Swin Transformer轻量化：融合权重共享、蒸馏与剪枝的高效策略

韩博[1]，周顺[2]，范建华[2]，魏祥麟[2]，胡永杨[2]，朱艳萍[1]
（1. 南京信息工程大学电子与信息工程学院，江苏 南京 210044；
2. 国防科技大学第六十三研究所，江苏 南京 210007）

摘　要：偏移窗口的分层视觉转换器（Swin Transformer）因其优秀的模型能力而在计算机视觉领域引起了广泛的关注，然而Swin Transformer模型有着较高的计算复杂度，限制了其在计算资源有限设备上的适用性。为缓解该问题，提出一种融合权重共享及蒸馏的模型剪枝压缩方法。首先，在各层之间实现了权重共享，并添加变换层实现权重变换以增加多样性。接下来，构建并分析变换块的参数依赖映射图，构建分组矩阵$\boldsymbol{F}$记录所有参数之间的依赖关系，确定需要同时剪枝的参数。最后，蒸馏被用于恢复模型性能。在ImageNet-Tiny-200公开数据集上的试验表明，在模型计算复杂度减少32%的情况下，最低仅造成约3%的性能下降，有效降低了模型的计算复杂度。为实现在计算资源受限环境中部署高性能人工智能模型提供了一种解决方案。

关键词：偏移窗口的分层视觉转换器；模型轻量化；推理加速；剪枝；蒸馏；权重共享
中图分类号：TP183
文献标志码：A
doi: 10.11959/j.issn.1000−0801.2024209

Swin Transformer lightweight: an efficient strategy that combines weight sharing, distillation and pruning

HAN Bo[1], ZHOU Shun[2], FAN Jianhua[2], WEI Xianglin[2], HU Yongyang[2], ZHU Yanping[1]
1. School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China
2. The Sixty-third Research Institute, National University of Defense Technology, Nanjing 210007, China

Abstract: Swin Transformer, as a layered visual transformer with shifted windows, has attracted extensive attention in the field of computer vision due to its exceptional modeling capabilities. However, its high computational complexity limits its applicability on devices with constrained computational resources. To address this issue, a pruning compression method was proposed, integrating weight sharing and distillation. Initially, weight sharing was implemented across layers, and transformation layers were added to introduce weight transformation, thereby enhancing diversity. Subsequently, a parameter dependency mapping graph for the transformation blocks was constructed and analyzed, and a grouping matrix $\boldsymbol{F}$ was built to record the dependency relationships among all parameters and identify param-

收稿日期：2024−01−01；修回日期：2024−09−01

eters for simultaneous pruning. Finally, distillation was then employed to restore the model's performance. Experiments conducted on the ImageNet-Tiny-200 public dataset demonstrate that, with a reduction of 32% in model computational complexity, the proposed method only results in approximately a 3% performance degradation at minimum. It provides a solution for deploying high-performance artificial intelligence models in environments with limited computational resources.

Key words: Swin Transformer, model lightweight, inference acceleration, pruning, distillation, weight sharing

0　引言

近年来，为了追求更好的模型性能，深度神经网络结构愈发复杂，导致模型计算复杂度不断攀升，对计算和存储能力带来了巨大的挑战。Swin Transformer[1]模型因其优秀的性能，得到广泛应用。该模型针对传统 Transformer[2]模型在计算机视觉任务中存在的问题进行了改进，引入了“Shifted Window”机制，成为一种在视觉任务中表现出色的模型架构。但该模型有着较高的计算复杂度，例如，在 ImageNet-1K 数据集上训练的 Swin-Large 有着高达 1.97 亿的参数量，使得这些模型不适用于有限计算资源的应用，如边缘计算、物联网和实时预测应用等。研究表明，大规模预训练模型存在过度参数化的问题[3]。因此，在尽可能不影响这些预训练模型性能的前提下，对预训练模型进行压缩和结构优化，消除模型的冗余参数是必要且可行的。经过压缩后的 Swin Transformer 模型将更加适应各种应用场景，同时控制部署成本，这对于构建高效的智能系统并确保其在多种环境下的有效应用是极其重要的。

虽然现有 Swin Transformer 模型压缩方法[4-5]已经取得了一定的成功，但是仍面临一些困难，如当前模型压缩方法尚未在计算复杂度和性能之间找到很好的平衡点。文献[4]提出了一种结构化剪枝方法，虽然能减少参数量，但性能损失较大。文献[5]提出一种通过权重多路复用技术压缩视觉变换器模型的方法，但此方法以计算量为代价，换取性能的提升。文献[6]提出一种模型压缩方法，包括网络剪枝、参数量化和知识蒸馏，通过精简结构、低精度表示和信息迁移，提高模型性能并降低计算复杂度。文献[7]通过自适应轮询式剪枝技术动态去除不重要的权重，减少模型存储与计算需求。Transformer 模型压缩方法[7]大多采用单一手段来实现加速，现有研究在如何协调多种压缩技术之间的关系、提升模型压缩效果方面，仍存在不足。此外，大多数研究集中于传统的 Transformer 模型，而针对 Swin Transformer 的压缩研究相对较少。Swin Transformer 通过引入层次化和移动窗口机制，虽然提高了模型适应性和精细度，但也相应增加了模型复杂度。因此，在尽量不损失模型性能的前提下，如何融合多种模型压缩技术降低计算复杂度，实现 Swin Transformer 模型的轻量化，是一个亟待解决的问题。

为此，本文设计了一种融合权重共享、剪枝与蒸馏的模型压缩方法，并根据该方法对 Swin Transformer 模型进行压缩加速并测试其性能。本文主要工作和贡献包括如下两个方面。

（1）本文设计了一种融合权重共享、剪枝与蒸馏的模型压缩方法，首先在 Transformer 各层之间实现了权重共享，并添加变换层实现权重变换以增加权重的多样性。其次构建并分析变换块的参数依赖映射图，得到参数间的相互依赖性，识别出保持网络整体性能的关键参数，从而确定需要同时剪枝的参数组。最后，使用 KL 散度（Kullback-Leibler divergence）蒸馏技术解决由权重共享和剪枝引起性能下降的问题。

（2）为了验证方法的性能，在 ImageNet-Tiny-200 公开数据集上的试验表明，在模型计算

复杂度减少32%的情况下，最低仅造成了约3%的性能下降，有效降低了模型计算复杂度，显著优于现有方法。

1 相关工作

Transformer及其变体模型显著的计算和存储成本，成为其在边缘计算、物联网和对实时性要求严苛场合广泛应用的重要制约因素。面对以上制约因素，学术界和工业界已经投入大量精力研发多种有效的模型压缩与优化策略，旨在大幅度降低Transformer模型的计算复杂度和存储开销，使其能够在更广泛的平台和应用场景上得到高效部署。以下是5种应用广泛的模型压缩技术：张量分解、量化、剪枝、知识蒸馏和权重共享。

张量分解（tensor decomposition）通过对模型权重矩阵进行低秩分解，可以实现参数量的有效削减，同时尽可能保留模型的表达能力和预测精度。这种方法通常包括典型多项式分解（canonical polyadic decomposition，CP）、Tucker分解和奇异值分解等。Ma等[8]提出一种结合CP和Tucker分解的方法，用于分解多头注意力中的线性层，以在保持模型性能的同时压缩注意力层参数，然而，该方法面临参数压缩量不足和过拟合的问题。张量分解方法存在信息损失、计算复杂度高、调优困难、适用性和解释性差等多个缺陷。

量化（quantization）将原本需要浮点数表示的模型参数转换为更低位宽的整数格式，如二值量化、四值量化或多比特量化，从而显著减少内存占用，并加快计算速度，尤其是在支持量化运算的硬件平台上。Bhandare等[9]首次在机器翻译的Transformer模型中引入8位整数量化，但仅对计算图中的矩阵相乘进行量化。量化方法对设备的硬件环境要求较高，难以部署。

剪枝（pruning）的目标是从给定网络中去除冗余参数，以减小其大小并有可能加快推理速度。主流的剪枝方法可以大致分为两种方案：结构性剪枝和非结构性剪枝。两者的核心区别在于，结构性剪枝通过物理去除分组参数来改变神经网络的结构，而非结构性剪枝在不改变网络结构的情况下对部分权值进行归零。在使用剪枝的工作中，对模型进行结构化剪枝的工作更值得关注。Michel等[10]和Voita等[11]分析多头注意力中的多头冗余性，并针对相对重要性较低的注意力头进行剪枝。Fan等[12]提出一种在训练中对网络层使用随机失活，在推理时按需进行层剪枝的方法。

知识蒸馏（knowledge distillation）利用大模型（教师模型）的知识来指导小模型（学生模型）的学习，使得小模型在保持较低复杂度的同时，能近似或逼近大模型的预测性能。此过程涉及从教师模型的输出分布中学到潜在的“软”标签信息。Sanh等[13]提出了三重损失概念，在预训练阶段使用知识蒸馏来训练一个较小的通用语言表示模型，实验证明，在保留97%语言理解能力的前提下，可以将BERT（bidirectional encoder representations from transformers）缩减40%。Tang等[14]提出使用学生logits与教师logits之间的KL散度将BERT模型蒸馏到单层双向LSTM（long short-term memory）模型。

权重共享（weight sharing）在模型的不同部分之间强制执行权重重复利用，典型做法如卷积核的深度可分离结构、循环神经网络中的门控机制或者设计全新的结构使得Transformer模型能够在多个位置共享相同的参数子集。神经网络中权重共享的最初想法是由Lecun和Hinton在20世纪90年代提出的[15]，最近被重新用于自然语言处理（natural language processing，NLP）中的Transformer模型压缩，最具代表性的工作是Albert[16]引入了一种跨层参数共享方法，以防止参数数量随着网络深度的增长而增加，这种技术可以在不严重损害性能的情况下显著减小模型尺寸，从而

提高参数利用效率。

总体来看，当前工业界已经针对 Transformer 模型压缩进行了一些探索，但仍然存在 3 个问题。一是目前的方法在平衡模型计算复杂度与性能之间仍存在不足，压缩方法性能有待提升；二是在融合多种压缩技术方面缺乏深入研究；三是大多数研究集中于传统的 Transformer 模型，而针对 Swin Transformer 的压缩研究则相对较少。Swin Transformer 通过引入层次化和移动窗口机制，虽然提高了模型适应性和精细度，但也相应增加了模型复杂度和压缩难度。

2　方法设计

本文提出一种融合权重共享、剪枝与蒸馏的模型压缩方法，对于需要压缩的 Swin Transformer 模型，首先在多层之间共享权重，提高参数利用率，添加变换层以增加参数多样性；其次构建并分析变换块的参数依赖映射图，得到参数间的相互依赖性，并构建一个二进制分组矩阵 $\boldsymbol{F}$ 来记录所有参数之间的依赖关系，通过深度优先搜索（depth-first search，DFS）算法实现参数分组，属于同一组的参数会被同时移除，实现降低模型计算复杂度；最后，使用 KL 散度蒸馏技术解决由权重共享和剪枝引起性能下降的问题。模型压缩方法流程如图 1 所示。

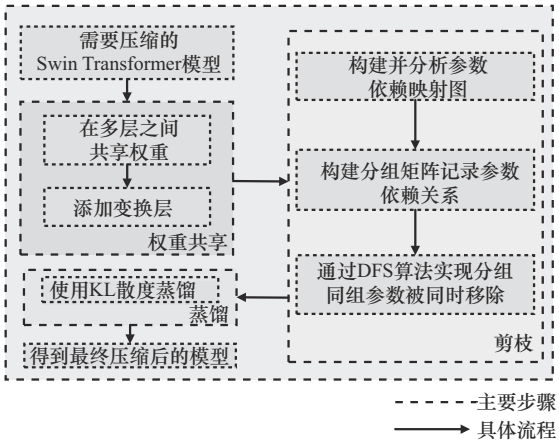

图 1　模型压缩方法流程

2.1　权重共享

权重共享是一种提高参数利用效率的有效方法，其核心思想是跨层共享参数。经典权重共享如图 2 所示。

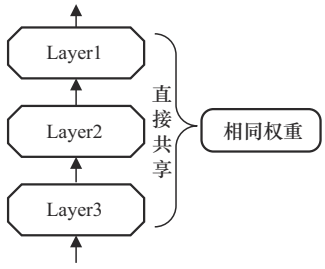

图 2　经典权重共享

权重共享可以表示为一个 Transformer 块的递归更新：

$$Z_{i+1}=f(Z_i,\theta),i=0,\cdots,L-1 \tag{1}$$

其中，Z_i 表示第 i 层序列的特征嵌入，L 为总层数，θ 为 Transformer 块跨各层共享权重。权重共享可以防止参数数量随着网络深度的增长而增长。本文提出一种权重共享方法，具体而言，首先将多层权重合并为一个共享单一权重，同时涉及变换以增加参数多样性，权重变换的关键思想是对共享权重进行变换，使得不同层具有稍有差异的权重，这样既能促进参数多样性，又能提高模型表达能力和训练稳定性。

权重变换共享如图 3 所示，本文将多层权重组合成共享部分上的单个权重，同时添加变换层以增加参数多样性。$T(X)$ 代表着对权重进行变换。

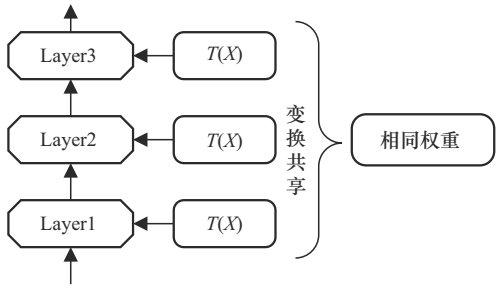

图 3　权重变换共享

本文对多头自注意力（multi-head self-attention，MSA）和多层感知机（multi-layer perceptron，MLP）的权重施加线性变换，作为权重共享的变换层，每一层包括单独的变换矩阵，因此MSA模块和MLP模块的权重在各层之间是不同的。在每个阶段共享MSA和MLP的权重，并增加两个变换层以增加参数的多样性。MSA和MLP线性变换如图4所示。

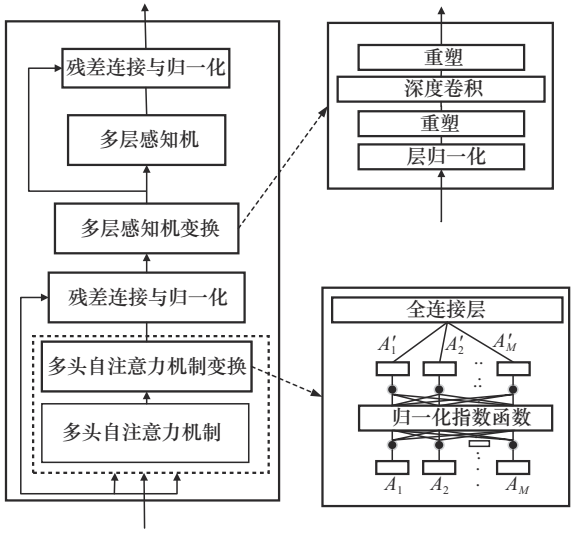

图4 MSA和MLP线性变换

MSA变换：为了提高参数多样性，在自注意力模块前后插入两个线性变换，变换的注意力定义为：

$$\boldsymbol{h}_k=\boldsymbol{A}_k'\boldsymbol{V}_k=\sum_{n=1}^{M}F_{kn}^{(1)}\boldsymbol{A}_n\boldsymbol{V}_k \tag{2}$$

$$\boldsymbol{A}_n=\text{soft max}\left(\sum_{m=1}^{M}F_{nm}^{(2)}\frac{\boldsymbol{Q}_m\boldsymbol{K}_m^{\mathrm{T}}}{\sqrt{d}}\right) \tag{3}$$

其中，M是注意力头的个数，d是查询、键和值矩阵的维度，在第K个头上，通过线性投影生成查询、键和值，分别用$\boldsymbol{Q}_k$、$\boldsymbol{K}_k$和$\boldsymbol{V}_k\in R^{N\times d}$表示，权重用$A_k$表示，归一化指数函数作用于输入矩阵的每一行，$F^{(1)},F^{(2)}\in R^{M\times M}$分别是归一化指数函数前后的线性变换，这样的线性变换可以使每个注意力矩阵$\boldsymbol{A}_n$不同，同时可以将注意力头之间的信息组合在一起，以增加参数方差，最后，将一个全连接层应用于所有头部输出的连接。

MLP变换：在MLP中引入一个轻量级转换，以提高参数多样性。特别地，令输入为$\boldsymbol{Y}=[\boldsymbol{y}_1,\cdots,\boldsymbol{y}_d]$，其中$\boldsymbol{y}_d$表示所有token的第$d$个位置的嵌入向量。然后引入$d$维线性转换将$Y$转换为$\boldsymbol{Y}'=\left[\boldsymbol{C}^{(1)}\boldsymbol{y}_1,\cdots,\boldsymbol{C}^{(d)}\boldsymbol{y}_d\right]$，其中$\boldsymbol{C}^{(1)},\cdots,\boldsymbol{C}^{(d)}\in\boldsymbol{R}^{N\times N}$为线性层独立权重矩阵，然后MLP输出被重新表示为:

$$\boldsymbol{H}=\sigma(\boldsymbol{Y}'\boldsymbol{W}^{(1)}+\boldsymbol{b}^{(1)})\boldsymbol{W}^{(2)}+\boldsymbol{b}^{(2)} \tag{4}$$

其中，σ表示激活函数，$\boldsymbol{W}^{(1)}\in\boldsymbol{R}^{d\times d'}$、$\boldsymbol{b}^{(1)}\in\boldsymbol{R}^{d'}$、$\boldsymbol{W}^{(2)}\in\boldsymbol{R}^{d'\times d}$和$\boldsymbol{b}^{(2)}\in\boldsymbol{R}^{d}$分别是第一层和第二层的权重和偏置，通常设置$d'>d$。为了减少参数量，同时在转换中引入局部性，使用深度卷积，以稀疏化和共享每个权重矩阵中的权重，使得参数量为K^2d,而不是N^2d（$K\ll N$），其中K是卷积核大小。转换后，MLP的输出更加多样化，提高了参数利用效率。

2.2 结构化剪枝

在本文中，通过构建参数依赖映射图，深入分析参数间的相互依赖性，并把相互依赖的参数分到一组，通过L2范数计算参数重要性，根据设置的剪枝比例，按照每组参数重要性程度进行剪枝。

不同依赖关系如图5所示，从一个由三个连续层组成的线性神经网络开始，这个简单的神经网络可以通过去除神经元来进行结构化修剪，使其变得纤细。在这种情况下，很容易发现参数之间出现了一些依赖关系，这迫使$\boldsymbol{W}_L$和$\boldsymbol{W}_{L+1}$同时被剪枝。

具体来说，为了修剪连接$\boldsymbol{W}_L$和$\boldsymbol{W}_{L+1}$的第k个神经元，$\boldsymbol{W}_L[k,:]$和$\boldsymbol{W}_{L+1}[:,k]$都将被删除，当希望通过剪枝某个神经元（黑色表示）实现加速时，与该神经元相连的多组参数需要被同时移除。这些相互依赖的参数就组成了结构化剪枝的最小单元，通常称为组（Group）。

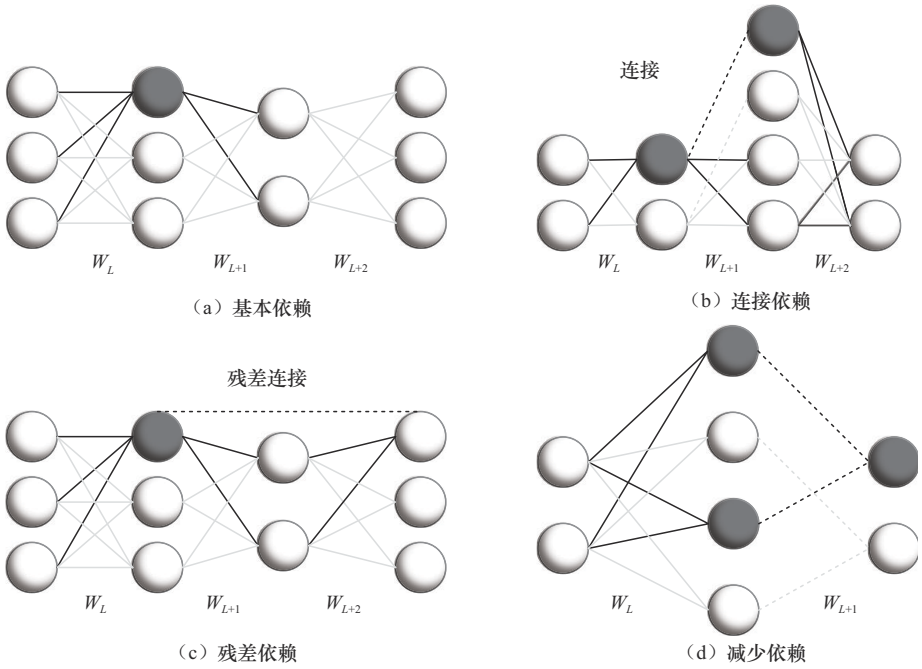

图 5　不同依赖关系

进而，对 Swin Transformer 模型中的参数依赖关系进行建模。在结构化剪枝中，同一分组内的参数是两两依赖的，当希望移除其中之一时，属于该组的参数都需要被同步移除，从而保证结构的正确性，通过构建一个二进制分组矩阵 $\boldsymbol{F}$ 来记录所有参数对之间的依赖关系，如果第 i 层的参数和第 j 层参数相互依赖，就用 $\boldsymbol{F}_{ij}=1$ 来表示。如此，参数分组就可以简单建模为一个查询问题。

$$f(i)=\{j|\boldsymbol{F}_{ij}=1\} \tag{5}$$

其中，i、j 分别代表第 i、j 层的参数，$\boldsymbol{F}_{ij}=1$ 代表第 i 和 j 层之间存在依赖关系。

然而，参数之间是否相互依赖并不仅仅由自身决定，还会受参数之间的中间层影响，相邻层之间的依赖关系是可以递推的。

依赖关系传递性如图 6 所示，相邻层 A、B 之间存在依赖，同时相邻层 B、C 之间也存在依赖，那么就可以递推得到 A 和 C 之间也存在依赖关系，尽管 A、C 并不直接连接。如果相邻层 A、B 之间存在依赖，但 B、C 之间不存在依赖，则 A 和 C 之间不存在依赖。进而，利用相邻层的局部依赖关系，递归地推导出需要的分组矩阵 $\boldsymbol{F}$。而这种相邻层间的局部依赖关系称之为依赖映射图，是一张稀疏且局部的关系图，因为仅对直接相连的层进行依赖建模。由此，分组问题可以简化成一个路径搜索问题，当依赖映射图中节点 i 和节点 j 之间存在一条路径时，可以得到 $\boldsymbol{F}_{ij}=1$，即 i 和 j 属于同一个分组，通过深度优先搜索（DFS）算法进行分组。以某个节点 i 作为起始，找到依赖映射图中与之相连的新节点 j，合并入当前组，直到不存在新的连通节点为止。然后，属于同一组的参数会被同时移除，实现模型压缩。

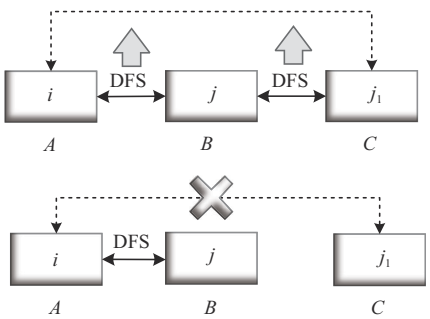

图 6　依赖关系传递性

2.3 蒸馏

为了解决由权重共享和剪枝引起的性能下降问题，进一步采用蒸馏将知识从大型模型转移到小型紧凑模型。对剪枝和权重共享得到的模型进行再训练，得到的模型作为蒸馏中的学生模型。

建立的蒸馏目标函数为：

$$L_{\text{train}} = L_{\text{pred}} + D_{\text{KL}} \tag{6}$$

考虑logits蒸馏，其中预测损失如下。

$$L_{\text{pred}} = \text{CE}\left(\text{softmax}\left(\frac{z_s}{T}\right), \text{softmax}\left(\frac{z_t}{T}\right)\right) \tag{7}$$

其中，T_{pred}和D_{KL}分别代表预测损失和KL散度，z_s和z_t分别是学生模型和教师模型预测的logits，T是温度值，用以控制logits的平滑度，在实验中，将T设为1，CE表示交叉熵损失。

同时，采用知识蒸馏中的KL散度方法，以提升Swin Transformer学生模型的性能。KL散度不仅允许学生模型复制教师模型的直接输出，而且还模仿其概率分布的细微变化，从而捕捉更丰富的类间关系和概率结构。通过最小化学生模型输出和教师模型输出之间的KL散度，学生模型可以更加精确地近似教师模型的决策边界，这在复杂的图像分类任务中尤其关键。此外，相比于传统的分类损失，KL散度为学生模型提供了更细致的错误信号，有助于在训练过程中实现更稳健的学习。

$$D_{\text{KL}}(P \| Q) = \sum_i P(i) \ln \frac{P(i)}{Q(i)} \tag{8}$$

其中，P、Q分别是教师和学生模型的输出概率分布，经过归一化指数函数处理，每个类别i的概率值为$P(i)$、$Q(i)$。

3 实验结果与分析

本节针对用于图像分类任务的Swin Transformer模型，评估了所提方法在ImageNet-Tiny-200数据集上的性能。首先，确立了一个未经压缩的Swin Transformer基线模型的性能基准。随后，分别对采用Torch-pruning[4]模型压缩方法压缩后的模型，以及运用本文所提方法进行压缩后的模型进行了独立测试，并对这3种不同模型的表现进行了比较分析。其中Torch-pruning模型压缩方法是一个专为Pytorch设计的剪枝库，具备模块化设计、多样化剪枝策略及高度灵活性，并与Pytorch深度集成，显著优化模型性能和计算效率。

实验结果显示，相较于原始未压缩的Swin Transformer模型，采用本文所提压缩方法后，在模型计算复杂度减少32%的情况下，最低仅造成约3%的性能下降。这一结论有力证明了所提压缩方案在保证较高性能的同时，有效实现了模型尺寸的精简。相比之下，本实验所提的Swin Transformer模型压缩方法超越了现有的Torch-pruning技术，在维持模型性能与压缩率之间的平衡方面展现出更为优越的表现。

3.1 试验设置与数据集

本文分别采用Swin-Tiny、Swin-Small和Swin-Base3种规模不同的模型作为基线模型。采用ImageNet-Tiny-200数据集，其中包含200种类别，每个类别由ImageNet大型数据集中的一个子集构成，提供100 000张图片，每个类别大约有500张训练图片、50张验证图片和50张测试图片。在训练过程中，本文采用学习率为10^{-3}的Adam优化器对模型进行训练，学习率按照指数形式进行衰减，学习率衰减因子设定为0.95以保证充分的训练。本实验代码采用Python语言编写，使用Pytorch深度学习框架，在NVIDIA 3090上进行训练测试。本实验与目前比较流行的模型压缩方法Torch-pruning进行对比，对比指标具体包括参数量、计算量、模型识别准确率等。

3.2 实验结果分析

对于不同规模的Swin Transformer模型，选择批大小为64，共训练50轮，并对比了原始模型、Torch-pruning和本文方法压缩后模型训练过程中精度的变化，3种模型在ImageNet-Tiny-200数据集上的训练精度变化如图7所示。

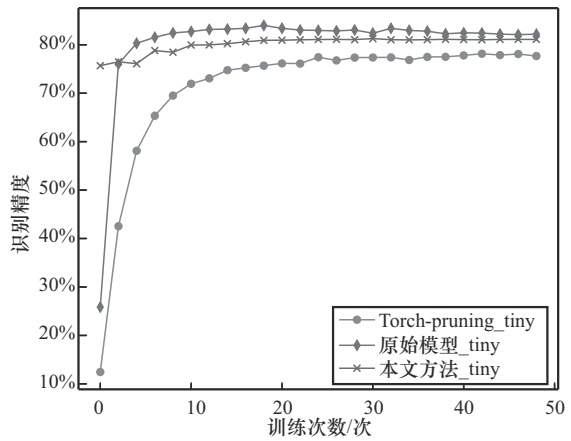

图 7　训练精度变化

图 7 中，本文方法与 Torch-pruning 方法相比，本文方法所得到的模型有着更高的识别准确度，并且与原始模型较为接近。模型性能、计算复杂度比较见表 1，其中性能损失和压缩率的计算方式分别为 Top-1 和参数量所下降的值占原始值的百分比，由表 1 可以得出，Torch-pruning 方法在 Tiny 规模的 Swin Transformer 模型上减少了 30.69% 的参数量，但对应模型的 Top-1 准确率下降了 6.94%。而本文方法在模型计算复杂度更低的情况下，模型平均性能损失仅为 4.5% 左右，优于对比方法。

可以看出本文方法压缩后模型的性能维持在一个较高水平，说明本方法降低模型计算复杂度的同时，在维持模型性能与压缩率之间的平衡方面展现出更为优越的表现。需要说明的是，图 7 中本文方法曲线之所以在第一轮识别准确率较高，是因为模型在蒸馏之前已经得到了再训练，图 7 中本文方法曲线是在最后蒸馏训练阶段中的性能变化曲线。

表 1 是原始模型、Torch-pruning 方法和本文方法压缩后模型性能与计算复杂度的比较，表中结果显示传统的模型压缩算法在高倍率压缩下，模型性能出现较大幅度下降，对模型性能损失较大。但本文方法较好地平衡了参数量、计算量和分类精度之间的关系。与 Torch-pruning 模型压缩算法相比，本文方法在模型计算复杂度更低的情况下，图片分类精度平均实现 3.5% 的上升，与原始基线模型相比，分类精度平均损失为 4.5% 左右。

4　结束语

针对目前 Swin Transformer 模型计算复杂度高的问题，本文提出一种融合权重共享、剪枝与蒸馏的模型压缩方法，实现了 Swin Transformer 模型的轻量化，较好地平衡了模型计算复杂度和性能之间的关系。在 ImageNet-Tiny-200 公开数据集上的试验表明，相较于原始未压缩的 Swin Transformer 模型，采用本文所提压缩方法后，在模型计算复杂度减少 32% 的情况下，最低仅造成了约 3% 的性能下降，有效降低了模型计算复杂度。下一步工作主要是针对 Transformer 类模型设计通用的模型压缩方法，提高模型压缩方法的通用性。

表 1　模型性能、计算复杂度比较

模型规模	模型	Top-1 Acc	Top-5 Acc	参数量/$\times10^6$	计算量/$\times10^9$	压缩率	性能损失
Tiny	基线 Swin	83.98%	95.71%	28.28	4.35	—	—
	Torch-pruning	78.15%	93.04%	19.60	3.01	30.69%	6.94%
	本文方法	81.26%	94.17%	19.18	2.94	32.17%	3.23%
Small	基线 Swin	88.30%	97.2%	49.60	8.52	—	—
	Torch-pruning	80.40%	93.71%	34.38	5.89	30.68%	7.81%
	本文方法	83.51%	95.47%	33.63	5.77	32.19%	5.42
Base	基线 Swin	88.50%	97.07%	87.76	15.14	—	—
	Torch-pruning	79.94%	93.27%	60.77	10.47	30.75%	9.67%
	本文方法	84.04%	95.67%	59.48	10.26	32.22%	5.03%

参考文献：

[1] LIU Z, LIN Y, CAO Y, et al. Swin transformer: hierarchical vision transformer using shifted windows[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2021: 9992-10002.

[2] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017, 30:5998-6008.

[3] KOVALEVA O, ROMANOV A, ROGERS A, et al. Revealing the dark secrets of BERT[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). [S.l.:s.n.], 2019: 4365-4374.

[4] FANG G, MA X, SONG M, et al. Depgraph: towards any structural pruning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR). Piscataway: IEEE Press, 2023: 16091-16101.

[5] ZHANG J, PENG H, WU K, et al. Minivit: compressing vision transformers with weight multiplexing[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR). Piscataway: IEEE Press, 2022: 12145-12154.

[6] 陈婉玉 . 面向视觉处理的 Transformer 模型压缩算法研究及实现[D]. 北京：北京邮电大学，2024.
CHEN W Y. Research and implementation of Transformer model compression algorithms for vision processing[D]. Beijing: Beijing University of Posts and Telecommunications, 2024.

[7] 蔡青竹, 刘强 . 一种高效 Swin Transformer 加速器设计[J]. 北京航空航天大学学报，2024:1-17.
CAI Q Z, LIU Q. A design of an efficient Swin Transformer accelerator[J]. Journal of Beijing University of Aeronautics and Astronautics, 2024: 1-17.

[8] MA X, ZHANG P, ZHANG S, et al. "A tensorized transformer for language modeling" [C]//Proceedings of the 33rd Conference on Neural Information Processing Systems(NeurIPS 2019). San Diego: NeurIPS Press , 2019: 2229-2239.

[9] BHANDARE A, SRIPATHI V, KARKADA D, et al. Efficient 8-bit quantization of transformer neural machine language translation model[C]//Proceedings of the International Conference on Machine Learning(ICML). [S.l.:s.n.], 2019.

[10] MICHEL P, LEVY O, NEUBIG G. Are sixteen heads really better than one? [J]. Advances in Neural Information Processing Systems, 2019(32): 14037 – 14047.

[11] VOITA E, TALBOT D, MOISEEV F, et al. "Analyzing multi-head self-attention: specialized heads do the heavy lifting, the rest can be pruned"[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics(ACL). [S.l.:s.n.], 2019: 5797 – 5808.

[12] FAN A, GRAVE E, JOULIN A. Reducing transformer depth on demand with structured dropout[C]//Proceedings of the International Conference on Learning Representations(ICLR). [S.l.: s.n.], 2020: 1-15.

[13] SANH V, DEBUT L, CHAUMOND J, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter"[J]. arXiv Preprint, arXiv: 1910.01108, 2019 .

[14] TANG R, LU Y, LIU L, et al. Distilling task-specific knowledge from BERT into simple neural networks[J]. CoRR, 2019.

[15] LECUN Y. Generalization and network design strategies[J]. Connectionism in Perspective, 1989, 19(143-155): 18.

[16] LAN Z, CHEN M, GOODMAN S, et al. Albert: a lite bert for self-supervised learning of language representations[C]//Proceedings of the International Conference on Learning Representations(ICLR). [S.l.:s.n.], 2020.

[作者简介]

韩博（2000– ），男，南京信息工程大学电子与信息工程学院硕士生，主要研究方向为深度学习和移动智能计算。

周顺（1983– ），男，博士，国防科技大学第六十三研究所助理研究员，主要研究方向为电磁空间信道建模与仿真计算、智能模型推理加速。

范建华（1971– ），男，国防科技大学第六十三研究所研究员、博士生导师，主要研究方向为软件无线电和频谱智能计算。

魏祥麟（1985– ），男，博士，国防科技大学第六十三研究所副研究员，主要研究方向为边缘计算、深度学习以及无线网络安全。

胡永扬（1977– ），男，博士，国防科技大学第六十三研究所高级工程师，主要研究方向为软件无线电和频谱智能计算。

朱艳萍（1980– ），女，博士，南京信息工程大学电子与信息工程学院副教授，主要研究方向为阵列信号处理、MIMO 雷达/通信频谱共享、智能信号处理等。