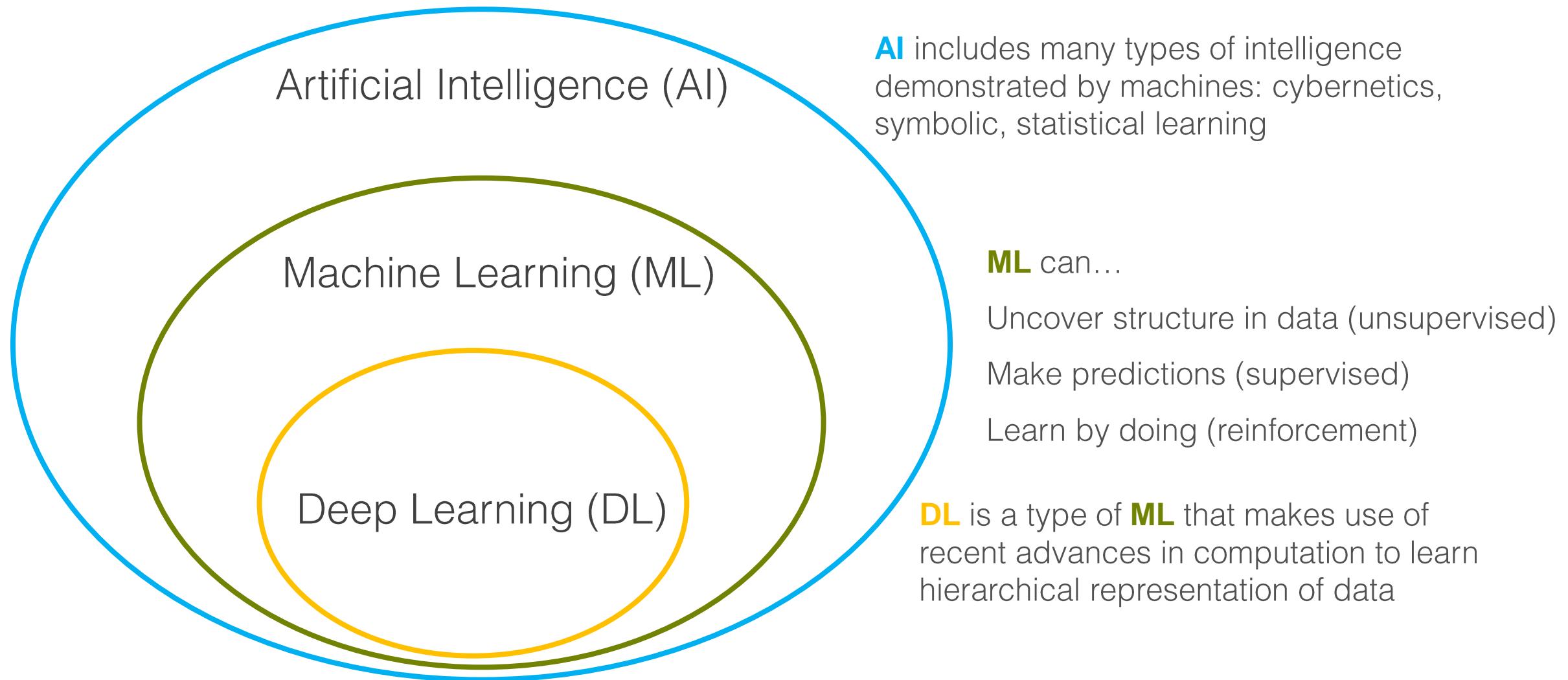


# Deep Learning

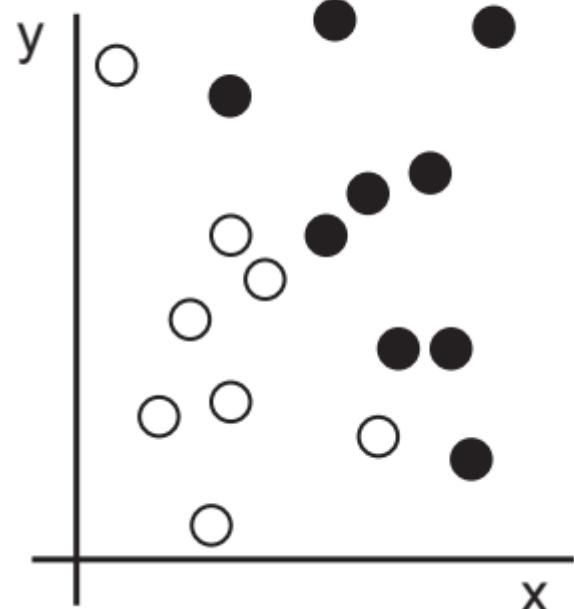
Lecture 18

# Hierarchy of Learning

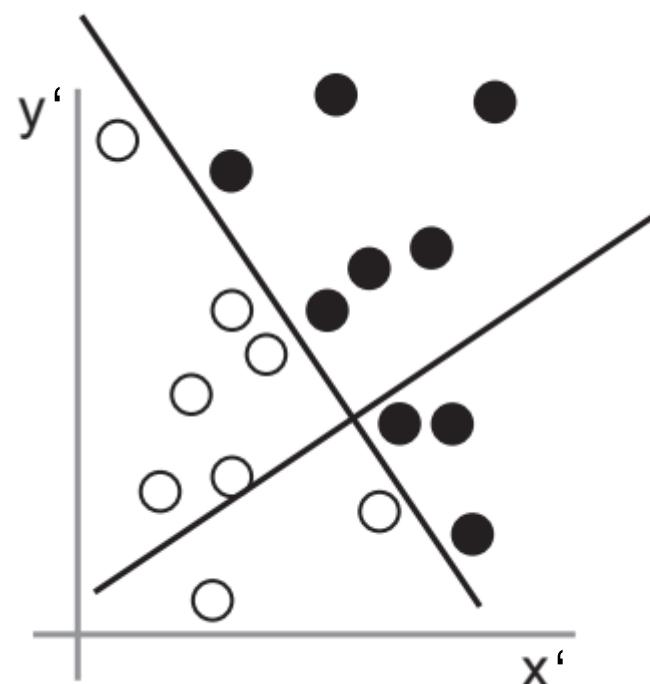


# Alternate representations of data

Raw data



Coordinate change



Improved representation

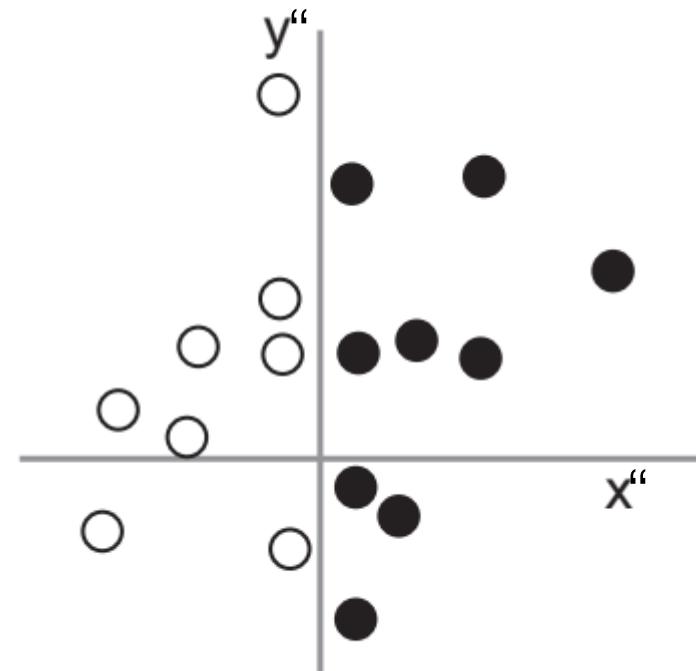
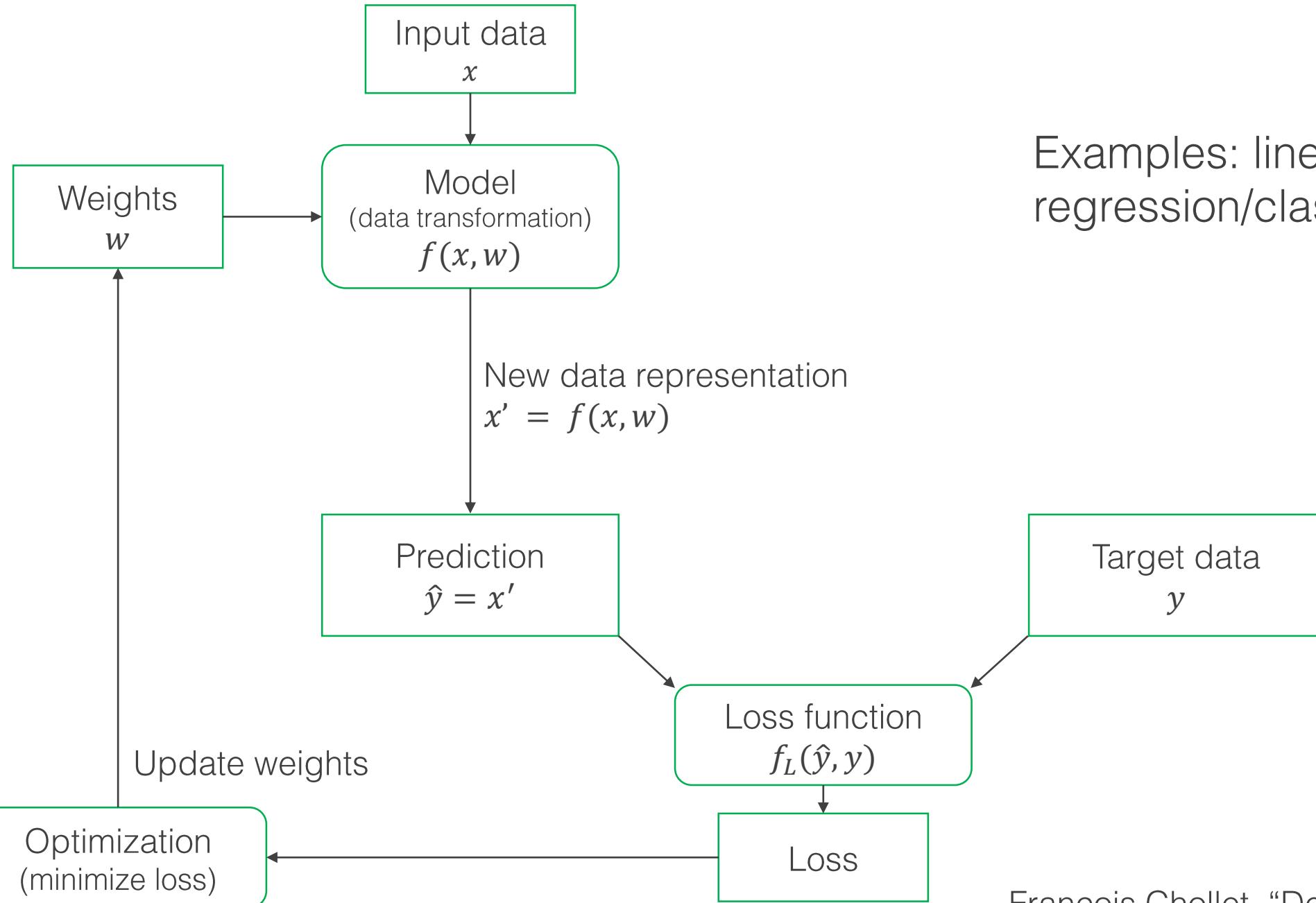
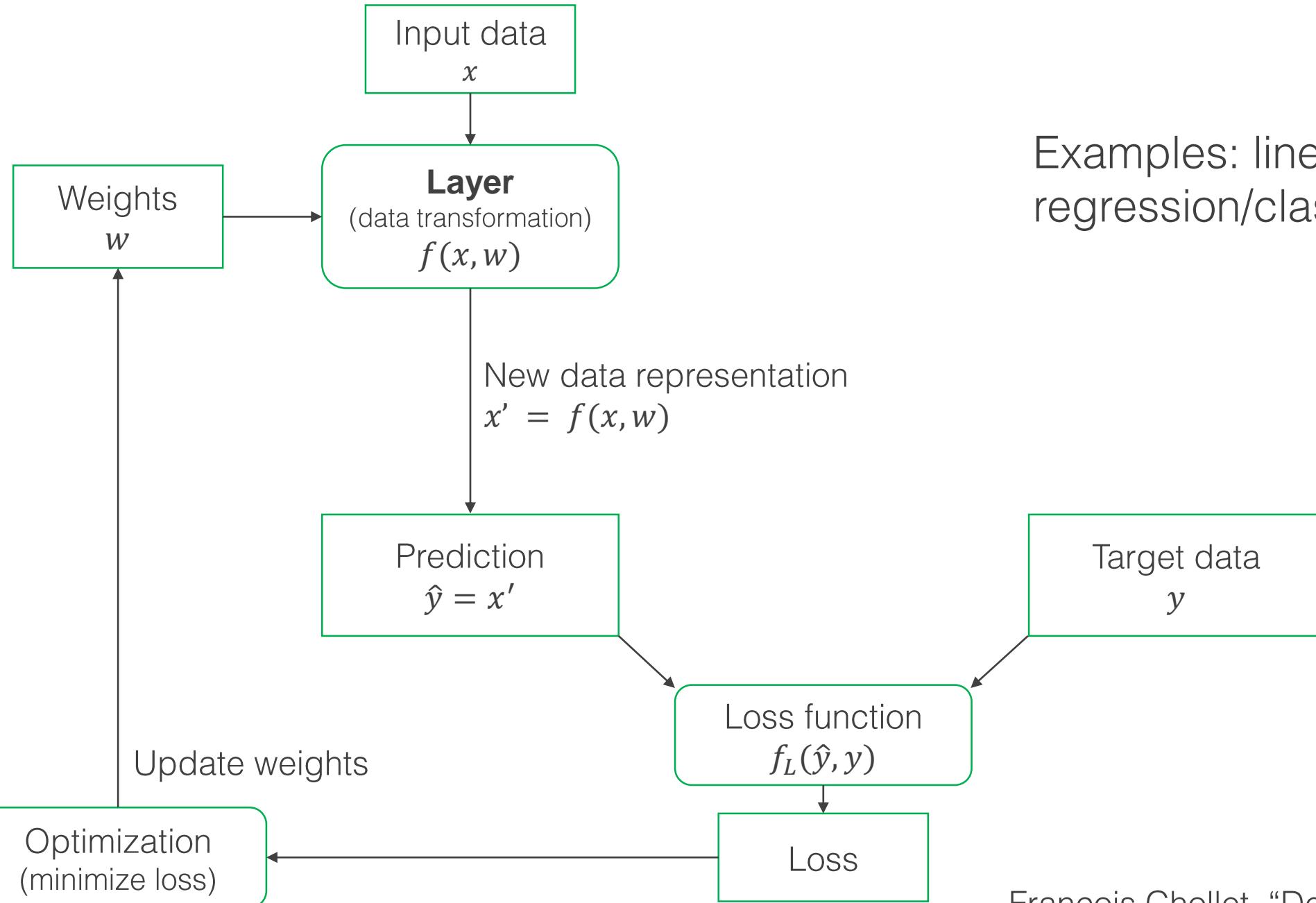


Figure from Francois Chollet, "Deep Learning with Python."



Examples: linear/logistic regression/classification

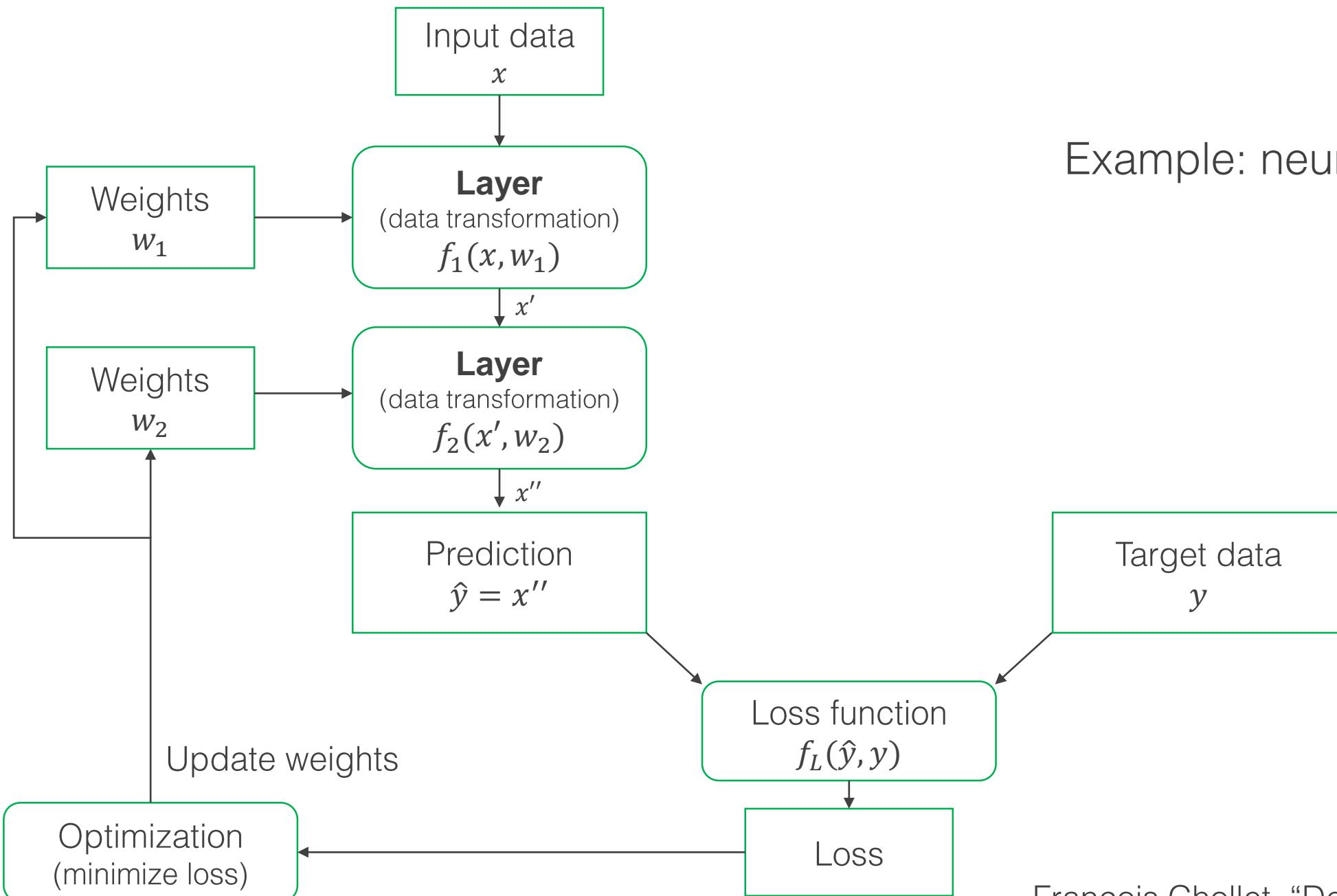
Francois Chollet, "Deep Learning with Python."



Examples: linear/logistic regression/classification

Francois Chollet, "Deep Learning with Python."

## Example: neural network



Francois Chollet, "Deep Learning with Python."

# Types of Deep Learning Tools

Autoencoders

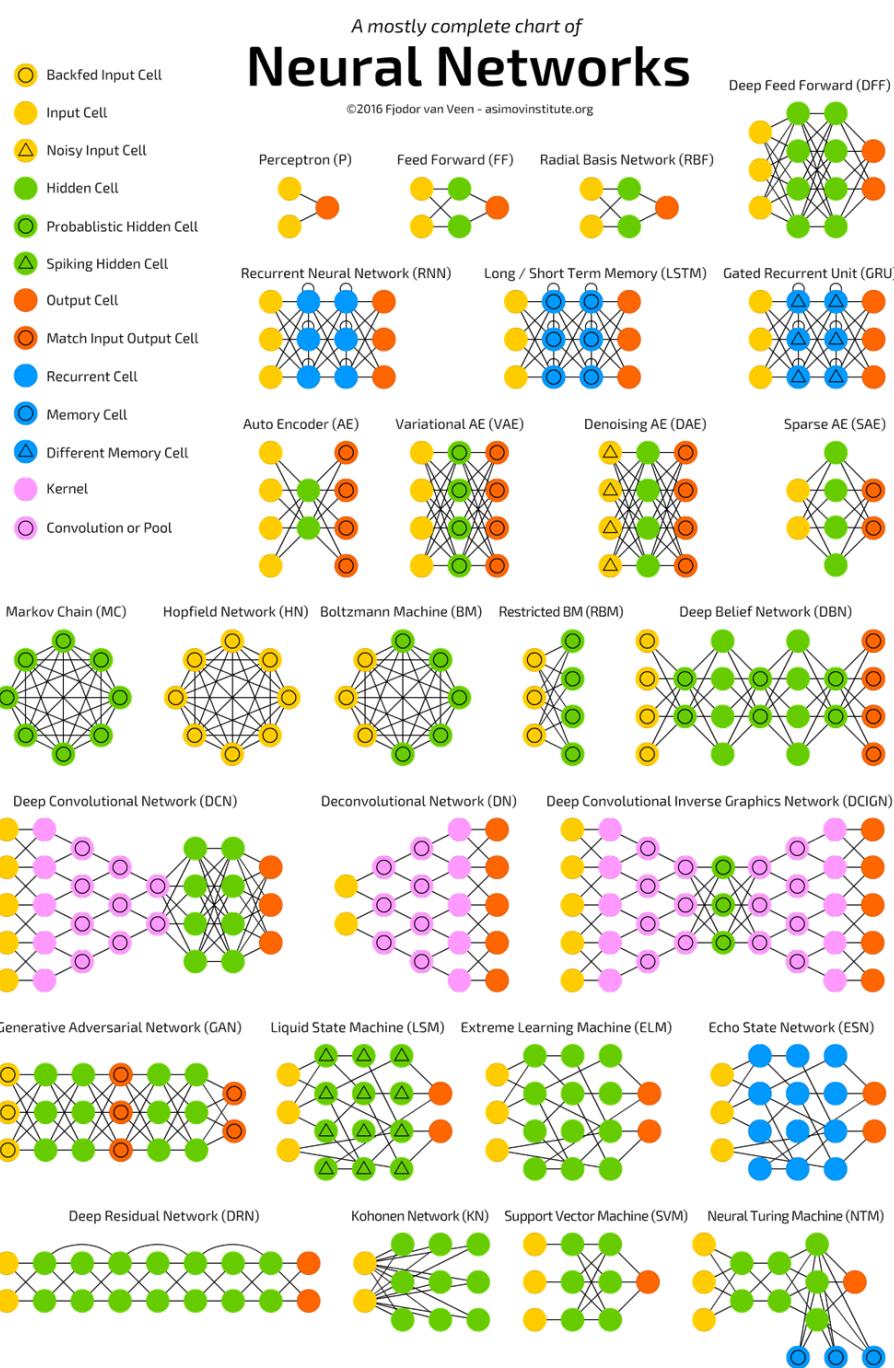
Deep Belief Networks

Convolutional Neural Networks

Recurrent Neural Networks (including LSTMs)

Generative Adversarial Networks (GANs)

Azimov Institute: <http://www.asimovinstitute.org/neural-network-zoo/>



# Autoencoders

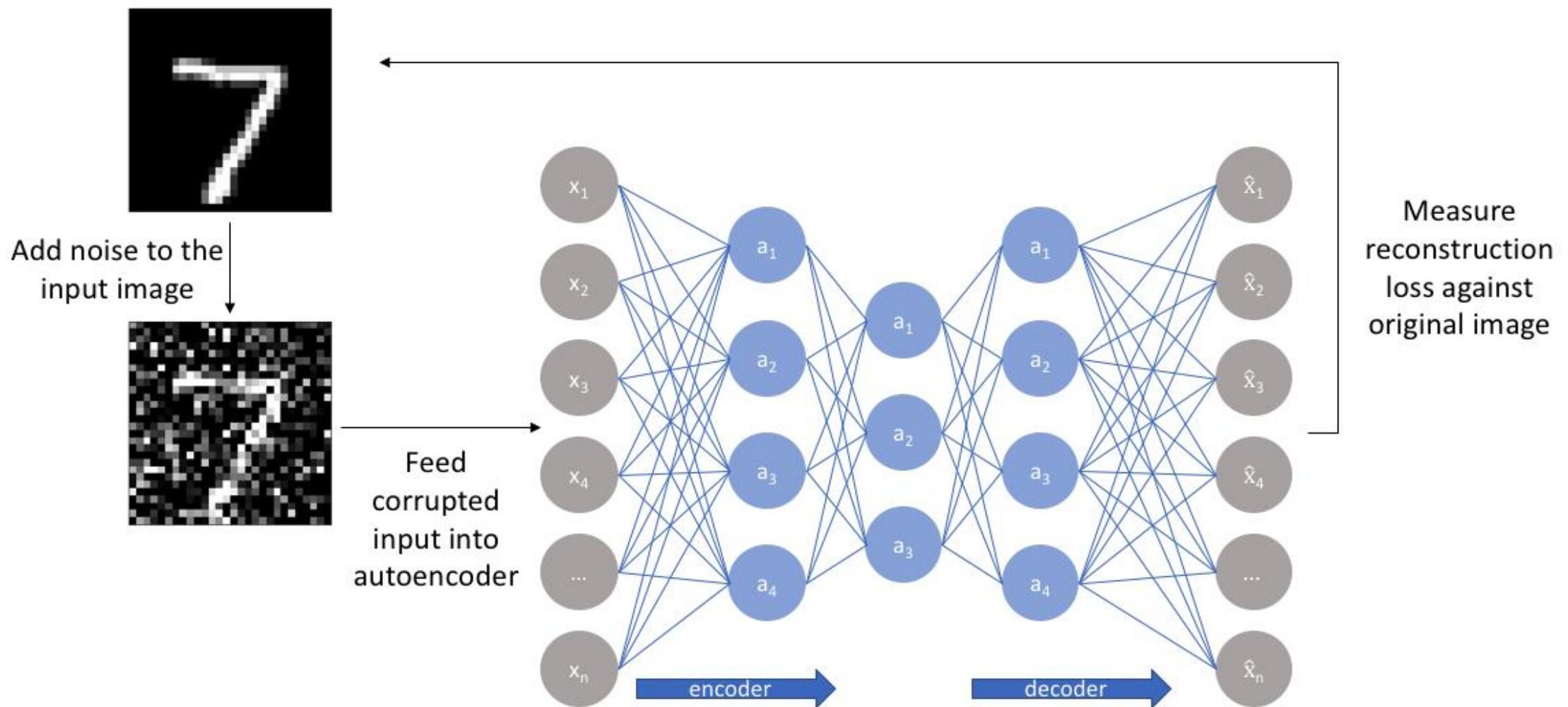
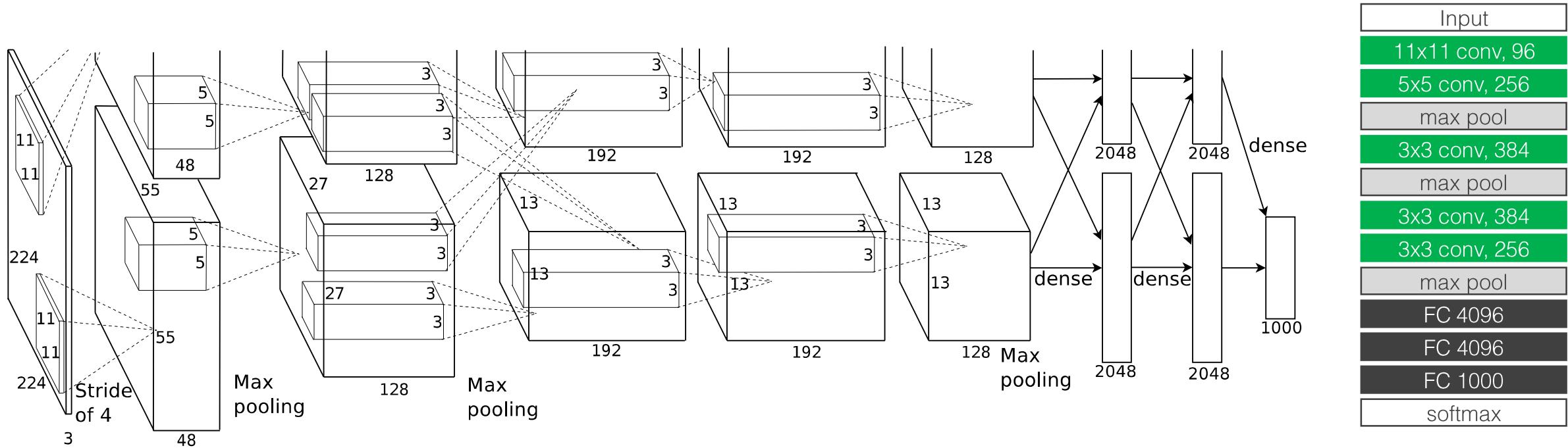


Image from: <https://www.jeremyjordan.me/autoencoders/>

# Convolutional Neural Networks

# AlexNet



Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

Key

Input or output layer
Convolutional Layer
Fully Connected Layer
max pooling layer

# Convolutional Neural Networks

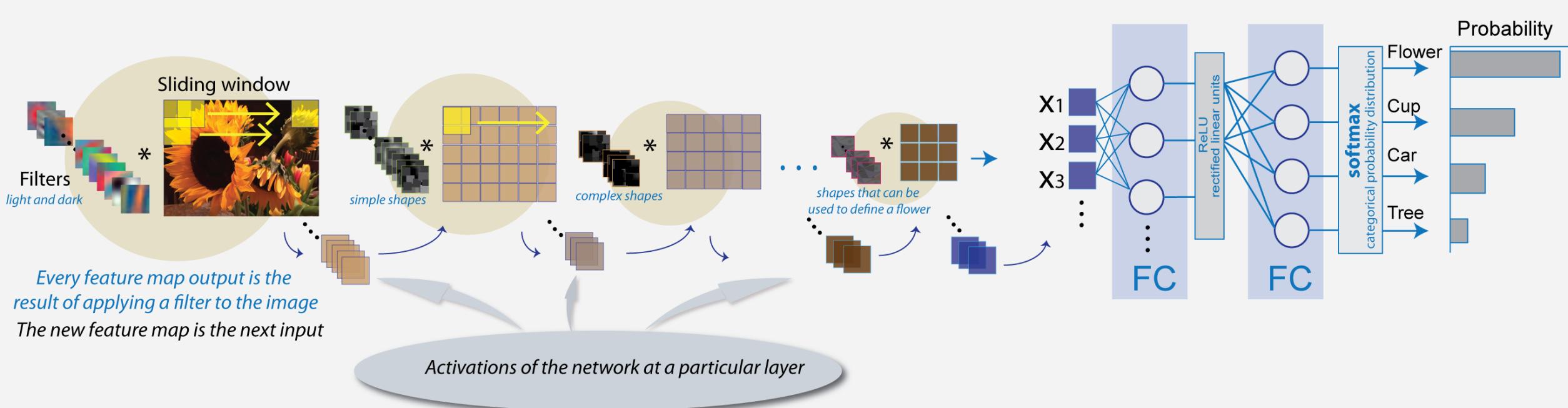
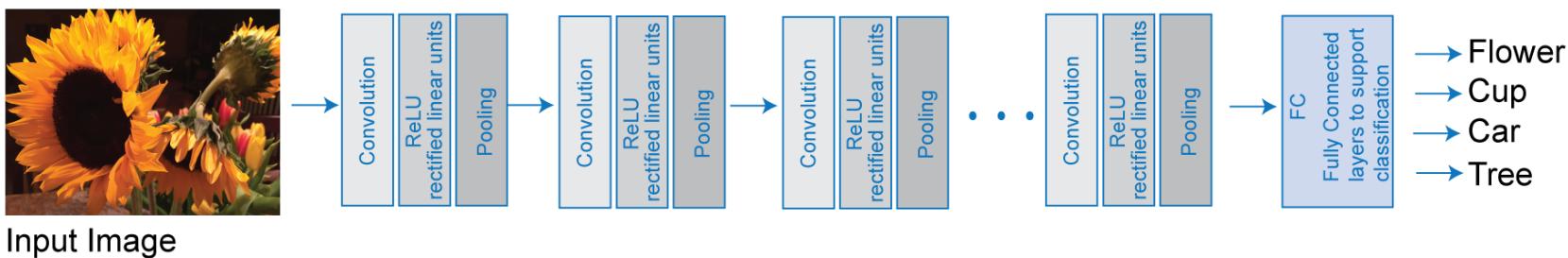


Image from the Mathworks

Data:  $x$

1	2	5	1	4	2
0	2	3	2	0	0
4	5	5	9	8	1
6	3	4	2	3	1
0	1	9	8	7	2
2	3	5	5	5	6

Weights:  $w$

$$\begin{matrix} * & \begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{matrix} \end{matrix}$$

Output:  $x * w$


=

# 2D Convolution

Data:  $x$

1	2	5	1	4	2
0	2	3	2	0	0
4	5	5	9	8	1
6	3	4	2	3	1
0	1	9	8	7	2
2	3	5	5	5	6

Weights:  $w$

1	1	1
0	0	0
-1	-1	-1

Output:  $x * w$


\*

\*

\*

=

Computing one output value:

$$1 \cdot 1 + 1 \cdot 2 + 1 \cdot 5 + \\ 0 \cdot 0 + 0 \cdot 2 + 0 \cdot 3 + \\ (-1) \cdot 4 + (-1) \cdot 5 + (-1) \cdot 5$$

# 2D Convolution

Data:  $x$

1	2	5	1	4	2
0	2	3	2	0	0
4	5	5	9	8	1
6	3	4	2	3	1
0	1	9	8	7	2
2	3	5	5	5	6

Weights:  $w$

\*

1	1	1
0	0	0
-1	-1	-1

Output:  $x * w$

-6			

Computing one output value:

$$1 \cdot 1 + 1 \cdot 2 + 1 \cdot 5 + \\ 0 \cdot 0 + 0 \cdot 2 + 0 \cdot 3 + \\ (-1) \cdot 4 + (-1) \cdot 5 + (-1) \cdot 5 = -6$$

# 2D Convolution

Data:  $X$

1	2	5	1	4	2
0	2	3	2	0	0
4	5	5	9	8	1
6	3	4	2	3	1
0	1	9	8	7	2
2	3	5	5	5	6

Weights:  $w$

1	1	1
0	0	0
-1	-1	-1

Output:  $X * w$

-6	-11		

Computing one output value:

$$1 \cdot 2 + 1 \cdot 5 + 1 \cdot 1 + \\ 0 \cdot 2 + 0 \cdot 3 + 0 \cdot 2 + \\ (-1) \cdot 5 + (-1) \cdot 5 + (-1) \cdot 9 = -11$$

# 2D Convolution

Data:  $\mathbf{X}$

1	2	5	1	4	2
0	2	3	2	0	0
4	5	5	9	8	1
6	3	4	2	3	1
0	1	9	8	7	2
2	3	5	5	5	6

Weights:  $\mathbf{w}$

1	1	1
0	0	0
-1	-1	-1

Output:  $\mathbf{X} * \mathbf{w}$

-6	-11	-12	

Computing one output value:

$$1 \cdot 5 + 1 \cdot 1 + 1 \cdot 4 + \\ 0 \cdot 3 + 0 \cdot 2 + 0 \cdot 0 +$$

$$(-1) \cdot 5 + (-1) \cdot 9 + (-1) \cdot 8 = -12$$

# 2D Convolution

Data:  $X$

1	2	5	1	4	2
0	2	3	2	0	0
4	5	5	9	8	1
6	3	4	2	3	1
0	1	9	8	7	2
2	3	5	5	5	6

Weights:  $w$

1	1	1
0	0	0
-1	-1	-1

Output:  $X * w$

-6	-11	-12	-11

Computing one output value:

$$1 \cdot 1 + 1 \cdot 4 + 1 \cdot 2 + \\ 0 \cdot 2 + 0 \cdot 0 + 0 \cdot 0 + \\ (-1) \cdot 9 + (-1) \cdot 8 + (-1) \cdot 1 = -11$$

# 2D Convolution

Data:  $\mathbf{X}$

1	2	5	1	4	2
0	2	3	2	0	0
4	5	5	9	8	1
6	3	4	2	3	1
0	1	9	8	7	2
2	3	5	5	5	6

Weights:  $\mathbf{w}$

1	1	1
0	0	0
-1	-1	-1

Output:  $\mathbf{X} * \mathbf{w}$

-6	-11	-12	-11
-7			

Computing one output value:

$$1 \cdot 0 + 1 \cdot 2 + 1 \cdot 3 + \\ 0 \cdot 4 + 0 \cdot 5 + 0 \cdot 5 +$$

$$(-1) \cdot 6 + (-1) \cdot 3 + (-1) \cdot 4 = -7$$

# 2D Convolution

Data:  $X$

1	2	5	1	4	2
0	2	3	2	0	0
4	5	5	9	8	1
6	3	4	2	3	1
0	1	9	8	7	2
2	3	5	5	5	6

$6 \times 6$

Weights:  $w$

1	1	1
0	0	0
-1	-1	-1

$*$

$*$

$3 \times 3$

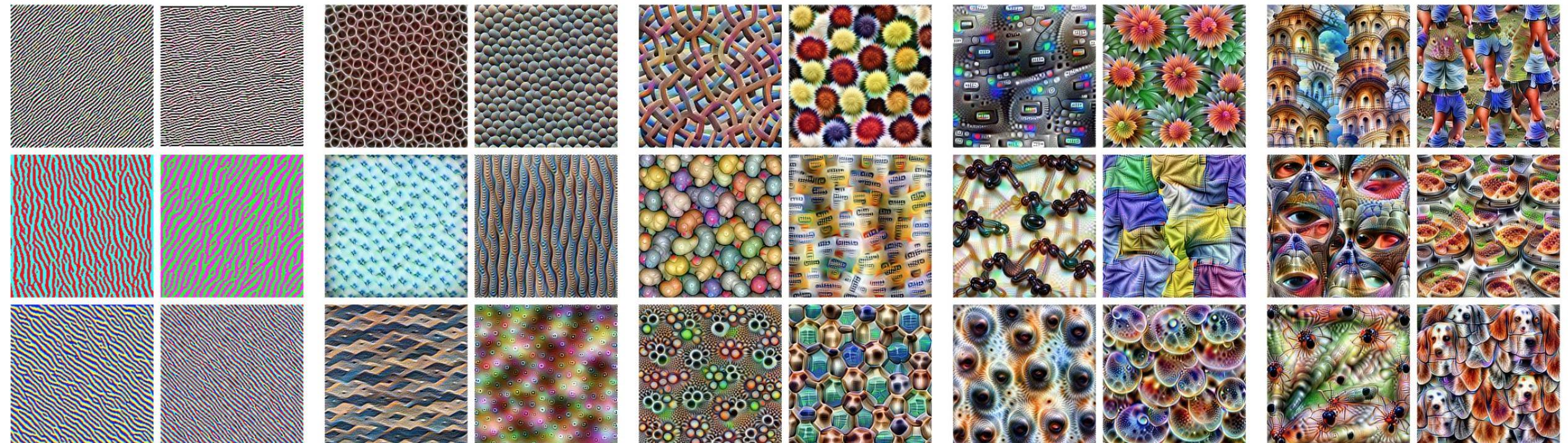
Output:  $X * w$

-6	-11	-12	-11
-7	-2	-2	-4
4	1	-2	1
3	-4	-6	-10

$4 \times 4$

## 2D Convolution

# Features



Edges (layer conv2d0)

Textures (layer mixed3a)

Patterns (layer mixed4a)

Parts (layers mixed4b & mixed4c)

Objects (layers mixed4d & mixed4e)

Olah et al, 2017: <https://distill.pub/2017/feature-visualization/>

# Features

**Dataset Examples** show us what neurons respond to in practice



**Optimization** isolates the causes of behavior from mere correlations. A neuron may not be detecting what you initially thought.



Baseball—or stripes?  
*mixed4a, Unit 6*

Animal faces—or snouts?  
*mixed4a, Unit 240*

Clouds—or fluffiness?  
*mixed4a, Unit 453*

Buildings—or sky?  
*mixed4a, Unit 492*

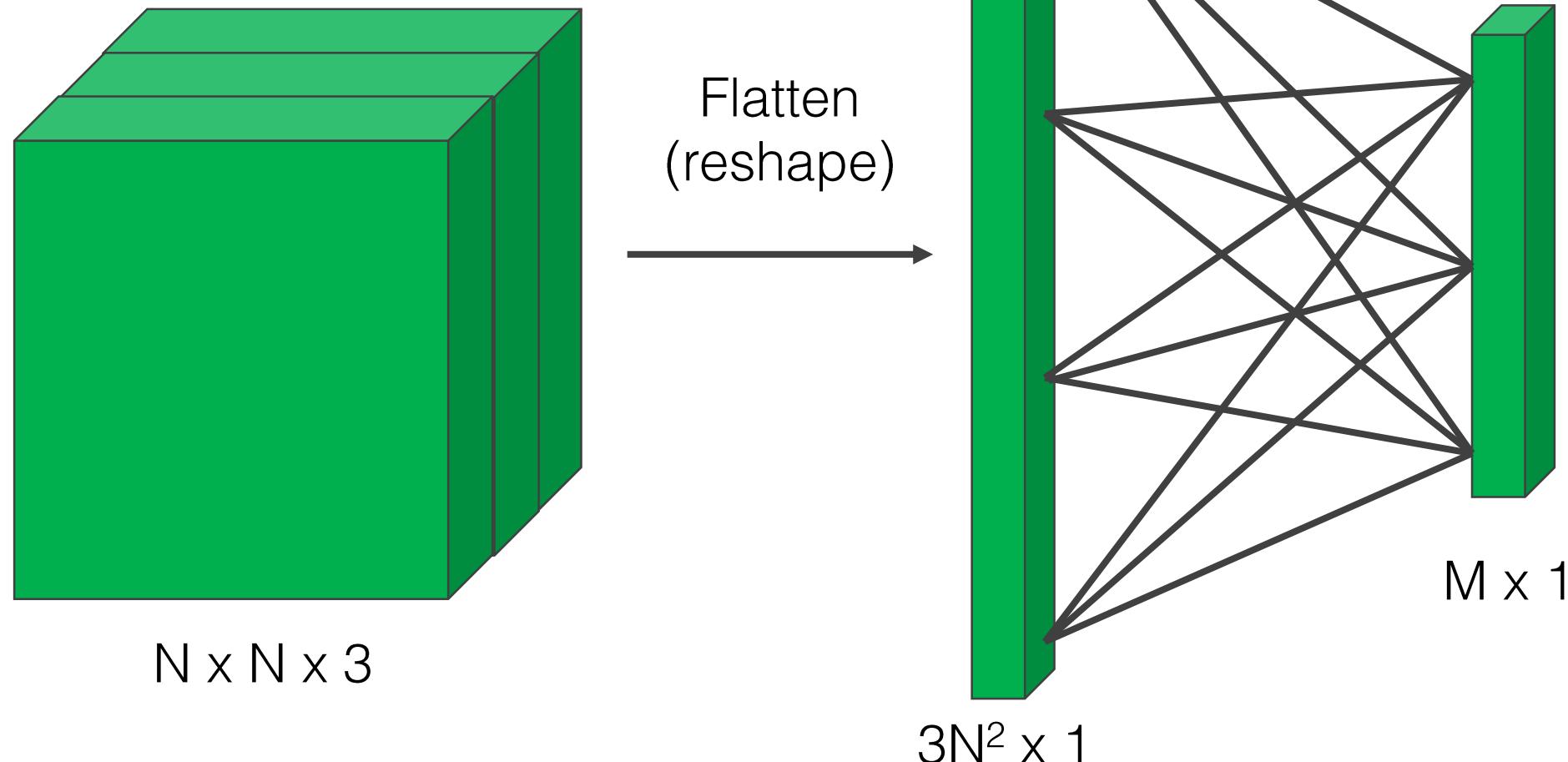
# Resources on Visualization of Features

Feature visualization: <https://distill.pub/2017/feature-visualization/>

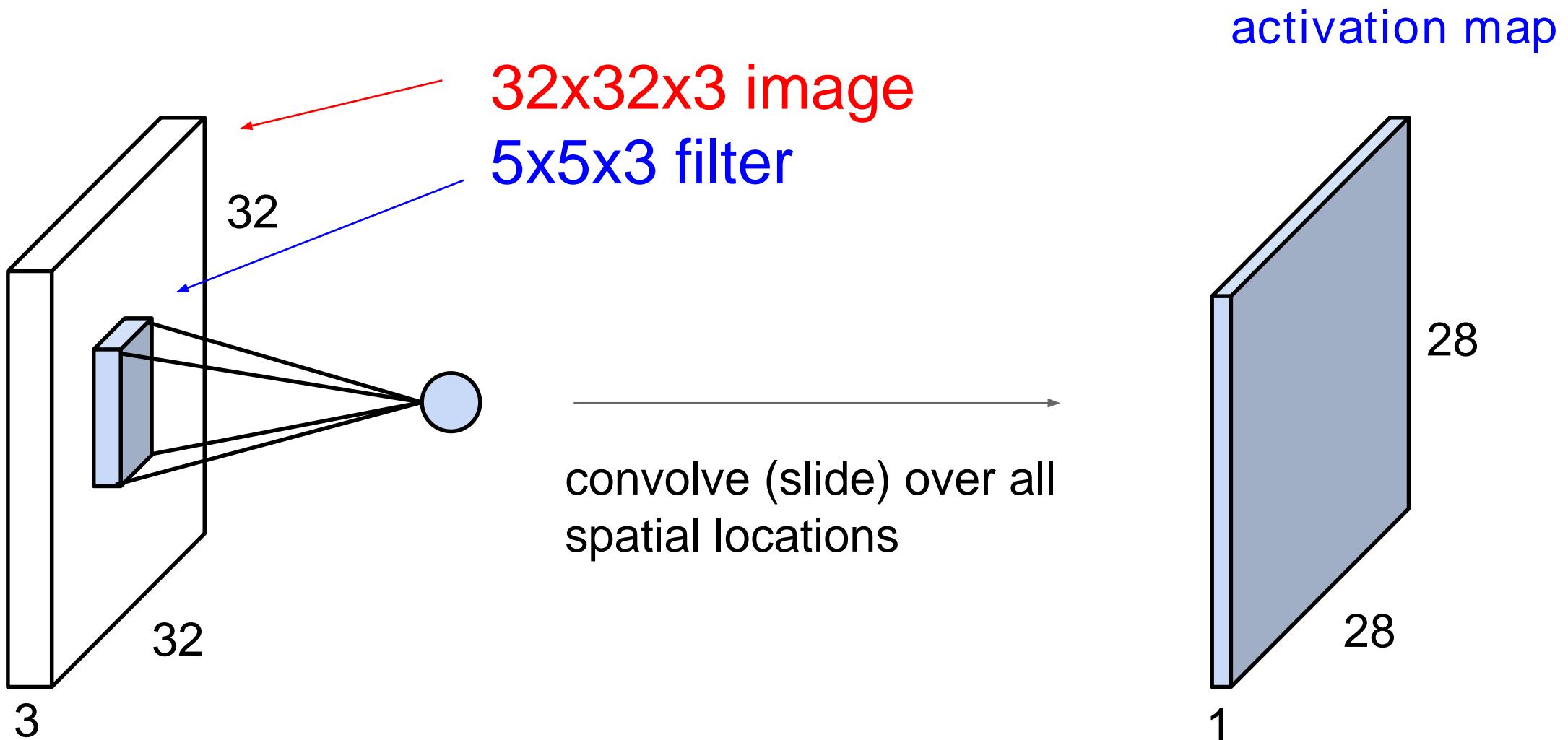
Building blocks of interpretability: <https://distill.pub/2018/building-blocks/>

Activation Activation Atlases: <https://distill.pub/2019/activation-atlas/>

# Fully Connected Layer

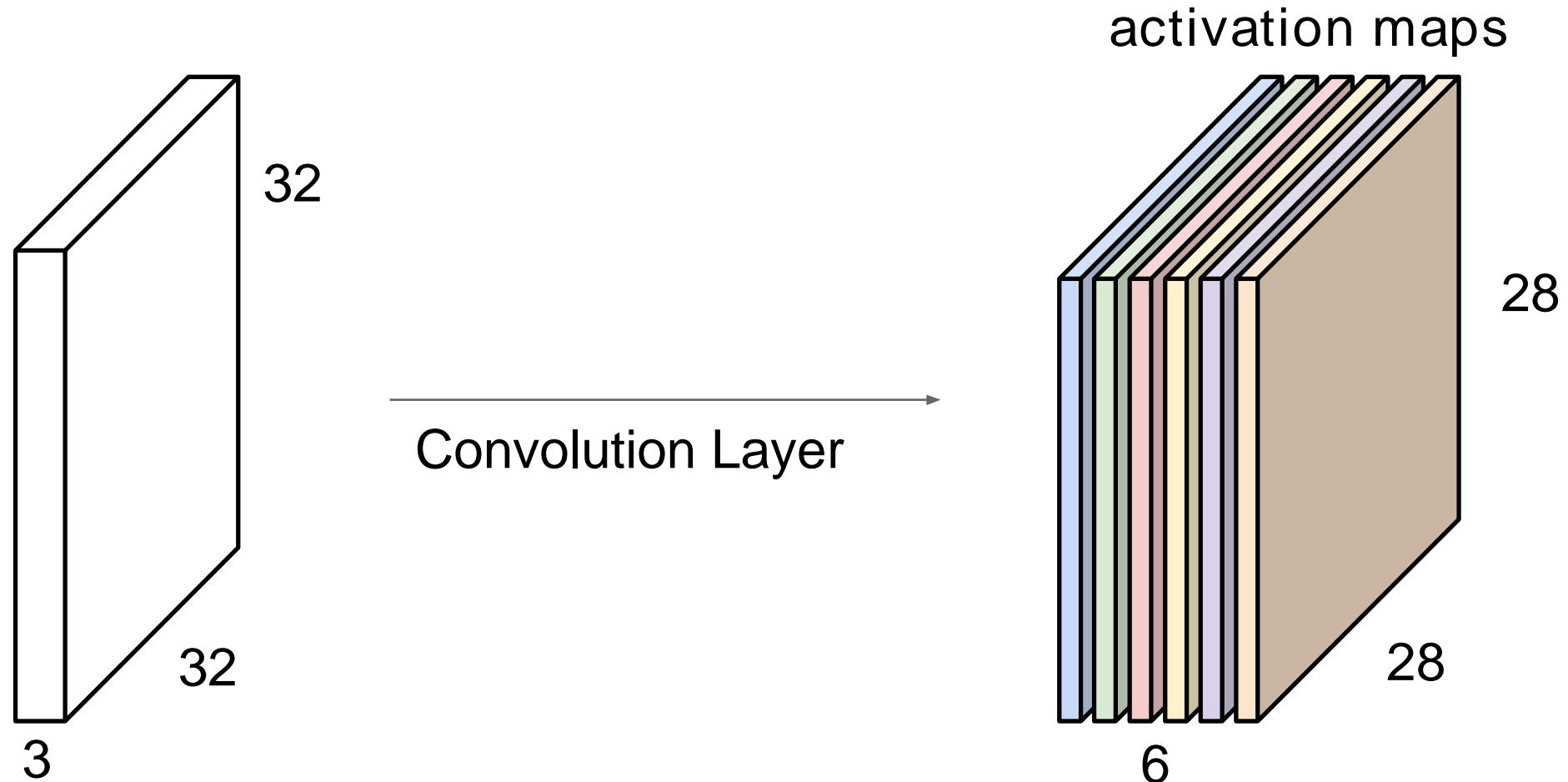


# Convolution Layer



From Fei-Fei Li, Justin Johnson, and Serena Young. CS231n, 2018

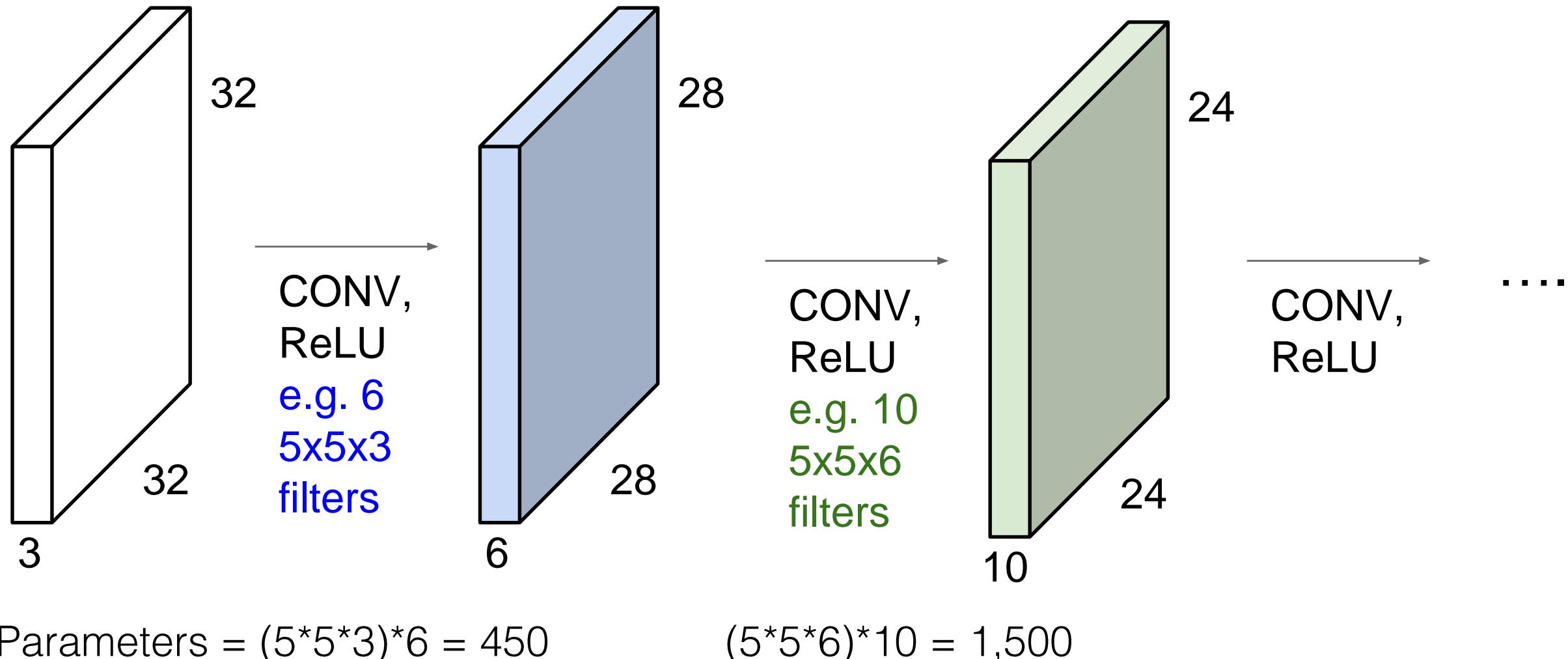
For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



We stack these up to get a “new image” of size  $28 \times 28 \times 6$ !

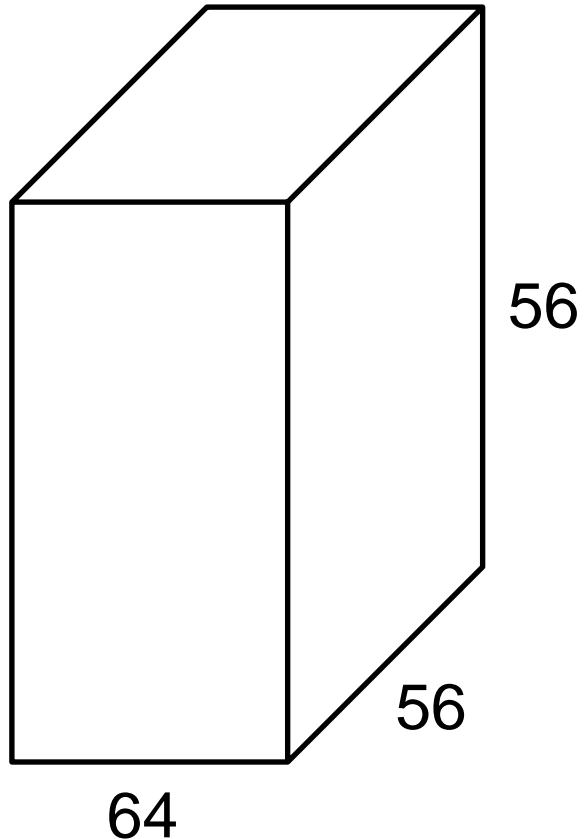
From Fei-Fei Li, Justin Johnson, and Serena Young. CS231n, 2018

Preview: ConvNet is a sequence of Convolution Layers, interspersed with activation functions



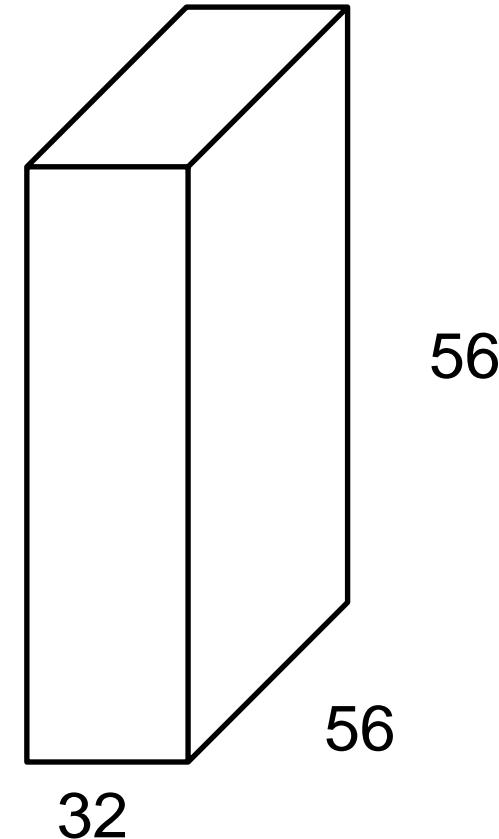
From Fei-Fei Li, Justin Johnson, and Serena Young. CS231n, 2018

# $1 \times 1$ Convolution Explained



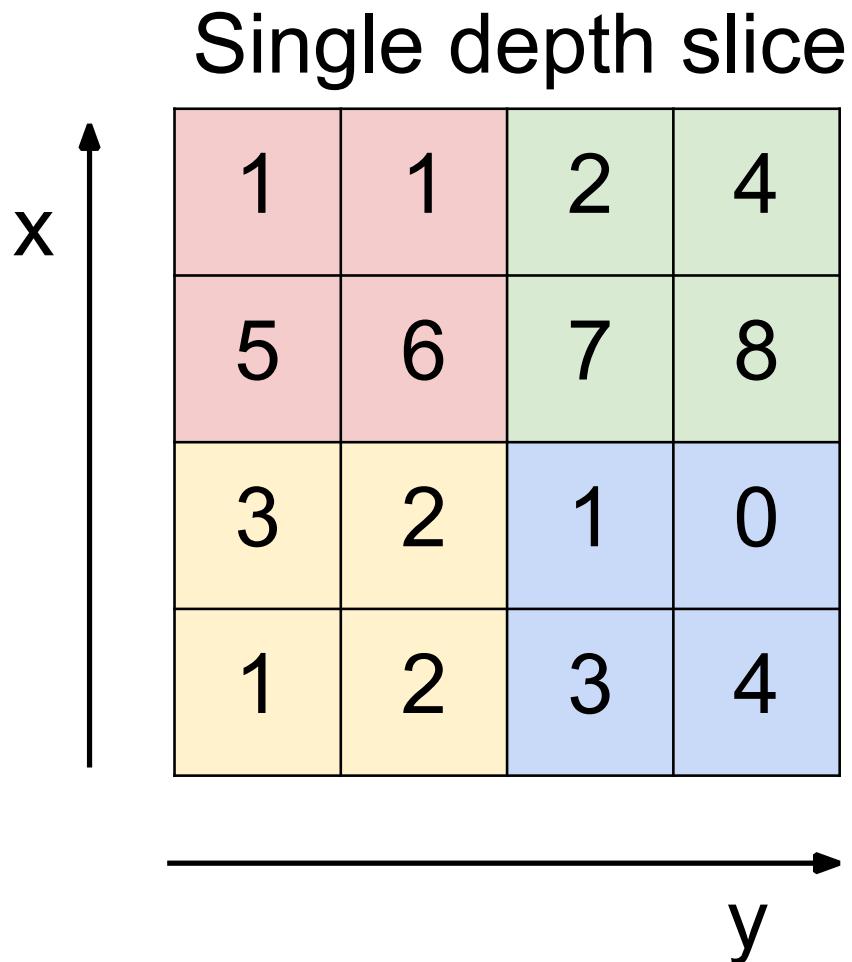
$1 \times 1$  CONV  
with 32 filters

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



From Fei-Fei Li, Justin Johnson, and Serena Young. CS231n, 2018

# Max Pooling

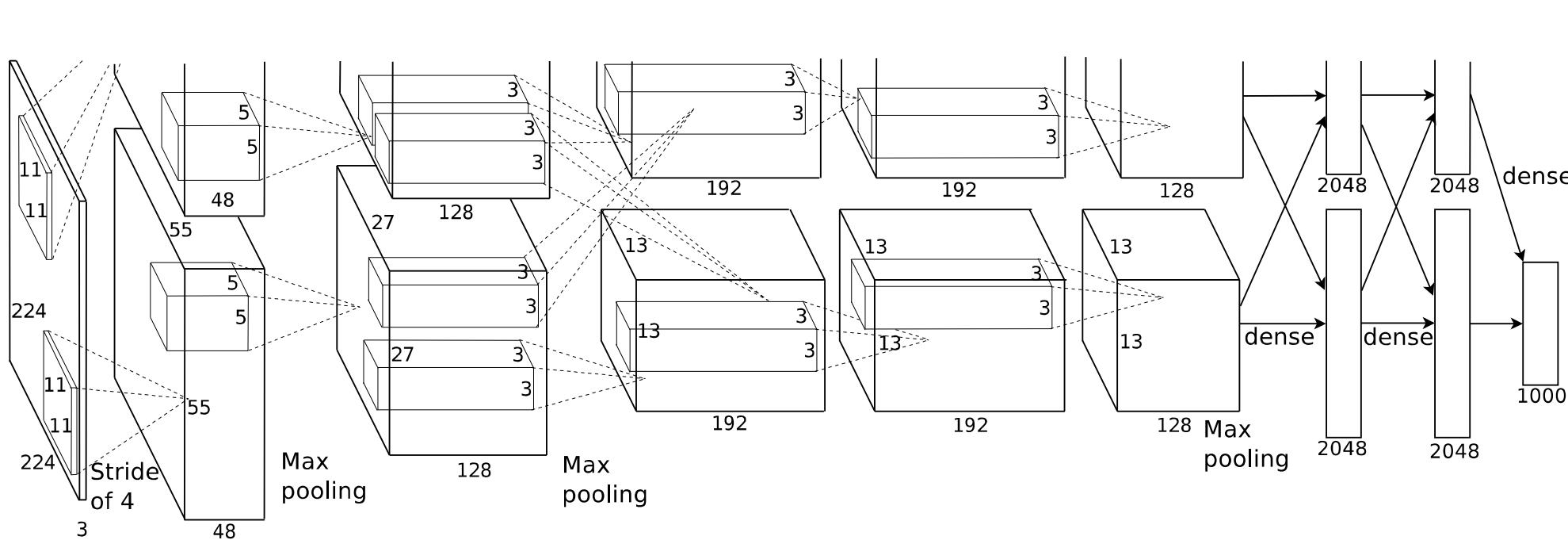


max pool with 2x2 filters  
and stride 2

6	8
3	4

From Fei-Fei Li, Justin Johnson, and Serena Young. CS231n, 2018

# AlexNet



Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

Key

Input or output layer
Convolutional Layer
Fully Connected Layer
max pooling layer

AlexNet  
(2012)



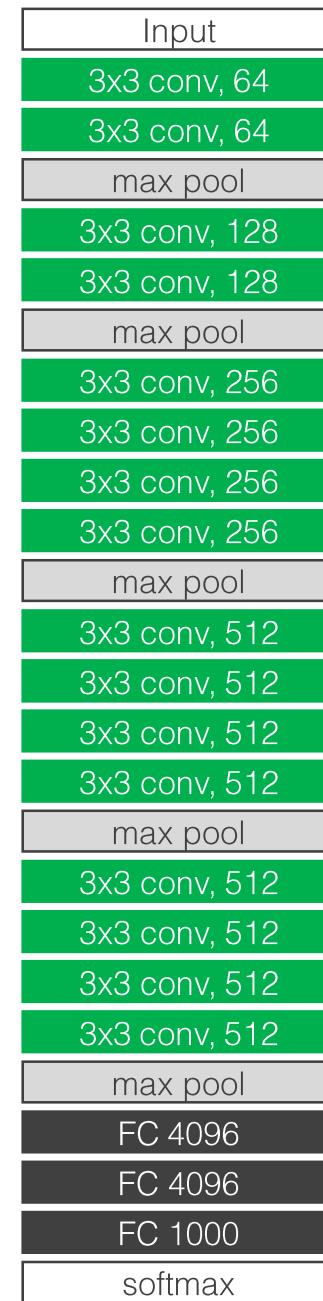
Note: an activation function is applied to the output of each layer

VGG16  
(2014)



Fewer layers,  
larger filters

VGG19  
(2014)



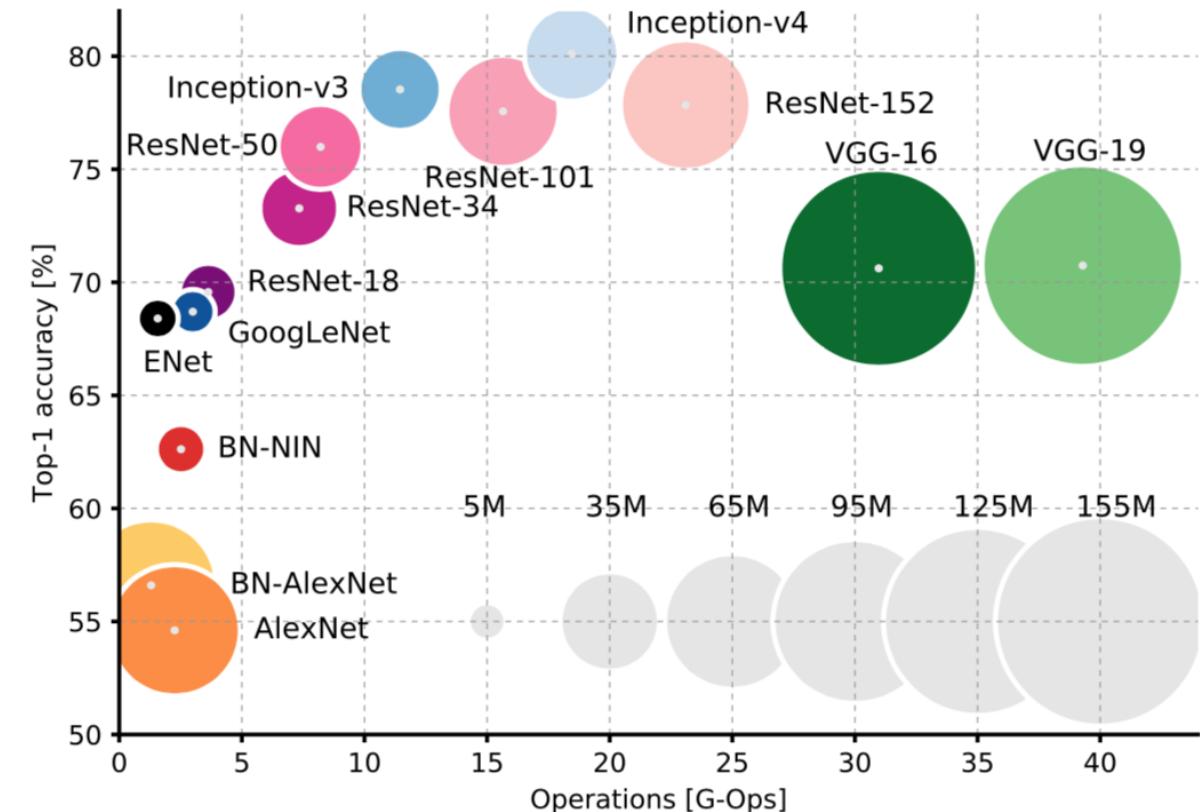
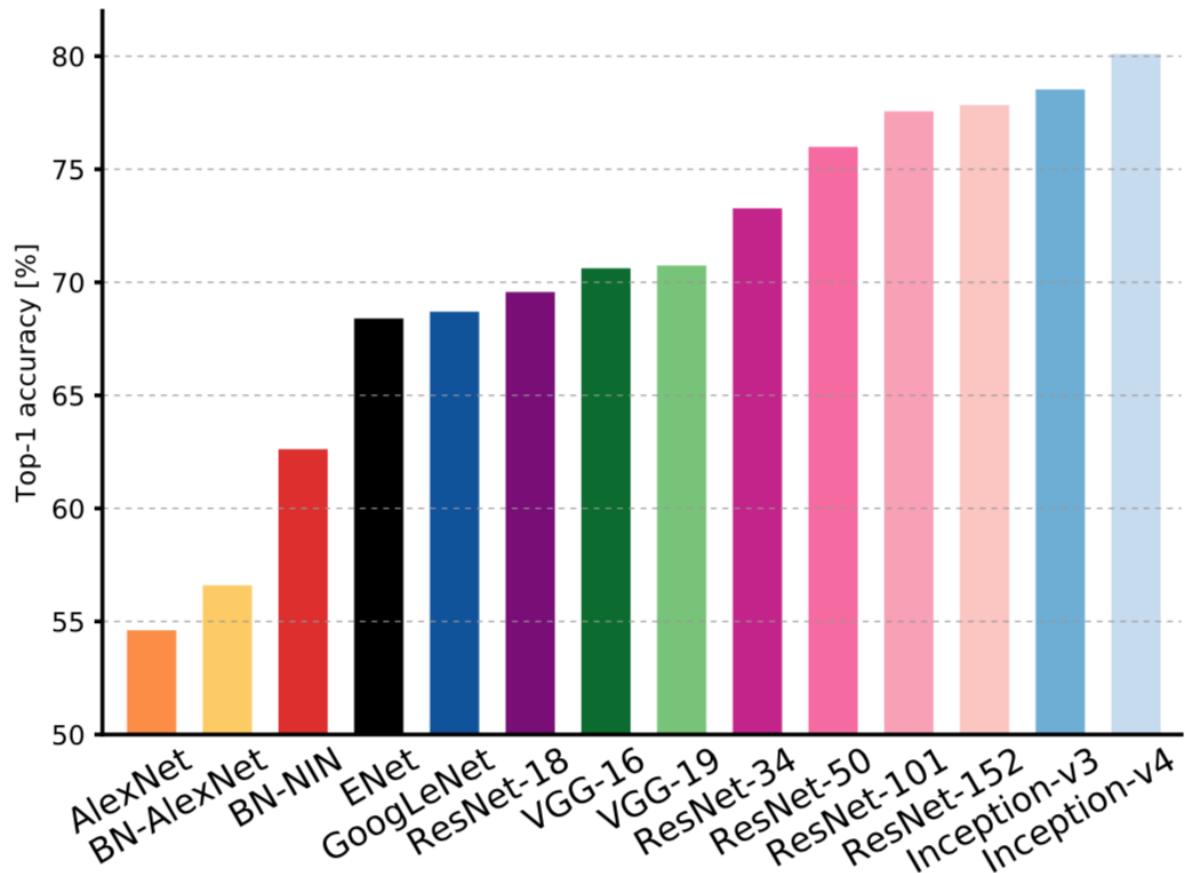
## Key

Input or output layer
Convolutional Layer
Fully Connected Layer
max pooling layer

# CNN Architectures

Adapted from Fei-Fei Li, Justin Johnson, and Serena Young. CS231n, 2018

# Deep Learning Models Compared



Models compared for ImageNet

A. Canziani, E. Culurciello and A. Paszke, "Evaluation of neural network architectures for embedded systems," 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, 2017, pp. 1-4.

# Generative Adversarial Networks

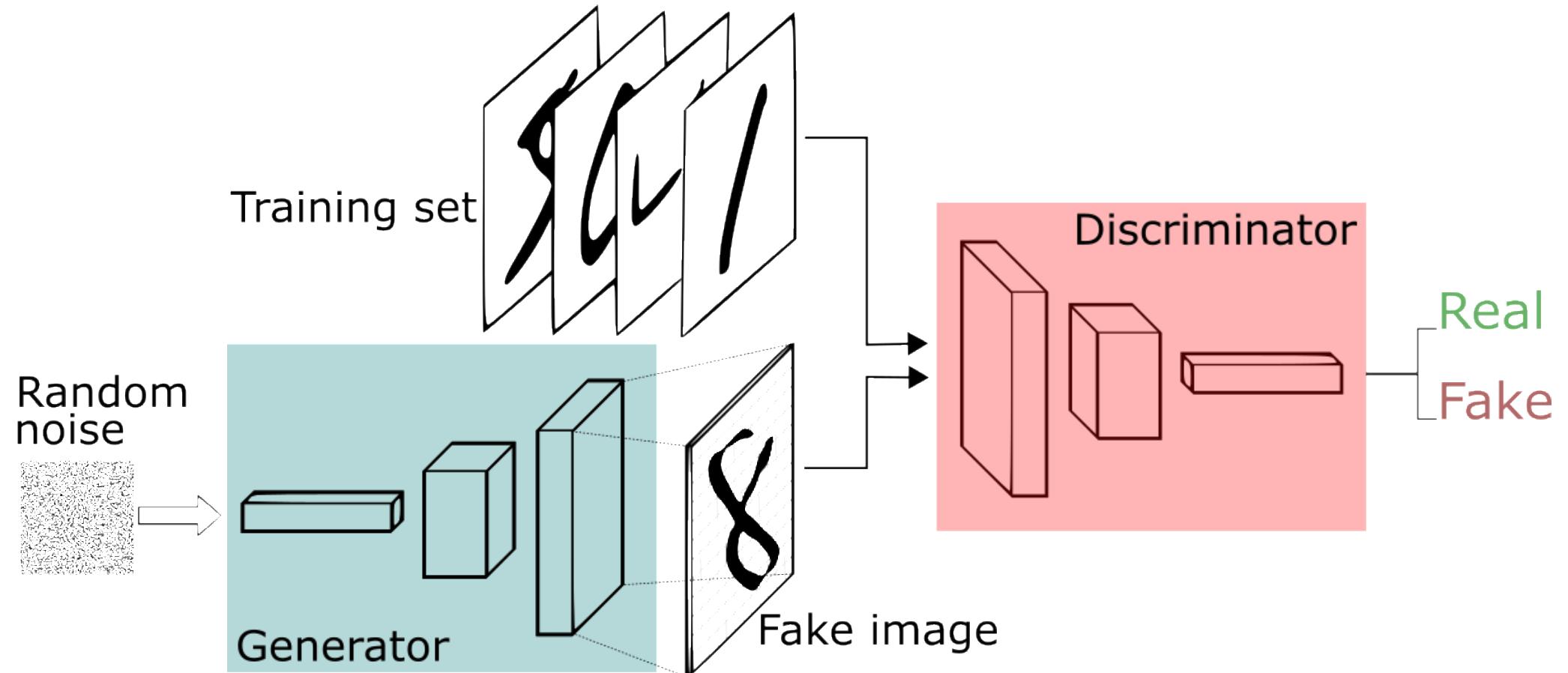


Image from: <https://skymind.ai/wiki/generative-adversarial-network-gan>