딥러닝 #3 CNN

AlexNet

VGG

GoogLeNet
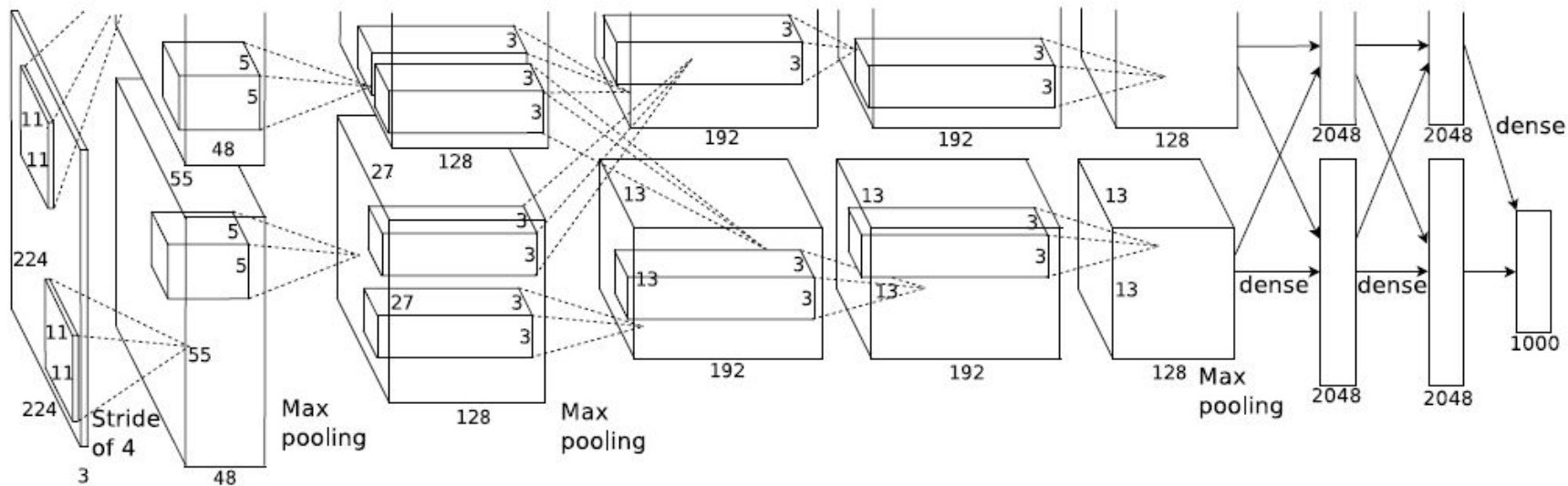
ResNet

DenseNet

Revolution of Depth

152 layers

22 layers   19 layers

8 layers   8 layers   shallow

3.57   6.7   7.3   11.7   16.4   25.8   28.2

ILSVRC'15   ILSVRC'14   ILSVRC'14   ILSVRC'13   ILSVRC'12   ILSVRC'11   ILSVRC'10
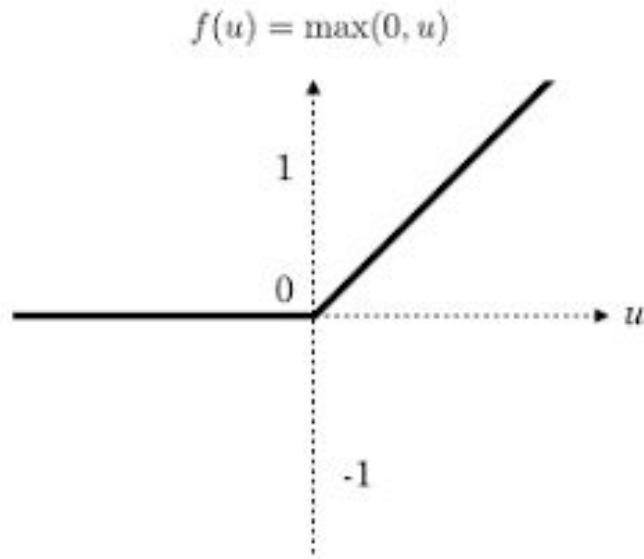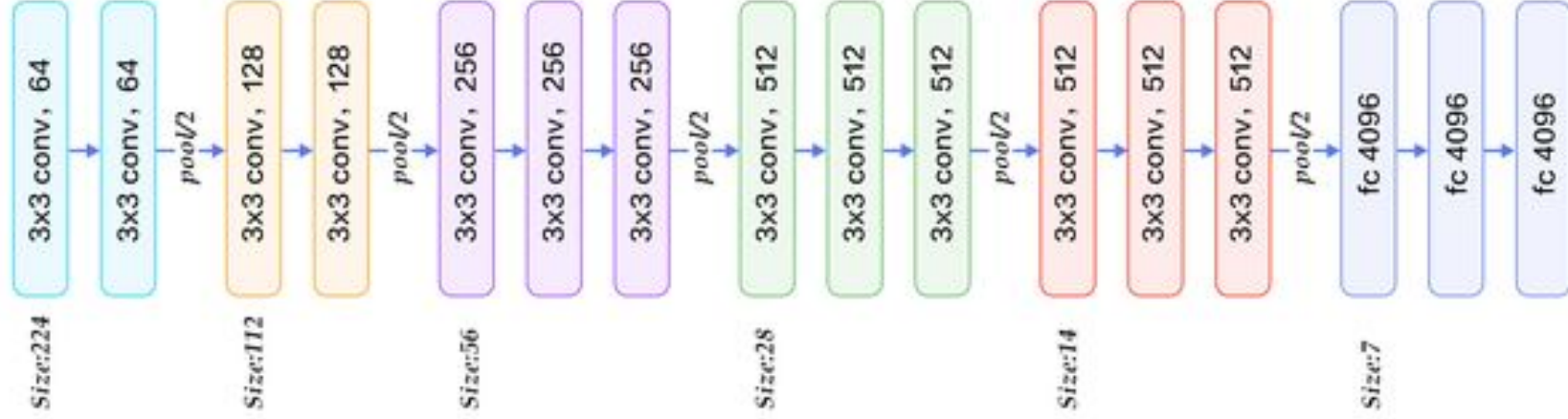ResNet   GoogleNet   VGG   AlexNet
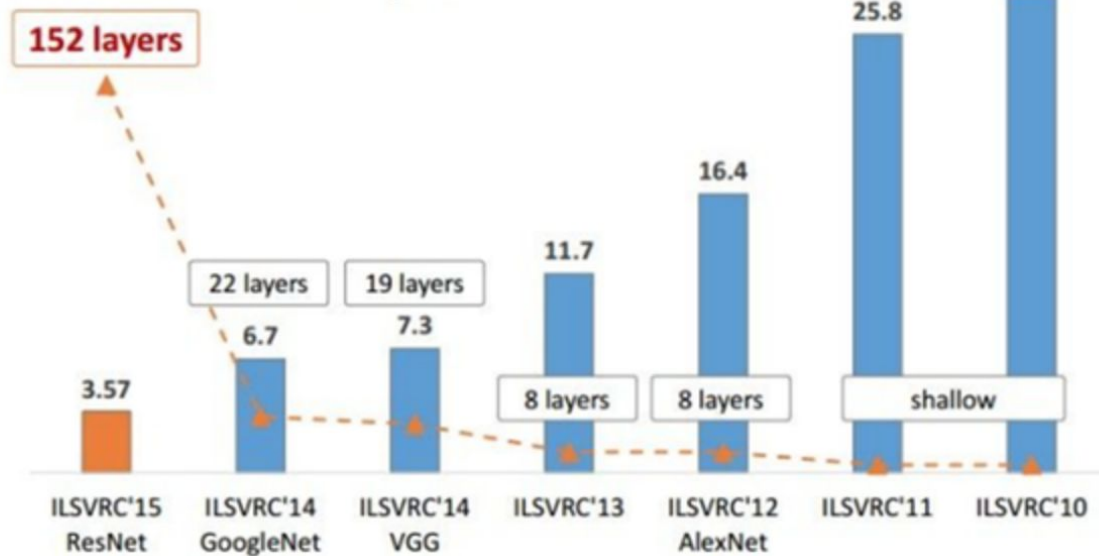
AlexNet

- ILSVRC의 Winner가 된 CNN 최초의 모델
- 두 개의 GPU를 활용한 병렬구조로 이루어졌다.

AlexNet

- ILSVRC의 Winner가 된 CNN 최초의 모델
- 두 개의 GPU를 활용하기 위한 병렬 구조
- relu 사용
- lrn – local response normalization
- overfitting 해결
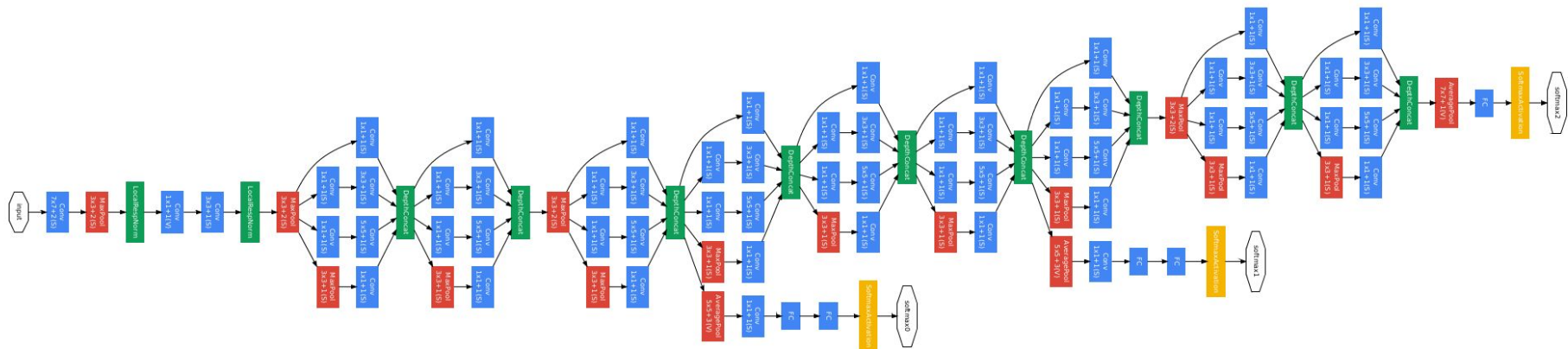 - data augmentation
  - dropout
  - batch normalize
  - l2 loss

$$f(u) = \max(0, u)$$

VGG16

VGG

- 3x3 convolution, stride 1 활용
- 간단한 구조로 좋은 성과를 내었다.
- 파라미터의 개수가 많다

| ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

GoogLeNet

GoogLeNet

- deep & wide
- 22 layers but 5 million parameters
- auxiliary classifier
- inception module 사용
  - 1x1 convolution filter를 통한 파라미터 조절
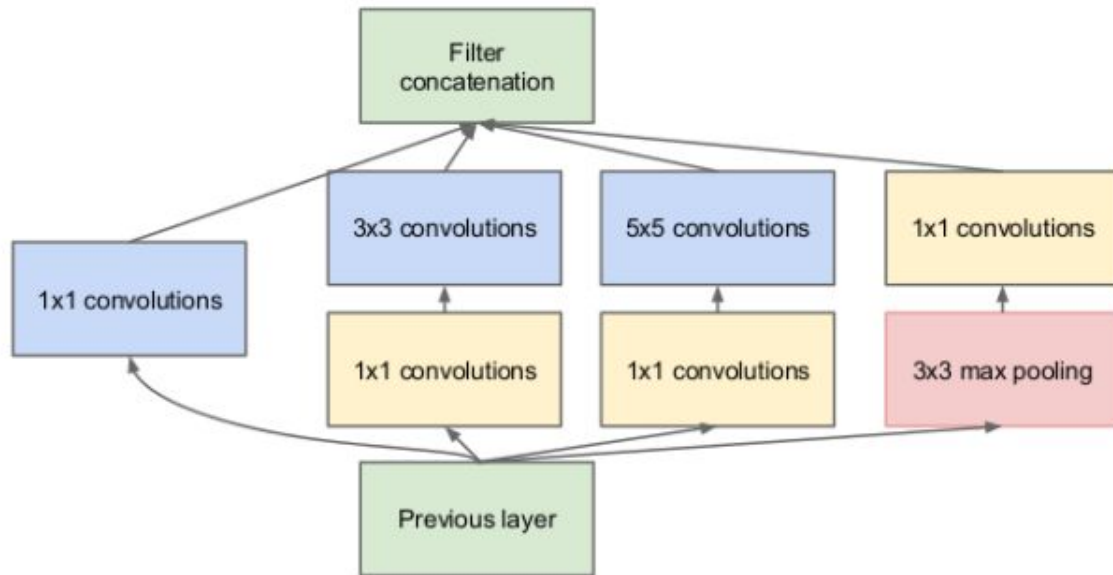 - channel-wise concat

GoogLeNet − Inception module
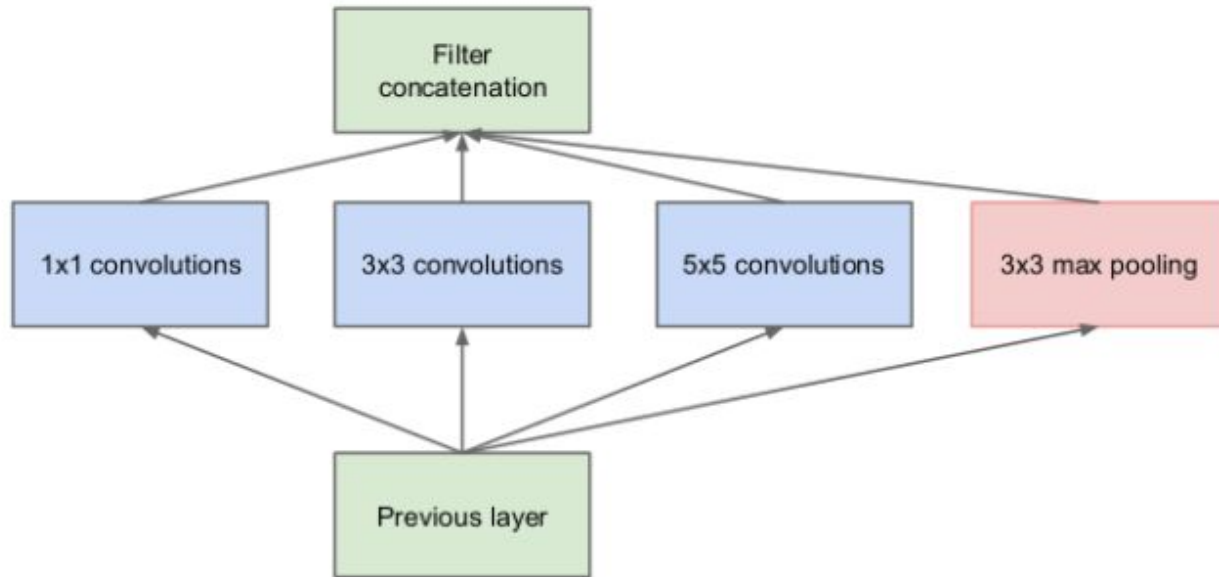
⟨naive inception module⟩

GoogLeNet − Inception module

⟨inception module⟩

| type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Table 1: GoogLeNet incarnation of the Inception architecture

ResNet
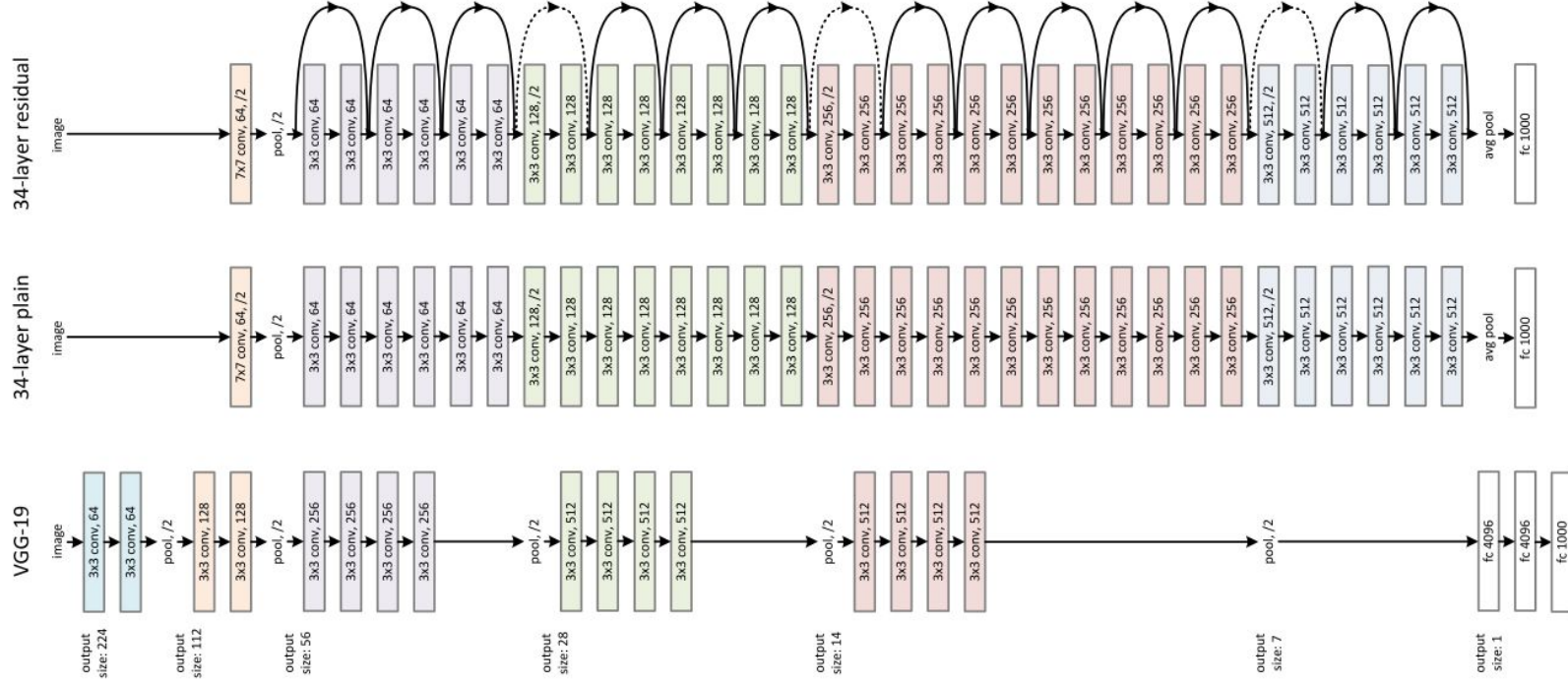
ResNet

- 100 layer 이상의 깊은 망
- skip connection
- residual learning building block
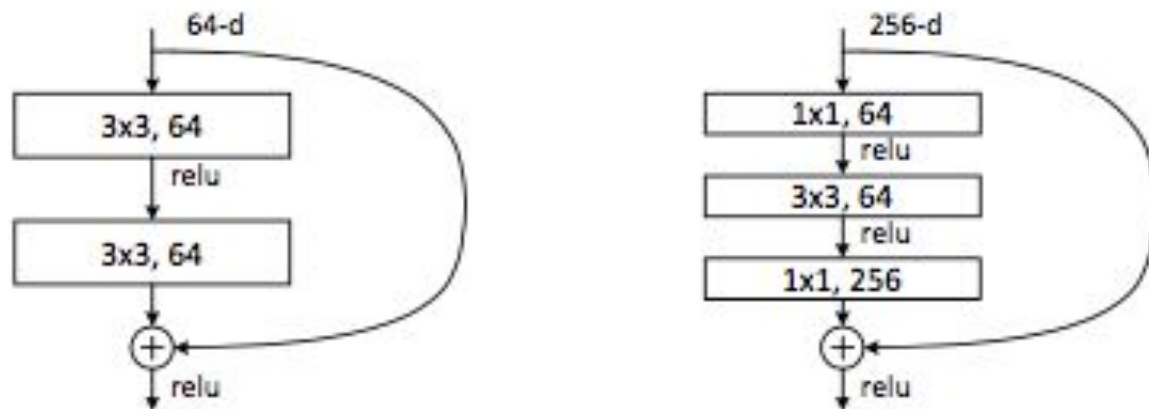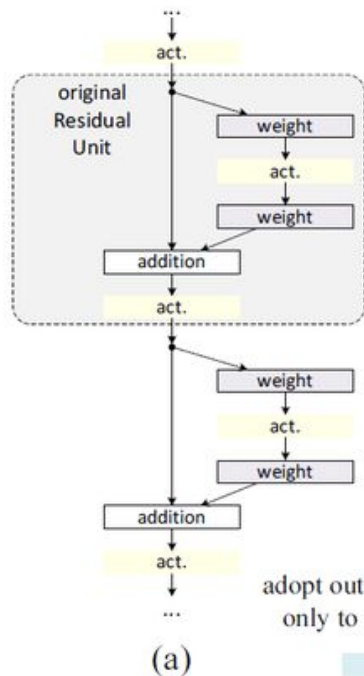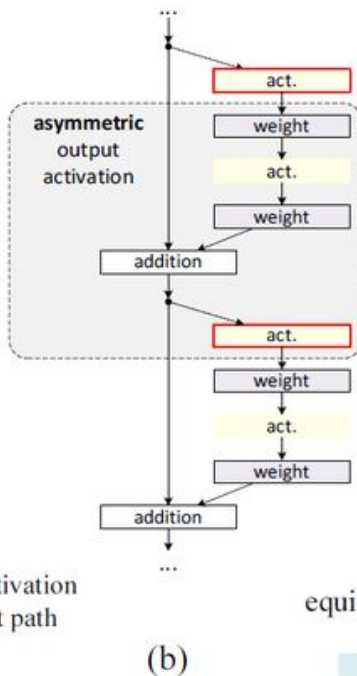- pre-activation 구조
- 깊은 망 학습의 표준 모델이 됨.
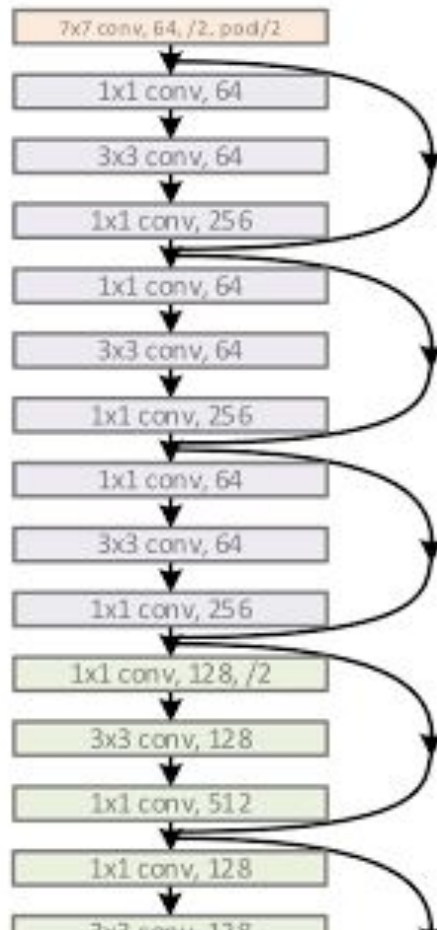
ResNet − residual block

Figure 5. A deeper residual function $\mathcal{F}$ for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a "bottleneck" building block for ResNet-50/101/152.
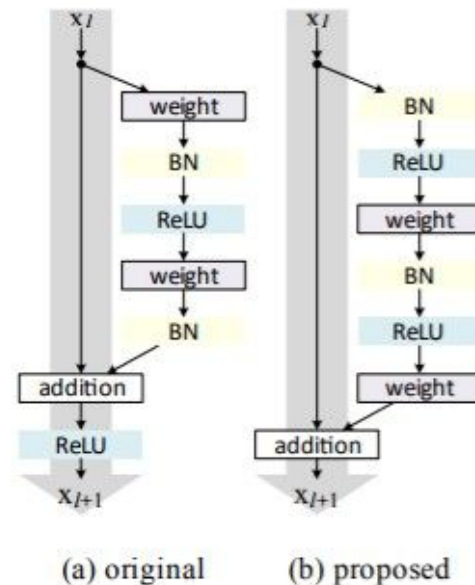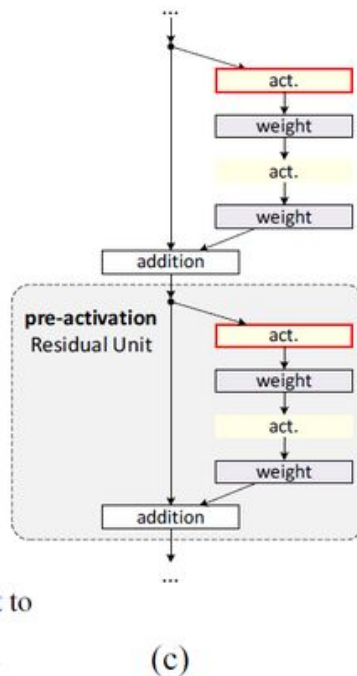
ResNet – pre-activation

ResNet − residual block

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3×3, 64 \\ 3×3, 64 \end{bmatrix}$ ×2	$\begin{bmatrix} 3×3, 64 \\ 3×3, 64 \end{bmatrix}$ ×3	$\begin{bmatrix} 1×1, 64 \\ 3×3, 64 \\ 1×1, 256 \end{bmatrix}$ ×3	$\begin{bmatrix} 1×1, 64 \\ 3×3, 64 \\ 1×1, 256 \end{bmatrix}$ ×3	$\begin{bmatrix} 1×1, 64 \\ 3×3, 64 \\ 1×1, 256 \end{bmatrix}$ ×3
conv3_x	28×28	$\begin{bmatrix} 3×3, 128 \\ 3×3, 128 \end{bmatrix}$ ×2	$\begin{bmatrix} 3×3, 128 \\ 3×3, 128 \end{bmatrix}$ ×4	$\begin{bmatrix} 1×1, 128 \\ 3×3, 128 \\ 1×1, 512 \end{bmatrix}$ ×4	$\begin{bmatrix} 1×1, 128 \\ 3×3, 128 \\ 1×1, 512 \end{bmatrix}$ ×4	$\begin{bmatrix} 1×1, 128 \\ 3×3, 128 \\ 1×1, 512 \end{bmatrix}$ ×8
conv4_x	14×14	$\begin{bmatrix} 3×3, 256 \\ 3×3, 256 \end{bmatrix}$ ×2	$\begin{bmatrix} 3×3, 256 \\ 3×3, 256 \end{bmatrix}$ ×6	$\begin{bmatrix} 1×1, 256 \\ 3×3, 256 \\ 1×1, 1024 \end{bmatrix}$ ×6	$\begin{bmatrix} 1×1, 256 \\ 3×3, 256 \\ 1×1, 1024 \end{bmatrix}$ ×23	$\begin{bmatrix} 1×1, 256 \\ 3×3, 256 \\ 1×1, 1024 \end{bmatrix}$ ×36
conv5_x	7×7	$\begin{bmatrix} 3×3, 512 \\ 3×3, 512 \end{bmatrix}$ ×2	$\begin{bmatrix} 3×3, 512 \\ 3×3, 512 \end{bmatrix}$ ×3	$\begin{bmatrix} 1×1, 512 \\ 3×3, 512 \\ 1×1, 2048 \end{bmatrix}$ ×3	$\begin{bmatrix} 1×1, 512 \\ 3×3, 512 \\ 1×1, 2048 \end{bmatrix}$ ×3	$\begin{bmatrix} 1×1, 512 \\ 3×3, 512 \\ 1×1, 2048 \end{bmatrix}$ ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		$1.8×10^9$	$3.6×10^9$	$3.8×10^9$	$7.6×10^9$	$11.3×10^9$

:tures for ImageNet. Building blocks are shown in brackets (see also Fig. 5), with the numbers of block

DenseNet

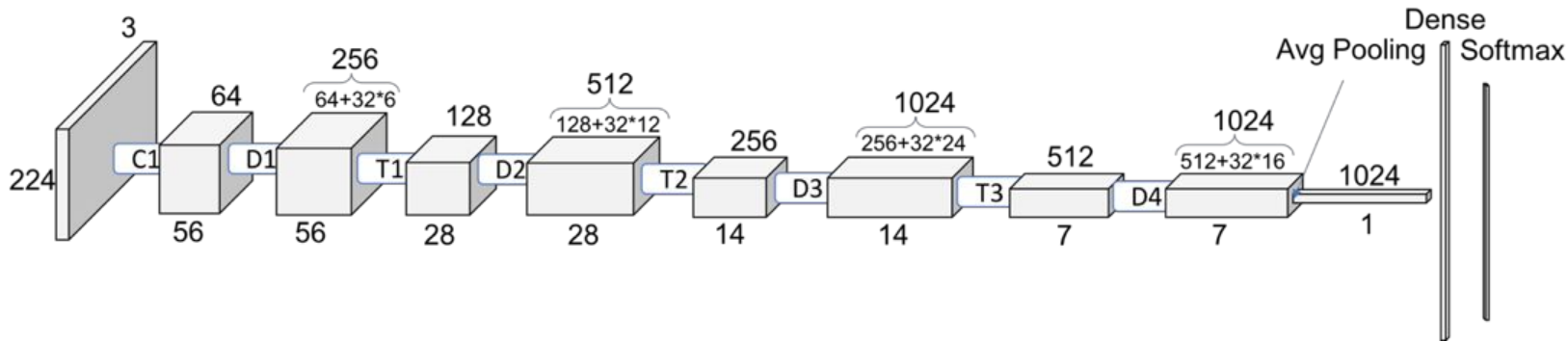Huang, Gao, et al. "Densely connected convolutional networks." *arXiv preprint arXiv:1608.06993* (2016).
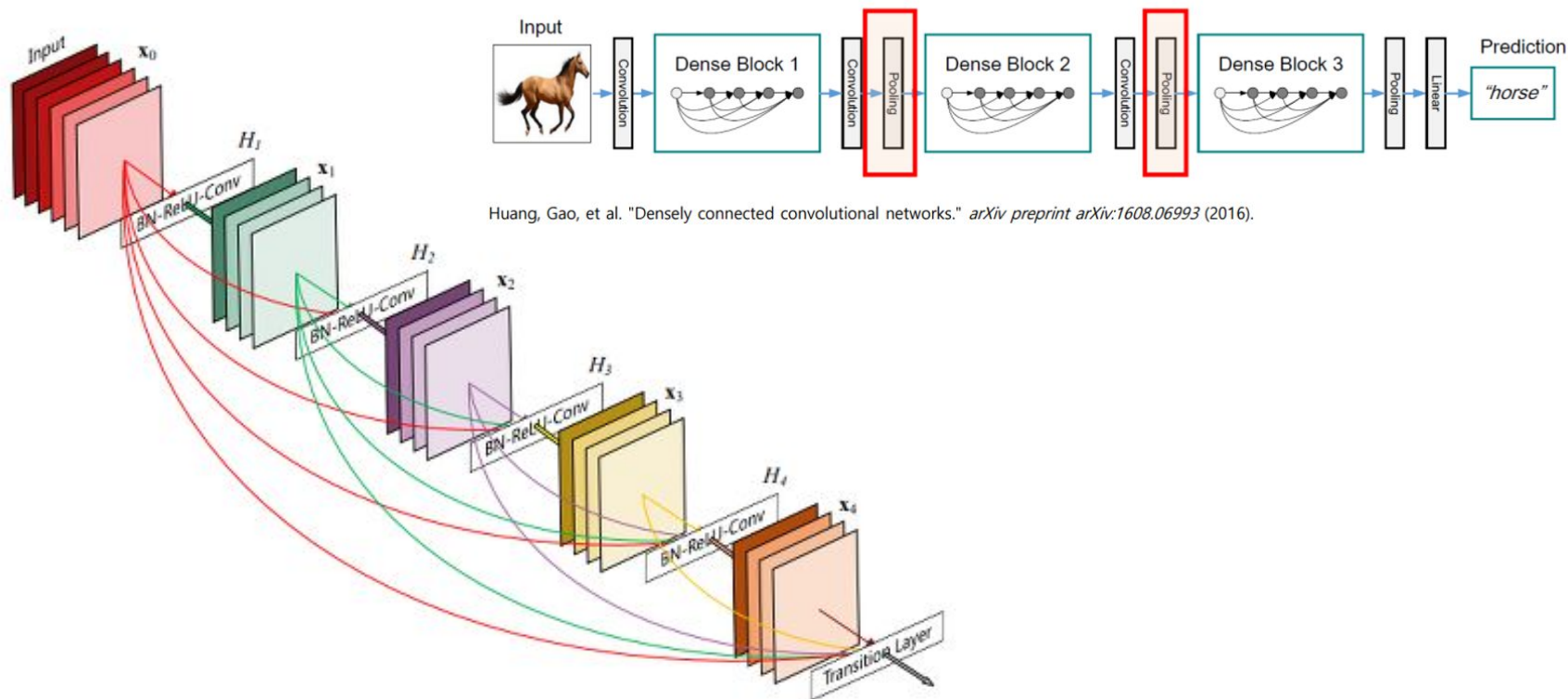
DenseNet

- ResNet의 아이디어 계승
- pre-activation 구조
- dense block
  - 이전의 특성 맵을 모두 누적해서 concat하는 결과

DenseNet

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	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			