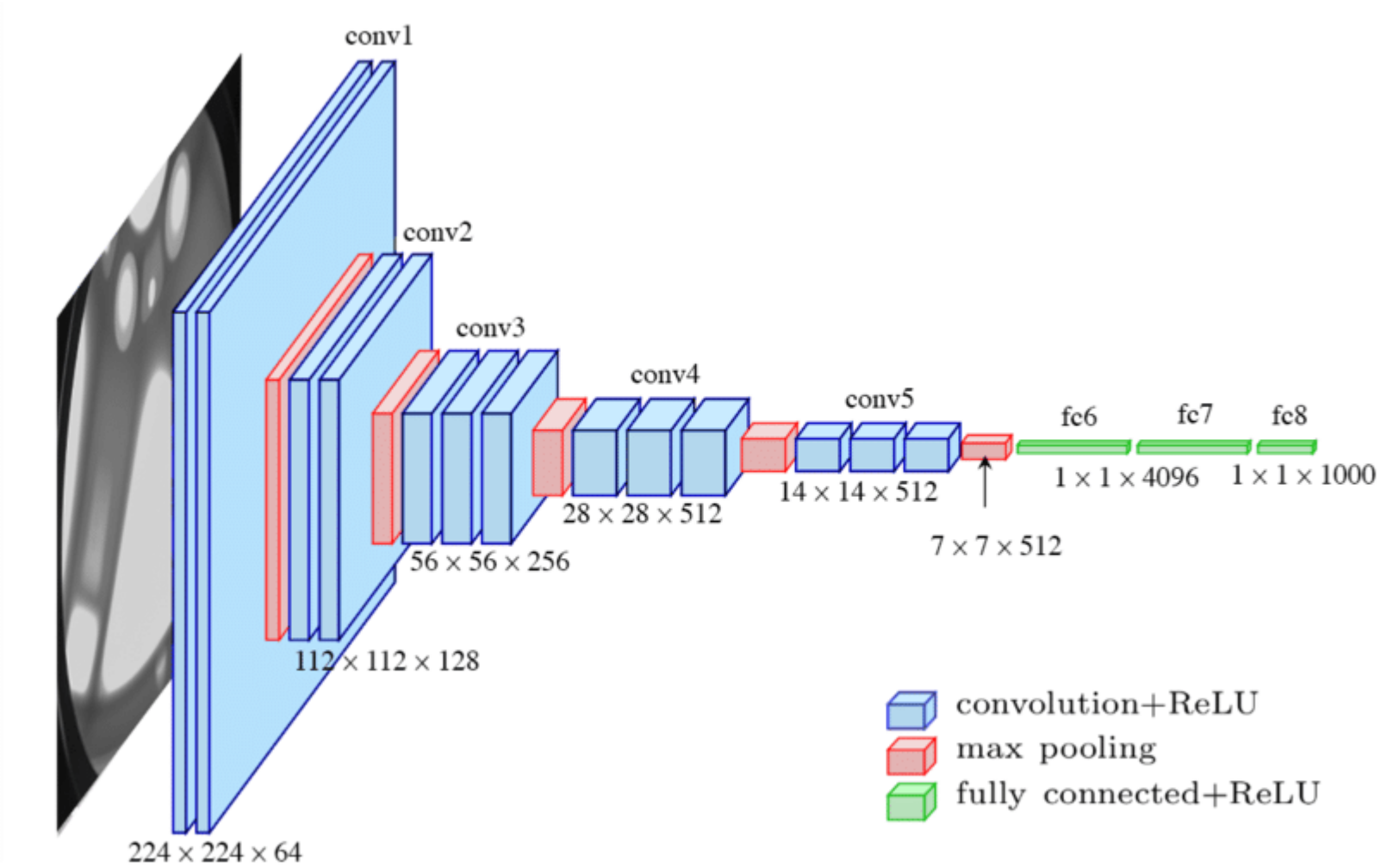
Deep Residual Learning for Image Recognition

Author	Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun
Field	CV
Journal	CVPR
# Published Year	2016
Speaker	강동규 이민지
Summary	ResNet
status	Finished!
link	https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf

Problem



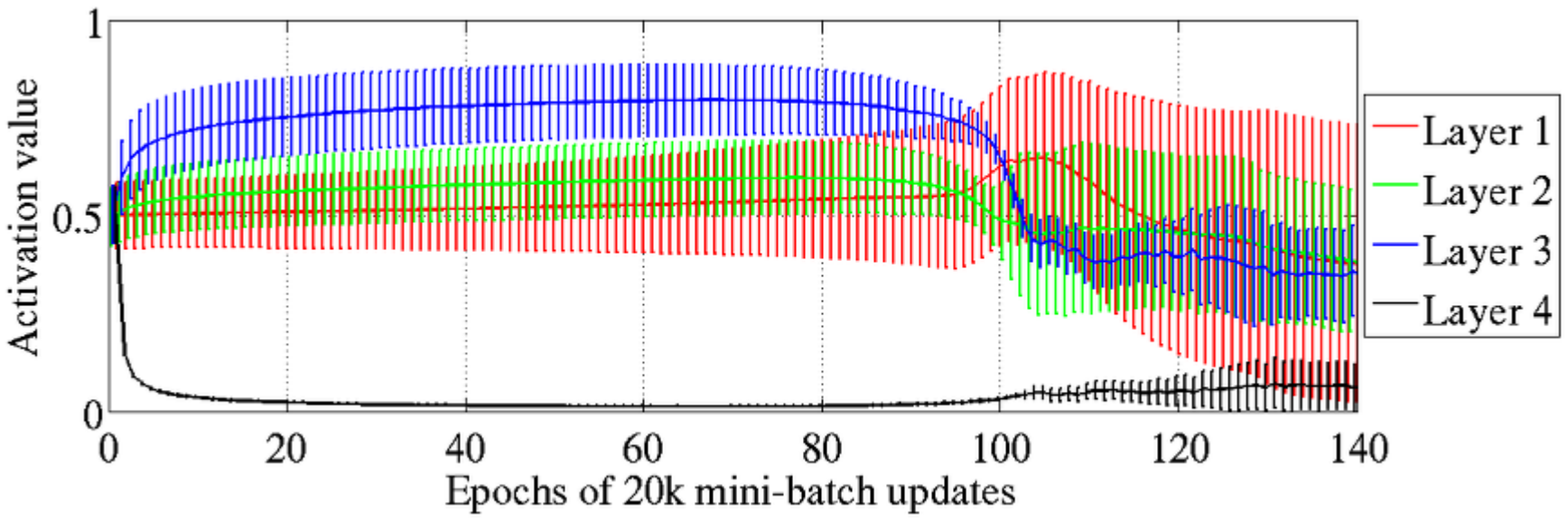
- Depth ⇒ Increase performance!!!

> ❓ *Is learning better networks as easy as stacking more layers?*

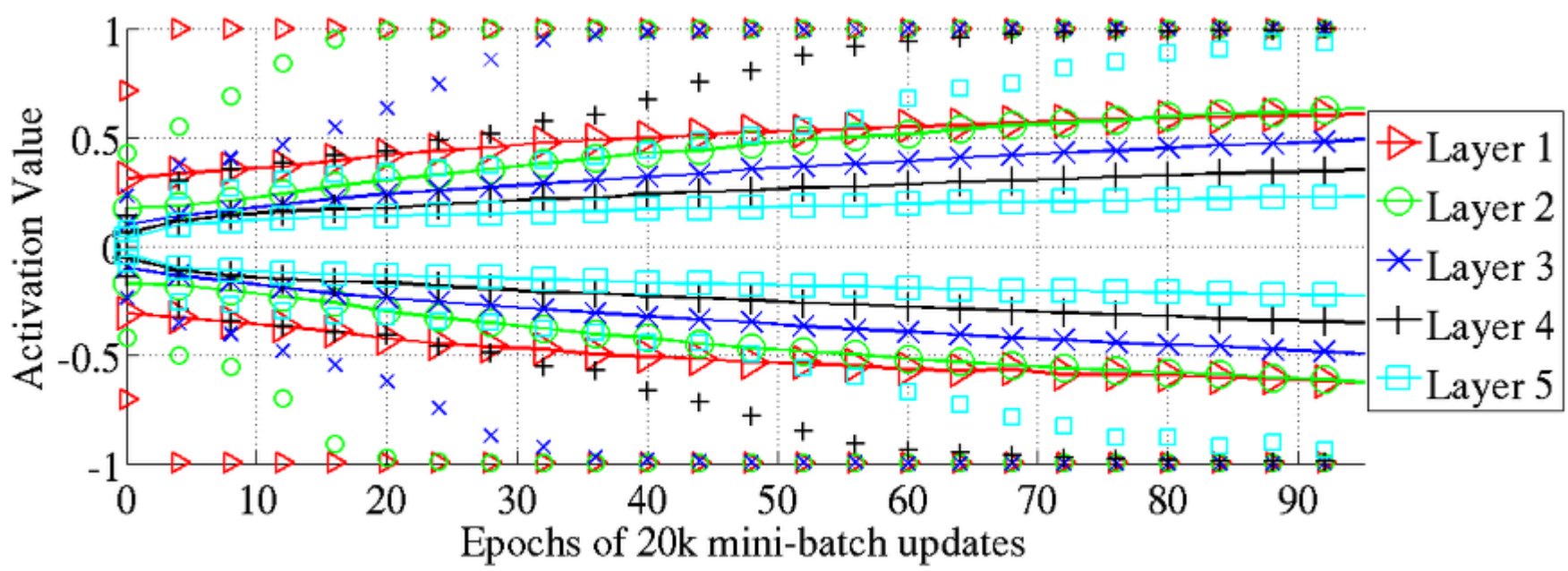
Vanishing/Exploding Gradient - Obstacle of converging

- Paper : Understanding the difficulty of training deep feedforward neural networks(2010)
- Simple network with 5 layers

- Graph for looking **saturation**



- Activation function : **Sigmoid**
- At **last hidden layer**, Layer 4 **saturates** very quickly to zero.



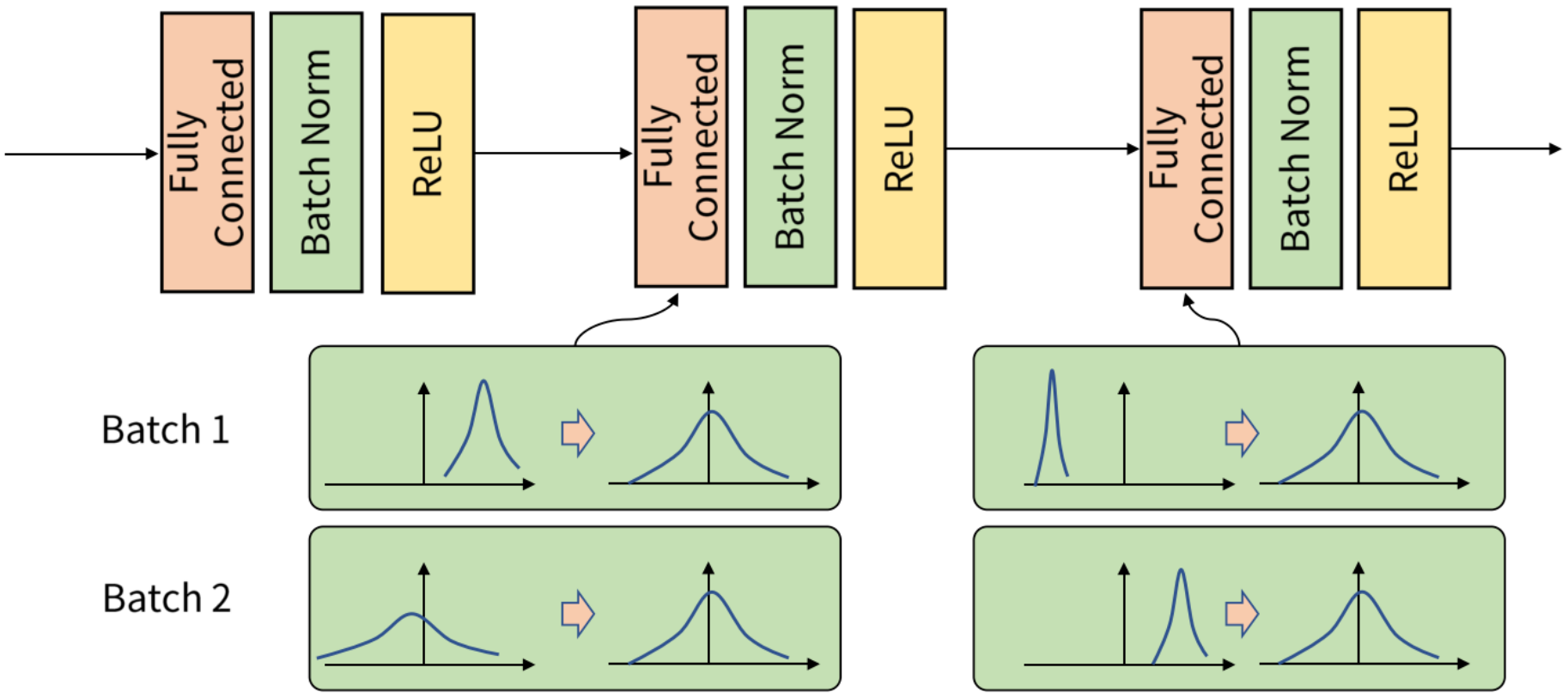
- Activation function : Hyperbolic Tangent [tanh(x)]
- The **deeper** the layer, the more **saturated** it becomes.

Effort to coverge

- Weight Initialization - Initialize the weight to the speficied range.

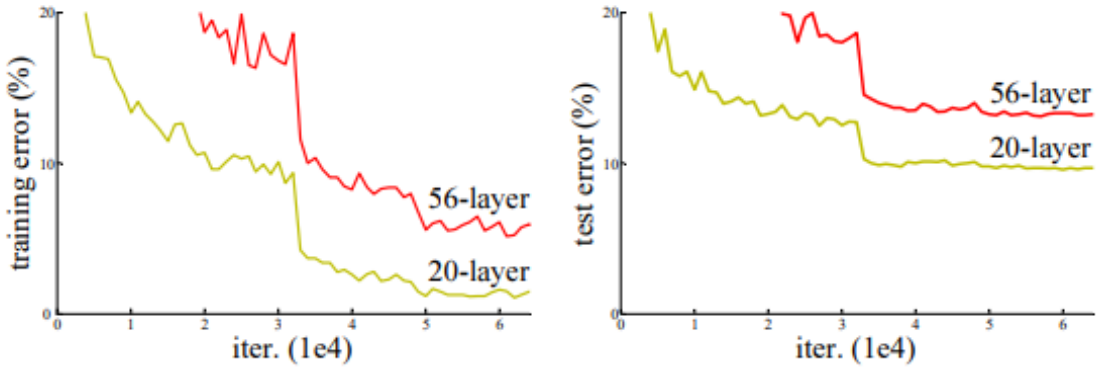
$$X \sim N(0, \frac{2}{fan_{in} + fan_{out}})$$

- Batch Normalization(2015) - Normalize intermediate weights to prevent saturation.



Reference : https://gaussian37.github.io/dl-concept-batchnorm/ (JINSOL KIM Blog)

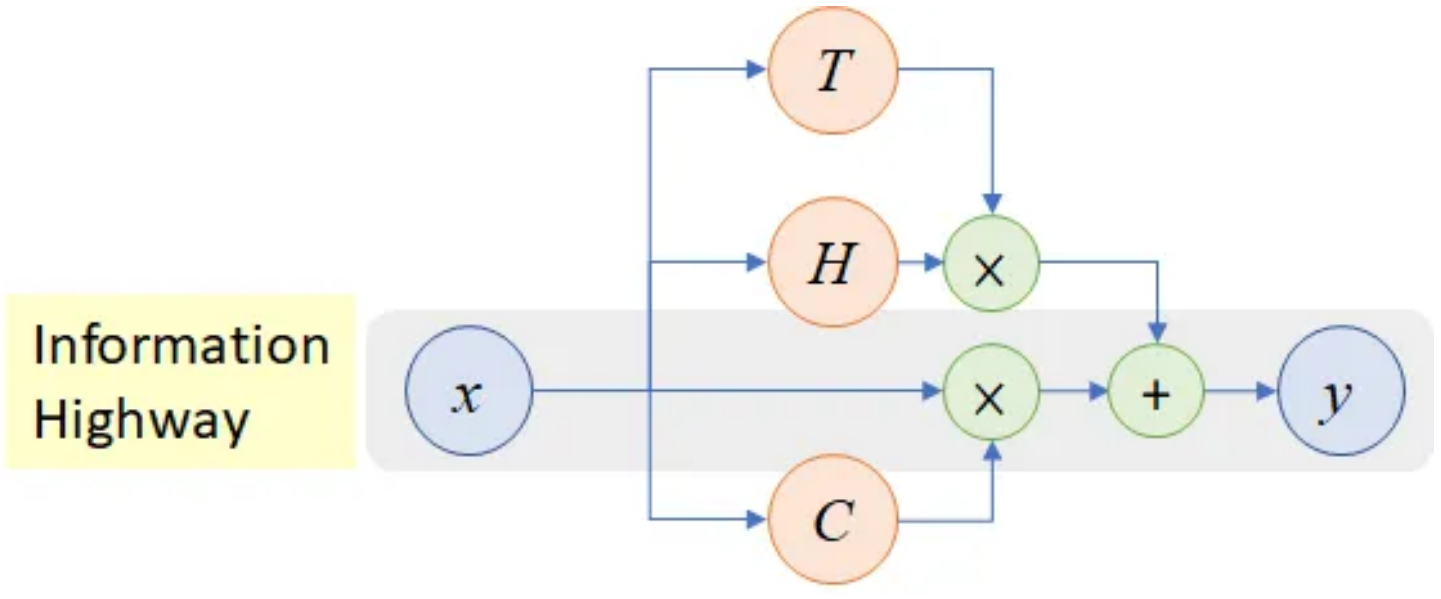
Degradation



- When **depth increasing**, accuracy gets **saturated** and then **degrades** rapidly.
- Adding more layers leads to higher error.
- The deeper the layer, the more difficult it is to optimize.

Related Work - Highway Network(2015)

💡 Make paths that flows without loss of information based on the gating mechanism of LSTM!



Reference : https://towardsdatascience.com/review-highway-networks-gating-function-to-highway-image-classification-5a33833797b5

- Highway equation

$$\mathbf{y} = H(\mathbf{x}, \mathbf{W_H}) \cdot T(\mathbf{x}, \mathbf{W_T}) + \mathbf{x} \cdot C(\mathbf{x}, \mathbf{W_C}).$$

- Simplify equation

$$\mathbf{y} = H(\mathbf{x}, \mathbf{W_H}) \cdot T(\mathbf{x}, \mathbf{W_T}) + \mathbf{x} \cdot (1 - T(\mathbf{x}, \mathbf{W_T})).$$

- $H(x, W_H)$: Linear transformation + Non-linear activation function
- $T(x, W_T)$: Transform Gate (Non-linear transform)
- $C(x, W_C)$: Carry Gate (Non-linear transform)

Pros

- When $T(x, W_T) \approx 0$, pass the input as output directly which creates an highway.
- Highway makes backpropagation flow better through the gate.

- Result

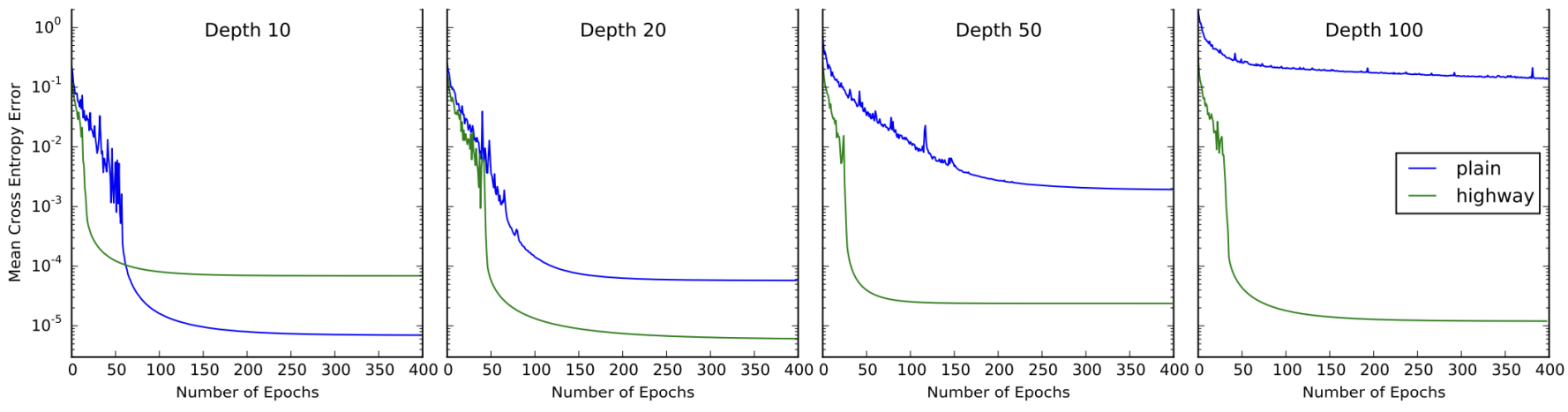


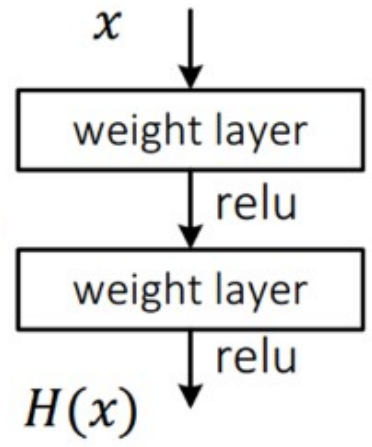
Figure 1. Comparison of optimization of plain networks and highway networks of various depths. All networks were optimized using SGD with momentum. The curves shown are for the best hyperparameter settings obtained for each configuration using a random search. Plain networks become much harder to optimize with increasing depth, while highway networks with up to 100 layers can still be optimized well.

Cons

- Gates are data-dependent and have parameters.
- When $C(x, W_C) \approx 0$, shortcut is “closed” ⇒ Equal to **plain layer**

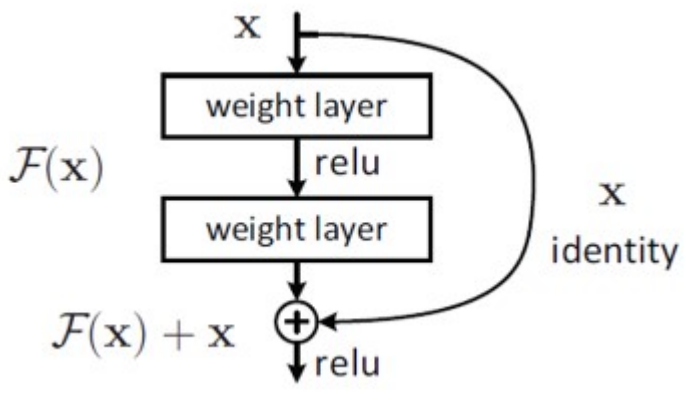
main idea

Identity Mapping/residual learning



기존 네트워크 : 입력 데이터 x를 받아 네트워크를 통해 H(x)를 출력

이때 H(x)는 입력 x를 원하는 목표값 y로 변환하는 함수



Residual Learning : 출력과 입력 차이인 H(x)-x를 최소화 하는 것, 즉 출력과 입력의 차이를 줄임

따라서 residual function인 F(x)=H(x)-x 를 최소화 하며, F(x)=0이 되는 것이 최적의 해가 된다.

F(x)=0 ⇒ H(x)=x 가 되고, 입력 x를 그대로 출력하는 함수 H(x)를 만드는 것이 학습 목표가 된다.

기존 네트워크에서는 H(x)를 최적의 값을 찾는 방향으로 만들었기에 어려움이 있었음

그러나 H(x)=x라는 최적의 목표 값이 사전에 주어져(pre-conditioning) Identity Mapping이 더 쉬움

⇒ why?

1. 사전 초기화된 목표 값 : 네트워크 학습 초기 단계에서 더 빠르게 수렴할 수 있음
2. shortcut 연결 : 입력과 출력을 직접적으로 연결하여 그래디언트를 효과적으로 전파하고 소멸 문제를 줄여줌
3. H(x) = x로 설정된 목표 값은 더 작은 오차를 가짐. 목표 값이 더 작은 오차를 가지면, 최적화 알고리즘은 빠르게 수렴하고 안정적으로 학습할 수 있음

F(x)=H(x) - x ⇒ H(x) = F(x) +x 이므로 네트워크 구조는 입력과 출력을 직접적으로 연결하는 shortcut 연결만 추가하면 됨. 이 연결은 파라미터와 연산량에 영향을 미치지 않음

shortcut 연결은 곱셈 연산이 덧셈 연산으로 변환되어 forward와 backward 경로가 단순해지고, 그래디언트 소멸 문제를 해결할 수 있다.

이를 식으로 증명하면,

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

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

x,y : 입력, 출력

아랫첨자 l : 레이어 번호

W : 가중치 행렬

F : Residual Function

f : activation fuction(=relu)

h : identity mapping을 위한 dimension 세팅용 함수

함수 f를 identity mapping이라고 생각하면, x_l+1 = y_l로 둘 수 있고, 이를 대입하면

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

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

즉, 순전파 상황에서 전체 네트워크를 residual function인 F의 합으로 나타낼 수 있다.

$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\frac{\partial \mathbf{x}_L}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\left(1 + \frac{\partial}{\partial \mathbf{x}_l}\sum_{i=l}^{L-1}\mathcal{F}(\mathbf{x}_i, \mathcal{W}_i)\right). \tag{5}$$

https://m.blog.naver.com/siniphia/221387516366

양변의 미분을 적용하면 역전파 상황에서 전체 네트워크를 알 수 있는데(ε은 Loss Function)

(1)**dε/dxL**

: 우변의 앞에 곱해져 있는 이 식은 ε에 대한 마지막 레이어의 미분이므로 중간에 어떤 가중치 행렬도 거치지 않기에 Vanishing이 발생하지 않는다.

(2)**괄호 안의 식**

: 이 경우 Wi라는 가중치 행렬들을 지나기는 한다. 하지만 학습 과정에서 괄호안의 우변이 Mini-batch마다 항상 -1이 되어 전체 식이 0이 되어 버리는 일은 거의 없기 때문에 역시 Vanishing이 발생할 확률은 극히 낮다.

즉, 순전파 상황 ⇒ 곱셈이 아닌 덧셈으로 쉬운 정보 이동

역전파 상황 ⇒ 가중치 행렬들의 곱셈으로 정보가 전달되는 것이 아니므로 vanishing 문제가 발생하지 않음

따라서 네트워크 깊이의 한계를 극복할 수 있음

Implementation

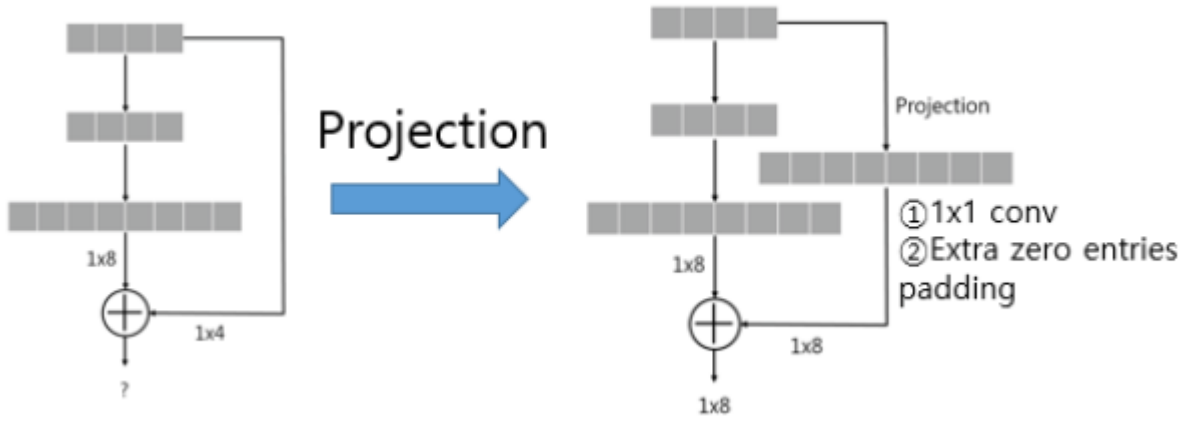
Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

plain network

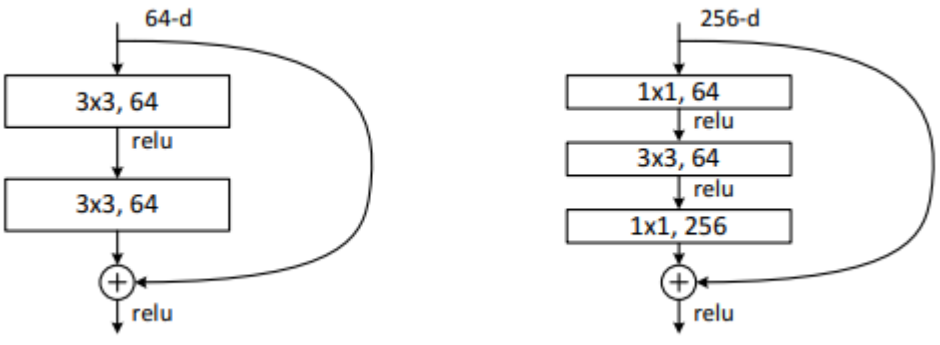
- vggnet을 참고하여 만듦. 3x3 conv, stride2, gap, 1000-way-fc layer 사용
- 최종 레이어 34개

residual network

- plain net에 shortcut connection 개념 도입
- identity shortcut은 input과 output을 같은 차원으로 맞춰야 함
  - 차원을 늘리기 위해 0을 넣어서 padding 하기
  - projection shortcut 사용하기 (1x1 conv, filter size=output 기준으로)



Deeper Bottleneck Architectures



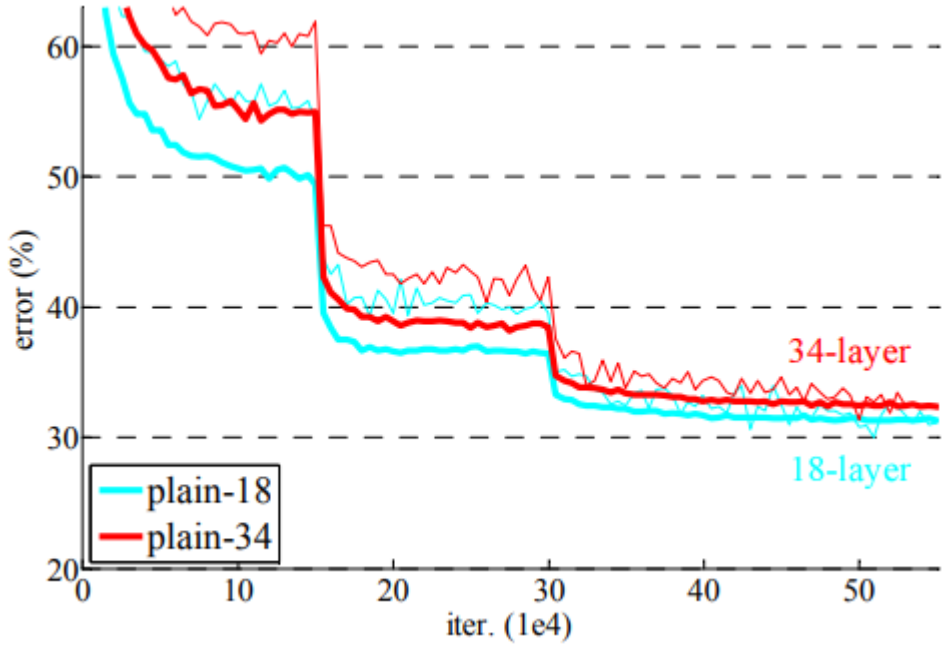
identity shortcut을 사용하기 위해 projection shortcut을 사용하는 것 대신에 1x1 conv로 차원을 맞춰줌.

이를 통해 모델의 크기와 연산량을 줄일 수 있었음

Result

1) 18 layer vs. 34 layer plain net

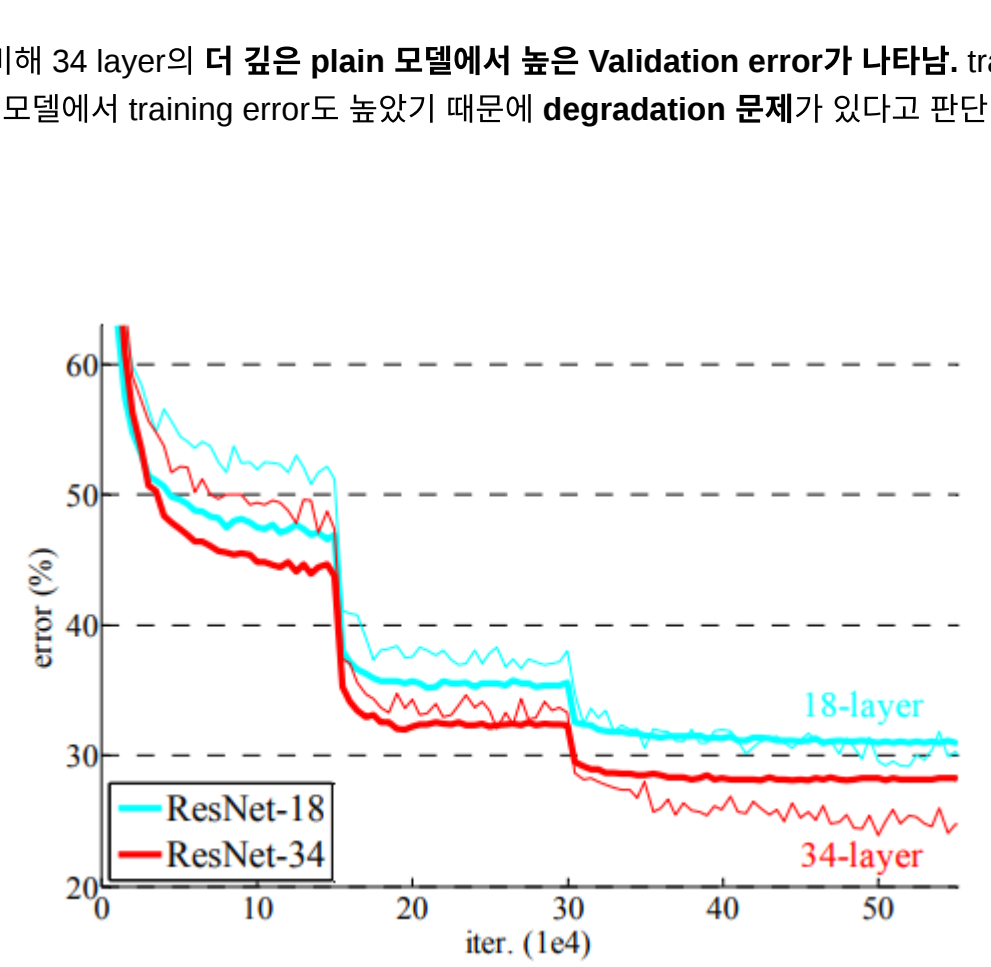
- 18 layer > 34 layer



- 18 layer의 얕은 plain 모델에 비해 34 layer의 **더 깊은 plain 모델에서 높은 Validation error가 나타남.** training / validation error 모두를 비교한 결과, 34 layer plain 모델에서 training error도 높았기 때문에 **degradation 문제**가 있다고 판단

2) 18 layer vs. 34 layer Resnet

- 18 layer < 34 layer



- 4-layer ResNet에서 상당히 낮은 training error를 보였고, 이에 따라 validation 성능 또한 높아짐. 이는 **degradation 문제가 잘 해결되었으며, depth가 증가하더라도 좋은 정확도를 얻을 수 있음을 의미**

Identity vs. Projection shortcut

residual connection에 대한 여러 옵션을 비교한 실험

A) Zero-padding shortcut

- zero padding을 사용하여 입력과 출력의 차원을 일치시킴. 모든 shortcut은 파라미터를 가지지 않고, identity 연결

B) Projection shortcut

- 입력과 출력의 차원이 다를 때에만 projection shortcut을 사용하여 차원을 일치시킴. 다른 모든 shortcut은 identity 연결

C) All Projection

- 모든 residual 연결을 projection shortcut으로 설정

결과

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	**21.43**	**5.71**

C>B>A

이유

Zero padding은 파라미터를 갖지 않으므로 residual learning이 잘 이루어지지 않았음. C는 가장 많은 파라미터를 가지므로 성능이 가장 우수하였으나 성능 대비 많은 자원을 사용하여 실험에서는 사용하지 않았음