Mathematical techniques in data science

Lecture 13: Convolutional neural networks (cont.)

Reminders

- Office hours:
  - MW 3pm-4pm, Ewing Hall 312
  - By appointments
- Homework 1: due today EOD

Computer vision

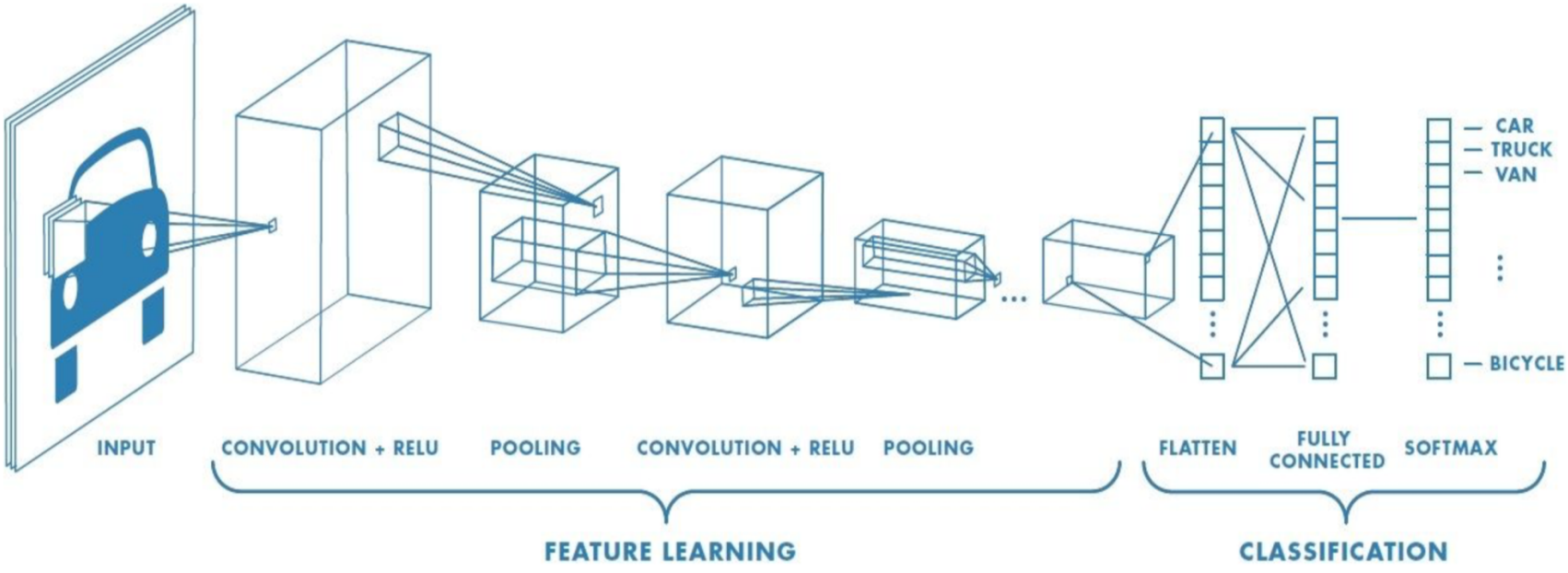
A field that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs

- Image classification/object recognition
- Object detection
- Image segmentation
- Image generation
- Image style transfer

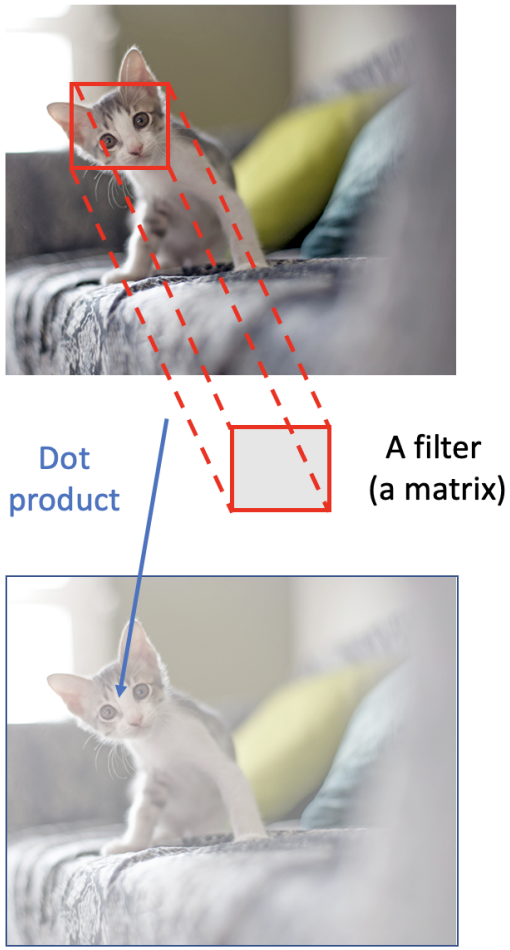
Feed-forward neural networks

- Structure:
  - Graphical representation
  - Activation functions
  - Loss functions
- Issues
  - Flat vectors lose spatial information
  - Sensitive to location of the object
  - Cannot capture small regions within an image
  - Redundant parameters

Convolutional neural networks

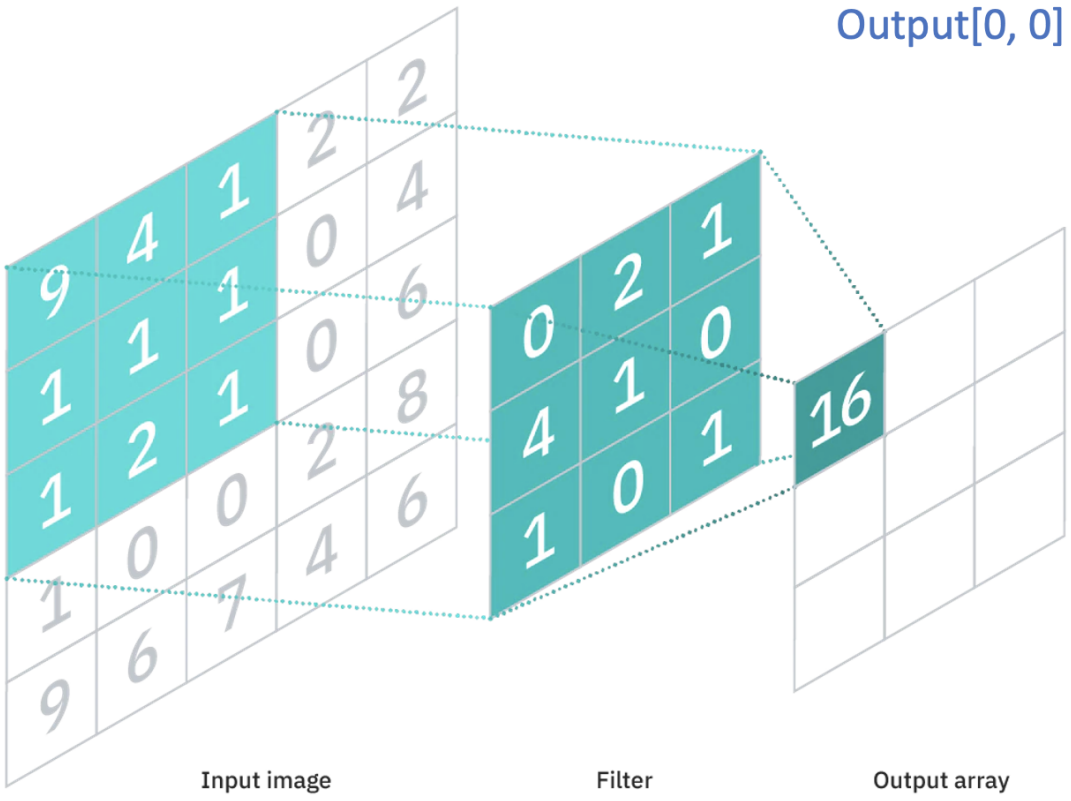


Convolutional layer



- Do not flatten the input image
- Apply a filter (kernel) to each local region of the image
- Slide the filter through all spatial locations to get the output

Applying a kernel



Output[0, 0] = (9*0) + (4*2) + (1*1) +
(1*4) + (1*1) + (1*0) +
(1*1) + (2*0) + (1*1)
= 16

(Adapted from IBM)

Examples of kernels

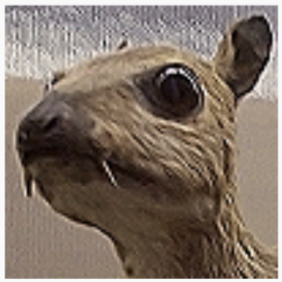
Edge detection

Kernel

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

Sharpen

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



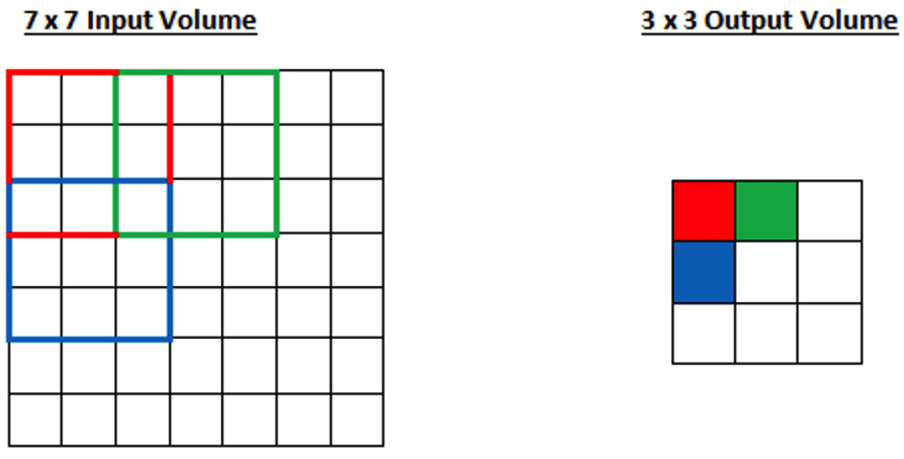
Feature extraction

- Historically, convolutional filters have been used to extract image features
- CNNs automate that process by considering the entries of the filters as model parameters

Stride

- Number of pixels to shift the filter
- Can be different for each dimension

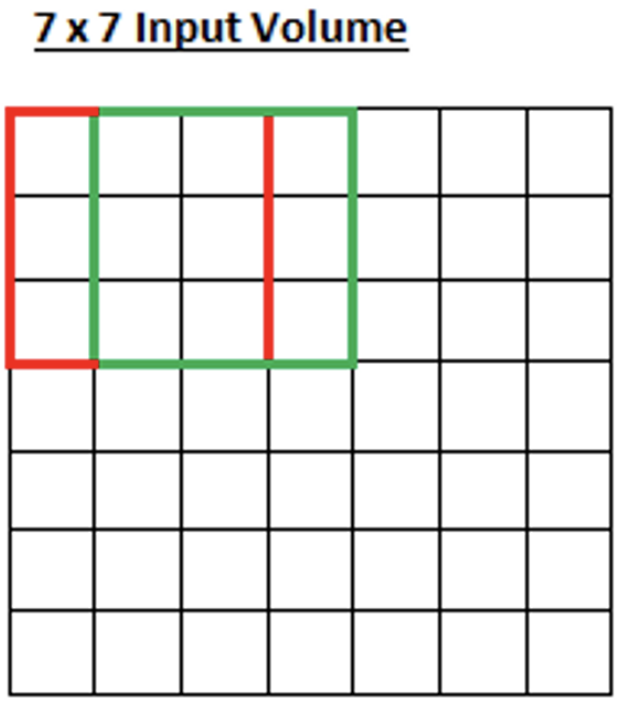
strides = (2, 2)

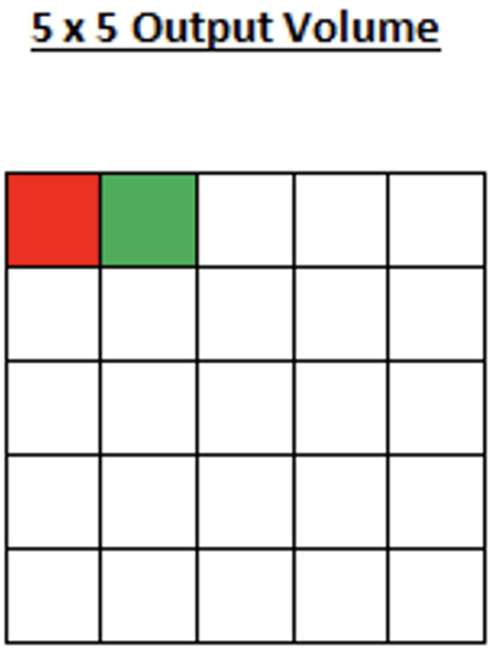


(Source: Adit Deshpande)

Padding

2 possible settings: Valid or Same

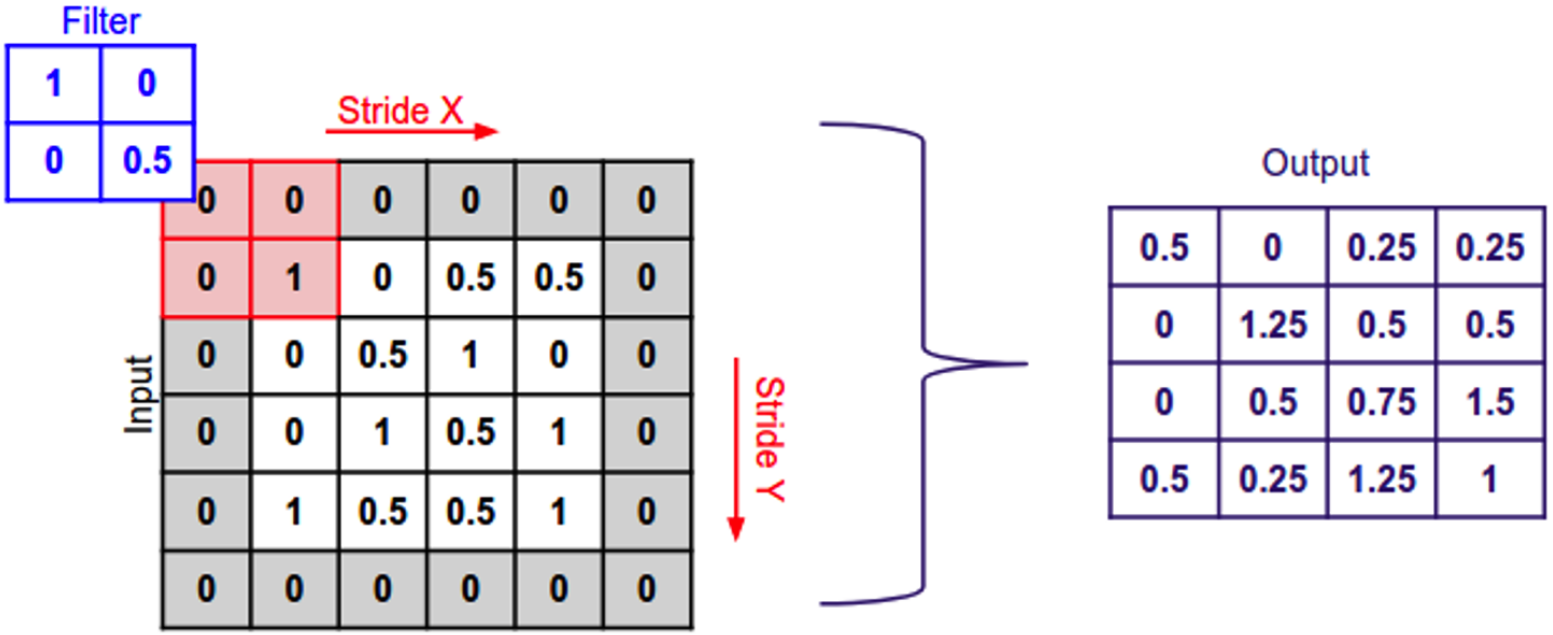




(Valid)

Padding

Same padding: add 0 pixels to boundary of input image to get similar output shape



(Same)

Applying a kernel

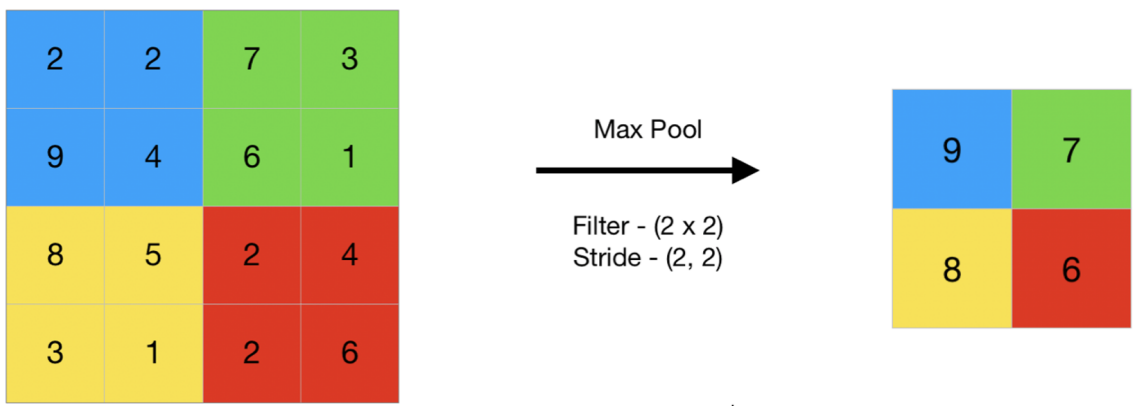
https://vucdinh.github.io/Presentation/figures/cnn/filter-run.gif

Pooling layer

- Down-sample the input image along its spatial dimensions
- Common types: max pooling and average pooling

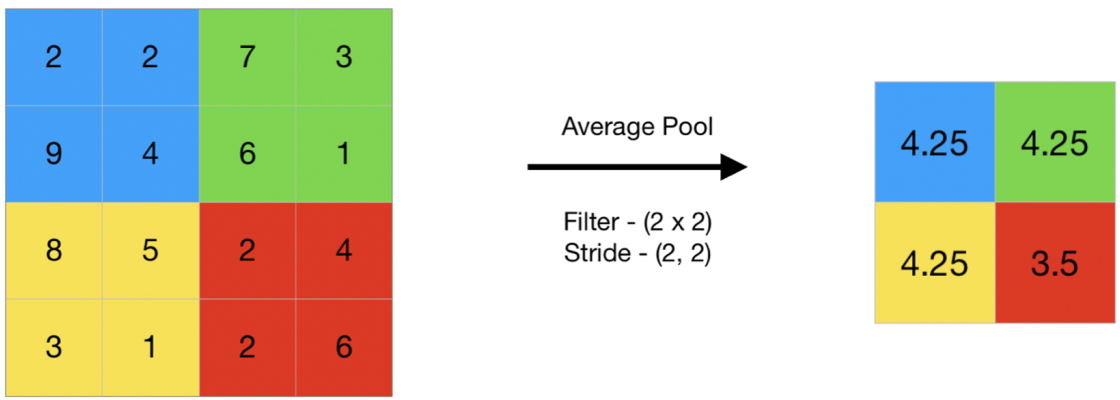
Max pooling

- Return max value when applying the filter
- Default strides = filter size

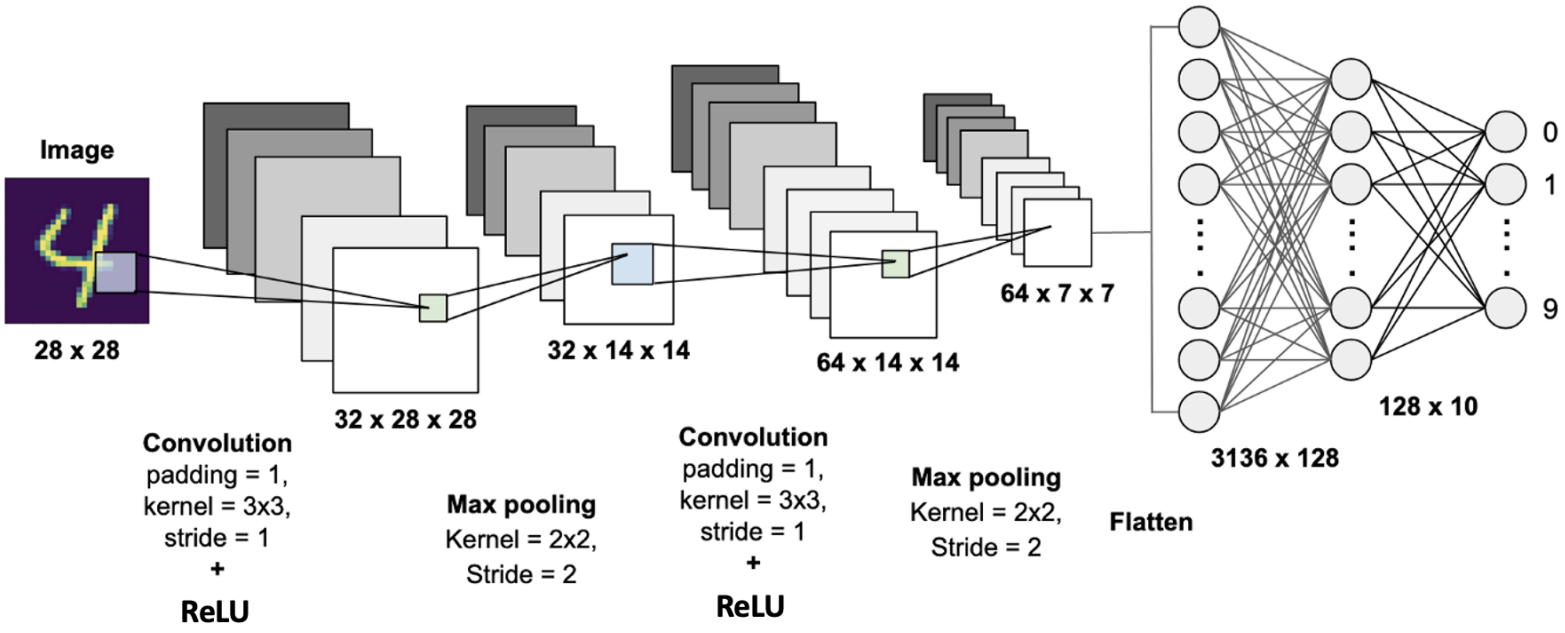


Average pooling

- Return average value when applying the filter
- Default strides = filter size



Example of a complete CNN



Convolutional layer on Keras

Conv2D class

```
tf.keras.layers.Conv2D(
    filters,
    kernel_size,
    strides=(1, 1),
    padding="valid",
    data_format=None,
    dilation_rate=(1, 1),
    groups=1,
    activation=None,
    use_bias=True,
    kernel_initializer="glorot_uniform",
    bias_initializer="zeros",
    kernel_regularizer=None,
    bias_regularizer=None,
    activity_regularizer=None,
    kernel_constraint=None,
    bias_constraint=None,
    **kwargs
)
```

Pooling layer on Keras

MaxPooling2D class

```
tf.keras.layers.MaxPooling2D(
    pool_size=(2, 2), strides=None, padding="valid", data_format=None, **kwargs
)
```

AveragePooling2D class

```
tf.keras.layers.AveragePooling2D(
    pool_size=(2, 2), strides=None, padding="valid", data_format=None, **kwargs
)
```

CIFAR10 dataset

- Low-resolution color images of size 32 x 32
- 10 classes

Demo: train a CNN using Keras