

convolution
↑ ↑
image kernel

Inclass 24: Convolutional Neural Network (CNN)

[SCS4049] Machine Learning and Data Science

Seongsik Park (s.park@dgu.ac.kr)

School of AI Convergence & Department of Artificial Intelligence, Dongguk University

vector $\rightarrow$ fully conn $\rightarrow$ Scalar.

weight

image $\rightarrow$ conv. layer $\rightarrow$ image

kernel, $\times$ or $\div$

Convolutional Neural Network

Image classification

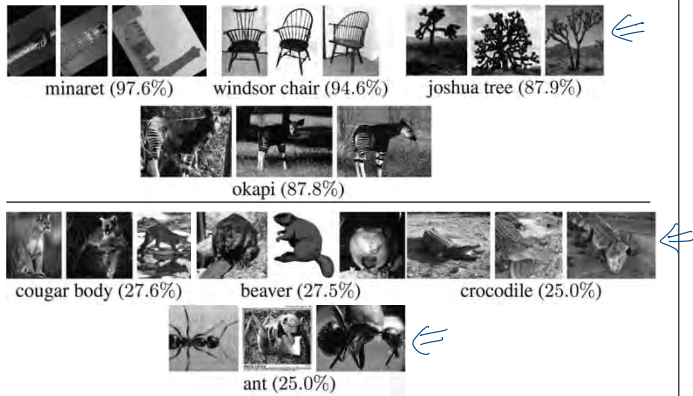
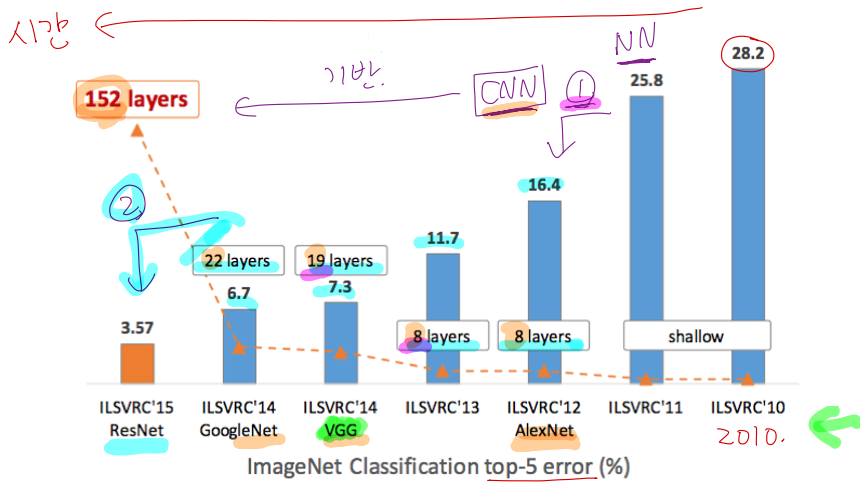


FIGURE 16.10: The spatial pyramid kernel is capable of complex image classification tasks. Here we show some examples of categories from the Caltech 101 collection on which the method does well (**top row**) and poorly (**bottom row**). The number is the percentage of images of that class classified correctly. Caltech 101 is a set of images of 101 categories of objects; one must classify test images into this set of categories (Section 16.3.2). *This figure was originally published as Figure 5 of “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” by S. Lazebnik, C. Schmid, and J. Ponce, Proc. IEEE CVPR 2006, © IEEE 2006.*

ImageNet challenge: image classification



더 깊게 풀러
구조에 영향.

The architecture of the visual cortex

사람이랑 비슷하더라.

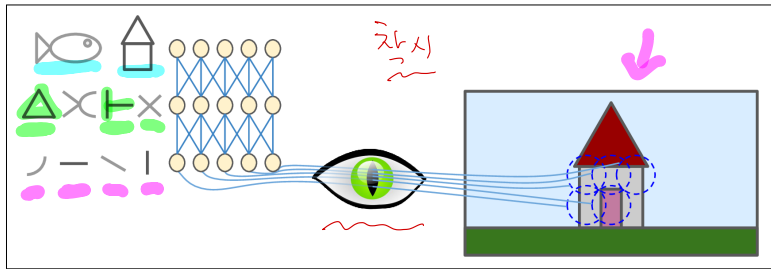


Figure 14-1. Local receptive fields in the visual cortex

In particular, they showed that many neurons in the visual cortex have a small **local receptive field**, meaning they react only to visual stimuli located in a limited region of the visual field (see Figure 14-1, in which the local receptive fields of five neurons are represented by dashed circles). The receptive fields of different neurons may overlap, and together they tile the whole visual field.

Convolutional layer

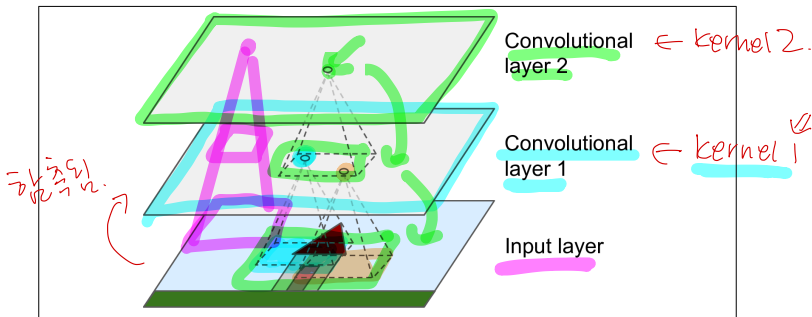


Figure 14-2. CNN layers with rectangular local receptive fields

The most important building block of a CNN is the **convolutional layer**: 6 neurons in the first convolutional layer are not connected to every single pixel in the input image (like they were in previous chapters), but only to pixels in their receptive fields (see Figure 14-2). In turn, each neuron in the second convolutional layer is connected only to neurons located within a small rectangle in the first layer. This architecture allows the network to concentrate on small low-level features in the first hidden layer, then assemble them into larger higher-level features in the next hidden layer, and so on.

Convolutional neural network

- CNN은 neural networks의 특별한 형태 ← conv.
- CNN은 Local Receptive Field 등에 의해서 생물학적인 영감
- 내재적이면서 자동적으로 유의미한 특징(feature)들을 추출
- CNN은 이미지로부터 위상학적 성질들을 추출
- CNN 역시 변형된 backpropagation 알고리즘으로 학습
- 최소한의 전처리 만으로도 픽셀로부터 시각적 패턴을 인식
- 극도로 변화가 많은 패턴들도 인식 가능함

image $\rightarrow$ conv $\rightarrow$ feature
 $\uparrow$
 kernel { 예: 경: 직접 설계.
 CNN: 자동 설계.

Convolution

Given a filter kernel $\mathcal{H}$, the convolution of the kernel with image $\mathcal{F}$ is an image $\mathcal{R}$. The (i,j) -th component of $\mathcal{R}$ is given by

$$R_{ij} = \sum_{u,v} H_{i-u,j-v} F_{uv}. \quad (1)$$

- **kernel** of the filter: the pattern of weights used for a linear filter
- **convolution**: the process of applying the filter

This operation is called **convolution**

$$R(f) = (h * f) \quad (2)$$

- *commutative*: $(g * h)(x) = (h * g)(x)$
- *associative*: $(f * (g * h)) = ((f * g) * h)$
- *distributive*: $f * (g + h) = f * g + f * h$

Padding (border effects)

- **zero**: set all pixels outside the source image to 0
- **constant**: set all pixels outside the source image to a specified border value
- **clamp**: repeat edge pixels indefinitely
- **wrap**: loop “around” the image in a “toroidal” configuration
- **mirror**: reflect pixels across the image edge
- **extend**: extend the signal by subtracting the mirrored version of the signal from the edge pixel value

Padding (border effects)

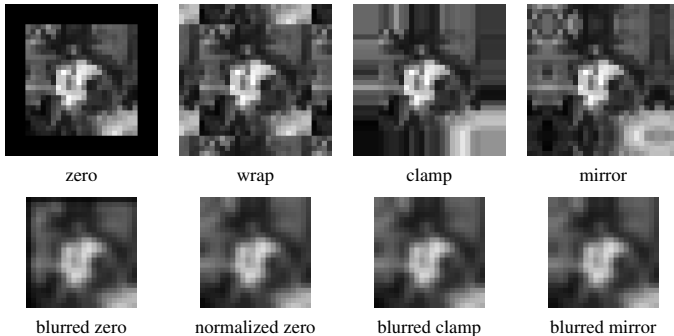


Figure 3.13 *Border padding (top row) and the results of blurring the padded image (bottom row). The normalized zero image is the result of dividing (normalizing) the blurred zero-padded RGBA image by its corresponding soft alpha value.*

Convolution: connections and padding

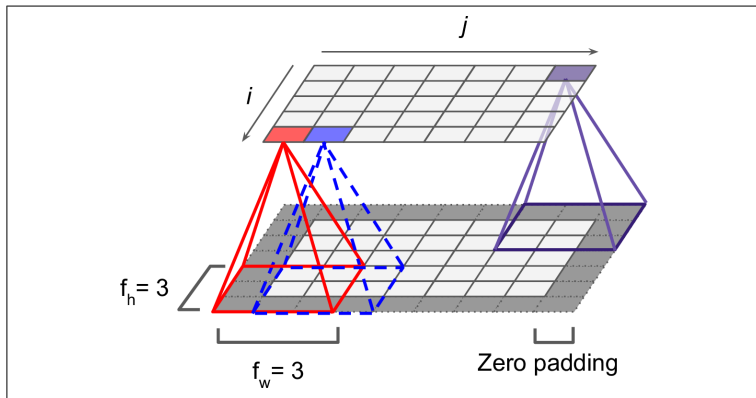


Figure 14-3. Connections between layers and zero padding

Convolution: stride and dimensional reduction

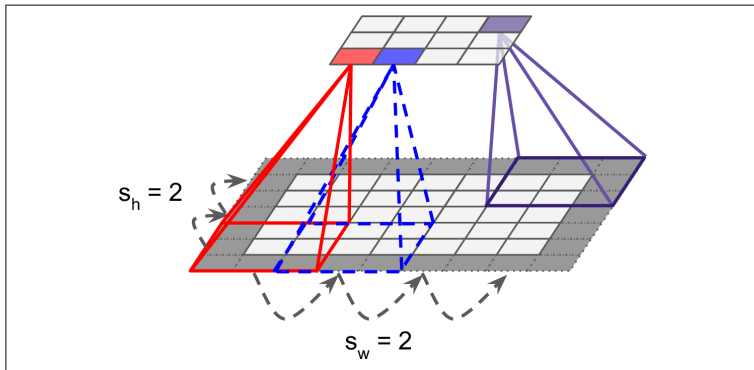


Figure 14-4. Reducing dimensionality using a stride of 2

Stride: 몇 칸씩 띄워서 움직여서 계산.

1x1

$$\begin{array}{|c|c|} \hline 0 & 0 \\ \hline 0 & 0 \\ \hline \end{array} \quad \begin{array}{|c|c|} \hline 0 & 0 \\ \hline 0 & 0 \\ \hline \end{array} \times \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \underline{0} & \underline{0} \\ \underline{0} & \underline{0} \end{bmatrix}$$

2 x 2

input size

6×6

6×6

6 6

12×24

12 24

kernel size

2×2

3×3

5×3

stride

1×1

2×2

2 2

3×2

3 2

padding

0

0

↗

0

output size

6×6

⇒ 3×3

3 3

4×12

Convolution: example

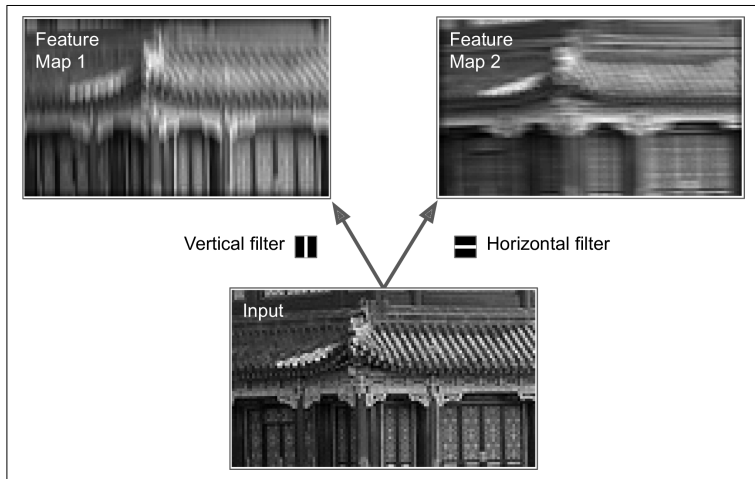
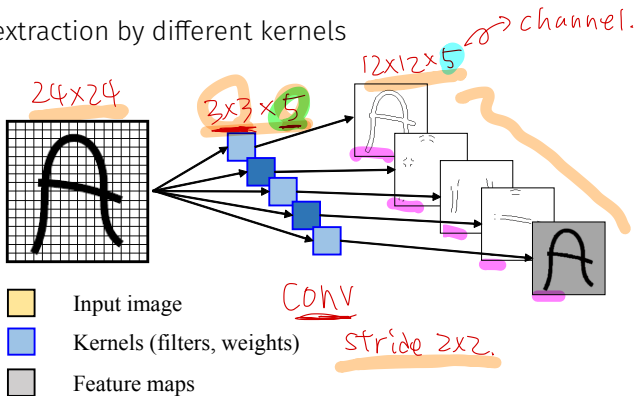


Figure 14-5. Applying two different filters to get two feature maps

Convolution: filter effect

Feature extraction by different kernels



- 컨볼루션은 입력 이미지의 다른 부분들에 존재하는 동일한 특징들을 찾아냄
- Feature map에 있는 뉴런은 template or kernel과 일치되는 경우에 활성화

CNN: convolution layers with multiple feature maps

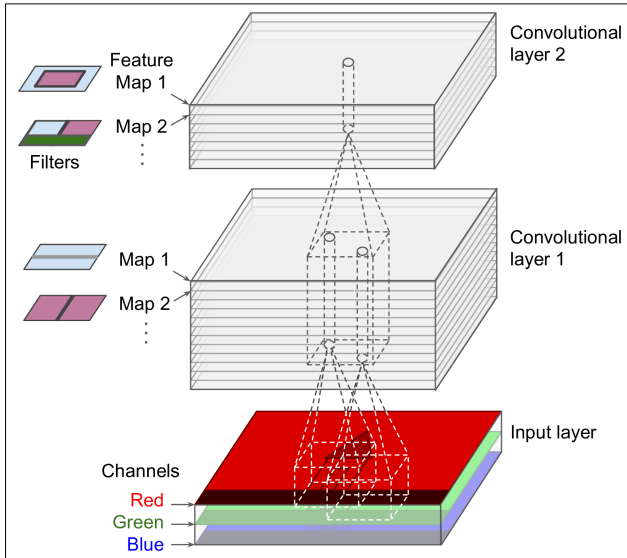
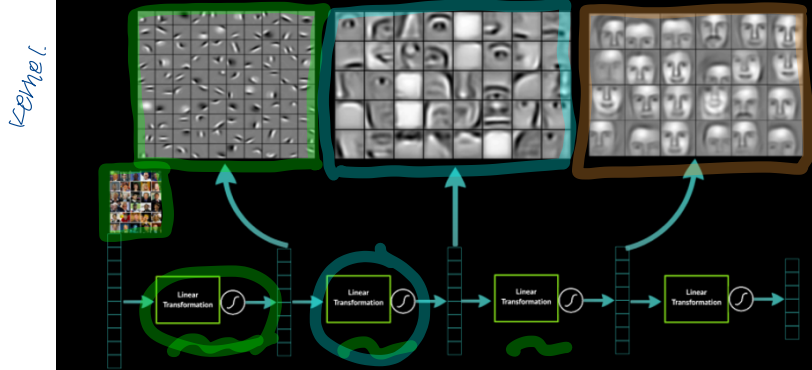


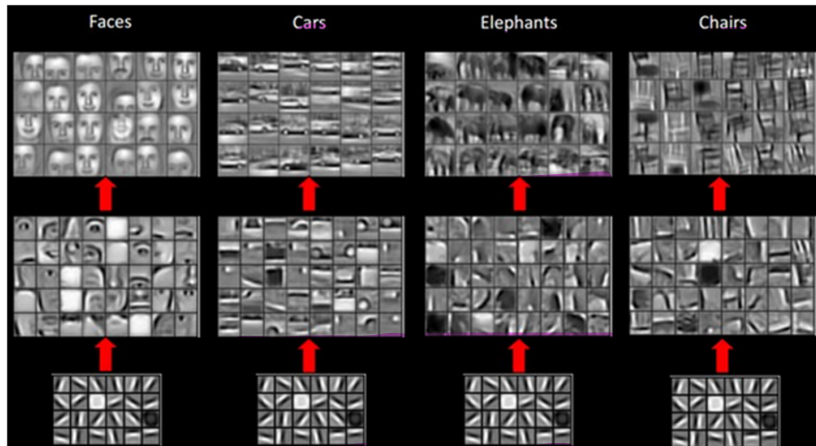
Figure 14-6. Convolution layers with multiple feature maps, and images with three color channels

CNN: hierarchical learning of features

Deep Learning learns layers of features



CNN: hierarchical learning of features



CNN: dimensional reduction by pooling layer

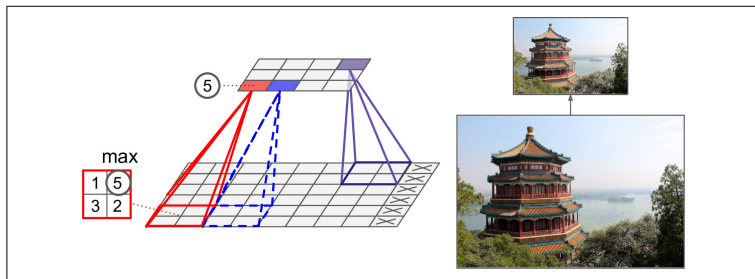


Figure 14-8. Max pooling layer (2×2 pooling kernel, stride 2, no padding)

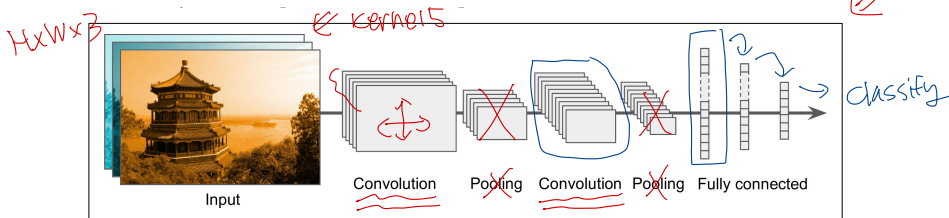


Figure 14-11. Typical CNN architecture

Residual Learning

ResNet.

ResNet: Deep Residual Learning for Image Recognition

Deep Residual Learning for Image Recognition, CVPR2016

ResNet

Deep residual learning for image recognition

[K He, X Zhang, S Ren, J Sun](#) - ... and pattern recognition, 2016 - openaccess.thecvf.com

... Deeper neural networks are more difficult to train. We present a **residual learning** framework to ease the training of networks that are substantially **deeper** than those used previously. ...

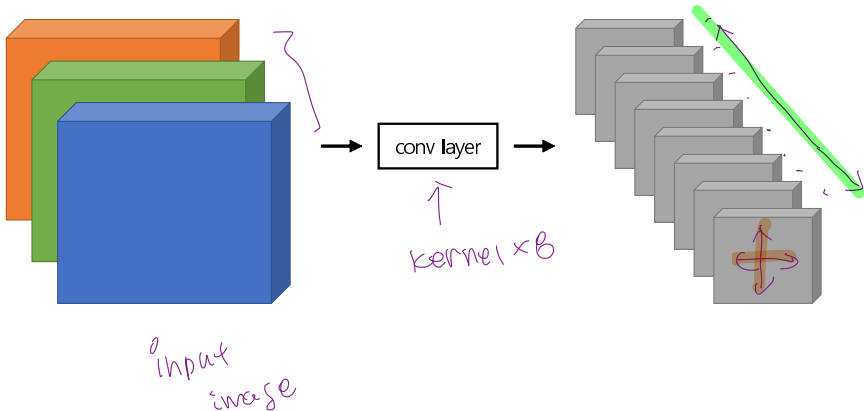
☆ 저장 99 인용 118285회 인용 관련 학술자료 전체 68개의 버전 EndNote로 가져오기



[PDF] thecvf.com

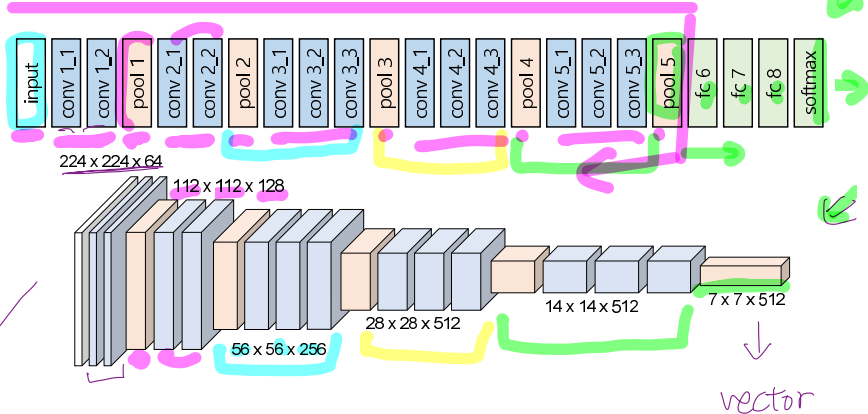
CNN 모델의 feature map

- CNN에서 레이어가 깊어질수록 채널의 수가 많아지고 너비와 높이는 줄어듦
- 서로 다른 kernel들은 적절한 feature를 추출하도록 학습



VGG-19

- 작은 크기(3×3)의 kernel로 깊이를 늘려 우수한 성능을 획득



CNN for deep layers

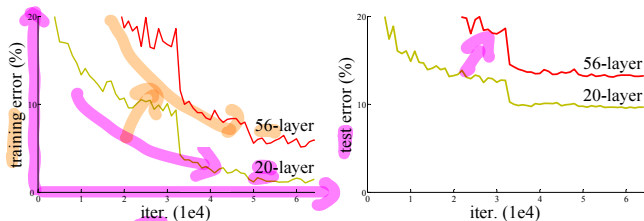


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “plain” networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

CNN: layer $\uparrow$, $\frac{W}{H}$ $\uparrow$, $\frac{C}{D}$ $\uparrow$

NN.

$$x \rightarrow W_1^T x \rightarrow \sigma(W_1^T x) \rightarrow W_2^T \sigma(W_1^T x) \\ \rightarrow \sigma(W_2^T \sigma(W_1^T x)) \dots$$

$$\dots \sigma(W^T \sigma(W^T \sigma(W^T \sigma(W^T x)))) = y \rightarrow \textcircled{E}$$

$$\textcircled{\frac{\partial E}{\partial W}}$$

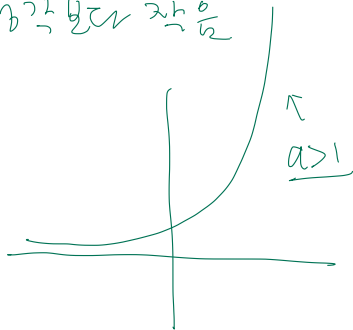
gradient의 연속적인 공.

1.00 ¹⁰ $\leftarrow$ layer $\uparrow$ $\Rightarrow 1$

1.01 ¹⁰ $\Rightarrow$ 생각보다 큼

0.99 ¹⁰ $\Rightarrow$ 생각보다 작음

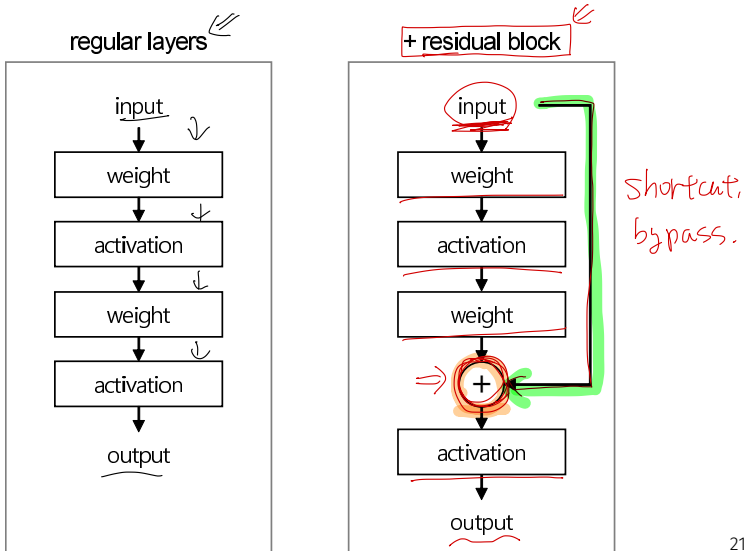
a^x



⁶ exploding / vanishing gradient problem. ⁹

Residual block

- 최적화를 용이하게 만들어주는 residual block

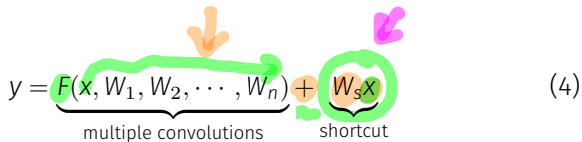


Residual block

- Regular layers

$$y = F(x) = \sigma_n(W_n \cdots \sigma_2(W_2 \sigma_1(W_1 x))) \quad (3)$$

- Residual block



The diagram shows the equation $y = F(x, W_1, W_2, \dots, W_n) + W_s x$ with several annotations. A green wavy line underlines the function F and its arguments, with an orange arrow pointing down to it. A green circle highlights the term $W_s x$, with a pink arrow pointing down to it. A bracket under the first part of the equation is labeled "multiple convolutions", and a bracket under the second part is labeled "shortcut".

$$y = \underbrace{F(x, W_1, W_2, \dots, W_n)}_{\text{multiple convolutions}} + \underbrace{W_s x}_{\text{shortcut}} \quad (4)$$

ImageNet result

plain-18,34

ResNet-18,34

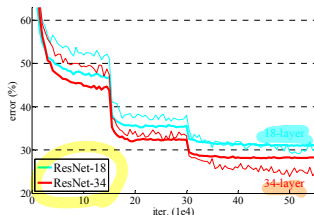
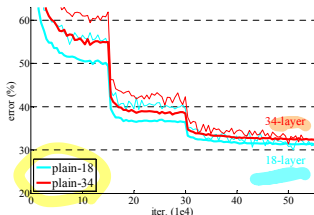


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

ImageNet result

Top-1 validation error [%]

하루고지.

	plain	ResNet
18 layers	<u>27.94</u>	<u>27.88</u>
34 layers	<u>28.54</u>	<u>25.03</u>

Validation error [%] of single model

하루

5월.

method	top-1	top-5
VGG (ILSVRC' 14)	-	8.43
GoogLeNet (ILSVRC' 14)	-	7.89
VGG (v5)	24.4	7.1
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