arXiv:1608.06993v5 [cs.CV] 28 Jan 2018

密集卷积网络

Gao Huang[1]
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

摘要

最近的工作表明，如果卷积网络包含靠近输入和靠近输出的层之间的较短连接，那么卷积网络可以更深入、更准确和更有效地进行训练。在本文中，我们接受这一观察，并介绍了密集卷积网络(DenseNet)，它以前馈的方式将每一层连接到每一层。而传统的具有 L 层的卷积网络具有 L 个连接,每层与其后续层之间有 1 个连接,我们的网络有$\frac{L(L+1)}{2}$个直接连接。对于每一层，所有前一层的特征图作为输入，其自身的特征图作为输入到所有后续层。DenseNets 有几个引人注目的优点：它们缓解了消失梯度问题，加强了特征传播，鼓励了特征重用，并且大大减少了参数的数量。我们在 4 个高度竞争的物体识别基准任务(CIFAR - 10、CIFAR - 100、SVHN 和 ImageNet)上评估了我们提出的架构。与最新技术相比，DenseNet 在大多数情况下获得了显著的改进，同时需要更少的计算量来实现高性能。代码和预训练模型可在 https://github.com/liuzhuang13/DenseNet 获得。

1. 引言

卷积神经网络(CNNs)已成为视觉目标识别的主流机器学习方法。虽然它们最初是在 20 多最初的 Le Net5 [19]有 5 层，VGG 有 19 层[29]，

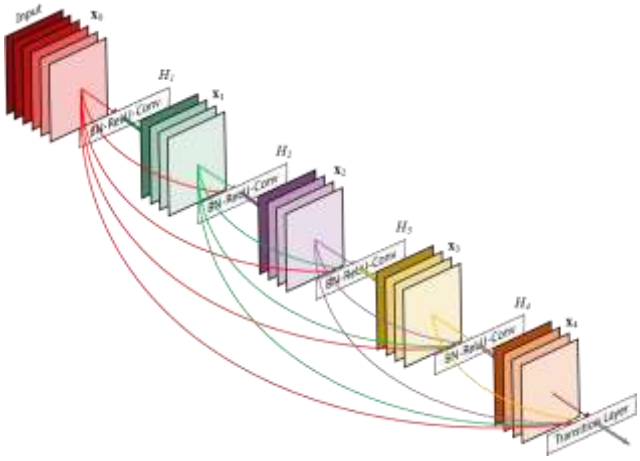

Figure 1: A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input.

只有去年的 Highway Networks 和 Residual Networks (深度残差网络) 超过了 100 层。

随着卷积神经网络的日益深入，一个新的研究问题出现了：当输入或梯度的信息经过许多层时，当它到达网络的末端(或开始)时，它就会消失和"洗出"。最近的许多出版物解决了这个或相关的问题。深度残差网络和 Highway Networks 通过身份连接将信号从一层绕过到下一层。随机深度通过在训练过程中随机丢弃层来缩短深度残差网络，以允许

[1] Authors contributed equally

更好的信息和梯度流。FractalNets 通过将多个并行层序列与不同数量的卷积块重复组合，在获得较大名义深度的同时，保持网络中较多的短路径。尽管这些不同的方法在网络拓扑结构和训练过程上有所不同，但它们都有一个关键的特征：它们从早期层到后期层都创建了短路径。

在本文中，我们提出了一种架构，将这种洞察力提炼为一种简单的连接模式：为了确保网络中各层之间的最大信息流，我们将所有层(与特征图尺寸匹配)直接相互连接。为了保持前馈特性，每一层从所有前一层获得额外的输入，并将自己的特征映射传递给所有后面的层。图 1 为该布局示意图。关键的是，与深度残差网络不同的是，在将特征传递到一个层之前，我们从来不会通过求和来组合特征；取而代之的是，我们通过将特征进行串联来进行组合。因此，第 e^{th} 层有 e 个输入，由前面所有卷积块的特征图组成。其自身的特征映射传递给所有 L - e 的后续层。这在一个 L 层网络中引入了$\frac{L(L+1)}{2}$个连接，而不是像传统架构那样只引入 L。由于其密集的连接模式，我们将我们的方法称为密集卷积网络(Dense Convolutional Network，DenseNet)。

这种密集连接模式的一个可能与直觉相反的效果是，它比传统的卷积网络需要更少的参数，因为不需要重新学习冗余的特征图。传统的前馈架构可以看作是一种状态的算法，这种状态是逐层传递的。每一层从其前一层读取状态并写入到后续层。它在改变状态的同时也传递着需要保存的信息。深度残差网络[11]通过加性恒等变换将这种信息保持显性化。深度残差网络[13]的最新变种表明，许多层的贡献很小，实际上可以在训练过程中随机丢弃。这使得深度残差网络的状态类似于(未滚动的)递归神经网络[21]，但由于每一层都有自己的权重，所以深度残差网络的参数数量大幅增加。 我们提出的 DenseNet 体系结构明确区分了添加到网络中的信息和保留的信息。DenseNet 层是非常窄的(例如,每层 12 个滤波器)，只向网络的"集体知识"添加一小部分特征图，并保持剩余的特征图不变，最终的分类器根据网络中的所有特征图做出决策。

除了更好的参数效率之外，DenseNets 的一大优势是它们在整个网络中改进了信息和梯度的流动，这使得它们易于训练。每一层都直接访问来自损失函数和原始输入信号的梯度，从而形成一个隐式的深度监督[20]。这有助于训练更深层次的网络架构。此外，我们还观察到密集连接具有正则化效应，从而减少了对具有较小训练集大小的任务的过拟合。

我们在四个极具竞争力的基准数据集(CIFAR - 10、CIFAR - 100、SVHN 和 ImageNet)上评估 DenseNets。我们的模型往往比现有的具有相当精度的算法需要更少的参数。此外，我们在大多数基准任务上都显著优于当前最先进的结果。

2. 相关工作

对网络体系结构的探索自最初发现以来一直是神经网络研究的一部分。最近神经网络的重新流行也使这一研究领域复兴。现代网络中越来越多的层放大了体系结构之间的差异，并激发了对不同连接模式的探索和对旧研究思路的重新审视。

类似于我们提出的密集网络布局的级联结构在 20 世纪 80 年代的神经网络文献中已经被研究过[3]。他们的开创性工作集中在以逐层方式训练的全连接多层感知器上。最近，采用批量梯度下降法训练的全连接级联网络被提出[40]。虽然这种方法在小数据集上有效，但它只能扩展到具有几百个参数的网络。在[9,23,31,41]中，通过跳跃连接使用 CNN 中的多级特征已被发现对各种视觉任务是有效的。与我们的工作平行，[1]推导出了与我们类似的具有跨层连接的网络的纯理论框架。

Highway Networks [34]是第一批提供有效训练 100 层以上端到端网络的架构之一。

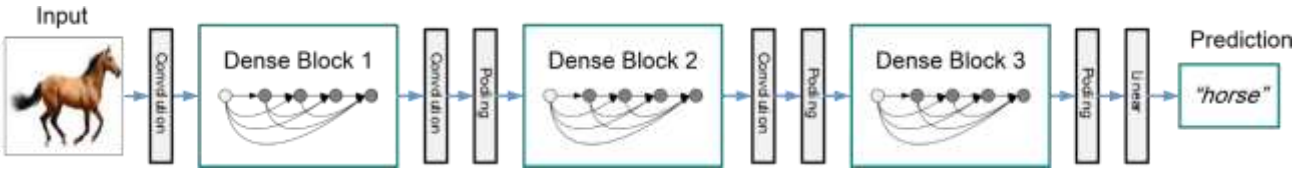

图 2：一个具有三个密集块的深度 DenseNet。两个相邻块之间的层被称为过渡层，并通过卷积和池化来改变特征图的大小。

使用绕行路径和门控单元，可以毫无困难地优化具有数百层的公路网。旁路路径被认为是缓解这些非常深的网络的训练的关键因素。这一点得到了深度残差网络[11]的进一步支持，深度残差网络[11]使用纯身份映射作为绕过路径。深度残差网络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印象深刻的、破纪录的性能，如 ImageNet 和 COCO 目标检测[11]。最近，随机深度作为一种成功训练 1202 层 Res Net 的方法被提出[13]。随机深度通过在训练过程中随机丢弃层来改进深度残差网络的训练。这表明并不是所有的层都需要，并强调了深度(残差)网络中存在大量冗余。我们的论文部分地受到了这种观察的启发。具有预激活功能的深度残差网络还有助于训练具有> 1000 层的最先进网络[12]。

一种使网络更深(例如,在跳过连接的帮助下)的正交方法是增加网络宽度。谷歌公司使用了一个" Inception 模块"，将不同大小的过滤器产生的特征图串联起来。在[38]中，提出了具有宽泛广义剩余块的深度残差网络的变体。事实上，在深度足够的情况下，简单地增加深度残差网络每层的滤波器数量可以提高其性能[42]。FractalNets 还利用宽广的网络结构在多个数据集上取得了有竞争力的结果。

DenseNets 没有从极深或极宽的体系结构中提取表示能力，而是通过特征重用来挖掘网络的潜力，产生易于训练和参数高效的浓缩模型。不同层学习到的特征图串联，增加了后续层输入的变化，提高了效率。这构成了 DenseNets 和深度残差网络之间的主要区别。与 Inception 网络[36、37] (也连接不同层的特性)相比，DenseNets 更简单、更高效。

还有其他值得注意的网络体系结构创新，取得了有竞争力的结果。网络中的网络(Network in Network，NIN) [22]结构将微型多层感知器包含在卷积层的滤波器中，以提取更复杂的特征。在深度监督网络(Deeply Supervised Network，DSN) [20]中，内部层由辅助分类器直接监督，可以加强早期层接收到的梯度。阶梯网络[27、25]在自编码器中引入了横向连接，在半监督学习任务上产生了令人印象深刻的精度。在[39]中，深度融合网络(Deeply-Fused Nets，DFNs)被提出，通过组合不同基础网络的中间层来改善信息流。 使用最小化重建损失的通路增强网络也被证明可以改善图像分类模型[43]。

3. DenseNets

考虑经过卷积网络的单幅图像 x_0。该网络由 L 层组成，每层实现一个非线性变换 $H_e(\cdot)$，其中 e 为该层的索引。$H_e(\cdot)$可以是批归一化(Batch Normalization，BN) [14]、整流线性单元(ReLU) [6]、池化(Pooling) [19]或卷积(Convolution，Conv)等操作的复合函数。我们把第 e 层的输出表示为 x_e。

深度残差网络：传统的卷积前馈网络将第 e 层的输出作为输入连接到 e+1 层，这就产生了如下的层转换：

$$X_e = H_e(x_{e-1}) + x_{e-1} \qquad (1)$$

深度残差网络的一个优点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_e 的输出是通过求和的方

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	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv; 3 × 3 conv] × 24	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48	[1 × 1 conv; 3 × 3 conv] × 64
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv; 3 × 3 conv] × 16	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

表 1：ImageNet 的 DenseNet 结构。所有网络的增长率均为 k = 32。值得注意的是，表中显示的每个' conv '层对应的是序列 BN - ReLU - Conv。

式结合在一起的，这可能会阻碍网络中的信息流动。

稠密连接：为了进一步改善层与层之间的信息流动，我们提出了一种不同的连接模式：我们引入了从任意层到所有后续层的直接连接。图 1 以示意图的方式说明了生成的 DenseNet 的布局。因此，第 e 层接收所有前一层的特征映射，$x_0 \sim x_{e-1}$，作为输入：

$$\mathbf{x}` = H`([\mathbf{x}_0,\mathbf{x}_1,...,\mathbf{x}_{e-1}]), (2)$$

其中[x0 , x1 , ... , x_{e-1}]是指在 0，..，e- 1 层产生的特征映射的级联。由于其密集的连通性，我们将这种网络架构称为密集卷积网络(Dense Convolutional Network，DenseNet)。为了便于实施，我们将公式（2）中 $H_e(\cdot)$的多个输入合并成一个单一的张量。

复合功能：受文献[12]的启发，我们将 H_e (•)定义为三个连续操作的复合函数：批归一化(BN) [14]，然后是校正线性单元(ReLU) [6]和 3 × 3 卷积(Conv)。

池化层：公式中使用的级联操作。(2)当特征图的大小发生变化时不可行。然而，卷积网络的一个重要部分是改变特征图大小的下采样层。为了便于在我们的架构中进行下采样，我们将网络划分为多个密集连接的密集块；见图 2。我们将块之间的层称为过渡层，过渡层做卷积和池化。实验中使用的过渡层由一个批归一化层和一个 1 × 1 的卷积层以及一个 2 × 2 的平均池化层组成。

增长率：如果每个函数 H 产生 k 个特征图，那么第层就有 k0+k×(`−1)个输入特征图，其中 k0 是输入层的通道数。DenseNet 与现有网络架构的一个重要区别在于，DenseNet 可以拥有非常窄的层，例如，k=12。我们将超参数 k 称为网络的增长率。我们在第 4 节中展示，对于我们测试的数据集，一个相对较小的增长率足以获得最先进的结果。对此的一种解释是，每个层都可以访问其块中所有前面的特征图，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。每个层向该状态添加 k 个自己的特征图。增长率调节每个层对全局状态贡献的新信息的数量。一旦写入全局状态，就可以从网络的任何位置访问它，并且与传统网络架构不同，无需在层与层之间复制它。

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 表示网络的增长率。超过所有竞争方法的结果是大胆的，总体最佳结果是蓝色的。'+' 表示标准数据增强(翻译和/或镜像)。DenseNets 未进行数据增强（C10、C100、SVHN）的所有结果都是使用 Dropout 得到的。与 ResNet 相比，DenseNet 在使用更少参数的情况下获得了更低的错误率。在没有数据增强的情况下，DenseNet 的表现要好得多。

瓶颈层:虽然每一层只产生 k 个输出特征图，但它通常有更多的输入。在[37、11]中已经注意到，可以在每个 3 × 3 卷积之前引入一个 1×1 卷积作为瓶颈层，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于 DenseNet 特别有效，我们将具有这样一个瓶颈层的网络称为 DenseNet - B，即 BN - ReLU - Conv (1 × 1) - BN-ReLU- Conv (3 × 3)版本的 H_e。

压缩:为了进一步提高模型的紧凑性，我们可以减少过渡层的特征图数量。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 b θ mc 个输出特征图，其中 0 < θ ⩽1 称为压缩因子。当 θ = 1 时，跨越过渡层的特征图数量保持不变。我们将 θ < 1 的 DenseNet 称为 DenseNet - C，并在实验中设置 θ = 0.5。当同时使用 θ < 1 的瓶颈层和过渡层时，我们称我们的模型为 DenseNet - BC。

实现细节：在除了 ImageNet 以外的所有数据集上，我们实验中使用的 DenseNet 有三个稠密块，每个稠密块具有相等的层数。在进入第一个稠密块之前，对输入图像进行具有 16 个(或者是 DenseNet - BC 增长速度的两倍)输出通道的卷积。对于核大小为 3 × 3 的卷积层，输入的每一边都被零填充一个像素，以保持特征图大小不变。我们使用 1 × 1 卷积和 2 × 2 平均池化作为两个相邻密集块之间的过渡层。在最后一个稠密块的末尾进行全局平均池化，然后附加一个 softmax 分类器。3 个密集块中的特征图大小分别为 32 × 32、16 × 16、8 × 8。我们对基本的 DenseNet结构进行了实验，配置为{ L = 40，k = 12 }，{ L = 100，k = 12 }和{ L = 100，k = 24 }。对于 Dense NetBC，对配置为{ L = 100，k = 12 }、{ L = 250，k = 24 }和{ L = 190，k = 40 }的网络进行评估。

在 ImageNet 上的实验中，我们在 224 × 224 的输入图像上使用了具有 4 个密集块的 DenseNet - BC 结构。初始卷积层包含 2k 个大小为 7 × 7、步幅为 2 的卷积；所有其他层中的特征图数量也从设置 k 开始。我们在 ImageNet 上使用的确切网络配置如表 1 所示。

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 中的前 1 位和前 5 位错误率 ImageNet 验证集，用单作物/ 10 作物测试。

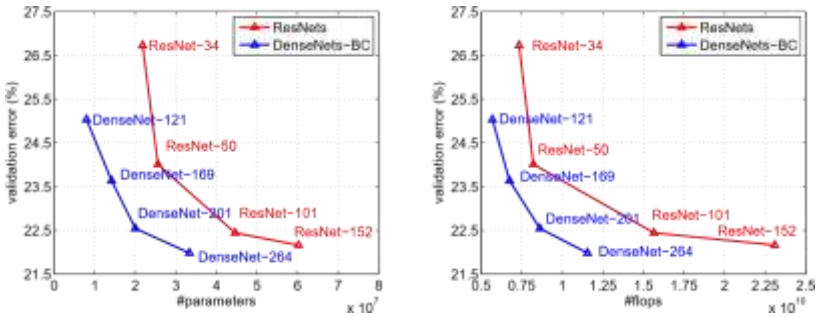

图 3：在 ImageNet,验证数据集上比较 DenseNets 和深度残差网络的 top - 1 错误率(单作物测试)，作为学习参数(左)和测试期间(右) FLOPs 的函数。

4. 实验

我们在几个基准数据集上实证了 DenseNet 的有效性，并与当前最先进的架构进行了比较，特别是与 ResNet 及其变体进行了比较。

4.1. 数据集

CIFAR： 两个 CIFAR 数据集[15]由 32 × 32 像素的彩色自然图像组成。CIFAR-10 (C10)由 10 和 100 类 CIFAR - 100 (C100)组成。训练集和测试集分别包含 50000 张和 10000 张图像，并保留 5000 张训练图像作为验证集。我们采用了这两个数据集[11,13,17,22,28,20,32,34]广泛使用的标准数据增强方案(镜像/平移)。我们在数据集名(例如, C10 +)的末尾用'+'标记表示该数据增强方案。对于预处理，我们使用通道均值和标准差对数据进行归一化。对于最后的运行，我们使用所有的 50000 张训练图像，并在训练结束时报告最终的测试误差。

SVHN： 街景房屋编号(Street View House Numbers，SVHN)数据集[24]包含 32 × 32 的有色数字图像。训练集有 73257 张图像，测试集有 26032 张图像，用于额外训练的图像有 531131 张。按照[7、13、20、22、30]的一般做法，我们使用所有的训练数据，不进行任何数据增强，并从训练集中分割出一个包含 6000 张图像的验证集。我们在训练过程中选择验证误差最低的模型，并报告测试误差。我们遵循[42]，将像素值除以 255，因此它们在[0、1]范围内。

ImageNet： 采用 ImageNet . ILSVRC 2012 分类数据集[2]包含 1 000 个类别的 120 万张图像用于训练，5 万张用于验证。我们对训练图像采用与[8、11、12]中相同的数据增强方案，并在测试时应用大小为 224 × 224 的单作物或 10 作物。遵循[11、12、13]，我们在验证集上报告了分类错误。

4.2. 训练

所有网络均采用随机梯度下降法(SGD)进行训练。在 CIFAR 和 SVHN 上，我们使用批处理大小 64 分别训练 300 和 40 个历元。初始学习率设置为 0.1，在训练历元总数的 50 % 和 75 %处分别除以 10。在 ImageNet 上，我们以 256 的批处理大小训练了 90 个历元的模型。学习率初始设置为 0.1，在历元 30 和 60 时降低 10 倍。需要注意的是，DenseNet 的一个幼稚的实现可能包含了内存的低效。为了减少 GPU 上的内存消耗，请参考我们关于 DenseNets 的存储高效实现的技术报告[26]。

在[8]之后，我们使用 10 - 4 的重量衰减和 0.9 的 Nesterov 动量[35]，而没有阻尼。我们采用[10]介绍的权重初始化。对于未进行数据增强的三个数据集，即 C10、C100 和 SVHN，我们在每个卷积层(除第一种)之后添加一个 dropout 层[33]，并将 dropout 率设置为 0.2。每个任务和模型设定仅评估一次试验误差。

4.3. Cifar 和 Svhn 的分类结果

训练不同深度的 Dense Nets，L 和增长率 k。在 CIFAR 和 SVHN 上的主要结果如表 2 所示。为了突出总体趋势，我们用粗体标出所有优于现有最先进水平的结果，用蓝色标出总体最佳结果。

准确度： 最显著的趋势可能源于表 2 的下一行，它表明 L = 190，k = 40 的 DenseNet

- BC 在所有 CIFAR 数据集上的表现都一致优于现有的最先进水平。其在 C10 +上 3.46 %的错误率和在 C100 +上 17.18 %的错误率明显低于宽 ResNet 架构实现的错误率[42]。我们在 C10 和 C100 (没有数据增强)上的最好结果更加令人鼓舞：两者都比使用 Drop - Path 正则化的 FractalNet 低近 30 % [17]。在 SVHN 上，当存在 dropout 时，L = 100，k = 24 的 DenseNet 也超过了当前宽 ResNet 取得的最好结果。然而，250 层的 DenseNet - BC 并没有比更短的 DenseNet - BC 进一步提高性能。这可能是由于 SVHN 是一个相对容易的任务，而极深的模型可能会过度拟合训练集。

容量： 在没有压缩层或瓶颈层的情况下，DenseNets 的性能一般会更好 L 和 k 增大。我们将这主要归因于模型能力的相应增长。这一点在 C10 +和 C100 +柱上得到了最好的证明。在 C10 +上，随着参数个数从 1.0 M 增加到 7.0 M 到 27.2 M，误差从 5.24 %下降到 4.10 %，最后下降到 3.74 %。在 C100 +上，我们观察到类似的趋势。这表明 DenseNets

可以利用更大、更深的模型增强的表征能力。这也表明它们不会遭受过拟合或残差网络的优化困难[11]。

参数设置：表 2 的结果表明，DenseNets 比备选架构(特别是深度残差网络)更有效地利用了参数。具有瓶颈结构和在过渡层降维的 DenseNetBC 特别具有参数效率。例如，我们的 250 层模型仅有 15.3 M 的参数，但它始终优于其他模型，如 FractalNet 和 Wide 深度残差网络，它们的参数都超过了 30M。我们还强调了 DenseNet - BC 在 L = 100 和 k = 12 的情况下，使用 90 %的参数实现了与 1001 层预激活 ResNet 相当的性能(例如, C10 +上 4.51 % vs 4.62 %的误差, C100 +上 22.27 % vs 22.71 %的误差)。图 4 (右面板)显示了这两个网络在 C10 +上的训练损失和测试误差。1001 层深度 ResNet 收敛于较低的训练损失值，但测试误差相似。下面我们对这一效应进行更详细的分析。

过拟合： 更有效地使用参数的一个积极的副作用是 DenseNets 倾向于减少过拟合。我们观察到，在没有数据增强的数据集上，DenseNet 架构相对于之前工作的改进尤为明显。在 C10 上，误差相对减少了 29 %，从 7.33 %减少到 5.19 %。在 C100 时，由 28.20 %降至 19.64 %，降幅约为 30 %。在我们的实验中，我们观察到了单一设置下的潜在过拟合：在 C10 上，增加 k = 12 到 k = 24 所产生的参数的 4 倍增长导致误差从 5.77 %小幅增加到 5.83 %。DenseNet - BC 瓶颈和压缩层似乎是对抗这一趋势的有效方法。

Imagenet 上的分类结果

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 架构进行了比较。为了确保两种架构之间的公平比较，我们通过采用[8]中公开的针对 ResNet 的 Torch 实现来消除所有其他因素，例如数据预处理和优化设置的差异。

我们简单地将 ResNet 模型替换为 DenseNetBC 网络，并保持所有实验设置与 ResNet 完全相同。

我们在表 3 中报告了 DenseNets 在 ImageNet 上的单作物和 10 作物验证误差。图 3 显示了 DenseNets 和深度残差网络的单作物 top - 1 验证误差随参数数量(左)和 FLOPs (右)的变化。图中显示的结果表明，DenseNets 的性能与最先进的深度残差网络相当，但需要更少的参数和计算量才能达到可比的性能。例如，具有 20M 参数的 DenseNet - 201 模型产生了与具有 40M 以上参数的 101 层 ResNet 相似的验证误差。从右面板可以观察到类似的趋势，它将验证误差绘制为 FLOPs 数量的函数：需要与 ResNet - 50 相同计算量的 DenseNet 和需要两倍计算量的 ResNet - 101。

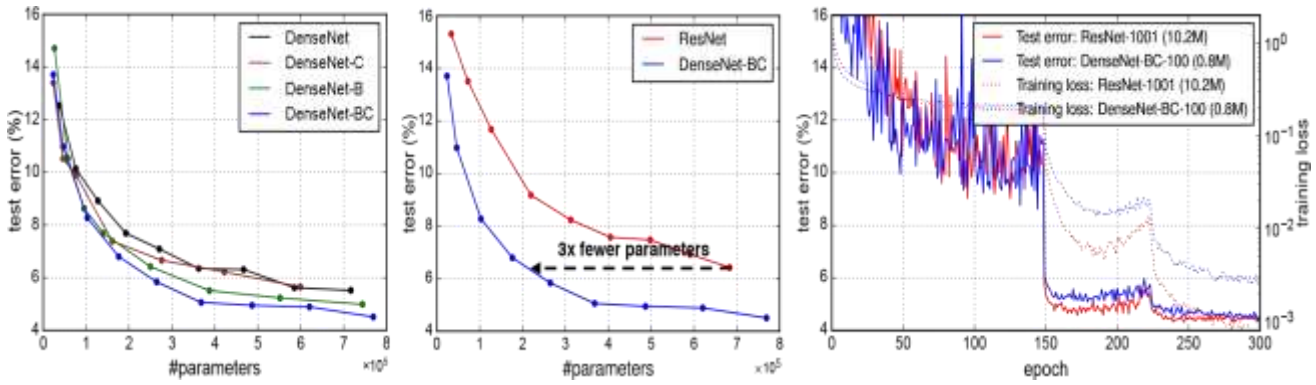

图 4：左边：DenseNet 变体在 C10 +上参数效率的比较。中：DenseNet － BC 与(预激活)深度残差网络的参数效率比较。DenseNet － BC 需要大约 1 / 3 的参数才能达到与 ResNet 相当的精度。右边：参数超过 10M 的 1001 层预激活 ResNet [12]和参数仅为 0.8 M 的 100 层 DenseNet 的训练和测试曲线。

值得注意的是，我们的实验设置暗示我们使用针对深度残差网络而不是针对 DenseNets 优化的超参数设置。可以想见，更广泛的超参数搜索可能会进一步提高 DenseNet 在 ImageNet 上的性能。

5. 讨论

从表面上看，DenseNets 与深度残差网络非常相似：(2)与式(1)不同。(1)只是对 H ' (•) 的输入是串联的，而不是加总的。然而，这种看似微小的修改所带来的影响，却导致了两种网络架构行为的本质不同。

模型紧凑度：作为输入串联的直接结果，任何一个 DenseNet 层学习到的特征图都可以被后续所有层访问。这鼓励了整个网络中的特征重用，并导致更紧凑的模型。

图 4 中的左两个图显示了一个实验的结果，该实验旨在比较 DenseNets 的所有变体(左)和一个可比较的 ResNet 体系结构(中)的参数效率。我们在 C10 +上训练多个深度不同的小型网络，并将它们的测试精度绘制为网络参数的函数。与其他流行的网络架构(如 AlexNet [16]或 VGG - net [29])相比，具有预激活的深度残差网络使用更少的参数，同时通常获得更好的结果[12]。因此，我们将 DenseNet (k = 12)与此体系结构进行比较。DenseNet 的训练设置与上一节相同。

从图中可以看出，DenseNet - BC 始终是 DenseNet 中参数效率最高的变体。此外，为了达到同样的准确度，DenseNet-BC 只需要深度残差网络(中间图)参数的 1 / 3 左右。这个结果与我们在图 3 中展示的 ImageNet 上的结果一致。图 4 右图显示，一个只有 0.8 M 可训练参数的 DenseNet - BC 能够达到与 1001 层(预激活) ResNet [12] (10.2 M 参数)相当的精度。

隐性的深度监管：密集卷积网络精度提高的一个解释可能是个别层通过较短的连接从损失函数中获得了额外的监督。人们可以通过解释 DenseNets 来执行一种"深度监督"。深度监督的好处已经在之前的深度监督网络(DSN)。)中显示出来，它在每个隐藏层上都有分类器，强制中间层学习具有判别性的特征。

DenseNets 以隐式的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个或三个过渡层向所有层提供直接监督。然而，DenseNets 的损失函数和梯度复杂度大大降低，因为所有层之间共享相同的损失函数。

随机连接 vs .确定性连接：密集卷积网络和残差网络的随机深度正则化之间存在有趣

的联系[13]。在随机深度中，残差网络中的层是随机掉落的，这使得周围的层之间产生了直接的联系。由于池化层从未被丢弃，该网络产生了与 DenseNet 类似的连接模式：如果所有中间层都被随机丢弃，那么在相同的池化层之间，任何两层都有小概率直接连接。虽然这些方法最终是完全不同的，但 DenseNet 对随机深度的解释可能为这种正则化器的成功提供了见解。

特征重用：通过设计，DenseNets 允许图层从其前面的所有层(虽然有时会穿过过渡层)中访问特征图。我们进行了一个实验，考察一个训练好的网络是否利用了这个机会。首先在 C10 + 上训练一个 Dense Net，L = 40，k = 12。对于一个块内的每个卷积层，我们计算分配给与层 s 的连接的平均(绝对)权重。图 5 显示了所有三个密集块的热图。平均绝对权重作为卷积层对其前几层的依赖性的替代。位于(' , s)位置的一个红点表明，该层对以前产生的 S-层蛋白特征图的平均使用较强。从图中可以观察到几种情况：

1. 所有层将其权重分散在同一块内的许多输入上。这表明，在同一个稠密块中，很早层提取的特征确实被深层直接使用。
2. 过渡层的权重也在前一个稠密块内的所有层中传播，表明信息从 DenseNet 的第一层到最后一层通过很少的间接流动。
3. 第二个和第三个密集块内的层一致为过渡层(三角形的顶行)的输出分配了最少的权重，表明过渡层输出了许多冗余特征(平均体重较低)。这与 DenseNet - BC 的强结果是一致的，即压缩了这些输出。
4. 虽然最后的分类层，显示在右边，也使用整个密集块的权重，但似乎有一个集中到最终的特征图，这表明在网络中可能会产生更多的高级特征。

6. 结论

我们提出了一种新的卷积网络架构，我们称之为密集卷积网络(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。我们发现 DenseNets 可以自然地扩展到数百层，同时没有优化困难。在我们的实验中，随着参数数量的增加，DenseNets 在准确率上往往会得到一致的提高，而没有任何性能下降或过拟合的迹象。在多个设置下，它在几个高度竞争的数据集上取得了最先进的结果。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。由于我们在研究中采用了针对残差网络优化的超参数设置，我们认为 DenseNets 在精度上的进一步提升可能是通过更细致地调整超参数和学习速率来实现的。

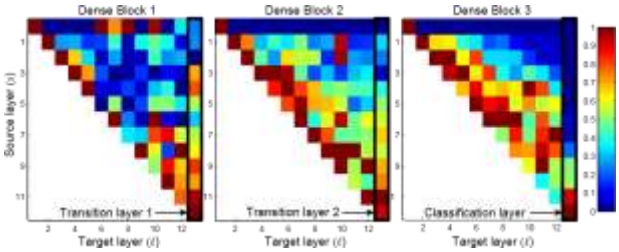

图 5：训练好的 DenseNet 中卷积层的平均绝对滤波权重。像素(s , ')的颜色编码了密集块内连接卷积层 s 到'的权值的平均 L1 范数(通过输入特征图的数量进行归一化)。黑色矩形突出的三列分别对应两个过渡层和分类层。第一行编码连接到密集块的输入层的权重。

DenseNets 遵循简单的连接规则，自然地融合了身份映射、深度监督和多样化深度的特性。它们允许在整个网络中进行特征重用，因此可以学习更紧凑的模型，并且根据我们的实验，可以学习更准确的模型。由于其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 来研究这种特征迁移。

致谢：作者得到了 NSF III - 1618134、III - 1526012、IIS - 1149882、海军研究基金办公室 N00014 - 17 - 1 - 2175 以及比尔和梅林达 • 盖茨基金会的部分支持。GH 获得中国博士后理事会国际博士后交流奖学金项目(No.20150015)资助。ZL 获得国家重点基础研究发展计划(2011CBA00300、2011CBA00301)、国家自然科学基金

(61361136003)的资助。我们还要感谢丹尼尔·塞德拉、杰夫·普莱斯和余淼杰进行了许多富有洞见的讨论。

7. References

[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. FeiFei. 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. 8

[5] L. Gatys, A. Ecker, and M. Bethge. A neural algorithm of artistic style. *Nature Communications*, 2015. 8

[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. 2, 3, 5, 7

[13] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016. 1, 2, 5, 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. 1, 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. Gradientbased 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. Deeplysupervised nets. In *AISTATS*, 2015. 2, 3, 5, 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

[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

[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. 6

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

[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. 2, 3

[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 2, 3, 4

[38] S. Targ, D. Almeida, and K. Lyman. Resnet in resnet: Generalizing residual architectures. *arXiv preprint arXiv:1603.08029*, 2016. 2

[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 dagcnns. 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