LESSON 10

Similarity and Deep Learning (with coding)

Exploratory Data Analysis

Feature engineering

Exam Simulation

NOTE

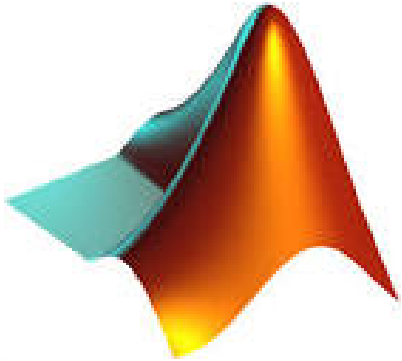
- **Since are going to reach the mileston of ½ of the course, we will do a simulation (lesson 13 or 14) of the exam to check your preparation and your method for studying**

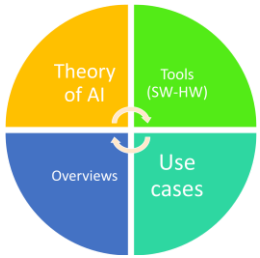
Lesson 11: Outline

- Similarity as a tool for deep learning
- Exploratory Data Analysis
- Feature engineering
  - Data preprocessing:
    a statistical point of view
- Application of EDA to NN
  - Inputs
  - Weights
  - Activations
  - Outputs
- **Main points**

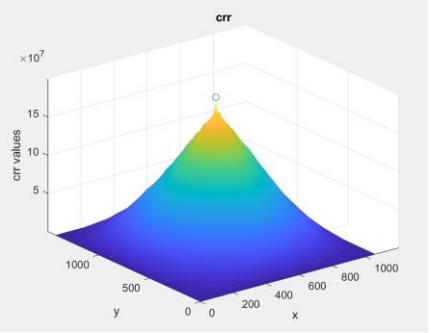
Toolboxes
Matlab

Homework solutions…

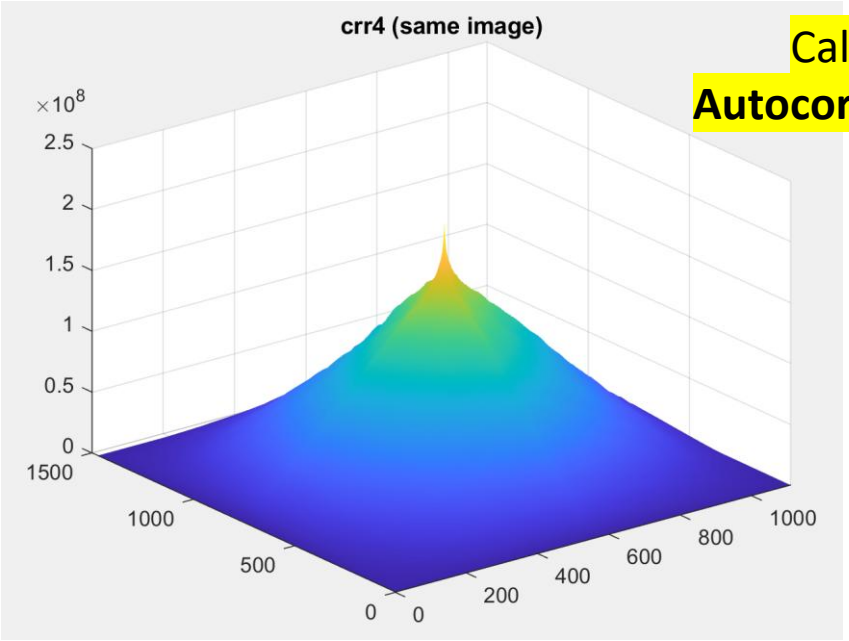




Homework #1



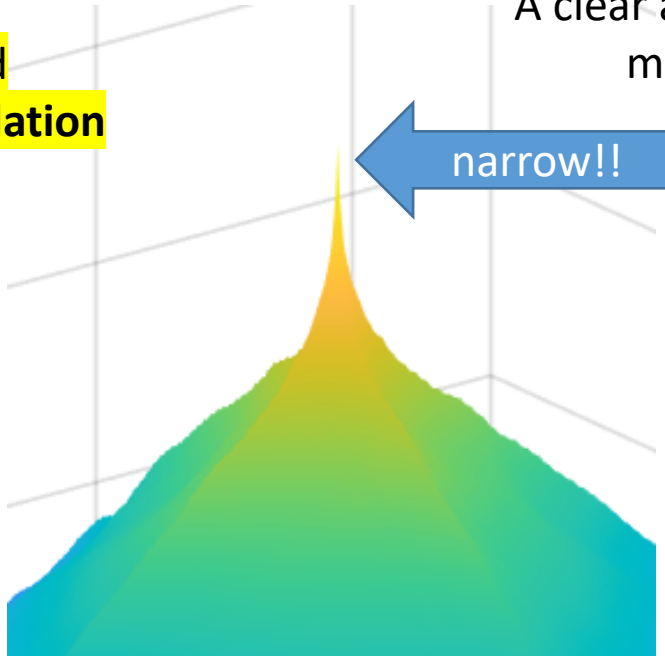
- How is changing the CRR if you put in input the same template image as subimage CRR(x,x) ?
  - What about the maximum?
  - What about the shape of the CRR?



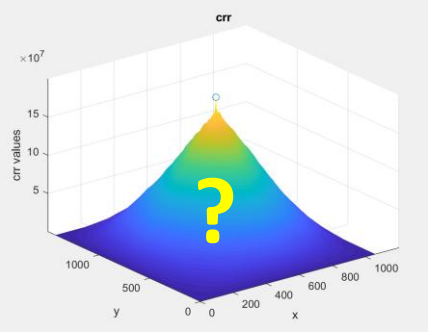
Called
Autocorrelation

A clear and separated maximum!

narrow!!



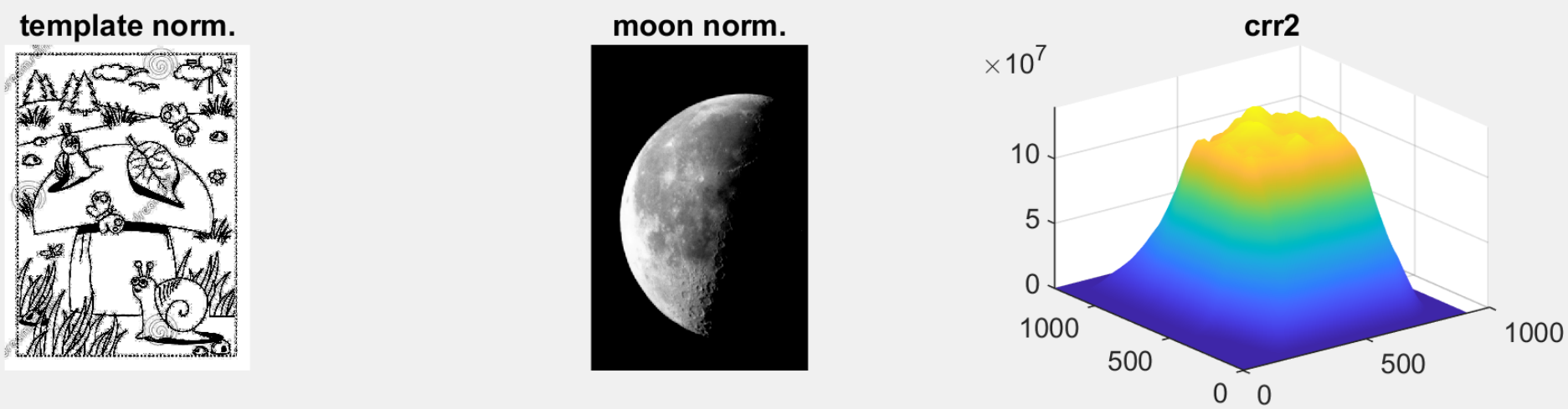
Homework #2



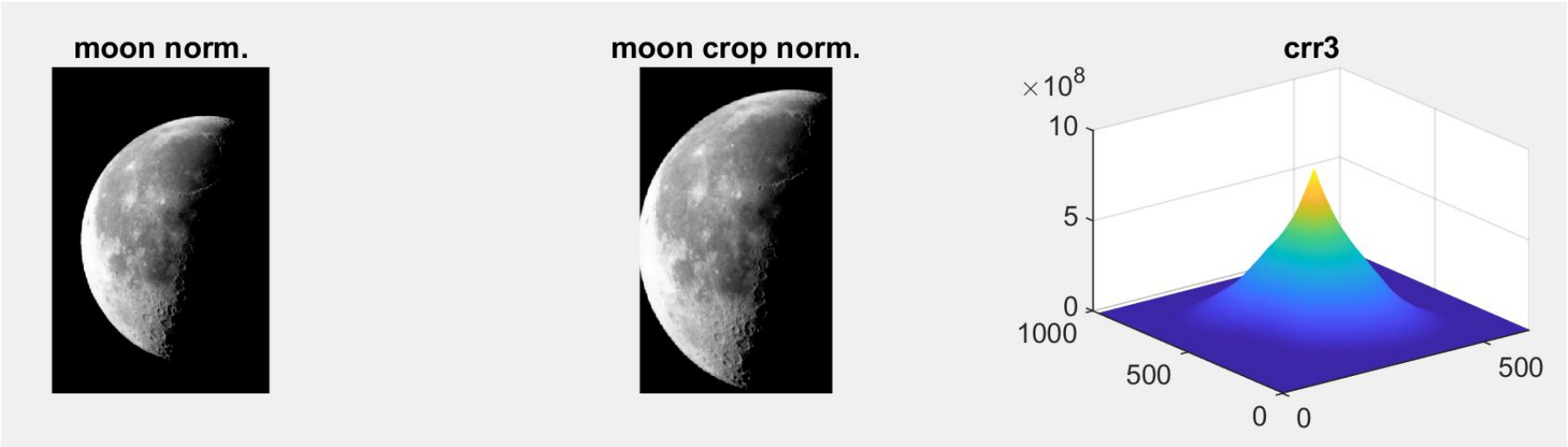
- How is changing the CRR if you put two completely different images in input?
  - What about the maximum?
  - What about the shape of the CRR?

```
%% comparison
% the moon image is builtin in matlab
img_moon_gray = imread('moon.tif');
img_moon_gray_norm = img_moon_gray - mean(mean(img_moon_gray)) ;
img_template_gray_norm = img_template_gray - mean(mean(img_template_gray));

% xcorr2(A,B) computes the crosscorrelation of matrices A and B.
crr2 = xcorr2(double(img_template_gray_norm) , double(img_moon_gray_norm) );
```

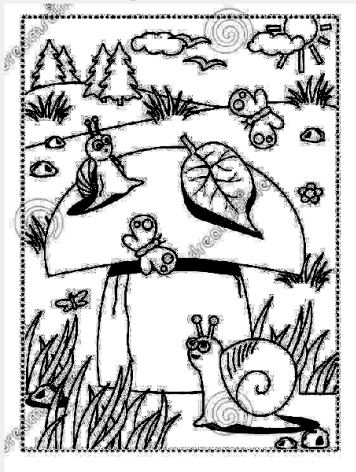
```
h3 = figure;
subplot(2,3,1);
imshow(img_template_gray_norm, []);
title('template norm.')

subplot(2,3,2);
imshow(img_moon_gray_norm, []);
title('moon norm.')

subplot(2,3,3);
surf(crr2);
shading interp;   % just to remove black line around the tiles
title('crr2')
```



```
% xcorr2(A,B) computes the crosscorrelation of matrices A and B.
img_crop = img_moon_gray_norm( [(1+50):(end-50)], [(1+50):(end-50)]) ;

crr3 = xcorr2( double( img_moon_gray_norm) , double(img_crop)  ) ;


subplot(2,3,4);
imshow(img_moon_gray_norm, []);
title('moon norm.')

subplot(2,3,5);
imshow(img_crop, []);
title('moon crop norm.')

subplot(2,3,6);
surf(crr3);
shading interp;   % just to remove black line around the tiles
title('crr3')
```

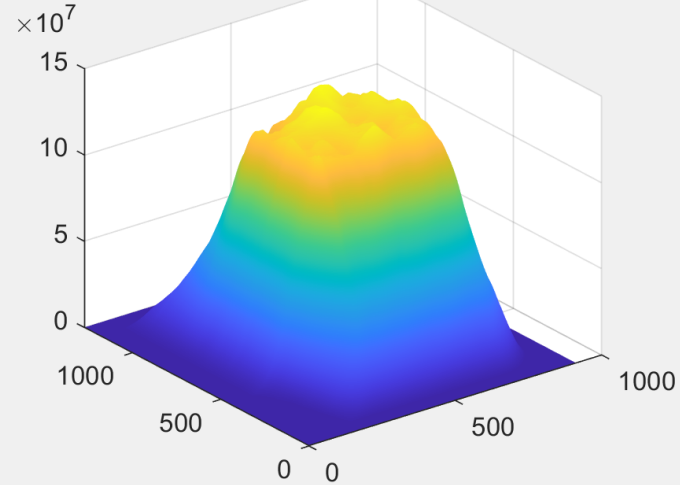


template norm.



moon norm.

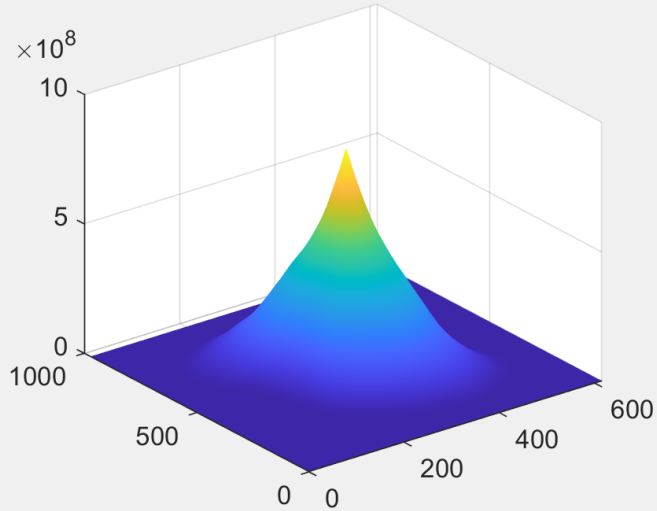
crr2



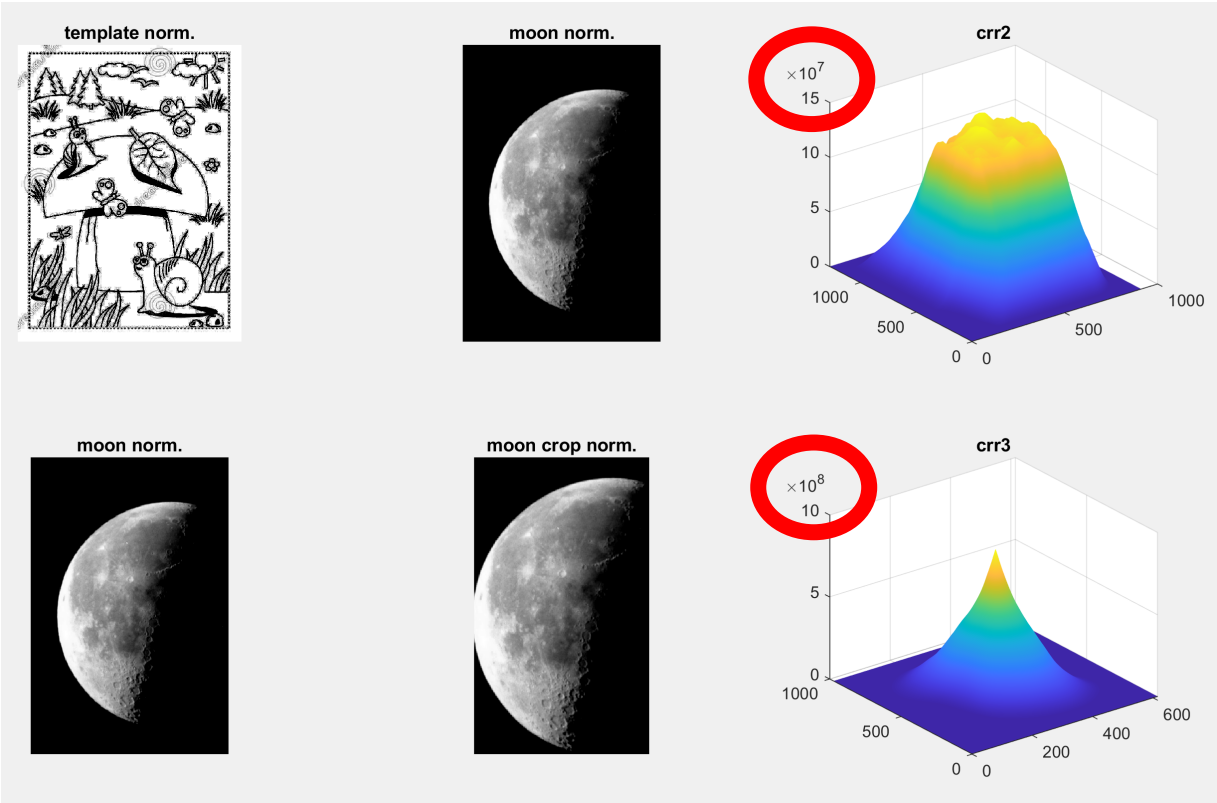
moon norm.

moon crop norm.

crr3



ccr max value comparison → normalization with respect to number of pixels is required



$$G = h \otimes F \qquad G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h[u,v] F[i+u, j+v]$$

Matlab

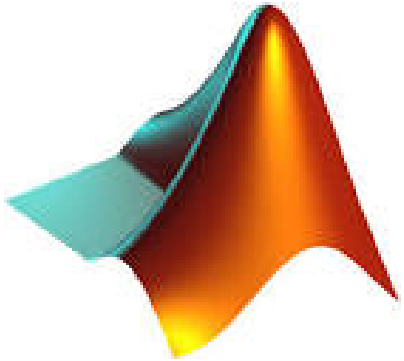
Max comparison → normalization with respect to number of pixels is required

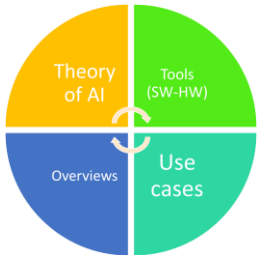
Normalized cross-correlation = (NCC) $t * f = \frac{1}{n} \sum_{x,y} \frac{1}{\sigma_f \sigma_t} (f(x,y) - \mu_f)(t(x,y) - \mu_t).$

1/n

Toolboxes/Theory
Similarity via conv.

The basis to Convolutional NNs.





Similarity ←→ Cross-correlation (convolution)

- It is important to understand very well the concept of similarity analysis via cross-correlation (convolution)
  → used in for Convolutional Neural Networks
  (CNNs, one of the most famous deep learning models)

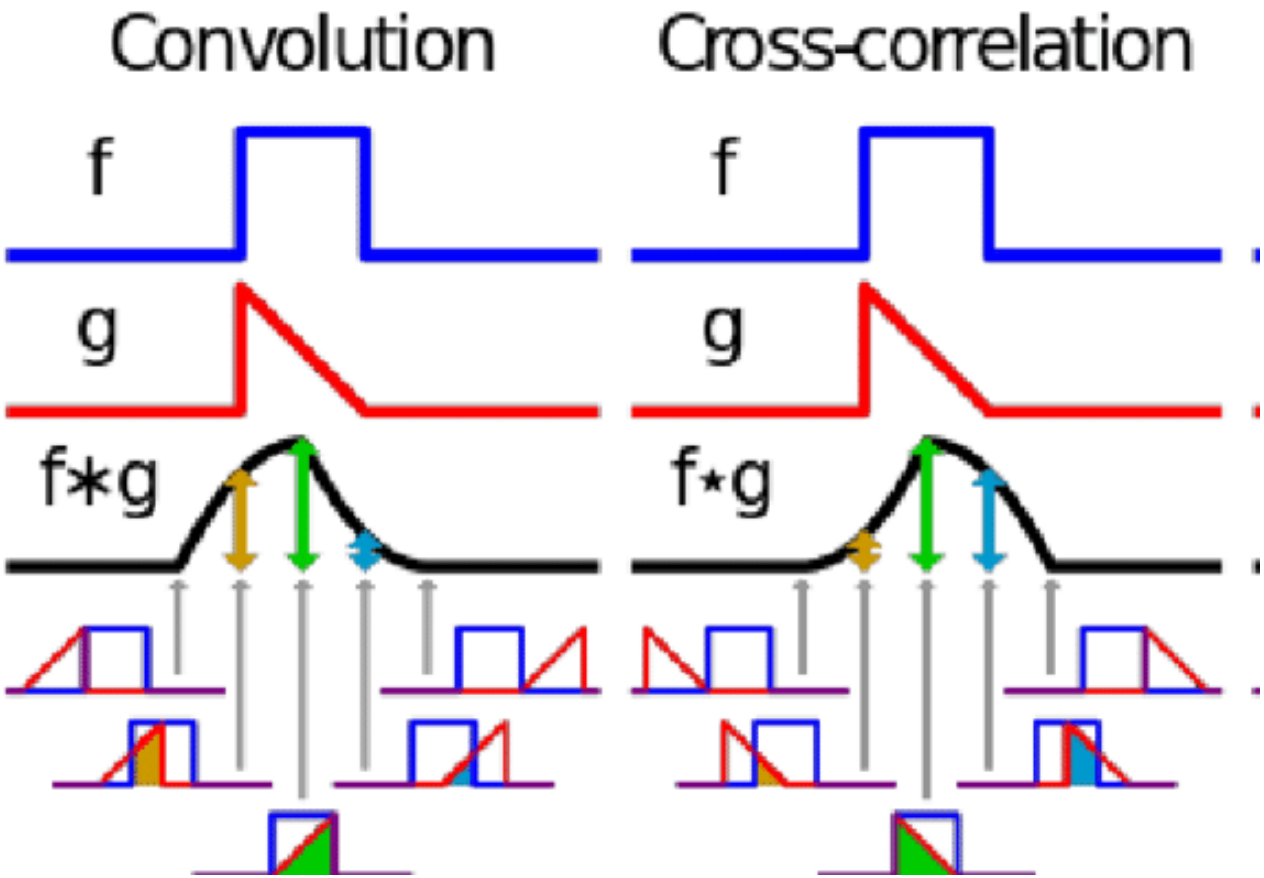
Cross-Correlation:

$$G = h \otimes F \qquad G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h[u,v] F[i+u, j+v]$$

Just the minus!

Convolution:

$$G = h * F \qquad G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h[u,v] F[i-u, j-v]$$



Cross-Correlation:

$$\text{CRR} = h \otimes F \quad \text{CRR}[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h[u,v] F[i+u, j+v]$$

Just the minus!

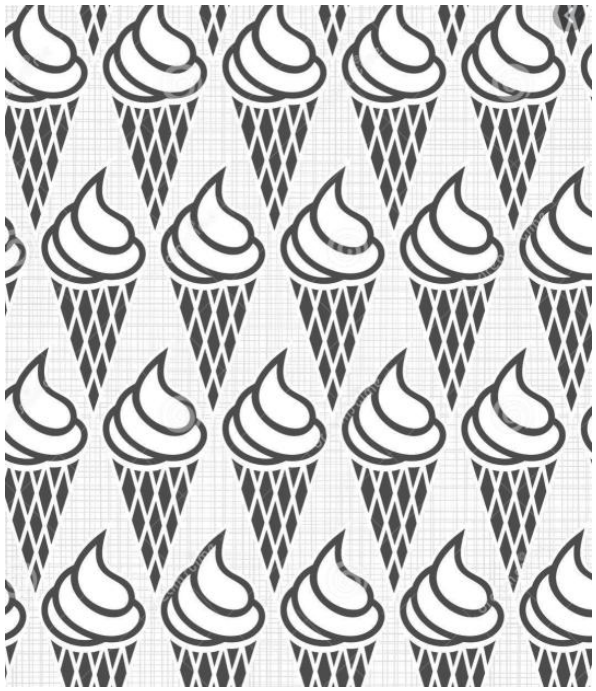
Convolution:

$$\text{CONV} = h * F \quad \text{CONV}[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h[u,v] F[i-u, j-v]$$

Similarity…

- Cross correlation and Internal similarity

(approaching the basis of CNNs)

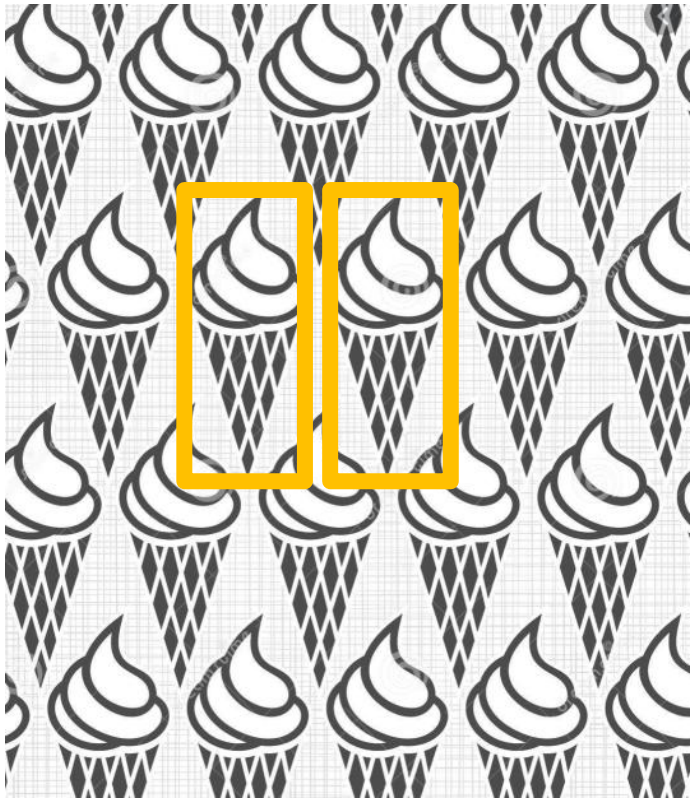


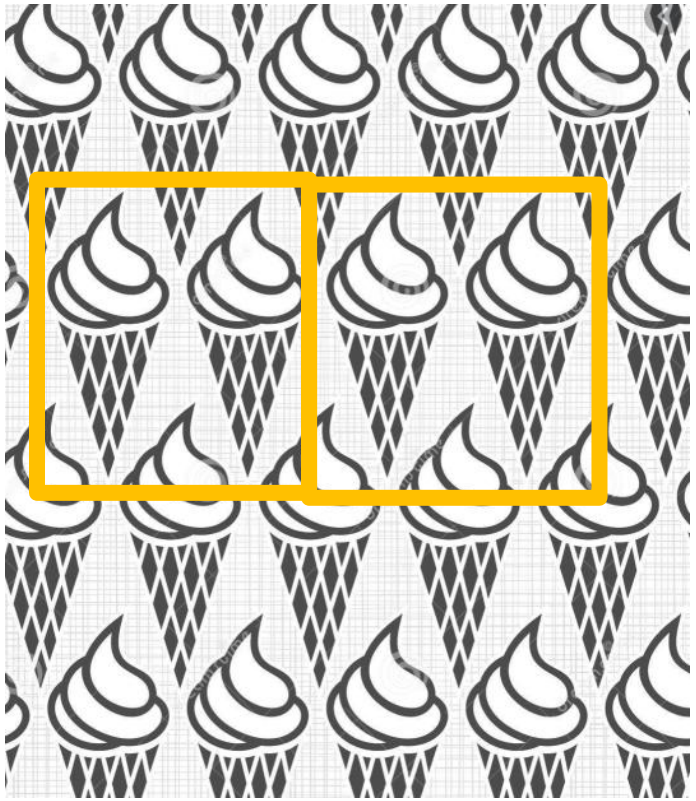
```
%% similarity
img_cream = double(rgb2gray( imread( 'ice_creams.jpg')));
img_cream_norm =  img_cream - mean(mean(img_cream));


h5 = figure
crr5 = xcorr2( double( img_cream_norm) , double(img_cream_norm)  ) ;
surf(crr5);
shading interp;   % just to remove black line around the
title('crr5 (same image)')
```

AUTOCORRELATION

Similarity…

- Internal similarity

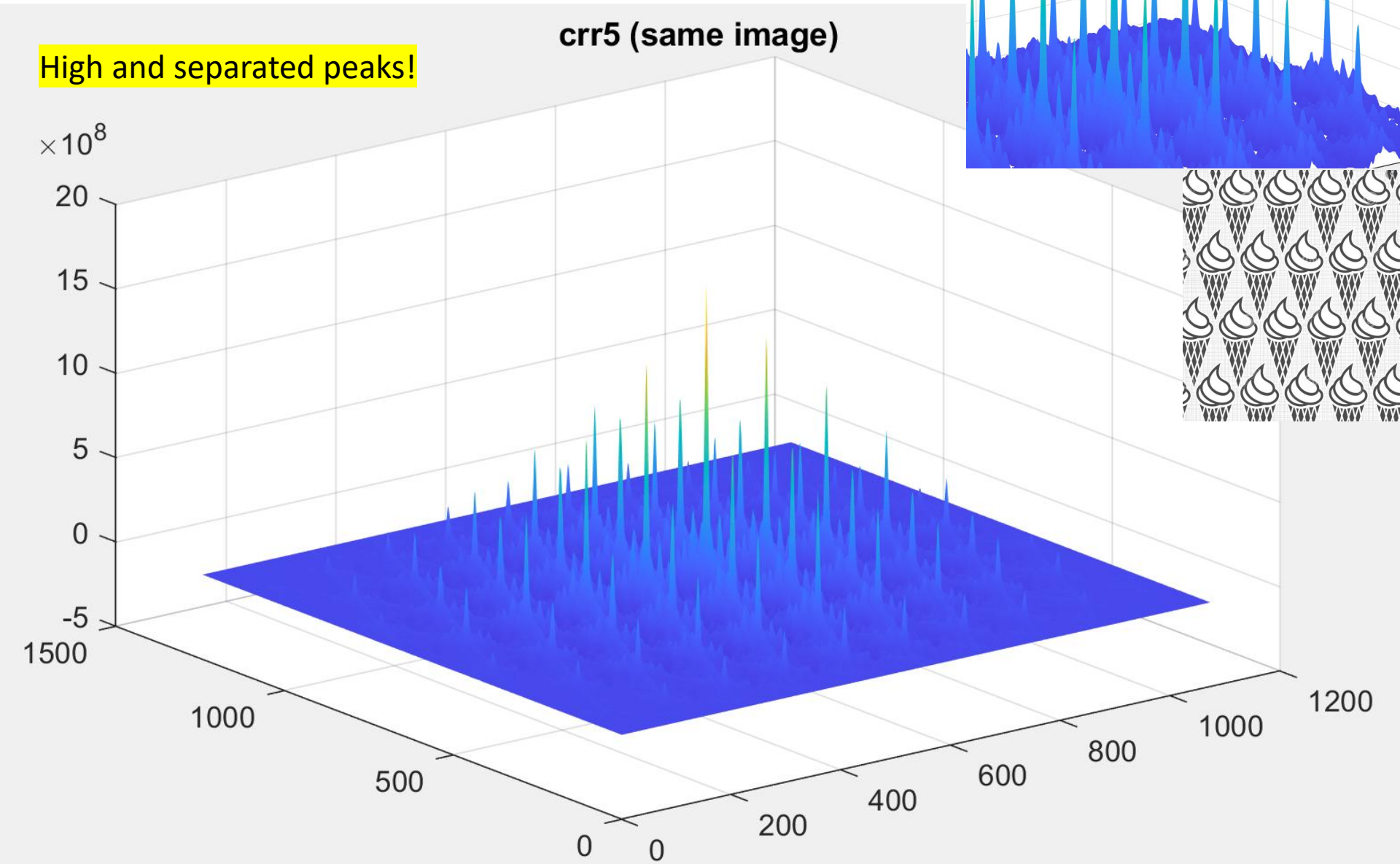




…

Similarity…

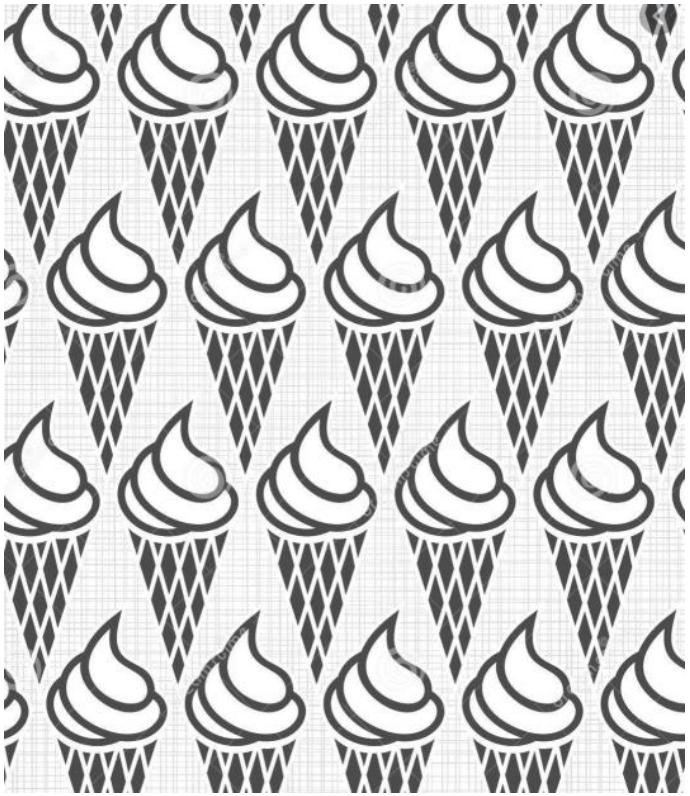
High and separated peaks!



Similarity…

General similarity between patterns…
Not just in images…. in data!

Images are different! (but similar)



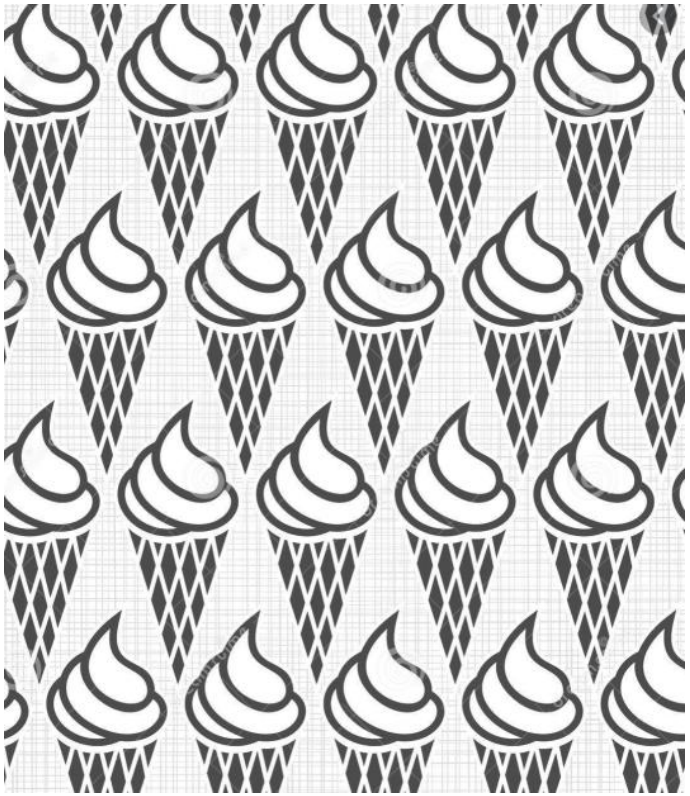
Find it!!

«KERNEL» of our search

Similarity…

General similarity between patterns…
Not just in images…. in data!

Images are different! (but similar)



Find it!!

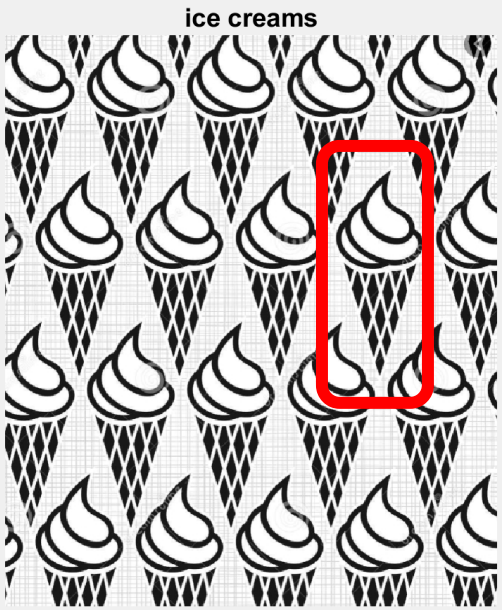
Let's ZOOM «KERNEL» of our search just for visualization

Prepare a similar pattern
(think to “patterns” not just an image…)

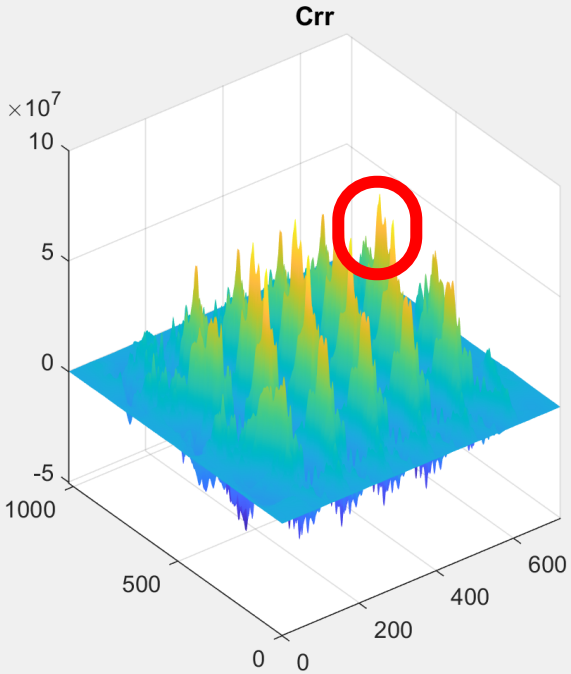
also with similar size!

```
%% similarity in CNN...
img_single = double(rgb2gray( imread( 'icecream2.jpg')));
img_single_norm =  img_single - mean(mean(img_single));


h6 = figure;
subplot(1,3,1); imshow( img_cream_norm, []); title('ice creams');
subplot(1,3,2); imshow( img_single_norm, []); title('similar single ice cream (kernel)');
crr6 = xcorr2( double( img_cream_norm) , double(img_single_norm)  ) ;
subplot(1,3,3);
surf(crr6);
shading interp;   % just to remove black line around the
title('Crr')
```

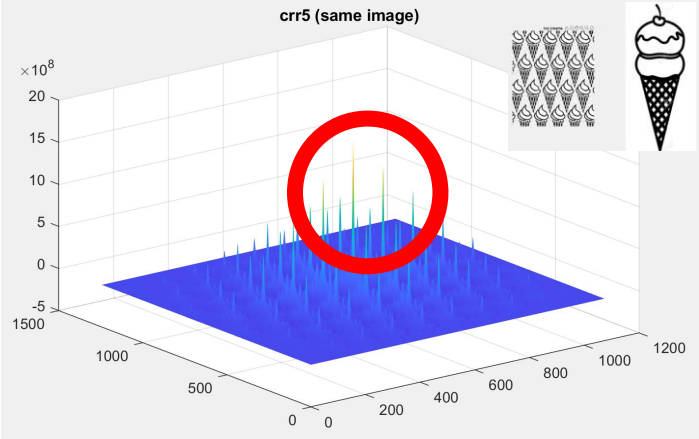
We found the similar pattern!

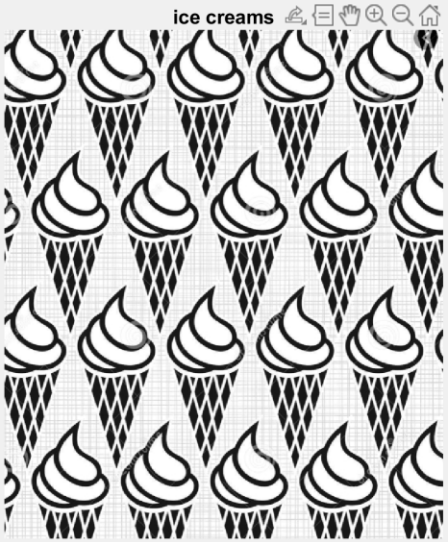


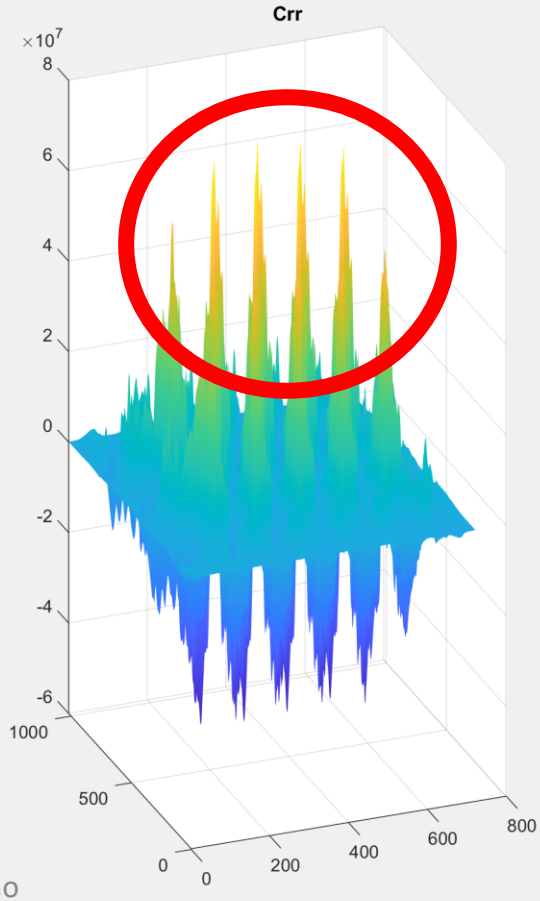
Note: this image is zoommed for visualizazion



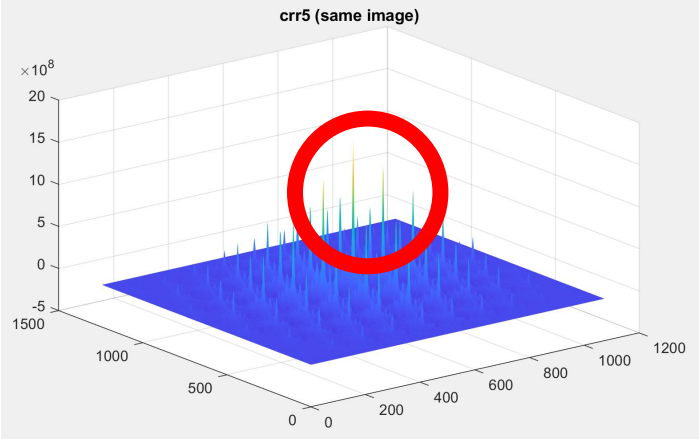
Peaks in the CCR



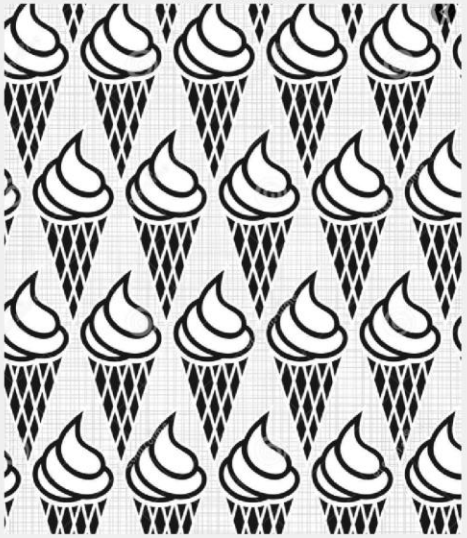




Rebuttal with a panda! →Noise in the CCR

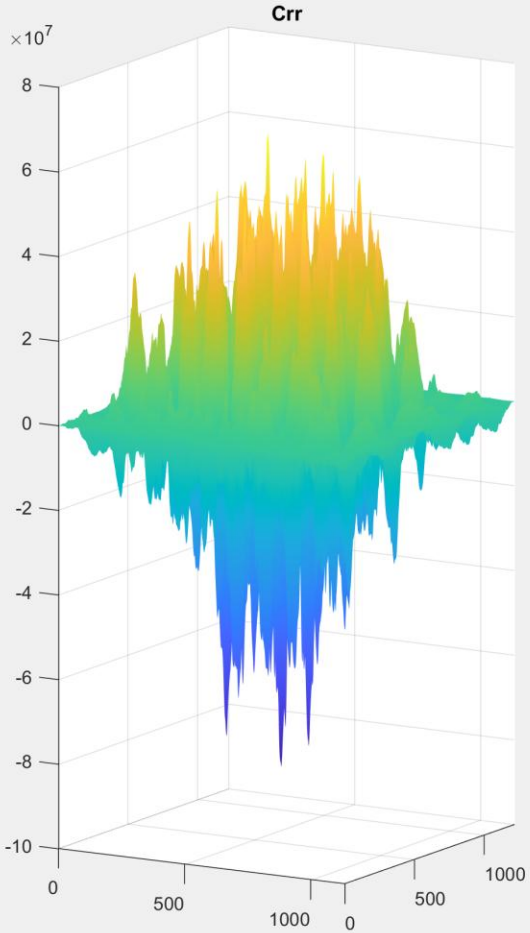


ice creams

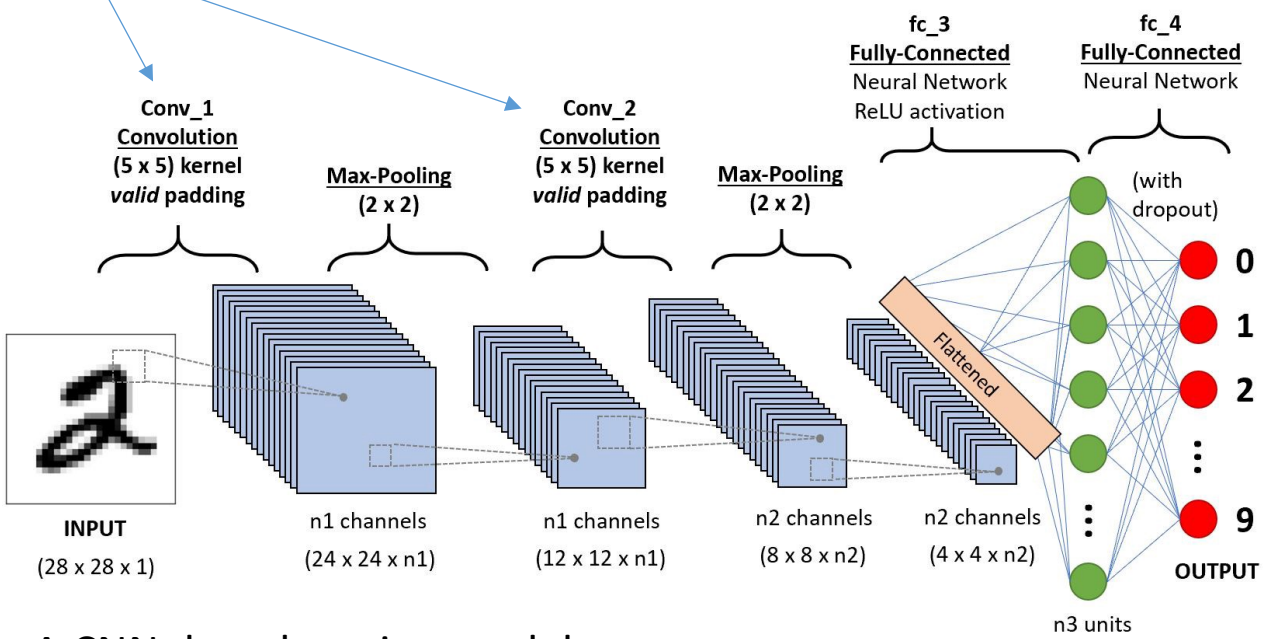


a panda (kernel)

Nevertheless, we have (noisy) peaks, since the panda is working as blob detector



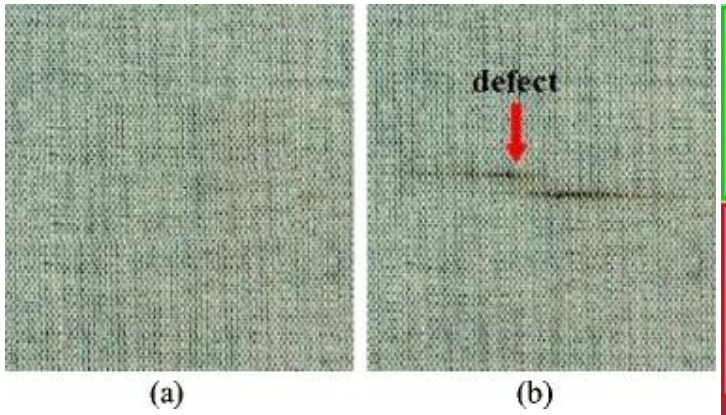
Convolution in deep learning

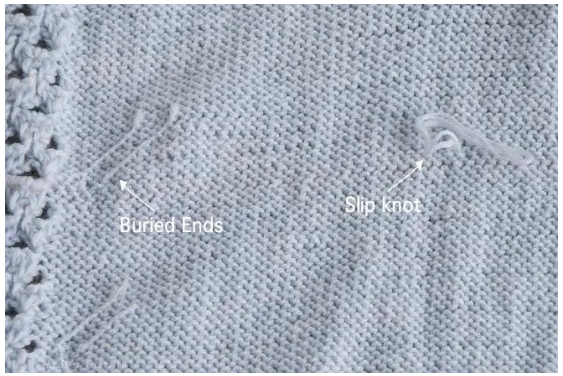


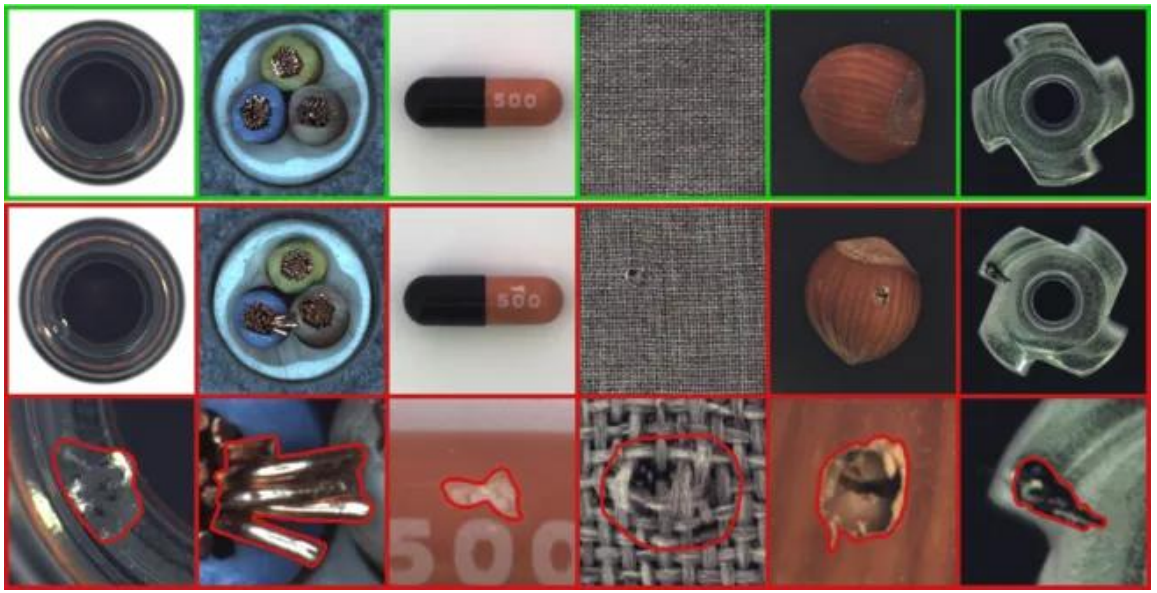
A CNN deep learning model

Similarity analysis to find defects in applications

- Defects in applications can be found as differences/lacks in the similarity pattern







</>
The code we used

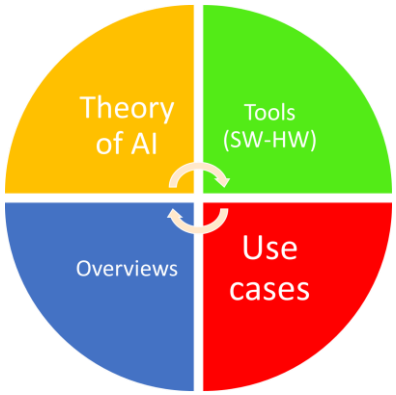
Note:
in the coding sessions
we did some simplifications
of the theory, just to get the main concepts without loosing ourselves in technicalities which are outside the scope of this course.

```
%% similarity
close all
img_cream = double(rgb2gray( imread( 'ice_creams.jpg')));
img_cream_norm =  img_cream - mean(mean(img_cream));


h5 = figure;
crr5 = xcorr2( double( img_cream_norm) , double(img_cream_norm)  ) ;
surf(crr5);
shading interp;   % just to remove black line around the
title('crr5 (same image)')


% similarity in CNN...
img_single = double(rgb2gray( imread( 'icecream2.jpg')));
img_single_norm =  img_single - mean(mean(img_single));


h6 = figure;
subplot(1,3,1); imshow( img_cream_norm, []); title('ice creams');
subplot(1,3,2); imshow( img_single_norm, []); title('similar single ice cream (kernel)');
crr6 = xcorr2( double( img_cream_norm) , double(img_single_norm)  ) ;
subplot(1,3,3);
surf(crr6);
shading interp;   % just to remove black line around the
title('Crr')

%
img_single = double(rgb2gray( imread( 'panda.jpg')));
img_single_norm =  img_single - mean(mean(img_single));


h6 = figure;
subplot(1,3,1); imshow( img_cream_norm, []); title('ice creams');
subplot(1,3,2); imshow( img_single_norm, []); title('a panda (kernel)');
crr6 = xcorr2( double( img_cream_norm) , double(img_single_norm)  ) ;
subplot(1,3,3);
surf(crr6);
shading interp;   % just to remove black line around the
title('Crr')
```

THEORY

Exploratory Data Analysis

Understanding what is inside your data

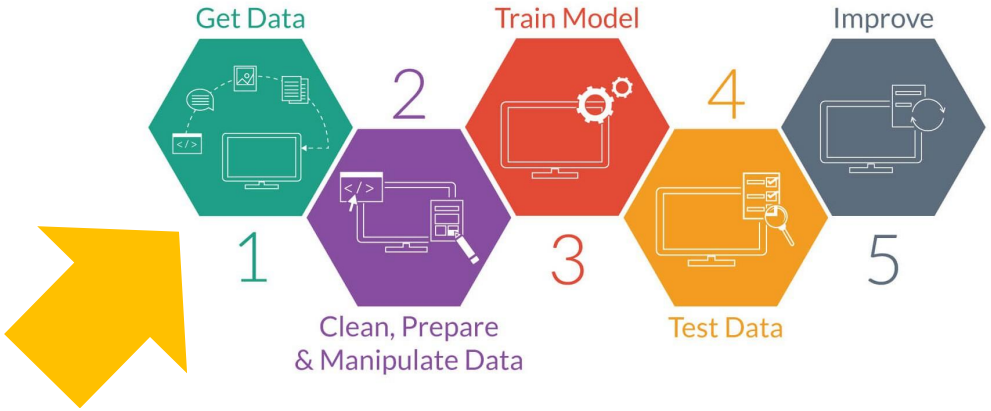


MEMO:
Step 1) of the ML workflow

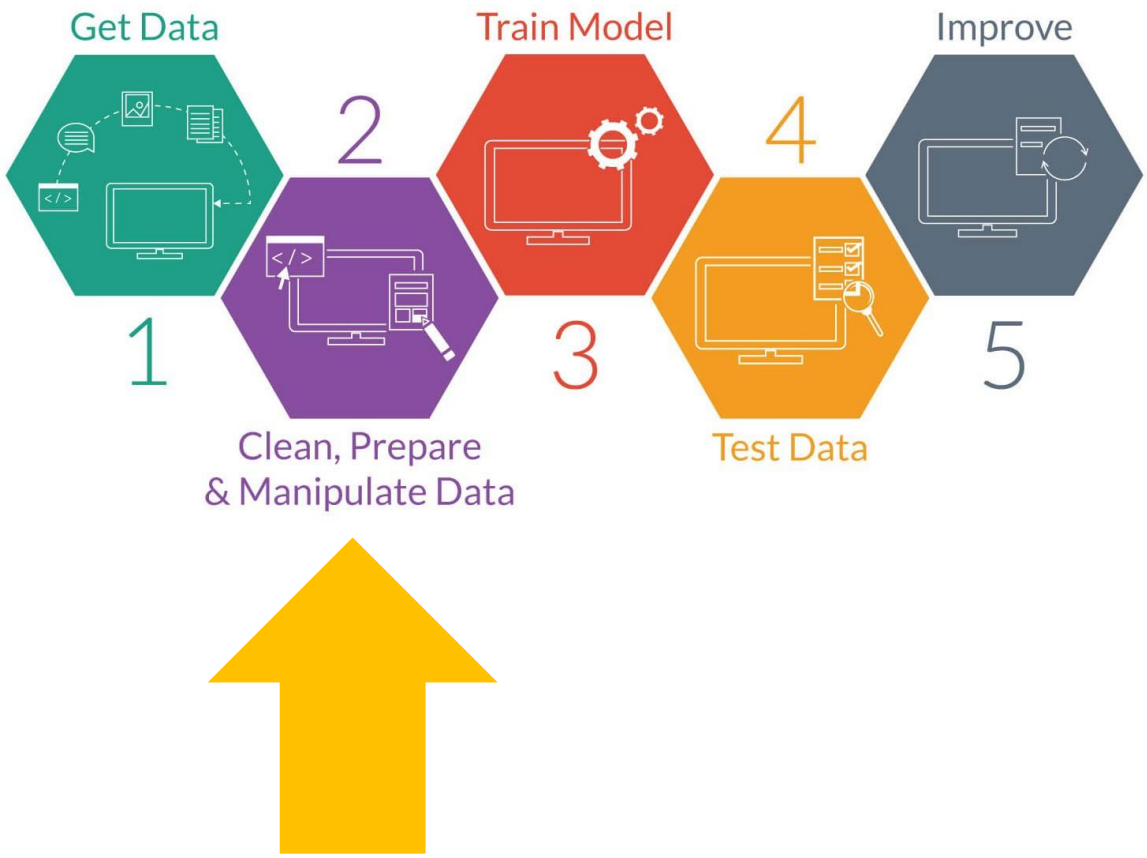
- Classical DB query
- File processing
- Custom formats
- Stream of data from IoT and AIoT
- Online platform (AWS, Azure, Google, …)

Outside the focus of this course

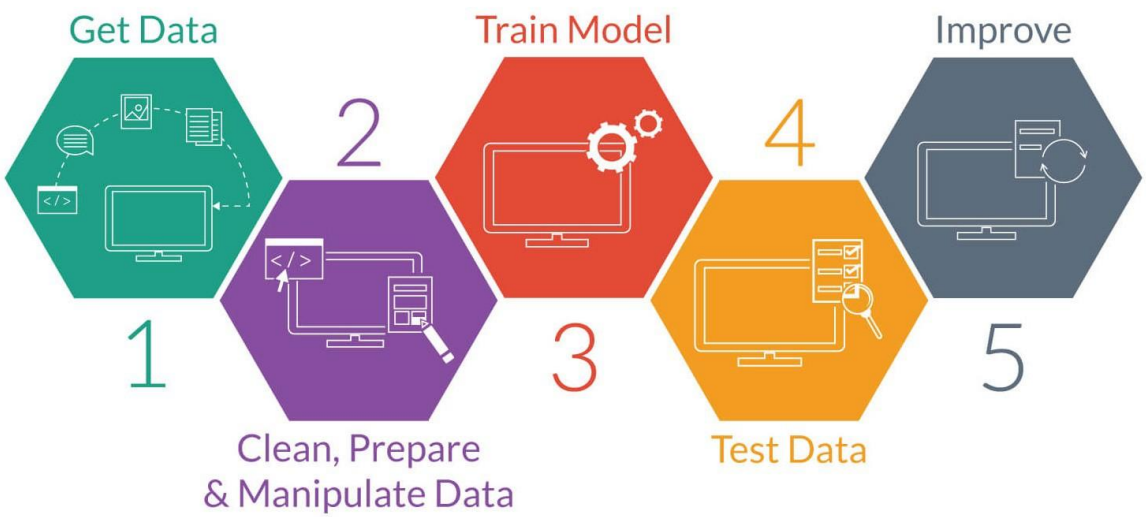
We have so many powerful sources capable to generate data



MEMO:
Step 2 of the ML workflow

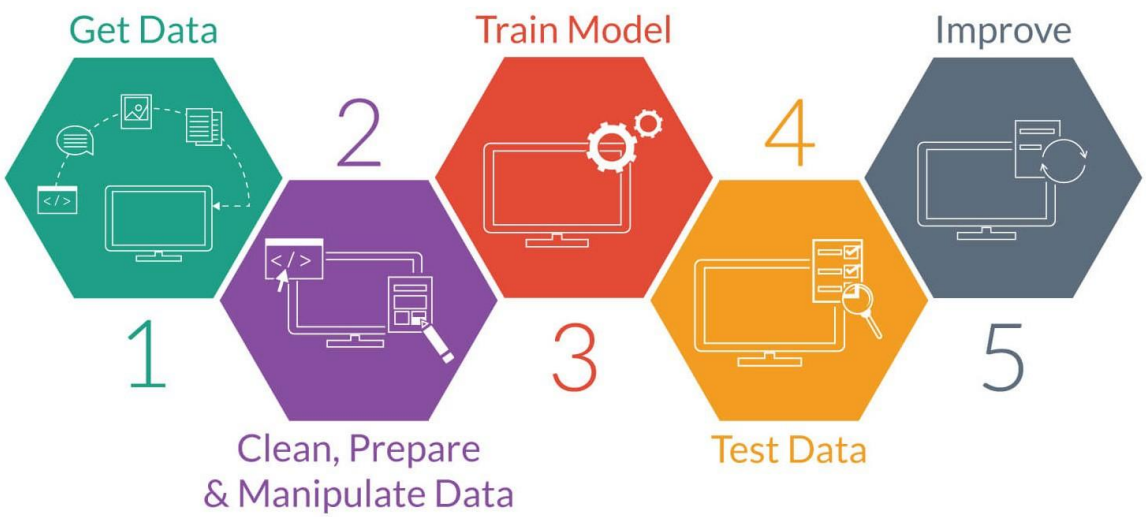


Step 2 of the ML workflow: Substeps



2a) Data Munging/Wrangling
2b) Exploratory Data Analysis
2c) Feature engineering

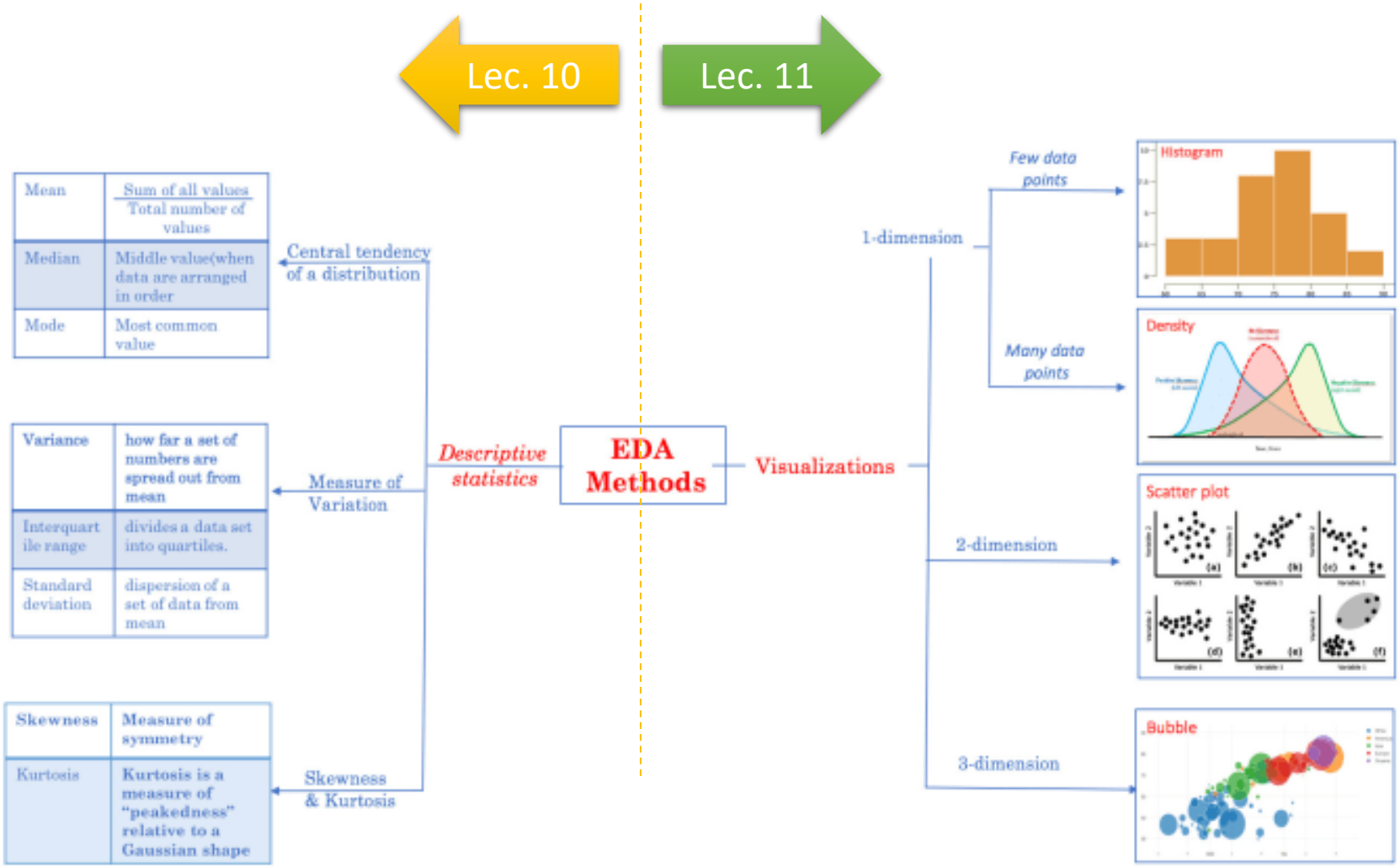
Step 2 of the ML workflow: Substeps



2a) Data Munging/Wrangling
2b) Exploratory Data Analysis
2c) Feature engineering

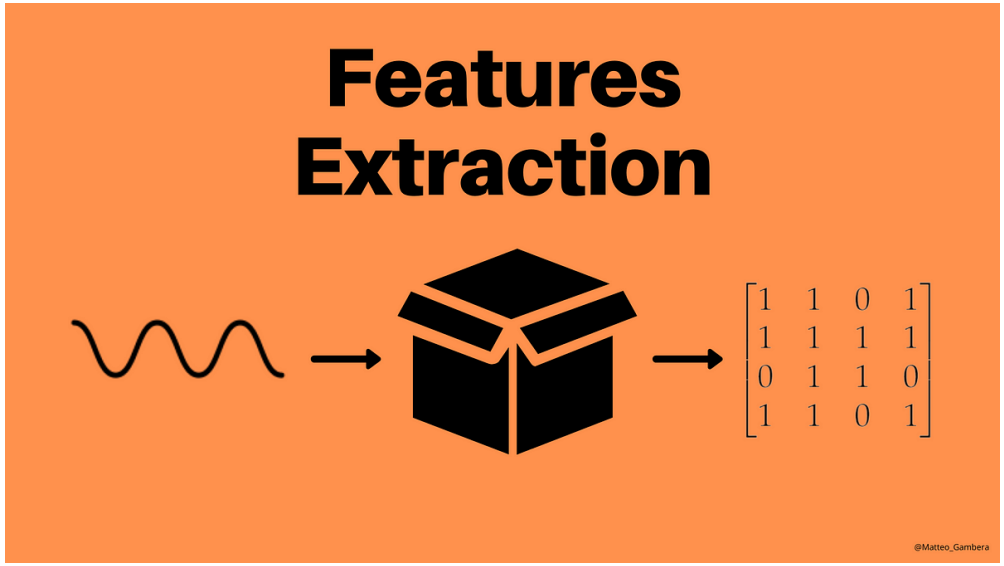
Exploratory Data Analysis

for ML



Different tools

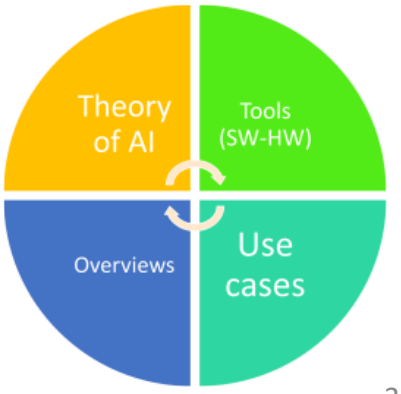
Exploratory Data Analysis in Matlab



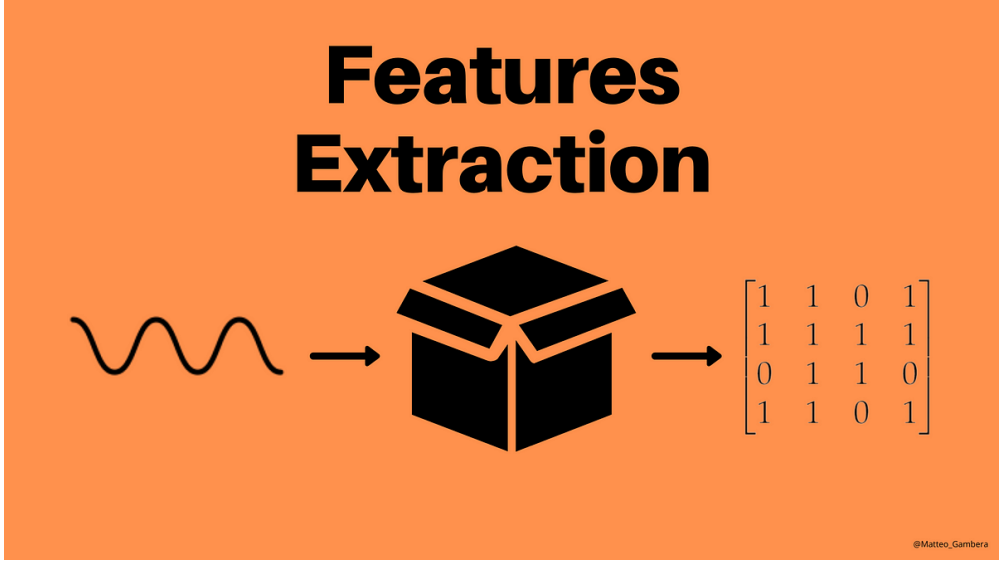
Theory

Time signals

and unstructured data



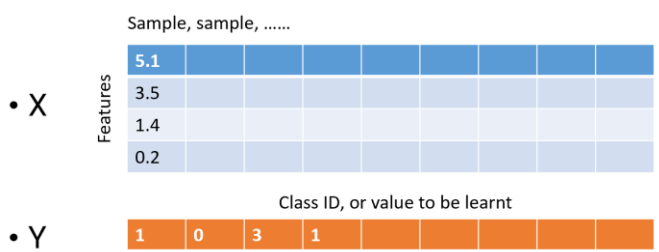
Feature extraction

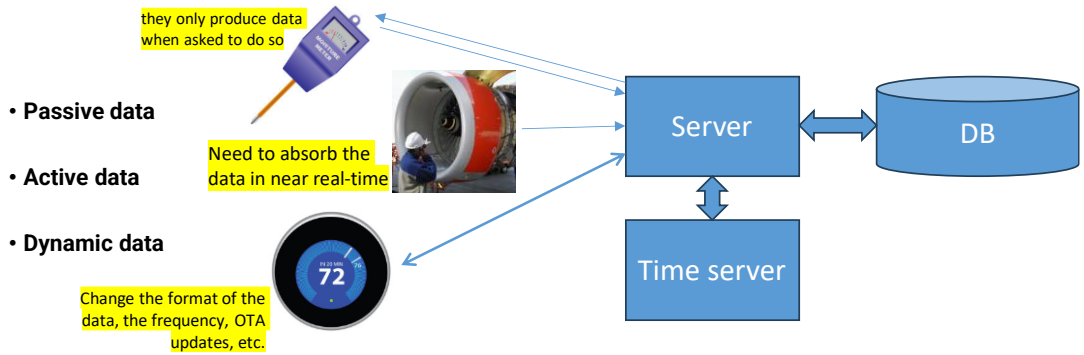


- The process of transforming raw data into a set of usable features sets that can be fed into a machine learning algorithm to train a model.
- This step is crucial because the quality and quantity of features used can significantly impact the model's performance.

Checking what is producing data (data context)

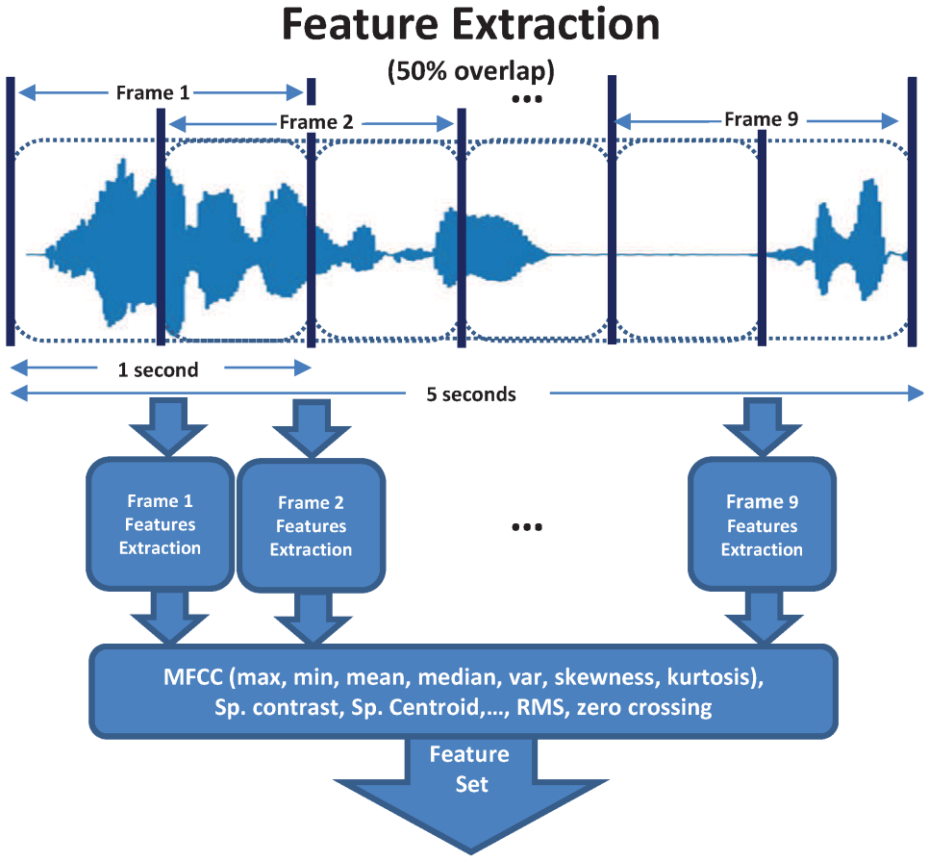
- **Source of the Data**: Knowing where the data comes from can help in assessing its quality, reliability, and relevance to the problem at hand.
- **Temporal and Spatial Context**: For time-series data or geospatial data, the context is crucial because patterns can change over time or across different locations.
- **First step**: What is your dataset? Is a table or a stream of data which is not yet structured?





Signals in time: what info I have to extract?

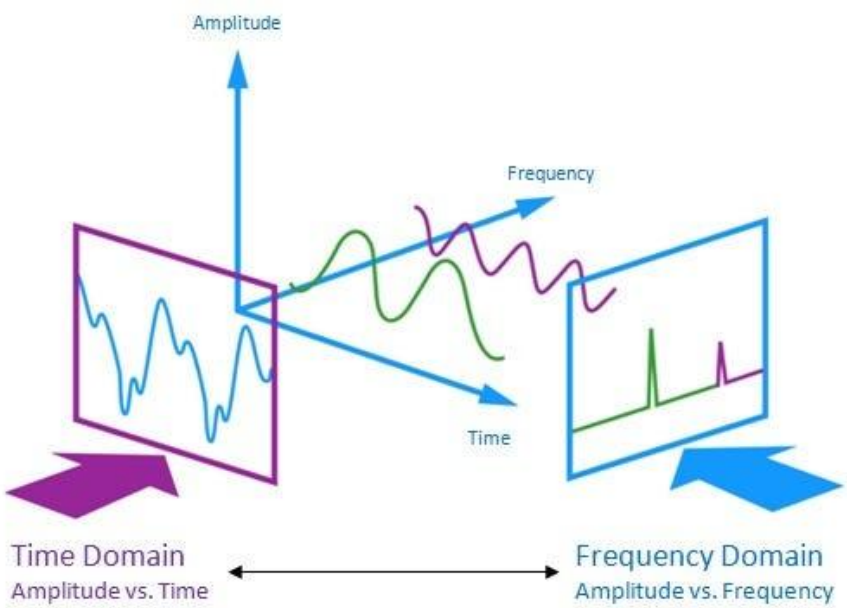
- For continuous signals, for each frame for each signal, a set of features is extracted.
- These features represent different characteristics of the signal within that frame.
- This is done because it's easier to analyze and classify discrete chunks of data than to work with a continuous signal.



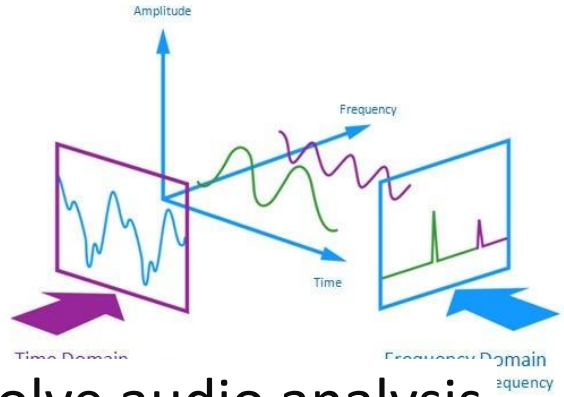
Process in the Time or Frequency domain?

- It is better to extract my feature in time or in the frequency domain?
- **Frequency Domain Representation**: On the right, we have the signal represented in the frequency domain, where the x-axis now represents frequency and the y-axis represents amplitude (or power). Each bar or line in the frequency domain represents the amplitude of a particular frequency component of the original time signal.

- This is a **spectrum** view showing the distribution of the signal's energy across different frequencies.



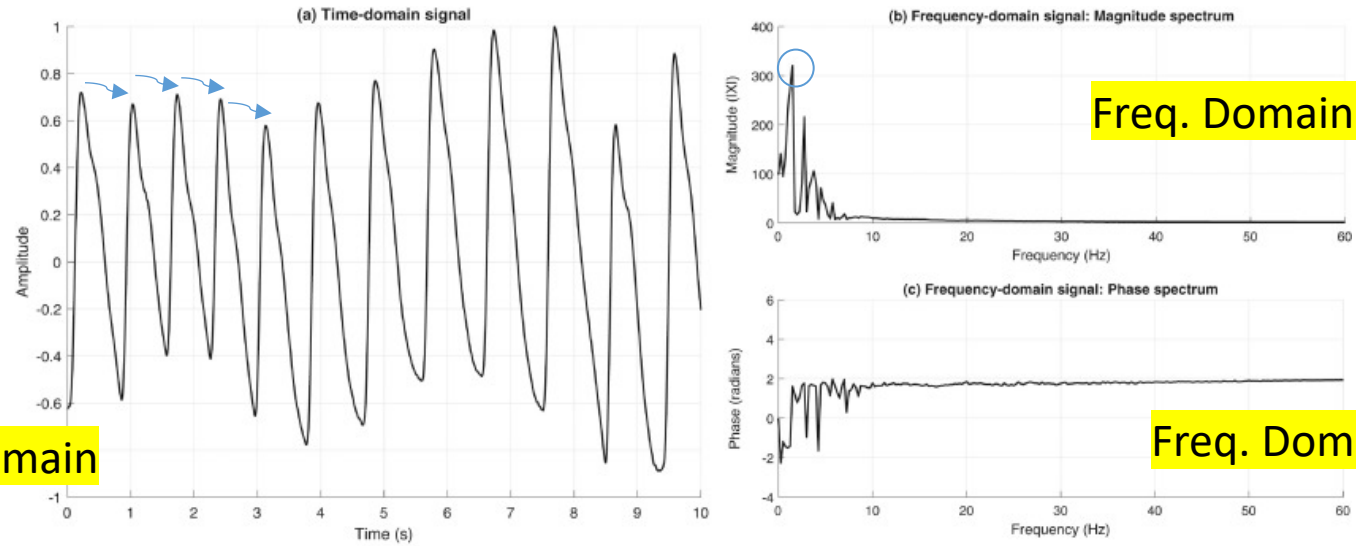
Process in the Time or Frequency domain?



In machine learning, particularly in tasks that involve audio analysis such as speech recognition, music classification, or sound event detection, it's crucial to analyze the signal in both domains:

- **Time domain** is good for understanding the temporal dynamics of the signal.
- **Frequency domain** provides insights into the harmonic content and can be essential for identifying the frequency-based features of the signal, such as vibrations, pitch or musical notes.

12 reps in 10s
→1.2Hz energy

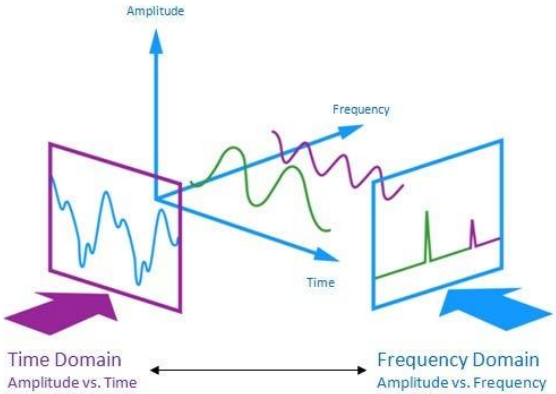


Time Domain

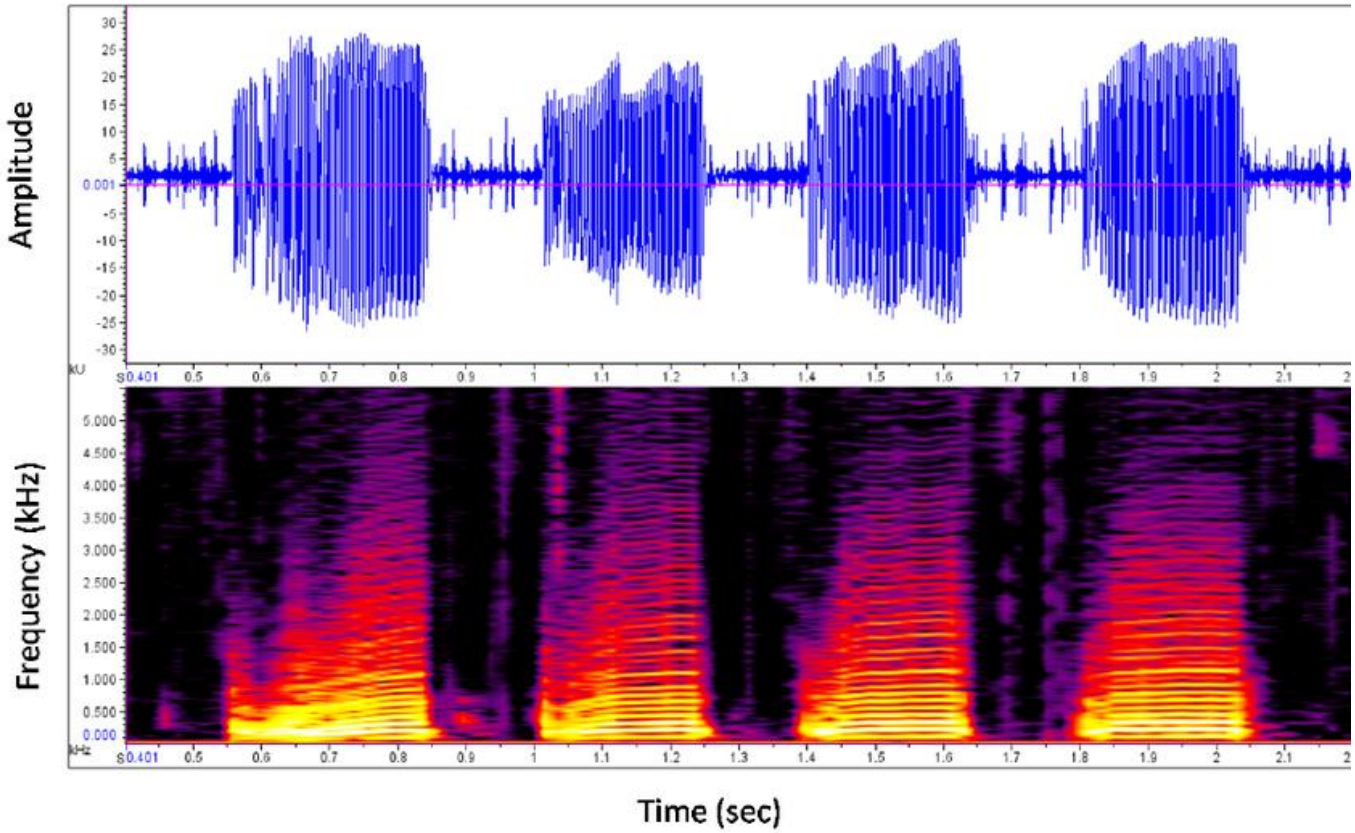
Freq. Domain (Magnitude)

Freq. Domain (Phase)

Process in the Time or Frequency domain?



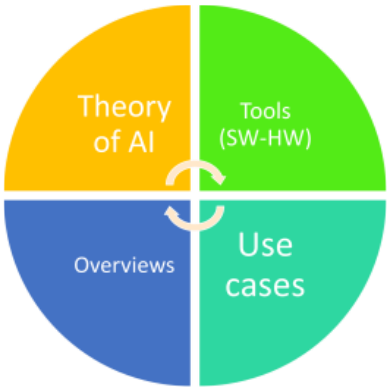
- Spectrogram
  indicates the amplitude or energy of a particular frequency at a particular time



Theory / Toolboxes

MathWorks®

Let’s start with the first tools about Exploratory Data Analysis (EDA) and Feature Engineering (FE)



Matlab's basic functions for Descriptive Statistics

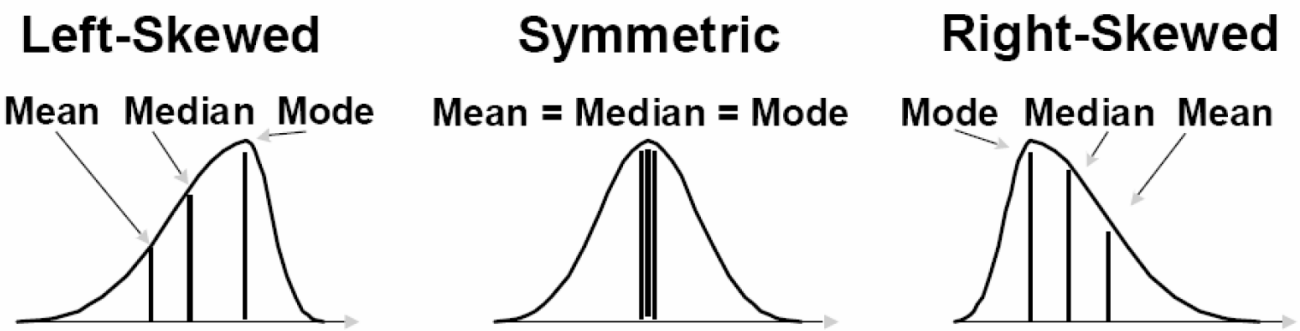
Very simple and effective commands

```
data = [1 2 3 4 50];

        % The arithmetic mean of the data:
mean(data) → → ans: 12

        %The median of the data:
median(data) → → ans: 3

        % The standard deviation:
std(data) → → ans: 21.27

        % The smallest value in the data:
min(data) → → ans: 1

        % The largest value in the data:
max(data ) → →  ans: 50
```

Descriptive Statistics (Exploratory Data Analysis) → Values used also in feature engineering

Shape of the distributions

1. Describes How Data Are Distributed
2. Measures of Shape
   - Skew = Symmetry

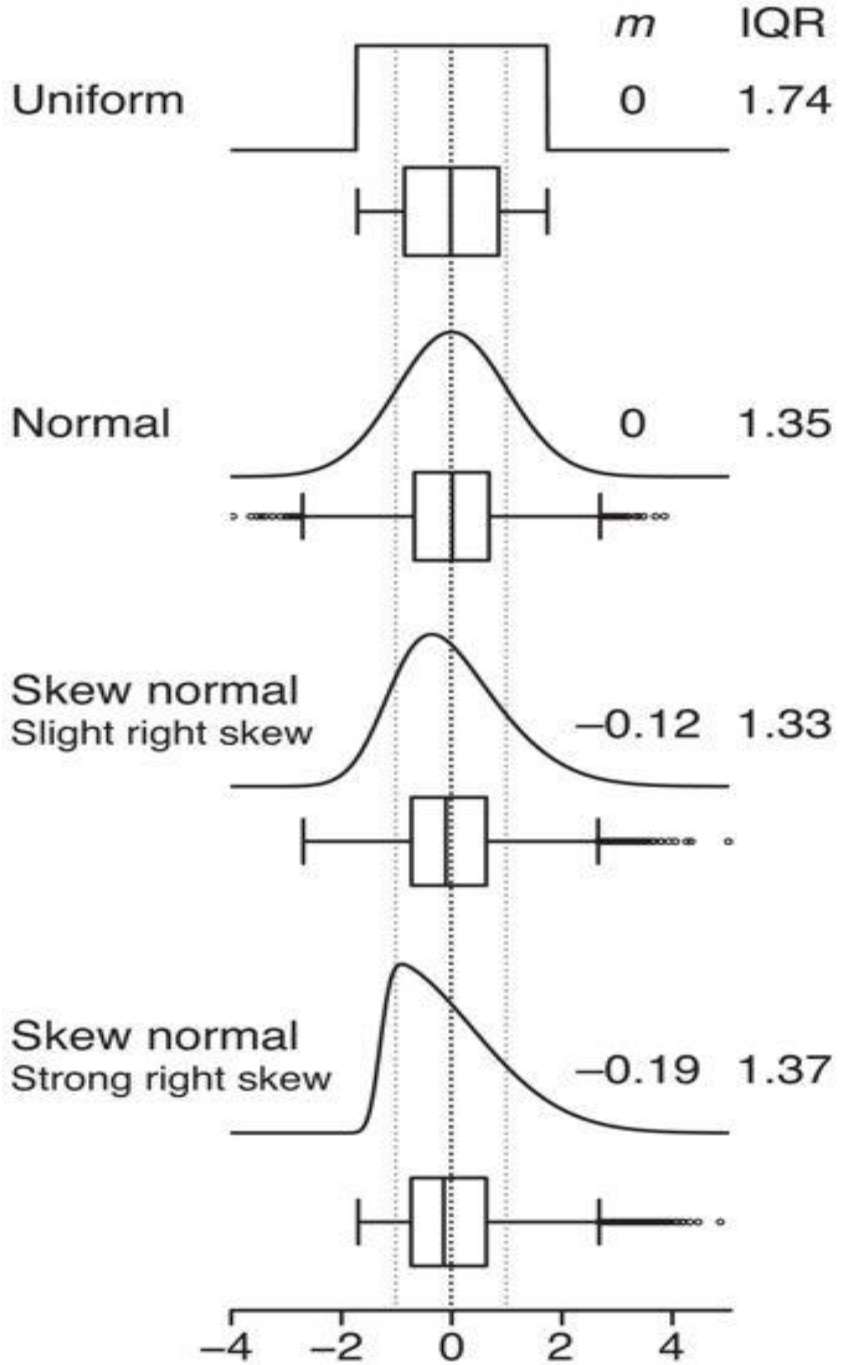


Example: in a regression error, the position and size of the tails are very important

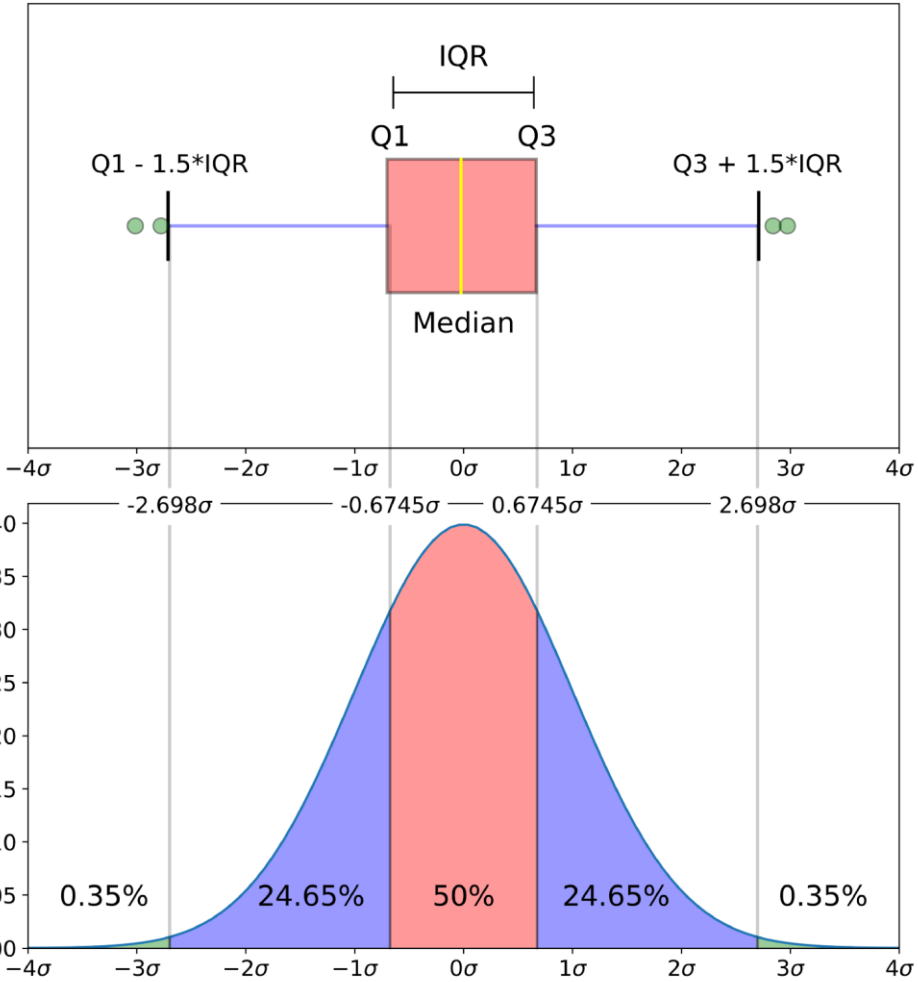
Distributions are more interesting…

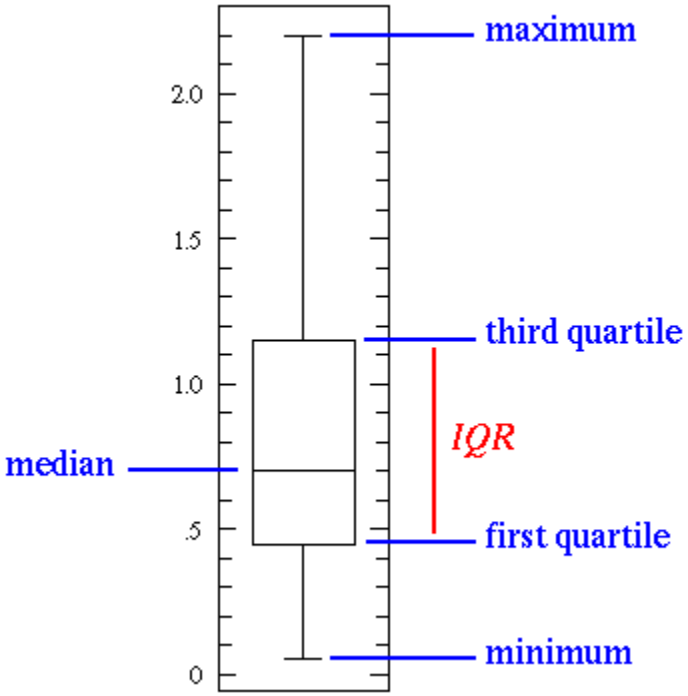
We have almost the same mean value and IQR, but the shape of the distributions and values range very differently

IQR = InterQuartile Range

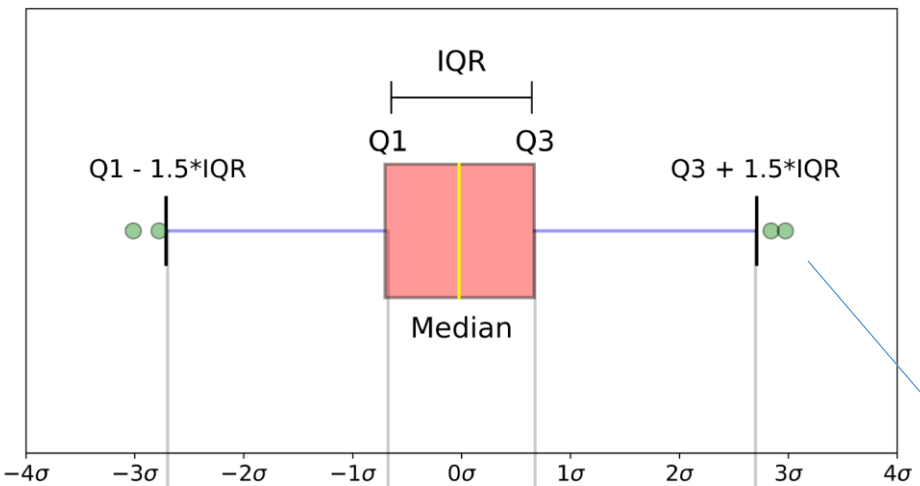


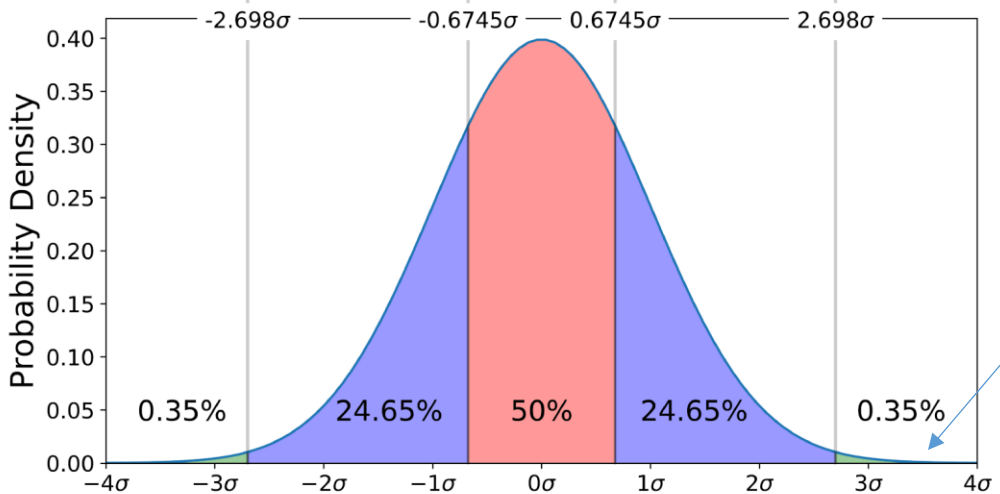
Box-plot

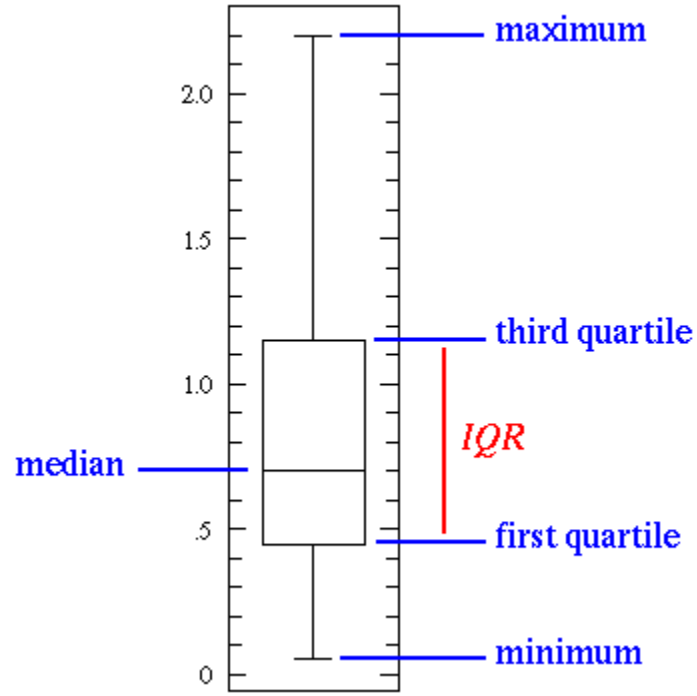

IQR
Q1
Q3
Q1 - 1.5*IQR
Q3 + 1.5*IQR
Median
−4σ −3σ −2σ −1σ 0σ 1σ 2σ 3σ 4σ
-2.698σ
-0.6745σ
0.6745σ
2.698σ
Probability Density
0.40 0.35 0.30 0.25 0.20 0.15 0.10 0.05 0.00
0.35%
24.65%
50%
24.65%
0.35%


maximum
2.0
1.5
third quartile
1.0
IQR
median
.5
first quartile
0
minimum

Choose box-plots with outliers!





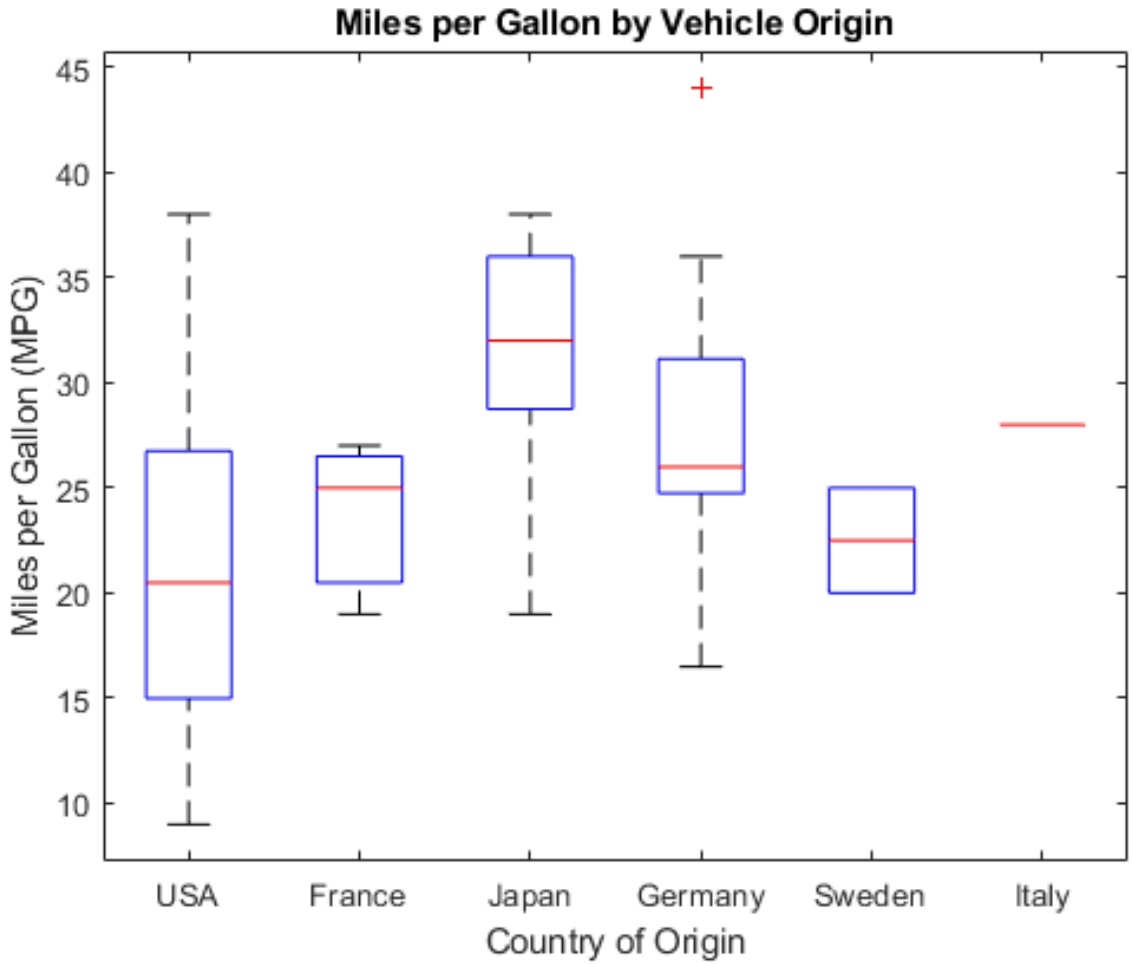


Outliers!!
Can you see them here?

Outliers can affect very much the normalization!!

The matlab boxplot

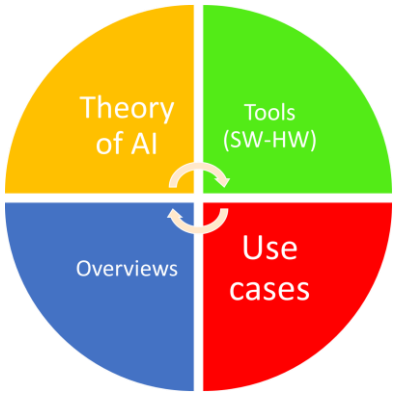
```
load carsmall
boxplot(MPG,Origin)

title('Miles per Gallon by Vehicle Origin')
xlabel('Country of Origin’)
ylabel('Miles per Gallon (MPG)')
```



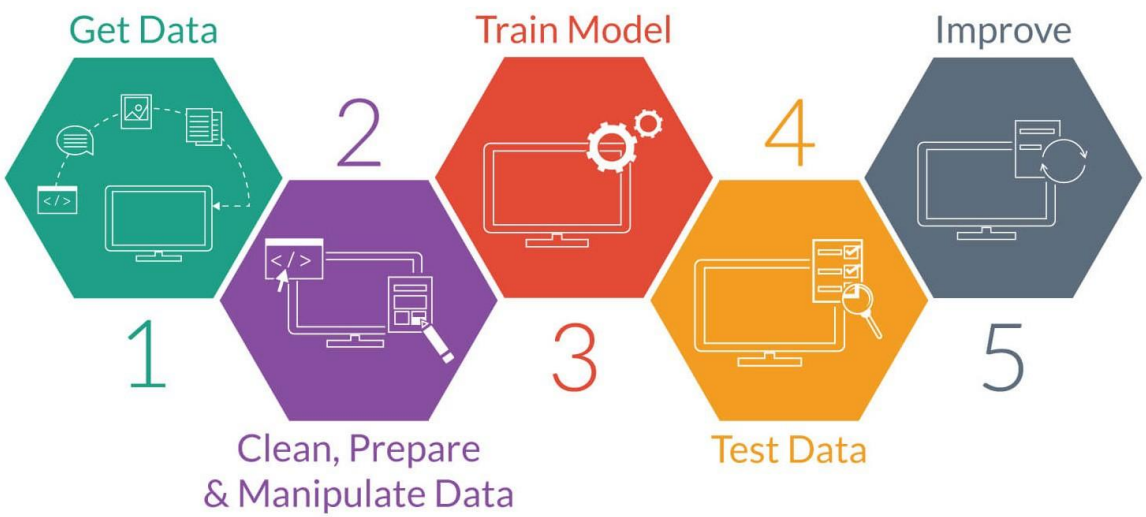
THEORY

Feature engineering

Design your features!



Step 2 of the ML workflow: Substeps

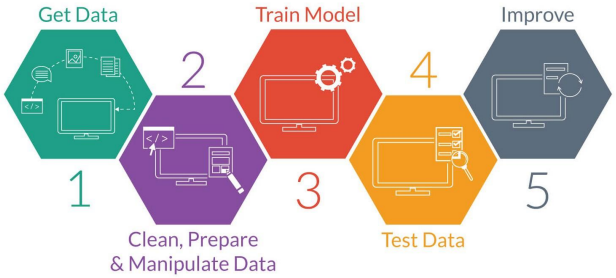


2a) Data Munging/Wrangling

2b) Exploratory Data Analysis

2c) Feature engineering

Feature engineering



- Brainstorming or testing features
- Deciding what features to create
- Creating features
- Checking how the features work with your model
- Improving your features if needed
- Go back to brainstorming/creating more features until the work is done.

AI Model → Y = FUNC(X)

- For every dataset in machine learning or toolbox, is all about to create X and Y

- X

Sample, sample, ……

Features

5.1								
3.5								
1.4								
0.2								

- Y

Class ID, or value to be learnt

1	**0**	**3**	**1**					

Y = FUNC(X)
FEATURE ENGINEERING (A)

- Optimizing one single feature at time!

SINGLE FEATURE

- X

Sample, sample, ……

Features

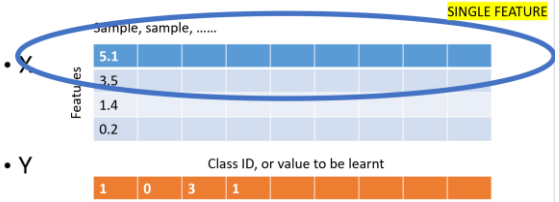
5.1								
3.5								
1.4								
0.2								

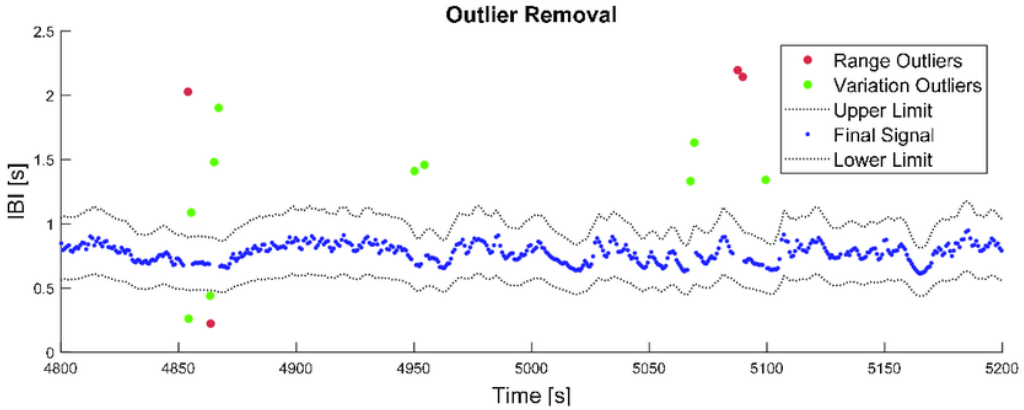
- Y

Class ID, or value to be learnt

1	0	3	1					

One single feature at time





- Outlier removal
- Harmonization
  - cm/mm/m → m
  - distances/similarities → distances
  - …

- Normalization/Standardization etc.. (see next slides)

Y = FUNC(X)
FEATURE ENGINEERING (B)

- Optimizing all features at time!

ALL FEATURES

Sample, sample, ……

- X

Features

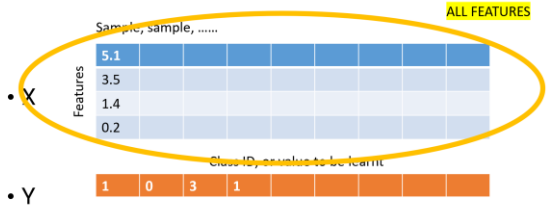
5.1								
3.5								
1.4								
0.2								

Class ID, or value to be learnt

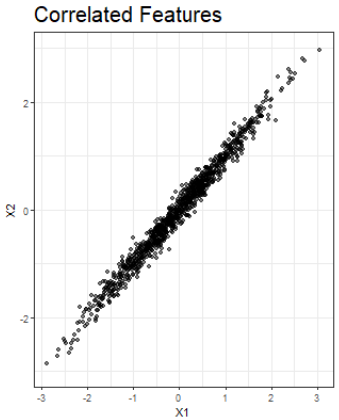
- Y

1	0	3	1					

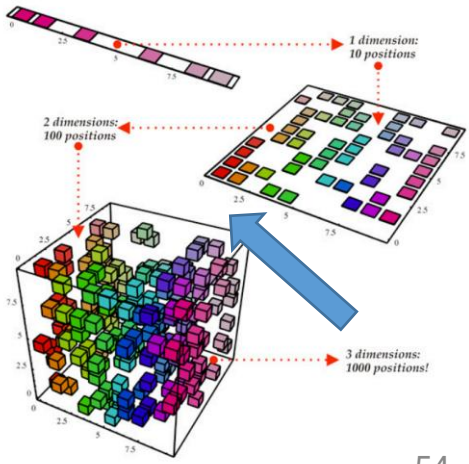
All features at time (no labels)



- X
- Y



- Remove strongly correlated features
- Creation of powerful combinations
- Principal Component Analysis and Dimensionality Reduction Methods
- etc…



Y = FUNC(X)
FEATURE ENGINEERING (C)

- Optimizing all the dataset at time!

- X

Sample, sample, ……

Features

5.1								
3.5								
1.4								
0.2								

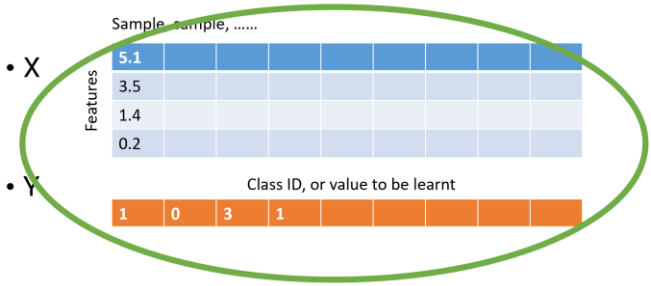
- Y

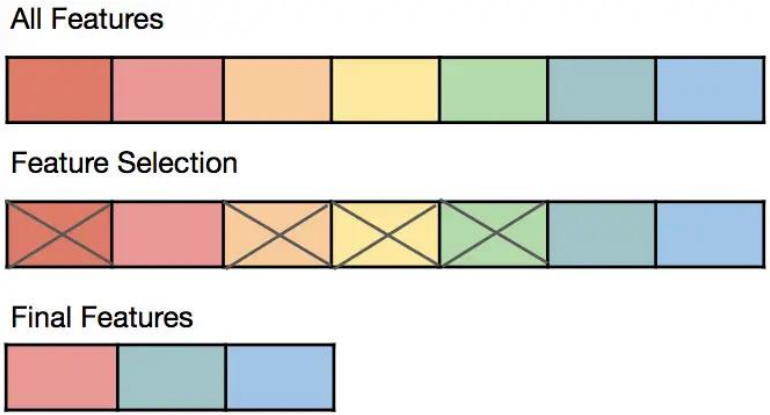
Class ID, or value to be learnt

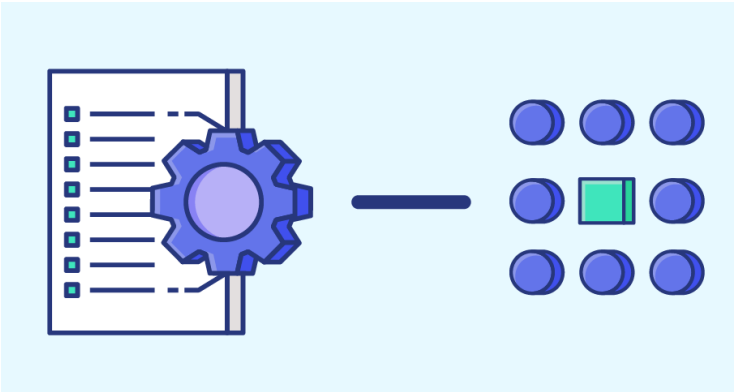
1	**0**	**3**	**1**					

Optimizing all features using labels

- Feature extraction
- Feature selection
- Feature engineering
- Etc…







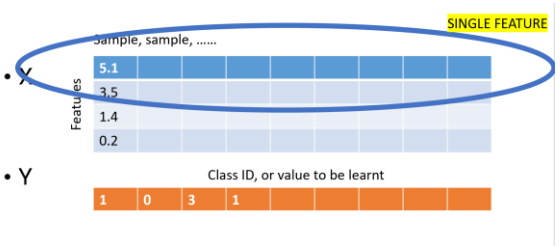
Data preprocessing: a **statistical point of view**

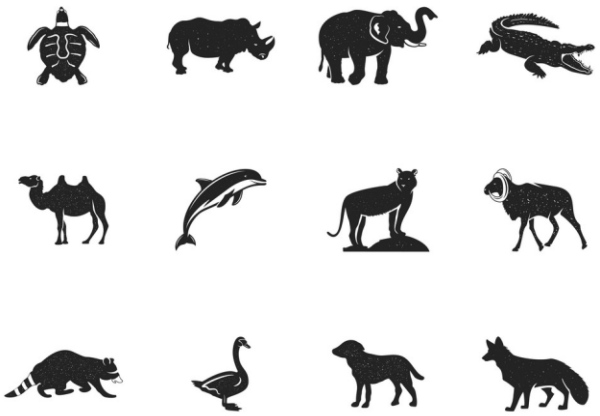
For intelligent systems

Note:
part of Exploratory Data Analysis (EDA)
and part of Feature Engineering (FE)

Feature engineering:
Normalization

For intelligent systems datasets





(shapes not sizes…)

Feature Processing and Normalization

- The goal of normalization is to transform features to be on a similar scale. This improves the performance and training stability of the model.

- 4 common Feature Processing techniques:
  - scaling to a range $x' = (x - x_{min})/(x_{max} - x_{min})$
  - clipping *if (x > T) x' = T, otherwise x' = x*
  - log scaling $x' = log(x)$
  - z-score $x' = (x - \mu)/\sigma$

A minor note…

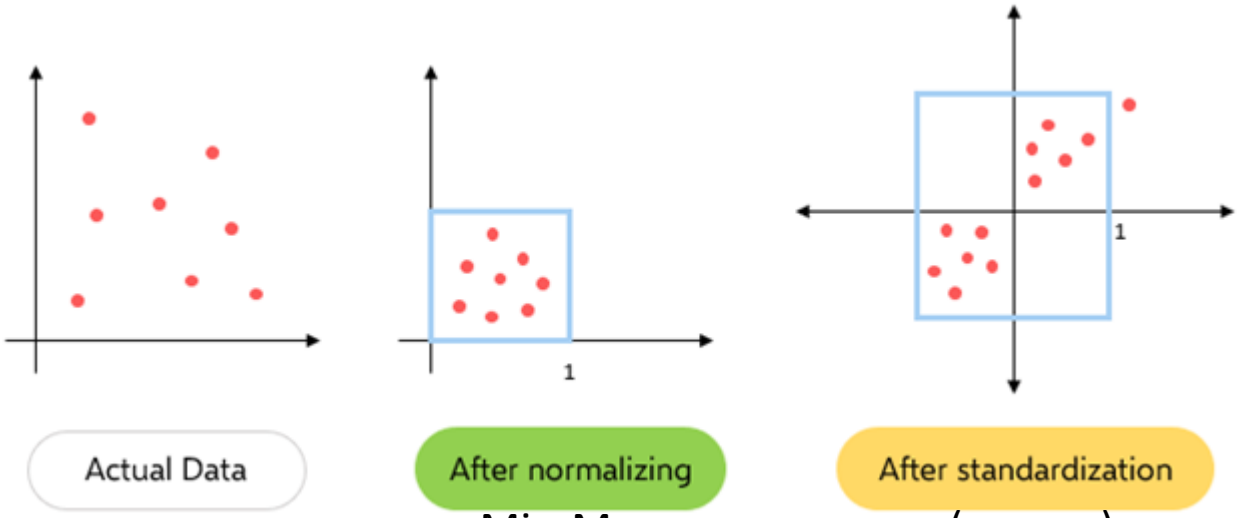
Feature scaling

Min-Max **Normalization**

Standardization (z-score)

$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

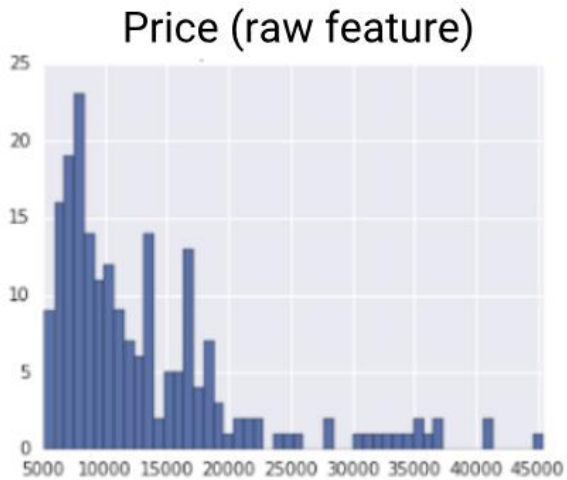
$$X' = \frac{X - \text{Mean}}{\text{Standard deviation}}$$

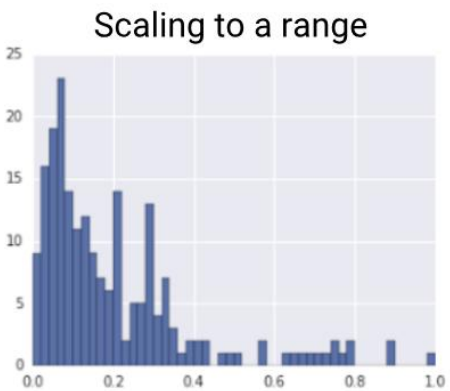


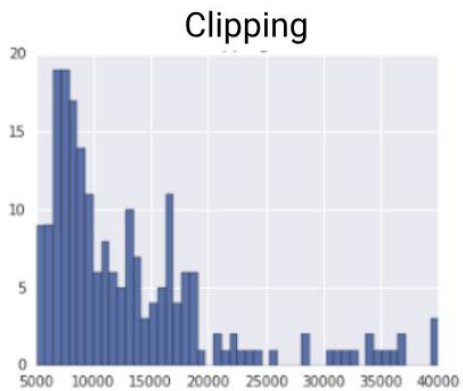
Min-Max (z-score)

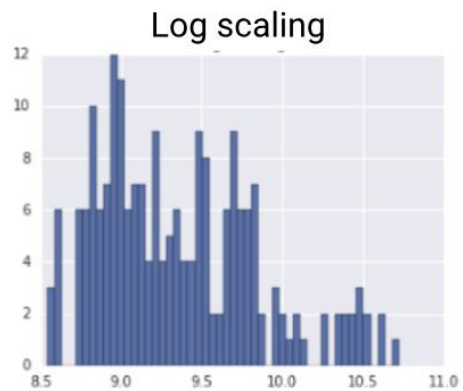
Examples of feature processing

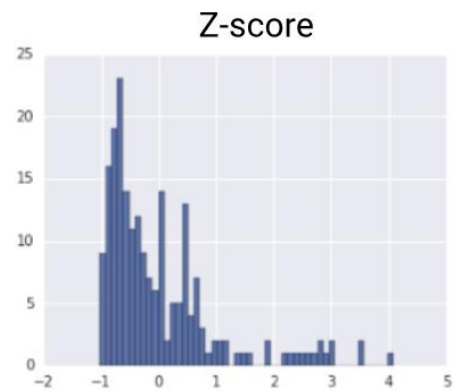
Ex.: Price for cars:
O EUR :0 Cars… …
12K EUR :some…
250k EUR :only few…
..









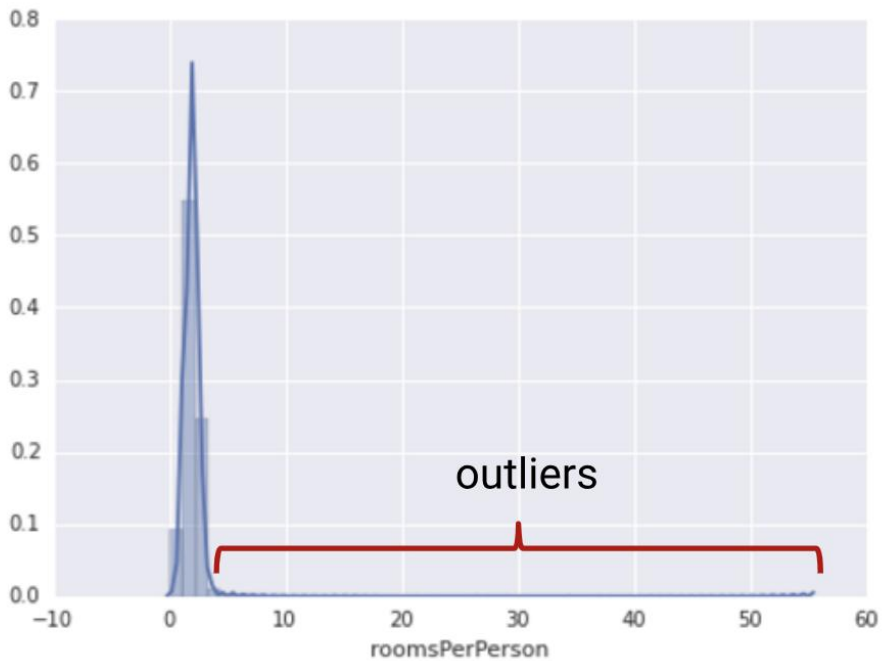


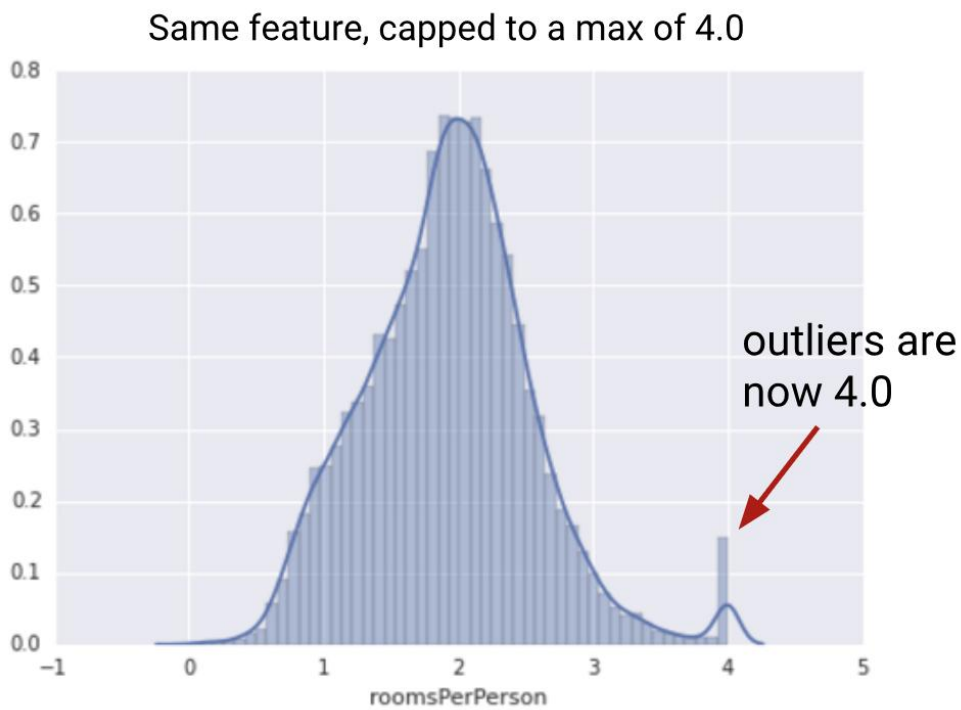
Feature Clipping

Hint!

Clip by z-score to **+-Nσ**
(for example, limit to +-3σ).
σ is the standard deviation

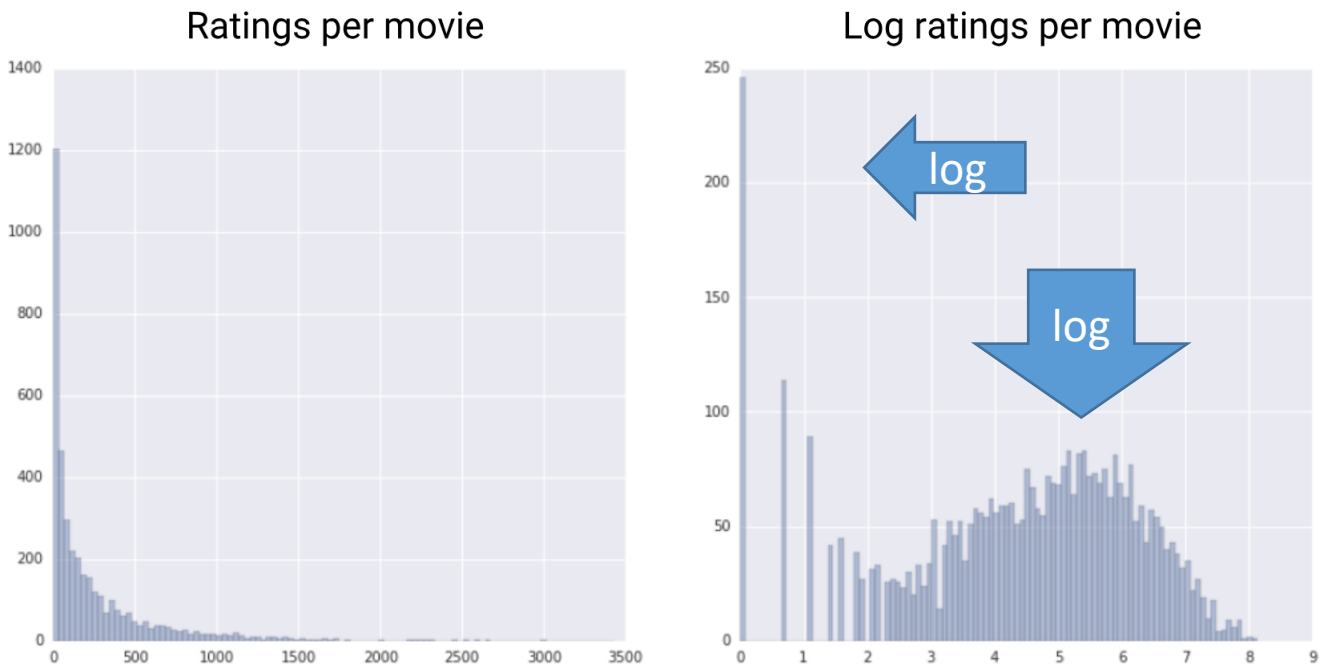
- If your data set contains extreme outliers, capping all feature values above (or below) a certain value to fixed value.





Log Scaling

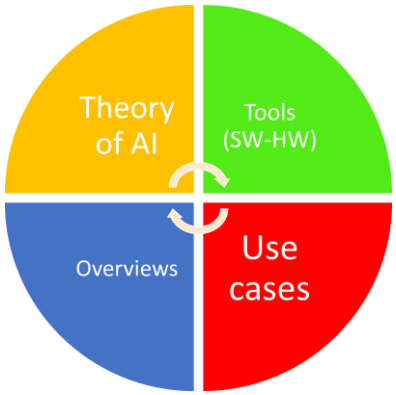
- When a handful of your values have many points, while most other values have few points.
  - This data distribution is known as the *power law* distribution.



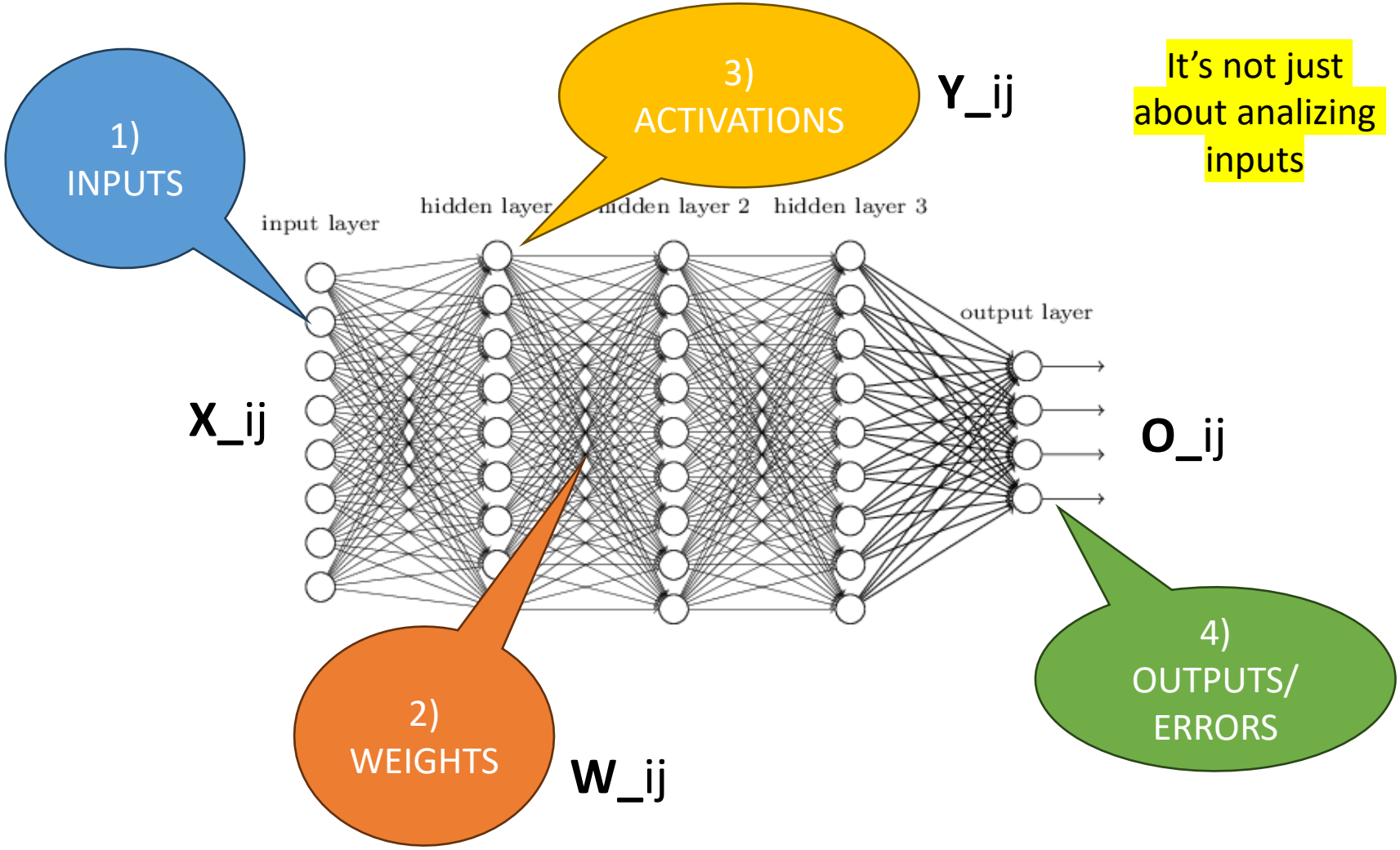
THEORY

Statistical Data Exploration in NN

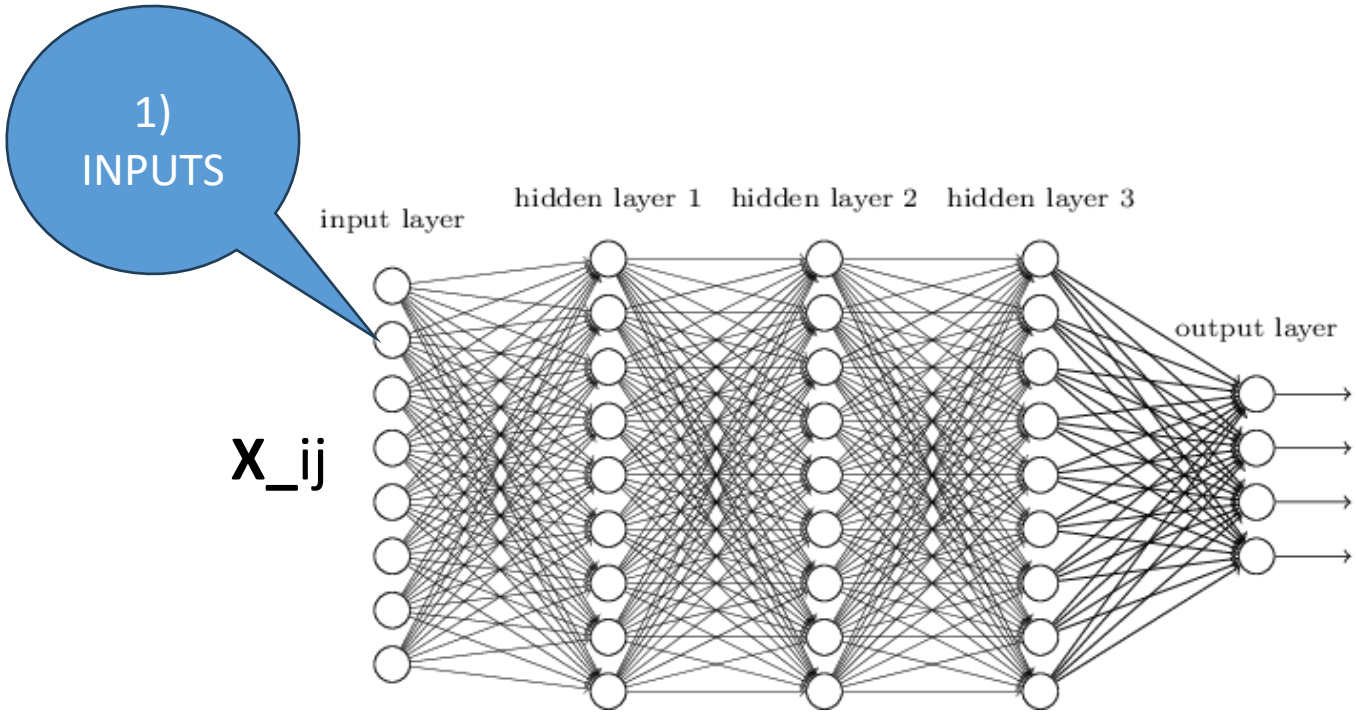
Design your features!



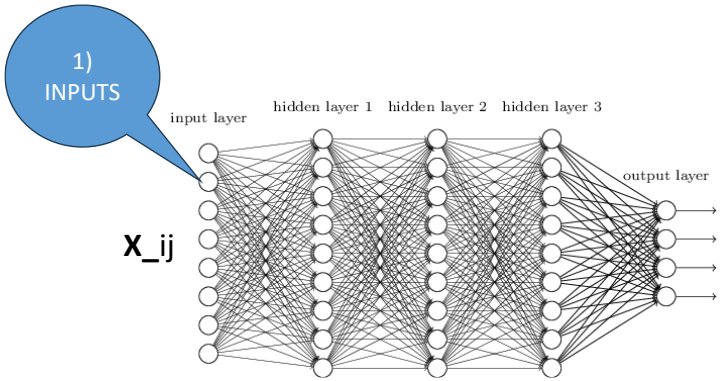
Where to probe/analize



Where to probe/analize



1) INPUTS (X_ij)



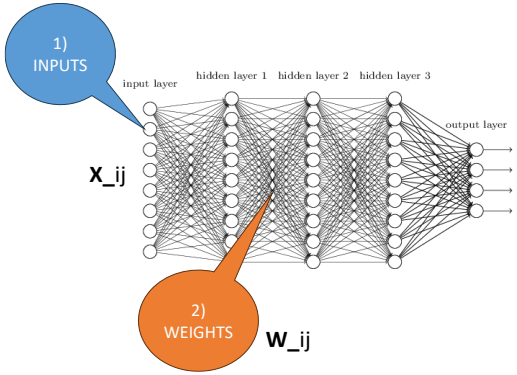
What to Analyze:
Distribution, outliers, correlations, dimensionality, and sparsity of input features.

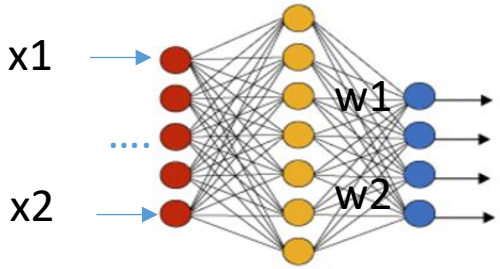
Why Important:
Garbage-in, garbage-out. Poor input quality limits learning. Detecting redundant, noisy, or missing features helps improve preprocessing and generalization.

How to Analyze:
•Statistical metrics (mean, std, min, max)
•PCA or t-SNE for structure
•Correlation matrices
•Check for class imbalance

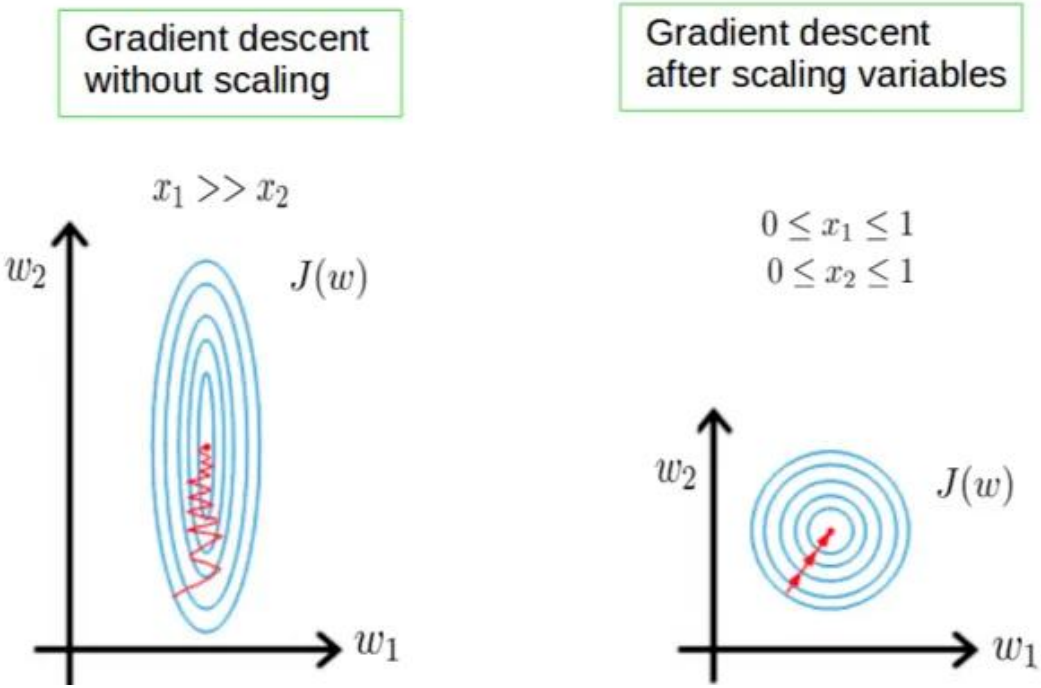
Visualization Tools:
•Histograms, scatter plots, boxplots
•Pairwise correlation heatmaps
•t-SNE/UMAP for 2D embeddings

Scaling can improve the learning phase



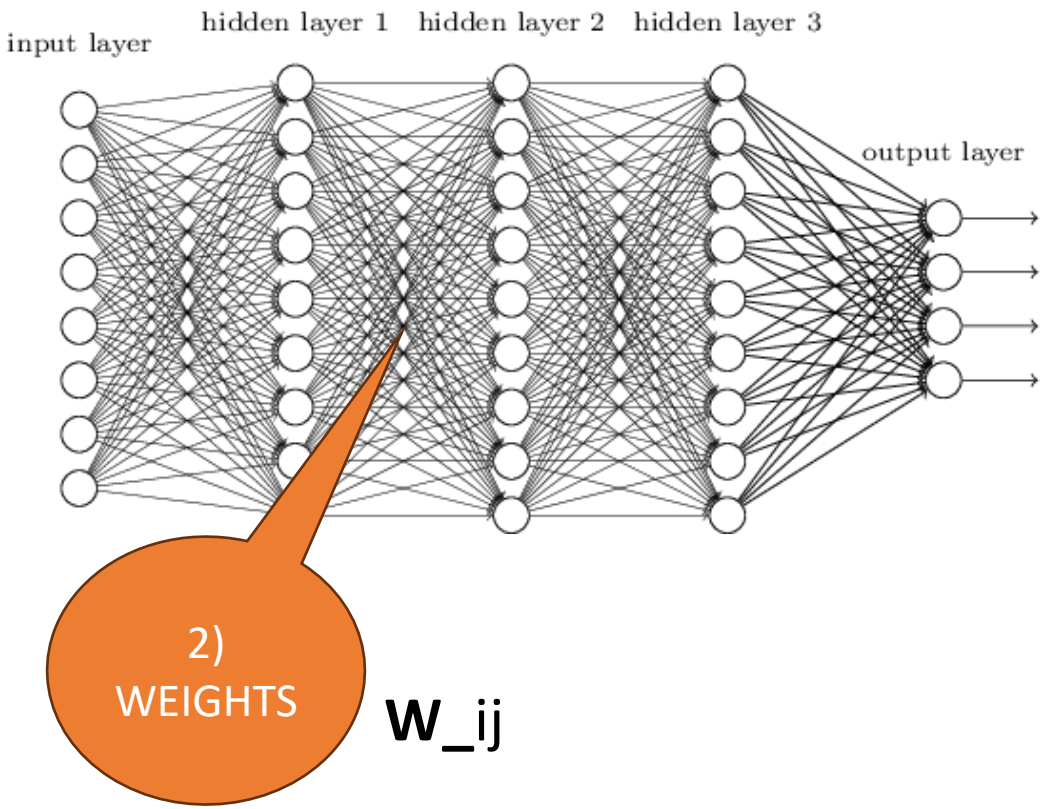


The learning algorithm is minimizing the learning error J() tuning by iterations (gradient descent) the parameter vector W composed by the single weights w1, w2, etc.

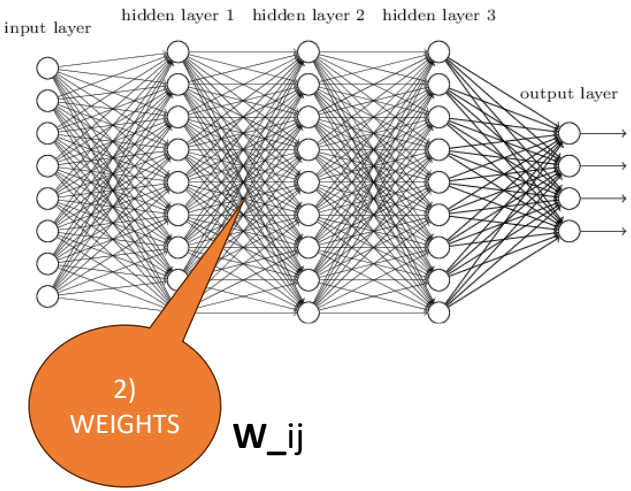


With a proper scaling, less iterations can be required to get the same minimum point in the parameter space during training

Where to probe/analize

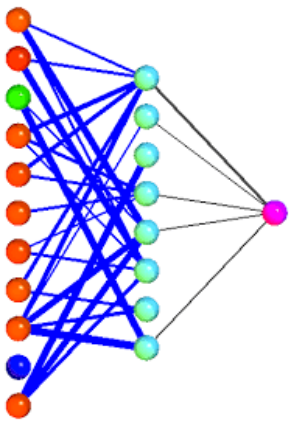


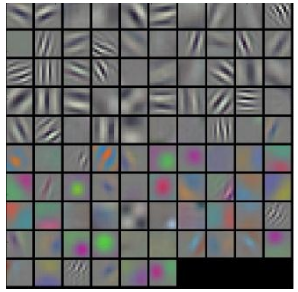
2) WEIGHTS (W_ij)



What to Analyze:
Distribution of weights and gradients across layers.

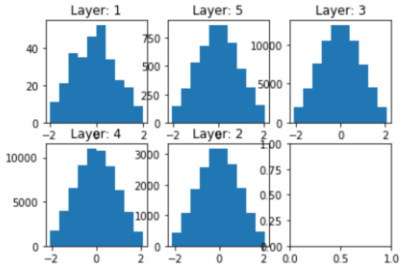
Why Important:
•Detect vanishing/exploding gradients
•Reveal overparameterization or undertraining
•Understand regularization effects (e.g., L1/L2)



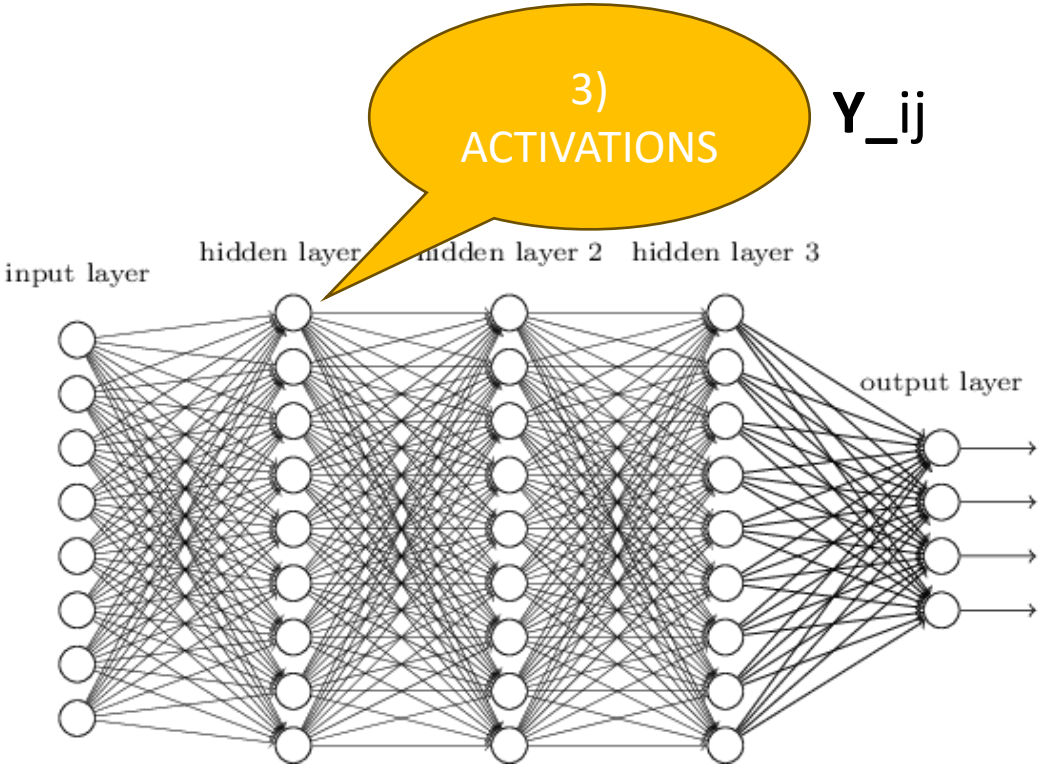


How to Analyze:
•Mean, std, max norm of weights
•Histogram of weights per layer
•Weight sparsity and pruning potential
•Entropy or kurtosis to detect weight collapse

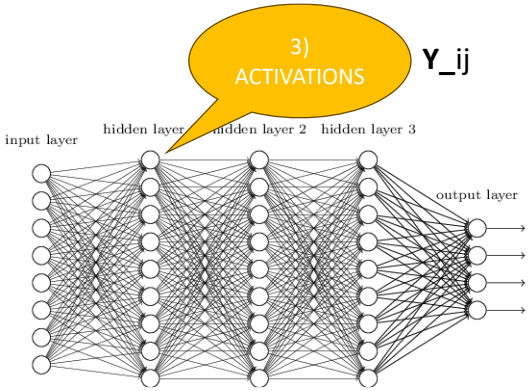
Visualization Tools:
•Weight histograms
•Norm evolution per epoch
•Gradient flow charts (e.g., TensorBoard



Where to probe/analize



3) ACTIVATIONS (Y_ij)

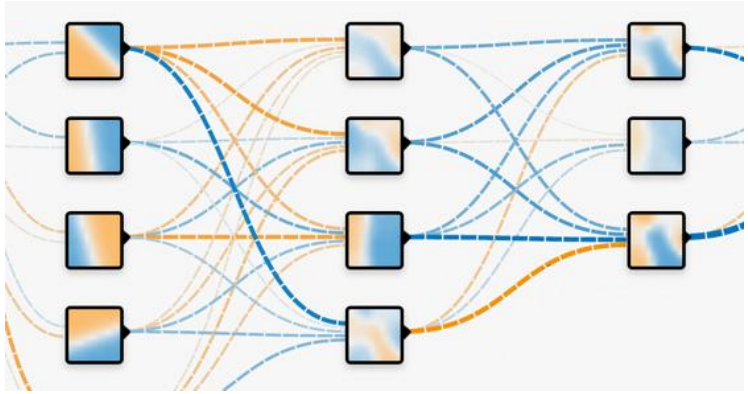


What to Analyze:
Output of each neuron after applying the activation function (e.g., ReLU, sigmoid).

Why Important:
•Detect dead neurons (e.g., ReLU always zero)
•Saturation (e.g., sigmoid close to 0 or 1)
•Layer dynamics and information propagation

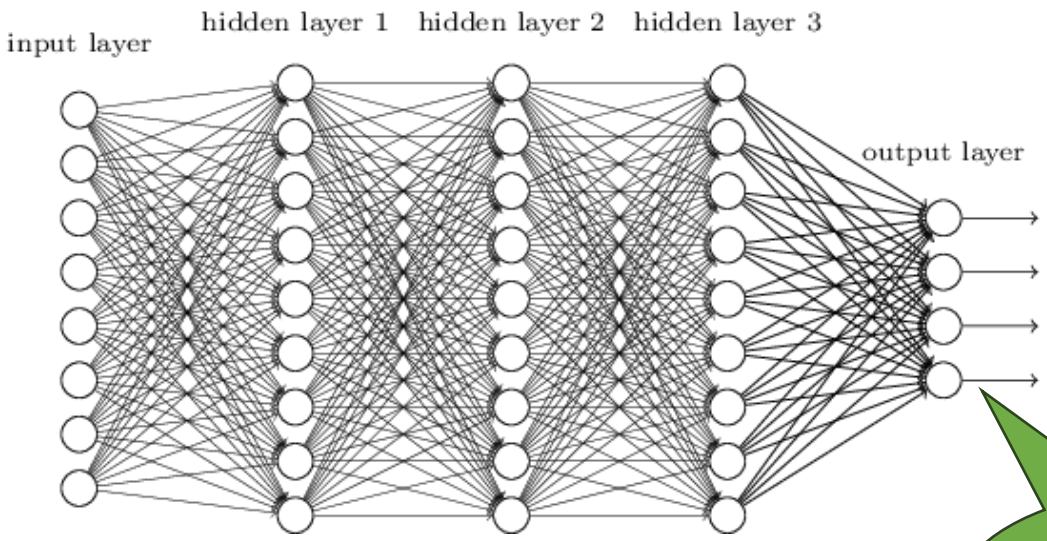
How to Analyze:
•Activation histograms
•Entropy of activation distribution
•Probing with specific inputs
•Layer-wise mean activation

Combination of activation and weights



Visualization Tools:
•Heatmaps
•Violin plots
•Activation dynamics over time/batches

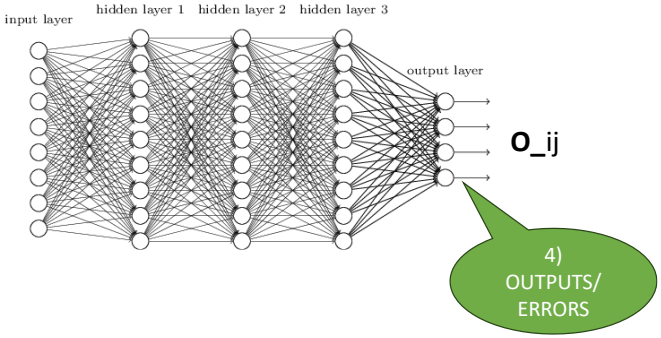
Where to probe/analize



o_ij

4)
OUTPUTS/
ERRORS

4) OUTPUTS (O_ij) Errors



What to Analyze:
Final predictions and associated errors or confidence levels.

Why Important:
•Guides model performance and diagnostics
•Helps identify systematic bias or overfitting
•Can guide loss weighting or calibration

How to Analyze:
•Confusion matrix
•Per-class accuracy / precision / recall / F1
•Calibration curves
•Error histograms vs confidence

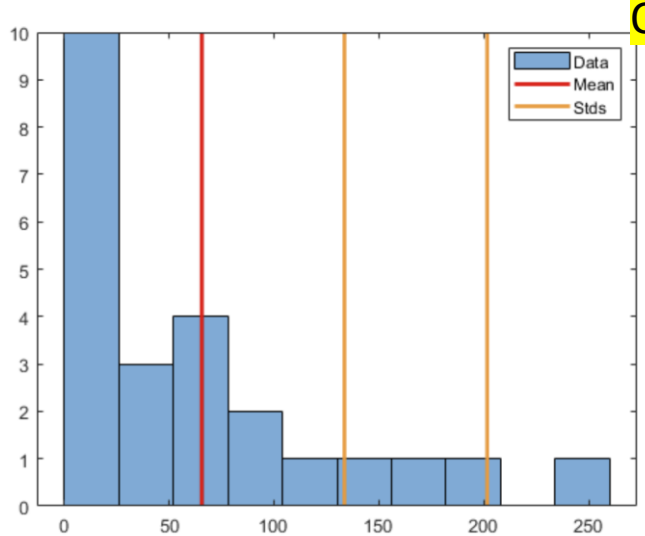
Visualization Tools:
•ROC/PR curves
•Misclassification heatmaps
•Output softmax distribution plots

Homework (not in the exam)
Preprocessing Data Example

https://it.mathworks.com/help/matlab/learn_matlab/data-analysis.html#PreprocessingDataExample-4

BASIC STATS

```
h = histogram(c3,10); % Histogram
N = max(h.Values); % Maximum bin count
mu3 = mean(c3); % Data mean
sigma3 = std(c3); % Data standard deviation

hold on
plot([mu3 mu3],[0 N],'r','LineWidth',2) % Mean
X = repmat(mu3+(1:2)*sigma3,2,1);
Y = repmat([0;N],1,2);
plot(X,Y,'Color',[255 153 51]./255,'LineWidth',2) % Standard deviations
legend('Data','Mean','Stds')
hold off
```



DEALING WITH OUTLIERS DATA

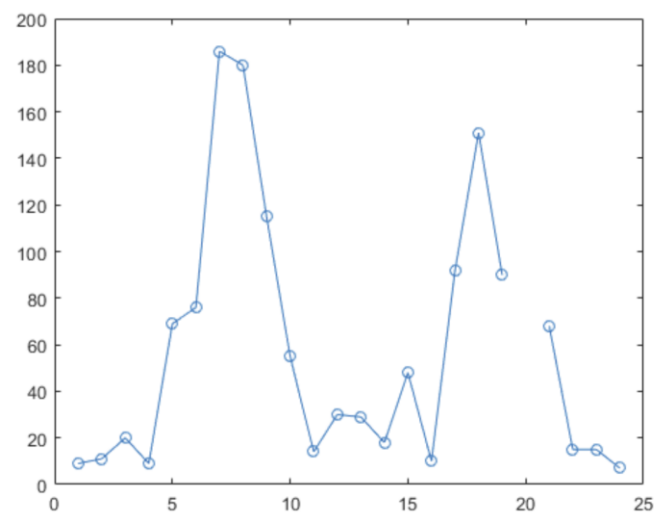
CLIPPING!

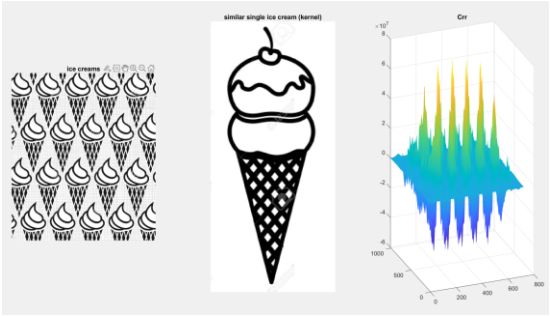
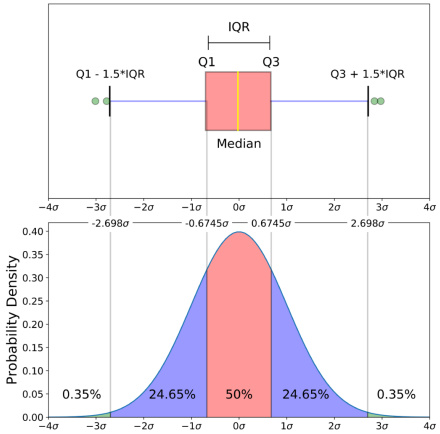
```
outliers = (c3 - mu3) > 2*sigma3;
c3m = c3; % Copy c3 to c3m
c3m(outliers) = NaN; % Add NaN values
```

Smoothing and Filtering

A time-series plot of the data at the third intersection (with the outlier removed in Outliers) resu

```
plot(c3m,'o-')
hold on
```



Main points

- Similarity as a tool for deep learning
- Exploratory Data Analysis
- Feature engineering
  - Data preprocessing: a statistical point of view
- Application of EDA to NN
  - Inputs
  - Weights
  - Activations
  - Outputs