RECOD
reasoning for complex data

Machine Learning and Pattern Recognition
A High Level Overview

Prof. Anderson Rocha
(Main bulk of slides kindly provided by **Prof. Sandra Avila**
and based on materials by Fei-Fei Li & Justin Johnson & Serena Yeung)
Institute of Computing (IC/Unicamp)

MC886/MO444

DNNs Architectures

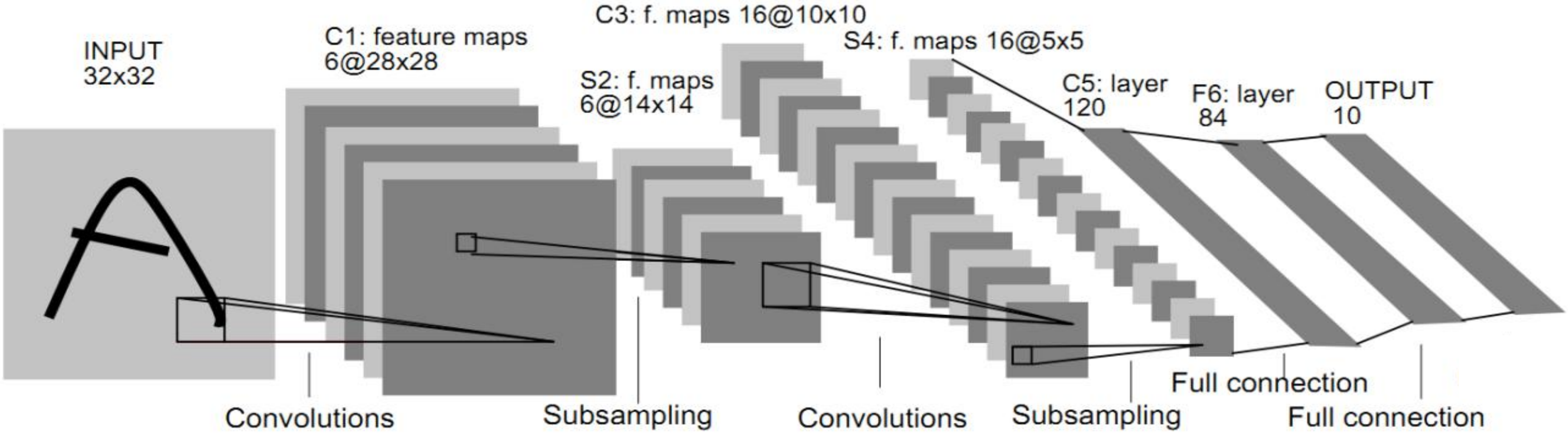
DNNs Architectures

- **LeNet** by Yann LeCun, Léon Bottou & Yoshua Bengio (1998)
- **AlexNet** by Alex Krizhevsky, Ilya Sutskever & Geoff Hinton (2012)
- **ZF Net** by Matthew Zeiler & Rob Fergus (2013)
- **VGGNet** by Karen Simonyan & Andrew Zisserman (2014)
- **GoogLeNet** by Szegedy et al. (2014)
- **ResNet** by Kaiming He et al. (2015)

DNNs Architectures

- **LeNet** by Yann LeCun, Léon Bottou & Yoshua Bengio (1998)
- **AlexNet** by Alex Krizhevsky, Ilya Sutskever & Geoff Hinton (2012)
- **ZF Net** by Matthew Zeiler & Rob Fergus (2013)
- **VGGNet** by Karen Simonyan & Andrew Zisserman (2014)
- **GoogLeNet** by Szegedy et al. (2014)
- **ResNet** by Kaiming He et al. (2015)

LeNet-5 [LeCun et al., 1998]



Convolution filters: 5x5 with stride 1

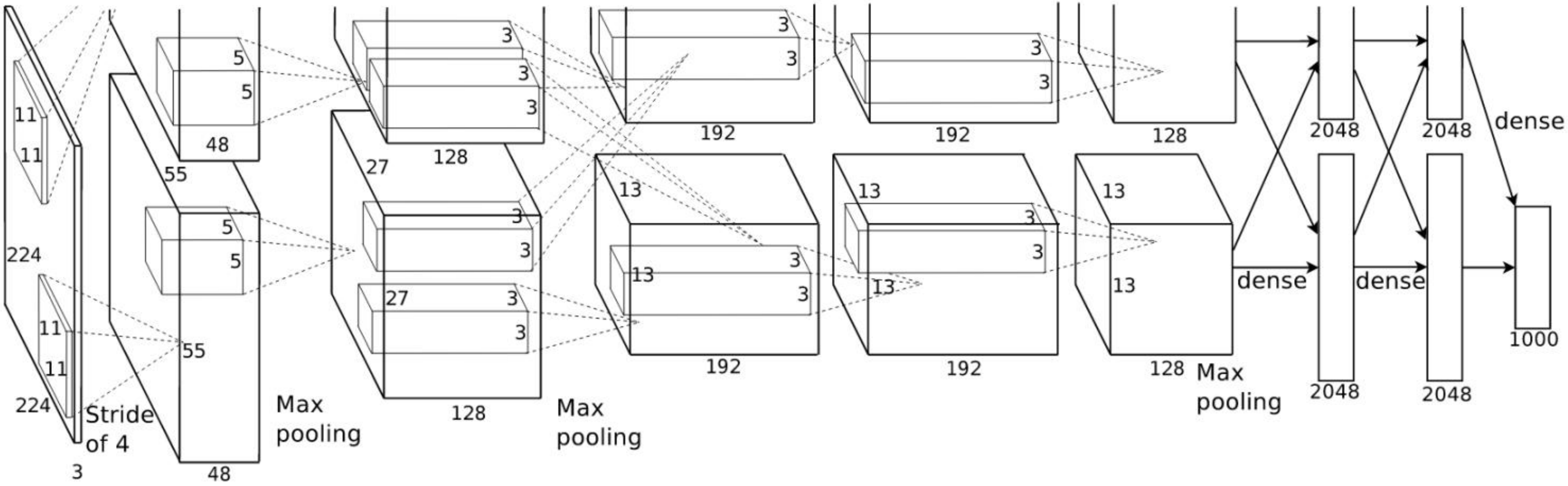
Subsampling (Pooling) layers: 2x2 with stride 2

[CONV-POOL-CONV-POOL-FC-FC]

DNNs Architectures

- **LeNet** by Yann LeCun, Léon Bottou & Yoshua Bengio (1998)
- **AlexNet** by Alex Krizhevsky, Ilya Sutskever & Geoff Hinton (2012)
- **ZF Net** by Matthew Zeiler & Rob Fergus (2013)
- **VGGNet** by Karen Simonyan & Andrew Zisserman (2014)
- **GoogLeNet** by Szegedy et al. (2014)
- **ResNet** by Kaiming He et al. (2015)

AlexNet [Krizhevsky et al., 2012]

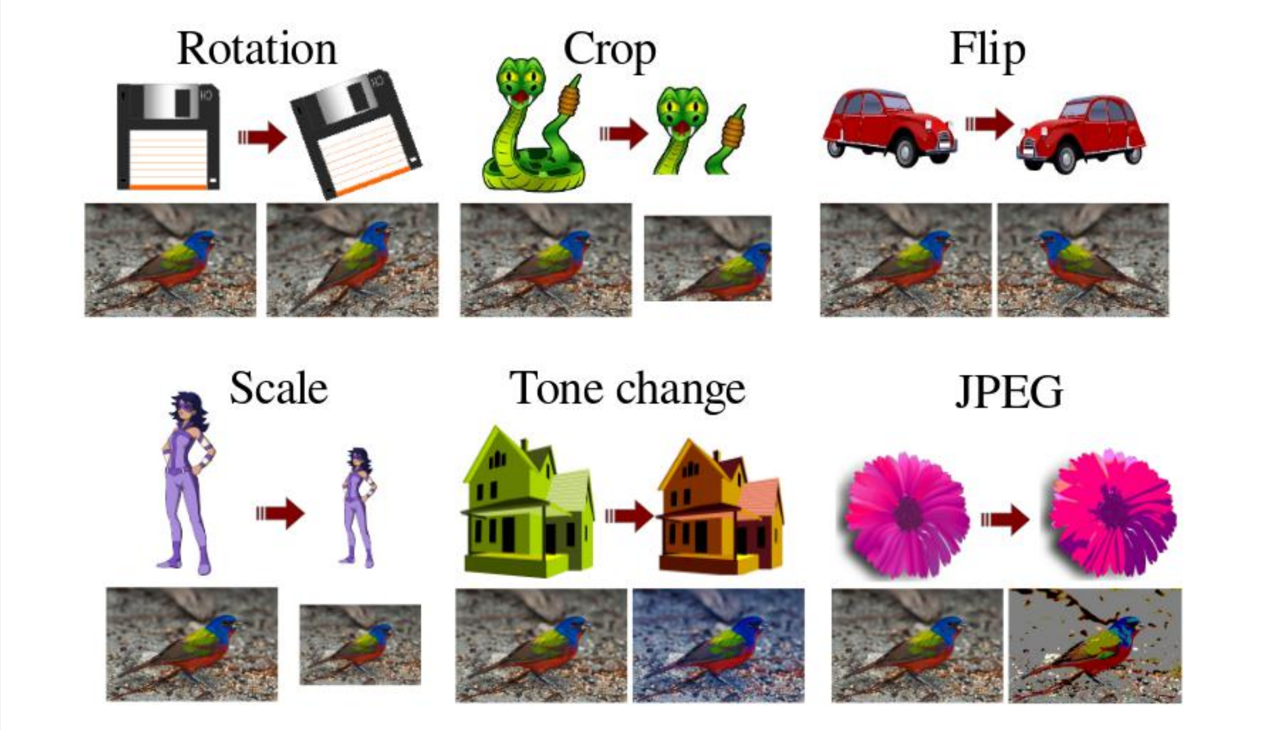


AlexNet [Krizhevsky et al., 2012]

Details:

- 60 million learned parameters
- first use of ReLU
- used Norm layers (not common anymore)
- heavy **data augmentation**
- dropout 0.5
- batch size 128
- 7 CNN ensemble: 18.2% -> 15.4%
- 5-6 days to train on 2 GTX 580 3GB GPUs

Data Augmentation



Credit: Plauin et al., Transformation Pursuit for Image Classification, CVPR 2014.

Simple to implement, use it

Especially useful for small datasets

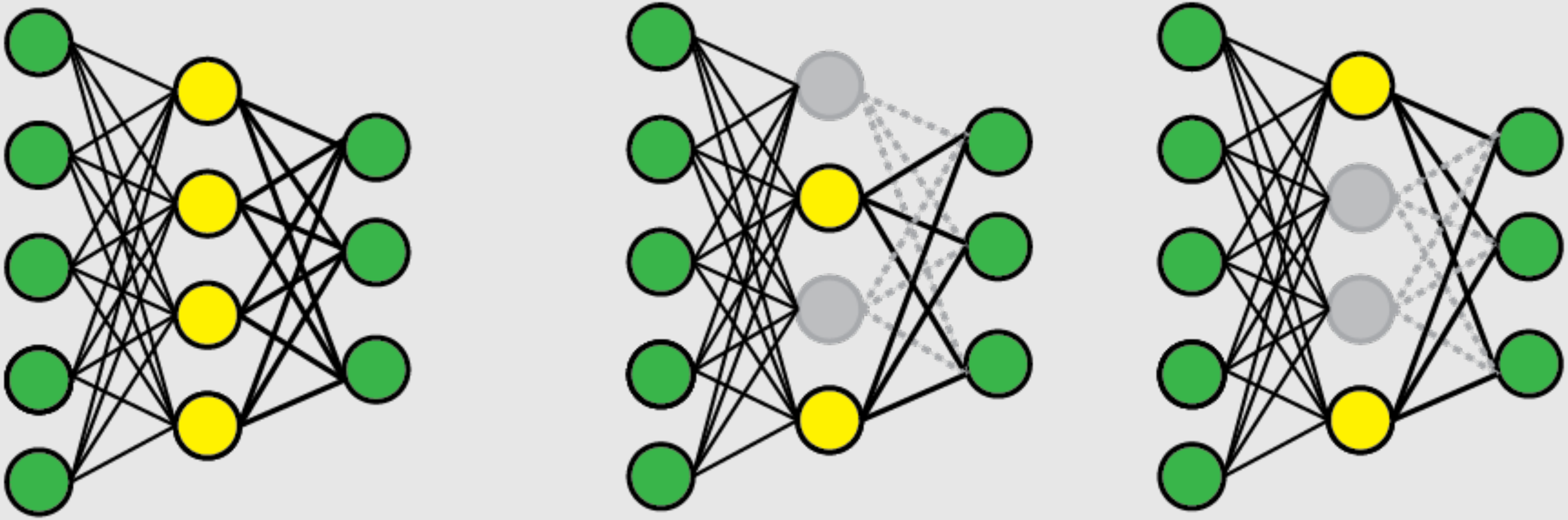
Apply on training and testing

AlexNet [Krizhevsky et al., 2012]

Details:

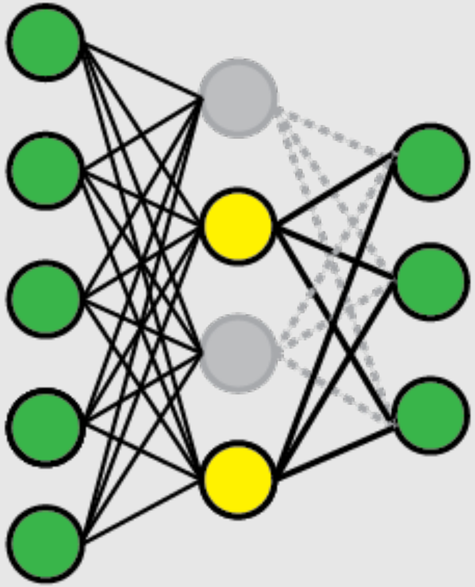
- 60 million learned parameters
- first use of ReLU
- used Norm layers (not common anymore)
- heavy data augmentation
- **dropout** 0.5
- batch size 128
- 7 CNN ensemble: 18.2% -> 15.4%

Dropout [Hinton et al., 2012]

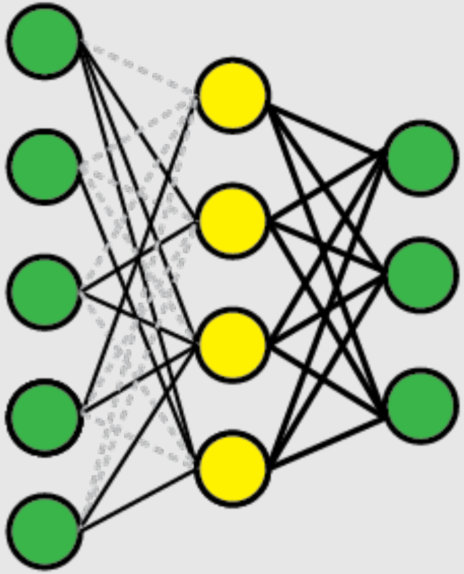


Standard Network After applying dropout

Dropout [Hinton et al., 2012] vs. DropConnect [Wan et al., 2013]



Dropout

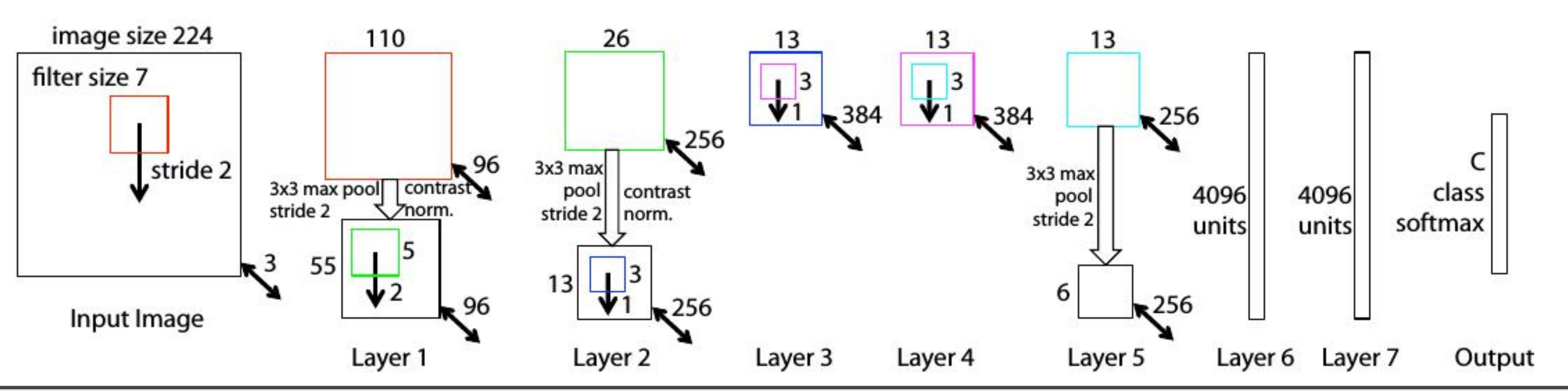


DropConnect

DNNs Architectures

- **LeNet** by Yann LeCun, Léon Bottou & Yoshua Bengio (1998)
- **AlexNet** by Alex Krizhevsky, Ilya Sutskever & Geoff Hinton (2012)
- **ZF Net** by Matthew Zeiler & Rob Fergus (2013)
- **VGGNet** by Karen Simonyan & Andrew Zisserman (2014)
- **GoogLeNet** by Szegedy et al. (2014)
- **ResNet** by Kaiming He et al. (2015)

ZFNet [Zeiler & Fergus, 2013]



AlexNet but:

CONV1: change from (11x11 stride 4) to (7x7 stride 2)

CONV3,4,5: instead of 384, 384, 256 filters use 512, 1024, 512

DNNs Architectures

- **LeNet** by Yann LeCun, Léon Bottou & Yoshua Bengio (1998)
- **AlexNet** by Alex Krizhevsky, Ilya Sutskever & Geoff Hinton (2012)
- **ZF Net** by Matthew Zeiler & Rob Fergus (2013)
- **VGGNet** by Karen Simonyan & Andrew Zisserman (2014)
- **GoogLeNet** by Szegedy et al. (2014)
- **ResNet** by Kaiming He et al. (2015)

VGGNet [Simonyan & Zisserman, 2014]

Small filters, Deeper networks

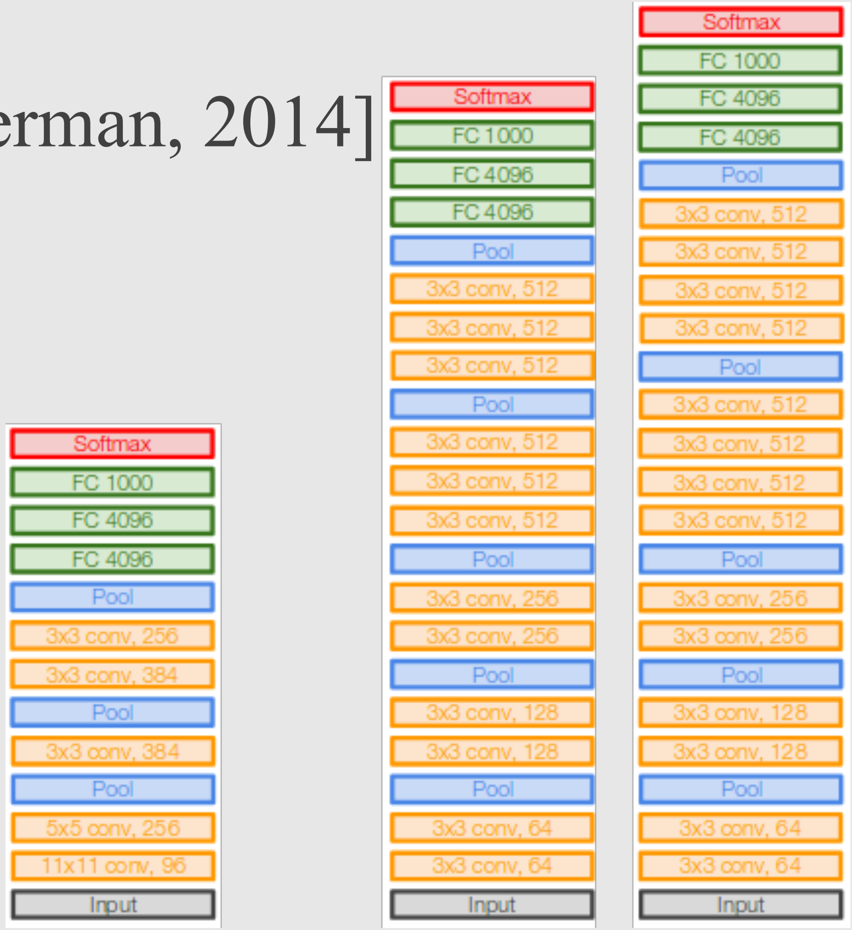
8 layers (AlexNet)

16 - 19 layers (VGG16Net)

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

11.7% in ILSVRC'13 (ZFNet)

7.3% in ILSVRC'14

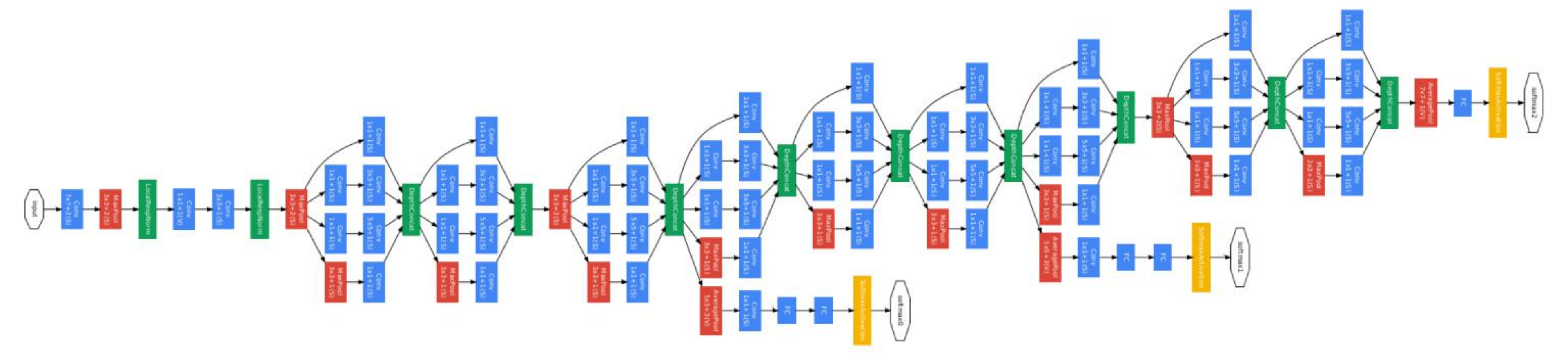


AlexNet VGG16 VGG19

DNNs Architectures

- **LeNet** by Yann LeCun, Léon Bottou & Yoshua Bengio (1998)
- **AlexNet** by Alex Krizhevsky, Ilya Sutskever & Geoff Hinton (2012)
- **ZF Net** by Matthew Zeiler & Rob Fergus (2013)
- **VGGNet** by Karen Simonyan & Andrew Zisserman (2014)
- **GoogLeNet** by Szegedy et al. (2014)
- **ResNet** by Kaiming He et al. (2015)

GoogLeNet [Szegedy et al., 2014]



11.7% in ILSVRC’13 (ZFNet)

7.3% in ILSVRC’14 (VGGNet)

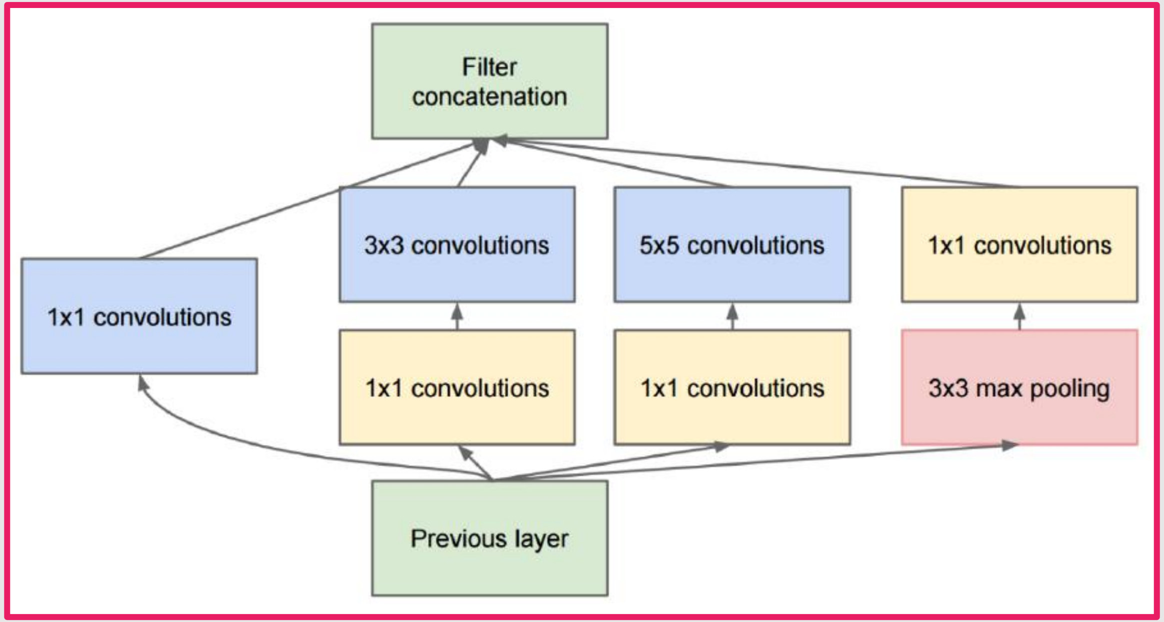
6.7% in ILSVRC’14 (GoogLeNet)

GoogLeNet [Szegedy et al., 2014]

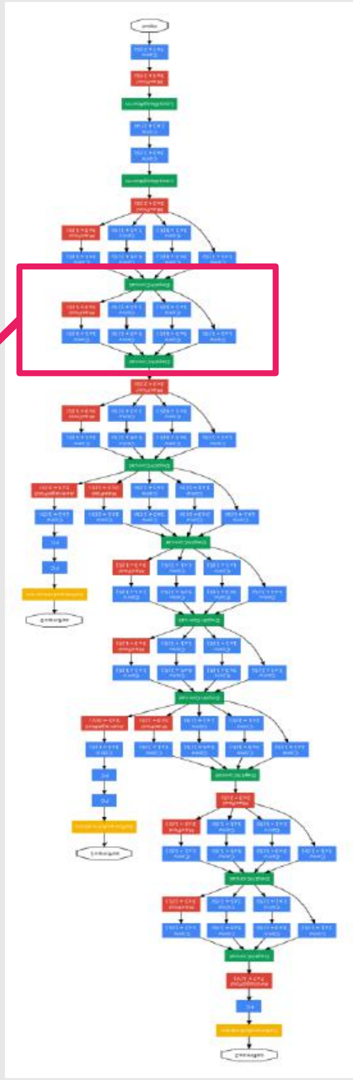
Deeper networks, with computational efficiency

- 22 layers
- Efficient “Inception” module
- No FC layers
- Only 5 million parameters!
  12x less than AlexNet

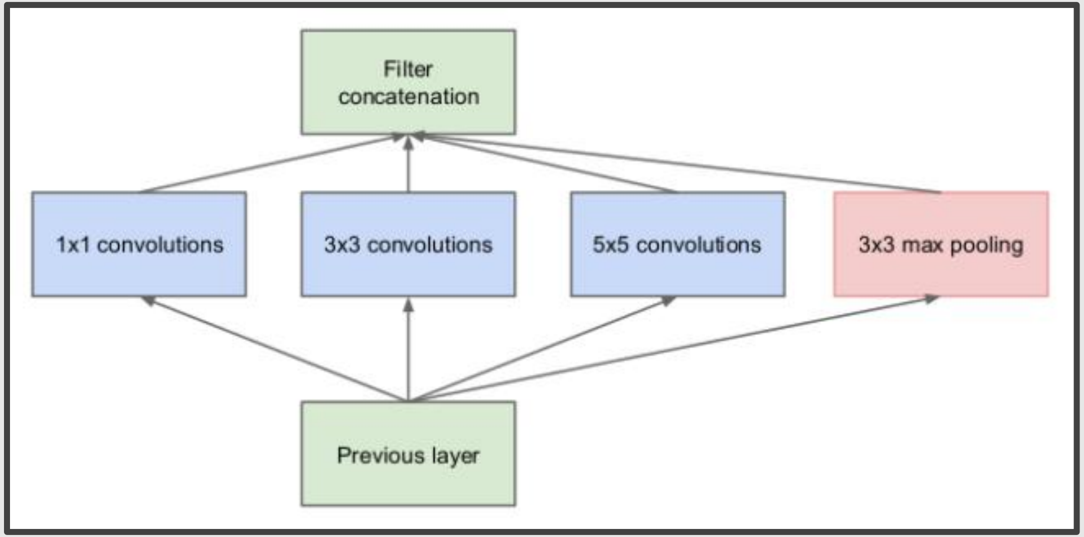
GoogLeNet [Szegedy et al., 2014]



Inception Module



GoogLeNet [Szegedy et al., 2014]



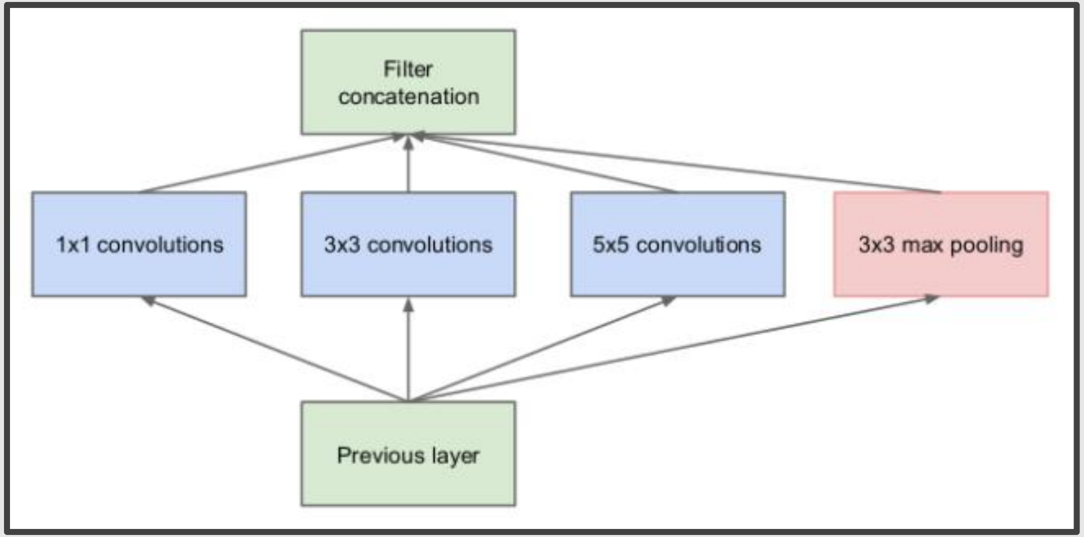
Naive Inception Module

Apply parallel filters 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

GoogLeNet [Szegedy et al., 2014]



Naive Inception Module

Apply parallel filters on the input from previous layer:

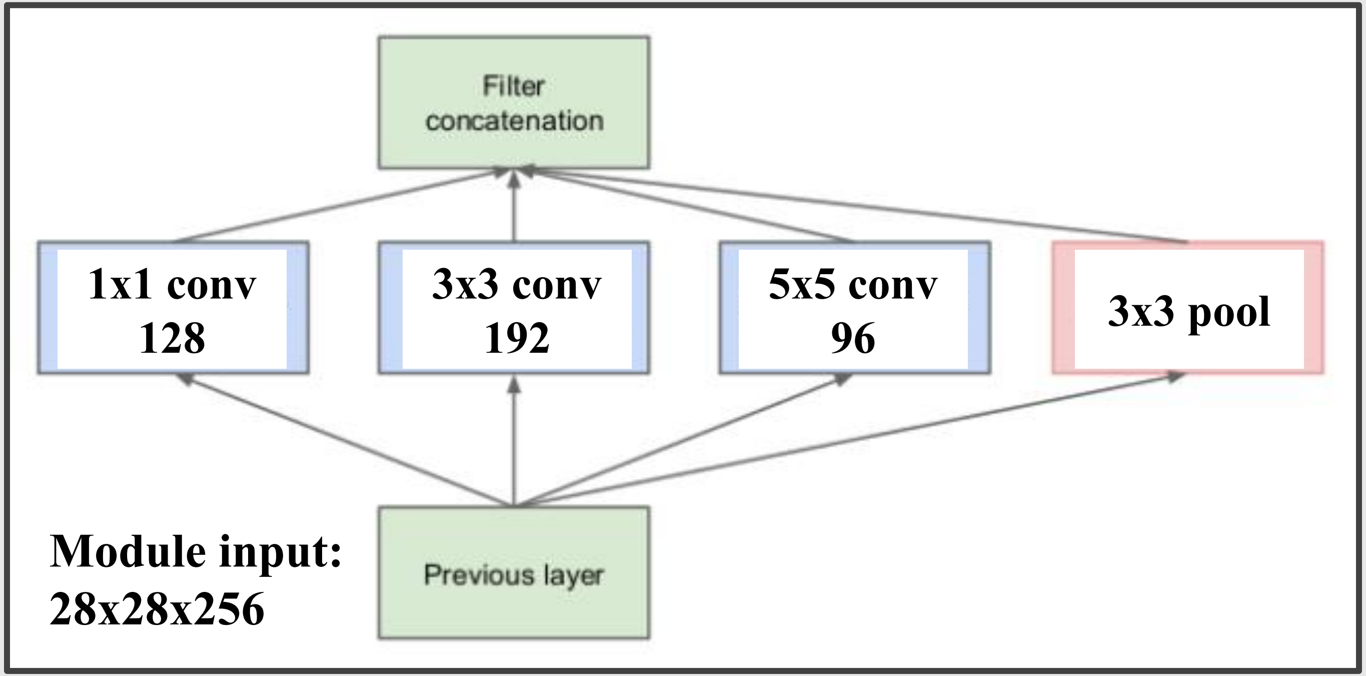
- Multiple receptive field sizes for convolution (1x1,3x3,5x5)
- Pooling operation (3x3)

Concatenate all filter outputs together depth-wise

Q: What is the problem with this?

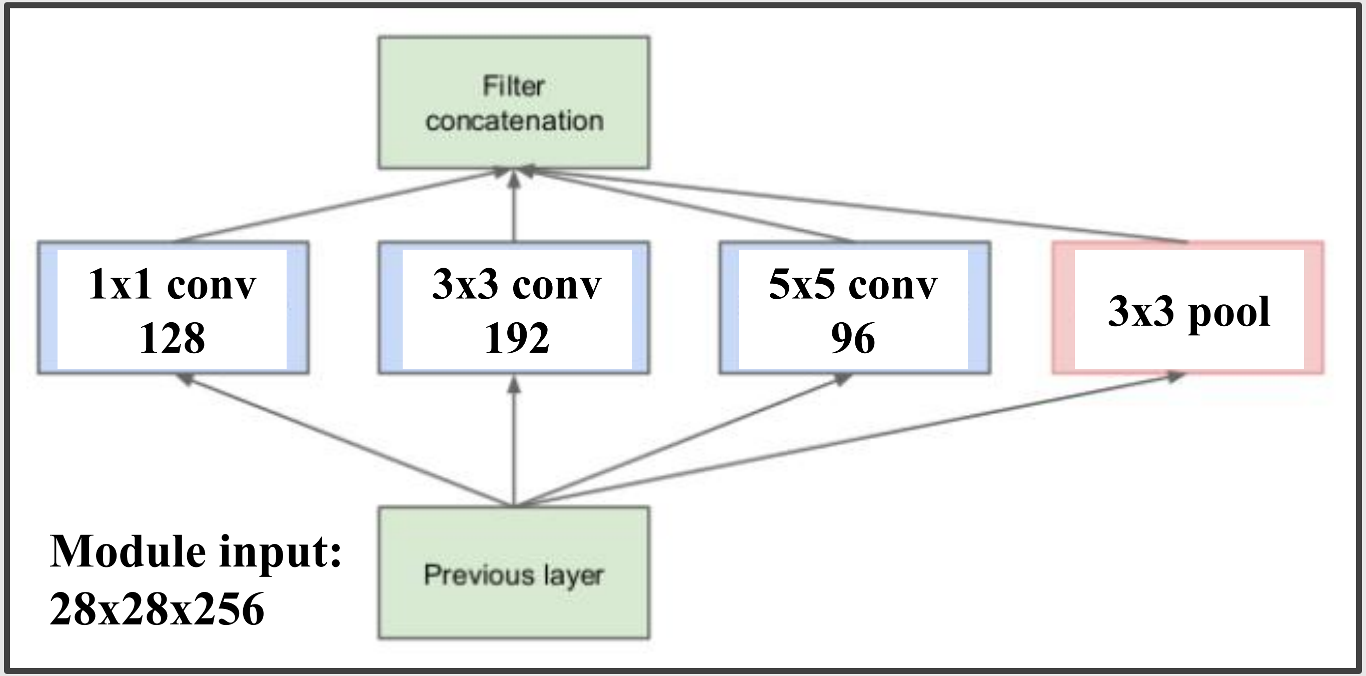
GoogLeNet [Szegedy et al., 2014]

Example:



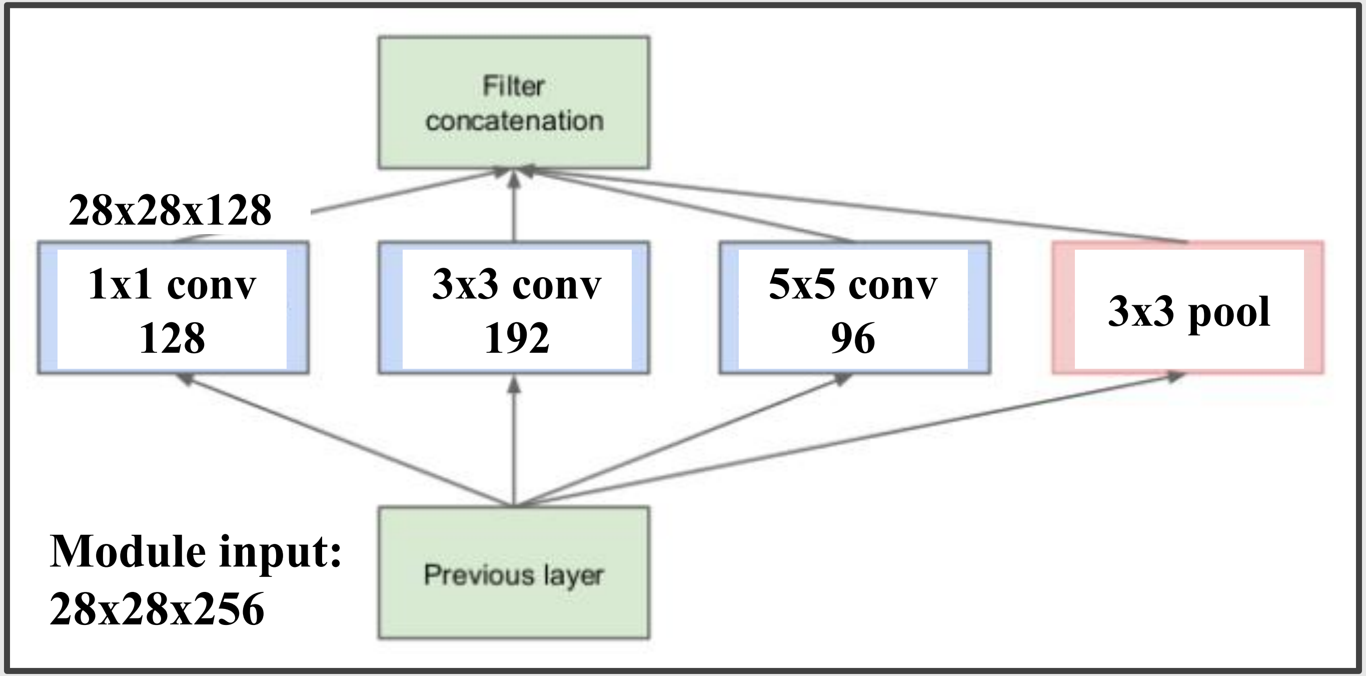
GoogLeNet [Szegedy et al., 2014]

Example: What is the output size of the **1x1 conv, with 128 filters**?



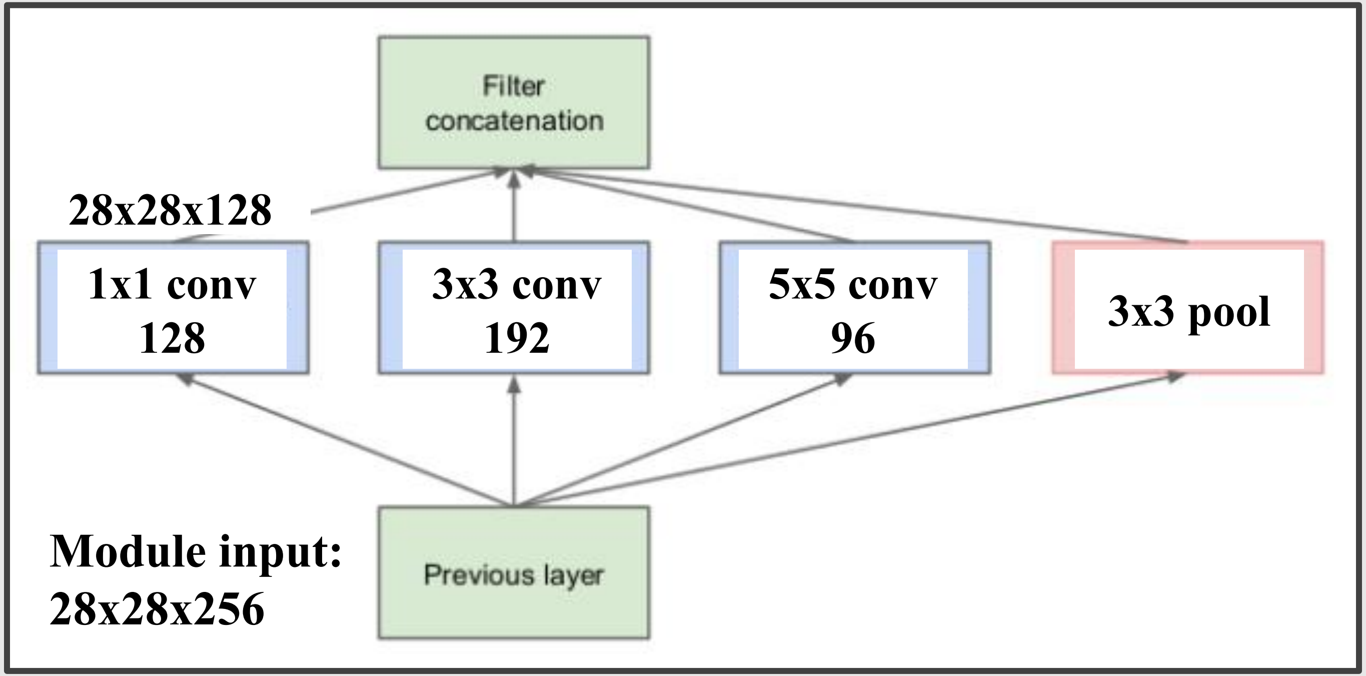
GoogLeNet [Szegedy et al., 2014]

Example: What is the output size of the **1x1 conv, with 128 filters**?



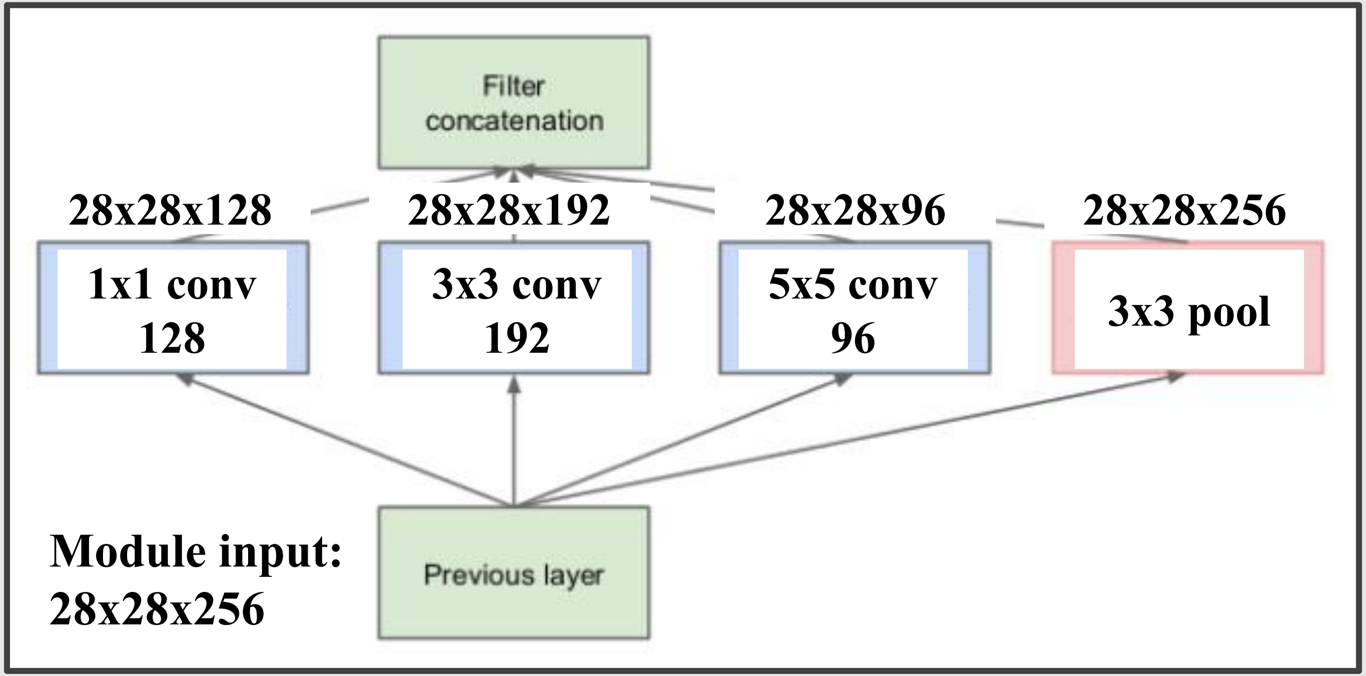
GoogLeNet [Szegedy et al., 2014]

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



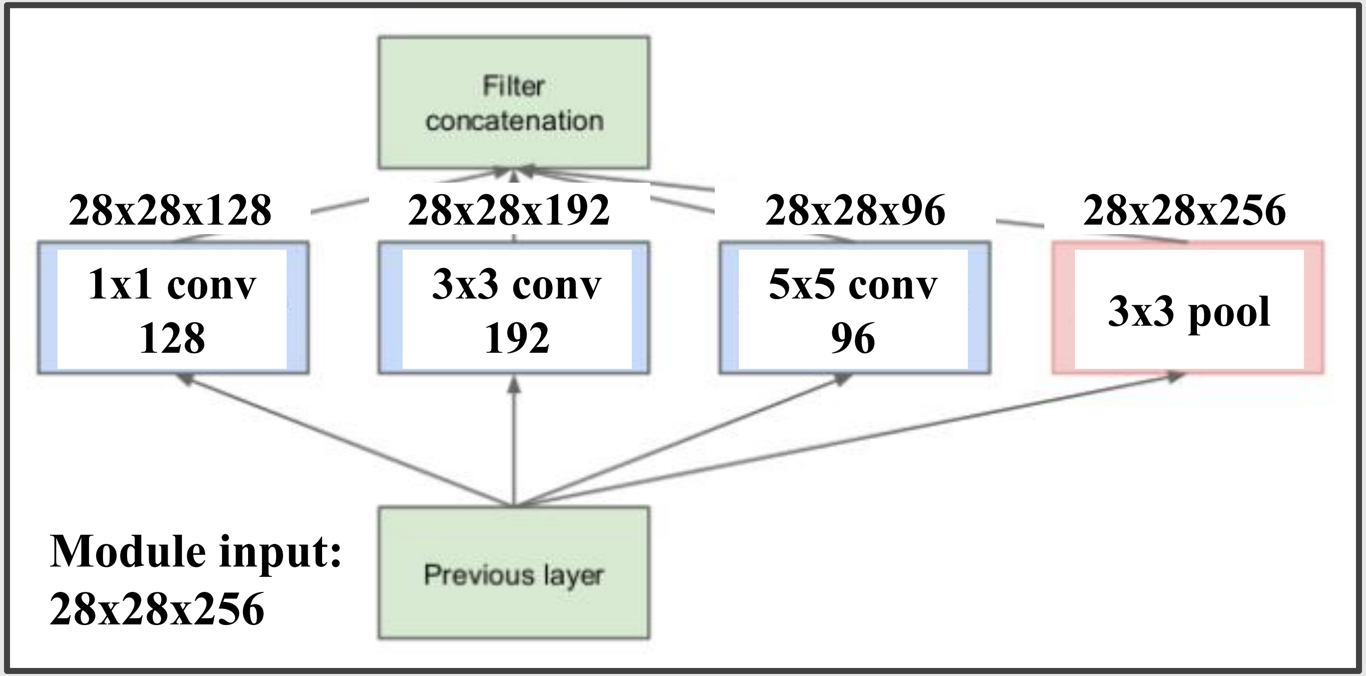
GoogLeNet [Szegedy et al., 2014]

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



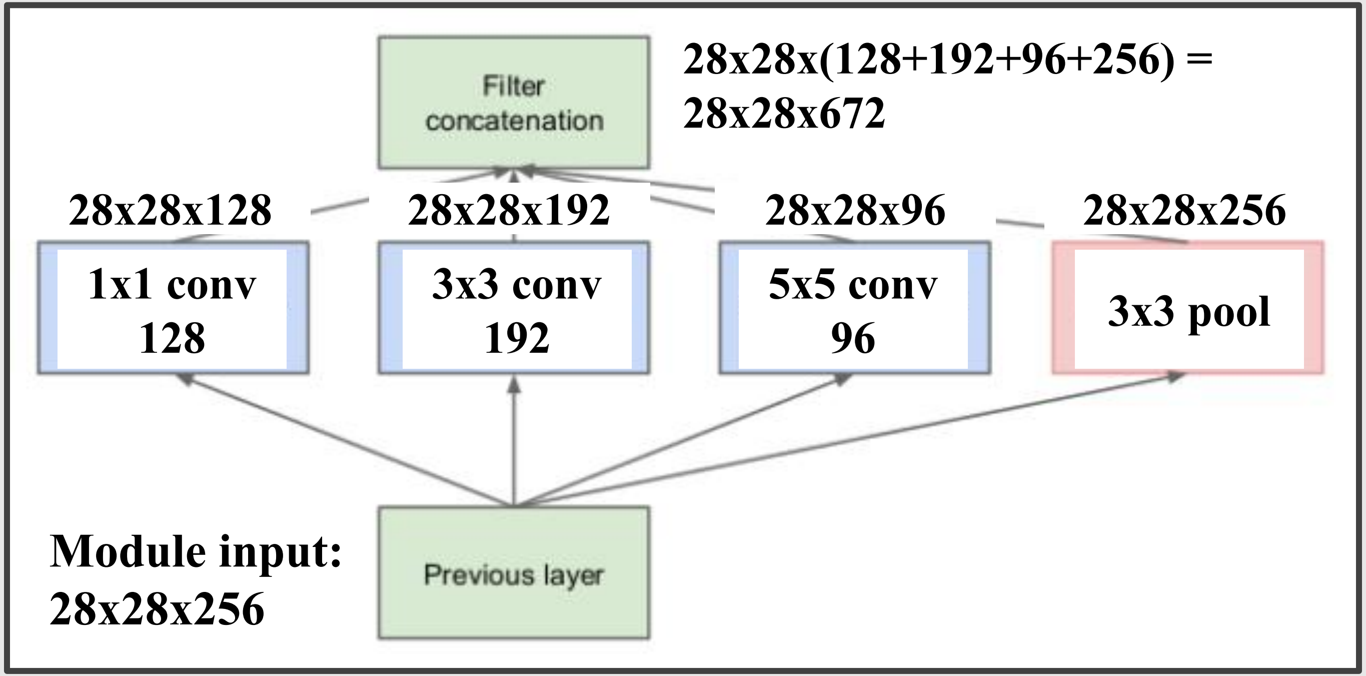
GoogLeNet [Szegedy et al., 2014]

Example: What is output size after filter concatenation?



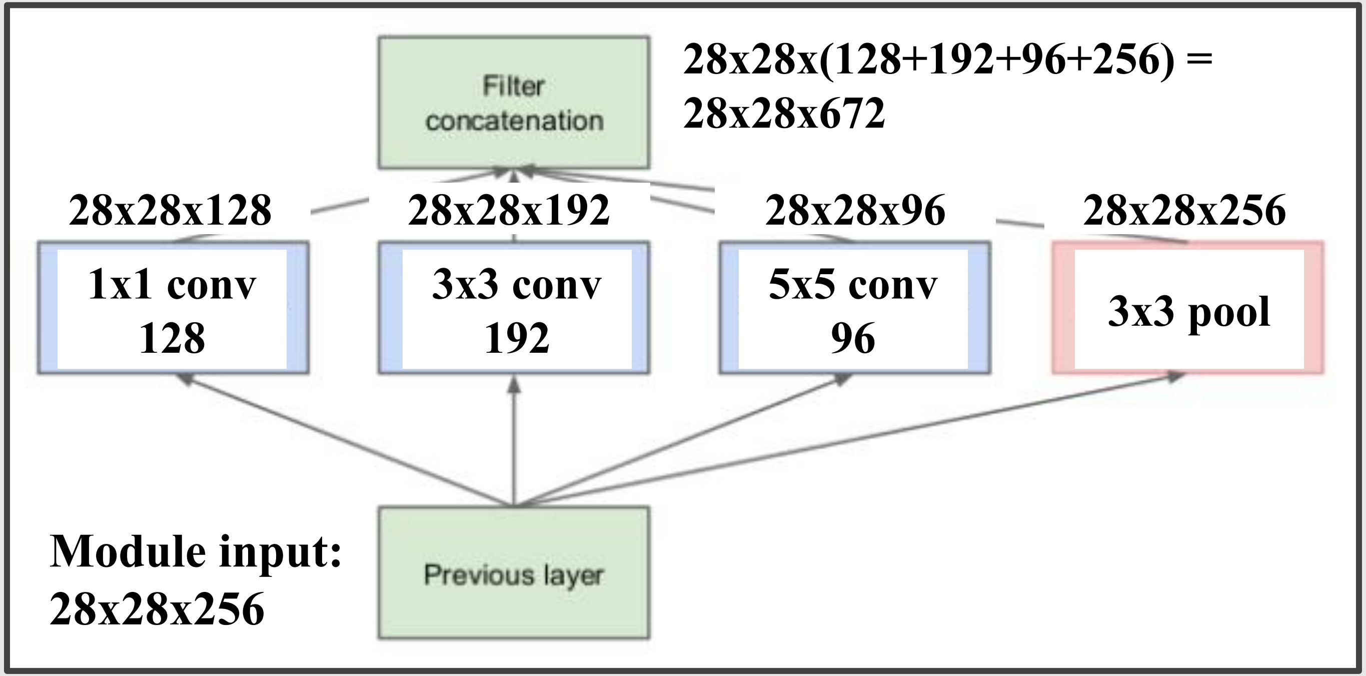
GoogLeNet [Szegedy et al., 2014]

Example: What is output size after filter concatenation?



GoogLeNet [Szegedy et al., 2014]

Example: What is output size after filter concatenation?

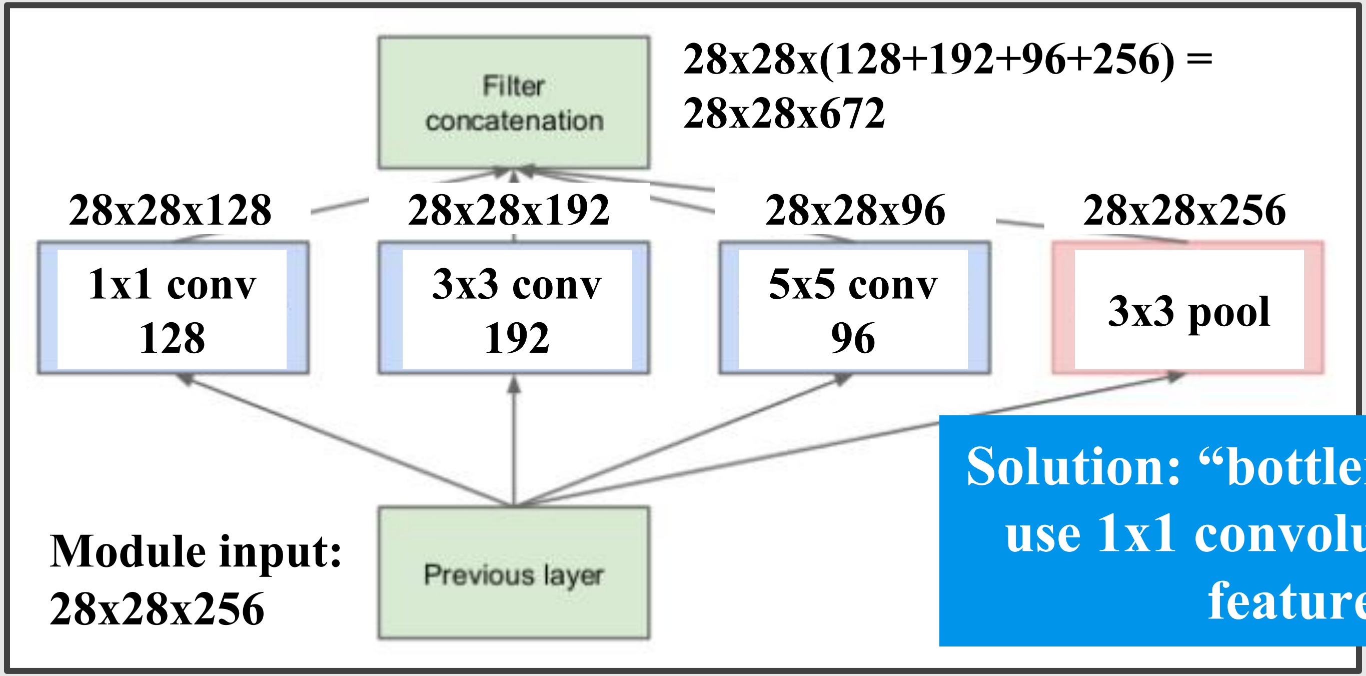


Conv Ops:

854M ops!!

GoogLeNet [Szegedy et al., 2014]

Example: What is output size after filter concatenation?



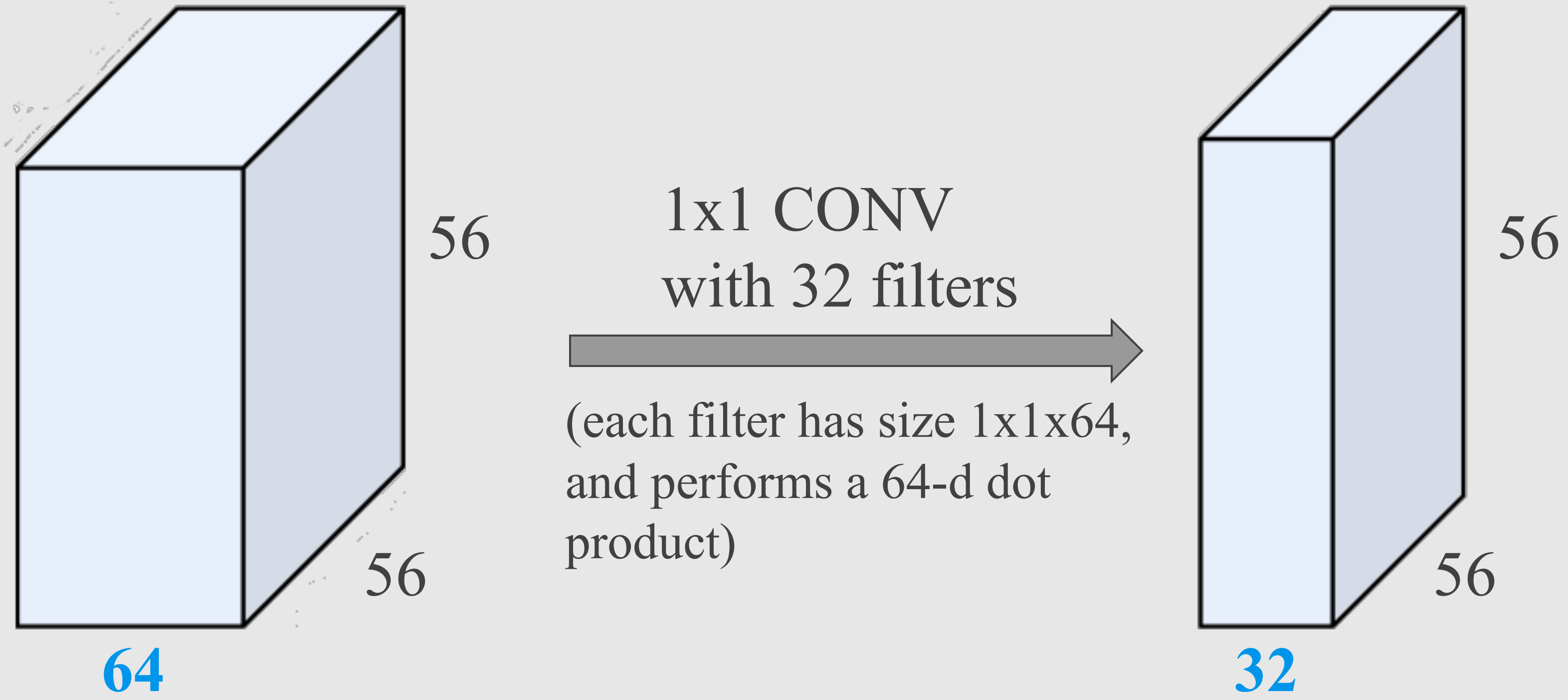
Conv Ops:

854M ops!!

Solution: "bottleneck" layers that use 1x1 convolutions to reduce feature depth

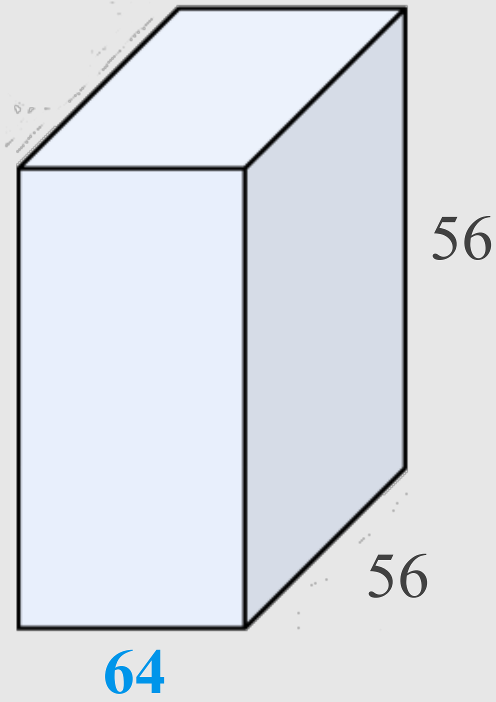
GoogLeNet [Szegedy et al., 2014]

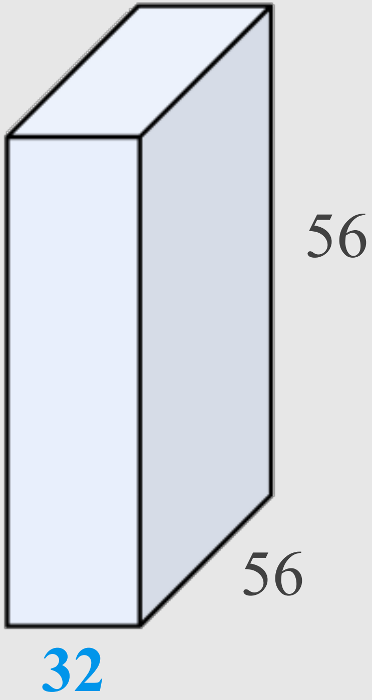
Reminder: 1x1 convolutions



GoogLeNet [Szegedy et al., 2014]

Reminder: 1x1 convolutions

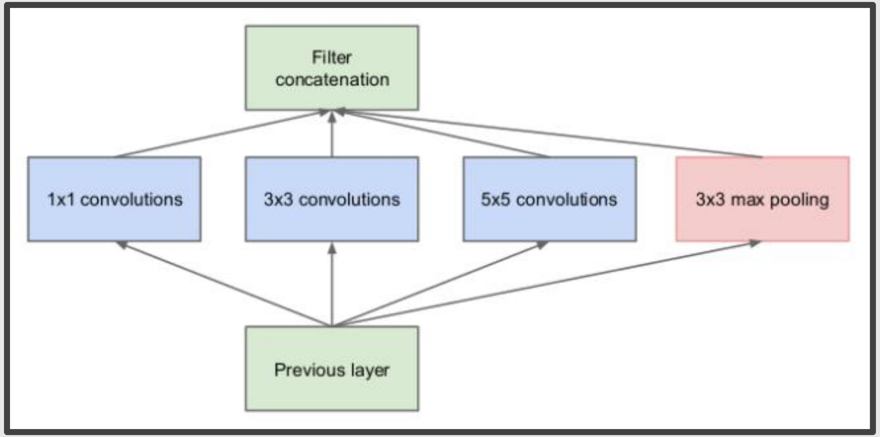




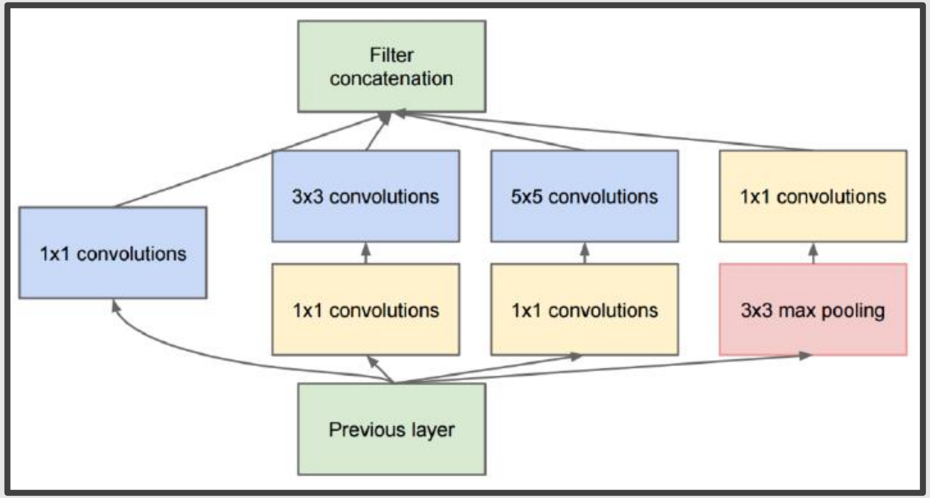
preserves spatial dimensions, reduces depth!

Projects depth to lower dimension (combination of feature maps)

GoogLeNet [Szegedy et al., 2014]



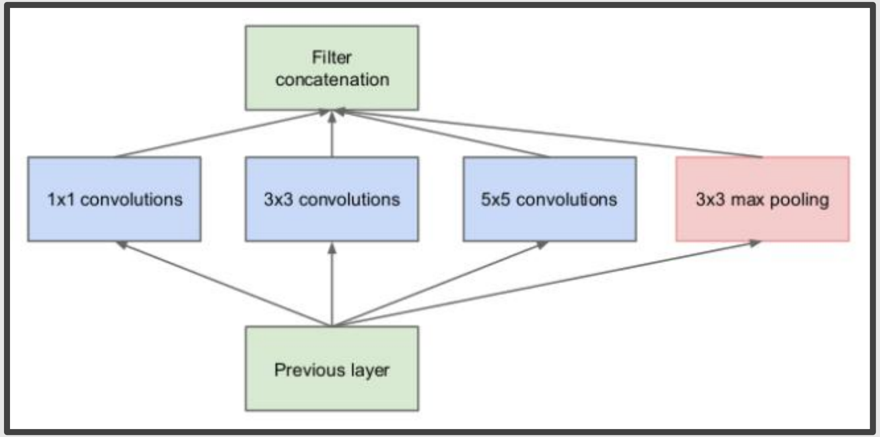
Naive Inception Module



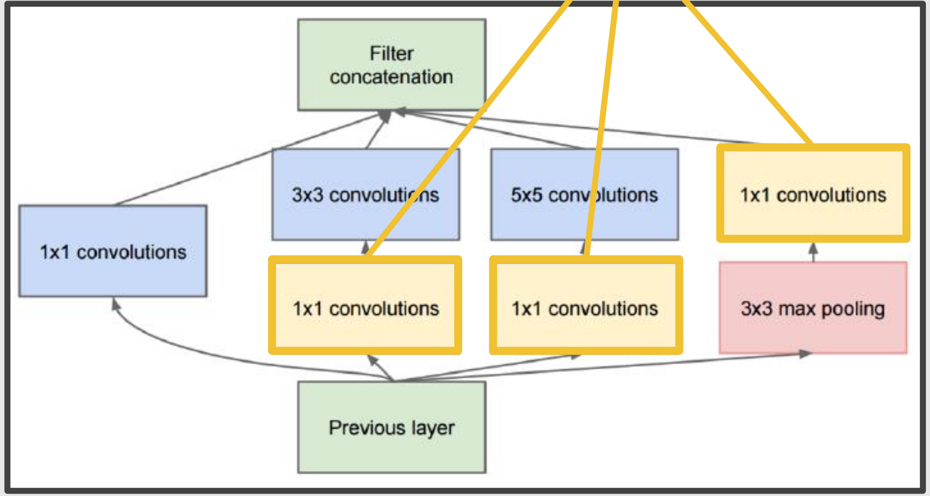
Inception Module

GoogLeNet [Szegedy et al., 2014]

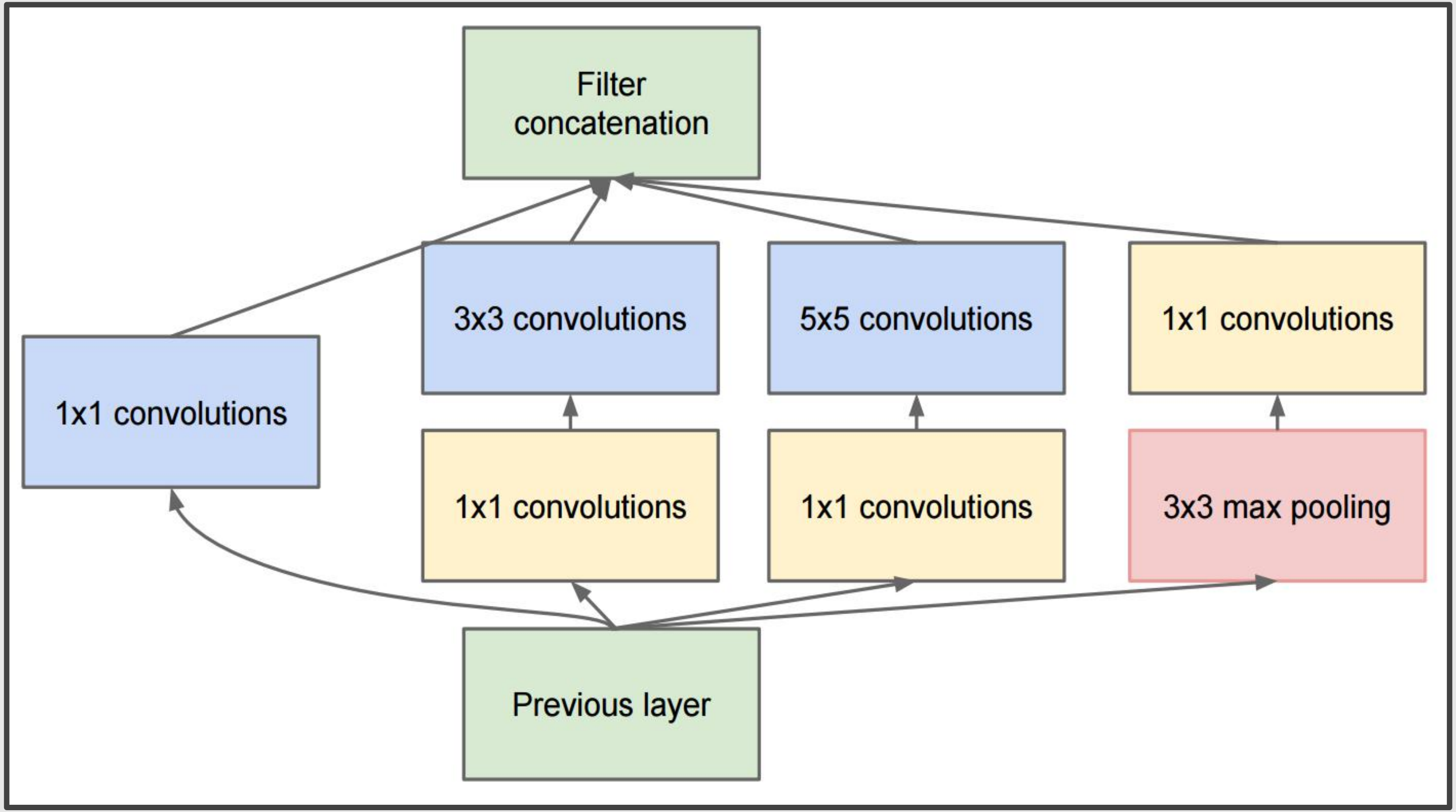
1x1 conv
“bottleneck” layers



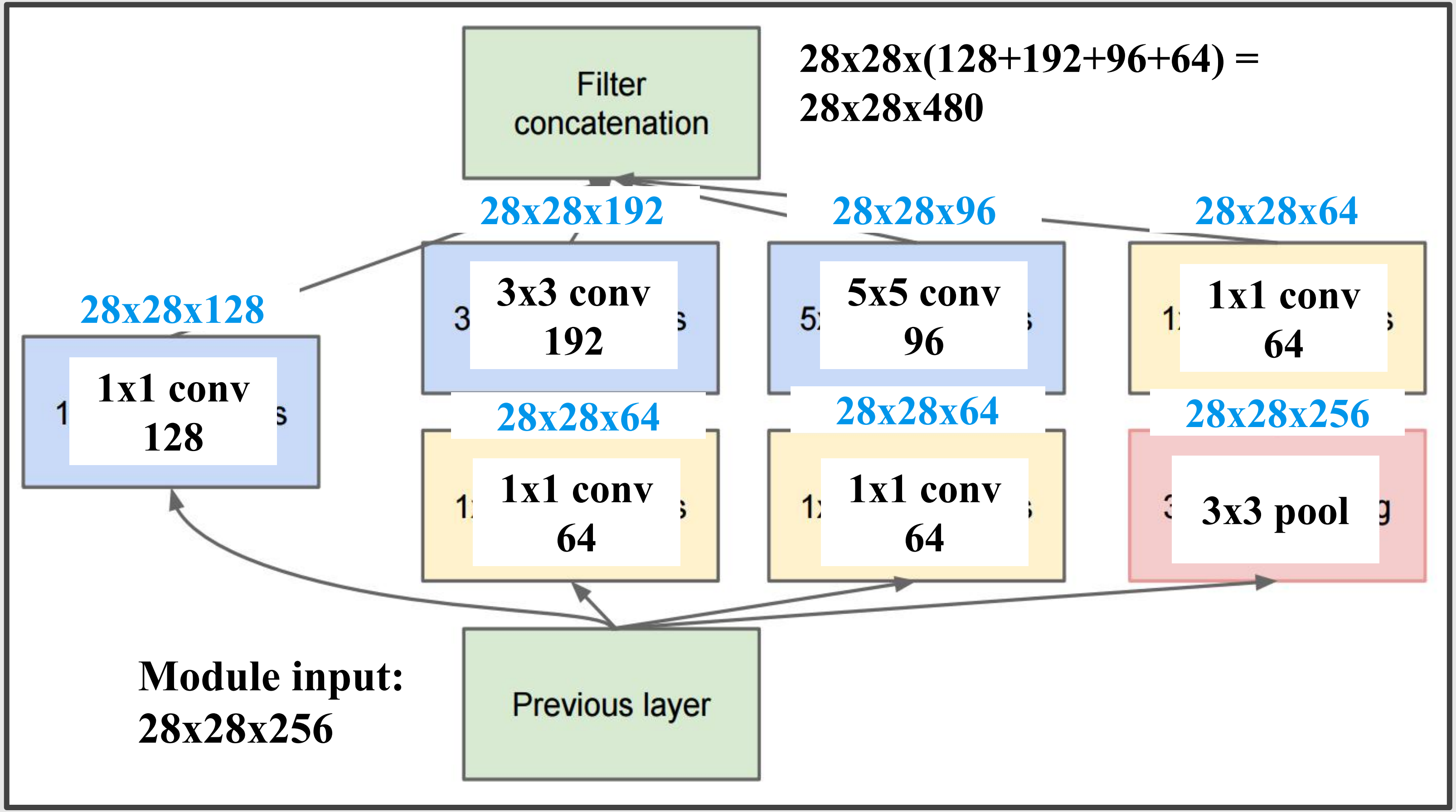
Naive Inception Module

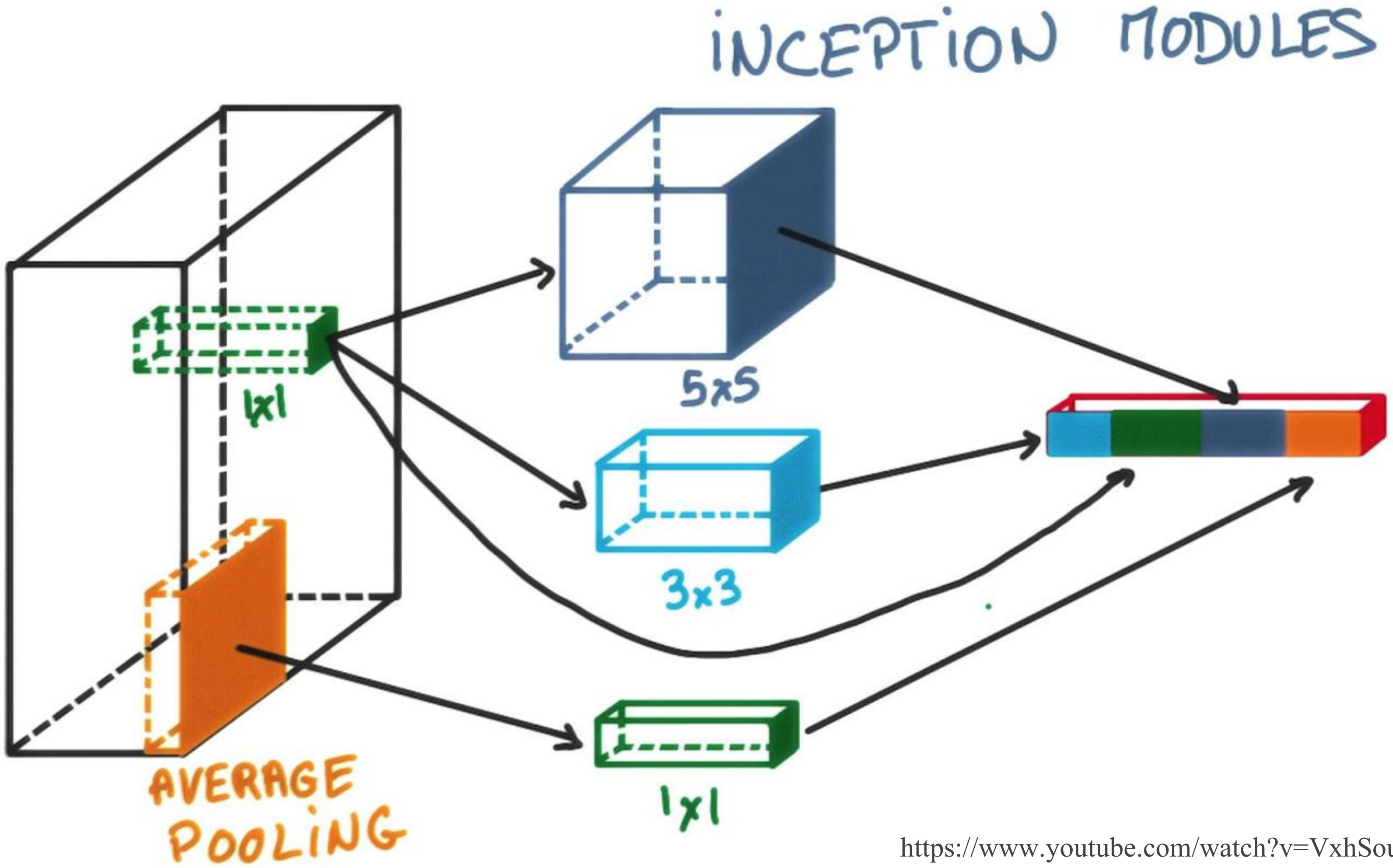


Inception Module


Filter concatenation
3x3 convolutions
5x5 convolutions
1x1 convolutions
1x1 convolutions
1x1 convolutions
1x1 convolutions
3x3 max pooling
Previous layer

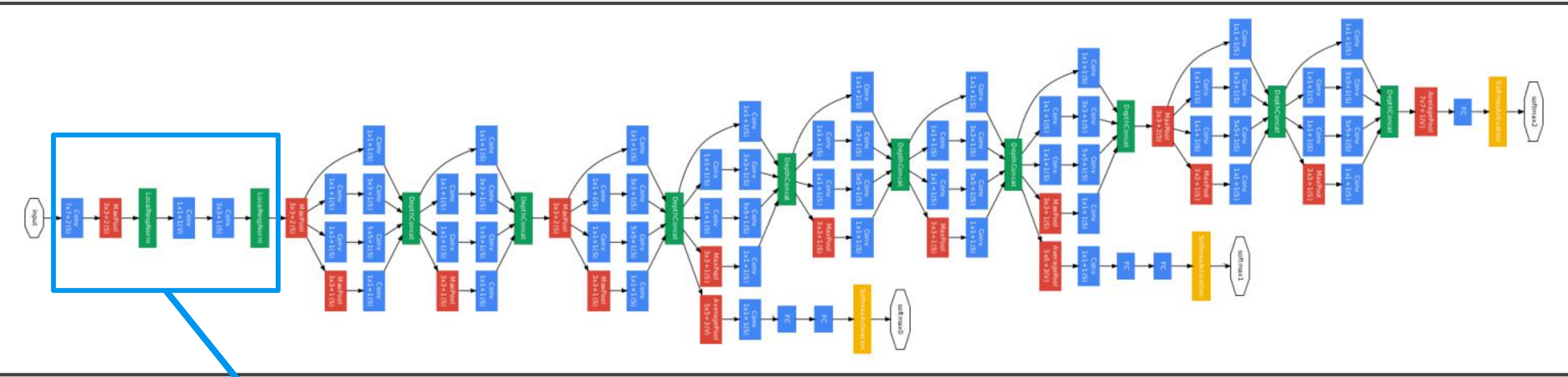
Conv Ops: 358M ops





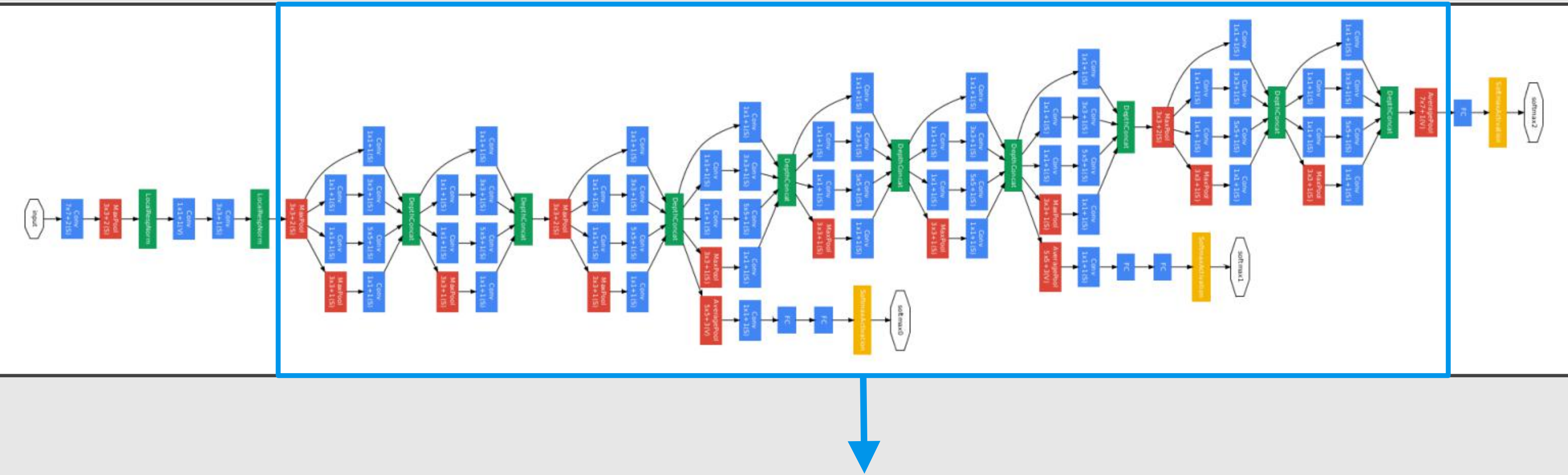
https://www.youtube.com/watch?v=VxhSouuSZDY

GoogLeNet [Szegedy et al., 2014]



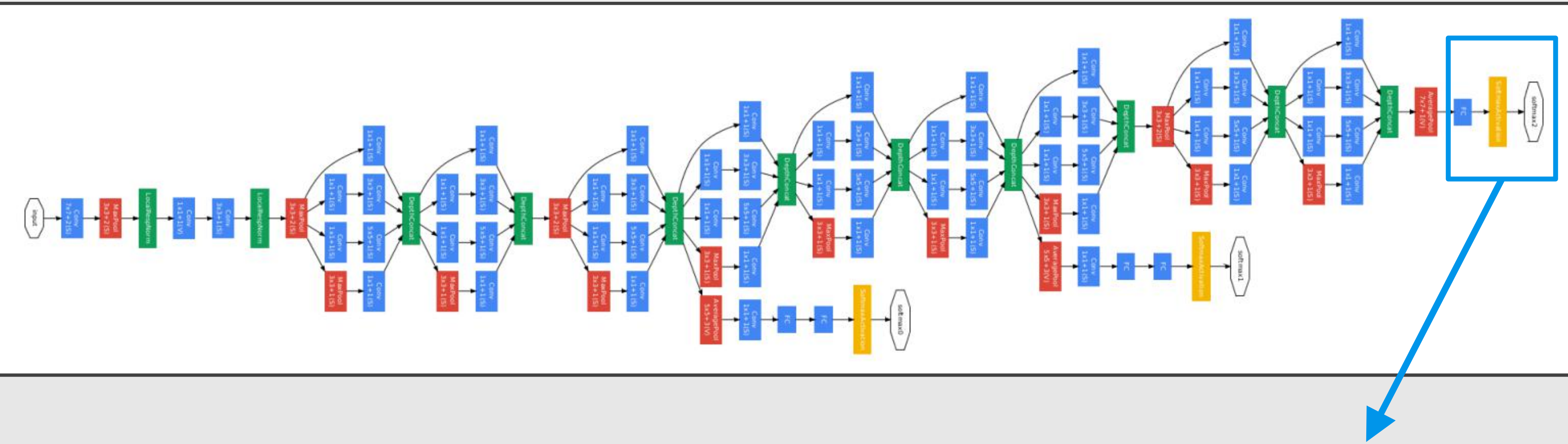
Conv-Pool 2x Conv-Pool

GoogLeNet [Szegedy et al., 2014]



Stacked Inception Modules

GoogLeNet [Szegedy et al., 2014]

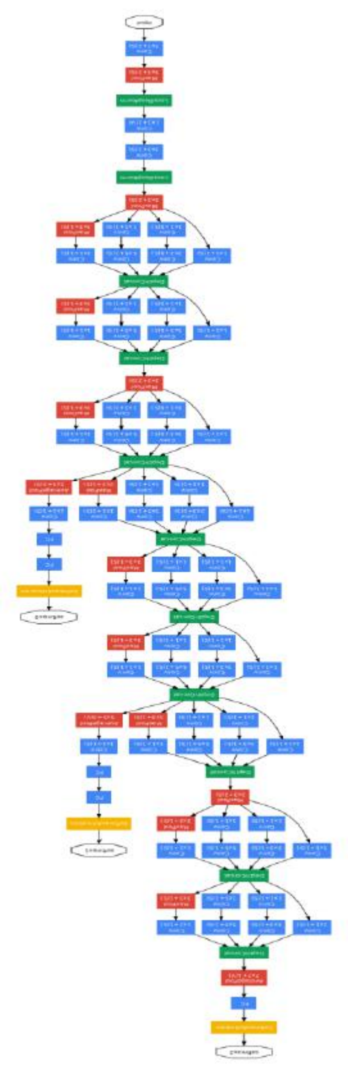


Classifier Output

GoogLeNet [Szegedy et al., 2014]

Deeper networks, with computational efficiency

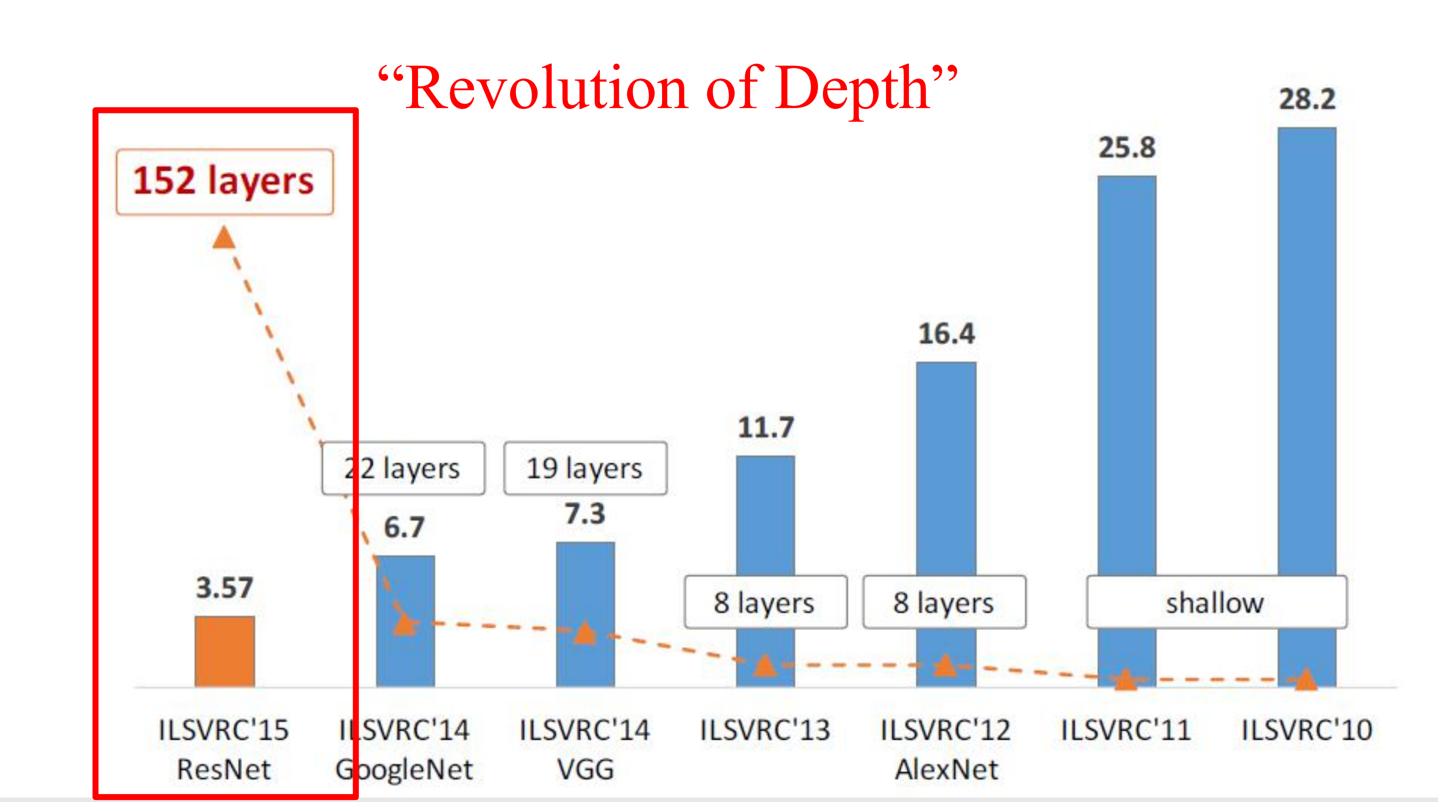
- 22 layers
- Efficient “Inception” module
- No FC layers
- Only 5 million parameters!
  12x less than AlexNet



DNNs Architectures

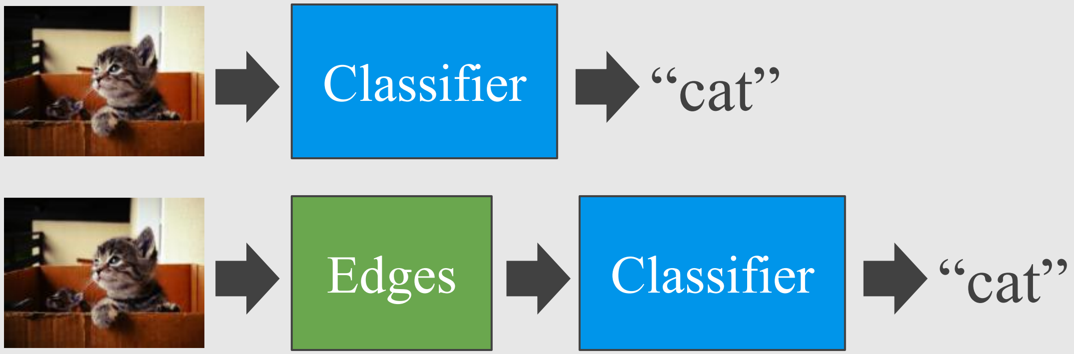
- **LeNet** by Yann LeCun, Léon Bottou & Yoshua Bengio (1998)
- **AlexNet** by Alex Krizhevsky, Ilya Sutskever & Geoff Hinton (2012)
- **ZF Net** by Matthew Zeiler & Rob Fergus (2013)
- **VGGNet** by Karen Simonyan & Andrew Zisserman (2014)
- **GoogLeNet** by Szegedy et al. (2014)
- **ResNet** by Kaiming He et al. (2015)

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

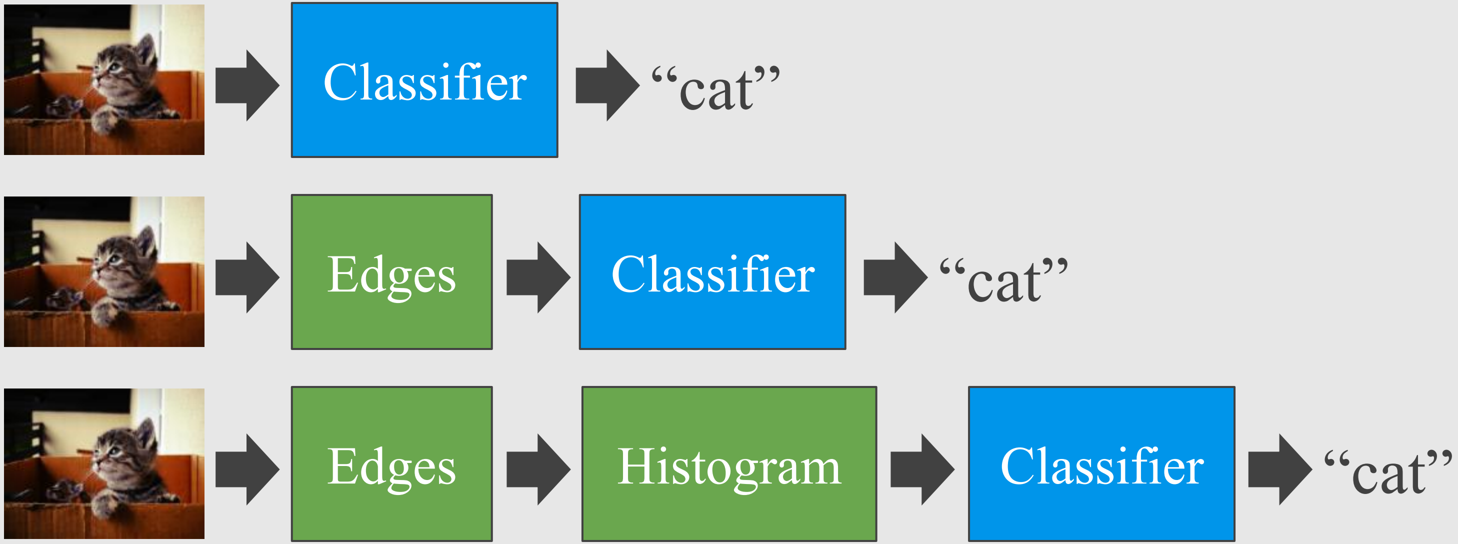


Traditional Recognition

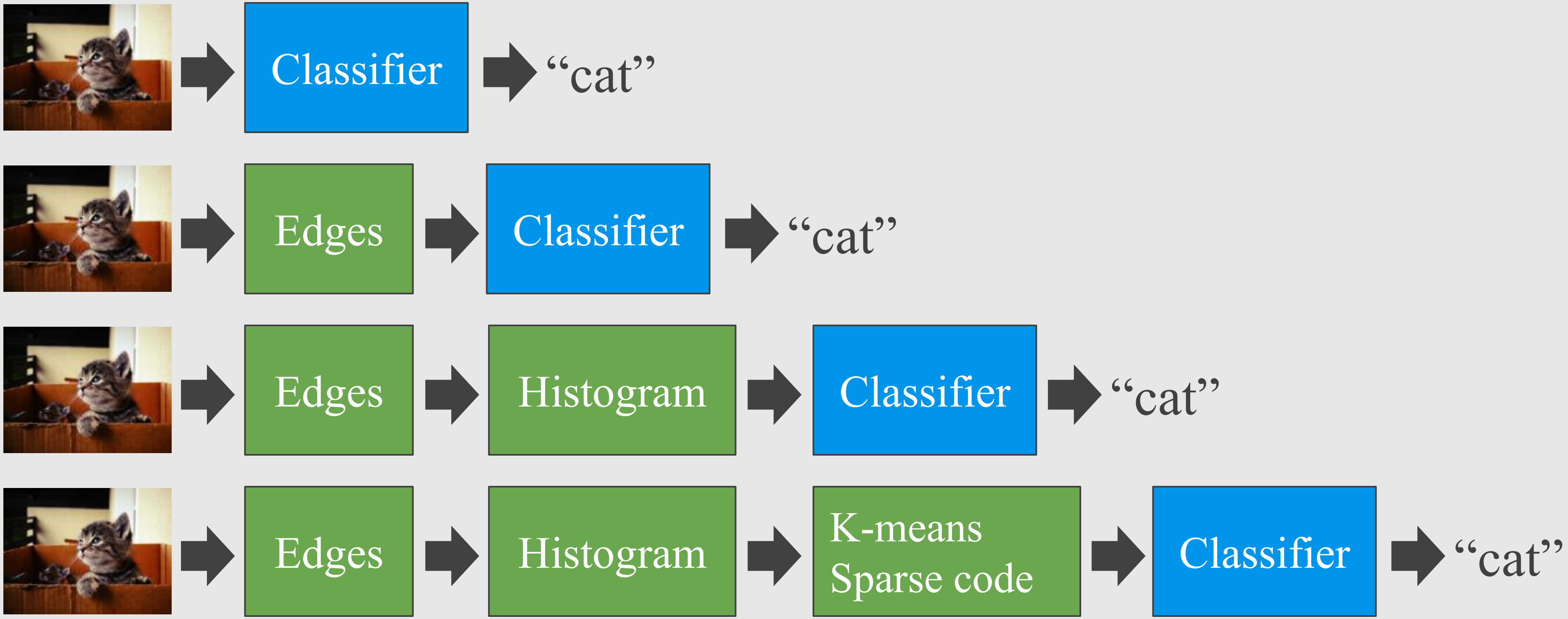
Traditional Recognition



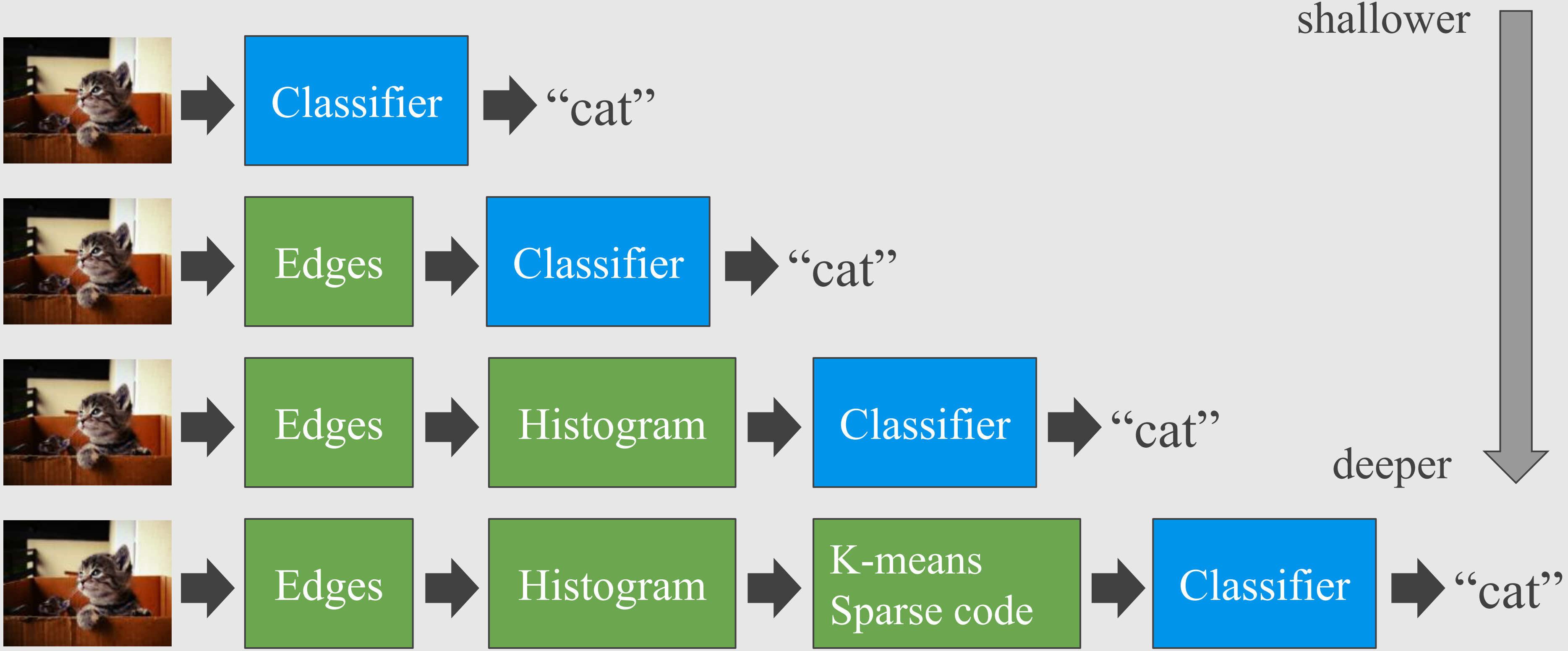
Traditional Recognition



Traditional Recognition

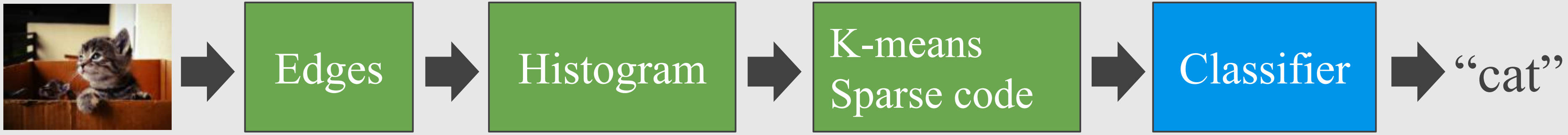


Traditional Recognition

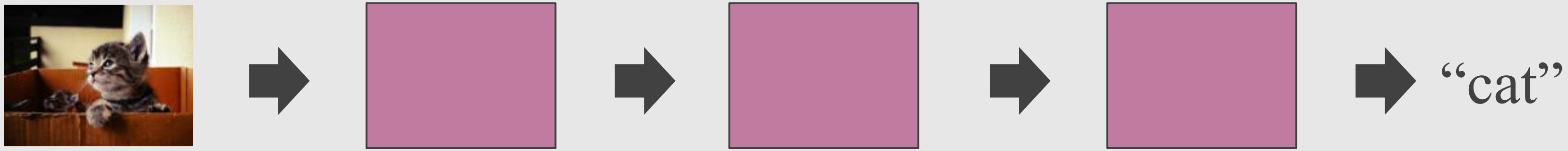


Deep Learning

Specialized components

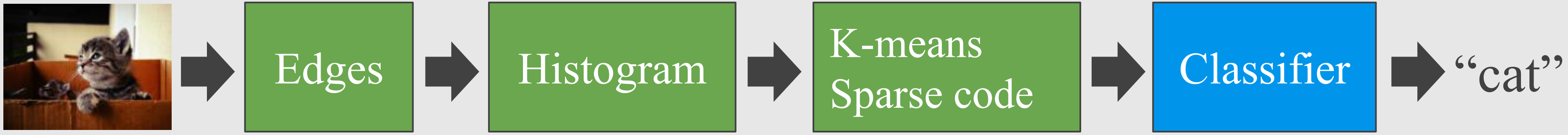


Generic components

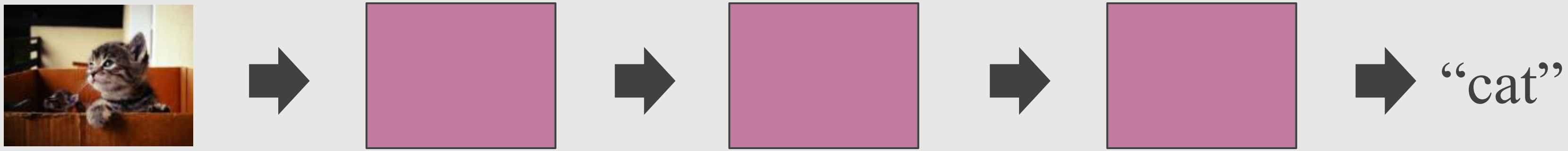


Deep Learning

Specialized components



Generic components



Generic components, going deeper

ResNet [He et al., 2015]

ResNet @ ILSVRC & COCO 2015 Competitions

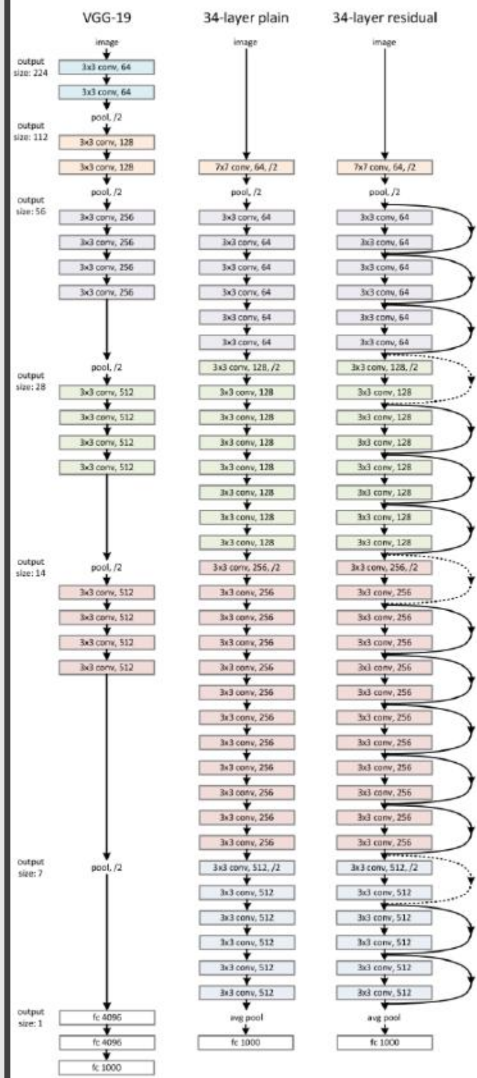
1st place in ALL five main tracks

- ImageNet Classification: “Ultra-deep” 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

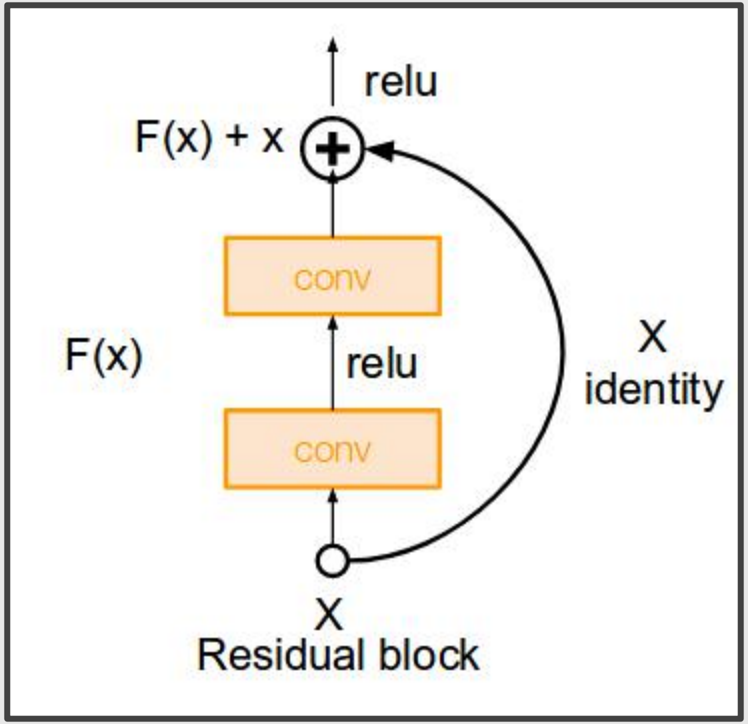
ResNet [He et al., 2015]

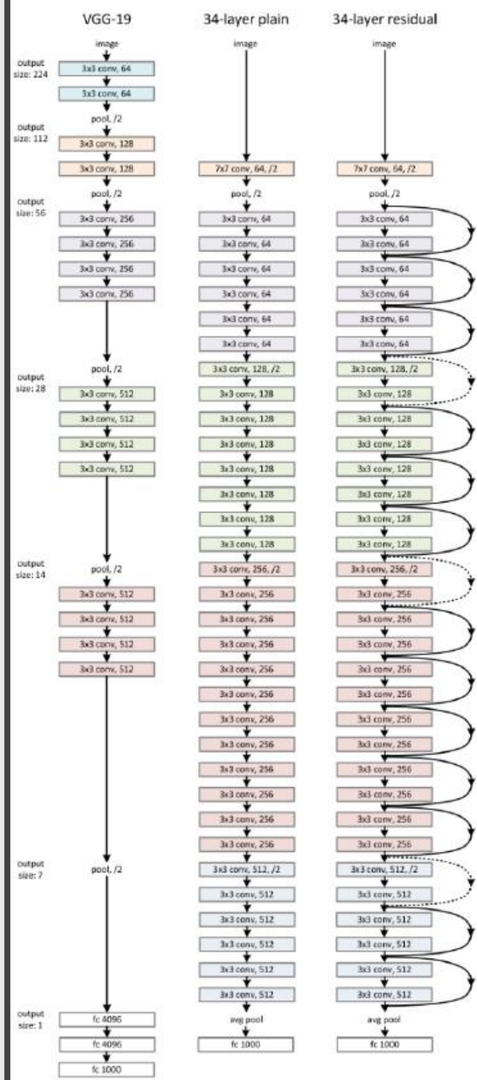
Very deep networks using residual connections

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



ResNet [He et al., 2015]



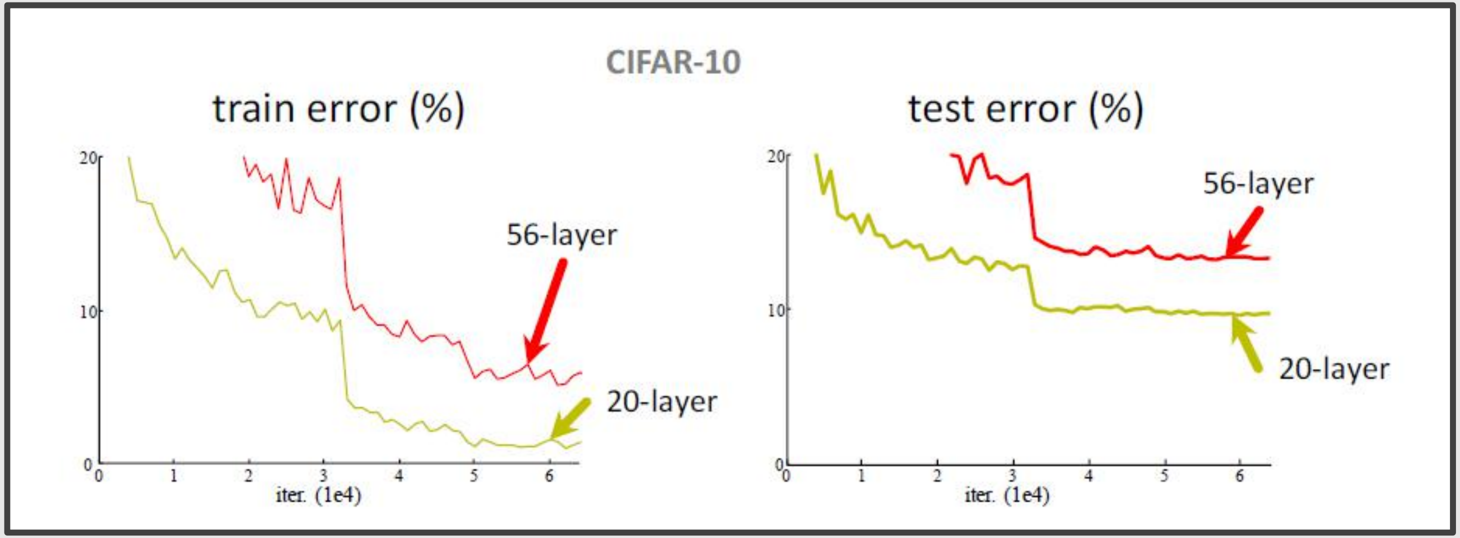


ResNet [He et al., 2015]

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

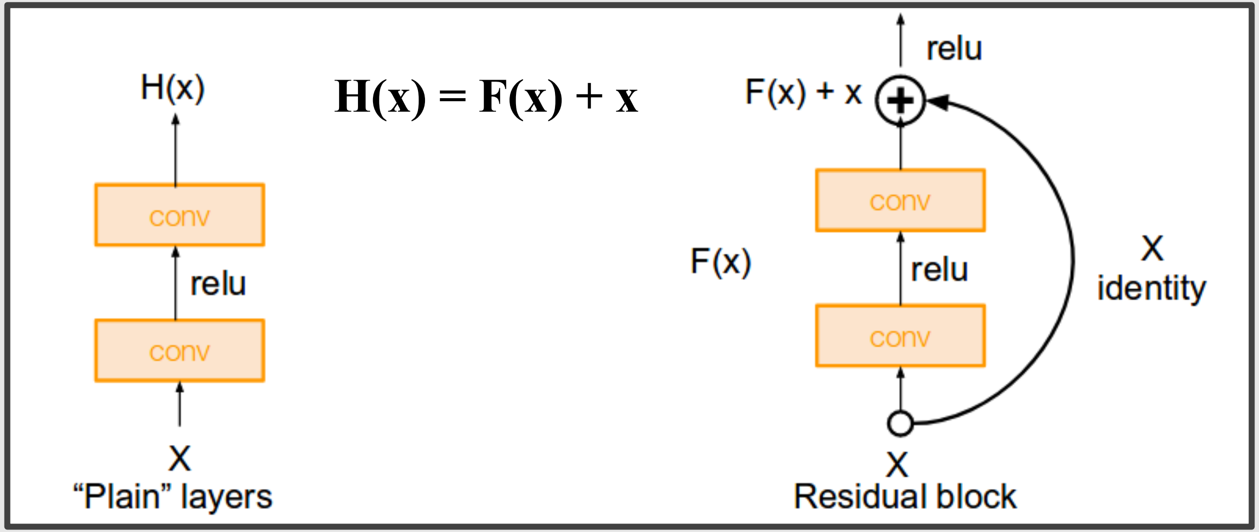
ResNet [He et al., 2015]

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



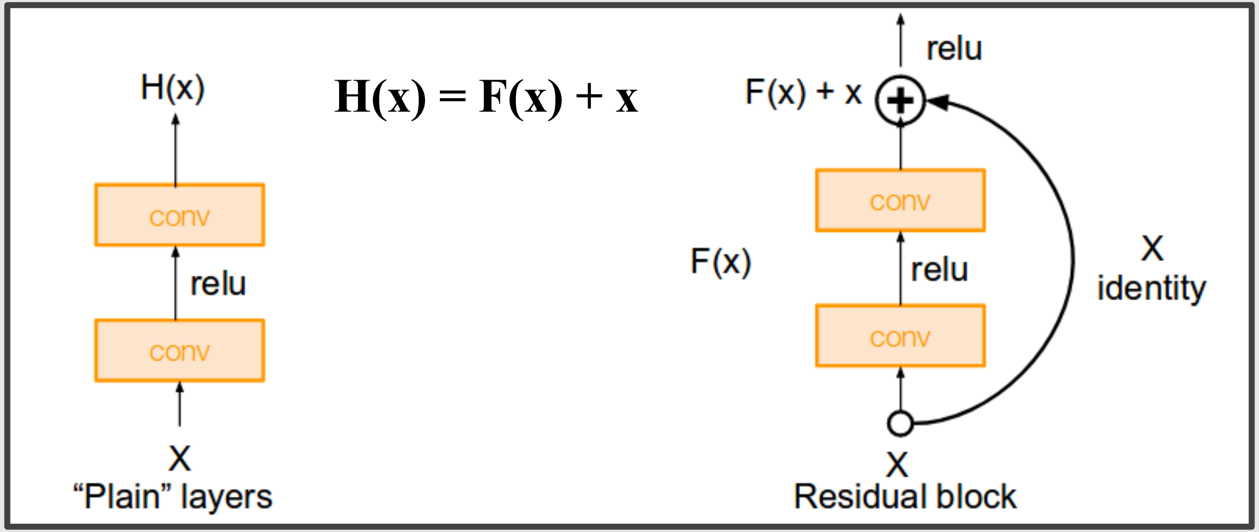
ResNet [He et al., 2015]

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



ResNet [He et al., 2015]

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

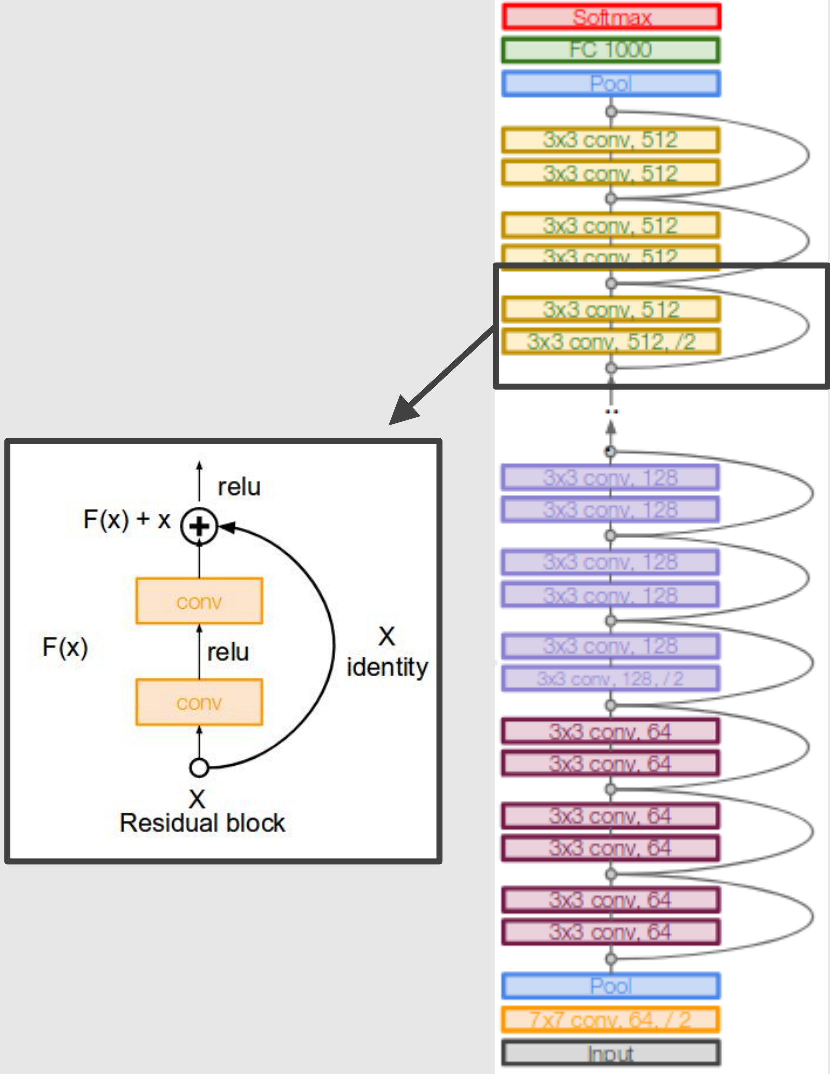


Use layers to fit residual F(x) = H(x) - x instead of H(x) directly

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
- No FC layers at the end (only FC 1000 to output classes)



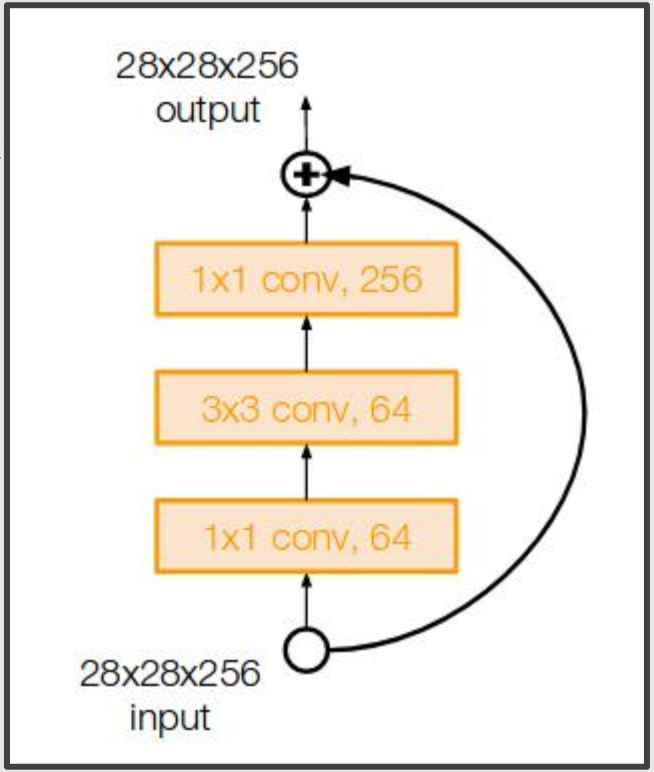
ResNet [He et al., 2015]

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

1x1 conv, 256 filters projects back to 256 feature maps (28x28x256)

3x3 conv operates over only 64 feature maps

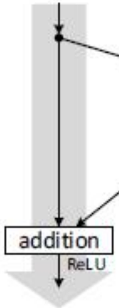
1x1 conv, 64 filters to project to 28x28x64



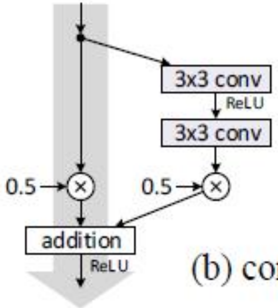
* ResNet-110 on CIFAR-10

$h(x) = x$

error: 6.6%



(a) original



(b) constant scaling

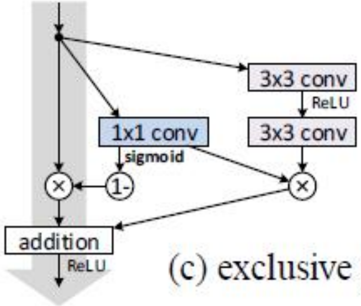
$h(x) = 0.5x$

error: 12.4%

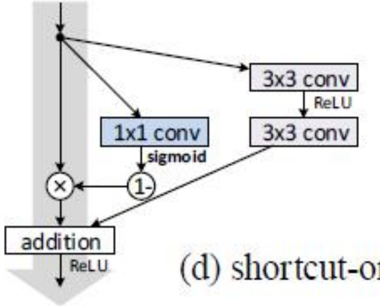
$h(x) = \text{gate} \cdot x$

error: 8.7%

*similar to "Highway Network"



(c) exclusive gating



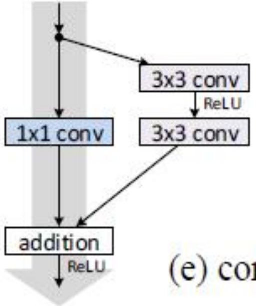
(d) shortcut-only gating

$h(x) = \text{gate} \cdot x$

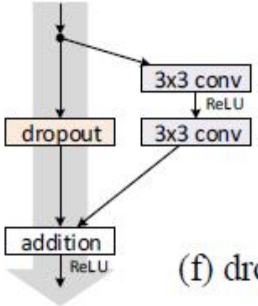
error: 12.9%

$h(x) = \text{conv}(x)$

error: 12.2%



(e) conv shortcut



(f) dropout shortcut

$h(x) = \text{dropout}(x)$

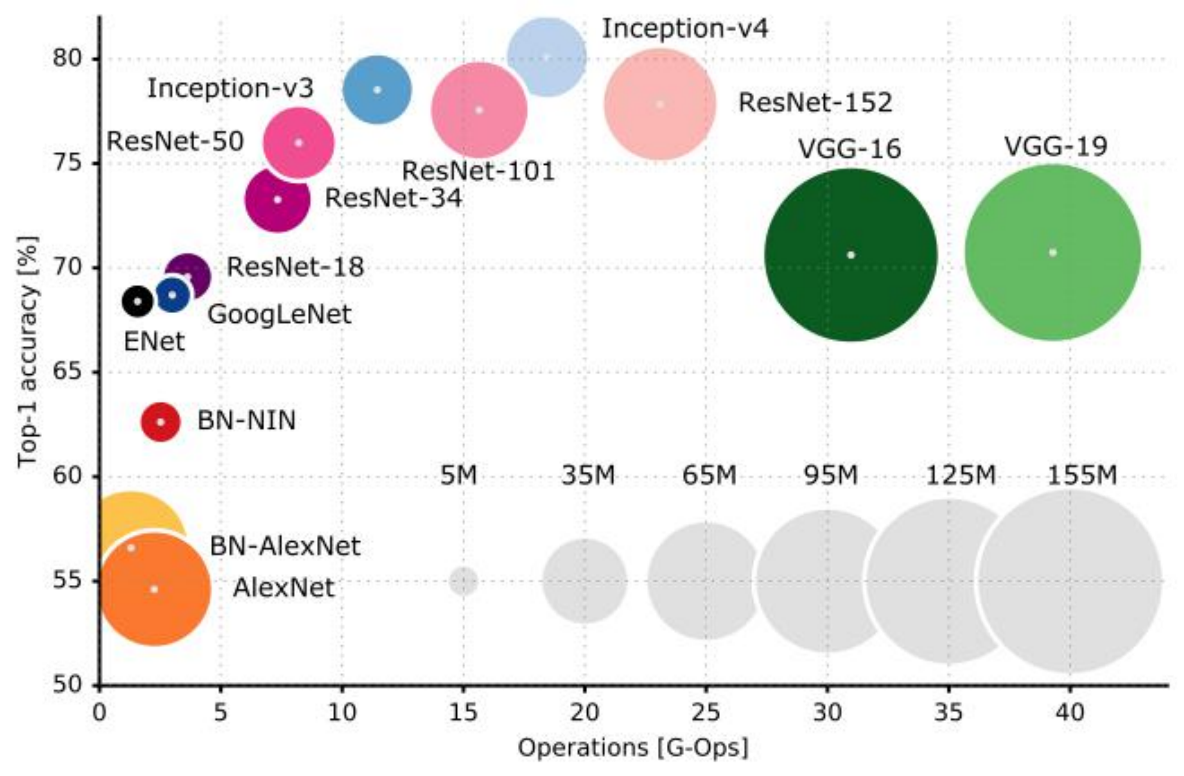
error: > 20%

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". arXiv 2016.

ResNet [He et al., 2015]

Details:

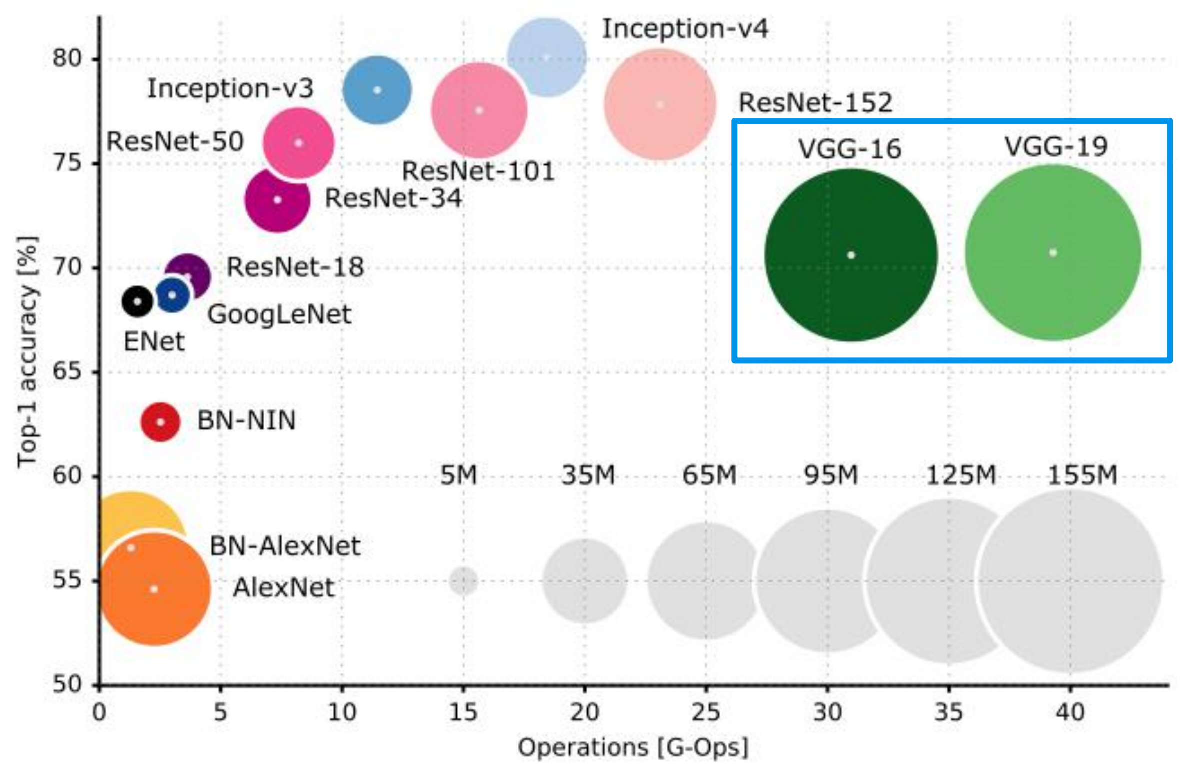
- Batch Normalization after every CONV layer
- Xavier/2 initialization from He et al.
- SGD + Momentum (0.9)
- Mini-batch size 256
- No dropout used



The size of the blobs is proportional to the number of network parameters.

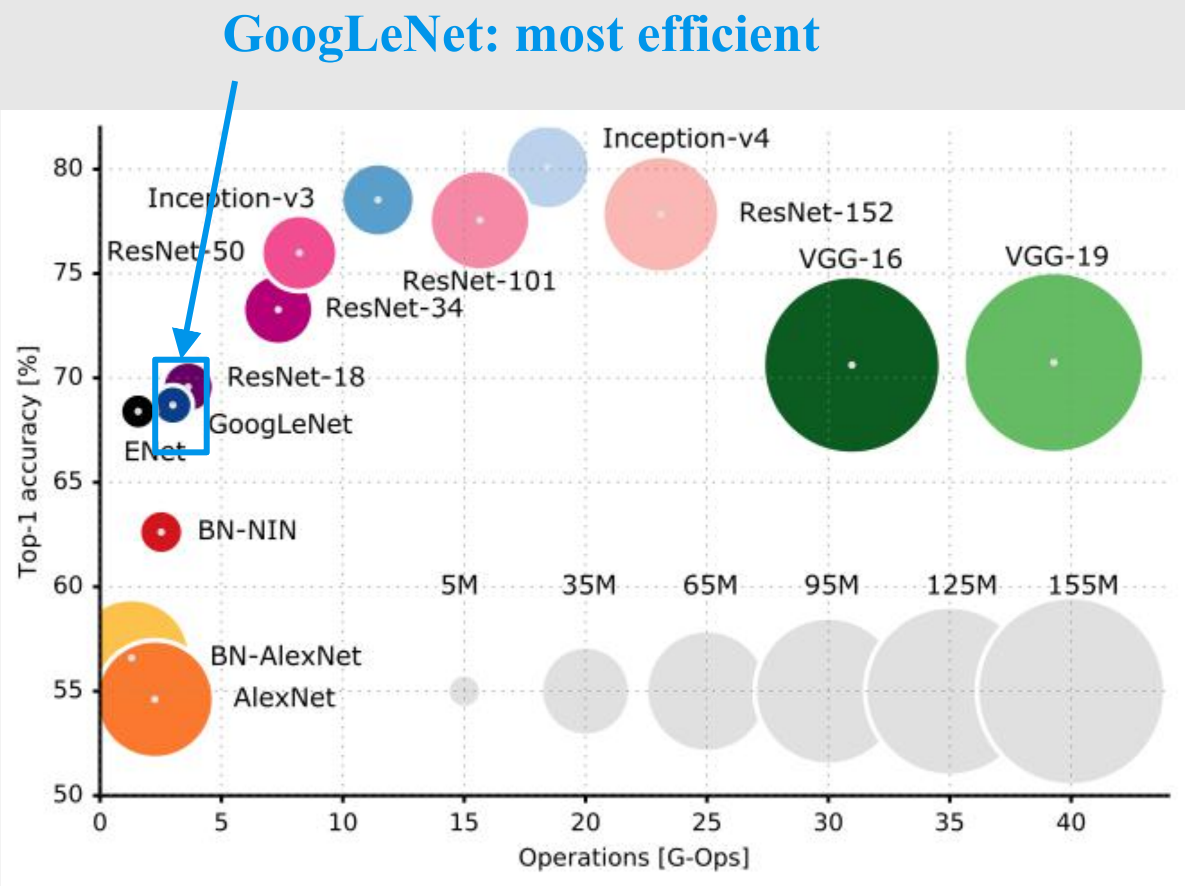
https://medium.com/towards-data-science/neural-network-architectures-156e5bad51ba

VGG: Highest memory, most operations



The size of the blobs is proportional to the number of network parameters.

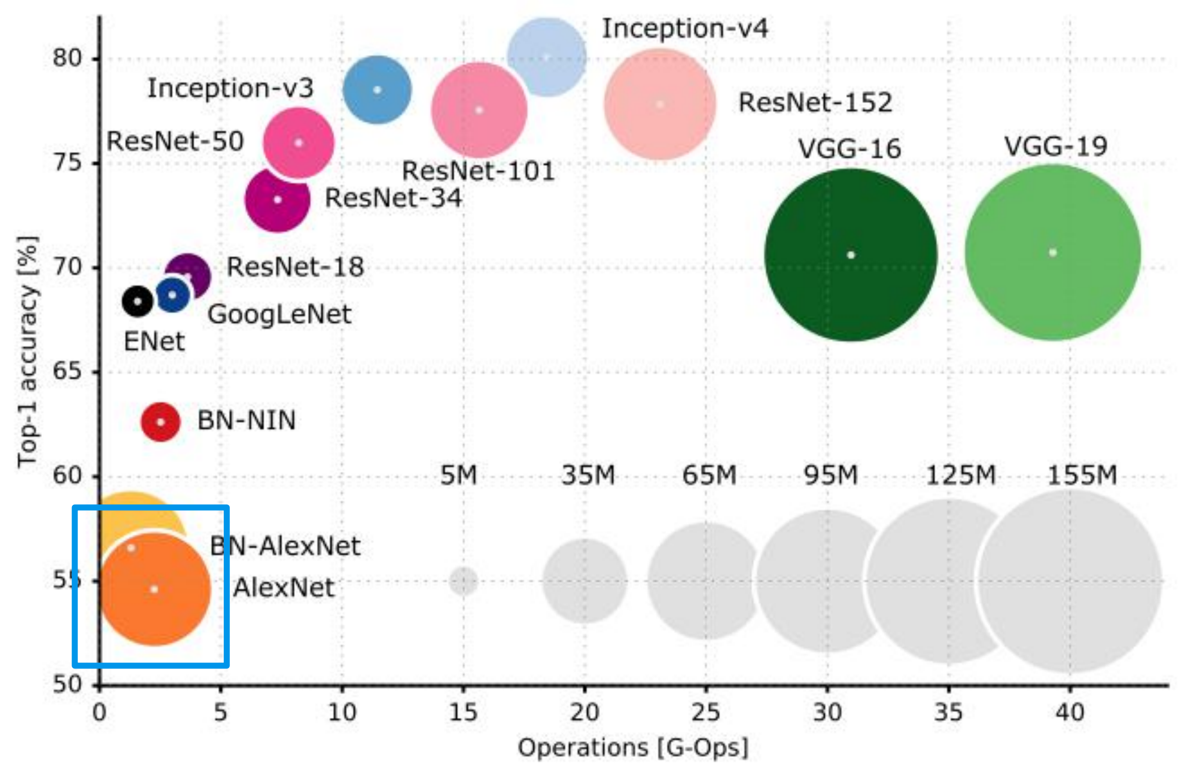
https://medium.com/towards-data-science/neural-network-architectures-156e5bad51ba



The size of the blobs is proportional to the number of network parameters.

https://medium.com/towards-data-science/neural-network-architectures-156e5bad51ba

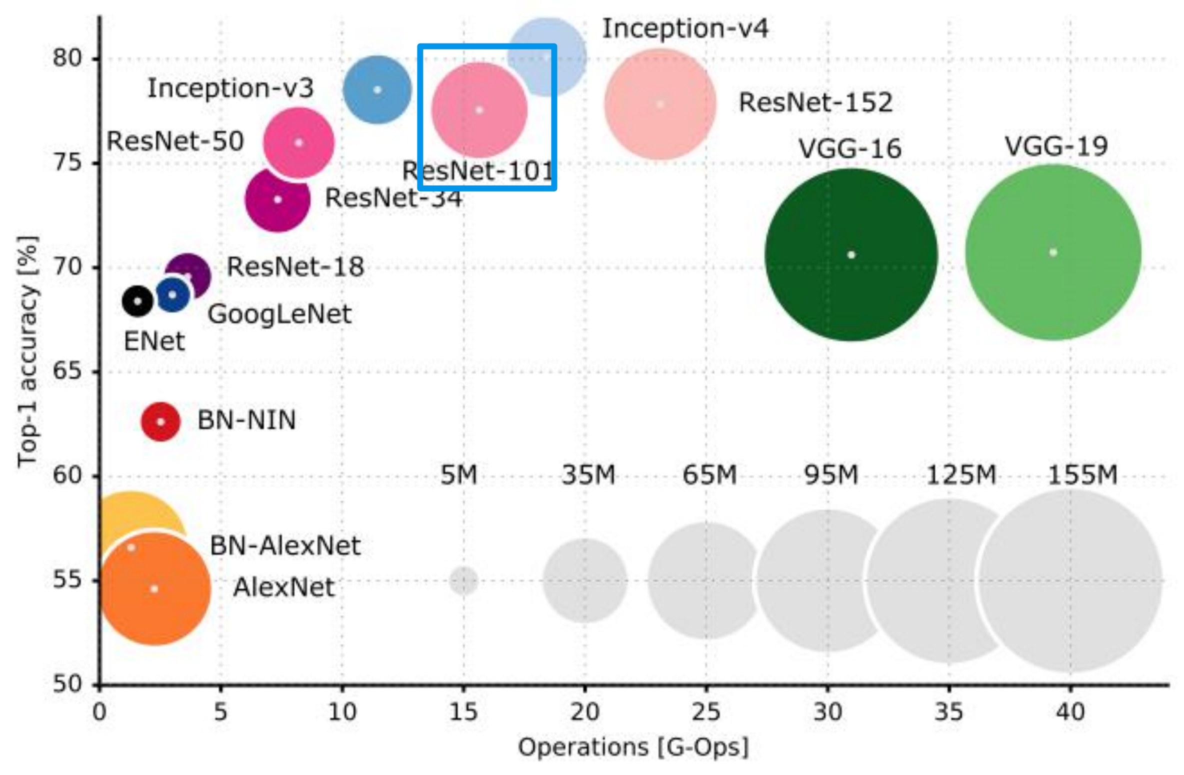
AlexNet: Smaller compute, still memory heavy, lower accuracy



The size of the blobs is proportional to the number of network parameters.

https://medium.com/towards-data-science/neural-network-architectures-156e5bad51ba

ResNet: Moderate efficiency depending on model, highest accuracy



The size of the blobs is proportional to the number of network parameters.

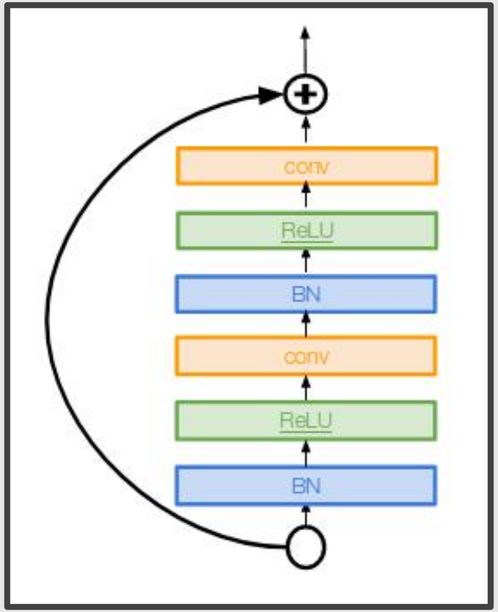
https://medium.com/towards-data-science/neural-network-architectures-156e5bad51ba

Other DNNs Architectures

Improving ResNet ...

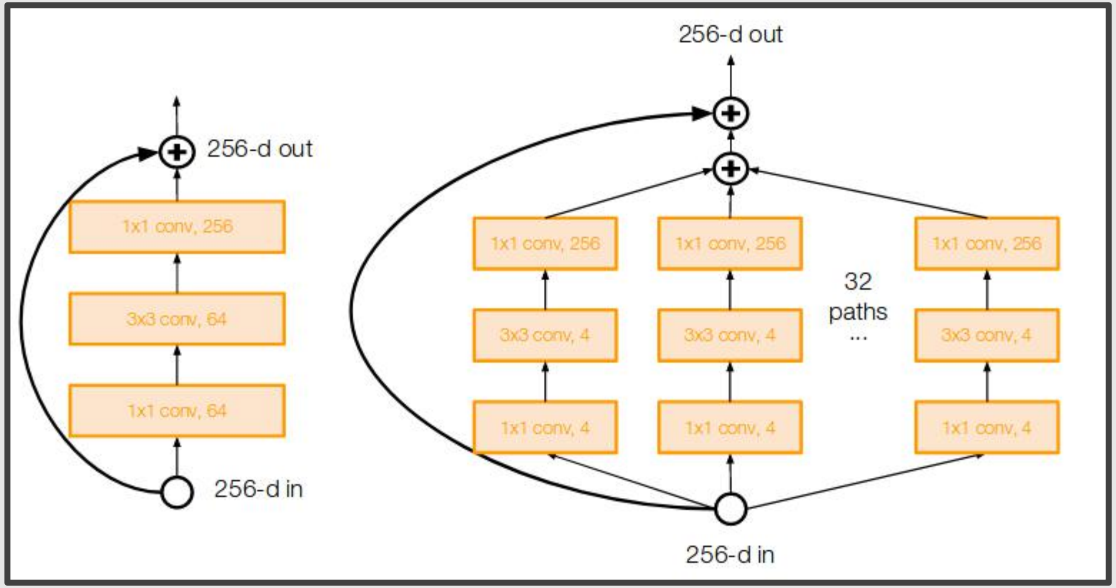
Identity Mappings in Deep Residual Networks [He et al., 2016]

- 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



Improving ResNet ...

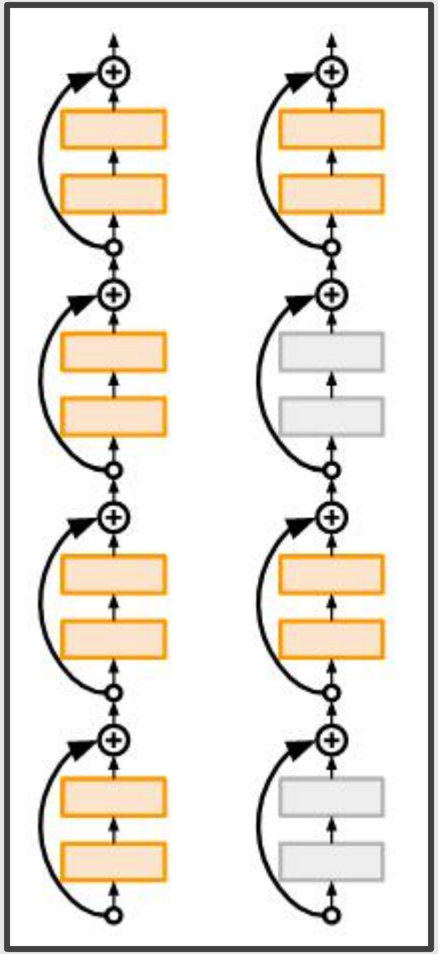
Aggregated Residual Transformations for Deep Neural Networks (**ResNeXt**) [Xie et al., 2016]



Improving ResNet ...

Deep Networks with Stochastic Depth [Huang et al., 2016]

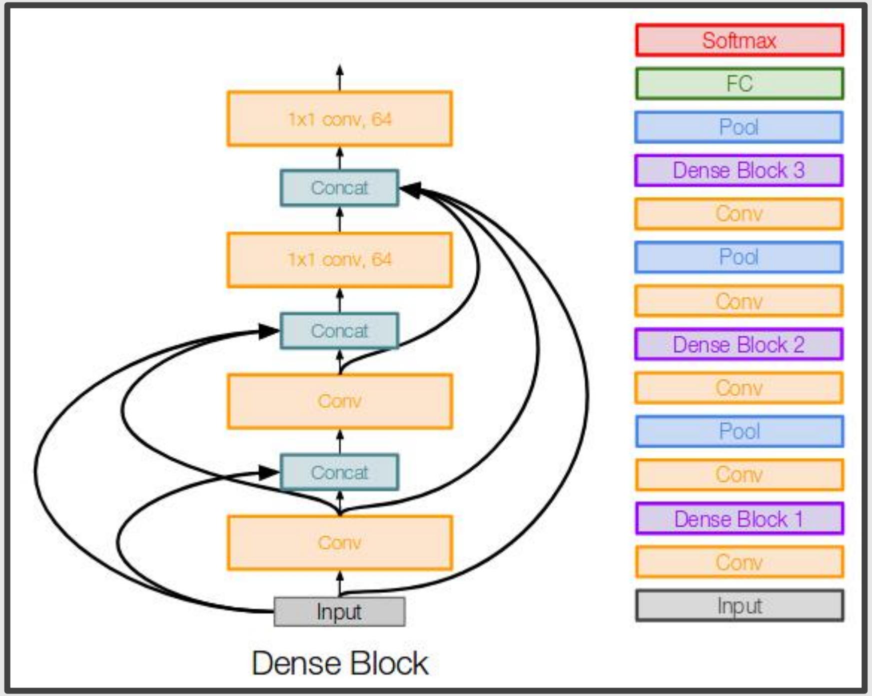
- Motivation: reduce vanishing gradients
- Randomly drop a subset of layers during each training pass
- Bypass with identity function



Beyond ResNet ...

Densely Connected Convolutional Networks (**DenseNet**) [Huang et al., 2017]

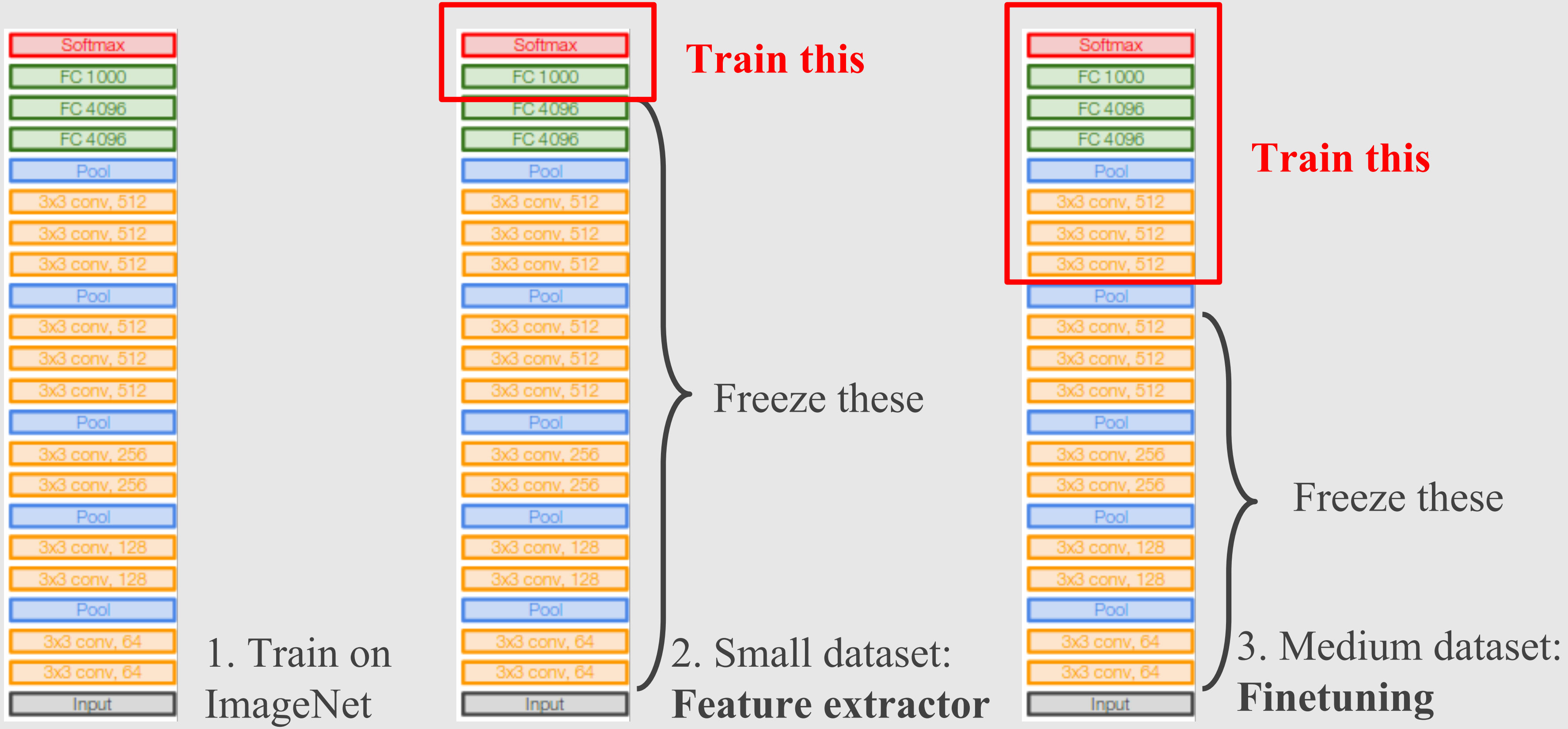
- Dense blocks where each layer is connected to every other layer in feedforward fashion
- Alleviates vanishing gradient, strengthens feature propagation, encourages feature reuse



Summary

- VGG, GoogLeNet, ResNet all in wide use, available in model zoos
- ResNet current best default
- Trend towards extremely deep networks
- Significant research centers around design of layer / skip connections and improving gradient flow
- Even more recent trend towards examining necessity of depth vs. width and residual connections

Transfer Learning with CNNs



To be continued ...

References

Machine Learning Books

- Hands-On Machine Learning with Scikit-Learn and TensorFlow, Chap. 11 & 13

Machine Learning Courses

- https://www.coursera.org/learn/neural-networks
- “The 3 popular courses on Deep Learning”: https://medium.com/towards-data-science/the-3-popular-courses-for-deeplearning-ai-ac37d4433bd