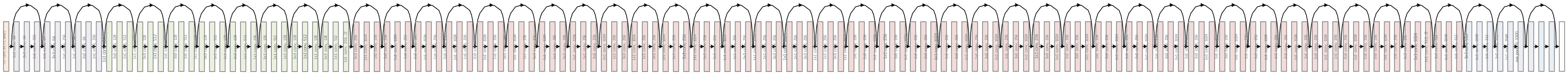
Deep Residual Networks (ResNet)

- Deep Residual Learning for Image Recognition
- Identity Mappings in Deep Residual Networks



Background of Resnet's appearance

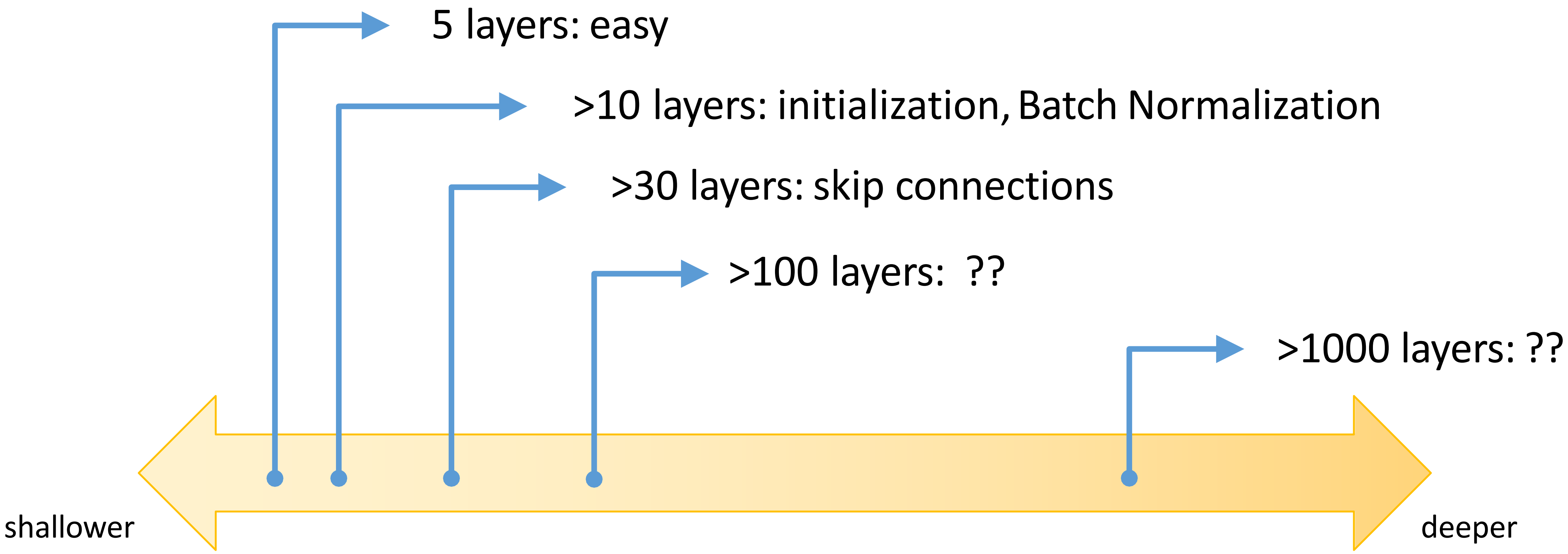
arXiv
https://arxiv.org › cs

[1512.03385] Deep Residual Learning for Image Recognition

K He 저술 · 2015 · 181713회 인용 — We present a **residual learning** framework to ease the

Table of Contents

Before ResNet

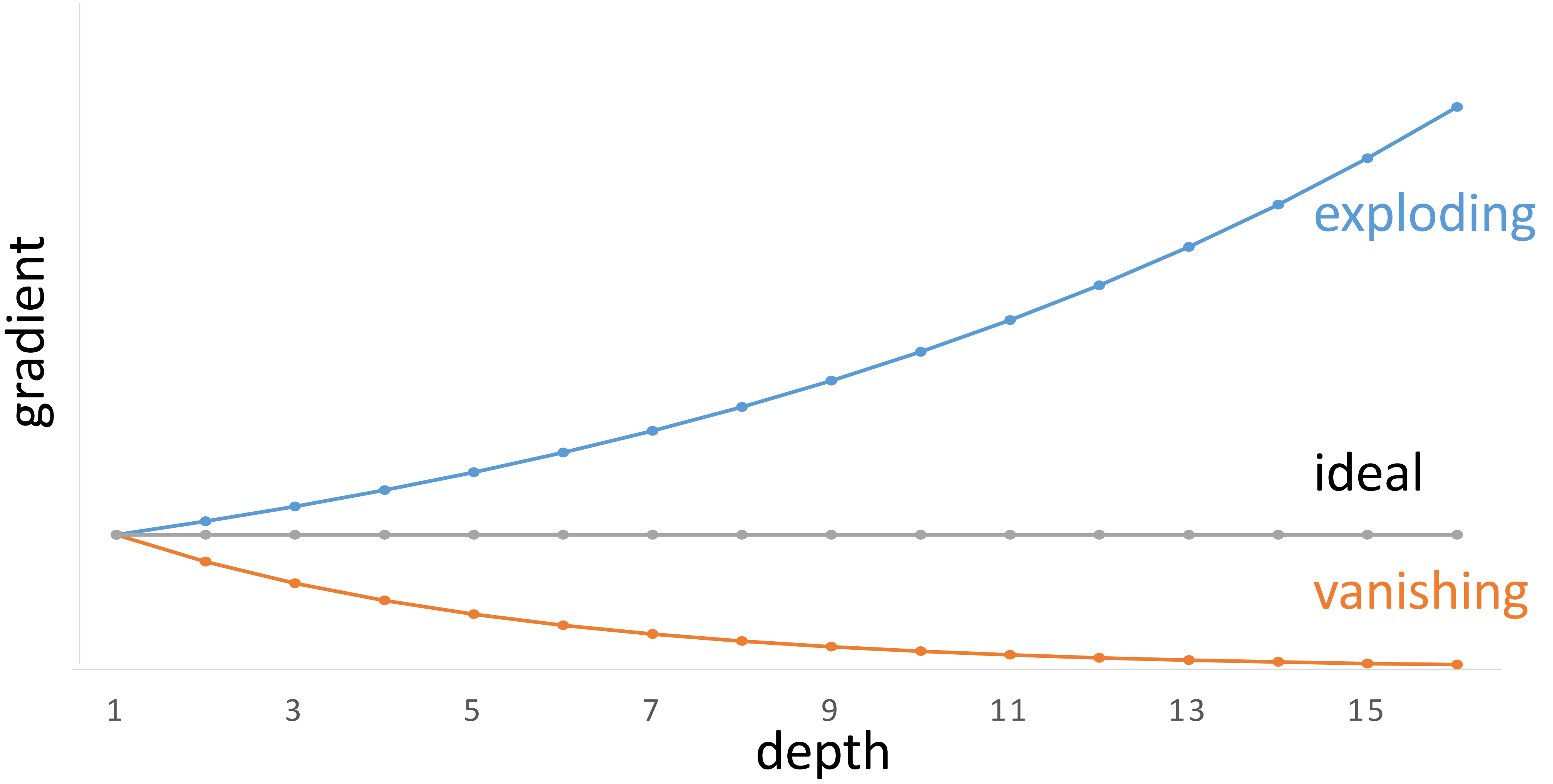


Problems with deep layers

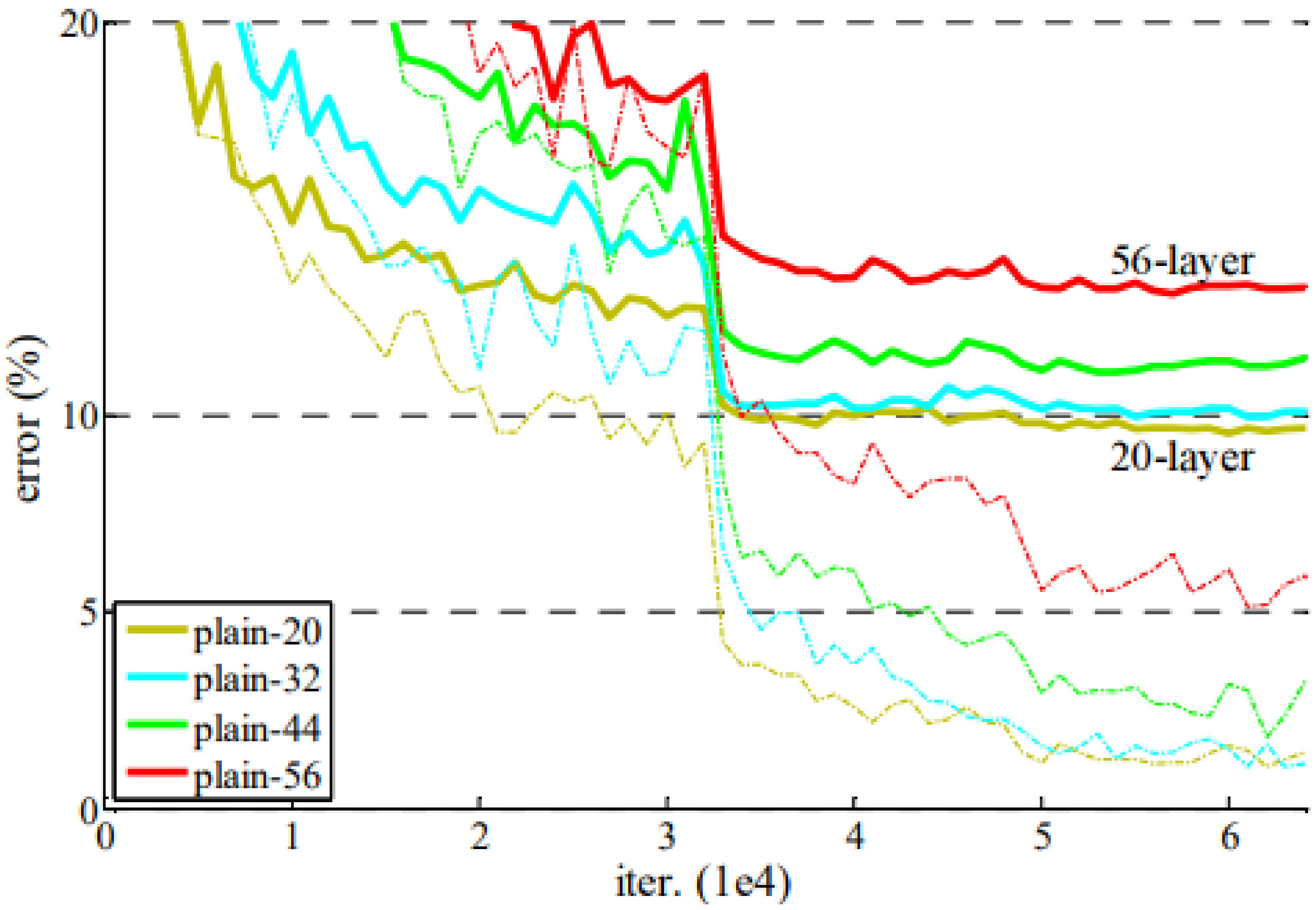
1. Vanishing gradient problem

2. Degradation problem

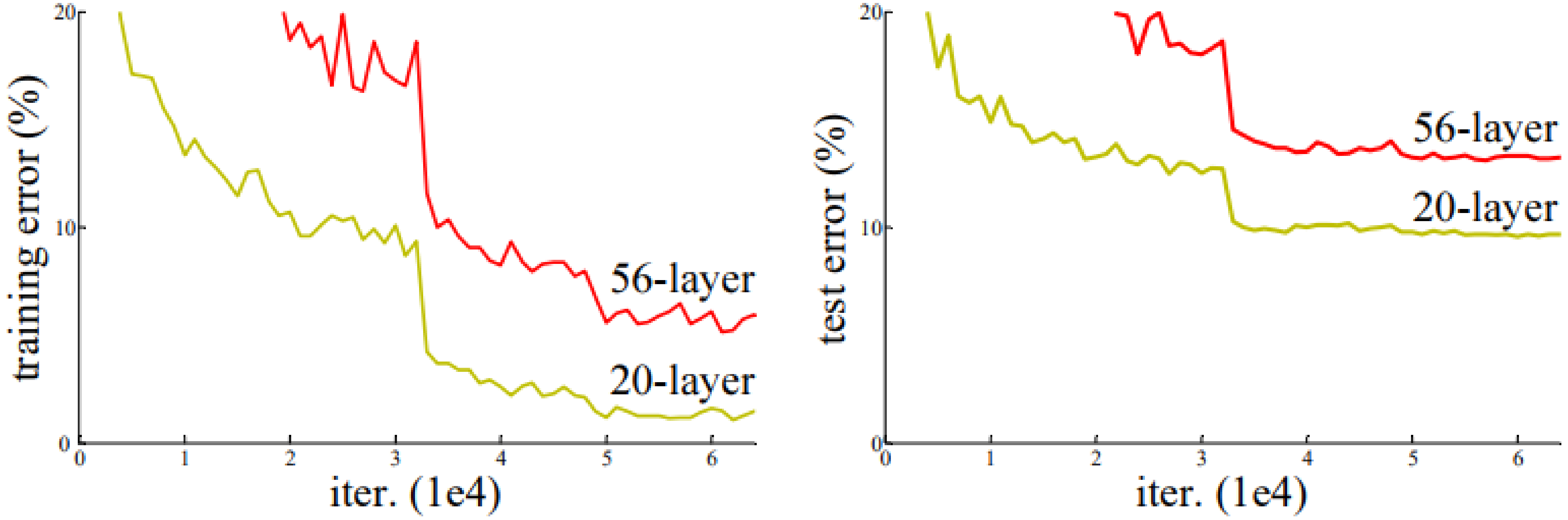
Vanishing/Exploding Gradient Problem



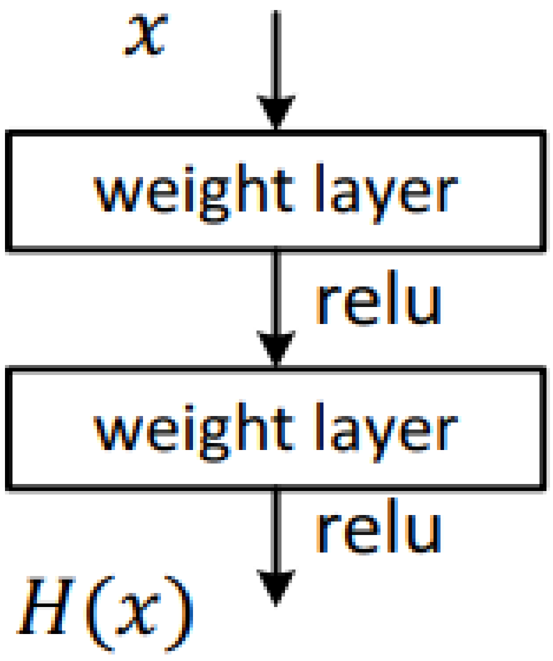
Degradation Problem

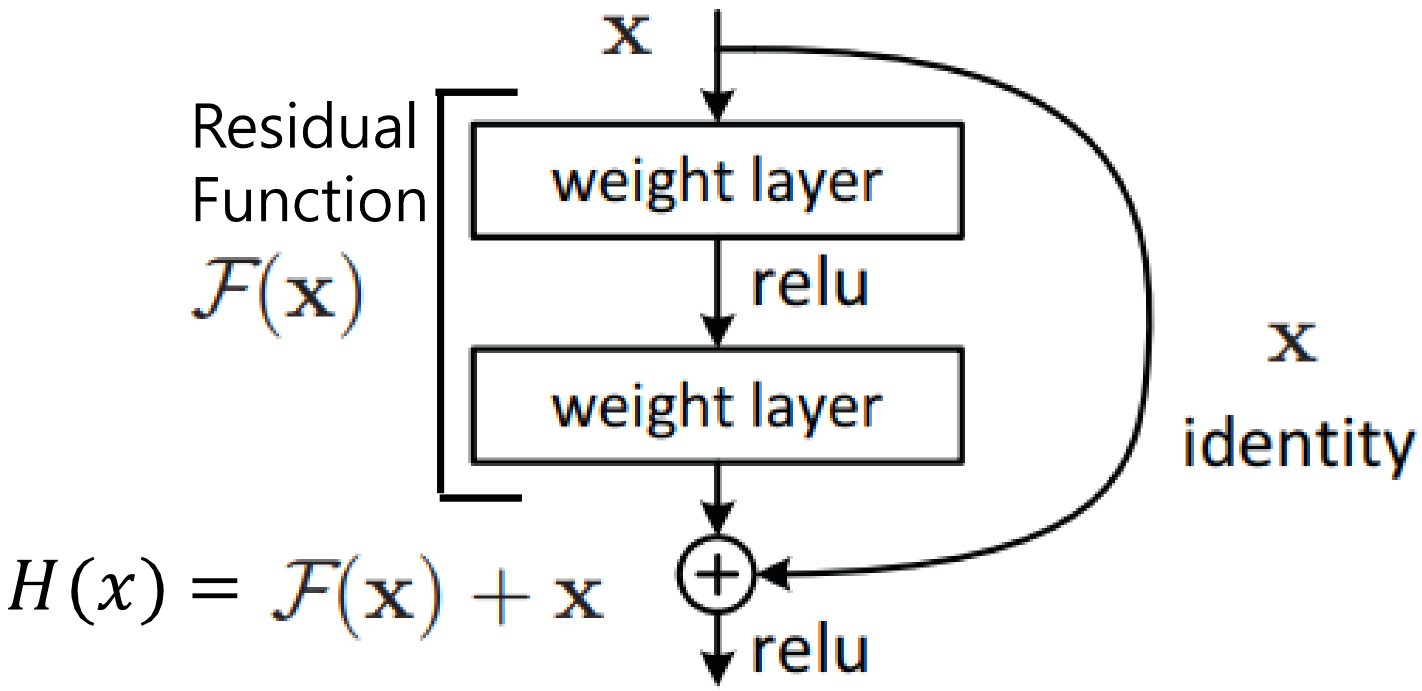


Degradation Problem



What is Residual Learning?

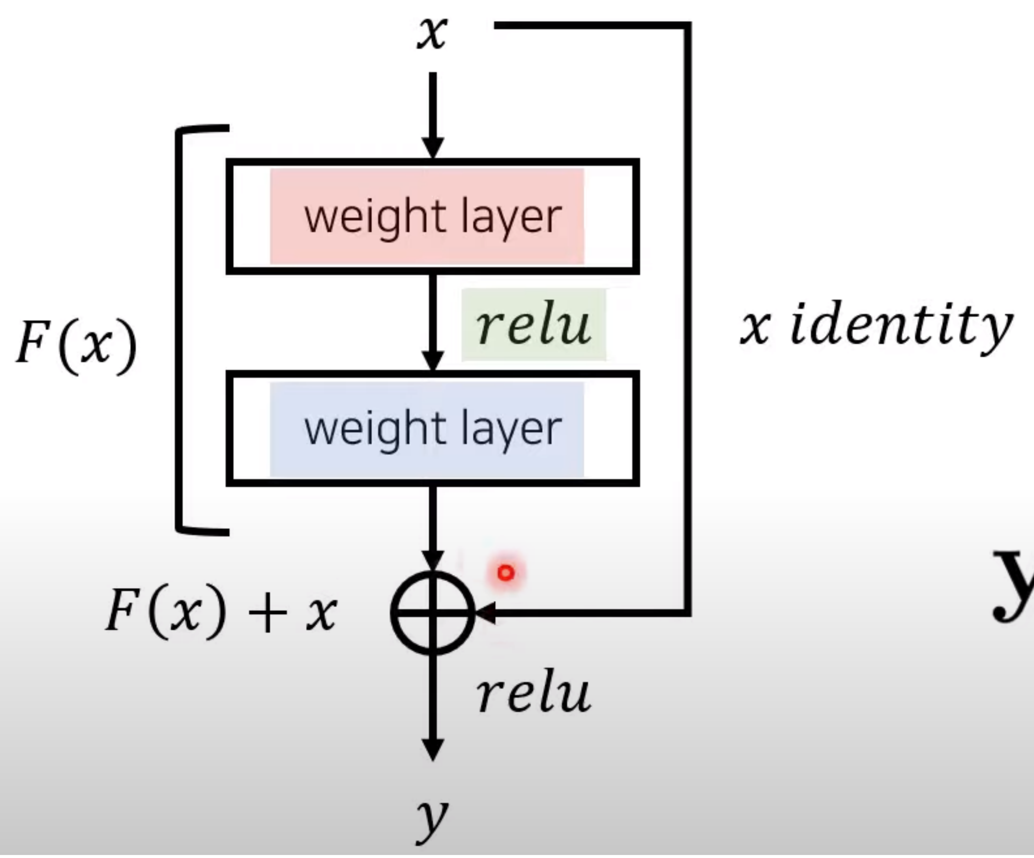




Plain net

Residual net

ResNet's Advantage
1) # of parameters doesn't increase

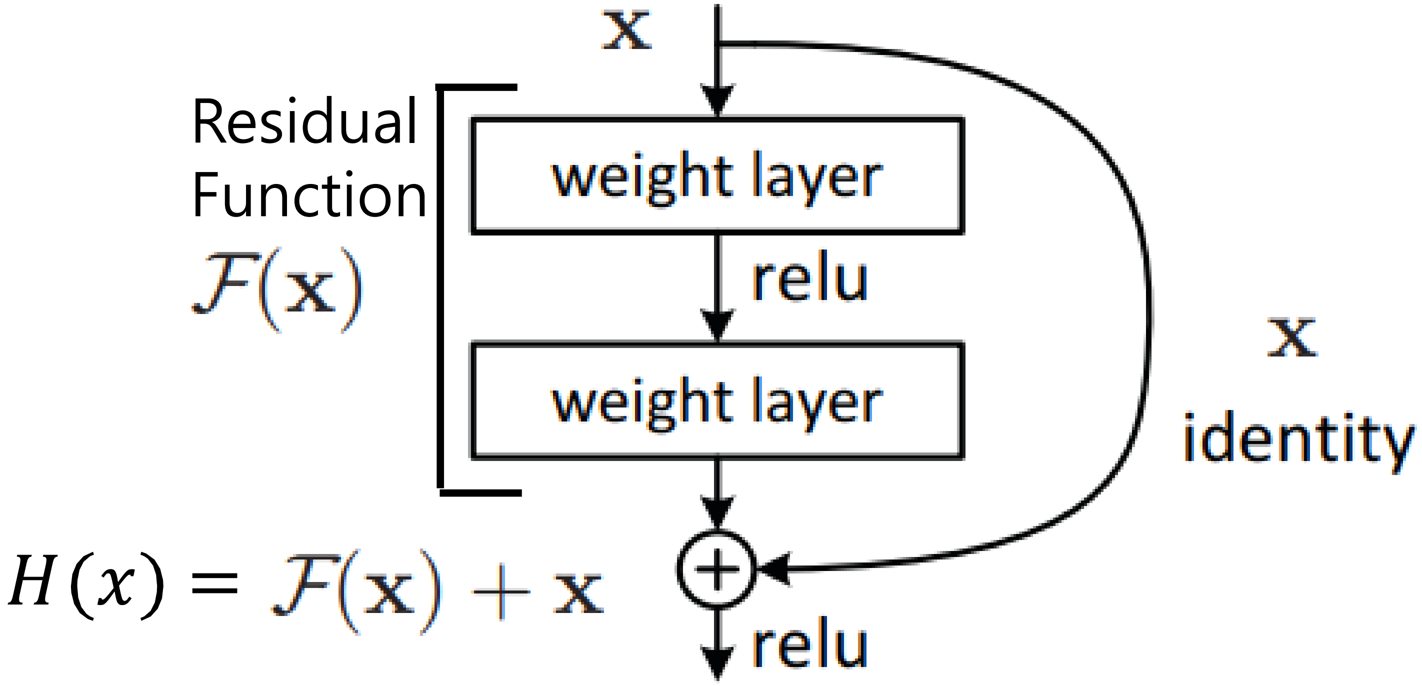


$$\mathcal{F} = W_2\sigma(W_1\mathbf{x})$$

일반적인 형태

$$\mathbf{y} = \underbrace{\mathcal{F}(\mathbf{x}, \{W_i\})}_{\text{multiple convolutional layers}} + \underbrace{W_s\mathbf{x}}_{\text{shortcut}}$$

ResNet's Advantage
2) Solve Vanishing Gradient Problem

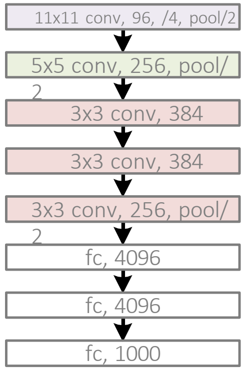


Residual net

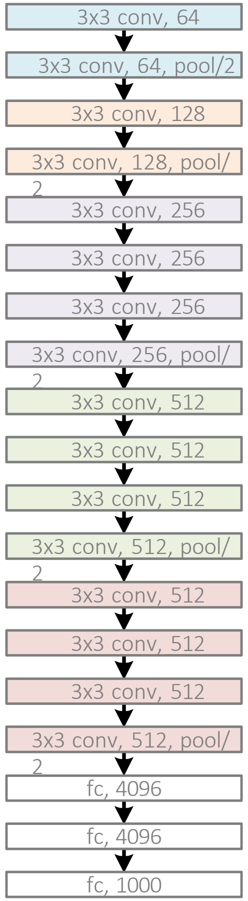
$$\frac{\partial \mathrm{H}}{\partial x} = \frac{\partial \mathrm{F}}{\partial x} + 1$$

ResNet's Advantage
3) High Accuracy in Deep structure

AlexNet, 8 layers
(ILSVRC 2012)



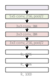
VGG, 19 layers
(ILSVRC 2014)



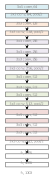
GoogleNet, 22 layers
(ILSVRC 2014)

ResNet's Advantage
3) High Accuracy in Deep structure

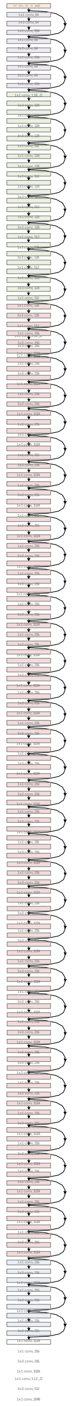
AlexNet, 8 layers
(ILSVRC 2012)



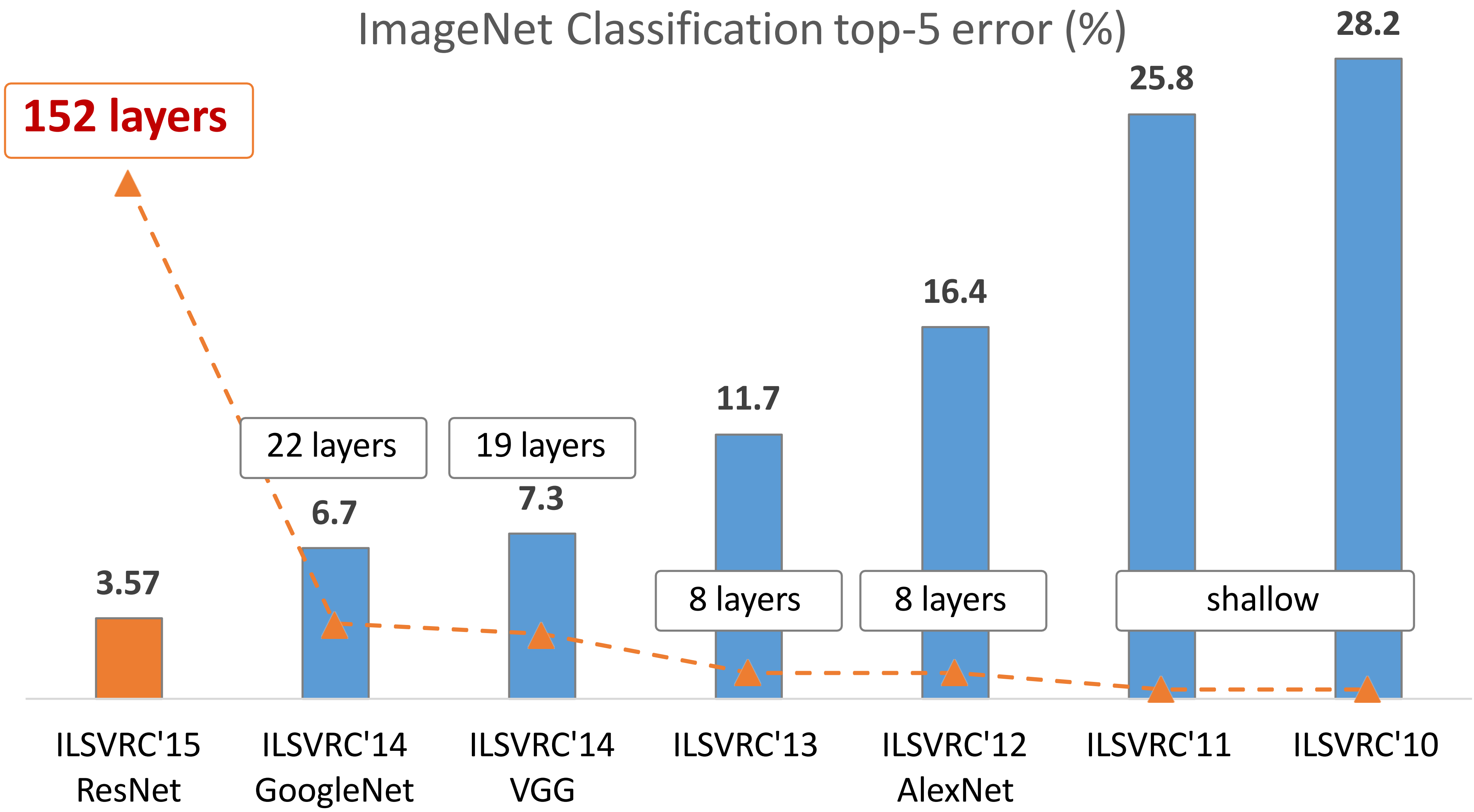
VGG, 19 layers
(ILSVRC 2014)



ResNet, 152 layers
(ILSVRC 2015)



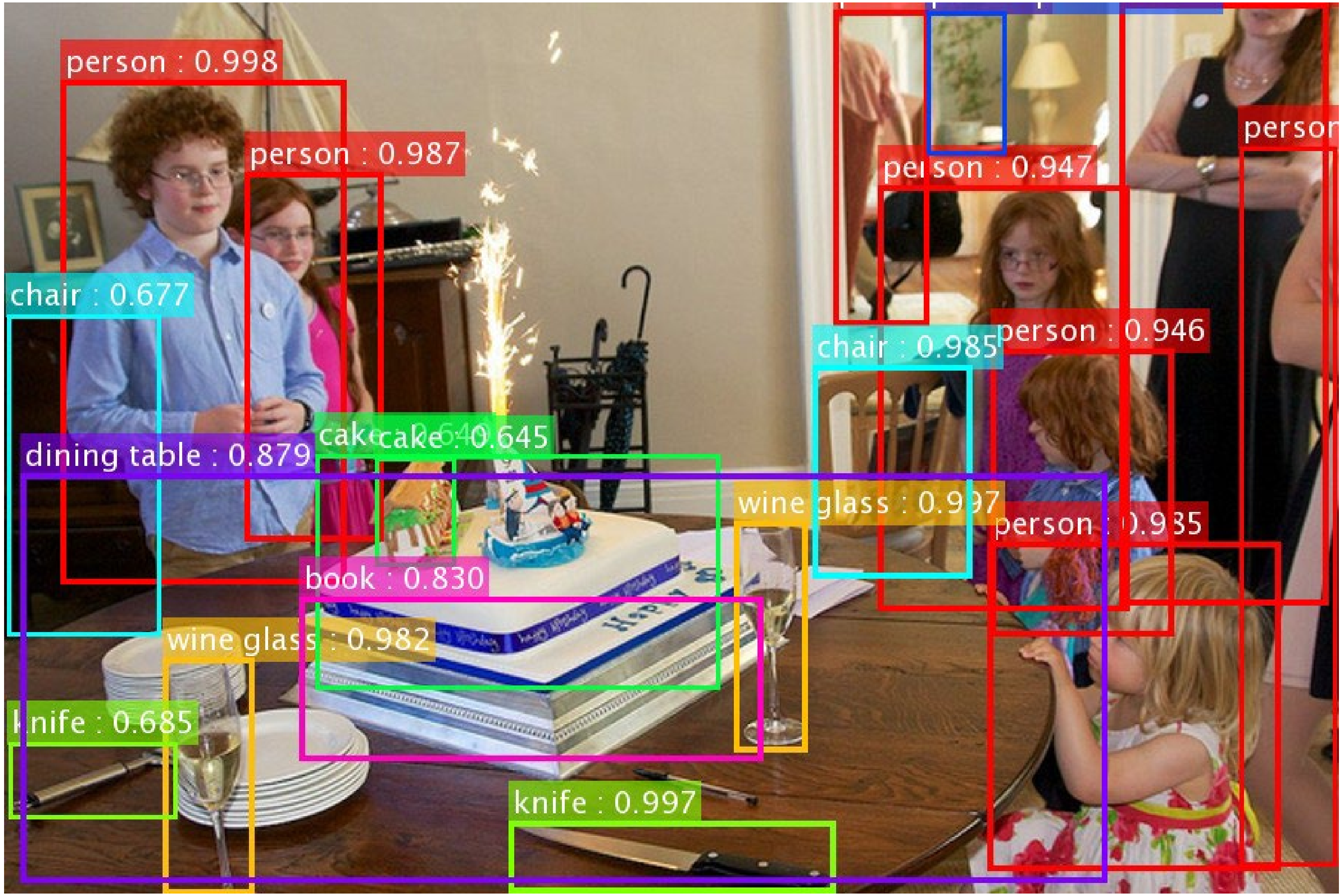
ResNet's Advantage
3) High Accuracy in Deep structure



ResNet at ILSVRC & COCO 2015 Competitions

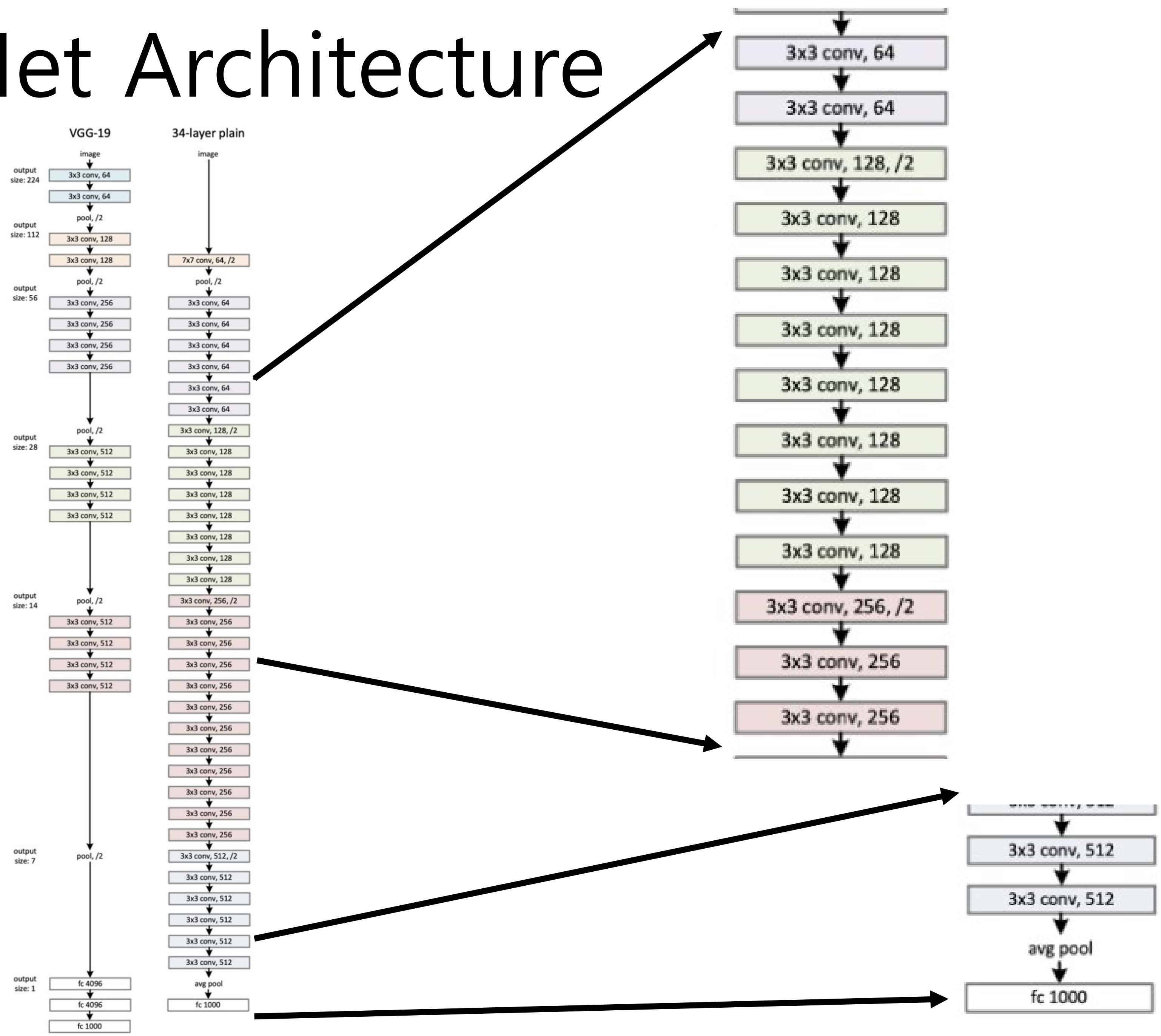
- **1st places in all five main tracks**
  - ImageNet Classification: "*Ultra-deep*" 152-layer nets
  - ImageNet Detection: 16% better than 2nd
  - ImageNet Localization: 27% better than 2nd
  - COCO Detection: 11% better than 2nd
  - COCO Segmentation: 12% better than 2nd

*improvements are relative numbers

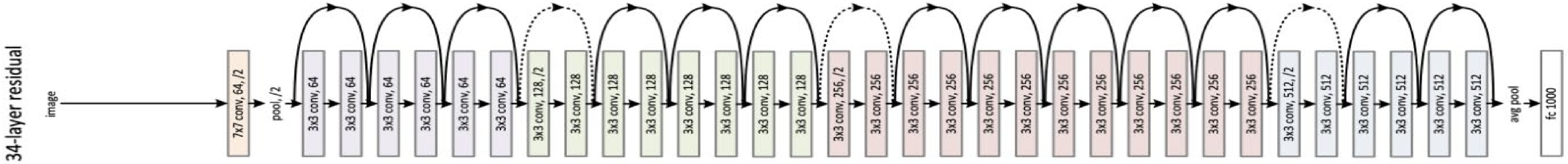


ResNet’s object detection result on COCO

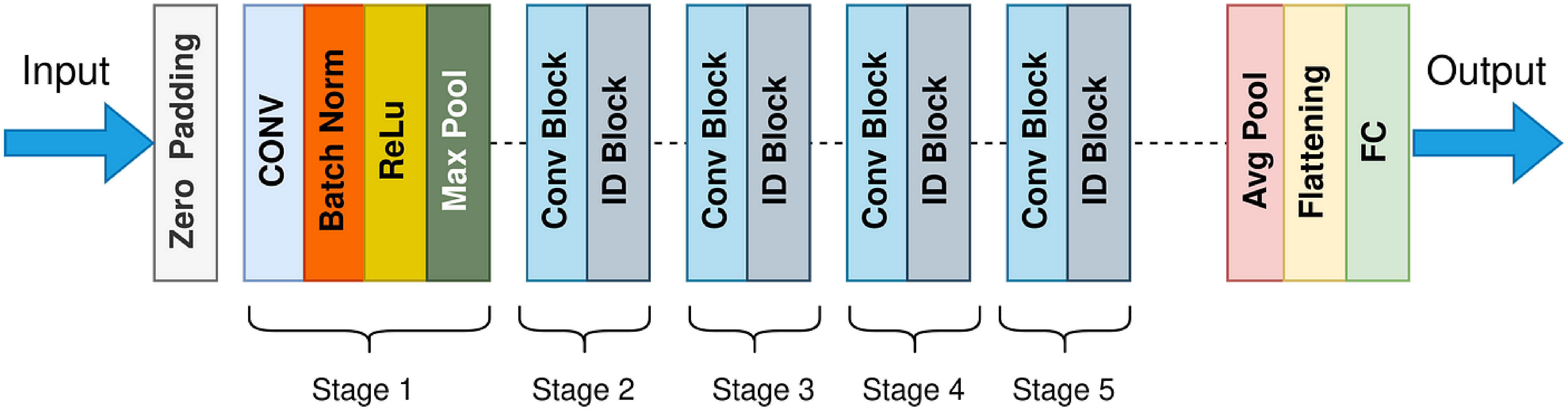
ResNet Architecture



ResNet Architecture



ResNet50 Model Architecture



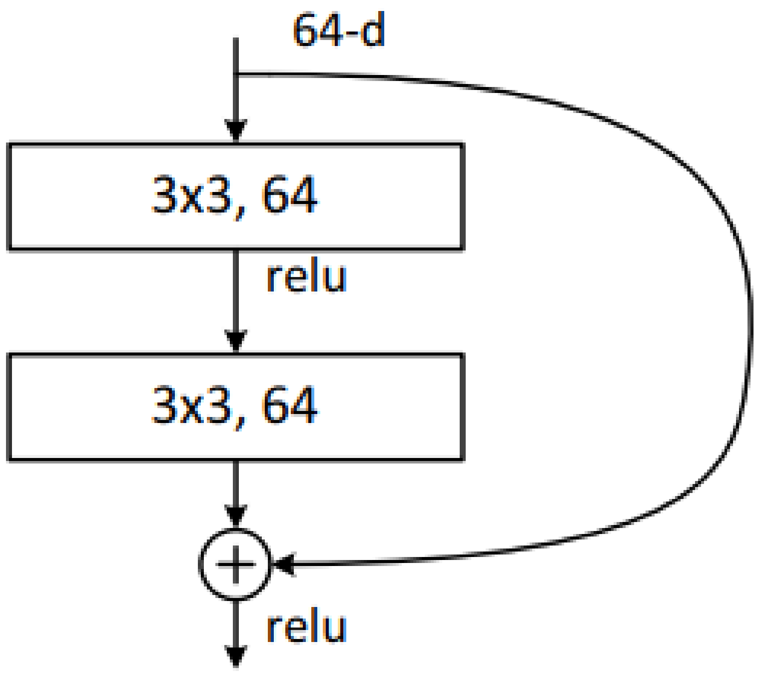
ResNet Architecture

- How to align dimensions of input and output

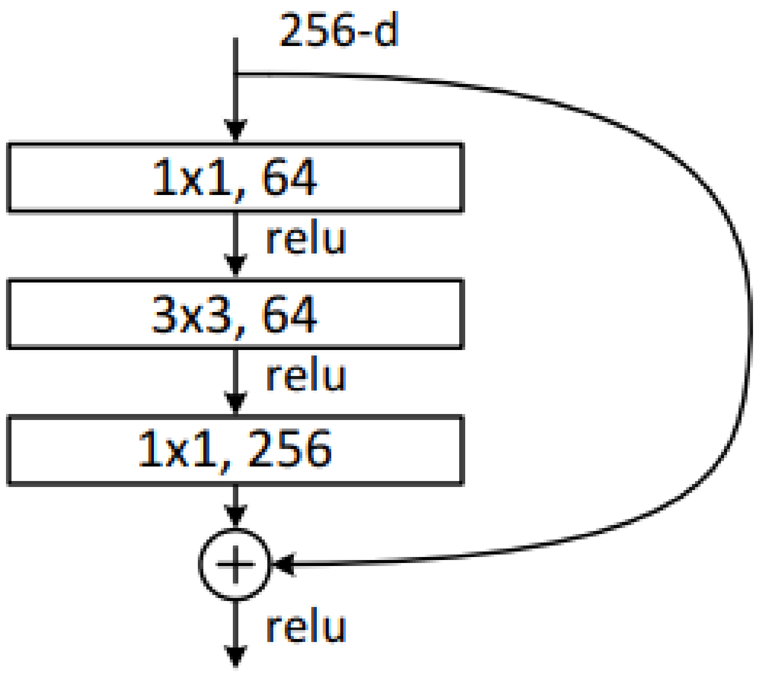
1. Zero padding

2. linear projection W_s 사용: $y = F(x, \{W_i\}) + W_s x$

Deeper Bottleneck Architecture

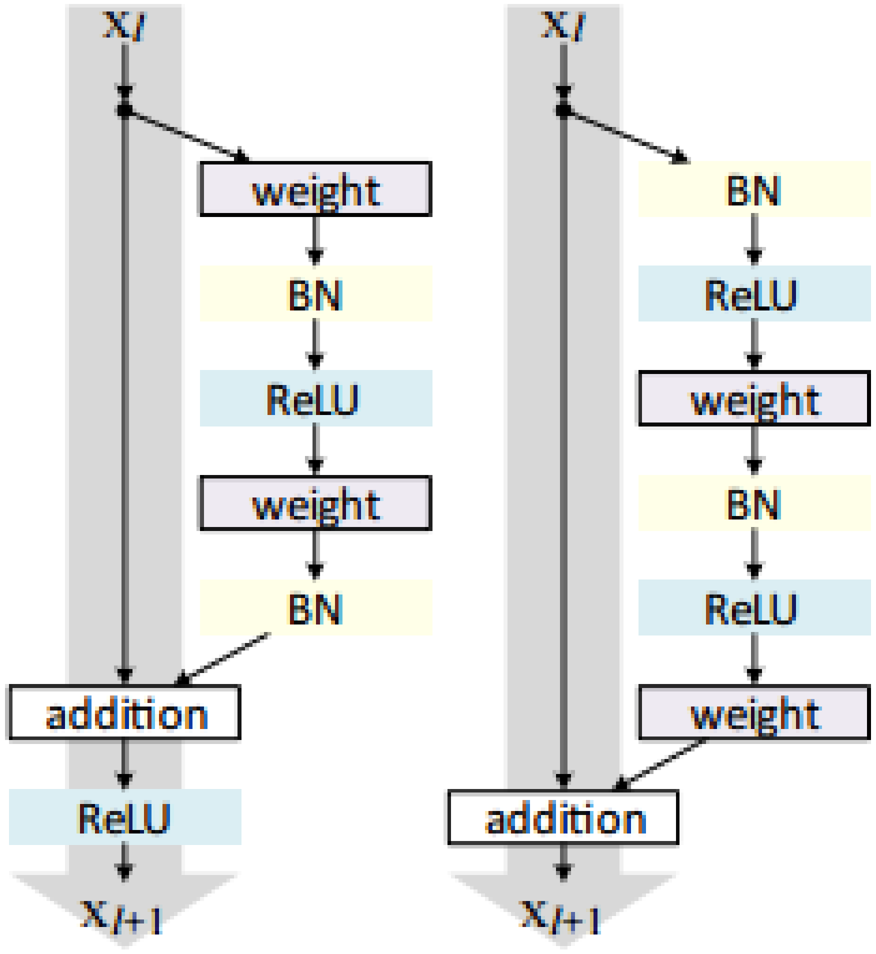


Residual net



Deeper Bottleneck

Pre-Activation ResNet



(a) original (b) proposed

Pre-Activation ResNet

$$\mathbf{y}_l = h(\mathbf{x}_l) + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l), \tag{1}$$

$$\mathbf{x}_{l+1} = f(\mathbf{y}_l). \tag{2}$$

x: residual unit input

y: residual unit output

l: number of each layer

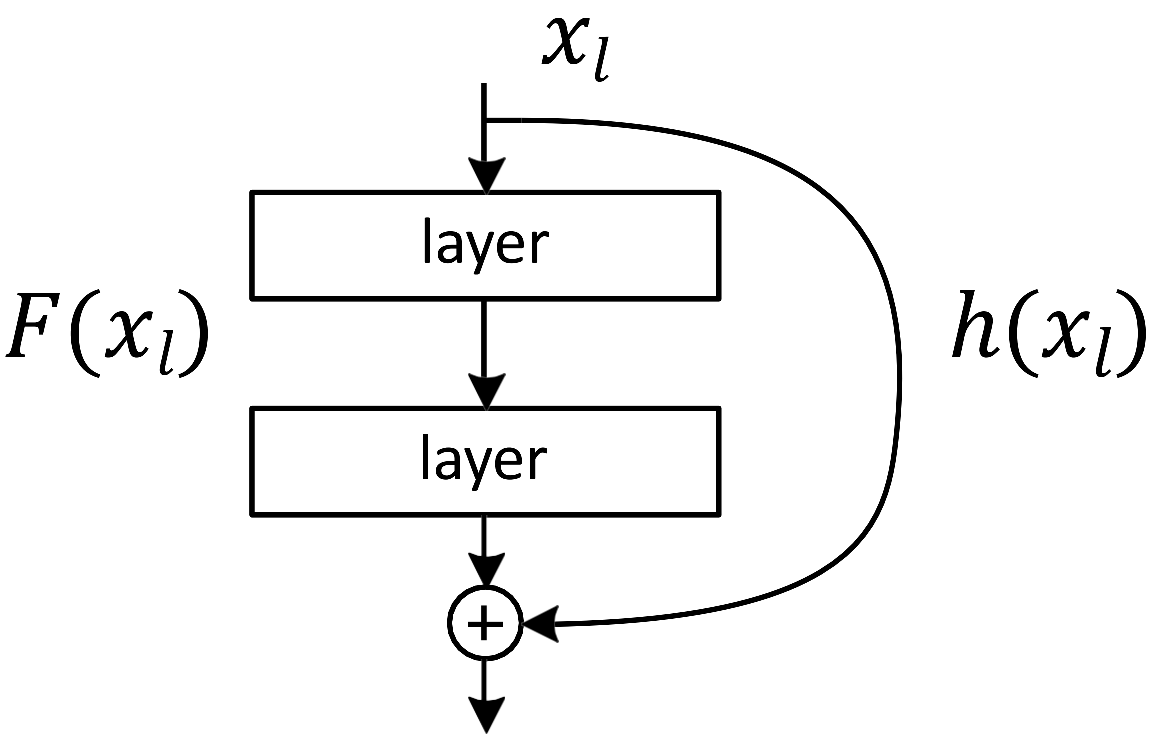
W: weight

F: residual function F(x)

f: activation function

H: identity function

Pre-Activation ResNet



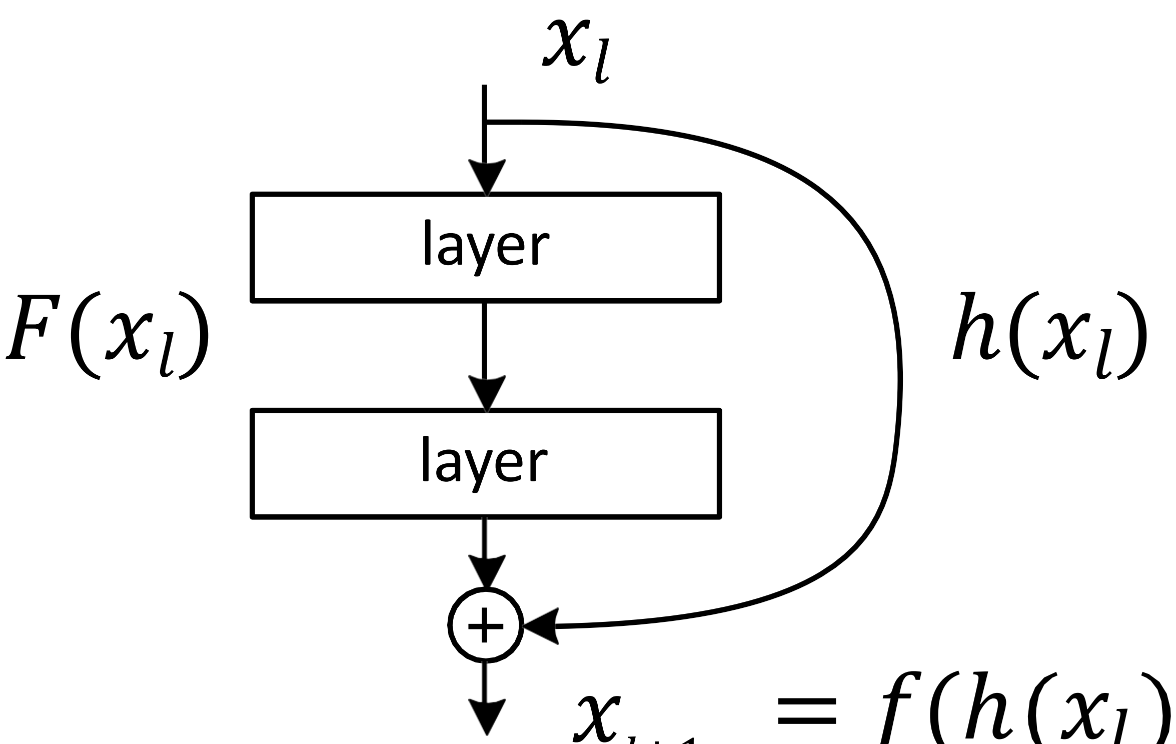
x: residual unit input
y: residual unit output
l: number of each layer
W: weight
F: residual function
f: activation function
H: identity function

$$\mathbf{y}_l = h(\mathbf{x}_l) + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l), \quad (1)$$

$$\mathbf{x}_{l+1} = f(\mathbf{y}_l). \quad (2)$$

$$x_{l+1} = f(h(x_l) + F(x_l))$$

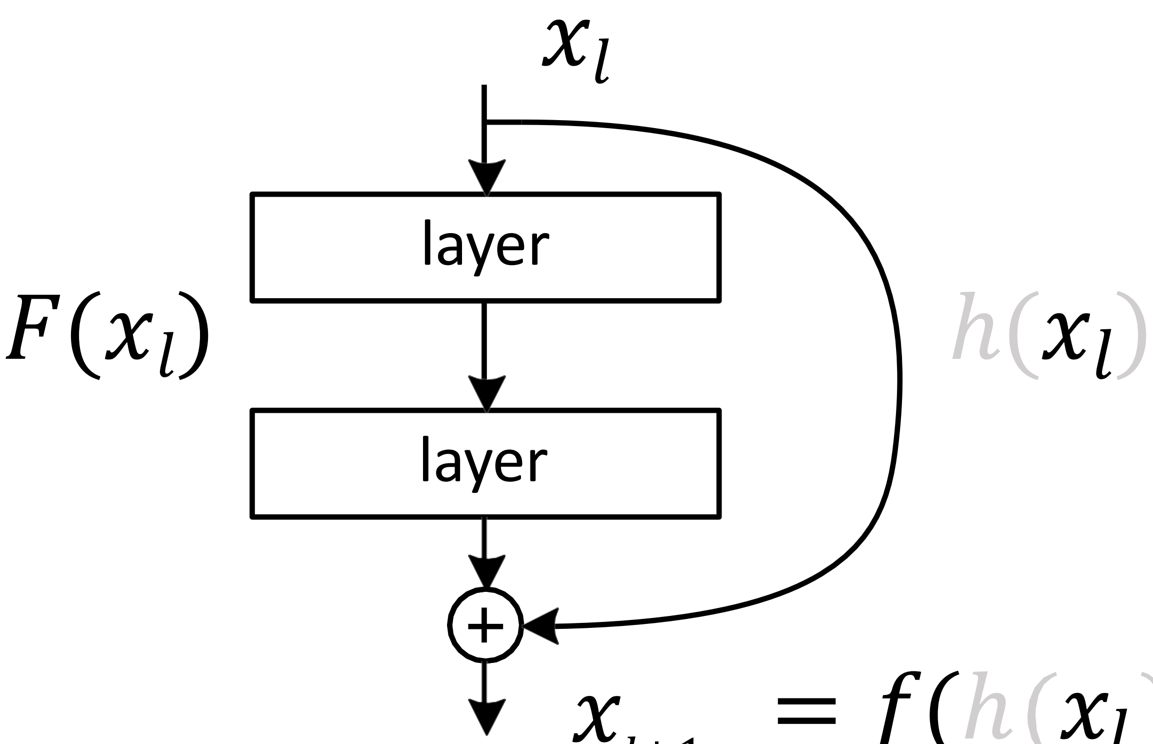
Pre-Activation ResNet



$$x_{l+1} = f(h(x_l) + F(x_l))$$

- shortcut mapping: h = identity
- ~~after-add mapping: f = ReLU~~
- What if f = identity?

Pre-Activation ResNet



$$x_{l+1} = f(h(x_l) + F(x_l))$$

- shortcut mapping: h = identity
- ~~f = ReLU~~
- What if f = identity?

Pre-Activation ResNet

$$\mathbf{x}_{l+1} = f(\mathbf{y}_l). \tag{2}$$

Change activation function f to identity mapping !

$$\mathbf{x}_{l+1} = \mathbf{y}_l$$

Pre-Activation ResNet

대입

$$\mathbf{x}_{l+1} = \mathbf{y}_l \quad \Rightarrow \quad \mathbf{y}_l = h(\mathbf{x}_l) + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l) \qquad (1)$$

$$\mathbf{x}_{l+1} = \mathbf{x}_l + \mathcal{F}(\mathbf{x}_l, \mathcal{W}_l) \qquad (3)$$

Pre-Activation ResNet

$$x_{l+1} = x_l + F(x_l)$$

$$\Downarrow$$

$$x_{l+2} = x_{l+1} + F(x_{l+1})$$

Pre-Activation ResNet

$$x_{l+1} = x_l + F(x_l)$$

$$x_{l+2} = x_{l+1} + F(x_{l+1})$$

$$x_{l+2} = x_l + F(x_l) + F(x_{l+1})$$

Pre-Activation ResNet

$$\mathbf{x}_L = \mathbf{x}_l + \sum_{i=l}^{L-1} \mathcal{F}(\mathbf{x}_i, \mathcal{W}_i) \qquad (4)$$

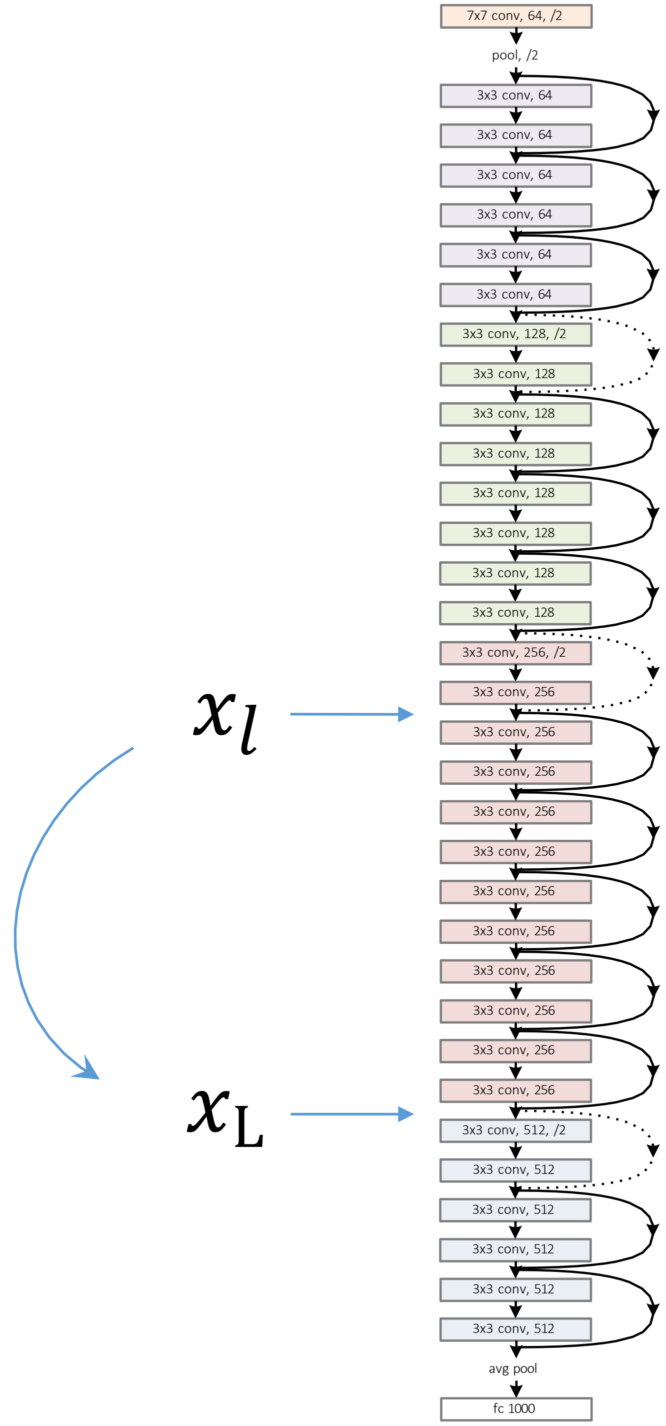
- When Feed forwarding, ResNet can be expressed as the sum of Residual Function F

Forward Propagation

$$\mathbf{x}_L = \mathbf{x}_l + \sum_{i=l}^{L-1} \mathcal{F}(\mathbf{x}_i, \mathcal{W}_i)$$

- Any x_l is directly forward propagation to any x_L, plus residual.

- Any x_L is an additive outcome.
  - in contrast to multiplicative: $x_\mathrm{L} = \prod_{\mathrm{i=1}}^{\mathrm{L\text{-}1}} W_\mathrm{i}\, x_l$



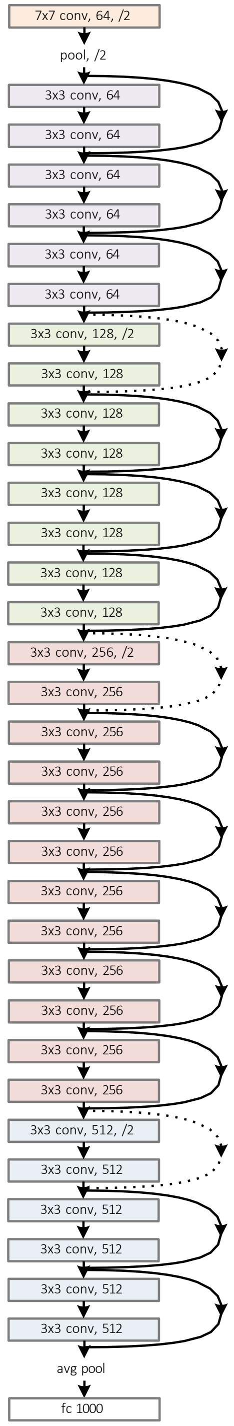
Back Propagation

$$\mathbf{x}_L = \mathbf{x}_l + \sum_{i=l}^{L-1} \mathcal{F}(\mathbf{x}_i, \mathcal{W}_i)$$

$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\frac{\partial \mathbf{x}_L}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\left(1 + \frac{\partial}{\partial \mathbf{x}_l}\sum_{i=l}^{L-1} \mathcal{F}(\mathbf{x}_i, \mathcal{W}_i)\right)$$

$$\frac{\partial E}{\partial x_{\mathrm{l}}}$$

$$\frac{\partial E}{\partial x_{\mathrm{L}}}$$



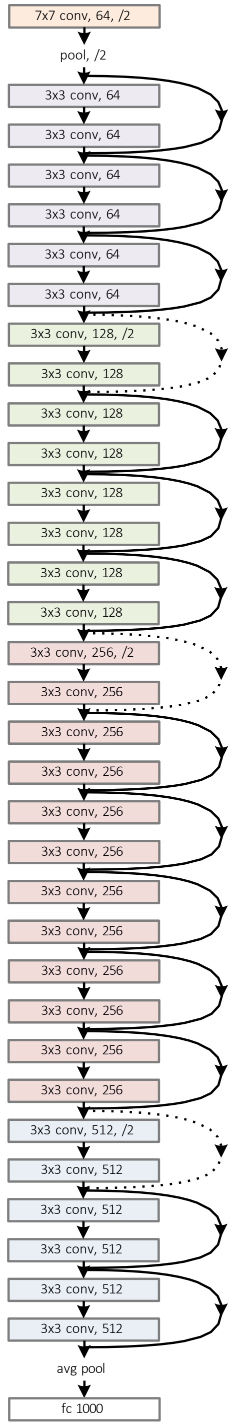
Back Propagation

$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\frac{\partial \mathbf{x}_L}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\left(1 + \frac{\partial}{\partial \mathbf{x}_l}\sum_{i=l}^{L-1}\mathcal{F}(\mathbf{x}_i, \mathcal{W}_i)\right)$$

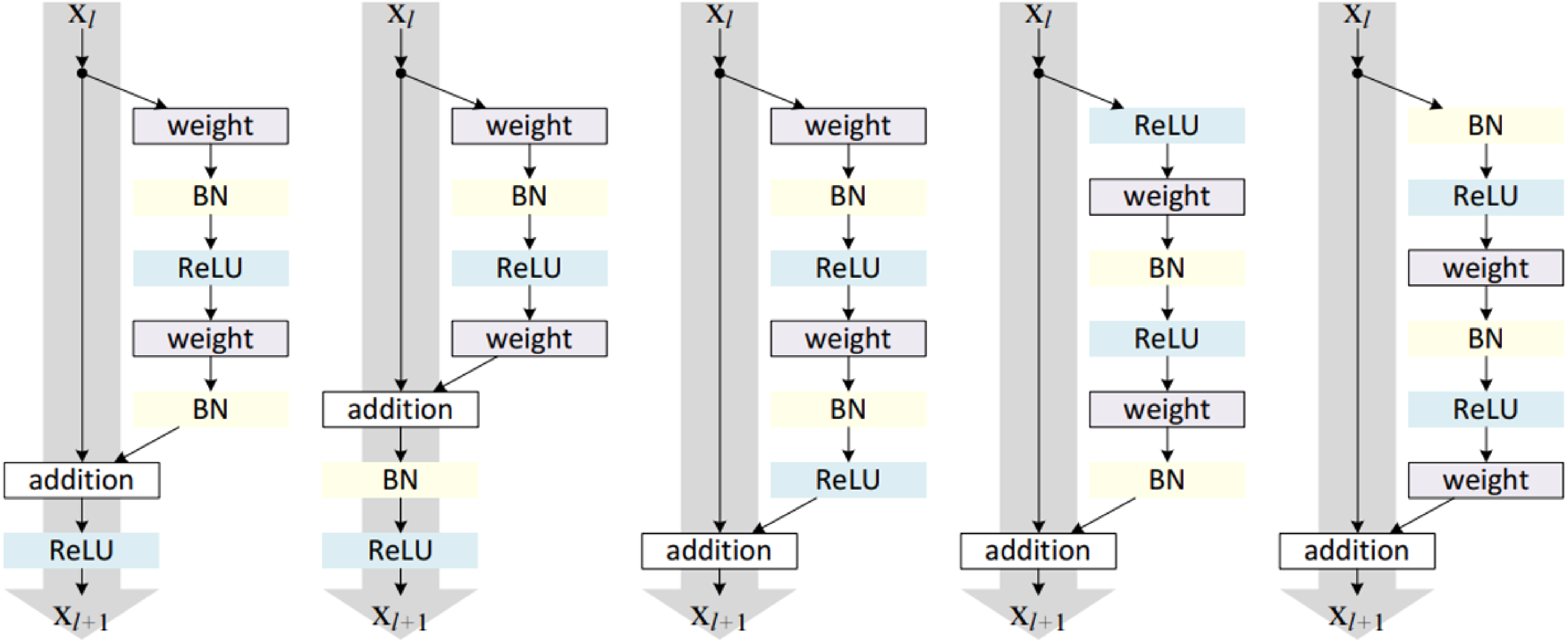
- Any $\frac{\partial E}{\partial x_{\mathrm{L}}}$ is directly back propagation to any $\frac{\partial E}{\partial x_{\mathrm{l}}}$ plus residual.
- Any $\frac{\partial E}{\partial x_{\mathrm{l}}}$ is additive; unlikely to vanish
  - in contrast to multiplicative: $\frac{\partial E}{\partial x_{\mathrm{l}}} = \prod_{\mathrm{i=1}}^{\mathrm{L-1}} W_{\mathrm{i}} \frac{\partial E}{\partial x_{\mathrm{L}}}$

$$\frac{\partial E}{\partial x_{\mathrm{l}}} \rightarrow$$

$$\frac{\partial E}{\partial x_{\mathrm{L}}} \rightarrow$$



Pre-Activation ResNet



(a) original

error : 6.61%

(b) BN after addition

error : 8.17%

(c) ReLU before addition

error : 7.84%

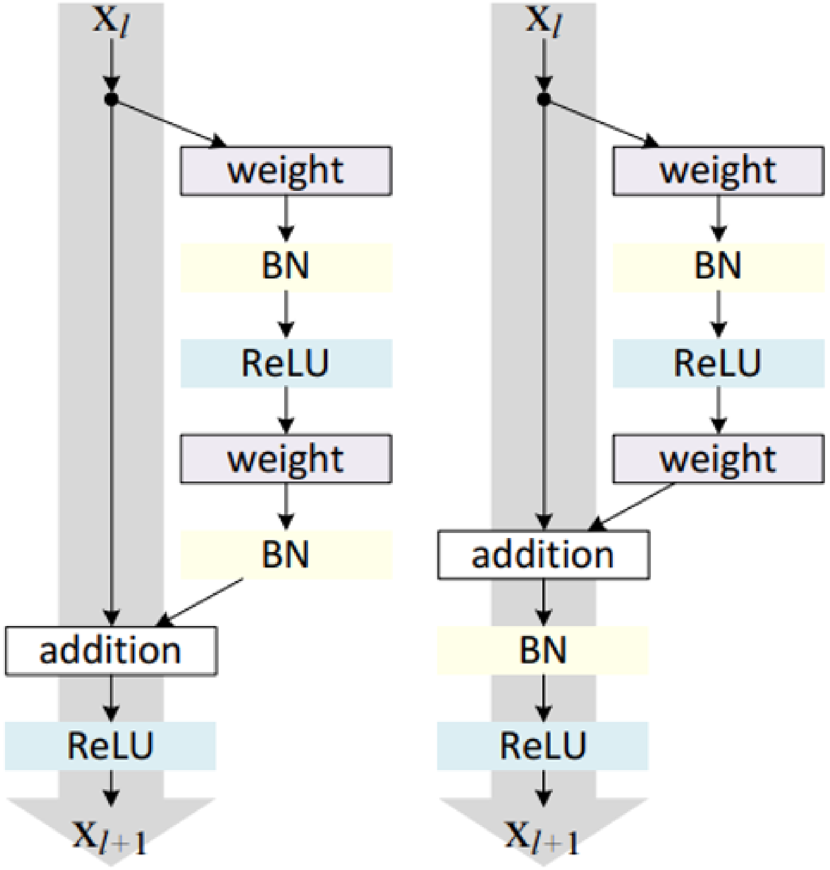
(d) ReLU-only pre-activation

error : 6.71%

(e) **full pre-activation**

error : 6.37%

Pre-Activation ResNet



(a) original

error : 6.61%

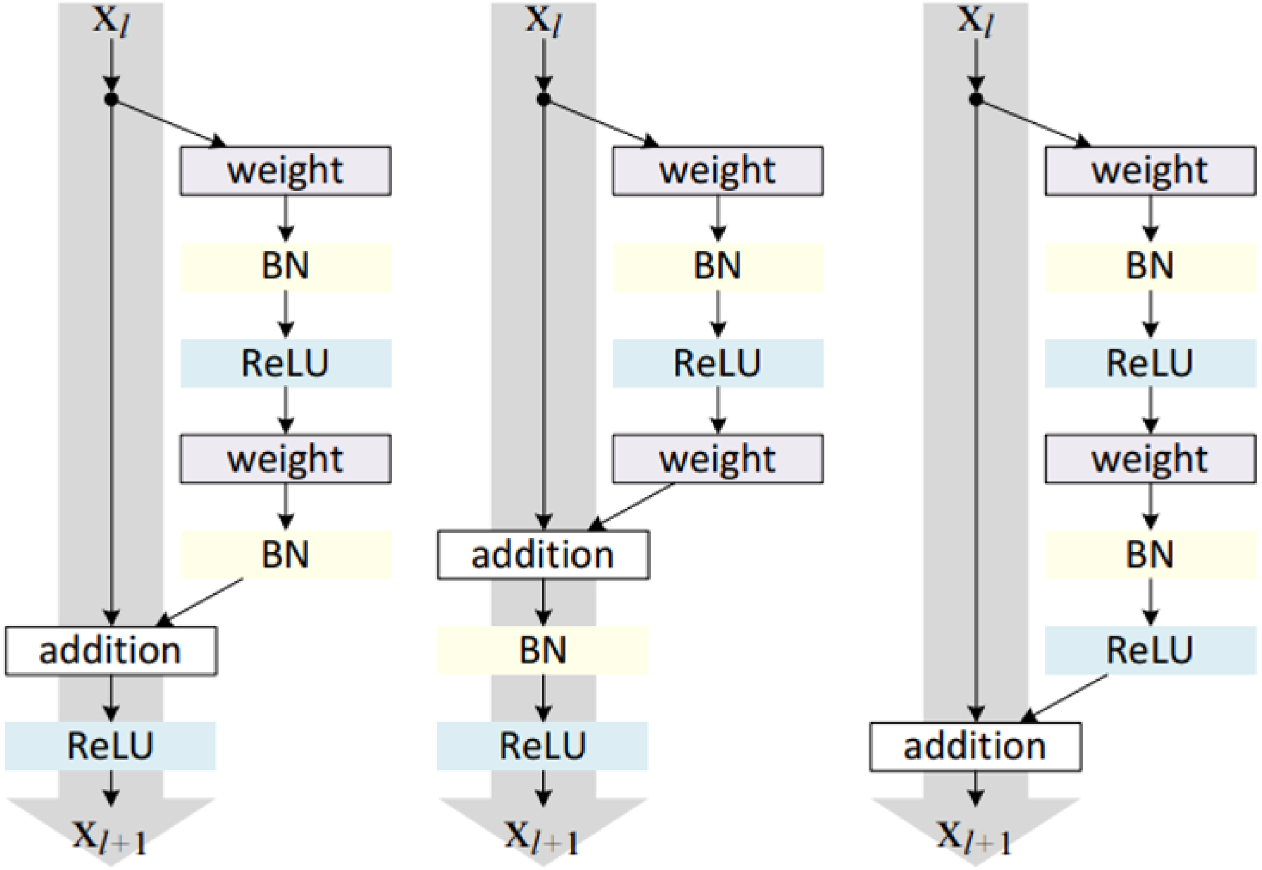
(b) BN after addition

error : 8.17%

Batch Normalization (BN)

- Normalizing input
- BN: normalizing each layer, for each mini-batch
- Batch: Number of data when the model updates parameters once
- Greatly accelerate training
- Improve regularization

Pre-Activation ResNet



(a) original

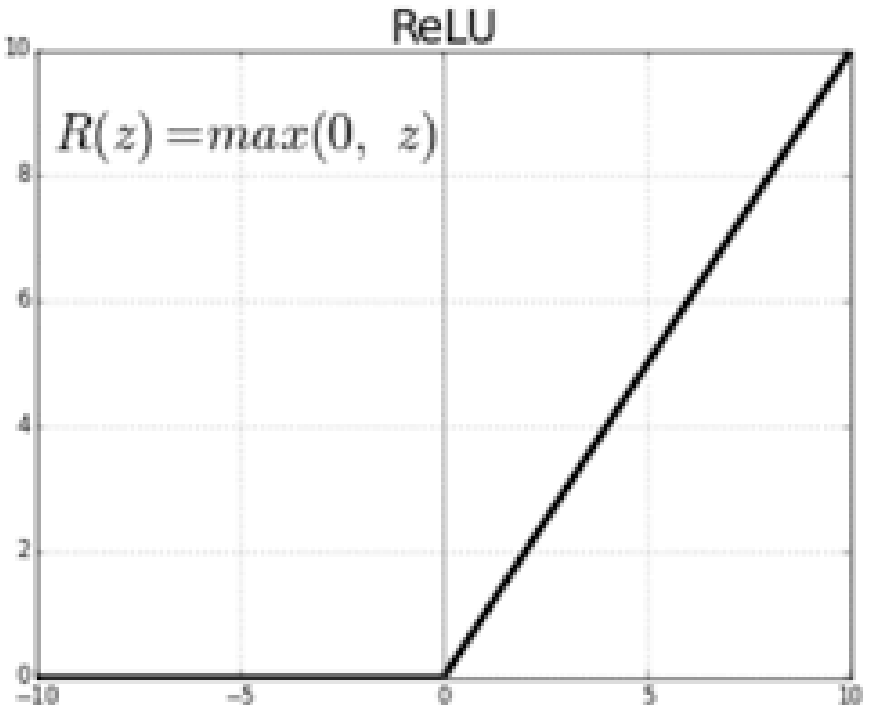
error : 6.61%

(b) BN after addition

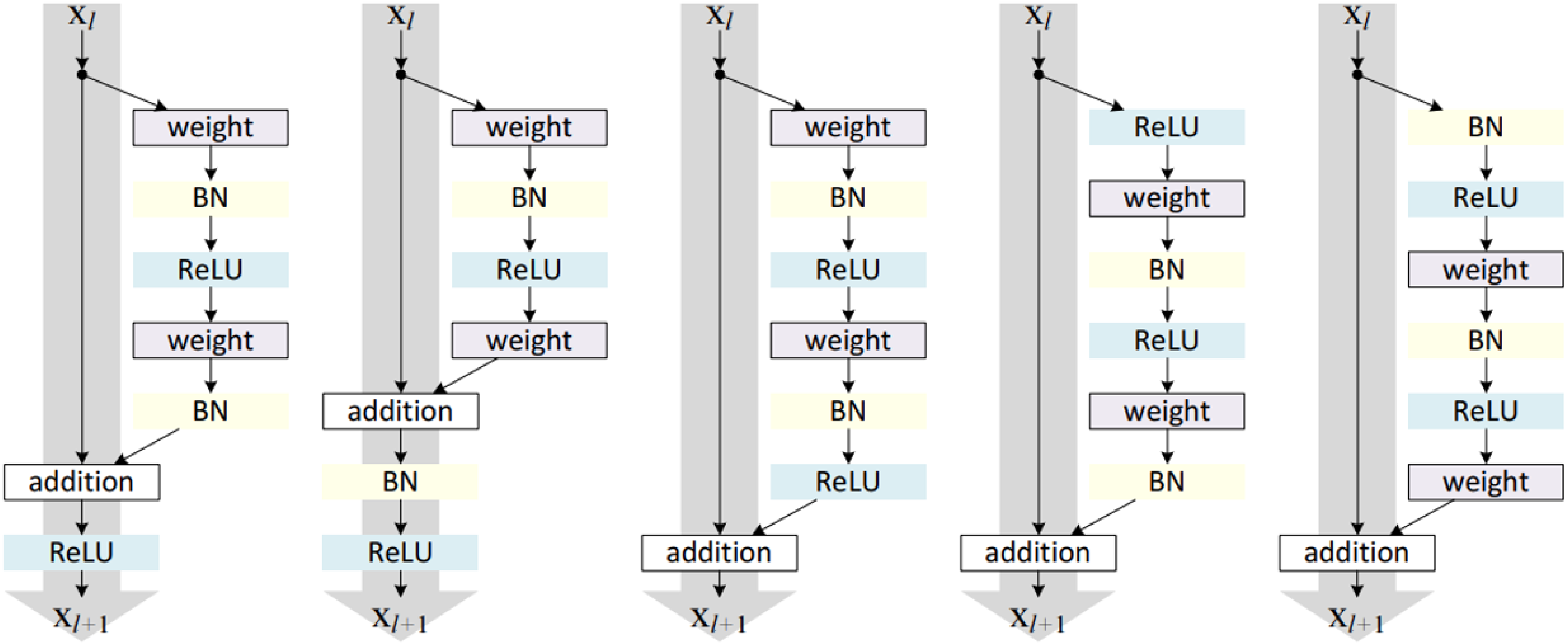
error : 8.17%

(c) ReLU before addition

error : 7.84%

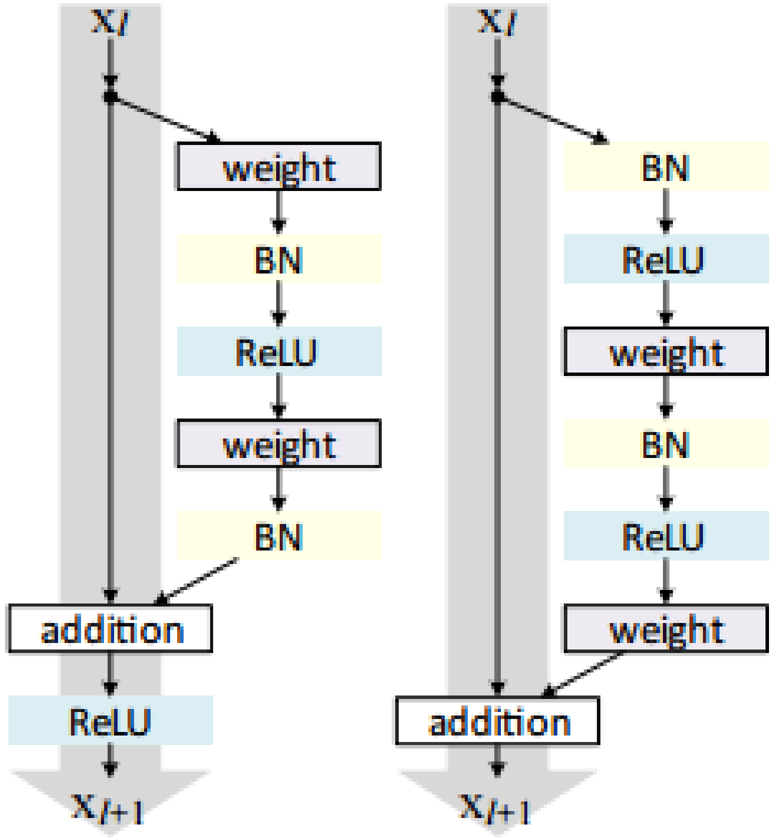


Pre-Activation ResNet

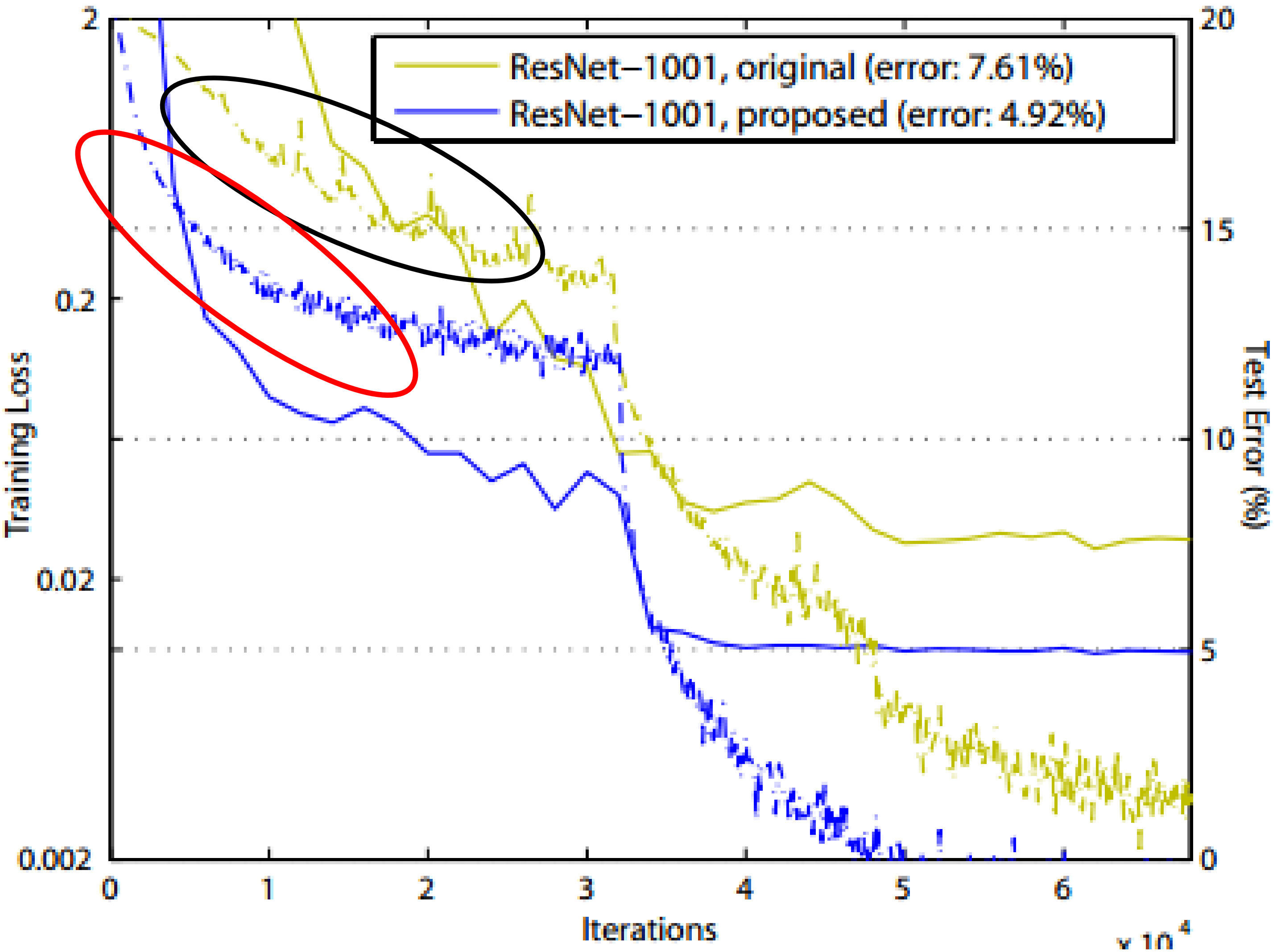


(a) original	(b) BN after addition	(c) ReLU before addition	(d) ReLU-only pre-activation	(e) **full pre-activation**
error : 6.61%	error : 8.17%	error : 7.84%	error : 6.71%	**error : 6.37%**

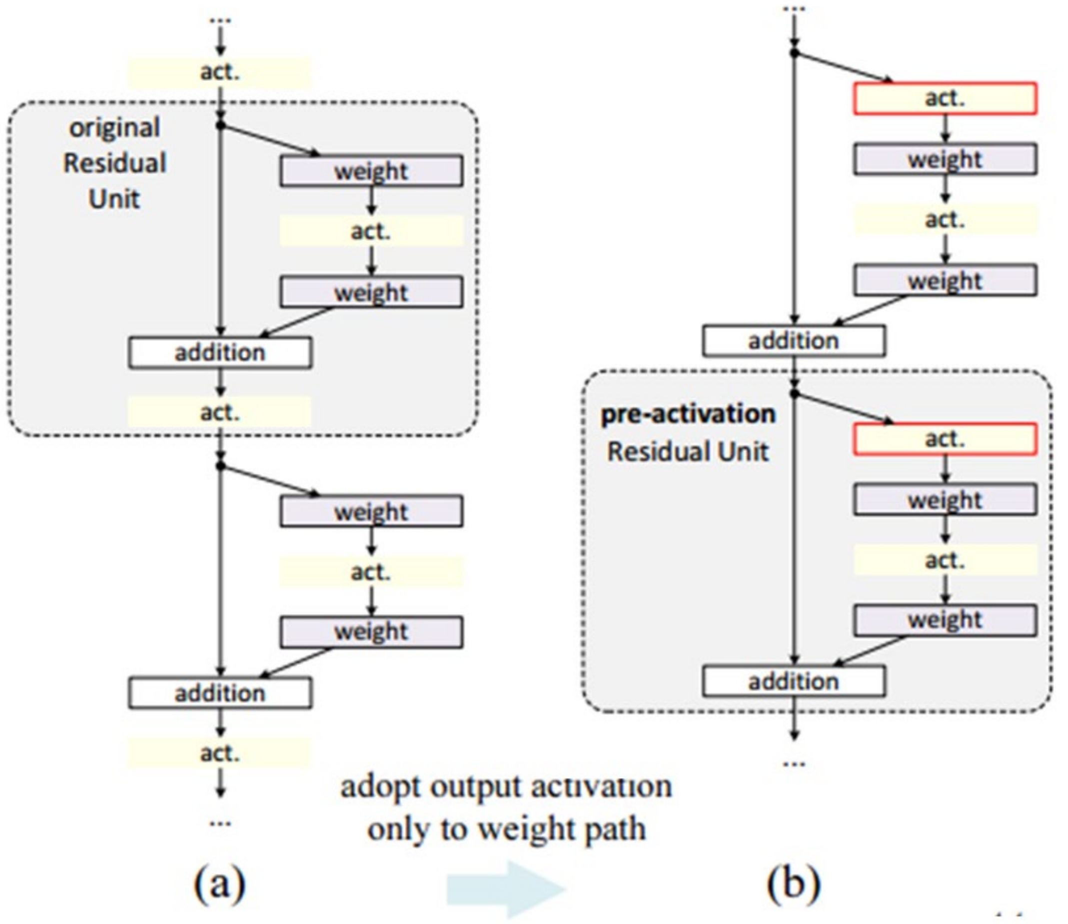
Pre-Activation ResNet

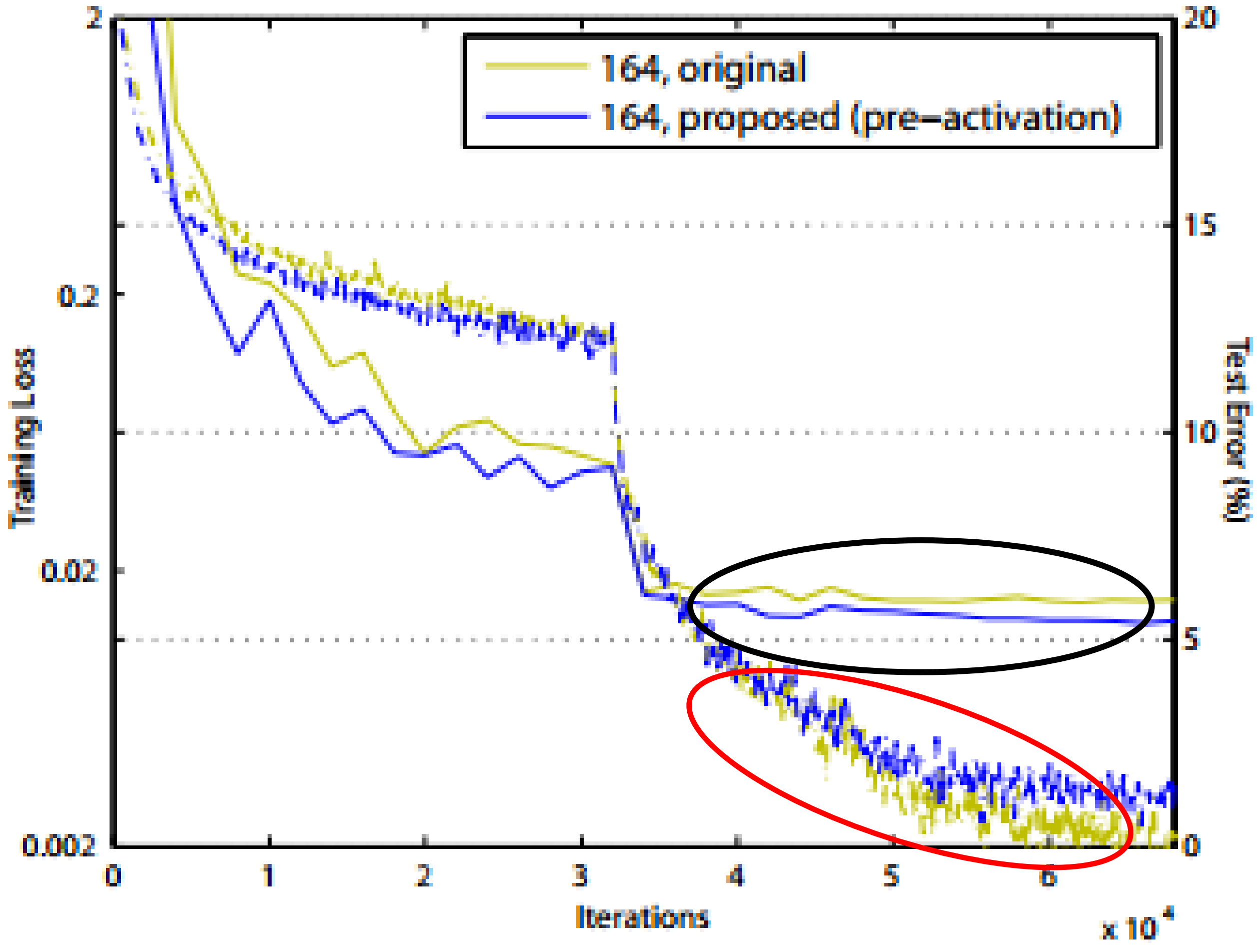


(a) original (b) proposed



Pre-Activation ResNet





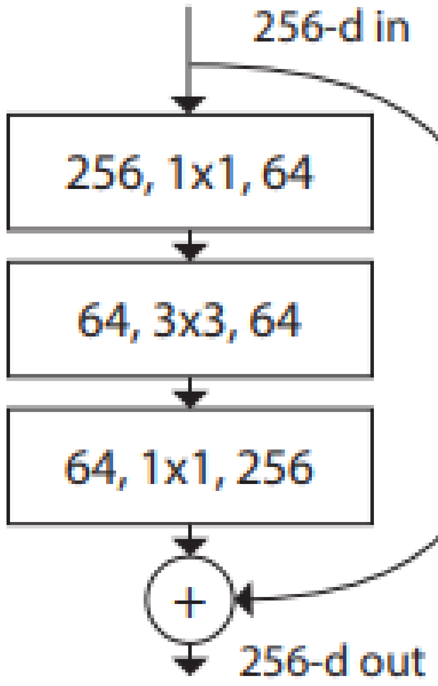
ResNeXt

arXiv
https://arxiv.org › cs

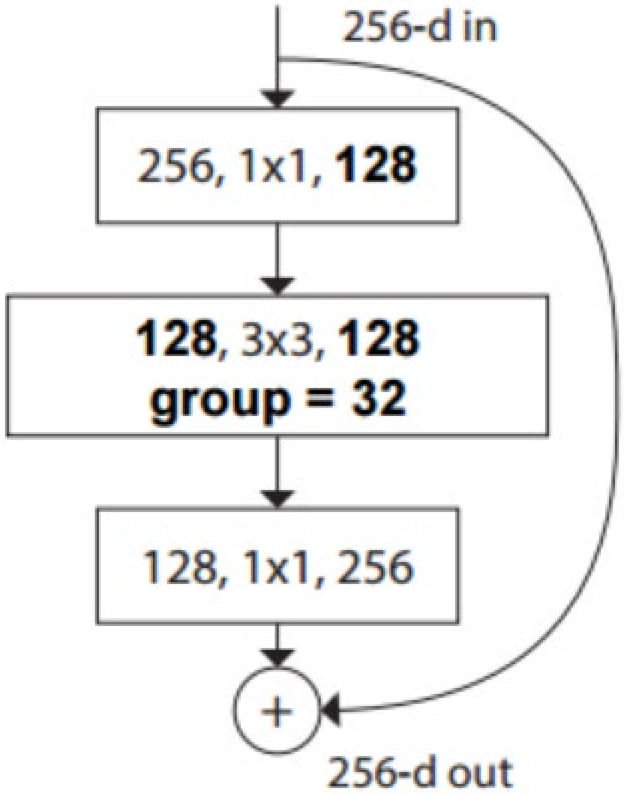
Aggregated Residual Transformations for Deep Neural ...

S Xie 저술 · 2016 · 10561회 인용 — We present a simple, highly modularized **network**

ResNeXt

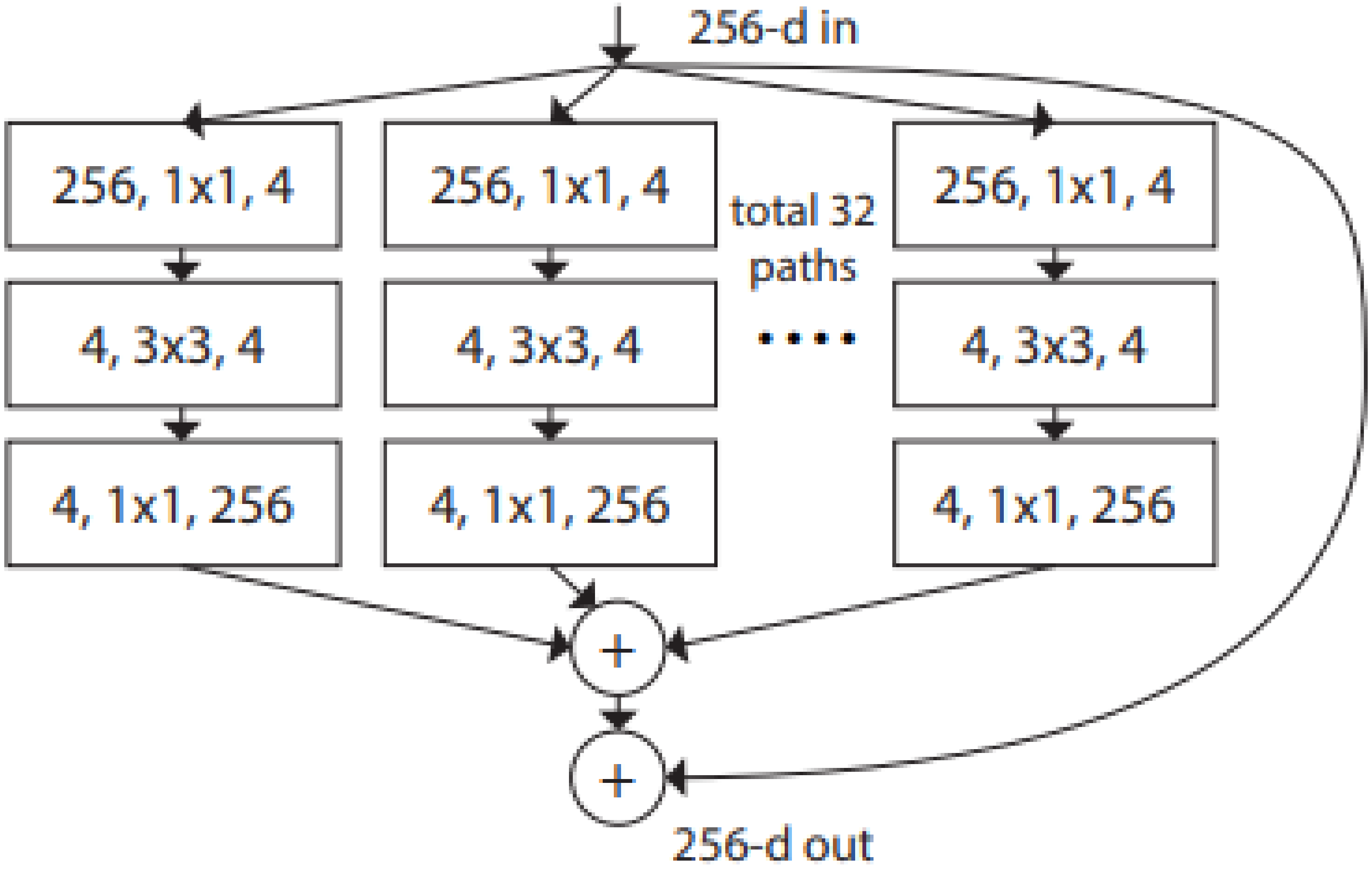


Deeper Bottleneck



ResNeXt

ResNeXt – Grouped Convolution



ResNeXt – Cardinality, Width

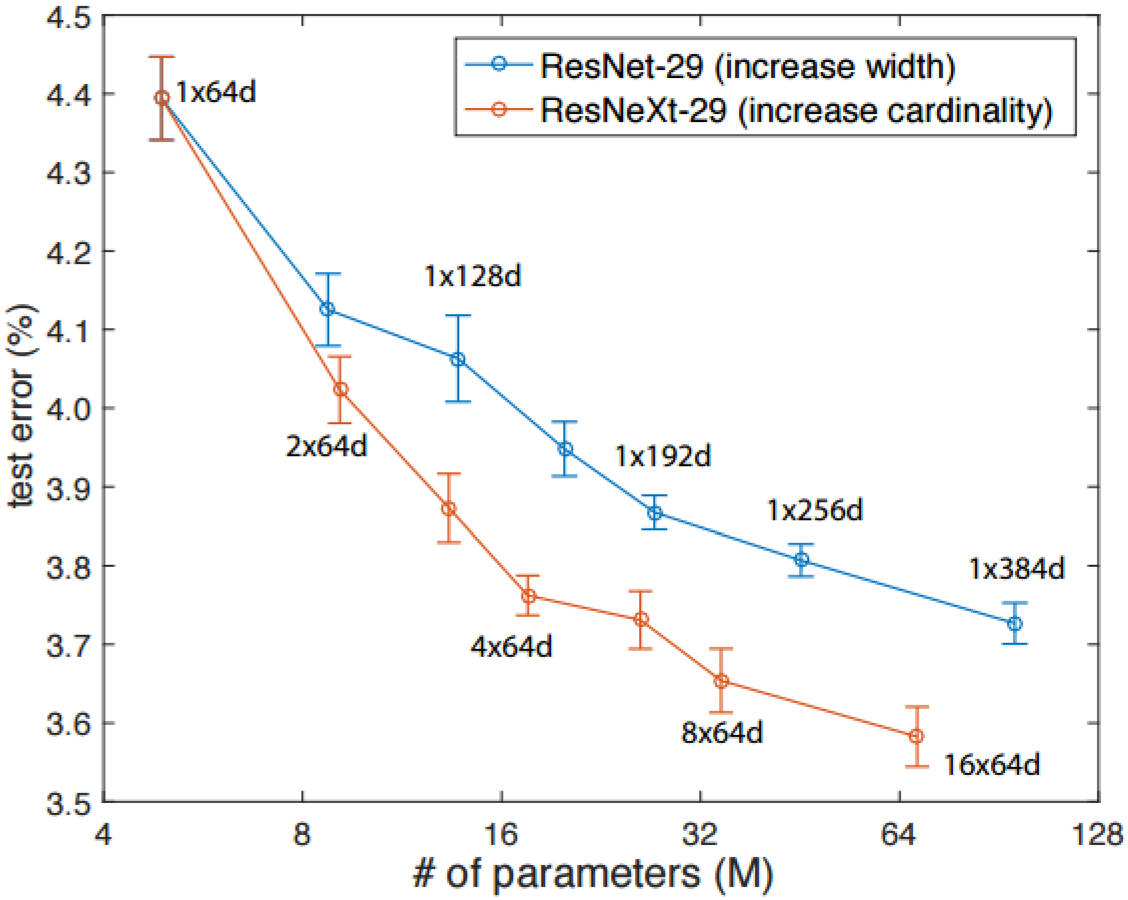
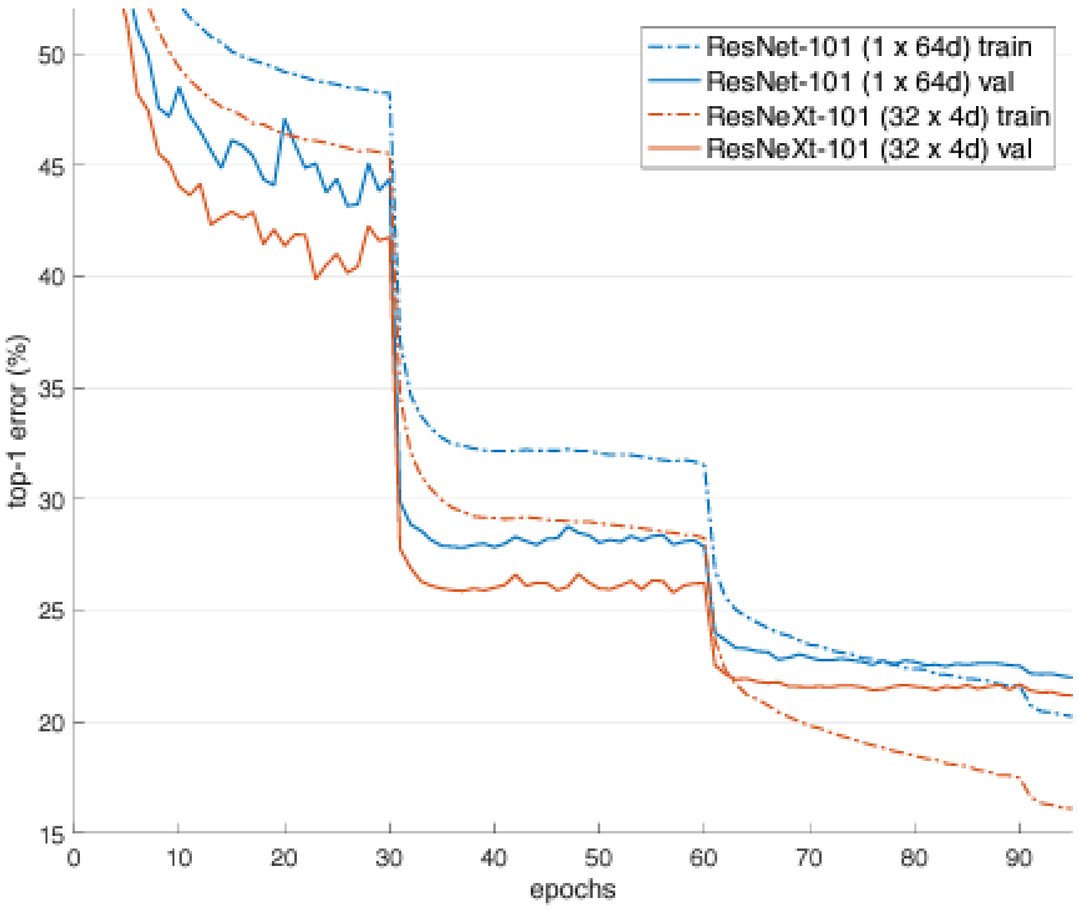


Figure 7. Test error *vs*. model size on CIFAR-10. The results are computed with 10 runs, shown with standard error bars. The labels show the settings of the templates.

- Cardinality: the number of groups to divide the total number of channels
- Width: the number of channels in one group
- Cardinality > Width > Depth

ResNeXt – Cardinality, Width



	setting	top-1 err (%)	top-5 err (%)
1× complexity references:			
ResNet-101	1 × 64d	22.0	6.0
ResNeXt-101	32 × 4d	21.2	5.6
2× complexity models follow:			
ResNet-**200** [15]	1 × 64d	21.7	5.8
ResNet-101, wider	1 × **100**d	21.3	5.7
ResNeXt-101	**2** × 64d	20.7	5.5
ResNeXt-101	**64** × 4d	**20.4**	**5.3**

Table 4. Comparisons on ImageNet-1K when the number of FLOPs is increased to 2× of ResNet-101's. The error rate is evaluated on the single crop of 224×224 pixels. The highlighted factors are the factors that increase complexity.

감사합니다