LESSON 8

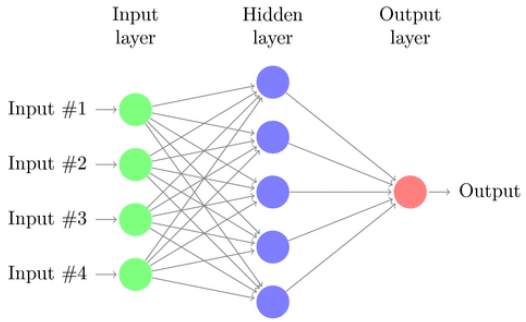
Degrees of freedom/parameters

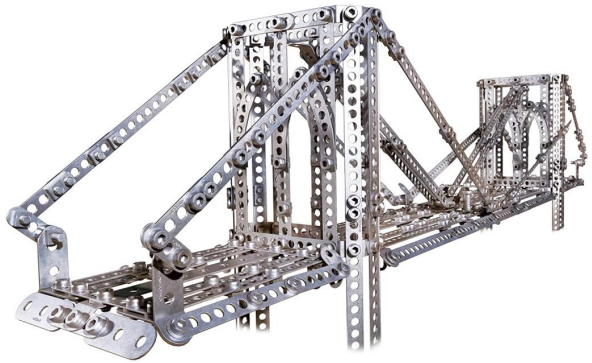
Data Leakage




Outline

- **Number of degrees of freedom/parameters**
- **Data Leakage**
- **Main points**



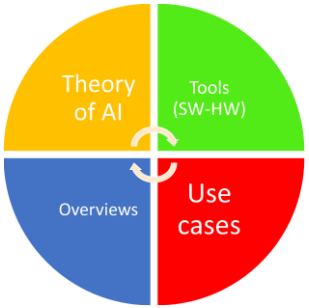


THEORY

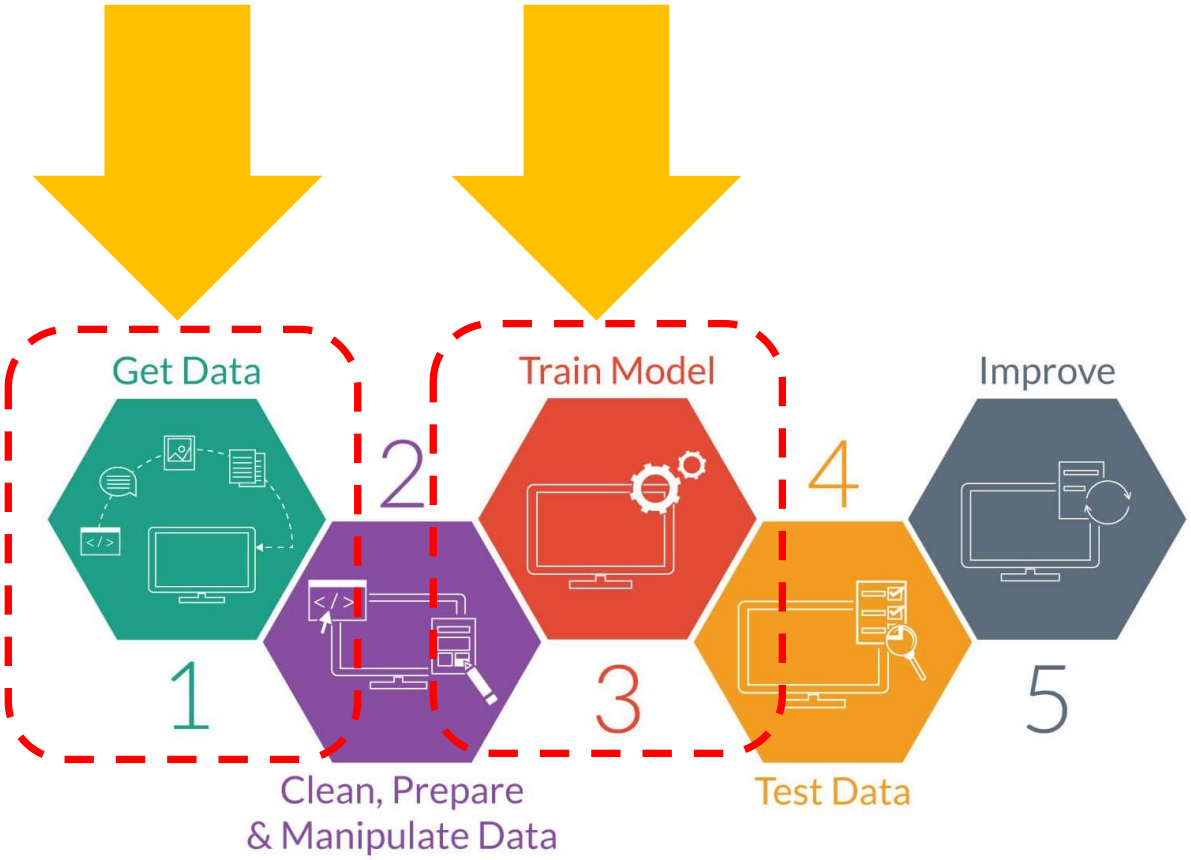
Degrees of freedom /parameters of the models

How much is complex your model?

How much data do you need to configure it?



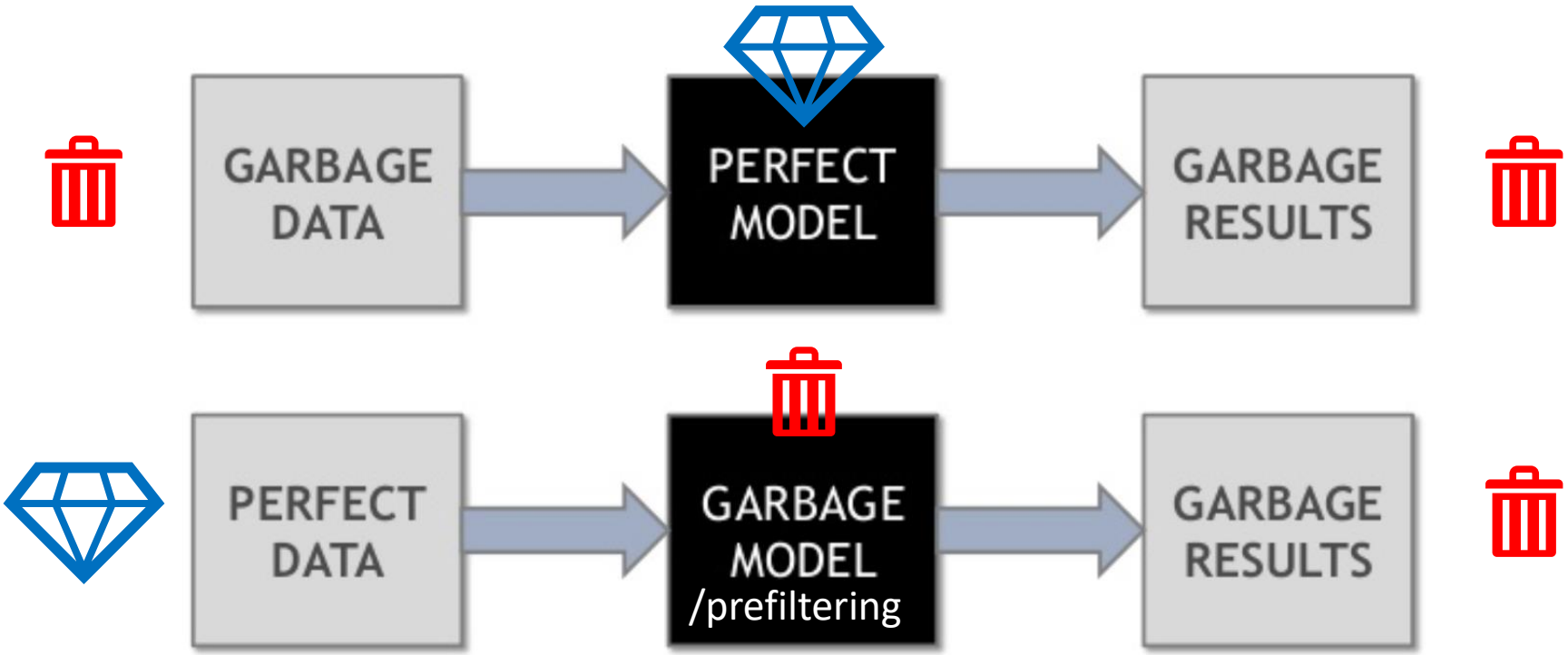
Step 1 and 3 of the ML workflow



GIGO

NO «MAGICAL» NEURAL NETWORK WILL SAVE YOU FROM THIS!!!

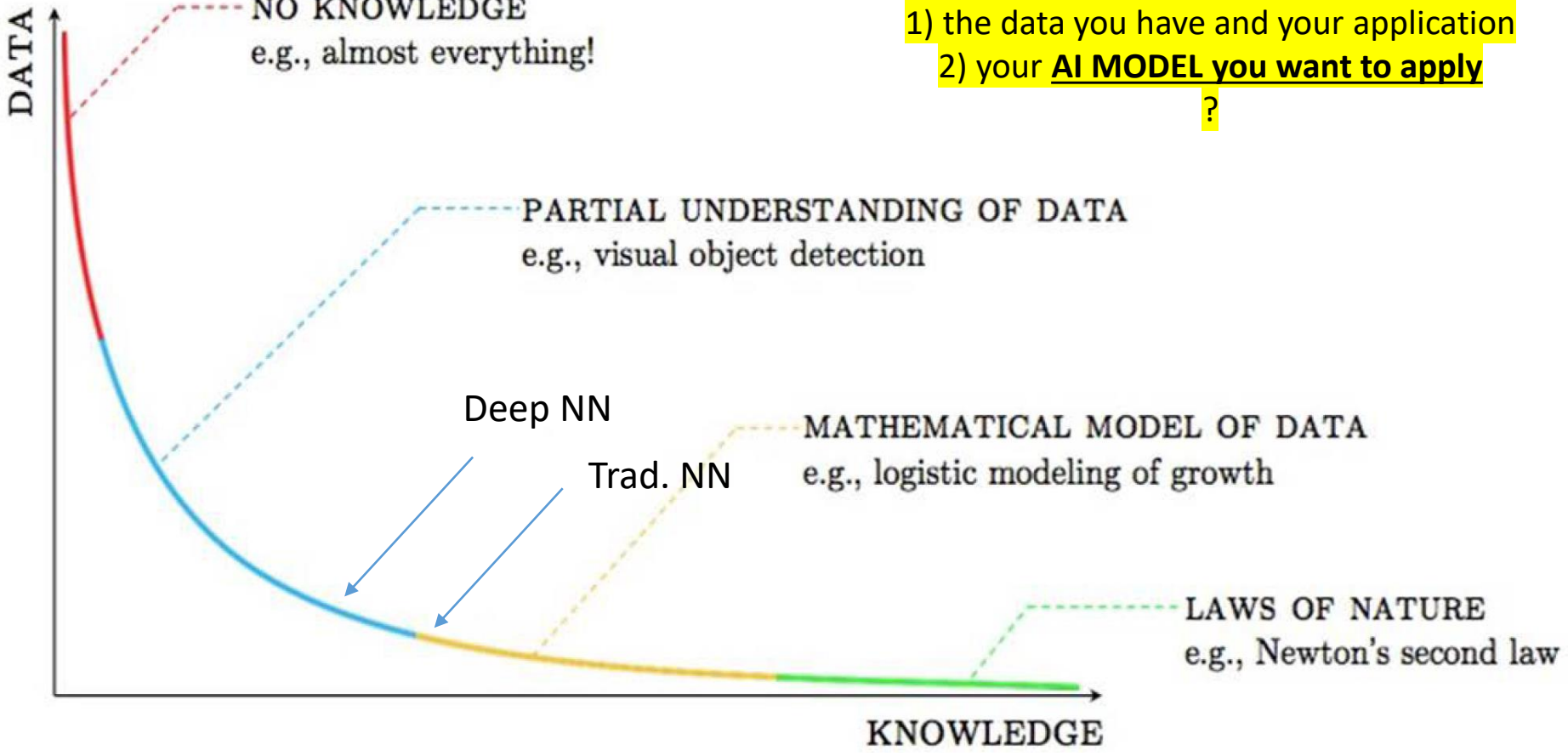
Even worse…



e.g., wrong prefiltering/feature engineering
wrong model choice,
badly trained NN,
overfitted NNs, etc.

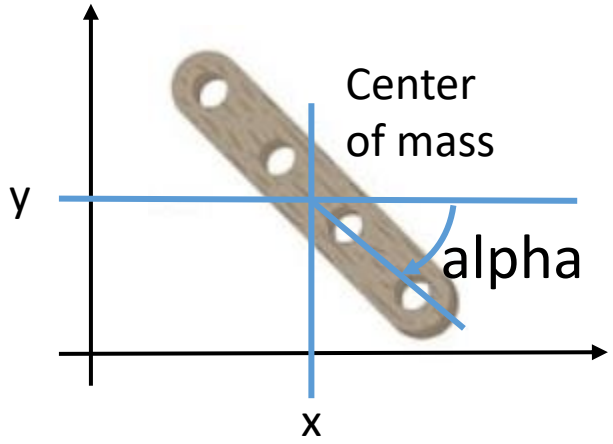
Now is important to remember…. Data knowledge spectrum

What do you really know about
1) the data you have and your application
2) your **AI MODEL you want to apply**
?



How much data?
Degree of freedom/parameters

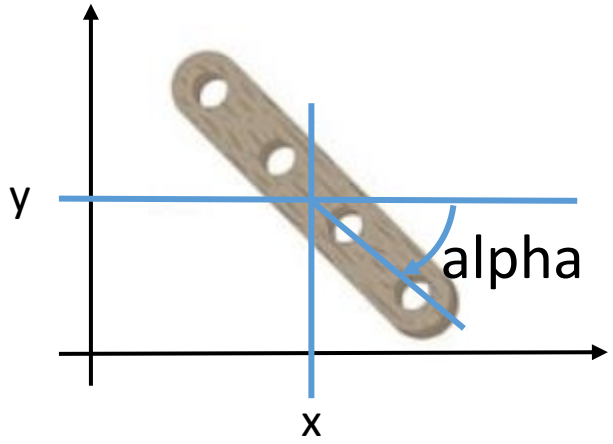
- Imaging a small mechanical component with its Degrees of Freedom (**DoF**) or Number of Parameters (**#Par**) in a 2D space
  - x,
  - y,
  - alpha

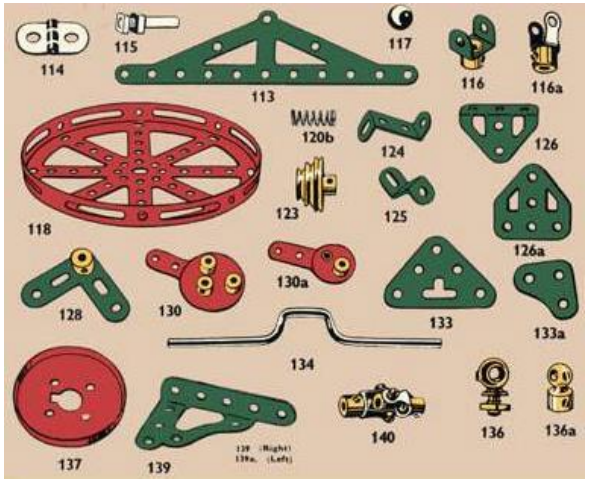


- In **physics**, the degree of freedom of a mechanical system is the number of independent parameters that define its configuration.
- Note: **DoF** is not exactly equivalent **#Par** for complex systems, but they are strongly related.

How much data? DoF/#Par

- Image a small set of components, each of them with its parameters





DoF/parameters (2)

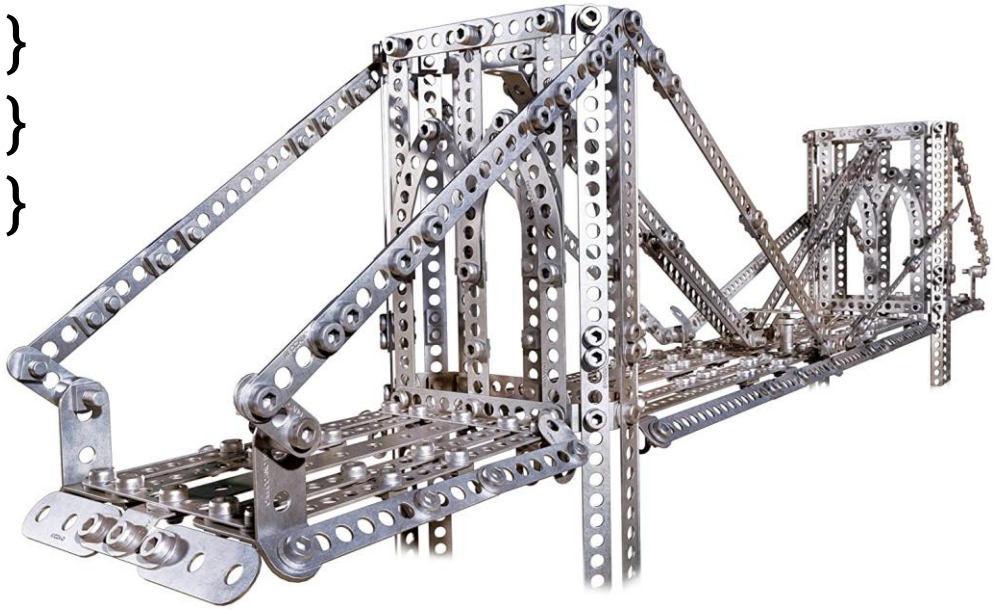
- A mechanical model (like a neural network) is a set of interconnected elements.

  PZ#1= {x1, y1, alpha1 }

  PZ#2= { x2, y2, alpha2 }

  PZ#3= { x3, y3, alpha3 }

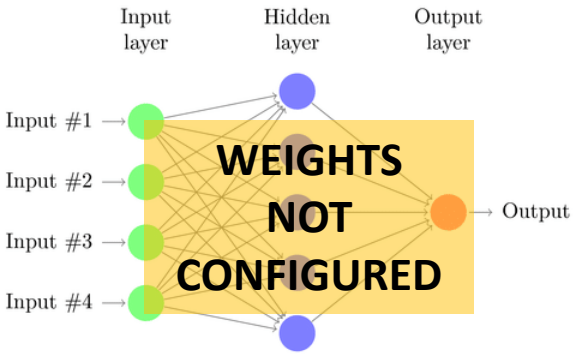
  …



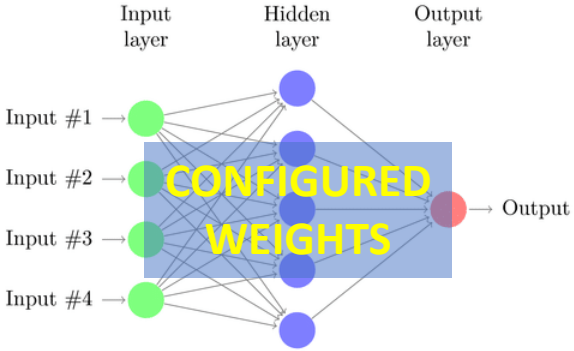
- The model is completed once you placed all elements
  → fixed their parameters

Note:
please neglect the different types of constraints

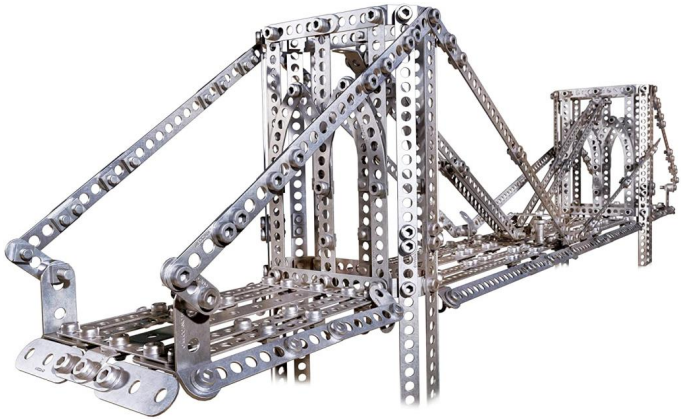
Similitude#1: weights/parameters



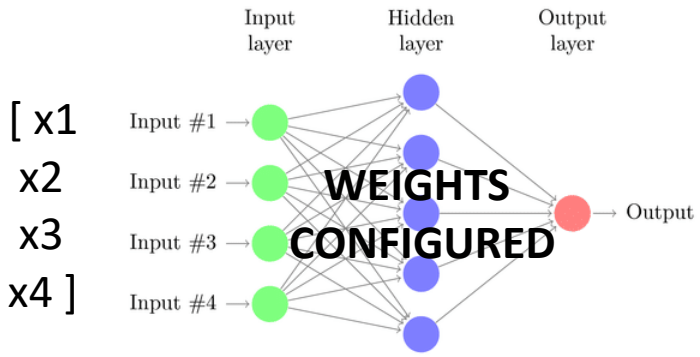
BEFORE TRAINING

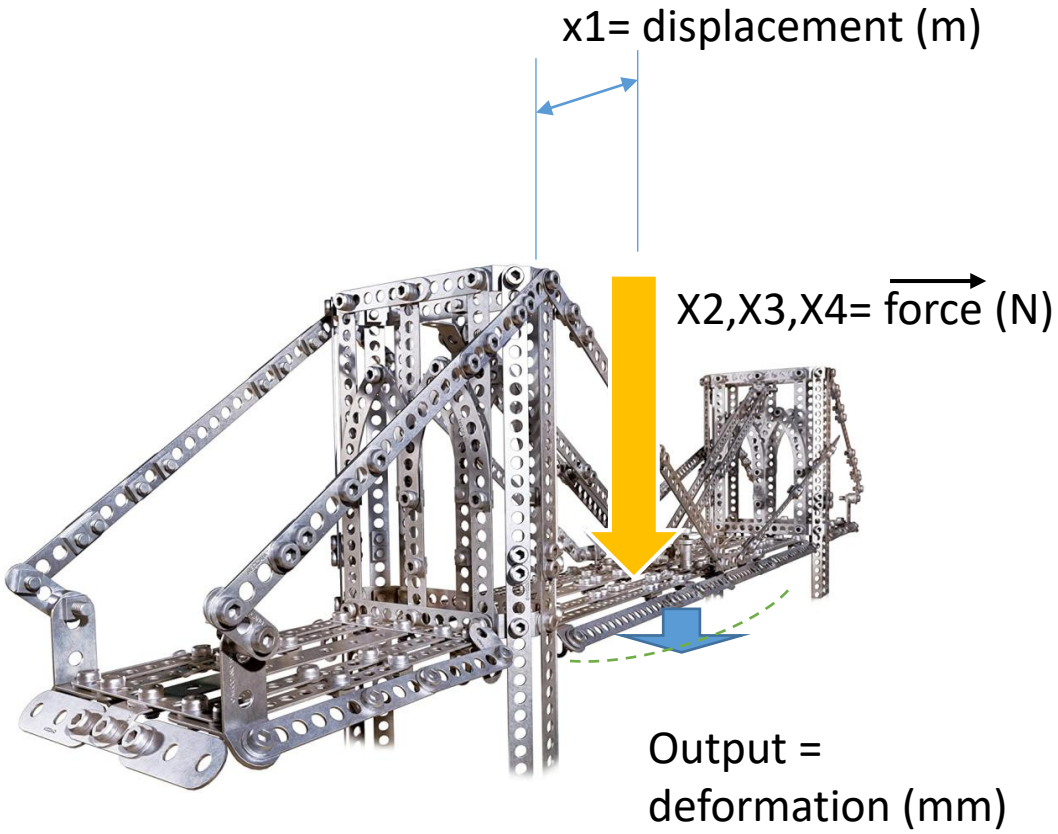


AFTER TRAINING



Similitude#2: inputs and output, behevior

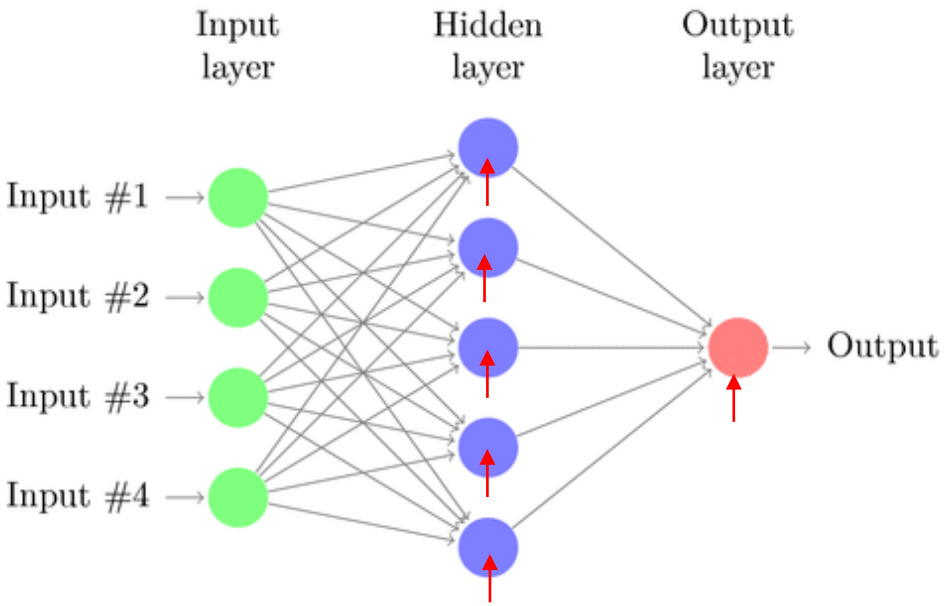




Problem1
Configure weights of the neural network to correctly classify in output the inputs [x1-x4] (minimum error in learning)

Problem2
Configure parameters of the elements to make a minimum deformation in output for all inputs x1 and x2 (minimum deformation in training)

Number of Parameters of NNs

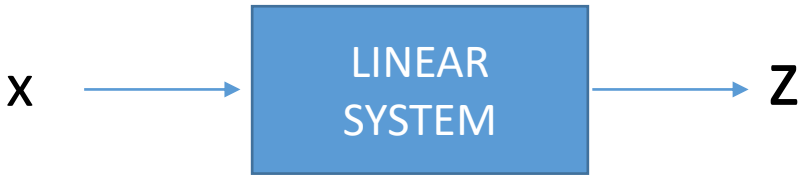


Bias weight

This simple neural network with 1 single hidden layer
has 4x5 (hidden weights) +5 (output neuron) = **25** neuron weights to be fixed
(plus 5+1 bias weight values in the neurons)

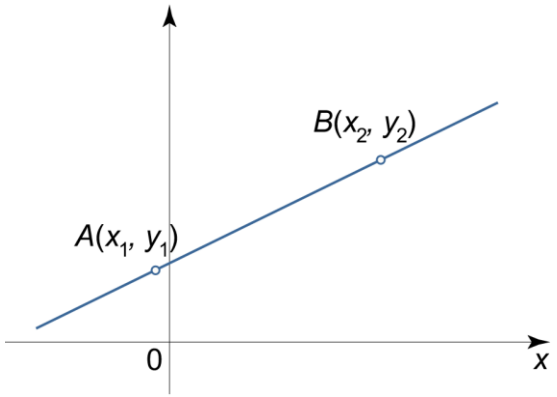
How many input data [x1,x2,x3,x4] are need in the training to fix properly the weights?

How much data?
Example: a 1D linear model



z = alpha x + beta

z = w1 x + b

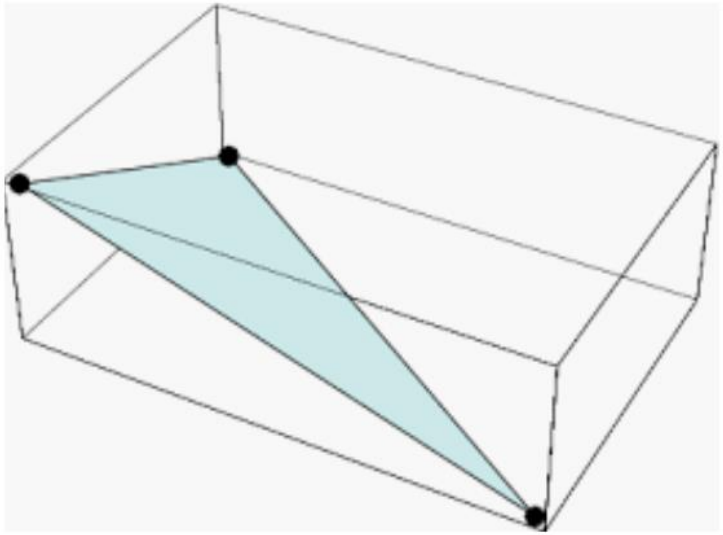


DoF = #Par = 2

To completely describe the model you need to fix 2 parameters: w1, b.

→ You need 2 data points!

How much data?
Example: a 2D linear model



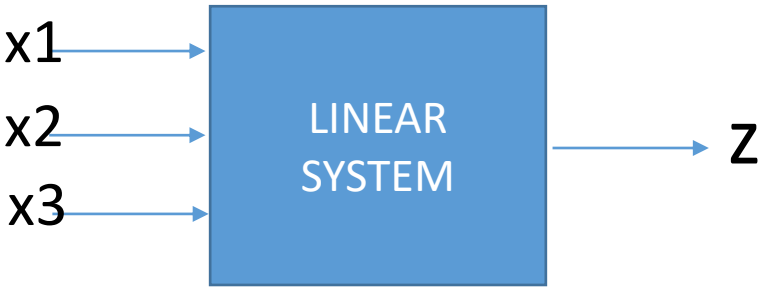
Vectorial form

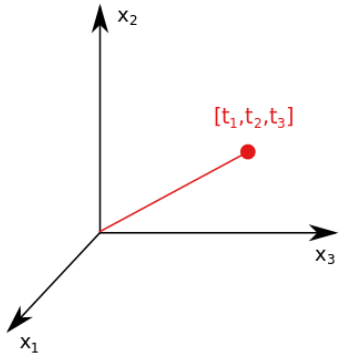
$$z = \mathbf{w} \cdot \mathbf{x} + b$$

$$z = [w_1\, w_2\,] \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + b$$

DoF = #Par = 3
To completely describe the model you need to fix 3 parameters: w1, w2, b.
→ You need 3 data points!

How much data?
Example: a 3D linear model





Vectorial form

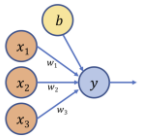
$$z = \mathbf{w} \cdot \mathbf{x} + b$$

$$z = [w_1\ w_2\ w_3] \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + b$$

To completely describe the model you need to fix 4 parameters: w1, w2, w3, b.

→ You need 4 data points!

That is (almost) true also for non linear systems



By the way…. that is (also) the linear output of a neuron

DoF in general

- The degrees of freedom for a given problem are the number of independent problem variables which must be specified to uniquely determine a solution.
- Degrees of freedom =
  # variables - # equations ≈ …

  #Par - #Data

#Inputs = 5

#Data = 10

	A	B	C	D	E	F
1	x_0	x_1	x_2	x_3	x_4	y
2	0.2856096	0.333708	0.8861952	0.3166665	0.388544	0
3	0.3936127	0.9554749	0.5033009	0.48748	0.0252602	1
4	0.6678283	0.4833514	0.8402296	0.2364408	0.2593407	1
5	0.4407239	0.2374104	0.3966326	0.3496683	0.0629024	0
6	0.2622265	0.2117203	0.1967281	0.5731789	0.5462414	1
7	0.7209833	0.357229	0.5069117	0.7213588	0.7241538	1
8	0.1890057	0.1909881	0.0255648	0.7950619	0.8394897	0
9	0.7827472	0.6916136	0.956046	0.4994256	0.4466799	1
10	0.5631773	0.8218485	0.0667687	0.9893122	0.7558692	1
11	0.1593291	0.736723	0.2591398	0.4339377	0.5188055	0

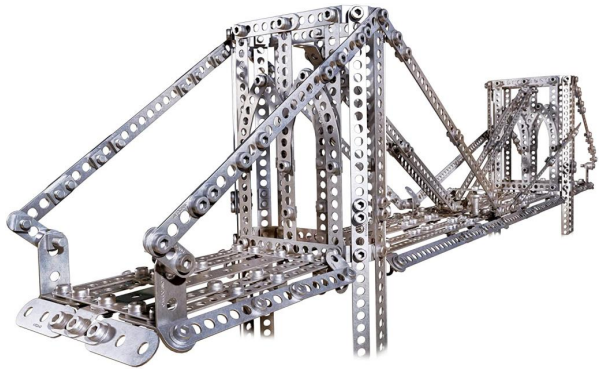
database = [Vectors]

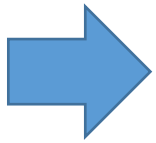
#Data = number of vectors in our database

Note: In complex non linear systems is not so simple, but that is a simple rule of thumb

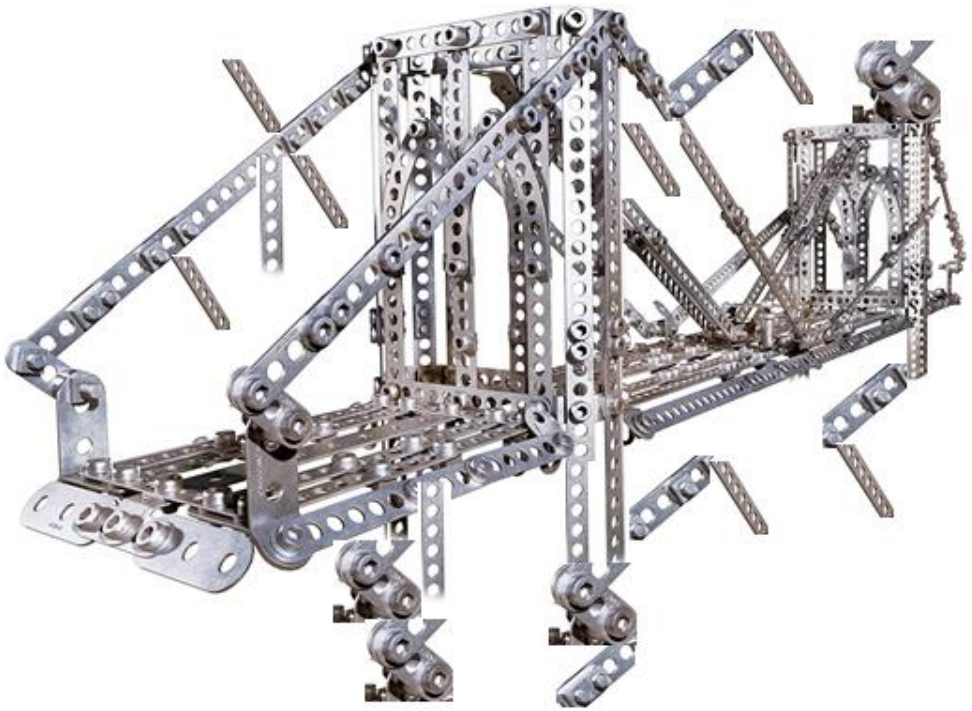
What if… Excessive #Par (elements)

Correct #Par
(after training)





Excessive #Par
(after training)

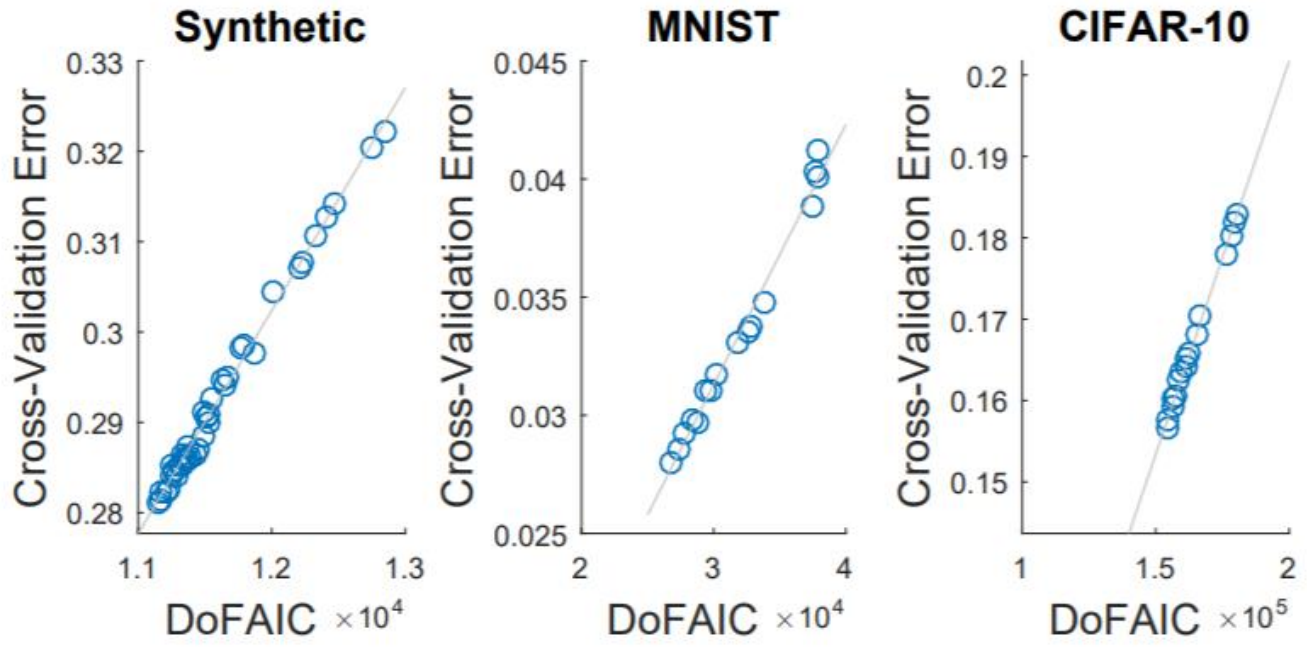


The learning method is not capable to deal with all the elements/parameters you inserted into the model with the given dataset!

The obtained model is **not optimal**, some parts or the model are **useless** or even you can get a **bad behavior**

Relationship Generalization Error VS **DoF** in NNs

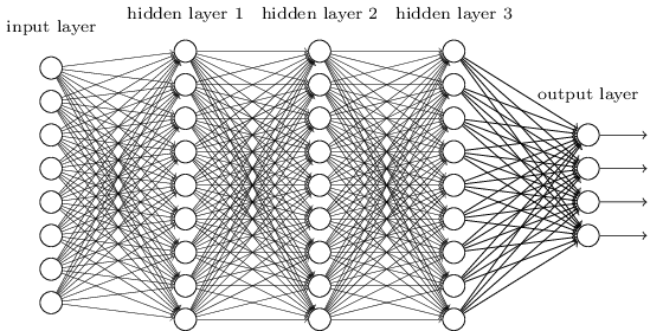
Example of generalization errors of deeplearning models
in large standard classification datasets



If you leave too much «free» weights (lack of data with respect to #Par) in the network the generalization capability will tend to be poor

Training large and deep learning networks

Trad. Feedforward NN



<300 parameters

Corr. To an image of 17x17 32bit/gray_level pixels

…Or in mm^2 a normal stamp 

VGGNet (2014)

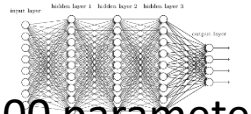
138 million parameters

Corr. To an image of 11700x11700 32bit/gray_level pixels

…Or in mm^2 a beach volley field

#Par: an intuitive perspective
1Par=1pixel

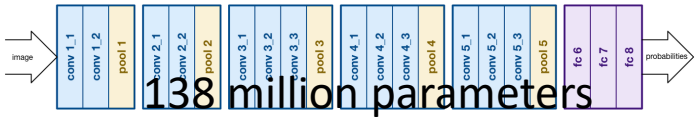
Trad. Feedforward NN



<300 parameters

8bit 17x17 image

VGGNet (2014)

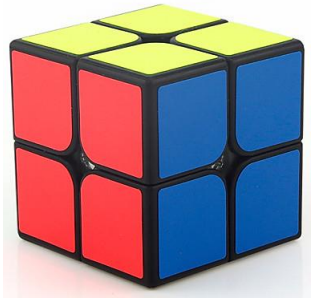


138 million parameters

8bit 10k image < 100Mpar

In brief…

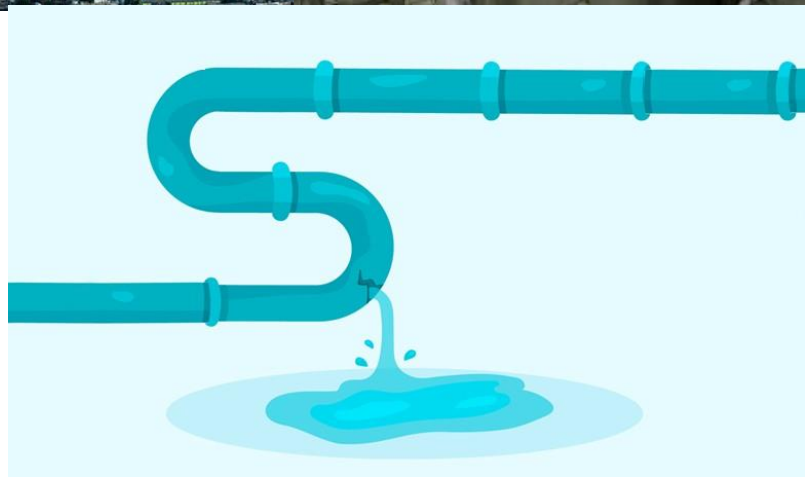
- The #Par of the model (e.g., neural network) must be carfully tuned according to
  - the size of the datasets
    - Number of vectors, Number of inputs
  - its complexity
    - Similar images? Very different examples?



- «Go deep» only if it is really necessary

  Applications of the Occam's razor

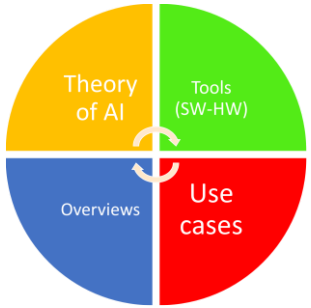


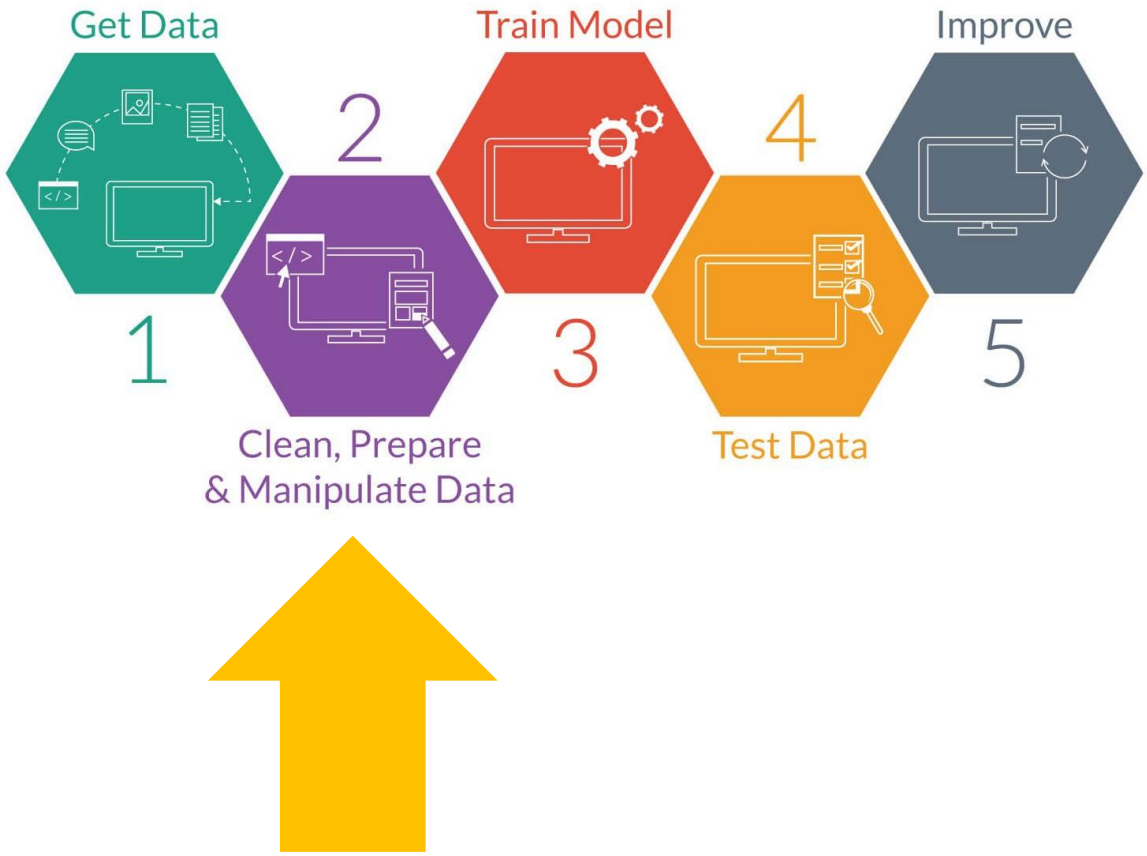
THEORY

Data leakage

One of the most relevant and limiting factor



Step 2 of the ML workflow



Data Leakage in Machine Learning

- Data Leakage is responsible for the cause of invalid Machine Learning/Deep Learning model due to the over optimization of the applied model.

I DID A VERY GOOD LEARNING!! I'M GOOD AT!!

DAMN!! IT OVERFITTED!!!

Data Leakage in Machine Learning (2)

Two main topics

A. Missing relevant features

B. Adding something more….

Health_0001.dat

```
Healthy
0.12
3,14
Bla bla bla
bla bla
```

Ill_0001.dat

```
Cancer
0.12
4,14
Bla bla bla
bla bla
```

A

Data Leakage in Machine Learning (2)

- Missing relevant features
  - For example, when we want to use a particular feature for performing **Predictive Analysis**, but that specific feature is not present at the time of training of dataset → data leakage will be introduced within the model.
  - Example:
    you want to add to your dataset
    the concentration of OrmonX to predict CancerZ but ….
    OrmonX is not (almost) present in the trainig dataset.

Data Leakage example #1

- Missing something in learning data

Learning phase (measuring learning accuracy)


Dog


Dog


Dog


Cat


Cat


Cat

- What is wrong about this training/learning dataset?
- Take 1 min. to think about it…

Data Leakage example #1

- Missing something in learning data

Learning phase (measuring learning accuracy)


Dog


Dog


Dog


Cat


Cat


Cat

- Overfitting is probable: the int. system will not learn the real differences between dogs and cats
  - Simplified rules
    - Dogs are yellow
    - Cats have ears up

A

Data Leakage example #1

Good generalization test!

- Missing something in learning data

Learning phase (measuring learning accuracy)


Dog


Dog


Dog


Cat


Cat


Cat

Test phase (measuring generalization accuracy)


Dog


Dog


Dog


Cat


Cat


Cat

A

Data Leakage example #1

- Missing something in learning data

Learning phase (measuring learning accuracy)


Dog


Dog


Dog


Cat


Cat


Cat

Test phase (measuring generalization accuracy)


Dog


Dog


Dog


Cat


Cat


Cat

A

Data Leakage example #1

- Missing something in learning data

Learning phase (measuring learning accuracy)


Dog


Dog


Dog


Cat


Cat


Cat

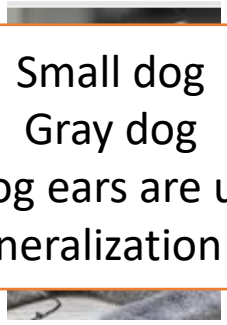
Test phase (measuring generalization accuracy)


Dog


Dog

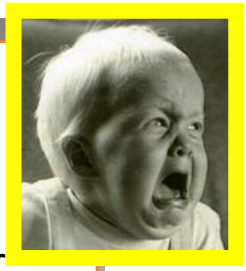

Dog


Cat


Cat


Cat

Small dog
Gray dog
Dog ears are up
→ generalization error



A

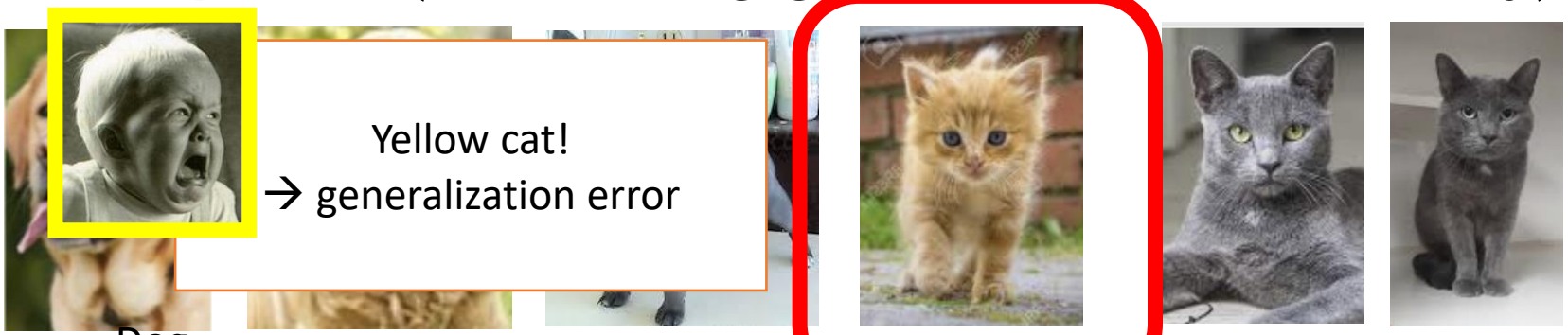
Data Leakage example #1

- Missing something in learning data

Learning phase (measuring learning accuracy)

Dog Dog Dog Cat Cat Cat

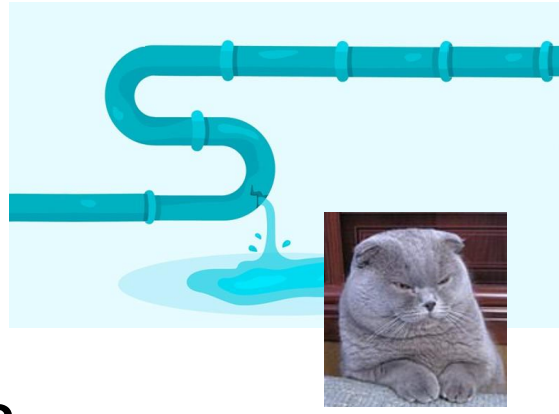
Test phase (measuring generalization accuracy)



Dog Dog Dog Cat Cat Cat

A

Data Leakage example #1

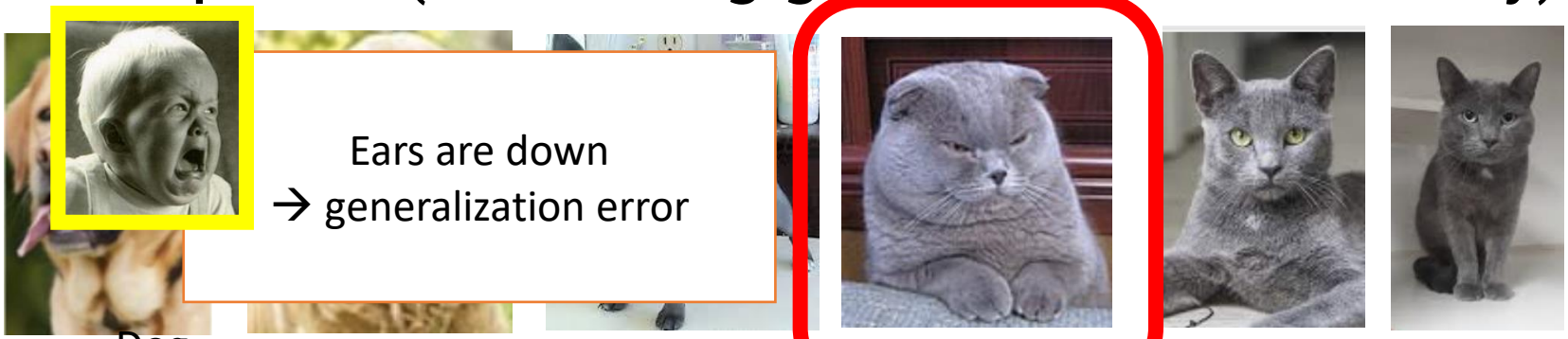


- Missing something in learning data

Learning phase (measuring learning accuracy)

Dog Dog Dog Cat Cat Cat

Test phase (measuring generalization accuracy)

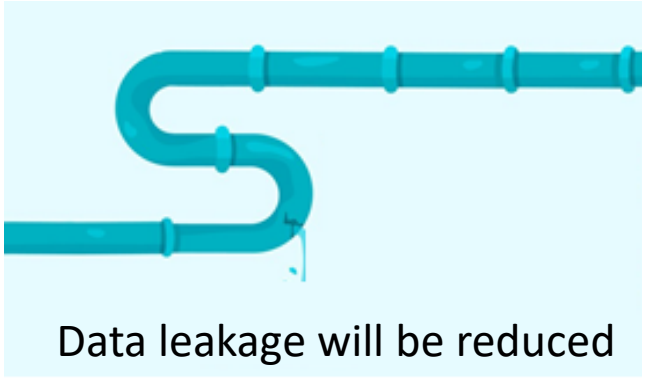


Ears are down
→ generalization error

Dog Dog Dog Cat Cat Cat

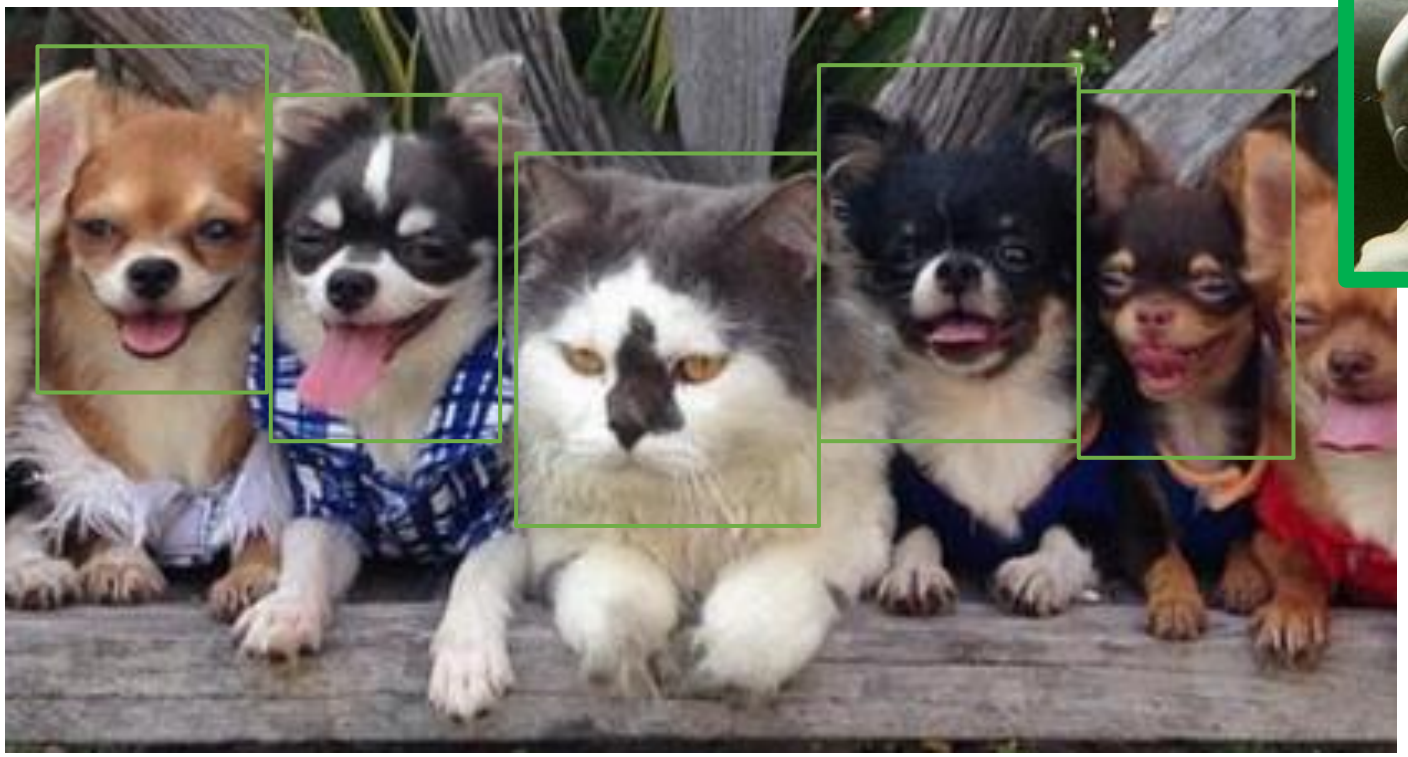
A

Data Leakage example #1



Data leakage will be reduced

You need more good “salient” images to let the intelligent systems understand the what are the important features to distinguish a cat from a dog



B

Data Leakage in Machine Learning

This is one of the most sneaky… probably the first case of critical data leakage

- Adding something more….
  - When information from outside the training dataset is used to create the model.
    - This additional information can allow the model to learn or know something that it otherwise would not know and in turn invalidate the estimated performance of the model being constructed.
  - This additional learning of information by the applied model will disapprove the computed estimated performance of the model.

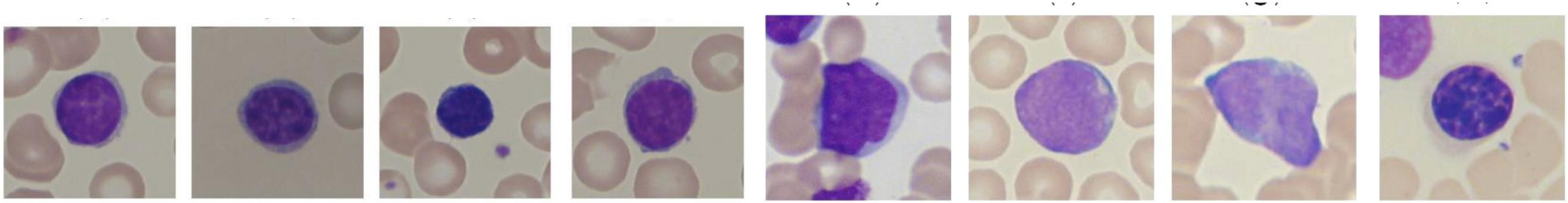
B

Data Leakage example #2

- Something you shouldn't know

Example: good/leukemia white cell images

Learning phase (measuring learning accuracy)



Healty cells from Hospital A

Leukemia cells from Hospital B

What is wrong with the dataset?

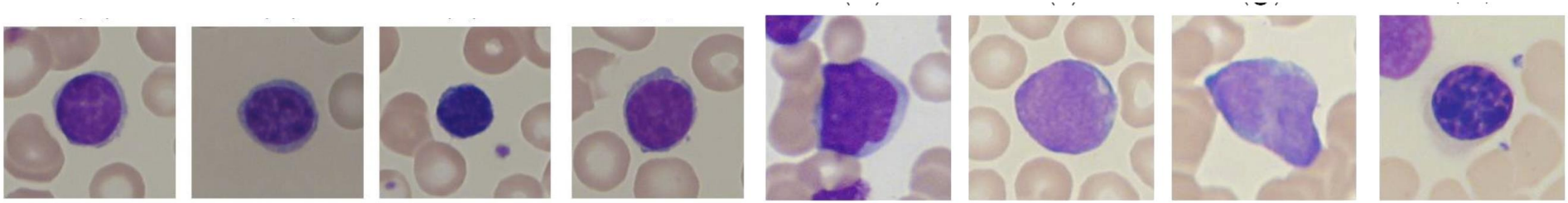
B

Data Leakage example #2

- Something you shouldn't know

Example: good/leukemia white cell images

Learning phase (measuring learning accuracy)



Healthy cells from Hospital A

Leukemia cells from Hospital B

- Images are slightly larger!
- Images are more light
- Noise level is different

The AI model can learn to check only this features of the images (→ Hospital B = cancer) rather than the shapes…

Data Leakage can happen!

The Leakage of data
from test dataset → to training data set

- Data specifically intended to check the generality of the model is spilled in the train dataset….
- Vice versa is slightly better, but still not ideal

Check this portions/chunks of data in Train (T) and Validation-Test (V)

- SLITELY BAD leakage:

[T1, T2, T3]+ [V1, V2, V3] → leakage → [T1, T2, T3]+ [V1, V2, V3, T3]

Some information you designed to be used only in validation, is now in training…

- VERY BAD leakage:

[T1, T2, T3]+ [V1, V2, V3] → leakage → [T1, T2, T3, V1]+ [V1, V2, V3]

Data Leakage can happen!

- Leakage of future data into the past data

- Usage of data outside the scope of the applied algorithm

  Example:
  Dataset of adults → Deployment of a classifier for diagnosis used also for kids

Panel 2: Typical adult reference ranges for thyroid function tests

TSH	0.5–5.5mIU/L
T4	60–135nmol/L
Free T4	9.4–25pmol/L
T3	1.1–2.8nmol/L
Free T3	3.0–8.6pmol/L

Data Leakage can happen!

- In brief, we have two primary sources of data leakage in Machine Learning algorithms:

A. **Feature attributes**
   (variables are saying too much…)

B. **Training data set**
   (chunk of data used in the wrong phase)

Checking the presence of Data Leakage within the applied model

- Data Leakage is observed at the time of usage of complex datasets such as:
  - At the time of dividing time series dataset into training and test, the dataset is a complex problem.
  - Implementation of sampling in a graphical problem is a complex task.
  - Storage of analog observations in the form of audios and images in separate files having a defined size and timestamp.

Data Leakage example #3

Can you see where is the problem?

Learning phase (measuring learning accuracy)


Dog


Dog


Dog


Cat


Cat


Cat

Test phase (measuring generalization accuracy)


Dog


Dog


Dog


Cat


Cat


Cat

C

Data Leakage example #3

Statistical independence is relevant!
The reuse of samples
(or cropped part of them),
or the reuse of the same users,
produces poor generalization

- The “Déjà vu” (warning, it is just a mnemonic reference...)

Learning phase (measuring learning accur

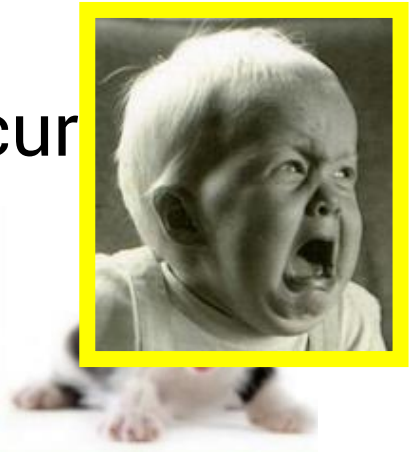

Dog


Dog

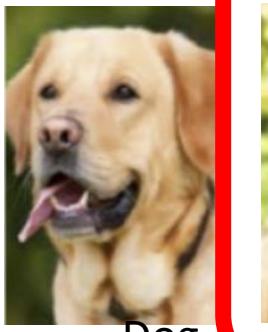

Dog


Cat


Cat


Cat

Test phase (measuring generalization accuracy)


Dog


Dog


Dog


Cat


Cat


Cat

Minimizing Data Leakage

XENONSTACK

- Recalculation of required data using cross-validation
  - Feature Selection
  - Outlier detection & removal
  - Projection Method
  - Scaling of selected features
- Dividing the Dataset
  - Training Dataset
  - Test Dataset

Main points

- Using datasets without previous activities is useless dangerous and it wastes your time
- Number of degrees of freedom /parameters
- **Data Leakage!**