电 子 科 技 大 学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

硕士学位论文

MASTER THESIS

论文题目 **基于 Transformer 的图像特征提取方法研究**

学科专业	控制科学与工程
学　　号	**201921060536**
作者姓名	杨有帅
指导教师	刘珊　　副教授
学　　院	自动化工程学院

分类号 TP18 密级 公开

UDC [注1] 621.3

学 位 论 文

基于 Transformer 的图像特征提取方法研究

（题名和副题名）

杨有帅

（作者姓名）

指导教师 刘珊 副教授

电子科技大学 成 都

（姓名、职称、单位名称）

申请学位级别 硕士 学科专业 控制工程与科学

提交论文日期 2022 年 4 月 26 日 论文答辩日期 2022 年 5 月 17 日

学位授予单位和日期 电子科技大学 2022 年 6 月

答辩委员会主席

评阅人

注 1：注明《国际十进分类法 UDC》的类号。

Research on Image Feature Extraction Method Based on Transformer

A Master Thesis Submitted to

University of Electronic Science and Technology of China

Discipline **Control Science and Engineering**

Student ID **201921060536**

Author **Yang youshuai**

Supervisor **Liu shan**

School **School of Electronic Science and Engineering**

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：杨有帅　　日期：2021 年 6 月 1 日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名：杨有帅　　导师签名：

日期：2021 年 6 月 1 日

摘 要

图像特征提取是当前计算机视觉领域的重要技术之一，很大程度上决定着许多视觉任务的精度与速度。随着大数据时代的发展，海量的图像数据与各种复杂的实际应用场景使得高效而精准地从图像中提取特征成为了挑战，学术界因此对图像特征提取方法进行了广泛的研究。

近年来，基于 Transformer 提取图像特征的方法被广泛研究，但相关模型仍存在一些需要改进的地方。首先，Transformer 的复杂度与输入的序列数量成二次关系，阻碍了基于 Transformer 设计的图像特征提取网络对高分辨率的图像建模，并且高昂的计算成本使其很难适用于边缘设备。其次，Transformer 在建模视觉结构时缺乏归纳偏置，使其需要采用超大数据集进行预训练。最后，与卷积神经网络相比，Transformer 模型可优化性较差，对于优化器的选择较为敏感，缺乏稳定性，收敛速度较慢。针对上述问题，本文主要工作如下：

（1）提出了两种加速 Transformer 模型的方法，分别从模型内部和外部两个角度去解决当前 Transformer 模型计算成本高，模型的复杂度与输入的 Token 数量成二次关系的问题。首先是将自注意力机制本身的二次复杂度降低为线性，从内部提高模型的处理速度；然后又提出了一个无参数，可以根据不同输入图片自适应采样从而筛掉不重要 Token 的轻量化剪枝方法，从外部减少无意义的输入。最后将两种方法合并得到了一种新的高效注意力机制（E-Attention）。实验表明，两种方法各自可降低原 Transformer 模型 30%-50%的计算量，而 E-Attention 可以减少原 Transformer 模型 60%-70%的计算量。

（2）在本文提出的 E-Attention 基础上，进一步结合深度卷积和空洞卷积，从平移不变性，局部性，尺度不变性三个角度引入 Transformer 模型缺乏的归纳偏置。然后再利用一个轻量化卷积模块改变传统 Transformer 模型对输入图片的处理方式，从而加快收敛速度，提升稳定性。最终得到了一个结合卷积的高效 Transformer 图像特征提取网络（CEFormer）。实验表明，CEFormer 在性能和运算速度之间均取得了良好的结果。

关键词：Transformer，特征提取，计算机视觉，轻量化

ABSTRACT

Image feature extraction is one of the most important technologies in the field of computer vision, which largely determines the accuracy and speed of many visual tasks. With the development of the era of big data, the massive image data and various complex practical application scenarios make it a challenge to efficiently and accurately extract features from images. Therefore, image feature extraction methods have been widely studied by the academic community.

In recent years, the method of image feature extraction based on Transformer is widely studied, but there are still some areas to improve the relevant models. First, the complexity of Transformer has a quadratic relationship with the number of input sequences, which prevents the image feature extraction network designed based on Transformer from modeling high-resolution images, and the high computational cost makes it difficult to apply to edge devices. Second, Transformer lacks inductive bias when modeling visual structures, resulting in its reliance on large data sets for pre-training. Finally, compared with the convolutional neural network, the Transformer model shows lower optimizability than the standard, especially for the optimizer, the selection of hyperparameters is more sensitive, the lack of stability, and the convergence speed is slow. For the problems described above, the main research work of this thesis is described as follows:

（1）Two methods of accelerating Transformer model are proposed to solve the problems of high computation cost and quadratic relationship between model complexity and input Token quantity in Transformer model from internal and external perspectives respectively. Firstly, the quadratic complexity of the self-attention mechanism itself is reduced to linearity to improve the processing speed of the model from the inside. Then, a parameterless lightweight pruning method is proposed, which can filter out unimportant tokens by adaptive sampling according to different input images and reduce meaningless inputs from the outside. Finally, a new efficient Attention mechanism (E-attention) was obtained by combining the two methods. Experiments show that the two methods can reduce the computation amount of the original Transformer model by 30% to 50% respectively, while E-attention can reduce the computation amount of the original Transformer model by 60% to 70%.

（2）On the basis of E-attention proposed in this thesis, further combining deep convolution and void convolution, inductive bias lacking in Transformer models is introduced from three perspectives of translation invariance, locality and scale invariance. Then, a lightweight convolution module is used to change the processing mode of traditional Transformer model to the input image, so as to accelerate the convergence speed and improve the stability. Finally, an efficient Transformer image feature extraction network (CEFormer) with convolution is proposed. Experiments show that CEFormer achieves good results in both performance and speed.

Keywords: Transformer, Feature Extraction, Computer Vision, Lightweight

目 录

第一章 绪论

1.1 研究背景与意义

作为深度学习领域最为成功的应用之一，计算机视觉包括人脸识别[1-2]、目标跟踪[3]、行人检测[4]、车牌识别[5-6]等。随着大数据时代的到来，互联网信息技术与智能技术进入了快速发展的阶段，随之而来的就是图像数据呈现指数级的增长[7]。在计算机性能飞速提升的帮助下，计算机视觉对于行业发展越发重要。计算机视觉在人工智能市场的占比也是排名居首。

而计算机视觉能有如此好的发展，得益于当前深度学习算法在包括目标检测、目标跟踪、语义分割、图像分类等诸多视觉子任务上表现良好，使得计算机视觉有广阔的市场前景。然而无论针对任何实际应用场景，这些视觉任务要想取得一个良好的结果，很大程度上依赖于对输入图像的特征提取部分。对图像特征信息的充分提取可以很好的提升各类视觉任务的精度与效率，因此对于图像特征提取方法的研究是极为关键且有必要的。

目前在深度学习领域，主要是通过卷积神经网络[8]（Convolutional Neural Network, CNN）对输入的图像进行特征的提取。在过去的十几年时间中，CNN 确实在处理图像方面有非常大的优势，比如图像数据的表示方式是分层的，高级特征（如语义等）会依赖底层特征（如边缘，纹理等），CNN 可以由浅入深的提取抽象的高级特征[9]。另外 CNN 当中的卷积核具有局部性和平移不变性等归纳偏置，可以捕获输入图像的局部信息[10]。然而卷积这种操作先天缺乏对输入图像的全局理解，没有办法对特征之间的依赖关系进行建模，以至于无法充分利用上下文信息。此外，固定的卷积权重无法动态适应输入的变化。也因为这一些缺陷，近一两年来，研究人员开始在计算机视觉领域应用自然语言处理领域的 Transformer 模型，尝试用它来解决各类视觉任务，基于 Transformer 对图像进行特征的提取。因为与 CNN 相比，Transformer 可以不受局部相互作用的限制，建模较长距离的依赖关系，还可以实现并行计算，在各类视觉任务当中均取得了良好的实验结果。

诚然利用 Transformer 去提取图像特征有许多好处，但这个新兴领域仍然有一些不足需要解决。首先，任何基于 Transformer 去提取图像特征的模型都存在一个先天瓶颈，那就是给定由输入图片切分并变换得到的 Token 序列作为输入，自注意力机制将序列当中任意一个 Token 与其他 Token 关联起来迭代学习特征表示，导致了模型的时间与空间复杂度均与输入的 Token 数量成二次关系。这种二次复杂度阻止了 Transformer 对高分辨率的图像建模，并且高昂的计算成本使其很难适用

于边缘设备；另外 Transformer 需要采用超大的数据集进行预训练在实验效果上才能和 CNN 媲美；然后就是与 CNN 相比，Transformer 稳定性较差，对于优化器，超参数的选择较为敏感，收敛速度较慢。

对于上述研究中的有待改进之处，本文对基于 Transformer 的图像特征提取方法进行了深入的研究。其中针对计算成本高昂，复杂度与输入 Token 序列成二次关系这个问题，本文提出了两种方法来加速 Transformer，首先从外部角度先对输入 Token 根据各自对最终用于分类的 Class Token 的重要程度进行评分，然后根据得分采样保留部分 Token，从 Token 维度进行剪枝；然后再从内部角度使用一个组合函数替换计算自注意力矩阵的 Softmax 算子，得到新的线性注意力；最后将两种方法合并，得到了一种新的高效的注意力机制（E-Attention），极大程度上降低了 Transformer 的计算量。而针对 Transformer 对训练数据的超大需求，深入分析后发现是因为 Transformer 缺乏类似于 CNN 的归纳偏置，因此本文在 E-Attention 的基础上从不同角度结合卷积引入平移不变性，尺度不变性，局部性。最后针对 Transformer 稳定性较差这个问题，再利用一个轻量化卷积模块改变了传统 Transformer 模型对输入图片的处理方式，从而加快收敛速度，提升了稳定性。最终得到了一个结合卷积的高效 Transformer 图像特征提取网络（CEFormer），在利用 Transformer 提取图像特征上取得了精度与速度的良好平衡。

1.2 国内外研究现状

将 Transformer 引入到计算机视觉领域，解决各类视觉任务是近两年较新的工作，本小节汇总了相关研究，从基于 Transformer 的特征提取方法和 Transformer 模型压缩两个方面去阐述。

1.2.1 基于 Transformer 的图像特征提取方法研究现状

2017 年谷歌提出了 Transformer[10]，这是一个在自然语言处理领域里程碑式的模型。2020 年 10 月 Dosovitski 等提出了 ViT[11]（Vision Transformer）模型，首次将 Transformer 引入到视觉领域，证明了 Transformer 作为提取图像特征的网络也可以取得较好的效果。从此开始，Transformer 作为图像特征提取网络的研究进入了飞速发展的过程，发展方向可以分为两大类，分别是训练策略和模型这两方面的改进。其中训练策略方面是指在训练过程中对 ViT 模型进行改进，而模型方面则是指对基于 Transformer 所设计的图像特征提取网络中的各个模块进行改进。

目前主流的训练策略主要是指 2020 年 12 月 Touvron 提出的模型 DeiT[12]。这个模型的提出是为了解决 ViT 模型需要利用超大数据集 JFT-300M 进行预训练的

缺点。DeiT 改进的核心是通过将蒸馏学习引入到 Transformer 模型的训练过程当中，并且还提供了一组实验效果良好的超参数，在此之后的大多数 Transformer 模型在实验过程中参考了这组超参数，本文实验当中的超参数设置也是如此。

模型方面的提升主要针对于以下五个部分：分别是 Token、位置编码、注意力、正则化位置改动和分类预测。

（1）Token 改进

模型当中 Token 部分的改进可以分为两个方向，首先是 Image to Token，这是指在模型的初始阶段，怎样将输入图片转化为 Token 序列。其次是 Token to Token，也就是指在 Transformer 模型的中间阶段如何在多个编码器间传递 Token 序列。下面分别进行叙述。

（a）Image to Token。主要包括有重叠和无重叠两种转换方式。ViT 和目前主流模型例如 2021 年 2 月南京大学提出的 PVT[13]和 2021 年 3 月微软亚洲研究院提出的 Swin Transformer[14]等都是采用了无重叠的转换方式，这种方式是将输入的图片切分为互不重合的块（Patch），然后每个 Patch 再单独变换从而得到最后的 Token 序列。这两种变换方式主要的不同是在于切分输入图片之后所得到的 Patch 是否有重合的地方，这两种方法在实验过程中并没有明显的区分，简单修改内部参数就可以实现这两种变换方式，与之相关的算法包括 2021 年 1 月新加坡国立大学提出的 T2T-ViT[15]和 2021 年 6 月南京大学和商汤共同提出的 PVTv2[16]。

（b）Token to Token。大多数 Transformer 模型在中间阶段的多个编码器之间传递 Token 序列的方法与 Image to Token 方法一样，然而少部分模型在这方面进行了对应的改进。可以将其分为固定窗口和动态窗口两类方法。固定窗口是指在一开始就把 Token 序列的变换方式定义好，多个编码器之间固定的规则进行 Token 序列的传递。比如上述的重叠和无重叠两种转换方式就是固定窗口，按照事先规定好的规则进行窗口的划分，与输入图片无关。而动态窗口则相反，输入图片会影响 Token 序列传递过程中窗口的划分，窗口随输入图片动态变化。与之相关的算法包括 2021 年 8 月牛津大学提出的 PS-ViT[17]和 2021 年 6 月谷歌提出的 Token Learner[18]。算法 PS-ViT 的提出是因为在作者看来 ViT 模型所采用的固定窗口划分机制比较死板，再加上实际输入的图片可能非常冗余，一旦采用固定的窗口去划分，那么对于最终的分类而言，可能只有极少数窗口中的 Token 有意义。假设最终需要分类的物体在图片正中间，那么待分类的物体周围的 Token 只会消耗计算成本。所以作者设计了一个可以根据输入图片自适应采样的 Token 化方法，摒弃最初固定划分窗口的方式，首先将采样点初始化，然后在训练过程中一直调整采样点的位置坐标。Token Learner 算法也是类似的设计思路，作者提出了一种可以根据空间注意力从

而在训练过程中自适应地学习出最有意义的 Token，最终将 ViT 模型原始的 1024 个 Token 减少到 8-16 个，在保持性能的前提下极大程度上减少了计算量。

（2）位置编码

位置编码是用来描述 Transformer 在初始阶段将输入图片切分得到的 Patch 之间的相对位置关系。可以根据位置编码向量是否显示设置分为两种类型，分别是显式位置编码和隐式位置编码。其中对于显式位置编码而言还可以分为绝对和相对位置编码，另外所有的位置编码也可以根据位置编码的参数是否可以学习划分为固定和可学习位置编码。而隐式位置编码是值位置编码向量根据不同输入图片的语义区分 Patch 之间的位置信息，也因此对于输入图片长度会发生改变的使用场景来说，隐式位置编码由于是根据输入图片的语义而生成的，通常更加适用于这类场景。由论文[19]可知 CNN 除了可以将位置信息编码之外，同时层数越深越包含更多的位置信息。作者由此得到结论，可以利用卷积自带位置信息的特性来隐式对位置向量进行编码。

后续的许多算法都直接参考了这一结论，比如 2021 年 2 月美团提出的 CPVT[20] 和 2021 年 7 月中科大和微软提出的 CSWin Transformer[21]均利用了这个特性来增强位置编码。

（3）自注意力

自注意力机制是 Transformer 模型最重要的部分，它最大的特点是不含归纳偏置，在输入海量数据的情况下就可以学习到拥有良好泛化性能的特征。在数据量充足的情况下，注意力机制有巨大的优势，但是一旦出现数据量缺乏的情况注意力机制就会变为 Transformer 模型的劣势。现今诸多算法的改进都是为了能够将归纳偏置引入到 Transformer 模型当中，实现在减少数据量的情况下加快模型的收敛速度，并且提升性能。除此之外，由于自注意力机制是全局的，对于高分辨率的输入图片而言，模型的计算消耗巨大，许多算法也是针对于这一缺点进行改进。

目前针对自注意力机制的改进可以分为全局注意力和引入额外的局部注意力这两个方向。

（a）全局注意力。最为典型的全局注意力就是 Transformer 模型当中的多头注意力，当输入图片的分辨率较高时，转化得到的 Token 数量较多，此时在计算注意力时计算成本消耗较大。改进方向大体可分为降低全局注意力计算量和线性注意力机制两类。这里主要概述前者，后者在第三章重点说明。

全局注意力计算主要包括两部分，分别是对于 **QK** 矩阵相似性的计算，以及后续与 **V** 矩阵相乘的计算。因此可以将 **QKV** 各自的维度减低，从而减少这部分的计算消耗。以 PVT 为例，它提出空间缩减模块，先将各个编码器的输入 Token 序列

还原出空间结构，再利用卷积来缩减空间结构的维度，最后再将其转化为 Token 序列输入到编码器当中，使得当前编码器内部的矩阵维度减少，然后再计算注意力。2021 年 4 月脸书提出的 MViT[22]的设计思路与 PVT 相似，并且在 2021 年 12 月进一步提出了 Imporved MViT[23]。

（b）引入额外的局部注意力。这是指在计算注意力的时候仅计算局部窗口，而非全局窗口，可以有效降低计算成本。在将局部注意力引入之后，需要跨窗口交互信息，否则性能会出现一定程度的下降。这种思想的典型算法是 Swin Transformer，它提前划分好窗口，将自注意力计算过程限制在其中，称为窗口注意力（Window based Self-Attention, W-MSA），可以显著降低计算量。同时为了实现不同窗口之间的交互，作者又在此基础上进一步提出移位窗口注意力（Shifted window based Self-Attention, SW-MSA），将窗口往右下方移位，然后在不同阶段之间交替使用 W-MSA 和 SW-MSA，这也要求实验过程中有偶数个阶段，从而在事先划好的窗口内部计算局部注意力，同时可以跨窗口的交互信息。

尽管在解决输入图片分辨率增加而导致的巨大计算成本消耗问题上 Swin Transformer 算法表现较好，但内部 SW-MSA 的结构设计过于复杂，并且难以部署。后续许多算法基于此提出了大量的针对性改进，首先是去掉 SW-MSA，依然需要全局注意力计算模块，也就是由带有减少计算量功能的全局注意力计算模块来实现跨窗口交互，如 Imporved MViT。另外一个改进是同样也是去掉 SW-MSA，但跨窗口信息交互由改进论文所提出的特定模块提供，如 2021 年 6 月腾讯提出的 Shuffle Transformer[24]和 2021 年 5 月中科院提出的 MSG-Transformer[25]。

从引入卷积归纳偏置角度，也有不少高效的改进，典型的例如 ViTAE[26]、ELSA[27]、ConViT[28]、PiT[29]、CvT[30]、LV-ViT[31]、GG-Transformer[32]、CMT[33]、GLiT[34]和 ConTNet[35]。

（4）正则化位置改动

在 Transformer 模型中正则化（Norm）通常是 Layer Norm，根据 Layer Norm 放在自注意力和前馈网络的前后，可以分成 pre norm 和 post norm 方式，需要根据实验来确定选择 pre norm 还是 post norm。绝大多数模型，例如 ViT 和 Swin Transformer 都是 pre norm。2021 年 11 月微软亚洲研究院提出的 Swin Transformer v2[36]则是 post norm，因为这样在实验过程中更加适应模型的参数量变化，也就是说对参数量变化更加鲁棒。

（5）分类预测

在 ViT 模型中通过额外添加一个 Class Token，将该 Token 对应的编码器输出输入到多层感知机分类头进行分类。而一些方法考虑像常规的图像分类一样，直接

聚合所有特征，不再单独引入一个 Class Token。CPVT 和 2021 年 5 月南京大学提出的 ResT[37]均采用平均池化聚合特征。

1.2.2 Transformer 模型压缩研究现状

尽管 Transformer 在许多视觉任务中都取得了成功，但由于对内存和计算资源的高要求阻碍了其在资源有限的边缘设备上的实现。因此目前有不少工作的着眼点放在压缩和加速 Transformer 上面。相关优化主要可以分为三个方向。首先是从运算方面考虑，通过将计算量减少的方式来加速 Transformer 的运算速度，主要的方法是量化和低秩分解；第二个方向是将现有的模型简化，在尽可能不改变精度的情况下去简化模型，主要方法为剪枝和知识蒸馏；最后一个方向是从一开始就直接设计一个轻量化的 Transformer，主要方法有自动神经网络结构搜索。

（1）量化压缩

量化是指使用低比特宽度的权重值来代替全精度值而不会改变原本的网络结构。对于整数表示来说，一个有符号整数假设有 n 位，那么它的范围就是[-2n,2n-1]。而对于浮点数的表示来说，给定一个浮点数 M，则 M=(-1)SxMx2E，其中字母 S 表示正负，M 代表尾数，E 表示阶数。比较经常使用的 32 位浮点数，或者记为 FP32 的表示范围是[-3.4x1038, -1.18x10-38]∪[1.18x10-38, 3.4x1038]。2021 年 6 月北大和华为联合提出了一种针对 Transformer 的后训练量化方法[38]，这种方法表现优于卷积神经网络中的量化算法。后训练量化是一种可以在不进行额外训练的情况下进行压缩的算法。首先研究人员将后训练量化建模为寻找最优的量化步长问题。为了更好地保留注意力层的功能，他们分析了注意力层量化前后特征的不同，引入了注意力层排序的损失函数。并与量化前后特征分布的相似度损失进行了联合优化。此外，作者还提出考虑到不同网络层的特征多样化不同，可以根据注意力特征和输出特征权重矩阵的核范数来决定每个网络层的量化比特。在不进行额外训练的情况下，Deit-B 的 8bit 模型在 ImageNet 图像分类任务上可以实现 81.29%的精度。2021 年 11 月北大为了实现 Transformer 的快速量化，开发了一个高效的框架 PTQ4ViT[39]。作者分析了 Transformer 的量化问题，同时发现 Softmax 和 GELU 函数激活值的分布与高斯分布有很大不同，还观察到常见的量化指标，例如均方误差和余弦距离，无法准确确定最佳缩放因子。为此作者提出了孪生均匀量化方法来减少这些激活值的量化误差，并且建议使用海森矩阵引导度量来评估不同的缩放因子，从而以较小的成本提高校准的准确性，整体结构如图 1-1 所示。实验表明，PTQ4ViT 在 ImageNet 分类任务上实现了接近无损的预测精度。2022 年 1 月中科院提出了一种完全可微的 Transformer 量化方法 Q-ViT[40]，第一次将 ViT 量化极限

推导 3-bit。研究人员提出了一种新技术来解决量化尺度和位宽联合训练中的收敛问题，并且通过分析 Transformer 层中所有组件的量化稳定性，表明多头自注意力和高斯误差线性单元是 Transformer 量化的关键。

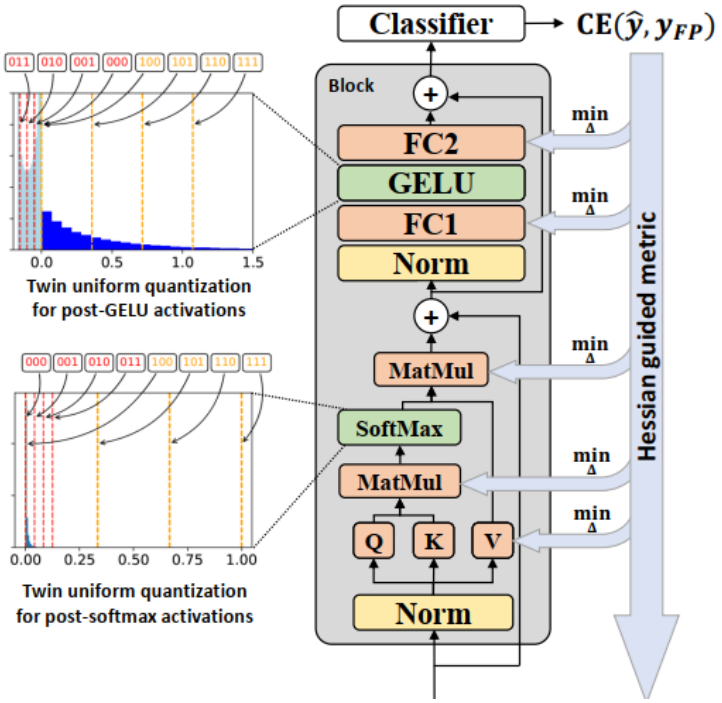


图 1-1 PTQ4ViT 结构图[38]

（2）低秩分解

低秩分解是指使用矩阵分解的方式将原矩阵分解为低阶近似的矩阵，从而达到减少模型参数量的目的。2021 年 12 月中科院提出了一种新的内存经济注意力机制 Couplformer[41]，利用空间信息耦合注意力图来代替传统的自注意力模块。作者将注意力提解耦成两个子矩阵，并根据空间信息生成对齐分数。Couplformer 在 ImageNet 分类任务上与常规 Transformer 相比可以显著降低 28%的内存消耗，同时满足足够的精度要求。

（3）剪枝

剪枝是将 Transformer 模型中冗余的部分去掉。2021 年 4 月浙江大学和华为首次提出了针对 Transformer 的剪枝方法[42]。该方法可以识别每个层中通道的影响，然后执行相应的修剪。通过促使 Transformer 通道的稀疏性，来使得重要的通道自动得到体现。同时为了获得较高的剪枝率，可以丢弃大量系数较小的通道，而不会造成显著的损害。2021 年 6 月北大和华为比较卷积神经网络剪枝和 Transformer 剪枝的差异，首次提出剪枝 Transformer 的 Patch 数来实现模型加速。作者根据注意力机制会逐层聚集不同的 Patch，提出 Patch Slimming 方法[43]，在自上而下的范式

中舍弃无用的 Patch。首先确定最后一层的有效 Patch，然后用它们来指导前面几层的 Patch 选择过程。对于每一层，Patch 对最终输出特征的影响是近似的，而影响较小的 Patch 将被删除。2021 年 10 月英伟达提出了一个延迟感知全局剪枝框架 NViT[44]来对 Transformer 进行压缩。它通过重分配使用的参数，能够更高效地利用参数，同时能够找到一个延迟和精度的平衡点。作者首先确定出剪枝空间，然后再对各组权重评估其重要性，迭代地砍掉重要性低的组，观察剪枝后网络结构的变化，总结出设计规则，并依靠规则得到最终的 NViT，从而实现对模型进行剪枝。2021 年 11 月南京大学提出了一个统一剪枝框架 UP-ViT[45]，此方法侧重于剪枝所有 ViT 组件，同时保持模型结构的一致性。实验表明剪枝后的模型可以很好的泛化到下游任务当中。2021 年 12 月莫纳什大学和悉尼大学联合提出了一种新的多头注意力和卷积操作之间的权重共享方案[46]，并将搜索问题转换为在每个多头注意力层中寻找参数子集。权重共享方案进一步允许设计一种 Single-Path Vision Transformer 剪枝方法（SPViT），在给定目标效率约束的情况下，快速将预训练的 ViT 剪枝成精确而紧凑的混合模型，并显著降低搜索成本。2022 年 1 月普林斯顿大学提出了一种新颖的 ViT 模型压缩框架[47]，该框架联合减少注意力头、神经元和序列维度的冗余。作者提出了一种基于统计依赖的剪枝标准，该标准可以推广到不同维度以识别有害成分。此外作者还将多维压缩作为一种优化，学习跨越三个维度的最佳剪枝策略，以在计算预算下最大化压缩模型的准确性。

（4）知识蒸馏

知识蒸馏旨在通过大型的教师网络转移知识训练得到轻量化的学生网络[48-50]。与教师网络相比，轻量化的学生网络更容易部署在资源有限的设备上。最早将知识蒸馏用于 Transformer 的模型就是 1.2.1 小节陈述过的 DeiT，这里不加赘述。2021 年 11 月北京大学提出了一种新的 Patch 级别的流形知识蒸馏策略[51]。与传统的知识蒸馏方法相比，本文提出通过图像和 Patch 之间的关系从教师 Transformer 中挖掘有用的信息，然后进一步探索了一种有效的细粒度流形蒸馏方法，该方法可以同时计算教师和学生模型中的图片，Patch 和随机选择的流形。

（5）轻量化模块设计

这是指手动设计参数量较小，运算量较低的 Transformer 模型，而不是直接将现有的模型压缩成更小的模型。2021 年 4 月俄勒冈大学提出了紧凑型 Transformer[52]，通过一种新颖的 Token 序列合并策略和卷积的使用，消除了对类标记和位置嵌入的需求。与卷积神经网络相比，紧凑型 Transformer 具有更少的参数，同时获得了相似的精度。2021 年 8 月北京大学提出了一种高效超分辨率 Transformer[53]，这是一种混合 Transformer，由轻量级的卷积神经网络主干和轻量

级 Transformer 主干构成。其中轻量级的卷积神经网络主干通过动态调整特征图的大小，以较低的计算成本提前深度超分辨率特征。而轻量级 Transformer 主干占用的内存很小，内部提出了一个特征分割模块将长序列分割成子段，可以显著减少内存占用。

（6）神经网络结构搜索

这是指构造一个搜索空间，这个空间可以产生多种网络结构，然后利用特定的搜索策略，找到这个搜索空间下的最优网络结构。2021 年 6 月香港大学提出了一种新的神经网络结构搜索方法，称为 HR-NAS[54]，它能在保持高分辨率表示的同时，通过有效地编码多尺度的上下文信息，找到适合不同任务的高效且准确的网络。在 HR-NAS 中同时更新搜索空间和搜索策略。为了更好地编码 HR-NAS 的搜索空间中的多尺度图像上下文，作者设计一个轻量级 Transformer，轻量级 Transformer 的计算复杂性可根据不同的目标函数和计算预算动态地改变。为了维持学习网络的高分辨率表示，HR-NAS 采用多分支架构，该架构可提供多个特征分辨率的卷积编码。最后，作者又提出了一个有效的细粒度搜索策略来训练 HR-NAS，这可以有效地探索了搜索空间，并找到给定各种任务和计算资源的最佳架构。2021 年 6 月悉尼大学提出了 Transformer 架构自动搜索器 ViTAS[55]，不同 Token 嵌入、序列大小、注意力头数量和宽高的架构都可以从构建 Transformer 中导出。此外研究人员还提出了一种用于 Transformer 的新型权重共享定制范式以便进一步提高搜索架构的性能。

1.2.3 现状总结

本小节先介绍了基于 Transformer 的图像特征提取方法的发展现状，并将其进行了完善的归纳总结。然后介绍了目前加速 Transformer 的一些工作，并对其中代表性的方法进行了介绍。从中可以看出这些加速模型，大多都需要引入额外的参数来判断哪些单元是否冗余，或者需要比较复杂的设计来实现轻量化的 Transformer。其次部分基于 Transformer 的图像特征提取方法缺乏对注意力本质的系统理解，不够全面。

1.3 本文的主要工作

Transformer 最近在计算机视觉领域受到了广泛且深入的研究，本文主要研究了基于 Transformer 的图像特征提取方法。为了降低 Transformer 的计算成本，引入其所缺乏的各种归纳偏置，加强稳定性，本文就这三个问题，开展了一系列研究，主要工作如下：

（1）提出了两种加速 Transformer 模型的方法，分别从模型内部和外部两个角度去解决当前 Transformer 模型计算成本高，计算时间与内存使用方面与输入序列成二次复杂度的问题。首先是将自注意力机制本身的二次复杂度降低为线性，从内部提高模型的处理速度。然后又提出了一个无参数，可以根据不同输入图片自适应采样从而筛掉不重要 Token 的轻量化剪枝方法，从外部减少无意义的输入。最后将两种方法合并得到了一种新的高效注意力机制（E-Attention）。实验表明，两种方法各自可降低原 Transformer 模型 30%-50%的计算量，而 E-Attention 可以减少原 Transformer 模型 60%-70%的计算量。

（2）在本文提出的 E-Attention 基础上，进一步结合深度卷积和空洞卷积，从平移不变性，局部性，尺度不变性三个角度引入 Transformer 模型缺乏的归纳偏置。然后再利用一个轻量化卷积模块改变了传统 Transformer 模型对输入图片的处理方式，从而加快收敛速度，提升稳定性。最终得到了一个结合卷积的高效 Transformer 图像特征提取网络（CEFormer）。实验表明，CEFormer 在性能和运算速度之间均取得了良好的结果。

1.4 论文结构安排

本文对于各个章节的结构安排如下所示：

第一章是为绪论部分。介绍了图像特征提取对当前大数据时代的研究价值以及背景意义。再然后围绕本文的主要研究内容，阐述了基于 Transformer 的图像特征提取方法和 Transformer 模型压缩的国内外研究现状。最后对本文的主要工作和结构安排进行了说明。

第二章是相关工作的理论基础，介绍了本文在基于 Transformer 去设计改进图像特征提取网络所涉及到的相关算法与理论知识。首先介绍了改进 Transformer 图像特征提取网络过程中涉及到的深度学习技术，最后系统介绍了 Transformer 的相关理论。

第三章详细介绍了本文提出的两种加速基于 Transformer 的图像特征提取网络的方法。为了解决 Transformer 自身计算成本高昂，模型的复杂度与输入的 Token 数量成二次关系的问题，本文从内部和外部两个角度去加速模型。首先从外部角度先对输入 Token 根据各自对最终用于分类的 Class Token 的重要程度进行评分，然后根据得分采样保留部分 Token，从 Token 维度进行剪枝。然后再从内部角度使用一个组合函数替换计算自注意力矩阵的 Softmax 算子，得到新的线性注意力。最后将两种方法合并得到了一种的高效注意力机制。

第四章设计了一种结合卷积的高效 Transformer 图像特征提取网络。首先在第三章所提出的高效注意力机制的基础上，结合深度卷积和空洞卷积，从平移不变性，局部性，尺度不变性三个角度引入归纳偏置。最后利用一个轻量化卷积模块改变 Transformer 模型对输入图片的传统处理方式，从而加快收敛速度，提升稳定性。

第五章对本文的研究工作进行了总结，对后续可优化的地方进行展望。

第二章 卷积神经网络与 Transformer 理论基础

本章将详细介绍后面两章会用到的部分基础知识，是后续章节创新性工作的基础，主要包括卷积神经网络的概述和 Transformer 的基础知识。

2.1 卷积神经网络概述

卷积神经网络[55]（Convolutional Neural Network, CNN）特别适合计算机视觉任务的神经网络，在图像分类、目标检测和跟踪等诸多视觉任务中得到了广泛的应用，它可以模拟视觉神经[56]对图像进行识别。CNN 基本架构如图 2-1 所示，使用图片作为输入，一系列卷积和池化操作在输入图片上作用，网络最后普遍会加入一系列全连接层。如果执行多分类任务，最终输出还要经过 Softmax 激活函数。每个 CNN 有 4 个基本部分：卷积层，激活函数，池化层和全连接层[57]。每层都是由包含许多神经元的特征图组成。下面具体介绍每一层的具体结构。

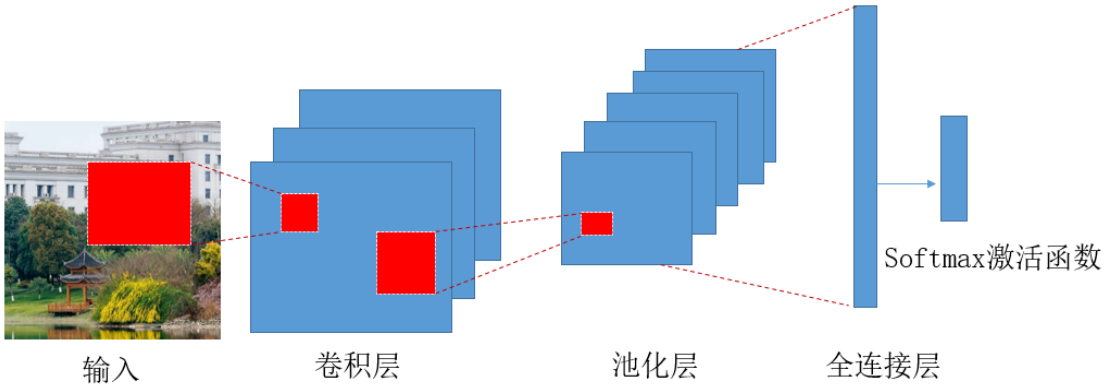


图 2-1 CNN 基本架构

2.1.1 卷积层

卷积层包含若干个许多卷积单元，主要通过卷积操作来提取输入图像的特征。卷积是 CNN 的基础操作，在实际工业应用中全连接层由卷积操作所替代。图像的局部信息是通过特定尺度的卷积核（Kernel）作用于图像区域的一部分而得到。

卷积核本身代表对空间区域内特定特征模型的提取，不管输入图片大小如何变化。比如，部分卷积核负责提取边缘特征，部分卷积核负责提取拐角特征，图像上不同区域共享卷积核。即使输入图片大小不同，也可以使用相同的卷核操作。

卷积核也被叫做滤波器，假设卷积核的高和宽分别为 k_h 和 k_w，则称之为 k_hxk_w 卷积，比如 3x5 卷积，就是指卷积核的高为 3，宽为 5。卷积核通过在图像上滑动

得到最终的特征图。每一层卷积所输出的特征图上每个点的数值，是由输入图片上 $k_h x k_w$ 的区域的数值与卷积核上的所有元素相乘再相加所得，所以输入图像上 $k_h x k_w$ 区域内每个元素数值都会影响输出特征图各点的数值。这个区域称之为感受野。卷积核的计算过程如图 2-2 所示。

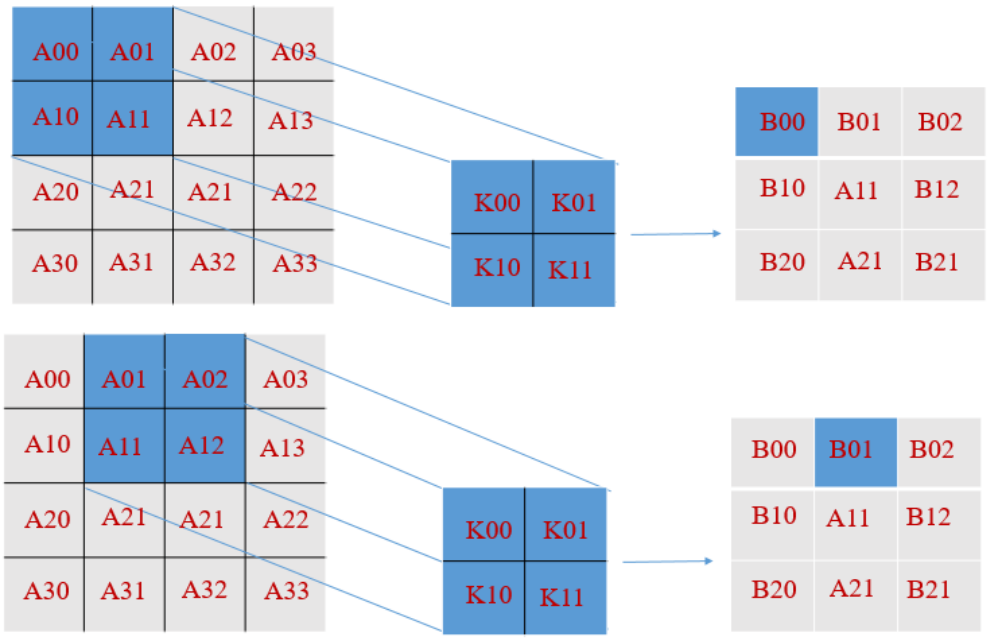


图 2-2 卷积计算过程

图 2-2 中 2x2 卷积核在 4x4 的图像上一步步卷积得到 3x3 的特征图。卷积核每移动一个像素位置，也就是卷积核的步长（Stride）为 1。另外为了避免卷积之后图片尺寸变小，通常会在图片的外围进行填充（Padding），一般是用 0 去填充。卷积核内部的参数是通过网络训练得到更新。

除了上述卷积的基本概念以外，与本文密切相关的还有卷积本身的特性。首先是不同层级卷积可以提取不同的特征。这样，通过加深网络层数，CNN 就可以有效地学习到图像从细节到全局的所有特征了。其次，卷积核的权重是共享的，因为卷积计算实际上是使用卷积核在图片上进行滑动，相乘再相加。

2.1.2 池化层

池化层[58]通常放在卷积层后面，可以将前面卷积层输出的特征图维度降低，还可以模拟人类的视觉神经，得到更高层次的抽象特征。池化层概念的引入是因为输入图像以及特征图一般具有较高的分辨率，一方面会占用较大的计算资源，另外一方面也很容易造成模型的过拟合。而池化层刚好可以降低输入图像以及特征图

的参数量，减少其冗余信息，并且还可以将卷积核的感受野在训练过程中增大，同时防止模型的过拟合，使得模型对输入图像的特征位置变化更加鲁棒。

有两种常见的池化层，分别是平均和最大池化。对于前者平均池化而言，它是指在整个过程中，先将给定的输入图片划分为若干个矩形子区域，然后再输出每个矩形子区域内部所有元素的平均值。这样可以提取特征图中所有特征的信息进入下一层，更大程度的保留图像的背景信息。而对于最大池化而言，它先将输入图片划分为若干个矩形子区域，然后再输出每个矩形子区域内部所有元素的最大值。这样可以将特征图当中响应最为强烈的部分提取出来进入下一层，将网络中大量冗余信息摒弃，使网络更加容易被优化。这两种池化的计算过程如图 2-3 所示。

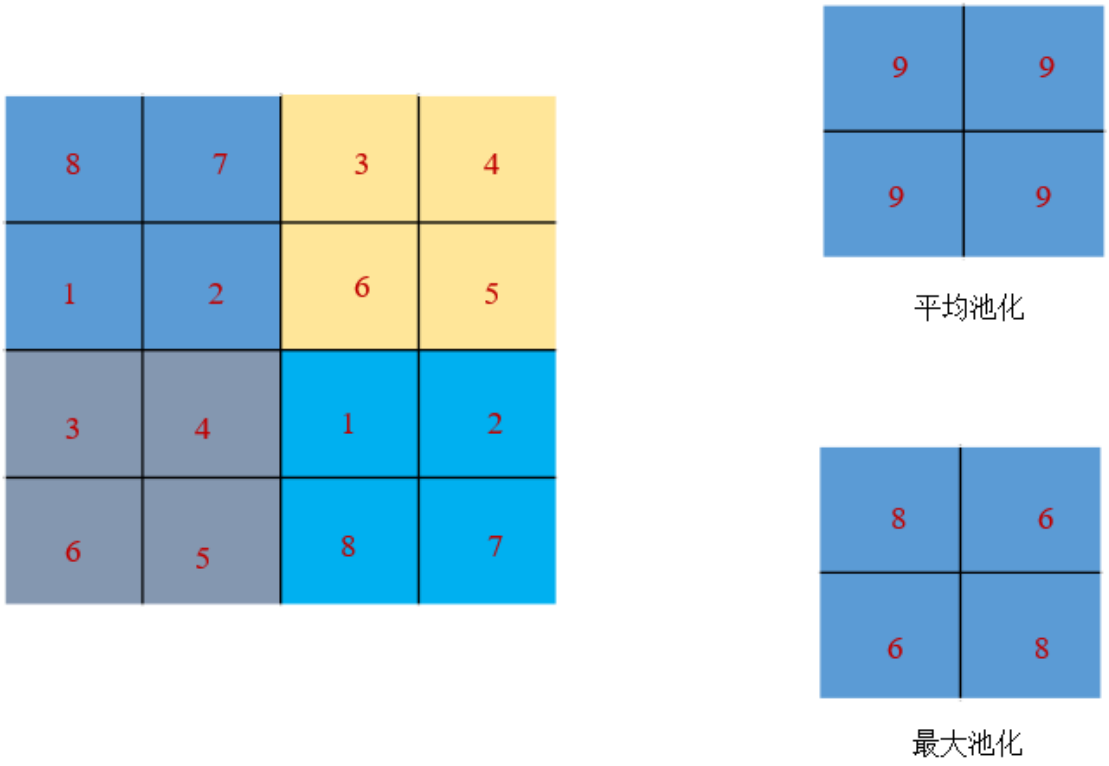


图 2-3 池化过程

2.1.3 激活函数

卷积层和池化层都是线性运算，而实际场景中的输入图像大部分都不是线性可分的，因此激活函数的引入，是为了帮助网络学习到输入图像当中的复杂模式。激活函数主要分为两种，分别是分段式线性函数和非线性函数。

（1）非线性函数

Sigmoid 激活函数是较为普遍的非线性激活函数。Sigmoid 函数的输出值被限制在 0 到 1 之间，所以每个神经元的输出均被归一化。同时由于概率的取值范围

是 0 到 1，所以它也非常适合用于将预测概率作为输出的模型。并且 Sigmoid 函数是可以微分的，也就是说 Sigmoid 曲线上任意两点的的斜率都可以被求出。同时因为梯度较为平滑，可以极大程度上避免输出值跳跃的情况。然而由于 Sigmoid 激活函数输出不是以 0 为中心的，这会导致权重更新的速度与准确率大幅度下降。同时也倾向于梯度消失，并且由于计算 Sigmoid 函数是指数级别，因此运算速度较为缓慢。数学表达式如 2-1 所示：

$$\text{Sigmoid}(x)=\frac{1}{1+e^{-x}} \tag{2-1}$$

Tanh 激活函数属于双曲正切函数，与 Sigmoid 函数的曲线相比，两者有一定程度的相像。但是它比起 Sigmoid 函数有几个优点。对于较为极端的输入，任何激活函数所得到的输出有较小的梯度，这不利于权重更新，这二个激活函数的在这一情况下有不同的输出间隔，Tanh 激活函数以 0 为中心，并且输出间隔为 1，相较于 Sigmoid 激活函数来说更好。对于大多数的二分类问题，Tanh 激活函数被广泛应用于隐藏层，而 Sigmoid 激活函数在输出层使用较多，Tanh 函数数学表达式如 2-2 所示：

$$\text{Tanh}(x)=\frac{2}{1+e^{-2x}}-1 \tag{2-2}$$

（2）分段线性函数

在深度学习领域，ReLU 激活函数是一种使用较为常见的激活函数。当输入数值为正数时，梯度不会出现饱和现象。另外 ReLU 激活函数中计算过程为线性，因此它的计算速度非常快。但是它存在一个缺点，也就是 Dead ReLU，在反向传播过程中，如果输入数值为负数，ReLU 激活函数会失去作用，梯度将变为 0。数学表达式如 2-3 所示：

$$\text{Relu}(x)=\begin{cases} x & x\geq 0 \\ 0 & x<0 \end{cases} \tag{2-3}$$

为了解决 Dead ReLU 问题，Leaky Relu 激活函数被专门设计出来。它修改 x 小于 0 的部分，引入了一个超参数来调整输入数值为负值时所出现的零梯度问题。数学表达式如 2-4 所示：

$$\text{Leaky Relu}(x)=\begin{cases} x & x\geq 0 \\ ax & x<0 \end{cases} \tag{2-4}$$

2.1.4 全连接层

全连接层负责整合前面卷积层和池化层提取到的特征信息，将二维的特征图转变为一维的特征向量，将高层的特征信息提取出来，在整个 CNN 当中起分类的作用。全连接层的结构如图 2-4 所示。

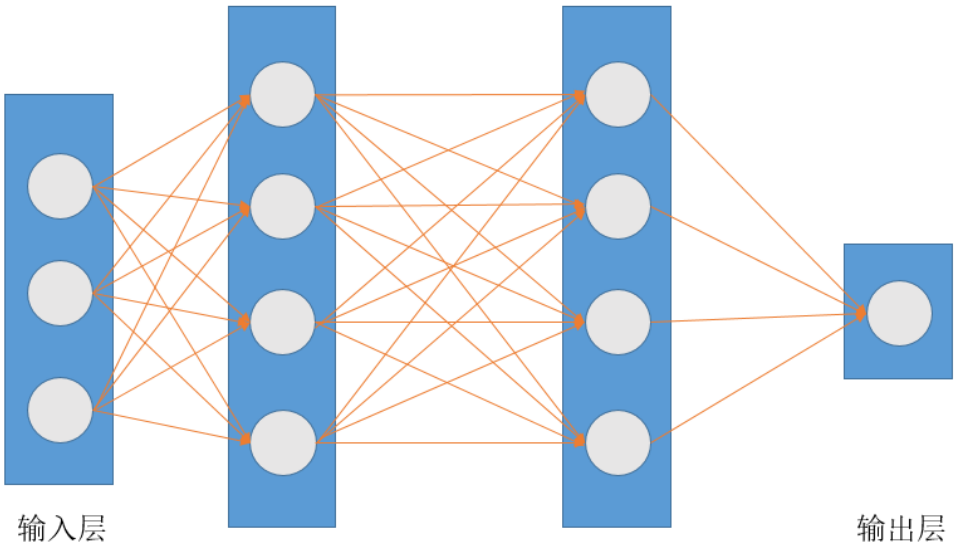


图 2-4 全连接层结构

然而由于全连接层参数及其冗余，可占整个网络参数 80%左右，冗余导致模型过拟合。因此在实际工程当中，全连接层一般被卷积操作所替代，以此来减少模型参数，提高准确率。

2.2 Transformer 概述

Transformer 是一种深度神经网络，基于自注意力机制，同时能够并行化处理数据。最初 Transformer 是自然语言处理领域的一个概念，近一两年被引入到计算机视觉领域。基于此，本小节先介绍注意力机制，然后再叙述原本自然处理领域当中 Transformer 的结构，最后阐述用于处理视觉任务的 Transformer。

2.2.1 注意力机制

注意力机制是指对于任何所需要的模态，无论是文本、图像、乃至点云还是其他，都希望神经网络可以在经过训练之后能自动关注到有意义的位置上，以目标检测和图像分类任务为例，神经网络在经过训练之后可以自动把焦点放在待检测和待分类物体上。

注意力机制可以快速的提取数据的重要特征，因而在机器翻译[60]，语音识别[61]等领域广泛应用。注意力机制可以提高神经网络的可解释性，是一种可以解决多种任务的先进算法。除此之外，它还克服了自然语言处理领域循环神经网络的一些问题，比如性能会随着输入长度的增加而降低，以及不合理的输入顺序使得计算效率较差。这是注意力机制会发展如此快速的几个原因。

除了自然语言处理领域之外，注意力机制在计算机视觉领域早已广泛应用，如 SENet[62]，如图 2-5 所示。SENet 通过 Squeeze-and-Excitation 模块来对注意力权重的概率分布进行计算，然后将其作用于特征图上以此实现重加权每个通道的功能。

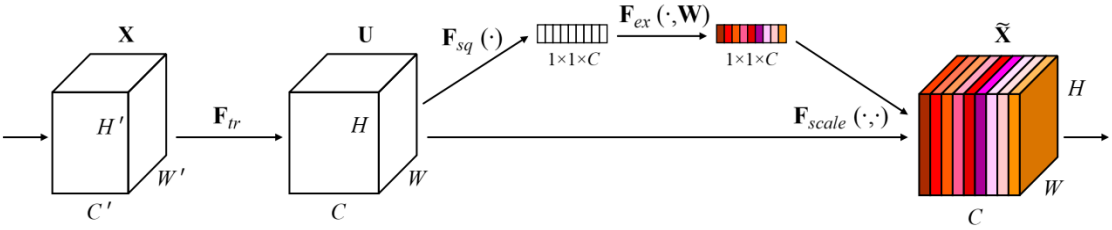


图 2-5 SENet 结构图[62]

自注意力机制[63]是注意力机制的一种改进，网络对外部信息的依赖得以减少，并且它更加擅长捕捉数据之间的相关性。

以一个训练完成的用于分类的神经网络为例，输入一张图片到网络中，然后将网络中的权重 $\mathbf{W}$ 和输入 $\mathbf{X}$ 进行注意力计算，从而在输入中提取有利于神经网络分类的特征，提取到的特征可以作为最终网络判定类别的依据。权重 $\mathbf{W}$ 和输入 $\mathbf{X}$ 都是矩阵，要想实现利用 $\mathbf{W}$ 来重加权 $\mathbf{X}$ 的目的，可以看成先点乘 $\mathbf{W}$ 和 $\mathbf{X}$，计算两者的相似度，然后再将其转换为权重概率分布，最后再作用到 $\mathbf{X}$ 上。计算过程如公式 2-5 所示：

$$\text{Att}(\mathbf{Q},\mathbf{K},\mathbf{V})=\text{Softmax}(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d_k}})\mathbf{V} \tag{2-5}$$

公式 2-5 当中的 $\mathbf{Q}$ 就是上述分类网络当中训练好的 $\mathbf{W}$ 矩阵，而 $\mathbf{K}$ 是图片输入 $\mathbf{X}$，$\mathbf{V}$ 和 $\mathbf{K}$ 相等。公式 2-5 可以解释为先对查询矩阵 $\mathbf{Q}$ 和矩阵 $\mathbf{K}$ 计算相似度，然后再利用 Softmax 算子将其转换为概率分布，然后将得到的概率分布右乘矩阵 $\mathbf{V}$，从而利用注意力权重分布实现对矩阵 $\mathbf{V}$ 的加权。公式 2-5 当中分母 d_k 的平方根的设计是为了避免出现梯度消失的情况，当向量值较大时，Softmax 算子会将几乎全部的概率分布都分配给最大值对应的位置，称为锐化，而通过除以分母可以有效避免梯度消失，从而稳定训练过程。公式 2-5 就是论文[11]提出的缩放点积注意力（Scaled Dot-Product Attention）的计算公式，先利用点乘计算 $\mathbf{QK}$ 两个矩阵的相似

度，除以分母 d_k 平方根进行缩放操作，然后 Softmax 算子再将其转换为概率右乘矩阵 **V**。一般来说 **K** 和 **V** 两个矩阵的维度相同，但是 **Q** 的维度和 **K** 不一定相同。可以通过改变这些维度来控制注意力层的计算复杂度，后续大部分算法都有利用这一点进行改进。

2.2.2 Transformer 结构

Transformer 的提出是为了解决机器翻译任务。机器翻译以理解为序列转序列问题，也就是 seq2seq 结构，对于这类问题一般是采用 encoder-decoder 结构去解决，Transformer 沿用了这种 encoder-decoder 结构。Transformer 完整结构如图 2-6 所示：

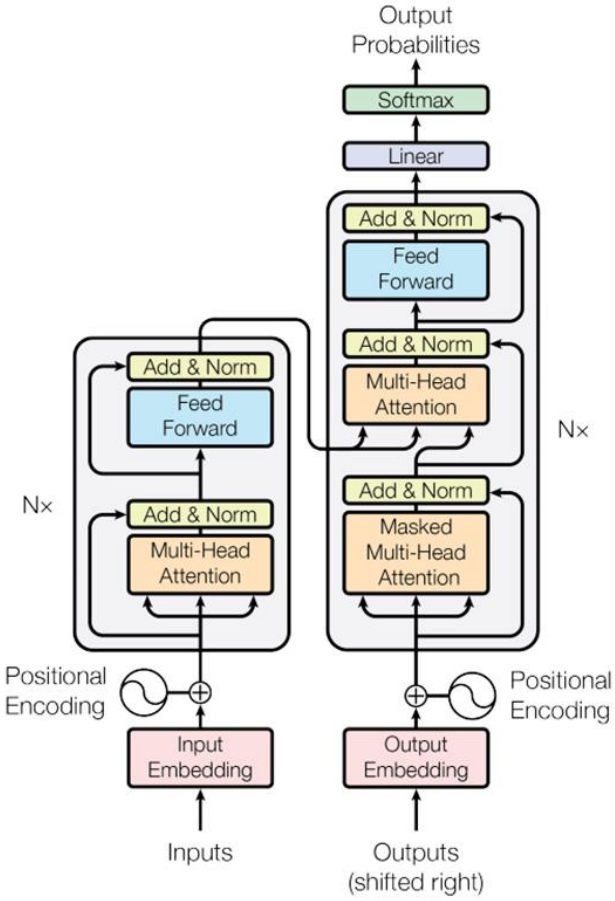


图 2-6 Transformer 结构[11]

Transformer 模型主要包括编码器和解码器，其中编码器包括自注意力模块和前馈神经网络，而解码器的内部结构与编码器类似，但内部多了编码器和解码器两者相交互的交叉注意力模块。普遍来说，标准的 Transformer 模型有 6 个编码器和解码器串行排列。整体流程如下：

（1）首先编码器接收源输入的翻译序列，通过内部的自注意力模块将序列中的必备特征提取出来， 通过前馈神经网络对提取出来的特征进一步处理。

（2）解码器的输入包括两个部分，一个是自注意力模块所提取的目标翻译序列的特征，一个是编码器提取的全局特征，这两个输入进行交叉注意力计算，提取出对目标序列分类的有利特征，然后通过前馈神经网络对特进一步处理。

（3）堆叠多个编码器和解码器，构成串行结构，最后利用解码器输出进行分类即可。

由于本文是基于 Transformer 设计图像的特征提取网络，通常不需要解码器模块，所以只需要关注编码器部分，其中主要是源句子词嵌入（Input Embedding）、多头自注意力（Muti-Head Attention）、位置编码（Positional Encoding）、前馈神经网络（Feed Forward）以及正则化（Norm）、随机失活（Dropout）和残差模块。

（1）源句子词嵌入。机器翻译的输入和输出都是由单词构成句子，将句子编码成程序可以理解的向量就叫做词嵌入，也就是 Word2Vec，对应到图像中则称为 Token 化过程，这是指将图像转换为更具语义的 Token。

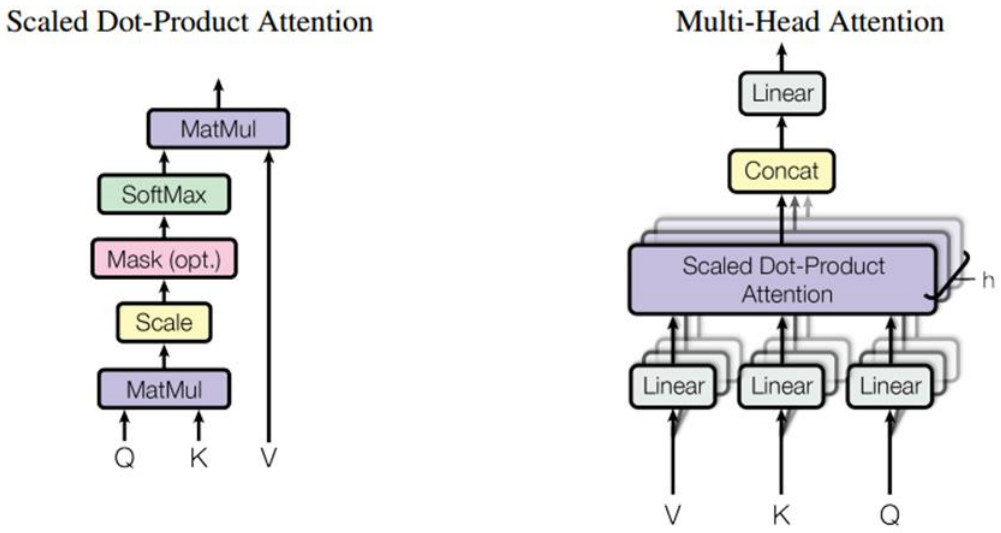


图 2-7 缩放点积注意力与多头注意力[12]

（2）多头自注意力。如图 2-7 所示，左边是缩放点积注意力，右边是多头注意力。为了使注意力提取的特征更加丰富，避免陷入到某种局部特性中，一般会在注意力层基础上引入多个投影头，将 **QKV** 矩阵的特征维度平均切分为若干个部分，每个部分再单独计算自注意力，再拼接得到的计算结果，这样可以使提取的注意力特征更加丰富。

（3）前馈神经网络。前馈神经网络主要是对特征进行变换，单独作用在每个序列上。

（4）Norm、Dropout 和残差。Dropout、残差和 Layer Norm 对整个算法的性能提升非常关键。选择 Layer Norm 而不是 Batch Norm 是机器翻译任务输入的句子不一定是等长的， Batch 训练时会存在大量 Padding 操作，如果在 Batch 这个维度进行 Norm 会出现大量无效统计，使得 Norm 值不稳定，而 Layer Norm 单独计算每个序列，不用考虑 Batch 的影响，比较符合不定长序列任务。如果换成图像分类任务，则可以考虑使用 Batch Norm 层，后续有算法是直接采用 Batch Norm 的。

（5）位置编码。由于每个字符都是单独和全局向量计算相似度，所以自注意力层在计算时不会考虑字符间的顺序，使得 Transformer 具有位置不变性。位置编码的引入就是为了解决这个问题，让模型知道输入语句是有先后顺序的。

2.2.3 视觉 Transformer

ViT 是第一篇将 Transformer 成功引入到视觉领域尝试，如图 2-8 所示。

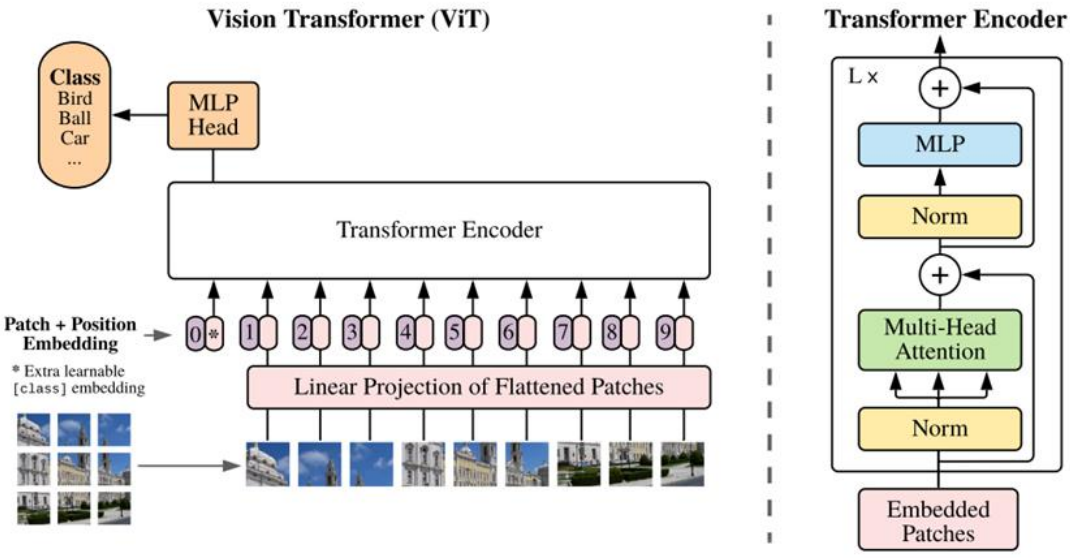


图 2-8 ViT 结构图[12]

整体流程如下：

（1）首先将输入的图片切分为无重叠的固定尺度的 Patch (例如 16x16)，然后将每个 Patch 变为一维向量，这样 n 个 Patch 就相当于自然语言处理领域中的输入序列 (假定输入图片尺寸是 224x224，每个 Patch 大小是 16x16，则 n 是 196)，而一维向量长度与词向量编码长度相等。

（2）由于一维向量维度较大，需要把拉伸之后的 Patch 序列线性投影压缩维度，同时也起到特征变换功能，这个过程称为图片 Token 化。

（3）考虑到后续分类，ViT 还引入一个可学习的 Class Token，将其插入到图片 Token 化后所得序列的最开始位置。

（4）然后将上述序列与可学习的位置编码相加，一起输入到 Transformer 中计算全局注意力和提取特征，Transformer 内部编码器中的多头自注意用于 Patch 间或序列间特征提取，之后的前馈神经网络再对每个 Patch 或者序列进行特征变换。

（5）之后再将最后一个 Transformer 编码器输出序列的第 0 位置(Class Token 位置对应输出)提取出来，用于多层感知机分类。

可以看出，对于图片分类任务而言，无需 Transformer 解码器，且编码器的结构几乎没有改动，只需单独引入一个 Image to Token 操作和 Class Token 即可。

2.3 本章小结

本章首先介绍了后续改进方法中所涉及到的深度学习技术，然后又介绍了注意力机制的原理，Transformer 最初在自然语言处理领域的结构，以及最近一两年引入到视觉领域的 Transformer 结构。本章内容为后续三四章的研究奠定了理论基础。

第三章 Transformer 图像特征提取网络的轻量化

3.1 引言

对许多视觉任务而言，最终的精准程度与时间效率很大程度上取决于负责提取图像特征的网络。因此设计一个良好的图像特征提取网络非常重要。

近一两年来，基于 Transformer 设计图像特征提取网络的工作开始涌现。然而任何基于 Transformer 的模型都存在一个瓶颈，那就是给定 Token 序列作为输入，自注意力机制将序列中的 Token 任意两两之间关联起来迭代学习特征表示，导致模型的时间与空间复杂度与输入的 Token 数量成二次关系。这种二次复杂度阻止了 Transformer 对高分辨率的图像建模，并且高昂的计算成本使其很难适用于边缘设备。

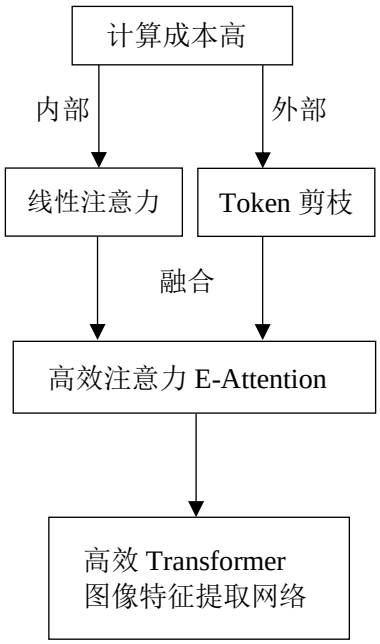


图 3-1 方法框架图

为此，本章提出了基于线性注意力和剪枝的 Transformer 图像特征提取网络，方法框架见图 3-1。该方法首先从外部角度对输入 Token 根据各自对最终用于分类的 Class Token 的重要程度进行评分，然后根据得分采样保留部分 Token，从 Token 维度进行剪枝。然后再从内部角度使用一个组合函数替换计算自注意力矩阵的 Softmax 算子，得到新的线性注意力。最后将两种方法合并得到了一种的高效的注意力机制（E-Attention）。实验结果表明该方法确实能够设计出基于 Transformer 的高效图像特征提取网络。

3.2 基于线性注意力的轻量化

本小节从内部将 Transformer 的注意力机制的复杂度降低为线性，设计了一个组合函数来替代原始的 Softmax 算子。

3.2.1 线性注意力基础

首先从数学角度描述 Transformer 的一般形式，给定输入在嵌入空间中表示为 $\mathbf{X}\in\mathbb{R}^{(n\times d)}$，将输入进入到 Transformer 模块当中所经历的变换记为 T，由此变换 T 定义如公式 3-1 所示：

$$\mathrm{T}(\mathbf{X})=\mathrm{F}(\mathrm{Att}(\mathbf{X})+\mathbf{X}) \tag{3-1}$$

式中 F 是前馈神经网络，包含残差连接；Att 是用来计算自注意矩阵的函数。

当前 Transformer 模型中的自注意力机制是缩放点积注意力，为方便叙述，隐去 Att 的缩放因子，定义如公式 3-2 所示：

$$\mathrm{Att}(\mathbf{Q},\mathbf{K},\mathbf{V})=\mathrm{Softmax}(\mathbf{Q}\mathbf{K}^{\mathrm{T}})\mathbf{V} \tag{3-2}$$

式中 $\mathbf{Q}\in\mathbb{R}^{(n\times d_q)}$、$\mathbf{K}\in\mathbb{R}^{(n\times d_k)}$、$\mathbf{V}\in\mathbb{R}^{(n\times d_v)}$，分别可由输入计算得到。

对公式 3-2 进行深入分析，可以发现制约缩放点积注意力机制的性能，使其复杂度是二次级别的正是公式 3-2 定义当中的 Softmax 算子，因为它的存在，这里才需要先对 $\mathbf{Q}\mathbf{K}^{\mathrm{T}}$ 计算，这一步所得到的是一个 $n\times n$ 的矩阵，从而复杂度是 $\mathrm{O}(n^2)$级别，如果没有 Softmax 算子，那么就是 $\mathbf{Q}\mathbf{K}^{\mathrm{T}}\mathbf{V}$ 三个矩阵连乘，而利用矩阵乘法的结合律，可以先计算后两个矩阵 $\mathbf{K}^{\mathrm{T}}\mathbf{V}$ 的乘积，得到一个 $d_k\times d_v$ 的矩阵，再让矩阵 $\mathbf{Q}$ 去左乘，由于实际情况中 d_k 和 d_v 均远远小于 n，所以整体复杂度可以看成 $\mathrm{O}(n)$级别。Softmax 算子具体定义如公式 3-3 所示：

$$\mathrm{Softmax}(z_i)=\frac{e^{z_i}}{\sum_{c=1}^{C}e^{z_c}} \tag{3-3}$$

式中 z_i 为第 i 个节点的输出值，C 为输出节点的个数，即分类的类别数。

通过 Softmax 算子可以将多分类的输出值转换为范围在 0 到 1 之间，加和为 1 的概率分布。将公式 3-3 引入后，公式 3-2 可改写成如下形式，如公式 3-4 所示：

$$\mathrm{Att}(\mathbf{Q},\mathbf{K},\mathbf{V})_i=\frac{\sum_{j=1}^{n}e^{\mathbf{q}_i^{\mathrm{T}}\mathbf{k}_j}\mathbf{v}_j}{\sum_{j=1}^{n}e^{\mathbf{q}_i^{\mathrm{T}}\mathbf{k}_j}} \tag{3-4}$$

式中 $\mathbf{q}_i$ 为 $\mathbf{Q}$ 的第 i 列向量，$\mathbf{k}_j$ 为 $\mathbf{K}$ 的第 j 列向量，$\mathbf{v}_j$ 为 $\mathbf{V}$ 的第 j 列向量。

根据公式 3-4，提出注意力机制的一般化形式，将式中指数形式换成关于 $\mathbf{q}_i$ 和 $\mathbf{k}_j$ 的一般函数 S(·)，具体定义如公式 3-5 所示：

$$\mathrm{Att}(\mathbf{Q},\mathbf{K},\mathbf{V})_i = \frac{\sum_{j=1}^{n} \mathrm{S}(\mathbf{q}_i,\mathbf{k}_j)\mathbf{v}_j}{\sum_{j=1}^{n} \mathrm{S}(\mathbf{q}_i,\mathbf{k}_j)} \tag{3-5}$$

为了保留注意力机制的相似特性，这里需要要求 S(·)≥0 恒成立。这种一般形式的注意力也被称为 Non-Local 网络[64]。

想要将注意力的复杂度降到最理想的线性级别，就需要一种可分解的计算方式来有效近似相似性函数 S(·)。继而可以利用矩阵乘法的结合律，先计算后两个矩阵的乘积，将复杂度降低为线性 O(*n*)级别，如图 3-2 所示。

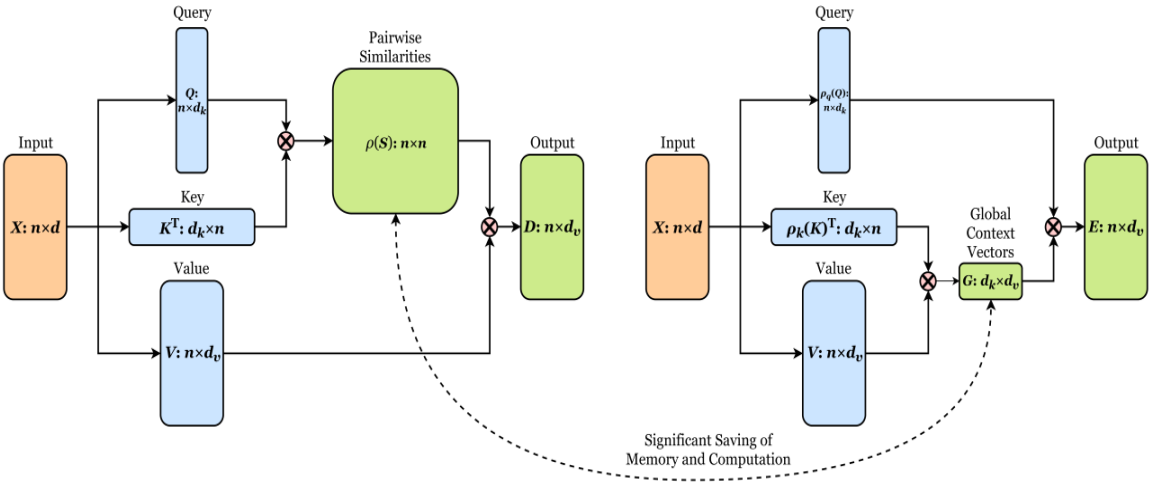


图 3-2 传统自注意力与线性自注意力[65]

3.2.2 现状分析

正如上文所述，要想实现线性注意力机制的关键是在于能够找到一个可分解的相似度函数 S(·)。

目前学术界所存在的大多数线性注意力机制都试图近似估计 Softmax 算子。例如，RFA[66]利用随机傅里叶特征定理，Performer[67]利用正随机特征来近似 Softmax 算子。然而根据经验发现，这些方法对采样率的选择很敏感，如果采样率过高，就会变得不稳定。因为这些方法仅在约束的理论范围内通过使用 Softmax 算子的有效近似来实现线性注意力机制，因此，当对应的假设不满足或近似误差累积时，这些方法可能并不总是优于普通结构。

因此考虑能否在保证实验效果的前提下，用一个可分解的相似度函数去直接替代 Softmax 算子。这需要明确当前注意力机制中的关键特性，由此设计出一个满足要求的可分解的相似度函数，以便实现线性注意力。

3.2.3 替代函数设计

通过深入阅读文献研究，总结了两个影响当前注意力机制性能的关键特性，分别是注意力矩阵中的非负元素[68-69]和非线性重加权方案[70-72]。

首先注意力矩阵中只保留正值，忽略了具有负相关性的特征，从而有效地避免聚合不相关的上下文信息。其次非线性重加权机制可以集中注意力权重的分布，从而稳定训练过程，也可以帮助模型将局部性放大，这种局部性是指很大一部分上下文依赖来自邻近的标记[73-74]。

结合上述结论，本小节提出一种用于替换 Softmax 的组合函数，这个组合函数满足以上两种特性，由两个子函数组成，分别用来实现非负性和非线性重加权，如公式 3-6 所示：

$$\mathrm{Att}(\mathbf{Q},\mathbf{K},\mathbf{V}) = g(f(\mathbf{Q},\mathbf{K}))\mathbf{V} \tag{3-6}$$

式中 $f(\cdot)$和 $g(\cdot)$都是自定义的两个函数，用于保证上述两个特性，定义如公式 3-7 和公式 3-8 所示：

$$f(x) = \begin{cases} x+1 & x \geq 0 \\ e^{x} & x < 0 \end{cases} \tag{3-7}$$

$$g(\mathbf{q}_i, \mathbf{k}_j) = \mathbf{q}_i \mathbf{k}_j^{\mathrm{T}} \sin\left(\frac{\pi(i+n-j)}{2n}\right) \tag{3-8}$$

式中 i，j=1，…，n。

公式 3-7 用来保证注意力矩阵的非负性。公式 3-8 会设计成这种形式有以下几个原因：

（1）实现局部性偏差，当 i 和 j 相近时，对应三角函数值接近于 1，当 i 和 j 较远时，两者之差接近于 n，对应三角函数值接近于 0，从而对应的相似度函数可忽略不计；

（2）三角函数本身非线性，这样设计可以集中注意力权重的分布，从而达到稳定训练过程的目的；

（3）公式 3-8 可以通过和差化积公式将其分解，从而能够利用矩阵乘法的结合律，以便将注意力机制的复杂度降低为线性。

下述为用于满足非线性重加权特性的子函数 $g(\cdot)$的拆解过程：

$$
\begin{aligned}
g(\mathbf{q}'_i,\mathbf{k}'_j) &= \mathbf{q}'_i\mathbf{k}'^{\mathrm{T}}_j \sin(\frac{\pi(i+n-j)}{2n}) \\
&= \mathbf{q}'_i\mathbf{k}'^{\mathrm{T}}_j \sin(\frac{\pi(i-j)}{2n}+\frac{\pi}{2}) \\
&= \mathbf{q}'_i\mathbf{k}'^{\mathrm{T}}_j (\cos(\frac{\pi i}{2n})\cos(\frac{\pi j}{2n})+\sin(\frac{\pi i}{2n})\sin(\frac{\pi j}{2n})) \\
&= (\mathbf{q}'_i\cos(\frac{\pi i}{2n}))(\mathbf{k}'^{\mathrm{T}}_j\cos(\frac{\pi j}{2n}))+(\mathbf{q}'_i\sin(\frac{\pi i}{2n}))(\mathbf{k}'^{\mathrm{T}}_j\sin(\frac{\pi j}{2n}))
\end{aligned}
$$

其中 $\mathbf{q}'_i = f(\mathbf{q}_i)$，$\mathbf{k}'_j = f(\mathbf{k}_j)$，$f(\cdot)$如公式 3-7 所示。

令 $\mathbf{q}_i^{\cos} = \mathbf{q}'_i\cos(\frac{\pi i}{2n})$，$\mathbf{q}_i^{\sin} = \mathbf{q}'_i\sin(\frac{\pi i}{2n})$，$\mathbf{k}_j^{\cos} = \mathbf{k}'^{\mathrm{T}}_j\cos(\frac{\pi j}{2n})$ 以及 $\mathbf{k}_j^{\sin} = \mathbf{k}'^{\mathrm{T}}_j\sin(\frac{\pi j}{2n})$。显然 $\mathbf{q}_i^{\sin}$ 和 $\mathbf{q}_i^{\cos}$ 为 $\mathbf{Q}$ 的第 i 列向量 $\mathbf{q}_i$ 经过自定义非负函数 $f(\cdot)$与正弦或余弦函数变换而来，记由此变换之后的 $\mathbf{Q}$ 为 $\mathbf{Q}^{\cos}$ 和 $\mathbf{Q}^{\sin}$。同理，记变换之后的 $\mathbf{K}$ 为 $\mathbf{K}^{\cos}$ 和 $\mathbf{K}^{\sin}$。从而得到线性分解之后的最终表达，如公式 3-9 所示：

$$
\mathrm{Att}(\mathbf{Q},\mathbf{K},\mathbf{V}) = g(f(\mathbf{Q},\mathbf{K}))\mathbf{V} = \mathbf{Q}^{\cos}(\mathbf{K}^{\cos}\mathbf{V}) + \mathbf{Q}^{\sin}(\mathbf{K}^{\sin}\mathbf{V}) \tag{3-9}
$$

本小节提出的注意力计算公式 3-9 与原始的缩放点积注意力计算公式 3-2 相比，能够利用矩阵乘法的结合律，使注意力机制的计算复杂度降低线性，两者计算流程如图 3-3 所示。

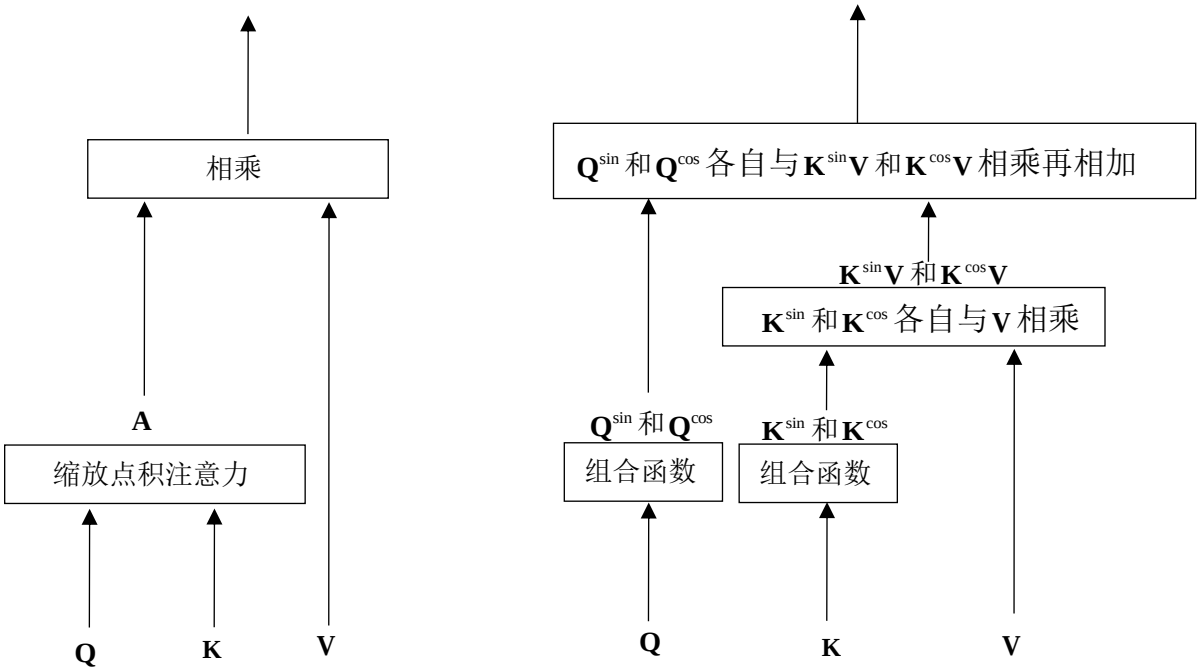


图 3-3 缩放点积注意力与线性注意力计算流程图

3.3 基于 Token 剪枝的轻量化

本小节从 Token 维度对基于 Transformer 所设计的图像特征提取网络进行剪枝从而降低计算成本。先对不同的输入图像转换而来的 Token 估算出各自的评分，然后再通过采样保留部分 Token。

3.3.1 现状分析

目前许多对基于 Transformer 的图像特征提取网络从 Token 维度进行剪枝的方法都是引入额外的神经网络进行训练，从而计算 Token 得分，并以此为依据进一步判断哪些 Token 是冗余的，而哪些 Token 是需要保留的。比如 Dynamic ViT[75]就是如此。

然而这一类方法在每个阶段都有一个固定的比率来减少 Token，尽管这样的确从外部角度减少了基于 Transformer 所设计的图像特征提取网络的计算量，但也引入了额外的计算开销，也就是评分网络本身，它需要与 Transformer 所设计的图像特征提取网络一起训练，需要添加额外的超参数和损失项来修改损失函数；以及它的另一个限制是：如果需要改变固定的剪枝 Token 比率，那么当需要部署在不同的边缘设备上时，网络就需要重新训练，这极大程度上限制了模型的应用场景。

因此本小节首先通过无参数的方法来计算得到输入的各个 Token 的评分，以此避免引入额外的参数，平白增加计算量；同时也由于无参数的良好特性，本小节设计的剪枝方法可以作为即插即用的模块引入到任何现成的经过预训练的基于 Transformer 所设计的图像特征提取网络当中。

另外在现实情况中，输入图像的部分信息可能冗余，这些信息并不能作为判别图像类别的依据，而这些信息的数量取决于图片自身。如果每个阶段的剪枝比率是固定的，那么为了实现计算量的减少，一方面如果剪枝比率过大可能会不自觉丢掉一部分重要信息，导致分类精度的降低；另外一方面如果剪枝比率过小，那么还有部分冗余 Token 保留，这样也会导致计算资源的浪费。而自适应的根据不同输入图片保留不同 Token 可以很好的解决这几点限制，它可以根据实际输入图像的不同，以采样的方式动态保留剩余 Token 的数量，以此适应不同的实际应用场景。

总体来说，本小节设计的从 Token 维度对基于 Transformer 所设计的图像特征提取网络进行剪枝的方法有两个特点：无参数评估 Token 得分；以及根据输入图片自适应采样，从而动态保留 Token。

3.3.2 评分

绕过训练额外参数得到输入 Token 的得分这一条路之后，可以将着眼点放在基于 Transformer 所设计的图像特征提取网络本身已有的信息上面。模型内部自注意力层输入的所有 Token 分为两部分，其中第一部分只包括第一个 Class Token，这是额外引入用于在最后阶段判断输入图像的类别，它放在 Token 序列的第一个位置。第二部分就是剩余的所有 Token，这是由输入图片切分并经过变换而得到的。

要从 Token 维度对基于 Transformer 所设计的图像特征提取网络进行剪枝，就是尽可能减少第二部分那些由图片转换而来的 Token，将这部分 Token 数量记为 m。也就是说输入有 m+1 个 Token，输出后有 r'+1 个 Token，这里的 r' 是经过剪枝之后保留下来的第二部分的 Token 数量，对应的数据范围是 $r' \le R \le m$。这里 R 是一个事先给定的参数，作用是控制经过采样之后保留下来的 Token 的最大数量。

在标准的自注意力层当中 $\mathbf{Q} \in \mathbb{R}^{((m+1)\times d_q)}$、$\mathbf{K} \in \mathbb{R}^{((m+1)\times d_k)}$、$\mathbf{V} \in \mathbb{R}^{((m+1)\times d_v)}$，是由输入 Token 计算而得，将其记为 $\mathbf{X} \in \mathbb{R}^{((m+1)\times d)}$。而自注意力矩阵 $\mathbf{A}$ 就是由 $\mathbf{Q}$ 和 $\mathbf{K}^{\mathrm{T}}$ 两者计算所得，需要注意此时注意力计算方式仍然是标准的缩放点积自注意力。注意力矩阵本身就表示输入的所有 Token 两两之间的相似性，或者说相关联程度。

比如说自注意力矩阵 $\mathbf{A}$ 的第 i 行第 j 列或者说第 j 行第 i 列元素就表示当前所输入的所有 Token 之中第 i 个 Token 和第 j 个 Token 之间的相关联程度，数值越大，代表两者越关联，说明彼此对另外一个 Token 越重要。

由此可以得到启发，既然当前所输入的 Token 序列的第一个 Class Token 用于最终阶段的分类，那么在自注意力矩阵当中，如果其他 Token 与第一个用于分类的 Class Token 关联程度越高，说明这些 Token 至少在当前阶段对于判别输入图像类别的 Class Token 越重要，而用于衡量这些 Token 与第一个 Class Token 的相似程度的就是当前自注意力矩阵的第一行中除去第一个之外的所有数值，当然也可以说是当前自注意力矩阵的第一列中除去第一个之外的所有数值，这两者是等价的。因此在设计用于计算除去 Class Token 以外的各个 Token 重要性的分数公式时，采用的计算方法如公式 3-10 所示：

$$h_j = \frac{a_{1,j}}{\sum_{i=2}^{m} a_{1,i}} \tag{3-10}$$

式中 $a_{1,j}$ 和 $a_{1,i}$ 表示自注意力矩阵 $\mathbf{A}$ 第 1 行第 j 列，第 1 行第 i 列的元素，h_j 为第 j 个 Token 的得分。

另外对于多头注意力层而言，可以先各自计算每个头部的得分，最后再将所有头部相加。

3.3.3 采样

3.3.2 小节已经通过无参数的方式得到了基于 Transformer 的图像特征提取网络当中除去用于分类的 Class Token 以外的各个 Token 重要性得分，现在可以根据这些得分，从自注意矩阵中删除部分 Token。可以很自然的想到先筛选出得分最高的一些 Token，将其保留，然后直接去掉剩余的得分较低的 Token。

但这个方法有两个问题：首先是筛去 Token 的数量不好确定，很难做到根据不同输入图片自适应动态变化；其次是如果一开始，就在某个阶段就把分数较低的 Token 直接筛掉了，那这些被筛掉的 Token 对于最后的分类来说并不是一定不重要。正如基于 CNN 的图像特征提取网络在初期提取的只是边缘，纹理等较为浅显的特征，但在后期阶段所提取到的就是较为高级的语义特征。不同的 Token 在不同的阶段可能有不同的作用，表示不同的意义，目前得到的分数只是在当前这一个阶段所得到的，如果只是简单根据某个中间阶段作用效果较低就将其粗暴删掉，那么被删除的 Token 就无法进入后续阶段，但它可能在后面某个阶段有重要的作用，这样的话反而会影响最后的实验结果。

因此可以考虑根据这些分数，通过采样的方式随机选取部分 Token 来保留。如果分数较低，它在目前这一阶段就有更低的概率被选中，从而无法得以保留，反之亦然。从实验结果来看，这种方法确实在实验效果上要好于粗暴删掉得分较低的 Token，说明这种随机性在一定程度上可以抵消直接删掉 Token 的不足，具体实验数据详见 3.5.6 小节的实验部分。

然后需要根据这些 Token 的分数来采样，使得分数较高的 Token 有更大的概率被保留，分数较低的 Token 有更小的概率被保留。也就是要找到一个服从这些得分的概率分布，再根据这个概率分布去采样。可以计算这些分数对应的累积分布函数，将其逆变换，从而得到服从该随机分布的反函数。

首先累积分布函数（Cumulative Distribution Function,CDF），又叫做分布函数，是概率密度函数的积分，可以完整描述一个随机变量 x 的概率分布，将这个概率分布记为 X。对于所有实数 x，累积分布函数定义[77]如公式 3-11 所示：

$$\mathrm{F_X}(x) = \mathrm{P}(\mathrm{X} \le x) \tag{3-11}$$

若累积分布函数 F 是连续的严格增函数，那么存在对应的反函数 $\mathrm{F}^{-1}(y)$，其中 y 在 0 到 1 之间。累积分布函数的反函数可以用来生成服从该随机分布的随机变量。也就是设若 $\mathrm{F_X}(x)$ 是概率分布 X 的累积分布函数，并存在反函数 $\mathrm{F_X}^{-1}$，若 a 是 [0,1)区间上均匀分布的随机变量，则 $\mathrm{F_X}^{-1}(\mathrm{a})$ 服从 X 分布。因此，可以根据下列公式计算累积分布函数，如公式 3-12 所示：

$$\mathrm{CDF}_i = \sum_{j=2}^{i} h_j \tag{3-12}$$

这里注意是从第二个 Token 开始累加的，因为第一个分类 Class Token 必须保留，在得到累积分布函数 CDF 之后，可以根据其逆形式得到采样函数，如公式 3-13 所示：

$$\eta(v) = \mathrm{CDF}^{-1}(v) \tag{3-13}$$

式中 v 在 0 到 1 之间。

具体来说，本小节的采样策略就是从均匀分布 0 到 1 之间随机选取一个数 v，然后计算得到对应的 $\eta(v)$，然后选择最近的整数作来抽样。这个操作需要执行固定的 R 次，在这 R 次采样中，可能会出现一个 Token 被多次选中的情况，这样实际采样得到的 Token 数量 r' 小于等于固定的 R。

在采样得到待保留的 Token 之后，最初的注意力矩阵 A 由最初的 m+1 行变为现在的 r'+1 行，然后再加入接下来的计算流程当中。

3.4 基于线性注意力与 Token 剪枝的轻量化

这一小节考虑将 3.2 小节提出的线性注意力机制与 3.3 小节提出的 Token 剪枝模块两者相结合，乍看两者似乎无法结合，因为线性注意力机制的核心是利用矩阵乘法的结合律先计算 $\mathbf{QK}^{\mathrm{T}}\mathbf{V}$ 三个矩阵当中后两个矩阵 $\mathbf{K}^{\mathrm{T}}\mathbf{V}$ 的乘积，然后再左乘矩阵 $\mathbf{Q}$。而 Token 剪枝模块当中对输入图像转换而来的 Token 评分利用的则是 $\mathbf{QK}^{\mathrm{T}}$ 两个矩阵相乘经过非线性变换之后的注意力矩阵。这两者本质上是冲突的。

但由于评分利用的是第一个用于分类的 Class Token 与其他 Token 之间的关联程度，也就是自注意力矩阵的第一行除去第一个之外的所有数值，当然也可以说是自注意力矩阵的第一列除去第一个之外的所有数值，这两者是等价的。比如注意力矩阵第二行第一列的元素就表示第二个 Token 与 Class Token 的关联程度，而这个元素是由矩阵 $\mathbf{Q}$ 的第二行的所有元素与矩阵 $\mathbf{K}^{\mathrm{T}}$ 第一列的所有元素对应相乘再相加得到的，由此得到启发，可以在线性注意力机制计算 $\mathbf{QK}^{\mathrm{T}}\mathbf{V}$ 三个矩阵当中后两个矩阵 $\mathbf{K}^{\mathrm{T}}\mathbf{V}$ 的乘积时，先保留矩阵 $\mathbf{K}^{\mathrm{T}}$ 的第一列元素，然后再额外将矩阵 $\mathbf{Q}$ 除去第一行之外的每一行与矩阵 $\mathbf{K}^{\mathrm{T}}$ 的第一列对应相乘再相加，以此在保证线性注意力机制的同时，还能够额外计算用于分类的 Class Token 与其他 Token 之间的关联程度，而这个计算过程本身是线性的，因此不会增加太多计算量。

在采样得到待保留的 Token 之后，3.3 小节的 Token 剪枝方法是将 $\mathbf{QK}^{\mathrm{T}}$ 计算得到的注意力矩阵 $\mathbf{A}$ 由最初的 m+1 行变为 r'+1 行，这里可以改为将矩阵 $\mathbf{Q}$ 由最初

的 m+1 行变为现在的 r'+1 行，然后再加入接下来的计算流程当中。两个方法融合之后的注意力计算方式如公式 3-14 所示：

$$\mathrm{Att}(\mathbf{Q}',\mathbf{K},\mathbf{V}) = g(f(\mathbf{Q}',\mathbf{K}))\mathbf{V} = \mathbf{Q}'^{\cos}(\mathbf{K}^{\cos}\mathbf{V}) + \mathbf{Q}'^{\sin}(\mathbf{K}^{\sin}\mathbf{V}) \tag{3-14}$$

线性注意力机制和 Token 剪枝方法融合之后得到了更为高效的注意力机制（Efficient Attention, E-Attention）。计算流程如图 3-4 所示。

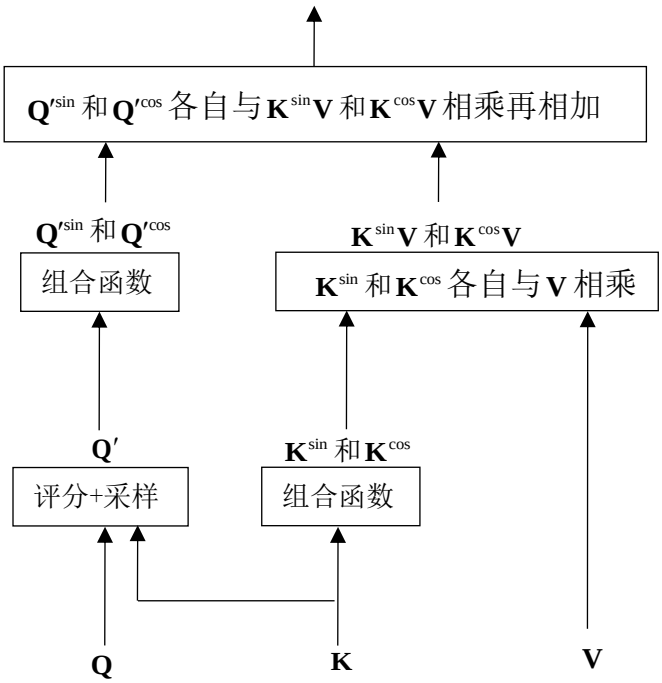


图 3-4 高效注意力计算流程图

3.5 实验与结果分析

本章提出了线性注意力机制以及 Token 剪枝两个方法，分别从内部和外部加速 Transformer，最后又将这两个方法融合，得到了 E-Attention。本小节首先对这几个轻量化方法进行内部实验探究，然后又引入了多个 Transformer 模型作为图像特征提取部分在图像分类和目标检测两大类任务上训练，得到原始的实验结果，最后再去验证单独引入线性注意力机制、Token 剪枝或者 E-Attention 之后的 Transformer 模型在分类与检测任务上的实验结果，并对其进行评估与分析。

3.5.1 数据集介绍

本章基于在图像分类领域应用较为广泛的 ImageNet1k 数据集和目标检测领域使用较为普遍的 COCO 数据集进行对比实验的验证。

ImageNet 数据集是当前人工智能图像领域应用较多的一个数据集，关于图像定位、分类、检测等工作大多基于此数据集研究。它在计算机视觉领域中应用非常广泛，由斯坦福大学的团队维护，使用方便。ImageNet 包含 1400 多万张图片，有 2 万多个分类类别。ISLVRC 竞赛使用的是轻量版的 ImageNet 数据集。在一些论文中，这个轻量版的数据被叫成 ImageNet 1K，表示有 1000 个类别。

COCO 数据集是由微软所维护的一个大型的检测分割数据集，主要在复杂的日常场景当中获取。COCO 数据集主要解决目标之间的上下文关系，目标检测与二维上的精确定位这三个问题。它包含 91 个类别，虽然类别数量比 ImageNet 数据集少很多，但是它里面每一个类别所包含的图片数量非常多，可以得到更多的每个类别当中的某种特定场景。它包含 20 万个图片，91 个类别当中有 80 个类别多于 50 万标注，它可以说是最广泛的公共目标检测数据集。COCO 包含 20G 的图片和 500M 的标签，训练集，测试集和验证集的比例为 2：1：1。

3.5.2 评估指标

在图像分类和目标检测领域当中，有较多类型的指标可以采用以便更好的评估模型。由于本章所提出的方法都是为了加速模型，需要考量模型的复杂度，速度，因此选取了 Params，FLOPs 和 FPS 这三个指标。另外一方面为了评估模型本身的性能优劣，也选取了 Top-1 Acc 和 mAP 这两个指标。

本章总共选取这五个指标来评估本章所提出的方法的优劣，接下来在介绍这五个指标之前先引入一些基本概念：

（1）模型的预测值是正例：正样本

（2）模型的预测值是负例：负样本

（3）模型预测为正样本，真实值为正样本：TP

（4）模型预测为负样本，真实值为负样本：TN

（5）模型预测为负样本，真实值为正样本：FN

（6）模型预测为正样本，真实值为负样本：FP

（7）召回率（Recall）：从真实的样本集的角度来统计的，在总的正样本中，模型找回了多少个正样本，如公式 3-15 所示：

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{3-15}$$

（8）精准率（Precision）：从预测结果的角度来统计的，预测为正样本的数据中，有多少个是真正的正样本，如公式 3-16 所示：

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{3-16}$$

（9）PR 曲线：顾名思义就是精准率（Precision）和召回率（Recall）的曲线，PR 曲线中精准率（Precision）为纵坐标，召回率（Recall）为横坐标。

（10）平均准确率（AP）：也就是 PR 曲线下面的面积。AP 越大，表明模型的平均准确率越高。

A. 图像分类指标

在计算机视觉领域的图像分类任务当中，评估一个分类模型的好坏主要是通过这个分类模型识别出来的准确率与错误率去评判。其中错误率包括 Top-5 error 和 Top-1 error 这两个，而准确率包括 Top-5 accuracy 和 Top-1 accuracy 这两个。本章选取 Top-1 accuracy 这个指标去检验引入线性注意力机制和 Token 剪枝这两个方法之后，模型分类的准确性。另外一个方面，本章选取 FLOPs 指标去描述模型的计算量。

（1）Top-1 accuracy：表示取最终得到的概率向量当中最大的一个作为预测结果。

（2）浮点数计算量（FLOPs）：是用来衡量模型整体计算复杂度的指标，主要统计模型架构中加法和乘法的计算次数。

B. 目标检测指标

在目标检测任务当中，去评估一个检测模型的好坏主要是通过这个检测模型 mAP 值，另外一个方面，当使用各种方法去加速检测模型，降低其复杂度的时候，对应的性能指标一般是用 FPS 和 Params。

（1）平均精度均值（mAP）：在目标检测任务中，一个目标检测模型通常会对较多种物体进行检测，从而每一类物体都能得到对应的 PR 曲线，最后计算出对应的 AP。这些类别的 AP 值的平均就是 mAP。它所衡量的是模型在所有类别上的优劣，是目标检测任务中非常有意义的一个指标。

（2）参数总量（Parameter）：表示模型在运行时消耗内存占用存储空间的大小。

（3）每秒帧数（FPS）：表示模型在进行实时检测任务中，每一秒能够检测多少张图片。

3.5.3 实验环境与超参数设置

本章实验使用 VS-code 编辑器，采用 Python3.6 版本部署实验，系统环境为 Windows 10 教育版 64 位系统，内存为 32GB，选择 Pytorch 作为神经网络库，使用 Nvidia GeForce RTX 2080Ti 11GB 作为验证模型的实验环境所用的图形处理器，采用 Intel(R) Core(TM)i7- 9700K 为处理器。

另外所有实验模型的超参数设置采用模型 DeiT 所提供的超参数。它可以在不改变 ViT 模型结构的前提下实现涨点。Epochs 为 300，Batch size 为 1024，基础学习率为 0.0005，优化器选择 AdamW，学习率衰减策略使用 Cosine，权重衰减为 0.05，Dropout 为 0.1，Warmup epochs 为 5，剪枝方法固定起始参数设为初始阶段输入 Token 数量的 70%。

3.5.4 基准模型

实验选取 DeiT，PVT，Swin Transformer，TNT[77]，T2T-ViT 和 CaiT[78]这六个模型作为验证本章提出的创新点的基准模型，并且在具体实验中，也对这六个模型的不同尺寸各自做了对比实验，从而进一步分析这三个创新点的效果。这六个模型都是基于 Transformer 从不同角度做了改进，接下来分别对这六个模型进行简要说明。

（1）DeiT：模型 DeiT 的提出是为了缓解传统 Transformer 模型的局限性，即只有在包含了三百万张图片的超大数据集 JFT-300 上才能有较好的性能与泛化。DeiT 的作者结合蒸馏的轻量化操作提出了一种新的训练方案，通过引入蒸馏 Token 使得学生通过注意力向老师学习，并且提出了一种特定于 Transformer 的师生策略。DeiT 模型只在 ImageNet 上训练的情况下，就得到了一个性能较好的无卷积 Transformer。DeiT 在训练中推理时，将额外引入的用于分类的 Class Token 和蒸馏 Token 的预测向量求平均之后再转换为概率分布。

（2）PVT：模型 PVT 的提出是为了克服传统视觉 Transformer 模型应用于不同密集预测任务时所遇到的困难，它是第一个可以应对不同分辨率密集预测任务的视觉 Transformer 模型。相较于以往计算消耗较高的 ViT 而言，它可以利用渐进式缩小金字塔来降低计算成本。

（3）Swin Transformer：模型 Swin Transformer 提出了分层的 Transformer，利用移动窗口来计算特征表示，以此来解决视觉实体分辨率的巨大差异。模型内部设计时先事先划分好互不重叠的局部窗口，将自注意力的计算限制于其中，并且结合跨窗口连接，提高模型效率。这种设计能够建模不同分辨率的图像，并且大幅度降低计算消耗成本。

（4）TNT：模型 TNT 的提出是因为传统的 Transformer 模型忽略了由输入图片转化而来的 Patch 的内部固有信息，因此 TNT 同时建模了 Patch 级别和像素级别的特征表示。

（5）T2T-ViT：模型 T2T-ViT 的提出是因为 ViT 无法建模相邻像素之间的边缘，线条等重要局部结构。T2T-ViT 通过将相邻对象逐层递归聚集，使得多个相邻 Token

聚集为一个 Token，从而实现建模 Token 周围的局部结构，同时也减少了 Token 序列的长度，降低了计算消耗。

（6）CaiT： CaiT 使更深的 Transformer 易于收敛，并能提高精度。并且提出了一种新的高效的处理分类 Token 的方式。进行了两次 Transformer 体系结构的更改，显著提高了深度 Transformer 的精度。

3.5.5 线性注意力内部实验探究

（1）不同的线性注意力比较

在这一小节，首先使用不同的线性注意力模型在 ImageNet1k 数据集进行图像分类实验，以此对比各个线性注意力机制的优劣，结果如表 3-1 所示。

表 3-1 线性注意力机制对比

不同线性注意力	复杂度	FLOPs	Top-1 ACC
DeiT-S（基准模型）	$O(n^2)$	4.6	79.8
Linformer	O(n)	2.4	78.7
Performer	O(n)	2.6	76.8
Nyströmformer	O(n)	2.4	79.3
Ours	O(n)	2.3	78.8

表 3-1 对比了本章所提的线性注意力机制与学术界目前所提出来的线性注意力机制的性能，从中可以看出相比原始 Transformer 模型 DeiT-S，线性 Transformer 能够大幅降低前向推理的计算量FLOPs，所有线性注意力机制都提升了50%左右。并且从表 3-1 还可以看出，本章所提出的线性注意力机制在 FLOPs 这一项指标上面提升最多，优于学术界其他线性注意力机制，在准确率方面分类精度仅次于 Nyströmformer，综合两项指标来看，本章所提出的线性注意力机制与同类型线性注意力机制相比占较大优势。

（2）不同关键特性比较

本章所提出的线性注意力机制的改进是来源于影响 Softmax 注意力性能的两个关键特性：注意力矩阵元素非负以及非线性重加权。本章所提出的方法是在保证这两点特性的基础上所完成的，因此本小节针对线性注意力机制的改进，验证这两点特性各自对于最终的实验效果有多大的提升。

首先考虑只保留注意力矩阵的非负性，并且舍弃非线性重加权方案，因此将公式 3-17 改为公式 3-18，从而得到仅保留注意力矩阵的非负性这一特性的新的注意力机制记为 Ours-A。

$$g(\mathbf{q}_i,\mathbf{k}_j)=\mathbf{q}_i\mathbf{k}_j^{\mathrm{T}}\sin(\frac{\pi(i+n-j)}{2n}) \tag{3-17}$$

$$g(\mathbf{q}_i,\mathbf{k}_j)=\mathbf{q}_i\mathbf{k}_j^{\mathrm{T}} \tag{3-18}$$

其次考虑只保留非线性重加权方案，并且舍弃注意力矩阵的非负性，因此将公式 3-19 改为公式 3-20，从而得到仅保留非线性重加权方案这一特性的新的注意力机制 Ours-B。

$$f(x)=\begin{cases}x+1 & x\geq 0\\ e^{x} & x<0\end{cases} \tag{3-19}$$

$$f(x)=x \tag{3-20}$$

最后将这两种新得到的仅保留了一项特性的注意力机制与本章所提出的保留了两项特性的注意力机制去对比，用这三种注意力机制去替换原始的 Transformer 模型 DeiT-S，并将由此得到的三种新的 Transformer 模型与最开始未做任何改变的 Transformer 模型在 ImageNet1k 数据集上做图像分类的对比实验，结果如表 3-2 所示。可以看出，如果只保留一种特性而得到的线性注意力机制，准确率远远不如保留两种特性的线性注意力机制。另外一方面，哪怕只保留一种特性，但由于注意力机制仍然改为了线性，因此 FLOPs 这一项指标还是有较大的提升。

表 3-2 线性注意力机制内部探究

不同线性注意力	复杂度	FLOPs	Top-1 ACC
DeiT-S（基准模型）	$O(n^2)$	4.6	79.8
Ours-A	O(n)	2.1	77.1
Ours-B	O(n)	2.3	76.2
Ours	O(n)	2.3	78.8

3.5.6 Token 剪枝内部实验探究

（1）不同评分方法比较

在 3.3.2 小节中提到过，本章最终参考 Class Token 所对应的注意力权重来计算最后各个候选 Token 的重要性得分，为了评估这个方法本身的效果，本小节选取了另外两种方法来计算每个候选 Token 的重要性分数。首先，第一种方法 all score 是将所有 Token 的注意力权重相加，从而找到最为重要的某些 Token。第二种方法就是随机选择除去分类 Token 以外的另一个 Token，根据它的注意力权重来计算分数。得到的实验结果如图 3-5 所示。

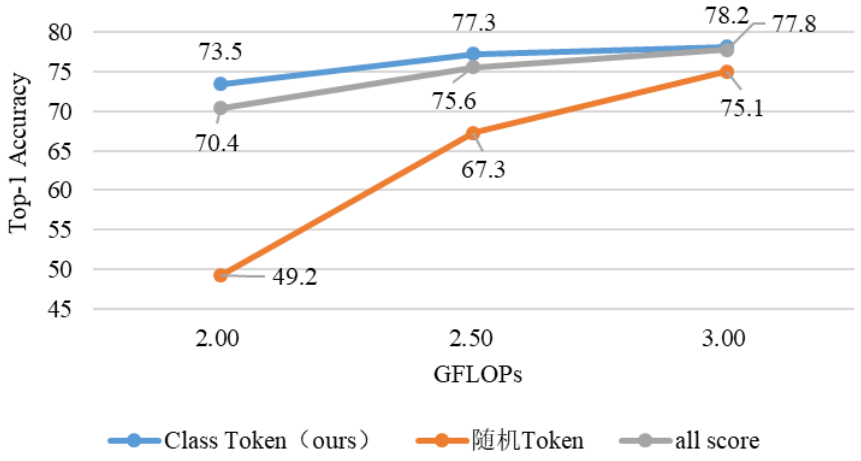


图 3-5 不同评分方法对比

由图 3-5 可以看出，使用 Class Token 的注意力权重表现得更好，说明 Class Token 的注意力权重对于评估候选 Token 来说更有参考价值。这是因为 Class Token 在模型的最后阶段将被用来预测类别概率，因此，Class Token 所对应的注意力权重表明了哪些 Token 对最后输出用于分类的 Token 的影响更大。而对所有注意力权重相加只显示了所有其他 Token 当中注意力权重最高的 Token，这对于分类 Token 不一定有用。最后随机选择一个 Token，参考其注意力权重计算最终得分效果是最差的。

（2）不同采样方法比较

本章是基于反函数实现对不同得分的输入 Token 的下采样，在得到这个方法之前，首先尝试了直接选择保留得分最高的前 k 个 Token，两个实验的对比结果如图 3-6 所示。

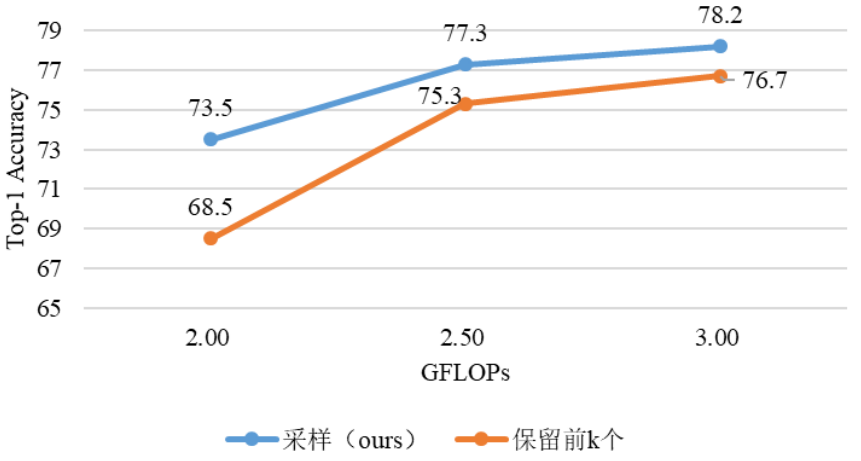


图 3-6 不同采样方法对比

显然本章所采用的方法更加优于直接选择前 k 个 Token，基于得分的累积分布函数的反变换采样方法并不一定会丢弃所有分数较低的 Token，因此为后面的阶段提供了一组更多样化的 Token 集合。此外，Top-K 选择方法将在每个阶段产生一个固定的 Token 选择率，这也限制了模型的性能。

3.5.7 图像分类实验

本小节实验是采取对于 3.5.4 小节所述的不同基准模型在 ImageNet1k 数据集上做 3 次重复的实验取平均值的方式来进行验证，从而保证实验结果尽可能的准确，使不确定因素导致的误差降到最低。其中 Acc 和 FLOPs 这两个指标下不同模型的实验结果如表 3-3 所示。

表 3-3 基准模型图像分类实验结果

模型	Top-1 Accuracy	Params (M)	FLOPs(G)
DeiT-S	79.8	22	4.6
PVT-S	79.8	25	3.8
Swin-T	81.3	29	4.5
TNT-S	81.5	24	5.2
T2T-ViT-14	81.5	22	5.2
CaiT-XXS36	79.7	17	3.8
PVT-M	81.2	44	6.7
Swin-S	83.0	50	8.7
T2T-ViT-19	81.9	39	8.9
CaiT-XS36	82.9	38	8.1
DeiT-B	81.8	86	17.5
PVT-L	81.7	62	9.8
Swin-B	83.3	88	15.4
TNT-B	82.9	66	14.1
T2T-ViT-24	82.3	64	14.0
CaiT-S36	83.9	68	13.9

接下来选取上表中的 DeiT-B、PVT-M、Swin-S、TNT-S、T2T-ViT-19 和 CaiT-XS36 这 6 个模型，将这 6 个不同的 Transformer 模型分别加入 Token 剪枝，线性注意力或者 E-Attention 之后再去对比 FLOPs 指标，结果如表 3-4 所示。

表 3-4 FLOPs 指标对比

模型	DeiT-B	PVT-M	Swin-S	TNT-S	T2T-ViT-19	CaiT-XS36
base	17.5	6.7	8.7	5.2	8.9	8.1
线性注意力	8.4	3.4	4.3	2.7	4.2	3.9
Token 剪枝	11.9	4.4	5.7	3.5	6.1	5.5
E-Attention	6.5	2.6	3.3	2.1	3.1	3.1

由表 3-4 可以看出对于选取的这 6 个模型而言，不管是单独引入哪一种改进方法，又或是将两种方法一起引入，FLOPs 这一项指标都有较大的提升，也验证了预期，因为这两种方法设计之初就是为了能从模型外部与内部两个角度各自去加速模型。对于模型 DeiT-B 来说，单独将注意力机制改为线性之后，FLOPs 指标提升了 52%，单独引入剪枝模块之后，FLOPs 指标提升了 32%，两者一起纳入模型的改进之后，总共提升了 63%。

具体来说，对于剩下的 PVT-M，Swin-S，TNT-S，T2T-ViT-19 和 CaiT-XS36 这五个模型，单独将注意力机制改为线性注意力机制之后的提升依次为 49%，51%，48%，53%，52%；单独引入模型轻量化 Token 剪枝模块之后的提升依次为 33%，35%，33%，31%，32%；两个改进均加入之后模型的提升依次为 62%，62%， 59%，65%，62%。

总体而言，单独将注意力机制改为线性之后，FLOPs 指标提升约在 50%-60%，单独引入剪枝模块之后，FLOPs 指标提升约在 30%-40%，同时将两种改进引入之后，提升范围为 60%-70%。

表 3-5 Acc 指标对比

模型	DeiT-S	PVT-L	Swin-B	TNT-B	T2T-ViT-24	CaiT-S36
base	79.8	81.7	83.3	82.9	82.3	83.9
线性注意力	78.8	80.5	81.8	81.3	80.2	82.2
Token 剪枝	79.6	81.5	83.1	82.7	82.0	83.7
E-Attention	78.6	80.4	81.4	81.1	80.0	82.1

接下来选取上表中的 DeiT-S、PVT-L、Swin-B、TNT-B、T2T-ViT-24 和 CaiT-S36 这 6 个模型，将这 6 个不同的 Transformer 模型分别加入 Token 剪枝，线性注意力或者 E-Attention 之后再去对比 Top-1 Accuracy 指标，结果如表 3-5 所示。

由表 3-5 可以看出对于选取的这 6 个模型而言，不管是单独引入哪一种改进方法，又或是将两种方法一起引入，Top-1 Accuracy 这一项指标都有一定程度的下降，因为相较于原模型而言，模型改进的两个方法都是为了加速模型，都是在牺牲一部分性能的情况下去提升模型的速度。

具体来说，对于 DeiT-S、PVT-L、Swin-B、TNT-B、T2T-ViT-24 和 CaiT-S36 这 6 个模型，单独将注意力机制改为线性注意力机制之后的下降依次为 1.3%，1.5%，1.8%，1.9%，2.5%，2.0%；单独引入模型轻量化 Token 剪枝模块之后的下降依次为 0.2%，0.2%，0.3%，0.2%，0.4%，0.3%；两个改进均加入之后模型的下降依次为 1.5%，1.6%，2.3%，2.2%，2.8%，2.2%。

单独改为线性注意力机制的性能下降在 1.5%到 2.5%之间，而单独引入模型轻量化 Token 剪枝模块之后，模型的性能下降在 0.2%到 0.5%之间，远远小于注意力机制改进引起的性能下降。引入两个改进之后，总体模型性能下降比例约在 1.5%-3%这个区间。

3.5.8 目标检测实验

本小节实验使用完全基于 Transformer 的目标检测器 Deformalbe DETR 在 COCO 数据集上训练，并将其特征提取部分替换为模型 DeiT-S 和 Swin-T，其他地方不做任何修改，然后再在其基础上分别加入 Token 剪枝，线性注意力或者 E-Attention 之后去对比指标 mAP 和 FPS。

表 3-6 目标检测实验结果

模型	mAP	Params (M)	FPS
DeiT-S	43.6	35	8.5
DeiT-S+线性	42.9	35	12.1
DeiT-S+剪枝	43.3	35	11.1
DeiT-S+ E-Attention	42.3	35	13.6
Swin-T	47.0	39	6.3
Swin-T+线性	46.3	39	9.5
Swin-T+剪枝	46.7	39	8.2
Swin-T+ E-Attention	45.7	39	10.1

由表 3-6 可以看出，针对这些模型而言，无论是单独将注意力机制改进为线性还是引入 Token 剪枝，又或者将两者一起纳入，指标 FPS 的变化规律与图像分类

实验的指标 FLOPs 相同，单独将注意力机制改为线性之后，FPS 指标提升约在 50%-60%，单独引入剪枝模块之后，FPS 指标提升约在 30%-40%，同时将两种改进引入之后，提升范围为 60%-70%。

而指标 mAP 的变化规律也是和图像分类实验中的指标 ACC 一样，都有一定程度的下降，因为改进本身是从内部和外部去加速模型，而这一提升是以牺牲模型的性能为代价的，由实验数据可知，单独引入 Token 剪枝之后性能下降范围在 0.5%-1%这个区间内，单独引入线性注意力之后性能下降范围在 1.5%-2%这个区间内，引入高效注意力之后性能下降范围在 3%左右，在接受范围之内。

3.6 本章小结

本章主要介绍了线性注意力机制和 Token 剪枝这两个方法。可以直接引入现有大多数 Transformer 模型，分别从内部和外部分别来降低计算成本，加速模型。最后将两种方法合并得到了一种的高效注意力机制（E-Attention）。实验表明，E-Attention 引入 Transformer 模型之后，在图像分类和目标检测任务当中可以使计算量降低 60%-70%，总体模型性能下降 1.5%-3%左右。

第四章 卷积-Transformer 图像特征提取网络

4.1 引言

除去高昂的计算成本之外，Transformer 还存在一系列缺陷：首先是自身需要采用超大的数据集进行预训练在实验效果上才能和 CNN 媲美；然后就是与 CNN 相比，Transformer 稳定性较差，对于优化器，超参数的选择较为敏感，收敛速度较慢。

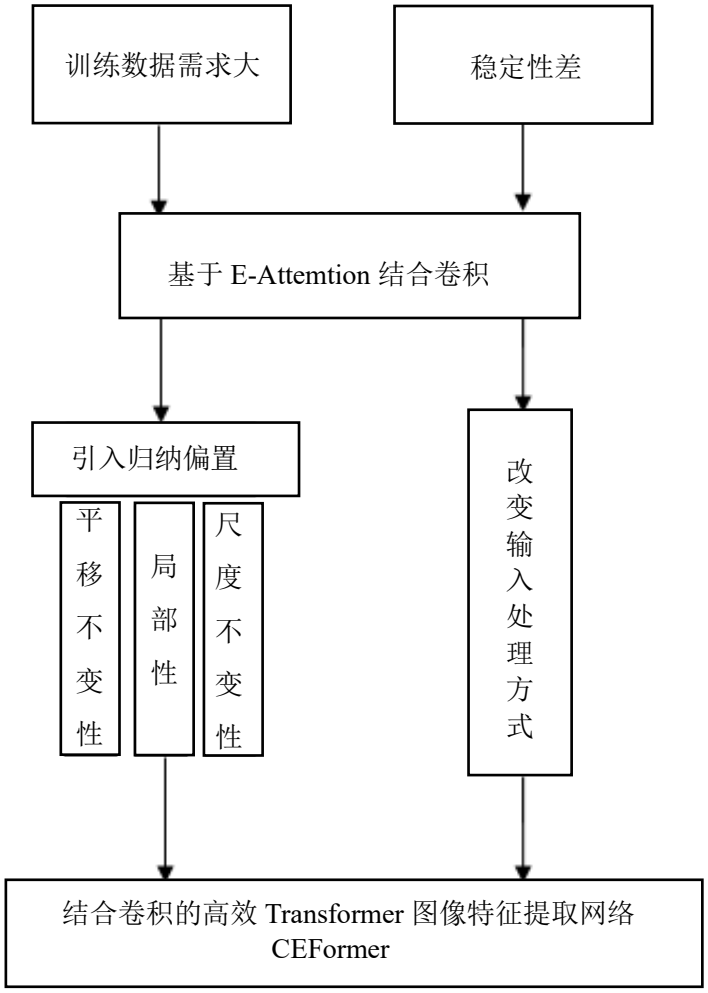


图 4-1 方法框架图

在第三章提出的 E-Attention 基础上，本章提出了结合卷积的高效 Transformer 图像特征提取网络（CEFormer），方法框架见图 4-1。首先 Transformer 对训练数据的超大需求本质上是因为缺乏类似于 CNN 的归纳偏置，因此本章从不同角度结合卷积引入平移不变性，局部性和尺度不变性。最后利用一个轻量化卷积模块改变 Transformer 模型对输入图片的传统处理方式，从而加快收敛速度，提升稳定性。

4.2 整体架构

在第三章提出的 E-Attention 基础上，本章提出了高效 Transformer 图像特征提取网络（Convolutional Efficient Transformer,CEFormer）。首先从不同角度结合卷积引入平移不变性，尺度不变性，局部性。最后利用一个轻量化卷积模块改变 Transformer 模型对输入图片的传统处理方式，从而加快收敛速度，提升稳定性。总体流程图如图 4-2 所示。

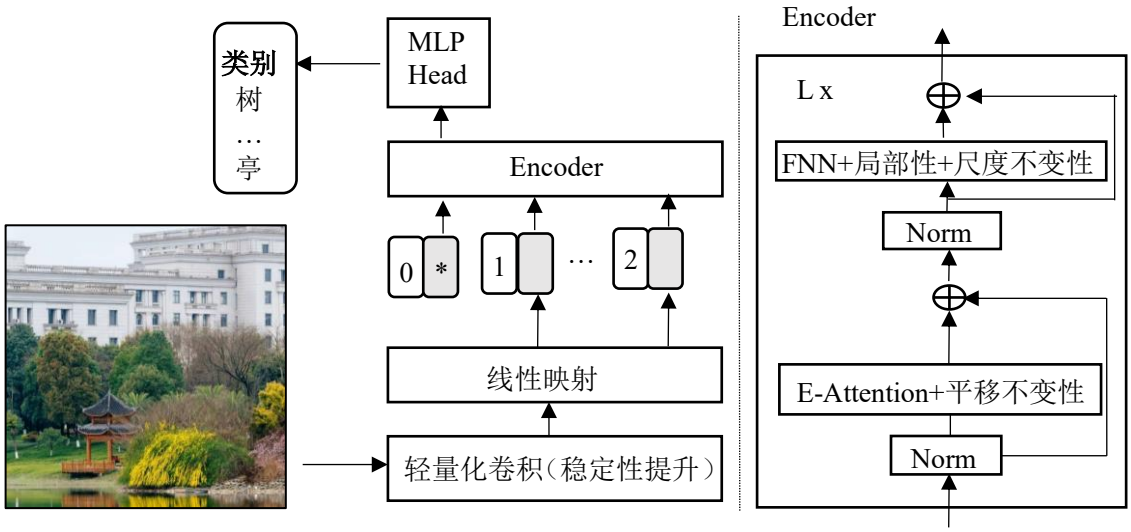


图 4-2 整体架构图

4.3 基于卷积的归纳偏置引入

相较于相同尺寸的 CNN 而言， Transformer 模型不仅性能略弱，而且所需的训练数据量也要大很多。针对这个问题，学术界进行了许多探究，目前被广泛接受的说法是 Transformer 模型缺乏某些固有的 CNN 架构中的理想属性，或者称之为归纳偏置，这些属性使得 CNN 模型很适合视觉任务[79-84]。

当使用 Transformer 模型中的自注意力机制去处理一张图片的时候，某个位置的像素产生 query，其他的各个像素产生 key。在做内积的时候，考虑的不是一个小的范围，而是一整张图片。但是当同样一张图片输入到 CNN 模型时，这个位置的像素无需考虑输入图片的全局信息，而是仅考虑感受野当中的信息即可。因此自注意力机制可以看作是一种泛化版本的 CNN。

在 CNN 中只考虑感受野里面的信息，而感受野的范围和大小是事先给定的。但是对于自注意力机制来说感受野的范围和大小就好像是自动被学出来的，因而 CNN 可以看做是自注意力机制的特例。

图像中相邻的像素通常高度相关，有明显的二维局部结构，而 CNN 就可以通过使用局部感受野，空间下采样等操作来捕获这种局部结构。此外，卷积核的层次结构从不同层次，不同角度学习到了局部空间上下文，从最为简单的低级边缘和纹理特征到较为高阶的语义信息。这些都是 CNN 具备的适用于视觉任务的属性。

因而可以结合卷积和 Transformer，引入归纳偏置。本小节选择引入深度卷积和空洞卷积。

4.3.1 深度可分离卷积

深度可分离卷积[85]的提出是因为 CNN 计算量较大，为了减小开销，方便部署在移动端。在卷积运算的基础上，学者们提出了深度可分离卷积。

深度可分离卷积除了涉及空间维度以外，还涉及深度维度（即 channel 维度）。通常输入图像会具有 3 个 channel。在经过一系列卷积操作后，输入特征图就会变为多个 channel。对于每个 channel 而言，可以看成对输入图像某种特定特征的解释说明。

深度可分离卷积包含深度卷积和逐点卷积这两个单独的小卷积核，各自进行深度卷积运算和逐点卷积运算。

首先对深度卷积运算进行说明。深度卷积运算其实就是逐通道对输入特征图进行卷积运算。深度卷积只在一个通道上进行卷积计算，而非像一般的卷积那样在每个通道上进行计算，这也导致深度卷积不会影响输入特征图的通道数目。以分辨率为 12x12x3 的输入图像为例，使用大小为 5x5 的深度卷积核进行逐通道运算，计算方式如图 4-3 所示。

这里其实就是使用 3 个 5x5x1 的卷积核分别提取输入图像中 3 个 channel 的特征，每个卷积核计算完成后，会得到 3 个 8x8x1 的输出特征图，将这些特征图堆叠在一起就可以得到大小为 8x8x3 的最终输出特征图。

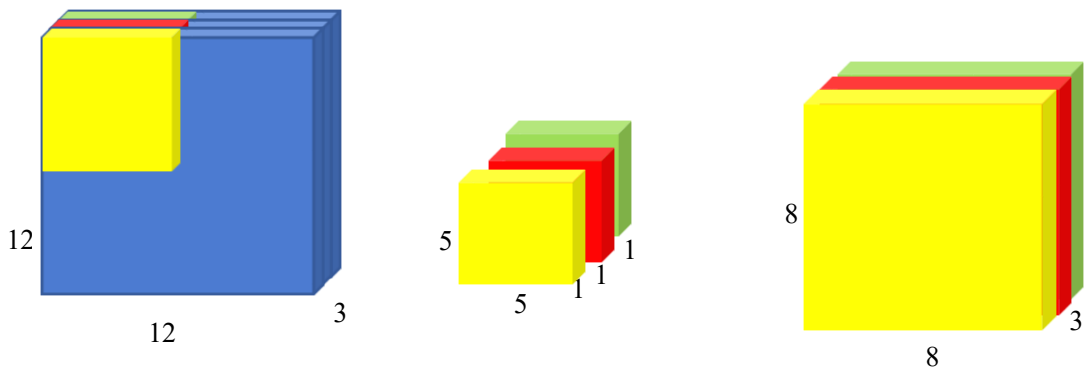


图 4-3 深度卷积示意图

但深度卷积运算有一个缺陷，那就是对于输入的特征图，它只会单独计算一个通道，而忽略了各个通道之间的信息交互，导致在后续信息流动中缺少通道之间的信息。因此需要连接一个逐点卷积来弥补它的缺点。

逐点卷积就是 1x1 卷积，因为其会遍历输入特征图的每个位置，所以称为逐点卷积。与忽略各通道信息交互的深度卷积不同，逐点卷积可以进一步融合通道之间的信息，对输入特征图进行维度的放缩。如图 4-4 所示，对图 4-3 得到的 8x8x3 的特征图可以使用一个 3 通道的 1x1 卷积进行运算，从而可以得到一个 8x8x1 的输出特征图。此时，使用逐点卷积实现了融合 3 个通道间特征的功能，起到了信息交互的作用。

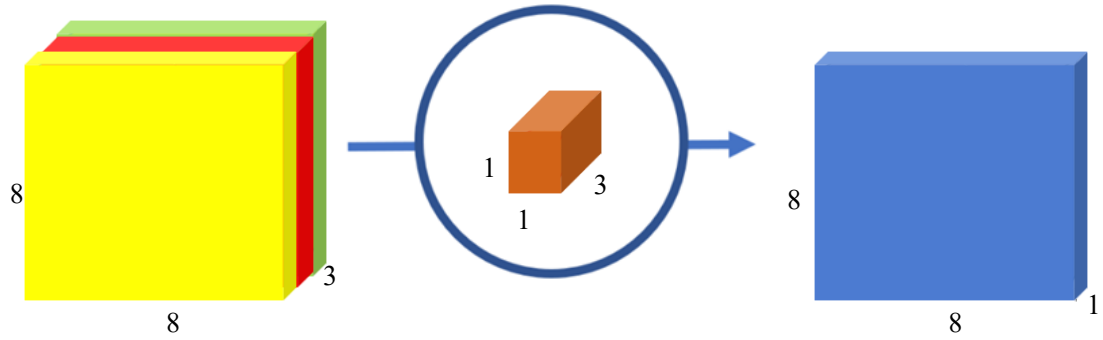


图 4-4 输出通道为 1 的逐点卷积

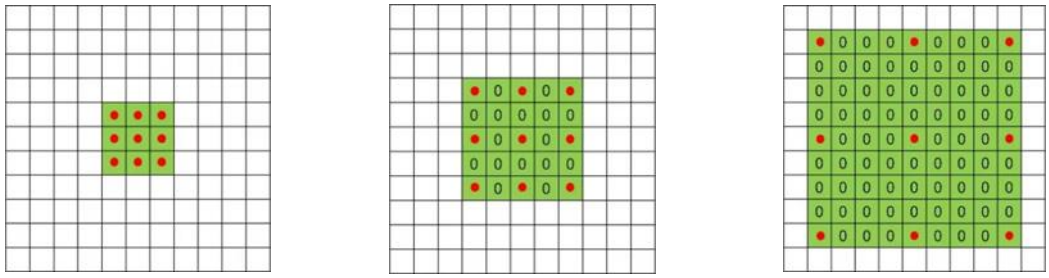
4.3.2 空洞卷积

最开始提出空洞卷积[86]是为了更好的解决图像分割所面临的图像分辨率降低、信息丢失问题。大多数图像分割算法为了增加局部感受野一般是通过卷积或者池化的方式，这样缩小了特征图尺寸，最后通常使用上采样的方法将图像的尺寸还原。然而这个过程会存在一些问题，那就是特征图先变小再放大的操作会造成精度不可避免的下降，从而丢失输入图片的细节信息。因此需要一种可以替代上采样和下采样的操作，使得在增加感受野的同时保证特征图的尺寸不变，而空洞卷积就是为了满足这种需求所设计出来的一种卷积方式。

对于一个尺寸为 3 x 3 的标准卷积来说，卷积核大小为 3 x 3，卷积核上共包含 9 个参数，卷积核会与输入特征图上对应位置的元素进行逐像素的乘积并求和。而与标准卷积相比，空洞卷积多了扩张率这一个参数，这个参数控制了卷积核中相邻元素间的距离，影响卷积核感受野的大小。

空洞卷积的核心思想就是通过添加空洞的方式来将感受野扩大，从而让最初尺寸为 3x3 的卷积核，能够在保持同样的参数量以及计算量的同时拥有 5x5（此时

对应的扩张率设置为 2）甚至更大的感受野，从而避免使用会造成精度受损的下采样方法。



(a) (b) (c)

图 4-5 扩张率为 1,2,4 的 3x3 空洞卷积

如图 4-5(a)所示，当超参数扩张率设置为 1 的时候，空洞卷积与标准卷积这两者的计算方式一样。然而当超参数扩张率大于 1 时，那么在标准卷积的基础上，会注入空洞，空洞中的数值全部填 0。图 4-5(b)和图 4-5(c)对应的分别就是扩张率为 2 和扩张率为 4 的情况。

空洞卷积主要有以下三个作用：

（1）扩大感受野。池化操作也可以扩大感受野，但会造成空间分辨率的降低，相比之下，空洞卷积可以保持像素的相对空间位置不变，在扩大感受野的同时不丢失分辨率。

（2）获取多尺度上下文信息。当多个带有不同扩张率的空洞卷积核叠加时，不同的感受野会带来多尺度信息。

（3）降低计算量。不需要引入额外的参数。

4.3.3 平移不变性

之所以选择将深度卷积与高效注意力结合，是因为深度卷积和高效注意力都可以表示为预先所定义的感受野中的值的加权和。深度卷积依赖于一个固定大小的单层卷积核来从局部感受野当中收集信息，公式如 4-1 所示：

$$y_i = \sum_{j \in L(i)} w_{ij} x_j \tag{4-1}$$

式中 x_i 和 y_i 分别是位置 i 的输入和输出， $L(i)$ 表示位置 i 的局部感受野。

高效注意力机制可写成类似公式 4-1 形式，公式如 4-2 所示：

$$y_i = \sum_{j \in G} x_i'^{\mathrm{T}} x_j' \sin\left(\frac{\pi(i+n-j)}{2n}\right) x_j \tag{4-2}$$

式中 $x_i' = f(x_i)$， $f(\cdot)$如公式 3-7 所示， G 表示全局空间，i，j=1，…，n。

在讨论如何最好地组合这两者之前，先比较一下各自的相对优势和劣势，这有助于找出最希望保留的良好特性。

（1）深度卷积核的权重是一个独立于输入的固定参数，而高效注意力的权重则相反，它是一个动态依赖于输入，随着输入值而变化的参数。所以高效注意力机制建模不同空间位置的复杂的关系是更容易的，但是也更容易过拟合，对数据集的数量要求较高。

（2）对于给定任意的位置对 (i, j)，深度卷积核权重只与这两个位置的相对位移有关，而与各自的绝对数值无关，这一性质来源于卷积所自带的权重共享特性，也将其称之为平移不变性，在数据集有限的情况下这个性质对于提升泛化能力帮助很大[87]。而高效注意力机制，乃至最原始的 Softmax 注意力机制都不具备这个性质，也因此当数据集较小时，CNN 的效果通常要比 Transformer 模型更好。

（3）深度卷积和高效注意力机制的感受野大小不同，通常来说，感受野越大，它就能提供更多的上下文信息，与之相对应的模型容量也会随之提高，这一点恰恰是注意力机制的优势所在。当然，感受野越大，需要的计算量也就越多，对于原始的 Softmax 注意力机制来说，计算复杂度是二次级别的，而这一问题已经被第三章改进之后的高效注意力机制所解决。

总而言之，可以将以上三点归纳为平移不变性（深度卷积所特有），全局感受野和自适应输入的权重（均为高效自注意力机制所特有）。因此，将这三点特性结合起来，具体来说，就是在保留高效自注意力机制的这两点特性之余，引入深度卷积所特有的平移不变性。

一个简单的方法就是将深度卷积的权重加入高效注意力机制当中去，如公式 4-3 所示：

$$y_i = \sum_{j \in G} (x_i'^{\mathrm{T}} + w_{ij})(x_j' + w_{ij}) \sin(\frac{\pi(i + n - j)}{2n}) x_j \tag{4-3}$$

至此，将深度卷积与高效注意力机制融合，从而具备以上三点特性，额外引入了平移不变性。

4.3.4 局部性

本小节通过在前馈神经网络当中结合深度卷积来引入局部性。首先前馈神经网络当中包含两个全连接层，在这两个全连接层之间，隐藏维度会被扩展以便提取更为丰富的特征。而如果从卷积的视角来看，这个操作等价于用来对特征图的维度进行升降的 1x1 点卷积，这与轻量化模型 MobileNet 系列当中的反向残差结构[88-90]类似，在反向残差块当中，两个 1x1 点卷积同样被扩展，如图 4-6 所示。

可以看出，如果将两个全连接层看成两个 1x1 点卷积的话，那 Transformer 模型当中的前馈神经网络和反向残差结构相比较，反向残差结构当中只是多了一个可以将局部信息聚合的深度卷积而已。由此受到启发，本小节尝试将深度卷积引入到 Transformer 模型编码器模块的前馈神经网络当中，并且直接将两个全连接层替换为两个 1x1 的点卷积。

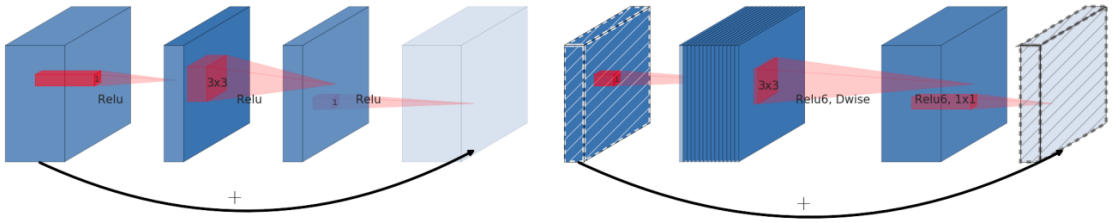


图 4-6 残差结构与反向残差结构[89]

由于 Transformer 模型当中，每个 block 之间输入与输出的都是 Token 序列，因此整个操作过程之前需要先将 Token 序列重新排列成特征图，卷积操作结束之后需要将特征图展平成最初的 Token 序列，需要注意的是整个过程所涉及到的 Token 序列是从第二个 Token 开始的，因为第一个 Token 与输入的图片无关，是额外引入用于在最后阶段判断图像类别。局部性引入之后前馈网络的整体计算流程图如图 4-7 所示。

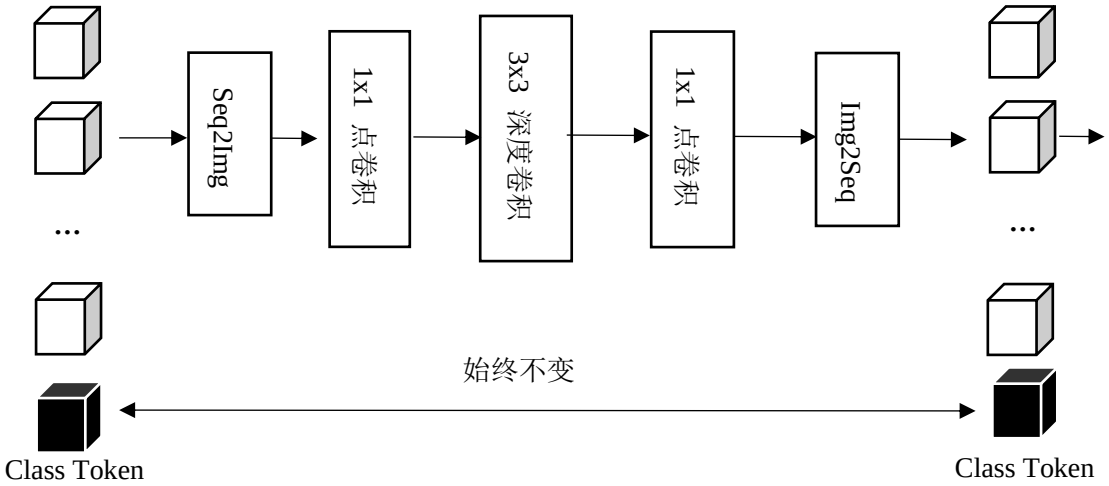


图 4-7 引入局部性的前馈网络计算流程图

4.3.5 尺度不变性

在 CNN 时代的各种下游任务（例如目标检测、语义分割）中，多分辨率多尺度特征可以提取不同物体尺度的特征。然而 ViT 仅仅是为图像分类而设计，内部的直筒输出结构使其无法直接应用于下游密集任务。

因此，多尺度信息的获取对于 Transformer 模型来说尤为重要。在 4.3.2 小节中提到过空洞卷积的一大作用就是可以获取多尺度上下文信息。也因此本小节考虑结合空洞卷积，以便获取多尺度信息，为基于 Transformer 设计的图像特征提取网络引入尺度不变性。

由于在 4.3.4 小节当中，前馈网络已经引入了局部性，所以本小节将空洞卷积同样引入前馈网络当中，由 Token 序列转变而来的特征图在经过 3x3 的深度卷积操作引入局部性之后，再经过不同扩张率的空洞卷积操作，额外引入尺度不变性，以获取多尺度信息。更新之后的前馈网络计算如图 4-8 所示。

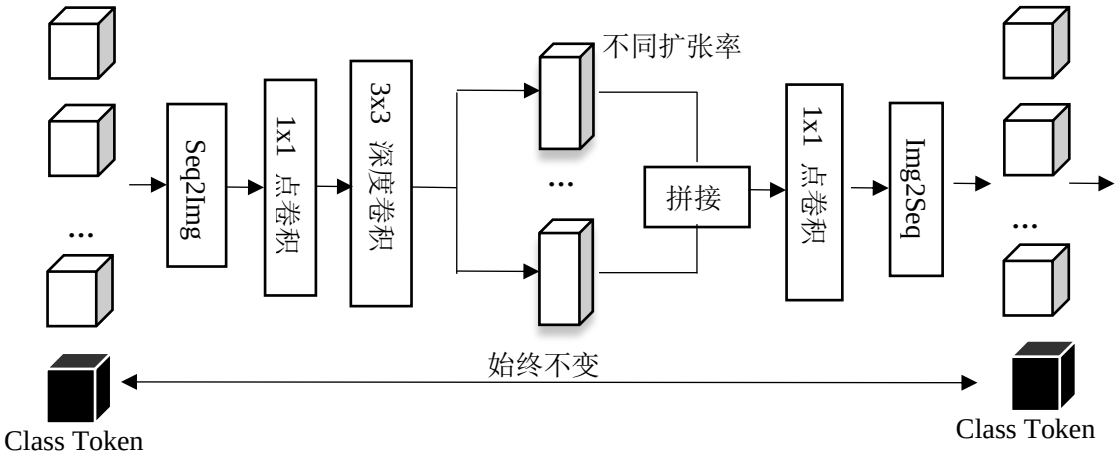


图 4-8 引入局部性和尺度不变性的前馈网络计算流程图

4.4 基于卷积的稳定性提升

Transformer 模型的优化十分受限制，除了需要精确的学习率和权重衰减，还需要使用 AdamW 优化器，并且收敛速度很慢。MoCov3[91]提到 ViT 架构会导致训练过程中出现衰退现象，并推测出 Patch 映射是衰退关键，于是通过固定住 Patch 映射的参数，从而缓解了衰退现象。由此推测 Transformer 模型不稳定的原因可能与最初将输入图片粗暴切分为一系列 Patch 送入模型有关，受此启发，可以从卷积角度观察这一做法。原始的 Transformer 模型将输入图片划分成无重叠的 pxp 个 Patch，然后对每个 Patch 进行映射转化成 d 维的特征向量。假设输入的图片尺寸为 224x224，每个 Patch 的尺寸为 16，那么 Patch 的数量为 14x14，Patch 映射等价于一个 16x16 大小，步长为 16 的大卷积核。然而在设计卷积神经网络时，一般都会

采用尽可能小而深的卷积核，一个 16x16 大小，步长为 16 的大卷积核对应的参数量也是巨大的，并且随机性很高，因此这样处理输入图片会造成较大的不稳定性。

针对这个问题，本小节从 VGG[91]网络中获得启发，它观察到 1 个 7 x 7 的卷积层的正则等效于 3 个 3 x 3 的卷积层的叠加。这样可以大减少较多参数，缓解过拟合现象。也因此，本小节参考这个思路，尝试采用一个轻量化的卷积模块去替代这个大卷积核，从而缓解不稳定性。这个轻量化的模块由一个卷积操作加上一个 Batch 正则化，然后再接一个最大池化操作。

4.5 实验结果与分析

本小节先是对本章提出的 CEFormer 做消融实验，验证各个改进模块的提升性能，然后再去验证模型的稳定性是否有所提升，最后再选取三大类图像特征提取网络，分别是纯卷积网络、纯 Transformer 网络、以及卷积和 Transformer 相结合的网络，将其分别在 ImageNet1k 数据集和 COCO 数据集上做图像分类和目标检测实验，实验结果与本章提出的 CEFormer 得到的实验结果相对比，进行评估与分析。

4.5.1 数据集介绍

本章实验所采用的数据集是 ImageNet1k 数据集和 COCO 数据集。这两个数据集在 3.5.1 小节中均已进行了详细的介绍。

4.5.2 评估指标

为了更好的评估本章所提出的模型与其他不同模型的性能差异，图像分类实验性能指标选择 Top-1 Acc、Params 和 FLOPs 这三个。目标检测实验中指标选择 mAP、Params 和 FLOPs 这三个。这四个指标在 3.5.2 小节中均已介绍过。

4.5.3 实验环境及模型设置

实验的软件和硬件环境如表 4-1 所示。

表 4-1 软件和硬件环境

类别	说明
操作系统	Windows 10 教育版 64 位系统
处理器	Intel(R) Core(TM)i7 9700K CPU@3.60GHz
图形处理器	Nvidia GeForce RTX 2070 SUPER 8GB
Pytorch	1.2.0

不同尺寸模型设置如表 4-2 所示。

表 4-2 不同尺寸模型设置

模型	Depth	Dim	Embed	Head	FLOPs(G)	Params
CEFormer-S	12	384	128	8	3.8	28
CEFormer-M	24	384	128	8	6.8	55
CEFormer-L	24	512	128	8	14.1	92

超参数方面设置如下：Epochs 为 300，Batch size 为 1024，基础学习率为 0.0005，优化器选择 AdamW，学习率衰减策略使用 Cosine，权重衰减为 0.05，Dropout 为 0.1，Warmup epochs 为 5，深度卷积与空洞卷积均为 3 x 3，轻量化卷积模块中的卷积为 7 x 7。

4.5.4 基准模型

为了尽可能的验证对比本章所提出来的 CEFormer 的实验效果，本章选取了三大类总共 24 个图像特征提取网络在 ImageNet1k 和 COCO 数据集上进行图像分类和目标检测实验进行比较。

其中纯卷积图像特征提取网络选择 ResNet[92]和 RegNet[93]；纯 Transformer 图像特征提取网络选择 ViT、DeiT、PVT、Swin、TNT、T2T-ViT、CaiT、DeepVit[94]、AutoFormer[95]、Shuffle Transformer、NesT[96]、Focal Transformer[97]、CrossFormer[98]和 CSWin 这 14 个；卷积与 Transformer 相结合的图像特征提取网络选择 ConViT、PiT、CvT、LV-ViT、GG-Transformer、CMT、GLiT 和 ConTNet 这 8 个。接下来分别对这 24 个模型进行说明。

（1）纯卷积图像特征提取网络

（a）ResNet：残差网络 ResNet 是深度学习领域比较经典的 CNN 模型之一，它是为了解决神经网络“退化”的现象而被设计出来，即随着卷积网络深度的增加，训练集的损失会出现先逐渐下降，然后趋于平缓的现象；当继续增加卷积网络的深度时，训练集的损失却会开始上升。残差网络由一系列残差块堆叠而成，而每个残差块由直接映射部分和残差部分组成。残差网络在设计时在网络中引入了直连通道这个概念，也就是允许将前面网络层当中一定比例的输出保留，使得前面网络层的信息直接传入后面的层中，这样当前这一层的神经网络就可以学习上一个网络输出的残差，而不用学习整个的输出。

（b）RegNet：RegNet 是一个结合手动设计和神经架构搜索优点的网络，通过对搜索空间的优化和限制，提高了搜索的速度和效果。这个模型在设计之初没有选择考虑设计单个的网络实例，反而通过引入设计空间将所有网络参数化，替代了原先的搜索空间，也就是将整个搜索空间重新设计，而非简单搜索可行的单个网络实例。RegNet 在多种任务上提供更快和简单的网络，在可比较的训练设定和浮点计算上，RegNet 建模可能比现有的 EfficientNet 快 5 倍以上。

（2）纯 Transformer 图像特征提取网络

由于纯 Transformer 模型中的 ViT 、DeiT、PVT、Swin、TNT、T2T-ViT 和 CaiT 这七个已经在第三章的实验准备中详细介绍了，所以略去，下面介绍剩下的这七个模型。

（a）DeepVit：模型 DeepVit 的提出是因为相较于 CNN 而言，ViT 在不断堆叠编码器，扩张深度时性能会快速饱和，这种现象称之为注意力崩溃。也就是随着 Transformer 模型的深度增加，模型内部所得到的注意力图不会发生太多改变，说明此时的自注意力机制无法学习到有效的特征表示，无法得到预期的性能提升。基于此，DeepVit 提出了 Re-attention，这种注意力机制可以通过将注意力图重新生成的方式来增加各层之间的多样性，同时这种实现可以极大程度上节省计算消耗。借助于 Re-attention，DeepVit 可以在训练更深的 ViT 模型时，依然保持模型性能不断提升。

（b）AutoFormer：AutoFormer 本身是将神经网络架构搜索引入到 Transformer 当中，使用神经网络架构搜索的方法通过设置对应超参搜索空间自动去选择 Transformer 设计中的关键参数，比如，网络深度，Embedding 的维度，多头自注意力机制中头的数目。另一方面训练超网，并在训练期间纠缠同一层不同块的权重，在预设置好的搜索空间采样到子网，更新子网的参数，冻结其余的参数不使其更新。收益于该策略，结果训练的超网可以很好的训练数千个子网，具体来说，这些从超网继承权重的子网的性能与从头开始训练的子网相当。并且采用进化算法得到参数量最小精度最高的模型，从而可以针对不同的场景需求可以直接得到相应的 ViT。

（c）Shuffle Transformer：Transformer 模型 Shuffle Transformer 的提出是因为传统 Transformer 模型使用的全局自注意力机制的具有关于输入 Tokens 数量的二次计算复杂度。这使得 ViT 很难在语义分割和物体检测等需要输入高分辨率图像的密集预测任务上应用。文中提出了一种新的 Transformer 结构，将空间 Shuffle 和基于窗口的自注意力机制结合，有效的建立起了跨窗口连接，增强了模型的表达能力。并且在窗口自注意力模块和前馈网络之间插入了一个带残差连接的深度分离

卷积层，并且它的卷积核尺寸大小与窗口大小相同。这一算子可以增强相邻窗口之间的信息流动。

（d）Focal Transformer：模型 Focal Transformer 的提出是为了缓解传统 Transformer 模型的计算开销。Focal Transformer 采用了一种新的注意力机制 Focal Self-Attention，这种注意力机制根据不同区域距离当前 Token 的远近进行不同颗粒度的关注，如果区域距离当前 Token 较近，则对其进行细粒度的关注，反之则进行粗粒度的关注。通过这样的方式，Focal Self-Attention 可以对局部和全局的注意力进行更加有效的捕获。不同于改进局部信息建模的 Transformer 模型，Focal Transformer 不是分别建模局部和全局信息，再将其简单串联或并联，而是在同一个结构中建模这两种信息。

（e）CrossFormer：模型 CrossFormer 的提出是因为当前许多 Transformer 模型将输入图像切分为相同大小的 Patch，继而生成一系列嵌入，因而在同一层中的所有嵌入具有相同的尺度，无法建立跨尺度的特征。另外许多 Transformer 模型会选择合并相邻的嵌入以降低自注意力模块计算消耗，然而这种合并的方式会使得这些模型丧失细粒度的特征，同样导致了这些模型无法构建跨尺度的注意力。模型 CrossFormer 的设计参考了 PVT，同样采用了金字塔式的分层结构，将模型分为了多个不同的阶段。模型内部的核心设计包括两个部分，分别是跨尺度嵌入模块和长短距离注意力模块，弥补了以往 Transformer 模型对于建立跨尺度注意力方面的不足。

（f）CSWin：模型 CSWin Transformer 的提出是为了改进模型 Swin Transformer 中的 SW-MSA 结构，这种结构实现过于复杂且对 CPU 设备不友好，难以部署。CSWin Transformer 提出了十字形状的自注意力机制，具备跨窗口局部注意力计算能力，摒弃了模型 Swin Transformer 中的 SW-MSA 结构，能够在水平和垂直两个方向上同时计算注意力权重。同时 Swin Transformer 还提出了一种新颖的局部增强位置编码，相比于传统的位置编码，Swin Transformer 提出的位置编码能够适应不同大小的输入特征，同时还具有更强的局部假设偏置。

（g）NesT：模型 NesT 的提出是因为 Transformer 模型自身缺乏类似于 CNN 中局部性和平移等效性之类的归纳偏置，尽管在较大规模的数据集上有较好的性能表现，但当数据集规模变小时，Transformer 模型在训练之后的性能无法与 CNN 相比。NesT 利用嵌套层次结构来对特征表示进行学习，同时也可以解藕抽象过程。另外 NesT 参考决策树过程提出了一种全新的模型可解释方法，称为 CradCAT，NesT 利用这种方法定位图像的显著性区域，通过引入块聚合函数来提高模型的整

体性能，同时保证原自注意力机制不变，为改进之后的 Transformer 模型提供了一定的可解释性。

（3）卷积与 Transformer 相结合的图像特征提取网络

（a）ConViT：模型 ConViT 的提出和 NesT 类似，也是因为当前基于自注意力机制的视觉 Transformer 模型缺乏归纳偏置。从而导致在小数据集上，这些模型想要学习到有意义的特征表征非常困难，而一旦数据集规模较大，这些模型的性能表现甚至可以超过 CNN。模型 ConViT 在 ViT 的基础上引入了软卷积归纳偏置，可以帮助模型不受限制地学习归纳偏置，极大程度上提升模型的学习效率。但是这中方式会在数据集规模不确定时受到约束，因此 ConViT 引入门控位置自注意力使得模型在训练过程中可以自行决定是否要保持卷积，模型在训练过程中会学习用于平衡基于内容的自注意力和卷积初始化位置自注意力的门控参数。

（b）PiT：模型 PiT 的提出是为了进一步研究当前 Transformer 模型的有效架构设计，PiT 参考 CNN 的设计原则，引入了空间维度转换，研究其对 Transformers 的架构的有效性。PiT 在 Transformer 模型中引入基于深度卷积的池化层，以较少的参数实现通道乘法，并将空间缩减，从而可以像在 ResNet 中一样减少 ViT 结构中的空间大小。

（c）CvT：模型 CvT 将卷积引入到 Transformer 模型当中，划分为多个阶段，形成一个包含卷积 Token 嵌入的分层 Transformer。卷积 Token 嵌入在每个阶段的开始位置，先将当前阶段输入的 Token 序列变为二维结构，然后利用卷积 Token 嵌入对其进行卷积操作，再进行 Layer 正则化操作。这个方法使得 CvT 在捕获到局部信息的同时，还可以将 Token 序列的长度减少，并且在不同阶段还可以增加 Token 特征的维数，达到对空间进行下采样的目的。除此之外，CvT 还设计了卷积 Transformer Block，用卷积投影替换每个自注意力 Block 之前的线性投影，使得改进之后的注意力机制可以进一步得到局部空间语义信息，尽可能减少语义歧义。另外 CvT 还可以降低计算复杂度，因为卷积可以用于对键和值矩阵进行下采样，以提高 4 倍或更多的效率，同时最小化性能的损失。

（d）LV-ViT：模型 LV-ViT 的提出主要是为了探索各种可以用于提升 ViT 性能的训练技巧，而不是像其他模型一样仅仅提供一种新颖的 Transofrmer 架构。LV-ViT 首先将前馈层的隐藏层维度降低，因为实验表明这不会影响模型性能；然后又添加了卷积层来增强块嵌入模块的容量从引入归纳偏置。同时为了从更大感受野当中提取信息，还采用了更小步长的卷积以提供相近 Token 的重叠信息。最后 LV-ViT 引入了 re-labeling 提供的稠密得分图为每个图像块与其对应 Token 提供一个独特标签。

（e）GG-Transformer：Transformer 模型 Glance and Gaze Transformer 具有高效的建模远程依赖关系的能力，在 GG-Transformer 中，Glance 和 Gaze 行为是通过两个并行分支实现的，Glance 分支是通过对输入的自适应扩张分区执行自注意力机制实现的，在导致线性复杂度的同时仍然享受全局感受野，Gaze 分支由一个简单的深度卷积层实现，将局部图像上下文补偿到 Glance 机制获得的特征。

（f）CMT：CMT 同时具有 Transformer 长距离建模与 CNN 局部特征提取能力，整体架构采用了 ResNet 的分阶段架构，正则化方面采用 CNN 中常用的 BN 而非 Transformer 中的 LN，在核心模块 CMTBlock 方面，内部设计了具有局部特征提取的 LPU，在降低计算量方面对 K 与 V 进行了特征分辨率的下降。

（g）ConTNet：ConTNet 结合了卷积层和 Transformer，以 ConvNet 的结构为基底，将 Transformer in-place 嵌入进卷积网络中，形成卷积层和 Transformer 交替提取特征的新结构（也可以看作是分层视觉 Transformer 中添加了卷积层）。

（h）GLiT：模型 GLiT 的提出是为了将卷积神经网络最大的特点共享的局部信息建模，引入至 Transformer 中，并探索如何自动学习二者在网络中的分配比例以及配置 Transformer 中其他模块的算力配比，比如卷积核大小和每个子模块的通道数。它是一种更适合图像任务的 Transformer 网络结构，GLiT 在 Transformer 的基础上引入了共享的局部信息建模，探索了基于全局与局部注意力的 Transformer 网络结构，进而实现了更好的分类效果，提升了网络针对计算机视觉任务中至关重要的局部信息建模的能力。

4.5.5 消融实验

本章所提出的模型结合了各种卷积以纳入相应的归纳偏置，从不同角度提升了模型的平移不变性，局部性，尺度不变性和稳定性。本小节以这四种特性为消融实验的对象，探究各个特性对最终模型性能的影响。

分别将这四种特性平移不变性，局部性，尺度不变性和稳定性命名为 C1，C2，C3 和 C4。然后对这四种特性分别消融，并将消融模型的某一特性记为 w/o，Full 表示保留所有特性，其他实验的设置不变。实验结果如表 4-3 所示。

表 4-3 消融实验结果

指标	Full	w/o C1	w/o C2	w/o C3	w/o C4
Top-1 Acc	83.6	83.4	83.3	83.3	83.1
GFLOPs	3.8	3.7	3.6	3.6	3.4

由表 4-3 可以看出，消融任意一种特性，准确率都会有一定程度的下降，其中消融 C4，也就是去掉稳定性之后，准确率下降的最多，可以看出这一特性对模型的增幅最大。其次消融 C2 或 C3，也就是去掉局部性或者尺度不变性之后，准确率下降相同，说明这两个特性的对模型的增幅相同。最后消融 C1，也就是去掉平移不变性之后，准确率下降最小，说明这四个特性当中，相对而言增益最小的是平移不变性。

另外从指标 FLOPs 可以看出，消融任一特性之后，模型前向推理的计算量都会有一定程度的下降，说明模型更快了，因为消融任一特性表明减少了模型对于某种卷积的引入，因此模型加速，前向推理计算量减少。并且可以看出，消融 C4，也就是去掉稳定性之后，模型前向推理的计算量减小的最多，可见这一特性对模型的负担最重，但其对应的准确率提升同样也是最大的，所以这两者相辅相成。

4.5.6 稳定性实验

本小节中通过实验验证本章引入的稳定性卷积设计是否有效，具体将舍弃稳定性与不舍弃稳定性的模型去从不同角度对比。实验数据基于数据集 ImageNet1k。

（1）收敛速度分析

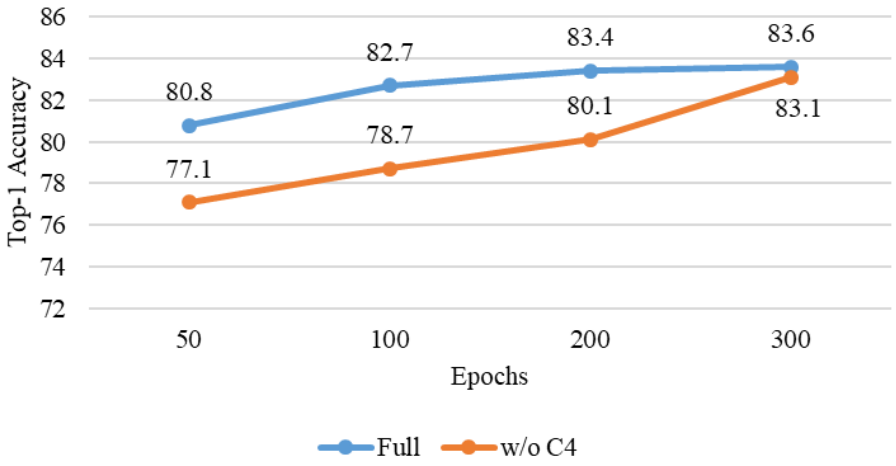


图 4-9 收敛速度实验结果

如图 4-9 所示，随着训练次数的增加，未舍弃稳定性的模型收敛速度远大于舍弃了稳定性的模型，未舍弃稳定性的模型在 200 epochs 的精度就和 300 epochs 的精度较为接近，然而舍弃了稳定性的模型在 200 epochs 的精度就和 300 epochs 的精度相差较远，说明本章的稳定性设计确实能够加速模型的收敛。

（2）优化器分析

Transformer 一般都使用 AdamW 作为优化器，原因是 SGD 优化器会在 ImageNet 数据集上使得精度下降大约 7 个点。但是卷积神经网络却没有这种情况，SGD 优化器一样可以把卷积神经网络优化得很好。相比于 AdamW 优化器来讲，SGD 优化器参数少，且占的显存只有 AdamW 优化器的一半左右。因此本小节的实验通过验证舍弃稳定性与不舍弃稳定性的模型各自使用 SGD 优化器或者 AdamW 优化器的情况去分析本章的稳定性设计能否缓解 Transformer 对于优化器的的敏感。实验结果如图 4-10 所示。

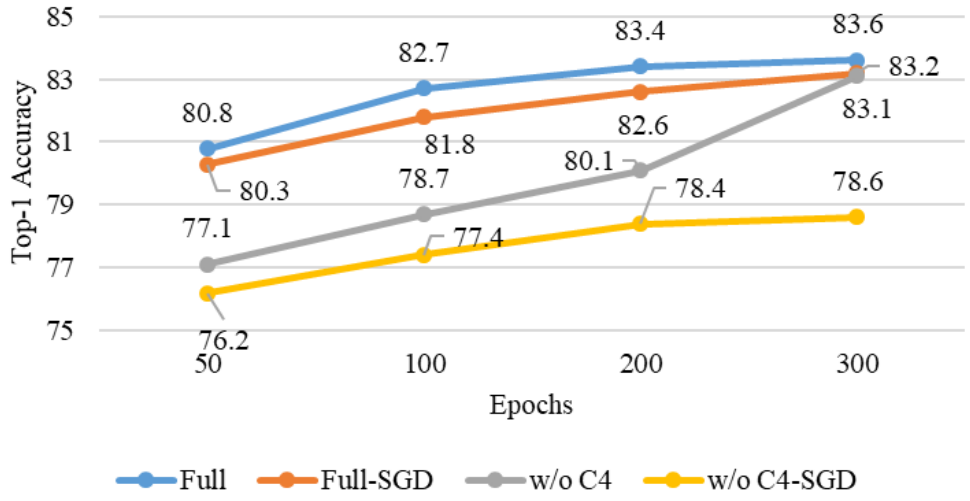


图 4-10 优化器变化实验结果

由图 4-10 可以看出保留了本章稳定性设计的模型无论使用 SGD 优化器还是 AdamW 优化器，准确率并没有太大变化。然而舍弃了本章稳定性设计的模型在使用了 SGD 优化器，准确率大幅度下降，并且 200 Epochs 之后，准确率提升极为微弱，说明本章的稳定性设计确实可以缓解 Transformer 对于优化器的的敏感程度。

4.5.7 图像分类实验

本小节实验是采取对于 4.5.4 小节所述的不同基准模型在 ImageNet1k 数据集上做 5 次重复的实验取平均值的方式来进行验证，从而保证实验结果尽可能的准确，使不确定因素导致的误差降到最低。

（1）卷积模型对比实验

Acc 和 FLOPs 这两个指标下完全基于卷积的模型与 CEFormer 的实验结果如表 4-4 所示，不同模型根据参数量分为小中大三个尺寸相对应比较，加粗数据为同尺寸系列最优。

表 4-4 卷积模型与 CEFormer 的 Acc 和 FLOPs

模型	Top-1 Acc	Params(M)	FLOPs(G)
ResNet-50	76.7	26	4.1
RegNetY-4GF	80.0	21	4.0
CEFormer-S	**83.6**	28	**3.8**
ResNet-101	78.3	45	7.8
RegNetY-8GF	81.7	39	8.0
CEFormer-M	**84.5**	55	**6.8**
ResNet-152	78.9	60	**11.5**
RegNetY-16GF	82.9	84	15.9
CEFormer-L	**85.0**	92	14.1

表 4-4 看出与 ResNet 和 RegNetY 这两个基于卷积的模型相比，CEFormer 系列的指标 Top-1 Acc 均取得了最优。CEFormer-S 比起次优模型 RegNetY-4GF 性能提升了 4.5%，CEFormer-M 比起次优模型 RegNetY-8GF 性能提升了 3.4%，CEFormer-L 比起次优模型 RegNetY-16GF 性能提升了 2.5%。

另外从指标 FLOPs 来看，除了 CEFormer-L 以外，CEFormer-S 与 CEFormer-M 均取得了最优，各自领先次优模型 5%和 13%，然后 CEFormer-L 却比最优模型 ResNet-152 落后 23%。

从这两个指标的比较情况可知，与完全基于卷积的模型相比，本章提出的模型 CEFormer 在指标 Top-1 Acc 较为领先，说明模型性能更好。但指标 FLOPs 并不能保证一定更占优势，由 3.5.7 小节的实验结果可知，单独将 Transformer 的注意力机制改进之后，模型前向推理的计算量可以提升 60%左右，但在本章中，考虑到模型 CEFormer 的改动除了将注意力机制修改之外，还引入了不少卷积以提升模型不同方面的性能，再加上对比的模型也做了不少性能改进，所以指标 FLOPs 并不能保证一定最优。

（2）Transformer 模型对比实验

Acc 和 FLOPs 这两个指标下完全基于 Transformer 的模型与 CEFormer 的实验结果如表 4-5 所示，不同模型根据参数量分为小中大三个尺寸相对应比较，加粗数据为同尺寸系列最优。

表 4-5 Transformer 模型与 CEFormer 的 Acc 和 FLOPs

模型	Top-1 Acc	Params(M)	FLOPs(G)
ViT-S	81.2	22	9.2
DeiT-S	79.8	22	4.6
PVT-S	79.8	25	3.8
T2T-ViT-14	81.7	22	6.1
CaiT-XXS36	79.7	17	3.8
AutoFormer-s	81.7	23	5.1
NesT-T	81.5	17	5.8
Focal-T	82.2	29	4.9
CrossFormer-S	82.5	31	4.9
CSWin-T	82.7	23	4.3
CEFormer-S	**83.6**	28	**3.8**
ViT-S/16	78.1	49	20.2
Swin-S	83.0	50	8.7
T2T-ViT-19	82.2	39	9.8
CaiT-XS36	82.9	38	8.1
DeepViT-L	82.2	55	12.5
AutoFormer-b	82.4	54	11
Shuffle-S	83.5	50	8.9
NesT-S	83.3	38	10.4
Focal-S	83.5	51	9.1
CSWin-S	83.6	35	6.9
CEFormer-M	**84.5**	55	**6.8**
ViT-B/16	77.9	86	17.6
Swin-B	83.3	88	15.4
TNT-B	82.9	66	14.1
CaiT-S36	83.9	68	**13.9**
NesT-B	83.8	68	17.9
Focal-B	83.8	90	16.0
CrossFormer-L	84.0	92	16.1
CSWin-B	84.2	78	15.0
CEFormer-L	**85.0**	92	14.1

由表 4-5 可以看出，与这些基于 Transformer 的模型相比，在指标 Top-1 Acc 方面，CEFormer 同样取得了最优结果，CEFormer-S，CEFormer-M 和 CEFormer-L 相比次优模型 CSWin-T，CSWin-S 和 CEFormer-B 均提升了 1%，这方面领先于完全基于 Transformer 的模型。

在指标 FLOPs 方面，CEFormer-S 与 CEFormer-M 取得了最优结果，CEFormer-L 取得了次优结果，以及对于模型 CEFormer-S 来说，PVT-S 与 CaiT-XXS36 的 FLOPs 与之相同，并不如 3.5.7 小节的实验结果那样占绝对领先优势，原因上文提到过，不再赘述。

（3）卷积与 Transformer 相结合模型对比实验

Acc 和 FLOPs 这两个指标下基于卷积与 Transformer 的模型与 CEFormer 的实验结果如表 4-6 所示，不同模型根据参数量分为小中大三个尺寸相对应比较，加粗数据为同尺寸系列最优。

表 4-6 卷积与 Transformer 相结合模型与 CEFormer 的 Acc 和 FLOPs

模型	Top-1 Acc	Params(M)	FLOPs(G)
CvT-13	81.6	20	4.5
CoAtNet-0	81.6	25	4.2
GG-Transformer-T	82.0	28	4.5
CMT-S	83.5	25	4.0
GLiT-Small	80.5	25	4.4
CEFormer-S	**83.6**	28	**3.8**
ConViT-S+	82.2	48	10
ConViT-S+	82.2	48	10
CvT-21	82.5	32	7.1
LV-ViT-M	84.1	56	16.0
CMT-B	84.5	46	9.3
CEFormer-M	**84.5**	55	**6.8**
ConViT-B	82.4	86	17
LV-ViT-L	**85.3**	150	59.0
CMT-L	84.8	75	19.5
GLiT-Base	82.3	96	17
CEFormer-L	85.0	92	**14.1**

由表 4-6 可以看出，与这些卷积与 Transformer 相结合的模型相比，在指标 Top-1 Acc 方面，CEFormer-S 与 CEFormer-M 取得了最优结果，CEFormer-L 取得了次优结果。其中 CEFormer-S 相较于次优模型 CMT-S 提升了 0.1%。CEFormer-M 与 CMT-B 相同，均为同系列最优。CEFormer-L 相较于最优模型 LV-ViT-L 落后了 0.3%。可以看出对于这个指标而言，CEFormer 取得了较优结果，与目前最优的模型相比不分伯仲。

而在指标 FLOPs 方面，CEFormer-S，CEFormer-M 与 CEFormer-L 均取得了最优结果，各自相比次优模型提升了 5%，4%和 17%。可以看出与同样引入了卷积的视觉 Transformer 模型相比，注意力机制改进之后的优势就体现出来了，这方面 CEFormer 大幅度领先同类项的模型。

4.5.8 目标检测实验

本小节实验使用目标检测器 Mask R-CNN 在 COCO 数据集上训练，并将其骨干网络替换为不同的模型，然后在与模型 CEFormer 对比指标 mAP 和 FPS。实验结果如表 4-7 所示，不同尺寸的模型以底纹颜色区分，加粗数据为同系列最优。

表 4-7 Mask R-CNN 下不同模型的 mAP 和 FLOPs

模型	mAP	Params(M)	FLOPs(G)
ResNet50	38.0	44	260
Swin-T	43.7	48	264
Twins-S	42.7	44	228
RegionViT-S+	44.2	51	**183**
CSWin-T	46.7	42	279
CEFormer-S	**46.9**	48	201
ResNet101	41.5	63	336
Twins-B	45.1	76	340
RegionViT-B+	45.4	93	**307**
CSWin-S	47.9	54	342
CEFormer-M	**48.1**	64	326
Twins-L	45.2	120	474
Focal-B	47.8	110	533
CSWin-B	48.7	97	526
CEFormer-L	**48.8**	102	**434**

由表 4-7 可以看出，针对这些模型而言，在指标 mAP 方面，CEFormer-S，CEFormer-M 与 CEFormer-L 均取得了最优结果，各自相比次优模型均提升了 0.4%。可以看出通过引入了不同特性之后，模型 CEFormer 确实有较大的提升

从指标 FLOPs 来看，CEFormer-L 取得了最优结果，与次优模型 Twins-L 相比提升了 8.4%。CEFormer-S 取得了次优结果，与最优模型 RegionViT-S+相比落后了 10%，与第三优模型 Twins-S 相比领先了 12%。CEFormer-M 取得了次优结果，与最优模型 RegionViT-B+相比落后了 6%，与第三优模型 ResNet101 相比领先了 3%。可以看出本章提出的模型 CEFormer 的前向推理的计算量 FLOPs 取得了较好结果。

4.6 本章小结

本章针对 Transformer 模型训练需要极大的数据量，并且不稳定的问题，引入各种卷积来缓解相应问题。从不同角度结合卷积引入平移不变性，局部性，尺度不变性。最后利用一个轻量化卷积模块改变模型对输入图片的传统处理方式，从而提升稳定性。提出了高效 Transformer 模型 CEFormer，其不同尺寸各自取得了 ImageNet1k Top-1 数据集上 83.6%，84.5%，85.0%的精度。最后将其作为骨干网络，放入 Mask R-CNN 当中，与其他骨干网络模型对比在指标 mAP 方面取得了最优结果。

第五章 总结与展望

5.1 全文总结

图像特征提取在计算机视觉领域极其重要，无论对于任何视觉任务而言都是如此，它很大程度上决定了视觉任务最终的精度与速度。过去都是基于卷积神经网络去对图像进行特征提取，近年来，基于 Transformer 的图像特征提取方法成为了研究热点，但相关模型仍存在一些需要改进的地方。首先，任何基于 Transformer 的模型都先天存在一个瓶颈，也就是给定一系列由输入图像转化来的 Token 作为输入，Transformer 的自注意力机制通过将一个 Token 与其他所有 Token 关联起来迭代学习特征表示，这也导致了模型的时间与空间复杂度均与输入的 Token 数量成二次关系。这种二次复杂度阻止了基于视觉 Transformer 的骨干网络对高分辨率的图像建模，并且如此高昂的计算成本使其很难适用于边缘设备。其次，Transformer 在建模视觉结构时缺乏归纳偏置，导致其对大数据集进行预训练的依赖。最后，与卷积神经网络相比，Transformer 模型表现出低于标准的可优化性，特别是对于优化器，超参数的选择较为敏感，缺乏稳定性，收敛速度较慢。针对上述问题，本文主要工作如下：

（1）提出了两种加速 Transformer 模型的方法，分别从模型内部和外部两个角度去解决当前 Transformer 模型计算成本高，模型的复杂度与输入的 Token 数量成二次关系的问题。首先是将自注意力机制本身的二次复杂度降低为线性，从内部提高模型的处理速度。然后又提出了一个无参数，可以根据不同输入图片自适应采样从而筛掉不重要 Token 的轻量化剪枝方法，从外部减少无意义的输入。最后将两种方法合并得到了一种新的高效注意力机制（E-Attention）。实验表明，两种方法各自可降低原 Transformer 模型 30%-50%的计算量，而 E-Attention 可以减少原 Transformer 模型 60%-70%的计算量。

（2）在本文提出的 E-Attention 基础上，进一步结合深度卷积和空洞卷积，从平移不变性，局部性，尺度不变性三个角度引入 Transformer 模型缺乏的归纳偏置。然后再利用一个轻量化卷积模块改变了传统 Transformer 模型对输入图片的处理方式，从而加快收敛速度，提升了稳定性。最终得到了一个结合卷积的高效 Transformer 图像特征提取网络（CEFormer）。实验表明，CEFormer 在性能和运算速度之间均取得了良好的结果。

5.2 后续工作展望

本文先是从线性注意力和 Token 剪枝的角度去加速 Transformer 模型，然后又与卷积结合引入了多种特性优化 Transformer 模型。

首先对于 Transformer 模型的轻量化剪枝方法除了从 Token 维度考虑之外，还可以从注意力头，神经元等维度考虑剪枝，因此剪枝方法在后续工作中可以与其他维度联合在一起对 Transformer 模型进一步加速。另外目前也有一些工作从蒸馏、量化的角度去加速 Transformer，这也可以尝试进一步研究。

其次对于提升 Transformer 稳定性的研究，目前只是用一个轻量化的卷积模块去替代，这一点还可以继续深挖，设计一个更为合理有效的卷积形式对输入图片进行处理。

致　谢

首先感谢杨波老师和刘珊老师在论文写作过程中对我的指点，感谢自己的辛苦付出。

最后感谢自己的父母含辛茹苦将我抚养长大。

参考文献

[1] Kang K. Comparison of face recognition and detection models: Using different convolution neural networks[J]. Optical Memory and Neural Networks, 2019, 28(2): 101-108.

[2] 徐建亮, 周明安, 毛建辉, 等. 基于一种卷积神经式类网络的实时人脸识别方法研究[J]. 计算机科学与应用, 2020, 10(1): 11-20.

[3] Chen K, Tao W. Convolutional regression for visual tracking[J]. IEEE Transactions on Image Processing, 2018, 27(7): 3611-3620.

[4] Brunetti A, Buongiorno D, Trotta G F, et al. Computer vision and deep learning techniques for pedestrian detection and tracking: A survey[J]. Neurocomputing, 2018, 300: 17-33.

[5] Liu Z, Cai Y, Chen L, et al. Vehicle license plate recognition method based on deep convolution network in complex road scene[J]. Proceedings of the Institution of Mechanical Engineers Part D Journal of Automobile Engineering, 2019, 233(9): 2284-2292.

[6] 马志峰. 基于卷积神经网络的中文车牌识别[D]. 兰州: 兰州大学, 2020, 29-36.

[7] Fan J, Han F, Liu H. Challenges of big data analysis[J]. National science review, 2014, 1(2): 293-314.

[8] 周飞燕, 金林鹏, 董军. 卷积神经网络研究综述[J]. 计算机学报, 2017, 40(06): 1229-1251.

[9] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[J]. Advances in neural information processing systems, 2014, 27.

[10] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[11] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint arXiv:2010.11929, 2020.

[12] Touvron H, Cord M, Douze M, et al. Training data-efficient image transformers & distillation through attention[C]. International Conference on Machine Learning. PMLR, 2021: 10347-10357.

[13] Wang W, Xie E, Li X, et al. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 568-578.

[14] 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. 2021: 10012-10022.

[15] Yuan L, Chen Y, Wang T, et al. Tokens-to-token vit: Training vision transformers from scratch on imagenet[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 558-567.

[16] Wang W, Xie E, Li X, et al. PVTv2: Improved Baselines with Pyramid Vision Transformer[J]. arXiv preprint arXiv:2106.13797, 2021.

[17] Yue X, Sun S, Kuang Z, et al. Vision transformer with progressive sampling[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 387-396.

[18] Ryoo M S, Piergiovanni A J, Arnab A, et al. TokenLearner: What Can 8 Learned Tokens Do for Images and Videos?[J]. arXiv preprint arXiv:2106.11297, 2021.

[19] Islam M A, Jia S, Bruce N D B. How much position information do convolutional neural networks encode?[J]. arXiv preprint arXiv:2001.08248, 2020.

[20] Chu X, Tian Z, Zhang B, et al. Conditional positional encodings for vision transformers[J]. arXiv preprint arXiv:2102.10882, 2021.

[21] Dong X, Bao J, Chen D, et al. Cswin transformer: A general vision transformer backbone with cross-shaped windows[J]. arXiv preprint arXiv:2107.00652, 2021.

[22] Fan H, Xiong B, Mangalam K, et al. Multiscale vision transformers[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 6824-6835.

[23] Li Y, Wu C Y, Fan H, et al. Improved multiscale vision transformers for classification and detection[J]. arXiv preprint arXiv:2112.01526, 2021.

[24] Huang Z, Ben Y, Luo G, et al. Shuffle transformer: Rethinking spatial shuffle for vision transformer[J]. arXiv preprint arXiv:2106.03650, 2021.

[25] Fang J, Xie L, Wang X, et al. Msg-transformer: Exchanging local spatial information by manipulating messenger tokens[J]. arXiv preprint arXiv:2105.15168, 2021.

[26] Xu Y, Zhang Q, Zhang J, et al. Vitae: Vision transformer advanced by exploring intrinsic inductive bias[J]. Advances in Neural Information Processing Systems, 2021, 34.

[27] Zhou J, Wang P, Wang F, et al. ELSA: Enhanced Local Self-Attention for Vision Transformer[J]. arXiv preprint arXiv:2112.12786, 2021.

[28] Ascoli S, Touvron H, Leavitt M L, et al. Convit: Improving vision transformers with soft convolutional inductive biases[C]. International Conference on Machine Learning. PMLR, 2021: 2286-2296.

[29] Heo B, Yun S, Han D, et al. Rethinking spatial dimensions of vision transformers[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 11936-11945.

[30] Wu H, Xiao B, Codella N, et al. Cvt: Introducing convolutions to vision transformers[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 22-31.

[31] Jiang Z H, Hou Q, Yuan L, et al. All tokens matter: Token labeling for training better vision transformers[J]. Advances in Neural Information Processing Systems, 2021, 34.

[32] Yu Q, Xia Y, Bai Y, et al. Glance-and-gaze vision transformer[J]. Advances in Neural Information Processing Systems, 2021, 34.

[33] Guo J, Han K, Wu H, et al. Cmt: Convolutional neural networks meet vision transformers[J]. arXiv preprint arXiv:2107.06263, 2021.

[34] Chen B, Li P, Li C, et al. Glit: Neural architecture search for global and local image transformer[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 12-21.

[35] Yan H, Li Z, Li W, et al. ConTNet: Why not use convolution and transformer at the same time?[J]. arXiv preprint arXiv:2104.13497, 2021.

[36] Liu Z, Hu H, Lin Y, et al. Swin Transformer V2: Scaling Up Capacity and Resolution[J]. arXiv preprint arXiv:2111.09883, 2021.

[37] Zhang Q, Yang Y B. Rest: An efficient transformer for visual recognition[J]. Advances in Neural Information Processing Systems, 2021, 34.

[38] Liu Z, Wang Y, Han K, et al. Post-training quantization for vision transformer[J]. Advances in Neural Information Processing Systems, 2021, 34.

[39] Yuan Z, Xue C, Chen Y, et al. PTQ4ViT: Post-Training Quantization Framework for Vision Transformers[J]. arXiv preprint arXiv:2111.12293, 2021.

[40] Li Z, Yang T, Wang P, et al. Q-ViT: Fully Differentiable Quantization for Vision Transformer[J]. arXiv preprint arXiv:2201.07703, 2022.

[41] Lan H, Wang X, Wei X. Couplformer: Rethinking Vision Transformer with Coupling Attention Map[J]. arXiv preprint arXiv:2112.05425, 2021.

[42] Zhu M, Han K, Tang Y, et al. Visual transformer pruning[J]. arXiv e-prints, 2021: arXiv: 2104.08500.

[43] Tang Y, Han K, Wang Y, et al. Patch slimming for efficient vision transformers[J]. arXiv preprint arXiv:2106.02852, 2021.

[44] Yang H, Yin H, Molchanov P, et al. Nvit: Vision transformer compression and parameter redistribution[J]. arXiv preprint arXiv:2110.04869, 2021.

[45] Yu H, Wu J. A unified pruning framework for vision transformers[J]. arXiv preprint arXiv:2111.15127, 2021.

[46] He H, Liu J, Pan Z, et al. Pruning self-attentions into convolutional layers in single path[J]. arXiv preprint arXiv:2111.11802, 2021.

[47] Hou Z, Kung S Y. Multi-Dimensional Model Compression of Vision Transformer[J]. arXiv preprint arXiv:2201.00043, 2021.

[48] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network[J]. arXiv preprint arXiv:1503.02531, 2015, 2(7).

[49] Buciluǎ C, Caruana R, Niculescu-Mizil A. Model compression[C]. Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. 2006: 535-541.

[50] Ba J, Caruana R. Do deep nets really need to be deep?[J]. Advances in neural information processing systems, 2014, 27.

[51] Jia D, Han K, Wang Y, et al. Efficient vision transformers via fine-grained manifold distillation[J]. arXiv preprint arXiv:2107.01378, 2021.

[52] Hassani A, Walton S, Shah N, et al. Escaping the big data paradigm with compact transformers[J]. arXiv preprint arXiv:2104.05704, 2021.

[53] Lu Z, Liu H, Li J, et al. Efficient transformer for single image super-resolution[J]. arXiv preprint arXiv:2108.11084, 2021.

[54] Ding M, Lian X, Yang L, et al. HR-NAS: searching efficient high-resolution neural architectures with lightweight transformers[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 2982-2992.

[55] Su X, You S, Xie J, et al. Vision transformer architecture search[J]. arXiv preprint arXiv:2106.13700, 2021.

[56] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25.

[57] Hubel D H, Wiesel T N. Receptive fields and functional architecture of monkey striate cortex[J]. The Journal of physiology, 1968, 195(1): 215-243.

[58] Rawat W, Wang Z. Deep convolutional neural networks for image classification: A comprehensive review[J]. Neural computation, 2017, 29(9): 2352-2449.

[59] 李彦冬, 郝宗波, 雷航. 卷积神经网络研究综述[J]. 计算机应用, 2016, 36(9): 2508-2515.

[60] 王红, 史金钏, 张志伟. 基于注意力机制的 LSTM 的语义关系抽取[J]. 计算机应用研究, 2018, 35(5): 1417-1420.

[61] 唐海桃, 薛嘉宾, 韩纪庆. 一种多尺度前向注意力模型的语音识别方法[J]. 电子学报, 2020, 48(7): 1255-1260.

[62] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.

[63] Lin Z, Feng M, Santos C N, et al. A structured self-attentive sentence embedding[J]. arXiv preprint arXiv:1703.03130, 2017.

[64] Wang X, Girshick R, Gupta A, et al. Non-local neural networks[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7794-7803.

[65] Shen Z, Zhang M, Zhao H, et al. Efficient attention: Attention with linear complexities[C]. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2021: 3531-3539.

[66] Peng H, Pappas N, Yogatama D, et al. Random feature attention[J]. arXiv preprint arXiv:2103.02143, 2021.

[67] Choromanski K, Likhosherstov V, Dohan D, et al. Rethinking attention with performers[J]. arXiv preprint arXiv:2009.14794, 2020.

[68] Tsai Y H H, Bai S, Yamada M, et al. Transformer Dissection: A Unified Understanding of Transformer's Attention via the Lens of Kernel[J]. arXiv preprint arXiv:1908.11775, 2019.

[69] Katharopoulos A, Vyas A, Pappas N, et al. Transformers are rnns: Fast autoregressive transformers with linear attention[C]. International Conference on Machine Learning. PMLR, 2020: 5156-5165.

[70] AUEB T R C. One-vs-each approximation to softmax for scalable estimation of probabilities[J]. Advances in Neural Information Processing Systems, 2016, 29.

[71] Gao B, Pavel L. On the properties of the softmax function with application in game theory and reinforcement learning[J]. arXiv preprint arXiv:1704.00805, 2017.

[72] Jang E, Gu S, Poole B. Categorical reparameterization with gumbel-softmax[J]. arXiv preprint arXiv:1611.01144, 2016.

[73] Clark K, Khandelwal U, Levy O, et al. What does bert look at? an analysis of bert's attention[J]. arXiv preprint arXiv:1906.04341, 2019.

[74] Kovaleva O, Romanov A, Rogers A, et al. Revealing the dark secrets of BERT[J]. arXiv preprint arXiv:1908.08593, 2019.

[75] Rao Y, Zhao W, Liu B, et al. Dynamicvit: Efficient vision transformers with dynamic token sparsification[J]. Advances in neural information processing systems, 2021, 34.

[76] 王江艳. 累积分布函数的光滑同时置信带[D]. 苏州: 苏州大学, 2013, 1.

[77] Han K, Xiao A, Wu E, et al. Transformer in transformer[J]. Advances in Neural Information Processing Systems, 2021, 34.

[78] Touvron H, Cord M, Sablayrolles A, et al. Going deeper with image transformers[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 32-42.

[79] Chen T, Cheng Y, Gan Z, et al. Chasing sparsity in vision transformers: An end-to-end exploration[J]. Advances in Neural Information Processing Systems, 2021, 34.

[80] Correia G M, Niculae V, Martins A F T. Adaptively sparse transformers[J]. arXiv preprint arXiv:1909.00015, 2019.

[81] Yuan L, Chen Y, Wang T, et al. Tokens-to-token vit: Training vision transformers from scratch on imagenet[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 558-567.

[82] Zhang H, Duan J, Xue M, et al. Bootstrapping ViTs: Towards Liberating Vision Transformers from Pre-training[J]. arXiv preprint arXiv:2112.03552, 2021.

[83] Cheng Y, Lu F. Gaze estimation using transformer[J]. arXiv preprint arXiv:2105.14424, 2021.

[84] Chen Z, Xie L, Niu J, et al. Visformer: The vision-friendly transformer[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 589-598.

[85] Chollet F. Xception: Deep learning with depthwise separable convolutions[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1251-1258.

[86] Yu F, Koltun V, Funkhouser T. Dilated residual networks[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 472-480.

[87] Mohamed M, Cesa G, Cohen T S, et al. A data and compute efficient design for limited-resources deep learning[J]. arXiv preprint arXiv:2004.09691, 2020.

[88] Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 1314-1324.

[89] Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 4510-4520.

[90] Chen X, Xie S, He K. An empirical study of training self-supervised visual transformers[J]. arXiv e-prints, 2021: arXiv: 2104.02057.

[91] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[92] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[93] Radosavovic I, Kosaraju R P, Girshick R, et al. Designing network design spaces[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 10428-10436.

[94] Zhou D, Kang B, Jin X, et al. Deepvit: Towards deeper vision transformer[J]. arXiv preprint arXiv:2103.11886, 2021.

[95] Chen M, Peng H, Fu J, et al. Autoformer: Searching transformers for visual recognition[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 12270-12280.

[96] Zhang Z, Zhang H, Zhao L, et al. Aggregating nested transformers[J]. arXiv preprint arXiv:2105.12723, 2021.

[97] Yang J, Li C, Zhang P, et al. Focal self-attention for local-global interactions in vision transformers[J]. arXiv preprint arXiv:2107.00641, 2021.

[98] Wang W, Yao L, Chen L, et al. Crossformer: A versatile vision transformer based on cross-scale attention[J]. arXiv e-prints, 2021: arXiv: 2108.00154.

攻读硕士学位期间取得的成果