



# ResNet

Deep Residual Learning for Image Recognition

Residual 개념을 도입!

## ▼ Abstract



딥러닝에서 모델의 깊이가 깊어질수록 성능이 저하되는 현상 발생(Overfitting, Vanishing Gradient, 연산량 ↑, Optimization ↓)

- **Residual의 개념을 도입**함으로써 깊은 네트워크를 쉽게 학습시킬 수 있도록 함!
- Residual Learning : 이전 layer의 결과를 다시 이용하는 것!

## ▼ Introduction



**Deep Residual Learning Framework를 통해 degradation 문제를 해결**

- Residual Mapping을 이용 :  $H(x) = F(x) + x$

→ Short-cut Connection = Skip Connection

두개의 layer를 건너뛸!  $x$ 는 identity mapping을 거치고, 이후 나온 결과를  $F(x)$ 에 더해 줌!

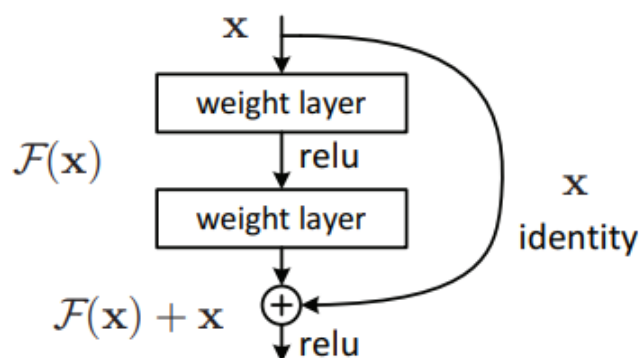


Figure 2. Residual learning: a building block.

- 계산을 늘리거나 parameter 수를 증가시키지 않으면서 optimization을 잘 시키며, depth가 증가할때 정확도를 증가시킴!

## ▼ Deep Residual Learning

### 1. Residual Learning

$H(x)$  : 몇개의 layer에 대한 Underlying Mapping,  $x$ : 입력

→ Residual Function  $F(x) = H(x) - x$ ,  $H(x) = F(x) + x$

→ residual learning reformulation으로 identity mapping이 optimal하면, solver는 weight를 얻기 쉬울것

→ 실제로 identity mapping이 optimal 할 것 같지 않지만, 적합한 함수가 identity mapping과 유사할때 작은 변화에 더 쉽게 반응할 수 있을 것!

### 2. Identity Mapping by Shortcuts

- Residual Block :

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

→ 모든 few stacked layer 마다 residual mapping을 적용

$F+x$  : shortcut connection, element-wise addition

$x$ 와  $F$ 이 차원이 같을 것!

- Residual Block 수식 변형:

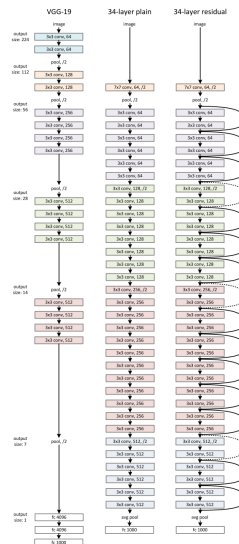
$$y = \mathcal{F}(x, \{W_i\}) + W_s x.$$

→  $x$ 와  $F$ 의 차원이 다를 경우, 차원을 맞추기 위해 shortcut connection에 linear projection  $W_s$ 를 적용!

- 논문에서는  $1*1$  conv layer를 활용해  $F$ 와 차원을 맞춰줌!

### 3. Network Architectures

- Plain Network - VGG-19 아키텍처에 영감 &  $3*3$  conv filter 사용
- Residual Network - plain network에 shortcut connection을 삽입



→ Input과 Output의 dimension이 다를 경우,

1. Identity mapping 그대로 진행, 나머지 dimension 0으로 채우기
2. projection shortcut 사용

#### 4. Implements

#### ▼ Experiments

##### 1. ImageNet Classification

- residual networks에서 dimension을 늘리기 위해 **zero padding** 사용!  
→ 성능 악화 문제 해결, 초반 단계에서 빠르게 convergence
- identity vs projection shortcuts  
: dimension 증가를 위해 zero padding 대신 **projection**을 이용!
- **Deeper Bottleneck**  
: 모델 깊게 만드려는데 학습시간 오래 걸리는거 줄이기 위해 **Bottleneck 구조** 제안

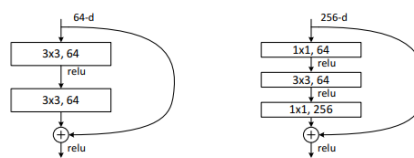


Figure 5. A deeper residual function  $\mathcal{F}$  for ImageNet. Left: a building block (on  $56 \times 56$  feature maps) as in Fig. 3 for ResNet-34. Right: a “bottleneck” building block for ResNet-50/101/152.

- 1 x 1 layer를 통해 dimension 줄였다가 키워짐
  - Identity Shortcut이 bottleneck 구조에서 효율적인 구조를 위해 꼭 필요!
- #### 2. CIFAR-10 and Analysis
- ResNet이 모델이 깊어질 수록 더 좋은 성능을 보이는 것을 확인
  - 더 적은 파라미터 수를 가지면서 더 높은 성능