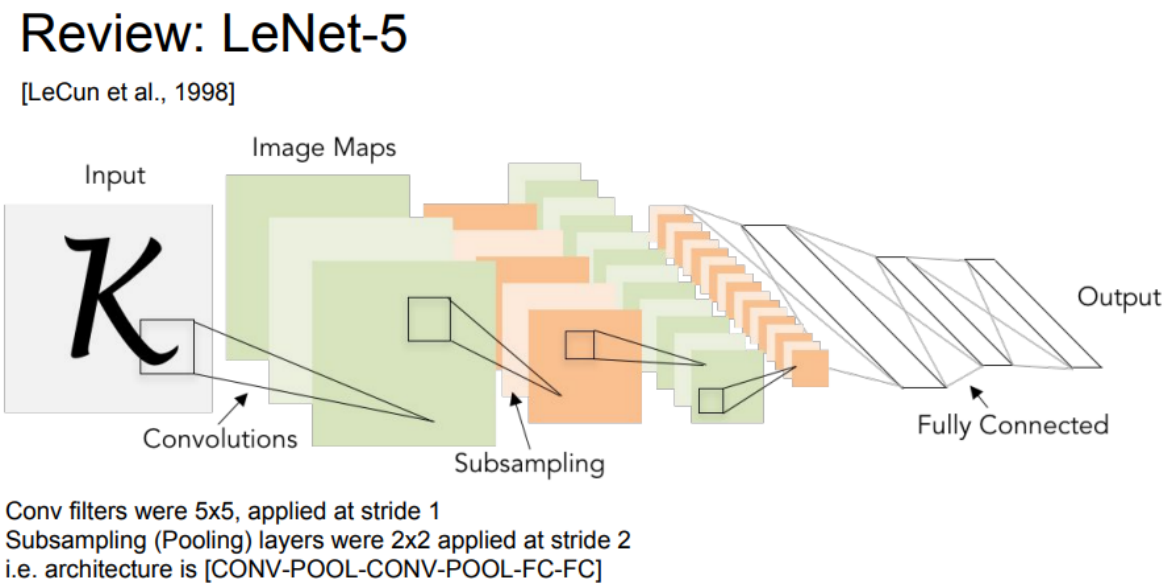
LeNet-5

- 산업에 성공적으로 적용된 최초의 ConvNet

- 이미지를 입력으로 받아 stride=1인 5X5 필터를 거치고 몇개의 Conv layer와 pooling layer을 거치는 형식으로 구성

- 마지막에 FC Layer 붙음

- 숫자 인식에 뛰어남



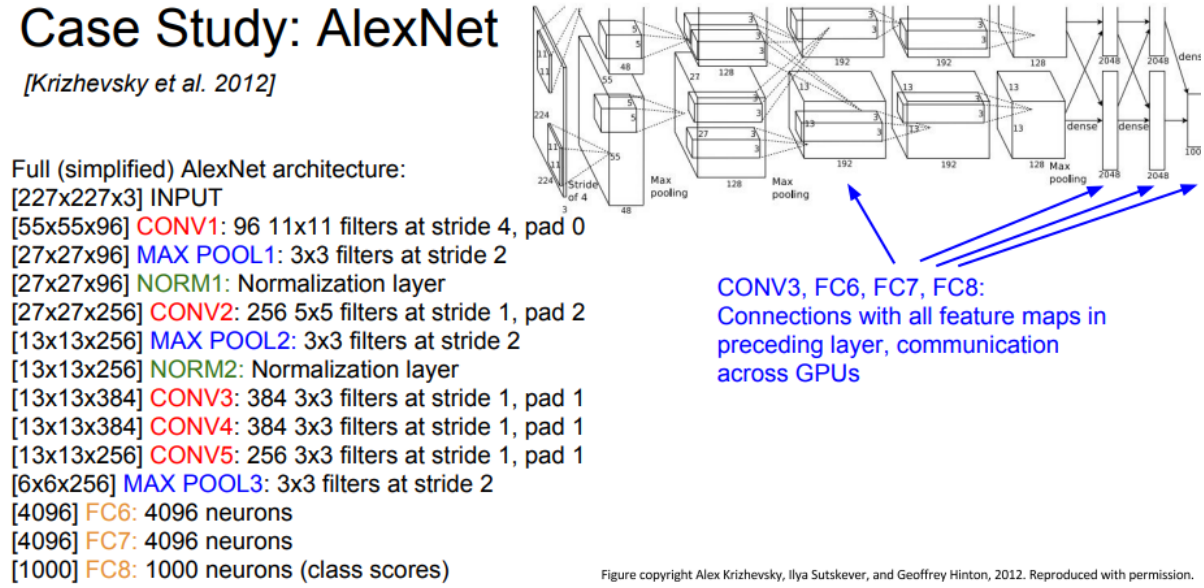
2012, AlexNet

- 최초의 large scale CNN

- ImageNet Classification Task 성능 좋음

- ConvNet 연구 부흥

- 기본적으로 cov - pool - normalization 구조가 두 번 반복. 그 뒤에 cov layer가 조금 더 붙고, 그 뒤에 pooling layer가 있음. 마지막에는 FC layer 붙음. (LeNet과 구조 유사)



- 총 5개의 Conv Layer, 2의 FC-Layer

- pooling layer에는 파라미터가 없다.

- 실제 입력은 227x227x3

- ReLU 사용, data augmentation 많이 함.

AlexNet 학습할 당시 GTX 580을 사용했고, 메모리가 3GB밖에 되지 않았기 때문에 전체 레이어를 single GPU에 분산시켜 넣음. 따라서 2개의 GPU에 feature map을 반씩 분산시켜 넣었다.

conv1,2,4,5의 경우 같은 오직 gpu의 feature map들과 연결되어 있고, conv3, fc 6,7,8의 경우 gpu간의 통신을 통해 전체 feature map과 연결되어 있음

ZFNet

- Alexnet의 하이퍼파라미터를 개선한 모델

- Alexnet과 같은 레이어 수와 구조.

- stride size, 필터 수 같은 하이퍼파라미터를 조절해서 AlexNet의 Error rate를 개선

VGGNet

- 네트워크가 훨씬 깊어짐

- 작은 필터 3x3(아웃 픽셀을 포함할 수 있는 가장 작은 필터) => depth를 더 키울 수 있음 => 계산량을 일정하게 유지

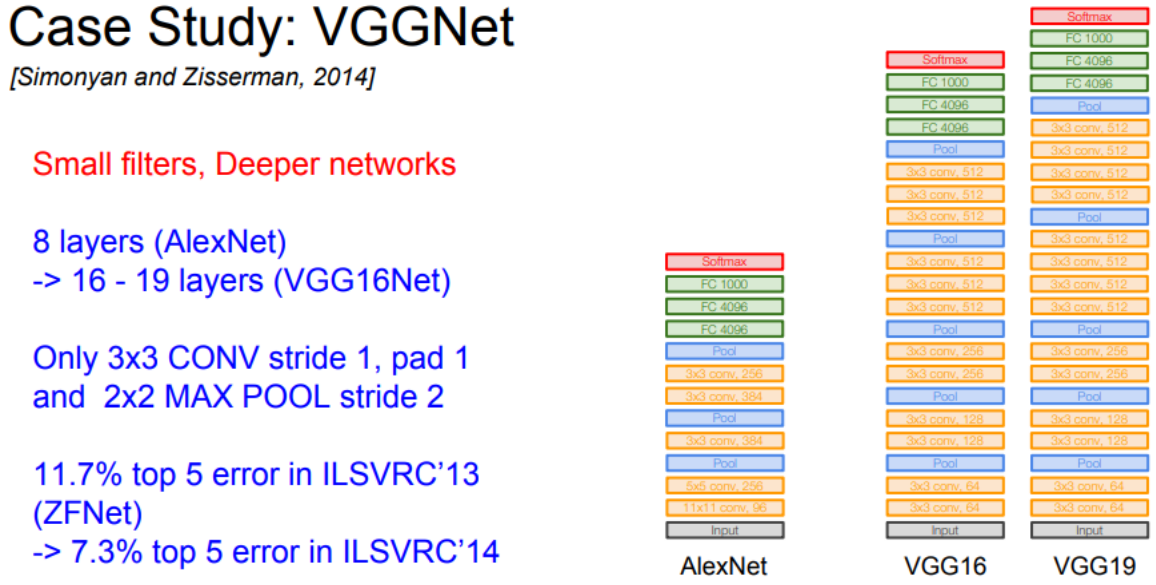
- 16~19개의 레이어

3x3 필터 3개와 7x7 하나의 receptive field는 같음. 작은 필터를 써서 parameter도 줄이고 비선형성을 더 추가하게 되고 더 깊게 만들 수 있음.

하지만 이미지 하나당 100MB라면 50장밖에 처리못함.

- 초기 레이어들이 메모리 사용량이 많음. spatial dimension이 큰 곳에 메모리 사용량 많음.

conv3-64 = 64개의 필터를 가진 3x3 conv 필터



GoogleNet

- 22개의 레이어

- Inception module을 여러개 쌓아서 만듦.

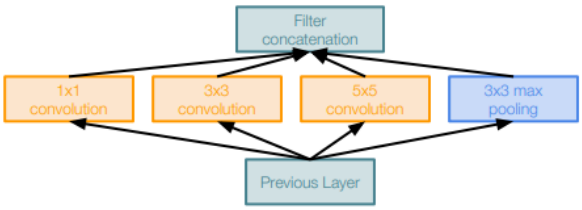
- 파라미터를 줄이기 위해 FC 레이어를 없앴으나, 더 깊게 구현했다.

- inception module

: 좋은 로컬 네트워크를 만들기 위해 네트워크 사이에 있는 네트워크라는 개념으로 local topology를 만들었고, 이 네트워크를 inception module이라고 함. 필터들이 병렬로 존재하고, 각각의 출력값들을 depth 방향으로 합치고, 이 합쳐진 하나의 tensor을 다음 레이어로 전달하는 방식.

Case Study: GoogLeNet

[Szegedy et al., 2014]



Naive Inception module

Apply parallel filter operations on the input from previous layer:

- Multiple receptive field sizes for convolution (1x1, 3x3, 5x5)
- Pooling operation (3x3)

Concatenate all filter outputs together depth-wise

문제점

: 계산 비용

모든 출력 값들을 합친 사이즈를 계산해보면 spatial dimension은 변하지 않지만 dept가 불어남. pooling layer는 입력의 depth를 그대로 유지하기 때문에 문제를 악화.

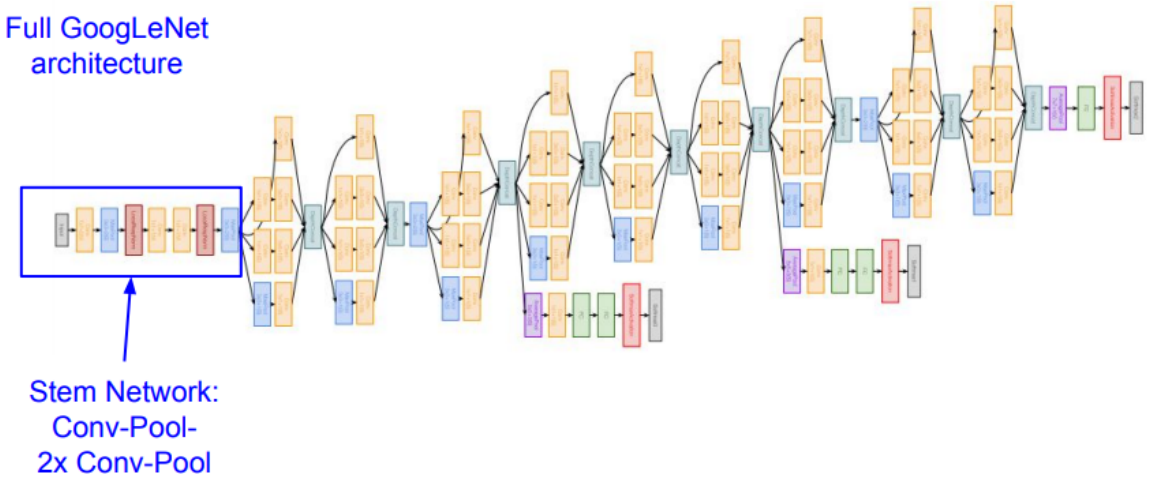
=> bottleneck layer 이용

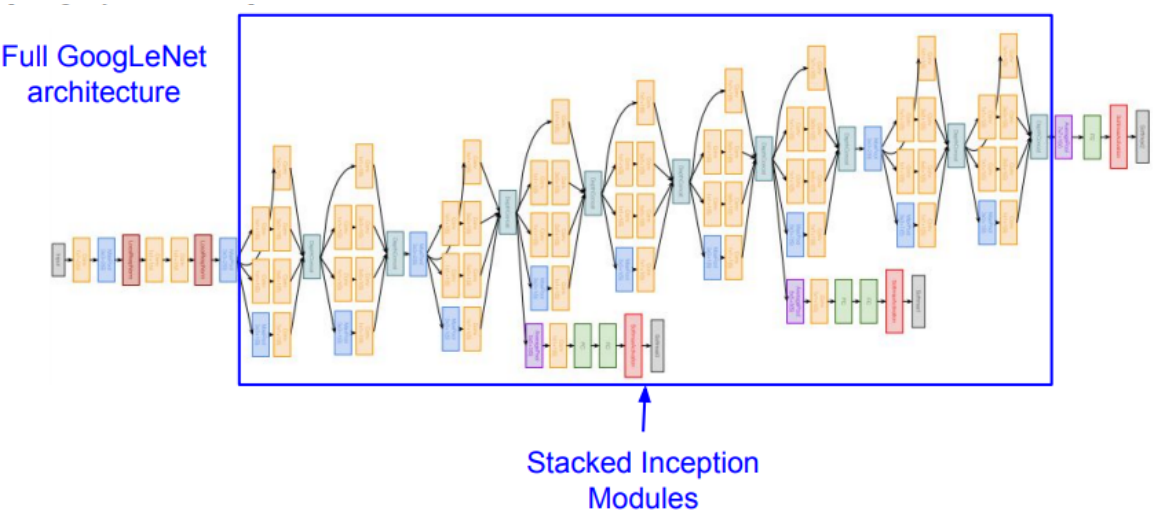
1x1 conv 필터를 이용해서 입력의 depth를 줄여 연산량을 줄일 수 있음.

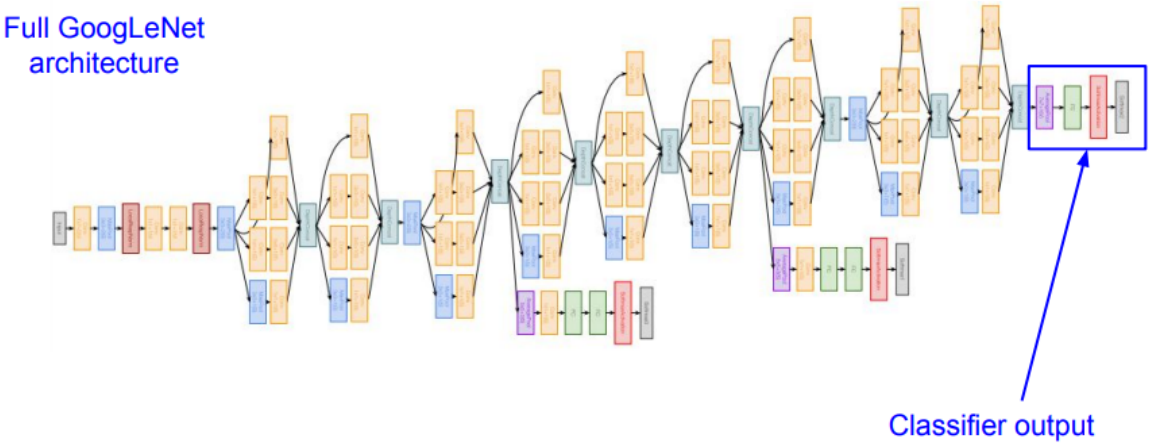
1x1 conv로 인해 정보손실이 있을 수 있으나 redundancy가 있는 input features들을 선형결합하는 것이고, 연산량을 줄여 다른 곳에 비선형 레이어를 추가함으로써 더 잘 동작하게 할 수 있음.

Case Study: GoogLeNet

[Szegedy et al., 2014]



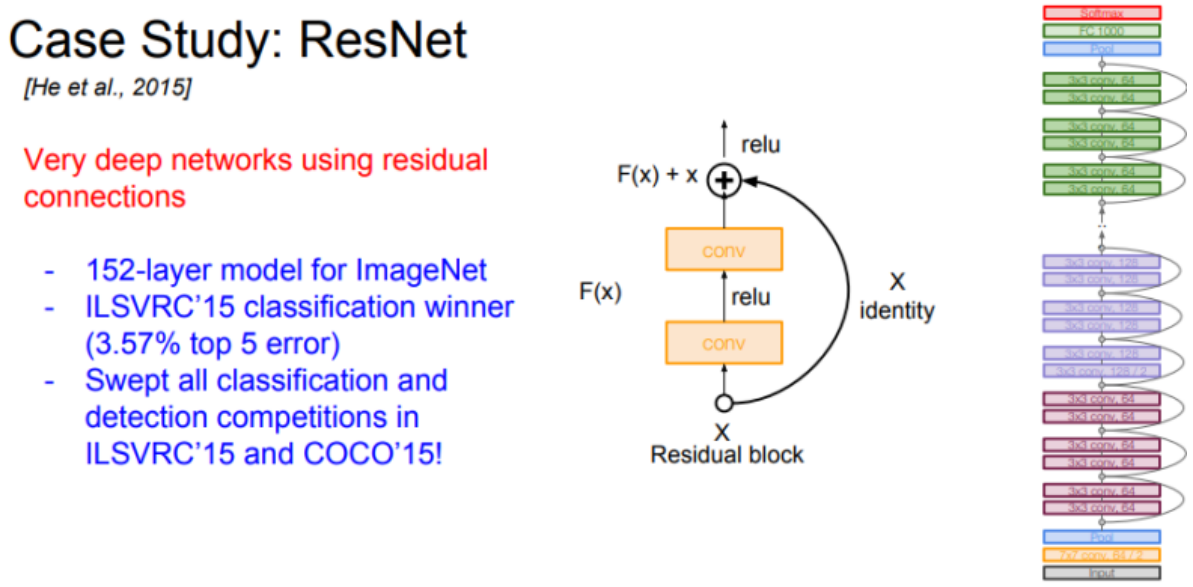




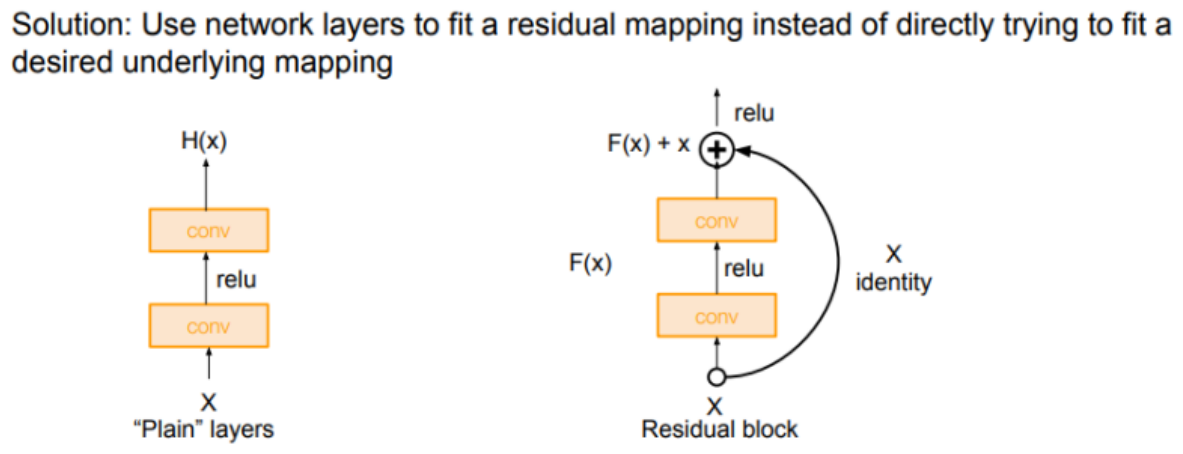
Resnet

- 엄청 깊어짐

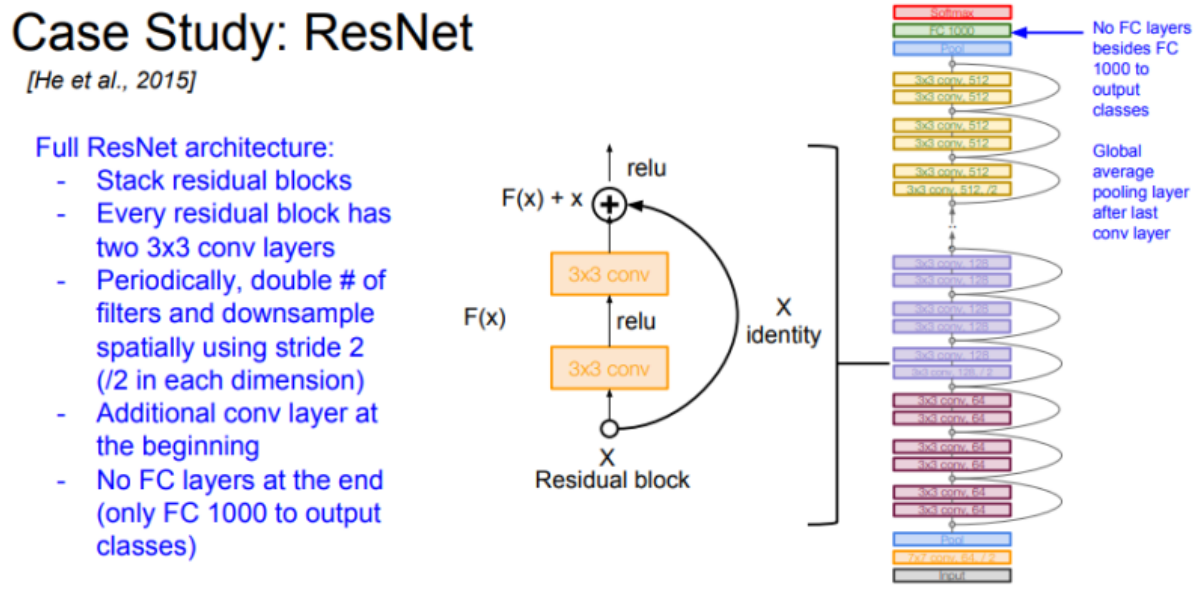
- residual connections 사용



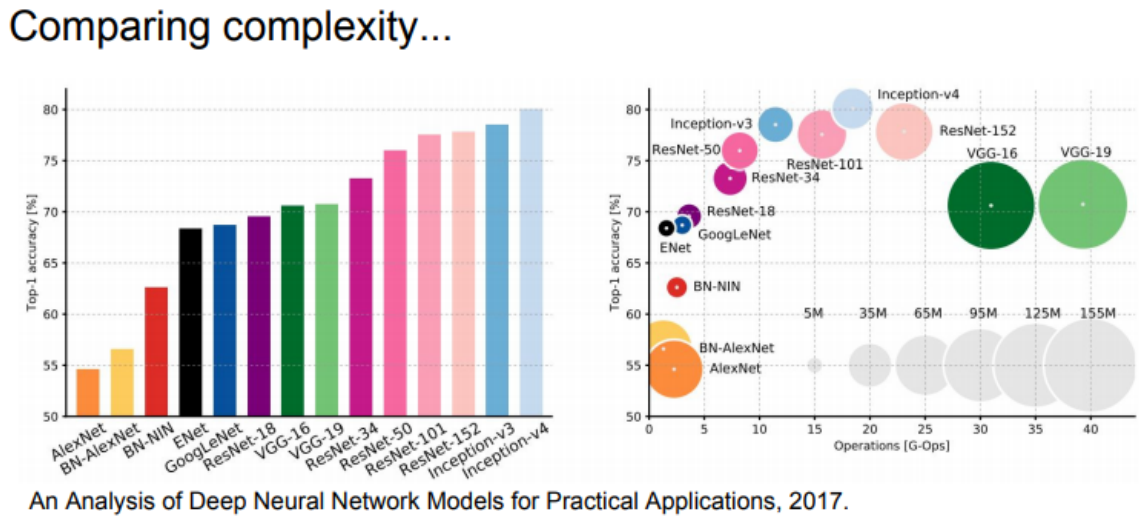
더 깊은 모델을 학습시 optimization의 문제라고 생각. 모델이 깊어질수록 최적화하기 어려워지는 것.



변화량 F(x)를 학습시키는 것이 더 쉬울 것이라 판단, 오른쪽 커브 화살표인 가중치가 없는 Skip connection을 추가하여 구성.레이어가 잘 동작하려면 레이어의 출력이 identity에 가까워야 한다는 가설을 세우고 진행. 실제 그 가설이 맞는지는 모르나 ResNet을 추가하면 성능 향상이 두드러진다.



모델 별 complexity



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

Google-inception v4(Resnet+inception) 이 가장 좋은 모델.

VGGNet: 효율성 작음. 메모리 효율 안좋고 계산량도 많다. but 성능은 봐줄만함

googlenet: 효율적.메모리 사용도 적다.

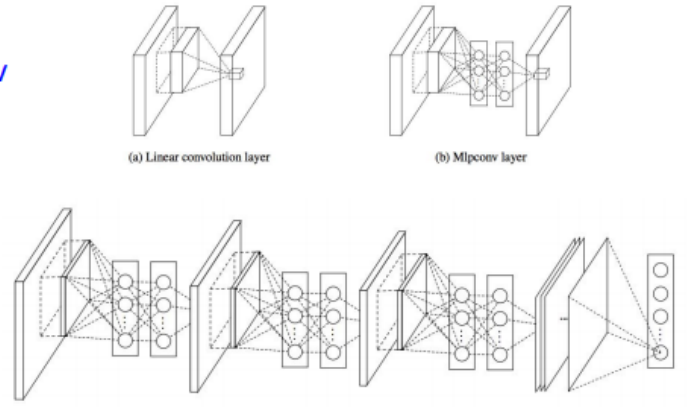
Alexnet: 초기모델인지라 성능 안좋음

Resnet: 적당한 효율성. accuracy 최상위

Network in Network (NiN)

[Lin et al. 2014]

- Mlpconv layer with "micronetwork" within each conv layer to compute more abstract features for local patches
- Micronetwork uses multilayer perceptron (FC, i.e. 1x1 conv layers)
- Precursor to GoogLeNet and ResNet "bottleneck" layers
- Philosophical inspiration for GoogLeNet



Figures copyright Lin et al., 2014. Reproduced with permission.

각 conv layer안에 Multi layer perceptron(FC)을 쌓아서 네트워크 안에 작은 네트워크를 만드는 방식. FC layer, 즉 1x1 conv layer을 사용해서 abstract features을 더 잘 뽑을 수 있도록 하고자 함. googlenet, resnet보다 먼저 보틀넥의 개념 정립

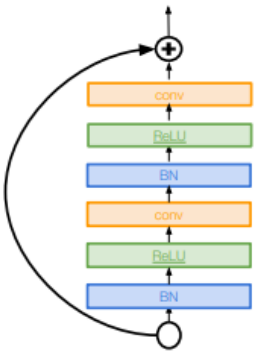
ResNet 블록 향상

Improving ResNets...

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



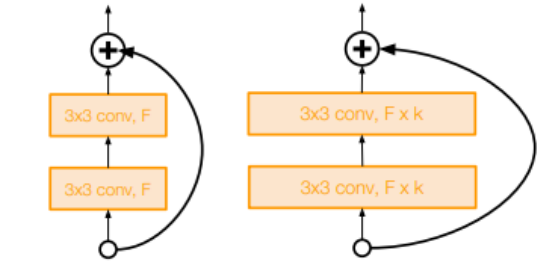
direct path를 늘려서 forward, backprop이 더 잘 될 수 있도록 설계

Improving ResNets...

Wide Residual Networks

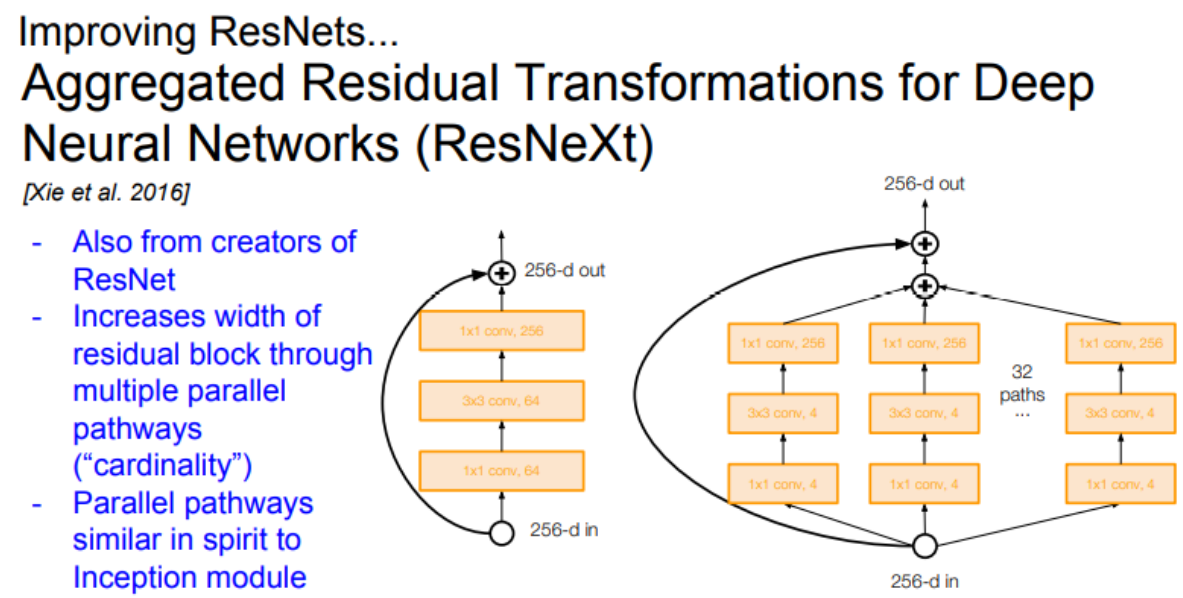
[Zagoruyko et al. 2016]

- 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)



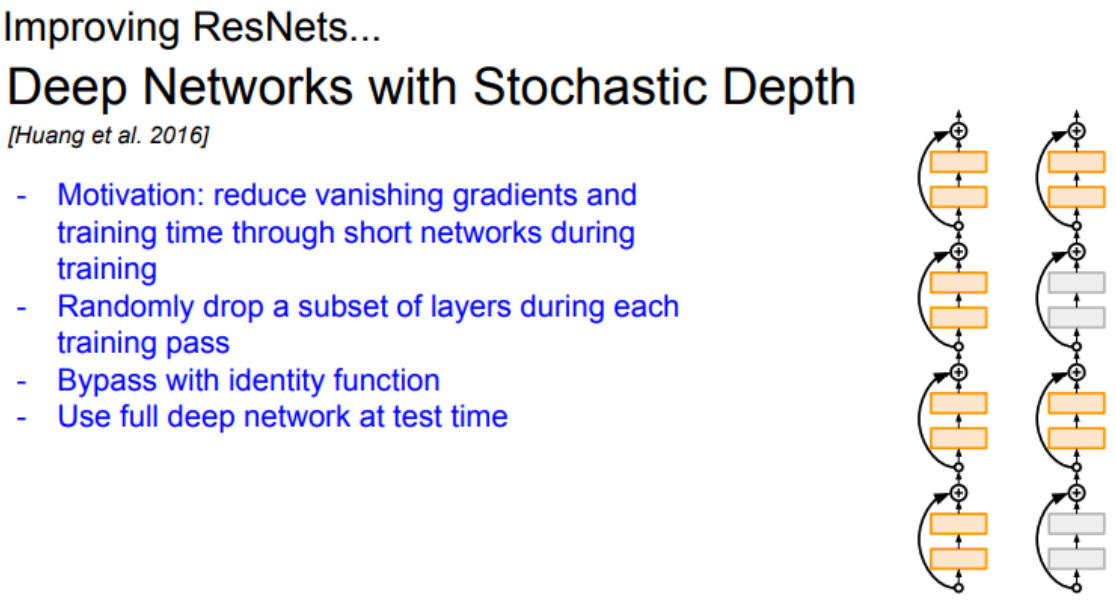
resnet에서 depth가 아닌 residual이 중요하다고 주장. residual block의 conv layer 필터를 더 많이 추가해서 더 넓게 만듦. 각 레이어가 넓어져서 50레이어만 있어도 152 레이어의 기존 성능보다 좋음

filter의 width를 늘리면 병렬화가 더 잘되어서 계산 효율이 증가해서 좋음.



ResNet+Inception

residual block 내에 다중 병렬 경로를 추가함. 마찬가지로 width늘리기



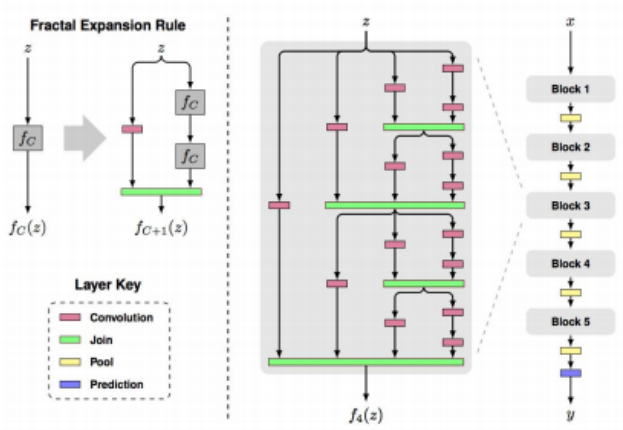
네트워크가 깊어질수록 Vanishing gradient 문제가 발생. train time에 일부 네트워크를 골라서 identity connection으로 만듦. shorter network가 되어 그라디언트가 더 잘 전달. Dropout과 유사. test time에는 full deep network 사용

Beyond ResNets...

FractalNet: Ultra-Deep Neural Networks without Residuals

[Larsson et al. 2017]

- Argues that key is transitioning effectively from shallow to deep and residual representations are not necessary
- Fractal architecture with both shallow and deep paths to output
- Trained with dropping out sub-paths
- Full network at test time



Figures copyright Larsson et al., 2017. Reproduced with permission.

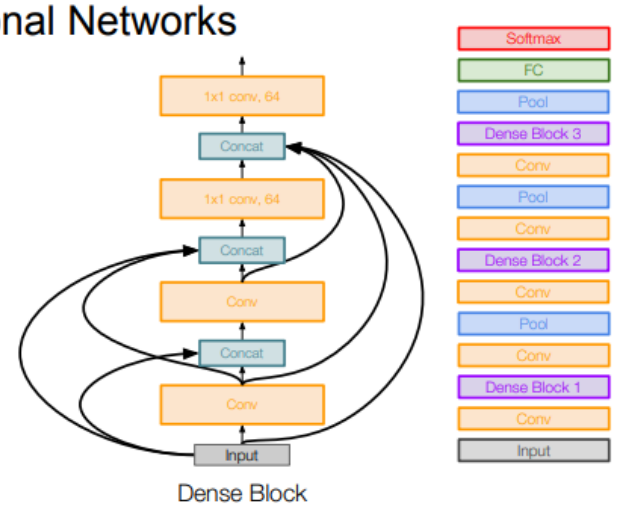
residual connection이 쓸모없다 주장. shallow/deep 경로를 출력에 모두 연결. train time에는 일부 경로만으로 학습. 성능 좋음

Beyond ResNets...

Densely Connected Convolutional Networks

[Huang et al. 2017]

- Dense blocks where each layer is connected to every other layer in feedforward fashion
- Alleviates vanishing gradient, strengthens feature propagation, encourages feature reuse



Dense Block

네트워크의 input 이미지가 모든 레이어의 input으로 들어감. 모든 레이어의 출력이 각 레이어의 출력과 concat됨. Dense connection이 vanishing gradient 문제를 해결할 수 있다고 주장. Dense connection이 Feature을 더 잘 전달하고 사용할 수 있게 해줌.