CS231n

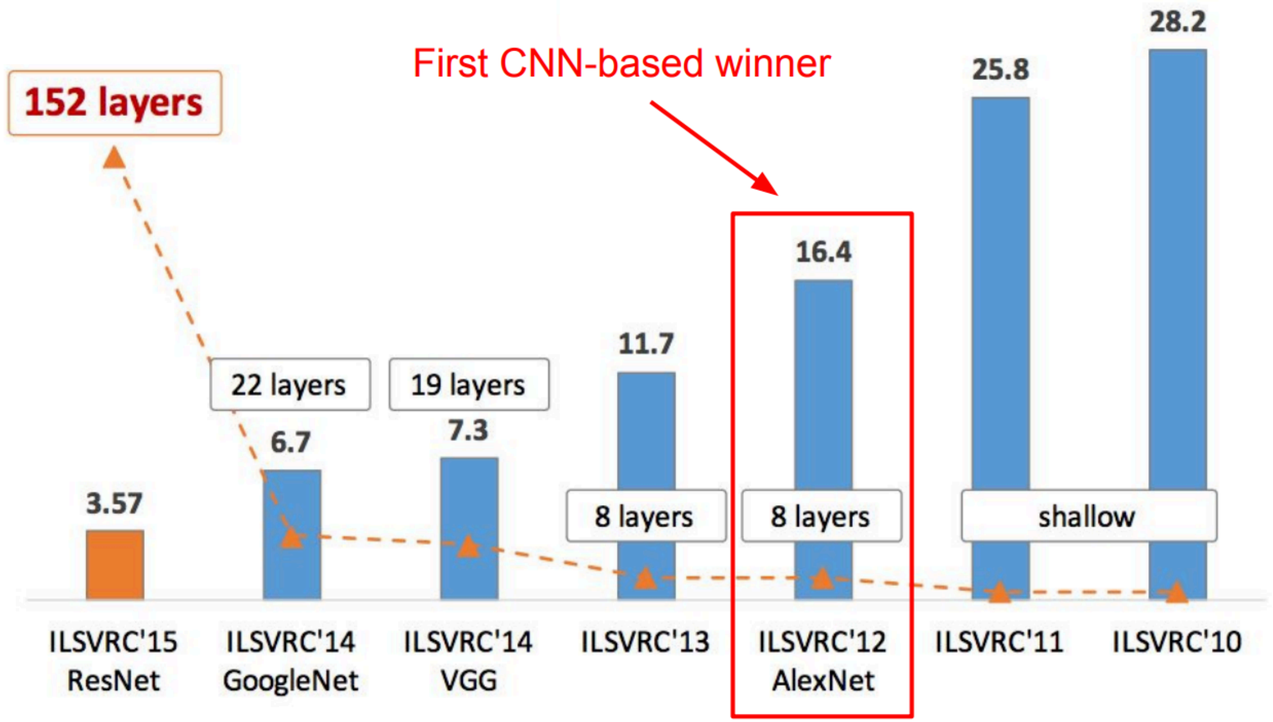
Lecture 9. CNN Architectures

Tobig's 14기 강의정

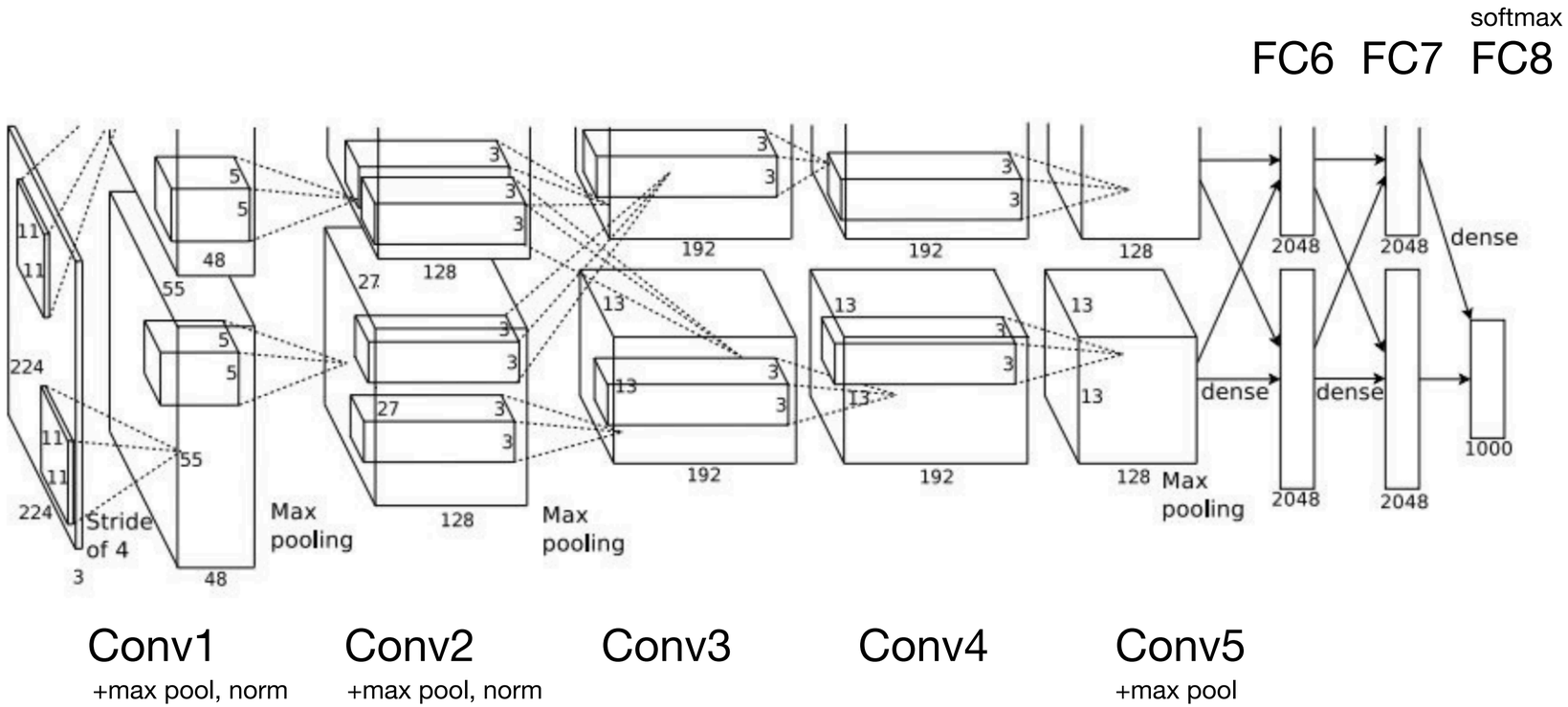
CNN Architectures

ImageNet Challenge

- AlexNet(2012)
- VGG(2014)
- GoogleNet(2014)
- ResNet(2015)

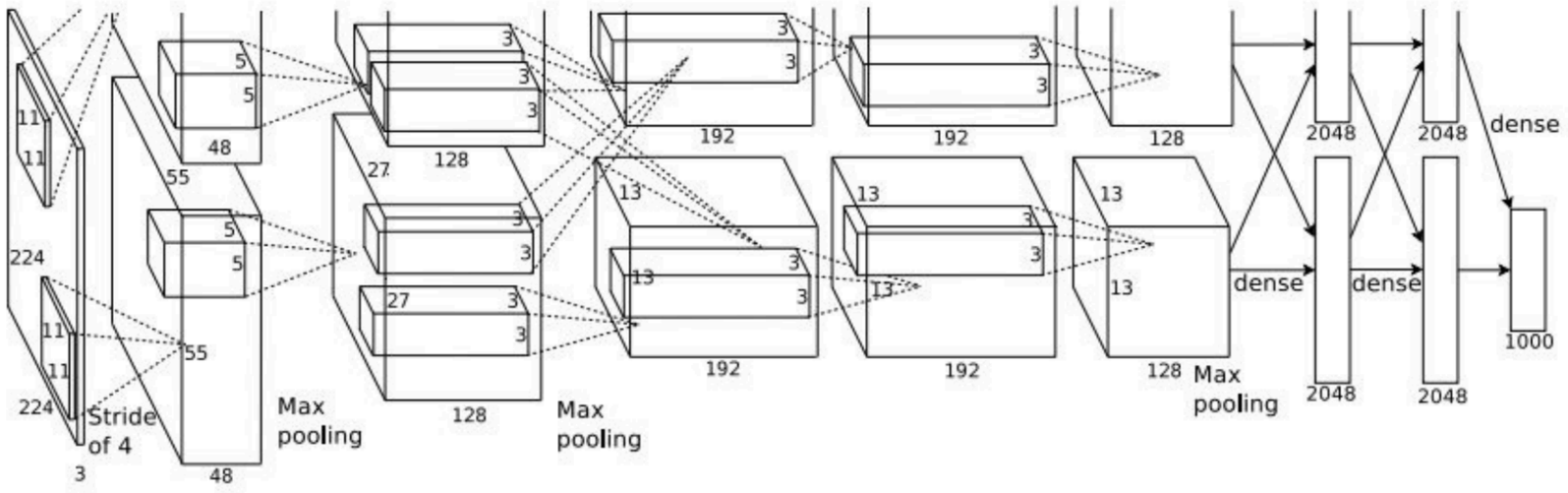


1. AlexNet



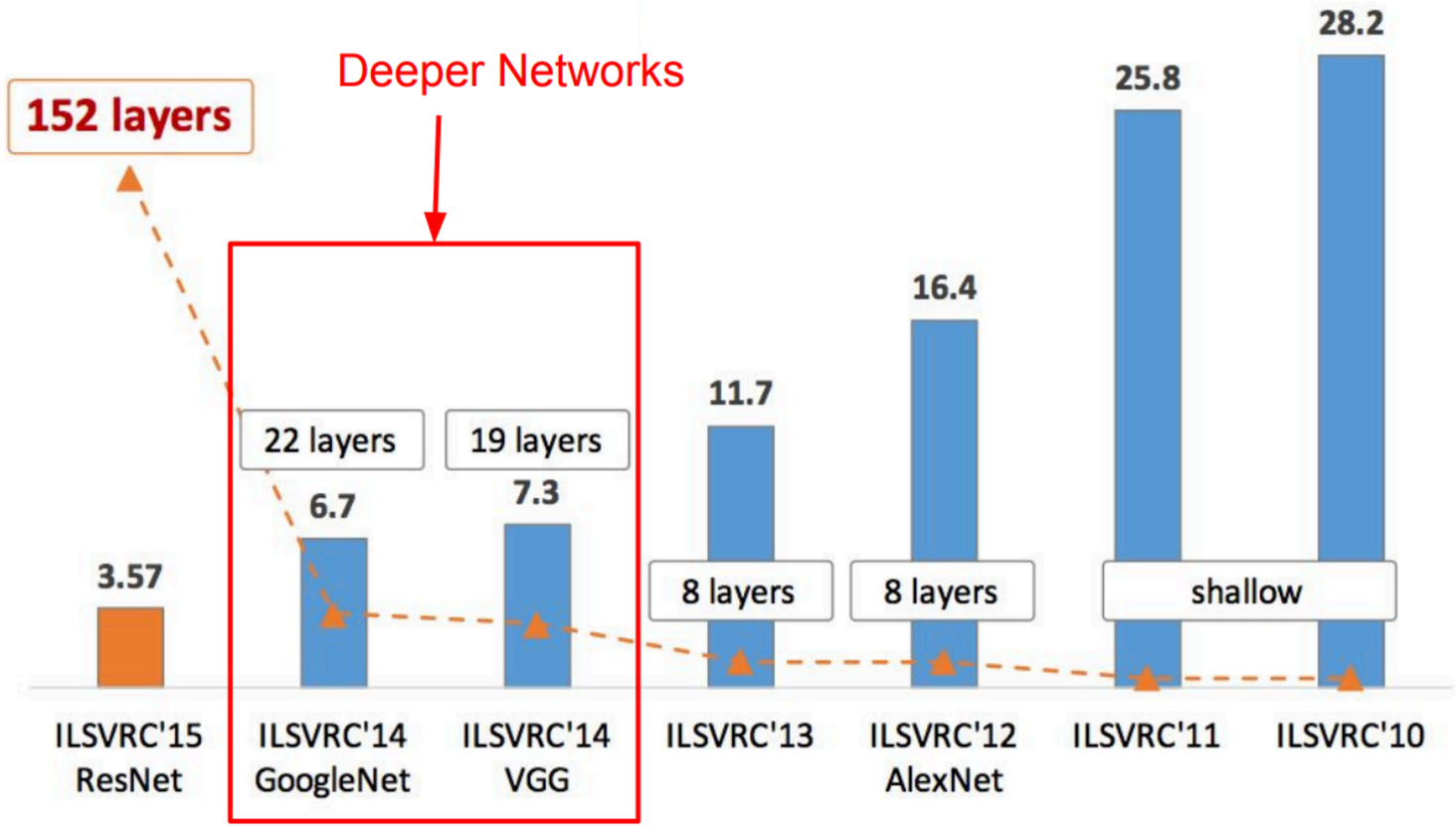
- CNN기반으로 우승한 최초의 모델 (DL, Conv Net 적용)
- 대부분의 CNN Architecture의 Base Model

1. AlexNet



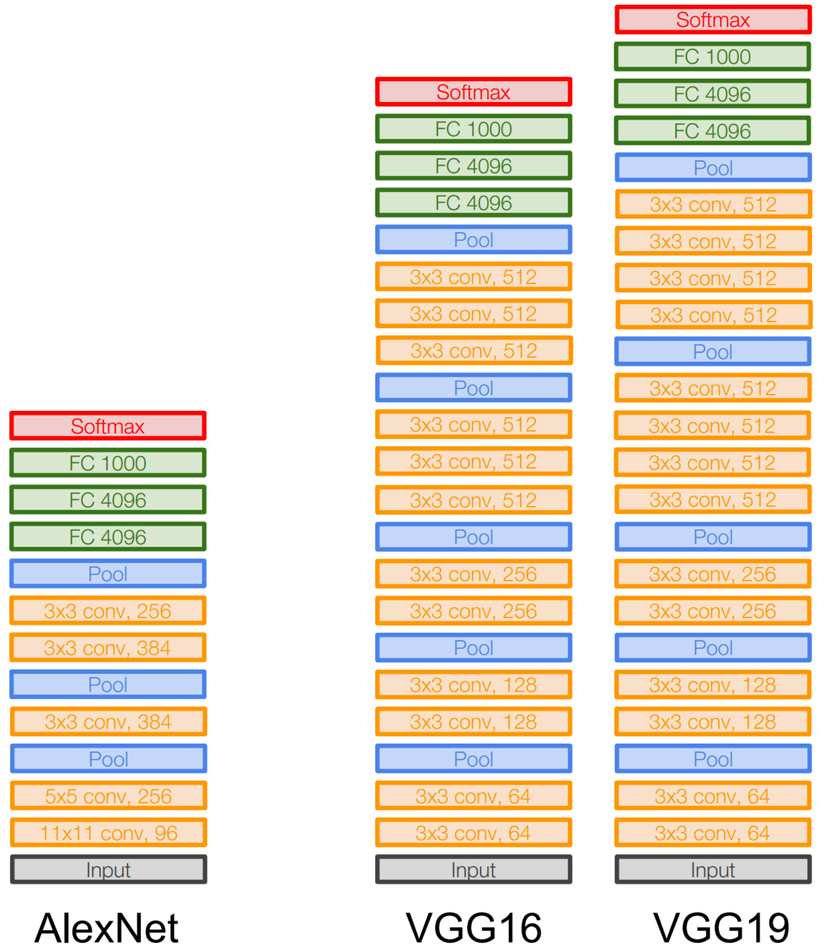
	Conv1	Pool1
Input	227 x 227 x 3	55 x 55 x 96
Fillter	96, 11 x 11 (st=4)	3 x 3 (st=2)
Output	55 x 55 x 96	27 x 27 x 96
Parameters	11*11*3*96	

VGG, GoogleNet



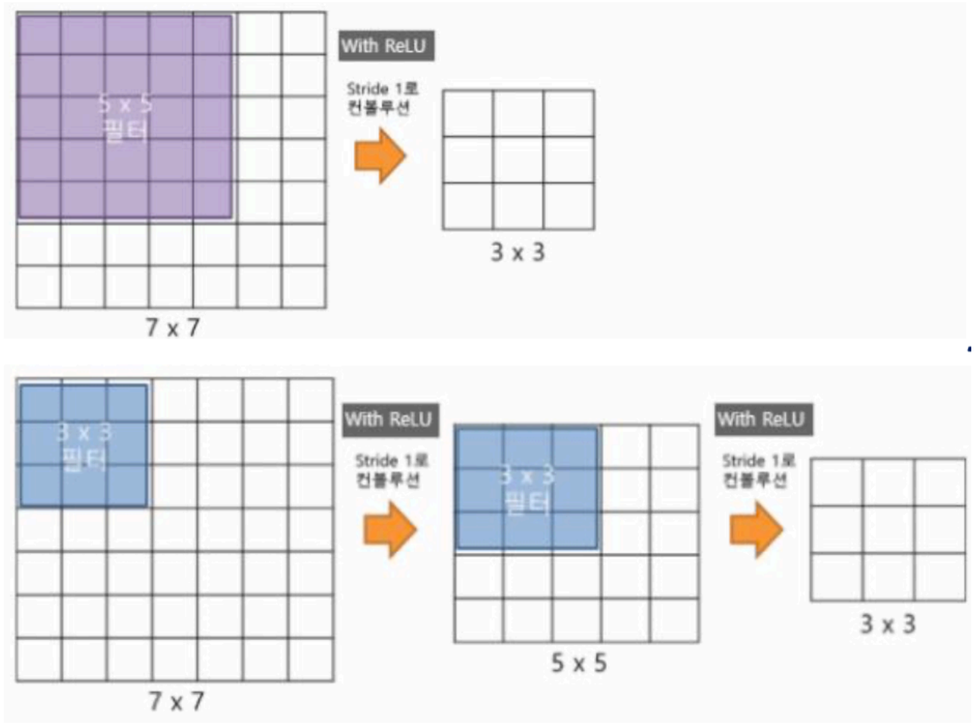
2. VGGNet

- Localization Challenge 1st
- Small Fillters, Deeper Networks
- 3 x 3 Conv, stride 1, pad 1
- 2 x 2 Max pool, stride 2
- Error : 7.3%

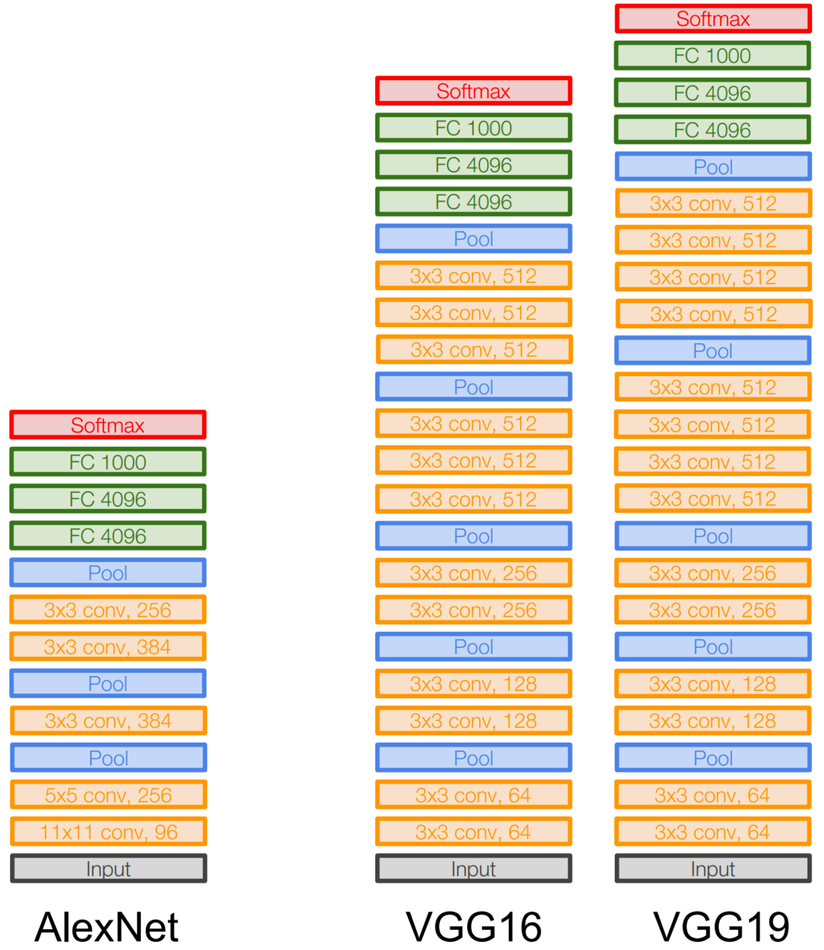


2. VGGNet

3 x 3 Fillter

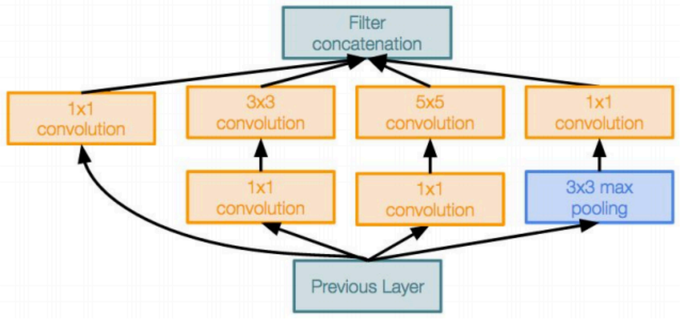


- 동일한 Recrptive Field
- 파라미터 수 81% 감소
- Non-linearity

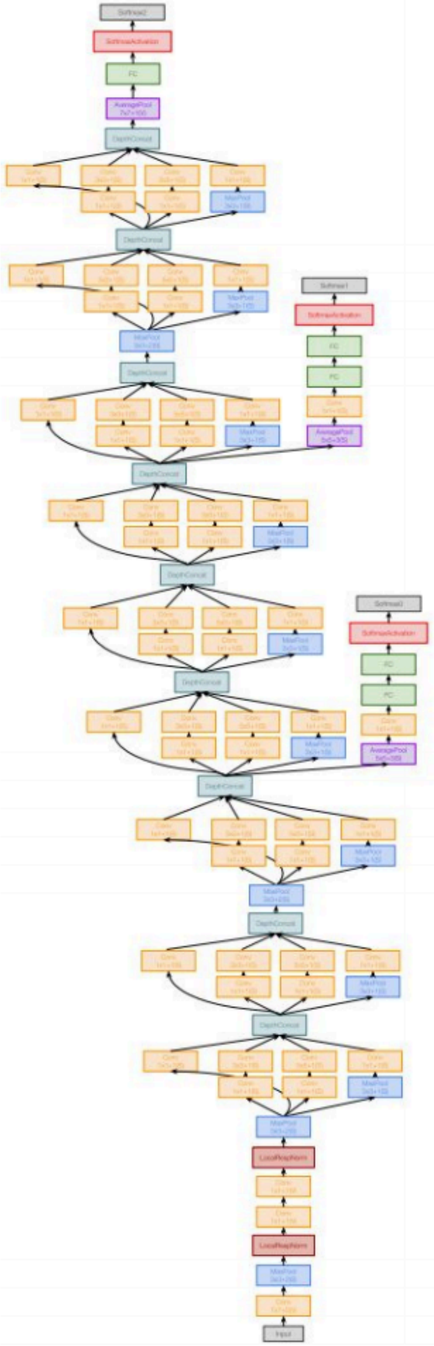


3. GoogLeNet

- Classification Challenge 1st
- Deeper networks, with computational efficiency (22 Layers)
- Inception module, No FC Layers
- 5M parameters (Alexnet : 60M parameters)
- Error : 6.7%



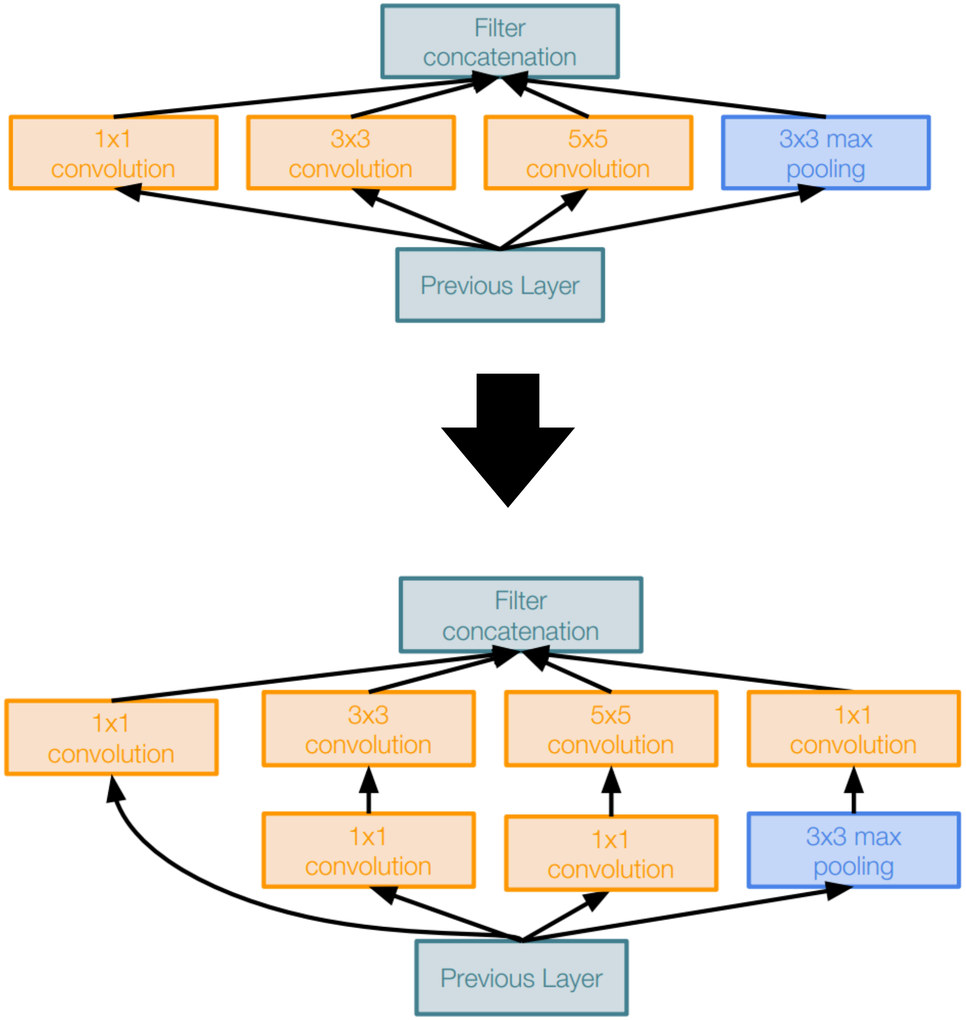
Inception module



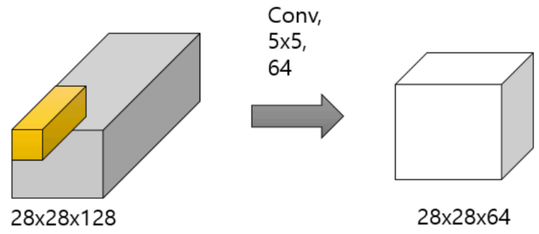
3. GoogLeNet

Inception module

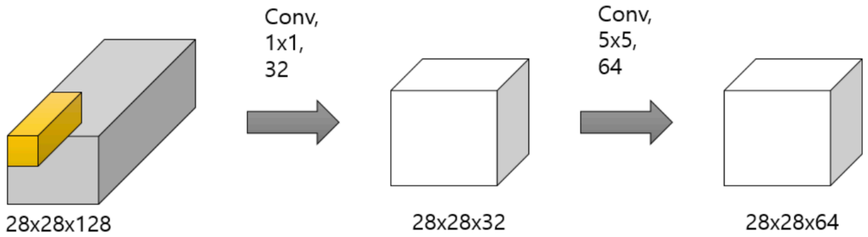
- 1 x 1 conv, 3 x 3 conv, 5 x 5 conv, 3 x 3 pool
- Bottleneck 구조



1 x 1 Convolution



#params = 28x28x64x5x5x128 = 160M

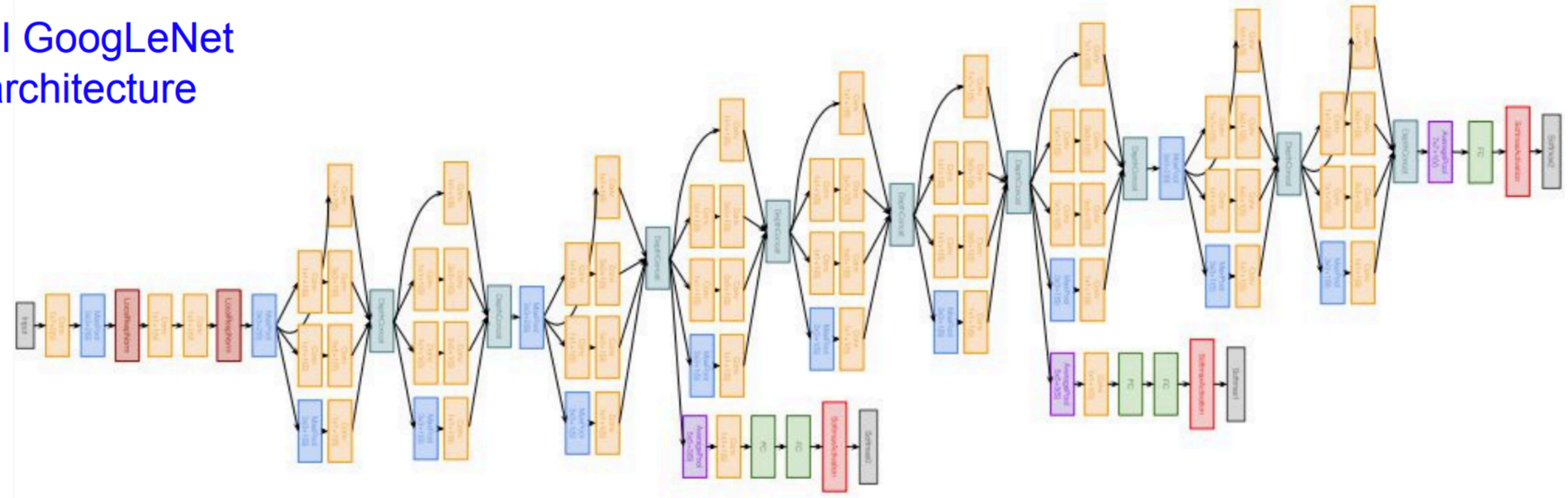


#params = 28x28x32x128x1x1 = 4.8M

#params = 28x28x64x5x5x32 = 40M

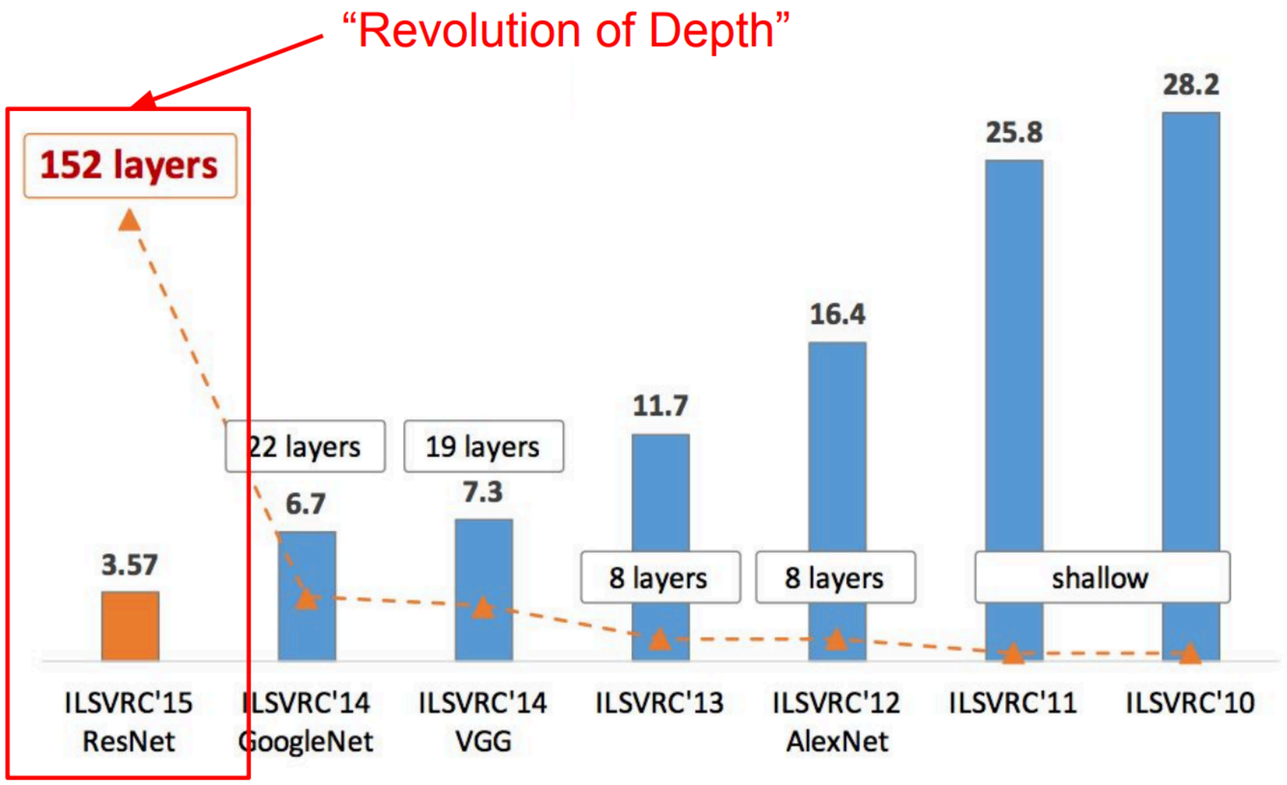
#total = 44.8M

3. GoogLeNet



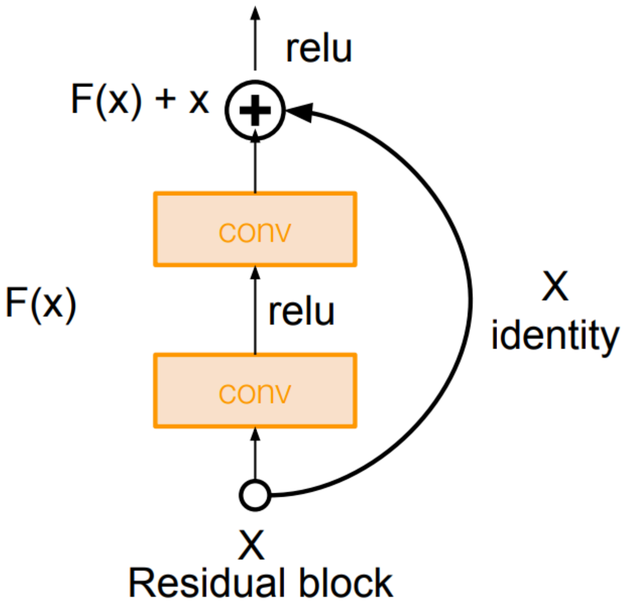
- Stem Network : 일반적인 CNN 구조
- Inception Module
- Auxiliary Classifier : gradient vanishing 문제 해결을 위해 추가적인 gradient를 얻은 보조 분류기
- Classifier

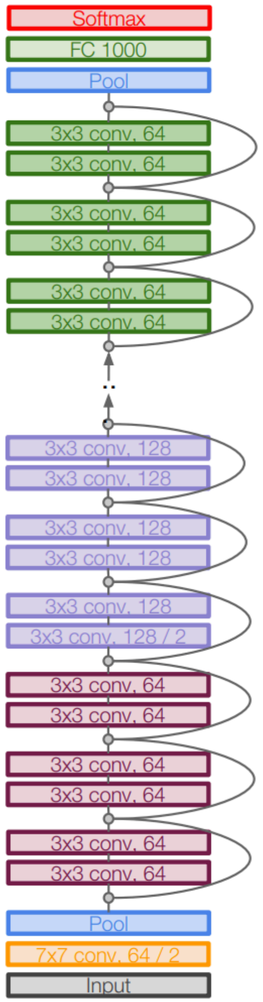
ResNet


“Revolution of Depth”
152 layers
22 layers
19 layers
8 layers
8 layers
shallow
3.57
6.7
7.3
11.7
16.4
25.8
28.2
ILSVRC'15
ResNet
ILSVRC'14
GoogleNet
ILSVRC'14
VGG
ILSVRC'13
ILSVRC'12
AlexNet
ILSVRC'11
ILSVRC'10

4. ResNet

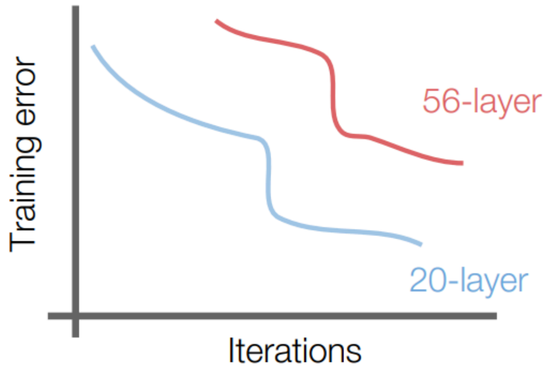
- all classification and detection competitions 1st
- very deep networks (152 Layers)
- residual connections
- Error : 3.57%

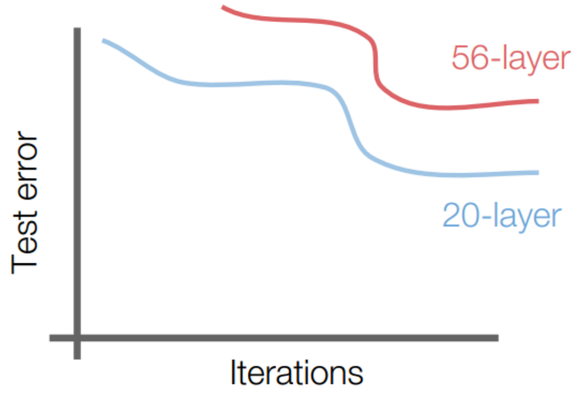




4. ResNet

[실험] Layer는 깊을 수록 좋은가?

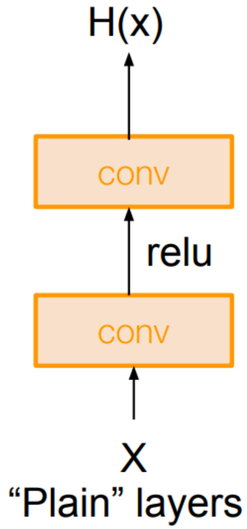


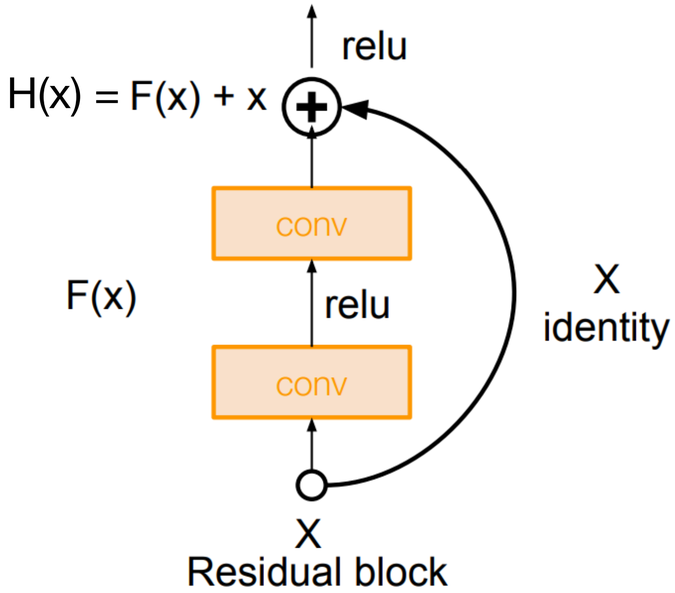


- 많은 파라미터 수는 Overfitting 문제를 줄 것이다. (X)
- 가설 : 깊은 Layer는 optimization에 문제가 생긴다.

4. ResNet

Residual block



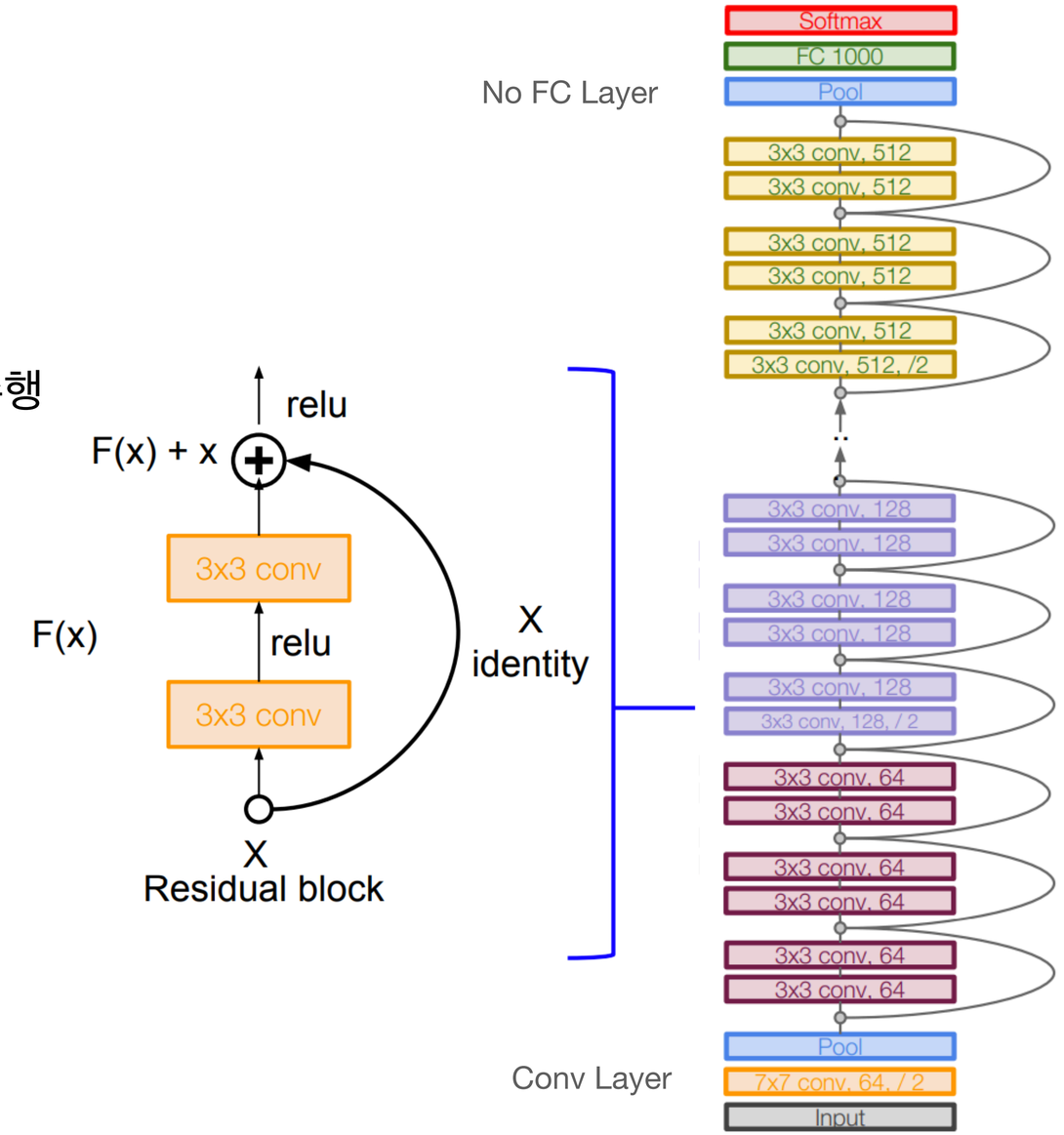


- 입력값을 추가하여 출력한다. H(x) = F(x) + x
- F(x) = H(x) - x 로 출력값에 입력값을 뺀 값, 즉 잔차이다.
- ResNet의 아이디어 : H(x) = F(x) + X 이므로 F(x)를 학습시켜보면 어떨까?
- 어떤 값이 나와야 하는가? 를 학습하는 것보다 어떻게 수정해주면 좋을까? 를 학습하자

4. ResNet

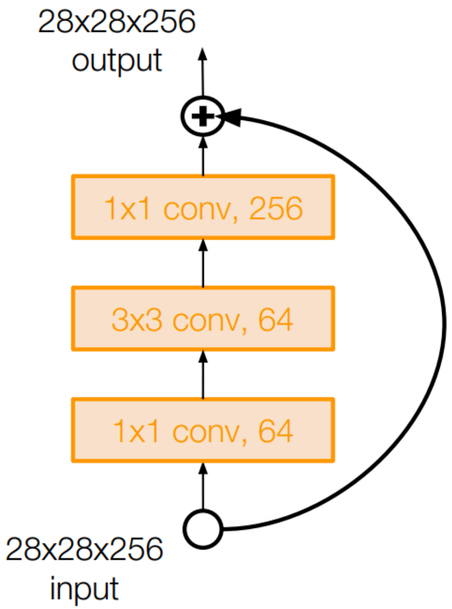
Full ResNet Architecture

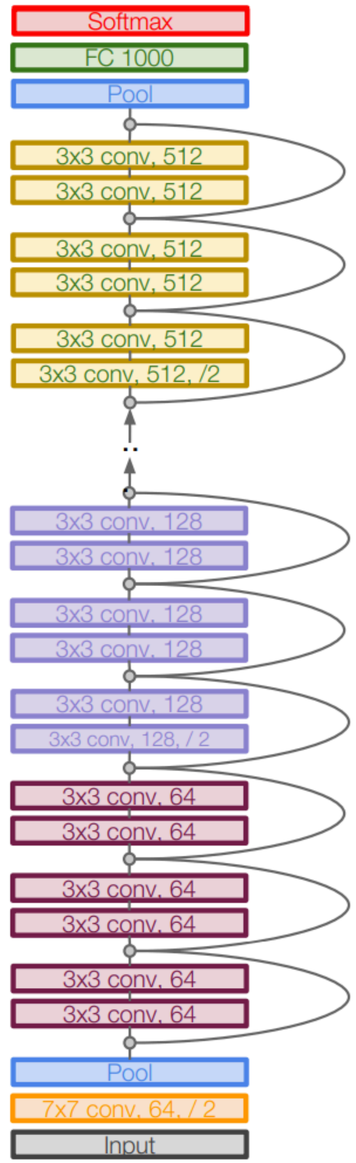
- Redisual block을 쌓은 구조 (two 3 x 3 conv)
- 주기적으로 fillter의 개수를 2배로 늘리고 Downsampling 수행
- 네트워크 처음 Conv Layer를 추가
- No FC Layer



4. ResNet

- Depth : 34, 50, 100 … 152(ImageNet문제 해결), … 1202(Cifar-10)
- Depth가 50 이상일 때 Bottleneck 도입 (1x1 conv)
- ResNet Error : 3.6%, Andrej Kapathy Error : about 5%

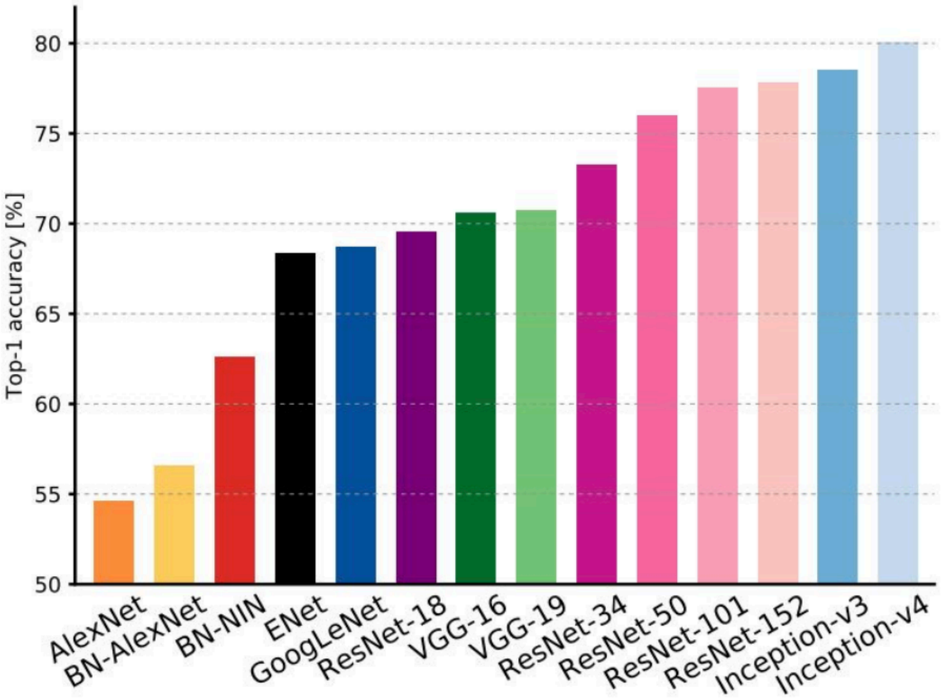


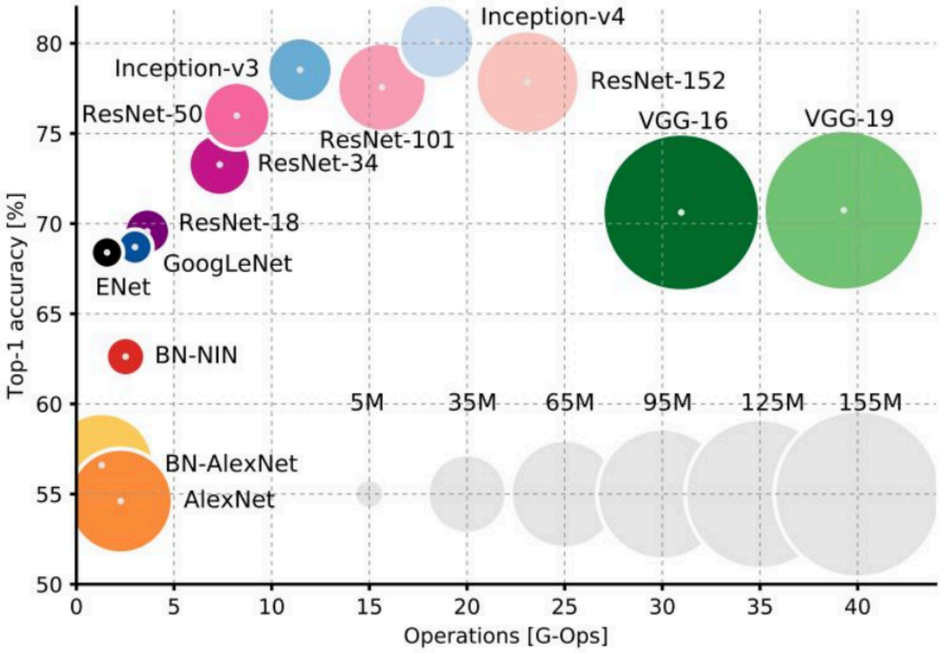


CNN Architectures

2017

Resnet + Inception



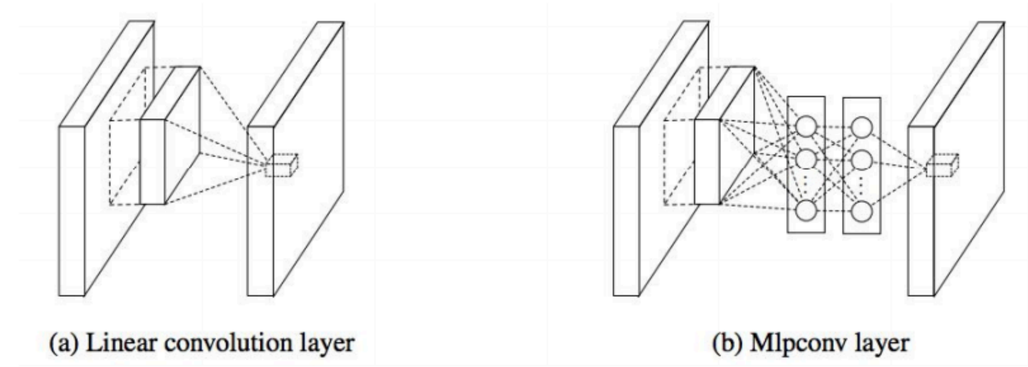


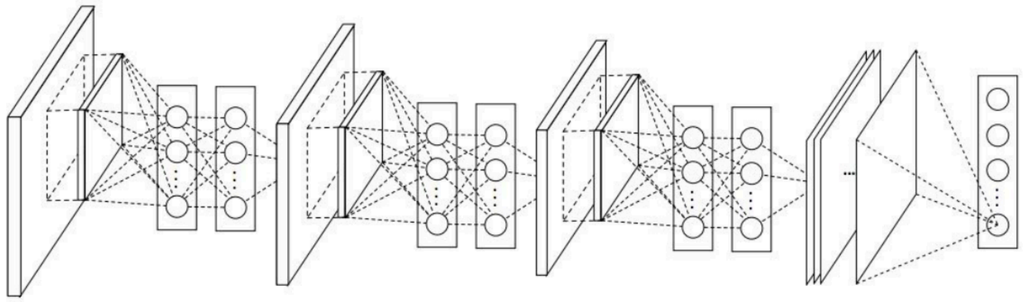
면적 : 메모리 사용량

Other Architectures

1. Network in Network(NIN)

- MLP Convolution Layer
- FC Layer (1 x 1 conv layer)
- Bottleneck 개념 정립

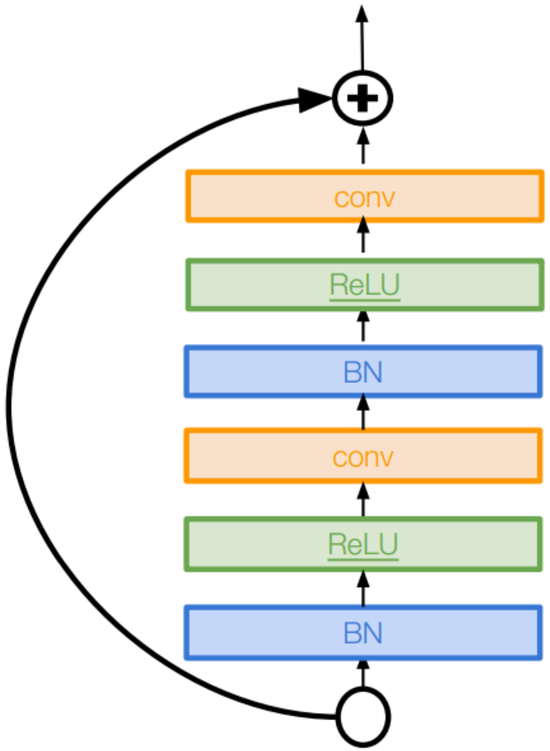




Other Architectures

2. Improved ResNet block

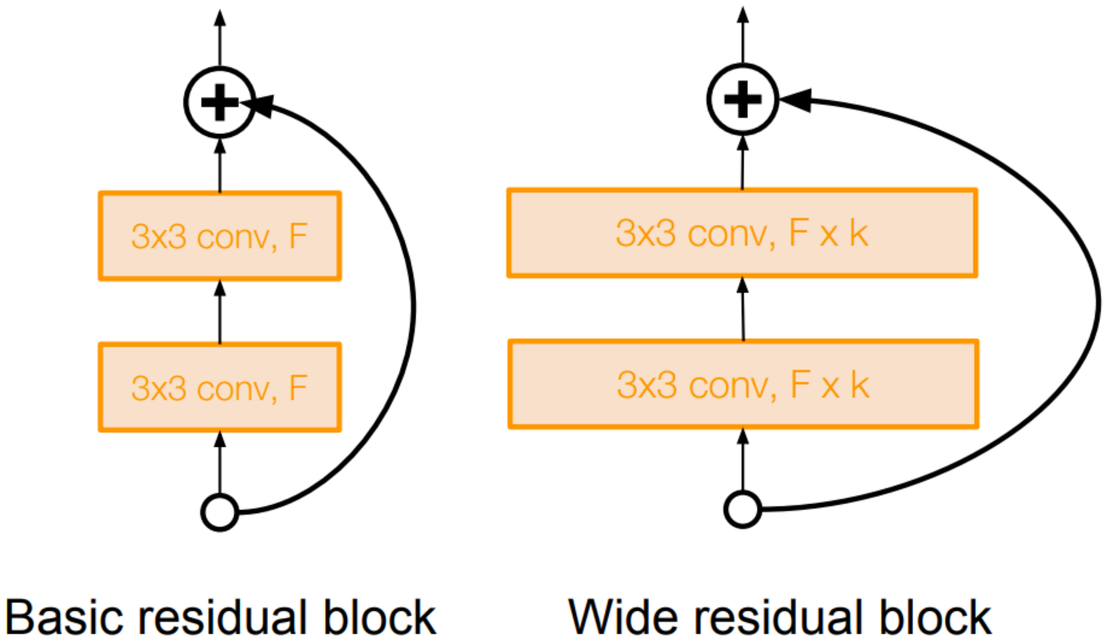
- propagation information을 위해 Direct path를 늘림
- 더 좋은 성능을 보였다.



Other Architectures

3. Wide Residual Networks

- Depth보다 Rediual이 더 중요하다는 주장
- Conv layer의 Fillter 개수 추가
- Wide ResNet(50 layers) > Basic ResNet(152 layers)

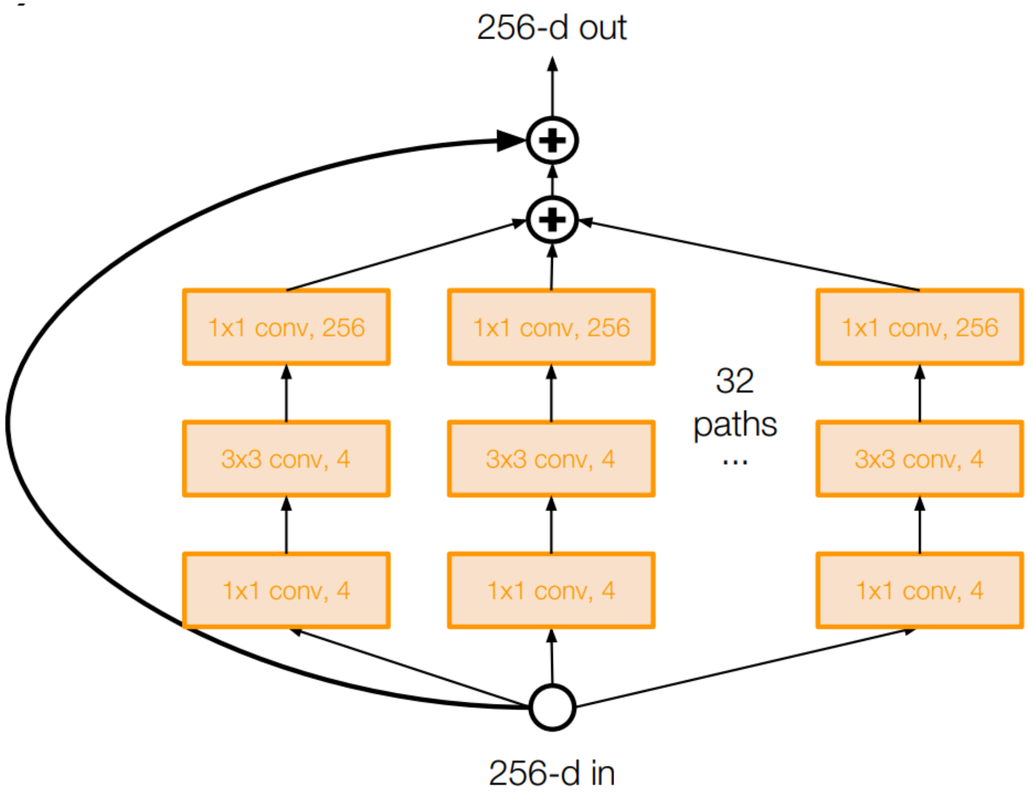


Basic residual block　　Wide residual block

Other Architectures

4. ResNeXt

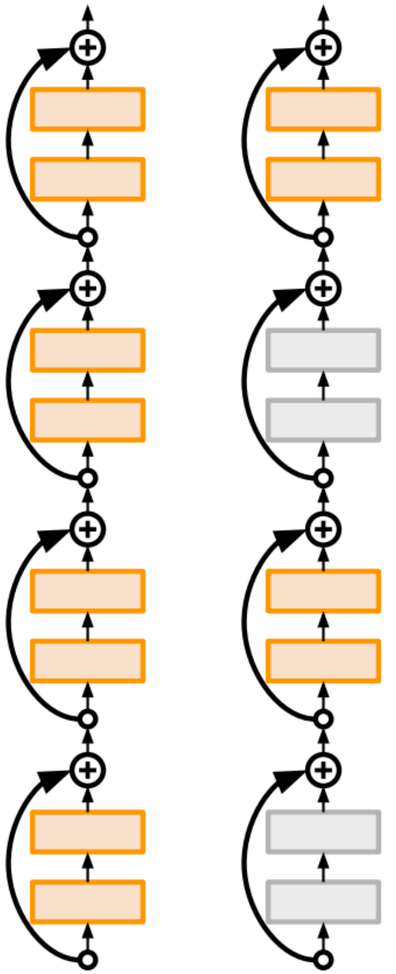
- Residual block내에 multiple parallel pathways 추가
- Inception module과 유사함



Other Architectures

5. Stochastic Depth

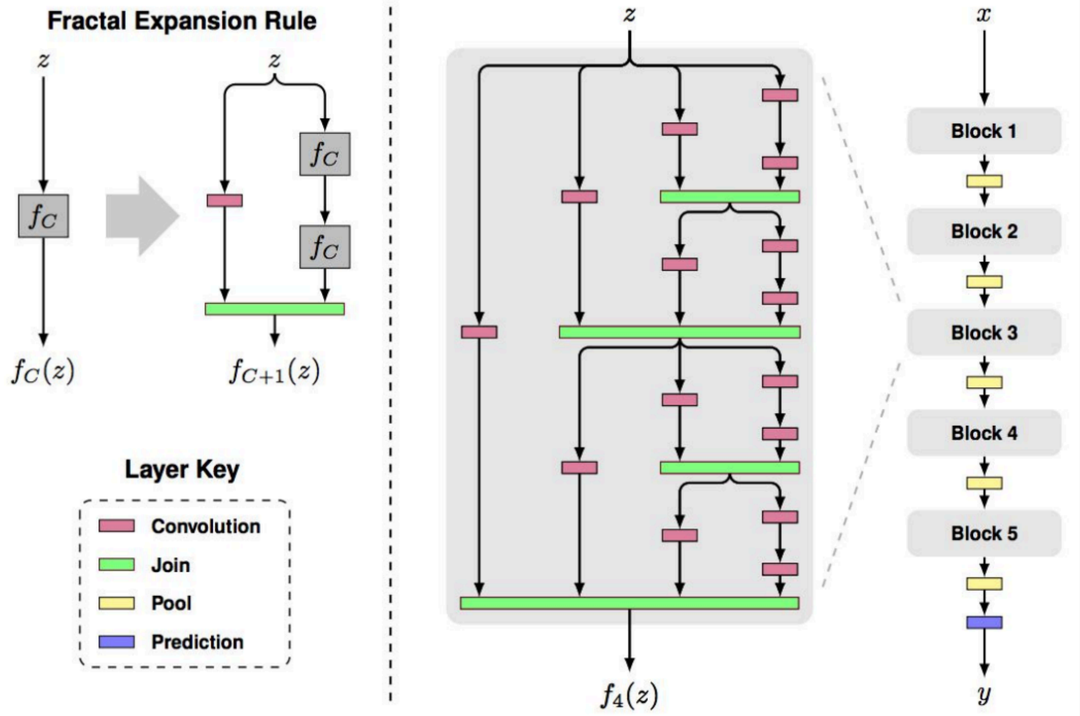
- Motivation: reduce vanishing gradients and training time
- Randomly drop a subset of layers during each training pass
- Dropout과 유사함



Other Architectures

6. FractalNet

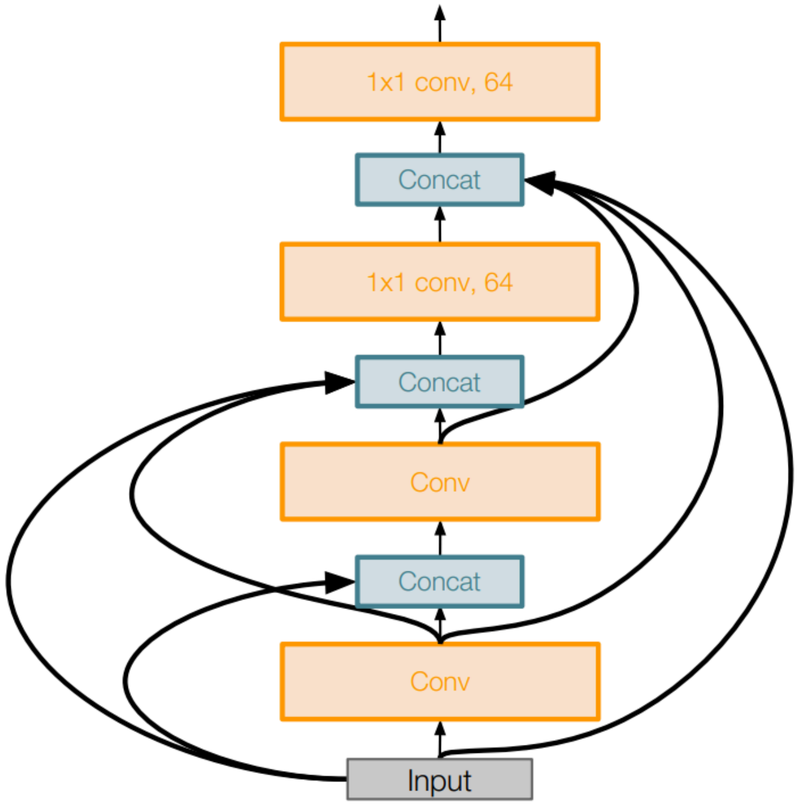
- Transitioning effectively from shallow to deep
- Residual representations are not necessary
- Trained with dropping out sub-paths
- Full network at test time



Other Architectures

7. DenseNet

- Dense Block
- 하위 레이어가 모두 연결되어 있음
- Alleviates vanishing gradient, strengthens feature propagation

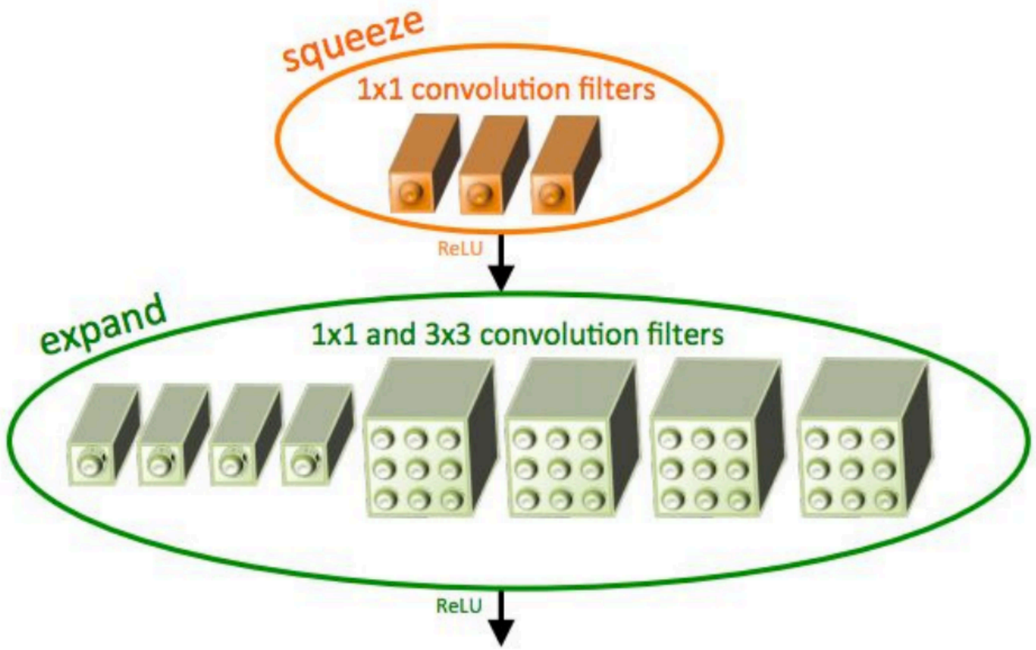


Dense Block

Other Architectures

8. SqueezeNet

- Fire Module (squeeze layer + expand layer)
- AlexNet 만큼의 Accuracy, 파라미터는 1/50



끝