Deep Residual Learning for Image Recognition

IEEE
2016
|
Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

KU 건국대학교 KONKUK UNIV.

202121206 하가형

Abstract

- Deep neural network는 학습 진행이 어려움
- 해당 논문에서는 residual learning framework 제안해 학습 과정을 쉽게 만듦
  이전에 나왔던 논문과는 달리 훨씬 깊은 네트워크를 사용한 것이 특징
- 결과
  - ImageNet에 대해 152 layer의 아주 깊은 residual net 평가 진행
    이전 연구에서 나왔던 VGG 네트워크에 비해 더 깊지만 복잡도 더 낮음 -> 성능이 훨씬 좋아져 2015 ImageNet 분류 대회 1등
  - CIFAR-10에 대해서도 실험 진행 -> 성능 많이 개선됨

➡ 특징을 표현하는 깊이는 중요한 역할을 수행함
resNet은 기본적으로 훨씬 깊은 네트워크를 학습 가능하게 함

Introduction

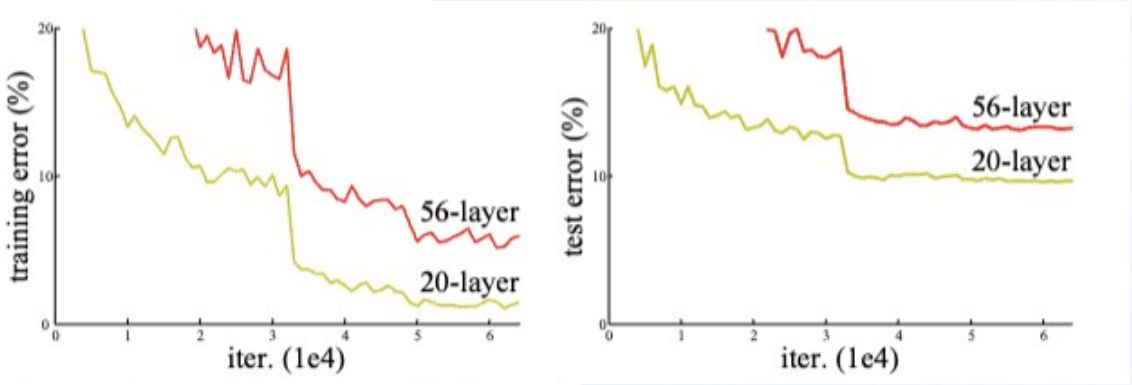
레이어를 깊게 쌓으면 좋은 학습이 가능한 것 아닌가?

레이어가 깊어짐에 따라 degradation problem이 발생할 수 있다고 주장

= 레이어가 깊으면 accuracy 무조건 높아지는 것이 아님

어느 정도 이상 높아지면 오히려 감소

이러한 문제는 단순히 overfitting 때문에 발생하는 건 아님



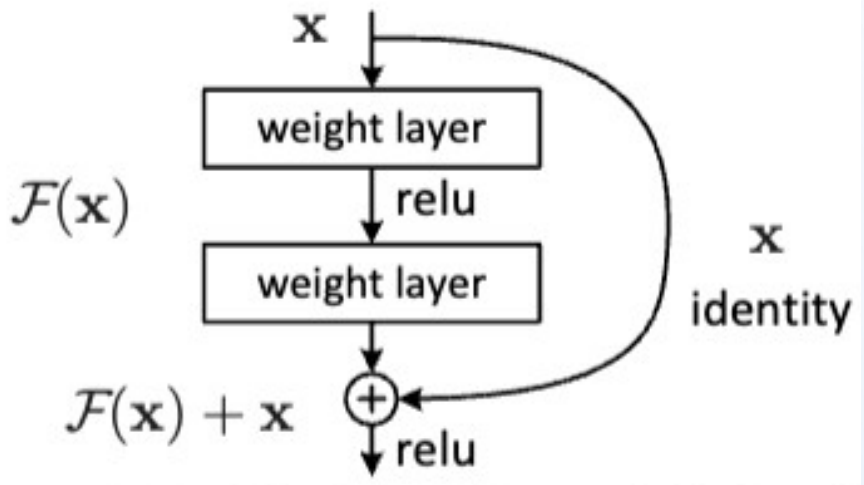
Plain network에서 레이어 수만 늘리는 것은

Train, test error 둘 다 증가시킴

Introduction

문제를 해결하기 위해 Residual function(잔차 함수)라는 개념 도입

Residual Learning



$$\mathcal{F} = W_2\sigma(W_1\mathbf{x})$$

H를 학습하기보다는 별도로 학습하기 쉬운 residual mapping를 정의해서 대신 학습

= 즉, 진짜 의도하는 H(x)가 아니라 F를 대신 학습

Introduction

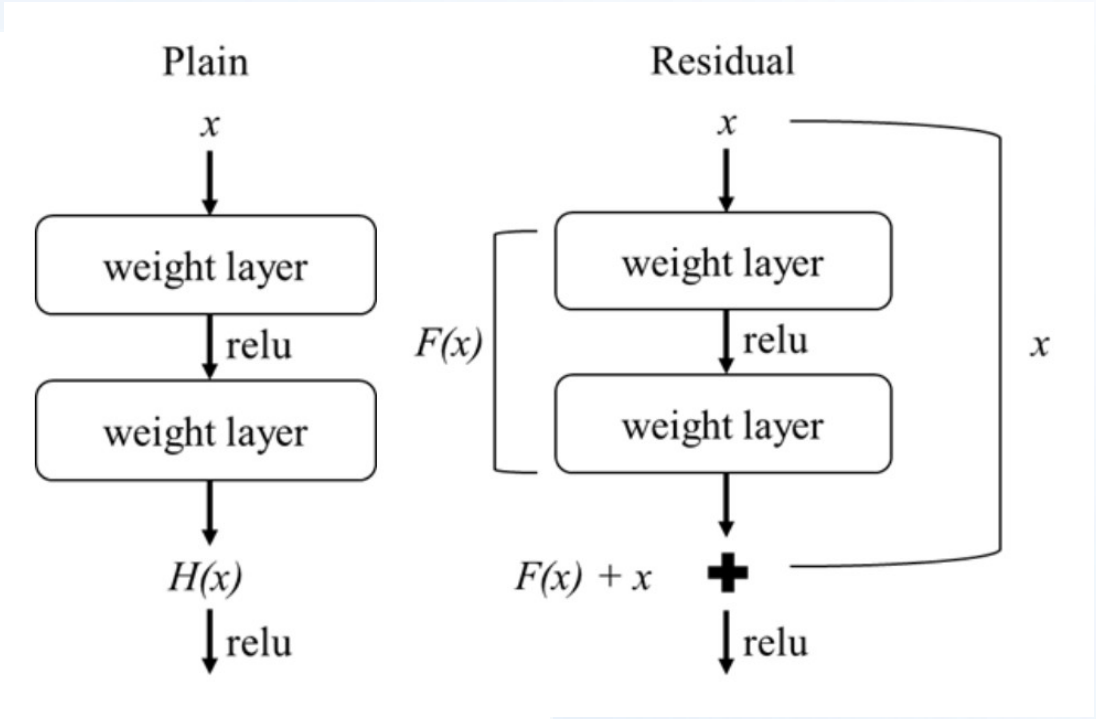
Residual Learning

장점

- 출력값에 x를 더하는 것이기 때문에 별도로 추가적인 파라미터 필요하지 않음
- 복잡도 더 증가하지 않음
- 구현 간단
- Resnet을 사용했을 때 학습 난이도가 쉬움
    - 깊이가 깊어질수록 높은 정확도 보임

Deep Residual Learning

3.1 Residual Learning



Deep Residual Learning

3.2 Identity Mapping by shortcuts

Residual Block

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

Biases 값은 고려하지 않음

Shortcut connection 이용할 땐 추가적인 파라미터 사용하지 않음

매개변수 수, 깊이, 폭, 계산 비용 등을 공평하게 비교했을 때도 더 우수한 결과를 보임

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

X를 차원을 매핑 시켜줄 때만 사용

Input과 output이 서로 일치하지 않는다고 하면 W_s를 곱해줌으로써 디멘션 값을 매치시키는 것

F는 여러 개의 레이어가 될 수 있지만, 한 개의 레이어인 경우 장점을 얻기 어려움

Deep Residual Learning

3.3 Network Architectures

1) Plain network

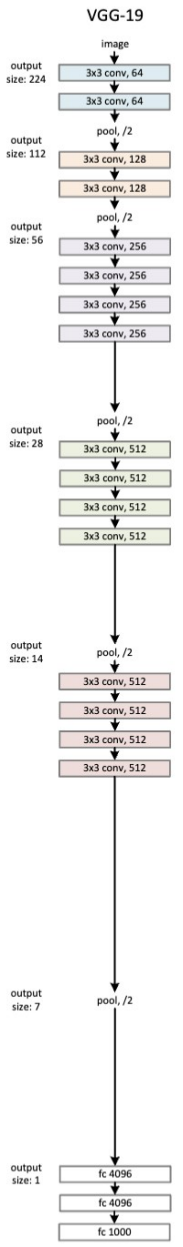
VGG 네트워크에서 제안된 기법에서 영감 얻음

3 * 3 작은 필터 이용

Output feature map 사이즈가 같도록 만들기 위해 같은 수의 필터 사용

절반으로 줄어들면 필터 수를 2배로 늘림

이러한 방법으로 레이어 당 시간 복잡도를 보존할 수 있는 형태로 구성



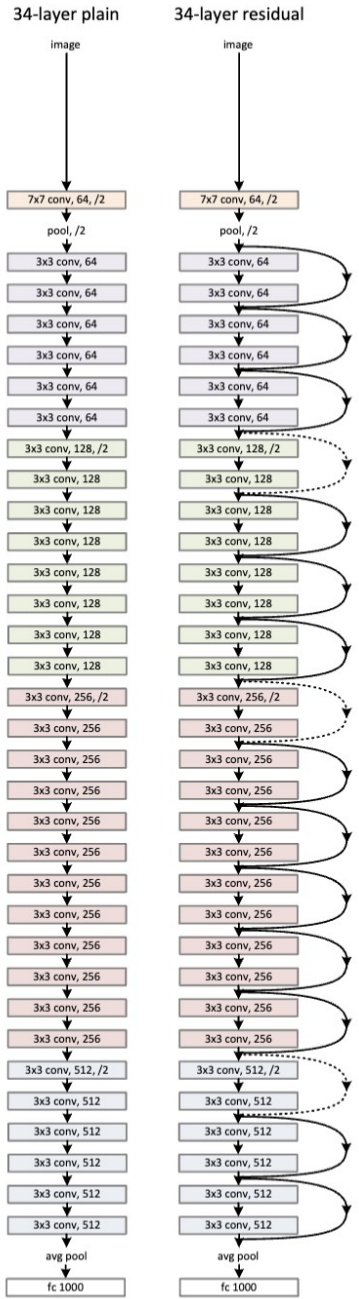
→ 논문에서의 모델은 VGG 네트워크와 비교했을 때 더 적은 파라미터를 사용하며, 복잡도는 낮음

Deep Residual Learning

3.3 Network Architectures

2) Residual network

- Residual block만 사용하는 형태로 네트워크를 바꾼 모델
- VGG와 비슷하게 3 * 3 작은 필터 이용
  - Convolution 필터를 2번씩 묶어 매번 residual function 형태로 학습 진행하도록
- 점선으로 표시된 부분 -> input과 output의 dimension이 일치하지 않아 이를 맞추는 기술이 가미된 shortcut connection
- Convolution layer를 2개씩 묶는 것 3번, 4, 6, 3번 반복

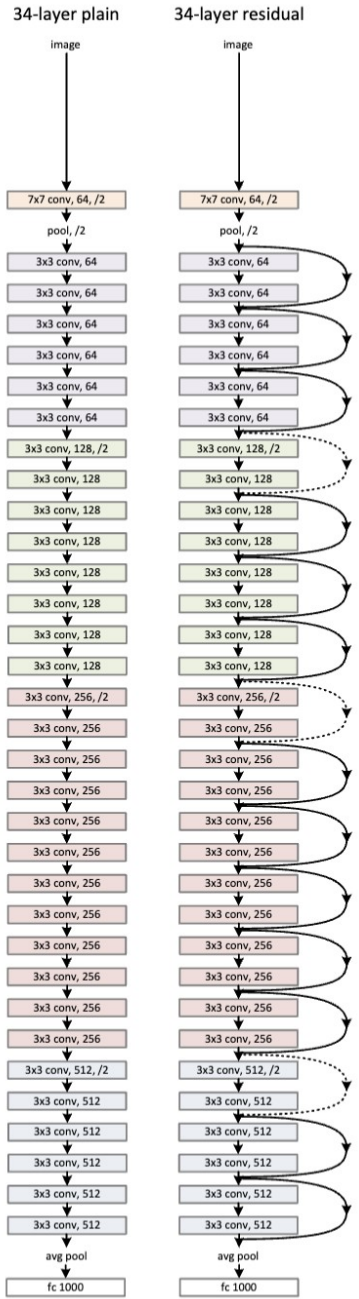


Deep Residual Learning

3.3 Network Architectures

2) Residual network

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	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

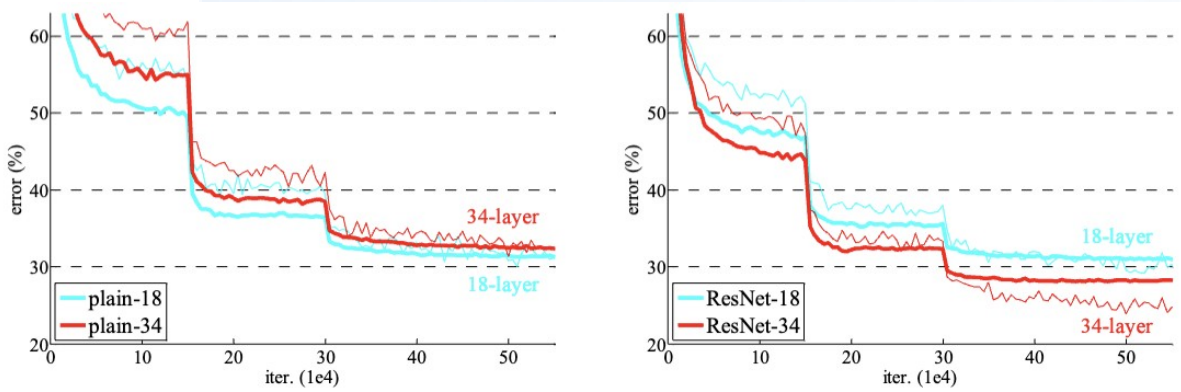


Experiments

ImageNet Classification

1) Plain network

ImageNet 2012년도 데이터 세트 이용해 평가 진행



	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

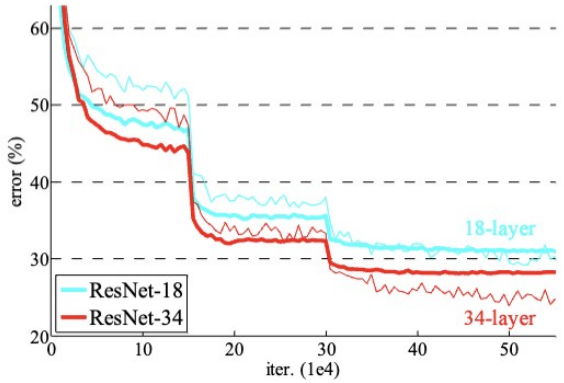
Plain 네트워크는 레이어 깊어질수록 정확도 감소

ResNet은 레이어 깊어질수록 정확도 증가

Experiments

ImageNet Classification

2) Residual network



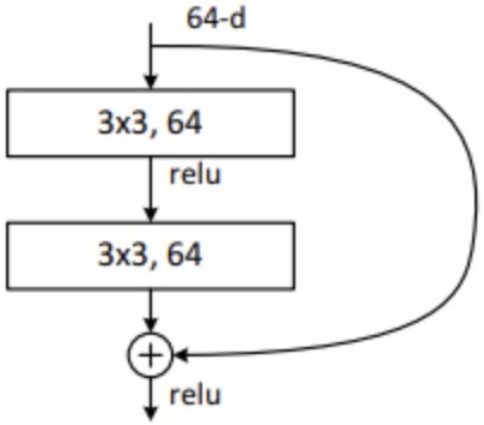
	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

- 더 깊은 레이어가 얕은 레이어에 비해 잘 동작함
- Training error 감소
- 일반화 성능 높아짐
- 수렴 속도 더 빠름

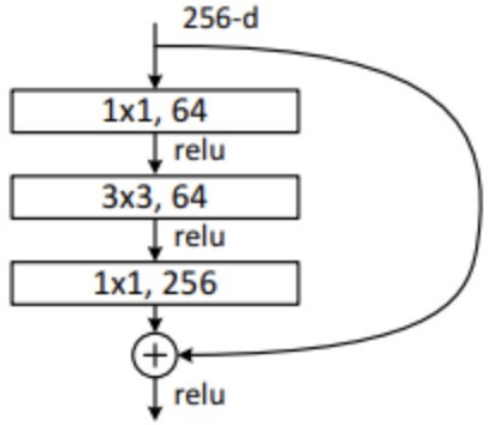
34-layer가 18-layer보다 2.8% 가량 우수한 성능 보임

Experiments

Deeper Bottleneck Architectures



18, 34 레이어에서 사용



50, 101, 152 레이어에서 사용

- 1*1, 3*3, 1*1 순서로 구성
  - 1*1 필터
    256의 dimension을 64개의 dimension으로
    차원을 축소
  - 3*3 필터
    공간적인 특징을 추출
  - 1*! 필터
    256개의 dimension으로 확장

Experiments

Comparisons with State-of-the-art Methods

Table 4

method	top-1 err.	top-5 err.
VGG [40] (ILSVRC'14)	-	8.43†
GoogLeNet [43] (ILSVRC'14)	-	7.89
VGG [40] (v5)	24.4	7.1
PReLU-net [12]	21.59	5.71
BN-inception [16]	21.99	5.81
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	**19.38**	**4.49**

이전 최고 단일 모델 결과와 비교

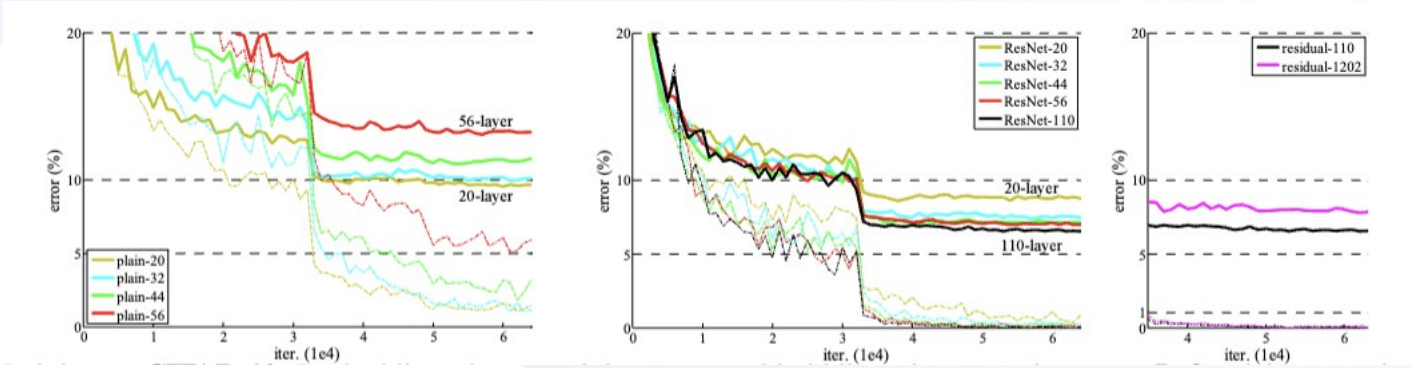
Table 5

method	top-5 err. (**test**)
VGG [40] (ILSVRC'14)	7.32
GoogLeNet [43] (ILSVRC'14)	6.66
VGG [40] (v5)	6.8
PReLU-net [12]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

단일 모델의 결과

Experiments

CIFAR-10 and Analysis

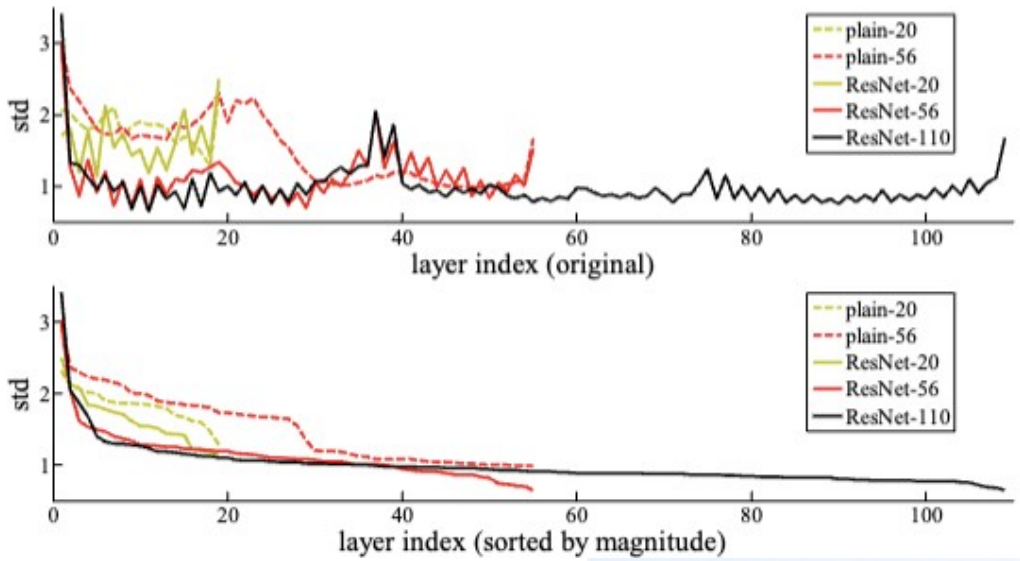


output map size	32×32	16×16	8×8
# layers	1+2n	2n	2n
# filters	16	32	64

method			error (%)
Maxout [9]			9.38
NIN [25]			8.81
DSN [24]			8.22
	# layers	# params	
FitNet [34]	19	2.5M	8.39
Highway [41, 42]	19	2.3M	7.54 (7.72±0.16)
Highway [41, 42]	32	1.25M	8.80
ResNet	20	0.27M	8.75
ResNet	32	0.46M	7.51
ResNet	44	0.66M	7.17
ResNet	56	0.85M	6.97
ResNet	110	1.7M	**6.43** (6.61±0.16)
ResNet	1202	19.4M	7.93

Experiments

Analysis of Layer Responses



training data	07+12	07++12
test data	VOC 07 test	VOC 12 test
VGG-16	73.2	70.4
ResNet-101	**76.4**	**73.8**

각 레이어의 response에 대해 평가