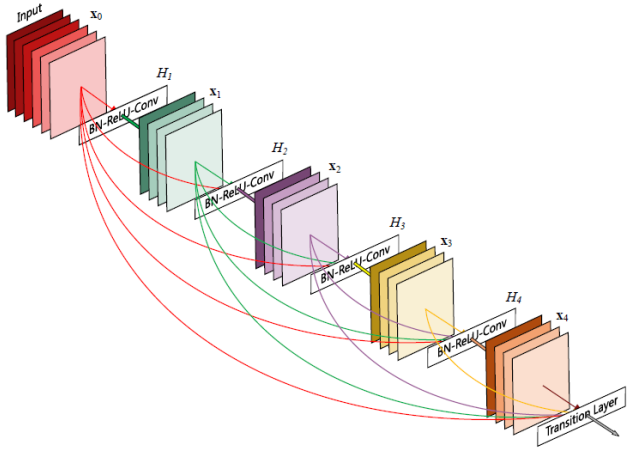
Densely Connected Convolutional Networks



2018.06.29
안건이

Content

- **Convolutional Neural Networks**
  - **History**
- **DenseNet**

01 History
LeNet-5

[LeCun et al., 1998]

- **Conv filters 5x5, stride 1**
- **Pooling 2x2, stride 2**
- **[Conv – Pool – Conv – Pool – FC – FC]**

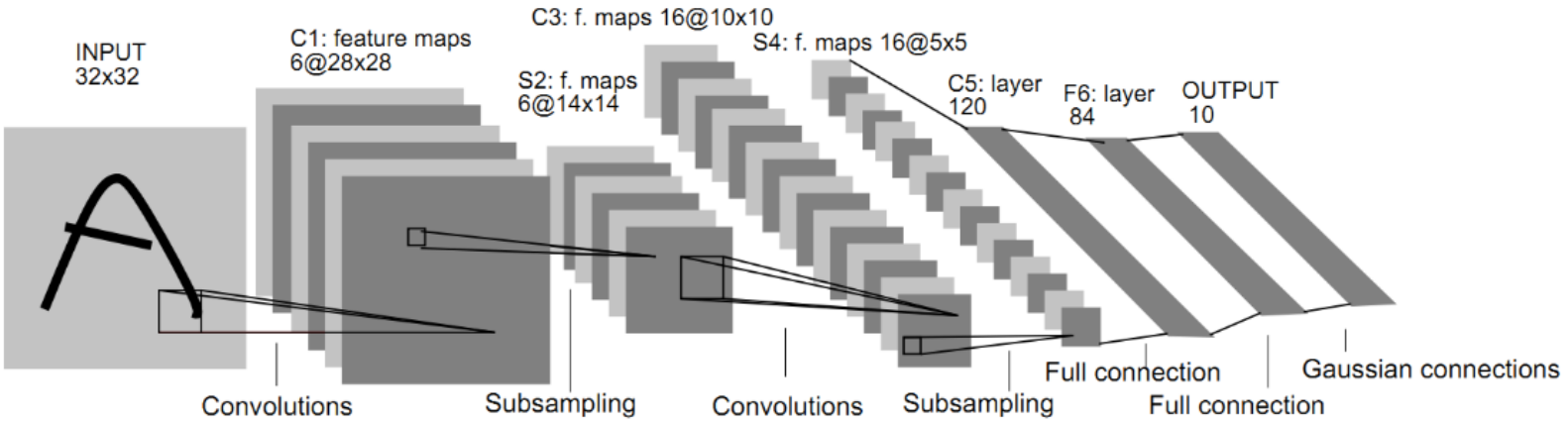
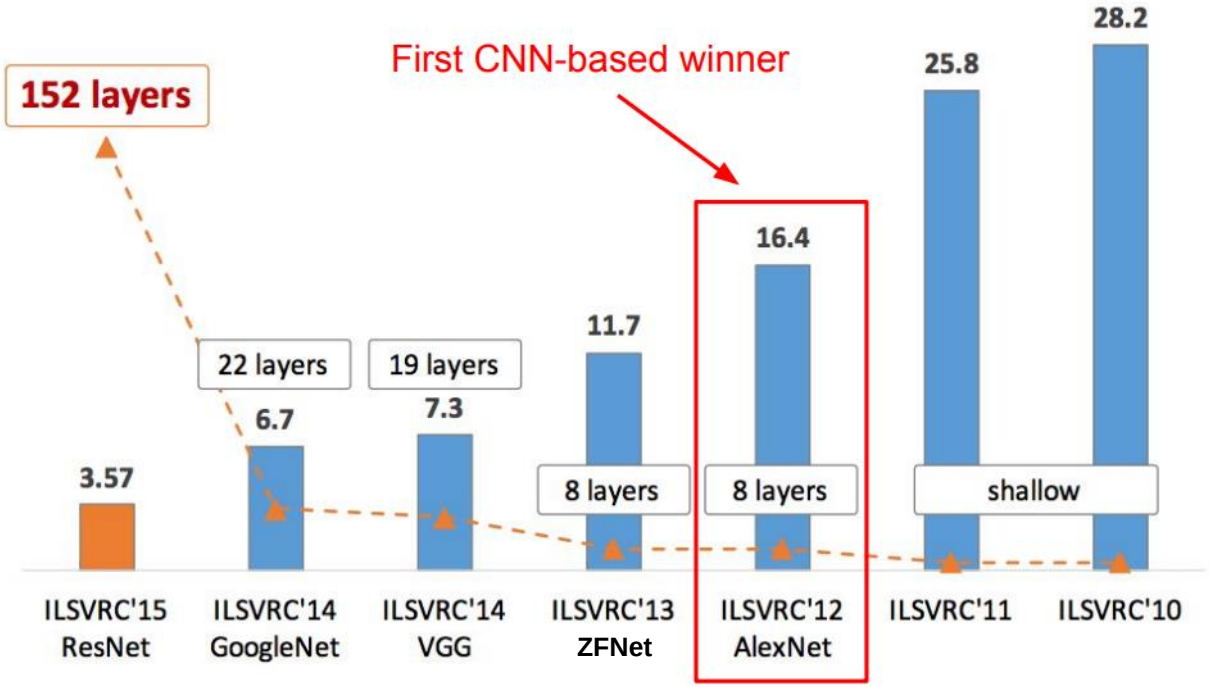


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Y.Lecun, L. Bottou, Y.Bengio, P.Haffner, “Gradient-based Learning Applied to Document Recognition”, IEEE, Volume 86 Issue 11 Nov, 1998

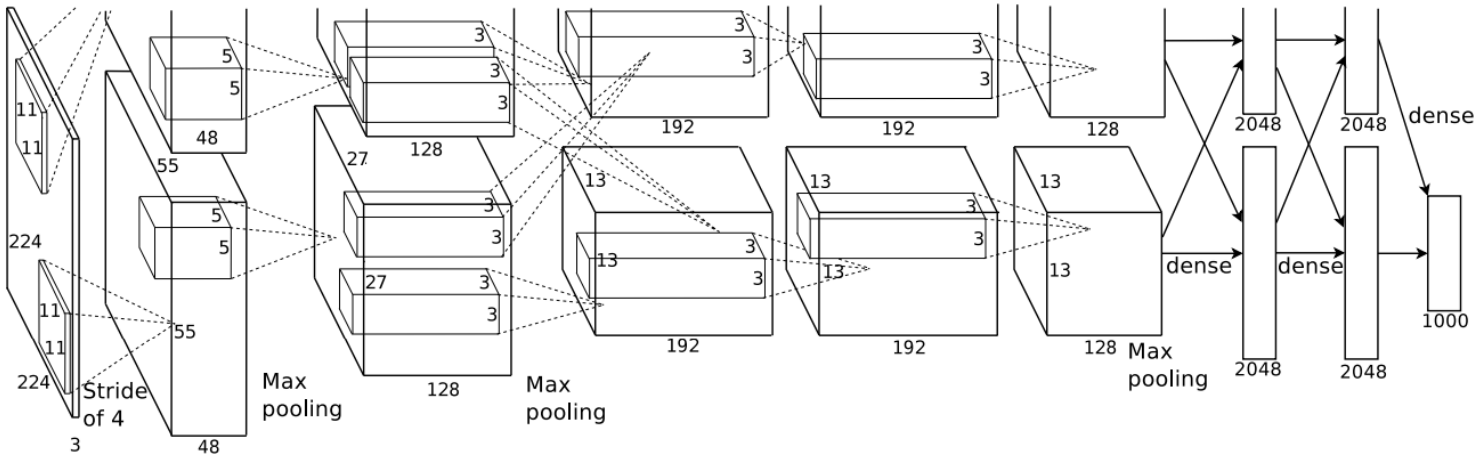
01 History
ILSVRC Winners

ImageNet Large Scale Visual Recognition Challenge(ILSVRC) winners



01 History
AlexNet

[Krizhevsky et al., 2012]

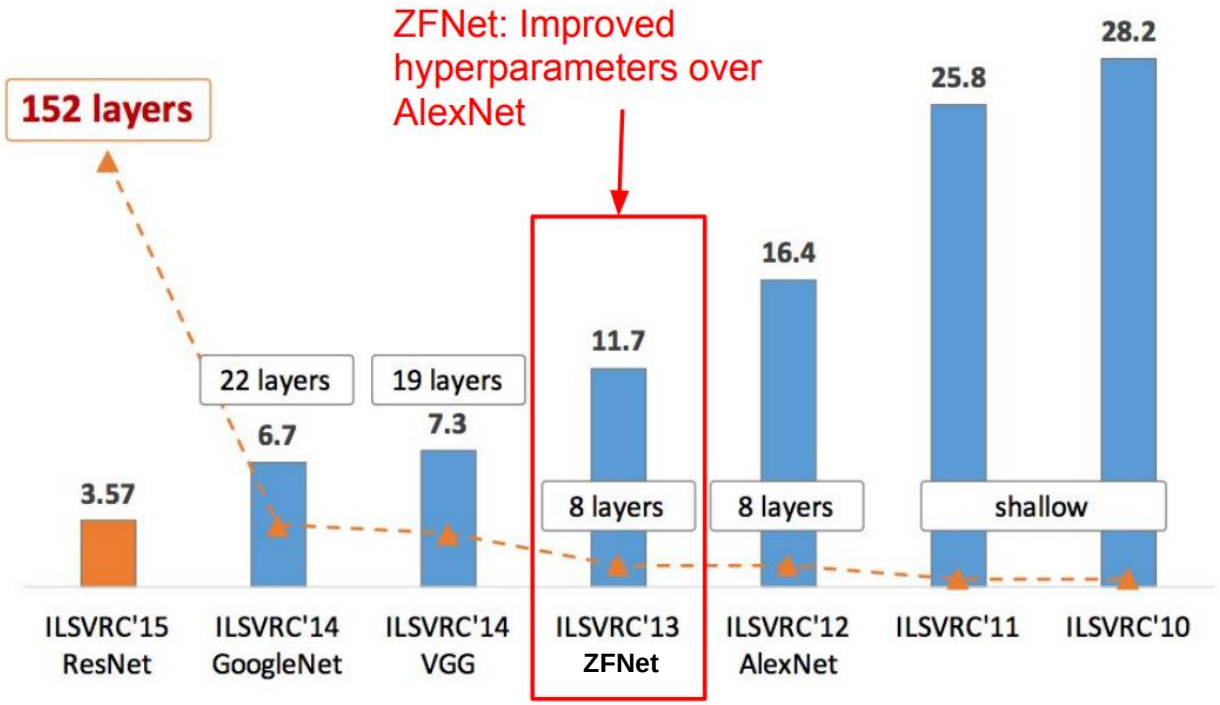


[227x227x3] INPUT
[55x55x96] Conv1: 96 11x11 filters at stride 4, pad 0
[27x27x96] MAX POOL1: 3x3 filters at stride 2
[27x27x96] NORM1: Normalization Layer
[27x27x256] Conv2: 256 5x5 filters at stride 1, pad 2
[13x13x256] MAX POOL2: 3x3 filters at stride 2
[13x13x256] NORM2: Normalization Layer
[13x13x384] Conv3: 384 3x3 filters at stride 1, pad 1
[13x13x384] Conv4: 384 3x3 filters at stride 1, pad 1
[13x13x256] Conv5: 256 3x3 filters at stride 1, pad 1
[6 x 6 x256] MAX POOL3: 3x3 filters at stride 2
[4096] FC6: 4096 neurons
[4096] FC7: 4096 neurons
[1000] FC8: 1000 neurons (class scores)

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey. ImageNet classification with deep convolutional neural networks. In NIPS. 2012.

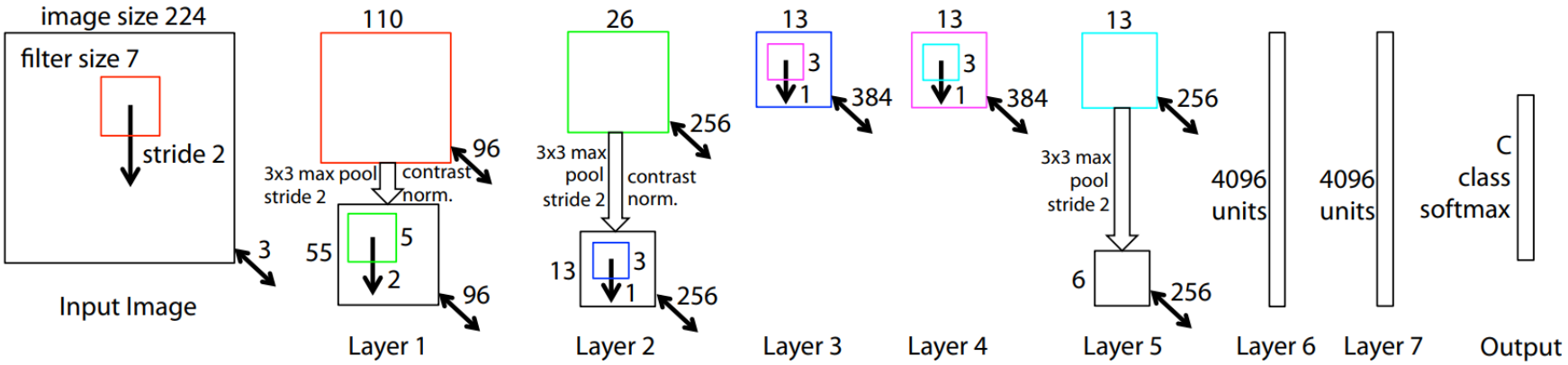
01 History
ILSVRC Winners

ImageNet Large Scale Visual Recognition Challenge(ILSVRC) winners



01 History
ZFNet

[Zeiler and Fergus, 2013]



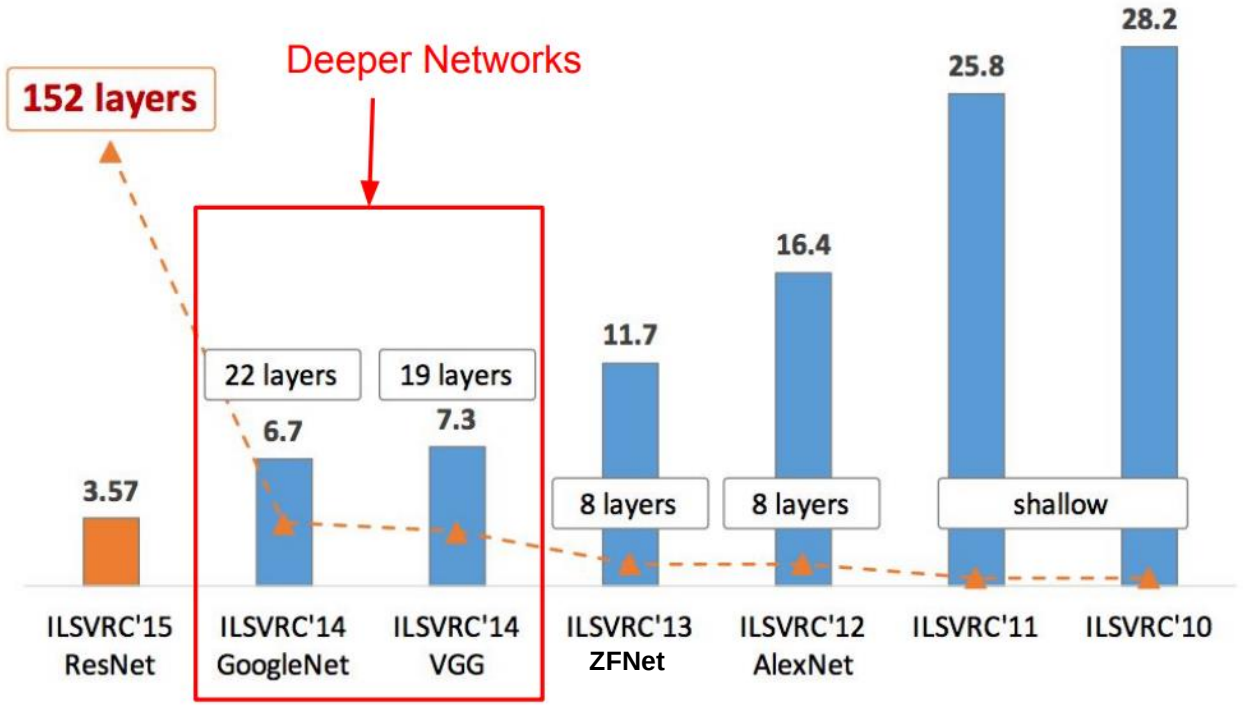
AlexNet but:
Conv1: change from (11x11 stride 4) to (7x7 stride 2)
Conv3,4,5: instead of 384, 384, 256 filters use 512, 1024, 512

ImageNet top 5 error: 16.4% → 11.7%

Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014).

01 History
ILSVRC Winners

ImageNet Large Scale Visual Recognition Challenge(ILSVRC) winners

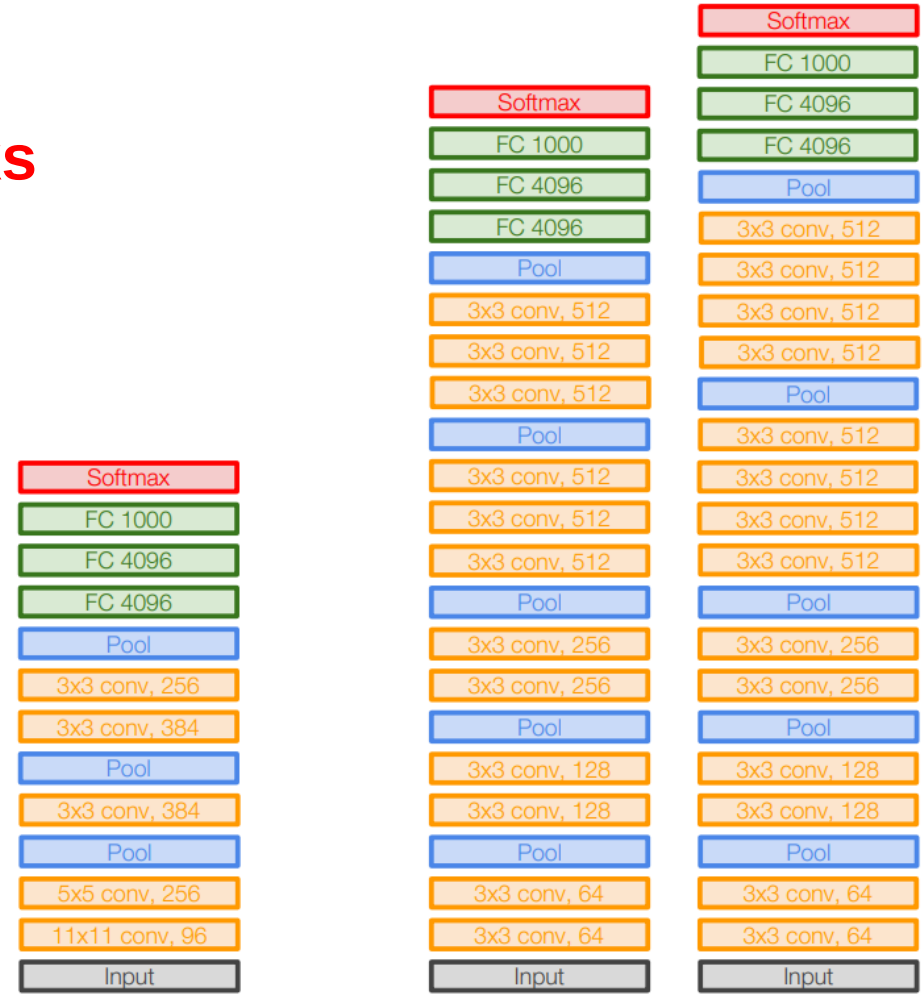


[Simonyan and Zisserman, 2014]

Small Filters, Deeper networks

AlexNet: 8 layers
VGGNet: 16-19 layers

Only 3x3 Conv stride 1, pad 1
2x2 MAX POOL stride 2



ImageNet top 5 error: 11.7% → 7.3%

Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2014)

[Simonyan and Zisserman, 2014]

Small Filters, Deeper networks

AlexNet: 8 layers
VGGNet: 16-19 layers

Only 3x3 Conv stride 1, pad 1
2x2 MAX POOL stride 2

-Stack of three 3x3 conv
# of param: 3 $*$ $(3^2 * C^2)$ = 29 $* C^2$

-Stack of one 7x7 conv
# of param: $7^2 * C^2$ = 49 $* C^2$

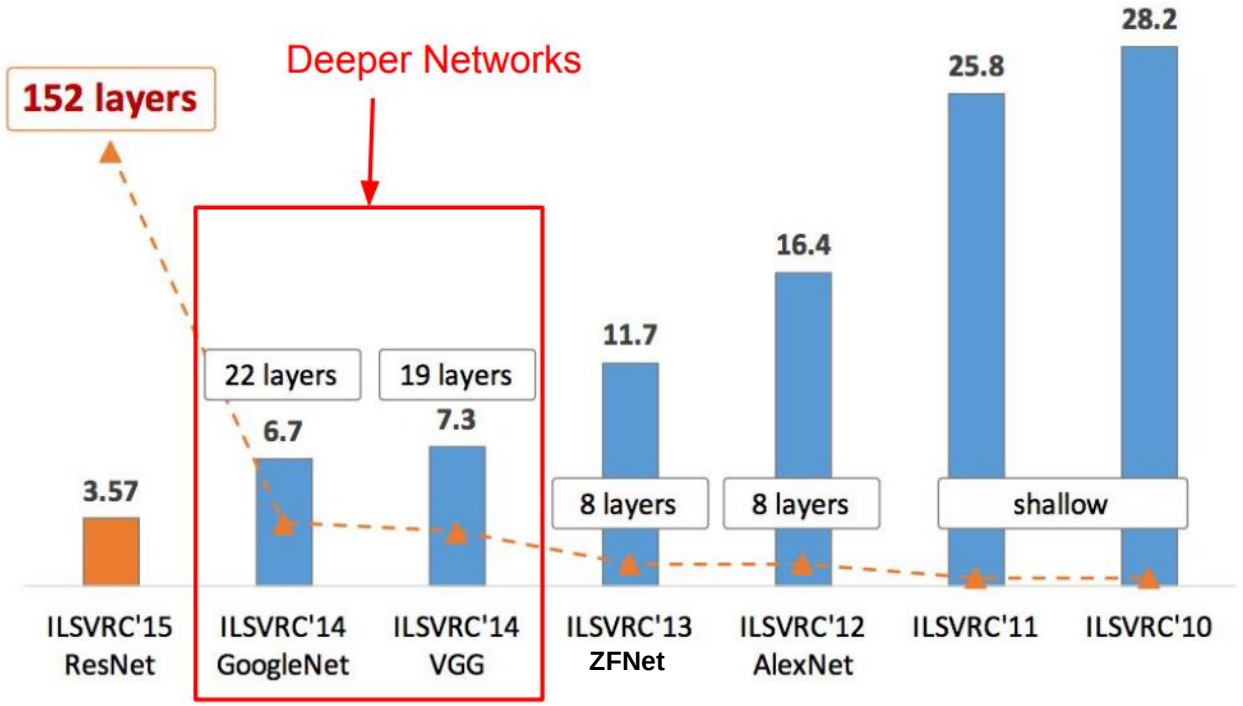
C: # of channel

ImageNet top 5 error: 11.7% → 7.3%

Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2014)

01 History
ILSVRC Winners

ImageNet Large Scale Visual Recognition Challenge(ILSVRC) winners

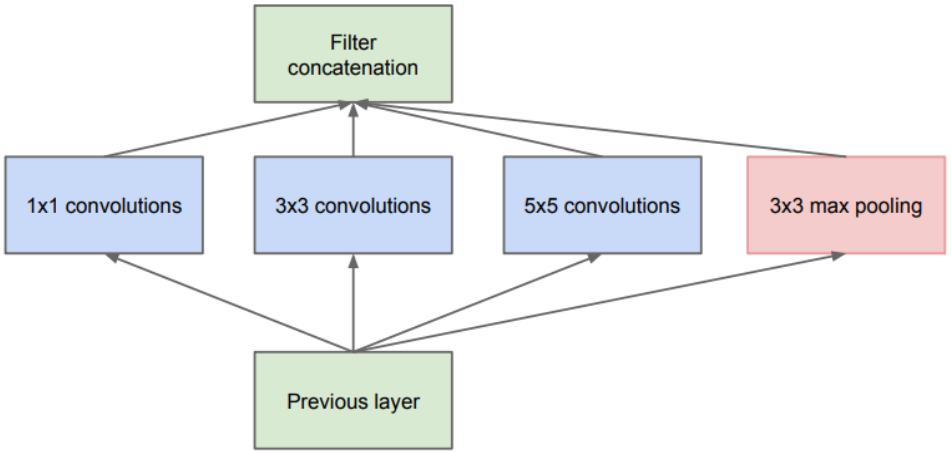


GoogLeNet

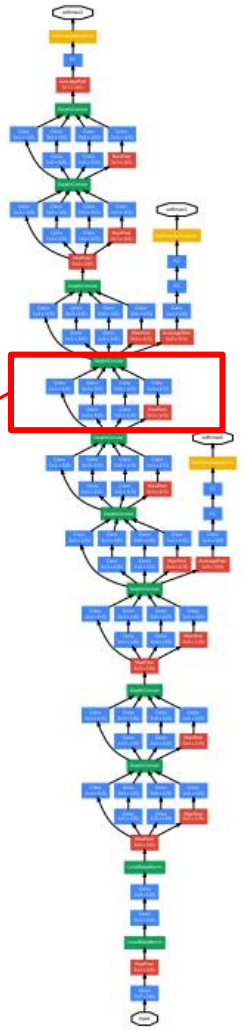
[Szegedy et al., 2014]

Deeper networks, with computational efficiency

VGGNet: 16-19 layers
GoogLeNet: 22 layers



Inception module

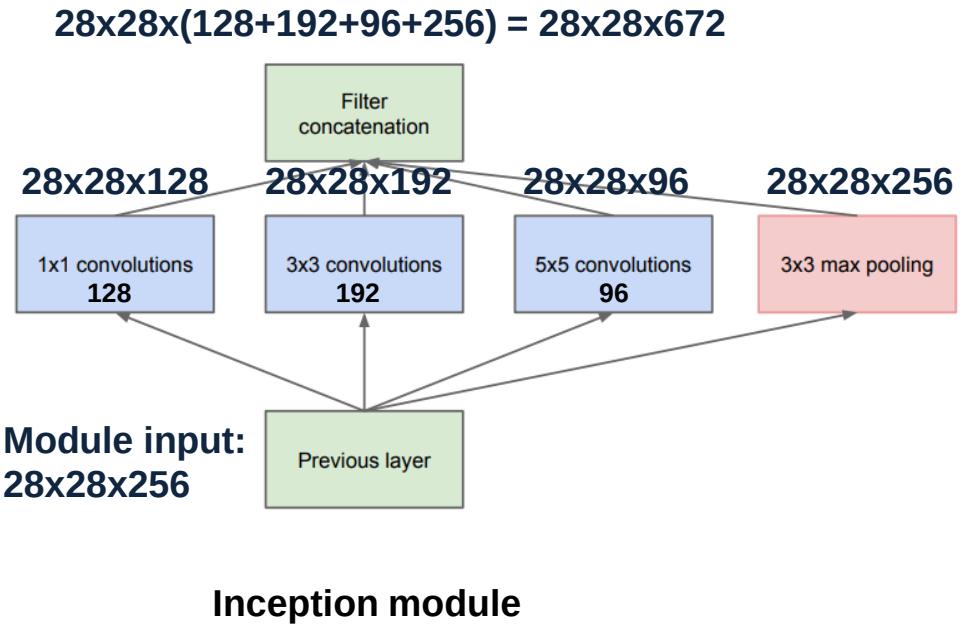


Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich,A. Going deeper with convolutions. CoRR, abs/1409.4842, 2014

GoogLeNet

[Szegedy et al., 2014]

Deeper networks, with computational efficiency



Inception module

Conv Ops:
[1x1 conv, 128] (28x28x128)x(1x1)x256
[3x3 conv, 192] (28x28x192)x(3x3)x256
[5x5 conv, 96] (28x28x96)x(5x5)x256

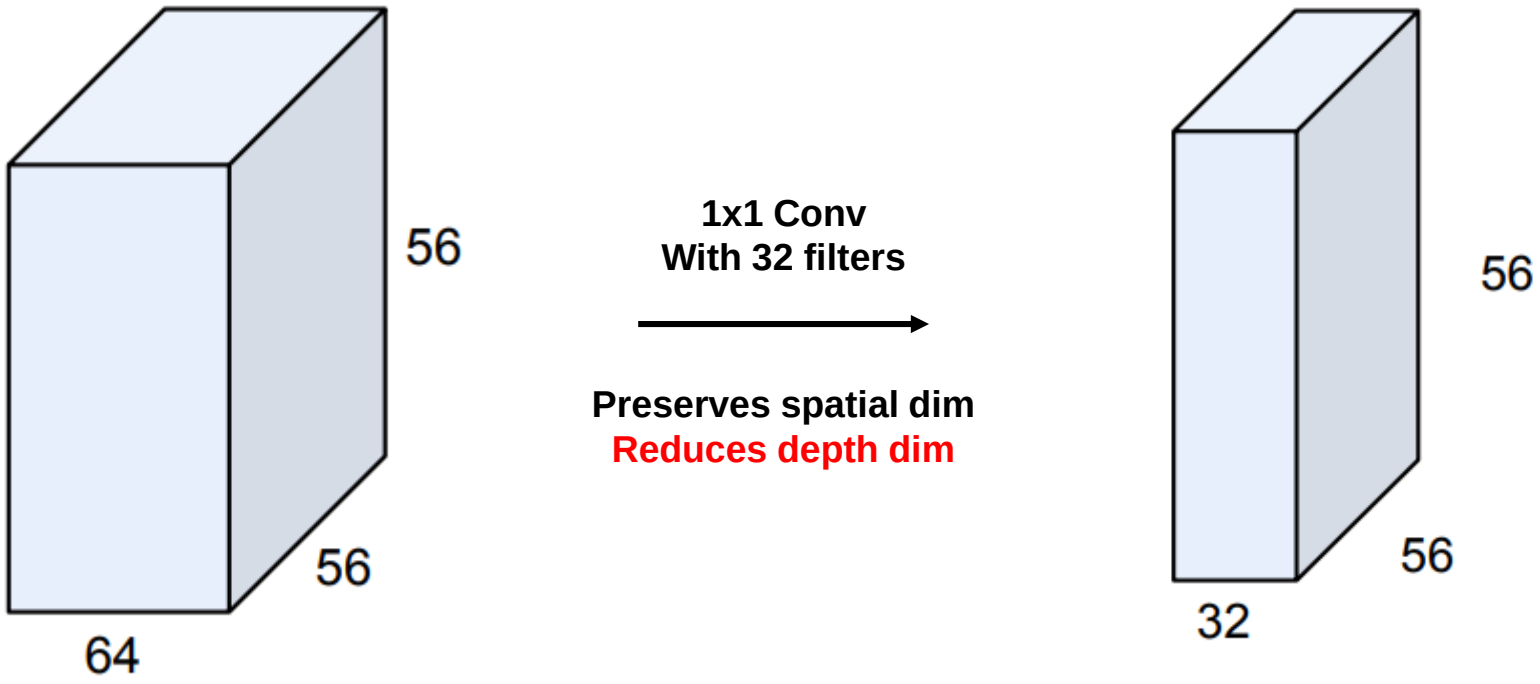
Total: 854M ops

Very expensive compute !!!

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich,A. Going deeper with convolutions. CoRR, abs/1409.4842, 2014

[Szegedy et al., 2014]

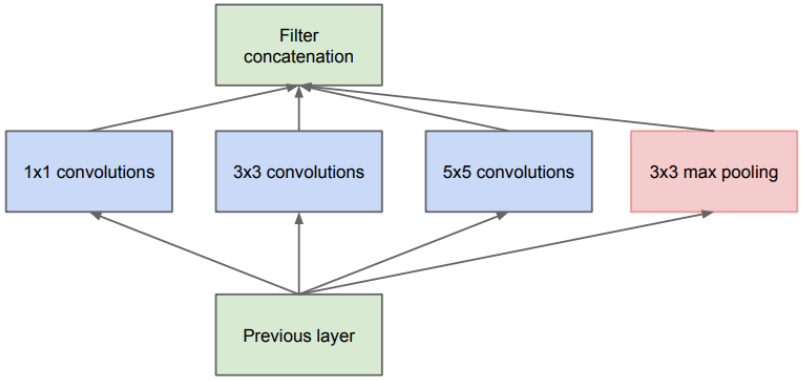
1x1 conv “bottleneck” layers



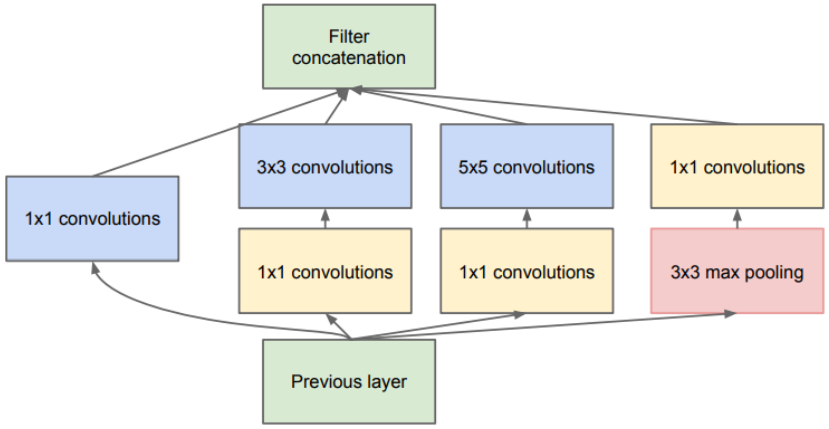
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich,A. Going deeper with convolutions. CoRR, abs/1409.4842, 2014

01 History
GoogLeNet

[Szegedy et al., 2014]



Inception module

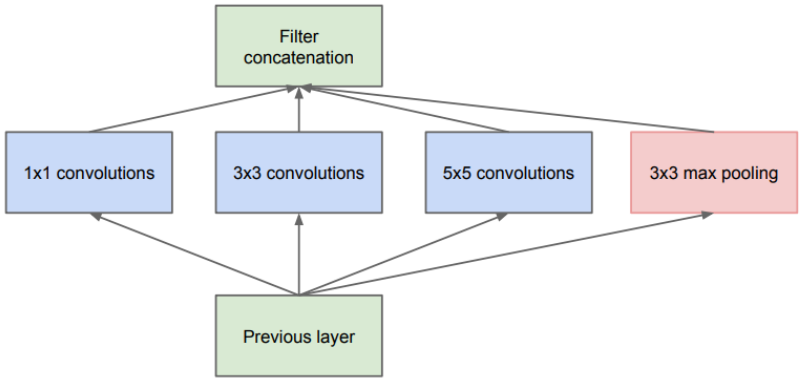


Inception module with dimension reduction

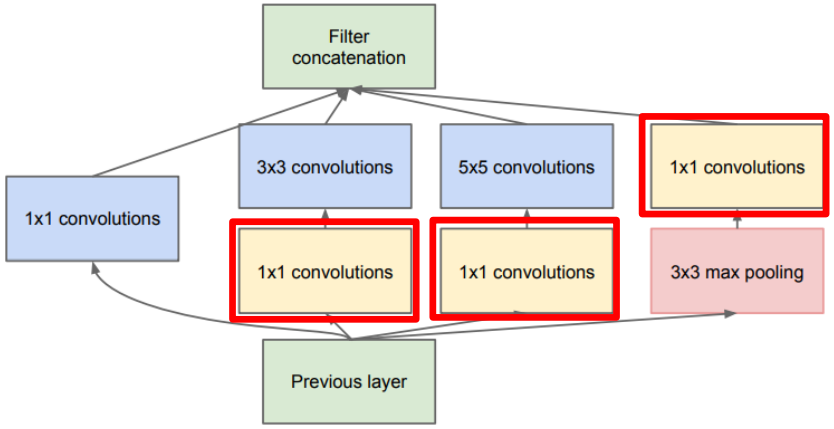
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich,A. Going deeper with convolutions. CoRR, abs/1409.4842, 2014

[Szegedy et al., 2014]

1x1 conv "bottleneck" layers



Inception module



Inception module with dimension reduction

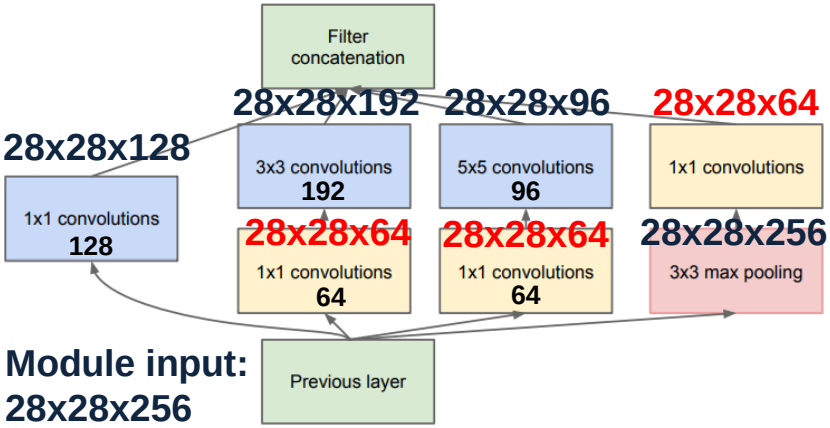
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich,A. Going deeper with convolutions. CoRR, abs/1409.4842, 2014

GoogLeNet

[Szegedy et al., 2014]

64, 1x1 conv "bottleneck" layers

28x28x(128+192+96+64) = 28x28x480



Conv Ops:

[1x1 conv, 64] (28x28x64)x(1x1)x256
[1x1 conv, 64] (28x28x64)x(1x1)x256
[1x1 conv, 128] (28x28x128)x(1x1)x256
[3x3 conv, 192] (28x28x192)x(3x3)x64
[5x5 conv, 96] (28x28x96)x(5x5)x64
[1x1 conv, 64] (28x28x64)x(5x5)x256

Total: 358M ops

854M → 358M

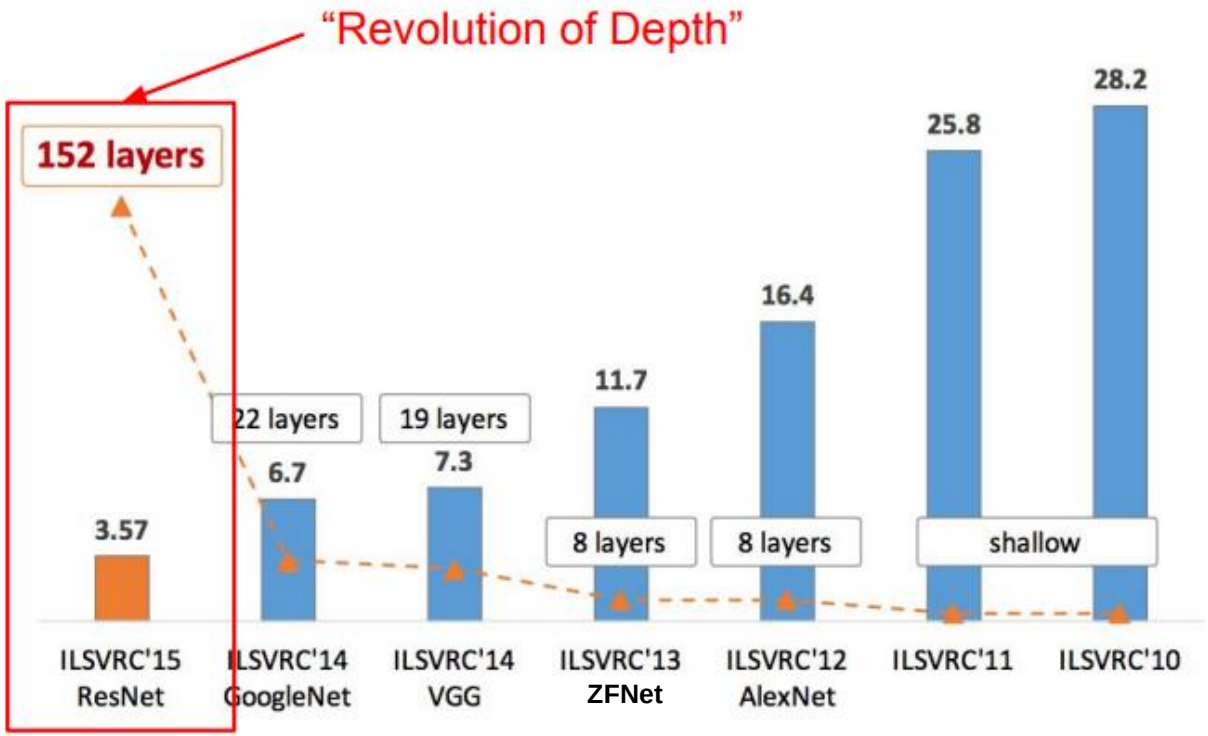
Inception module with dimension reduction

ImageNet top 5 error: 7.3% → 6.7%

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich,A. Going deeper with convolutions. CoRR, abs/1409.4842, 2014

01 History
ILSVRC Winners

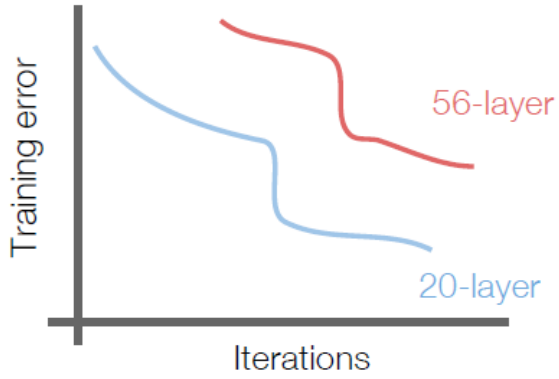
ImageNet Large Scale Visual Recognition Challenge(ILSVRC) winners

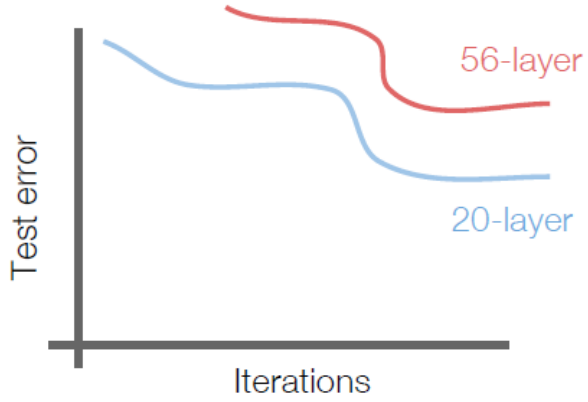


01 History ResNet

[He et al., 2015]

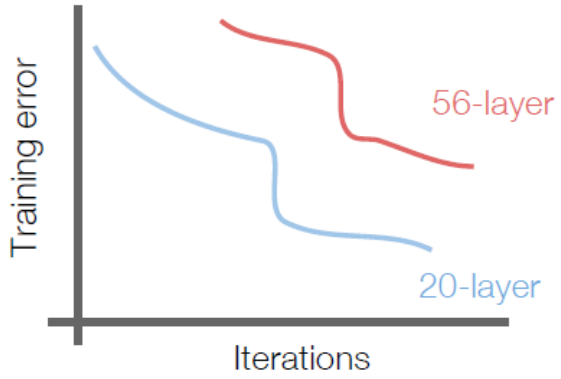
56-layer model performs worse on both training and test error
The deeper model performs worse, but it's not caused by overfitting

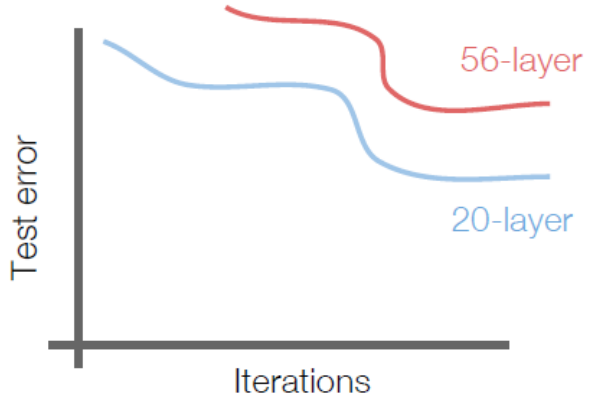


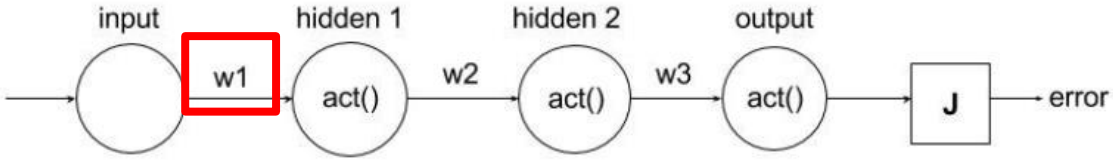


[He et al., 2015]

56-layer model performs worse on both training and test error
The deeper model performs worse, but it's not caused by overfitting







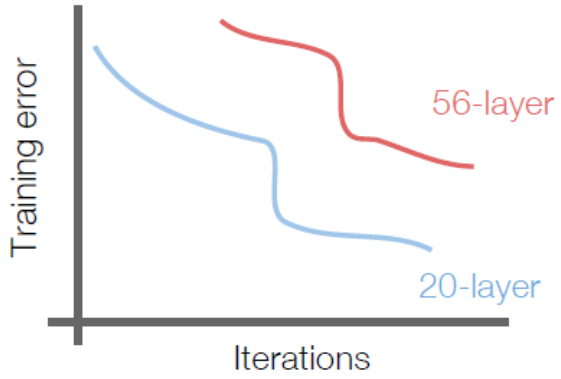
$$\frac{\partial output}{\partial hidden2}\frac{\partial hidden2}{\partial hidden1} = \frac{\partial Sigmoid(z_1)}{\partial z_1}^{<1/4} w3^{<1} * \frac{\partial Sigmoid(z_2)}{\partial z_2}^{<1/4} w2^{<1}$$

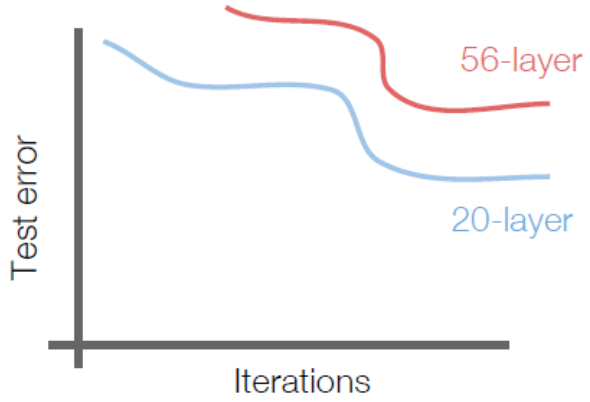
0.02625 = 0.25 x 0.6 x 0.25 x 0.7

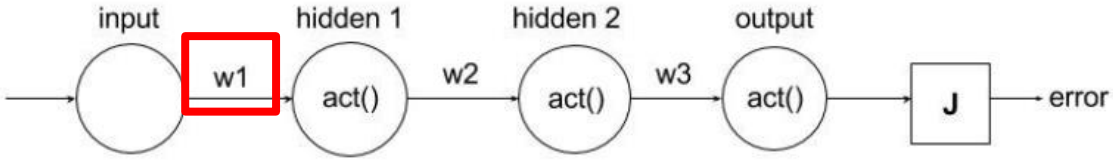
01 History ResNet

[He et al., 2015]

56-layer model performs worse on both training and test error
The deeper model performs worse, but it's not caused by overfitting







Caused by Gradient vanishing !!!

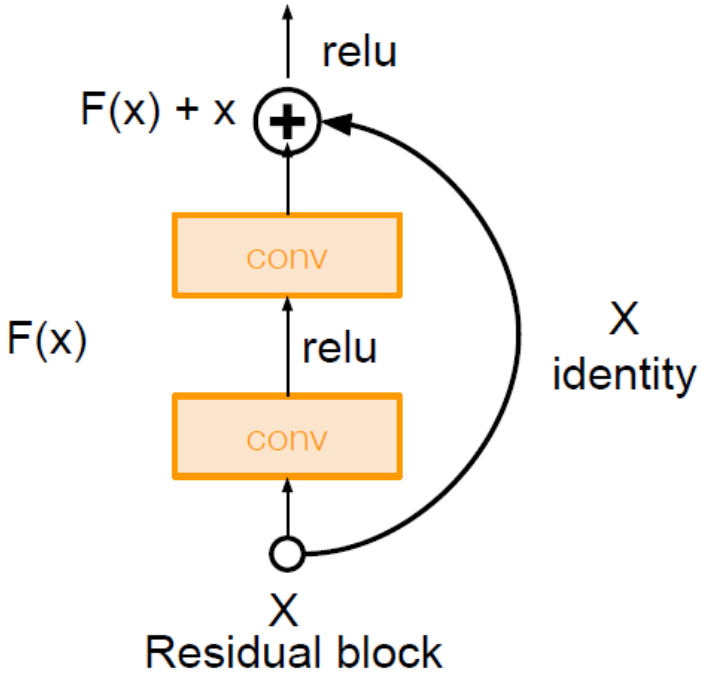
$$\frac{\partial output}{\partial hidden2}\frac{\partial hidden2}{\partial hidden1} = \frac{\partial Sigmoid(z_1)}{\partial z_1}^{<1/4} w3^{<1} * \frac{\partial Sigmoid(z_2)}{\partial z_2}^{<1/4} w2^{<1}$$

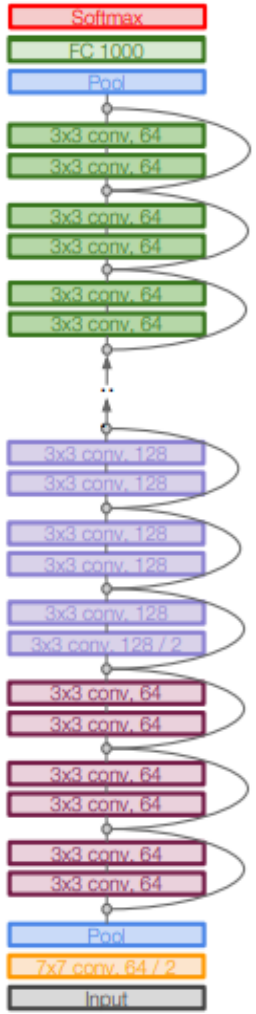
0.02625 = 0.25 x 0.6 x 0.25 x 0.7

[He et al., 2015]

Very deep networks using residual connections

GoogLeNet: 22 layers
ResNet: 152 layers



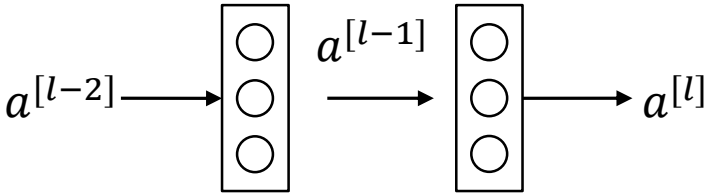


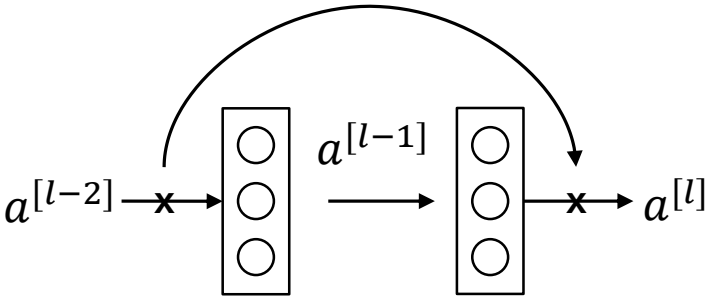
He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385

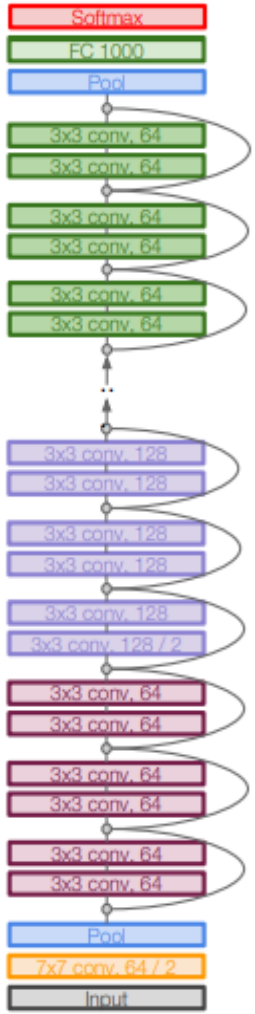
[He et al., 2015]

Very deep networks using residual connections

GoogLeNet: 22 layers
ResNet: 152 layers





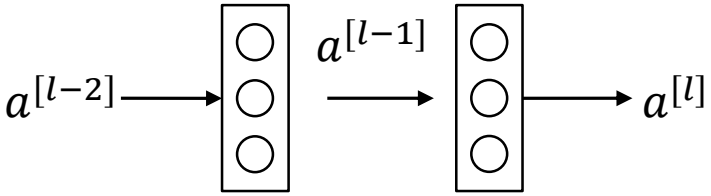


He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385

[He et al., 2015]

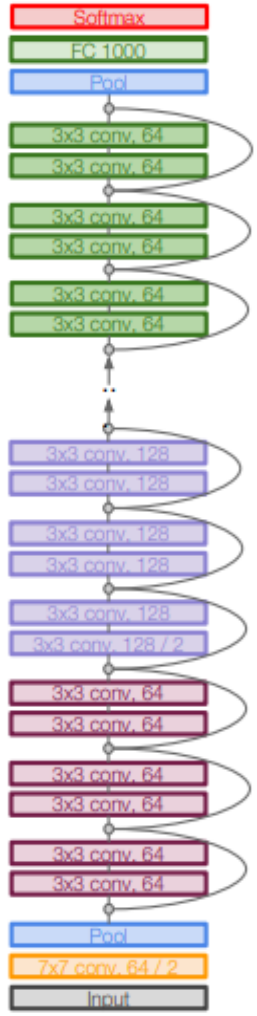
Very deep networks using residual connections

GoogLeNet: 22 layers
ResNet: 152 layers



$a^{[l-2]}$ → **Linear** → **ReLU** → $a^{[l-1]}$ → **Linear** → **ReLU** → $a^{[l]}$

$$z^{[l-1]} = W^{[l-1]}a^{[l-2]} + b^{[l-1]}$$
$$a^{[l-1]} = g(z^{[l-1]})$$
$$z^{[l]} = W^{[l]}a^{[l-1]} + b^{[l]}$$
$$a^{[l]} = g(z^{[l]})$$

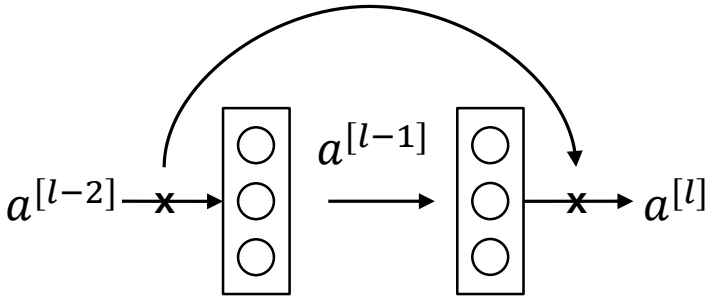


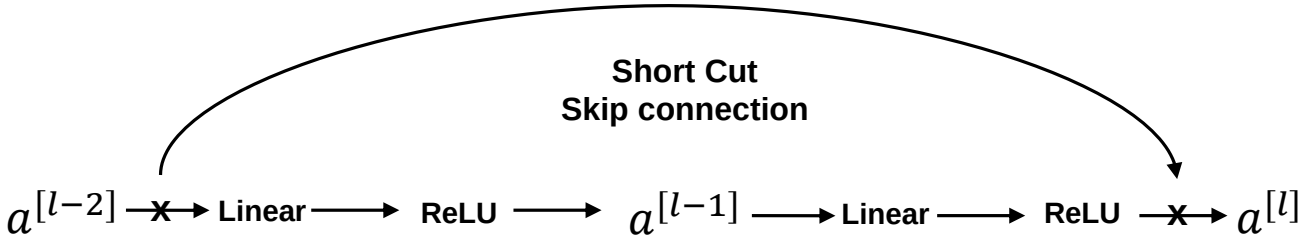
He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385

[He et al., 2015]

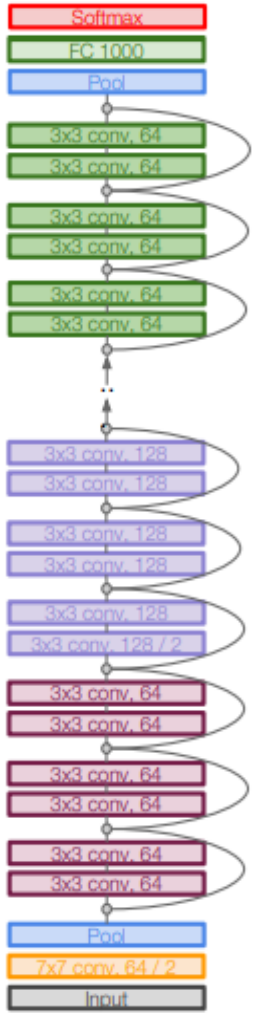
Very deep networks using residual connections

GoogLeNet: 22 layers
ResNet: 152 layers





$$z^{[l-1]} = W^{[l-1]}a^{[l-2]} + b^{[l-1]}$$
$$a^{[l-1]} = g(z^{[l-1]})$$
$$z^{[l]} = W^{[l]}a^{[l-1]} + b^{[l]}$$
$$a^{[l]} = g(z^{[l]}) + a^{[l-2]}$$

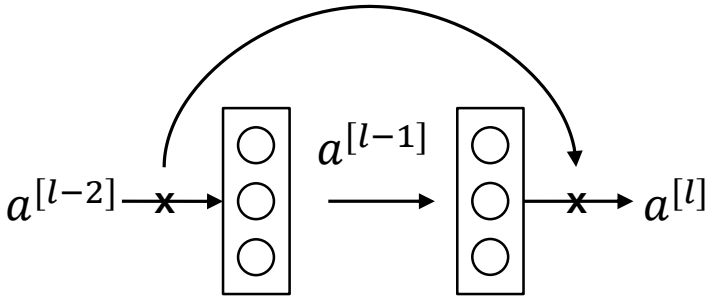


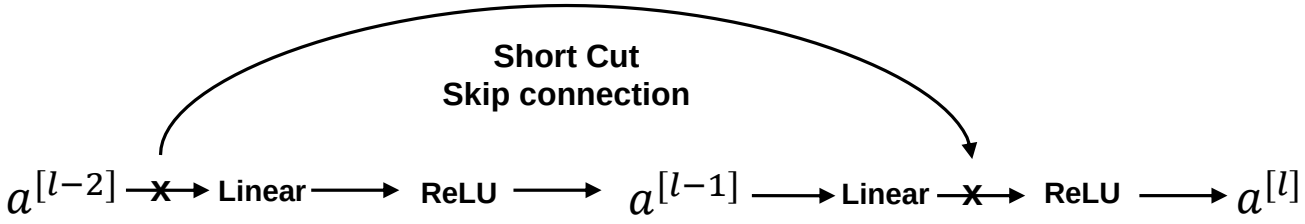
He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385

[He et al., 2015]

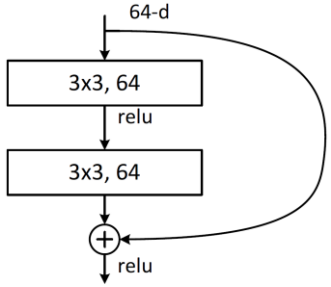
Very deep networks using residual connections

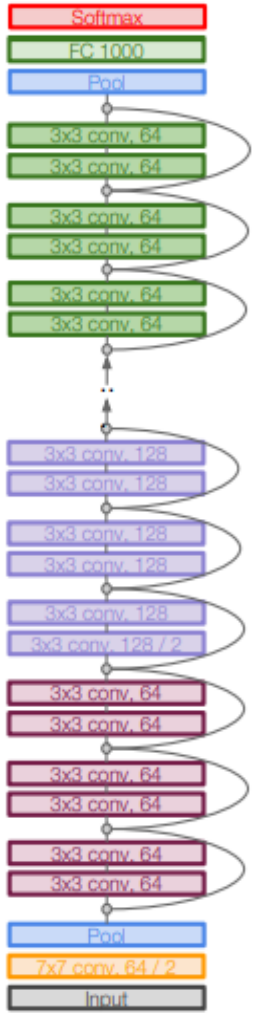
GoogLeNet: 22 layers
ResNet: 152 layers





$$z^{[l-1]} = W^{[l-1]}a^{[l-2]} + b^{[l-1]}$$
$$a^{[l-1]} = g(z^{[l-1]})$$
$$z^{[l]} = W^{[l]}a^{[l-1]} + b^{[l]}$$
$$a^{[l]} = g(z^{[l]} + a^{[l-2]})$$



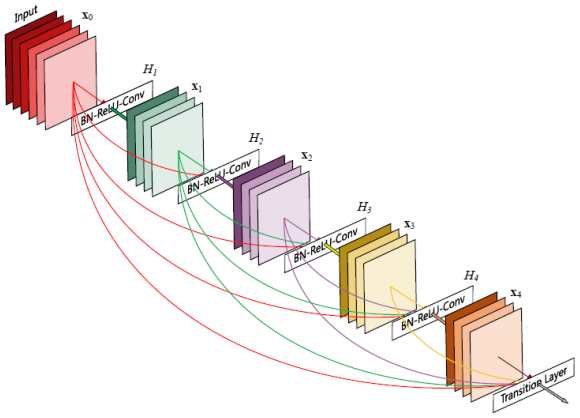


ImageNet top 5 error: 6.7% → 3.57%

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385

02 Densely Connected Convolutional Networks
DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017



Skip-Connection

Dense Connectivity

DenseNet Advantage

1. **They alleviate the vanishing-gradient problem**
2. **Strengthen feature propagation**
3. **Encourage feature reuse**
4. **Substantially reduce the number of parameters**

Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. “Densely Connected Convolutional Network”. arXiv:1608.06993

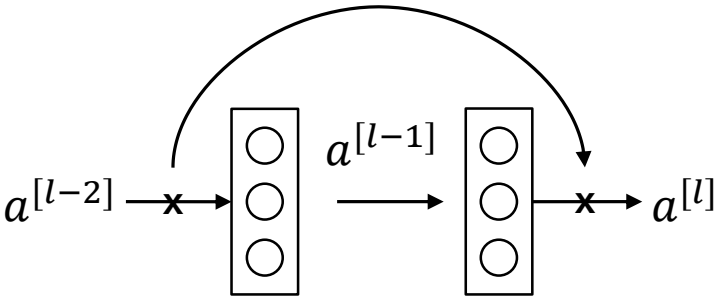
02 Densely Connected Convolutional Networks
DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017

Skip-Connection (ResNet)

$$a^{[l]} = H_l(a^{[l-1]}) + a^{[l-2]}$$

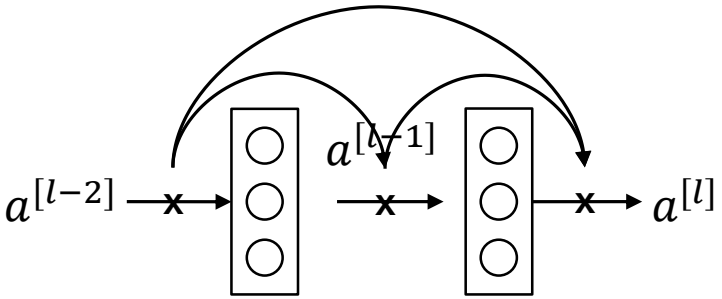
L + 1



Dense Connectivity (DenseNet)

$$a^{[l]} = H_l([a^{[0]}, a^{[1]}, \ldots, a^{[l-1]}])$$

L(L + 1) / 2



Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. “Densely Connected Convolutional Network”. arXiv:1608.06993

02 Densely Connected Convolutional Networks
DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017

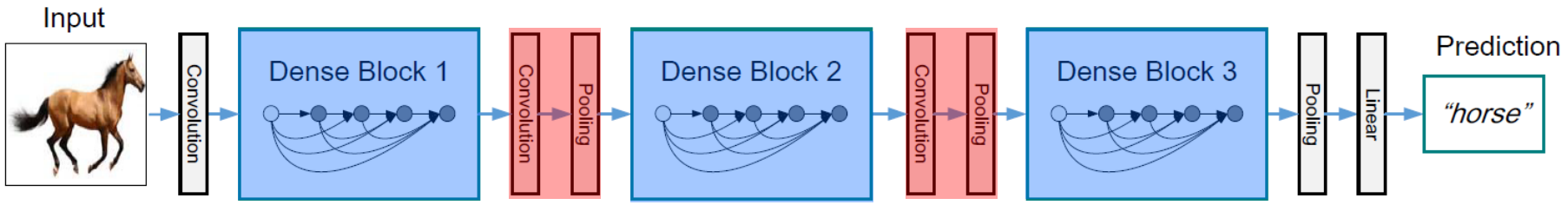
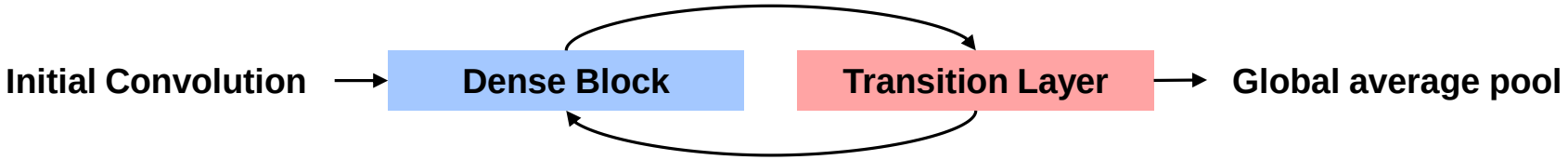


Figure 2: A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change feature-map sizes via convolution and pooling.



H_l(-):
BN → ReLU → 3x3 Conv

Down-sampling layers that change the size of feature maps

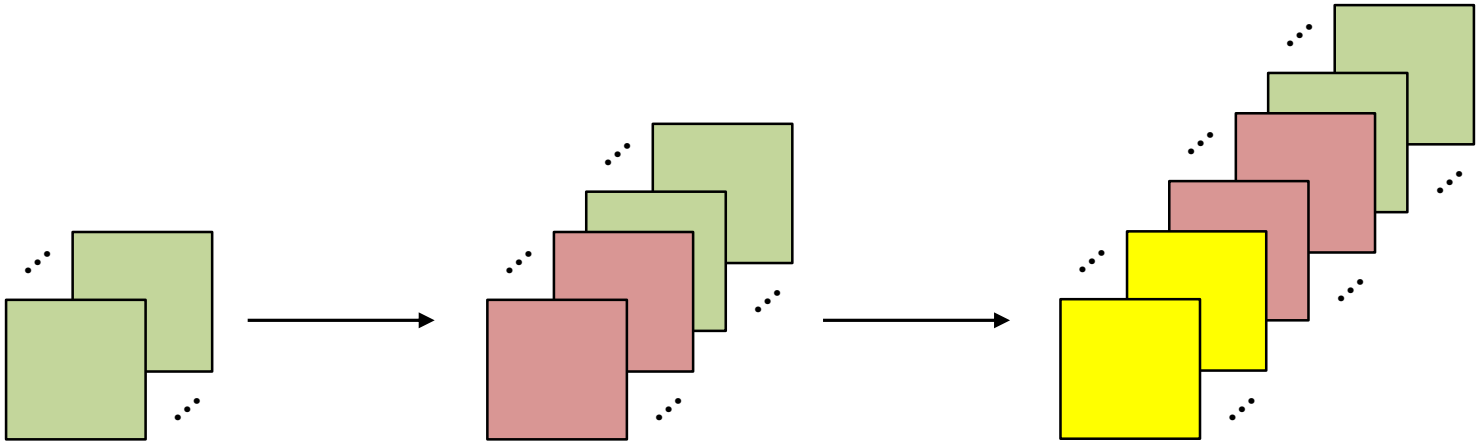
Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. "Densely Connected Convolutional Network". arXiv:1608.06993

02 Densely Connected Convolutional Networks
DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017

Dense Block

H_l(-):
BN → ReLU → 3x3 conv



Not viable when size of feature maps changes

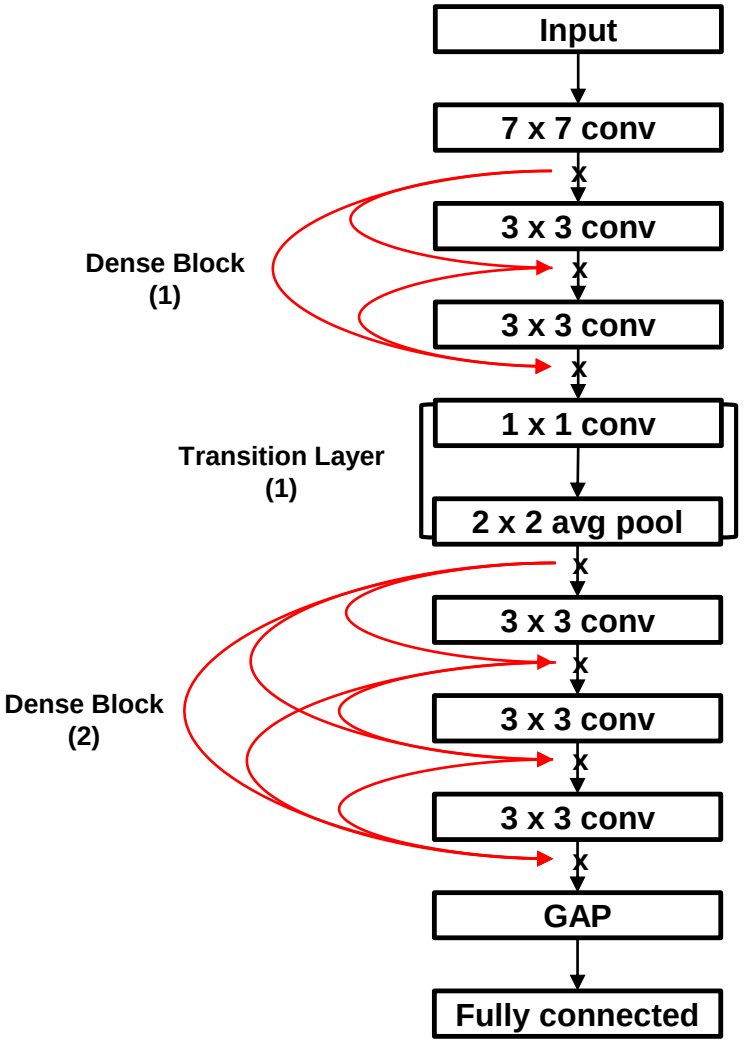
Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. “Densely Connected Convolutional Network”. arXiv:1608.06993

02 Densely Connected Convolutional Networks
DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017

Simple DenseNet

Convolution	:	**7 x 7 conv**
Dense Block (1)	:	**[3 x 3 conv] x 2**
Transition Layer (1)	:	**1 x 1 conv** **2 x 2 average pool**
Dense Block (2)	:	**[3 x 3 conv] x 3**
Classification Layer	:	**global average pool** **2D fully-connected, softmax**



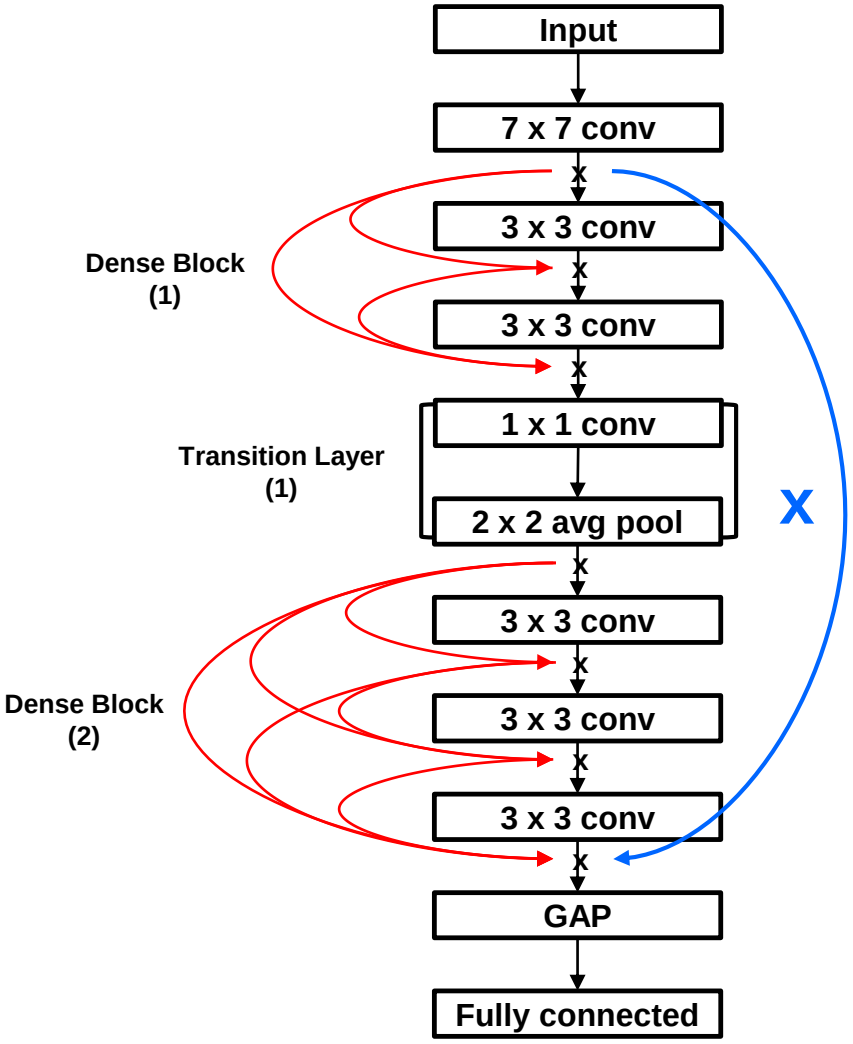
Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. “Densely Connected Convolutional Network”. arXiv:1608.06993

02 Densely Connected Convolutional Networks
DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017

Simple DenseNet

Convolution	:	**7 x 7 conv**
Dense Block (1)	:	**[3 x 3 conv] x 2**
Transition Layer (1)	:	**1 x 1 conv** **2 x 2 average pool**
Dense Block (2)	:	**[3 x 3 conv] x 3**
Classification Layer	:	**global average pool** **2D fully-connected, softmax**

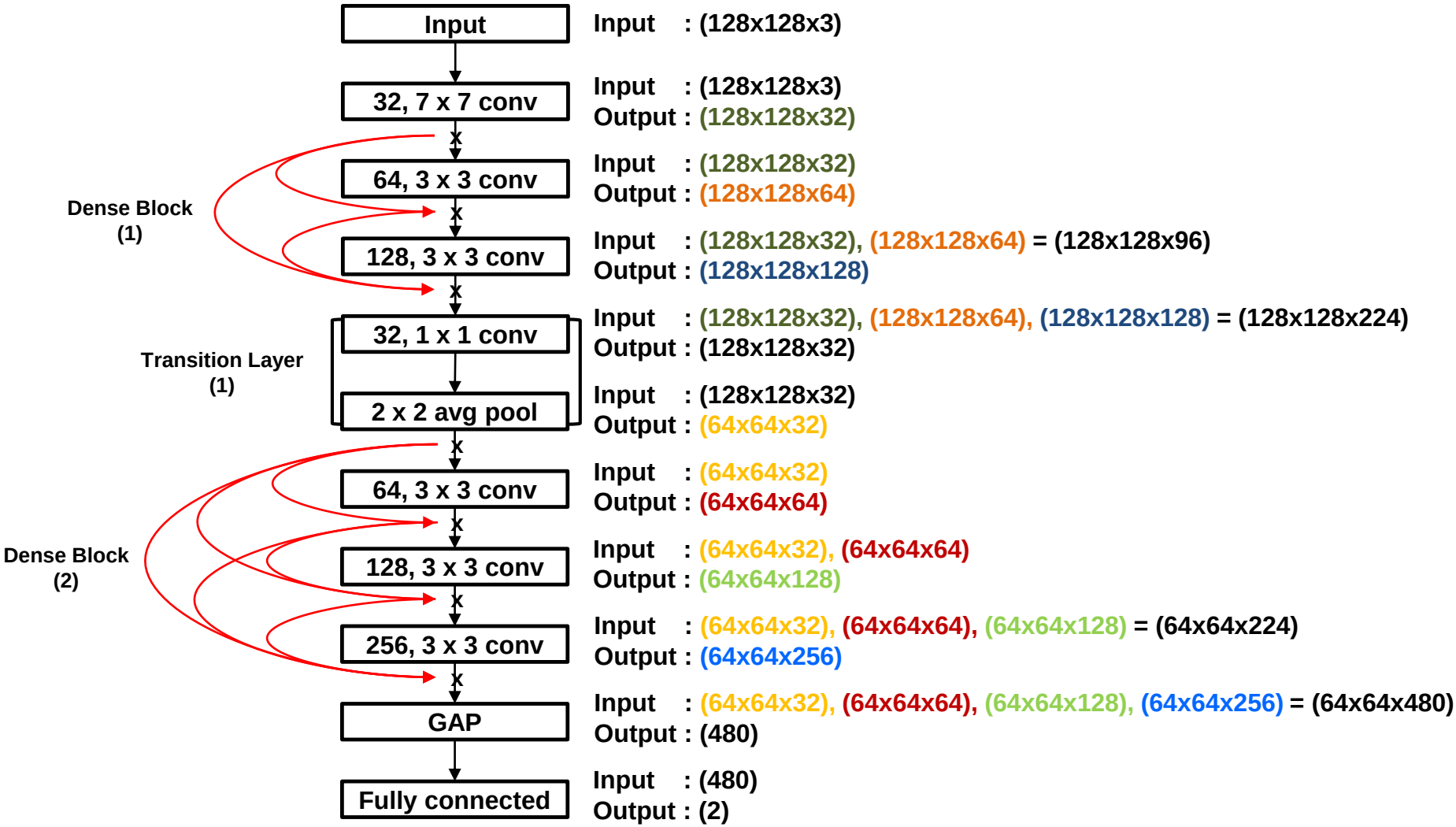


Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. “Densely Connected Convolutional Network”. arXiv:1608.06993

02 Densely Connected Convolutional Networks

DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017

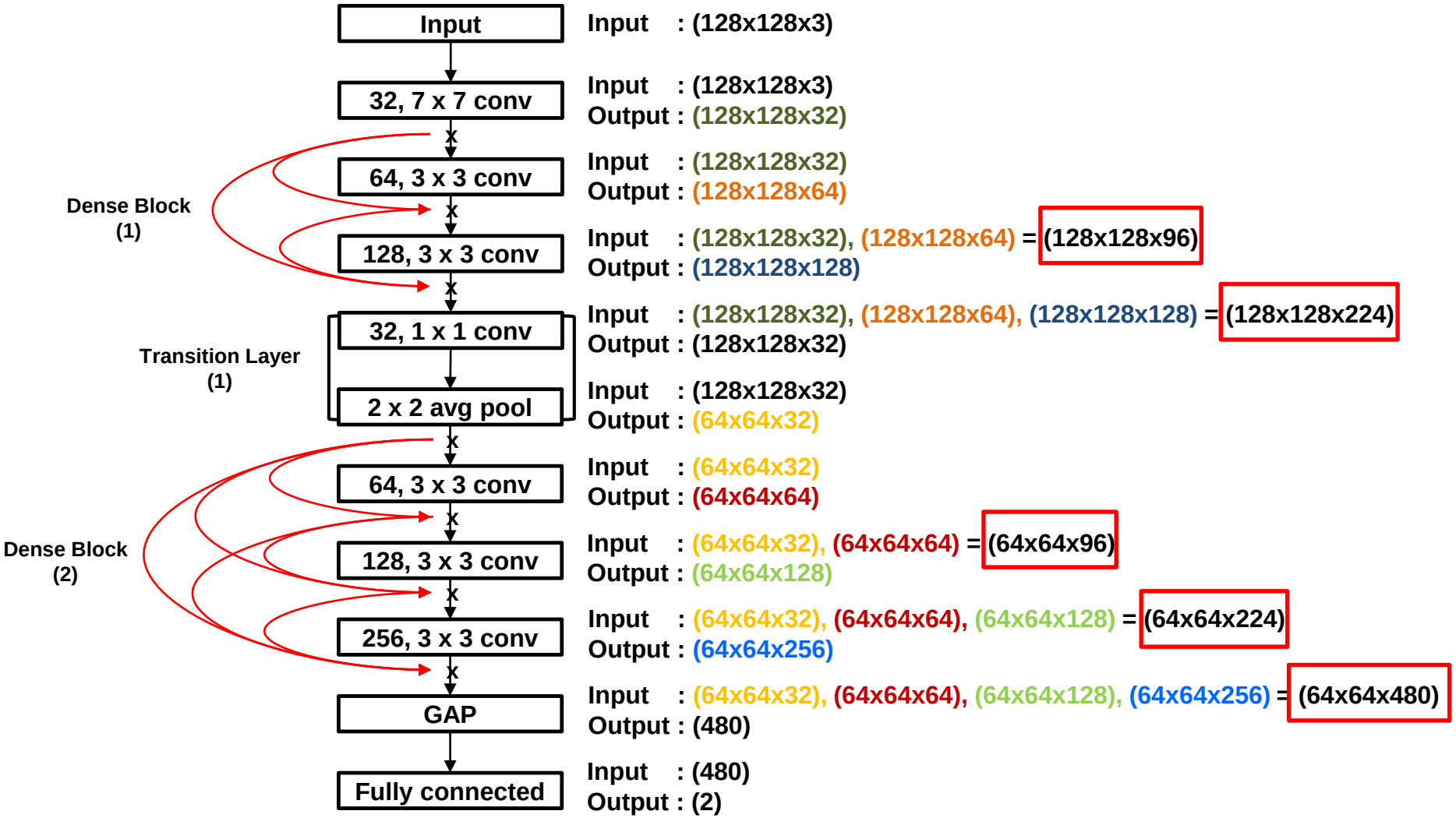


Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. “Densely Connected Convolutional Network”. arXiv:1608.06993

02 Densely Connected Convolutional Networks

DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017



Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. “Densely Connected Convolutional Network”. arXiv:1608.06993

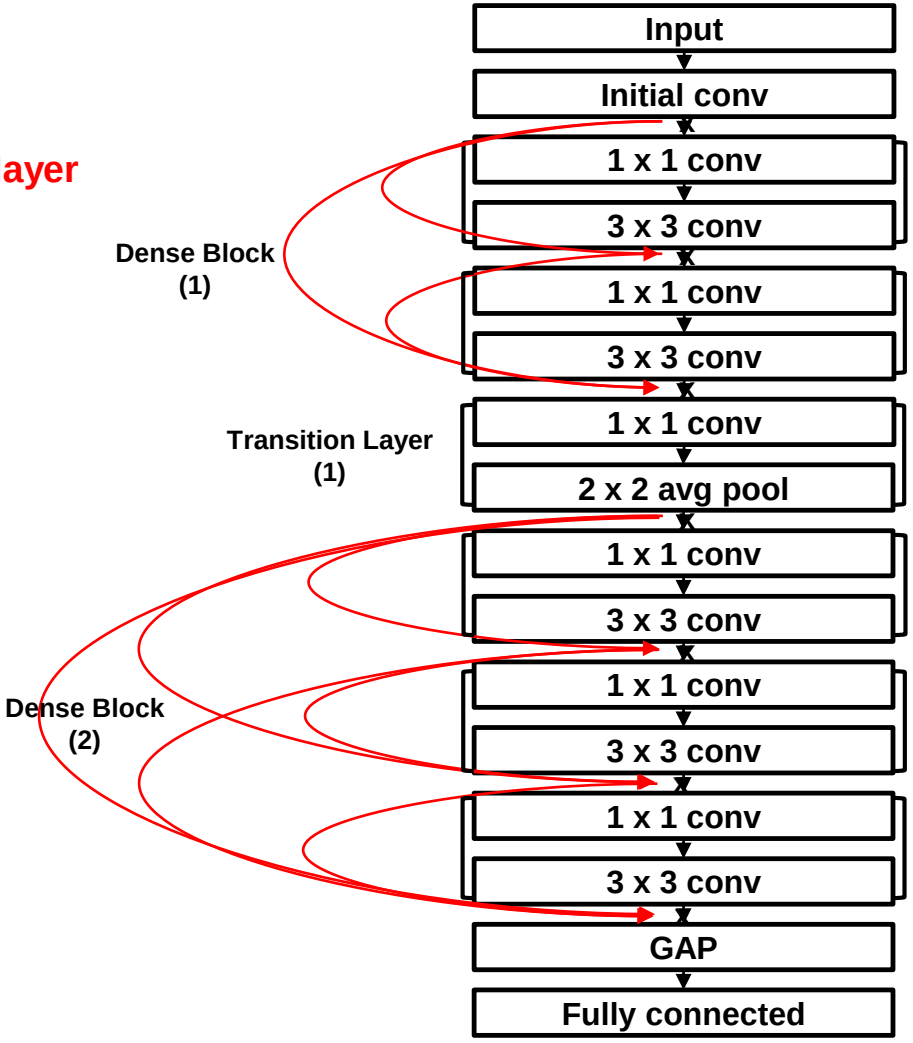
02 Densely Connected Convolutional Networks
DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017

Simple DenseNet

Bottleneck layer

Convolution	:	7 x 7 conv
Dense Block (1)	:	[1 x 1 conv, 3 x 3 conv] x 2
Transition Layer (1)	:	1 x 1 conv 2 x 2 average pool
Dense Block (2)	:	[1 x 1 conv, 3 x 3 conv] x 3
Classification Layer	:	global average pool 2D fully-connected, softmax



Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. “Densely Connected Convolutional Network”. arXiv:1608.06993

02 Densely Connected Convolutional Networks
DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017

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			

Table 1: DenseNet architectures for ImageNet. The growth rate for all the networks is $k = 32$. Note that each "conv" layer shown in the table corresponds the sequence BN-ReLU-Conv.

Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. "Densely Connected Convolutional Network". arXiv:1608.06993

02 Densely Connected Convolutional Networks
DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017

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			

Table 1: DenseNet architectures for ImageNet. The growth rate for all the networks is $k = 32$. Note that each "conv" layer shown in the table corresponds the sequence BN-ReLU-Conv.

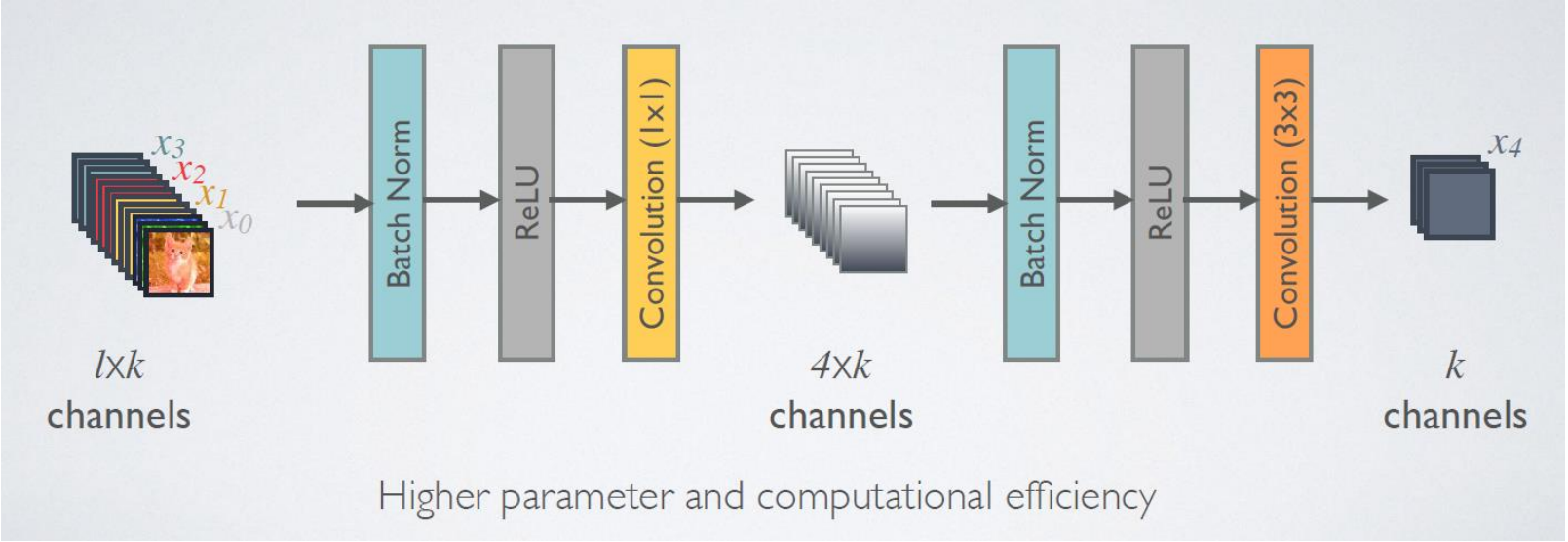
Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. "Densely Connected Convolutional Network". arXiv:1608.06993

02 Densely Connected Convolutional Networks
DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017

K : Growth Rate
(Hyperparameter)

DenseNet can have very narrow layers



H_l(-):
BN → ReLU → 3x3 conv

H_l'(-):
BN → ReLU → 1x1 conv → BN → ReLU → 3x3 conv

Computational efficiency ↑

Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. "Densely Connected Convolutional Network". arXiv:1608.06993

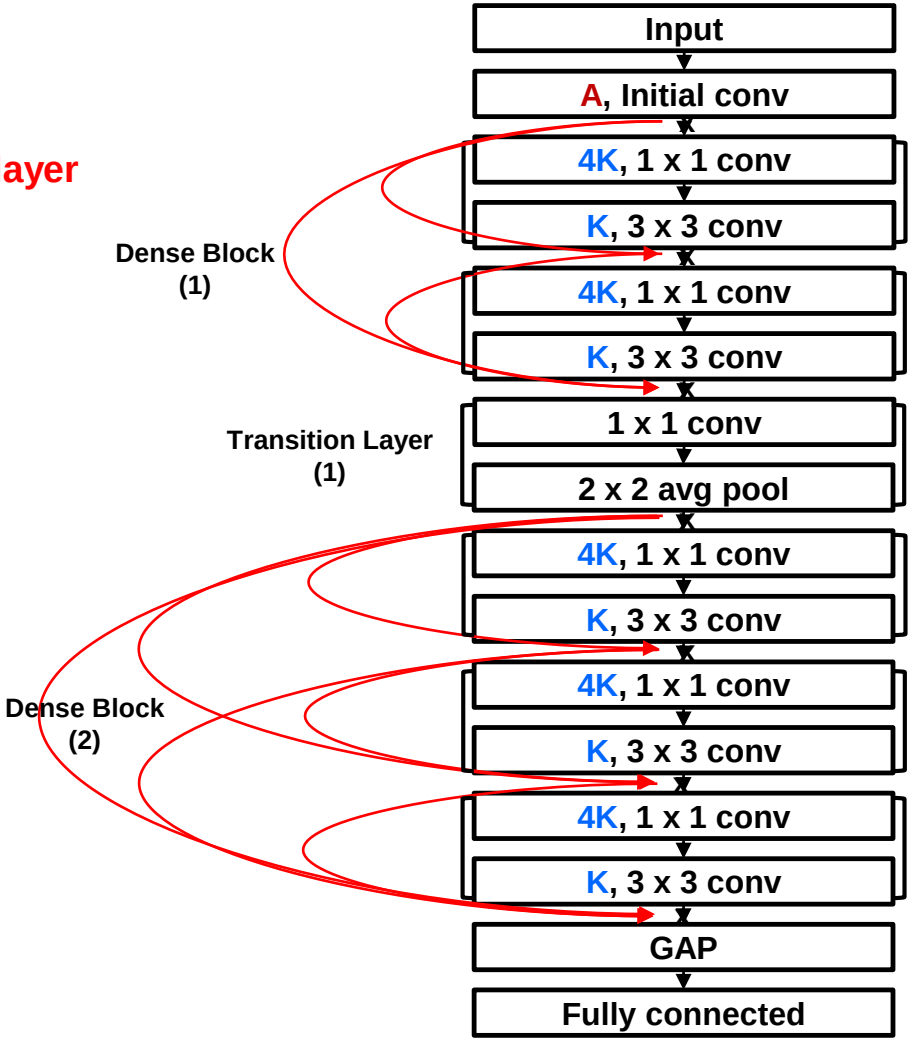
02 | Densely Connected Convolutional Networks
DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017

Simple DenseNet

Bottleneck layer

Convolution	:	7 x 7 conv
Dense Block (1)	:	[1 x 1 conv 3 x 3 conv] x 2
Transition Layer (1)	:	1 x 1 conv 2 x 2 average pool
Dense Block (2)	:	[1 x 1 conv 3 x 3 conv] x 3
Classification Layer	:	global average pool 2D fully-connected, softmax



Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. “Densely Connected Convolutional Network”. arXiv:1608.06993

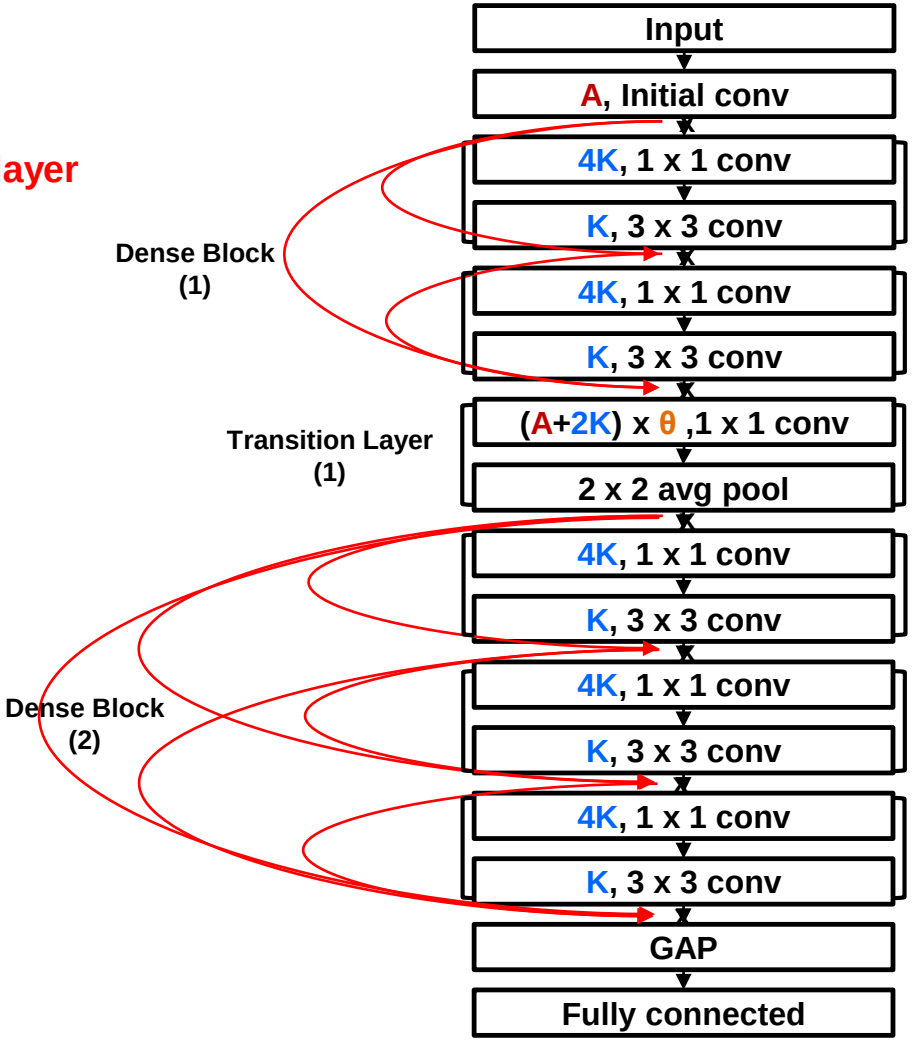
02 | Densely Connected Convolutional Networks
DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017

Simple DenseNet

Bottleneck layer

Convolution	:	7 x 7 conv
Dense Block (1)	:	[1 x 1 conv, 3 x 3 conv] x 2
Transition Layer (1)	:	1 x 1 conv 2 x 2 average pool
Dense Block (2)	:	[1 x 1 conv, 3 x 3 conv] x 3
Classification Layer	:	global average pool 2D fully-connected, softmax



Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. “Densely Connected Convolutional Network”. arXiv:1608.06993

02 | Densely Connected Convolutional Networks
DenseNet

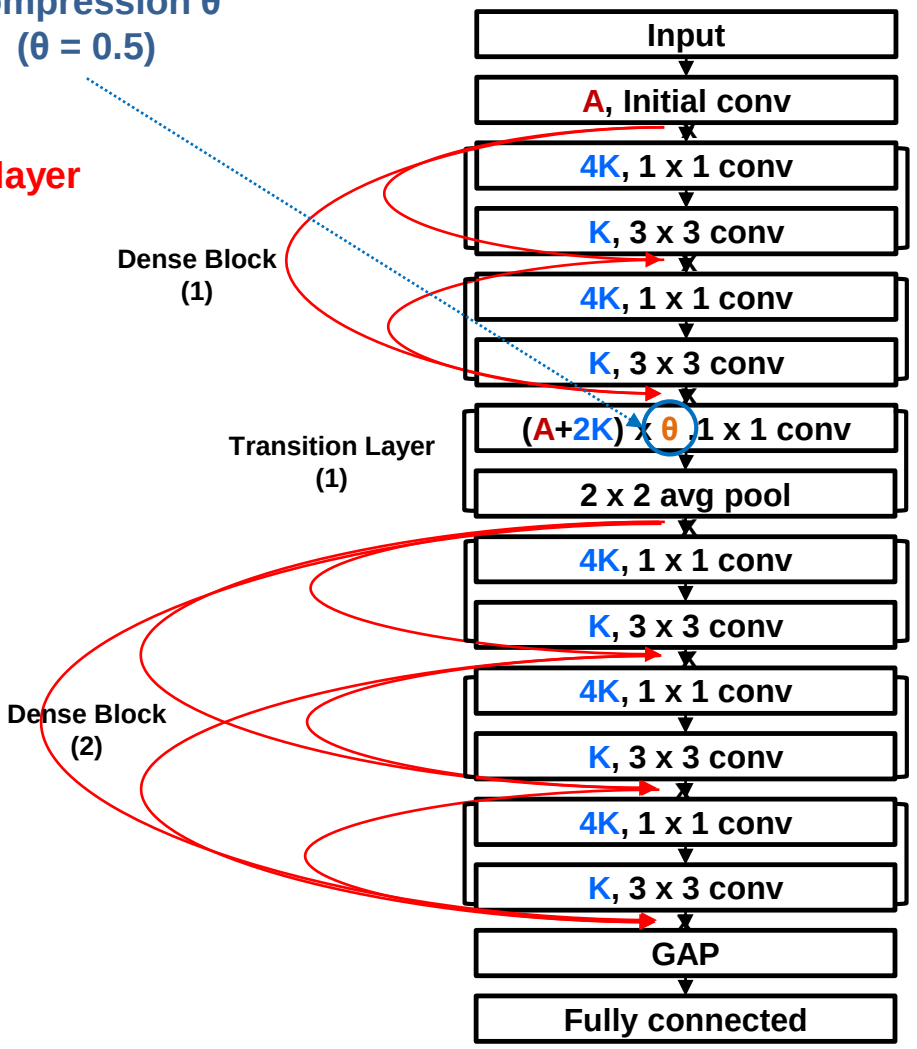
[Huang et al, 2016]
Best Paper in CVPR 2017

Simple DenseNet

Convolution	:	7 x 7 conv
Dense Block (1)	:	[1 x 1 conv, 3 x 3 conv] x 2
Transition Layer (1)	:	1 x 1 conv 2 x 2 average pool
Dense Block (2)	:	[1 x 1 conv, 3 x 3 conv] x 3
Classification Layer	:	global average pool 2D fully-connected, softmax

Compression θ
(θ = 0.5)

Bottleneck layer



Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. “Densely Connected Convolutional Network”. arXiv:1608.06993

02 Densely Connected Convolutional Networks
DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017

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**	-

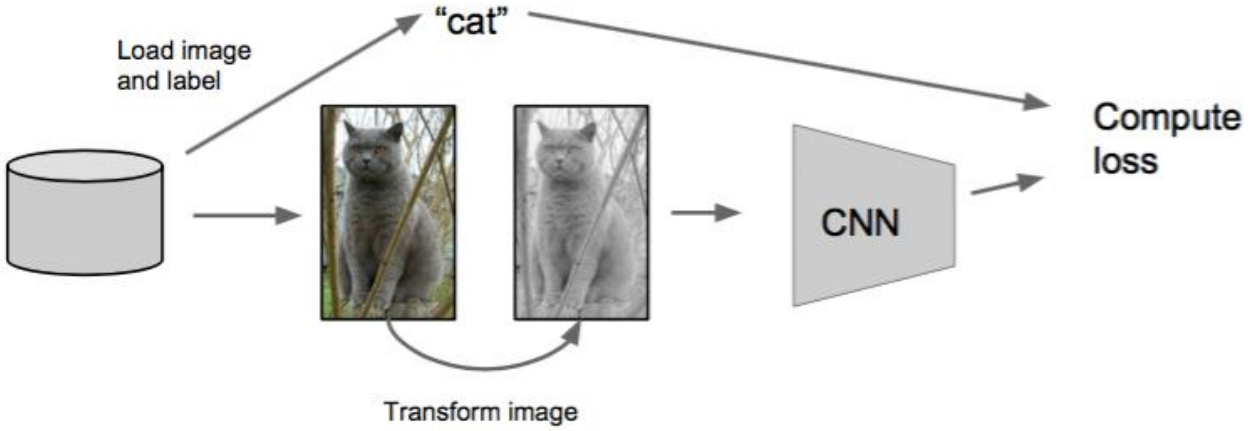
Table 2: Error rates (%) on CIFAR and SVHN datasets. k denotes network's growth rate. Results that surpass all competing methods are **bold** and the overall best results are **blue**. "+" indicates standard data augmentation (translation and/or mirroring). $*$ indicates results run by ourselves. All the results of DenseNets without data augmentation (C10, C100, SVHN) are obtained using Dropout. DenseNets achieve lower error rates while using fewer parameters than ResNet. Without data augmentation, DenseNet performs better by a large margin.

CIFAR-10 : Training : 50,000, Testing: 10,000, Class : 10
CIFAR-100 : Training : 50,000, Testing: 10,000, Class : 100
CIFAR-10+ : CIFAR-10 + Mirroring/Shifting (Data augmentation)
CIFAR-100+ : CIFAR-100 + Mirroring/Shifting (Data augmentation)
SVHN(Street View House Numbers) : Training : 73,257, Testing : 26,032

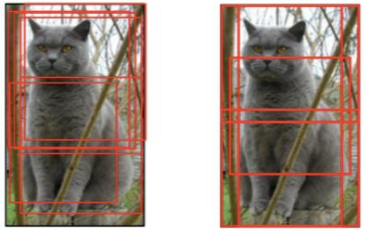
Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. "Densely Connected Convolutional Network". arXiv:1608.06993

[Huang et al, 2016]
Best Paper in CVPR 2017

Data augmentation



Random crops



이미지를 랜덤하게 잘라서 샘플링에 사용하는 방식

Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. "Densely Connected Convolutional Network". arXiv:1608.06993

02 Densely Connected Convolutional Networks
DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017

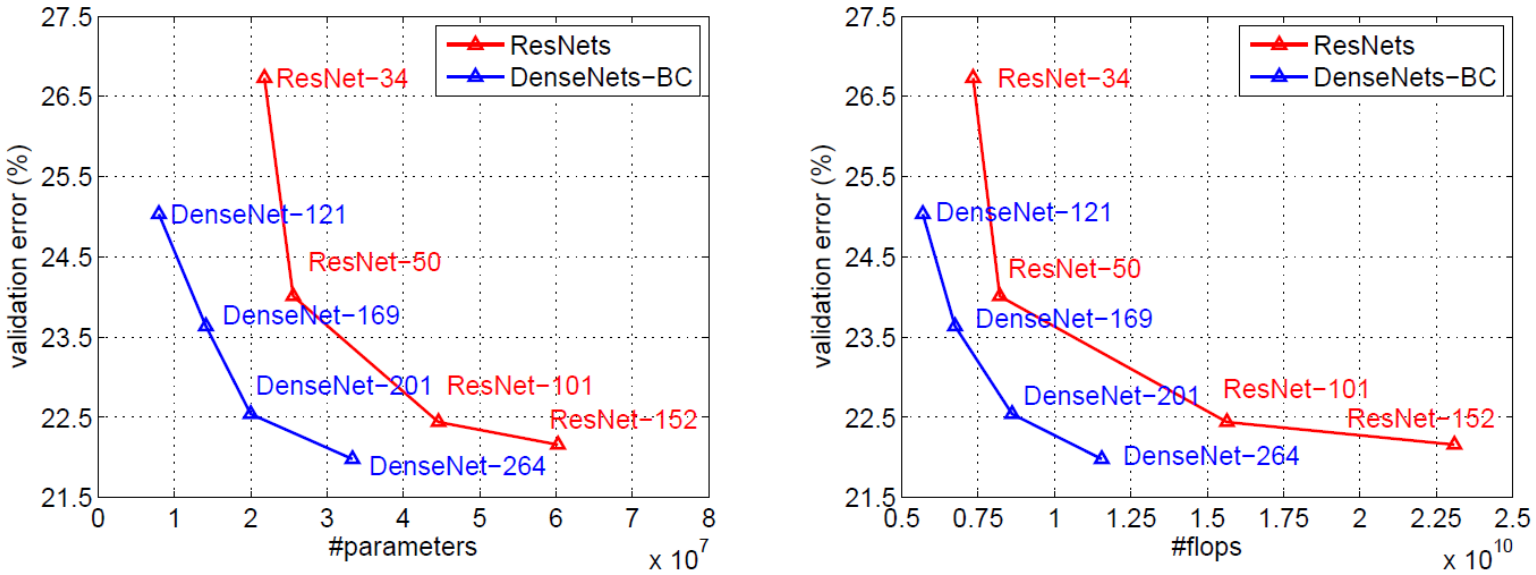


Figure 3: Comparison of the DenseNets and ResNets top-1 error rates (single-crop testing) on the ImageNet validation dataset as a function of learned parameters (*left*) and FLOPs during test-time (*right*).

*** FLOPs(floating-point operations per second)**
컴퓨터의 연산 속도를 나타내는 단위

Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. “Densely Connected Convolutional Network”. arXiv:1608.06993

02 Densely Connected Convolutional Networks
DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017

1. **They alleviate the vanishing-gradient problem**
2. **Strengthen feature propagation**
3. **Encourage feature reuse**
4. **Substantially reduce the number of parameters**

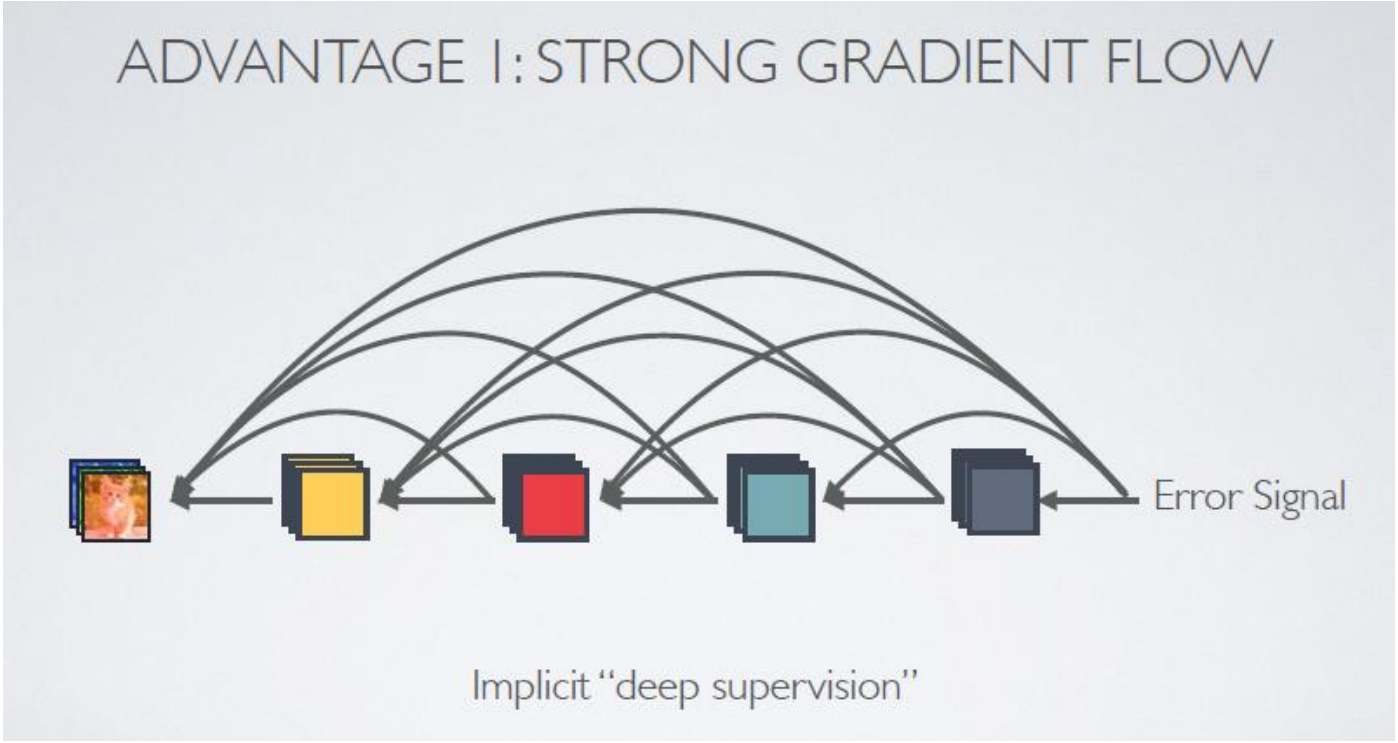
Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. “Densely Connected Convolutional Network”. arXiv:1608.06993

02 Densely Connected Convolutional Networks
DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017

1. **They alleviate the vanishing-gradient problem**

2. **Strengthen feature propagation**

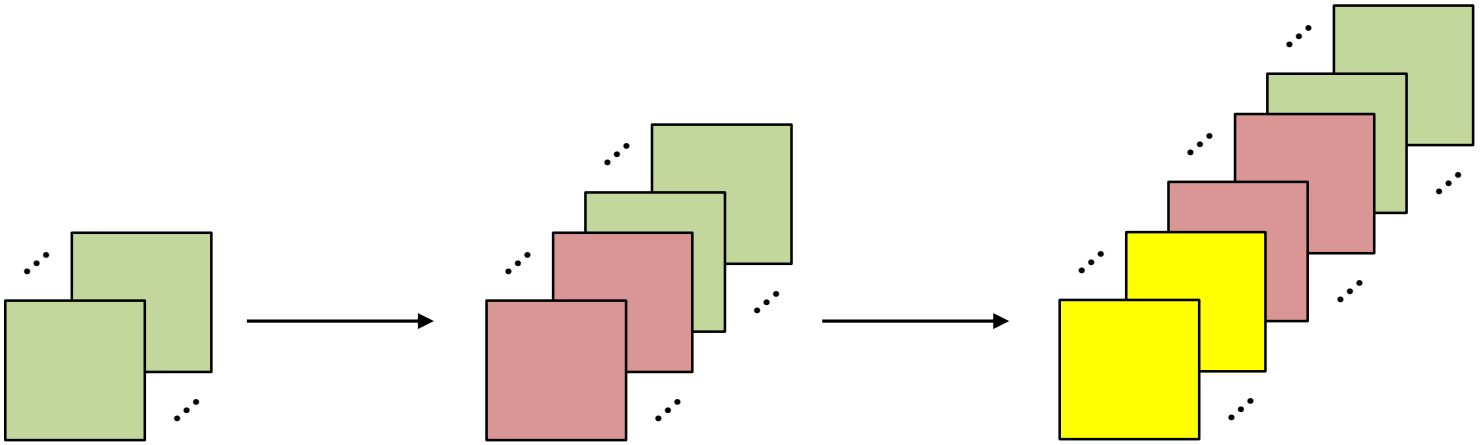


Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. "Densely Connected Convolutional Network". arXiv:1608.06993

02 Densely Connected Convolutional Networks
DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017

1. **They alleviate the vanishing-gradient problem**
2. **Strengthen feature propagation**
3. **Encourage feature reuse**



Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. “Densely Connected Convolutional Network”. arXiv:1608.06993

02 Densely Connected Convolutional Networks
DenseNet

[Huang et al, 2016]
Best Paper in CVPR 2017

1. **They alleviate the vanishing-gradient problem**
2. **Strengthen feature propagation**
3. **Encourage feature reuse**
4. **Substantially reduce the number of parameters**

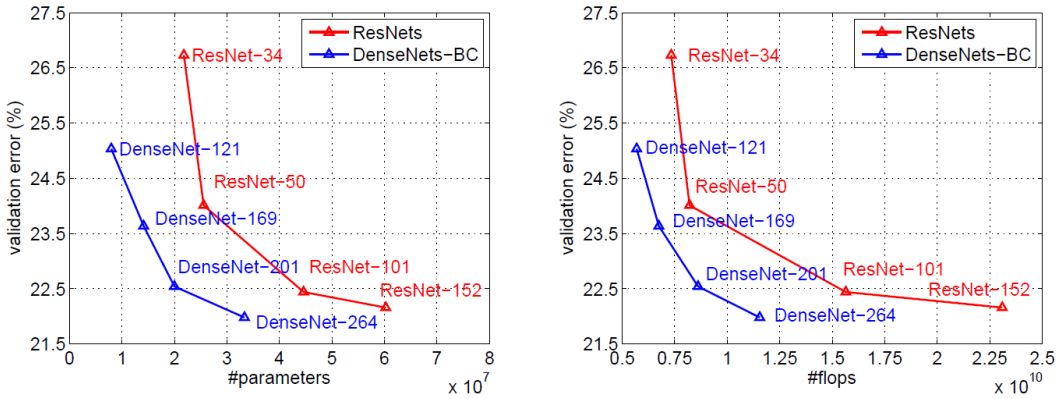


Figure 3: Comparison of the DenseNets and ResNets top-1 error rates (single-crop testing) on the ImageNet validation dataset as a function of learned parameters (*left*) and FLOPs during test-time (*right*).

Gao Huang*, Zhuang Liu*, Laurens van der Maaten and Kilian Weinberger. 2017. “Densely Connected Convolutional Network”. arXiv:1608.06993

Q & A