Aprendizado Profundo (Deep Learning)

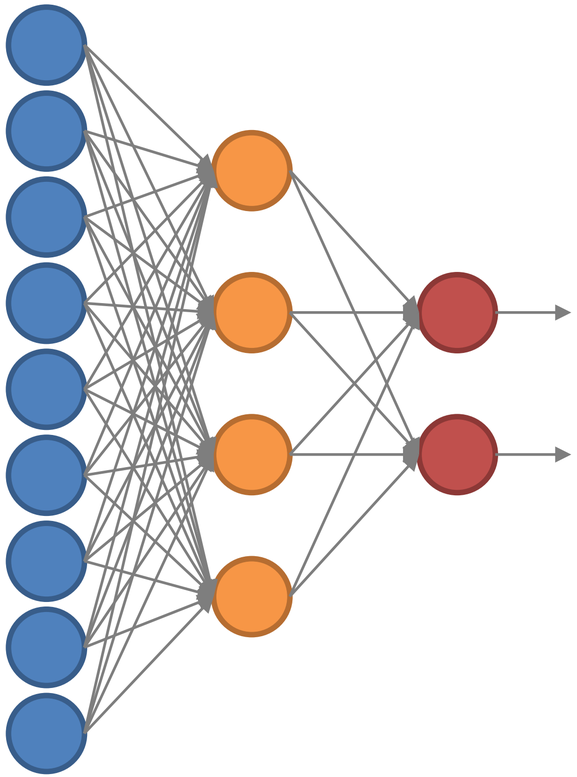
CNN Architectures

Dario Oliveira (dario.oliveira@fgv.br)

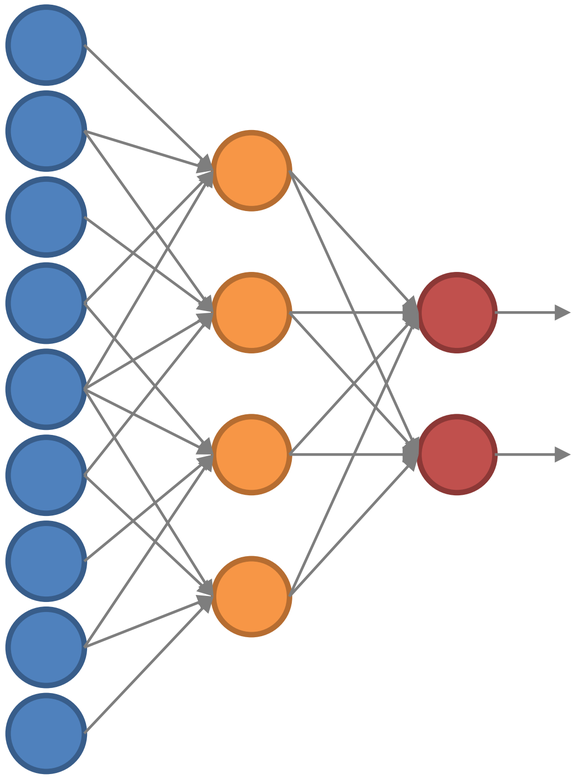
CNN Recap

MLP to CNN

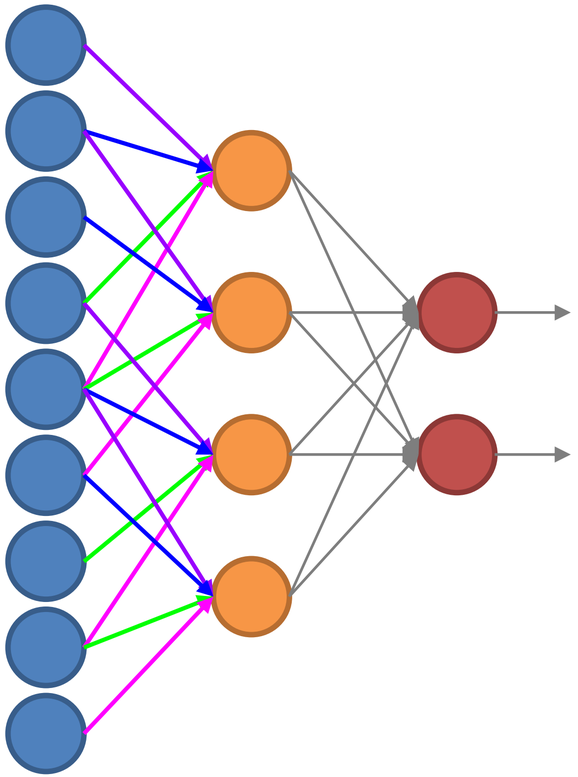
Standard MLP



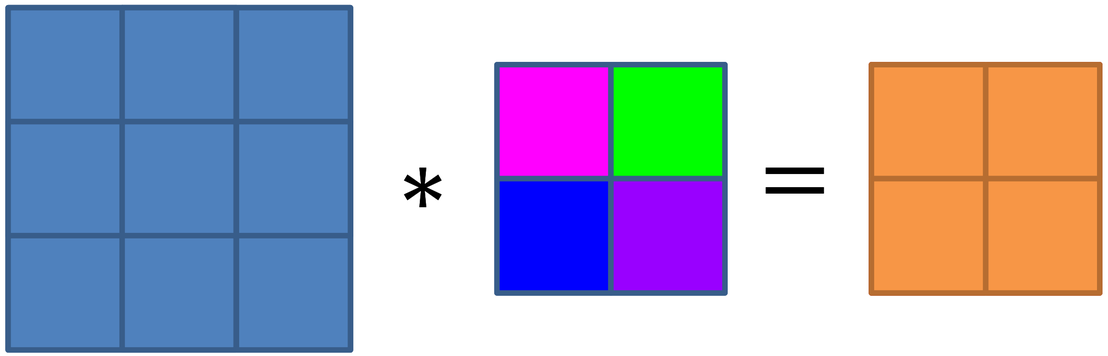
Cutting connections

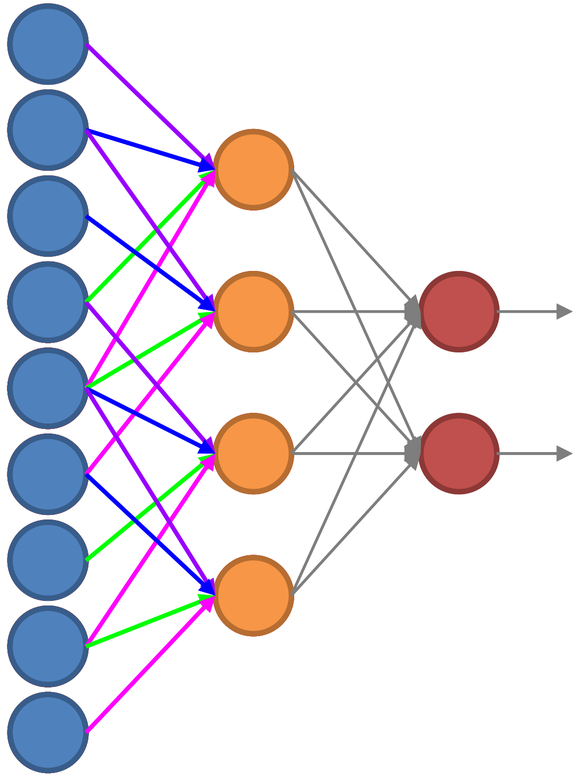


Sharing weights



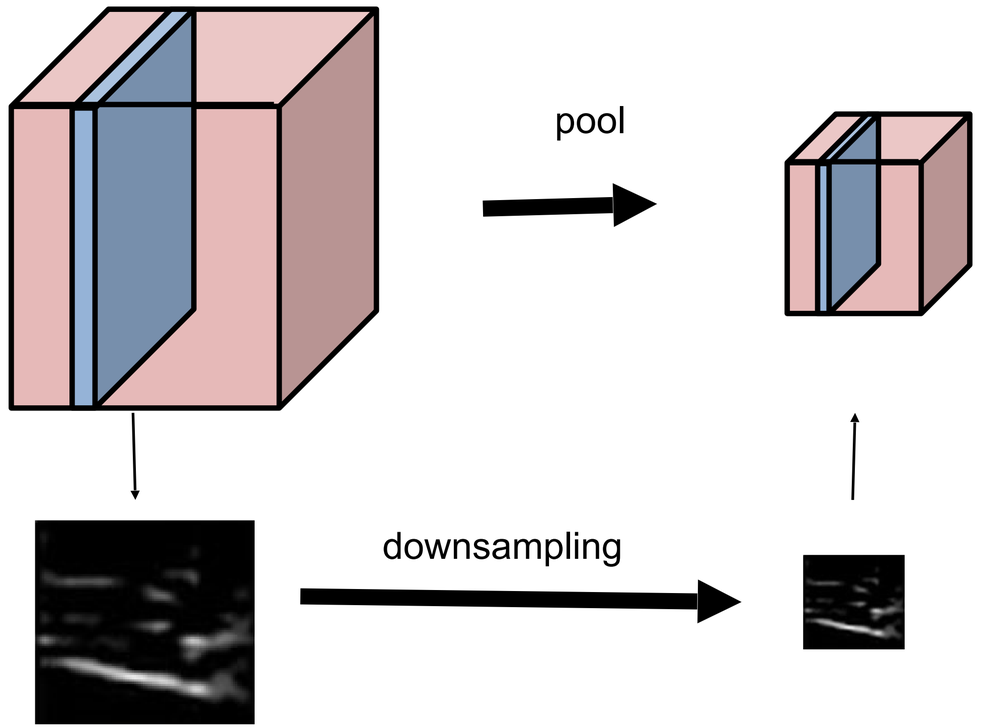
MLP to CNN



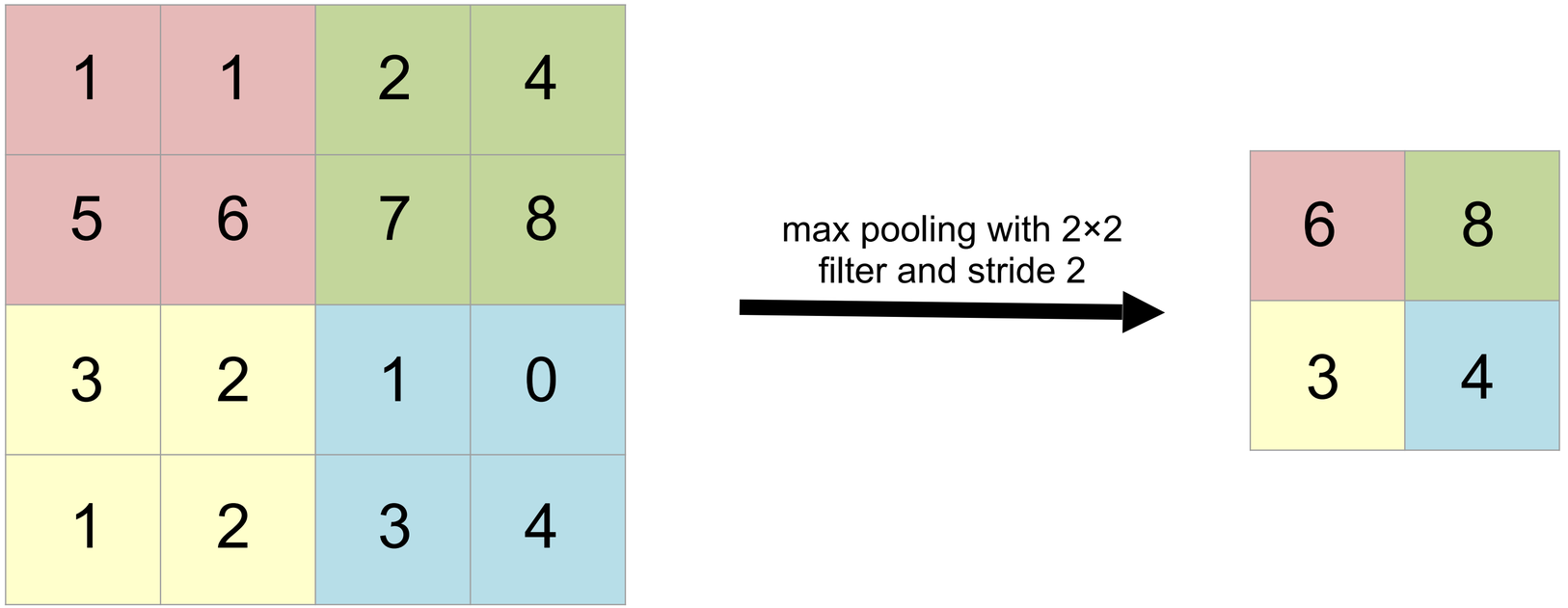


Pooling

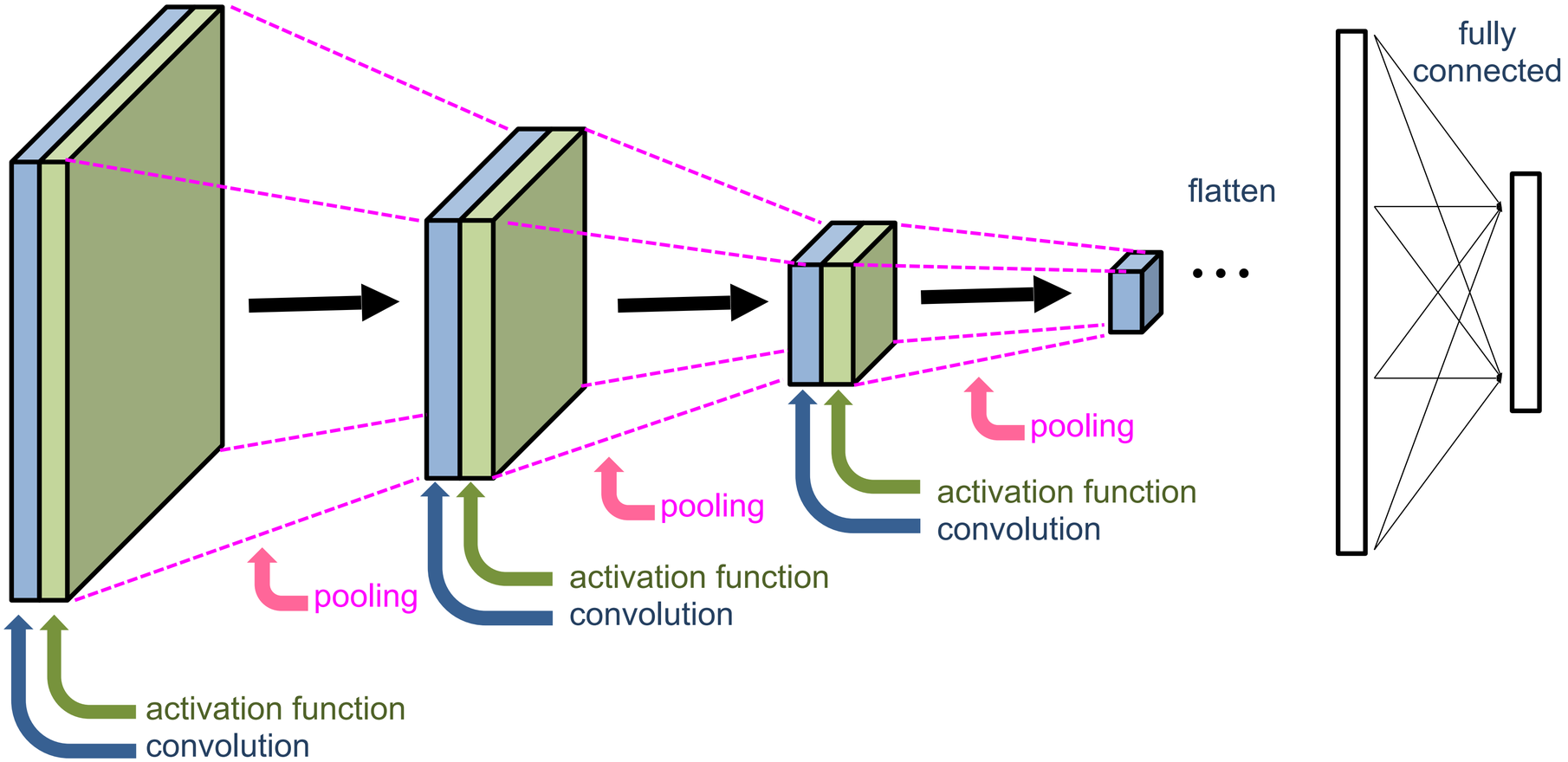
- replaces the output of the net at a certain location with a summary statistic of the nearby outputs.
- In all cases, pooling helps to make the representation become approximately *invariant* to small translations of the input.



Max Pooling


1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
max pooling with 2×2
filter and stride 2
6 8
3 4

Typical ConvNet for Image Classification



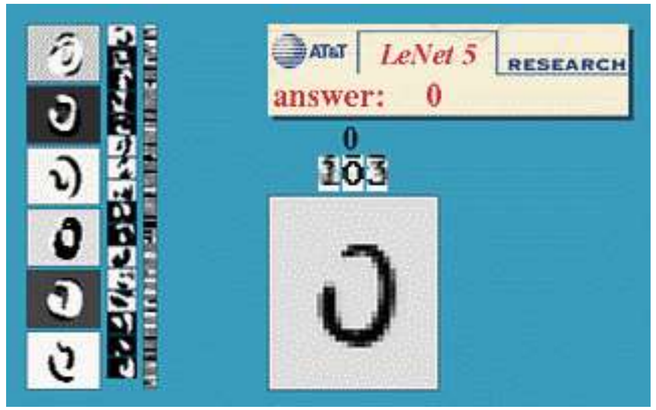
Content

- LeNet
- AlexNet
- ZFnet
- VGGNet
- GoogLeNet
- ResNet
- DenseNet

LeNet-5

- LeNet-5: pioneering 7-level convolutional network by LeCun et al. in 1998.
- Classifies digits: was applied by several banks to recognize hand-written numbers on checks digitized in 32x32 pixel greyscale input images.
- Error rate < 1% on MNIST.

Image from http://yann.lecun.com/exdb/lenet/



LeCun et al., 1998, Gradient-Based Learning Applied to Document Recognition

LeNet-5

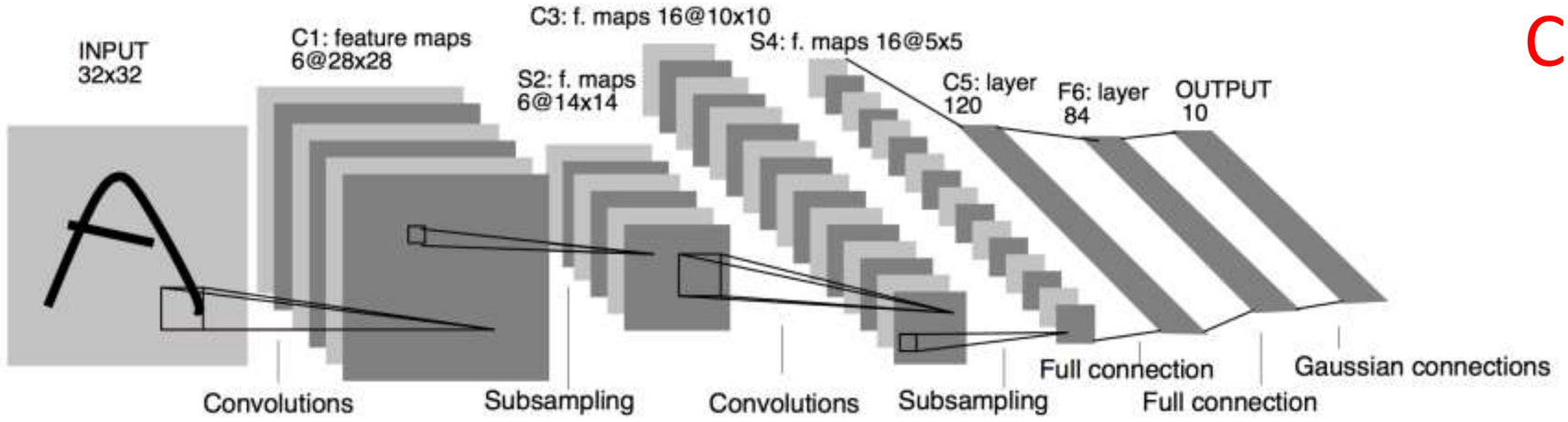
- Before LeNet: character recognition had been done mostly by using hand-crafted features, followed by a machine learning model.
- LeNet made hand engineering features redundant: the network learns the best internal representation from raw images.

LeCun et al., 1998, Gradient-Based Learning Applied to Document Recognition

LeNet-5

- Three 5×5 convolutional layers, *tanh* activations.
- Two 2×2 average pooling layers.
- Two fully connected layers, *tanh* activations.

CONV1
AVGPOOL2
CONV3
AVGPOOL4
CONV5
FC6
FC7



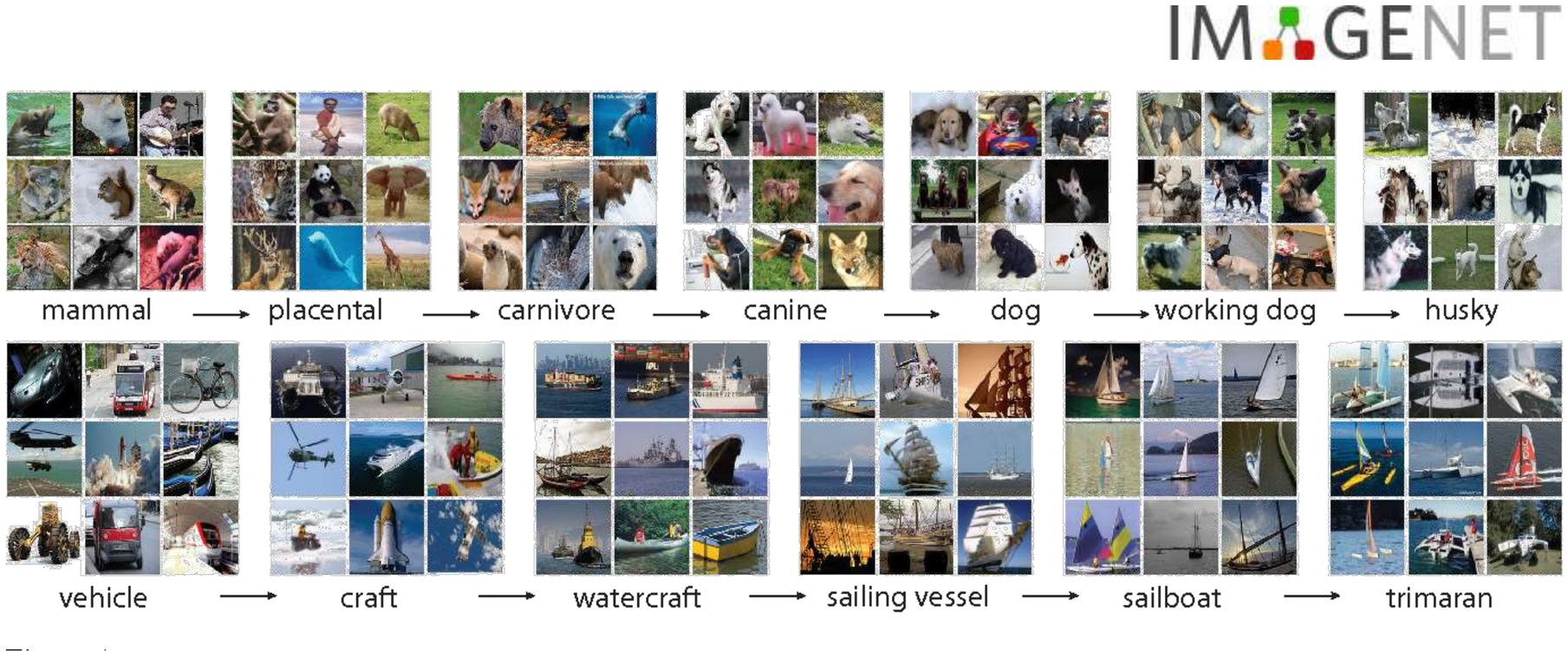
LeCun et al., 1998, Gradient-Based Learning Applied to Document Recognition

Content

- LeNet
- AlexNet
- ZFnet
- VGGNet
- GoogLeNet
- ResNet
- DenseNet

ImageNet Challenge

- A large visual database for object recognition research.
- Active until 2016.

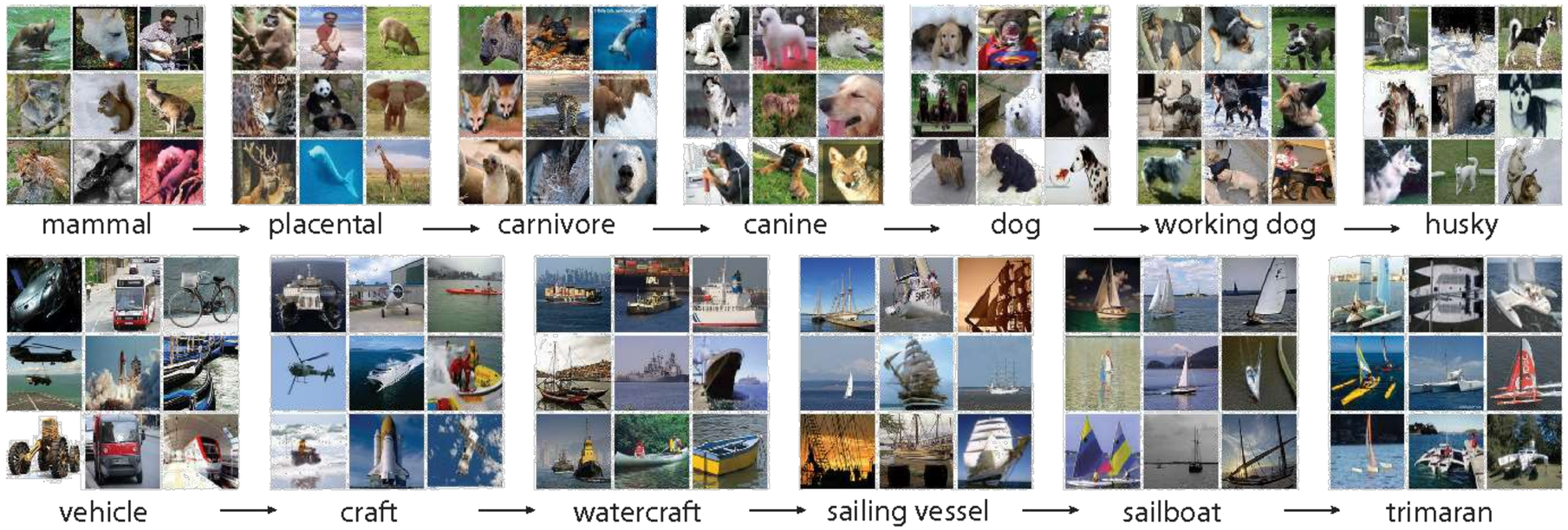


ImageNet: http://www.image-net.org/

ImageNet Challenge

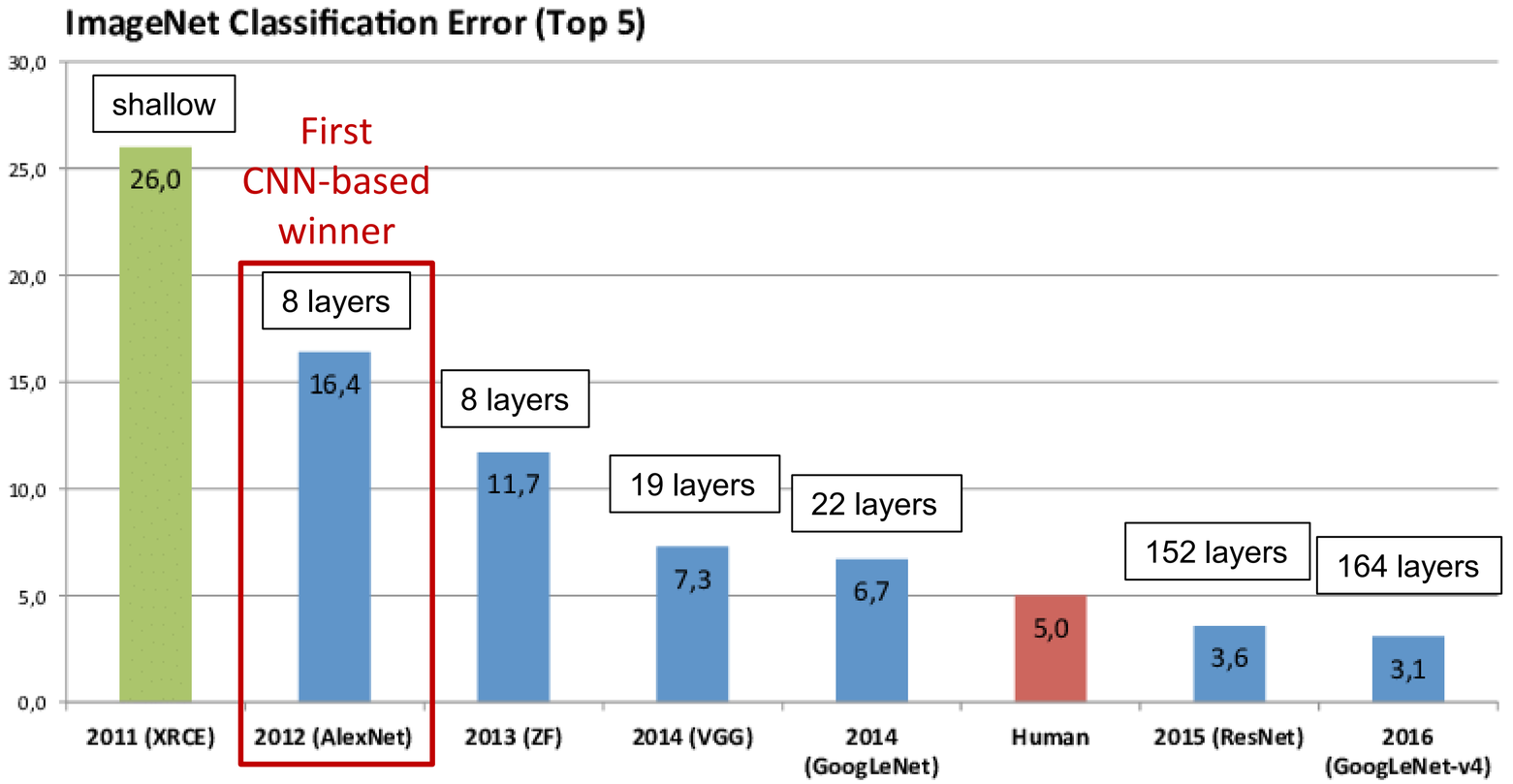
- > 14 million hand annotated images.
- ~1 M, 50K, 100K training, validation and testing images.
- 1000 categories.

IMAGENET



ImageNet: http://www.image-net.org/

ImageNet Challenge



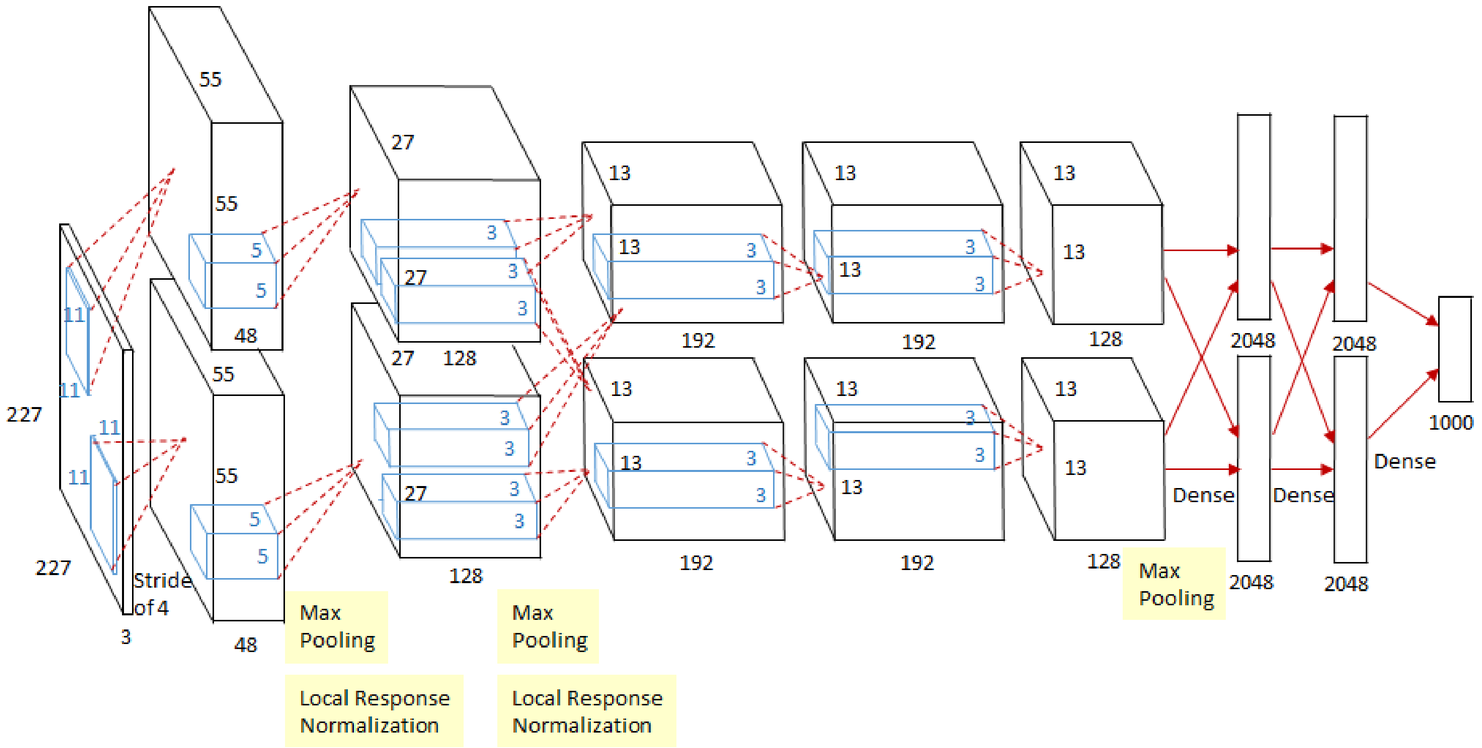
AlexNet: marked the start of an industry-wide artificial intelligence boom!

https://www.researchgate.net/figure/Winner-results-of-the-ImageNet-large-scale-visual-recognition-challenge-LSVRC-of-the_fig7_324476862

AlexNet

- ReLU instead of *tanh* → 6 times faster.
- Dropout (0.5) before FC6 and FC7.

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
MAX POOL3
FC6
FC7
FC8

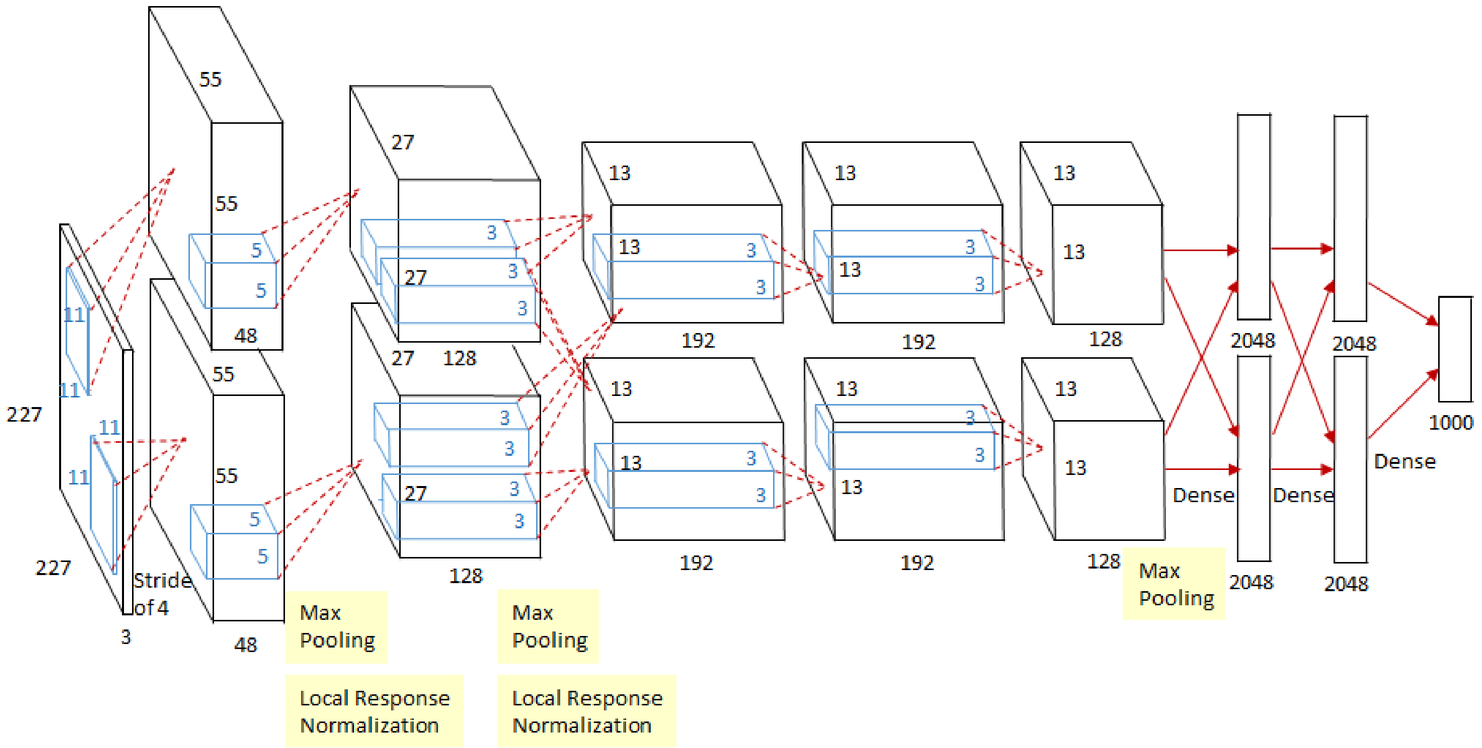


Krizhvsky et al., 2012, ImageNet Classification with Deep Convolutional Neural Networks

AlexNet

- ReLU instead of *tanh* → 6 times faster.
- Dropout (0.5) before FC6 and FC7.

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
MAX POOL3
FC6
FC7
FC8

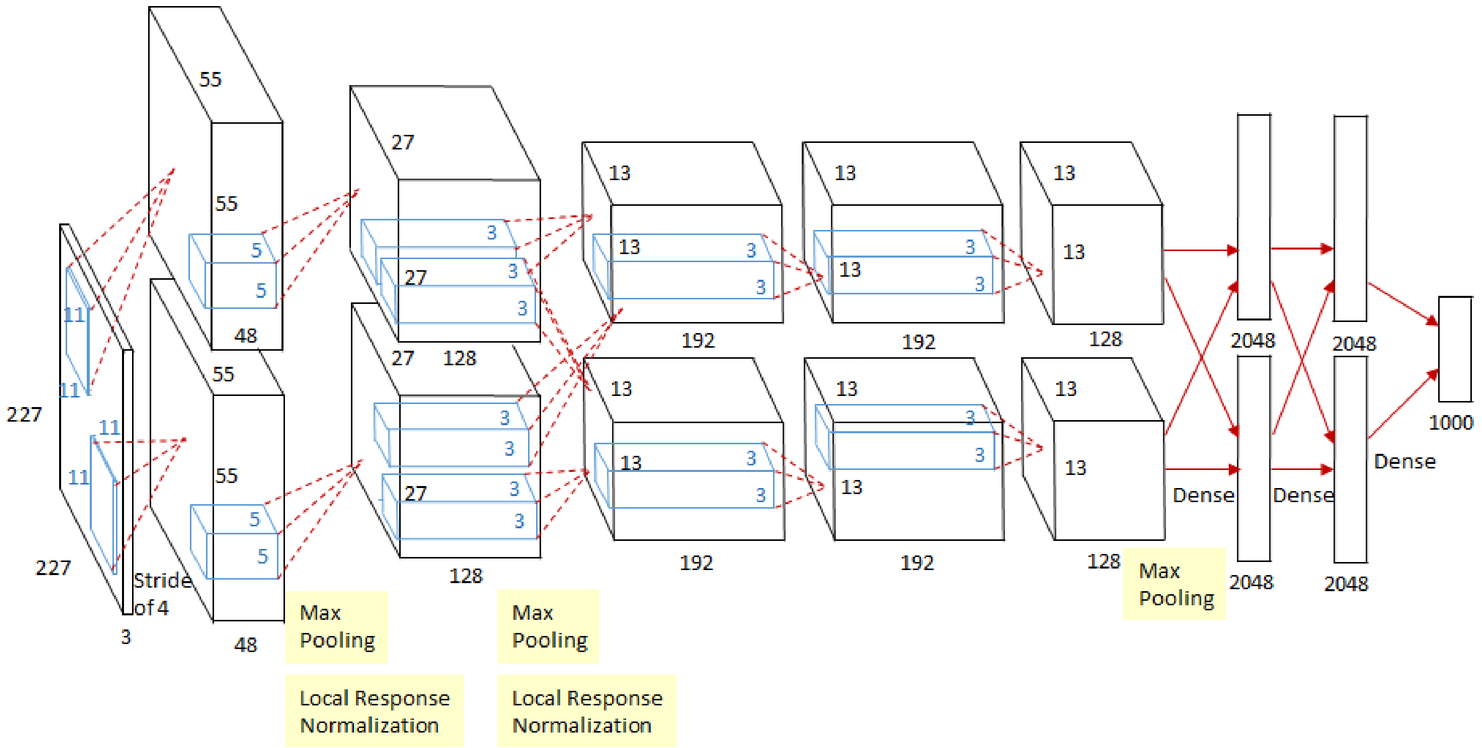


Krizhvsky et al., 2012, ImageNet Classification with Deep Convolutional Neural Networks

AlexNet

- 62.3 million parameters: 1.1 billion computation units in a forward pass!

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
MAX POOL3
FC6
FC7
FC8



Krizhvsky et al., 2012, ImageNet Classification with Deep Convolutional Neural Networks

AlexNet

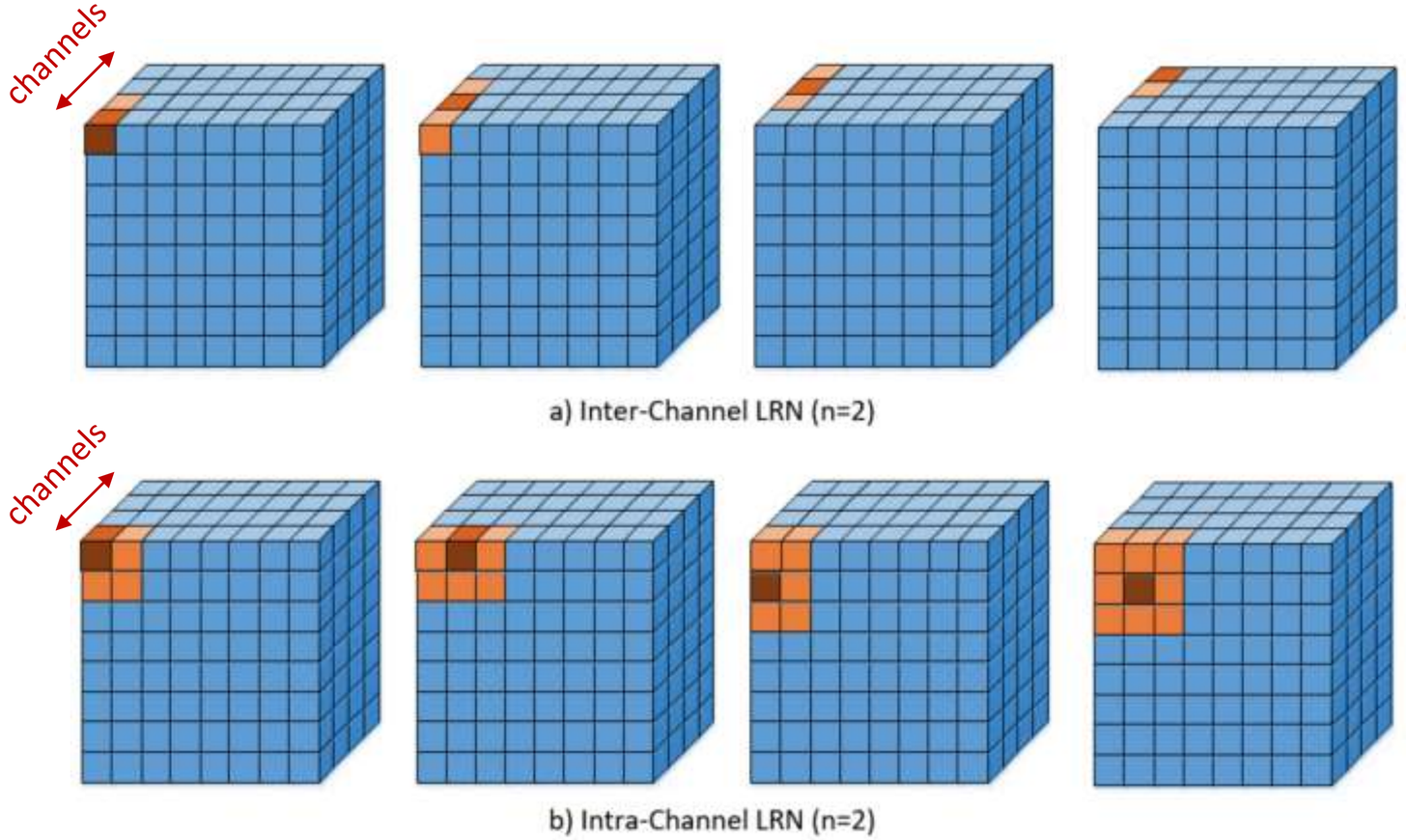
- Normalization to limit the unbounded activation from increasing the output layer values.
- Used just before the activation function (ReLU).
- AlexNet introduced Local Response Normalization (LRN).
- LRN is a non-trainable layer that square-normalizes the pixel values in feature maps within a local neighborhood.

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^{\beta}$$

Krizhvsky et al., 2012, ImageNet Classification with Deep Convolutional Neural Networks

AlexNet

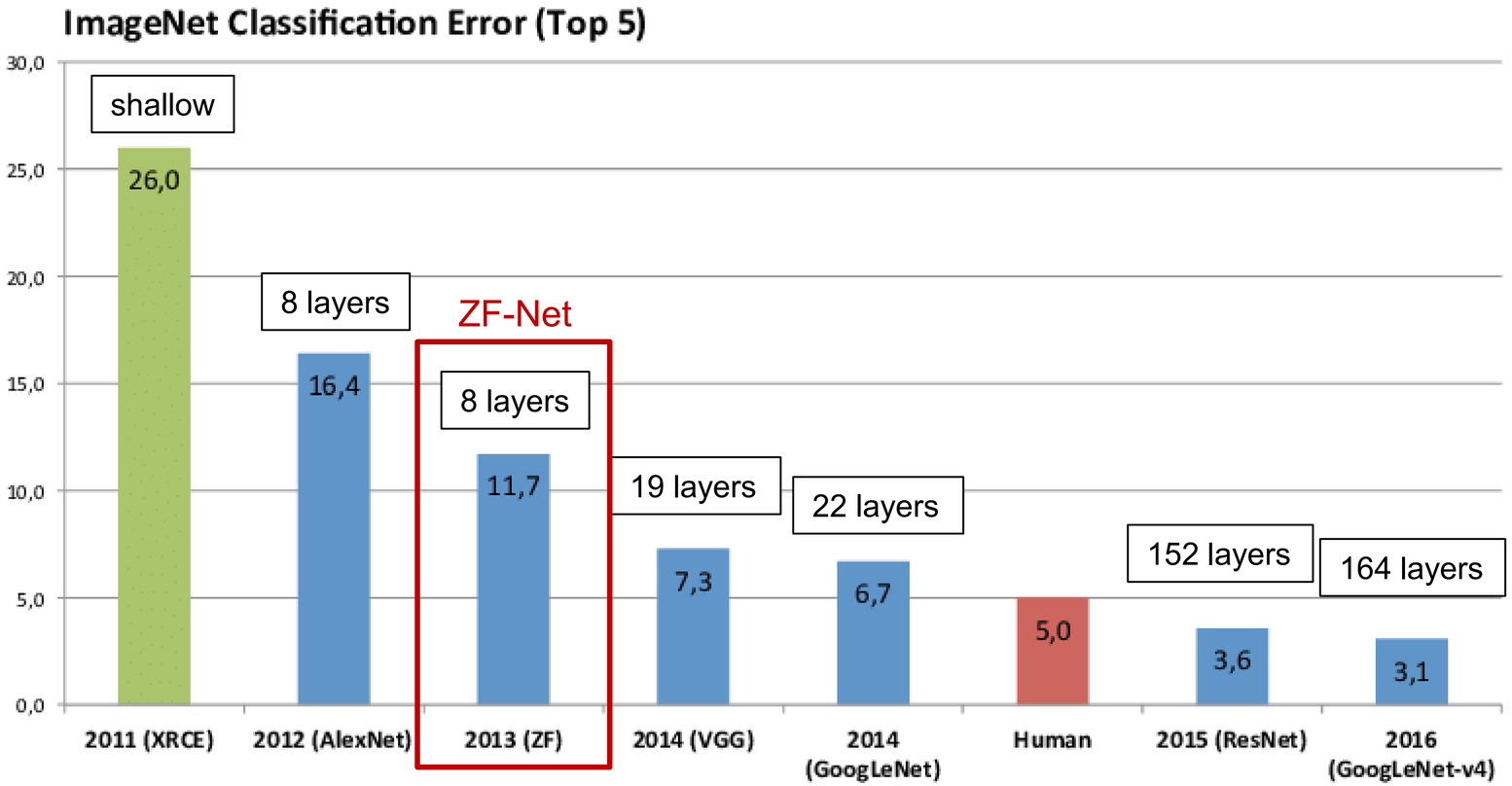
Two types of LRN based on the neighborhood: inter-channel (used in AlexNet) and intra-channel:



Content

- LeNet
- AlexNet
- ZFnet
- VGGNet
- GoogLeNet
- ResNet
- DenseNet

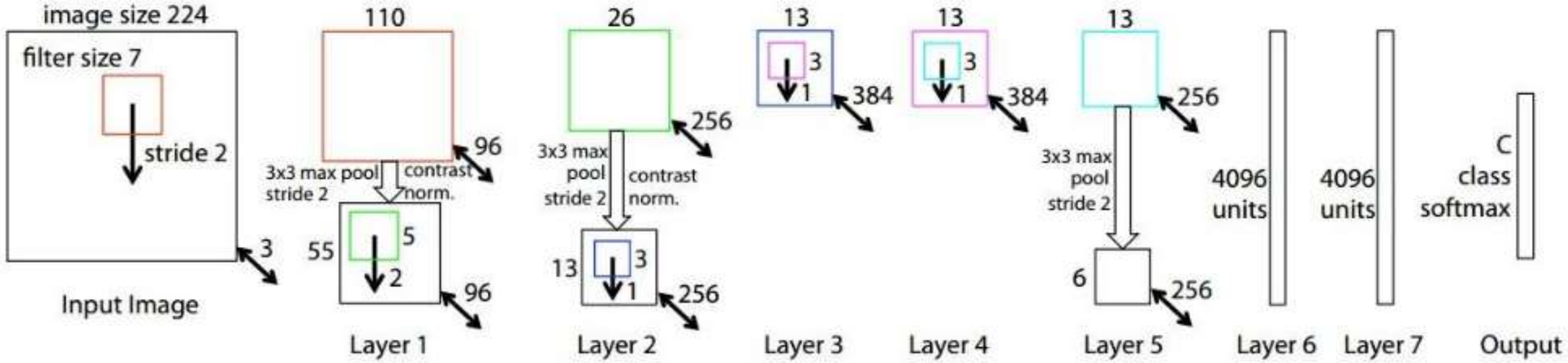
ImageNet Challenge



https://www.researchgate.net/figure/Winner-results-of-the-ImageNet-large-scale-visual-recognition-challenge-LSVRC-of-the_fig7_324476862

ZFNet

Just improved AlexNet hyperparameters (stride sizes, numbers of filters, normalization).



Zeiler and Fergus, 2013: https://cs.nyu.edu/~fergus/papers/zeilerECCV2014.pdf

ZFNet

Just improved AlexNet hyperparameters (stride sizes, numbers of filters, normalization).

Local Contrast Normalization: enforces local competition between adjacent features in a feature map, and between features at the same spatial location in different feature maps.

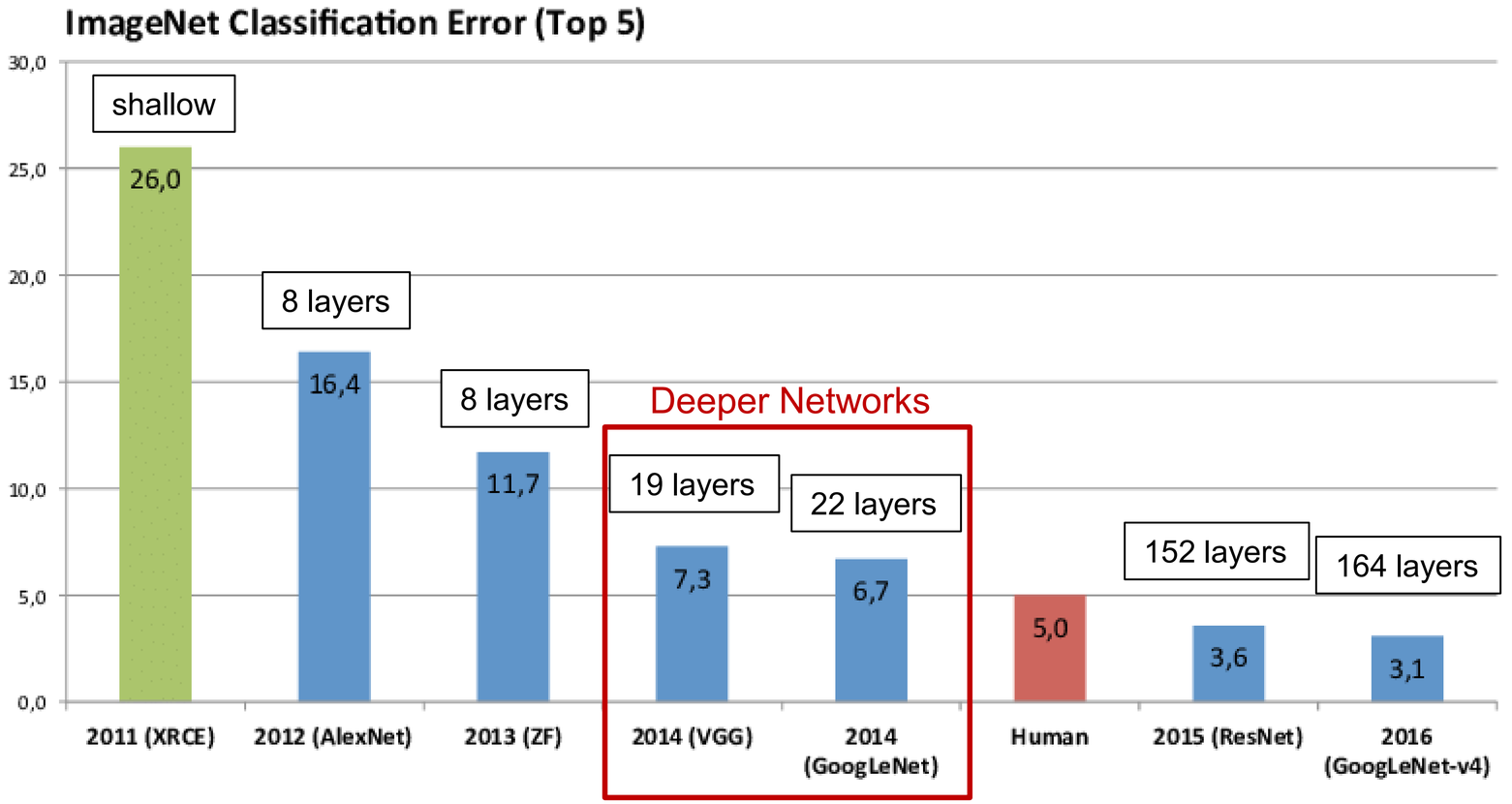
$$b_{x,y}^{i} = \sqrt{\frac{1}{d \times n} \sum_{k=1}^{d} \sum_{j=i-n/2}^{i+n/2} \left(a_{x,y}^{j} - \bar{a}\right)^2} \qquad \bar{a} = \frac{1}{d \times n} \sum_{k=1}^{d} \sum_{j=i-n/2}^{i+n/2} a_{x,y}^{j}$$

Zeiler and Fergus, 2013: https://cs.nyu.edu/~fergus/papers/zeilerECCV2014.pdf

Content

- LeNet
- AlexNet
- ZFnet
- VGGNet
- GoogLeNet
- ResNet
- DenseNet

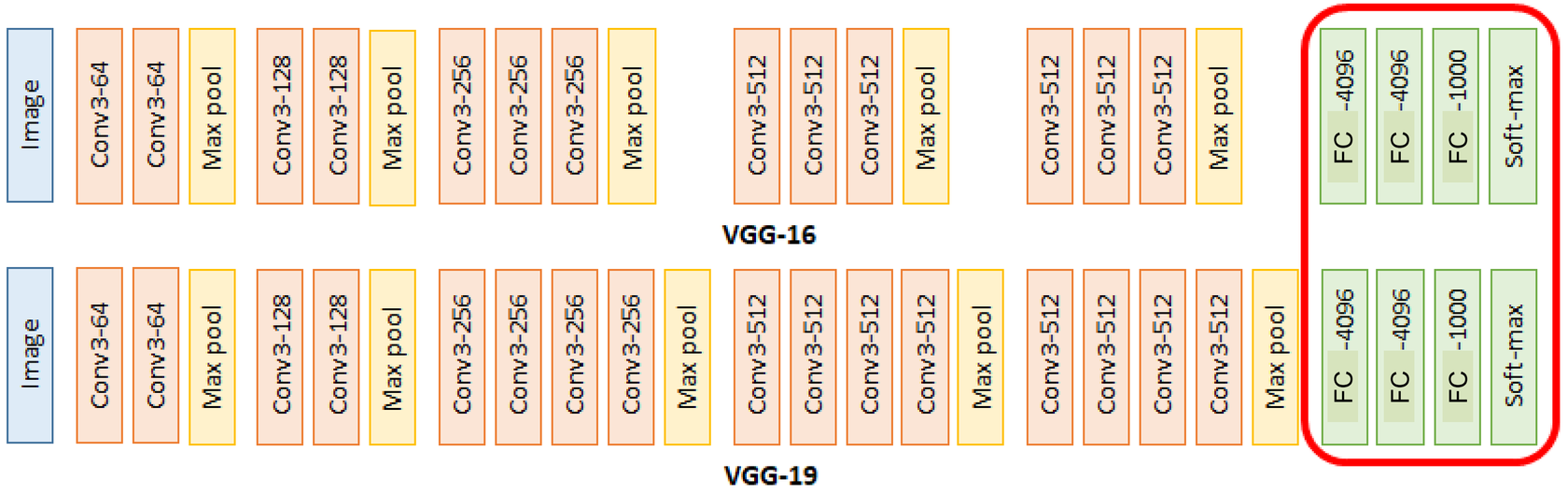
ImageNet Challenge



https://www.researchgate.net/figure/Winner-results-of-the-ImageNet-large-scale-visual-recognition-challenge-LSVRC-of-the_fig7_324476862

VGGNet

- Uniform architecture: 3×3 convs + 2×2 maxpooling.
- Variants with 16 (VGG-16) and 19 layers (VGG-19).



- Number of parameters = 138 million.

Figure from : https://www.quora.com/What-is-the-VGG-neural-network

VGGNet

- Still used as feature extractor (backbone).
- Still used for image classification (fine-tuning).

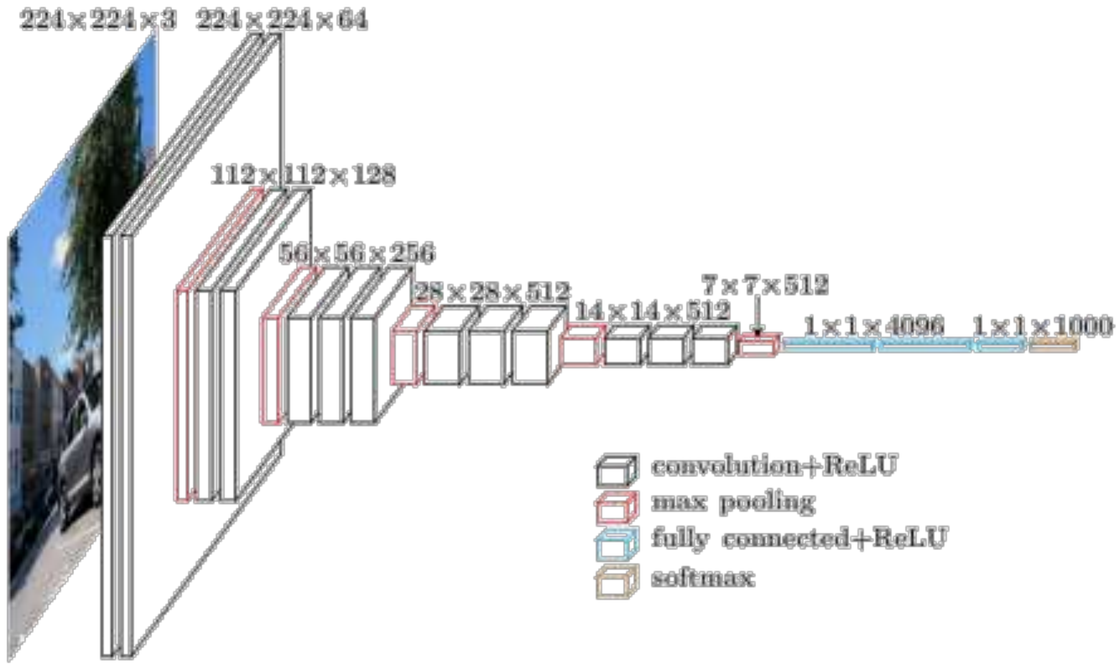


Figure from : https://www.quora.com/What-is-the-VGG-neural-network

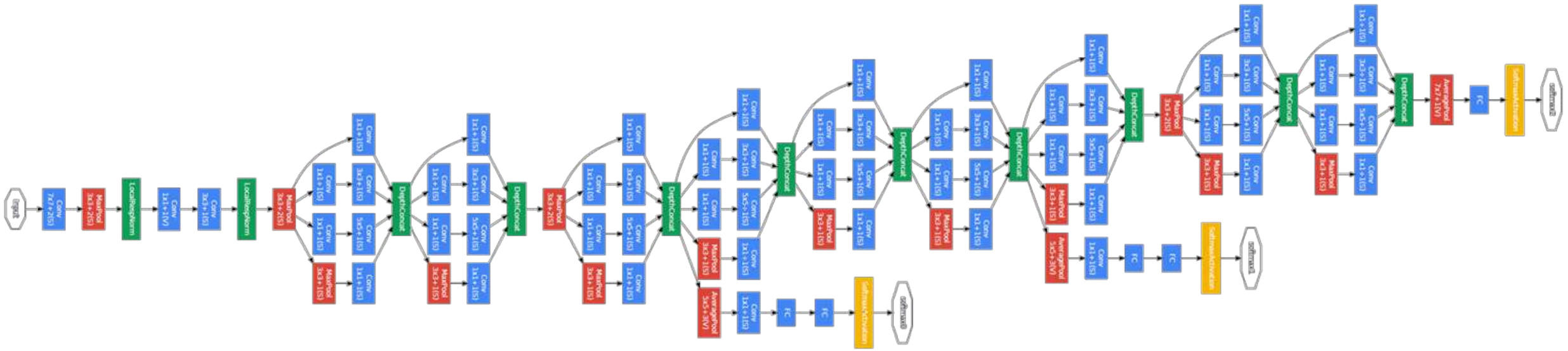
Content

- LeNet
- AlexNet
- ZFnet
- VGGNet
- GoogLeNet
- ResNet
- DenseNet

GoogLeNet (Inception v1)

- Until then, just stacking layers deeper and deeper.
- GoogLeNet introduced tricks to push speed and accuracy.
- Involves several versions:
  - Inception v1 (aka GoogLeNet)
  - Inception v2 and Inception v3
  - Inception v4 and Inception ResNet

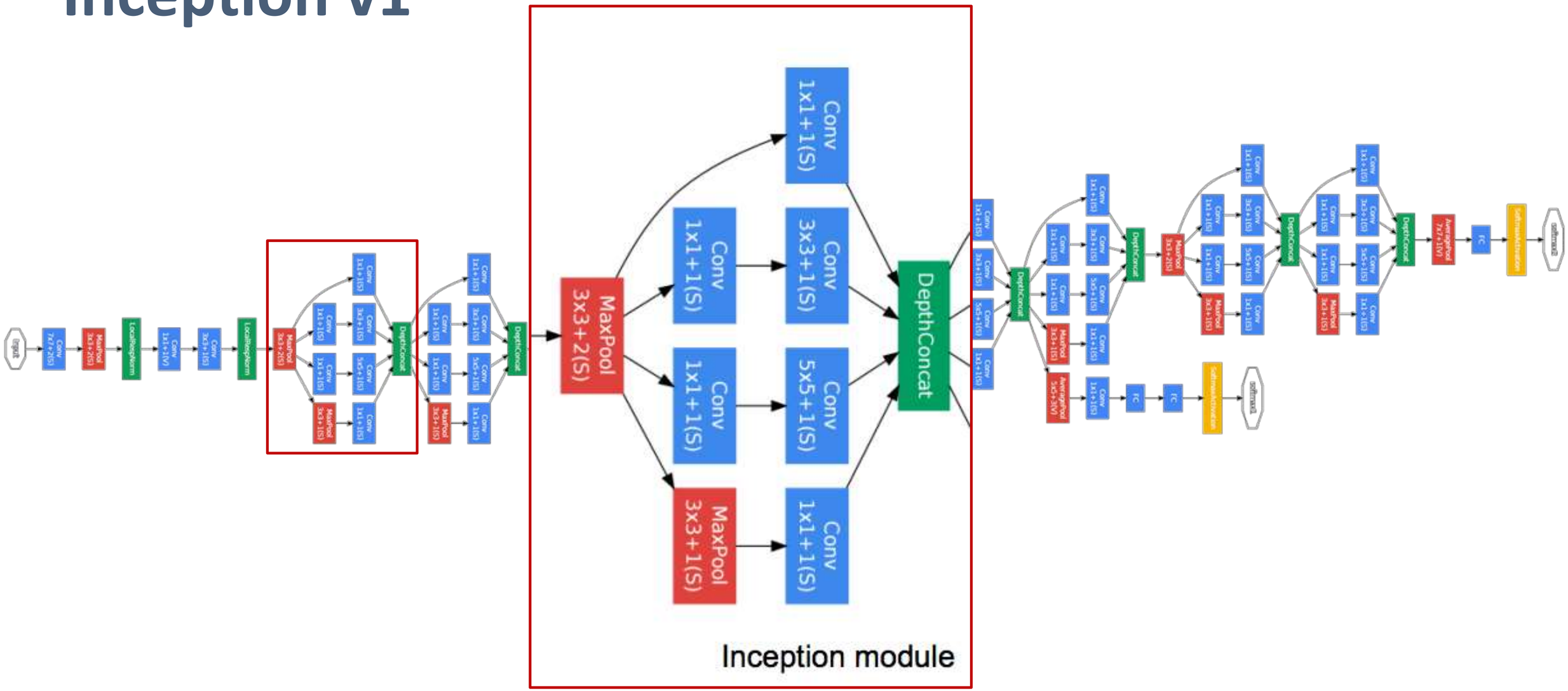
Inception v1



Convolution **Pooling** **Softmax** **Concat/Normalize**

Szegedy et al., 2014 Going deeper with convolutions, available at https://arxiv.org/pdf/1409.4842v1.pdf

Inception v1



Szegedy et al., 2014 Going deeper with convolutions, available at https://arxiv.org/pdf/1409.4842v1.pdf

Inception v1

Idea 1: filters of multiple sizes at the same module

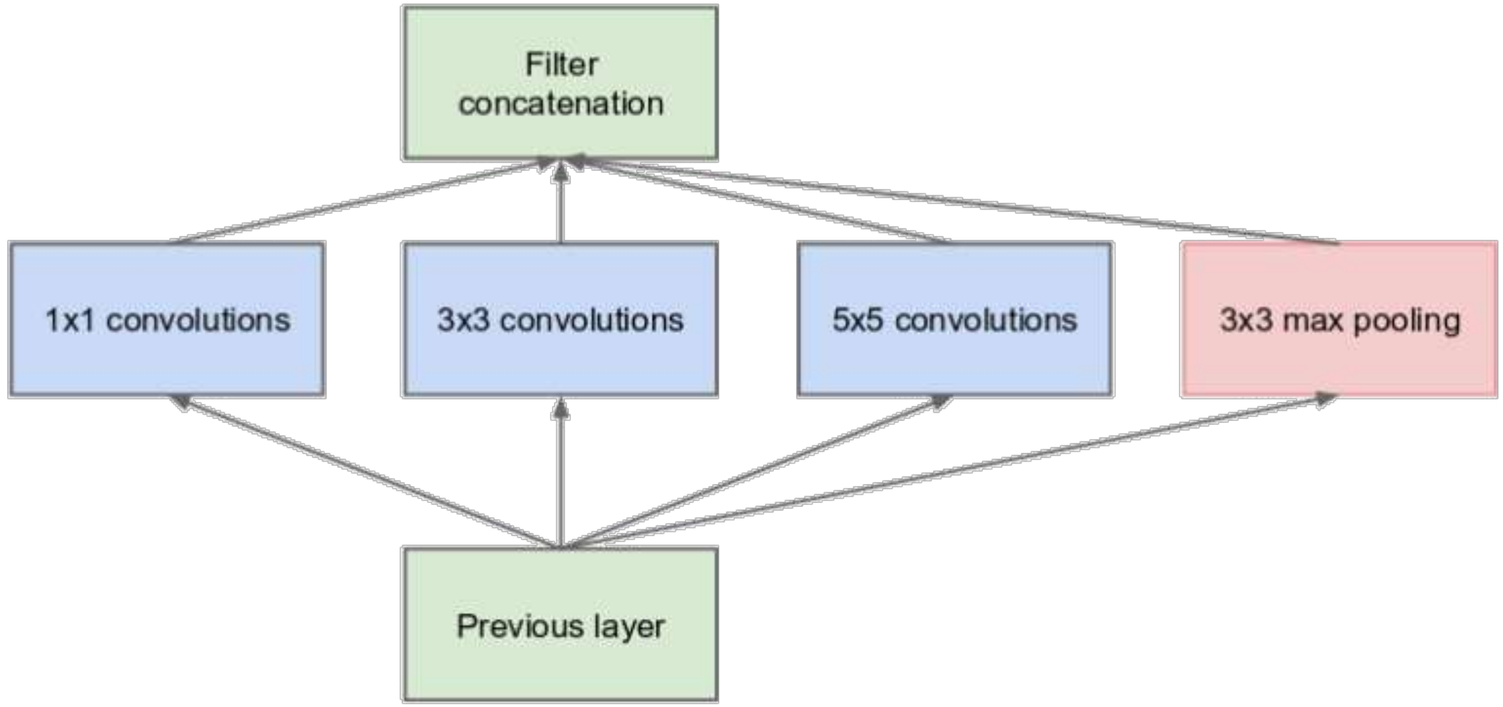
- Objects come about at different sizes/scales.
- Larger kernels good for information distributed globally.
- Smaller kernels good for information distributed locally.



Inception v1

Idea 1: filters of multiple sizes at the same layer

- Network gets “wider”.

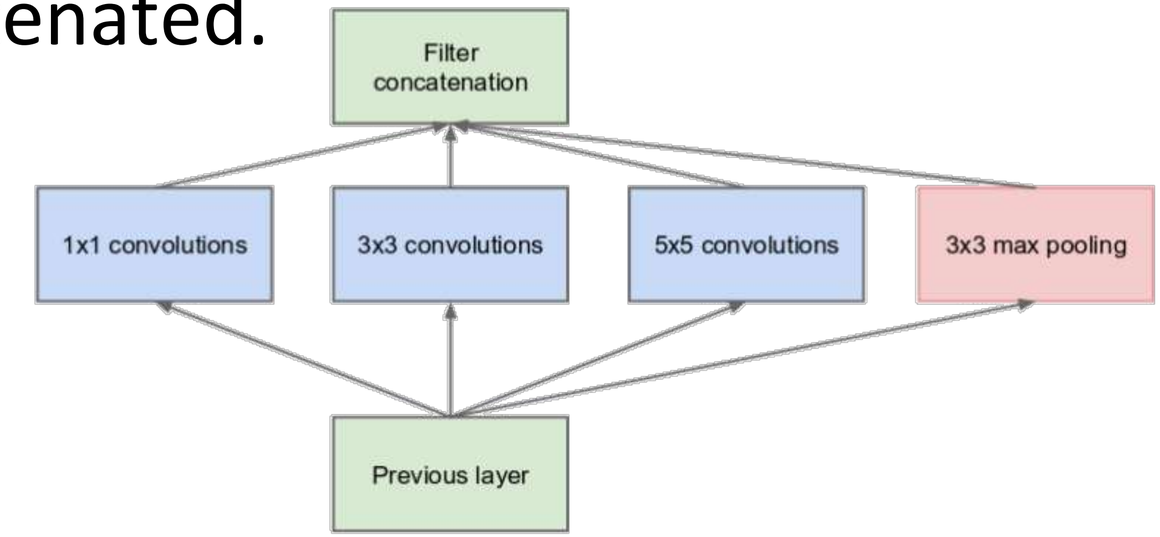


Näive inception module

Inception v1

Idea 1: filters of multiple sizes at the same layer

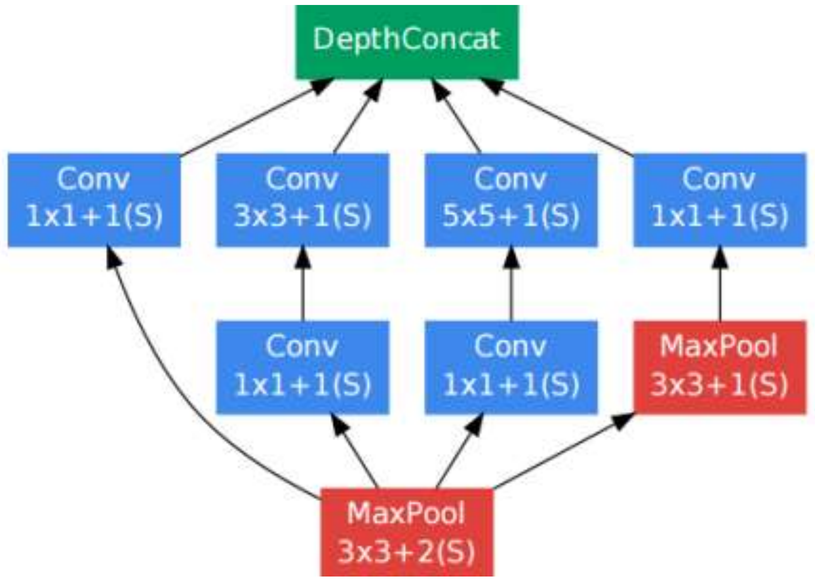
- The outputs of the operations preserve the spatial dimensions.
- For the convolutional layers: padding.
- For the maxpooling: stride of 1.
- The outputs are depth concatenated.



Inception v1

Idea 2: the 1x1 convolution

- Originally, aimed to add more non-linearity and increase representational power.
- Here: to reduce the number of parameters → computational demand → overfitting.



Inception v1

Without 1x1 convolution:

Number of parameters = (48×5×5×480) = **576K**

Number of operations = 576K×14×14 = **112.9M**

With 1x1 convolution:

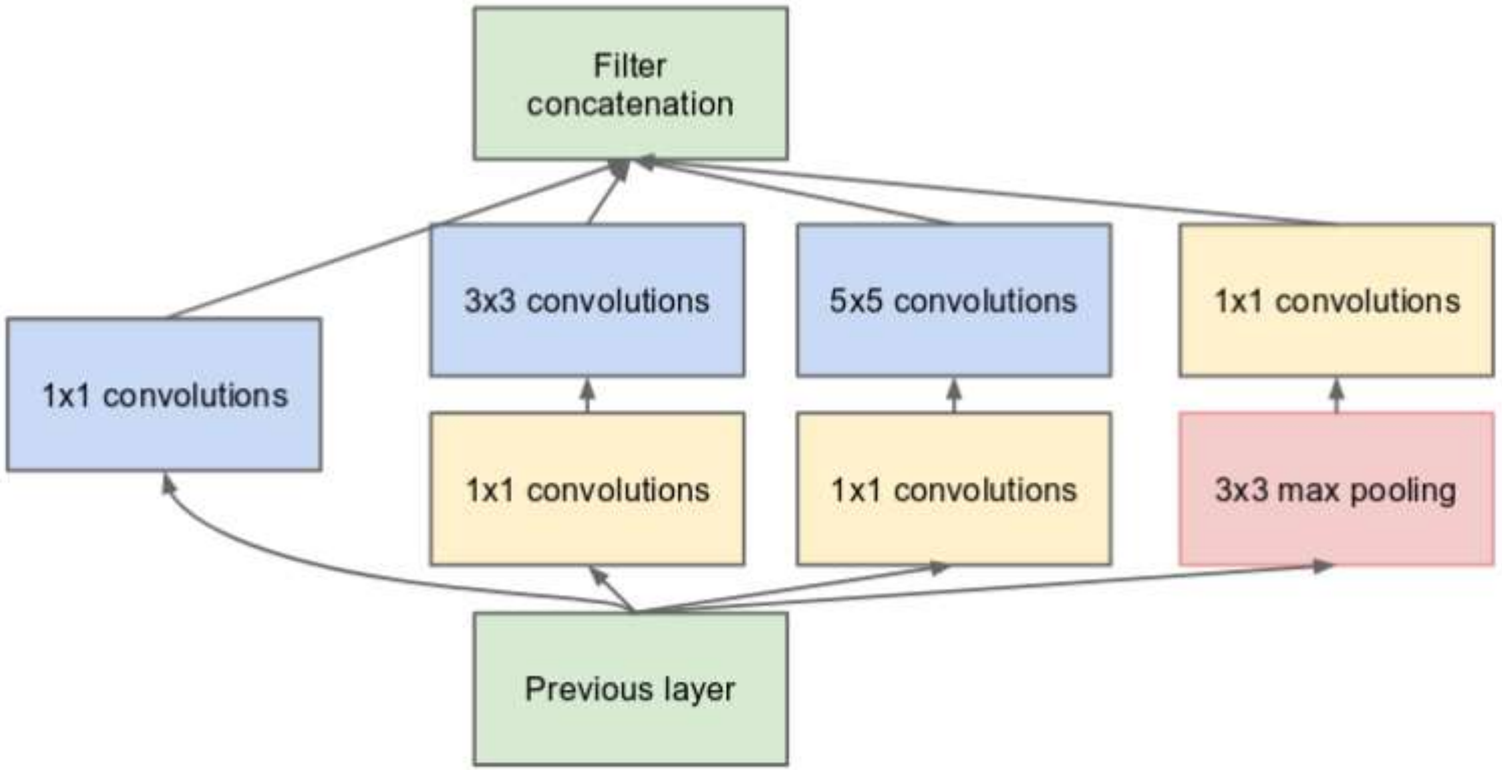
Number of parameters = (16×1×1×480) + (48×5×5x16) = **26.9K**

Number of operations = (7680×14×14) + (19200×14×14) = **5.27M**

Inception v1

Idea 2: filters of multiple sizes at the same layer

- With 1×1 convolutions: reduction layers.

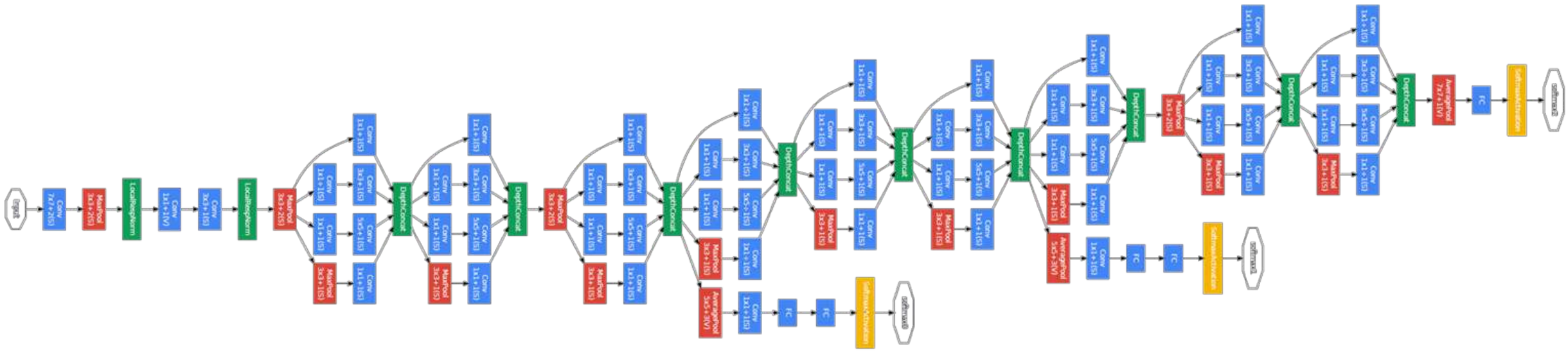


Inception module with dimension reduction

Inception v1

Idea 3: auxiliary classifiers

- Deep networks: subjected to vanishing gradient.

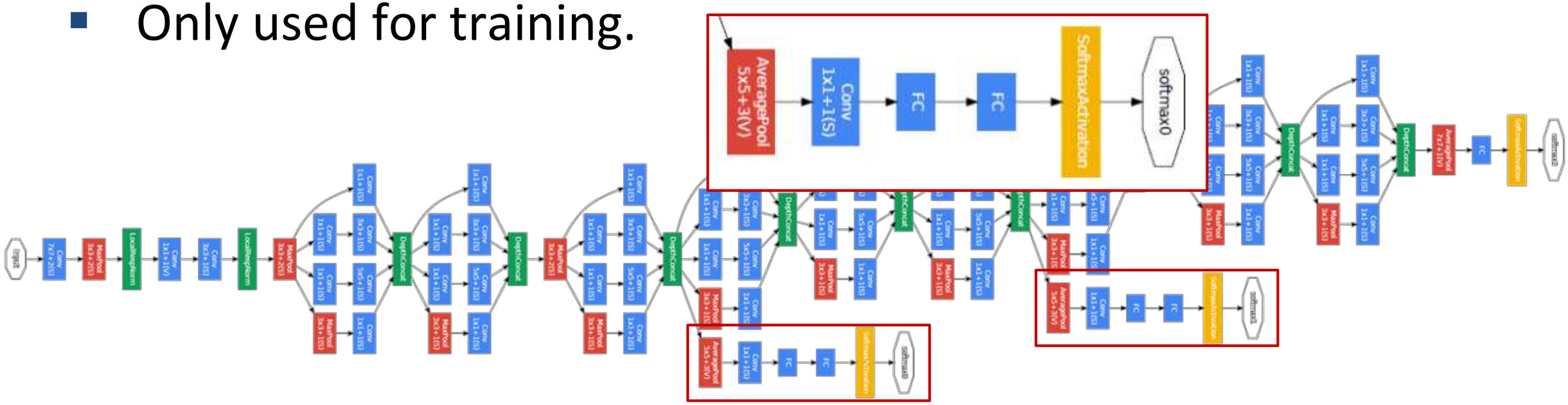


Convolution Pooling Softmax Concat/Normalize

Inception v1

Idea 3: auxiliary classifiers

- Two auxiliary classifiers (softmax) for outputs of two modules.
- The total loss is a weighted sum of auxiliary and main loss.
- Only used for training.



Convolution Pooling Softmax Concat/Normalize

Inception v2

- Built upon Inception v1.
- Aimed at increasing accuracy and reducing computational complexity.

Szegedy et al., 2015, Rethinking the Inception Architecture for Computer Vision, available at: https://arxiv.org/pdf/1512.00567v3.pdf

Inception v2

- Convolutions are associative operations

$$f * h * g = f * (h * g) = (f * h) * g$$

- Example 1:

h

2	0	3
3	3	2
2	3	2

$*$

g

1	2	0
2	0	0
1	1	2

$=$

$h * g$

4	2	6	2	2
6	9	14	5	6
4	12	17	15	8
0	6	13	13	6
0	4	8	7	2

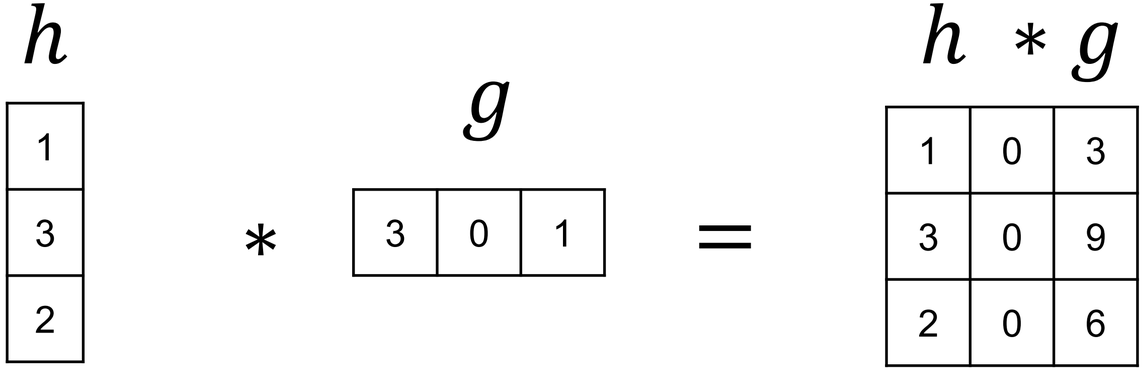
18 instead of 25 parameters!

Inception v2

- Convolutions are associative operations

$$f * h * g = f * (h * g) = (f * h) * g$$

- Example 2:

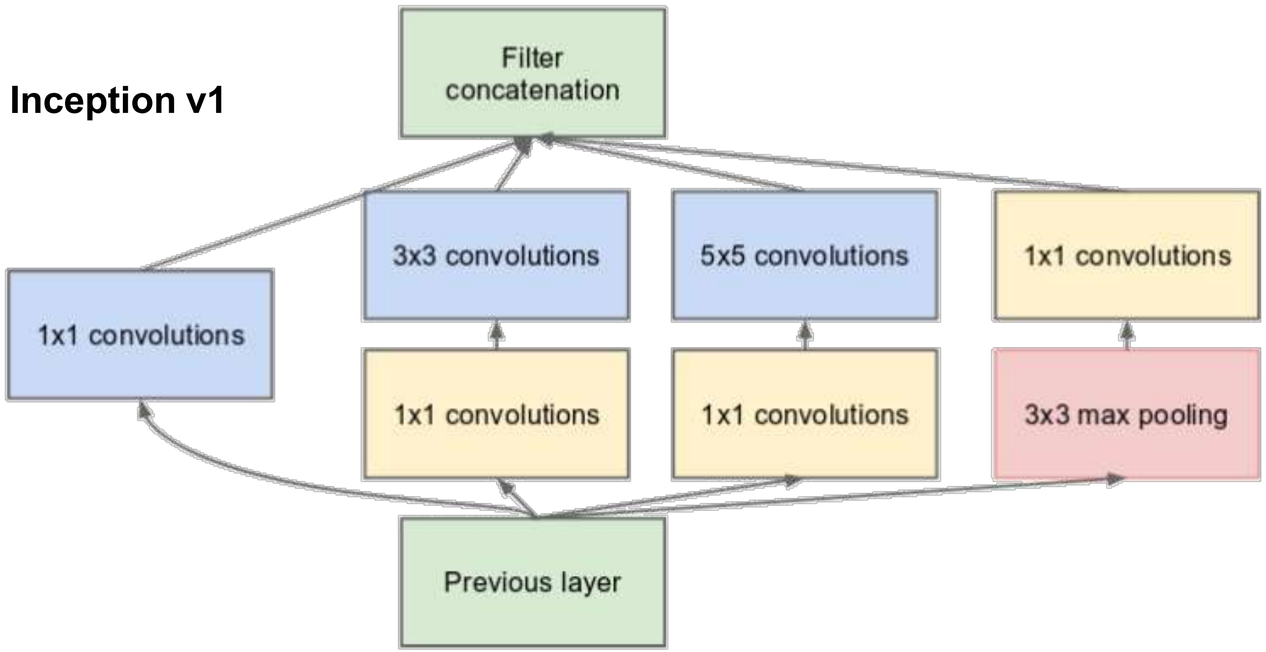


6 instead of 9 parameters!

Inception v2

Idea: factorize convolutions

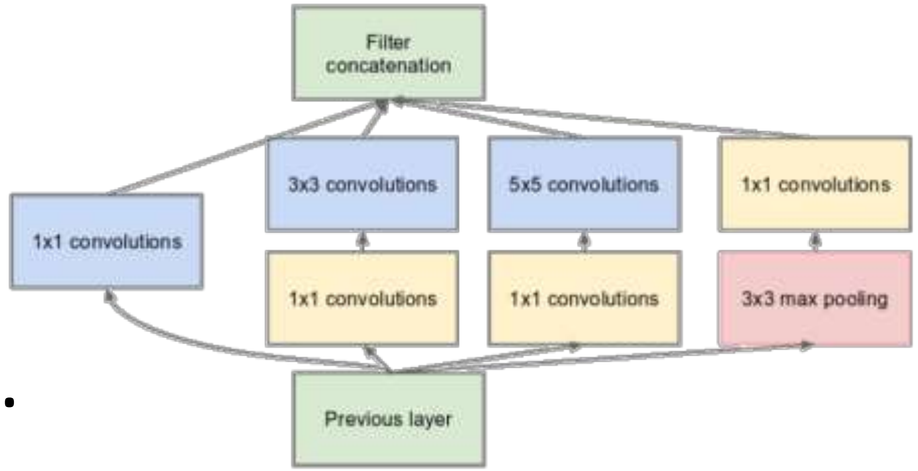
- Replace 5×5 convs by two 3×3 convs.
- Replace n×n convs by 1×n and n×1 convs.

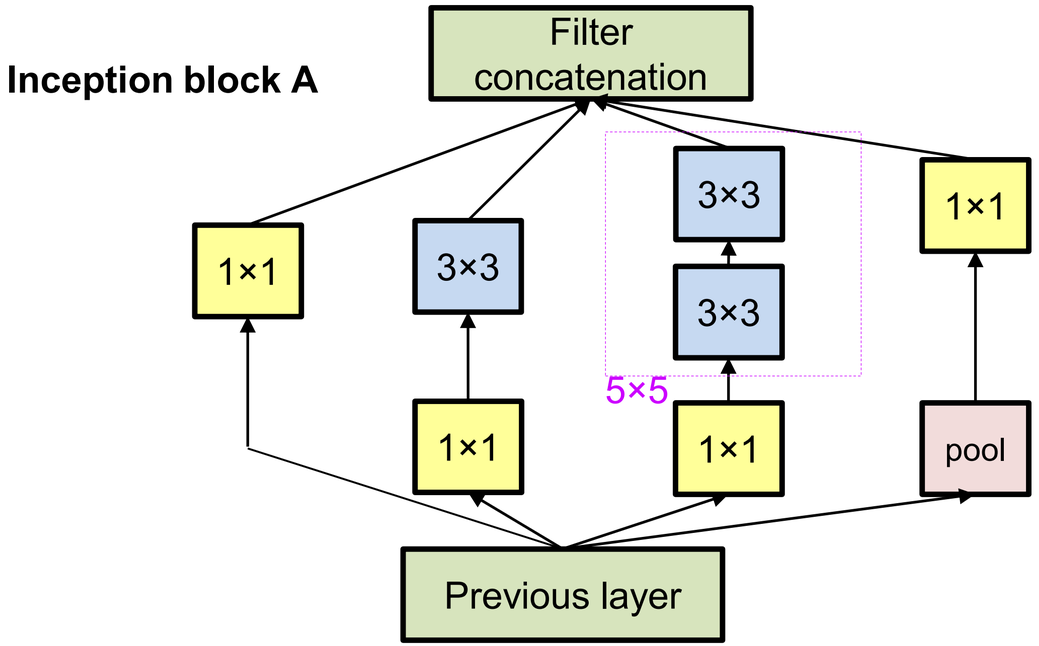


Inception v2

Idea: factorize convolutions

- Replace 5×5 convs by two 3×3 convs.

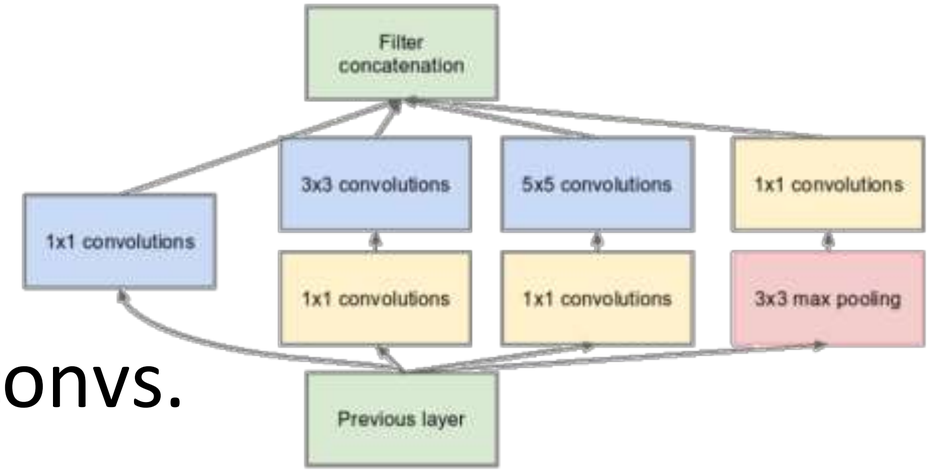


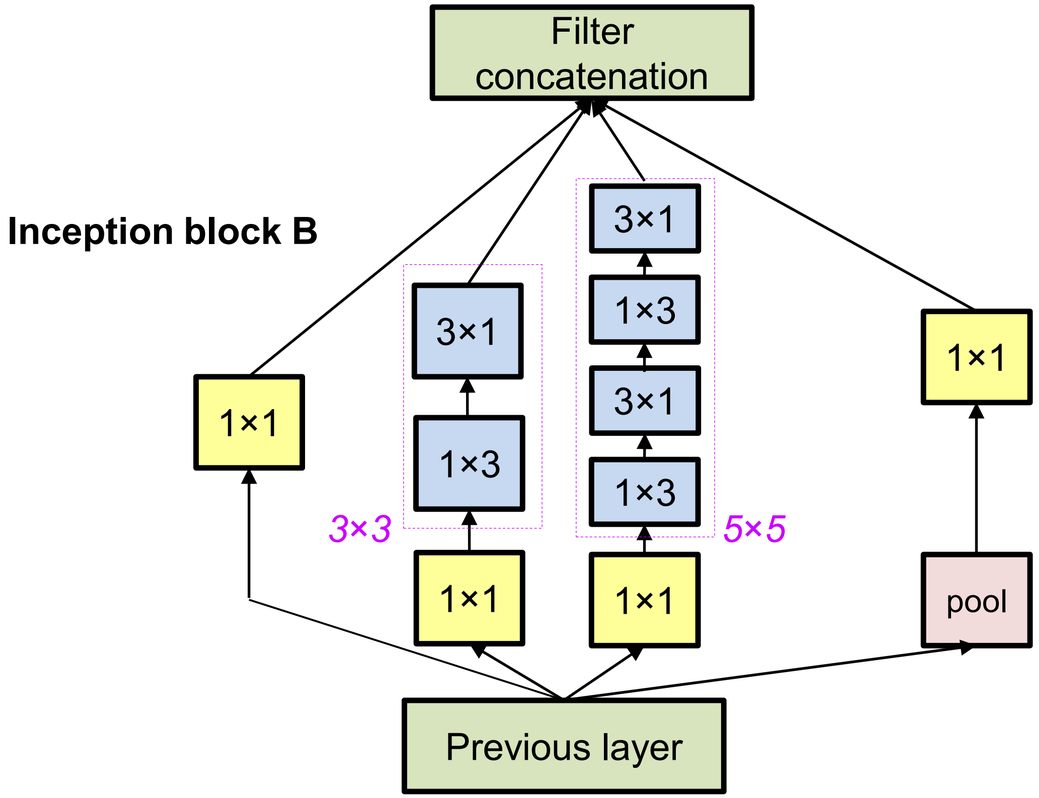


Inception v2

Idea: factorize convolutions

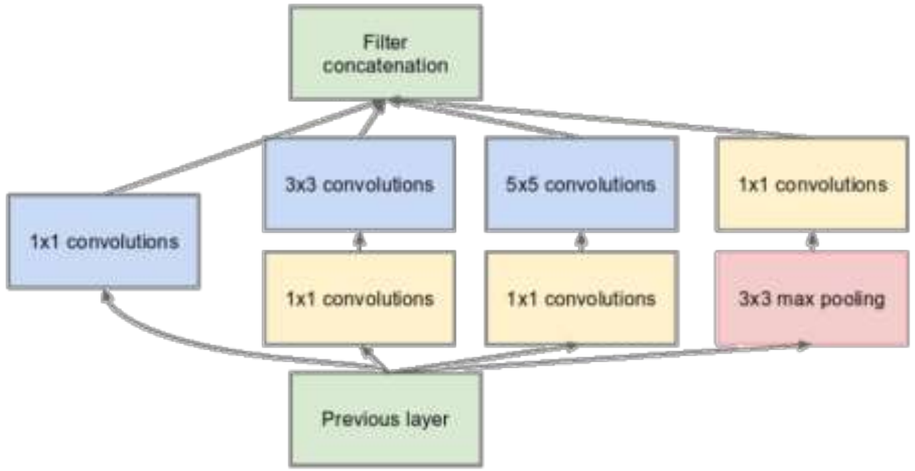
- Replace n×n convs by 1×n and n×1 convs.

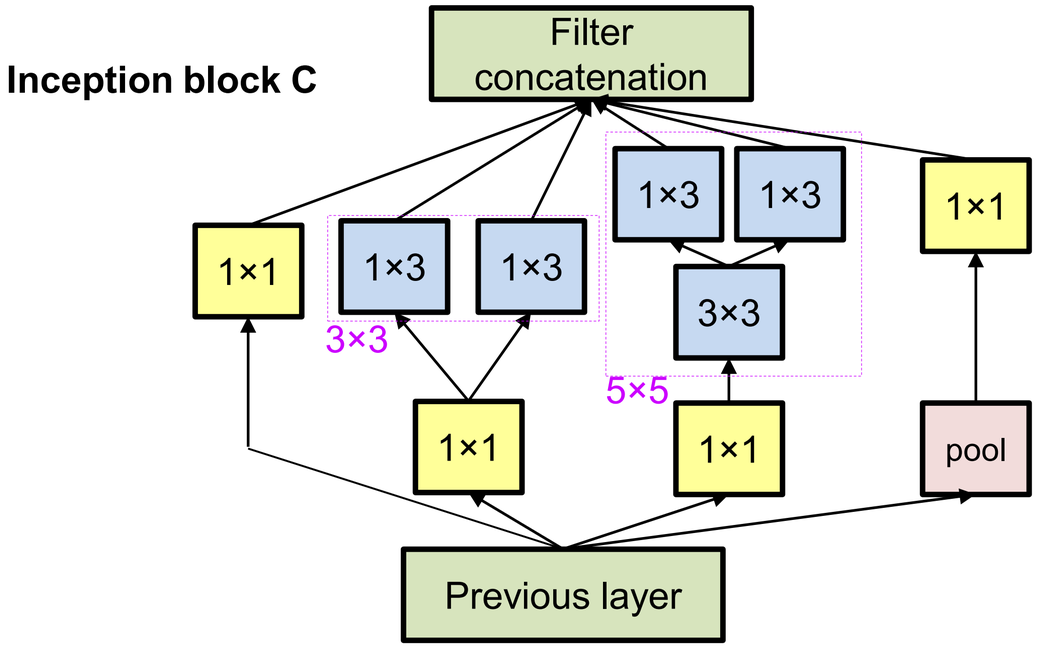




Inception v2

Idea: factorize convolutions





Inception v2

type	patch size/stride or remarks	input size
conv	3×3/2	299×299×3
conv	3×3/1	149×149×32
conv padded	3×3/1	147×147×32
pool	3×3/2	147×147×64
conv	3×3/1	73×73×64
conv	3×3/2	71×71×80
conv	3×3/1	35×35×192
3×Inception	Inception A	35×35×288
5×Inception	Inception B	17×17×768
2×Inception	Inception C	8×8×1280
pool	8 × 8	8 × 8 × 2048
linear	logits	1 × 1 × 2048
softmax	classifier	1 × 1 × 1000

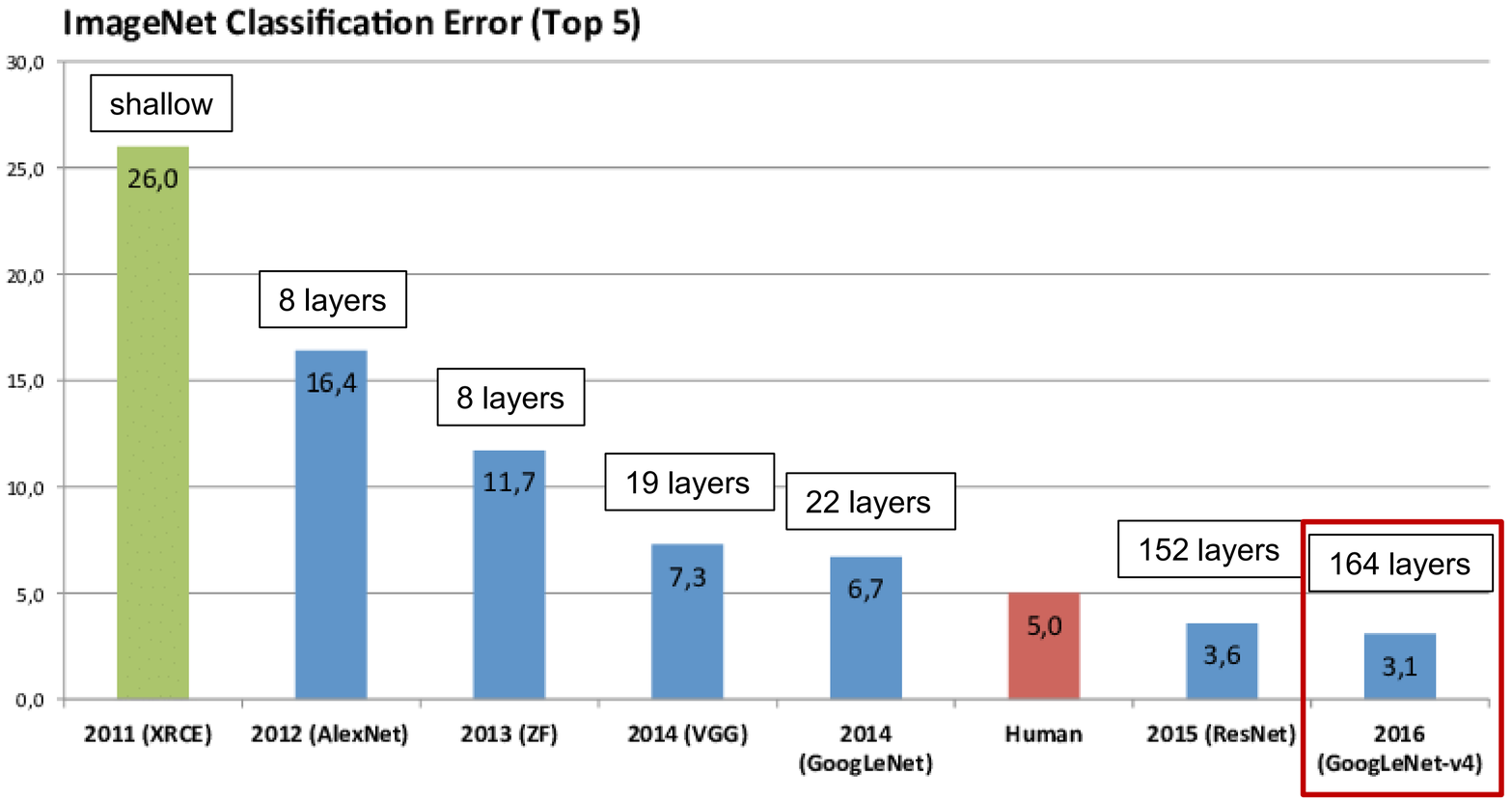
Inception v3

- Incorporated all features of previous versions.
- In addition:
  - RMSProp optimizer.
  - Factorized 7×7 convolutions.
  - BatchNorm in the Auxiliary Classifiers.
  - Label smoothing (a type of regularization):

$$y\_ls = (1 - \alpha) \times y\_hot + \alpha / K$$

Szegedy et al., 2015, Rethinking the Inception Architecture for Computer Vision, available at: https://arxiv.org/pdf/1512.00567v3.pdf

ImageNet Challenge

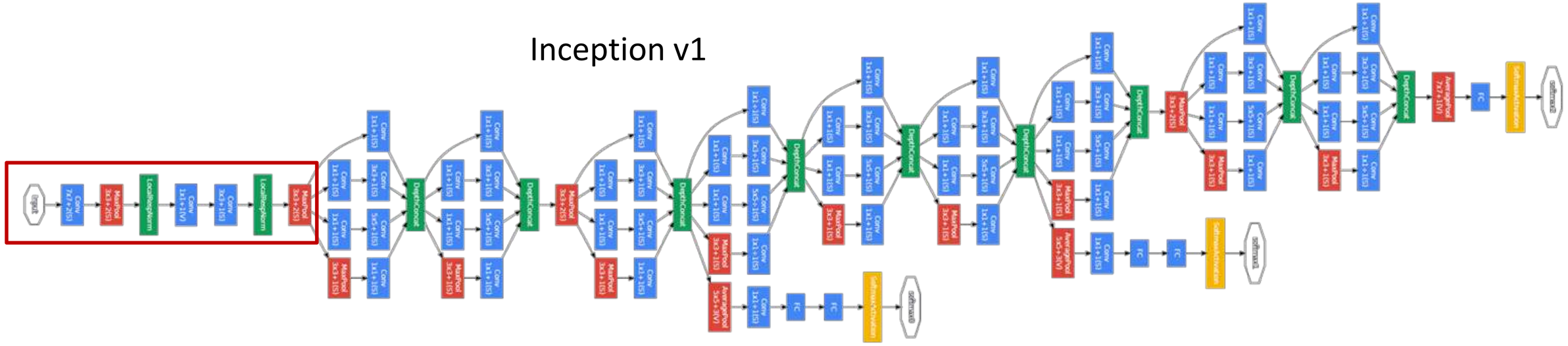


https://www.researchgate.net/figure/Winner-results-of-the-ImageNet-large-scale-visual-recognition-challenge-LSVRC-of-the_fig7_324476862

Inception v4

- Made the modules of previous Inception versions less complicated/more uniform.
- Modified “stem” (prior the inception blocks).

Inception v1

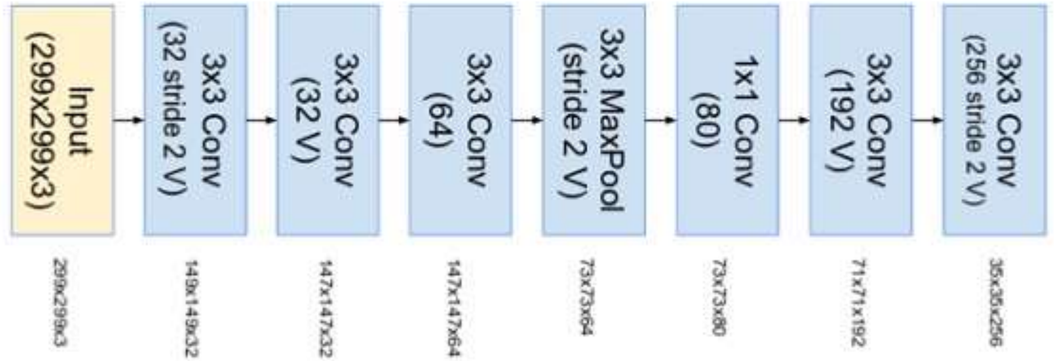


Szegedy et al., 2016, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, available at: https://arxiv.org/pdf/1602.07261.pdf

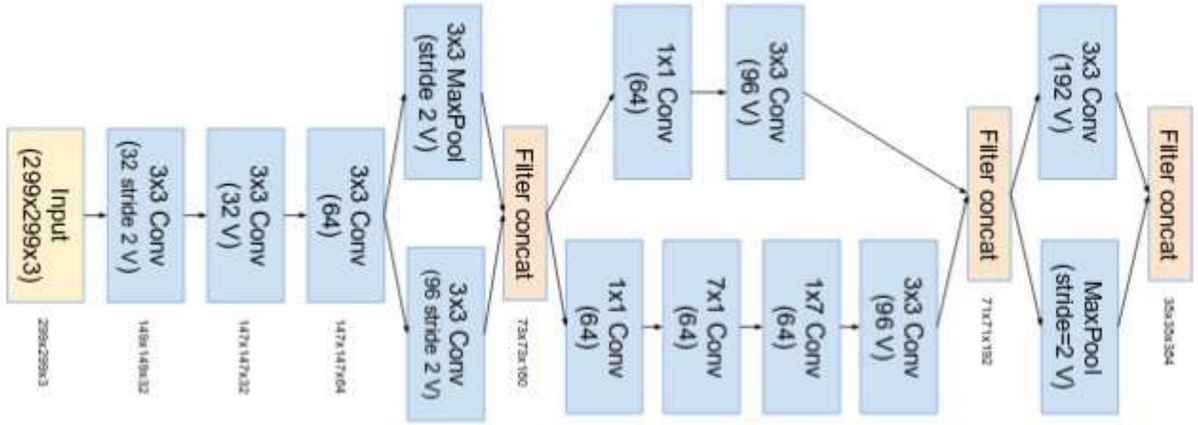
Inception v4

- Modified “stem” (prior the inception blocks).

In Inception v1



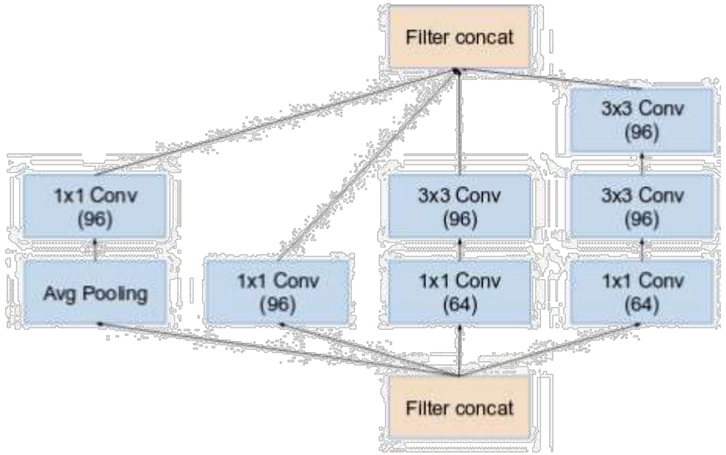
In Inception v4



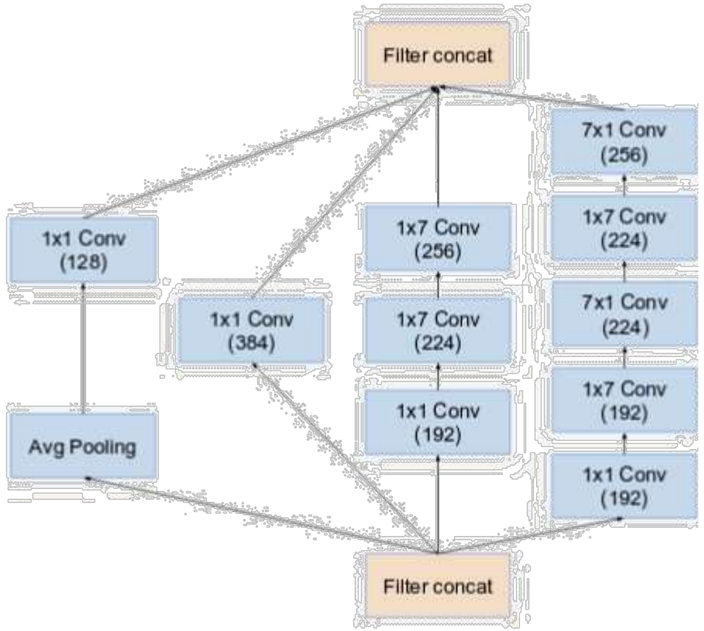
Szegedy et al., 2016, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, available at: https://arxiv.org/pdf/1602.07261.pdf

Inception v4

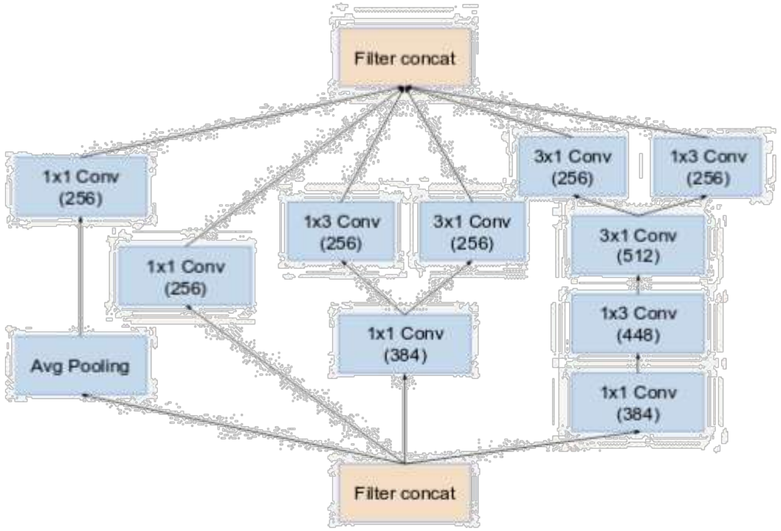
- Simpler, more uniform and more inception modules.



Inception A



Inception B



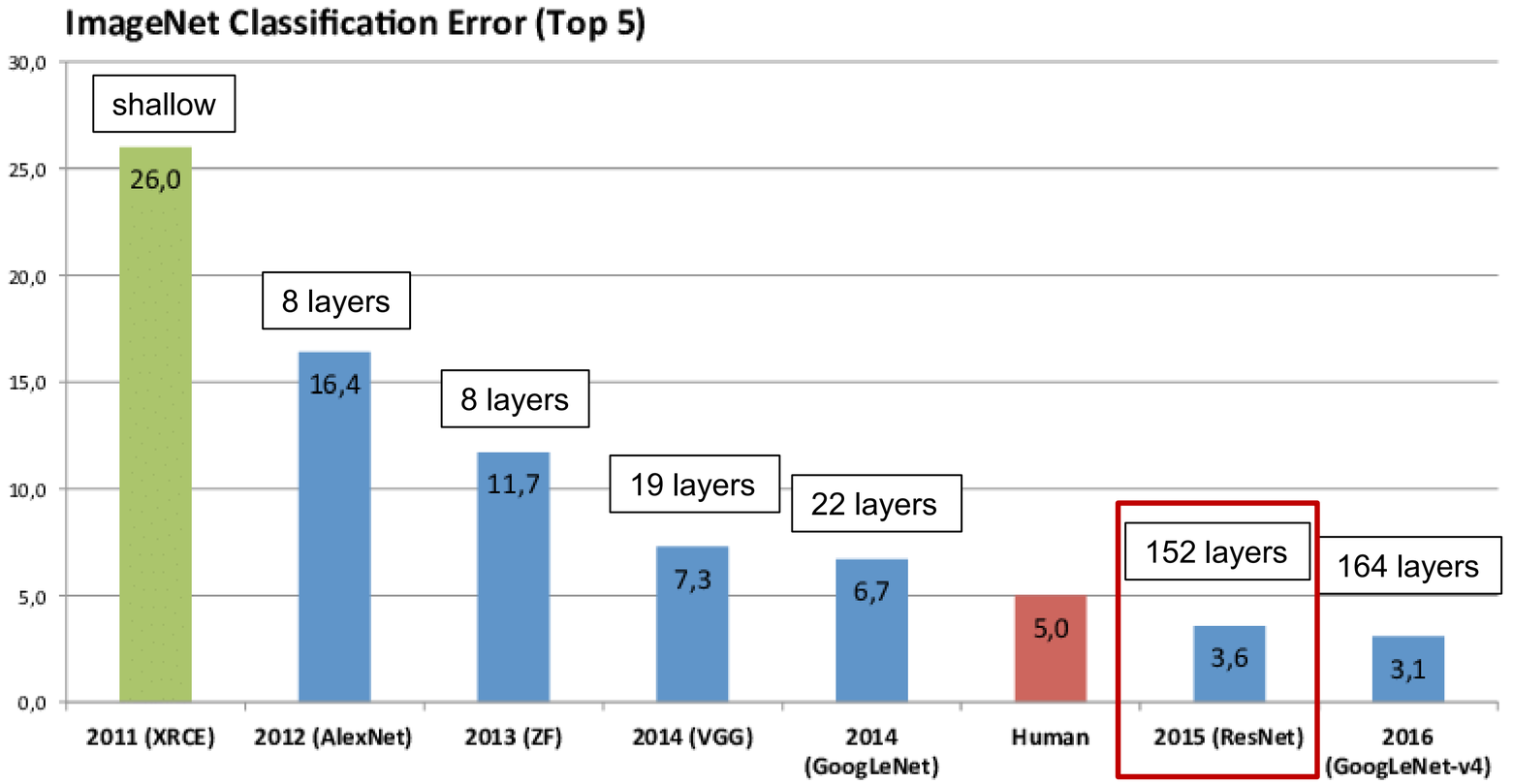
Inception C

Szegedy et al., 2016, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, available at: https://arxiv.org/pdf/1602.07261.pdf

Content

- LeNet
- AlexNet
- ZFnet
- VGGNet
- GogLeNet
- ResNet
- DenseNet

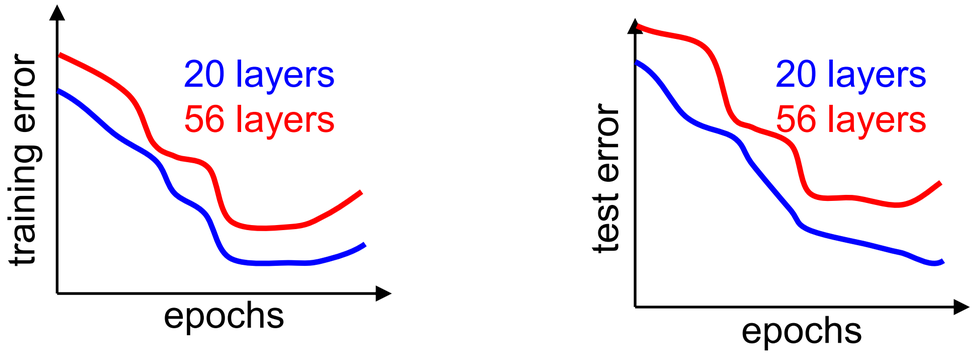
ImageNet Challenge



https://www.researchgate.net/figure/Winner-results-of-the-ImageNet-large-scale-visual-recognition-challenge-LSVRC-of-the_fig7_324476862

ResNet

- Adding additional layers will not necessarily improve accuracy.
- This is not necessarily due to overfitting: vanishing gradient.

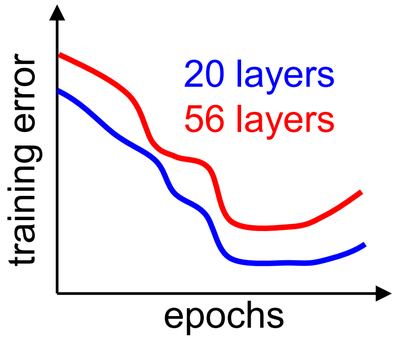


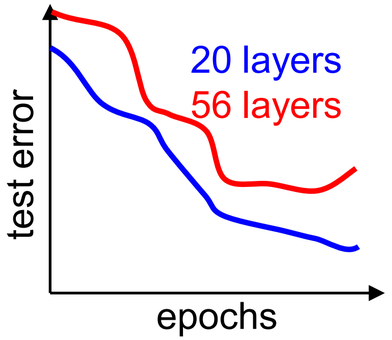
- Solutions were previously proposed, e.g. auxiliary classifiers in Inception v1.

He et al., 2015, Deep Residual Learning for Image Recognition, available at https://arxiv.org/pdf/1512.03385.pdf

ResNet

- Adding additional layers will not necessarily improve accuracy.
- This is not necessarily due to overfitting: vanishing gradient.





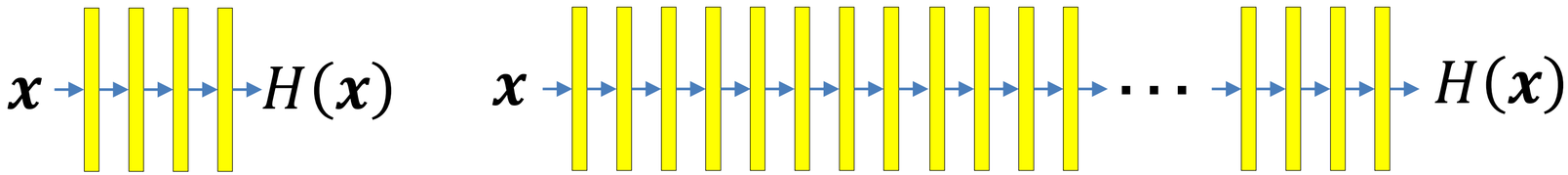
- ResNet proposes an alternative (definite) solution: residual blocks.
- Inspired in pyramidal neurons (prefrontal cortex).

He et al., 2015, Deep Residual Learning for Image Recognition, available at https://arxiv.org/pdf/1512.03385.pdf

ResNet

Intuition:

- Assume that a shallow network learns the function $\boldsymbol{y}=H(\boldsymbol{x})$ pretty well.



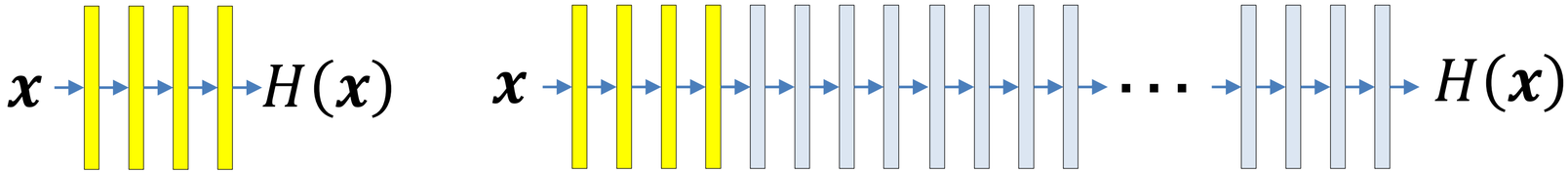
- What happens if we add layers?
- Will the deeper net do better than the shallow one?

He et al., 2015, Deep Residual Learning for Image Recognition, available at https://arxiv.org/pdf/1512.03385.pdf

ResNet

Intuition:

- Assume that a shallow network learns the function $\boldsymbol{y}=H(\boldsymbol{x})$ pretty well.

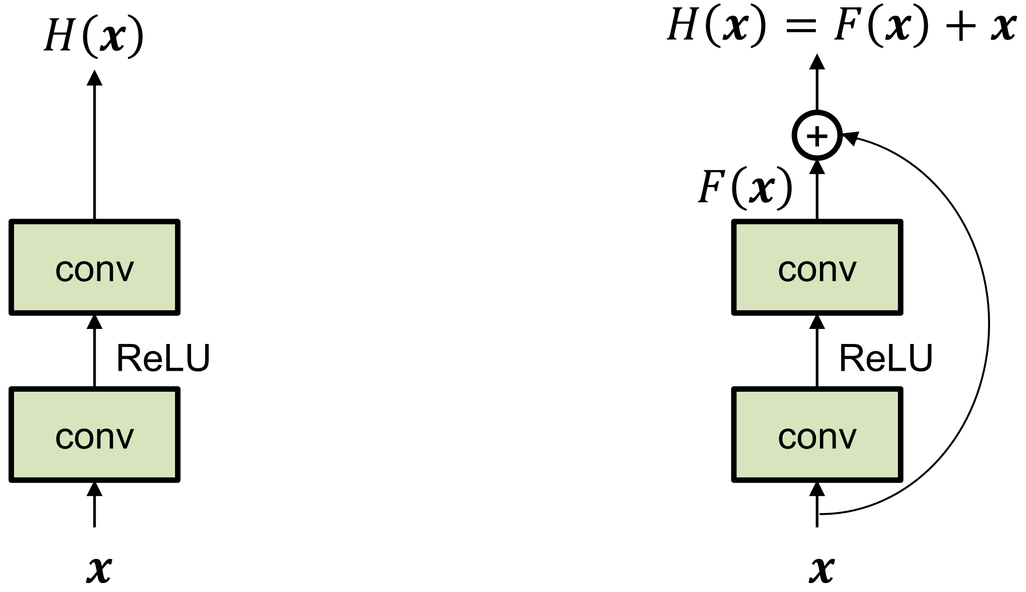


- Solution by construction: copy learned shallow part and set the additional layers to identity mapping.

He et al., 2015, Deep Residual Learning for Image Recognition, available at https://arxiv.org/pdf/1512.03385.pdf

ResNet

- Use network layers to fit a **residual mapping** instead of directly trying to fit the underlying mapping.

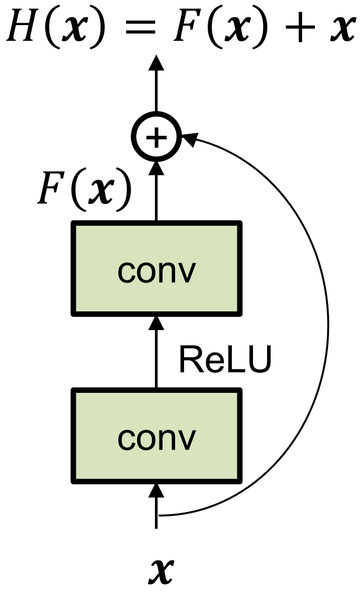


- $F(\boldsymbol{x}) = 0$ is easier to learn than $H(\boldsymbol{x}) = \boldsymbol{x}$

He et al., 2015, Deep Residual Learning for Image Recognition, available at https://arxiv.org/pdf/1512.03385.pdf

ResNet

- Instead of learning $H(\boldsymbol{x})$ the convolutions learn what must be added or subtracted from the input to generate $H(\boldsymbol{x})$.
- The convolutions learn what to change in the input to obtain the desired output.

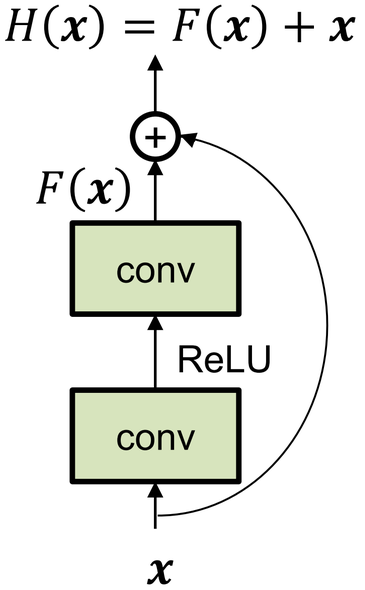


He et al., 2015, Deep Residual Learning for Image Recognition, available at https://arxiv.org/pdf/1512.03385.pdf

ResNet

- Residual blocks: prevent gradient to vanish.
- The core idea is to backpropagate through the identity function by just using a vector addition.
- Then the gradient would simply be multiplied by one and its value will be maintained in the earlier layers.

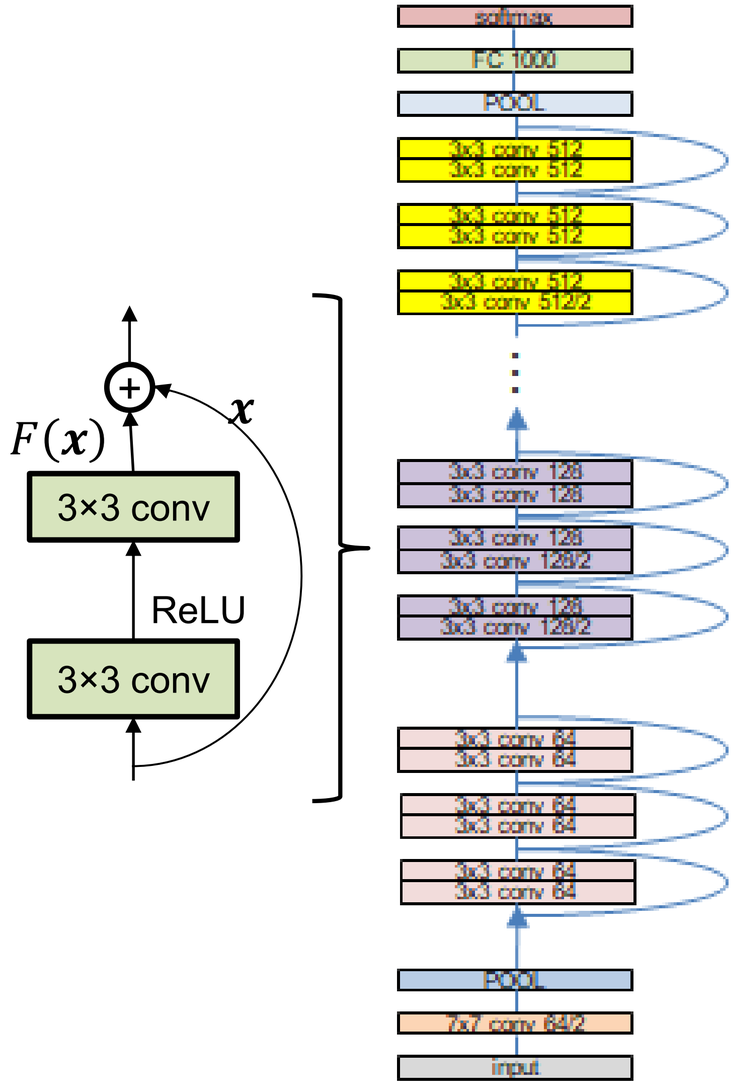
$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial H}\frac{\partial H}{\partial x} = \frac{\partial L}{\partial H}\left(\frac{\partial F}{\partial x} + 1\right) = \frac{\partial L}{\partial H}\frac{\partial F}{\partial x} + \frac{\partial L}{\partial H}$$



He et al., 2015, Deep Residual Learning for Image Recognition, available at https://arxiv.org/pdf/1512.03385.pdf

ResNet

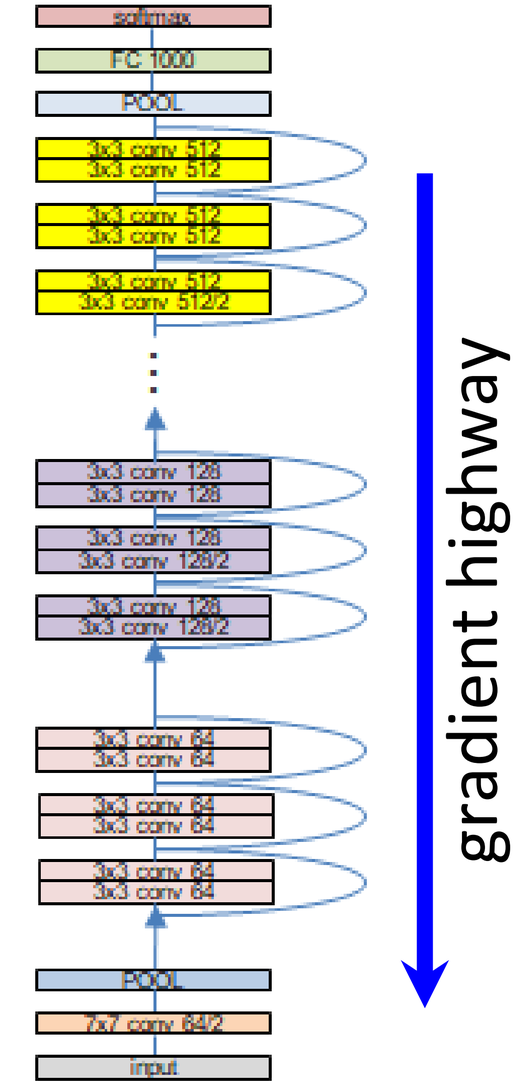
- Stack of residual blocks.
- Every residual block has two 3×3 conv layers.
- Periodically, double # of filters and downsample spatially using stride 2.
- Additional conv layer at the beginning.



He et al., 2015, Deep Residual Learning for Image Recognition, available at https://arxiv.org/pdf/1512.03385.pdf

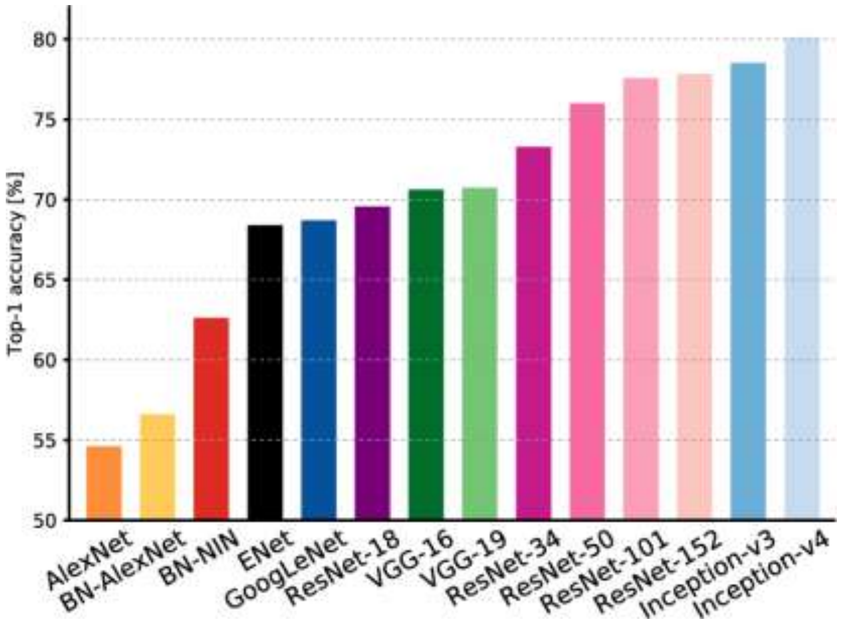
ResNet

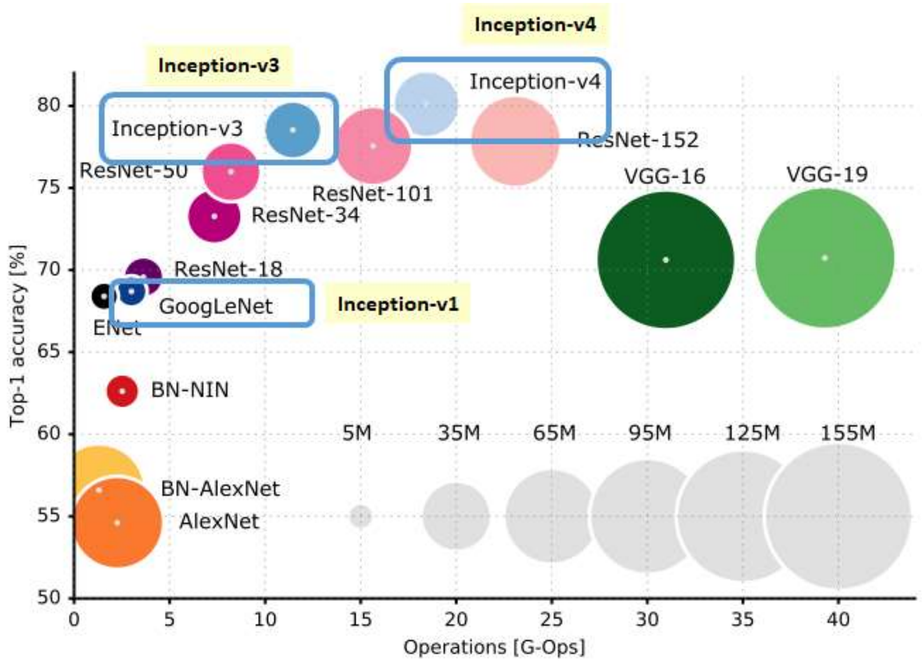
- The bypass connections build a “highway” for gradient propagation.
- Faster learning and allows deeper architectures.



He et al., 2015, Deep Residual Learning for Image Recognition, available at https://arxiv.org/pdf/1512.03385.pdf

Complexity × Accuracy





Canziani et al., 2017, An Analysis of Deep Neural Network Models for Practical Applications, available at https://arxiv.org/pdf/1605.07678.pdf

Content

- LeNet
- AlexNet
- ZFnet
- VGGNet
- GogLeNet
- ResNet
- DenseNet

DenseNet

- Dense blocks: each layer is connected to every other layer in feedforward fashion.

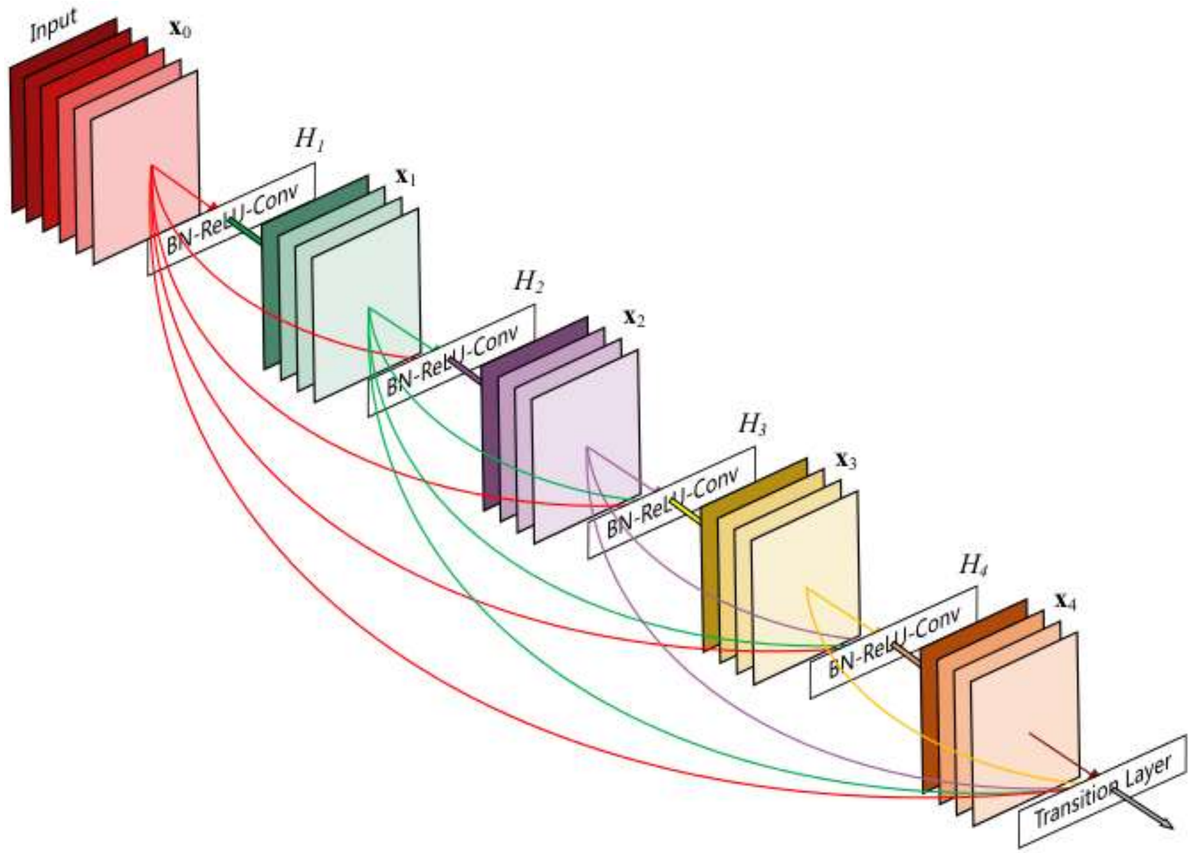


Figure from Huang al., 2018, Densely Connected Convolutional Networks, available at https://arxiv.org/pdf/1608.06993.pdf

DenseNet

- Alleviates vanishing gradient, strengthens feature propagation, encourages feature reuse.

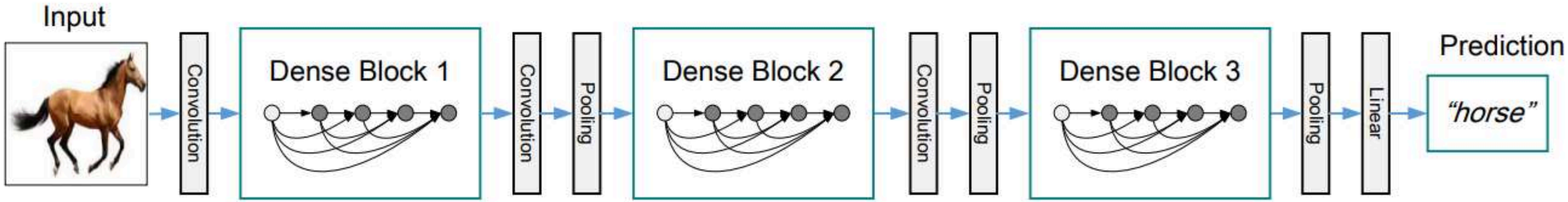


Figure 2: A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change feature-map sizes via convolution and pooling.

Figure from Huang al., 2017, Densely Connected Convolutional Networks, available at https://arxiv.org/pdf/1608.06993.pdf

Next Lecture

Thursday: Lab 1

Introduction to Colab
Backpropagation
Optimization

See you next class!