紧密连接卷积网络

摘要

最近的研究表明，如果在靠近输入的层和靠近输出的层之间包含较短的连接，卷积网络可以更深入、更准确、更高效。本文中，我们采用了这个观点，并且介绍了紧密卷积网络，该网络以一种前馈方式将每一层与其他层连接起来。然而，传统的L层卷积网络仅有L个连接——位于每一层和随后一层之间——我们的网络有$\frac{L(L+1)}{2}$个直接连接。对于每一层，所有前置层的特征图都将作为输入，而它自身的特征图将作为所有后置图的输入。紧密网络有几个竞争优势：它们缓解了梯度消失问题，增强了特征的传播，鼓励特征重用，并且充分的减少了参数个数。我们采用了四个高竞争性的对象识别指标任务（CIFAR-10, CIFAR-100, SVHN, ImageNet）来评估我们提出的模型。紧密网络相较于大部分顶尖的网络有着显著的优势，并且它用更少的计算来达到更高的表现。

1. 简介

卷积神经网络已经成为机器学习方法中视觉对象识别的主导方法。尽管它们最初在20年前就被提了出来， 计算机硬件和网络上的进步使得CNNs最近才得以实现。最早的LeNet5模型由5层构成，VGG有19层，而去年的 Highway Networks 和 Residual Networks(ResNets) 突破了100层的障碍。

随着CNNs越来越深入，一个新的研究问题出现了： 当输入或者梯度相关的信息经过很多层时，在它到达终点时会渐渐消失或者被'洗净'。最近许多出版物提出了这个或与之相关的问题。 ResNets 和 Highwan Networks 通过身份识别将信号从一层传递到下一层。随机深度通过训练时随机放弃某些层来改进信息和梯度流，并且缩短了 ResNets。FractalNets用不同数量的卷积块重复组合一系列平行的层序列，来获得一个名义上相当大的深度。尽管这些不同的方法在网络拓扑和训练程序上有所不同，但它们都有一个核心的特点：它们都在前置层和后置层中构建了短路径。

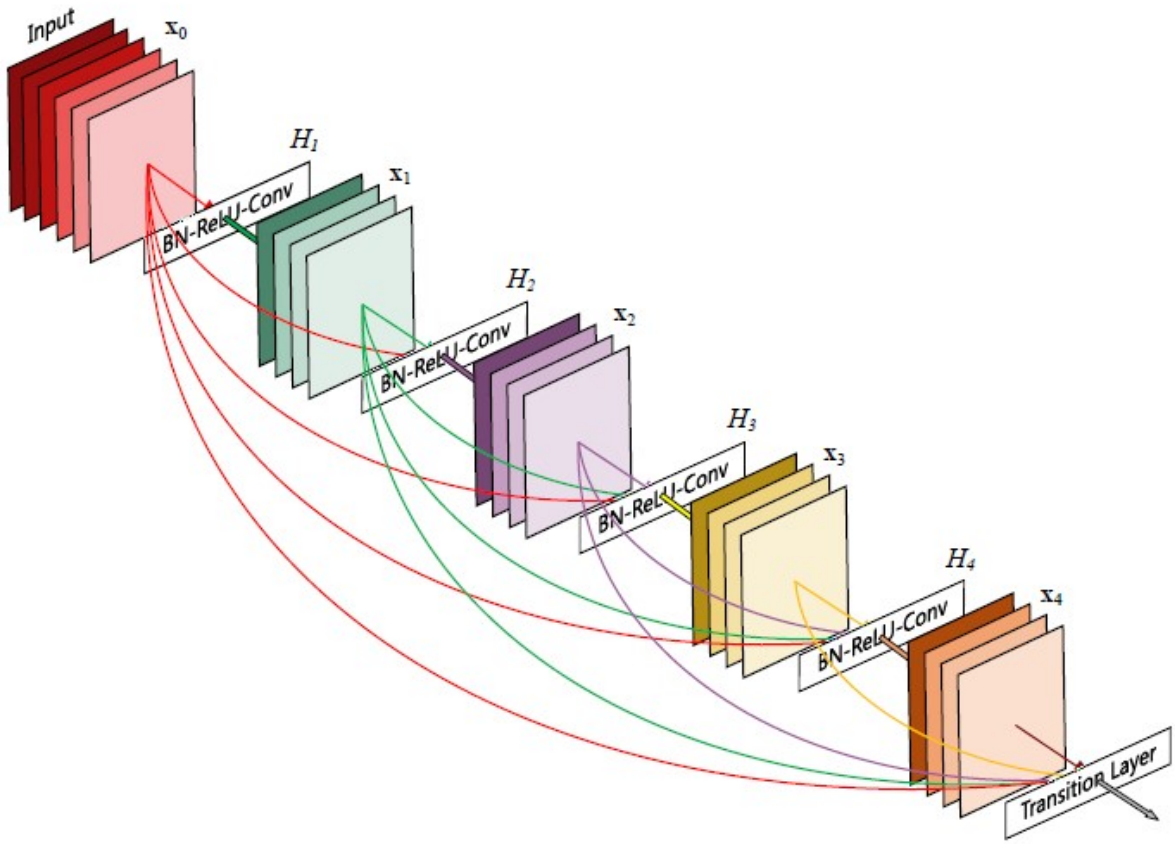

图1： 一个五层的紧密块，其增长速率为$k = 4$。每一层都将所有前置层的特征图作为输入。

本文中，我们将这种观点放入一个简单的连接模式并提出了这样一种结构：为了最大限度确保网络中层与层之间的信息流，我们将所有层直接连接起来（用相匹配大小的特征图）。为了保持前馈性，每一层接受来自所有前置层的额外输入，并将其自身的特征图传递给所有后置层。图1为这种布局的示意图。最关键的是，与 ResNets 不同，我们从不在它们被传递到下一层时利用求和来组合这些特征，而是采用串接的方式。因此，ℓ^{th} 有ℓ 个输入，这些输入由所有前置卷积块的特征图构成。它自身的特征图将会被传递给所有$L - \ell$ 个后置层。这使得 L 层的网络共有$\frac{L(L+1)}{2}$ 个连接，而不是和传统卷积网络一样只有 L 个。由于这种紧密连接的特性，我们把我们的这种方法称之为*紧密卷积网络*（紧密网络）。

一个可能这种密集的连接模式的直观效果是它需要的参数比传统的卷积网络少，无需重新学习冗余特征图。传统的前馈结构可以看作是一种状态的算法，它是从层到层传递的。每个层从其上一层读取状态并写入后续层，它改变了状态但也传递了需要保存的信息。ResNets 清楚的保存信息是通过附加的标识转换。 ResNets 最近的改进表明，有很多层的贡献很少，实际上可以在训练期间随机丢弃，使得 ResNet 的状态类似于（展开的）递归神经网络，但是 ResNets 的参数数目要大得多，因为每一个层都有自己的权重。我们提出的 DenseNet 体系结构清楚的区分添加到网络的信息和保存的信息。 DenseNet 层很窄（例如每层12个filter），添加少量的特征图的‘集体知识’的网络，保持其余的特征图不变，最终的分类器的结果是基于网络的所有特征图。

除了更好的参数效率之外，DenseNets 的一大优势是改善了整个网络中的信息流和梯度，这使得它们易于训练。每一层都可以直接从损失函数和原始输入信号中获得梯度，从而产生隐含的深层监督。这有助于深入网络体系结构的训练。此外，我们还观察到密集连接具有正则化效应，缓解了训练集过小的过拟合现象。

我们在四个高竞争性框架的数据集（CIFAR-10, CIFAR-100, SVHN, ImageNet）上评估了DenseNets。我们的模型能够确保算法保持高准确度的同时需要更少的参数。另外，我们的算法在大部分数据集上都比如今顶尖的算法更加有优势。

2. 相关工作

网络架构的探索自从最初的发现以来一直是神经网络研究的一部分。最近神经网络的流行再度兴起了这个研究领域。现代网络中越来越多的层放大了架构之间的差异，并且激发了对不同连接模式的探索以及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已经在20世纪80年代的神经网络文献中被研究。他们的开创性工作主要集中在逐层训练的全连接的 multi-layer 上。在[9,23,30,40]中，已经发现通过跳跃连接来利用CNN中的多级特征对于各种视觉任务是有效的。 在我们的研究工作的同时，[1]得出了一个与我们的跨层连接类似的网络的纯理论框架。

Highway Networks [33]是最早提出100层网络架构之一，它提供了一种有效的方式来训练超过100层的end-to-end网络。使用bypassing paths和门控单元，数百层的Highway Networks可以毫无困难地优化。bypassing paths（旁路路径）被认为是训练这些非常深网络的关键因素。ResNets [11]进一步支持这一点，其中纯身份映射被用作旁路路径。 ResNet在许多具有挑战性的图像识别，定位和检测任务（如ImageNet和COCO目标检测）方面取得了非常厉害的创纪录的性能[11]。最近，Stochastic depth成功训练了1202层ResNet [13]。 Stochastic depth通过在训练期间随机丢弃层来改善深度残留网络的训练。这表明并不是所有的层都是需要的，并且强调在深度（剩余）网络中存在大量的冗余。 我们的论文部分受到这一现象的启发。 具有pre-activation的ResNets也有助于训练1000多层最先进的网络[12]。

使网络更深的正交方法（例如，借助跳过连接）是增加网络宽度。GoogLeNet [35,36]使用一个“Inception module”，将不同大小的滤波器产生的特征图连接起来。 在[37]中，提出了具有广泛的广义残差块的ResNet的变体。事实上，只要增加每层ResNets中的滤波器数量，只要深度足够，就可以提高其性能[41]。FractalNets也使用广泛的网络结构在几个数据集上获得有竞争力的结果[17]。

DenseNets不是从极其深层或广泛的体系结构中绘制代表性的力量，而是通过特性重用来挖掘网络的潜力，产生易于训练和具有高效的参数精简模型。把不同层学习的特征图连接起来会增加后续层输入的变化，并提高效率。这构成了DenseNets和ResNet之间的主要区别。 与Inception网络[35,36]相比，DenseNets更加简单和高效。

还有其他引人注目的网络架构创新，已经取得了有竞争力的成果。NIN网络结构包括卷积层滤波器的多层感知器，用来提取更复杂的特征。DSN网络的内部层被辅助分类器直接监督的，它能够加强前面层接收的梯度。Ladder Networks （梯形网络）[26, 25]引入了自动编码器的横向连接，在半监督的学习任务中产生了非常高的精度。在[38]中，DFNS通过结合不同基础网络的中间层来提高信息流，增加具有最小化重建损失值性质的网络路径，可以改善图像分类模型。

3. 紧密网络

考虑一张单一的图像x_0 通过一个卷积神经网络。这个网络有 L 层，每层实现一个非线性变换$H_\ell(\cdot)$，ℓ 是层的编号，$H_\ell(\cdot)$ 可以是批量归一化的复合函数（例如Batch Normalization），rectified线性单元 (ReLU)，池化或者卷积，x_ℓ 作为 ℓ^{th} 层的输出。

ResNets. 传统的卷积是把前馈网络 ℓ^{th} 层的输出作为$(\ell+1)^{th}$ 层的输入[16]，产生了如下的转换关系：$x_\ell = H_\ell(x_{\ell-1})$ 。ResNets [11]添加一个跳过连接绕过非线性变换的特征函数：

$$x_\ell = H_\ell(x_{\ell-1}) + x_{\ell-1} \tag{1}$$

resnets的优点是梯度直接通过特征函数从后面层流向前面的层，然而，特征函数和Hℓ输出求和可能阻碍网络中的信息流。

Dense connectivity. 为了进一步改善层之间的信息流，我们提出了不同的连接模式：我们引入了从任何层到所有后续层的直接连接。 图1示意性地示出了由此产生的DenseNet的布局。 因此，第ℓ^{th}层接收所有前面的层$x_0,\dots,x_{\ell-1}$,的特征图作为输入：

$$x_\ell = H_\ell([x_0, x_1, \dots, x_{\ell-1}]) \tag{2}$$

其中$[x_0, x_1, \dots, x_{\ell-1}]$ 是指在层$0,\dots,\ell-1$ 中产生的特征图的拼接。 由于密集的连接性，我们把这种网络结构称为密集卷积网络（DenseNet）。$H_\ell(\cdot)$ 的多输入在式（2）为一个张量。

Composite function. 由[12]推导，我们定义$H_\ell(\cdot)$ 为三个连续操作的复合函数：批量归一化（BN）[14]，其次是（ReLU）[6]和3×3卷积（Conv）。

Pooling layers. 方程（2）中使用的连接操作在特征图的大小改变时是不可行的。（个人理解是，第一步的dense connectivity，如果特征图的尺寸不一致，不能进行连接操作的） 然而，卷积网络的一个重要组成部分是下采样层，它改变了特征图的大小。 为了简化体系结构中的下采样，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*（transition layers），对它们做卷积和池化。 在我们的实验中使用的过渡层由batch normalization层和1×1卷积层，然后是2×2平均池化层组成。

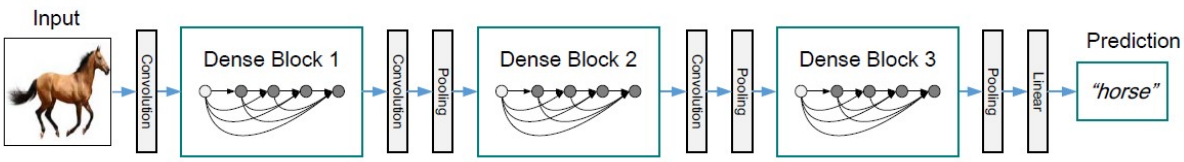

图2：一个拥有三个密集块的深层密集网络。两个邻近块之间的层是指转换层，通过卷积和池化来改变特征图大小。

Growth rate. 如果每个函数H_ℓ 产生k个 f特征图，由此可见，ℓ^{th} 层有$k_0 + k \times (\ell-1)$ 输入特征图，其中k_0 是输入层的通道数（如果是初始的RGB，就是k_0 就是3）。DenseNet和现有的网络体系结构的一个重要的区别是，DenseNet可以有很窄的层，例如，$k = 12$ 。我们将超参数k 作为网络的增长率。我们在第4节中提到，一个相对较小的增长率足以在我们测试的数据集上获得了最先进的结果。对此的一个解释是，每一层都可以访问其块中的所有前面的特征图，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。 每个层都将自己的k 个特征图添加到这个状

态。 增长率规定了每个层对全局状态贡献的新信息量。 全局状态一旦写入，就可以从网络中的任何地方访问，与传统的网络体系结构不同的是，并不需要一层一层地复制它。

Bottleneck layers. 虽然每个层只输出k 个特征图，但它通常有更多的输入。在[36, 11] 已经指出，1×1卷积可以作为Bottleneck layers在每3×3卷积来减少输入特征图的数量（这部分是降维），从而提高计算效率。我们发现这个设计对于DenseNet特别有效，我们把这个Bottleneck layers称为我们的网络，即对于H_ℓ 层的BN-ReLU-Conv（1×1）-BN-ReLU-Conv（3×3）版本，DenseNet-B。在我们的实验中，我们让每个1×1卷积产生4k个特征图

Compression. 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。 如果一个密集块包含m 个特征图，我们让下面的过渡层产生$\lfloor \theta m \rfloor$ 输出特征图，其中$0 < \theta < 1$ 被称为压缩因子。 当$\theta = 1$ 时，过渡层上的特征图的数量保持不变。 我们称DenseNet的$\theta < 1$ 为DenseNet-C，在实验中设定$\theta = 0.5$ 。 当使用$\theta < 1$ 的瓶颈和过渡层时，我们将模型称为DenseNet-BC。

Implementation Details. 在除ImageNet以外的所有数据集上，我们实验中使用的DenseNet有三个密集块，每个块都有相同数量的层。在进入第一密集块之前，对输入图像执行 16（或DenseNet-BC增长率的两倍）输出通道的卷积。 对于卷积核大小为 3×3 的卷积层，输入的每一边都被填充一个像素以保持特征图大小的不变。我们使用1×1卷积，然后使用2×2平均池化作为两个连续密集块之间的过渡层。 在最后一个密集块的末尾，采用一个全局平均池化，然后附加一个 softmax 分类器。三个密集块中的特征图大小分别是32×32,16×16和8×8。 我们试验了配置分别为$L = 40, k = 12$ ，$L = 100, k = 12$ 和$L = 100, k = 24$ 的基本DenseNet结构。对于DenseNetBC，评估配置为$L = 100, k = 12$ ， $L = 250, k = 24$ 和$L = 190, k = 40$ 的网络。

在ImageNet的实验中，我们使用了DenseNet-BC结构，在224×224的输入图像上有4个密集块。 初始卷积层包括为步长为 2 的 7×7 卷积的 2k 个;所有其他层中的特征图的数量设置为 k 。我们在 ImageNet 上使用的确切网络配置如表1所示。

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： ImageNet的DenseNet架构。所有网络的增长率为$k = 32$ 。表中所示的每个“conv”层和 BN-ReLU-Conv 相一致。

4. 实验

我们的经验验证了 DenseNet 在几个标准数据集上的有效性，并与现有的 ResNe t及其变体架构进行了比较。

4.1 Datasets

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个图像和100个类别的CIFAR-100（C100）组成。 训练和测试集合分别包含50,000和10,000个图像，并且我们有5000个训练图像作为验证集。我们采用广泛用于这两个数据集的标准数据增强方案（镜像/移位）[11,13,17,22,27,20,31,33]。 我们用数据集名称末尾的“+”标记（例如C10 +）表示采用的数据增强。对于预处理，我们使用the channel means 和 standard deviations对数据进行归一化处理。 为了最后的运行，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. 街景房屋号码（SVHN）数据集[24]包含32×32彩色数字图像。 训练集中有73,257幅图像，测试集中有26,032幅图像，531,131幅图像用于额外训练。按照惯例[7,13,20,22,29]，我们使用所有的训练数据没有任何数据增强，并且从训练集中分离出具有6000个图像的验证集合。 我们选择训练期间验证错误最低的模型并报告测试错误。 我们遵循[41]并将像素值除以255，使其在[0,1]范围内。

ImageNet. ILSVRC 2012 分类数据集[2]包括来自1,000个类的120万个用于训练的图像和50,000个用于验证的图像。 我们采用与[8,11,12]中相同的数据增强方案来训练图像，并且在测试时应用尺寸为224×224的single-crop 或 10-crop 。 在[11,12,13]之后，我们在验证集上报告分类错误。

4.2 训练

所有的网络都使用随机梯度下降（SGD）进行训练。 在CIFAR和SVHN上，我们用64 batch size 分别训练300和40轮。 最初的学习率设定为0.1，在训练时期总数中的50％和75％除以10。 在ImageNet上，采用batch size为256训练90轮。学习速率初始设置为0.1，在30和60轮时降低10倍。由于GPU内存的限制，我们最大的型号（DenseNet-161）以mini-batch size 128进行训练。为了弥补较小batch size，我们训练这个模型100轮，在90轮时采用除以10的学习率。

我们使用10^{-4} 的权重衰减和 0.9 的Nesterov动量[34]，其中Nesterov动量不衰减。 我们采用由[10]引入的权重初始化。 对于没有数据增强的三个数据集，即C10，C100和SVHN，我们在每个卷积层（除了第一个卷积层）之后添加一个丢失层[32]，并将丢失率设置为0.2。 测试错误仅针对每个任务和模型设置评估一次。

4.3 CIFAR 和 SVHN 上的分类结果

我们用不同的深度L和增长率k来训练DenseNets。 表2列出了CIFAR和SVHN的主要结果。为了突出总体趋势，我们将超过现有技术水平的以粗体显示，并以蓝色表示最好的结果。

Method	Depth	Params	C10	C10+	C100	C100+	SVHN
Network in Network [22]	-	-	10.41	8.81	35.68	-	2.35
All-CNN [31]	-	-	9.08	7.25	-	33.71	-
Deeply Supervised Net [20]	-	-	9.69	7.97	-	34.57	1.92
Highway Network [33]	-	-	-	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 [41]	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 代表网络的增长率。超过所有竞争方法的结果用 **粗体** 标明,总体上最好的结果用 蓝色 标明。“ + ” 表示标准数据增强（转化 和\或 镜像）。“ * ”表示我们自己运行的结果。所有没有数据增强的数据集（C10, C100, SVHN）的结果都是用 Dropout （随机丢弃）得到的。 Dense 比 ResNet 用了更少的参数，但得到了更低的错误率。如果没有数据增强，DenseNet 的表现会更好。

Accuracy. 可能最明显的趋势可能起源于表2的最后一行，这表明在所有CIFAR数据集上，具有 $L = 190$ 和 $k = 40$ 的DenseNet-BC一致地优于现有技术水平。 它在 C10+ 上的错误率为3.46%，在 C100+ 上的错误率为17.18%，明显低于 ResNet 架构[41]的错误率。我们在 C10 和 C100 上的最好结果（没有数据增加）更令人鼓舞：两者都比FractalNet低30%，并且具有下降路径正则化[17]。在SVHN上，当丢失层时，$L = 100$ 和 $k = 24$ 的DenseNet也超过了ResNet所取得的最好结果。然而，250层的DenseNet-BC并没有进一步改善其性能。 这可以解释为SVHN是一个相对容易的任务，而且深的模型可能会产生过拟合现象。

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：ImageNet 验证集上的 top-1 和 top-5 错误率（single-crop / 10-crop 测试）

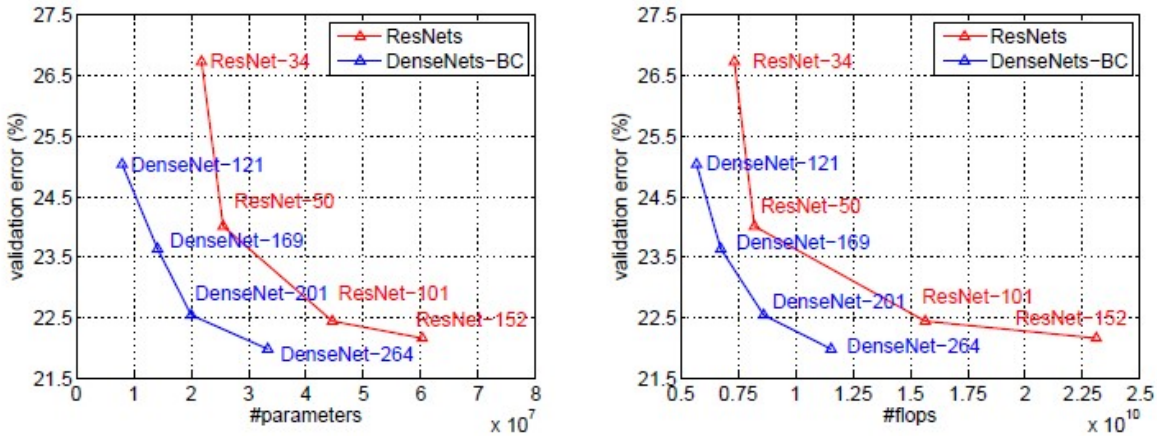

图3：ImageNet验证集上 DenseNets 和 ResNets 的 top-1 （single-crop）错误率随参数个数（左）和FLOPS（右）变化曲线的比较

Capacity. 在没有压缩或Bottleneck layers的情况下，随着L和k增加，DenseNets表现出更好的性能。 我们把这主要归因于模型能力的相应增长。 这最好由 C10 + 和 C100 + 列表示。 在C10 +上，随着参数从1.0M增加到7.0M，再增加到 27.2M，误差从 5.24% 下降到 4.10%，最终下降到 3.74%。在C100 +上，我们观察到了类似的趋势。 这表明 DenseNets 可以利用越来越深的模型的代表性力量。 这也表明它们不会出现过拟合或残余网络的优化困难[11]。

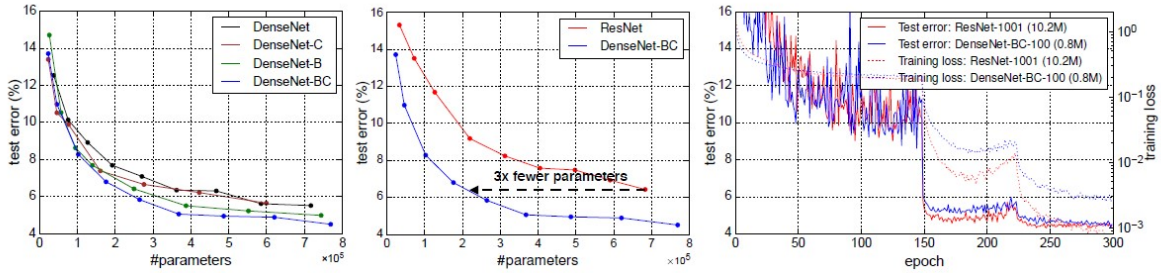

图4：*左：不同 DenseNet 版本在 C10+ 上参数效率的比较。中*：DenseNet-BC 和（前馈）ResNets 的参数效率比较。达到相同准确率时，DenseNet-BC 只需要 前馈 ResNets 参数个数的 1/3 。*右*：共1001层、参数个数为 10M 的前馈 ResNet网络和 100层、参数个数为0.8M 的 DenseNet 的训练和测试曲线。

Parameter Efficiency. 表2的结果表明，densenets利用参数比其他结构更有效（特别是resnets）。在过渡层的瓶颈结构和降维densenet BC特别有效。例如，我们的250层模型只有15.3M的参数，但它一直优于其他模型，如FractalNet和Wide ResNets有超过30M的参数。 我们还强调，$L = 100$ 和 $k = 24$ 的DenseNet-BC（例如，C10+ 上的4.51%比4.62%，C100 +上的22.27%比22.71%） 与1001层预激活ResNet使用90%参数的可达到同等的性能。图4显示了这两个网络在C10 +上的训练损失和测试误差。 1001层深的ResNet收敛到较低的训练损失值，但有类似的测试错误。我们在下面进行更详细地分析。

Overfitting. 更有效地使用参数的一个积极副作用是 DenseNets 不易过拟合。 我们观察到，在没有数据增强的数据集上，DenseNet 体系结构相对于之前工作的改进尤其明显。 在C10上，这一改进使得错误率从 7.33% 降至 5.19%，相对减少了 29%。 在 C100 上，减幅从 30% 左右到 28.20% 再下降到 19.64%。在我们的实验中，我们发现潜在的过拟合单一的设置：在C10，由增加 $k = 12$

到 $k = 24$ 产生的参数的 4 倍增长导致误差从 5.77% 适度增加到 5.83%。 DenseNet-BC bottleneck 和压缩层似乎是对付这一趋势的有效方法。

4.4 ImageNet上的分类结果

我们在ImageNet分类任务中评估不同深度和增长率的DenseNet-BC，并将其与最先进的ResNet体系结构进行比较。 为了确保两种架构之间的公平比较，我们消除了所有其他因素，如数据预处理和优化设置的差异，采用[8]的ResNet的公开Torch实现。我们只需用DenseNet-BC网络替换ResNet模型，并保持所有的实验设置与用于ResNet的设置完全相同。 唯一的例外是，由于GPU内存限制，我们最大的DenseNet模型采用mini-batch size为128来进行训练，在训练这个模型100轮，第90次学习率下降，以补偿较小的batch size。我们在表3中的ImageNet上报告了DenseNets的single-crop和10-crop验证错误。图3显示了DenseNets和ResNets的single-crop top-1验证错误作为参数数量（左）和FLOP（右）。图中显示的结果表明，DenseNets与最先进的ResNet相媲美，同时需要显着减少的参数和计算来实现比较好的性能。 例如，具有20M参数模型的DenseNet-201产生与具有超过40M参数的101层ResNet相似的验证误差。从右侧面可以观察到类似的趋势，它将验证误差绘制为FLOP数量的函数：一个DenseNet需要尽可能多的ResNet-50执行与ResNet-101相同的计算，这需要两倍计算。

值得注意的是，我们的实验设置意味着我们使用为优化的ResNets而不是DenseNets超参数设置。 可以想像，通过最多两个或三个过渡层提供对所有层的直接监督。然而，损失函数和梯度densenets基本上不太复杂，因为同样的损失函数是各层之间共享。可以想象，更广泛的超参数搜索可以进一步提高ImageNet上DenseNet的性能。

5. Discussion

从表面上看，DenseNets和ResNet非常相似：式（2）与式（1）的区别只是输入$H_\ell(\cdot)$ 连接起来而不是加和。 然而，这个看起来很小的修改的影响导致两个网络架构的本质大不相同。

Model compactness. 输入连接的直接结果是，由任何DenseNet层学习的特征图可以被所有后续层访问。 这使得整个网络的功能重用，并使模型更紧凑。图4中的左边两张图显示实验的结果，该实验的目的是比较所有DenseNets变体的参数效率（左）以及同等的ResNet架构（中）。 我们在C10+上训练多个不同深度的小网络，并将他们的测试精度绘制成网络参数的函数。 与其他流行的网络体系结构（如AlexNet [16]或VGG-net [28]）相比，具有预激活功能的ResNets使用更少的参数，却通常会取得更好的结果[12]。 因此，我们比较DenseNet（$k = 12$ ）与这个架构。 DenseNet的训练设置与上一节保持一致。

该图显示DenseNet-BC始终是DenseNet的最多参数有效变体。 此外，为了达到相同的准确度，DenseNet-BC只需要ResNets（中间图）的大约1/3的参数。 这个结果与我们在图3中给出的ImageNet的结果是一致的。图4中的右图显示了只有0.8M可训练参数的DenseNet-BC能够达到与1001层（预激活） ResNet [12]有10.2M参数。

Implicit Deep Supervision. 对密集卷积网络的改进精确度的一个解释可能是单个层通过较短的连接从损失函数接收额外的监督。 人们可以解释DenseNets进行一种“深度监督”。 深度监督的好处之前已经在深度监督的网络（DSN; [20]）中显示出来，它们在每一个隐藏层都附加了分类器，加强中间层学习判别特征。DenseNets以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至多两

个或三个过渡层提供对所有层的直接监督。 然而，DenseNets的损失函数和梯度要复杂得多，因为所有层之间共享相同的损失函数。

Stochastic vs. deterministic connection. 密集卷积网络与残差网络的随机深度正则化之间存在着一个有趣的联系[13]。 在随机深度中，剩余网络中的层随机丢弃，这会在周围层之间建立直接连接。 由于池层永远不会丢失，因此网络会产生与DenseNet相似的连接模式：如果所有中间层都是随机丢弃的，则在相同池层之间的任何两层之间有一个小概率被直接连接。 尽管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正规化的成功提供了见解。

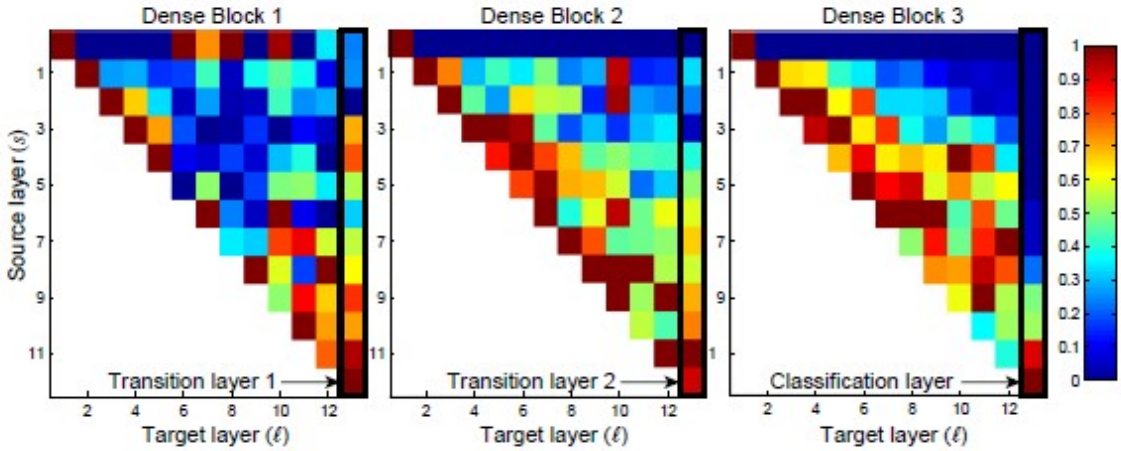

图5：训练Denser时卷积层的绝对平均滤波器权重。像素(s, ℓ) 的颜色表示同一个密集块中连接卷积层 s 和 ℓ 的weights 的 L1 范数。用黑色方框强调的三列对应两个转化层和一个分类层。第一行表示密集块中连接输出层的 weights。

Feature Reuse 通过设计，DenseNets允许层访问来自其所有先前层的特征图（尽管有时通过过渡层）。 我们进行一项实验来调查一个训练有素的网络是否利用这个机会。 我们首先在 C10+ 上训练一个DenseNet，$L = 40$ 和 $k = 12$ 。每个卷积层ℓ 块内，我们计算分配给层 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。 平均绝对权重用作卷积层在其前面层上的依赖性的替代物。 位置 (ℓ, s) 上的红点表示层 ℓ ，平均使用之前产生的 s 层的特征图。 可以从图中得到几个观察结果：

1. 所有层在同一个块内的许多输入上分布权重。 这表明，非常早期层提取的特征实际上直接被整个同一密集块中的深层使用。

2. 过渡层的权重也将它们的权重分布在前一个密集块内的所有层上，这表明DenseNet从第一层到最后一层的信息流通过几乎没有间接传递。

3. 第二和第三密集块内的层一致地将最小权重分配给过渡层的输出（三角形的顶部行），表明过渡层输出许多冗余特征（平均具有低权重）。 这与DenseNet-BC的强大结果保持一致，在这些结果中，这些输出被压缩。

4. 虽然右边显示的最后一个分类层也使用整个密集块的权重，但似乎更专注最后的特征图，它表明在网络后面层可能会产生更多的高级特征。

6. Conclusion

我们提出了一种新的卷积网络体系结构，我们称之为密集卷积网络（DenseNet）。 它引入具有相同特征图大小的任何两个层之间的直接连接。 我们发现DenseNets可以自然地扩展到数百层，同时不会出现优化困难。在我们的实验中，DenseNets随着参数数量的增加精度也随之提高，而没有任何性能下降或过拟合的现象。 在多种设置下，它在几个常用数据集上实现了最先进的结果。 而且，DenseNets采用更少的参数和更少的计算来实现最先进的性能。因为我们在本研究中采用了针对残差网络优化的超参数设置，所以我们相信通过更详细地调整超参数和学习率，可以使DenseNets准确度的进一步提高。 DenseNets遵循一个简单的连接规则，自然而然地结合了恒等映射，深度监督和深度多样化。它们允许在整个网络中重用特征，因此它们可以学习更紧凑，并且在我们的实验结果中更精确的模型。因为DenseNets内部表示紧凑，特征冗余较少，所以可以很好地进行各种基于卷积特征的计算机视觉任务的特征提取，例如[4,5]。 我们计划在未来的工作中研究DenseNets的这种特征变换。