

秩一专家混合用于多任务学习

杨恩能¹⁾ 唐安科²⁾ 郭贵冰¹⁾ 姜琳颖¹⁾ 孙福辉³⁾
王晓燕³⁾ 沈力⁴⁾

¹⁾(东北大学软件学院 沈阳 110169)

²⁾(武汉大学计算机学院 武汉 430062)

³⁾(最高人民法院信息技术服务中心 北京 100745)

⁴⁾(中山大学网络空间安全学院 广东 深圳 518100)

摘要 多任务学习系统通过促进任务间的知识共享与迁移,能够同时处理来自不同领域的任务。近年来,基于任务算术的多任务学习方法取得了显著进展,研究表明,通过在参数层面直接将多个下游任务上独立微调的专家模型合并到预训练模型中,可以生成具备解决相应下游任务能力的统一模型,从而为多任务学习提供了一种高效且灵活的解决方案。然而,现有的模型合并方法通常面临两个挑战:一是完全静态合并的方法由于难以解决任务间的潜在干扰和参数冲突,导致合并的多任务学习模型性能下降;二是现有动态合并的方法需要额外维护参数数量庞大的专家模块,显著增加了模型成本。为此,本文提出了一种高效的模型合并方案,专门用于基于任务算术的多任务学习,称为秩一专家混合(RankOne-MoE)。具体而言,依据微调模型中不同模块所包含的下游任务知识量,本文对专家模型的绝大部分模块进行基于任务算术的静态合并;而对于与下游任务更相关的模块,我们对其线性层的参数进行奇异值分解,生成一系列秩为一的专家用于构建一个跨任务共享的秩一专家库,并通过路由机制动态组合输入实例所需的秩一专家,提升合并模型的多任务性能和适应性。实验结果表明,当合并八个ViT-B/32模型时,提出的方法相比最先进的静态合并方法在多任务精度上平均提升了5.40%;相比最先进的动态合并方法,总参数量减少了81.45%左右,充分验证了提出方法的有效性和高效性。

关键词 多任务学习;模型融合;知识迁移;机器学习;人工智能

中图法分类号 TP18 **DOI号** 10.11897/SP.J.1016.2025.02317

Mixture of Rank-One Experts for Multi-Task Learning

YANG En-Neng¹⁾ TANG An-Ke²⁾ GUO Gui-Bing¹⁾ JIANG Lin-Ying¹⁾ SUN Fu-Hui³⁾
WANG Xiao-Yan³⁾ SHEN Li⁴⁾

¹⁾(Software College, Northeastern University, Shenyang 110169)

²⁾(School of Computer Science, Wuhan University, Wuhan 430062)

³⁾(Information Technology Service Center of People's Court, Beijing 100745)

⁴⁾(School of Cyber Science and Technology, Sun Yat-sen University, Shenzhen, Guangdong 518100)

Abstract Multi-task learning (MTL) systems handle multiple tasks from different domains by promoting knowledge sharing and transfer. Recent progress in task arithmetic-based MTL has shown that merging independently fine-tuned expert models into a pre-trained model at the parameter level can produce a unified model capable of solving downstream tasks, offering an

收稿日期:2024-11-29;在线发布日期:2025-05-23。本课题得到国家自然科学基金青年基金(62032013)、辽宁省自然科学基金优秀青年基金(2023JH3/10200005)、中央高校基本科研业务费专项资金(N2317002)以及国家留学基金委(CSC)的资助。杨恩能,博士研究生,主要研究领域为机器学习与推荐系统。E-mail: ennengyang@stumail.neu.edu.cn。唐安科,博士研究生,主要研究领域为迁移学习和多任务学习。郭贵冰,博士,教授,主要研究领域为推荐系统和自然语言处理。姜琳颖(通信作者),硕士,副教授,主要研究领域为人工智能和自然语言处理。E-mail: jiangly@swc.neu.edu.cn。孙福辉,博士,主要研究领域为可信数据管理。王晓燕,博士,高级工程师,主要研究领域为数据管理和区块链。沈力(通信作者),博士,副教授,主要研究领域为人工智能、深度学习和强化学习。E-mail: mathshenli@gmail.com。

efficient and flexible solution. However, static merging methods often face task interference and parameter conflicts, degrading performance, while dynamic methods require maintaining many expert modules, significantly increasing costs. To address these challenges, this paper proposes the Mixture of Rank-One Experts (RankOne-MoE), an efficient model merging approach. By statically merging most modules in fine-tuned models using task arithmetic and applying singular value decomposition (SVD) to generate dynamically routed rank-one experts for task-relevant modules, the method constructs a cross-task shared rank-one expert library, enhancing performance and adaptability. Experiments on merging eight ViT-B/32 models show a 5.40% average improvement in MTL accuracy over the state-of-the-art (SOTA) static method and an 81.45% reduction in parameter count compared to the SOTA dynamic method, demonstrating the approach's effectiveness and efficiency.

Keywords multi-task learning; model merging; knowledge transfer; machine learning; artificial intelligence

1 引 言

多任务学习 (Multi-task Learning, MTL) 是一种重要的深度学习范式,它通过使用一个统一的神经网络模型同时优化多个相关任务的目标,从而实现任务间的知识共享与迁移^[1-2]。相比于单任务学习或集成学习^[3],多任务学习能够在仅需存储一份模型参数的情况下,处理多种任务需求,不仅显著降低了存储和维护成本,还提升了资源利用效率和模型的泛化能力。得益于这些优异特性,多任务学习在计算机视觉^[4-7]、自然语言处理^[8-11]、语音识别^[12]、推荐系统^[13-16]等多个领域取得了广泛应用。尽管多任务学习在理论和实践中展示出强大的能力,但其依然面临一些挑战。例如,传统的多任务学习范式通常遵循“收集数据-联合训练”的框架,即需要预先收集所有任务的原始训练数据,并使用这些数据联合优化神经网络模型参数。然而,当任务数量或数据规模过大时,这种范式面临两大问题:首先,数据管理成本迅速攀升,不仅需要大规模的存储和计算资源,还可能引发任务间数据冲突和样本不平衡的问题。其次,任务数据的集中化存储与处理容易引发数据隐私和安全风险,特别是在医疗、金融等敏感领域。

近年来,基于任务算术的多任务学习方法为上述挑战提供了一种新颖且正交的解决思路^[17-21]。这类方法不需要联合训练所有任务,而是通过将多个独立训练的神经网络模型在参数层面进行算术合并,生成一个统一的多任务模型。这一范式具有以下显著优点:

(1)降低存储成本

通过跨任务重用单个任务训练的模型,大幅减少了存储多份模型参数的需求^[17,22,23]。

(2)提高泛化能力

合并多个模型的能力能够实现任务间的知识共享,从而更好地泛化到新任务^[20,24-27]。

(3)支持分散开发

模型合并允许多个贡献者独立构建和微调自己的模型,而后通过参数合并进行整合,从而促进分散化和功能化的模型开发^[26,28-29]。

基于任务算术的模型合并方法近年来逐渐成为多任务学习领域的重要研究方向^[21,30-31]。需要强调的是,最简单的任务算术合并方法^[17,32,33]往往面临显著的性能下降问题。换句话说,合并后统一的多任务模型的性能通常低于传统联合训练的多任务学习模型,同时也明显低于独立训练的单任务模型。这种性能下降主要归因于任务间的潜在干扰以及参数冲突,即不同任务模型的优化目标不一致可能导致合并后的模型参数无法有效地平衡各任务的需求。为了缓解性能下降问题,研究者提出了多种改进的模型合并算法,试图通过更细粒度和更结构化的方法优化模型合并过程^[21,31]。其中,现有方法可以粗略分为两个主要类别:(1)基于重要性加权的合并方法:这类方法关注不同任务对应模型的重要性差异,通过为每个模型或更细粒度的层级、参数级别分配重要性系数来优化合并过程^[20,34,35]。这些加权系数通常由各模型在特定任务上的性能、任务相关性或模型参数的重要性来决定。(2)基于子空间的稀疏合并方法:这类方法则利用神经的过参数化特性,在参数空间的子空间中进行模型合并^[19,36]。通

将稠密的模型投影到一个稀疏子空间中,这些方法能够显著减少任务间参数冲突。

上述方法在缓解任务冲突和提升合并模型性能方面取得了显著进展,但它们都属于静态合并方法。也就是说,这些方法在完成模型合并后,所生成的统一模型对于所有任务和输入样本而言是固定不变的。这种静态特性在任务需求多样化的场景中可能存在局限性,例如无法针对不同任务或输入实例动态调整模型行为^[23]。为克服静态合并方法的不足,最近的研究提出了一种名为 WEMoE^[23]的动态

合并策略用于多任务学习。该方法通过引入任务特定模块的动态路由机制,根据输入实例的特征动态调整模块的合并方式,从而实现更灵活的模型适应性,取得了接近传统多任务学习的性能。然而,WEMoE的动态性带来了显著的额外参数开销,尤其是在任务数量较多或模型深度较大的情况下。例如图1中,当合并八个CLIP-ViT-B/32或CLIP-ViT-L/14模型^[37-38]时,WEMoE的总参数量是其他静态合并方法的五到六倍以上。这一问题限制了其在计算资源有限场景中的实际应用价值。

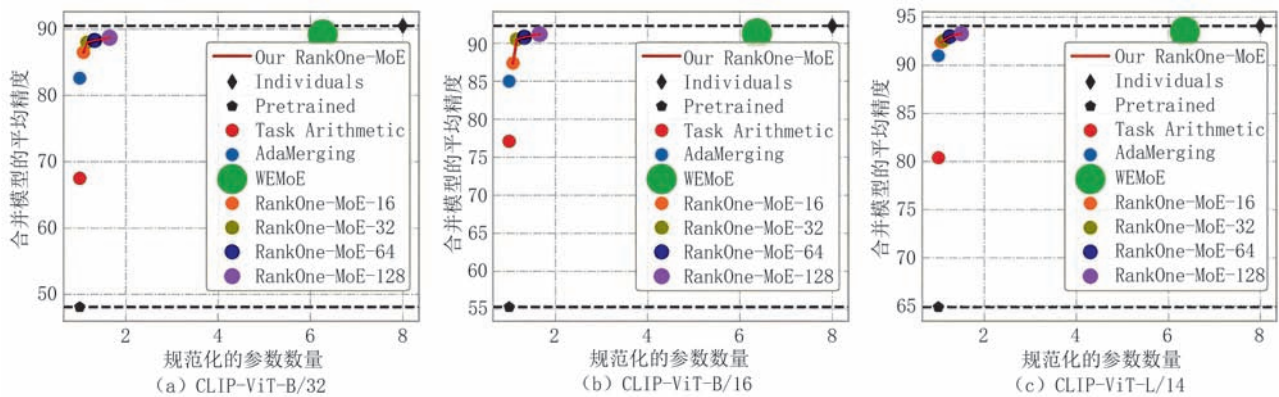


图1 不同模型合并方法在三种架构上合并八个任务时规范化的总参数量与性能关系图(规范化参数数量的定义: $\text{规范化的参数数量} = \frac{\text{合并后的模型参数量}}{\text{预训练模型(或单个专家模型)的参数量}} \in (1, T)$, 其中 T 表示任务数量,该指标用于评估不同方法的参数效率。我们可以观察到,本文提出的 RankOne-MoE 方法通过调整秩一专家数量(如 16、32、64、128),在总参数量(横轴或图形大小)和性能(纵轴高度)之间实现了优异的平衡。图中点位越接近左上角表示性能越高且参数量越小。)

综上所述,静态合并方法在计算资源和参数量控制方面具有明显优势,其简洁的设计使得模型在存储和计算效率上表现优异。然而,由于无法灵活适应不同任务间的多样化需求,静态合并方法通常面临性能不足的问题。相比之下,动态合并方法通过引入任务特定模块或动态路由机制,能够根据输入实例自适应地组合模型参数,从而实现接近甚至优于传统联合训练的多任务学习性能。然而,这种动态性带来的显著代价是需要额外维护大量参数,尤其是专家模块的存储需求急剧增加,导致在资源受限的场景中难以广泛应用。一个自然的问题是:“如何在静态和动态方法之间取得更优的平衡,既保留静态合并方法的高效性,又能部分实现动态合并的灵活性”?

在本文中,我们提出了一个称为“秩一专家混合(RankOne-MoE)用于多任务学习”的方法,用于解决多任务学习中的模型合并问题。如图1所示,该方法在显著减少参数量的同时,能够实现与传统联

合训练多任务学习方法相当的性能,兼具高效性与灵活性。具体而言,我们跟随 WEMoE^[23]中的分析方法对独立训练模型中的主要模块进行详细分析,按照它们对下游任务的相关性强弱,将模型参数划分为两组。在此基础上,本文提出的 RankOne-MoE 完整工作流程如下:首先,对于与下游任务关联性较弱的参数组,我们采用基于任务算术的静态合并方法,将这些参数直接合并到预训练模型中,从而减少冗余存储需求。而对于与下游任务强相关的参数组,我们对其线性层权重参数进行奇异值分解(SVD),将权重分解为左奇异向量和右奇异向量的低秩形式,从而构建出一系列秩为一的专家模块。接着,我们从每个任务的专家模块中挑选出最具代表性的 k 个秩一专家,并将它们统一添加到一个跨任务共享的专家库中,形成一个高效的秩一专家池。然后,为了进一步提升模型的适应性与性能,我们设计了一个动态路由机制,能够根据输入实例的特征,从专家池中动态选择合适的秩一专家进行组

合。最后,我们将静态合并得到的模块和动态合并得到的模块按照原始模型的执行逻辑进行堆叠,得到最终完整的合并模型。通过这种合并策略,我们提出的RankOne-MoE方法实现了性能与参数量的双重优势。实验结果表明,RankOne-MoE在合并多个ViT-B/32模型时,不仅能够相比最先进的静态合并方法在多任务精度方面提升5.40%,同时相比于现有动态合并方法,其总参数量减少了81.45%,充分验证了该方法的有效性和高效性。

总的来说,本文的主要贡献可以概括总结为以下三个关键方面:

(1)本文利用奇异值分解对模型的线性层参数进行分解,构建秩为一的共享专家池。秩一专家的设计显著降低了参数存储需求,同时提升了任务间知识共享的效率,为多任务学习提供了一种高效的低秩模型合并方案。

(2)本文提出了结合静态与动态合并优点的“秩一专家混合(RankOne-MoE)”方法。通过将下游任务相关性较弱的参数组进行基于任务算术的静态合并,以及对相关性较强的参数组进行动态化处理,该方法克服了静态合并性能不足和动态合并资源消耗过高的双重局限。

(3)本文在三种不同参数规模的模型和两种不同任务数量上进行了广泛的实验验证,进一步证明了提出方法的具有较好的模型合并性能和参数高效性,因此具有实用性。

本文的结构组织如下:第1节介绍了研究背景、研究动机以及本文方法的简要概述;第2节回顾了与本文相关的研究工作;第3节介绍了模型合并的符号、问题定义等预备知识;第4节详细阐述了本文提出的方法及其设计细节;第5节展示了大量实验结果,以验证所提方法的有效性和高效性;最后,第6节对本文的研究内容进行了总结与展望。

2 相关工作

在本节中,我们从两个主要方面对本文的相关工作进行详细阐述:一是基于模型合并的多任务学习方法,二是模型微调以及模型压缩技术。

2.1 模型合并用于多任务学习

模型合并(Model Merging)的概念与集成学习(Ensemble Learning)的概念存在一定的相关性,两者的核心思想都是通过融合多个模型的知识来提升目标任务的预测性能。集成学习^[3]需要保留所有

参与融合的模型,并通过对多个模型的预测或输出进行加权平均、投票或其他形式的组合,来生成最终的预测结果。相比之下,模型合并^[21-31]在参数层面对多个模型进行融合,将独立训练的模型参数整合到一个统一的模型中,从而不需要保留原始模型。这种方法在减少存储成本和计算资源需求方面具有明显优势。因此,模型合并是一种更高效的进行跨任务知识迁移的方式。

由于任务之间的潜在干扰以及参数的冲突,最简单的直接参数合并方法可能会导致合并后模型性能的显著下降^[17,32,39,40]。为了解决这一问题,研究者提出了多种优化的模型合并方法,这些方法可以大致分为两类:基于加权合并以及基于子空间合并。(1)基于加权的方法通过为不同模型或其参数分配合适的合并系数,灵活地控制每个模型在最终合并模型中的贡献比例。例如Task Arithmetic^[17]通过网格搜索的方式为每个专家模型搜索一个合并系数。Fisher Merging^[34]根据Fisher信息^[41]来衡量每个参数的重要性,使用这些重要性权重进行加权合并。SLERP^[42]在两个模型的参数之间进行球面插值,以提升合并效果。RegMean^[35]针对线性层设计了一种基于输入数据统计信息的加权合并方法,并利用封闭解提升计算效率。类似的,MATS^[43]利用共轭梯度法求解模型合并系数。MetaGPT^[44]将微调模型相比预训练模型的参数改变量级作为参数重要性衡量标准,并以此为依据合并模型。AdaMerging^[20]提出了一种无监督的熵最小化代理损失,用于自动优化不同模型层级的融合系数。(2)基于子空间的方法通过将稠密的独立模型投影到稀疏的参数子空间中进行合并,有效减少了任务间的参数冲突与干扰。这类方法利用神经网络的过参数化特性,通过剪枝或参数稀疏化的方式,仅保留最重要的参数进行合并。Ties-Merging^[19]基于神经元的参数量级对模型进行剪枝,将绝对值较小的参数(被认为是不重要的参数)移除,仅保留关键参数,并在合并前对剩余参数进行符号对齐。DARE^[28]借鉴Dropout机制^[45],通过随机删除专家模型中一定比例的神经元,并对剩余神经元的量级进行重新缩放,以平衡模型的稀疏性与信息保留。PCB-Merging^[46]同时考虑任务内部和任务之间的重要性来丢弃重要性分数较低的神经元。Model breadcrumbs^[47]认为神经元中较大的或较小的参数可能是异常值,这些参数会对合并过程产生不利影响,因此可以裁剪被认为是异常的参数来提升模型的合并效果。

尽管这两类方法相比简单的参数合并方法在性能上取得了显著提升,但它们本质上仍属于静态合并策略。这意味着合并后的模型参数在推理时是固定的,难以根据不同样本或任务的特定需求进行动态调整,从而在任务多样性较高的场景中表现出适应性不足的问题^[23,30,48]。与本文最相关的一个工作是WEMoE^[23],它提出动态的合并模型参数来提高合并模型的适应性。然而,WEMoE的主要局限在于其需要维护的模型参数量远高于静态合并方法,是其数倍之多,这使得该方法在计算资源受限的场景中难以广泛应用。相反地,本文提出的方法继承了动态模型合并方法的灵活性,同时也保留了静态方法参数量方面的优势,更具实用价值,尤其适用于资源受限但任务多样性丰富的应用场景。

2.2 模型微调以及模型压缩技术

本文的核心技术是实现高效且有效的模型融合,这一技术方向与参数高效微调(或模型复用)以及模型压缩等主流技术具有一定相关性,但也有本质的不同。

模型的参数高效微调(如Adapter微调^[49],低秩(LoRA)微调^[50]和提示(Prompt^[51,52])微调等)相比于传统的全参数微调更节约计算资源,是一种近年来在深度学习领域流行的技术方法^[53-55]。尽管模型合并与参数高效微调都旨在高效复用已有模型,它们的侧重点存在明显差异:参数高效微调主要关注于重用预训练模型,优化其在单个下游任务上的表现。其目标是通过轻量级模块适配特定任务,而无需重新训练整个模型。模型合并则致力于将多个经过训练或微调的专家模型(即在特定任务上表现优异的任务特定模型)融合为一个统一的多任务模型,从而实现对多个任务的综合支持。

模型压缩是一种通过减少模型参数量或计算量来优化模型资源使用的技术方法,其目标是使模型在推理阶段更高效,适用于资源受限的设备或场景^[56-57]。模型压缩的主要方法包括剪枝、量化和低秩分解等。模型压缩与模型合并均注重对模型参数的优化,以减少存储和计算资源开销。然而,模型压缩关注如何减少单个模型的参数量或计算复杂度,重点考虑如何尽可能保持压缩后的模型在单一任务上的性能。然而,模型合并聚焦于将多个相同架构的能力各异的专家模型聚合为一个综合模型,从而避免单独地维护多个专家模型,重点考虑如何避免任务之间的参数冲突等问题。

3 预备知识

本节首先在3.1节介绍模型合并的符号和问题定义,接着在3.2节介绍参数模块划分等预备知识。

3.1 问题定义和符号约定

假设有 T 个需要合并的模型,其参数分别表示为 $\Theta_1, \Theta_2, \dots, \Theta_T$,这些模型均基于同一个主流预训练模型 Θ_0 进行微调。需要注意的是,模型合并设置下不允许访问模型的原始训练数据,而只能获得微调后的模型参数。不失一般性,本文主要关注视觉分类任务,并遵循之前的模型合并方法^[17,20,23]的设置,采用三个不同规模的CLIP-ViT模型^[38]作为预训练模型。每个模型 Θ_i 是一个基于Transformer^[58]架构的模型,共包括 L 个Transformer块(Block),其形式化表示为 $\Theta_i = \{\Theta_i^1, \Theta_i^2, \dots, \Theta_i^L\}$ 。特别地,基于任务算术的模型合并领域常用的一个重要概念是“任务向量”^[17],它由每个任务 t 对应的微调模型 Θ_t 减去预训练模型 Θ_0 得到,任务向量 t 记为 $\tau_t = \Theta_t - \Theta_0$ 。

在基于任务算术的多任务学习中,我们希望通过一种有效的策略将所有任务向量 $\{\tau_t\}_{t=1}^T$ 的知识融合到预训练模型 Θ_0 中得到一个统一的合并模型 Θ_{merge} ,同时尽可能减少任务间的冲突与参数冗余。形式化地,模型合并的目标是找到一个最优合并模型参数集 $\Theta_{\text{merge}}^* = \text{merge}(\Theta_0; \tau_1, \tau_2, \dots, \tau_T)$,以使得合并后的模型能够在所有任务的测试集上达到较高的性能。即,模型合并的优化问题可以表示为:

$$\Theta_{\text{merge}}^* = \arg \min_{\Theta_{\text{merge}}} \frac{1}{T} \sum_{t=1}^T \mathcal{L}_t(f(\Theta_{\text{merge}}; \mathcal{D}_t)) \quad (1)$$

其中, $f(\Theta_{\text{merge}}; \mathcal{D}_t)$ 表示模型参数 Θ_{merge} 在任务 t 的测试数据集 \mathcal{D}_t 上的性能, $\mathcal{L}_t(\cdot)$ 是一个关于任务 t 的可量化的损失函数。

3.2 参数模块划分

在执行模型合并时,对所有参数模块都进行静态合并的方法性能难以令人满意。反之,直接将所有模型的所有参数 $\{\Theta_1, \Theta_2, \dots, \Theta_T\}$ 完整存储用于后续执行动态合并,显然会带来巨大的内存成本。随着任务数量 T 或网络层数 L 的增加,这种存储需求呈线性增长,难以在实际应用中接受,同时也违背了多任务学习追求参数高效性的核心原则。针对这一问题,一个可行的策略是基于“任务相关性分析”的对专家向量的分组^[23],即将模型参数划分为重要

性不同的两部分:(1)弱相关参数组:如果微调后的参数 $\theta_i^{l,i}$ 与预训练模型的参数 $\theta_0^{l,i}$ (l 是Transformer块的第 l 层, i 是这个层中的第 i 个子模块)距离更接近,表示这组参数对下游任务的影响更小,这些参数往往倾向于利用预训练模型本身来捕捉的是任务间的通用特征。我们可以采用静态合并的方法,将这部分参数直接算术合并到预训练模型中,以降低存储成本。(2)强相关参数组:如果微调后的参数 $\theta_i^{l,i}$ 与预训练模型的参数 $\theta_0^{l,i}$ 距离更遥远,那么这组参数与下游任务更相关,即容纳了更多下游任务的知识。因此它需要动态地选择合并来减少任务间的干扰。

以CLIP-ViT架构的Transformer模型为例,如图3所示,它主要包含归一化层、多头注意力层(ATT)和多层感知机层(MLP)。其中,模型的大部分参数量集中在ATT和MLP两个组件中,对模型的存储与计算成本起主导作用。为了优化参数合并效率并降低存储开销,我们需要分析这两个层中参数的重要性及其在任务间的变化程度。跟随WEMoE^[23]的分析方法,我们对MLP层和ATT层参数的变化进行量化分析,使用 L_2 距离衡量它们在不同任务模型间的差异。以第 t 个模型的第 l 块为例,ATT和MLP两个层分别对应的任务向量的 L_2 距离记为:

$$d(\tau_i^{l,ATT}) = \|\theta_i^{l,ATT} - \theta_0^{l,ATT}\|_2 \quad (2)$$

$$d(\tau_i^{l,MLP}) = \|\theta_i^{l,MLP} - \theta_0^{l,MLP}\|_2 \quad (3)$$

为了进一步分析任务间参数变化的整体趋势,我们可以统计 T 个模型在 L 个块中ATT和MLP的平均 L_1 距离如下:

$$\begin{aligned} d_{ATT} &= \frac{1}{T \times L} \sum_{t=1}^T \sum_{l=1}^L d(\tau_i^{l,ATT}) \\ d_{MLP} &= \frac{1}{T \times L} \sum_{t=1}^T \sum_{l=1}^L d(\tau_i^{l,MLP}) \end{aligned} \quad (4)$$

在CLIP-ViT-B/32,CLIP-ViT-B/16和CLIP-ViT-L/14三种主流视觉架构上的结果如图2所示,MLP层的参数变化通常比ATT层更显著,这表明MLP层对任务特定知识的适应性更强。这一结论与WEMoE^[23]中的分析结果保持一致。

综上,在本文中,我们将多层感知机层划分到强相关参数组采取更细粒度的动态合并策略,而将多头注意力层和归一化层划分到弱相关参数组并执行静态合并。

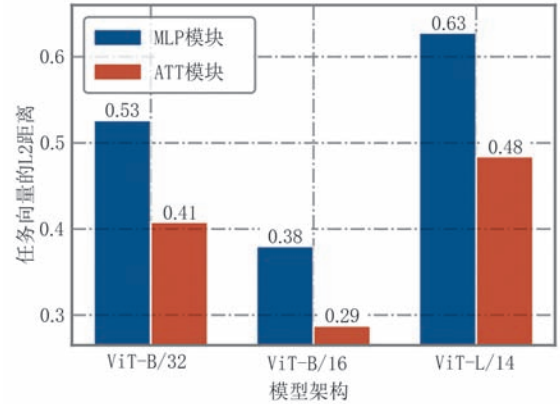


图2 CLIP-ViT架构的Transformer块中MLP模块和ATT模块对应参数的平均 L_2 距离

4 方 法

本节首先在4.1节对提出的秩一专家混合模型进行详细阐述,包括模型总览、秩一专家池构建、动态路由合并和动静模块整合等细节。接着,4.2节中讨论了提出方法的优势。

4.1 秩一专家混合 (RankOne-MoE)

4.1.1 方法动机以及模型总览

正如我们之前在第1节和第2节中讨论的,现有的静态合并的方法(例如,Weight Averaging Modelsoups^[32]、Task Arithmetic^[17]、Ties-Merging^[19]、AdaMerging^[20]、DARE^[28]等)在总参数量方面有明显优势,但参数固定导致总体平均性能较差,在精度要求较高的场景中应用受限。与之相对,存在的动态合并的方式(即WEMoE^[23])则通过根据输入样本动态调整模型参数组合,几乎能够达到传统多任务学习或独立专家模型的性能。然而,WEMoE方法需要维护大量额外参数(如动态路由模块和任务特定专家模块),在计算资源有限的场景中难以广泛应用。

针对上述问题,本文提出了一种全新的秩一专家混合(RankOne-MoE)方法,用于多任务模型合并。RankOne-MoE结合了静态合并和动态合并的双重优势,通过设计跨任务共享的秩一专家池和动态路由机制,在保持参数量接近静态合并方法的同时,显著提升了模型的综合性能。图3展示了本文提出的RankOne-MoE模型的结构总览。具体来说,RankOne-MoE主要包括以下三个关键步骤:(1)秩一专家池构建:通过奇异值分解(SVD)构建跨任务共享的秩一专家池,有效减少参数冗余。

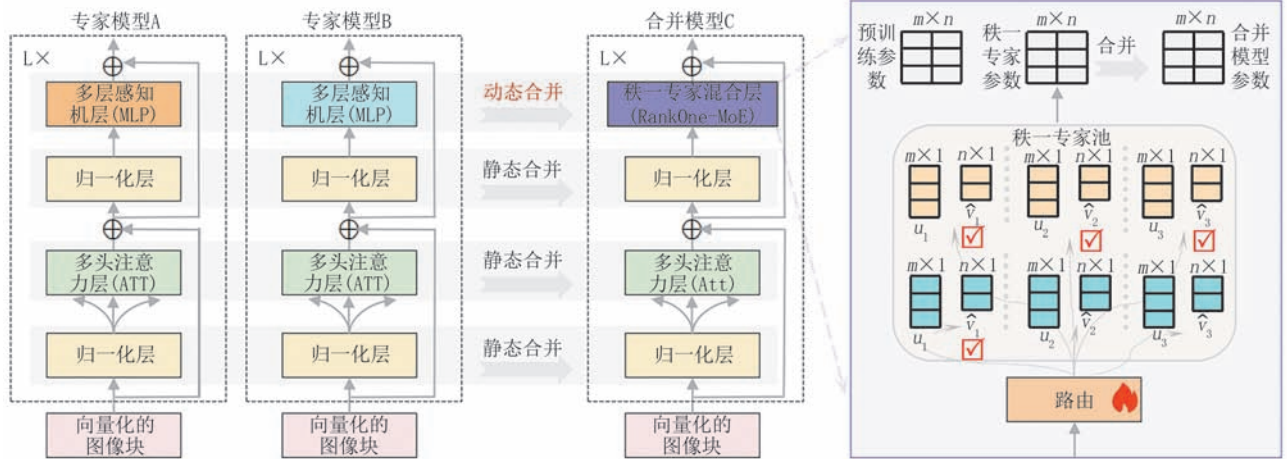


图3 秩一专家混合(RankOne-MoE)模型总览: RankOne-MoE对归一化层和多头注意力层采用基于任务算术的静态合并, 对多层感知机层的权重进行奇异值分解构建秩一专家池并进行动态合并。

(2)动态路由合并:设计动态路由机制,根据输入实例从秩一专家池中选择最相关的专家进行合并。(3)动静模块整合:将动态合并的秩一专家模块和静态合并的模块组装为一个完整的模型。我们在第4.1.2, 4.1.3, 4.1.4三个子节中分别进行详细介绍。

4.1.2 秩一专家池构建

第3.2节的结果表明,在Transformer块中,MLP层的参数变化更显著,因而更适合作为强相关参数组进行动态合并,以适应任务间的个性化需求。然而,若直接保留所有MLP参数用于动态合并(如WEMoE^[23]所采用的方法),将带来昂贵的内存开销,这种高成本显著限制了其在低资源场景中的实际应用。

为解决上述问题,本节提出了一种高效的解决方案:对MLP层的权重矩阵进行奇异值分解(SVD,详见定义4.1),将其分解为低秩的表示,进而构建一系列参数高效的秩一专家模块。为便于理解后续内容,我们首先在定义4.1和定义4.2中给出奇异值分解和矩阵的秩一近似的标准定义。

定义4.1 矩阵的奇异值分解. 对于任意一个 $m \times n$ 的实矩阵 W , 可以将其分解为: $W = U\Sigma V^T$ 。其中, U 是一个 $m \times m$ 的正交矩阵, 称为左奇异向量矩阵, 它的每一列被称作一个左奇异向量。 Σ 是一个 $m \times n$ 的对角矩阵, 主对角线上的元素是 U 的奇异值, 其他非对角元素为零, 且奇异值按降序方式排列; V 是一个 $n \times n$ 的正交矩阵, 称为右奇异向量矩阵, 它的每一列被称作一个右奇异向量。

定义4.2 矩阵的秩一分解. 根据定义4.1, 任意实矩阵 W 的奇异值分解(SVD)可以自然地表述

示为一系列秩一矩阵的线性组合形式:

$$W = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^T \approx \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^T \quad (5)$$

其中, r 是矩阵 W 的秩, 即非零奇异值的个数。 σ_i 是第 i 个奇异值, \mathbf{u}_i (左奇异向量) 和 \mathbf{v}_i (右奇异向量) 分别是 U 和 V 第 i 列。 $\mathbf{u}_i \mathbf{v}_i^T$ 是一个“秩一矩阵”。在实际使用中, 我们可以挑选前 k 个奇异值对应的秩一矩阵来近似矩阵 W 。

在本文的设置中, 每个Transformer块的MLP层包含两个线性(Linear)层, 它们的任务向量分别表示为 $\tau_i^{l, \text{MLP}_1} \in \mathbb{R}^{h \times o}$ 和 $\tau_i^{l, \text{MLP}_2} \in \mathbb{R}^{o \times h}$, 其中 l 表示第 l 个任务或模型, l 表示第 l 个Transformer块, o 和 h 分别表示线性层的输入和输出维度。在实际应用中, 这些维度通常较大。例如, 在ViT-L/14模型中, $h = 4096$ 和 $o = 1024$ 。由于线性层的参数量为维度的乘积, MLP层中这两个线性层的参数量极为庞大, 在模型合并时需要占用大量存储和计算资源。为此, 本文提出对MLP层的两个线性层分别进行奇异值分解, 以提取最重要的低秩特征, 并用 k 个秩一专家的形式对任务相关性进行紧凑建模:

$$\begin{aligned} \tau_i^{l, \text{MLP}_1} &\approx \sum_{i=1}^k \sigma_{i,i}^{l, \text{MLP}_1} \mathbf{u}_{i,i}^{l, \text{MLP}_1} \mathbf{v}_{i,i}^{l, \text{MLP}_1 T} \\ \tau_i^{l, \text{MLP}_2} &\approx \sum_{i=1}^k \sigma_{i,i}^{l, \text{MLP}_2} \mathbf{u}_{i,i}^{l, \text{MLP}_2} \mathbf{v}_{i,i}^{l, \text{MLP}_2 T} \end{aligned} \quad (6)$$

为了在多个任务间进行知识迁移, 我们将所有任务对应层的秩一专家统一添加到一个共享的专家池 $P^{l,i}$ 中, 其中 l 表示Transformer块的层号, i 表示MLP模块的具体子部分(即 MLP_1 或 MLP_2)。具体而言, 对于第 l 层的 MLP_1 模块, 其专家池 P^{l, MLP_1} 定义为

$$P^{l, \text{MLP}_1} = \left\{ \left(\mathbf{u}_{i,i}^{l, \text{MLP}_1}, \hat{\mathbf{v}}_{i,i}^{l, \text{MLP}_1} \right)_{i=1}^k \mid l=1, 2, \dots, T \right\} \quad (7)$$

其中, $\hat{\mathbf{v}}_i^{l, \text{MLP}_1} = \sigma_i^{l, \text{MLP}_1} \cdot \mathbf{v}_i^{l, \text{MLP}_1}$ 。同理, 可以构建 MLP_2 模块的专家池 P^{l, MLP_2} 。

通过这种方法, 我们仅需存储秩一专家池中的低维向量 ($\mathbf{u}_{i,i}^{l, \text{MLP}_1}, \hat{\mathbf{v}}_{i,i}^{l, \text{MLP}_1}$), 而无需存储完整的线性层任务向量参数 τ_i^{l, MLP_1} , 因此显著降低了存储成本。以 τ_i^{l, MLP_1} 为例, 在 ViT-L/14 模型中, 完整的线性层参数量为 4096×1024 , 即约 419 万个参数。而使用我们的方法, 经过秩一分解后, 仅需存储 $(4096 \times k + k \times 1024)$ 个参数, 其中 k 表示奇异值分解后的秩。在本文的默认实验设置中, $k = 32$, 因此所需存储的参数量为 $4096 \times 32 + 32 \times 1024 = 163, 840$ 个, 仅占原始参数量的 3.90%, 表明我们方法在参数方面是高效的。

4.1.3 动态路由合并

在上一节中, 我们为每个 MLP 子模块构建了一个跨任务共享的秩一专家池, 它有效存储了任务相关性强的核心参数。在本节中, 我们进一步阐述如何设计动态路由机制, 为输入实例自动挑选最合适的秩一专家, 并将其动态合并到预训练模型中, 以实现实例级的参数自适应。

为了实现这一目标, 我们定义了一个路由函数 $\mathcal{R}(x): \mathbb{R}^{h_x} \rightarrow \mathbb{R}^{kT}$, 其中, x 是输入实例的隐式特征, h_x 表示特性 x 的维度。路由的输出维度是 $k \times T$, 其中 T 是任务数量, k 是每个任务向专家池 $P^{l,i}$ 中添加的秩一专家的数量。换句话说, 路由函数 $\mathcal{R}(x)$ 的作用是根据输入特征 x 为每个任务的秩一专家分配一个合并系数。不失一般性, 本文采用单层线性层来实现路由函数, 且每个 Transformer 块仅需一个路由。具体而言, 对于第 l 层 MLP_i 子模块, 其秩一专家池 $P^{l,i}$ 中的专家将根据 $\mathcal{R}(x)$ 的输出分配权重 $\alpha = \{\alpha_{i,j}\}_{i=1,j=1}^{T,k}$ 。最终合并的参数表示为:

$$\Theta_{\text{merge}}^{l,i} = \Theta_0^{l,i} + \sum_{i=1}^T \sum_{j=1}^k \mathbb{I}(\alpha_{i,j}) \cdot \alpha_{i,j} \cdot ((u_{i,j}^{l,i} \cdot \hat{v}_{i,j}^{l,i})^T) \quad (8)$$

其中, $\Theta_0^{l,i}$ 是预训练模型的基础参数, $\mathbb{I}(\alpha_{i,j})$ 是一个指示函数, 如果 $\alpha_{i,j}$ 是加权向量 α 中前 $p\%$ (本文中 p 默认取值为 75) 最大的值, 则 $\mathbb{I}(\alpha_{i,j})$ 取值为 1, 否则为 0。通过这种机制, 模型仅选择与当前输入相关性最强的专家参与合并, 从而实现动态路由的高效性和参数选择的精确性。这样不仅减少了不必要的计算, 还显著提升了合并模型的适应性和推理性能, 特别是在任务多样性较高的场景中。

路由网络需要被更新, 以便准确识别出最相关的专家。然而, 在模型合并的设置下, 原始训练数据通常不可用, 这对路由网络的更新提出了挑战。跟

随 AdaMerging^[20] 和 WeMoE^[23] 的设置, 我们利用无标签的测试数据调整路由网络的参数。具体而言, 路由函数通过最小化基于无监督代理损失的目标函数 (如熵最小化) 进行优化, 从而确保路由机制能够高效地选择适合当前输入的专家。路由函数的优化目标基于熵最小化, 定义如下:

$$-\mathbb{E}_{t \in [T], \mathbf{x}_t \sim \mathcal{D}_t^{\text{test}}} f(\Theta_{\text{merge}}; \mathbf{x}_t) \log f(\Theta_{\text{merge}}; \mathbf{x}_t) \quad (9)$$

其中, $f(\Theta_{\text{merge}}; \mathbf{x}_t)$ 表示完整的合并模型 Θ_{merge} 对测试数据 $\mathbf{x}_t \in \mathcal{D}_t^{\text{test}}$ 的类别预测概率。由于优化过程仅涉及简单的熵计算和少量门控网络参数更新, 其计算开销和显存使用较低。例如, 在来自八个任务的 ViT-B/32 模型实验中, 可以在 8 分钟内完成路由网络的优化, 为实际部署提供了高效的解决方案。

4.1.4 动静模块整合

通过 4.1.3 中的方案, 我们对高任务相关度的参数模块 (即 MLP) 完成了动态合并。对于剩余的需要执行静态合并的参数模块 (例如多头注意力层以及归一化层), 我们采用基于 Task Arithmetic^[17] 的策略进行合并。以第 l 个 Transformer 块的多头注意力模块 (ATT) 为例, 其合并规则表述如下:

$$\Theta_{\text{merge}}^{l, \text{ATT}} = \Theta_0^{l, \text{ATT}} + \lambda \cdot \sum_{i=1}^T \tau_i^{l, \text{ATT}} \quad (10)$$

其中, λ 是一个超参数, 用于控制预训练模型参数与任务特定参数之间的平衡, 默认取值为 0.3。其他 ATT 层以及归一化层采用类似的合并策略。

最终, 我们将合并完成的 MLP 模块, 多头注意力模块以及 LayerNorm 归一化模块按照原始的模块顺序和链接方式堆叠为完整的合并模型, 该模型用于进行部署以及完成推理任务。

4.2 讨论

在 4.1 节中, 我们详细阐述了本文提出的 RankOne-MoE 方案的实现细节。在本节中, 我们对该方案进行全面总结, 以清晰地概括其核心贡献与优势:

(1) 静态与动态合并的结合

RankOne-MoE 结合了静态合并和动态合并的优点。对于与下游任务相关性较弱的模块 (如多头注意力层 ATT 和归一化层), 通过基于 Task Arithmetic 的静态合并策略, 有效减少了参数存储需求; 而对于任务相关性较强的模块 (如多层感知机 MLP 层), 引入秩一专家与动态路由机制, 以实现任务特定特征的精细建模。基于此, RankOne-MoE 在性能与效率之间实现了良好的平衡。

(2) 秩一专家的高效构建

秩一专家是线性层的最小构成单元。RankOne-MoE将线性层分解为一系列秩一专家，并构建一个跨任务共享的秩一专家池，有效促进了任务间的知识迁移，同时也大幅降低存储和计算成本。此外，在挑选专家时，我们设置了可控的挑选比例，从而隔绝了不相关或者相关性较弱的秩一专家。

(3) 无标签依赖与计算高效

RankOne-MoE中路由的优化目标函数完全基于无标签的测试数据设计，适用于模型合并场景中原始训练数据不可用的情况。此外，其优化过程仅涉及简单的熵计算和参数更新，其计算开销较低，可在数分钟到数十分钟内完成整个优化过程。

通过以上创新设计，RankOne-MoE在多任务模型合并的性能、效率、适用性和扩展性方面展现了综合优势，为大规模多任务学习场景提供了一种高效的基于模型融合的解决方案。

5 实验

在本节中，我们通过大量实验验证了所提方法的有效性。首先，在第5.1节中介绍了实验设置；接着，在第5.2节详细展示了多任务模型合并的性能表现；最后，5.3节从多个角度对提出的方法进行了深入分析。我们的RankOne-MoE代码开源地址：<https://github.com/EnnengYang/RankOne-MoE>。

5.1 实验设置

(1) 数据集

基于FusionBench^[59]开源库，我们严格遵循主流模型合并工作的设置^[17, 20, 23]，在主要实验和分析中，选用了八个广泛使用的视觉数据集作为任务验证模型合并方法的有效性。这八个数据集包括：SUN397^[60]（场景分类）、Cars^[61]（汽车分类）、RESISC45^[62]（遥感图像分类）、EuroSAT^[63]（地理图像分类）、SVHN^[64]（街景数字识别）、GTSRB^[65]（交通标志分类）、MNIST^[66]（手写数字分类）、DTD^[67]（图像纹理分类）。这些数据集覆盖了广泛的类别，具有较高的任务多样性和挑战性。此外，我们也进行了更大任务数量的模型合并，即20个视觉分类任务^[68]，从而验证我们的方法在不同模型规模和任务数量下的有效性。

(2) 模型架构

为了验证模型合并方法的普适性和性能，我们遵循之前的工作^[20, 23]选择了三种不同规模的ViT

预训练模型：CLIP-ViT-B/32、CLIP-ViT-B/16和CLIP-ViT-L/14^[37-38]。这些预训练模型是当前视觉任务中常用的强基线模型，具有不同的参数规模和计算复杂度，能够为实验提供多样的基准。在实验中，我们分别使用上述八个数据集对这三种预训练模型进行微调，从而为每个任务生成对应的专家模型。微调后的专家模型参数将作为输入，用于本文提出的RankOne-MoE方法以及其他对比方法的合并实验。

(3) 对比基线

本文对比了八种流行且前沿的模型合并方法，涵盖静态合并和动态合并两大类，其中静态合并的方法包括：基于加权合并的Weight Averaging^[32]、Fisher Merging^[34]、RegMean^[35]、Task Arithmetic^[17]和AdaMerging^[20]、基于子空间合并的DARE^[28]和Ties-Merging^[19]。动态合并的方法包括WEMoE^[23]。这些方法的介绍详见第2节。此外，我们还提供了三个非模型合并基线的性能作为参考，它们分别是直接利用预训练模型进行分类(Pre-trained)，使用每个任务对应的微调模型分类该任务(Individual)，使用传统多任务学习方法联合训练一个MTL模型(Traditional MTL)。这些基线方法覆盖了多种合并策略，为实验提供了全面的对比维度。

(4) 评估指标

在本文中，我们采用Top-1分类准确率作为每个任务的评估指标，即模型在给定输入的情况下正确预测目标类别的比例，这是视觉分类任务中最常用的精度评价标准。此外，我们还汇报了八个任务的平均准确率，以作为各方法性能的综合评估指标。

(5) 实现细节

为了确保实验的公平性和结果的可比性，我们基于模型合并统一评估框架FusionBench^[59]对所有基线方法进行实现，并参考了原始论文中的超参数设置。在该框架下，每个方法均在相同的硬件环境和数据划分下进行测试，从而消除环境差异对结果的影响。对于本文提出的RankOne-MoE方法，我们采用Adam优化器^[69]来微调路由参数，以确保动态路由机制能够高效地调整秩一专家分配权重。优化器的学习率设置为 1×10^{-4} ，批处理大小为16，训练步骤固定为1000步，以保证路由网络在合理时间内优化。对于RankOne-MoE中的关键超参数设置，我们在[16, 32, 64, 128, 256, 512]中搜索添加到专家池的秩一专家数量 k 。另外，我们在[25%, 50%, 75%, 100%]中搜索专家选择的比例，

即动态路由机制从专家池中选择的前 $p\%$ 重要专家,用于动态参数合并。除非特别说明,本文实验的默认设置为:秩一专家数量 $k=32$,专家选择比例 $p=75\%$ 。这一默认配置在性能和资源效率之间实现了较好的平衡。

5.2 多功能合并性能

我们在5.2.1和5.2.2节中分别将本文提出的RankOne-MoE方法与现有的静态合并方法(以及

三个非模型合并方法)和动态合并方法进行详细比较。这些实验旨在全面评估RankOne-MoE在多任务模型合并场景中的性能表现,尤其是其在精度与参数量之间的权衡能力。

5.2.1 对比静态合并的方法和非模型合并方法

表1、表2和表3分别展示了不同模型合并方法在ViT-B/32、ViT-L/14和ViT-B/16三种架构上的多任务合并性能。基于这些实验结果,有以下重要发现。

表1 合并八个CLIP-ViT-B/32模型时的多任务性能对比

方 法	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	平均精度
Pre-trained	63.2	59.6	60.2	45.0	31.6	32.6	48.3	44.4	48.1
Individual	75.3	77.7	96.1	99.9	97.5	98.7	99.7	79.4	90.5
Traditional MTL	73.9	74.4	93.9	98.2	95.8	98.9	99.5	77.9	88.9
多任务模型合并方法									
Weight Averaging	65.4	62.4	70.6	75.7	64.5	54.9	86.2	50.5	66.3
Fisher Merging (NeurIPS 2022)	67.0	68.2	72.8	75.7	80.1	56.3	88.7	53.5	70.3
RegMean (ICLR 2023)	67.8	68.9	82.7	94.3	90.5	78.9	97.7	64.2	80.6
Task Arithmetic (ICLR 2023)	57.0	55.7	64.7	73.2	77.9	68.4	96.0	47.1	67.5
DARE (ICML 2024)	57.0	55.3	64.6	73.2	77.6	68.2	96.1	47.7	67.5
Ties-Merging (NeurIPS 2023)	67.0	64.1	74.3	74.5	77.7	69.3	94.1	53.9	71.9
AdaMerging (ICLR 2024)	67.8	71.2	83.8	92.1	87.9	93.0	98.2	66.9	82.6
WEMoE (ICML 2024)	74.5	77.0	93.5	96.8	96.8	98.8	99.5	76.7	89.2
RankOne-MoE (我们的)	72.1	74.6	92.6	97.5	94.9	98.0	99.3	75.1	88.0

表2 合并八个CLIP-ViT-L/14模型时的多任务性能对比

方 法	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	平均精度
Pre-trained	68.2	77.9	71.3	61.3	58.4	50.6	76.4	55.4	64.9
Individual	82.3	92.4	97.4	99.9	98.1	99.2	99.7	84.1	94.1
Traditional MTL	80.8	90.6	96.3	96.3	97.6	99.1	99.6	84.4	93.5
多任务模型合并方法									
Weight Averaging	72.5	81.5	82.3	88.5	81.6	74.0	96.6	61.7	79.8
Fisher Merging (NeurIPS 2022)	69.7	77.8	69.8	99.0	62.2	58.2	84.7	57.0	72.3
RegMean (ICLR 2023)	75.3	88.2	90.6	96.8	95.8	92.1	98.4	72.9	88.8
Task Arithmetic (ICLR 2023)	71.9	78.9	80.5	84.6	87.4	83.4	98.0	58.5	80.4
DARE (ICML 2024)	72.0	78.9	80.5	84.5	87.6	83.5	98.0	58.7	80.5
Ties-Merging (NeurIPS 2023)	74.7	83.1	86.5	89.7	89.6	85.1	97.7	63.8	83.8
AdaMerging (ICLR 2024)	78.0	90.7	90.9	96.2	94.9	97.4	98.5	81.5	91.0
WEMoE (ICML 2024)	81.5	92.5	95.8	98.1	97.5	99.3	99.5	83.8	93.5
RankOne-MoE (我们的)	79.5	91.1	94.5	97.8	96.0	99.2	99.2	83.6	92.6

(1)对比三种非模型合并类的方法,我们可以得出以下观察:首先,直接使用预训练模型(Pre-trained)对八个下游任务进行预测时,性能较差。这是因为预训练模型主要学习了通用的知识表示,而未针对任何具体的下游任务进行优化,因此难以满足目标任务的特定需求。其次,独立训练的专家模

型(Individual)通常表现最好,因为每个模型专注于一个特定任务,避免了多任务学习中常见的任务冲突或干扰。然而,这种方法需要为每个任务维护一份独立的模型参数,导致内存成本和存储需求非常高,在任务数量较多时不具备实际可行性。最后,传统的多任务学习(Traditional MTL)方法表现也

表3 合并八个 CLIP-ViT-B/16 模型时的多任务性能对比

方法	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	平均精度
Pre-trained	65.5	64.6	66.3	54.1	51.9	43.4	51.7	44.9	55.3
Individual	78.9	85.9	96.6	99.0	97.6	99.0	99.7	82.3	92.3
多任务模型合并方法									
Weight Averaging	68.7	69.0	75.0	83.2	74.9	62.5	93.7	51.1	72.3
Fisher Merging (NeurIPS 2022)	70.8	71.8	76.2	93.4	77.4	61.2	90.7	52.3	74.2
RegMean (ICLR 2023)	71.1	76.4	86.0	95.4	93.9	86.5	98.4	64.3	84.0
Task Arithmetic (ICLR 2023)	65.9	68.3	75.4	84.5	88.8	81.9	98.0	53.9	77.1
DARE (ICML 2024)	65.9	68.2	75.4	84.2	88.6	81.7	98.0	54.2	77.0
Ties-Merging (NeurIPS 2023)	70.6	71.2	79.8	87.5	83.2	76.2	96.4	55.4	77.5
AdaMerging (ICLR 2024)	70.6	79.6	86.1	93.6	93.5	95.4	98.1	62.9	85.0
WEMoE (ICML 2024)	76.5	85.1	94.4	98.8	97.1	99.0	99.6	80.2	91.3
RankOne-MoE (我们的)	76.0	83.5	93.6	98.1	96.6	98.4	99.4	78.6	90.5

较为优秀。通过联合训练的方式,这类方法基于所有任务数据共同优化一个多任务模型,有效利用了任务间的共享特性。然而,这种方法要求显式收集所有任务的数据,面临较高的数据管理成本,并且在实际应用中可能存在数据隐私泄漏的风险,特别是在跨组织或跨领域场景中。

(2)对比七种静态合并类的方法,我们观察到以下结果:最简单的权重平均合并方法(Weight Averaging)表现较差,因为它未能区分不同模型对最终合并模型的贡献,导致任务相关性较弱的模型对整体性能的负面影响较大。任务算术(Task Arithmetic)通过从独立模型中提取任务相关的专家向量并合并到预训练模型中,同时设置了更合适的合并系数,通常能够优于简单的权重平均方法,因其更注重任务特定知识的保留。基于稀疏子空间的合并方法 DARE 和 Ties-Merging 也被广泛采用。DARE 通过随机移除部分参数实现稀疏化并减少存储需求,但这一策略可能丢失下游任务的重要信息,因此其性能相较任务算术并没有显著提升。相比之下,Ties-Merging 在合并过程中对齐参数的符号,有效减少了任务间的冲突,因此在任务算术的基础上进一步提升了模型的性能。先进的基于加权的方法(如 Fisher Merging、RegMean、AdaMerging)通过预定义的规则或可学习的方式,为不同模型的参数分配不同的贡献权重,与简单的权重平均相比,在性能上有显著提升。其中,AdaMerging 利用无标签测试数据优化合并系数,使得其在静态合并方法中表现最佳。例如,在 ViT-B/32 架构上,AdaMerging 达到了 82.6 的精度,明显优于静态方法中表现第二好的 RegMean,后者精度为 80.6。

(3)对比三种不同的合并架构(即 ViT-B/32、ViT-B/16 和 ViT-L/14),我们发现参数规模越大的模型(如 ViT-L/14)在合并过程中表现出更高的性能和适应性。这可能是由于大规模模型通常具有更多的参数冗余,这为任务间的知识整合提供了更大的灵活性。此外,参数规模较大的模型对合并过程中出现的参数偏移或冲突可能具有更强的鲁棒性。

(4)对比不同的任务数量(即 ViT-B/32 上合并 8 个任务和 ViT-B/32 上合并 20 个任务)下的模型合并,我们发现在较大任务数量下动态合并的方法(WEMoE 和 RankOne-MoE)仍然一致地领先于各个静态合并的方法。另外,如表 5 所示,在 20 个任务中,基于任务算数的 Task Arithmetic 和 Ties-Merging 与普通的 Weight Averaging 的相比没有明显的提升,这是由于它们对类型丰富的任务仍然采用单一的合并缩放系数,限制了合并模型的有效性。这一点也能从 AdaMerging 的性能明显提升看出,仔细地权衡各个层的合并系数对最终的合并性能非常关键。

(5)对比本文提出的 RankOne-MoE 方法与其他静态模型合并方法的实验结果,我们发现 RankOne-MoE 在三种架构上均表现出一致的优异性能,跨越多个任务的合并性能显著优于所有静态方法。更重要的是,RankOne-MoE 的合并性能非常接近传统有数据联合训练的多任务学习基线(Traditional MTL),展现了其在基于模型合并的多任务场景中的强大潜力。RankOne-MoE 方法的优异性能主要得益于其对关键参数模块的细粒度动态组合策略。

表4 不同变种在合并八个 CLIP-ViT-B/32 模型时的多任务性能对比

方法	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	平均精度
Task Arithmetic	57.0	55.7	64.7	73.2	77.9	68.4	96.0	47.1	67.5
RankOne-MoE (完整的)	72.1	74.6	92.6	97.5	94.9	98.0	99.3	75.1	88.0
变种一(消除“秩一分解”)	62.7	59.9	89.4	92.6	93.5	91.0	97.0	54.4	80.0
变种二(消除“动态路由”)	64.5	63.5	73.6	88.9	83.4	74.0	96.8	55.1	75.0

表5 合并二十个 CLIP-ViT-B/32 模型时的多任务性能对比

方法	平均精度
Weight Averaging	61.10
Fisher Merging (NeurIPS 2022)	62.11
RegMean (ICLR 2023)	70.02
Task Arithmetic (ICLR 2023)	61.38
Ties-Merging (NeurIPS 2023)	60.63
AdaMerging (ICLR 2024)	69.41
WEMoE (ICML 2024)	74.27
RankOne-MoE (我们的)	72.90

5.2.2 对比动态合并的方法

在本节中,我们从模型合并性能、合并模型的总参数量两个角度来对比与本文最相关的动态合并方法 WEMoE。

从性能的角度,如表1、表2、和表3所示,本文提出的 RankOne-MoE 方法与当前最先进的动态合并方法 WEMoE 在多任务精度上非常接近,性能差距均在 1% 左右。这表明 RankOne-MoE 在实现高性能合并方面具有较强的竞争力。

然而,需要特别强调的是,RankOne-MoE 在参数量方面表现出显著的高效性。如图1所示,默认配置下的 RankOne-MoE-32 相较于 WEMoE 在存储需求上有大幅减少。具体而言,独立模型不减少模型参数,因此规范化后的参数量是预训练模型的 T 倍。Task Arithmetic、AdaMerging 等方法执行模型合并后,合并模型的参数量与预训练模型完全等价,因此规范化后的参数量是 1,而 WEMoE 需要维护多个满秩的专家和路由模型,因此规范化后的参数量明显大于 1(例如在 ViT-B/32 和 ViT-L/14 上分别为 6.26 和 6.35)。本文提出的 RankOne-MoE 相比 WEMoE 仅需要保存少量的秩一矩阵,因此参数量非常接近预训练模型,例如在模型配置 RankOne-MoE-32 时,在 ViT-B/32 和 ViT-L/14 上的参数量分别为 1.16 和 1.12。这表明 RankOne-MoE 在保持高性能的同时,大幅降低了存储需求,尤其适合多任务场景中的资源受限应用。

在图4中,我们进一步统计了合并不同数量的模型(例如[2, 3, 4, 5, 6, 7, 8])时,RankOne-MoE 与

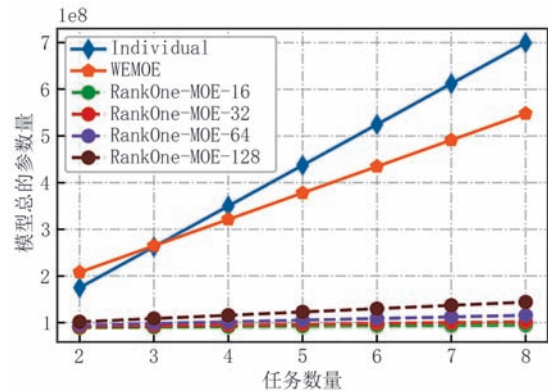


图4 不同任务数量下模型的总参数量对比(Individuals)

独立模型(Individual)和 WEMoE 的总参数量对比情况。通过观察,我们发现以下关键趋势:(1)独立模型的线性增长:随着任务数量的增加,独立模型的参数量呈线性增长趋势,因为每个任务都需要维护一份完整的参数,这种增长在任务数量较多时带来了显著的存储成本。(2)WEMoE 的参数优势限制:WEMoE 的参数量相比独立模型在任务数量较多时才开始体现出优势。例如,当任务数量为 2 或 3 时,由于 WEMoE 需要额外维护一个路由网络,其参数量甚至与独立模型持平或稍高。(3)RankOne-MoE 的显著高效性:相比之下,RankOne-MoE 在所有任务数量下均展现出显著的参数效率。通过秩一专家的设计和跨任务共享的专家池机制,RankOne-MoE 的参数量增长极为缓慢。RankOne-MoE 的参数效率随着任务数量的增加而持续放大,这使其特别适合任务规模大、场景复杂的多任务学习应用。

5.2.3 总结

前两节的实验结果显示,本文提出的 RankOne-MoE 方法在性能和参数效率之间实现了良好的平衡,显著优于静态方法,并在参数量大幅减少的同时接近甚至持平动态方法的性能。这些优势使得 RankOne-MoE 成为一种在性能和资源效率上均具优秀表现的多任务模型合并解决方案。

5.3 分析

本节从多个层面对提出的 RankOne-MoE 方法进行深入分析,以全面理解其在以下几个关键方面的

表现和机制:各个核心模块有效性、优化阶段计算资源消耗、推理阶段计算资源消耗、方法稳定性分析、超参数的鲁棒性和不同任务对专家选择机制的影响等。

5.3.1 模块有效性分析

在本节中,我们对 RankOne-MoE 的创新模块(如秩一分解和动态路由)进行消融分析,以便确定各模块的独立贡献。具体来说,为了定量分析“秩一分解”模块和“动态路由”模块对最终性能的影响,我们设计了两个新的变种:(1)变种一:将 RankOne-MoE 中的选择秩为 1 专家替换为选择一个秩 k 的专家,从而验证细粒度的“秩一分解”策略的有效性。(2)变种二:将 RankOne-MoE 中的动态路由模块调整为静态路由的方式,从而验证“动态路由”策略的有效性。在 ViT-B/32 上的结果如表 4 所示,我们观察到,两个变种的性能(变种 1 和变种 2 的平均精度分别为 80.0 和 75.0)相比完整版本的 RankOne-MoE 性能(即 88.0)都有所下降,但仍然优于普通的模型合并(即 Task Arithmetic 的 67.5)方法,这些定量的性能比较展示了两个创新模块的有效性。

5.3.2 优化阶段计算资源消耗分析

在本文涉及的模型合并方法中,AdaMerging、WEMoE 和我们的 RankOne-MoE 都需要在无标签的测试数据上进行适应,以优化合并权重或路由参数。为了评估这些方法在优化阶段的资源消耗,我

们也比较了它们在优化时间开销和显存消耗方面的差异。为确保对比的公平性,所有实验均在单张 NVIDIA A6000 显卡(48 GB 显存)上进行。具体来说,除了度量在 ViT-B/32 架构上合并 8 个任务模型的资源消耗,我们也扩展了显存消耗统计的实验:(1)任务扩展:在 ViT-B/32 架构下,将任务数量从 8 增加到 20,评估显存消耗和优化时间的变化。(2)模型扩展:增加了 ViT-L/14 架构(也即更大、更深的模型)在 8 个任务场景下的显存开销和优化时间对比。

优化时显存消耗的结果如表 6 所示,我们可以观察到:(1)在更大的架构下,RankOne-MoE 在微调阶段的资源消耗的优势得以进一步凸显。在 ViT-L/14(8 个任务)的场景中,RankOne-MoE 的显存开销为 17568.36 MB,比 WEMoE 和 AdaMerging 的 24535.53 MB 和 28285.94 MB 明显更低(这个资源要求已经超出了单张 24GB 消费级显卡的显存容量)。(2)随着任务数量从 8 增加到 20,RankOne-MoE 的显存开销增幅远低于 WEMoE 和 AdaMerging,这得益于秩一专家的共享机制,可以有效避免显存消耗的线性增长。例如在 20 个任务的场景中,RankOne-MoE 的显存消耗依然显著低于两种基线方法,仅为 2179.36 MB。这些结果验证了 RankOne-MoE 在涵盖更多任务和更大模型架构场景下的显存效率优势。

表 6 不同模型合并方法在不同架构和任务数量下优化阶段的显存开销对比

方法	ViT-B/32 (8个任务)	ViT-B/32 (20个任务)	ViT-L/14 (8个任务)
AdaMerging (静态合并方法)	4863.91 MB	8863.58 MB	28285.94 MB
WEMoE(动态合并方法)	3744.19 MB	6361.93 MB	24535.53 MB
RankOne-MoE(我们的方法)	1963.20 MB	2179.36 MB	17568.36 MB

优化时间消耗的结果如表 7 所示,我们可以观察到如下现象:(1)AdaMerging 耗时最长,RankOne-MoE 的优化时间介于 AdaMerging 和 WEMoE 之间。(2)随着任务数量增加(例如从 8 个任务增加到 20 个任务时),三个方法的优化时间都会增加,这是因为它们都需要在所有任务的无标签

数据下优化模型,因此增加任务数量会增加每个优化步骤中总损失的计算时长。(3)在更大的架构下,优化时间也会增加,这是很自然的,因为大的架构(神经网络层数更深)每条样本的处理时间明显需要更久,参数更新时需要进行反向传播计算的时间自然也更长。

表 7 不同模型合并方法在不同架构和任务数量下优化阶段的优化时间对比

方法	ViT-B/32 (8个任务)	ViT-B/32 (20个任务)	ViT-L/14 (8个任务)
AdaMerging (静态合并方法)	10.47 min	43.27 min	62.53 min
WEMoE(动态合并方法)	7.07 min	26.01 min	56.84 min
RankOne-MoE(我们的方法)	9.13 min	34.88 min	55.38 min

5.3.3 推理阶段计算资源消耗分析

在本部分,我们提供了 Task Arithmetic、

AdaMerging、WEMoE 和 RankOne-MoE 在推理阶段显存消耗与总推理时间的对比实验,以提供更全

面的推理成本分析结果。具体来说,我们在 ViT-B/32(8个任务和20个任务)和 ViT-L/14(8个任务)架构下,统计了四种模型合并方法的推理阶段开销,包括显存消耗(表8)和推理总时间(表9)。以下是实验结果的主要发现:

对比推理显存消耗情况,我们有如下观察:(1)静态合并方法(Task Arithmetic 和 AdaMerging)显存消耗最低,因为它们只有相同于预训练模型的参数,例如在 ViT-B/32(8个任务)场景中,显存消耗分别为 963.42 MB 和 975.91 MB。(2)动态合并方

法(WEMoE)显存消耗显著高于其他方法,例如在 ViT-L/14(8个任务)场景中,消耗达到 10 063.64 MB,是 Task Arithmetic 的约 2.6 倍。随任务数量增加,显存消耗大幅增加(如 ViT-B/32 从 8 个任务的 2750.65 MB 增加至 20 个任务的 5346.00 MB)。(3)RankOne-MoE 显存消耗在静态方法与动态方法之间,略高于静态方法但显著低于动态方法。例如,在 ViT-B/32(20 个任务)场景中,RankOne-MoE 的显存消耗为 1247.25 MB,比 WEMoE 的低 76.7%。

表 8 不同模型合并方法在不同架构和任务数量下推理/评估阶段的显存开销对比

方 法	ViT-B/32 (8个任务)	ViT-B/32 (20个任务)	ViT-L/14 (8个任务)
Task Arithmetic(静态合并方法)	963.42 MB	963.42 MB	3772.63 MB
AdaMerging (静态合并方法)	975.91 MB	976.85 MB	3787.01 MB
WEMoE(动态合并方法)	2750.65 MB	5346.00 MB	10 063.64 MB
RankOne-MoE(我们的方法)	1094.31 MB	1247.25 MB	4086.68 MB

表 9 不同模型合并方法在不同架构和任务数量下推理/评估阶段的时间开销对比

方 法	ViT-B/32 (8个任务)	ViT-B/32 (20个任务)	ViT-L/14 (8个任务)
Task Arithmetic(静态合并方法)	2.88 min	5.41 min	14.92 min
AdaMerging (静态合并方法)	2.66 min	5.01 min	14.74 min
WEMoE(动态合并方法)	2.72 min	5.85 min	16.21 min
RankOne-MoE(我们的方法)	2.69 min	5.38 min	15.25 min

对比推理时间统计,我们可以发现:

(1)整体来看,所有方法的推理时间都非常接近,也即提出的 RankOne-MoE 相比基线方法在推理时并没有明显的延迟,主要得益于以下两个方面:

①高效的矩阵运算

GPU 对矩阵运算的高效支持。虽然 RankOne-MoE 通过秩一分解引入了多个奇异向量(秩一矩阵)的计算需求,但这些计算可以被写成一次性的大矩阵运算,从而充分利用 GPU 的并行计算能力。

②批量测试优化

在批量推理场景下(如实际多任务应用中通常会以批量处理输入),RankOne-MoE 的效率进一步提升。这是因为合并后的权重可以针对整个批次的输入实例一次性计算并共享,而不需要为每个样本单独执行权重组合的过程。

(2)当任务数量增加(即当任务数量从 8 增加到 20 时)或者网络架构增大时(即采用 ViT-L/14 架构),所有方法的总推理时间都会变长,这也是符合实际的。

总的来说,RankOne-MoE 在推理阶段的显存和时间开销均低于动态方法(WEMoE),同时较接

近静态方法(Task Arithmetic 和 AdaMerging),因此提供了在精度和效率间更好的权衡和选择方案。

5.3.4 对比参数量化方法

方法部分提到最相关的基线 WEMoE^[23]的一个主要问题是“它直接保留所有的 MLP 参数用于动态合并,将带来昂贵的内存开销”。一个潜在的解决方法是参数量化,也即对 WEMoE 中的 MLP 参数进行低精度量化。基于 PyTorch 对 qint8, quint8, qint32, float16 等数据类型¹的支持,我们设计了两种量化变体,分别命名为 Quantized-WEMoE-FP16 和 Quantized-WEMoE-INT8。前者 and 后者分别将 MLP 参数从 FP32 转换为 FP16 和 INT8,从而降低推理的显存开销和存储成本。在 ViT-B/32 与 ViT-L/14 两种架构下,我们合并了八个任务的专家模型,并比较了多种方法:原始 WEMoE、量化后的 Quantized-WEMoE-FP16 与 Quantized-WEMoE-INT8、本文提出的 RankOne-MoE、经典模型合并基线 Task Arithmetic,以及仅使用预训练模型。比较维度包括合并模型的多任务精度、推理显存开销和存储成本。

结果如图5所示,我们有以下观察:(1)在精度方面,原始 WEMoE、量化后的 Quantized-WEMoE-FP16、Quantized-WEMoE-INT8,以及 RankOne-MoE 的表现几乎一致,并明显优于静态合并基线 Task Arithmetic。这表明动态合并能根据输入实例或任务特性分配合并权重,具备显著优势。(2)在推

理显存开销方面,原始 WEMoE 的开销最高,因为所有 MLP 参数均为 Float32。量化后版本显著降低了显存占用,尤其在 CLIP-ViT-L/14 架构下,Quantized-WEMoE-INT8 将 WEMoE 的显存成本从 10 063.64 MB 降至 5441.25 MB(参见图 5(b)),但 RankOne-MoE 仍保持最低的 4086.68 MB。

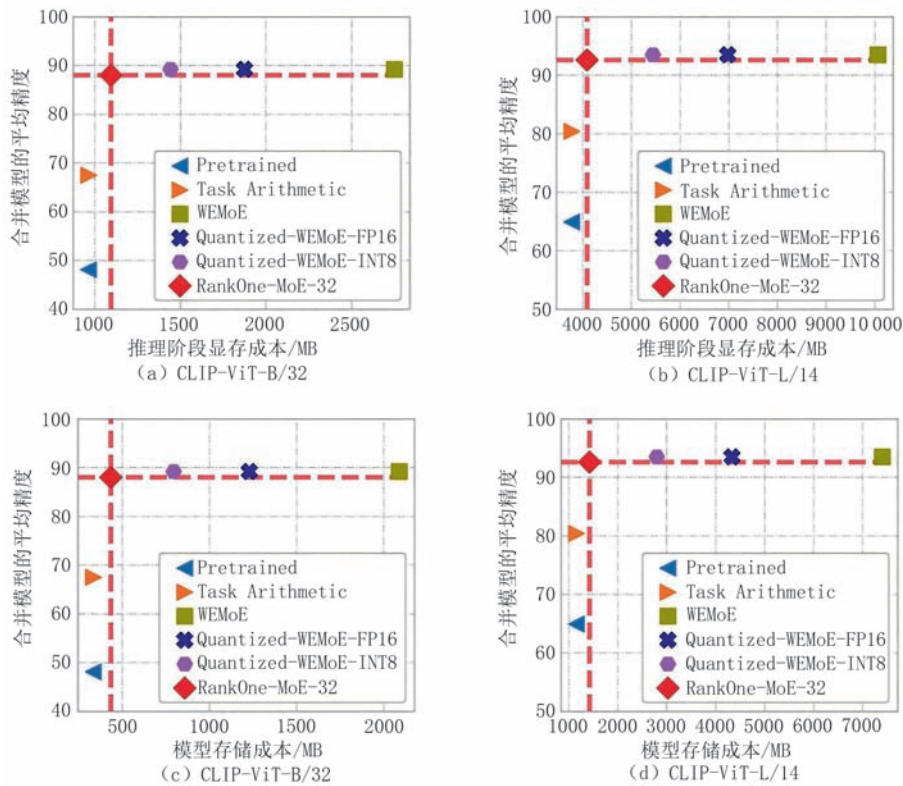


图5 不同模型合并方法在 CLIP-ViT-B/32 和 CLIP-ViT-L/14 架构上合并八个任务时的资源开销与精度的综合对比。(a)与(b)表示在推理阶段显存消耗(横轴)与合并精度(纵轴)之间的关系;(c)与(d)表示合并模型存储开销(横轴)与合并精度(纵轴)之间的关系。图中点位越接近左上角表示性能越高且资源开销越小,也即更接近理想的解。

(3)在存储成本方面,与推理显存开销的趋势相同:WEMoE 的存储成本最高,量化后的 WEMoE-FP16 与 WEMoE-INT8 通过降低数据精度显著节省了存储空间,而 RankOne-MoE 则能与静态合并基线 Task Arithmetic 更接近。例如,在 CLIP-ViT-L/14 上,WEMoE、Quantized-WEMoE-FP16、Quantized-WEMoE-INT8、RankOne-MoE、Task Arithmetic 的存储成本依次为 7403.52 MB、4331.52 MB、2795.52 MB、1423.36 MB、1157.12 MB(参见图 5(d))。

从上述结果可见,量化版本的 WEMoE 确实显著降低了原始 WEMoE 的推理显存与模型存储开销。然而,量化在实际应用中往往需要特定硬件的支持来真正实现加速,特别是 INT8 量化需要相应

的库(如 TensorRT、bitsandbytes 等)。相较之下,本文提出的 RankOne-MoE 不依赖对数据类型的改变,因此对硬件的特殊支持要求更低。整体而言,量化版本的 WEMoE 与 RankOne-MoE 在特性上可形成互补:若硬件环境支持量化,可考虑采用前者以减少存储;若硬件环境对量化支持不足,RankOne-MoE 依旧具备有效的资源降低与精度保持能力。这两种方法的使用场景差异进一步说明本文方法具有实际应用价值。

5.3.5 方法稳定性分析

在主要实验中,为了保证结果的公平性,我们固定所有方法在相同的实验设置下执行了单轮测试,并报告其结果。为进一步增强结果的可信度和统计意义,我们针对本文提出的 RankOne-MoE 方法,在

三个不同架构(ViT-B/32、ViT-B/16和ViT-L/14)上,对每个实验进行了10次独立执行。每次执行随机初始化相关权重或数据顺序,以减少实验中的随

机性对结果的影响。实验结果如表10所示。我们观察到,十次独立执行的结果表现出较小的标准差,表明方法的性能具有显著的稳定性。

表10 提出RankOne-MoE方法在独立执行十次时多任务合并模型性能对比

实验ID	1	2	3	4	5	6	7	8	9	10	均值	标准差
ViT-B/32	88.07	87.91	87.71	87.90	87.66	87.51	87.90	87.75	87.86	87.70	87.79	0.152
ViT-B/16	90.46	90.40	90.42	90.42	90.44	90.44	90.40	90.58	90.44	90.58	90.45	0.063
ViT-L/14	92.85	93.00	92.90	92.72	92.77	92.94	92.79	92.89	92.76	92.80	92.84	0.084

5.3.6 超参数分析

本文的RankOne-MoE方法引入了两个关键超参数:每个任务向专家池中添加的秩一专家数量 k 和推理时从秩一专家池中选择的比例 p 。如实现细节中所述,我们设置 k 的取值范围为 $[16, 32, 64, 128, 256, 512]$, p 的取值范围为 $[25\%, 50\%, 75\%, 100\%]$ 。图6展示了在不同模型架构下,结合不同 (k, p) 超参数组合时模型合并的性能表现。基于实验结果,我们有以下关键观察:

(1)秩一专家数量 k 的影响:更大的秩一专家数量 k 通常对应更强的合并性能。这是符合直觉的,因为每个秩一专家提取了原始MLP层的信息,增加秩一专家的数量能够更全面地保留任务相关特性,从而提高模型的合并性能。然而,过大的 k 也会导致参数数量的增加,从而提升存储和计算成本,需要在性能与资源之间找到平衡。

(2)专家选择比例 p 的影响:更大的专家选择比

例 p 通常会带来更好的性能表现。每个秩一专家在特定参数子空间中具有特定的处理能力,选择更多的秩一专家能够在推理过程中整合更多的信息,从而获得综合能力更强的合并模型。然而,选择 $p=100\%$ 并不总是最优,尤其是在ViT-L/14上,我们观察到某些设置下性能反而下降。这可能是因为在不是所有任务的秩一专家对其他任务都有积极贡献,不相关或冲突的专家可能会引入干扰,降低模型的整体表现。

(3)超参数的鲁棒性和灵活性:总体来看,RankOne-MoE对 k 和 p 均表现出较强的鲁棒性。在性能和参数数量的平衡方面,我们的默认设置为 $k=32$ 和 $p=75\%$ 。这一配置能够在大多数场景下实现良好的性能和效率。如果任务对性能要求更高,可以适当增加 k ;反之,如果需要进一步减少存储需求,可以选择较小的 k 。RankOne-MoE的灵活设计使其在性能与资源效率之间提供了广泛的调节空间。

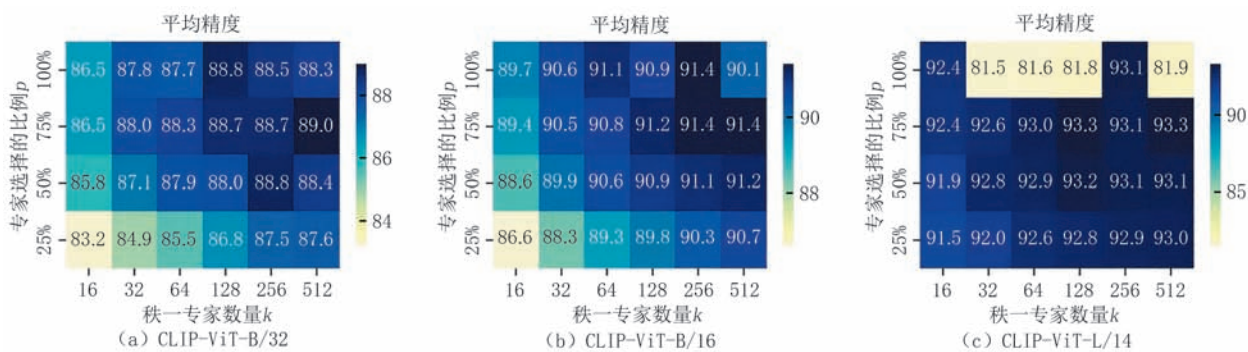


图6 不同的秩一专家数量 k 和专家选择比例 p 下合并模型的平均精度对比

综上所述,RankOne-MoE在超参数设置上具有较强的可控性和适应性,能够针对不同场景和需求提供高效的解决方案。这种灵活性进一步表明了其在多任务模型合并中的实用性和广泛适用性。

5.3.7 路由分析

为了探究不同Transformer块中八个任务在秩一专家选择上的差异,我们从ViT-B/32模型中选取

了三个具有代表性的块:靠近输入层的第 $l=1$ 块,靠近输出层的第 $l=9$ 块,以及中间的第 $l=5$ 块。为便于可视化分析,我们将超参数设置为 $k=16, p=100\%$ 。随后,我们收集了每个任务在这三个块中对128个秩一专家(8个任务,每个任务添加16个秩一专家)的权重分配 α 。对每个任务,我们对所有样本的权重进行平均,并进行可视化展

示。需要注意的是,秩一专家的序号按照任务顺序排列,例如在图7中,前16个专家由SUN397数据集添加,接下来的16个专家由CARS数据集添加,最后16个专家由DTD数据集添加。

如图7所示,我们可以观察到以下有趣的现象:

(1)浅层的均匀选择:在靠近输入的浅层块中,所有任务对秩一专家的选择相对均匀,图中颜色接近且整体较浅。这表明浅层块更倾向于利用预训练模型的共享特征,而不是任务特定的秩一专家。这一现象与深度学习模型的特性一致,即浅层通常处理粗粒度的图像特征,如边缘、纹理等,与具体任务无关。(2)深层的任务特异性选择:在靠近输出的深层块中,所有任务对自身任务添加的秩一专家表现

出更强的偏好,图中对角线上的颜色明显更深。这表明深层块更加依赖任务特定的秩一专家,用于处理与下游任务密切相关的细粒度特征。(3)中间层的过渡特性:中间层块的权重分布介于浅层和深层之间,既有一定程度的任务共享,又开始体现出对任务特定特征的选择倾向。

上述现象启发我们在未来研究中可以进一步优化静态与动态合并的策略。例如,可以考虑对浅层的MLP层采用静态合并策略,仅在深层MLP层中进行动态合并。这种层次化的合并策略能够进一步减少计算和存储成本,同时保留任务特定知识的动态适应性。这样的改进可能为多任务模型合并带来更高的参数效率。

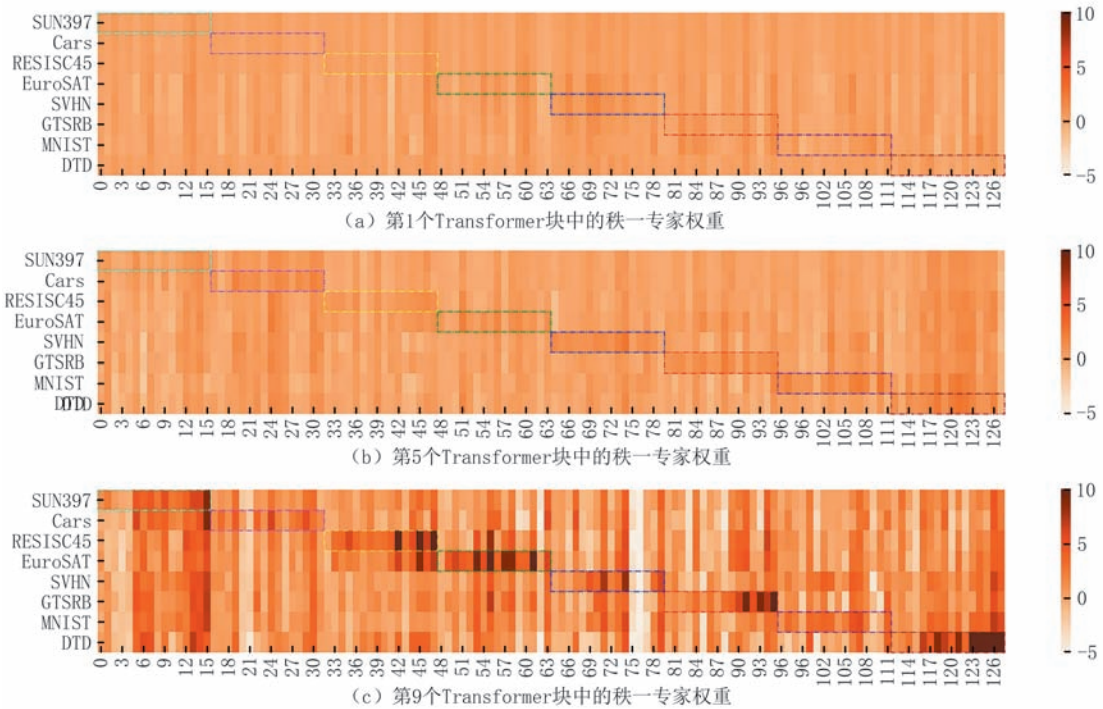


图7 不同的Transformer块中专家路由的选择情况可视化(基于ViT-B/32架构,秩一专家数量 $k=16$,秩一专家选择比例 $p=100\%$),横轴表示秩一专家的序号,纵轴表示任务/数据集

6 总结及未来展望

在本文中,我们提出了一种高效的多任务模型合并方法,称为秩一专家混合(RankOne-MoE)。该方法通过结合静态合并和动态合并的优点,在性能与参数效率之间实现了良好的平衡。具体而言,我们首先分析了待合并模型中不同参数模块与下游任务的相关性,并将参数划分为两组:低相关组采用静态合并以减少存储成本,高相关组通过动态合并保

留任务特定特性。随后,我们对高相关组的参数进行奇异值分解(SVD),提取一系列秩一专家,用于构建跨任务共享的秩一专家池。最后,通过动态路由机制,根据输入实例选择最相关的秩一专家进行合并。实验表明,我们的RankOne-MoE方法在多任务学习场景中显著优于传统静态方法,并以远低于动态方法的参数量实现了接近动态方法的性能,为多任务模型合并提供了一种高效、灵活的解决方案。

我们的工作仍有许多可扩展性值得进一步研究,包括:(1)探索RankOne-MoE在其他领域的潜

力,例如大语言模型(LLMs)的参数合并、多模态模型的跨模态合并,以及更多复杂场景下的多任务学习。(2)将路由机制进一步优化为一种无需微调(fine-tuning-free)的方法,从而避免当前需要无标签数据进行无监督微调的限制,提升其部署效率。(3)引入更高级的模型压缩技术,例如对合并后的模型进一步执行稀疏化策略、低秩分解或低比特量化方法,以进一步减少合并模型的存储和计算开销,同时保持性能。(4)现有的模型合并方法主要用于全参数微调的专家模型,未来探索提出的模型合并技术如何进一步与高效微调的专家模型结合也是一个有趣的方向。

参 考 文 献

- [1] Caruana R. Multitask learning. *Machine learning*, 1997, 28(1): 41-75
- [2] Zhang Yu, Liu Jian-Wei, Zuo Xin, learningMultitask. *Chinese Journal of Computers*, 2020, 43(7): 1340-1378 (in Chinese)
(张钰, 刘建伟, 左信. 多任务学习. *计算机学报*, 2020, 43(7): 1340-1378)
- [3] Dietterich T G, et al. Ensemble learning. *The Handbook of Brain Theory and Neural Networks*, 2002, 2(1): 110-125
- [4] Chen Z, Badrinarayanan V, Lee C Y, et al. Gradnorm: gradient normalization for adaptive loss balancing in deep multitask networks//*Proceedings of the 35th International Conference on Machine Learning*. Stockholm, Sweden, 2018: 794-803
- [5] Liu S, Johns E, Davison A J. End-to-end multi-task learning with attention//*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. California, USA, 2019: 1871-1880
- [6] Sun X, Panda R, Feris R, et al. Adashare: Learning what to share for efficient deep multi-task learning//*Proceedings of the Advances in Neural Information Processing Systems*. Virtual, 2020: 8728-8740
- [7] Shen Zhen, Cui Chao-Ran, Dong Gui-Xin, et al. Research on joint prediction of image aesthetics and emotion based on deep multi task learning. *Journal of Software*, 2023, 34(5): 2494-2506 (in Chinese)
(申朕, 崔超然, 董桂鑫, 等. 基于深度多任务学习的图像美感与情感联合预测研究. *软件学报*, 2023, 34(5): 2494-2506)
- [8] Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning//*Proceedings of the 25th International Conference on Machine Learning*. Helsinki, Finland, 2008: 160-167
- [9] Dong D, Wu H, He W, et al. Multi-task learning for multiple language translation//*Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. Beijing, China, 2015: 1723-1732
- [10] Liu Si-Jin, Zhu Xiao-Fei, Peng Zhan-Wang. Joint multi-task learning for dialogue emotion classification and behavior recognition. *Journal of Software*, 2023, 46(9): 1947-1960 (in Chinese)
(刘思进, 朱小飞, 彭展望. 联合多任务学习的对话情感分类和行为识别. *计算机学报*, 2023, 46(9): 1947-1960)
- [11] Kang Xiao-Mian, Zong Chen-Qing. Neural machine translation based on multi-task learning of discourse structure. *Journal of Software*, 2022, 33(10): 3806-3818 (in Chinese)
(亢晓勉, 宗成庆. 基于篇章结构多任务学习的神经机器翻译. *软件学报*, 2022, 33(10): 3806-3818)
- [12] Qin Chen-Guang, Wang Hai, Ren Jie, et al. Dialect language recognition based on multi-task learning. *Journal of Computer Research and Development*, 2019, 56(12): 2632-2640 (in Chinese)
(秦晨光, 王海, 任杰, 等. 基于多任务学习的方言语种识别. *计算机研究与发展*, 2019, 56(12): 2632-2640)
- [13] Ma J, Zhao Z, Yi X, et al. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts//*Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. London, UK, 2018: 1930-1939
- [14] Tang H, L J, Zhao M, et al. Progressive layered extraction (ple): a novel multi-task learning (mtl) model for personalized recommendations//*Proceedings of the 14th ACM Conference on Recommender Systems*. Virtual, 2020: 269-278
- [15] Yang E, Pan J, Wang X, et al. Adatask: a task-aware adaptive learning rate approach to multi-task learning//*Proceedings of the AAAI Conference on Artificial Intelligence*. Washington, USA, 2023, 37(9): 10745-10753
- [16] He Y, Feng X, Cheng C, et al. Metabalance: improving multi-task recommendations via adapting gradient magnitudes of auxiliary tasks//*Proceedings of the ACM Web Conference*. Lyon, France, 2022: 2205-2215
- [17] Ilharco G, Ribeiro M T, Wortsman M, et al. Editing models with task arithmetic//*Proceedings of the Eleventh International Conference on Learning Representations*. Kigali, Rwanda, 2023
- [18] Ortiz-Jimenez G, Favero A, Frossard P. Task arithmetic in the tangent space: Improved editing of pre-trained models//*Proceedings of the Advances in Neural Information Processing Systems*. New Orleans, USA, 2023: 66727-66754
- [19] Yadav P, Tam D, Choshen L, et al. Ties-merging: resolving interference when merging models//*Proceedings of the Advances in Neural Information Processing Systems*. New Orleans, USA, 2023: 7093-7115
- [20] Yang E, Wang Z, Shen L, et al. Adamerging: adaptive model merging for multi-task learning//*Proceedings of the Twelfth International Conference on Learning Representations*. Vienna, Austria, 2024: 1-21
- [21] Yang E, Shen L, Guo G, et al. Model merging in llms, mllms, and beyond: methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*, 2024
- [22] Stoica G, Bolya D, Bjorner J B, et al. Zipit! merging models from different tasks without training//*Proceedings of the*

- Twelfth International Conference on Learning Representations. Vienna, Austria, 2024:1-23
- [23] Tang A, Shen L, Luo Y, et al. Merging multi-task models via weight-ensembling mixture of experts//Proceedings of the 41st International Conference on Machine Learning. Vienna, Austria, 2024: 47778-47799
- [24] Ramé A, Vieillard N, Hussenot L, et al. WARM: on the benefits of weight averaged reward models//Proceedings of the 41st International Conference on Machine Learning. Vienna, Austria, 2024: 42048-42073
- [25] Ramé A, Ahuja K, Zhang J, et al. Model ratatouille: recycling diverse models for out-of-distribution generalization//Proceedings of the International Conference on Machine Learning. Hawaii, USA, 2023:28656-28679
- [26] Huang C, Liu Q, Lin B Y, et al. Lorahub: efficient cross-task generalization via dynamic lora composition. arXiv preprint arXiv:2307.13269, 2023
- [27] Ostapenko O, Su Z, Ponti E M, et al. Towards modular llms by building and reusing a library of loras//Proceedings of the 41-th International Conference on Machine Learning. Vienna, Austria, 2024: 38885-38904
- [28] Yu L, Yu B, Yu H, et al. Language models are super mario: absorbing abilities from homologous models as a free lunch//Proceedings of the 41-th International Conference on Machine Learning. Vienna, Austria, 2024: 57755-57775
- [29] Zhao Z, Shen T, Zhu D, et al. Merging loras like playing lego: pushing the modularity of lora to extremes through rank-wise clustering. arXiv preprint arXiv:2409.16167, 2024
- [30] Yadav P, Raffel C, Muqeeth M, et al. A survey on model moering: recycling and routing among specialized experts for collaborative learning. arXiv preprint arXiv:2408.07057, 2024
- [31] Zheng H, Shen L, Tang A, et al. Learn from model beyond fine-tuning: A survey. arXiv preprint arXiv:2310.08184, 2023
- [32] Wortsman M, Ilharco G, Gadre S Y, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time//Proceedings of the International Conference on Machine Learning. Baltimore, USA, 2022: 23965-23998
- [33] Zhang J, Chen S, Liu J, et al. Composing parameter-efficient modules with arithmetic operations//Proceedings of the Advances in Neural Information Processing Systems. New Orleans, USA, 2023: 12589-12610
- [34] Matena M S, Raffel C A. Merging models with fisher-weighted averaging//Proceedings of the Advances in Neural Information Processing Systems. New Orleans, USA, 2022: 17703-17716
- [35] Jin X, Ren X, Preotiuc-pietro D, et al. Dataless knowledge fusion by merging weights of language models//Proceedings of the Eleventh International Conference on Learning Representations. Kigali, Rwanda, 2023:1-19
- [36] Tang A, Shen L, Luo Y, et al. Concrete subspace learning based interference elimination for multi-task model fusion. arXiv preprint arXiv:2312.06173, 2023
- [37] Dosovitskiy A, Beyler L, Kolesnikov A, et al. An image is worth 16x16 words: transformers for image recognition at scale//Proceedings of the International Conference on Learning Representations. Virtual 2021:1-21
- [38] Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision//Proceedings of the International Conference on Machine Learning. Virtual, 2021: 8748-8763
- [39] Singh S P, Jaggi M. Model fusion via optimal transport//Proceedings of the Advances in Neural Information Processing Systems. Virtual, 2020: 22045-22055
- [40] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data//Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Fort Lauderdale, USA, 2017: 1273-1282
- [41] Fisher R A. On the mathematical foundations of theoretical statistics. Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 1922, 222(594-604): 309-368
- [42] Goddard C, Siriwardhana S, Ehghaghi M, et al. Arcee's mergekit: a toolkit for merging large language models//Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track. Florida, USA, 2024: 477-485
- [43] Tam D, Bansal M, Raffel C. Merging by matching models in task subspaces. arXiv preprint arXiv: 2312.04339, 2023
- [44] Zhou Y, Song L, Wang B, et al. Metagpt: merging large language models using model exclusive task arithmetic//Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. Florida, USA, 2024: 1711-1724
- [45] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958
- [46] Du G, Lee J, Li J, et al. Parameter competition balancing for model merging//Proceedings of the Advances in Neural Information Processing Systems. Vancouver, Canada, 2024:1-31
- [47] Davari M R, Belilovsky E. Model breadcrumbs: scaling multi-task model merging with sparse masks//Proceedings of the European Conference on Computer Vision. Milan, Italy, 2024: 270-287
- [48] Lu Z, Fan C, Wei W, et al. Twin-merging: dynamic integration of modular expertise in model merging//Proceedings of the Advances in Neural Information Processing Systems. Vancouver, Canada, 2024: 78905-78935
- [49] Houshy N, Giurugi A, Jastrzebski S, et al. Parameter-efficient transfer learning for NLP//Proceedings of the International Conference on Machine Learning. California, USA, 2019: 2790-2799
- [50] Hu E J, Shen Y, Wallis P, et al. Lora: low-rank adaptation of large language models//Proceedings of the Tenth International Conference on Learning Representations. Virtual, 2021:1-13
- [51] Lester B, Al-Rfou R, Constant N. The Power of scale for parameter-efficient prompt tuning//Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics. Virtual, 2021: 3045--3059

- [52] Jia M, Tang L, Chen B C, et al. Visual prompt tuning// Proceedings of the European Conference on Computer Vision. Springer, 2022: 709-727
- [53] Zhao W X, Zhou K, Li J, et al. A survey of large language models. arXiv preprint arXiv:2303.18223, 2023, 1(2)
- [54] Wei J, Bosma M, Zhao V, et al. Finetuned language models are zero-shot learners//Proceedings of the International Conference on Learning Representations. Virtual, 2022
- [55] Han Z, Gao C, Liu J, et al. Parameter-efficient fine-tuning for large models: a comprehensive survey. arXiv preprint arXiv:2403.14608, 2024
- [56] Cheng Y, Wang D, Zhou P, et al. A survey of model compression and acceleration for deep neural networks. arXiv preprint arXiv:1710.09282, 2017
- [57] Choudhary T, Mishra V, Goswami A, et al. A comprehensive survey on model compression and acceleration. Artificial Intelligence Review, 2020, 53(7): 5113-5155
- [58] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need//Proceedings of the Advances in Neural Information Processing Systems. California, USA, 2017: 5998--6008
- [59] Tang A, Shen L, Luo Y, et al. Fusionbench: a comprehensive benchmark of deep model fusion. arXiv preprint arXiv:2406.03280, 2024
- [60] Xiao J, Ehinger K A, Hays J, et al. Sun database: exploring a large collection of scene categories. International Journal of Computer Vision, 2016, 119(1): 3-22
- [61] Krause J, Stark M, Deng J, et al. 3d object representations for fine-grained categorization//Proceedings of the IEEE International Conference on Computer Vision workshops. Sydney, Australia, 2013: 554-561
- [62] Cheng G, Han J, Lu X. Remote sensing image scene classification: benchmark and state of the art. Proceedings of the IEEE, 2017, 105(10): 1865-1883
- [63] Helber P, Bischke B, Dengel A, et al. Eurosat: a novel dataset and deep learning benchmark for land use and land cover classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2019, 12(7): 2217-2226
- [64] Netzer Y, Wang T, Coates A, et al. Reading digits in natural images with unsupervised feature learning. NIPS workshop on deep learning and unsupervised feature learning. 2011, 2011(2): 4
- [65] Stallkamp J, Schlipsing M, Salmen J, et al. The German traffic sign recognition benchmark: a multi-class classification competition//Proceedings of the 2011 International Joint Conference on Neural Networks. California, USA, 2011: 1453-1460
- [66] Lecun Y. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998
- [67] Cimpoi M, Maji S, Kokkinos I, et al. Describing textures in the wild//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Columbus, USA, 2014: 3606-3613
- [68] Wang K, Dimitriadis N, Ortiz-Jiménez G, et al. Localizing task information for improved model merging and compression// Proceedings of the 41st International Conference on Machine Learning. Vienna, Austria, 2024: 50268-50287
- [69] Kingma D P, Ba J. Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014



YANG En-Neng, Ph. D. candidate.

His research interests include machine learning and recommendation systems.

TANG An-Ke, Ph. D. candidate.

His research interests include transfer learning and multi-task learning.

GUO Gui-Bing, Ph. D., professor. His research interests include recommender system and natural language processing.

JIANG Lin-Ying, M. S., associate professor. Her

research interests include natural language processing and artificial intelligence.

SUN Fu-Hui, Ph. D. Her research interest is trusted data management.

WANG Xiao-Yan, Ph. D., senior engineer. Her research interests are data management and block chain.

SHEN Li, Ph. D., associate professor. His research interests include artificial intelligence, deep learning, and reinforcement learning.

Background

The main research background of this paper lies in multi-task learning within the field of artificial intelligence, specifically addressing the critical challenges of multi-task model merging. Existing model merging methods can be categorized into static merging and dynamic merging. While static merging achieves high parameter efficiency, its performance is often limited. On the other hand, dynamic merging delivers superior performance but comes with significant resource consumption. Striking

a balance between performance and efficiency remains challenging, particularly in scenarios involving a large number of tasks or resource constraints. To address this, this paper proposes an innovative method called RankOne-MoE, which combines the strengths of static and dynamic merging to achieve an optimal trade-off between performance and resource efficiency.

RankOne-MoE leverages task-specific analysis of parameter correlations, dividing model parameters into low-correlation and

high-correlation groups. It applies static merging to the low-correlation group and dynamic merging to the high-correlation group. For the high-correlation group, rank-one experts are extracted via singular value decomposition (SVD) and stored in a shared rank-one expert pool. The most relevant experts are dynamically selected based on input instances using a dynamic routing mechanism. Experimental results demonstrate that RankOne-MoE achieves multi-task accuracy comparable to state-of-the-art dynamic methods, while significantly reducing parameter requirements. For instance, on ViT-B/32, RankOne-MoE reduces parameter usage by approximately 81.45%, showcasing outstanding efficiency.

This research is of substantial significance. It enhances both

the performance and efficiency of multi-task learning while reducing dependence on extensive data collection and storage, thereby mitigating risks of data privacy leakage. Furthermore, RankOne-MoE opens up new directions for model merging in areas such as large language models and multimodal models.

This work was supported by the National Natural Science Foundation of China (Grant No. 62032013), the Science and Technology Projects in Liaoning Province (Grant No. 2023JH3/10200005), and the Fundamental Research Funds for the Central Universities (Grant No. N2317002), and the China Scholarship Council (CSC). These projects provide the basis for the method motivation and computing resources of this project.