导语

shortcut(或shortpath，中文“直连”或“捷径”)是CNN模型发展中出现的一种非常有效的结构，本文将从Highway networks到ResNet再到DenseNet概述shortcut的发展。

前言

自2012年Alex Krizhevsky利用深度卷积神经网络（CNN）（AlexNet[1]）取得ImageNet比赛冠军起，CNN在计算机视觉方面的应用引起了大家广泛地讨论与研究，也涌现了一大批优秀的CNN模型。研究人员发现，网络的深度对CNN的效果影响非常大，但是单纯地增加网络深度并不能简单地提高网络的效果，由于梯度发散，反而可能损害模型的效果。而shortcut的引入就是解决这个问题的妙招。本文主要就模型发展中的shortcut展开讨论。欢迎大家多多批评指正。

一、Highway networks

Highway [2] 是较早将shortcut的思想引入深度模型中一种方法，目的就是为了解决深度网络中梯度发散，难以训练的问题。我们知道，对于最初的CNN模型（称为“plain networks”，并不特指某个模型框架），只有相邻两层之间存在连接，如图1所示（做的图比较丑，请多担待），x、y是相邻两层，通过W_H连接，通过将多个这样的层前后串接起来就形成了深度网络。相邻层之间的关系如下，

$$\mathbf{y} = H(\mathbf{x}, \mathbf{W_H}).$$

其中H表示网络中的变换。

图1

为了解决深度网络的梯度发散问题，Highway在两层之间增加了（带权的）shortcut（原文中并没有使用这个名词，为统一起见，采用术语shortcut）。两层之间的结构如图2所示，

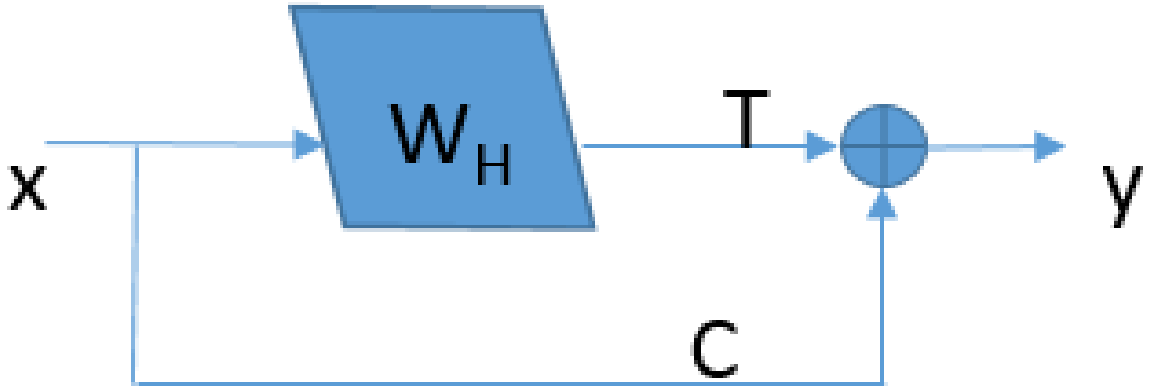


图2

x，y的关系如下式，

$$\mathbf{y} = H(\mathbf{x}, \mathbf{W_H}) \cdot T(\mathbf{x}, \mathbf{W_T}) + \mathbf{x} \cdot C(\mathbf{x}, \mathbf{W_C}).$$

其中设置C=1-T，可以将上式改写为，

$$\mathbf{y} = H(\mathbf{x}, \mathbf{W_H}) \cdot T(\mathbf{x}, \mathbf{W_T}) + \mathbf{x} \cdot (1 - T(\mathbf{x}, \mathbf{W_T})).$$

作者将T称为“transform gate”，将C称为“carry gate”。输入层x是通过C的加权连接到输出层y。通过这种连接方式的改进，缓解了深度网络中的梯度发散问题。Highway networks与plain networks的训练误差对比如图3所示。可以看到对于plain networks，随着层数的增加，训练误差在逐步扩大，而对于highway networks，训练误差比较稳定，显著低于plain networks的误差，尤其是在层数非常深的时候。

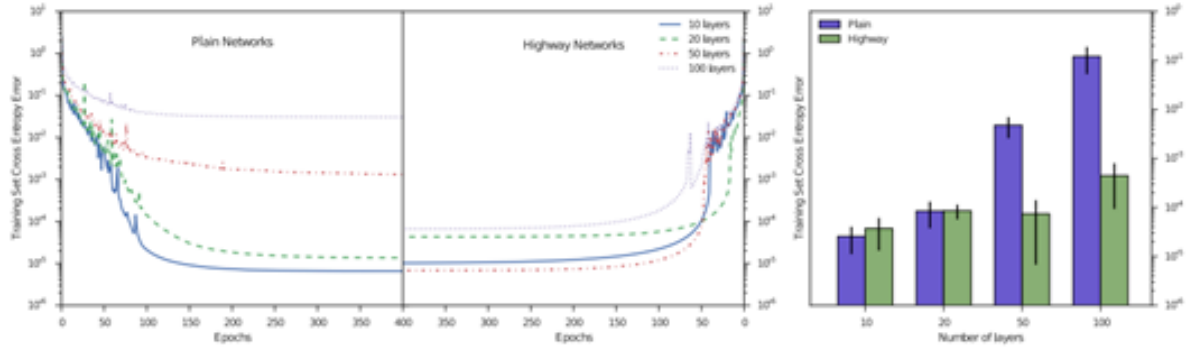


图3

算法在CIFAR数据集上的分类结果如图4所示。

Network	CIFAR-10 Accuracy (in %)	CIFAR-100 Accuracy (in %)
Maxout [20]	90.62	61.42
dasNet [36]	90.78	66.22
NiN [35]	91.19	64.32
DSN [24]	92.03	65.43
All-CNN [37]	**92.75**	66.29
Highway Network	92.40 (92.31±0.12)	**67.76 (67.61±0.15)**

图4

尽管在实验结果上，highway networks并没有比之前的一些模型取得显著地提升，但是它的这种思想对后面的模型改进影响非常大。

二、ResNet

ResNet [3]的动机依然是解决深度模型中的退化问题：层数越深，梯度越容易发散，误差越大，难以训练。理论上，模型层数越深，误差应该越小才对，因为我们总可以根据浅层模型的解构造出深层模型的解（将深层模型与浅层模型对应的层赋值为浅层模型的权重，将后面的层取为恒等映射），使得这个深层模型的误差不大于浅层模型的误差。但是实际上，深度模型的误差要比浅层模型的误差要大，在CIFAR-10上面的训练和测试误差如图5所示。

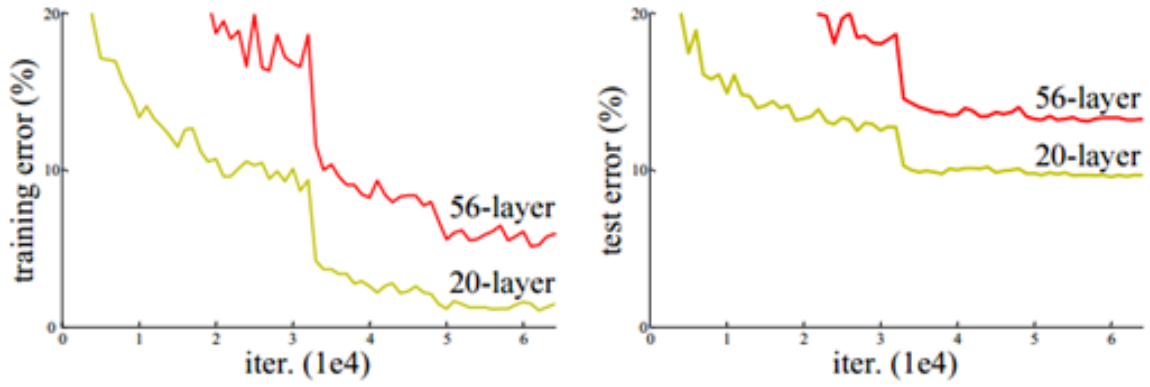


图5

作者认为产生这种现象的原因是深度模型难以优化，难以收敛到较优的解，并假设相比于直接优化最初的plain networks的模型F(x)=y，残差F(x)=y-x更容易优化。对于plain networks的模型，形式化地表示为图6（本质上与图1的结构类似，采用图6主要是为了与论文中的描述一致），F就是要优化的目标F(x)=y。

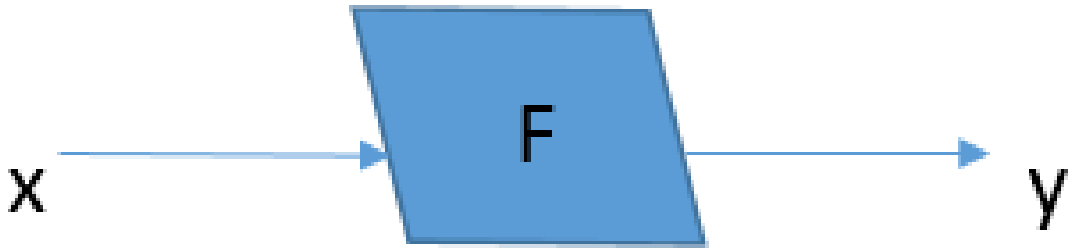


图6

而对于ResNet，形式化地表示为图7，优化的目标F为F(x)=y-x，即为残差。

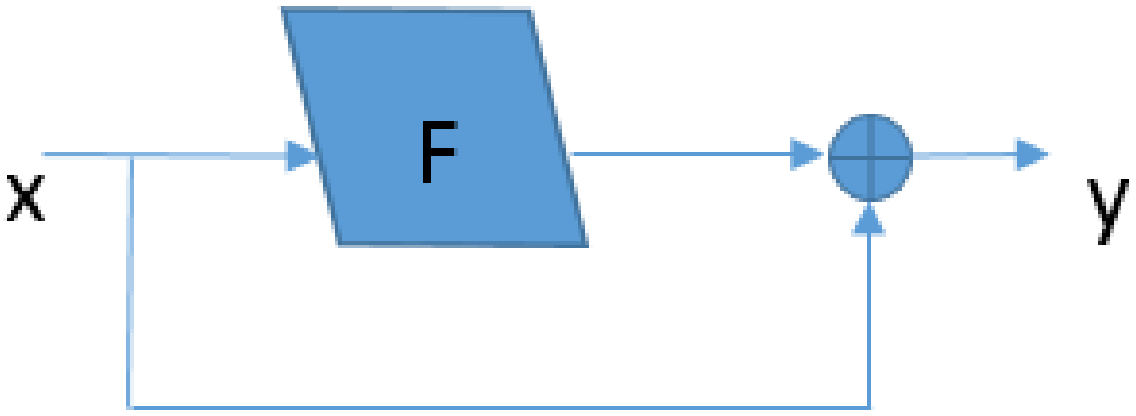


图7

需要注意的是，变换F可以是很多层，也就是说shortcut不一定只跨越1层。并且实际中，由于shortcut只跨越单层没有优势，ResNet中是跨越了2层或3层，如图8所示。ResNet-34中，采用图8左侧的shortcut跨越方式；ResNet-50/101/152采用图8右侧的shortcut跨越方式。

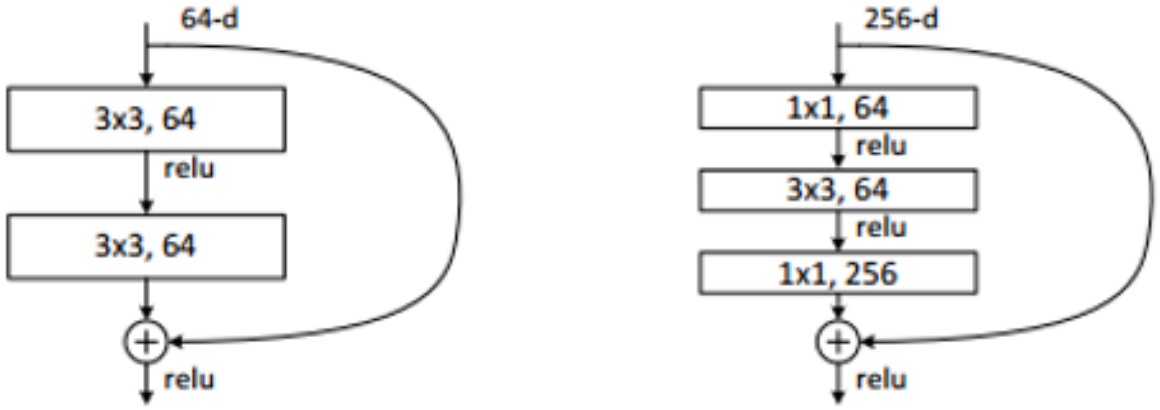
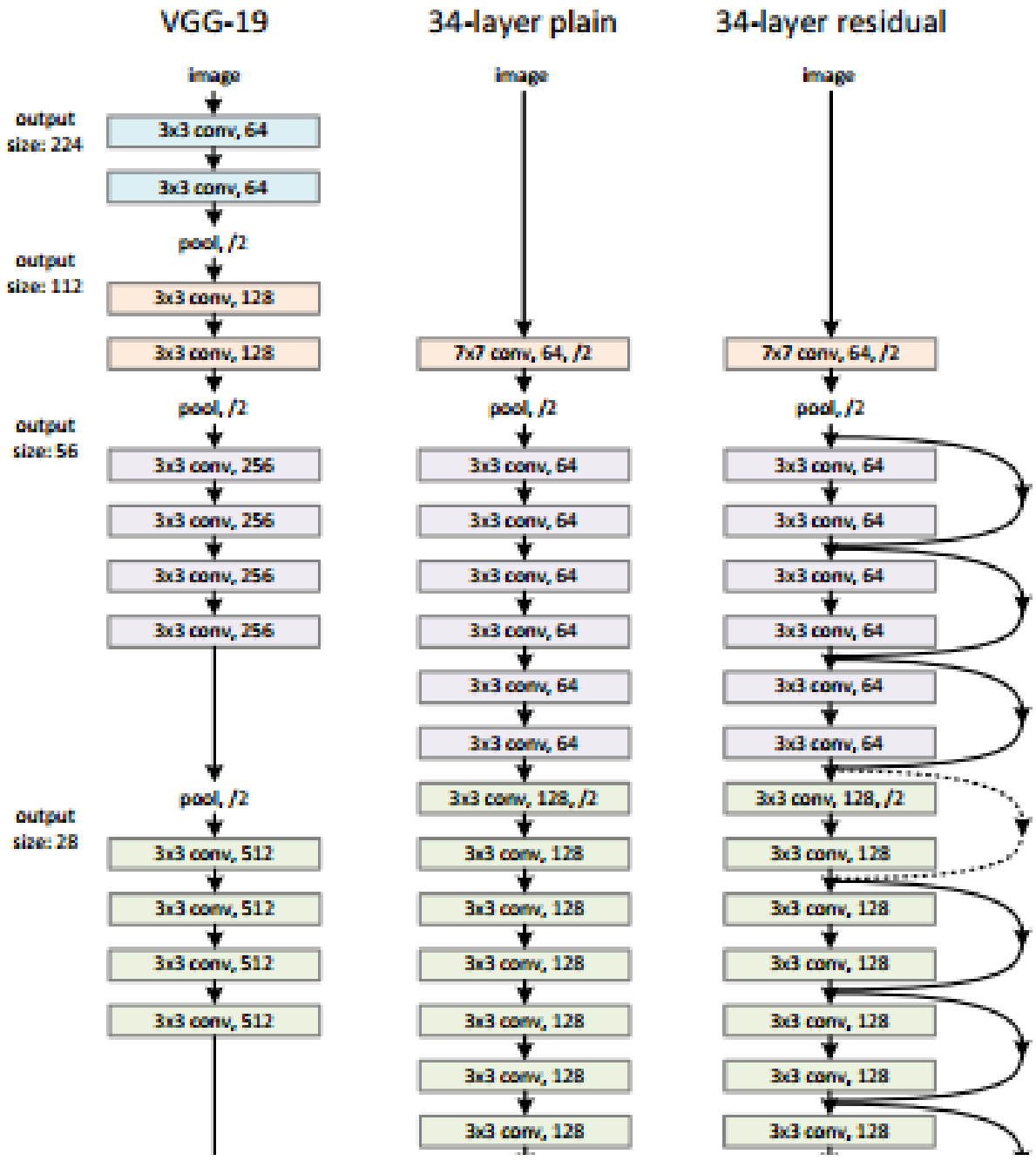


图8

ResNet-34与其他两种模型的对比如图9所示。



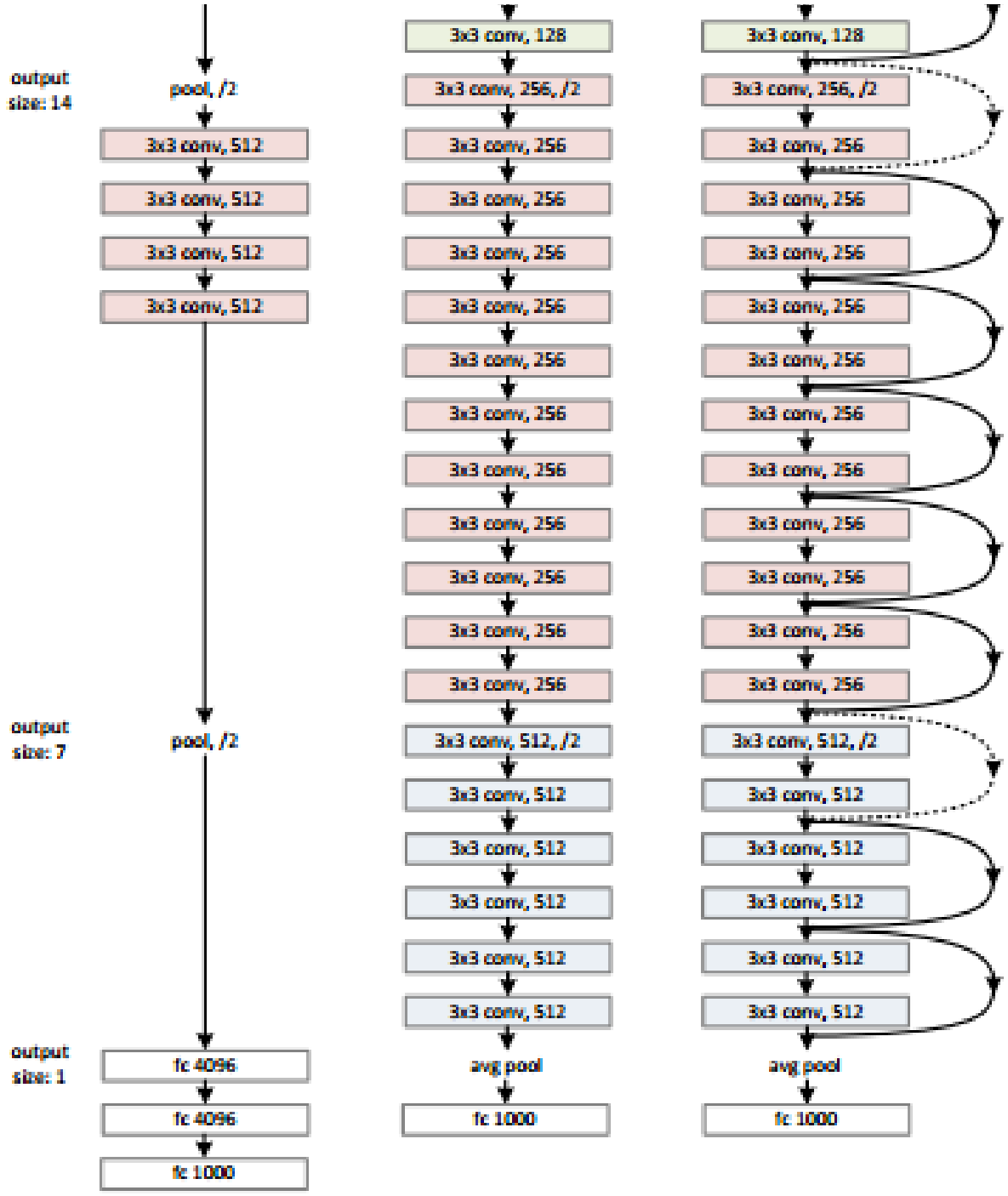


图9

经过改进之后，ResNet与plain networks在ImageNet上的训练误差对比如图10。对于plain networks，34层的模型误差要比18层的误差大，而对于ResNet，34层的模型误差要小于18层的误差。

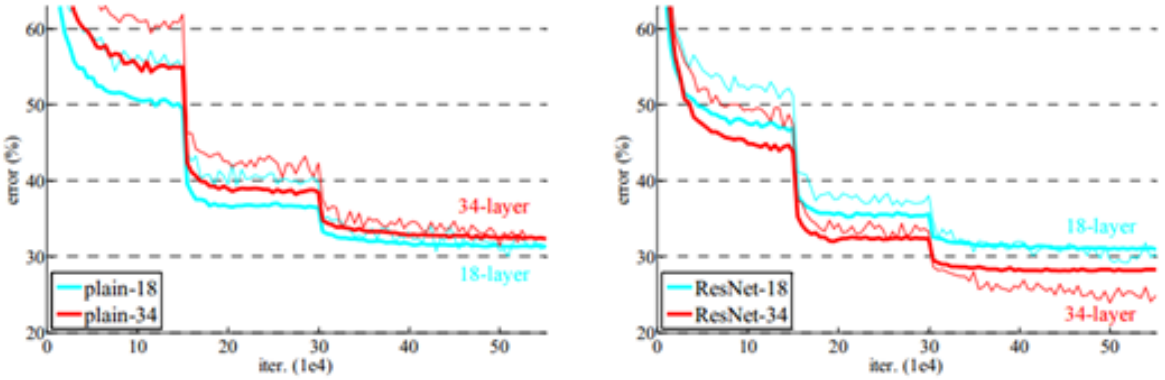


图10

在ImageNet和CIFAR-10上面的结果对比如图11所示。

method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43†
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	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**

Table 4. Error rates (%) of **single-model** results on the ImageNet validation set (except † reported on the test set).

method			error (%)
Maxout [10]			9.38
NIN [25]			8.81
DSN [24]			8.22
	# layers	# params	
FitNet [35]	19	2.5M	8.39
Highway [42, 43]	19	2.3M	7.54 (7.72±0.16)
Highway [42, 43]	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

Table 6. Classification error on the **CIFAR-10** test set. All methods are with data augmentation. For ResNet-110, we run it 5 times and show "best (mean±std)" as in [43].

图11

对比highway networks和ResNet，可以看到ResNet的改进主要在以下方面，

1，将highway networks的T和C都设为1，降低模型的自由度（深度模型中，自由度越大未必越好。自由度越大，训练会比较困难）。

2，shortcut不仅限于跨越1层，而可以跨越2层或3层。

三、DenseNet

DenseNet [4]的初衷依然是为了解决深度模型的退化问题——梯度发散，借鉴highway networks和ResNet的思路，DenseNet将shortcut用到了“极致”——每两层之间都添加shortcut，L层的网络共有L*(L-1)/2个shortcut（这样会不会太简单粗暴了？模型会不会太大？参数会不会太多？计算会不会太慢？放心，作者当然不会直接这么做）。通过shortcut可以直接将浅层的信息传递到深层，一方面可以解决退化问题，另一方面也可以看作是特征重用（feature reuse）。

首先来回顾一下highway networks和ResNet的连接单元，为了与文中表达式保持一致，又做了几幅丑图，见谅。对于plain networks，相邻两层之间有，

$$x_l = H_l(x_{l-1})$$

连接单元如图12所示，

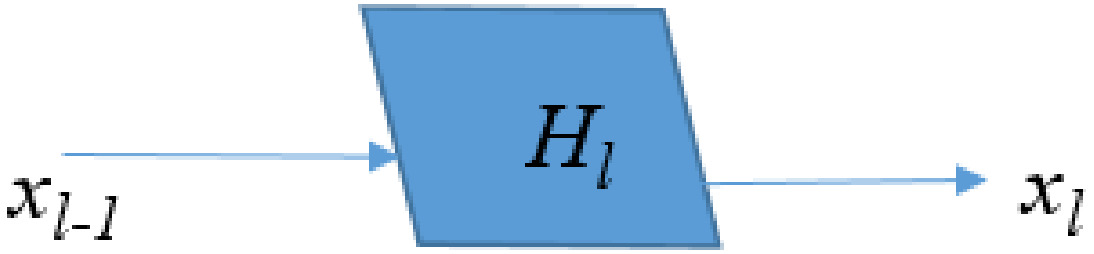


图12

对于ResNet，相邻两层之间有，

$$x_l = H_l(x_{l-1}) + x_{l-1}$$

连接单元如图13所示，

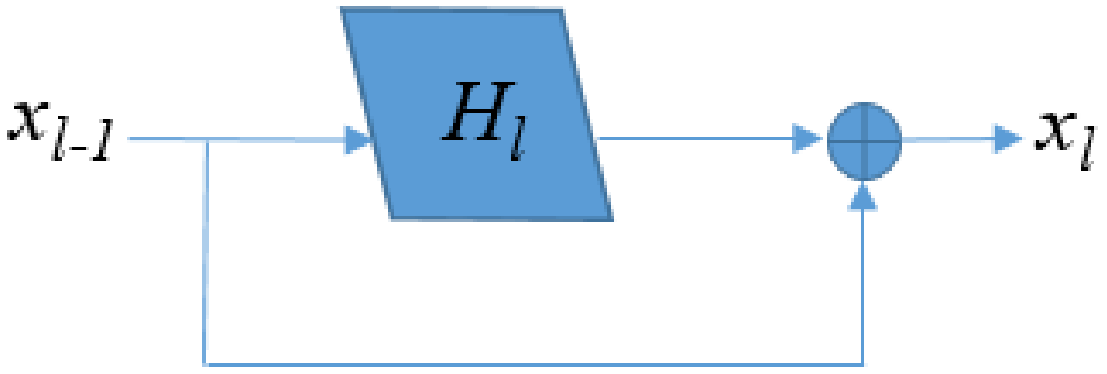


图13

而对于DenseNet，则有，

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

连接单元如图14所示，每层的输出结果都会通过shortcut连接到后面的层。

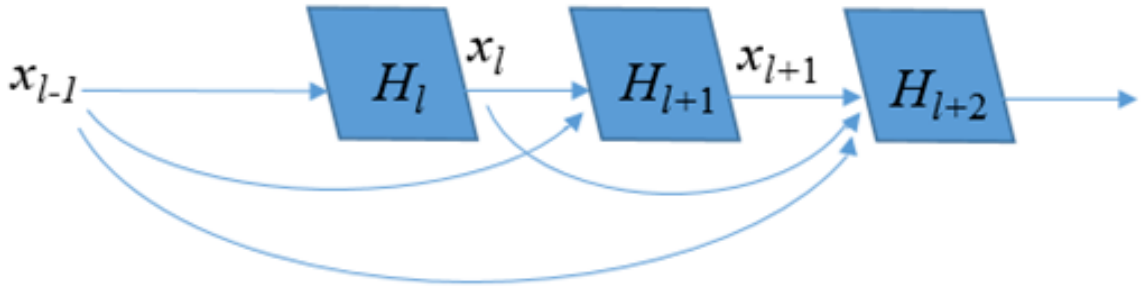


图14

如果真的每层的输出都稠密地连接到后面的所有层，那么模型将变得非常“宽”，计算将会很慢。因此，作者采用的是“局部”稠密连接，如图15所示，每个block里面才进行稠密连接。每个block里面的连接方式如图16所示，前面层的输出通过shortcut直接连接到block中后面的其他层。block之间通过transition层连接。

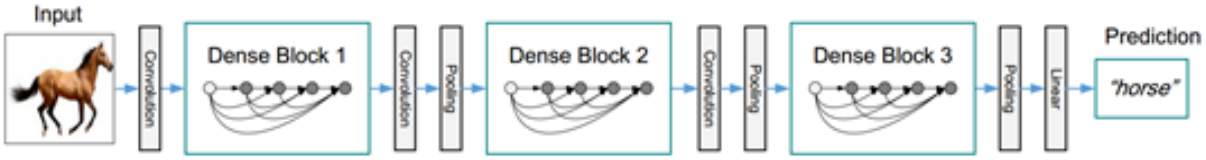


图15

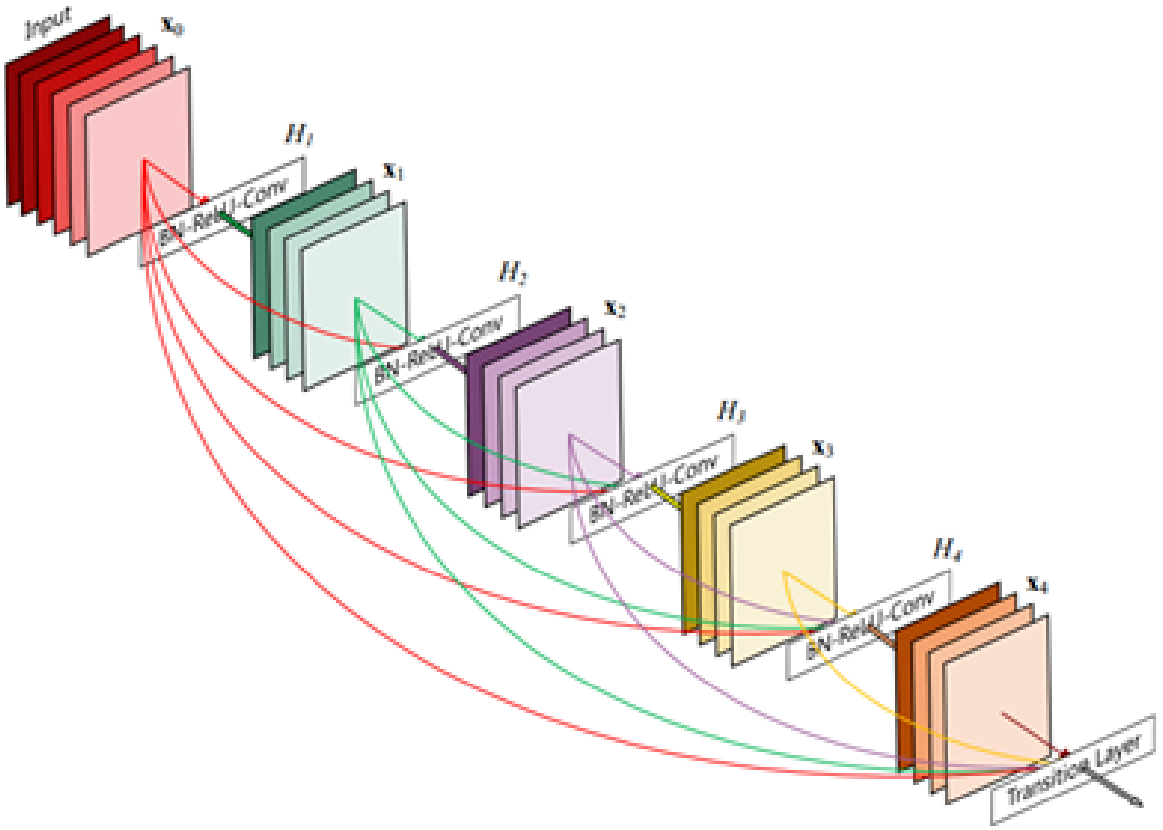


图16

对于一个包括t层的block，假设每层输出k个feature map（或通道），则第i（1 ≤i≤ t）层的输入feature map数为k*(i-1)+k0，其中k0为block的输入的通道数。将层分block只是限制了i的大小，如果每层的输出数k比较大的话，计算仍然很慢，因此作者也对k进行了限制，文中k称为growth rate。此外为了将模型进一步压缩，作者还采用了bottleneck layer和对transition的输出进行压缩（DenseNet-BC）。

在ImageNet任务上，不同层数的DenseNet的架构如图17所示，

Layers	Output Size	DenseNet-121($k=32$)	DenseNet-169($k=32$)	DenseNet-201($k=32$)	DenseNet-161($k=48$)
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] × 36
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] × 24
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

图17

相比ResNet，DenseNet的参数更少（主要是因为feature map少），计算更快。对比如图18所示，

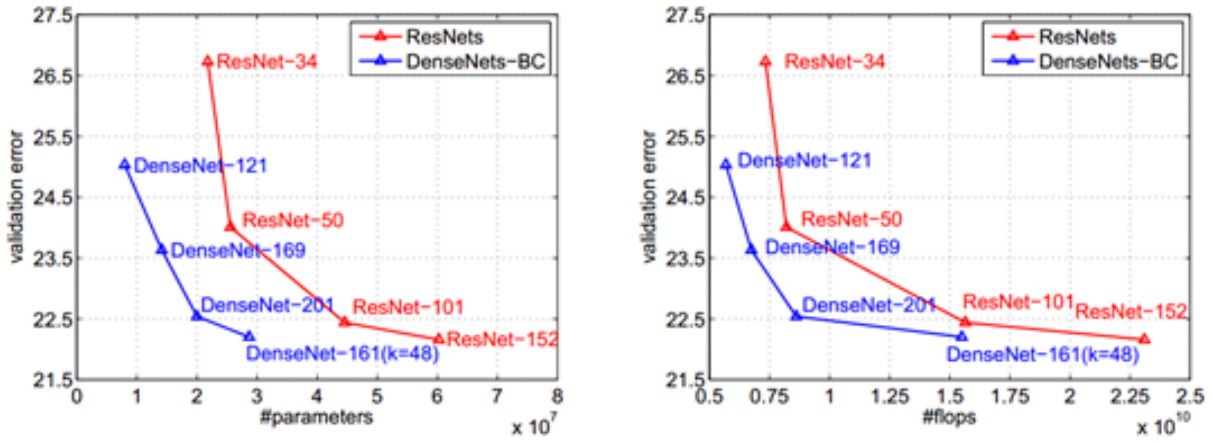


图18

DenseNet在CIFAR和SVHN数据集上的误差对比如图19所示，可以看出，DenseNet在模型大小和算法精度上都具有非常大的优势。从实用角度来讲，DenseNet获得CVPR2017 best paper也不足为奇。

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**	-

图19

对比highway networks和ResNet，可以看到DenseNet的改进主要在shortcut的使用上，将网络层进行稠密连接，shortcut可以跨越很多层并可以同时存在，通过将网络分为block和限制每层的输出通道数来减少参数和降低计算复杂度。

总结

为了解决深度模型中的梯度发散问题，很多技术方法被提了出来，shortcut是其中一种非常有效的方法。本文主要概述了shortcut使用的一些历程，希望通过本文能给其他技术方法的改进带来一丝启发。不足之处还请多多指正。谢谢！

参考文献：

1 ImageNet Classification with Deep Convolutional Neural Networks.

2 Training Very Deep Networks.

3 Deep Residual Learning for Image Recognition.

4 Densely Connected Convolutional Networks.