计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.241000036

基于深度神经网络的大样本作战仿真资源分配方法

叶 帅 李 豪 史培腾 黄昱霖
军事科学院战争研究院 北京 100091
(yeshuai09@nudt.edu.cn)

摘 要 随着人工智能的发展,作战实验呈现智能化趋势。大样本仿真是开展智能化作战实验的重要支撑,是解决作战实验变量因子多、组合复杂等问题的有效手段,具有样本数量大、速率要求高的特点。海量仿真样本的高速运行依赖于高性能硬件集群的高效调度,面临样本计算资源需求差异大、人工分配难的问题。如何精准预测并动态分配各个样本所需的计算资源,是提高大样本仿真效率的关键。为此,提出了一种基于深度神经网络(DNN)的大样本作战仿真计算资源预测模型。该方法首先构建了深度神经网络在环的仿真资源管理架构。其次,对作战仿真样本文件进行特征提取和学习构建深度神经网络预测模型。在大样本仿真运行时,通过在线预测每个样本所需的计算资源,实现海量作战仿真作业资源的精准预测与动态分配。测试结果表明,在千级样本的典型作战实验仿真场景中,相比于传统配置方法,提出的预测模型在10个高性能服务器节点上的完成时间减少了20.8%。

关键词:深度神经网络;大样本仿真;资源预测;集群管理
中图分类号 TP391.9

Deep Neural Network-based Resource Allocation for Large-scale Operation Simulation

YE Shuai,LI Hao,SHI Peiteng and HUANG Yulin
Academy of Military Science,Beijing 100091,China

Abstract With the development of artificial intelligence,operation experiments tend to be intelligent. Large-scale operation simulation is an important support for conducting intelligent operation experiments and an effective means to solve problems such as multiple variables and complex combinations in operation experiments. It has the characteristics of large sample size and high speed requirements. The high-speed operation of massive simulation samples depends on the efficient scheduling of high-performance hardware clusters,which faces the problems of large differences in computing resource requirements and difficult manual allocation. How to accurately predict and dynamically allocate the resources required for each sample is the key to improving the efficiency of large-scale simulation. This paper proposes a deep neural network(DNN)-based resource prediction model for large-scale operation simulation. The method firstly constructs a deep neural network in-loop simulation resource management architecture. Secondly,it constructs a deep neural network prediction model by extracting features and learning from combat simulation sample files. During the operation of large-scale simulation,it achieves accurate prediction and dynamic allocation of massive operation simulation job resources by online predicting the computing resources required for each sample. Test results show that in a typical operation experiment simulation scenario with thousands of samples,the proposed prediction model reduces the completion time by 20.8% on 10 high-performance server nodes compared to traditional configuration methods.

Keywords Deep neural network,Large-scale simulation,Resource prediction,Cluster management

1 引言

作战实验是指在可控、可测、近似真实的模拟对抗环境中,运用作战模拟手段研究作战问题的实验活动,是开展作战力量运用方式探索、作战方案推演、武器效能评估的重要手段,具有高可控、高灵活、可重复、高效费比等优势。随着人工智能技术的蓬勃发展,作战实验正呈现出智能化的趋势。在理论上,智能化科研(AI4R)[1]作为科学研究的第五范式,正在引领作战实验研究范式的转变,智能算法逐渐融入实验过程,指导实验因素调整,加快实验空间探索和发现。在技术上,随着无人自主作战平台的运用,作战实验力量要素的智能化水平也持续上升,深度学习、强化学习等人工智能技术融入作战力量,遂行观察、控制、打击、评估等全过程作战任务。

大样本作战仿真是开展智能化作战实验的重要支撑手段,其根据实验目的,调整天气、兵力、战法等实验因子,通过采样等方法形成大量的仿真样本,探索实验人员关切的问题空间。大样本仿真能够有效处理大规模作战行动中的复杂性,模拟众多作战单元和变量的相互作用。通过应用深度学

基金项目:国家自然科学基金青年科学基金(62102446)
This work was supported by the Young Scientists Fund of the National Natural Science Foundation of China(62102446).
通信作者:李豪(lihao07@nudt.edu.cn)

习和数据挖掘技术[2]，大样本作战仿真能够优化作战行动策略和价值网络，实现对抗条件下的方案推演分析和战法战术优化分析。此外，大样本作战仿真还支持动态推演，允许在仿真过程中根据实时反馈调整策略，并且能够利用并行计算能力处理大量的仿真数据和运算。大量的空间采样有助于开展复杂的实验设计和验证，确保仿真结果的科学性和可靠性[3]。通过分析大量的仿真数据，可以获得对作战行动的深入洞察，帮助理解作战规律和启发作战思想，为指挥人员认识作战过程提供支撑，给战法战术研究提供“准实践”环境。

由于样本数量多，计算资源需求高，智能化作战实验的运行往往依托于高性能仿真集群。在运行前，需要为每个实验样本分配 CPU 核、内存等硬件资源。由于样本之间的兵力、战法等实验因子存在差异，每个样本所需要的计算资源各不相同。硬件资源的分配方式将影响大样本仿真的效率。分配的资源过低将导致单个样本的计算资源不足，降低大样本仿真效率；而分配的资源过高，单个样本可以快速推演，然而由于集群硬件资源有限，运行的样本数量也有限，样本作业的排队时间将增长，同时由于资源空闲，集群的实际利用率不高。如何为海量的仿真样本作业快速合理地分配硬件计算资源，是高效开展大样本仿真的关键。

在传统的高性能集群计算环境中，资源分配主要依赖于人工在运行前预先设定[4]。这种人工配置的方法无法适应大样本仿真过程中的差异化资源需求，也无法精准预测每个样本的计算资源。本文针对大样本作战仿真计算资源差异大、人工分配难的问题，提出了一种基于深度学习的大样本仿真的计算资源预测方法，设计实现了基于该方法的资源预测工具。该方法首先通过深度神经网络（DNN）对样本文件进行特征提取和学习，预测每个样本所需的计算资源，并实现动态资源分配，以实现大样本仿真计算资源的快速预测与精准匹配，提升仿真运行效率，同时提升仿真硬件集群的利用率。

本文第 1 章介绍了集群资源分配的相关研究现状；第 2 章介绍了基于深度学习的大样本行为仿真计算资源预测方法的架构和使用流程；第 3 章给出了实验测试结果；最后总结全文。

2 相关工作

传统的高性能计算集群主要服务于大规模科学计算，主要运行 OpenFOAM[5]、风雷[6]等并行计算作业。大规模科学计算的资源分配方法主要包括静态资源分配、动态资源调整，以及基于机器学习的预测技术[7]。

静态资源分配方法通常在作业开始前根据经验配置资源需求，并在整个计算过程中保持不变。Lee 等提出的资源分配框架采用了静态分配策略[8]，提高了云计算环境中 YARN 资源共享的公平性和效率，但无法满足大样本仿真中计算资源的差异化需求。

动态资源分配技术实时监控仿真负载，通过动态调整资源分配来提高资源利用率。Chen 等提出了一种自适应资源管理器，能够根据实时反馈来动态调整资源分配[9]。然而，这些方法通常依赖于复杂的调度算法，且主要面向大数据处理作业需求，不适用于大样本仿真作业资源管理。

基于机器学习的预测技术利用历史数据训练模型，在线预测未来的资源需求。在深度学习中，深度神经网络（DNN）具有强大的特征学习能力，在资源预测方面显示出了巨大潜力。Hu 等提出的 ReLoca 系统使用 DNN 来优化 Apache Spark 系统的大数据分析应用程序，平均作业完成时间下降了 29%[10]。ReLoca 在数据并行作业场景下表现出色，但该方法主要是针对单个作业的资源需求进行预测，不适用于大样本作战仿真。除了深度学习以外，随机森林[11]、支持向量机[12]、集成学习方法[13]也被应用于资源预测，但这些算法本身较复杂，预测速度低，无法实现算法在环的高效预测。

3 资源预测模型设计

3.1 概述

大样本作战仿真资源预测工具是开展智能化作战实验的功能模块，其功能定位如图 1 所示。实验人员通过实验设计工具在实验空间内生成大量的仿真样本，指定样本的仿真引擎执行路径。资源预测工具为这些仿真样本设置 CPU 核数等计算资源，形成仿真作业。设置完成后，通过集群管理软件发布仿真样本作业，并根据资源需求调度执行。同时，资源预测工具利用集群管理工具采集运行数据，进行学习训练。

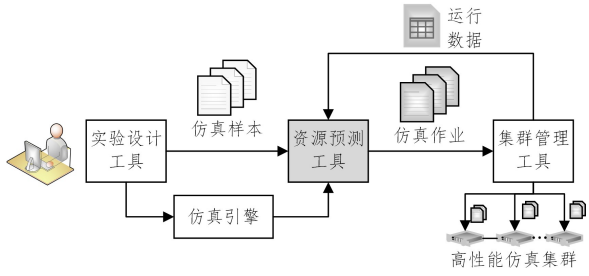


图 1 资源预测工具的功能定位图

Fig. 1 Function of the resource prediction tool

资源预测工具主要负责在线预测并设置各个仿真样本的计算资源需求，由数据收集、特征提取、神经网络模型 3 个子模块组成，具体如图 2 所示。

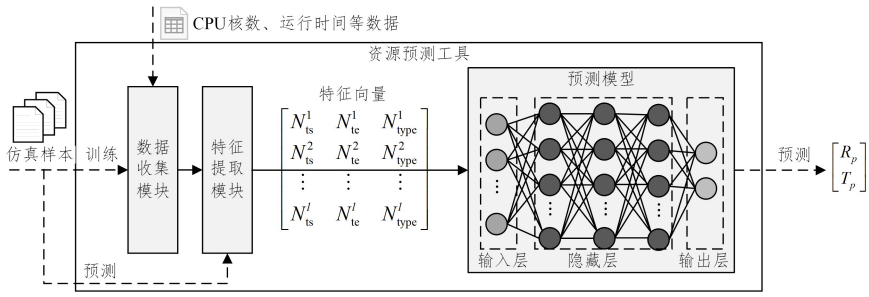


图 2 资源预测工具组成

Fig. 2 Component of the resource prediction tool

3.2　数据收集

数据收集模块负责收集大样本仿真的相关运行数据，包括 CPU 核心数、仿真作业的运行时间、排队时间、完成总时间等，在经过处理后输入后续模块。为捕获不同 CPU 核心数配置下样本仿真作业的性能表现与资源消耗，首先对每个样本的运行 CPU 核心数采样，记为$\{R_1,R_2,\cdots,R_n\}$，n 为采样数，统计相应配置下的运行完成时间数据，记为$\{T_1,T_2,\cdots,T_n\}$。由于硬件资源有限，分配的 CPU 核数越多，单个节点上能够运行的作业数越少。具体而言，R_i 配置下单节点能够支持的作业数$J_i=\lfloor N/R_i \rfloor$，其中 N 为单节点上 CPU 总核数。定义单位时间内单节点能够完成的相对作业吞吐量为：

$$O_i=\frac{J_i}{T_i}$$

为方便对比，下文所给出的作业吞吐量均为相对值，即$O_i^*=O_i/O_1$。在此基础之上，定义最佳 CPU 核心数配置 R_{opt}：

$$R_{\text{opt}}=\arg\max_{k\in[1,n]}\{O_k\}$$

即单节点上单位时间内作业吞吐量的最大值配置。

在数据收集后，进行异常值识别与处理、数据归一化以及特征编码等预处理操作，将原始数据转换为适合模型训练的格式。在实现时，数据收集模块通常与集群管理工具协同处理，以实现数据的自动化收集和处理。

3.3　特征提取

特征提取模块主要是在数据收集后，对样本文件进行分析，提取关键特征。根据作战仿真样本的特点，仿真样本的运行依赖于仿真引擎的具体实现，本文中的仿真引擎主要采用行为树引擎驱动仿真运行。在运行时，行为树引擎将在作战行为开始时分配逻辑线程，在行为结束时释放线程，不同的行为类型用节点类型区分。因此，为捕捉每个样本所对应行为树的结构特性和动态特性，提取仿真样本中第 i 个作战行为的开始时间、结束时间、节点类型作为特征提取的核心指标，记为$C^i=\{N_{\text{ts}}^i,N_{\text{te}}^i,N_{\text{type}}^i\}$，以表征仿真样本对计算资源的需求。则包含 l 个作战行为的仿真样本对对应的特征向量即为$\boldsymbol{V}=\{\{N_{\text{ts}}^1,N_{\text{te}}^1,N_{\text{type}}^1\},\{N_{\text{ts}}^2,N_{\text{te}}^2,N_{\text{type}}^2\},\cdots,\{N_{\text{ts}}^l,N_{\text{te}}^l,N_{\text{type}}^l\}\}$。该向量将作为深度神经网络模型的输入。为了消除不同节点类型带来的影响，在特征向量中将节点类型信息映射为数字信息，此外特征提取模块还需识别并处理缺失值和异常值，确保数据集的完整性，以避免对模型训练产生负面影响。

3.4　预测模型

预测模型主要负责从样本特征中学习并预测资源需求。该模型采用经典的深度神经网络，由输入层、隐藏层以及输出层等组成多层感知器架构。其中，输入层接收来自特征提取模块的高维数据 V。隐藏层包含 3 个全连接层，每层包含 K 个神经元。考虑到预测模型的性能需求和鲁棒性需求，在每个隐藏层设置 ReLU 激活函数，通过引入非线性特性，增强模型对复杂模式的捕捉能力。预测模型的输出层分别给出每个样本的 CPU 核心数需求预测(R_p)和运行时间预测(T_p)。

预测模型使用均方误差(MSE)作为损失函数，采用Adam优化器来更新模型的权重和偏置。Adam 优化器结合了动量(Momentum)和 RMSProp 的优点，其适应性强，能够在各种不同的数据集上提供稳定的性能。

4　实验测试

本文设计了对比实验，以说明大样本仿真资源预测方法的有效性。实验运行的硬件环境是由 10 台计算服务器组成的高性能仿真集群。每台服务器搭载两颗 Intel Xeon Gold 6240R 处理器，处理器主频为 2.4 GHz，包含 24 个物理核心，内存大小为 16 * 64 GB。实验运行的操作系统为 CentOS 8，集群管理软件为 slurm 20.11.9。

DNN 使用 PyTorch 2.1.2[14]，在实现时，隐藏层的神经元个数设为 128。

4.1　实验设计

本实验面向典型场景，考虑天气、作战行动时间等实验因子，共设计生成 1 200 个仿真样本，编号记为{1,…,1200}。其中，{1001,…,1200}编号为 200 个样本作为预测模型的训练集，编号为{1,…,1000}的仿真样本作为本次实验的测试集。实验记录对比不同资源配置方法下，1 000 个仿真样本作业的运行完成时间。

为方便表述，文中使用的符号声明如下：

S_k：使用静态资源配置方法，CPU 核数设置为 k。

D_{DNN}：使用动态资源配置方法，基于 DNN 神经网络设置 CPU 核数。

$R_{\#}^i$：第 i 个仿真样本使用 # 方法所分配的 CPU 核数，其中$\#=\{S_k,D_{\text{DNN}}\}$。

$T_{\#}^i$：第 i 个样本使用 # 方法分配资源的运行时间，单位为秒(s)。

$W_{\#}$：使用 # 方法分配资源的排队时间，单位为秒(s)。

$\bar{T}_{\#}$：使用 # 方法分配资源的平均运行时间，单位为秒(s)。

$\bar{W}_{\#}$：使用 # 方法分配资源的平均排队时间，单位为秒(s)。

T_{DNN}：使用预测模型时，预测模型本身的时间开销，单位为秒(s)。

$Total_{\#}$：1 000 个样本使用 # 方法分配资源的总运行时间，单位为秒(s)。

所有数据均为 3 次运行结果的平均值。

4.2　实验结果

为确定静态配置方法下的最佳 CPU 配置 R_{opt}，实验统计了$\{S_1,S_2,S_4,S_8,S_{16},S_{32}\}$时样本的时间开销，通过计算单节点的相对作业吞吐量 O，分析每个样本的最佳 CPU 配置。图 3 给出了 1 号样本作业在不同配置下的相对作业吞吐量。

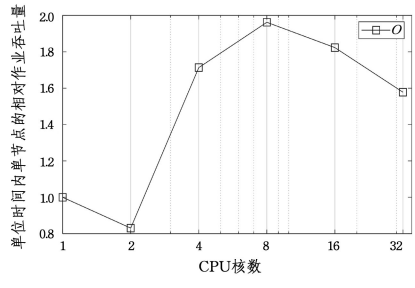


图 3　不同配置下 1 号样本作业相对作业吞吐量

Fig. 3　Throughput of the No. 1 sample job with different CPU configure

从图 3 中可以看出，1 号样本的 CPU 配置为 8 时，单位时间内单节点的作业吞吐量最大，当分配的 CPU 核数超过

8 时，单样本的完成时间减少，但由于排队时间增长，多个具有 1 号样本特征的作业完成时间也将增大，因此，1 号样本作业的最佳 CPU 配置 $R^1_{\text{opt}}=8$。图 4 给出了 1 000 个样本的最佳 CPU 配置统计。

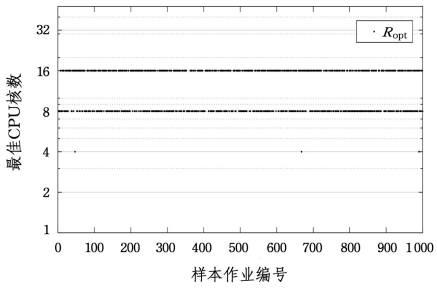


图 4 1 000 个样本作业的最佳 CPU 核数配置

Fig. 4 Outperform configure of the CPU cores of all the 1 000 sample jobs

从图中可以看出，1 000 个测试样本的最佳 CPU 配置集中在 8 和 16，少部分样本的最佳 CPU 配置为 4，这表明不同样本作业的 CPU 需求存在差异。

表 1 列出了 1 000 个样本作业的总耗时、平均运行时间和平均排队时间。

表 1 使用静态配置方法(S_k)时 1 000 个样本的总耗时

Table 1 Total time cost of the test with static configure(S_k)

S_k	$Total_S$	$\overline{T}_S$	$\overline{W}_S$
$k=1$	7748.5	3410.3	4338.2
$k=2$	7834.5	1664.0	6170.5
$k=4$	5641.8	630.7	5011.1
$k=8$	3923.7	226.2	3697.5
$k=16$	**3887.9**	111.2	3776.7
$k=32$	5220.1	50.7	5169.4

从表 1 中可以看出，CPU 核数 k 设置为 16 时，1 000 个样本的完成时间最短，为 3 887.9 s。当 k 设置为 32 时，单个样本作业的完成时间减小，但由于单节点 CPU 资源受限(48 核)，单节点上能够同时运行的作业数减小，样本作业的平均排队时间大幅增大，1 000 个样本的总完成时间反而变长。在下文的对比实验中，静态配置 CPU 核数 k 取 16。

在使用基于模型预测的动态配置方法时，1 000 个样本的 CPU 核数配置如图 5 所示。

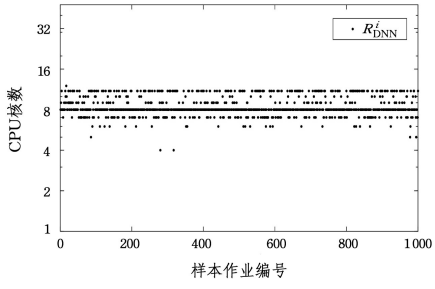


图 5 1 000 个样本作业使用 DNN 模型时的 CPU 预测配置

Fig. 5 Prediction of the CPU cores of all the 1 000 sample jobs based on DNN

从图 5 中可以看出，在使用 DNN 预测时，CPU 核数的配置发生动态变化，大部分预测集中在[6,12]内，少部分分散在 4 附近。

基于预测模型的方法与静态配置方法 S_{16} 的 1 000 个样本的运行时间对比以及排队时间对比分别如图 6、图 7 所示。

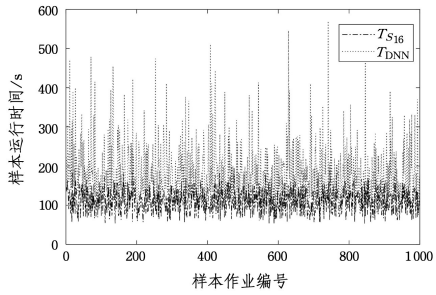


图 6 1 000 个样本作业使用 DNN 模型以及静态配置方法的运行时间对比

Fig. 6 Comparison of running time based on DNN and running time based on static configure

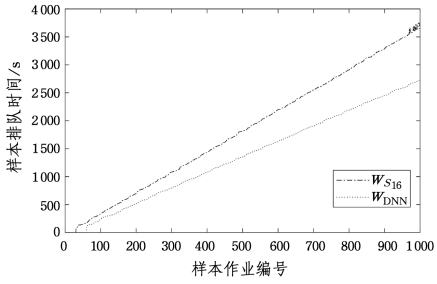


图 7 1 000 个样本作业使用 DNN 模型以及静态配置方法的排队时间对比

Fig. 7 Comparison of pending time based on DNN and pending time based on static configure

从图 6 中可以看出，在使用模型预测方法时，由于模型预测分配的 CPU 核数集中在[6,12]内，小于静态分配方法的 16 核配置，因此，样本作业的运行时间比静态方法长。同时，从图 7 中可以看出，由于硬件资源受限，作业分配的核数越少，后续作业的排队时间越短，因此，使用模型预测方法时，样本作业的排队时间相比使用静态方法更短。表 2 列出了完成 1 000 个样本的总耗时对比。

表 2 1 000 个样本作业的总耗时对比表

Table 2 Total time cost of the test with different methods

Method(#)	$Total_{\#}$	$\overline{T}_{\#}$	$\overline{W}_{\#}$
S_{16}	3887.9	111.2	3776.7
D_{DNN}	3081.0	179.0	2902.0

从表 2 中可以看出，使用模型预测时，1 000 个样本作业的总耗时为 3081.0 s，相比 S_{16} 方法的总耗时减少 20.8%。其中，单样本作业的模型预测过程本身耗时约为 0.006 s，表明预测模型本身的高效性。

结束语 本文针对大样本仿真时资源配置所面临的差异性大、人工配置难的问题，设计了基于 DNN 的大样本仿真计算资源预测方法，通过在线预测样本的计算资源，有效提升了仿真效率和硬件集群的资源利用率。本文在 10 个服务器节点组成的仿真集群上开展了实验验证，实验结果表明，与静态配置相比，基于 DNN 模型预测的动态资源分配策略千级样本的运行时间减少 20.8%。下一步，将本文预测模型适应于更多的大样本仿真场景，以提升预测模型的适应性。

参 考 文 献

[1] LI G J. AI4R：The fifth scientific research paradigm[J]. Bulletin of Chinese Academy of Sciences，2024，39(1)：1-9.

[2] WANG J，CHEN R，TANG L. Research on the technology of Army intelligent combat exercise with large sample[J]. National Defense Science and Technology，2020，41(1)：41-44.

[3] WU X P，CAO D，DONG S，et al. Large-sample simulation design and verification of system simulation platform[J/OL]. Computer Measurement and Control，1-8[2024-05-17]. http://kns. cnki. net/kcms/detail/11. 4762. TP. 20240221. 2108. 006. html.

[4] LU W R. Research and improvement of container cloud resource scheduling strategy based onKubernetes[D]. Hangzhou：Zhejiang Sci-Tech University，2023.

[5] JASAK，ALEKSANDAR J H，TUKOVIĆ，et al. OpenFOAM：A C++ Library for Complex Physics Simulations[C]// Coupled Methods in Numerical Dyamics. 2007.

[6] ZHAO Z，et al. Design of general CFD software PHengLEI [J]. Computer Engineering & Science，2020，42(2)：210-219.

[7] LIU H，DONG X Y，YANG Z H. BiGRU-LGB cloud load prediction model based on stacking framework[J]. Journal of Xidian University，2023，50(3)：83-94，104.

[8] THINAKARAN P，GUNASEKARAN J R，SHARMA B，et al. Kube-Knots：Resource Harvesting through Dynamic Container Orchestration in GPU-based Datacenters[C]// 2019 IEEE International Conference on Cluster Computing(CLUSTER). 2019：1-13.

[9] TANG S，LEE B S，HE B. Fair Resource Allocation for Data-Intensive Computing in the Cloud[J]. IEEE Transactions on Services Computing，2018，11(1)：20-33.

[10] CHEN Z，HU J，MIN G，et al. Adaptive and Efficient Resource Allocation in Cloud Datacenters Using Actor-Critic Deep Reinforcement Learning[C]// IEEE Transactions on Parallel and Distributed Systems. 2022：1911-1923.

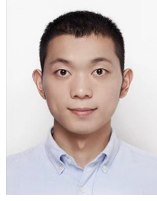
[11] HU Z，LI D，ZHANG D，et al. ReLoca：Optimize Resource Allocation for Data-parallel Jobs using Deep Learning[C]// IEEE INFOCOM 2020－IEEE Conference on Computer Communications. 2020：1163-1171.

[12] CHEN S Y，ZHUANG Y，LI J. Multiple load prediction model for mobile cloud computing based on LSTM network[J]. Computer and Modernization，2021(6)：74-85.

[13] LIANG R H，XIE X L，ZHAI Q H，et al. Research on container cloud load prediction based on improved stacking ensemble model[J]. Computer Applications & Software，2023，40(12)：48-55，100.

[14] PASZKE A，GROSS S，MASSA F，et al. Pytorch：An imperative style，high-performance deep learning library[C]// Advances in Neural Information Processing Systems. 2019.

YE Shuai，born in 1991，Ph.D，research assistant. His main research interest is parallel operation simulation.



LI Hao，born in 1990，Ph.D，research assistant. His main research interest is parallel operation simulation.