Densely Connected Convolutional Networks

Gao Huang*
Cornell University
gh349@cornell.edu

Zhuang Liu*
Tsinghua University
liuzhuang13@mails.tsinghua.edu.cn

Laurens van der Maaten
Facebook AI Research
lvdmaaten@fb.com

Kilian Q. Weinberger
Cornell University
kqw4@cornell.edu

Abstract

提示：该*PDF*由*SpeedPaper*生成，版权归原文作者所有。翻译内容仅供参考，请仔细鉴别并以原文为准。

查看更多论文翻译与复现代码：*https://github.com/hanknewbird/SpeedPaper*

最近的研究表明，如果卷积网络中包含介于输入层和输出层附近的层之间的短连接，那么这种网络可以更深、更准确且更易训练。在本文中，我们接受了这一观察结果，并引入了*Dense Convolutional Network*（*DenseNet*）。该方法以前馈方式将每一层连接到其他所有层。传统的具有L层的卷积网络有L个连接—每一层与其后续层之间有一个连接—而我们的网络有$\frac{L(L+1)}{2}$个直接连接。对于每一层，前面所有层的特征图被用作输入，而该层的特征图被用作所有后续层的输入。*DenseNets*具有几个引人注目的优点：它们缓解了梯度消失问题，增强了特征传播，鼓励了特征重复使用，并显著减少了参数数量。我们在四个极具竞争力的目标识别基准任务（*CIFAR-10*、*CIFAR-100*、*SVHN* 和 *ImageNet*）上评估了我们提出的架构。*DenseNets*在大部分任务中明显优于最先进技术，同时在实现高性能时需要较少的计算量。代码和预训练模型可在*https://github.com/liuzhuang13/DenseNet*上获得。

*Authors contributed equally

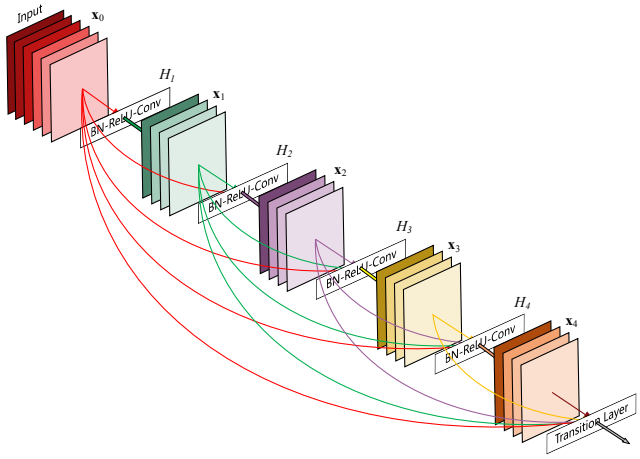

图 1: 一个dense block，共有5个层，增长速率为$k = 4$。每一层将所有前一层的特征图作为输入。

1. Introduction

卷积神经网络（CNNs）已成为视觉对象识别的主导机器学习方法。尽管它们最初是在20多年前引入的[18]，但计算机硬件和网络结构的改进最近才使得训练真正深层的CNNs成为可能。最初的LeNet5[19]由5层组成，VGG有19层[29]，仅在去年Highway Networks[34]和残差网络（ResNets）[11]才突破了100层的限制。

随着卷积神经网络的深度日益增加，一个新的研究问题浮现出来：随着关于输入或梯度的信息通过许多层传递，当它到达网络的末尾（或开始）时，可能

会消失并被“冲淡”。许多最近的出版物解决了这个或相关的问题。 ResNets[11] 和 Highway Networks[34] 通过恒等连接将信号从一层传递到下一层。随机深度[13]通过在训练过程中随机丢弃层来缩短 ResNets，以允许更好的信息和梯度流动。 FractalNets [17] 反复组合几个具有不同卷积块数量的并行层序列，以获得大量名义深度，同时在网络中保持许多短路径。虽然这些不同的方法在网络拓扑和训练程序上有所不同，但它们都有一个关键特征：它们在早期层和后期层之间创建了短路径。

在本文中，我们提出了一种体系结构，将这一见解融入简单的连接模式中：为了确保网络各层之间的最大信息流动，我们将*所有层*（具有匹配的特征图大小）直接连接起来。为了保持前馈性质，每一层从所有前面的层获取额外的输入，并将自己的特征图传递给所有后续的层。图1形象地说明了这种布局。至关重要的是，与 ResNets 不同，我们在将特征传递到层之前从不通过求和来合并特征；相反，我们通过连接它们来合并特征。因此，第 ℓ 层具有 ℓ 个输入，包括所有前面卷积块的特征图。它自己的特征图传递给所有 $L-\ell$ 个后续层。这在一个 L 层网络中引入了 $\frac{L(L+1)}{2}$ 个连接，而不是如传统架构中只有 L 个。由于其密集连接模式，我们将我们的方法称为*Dense Convolutional Network (DenseNet)* 。

这种密集连接模式可能具有令人费解的效果，即其所需的参数比传统卷积网络*更少*，因为不需要重新学习冗余的特征图。传统的前馈结构可以被视为具有状态的算法，该状态从层传递到层。每一层从其前面的层读取状态，并将其写入后续层。它改变状态但也传递需要被保留的信息。 ResNets[11] 通过加性恒等变换明确地表达了这种信息的保持。最近的 ResNets 变体[13] 表明，许多层贡献很少，实际上在训练过程中可以随机丢弃。这使得 ResNets 的状态类似于（展开的）递归神经网络[21]，但 ResNets 的参数数量要大得多，因为每个层都有自己的权重。我们提出的 DenseNet 架构明确区分了添加到网络中的信息和需要被保留的信息。 DenseNet 层非常窄（*例如*，每层 12 个过滤器），仅向网络的“集体知识”添加了一小组特征图并保持其余特征图不变——最终分类器基于网络中的所有特征图做出决策。

除了更好的参数效率外，DenseNet 的一个重要优势是其改进的信息和梯度在整个网络中的流动，使得网络易于训练。每一层直接接收来自损失函数和原始输入信号的梯度，导致隐式深度监督[20]。这有助于训练更深的网络架构。此外，我们还观察到密集连接具有正则化效果，可以减少对训练集大小较小任务的过拟合。

我们在四个竞争激烈的基准数据集（CIFAR-10、CIFAR-100、SVHN 和 ImageNet）上评估了 DenseNet。我们的模型往往比现有算法在参数数量上需要更少，同时具有相当的准确性。此外，在大多数基准任务中，我们明显优于当前的最先进结果。

2. Related Work

网络架构的探索自最初发现以来一直是神经网络研究的一部分。神经网络的近期复兴也使得这一研究领域重新活跃起来。现代网络中层数的增加加剧了架构之间的差异，并激励了对不同连接模式的探索以及对旧研究思想的重新审视。

类似于我们提出的密集网络布局的级联结构已经在上世纪80年代的神经网络文献中进行过研究 [3]。他们的开创性工作集中在全连接的逐层训练多层感知机上。最近，提出了用批量梯度下降训练的全连接级联网络 [40]。尽管在小数据集上有效，但这种方法只适用于参数数量较少的网络。在 [9, 23, 31, 41]中，通过跳跃连接在CNN中利用多级特征已被发现对各种视觉任务有效。与我们的工作平行，[1]提出了一个与我们的具有跨层连接相似的网络的纯理论框架。

高速网络 [34] 是最早提供有效训练超过100层端到端网络的架构之一。利用旁路路径和门控单元，高速网络可轻松优化具有数百层的网络。旁路路径被认为是简化这些非常深层网络训练的关键因素。这一点在 ResNets [11]中得到进一步支持，其中纯标识映射被用作旁路路径。ResNets在许多具有挑战性的图像识别、定位和检测任务上取得了令人瞩目的、创纪录的性能，如ImageNet和COCO物体检测 [11]。最近，提出了 *随机深度* 作为成功训练1202层ResNet的一种方法 [13]。随机深度通过在训练过程中随机丢弃层来改善深度残差网络的训练。这表明并非所有层都是必要的，并强调了深度（残差）网络中存在大量冗余的事实。我们的论文在一定程度上受到了这一观察的启发。具有 *预激活* 功能的ResNets也有助于训练具有 >1000 层的最先

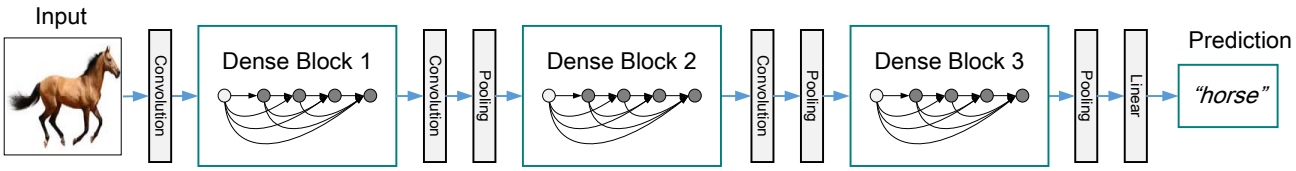

图 2: 一个具有三个密集块的深度DenseNet。两个相邻块之间的层被称为过渡层，并通过卷积和池化来改变特征图尺寸。

进网络 [12]。

使网络更深（例如，通过跳跃连接的帮助）的一种正交方法是增加网络的宽度。GoogLeNet [36, 37] 使用“Inception模块”，该模块串联由不同尺寸滤波器产生的特征图。在 [38] 中，提出了一种带有宽通用残差块的ResNets变体。事实上，简单地增加ResNets每层中的滤波器数量可以提高其性能，前提是深度足够 [42]。FractalNets 也利用宽网络结构在几个数据集上取得了竞争性结果 [17]。

与从极深或宽架构获取表征能力的方法不同，DenseNets 通过特征重用发挥网络的潜力，产生易于训练且高度参数高效的简化模型。连接由不同层学习的特征图可以增加输入的变化，提高效率。这构成了 DenseNets 与 ResNets 之间的主要区别。与也从不同层级串联特征的Inception网络 [36, 37] 相比，DenseNets 更简单更高效。

还有其他一些引人注目的网络架构创新取得了有竞争力的结果。网络中的网络（NIN）[22] 结构将微型多层感知器嵌入到卷积层的滤波器中，以提取更复杂的特征。在深度监督网络（DSN）[20]中，内部层直接由辅助分类器监督，这可以加强早期层接收到的梯度。梯度网络（Ladder Networks）[27, 25]将侧连接引入到自编码器中，在半监督学习任务上实现了令人印象深刻的准确性。在 [39] 中，提出了深度融合网络（DFNs）来通过组合不同基础网络的中间层来改善信息流。将网络增强通路以减少重构损失已被证明可以提高图像分类模型的性能 [43]。

3. DenseNets

考虑一个通过卷积网络传递的单个图像 $\mathbf{x}_0$。该网络包含 L 层，每一层实现非线性变换 $H_\ell(\cdot)$，其中 ℓ 表示层索引。$H_\ell(\cdot)$ 可以是批量归一化 (BN) [14]、修正线性单元 (ReLU) [6]、池化 (Pooling) [19] 或卷积 (Conv) 等操作的复合函数。我们将第 ℓ 层的输出表示为 $\mathbf{x}_\ell$。

ResNets. 传统的卷积前馈网络将第 ℓ 层的输出连接到第 $(\ell+1)$ 层的输入 [16]，这导致以下层转换：$\mathbf{x}_\ell = H_\ell(\mathbf{x}_{\ell-1})$。ResNets [11] 添加了一个跳跃连接，通过恒等函数绕过非线性变换：

$$\mathbf{x}_\ell = H_\ell(\mathbf{x}_{\ell-1}) + \mathbf{x}_{\ell-1}. \tag{1}$$

ResNets的一个优势是梯度可以直接通过恒等函数从后面的层流向前面的层。然而，恒等函数和$H_\ell$ff的输出是通过求和相结合的，这可能妨碍网络中的信息流动。

密集连接。 为了进一步改善层间的信息流动，我们提出了一种不同的连接模式：我们引入直接连接，将任意一层的特征传递到所有后续层。图 1 简要展示了结果 DenseNet 的布局。因此，第 ℓ 层接收所有前序层的特征图 $\mathbf{x}_0, \ldots, \mathbf{x}_{\ell-1}$ 作为输入：

$$\mathbf{x}_\ell = H_\ell([\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{\ell-1}]), \tag{2}$$

其中$[\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{\ell-1}]$ 表示在层 $0, \ldots, \ell-1$ 中产生的特征图的串联。由于其密集连接，我们将这种网络结构称为*Dense Convolutional Network (DenseNet)*。为了实现方便，我们将方程 (2) 中 $H_\ell(\cdot)$ 的多个输入连接成单个张量。

复合函数。 受[12]的启发，我们将$H_\ell(\cdot)$定义为三个连续操作的复合函数：批量归一化（BN）[14]，接着是修正线性单元（ReLU）[6]，最后是3×3卷积（Conv）。

汇聚层。 如公式 (2) 所用的连接操作，在特征图大小改变时并不可行。然而，卷积网络的一个关键组成部分是改变特征图大小的下采样层。为了在我们的架构中方便下采样，我们将网络分为多个密集连接的密集块；详见图 2。我们将块之间的层称为过渡层，它们执行卷积和汇聚操作。我们实验中使用的过渡层由批量归一化层、一个1×1卷积层和一个2×2平均汇聚层构成。

增长率。 如果每个函数H_ℓ产生k个特征图，那么第ℓ层将具有$k_0 + k \times (\ell - 1)$个输入特征图，这里$k_0$是

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

表 1: DenseNet在ImageNet上的架构。所有网络的增长率均为$k = 32$。请注意表中显示的每个“conv”层都对应序列BN-ReLU-Conv。

输入层中的通道数。DenseNet和现有网络架构之间的一个重要区别是DenseNet可以有非常窄的层，例如$k = 12$。我们将超参数k称为网络的*growth rate*。我们在第 4节中展示了一个相对较小的growth rate就足以在我们测试的数据集上获得最先进的结果。其中一个解释是每一层都可以访问其块中所有先前的特征图，因此可以访问网络的“集体知识”。可以将特征图视为网络的全局状态。每一层将自己的k个特征图添加到这个状态中。 growth rate 控制着每一层向全局状态中贡献多少新信息。全局状态一旦写入后，可以从网络中的任何地方访问，与传统网络架构不同的是，无需将其从一层复制到另一层。

瓶颈层。 尽管每一层仅产生k个输出特征图，但通常具有更多的输入。据[37, 11]指出，可以将1×1卷积引入作为*瓶颈层*，放置在每个3×3卷积之前，以减少输入特征图的数量，从而提高计算效率。我们发现这种设计对DenseNet特别有效，我们将具有这种瓶颈层的网络称为，即，DenseNet-B的H_ℓ版本，其中包含BN-ReLU-Conv(1×1)-BN-ReLU-Conv(3×3)。在我们的实验中，我们让每个1×1卷积产生4k个特征图。

压缩。 为了进一步改善模型的紧凑性，我们可以在过渡层减少特征图的数量。如果一个稠密块包含m个特征图，我们让接下来的过渡层生成$\lfloor \theta m \rfloor$个输出特征图，其中$0 < \theta \leq 1$被称为压缩因子。当$\theta = 1$时，过渡层之间的特征图数量保持不变。我们将$\theta < 1$的DenseNet称为 DenseNet-C，我们在实验中设置 $\theta = 0.5$。当瓶颈和过渡层都使用 $\theta < 1$ 时，我们将模型称为 DenseNet-BC。

实现细节。 在我们的实验中，除了ImageNet数据集外，DenseNet具有三个每个具有相同层数的密集块。在进入第一个密集块之前，在输入图像上执行一个输出通道为16（或DenseNet-BC的growth rate的两倍）的卷积。对于核大小为3×3的卷积层，输入的每一边都会通过一个像素进行零填充以保持特征图大小固定。我们在两个连续的密集块之间使用1×1卷积后跟2×2平均池化作为过渡层。在最后一个密集块结束时，执行全局平均池化，然后附加一个softmax分类器。三个密集块中的特征图大小分别为32× 32，16×16和8×8。我们尝试了基本的DenseNet结构，包括配置$\{L = 40, k = 12\}$，$\{L = 100, k = 12\}$和$\{L = 100, k = 24\}$。对于DenseNet-BC，评估了具有配置$\{L = 100, k = 12\}$，$\{L = 250, k = 24\}$和$\{L = 190, k = 40\}$的网络。

在ImageNet上的实验中，我们在224×224的输入图像上使用具有4个密集块的DenseNet-BC结构。初始卷

积层由尺寸为7×7、步幅为2的2k卷积组成；所有其他层中的特征图数目也由设置k决定。我们在ImageNet上使用的确切网络配置显示在Table 1中。

4. Experiments

我们在几个基准数据集上经验性地展示了DenseNet 的有效性，并将其与现有最先进的架构进行比较，特别是与 ResNet 及其变种。

4.1. Datasets

CIFAR. 两个 CIFAR 数据集[15] 包含了32×32像素的彩色自然图像。CIFAR-10（C10）包括来自10个类别的图像，CIFAR-100（C100）来自100个类别。训练集和测试集分别包含50,000和10,000张图像，我们将5,000张训练图像留出作为验证集。我们采用了一个在这两个数据集上广泛使用的标准数据增强方案（镜像/平移）[11, 13, 17, 22, 28, 20, 32, 34]。我们通过在数据集名称后面的“+”符号来表示这个数据增强方案（例如，C10+）。对于预处理，我们使用通道均值和标准偏差对数据进行归一化。在最终运行中，我们使用所有50,000张训练图像，并在训练结束时报告最终的测试错误。

SVHN. 街景门牌号码（SVHN）数据集[24]包含32×32像素的彩色数字图像。训练集中有73,257张图像，测试集中有26,032张图像，另有531,131张图像用于额外训练。按照常规做法 [7, 13, 20, 22, 30]，我们使用所有训练数据而不进行任何数据增强，并从训练集中拆分出6,000张图像作为验证集。我们选择在训练过程中具有最低验证错误的模型，并报告测试错误。我们遵循[42]，将像素值除以255，使它们位于$[0, 1]$范围内。

ImageNet. ILSVRC 2012分类数据集[2]包括120万张用于训练的图像和5万张用于验证的图像，涵盖了1000个类别。我们采用与[8, 11, 12]相同的训练图像数据增强方案，并在测试时采用单裁剪或10裁剪的方式，裁剪大小为224×224。根据[11, 12, 13]的做法，我们在验证集上报告分类错误率。

4.2. Training

所有的网络都是使用随机梯度下降（SGD）进行训练。在CIFAR和SVHN数据集上，我们分别使用批量大小为64进行训练，分别进行300和40个epochs。初始学习率设置为0.1，并在训练epochs的50%和75%时递减为原来的十分之一。在ImageNet数据集上，我们使用批量大小为256进行90个epochs的模型训练。学习率初始值为0.1，并分别在第30和60个epoch时减小十倍。需要注意的是，DenseNet的朴素实现可能存在内存效率问题。为了减少在GPU上的内存消耗，请参考我们关于内存高效实现DenseNets的技术报告[26]。

根据[8]，我们使用10^{-4}的权重衰减和Nesterov动量[35]为0.9（不减弱）。我们采用[10]介绍的权重初始化方法。对于三个没有数据增强的数据集，即C10、C100和SVHN，我们在每个卷积层之后（除第一个外）加入一个dropout层[33]，并将dropout率设置为0.2。测试误差仅针对每个任务和模型设置评估一次。

4.3. Classification Results on CIFAR and SVHN

我们训练具有不同深度L和步长k的DenseNet。CIFAR和SVHN上的主要结果如表 2所示。为了突出一般趋势，我们将所有优于现有最先进技术的结果用**粗体**标记，并将整体最佳结果标记为蓝色。

准确性。 可能最明显的趋势可能源自表 2 的底部一行，其中显示在所有 CIFAR 数据集上，具有 $L = 190$ 和 $k = 40$ 的 DenseNet-BC 始终优于现有的最先进技术。其在 C10+ 上的错误率为 3.46%，在 C100+ 上为 17.18%，远低于 wide ResNet 结构 [42] 的错误率。我们在 C10 和 C100 上的最佳结果（不包括数据增强）更令人鼓舞：两者都比采用 drop-path 正则化的 FractalNet [17] 低近 30%。在 SVHN 数据集上，带有 dropout 的 DenseNet 使用 $L = 100$ 和 $k = 24$ 也超过了 wide ResNet 实现的当前最佳结果。然而，250 层的 DenseNet-BC 并未进一步提高其较短对应模型的性能。这可能是因为 SVHN 是一个相对较简单的任务，而极深的模型可能过度拟合训练集。

容量。 没有压缩或瓶颈层的情况下，一般趋势是随着L和k的增加，DenseNets的性能会更好。我们主要归因于模型容量的相应增长。最好的例证是C10+和C100+的列。在C10+上，随着参数数量从1.0M增加到7.0M，最后到27.2M，错误率从5.24%下降到4.10%，最终降至3.74%。在C100+上，我们观察到类似的趋势。这表明DenseNets能够利用更大更深模型

Method	Depth	Params	C10	C10+	C100	C100+	SVHN
Network in Network [22]	-	-	10.41	8.81	35.68	-	2.35
All-CNN [32]	-	-	9.08	7.25	-	33.71	-
Deeply Supervised Net [20]	-	-	9.69	7.97	-	34.57	1.92
Highway Network [34]	-	-	-	7.72	-	32.39	-
FractalNet [17]	21	38.6M	10.18	5.22	35.34	23.30	2.01
with Dropout/Drop-path	21	38.6M	7.33	4.60	28.20	23.73	1.87
ResNet [11]	110	1.7M	-	6.61	-	-	-
ResNet (reported by [13])	110	1.7M	13.63	6.41	44.74	27.22	2.01
ResNet with Stochastic Depth [13]	110	1.7M	11.66	5.23	37.80	24.58	1.75
	1202	10.2M	-	4.91	-	-	-
Wide ResNet [42]	16	11.0M	-	4.81	-	22.07	-
	28	36.5M	-	4.17	-	20.50	-
with Dropout	16	2.7M	-	-	-	-	1.64
ResNet (pre-activation) [12]	164	1.7M	11.26*	5.46	35.58*	24.33	-
	1001	10.2M	10.56*	4.62	33.47*	22.71	-
DenseNet ($k = 12$)	40	1.0M	**7.00**	5.24	**27.55**	24.42	1.79
DenseNet ($k = 12$)	100	7.0M	**5.77**	**4.10**	**23.79**	**20.20**	1.67
DenseNet ($k = 24$)	100	27.2M	**5.83**	**3.74**	**23.42**	**19.25**	**1.59**
DenseNet-BC ($k = 12$)	100	0.8M	**5.92**	4.51	**24.15**	22.27	1.76
DenseNet-BC ($k = 24$)	250	15.3M	**5.19**	**3.62**	**19.64**	**17.60**	1.74
DenseNet-BC ($k = 40$)	190	25.6M	-	**3.46**	-	**17.18**	-

表 2: CIFAR和SVHN数据集上的错误率（%）。k表示网络的growth rate。超过所有竞争方法的结果为**粗体**，所有最佳结果为蓝色。"+"表示标准数据增强（平移和/或镜像）。*表示我们自己运行的结果。所有DenseNets没有数据增强（C10、C100、SVHN）均使用Dropout获得。与ResNet相比，DenseNets在使用更少参数的情况下实现更低的错误率。没有数据增强时，DenseNet明显表现更好。

Model	top-1	top-5
DenseNet-121	25.02 / 23.61	7.71 / 6.66
DenseNet-169	23.80 / 22.08	6.85 / 5.92
DenseNet-201	22.58 / 21.46	6.34 / 5.54
DenseNet-264	22.15 / 20.80	6.12 / 5.29

表 3: 在ImageNet验证集上，使用单个裁剪/10个裁剪测试的top-1和top-5错误率。

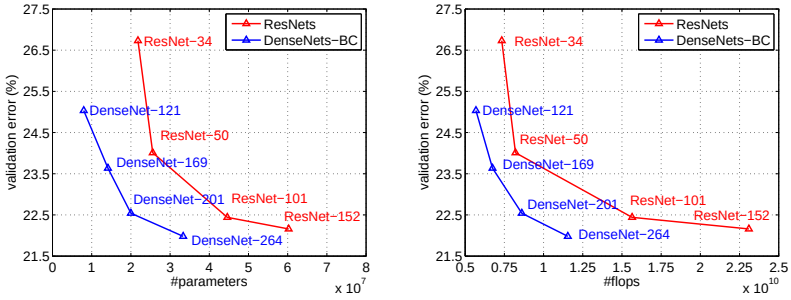

图 3: Comparison of the DenseNets and ResNets top-1 error rates (single-crop testing) on the ImageNet validation dataset as a function of learned parameters (*left*) and FLOPs during test-time (*right*).

的增强表征能力。这也表明它们不会受到过拟合或残差网络优化困难的影响 [11]。

参数效率。 表 2 中的结果表明，相较于其他架构（特别是 ResNets），DenseNets 更高效地利用参数。具有瓶颈结构和过渡层降维的 DenseNet-BC 尤其参数高效。

例如，我们的250层模型仅有15.3M参数，但始终优于其他拥有超过30M参数的模型，如FractalNet和Wide ResNets。我们还强调，DenseNet-BC，其中$L=100$和$k=12$，在C10+上达到了相当的性能（例如，误差为4.51% vs 4.62%，在C100+上的误差为22.27% vs 22.71%），而仅使用90%更少的参数，与拥有1001层的预激活ResNet性能相当。图4（右侧面板）展示了这两个网络在C10+上的训练损失和测试错误。1001层深度ResNet收敛到较低的训练损失值，但类似的测试错误。我们将在下面更详细地分析这种效应。

过拟合。 参数更有效地利用的一个正面效果是DenseNets更不容易过拟合。我们观察到，在没有数据增强的数据集上，DenseNet架构相比之前的工作表现出的改进尤为显著。在C10上，改进表现为错误率从7.33%降低到5.19%，相对减少了29%。在C100上，这一减少约为30%，从28.20%降至19.64%。在我们的实验中，我们观察到在某个设置中潜在的过拟合现象：在C10上，通过将k从12增加到24使参数增加4倍，错误率从5.77%升至5.83%。DenseNet-BC的瓶颈和压缩层似乎是对抗这一趋势的有效方式。

4.4. Classification Results on ImageNet

我们在ImageNet分类任务上评估了不同深度和增长率的DenseNet-BC，并将其与最先进的ResNet架构进行比较。为了确保两种架构之间的公平比较，我们通过采用[8]提供的ResNet的Torch实现[1]，消除了数据预处理和优化设置等其他因素的差异。我们简单地将ResNet模型替换为DenseNet-BC网络，并将所有实验设置完全保持与ResNet相同。

我们在表3中报告了DenseNets在ImageNet上的单裁剪和10裁剪验证错误。图3显示了作为参数数量（左）和FLOPs（右）函数的DenseNets和ResNets的单裁剪top-1验证错误。图中呈现的结果显示，DenseNets的表现与最先进的ResNets相当，同时需要较少的参数和计算来实现可比较的性能。例如，一个具有20M参数的DenseNet-201模型产生了与具有超过40M参数的101层ResNet类似的验证错误。右图中也可以观察到类似的趋势，它将验证错误作为FLOPs数量的函数绘制出来：一个需要与ResNet-50一样的计算量的DenseNet与一个需要两倍计算量的ResNet-101表现相当。

值得注意的是，我们的实验设置意味着我们使用了针对ResNets而不是DenseNets进行优化的超参数设置。可以想象，更广泛的超参数搜索可能会进一步改善DenseNet在ImageNet上的性能。

5. Discussion

从表面上看，DenseNet在某种程度上与ResNets相似：方程(2)与方程(1)的区别仅在于$H_\ell(\cdot)$的输入是连接起来而不是相加。然而，这个看似小的修改所导致的影响会使这两种网络架构的行为显著不同。

模型紧凑性。 由于输入连接的直接结果，任何DenseNet层学习的特征图都可以被所有后续层访问。这促进了整个网络中的特征重用，从而导致更加紧凑的模型。

在图4中左侧的两个图表显示了一个旨在比较所有DenseNets变体的参数效率（左）以及一种可比较的ResNet架构（中）的实验结果。我们在C10+数据集上训练了多个深度不同的小型网络，并绘制了它们的测试准确率与网络参数的关系。与其他流行的网络架构如AlexNet[16]或VGG-net[29]相比，使用预激活的ResNets需要更少的参数，通常可以获得更好的结果[12]。因此，我们将DenseNet（$k=12$）与该架构进行比较。对于DenseNet的训练设置与上一节相同。

图表显示DenseNet-BC一直是DenseNet中参数效率最高的变体。另外，为了获得相同水平的准确率，DenseNet-BC只需要ResNets的约1/3的参数（中间图）。这个结果与我们在图3中展示的ImageNet数据集结果一致。图4中右侧的图表显示，具有仅0.8M可训练参数的DenseNet-BC能够获得与具有10.2M参数的1001层（预激活）ResNet[12]相当的准确率。

隐式深度监督。 dense convolutional network的提高准确性的一个解释可能是各个层通过更短的连接从损失函数获得了额外的监督。可以解释DenseNets执行一种“深度监督”。之前已经在深度监督网络（DSN; [20]）中展示了深度监督的好处，其中每个隐藏层都附有分类器，强制中间层学习具有区分性的特征。

DenseNets通过一种隐式方式执行类似的深度监督：网络顶部的单个分类器通过最多两到三个转换层

[1] https://github.com/facebook/fb.resnet.torch

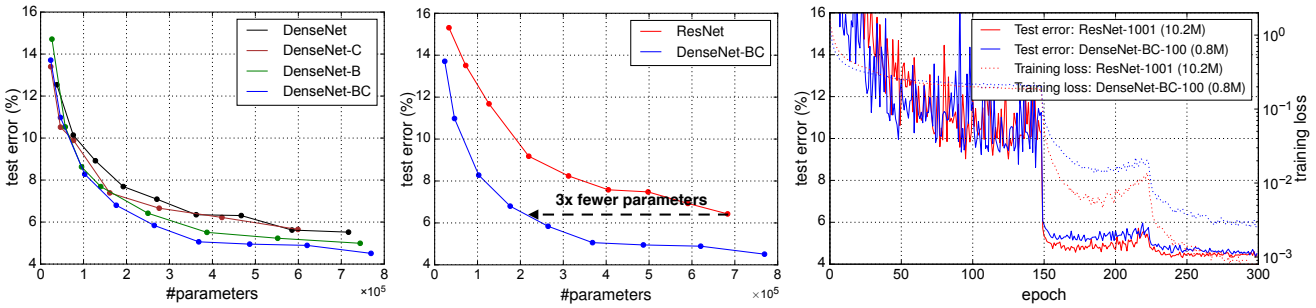

图 4: *左：* 对比DenseNet 变体之间在 C10+ 上的参数效率。 *中：* 对比 DenseNet-BC 与（预激活）ResNets 之间的参数效率。DenseNet-BC 达到可比较的准确性所需的参数约为 ResNet 的三分之一。 *右：* 具有超过 10M 参数的 1001 层预激活 ResNet [12] 以及仅有 0.8M 参数的 100 层 DenseNet 的训练和测试曲线。

直接监督所有层。然而，DenseNets 的损失函数和梯度要简单得多，因为所有层之间共享相同的损失函数。

随机 vs. 确定性连接。 dense convolutional network 和残差网络的随机深度正则化 [13] 之间存在有趣的联系。在随机深度中，残差网络中的层被随机丢弃，从而在周围层之间创建直接连接。由于池化层从不被丢弃，网络的连接模式与 DenseNet 类似：在相同的池化层之间，任意两个层之间存在着直接连接的小概率—如果所有中间层都被随机丢弃。尽管这些方法在根本上相当不同，DenseNet 对随机深度的解释可能有助于理解这种正则化方法的成功。

特征重用。 按设计，DenseNets 允许层访问其所有前序层的特征图（尽管有时是通过过渡层）。我们进行了实验，以调查经过训练的网络是否利用了这一机会。首先，在 C10+ 上训练一个 $L=40$，$k=12$ 的 DenseNet。对于每个块内的卷积层 ℓ，我们计算其与第 s 层的连接的平均（绝对）权重。图 5 显示了三个稠密块的热度图。平均绝对权重作为卷积层对其前序层依赖性的替代指标。图中的红点 (ℓ, s) 指示层 ℓ 平均而言强烈利用产生自第 s 层的特征图。从图中可以得出几点观察结果：

1. 所有层在同一块中将它们的权重分布在许多输入上。这表明，确实，由最早的层提取的特征直接被整个相同的密集块中的深层使用。
2. 转换层的权重还将它们的权重分布到之前稠密块中的所有层中，表明信息通过少量间接传递从DenseNet的第一层流向最后一层。
3. 在第二和第三密集块中，层始终对过渡层的输出分配最低的权重（三角形的顶部行），这表明过渡层输出许多冗余特征（平均权重较低）。这与DenseNet-BC的显著结果保持一致，其中恰好对这些输出进行了压缩。
4. 尽管最终的分类层位于最右侧，也跨越整个密集块使用权重，但似乎集中于最终特征图，这表明网络中可能产生了一些更高级别的特征。

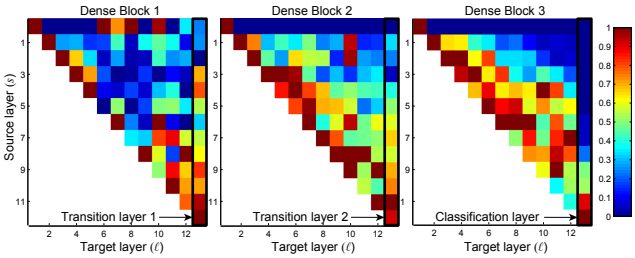

图 5: 在经过训练的DenseNet中卷积层的平均绝对过滤器权重。像素(s,ℓ)的颜色编码了连接密集块内卷积层s到ℓ的权重的平均$L1$范数（通过输入特征图数量归一化）。黑色矩形突出显示的三列对应于两个过渡层和分类层。第一行编码了连接到密集块输入层的权重。

6. Conclusion

我们提出了一种新的卷积网络架构，我们称之为Dense Convolutional Network（DenseNet）。它引入了具有相同特征图大小的任意两层之间的直接连接。我们展示了DenseNets自然地扩展到数百层，同时没有展现出任何优化困难。在我们的实验中，DenseNets倾向于随着参数数量的增长在准确性上持续改进，而不显示出性能退化或过拟合的迹象。在多个设置下，它在几个高度竞争的数据集上实现了最先进的结果。此外，DenseNets需要更少的参数和更少的计算来实现最先进

的性能。因为我们在研究中采用了针对残差网络优化的超参数设置，我们相信通过更详细地调整超参数和学习速率表，DenseNets的准确性可能会进一步提高。

虽然遵循简单的连接规则，DenseNets自然地整合了恒等映射、深度监督和多样深度的特性。它们允许网络中的特征重用，并因此可以学习更紧凑的、根据我们的实验更准确的模型。由于它们紧凑的内部表示和减少的特征冗余，DenseNets可能是各种基于卷积特征构建的计算机视觉任务的良好特征提取器，例如[4, 5]。我们计划在未来的工作中使用DenseNets研究这种特征转移。

致谢。 The authors are supported in part by the NSF III-1618134, III-1526012, IIS-1149882, the Office of Naval Research Grant N00014-17-1-2175 and the Bill and Melinda Gates foundation. GH is supported by the International Postdoctoral Exchange Fellowship Program of China Postdoctoral Council (No.20150015). ZL is supported by the National Basic Research Program of China Grants 2011CBA00300, 2011CBA00301, the NSFC 61361136003. We also thank Daniel Sedra, Geoff Pleiss and Yu Sun for many insightful discussions.

参考文献

[1] C. Cortes, X. Gonzalvo, V. Kuznetsov, M. Mohri, and S. Yang. Adanet: Adaptive structural learning of artificial neural networks. *arXiv preprint arXiv:1607.01097*, 2016. 2

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[3] S. E. Fahlman and C. Lebiere. The cascade-correlation learning architecture. In *NIPS*, 1989. 2

[4] J. R. Gardner, M. J. Kusner, Y. Li, P. Upchurch, K. Q. Weinberger, and J. E. Hopcroft. Deep manifold traversal: Changing labels with convolutional features. *arXiv preprint arXiv:1511.06421*, 2015. 9

[5] L. Gatys, A. Ecker, and M. Bethge. A neural algorithm of artistic style. *Nature Communications*, 2015. 9

[6] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *AISTATS*, 2011. 3

[7] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. In *ICML*, 2013. 5

[8] S. Gross and M. Wilber. Training and investigating residual nets, 2016. 5, 7

[9] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 2

[10] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 5

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 3, 4, 5, 6

[12] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, 2016. 3, 5, 6, 7, 8

[13] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016. 2, 5, 6, 8

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 3

[15] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Tech Report*, 2009. 5

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 3, 7

[17] G. Larsson, M. Maire, and G. Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648*, 2016. 2, 3, 5, 6

[18] 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*, 1(4):541–551, 1989. 1

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1, 3

[20] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *AISTATS*, 2015. 2, 3, 5, 6, 7

[21] Q. Liao and T. Poggio. Bridging the gaps between residual learning, recurrent neural networks and visual cortex. *arXiv preprint arXiv:1604.03640*, 2016. 2

[22] M. Lin, Q. Chen, and S. Yan. Network in network. In *ICLR*, 2014. 3, 5, 6

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

[24] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning, 2011. In *NIPS Workshop*, 2011. 5

[25] M. Pezeshki, L. Fan, P. Brakel, A. Courville, and Y. Bengio. Deconstructing the ladder network architecture. In *ICML*, 2016. 3

[26] G. Pleiss, D. Chen, G. Huang, T. Li, L. van der Maaten, and K. Q. Weinberger. Memory-efficient implementation of densenets. *arXiv preprint arXiv:1707.06990*, 2017. 5

[27] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-supervised learning with ladder networks. In *NIPS*, 2015. 3

[28] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015. 5

[29] 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. *IJCV*. 1, 7

[30] P. Sermanet, S. Chintala, and Y. LeCun. Convolutional neural networks applied to house numbers digit classification. In *ICPR*, pages 3288–3291. IEEE, 2012. 5

[31] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In *CVPR*, 2013. 2

[32] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 5, 6

[33] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 2014. 5

[34] R. K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. In *NIPS*, 2015. 1, 2, 5, 6

[35] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013. 5

[36] 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. 3

[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 3, 4

[38] S. Targ, D. Almeida, and K. Lyman. Resnet in resnet: Generalizing residual architectures. *arXiv preprint arXiv:1603.08029*, 2016. 3

[39] J. Wang, Z. Wei, T. Zhang, and W. Zeng. Deeply-fused nets. *arXiv preprint arXiv:1605.07716*, 2016. 3

[40] B. M. Wilamowski and H. Yu. Neural network learning without backpropagation. *IEEE Transactions on Neural Networks*, 21(11):1793–1803, 2010. 2

[41] S. Yang and D. Ramanan. Multi-scale recognition with dag-cnns. In *ICCV*, 2015. 2

[42] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 3, 5, 6

[43] Y. Zhang, K. Lee, and H. Lee. Augmenting supervised neural networks with unsupervised objectives for large-scale image classification. In *ICML*, 2016. 3