Journal of System Simulation

Volume 35 | Issue 12 Article 15

12-30-2023

Research and Development of Simulation Training Platform for Multi-agent Collaborative Decision-making

Cheng Cheng
School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China, sy2303801@buaa.edu.cn

Zhijie Chen
School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

Ziming Guo
AVIC Shenyang Aircraft Design and Research Institute, Shenyang 110035, China

Ni Li
School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China, lini@buaa.edu.cn

Research and Development of Simulation Training Platform for Multi-agent Collaborative Decision-making

Abstract

Abstract: Reinforcement learning simulation platform can be an interactive and training environment for reinforcement learning. In order to make the simulation platform compatible with the multi-agent reinforcement learning algorithms and meet the needs of simulation in military field, the similar processes in multi-agent reinforcement learning algorithms are refined and a unified interface is designed to embed and verify different types of deep reinforcement learning algorithms on the simulation platform and to optimize the back-end service of the simulation platform to accelerate the training process of the algorithm model. The experimental results show that, by unifing the interface, the simulation platform can be compatible with many different types of multi-agent reinforcement learning algorithms, and the algorithm training efficiency can be significantly improved after the back-end service framework reconstruction and parameter quantization.

Keywords

artificial intelligence, multi-agent, reinforcement learning, virtual simulation, training acceleration

第 35 卷第 12 期 系统仿真学报© Vol. 35 No. 12
2023 年 12 月 Journal of System Simulation Dec. 2023

多智能体协同决策仿真平台研究与开发

成城[1]，陈智杰[1]，郭子铭[2]，李妮[1*]
(1. 北京航空航天大学 自动化科学与电气工程学院，北京 100191；2. 中国航空工业集团公司 沈阳飞机设计研究所，辽宁 沈阳 110035)

摘要：强化学习仿真平台为强化学习提供交互和训练的环境。为了使仿真平台兼容多智能体强化学习算法，满足军事领域仿真的需求，提炼多智能体强化学习算法中的相似流程，*设计统一接口，将多种不同类型深度强化学习算法在仿真平台进行嵌入验证；优化仿真平台后端服务框架以加速算法模型的训练过程。*实验结果表明：在仿真平台中统一接口规范，能够兼容多种不同类型的多智能体强化学习算法，显著提升了后端服务框架重构和参数量化后算法训练效率。

关键词：人工智能；多智能体；强化学习；虚拟仿真；训练加速
中图分类号：TP391.9 文献标志码：A 文章编号：1004-731X(2023)12-2669-11
DOI: 10.16182/j.issn1004731x.joss.23-FZ0821
引用格式: 成城, 陈智杰, 郭子铭, 等. 多智能体协同决策仿真平台研究与开发[J]. 系统仿真学报, 2023, 35(12): 2669-2679.
Reference format: Cheng Cheng, Chen Zhijie, Guo Ziming, et al. Research and Development of Simulation Training Platform for Multi-agent Collaborative Decision-making[J]. Journal of System Simulation, 2023, 35(12): 2669-2679.

Research and Development of Simulation Training Platform for Multi-agent Collaborative Decision-making

Cheng Cheng[1], *Chen Zhijie*[1], *Guo Ziming*[2], *Li Ni*[1*]
(1. School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China;
2. AVIC Shenyang Aircraft Design and Research Institute, Shenyang 110035, China)

Abstract: Reinforcement learning simulation platform can be an interactive and training environment for reinforcement learning. In order to make the simulation platform compatible with the multi-agent reinforcement learning algorithms and meet the needs of simulation in military field, *the similar processes in multi-agent reinforcement learning algorithms are refined and a unified interface is designed to embed and verify different types of deep reinforcement learning algorithms on the simulation platform* and to *optimize the back-end service of the simulation platform to accelerate the training process of the algorithm model.* The experimental results show that, by unifing the interface, the simulation platform can be compatible with many different types of multi-agent reinforcement learning algorithms, and the algorithm training efficiency can be significantly improved after the back-end service framework reconstruction and parameter quantization.

Keywords: artificial intelligence; multi-agent; reinforcement learning; virtual simulation; training acceleration

0 引言

计算机生成兵力领域涌现大量关于多智能体协同决策的研究，多智能体强化学习算法成为了焦点，在具备完善规则约束下的博弈场景中，如

收稿日期：2023-07-03 修回日期：2023-10-01
第一作者：成城(2001-)，男，硕士生，研究方向为系统仿真与智慧制造。E-mail：sy2303801@buaa.edu.cn
通讯作者：李妮(1980-)，女，教授，博士，研究方向为虚拟样机技术及分布式交互仿真技术。E-mail：lini@buaa.edu.cn

围棋、电子竞技等，多智能体强化学习能够有效地进行学习和推理。而训练多智能体强化学习算法需要特定的环境－多智能体协同决策仿真平台(简称仿真平台)，仿真平台为强化学习算法提供统一的接口及调用方式，平台中的行为决策模型通过强化学习算法实现，仿真平台同时为集成强化学习算法的模型提供交互环境。用户可根据自定义算法的实际情况对网络参数、环境参数进行实时调整，得到理想的训练效果。

2013 年，Google 旗下的 DeepMind 实验室在 NeurIPS 上发表并提出了深度强化学习算法(DRL)[1]的概念；2017 年，谷歌 DeepMind 研发了 AlphaGo-Zero，与前代相比，Zero 的网络更加精简，采用完全自主学习的方式进行训练，完全不用人类棋谱数据，就能达到更强的性能；2022 年，OpenAI 提出了基于人类反馈强化学习[2]的智能问答机器 ChatGPT，短短十年，DRL 已经成为人类解决未知棘手问题的重要方式。对于 DRL 算法而言，需要一个“训练场”将算法进行嵌入训练，训练环境通常包括多个环境的逻辑，且满足特定的接口，Gym[3]是著名的测试强化学习算法的平台，提供了一套标准的接口，在算法和环境之间进行交互，其中，大多数环境面向的是单智能体算法。目前，能够支撑基于多智能体强化学习算法进行战术行为决策训练的仿真平台较少，国内科技公司如阿里巴巴，与伦敦大学合作，推出强化学习算法仿真研究环境平台 Gym StarCraft，将双向协调网络引入，通过学习战斗场景中的最优策略来指挥智能体协作[4]。中国电子科技集团公司第二十八研究所的 NRIEE 智能博弈平台，支持联合作战仿真推演和博弈对抗 AI 训练。这些仿真平台主要面向的是单智能体强化学习算法的训练，对于多智能体强化学习，算法的复杂程度更高，更难适配统一的仿真平台接口。目前，开源仿真平台大多与特定 CGF(computer generated force)系统深度绑定在一起，不具备通用性，也导致构建的行为模型无法在多种仿真系统间重用。随着智能博弈的兴起，不同类型的仿真系统不断涌现，国内外高校、研究所都已逐渐开发完成各种类型的智能博弈平台，但是不同平台之间并不互通。智能行为建模的算法接口虽然在不同平台表现不同，但本质都一样，因此，设计开发可兼容多种仿真系统、具备一定通用性的多智能体仿真平台，对提升作战装备行为模型的可重用性、促进模型资源的积累具有重要的意义。

本文研究了一种多智能体仿真平台优化方案，设计的统一接口能够兼容多种多智能体强化学习算法，可将多种不同类型的多智能体强化学习算法进行嵌入验证；对后端服务框架中算法训练过程的效率进行优化，保证兼容性的同时也对运行效率进行了改善。

1 仿真平台总体框架及功能实现

1.1 仿真平台总体框架

仿真平台总体框架如图 1 所示，用户界面包括基本的参数配置、过程控制、状态监控，提供可视化的操作界面；智能服务对强化学习算法中的推理决策、奖励设计、经验存储、训练优化等方法进行封装，模型可以兼容单智能体强化学习、多智能体强化学习算法；智能行为代理模型与对应的仿真模型通过路由进行数据交互，实现对智能模型行为决策的代理。

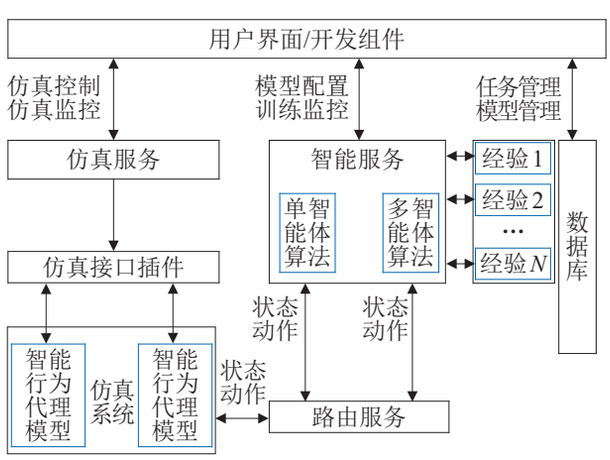

图 1 仿真平台总体框架

Fig. 1 Overall framework of simulation platform

DOI: 10.16182/j.issn1004731x.joss.23-FZ0821

多智能体协同决策仿真平台的后端为强化学习算法的训练提供相应支持，在仿真系统中，以仿真引擎为媒介，实现模型调度、数据交互及其他管理功能。仿真环境提供了基础的物理模型，能够在仿真环境中运行，而智能体是行为模型，物理模型和行为模型间的交互接口分为状态和动作两部分，如图2中平台提供的典型1V1空战场景红蓝双方对抗飞机中的物理模型，实现了运动、雷达、武器、毁伤、火控等基础模块，其中，蓝方飞机的状态包括自身三维位置、三轴姿态、标量速度、航向角、携弹量等信息以及类似的红方飞机信息；动作包括战术机动、雷达操纵、武器操纵等。仿真平台对多种仿真粒度提供支持，如双机对抗对应的是交战级粒度仿真，多架无人机路径探索对应的是任务级粒度仿真。平台自身对仿真粒度不作限制，具体取决于用户所选择的智能体，即行为模型的要求，用户可以根据任务需求对行为模型的动作及状态进行自定义配置。

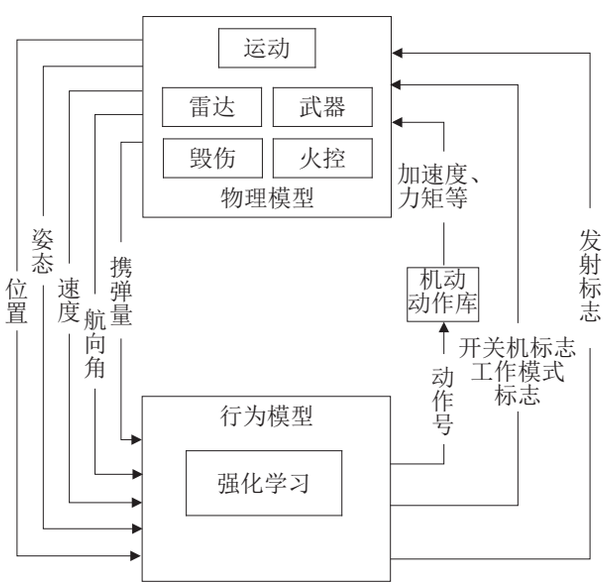

图2　物理模型及行为模型交互关系
Fig. 2　Interaction between physical model and behavioral model

物理模型和行为模型之间存在智能行为代理模型，这是一类与特定仿真系统相关的仿真模型，通常是行为模型，其本身并不执行行为决策的具体逻辑，而是远程调用协议(remote procedure call, RPC)[5]对实际进行行为决策的模型服务进行代理。智能代理通过仿真模型间的数据交互获取模型服务所需的状态数据，经由RPC协议发送至路由服务并获得动作数据，再通过数据交互将其分发给对应的仿真模型，从而实现对智能行为决策的代理。

仿真平台的功能在于为模型提供可以交互运行的环境，同时能够支撑智能体的迭代训练。使用平台服务时，首先，在平台中构建想定和模型，定义所需要的智能体，即行为模型与仿真平台的接口，通过配置文件生成智能行为代理模型，在仿真平台中自动配置物理模型和智能体之间的交互关系，统一管理模型数据的收发。随后，在智能推理模块中即可进行强化学习算法的选择及算法超参数、奖励函数等的配置。之后，进行迭代训练，训练完成后的神经网络权重参数等可以上传至云端或本地进行存储，便于日后部署及优化运行。

仿真平台的智能服务对强化学习算法模型推理决策、奖励计算、经验存储、优化训练等操作的时序结构关系进行封装，使强化学习模型在被调用一次的情况下便能完成完整的训练流程。模型服务中包含一个实现了强化学习模型统一接口的强化学习模型，该模型可以兼容单智能体、多智能体等多种强化学习算法，还包括一个人为设计或从专家经验学得的奖励函数，用于指导强化学习模型的策略优化。

1.2 仿真平台工作流程

仿真平台的工作流程如图3所示。用户在平台UI(user interface)选择模型训练后，仿真控制模块通过RPC调用控制仿真引擎开始仿真，指定仿真引擎使用的想定文件，并对仿真过程中的必要参数进行初始化；仿真控制模块通过RPC调用发送单步推进指令；仿真引擎模块通过智能模型使用平台统一模型接口规范，获取当前被训练模型

关注的态势数据，并将其重组为神经网络所需的数据结构，然后通过RPC调用将其发送到推理服务；推理服务模块根据神经网络定义的态势评估方法，对传入态势数据进行评估，完成决策计算，将行为决策通过RPC返回；仿真引擎模块执行返回的行为决策，并将智能模型的输出发送到仿真引擎，完成仿真系统完整的一帧推进；在获取态势数据后，仿真引擎模块通过RPC调用发送仿真数据，存储服务将相应的数据存储到经验回放池中，对神经网络参数进行优化训练；仿真控制模块判断当前仿真态势是否越过预先设定的状态边界，如果越过边界，则结束当前仿真回合(想定的一次运行)，开启新的仿真回合，否则进行下一个仿真帧的循环。

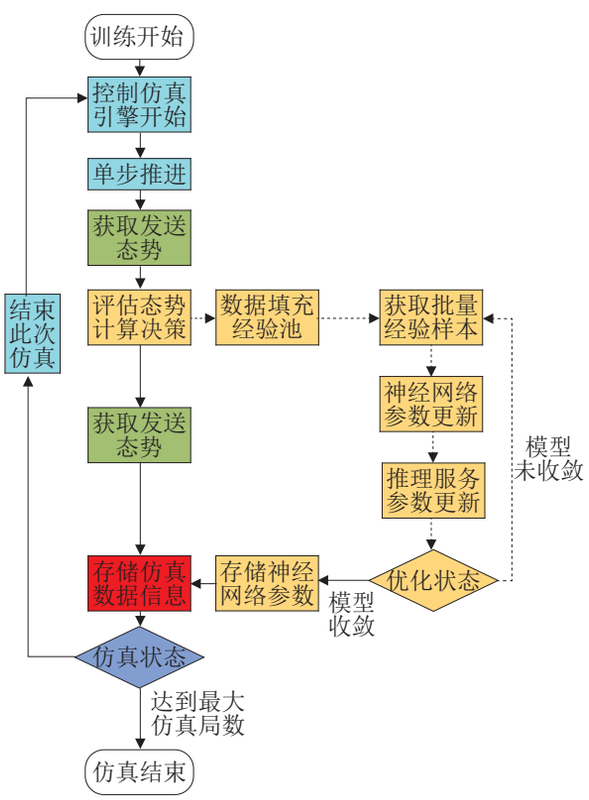

图3 仿真平台工作流程

Fig. 3 Simulation platform function implementation flowchart

2 仿真平台训练框架后端服务开发

强化学习模型与仿真环境的交互过程满足一定的流程逻辑，按照面向对象的思想，可以将强化学习模型看作一个对象，具备一些共有且固有的操作，称之为强化学习模型的统一接口。后端服务框架中的智能代理模型执行模型训练时的行为，但不执行具体逻辑。现有仿真平台的后端服务框架存在着运行效率低的问题，采用的实验方法包括使用异步改善原有代码中的同步阻塞及网络交互、对神经网络进行参数量化[6]。

2.1 异步协程优化网络交互

传统的python后端框架采用同步阻塞形式，即按序执行相应事务请求，只有当上一个事务请求完成时，服务端才会发送下一个请求到相应模块触发事务，当一个事务执行过程中发生异常时，整个程序将会发生阻塞。这种网络交互形式虽然简单稳定，但存在着大量的CPU时间资源调度的浪费。为此，python提供了异步编程的方法—协程。协程是一种轻量级的用户态模型，其本质是单线程。由于全局解释器锁(global interpreter lock, GIL)的存在，导致python在CPU上的多核并行效率较低，协程是在一个线程实现的并发方式，避免了线程的切换，同时，协程的编程风格与同步编程方式极其类似，主要通过python的第三方库asyncio提供的API实现。

2.1.1 后端同步阻塞模型

多智能体协同决策仿真平台的网络交互主要发生在服务端和智能体模型之间，常规后端框架设计采用的是同步阻塞形式，如图4所示。

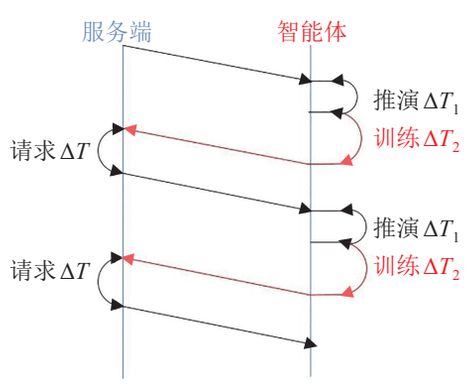

图4 后端同步阻塞模型网络交互

Fig. 4 Synchronous back-end framework network interaction

DOI: 10.16182/j.issn1004731x.joss.23-FZ0821

后端的网络交互模型可以简化如下：智能体模型中的阻塞部分主要发生在推演过程和训练过程，服务端在收到每一次训练周期结束的信号后，间隔一定时间ΔT发出下一次请求。假设训练的总帧数为F，每一帧的推演时间为ΔT_1，训练时间为ΔT_2，则训练全过程耗时为

$$T=F\times(\Delta T+\Delta T_1+\Delta T_2)-\Delta T \tag{1}$$

2.1.2 优化后异步非阻塞模型

采用异步协程对原有框架进行优化，将涉及阻塞部分的函数定义为async形式，阻塞部分相应代码调用await方法，这样每当程序在该处发生阻塞时，将会自动切换以执行其他的任务，实现异步。在原后端同步阻塞模型中，服务端需要等待智能体的每一帧训练过程结束后发送请求，在异步非阻塞模型中，每次智能体进行推演过程之前，将先判断上一帧训练过程是否执行完毕，若已经完成训练，则直接向服务器返回请求；若未完成训练，则调用阻塞代码进行训练，由此减少了对训练过程的重复等待。

当服务端发送请求的时间间隔ΔT较大时，异步非阻塞后端框架的网络交互如图5所示。

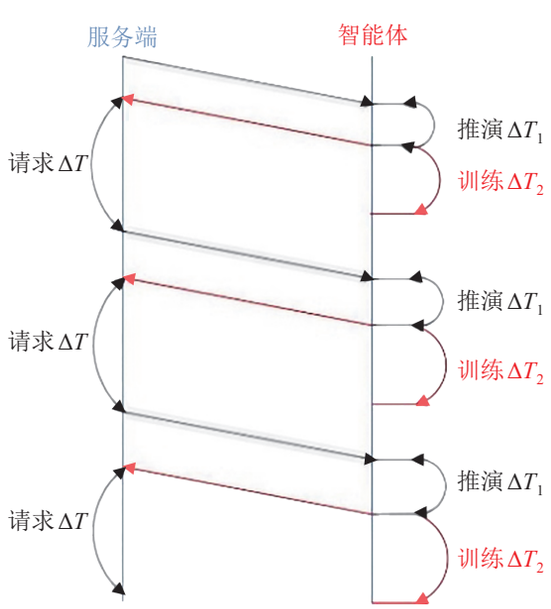

图5　异步后端框架网络交互($\Delta T>\Delta T_2$)

Fig. 5　Aynchronous back-end framework network interaction ($\Delta T>\Delta T_2$)

除第一次执行过程外，之后每一帧智能体在进行推演时上一帧训练过程已经结束，因此，在此处不发生阻塞，直接向服务器返回请求。在等待服务器的下一次请求时，同时执行本次的训练过程。训练的全过程耗时为

$$T=F\times(\Delta T+\Delta T_1)-\Delta T \tag{2}$$

当服务端发送请求的时间间隔ΔT较小时，后端框架网络交互如图6所示。

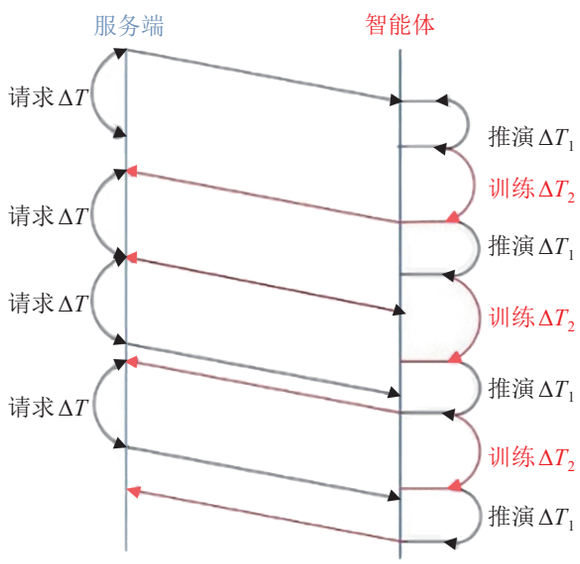

图6　异步后端框架网络交互($\Delta T<\Delta T_2$)

Fig. 6　Asynchronous back-end framework network interaction ($\Delta T<\Delta T_2$)

与ΔT较大时相同，除了第一次执行过程外，之后的每一帧智能体在推演时上一帧训练过程已经完成，因此，无需等待直接返回。由于ΔT较小，下一个请求已经到来，所以，直接开始下一次的执行过程。训练全过程耗时为

$$T=F\times(\Delta T_1+\Delta T_2) \tag{3}$$

综上，若不考虑仿真训练过程中的第一帧及最后一帧耗时的特殊情况，能够近似得到优化后的异步非阻塞框架网络交互运行耗时的公式：

$$T\approx F\times(\Delta T_1+\max(\Delta T_2,\Delta T)) \tag{4}$$

与原同步阻塞后端框架相比，相对效率得到了提升：

$$\tau=\frac{F\times(\Delta T+\Delta T_2)-F\times\max(\Delta T_2,\Delta T)}{F\times(\Delta T+\Delta T_1+\Delta T_2)} \tag{5}$$

将原有同步阻塞框架与优化的框架进行对比消融实验，分别进行3次帧数为5 000的训练，训练耗时结果如表1所示，取3次实验结果平均，得到相对效率提升为29.08%。

表1 异步后端框架训练加速实验结果
Table 1 Asynchronous back-end framework trainingacceleration experiment results s

序号	原同步框架训练耗时	优化后框架训练耗时
1	83.98	59.41
2	83.56	59.21
3	83.71	59.59
平均值	83.75	59.40

2.2 神经网络参数量化

在深度强化学习训练中，使用了神经网络，神经网络中常见的参数精度有float16(半精度)、float32(单精度)、float64(双精度)3种。对于复杂问题进行求解时，有时采取合适的精度进行训练，可以减小GPU占用的内存，加快训练速度。

在不考虑数据类型转换的效率下，半精度运算速度应该为单精度的2倍左右。但由于半精度数据类型比较特殊，只用2个字节存储，所以容易造成数据溢出，在神经网络计算梯度的过程中尤为明显，因为梯度通常比权重小得多，往往容易出现下溢的情况。单精度与双精度在搭建神经网络时更为常用，在pandas库中对2种数据类型占用内存的情况进行了分析。对于一个长度为8的numpy数组，当数据类型为单精度时，它所占用的内存为160 bytes；当数据类型为双精度时，它所占用的内存为192 bytes，两者相差约17%。

在实验过程中，采用深度确定性策略梯度算法，对训练过程进行5 000次模拟，分别测量半精度、单精度、双精度数据类型的运行效率，所消耗时间测量结果如表2所示。

由实验结果分析可知，当数据类型从单精度变为半精度时，效率不仅没有提高，还有轻微的下降。原因在于神经网络的计算过程中有一部分过程默认是单精度计算，强行将数据转换为半精度需要消耗一定的时间；从双精度转换成单精度时，经计算平均效率提升为17.42%，与上文提到单精度和双精度数据类型的占用内存差距相当。

表2 参数量化实验结果
Table 2 Parameter quantization experimental results s

序号	半精度数据类型训练耗时	单精度数据类型训练耗时	双精度数据类型训练耗时
1	37.34	36.18	43.00
2	37.45	35.56	43.76
3	37.21	35.79	43.44

从表2数据可以分析出，在神经网络训练过程中，若对数据精度没有特殊要求，尽量使用单精度数据类型进行神经网络训练。经过更进一步的调研，发现可以通过混合精度训练[7]的方式来提升效率，即采用单精度与半精度混合的方式，其原理是将权重和梯度以半精度的方式进行存储，在训练过程中以单精度进行训练。在实验过程中对该方法也进行了尝试，得到的效果并不理想。原因在于混合精度训练在小批量模型的计算中并不会带来速度的提升，因为小批量模型的计算效率主要由IO影响，即CPU与GPU之间数据传输的效率，模型本身的计算已经很快，引入混合精度训练后，触发了半精度与单精度之间的数据类型转换，消耗了更多的时间。

3 多智能体强化学习算法嵌入验证

3.1 仿真平台算法统一接口设计

为了实现仿真平台对不同类型强化学习算法的兼容性，提出了强化学习算法嵌入的统一接口设计。主要包括初始化、推演、训练、存储4个模块。

仿真平台为强化学习算法嵌入设计了一套统一接口，基本方法如图7所示。在基类中定义了强化学习算法嵌入的规范，主要包括：初始化接口，用于初始化算法的输入输出维度、学习率、网络层数、智能体数量，以及初始化经验回放池等算法的固有属性；推演决策接口，用于定义智

DOI: 10.16182/j.issn1004731x.joss.23-FZ0821

能体执行动作的方法，返回值为与环境交互得到的动作；经验存储接口，将智能体执行动作后得到的四元组(状态、动作、下一状态、奖励)和训练结束标志terminated等存入经验回放池中，以备后续训练时采样使用；训练接口，用于执行强化学习算法各网络的训练部分；数据获取接口用于获取网络的权重参数和训练过程等信息。

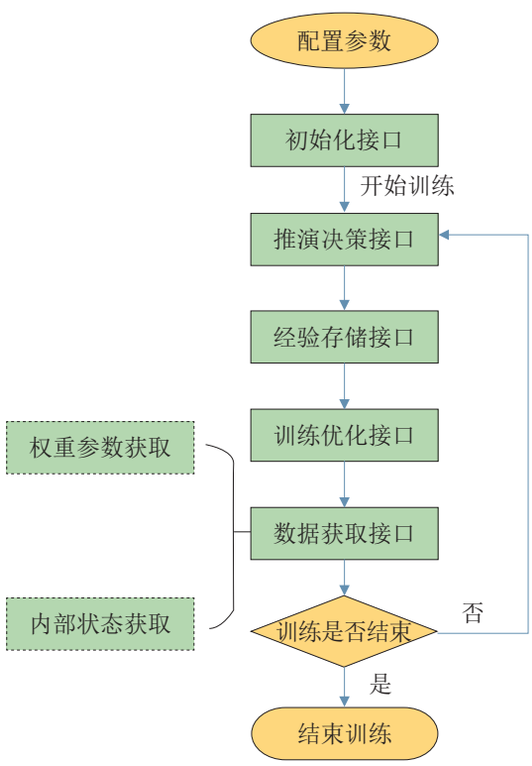

图7　仿真平台统一接口设计
Fig. 7　Unified interface design of simulation platform

3.2 典型多智能体算法嵌入过程

将强化学习算法的统一接口设计并封装后，采用典型多智能体算法MADDPG(multi-agent deep deterministic policy gradients)[8]按照统一接口进行嵌入，MADDPG算法是典型的中心化决策、去中心化执行的多智能体强化学习算法。首先是MADDPG算法的初始化部分，对于每个智能体，在全体智能体为其添加相应的Actor网络和Critic网络及各自的目标网络，并将目标网络的trainable参数调为False，避免目标网络的权重在计算梯度时与价值网络一同更新，同时初始化各智能体的经验回放池。

Actor和Critic网络[9]的定义如图8所示。每个网络除输入层和输出层外，各有自定义的隐藏层，此处隐藏层规模为2×256。在Actor网络的输出层后还加入了Lambda层用来对输出动作进行范围限制，保证输出动作的合法性；对于Critic，由于需要观测全局信息，所以输入层由所有智能体的状态和动作拼接而成。

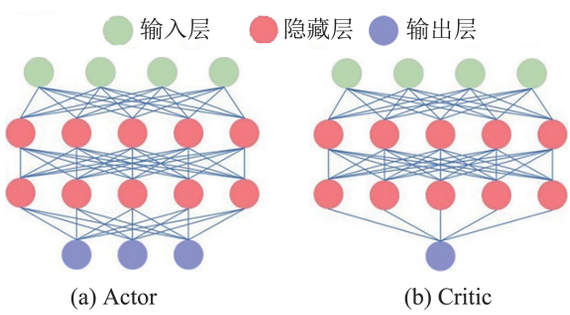

图8　MADDPG网络结构设计
Fig. 8　Neural networks design of MADDPG

MADDPG的推演决策接口，首先通过Actor网络依据当前策略对每个智能体执行动作与环境进行交互，随后给动作添加一定的噪声。本文采用的噪声是高斯噪声，主要通过定义一个噪声类来实现，同时对添加噪声的动作按环境所要求的动作范围进行裁剪，保证所得到的动作值均为环境所允许的合法动作。

MAADPG的训练优化接口部分，首先判断经验池是否已经有足够多的经验，当经验池中已经积累了满足训练要求的经验数目时，执行下一步训练过程，否则继续执行推理进行经验存储。训练前每次先从经验池中取出一定批量的经验，对于每个智能体采样的必须是同一时刻的经验，否则会造成训练不稳定，Actor网络会难以学习到合适的策略。

训练优化接口中，对所有智能体的状态动作进行拼接，对每个智能体的Critic网络计算时序差分误差，从而更新Critic网络；根据Critic更新的策略，对当前智能体的动作进行重新预测；和其他智能体的动作进行拼接，输入Critic网络，以得

到新的Q值来指导Actor网络的更新；对目标网络进行软更新。

在MADDPG中还继承了基类统一接口的经验存储、状态获取，方法与基类中基本一致。多智能体微粒环境[10](multi-agent particle environment, MPE)是常见的测试多智能体强化学习算法的交互环境，环境中提供多个微粒智能体，对相应目的地进行占领，奖励函数与智能体和目的地间距负相关，在MPE环境中进行算法嵌入验证，分别得到Actor网络的Loss曲线和奖励函数曲线，如图9~10所示。算法经过多次迭代后已经收敛，并且能学习到较优的策略。

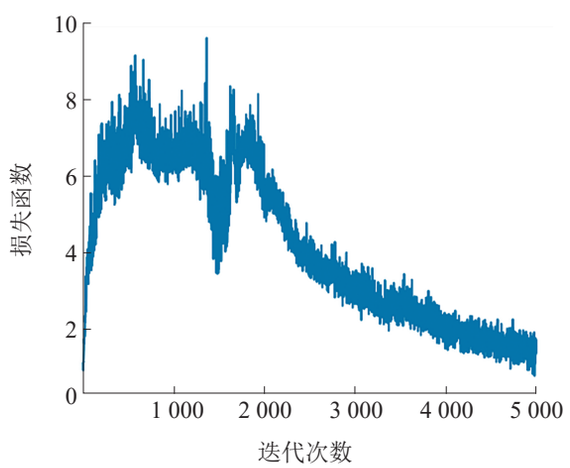

图9 MADDPG损失函数曲线
Fig. 9 Loss function curve of MADDPG

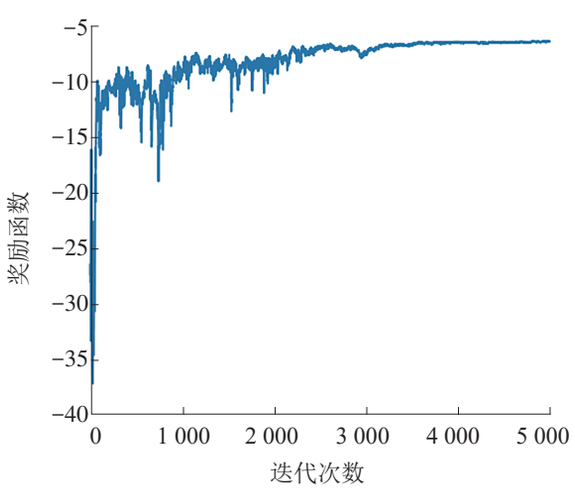

图10 MADDPG奖励曲线
Fig. 10 Reward function curve of MADDPG

4 MADDPG算法仿真平台案例验证

4.1 仿真平台验证案例想定

多智能体强化学习算法验证实验方案如表3所示。实验场景为红方多飞行器对未知区域的蓝方舰船的目标探测，红方实验组采用嵌入的多智能体强化学习算法模型，蓝方基准组给定拦截概率对红方飞行器进行拦截。实验的全局状态空间定义为红方实验组的飞行器和蓝方基准组的舰船所处位置，动作空间依据红方飞行器的动力学模型创建，并以成功探测目标作为终局奖励；实验方案为选取MADDPG算法，让红方实验组的4个飞行器探测搜索蓝方的1~2个舰船。

表3 仿真平台想定案例设计
Table 3 Simulation platform scenario design

智能体类型	算法模型	奖励设计	无人机/舰船数量
无人机	MADDPG	势函数方法设计	2~4
舰船	—	—	1~2

注：“—”代表舰船模型不采用具体算法进行驱动，根据自定义规则进行移动或随机移动。

想定案例中的奖励函数是基于目标距离的势函数设计[11]的，具体表达式为

$$\varphi = -\sum_{i=1}^{n} w_i \times (d(u_i, s))^2 \tag{6}$$

式中：w_i为每架无人机的可调权重参数；$d(u_i, s)$为第i架无人机与目标舰船之间的经纬距离，由两者所处经度和纬度之差的平方和求得。

4.2 MADDPG算法案例验证

在MADDPG等强化学习算法中使用到了Tensorflow的GPU版本的相关用法，所以在计算机硬件方面需要有NVIDIA独立显卡的支持。为了防止造成本地计算资源环境的混乱，仿真平台使用了Docker容器对相关依赖进行封装，只需要在初次运行仿真平台时调用相关命令打包生成镜像，随后在Docker中运行形成容器，后续使用时即可实现一次调用，无需重复构建环境。默认情

DOI: 10.16182/j.issn1004731x.joss.23-FZ0821

况下平台使用单卡训练，当存在多张GPU能支持强化学习并行训练时，可使用Tensorflow提供的分布式策略将计算图与变量进行拷贝，并将总数据批量大小均分在各GPU上，从而实现大模型的训练。

按照仿真平台统一接口的设计，在平台中进行了多种多智能强化学习算法的嵌入。以MADDPG算法为例，在实验案例中给定4架无人飞行器对位置未知的潜艇进行探索，当任意1架无人机探测到潜艇所在位置时即视为探索成功。仿真平台的前端用户界面为用户提供了可视化的操作接口，并能实时监视算法训练过程中的状态。在平台用户界面对奖励函数及算法参数等进行配置，仿真平台算法参数配置界面及训练过程分别如图11~12所示。

训练过程中无人机会以各自的策略对潜艇的位置进行协同探索，当存在某一架无人机成功侦测到敌方舰船所在位置时，则该局结束。训练结果中的Actor Loss和Critic Loss曲线如图13~14所示，经过3万次左右的训练，模型已经得到收敛，证明嵌入的MADDPG算法能在仿真平台案例中正常训练。

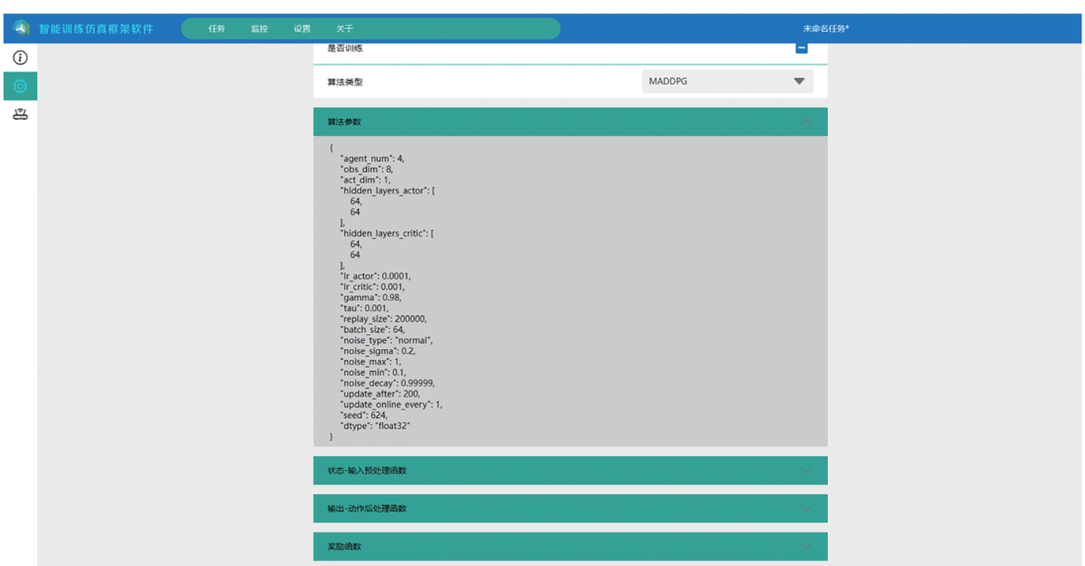

图11　仿真平台用户算法参数配置界面

Fig. 11　Simulation platform user algorithm parameter configuration interface

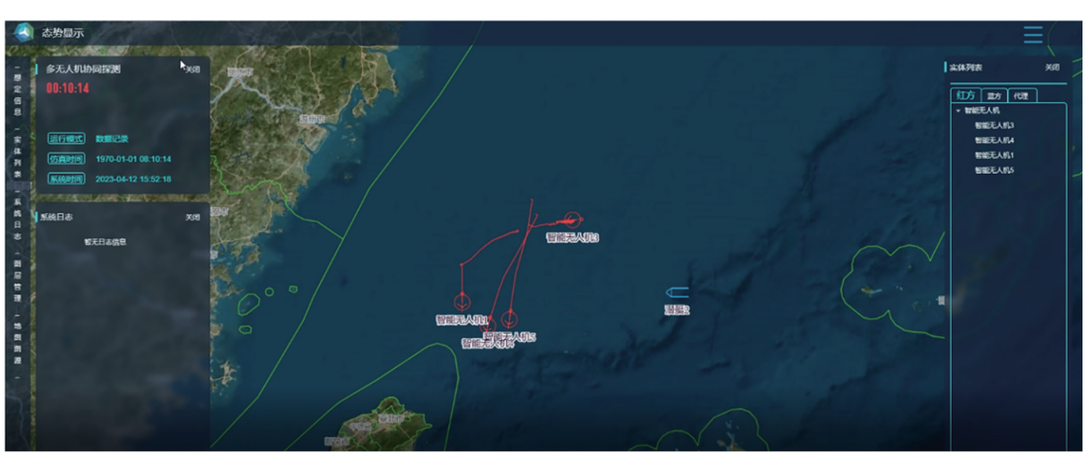

图12　仿真平台案例训练视图

Fig. 12　Simulation platform scenario case training process

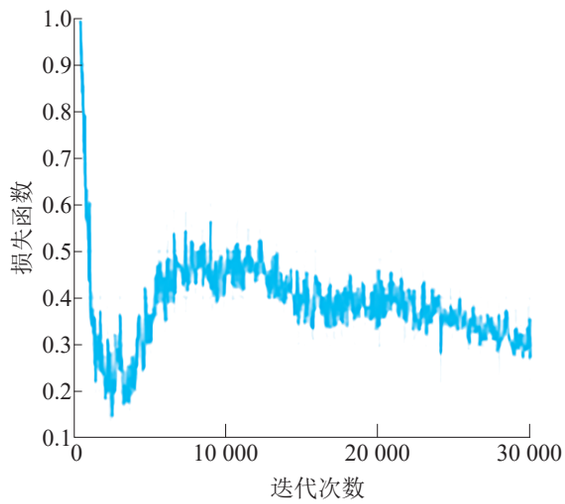

图 13　仿真平台案例 Actor 网络损失函数曲线
Fig. 13　Loss function curve of Actor network

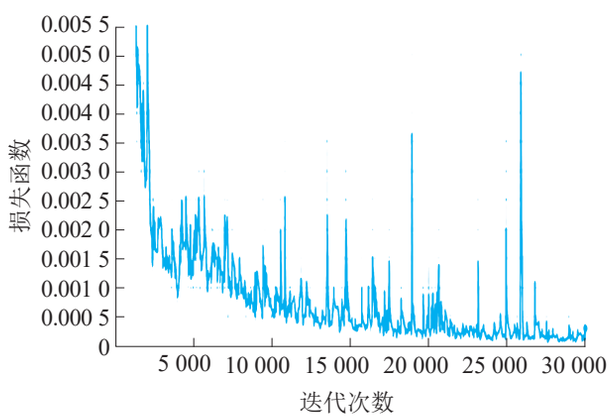

图 14　仿真平台案例 Critic 网络损失函数曲线
Fig. 14　Loss function curve of Critic network

5　结论

本文主要对多智能体协同决策仿真平台进行研究与开发，在研究过程中，在后端模型训练加速时采用异步非阻塞方式，对一般的推理—训练的阻塞过程进行改善，并同时对强化学习神经网络训练过程中的参数进行量化，分别将训练过程的速度提高了 29.08% 和 17.42%，提高了模型训练的效率。在多智能体强化学习算法的嵌入过程中，对多智能体强化学习算法的流程进行提炼，对仿真平台算法接口进行了统一设计及封装，使得各种强化学习算法能在仿真平台上进行统一调用，在保证算法形式多样性的同时，也对统一性进行了规范。为了保证算法在仿真平台上的运行效率，在仿真平台的案例想定下进行了 MADDPG 算法的验证。

目前的多智能体协同决策仿真平台已经集成单智能体强化学习、多智能体强化学习算法的训练，但 GPU 利用效率仍有待改善，可从算法的嵌入部分入手，对训练过程中所用到的方法及统一接口设计进行进一步的精炼和改善。后续也可将更多的强化学习算法嵌入平台，提高平台功能的多样性和实用性。

参考文献：

[1] 邹启杰, 蒋亚军, 高兵, 等. 协作多智能体深度强化学习研究综述[J]. 航空兵器, 2022, 29(6): 78-88.
Zou Qijie, Jiang Yajun, Gao Bing, et al. An Overview of Cooperative Multi-agent Deep Reinforcement Learning[J]. Aero Weaponry, 2022, 29(6): 78-88.

[2] Christiano P F, Leike J, Brown T W, et al. Deep Reinforcement Learning from Human Preferences[C]// Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc., 2017: 4302-4310.

[3] Brockman G, Cheung V, Pettersson L, et al. OpenAI Gym [EB/OL]. (2016-06-05) [2023-05-21]. https://arxiv. org/abs/1606.01540.

[4] Alibaba. Gym StarCraft[EB/OL]. [2023-05-16]. https://github.com/alibaba/gym-starcraft.

[5] 周亮, 王震, 王冠. 远程过程调用技术在分布式软件系统中的应用[J]. 航空电子技术, 2020, 51(4): 47-52.
Zhou Liang, Wang Zhen, Wang Guan. Application of Remote Procedure Calling Technology in Distributed Software System[J]. Avionics Technology, 2020, 51(4): 47-52.

[6] 张红杰. 深度强化学习训练与推理的性能优化[D]. 合肥: 中国科学技术大学, 2021.
Zhang Hongjie. Performance Optimization of Training and Inference of Deep Reinforcement Learning[D]. Hefei: University of Science and Technology of China, 2021.

[7] Martín Abadi, Agarwal A, Barham P, et al. TensorFlow: Large-scale Machine Learning on Heterogeneous Distributed Systems[EB/OL]. (2016-03-16) [2023-04-23]. https://arxiv.org/abs/1603.04467.

[8] Lowe R, Wu Yi, Tamar A, et al. Multi-agent Actor-critic for Mixed Cooperative-competitive Environments[C]//

DOI: 10.16182/j.issn1004731x.joss.23-FZ0821

Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc., 2017, 30: 6382-6393.

[9] Konda V R, Tsitsiklis J N. Actor-critic Algorithms[C]// Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2000: 1008-1014.

[10] Terry J, Black B, Grammel N, et al. PettingZoo: Gym for Multi-agent Reinforcement Learning[C]//Advances in Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates, Inc., 2021: 15032-15043.

[11] 王壮, 艾毅, 文旭光, 等. 航空器智能引导机动决策奖励重塑方法[J]. 科学技术与工程, 2023, 23(8): 3535-3543.
Wang Zhuang, Ai Yi, Wen Xuguang, et al. Reward Shaping for Intelligent Maneuver Decision Generation in Aircraft Guidance[J]. Science Technology and Engineering, 2023, 23(8): 3535-3543.