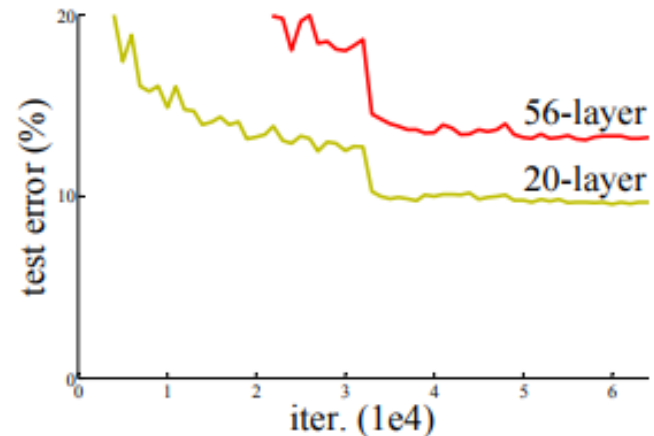
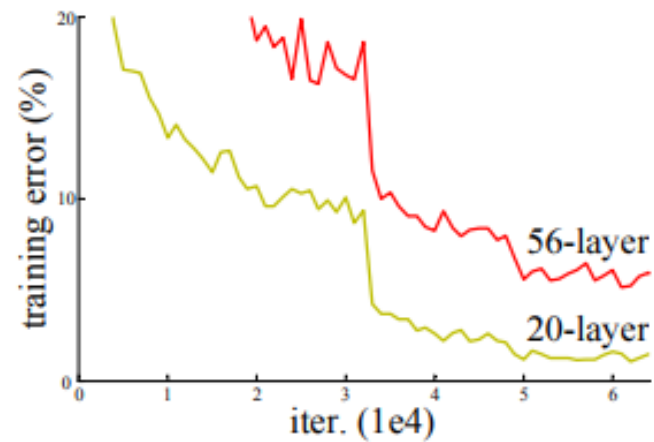


# Deep Residual Learning for Image Recognition

# Background

- 컴퓨터 비전 작업에서 모델이 깊어질 수록 더 좋은 성능을 낸다. 하지만 **모델이 깊어질 수록 학습이 잘되지 않는 문제가 있다**. 이 문제를 Gradient vanishing/ Exploding과 Degradation 이다.
- Degradation 문제**는 정확도가 어느 순간부터 정체되고 레이어가 더 깊어질수록 성능이 더 나빠지는 현상을 의미한다. 이는 과적합과 다르게 **학습 에러가 높을 때, 테스트 에러도 높아지는 것이다**.
- 오른쪽 그림을 보면 깊이가 증가할 때 학습과 테스트의 에러 둘 다 감소하는 것을 볼 수 있다.



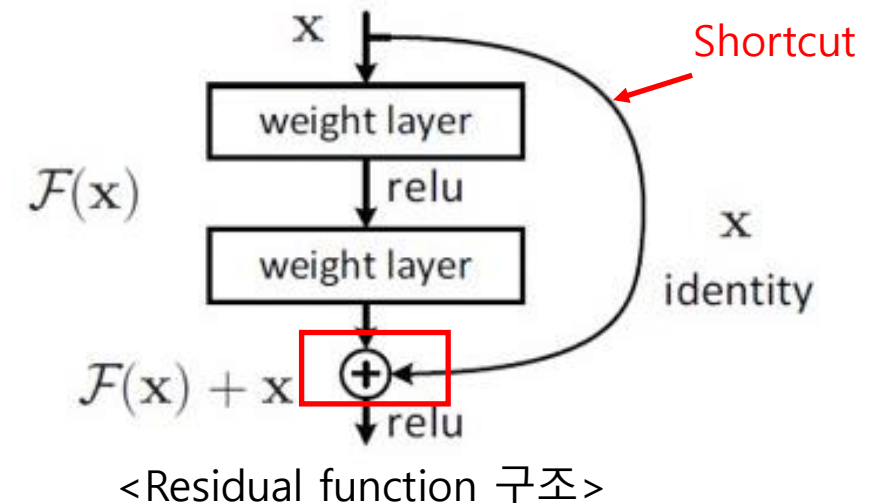
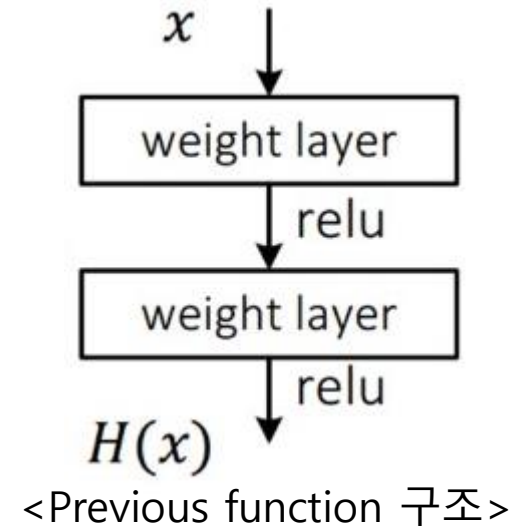
<VGG기반 신경망 깊이에 따른 학습과 테스트 에러>

# Motivation

- VLAD와 Fisher Vector 등 다른 사례에 대해 설명하고 있는데, 핵심 내용은 벡터 양자화에 있어 **residual vector**를 인코딩하는 것이 **original vector**보다 훨씬 효과적이라는 것이다. 여기서 벡터 양자화란 특징 벡터  $X$ 를 클래스 벡터  $Y$ 로 mapping 하는 것을 의미한다.

# Model architecture

- 기존 방식은 입력 값을 참고하지 않는 unreference 방식이다. 입력 값을 참고 하여 학습을 쉽게 하는 방법을 설계
- **입력 값과 출력 값을 연결**하는 Shortcut과 **입력 값과 출력 값을 더하는** 것이 residual function이다. 이는 별다른 연산 없이 입력과 출력을 더해주기 **파라미터가 증가하지 않는다**.
- 이런 구조는  $F(x) = H(x) - x$ 로 표현된다. 즉, **출력과 입력의 차를 줄인다는** 의미가 된다. 여기서  $x$ 의 값은 도중에 바뀌지 못하는 입력 값이므로  $F(x)$ 가 0이 되는 것이 최적의 해이고, 결국  $0 = H(x) - x$ 로  $H(x) = x$ 가 된다. 즉,  $H(x)$ 를  $x$ 로 mapping 하는 것이 학습의 목표가 된다.



# Model architecture mathematical formula

- Residual function을 사용하면 Feedforwarding 과정이 곱셈 연산에서 덧셈 연산으로 바뀐다.
- 그리고 Feedforwarding 과정 양변을 미분하여 backpropagation 과정을 보면,
  - $d\epsilon/dx_l$   $\epsilon$ 에 대한 **마지막 레이어의 미분이므로** 중간에 어떤 가중치 행렬도 거치지 않기에 Vanishing이 발생하지 않는다.
  - 괄호안의 우변이 Mini-batch마다 **항상 -1이 되어 전체 식이 0이 되어버리는 일은 거의 없기** 때문에 역시 Vanishing이 발생할 확률은 극히 낮다.

$$y_l = h(x_l) + \mathcal{F}(x_l, \mathcal{W}_l), \\ x_{l+1} = f(y_l).$$

$$x_{l+1} = x_l + \mathcal{F}(x_l, \mathcal{W}_l).$$

$$x_L = x_l + \sum_{i=l}^{L-1} \mathcal{F}(x_i, \mathcal{W}_i),$$

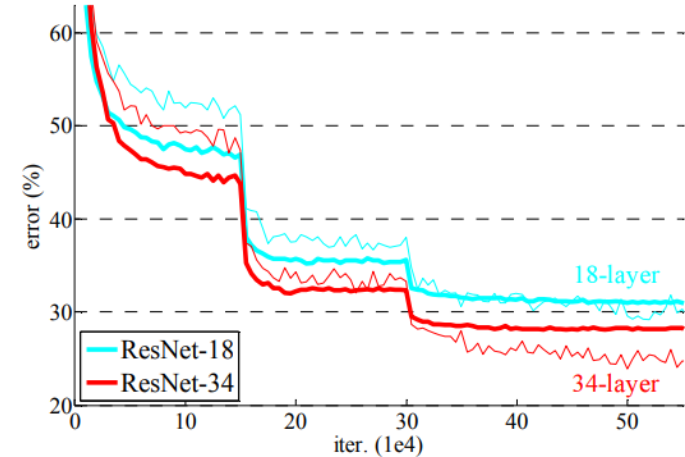
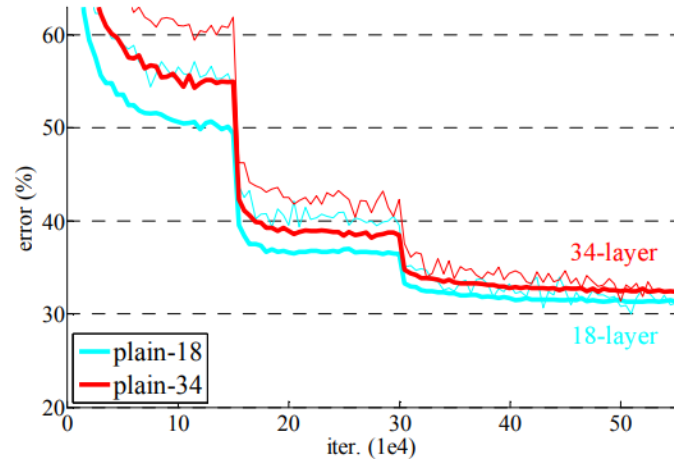
<Feedforwarding 연산>

$$\frac{\partial \mathcal{E}}{\partial x_l} = \frac{\partial \mathcal{E}}{\partial x_L} \frac{\partial x_L}{\partial x_l} = \frac{\partial \mathcal{E}}{\partial x_L} \left( 1 + \frac{\partial}{\partial x_l} \sum_{i=l}^{L-1} \mathcal{F}(x_i, \mathcal{W}_i) \right)$$

<Feedforwarding 양변 미분한 수식>

# Result

- Residual function을 통해 모델이 깊어져도 학습이 잘되게 하고자 했다. 이를 위해 VGG 기반 plain과 ResNet의 깊이를 증가시키며 학습과 테스트의 에러를 비교했다.
- 그 결과 ResNet은 모델이 깊어져도 학습이 잘되고, 테스트 또한 Plain 보다 좋게 나왔다.
- 그리고 얼마나 모델이 깊어질 수 있는지 실험하였을 때, 152개의 레이어를 쌓을 수 있었다.



<Plain, ResNet depth에 따른 실험 결과>

method	top-1 err.	top-5 err.
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>

<ResNet depth에 따른 실험 결과>