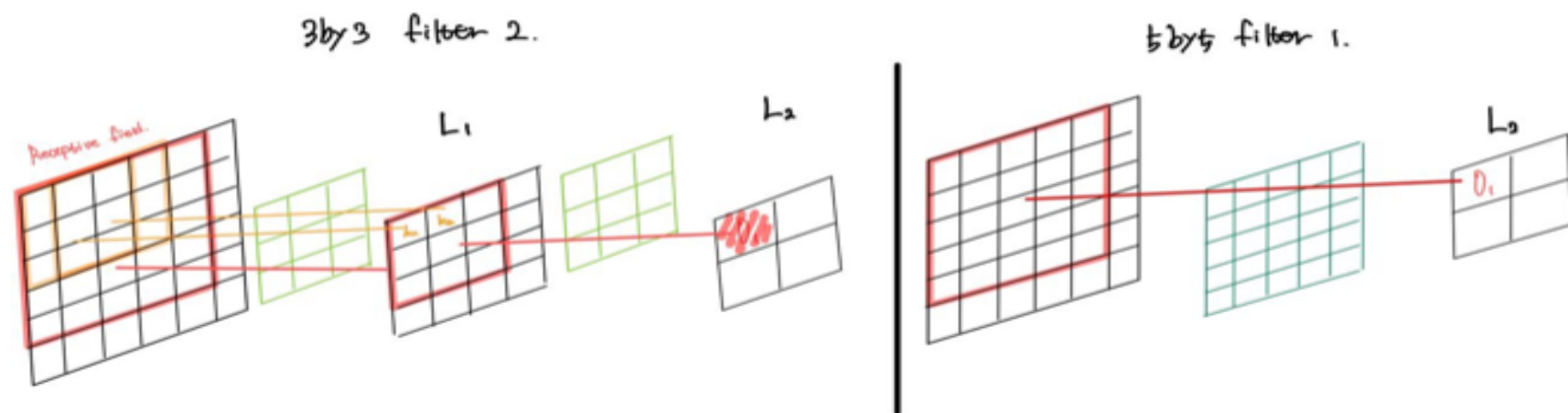


VGGNet

VGGNet

- 3by3 필터만을 사용
- Receptive field를 유지하면서 파라미터의 갯수 최소화.

Receptive field : 하나의 필터의 값을 얻기 위해서 고려해야 하는 입력 데이터의 영역



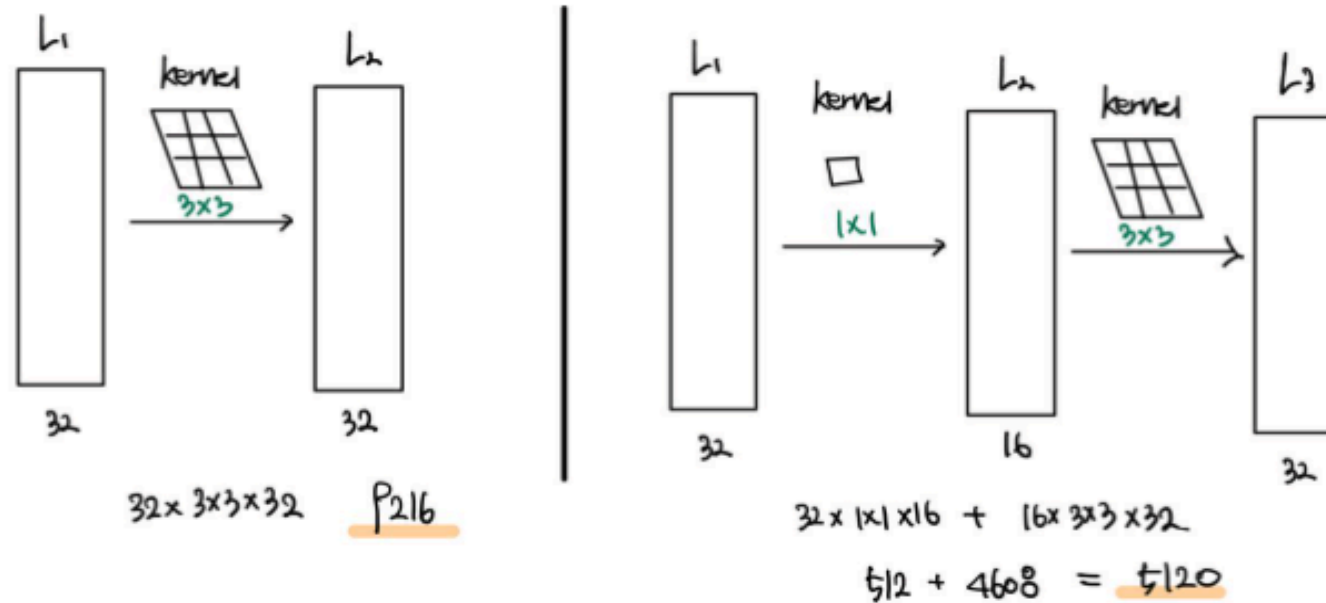
위에서 파라미터를 줄이고 신경망을 깊게 쌓아야 한다는 맥락에 맞는 방법론이다. 3by3 커널을 2개 쌓는 것과 5by5 커널 하나를 쌓을 때 둘다 하나의 입력에 대해 **Receptive field**가 동일하다. 그만큼 동일 출력값에 대한 정보량이 동일하다고 볼 수 있다. 하지만 둘의 파라미터 수를 센다면 다르다. 3by3의 경우 각각 9개씩해서 18개의 파라미터를 갖는다. 하지만 5by5의 경우 25개의 파라미터가 필요하다. 그림의 예시는 단일 채널이지만, 채널의 수가 많아진다면 차이는 그만큼 커지는 것을 알 수 있다. VGGNet은 3by3 커널을 연속해서 쌓아 파라미터의 수를 줄이는데 기여 하였다.

GoogLeNet

GoogLeNet

- inception block
 - 하나의 입력이 들어왔을때 여러개로 퍼졌다가 다시 합쳐진다
- 1 by 1 convolution!! 이 중요..
 - 전체적인 네트워크의 파라미터가 줄어 든다.
- network in network

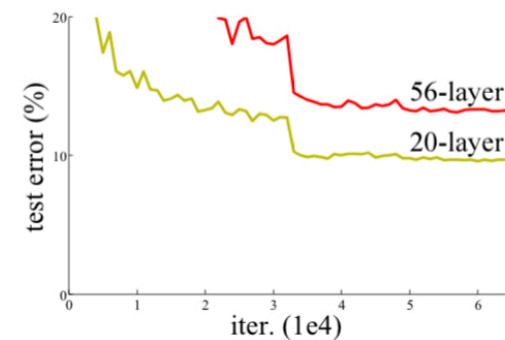
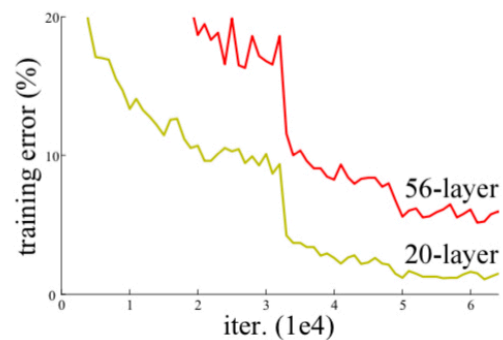
1 by 1 convolution(dimension reduction)



ResNet

신경망은 깊게!

신경망 깊이 의 한계 = Degradation problem



아마도 기울기 소실 문제 더많은 데이터를 설명하기 위해서

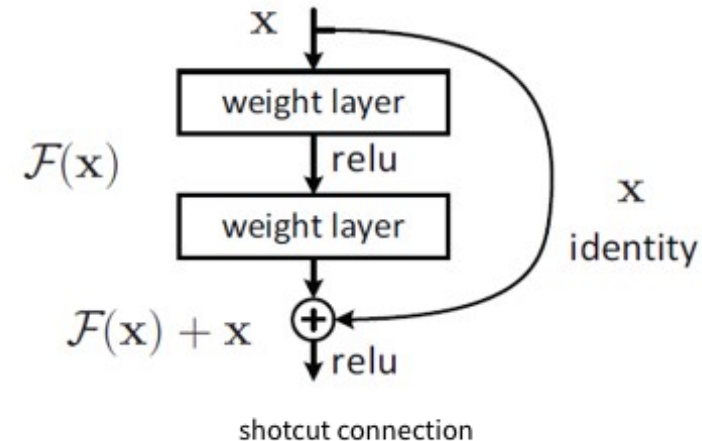
층을 깊게 쌓으면

그만큼 최 하단 노드(input데이터와 가까운)까지의 정보소실이 크다.

ResNet

학습된 노드를 하나의 잔차(Residual) 라고 설명하면 어떨까?

즉 노드를 거쳐서 나온 $F(x)$ 가
원래 의 input x 와 후에 나온 $H(x)$ 가 같다고 생각하고
 $F(x)$ 는 잔차(쉽게 말해서 에러)
목표는 이러한 에러를 최소화 하는 방향으로 접근



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

원래

ResNet

$$\min(y - h(x))$$

$$h(x) = x + F(x)$$

$$to\ be \dots h(x) = x$$

$$\min(F(x))$$

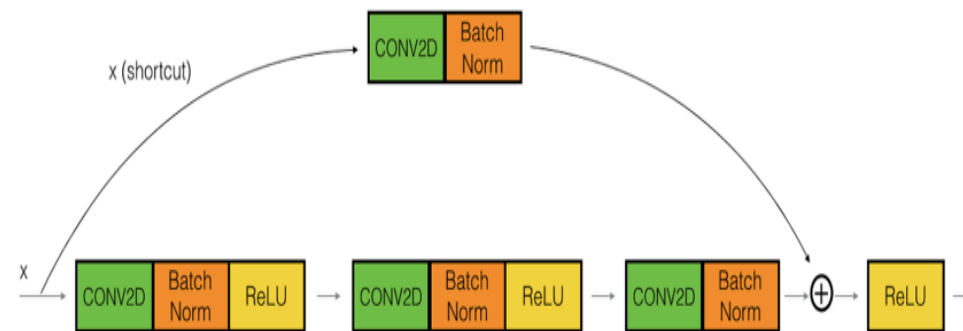
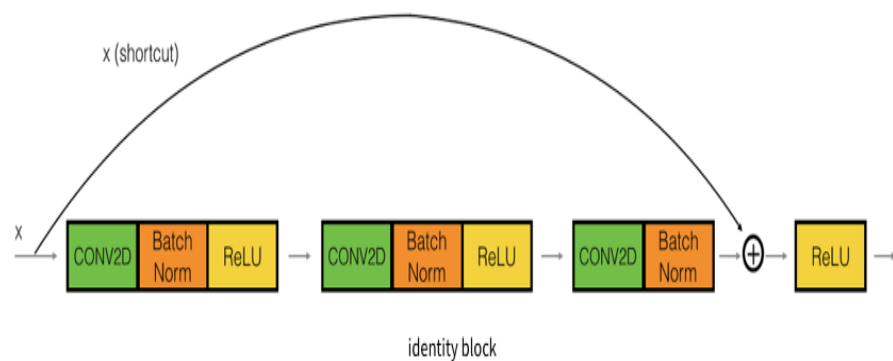
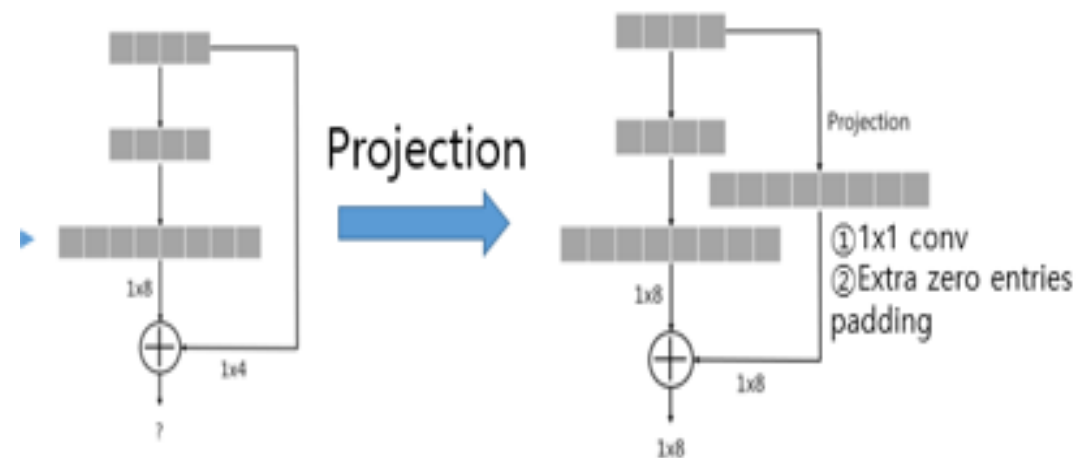
어찌 합치지?

코드는 어떻게 돌아가지?

그렇다면 역전파는?

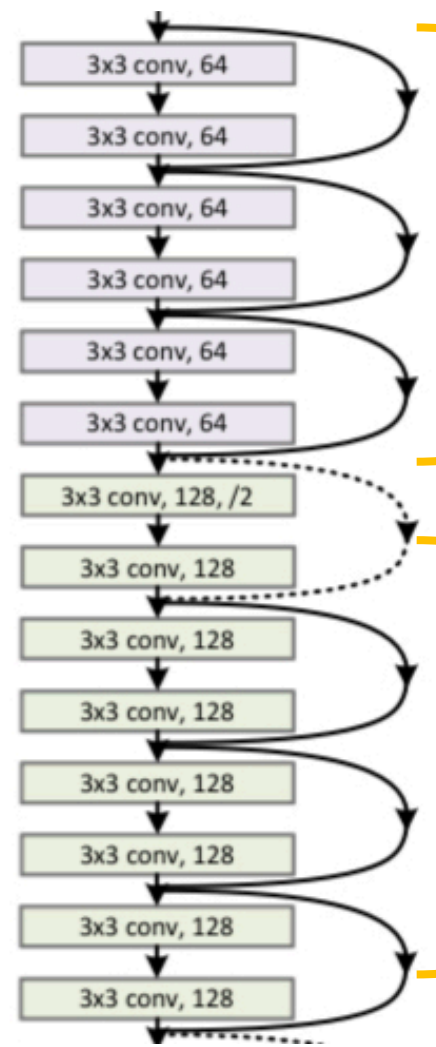
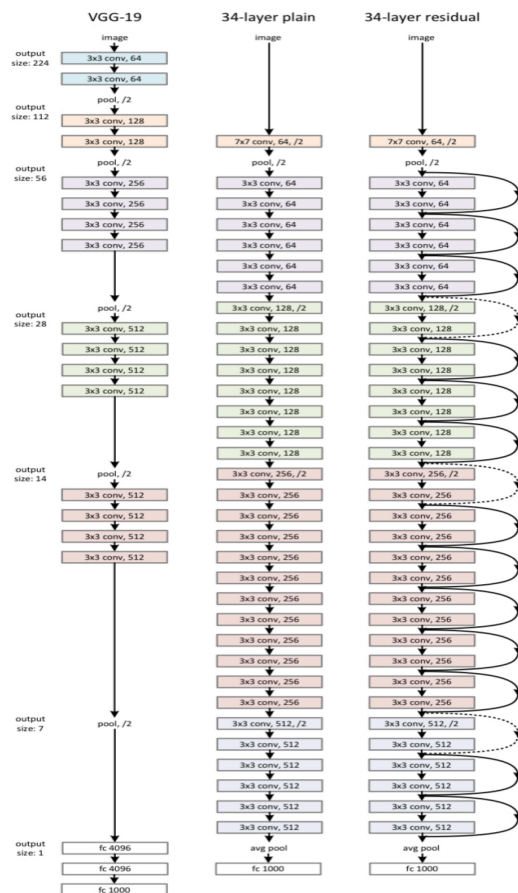
ResNet

x identity 와 F(x) 의 차원이 다르다면?



ResNet

논문에서 제안하는 구조들 및 간단 코드 리뷰



하나의 basic block (2번의 conv)

이 3개 묶음

Stride 길이 변환을 통한 크기 축소

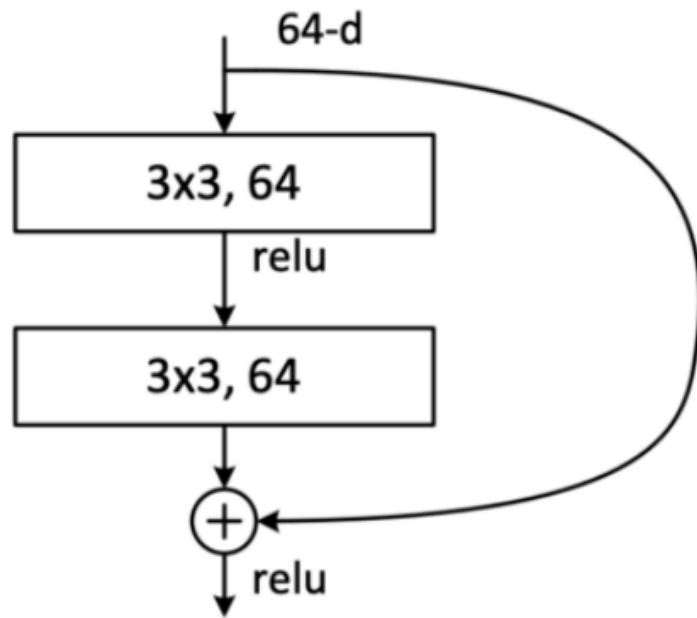
1번의 trick 과 3번의 basic block

ResNet

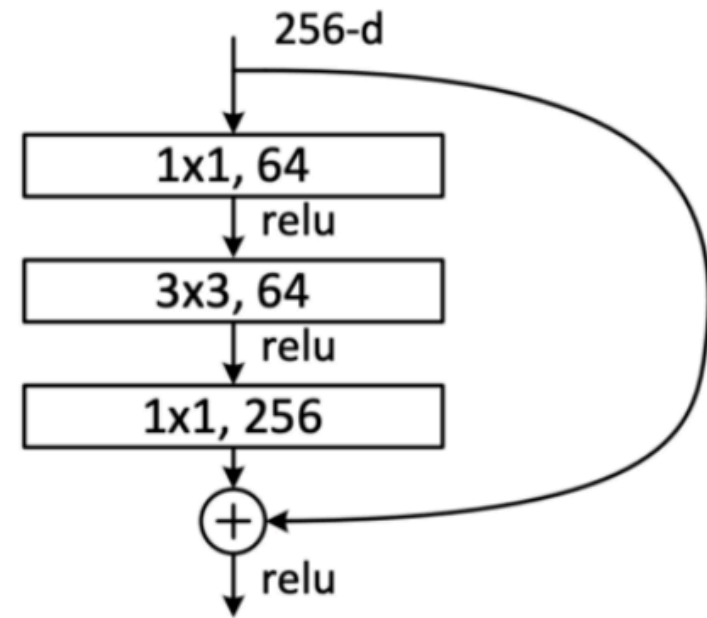
논문에서 제안하는 구조들 및 간단 코드 리뷰

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	$7 \times 7, 64, \text{stride } 2$				
conv2_x	56×56	$3 \times 3 \text{ max pool, stride } 2$				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

ResNet

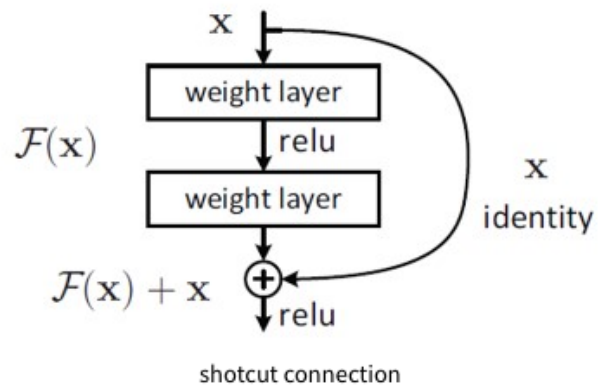


BottleNeck



ResNet

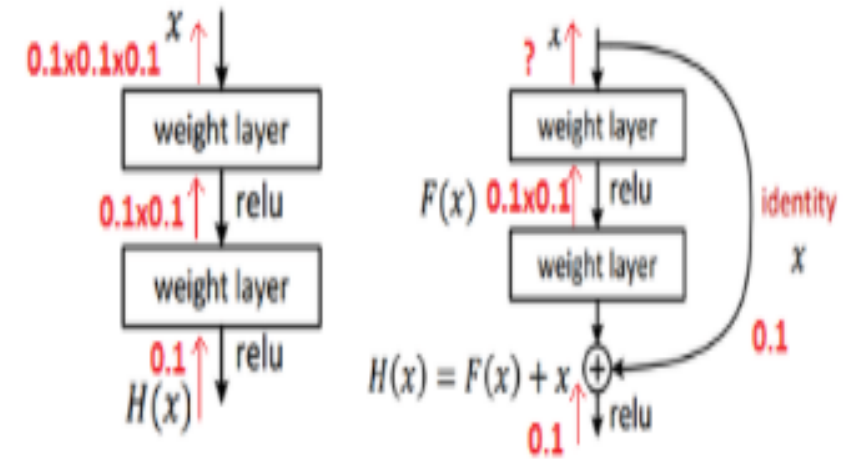
backpropagation



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

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

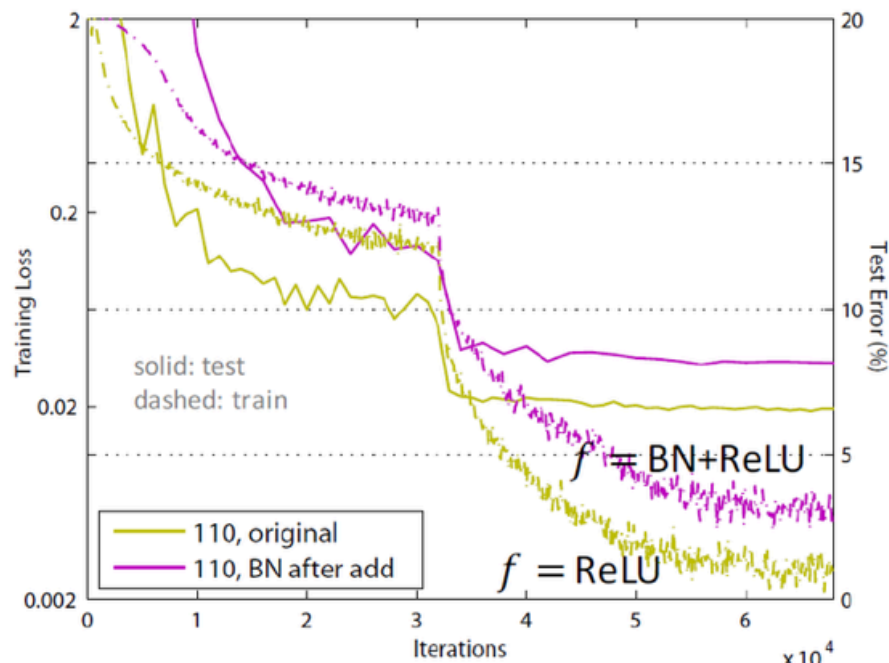
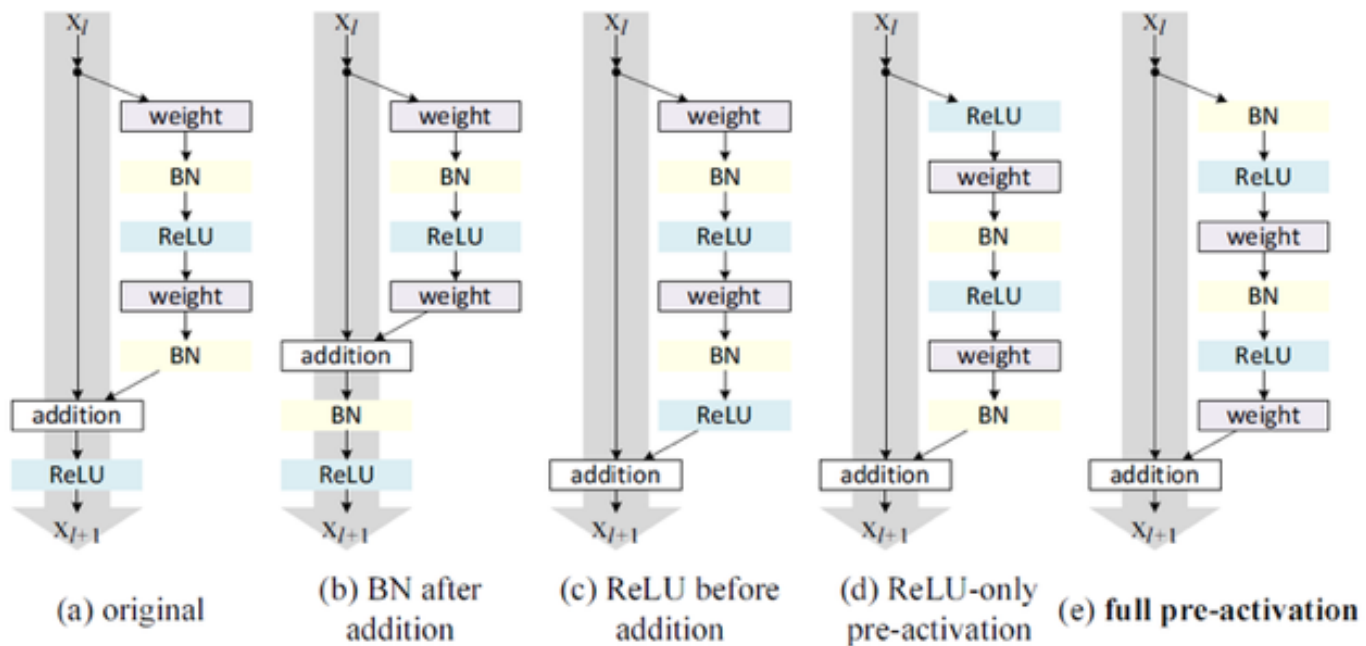
$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L} \frac{\partial \mathbf{x}_L}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L} \left(1 + \frac{\partial}{\partial \mathbf{x}_l} \sum_{i=l}^{L-1} \mathcal{F}(\mathbf{x}_i,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



저 남은1 때문에 기울기가 소실될 가능성이 낮다.
지속적으로 정보를 가져 갈 수 있다.

ResNet

개선된 방향 basicblock을 좀더 다양한 방식으로 변환해 보자.



출처..

resNet Deep Residual Learning for Image Recognition <https://arxiv.org/pdf/1512.03385.pdf>
Identity Mappings in Deep Residual Networks. <https://arxiv.org/pdf/1603.05027.pdf>
동빈나님 사랑합니다. <https://github.com/ndb796/Deep-Learning-Paper-Review-and-Practice>

Blog

<http://openresearch.ai/t/identity-mappings-in-deep-residual-networks/47>
<https://stackoverflow.com/questions/44512126/how-to-calculate-gradients-in-resnet-architecture>
<https://stats.stackexchange.com/questions/268820/gradient-backpropagation-through-resnet-skip-connections>
<https://medium.com/@codecompose/resnet-e3097d2cfe42>
<https://mole-starseeker.tistory.com/12>

추가 참고..

DenseNet 활용하여 table 데이터 예측 http://sclab.yonsei.ac.kr/publications/Papers/KC/2017_BI_JYK.pdf
DenseNet 활용하여 table 데이터 예측 <https://arxiv.org/pdf/1709.02956.pdf>