https://doi.org/10.1051/jnwpu/20244230396

深度 Q 网络在月球着陆任务中的性能评估与改进

岳颀[1], 石伊凡[1], 褚晶[1], 黄勇[2]

(1.西安邮电大学 自动化学院, 陕西 西安 710121; 2.西北工业大学 航天学院, 陕西 西安 710072)

摘 要:基于深度 Q 网络(DQN)技术的强化学习方法得到越来越广泛的应用,但该类算法的性能深受多因素影响。文中以月球登陆器为例,探讨不同超参数对 DQN 性能的影响,在此基础上训练得到性能较优的模型。目前已知 DQN 模型在 100 个测试回合下平均奖励为 280+,文中模型奖励值可达到 290+,并且通过在原始问题中引入额外的不确定性测试验证了文中模型的鲁棒性。另外,引入模仿学习的思想,基于启发式函数的模型指导方法获取演示数据,加快训练速度并提升性能,仿真结果证明了该方法的有效性。

关 键 词:深度强化学习;深度 Q 网络;模仿学习

中图分类号:TP181 **文献标志码**:A **文章编号**:1000-2758(2024)03-0396-10

深度强化学习[1]是结合深度学习[2]和强化学习的一种方法,旨在打破传统强化学习在高维度场景下的局限性。然而,深度强化学习也面临诸多挑战[3]。由于深度强化学习涉及大量超参数的选择和训练的不稳定性,如过拟合、前期探索缓慢和探索-利用平衡等问题,如何以高性能解决问题仍然是深度强化学习中需要解决的一大难题。

深度 Q 网络[4](deep Q network,DQN)首次将深度学习技术引入强化学习领域,在深度强化学习领域具有里程碑意义,通过使用深度神经网络来近似 Q 函数,处理高维度输入,如原始像素数据,从而能够在图像等复杂输入上学习到更为复杂的策略。DQN 作为当前研究热点,在多个领域均得到应用与发展[5-6]。虽然 DQN 在很多任务上取得了显著的成果,但它也存在以下局限性:①高样本复杂度[7];②基于离散动作空间[8];③采样效率低[9];④对超参数敏感[9]。

对于上述问题,相关人员提出相对应的解决方法:针对高样本复杂度问题,使用更加高效的训练策略,例如通过结合模仿学习[10]与强化学习算法提高模型的训练效率和性能。深度确定性策略梯度(deep deterministic policy gradient,DDPG)算法被用于解决连续动作空间问题[11],通过引入确定性策略网络和动作值函数网络,实现在连续动作空间中进行高效训练和优化。针对采样效率低问题,Zhai 等[12]提出优先采样等方法,以提高采样的效率和样本的利用效率。上述方法对于模型性能有一定程度提升,然而,模型最为依赖的是各种超参数[13],超参数塑造了模型的架构和训练策略,精心选择适当的超参数不仅能够加速模型的收敛速度,而且能够显著提升最终模型性能。为了寻找最佳超参数组合,早期采用的调优方法如网格搜索和随机搜索[14]虽然直观,但在大型超参数空间尝试过程中存在计算成本高的问题。近年来,元启发式方法在超参数优化领域崭露头角,取得了显著的进展。特别值得关注的是进化算法和遗传算法等经典元启发式方法,它们在超参数优化中逐渐彰显强大的工具性。具体而言,针对 DQN 的超参数优化,研究者们成功地应用这些元启发式方法[15],演化出了在不同问题上表现卓越的超参数组合,显著提升了模型的性能;同时,随着学习任务的不断增多,研究者们在资源有限的情况下更趋向于基于历史经验[16]进行超参数的初步选取,以更有效地利用有限资源进行模型调优。在现有的工作中,结合系统全面地选择 DQN 模型超参数并验证其鲁棒性的研究较少,同时也缺少针对

收稿日期:2023-05-30

基金项目:国家自然科学基金(61703336)与陕西省自然科学基金(2023-JC-QN-0727)资助

作者简介:岳颀(1981—),副教授

通信作者:岳颀(1981—) e-mail:yueqi6@163.com

具体问题引入模仿学习进一步提升 DQN 模型性能的论述。

1　问题描述

本文的目标是分析影响 DQN 模型的因素,同时对模型的性能加以优化。通过求解 OpenAI Gym 开发的 Box2d 模拟器下的月球着陆器问题(见图 1),对 DQN 模型进行研究。该环境的目标是训练智能体高效、快速、安全地降落在“月球”表面随机产生的区间内,即黄色旗帜所标示的区域。在这一环境中,着陆器的控制是通过调整其引擎的推力和方向来实现的。同时,着陆器还配备了可伸缩的腿部,在着陆过程中发挥减缓冲击和稳定着陆器的作用。

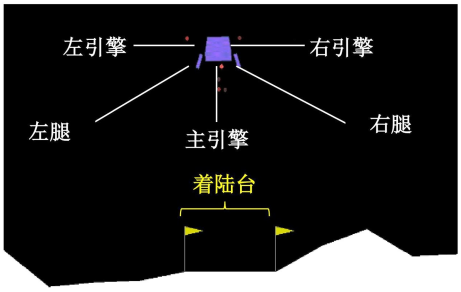


图 1　月球着陆器环境

1.1　状态空间和动作空间

一个 8 维向量构成月球着陆器的连续状态空间,具体包括:水平坐标、垂直坐标、水平速度、垂直速度、角度、角速度以及 2 个布尔值分别表示腿 1 和腿 2 是否接触月面,如(1)式所示。

$$\boldsymbol{s} = (x, y, v_x, v_y, \theta, \omega, l_{\text{left}}, l_{\text{right}}) \tag{1}$$

本文采用离散的动作空间,动作空间分为 4 个可用控制指令:不采取任何动作($a = 0$)、启动主引擎($a = 2$)、启动左侧方向引擎($a = 1$)和启动右侧方向引擎($a = 3$)。引擎的每次启动会导致着陆器的状态发生变化,智能体通过分析环境回传的状态变量,自行判断采取恰当的动作完成任务。

1.2　奖励机制

合理有效的奖励函数设置将激励智能体不断调整动作,降低与着陆台之间的距离,平稳着陆,追求奖励的最大化。着陆器从屏幕顶部移动到着陆平台的奖励取决于着陆器的实时位置。具体来说,t 时刻的奖励定义为

$$r = r_{s_t} + r_{a_t} + r_{\text{lander}} \tag{2}$$

$$r_{s_t} = -100 \times (p_t - p_{t-1}) - 100 \times (v_t - v_{t-1}) - 100 \times |\theta_t - \theta_{t-1}| + 10 \times D_{\text{landed}(t)} \tag{3}$$

$$r_{a_t} = \begin{cases} 0 & \text{不点火} \\ -0.03 & \text{启动左 / 右引擎} \\ -0.3 & \text{启动主引擎} \end{cases} \tag{4}$$

$$r_{\text{lander}} = \begin{cases} -100 & \text{着陆器坠毁} \\ 0 & \text{着陆器处于正常降落过程} \\ +100 & \text{着陆器静止在着陆平台上} \end{cases} \tag{5}$$

奖励函数包括三部分:状态奖励 r_{s_t}、动作奖励 r_{a_t} 以及来自于着陆器的奖励 r_{lander}。对于状态奖励,p 为着陆器相对于着陆台的位置,v 和 θ 分别代表智能体的速度和角度,$D_{\text{landed}(t)}$ 是 Boolean 状态值的线性组合,表示着陆器的腿部是否接触着陆台。每一次动作选择都对应着一定的奖励,体现在动作奖励函数上。针对着陆器,当处于坠毁或完全静止时,此次训练结束,触发额外奖励 r_{lander}。鉴于上述奖励函数,鼓励着陆器减少与着陆台之间的距离,降低速度来平稳着陆,减少角速度防止侧翻,同时保证着陆器抵达着陆台不再起飞,追求奖励最大化。

若用奖励值来衡量实验的结果,那该环境下的目标是:如果在连续 100 次的着陆尝试中,平均得分达到 200 分或更高,则认为问题得到解决。

1.3　相关原理

DQN 是深度学习与 Q-Learning 的结合。Q-Learning 的更新规则为

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \tag{6}$$

式中:α 为学习率;γ 为折扣因子。α 用于控制每次更新网络权重时的步长,γ 影响智能体对未来奖励的重视程度。DQN 基于 Q-Learning 使用奖励来构造目标,目标值为

$$y = r_t + \gamma \max a_{t+1} Q(s_{t+1}, a_{t+1}, \omega^*) \tag{7}$$

损失函数定义为

$$J_{\text{DQ}}(Q) = \text{E}[(y - Q(s_t, a_t, \omega))^2] \tag{8}$$

ω 和 ω^* 表示网络权重,通过损失函数对神经网络进行训练,使其逐步收敛至最优状态。

为了提高 DQN 的训练稳定性,3 项关键技术被引入:①经验回放通过构建回放缓冲区,存储并随机采样多种策略的经验数据,有效打散相关性,稳定训练过程;②目标网络技术通过构建与原评估网络完全相同的网络,在网络更新中只调整评估网络权重,定期将评估网络的权重复制给目标网络,这种定期

更新机制有效避免了学习过程中的级联反应和不稳定性问题;③探索-利用平衡技术通过策略从“强探索弱利用”到“弱探索强利用”的渐进调整,合理平衡了探索和利用,提高了智能体学习能力。

模仿学习是一种解决强化学习中样本效率、探索和安全性问题的监督式学习方法。通过利用多样性的专家示范,如轨迹数据和演示视频,以提高智能体性能。广泛应用于自动驾驶、机器人学、自然语言处理和游戏等领域。

2 DQN 双阶段:超参调优与模仿学习

本文应用融合 DQN 方法与模仿学习的双阶段训练框架。首先提出框架方法的总体结构,介绍该架构的工作流程;然后在深度强化学习阶段,基于 DQN 算法理论介绍影响其性能的超参数,并提出验证具有优化超参数模型鲁棒性的设想;最后,在模仿学习阶段,引入启发式搜索方法获取演示数据集及其训练过程。

2.1 总体框架

本文方法的总体架构如图 2 所示。利用价值网络来评估当前状态的价值,整个训练过程分为 2 个阶段:深度强化学习和模仿学习。

首先,基于速度、稳定性以及奖励值 3 个性能指标,在深度强化学习阶段全面分析每一个超参数对性能的影响,通过这种方式找到一个具备优化超参数的模型,并验证其鲁棒性。其次,为了提高训练效率,减少探索的盲目性,引入模仿学习思想:第一步,将启发式函数设计为一种生成演示数据的方法,多次运行保存轨迹元组作为演示数据;第二步,利用演示数据集训练价值网络 1,需要注意的是:价值网络 1、价值网络 2 和目标网络拥有相同的结构,通过这种方式,价值网络 1 可以学习到与启发式函数相同的行为策略;第三步,利用学习到的价值网络 1 对价值网络 2 和目标网络进行初始化;第四步,采用 DQN 方法逐步优化策略。

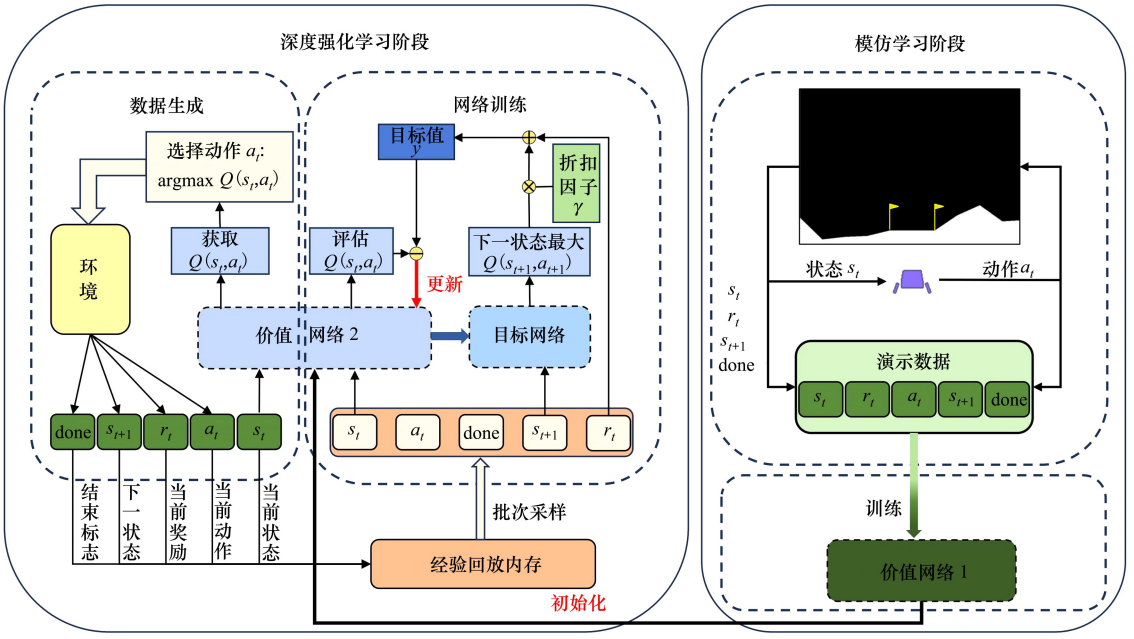


图 2 整体训练结构

2.2 深度强化学习阶段

为了成功应用 DQN,必须仔细选择和调整多个关键超参数,这些超参数对于模型的训练和性能至关重要;同时训练出来的模型要具备很强的鲁棒性。

2.2.1 参数选取

本文中探讨了多个关键超参数对于 DQN 模型性能的影响。这些超参数包括神经网络规模、经验回放内存大小、数据采样大小、学习率、折扣因子、探索-利用平衡参数以及目标网络更新频率。

①神经网络规模是一个关键的超参数,网络深度和宽度都是优化性能的重要因素。网络深度是指网络中包含的隐藏层数量,网络宽度是指隐藏层中神经元数量。②经验回放内存大小是决定训练数据多少和多样性的超参数。内存大小会对训练时间和

计算资源造成影响,选择合适的经验回放内存通常需要在实际应用中进行调试和优化。③数据采样大小影响了估计的精度和可信。较小的数据采样值可能会导致参数更新不稳定。然而,较大的数据采样值会占用更多的内存和计算资源。④学习率 α 控制了模型在每一次更新中对于梯度的调整幅度。α 设置得过大,梯度更新幅度会增加,之前训练得到的有用信息可能会被快速覆盖,导致训练效果较差。⑤折扣因子 γ 决定了模型对未来奖励的重视程度;当 γ 接近 1 时,算法更加关注长期奖励;而当 γ 接近 0 时,算法更加关注即时奖励。通过调节 γ,可以平衡智能体在长期目标和即时反馈之间的权衡。⑥ 探索 - 利用平衡参数通常用 ε 表示,负责平衡智能体在学习过程中的探索和利用策略。在实际应用中,为了完成从“强探索弱利用” 到“弱探索强利用” 的过渡,本文额外引入 2 个参数:$\varepsilon_{\text{decay}}$,$\varepsilon_{\text{final}}$。每一次训练结束对 ε 进行衰减:$\varepsilon = \varepsilon \cdot \varepsilon_{\text{decay}}$。同时,初始的 ε,即 $\varepsilon_{\text{start}}$ 仍需要调整设置。⑦目标网络更新频率直接影响到训练的稳定性和速度。频繁的更新可能导致训练不稳定,而较慢的更新可能导致训练收敛缓慢。

2.2.2　模型鲁棒性研究

一个鲁棒的模型更能适应未知的数据,并具有更好的泛化能力,从而能更可靠地应用于实际问题中。真实的外太空环境存在许多不确定因素,例如漂浮的陨石、磁场紊乱区等,这是着陆器降落过程不能经过的区域,即“禁飞区域”。

通过定制环境并创建障碍物模拟“禁飞区域”,如图 3 所示,借此分析其如何影响模型的性能。引入“禁飞区域”后,不仅需要实现月球着陆器的安稳降落,同时要远离“禁飞区域”,这是一个多目标优化问题。为了评估“禁飞区域”规避的有效性并确定奖励,基于“禁飞区域”信息创建规避函数。

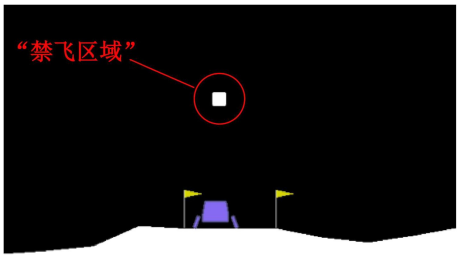


图 3　自定义的“禁飞区域”

当前“禁飞区域” 与着陆器之间的距离反映规避质量,并将其作为输入,然后输出奖励值。“禁飞区域”规避的奖励函数定义为

$$r_{AO} = \begin{cases} W_{AO} & d_{AO} \in [a,b] \\ 0 & \text{其他} \end{cases} \tag{9}$$

式中:d_{AO} 表示“禁飞区域” 与着陆器之间的距离;W_{AO} 表示奖惩值,当且仅当该区域与着陆器之间的距离属于 $[a,b]$ 时才触发奖罚机制。

2.3　模仿学习阶段

通过利用演示数据,可提高深度强化学习的效率,并帮助智能体在复杂任务中学习到安全且高效的行为策略。

2.3.1　演示数据集获取

本文利用启发式函数训练智能体。启发式函数是一类用于问题求解的函数,运用基于经验和启发性的方法,为问题提供迅速且近似的解决方案。针对本文研究,该函数接收环境和状态作为输入,返回启发式值动作为输出。借助月球着陆器的样本示例,根据对着陆器期望行为的经验和启发式规则,通过计算目标角度 T_{angle} 和目标悬停高度 T_{hover}(见(10)~(12)式),为着陆器提供一个启发性的目标值,以指导其下一步的行动选择。

$$T_{\text{angle}} = s[0] \times 0.5 + s[2] \times 1.0 \tag{10}$$

$$T_{\text{angle}} = \begin{cases} 0.4 & T_{\text{angle}} > 0.4 \\ -0.4 & T_{\text{angle}} < -0.4 \end{cases} \tag{11}$$

$$T_{\text{hover}} = 0.55 \times |s[0]| \tag{12}$$

计算 T_{angle} 和 T_{hover} 时,需基于启发式规则和系数,这些规则和系数根据经验或先验知识设定。

根据当前状态和期望行为的差距,计算出角度调整量 A_{todo} 和垂直调整量 H_{todo},见(13) ~ (14) 式。它们告诉着陆器在下一步应该如何调整自身的角度和垂直位置以接近期望状态。

$$A_{\text{todo}} = \begin{cases} 0 & s[6] \text{ 或 } s[7] \text{ 为真} \\ (T_{\text{angle}} - s[4]) \times 0.5 - (s[5]) \times 1.0 & \text{其他} \end{cases} \tag{13}$$

$$H_{\text{todo}} = \begin{cases} -(s[3]) \times 0.5 & s[6] \text{ 或 } s[7] \text{ 为真} \\ (T_{\text{hover}} - s[1]) \times 0.5 - (s[3]) \times 0.5 & \text{其他} \end{cases} \tag{14}$$

通过计算得到的 A_{todo} 和 H_{todo} 来确定最终的行动 a,见(15) 式。基于启发式函数选择动作,记录着陆器下降轨迹,得到演示数据,用于之后的模仿学习阶段。

$$a=\begin{cases}2 & H_{\text{todo}}>0.05 \text{ 且 } H_{\text{todo}}>|A_{\text{todo}}| \\ 1 & A_{\text{todo}}>0.05 \\ 3 & A_{\text{todo}}<-0.05 \\ 0 & \text{其他}\end{cases} \tag{15}$$

2.3.2 训练过程

通过上述方法设计得到演示数据集,在模仿学习训练阶段,智能体从演示数据集进行小批量采样,设计应用 2 种损失来更新网络:一步 DQN 损失和监督分类损失。监督分类损失用于对演示者的动作进行分类,一步 DQN 损失用于更新神经网络的参数,该损失函数旨在最小化预测的 Q 值与目标 Q 值(见(8)式)之间的差异,这有助于传播演示数据值,从而更好地进行模仿学习。因为演示数据只覆盖了部分状态空间且没有采取所有可能的行动,即存在许多的状态-行动从未被探索过,也没有相关数值来支撑,所以本文自行设计添加一个大边际分类损失

$$J_{\text{E}}(Q)=\max[Q(s,a)+l(a_{\text{E}},a)-Q(s,a_{\text{E}})] \tag{16}$$

式中:a_{E} 是演示数据中状态 s 下的动作;$l(a_{\text{E}},a)$ 是边际函数,当 $a_{\text{E}}=a$ 时为 0,否则为正。这样处理的目的是确保模型能够学习到合理的动作值函数,其他动作的价值至少比演示者行动的价值低一些,并尽可能减少对不可见行为的无理估计。若仅使用监督损失而不使用 DQN 损失可能会限制模型的优化及探索能力,容易受到专家演示限制。因此在模仿学习训练阶段,本文用于更新网络的总损失是这 2 种损失的组合,即

$$J(Q)=J_{\text{DQ}}(Q)+J_{\text{E}}(Q) \tag{17}$$

先基于演示数据集进行模仿训练,接下来实现 DQN 来优化模仿学习后的行为策略。

3 实 验

在本节中给出并讨论了各种仿真结果,分析并优化 DQN 性能。

3.1 参数选取

初始值是基于经验设定的:模型隐层个数为 2,分别有 256 和 128 个神经元,经验回放区内存 S_{memory} 为 65 536,样本采样大小 S_{sample} 为 32,目标网络更新频率 S_{update} 为 1,学习率 α 为 0.000 1,折扣因子 γ 为 0.99,探索-利用平衡参数 $\varepsilon_{\text{start}}=1$,$\varepsilon_{\text{decay}}=0.998$,$\varepsilon_{\text{final}}=0$。

3.1.1 网络规模选择

针对本文任务,分别设计隐层层数为 1,2,3。在第一种情况下,神经元的数量被设计为 64,128 或 256。第二种情况下神经元数量为 128/64 和 256/128。第三种情况下神经元数量为 128/64/32 和 256/128/64。训练时,以 Relu 作为激活函数。本文记录了训练中最后 100 回合的平均奖励,以及每个模型在测试时的平均奖励,所有网络训练和测试的平均奖励用图 4 的小提琴图表示。

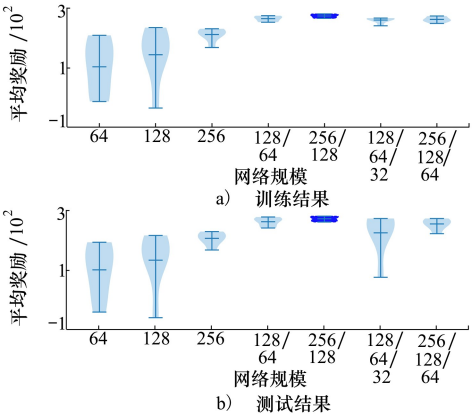
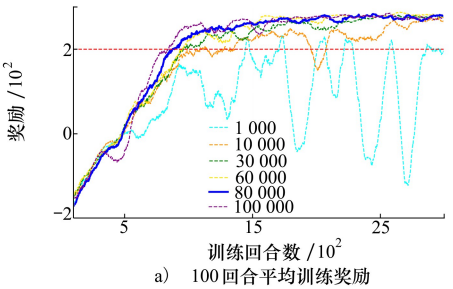


图 4 网络规模与模型性能之间的关系

分析图 4,具有大量神经元的网络结构性能优于具有隐层多的结构。此外,本文还对包含 512/256 个神经元的 2 个隐层结构进行实验,与 256/128 个神经元相比,性能提升不明显,且计算成本呈指数级增长。因此,神经元个数为 256/128 的网络结构性能是较优的。

3.1.2 经验回放内存大小选择

本文任务的经验回放内存调优如图 5 所示。本文分别对内存大小为 1 000,10 000,30 000,60 000,80 000,100 000 的模型进行训练与测试。对图 5b)分析,$S_{\text{memory}}=80\ 000$ 时模型性能较优。较大的内存可以提供更好的样本多样性和覆盖性,但会导致经验更新的延迟且使样本之间的相关性增加,进而降低模型性能。



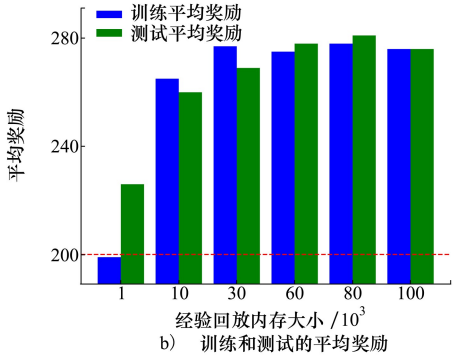


图 5　经验回放内存大小与模型性能关系

3.1.3　数据采样大小选择

为了进行更全面的比较,图 6 分别记录采样值为 4,8,16,32,48 和 64 时模型的训练和测试结果。分析图 6,每个模型都可以完成给定的目标,但存在细微的差异。数据采样大小并没有很强的影响,只要不是太小;过大的数据采样值意味着更多的训练,这会导致网络过拟合,当数据采样值为 32 时,模型性能较优。

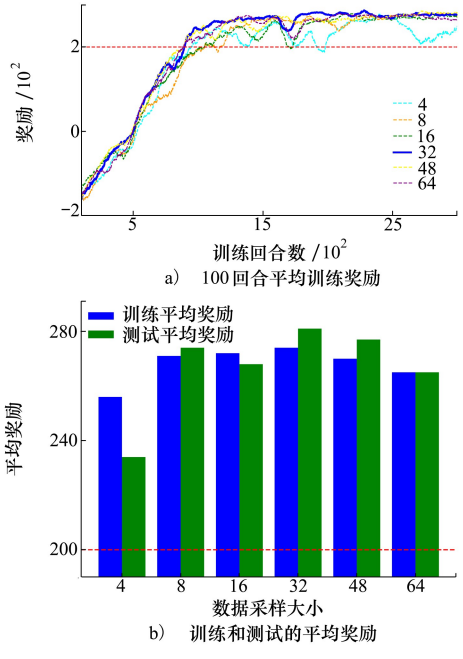


图 6　数据采样值与模型性能之间的关系

3.1.4　学习率 α

本文设置 5 个较小的学习率值,见图 7。通过比对分析可知,较小的学习率有助于让模型更稳定,综合来看,当 $\alpha=0.000\ 1$ 时模型性能较优。

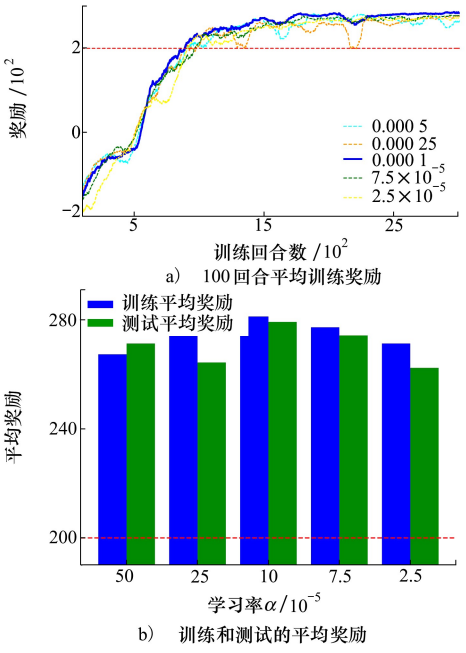


图 7　学习率 α 与模型性能之间的关系

3.1.5　折扣因子 γ

针对 γ,分别取 $\gamma=0,0.5,0.9,0.99,0.996,1$,实验结果如图 8 所示。

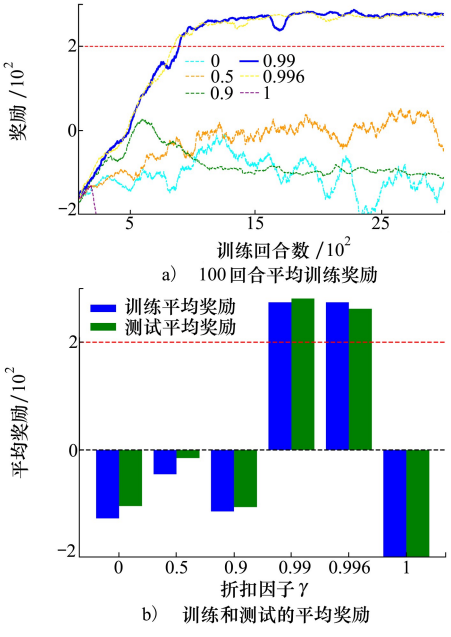


图 8　折扣因子 γ 与模型性能之间的关系

当 $\gamma=0$ 时,智能体得到局部最优解,无法达到全局最优解。当 $\gamma=1$ 时,智能体会过于乐观地估计未来的奖励,导致训练发散,模型无法稳定。当 $\gamma=0.99$ 时,模型的整体效果较优。

3.1.6 探索-利用平衡参数 ε

从图 9 中发现，较小的 ε_{start} 会使得模型更快地收敛，这是因为模型探索程度被降低，鼓励利用；但这会导致模型的训练以及测试效果不佳。分析奖励值，可以发现，当 $\varepsilon_{start}=1.0$ 时，模型的效果较佳。

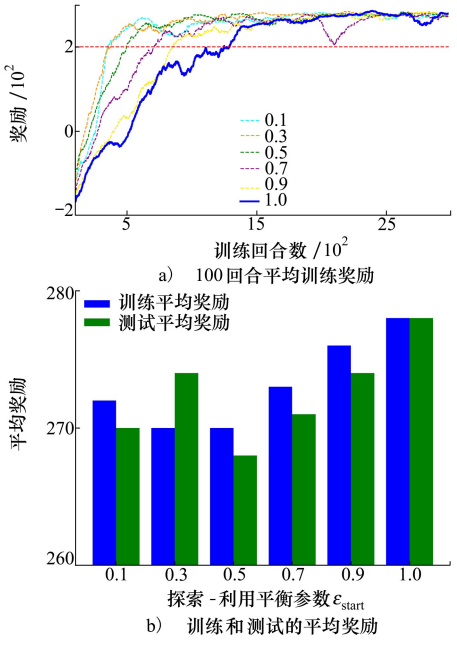


图 9 ε_{start} 与模型性能之间的关系

模型需要减少探索的程度，引入 ε_{decay} 参数控制探索程度。本文取 6 个不同值进行实验，实验结果如图 10 所示。发现当 ε 减少程度较大时，模型的性能不稳定；但是 ε_{decay} 一旦超出 0.998 时，模型性能大打折扣，这是因为模型在很大程度上鼓励探索，ε 减少程度过小不利于模型收敛。当且仅当 $\varepsilon_{decay}=0.998$ 时，模型的性能较优。

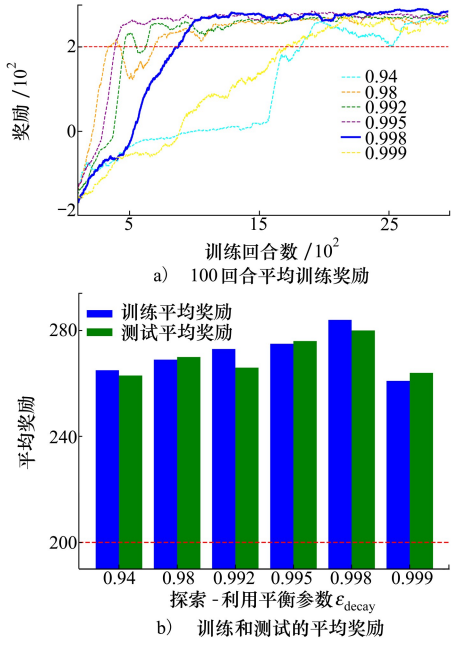


图 10 ε_{decay} 与模型性能之间的关系

当 ε 值达到 ε_{final} 时，模型会趋向于纯粹利用。图 11 展示了 6 个不同的 ε_{final} 值，从图中可以看出，ε_{final} 值越小，模型的性能越好，因为此时模型已学到足够的知识，针对本任务，更易获得高奖励，因此取 $\varepsilon_{final}=0$。

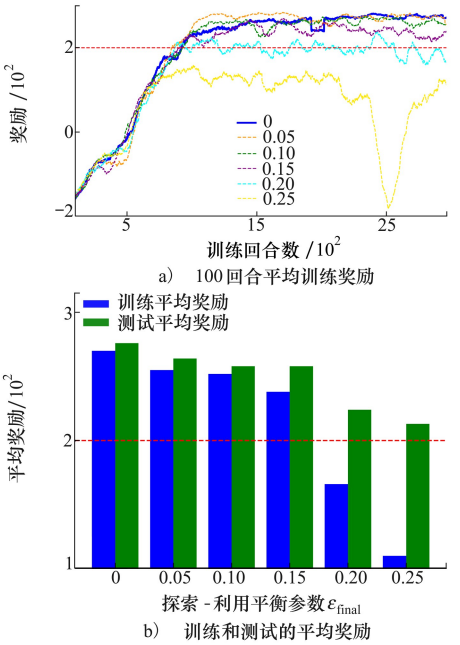


图 11 ε_{final} 与模型性能之间的关系

3.1.7 目标网络更新频率

目标网络的引入增加了学习的稳定性。从稳定性出发，本文设置不同的目标网络更新数值 $S_{update}=1$（即时更新，相当于未引入目标网络），500，700，900。上述数值表示经过这些步，目标网络开始更新。分析图 12 可知，随着目标网络更新频率的减慢（即目标网络更新数值增大），模型的稳定性得到提升。但是当超过一定数值时，模型的性能会大打折扣，因为它无法及时反映最新的训练进展，会导致目标 Q 值与当前策略的差异较大，学习不稳定或延缓学习的收敛。与此同时，模型的训练性能也影响着测试结果：训练得到的模型越稳定，测试效果越好。当目标网络更新数值 $S_{update}=700$ 时，模型性能较为稳定，测试效果较优。

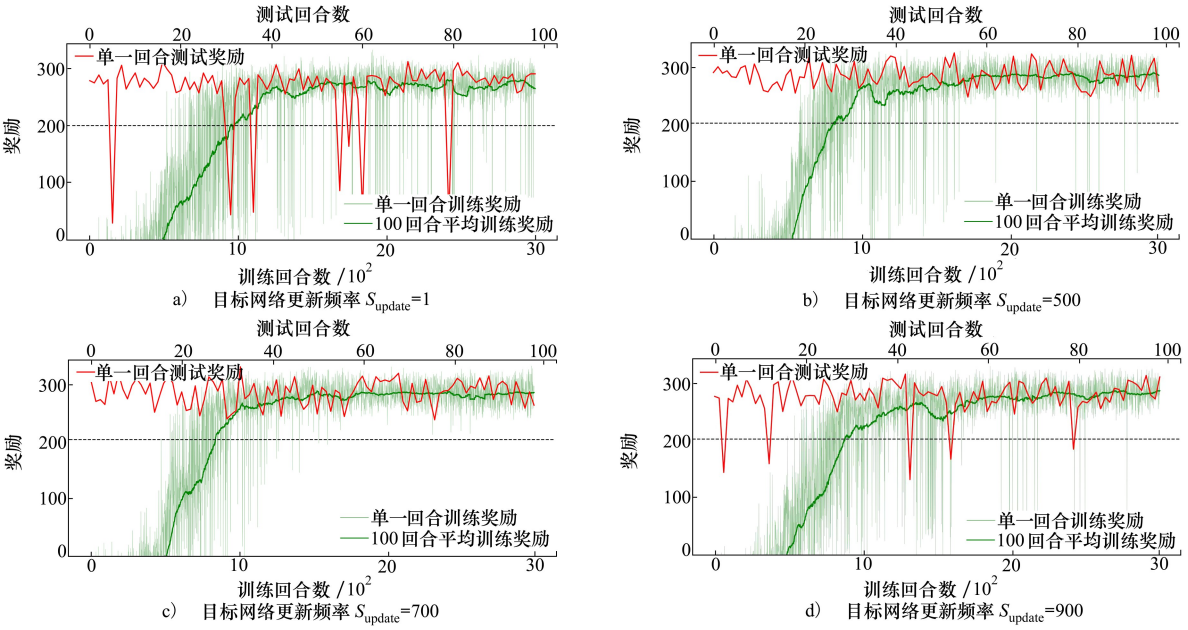


图 12　目标网络更新频率与模型性能之间的关系

3.1.8　具备优化超参数的模型

利用上述实验得到的优化超参数,重新进行模型训练,实验结果如图 13 所示,图 12a)是本文设置的初始超参数值得到的结果。

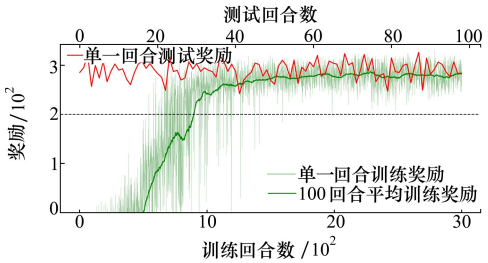


图 13　超参数优化的 DQN 模型

从稳定性分析,通过细致调参之后的 DQN 模型更稳定且更容易收敛。分析奖励值,针对基于 DQN 模型的月球着陆器任务,已知在测试阶段获得最高平均奖励是 280+,经过细致调参后,本文的得分是 290+。通过实验验证对比,细致调参对于模型性能十分有益。

3.2　模型鲁棒性测试

(18)式是“禁飞区域”规避的奖励函数

$$r_{AO}=\begin{cases}-10 & d_{AO}\in[0,1]\\ 0 & \text{其他}\end{cases}\tag{18}$$

当且仅当该区域与着陆器之间的距离属于[0,1],智能体受到惩罚。

图 14 是基于优化超参数和初始超参数,再加上“禁飞区域”规避奖励函数,重新训练模型得到的测试结果。分析获得的奖励值,具有优化超参数的 DQN 模型更具鲁棒性,能在性能所受影响较小的情况下完成着陆任务。

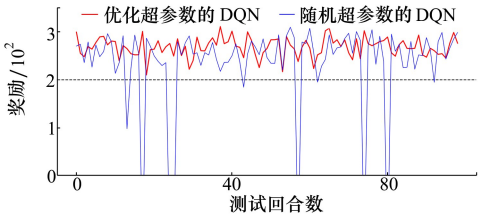


图 14　“禁飞区域”下不同 DQN 对比

3.3　模仿学习下的 DQN

本文利用启发式函数来模拟控制器,获得少量的演示数据集。在进行深度强化学习训练之前,首先利用获取到的演示数据集,结合损失函数(17),对模型进行预处理。需要注意的是 ε_{start} 有所调整,因为加入演示数据集进行模仿学习是要从中学到正确的决策,这种情况下不需要过多探索,这也是本文引入模仿学习的初衷;但是不能完全放弃探索,因为所使用的演示数据集覆盖程度有限,还需要一定程度探索。本文通过多次取值,分析获得的数据,最终调整 $\varepsilon_{start}=0.15$,实验结果如图 15 所示,图中未展示模仿学习阶段是为了更直观地反映应用模仿学习技术下的 DQN 模型效果。

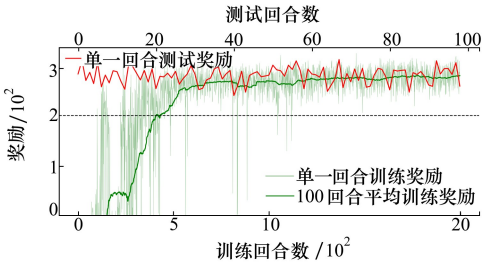


图 15 模仿学习与 DQN 相结合

从快速性分析,具备优化超参数的模型大约在训练 1 000 回合之后抵达最大奖励,并在其范围稳定。而添加演示数据集的模型以成倍的速度达到任务指标,即在训练 500 回合之后稳定收敛。从稳定性分析,两者都使用了目标网络,整体波动甚微。从奖励值分析,具备优化超参数模型在 100 个测试回合下的平均奖励为 290+,添加模仿学习后的 DQN 模型奖励值可达 280+,两者都以较高的奖励值完成任务。这表明添加演示数据集进行模仿学习训练对于 DQN 深度强化学习模型的性能提高十分有效。

4 结 论

本文应用月球着陆器环境,对 DQN 算法进行全面的性能分析与改进。首先,通过识别并定量分析影响性能的各种超参数,经过精心调整,获得了在各个方面表现较优的模型,与未经调整的模型相比,性能在各个方面都得到显著提升。其次,在原始问题的基础上引入额外的不确定性验证模型鲁棒性,得到的具有优化超参数的模型能够很好地处理随机性。最后,借助模仿学习,采用双阶段训练框架,提升模型训练效率,与上述具备优化超参数的模型训练过程相比,整体训练过程缩减了一半,解决了深度强化学习方法普遍存在的学习效率低下问题。

未来的工作可着眼于将 DQN 模型拓展至更广泛的应用领域,并将其作为融合多目标的基准。这一拓展包括重新审视和调整奖励函数,通过深入研究奖励结构、结合领域专家的知识,针对某一任务建立更为准确和有效的奖惩机制,以引导模型更好地学习任务。同样重要的是,专注于设计适用于不同任务的辅助学习数据。通过引入额外的学习信号和环境信息,为模型提供更全面、多样的训练数据,从而增强其泛化能力和对复杂任务的适应性。这一努力可能包括整合领域专家的经验知识、收集更多实际场景数据,以及运用其他先进的学习技术生成或合成数据。通过上述探索,使 DQN 模型更具适应性,以应对更为复杂和多样的实际应用挑战。

参考文献:

[1] FRANÇOIS-LAVET V, HENDERSON P, ISLAM R, et al. An introduction to deep reinforcement learning[J]. Foundations and Trends in Machine Learning, 2018, 11(3/4): 219-354

[2] CHOI R Y, COYNER A S, KALPATHY-CRAMER J, et al. Introduction to machine learning, neural networks, and deep learning[J]. Translational Vision Science & Technology, 2020, 9(2): 14

[3] LYU L, SHEN Y, ZHANG S. The advance of reinforcement learning and deep reinforcement learning[C]//2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms, 2022: 644-648

[4] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533

[5] 付一豪, 鲍泓, 梁天骄, 等. 基于视觉 DQN 的无人车换道决策算法研究[J]. 传感器与微系统, 2023, 42(10): 52-55
FU Yihao, BAO Hong, LIANG Tianjiao, et al. Research on decision algorithm for autonomous vehicle lane change based on vision DQN[J]. Transducer and Microsystem Technologies, 2023, 42(10): 52-55(in Chinese)

[6] 林歆悠, 叶卓明, 周斌豪. 基于 DQN 强化学习的自动驾驶转向控制策略[J]. 机械工程学报, 2023, 59(16): 315-324
LIN Xinyou, YE Zhuoming, ZHOU Binhao. DQN reinforcement learning-based steering control strategy for autonomous driving [J]. Jouranal of Mechanical Engineering, 2023, 59(16): 315-324 (in Chinese)

[7] STREHL A L, LI L, WIEWIORA E, et al. PAC model-free reinforcement learning[C]//Proceedings of the 23rd International Conference on Machine Learning, 2006: 881-888

[8] ZHU J, WU F, ZHAO J. An overview of the action space for deep reinforcement learning[C]//Proceedings of the 2021 4th International Conference on Algorithms, Computing and Artificial Intelligence, 2021: 1-10

[9] JÄGER J, HELFENSTEIN F, SCHARF F. Bring color to deep Q-networks: limitations and improvements of DQN leading to rainbow DQN// Reinforcement learning algorithms: analysis and applications[M]. Cham: Springer International Publishing,

2021: 135-149
[10] HUSSEIN A, GABER M M, ELYAN E, et al. Imitation learning: a survey of learning methods[J]. ACM Computing Surveys, 2017, 50(2): 1-35
[11] 况立群, 李思远, 冯利, 等. 深度强化学习算法在智能军事决策中的应用[J]. 计算机工程与应用, 2021, 57(20): 271-278
KUANG Liqun, LI Siyuan, FENG Li, et al. Application of deep reinforcement learning algorithm on intelligent military decision system[J]. Computer Engineering and Applications, 2021, 57(20): 271-278 (in Chinese)
[12] ZHAI J, LIU Q, ZHANG Z, et al. Deep Q-learning with prioritized sampling[C]//23rd International Conference on Neural Information Processing, Tokyo, Japan, 2016: 13-22
[13] ZHANG B, RAJAN R, PINEDA L, et al. On the importance of hyperparameter optimization for model-based reinforcement learning[C]//International Conference on Artificial Intelligence and Statistics, PMLR, 2021: 4015-4023
[14] BERGSTRA J, BENGIO Y. Random search for hyper-parameter optimization[J]. Journal of Machine Learning Research, 2012, 13(2): 281-305
[15] ZENG D, YAN T, ZENG Z, et al. A hyperparameter adaptive genetic algorithm based on DQN[J]. Journal of Circuits, Systems and Computers, 2023, 32(4): 1-24
[16] WU J, CHEN X Y, ZHANG H, et al. Hyperparameter optimization for machine learning models based on Bayesian optimization [J]. Journal of Electronic Science and Technology, 2019, 17(1): 26-40

Performance evaluation and improvement of deep Q network for lunar landing task

YUE Qi[1], SHI Yifan[1], CHU Jing[1], HUANG Yong[2]

(1.School of Automation, Xi'an University of Posts & Telecommunications, Xi'an 710121, China;
2.School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: Reinforcement learning is now being applied more and more in a variety of scenarios, the majority of which are based on the deep Q network (DQN) technology. However, the algorithm is heavily influenced by multiple factors. In this paper, we take the lunar lander as a case to study how various hyper-parameters affect the performance of the DQN algorithm, based on which we tune to get a model with better performance. At present, it is known that the DQN model has an average reward of 280+ on 100 test episodes, and the reward value of the model in this article can reach 290+. Meanwhile, its robustness is tested and verified by introducing additional uncertainty tests into the original problem. In addition, to speed up the training process, imitation learning is incorporated in our model, using heuristic function model guidance method to obtain demonstration data, which accelerates training speed and improves performance. Simulation results have proven the effectiveness of this method.

Keywords: deep reinforcement learning; DQN; imitation learning

引用格式:岳顾, 石伊凡, 褚晶, 等. 深度 Q 网络在月球着陆任务中的性能评估与改进[J]. 西北工业大学学报,2024,42(3): 396-405
YUE Qi, SHI Yifan, CHU Jing, et al. Performance evaluation and improvement of deep Q network for lunar landing task [J]. Journal of Northwestern Polytechnical University, 2024, 42(3): 396-405 (in Chinese)