

浙江大學

本科生毕业论文（设计）



题目 基于特征融合的容器调度优化研究

姓名与学号 张云翔 3200100213

指导教师 张帆

年级与专业 2020 级 海洋工程与技术

所在学院 海洋学院

提交日期

摘 要

容器调度,是经典的组合优化问题。论文回顾了容器重定位算法的发展历程,大致呈显数学模型的建构、启发式方法的融入和机器学习的兴起三个阶段。以容器布局为数据的机器学习研究尚处于起步阶段,缺乏详实的理论基础和实践探索。论文以容器布局数据的特征提取研究为主线,主要开展了以下研究工作:(1)系统全面地对比了三种主流的容器布局特征提取方法——基于栈内局部信息的人工特征提取器、基于卷积神经网络(CNN)的特征提取器和基于长短期记忆递归神经网络(LSTM)的特征提取器。实验结果表明,基于LSTM的特征提取器在LB1特征提取方面表现突出,而基于CNN的特征提取器在大规模容器布局重定位次数最优解识别任务中表现更佳;(2)提出了一种基于特征融合的重定位次数最优解识别模型,通过特征拼接方式融合了栈内局部信息、CNN与LSTM分别所关注的特征尺度,超越了各自独立进行特征提取的效果,并研究了训练集规模的边际效应和类别均衡的重要性;(3)提出了基于特征融合模型的容器调度算法,在公开数据集上的平均容器重定位次数优于现有基于机器学习的近似解算法的最好成绩。论文的研究为未来以容器布局为数据的机器学习研究提供了较为全面的参考。

关键词: 容器调度; 特征融合; 集装箱重定位; 容器布局; 机器学习

Abstract

Container scheduling is a classic combinatorial optimization problem. This paper reviews the development of container relocation algorithms, highlighting three main stages: the construction of mathematical models, the integration of heuristic methods, and the rise of machine learning. The study of machine learning applied to container layout data is still in its early stages, lacking a solid theoretical foundation and sufficient practical exploration. Centered on the research of feature extraction from container layout data, this paper undertakes the following main research tasks: (1) A comprehensive comparison of three mainstream container layout feature extraction methods: manually crafted feature extractors based on stack-local information, convolutional neural network based feature extractors, and long short-term memory based feature extractors. Experimental results indicate that LSTM-based feature extractors excel in extracting LBI features, while CNN-based feature extractors perform better in identifying the optimal number of relocations for large-scale container layouts; (2) The proposal of an optimal relocation identification model based on feature fusion. This model fuses features by concatenating those obtained from stack-local information, CNN, and LSTM, surpassing the performance of independent feature extraction methods. The study also examines the marginal effects of training set size and the importance of class balance; (3) The introduction of a container scheduling algorithm based on the feature fusion model, which achieves an average number of container relocations on public datasets that surpasses the best performance of existing machine learning based approximate algorithms. The research presented in this paper provides a comprehensive reference for future studies on machine learning with container layout data.

Keywords: Container Scheduling; Feature Fusion; Container Relocation; Container Layout; Machine Learning

目 录

1 绪论.....	1
1.1 研究背景与意义.....	1
1.2 研究现状.....	2
1.3 论文的主要研究工作.....	3
1.4 论文的组织结构.....	3
2 问题描述、评价指标与实验设置.....	5
2.1 问题描述.....	5
2.1.1 容器重定位问题.....	5
2.1.2 容器布局的标签.....	6
2.2 评价指标.....	9
2.2.1 回归模型经典评估方法.....	9
2.2.2 针对容器调度的回归模型实用评估方法.....	9
2.3 论文的实验设置.....	10
3 容器布局特征提取器研究.....	11
3.1 容器布局的标准化方法与有效性实验.....	11
3.1.1 容器布局标准化方法.....	11
3.1.2 容器布局标准化的有效性实验.....	12
3.2 基于栈内局部信息的人工特征提取器.....	12
3.2.1 原理与识别网络设计.....	13
3.2.2 实验情况——通过与基于原始数据的全连接网络对比.....	14
3.3 基于卷积神经网络的特征提取器.....	15
3.3.1 原理与识别网络设计.....	16
3.3.2 实验情况——通过与基于两类卷积核的 CNN 对比.....	16
3.4 基于长短期记忆递归神经网络的特征提取器.....	18
3.4.1 原理与识别网络设计.....	18
3.4.2 实验情况——通过本章介绍的三种特征提取器对比.....	19
4 基于特征融合的容器调度优化方法.....	22

4.1 利用多个特征提取器实现特征融合	22
4.1.1 模型架构.....	22
4.1.2 特征融合策略的有效性实验.....	22
4.1.3 训练集规模的边际效应实验.....	23
4.1.4 训练集类别均衡实验.....	25
4.2 容器调度实验.....	25
4.2.1 容器调度算法.....	26
4.2.2 数据集设置和模型训练.....	26
4.2.3 容器调度结果分析.....	28
5 总结与展望.....	28
参考文献.....	29

1 绪论

1.1 研究背景与意义

在国际航运领域，“容器”即为“集装箱”。集装箱广泛应用于船舶、港口、航线、公路、中转站、桥梁、隧道、多式联运相配套的物流系统，使码头装卸时间大大缩短，让动辄几十吨的货物实现了标准化操作，直接推动了全球贸易便利化与全球经济一体化进程^[1]。集装箱的调度效率，对港口的吞吐效率产生重要影响。官方数据^[2]显示，2023年上海港集装箱吞吐量突破4900万标准箱，连续14年位居世界第一，更高吞吐使港口运营面临更大挑战。

在实际的港口运营中，大量的集装箱在存储场所成层堆叠，形成数个被称为贝（bay）的二维单元。以上海港为例，每贝可存放6层（tier）10栈（stack）的集装箱。由于集装箱的到达与移出存在时间差，起重机进行调度时常常遇到目标集装箱被其他集装箱阻挡的情况，因此不可避免地需要进行若干次重定位操作。但是，不恰当的重定位操作反而导致后续目标被进一步阻挡，因而智能高效的重定位算法有利于提升集装箱¹调度的效率。

这样的运筹调度场景被抽象为容器重定位问题（container relocation problem, CRP），也以块重定位问题（block relocation problem, BRP）闻名，旨在以最少的重定位次数以给定的优先级顺序取出单个贝中的所有集装箱。根据是否允许出现相同的优先级，研究可划分为异优先级CRP（CRP with distinct priorities）与群优先级CRP（CRP with group priorities）。根据重定位操作起点是否受限于目标容器所在栈，研究可划分为有限制CRP（Restricted CRP, RCRP）与无限制CRP（Unrestricted CRP, UCRP）。

根据求解目标是否是精确解，对CRP的研究可划分为精确解算法（exact algorithms）和近似解算法（approximate algorithm）。根据求解方法的不同，研究大致有分支定界算法（branch and bound algorithms）、启发式算法（heuristic algorithms）、元启发式算法（metaheuristic algorithms）与借助机器学习的算法（algorithms based on machine learning）四种方向。研究^[3]证明容器重定位问题是NP-hard问题，因此精确算法的时间复杂度往往是不可接受的。一些启发式与元启发式方法虽然能够借助一些贪心策略极大改进解的搜索速度，然而准确度往往

¹ 为了术语的一致性，后续将统一使用“容器”取代“集装箱”阐述其重定位问题。

大打折扣，最坏情况下只能获得局部最优解。随着人工智能技术的迅猛发展，在许多组合优化问题领域，借助机器学习的算法愈发受到关注，有望进一步改进容器重定位问题的近似解算法。

1.2 研究现状

容器重定位问题的精确解算法研究主要有一般整数规划模型与启发式搜索模型。Kim和Hong^[4]最早将CRP作为组合优化问题进行研究，构造出适用于群优先级CRP的分支定界算法。他们使用了一种简单而实用的重定位次数下界的定义，根据该定义每有一个容器阻挡了其它容器，重定位次数就要至少增加一次。同样基于此定义，多项研究^[5-6]构造出适用于异优先级RCRP的分支定界算法。模型的数学建构方面，Caserta等^[3]提出了一种适用于小型异优先级CRP的0-1整数规划模型，此后多项研究^[6-8]对其中针对RCRP的模型做出了纠正与改进。Petering and Hussein^[9]提出了一种适用于异优先级UCRP的混合整数规划模型。de Melo da Silva^[10]充分考虑四种CRP变体的情况，提出了统一的整数规划模型。许多研究^[11-14]持续从不同角度以进一步改进模型效率为目标而努力。

朱文斌等^[15]最早应用启发式搜索算法求解CRP，构造出适用于异优先级UCRP的IDA*算法，相比一般整数规划模型求解速度得到极大改善。此外，他们进一步提出更好的重定位次数下界LB3，并基于此构造出适用于异优先级RCRP的IDA*算法。随后，Expósito Izquierdo等^[16]构造出了适用于异优先级CRP的A*算法。启发式方法也在近似解算法中得到应用，这里也将笼统地包括对元启发式算法研究的回顾。Bacci等^[17]构造出了一种适用于异优先级RBRP的束搜索算法，并利用了他们提出的另一个下界UBALB（unordered blocks assignment lower bound）。UBALB与上述下限的主要区别在于，它没有忽略栈的高度限制。Jovanovic等^[18]提出了一种蚁群优化元启发式算法，该算法适用于异优先级CRP。Gulić等^[19]应用遗传算法于RCRP，得到了较好的结果。

随着机器学习技术应用于越来越多的工程领域，许多研究也将机器学习与容器调度领域相结合。Hottung等^[20]结合了深度卷积神经网络（DNN）和启发式树搜索算法，利用机器学习协助搜索树的分支和剪枝，用于解决容器预编组问题（CPMP）。张灿荣等^[21]利用许多小规模实例的最优解，基于随机森林和联合学习方法训练了一个分支修剪器，然后将其应用到分支定界算法和束搜索算法中以

提高性能。Ye等^[22]采用随机森林、支持向量机等多种监督学习方法，基于归因分析确定影响重定位次数重要的特征因素。近年来，利用强化学习基于一定的启发式规则进行容器调度的研究方兴未艾，Jiang等^[23]、Wei等^[24]均提出了针对CRP的强化学习架构，表现出良好的性能。

1.3 论文的主要研究工作

通过回顾容器重定位算法的发展历程，大致呈现从数学模型的建构、启发式方法的融入到机器学习的兴起三个阶段。然而，机器学习对于容器布局数据的研究尚处于起步阶段，相比其他领域仍然缺少详实的理论基础和充分的实践探索。针对这一问题，论文的研究工作以容器布局的特征提取研究为主线，旨在为后续以容器布局为数据的机器学习研究提供较为全面的参考，主要工作如下：

(1) 系统全面地对比三种主流的容器布局特征提取器方法。归纳了三种常见的容器布局特征提取器，分别是基于栈内局部信息（Stack-local information）的人工特征提取器、基于卷积神经网络（CNN）的特征提取器与基于长短期记忆递归神经网络（LSTM）的特征提取器。探究不同的容器布局特征提取器在不同标签识别任务——重定位次数下界 LB1 以及重定位次数最优解，和不同容器布局规模——以 10-4 为代表的小规模容器布局以及以 7-6 为代表的大规模容器布局的性能表现。实验结果表明，基于 LSTM 的特征提取器在提取有关 LB1 特征方面具有突出表现，在大规模容器布局重定位次数最优解识别任务中基于 CNN 的特征提取器性能表现更好；

(2) 提出一种基于特征融合的重定位次数最优解识别模型。基于几种独立的特征提取方法以特征拼接的方式实现特征融合，充分利用了栈内局部信息、CNN 与 LSTM 所分别关注的特征尺度，达到超越各自独立进行特征提取的效果，并研究了训练集规模的边际效应与类别均衡的重要性；

(3) 提出基于特征融合模型的容器调度算法，在公开数据集上的平均容器重定位次数超越了论文中基于机器学习的近似解算法的最好成绩。

1.4 论文的组织结构

论文对组织结构做出如下安排：

第一章是绪论，旨在使读者对容器调度特别是基于机器学习的容器重定位问题研究的背景和已有的进展建立充分的认识，并介绍了论文的主要工作。

第二章是为介绍具体研究工作而做的铺垫，旨在阐述容器重定位问题的具体定义，介绍本研究将容器布局数据作为机器学习技术研究对象的主要思路，介绍了几种重要的标签，并介绍了针对回归模型的评价指标和本研究的实验设置。

第三章介绍了第（1）项主要研究工作，旨在系统全面地对比三种主流的容器布局特征提取方法，探究不同的容器布局特征提取器在不同标签识别任务和不同容器布局规模的性能表现。

第四章介绍了第（2）与第（3）项主要研究工作，旨在介绍所提出的特征融合策略的有效性，探究训练集规模对重定位次数最优解标签识别的边际效应和训练集类别均衡策略的有效性，最后将提出的特征融合重定位次数最优解识别模型应用到容器调度的实际场景中并基于公开数据集分析其表现。

第五章是总结与展望。回顾论文的全部研究内容，指出研究存在的问题与不足，对下一步以容器布局为数据的机器学习研究工作给出展望。

2 问题描述、评价指标与实验设置

2.1 问题描述

本章节阐述容器重定位问题的具体定义，介绍本研究将容器布局数据作为机器学习技术研究对象的主要思路，并介绍了几种重要的标签。

2.1.1 容器重定位问题

考虑多层 (tiers) 多栈 (stacks) 的有限制异优先级 CRP，将容器布局含有的容器总数记为 N ，层数记为 T ，栈数记为 S ，显然 $N \leq T \times S$ 。所有容器需要按照一个固定且已知的顺序依次堆出。每个容器的堆出顺序（即优先级）是唯一的，并根据堆出优先级编号，即 $1, 2, \dots, N$ 。于是我们将优先级编号为 i 的容器记为容器 i 。编号较小的容器意味着具有更高的堆出优先级，因此应被更早堆出。容器布局中的最高优先级容器称为目标容器，在优先级编号归一化后，它的优先级编号一定是 1。当容器 1 同一栈的上层还存在另一个或几个优先级更低的容器时，后者应该被首先重定位 (relocation) 到容器布局的其它栈，以便容器 1 能被顺利堆出。

根据实际的应用场景和研究的严谨性，论文对研究范畴做出以下假设：层数和栈数是预先给定的，在容器调度过程中不会改变；在所有容器堆出完毕之前，不会有新容器进入布局；只有目标容器所在栈上层的容器可以被作为被重定位的对象（有限制 CRP）；容器重定位的接收栈不能是满栈，即容器总数已经达到层数的栈。

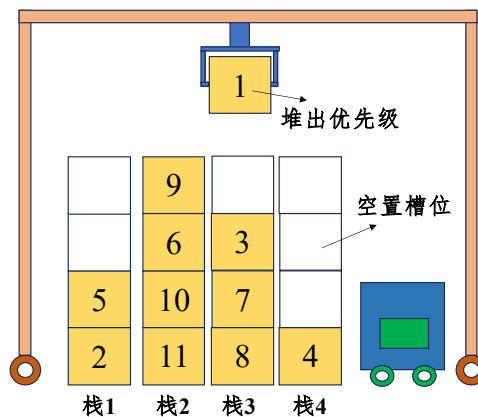


图 2.1 容器布局示意图

图 2.1 展示了一个包含 11 个集装箱的 4 层 4 栈容器布局的容器重定位过程，

起重机正在堆出容器 1。容器槽位（slot）中的数字代表容器的堆出优先级。在该示例中，容器 1 由于此前在其栈容器中位于最上层被直接堆出。但是，本应按照顺序在下一个被堆出的容器 2 却被其上层的容器 5 阻挡。因此，不得不将容器 5 重定位到其他栈中。

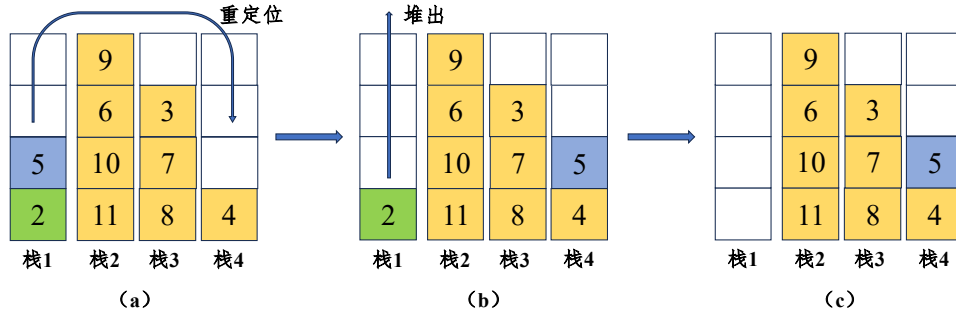


图 2.2 容器重定位过程示意图

这样重定位的过程如图 2.2(a)-(c)所示。容器 5 可以被重定位到栈 3 或者栈 4，因为这两个栈还未满。在该示例中，容器 5 被重定位到栈 4，于是容器 2 便可以堆出。每次重定位都会导致一个新的容器布局，并影响后续重定位的决策。不恰当的决策可能会在未来的容器调度过程中增加额外的重定位次数，使调度成本提高。因此，CRP 的目标即是以最少的重定位次数按照给定的顺序堆出所有 N 个容器。

2.1.2 容器布局的标签

经过前一小节对 CRP 的详细介绍，我们已经对其研究对象——容器布局有了直观的感受。随着机器学习技术在图像、自然语言、语音等数据领域的研究层出不穷，将容器布局数据作为研究对象的却非常少。面对 CRP 的实际需要，将机器学习应用于容器布局数据的根本使命在于抽象出有关所需重定位次数的特征，以解答“什么样的容器布局是更好的容器布局”这一问题，智能地在重定位后所有可能的新布局中做出选择。

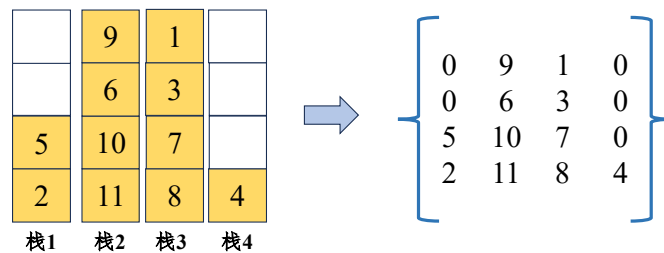


图 2.3 容器布局的数据结构

为了实现这一目标，同其他数据类型一致地，应该首先界定容器布局的数据结构、值域及其标签。如图 2.3 所示，容器布局数据是一通道二维结构，空槽位取值为 0，其余槽位取值为对应堆出优先级编号。本小节将详细介绍容器布局的几种标签，此后在第三章中将设计实验探究容器布局特征提取器对其中两种经典的标签的识别性能。

	9	1	
	6	3	
5	10	7	
2	11	8	4
栈1	栈2	栈3	栈4

图 2.4 容器布局的 LB1

在 CRP 的传统研究中，重定位次数下界（Lower Bound, LB）是有利于更快获得最优解或近似解的重要启发式信息。在基于强化学习方法的几项研究中该信息被作为标签用于网络参数的初始化。在 Kim 和 Hong^[4]最早将 CRP 作为组合优化问题进行研究时，就提出了一种简单而实用的定义，后被称为 LowerBound1（LB1）。根据 LB1 的定义，每有一个容器阻挡了其它容器，重定位次数就要增加一次。图 2.4 用棕色标注了示例容器布局的阻挡容器，它们的个数（2）就是该容器布局的 LB1 值。

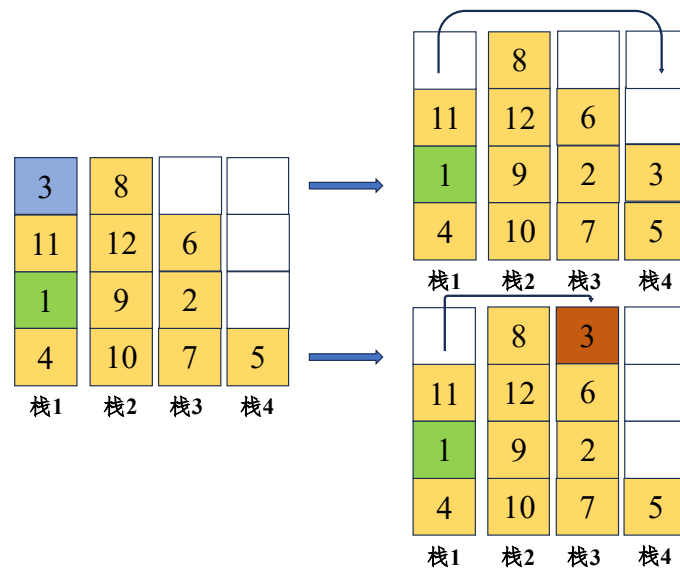


图 2.5 容器布局的 LB2 的检查步骤

LB1 是简单明了但十分不准确的。研究^[15]进一步优化了 LB1，分别提出了 LowerBound2 (LB2) 和 LowerBound3 (LB3)。观察图 2.5 的容器重定位过程，可以看到阻挡了容器 1 的重定位容器 3 有两个可能的目的地——栈 3 和栈 4。然而，如果选择栈 3 作为接收栈，容器 3 的下层会有一个优先级更高的容器 2。因此，容器 3 必须至少再重定位一次。而如果选择栈 4 作为接收栈，就没有这种担忧。因此，在这种情形下重定位次数的额外增加被避免。这一检查步骤可以适用于目标容器（容器 1）上层的所有容器（图中容器 3 和容器 11），将属于最坏情况（没有任何一个其它栈可以在不引入额外重定位次数的条件下接收该重定位容器）的容器数量添加到 LB1 中，这样就得到了 LB2。

	8		
	12	6	
	9	2	
4	10	7	5
栈1	栈2	栈3	栈4

图 2.6 容器布局的 LB3 所考虑的中间状态

进一步地，LowerBound3 (LB3) 扩展了 LB2 所提出的检查步骤到全部容器上。依然考虑图 2.5 中给出的初始容器布局。考虑下一个将要堆出的容器 2 所在栈。假设所有优先级小于 2 的容器（在图中的例子中是容器 1）以及它们上方的所有容器（在图中的例子中是容器 3 和容器 11）都已被移除，这产生了一种新的容器数较小的布局，如图 2.6 所示。然后我们可以使用与 LB2 相同的检查步骤来分析在容器 2 上层的容器 6。由于重定位容器 6 的所有可能接收栈，都会有一个优先级小于 6 的容器。因此，容器 6 不得不再至少重定位一次。继续执行这一检查过程直到所有容器被堆出，最终得到 LB3。

从 LB1 到 LB3，它们基于一定的数学规则逐步更加逼近实际容器布局所需的最少重定位次数。但毫无疑问，重定位次数最优解是描述容器布局调度难易的最佳标签。我们的初步实验表明，使用 LB2 和 LB3 作为标签的特征提取器识别性能较差，因此本研究着重考察 LB1 标签与重定位次数最优解标签。

2.2 评价指标

2.2.1 回归模型经典评估方法

论文采用回归方法实现容器布局的标签识别。为了更好地评估回归模型的性能,对回归模型使用了两种参考指标:均方误差(MSE)和平均绝对误差(MAE)。其中,在模型训练时使用 MSE 作为损失函数。

MSE、MAE 是衡量回归模型性能的常用指标。MSE 表示预测值与真实值之间平方差的平均值,MAE 表示预测值与真实值之间绝对差的平均值。与 MSE 相比,MAE 更不容易受异常值影响。他们的公式如下:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.1)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.2)$$

2.2.2 针对容器调度的回归模型实用评估方法

准确率 (Accuracy) 这一评价指标一般被用于分类模型。论文根据容器调度过程的实际需求,设计了两种准确率指标,分别是 AccuracyTOP1 与 AccuracyTOP2。这两种准确率指标能够很好地反映标签预测值与真实值的接近程度,他们的计算方式如下:

$$\text{Accuracy}_{\text{TOP1}} = \frac{\text{满足}|\text{预测值} - \text{真实值}| \text{的样例数} < 0.5}{\text{总样例数}} \quad (2.3)$$

$$\text{Accuracy}_{\text{TOP2}} = \frac{\text{满足}|\text{预测值} - \text{真实值}| \text{的样例数} < 1}{\text{总样例数}} \quad (2.4)$$

如公式 (2.3) 所示, AccuracyTOP1 是标签预测值与真实值差距小于 0.5 的数据样例占比,满足该要求的数据样例如同在数轴上分布在以其标签为中心的独属于其标签的区间内,因而能够与不同标签的其他实例彻底分隔。在这种情况下,标签相邻(即标签差值为 1)的数据样例的预测值与真实值恒满足具有相同的大小关系。因此该指标可以很好地印证将于第四章介绍的容器调度算法的性能, AccuracyTOP1 越接近 1,提出的容器调度算法就越接近最优解算法。

如公式 (2.4) 所示,仅改变差距为 1,可以在对归属于不同标签数据样例的分隔程度要求不高的情况下成为稍弱的评估指标。若假定 AccuracyTOP2 趋于 1

且样例均匀分布在以标签为中心长度为 2 的区间内，不难计算此时标签相邻的数据样例的预测值满足具有与标签相同大小关系这一条件的占比是 75%，也就是说模型对相邻标签数据样例具有 75%的区分度。

2.3 论文的实验设置

论文的实验基于 Python 3.10 和 Pytorch2.1.0 开展，在配备 Ubuntu22.04 操作系统、Intel Xeon Silver 4214R 处理器和 NVIDIA GeForce RTX™ 3080 Ti 12GB 的高性能计算平台上运行。

论文按照容器布局规模、标签类型设置四个实验组别，如表 3.1 所示。后文将不再说明地使用实验组别序号。

表 2.1 实验组别设置

组别	规模	标签
A	小规模容器布局（10-4）	重定位次数最优解
B	大规模容器布局（7-6）	重定位次数最优解
C	小规模容器布局（10-4）	LB1
D	大规模容器布局（7-6）	LB1

论文的数据集是基于程序随机地生成，并在生成时剔除不合法数据（无法调度的容器布局）。其中，获得重定位次数最优解标签时使用了 Shunji Tanaka^[25]提出的精确解算法。考虑到容器调度算法的实际需求，论文不考虑容器数为 1、2 以及 N 的容器布局数据。第三章中，以总容器数为类别每类生成数目为 10240 的容器布局，最终得到了总数为 778240 的数据集，其中小规模容器布局（10-4）与大规模容器布局（7-6）数据集数目分别为 378880、399360。将所产生的数据集每类别以 4: 1 的比例随机地划分为训练集与测试集。第四章的数据集产生规则将在当章介绍。

根据初步实验的结果，论文采用以下机器学习组件与参数：优化器选取 Adam，学习率设置为 0.002，损失函数选取 MSE，批大小（batch_size）设置为 64（除 4.2.2 章节，批大小的设置将在当节介绍），轮次（epoch）设置为 200（第四章中使用大规模数据集时将适当减少），激活函数选取 ReLU。

3 容器布局特征提取器研究

3.1 容器布局的标准化方法与有效性实验

在大部分机器学习问题上,对原始数据进行标准化更有利于学习到重要的特征。本章节将提出一种适用于容器布局的标准化方法,并通过设计实验验证提出的标准化方法的有效性。

3.1.1 容器布局标准化方法

由于容器布局在 CRP 所具有的栈顺序无关性,即任意对各栈做出交换不影响重定位次数的结果,标准化操作旨在按照一定的规则对容器布局各栈做出排序,以便本章后面将要介绍的几种容器布局特征提取器提取到更重要的特征。本小节将介绍五种用于标准化的排序规则,同时这一思想也直接用于 3.2 节作为人工提取的栈内局部信息。

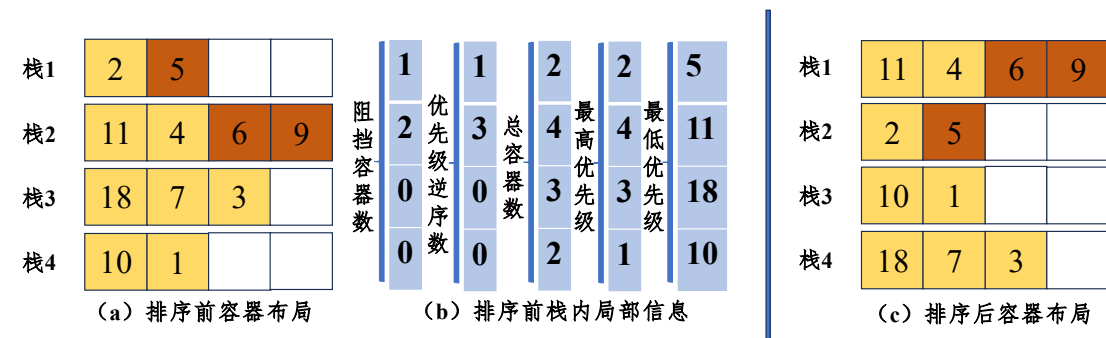


图 3.1 容器布局的标准化

如图 3.1 (b) 所示,论文设计了栈内阻挡容器数、栈内优先级逆序数、栈内总容器数、栈内最高优先级与栈内最低优先级五种排序标准。栈内阻挡容器数是前文介绍的 LB1 在单个栈内的体现。栈内优先级逆序数进一步增强了顺序表征能力。视栈内所有容器的优先级编号为一个排列,当没有阻挡容器时优先级逆序数显然为 0,当与这一再好不过的排序完全相反时优先级逆序数为 $n \times (n-1)/2$ (n 为该栈总容器数)。余下的三种局部信息是简单明了的。表 3.2 详细归纳了这五种局部信息的特点。

表 3.2 人工提取的栈内局部信息特点

栈内局部信息类型	特点
阻挡容器数	相对特征,全局的阻挡容器数是衡量所需重定位次数的简单指标,其值越大,该栈容器越阻塞。

表 3.2（续） 人工提取的栈内局部信息特点

栈内局部信息类型	特点
优先级逆序数	相对特征，与阻挡容器数相似，其值越大，该栈容器优先级越呈现逆序，相比阻挡容器数具有更强的顺序表征能力。
总容器数	相对特征，进一步补充有效槽位信息。
最高/低优先级	绝对特征，描述绝对的优先级范围信息。

基于这些很容器经计算得到的局部信息，论文执行以下容器布局标准化规则：从主要到次要依次将栈内阻挡容器数、栈内优先级逆序数、栈内最高优先级、栈内最低优先级和栈内总容器数作为关键字分别执行栈的逆序、逆序、正序、逆序、正序排序。图 3.1（c）就是从（a）所显示的容器布局根据这一规则标准化后所得到的布局。

3.1.2 容器布局标准化的有效性实验

论文以组别 A 为例设置实验，验证了所提出的容器布局标准化方法在将于本章后面介绍的基于卷积神经网络的特征提取器和基于长短期记忆递归神经网络的特征提取器识别性能上的有效性。图 3.2 展示了 CNN 与 LSTM 对有序数据集（图中红粉色）和无序数据集（图中紫蓝色）标签识别的性能表现。使用标准化策略明显起到了正向的提升效果，损失函数更低。它们具体的网络设计将在后几节中介绍。

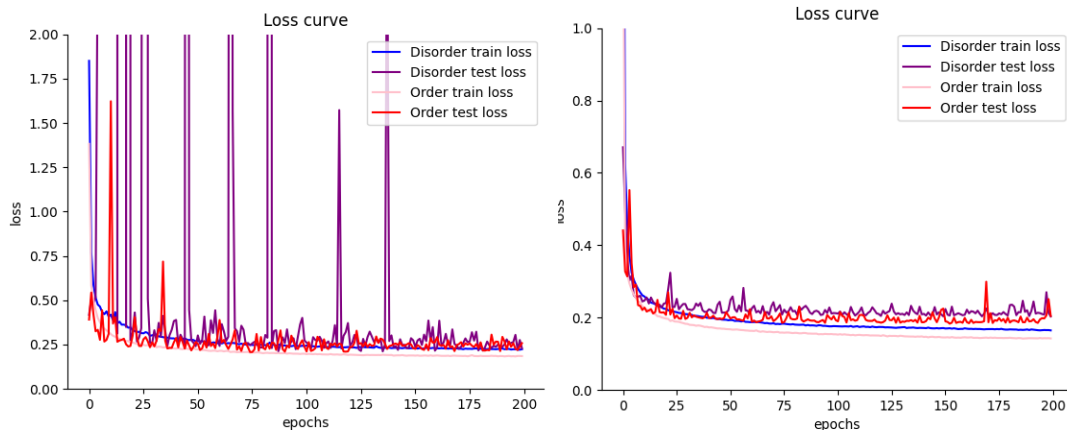


图 3.2 准化方法的有效性实验损失函数曲线（组别 A）

左图：基于 CNN 的特征提取器；右图：基于 LSTM 的特征提取器

3.2 基于栈内局部信息的人工特征提取器

上一节所介绍的五种标准化排序规则建立于比较全面的栈内局部信息之上，充分考虑到了相对特征（栈内阻挡容器数、总容器数与优先级逆序数）与绝对特

征（最高优先级与最低优先级），这些局部信息可以直接作为整个容器布局的抽象表达。在这一节，将把这些人工提取得到的局部信息展开为一维特征，进而借助全连接网络实现标签识别。实验表明，相比直接采用容器布局的原始优先级编号为特征表示的方法，基于我们的规则提取得到的特征对于网络的识别性能具有较大提升。

3.2.1 原理与识别网络设计

全连接网络（FCN）是最简单的人工神经网络，也常常作为其他人工神经网络的重要末端结构。在本研究中，FCN 既作为基于栈内局部信息的人工特征提取器的网络末端结构，也作为将于后几节介绍的特征提取器以及第五章基于特征融合策略的网络末端结构。

在 FCN 的网络结构中，每个神经元与前一层的所有神经元相连，并利用非线性激活函数计算输出——本研究始终采用修正线性单元（ReLU）作为激活函数。每个神经元的输出作为下一层神经元的输入，直到输出层输出网络的预测结果。FCN 由输入层、一个或多个隐藏层和输出层组成。FCN 可以训练的参数是各个神经元的权重和偏置。

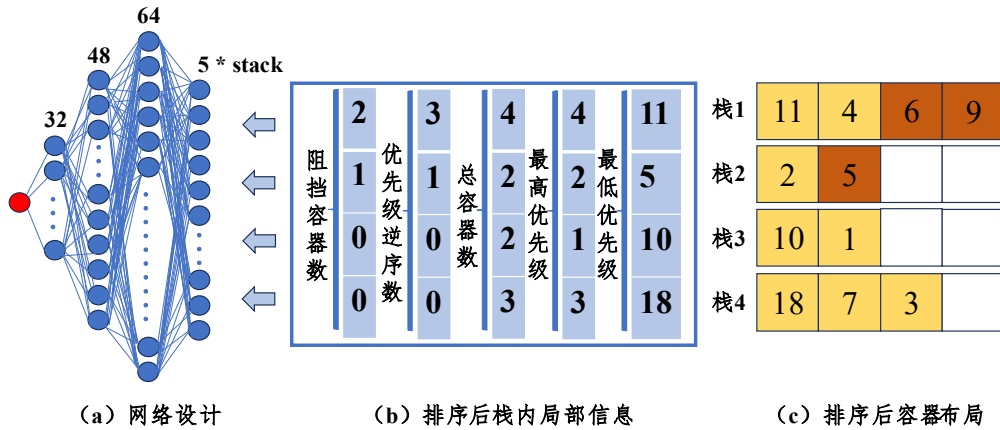


图 3.3 基于栈内局部信息的人工特征提取器的识别网络设计

图 3.3 (c) 显示的排序后的容器布局经人工提取得到的特征矩阵如 (b) 所示，包含每个栈内的阻挡容器数、总容器数、优先级逆序数、最高优先级与最低优先级的数值。如 (a) 所示，特征矩阵展开得到的一维向量，输入 FCN 的输入层。输入层的神经元数目等于局部信息数（即 5）× 栈数（S），在图中的例子中为 $5 \times 4 = 20$ 。其后分别经过神经元数目为 64、48、32 的隐藏层，最后到达神经元数目为 1 的输出层，即获得根据输入的局部信息所预期的重定位次数标签。后

文将要介绍的所有网络设计的末端四层结构与（a）一致。

3.2.2 实验情况——通过与基于原始数据的全连接网络对比

由于人工提取的局部信息中已经隐含了 LB1 的局部成分，因此基于这种方法识别 LB1 标签是无意义的，在这种情况下网络只需学习到一个简单的加法。我们的实验结果也证明了这一点，识别准确率以极快的速度收敛于 1。因此，本小节仅设置 A、B 两个实验组别。为了验证人工提取局部信息这一策略的有效性，本小节对比了基于所介绍的特征提取器的识别网络与直接基于原始数据的全连接识别网络。其中，后者的网络设计仅在输入层神经元个数上存在差别，即等于 N。同特征矩阵一致，将原始容器布局的优先级编号矩阵逐栈展开作为 FCN 的输入。

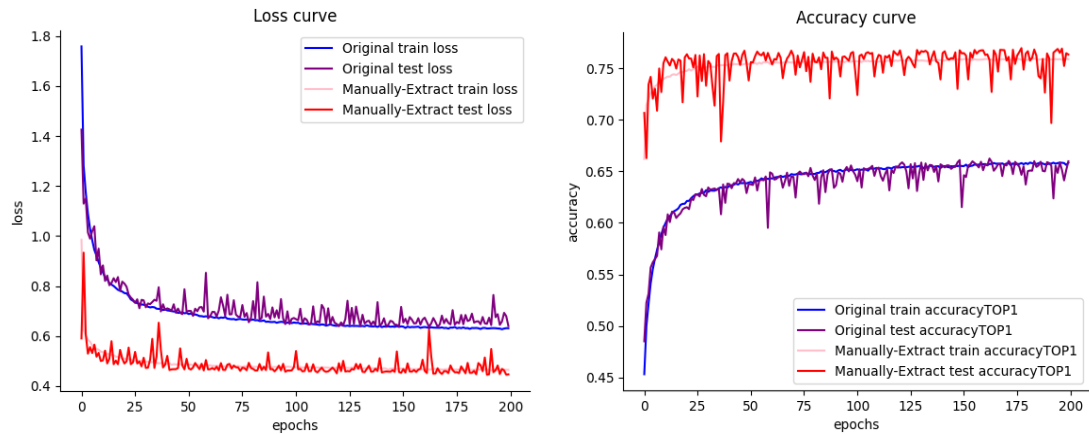


图 3.4 基于栈内局部信息的人工特征提取器的识别网络实验情况（组别 A）

左图：损失函数曲线；右图：准确率曲线

在小规模容器的布局重定位次数最优解标签识别问题（图 3.4）上，基于人工提取特征策略的全连接识别网络（图中红粉色）在训练集与测试集均明显表现出相比没有该策略下的全连接识别网络（图中紫蓝色）更好的性能，识别正确率约增长了 10%。

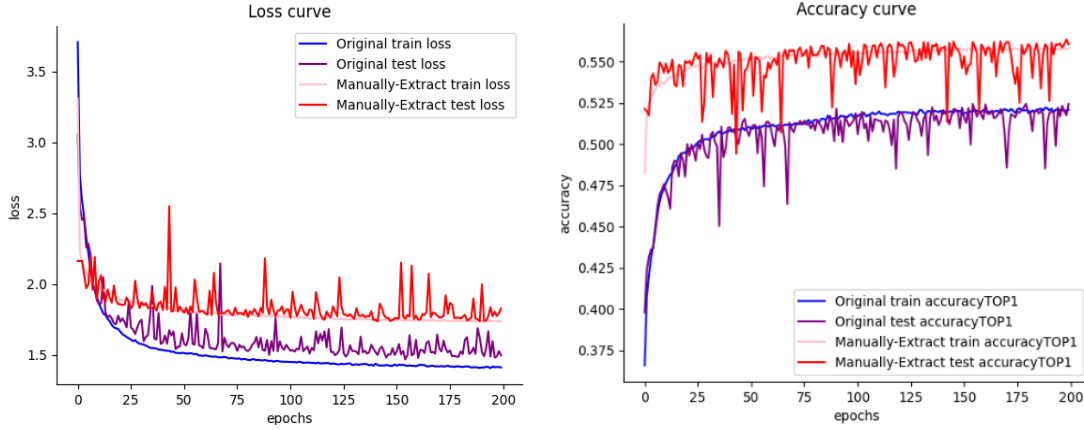


图 3.5 基于栈内局部信息的人工特征提取器的识别网络实验情况（组别 B）

左图：损失函数曲线；右图：准确率曲线

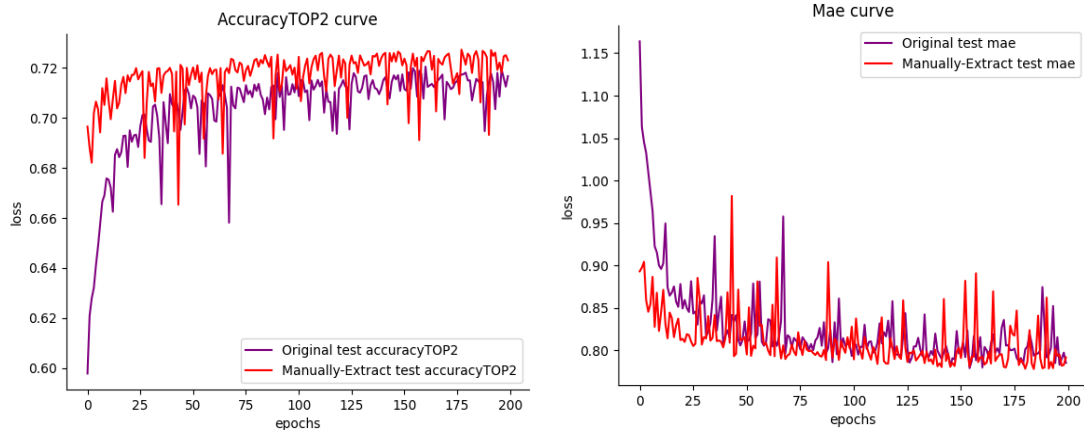


图 3.6 组别 B 的 AccuracyTOP2 曲线与 Mae 曲线

在大规模容器的布局重定位次数最优解标签识别问题（图 3.5）上，提出的策略在 MSE 指标上的表现反而有所降低。但是根据 AccuracyTOP1（图 3.5）、MAE 与 AccuracyTOP2 指标（图 3.6），性能表现则有所提升。

实验表明，人工提取特征的策略的确能够提升容器布局标签识别的性能表现。然而，由于其实质是用少量的局部信息代替原始数据信息，因此随着容器布局规模的增大，单纯依靠该策略的增益将逐渐降低乃至具有负面作用。

3.3 基于卷积神经网络的特征提取器

卷积神经网络（CNN）是目前深度学习领域最为主流的人工神经网络类型。由于具有的平移不变性和卷积核的共享权重架构等性质，CNN 常用于视觉图像领域。由于容器布局也具有栈尺度上的平移不变性以及更看重容器优先级间的相对关系，本节将 CNN 应用于容器布局数据领域加以研究。

3.3.1 原理与识别网络设计

卷积神经网络是因使用一定大小的移动窗口对数据进行逐元素乘加操作而得名。CNN 一般由卷积层、池化层和全连接层组成。对于容器布局数据而言，由于各个容器的优先级编号同等地重要，因此主要作用是以相邻值域取代的方式大幅缩减数据处理规模的池化层是不必要的组件。如图 3.7 所示，论文设计的 CNN 由输入层、若干卷积层、若干全连接层和输入层组成。

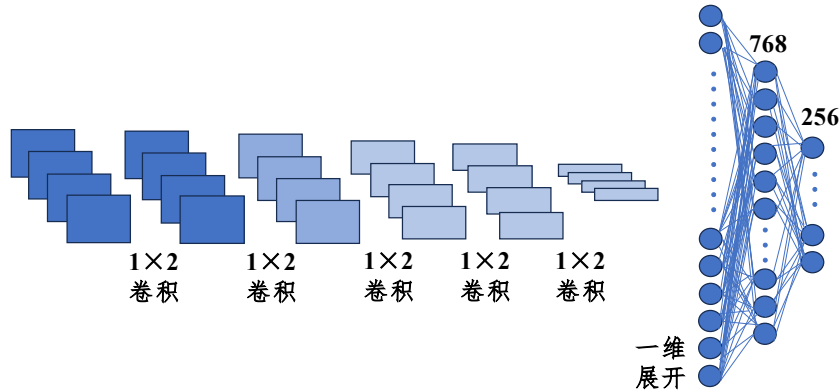


图 3.7 卷积神经识别网络设计

卷积层是 CNN 的核心层次。由于容器布局的栈内层间梯度信息更为重要，论文采用了 1（栈） \times 2（层）大小的卷积核，并将在后一小节通过与另一类型卷积核（2 \times 3）的对比验证这一结论。为了能更好提取到容器布局的边缘特征，在第一个卷积层中设计了填充（padding）操作。对于基于 1 \times 2 卷积核的 CNN，为小规模容器布局与大规模容器布局分别设计了 5、7 层卷积，以达到层层深入卷积、获得更详细梯度信息的目的，各层卷积核个数分别由 8、4 起以 2 为倍数等比增至 128、256。对于基于 2 \times 3 卷积核的 CNN，为小规模容器布局与大规模容器布局分别设计了 3、4 层卷积，各层卷积核个数分别由 32、16 起以 2 为倍数等比增至 128、256。将最后一层卷积层的输出一维展开作为全连接层的输入，经数个全连接层特征个数减为 1，即为所预期的重定位次数标签。

3.3.2 实验情况——通过与基于两类卷积核的 CNN 对比

实验设置 A、B、C、D 四个组别，图 3.8~3.11 分别展示了各个组别的实验结果。结果显示，在以 LB1 作为标签的实验组别 C、D 中，基于两类卷积核的 CNN 性能表现没有明显差别。但在以重定位次数最优解作为标签的实验组别 A、B 中，能够明显观察到基于 1 \times 2 卷积核的 CNN 具有更好的性能。

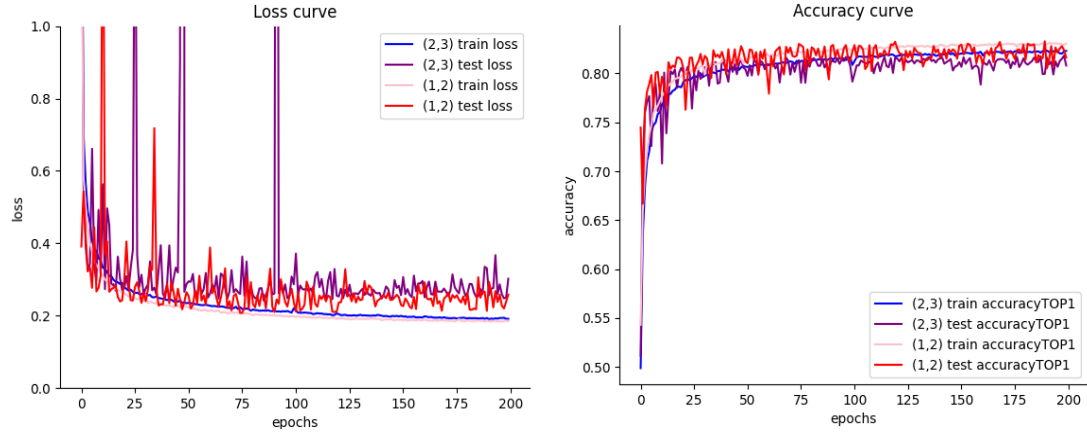


图 3.8 基于卷积神经网络的特征提取器的识别网络实验情况（组别 A）

左图：损失函数曲线；右图：准确率曲线

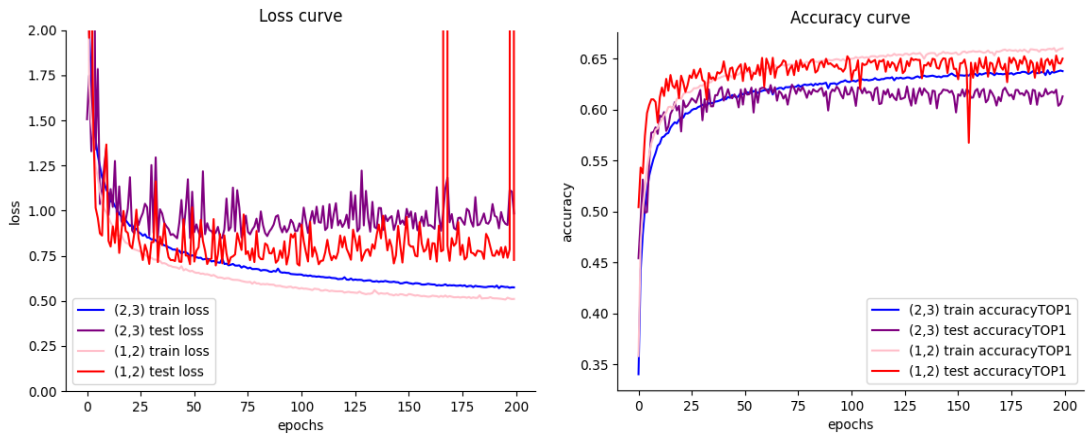


图 3.9 基于卷积神经网络的特征提取器的识别网络实验情况（组别 B）

左图：损失函数曲线；右图：准确率曲线

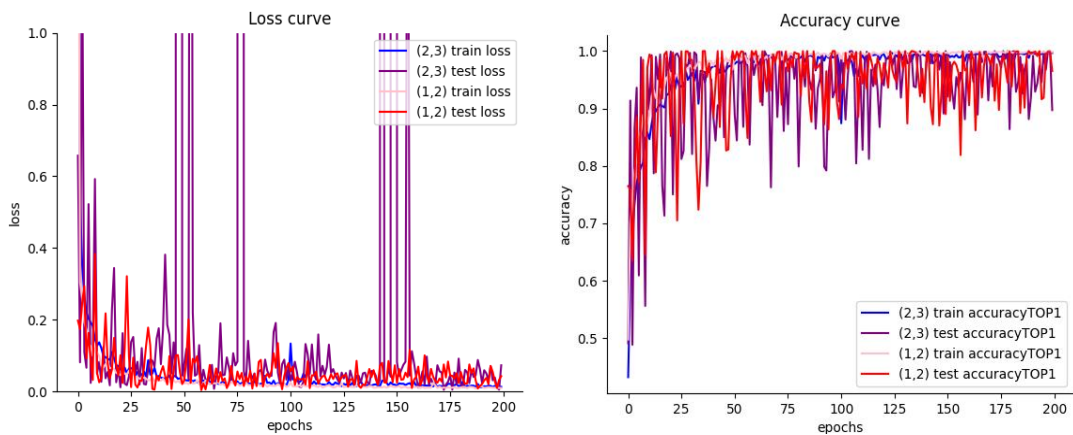


图 3.10 基于卷积神经网络的特征提取器的识别网络实验情况（组别 C）

左图：损失函数曲线；右图：准确率曲线

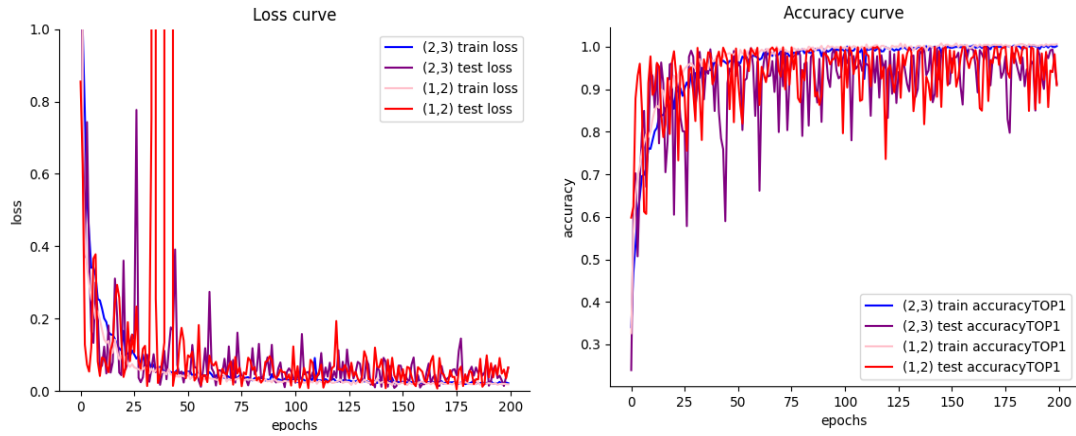


图 3.11 基于卷积神经网络的特征提取器的识别网络实验情况（组别 D）

左图：损失函数曲线；右图：准确率曲线

3.4 基于长短期记忆递归神经网络的特征提取器

长短期记忆神经网络（LSTM）是由多个 LSTM 单元组成的网络，因其独特的网络结构通常被用于处理时间序列数据。已有研究将 LSTM 运用到容器布局的标签识别上，论文的实验也显示了 LSTM 在处理小规模容器布局问题和 LB1 识别问题上的突出表现。

3.4.1 原理与识别网络设计

输入序列和隐藏状态是 LSTM 单元的两个最重要的组成部分。输入序列是每个单元的输入，隐藏状态是根据当前输入和历史信息计算的。随着连接的 LSTM 单元数量增加，隐藏状态对历史信息有越来越多的了解。已有的研究证明了使用从栈到栈（stack-to-stack）输入方式的优越性，本研究也沿用这一输入方式。

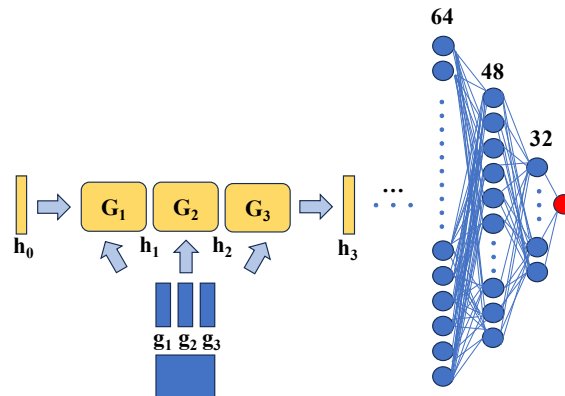


图 3.12 长短期记忆递归神经识别网络设计

如图 3.12 所示,所构建的网络由数个 LSTM 单元和末端数个全连接层组成。将第 i 个 LSTM 单元的隐藏状态记为 h_i , 将第一个 LSTM 单元的隐藏状态 h_0 初始化为零向量。从容器布局优先级编号矩阵中逐栈选取的列向量 $g_i (i=1, 2, \dots, S)$ 作为 LSTM 单元 G_i 的输入序列。当最后一个 LSTM 单元 G_S 输出隐藏状态 h_S 后,再将其同全连接层连接。第一层全连接层的神经元数量由 h_S 的维度确定,输出层只有一个神经元,即为所预期的重定位次数标签。

3.4.2 实验情况——通过本章介绍的三种特征提取器对比

本小节将系统对比本章所介绍的基于栈内局部信息的人工特征提取器、基于 CNN 的特征提取器的实验数据与基于 LSTM 的容器布局特征提取器的性能表现。实验设 A、B、C、D 四组别,其中如 3.2.2 所介绍的,C、D 组别不考虑使用基于栈内局部信息的人工特征提取器。

图 3.13 与图 3.14 展示了三种特征提取器用于容器重定位次数最优解标签识别的性能表现。实验结果显示,基于栈内局部信息的人工特征提取器的识别性能最差。基于 LSTM 的特征提取器在小规模容器布局数据上表现最好,然而在大规模容器布局数据上被基于 CNN 的特征提取器反超。不过,图 3.14 显示了后者存在的剧烈波动现象,而基于 LSTM 的特征提取器则更加平稳。需要补充的是,在上文 CNN 章节的实验这一波动现象也频频发生,幸运的是其性能表现最后总能回到正常水平。

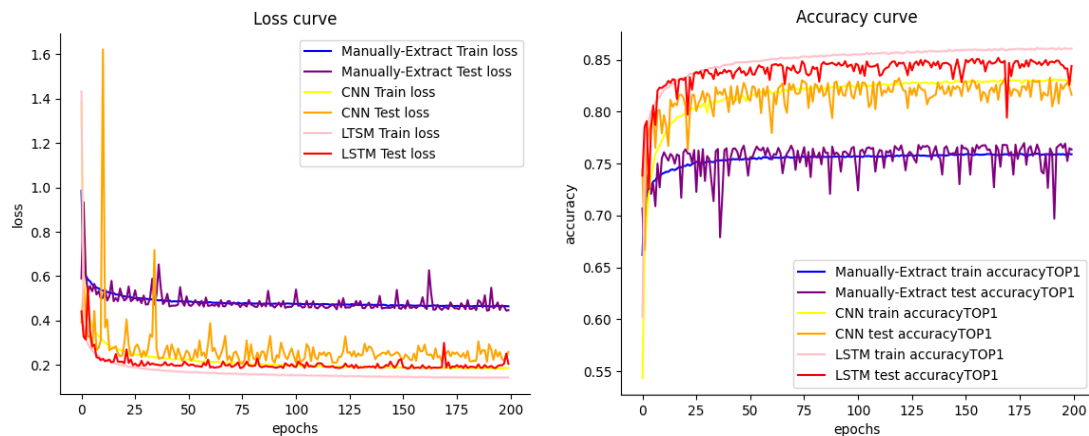


图 3.13 本章介绍的三种特征提取器性能对比 (组别 A)

左图: 损失函数曲线; 右图: 准确率曲线

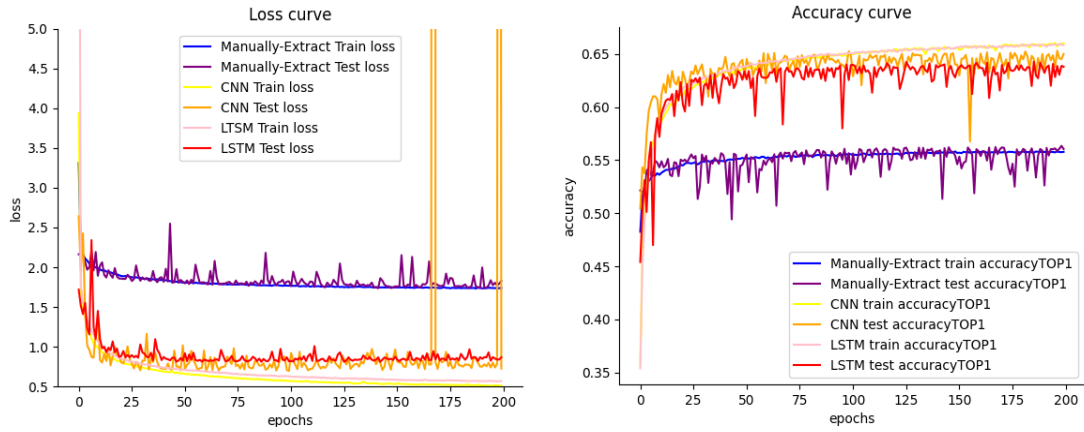


图 3.14 本章介绍的三种特征提取器性能对比（组别 B）

左图：损失函数曲线；右图：准确率曲线

图 3.15 与图 3.16 展示了本章介绍的两种特征提取器对 LB1 标签的识别性能。明显观察到，无论应对何种容器布局规模，基于 LSTM 的特征提取器的识别性能远远超越了基于 CNN 的特征提取器，损失函数收敛于 0，准确率收敛于 1。在提取有关 LB1 的特征方面，基于 LSTM 的特征提取器达到最优效果，已不再需做任何改进。

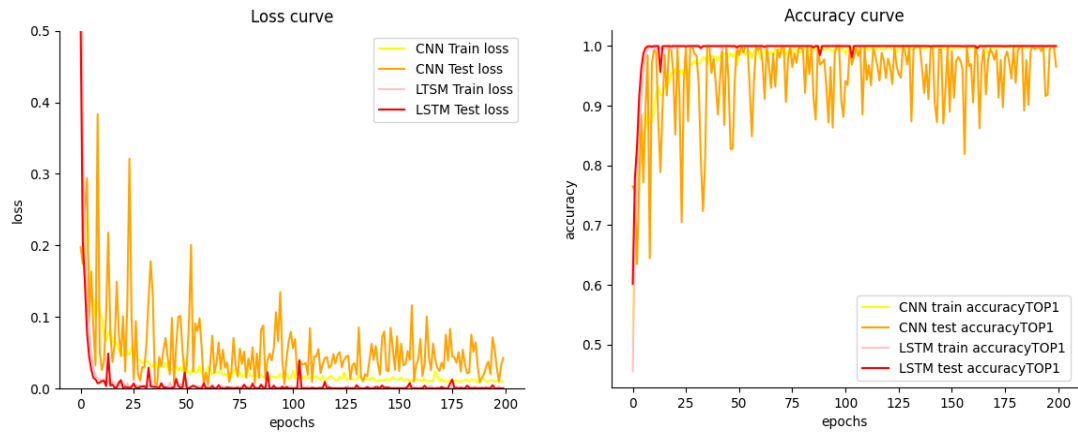


图 3.15 本章介绍的两种特征提取器性能对比（组别 C）

左图：损失函数曲线；右图：准确率曲线

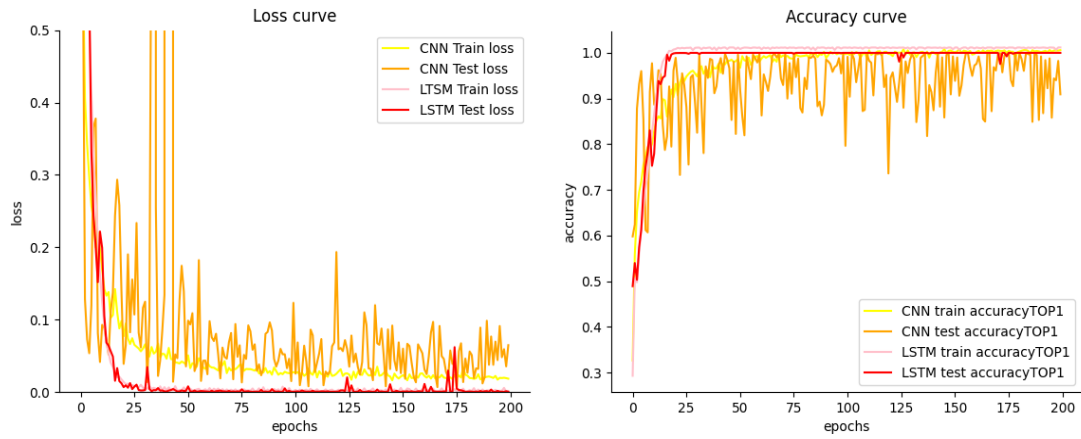


图 3.16 本章介绍的两种特征提取器性能对比（组别 D）

左图：损失函数曲线；右图：准确率曲线

4 基于特征融合的容器调度优化方法

4.1 利用多个特征提取器实现特征融合

通过第三章的分析，对于 LB1 标签的识别算法而言，使用 LSTM 是最优的选择。然而，对于以容器重定位次数最优解为标签的识别算法，仍需要进行更多研究以提高其性能。我们发现三种特征提取器各有其关注的特征尺度，得以在不同类型问题上具有不同的表现。基于这种观察，本章提出了利用多个特征提取器实现特征融合的策略，以改进以容器重定位次数最优解为标签的识别算法，以实现“1+1>2”的目标。

4.1.1 模型架构

图 4.1 具体展示了论文提出的识别模型的核心思想，即以特征拼接的方式实现特征融合。在第三章分析各种特征提取器时，均强调了它们末端结构的共同性——全连接网络。我们很容易想到使用特征拼接这一简单的方式，即可实现不同特征提取器的特征融合。融合后的特征输入神经元个数依次是 96、48、32、1 的四层全连接网络结构。而在各个特征提取部分的网络结构与第三章所介绍的完全一致。

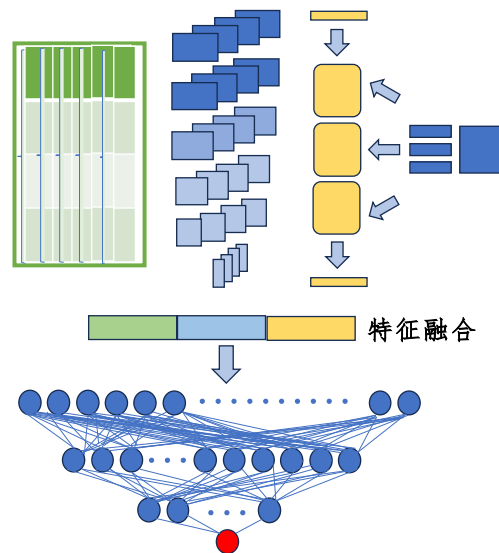


图 4.1 基于特征融合的认可网络设计

4.1.2 特征融合策略的有效性实验

为了验证所提出特征融合策略的有效性，本实验沿用第三章所使用的数据集，仅增加了本章所提出的识别网络模型这一实验组，并与单独表现较好的基于

CNN 和 LSTM 的识别网络对比。实验设置 A、B 两组别。

在小型规模容器布局数据（图 4.2）上，提出的基于特征融合的认可网络超越了基于 CNN 的认可网络性能表现，但仍落后于基于 LSTM 的认可网络。这证明了 LSTM 在小规模容器布局数据方面的突出表现。

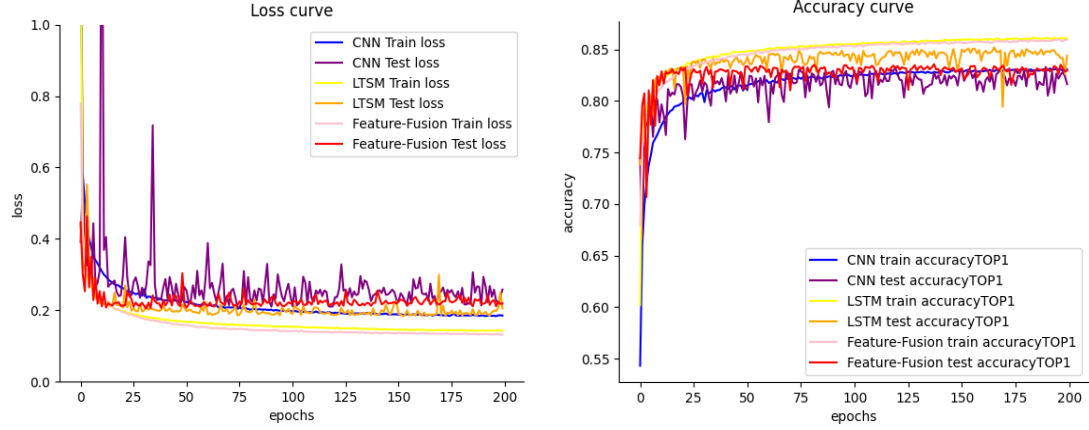


图 4.2 基于特征融合的认可网络性能对比（组别 A）

左图：损失函数曲线；右图：准确率曲线

在更贴合实际应用的大规模容器布局数据（图 4.3）上，论文提出的基于特征融合的认可网络超越了所有单独进行特征提取的认可网络，且相比原来性能最佳的基于 CNN 的认可网络训练过程更加稳定。这证明了论文所提出的特征融合策略的有效性。

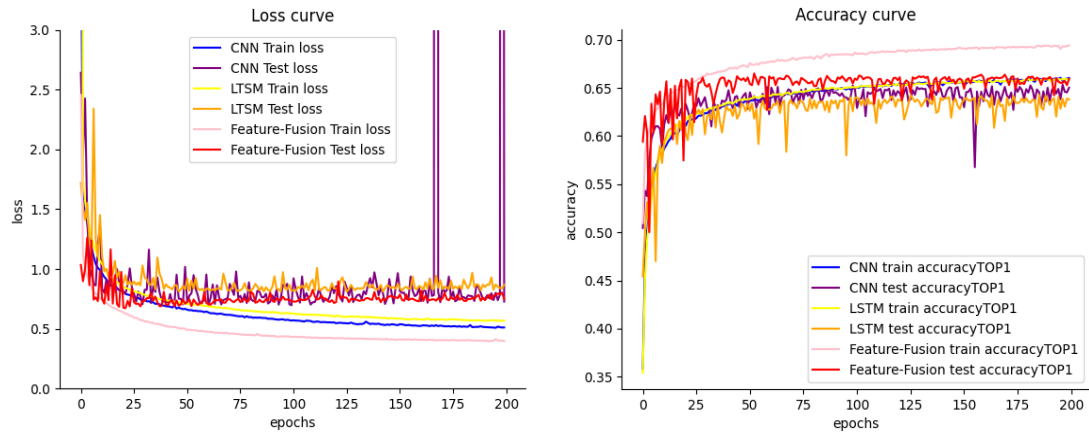


图 4.3 基于特征融合的认可网络性能对比（组别 B）

左图：损失函数曲线；右图：准确率曲线

4.1.3 训练集规模的边际效应实验

不同于传统的图像、语音等需要从现场捕获物理信号产生的数据类型，容器

布局的数据集完全由程序生成。理论上，我们可以生成所有海量级的容器布局数据。越大的数据集，对应着越高的识别准确率。然而，我们的实验（图 4.4~4.5）显示，随着数据集大小的指数级增长，识别准确率提高的边际效应愈发缩小。识别准确率最终收敛于一定值，这可能由模型本身的特征提取性能所决定。

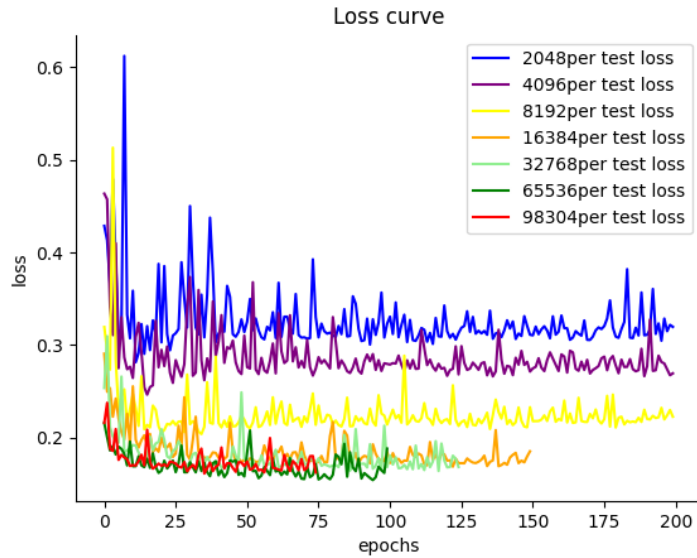


图 4.4 数据集大小边际效应实验的损失函数曲线（组别 A）

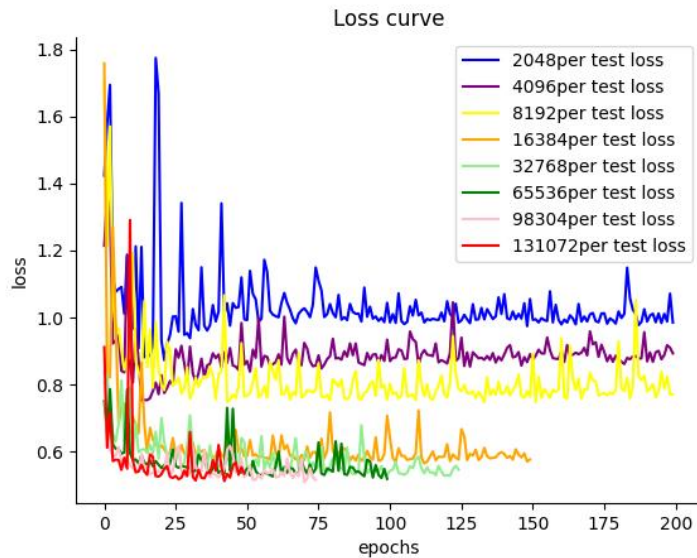


图 4.5 数据集大小边际效应实验的损失函数曲线（组别 B）

本实验以总容器数为类别每类生成数目分别为 2048、4096、8192、16384、32768、65536 与 98304 的容器布局训练集，在组别 B 额外增加了每类数目为 131072 的容器布局训练集。为了统一比较，所有实验组的测试集采用相同的每

类生成数目为 6144 的容器布局数据集。识别网络采用本章提出的基于特征融合的识别网络。在较大规模训练集上适当减少了训练轮次。

4.1.4 训练集类别均衡实验

上文部署的所有实验均基于以总容器数为类别均匀生成的数据集产生规则。基于特征提取器对容器数较少的布局更加容易学习到标签所对应特征的观察, 本小节考虑增大容器数较多的布局数据在训练集中的占比。现在给出下面的定义: 容器数小于或等于 2 倍栈数的容器布局称为简单样本, 容器数大于 2 倍栈数的容器布局属于困难样本。实验结果 (图 4.6) 证明了对数据集采取一定的类别均衡措施的有效性。

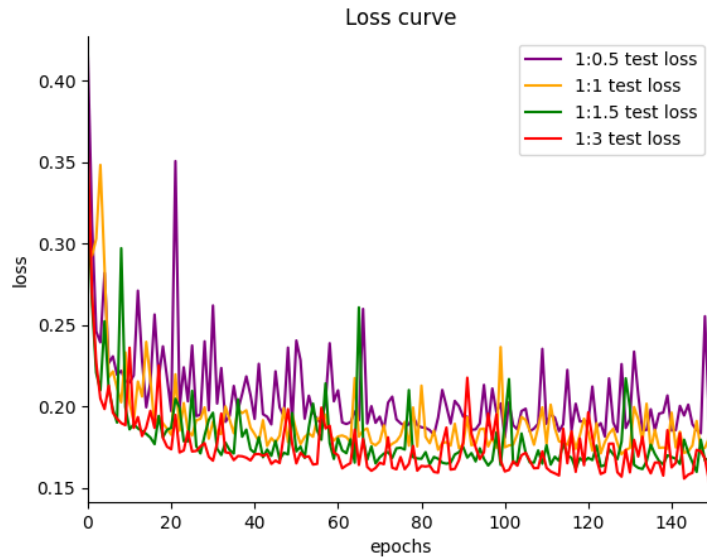


图 4.6 训练集类别均衡实验的损失函数曲线 (组别 B)

本实验以组别 B 为例, 设置简单样本与困难样本每类数目比例分别 (近似) 为 1: 0.5、1: 1、1: 1.5、1: 3 四组实验。训练集总规模设置为 606208 个样例, 使用本章提出的基于特征融合的识别网络开展实验。测试集均沿用 4.1.3 按照以总容器数为类别每类均匀生成数目为 6144 的数据集。训练轮次设置为 150。

4.2 容器调度实验

我们对容器布局进行标签识别的出发点, 是为了利用机器学习的特征提取能力学习到更“好”的容器布局。对于容器调度问题, “好”即对应着所需重定位次数最少, 更“好”的容器布局在调度决策中将被优先考虑。为了完成这一最终任务, 本节将介绍基于论文提出的特征融合识别网络实现容器调度的算法流程与

良好的性能。

4.2.1 容器调度算法

图 4.2 展示了论文提出的容器调度算法流程示例图。每次调度按照下面的步骤考察所要调度的容器布局。

- (1) 若容器 1 位于栈顶，则直接堆出；
- (2) 若容器 1 位于除栈顶以外的槽位，则考察所有可能的重定位方案，逐方案将重定位后的容器布局标准化后输入基于特征融合的重定位次数最优解识别网络，将网络输出作为重定位次数预测值，选择预测值最小的重定位方案；
- (3) 若容器优先级编号最小者大于 1，则需对所有容器优先级编号执行归一化操作；
- (4) 若容器已经全部堆出，调度结束。

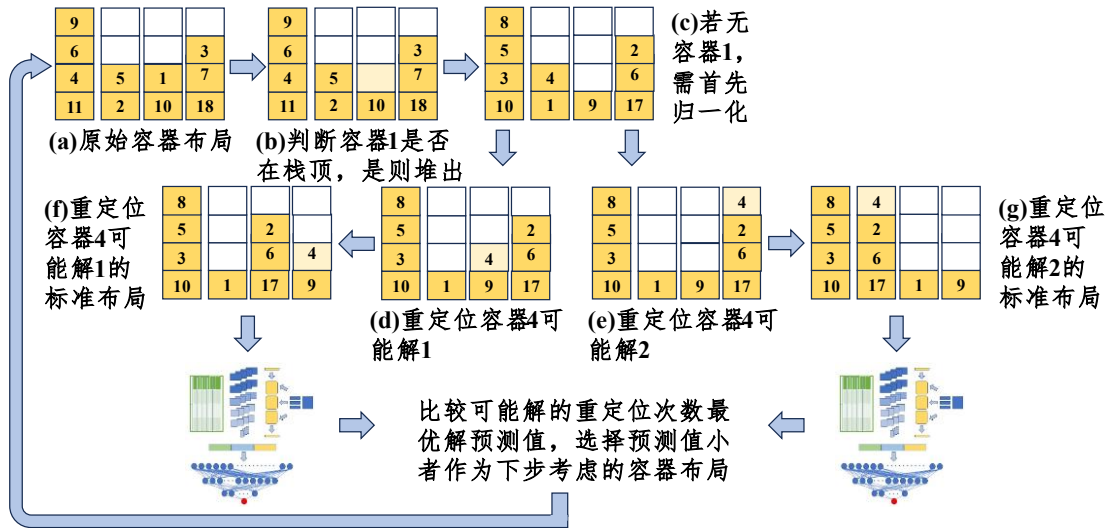


图 4.7 容器调度过程示例图

4.2.2 数据集设置和模型训练

为了获得更好的调度性能，我们基于本章的类别均衡策略针对小规模容器布局（10-4）和大规模容器布局（7-6）分别随机产生了总数为 5194176、5560320 的训练集，简单样本与困难样本每类数目比例分别（近似）为 1: 7、1: 8。

使用本章提出的基于特征融合的容器布局重定位次数最优解识别网络，设置轮次为 200，设置了 6 小组实验用于获得性能最好的批大小参数值——分别是 256、512、1024、2048、4096、8192，组别 A 与组别 B 的训练结果分别如图 4.3 与 4.4 所示。结果显示，批大小为 2048 时模型的识别性能达到最好，小规模容

器布局（10-4）的测试集 MSE 收敛于 0.10 上下，MAE 收敛于 0.15 上下；大容器布局（7-6）的测试集 MSE 收敛于 0.36 上下，MAE 收敛于 0.34 上下。因此，确定批大小参数为 2048，将基于该参数所训练的模型（其余未提及的参数与机器学习组件沿用第三章的实验设置）作为特征融合模型用于下节将介绍的容器调度算法。

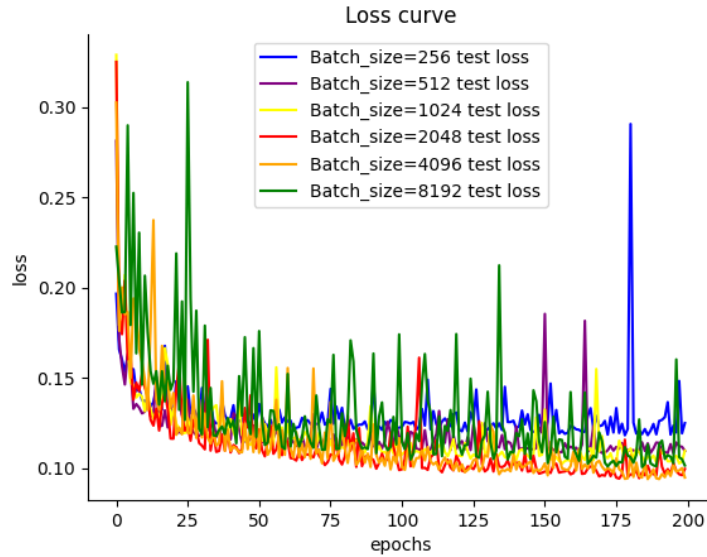


图 4.8 用于容器调度的特征融合模型测试集损失函数曲线（组别 A）

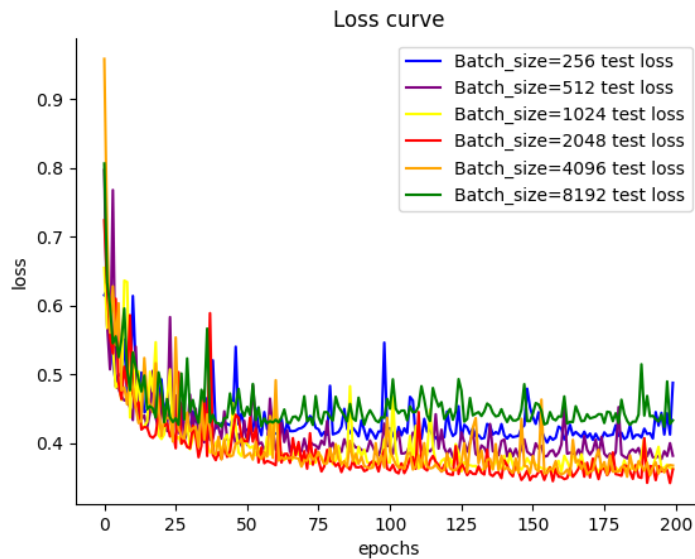


图 4.9 用于容器调度的特征融合模型测试集损失函数曲线（组别 B）

4.2.3 容器调度结果分析

将上一小节训练获得的特征融合模型用于我们所介绍的容器调度算法,通过与 Shunji Tanaka 提出的精确解算法^[25]进行比较,在 ZQLZ12 数据集^[15]10-4-40、7-6-42 (S-T-N) 上测试论文提出的算法性能。如表 4.1 所示,在小规模容器布局 (10-4) 和大规模容器布局 (7-6) 上分别获得了与精确解算法仅 0.688%、3.123% 的差距,超越了论文中基于机器学习的近似解算法的最好成绩。

表 4.1 容器调度的实验结果

容器布局规模	平均重定位次数		近似解算法与精确解算法的差距	
	本研究的调度算法	精确解算法	本研究	基于 LSTM 和强化学习的研究
10-4-40	21.96	21.81	0.688%	0.897% (10-4-37)
7-6-42	34.67	33.62	3.123%	10.273%

表中近似解算法与精确解算法的平均重定位次数差距的计算公式如下:

$$\text{差距} = \frac{\text{近似解算法的重定位次数} - \text{精确解算法的重定位次数}}{\text{精确解算法的重定位次数}} \quad (4.1)$$

5 总结与展望

论文针对有限制异优先级容器调度问题进行研究,给出了基于特征融合的容器布局重定位次数最优解识别模型和基于此模型的容器调度算法并在前人的数据集上进行了数值实验,本研究的创新点与结论包括:

(1) 系统全面地对比研究了几种特征提取器对与 LB1 与重定位次数最优解有关特征的提取能力;

(2) 提出了基于特征融合的容器布局重定位次数最优解识别模型,在大规模容器布局数据上超越了所有基于单独特征提取器的识别模型,并研究了训练集规模的边际效应与类别均衡的重要性;

(3) 提出基于特征融合模型的容器调度算法,在公开数据集上的平均容器重定位次数超越了论文中基于机器学习的近似解算法的最好成绩。

本研究仍存在一些不足,例如仅考虑了有限制异优先级的 CRP,提出的算法训练计算成本比较高昂,仅研究了两种具有代表性的容器布局规模等。如何进一步提高识别性能、模型的泛化能力和训练成本,可以作为未来研究的改进方向。

参考文献

- [1] Levinson M. The box: how the shipping container made the world smaller and the world economy bigger[M]. Second edition. Princeton Oxford: Princeton University Press, 2016.
- [2] 上海市交通委员会.上海港集装箱吞吐量连续十四年排名全球第一[EB/OL].(2024-01-08)[2024-02-27].
<https://jtw.sh.gov.cn/jtyw/20240108/3e7cd279bb6d415d8c053bf2549b64e7.html> .
- [3] Caserta M, Schwarze S, Voß S. A mathematical formulation and complexity considerations for the blocks relocation problem[J]. European Journal of Operational Research, 2012, 219(1): 96-104.
- [4] Kim K H, Hong G P. A heuristic rule for relocating blocks[J]. Computers & Operations Research, 2006, 33(4): 940-954.
- [5] Ünüyurt T, Aydın C. Improved rehandling strategies for the container retrieval process[J]. Journal of Advanced Transportation, 2012, 46(4): 378-393.
- [6] Expósito-Izquierdo C, Melián-Batista B, Moreno-Vega J M. An exact approach for the Blocks Relocation Problem[J]. Expert Systems with Applications, 2015, 42(17-18): 6408-6422.
- [7] Zehendner E, Feillet D. A branch and price approach for the container relocation problem[J]. International Journal of Production Research, 2014, 52(24): 7159-7176.
- [8] Zehendner E, Caserta M, Feillet D, et al. An improved mathematical formulation for the blocks relocation problem[J]. European Journal of Operational Research, 2015, 245(2): 415-422.
- [9] Petering M E H, Hussein M I. A new mixed integer program and extended look-ahead heuristic algorithm for the block relocation problem[J]. European Journal of Operational Research, 2013, 231(1): 120-130.
- [10] De Melo Da Silva M, Toulouse S, Wolfler Calvo R. A new effective unified model for solving the Pre-marshalling and Block Relocation Problems[J]. European Journal of Operational Research, 2018, 271(1): 40-56.
- [11] Galle V, Barnhart C, Jaillet P. A new binary formulation of the restricted Container Relocation Problem based on a binary encoding of configurations[J]. European Journal of Operational Research, 2018, 267(2): 467-477.
- [12] Bacci T, Mattia S, Ventura P. A branch-and-cut algorithm for the restricted Block Relocation Problem[J]. European Journal of Operational Research, 2020, 287(2): 452-459.
- [13] Lu C, Zeng B, Liu S. A Study on the Block Relocation Problem: Lower Bound Derivations and Strong Formulations[J]. IEEE Transactions on Automation Science and Engineering, 2020, 17(4): 1829-1853.
- [14] Tanaka S, Voß S. An exact approach to the restricted block relocation problem based on a new integer programming formulation[J]. European Journal of Operational Research, 2022, 296(2): 485-503.
- [15] Zhu W, Qin H, Lim A, et al. Iterative Deepening A* Algorithms for the Container Relocation Problem[J]. IEEE Transactions on Automation Science and Engineering, 2012, 9(4): 710-722.
- [16] Expósito-Izquierdo C, Melián-Batista B, Marcos Moreno-Vega J. A domain-specific knowledge-based heuristic for the Blocks Relocation Problem[J]. Advanced Engineering Informatics, 2014, 28(4): 327-343.
- [17] Bacci T, Mattia S, Ventura P. The bounded beam search algorithm for the block relocation problem[J]. Computers & Operations Research, 2019, 103: 252-264.

- [18] Jovanovic R, Tuba M, Voß S. An efficient ant colony optimization algorithm for the blocks relocation problem[J]. *European Journal of Operational Research*, 2019, 274(1): 78-90.
- [19] Gulić M, Maglić L, Krljan T, et al. Solving the Container Relocation Problem by Using a Metaheuristic Genetic Algorithm[J]. *Applied Sciences*, 2022, 12(15): 7397.
- [20] Hottung A, Tanaka S, Tierney K. Deep learning assisted heuristic tree search for the container pre-marshalling problem[J]. *Computers & Operations Research*, 2020, 113: 104781.
- [21] Zhang C, Guan H, Yuan Y, et al. Machine learning-driven algorithms for the container relocation problem[J]. *Transportation Research Part B: Methodological*, 2020, 139: 102-131.
- [22] Ye R, Ye R, Zheng S. Machine Learning Guides the Solution of Blocks Relocation Problem in Container Terminals[J]. *Transportation Research Record: Journal of the Transportation Research Board*, 2023, 2677(3): 721-737.
- [23] Jiang T, Zeng B, Wang Y, et al. A New Heuristic Reinforcement Learning for Container Relocation Problem[J]. *Journal of Physics: Conference Series*, 2021, 1873(1): 012050.
- [24] Wei L, Wei F, Schmitz S, et al. Optimization of container relocation problem via reinforcement learning[J]. *Logistics Journal: Proceeding*, 2021, 2021(17): 620-629.
- [25] Tanaka S, Takii K. A Faster Branch-and-Bound Algorithm for the Block Relocation Problem[J]. *IEEE Transactions on Automation Science and Engineering*, 2016, 13(1): 181-190.

