图像识别的深度残差学习

何凯明　　张翔宇　　任小青　　孙坚
微软研究院
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

摘要

更深层次的神经网络更难训练。我们提出了一个剩余的学习框架来简化网络的训练，这些网络比以前使用的网络要深入得多。我们显式地将层重新构造为参考层输入的学习残差函数，而不是学习未引用的函数。我们提供了全面的经验证据表明，这些残差网络更容易优化，并能从相当大的深度获得准确性。ImageNet数据集我们评估剩余网的152 layers-8×比VGG网[40]但仍有较低的复杂度。这些残差网的集合在ImageNet测试集中达到了3.57%的误差。这个结果在ILSVRC 2015分类任务中获得了第一名。我们还对CIFAR-10进行了100层和1000层的分析。表示的深度对于许多视觉识别任务来说是至关重要的。仅仅由于我们的深度表示法，我们在COCO对象检测数据集上获得了28%的相对改进。深度残差网是我们提交给ILSVRC & COCO 2015 competitions1的基础，我们还在ImageNet检测、ImageNet定位、COCO检测和COCO分割方面获得了第一名。

1、介绍

深度卷积神经网络[22,21]在图像分类方面取得了一系列突破[21,49,39]。深度网络自然地以端到端多层的方式将低层/中层/高层特性[49]和分类器集成在一起，并且通过堆叠层(深度)的数量可以丰富特性的“级别”。最近的证据[40,43]表明网络深度至关重要，而富有挑战性的ImageNet数据集[35]的领先结果[40,43,12,16]都采用了“非常深”的[40]模型，深度为16[40]到30[16]。许多其他重要的视觉识别任务[7,11,6,32,27]也从非常深入的模型中获益良多。受深度重要性的驱动，一个问题出现了:学习更好的网络是否就像堆叠更多的层一样容易?回答这个问题的一个障碍是众所周知的梯度消失/爆炸问题[14,1,8]，它从一开始就阻碍了收敛。然而，这个问题主要通过规范化初始化[23,8,36,12]和中间规范化层[16]来解决，这使得具有数十个层的网络开始收敛于具有反向传播[22]的随机梯度下降(SGD)。

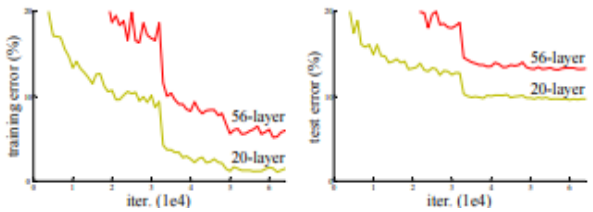

图1 在20层和56层“普通”网络的CIFAR-10上，训练错误(左)和测试错误(右)。网络越深，训练误差越大，测试误差越大。在ImageNet上类似的现象如图4所示

当较深的网络能够开始收敛时，一个退化问题就暴露出来了:随着网络深度的增加，精度达到饱和(这可能不足为奇)，然后迅速退化。出乎意料的是，这种退化并不是由过拟合引起的，在合适的深度模型中添加更多的层会导致更高的训练误差，正如文献[10,41]所报道的，我们的实验也完全验证了这一点。图1是一个典型的例子。

训练精度的下降表明并不是所有的系统都同样容易优化。让我们考虑一个更浅的体系结构和它的更深层的对等物，它在其中添加了更多的层。通过构建更深层次的模型，存在一种解决方案:添加的层是identity映射，其他层是从学习的较浅层次模型复制而来的。所构造的解的存在性表明，较深的模型与较浅的模型相比不会产生更高的训练误差。但实验表明，我们现有的求解器无法找到比构建的解更好或更好的解(或无法在可行的时间内这样做)。

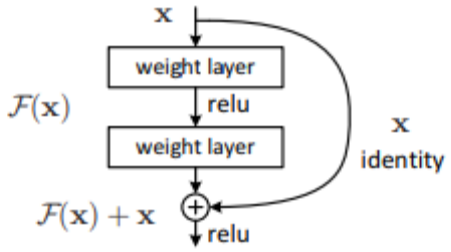

图2 残差学习:构建模块

本文通过引入深度残差学习框架来解决退化问题。我们并没有希望每个堆叠的层直接适合所需的底层映射，而是显式地让这些层适合残差映射。正式表示所需的底层映射为H(x),我们让堆叠非线性层适合另一个映射F(x):= H(x)−x。原始映射被重新转换为F(x)+x。我们假设优化残差映射比优化原始的、未引用的映射更容易。在极端情况下，如果身份映射是最优的，那么将残差推到零比通过一堆非线性层来匹配身份映射更容易。

F(x) +x的公式可以通过具有“快捷连接”的前馈神经网络实现(图2)。在我们的例子中，快捷连接仅仅执行identity映射，它们的输出被添加到堆叠层的输出中(图2)。整个网络仍然可以由SGD端到端的反向传播训练，并且可以很容易地使用公共库(例如Caffe[19])实现，而无需修改求解器。

我们在ImageNet[35]上进行了全面的实验来展示降解问题，并对我们的方法进行了评价。我们发现:1)我们的极深残差网很容易优化，但是当深度增加时，相应的“普通”网(简单的堆叠层)显示出更高的训练误差;2)我们的深度残差网可以很容易从大大增加的深度地享受精度收益，产生的结果比以前的网络好得多。

在CIFAR-10集合[20]上也出现了类似的现象，这表明我们的方法的优化困难和效果并不仅仅与特定的数据集相似。我们在超过100层的数据集中展示了成功训练的模型，并探索了超过1000层的模型。在ImageNet分类数据集[35]上，我们通过极深的残差网获得了很好的结果。我们的152层残差网络是在ImageNet上呈现的最深的网络，同时仍然比VGG net[40]的复杂度低。我们的集成在ImageNet测试集中有3.57%的前5个错误，并赢得了ILSVRC 2015分类竞赛第一名。在2015年ILSVRC & COCO竞赛中，极深的表现形式在其他识别任务上也具有出色的泛化性能，使我们在ImageNet检测、ImageNet定位、COCO检测、COCO分割等领域获得第一名。这一强有力的证据表明，残差学习原理是通用的，我们期望它能适用于其他视觉和非视觉问题。

2、相关工作

残差表示：在图像识别中，VLAD[18]是由残差向量对字典进行编码的表示，Fisher向量[30]可以表示为VLAD的概率版本[18]。它们都是用于图像检索和分类的强大的浅层表示[4,47]。对于矢量量化，编码残差向量[17]比编码原始向量更有效。在低级视觉和计算机图形学中，为了求解偏微分方程(偏微分方程)，广泛使用的多网格方法[3]将系统重新定义为多尺度的子问题，其中每个子问题负责粗尺度和细尺度之间的剩余解。多重网格的另一种替代方法是层次基础预处理[44,45]，它依赖于表示两个尺度之间残差向量的变量。已经证明[3,44,45]，这些解的收敛速度要比不知道解的剩余性质的标准解的收敛速度快得多。这些方法表明，良好的重新配方或预处理可以简化优化过程。

快捷方式连接：导致快捷连接的实践和理论[2,33,48]已经研究了很长时间。多层感知器(MLPs)的早期训练实践是将网络输入连接到输出的线性层添加到输出中[33,48]。在[43,24]中，一些中间层直接连接到辅助分类器，用于处理消失/爆炸梯度。文献[38,37,31,46]提出了通过快捷连接实现定心层响应、梯度和传播错误的方法。在[43]中，“初始”层由一个快捷分支和几个更深的分支组成。

3、深度残差网络

3.1、残差学习

让我们将H(x)看作是一个基础映射，由几个堆叠层(不一定是整个网络)来匹配，其中x表示第一个层的输入。如果假设多个非线性层可以渐近逼近复杂函数，那么就相当于假设它们可以渐近逼近残差函数，即H(x)−x(假设输入和输出是相同的尺寸)。因此而不是期望堆叠层近似H(x),我们明确地让这些层近似剩余函数F(x):= H(x)−x。最初的功能从而成为F(x)+ x。虽然两种形式都应该能够渐进地逼近所期望的函数(如假设的那样)，但学习的难度可能有所不同。

这种重新表述的动机是关于降解问题的反直觉现象(图1，左)。正如我们在介绍中所讨论的，如果添加的层可以构造为identify映射，那么更深层次的模型的训练错误应该不大于较浅的模型。退化问题表明，求解器在用多个非线性层近似表示identify映射时可能存在困难。在残差学习重构中，如果identify映射是最优的，解方可能会简单地将多个非线性层的权值驱动为零，以接近identify映射。

在实际情况下，identify映射不太可能是最优的，但我们的重新构造可能有助于预先处理问题。如果最优函数更接近于恒等映射而不是零映射，那么求解器就更容易找到与identify映射相关的扰动，而不是把函数作为一个新的函数来学习。我们通过实验证明(图7)，学习的残差函数一般都有很小的响应，说明identify映射提供了合理的预处理。

3.2、快捷identify映射

我们对每几个堆叠层都采用残差学习。一个构建块如图2所示。在形式上，本文将构建块定义为:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}. \qquad (1)$$

这里x和y是考虑的层的输入和输出向量。函数F(x，{Wi})表示要学习的残差映射。对于图2中的示例有两层,F = $W_2\sigma$ (W_1x),其中σ 表示ReLU[29]和偏置省略简化符号。操作F + x通过快捷连接和元素加法来执行。我们采用加法后的第二个非线性。σ (y),请参见图2)

等式(1)中的快捷连接既不增加参数，也不增加计算复杂度。这不仅在实践中具有吸引力，而且在我们比较普通网络和剩余网络时也很重要。我们可以比较同时具有相同数量的参数、深度、宽度和计算成本的普通/剩余网络(除了可忽略的元素加法)。

x和F的维数必须等于等式(1)如果不是这样(例如，当改变输入/输出通道时)，我们可以通过快捷连接执行线性投影W_s来匹配维数:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}. \qquad (2)$$

我们也可以在等式(1)中使用一个方阵W_s。但我们将通过实验证明，identify映射足以解决退化问题，而且是经济的，因此W_s只在匹配维度时使用。

残差函数F的形式是灵活的。本文的实验涉及到一个有两层或三层的函数F(图5)，而更多的层是可能的。但如果F只有一个单层，等式(1)类似于线性层:y = W_1x + x，对此我们没有观察到优势。我们还注意到，尽管上面的符号是为了简单起见而完全连接的层，但是它们适用于卷积层。函数F(x，{Wi})可以表示多个卷积层。元素级的添加是在两个feature map上一个通道一个通道执行的。

3.3、网络架构

我们测试了各种普通/残差网，并观察到了一致的现象。为了提供讨论的实例，我们将ImageNet的两个模型描述如下。

普通网络：我们的普通基线(图3，中间)主要是受VGG nets[40]的理念启发(图3，左)。卷积层主要有3×3滤波器器和遵循两个简单的设计规则:(i)为了大小相同的输出特性,各层具有相同数量的滤波器;并且(ii)如果feature map的大小减半，滤波器的数量就会翻倍，以保持每层的时间复杂度。我们直接通过卷积层进行下行采样，卷积层的步长为2。网络以全局平均池化层和使用softmax的1000路完全连接层结束。图3(中间)中加权层数为34。值得注意的是，与VGG nets[40]相比，我们的模型具有更少的滤波器和更低的复杂性(图3，左)。我们的34层基线有36亿次浮点运算(多层的)，仅占VGG-19(196亿次浮点运算)的18%。

残差网络：在上面的纯网络的基础上，我们插入了快捷连接(图3，右)，将网络转化为对应的残差版本。identify快捷直连(等式1)时可以直接使用输入和输出是相同的维度(实线快捷连接图3)。当维度增加(图3中虚线的快捷方式),我们认为两个选择:(A)快捷方式仍然执行identify映射,用额外的零增加维度的条目填充。这个选项不引入额外的参数;(B)投影在（等式2）捷径是用于匹配维度(由1×1旋转)。对于这两个选项，当快捷方式跨越两个大小的feature map时，它们的执行步长为2。

3.4、实现

我们的ImageNet实现遵循了[21,40]中的实践。将图像的短边随机采样[256,480]，进行缩放[40]。224×224作物从一个图像或其水平翻转,随机抽样与逐像素均值减去[21]。使用[21]中的标准颜色增强。我们在每次卷积后和激活前，在[16]之后，采用

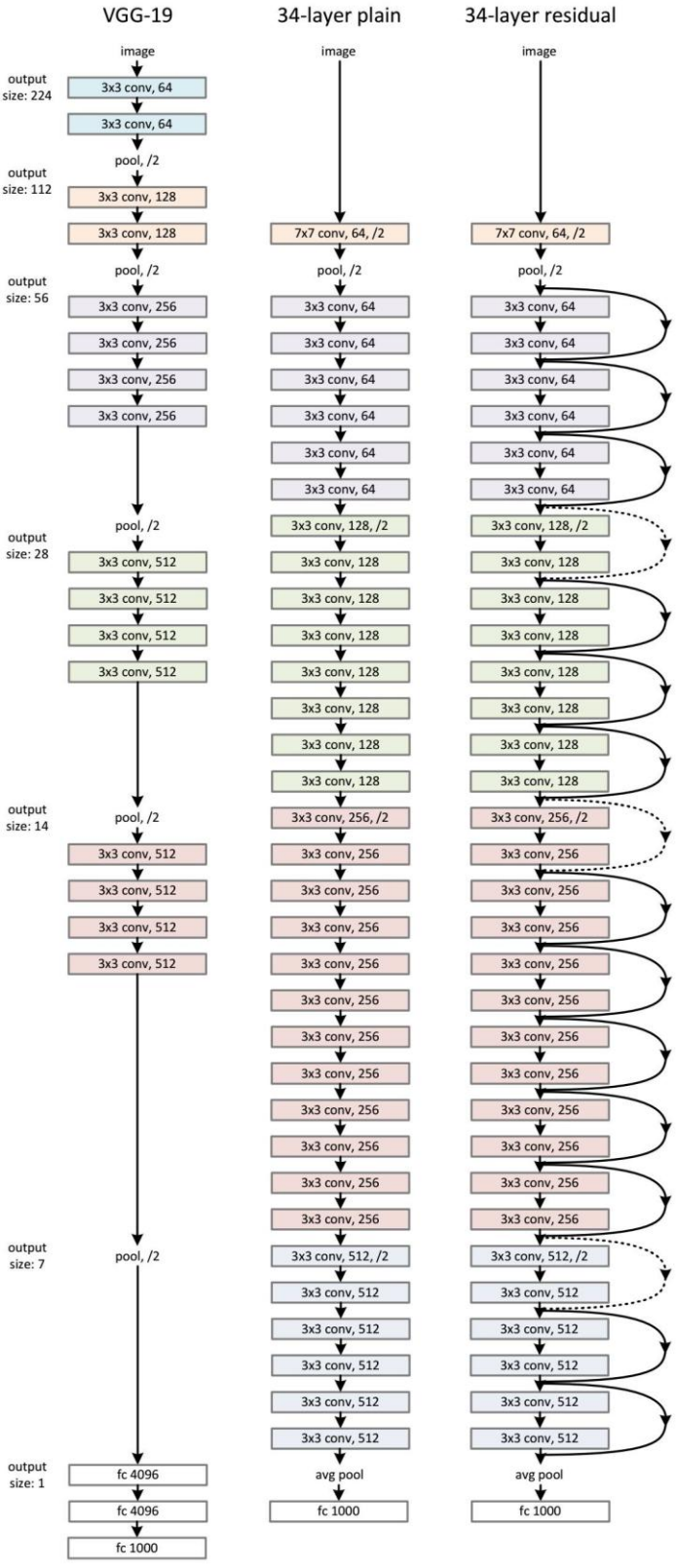

图3 ImageNet的网络架构示例。左:VGG-19模型[40](196亿浮点运算)作为参考。中间:纯网络与34层参数(36亿浮点运算)。右:残余网络与34层参数(36亿浮点运算)。虚线快捷键增加维度。表1显示了更多细节和其他变体

批正则化(BN)[16]。我们将权重初始化为[12]，并从头开始训练所有的纯/残差网络。我们使用的SGD的小批量大小为256。学习速率从0.1开始,除以10错误高原时,和模型训练长达60×10^4迭代。我们使用权重衰减0.0001和动量0.9。我们不使用dropout[13]，遵循[16]中的实践。在测试中，对于比较研究，我们采用了标准的10-crop测试[21]。为了获得最好的结果，我们采用了[40,12]中的完全卷积形式，并在多个尺度上平均得分(图像大小调整为{224,256,384,480,640})。

4、 实验

4.1、ImageNet分类

我们在ImageNet 2012分类数据集[35]上评估我们的方法，该数据集包含1000个类。模型在128万张训练图像上进行训练，并在50k验证图像上进行评估。我们还获得了测试服务器报告的100k测试映像的最终结果。我们评估前1和前5的错误率。

普通网络：我们首先评估18层和34层的普通网。图3(中间)为34层普通网。18层的普通网也是类似的形式。详见表1。

从表2的结果可以看出，较深的34层纯网络比较浅的18层纯网络具有更高的验证误差。为了揭示原因，在图4(左)中，我们比较了他们在训练过程中的训练/验证错误。我们观察到了退化问题——在整个训练过程中，34层的普通网络的训练误差更高，尽管18层素网络的解空间是34层素网络的子空间。

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	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$
	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 ImageNet架构。构建块显示在方括号中(也见图5)，块的数量是堆叠的。向下采样由conv3_1、conv4_1和conv5_1进行，步长为2

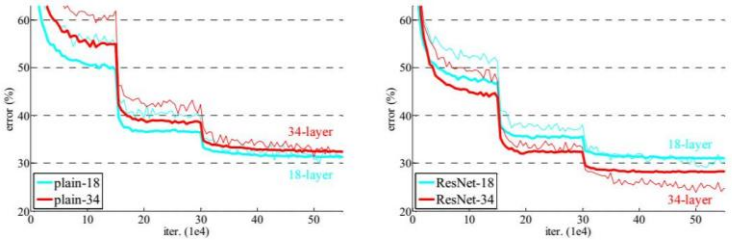

图4 ImageNet训练。细曲线表示训练误差，粗曲线表示中心作物验证误差。左:18层和34层的普通网络。右:18层和34层的ResNets。在这个图中，剩余网络与普通网络相比没有额外的参数。

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

表2。在ImageNet验证中，Top-1错误(%)。与普通的ResNets相比，这里没有额外的参数。图4显示了训练过程。

我们认为这种优化困难不太可能是由于梯度消失造成的。这些普通网络用BN[16]进行训练，这保证了正向传播信号具有非零方差。我们还验证了反向传播梯度显示了BN的健康规范。所以前向和后向信号都不会消失。实际上，34层的素网仍然能够达到竞争精度(表3)，这表明求解器在一定程度上是可行的。我们推测深普通网可能具有指数级的低收敛率，这将影响训练误差的训练。这种优化困难的原因将在未来研究。

残差网络：接下来我们评估18层和34层残余网(ResNets)。基线架构是一样的上面的纯网,希望快捷连接添加到每一对3×3过滤器如图3(右)。在第一个比较中(表2和图4右侧)，我们对所有的快捷方式使用标识映射，对增加的维度使用零填充(选项A)，因此与普通的对应项相比，它们没有额外的参数。我们从表2和图4中得到了三个主要的观察结果。首先，随着残差学习的进行，情况发生了逆转——34层的ResNet比18层的ResNet好(高出2.8%)。更重要的是，34层的ResNet显示出相当低的训练错误，并可推广到验证数据。这表明降解问题在这个设置中得到了很好的解决，我们通过增加深度获得了精度增益。其次，与普通的ResNet相比，34层的ResNet将前1级的错误减少了3.5%(表2)，这是由于成功地减少了训练错误(图4左右)。这一比较验证了残差学习在极深系统中的有效性。

最后，我们还注意到18层的纯网格/残差网络的精度相当精确(表2)，但是18层的ResNet收敛速度更快(图4右与左)。当网络“不太深”(这里是18层)时，当前的SGD求解器仍然能够找到对纯网络的好的解决方案。在这种情况下，ResNet通过在早期提供更快的收敛来简化优化。

model	top-1 err.	top-5 err.
VGG-16 [40]	28.07	9.33
GoogLeNet [43]	-	9.15
PReLU-net [12]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

表3 ImageNet验证的错误率(%)。VGG-16是基于我们的测试。ResNet-50/101/152是选项B，它只使用投影来增加维度。

method	top-1 err.	top-5 err.
VGG [40] (ILSVRC'14)	-	8.43†
GoogLeNet [43] (ILSVRC'14)	-	7.89
VGG [40] (v5)	24.4	7.1
PReLU-net [12]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

表4 在ImageNet验证集上的单模型结果的错误率(%)(测试集上的报告除外)。

method	top-5 err. (**test**)
VGG [40] (ILSVRC'14)	7.32
GoogLeNet [43] (ILSVRC'14)	6.66
VGG [40] (v5)	6.8
PReLU-net [12]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

表5 合奏的错误率(%)。前5个错误出现在ImageNet的测试集中，并由测试服务器报告。

identify*VS*投影捷径：我们已经证明，没有参数的**identify**快捷方式有助于训练。接下来我们研究投影捷径(等式2)。在表3中我们比较了三个选项:(A)增加维度时使用了零填充快捷方式，所有的快捷方式都是无参数的(与表2和图4相同);(B)投影快捷键用于增加维度，其他快捷键为identify;(C)所有的捷径都是预测。

表3显示，这三个选项都比普通选项好得多。B略好于A，我们认为这是因为A中的零填充维度确实没有残差学习。C略好于B，我们将其归因于由许多(13)投影快捷方式引入的额外参数。但A/B/C之间的微小差异表明，投影捷径对于解决退化问题并不是必不可少的。因此，在本文的其余部分中，我们不使用选项C，以减少内存/时间复杂度和模型大小。identify快捷方式对于不增加下面介绍的瓶颈体系结构的复杂性特别重要。

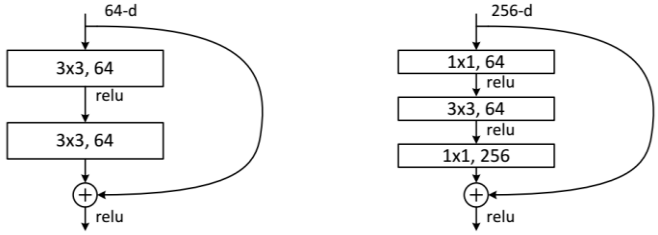

图5 ImageNet的一个较深的残馀函数F。左:一个构建块(56×56特征图)作为ResNet34在图3。右:ResNet-50/101/152的“瓶颈”构建块。

更深层次的瓶颈架构：接下来，我们将描述用于ImageNet的更深层次的网络。由于考虑到我们所能负担的训练时间，我们将构建块作为瓶颈设计进行了修改。对于每个残差函数F,我们用一堆3层代替2(图5)。三层1×1,3×3,1×1的隆起,其中1×1层负责降低,然后增加(恢复)维度,离开3×3层输入/输出尺寸较小的一个瓶颈。图5显示了一个示例，其中两个设计的时间复杂度相似。对于瓶颈体系结构，无参数identify快捷方式尤为重要。如果将图5(右)中的identify快捷方式替换为投影，则可以看出，由于该快捷方式连接到两个高维末端,时间复杂度和模型大小加倍。因此，identify快捷方式为瓶颈设计提供了更有效的模型。

50层ResNet:我们将34层net中的每个2层块替换为这个3层瓶颈块，得到50层ResNet(表1)。这种模式有38亿次浮点运算。

101-层和152-层ResNets:我们使用更多的3层block构建101-层和152-层ResNets(表1)。值得注意的是，虽然深度显著增加，152-层ResNet(113亿次浮点运算)的复杂度仍然低于VGG-16/19 nets(15.3/ 196亿FLOPs)。50/101/152层的ResNets比34层的要精确得多(表3和表4)。我们没有观察到降解问题，因此在相当大的深度上获得了显著的精度收益。所有评价指标都可以看到深度的好处(表3和表4)。

与最先进的方法进行比较：在表4中，我们将与以前最好的单模型结果进行比较。我们的基线34层resneet已经达到了非常有竞争力的精确度。我们的152层ResNet的单模型top-5验证错误为4.49%。这个单模型的结果优于之前所有的集成结果(表5)。我们将六个不同深度的模型组合成一个集成(提交时只有两个152层模型)。这导致测试集上的前5个错误3.57%(表5)。

4.2、CIFAR-10和分析

我们对CIFAR-10数据集[20]进行了更多的研究，该数据集由10个类的50000张训练图像和10000张测试图像组成。我们在训练集上进行实验，并在测试集上进行评估。我们关注的是极深网络的行为，而不是推动最先进的结果，所以我们故意使用简单的架构如下。普通/剩余体系结构遵循图3(中/右)的形式。网络输入32×32图片,单像素均值减去。第一层是3×3的隆起。然后我们使用一堆6 n层特性的地图尺寸3×3卷曲{ 8 } 32岁,16日分别为2 n层每个特性图的大小。过滤器的数量分别为{16,32,64}。子抽样通过卷积来执行，步长为2。网络以全球平均池、10路完全连接层和softmax结束。共有6n+2个叠加加权层。下表总结了架构:

output map size	32×32	16× 16	8×8
# layers	1+2n	2n	2n
# filters	16	32	64

使用快捷方式连接时,连接到双3×3层(完全3 n快捷方式)。在此数据集中，我们在所有情况下都使用标识快捷方式(即因此，我们的残差模型的深度、宽度和参数数量与普通模型完全相同。

method			error (%)
Maxout [9]			9.38
NIN [25]			8.81
DSN [24]			8.22
	# layers	# params	
FitNet [34]	19	2.5M	8.39
Highway [41, 42]	19	2.3M	7.54 (7.72±0.16)
Highway [41, 42]	32	1.25M	8.80
ResNet	20	0.27M	8.75
ResNet	32	0.46M	7.51
ResNet	44	0.66M	7.17
ResNet	56	0.85M	6.97
ResNet	110	1.7M	**6.43** (6.61±0.16)
ResNet	1202	19.4M	7.93

表6 CIFAR-10测试集的分类错误。所有方法都带有数据增强。resnet - 110,我们运行5次,显示 “最佳(平均±std)” [42]。

我们使用了0.0001的权值衰减和0.9的动量，在[12]和BN[16]中采用了权值初始化，但没有dropout。这些模型在两个gpu上以128的小批量进行训练。我们从学习率0.1开始，在32k和48k迭代时除以10，在64k迭代时终止训练，这是在45k/5k火车/val分割时确定的。我们遵循简单的数据增加[24]培训:4像素填充两边,一个32×32作物从填充图像或其随机抽样水平翻转。对于测试,我们只评估最初的32×32的单一视图的形象。

我们比较n ={3,5,7,9}，得到20,32,44和56层网络。图6(左)为素网的行为。深平原网深度增加，深度越深训练误差越大。这种现象与ImageNet(图4，左)和MNIST(见[41])类似，说明这种优化困难是一个基本问题。

图6(中间)为ResNets的行为。同样类似于ImageNet的例子(图4，右)，我们的resneet成功地克服了优化的困难，并且随着深度的增加显示了精度的提高。

我们进一步研究了n = 18得到110层的ResNet。在这种情况下，我们发现0.1的初始学习速率有点太大，无法开始收敛。因此，我们使用0.01对训练进行预热，直到训练误差低于80%(大约400次迭代)，然后返回0.1继续训练。其余的学习计划和之前一样。这个110层网络收敛良好(图6，中间)。与FitNet[34]和公路[41]等深细网络相比，其参数更少(表6)，但仍属于最先进的结果(6.43%，表6)。

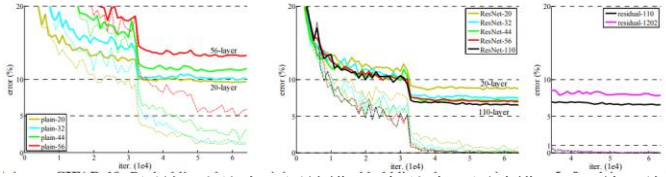

图6 CIFAR-10训练。虚线表示训练错误，粗线表示测试错误。左:纯网络。plain-110的误差大于60%，未显示。中间:ResNets。右:110层和1202层。

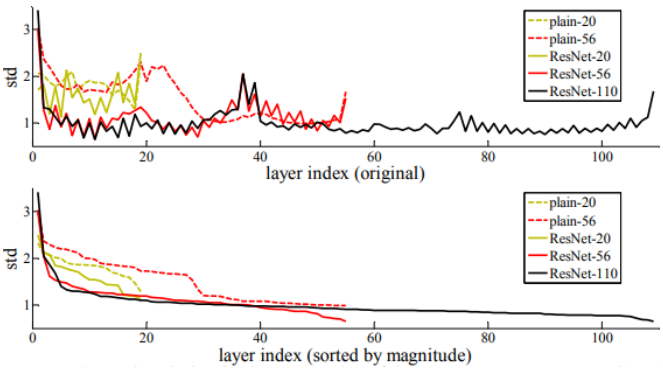

图7。CIFAR10层反应的标准偏差(std)。每个3×3的响应输出层,在BN和非线性。顶部:层按其原始顺序显示。底部:回答按照降序排列。层响应分析。

层响应分析：图7显示了层响应的标准差(std)。每个3×3的响应输出层,后BN和之前其他非线性(ReLU /之外)。对于ResNets，该分析揭示了残差函数的响应强度。从图中可以看出，ResNets通常比普通ResNets的响应小。这些结果支持了我们的基本动机(章节3.1)，残差函数可能比非残差函数更接近于零。我们还注意到，在图7中ResNet-20、56和110之间的比较表明，深度ResNet的响应幅度较小。当有更多层时，单个层的反射倾向于较少地修改信号。

探索超过1000层：我们探索了超过1000层的深度模型。我们将n = 200设为一个1202层的网络，如上所述进行训练。我们的方法没有优化难度，这个103层的网络可以实现训练误差<0.1%(图6，右)。它的测试错误仍然相当好(7.93%，表6)。

但在如此激进的深度模型上，仍存在一些有待解决的问题。这个1202层网络的测试结果比我们的110层网络的测试结果要差，虽然都有类似的训练误差。我们认为这是因为过度拟合。对于这个小数据集，1202层网络可能是不必要的大(194m)。采用maxout[9]或dropout[13]等强正则化方法在此数据集上获得最佳结果([9,25,24,34])。在本文中，我们没有使用maxout/dropout，只是通过设计简单地通过深而细的体系结构进行正则化，而没有转

移对优化难点的关注。但结合更强的正则化可能会改善结果，我们将在未来研究。

4.3、在PASCAL和MS COCO上进行目标检测

该方法对其他识别任务具有较好的泛化性能。表7和表8显示了PASCAL VOC 2007和2012[5]和COCO[26]的目标检测基线结果。我们采用更快的R-CNN[32]作为检测方法。在这里，我们对用ResNet-101替换VGG-16[40]的改进感兴趣。使用这两个模型的检测实现(见附录)是相同的，因此收益只能归功于更好的网络。最引人注目的是，在具有挑战性的COCO数据集上，我们获得了6%的COCO标准度量(mAP@[)增长。，这是28%的相对改善。这种收获完全是由于学习的表征。

基于深度残差网络，我们在2015年的ILSVRC和COCO竞赛中获得了包括ImageNet检测、ImageNet定位、COCO检测和COCO分割在内的多个项目的第一名。

参考

[1] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2):157–166, 1994.
[2] C. M. Bishop. Neural networks for pattern recognition. Oxford university press, 1995.
[3] W. L. Briggs, S. F. McCormick, et al. A Multigrid Tutorial. Siam, 2000.
[4] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In BMVC, 2011.
[5] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. IJCV, pages 303–338, 2010.
[6] R. Girshick. Fast R-CNN. In ICCV, 2015.
[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.
[8] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In AISTATS, 2010.
[9] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. arXiv:1302.4389, 2013.
[10] K. He and J. Sun. Convolutional neural networks at constrained time cost. In CVPR, 2015.
[11] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV, 2014.
[12] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In ICCV, 2015.
[13] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing coadaptation of feature detectors. arXiv:1207.0580, 2012.
[14] S. Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. Diploma thesis, TU Munich, 1991.
[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.
[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015.
[17] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. TPAMI, 33, 2011.
[18] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. TPAMI, 2012.
[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv:1408.5093, 2014.
[20] A. Krizhevsky. Learning multiple layers of features from tiny images. Tech Report, 2009.
[21] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.
[22] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural computation, 1989.
[23] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Muller. Efficient backprop. ¨In Neural Networks: Tricks of the Trade, pages 9–50. Springer, 1998.
[24] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeplysupervised nets. arXiv:1409.5185, 2014.
[25] M. Lin, Q. Chen, and S. Yan. Network in network. arXiv:1312.4400, 2013.
[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft COCO: Common objects in ´context. In ECCV. 2014.
[27] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015.
[28] G. Montufar, R. Pascanu, K. Cho, and Y. Bengio. On the number of ´ linear regions of deep neural networks. In NIPS, 2014.
[29] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In ICML, 2010.
[30] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In CVPR, 2007.
[31] T. Raiko, H. Valpola, and Y. LeCun. Deep learning made easier by linear transformations in perceptrons. In AISTATS, 2012.
[32] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.
[33] B. D. Ripley. Pattern recognition and neural networks. Cambridge university press, 1996.
[34] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In ICLR, 2015.
[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. arXiv:1409.0575, 2014.
[36] A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv:1312.6120, 2013.
[37] N. N. Schraudolph. Accelerated gradient descent by factor-centering decomposition. Technical report, 1998.
[38] N. N. Schraudolph. Centering neural network gradient factors. In Neural Networks: Tricks of the Trade, pages 207–226. Springer, 1998.
[39] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In ICLR, 2014.
[40] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.
[41] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. arXiv:1505.00387, 2015.
[42] R. K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. 1507.06228, 2015.
[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In CVPR, 2015.
[44] R. Szeliski. Fast surface interpolation using hierarchical basis functions. TPAMI, 1990.
[45] R. Szeliski. Locally adapted hierarchical basis preconditioning. In SIGGRAPH, 2006.
[46] T. Vatanen, T. Raiko, H. Valpola, and Y. LeCun. Pushing stochastic gradient towards second-order methods–backpropagation learning with transformations in nonlinearities. In Neural Information Processing, 2013.
[47] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.
[48] W. Venables and B. Ripley. Modern applied statistics with s-plus. 1999.
[49] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. In ECCV, 2014.