Deep Learning

Big Data & Machine Learning Bootcamp - Keep Coding

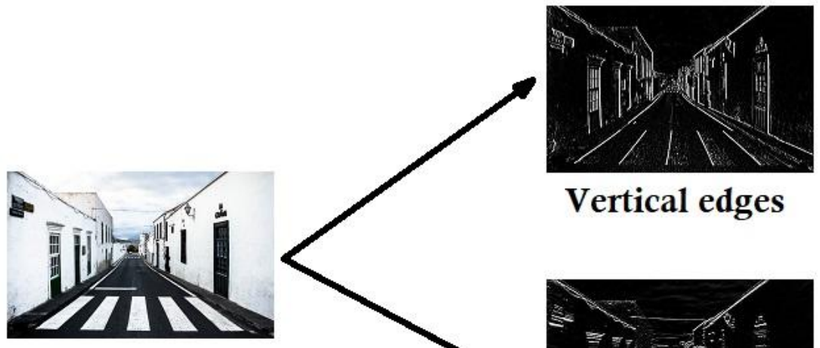
Outline

1. Edge detection
2. Padding
3. Strided convolutions
4. One layer of a convolutional network
5. Simple convolutional network
6. Pooling layers
7. Convolutional neural network (CNN) example
8. Why convolutions?

Edge detection

We've previously talked about how the neural networks extract different features at different layers.

But how the CNN detect edges?



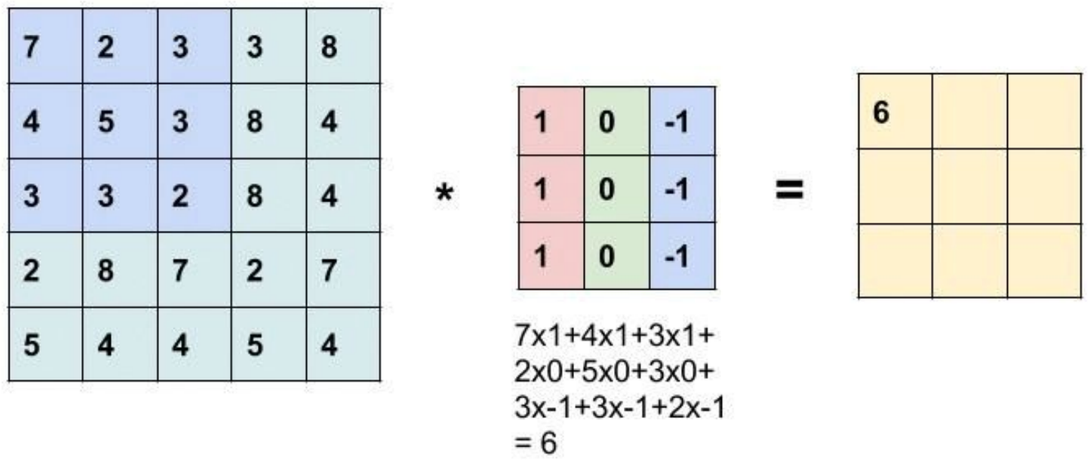
Vertical edges

Horizontal edges

Sources:
- Coursera
- http://datahacker.rs/edge-detection/

Edge detection

First, how convolution works:



Sources:
- Coursera
- https://medium.com/datadriveninvestor/convolutional-neural-networks-3b241a5da51e

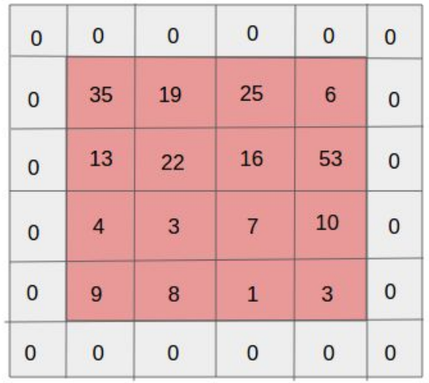
Padding

Padding is a modification made to the basic convolutional operation that solves two problems:

- Shrinking output
- Throwing away information from edge

This modification allows us to have **same size of image after convolution** and also **process more than one time the edge pixels** by just adding zeros around the input image.

In this image padding = 1



Sources:
- Coursera
- https://blog.xrds.acm.org/2016/06/convolutional-neural-networks-cnns-illustrated-explanation/

Padding

There are two common choices on how much to pad:

- **Valid convolutions** -> NO PADDING.
  The output image will be (n - f +1) x (n - f + 1) where n is the size on the input image and f the size of the filter/kernel

- **Same convolution** -> Pad so that output size is the same as the input size
  Padding = (filter size - 1) / 2

  As as side note, the size of the filter is usually odd! i.e. 3, 5, 7

Sources:
- Coursera

Strided convolutions

Stride is another modification of the basic convolution operation.

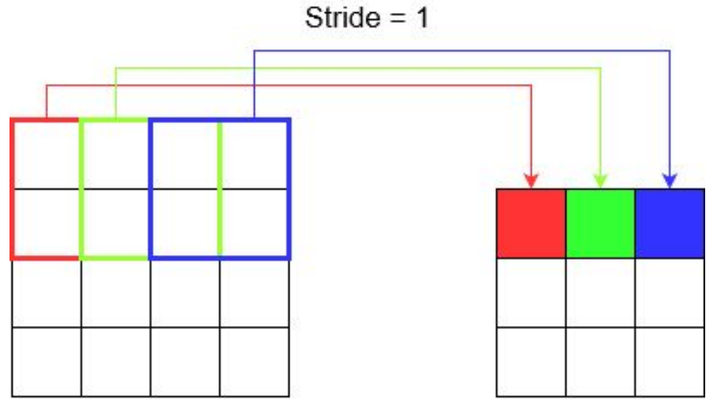
Basically, stride is the **number of rows and columns we traverse per slide** when doing the convolution.

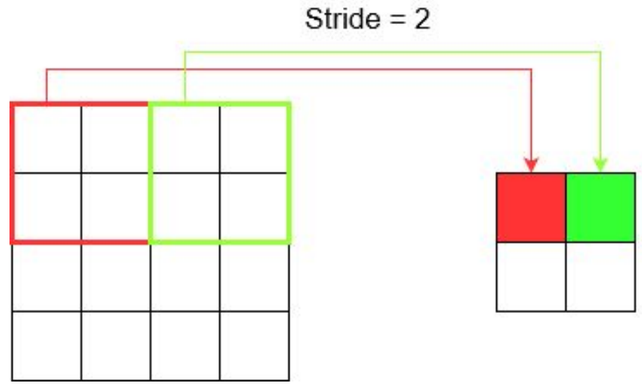
The output width and height are specified by these equations:

$$\text{output width} = \frac{W - F_w + 2P}{S_w} + 1$$

$$\text{output height} = \frac{H - F_h + 2P}{S_h} + 1$$

When the output values are not integers, we take the floor operation. This means we round the values down





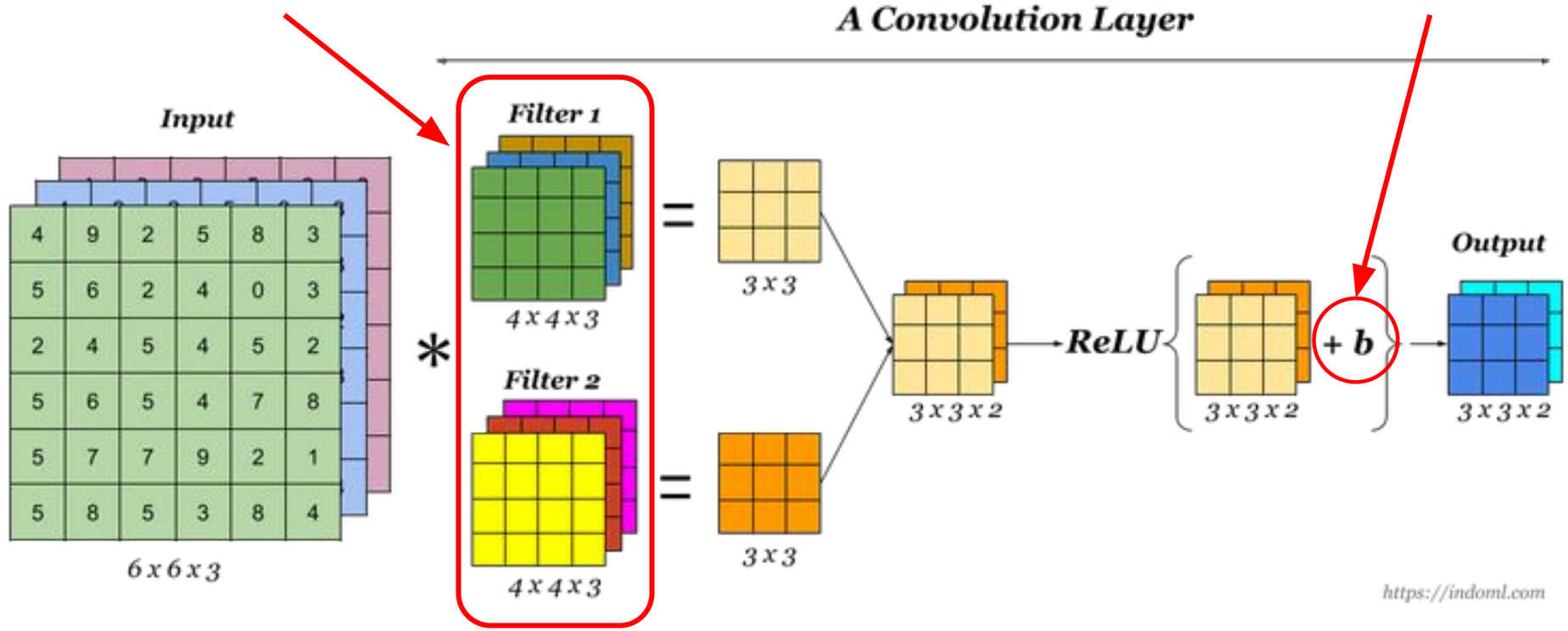
Sources:
- Coursera
- https://stackoverflow.com/questions/42883547/intuitive-understanding-of-1d-2d-and-3d-convolutions-in-convolutional-neural-n

5. One layer of a convolutional network

Convolution is a linear operation

This play the role of W
Remember the linear equation?

We add the bias before the activation function!

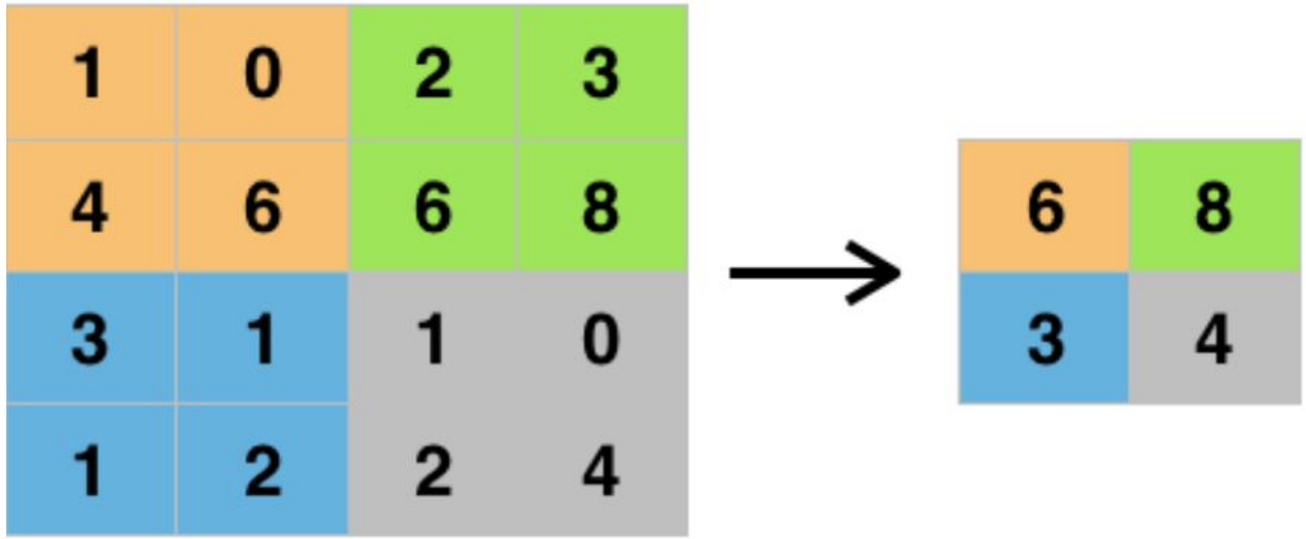


Sources:
- Coursera
- https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/

Pooling layers

Pooling layers are used to **reduce the size of the representation**, to speed up the computation and to make the features detected a bit more robust.

There are different types of pooling. One of the most commonly used is the max pooling:



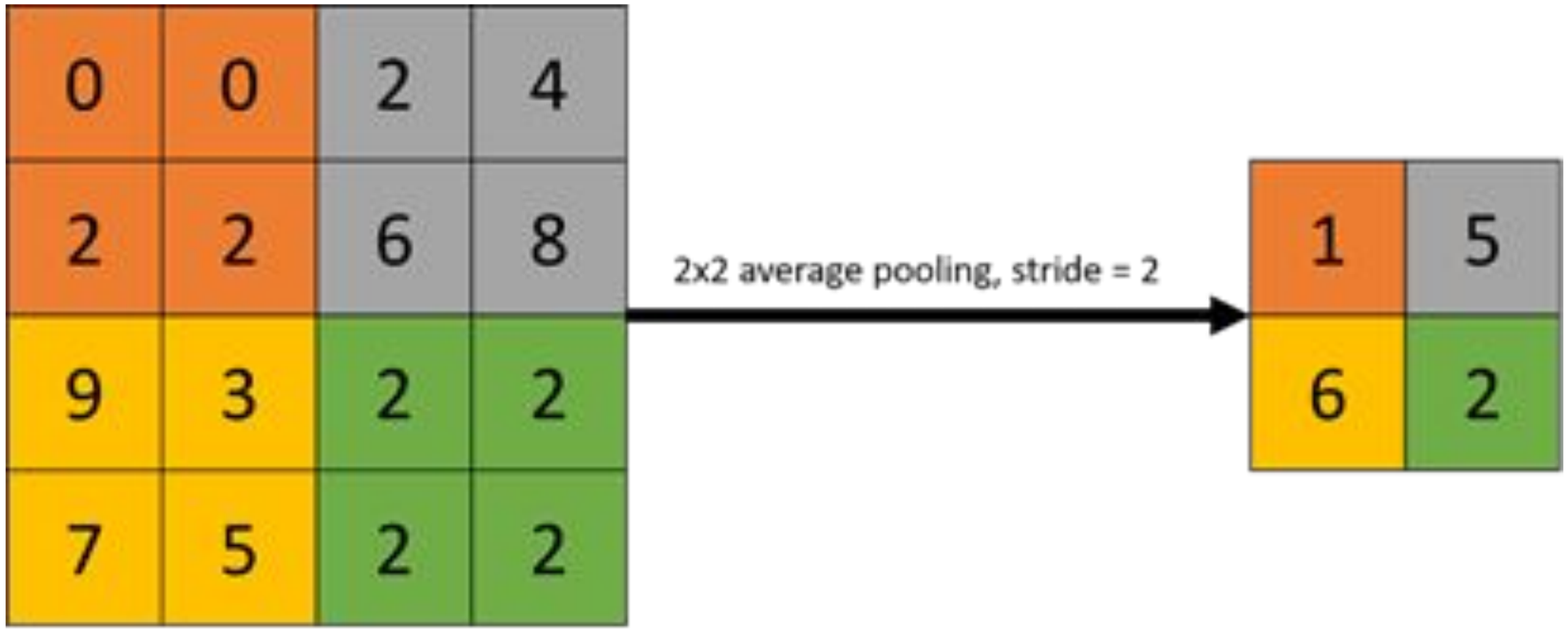
Hyperparameters:
Filter size = 2
Stride = 2

- Pooling **doesn't have parameters to learn**
- Pooling is applied to a single slice and it doesn't change the number of slices

Sources:
- Coursera
- https://deepai.org/machine-learning-glossary-and-terms/max-pooling

Pooling layers

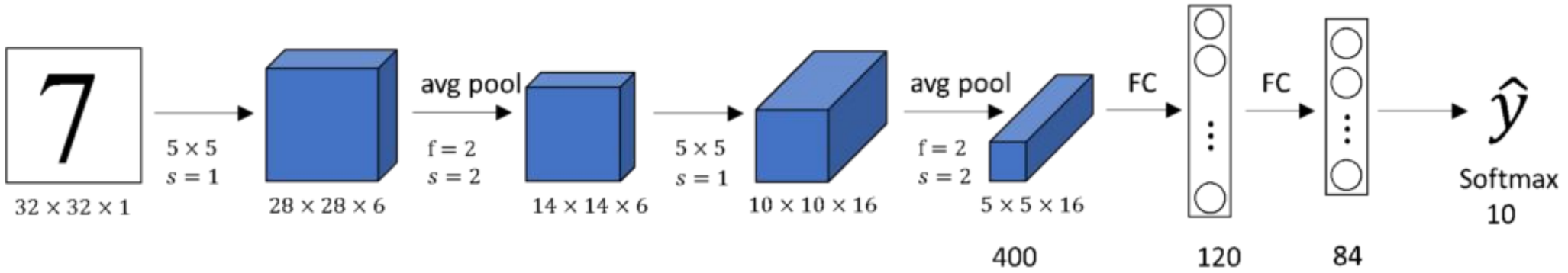
There is a less used type of pooling called **average pooling**:



Sources:
- Coursera
- https://embarc.org/embarc_mli/doc/build/html/MLI_kernels/pooling_avg.html

CNN example

Let's count the number of parameters



	Activation Shape	Activation Size	# Parameters
Input	(32,32,1)	1.024	0
CONV1 (f=5, s=1)	(28,28,6)	4.704	156
POOL1	(14,14,6)	1.176	0
CONV2 (f=5, s=1)	(10,10,16)	1.600	2416
POOL2	(5,5,16)	400	0
FC	(120,1)	120	48.120
FC	(84,1)	84	10.164
FC + Softmax	(10,1)	10	850

How can we compute this value???

You can see that most of the parameters come from the FC layers

Sources:
- Coursera

Why convolutions?

Parameters sharing: A feature detector (such as a vertical edge detector) that's useful in one part of the image is probably useful in another part of the image.

Sparsity of connections: In each layer, each output value depends only on a small number of input.

Translation invariance: A cat that is shifted or mirrored it is still cat in convolution neural networks

AND we still can use gradient descent to train it

Sources:
- Coursera