Journal of System Simulation

Volume 33 | Issue 10 Article 17

10-18-2021

DQN-based Path Planning Method and Simulation for Submarine and Warship in Naval Battlefield

Xiaodong Huang
1. Naval Aeronautical University, Shandong 264001, China; ;

Haitao Yuan
2. School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China; ;

Bi Jing
3. Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; ;

Liu Tao
4. School of Software Engineering, Beijing Jiaotong University, Beijing 100044, China;

DQN-based Path Planning Method and Simulation for Submarine and Warship in Naval Battlefield

Abstract

Abstract: To realize multi-agent intelligent planning and target tracking in complex naval battlefield environment, the work focuses on agents (submarine or warship), *and proposes a simulation method based on reinforcement learning algorithm called Deep Q Network (DQN). Two neural networks with the same structure and different parameters are designed to update real and predicted Q values for the convergence of value functions. An ε-greedy algorithm is proposed to design an action selection mechanism, and a reward function is designed for the naval battlefield environment to increase the update velocity and generalization ability of Learning with Experience Replay (LER).* Simulation results show that compared with existing path routing algorithms and multi-agent path routing algorithms, each agent can effectively avoid obstacles in unfamiliar environments and achieve more intelligent path planning and target tracking through a certain number of steps of learning.

Keywords

Deep Q network, reinforcement learning, multiple agents, path planning, target tracking

Recommended Citation

Huang Xiaodong, Yuan Haitao, Bi Jing, Liu Tao. DQN-based Path Planning Method and Simulation for Submarine and Warship in Naval Battlefield[J]. Journal of System Simulation, 2021, 33(10): 2440-2448.

第 33 卷第 10 期 系统仿真学报© Vol. 33 No. 10
2021 年 10 月 Journal of System Simulation Oct. 2021

基于 DQN 的海战场舰船路径规划及仿真

黄晓冬[1]，苑海涛[2]，毕敬[3*]，刘涛[4]
(1. 海军航空大学，山东 烟台 264001；2. 北京航空航天大学 自动化科学与电气工程学院，北京 100191；
3. 北京工业大学 信息学部，北京 100124；4. 北京交通大学 软件学院，北京 100044)

摘要：为实现海战场环境下多智能体路径规划及目标追踪，以智能体(潜艇或者舰艇)为研究对象，*提出一种基于强化学习的深度 Q 网络算法。通过设计两个结构相同但参数不同的神经网络，分别对其 Q 实际值和估计值的更新来实现价值函数的收敛。运用 ε-贪婪算法设计动作选择机制和基于应用环境设计奖励函数，显著提高 LER 算法的更新速度和泛化能力等。*仿真结果表明，与现有的路径规划算法和多智能体路径规划算法相比，每个智能体能够在陌生环境中有效躲避障碍物，并且通过一定步数的学习实现更加高效的智能规划路线及追踪目标。

关键词：深度 Q 网络；强化学习；智能体；路径规划；目标追踪
中图分类号：TP311.5; TP391 文献标志码：A 文章编号：1004-731X (2021) 10-2440-09
DOI: 10.16182/j.issn1004731x.joss.21-0229

DQN-based Path Planning Method and Simulation for Submarine and Warship in Naval Battlefield

Huang Xiaodong[1], *Yuan Haitao*[2], *Bi Jing*[3*], *Liu Tao*[4]
(1. Naval Aeronautical University, Shandong 264001, China;
2. School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China;
3. Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China;
4. School of Software Engineering, Beijing Jiaotong University, Beijing 100044, China)

Abstract: To realize multi-agent intelligent planning and target tracking in complex naval battlefield environment, the work focuses on agents (submarine or warship), *and proposes a simulation method based on reinforcement learning algorithm called Deep Q Network (DQN). Two neural networks with the same structure and different parameters are designed to update real and predicted Q values for the convergence of value functions. An ε-greedy algorithm is proposed to design an action selection mechanism, and a reward function is designed for the naval battlefield environment to increase the update velocity and generalization ability of Learning with Experience Replay (LER).* Simulation results show that compared with existing path routing algorithms and multi-agent path routing algorithms, each agent can effectively avoid obstacles in unfamiliar environments and achieve more intelligent path planning and target tracking through a certain number of steps of learning.

Keywords: Deep Q network, reinforcement learning, multiple agents, path planning, target tracking

引言

为了模拟海战场环境中多智能体之间的协作、追踪、路径规划等行为，需要解决避障和路径规划问题。路径规划的任务是找到当前实体与目标实体之间最优的路径，而目标追踪意味着实时的调整目

收稿日期：2021-03-19 修回日期：2021-04-15
基金项目：装备预研基金(41401020401，41401050102)；国家自然科学基金(62173013，62073005，61802015)
第一作者：黄晓冬(1975-)，男，博士后、教授，研究方向计算机软件、系统建模与仿真、人工智能应用等。E-mail：3065351527@qq.com
通讯作者：毕敬(1979-)，女，博士，研究方向为计算智能、深度学习等。E-mail：bijing@bjut.edu.cn

标位置并优化路径。目前，解决单实体的路径规划方法如 Dijkstra 算法[1]、粒子群算法[2]、遗传算法[3]、蜜蜂算法等已得深入研究和应用。而多智能体之间相互追逐、相互对抗等情况下的多智能体路径规划问题尚有许多问题未能解决。比如，多智能体带来了算法规模的增大和算法复杂度的大幅增加[4]，使问题求解效率不能满足实际需求。

本文提出了一种基于强化学习的 DQN (Deep Q Network)算法[5]来解决海战场环境下多智能体路径规划的问题。

1 相关研究

1.1 卷积神经网络

现有研究工作中，卷积神经网络(Convolutional Neural Network，CNN)在计算机视觉中有着广泛应用，它使用空间信息来识别图像。通常包含三个主要层：卷积层、池化层和全连接层。文献[6]通过把奖励和动作作为输入放到 CNN 中来不断地进行迭代更新参数，最终获取了最优的参数，即获得最优的导航策略。文献[7]中将 CNN 与强化学习中 Q-Learning 算法结合，提出了 NIPS DQN (Neural Information Processing Systems Deep Q-Network)算法，通过实验证明了 DQN 算法训练得到的模型拥有能够与人类玩家进行对战的能力。文献[8]将当前智能体的状态作为深度 CNN 网络的输入，输出结果是该状态下每一个动作的 Q 值，把输出的 Q 值作为 CNN 的标签，运用梯度下降的方法，通过最小化均方误差来更新神经网络的参数。

1.2 强化学习

1.2.1 基本原理

多智能体强化学习基本原理如图 1 所示。图 1 表明多智能体处于同一环境中，并不断与环境进行交互。智能体接受环境反馈的状态信息(State)进行状态转移，接受环境反馈的奖励(Reward)来评判当前环境下采取当前动作(Action)的好坏。多智能体之间属于完全合作关系，并且其共享经验池(Experience Replay)。DQN 算法将这些历史信息放到一个 Q 表中，使用表格 Q 来存储每一个状态，以及在这个状态中的每个行为所拥有的 Q 值。而现如今问题由于较为复杂，因此在机器学习中可以通过引入神经网络来解决复杂度的难题。具体可将状态和动作当成神经网络的输入，并经过神经网络分析后得到动作的 Q 值。

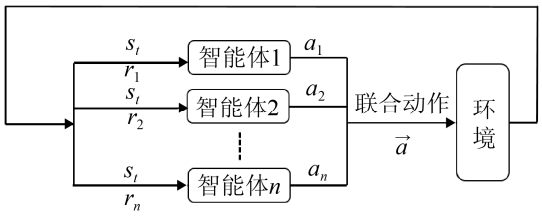


图 1 多智能体强化学习环境交互示意图
Fig. 1 Interaction of multi-agent reinforcement learning environments

强化学习最初用来描述特定刺激使生物更趋向于采用某些策略的现象。在人工智能领域内，强化学习是一类特定的机器学习问题[8-9]。强化学习系统中有几个关键，奖励、策略、价值函数和模型。某一策略下的价值函数可表示为：

$$v_\pi(s) = E_\pi[R_{t+1} + \gamma R_{t+2} + ... + | S_t = s] \tag{1}$$

模型至少要解决两个问题：

(1) 确定状态转化概率，即预测下一个可能状态发生的概率，具体为：

$$P_{ss'}^a = P[S_{t+1} = s' | S_t = s, A_t = a] \tag{2}$$

(2) 预测可能获得的即时奖励，具体为：

$$R_s^a = E[R_{t+1} | S_t = s, A_t = a] \tag{3}$$

1.2.2 马尔科夫链和贝尔曼方程

MDP (Markov Decision Process)下的状态价值函数和行为价值函数可用下一时刻状态价值函数或行为价值函数来表达：

$$v_\pi(s) = E_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s] \tag{4}$$

$$q_\pi(s,a) = E_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a] \tag{5}$$

通过将式(4), (5)结合起来，可得到式(6), (7), 即：

https://dc-china-simulation.researchcommons.org/journal/vol33/iss10/17
DOI: 10.16182/j.issn1004731x.joss.21-0229

$$v_\pi(s)=\sum_{a\in A}\pi(a\mid s)(R_s^a+\gamma\sum_{s'\in S}P_{ss'}^a v_\pi(s')) \quad (6)$$

$$q_\pi(s,a)=R_s^a+\gamma\sum_{s'\in S}P_{ss'}^a\sum_{a'\in A}\pi(a'\mid s')q_\pi(s',a') \quad (7)$$

文献[10]在 CNN 训练的基础上运用强化学习的决策能力，得出最佳导航策略。文献[11]利用强化学习中 Q-Learning 算法，结合神经网络，通过学习实现了自主避障。文献[12]在面对路径规划问题上，采用了分层强化信息的方法，将路径规划系统从上至下分为根任务协作层、子任务选择层和环境交互层三层结构，并将路径规划任务划分为静态障碍物避障、动态障碍物避障及趋向目标点运动等多个基本子任务，减小状态空间及系统学习的难度。

1.2.3 深度强化学习

强化学习作为机器学习领域的研究热点，已经广泛应用于现代工业中。强化学习拥有不错的决策能力，但是其依赖人工提取的特征，难以解决复杂困难问题。而深度学习则具有良好的提取特征的能力，能够从高维度数据提取出有用的信息，所以两者相互补充形成了深度强化学习(Deep Reinforce Learning, DRL)[8]。其中应用最广泛的深度强化学习就是 DQN，它是一种融合了神经网络和 Q-Learning 算法的方法，神经网络可解决需要存储检索大量 Q 值的问题。

文献[13]运用深度强化学习来对机械臂进行训练，通过在虚拟环境中训练实现收敛，完成特定的动作，最终达到工业生产的标准。文献[14]提出基于深度强化学习进行仿真并优化的方法，使用深度强化学习降低了系统的平均利润。文献[15]将深度学习运用在故障诊断领域，应用了长短时记忆网络建立了时间序列模型，显著提高了电源车故障诊断的准确率和可靠性。

1.2.4 多 Agent 仿真建模

多 Agent 相对于单 Agent 来说，不仅要考虑其路径规划和导航决策的问题，还需要考虑每个 Agent 之间的相互作用和制约。文献[9]通过建立多 Agent 强化学习模型以及积累经验优化策略，实现了多 Agent 自主决策。该工作基于强化学习，运用 Q-Learning 学习算法，利用多个 Agent 在环境中不断进行“探索-学习-利用”，积累经验并评估奖励和行为，最终实现多 Agent 路径规划任务。

文献[16]提出一种基于分层强化学习及神经网络的多 Agent 动态编队方法。首先，在多 Agent 动态编队中，引入“抽象机制”把整个任务分为三个任务层次（根任务协作层、动作子任务选择层和基本动作执行层）对状态空间进行降维和学习任务分解。其次，将神经网络作为状态泛化方法与分层强化学习中的 Q-Learning 方法相结合，通过状态变量的分割降级来减少空间存储量。

2 基于 DQN 的路径规划方法

2.1 DQN 网络 Q 值更新方法

DQN 网络模型的关键设计要点在于 Q 值的更新[10]，针对项目特点，本文工作通过设计两个结构相同但参数不同的神经网络，分别通过对 Q 实际值和估计值的更新来实现价值函数的收敛，其更新过程如图 2 所示。

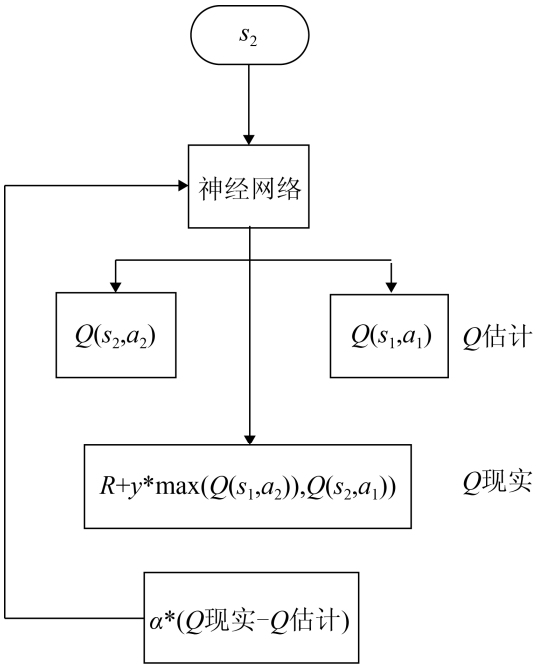


图 2 DQN 更新过程
Fig. 2 Update process of DQN

同时，DQN 使用一个记忆库来学习之前的经历，每次更新的时候，随机抽取一些之前的经历进行学习，而目标网络用两个结构相同参数不同的网络结构进行更新，使得神经网络更新更加有效率。具体的网络结构如图 3 所示。

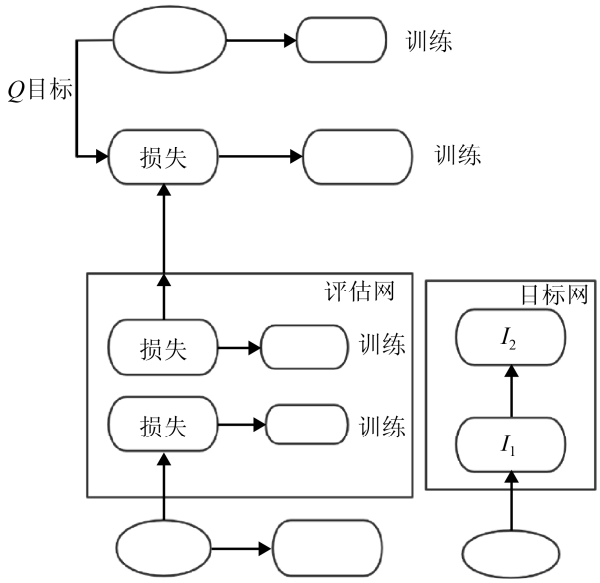


图 3 DQN 网络结构
Fig. 3 Network structure of DQN

当无法保持收敛性时，神经网络会出现训练不稳定或训练困难等问题。针对这些问题，DQN 使用经验回放和目标网络来进行改进[16]。经验回放通常用来存放收集到的观察序列(s, a, r, s')，把这一转移信息保存在经验回放的缓冲区中，缓冲区内的信息就成为智能体的经验[17]。在使用蒙特卡洛算法对一个估值进行更新时，神经网络从每一个循环开始，一直到循环结束，才会进行奖励回溯，所以会造成估计不准确的问题。而经验回放的核心思想就是利用缓存中的转移信息来训练 DQN，而不是循环结束时的信息[18]。每一次循环之间的经验是相互关联的，所以从回放缓存中随机选择的一批训练样本将会减少经验之间的相关性，并有助于增强智能体的泛化能力。在损失函数中，利用公式(8)计算 Q 估计值和 Q 现实值的均方差，具体为：

$$loss = (R + \gamma \max(Q(s', a', \theta) - Q(s, a, \theta)))^2 \quad (8)$$

公式(8)采用了相同的估计值、现实值及权重。由于用同一网络来计算预测值和目标值易使两者之间存在很大差别。因此，本文工作采用目标网络来计算目标值[19]，采用损失函数为：

$$loss = (R + \gamma \max(Q(s', a', \theta') - Q(s, a, \theta)))^2 \quad (9)$$

DQN 使用了两个结构相同但参数不同的神经网络，预测 Q 估计值的神经网络使用最新的参数，而预测 Q 实际值的神经网络使用旧参数，这样可以使训练过程更加稳定[16]。

基于 LER (Learning with Experience Replay)的更新算法具体步骤为：①初始化容量、权重、重放记忆量；②进行 Episode 次循环，以概率 ε 随机选择动作，在模拟器中观察该动作下的奖励，对结果进行存储，并且根据梯度下降计算误差，最终训练出最优的模型；③每 C 次更新一下 Q，即每隔 C 步就令 Q 更新为最新的参数。

算法 1: LER

1 将重放记忆 D 初始化为容量 N

2 用随机权重 θ 初试化动作值函数 Q

3 用权重 $\theta'=\theta$ 初始化目标动作函数 Q

4 对于每一个循环 Episode，初始化序列 $S_1 = \{x_1\}$ 和预处理序列 $\phi_1 = \phi_{s_1}$

5 t 初始化为 1，每次 t+1

6 以概率 ε 选择一个随机动作

7 否则选择 $a_t = \arg\max Q(\phi(s_t), a, \theta)$

8 在模拟器中执行动作 a_t，并观察奖励 r_t 和图像 x_{t+1}

9 设置 $S_{t+1} = S_t, a_t, x_{t+1}$ 和预处理 $\phi_{t+1} = \phi(s_{t+1})$

10 在 D 中存储转换 $(\phi_t, a_t, r_t, \phi_{t+1})$

$y_j = r_j \leftarrow (j+1)$

$y_j = r_j + \gamma \max_d \tilde{Q}(\varphi_{j+1}, d, \theta') \leftarrow otherwise$

11 从 D 中的随机小批量转换 $(\phi_t, a_t, r_t, \phi_{t+1})$

12 设置对网络参数 θ 执行梯度下降步骤 $(y_j - Q(\phi_j, a_j, \theta))^2$

13 每 C 步重置 $\tilde{Q}$ 值，$\tilde{Q} = Q$

https://dc-china-simulation.researchcommons.org/journal/vol33/iss10/17
DOI: 10.16182/j.issn1004731x.joss.21-0229

第 33 卷第 10 期 系统仿真学报 Vol. 33 No. 10
2021 年 10 月 Journal of System Simulation Oct. 2021

2.2 动作选择

强化学习类似于一个试错的学习过程，个体需要从其与环境的交互中寻找一个好的策略，同时又不至于在试错的过程中丢失太多的奖励[20]。探索表示智能体需要不断进行试错来收集更多的转移信息，而利用则是利用记忆库来做出当前状态下能做出的最佳选择，探索和利用是个体进行决策时需要平衡的两个方面。虽然探索在短期内不会带来大的奖励，但是它会扩充记忆库，这有益于获得长期内的奖励；而利用则是注重于短期奖励，它会获得当前状态下的最佳奖励，但是会导致长期整体利益受损。因此，本文结合 ε-贪婪算法($0<\varepsilon<1$)，其中，设置 ε=0.9，即 90%的情况下选择当前最佳的选择，10%则从行为集合中随机选择，进行大胆的探索。通过这样的设置，既可以保证在短期内选择当前状态下的最佳行为，又能保证进行进一步的探索，不断扩充记忆库，为长期收益打下基础。

贪婪算法在求解问题时，总是做出当前最好的选择，也就是说，每次都是某种意义上的局部最优解。在本文实验中，通过使用贪婪算法，在每一个过程中都选择当前最佳的行为。将整体问题分解为若干个子问题，对每一个子问题采用 90%的贪婪算法和 10%的随机行为选择。这样不仅能够尽快找到最佳行为，同时，也能避免贪婪算法的局限性。

2.3 奖励函数

强化学习的最终目标就是最大化奖励，而奖励函数定义了在当前环境下，智能体执行不同的行为所能获得的价值[21]。R 是一个奖励函数，S 状态下的奖励是某一时刻 t 处在状态 s 下，下一时刻 t+1 能获得的奖励期望：$R_s = E[R_{t+1} \mid S_t = s]$。而总奖励函数，也就是收获，计算公式为：

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{10}$$

式中：衰减系数 γ 体现了未来的奖励在当前时刻的价值比例。在 t+k+1 时刻获得的奖励 R 在 t 时刻体现出的价值是 $\gamma^k R$。γ 接近 0，则表明趋向于“近视”性评估；γ 接近 1 则表明偏重考虑远期的利益[16]。奖励函数值的设置会影响 LER 算法的更新速度和泛化能力等。对于避障任务的奖励值的设置，本文定义奖励函数如表 1 所示。

表 1 奖励表
Tab. 1 Table of reward setting

奖励值	情况
+1	到达目标点
–1	碰到障碍物
0	其他

多个实体的共同目的就是让所有的实体采取同一组行为并获得最大的奖励的序列，从而能够找到敌方。当智能体到达目标点时，会给智能体一个+1 的奖励；如果在寻找目标的路上碰到了障碍物，则给一个–1 的奖励，而且当前 Episode 结束，进入到下一个循环；其他情况奖励为 0。

3 实验仿真和分析

3.1 实验场景描述

为了验证算法的有效性，本文构造了一个网格化场景，如图 4 所示。场景设置了两个追逐舰艇(红方)和一个潜艇(蓝方)，并设置了 8 个障碍。与网格化场景相对应的模拟仿真场景，如图 5 所示。

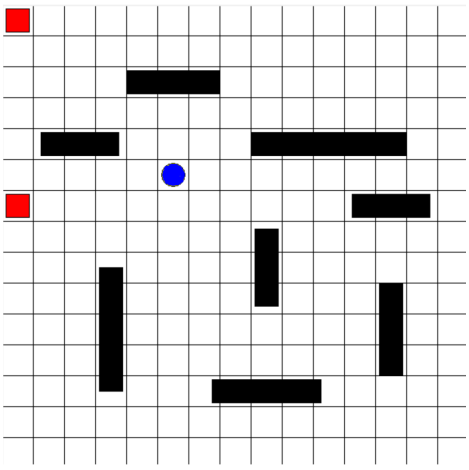


图 4 仿真环境
Fig. 4 Simulation scenario

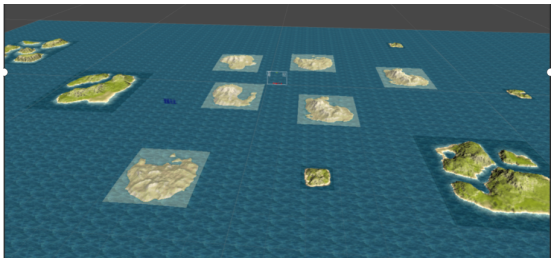


图 5 实际场景
Fig. 5 Real-life scenario

本文模拟软件使用的实验环境，如表 2 所示。

表 2 实验环境配置表
Tab. 2 Table of experimental setting

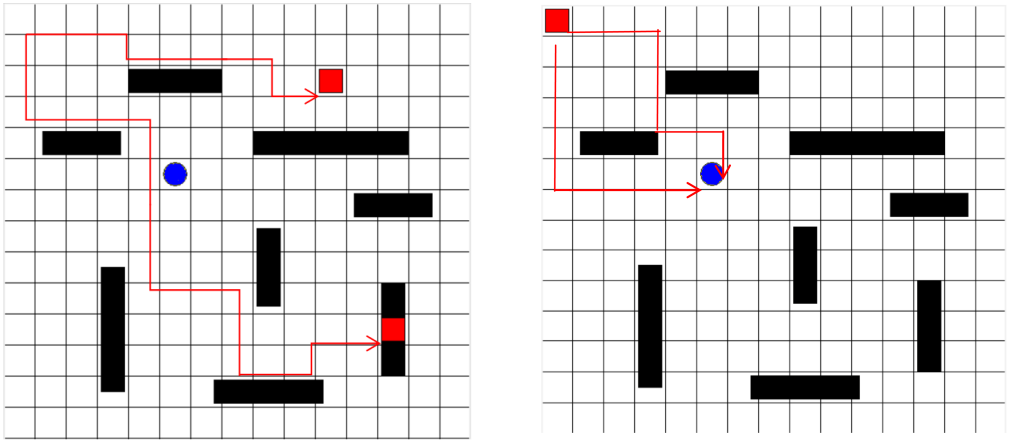
名称	版本型号
CPU	CORE i5 7TH Gen
内存	8G
GPU	NVIDIA GETFORCE 930MX
系统	Windows 10
Tensorflow-GPU	1.4

3.2 实验结果

基于上述场景和实验环境，仿真实验每次都进行 3000 次的 Episode，并且每隔 200 次替换目标网络中相对滞后的参数。在初始化之后，一开始不训练，先积累一些记忆。然后，算法根据观测值选择行为，强化学习环境给出下一个状态和奖励，DQN 中的神经网络开始储存这些记忆。每个智能体根据记忆库中的信息进行学习。起初可能因为记忆库中信息较少，此时智能体还不具备智能。所以舰艇会不断进行探索，也可能会撞上障碍物。此外，路径也可能较为杂乱，无法找到最短路径。具体如图 6(a)所示。

经过一段时间记忆的累计和不断的学习，智能体能够较好地进行避障，并能够学习到距离目标最短的路径。一旦智能体能够寻找到最短路径后，以后的每一个 Episode 它都会按照这个最短路径去搜寻目标，直到有更短的路径被探索出来。针对本文的实验环境，在进行了一段时间的学习之后，会将 ε-贪婪值调整变小，相应的利用会增加，这样能够更准确地找到当前状态下的最优解。所以此时观察到智能体的路径能够在避开障碍物的情况下，直接快速地找出最短路径，如图 6(b)所示。

为了观察智能体在多个 Episode 中的学习状况，本文输出了学习过程中的代价曲线，如图 7 所示。通过图 7 可以看出曲线并不是平滑下降的，这是因为 DQN 中的输入数据不是一成不变，而是随着迭代过程一步步改变的，而且会根据学习情况，获取到不同的数据。图 7 中横坐标代表的是训练的次数，纵坐标代表的函数的代价变化，由于数据的变化，代价函数并不是一直递增的状态。



(a) 早期记忆库作用下的智能体轨迹 (b) 丰富记忆库作用下的智能体寻优轨迹

图 6 不同记忆库作用下的智能体寻优轨迹
Fig. 6 Search trajectories of agents with different memory libraries

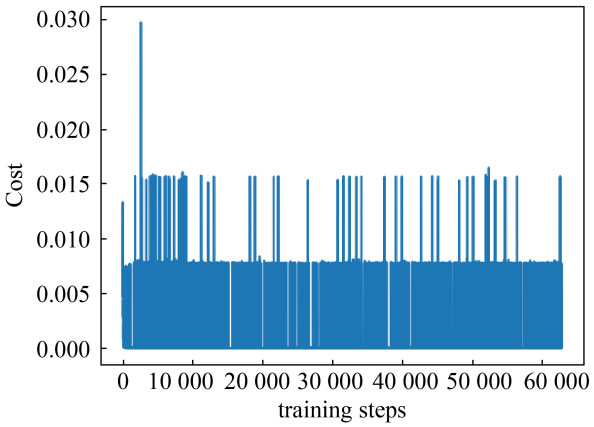


图 7 代价变化曲线
Fig. 7 Curve of cost variations

本文所提的强化学习与 Sarra 和 Actor-Critic 算法进行对比，图 8 为三种算法的累计 Reward 对比结果。

可以看出，本文使用的 Q-learning 融合神经网络的算法达到了最好的结果。此外，由于 ε-贪婪值的存在，累计总奖励值无法持续增长，并出现负值。将本文给出的方法与经典的避障路径规划[12]、多 Agent[22]路径规划方法总探索规划步数进行对比，如图 9 所示。

由图 9 可以看出，本文采用的算法所用的探索步数明显少于其他两种算法，这表明了本文提出的算法具有更高的避障和路径规划能力。

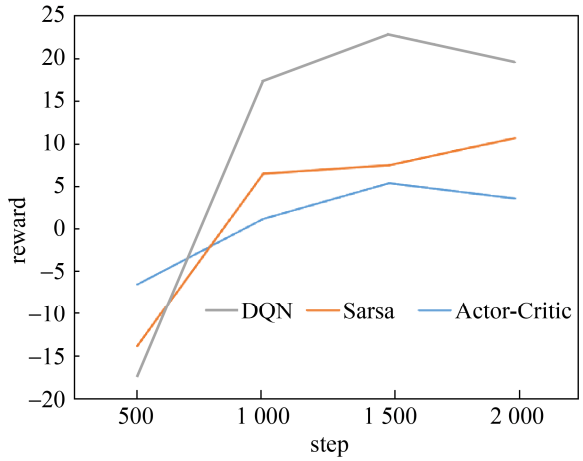


图 8 三种算法 reward 对比图
Fig. 8 Reward comparison of three algorithms

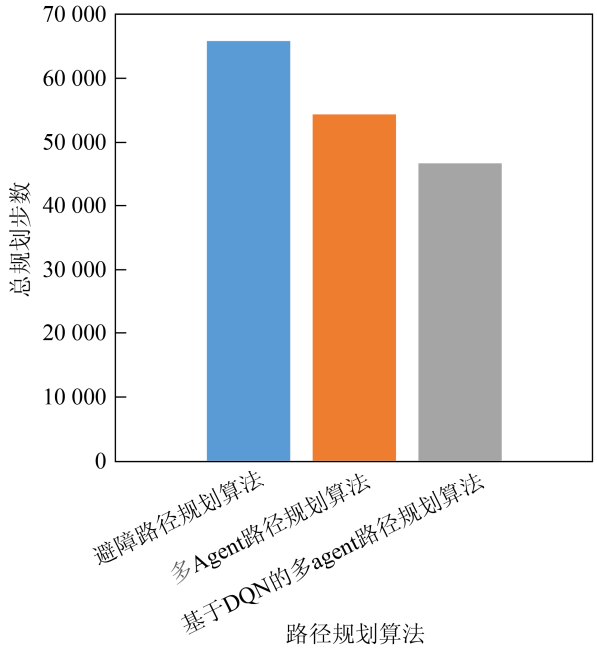


图 9 三种算法总探索规划步数对比图
Fig. 9 Number of total exploration steps of three algorithms

4 结论

本文提出了一种海战场环境下基于强化学习的深度 Q 网络算法(DQN)，并给出了多智能体路径寻优、避障和跟踪的策略方法。仿真实验表明，与现有的避障路径规划算法和多智能体路径规划算法相比，本文所提算法仅需要较少的总探索规划步数即可达到相同目标。可以证明，本文所提算法在海战场环境下具有更高的智能路径规划、避障及追踪能力。

参考文献：

[1] 姜涛, 王建中, 施家栋. 小型移动机器人自主返航路径规划方法[J]. 计算机工程, 2015, 41(1): 164-168.
Jiang Tao, Wang Jianzhong, Shi Jiadong, Autonomous Return Path Planning for Small Mobile Robots[J]. Computer Engineering, 2015, 41(1): 164-168.

[2] 刘洁, 赵海芳, 周德廉. 一种改进量子行为粒子群优化算法的移动机器人路径规划[J]. 计算机科学, 2017, 44(增 2): 123-128.
Liu Jie, Zhao Haifang, Zhou Delian. Path Planning of Mobile Robot Based on Improved Quantum Behavior Particle Swarm Optimization[J]. Computer Engineering,

2017, 44(S2): 123-128.

[3] 赵晓, 王铮, 黄程侃, 等. 基于改进 A*算法的移动机器人路径规划[J]. 机器人, 2018, 40(6): 903-910.
Zhao Xiao, Wang Zheng, Huang Chengkan, et al. Path Planning of Mobile Robot Based on Improved A* Algorithm[J]. Robot, 2018, 40(6): 903-910.

[4] 郭鹏, 余建波. 基于深度强化学习的制造过程 Run-to-Run 控制[[J/OL]]. 自动化学报, [2021-02-06]. https://doi.org/10.16383/j.aas.c190546.
Guo Peng, Yu Jianbo, Run-to-Run Control of Manufacturing Process Based on Deep Reinforcement Learning[J]. Acta Automatica Sinica, [2021-02-06]. https://doi.org/10.16383/j.aas.c190546.

[5] Hasselt H V, Guez A, Silver D. Deep Reinforcement Learning with Double Q-learning[C]// Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16). Arizona, USA: AAAI, 2016: 2094-2100.

[6] 王大方. 基于深度强化学习的机器人导航研究[D]. 徐州: 中国矿业大学, 2019.
Wang Dafang. Research on Robot Navigation Based on Deep Reinforcement Learning[D]. Xuzhou: China University of Mining and Technology, 2019.

[7] 邓悟. 基于深度强化学习的智能体避障与路径规划研究与应用[D]. 成都: 电子科技大学, 2019.
Deng Wu. Research and Application of Agent Obstacle Avoidance and Path Planning Based on Deep Reinforcement Learning[D]. Chengdu: University of Electronic Science and Technology, 2019.

[8] 江其洲, 曾碧. 基于深度强化学习的移动机器人导航策略研究[J]. 计算机测量与控制, 2019, 27(8): 217-221.
Jiang Qizhou, Zeng Bi. Research on Navigation Strategy of Mobile Robot Based on Deep Reinforcement Learning[J]. Computer Measurement and Control, 2019, 27(8): 217-221.

[9] 张心怡, 张智鹏, 张铁赢, 等. RLO: 一个基于强化学习的连接优化方法[J]. 中国科学: 信息科学, 2020, 50(5): 637-648.
Zhang Xinyi, Zhang Zhipeng, Zhang Tieying, et al. RLO: a Reinforcement Learning-based Method for Join Optimization[J]. Scientia Sinica (Informationis), 2020, 50(5): 637-648.

[10] 乔俊飞, 侯占军, 阮晓钢. 基于神经网络的强化学习在避障中的应用[J]. 清华大学学报(自然科学版), 2008(增 2): 1747-1750.
Qiao Junfei, Hou Zhanjun, Ruan Xiaogang. Application of Reinforcement Learning Based on Neural Network in Obstacle Avoidance[J]. Journal of Tsinghua University (Science and Technology), 2008(S2): 1747-1750.

[11] 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.
Wang Yiran, Jing Xiaochuan, Tian Tao, et al. Research on Multi-agent Path Planning Method Based on Reinforcement Learning[J]. Multi-agent Path Planning based on Reinforcement Learning, 2019, 36(8): 165-171.

[12] 高慧. 基于强化学习的移动机器人路径规划研究[D]. 成都: 西南交通大学, 2016.
Gao Hui. Mobile Robot Path Planning Based on Deep Reinforcement Learning[D]. Chengdu: Southwest Jiaotong University, 2016.

[13] 李鹤宇, 赵志龙, 顾蕾, 等. 基于深度强化学习的机械臂控制方法[J]. 系统仿真学报. 2019, 31(11): 2452-2457.
Li Heyu, Zhao Zhilong, Gu Lei, et al. Robot Arm Control Method Based on Deep Reinforcement Learning[J]. Journal of System Simulation, 2019, 31(11): 2452-2457.

[14] 周建频, 张姝柳. 基于深度强化学习的动态库存路径优化[J]. 系统仿真学报. 2019, 31(10): 2155-2163.
Zhou Jianpin, Zhang Shuliu. Dynamic Inventory Path Optimization Based on Deep Reinforcement Learning[J]. Journal of System Simulation, 2019, 31(10): 2155-2163.

[15] 闫丰亭, 贾金原. 基于深度学习序贯检验的电源车故障诊断方法[J]. 系统仿真学报, 2019, 31(1): 16-26.
Yan Fengting, Jia Jinyuan. Power Vehicle Fault Diagnosis Method based on Deep Learning Sequential Inspection [J]. Journal of System Simulation, 2019, 31(1): 16-26.

[16] 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.
Liu Jianwei, Gao Feng, Luo Xionglin. Survey of Deep Reinforcement Learning Based on Value Function and Policy Gradient[J]. Chinese Journal of Computers, 2019, 42(6): 1406-1438.

[17] 徐志雄, 曹雷, 陈希亮, 等. 基于强化学习的无人坦克对战仿真研究[J]. 计算机工程与应用, 2018, 13(8): 266-272.
Xu Zhixiong, Cao Lei, Chen Xiliang. Research on the Simulation of Unmanned Tank Battle based on Reinforcement Learning[J]. Computer Engineering and Application, 2018, 13(8): 266-272.

第 33 卷第 10 期 系统仿真学报 Vol. 33 No. 10
2021 年 10 月 Journal of System Simulation Oct. 2021

[18] 刘全, 闫岩, 朱斐, 等. 一种带探索噪音的深度循环 Q 网络[J]. 计算机学报, 2019, 42(7): 1588-1604.
Liu Quan, Yan Yan, Zhu Fei, et al. A Deep Recurrent Q Network with Exploratory Noise[J]. Chinese Journal of Computers, 2019, 42(7): 1588-1604.

[19] Sutton R S, Barto A G. Reainforment Learning: An Introduction[M]. Cambridge: The MIT Press, 1998.

[20] 何柳柳, 杨羊, 李征, 等. 面向持续集成测试优化的强化学习奖励机制[J]. 软件学报, 2019, 30(5): 1438-1449.
He Liuliu, Yang Yang, Li Zheng, et al. Reward of Reinforcement Learning of Test Optimization for Continuous Integration[J]. Journal of Software, 2019, 30(5): 1438-1449.

[21] 杜威, 丁世飞. 多智能体强化学习综述[J]. 计算机科学, 2019, 46(8): 1-8.
Du Wei, Ding Shifei. Overview on Multi-agent Reinforcement Learning[J]. Computer Science, 2019, 46(8): 1-8.

[22] 李波. 基于分层强化学习的多 Agent 路径规划与编队方法研究[D]. 新乡: 河南师范大学. 2016.
Li Bo. Research on Multi-agent Path Planning and Formation Method based on Hierarchical Reinforcement Learning[D]. Xinxiang: Henan Normal University, 2016.

