

# | Deep Residual Learning for Image Recognition |

보아즈 분석 23기

**김동환 김윤희**

2024.09.26

# 목 차

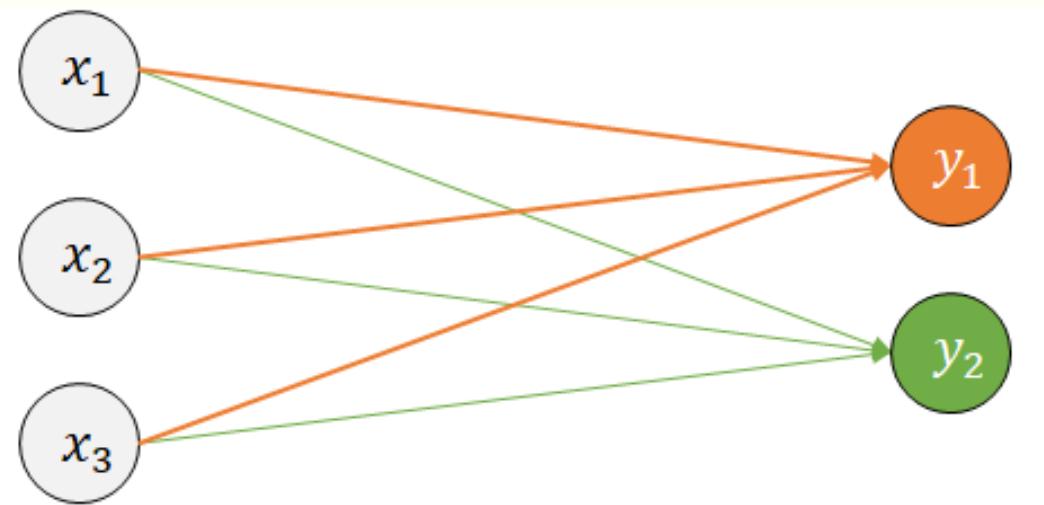
0  
**Abstract**

1  
**Introduction**

2  
**Related Work**

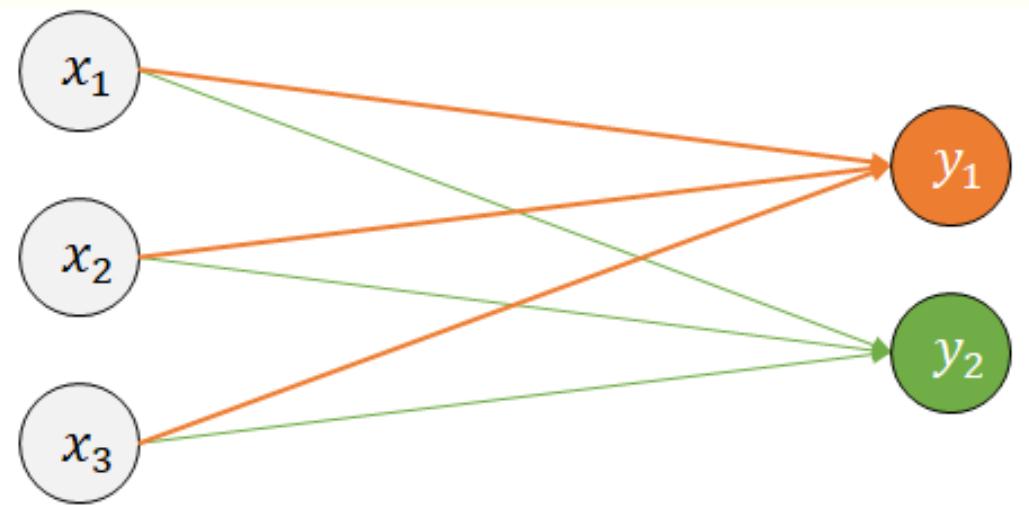
3  
**Deep Residual Learning**

4  
**Experiments**



$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \times \begin{bmatrix} w_1 & w_4 \\ w_2 & w_5 \\ w_3 & w_6 \end{bmatrix} + \begin{bmatrix} b_1 & b_2 \end{bmatrix} = \begin{bmatrix} y_1 & y_2 \end{bmatrix}$$

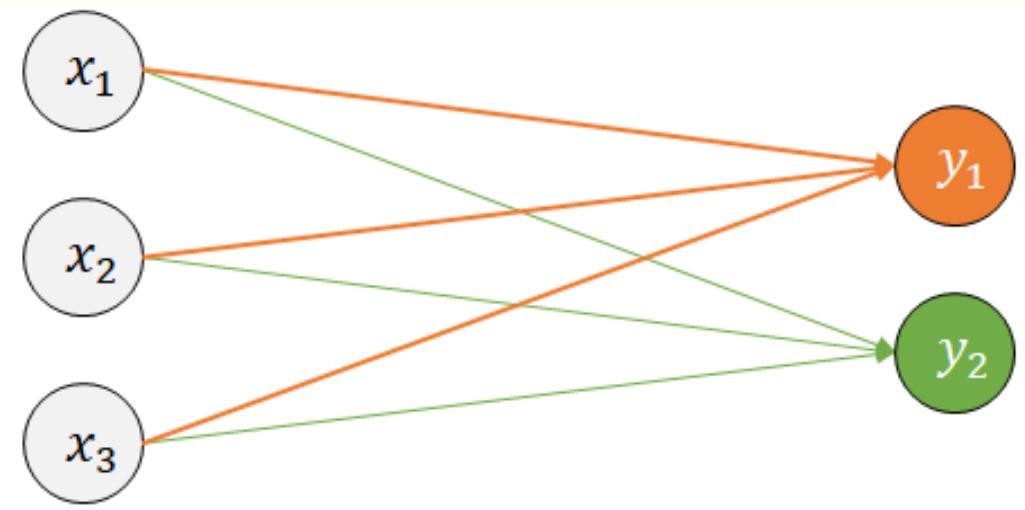
딥러닝의 동작 원리



$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \times \begin{bmatrix} w_1 & w_4 \\ w_2 & w_5 \\ w_3 & w_6 \end{bmatrix} + \begin{bmatrix} b_1 & b_2 \end{bmatrix} = \begin{bmatrix} y_1 & y_2 \end{bmatrix}$$

행렬 곱

딥러닝의 동작 원리



$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \times \begin{bmatrix} w_1 & w_4 \\ w_2 & w_5 \\ w_3 & w_6 \end{bmatrix} + \begin{bmatrix} b_1 & b_2 \end{bmatrix} = \begin{bmatrix} y_1 & y_2 \end{bmatrix}$$

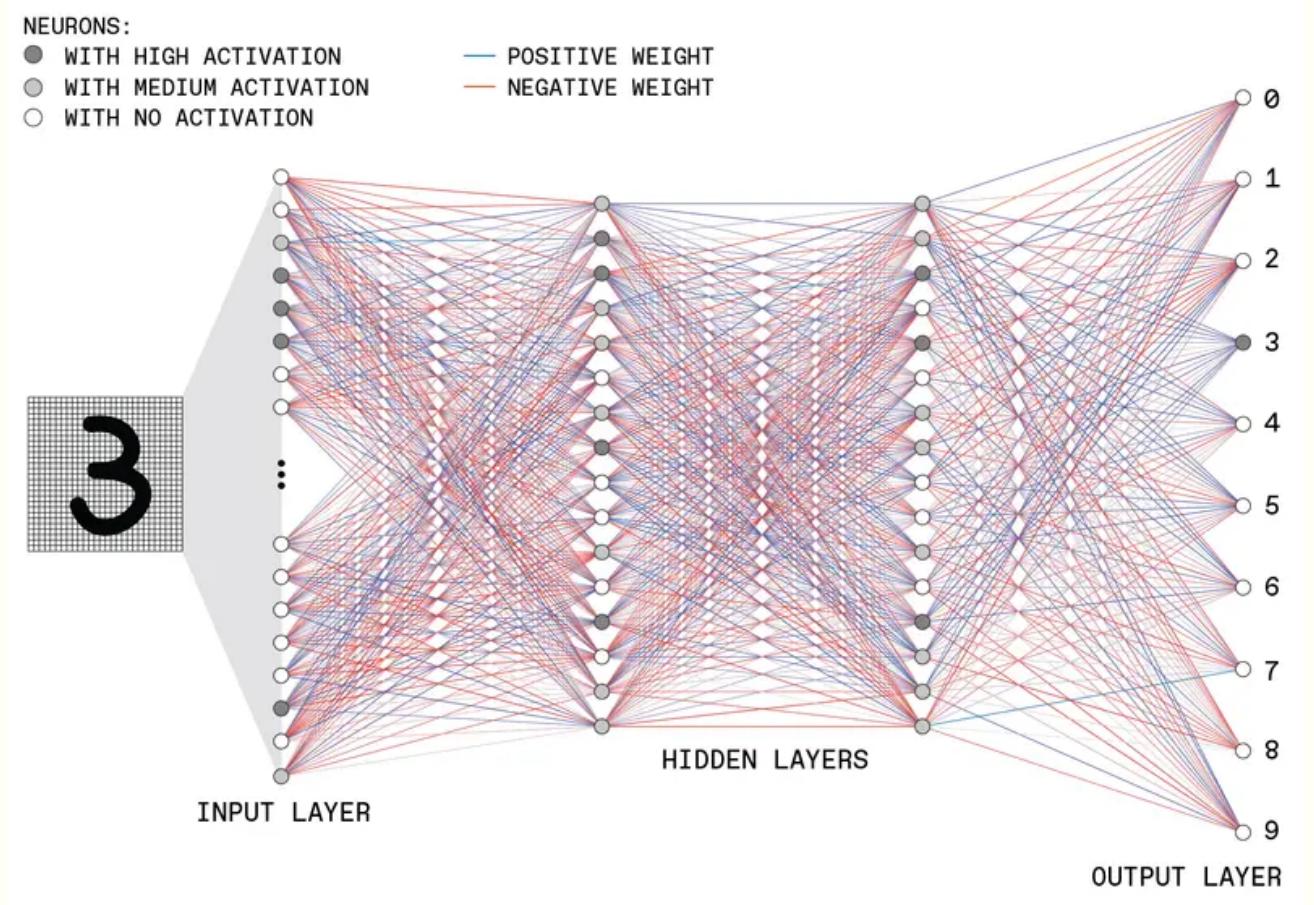
행렬 곱

force      mass      acceleration

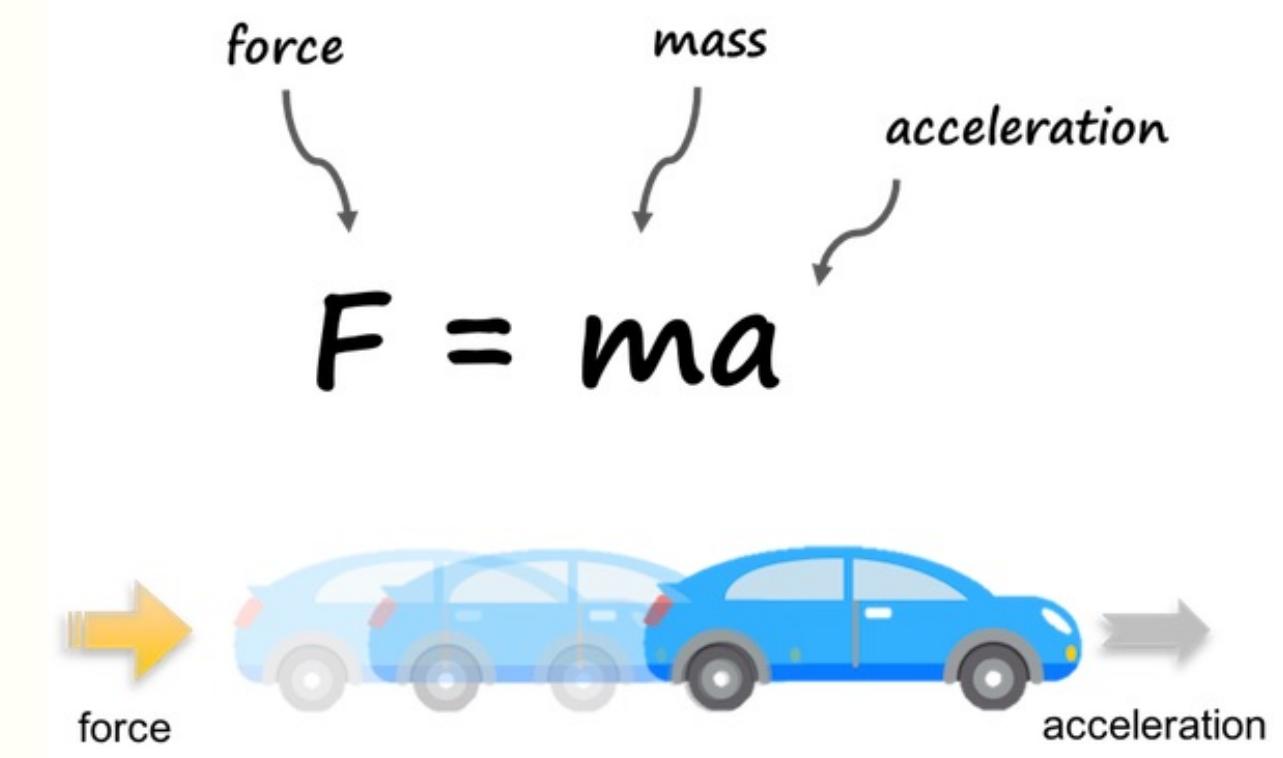
$$F = ma$$



의미



행렬 곱

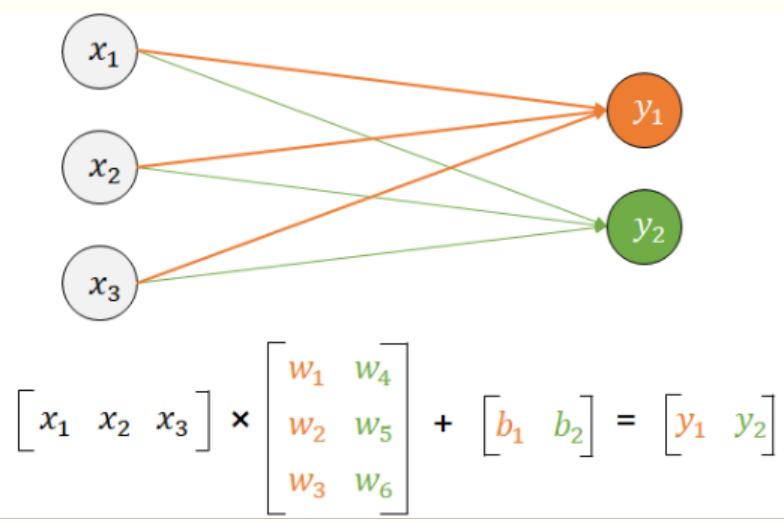


의미

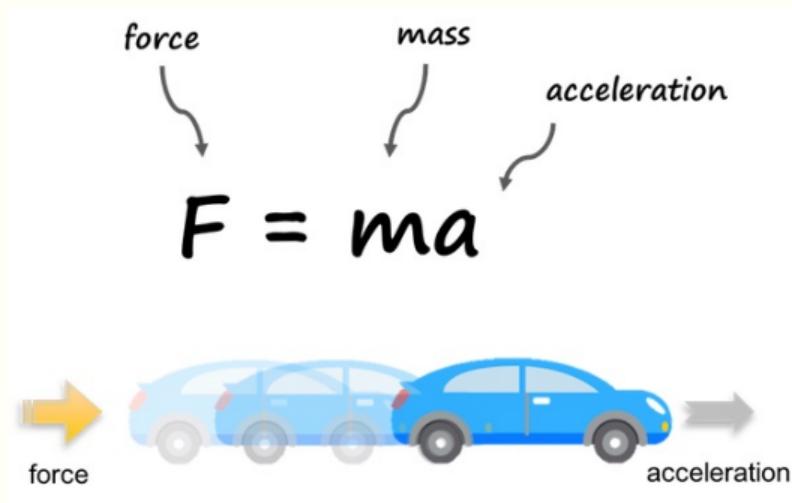
# 꽃 / 김춘수

내가 그의 이름을 불러 주기 전에는  
그는 다만  
하나의 꽃짓에 지나지 않았다.

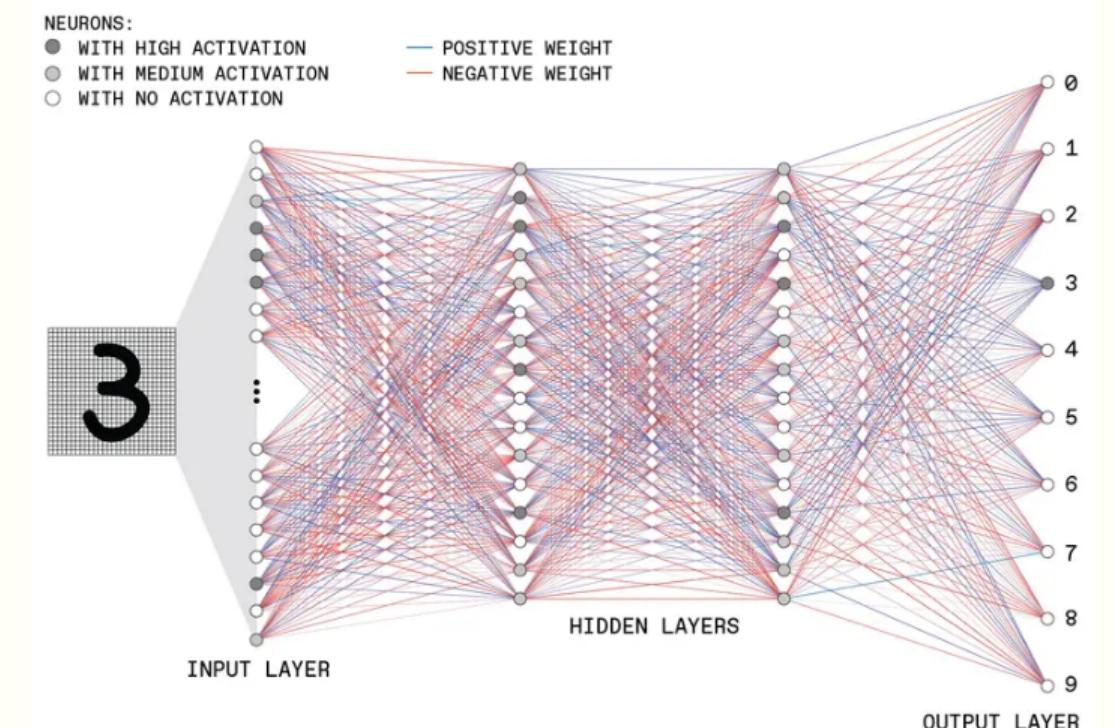
내가 그의 이름을 불러 주었을 때  
그는 나에게로 와서  
꽃이 되었다.



행렬 곱



의미



Deep Residual Learning  
for Image Recognition

---

생각이 많으면 좋은 걸까

생각이 너무 많으면 성능 저하

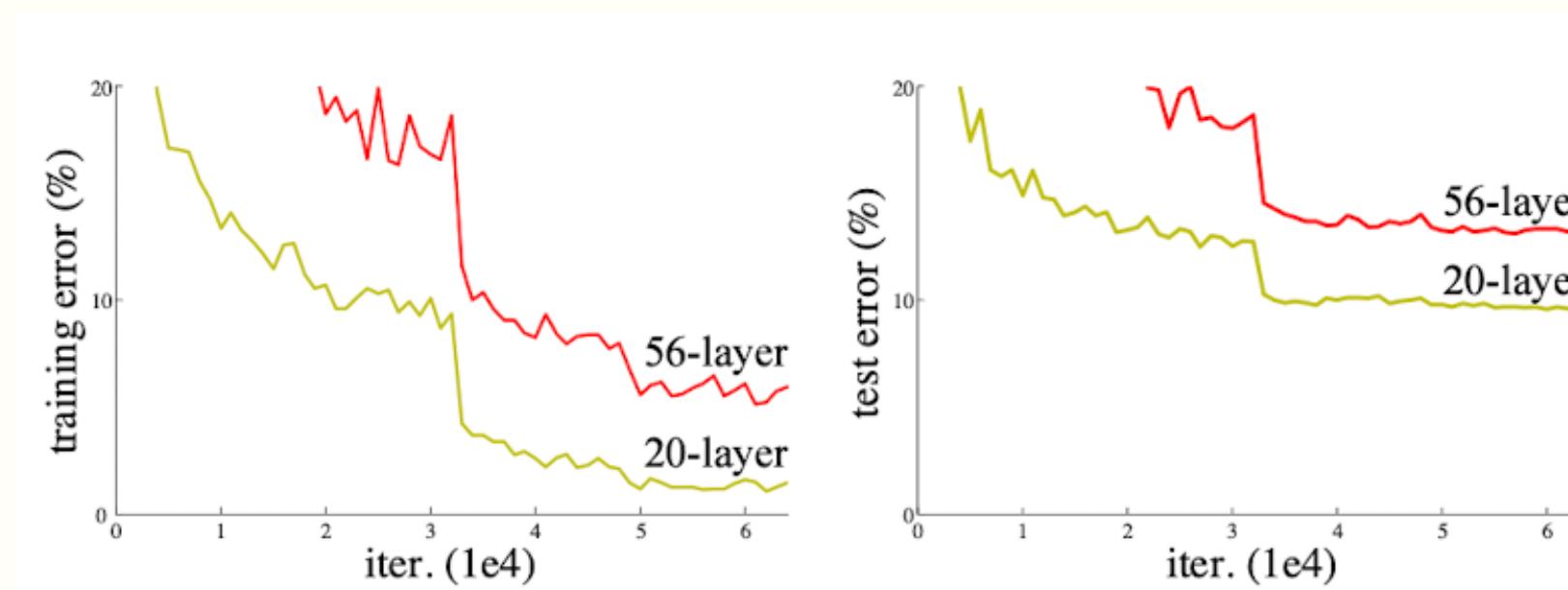
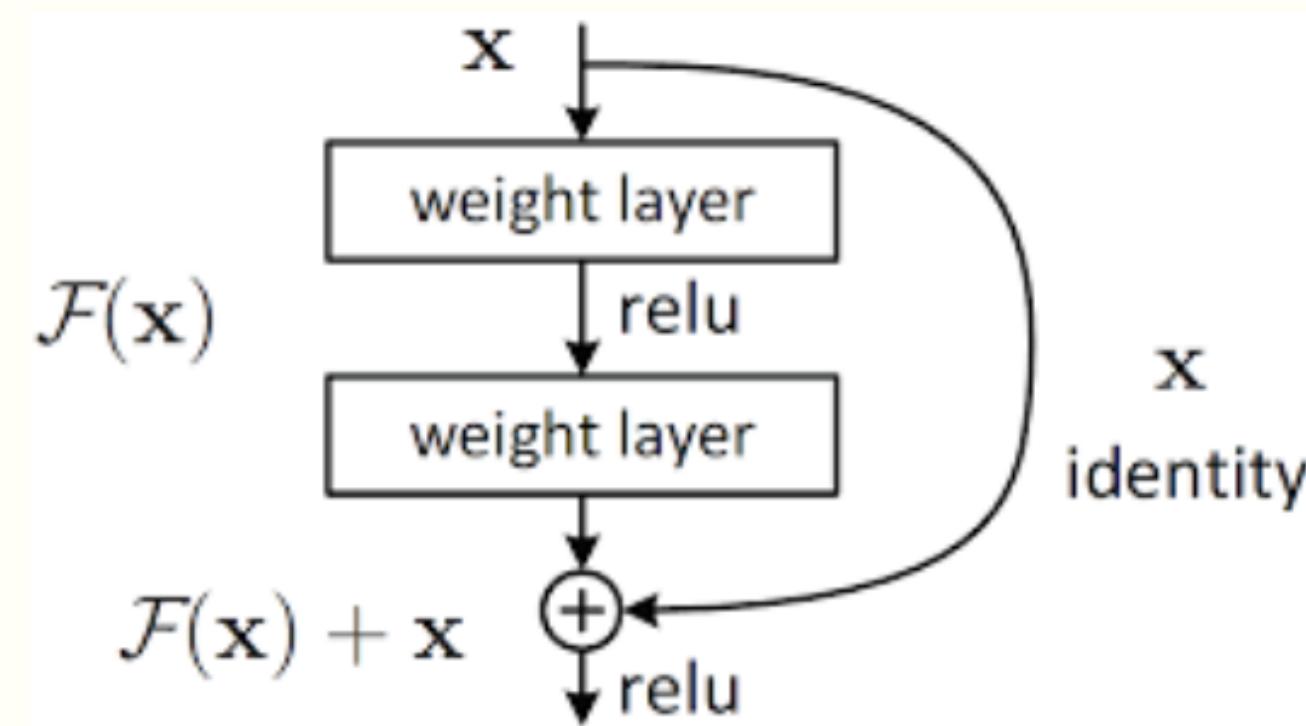


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “plain” networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

---

때에 따라서는  
단순하게 생각하는게 도움이 된다

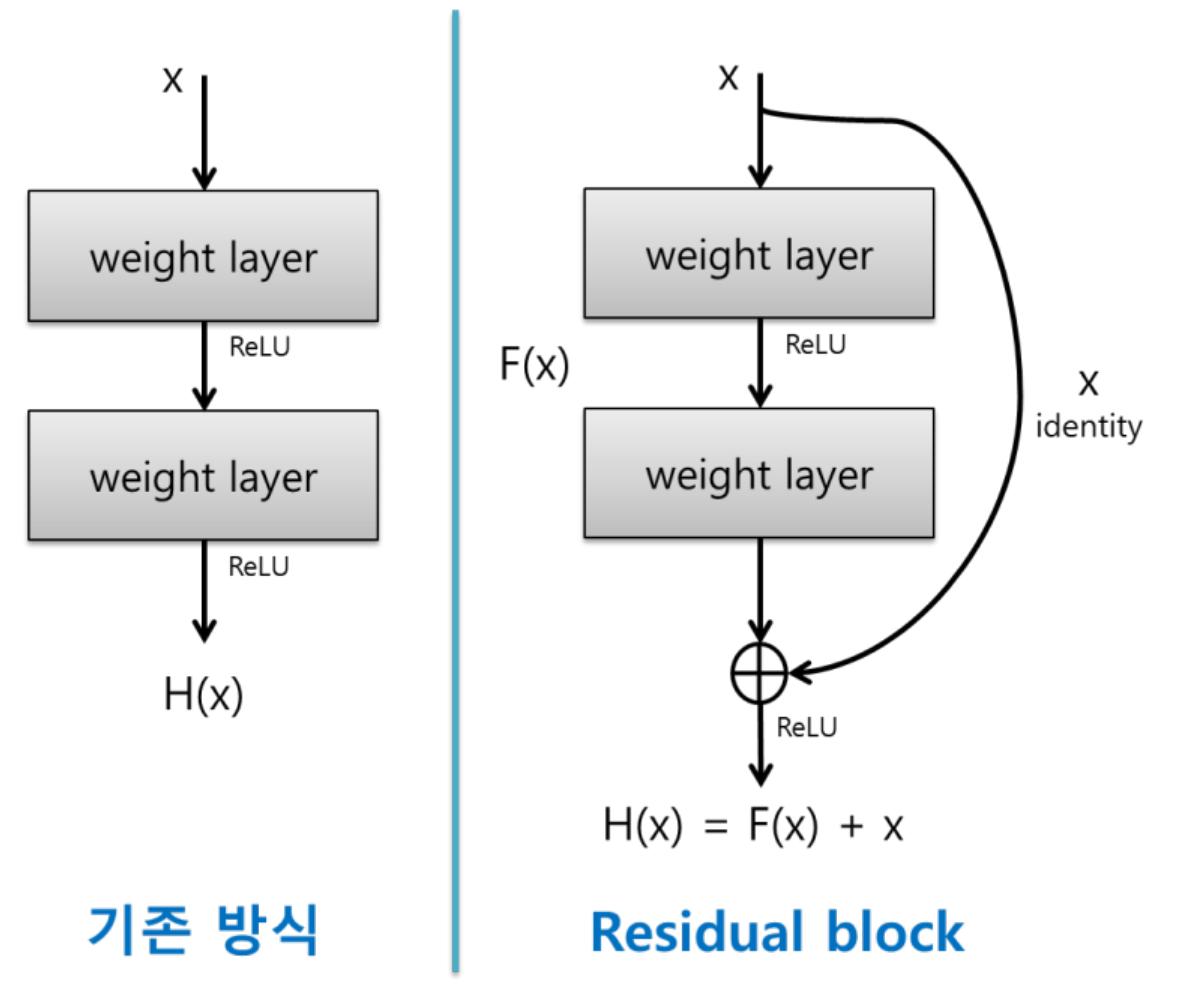
## 단순하게 생각하기



복잡한 함수  $F$  보다는, "잔차"를 학습하도록

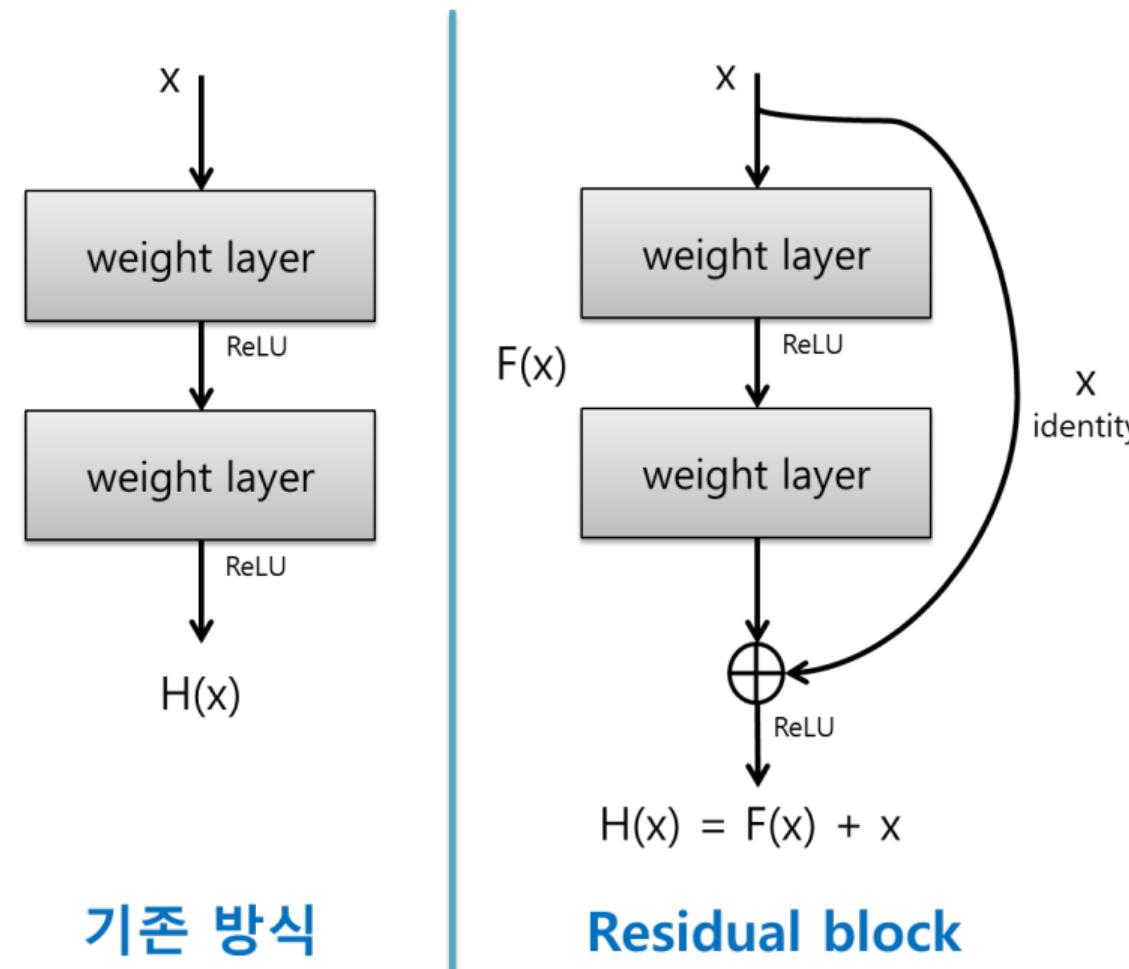
# Deep Residual Learning

## Residual Learning



# Deep Residual Learning

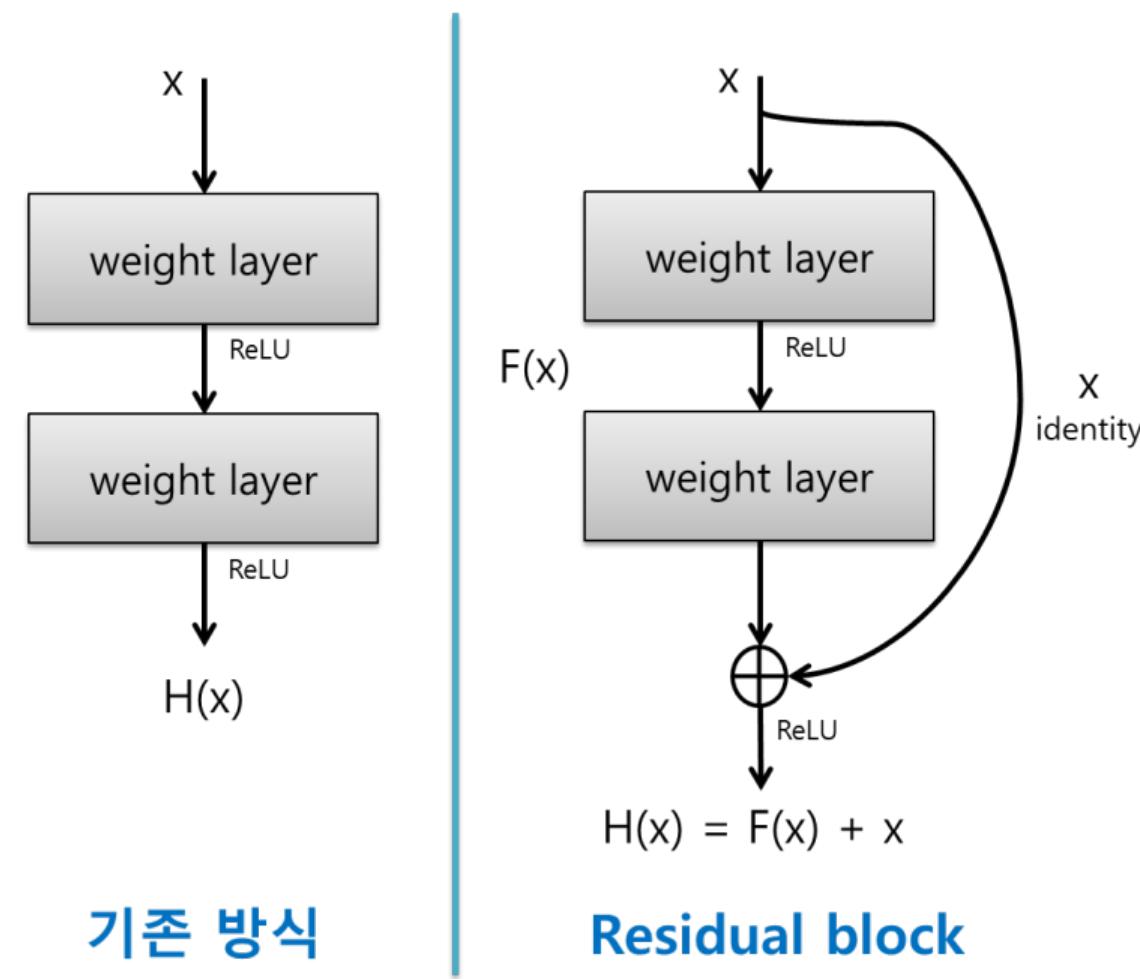
## Residual Learning



만일 Identity Mapping이 최적이라면,  
단순히  $F(x)=0$  이면 됨

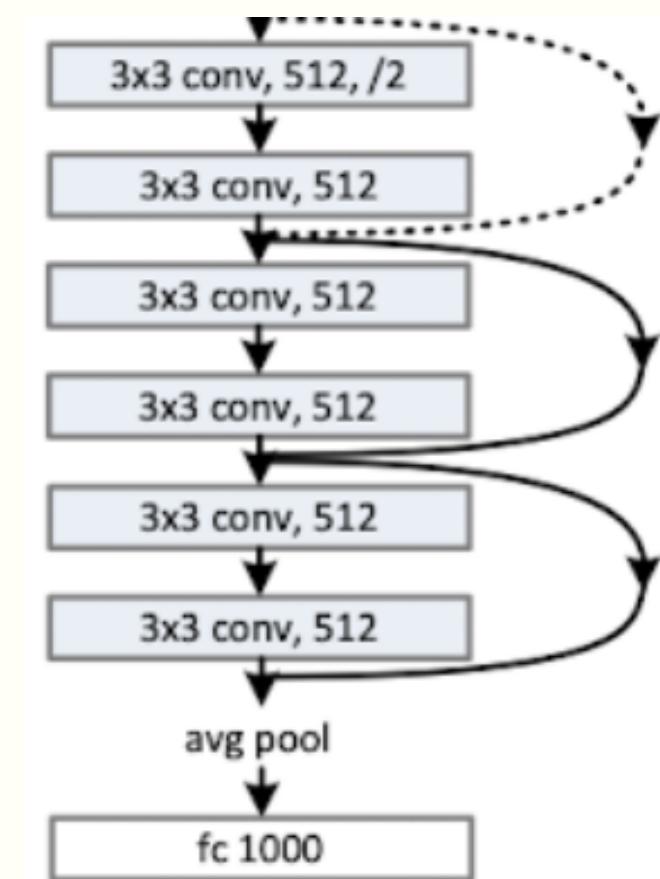
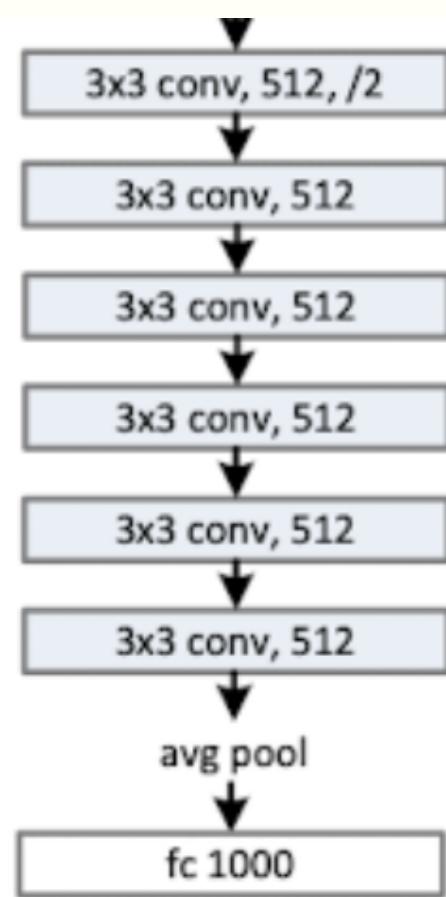
# Deep Residual Learning

## Residual Learning

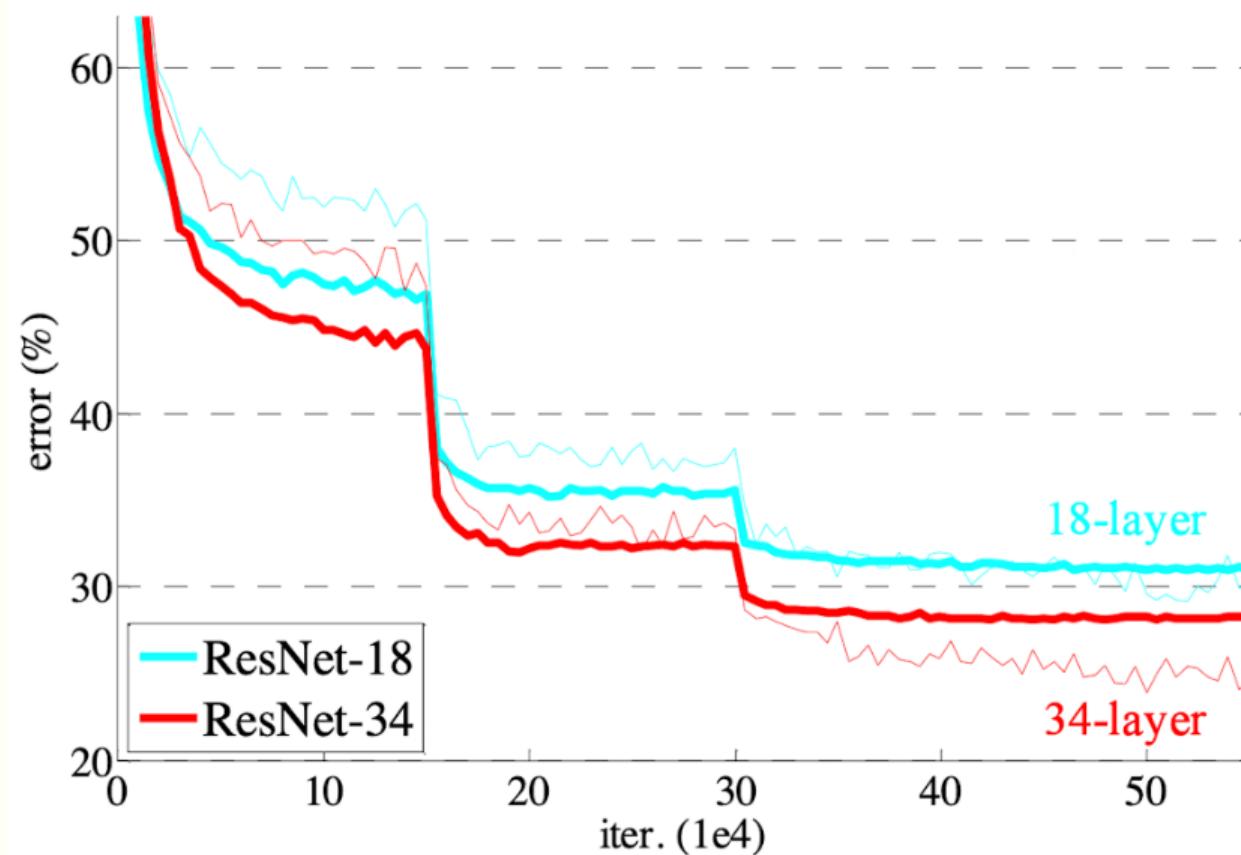


$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}.$$

# 단순하게 생각하기

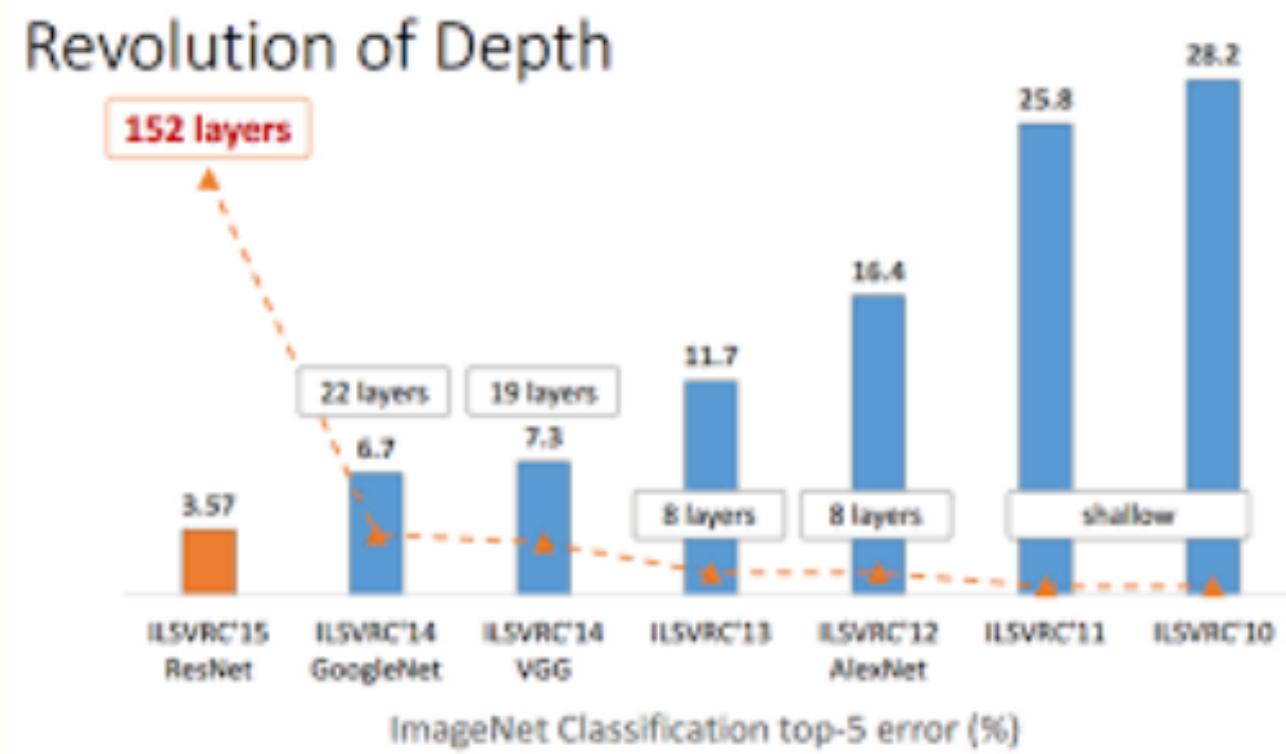


효과



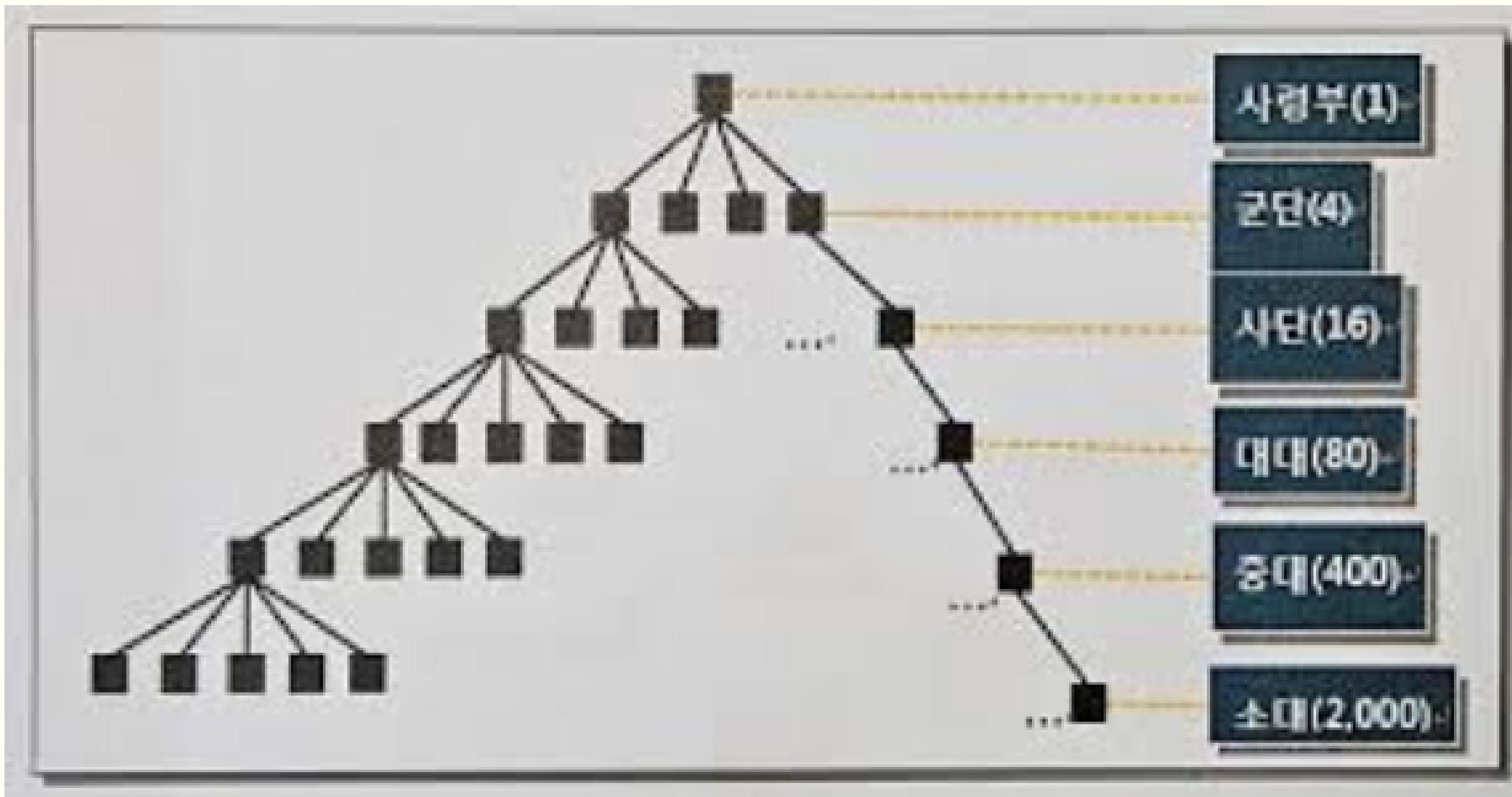
Deep Residual Learning  
for Image Recognition

효과



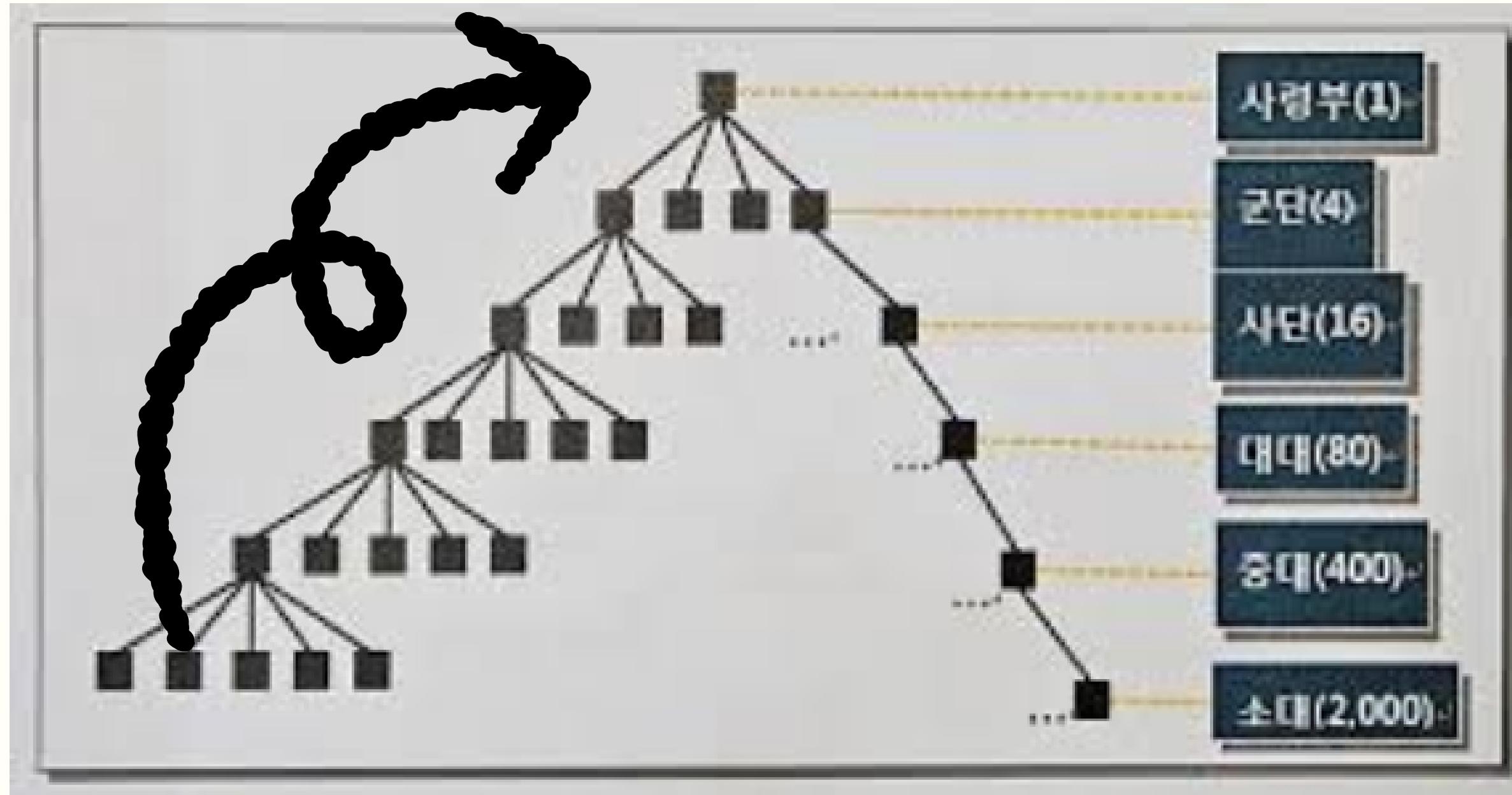
Deep Residual Learning  
for Image Recognition

# 기울기 소실 완화



Deep Residual Learning  
for Image Recognition

# 기울기 소실 완화



Deep Residual Learning  
for Image Recognition

Question on "Identity Mapping by Shortcuts" in Deep Residual Learning for Image Recognition

✉ 김동환 <forwarder1121@konkuk.ac.kr>

받는 사람: kaiming@mit.edu

ResNet.jpeg  
991.5 KB

다운로드 · 미리 보기

Dear Professor Kaiming He,

My name is Dong Hwan Kim, and I am currently studying at Konkuk University in South Korea. I have recently started studying computer vision, and your paper *Deep Residual Learning for Image Recognition* has been incredibly helpful. I was particularly impressed by the way **residual connections** provide the model with a "choice." I understood that allowing the model to decide between learning a simple **identity mapping** or a more complex **F function** is an effective method for training deep neural networks.

Additionally, while reading the paper, I found an interesting analogy between this approach and the structure of large corporations. Large organizations often have complex hierarchical structures, making it difficult for lower-level employees to communicate their ideas to upper management. To resolve this, I believe lower-level employees should be given the opportunity to communicate directly with senior executives. In a similar way, I understood your **residual connections** as providing this "shortcut" to bypass the complexity of deeper layers, allowing information to flow more directly.

Now, I would like to ask you about a specific part of the paper, particularly in **section 3.2, "Identity Mapping by Shortcuts."** In this section, you discuss how **identity mapping** is used when the input and output dimensions are the same, and how **1×1 convolutions** are used for **projection** when the dimensions differ. However, earlier in the paper, you mention that when  $F(x)$  is a single layer,  $y=W_1x+x$  is **similar to a linear layer**. My understanding is that if a **ReLU** or another non-linear activation function is applied, this would become a **non-linear layer**. I was wondering if this section assumes that there is no activation function involved, or if I might be misunderstanding something.

I have attached the part of the paper I am curious about for your reference. I would greatly appreciate your response whenever you have time, as I am eager to deepen my understanding of this concept.

Thank you very much for your time.

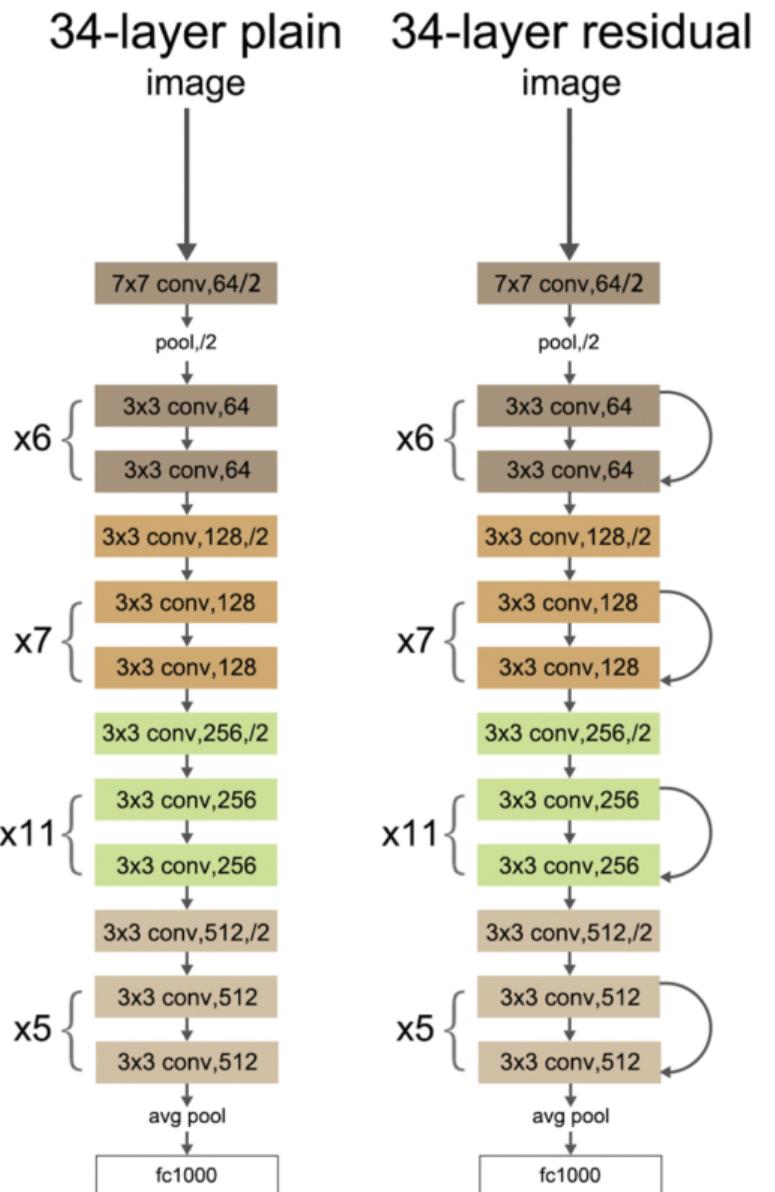
Best regards,  
Dong Hwan Kim  
Konkuk University, South Korea  
[forwarder1121@konkuk.ac.kr](mailto:forwarder1121@konkuk.ac.kr)

# Deep Residual Learning for Image Recognition

# I ResNet Experiments

보아즈 분석 23기  
**김동환 김윤희**

# ImageNet Classification



layer name	output size	18-layer	34-layer
conv1	112×112		
conv2_x	56×56	$\left[ \begin{smallmatrix} 3\times 3, 64 \\ 3\times 3, 64 \end{smallmatrix} \right] \times 2$	$\left[ \begin{smallmatrix} 3\times 3, 64 \\ 3\times 3, 64 \end{smallmatrix} \right] \times 3$
conv3_x	28×28	$\left[ \begin{smallmatrix} 3\times 3, 128 \\ 3\times 3, 128 \end{smallmatrix} \right] \times 2$	$\left[ \begin{smallmatrix} 3\times 3, 128 \\ 3\times 3, 128 \end{smallmatrix} \right] \times 4$
conv4_x	14×14	$\left[ \begin{smallmatrix} 3\times 3, 256 \\ 3\times 3, 256 \end{smallmatrix} \right] \times 2$	$\left[ \begin{smallmatrix} 3\times 3, 256 \\ 3\times 3, 256 \end{smallmatrix} \right] \times 6$
conv5_x	7×7	$\left[ \begin{smallmatrix} 3\times 3, 512 \\ 3\times 3, 512 \end{smallmatrix} \right] \times 2$	$\left[ \begin{smallmatrix} 3\times 3, 512 \\ 3\times 3, 512 \end{smallmatrix} \right] \times 3$
	1×1		aver
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$

## 연구 방법

- VGG-19는 convolution layer와 pooling layer로 구성된 전통적인 네트워크로, residual learning을 사용하지 않음
- 34-layer plain 네트워크는 일반적인 네트워크로, shortcut connection이 없고 단순히 레이어를 많이 쌓은 형태
- ResNet-34는 residual learning을 통해 깊은 네트워크에서도 효과적으로 학습할 수 있도록 shortcut connection으로 입력을 직접 더하는 방식으로 만듦

# ImageNet Classification

## ImageNet 2012 분류 데이터셋을 사용



n02097047 (196)



n01682714 (40)



n03134739 (522)



n04254777 (806)



n02859443 (449)



n02096177 (192)



n02107683 (239)



n01443537 (1)



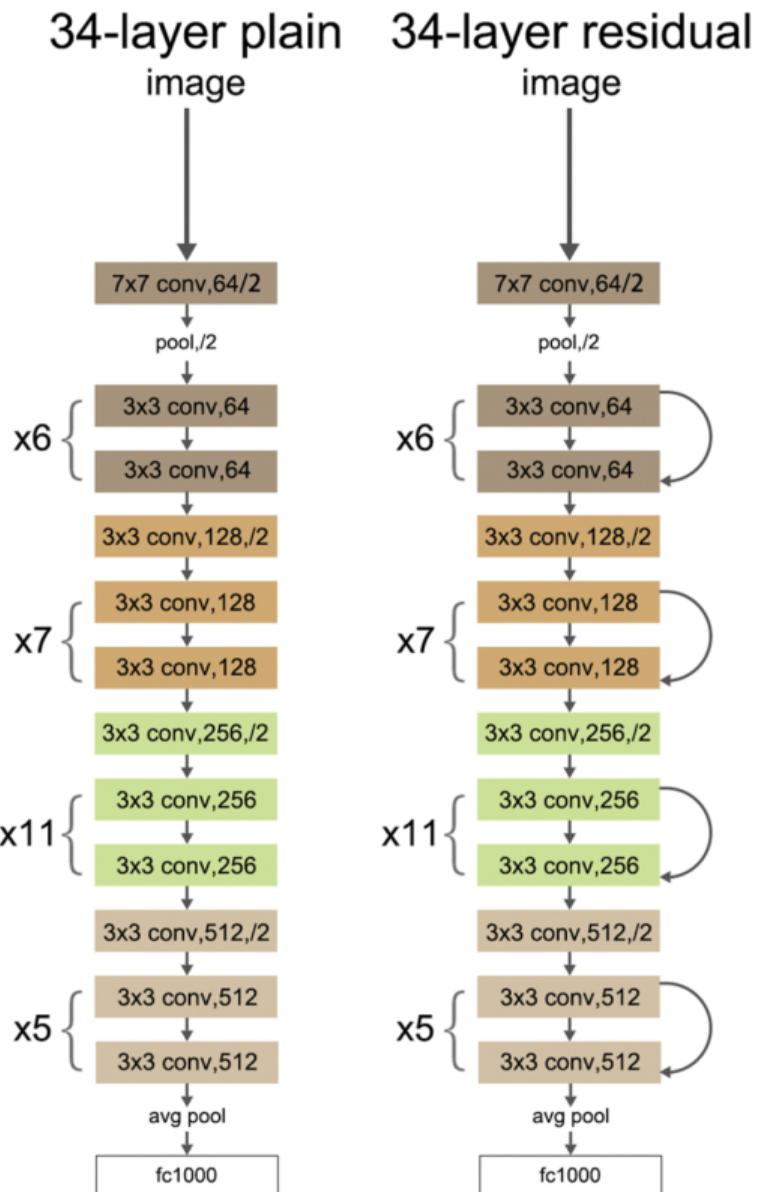
n02264363 (318)

## 연구 방법

- 1000개의 클래스
- 128만 개의 training 이미지
- 5만 개의 validation 이미지
- 10만 개의 test 이미지
- 성능 평가는 top-1 error rate와 top-5 error rate를 사용

Deep Residual Learning  
for Image Recognition

# ImageNet Classification



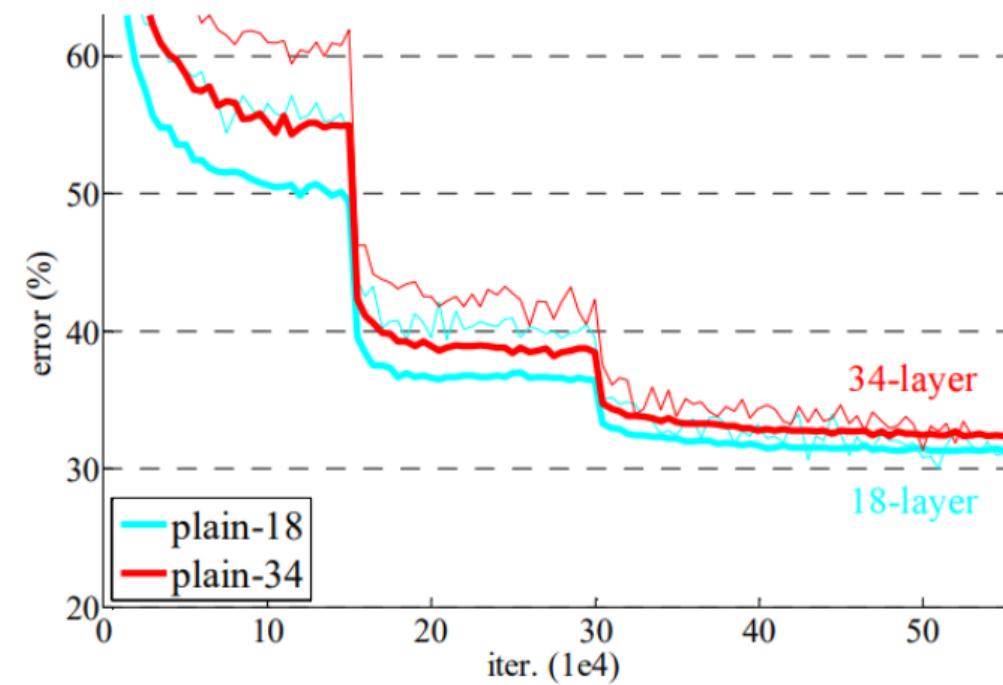
layer name	output size	18-layer	34-layer
conv1	112×112		
conv2_x	56×56	$\left[ \begin{smallmatrix} 3\times 3, 64 \\ 3\times 3, 64 \end{smallmatrix} \right] \times 2$	$\left[ \begin{smallmatrix} 3\times 3, 64 \\ 3\times 3, 64 \end{smallmatrix} \right] \times 3$
conv3_x	28×28	$\left[ \begin{smallmatrix} 3\times 3, 128 \\ 3\times 3, 128 \end{smallmatrix} \right] \times 2$	$\left[ \begin{smallmatrix} 3\times 3, 128 \\ 3\times 3, 128 \end{smallmatrix} \right] \times 4$
conv4_x	14×14	$\left[ \begin{smallmatrix} 3\times 3, 256 \\ 3\times 3, 256 \end{smallmatrix} \right] \times 2$	$\left[ \begin{smallmatrix} 3\times 3, 256 \\ 3\times 3, 256 \end{smallmatrix} \right] \times 6$
conv5_x	7×7	$\left[ \begin{smallmatrix} 3\times 3, 512 \\ 3\times 3, 512 \end{smallmatrix} \right] \times 2$	$\left[ \begin{smallmatrix} 3\times 3, 512 \\ 3\times 3, 512 \end{smallmatrix} \right] \times 3$
	1×1		aver
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$

## 연구 방법

- VGG-19는 convolution layer와 pooling layer로 구성된 전통적인 네트워크로, residual learning을 사용하지 않음
- 34-layer plain 네트워크는 일반적인 네트워크로, shortcut connection이 없고 단순히 레이어를 많이 쌓은 형태
- ResNet-34는 residual learning을 통해 깊은 네트워크에서도 효과적으로 학습할 수 있도록 shortcut connection으로 입력을 직접 더하는 방식으로 만듦

# ImageNet Classification

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

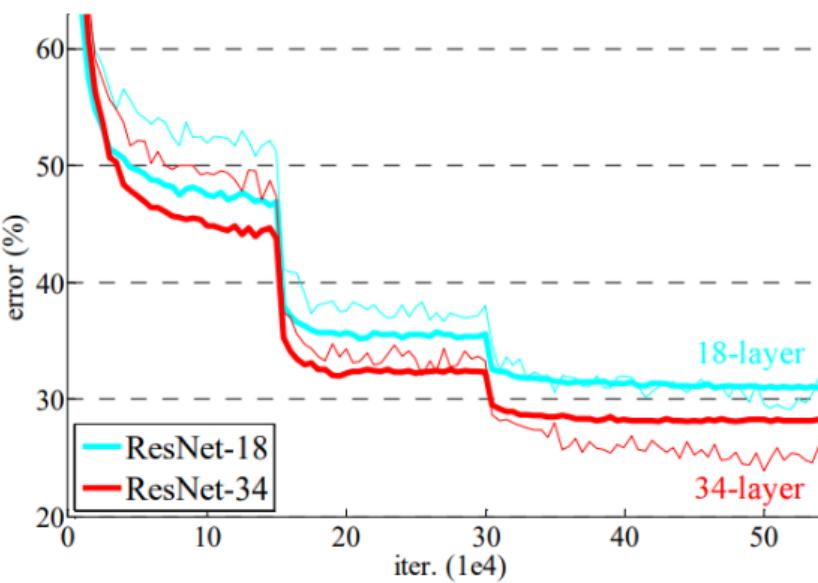
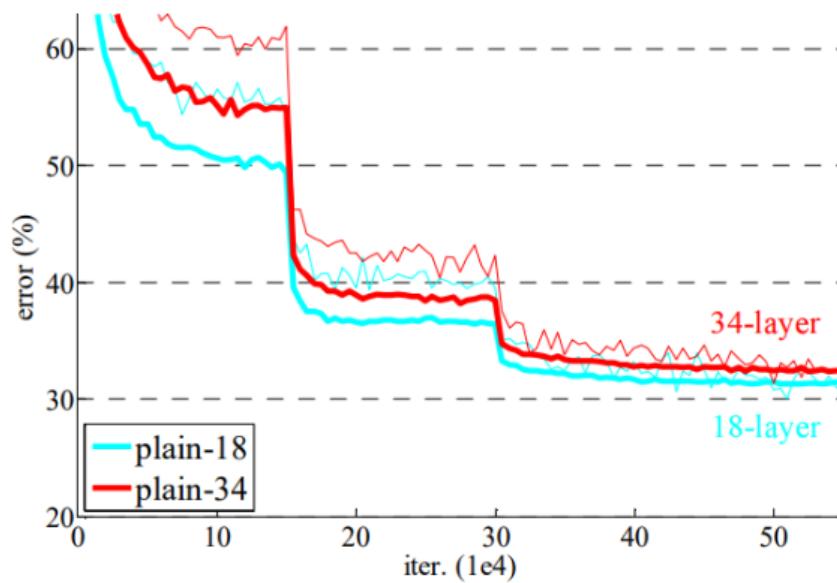


## Plain Networks 연구 결과

- 더 깊은 34 layer plain 네트워크가 18-layer plain 네트워크 보다 검증 오류가 더 높음
- 훈련 과정 동안 성능 저하 문제(degradation problem) 확인
- 34-layer 네트워크의 solution space가 18-layer 네트워크보다 더 크게 갖지만, 최적의 답을 찾는 것이 더 어렵다는 점을 알 수 있음
- 실험에서 사용한 네트워크들은 Batch Normalization를 사용했으며, 어느 정도의 정확도를 달성했으므로, 기울기 소실 문제가 아닌, 수렴 속도가 느려지는 등의 다른 최적화 문제의 가능성성이 존재

# ImageNet Classification

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



## Residual Networks 연구 결과

- 18-layer와 비교하였을 때 34-layer ResNet이 훨씬 낮은 훈련 오류
- plain 네트워크와 비교할 때 top-1 error rate를 3.5% 감소시킨 것으로 깊이가 증가함에 따라 정확도 향상이 가능하다는 것을 보여줌
- 18-layer plain/residual 네트워크는 비슷한 정확도를 보이지만(표), 18-레이어 ResNet은 더 빠르게 수렴(그래프)

# ImageNet Classification

model	top-1 err.	top-5 err.
VGG-16 [41]	28.07	9.33
GoogLeNet [44]	-	9.15
PReLU-net [13]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	<b>21.43</b>	<b>5.71</b>

## Identity vs. Projection Shortcuts 연구 결과

- (A)는 identity shortcuts을 사용하며, 차원이 변할 때만 0으로 패딩하여 처리하는 방식
- (B)는 차원을 증가시키기 위해 projection shortcut을 사용하며, 다른 지름길은 identity shortcut
- (C)는 모든 지름길이 projection shortcuts

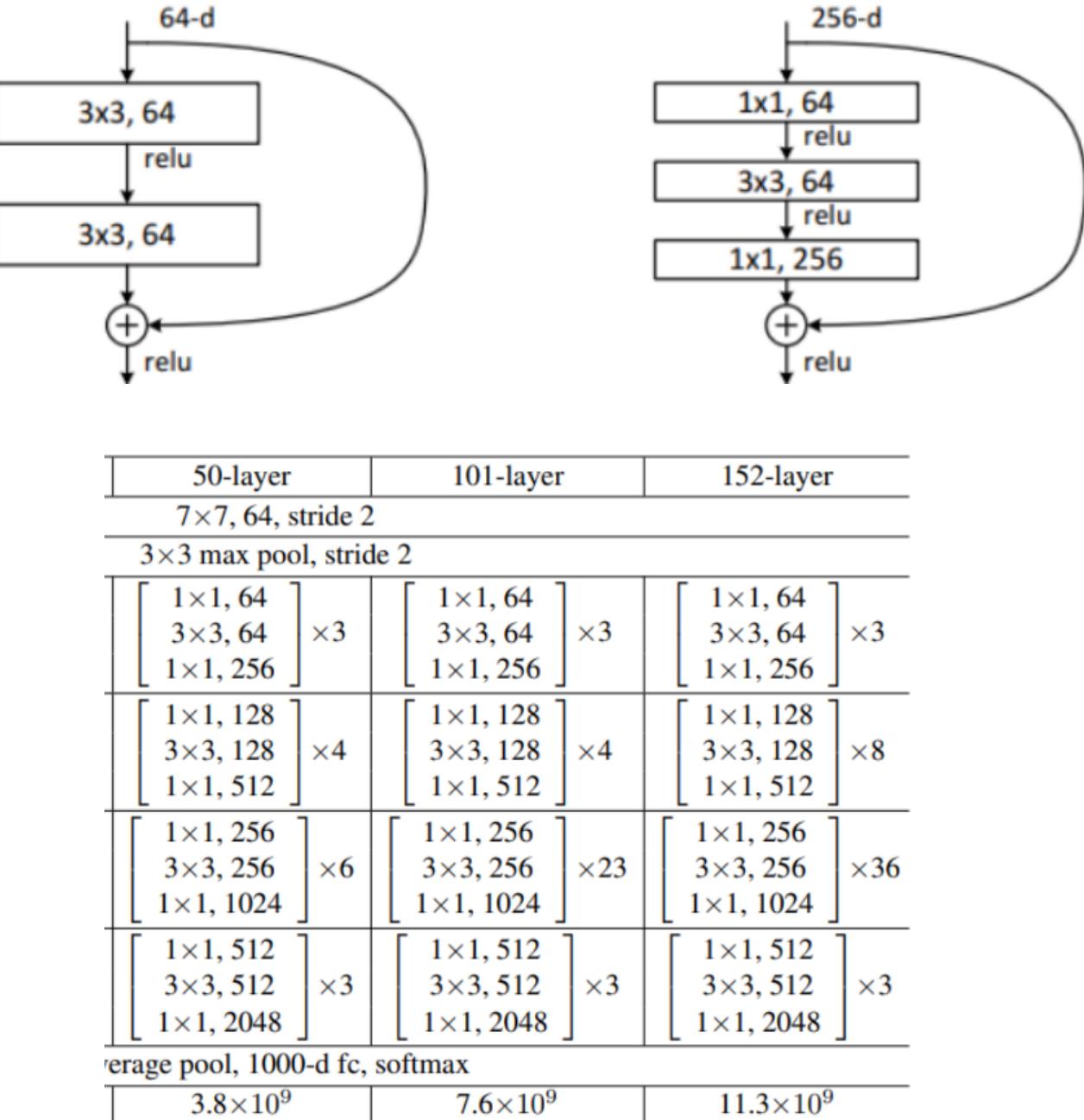
# ImageNet Classification

model	top-1 err.	top-5 err.
VGG-16 [41]	28.07	9.33
GoogLeNet [44]	-	9.15
PReLU-net [13]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	<b>21.43</b>	<b>5.71</b>

## Identity vs. Projection Shortcuts 연구 방법

- plain 네트워크보다 훨씬 더 우수한 성능을 보였음
- B가 A보다 약간 더 나은 이유는 A에서 0으로 패딩된 차원이 실제로 잔차 학습을 하지 않았기 때문으로 추정
- C는 B보다 약간 더 나은 성능을 보였으며, 이는 많은 projection shortcuts(13개)를 통해 추가 매개변수가 도입되었기 때문
- 그러나 A/B/C 간의 작은 차이는 projection shortcuts이 성능 저하 문제를 해결하는데 필수적이지 않다는 것으로 결론

# ImageNet Classification



## Deeper Bottleneck Architectures 연구

- 1×1 컨볼루션: 차원 축소
- 3×3 컨볼루션: 축소된 차원에서 feature map 추출
- 1×1 컨볼루션: 차원 확대
- identity shortcut 사용
- 기존의 34-layer 네트워크에서 bottleneck block으로 교체하고 옵션 B를 사용하여 차원을 증가시킨 결과 ResNet-50은 38억 FLOPs를 가짐
- ResNet-152는 깊이가 크게 증가했어도 11.3억 FLOPs로 VGG-16(15.3억 FLOPs)과 VGG-19(19.6억 FLOPs)보다 복잡도가 낮음

# ImageNet Classification

model	top-1 err.	top-5 err.
VGG-16 [41]	28.07	9.33
GoogLeNet [44]	-	9.15
PReLU-net [13]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	<b>21.43</b>	<b>5.71</b>

method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43 <sup>†</sup>
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	<b>5.71</b>
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	<b>19.38</b>	<b>4.49</b>

method	top-5 err. ( <b>test</b> )
VGG [41] (ILSVRC'14)	7.32
GoogLeNet [44] (ILSVRC'14)	6.66
VGG [41] (v5)	6.8
PReLU-net [13]	4.94
BN-inception [16]	4.82
<b>ResNet (ILSVRC'15)</b>	<b>3.57</b>

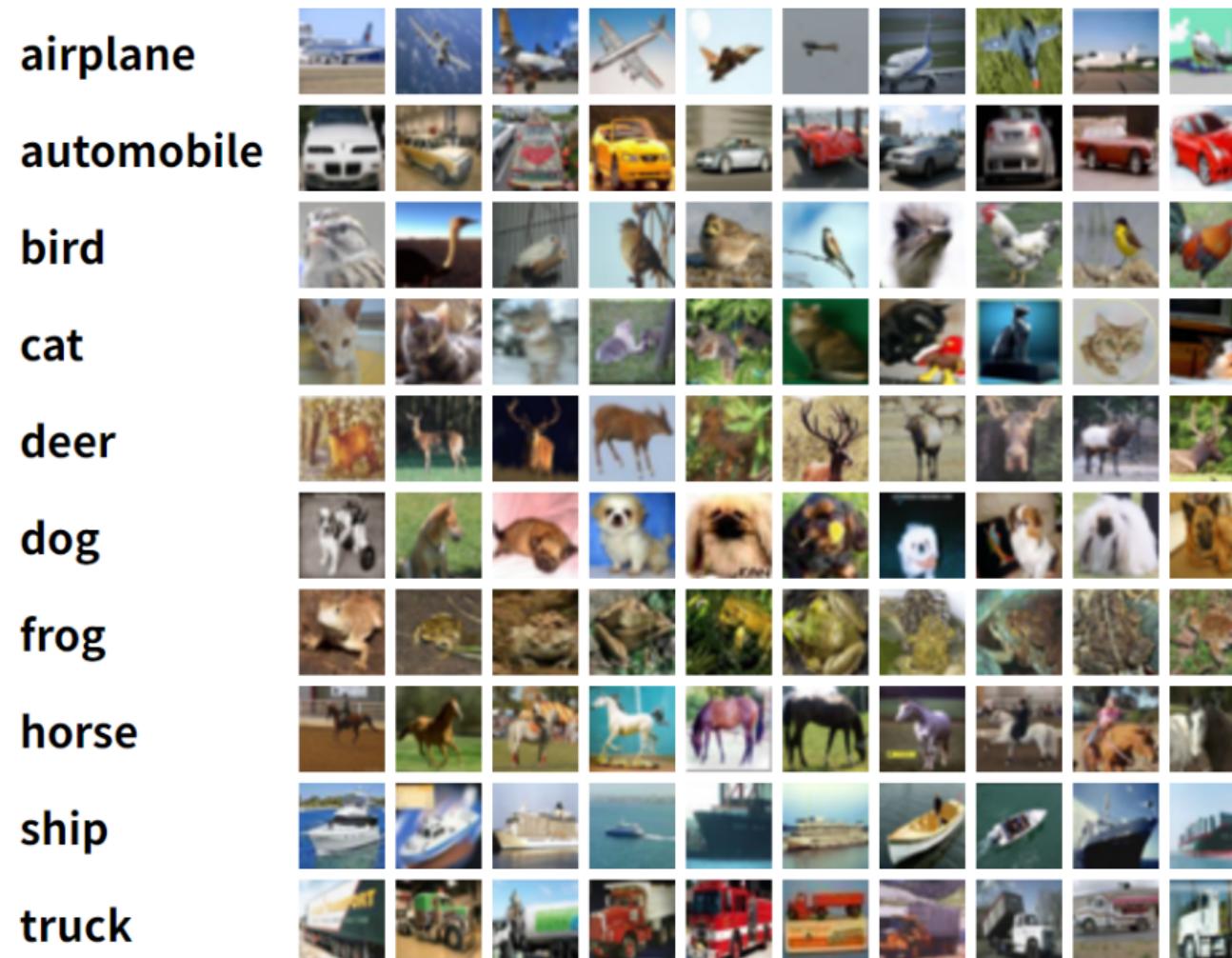
## Deeper Bottleneck Architectures 연구

- 50/101/152 layer ResNet들은 기존의 34-layer 네트워크보다 훨씬 더 정확했으며, 성능 저하 문제가 발생하지 않음
- 이전의 단일 모델 결과와 비교한 결과, ResNet-34는 매우 경쟁력 있는 정확도를 달성
- ResNet-152는 단일 모델 기준으로 top-5 error rate가 4.49%로, 이전의 모든 양상을 결과보다 우수한 성능을 보임
- 6개 모델을 결합한 양상을 구성했으며, 이 중 2개는 152층 모델을 사용했을 때 ImageNet 테스트에서 3.57%의 top-5 error rate를 달성했고, 이는 ILSVRC 2015에서 1위를 차지

# CIFAR-10 and Analysis

## CIFAR-10 dataset

Here are the classes in the dataset, as well as 10 random images from each:



## 연구 방법

- 32x32 픽셀의 10개의 클래스
- 5만 개의 training 이미지
- 1만 개의 test 이미지
- 훈련 세트에서 모델을 학습시키고 테스트 세트에서 평가
- 총  $6n+2$ 개의 레이어와 옵션 A 방식으로 shortcut connection 구성

output map size	$32 \times 32$	$16 \times 16$	$8 \times 8$
# layers	$1+2n$	$2n$	$2n$
# filters	16	32	64

# CIFAR-10 and Analysis

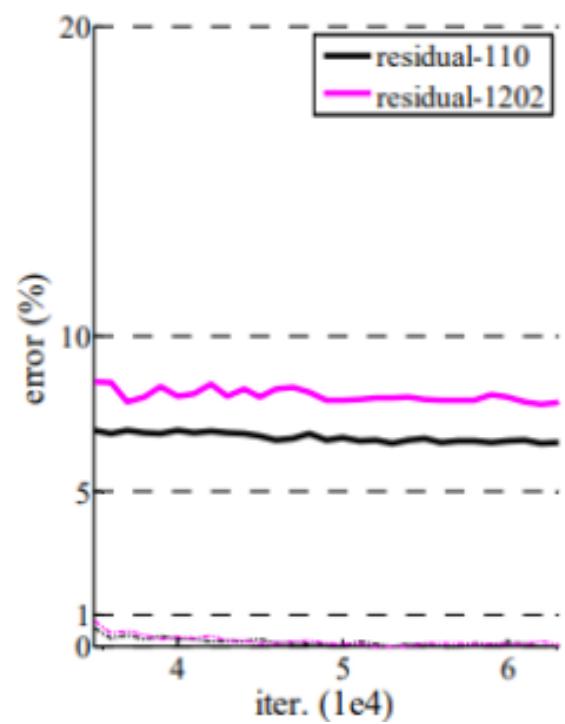
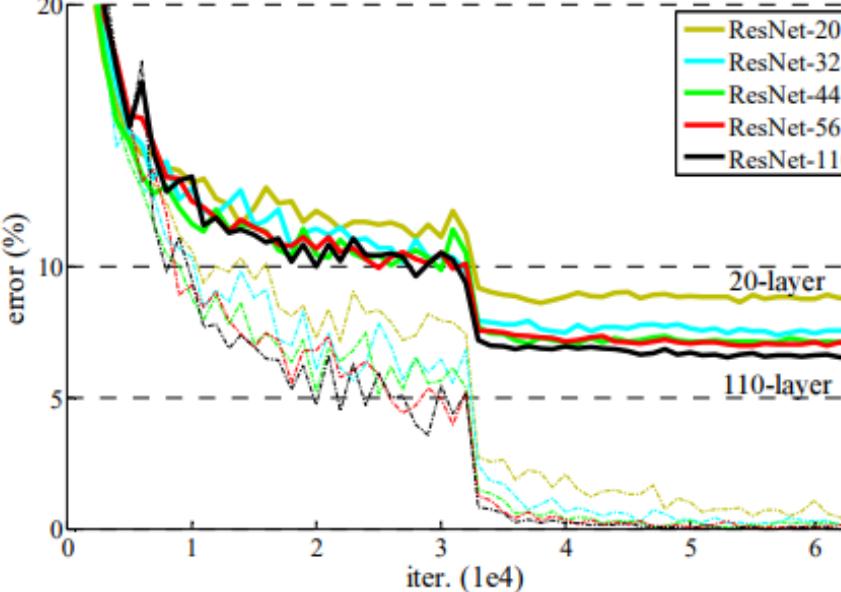
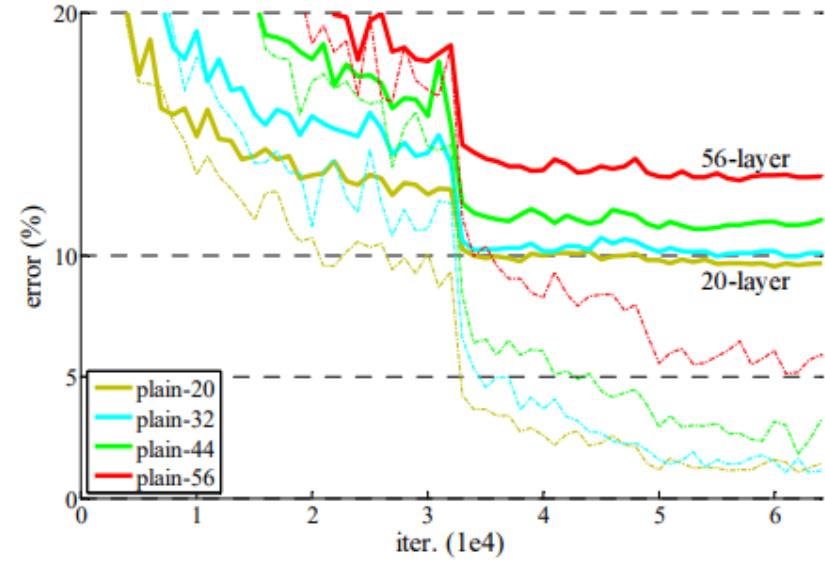
## 이미지 증강

- 테스트할 때는 학습 중에 사용한 증강 기법 없이, 원본  $32 \times 32$  이미지만을 평가
- 이미지를 4pixels씩 padding을 추가하고 랜덤하게 다시  $32 \times 32$ 로 crop

## 모델 훈련

- 45000개 훈련 데이터와 5000개 검증 데이터로 사용
- weight decay = 0.0001
- momentum = 0.9
- learning rate = 0.1
- 64,000번 반복 후 학습을 종료

# CIFAR-10 and Analysis

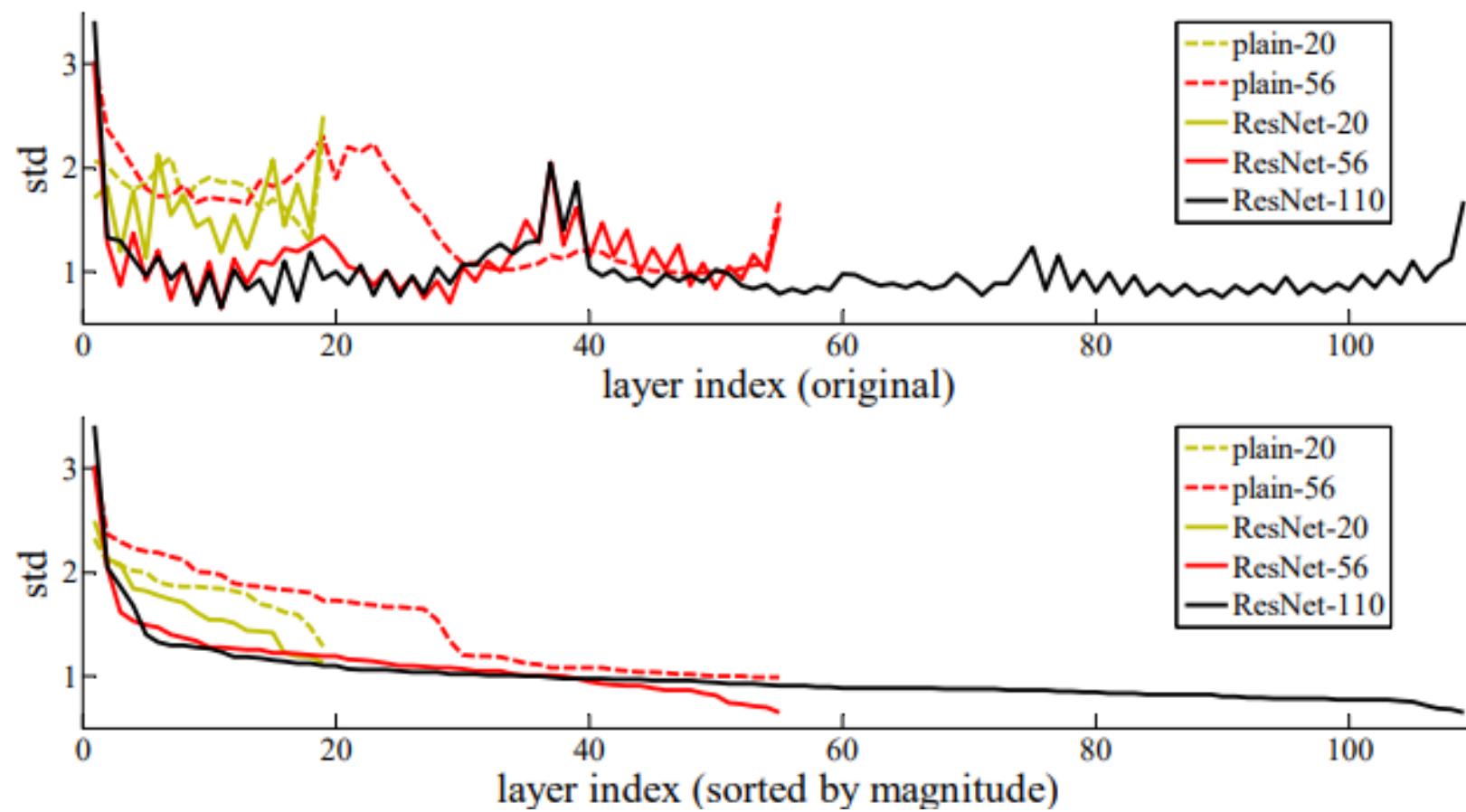


method	error (%)	
Maxout [10]	9.38	
NIN [25]	8.81	
DSN [24]	8.22	
	# layers	# params
FitNet [35]	19	2.5M
Highway [42, 43]	19	2.3M
Highway [42, 43]	32	1.25M
ResNet	20	0.27M
ResNet	32	0.46M
ResNet	44	0.66M
ResNet	56	0.85M
ResNet	110	1.7M
ResNet	1202	19.4M

## 깊이 별 네트워크 성능 비교

- $n = \{3, 5, 7, 9\}$ 를 설정해 20층, 32층, 44층, 56층의 네트워크를 구성
- plain 네트워크는 깊이가 깊어질수록 성능 저하를 겪고 훈련 오류가 증가
- ResNet은 깊이가 깊어질수록 성능이 향상
- $n=18$ 의 110층 ResNet의 경우 0.01의 학습률로 워밍업 (warming up) 과정을 거친 후 학습을 진행해서 다른 깊은 모델들에 비해 더 적은 매개변수로도 높은 성능
- $n=200$ 의 1202층 네트워크는 훈련 오류가 0.1% 미만에 도달 할 정도로 잘 학습되었으나 테스트 오류는 110층 네트워크 보다 더 하락

# CIFAR-10 and Analysis



## Layer Responses 분석

- 각 레이어의 배치 정규화(BN) 이후이면서 ReLU, residual 덧셈 연산 전에 발생한 출력값 측정
- ResNet과 플레인 네트워크를 비교했을 때, ResNet은 전반적으로 작은 응답을 보임
- ResNet의 레이어 수가 많아질수록 응답 크기가 더 작아짐

# Object Detection on PASCAL and MS COCO

training data	07+12	07++12
test data	VOC 07 test	VOC 12 test
VGG-16	73.2	70.4
ResNet-101	<b>76.4</b>	<b>73.8</b>

metric	mAP@.5	mAP@[.5, .95]
VGG-16	41.5	21.2
ResNet-101	<b>48.4</b>	<b>27.2</b>

## Object Detection on PASCAL and MS COCO

- PASCAL VOC 2007/2012와 COCO 데이터셋에서 Faster R-CNN를 객체 탐지 방법으로 사용
- COCO의 표준 메트릭인 mAP@[.5, .95]에서 6.0%의 증가를 얻었으며, 28%의 상대적 성능 향상 결과 도출

감사합니다

Thank you