



Type something...

01) CNN Architectures

01_Case Study

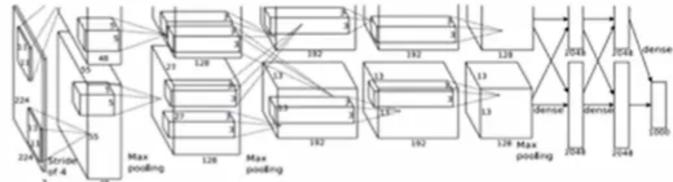
1. AlexNet

Case Study: AlexNet

[Krizhevsky et al. 2012]

Full (simplified) AlexNet architecture:

[227x227x3] INPUT
 [55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
 [27x27x96] MAX POOL1: 3x3 filters at stride 2
 [27x27x96] NORM1: Normalization layer
 [27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
 [13x13x256] MAX POOL2: 3x3 filters at stride 2
 [13x13x256] NORM2: Normalization layer
 [13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
 [13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
 [13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
 [6x6x256] MAX POOL3: 3x3 filters at stride 2
 [4096] FC6: 4096 neurons
 [4096] FC7: 4096 neurons
 [1000] FC8: 1000 neurons (class scores)



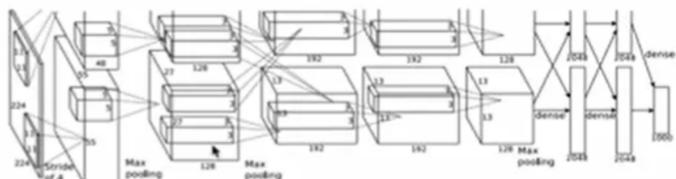
- 이전 딥러닝 실습 강의에서 구현한 AlexNet이다
- Conv1 → max1 → norm1
- conv2 → max2 → norm2
- conv3 → conv4 → conv5
- max3
- FC6 → FC7 → FC8
- norm은 지금은 사용하지 않는다.

Case Study: AlexNet

[Krizhevsky et al. 2012]

Full (simplified) AlexNet architecture:

[227x227x3] INPUT
 [55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
 [27x27x96] MAX POOL1: 3x3 filters at stride 2
 [27x27x96] NORM1: Normalization layer
 [27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
 [13x13x256] MAX POOL2: 3x3 filters at stride 2
 [13x13x256] NORM2: Normalization layer
 [13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
 [13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
 [13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
 [6x6x256] MAX POOL3: 3x3 filters at stride 2
 [4096] FC6: 4096 neurons
 [4096] FC7: 4096 neurons
 [1000] FC8: 1000 neurons (class scores)



Details/Retrospectives:

- first use of ReLU
- used Norm layers (not common anymore)
- heavy data augmentation
- dropout 0.5
- batch size 128
- SGD Momentum 0.9
- Learning rate 1e-2, reduced by 10 manually when val accuracy plateaus
- L2 weight decay 5e-4
- 7 CNN ensemble: 18.2% -> 15.4%

- 최초로 ReLU 사용
- dropout = 0.5
- batch size = 128



- SGD momentum = 0.9
- ensemble 18.2% → 15.4%

2. VGGNet

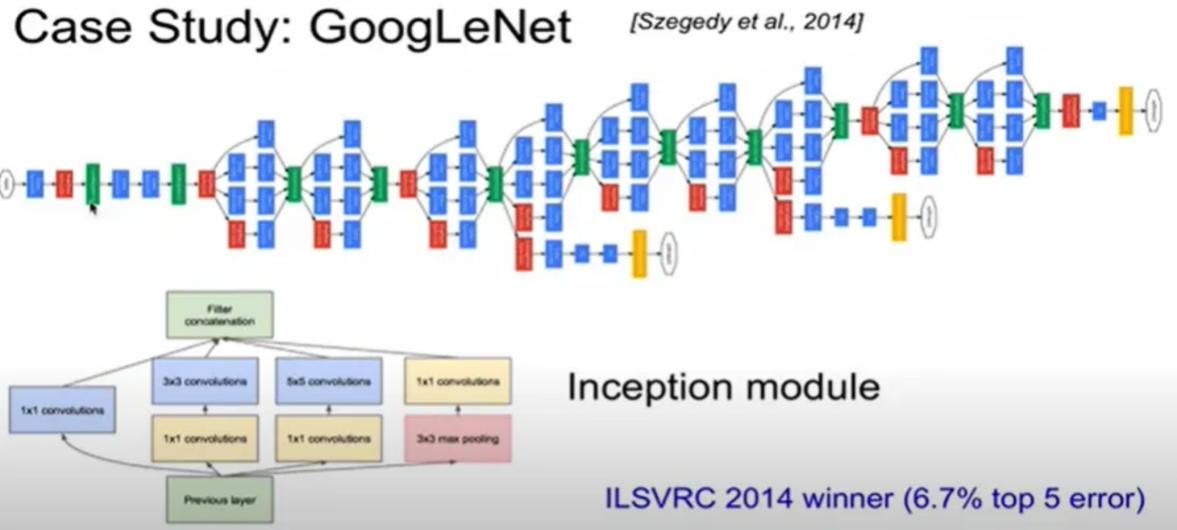
- 3x3 conv만 사용한 모델이다.

INPUT: [224x224x3] memory: 224*224*3=150K params: 0 (not counting biases)
 CONV3-64: [224x224x64] memory: 224*224*64=3.2M params: (3*3*3)*64 = 1,728
 CONV3-64: [224x224x64] memory: 224*224*64=3.2M params: (3*3*64)*64 = 36,864
 POOL2: [112x112x64] memory: 112*112*64=800K params: 0
 CONV3-128: [112x112x128] memory: 112*112*128=1.6M params: (3*3*64)*128 = 73,728
 CONV3-128: [112x112x128] memory: 112*112*128=1.6M params: (3*3*128)*128 = 147,456
 POOL2: [56x56x128] memory: 56*56*128=400K params: 0
 CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*128)*256 = 294,912
 CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*256)*256 = 589,824
 CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*256)*256 = 589,824
 POOL2: [28x28x256] memory: 28*28*256=200K params: 0
 CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*256)*512 = 1,179,648
 CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*512)*512 = 2,359,296
 CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*512)*512 = 2,359,296
 POOL2: [14x14x512] memory: 14*14*512=100K params: 0
 CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
 CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
 CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
 POOL2: [7x7x512] memory: 7*7*512=25K params: 0
 FC: [1x1x4096] memory: 4096 params: 77*512*4096 = 102,760,448
 FC: [1x1x4096] memory: 4096 params: 4096*4096 = 16,777,216
 FC: [1x1x1000] memory: 1000 params: 4096*1000 = 4,096,000

B	C	D
13 weight layers	16 weight layers	16 weight layers
put (224 × 224 RGB image)		
conv3-64	conv3-64	conv3-64
conv3-64	conv3-64	conv3-64
maxpool		
conv3-128	conv3-128	conv3-128
conv3-128	conv3-128	conv3-128
maxpool		
conv3-256	conv3-256	conv3-256
conv3-256	conv3-256	conv3-256
conv1-256	conv1-256	conv1-256
maxpool		
conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512
conv1-512	conv1-512	conv1-512
maxpool		
conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512
conv1-512	conv1-512	conv1-512
maxpool		
FC-4096		
FC-4096		
FC-1000		
soft-max		

- 이미지 1장당 200mb 용량을 사용한다.
- AVG pooling을 FC 대신해서 사용할 수 있다.

3. GoogLeNet

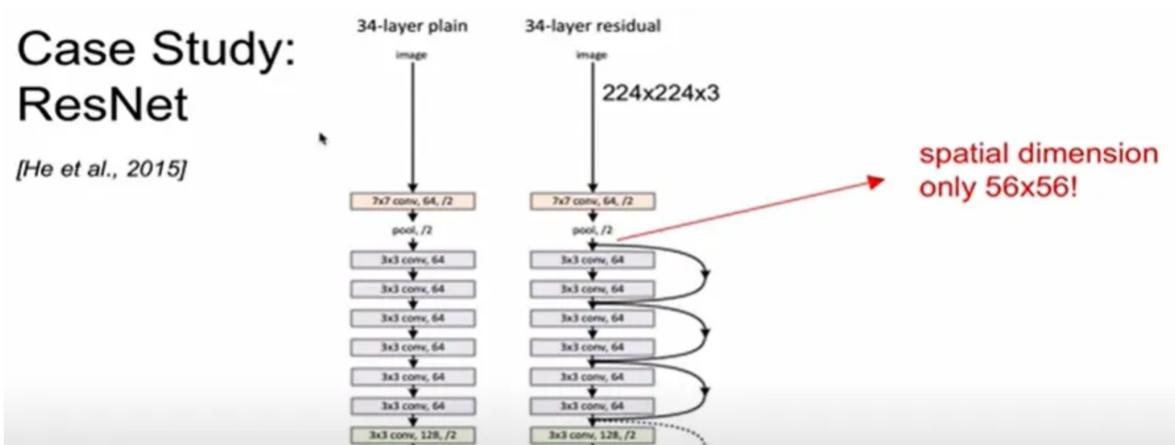


- Inception이라는 module을 반복적으로 사용한 모델이다.

4. ResNet

Case Study: ResNet

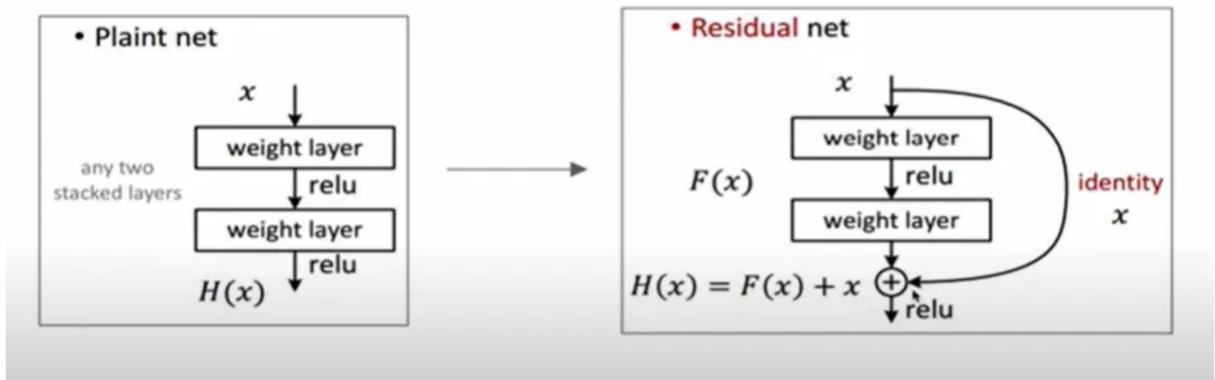
[He et al., 2015]





- 초기 사이즈를 줄여서 효율적인 연산이 가능함.

Case Study: ResNet [He et al., 2015]



- residual net을 활용하였다.

Case Study: ResNet [He et al., 2015]

- 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

- He initialization 사용
- Learning rate를 10씩 계속 나눠가면서 사용
- batch normalization을 사용해서 dropout을 사용하지 않았다.

