

Lecture 6: CNN Architectures

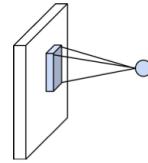
Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 1

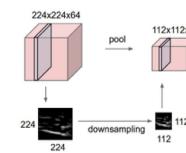
April 20, 2023

Components of CNNs

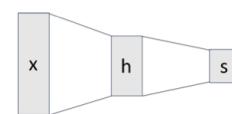
Convolution Layers



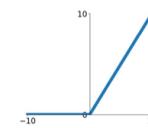
Pooling Layers



Fully-Connected Layers



Activation Function



Normalization

$$\hat{x}_{i,j} = \frac{x_{i,j} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

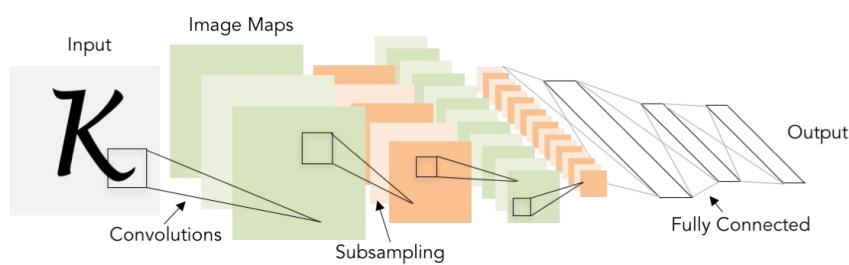
Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 3

April 20, 2023

Review: LeNet-5

[LeCun et al., 1998]



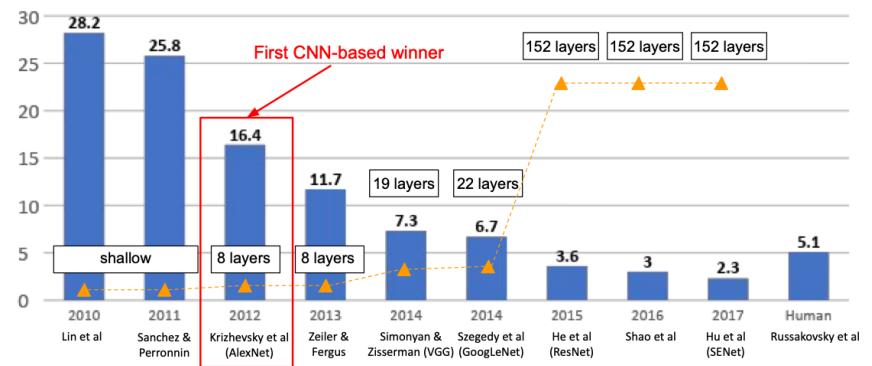
Conv filters were 5x5, applied at stride 1
Subsampling (Pooling) layers were 2x2 applied at stride 2
i.e. architecture is [CONV-POOL-CONV-POOL-FC-FC]

Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 20

April 20, 2023

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 26

April 20, 2023

Case Study: AlexNet

[Krizhevsky et al. 2012]

Architecture:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

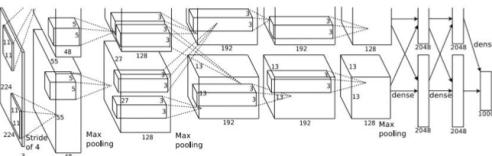


Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 27

April 20, 2023

Case Study: AlexNet

[Krizhevsky et al. 2012]

Input: 227x227x3 images

First layer (CONV1): 96 11x11 filters applied at stride 4

=>

Output volume [55x55x96]

$$W' = (W - F + 2P) / S + 1$$

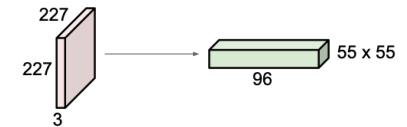


Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

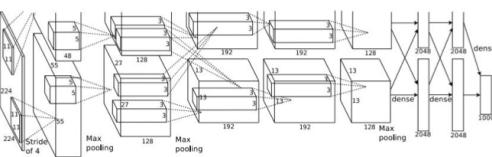
Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 29

April 20, 2023

Case Study: AlexNet

[Krizhevsky et al. 2012]



Input: 227x227x3 images

First layer (CONV1): 96 11x11 filters applied at stride 4

=>

Output volume [55x55x96]

Parameters: $(11 \times 11 \times 3 + 1) \times 96 = 35K$

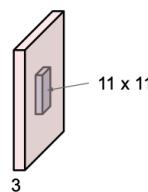


Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 31

April 20, 2023

Case Study: AlexNet

[Krizhevsky et al. 2012]

Input: 227x227x3 images

After CONV1: 55x55x96

Second layer (POOL1): 3x3 filters applied at stride 2

Output volume: 27x27x96

Parameters: 0!

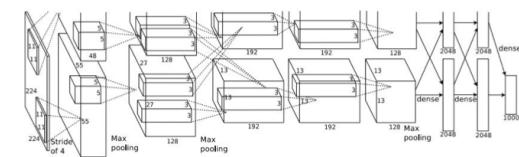


Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

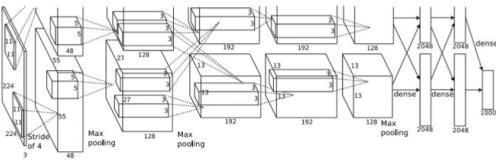
Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 34

April 20, 2023

Case Study: AlexNet

[Krizhevsky et al. 2012]



Input: 227x227x3 images

After CONV1: 55x55x96

After POOL1: 27x27x96

...

Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 35

April 20, 2023

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 LRN 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%

Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

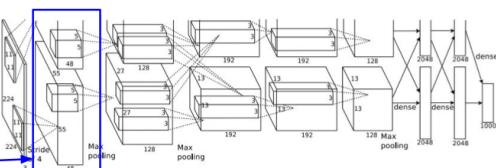
Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 37

April 20, 2023

Case Study: AlexNet

[Krizhevsky et al. 2012]



Full (simplified) AlexNet architecture:

[227x227x3] INPUT

[55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0

[55x55x48] x 2

[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)

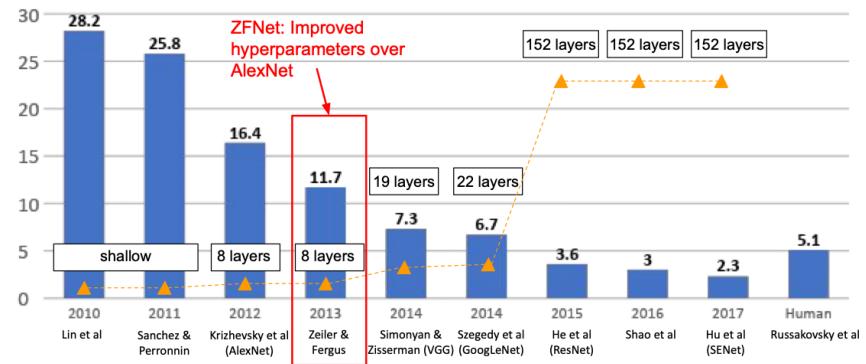
Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 38

April 20, 2023

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

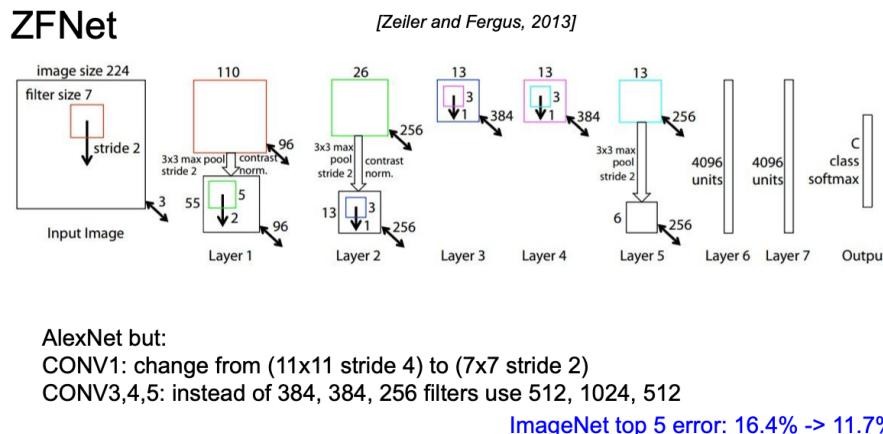


Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 42

April 20, 2023

ZFNet

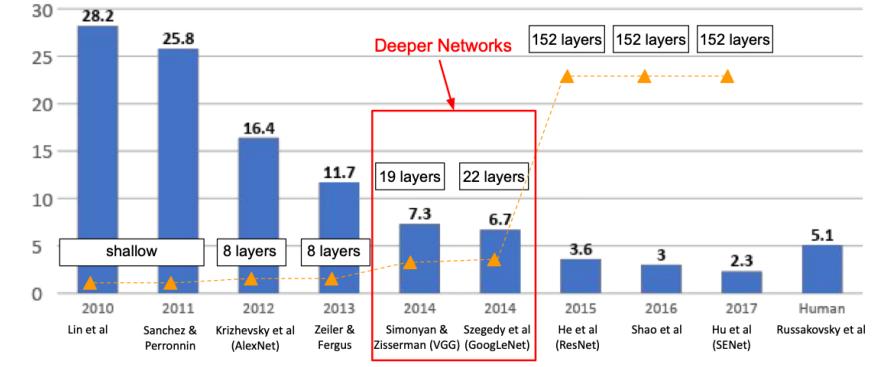


Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 43

April 20, 2023

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 44

April 20, 2023

Case Study: VGGNet

[Simonyan and Zisserman, 2014]

Small filters, Deeper networks

8 layers (AlexNet)

-> 16 - 19 layers (VGG16Net)

Only 3x3 CONV stride 1, pad 1 and 2x2 MAX POOL stride 2

11.7% top 5 error in ILSVRC'13 (ZFNet)

-> 7.3% top 5 error in ILSVRC'14



Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 45

April 20, 2023

Case Study: VGGNet

[Simonyan and Zisserman, 2014]

Q: Why use smaller filters? (3x3 conv)

Stack of three 3x3 conv (stride 1) layers has same effective receptive field as one 7x7 conv layer

Q: What is the effective receptive field of three 3x3 conv (stride 1) layers?



Fei-Fei Li, Yunzhu Li, Ruohan Gao

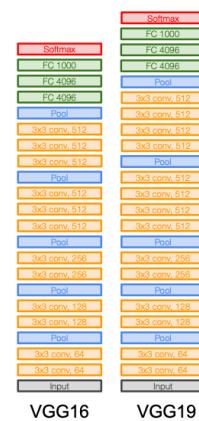
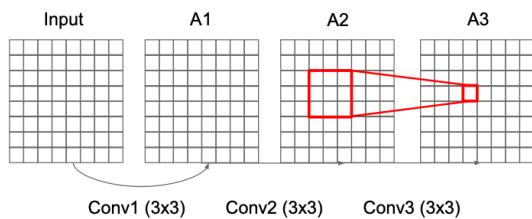
Lecture 6 - 47

April 20, 2023

Case Study: VGGNet

[Simonyan and Zisserman, 2014]

Q: What is the effective receptive field of three 3x3 conv (stride 1) layers?



Fei-Fei Li, Yunzhu Li, Ruohan Gao

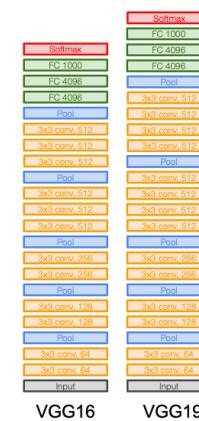
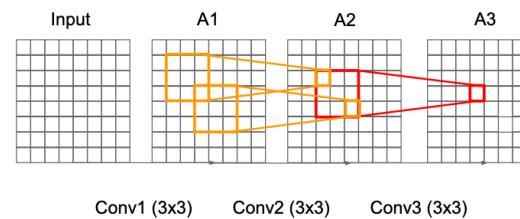
Lecture 6 - 48

April 20, 2023

Case Study: VGGNet

[Simonyan and Zisserman, 2014]

Q: What is the effective receptive field of three 3x3 conv (stride 1) layers?



Fei-Fei Li, Yunzhu Li, Ruohan Gao

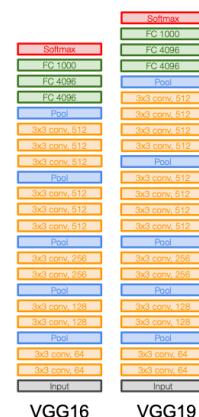
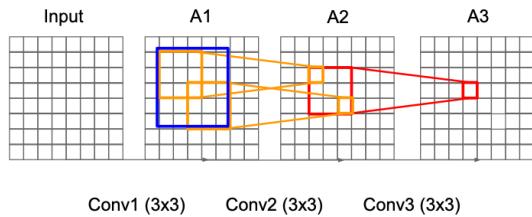
Lecture 6 - 49

April 20, 2023

Case Study: VGGNet

[Simonyan and Zisserman, 2014]

Q: What is the effective receptive field of three 3x3 conv (stride 1) layers?



Fei-Fei Li, Yunzhu Li, Ruohan Gao

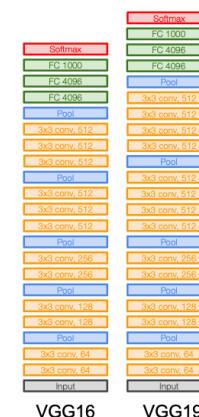
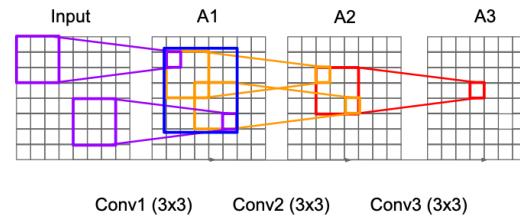
Lecture 6 - 50

April 20, 2023

Case Study: VGGNet

[Simonyan and Zisserman, 2014]

Q: What is the effective receptive field of three 3x3 conv (stride 1) layers?



Fei-Fei Li, Yunzhu Li, Ruohan Gao

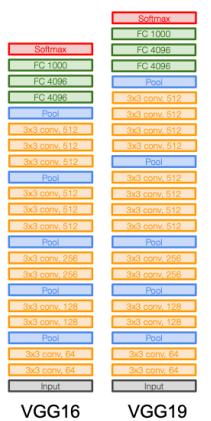
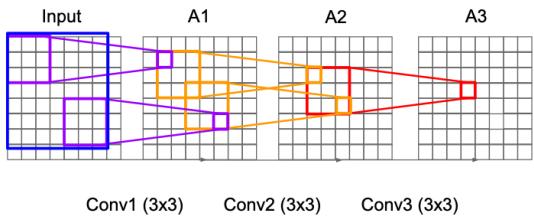
Lecture 6 - 51

April 20, 2023

Case Study: VGGNet

[Simonyan and Zisserman, 2014]

Q: What is the effective receptive field of three 3x3 conv (stride 1) layers?



Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 52

April 20, 2023

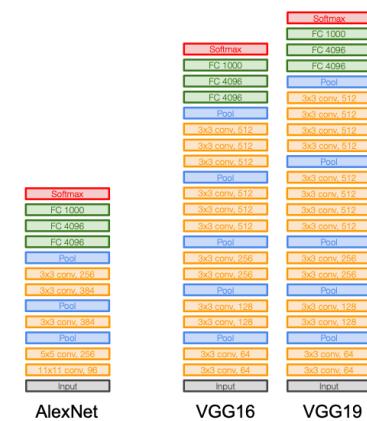
Case Study: VGGNet

[Simonyan and Zisserman, 2014]

Q: Why use smaller filters? (3x3 conv)

Stack of three 3x3 conv (stride 1) layers
has same **effective receptive field** as
one 7x7 conv layer

[7x7]



Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 53

April 20, 2023

Case Study: VGGNet

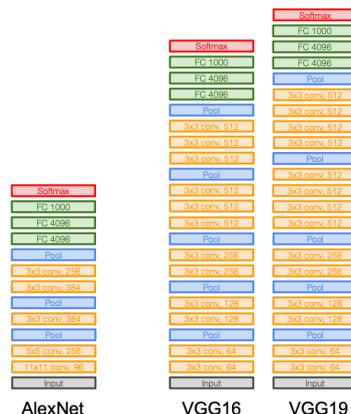
[Simonyan and Zisserman, 2014]

Q: Why use smaller filters? (3x3 conv)

Stack of three 3x3 conv (stride 1) layers
has same **effective receptive field** as
one 7x7 conv layer

But deeper, more non-linearities

And fewer parameters: $3 * (3^2 C^2)$ vs.
 $7^2 C^2$ for C channels per layer

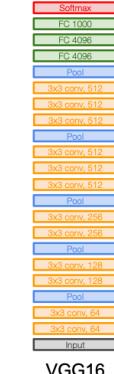


Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 54

April 20, 2023

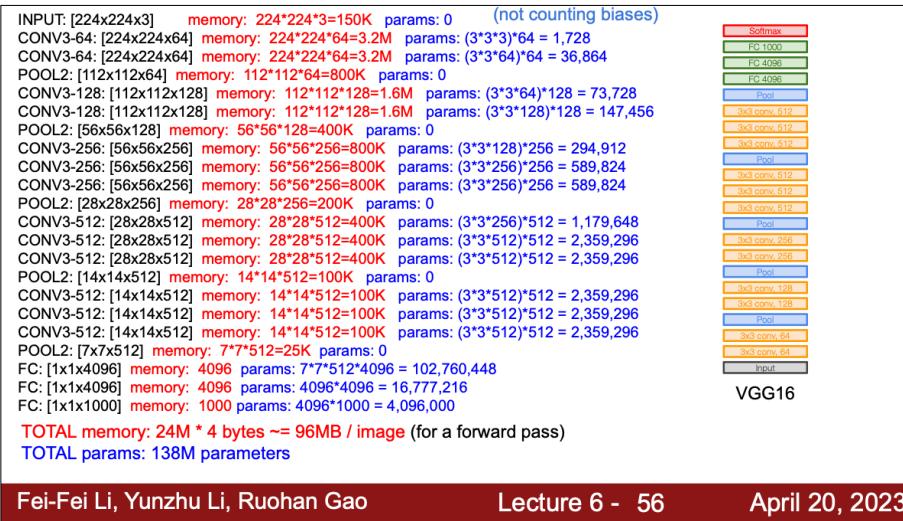
INPUT: [224x224x3] memory: 224*224*3=150K params: 0 (not counting biases)
 CONV3-64: [224x224x64] memory: 224*224*64=3.2M params: (3*3*3)*64 = 1,728
 CONV3-64: [224x224x64] memory: 224*224*64=3.2M params: (3*3*64)*64 = 36,864
 POOL2: [112x112x64] memory: 112*112*64=800K params: 0
 CONV3-128: [112x112x128] 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: 77*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



Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 55

April 20, 2023



Fei-Fei Li, Yunzhu Li, Ruohan Gao

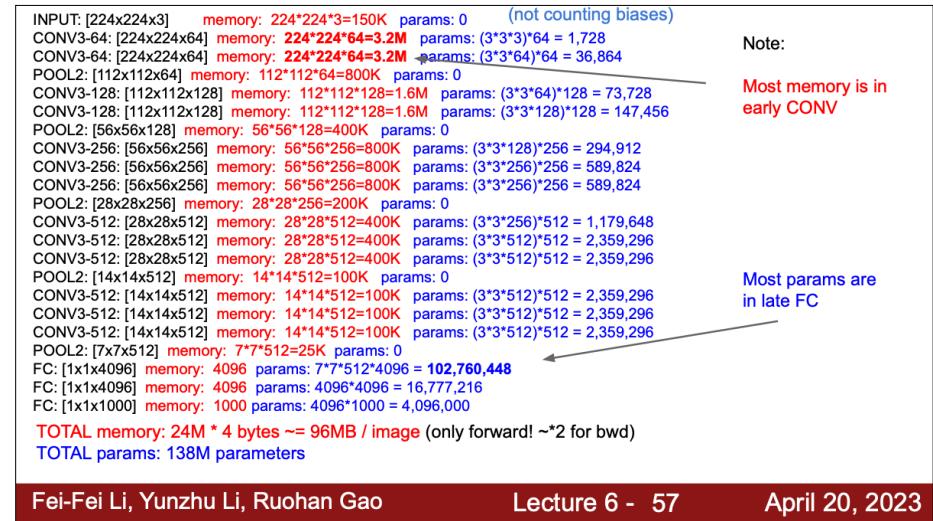
Lecture 6 - 56

April 20, 2023

Note:

Most memory is in early CONV

Most params are in late FC



Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 57

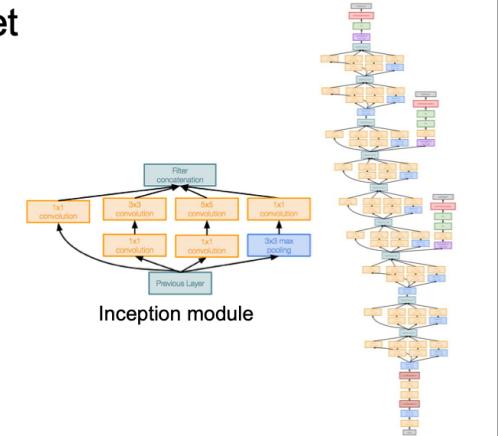
April 20, 2023

Case Study: GoogLeNet

[Szegedy et al., 2014]

Deeper networks, with computational efficiency

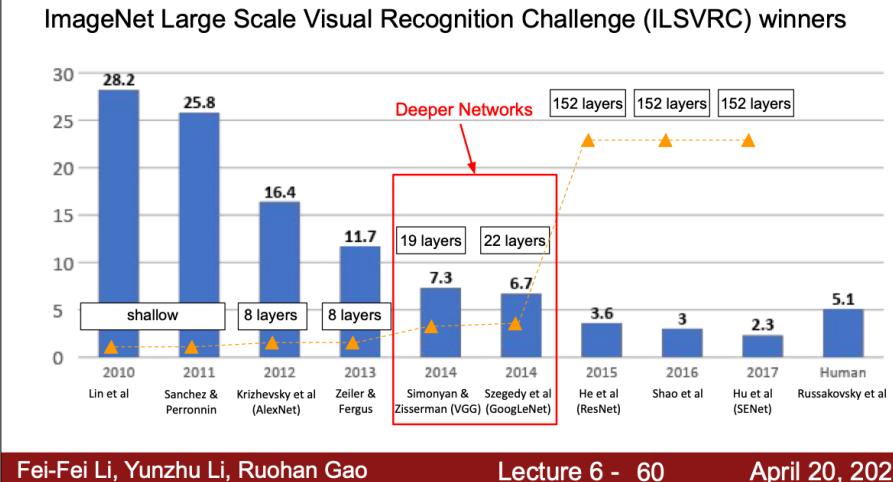
- ILSVRC'14 classification winner (6.7% top 5 error)
- 22 layers
- Only 5 million parameters!
- 12x less than AlexNet
- 27x less than VGG-16
- Efficient "Inception" module
- No FC layers



Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 61

April 20, 2023



Fei-Fei Li, Yunzhu Li, Ruohan Gao

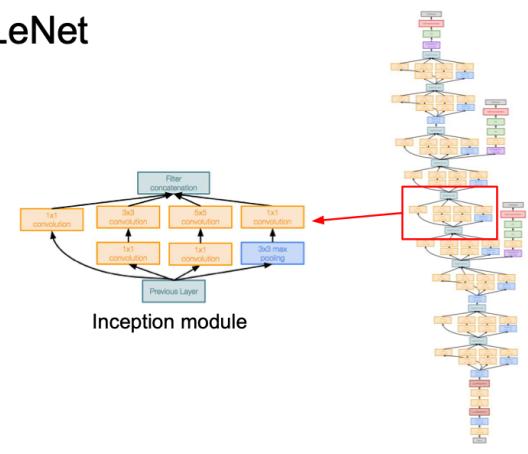
Lecture 6 - 60

April 20, 2023

Case Study: GoogLeNet

[Szegedy et al., 2014]

"Inception module": design a good local network topology (network within a network) and then stack these modules on top of each other



Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 62

April 20, 2023

Case Study: GoogLeNet

[Szegedy et al., 2014]

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 channel-wise

Q: What is the problem with this?
[Hint: Computational complexity]

Fei-Fei Li, Yunzhu Li, Ruohan Gao

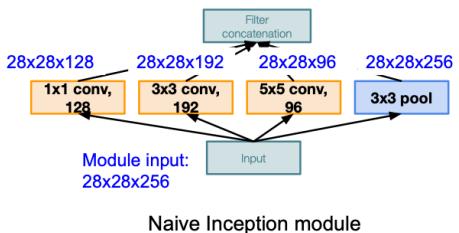
Lecture 6 - 64

April 20, 2023

Case Study: GoogLeNet

[Szegedy et al., 2014]

Example: Q1: What are the output sizes of all different filter operations?



Fei-Fei Li, Yunzhu Li, Ruohan Gao

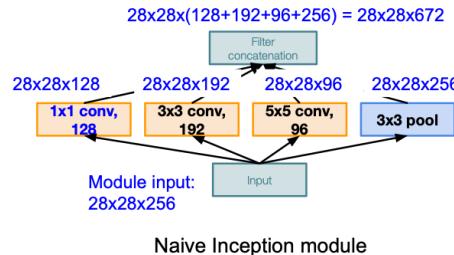
Lecture 6 - 67

April 20, 2023

Case Study: GoogLeNet

[Szegedy et al., 2014]

Example: Q2: What is output size after filter concatenation?



Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 69

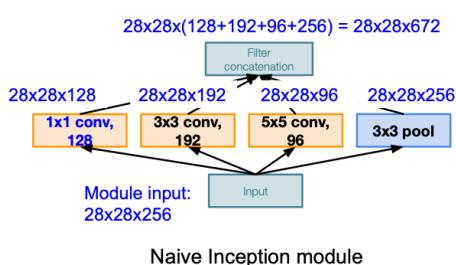
April 20, 2023

Case Study: GoogLeNet

[Szegedy et al., 2014]

Example:

Q2: What is output size after filter concatenation?



Q: What is the problem with this?
[Hint: Computational complexity]

Conv Ops:

[1x1 conv, 128] 28x28x128x1x1x256
[3x3 conv, 192] 28x28x192x3x3x256
[5x5 conv, 96] 28x28x96x5x5x256

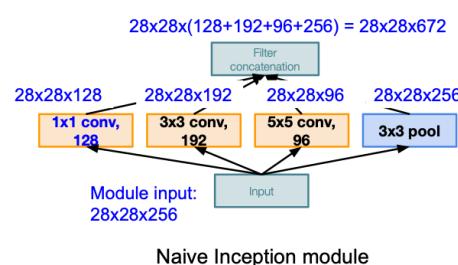
Total: 854M ops

Case Study: GoogLeNet

[Szegedy et al., 2014]

Example:

Q2: What is output size after filter concatenation?



Conv Ops:

Q: What is the problem with this?
[Hint: Computational complexity]

Conv Ops:

[1x1 conv, 128] 28x28x128x1x1x256
[3x3 conv, 192] 28x28x192x3x3x256
[5x5 conv, 96] 28x28x96x5x5x256

Total: 854M ops

Very expensive compute

Pooling layer also preserves feature depth, which means total depth after concatenation can only grow at every layer!

Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 70

April 20, 2023

Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 71

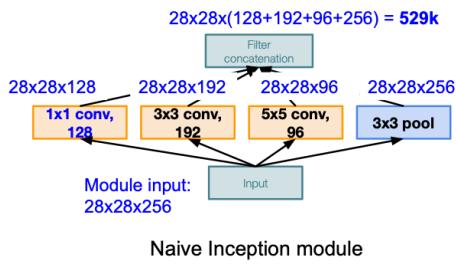
April 20, 2023

Case Study: GoogLeNet

[Szegedy et al., 2014]

Example:

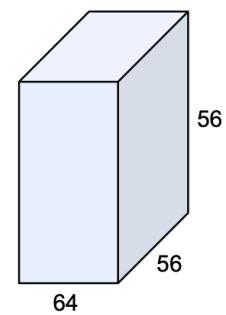
Q2: What is output size after filter concatenation?



Q: What is the problem with this?
[Hint: Computational complexity]

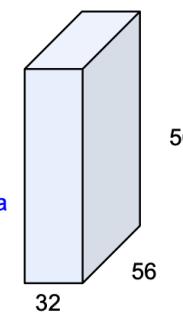
Solution: “bottleneck” layers that use 1x1 convolutions to reduce feature channel size

Review: 1x1 convolutions



1x1 CONV
with 32 filters

(each filter has size
1x1x64, and performs a
64-dimensional dot product)



Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 72

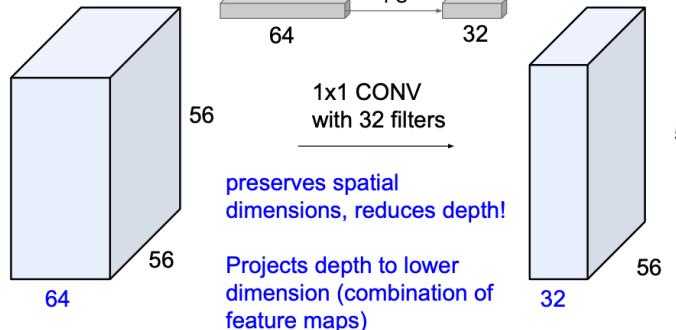
April 20, 2023

Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 73

April 20, 2023

Review: 1x1 convolutions



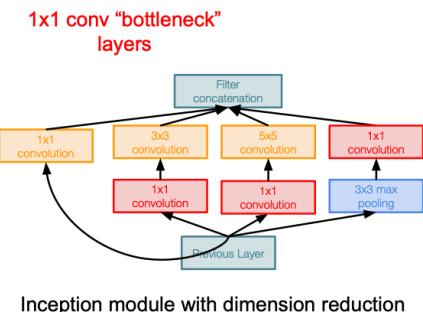
Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 75

April 20, 2023

Case Study: GoogLeNet

[Szegedy et al., 2014]



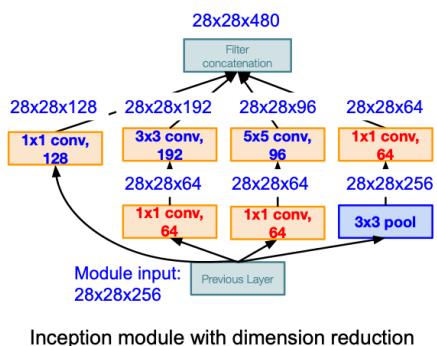
Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 77

April 20, 2023

Case Study: GoogLeNet

[Szegedy et al., 2014]



Fei-Fei Li, Yunzhu Li, Ruohan Gao

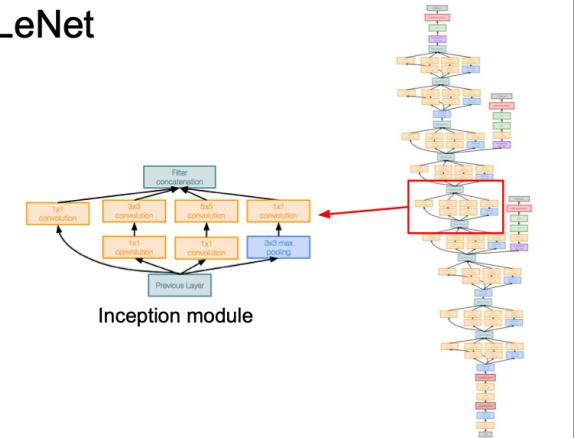
Lecture 6 - 78

April 20, 2023

Case Study: GoogLeNet

[Szegedy et al., 2014]

Stack Inception modules with dimension reduction on top of each other



Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 79

April 20, 2023

Case Study: GoogLeNet

[Szegedy et al., 2014]

Full GoogLeNet
architecture

Stem Network:
Conv-Pool-
2x Conv-Pool

Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 80

April 20, 2023

Case Study: GoogLeNet

[Szegedy et al., 2014]

Full GoogLeNet
architecture

Stacked Inception
Modules

Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 81

April 20, 2023

Case Study: GoogLeNet

[Szegedy et al., 2014]

Full GoogLeNet
architecture

Classifier output

Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 82

April 20, 2023

Case Study: GoogLeNet

[Szegedy et al., 2014]

Full GoogLeNet
architecture

Note: after the last convolutional layer, a global average pooling layer is used that spatially averages across each feature map, before final FC layer. No longer multiple expensive FC layers!



Classifier output

Fei-Fei Li, Yunzhu Li, Ruohan Gao

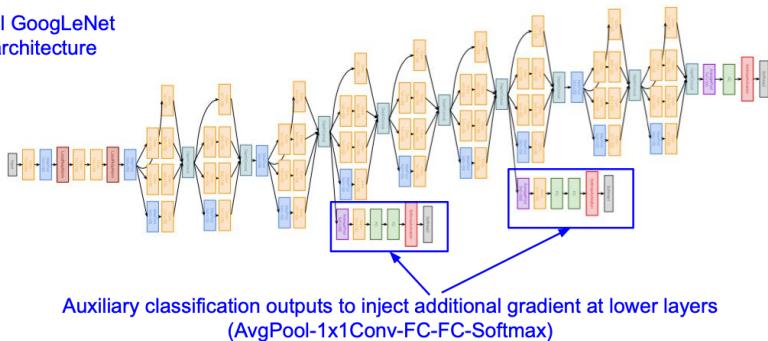
Lecture 6 - 83

April 20, 2023

Case Study: GoogLeNet

[Szegedy et al., 2014]

Full GoogLeNet architecture



Fei-Fei Li, Yunzhu Li, Ruohan Gao

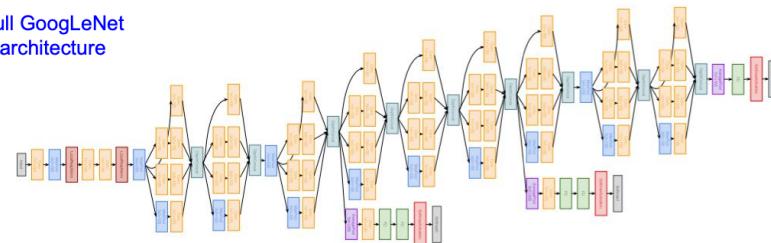
Lecture 6 - 84

April 20, 2023

Case Study: GoogLeNet

[Szegedy et al., 2014]

Full GoogLeNet architecture



22 total layers with weights
(parallel layers count as 1 layer => 2 layers per Inception module. Don't count auxiliary output layers)

Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 85

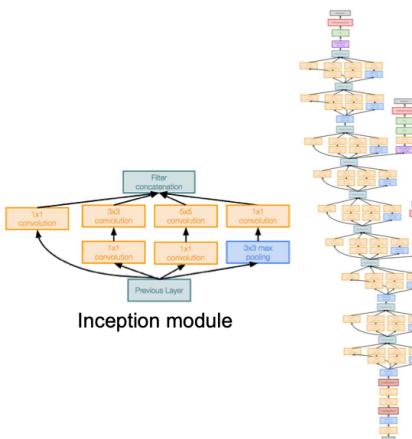
April 20, 2023

Case Study: GoogLeNet

[Szegedy et al., 2014]

Deeper networks, with computational efficiency

- 22 layers
- Efficient "Inception" module
- Avoids expensive FC layers
- 12x less params than AlexNet
- 27x less params than VGG-16
- ILSVRC'14 classification winner (6.7% top 5 error)



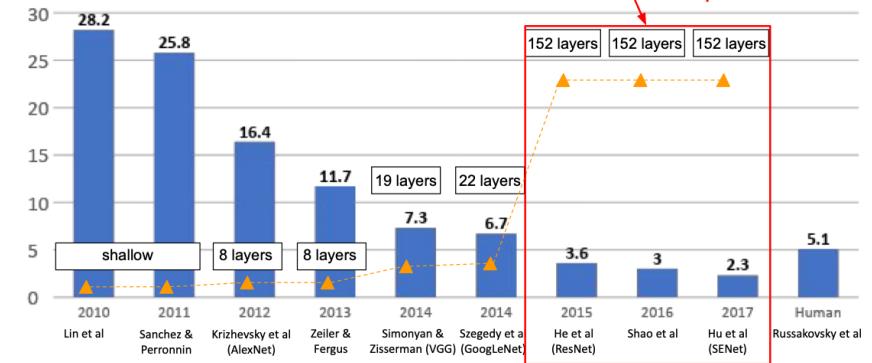
Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 86

April 20, 2023

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

"Revolution of Depth"



Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 87

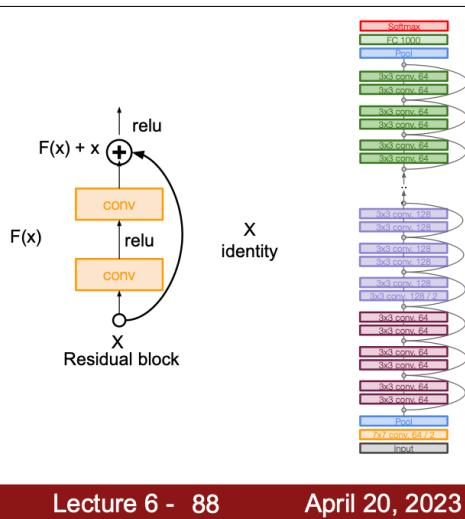
April 20, 2023

Case Study: ResNet

[He et al., 2015]

Very deep networks using residual connections

- 152-layer model for ImageNet
- ILSVRC'15 classification winner (3.57% top 5 error)
- Swept all classification and detection competitions in ILSVRC'15 and COCO'15!



Fei-Fei Li, Yunzhu Li, Ruohan Gao

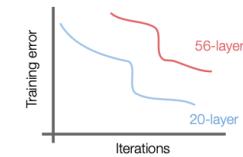
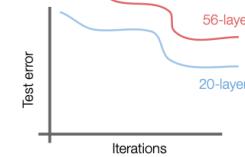
Lecture 6 - 88

April 20, 2023

Case Study: ResNet

[He et al., 2015]

What happens when we continue stacking deeper layers on a “plain” convolutional neural network?



56-layer model performs worse on both test and training error
-> The deeper model performs worse, but it's not caused by overfitting!

Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 91

April 20, 2023

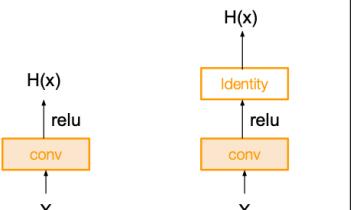
Case Study: ResNet

[He et al., 2015]

Fact: Deep models have more representation power (more parameters) than shallower models.

Hypothesis: the problem is an *optimization* problem, deeper models are harder to optimize

What should the deeper model learn to be at least as good as the shallower model?

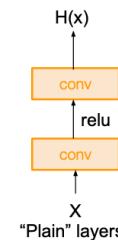


A solution by construction is copying the learned layers from the shallower model and setting additional layers to identity mapping.

Case Study: ResNet

[He et al., 2015]

Solution: Use network layers to fit a residual mapping instead of directly trying to fit a desired underlying mapping



Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 93

April 20, 2023

Fei-Fei Li, Yunzhu Li, Ruohan Gao

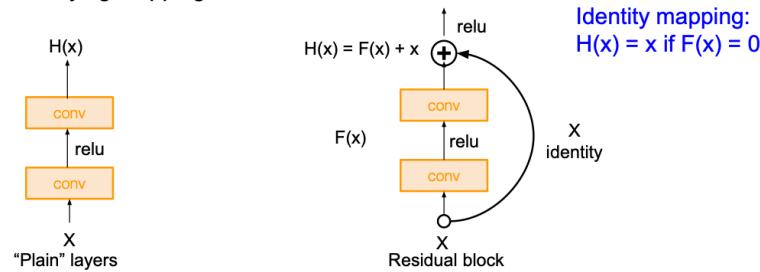
Lecture 6 - 94

April 20, 2023

Case Study: ResNet

[He et al., 2015]

Solution: Use network layers to fit a residual mapping instead of directly trying to fit a desired underlying mapping



Fei-Fei Li, Yunzhu Li, Ruohan Gao

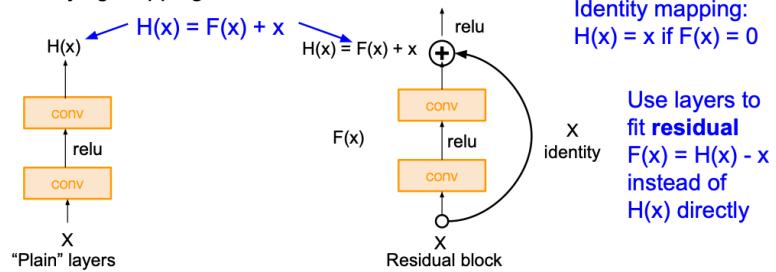
Lecture 6 - 95

April 20, 2023

Case Study: ResNet

[He et al., 2015]

Solution: Use network layers to fit a residual mapping instead of directly trying to fit a desired underlying mapping



Fei-Fei Li, Yunzhu Li, Ruohan Gao

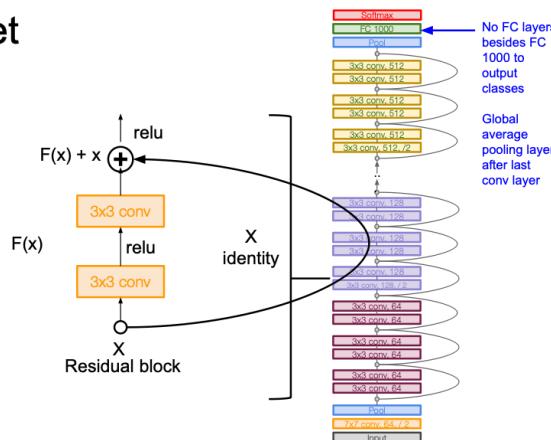
Lecture 6 - 96

April 20, 2023

Case Study: ResNet

[He et al., 2015]

- Full ResNet architecture:
- Stack residual blocks
- Every residual block has two 3x3 conv layers
- Periodically, double # of filters and downsample spatially using stride 2 (/2 in each dimension)
- Additional conv layer at the beginning (stem)
- No FC layers at the end (only FC 1000 to output classes)
- (In theory, you can train a ResNet with input image of variable sizes)



Fei-Fei Li, Yunzhu Li, Ruohan Gao

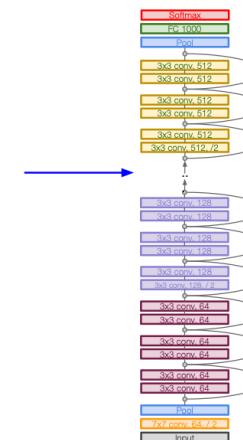
Lecture 6 - 100

April 20, 2023

Case Study: ResNet

[He et al., 2015]

Total depths of 18, 34, 50, 101, or 152 layers for ImageNet



Fei-Fei Li, Yunzhu Li, Ruohan Gao

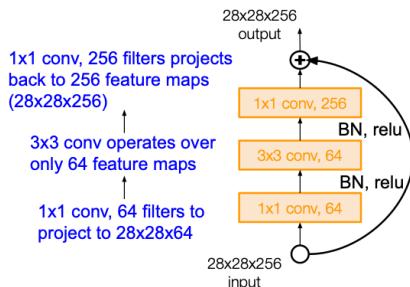
Lecture 6 - 101

April 20, 2023

Case Study: ResNet

[He et al., 2015]

For deeper networks (ResNet-50+), use "bottleneck" layer to improve efficiency (similar to GoogLeNet)



Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 103

April 20, 2023

Case Study: ResNet

[He et al., 2015]

Training ResNet in practice:

- Batch Normalization after every CONV layer
- Xavier 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

Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 104

April 20, 2023

Case Study: ResNet

[He et al., 2015]

Experimental Results

- Able to train very deep networks without degrading (152 layers on ImageNet, 1202 on Cifar)
- Deeper networks now achieve lower training error as expected
- Swept 1st place in all ILSVRC and COCO 2015 competitions

MSRA @ ILSVRC & COCO 2015 Competitions

• 1st places in all five main tracks

- ImageNet Classification: "Ultra-deep" (quote Yann) 152-layer nets
- ImageNet Detection: 16% better than 2nd
- ImageNet Localization: 27% better than 2nd
- COCO Detection: 11% better than 2nd
- COCO Segmentation: 12% better than 2nd

ILSVRC 2015 classification winner (3.6% top 5 error) -- better than "human performance"! (Russakovsky 2014)

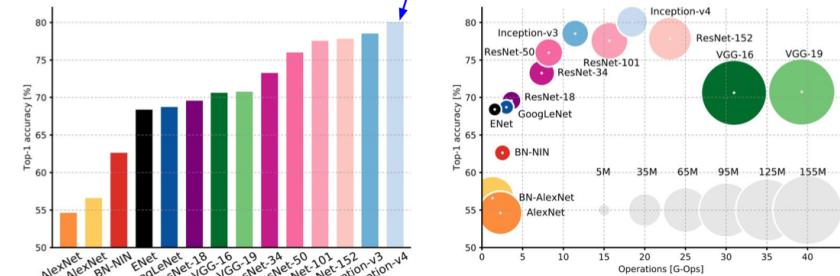
Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 106

April 20, 2023

Comparing complexity...

Inception-v4: Resnet + Inception!



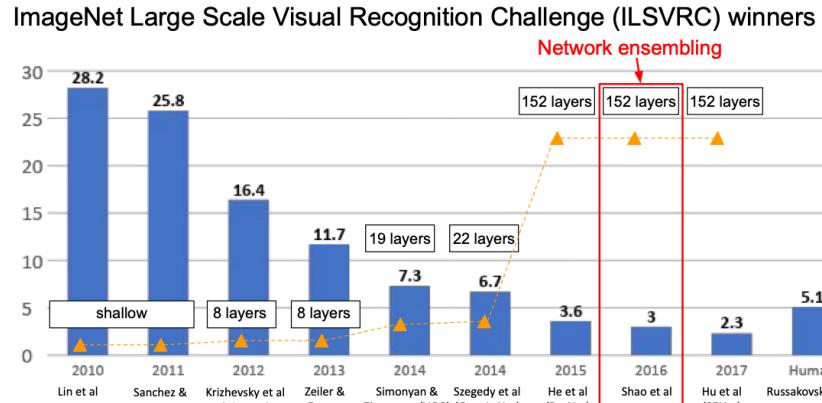
An Analysis of Deep Neural Network Models for Practical Applications, 2017.

Figures copyright Alfredo Canziani, Adam Paszke, Eugenio Culurciello, 2017. Reproduced with permission.

Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 108

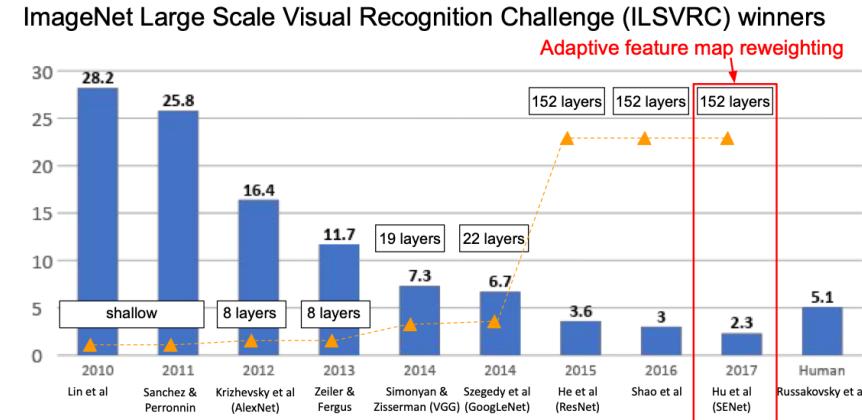
April 20, 2023



Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 113

April 20, 2023



Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 115

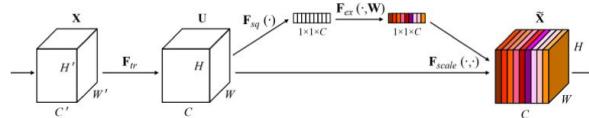
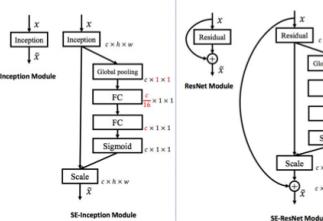
April 20, 2023

Improving ResNets...

Squeeze-and-Excitation Networks (SENet)

[Hu et al. 2017]

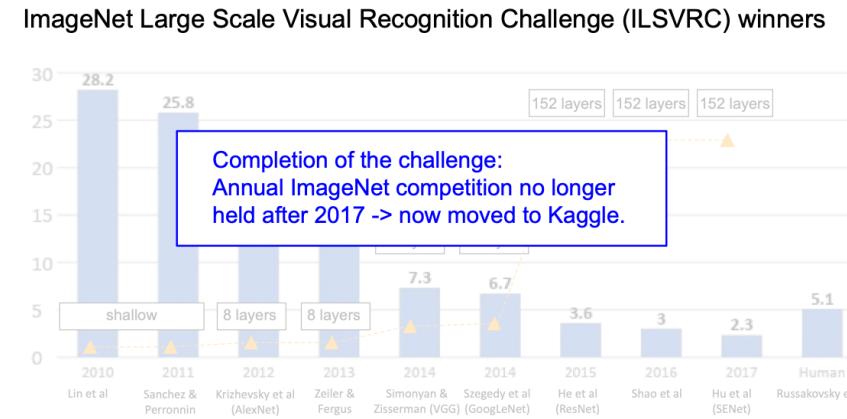
- Add a “feature recalibration” module that learns to adaptively reweight feature maps
- Global information (global avg. pooling layer) + 2 FC layers used to determine feature map weights
- ILSVRC’17 classification winner (using ResNeXt-152 as a base architecture)



Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 116

April 20, 2023



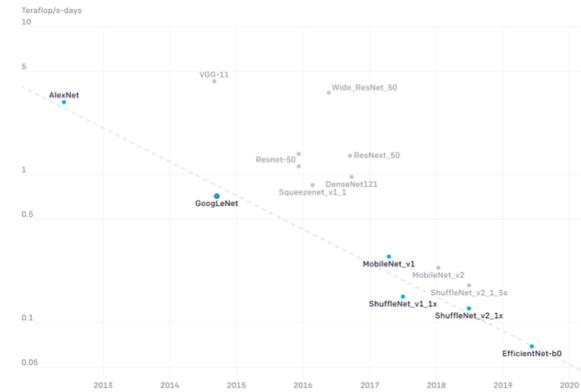
Fei-Fei Li, Yunzhu Li, Ruohan Gao

Lecture 6 - 118

April 20, 2023

But research into CNN architectures is still flourishing

Efficient networks...



Summary: CNN Architectures

Case Studies

- AlexNet
- VGG
- GoogLeNet
- ResNet

Also....

- SENet
- Wide ResNet
- ResNeXT
- DenseNet
- MobileNets
- NASNet