

Convolutional Neural Network Architectures: from LeNet to ResNet

Lana Lazebnik

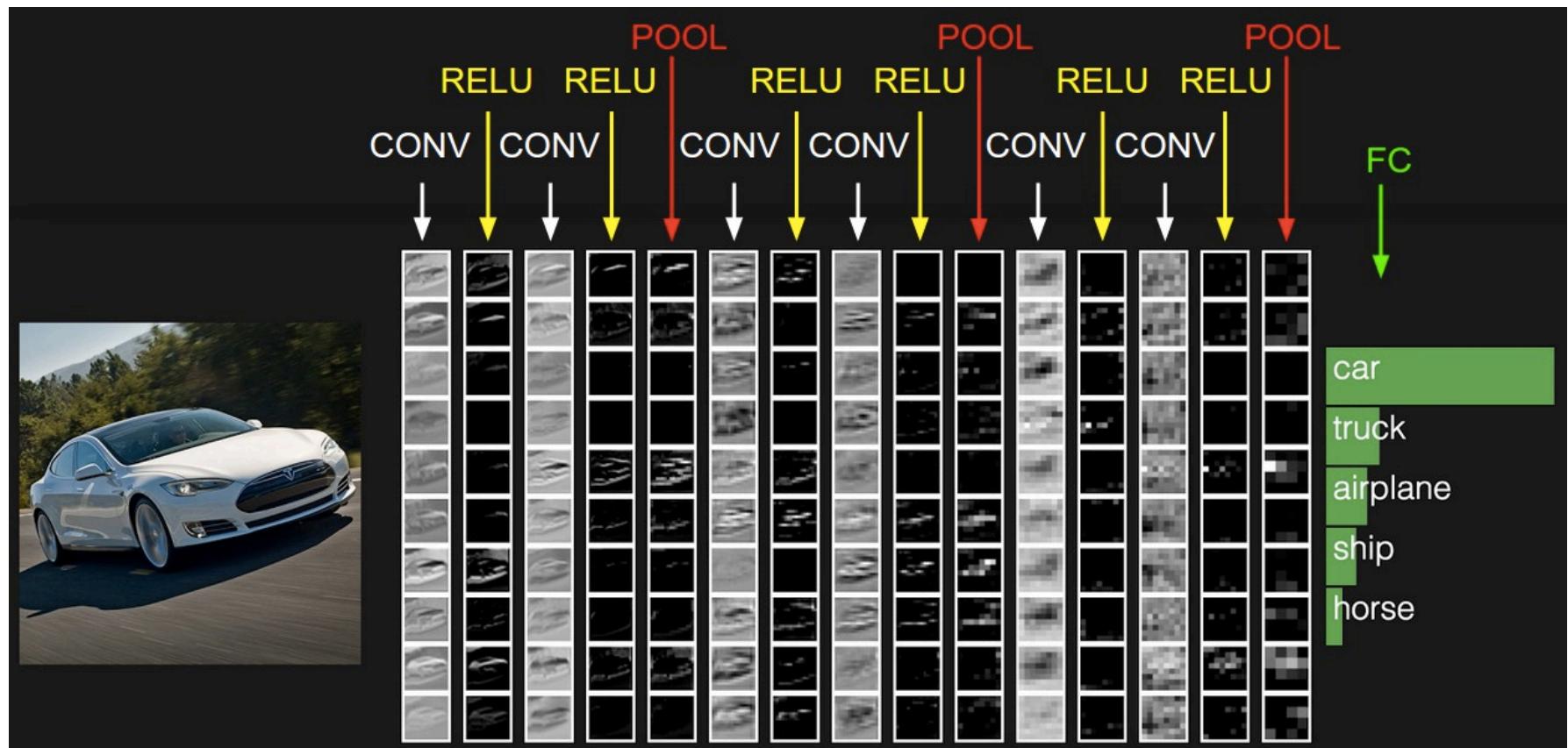


Figure source: A. Karpathy

What happened to my field?

Classification: ImageNet Challenge top-5 error

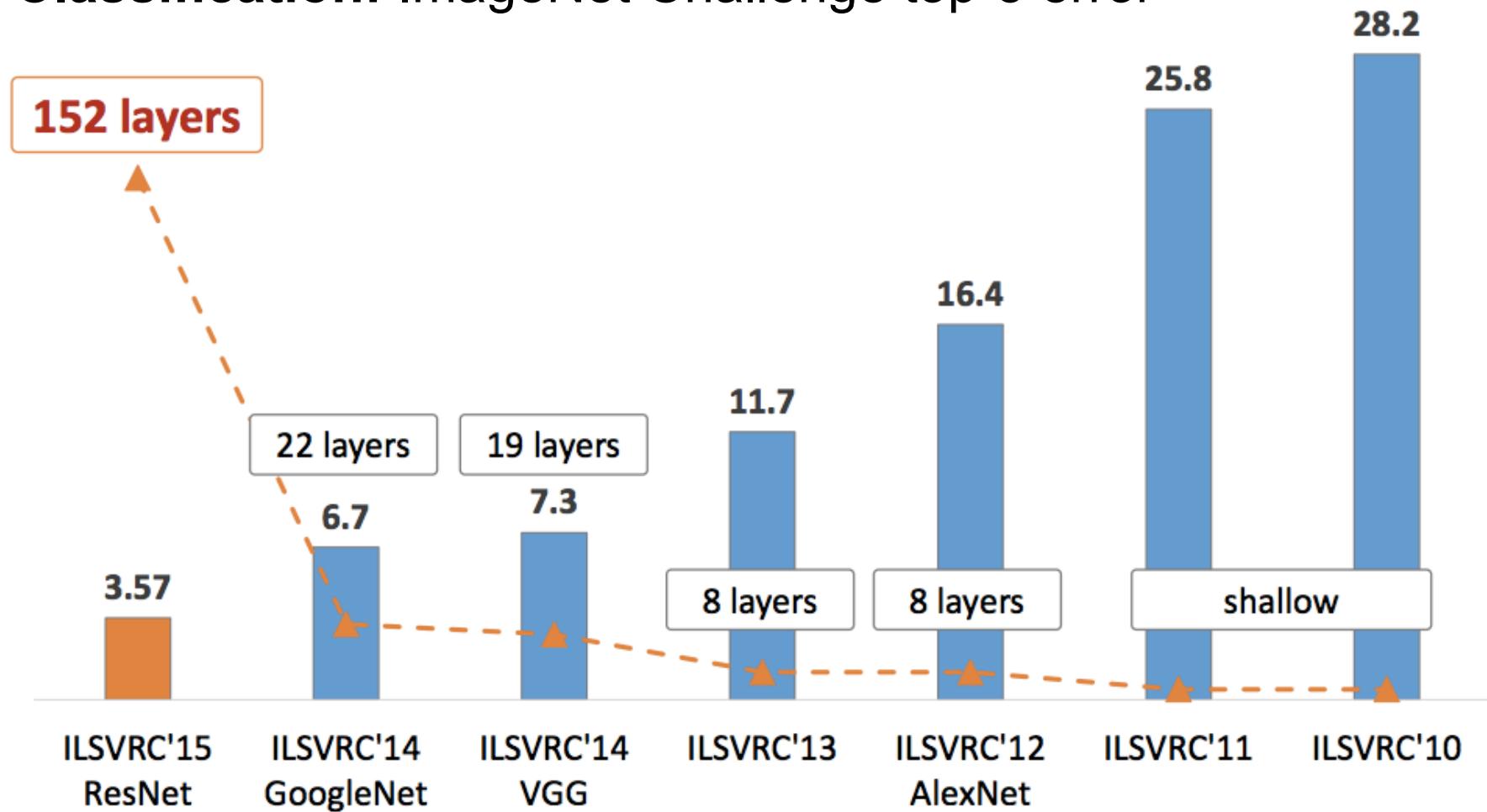
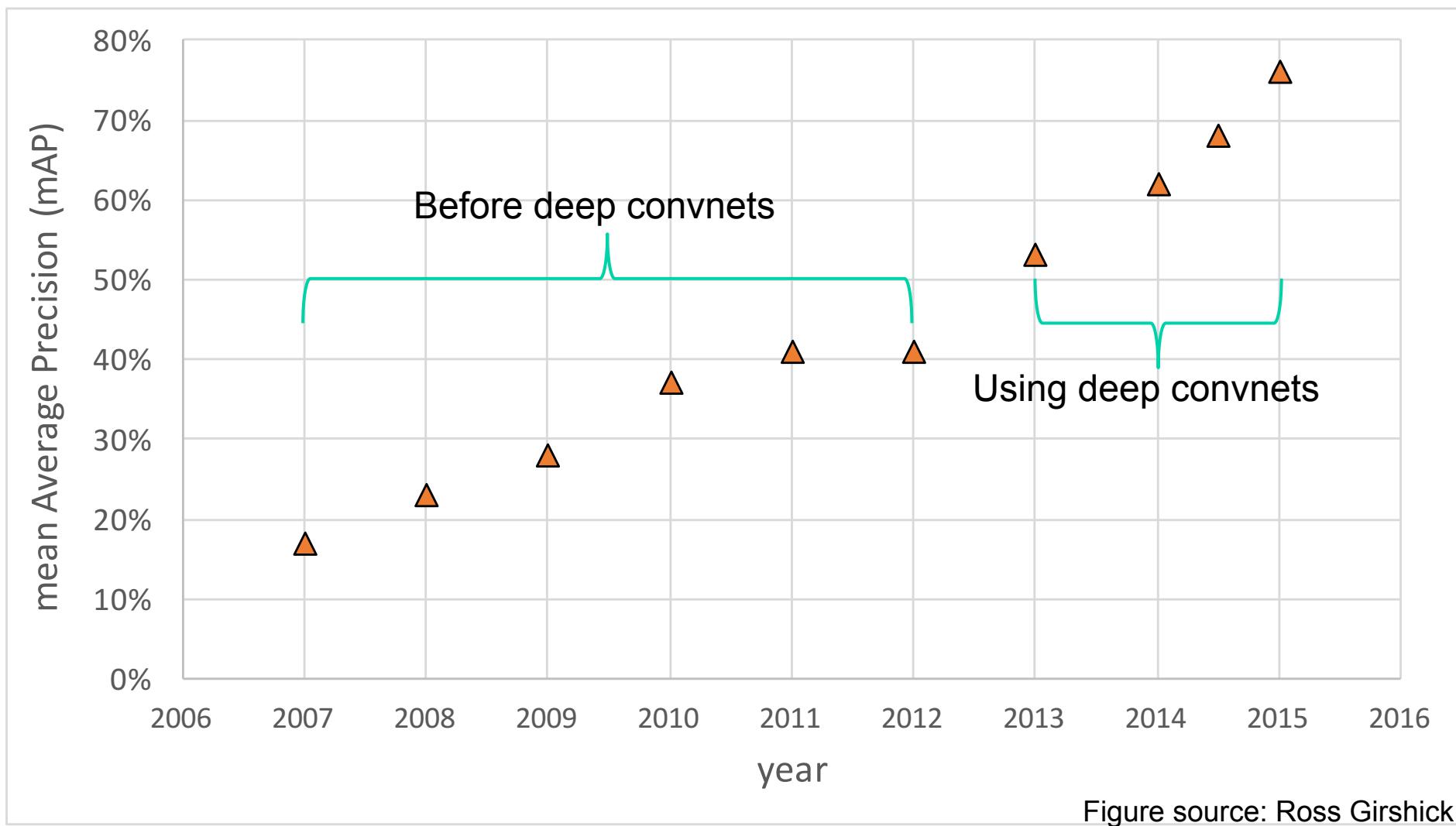


Figure source: [Kaiming He](#)

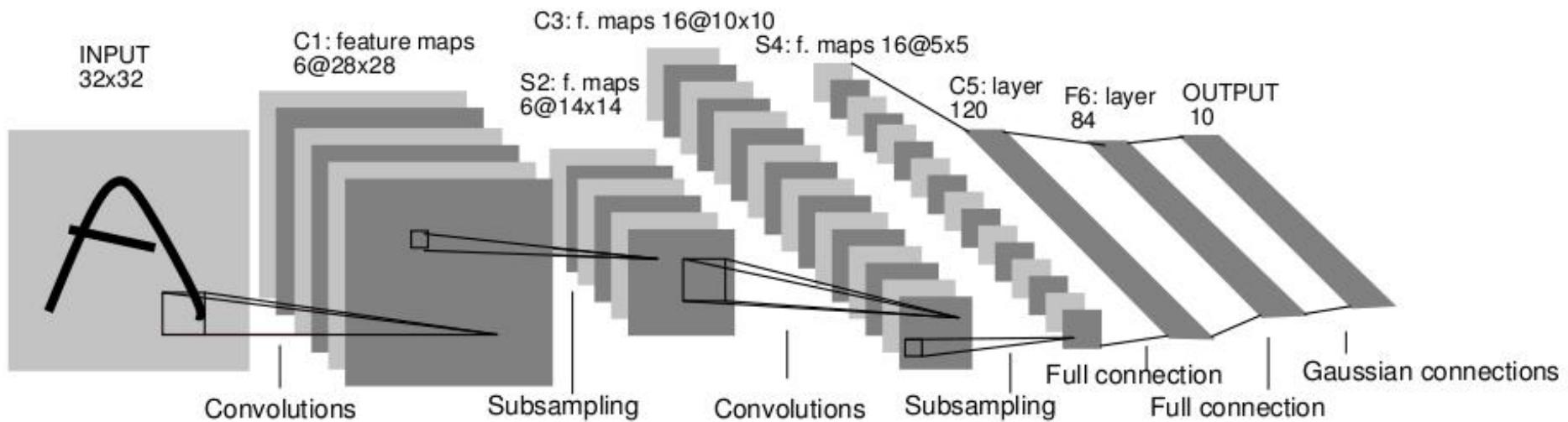
What happened to my field?

Object Detection: PASCAL VOC mean Average Precision (mAP)



Actually, it happened a while ago...

LeNet 5

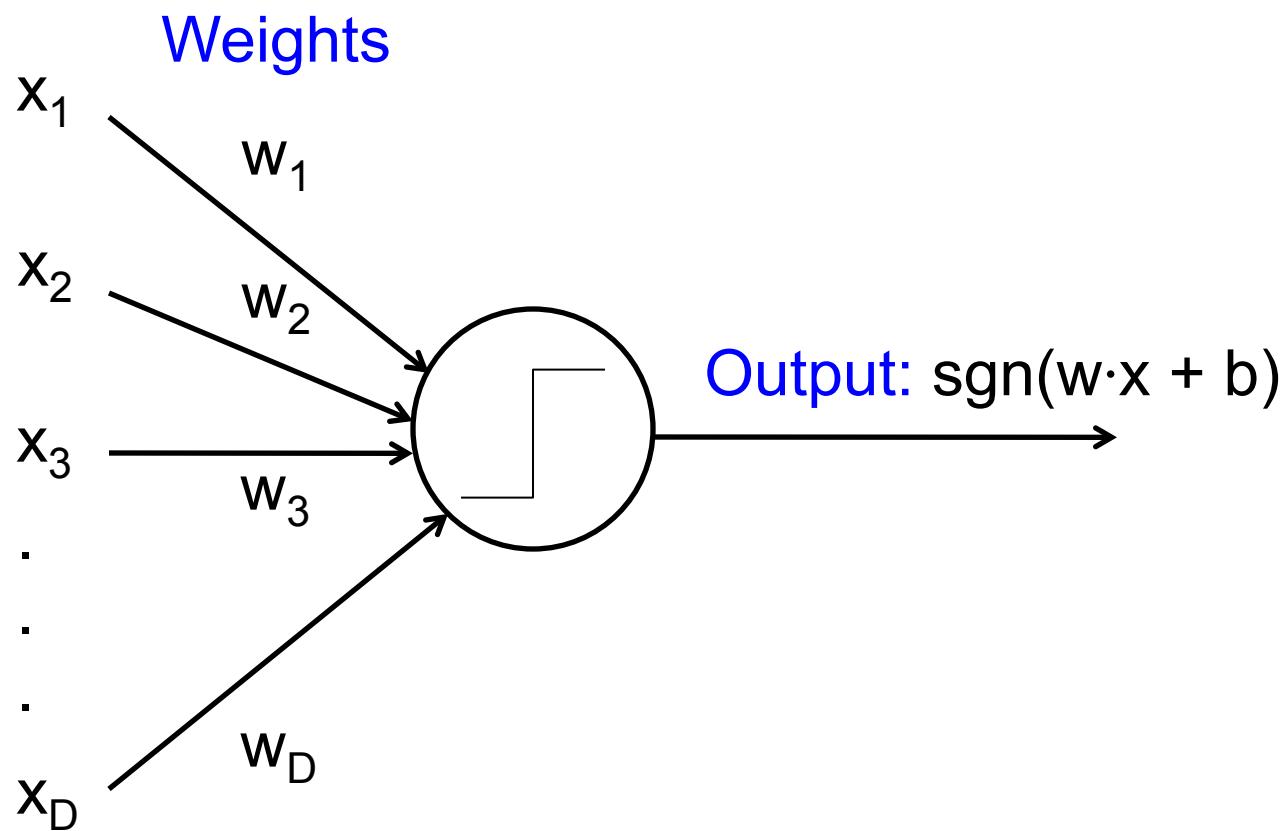


Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner,
[Gradient-based learning applied to document recognition](#), Proc. IEEE 86(11): 2278–2324, 1998.

Let's back up even more...

The Perceptron

Input



Rosenblatt, Frank (1958), The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Cornell Aeronautical Laboratory, Psychological Review, v65, No. 6, pp. 386–408.

Let's back up even more...

NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo
of Computer Designed to
Read and Grow Wiser

WASHINGTON, July 7 (UPI)—The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's \$2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of \$100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human be-

ings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

Without Human Controls

The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted.

Mr. Rosenblatt said in principle it would be possible to build brains that could reproduce themselves on an assembly line and which would be conscious of their existence.

1958 New York
Times...

In today's demonstration, the "704" was fed two cards, one with squares marked on the left side and the other with squares on the right side.

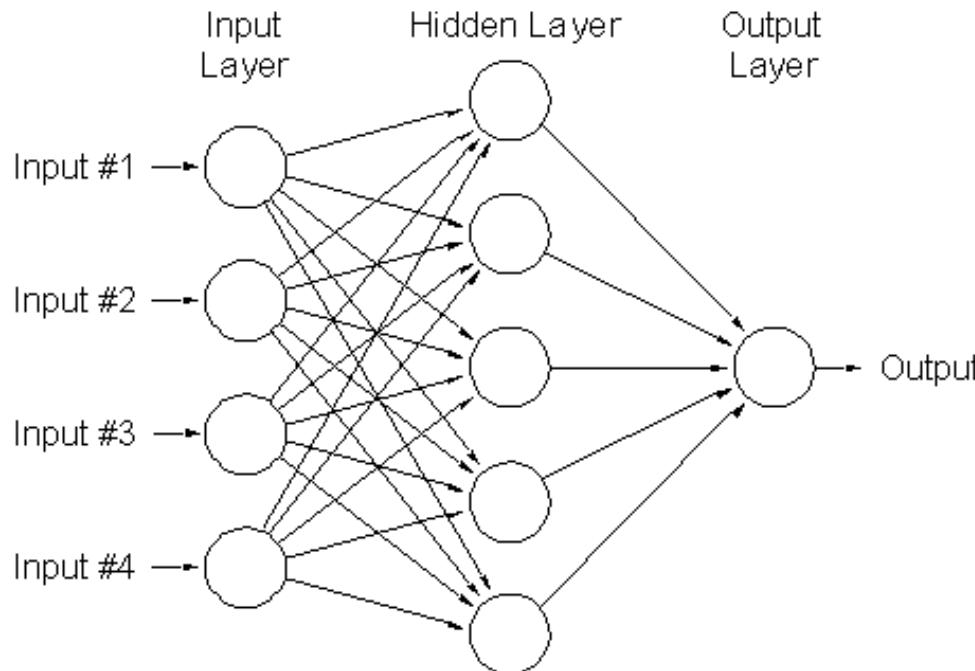
Learns by Doing

In the first fifty trials, the machine made no distinction between them. It then started registering a "Q" for the left squares and "O" for the right squares.

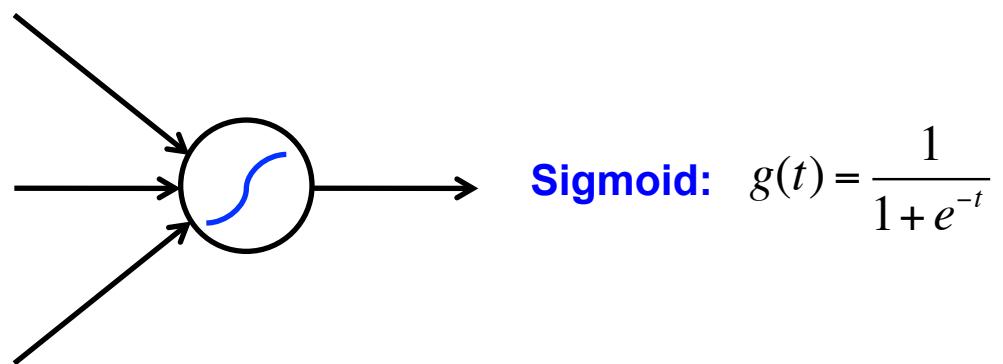
Dr. Rosenblatt said he could explain why the machine learned only in highly technical terms. But he said the computer had undergone a "self-induced change in the wiring diagram."

The first Perceptron will have about 1,000 electronic "association cells" receiving electrical impulses from an eye-like scanning device with 400 photo-cells. The human brain has 10,000,000,000 responsive cells, including 100,000,000 connections with the eyes.

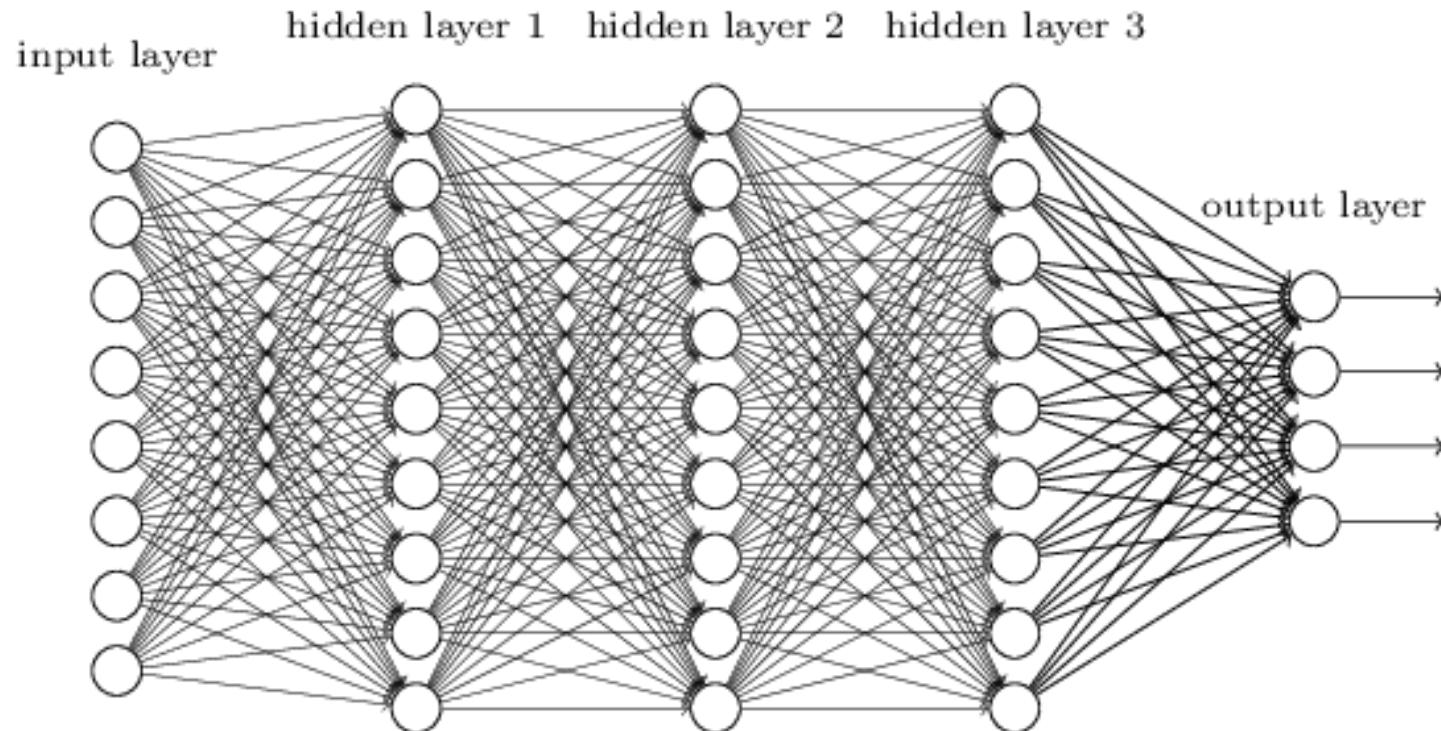
Two-layer neural network



- Can learn nonlinear functions provided each perceptron has a differentiable nonlinearity



Multi-layer neural network

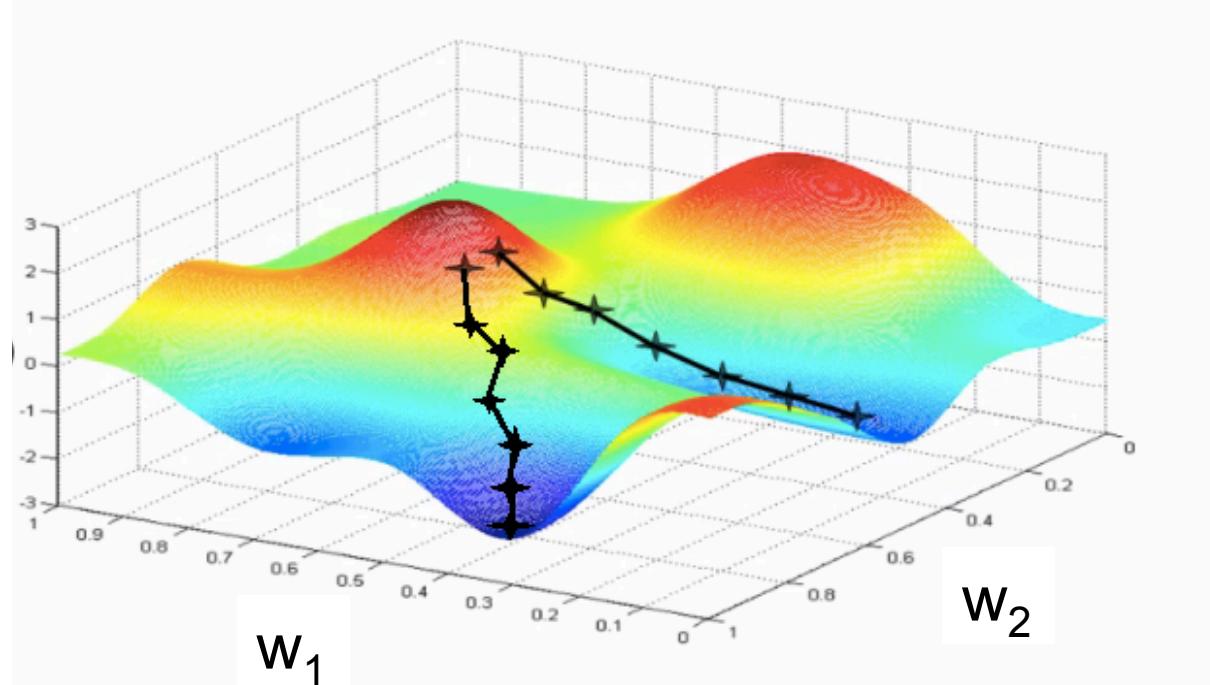


Training of multi-layer networks

- Find network weights to minimize the *training error* between true and estimated labels of training examples, e.g.:

$$E(\mathbf{w}) = \sum_{i=1}^N (y_i - f_{\mathbf{w}}(\mathbf{x}_i))^2$$

- Update weights by **gradient descent**: $\mathbf{w} \leftarrow \mathbf{w} - \alpha \frac{\partial E}{\partial \mathbf{w}}$



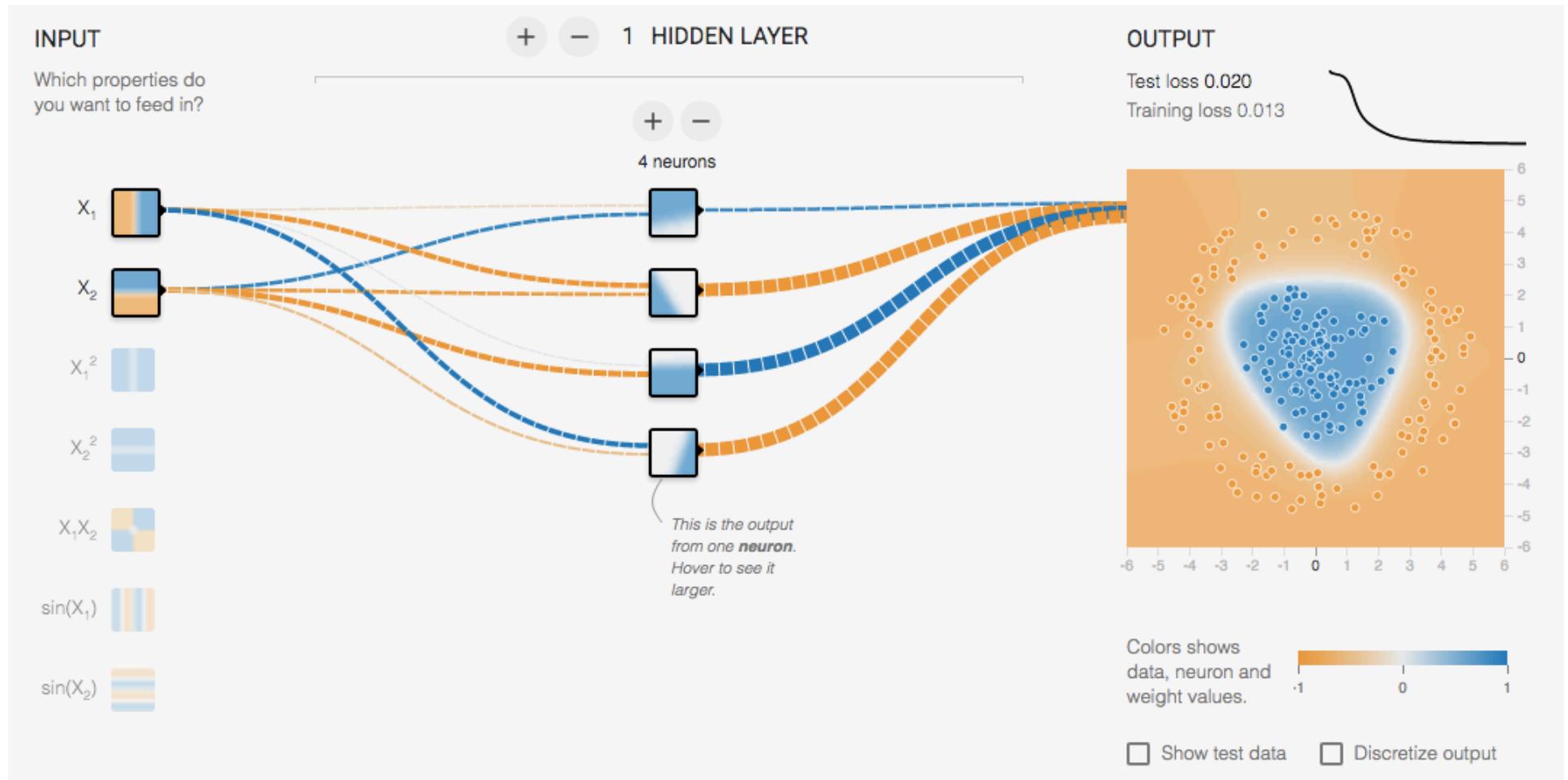
Training of multi-layer networks

- Find network weights to minimize the *training error* between true and estimated labels of training examples, e.g.:

$$E(\mathbf{w}) = \sum_{i=1}^N (y_i - f_{\mathbf{w}}(\mathbf{x}_i))^2$$

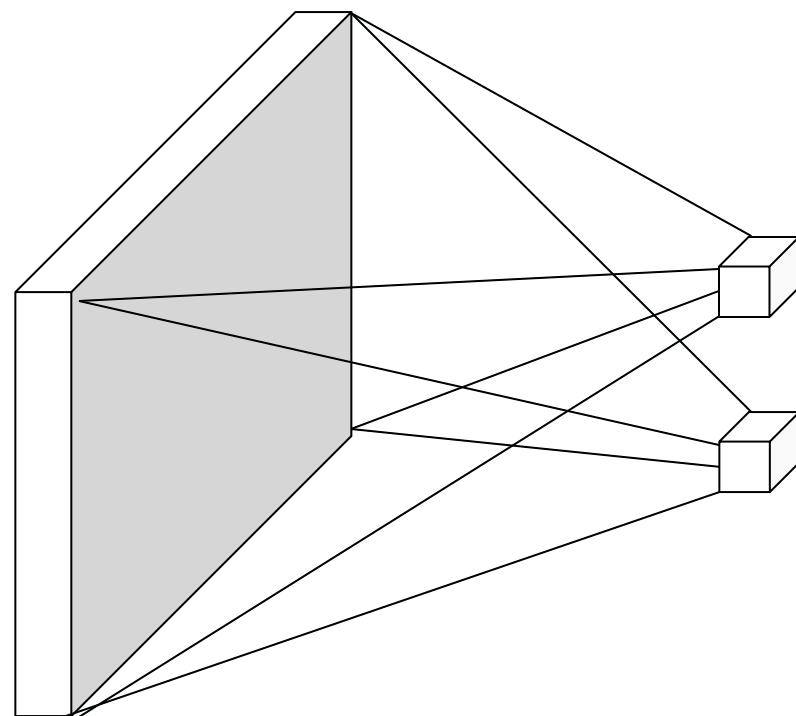
- Update weights by **gradient descent**: $\mathbf{w} \leftarrow \mathbf{w} - \alpha \frac{\partial E}{\partial \mathbf{w}}$
- **Back-propagation**: gradients are computed in the direction from output to input layers and combined using chain rule
- **Stochastic gradient descent**: compute the weight update w.r.t. one training example (or a small batch of examples) at a time, cycle through training examples in random order in multiple epochs

Multi-Layer Network Demo



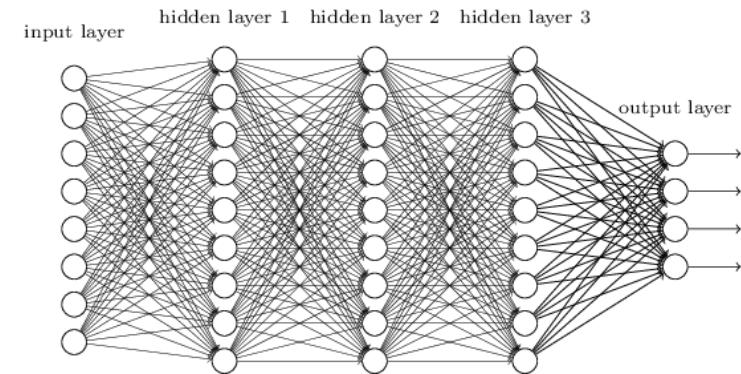
<http://playground.tensorflow.org/>

From fully connected to convolutional networks

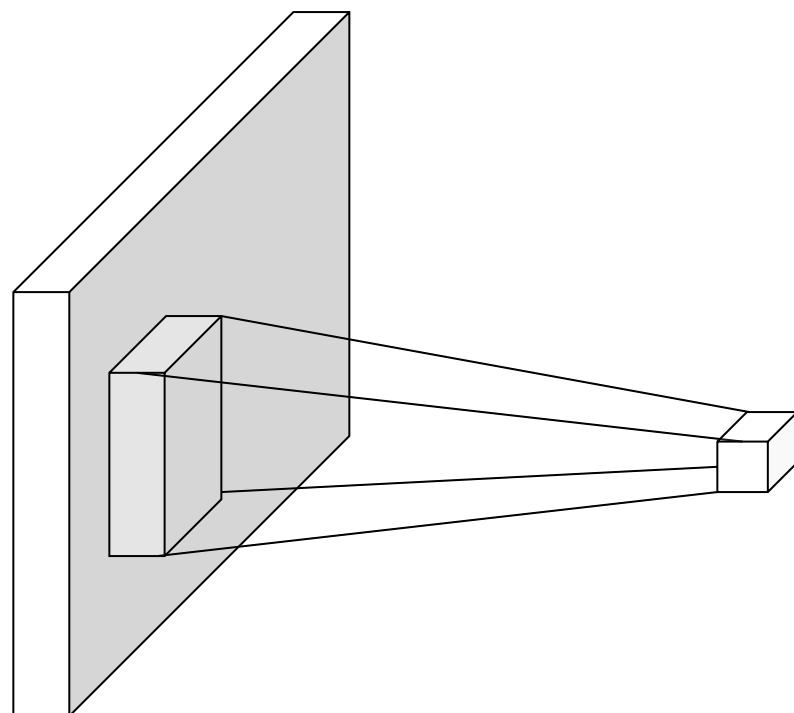


image

Fully connected layer



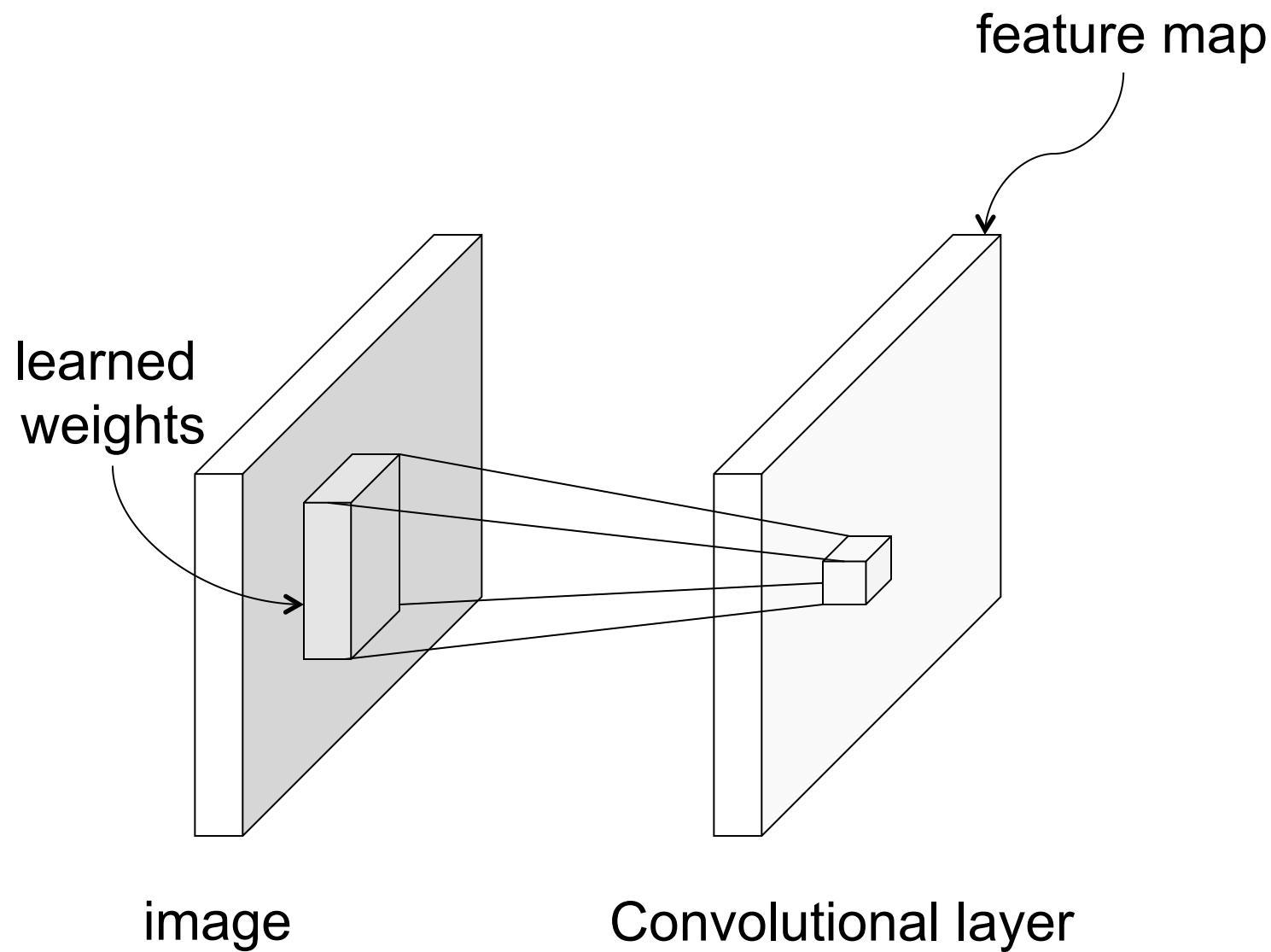
From fully connected to convolutional networks



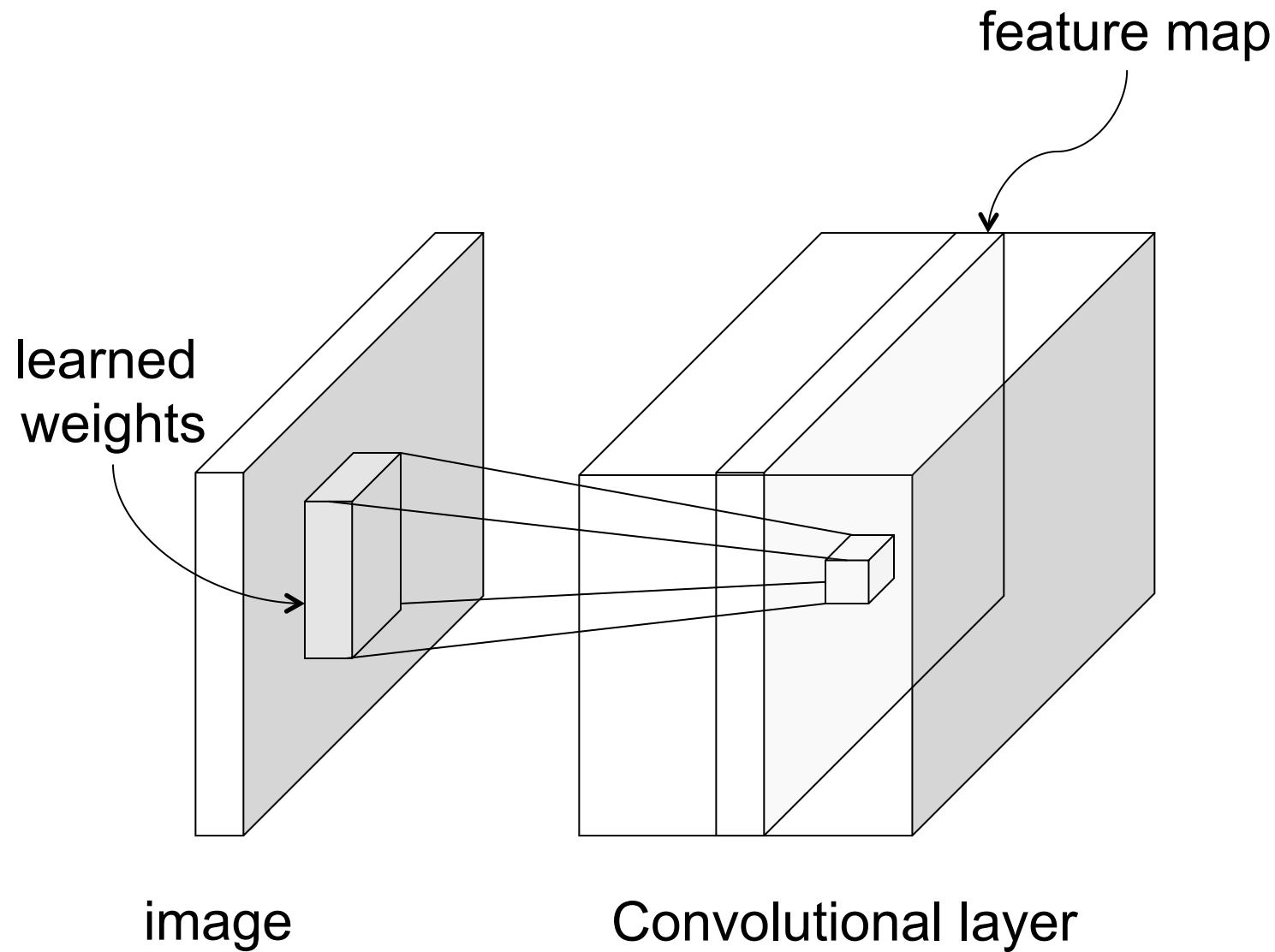
image

Convolutional layer

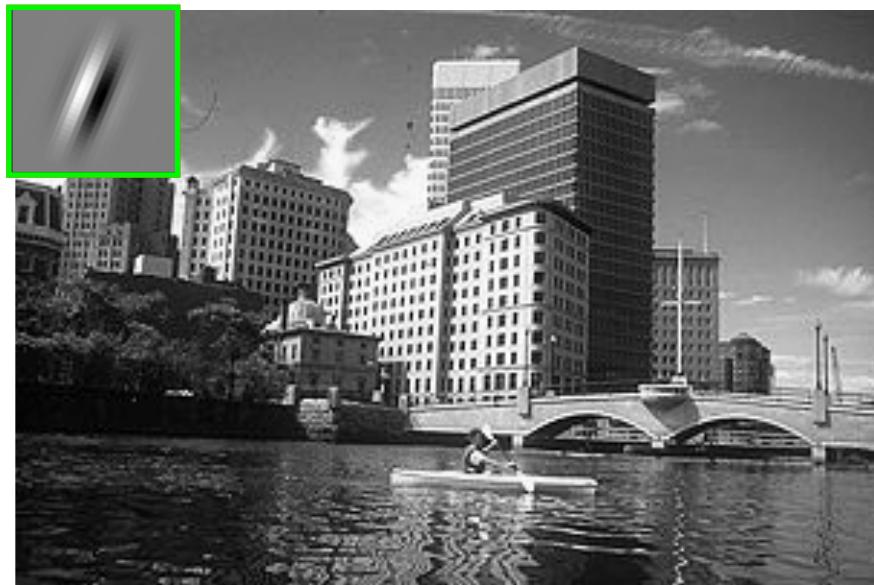
From fully connected to convolutional networks



From fully connected to convolutional networks



Convolution as feature extraction

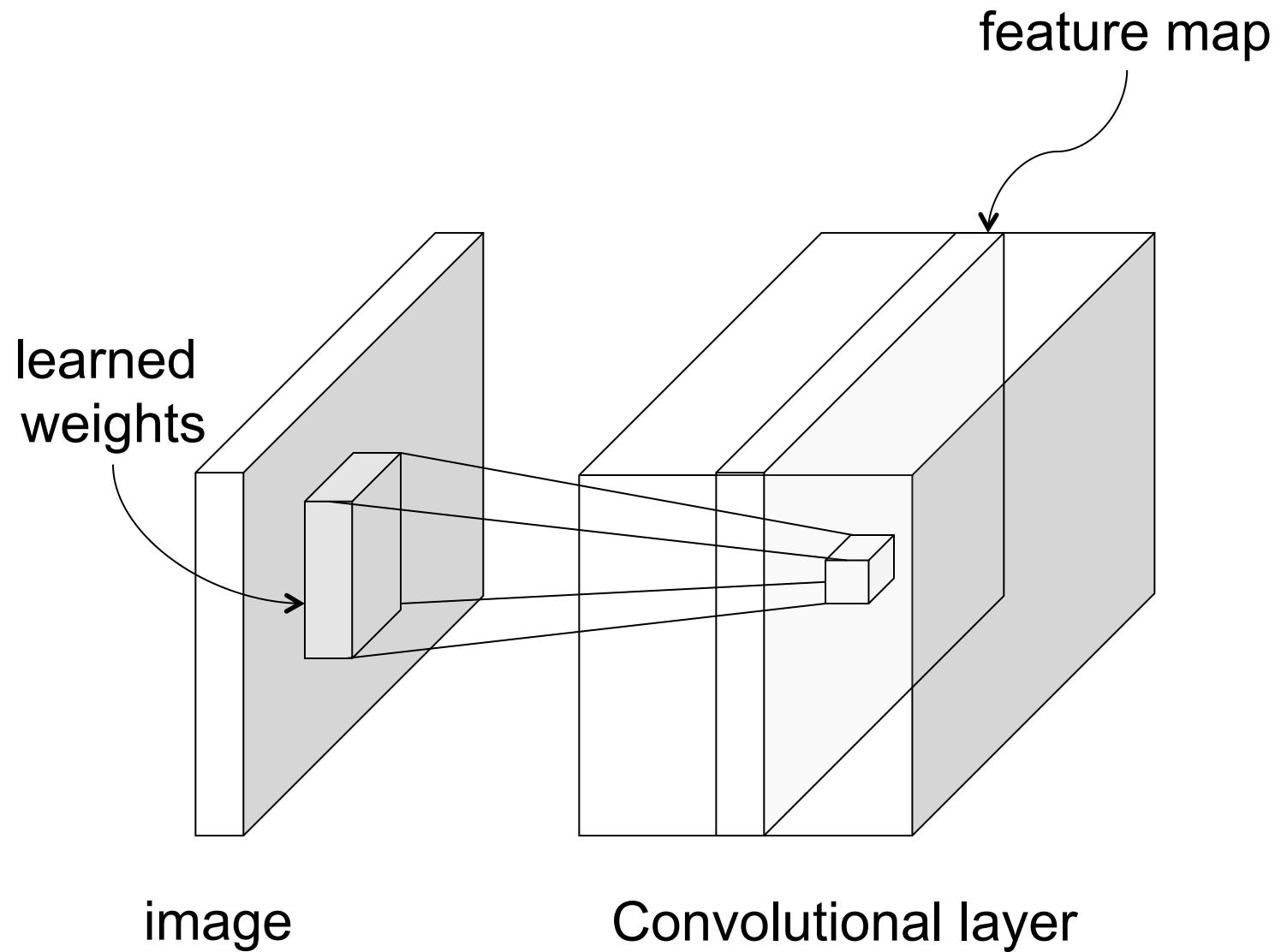


Input

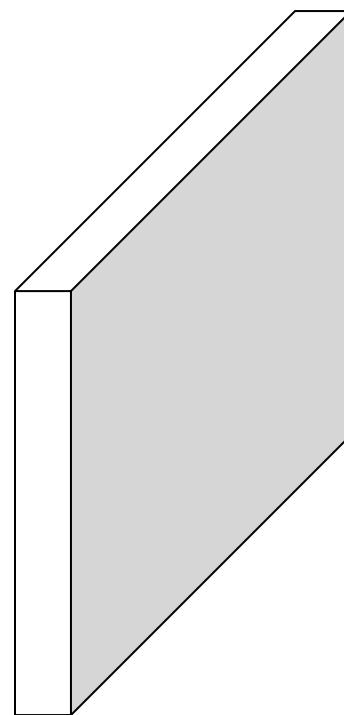


Feature Map

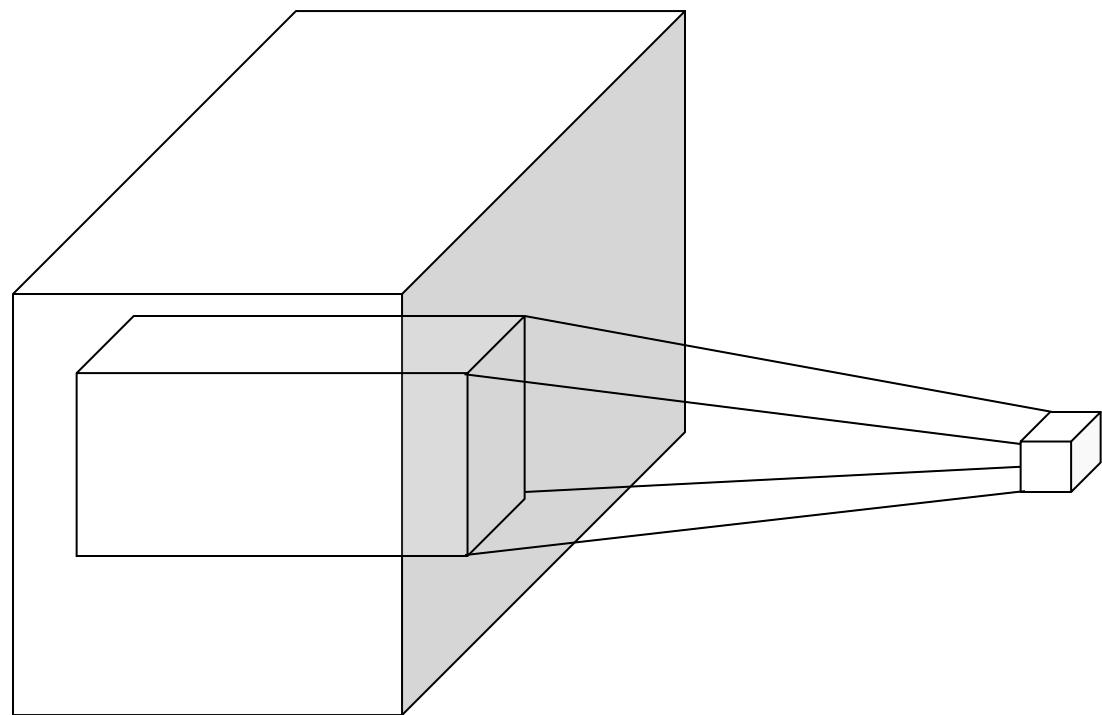
From fully connected to convolutional networks



From fully connected to convolutional networks



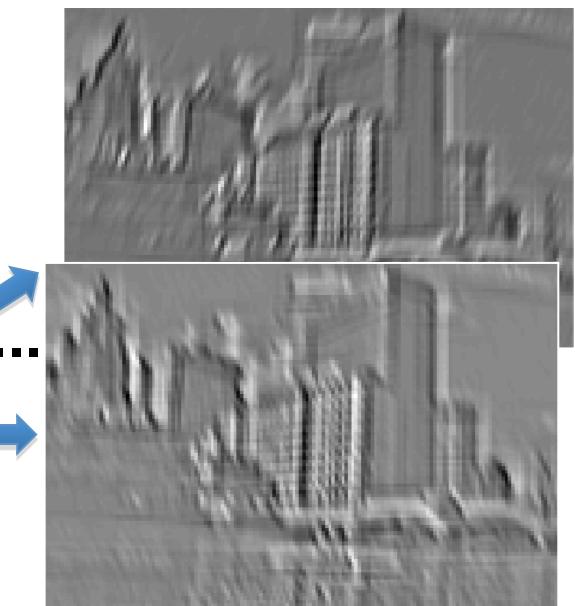
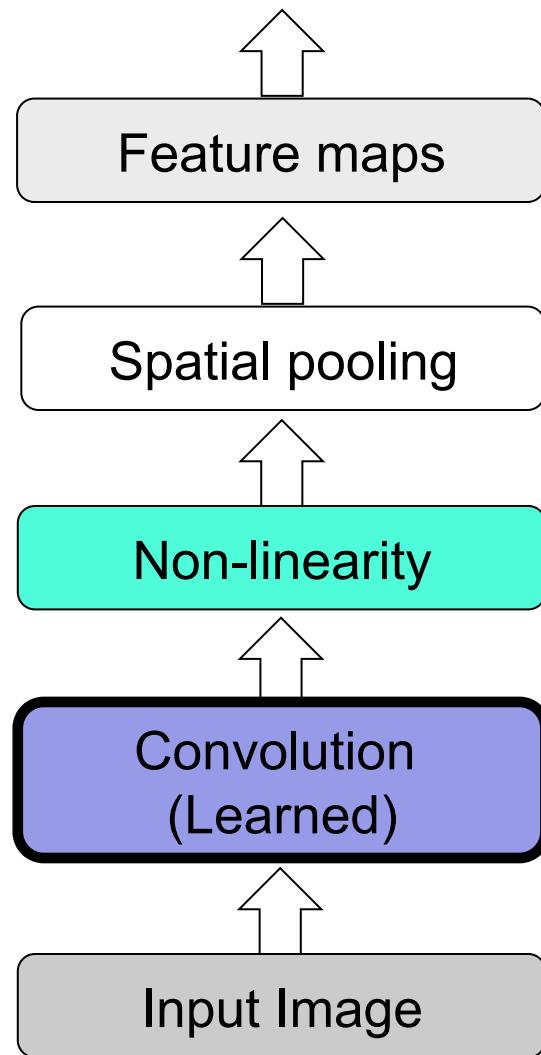
image



Convolutional layer

next
layer

Key operations in a CNN

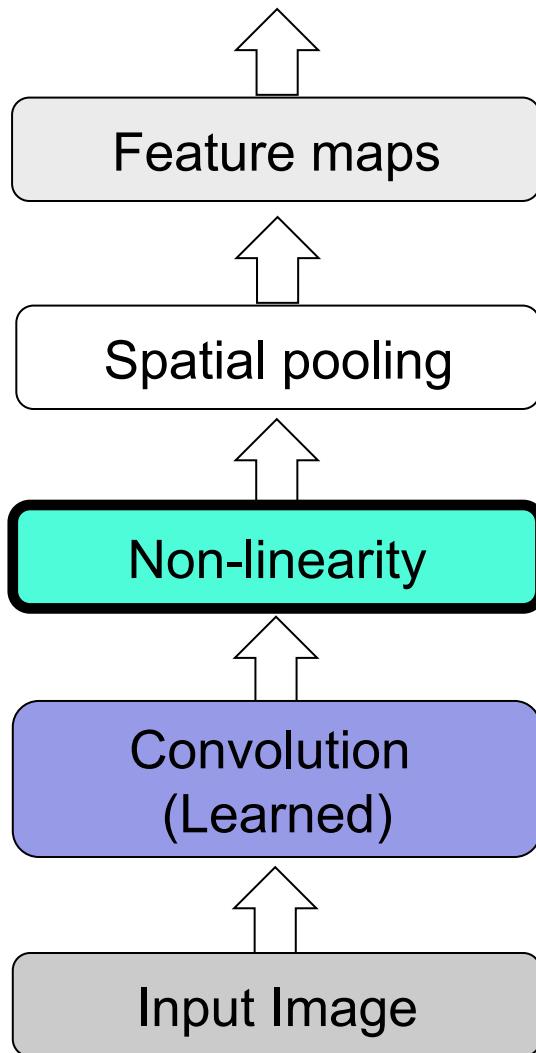


Input

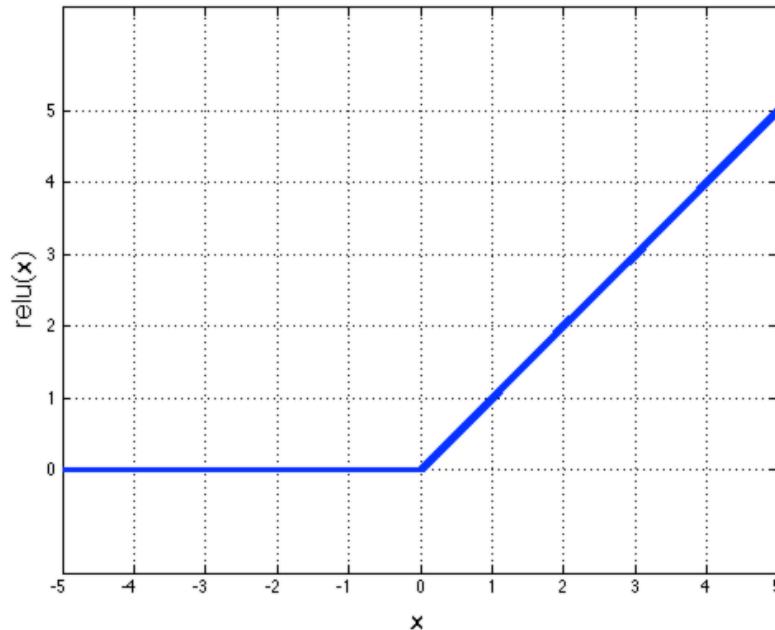
Feature Map

Source: R. Fergus, Y. LeCun

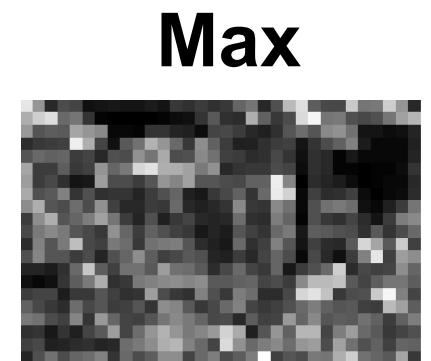
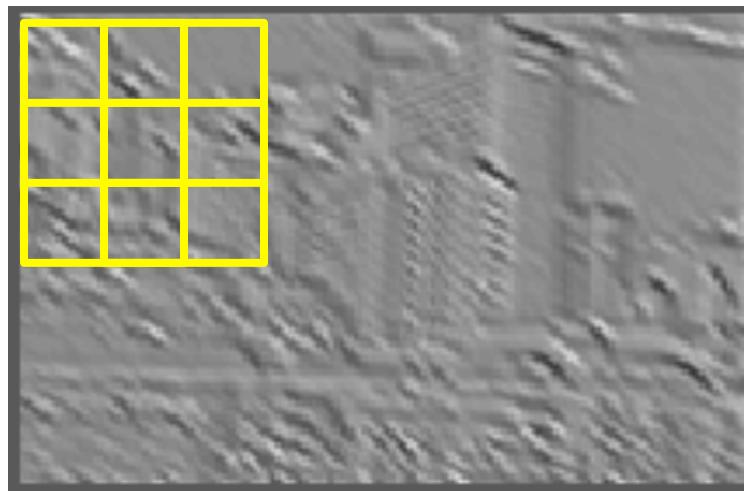
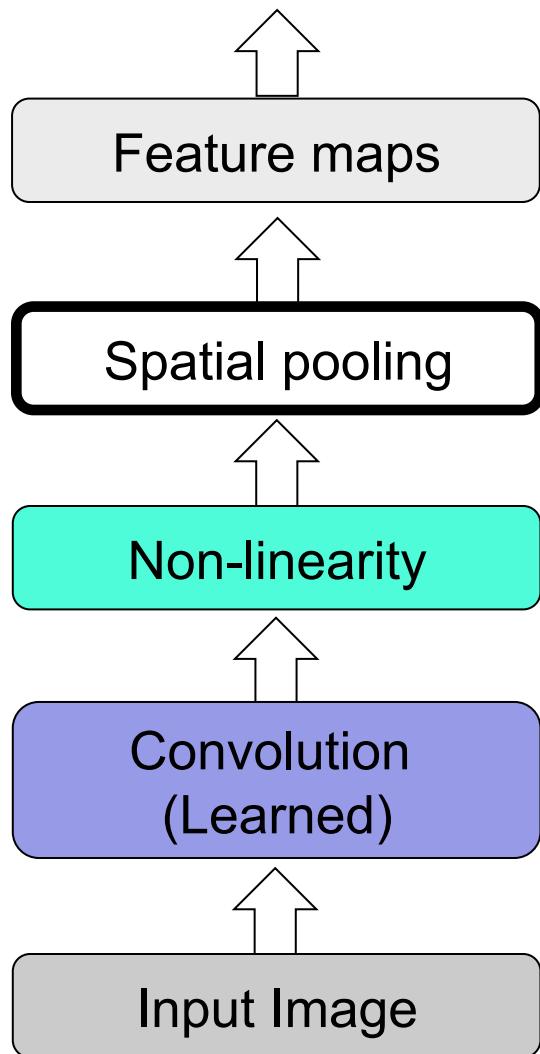
Key operations



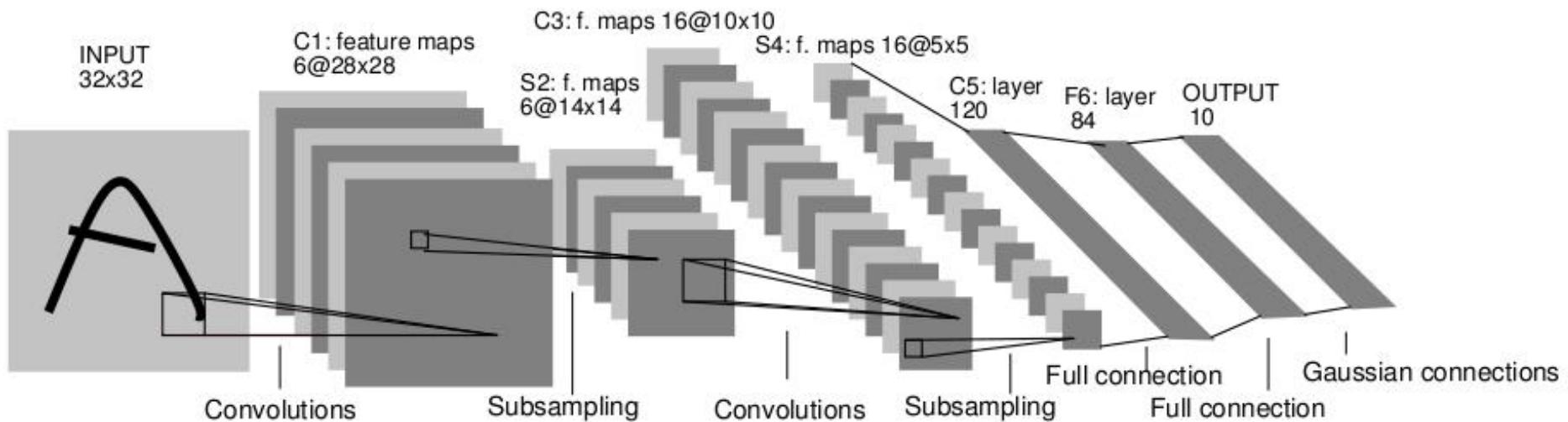
Rectified Linear Unit (ReLU)



Key operations



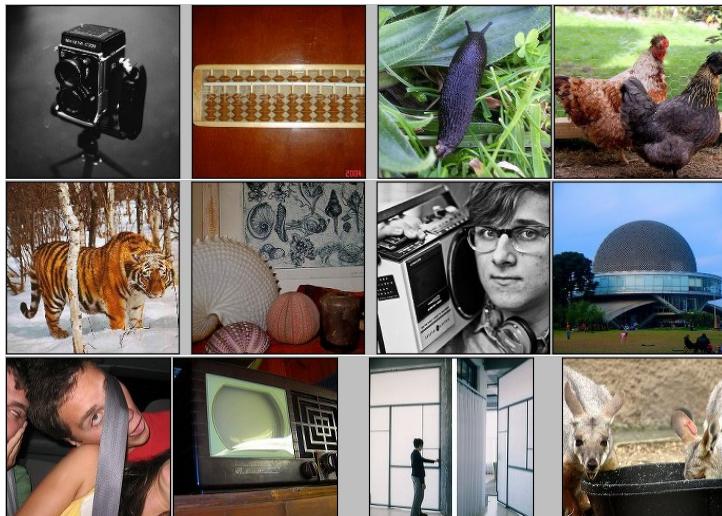
LeNet-5



- Average pooling
- Sigmoid or tanh nonlinearity
- Fully connected layers at the end
- Trained on MNIST digit dataset with 60K training examples

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner,
[Gradient-based learning applied to document recognition](#), Proc. IEEE 86(11): 2278–2324, 1998.

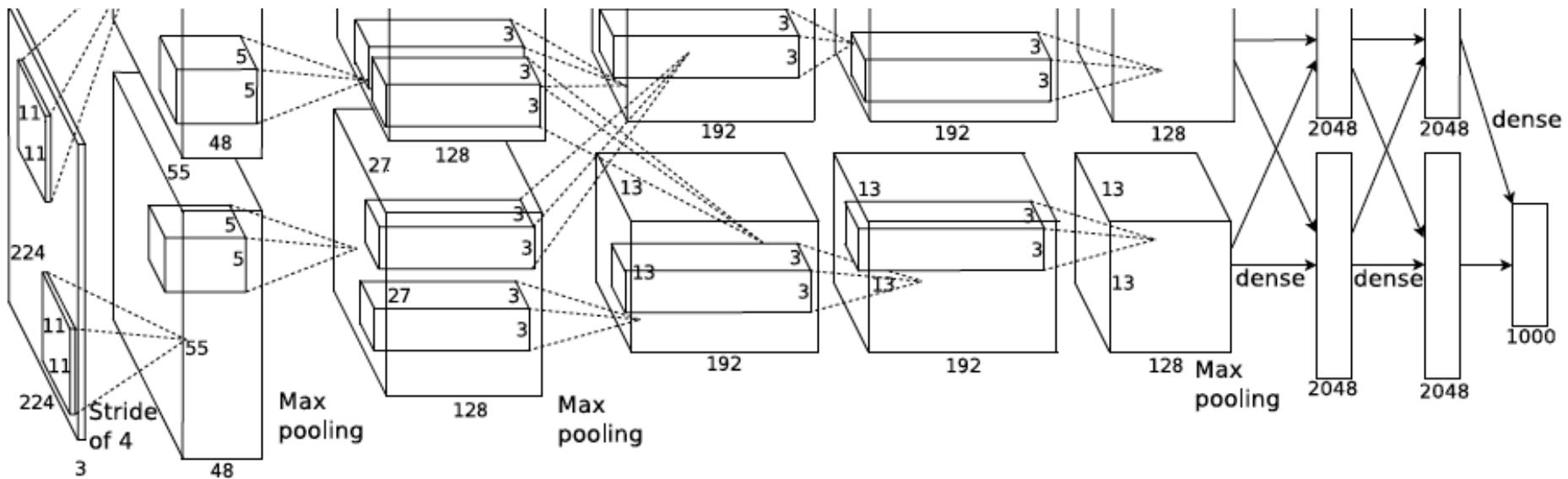
Fast forward to the arrival of big visual data...



- ~14 million labeled images, 20k classes
- Images gathered from Internet
- Human labels via Amazon MTurk
- ImageNet Large-Scale Visual Recognition Challenge (ILSVRC): 1.2 million training images, 1000 classes

www.image-net.org/challenges/LSVRC/

AlexNet: ILSVRC 2012 winner



- Similar framework to LeNet but:
 - Max pooling, ReLU nonlinearity
 - More data and bigger model (7 hidden layers, 650K units, 60M params)
 - GPU implementation (50x speedup over CPU)
 - Trained on two GPUs for a week
 - Dropout regularization

A. Krizhevsky, I. Sutskever, and G. Hinton,

[ImageNet Classification with Deep Convolutional Neural Networks](#), NIPS 2012

Clarifai: ILSVRC 2013 winner

- Refinement of AlexNet

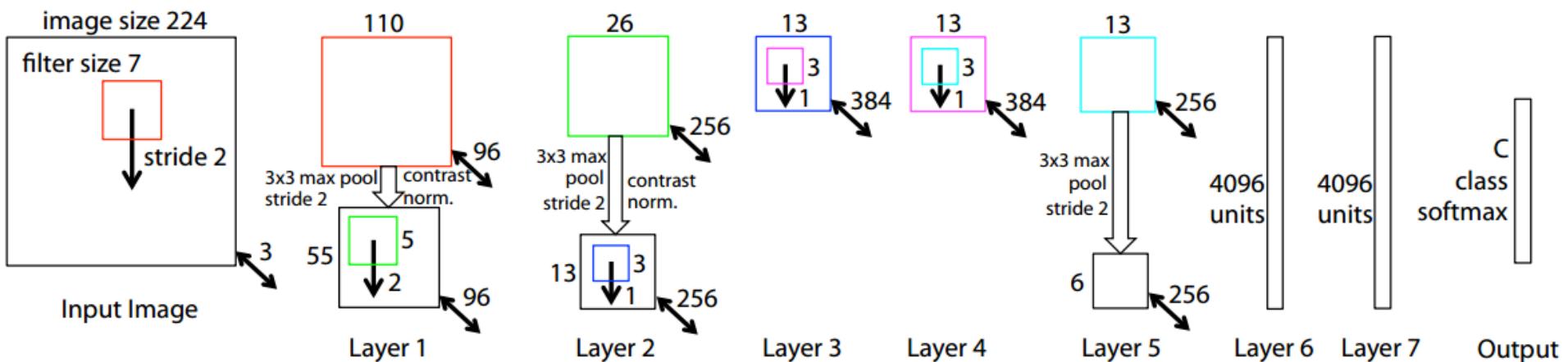


Figure 3. Architecture of our 8 layer convnet model. A 224 by 224 crop of an image (with 3 color planes) is presented as the input. This is convolved with 96 different 1st layer filters (red), each of size 7 by 7, using a stride of 2 in both x and y. The resulting feature maps are then: (i) passed through a rectified linear function (not shown), (ii) pooled (max within 3x3 regions, using stride 2) and (iii) contrast normalized across feature maps to give 96 different 55 by 55 element feature maps. Similar operations are repeated in layers 2,3,4,5. The last two layers are fully connected, taking features from the top convolutional layer as input in vector form ($6 \cdot 6 \cdot 256 = 9216$ dimensions). The final layer is a C -way softmax function, C being the number of classes. All filters and feature maps are square in shape.

M. Zeiler and R. Fergus, [Visualizing and Understanding Convolutional Networks](#),
ECCV 2014 (Best Paper Award winner)

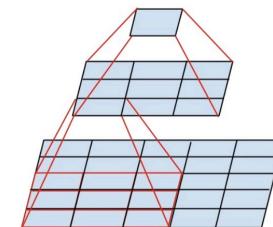
VGGNet: ILSVRC 2014 2nd place

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256	conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv1-256	conv3-256 conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512	conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv1-512	conv3-512 conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512	conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv1-512	conv3-512 conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

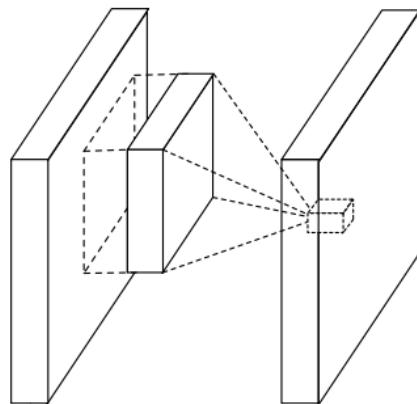
Table 2: Number of parameters (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

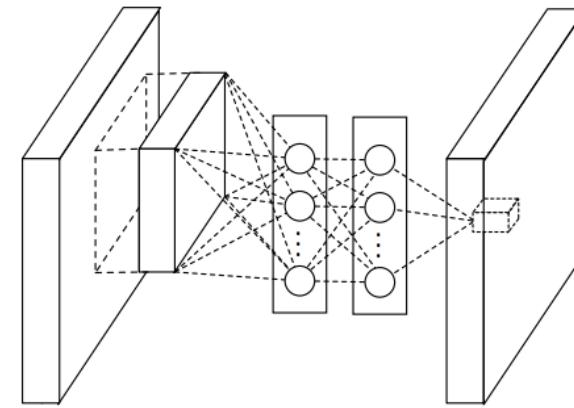
- Sequence of deeper networks trained progressively
- Large receptive fields replaced by successive layers of 3x3 convolutions (with ReLU in between)
- One 7x7 conv layer with C feature maps needs $49C^2$ weights, three 3x3 conv layers need only $27C^2$ weights
- Experimented with 1x1 convolutions



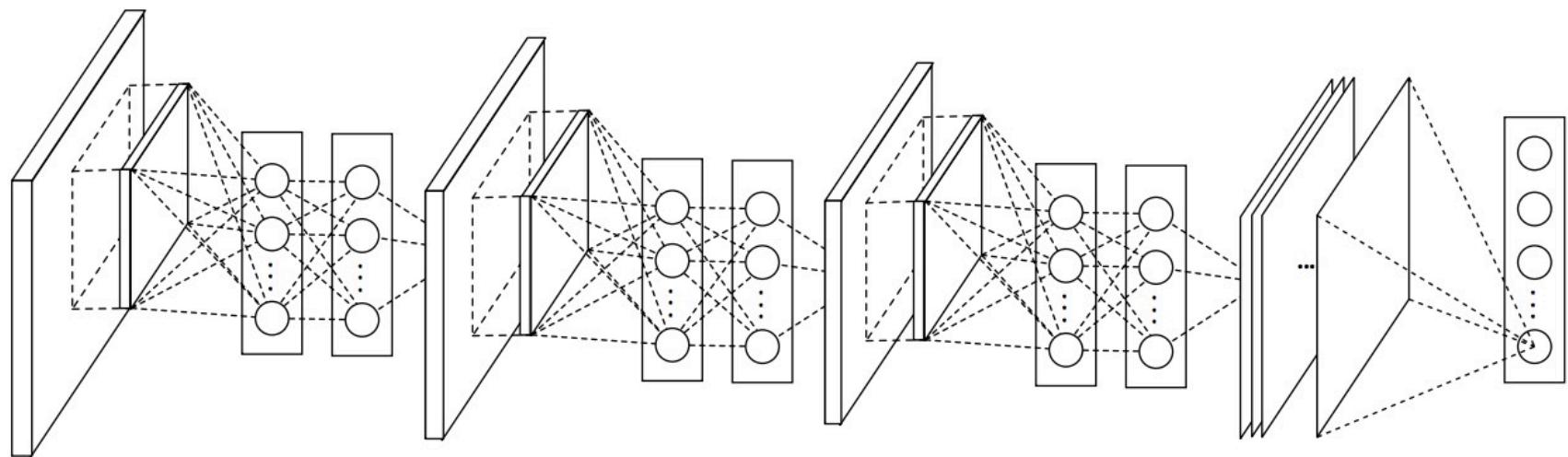
Network in network



(a) Linear convolution layer

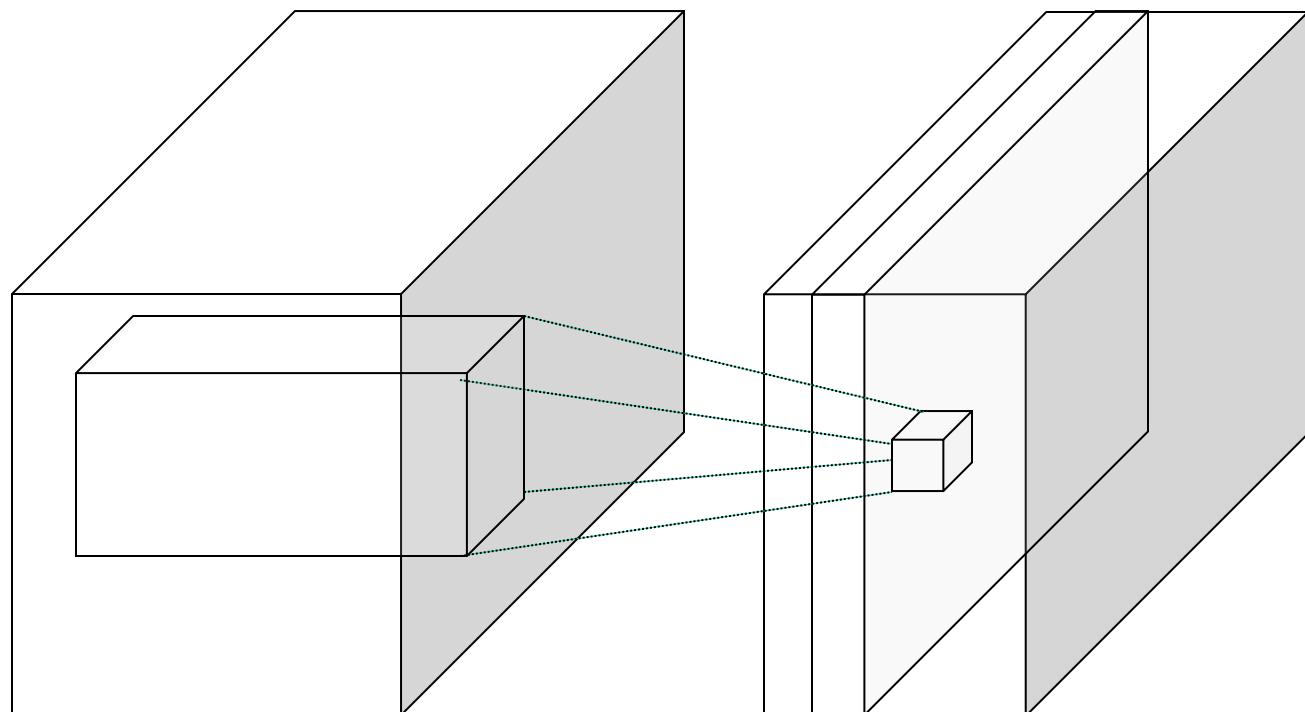


(b) Mlpconv layer



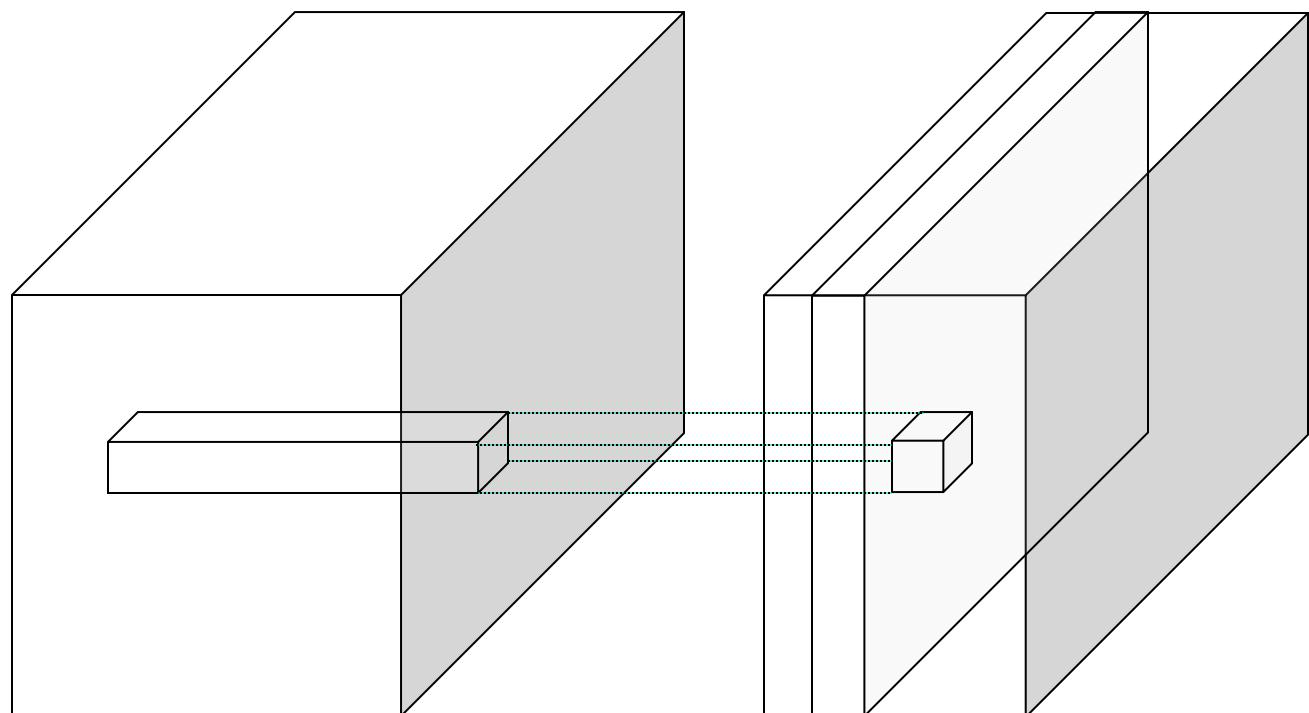
M. Lin, Q. Chen, and S. Yan, [Network in network](#), ICLR 2014

1x1 convolutions



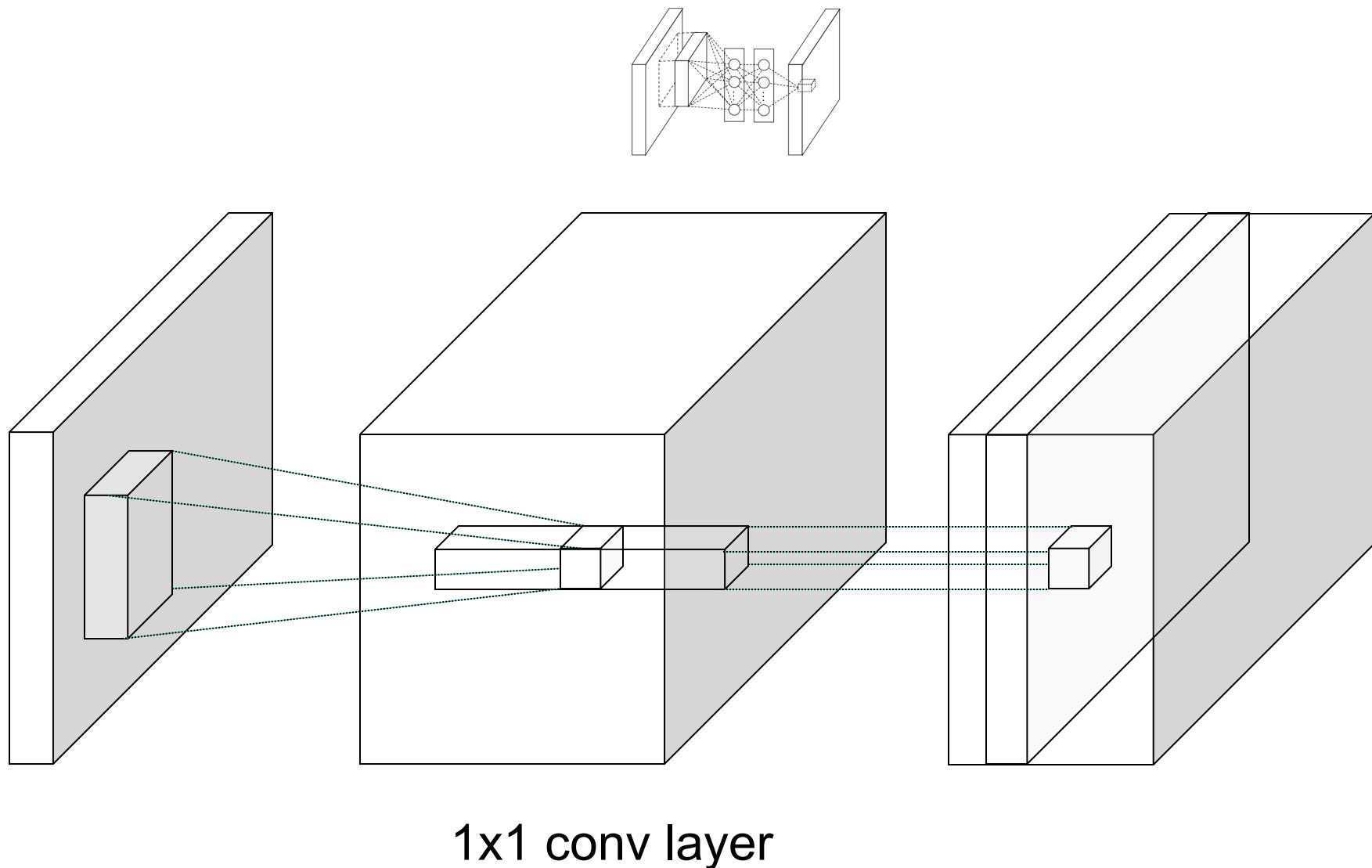
conv layer

1x1 convolutions



1x1 conv layer

1x1 convolutions



GoogLeNet: ILSVRC 2014 winner

- The Inception Module

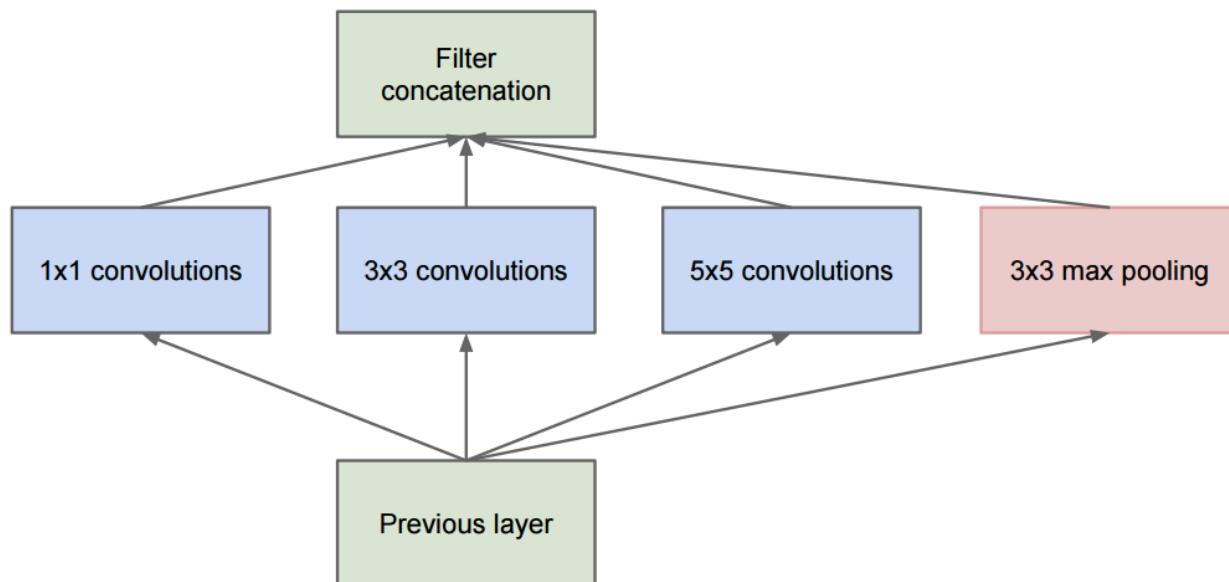


<http://knowyourmeme.com/memes/we-need-to-go-deeper>

C. Szegedy et al., [Going deeper with convolutions](#), CVPR 2015

GoogLeNet

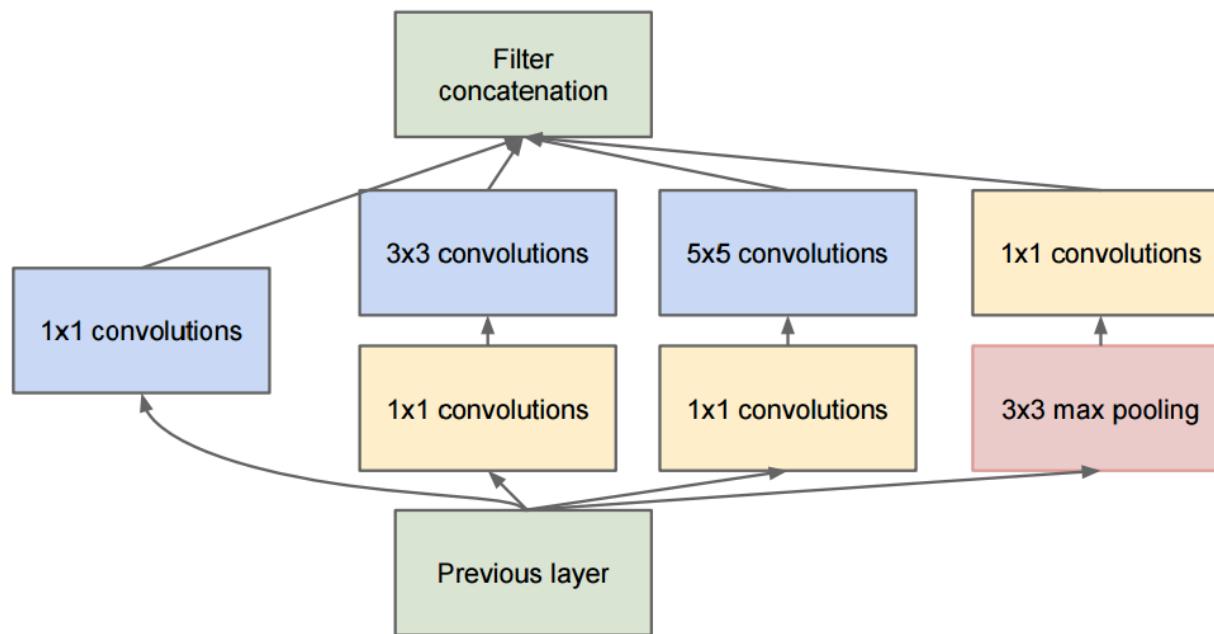
- The Inception Module
 - Parallel paths with different receptive field sizes and operations are meant to capture sparse patterns of correlations in the stack of feature maps



C. Szegedy et al., [Going deeper with convolutions](#), CVPR 2015

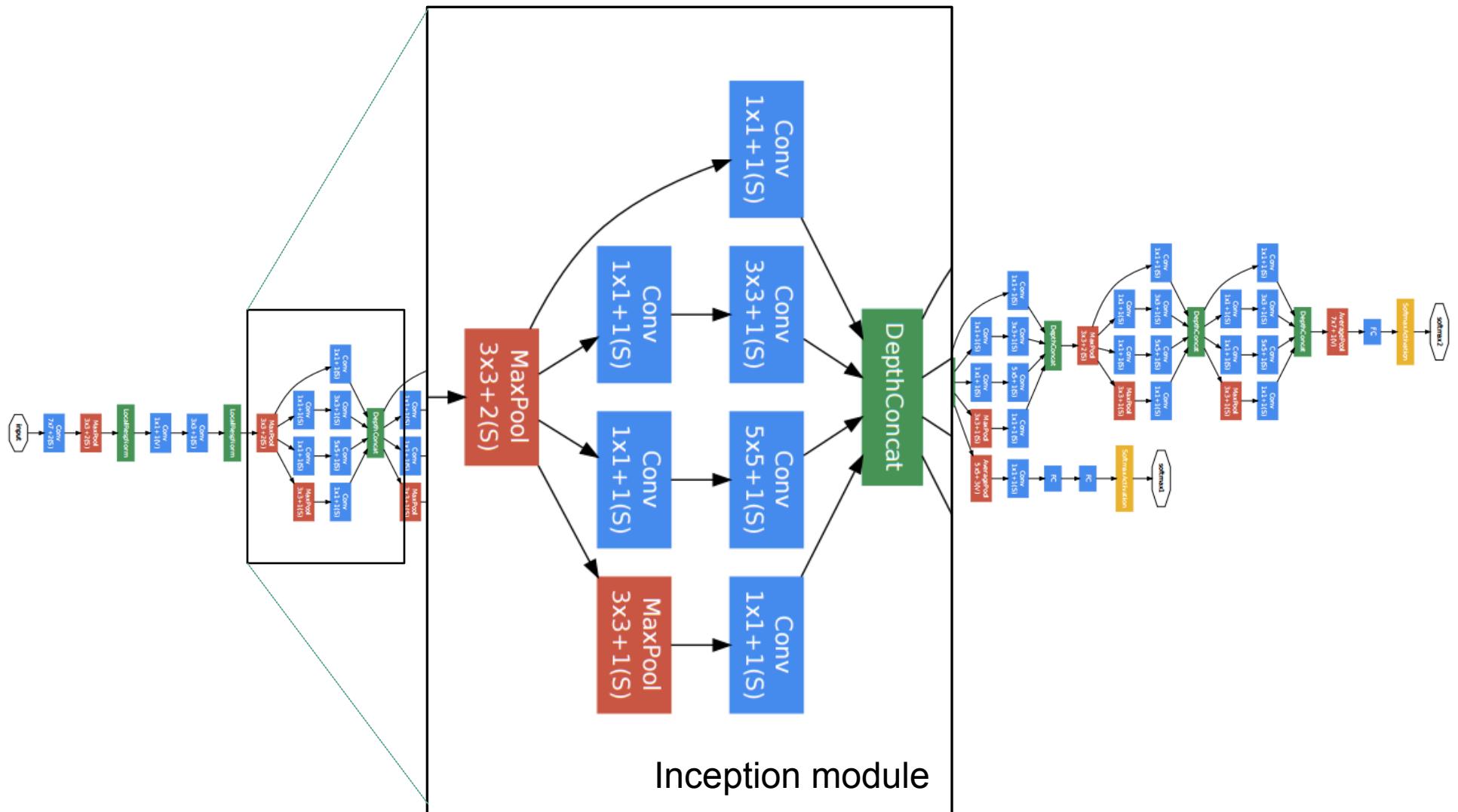
GoogLeNet

- The Inception Module
 - Parallel paths with different receptive field sizes and operations are meant to capture sparse patterns of correlations in the stack of feature maps
 - Use 1×1 convolutions for dimensionality reduction before expensive convolutions



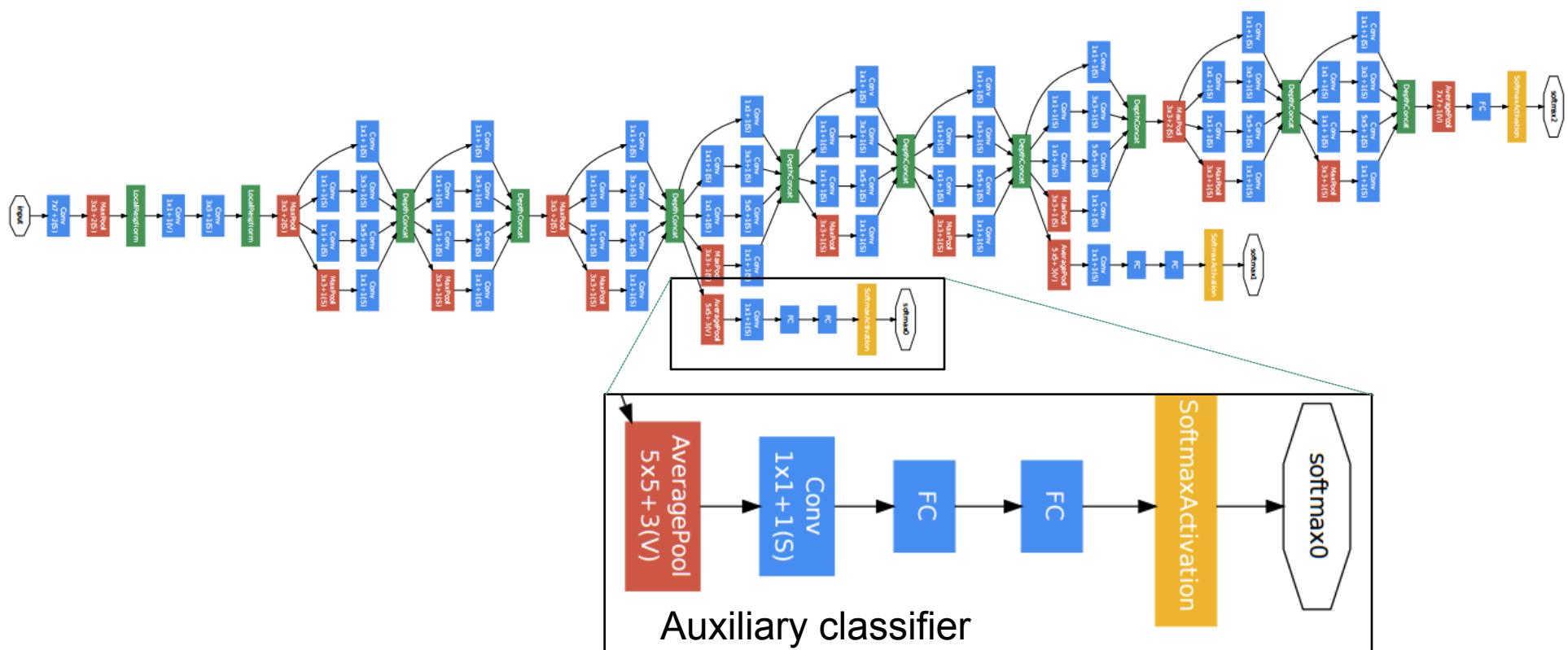
C. Szegedy et al., [Going deeper with convolutions](#), CVPR 2015

GoogLeNet



C. Szegedy et al., [Going deeper with convolutions](#), CVPR 2015

GoogLeNet



C. Szegedy et al., [Going deeper with convolutions](#), CVPR 2015

GoogLeNet

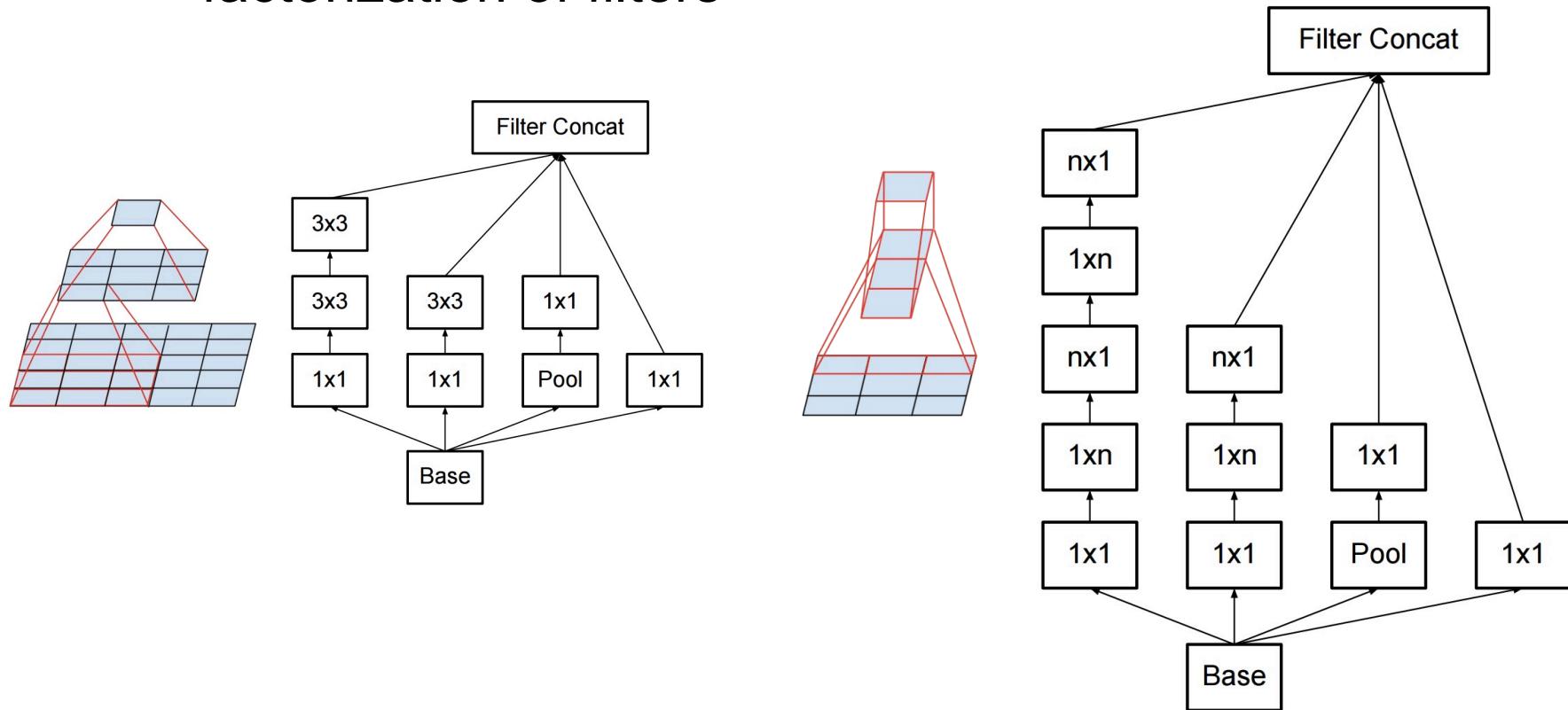
An alternative view:

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

C. Szegedy et al., [Going deeper with convolutions](#), CVPR 2015

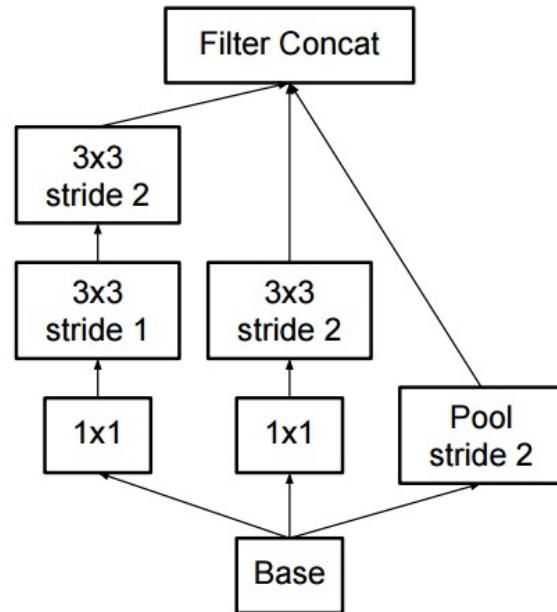
Inception v2, v3

- Regularize training with batch normalization, reducing importance of auxiliary classifiers
- More variants of inception modules with aggressive factorization of filters



Inception v2, v3

- Regularize training with [batch normalization](#), reducing importance of auxiliary classifiers
- More variants of inception modules with aggressive factorization of filters
- Increase the number of feature maps while decreasing spatial resolution (pooling)



ResNet: ILSVRC 2015 winner

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



ResNet, **152 layers**
(ILSVRC 2015)



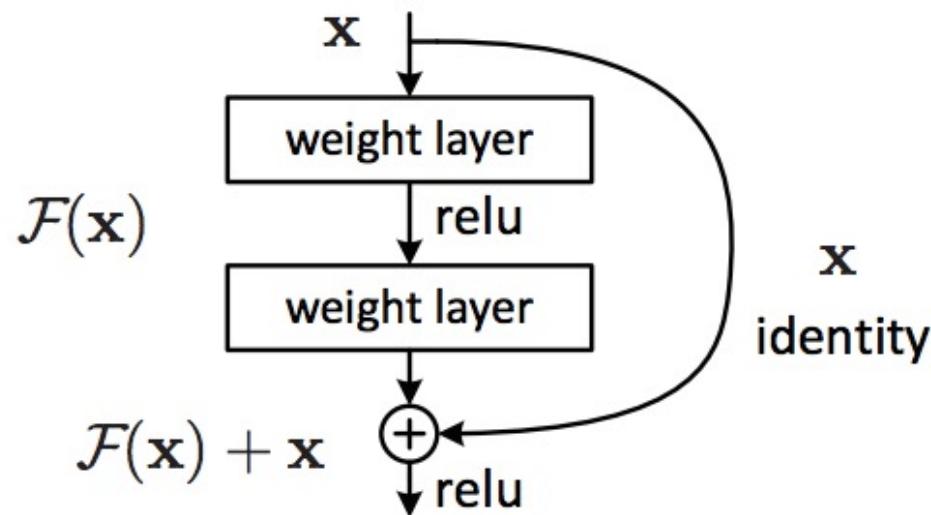
Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun,
[Deep Residual Learning for Image Recognition](#), CVPR 2016



[Source \(?\)](#)

ResNet

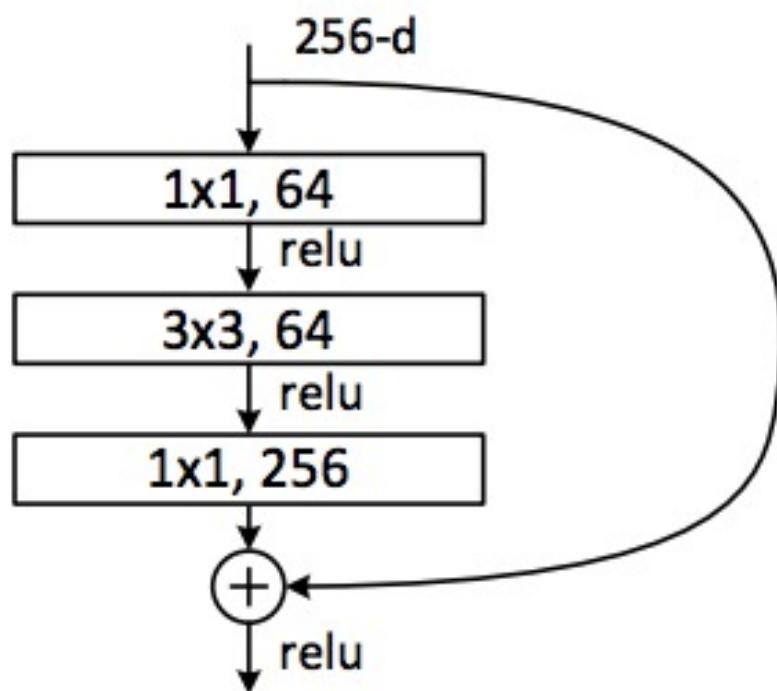
- The residual module
 - Introduce *skip* or *shortcut* connections (existing before in various forms in literature)
 - Make it easy for network layers to represent the identity mapping
 - For some reason, need to skip at least two layers



Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun,
[Deep Residual Learning for Image Recognition](#), CVPR 2016 (Best Paper)

ResNet

Deeper residual module (bottleneck)



- Directly performing 3x3 convolutions with 256 feature maps at input and output:
 $256 \times 256 \times 3 \times 3 \sim 600K$ operations
- Using 1x1 convolutions to reduce 256 to 64 feature maps, followed by 3x3 convolutions, followed by 1x1 convolutions to expand back to 256 maps:
 $256 \times 64 \times 1 \times 1 \sim 16K$
 $64 \times 64 \times 3 \times 3 \sim 36K$
 $64 \times 256 \times 1 \times 1 \sim 16K$
Total: $\sim 70K$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun,
[Deep Residual Learning for Image Recognition](#), CVPR 2016 (Best Paper)

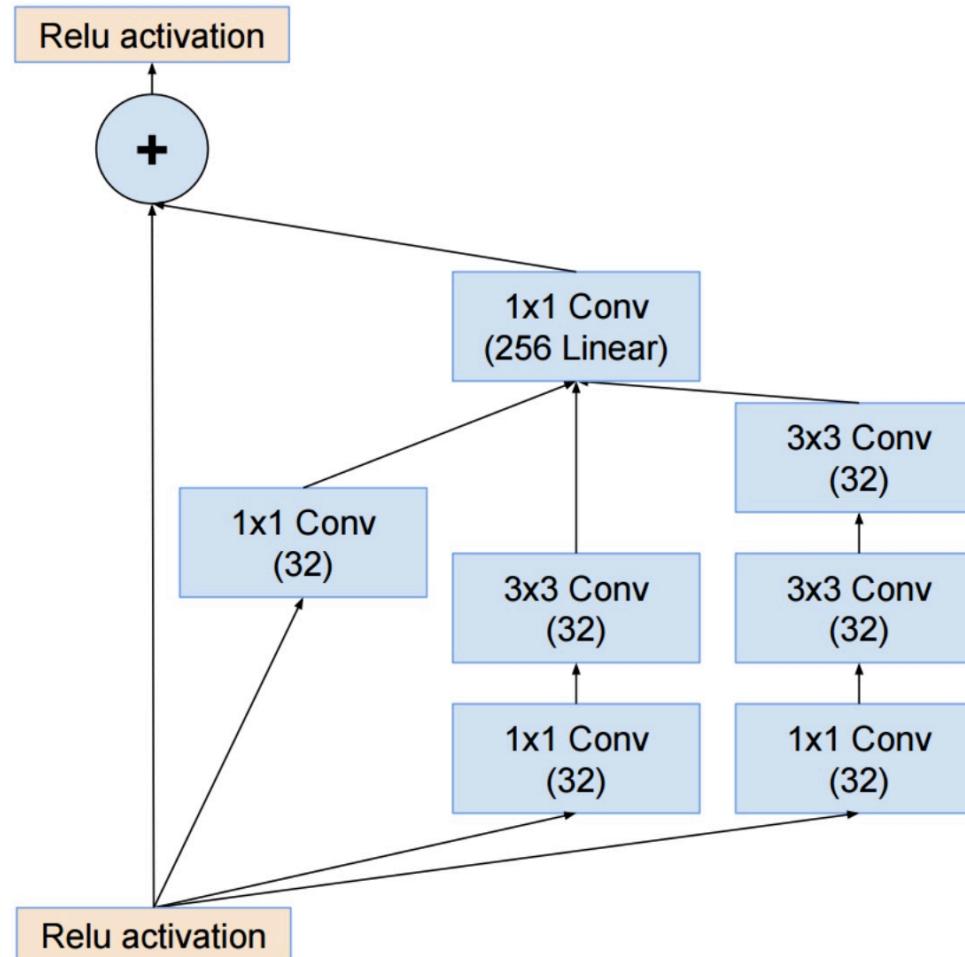
ResNet

Architectures for ImageNet:

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112			7×7, 64, stride 2		
conv2_x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun,
[Deep Residual Learning for Image Recognition](#), CVPR 2016 (Best Paper)

Inception v4



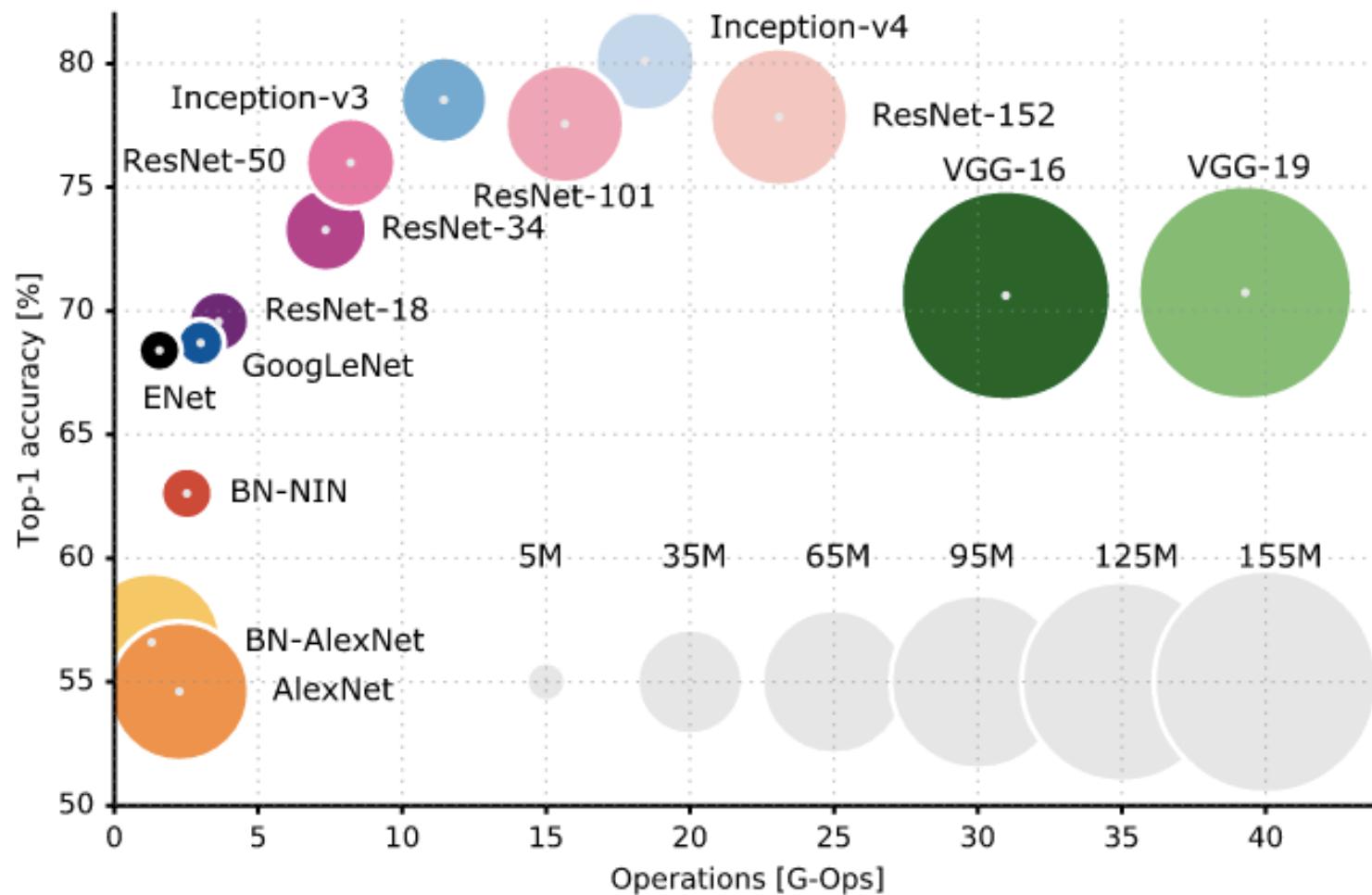
C. Szegedy et al.,
[Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,](#)
arXiv 2016

Summary: ILSVRC 2012-2015

Team	Year	Place	Error (top-5)	External data
SuperVision – Toronto (AlexNet, 7 layers)	2012	-	16.4%	no
SuperVision	2012	1st	15.3%	ImageNet 22k
Clarifai – NYU (7 layers)	2013	-	11.7%	no
Clarifai	2013	1st	11.2%	ImageNet 22k
VGG – Oxford (16 layers)	2014	2nd	7.32%	no
GoogLeNet (19 layers)	2014	1st	6.67%	no
ResNet (152 layers)	2015	1st	3.57%	
Human expert*			5.1%	

<http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/>

Accuracy vs. efficiency



<https://culurciello.github.io/tech/2016/06/04/nets.html>

Design principles

- Reduce filter sizes (except possibly at the lowest layer), factorize filters aggressively
- Use 1x1 convolutions to reduce and expand the number of feature maps judiciously
- Use skip connections and/or create multiple paths through the network

What's missing from the picture?

- Training tricks and details: initialization, regularization, normalization
 - Training data augmentation
 - Averaging classifier outputs over multiple crops/flips
 - Ensembles of networks
-
- What about [ILSVRC 2016](#)?
 - No more ImageNet classification
 - No breakthroughs comparable to ResNet

Reading list

- <https://culurciello.github.io/tech/2016/06/04/nets.html>
- Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner,
[Gradient-based learning applied to document recognition](#), Proc. IEEE 86(11): 2278–2324, 1998.
- A. Krizhevsky, I. Sutskever, and G. Hinton,
[ImageNet Classification with Deep Convolutional Neural Networks](#), NIPS 2012
- M. Zeiler and R. Fergus, [Visualizing and Understanding Convolutional Networks](#), ECCV 2014
- K. Simonyan and A. Zisserman,
[Very Deep Convolutional Networks for Large-Scale Image Recognition](#), ICLR 2015
- M. Lin, Q. Chen, and S. Yan, [Network in network](#), ICLR 2014
- C. Szegedy et al., [Going deeper with convolutions](#), CVPR 2015
- C. Szegedy et al., [Rethinking the inception architecture for computer vision](#), CVPR 2016
- K. He, X. Zhang, S. Ren, and J. Sun,
[Deep Residual Learning for Image Recognition](#), CVPR 2016