Deep Residual Learning

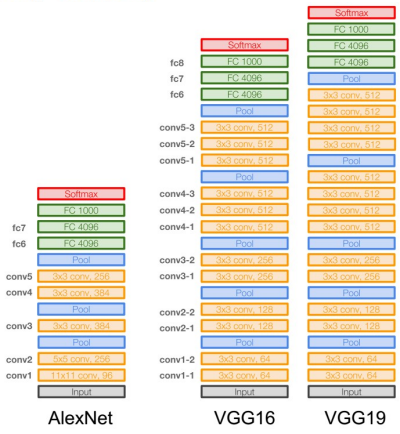
Warm-up

AlexNet

- ReLU
  - the standard of activation fn
- Local Response Normalization
  - 일정 W이 높은 값을 갖도록 처리
- Data augmentation
  - flip, crop
    256×256 → 244×244 = 32×32×2
  - color variation
- Dropout
  - output × 0.5 (not general)

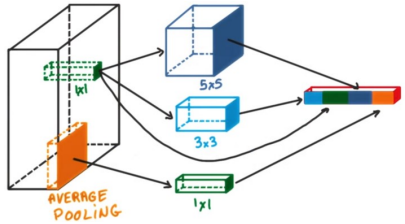
VGG

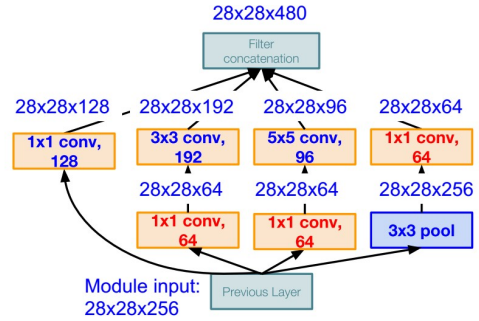
- 3 * 3 Convolution
- 간단한 방법으로 좋은 결과
  - stride 1, pad 1
  - maxpooling 혹은 avgpooling
- No Local Respones Normalization
- Use VGG16 or VGG19
  VGG19 only slightly better,
  more memory



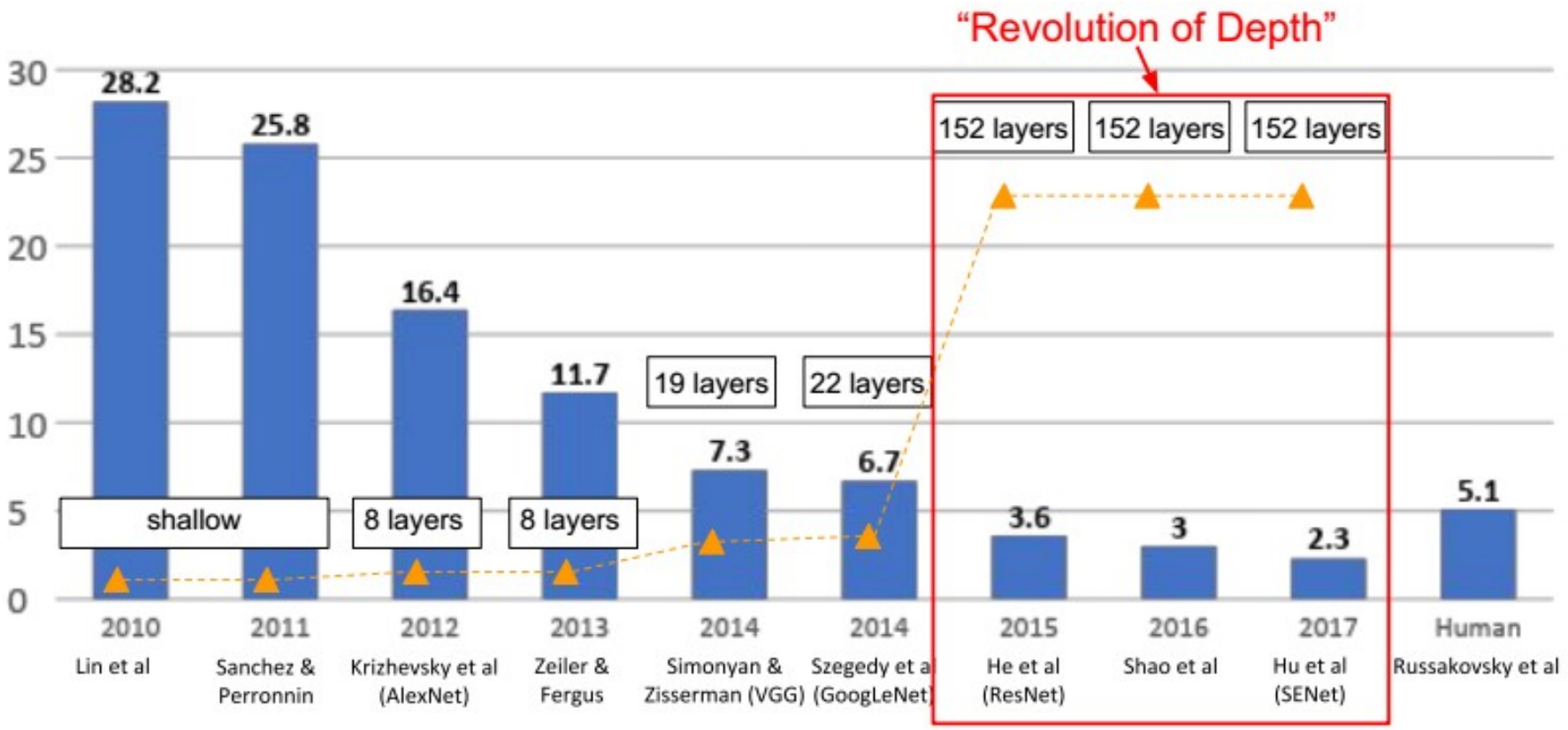
GoogLeNet

- 다양한 연결
  - Inception Module
- Dimention Reduction
  - 1 × 1 convolution
    854M → 358M operation

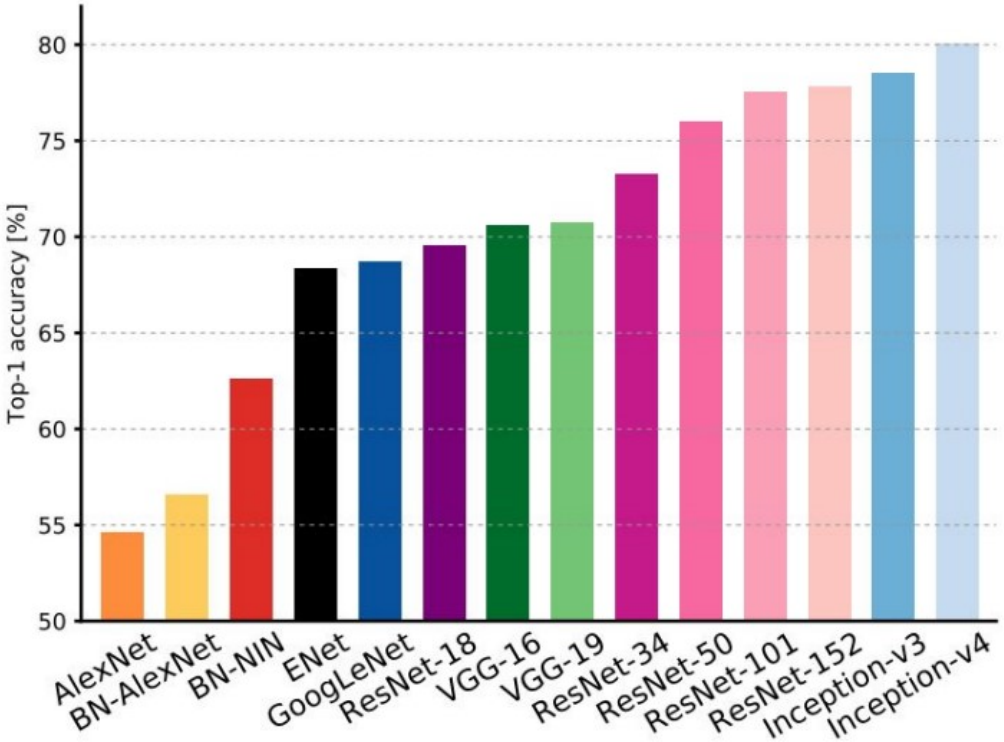




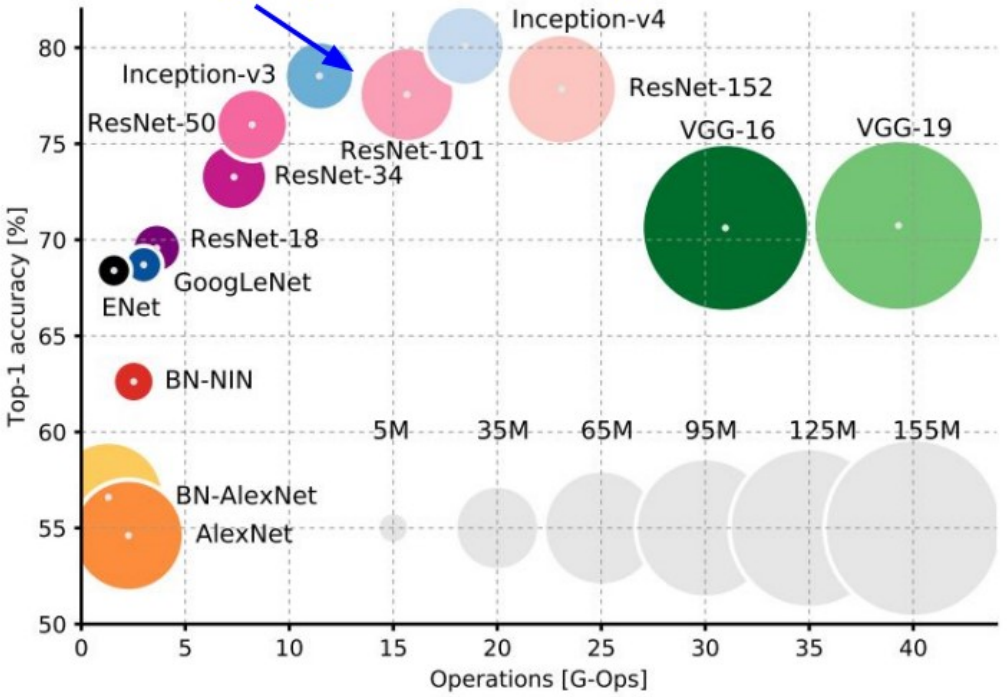
ImageNet Large Scale Visual Recongnition Challenge



Comparing complexity



ResNet:
Moderate efficiency depending on model, highest accuracy

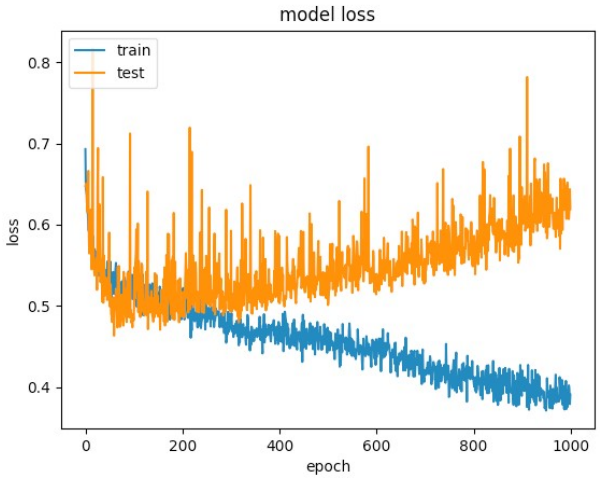


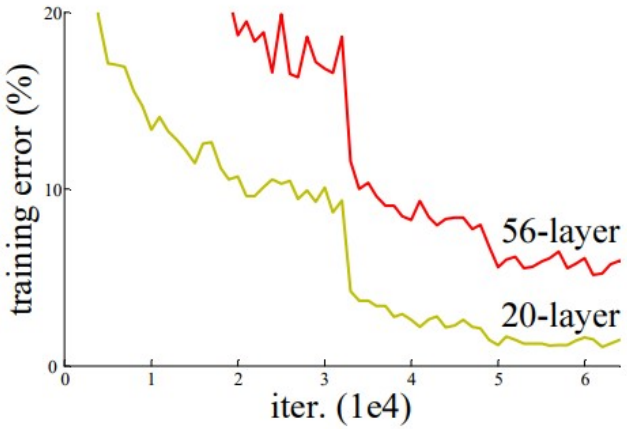
An Analysis of Deep Neural Network Models for Practical Applications, 2017.

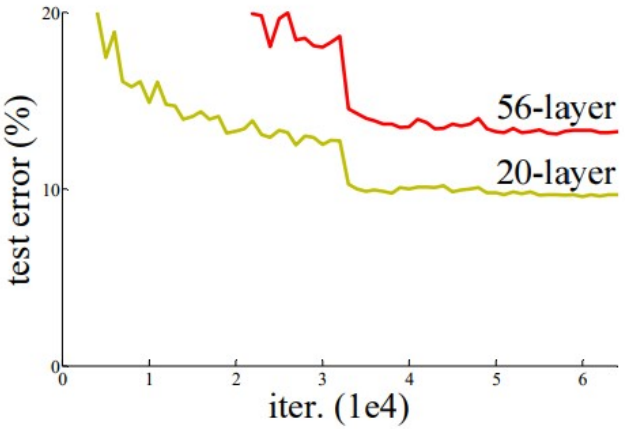
Degradation Problem

Layer가 늘어날 수록 학습자체가 어려워지는 것은 아닐까?
– 논문에서는 nonlinear function이 많아질 수록 추론이 어려워진다고 예상 (이유 없음)

Vanishing/Expliding Gradient와 Overfitting과는 다른 현상





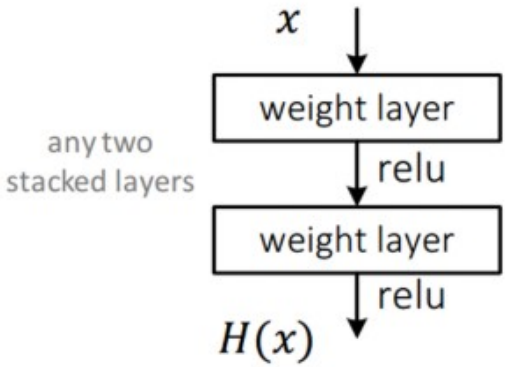


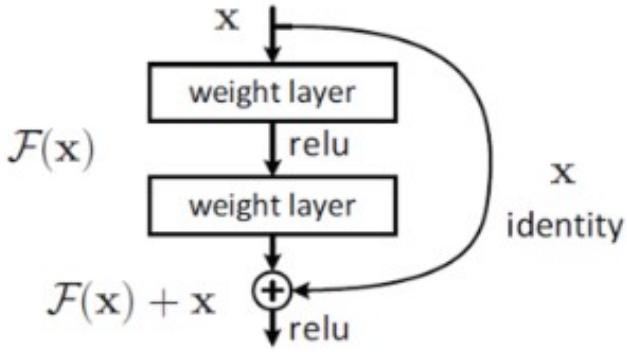
Solution : Residual mapping

깊은 망을 학습하기 위해 H(x)가 아닌 H(x) – x 를 추론하도록 관점을 전환

학습해야 할 target function

결국 학습망은 Input과 Output의 차이만을 학습하면 되므로 Residual learning이라고 칭함





Structure

34, 50, 101, 152 순으로 Layer가 높을 수록 예측 성능이 증가됨

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**

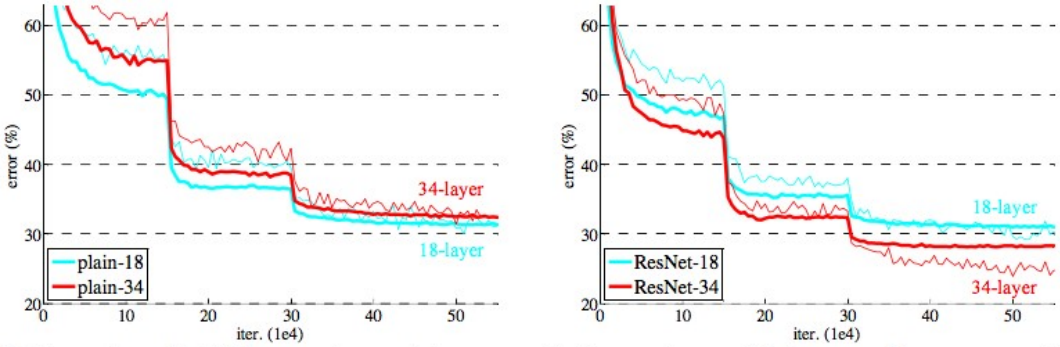
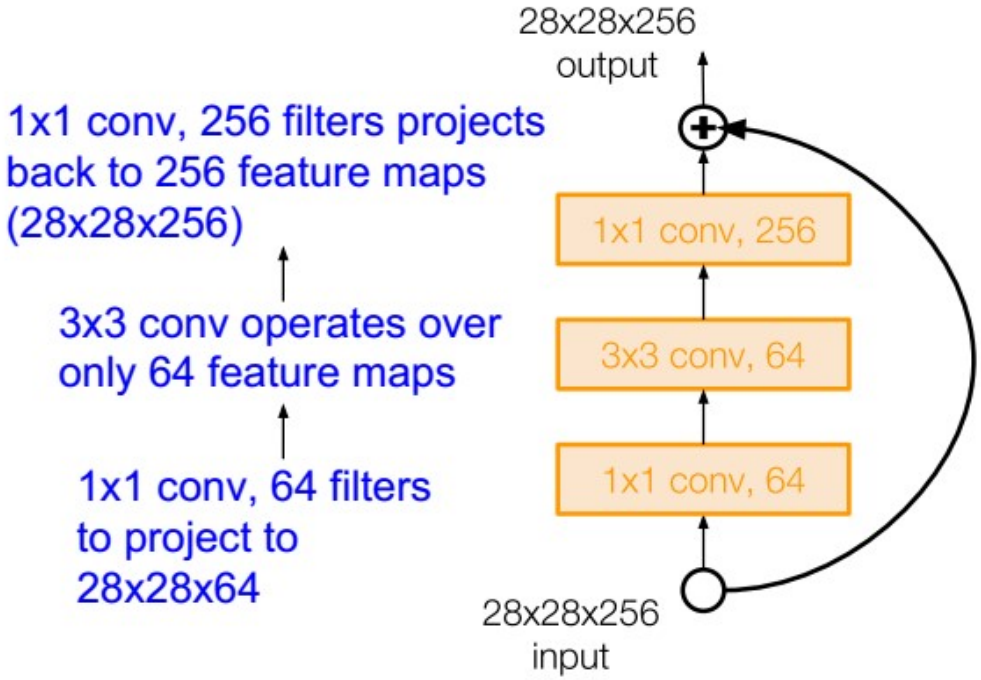


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

Structure

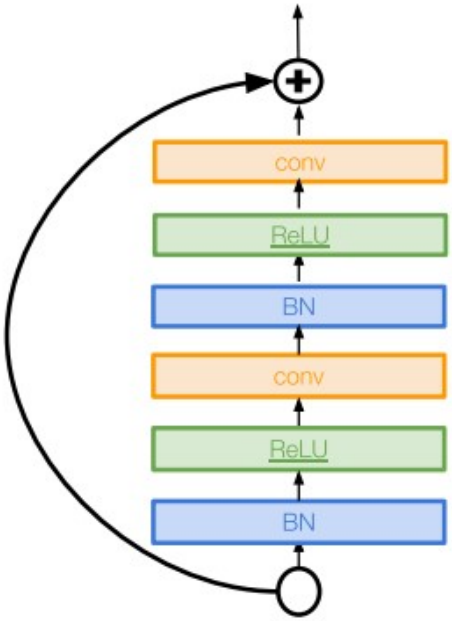
파라미터 수를 줄이기 위해 50개 레이어 이상에서는 GoogLeNet과 같이 bottleneck 기법을 사용함. 기존 제안한 34개 레이어에 1×1 convolution layer를 추가하면 50 layer로 확대되며, 이를 더 깊이 확장하여 101, 152 Layer 모델을 생성



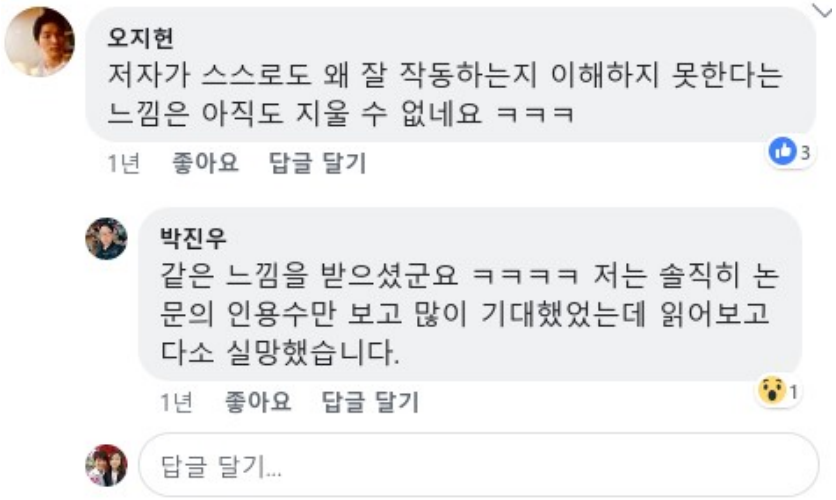
Identity Mappings in Deep Residual Networks

[He et al. 2016]

- Improved ResNet block design from creators of ResNet
- Creates a more direct path for propagating information throughout network (moves activation to residual mapping pathway)
- Gives better performance



Why better


오지헌
저자가 스스로도 왜 잘 작동하는지 이해하지 못한다는 느낌은 아직도 지울 수 없네요 ㅋㅋㅋ
1년 좋아요 답글 달기
3
박진우
같은 느낌을 받으셨군요 ㅋㅋㅋㅋ 저는 솔직히 논문의 인용수만 보고 많이 기대했었는데 읽어보고 다소 실망했습니다.
1년 좋아요 답글 달기
1
답글 달기...

Why better

Shortcut(Skip connection)

그래디언트가 잘 전달될 수 있도록 지름길을 연결

$$y_l = H(x_l) + F(x_l, W_l)$$

Original model와 같이 $H(x_l)$를 x로 설정했을 때 이를 미분하여 back propagation 특성을 살펴보면,

$$y_l = h(x_l) + F(x_l, W_l) \qquad x_{l+1} = f(y_l)$$

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

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

결국 Layer가 깊어도 더 낮은 레이어로 Directly Propagate 되도록 보장하는 Additive Term이 존재하게 됨
$H(x_l)$를 λ x로 설정한다면, 미분결과는 아래와 같으며 λ 이 1보다 작으면 0으로 수렴하거나 1보다 크다면 발산

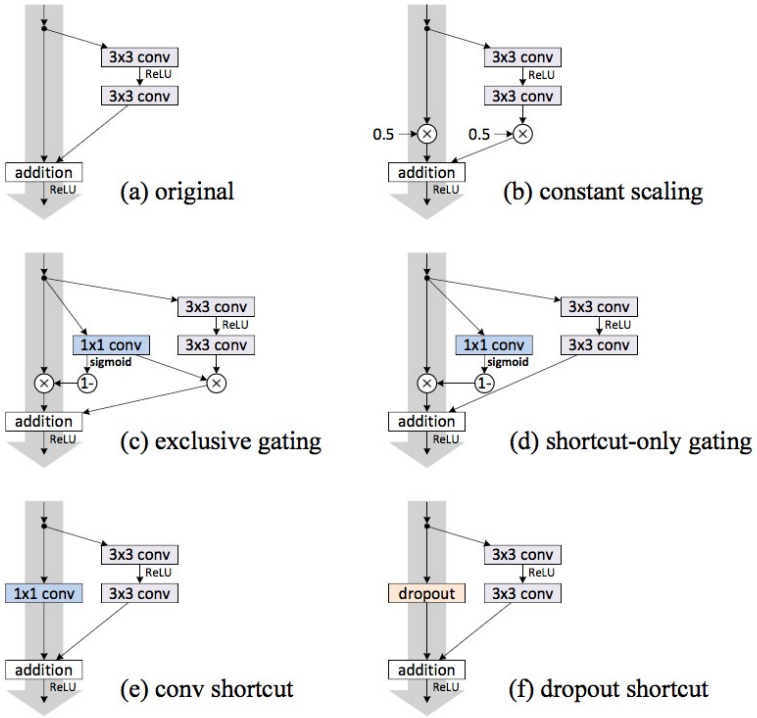
$$x_L = (\prod_{i=l}^{L-1} \lambda_i)x_l + \sum_{i=l}^{L-1} F(x_i, W_i)$$

$$\frac{\partial \epsilon}{\partial x_l} = \frac{\partial \epsilon}{\partial X_l}((\prod_{i=l}^{L-1} \lambda_i)x_l + \sum_{i=l}^{L-1} F(x_i, W_i))$$

※ Identity Mapping in Deep Residual Networks(2016)

Why better

original 모델의 성능이 우수



case	Fig.	on shortcut	on $\mathcal{F}$	error (%)	remark
original [1]	Fig. 2(a)	1	1	**6.61**	
constant scaling	Fig. 2(b)	0	1	fail	This is a plain net
		0.5	1	fail	
		0.5	0.5	12.35	frozen gating
exclusive gating	Fig. 2(c)	$1-g(\mathbf{x})$	$g(\mathbf{x})$	fail	init b_g=0 to −5
		$1-g(\mathbf{x})$	$g(\mathbf{x})$	8.70	init b_g=-6
		$1-g(\mathbf{x})$	$g(\mathbf{x})$	9.81	init b_g=-7
shortcut-only gating	Fig. 2(d)	$1-g(\mathbf{x})$	1	12.86	init b_g=0
		$1-g(\mathbf{x})$	1	6.91	init b_g=-6
1×1 conv shortcut	Fig. 2(e)	1×1 conv	1	12.22	
dropout shortcut	Fig. 2(f)	dropout 0.5	1	fail	

※ Identity Mapping in Deep Residual Networks(2016)

Why better

full pre-activation 모델이 우수함, ReLU에 의해 zero-out되는 Weight를 줄인 효과로 설명1

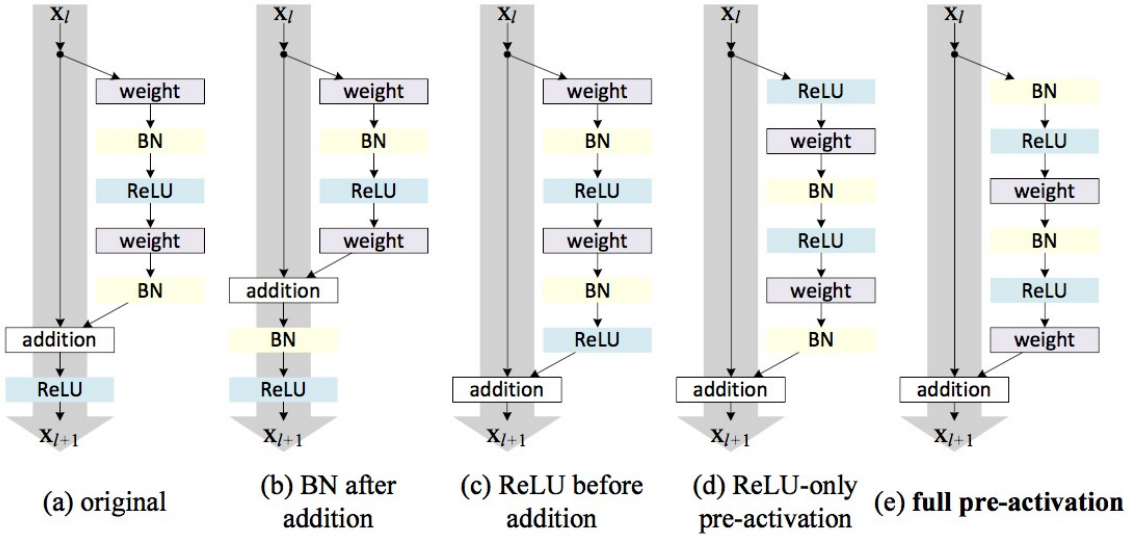


Figure 4. Various usages of activation in Table 2. All these units consist of the same components — only the orders are different.

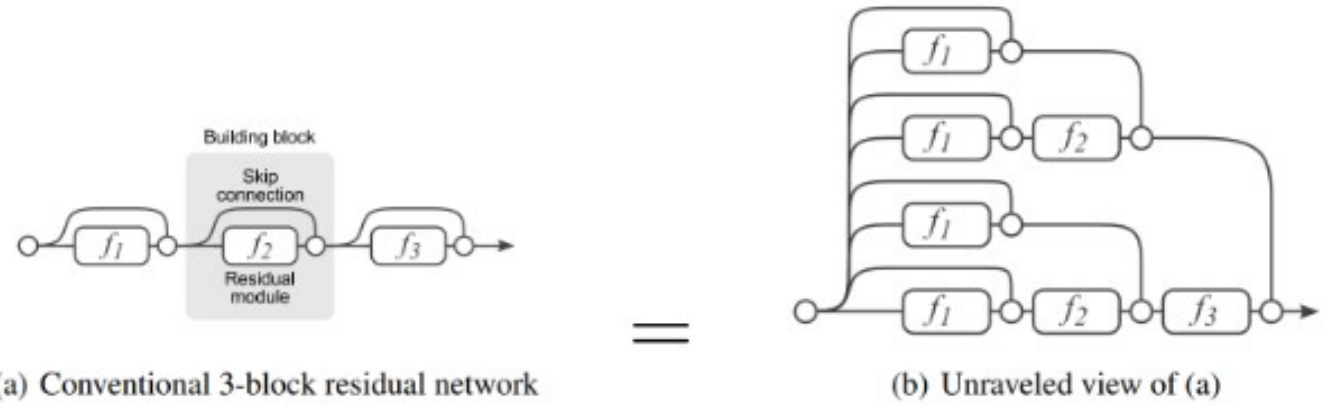
case	Fig.	ResNet-110	ResNet-164
original Residual Unit [1]	Fig. 4(a)	6.61	5.93
BN after addition	Fig. 4(b)	8.17	6.50
ReLU before addition	Fig. 4(c)	7.84	6.14
ReLU-only pre-activation	Fig. 4(d)	6.71	5.91
full pre-activation	Fig. 4(e)	**6.37**	**5.46**

※ Identity Mapping in Deep Residual Networks(2016)

Why better

Ensemble effect

- shortcut pass가 skip connection 덕분에 통로가 크게 늘어남(n개의 Skip connection이 있다면 2^n의 다른 통로 존재)



(a) Conventional 3-block residual network

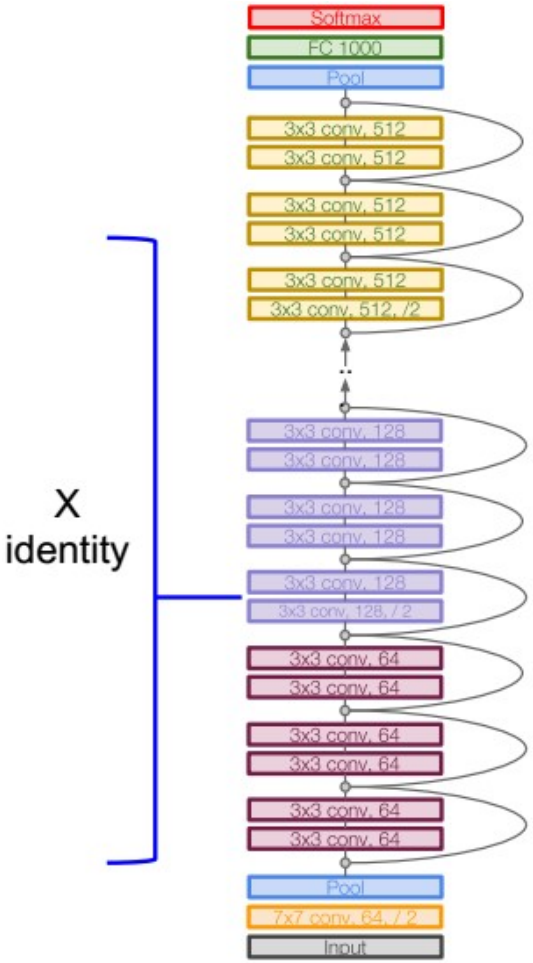
(b) Unraveled view of (a)

주요 특징

- **Stack residual blocks**
- **Every reidual block has two 3*3 conv layer**
- **Periodical, double # of filters and downsample spatially using stride 2**
- **No FC layers at the end (only FC 1000 to output classes)**

Training ResNet in practice:

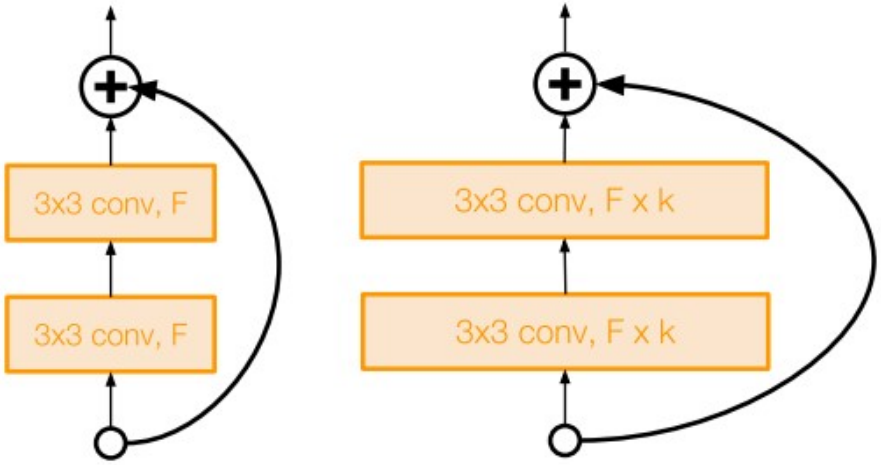
- Batch Normalization after every CONV layer
- Xavier 2/ initialization from He et al.
- SGD + Momentum (0.9)
- Learning rate: 0.1, divided by 10 when validation error plateaus
- Mini-batch size 256
- Weight decay of 1e-5
- No dropout used



Wide Residual Networks

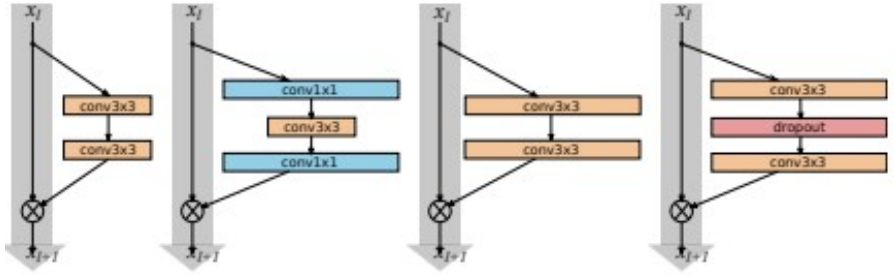
[Zagoruyko et al. 2016]

- Argues that residuals are the important factor, not depth
- User wider residual blocks (F x k filters instead of F filters in each layer)
- 50-layer wide ResNet outperforms 152-layer original ResNet
- Increasing width instead of depth more computationally efficient (parallelizable)



Basic residual block Wide residual block

WRN Structure



(a) basic (b) bottleneck (c) basic-wide (d) wide-dropout

1. $B(3,3)$ - original «basic» block
2. $B(3,1,3)$ - with one extra 1×1 layer
3. $B(1,3,1)$ - with the same dimensionality of all convolutions, «straightened» bottleneck
4. $B(1,3)$ - the network has alternating 1×1 - 3×3 convolutions everywhere
5. $B(3,1)$ - similar idea to the previous block
6. $B(3,1,1)$ - Network-in-Network style block

group name	output size	block type = $B(3,3)$
conv1	32×32	[3×3, 16]
conv2	32×32	$\begin{bmatrix} 3\times3, 16\times k \\ 3\times3, 16\times k \end{bmatrix} \times N$
conv3	16×16	$\begin{bmatrix} 3\times3, 32\times k \\ 3\times3, 32\times k \end{bmatrix} \times N$
conv4	8×8	$\begin{bmatrix} 3\times3, 64\times k \\ 3\times3, 64\times k \end{bmatrix} \times N$
avg-pool	1×1	$[8 \times 8]$

Table 1: Structure of wide residual networks. Network width is determined by factor k. Original architecture [13] is equivalent to $k = 1$. Groups of convolutions are shown in brackets where N is a number of blocks in group, downsampling performed by the first layers in groups `conv3` and `conv4`. Final classification layer is omitted for clearance. In the particular example shown, the network uses a ResNet block of type $B(3,3)$.

WRN Structure

width channel + dropout performance

depth	k	# params	CIFAR-10	CIFAR-100
40	1	0.6M	6.85	30.89
40	2	2.2M	5.33	26.04
40	4	8.9M	4.97	22.89
40	8	35.7M	4.66	-
28	10	36.5M	**4.17**	20.50
28	12	52.5M	4.33	**20.43**
22	8	17.2M	4.38	21.22
22	10	26.8M	4.44	20.75
16	8	11.0M	4.81	22.07
16	10	17.1M	4.56	21.59

depth	k	dropout	CIFAR-10	CIFAR-100	SVHN
16	4		5.02	24.03	1.85
16	4	✓	5.24	23.91	1.64
28	10		4.00	19.25	-
28	10	✓	**3.89**	**18.85**	-
52	1		6.43	29.89	2.08
52	1	✓	6.28	29.78	1.70

	depth-k	# params	CIFAR-10	CIFAR-100
NIN [20]			8.81	35.67
DSN [19]			8.22	34.57
FitNet [24]			8.39	35.04
Highway [25]			7.72	32.39
ELU [6]			6.55	24.28
original-ResNet[11]	110	1.7M	6.43	25.16
	1202	10.2M	7.93	27.82
stoc-depth[14]	110	1.7M	5.23	24.58
	1202	10.2M	4.91	-
pre-act-ResNet[13]	110	1.7M	6.37	-
	164	1.7M	5.46	24.33
	1001	10.2M	4.92(4.64)	22.71
WRN (ours)	40-4	8.9M	4.53	21.18
	16-8	11.0M	4.27	20.43
	28-10	36.5M	**4.00**	**19.25**

Model	top-1 err, %	top-5 err, %	#params	time/batch 16
ResNet-50	24.01	7.02	25.6M	49
ResNet-101	22.44	6.21	44.5M	82
ResNet-152	22.16	6.16	60.2M	115
WRN-50-2-bottleneck	21.9	6.03	68.9M	93
pre-ResNet-200	21.66	5.79	64.7M	154

Dataset	model	dropout	test perf.
CIFAR-10	WRN-40-10	✓	3.8%
CIFAR-100	WRN-40-10	✓	18.3%
SVHN	WRN-16-8	✓	1.54%
ImageNet (single crop)	WRN-50-2-bottleneck		21.9% top-1, 5.79% top-5
COCO test-std	WRN-34-2		35.2 mAP