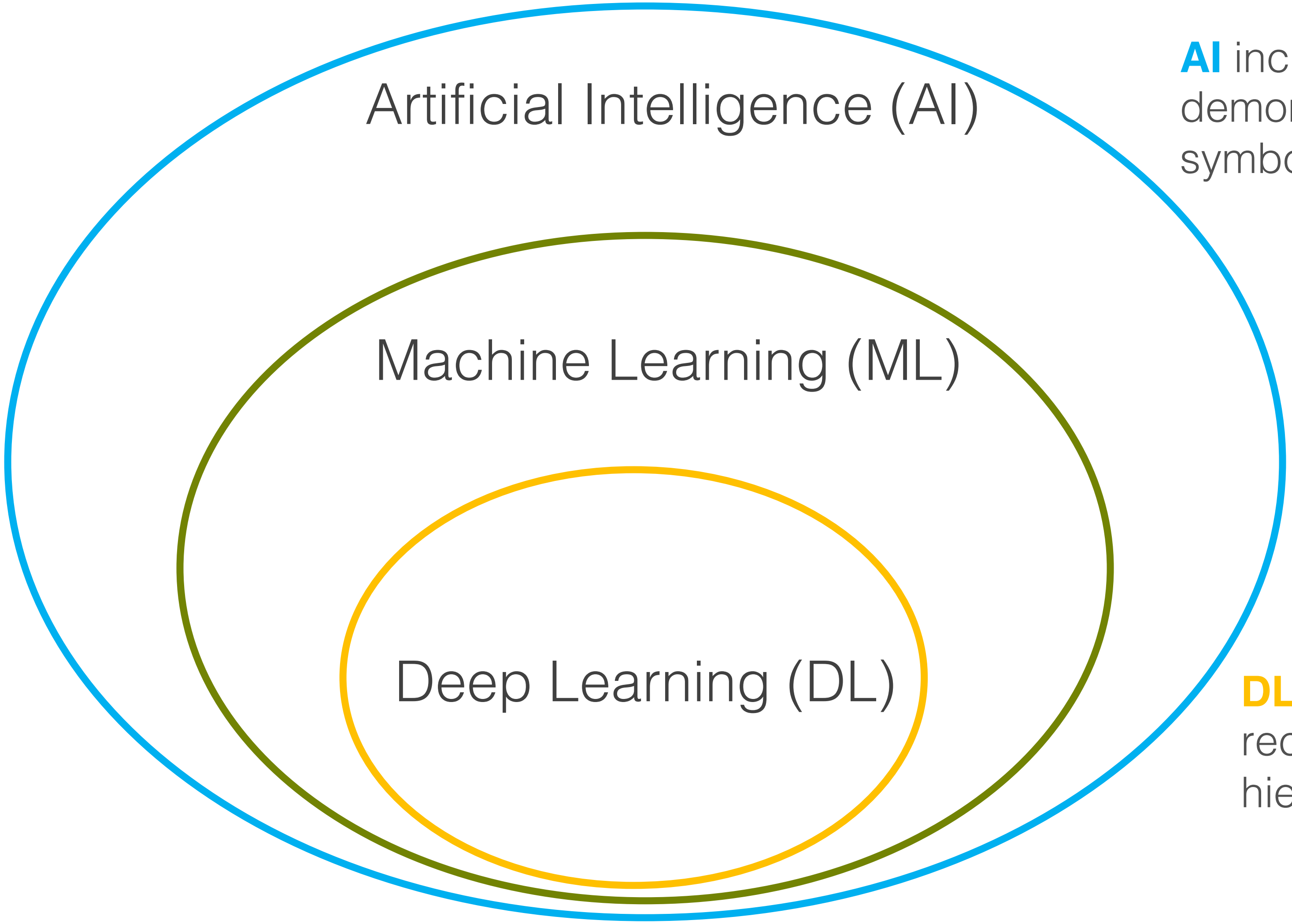
Deep Learning

Lecture 18

Hierarchy of Learning



AI includes many types of intelligence demonstrated by machines: cybernetics, symbolic, statistical learning

ML can…

Uncover structure in data (unsupervised)

Make predictions (supervised)

Learn by doing (reinforcement)

DL is a type of **ML** that makes use of recent advances in computation to learn hierarchical representation of data

Alternate representations of data

Raw data | Coordinate change | Improved representation

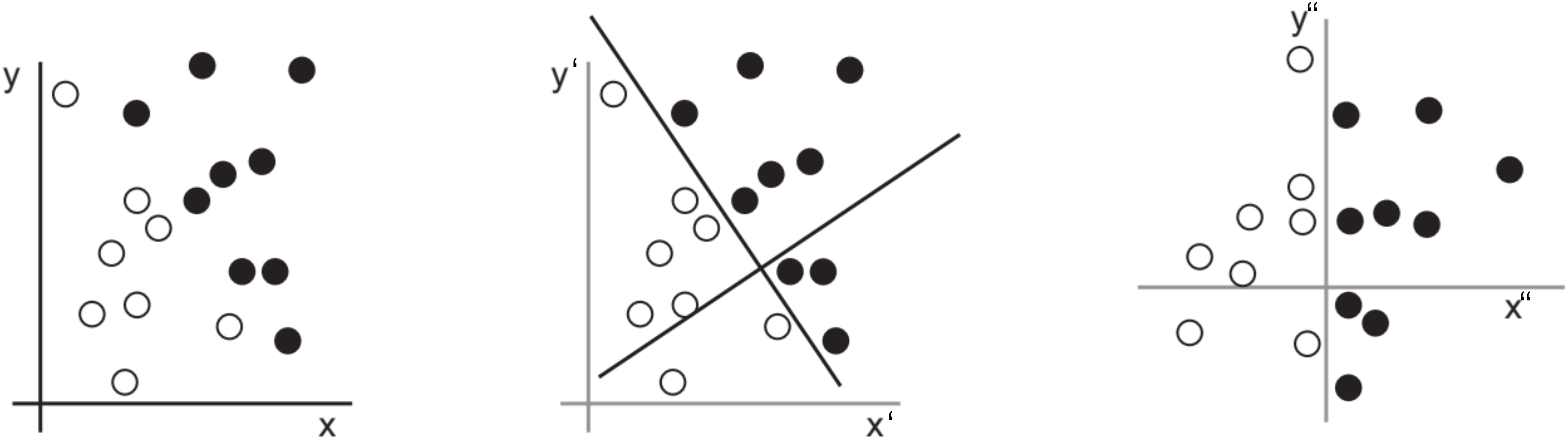
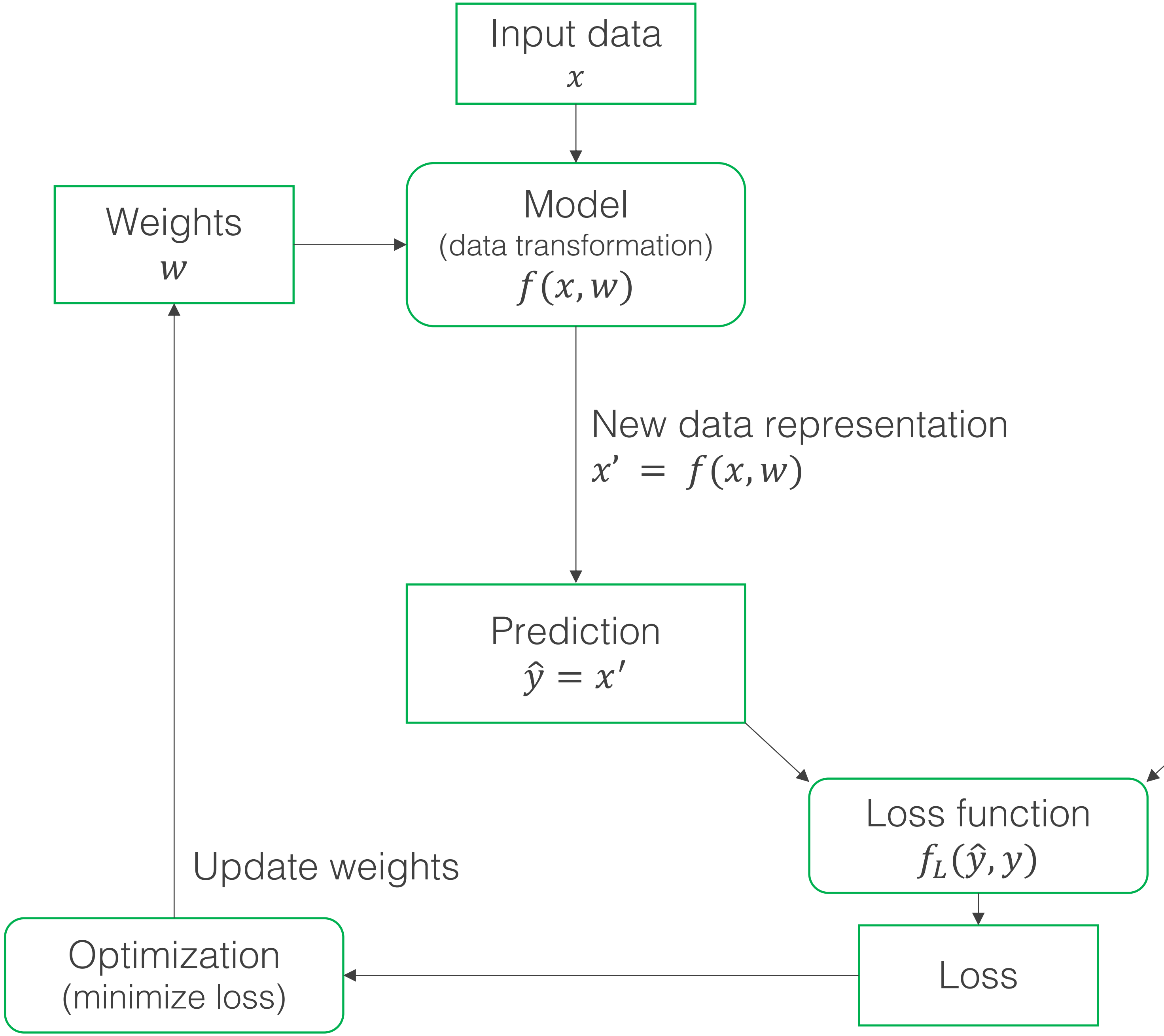
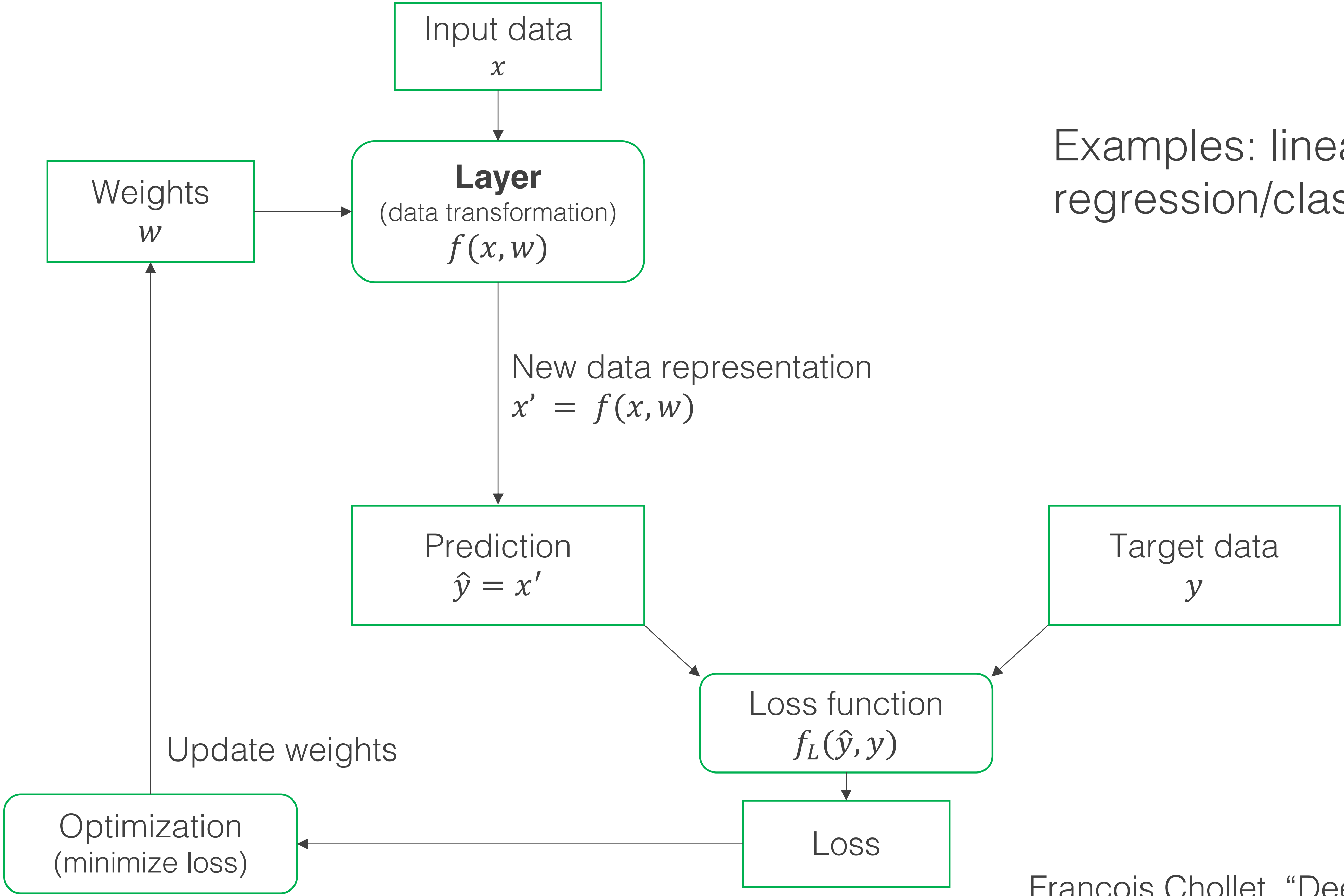


Figure from Francois Chollet, “Deep Learning with Python.”



Examples: linear/logistic regression/classification

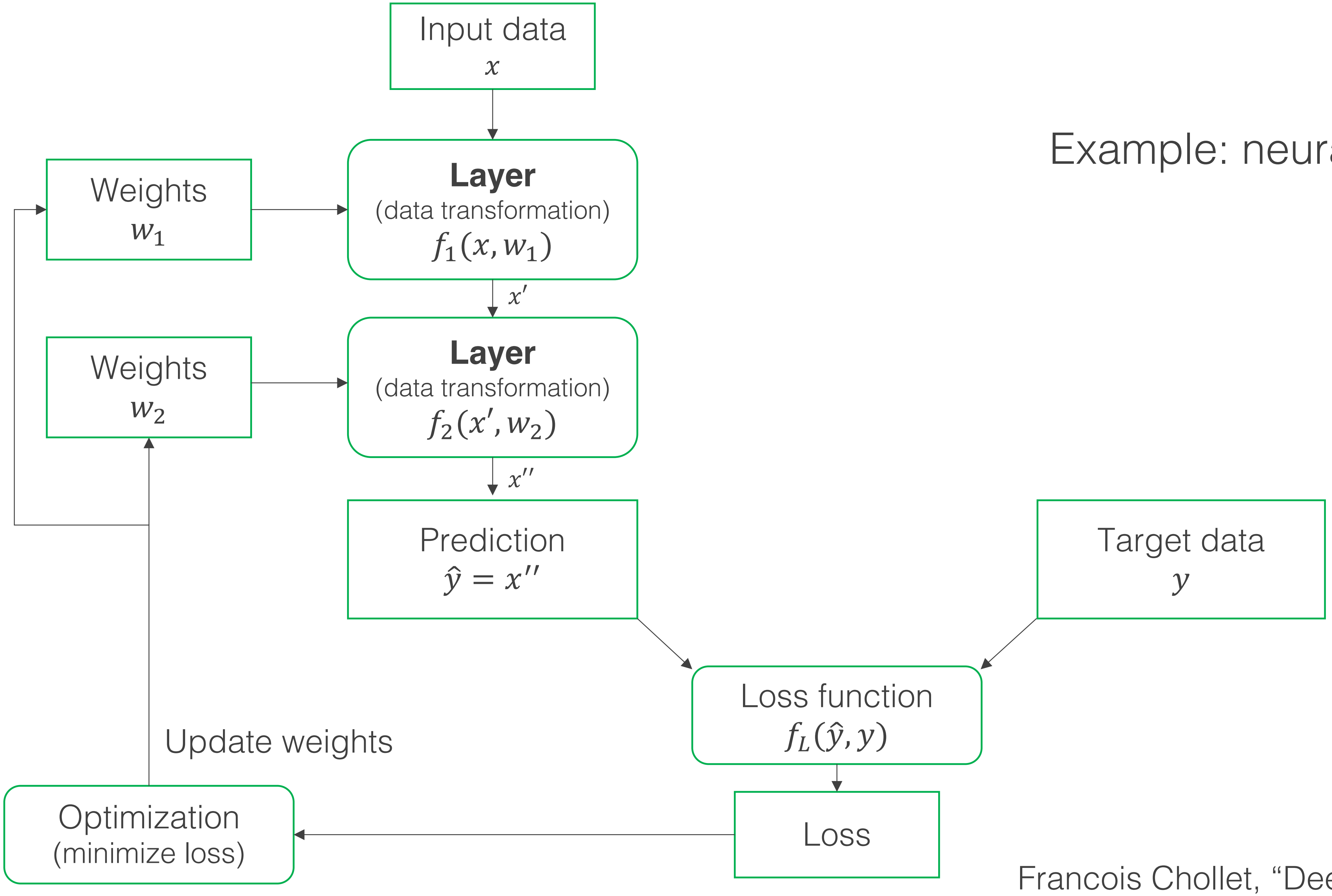
Francois Chollet, “Deep Learning with Python.”



Examples: linear/logistic regression/classification

Francois Chollet, “Deep Learning with Python.”

Example: neural network



Francois Chollet, "Deep Learning with Python."

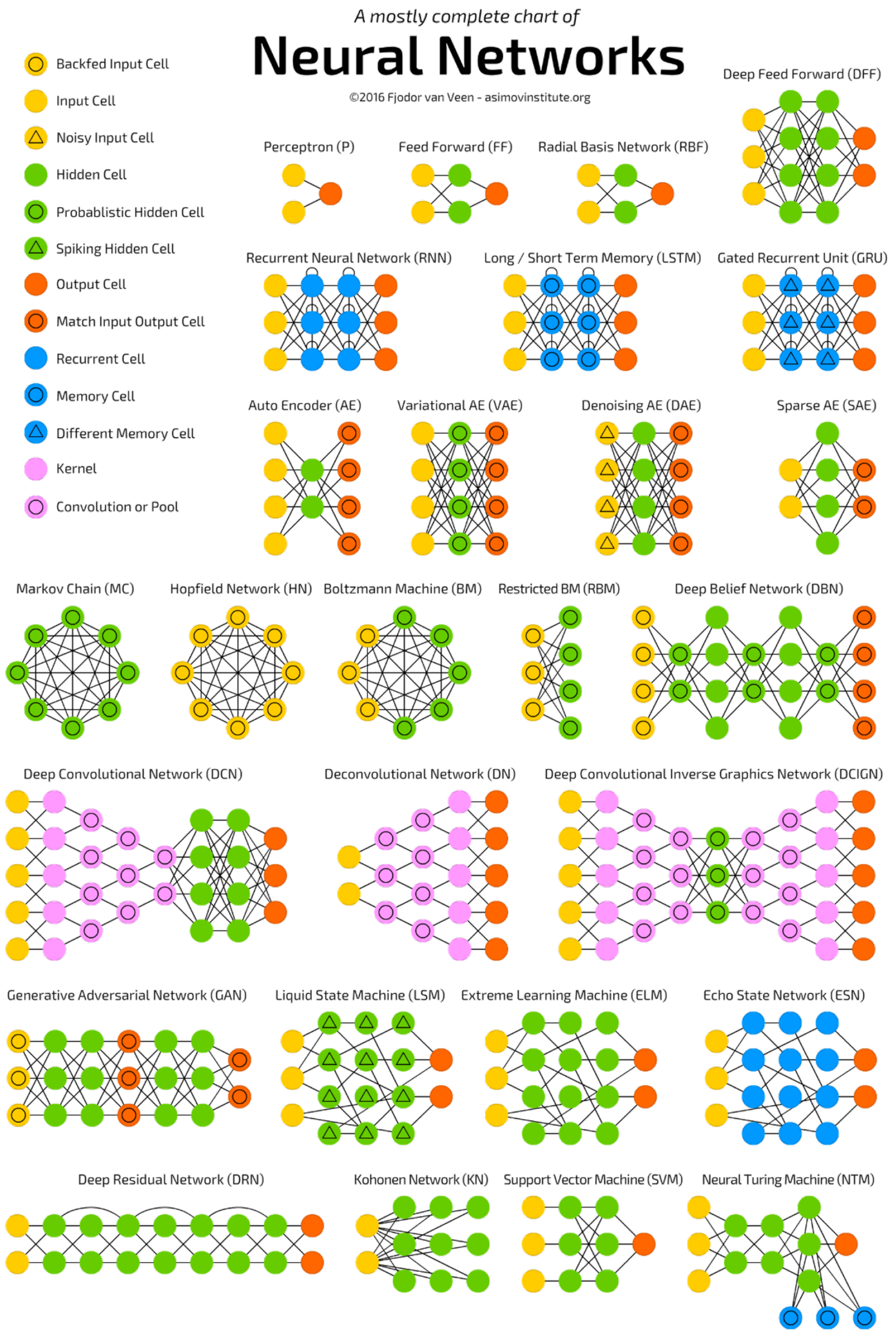
Types of Deep Learning Tools

Autoencoders

Convolutional Neural Networks

Recurrent Neural Networks (including LSTMs)

Generative Adversarial Networks (GANs)



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

Autoencoders

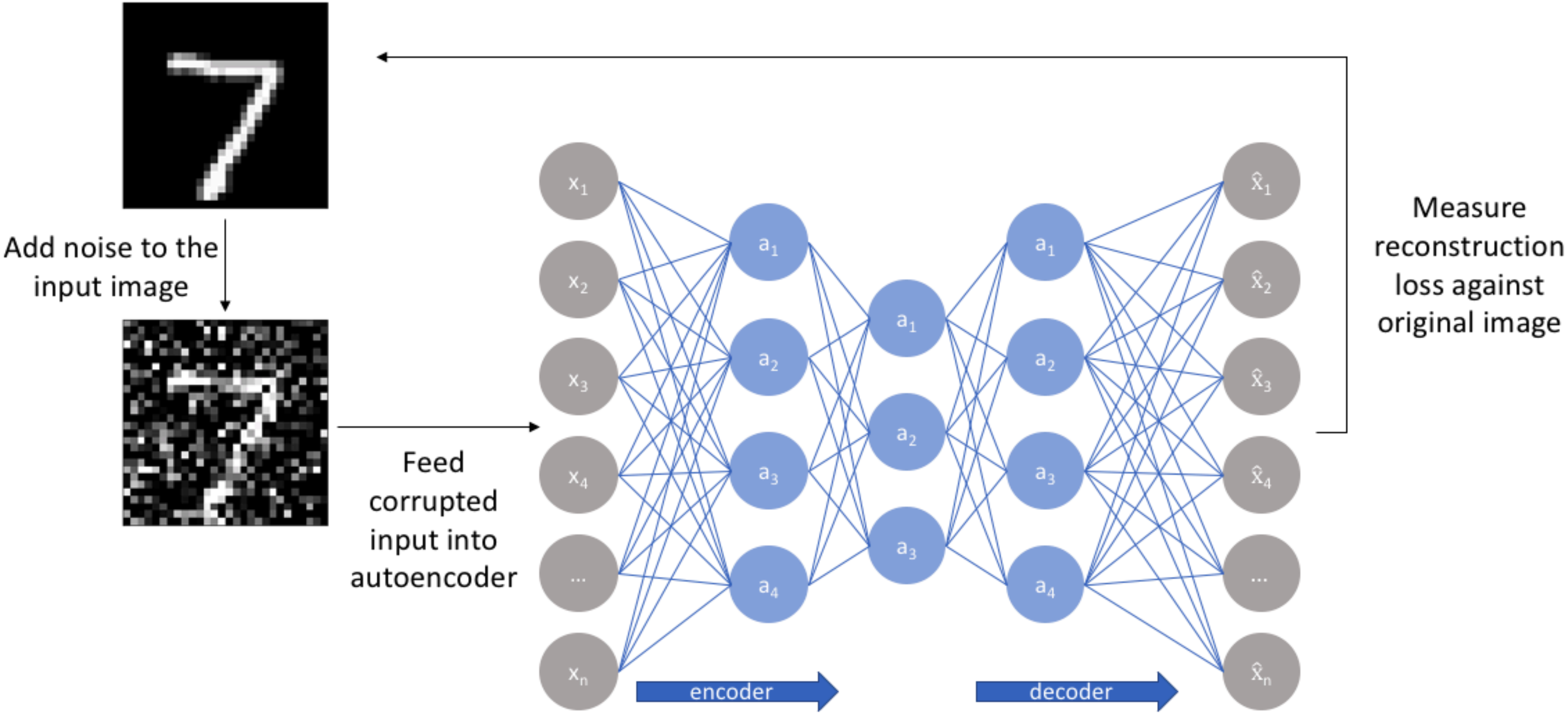
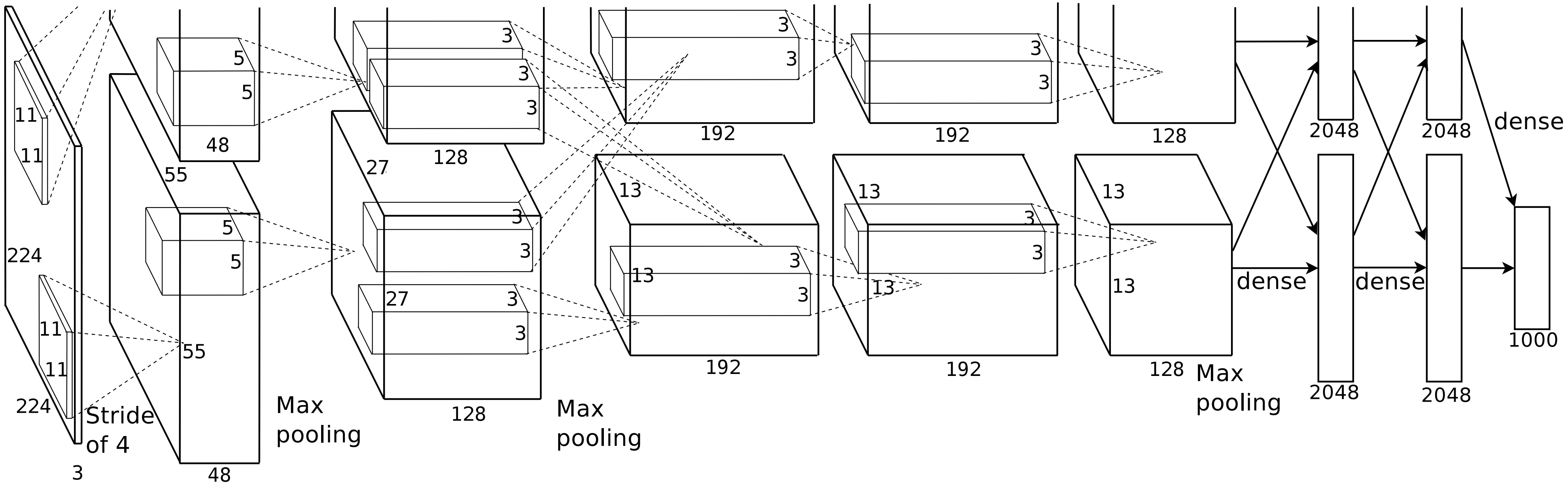


Image from: https://www.jeremyjordan.me/autoencoders/

Convolutional Neural Networks

AlexNet



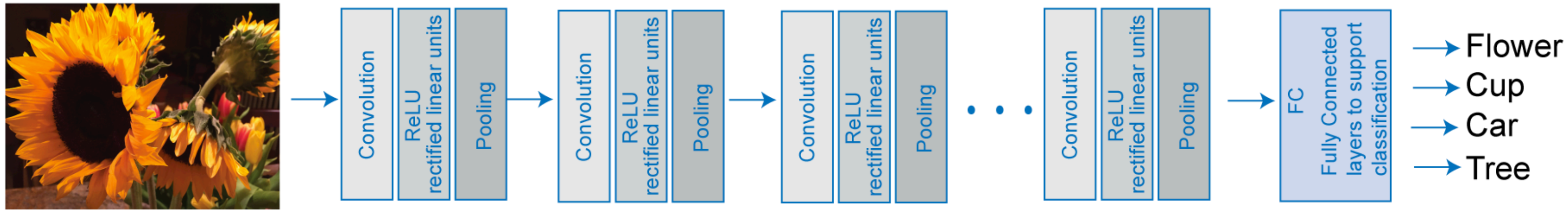
Input
11x11 conv, 96
5x5 conv, 256
max pool
3x3 conv, 384
max pool
3x3 conv, 384
3x3 conv, 256
max pool
FC 4096
FC 4096
FC 1000
softmax

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



Input Image

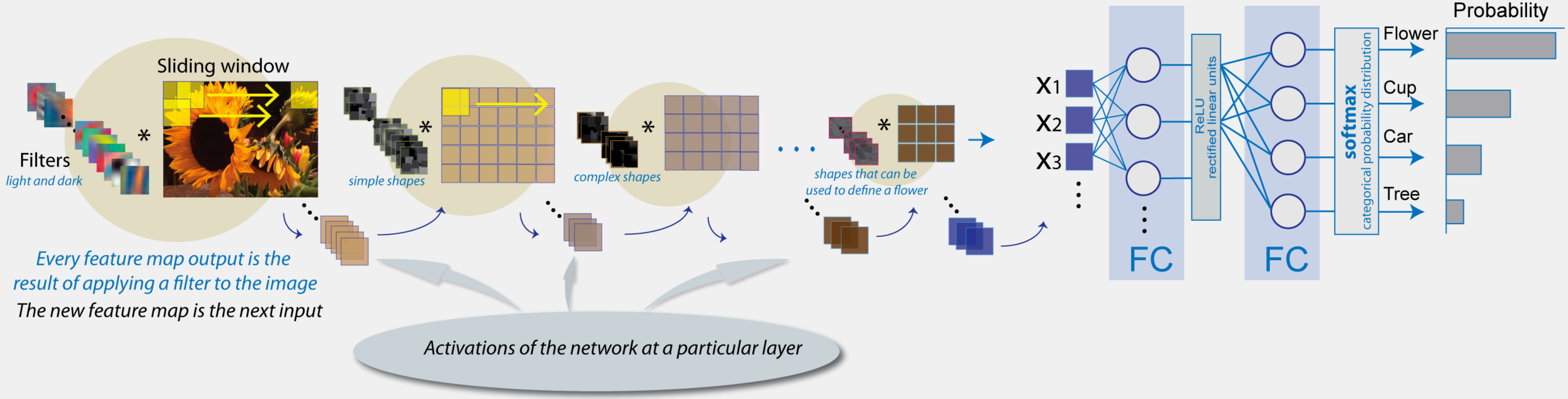


Image from the Mathworks

Data: $\boldsymbol{x}$

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: $\boldsymbol{w}$

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

$=$

Output: $\boldsymbol{x} * \boldsymbol{w}$

2D Convolution

Data: $\boldsymbol{x}$

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: $\boldsymbol{w}$

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

$=$

Output: $\boldsymbol{x} * \boldsymbol{w}$

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: $\boldsymbol{x}$

$$\begin{bmatrix} \boxed{\begin{matrix} 1 & 2 & 5 \\ 0 & 2 & 3 \\ 4 & 5 & 5 \end{matrix}} & \begin{matrix} 1 & 4 & 2 \\ 2 & 0 & 0 \\ 9 & 8 & 1 \end{matrix} \\ \begin{matrix} 6 & 3 & 4 \\ 0 & 1 & 9 \\ 2 & 3 & 5 \end{matrix} & \begin{matrix} 2 & 3 & 1 \\ 8 & 7 & 2 \\ 5 & 5 & 6 \end{matrix} \end{bmatrix}$$

$*$

Weights: $\boldsymbol{w}$

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$=$

Output: $\boldsymbol{x} * \boldsymbol{w}$

$$\begin{bmatrix} \mathbf{-6} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}$$

Computing one output value:

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

2D Convolution

2D Convolution

Data: $\boldsymbol{X}$

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: $\boldsymbol{w}$

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

=

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

-6	**-11**		

Computing one output value:

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

Data: $\boldsymbol{X}$

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: $\boldsymbol{w}$

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

$=$

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

-6	**-11**	**-12**	

Computing one output value:

$$1\cdot 5 + 1\cdot 1 + 1\cdot 4 +$$
$$0\cdot 3 + 0\cdot 2 + 0\cdot 0 +$$
$$(-1)\cdot 5 + (-1)\cdot 9 + (-1)\cdot 8 = \mathbf{-12}$$

2D Convolution

Data: $\boldsymbol{X}$

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: $\boldsymbol{w}$

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

=

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

-6	**-11**	**-12**	**-11**

Computing one output value:

$$1\cdot1 + 1\cdot4 + 1\cdot2 + 0\cdot2 + 0\cdot0 + 0\cdot0 + (-1)\cdot9 + (-1)\cdot8 + (-1)\cdot1 = \mathbf{-11}$$

2D Convolution

Data: $\boldsymbol{X}$

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: $\boldsymbol{w}$

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

$=$

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

-6	**-11**	**-12**	**-11**
-7			

Computing one output value:

$$1 \cdot 0 + 1 \cdot 2 + 1 \cdot 3 + 0 \cdot 4 + 0 \cdot 5 + 0 \cdot 5 + (-1) \cdot 6 + (-1) \cdot 3 + (-1) \cdot 4 = \mathbf{-7}$$

2D Convolution

Data: $\boldsymbol{X}$

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 x 6

$*$

Weights: $\boldsymbol{w}$

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

3 x 3

$=$

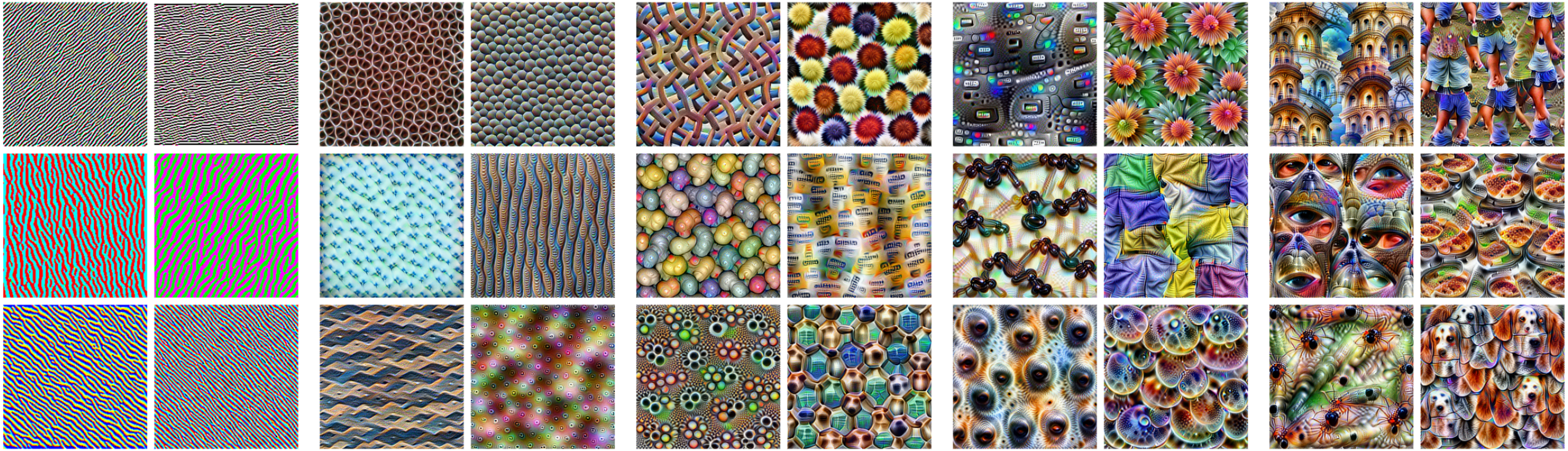
Output: $\boldsymbol{X} * \boldsymbol{w}$

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

4 x 4

2D Convolution

Features



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*

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

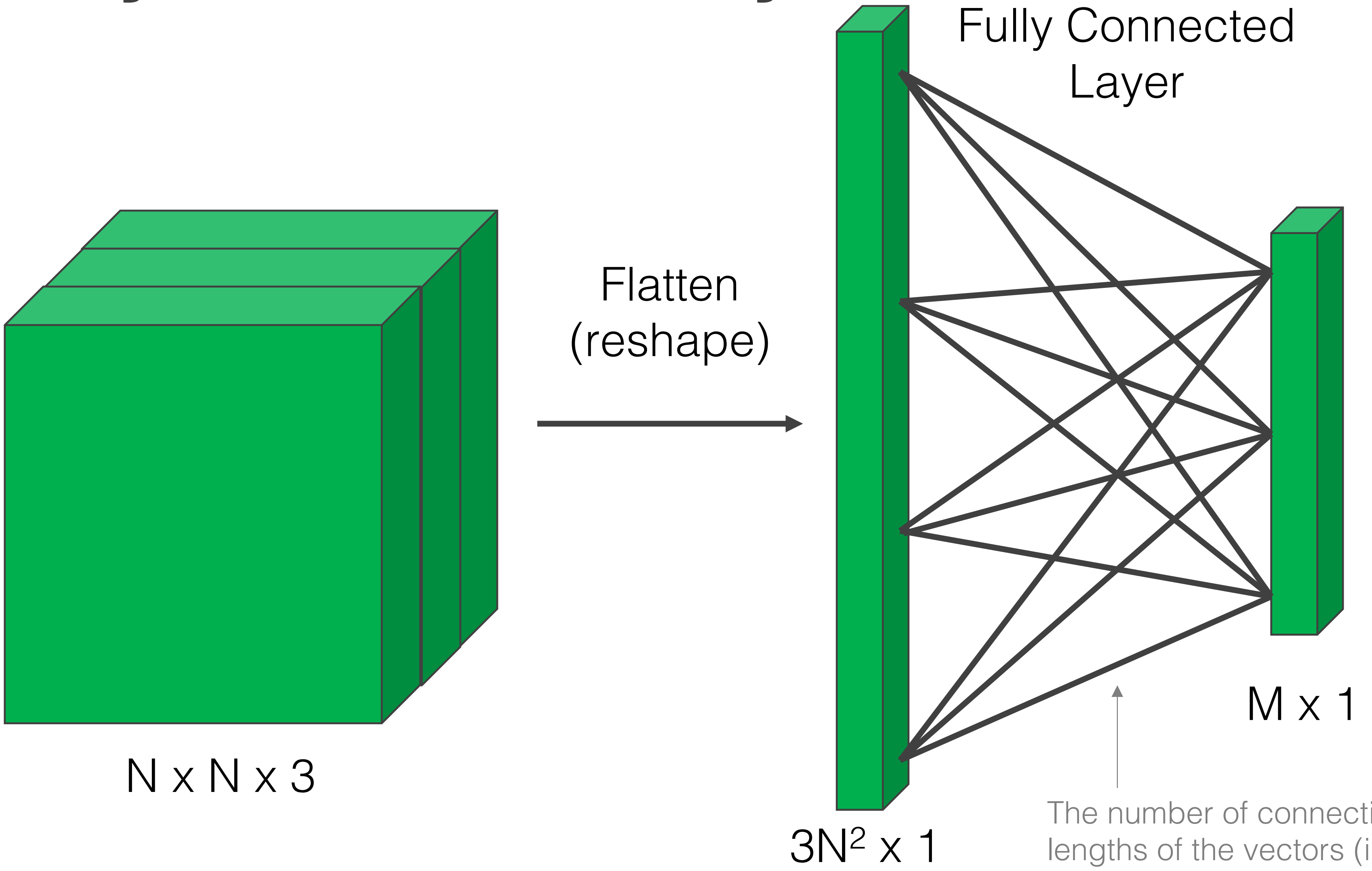
Resources on Visualization of Features

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

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

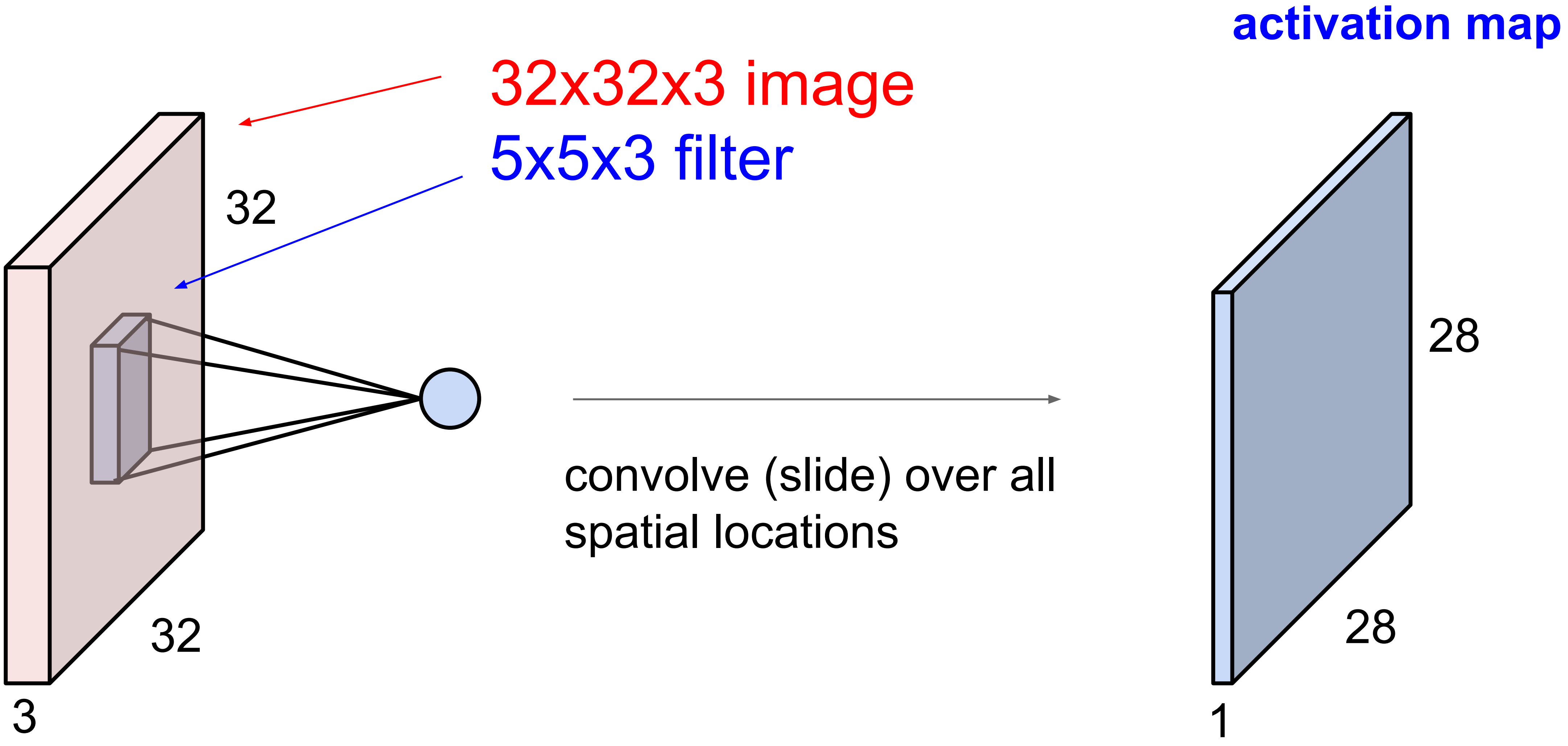
Activation Activation Atlases: https://distill.pub/2019/activation-atlas/

Fully Connected Layer



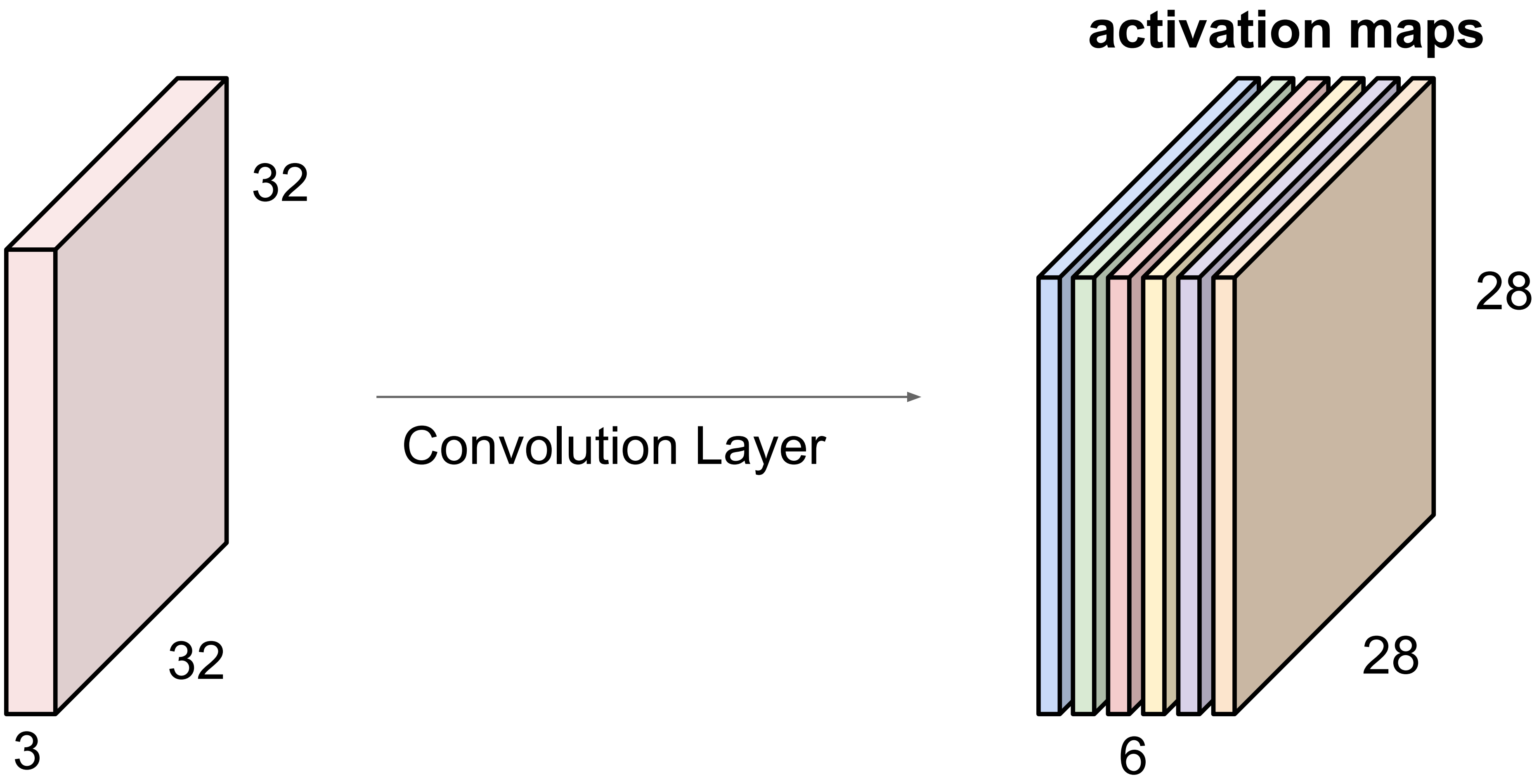
The number of connections is the product of the lengths of the vectors (in this case $3N^2M$)

Convolution Layer



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

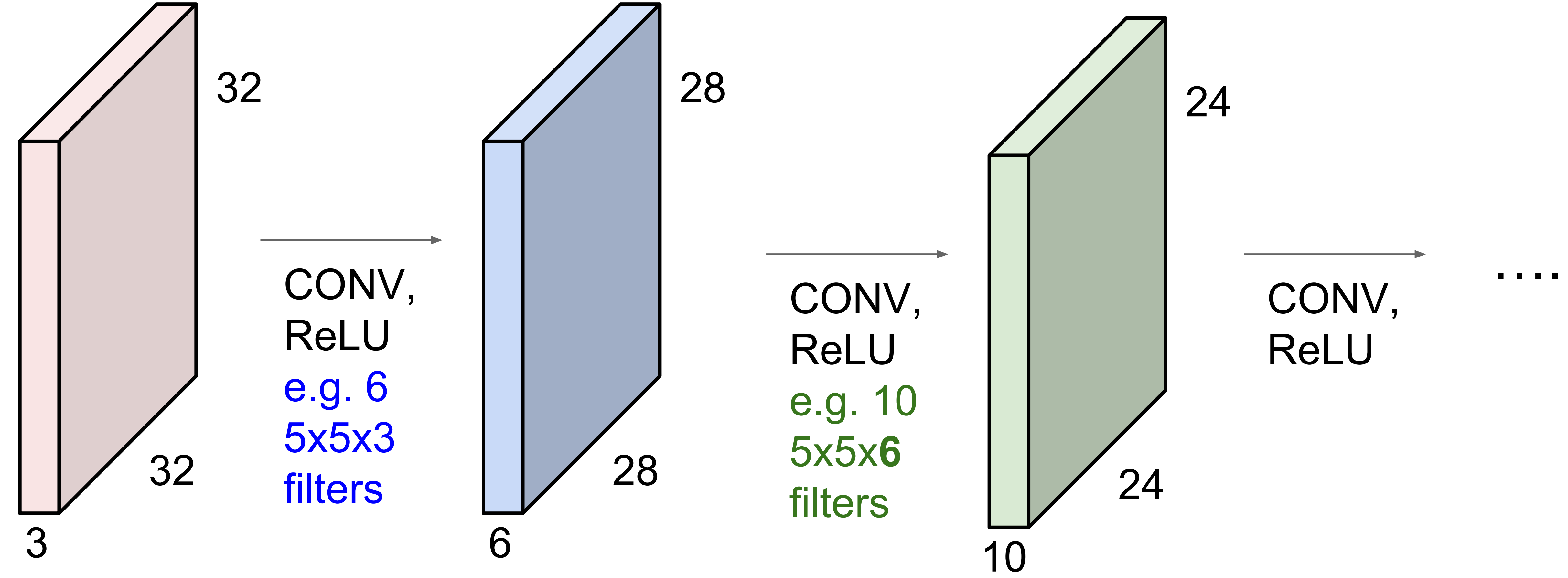
For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



We stack these up to get a "new image" of size 28x28x6!

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

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

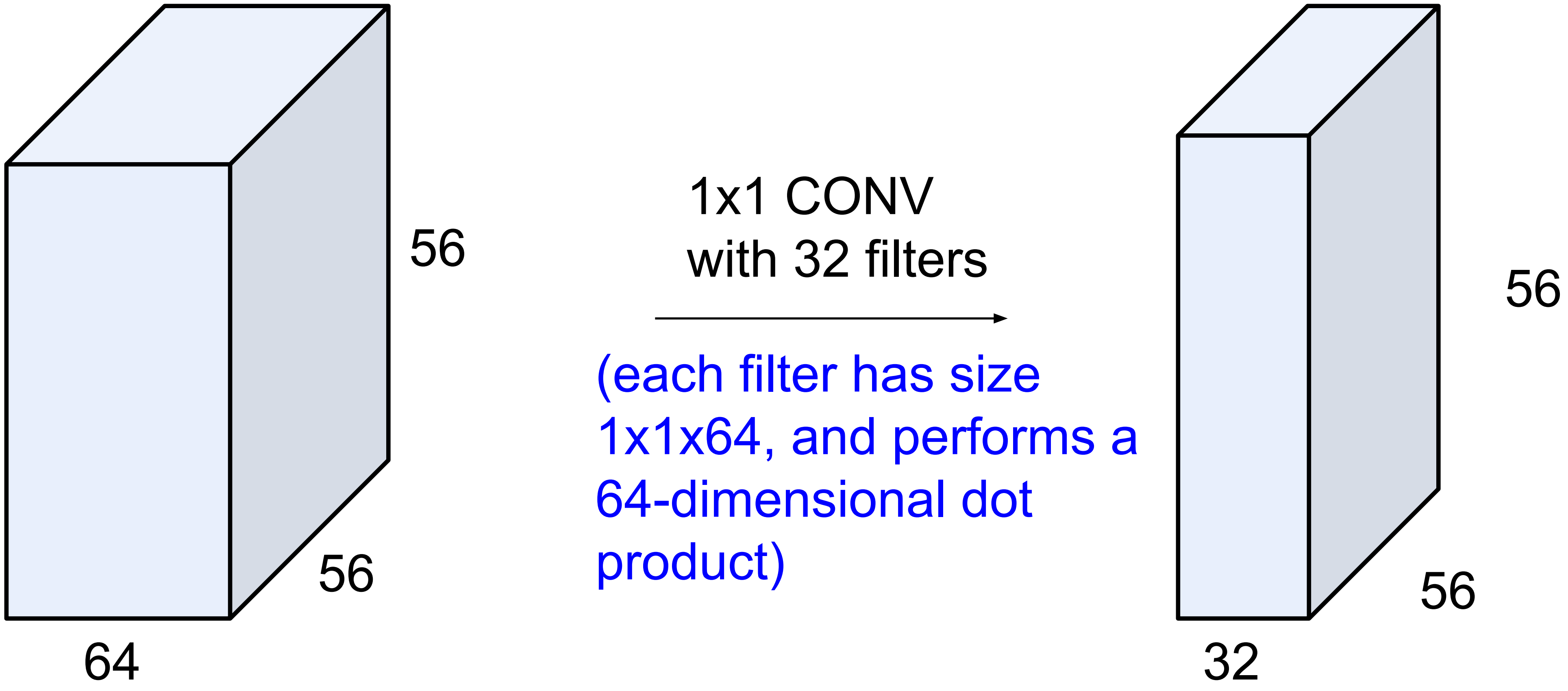


Parameters = (5*5*3)*6 = 450

(5*5*6)*10 = 1,500

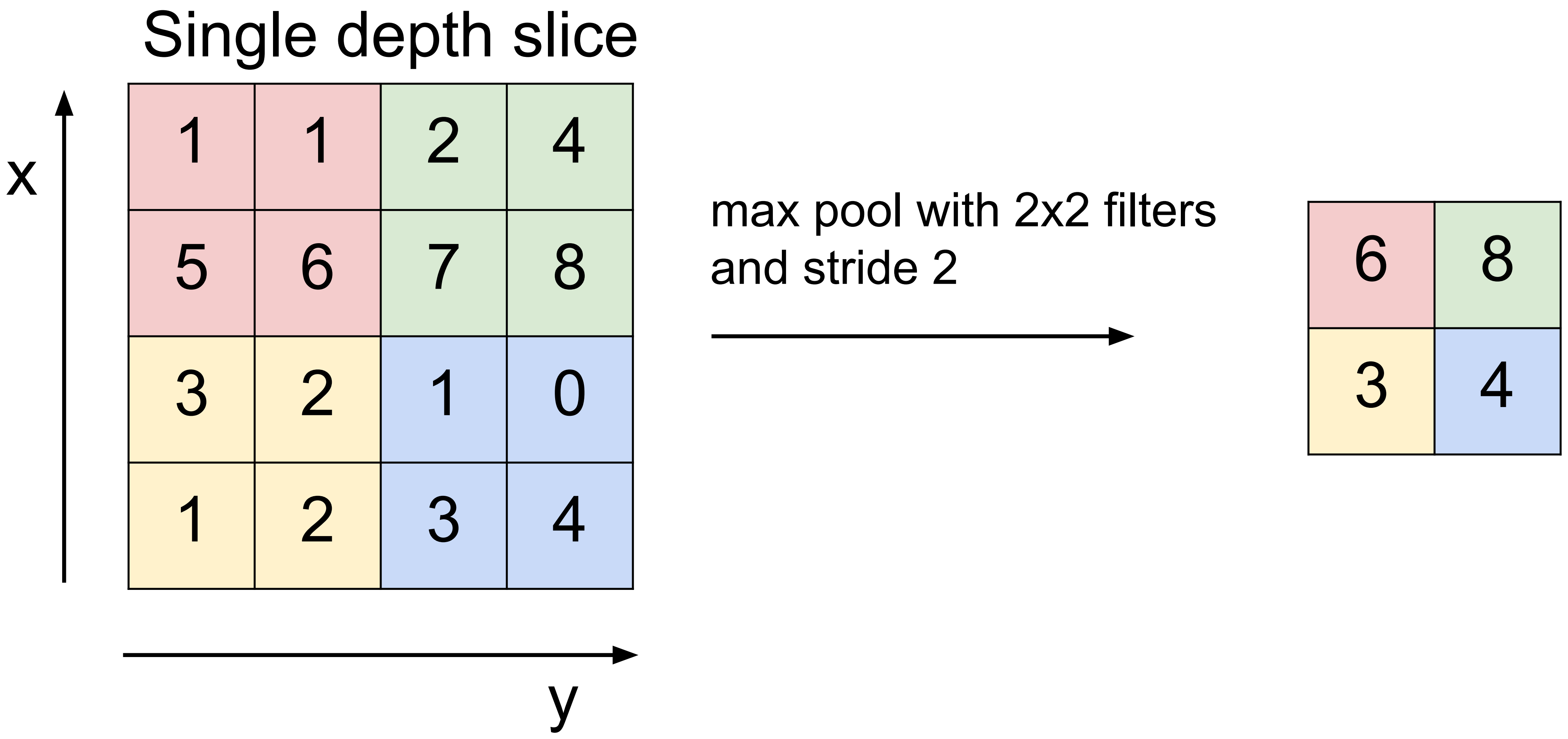
From Fei-Fei Li, Justin Johnson, and Serena Young. CS231n, 2018

1 x 1 Convolution Explained



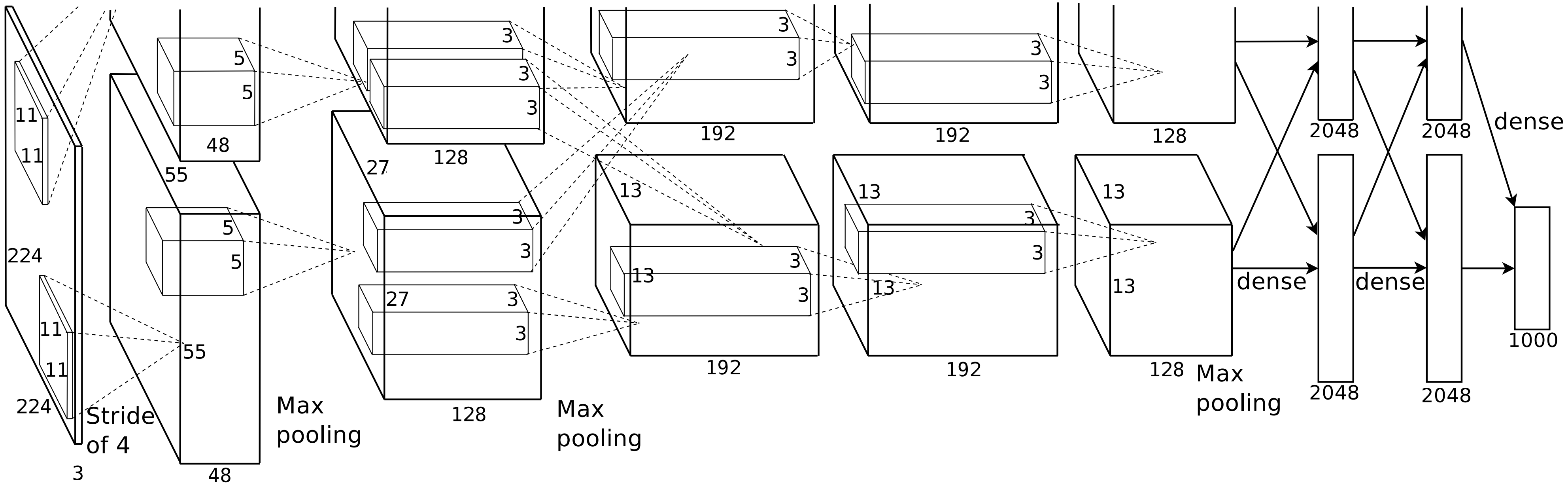
From Fei-Fei Li, Justin Johnson, and Serena Young. CS231n, 2018

Max Pooling



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

AlexNet



Input
11x11 conv, 96
5x5 conv, 256
max pool
3x3 conv, 384
max pool
3x3 conv, 384
3x3 conv, 256
max pool
FC 4096
FC 4096
FC 1000
softmax

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

CNN Architectures

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

AlexNet (2012)

- Input
- 11x11 conv, 96
- 5x5 conv, 256
- max pool
- 3x3 conv, 384
- max pool
- 3x3 conv, 384
- 3x3 conv, 256
- max pool
- FC 4096
- FC 4096
- FC 1000
- softmax

Fewer layers, larger filters

VGG16 (2014)

- Input
- 3x3 conv, 64
- 3x3 conv, 64
- max pool
- 3x3 conv, 128
- 3x3 conv, 128
- max pool
- 3x3 conv, 256
- 3x3 conv, 256
- 3x3 conv, 256
- max pool
- 3x3 conv, 512
- 3x3 conv, 512
- 3x3 conv, 512
- max pool
- 3x3 conv, 512
- 3x3 conv, 512
- 3x3 conv, 512
- max pool
- FC 4096
- FC 4096
- FC 1000
- softmax

VGG19 (2014)

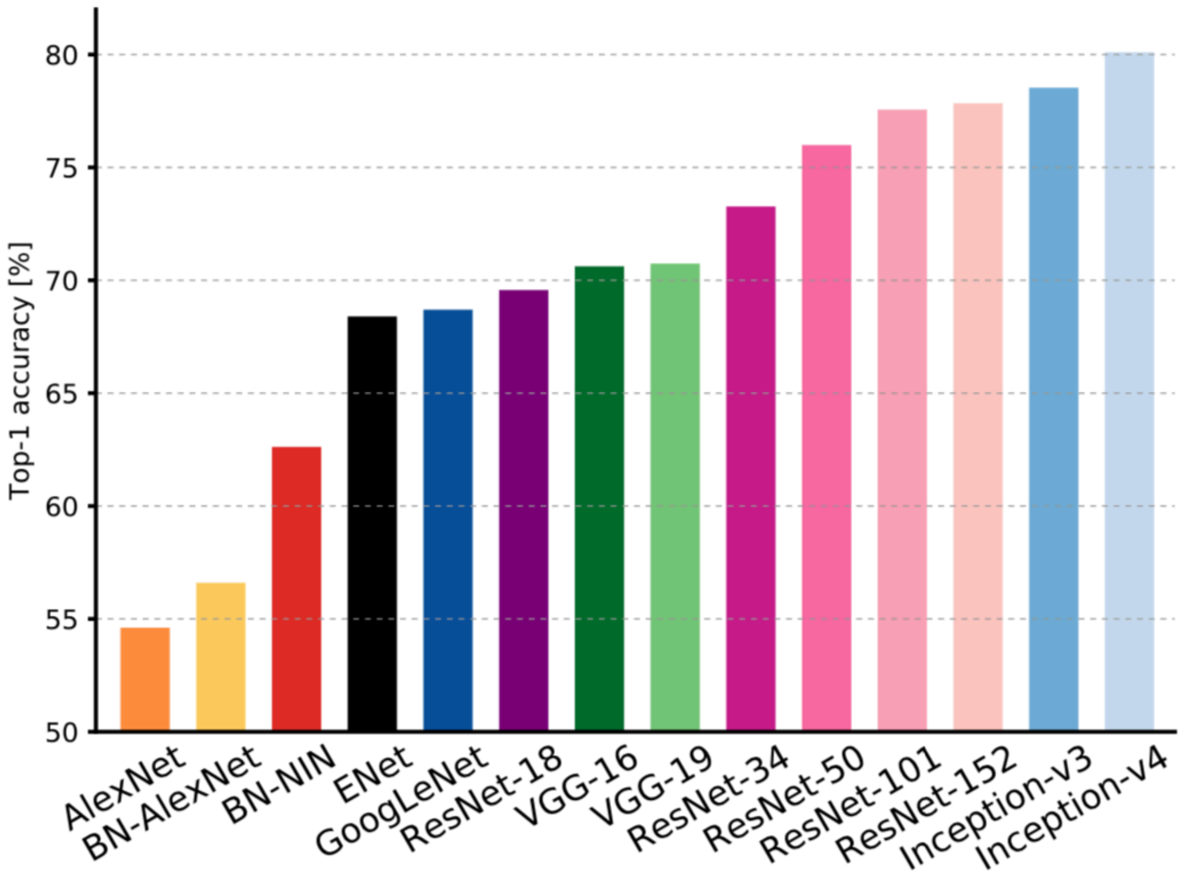
- Input
- 3x3 conv, 64
- 3x3 conv, 64
- max pool
- 3x3 conv, 128
- 3x3 conv, 128
- max pool
- 3x3 conv, 256
- 3x3 conv, 256
- 3x3 conv, 256
- 3x3 conv, 256
- max pool
- 3x3 conv, 512
- 3x3 conv, 512
- 3x3 conv, 512
- 3x3 conv, 512
- max pool
- 3x3 conv, 512
- 3x3 conv, 512
- 3x3 conv, 512
- 3x3 conv, 512
- max pool
- FC 4096
- FC 4096
- FC 1000
- softmax

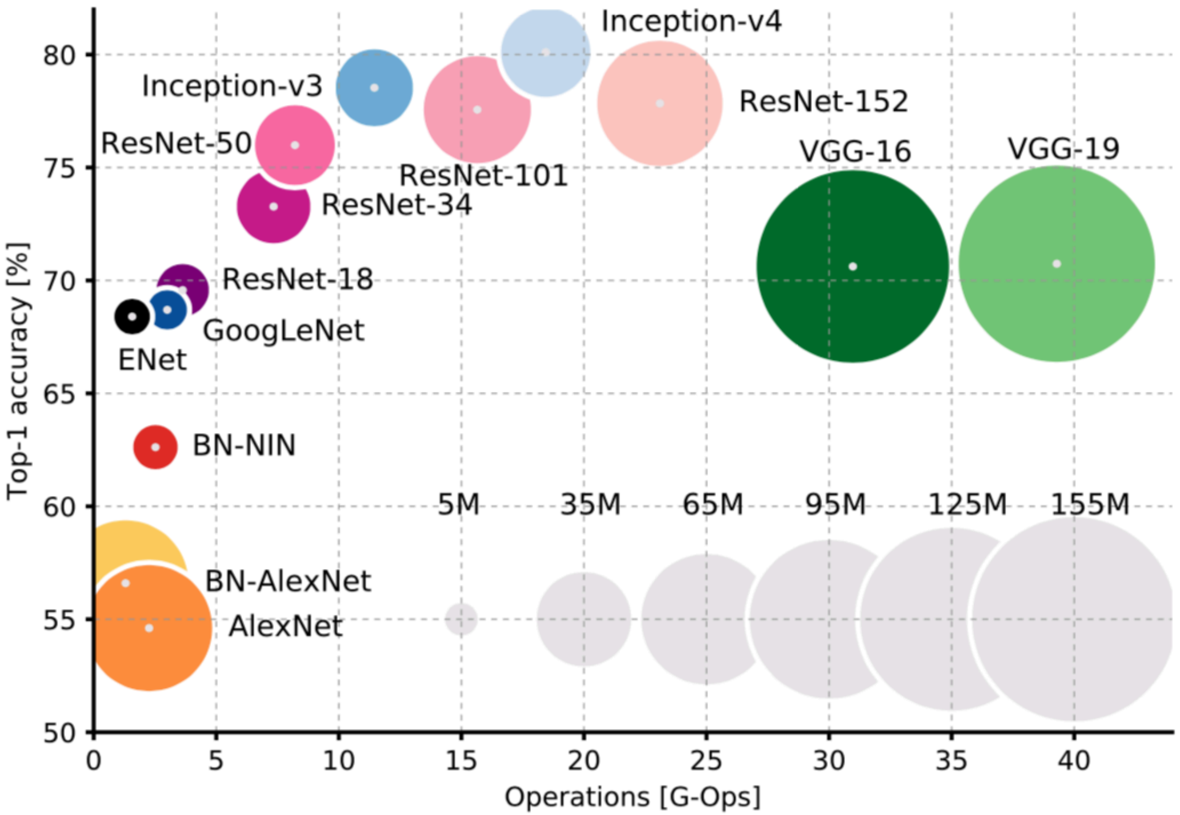
Key

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

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

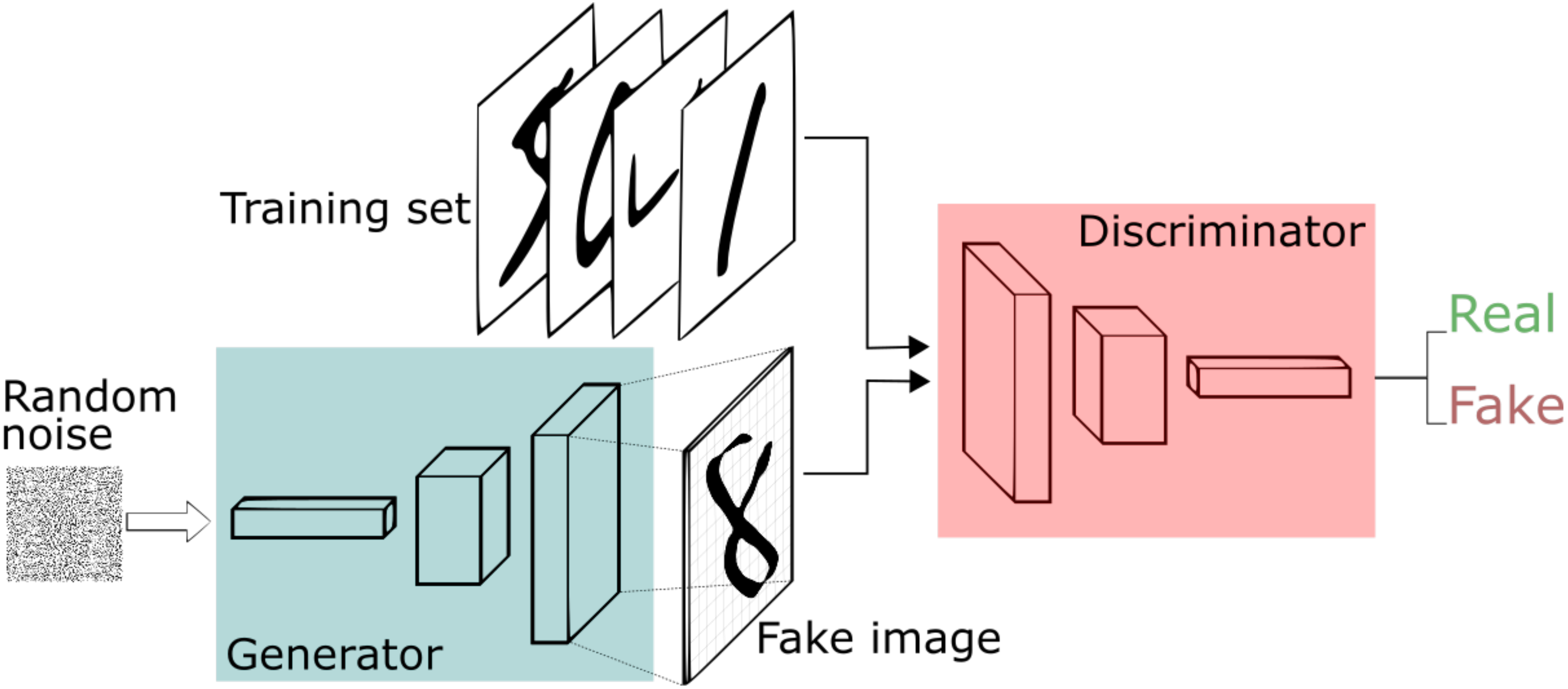


Image from: https://skymind.ai/wiki/generative-adversarial-network-gan