Deep Learning

Machine Learning

Prof. Sandra Avila
Institute of Computing (IC/Unicamp)

MC886/MO444, October 25, 2022

Today's Agenda

- What is Deep Learning?
- Deep Learning & Applications
- Neural Networks *vs.* Convolutional Networks
- What is a convolution?
- Convolutional Neural Networks
    - Convolution Layer

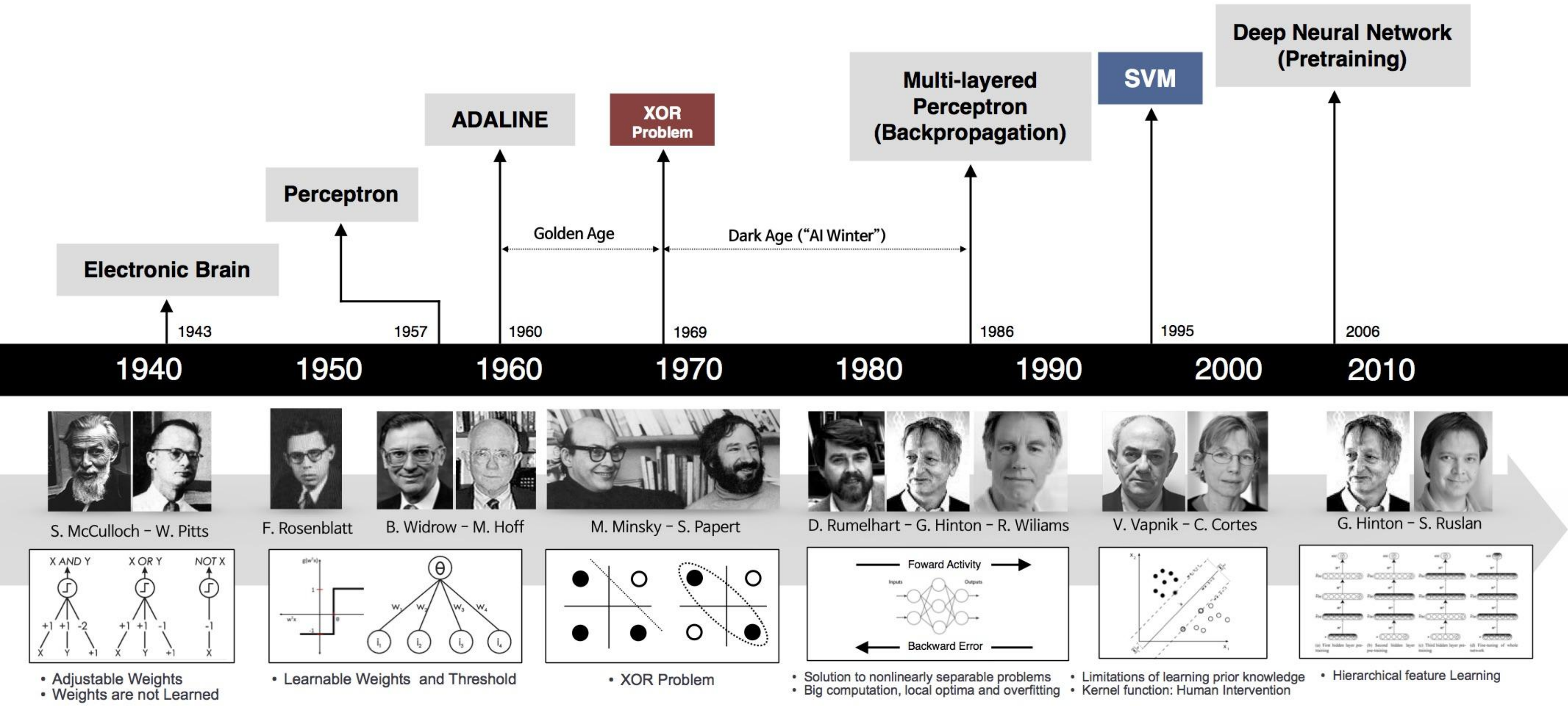
What is
Deep Learning?

"Deep learning allows computers to learn from experience and understand the world in terms of a hierarchy of concepts, with each concept defined through its relation to simpler concepts.

[Goodfellow & Bengio & Courville, 2016]

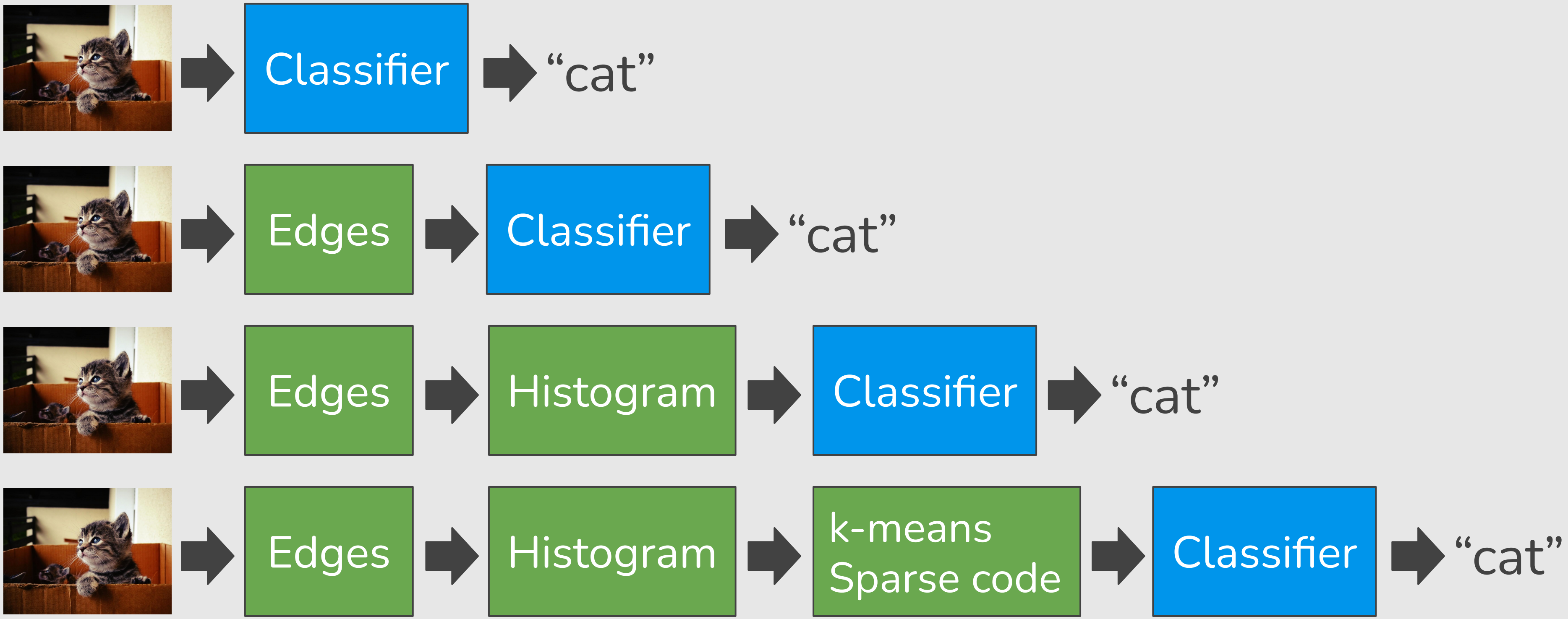
"Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction."

[LeCun & Bengio & Hinton, 2015]



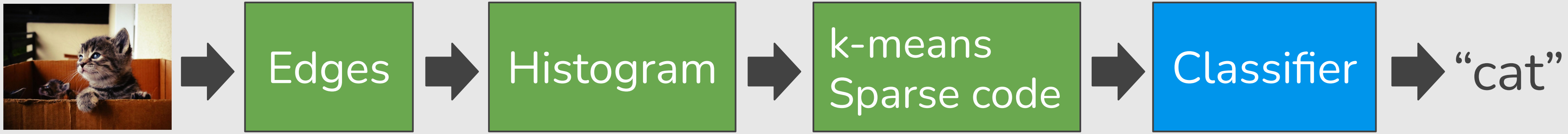
Note: I am not the creator of the above image. I have been unable to locate the original source. If this is your image please let me know so I can give you proper attribution.

Traditional Recognition

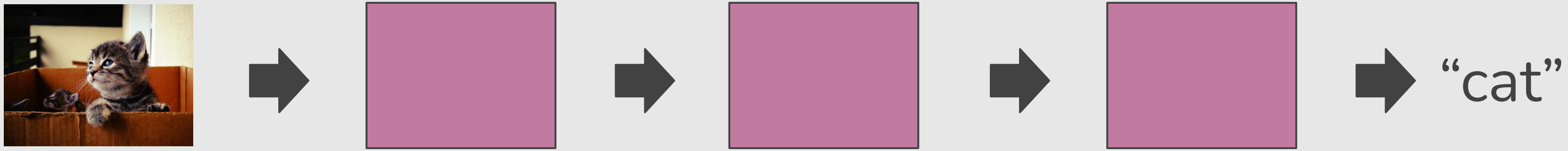


Deep Learning

Specialized components



Generic components



Generic components, going deeper

Deep Learning: Applications

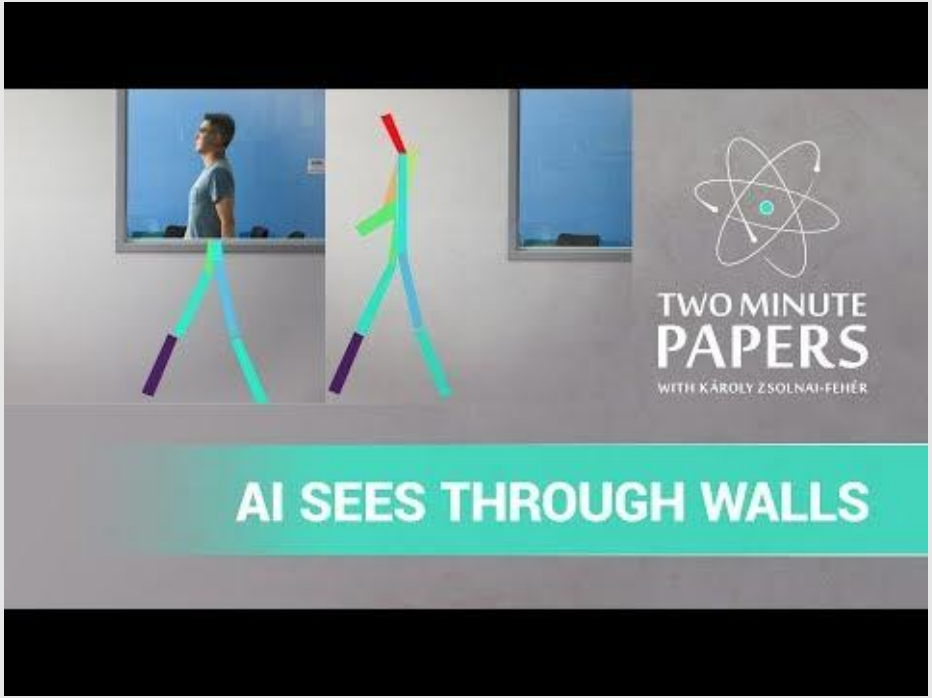
KTH Dataset (2005)

http://www.nada.kth.se/cvap/actions

DL is everywhere ... pose estimation

“Realtime Multi-Person 2D Human Pose Estimation using Part Affinity Fields”, CVPR 2017

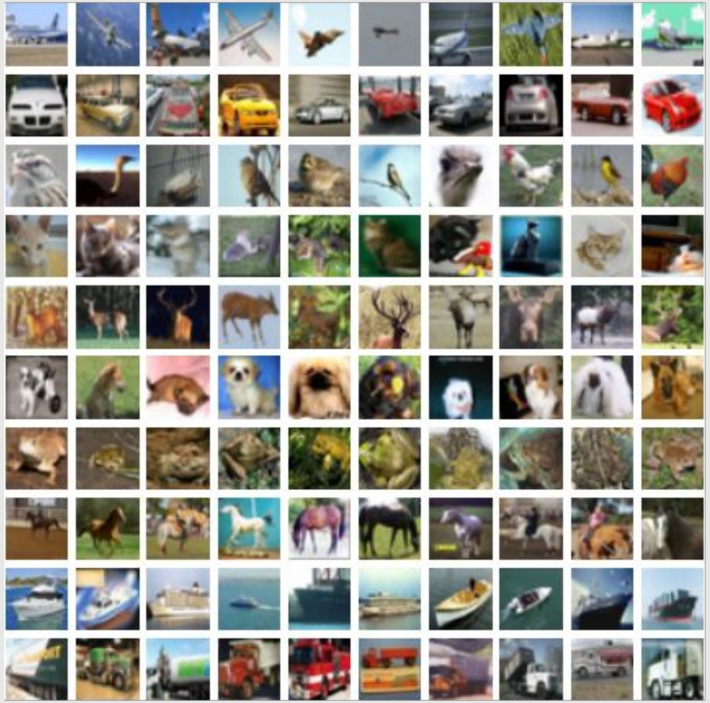
DL is everywhere ... pose estimation



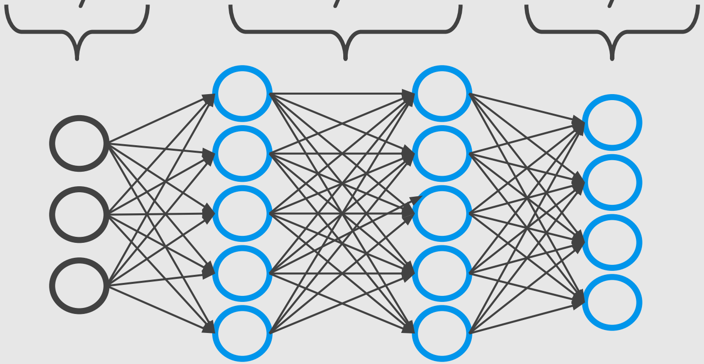
"Through-Wall Human Pose Estimation Using Radio Signals", CVPR 2018

Neural Networks vs. Convolutional Networks

Neural Networks



CIFAR-10



32 x 32 x 3 image ⇒ stretch to 3072 x 1

Neural Networks

Neural Networks

0	0	3	2
1	1	0	1
4	2	1	2
0	2	1	5

Neural Networks

0
0
3
2

0	0	3	2
1	1	0	1
4	2	1	2
0	2	1	5

Neural Networks

0	0	3	2
1	1	0	1
4	2	1	2
0	2	1	5

0
0
3
2
1
1
0
1

Neural Networks

0	0	3	2
1	1	0	1
4	2	1	2
0	2	1	5

0
0
3
2
1
1
0
1
⋮
2
1
5

What is a Convolution?

What is a Convolution?

Convolution is the process of adding each element of the image to its local neighbors, **weighted by the kernel**.

What is a Convolution?

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

5 x 5 matrix
(image)

3 x 3 filter

What is a Convolution?

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

3 x 3 filter

5 x 5 matrix
(image)

What is a Convolution?

$$
\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}
*
\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}
\rightarrow
\begin{bmatrix} 4 & & \\ & & \\ & & \end{bmatrix}
$$

5 x 5 matrix
(image)

3 x 3 filter

$$
\begin{aligned}
&1*1 + 1*0 + 1*1 + \\
&0*0 + 1*1 + 1*0 + \\
&0*1 + 0*0 + 1*1 = 4
\end{aligned}
$$

What is a Convolution?

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 3 & \\ & & \\ & & \end{bmatrix}$$

5 x 5 matrix
(image)

3 x 3 filter

$$1*1 + 1*0 + 0*1 + 1*0 + 1*1 + 1*0 + 0*1 + 1*0 + 1*1 = 3$$

What is a Convolution?

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

5 x 5 matrix
(image)

*

1	0	1
0	1	0
1	0	1

3 x 3 filter

→

4	3	4

1*1 + 0*0 + 0*1 +
1*0 + 1*1 + 0*0 +
1*1 + 1*0 + 1*1 = 4

What is a Convolution?

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

5 x 5 matrix
(image)

*

1	0	1
0	1	0
1	0	1

3 x 3 filter

→

4	3	4
2		

0*1 + 1*0 + 1*1 +
0*0 + 0*1 + 1*0 +
0*1 + 0*0 + 1*1 = 2

What is a Convolution?

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

5 x 5 matrix
(image)

*

1	0	1
0	1	0
1	0	1

3 x 3 filter

4	3	4
2	4	

$$1*1 + 1*0 + 1*1 +$$
$$0*0 + 1*1 + 1*0 +$$
$$0*1 + 1*0 + 1*1 = 4$$

What is a Convolution?

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

5 x 5 matrix
(image)

*

1	0	1
0	1	0
1	0	1

3 x 3 filter

→

4	3	4
2	4	3

1*1 + 1*0 + 0*1 +
1*0 + 1*1 + 1*0 +
1*1 + 1*0 + 0*1 = 3

What is a Convolution?

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

5 x 5 matrix
(image)

*

1	0	1
0	1	0
1	0	1

3 x 3 filter

4	3	4
2	4	3
2		

$$0*1 + 0*0 + 1*1 + 0*0 + 0*1 + 1*0 + 0*1 + 1*0 + 1*1 = 2$$

What is a Convolution?

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

5 x 5 matrix
(image)

*

1	0	1
0	1	0
1	0	1

3 x 3 filter

→

4	3	4
2	4	3
2	3	

$$0*1 + 1*0 + 1*1 + 0*0 + 1*1 + 1*0 + 1*1 + 1*0 + 0*1 = 3$$

What is a Convolution?

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

5 x 5 matrix
(image)

3 x 3 filter

$$1*1 + 1*0 + 1*1 + 1*0 + 1*1 + 0*0 + 1*1 + 0*0 + 0*1 = 4$$

What is a Convolution?

*

0	1	0
1	-4	1
0	1	0

What is a Convolution?

*

Edge Detection

0	1	0
1	-4	1
0	1	0

What is a Convolution?

Emboss

$$* \begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \rightarrow$$

What is a Convolution?

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

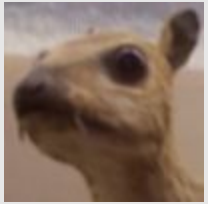
Edge Detection

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Sharpen

$$1/9 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Box blur



$$1/16 \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

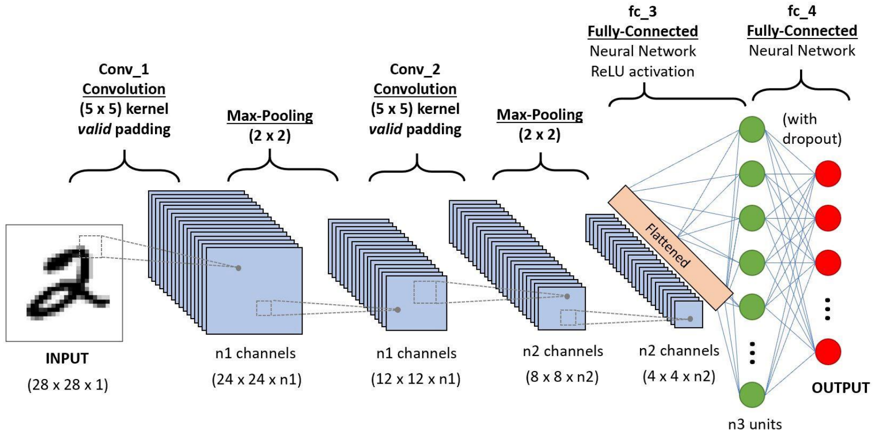
Gaussian blur 3×3

Today’s Agenda

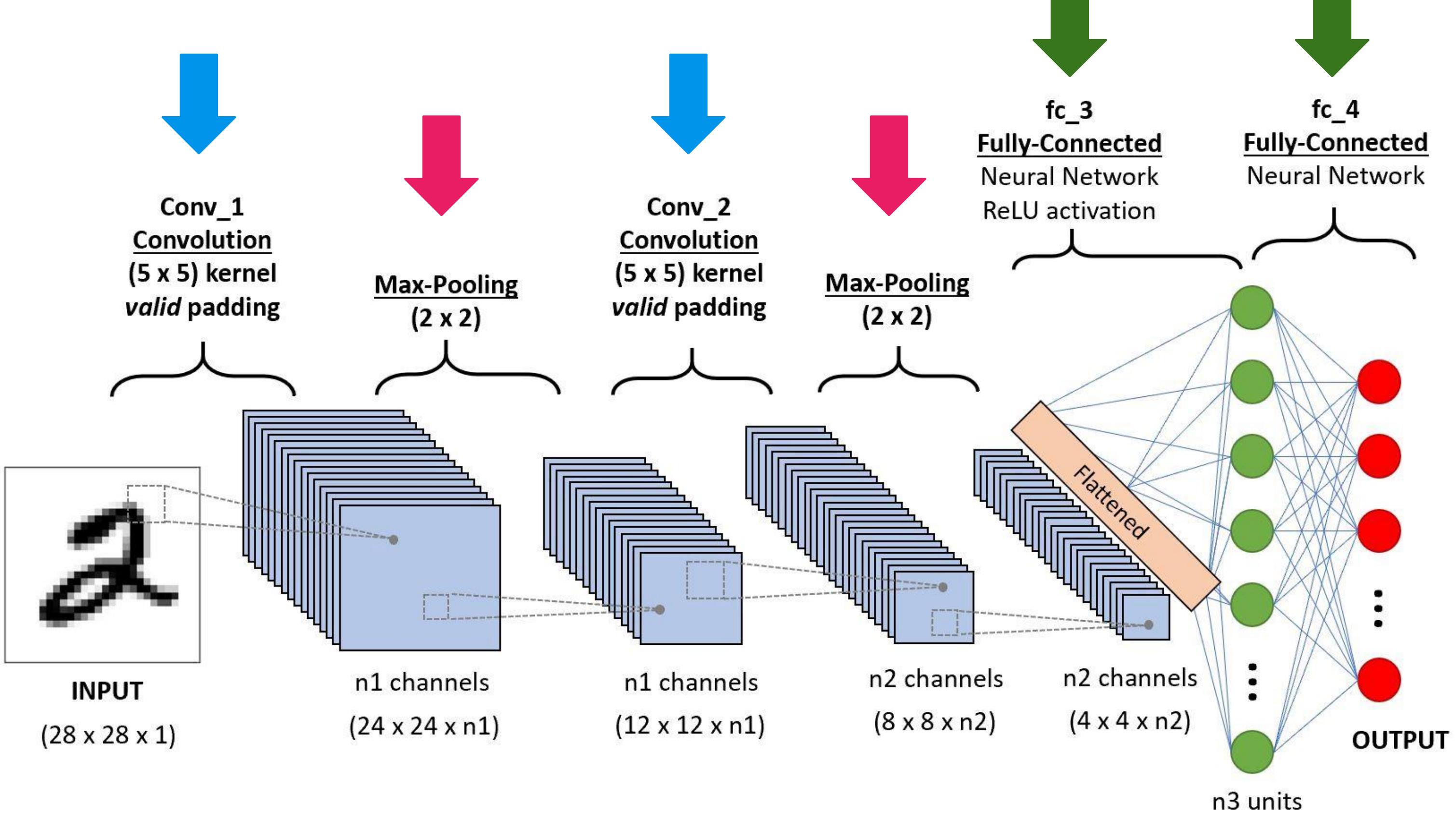
- What is Deep Learning?
- Deep Learning & Applications
- Neural Networks *vs.* Convolutional Networks
- What is a convolution?
- Convolutional Neural Networks
    - Convolution Layer

Convolutional Neural Networks (CNNs)

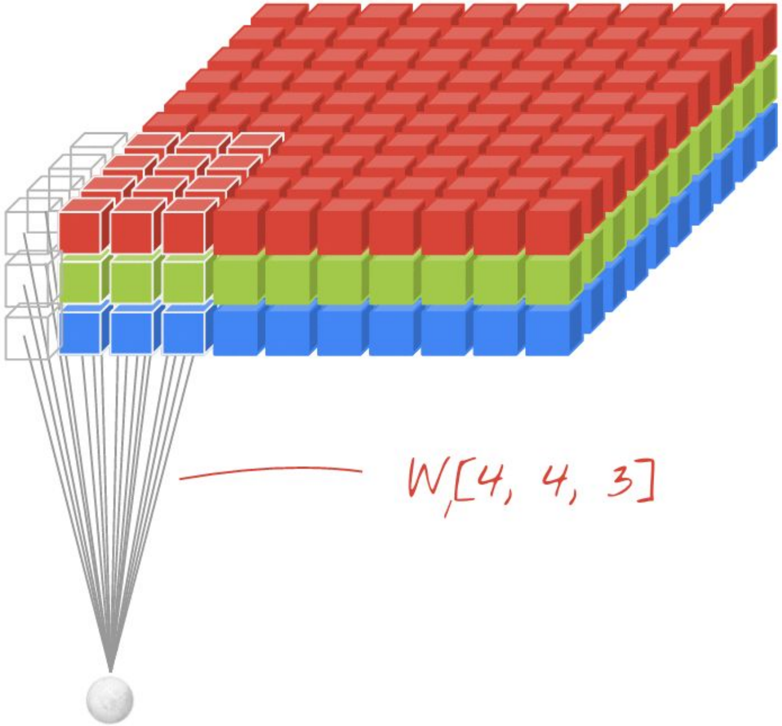
“Gradient-based learning applied to document recognition”, 1998 http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf



https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53



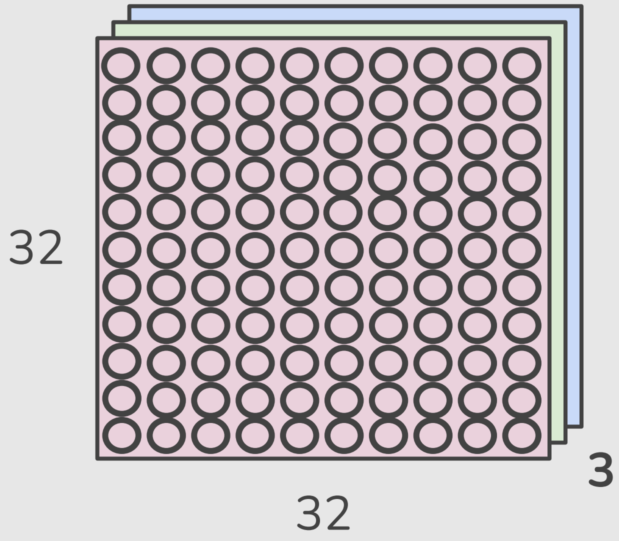
There are a few distinct types of layers (e.g., CONV/POOL/FC are by far the most popular).



Credit: Martin Görner, @martin_gorner (twitter)

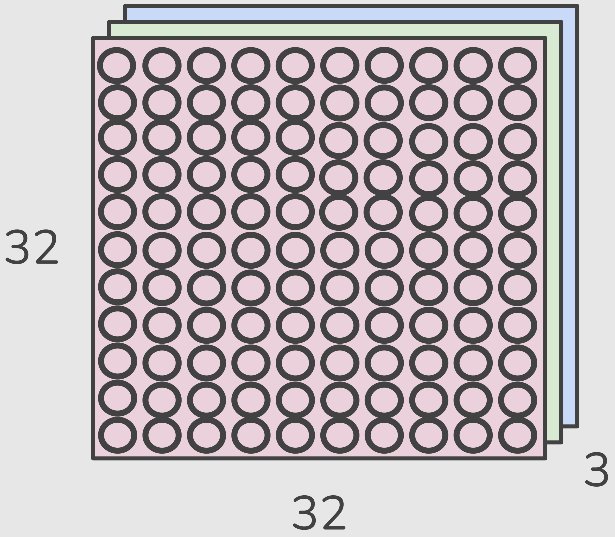
Convolution Layer

32 x 32 x 3 image ⇒ preserve spatial structure



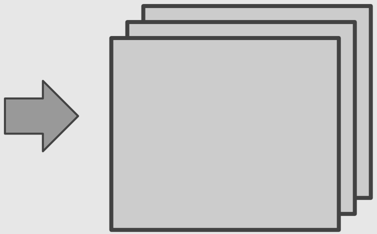
Convolution Layer

32 x 32 x 3 image ⇒ preserve spatial structure



Convolve the filter with the image i.e. "slide over the image spatially, computing dot products"

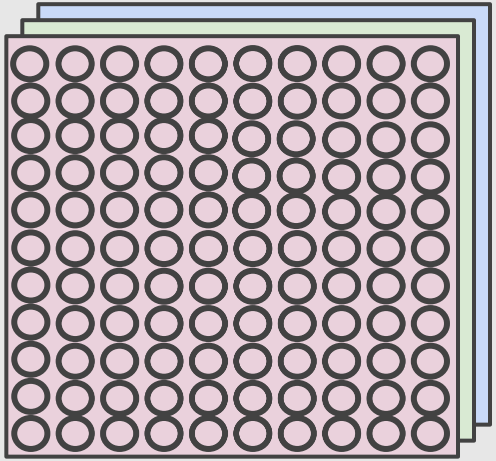
5 x 5 x 3 filter



Convolution Layer

32 x 32 x 3 image ⇒ preserve spatial structure

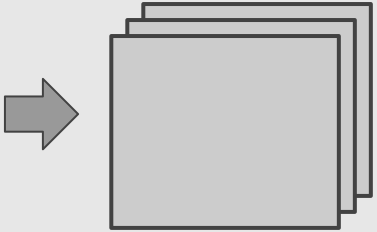
32



32 3

Convolve the filter with the image i.e. "slide over the image spatially, computing dot products"

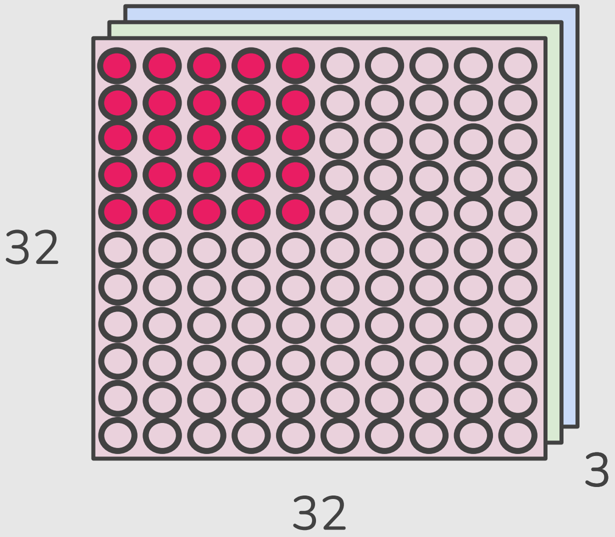
5 x 5 x 3 filter



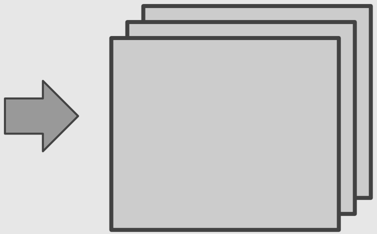
Filters always extend the full depth of the input volume

Convolution Layer

32 x 32 x 3 image ⇒ preserve spatial structure

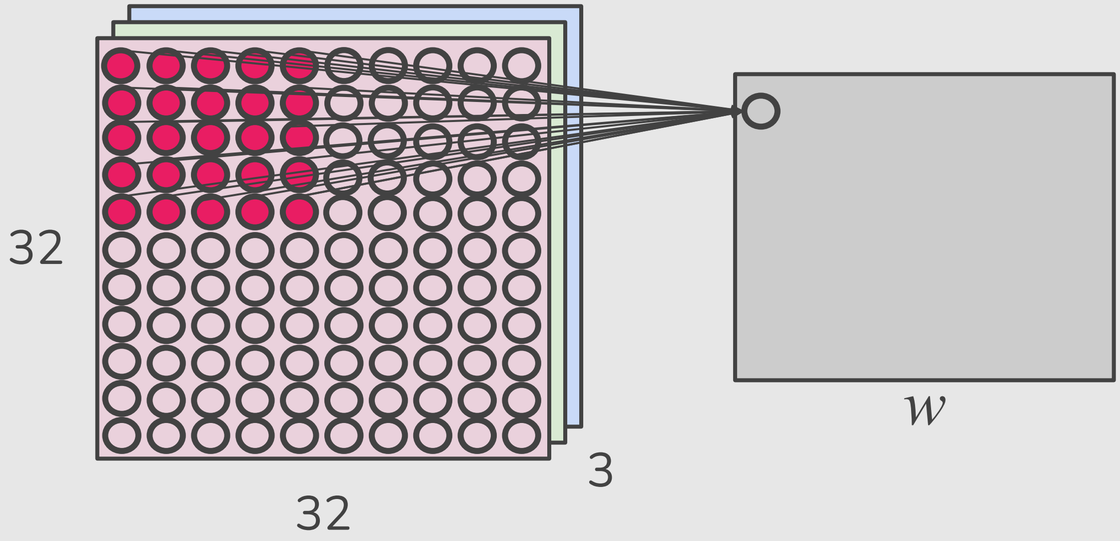


5 x 5 x 3 filter



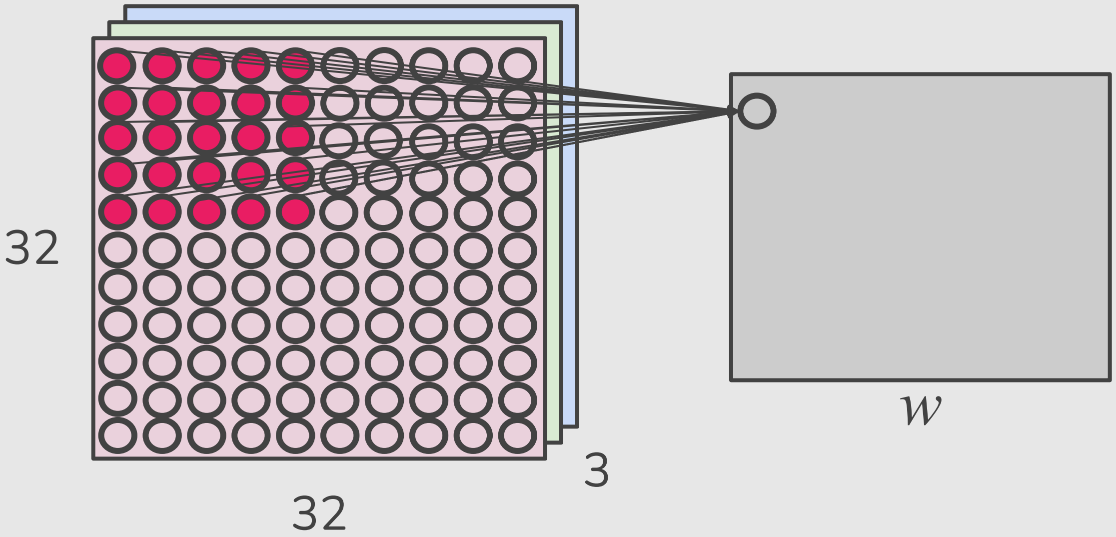
Convolution Layer

32 x 32 x 3 image ⇒ preserve spatial structure



Convolution Layer

32 x 32 x 3 image ⇒ preserve spatial structure



1 number:

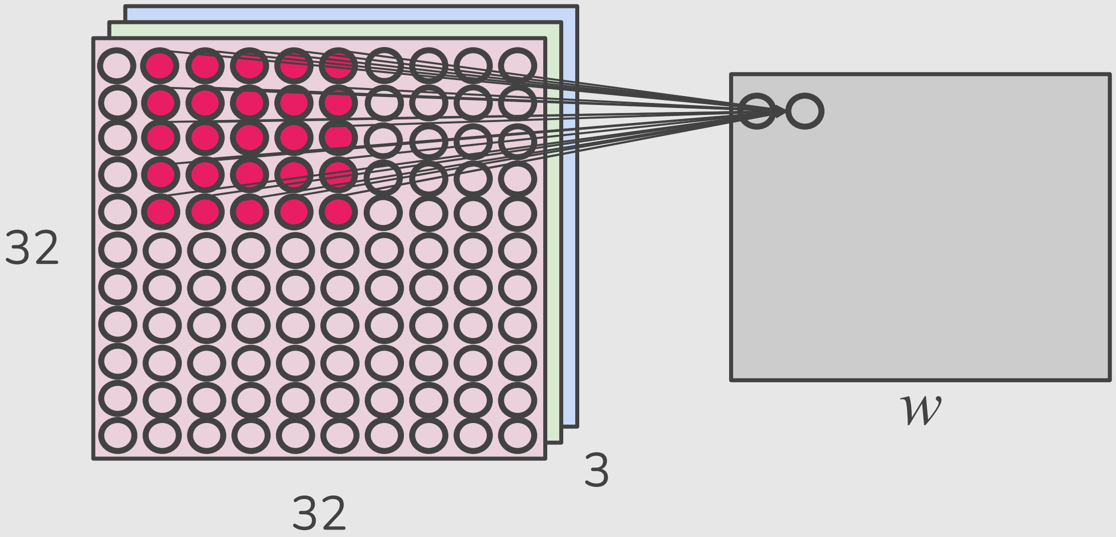
5*5*3 = 75-dimensional

dot product + bias)

$w^{\mathrm{T}}x + b$

Convolution Layer

32 x 32 x 3 image ⇒ preserve spatial structure



1 number:

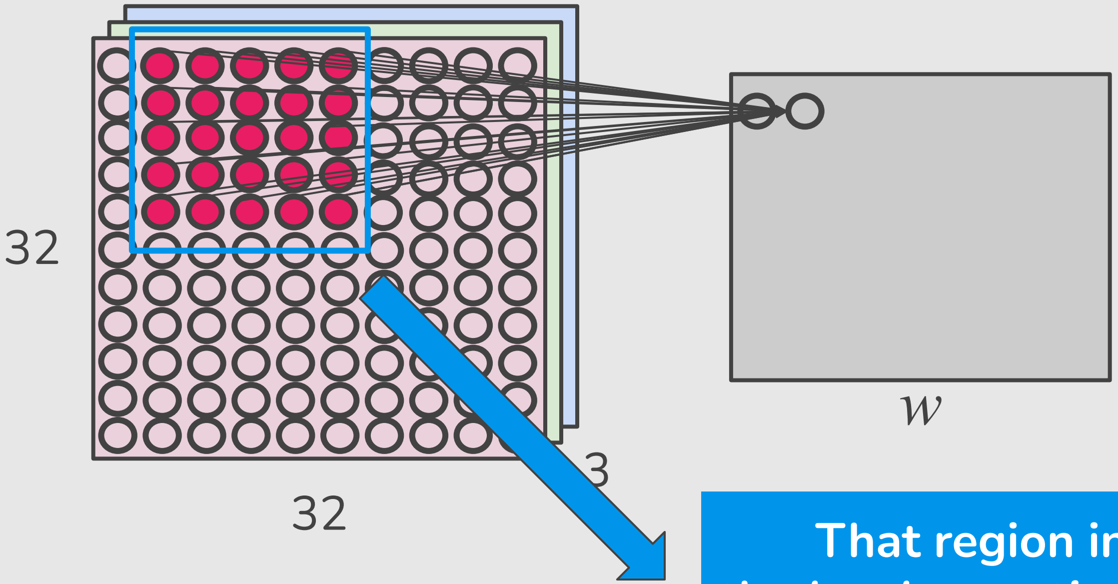
5*5*3 = 75-dimensional

dot product + bias)

$w^{\mathrm{T}}x + b$

Convolution Layer

32 x 32 x 3 image ⇒ preserve spatial structure



1 number:

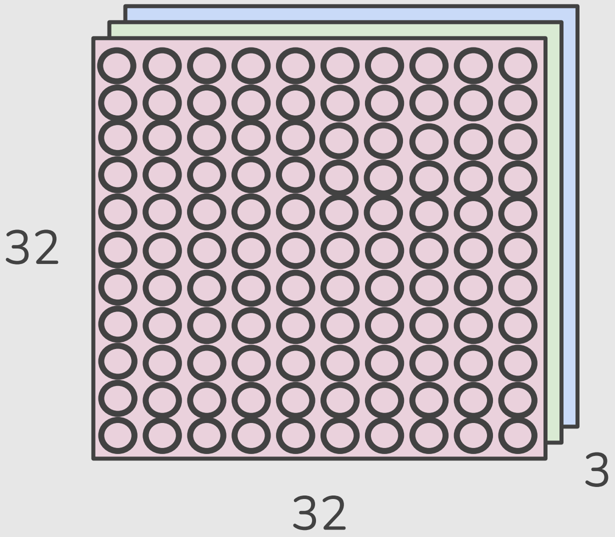
5*5*3 = 75-dimensional dot product + bias)

$w^{\mathrm{T}}x + b$

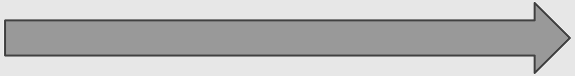
That region in the input image is called the *local receptive field* for the hidden neuron.

Convolution Layer

32 x 32 x 3 image ⇒ preserve spatial structure

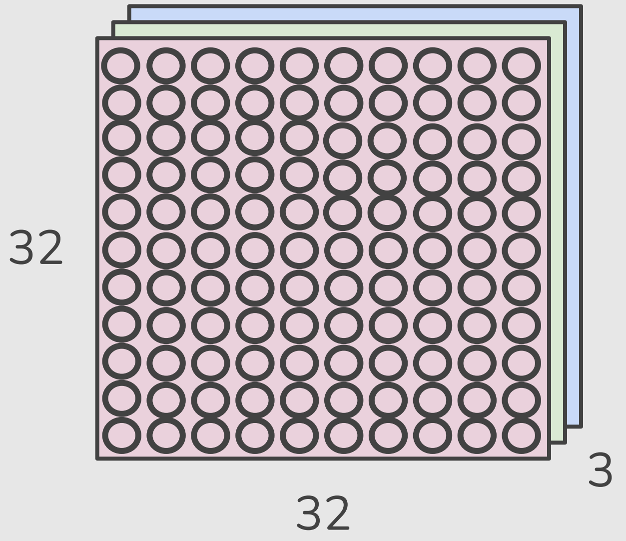


Convolve (slide) over all spatial locations

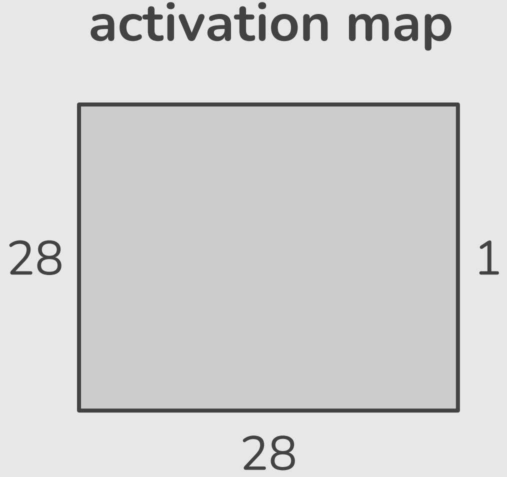


Convolution Layer

32 x 32 x 3 image ⇒ preserve spatial structure



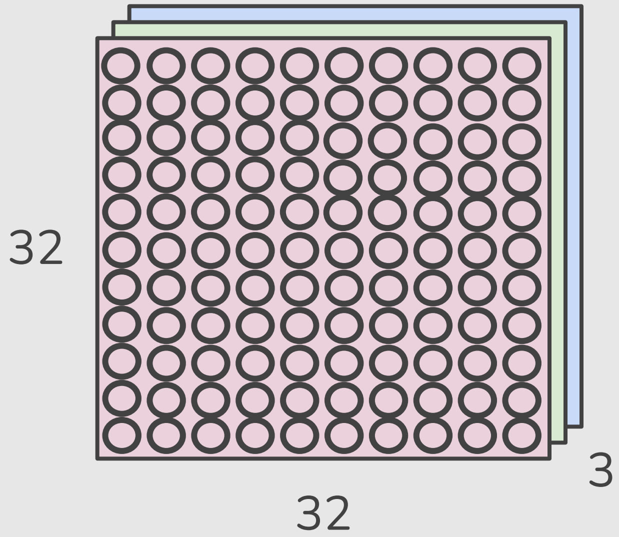
Convolve (slide) over all spatial locations



32 x 32 x 3 image
5 x 5 x 3 filter

Convolution Layer

32 x 32 x 3 image ⇒ preserve spatial structure

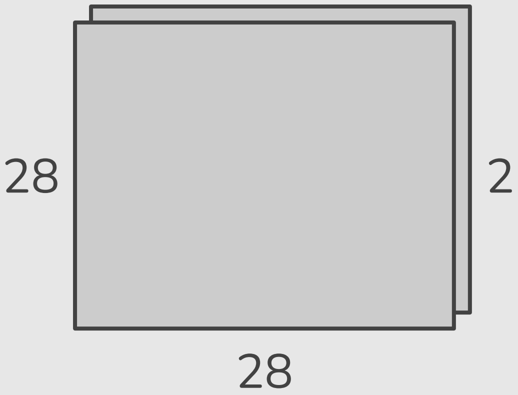


Convolve (slide) over
all spatial locations

32 x 32 x 3 image
5 x 5 x 3 filter

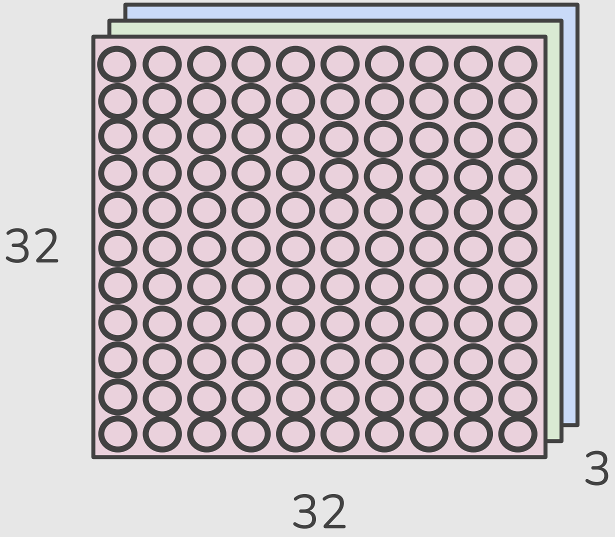
(considering a second filter)

activation maps

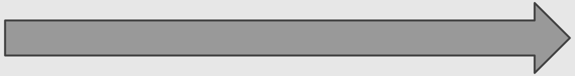


Convolution Layer

32 x 32 x 3 image ⇒ preserve spatial structure



Convolve (slide) over
all spatial locations

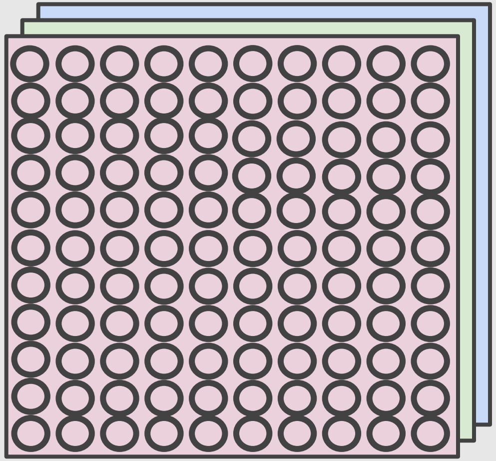


32 x 32 x 3 image
5 x 5 x 3 filter

If we had 6 5 x 5 x 3 filters ...

Convolution Layer

32 x 32 x 3 image ⇒ preserve spatial structure

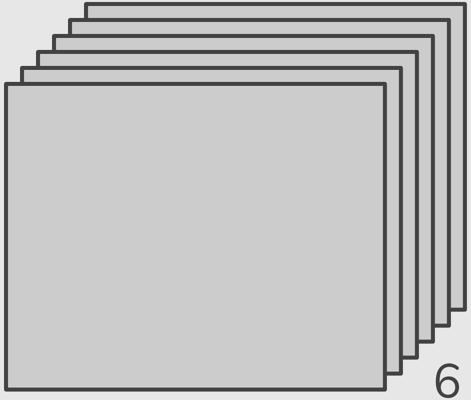


Convolve (slide) over
all spatial locations

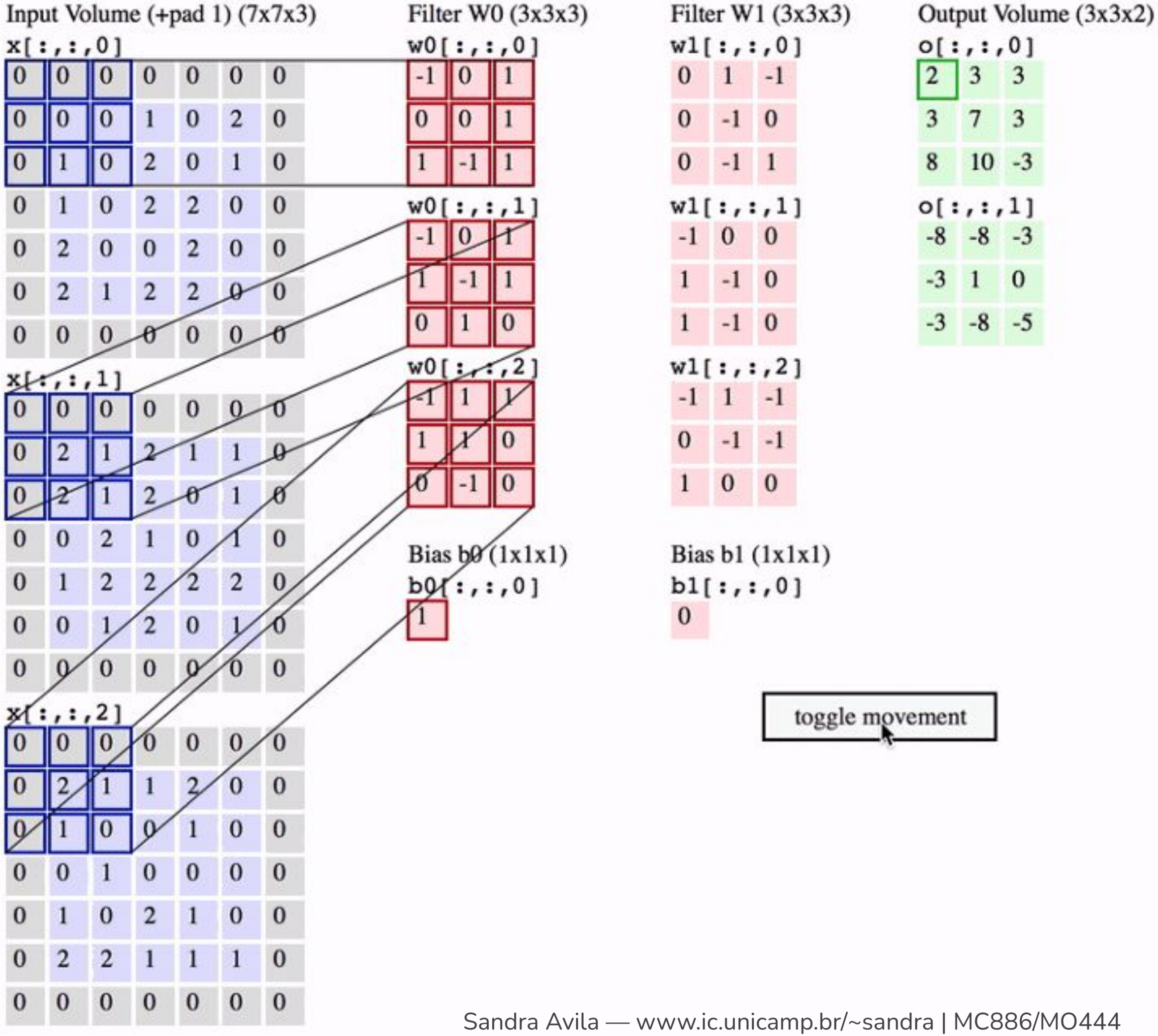
32 x 32 x 3 image
5 x 5 x 3 filter

If we had 6 5 x 5 x 3 filters ...

6 activation maps

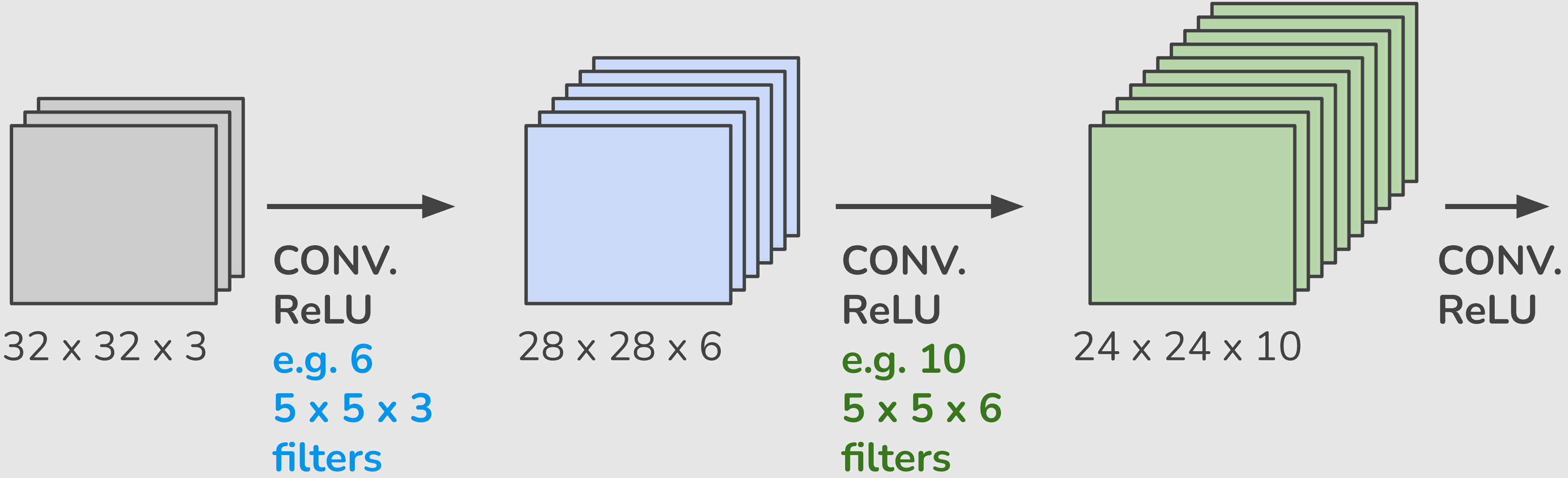


http://cs231n.github.io/convolutional-networks

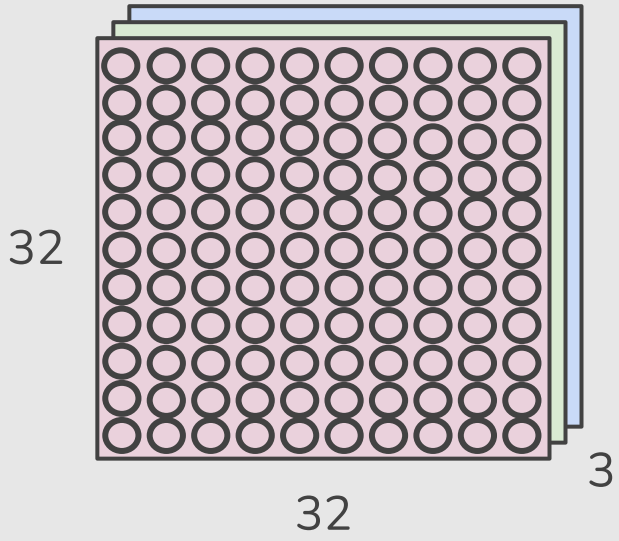


Convolutional Networks

Sequence of Convolutional Layers, interspersed with activation functions.



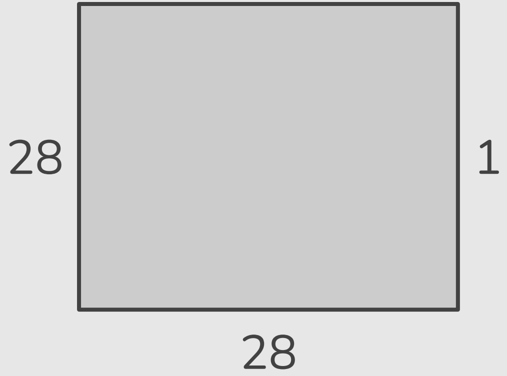
A Closer Look at Spatial Dimensions



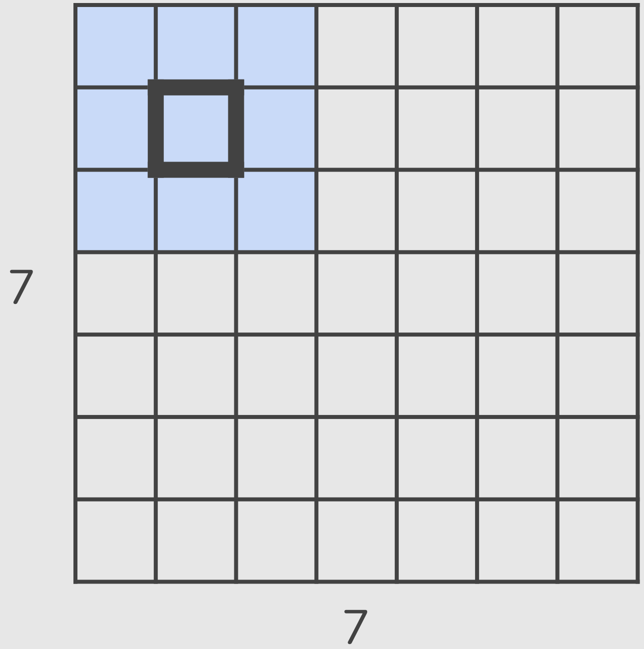
Convolve (slide) over
all spatial locations

32 x 32 x 3 image
5 x 5 x 3 filter

activation map

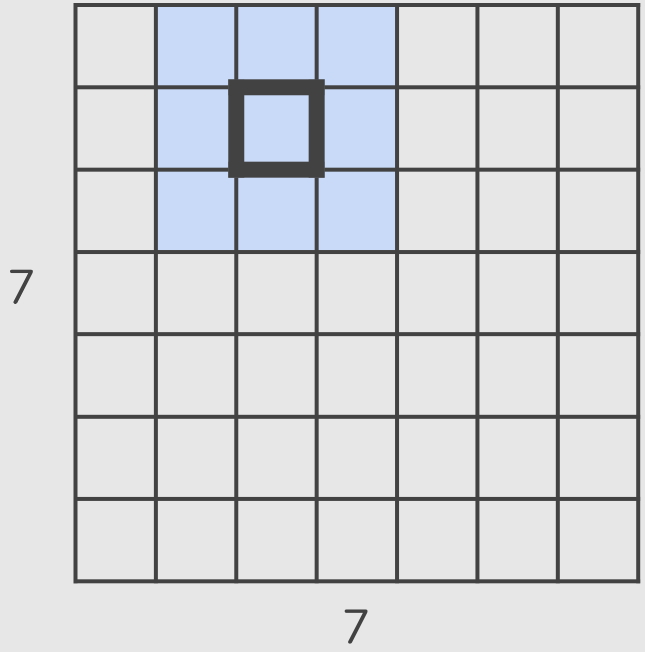


A Closer Look at Spatial Dimensions



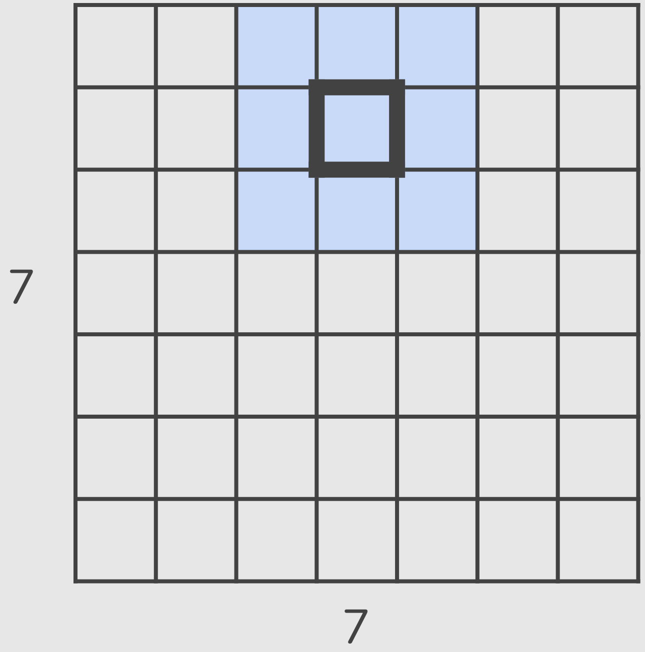
7 x 7 input (spatially)
assume 3 x 3 filter

A Closer Look at Spatial Dimensions



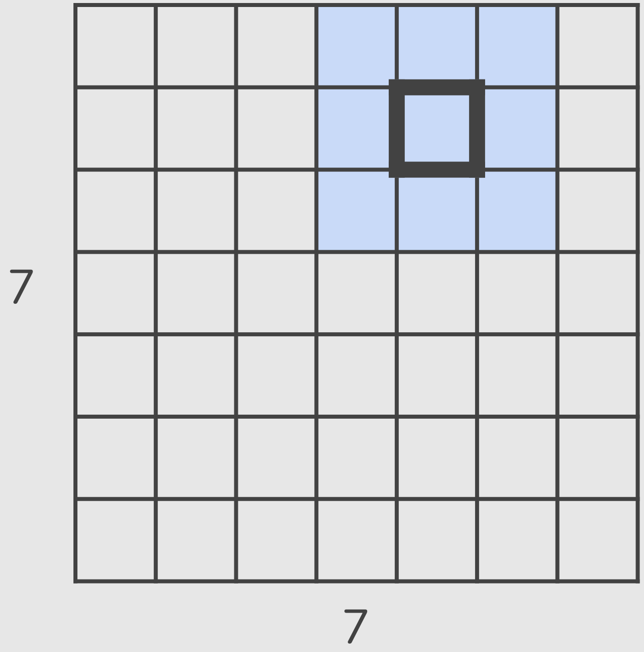
7 x 7 input (spatially)
assume 3 x 3 filter

A Closer Look at Spatial Dimensions



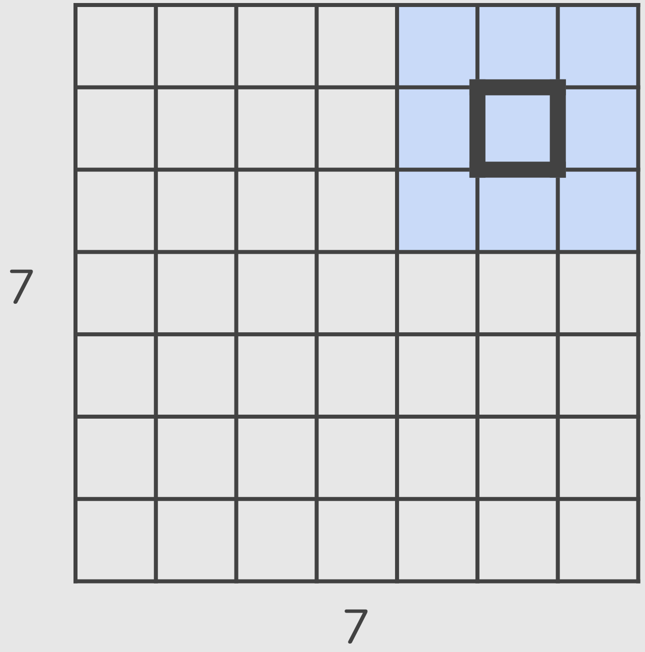
7 x 7 input (spatially)
assume 3 x 3 filter

A Closer Look at Spatial Dimensions



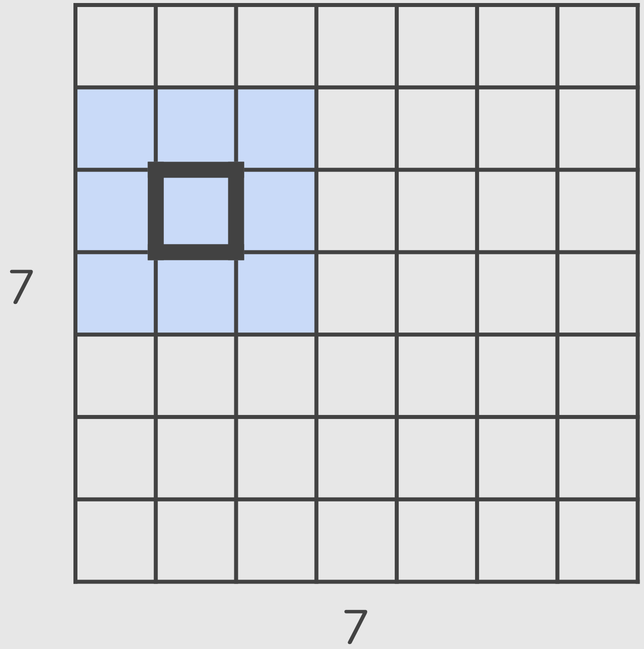
7 x 7 input (spatially)
assume 3 x 3 filter

A Closer Look at Spatial Dimensions



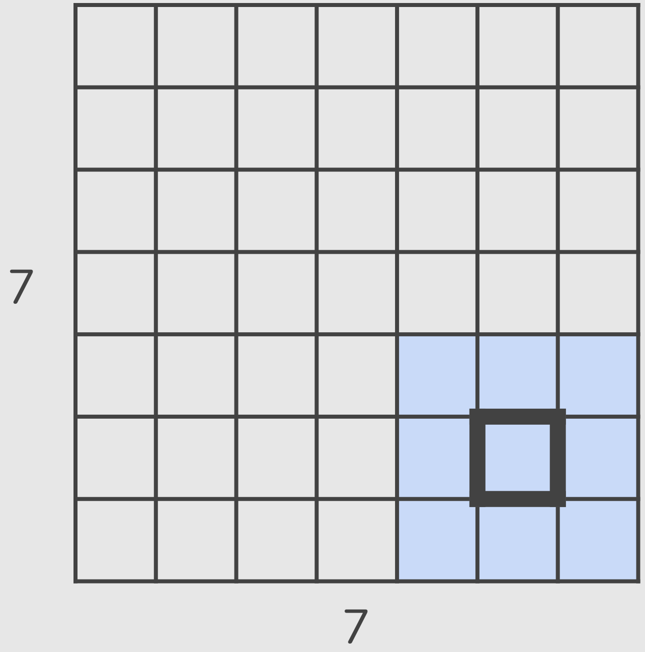
7 x 7 input (spatially)
assume 3 x 3 filter

A Closer Look at Spatial Dimensions



7 x 7 input (spatially)
assume 3 x 3 filter

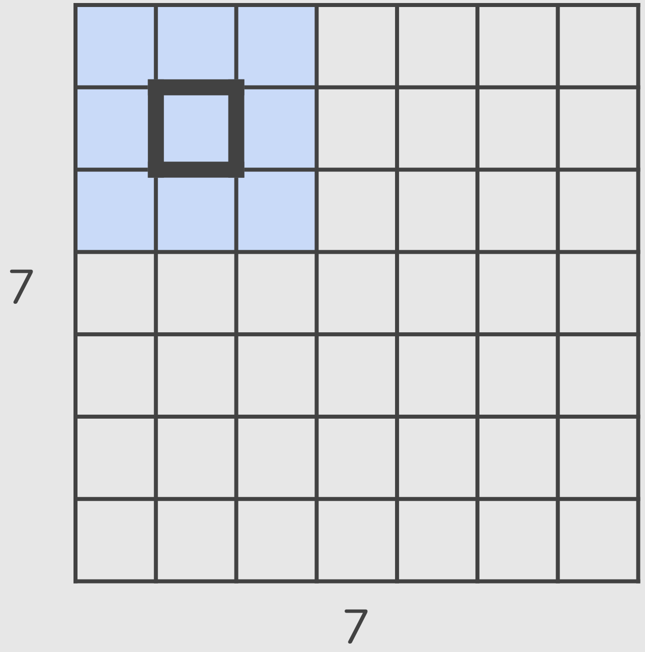
A Closer Look at Spatial Dimensions



7 x 7 input (spatially)
assume 3 x 3 filter

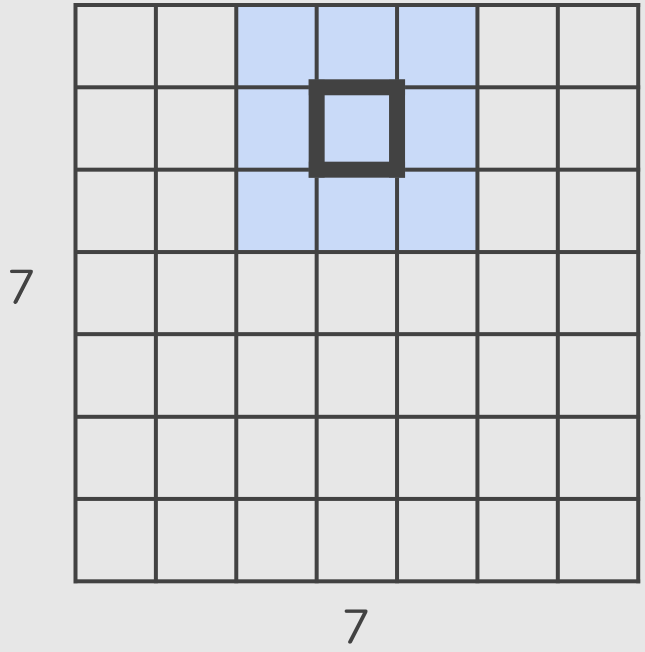
⇒ 5 x 5 output

A Closer Look at Spatial Dimensions



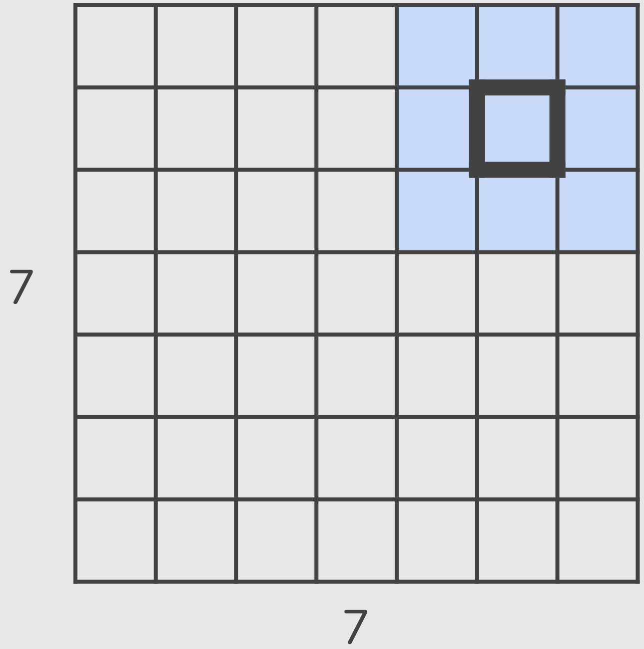
7 x 7 input (spatially)
assume 3 x 3 filter
applied with **stride 2**

A Closer Look at Spatial Dimensions



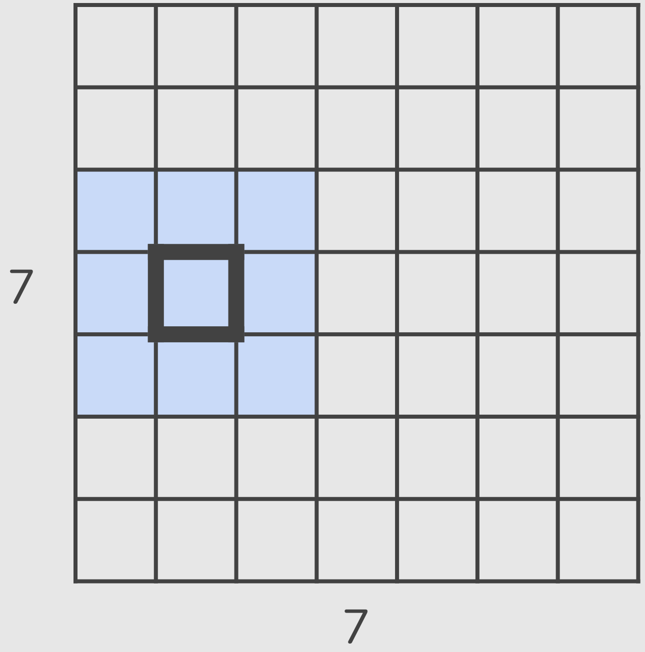
7 x 7 input (spatially)
assume 3 x 3 filter
applied with **stride 2**

A Closer Look at Spatial Dimensions



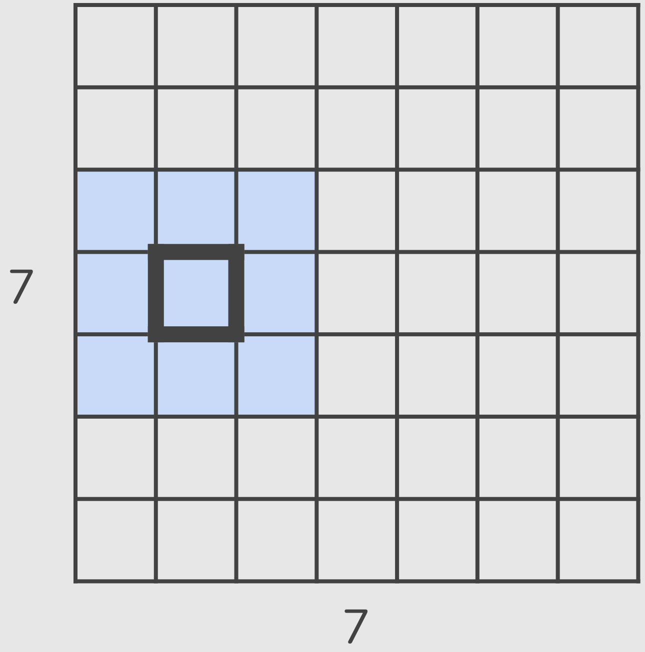
7 x 7 input (spatially)
assume 3 x 3 filter
applied with **stride 2**

A Closer Look at Spatial Dimensions



7 x 7 input (spatially)
assume 3 x 3 filter
applied with **stride 2**

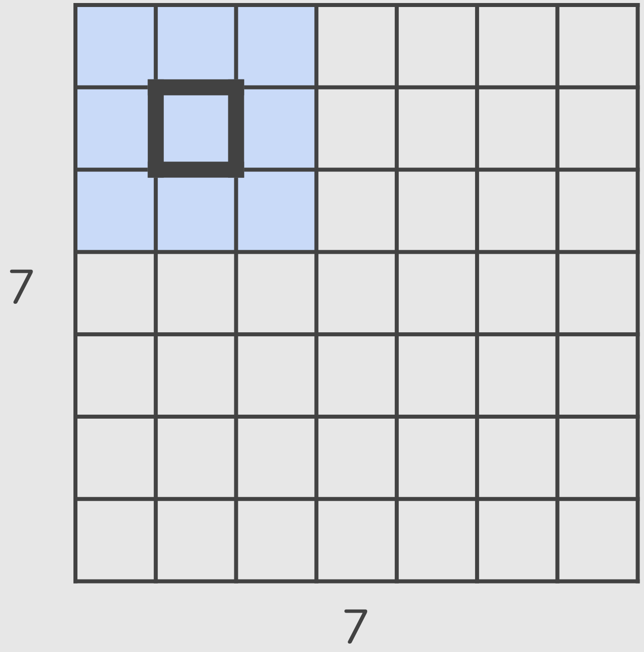
A Closer Look at Spatial Dimensions



7 x 7 input (spatially)
assume 3 x 3 filter
applied with **stride 2**

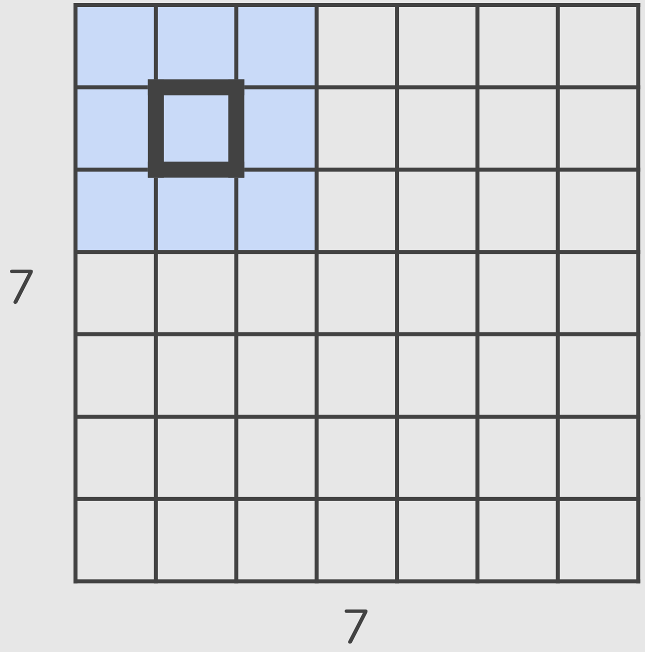
⇒ 3 x 3 output

A Closer Look at Spatial Dimensions



7 x 7 input (spatially)
assume 3 x 3 filter
applied with **stride 3?**

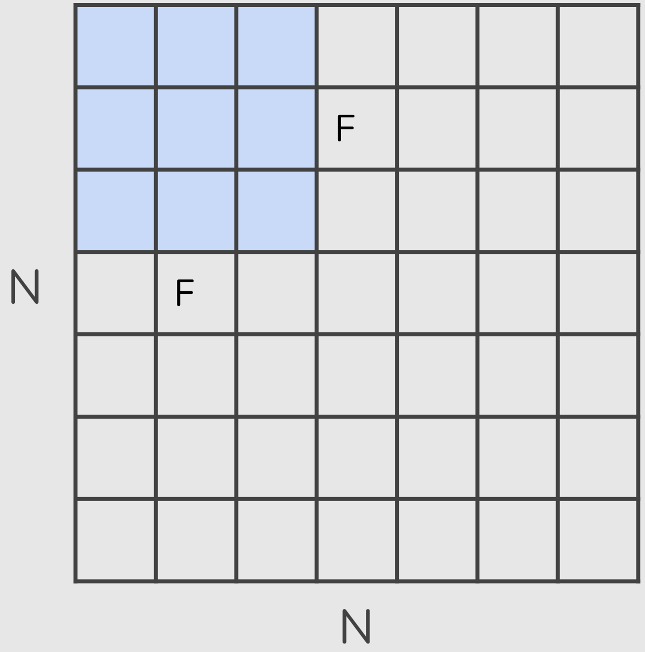
A Closer Look at Spatial Dimensions



7 x 7 input (spatially)
assume 3 x 3 filter
applied with **stride 3?**

Doesn't fit!
cannot apply 3 x 3 filter on
7 x 7 input with stride 3.

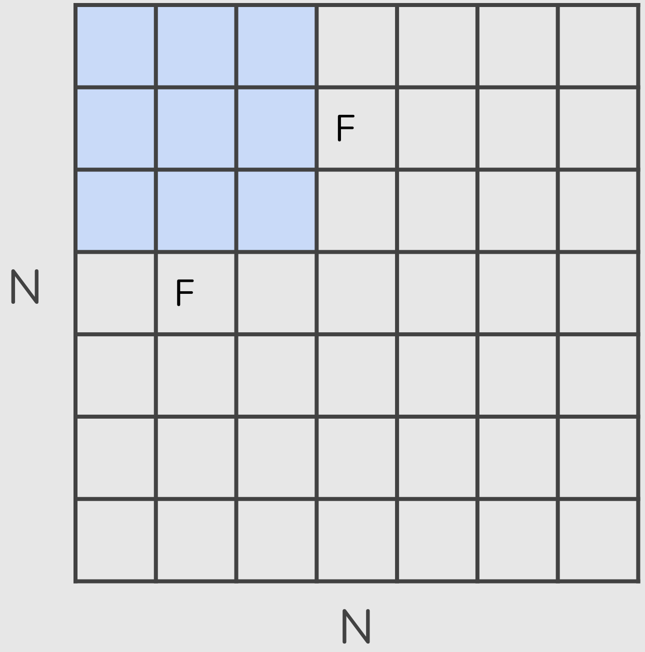
A Closer Look at Spatial Dimensions



Output size:

(N - F) / stride + 1

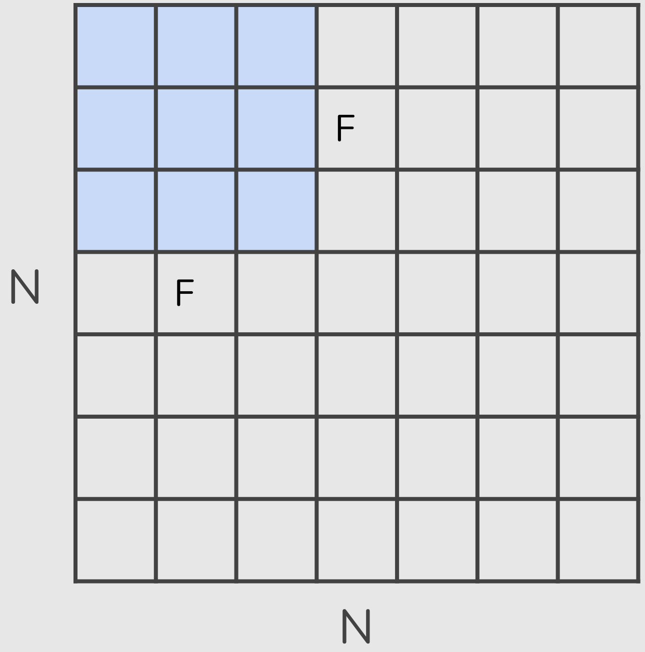
A Closer Look at Spatial Dimensions



Output size:
(N - F) / stride + 1

e.g. N = 7, F = 3:
stride 1 ⇒ (7 - 3)/1 + 1 = 5

A Closer Look at Spatial Dimensions



Output size:

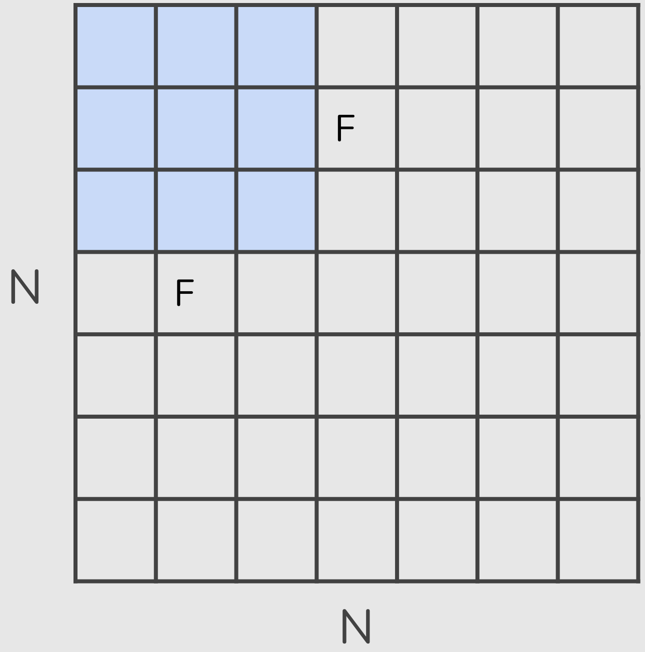
(N - F) / stride + 1

e.g. N = 7, F = 3:

stride 1 ⇒ (7 - 3)/1 + 1 = 5

stride 2 ⇒ (7 - 3)/2 + 1 = 3

A Closer Look at Spatial Dimensions



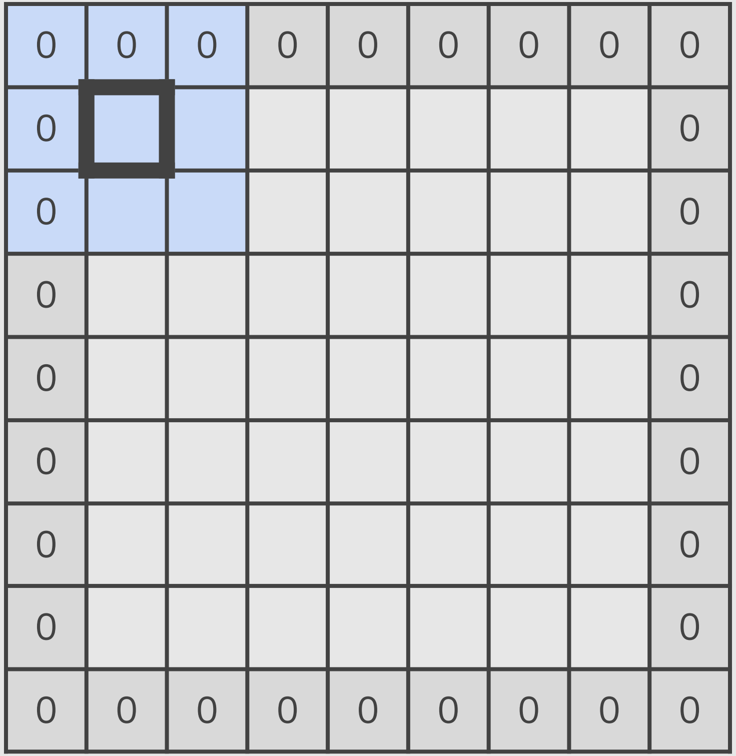
Output size:
(N - F) / stride + 1

e.g. N = 7, F = 3:
stride 1 ⇒ (7 - 3)/1 + 1 = 5
stride 2 ⇒ (7 - 3)/2 + 1 = 3
stride 3 ⇒ (7 - 3)/3 + 1 = 2.33

In Practice: Common to zero pad the border

0	0	0	0	0	0	0	0	0
0								0
0								0
0								0
0								0
0								0
0								0
0								0
0	0	0	0	0	0	0	0	0

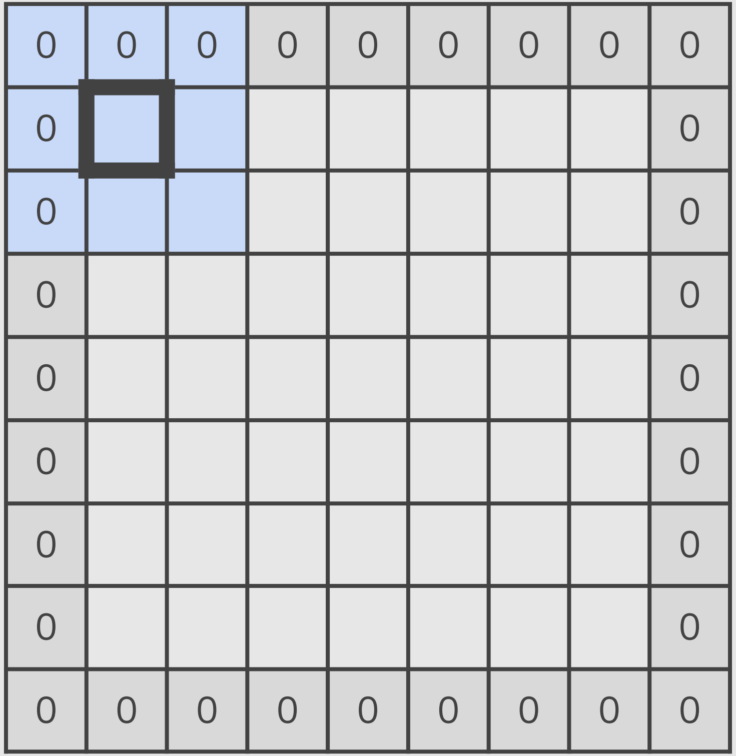
In Practice: Common to zero pad the border



7 x 7 input,
3 x 3 filter applied
with **stride 1 with pad 1**

What is the output?

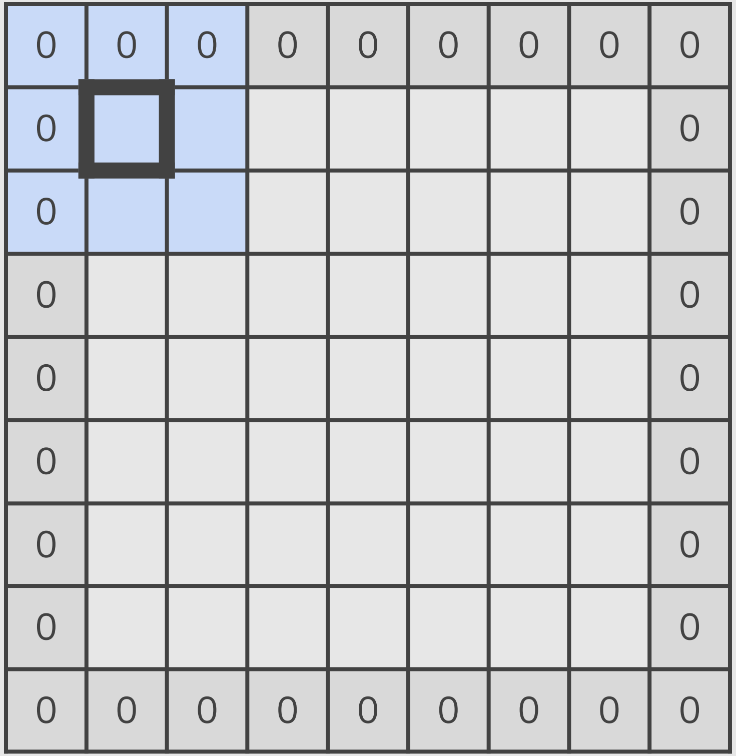
In Practice: Common to zero pad the border



7 x 7 input,
3 x 3 filter applied
with **stride 1 with pad 1**

What is the output?
7 x 7 output

In Practice: Common to zero pad the border



In general, common to see CONV layers with stride 1, filters of size F x F, and zero-padding with (F-1)/2 **(will preserve size spatially)**.

e.g. F = 3 ⇒ zero pad with 1

F = 5 ⇒ zero pad with 2

F = 7 ⇒ zero pad with 3

Padding in Keras: “valid” & “same”

“valid”: no padding.

“same”: Output size is the same as the input size.

```python
from tensorflow.keras import layers

model = tf.keras.Sequential()
#Camada convolucional com 10 filtros de tamanho 3x3 e ativação ReLU
model.add(layers.Conv2D(10, 3, padding='valid', activation='relu', input_shape=(28,28,1)))




model.summary()
```

MNIST 28 x 28

```python
from tensorflow.keras import layers

model = tf.keras.Sequential()
#Camada convolucional com 10 filtros de tamanho 3x3 e ativação ReLU
model.add(layers.Conv2D(10, 3, padding='valid', activation='relu', input_shape=(28,28,1)))

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 26, 26, 10)        100
```

MNIST 28 x 28

Number of Parameters

Input volume: **32 x 32 x 3**

10 5 x 5 filters with stride 1, pad 2

Number of parameters in this layer?

Number of Parameters

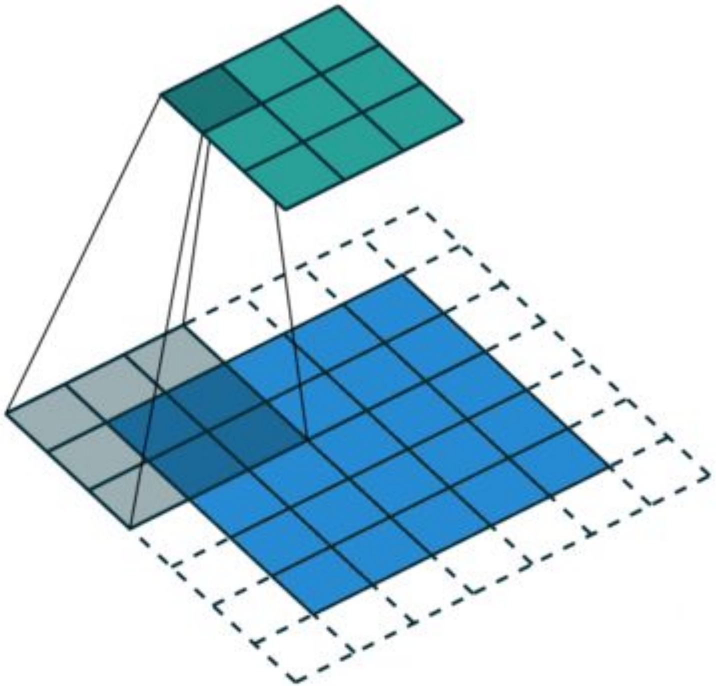
Input volume: **32** x **32** x **3**

10 **5** x **5** filters with stride 1, pad 2

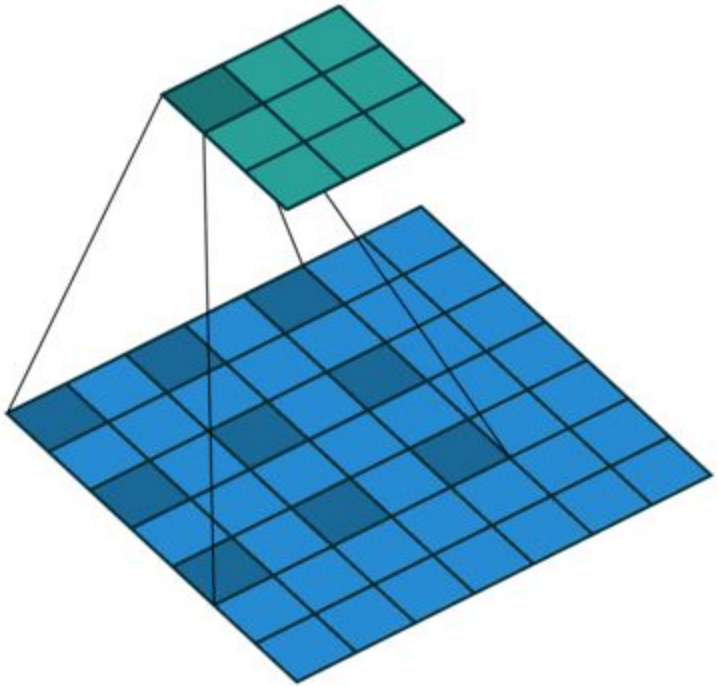
Number of parameters in this layer?

Each filter has 5*5*3 + 1 = **76** parameters (+1 for bias)

=> **76***10** = 760



Standard Convolution



Dilated Convolution

Convolutions

"A Guide to Convolution Arithmetic for Deep Learning"
https://arxiv.org/pdf/1603.07285.pdf (Jan. 2018)

"Convolution animations" https://github.com/vdumoulin/conv_arithmetic

"A Comprehensive Introduction to Different Types of Convolutions in Deep Learning" (Jan. 2019)
https://towardsdatascience.com/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215

Convolutions Layers

https://keras.io/api/layers/convolution_layers

K Keras

Search Keras documentation...

About Keras

Getting started

Developer guides

Keras API reference

Models API

Layers API

Callbacks API

Data preprocessing

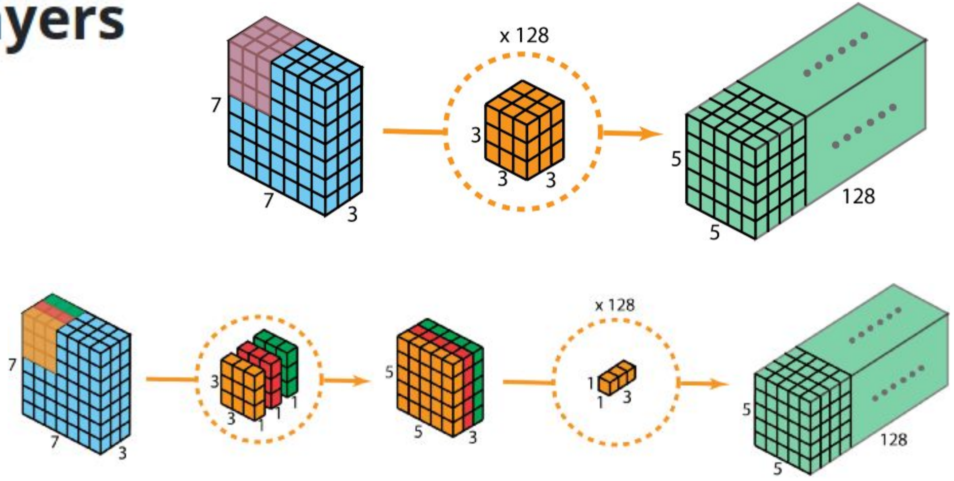
Optimizers

Metrics

» Keras API reference / Layers API / Convolution layers

Convolution layers

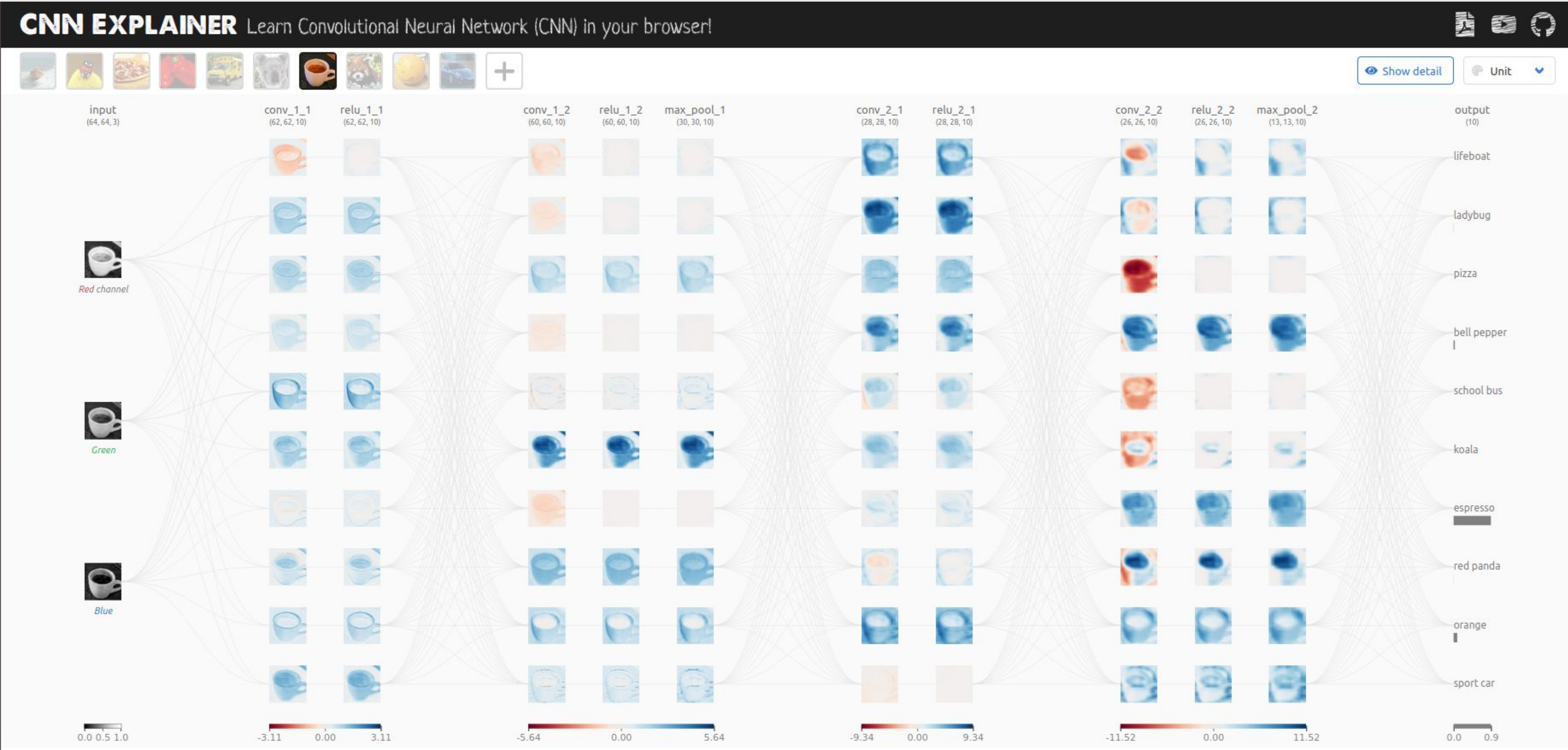
- Conv1D layer
- Conv2D layer
- Conv3D layer
- SeparableConv1D layer
- SeparableConv2D layer
- DepthwiseConv2D layer
- Conv2DTranspose layer
- Conv3DTranspose layer



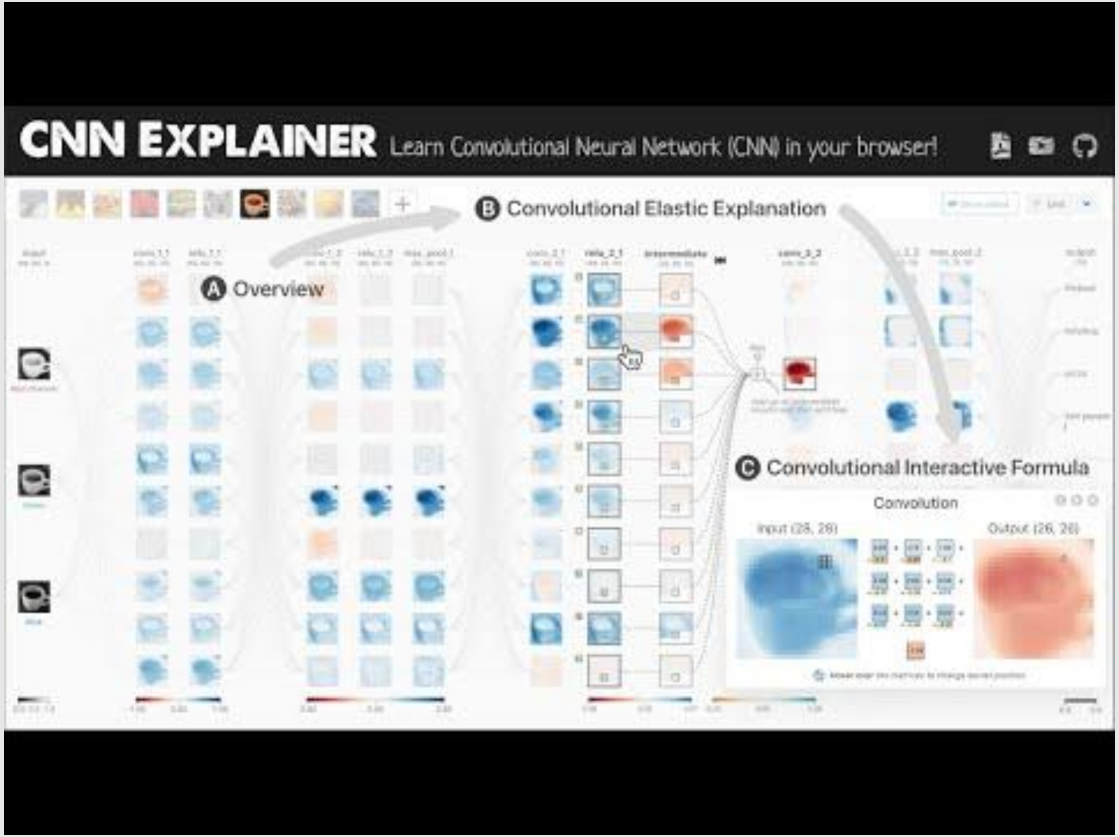
Today’s Agenda

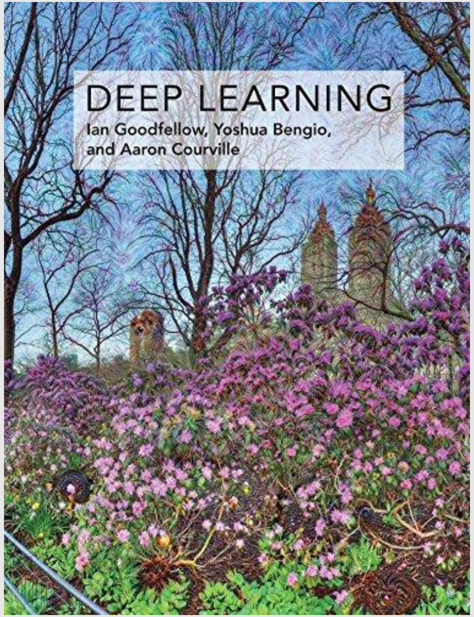
- What is Deep Learning?
- Deep Learning & Applications
- Neural Networks *vs.* Convolutional Networks
- What is a convolution?
- Convolutional Neural Networks
    - Convolution Layer

https://poloclub.github.io/cnn-explainer



https://youtu.be/HnWIHWFbuUQ (3 min)





"Deep Learning", Goodfellow & Bengio & Courville, 2016.

http://www.deeplearningbook.org/contents/convnets.html

Chapter 9

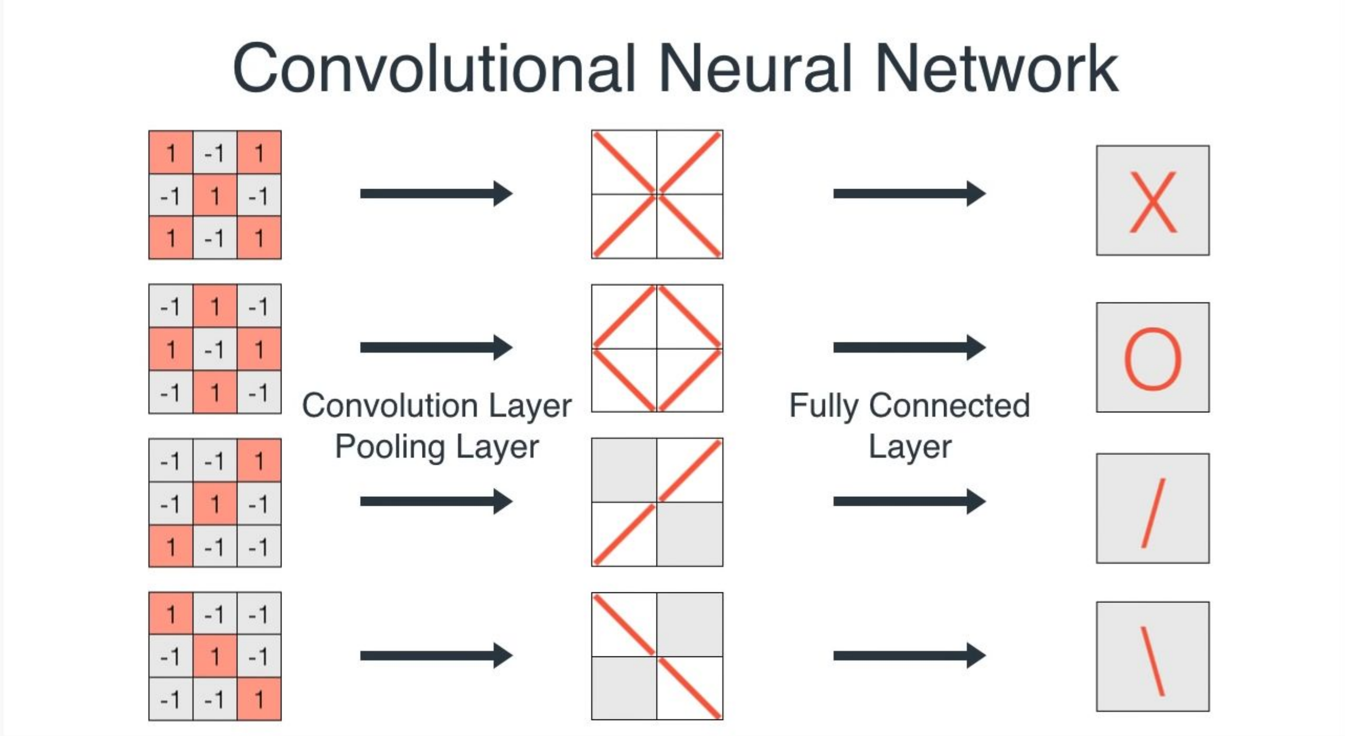
Convolutional Networks

Convolutional networks (LeCun, 1989), also known as **convolutional neural networks**, or CNNs, are a specialized kind of neural network for processing data that has a known grid-like topology. Examples include time-series data, which can be thought of as a 1-D grid taking samples at regular time intervals, and image data, which can be thought of as a 2-D grid of pixels. Convolutional networks have been tremendously successful in practical applications. The name "convolutional neural network" indicates that the network employs a mathematical operation called **convolution**. Convolution is a specialized kind of linear operation. *Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.*

In this chapter, we first describe what convolution is. Next, we explain the motivation behind using convolution in a neural network. We then describe an operation called **pooling**, which almost all convolutional networks employ. Usually, the operation used in a convolutional neural network does not correspond precisely to the definition of convolution as used in other fields, such as engineering or pure mathematics. We describe several variants on the convolution function that are widely used in practice for neural networks. We also show how convolution may be applied to many kinds of data, with different numbers of dimensions. We then discuss means of making convolution more efficient. Convolutional networks stand out as an example of neuroscientific principles influencing deep learning. We discuss these neuroscientific principles, then conclude with comments about the role convolutional networks have played in the history of deep learning. One topic this chapter does not address is how to choose the architecture of your convolutional network. The goal of this chapter is to describe the kinds of tools that convolutional networks provide, while chapter 11 describes general guidelines

326

“A friendly introduction to Convolutional Neural Networks and Image Recognition” https://youtu.be/2-Ol7ZB0MmU



A friendly introduction to Convolutional Neural Networks and Image Recognition

References

Machine/Deep Learning Books

- “Hands-On Machine Learning with Scikit-Learn and TensorFlow” (2016), **Convolutional Neural Networks,** Chap. 13
- “Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow” (2019), **Deep Computer Vision Using Convolutional Neural Networks**, Chap. 14
- “Deep Learning with PyTorch” (2020), **Using convolutions to generalize**, Chap. 8

Deep Learning Courses

- “Convolutional Neural Networks for Visual Recognition” (2017), **Convolutional Neural Networks**, Fei-Fei Li & others https://youtu.be/bNb2fEVKeEo (70 min)
- Deeplearning.ai (2017), “Convolutional Neural Networks” (11 videos, 5-16min) https://www.youtube.com/playlist?list=PLkDaE6sCZn6Gl29AoE31iwdVwSG-KnDzF