Vision Transformer 和 ResNet 在 CIFAT10 数据集上的图像分类任务的研究

李文彬

（江南大学人工智能与计算机学院 无锡 085404）

摘　要　本文研究了 Vision Transformer和 ResNet 在 CIFAR-10 数据集上的图像分类任务中的表现。尽管 ViT 在大规模数据集上已显示出显著的优势，但在 CIFAR-10 这一相对较小的数据集上，ViT 的表现明显逊色于传统的卷积神经网络架构 ResNet。本文通过实验验证了训练了 30 个 epoch 的 ViT-B-16 在 CIFAR-10 上的准确率约为 65%，而仅训练了 20 个 epoch 的 ResNet-18 的准确率为 88% 左右。我们分析了 ViT 和 ResNet 在结构上的主要差异，探讨了 ViT 在小数据集上训练时所面临的挑战，包括其对数据量的需求、局部特征捕捉能力的不足等。同时，本文还讨论了 Transformer 架构在小型数据集上的局限性，并对未来可能的改进措施进行了展望，尤其是如何通过预训练模型、数据增强来提升 ViT 的表现。实验使用了 PyTorch 框架，模型分别为 ViT-B-16 和 ResNet-18。本项目代码编写框架参考：pytorch-CycleGAN-and-pix2pix（*junyanz/pytorch-CycleGAN-and-pix2pix: Image-to-Image Translation in PyTorch*）.

关键词　图像分类，Vision Transformer，ResNet，CIFAR10

Research on Image Classification Tasks of Vision Transformer and ResNet on the CIFAR-10 Dataset

LI Wen-bin

(School of Computer Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 085404, China)

Abstract　This paper investigates the performance of Vision Transformer (ViT) and ResNet on the image classification task using the CIFAR-10 dataset. Although ViT has shown significant advantages on large-scale datasets, its performance on CIFAR-10, a relatively small dataset, is notably inferior compared to the traditional Convolutional Neural Network (CNN) architecture ResNet. Through experiments, we verified that a ViT-B-16 model trained for 30 epochs achieved an accuracy of approximately 65% on CIFAR-10, while a ResNet-18 model trained for only 20 epochs reached an accuracy of around 88%. We analyzed the key structural differences between ViT and ResNet and discussed the challenges faced by ViT when trained on small datasets, including its demand for a larger amount of data and limited ability to capture local features. Additionally, this paper discusses the limitations of the Transformer architecture on small datasets and provides perspectives on possible future improvements, particularly how pretraining models and data augmentation can enhance the performance of ViT. The experiments were conducted using the PyTorch framework, with the models being ViT-B-16 and ResNet-18.The code framework for this project is based on the structure of the pytorch-CycleGAN-and-pix2pix (*junyanz/pytorch-CycleGAN-and-pix2pix: Image-to-Image Translation in PyTorch*).

Keywords　Image Classification, Vision Transformer, ResNet, CIFAR10

1. 引言

伴随新时代硬件的发展，更庞大的算力得以提供，神经网络的研究又迎来了热潮。越来越多的目光聚焦于更大规模的数据和更深、更庞大的网络，通过构建合适的模型和不断增加算力训练学习，产生了越来越多大型的智能模型。但当我们把目光回转到计算机视觉领域，卷积神经网络在图像分类任务上的表现仍然值得肯定，尤其是 ResNet 的提出，有效地解决了深度网络中的梯度消失问题，成为了计算机视觉中的经典模型之一。

在此之后，Transformer 模型在自然语言处理领域展现了其超高的鲁棒性。于是就有研究者设想能否在尽可能较少改变 Transformer 模型的基础上让其在计算机视觉领域也能有出色表现，Vision Transformer 由此诞生。Vision Transformer 作为基于 Transformer 的图像分类模型，通过将图像划分为小块（patches），并将这些小块作为输入经过 Transformer 编码器处理，展现了与传统卷积神经网络截然不同的模型设计理念。ViT 在大规模数据集（如 ImageNet）上表现出了优异的性能，并成为了视觉任务中的一种新兴的研究方向。

ViT 在大规模数据集上表现优异，但相对的，其在小规模数据集上却较为逊色。由于 Transformer 架构缺乏空间结构感知能力，ViT 需要更多的数据来有效地学习图像特征，这使得其在小规模数据集上训练时面临诸多挑战。而传统的卷积神经网络，如 ResNet，则能够较好地适应小数据集的训练，且在有限的训练数据下能够快速收敛并取得较高的准确率。

本研究旨在探讨 Vision Transformer 和 ResNet 在小规模数据集上的图像分类性能，分析两者在小数据集上训练的优势与劣势。通过对比 ViT 和 ResNet 在 CIFAR-10 数据集上的表现，我们希望能够为未来的模型设计提供有益的见解，特别是在处理小数据集时如何提高 Transformer 架构的性能。

2. 相关工作

近年来，深度学习在图像分类领域取得了显著进展，尤其是在卷积神经网络（CNN）和基于 Transformer 的模型（如 Vision Transformer, ViT）方面。其中也不乏有研究致力于比较不同网络架构在标准数据集上的表现，尤其是在 CIFAR-10 等小型数据集上。

ResNet（残差网络）的提出旨在解决神经网络训练中常见的两个问题：梯度消失和梯度爆炸[1]。该模型引入了残差块，使得信息能够在深层网络中有效传递。ResNet 在许多图像分类任务中取得了很好的效果，尤其是在小型数据集（如 CIFAR-10）上，具有较强的局部特征提取能力。He 等人的研究表明，深层网络通过残差结构能够避免过拟合，提高模型的训练效率[1]。许多后续工作也采用了 ResNet 作为基准模型，并在其基础上进行了改进[2]。

在 CIFAR-10 数据集上，ResNet-18 模型以其高效的训练方式和良好的分类性能，成为标准的基准方法之一[3]。这些研究表明，ResNet 网络通过深度卷积结构，能够有效地学习图像的局部特征，并在小数据集上表现优异。

随着 Transformer 模型在自然语言处理任务中的成功，ViT 被提出作为一种新的图像分类方法[4]。与传统的 CNN 模型不同，ViT 将图像分割成固定大小的补丁（patches），并将其作为输入序列传递到 Transformer 模型中进行处理。这一创新的设计使得 ViT 能够捕捉长程依赖关系，提升了模型对全局信息的理解能力。Dosovitskiy 等人首先提出了 ViT，并证明了该模型在大规模数据集（如 ImageNet）上的卓越表现[4]。

然而，在小数据集上，ViT 的表现往往不如 CNN，主要原因是其缺乏归纳偏执，对大量数据的需求较高。Touvron 等人的研究发现，ViT 在小型数据集（如 CIFAR-10）上的表现远逊色于卷积神经网络，尤其是在缺乏足够训练数据时，ViT 容易出现过拟合[5]。此类问题通常通过对 ViT 模型进行预训练或数据增强来缓解，但这些方法会显著增加训练成本和计算复杂度。

ViT 和传统的 CNN 架构之间的比较引起了广泛关注。近期的研究表明，ViT 在大数据集上优于 CNN，但在小数据集上的表现往往受到数据量限制。Chen 等人的研究表明，ViT 在 CIFAR-10 数据集上的分类准确率远不如 ResNet[6]。同时，研究也表明，当数据集规模较小时，卷积神经网络能够更有效地提取局部特征，而 ViT 则可能受到数据稀缺性的限制，表现不佳。

在更深入的实验中，Tran 等人进一步验证了 ViT 的计算资源消耗和训练时间相对较长的问题，这使得 ViT 在小数据集上的应用受到限制[7]。因此，尽管 ViT 在理论上能够捕捉更复杂的长程依赖关系，但其在小数据集上的表现仍需要通过模型优化和数据增强来改进。

3. 相关方法

这一章节主要介绍 ResNet 和 Vision Transformer 模型的架构和工作原理。同时还详细介绍了模型中用到的一些关键技术。最后补充了使用模型和相关技术时需要注意的一些细节。

3.1. 深度残差学习

在没有引入深度残差学习的概念之前，增加神经网络的层数是成为习得更多特征的常见方法。但仅在增加网络的层数后，模型训练和测试的错误率反而提高了。如下图图 1 所示，层数更多的神经网络在训练和测试上的错误率明显高于浅层神经网络。

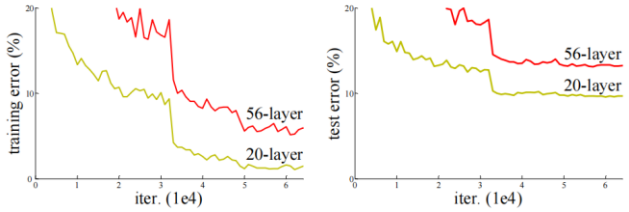

图 1 不同层数的 CNN 训练和测试时的错误率

上面的现象说明 CNN 的优化并不只是增加卷积层这么简单。于是我们设想在层数较少的网络上增加恒等映射层，以此作为该网络的深层版。理论上通过这种方法应该至少能保证深层网络的训练错误率不高于浅层的网络，但实际上当前并没有办法能够很好地证明这一点。

传统的神经网络是通过优化目标函数来学习输入到输出的映射关系，例如，对于某个输入 xxx，我们期望神经网络能够学习到一个映射 $H(x)$，使得 $H(x)$ 尽量接近目标输出。然而，在深度网络中，直接拟合复杂的映射关系 $H(x)$ 难度较大。

深度残差学习提出了残差映射的概念，假设目标输出是 $H(x)$，而网络不再直接学习这个映射。相反，网络通过学习输入与输出之间的残差（residual），即 $F(x) := H(x) - x$，来更容易地优化目标函数。这种转化形式使得网络优化更加高效。

如下图 2 所示，为了实现残差学习，深度残差网络引入了残差连接。在一个基本的残差模块中，输入 x 经过若干卷积层、激活函数和批量归一化层后，输出 $F(x)$ 将与原始输入 x 相加，得到最终的输出 $y = F(x) + x$。这种残差连接的关键思想是，网络不直接学习输入到输出的完整映射，而是通过学习输入与输出之间的差异（残差）来优化网络。这意味着，如果网络没有有效的映射，残差部分就会变成恒等映射 $F(x) = 0$，这样网络能够直接通过短路连接（identity shortcut connection）保留原始输入 x。

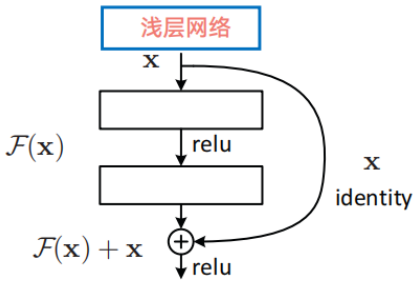

图 2 一个残差块

下面在 ImageNet 数据集上对深度残差神经网络进行了评估，如图 3 所示：

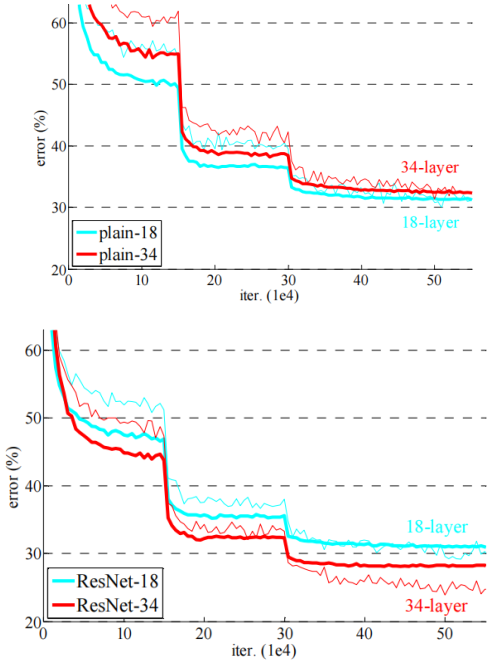

图 3 原始和残差神经网络在 ImageNet 上的训练和验证错误率

上图展示了未加入残差块的 CNN 和 ResNet 在 ImageNet 数据集上测试和验证过程中的错误率变化情况。其中细线表示训练误差，粗线表示验证误差。我们可以看到，ResNet 更容易优化，而普通的 CNN 训练错误率更高。同时在残差神经网络中添加层数确实可以很好地提升准确率。

3.2. ResNet

ResNet 的架构基于传统的卷积神经网络架构 VGG，但它通过残差块引入了跳跃连接。每个残差块的输入通过跳跃连接直接与其输出相加，使得信息可以直接从较浅的层传递到较深的层。这样，在每个残差块中，网络不仅学习输入到输出的映射，还学习了输入到输出的残差。

ResNet 的基本组成单元是残差块，在深层网络中，这些残差块堆叠在一起，形成了完整的网络结构。在经典的 ResNet 中，一个基本残差块通常由以下几部分组成：

1. 卷积层：通过卷积操作提取特征。
2. 批量归一化：对卷积层的输出进行标准化，改善训练稳定性，加速收敛。
3. ReLU 激活函数：非线性激活函数，增加网络的表达能力。
4. 跳跃连接：直接将输入 x 加到输出 $F(x)$ 上，形成 $y = F(x) + x$，这个操作可以通过一个 shortcut connection 来实现。

基于图 4 中未加入残差连接的神经网络，我们加入快捷连接（shortcut connection）使其变为残差神经网络。当输入维度和输出维度相同时，这种基于恒等映射的 shortcut 可以直接使用（图 4 中的实线），如下公式（1）所示：

$$\mathbf{y} = F(\mathrm{x}, \{W_i\}) + x \tag{1}$$

当输出的维度增加时（图 4 中的虚线），我们考虑两种方法：（1）shortcut 仍然执行恒等映射，并为增加维度填充额外的零项。此选项不引入任何额外的参数；（2）添加一个卷积层作为投影，使这个卷积层的输入和输出分别对应原来和方法后的维度，如下公式（2）所示：

$$\mathbf{y} = F(\mathrm{x}, \{W_i\}) + W_s\mathrm{x} \tag{2}$$

在这两种情况下，特征映射的步幅都为 2。

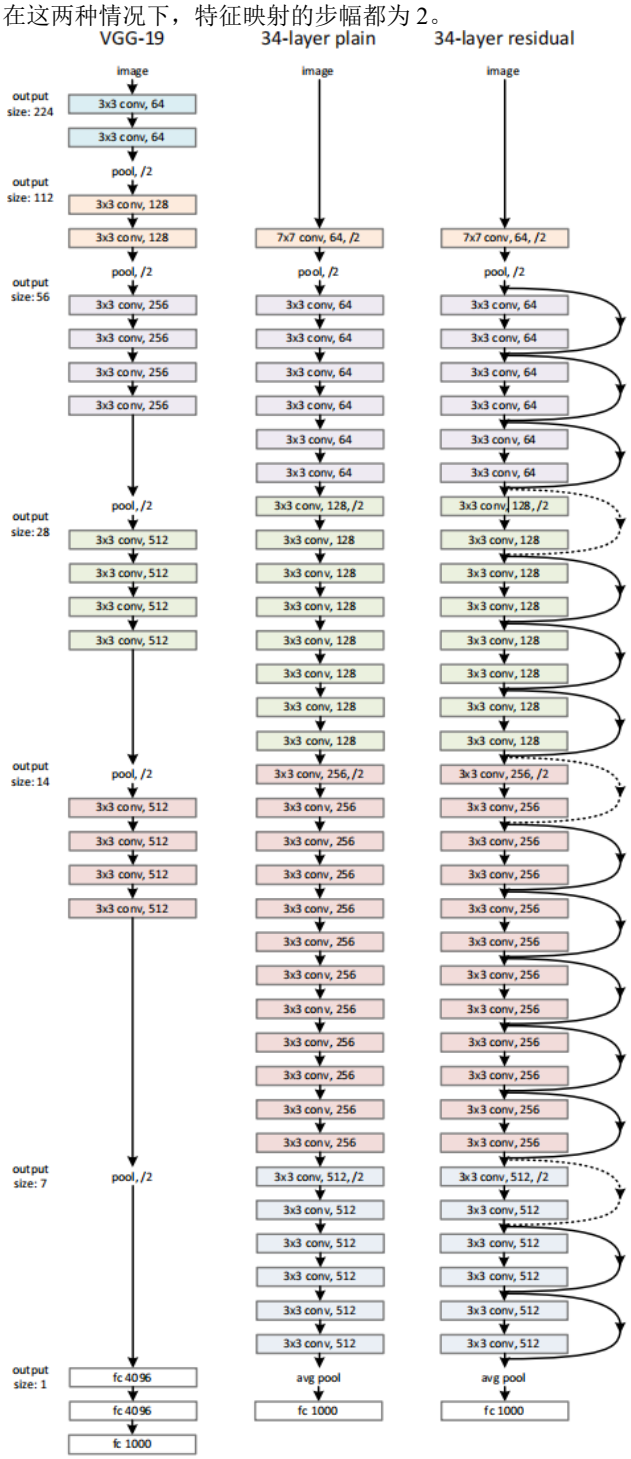

图 4 VGG-19 和含有 34 层的普通和残差神经网络结构图

ResNet 提出了不同深度的变体，其中，ResNet-50 是一种常见的变体，它在每个残差块中引入了瓶颈结构，即用 1×1 的卷积层来减少特征图的通道数，从而减少计算量，提升计算效率。表 1 详细列出了其他不同规模的 ResNet 变体。

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	[3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

表 1. ResNet 变体的详细信息

3.3. 自注意力机制

在深度学习中，注意力机制通过为输入的各个部分（如词、图像块等）分配不同的权重（即注意力权重），使模型在处理时可以“聚焦”在最有用的信息上。这些权重通常是动态计算的，基于输入的内容或上下文决定。

加性注意力（Additive Attention）和点积注意力（Dot-Product Attention）都是常见的注意力机制，但相对于加性注意力，由于可以使用高度优化的矩阵乘法代码，点积注意力计算更快且在实际使用过程中能有效节省空间，因此 Transformer 中的自注意力机制使用的就是点积注意力。为了防止在乘法计算过程中产生的数值过大，Transformer 还对点积注意力进行了缩放，具体公式如下：

$$Attention(Q,K,V)=softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

其中，Q 为查询矩阵，K 为键矩阵，V 为值矩阵，V 的维度为 d_k。

如下图 5 所示，对点积后的值经过 Mask 处理，mask 常设置为一个很小的负值近似 -inf。对索引 i 开始的输出经过以上运算会转化为一个极小值，后在 *softmax* 的计算中，掩蔽的部分会被赋予一个极低的得分，确保在计算注意力权重时这些位置的贡献为 0。这样一来就保证第 i 个输出的结果至于前 i−1 个结果有关，不会观测未来的值。Mask 的结果经过 softmax 函数激活后再与 V 进行点积就完成了缩放点积注意力。

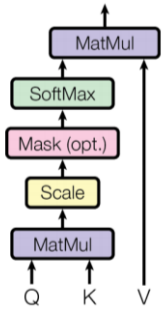

图 5 缩放点积注意力工作原理图

Multi-Head Attention 会将 V，K，Q 分别投影 h 次到更低的维度。假设对于每个 head，Q 和 K 的维度为 d_k，V 的维度为 d_v。这 h 个 head 会分别并行通过缩放点积注意力机制。在多头注意力中经过 h 个缩放点积注意力会生成 h 个维度为 d_v 的输出。将这些输出合并，最终能得到一个与原先 V 相同维度大小的结果，再通过一次线性投影就能获得最终的结果。Multi-Head Attention 的公式如下：

$$\begin{aligned} Mutihead(Q,K,V) &= Concat(head_1, head_2 \ldots)W_0 \\ where\ head_i &= Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \end{aligned} \tag{4}$$

其中，W_0，$W\_i\hat{}Q$，$W\_i\hat{}K$，$W_i\hat{}V$都是参数矩阵。多头注意力机制的工作原理图如下图 6 所示：

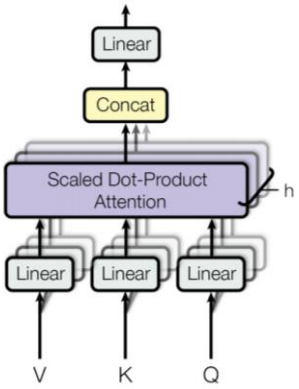

图 6 多头注意力机制工作原理图

3.4. Vision Transformer

Vision Transformer 的模型概述如下图 7 所示，在传统的 CNN 模型中，图像会直接作为输入经过卷积层处理。而在 Vision Transformer 中，图像首先会被切分成固定大小的非重叠小块（patches）。假设输入的图像为 $x\in\mathbb{R}^{H\times W\times C}$，其中，$H$ 为图像的高度，W 是图像的宽度，C 是图像的通道数。ViT 会将图像划分为 $P\times P$ 大小的小块（patch），每个小块的大小为 $P\times P\times C$。

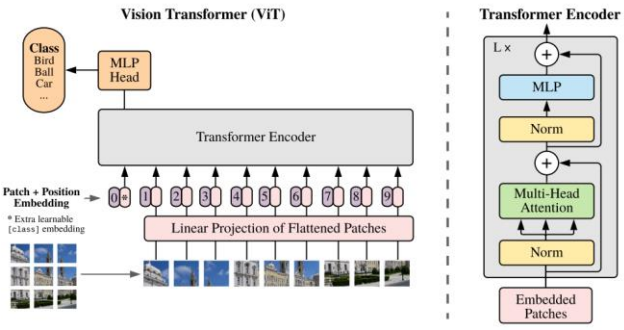

图 7 Vision Transformer 模型概况

这些图像块（patches）被扁平化（flatten），并通过一个线性投影映射到一个新的嵌入空间。每个图像块的嵌入向量大小为 D，其中 D 是 Transformer 中使用的潜在向量的维度。公式如下：

$$z_p = Linear\left(Flatten\left(Patch_p\right)\right) \tag{5}$$

其中，$Patch_p$ 是每个图像块，z_p 是对应的嵌入向量。

与 BERT 模型使用的 [class] token 类似，我们在经过嵌入处理后的 patches 序列准备了一个可学习的嵌入 $(z_0^0 = \mathrm{x}_{class})$，它经过 Transformer 编码（$z_L^0$）后的输出代表整个图像的特征（由公式（9）所示）。在预训练和微调阶段，分类头均由 z_L^0，表示，该分类头是通过一个多层感知机（MLP）实现的。MLP 在预训练阶段包含一个隐藏层，而在微调阶段则由一个单一的线性层构成。

开始的设想是使用全局平均池化（GAP），对整个图像进行 embedding 操作后使用线性分类器来提取图像特征，但实际效果很差。后来发现并不是因为 CLS-Token，也不是因为 GAP 效果差，而是不同的方法需要设置合适的学习率，如下图 8 所示，使用 GAP 在学习率为 8e-4 的情况下模型在 ImageNet 上的准确率相较于使用 CLS-Token 明显较差；而调整学习率为 3e-4 后，模型的准确率能达到与 CLS-Token 相近的水平，甚至更高。

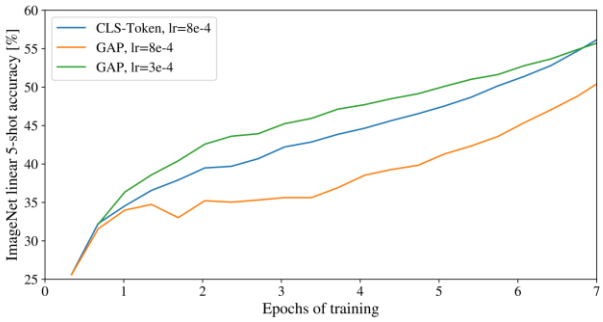

图 8 使用 CLS-Token 和 GAP 分类器的比较

由于 Transformer 模型本身不具备处理图像空间位置信息的能力，因此需要加入位置嵌入来保留图像块之间的空间关系。ViT 使用的是可学习的位置嵌入，这些嵌入向量被加到每个图像块的嵌入向量中，以提供位置信息。这些位置嵌入向量是与图像块的顺序相对应的，因此每个图像块的嵌入向量在加上对应的位置嵌入后，可以保持图像块的空间位置信息[9]。

如图 5 所示，Transformer 编码器由多层交替的多头注意力和 MLP 块组成（公式（7，8））[10]，在这两者前后分别会加入一个归一化层和残差连接。

$$z_0 = \left[\mathrm{x}_{class}; \mathrm{x}_p^1\mathbf{E}; \mathrm{x}_p^2\mathbf{E}; \ldots; \mathrm{x}_p^N\mathbf{E}\right] + \mathbf{E}_{pos}, \quad \mathbf{E} \in \mathbb{R}^{D\times(P^2\cdot C)}, \mathbf{E}_{pos} \in \mathbb{R}^{D\times(N+1)} \tag{6}$$

$$z'_\ell = MSA\left(LN(z_{\ell-1})\right) + z_{\ell-1}, \ell = 1 \ldots L \tag{7}$$

$$z_\ell = MLP(LN(z'_\ell)) + z'_\ell, \ell = 1 \ldots L \tag{8}$$

$$\boldsymbol{y} = LN(z_L^0) \tag{9}$$

其中，z_0 是 Transformer 编码器的输入，其中 x_{class} 为 CLS token，x_p^i 是第 i 个 patch 的数据，E 为 embedding 参数矩阵，E_{pos} 为位置参数矩阵。z'_ℓ 是第一子层的输出；z_ℓ 是第二子层的输出；z_L^0 是 CLS token 对应位置的结果，该结果经过一次 LN 就得到了输出 y 当做整体图像的特征。

3.5. 方法实现

由于 CIFAR10 数据集中的图片大小为 32×32 像素，而默认情况下，本实验采用的 VIT-Base-16 和 ResNet-18 模型的输入图片大小为 224×224，因此在将图像数据作为模型的输入之前需要对图片的大小统一为 224×224，本实验采用第三方库 Pytorch 的 Resize() 方法实现。此外，VIT-Base-16 和 ResNet-18 模型的默认输出特征数量为 1000，而 CIFAR10 数据集的类别只有 10，因此还需要对两模型的结构进行修改。想要输出的结果为对应的十分类，通常有两种方法：（1） 修改模型的输出层（全连接层）使其输出的特征数为 10；（2） 在模型的输出层后再添加一个线性层，将输出的 1000 特征值经过线性层后转化为 10 个特征值。本实验中，VIT-Base-16 和 ResNet-18 都采用了方法 1。

训练参数方面，两模型训练的 batch size 都是 64，VIT-Base-16 的学习率为 1e-4，ResNet-18 开始的学习率为 1e-3，后又设置为 1e-4 训练了多个 epoch。数据预处理阶段，将图片大小 Resize 到 224×224 后，对图片进行翻转，接着将图片转化为向量并标准化。

4. 实验过程及分析

该章节主要介绍 VIT 和 ResNet 模型在 CIFAR10 小规模数据集上的图像分类表现。VIT 选用的模型为 VIT-Base-16，ResNet 选用 ResNet-18 模型，实验在较为有限的资源和时间内实现。尽管最终的模型并没有达到原论文的表现力，但能够一定程度上的看到两模型在小规模数据集上的训练趋势和区别，整体实验结果符合原论文提出的效果。

4.1. 数据集

本课题所做的研究均在 CIFAR-10 数据集上进行，该数据集包含了 60,000 张 32×32 像素的彩色图像，图像分布在 10 个类别中。每个类别包含 6,000 张图像。数据集的详细信息如下：

- 训练集（Training Set）：50,000 张图像
- 测试集（Test Set）： 10,000 张图像

每张图像的大小为 32×32 像素，图像为彩色图像，即每张图像由 3 个通道（红色、绿色、蓝色，RGB）组成。图像的标签为整数 0~9 分别代表这 10 个不同的类别。数据集链接：*https://www.cs.toronto.edu/~kriz/cifar.html.*

4.2. 数据读取和预处理

在 CIFAR10 官网下载完数据集后，使用 Transform 对数据进行预处理。预处理步骤主要包括：调整图片大小为 224×224，将图像进行翻转，将图像转化为 Tensor 和标准化 Tensor。下面展示的是一个经过图像预处理的例子，如图 9 所示，左图为原始图像，右图为经过 Transform 图像预处理后的图像。

图 9 图像预处理前后对比

本次研究做了相关消融实验来证明图像预处理的必要性。以 VIT-Base-16 为例，下图是模型在未经数据预处理时模型训练准确率随训练轮数的变化情况图。可以看到，在数据未经过处理的情况下，模型训练起步的准确率相当低，且在训练后模型的准确率提升的特别慢。而在经过处理的图像上训练的模型，准确率明显高于未经过处理的，能够更好地学习到图像的特征（如图 10 所示）。上图为未经数据处理的模型训练情况，下图为经过数据处理的模型训练情况。

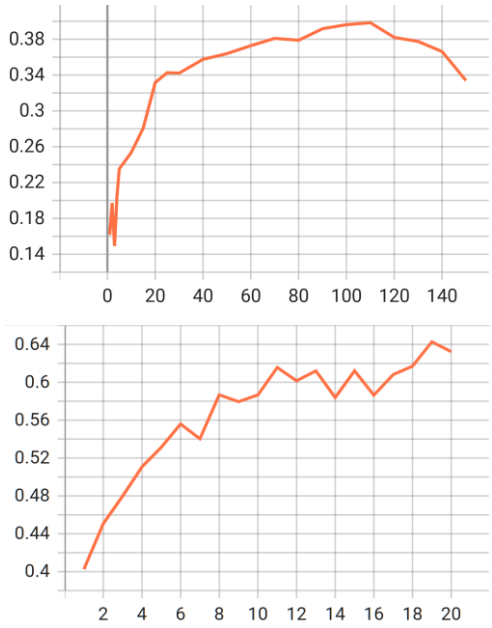

图 10 数据预处理对模型训练的影响

4.3. 模型构建

ResNet-18 是 ResNet 系列的一个较小版本，包含 18 层，它通过引入残差连接解决了深度网络中的梯度消失问题。它主要由初始卷积层，残差块，分阶段残差块，全局平均池化和分类头五个部分组成。由于 CIFAR10 数据集包含有 10 个类，因此我们要修改原先模型的全连接层，使其输出的特征值数量为 10。

在开始时，输入的图像经过处理后会变成 224×224×3 的尺寸，并通过卷积层和最大池化层进行初步处理，从而生成初步的特征图。接下来，通过一个卷积层将输入图像映射到 64 个特征维度，这一卷积层的卷积核大小为 7×7，步幅设置为 2，输出的特征图大小为 112×112×3。之后，图像进入 ResNet 的核心残差块，每个残差块由两个卷积层组成，每个卷积层后接批量归一化层和 ReLU 激活函数。此外，该部分包含两个 3×3 的卷积层，通过跳跃连接将输入图像直接与卷积层的输出相加，确保梯度能直接通过跳跃连接流动，避免梯度消失。接下来会经过分段残差块，该部分包含四个阶段，每个阶段都会输出不同维度的特征。在经过所有残差块后，使用全局平均池化层，将每个通道的特征图缩小为一个单一的值。通过对每个通道的输出进行平均，输出一个 512 维的向量。最后，通过全连接层将全局池化后的输出映射到类别空间。

Vision Transformer 通过使用 Transformer 架构来处理图像分类任务，其设计思想与传统的卷积神经网络（有很大不同。ViT 采用了一种新的方法，将图像分成小块（patches），然后用这些小块作为输入，经过 Transformer 编码器处理，最终得到图像的表示。

开始时，每个图像块会被展平，并通过一个线性嵌入（linear embedding）转换为一个固定维度的向量。接下来会对每个 patch 进行 embedding，将输入的图像划分成 $\frac{H}{P} \times \frac{W}{P}$ 个图像块，每个图像块的大小为 $P \times P$。对每个图像块，首先将其展平为一维向量，然后通过一个全连接层将其映射到嵌入空间，输出的维度是 D，这是模型的隐藏维度，在 VIT-Base-16 模型中 $D = 768$。然后要对每个 patch 进行位置编码，因为 Transformer 本身没有空间结构感知能力，所以必须使用位置编码来保持输入图像块的空间信息。通常使用正弦和余弦函数来生成位置编码，并将其加到每个图像块的嵌入向量中。位置编码的大小与图像块的维度一致。接着就要进入 Transformer 编码器，ViT 的编码器由多个 Transformer 层堆叠而成，本模型中编码器有 12 层。在 Transformer 编码器的最后一层，输出的特征是多个图像块的表示。通常选择[CLS]标记的输出作为图像的最终表示。最后，使用一个全连接层将该表示映射到类别空间，输出图像的类别预测。

下表 2 展示了不同规模的 ViT 模型构建的细节。

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

表 2 Vision Transformer 变体的详细信息

4.4. 模型编译

该部分主要确定模型采用的损失函数和优化器。由于该任务是多分类问题，使用交叉熵损失函数能有效地衡量预测类别概率与真实标签之间的差距。与平方误差损失相比，交叉熵损失对类别不平衡问题更为稳定，不容易让模型在某个类别上过度拟合，尤其是在数据集中某些类别的样本数很少的情况下。其定义如下：

$$L = -\sum_{i=1}^{N} y_i \log p_i \tag{10}$$

其中，y_i 是真实标签的概率分布；p_i 是模型预测的概率分布。

本实验优化器采用 Adam (Adaptive Moment Estimation)

优化算法，它结合了动量和自适应学习率的优点，能够更高效地进行梯度下降。Adam 优化器不仅能加速训练过程，还能在训练初期提供更加稳定的学习过程。它通过对每个参数维护一个均值和方差的估计，能够自适应地调整每个参数的更新步长，从而避免了传统 SGD 在学习过程中可能产生的大幅度波动。

4.5. 模型训练

模型训练的数据集大小为 50000，每个训练的 batch size 为 64。VIT-Base-16 和 ResNet-18 的学习率分别设置为 1e-4 和 1e-3。模型训练的环境分别为 RTX 4090D 和 RTX 3060。ResNet-18 训练了 25 个 epoch，准确率趋于饱和；VIT-Base-16 训练了 150 个 epoch，准确率稳步上升，但速度较慢。

ResNet-18 模型在开始训练时就有较低的 loss，且在开始阶段 loss 下降较快，如下图 11 所示，横坐标为训练迭代次数，纵坐标为训练过程中 loss 的值。

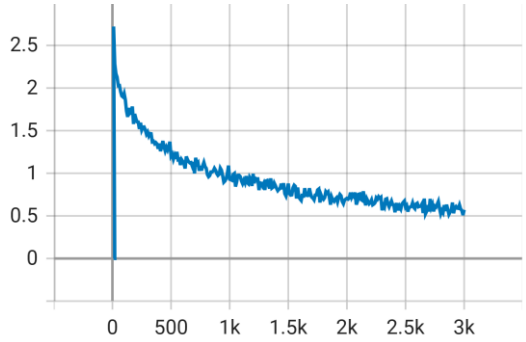

图 11 ResNet-18 训练时 loss 下降情况

ViT-Base-16 模型起步训练时的 loss 值较大，且整体下降较为缓慢，如下图 12 所示，横坐标为训练迭代次数，纵坐标为训练过程中 loss 的值。

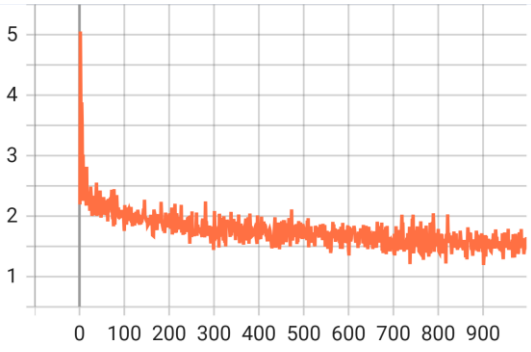

图 12 VIT-Base-16 训练时 loss 下降情况

从上面两图可以看出，ResNet-18 模型的 loss 下降速率略快于 ViT-Base-16，且起步的 loss 较低，但两者 loss 的整体趋势都在下降。

4.6. 模型预测和评估

本实验主要以准确率作为模型评估的主要依据，根据模型在测试集上的准确率来评估模型在图像分类任务上的好坏。模型的准确率通过预测准确的样本数量和预测总样本数量的比值来计算。但在代码的实现过程中，由于预测的结果 preds 和样本的标签 targets 维度相同，因此可以直接比较两者数值是否相等从而得到一个布尔类型的序列。其中，True 的个数即为模型预测准确的个数，又由于 True 和 False 代表的值分别为 1 和 0，因此通过对该布尔序列求和就能得到模型测准确的个数。

本实验测试集大小为 10000，实验评估并不使用所有的数据集样本，而是选择 100 个 batch 大小为 64 的样本进行测试，即 6400 个样本。

ResNet-18 测试结果如下图 13 所示，横坐标为模型训练的 epoch 数量，纵坐标表示模型在抽取训练样本上的准确率。可以看到，ResNet 模型在训练起步阶段准确率上升很快，在 CIFAR10 这种小规模数据集上有着较好的表现。当训练轮数达到 20 后，模型的准确率趋于稳定，大概在 88%到 89%之间。

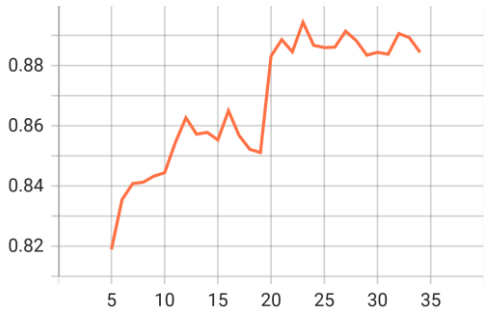

图 13 ResNet-18 准确率随 epoch 的变化情况

查看 ResNet 原论文中模型在 CIFAR10 上的测试情况（如下图 14 所示），可以看到与本实验规模相近的 ResNet-20 在 CIFAR10 上测试的错误率大概在 7% 到 8%左右，即准确率约在 92%到 93%，与本实验得到的准确率相近，实验达到预期效果。

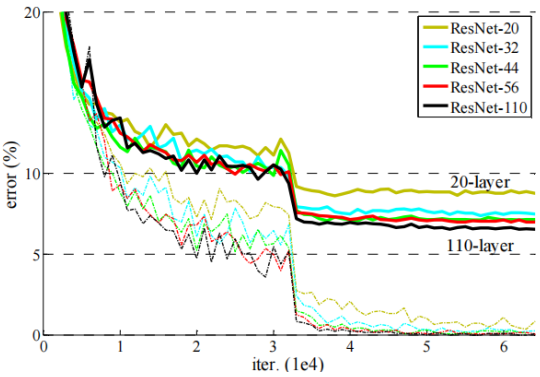

图 14 不同规模的 ResNet 模型在 CIFAR10 上的测试错误率

相比之下由于 ViT 模型训练收敛较慢，以及时间，设备等各方面的影响，本实验训练的 ViT-Base-16 模型没能够达到原论文中的效果。但从训练的整体趋势来看，ViT-Base-16 模型的准确率整体上成上升趋势，符合论文中预期的效果。如图 15 所示，ViT-Base-16 在前 10 轮训练中准确率以较快的速度提升，后面提升速度较慢，但整体呈上升趋势，经过 45 个 epoch 后准确率达到 65%左右。

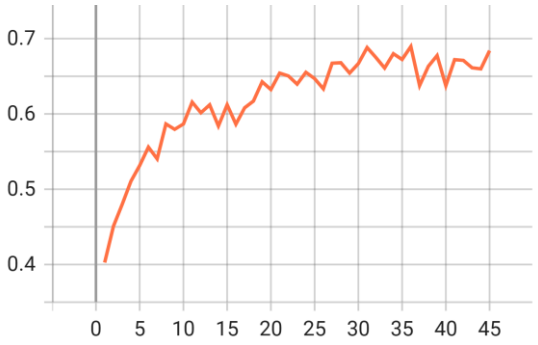

图 15 ViT-Base-16 准确率随 epoch 的变化情况

4.7. 实验结果分析

在本研究中，我们对 ViT-B-16 和 ResNet-18 模型在 CIFAR-10 数据集上的图像分类性能进行了比较。复现并验证 ViT 模型的特点及其在小规模数据集上局限性。实验通过使用 PyTorch 框架实现，我们对这两种模型进行了相同的训练和测试流程，结果表明，ViT 在 CIFAR-10 小型数据集上，其表现显著低于 ResNet。但从整体趋势上来看，ViT 模型的准确率整体保持上升趋势且没有饱和的迹象，而 ResNet 虽然通过较少的训练就能达到比较好的效果，但是其准确率很快就达到了饱和。

通过分析，实验结果也一定程度上符合 Visual Transformer 原论文中给出的结果。从下图 16 中可以看出，对于规模越大的 ViT 模型，其在越小的数据集上表现越差。因此我们从图中看到，ViT-B/16 的模型在 ImageNet 上能够取得最好的成绩，越能达到 75% 的准确率，但与 ResNet 构成的模型 BiT 相比有着显著差别。我们不难推断，在比 ImageNet 规模更小的数据集 CIFAR10 上，两者会有更大的差距。

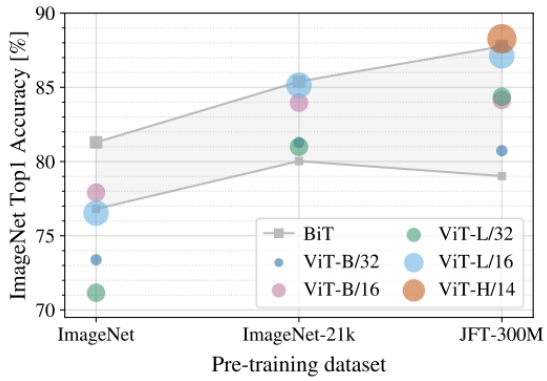

图 16 ViT 变体模型和 BiT 在不同规模数据集上的表现图

5. 结束语

本文研究了 Vision Transformer (ViT) 和 ResNet 在 CIFAR-10 数据集上的图像分类性能，并对比了这两种模型在小规模数据集上的训练效果。实验中，我们首先介绍了两种模型的结构和工作原理。ResNet-18 以传统卷积模型 VGG 为基础，通过引入残差连接解决训练和测试过程中的准确率下降问题。而 ViT 则采用 Transformer 架构，通过将图像划分为小块（patches）并通过 Transformer 编码器进行处理，展示了与传统卷积网络不同的设计理念。

在数据预处理方面，我们对 CIFAR-10 数据集进行了随机采样，并将图像大小调整、翻转和标准化处理。通过消融实验验证了数据预处理对模型训练过程的显著影响，尤其是在提高模型收敛速度和准确率方面。

实验结果表明，在 CIFAR-10 数据集上，尽管 ViT 在大规模数据集上表现出色，但在小规模数据集上的性能显著逊色于 ResNet。具体而言，经过 45 个 epoch 的训练，ViT-Base-16 的测试准确率约为 65%，而 ResNet-18 经过 25 个 epoch 后的准确率达到了 88%。这一结果表明，ViT 在处理小规模数据集时可能存在较大的数据需求和局部特征捕捉能力不足的问题。

通过与 ResNet 在 CIFAR-10 上的对比实验，本文还探讨了两种模型的优劣势。ResNet 在小规模数据集上表现稳定且高效，能够较快收敛并达到较高的准确率。相比之下，ViT 由于其模型结构的特殊性，需要更多的训练数据和更长时间的训练才能达到较为理想的结果。

最后，本文还对未来的研究方向进行了展望，特别是在 ViT 模型优化方面，提出了可以通过预训练、数据增强和更深层次的模型架构调整来改善其在小数据集上的表现。总的来说，本实验为深度学习领域中的图像分类任务提供了关于不同模型架构在小规模数据集上表现的有益见解，并为未来的研究提供了参考。

参考文献

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.

[2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity Mappings in Deep Residual Networks. Proceedings of the European Conference on Computer Vision (ECCV), 630-645.

[3] He, K., Zhang, X., & Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 1026-1034.

[4] Dosovitskiy, A., & Brox, T. (2016). Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 38(9), 1734-1747.

[5] Touvron, H., Cord, M., & Douze, M. (2021). Training data-efficient image transformers & distillation through attention. Proceedings of the International Conference on Machine Learning (ICML).

[6] Chen, M., Chen, X., & Hwang, T. (2021). A Comprehensive Evaluation of Vision Transformer. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[7] Tran, D., Wang, C., & Smolyansky, M. (2021). Vision Transformer: A Comprehensive Review. arXiv preprint arXiv:2104.06896.

[8] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[9] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., & Zhai, X. (2021). An Image is Worth 16x16 Words:

Transformers for Image Recognition at Scale. arXiv preprint arXiv:2010.11929.
[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NIPS, 2017.
[11] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. Learning deep transformer models for machine translation. In ACL, 2019.
[12] I. Bello, B. Zoph, Q. Le, A. Vaswani, and J. Shlens. Attention augmented convolutional networks.In ICCV, 2019.