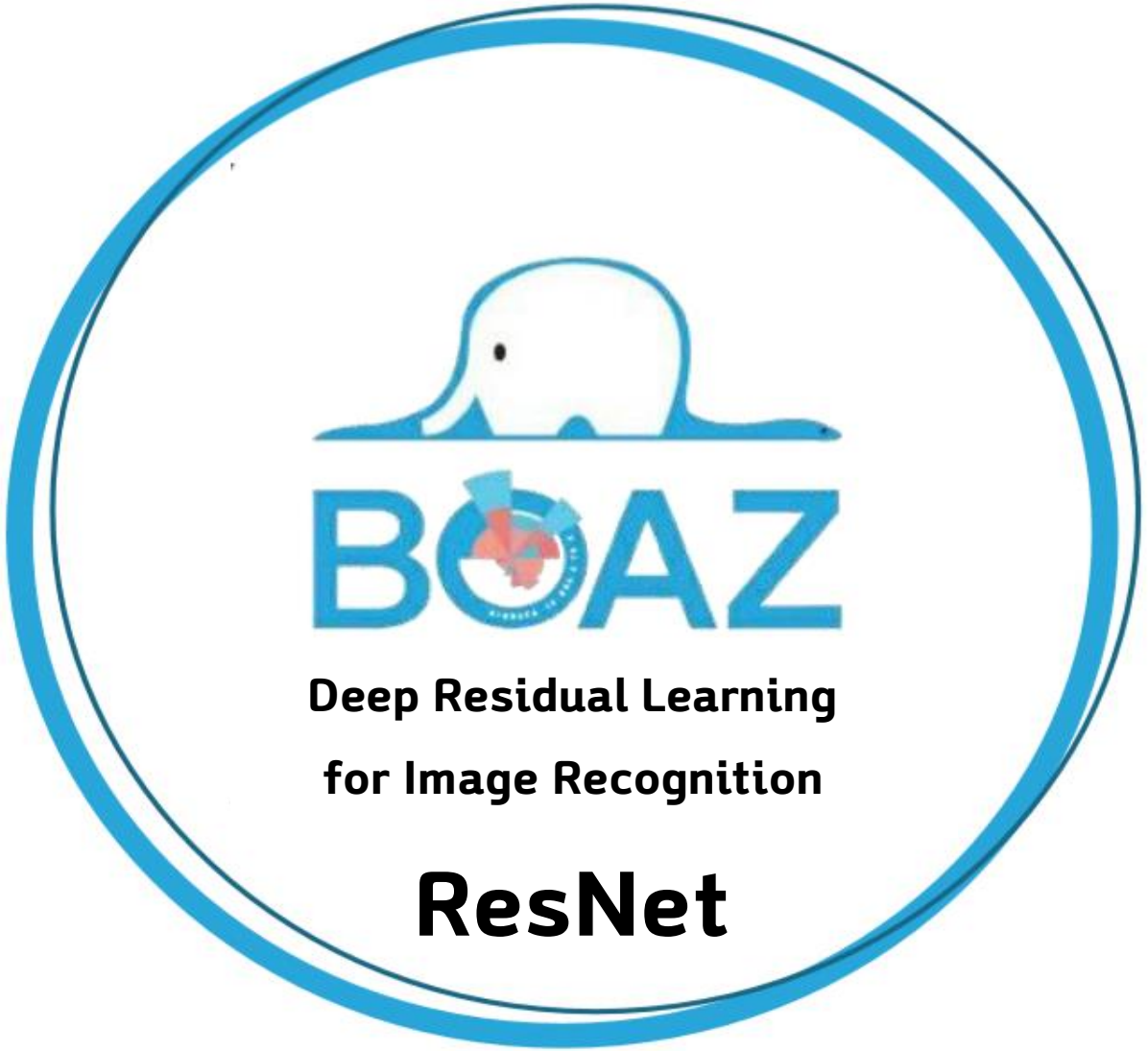


발표자: 김영민, 이다인

Image generator : 김영민, 김지수, 이다인

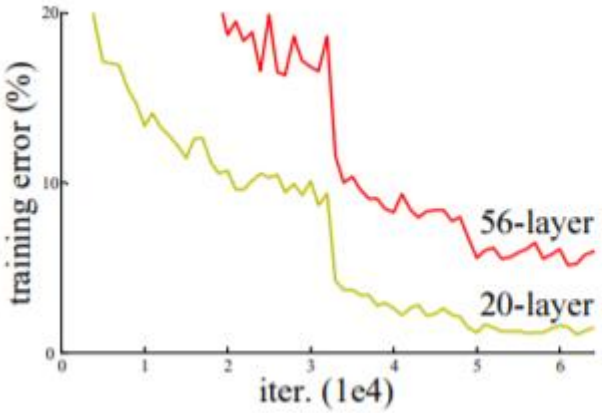
INDEX

1. Introduction

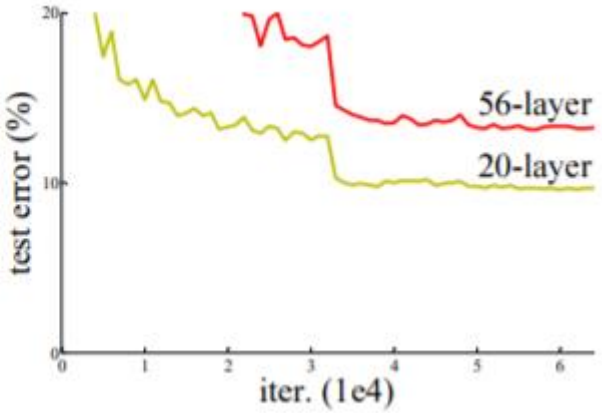
Question

많은 층으로 구성되어 있는 네트워크가 항상 좋은 성능을 낼까?

NO!!!



Train error rate



Test error rate

Train & Test 셋
둘 다 층이 깊어질수록

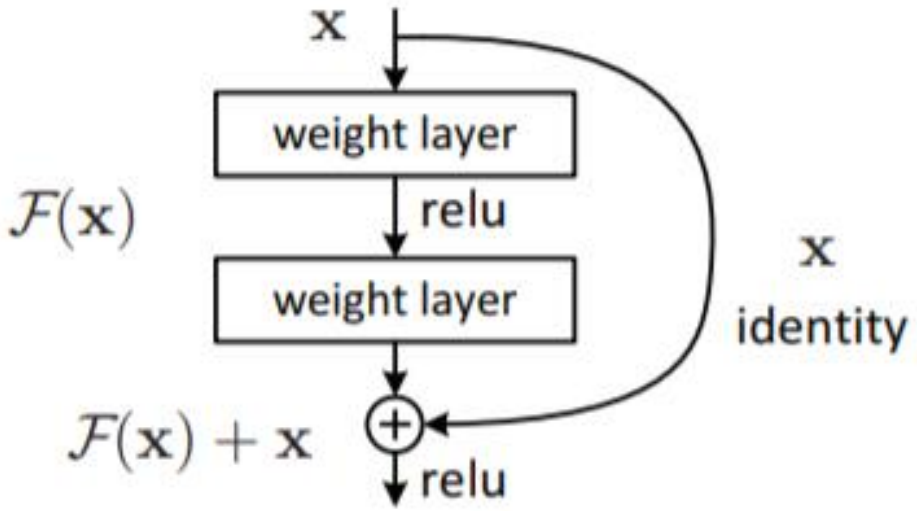
에러율 ⇑

1. Introduction

Overfitting Train 성능 ↑ Test 성능 ↓

Degradation Train 성능 ↓ Test 성능 ↓

What is Solution? → Residual Learning



Residual Block

	Optimization
Original Mapping	Hard
Residual Mapping	Easy

2. Deep Residual Learning

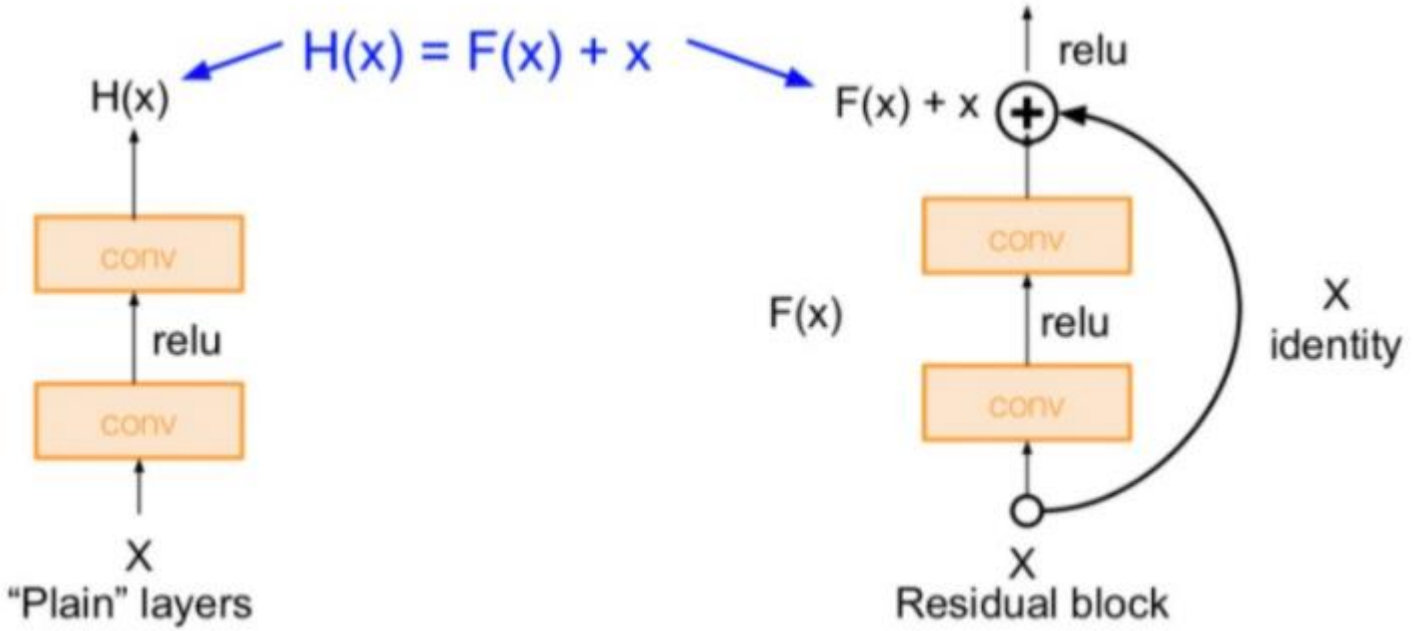
Plain Layers $H(x) = F(x) + x$

Residual Layers $F(x) = H(x) - x$

$H(x)$: *Original mapping*

$F(x)$: *Residual*

x: *Input*



여러 비선형 Layer들이 복잡한 함수하고
Identity mapping이 최적이라면

H(x)를 mapping 시키는 것 보다
잔차인 F(x)=0으로 만드는 것이 더 쉽다

2. Deep Residual Learning

Shortcut connection 한 개 이상의 layer를 skip 하는 것

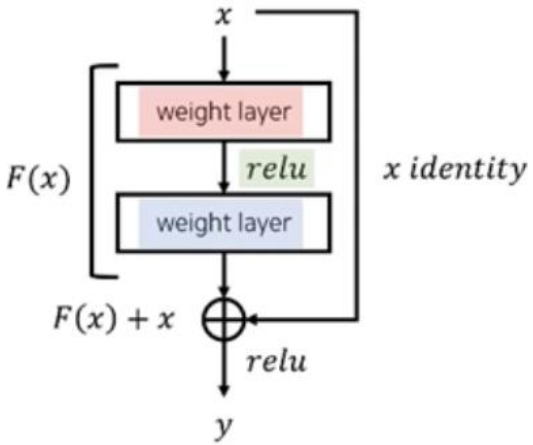
조건 !! **Input layer dimension == output layer dimension**

1. If input.ndim == output.ndim

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

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

x: *Input*
y: *Output vector*
W: *weight layer*



Identity shortcut

2. If input.ndim != output.ndim

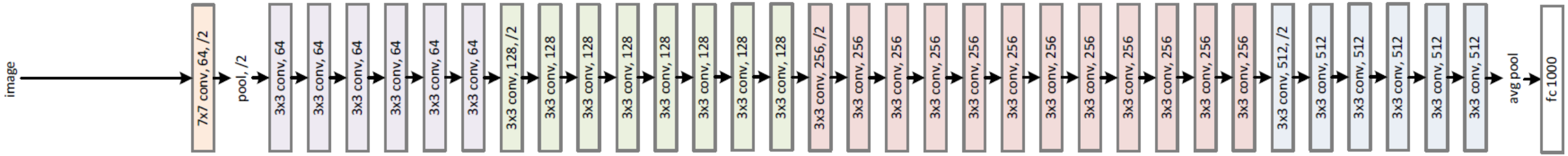
$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}.$$

x: *Input*
y: *Output vector*
W: *weight layer*
$W_s x$: *Square matrix*

Projection shortcut

3. Experiments

(0) Network Architectures : 34-layers plain vs 34- layers ResNet

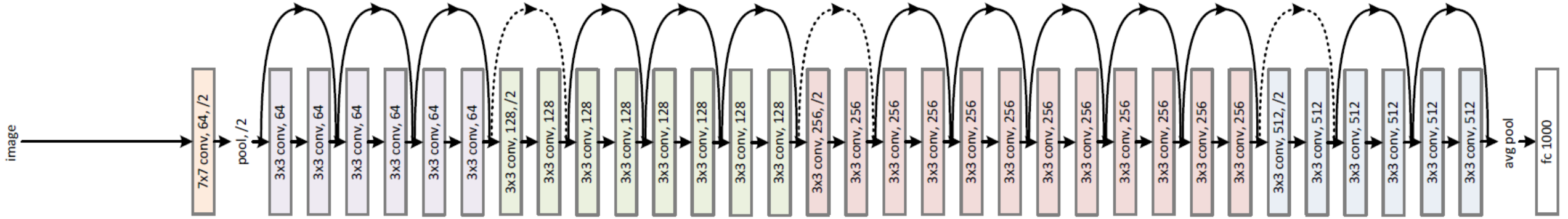


34-layers Plain network

· 기본 네트워크는 VGG 네트워크에서 제안된 기법들을 적절히 이용
· 3x3의 작은 filter이용, output feature map size에 같도록 하기 위해 같은 개수의 filter 사용
· feature map의 사이즈가 절반으로 줄어들 때는 filter의 개수를 2배로 늘려서(다음 layer의 channel 값을 2배로 늘림) 이런 방법으로 layer마다 time-complexity를 보존하는 형태로 구성
· 별도의 pooling layer 사용 X , convolution layer에서 stride 값을 2로 설정해 downsampling 진행
· layer의 마지막 부분에서 average pooling을 사용해 1000개의 클래스로 분류할 수 있도록 함
· 본 모델은 일반적인 VGG 네트워크와 비교했을 때 더 적은 파라미터 사용, 복잡도가 낮음

3. Experiments

(0) Network Architectures : 34-layers plain vs 34-layers ResNet



34-layers ResNet

· convolution filter를 두개씩 묶어서 매번 residual function 형태로 학습을 진행함

· 점선으로 표시된 것은 입력단과 출력단의 dimension이 일치하지 않아 dimension을 맞춰줄 수 있는 테크닉이 가미된 shortcut connection

· convolution layer를 두개씩 묶는 것을 총 3번반복, 그 다음 크기를 바꾸고 4번 반복, 또 다시 크기를 바꾸고 6번 반복, 크기 바꾸고 3번 반복하는 식으로 구성됨

· VGG와 비교 했을 때 FLOPs(계산 복잡도를 나타내는 척도)가 감소

3. Experiments

(1) ImageNet

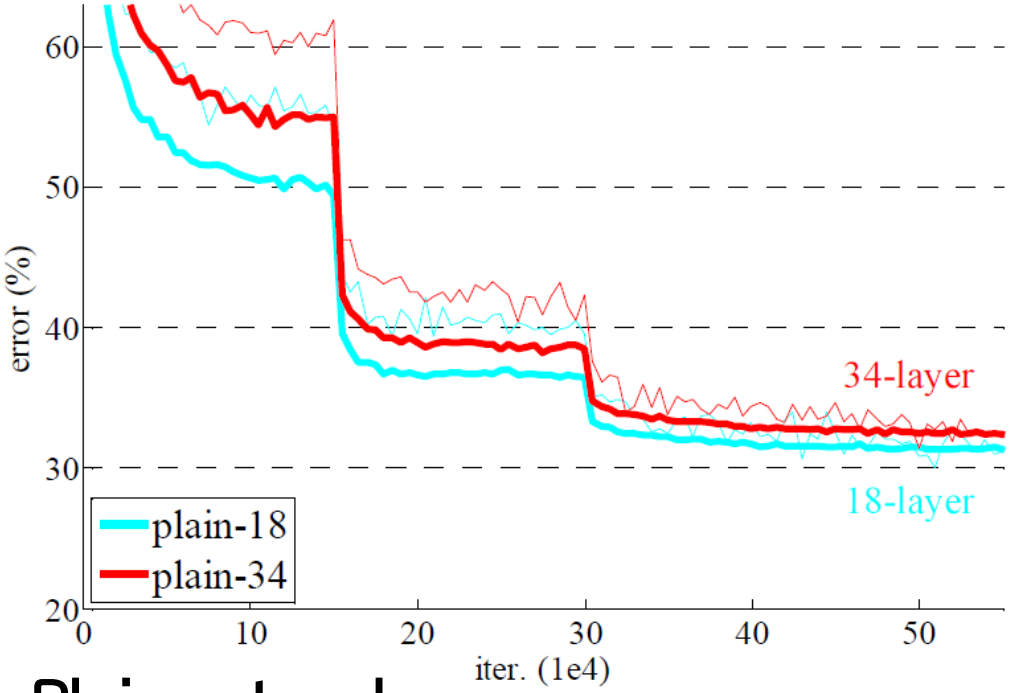
Num of Class	Num of Training images	Num of Validation images	Num of Test images
1,000	1,280,000	50,000	100,000

ImageNet Classification에서는 3가지 포인트에 대해 실험 진행

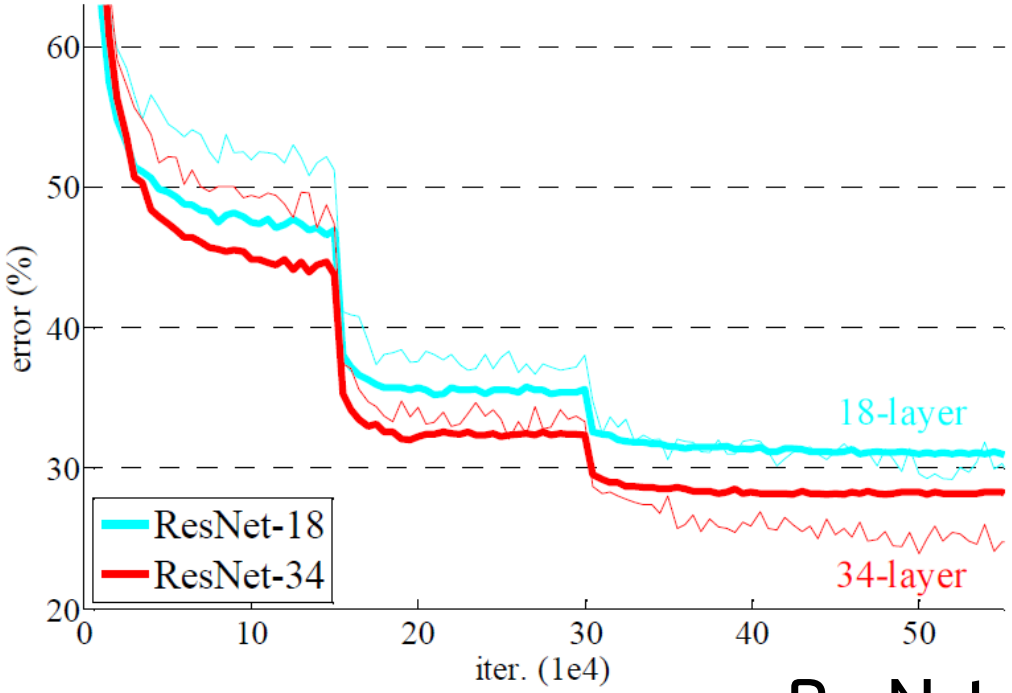
1) Comparing Plain network and ResNet
2) Identity and projection shortcut impacts on the performance
3) Comparing 34, 50, 101 and 152 layer ResNet

3. Experiments

(1)-1 ImageNet Classification : Comparing Plain network and ResNet



Plain network



ResNet

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

3. Experiments

(1)-2 Identity and projection shortcut impacts on the performance

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**

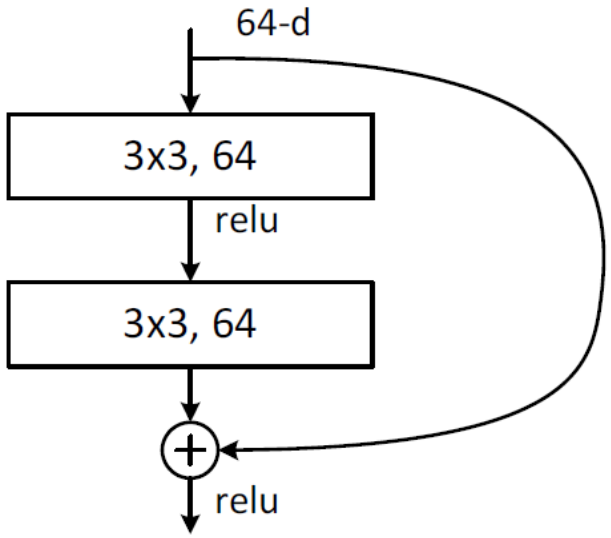
A : zero – padding을 사용해 dimension을 늘려주고 identity mapping 사용
B : dimension이 증가할 때만 projection 연산 수행
C : 모든 연산에 대해서 projection 수행

ResNet-C가 가장 좋은 결과를 보여주지만 그 차이가 미세하고 memory/ time complexity 가 늘어나기 때문에 효율적이지 X

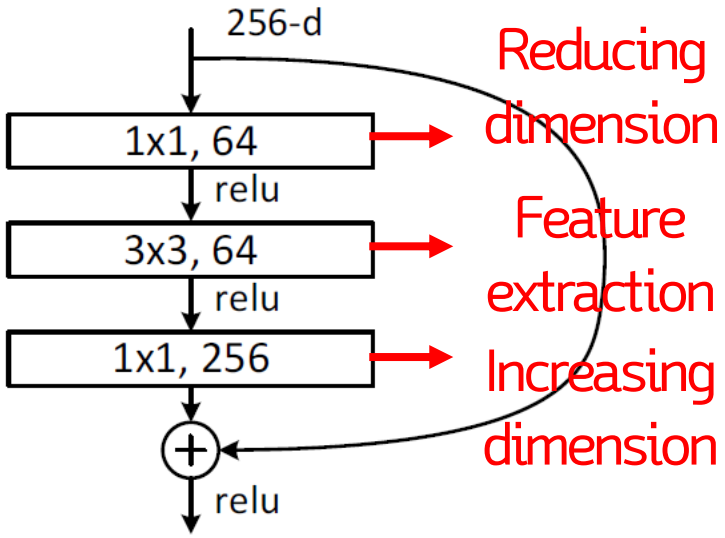
3. Experiments

(1)-3 Comparing 34, 50, 101 and 152 layer ResNet with Bottleneck Architecture

Not Bottleneck



Bottleneck



Deeper Bottleneck result

method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43†
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	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**

3. Experiments

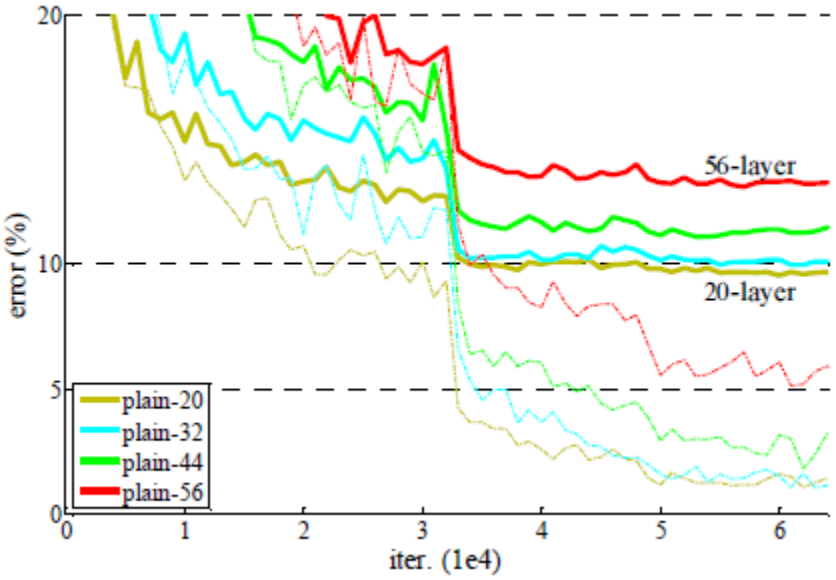
(2) CIFAR-10

Num of Class	Num of Training images	SIZE OF IMAGE	Num of Test images
10	50,000	32x32	10,000

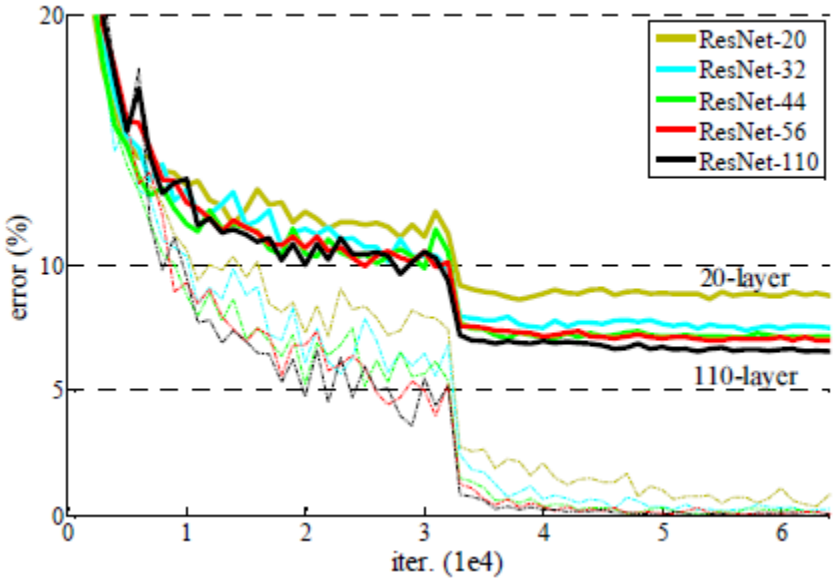
CIFAR-10 Classification은 보다 훨씬 더 깊은 네트워크에서 ResNet이 error-rate를 줄여줄 수 있는지 확인하기 위해 실험 진행

3. Experiments

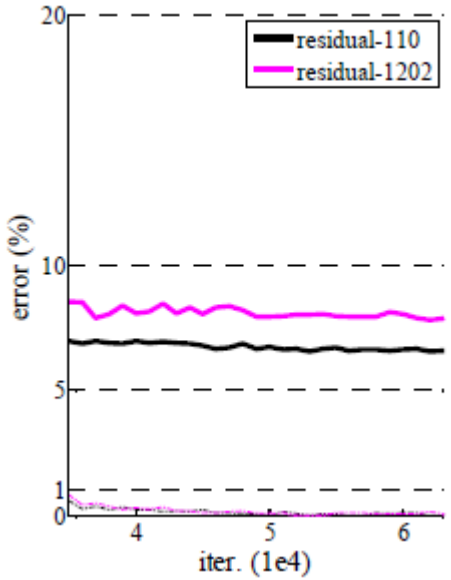
(2) CIFAR-10



Plain Network



ResNet



ResNet 110 vs 1202

3. Experiments

(3) Object detection & Segmentation task

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**

[PASCAL]

metric	mAP@.5	mAP@[.5, .95]
VGG-16	41.5	21.2
ResNet-101	**48.4**	**27.2**

[MS COCO]

-> mAP(%)값을 비교한 결과 기존 VGG 네트워크에 비해 더욱 좋은 성능을 보임

감사합니다 ^^