DOI:10.16356/j.2097-6771.2026.01.021

面向轻量化神经网络分类的微调方法

徐帅杰,王士同
(江南大学人工智能与计算机学院,无锡 214122)

摘要: 轻量化神经网络是指通过优化,减少资源消耗,使其能够在资源受限的环境中高效运行的神经网络。其训练过程通常以整体最优为目标,然而在实际应用中,可能存在某些感兴趣类别的分类精度偏低的问题,这些类别对于用户或应用而言,其准确性比其他类更重要。为解决上述问题,提出了一种适用于轻量化神经网络的结构微调方法——基于次小值阈值选取的突触连接方法(Synaptic join method based on the sub-minimum value threshold, SMVT-SJ)。该方法通过次小值选取策略划定新突触的权值阈值,从隐藏层向输出层目标神经元跨层添加新突触,从而特异性地提升用户关注类别的分类精度。为了筛选更高效的新突触,SMVT-SJ提出突触评估过程,根据所有可能的适当权值的分布来评估每个候选突触的性能。在多个不同数据集上的实验结果表明,该方法能够有效地提高特定目标类别的分类精度,并维持总体精度不发生明显降低,具有很好的泛化性和鲁棒性。

关键词: 突触连接;网络结构微调;分类;轻量化神经网络;突触评估;机器学习
中图分类号: TP391 **文献标志码:** A **文章编号:** 1005-2615(2026)01-0223-12

Fine-Tuning Method for Lightweight Neural Network Classification

XU Shuaijie, *WANG Shitong*
(School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China)

Abstract: Lightweight neural networks are a special type of neural networks which can operate efficiently through resource consumption optimization in resource-constrained environments. Their training process usually aims to optimize the overall performance. However, in practical applications, such a trained neural network sometimes suffers from low classification accuracy for some classes of interest, which are more important to users or applications than others. To address this problem, a structural fine-tuning method for lightweight neural networks—Synaptic join method based on the sub-minimum value threshold (SMVT-SJ) is proposed. This method defines the threshold for new synaptic weights using the second-smallest value strategy and selectively improves the classification accuracy for the target class by adding a small number of cross-layer synapses from hidden layers to the corresponding output neuron. In order to select more efficient new synapses, the SMVT-SJ method proposes a synaptic evaluation process to evaluate the performance of each candidate synapse according to the distribution of all possible appropriate weights. Experimental results on different datasets show that the proposed method can effectively enhance the classification accuracy of specific target class and maintain the overall accuracy without a significant decrease, and has good generalization and robustness.

Key words: synaptic join; network structural fine-tuning; classification; lightweight neural network; synaptic evaluation; machine learning

基金项目: 国家自然科学基金(61972181)。
收稿日期: 2025-04-01;**修订日期:** 2025-07-03
通信作者: 王士同,男,教授,博士生导师,E-mail:wxwangst@aliyun.com。

引用格式: 徐帅杰,王士同. 面向轻量化神经网络分类的微调方法[J]. 南京航空航天大学学报(自然科学版),2026,58(1):223-234. XU Shuaijie, WANG Shitong. Fine-tuning method for lightweight neural network classification[J]. Journal of Nanjing University of Aeronautics & Astronautics(Natural Science Edition),2026, 58(1):223-234.

近年来,深度神经网络凭借强大的特征提取与非线性建模能力,在计算机视觉、语音识别、自然语言处理等领域取得了显著成果[1-4]。然而,深层网络通常参数众多、计算复杂,难以满足边缘计算和嵌入式设备等资源受限环境的部署需求[5-8]。

为此,轻量化神经网络[9]应运而生。以ShuffleNet[10]等轻量化卷积神经网络为例,它们在保持较高准确率的同时,显著减少了模型的参数量和计算量[11]。这类网络已广泛应用于自然语言处理[12-13]、数据挖掘[14]和医学图像处理[15]等领域。本文聚焦前馈式轻量化神经网络结构,其在多个应用中凭借结构简洁、计算成本低等优势展现出良好的部署适应性[16]。

尽管轻量化模型整体性能良好,但其训练过程多以提升整体准确率为目标[17],常忽视不同类别间精度差异。实际应用中,部分目标类别的识别准确性尤为重要,例如医学图像诊断中"玻璃结节"这类肺癌早期信号。因此,如何在轻量化模型结构基本不变的前提下,提升特定目标类的识别能力,同时保持整体性能稳定,成为亟待解决的挑战。

围绕该问题,已有研究提出了多种策略。如在损失函数中引入类别权重以加强对特定类的关注[18];Chao等[19]通过输出校准引导模型偏向目标类别。然而这类方法多依赖训练阶段,难以适配结构受限的轻量模型。近年来,Kim等[20]提出Synaptic join方法,通过结构微调方式,从隐藏层神经元向目标类输出神经元添加特定跨层新突触,以增强目标类表现。该方法以结构干预替代训练再优化,在特定场景中表现出良好效果。然而,Synaptic join方法在应用于轻量化网络时存在两个关键问题:(1)其突触筛选阈值使用最小值策略,导致评估指标偏小,目标特异性不强,活跃突触候选空间受限;(2)其仅对部分候选突触进行评估,缺乏全局最优性,可能影响添加方案的有效性。

为解决上述问题,本文在已有的突触连接方法基础上进行改进,面向轻量化神经网络,提出了一种适用于轻量化神经网络的结构微调方法——基于次小值阈值选取的突触连接方法(Synaptic join method based on the sub-minimum value threshold,SMVT-SJ)。针对Synaptic join方法中突触筛选阈值使用最小值导致评估指标偏小、目标类特异性不强的问题,SMVT-SJ引入次小值策略设定突触阈值,在一定程度上扩大了连接筛选范围,提升了突触连接的灵活性与目标类别的响应能力;同时,对所有候选突触进行系统性评估,不再局限于部分连接方案,从而增强添加策略的全局优化效果。该方法最终在保持整体性能稳定的前提下,提升了目标类的精度表现,并进一步增强了突触结构与原模型之间的适配性。

1 相关理论基础

人工神经网络是一种基于生物神经元信息处理机制构建的非线性计算模型,其基本单元通过对输入信号加权求和、非线性变换,实现复杂数据映射[21]。其中,隐藏层在特征提取中发挥核心作用,是构建网络深度与学习能力的关键。尽管深度模型在诸多任务中展现出强大性能,其高计算代价限制了其在边缘计算等场景下的部署效率[22]。

相比之下,轻量化神经网络在兼顾准确率的同时,通过减少参数规模、计算量与功耗,实现了更高效的部署适应性。例如,MobileNetV2[23]结合深度可分卷积和ResNet[5]中的残差结构,仅用3.4M个参数和300M次计算,就能在ImageNet图像分类任务中达到Top 1的74.7%;ShuffleNetV2则进一步提升了速度与准确率平衡性[24]。

目前,轻量化神经网络的设计方法主要包括人工设计轻量级网络结构、模型压缩、剪枝、量化和知识蒸馏等[25-28]。然而,这些方法普遍缺乏对特定类别的分类性能调控,易造成目标类精度受限,尤其在不平衡或高敏感性任务中影响更为明显。

因此,本文聚焦轻量化神经网络的"结构微调"问题,即在不重构模型整体结构、不进行再训练的前提下,如何通过局部结构调整提升特定类别性能。本文通过添加少量高效突触实现目标类性能提升,为轻量化模型的精细化调控提供了可行路径。

2 基于次小值阈值选取的突触连接方法

本节介绍了一种面向轻量化神经网络的结构微调方法SMVT-SJ,其核心思想是在保持原模型整体结构与参数规模基本不变的前提下,通过在隐藏层与目标类输出神经元之间添加少量跨层新突触,定向提升目标类别的分类性能。图1直观展示了该方法在轻量化神经网络结构上"数据构造→突触评估→结构微调"的整体流程和核心原理。

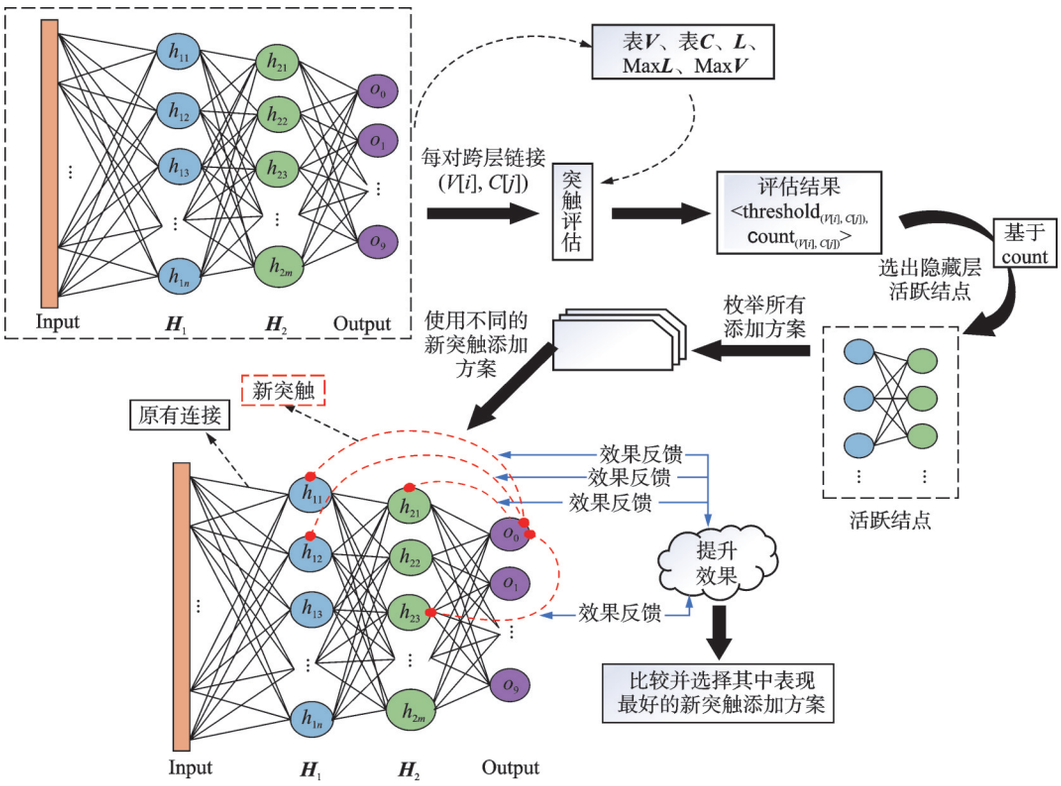

图 1　SMVT-SJ方法结构图

Fig.1　SMVT-SJ method structure diagram

2.1　突触评估所用数据构造

本节说明SMVT-SJ方法中突触评估所依赖的数据构造过程。为了评估从隐藏神经元到输出层目标类神经元的潜在高效新突触，首先需要构造两个大规模数据表项：隐藏结点表 V 和输出结点表 C。两者通过原始训练数据 D 在原神经网络上前向传播获得，如图2所示。表 V 和表 C 的列数均为训练数据样本总数 $|D|$，分别记录所有隐藏层与输出层神经元在各训练样本下的激活值。

具体来说，表 V 中第 i 行第 k 列的数值表示第 k 个训练样本在神经网络的前向传播过程中神经元 $V[i]$ 的状态值，记为 $V[i][k]$；类似地，表 C 中第 j 行第 k 列的数值表示第 k 个训练样本在神经网络的前向传播过程中输出神经元 $C[j]$ 的值，记为 $C[j][k]$。表 V 和表 C 的每一行（即每个元组）都分别代表1个隐藏神经元（记作 $V[i]$）、输出神经元（记作 $C[j]$）。以元组对（$V[i]$, $C[j]$）来表示1个跨层新突触。

此外，突触评估还用到辅助数据构造 L、Max$L$、Max$V$。L 表示训练样本的真值数组，即真实标签，Max$L$ 表示训练样本在前馈过程中预测的标签数组，Max$V$ 表示输出层神经元中的最大值所构成的数组。它们的长度均等于训练数据集的样本总数 $|D|$。图2也给出了相应真实数据的示例。

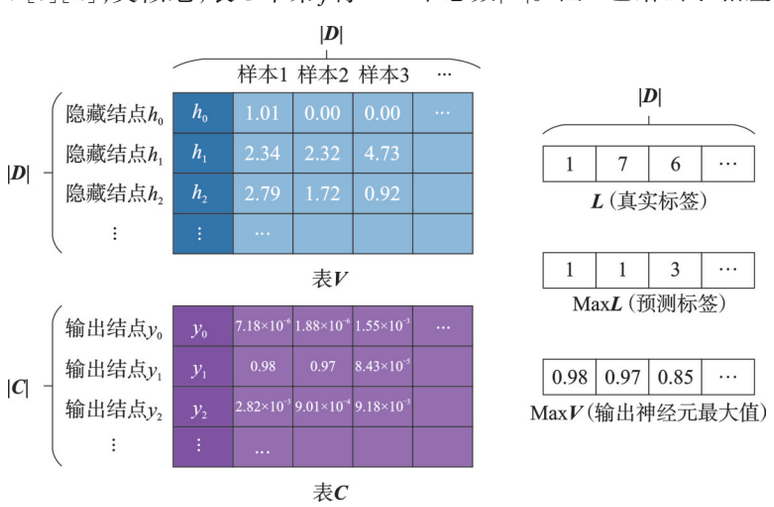

图 2　表 V、表 C 和相关辅助数据构造的示例图

Fig.2　Example diagrams of Table V, Table C and related auxiliary data construction

2.2 突触评估

在2.1节构造的V、C和L、Max$L$、Max$V$的基础上,本节提出算法1,对每个从隐藏神经元到目标类输出神经元的候选新突触($V[i]$, $C[j]$)进行评估。评估的目标是为每个候选突触计算两个指标:阈值threshold$_{(V[i],C[j])}$,刻画在不破坏非目标类预测前提下的安全权值边界;以及更正次数count$_{(V[i],C[j])}$,刻画该突触对目标类误分样本的修正能力。

算法1 SMVT-SJ方法的突触评估过程

输入:$V[i]$为训练数据样本前馈过程中隐藏层神经元的值的第i个元组;$C[j]$为训练数据前馈过程中输出神经元的值的第j个元组;L为存放真实类标签;Max$L$为存放前馈时神经网络预测输出的类标签;Max$V$为存放前馈过程中输出层神经元中的最大值;j为目标类别;ε_1、ε_2为给定极小正值。

输出:<threshold$_{(V[i],C[j])}$, count$_{(V[i],C[j])}$>:<对应突触的最合适权重边界值,目标类数据样本分类被更正次数>。

/*将训练样本分成两部分*/

(1) $\boldsymbol{D}_w \leftarrow \varnothing$, $\boldsymbol{D}_c \leftarrow \varnothing$;

(2) for $0 \leqslant k \leqslant |\boldsymbol{D}|-1$ do

(3) if $L[k]=j$ and $L[k] \neq \text{Max}L[k]$ then

(4) $\boldsymbol{D}_w \leftarrow \boldsymbol{D}_w \cup \{k\}$ /*属于目标类,但被错误分类为其他类别的样本集合*/;

(5) else if $L[k] \neq j$ and $L[k]=\text{Max}L[k]$ then

(6) $\boldsymbol{D}_c \leftarrow \boldsymbol{D}_c \cup \{k\}$ /*属于非目标类且被正确分类为对应类别的样本集合*/;

(7) end if

(8) end for

/*找到一组可以更正$\boldsymbol{D}_w$分类错误的突触权值集合$\boldsymbol{E}_w$*/

(9) $\boldsymbol{E}_w \leftarrow [\cdot]$;

(10) for each $k \in \boldsymbol{D}_w$ do

(11) $$\boldsymbol{E}_w \leftarrow \boldsymbol{E}_w \cup \left\{ \frac{\text{Max}V[k]-C[j][k]}{V[i][k]+\varepsilon_1} \right\};$$

(12) end for

/*找到一组可以保证$\boldsymbol{D}_c$分类正确性的突触权值集合$\boldsymbol{E}_c$*/

(13) $\boldsymbol{E}_c \leftarrow [\cdot]$;

(14) for each $k \in \boldsymbol{D}_c$ do

(15) $$\boldsymbol{E}_c \leftarrow \boldsymbol{E}_c \cup \left\{ \frac{\text{Max}V[k]-C[j][k]}{V[i][k]+\varepsilon_1} \right\};$$

(16) end for

/*找到能保证非目标类分类正确性的最大权值边界,即阈值threshold*/

(17) $E_{cs} \leftarrow [\cdot]$;

(18) $E_{cs} \leftarrow$ 根据E_c的元素绝对值进行升序排序;

(19) threshold$_{(V[i],C[j])} \leftarrow \text{abs}(E_{cs}[1])-\varepsilon_2$;

/*计算候选突触能够更正错误分类样本的个数,即count值*/

(20) count$_{(V[i],C[j])} \leftarrow 0$;

(21) for each weight$\in \boldsymbol{E}_w$ do

(22) if abs(weight)$\leqslant$threshold$_{(V[i],C[j])}$ then

(23) count$_{(V[i],C[j])} \leftarrow$ count$_{(V[i],C[j])}+1$;

(24) end if

(25) end for

(26) return <threshold$_{(V[i],C[j])}$, count$_{(V[i],C[j])}$>;

理想的突触应在不显著降低整体精度的前提下,提升目标类样本的分类正确率,同时尽可能不影响非目标类样本的预测结果。为此,需分别关注以下两类训练样本:(1)属于目标类j,但被错误分类的样本序号集合$\boldsymbol{D}_w$;(2)属于非目标类,且被正确分类的样本序号集合$\boldsymbol{D}_c$。

需要说明的是,$\boldsymbol{D}_w$、$\boldsymbol{D}_c$与随后定义的$\boldsymbol{E}_w$、$\boldsymbol{E}_c$在符号形式上虽保持一致,用以区分目标类与非目标类,但语义上分别表示训练样本编号集合与权值边界数值集合,两者无直接对应关系,仅为逻辑结构一致性而统一命名。

如果新突触的权值满足条件(1),则可更正$\boldsymbol{D}_w$中数据对象k的分类。因为该突触的权重乘以数据对象k在通过隐藏神经元$V[i]$时的值,能使得数据对象k在通过输出神经元$C[j]$时的值超过现有的最大输出值Max$V[k]$,从而数据对象k的分类结果就会被更正为它所属于的目标类别。对于$\boldsymbol{D}_w$中每个数据对象k,根据下列条件

$$\text{weight} \cdot V[i][k]+C[j][k] > \text{Max}V[k] \quad (1)$$

求出突触权值绝对值的下确界,并将其值存放至列表$\boldsymbol{E}_w$。

类似地,如果新突触的权值满足式(2),则可保持$\boldsymbol{D}_c$中数据对象k的正确分类。因为该突触的权重乘以数据对象k在通过隐藏神经元$V[i]$时的值能使得数据对象k在通过输出神经元$C[j]$时的值不超过现有的最大输出值Max$V[k]$,从而维持数据对象k原有的分类结果。对于$\boldsymbol{D}_c$中每个数据对象k,根据下列条件

$$\text{weight} \cdot V[i][k]+C[j][k] < \text{Max}V[k] \quad (2)$$

求出突触权值绝对值的上确界,并将其值存放至列表$\boldsymbol{E}_c$。

如果新突触的权值同时满足条件(1)和(2),那么它就既可以更正数据$\boldsymbol{D}_w$中的一些错误分类,又可以维持数据$\boldsymbol{D}_c$中的正确分类。为了使数据$\boldsymbol{D}_w$

中错误分类的更正次数进一步最大化，SMVT-SJ选择$\boldsymbol{E}_{\mathrm{c}}$中的绝对值次小值，将其划定为阈值threshold$_{(V[i],\,C[j])}$。这样$\boldsymbol{E}_{\mathrm{w}}$中几乎所有绝对值小于threshold$_{(V[i],\,C[j])}$的权值都可以同时满足式(1)和(2)，仅有极少部分仅满足式(1)。

计算出的threshold$_{(V[i],\,C[j])}$表示新突触($V[i]$，$C[j]$)的权值范围边界，该值既用于评估是否满足添加条件，同时也直接作为该突触最终的连接权值(即weight)。为避免边界干扰，实际使用中会在计算出的阈值基础上减去一个极小值。

统计$\boldsymbol{E}_{\mathrm{w}}$中在该阈值范围内的权值个数，即为count$_{(V[i],\,C[j])}$，表示模拟当前新突触($V[i]$，$C[j]$)添加后，对错误分类的训练数据样本的更正次数。需要说明的是，count并非来自训练优化过程，而是在评估阶段临时引入突触并重新前向传播后得到的结果，衡量了候选突触的直接修正能力。

因此，对于不同的候选新突触而言，较大的threshold和count，代表它在提升目标类性能方面更具特定性。

所有候选突触的评估结果各有差异，将count分布中的最大值记作max_count。对于每个候选新突触，如果其count满足count$\geqslant\frac{1}{2}$max_count，那么它被认为对于提升目标精度相对较为高效，这个突触所对应的隐藏神经元即为目标类的活跃神经元。

需要指出的是，SMVT-SJ方法所提出的“次小值阈值”策略，指的是从候选突触的权值上界集合$\boldsymbol{E}_{\mathrm{c}}$中选取绝对值次小的权值作为最终连接阈值。该策略相比Synaptic join方法中直接使用最小值阈值的方式，能在更大程度上扩展可选突触空间，提升评估灵敏度并避免非目标类特征空间的误扰，是SMVT-SJ方法中关键性的结构调控手段之一。

2.3　SMVT-SJ方法

基于所有候选新突触的突触评估结果，SMVT-SJ需要据此确定哪些突触应当被实际添加到网络中以及它们的权值。本节在此基础上给出整体的连接策略。

本文提出的SMVT-SJ方法通过将突触评估中筛选出的活跃神经元跨层连接至输出层目标类神经元，从而增强其面向目标类的表达能力。根据神经网络的普遍逼近定理[29]，这种在同一层上的神经元增加客观上会提高网络对目标类的分类准确性。因而，定性地看，该方法能够提高目标类的分类准确性。

与Kim等选取$\boldsymbol{E}_{\mathrm{c}}$内的绝对值最小值作为阈值不同，SMVT-SJ采用次小值策略，使得候选突触获得更大的count。这不仅增强了对目标类的响应能力，也提升了活跃神经元的可选空间，从而进一步提高了目标类的分类精度。

突触评估过程针对每个候选新突触($V[i]$,$C[j]$)给出评估结果：<threshold$_{(V[i],\,C[j])}$，count$_{(V[i],\,C[j])}$>。在具体选择突触添加到网络时，优先考虑具有较大count的候选新突触，当count相同时，可进一步参考threshold。

为避免过拟合，SMVT-SJ选择count最大的前n个候选新突触进行添加，而非只添加单个最优突触。为了降低同时添加多个突触的扰动，突触权值均缩放为原值的$1/n$。这是一种启发式的策略，可减轻网络整体精度的负面影响。图3展示了不同n值对目标类与总体精度的影响曲线。

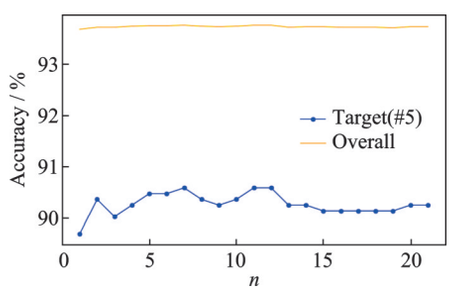

图3　目标类和总体精度随n的变化(MNIST数据集)

Fig.3　Target class and overall accuracy while varying the n value (for MNIST dataset)

由于轻量化网络的候选新突触较少，可枚举不同添加方案(即n从1到最大活跃突触数)对应的效果，选取目标类精度提升最优且整体精度下降最小的方案。已有实验表明[18]，当n的值达到某一范围时效果趋于平稳。故实际n不必过大。

此外，为进一步提升目标类精度，SMVT-SJ引入超参数scale，将新增突触的权值统一放大scale倍。图4展示了精度随scale变化的曲线，随着scale增加，目标类精度不断提高，总体精度不断降低，降低的速度先慢后快，最终趋于平稳。

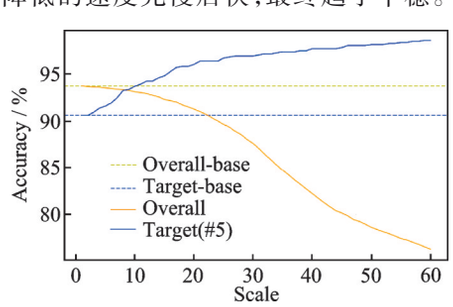

图4　目标类和总体精度随scale的变化(MNIST数据集，n=7)

Fig.4　Target class and overall accuracy while varying the scale value (for MNIST dataset, n=7)

scale值的选取不同于枚举确定的n值。SMVT-SJ以启发式方法来确定scale值:选取更大程度提升目标类精度的同时,又确保总体精度不发生明显下降的折衷scale取值。

SMVT-SJ方法首先对隐藏神经元到输出目标神经元的所有候选新突触进行评估,根据突触评估结果筛选出相对更加高效的新突触,找到其对应的活跃神经元;然后,基于筛选出的高效新突触,枚举出所有新突触添加方案,根据不同添加方案下的提升效果选取其中最好的新突触添加方案,即确定n值;最后,选取scale的折衷取值,进一步调整新突触的权值,从而实现对目标类精度的进一步提升。算法2给出了SMVT-SJ方法总体流程的伪代码。

算法2 SMVT-SJ方法总体框架

输入:数据集$\boldsymbol{D}$;测试集$\boldsymbol{D}_{\mathrm{t}}$;训练好的原神经网络$nn$;目标类别$j$。

输出:经过添加突触调整后的新神经网络nn';目标类分类精度TA;总体精度OA。

(1) 在$\boldsymbol{D}$的训练过程中,构造所需数据构造表$\boldsymbol{V}$,表$\boldsymbol{C}$,$\boldsymbol{L}$,Max$\boldsymbol{L}$,Max$\boldsymbol{V}$;

(2) count_set←∅/*用来存储每个候选突触的评估结果中count值*/;

/*评估每个候选新突触($V[i]$, $C[j]$) */

(3) for each ($V[i]$, $C[j]$) do

(4) ＜threshold$_{(V[i],\,C[j])}$, count$_{(V[i],\,C[j])}$＞←调用算法1;

(5) count_set←count_set∪count$_{(V[i],\,C[j])}$;

(6) end for

/*根据count值筛选活跃神经元*/

(7) max_count←max(count_set);

(8) active_neuron←∅;

(9) for each count$_{(V[i],\,C[j])}$∈count_set do

(10) if count$_{(V[i],\,C[j])}\geqslant\frac{1}{2}$max_count then

(11) active_neuron←active_neuron∪($V[i]$, $C[j]$);

(12) end if

(13) end for

(14) 根据count$_{(V[i],\,C[j])}$值对active_neuron降序排序;

/*枚举所有的新突触添加方案*/

(15) while 仍有未添加的新突触添加方案 do

(16) nn'←添加新突触到nn;

(17) 计算D_{t}在nn'的输出;

(18) TA,OA←根据步骤(17)的输出结果计算;

(19) end while

(20) 选用这些添加方案中表现最好的新突触添加方案;

(21) 选取scale的折中取值;

(22) return nn',TA,OA。

2.4 算法复杂度

本节对算法2(SMVT-SJ方法)各步骤的时间复杂度进行分析。

在步骤(1)中,须构造隐藏神经元状态表$\boldsymbol{V}$与输出神经元状态表$\boldsymbol{C}$,其时间复杂度为$O(|\boldsymbol{V}|+|\boldsymbol{C}|)$,其中$|\boldsymbol{V}|$为隐藏神经元数,$|\boldsymbol{C}|$为类别数。步骤(3~6)为算法的核心部分,通过对每个候选突触($V[i]$, $C[j]$)调用算法1进行评估。算法1的时间复杂度主要依赖其步骤(2~8),该部分通过遍历全部训练样本构造目标类中被错误分类样本集合$\boldsymbol{D}_{\mathrm{w}}$和非目标类中被正确分类样本集合$\boldsymbol{D}_{\mathrm{c}}$,算法1的时间复杂度为$O(|\boldsymbol{D}|)$。算法2的步骤(3~6)须对所有$|\boldsymbol{V}|\times|\boldsymbol{C}|$个候选突触调用算法1,故其总时间复杂度为$O(|\boldsymbol{V}|\times|\boldsymbol{C}|\times|\boldsymbol{D}|)$。步骤(7~13)根据突触评估结果筛选出满足条件的活跃神经元,该过程仅需一次线性扫描,时间复杂度为$O(|\boldsymbol{V}|\times|\boldsymbol{C}|)$。在步骤(14)中,需要对活跃神经元集合active_neuron按照评估指标count$_{(V[i],\,C[j])}$进行降序排序。设该集合大小为k,由于$k\leqslant|\boldsymbol{V}|\times|\boldsymbol{C}|$,排序的时间复杂度为$O(k\log k)$。由于活跃神经元的数量$k$通常远小于$|\boldsymbol{V}|\times|\boldsymbol{C}|$,因此该步骤不会显著增加整体开销。步骤(15~19)枚举所有添加方案并计算其对应的提升效果,选出最优方案,其时间复杂度与活跃神经元的数量成正比。具体而言,该过程基于步骤(14)中已按count$_{(V[i],\,C[j])}$值排序的活跃神经元集合(大小为k),依次构造从前1个到前k个新突触的添加方案,因此添加方案总数为k。该过程整体时间复杂度为$O(k)$。步骤(21)用于选取scale折衷取值的过程仅涉及常数级别的操作,其计算代价可以忽略不计。

综上,算法2的整体时间复杂度为$O(|\boldsymbol{V}|\times|\boldsymbol{C}|\times|\boldsymbol{D}|+k\log k)$,由于$k\leqslant|\boldsymbol{V}|\times|\boldsymbol{C}|$,最终可将复杂度表示为$O(|\boldsymbol{V}|\times|\boldsymbol{C}|\times|\boldsymbol{D}|)$,明显与$|\boldsymbol{V}|$、$|\boldsymbol{C}|$、$|\boldsymbol{D}|$各自成线性关系。因此SMVT-SJ方法在提升目标类样本更正次数的同时,并未显著增加运行成本,保持了较好的计算效率。

3 实验与结果分析

本节记录了不同方法在多个数据集上的实验结果。实验环境为Win11,Python版本为3.9,基于Pytorch-CPU 2.2.2的框架下实现。电脑配置为CPU为i7-12700,2.10 GHz,内存为16 GB。

3.1　数据集与网络结构

MNIST 数据集包含 60 000 张训练和 10 000 张测试样本，每张为 28 像素×28 像素的手写数字图像。样本处理时，像素值从[0, 255]缩放至[0, 1]，并归一化为均值 0.130 7、标准差 0.308 1。

Fashion-MNIST 数据集包含 10 类服装物品的灰度图像，与 MNIST 数据集的结构相似，但更复杂，也更贴近实际应用场景，其一共包括 60 000 个训练样本和 10 000 个测试样本，均为灰度图像。样本处理方式与 MNIST 相同。

SVHN 数据集由 Google 街景图像中门牌号数字构成，共 600 000 张样本，包含 73 257 张训练图像和 26 032 张测试图像，标签为 1～9 和 0（标签为 10），均值和标准差为 0.5。

EMNIST Balanced 是 MNIST 的扩展，包含英文大小写字母和数字的手写图片，总计 131 600 张图像，分为 47 个类别，训练图像每类 2 800 张，测试图像每类 400 张，均值和标准差均为 0.5。表 1 列出了各数据集的详细信息。

表 1　数据集信息

Table 1　Dataset information　　个

数据集	样本数	特征数	类别数
MNIST	70 000	784	10
Fashion-MNIST	70 000	784	10
SVHN	99 289	3 072	10
EMNIST Balanced	131 600	784	47

本实验采用人工设计的轻量化全连接神经网络，包含 3 个全连接层。输入层神经元数由图像像素和通道数决定，隐藏层神经元数量固定，输出层神经元数依数据集类别数设置。隐藏层使用 ReLU 激活函数，输出层使用 Softmax 实现归一化分类。网络结构如表 2 所示。

表 2　神经网络结构

Table 2　Neural network structure

层	类型	数量/个
1	Input	—
2	Fully-connected ReLU	100
3	Fully-connected ReLU	64
4	Output Softmax	—

3.2　实验设置

本节将本文提出的 SMVT-SJ 方法与 Kim 等[20]提出的 Synaptic join（SJ）方法、模型原始精度进行对比。模型均使用人工设计的轻量化神经网络（表 2）。为了保证不同方法和不同数据集上对比的客观性，实验选取了相同的优化器、学习率、训练批次大小及相同的损失函数。其中，优化器选择随机梯度下降法，学习率均设置为 0.001，训练批次大小设置为 64，损失函数选用交叉熵损失函数。

图 5 展示了部分数据集在迭代过程中不同类别的测试精度和总体测试精度的变化曲线。对于 MNIST 数据集，经过 32 000 次迭代后总体精度可达到 93.75%，其中类别 5 在所有类别中的分类精度最低，为 89.69%，考虑将类别 5 作为 MNIST 数据集的目标类；对于 Fashion-MNIST 数据集，经过 227 000 次迭代后总体精度可达到 87.96%，其中类别 6 在所有类别中的分类精度最低，为 70.50%，考虑将类别 6 作为 Fashion-MNIST 数据集的目标类；对于 SVHN 数据集，经过 275k 次迭代后总体精度可达到 82.44%，其中类别 8 在所有类别中的分类精度最低，为 70.78%，考虑将类别 8 作为 SVHN 数据集的目标类。对于 EMNIST Balanced 数据集，经过 468 000 次迭代后总体精度可达到 82.17%，其中类别 41 在所有类别中的分类精度最低，为 50.50%，考虑将类别 41 作为 EMNIST Balanced 数据集的目标类。

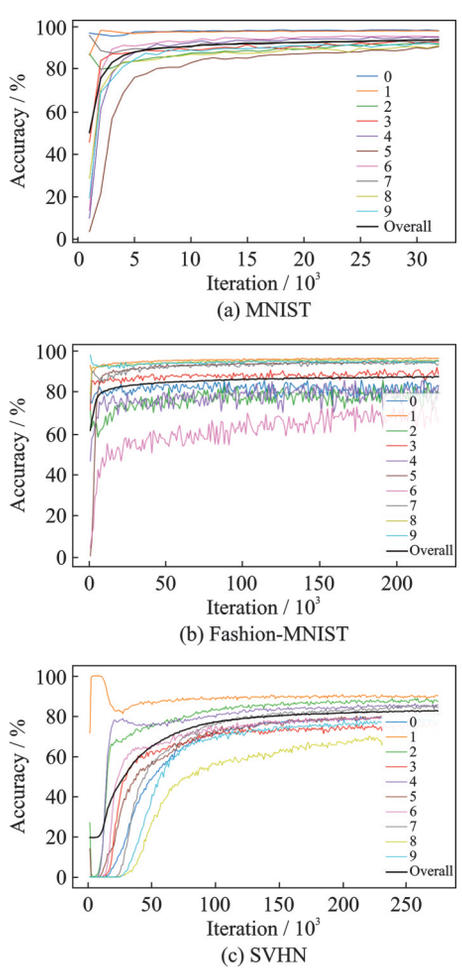

图 5　不同数据集总体测试精度和每类测试精度的变化曲线

Fig.5　Change curves of overall test accuracy and each type of test accuracy on different datasets

3.3 评估指标

本文选用常见的目标类分类精度TA、总体精度OA作为主要评估指标来衡量网络结构经过方法修改或者调整后的性能。

对第j类作为目标类的分类精度是本文的首要关注点,因此目标类j的分类精度是评估方法性能的首要标准,即

$$\mathrm{TA}_j=\frac{\mathrm{TP}_j}{\mathrm{TP}_j+\mathrm{FN}_j} \tag{3}$$

式中:TP_j表示目标类j的所有样本中被正确预测的样本数量,FN_j表示属于目标类j的所有样本中被错误预测为其他类别的样本数量。

总体精度反映了模型在所有类别的样本中的预测准确性

$$\mathrm{OA}=\frac{1}{C}\sum_{j=1}^{C}\mathrm{TA}_j \tag{4}$$

式中:TA_j表示第j类被正确预测为正类的样本数量,C为类别的总数。

本文还引入方差的变体指标DOT-RV(alteRed varianc of decreased off-target class),用于量化非目标类精度下降的整体影响。该指标计算方式为:对所有精度较原始模型下降的非目标类别,计算其精度降低值的平方和,并除以这部分类别的数量。DOT-RV值越大,表示非目标类精度下降幅度越大或分布越不均衡。该指标可作为评估方法性能的第三参考维度,用以衡量在提升目标类精度的同时,模型对其他类别性能的干扰程度

$$\mathrm{DOT-RV}=\frac{1}{|S_\mathrm{d}|}\sum_{i\in S_\mathrm{d}}(p_i-o_i)^2 \tag{5}$$

式中:S_d表示非目标类中精度发生降低的类别集合,p_i表示网络结构调整后S_d中第i类的测试精度,o_i表示S_d中第i类的原始精度。

3.4 对比实验

本节重点比较本文提出的SMVT-SJ方法与SJ方法[20]在不同数据集上的表现。

表3给出各方法在不同数据集上的超参数取值;表4比较了SMVT-SJ、SJ以及原始模型(Original)在目标类分类精度TA、总体精度OA以及非目标类精度方差变体DOT-RV等方面的表现。

表3 超参数取值及活跃结点数量

Table 3 Hyperparameter values and number of active neurons

数据集	SMVT-SJ		SJ	
	n	scale	n	scale
MNIST	7	4	2	4
Fashion-MNIST	8	4	2	12
SVHN	4	4	1	2
EMNIST Balanced	7	2	2	2

实验结果表明,SMVT-SJ在多个数据集上整体优于SJ方法。从表4可知,相比原始模型,SMVT-SJ在MNIST、Fashion-MNIST、SVHN和EMNIST Balanced这4个数据集上的提升幅度分别为1.68%、1.10%、1.75%和2.00%,均明显高于SJ方法对应的1.34%、0.60%、0.91%和1.00%。换言之,在4个数据集上SMVT-SJ相比SJ额外获得了0.34%~1.00%的目标类精度增益,其中在EMNIST Balanced上的提升幅度约为SJ的两倍。与此同时,两种方法的总体精度均保持在接近原始模型的水平,说明在整体性能不显著下降的前提下,SMVT-SJ能够在各数据集上持续取得比SJ更大的目标类提升。

DOT-RV衡量非目标类的精度变化整体情况。较小的DOT-RV表示对非目标类的影响更小、性能更稳定。结果显示,SMVT-SJ在MNIST和Fashion-MNIST上均具备更低的DOT-RV,在SVHN上略高则与其更显著的目标类提升有关。

表5以Fashion-MNIST为例,详细展示了SMVT-SJ和原始模型在各类别和总体精度上的比较。可以看到,SMVT-SJ在提升目标类精度的

表4 不同方法在不同数据集上的实验结果

Table 4 Experimental results of different methods on different datasets

数据集	SMVT-SJ			SJ			Original		
	TA/%	OA/%	DOT-RV	TA/%	OA/%	DOT-RV	TA/%	OA/%	DOT-RV
MNIST	91.37	93.62	2.433×10^{-5}	91.03	93.61	2.490×10^{-5}	89.69	93.75	—
Fashion-MNIST	71.60	87.97	5.200×10^{-6}	71.10	87.90	9.911×10^{-3}	70.50	87.96	—
SVHN	72.53	82.44	3.785×10^{-6}	71.69	82.45	7.149×10^{-7}	70.78	82.44	—
EMNIST Balanced	53.50	82.11	5.490×10^{-5}	51.50	82.09	5.470×10^{-5}	50.50	82.17	—

表5 每个类别精度及总体精度(SMVT-SJ对于Fashion-MNIST数据集)

Table 5 Accuracy of each category and overall accuracy (SMVT-SJ for Fashion-MNIST dataset) %

方法	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	OA
Original	81.80	96.90	79.50	90.00	80.50	94.40	70.50	95.10	95.70	95.20	87.96
SMVT-SJ	81.40	96.90	79.40	89.80	80.40	94.40	71.60	95.10	95.50	95.20	87.97

同时，对非目标类影响较小，验证其对整体性能的保护能力。

表 6 展示了在不同数据集上，不同方法的活跃神经元数量（num）和部分候选新突触的前若干大值（count）。SMVT-SJ 在各数据集上分别可筛选出 21、13、27 和 15 个活跃神经元，明显多于 SJ 方法（19、2、6、4）；同时，其筛选出的候选突触具有更高的 count 值，例如在 MNIST 上最大为 14，而 SJ 仅为 7，这意味着 SMVT-SJ 更正了更多被错误分类的目标类别样本，进一步提高了目标类的分类精度。

表 6　活跃神经元数量和部分候选新突触的 count 值

Table 6　Number of active neurons and the count value of partial synapses

数据集	SMVT-SJ		SJ	
	num	count	num	count
MNIST	21	14,13,10,9	19	7,7,6,4
Fashion-MNIST	13	6,5,4,3	2	3,2
SVHN	27	2,2,2,1	6	1,1,1,1
EMNIST Balanced	15	4,4,4,3	4	3,2,2,2

综上，SMVT-SJ 在目标类提升、非目标类干扰控制及突触筛选效率等方面均优于 Synaptic join 方法，同时也验证了其在轻量化神经网络调控任务中的有效性与适用性。

3.5　不同网络规模下 SMVT-SJ 方法的适用性分析

本节探讨 SMVT-SJ 方法在不同规模模型上的性能，实验均基于 MNIST 数据集。将表 2 中的网络设为模型 1，引入结构如表 7 所示的模型 2，输入输出层大小均符合 MNIST 设定。

表 7　模型 2 的网络结构

Table 7　Network structure of model 2

层	类型	数量/个
1	Input	784
2	Fully-connected ReLU	128
3	Fully-connected ReLU	96
4	Fully-connected ReLU	64
5	Output Softmax	10

实验使用 ptflops 工具计算模型的单次前向传播的浮点运算量（Floating point operations，FLOPs）和参数量，表 8 展示了引入 SMVT-SJ 前后的变化。模型 1 参数量由 85 614 增至 85 621，FLOPs 由 85 778 增至 86 226；模型 2 参数量由 119 722 增至 119 723，FLOPs 由 120 010 增至 120 074。可见，SMVT-SJ 仅增加极少量跨层新突触，对参数和 FLOPs 影响极小，保持了良好的轻量化和计算效率。

需要说明的是，模型 2 参数量仅增加 1，FLOPs 增加 64，是因为该方法在评估后仅可以添加 1 个最有效的新突触。这种最小结构增量正体现了方法“低开销、高增益”的设计原则，符合轻量化微调的实际需求。

表 8 比较了 SMVT-SJ 在两模型上的性能，均显著提升目标类精度，且总体精度基本稳定。尤其在规模较大的模型 2 上，目标类精度提升同时总体精度完全保持不变，表明 SMVT-SJ 对不同规模网络均具有良好的适用性。

表 8　不同模型的效果比较（SMVT-SJ 对于 MNIST 数据集）

Table 8　Comparison of the effects of different models (SMVT-SJ for MNIST dataset)

模型	方法	TA/%	OA/%	参数数量	FLOPs
1	Original	89.69	93.75	85 614	85 778
	SMVT-SJ	91.37	93.62	85 621	86 226
2	Original	89.35	93.76	119 722	120 010
	SMVT-SJ	91.03	93.76	119 723	120 074

图 6 显示了 SMVT-SJ 前后两模型的单样本推理时间。推理时间通过 PyTorch eval 模式、无梯度上下文多次前向传播取均值得出，测试在 CPU 环境下进行且禁用并行加速。结果显示，模型 1 推理时间由 0.005 7 ms 增至 0.007 7 ms，模型 2 由 0.006 6 ms 增至 0.008 6 ms，时间增长均仅 0.002 ms，影响微乎其微。

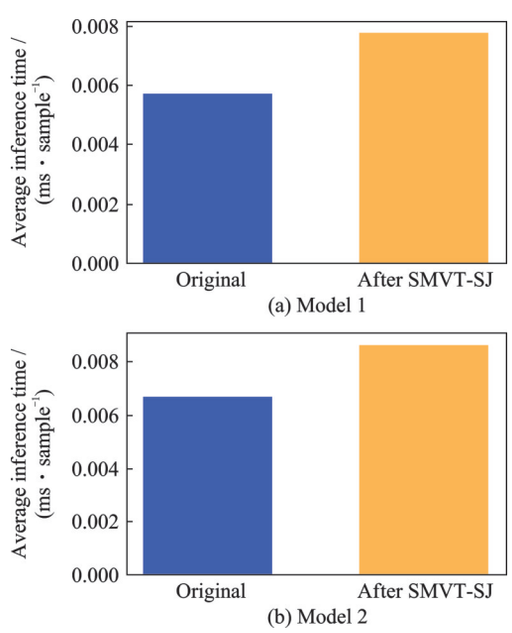

图 6　不同模型单样本推理时间对比

Fig.6　Comparison of single-sample inference time for different models

综上，SMVT-SJ 几乎不增加模型复杂度，能有效提升目标类分类精度，并保持总体精度不发生显著降低，展现出良好的轻量级特性和对不同规模网络的适用性。

3.6 不同精度目标类选取

本节分析SMVT-SJ方法针对不同原始精度类别作为目标类时的提升效果。表9列出以原始精度最低(类5,89.69%)、中等(类3,92.67%)和最高(类0,98.16%)类别为目标时的目标类及总体精度表现,原始总体精度为93.75%。结果显示,原始精度最低的类5提升最显著,达1.68%;类3和类0作为目标类时的提升分别为0.99%和0.51%。总体精度变化不大,差异主要体现在目标类精度上。

因此,SMVT-SJ方法对目标类提升效果与目标类的原始精度高低呈现一定的相关性,即原始精度越低的类别,SMVT-SJ的提升效果越明显,其原因在于原始精度较低的类别通常包含更多被错误分类的样本,SMVT-SJ方法在突触评估过程中能够从这些样本中识别出影响更大的跨层连接并进行补充,增强该类判别边界,提升识别能力。相比之下,原始精度较高的类别由于误分类样本较少、结构已较稳定,微调空间受限,故精度提升相对有限。

综上所述,SMVT-SJ方法展现出"弱类优先补偿"的特性,通过对低精度类别的针对性优化,在保持整体结构稳定的同时,实现了更显著的性能提升。这一特性使得该方法尤其适用于提升原始模型中分类效果不佳的类别识别性能。

表9 选取不同精度目标类的实验结果(SMVT-SJ对于MNIST数据集)

Table 9 Experimental results of selecting target classes with different precision levels (SMVT-SJ for MNIST dataset)

目标类别	类5(最低)	类3(中)	类0(最高)
TA_i/%	91.37	93.66	98.67
OA/%	93.62	93.68	93.62
DOT-RV	2.433×10^{-5}	3.389×10^{-6}	1.855×10^{-5}

3.7 不同的阈值选取策略对比

为验证SMVT-SJ基于次小值阈值选取策略的高效性和正确性,本节以MNIST数据集对比了次小值、第三小值和第四小值这3种策略在目标类与总体精度上的表现(表10)。

此外,为进一步分析不同阈值选取策略的稳定性差异,本节针对次小值与第三小值策略,分别进行了5次独立实验(表11),并通过配对t检验(paired t-test)进行统计显著性分析。检验结果显示p值为0.015 7,小于显著性水平0.05,表明两者在总体精度上差异显著,验证了次小值策略在保持整体性能方面的稳定性和优势。

表10 MNIST数据集上不同阈值选取策略的实验结果

Table 10 Experimental results of different threshold selection strategies on MNIST dataset %

阈值选取策略	TA	OA
次小值	91.37	93.62
第三小值	91.48	93.54
第四小值	91.82	93.56
第五小值	92.26	93.47

表11 MNIST数据集上"次小值"与"第三小值"选取策略的OA对比

Table 11 Comparison of OA values between the "second smallest value" and "third smallest value" selection strategies on MNIST dataset %

实验编号	次小值选取策略	第三小值选取策略
1	93.62	93.54
2	93.82	93.76
3	93.97	93.94
4	93.38	93.38
5	93.77	93.75

由表10表明,虽然较高阶小值策略能进一步提升目标类精度,但非目标类性能损失更大。次小值策略在提升目标类精度(提升1.68%)的同时,仅导致总体精度微降0.13%,实现了更优的整体性能平衡。原理上,阈值提升扩大了新突触对目标类的响应范围,有利于纠正误分类样本,提高目标类精度,但也易错误分类非目标类,导致总体性能下降。次小值策略通过精准控制阈值边界,有效平衡了目标类提升与非目标类扰动,避免了高阶小值策略放宽阈值所带来的特征空间扭曲。此外,不同阈值选取策略仅影响新增突触的评估排序,不改变网络结构和参数规模。SMVT-SJ仅在隐藏层与目标输出间引入极少量跨层突触,新增结构占比极小,推理复杂度无显著差异。主要差异体现在训练时的突触评估计算,部署效率影响可忽略。

综上所述,基于次小值选取策略的SMVT-SJ方法在保障整体精度的同时,可有效提升目标类性能,是相对最具平衡性的实现方式。

3.8 与剪枝方法的对比实验

为进一步验证SMVT-SJ方法在提升目标类精度方面的有效性和实用性,本文构造了两种基于权值重要性的剪枝基线方法作为对照:类权重调整剪枝(Class-weighted pruning, CW-P)和类别敏感性剪枝[30](Class-sensitive pruning, CS-P)。其中,CW-P结合类别加权损失与全局幅值剪枝:先在损失函数中对目标类赋予更高权重,对原网络进行少量轮次的微调训练,然后按照权值绝对值大小在全连接层上执行全局剪枝,仅保留幅值较大的连接;CS-P则参考Cross-layer importance评价策略[30],计算不同层连接对各类别的敏感度,并剪除对目标类贡献较小的神经元连接。两种方法均代表了基于

网络重要性分析的典型剪枝思路，用于从“模型压缩＋目标类增强”的角度，与SMVT-SJ进行对比。

在MNIST数据集和本文的轻量化神经网络（表2）上进行实验，结果如表12所示。在CW-P方法中，微调阶段将学习率设置为原学习率的0.1倍。

表12　MNIST数据集上SMVT-SJ与剪枝方法的比较

Table 12　Comparison of SMVT-SJ and pruning methods on MNIST dataset

方法	TA/%	OA/%	参数量
Original	89.69	93.75	85 614
CW-P	90.81	93.53	84 757
CS-P	89.69	93.75	83 826
SMVT-SJ	91.37	93.62	85 621

从结果可见，CS-P方法在目标类精度TA和总体精度OA方面均未提升，与原始模型保持一致，仅在某些非目标类上出现微小波动。CW-P在目标类精度上实现了小幅提升（由89.69%升至90.81%），但总体精度下降更为显著，反映出精度提升的结构不均衡性。相比之下，SMVT-SJ方法实现了所有方法中最高的目标类精度（91.37%），同时仅带来约0.1%的总体精度下降，在目标类增强与整体稳定性之间取得更优平衡。

在模型结构方面，CW-P与CS-P方法通过参数剪枝实现了不同程度的压缩，参数量分别减少了1%和2%，但前者伴随精度损失，后者则未带来明显性能提升。而SMVT-SJ方法仅通过添加7个跨层突触连接（参数量由85 614增至85 621），便在结构几乎不变的前提下实现了较为显著的目标类精度优化。

综上，剪枝方法在参数压缩方面略具优势，但可能引发精度波动或整体性能下降，而SMVT-SJ方法虽然不以压缩模型为目标，但仅通过极少结构改动，实现了更均衡且稳定的性能提升，结构几乎不变，体现出更佳的综合效果。

4　结　　论

为在资源受限的轻量化神经网络中特异性提升目标类别分类精度，本文提出了一种轻量级的结构微调方法SMVT-SJ。该方法在保持原有网络整体结构与参数规模基本不变的前提下，对隐藏层到目标输出神经元的少量跨层候选突触进行评估，并采用次小值策略确定高效连接的权值阈值，从而强化目标类的表达能力，同时兼顾轻量化特性。

SMVT-SJ方法从结构层面优化模型输出，实现特异性精度提升。相较于SJ方法[20]使用最小值阈值策略，SMVT-SJ方法采用次小值作为新突触权值的阈值边界，扩大了候选突触空间，提升了更正误分样本的能力，同时增强了对活跃神经元的识别效果。实验验证表明，次小值策略在提升目标类精度的同时有效抑制了非目标类扰动，在多个数据集上均展现出更优的精度平衡性与稳定性，且SMVT-SJ方法引入的额外参数和计算开销极小，符合轻量化微调的要求。

与传统“整体最优”训练范式不同，本文从网络结构层面出发，通过精细控制少量结构增量，引导模型在资源受限条件下对关注类别做出更符合任务需求的响应。这种面向特定类别的精度调控能力，为轻量化模型在实际部署中的差异化优化提供了新的途径，也为提升神经网络的可控性与可解释性提供了方法基础。

当然，本文研究仍存在一定局限性：(1)突触添加过程未充分引入用户的偏好或交互机制，未来可探索加入用户指定的连接约束，实现更具定制化的微调方案；(2)本文主要聚焦轻量网络，对于复杂深层结构的适配性有待进一步验证。未来可扩展至更深层次网络，结合层间关系与多跳连接机制，以进一步提升方法的通用性与适应性。

参考文献：

[1] 孙涵，刘译善，林昱涵. 基于深度学习的显著性目标检测综述[J]. 数据采集与处理，2023，38(1)：21-50. SUN Han，LIU Yishan，LIN Yuhan. Deep learning based salient object detection：A survey[J]. Journal of Data Acquisition and Processing，2023，38(1)：21-50.

[2] JAIN N，KALEV A. QuFeX：Quantum feature extraction module for hybrid quantum-classical deep neural networks[J]. Quantum Science and Technology，2024，9(3)：035017.

[3] TU Y，LIN Y. Deep neural network compression technique towards efficient digital signal modulation recognition in edge device[J]. IEEE Access，2019，7：58113-58119.

[4] SARASWAT S，GUPTA A，GUPTA H P，et al. An incremental learning based gesture recognition system for consumer devices using edge-fog computing[J]. IEEE Transactions on Consumer Electronics，2020，66(1)：51-60.

[5] HE K，ZHANG X，REN S，et al. Deep residual learning for image recognition[C]//Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.]：IEEE，2016：770-778.

[6] SIMONYAN K，ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[C]//Proceedings of the 3rd International Conference on Learning Representations (ICLR). San Diego，CA，USA：ICLR，2015：1-14.

[7] ALON G，DAR Y. How does overparameterization affect machine unlearning of deep neural networks?[C]//Proceedings of the 41st International Conference on Machine Learning (ICML). Vienna，Austria：

PMLR, 2024: 1255-1279.
[8] MINGARD C, REES H, VALLE-PÉREZ G, et al. Deep neural networks have an inbuilt Occam's razor[EB/OL]. (2025-02-10). https://www.nature.com/articles/s41467-024-54813-x.
[9] 徐光柱, 朱泽群, 尹思璐, 等. 基于轻量级深层卷积神经网络的花卉图像分类系统[J]. 数据采集与处理, 2021, 36(4): 756-768.
XU Guangzhu, ZHU Zequn, YIN Silu, et al. Flower image classification system based on lightweight DCNN[J]. Journal of Data Acquisition and Processing, 2021, 36(4): 756-768.
[10] ZHAO H, GAO Y, DENG W. Defect detection using ShuffleNet-CA-SSD lightweight network for turbine blades in IoT[J]. IEEE Internet of Things Journal, 2024, 11(20): 32804-32812.
[11] 卢宏涛, 罗沐昆. 基于深度学习的计算机视觉研究新进展[J]. 数据采集与处理, 2022, 37(2): 247-278.
LU Hongtao, LUO Mukun. Survey on new progresses of deep learning based computer vision[J]. Journal of Data Acquisition and Processing, 2022, 37(2): 247-278.
[12] CHEN B, LI P, LI B, et al. PSViT: Better vision transformer via token pooling and attention sharing [C]//Proceedings of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, TN, USA: IEEE, 2021: 10518-10527.
[13] ROHANIAN O, NOURIBORJI M, JAUNCEY H, et al. Lightweight Transformers for clinical natural language processing[J]. Natural Language Engineering, 2024, 30(5): 887-914.
[14] FENG M, ZHENG J, REN J, et al. Big data analytics and mining for effective visualization and trends forecasting of crime data[J]. IEEE Access, 2019, 7: 106111-106123.
[15] ZHAO Y, LIN L. A lightweight U-Net for medical image segmentation[C]//Proceedings of 2024 Photonics & Electromagnetics Research Symposium (PIERS). Chengdu, China: IEEE, 2024: 1-5.
[16] 葛道辉, 李洪升, 张亮, 等. 轻量级神经网络架构综述[J]. 软件学报, 2020, 31(9): 2627-2653.
GE Daohui, LI Hongsheng, ZHANG Liang, et al. Survey of lightweight neural network[J]. Journal of Software, 2020, 31(9): 2627-2653.
[17] SZEGEDY C, VANHOUCKE V, IOFFE S, et al. Rethinking the inception architecture for computer vision[C]//Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.]: IEEE, 2016: 2818-2826.
[18] FERNANDO K R M, TSOKOS C P. Dynamically weighted balanced loss: Class imbalanced learning and confidence calibration of deep neural networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(7): 2940-2951.
[19] CHAO W L, CHANGPINYO S, GONG B, et al. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild[C]//Proceedings of Computer Vision—ECCV 2016. Cham, Switzerland: Springer, 2016: 52-68.
[20] KIM J, YOON H, KIM M S. Tweaking deep neural networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(9): 5715-5728.
[21] ABIODUN O I, JANTAN A, OMOLARA A E, et al. Comprehensive review of artificial neural network applications to pattern recognition[J]. IEEE Access, 2019, 7: 158820-158846.
[22] LI Y, LIU J, WANG L. Lightweight network research based on deep learning: A review[C]//Proceedings of the 37th Chinese Control Conference (CCC). Wuhan, China: IEEE, 2018: 9021-9026.
[23] SANDLER M, HOWARD A, ZHU M, et al. MobileNetV2: Inverted residuals and linear bottlenecks [C]//Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Salt Lake City, UT, USA: IEEE, 2018: 4510-4520.
[24] MA N, ZHANG X, ZHENG H T, et al. ShuffleNet V2: Practical guidelines for efficient cnn architecture design[C]//Proceedings of Computer Vision—ECCV 2018. Cham: [s.n.], 2018: 122-138.
[25] HAN S, MAO H, DALLY W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding[C]//Proceedings of the 4th International Conference on Learning Representations (ICLR). San Juan, Puerto Rico: ICLR, 2016: 1-14.
[26] HE Y, LIN J, LIU Z, et al. AMC: AutoML for model compression and acceleration on mobile devices [C]//Proceedings of Computer Vision—ECCV 2018. Munich, Germany: Springer, 2018: 815-832.
[27] TAN M, CHEN B, PANG R, et al. MnasNet: Platform-aware neural architecture search for mobile[C]//Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.]: IEEE, 2019: 2815-2823.
[28] ZHANG X, ZHOU X, LIN M, et al. ShuffleNet: An extremely efficient convolutional neural network for mobile devices[C]//Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Salt Lake City, UT, USA: IEEE, 2018: 6848-6856.
[29] 李道伦, 沈路航, 查文舒, 等. 基于神经算子与类物理信息神经网络智能求解新进展[J]. 力学学报, 2023, 56(4): 875-889.
LI Daolun, SHEN Luhang, ZHA Wenshu, et al. New progress in intelligent solution of neural operators and physics-informed-based methods[J]. Chinese Journal of Theoretical and Applied Mechanics, 2024, 56(4): 875-889.
[30] LIAN Y, PENG P, JIANG K, et al. Cross-layer importance evaluation for neural network pruning[J]. Neural Networks, 2024, 179: 10696.

(编辑:刘彦东)