加过多导致算法不好收敛,成功率明显低于 changed_DQN .

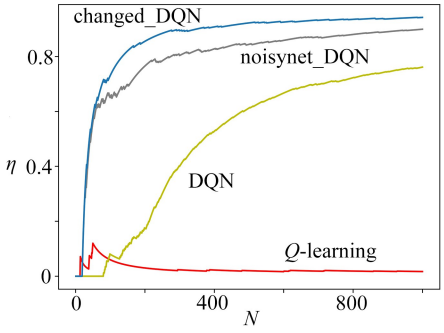


图 6 成功率对比图

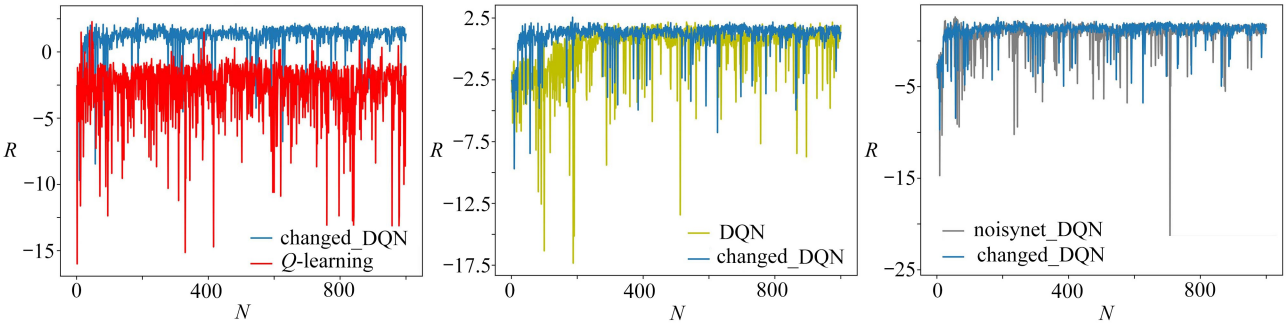
Fig. 6 Comparison of success rates

训练过程中每回合的平均奖励对比如表 2 与图 7 所示.从图 7(a)中可以看出,运用 changed_DQN 算法的小车所获得的平均奖励基本全部高于运用 Q-learning 算法的小车,且收敛在较高水平.从图 7(b)可看出,在前期 100 回合左右时,运用 changed_DQN 算法的小车的平均奖励已经开始收敛,而运用 DQN 算法的小车的平均奖励到 400 回合左右才开始收敛,且后期波动较大.从表 2 可看出,训练过程中,changed_DQN 算法的平均奖励均值最大,方差最小,这说明此算法稳定并且奖励收敛在较高的水平.

表 2 平均奖励的均值与方差

Table 2 The mean and variance of the mean rewards

	changed_DQN	noisynet_DQN	DQN	Q-learning
mean	1.081 32	0.707 14	0.091 377 8	−2.808 89
variance	1.460 35	3.299 63	4.695 35	5.056 47



(a) Q-learning 与 changed_DQN 平均奖励对比

(a) Mean reward comparison between Q-learning and changed_DQN

(b) DQN 与 changed_DQN 平均奖励对比

(b) Mean reward comparison between DQN and changed_DQN

(c) Noisynet_DQN 与 changed_DQN 平均奖励对比

(c) Mean reward comparison between noisynet_DQN and changed_DQN

图 7 平均奖励对比图

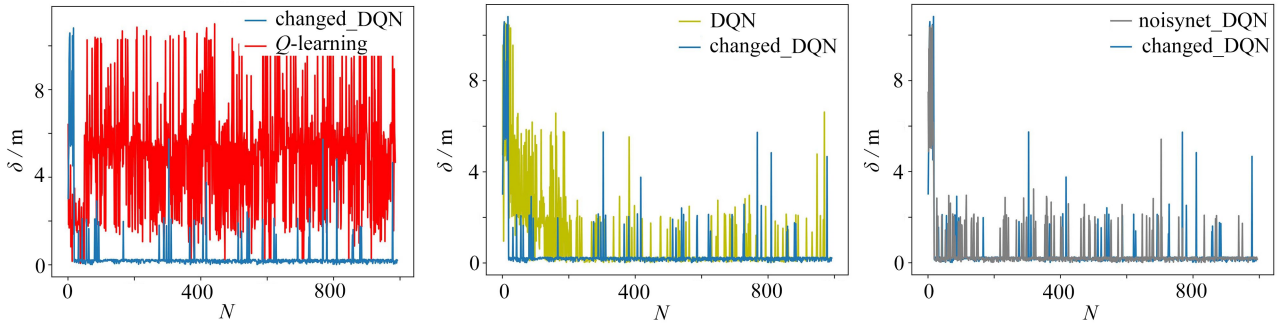
Fig. 7 Comparison of mean rewards

本实验的目的为无人车能够无碰撞地到达目标点,故选择无人车结束本回合训练时的位置与目标点的距离作为误差,误差越小表明越接近目标点.图 8(a)表明选择 changed_DQN 算法的小车在 50 回合后可较好地完成任务,收敛后还有波动是由于动作策略的探索导致的,选择 Q-learning 算法的无人车在训练中的大多数回合都未完成任务.从图 8(b)的对比可知, changed_DQN 算法比 DQN 算法收敛更快,且收敛后波动较小.从表 3 中可知,changed_DQN 算法误差的均值最小,方差最小,说明此算法稳定并且能更加精确地完成任务.

表 3　误差的均值与方差

Table 3　The mean and variance of the errors

	changed_DQN	noisynet_DQN	DQN	*Q*-learning
mean	0.379 32	0.454 48	0.873 16	4.911 19
variance	1.120 79	1.213 50	2.741 73	6.128 13



(a) *Q*-learning 算法与 changed_DQN 算法误差对比

(a) Error comparison between *Q*-learning and changed_DQN

(b) DQN 算法与 changed_DQN 算法误差对比

(b) Error comparison between DQN and changed_DQN

(c) Noisynet_DQN 算法与 changed_DQN 算法误差对比

(c) Error comparison between noisynet_DQN and changed_DQN

图 8　误差对比图

Fig. 8　Error comparison diagrams

对训练好的算法进行 50 次测试实验,*Q*-learning、DQN、changed_DQN 与 noisynet_DQN 算法的成功率分别为 46%,98%,96%,94%,从实验结果可以看出训练好的 changed_DQN 算法成功率略低于 DQN 算法,但相差不大(图 9).从完成任务的规划时间来看,changed_DQN 算法所需时间稳定且总体低于其他算法(图 10).这表明 changed_DQN 算法具有高效性.

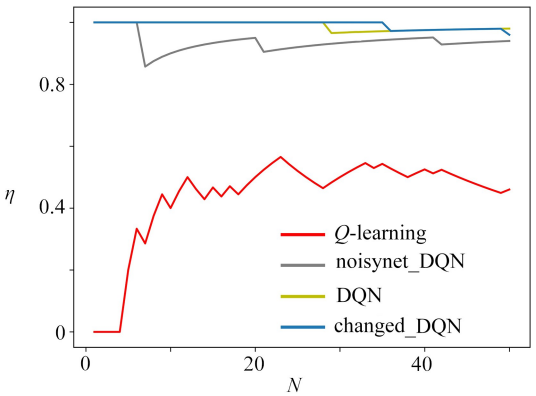


图 9　测试实验成功率

Fig. 9　Success rates of testing

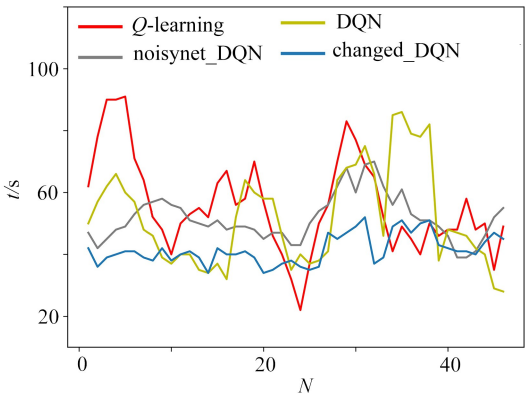


图 10　测试实验规划时间

Fig. 10　Programming time of testing

图 11、12 分别为 *Q*-learning 算法和 changed_DQN 算法在 1 000 回合左右的路径图.路径图以初始点为原点,运用 *Q*-learning 算法的小车在 1 000 回合左右未能规划出从初始点到目标点的无碰撞路径,运用 changed_DQN 算法的小车能规划出较为完美的无碰撞路径.

为了验证在输出层添加噪声的 DQN 算法的有效性和鲁棒性,分别在改变起始点、改变目标点和改变障碍物的环境中进行仿真实验.仿真结果表明,运用 changed_DQN 算法的无人车均能快速地找到从起始点到目标点的相对较优的无碰撞路径.图 13、14、15 分别为改变起始点、目标点和障碍物的路径规划图.图 16 为障碍物改变后的 Gazebo 仿真环境.

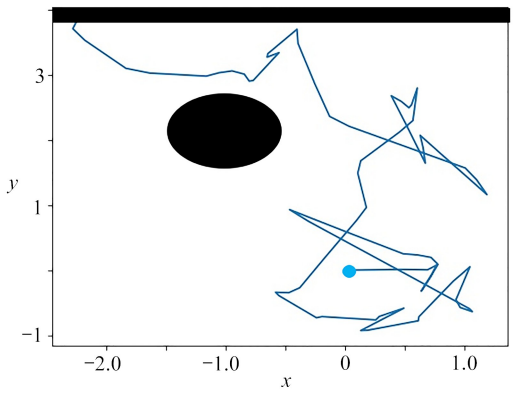


图 11 *Q*-learning 算法路径规划效果图

Fig. 11 Path programming effects based on *Q*-learning

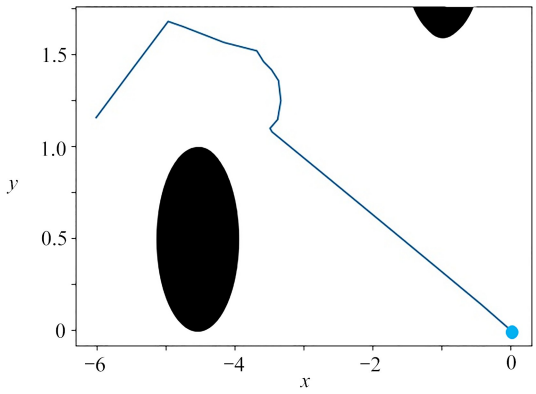


图 12 Changed_DQN 算法路径规划效果图

Fig. 12 Path programming effects based on changed_DQN

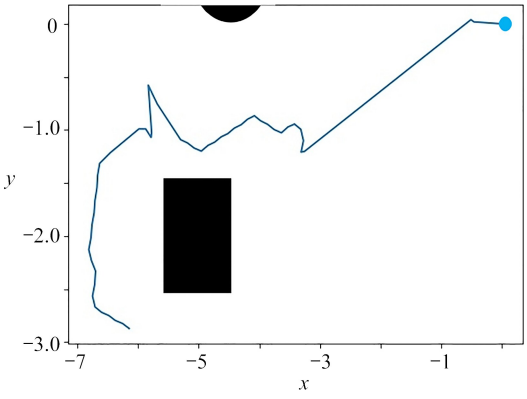


图 13 目标点改变时 changed_DQN 算法路径规划效果图

Fig. 13 Path programming effects based on changed_DQN with changing target point

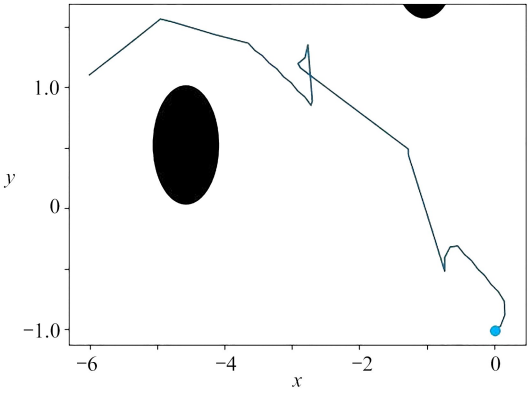


图 14 起始点改变时 changed_DQN 算法路径规划效果图

Fig. 14 Path programming effects based on changed_DQN with changing starting point

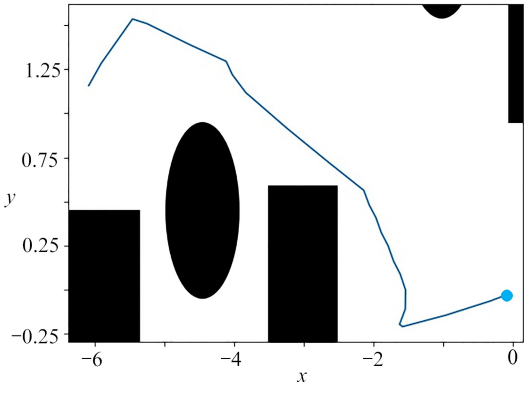


图 15 障碍物改变后的 changed_DQN 算法路径规划效果图

Fig. 15 Path programming effects of changed_DQN after the obstacle change

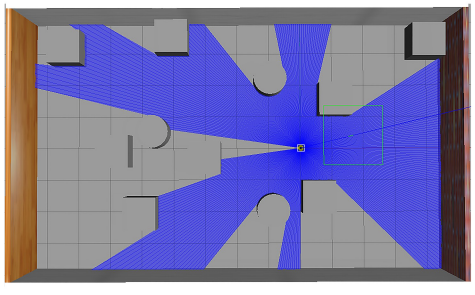


图 16 障碍物改变后的 Gazebo 仿真环境

Fig. 16 The Gazebo simulation environment after the obstacle change

4 结 束 语

本文针对离散状态空间中状态数量较多的无人车路径规划问题,提出了在输出层添加分解 Gauss 噪声的 DQN 算法进行路径规划,该算法节省了计算成本,平衡了探索与利用.仿真实验表明,算法的收敛速度和规划路线的优越性都高于处理状态离散问题常用的 *Q*-learning 算法,且误差收敛到 0 的速度更快,误差更

小,通过与 DQN、noisynet_DQN 算法实验对比发现,本文所采用的算法成功率更高,效果更好.在初始点、目标点和障碍物方面对本文的算法进行了泛化性能的测试,验证了该策略的有效性和鲁棒性.

参考文献(References):

[1] 王洪斌, 尹鹏衡, 郑维, 等. 基于改进的 A* 算法与动态窗口法的移动机器人路径规划[J]. 机器人, 2020, **42**(3): 346-353.(WANG Hongbin, YIN Pengheng, ZHENG Wei, et al. Mobile robot path planning based on improved A* algorithm and dynamic window method[J]. *Robot*, 2020, **42**(3): 346-353.(in Chinese))

[2] SONG Q, LI S, ZHE L. Automatic guided vehicle path planning based on improved genetic algorithm[J]. *Modular Machine Tool and Automatic Processing Technology*, 2020(7): 88-92.

[3] ZHANG S, PU J, SI Y, et al. Review on the application of ant colony algorithm in path planning of mobile robots[J]. *Computer Engineering and Applications*, 2020, **56**(8): 10-19.

[4] KOVÁCS B, SZAYER G, TAJTI F. A novel potential field method for path planning of mobile robots by adapting animal motion attributes[J]. *Robotics and Autonomous Systems*, 2016, **82**: 24-34.

[5] 马丽新, 刘晨, 刘磊. 基于 actor-critic 算法的分数阶多自主体系统最优主-从一致性控制[J]. 应用数学和力学, 2022, **43**(1): 104-114.(MA Lixin, LIU Chen, LIU Lei. Optimal leader-following consensus control of fractional-order multi-agent systems based on the actor-critic algorithm[J]. *Applied Mathematics and Mechanics*, 2022, **43**(1): 104-114.(in Chinese))

[6] 刘晨, 刘磊. 基于事件触发策略的多智能体系统的最优主-从一致性分析[J]. 应用数学和力学, 2019, **40**(11): 1278-1288.(LIU Chen, LIU Lei. Optimal leader-following consensus of multi-agent systems based on event-triggered strategy[J]. *Applied Mathematics and Mechanics*, 2019, **40**(11): 1278-1288.(in Chinese))

[7] CHEN Y F, LIU M, EVERETT M, et al. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning[C]//*2017 IEEE International Conference on Robotics and Automation*. Singapore, 2017: 285-292.

[8] 高阳, 陈世福, 陆鑫. 强化学习研究综述[J]. 自动化学报, 2004, **30**(1): 86-100.(GAO Yang, CHEN Shifu, LU Xin. A review of reinforcement learning[J]. *Journal of Automatica Sinica*, 2004, **30**(1): 86-100.(in Chinese))

[9] YUAN X. faster finding of optimal path in robotics playground using *Q*-learning with "exploitation-exploration trade-off"[J]. *Journal of Physics: Conference Series*, 2021, **1748**(2): 022008.

[10] MAOUDJ A, HENTOUT A. Optimal path planning approach based on *Q*-learning algorithm for mobile robots [J]. *Applied Soft Computing Journal*, 2020, **97**(A): 106796.

[11] 张宁, 李彩虹, 郭娜, 等. 基于 CM-*Q* 学习的自主移动机器人局部路径规划[J]. 山东理工大学学报(自然科学版), 2020, **34**(4): 37-43.(ZHANG Ning, LI Caihong, GUO Na, et al. Local path planning of autonomous mobile robot based on CM-*Q* learning[J]. *Journal of Shandong University of Technology* (*Natural Science*), 2020, **34**(4): 37-43.(in Chinese))

[12] 张福海, 李宁, 袁儒鹏, 等. 基于强化学习的机器人路径规划算法[J]. 华中科技大学学报(自然科学版), 2018, **46**(12): 65-70.(ZHANG Fuhai, LI Ning, YUAN Rupeng, et al. Robot path planning algorithm based on reinforcement learning[J]. *Journal of Huazhong University of Science and Technology* (*Natural Science Edition*), 2018, **46**(12): 65-70.(in Chinese))

[13] 王沐晨, 李立州, 张珺, 等. 基于卷积神经网络气动力降阶模型的翼型优化方法[J]. 应用数学和力学, 2022, **43**(1): 77-83.(WANG Muchen, LI Lizhou, ZHANG Jun, et al. An airfoil optimization method based on the convolutional neural network aerodynamic reduced order model[J]. *Applied Mathematics and Mechanics*, 2022, **43**(1): 77-83.(in Chinese))

[14] 高普阳, 赵子桐, 杨扬. 基于卷积神经网络模型数值求解双曲型偏微分方程的研究[J]. 应用数学和力学, 2021, **42**(9): 932-947.(GAO Puyang, ZHAO Zitong, YANG Yang. Study on numerical solutions to hyperbolic partial differential equations based on the convolutional neural network model[J]. *Applied Mathematics and Mechanics*, 2021, **42**(9): 932-947.(in Chinese))

[15] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing Atari with deep reinforcement learning[Z/OL]. 2013

[2022-03-07]. https://arxiv.org/abs/1312.5602.

[16] 董永峰, 杨琛, 董瑶, 等. 基于改进的 DQN 机器人路径规划[J]. 计算机工程与设计, 2021 ,**42**(2): 552-558.(DONG Yongfeng, YANG Chen, DONG Yao, et al. Robot path planning based on improved DQN[J]. *Computer Engineering and Design*, 2021, **42**(2): 552-558.(in Chinese))

[17] 姜兰. 基于强化学习的智能小车路径规划[D]. 硕士学位论文. 杭州: 浙江理工大学, 2019.(JIANG Lan. Intelligent car path planning based on reinforcement learning[D]. Master Thesis. Hangzhou: Zhejiang Sci-Tech University, 2019.(in Chinese))

[18] 丁志强. 基于 *Q* 学习算法的快速避障路径规划方法研究[D]. 硕士学位论文. 大连: 大连理工大学, 2021.(DING Zhiqiang. Research on fast obstacle avoidance path planning method based on *Q*-learning alorithm[D]. Master Thesis. Dalian: Dalian University of Technology, 2021.(in Chinese))

[19] FORTUNATO M, AZAR M G, PIOT B, et al. Noisy networks for exploration[Z/OL]. 2018[2022-03-07]. https://arxiv. org/abs/1706.10295.pdf.

[20] 胡刚. 基于强化学习的无地图搜索导航[D]. 硕士学位论文. 哈尔滨: 哈尔滨工业大学, 2019.(HU Gang. Mapless exploration navigation based on reinforcement learning[D]. Master Thesis. Harbin: Harbin Industrial University, 2019.(in Chinese))

[21] 王健, 赵亚川, 赵忠英, 等. 基于 $Q(\lambda)$-learning 的移动机器人路径规划改进探索方法[J].自动化与表, 2019, **34**(11): 39-41.(WANG Jian, ZHAO Yachuan, ZHAO Zhongying, et al. Improved exploration method for mobile robot path planning based on $Q(\lambda)$-learning[J]. *Automation and Instrument*, 2019, **34**(11): 39-41.(in Chinese))

[22] 吴夏铭. 基于深度强化学习的路径规划算法研究[D]. 硕士学位论文. 长春: 长春理工大学, 2020.(WU Xiaming. Research on path planning algorithm based on deep reinforcement learning[D]. Master Thesis. Changchun: Changchun University of Science and Technology, 2020.(in Chinese))

[23] 吴俊塔. 基于集成的多深度确定性策略梯度的无人驾驶策略研究[D]. 硕士学位论文. 深圳: 中国科学院深圳先进技术研究院, 2019.(WU Junta. Research of unmanned driving policy based on aggregated multiple deterministic policy gradient[D]. Master Thesis. Shenzhen: Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, 2019.(in Chinese))

[24] 于乃功, 王琛, 默凡凡, 等. 基于 *Q* 学习算法和遗传算法的动态环境路径规划[J]. 北京工业大学学报, 2017, **43**(7): 1009-1016.(YU Naigong, WANG Chen, MO Fanfan, et al. Dynamic environment path planning based on *Q*-learning algorithm and genetic algorithm[J]. *Journal of Beijing University of Technology*, 2017, **43**(7): 1009-1016.(in Chinese))

[25] 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.(ZHOU Yi, CHEN Bo. Method for obstacle avoidance based on improvement dueling Networks[J]. *Journal of Xidian University*, 2019, **46**(1): 46-50.(in Chinese))