



Formation

# Introduction au Deep Learning

Formation Permanente CNRS  
SARI / DEVLOG  
9 -10 mars

A **thousand thanks** to all those who made this training possible !

In particular :



Service Formation Permanente du CNRS  
Réseaux SARI et DEVLOG

Unité Mixte de Service GRICAD

Centre de calcul du CNRS IDRIS

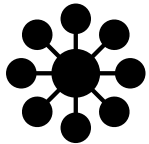
# Objectifs



**Understand** what Deep Learning is,  
its concepts and basics,



Develop a **first experience** on  
simple and ...understandable cases



Learn how to use **tools** and **mutualized  
resources** (Jupyter, GRICAD Mesocentre)

Presentation  
Tour de table

Introduction  
Context, tools and ressources

**1 From the linear regression  
to the first neuron**

**2 Neurons in controversy**

**3 Data and neurons**

Perspectives

End of training, conclusions



# Roadmap

Presentation  
Tour de table

Introduction  
Context, tools and ressources

**1 From the linear regression  
to the first neuron**

**2 Neurons in controversy**

**3 Data and neurons**

Perspectives

## Notebooks :



**Basic  
Regression**  
DNN



**Basic  
Classification**  
DNN



**Hight  
Dimensionnal  
Data**  
CNN



**Sparse  
data**  
Embedding



**Sequences data**  
RNN



**Reinforcement  
learning**



**Variational  
Antoencoder**  
VAE



**Generative  
Adversarial  
Network**  
GAN

 Jupyter

 NumPy

 Keras

 TensorFlow





<https://gricad-gitlab.univ-grenoble-alpes.fr/talks/fidle>

Material courses

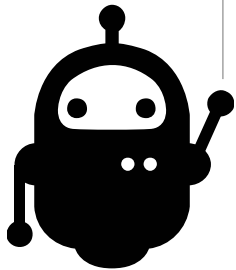




<https://gricad-gitlab.univ-grenoble-alpes.fr/talks/fidle>  
or <http://bit.ly/fidle432>

To connect to the workstations

Logins AGALAN/BIPER

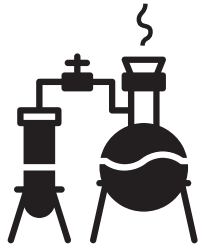


# Introduction

## Context, tools and ressources



1<sup>st</sup> paradigm



Experimental science

2<sup>nd</sup> paradigm

$$i\hbar \frac{d}{dt} |\Psi(t)\rangle = \hat{H} |\Psi(t)\rangle$$

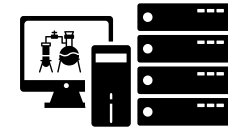
$$\nabla \times H = J + \frac{\partial D}{\partial t}$$

$$F = G \cdot \frac{m_1 \cdot m_2}{r^2}$$

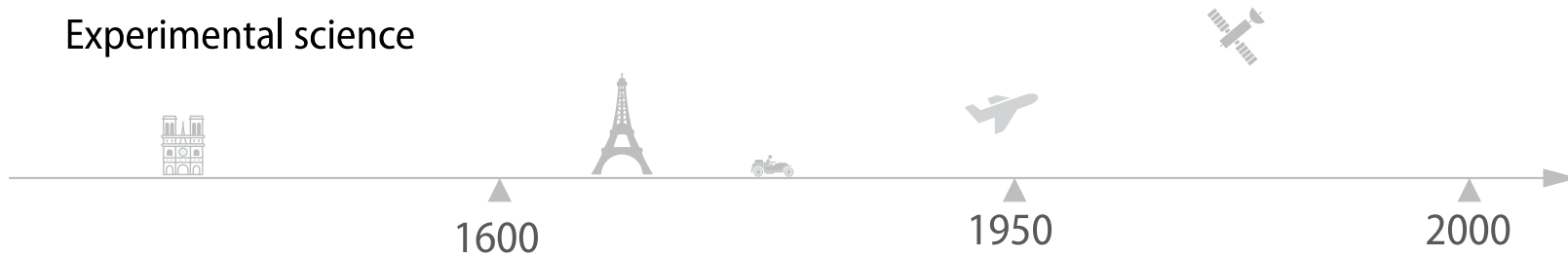
Theoretical science

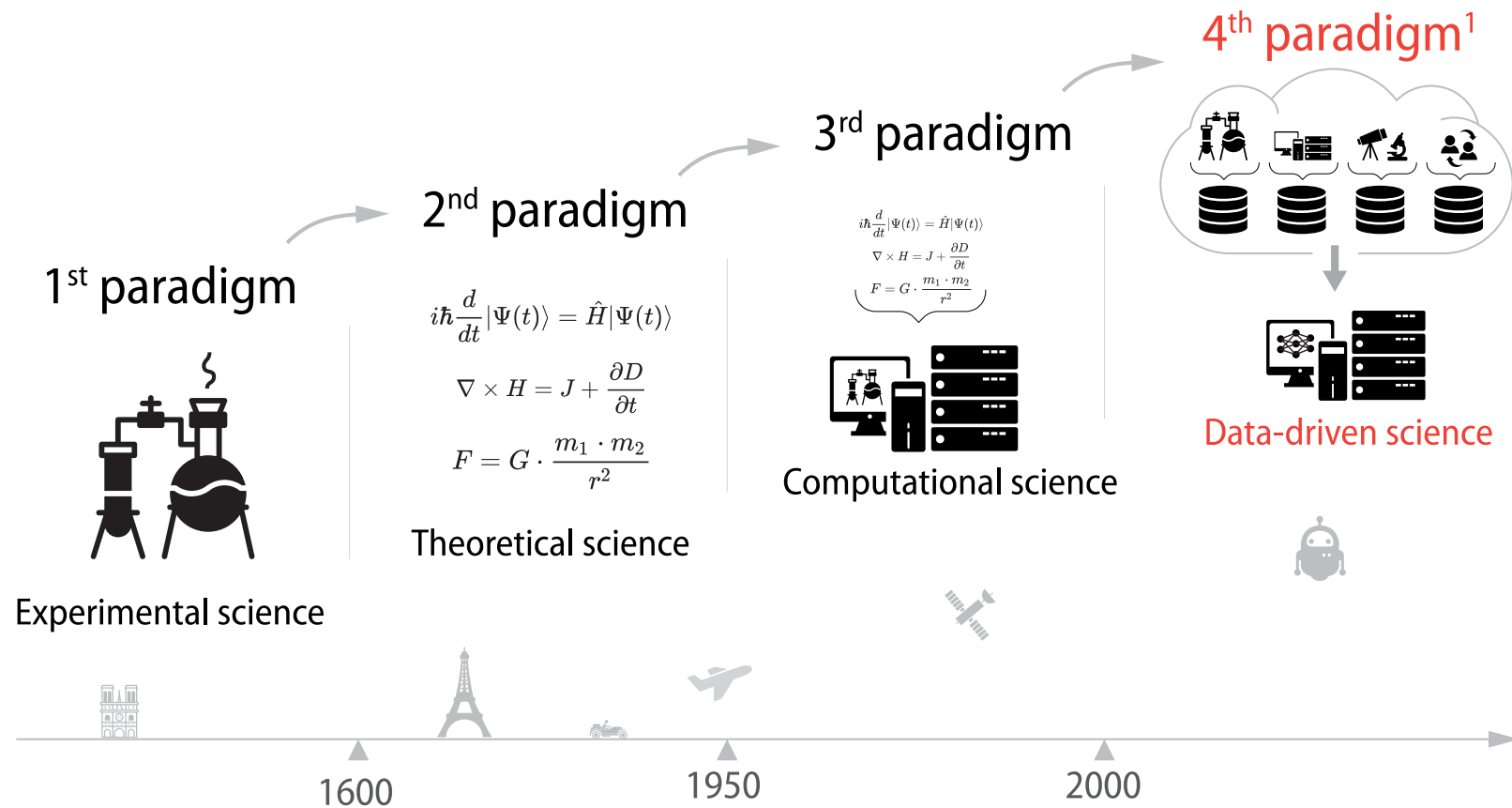
3<sup>rd</sup> paradigm

$$\begin{aligned} i\hbar \frac{d}{dt} |\Psi(t)\rangle &= \hat{H} |\Psi(t)\rangle \\ \nabla \times H &= J + \frac{\partial D}{\partial t} \\ F &= G \cdot \frac{m_1 \cdot m_2}{r^2} \end{aligned}$$



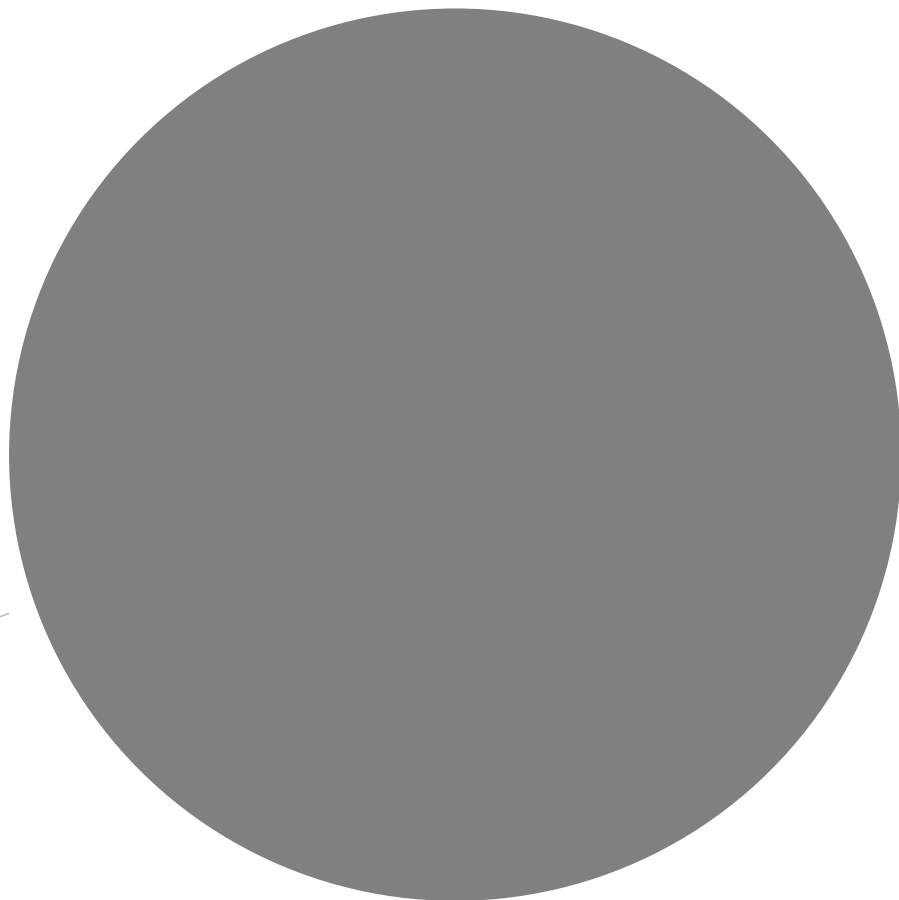
Computational science

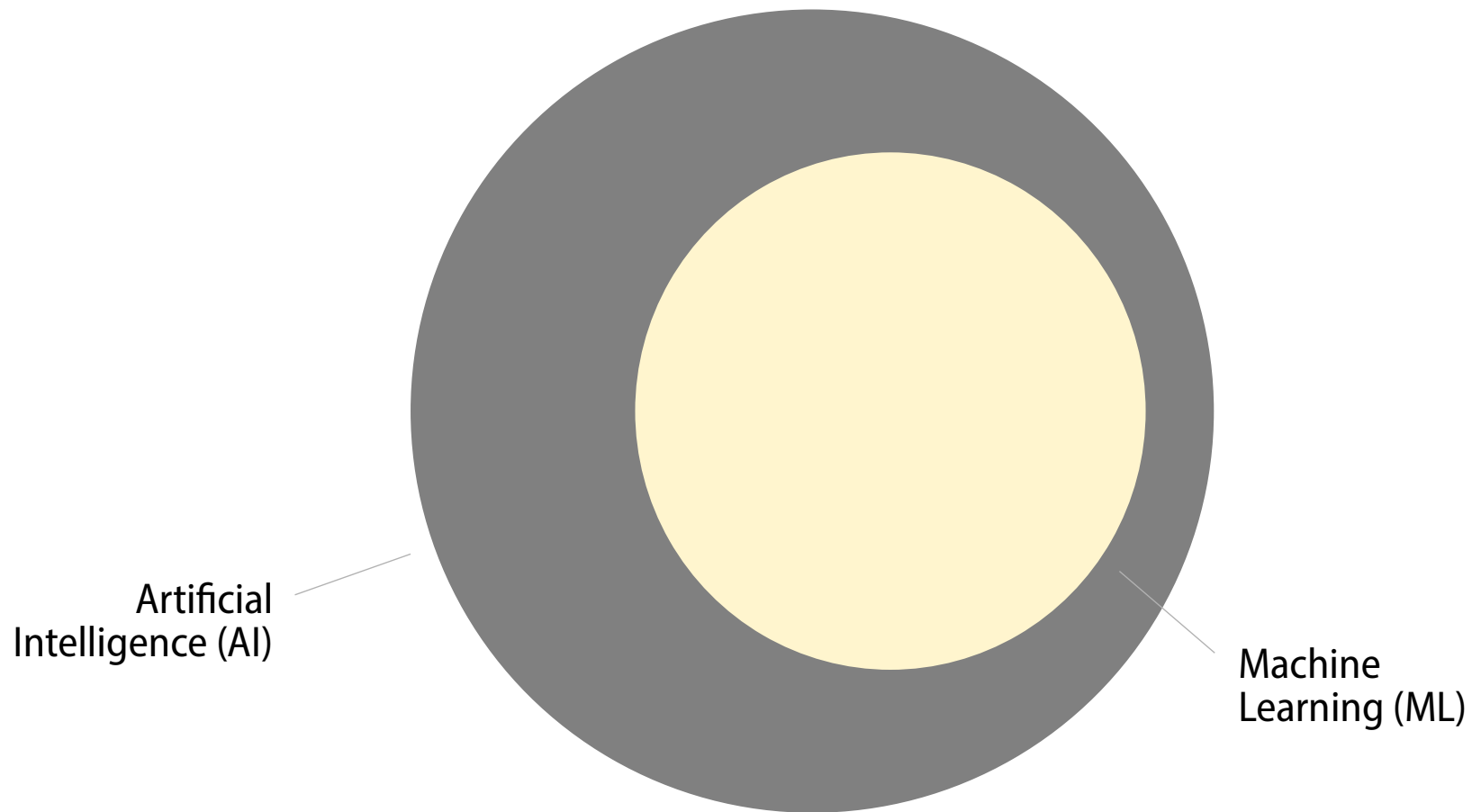




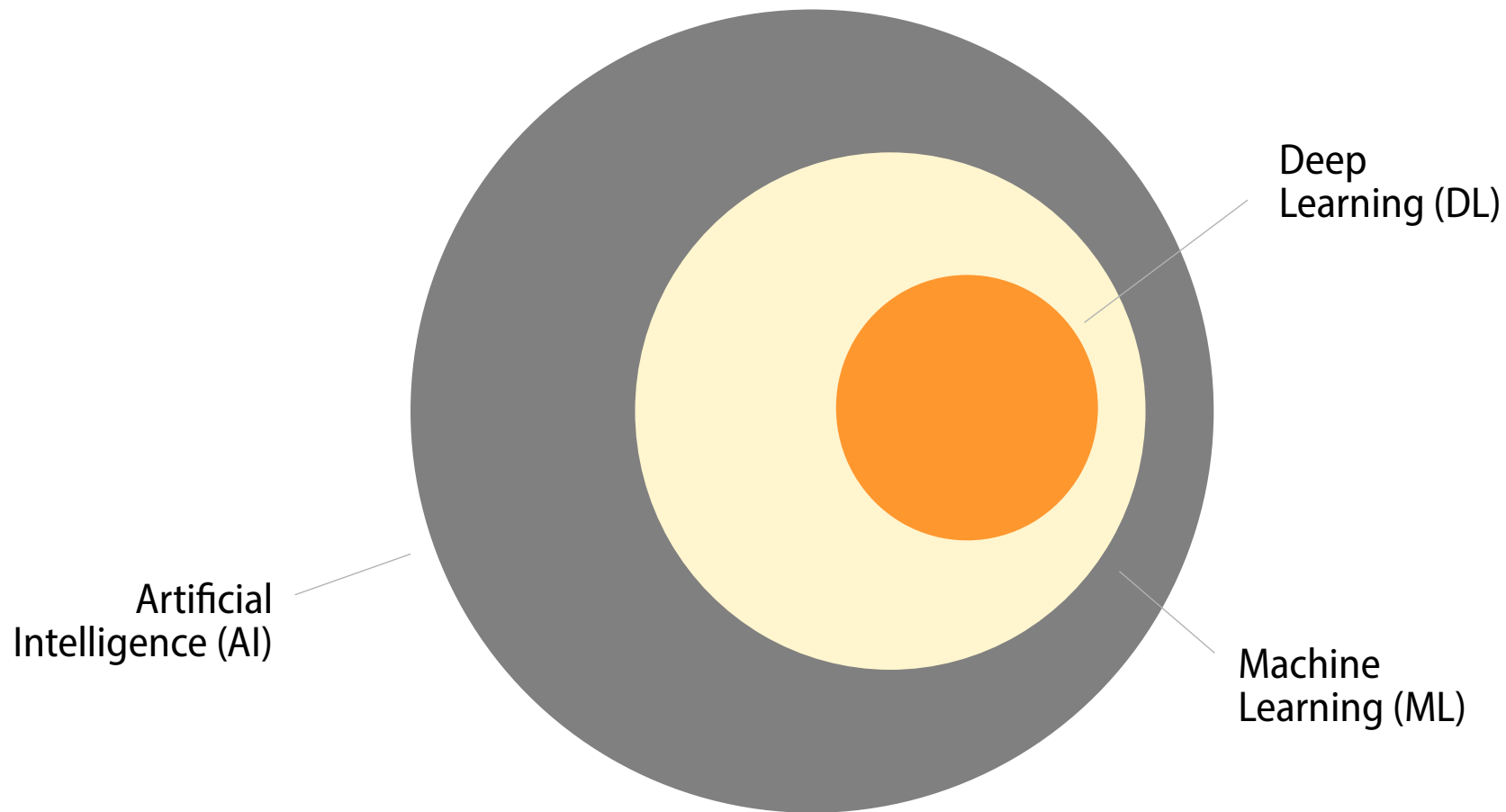
<sup>1</sup> Jim Gray, 2007 [GRAY]

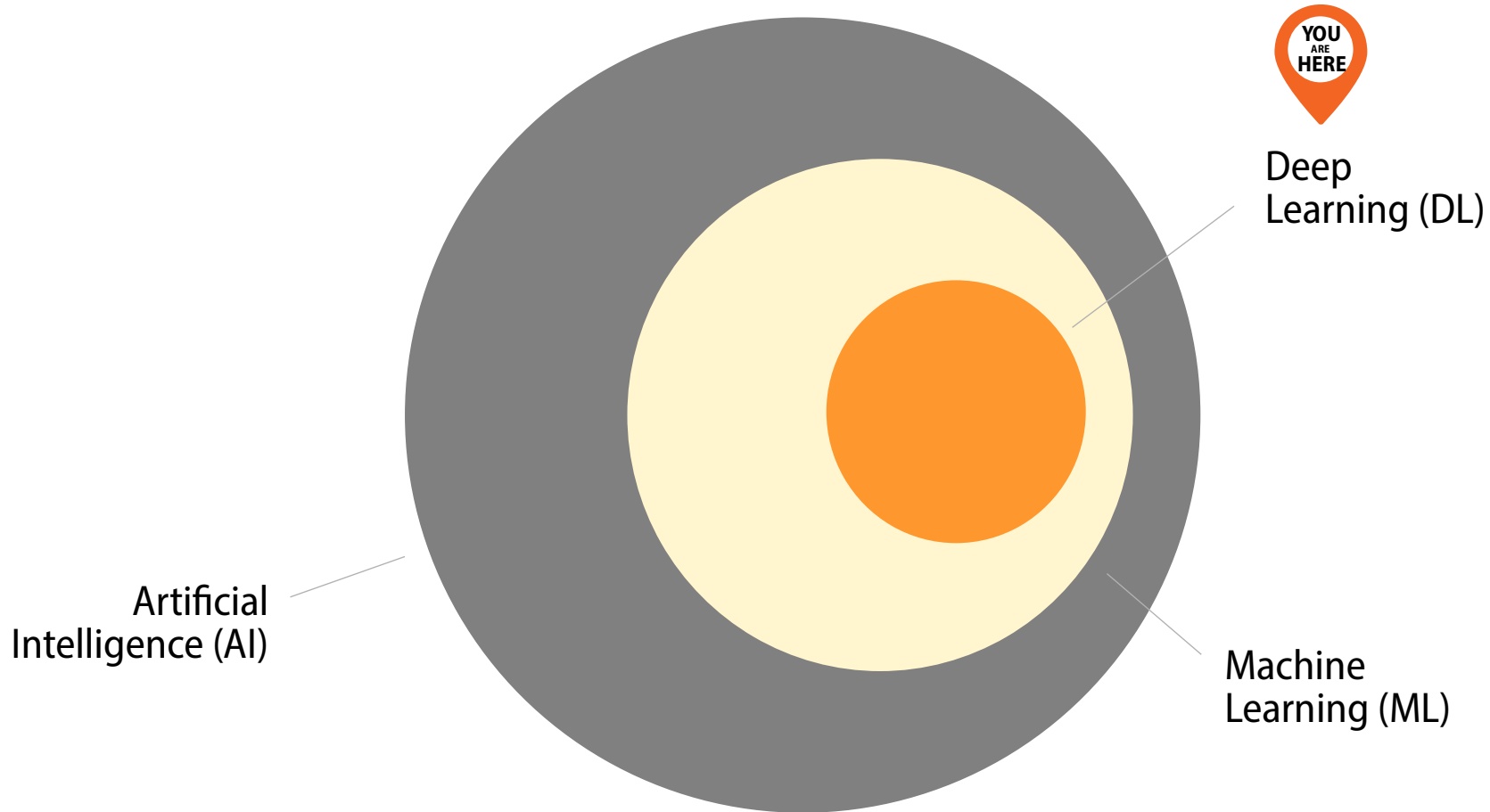
Artificial  
Intelligence (AI)

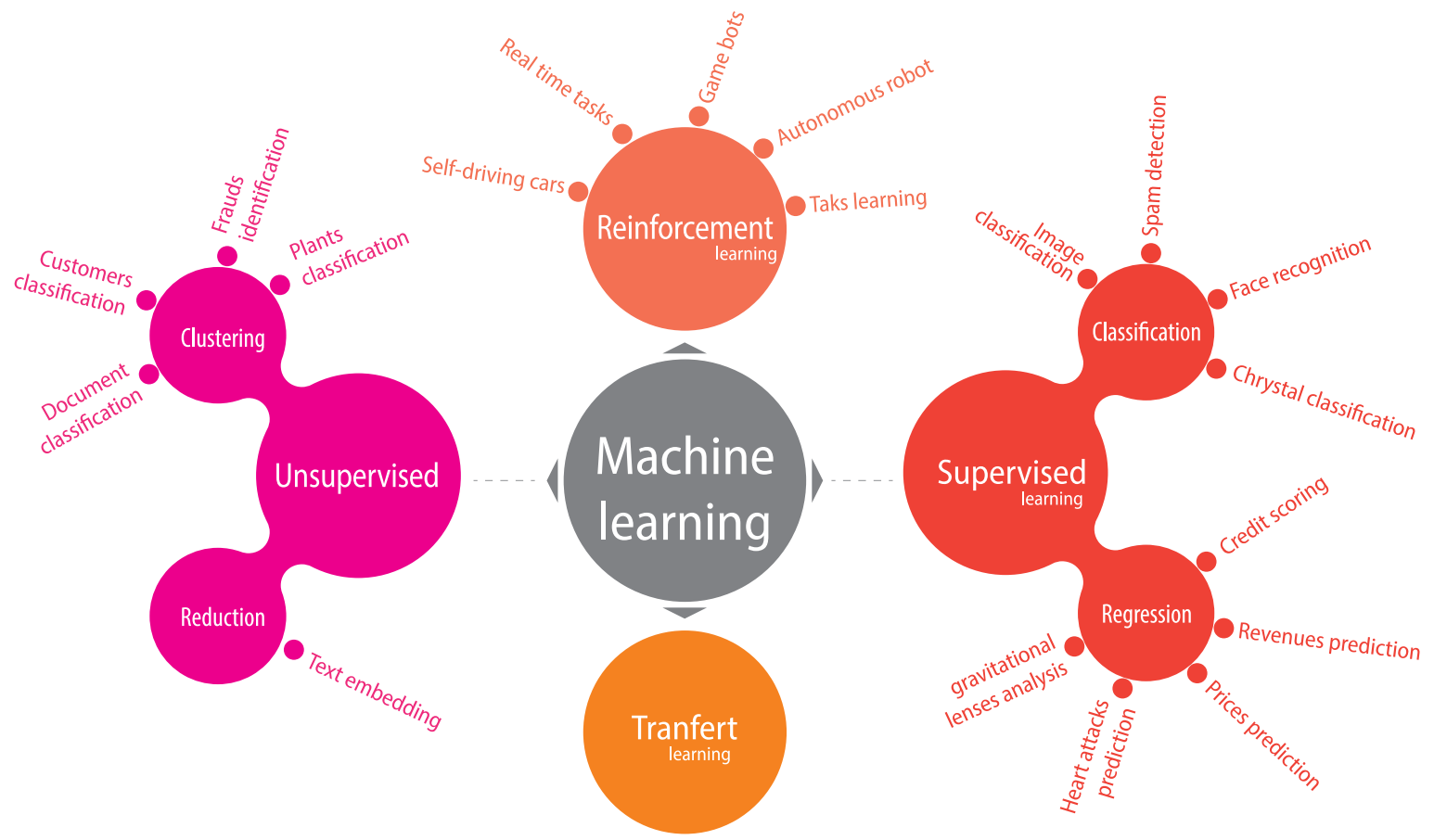


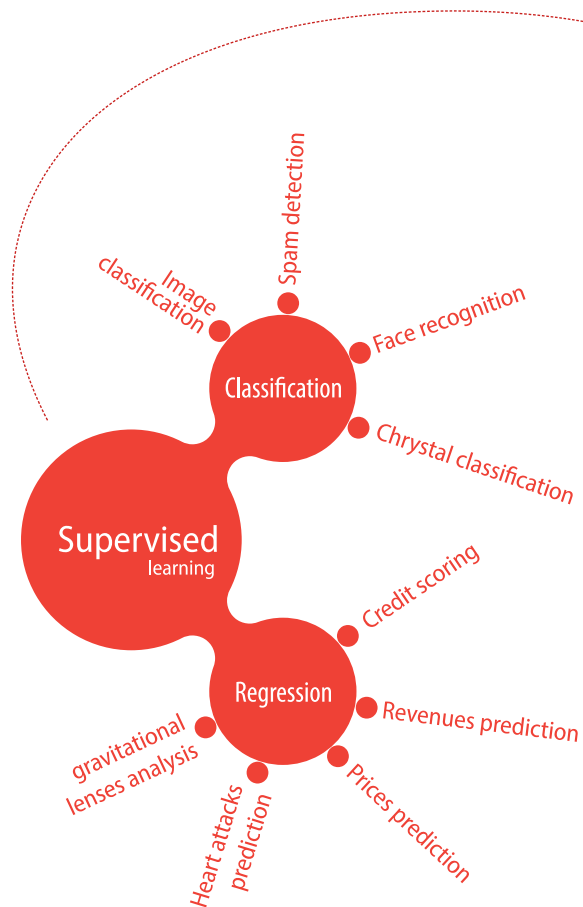






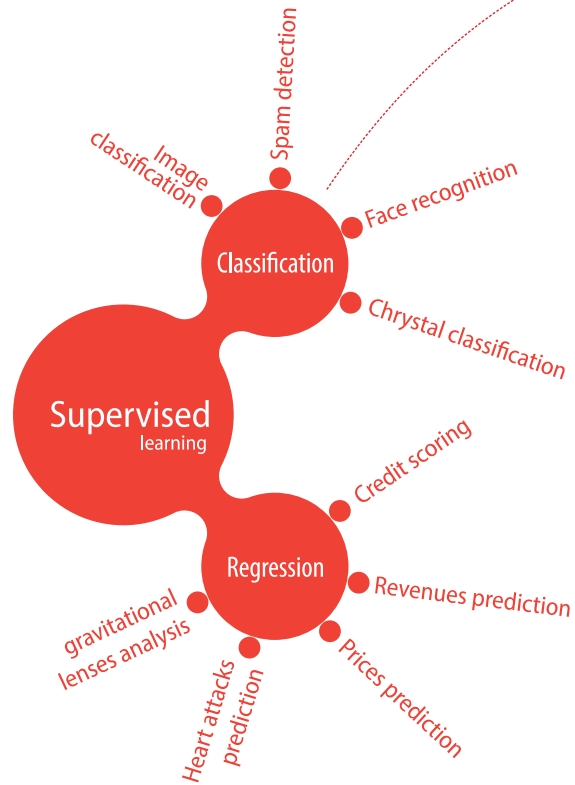






**Learning from examples**

# Supervised learning



## Classification :

Predict qualitative informations



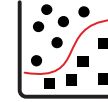
This is a cat



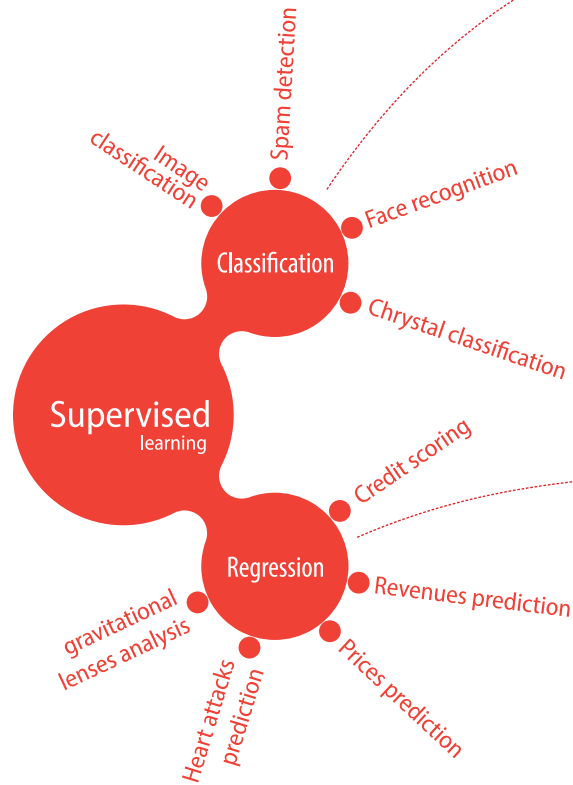
This is a rabbit



Tell me,  
what is it ?



# Supervised learning



## Classification :

Predict qualitative informations



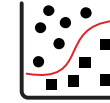
This is a cat



This is a rabbit



Tell me,  
what is it ?



## Régression :

Predict quantitative informations



150 K€



400 K€



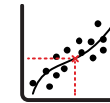
120 K€



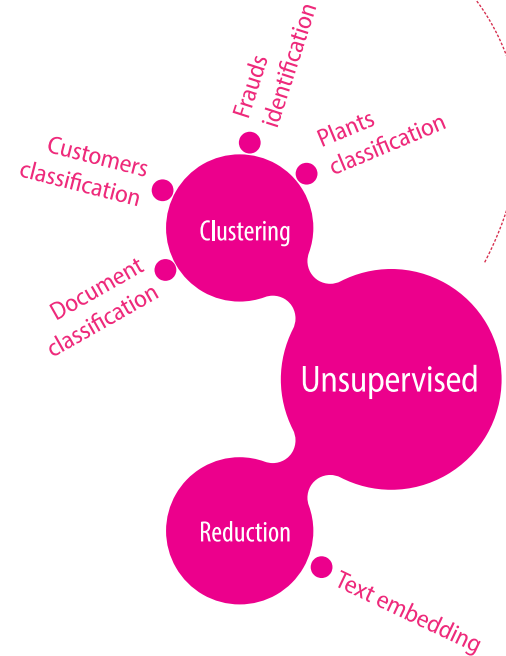
100 K€



Tell me,  
what's the  
price ?



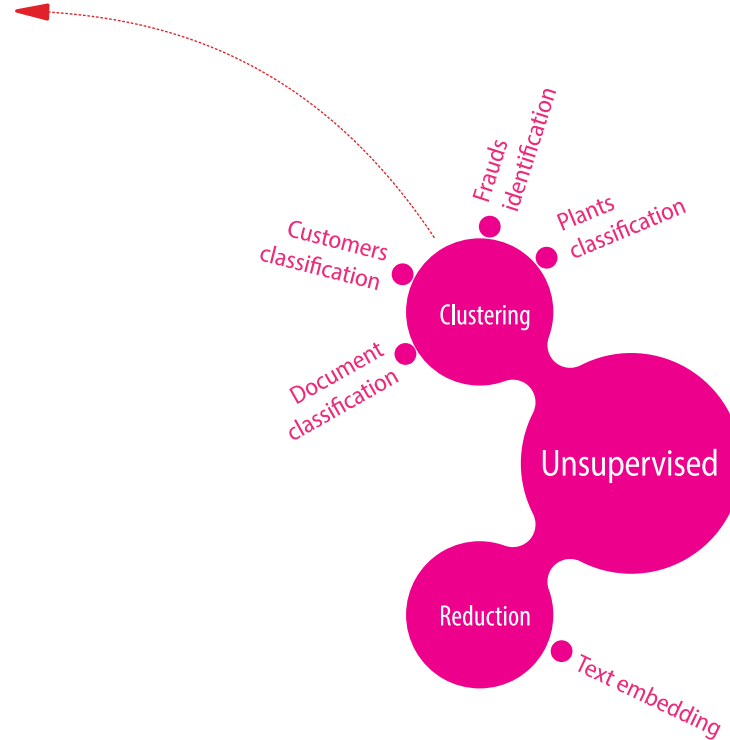
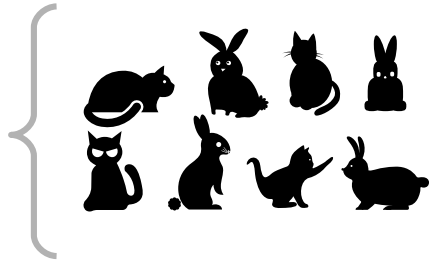
## Learning from data alone



## Clustering: Finding Common Relationships



What is the relationship between these data?





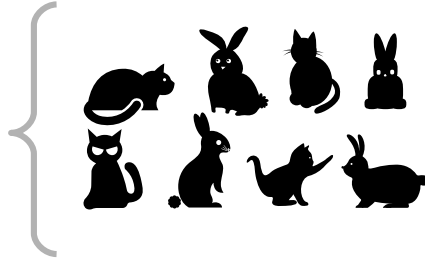
# Unsupervised learning

## Clustering:

Finding Common Relationships



What is the relationship between these data?

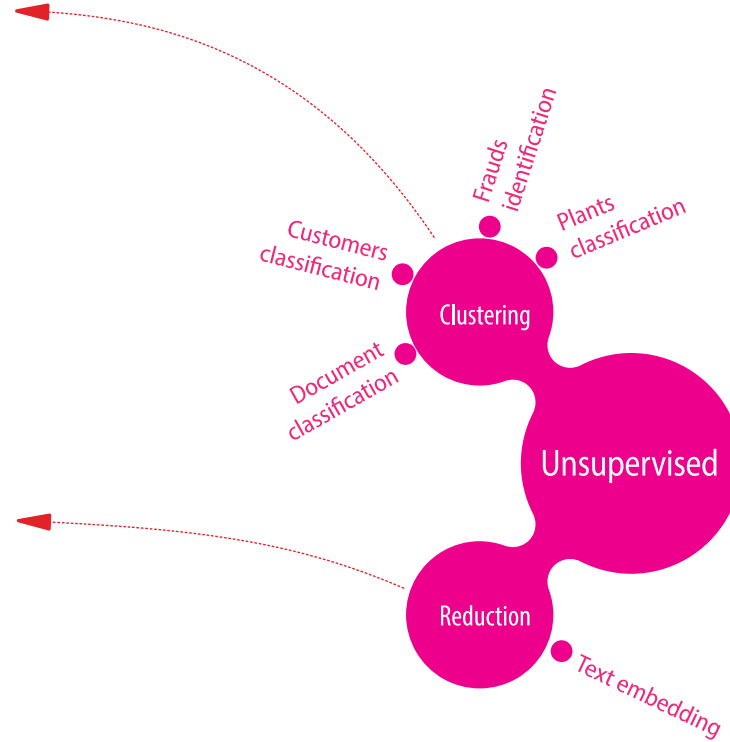


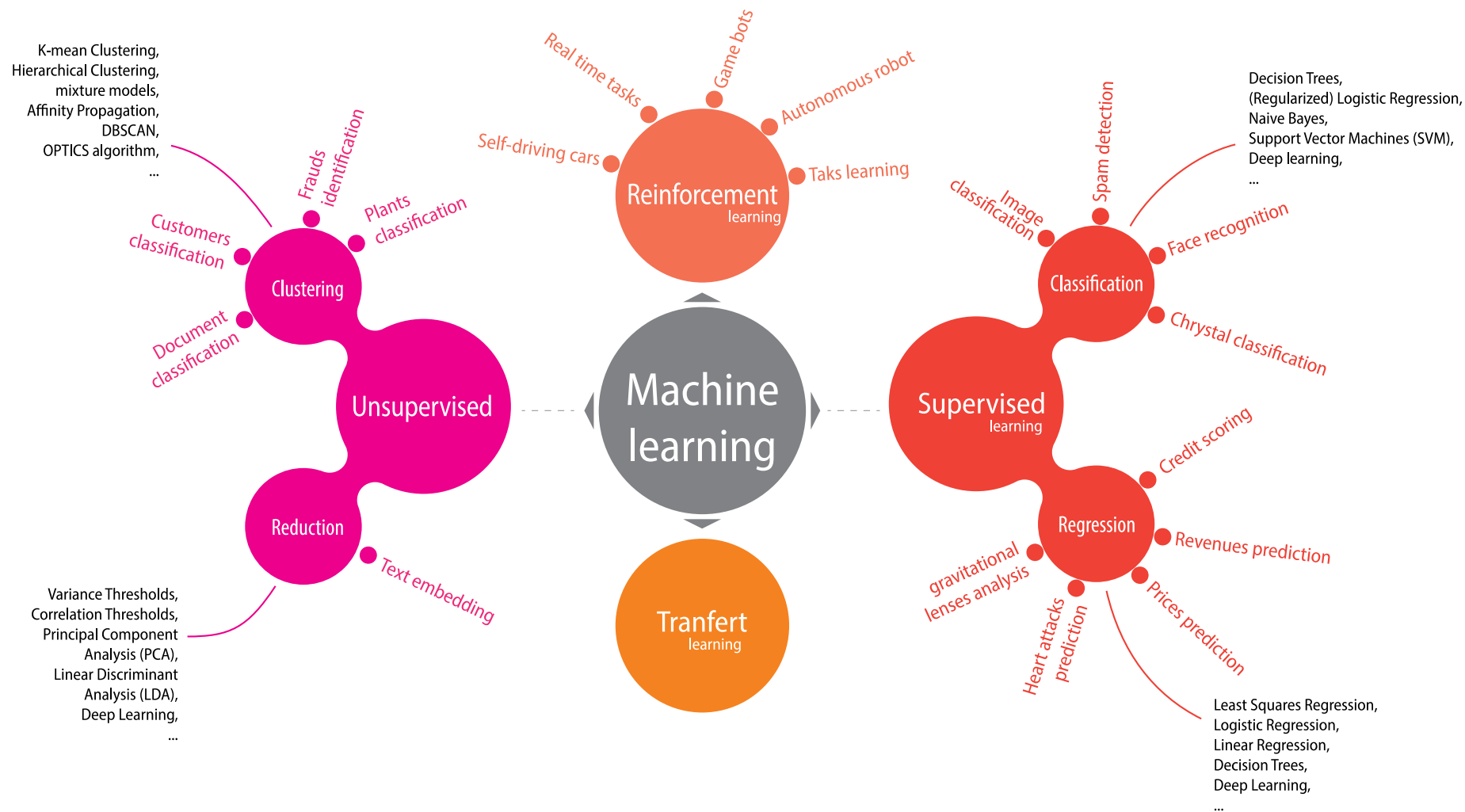
## Reduction:

Reduce the number of dimensions



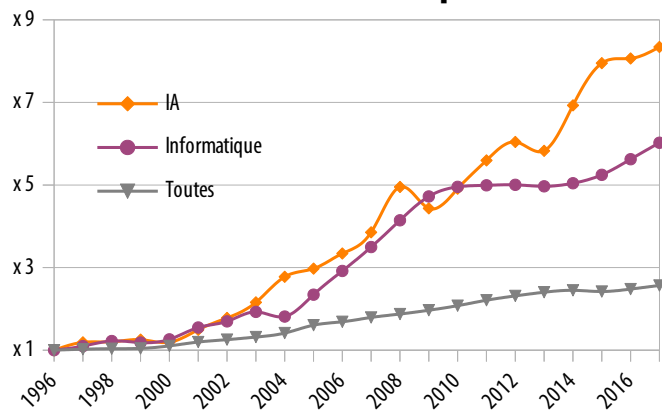
Simplify while keeping meaning



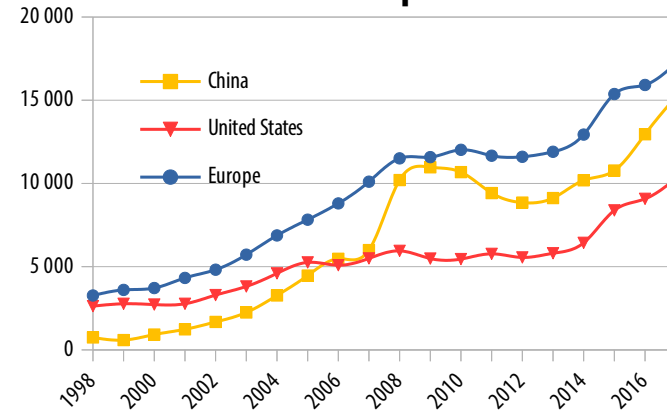


# A strong competition

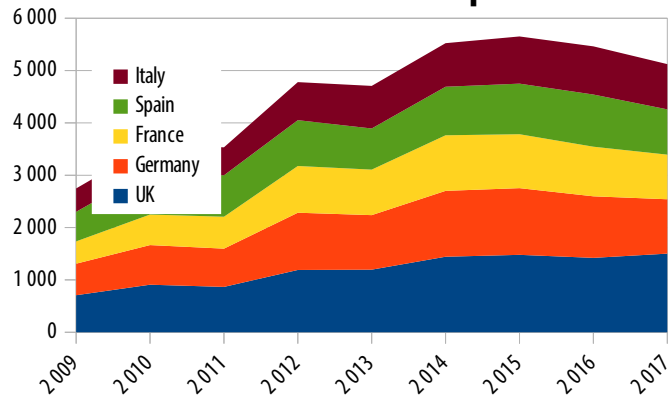
## Growth in the number of publications



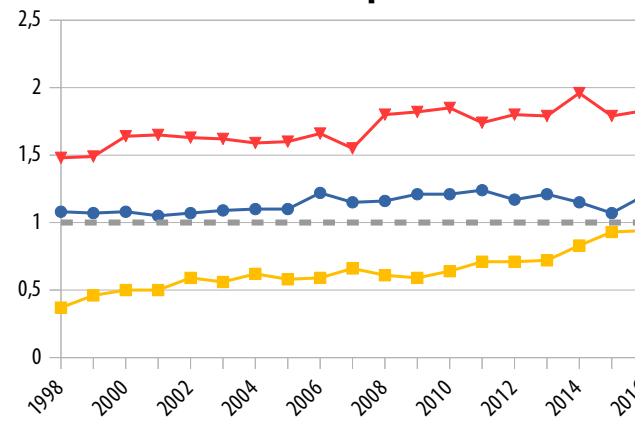
## Number of annual publications



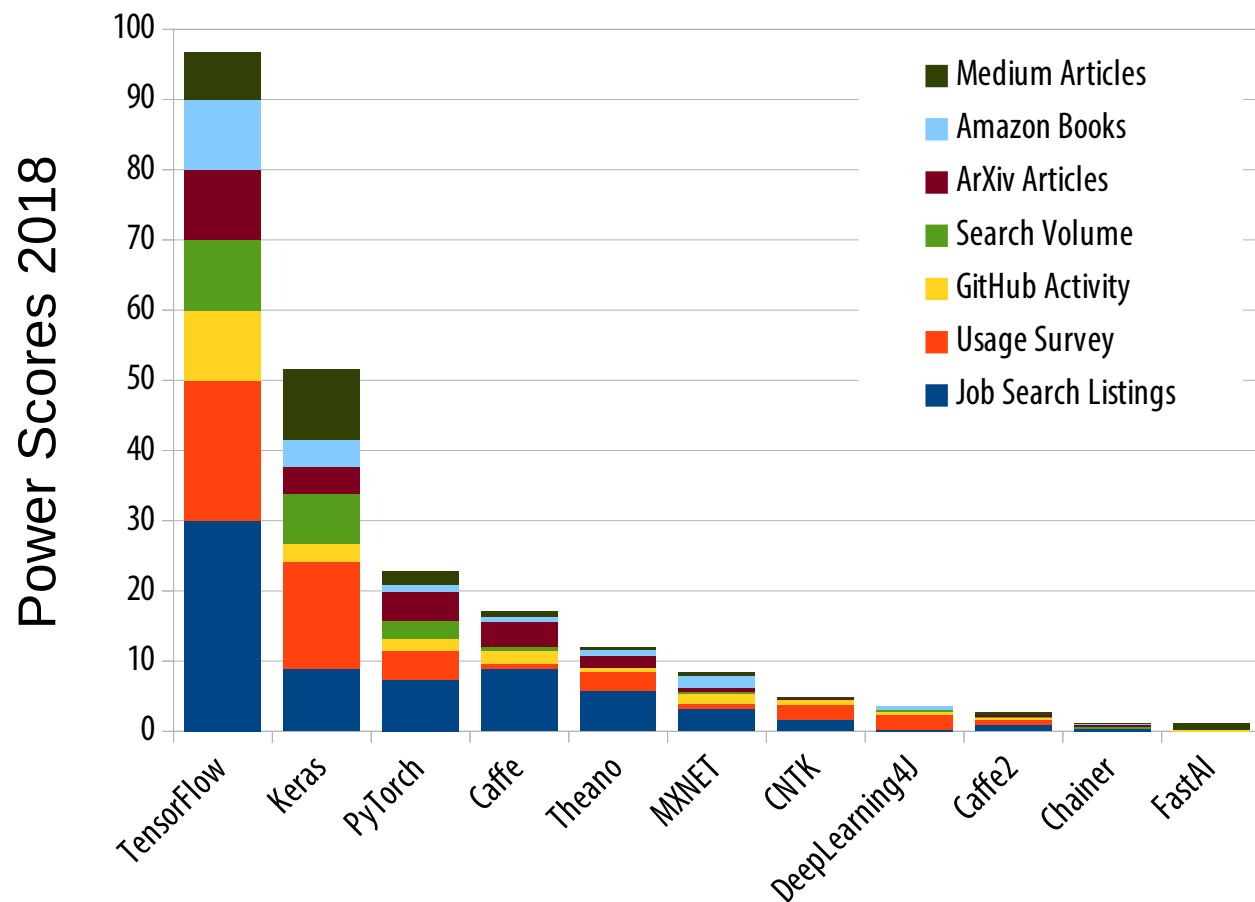
## Publications in Europe



## FWCI impact



# A Python centered world



Most used DL framework  
Supported by Google  
Low level API – an hard way  
Apache licence

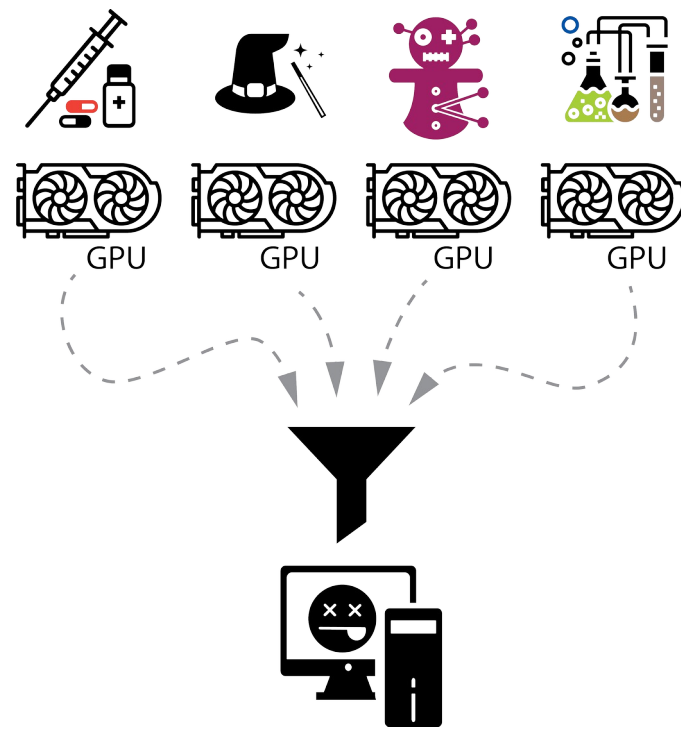
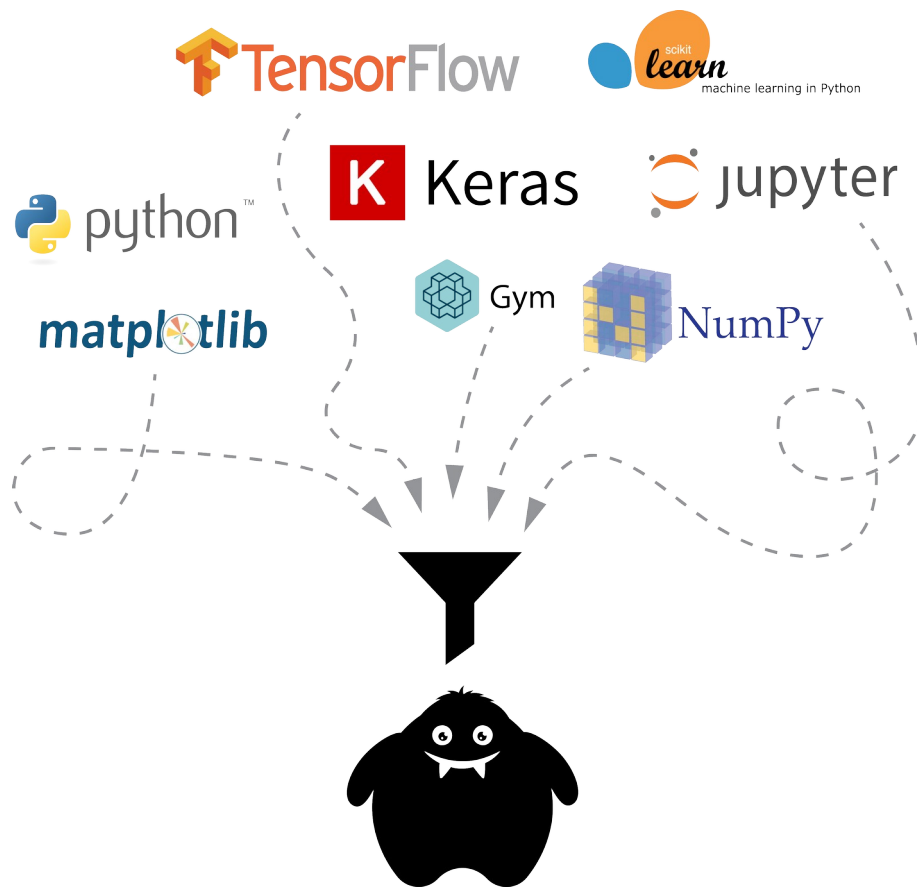


By François Cholet (Google)  
High level API  
Part on TensorFlow since 2017  
MIT licence

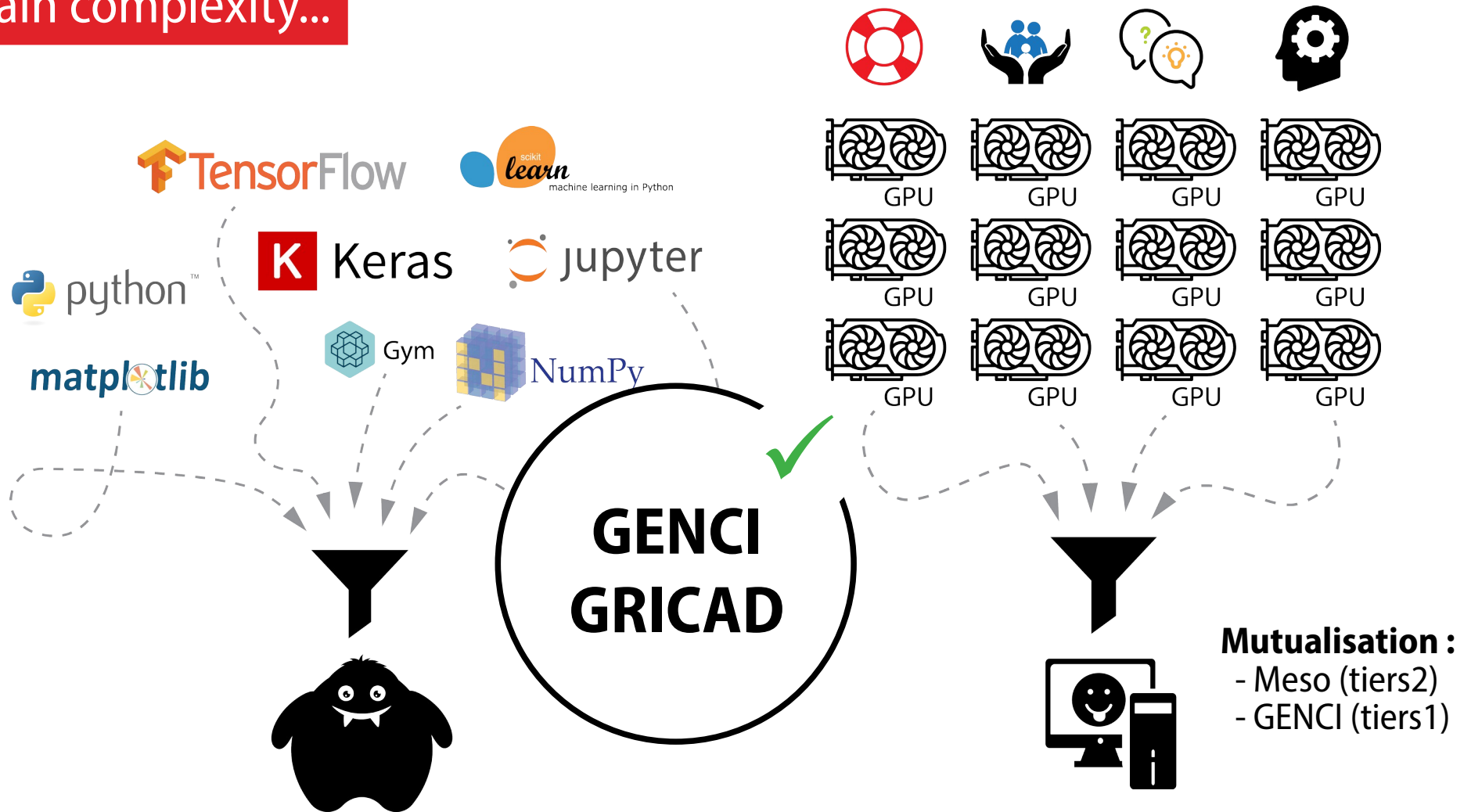


From Torch library  
Supported by Facebook  
BSD licence

# A certain complexity...



A certain complexity...



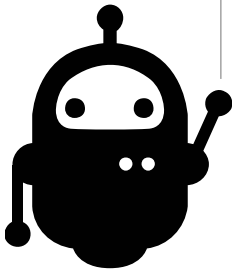


## **Connect** to GRICAD

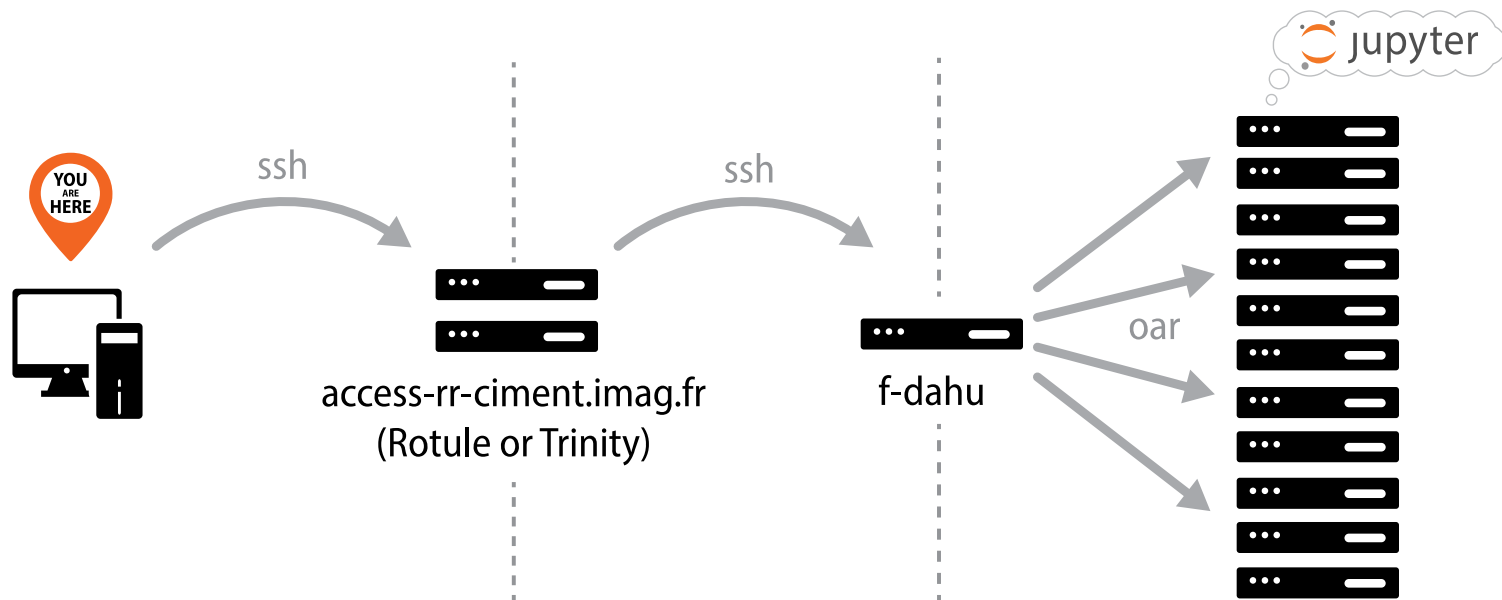
- SSH configuration

## **Clone** the Git repository

- Clone or copy



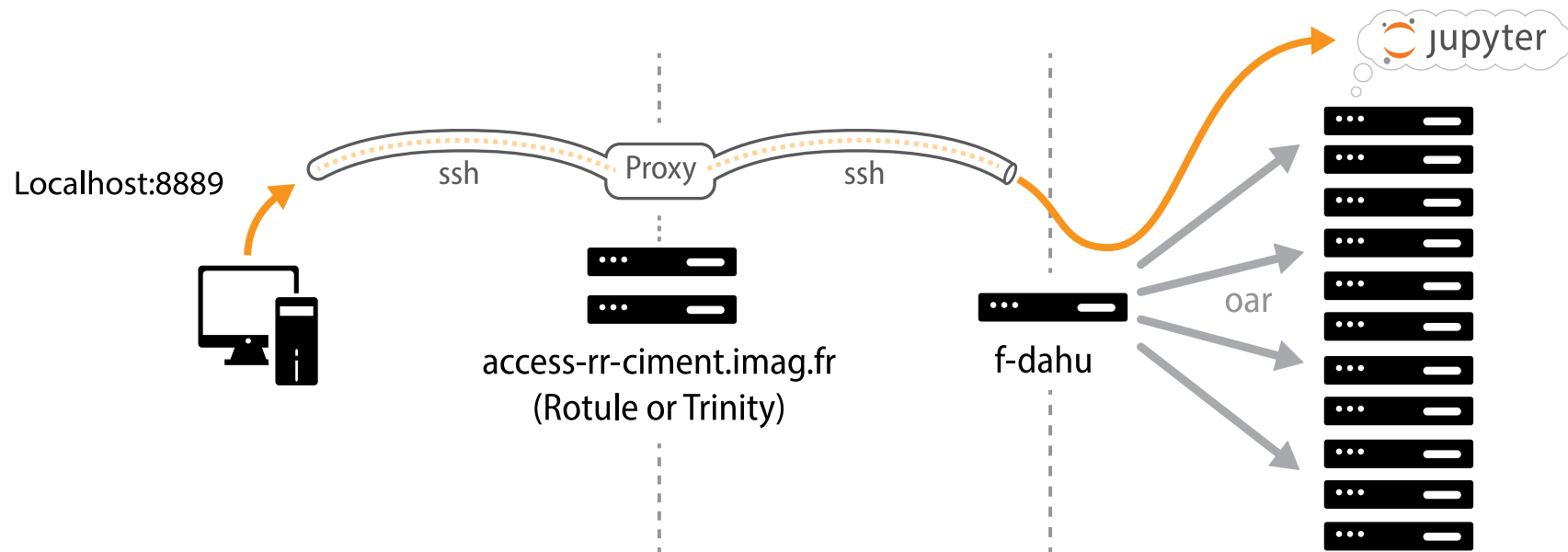
# Connect to GRICAD



Access to GRICAD's clusters requires passing through a **bastion**



# Connect to GRICAD



It is possible to configure your ssh client to make this **transparent**.



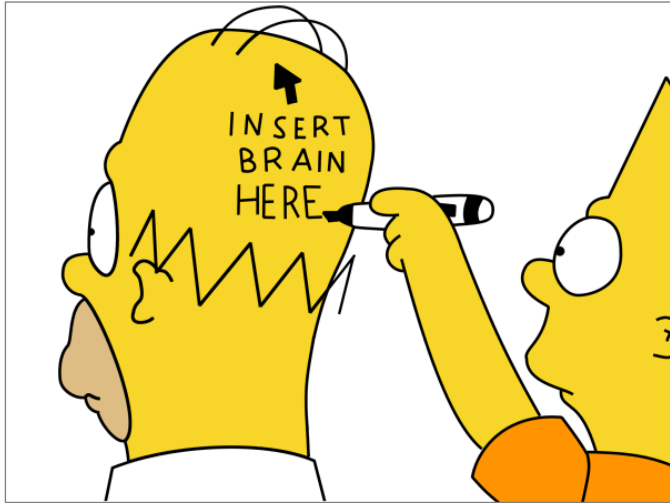
Wiki Fidle 🐳

<https://gricad-gitlab.univ-grenoble-alpes.fr/talks/fidle/-/wikis/home>

GRICAD documentation :

<https://gricad-doc.univ-grenoble-alpes.fr/hpc/connexion/>



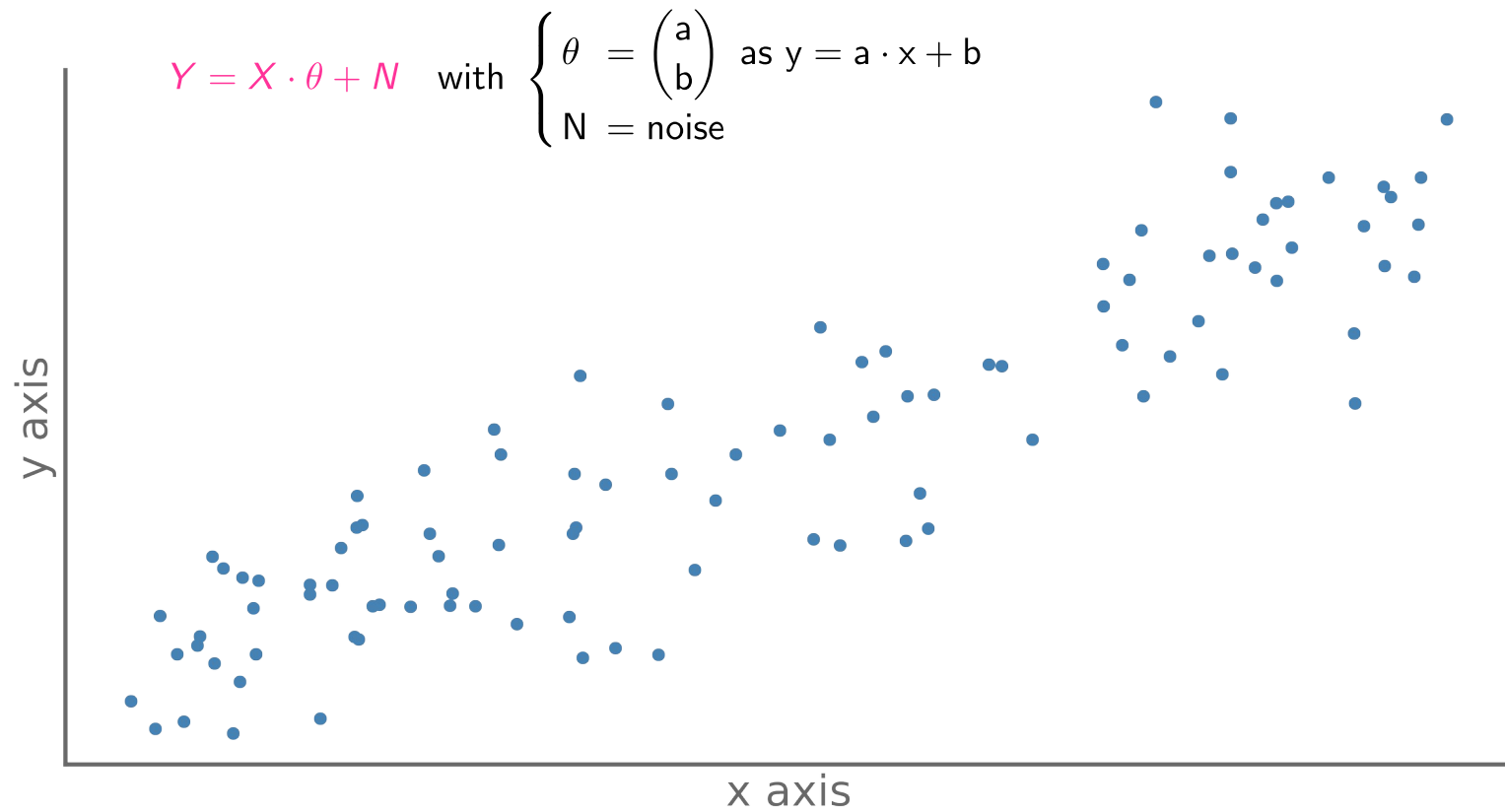


Fine, but  
Deep Learning  
What's that?

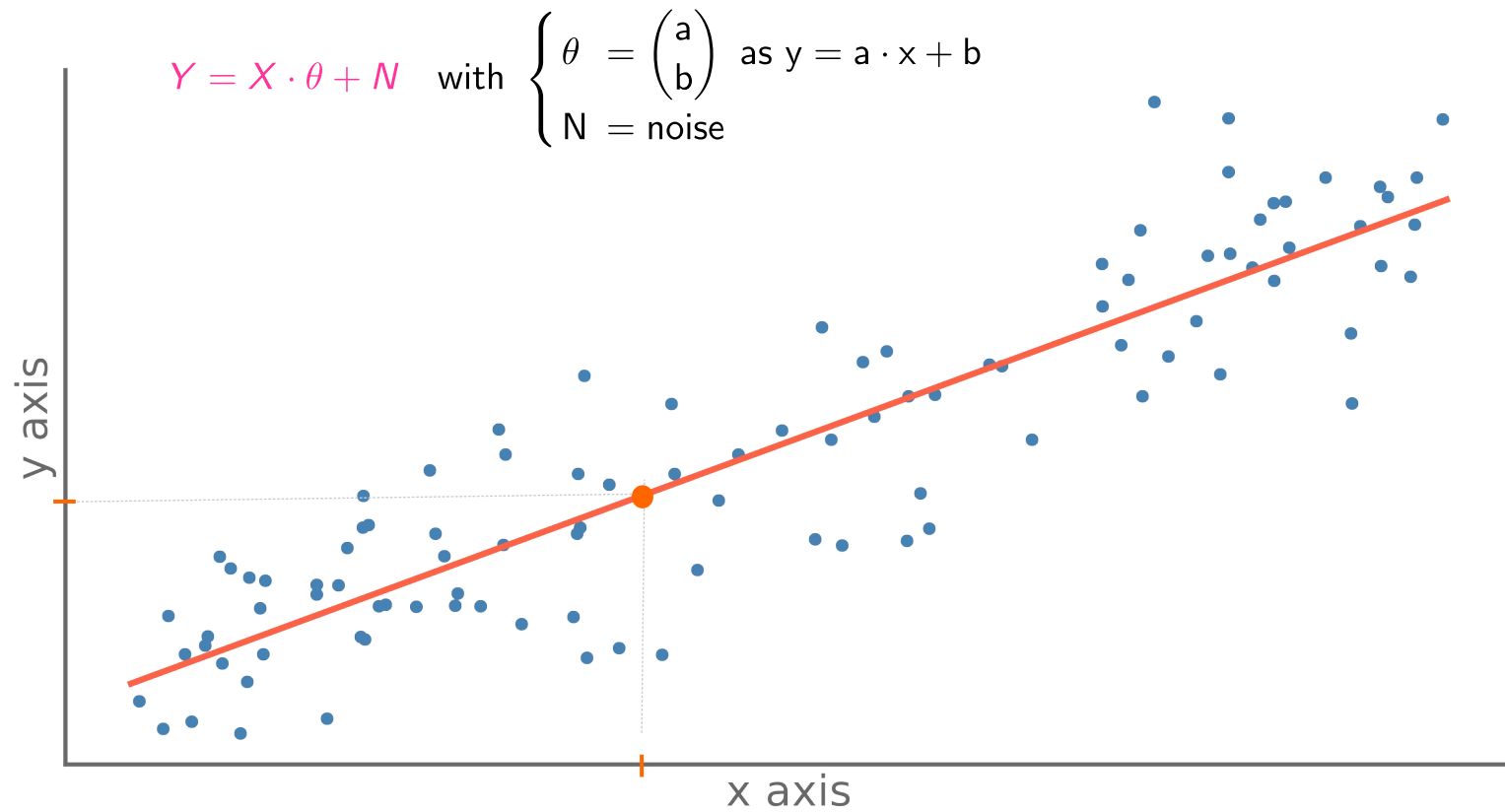
# From the linear regression to the first neuron



# Linear regression



# Linear regression



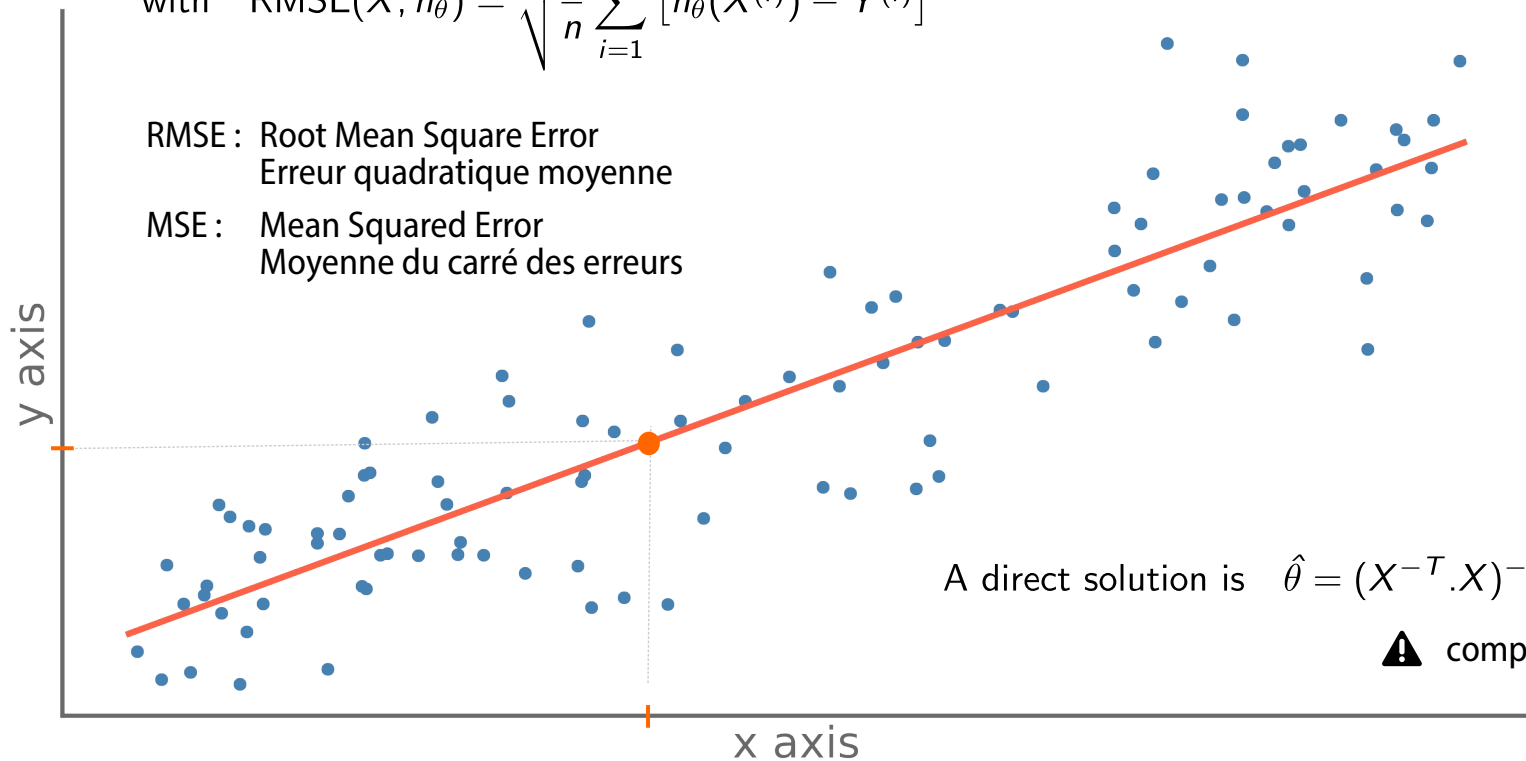
# Linear regression

We search  $\hat{\theta} = \begin{pmatrix} \hat{a} \\ \hat{b} \end{pmatrix}$  for which  $\text{RMSE}(X, \hat{\theta})$  is minimal

with 
$$\text{RMSE}(X, h_{\theta}) = \sqrt{\frac{1}{n} \sum_{i=1}^n [h_{\theta}(X^{(i)}) - Y^{(i)}]^2}$$

RMSE : Root Mean Square Error  
Erreur quadratique moyenne

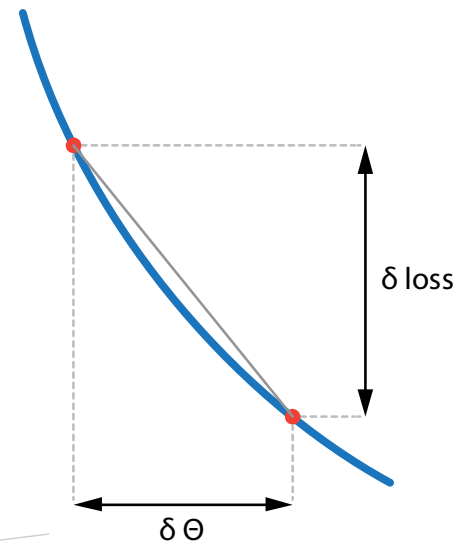
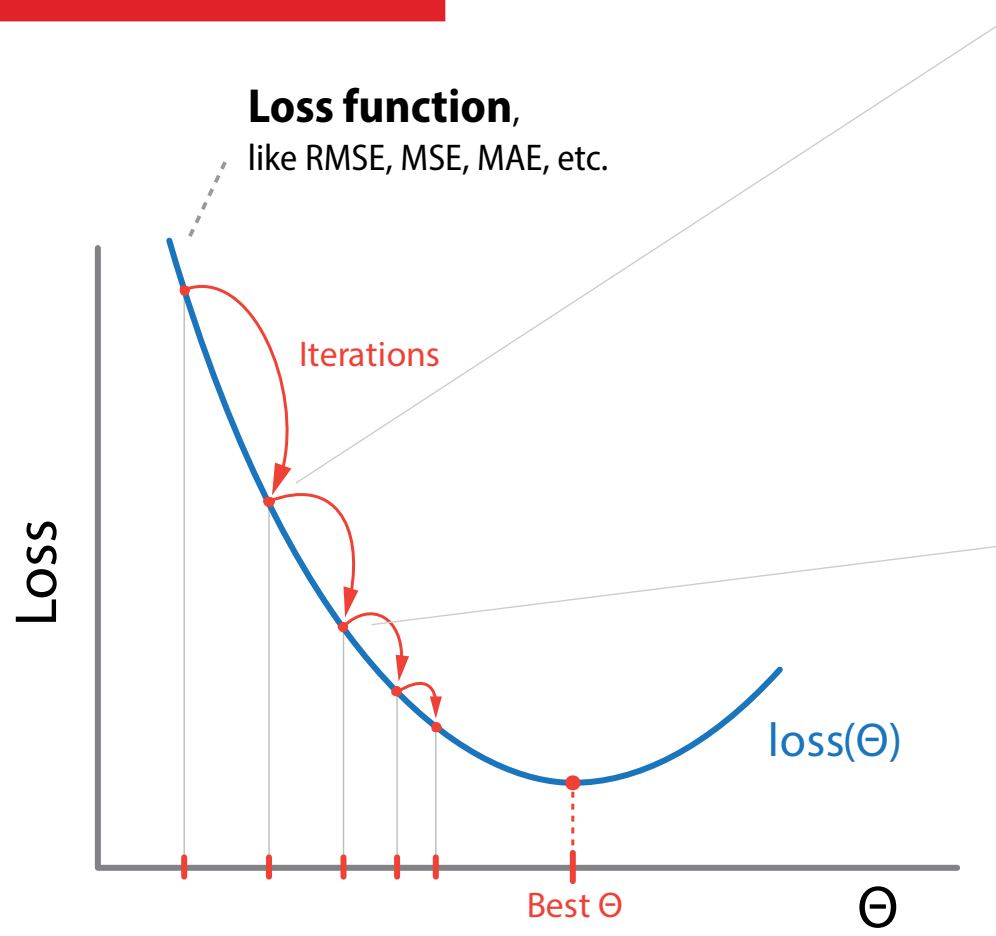
MSE : Mean Squared Error  
Moyenne du carré des erreurs



A direct solution is  $\hat{\theta} = (X^{-T}.X)^{-1}.X^{-T}.Y$

⚠ complexity in  $n^3$

# Gradient descent



$$\text{gradient} = \frac{\delta \text{loss}}{\delta \theta}$$

Iterative solution is :  $\theta \leftarrow \theta - \eta \cdot \frac{\delta \text{loss}}{\delta \theta}$

where  $\eta$  is the learning rate

This process is called **stochastic gradient descent** and the function used to optimize the descent, **optimization function**

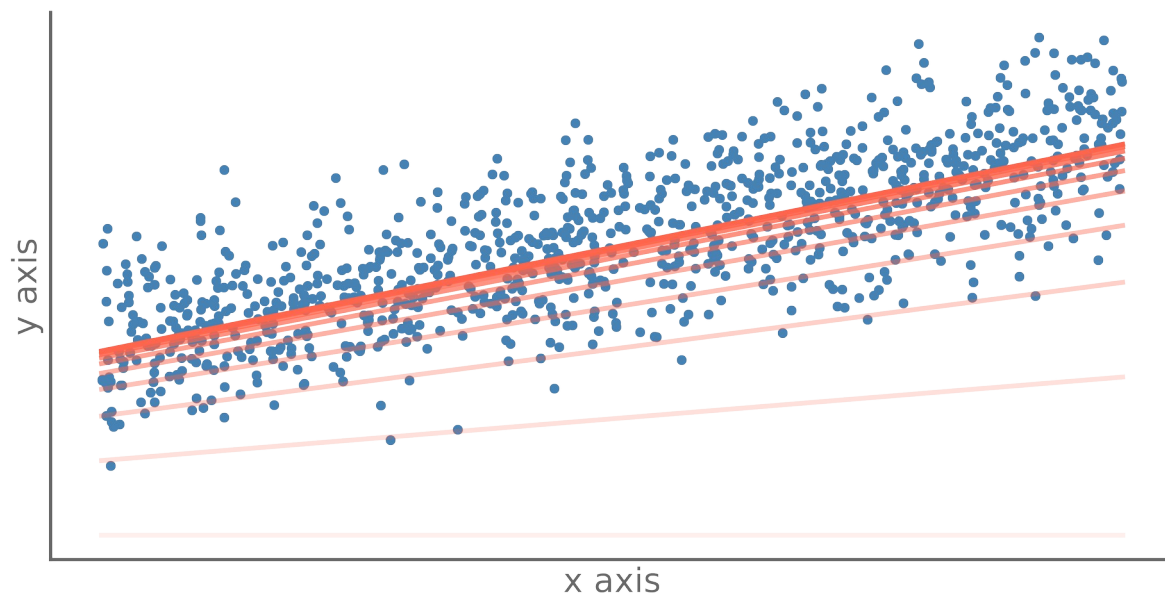


# Gradient descent

$$MSE(X, h_{\theta}) = \frac{1}{n} \sum_{i=1}^n \left[ h_{\theta}(X^{(i)}) - Y^{(i)} \right]^2$$

$$\nabla_{\theta} MSE(\Theta) = \begin{bmatrix} \frac{\partial}{\partial \theta_0} MSE(\Theta) \\ \frac{\partial}{\partial \theta_1} MSE(\Theta) \\ \vdots \\ \frac{\partial}{\partial \theta_n} MSE(\Theta) \end{bmatrix} = \frac{2}{m} X^T \cdot (X \cdot \Theta - Y)$$

Iterative solution is :  $\Theta \leftarrow \Theta - \eta \cdot \nabla_{\theta} MSE(\Theta)$   
where  $\eta$  is the learning rate

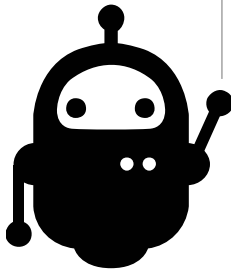


#i	Loss	Gradient		Theta	
0	+12.481	-6.777	-1.732	-3.388	+0.000
20	+4.653	-4.066	-1.039	-2.033	+0.346
40	+1.835	-2.440	-0.624	-1.220	+0.554
60	+0.821	-1.464	-0.374	-0.732	+0.679
80	+0.455	-0.878	-0.224	-0.439	+0.754
100	+0.324	-0.527	-0.135	-0.263	+0.799
120	+0.277	-0.316	-0.081	-0.158	+0.826
140	+0.260	-0.190	-0.048	-0.095	+0.842
160	+0.253	-0.114	-0.029	-0.057	+0.851
180	+0.251	-0.068	-0.017	-0.034	+0.857
200	+0.250	-0.041	-0.010	-0.020	+0.861



# Linear regression with gradient descent

Notebook : **[GRAD1]**

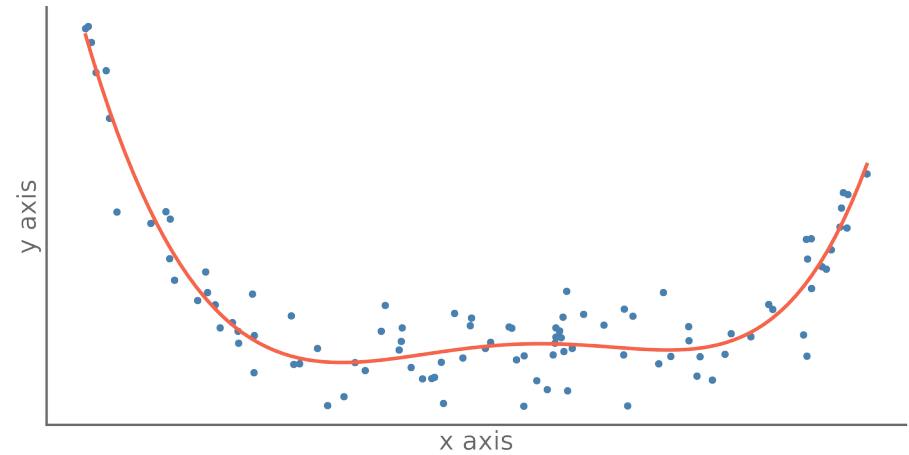
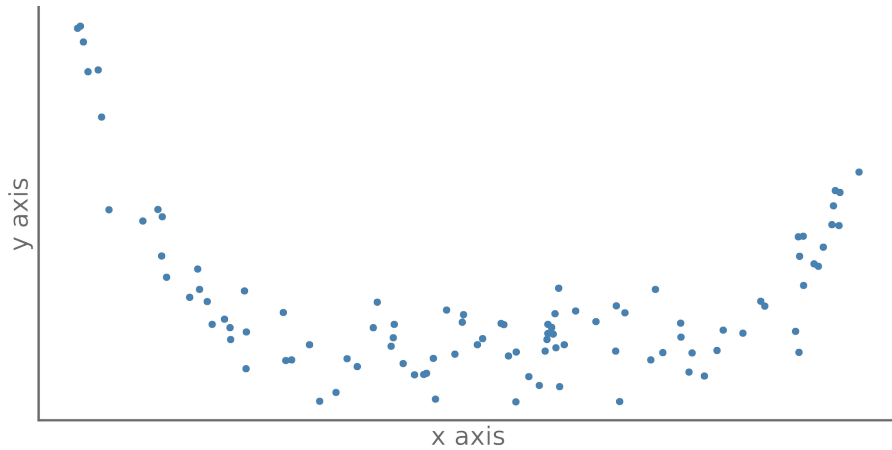


## **Objective :**

See by example a gradient descent

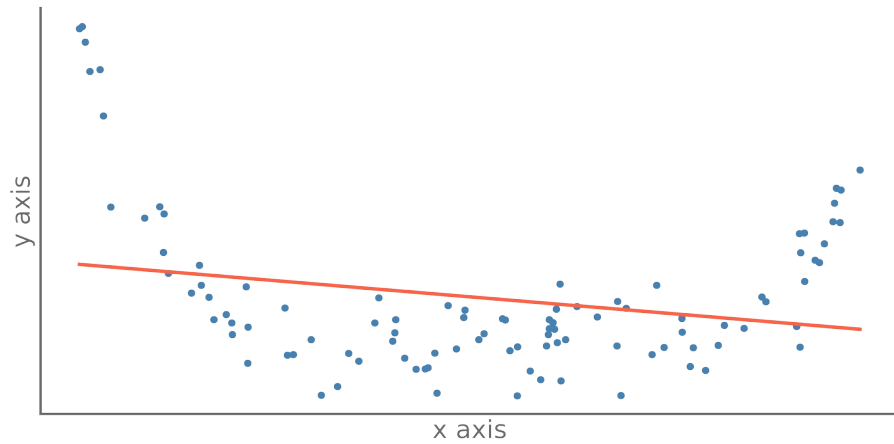


# Polynomial regression

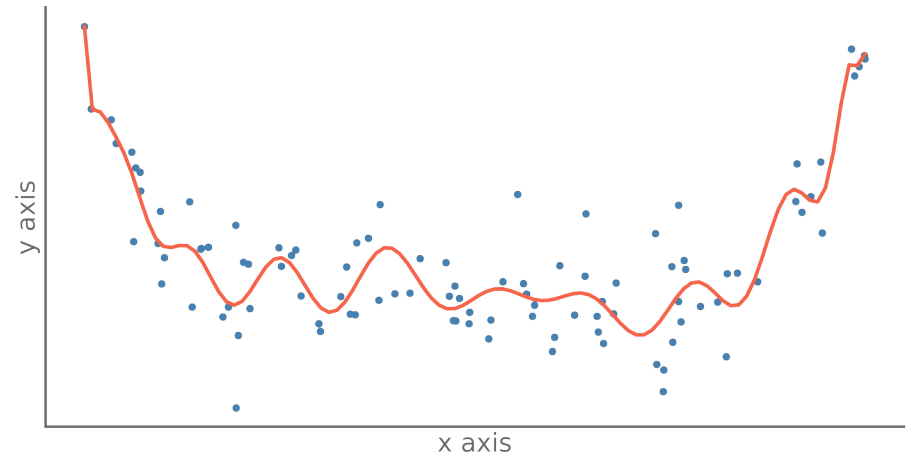


$$P_n(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \cdots + a_n \cdot x^n = \sum_{i=0}^n a_i \cdot x^i$$

# Polynomial regression



Underfitting



Overfitting

# Logistic regression

A logistic regression is intended to provide a probability of belonging to a class.

**Dataset :** X Observations  
y Classe

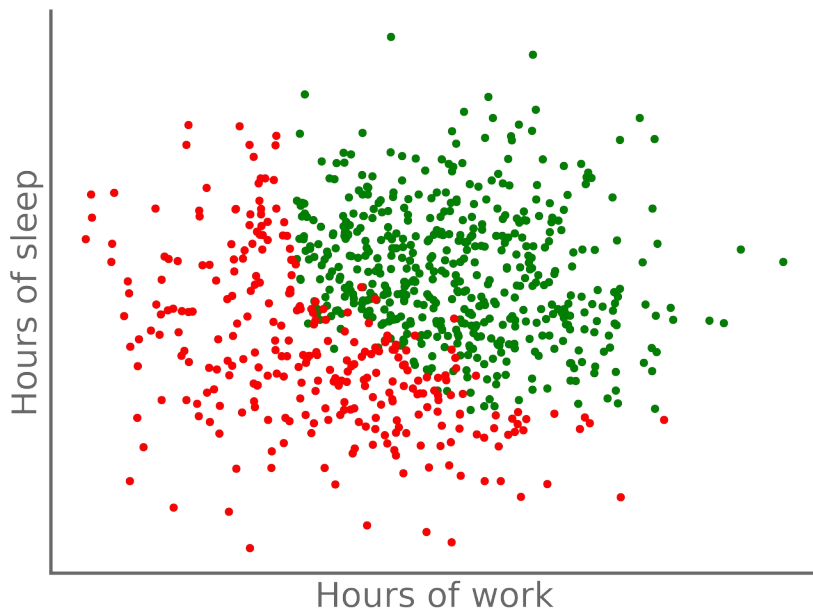


$$(X_i, y_i) \left\{ \begin{array}{l} X_i = \begin{pmatrix} x_{i1} = \text{Hours of work} \\ x_{i2} = \text{Hours of sleep} \end{pmatrix} \\ y_i = \begin{cases} 1 & \text{belong to the class} \\ 0 & \text{don't belong} \end{cases} \end{array} \right.$$

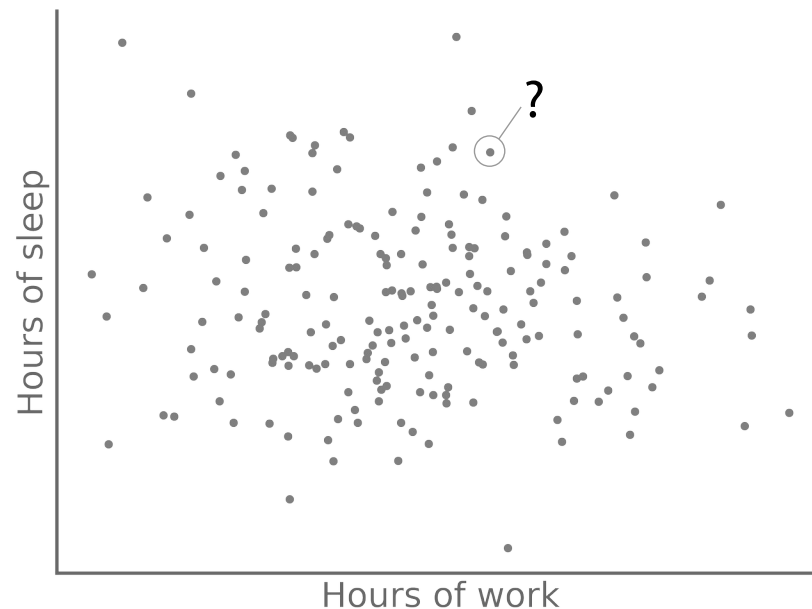
# Logistic regression

A logistic regression is intended to provide a probability of belonging to a class.

**Dataset :** X Observations  
y Classe

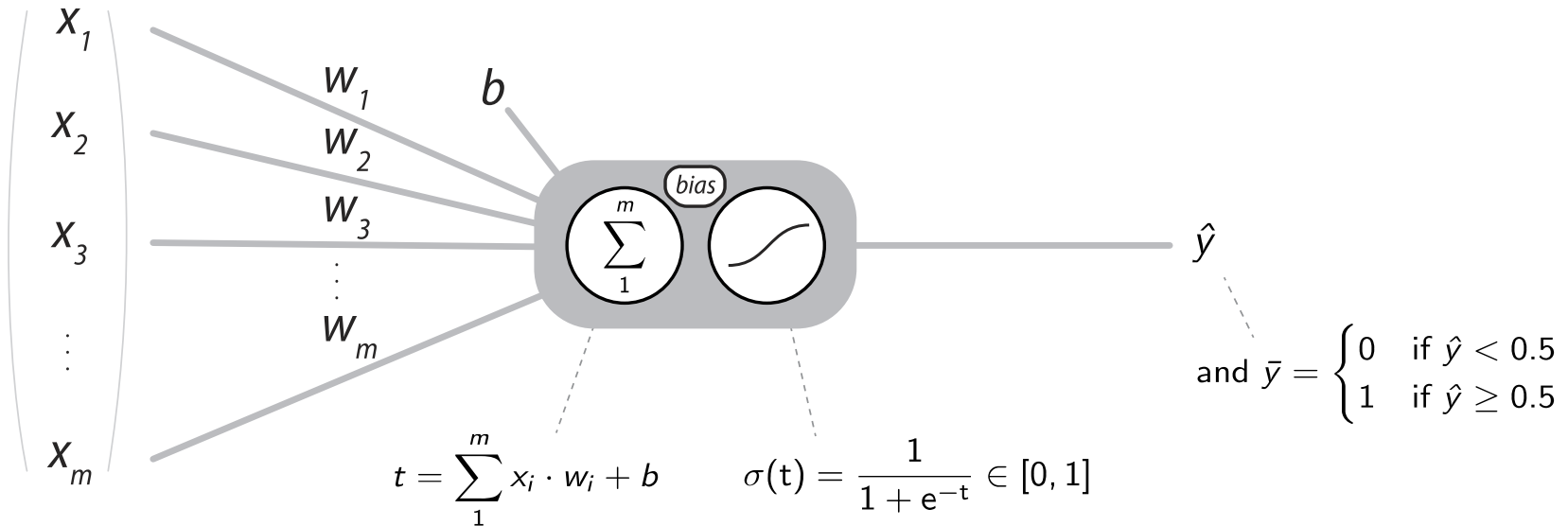


**Objective :** Predict the class  
x given, we want to predict y  
 $y_{\text{pred}} = f(x)$



# Logistic regression

$$\hat{y} = \sigma(\Theta^T \cdot X + b)$$



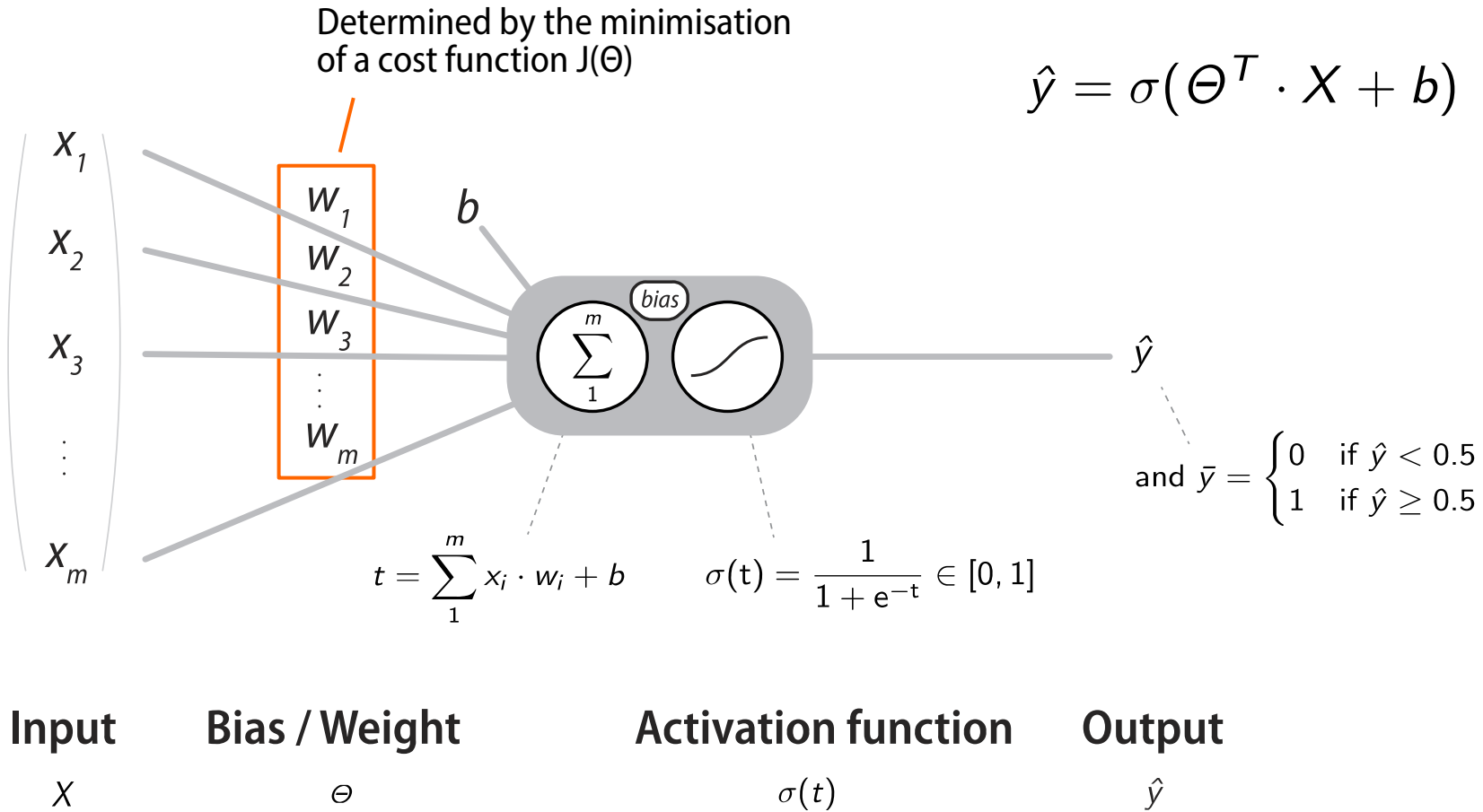
**Input**  
 $X$

**Bias / Weight**  
 $\Theta$

**Activation function**  
 $\sigma(t)$

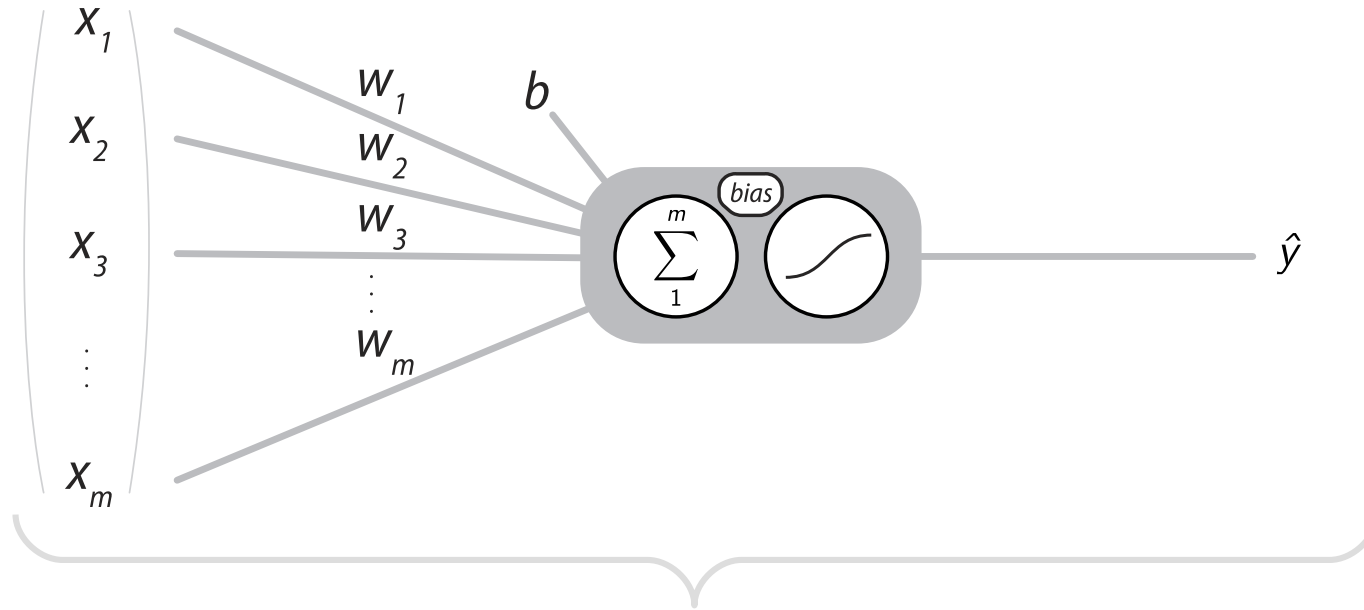
**Output**  
 $\hat{y}$

# Logistic regression





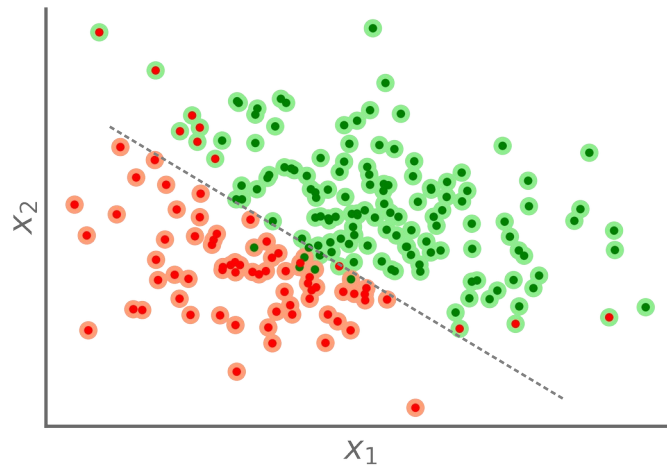
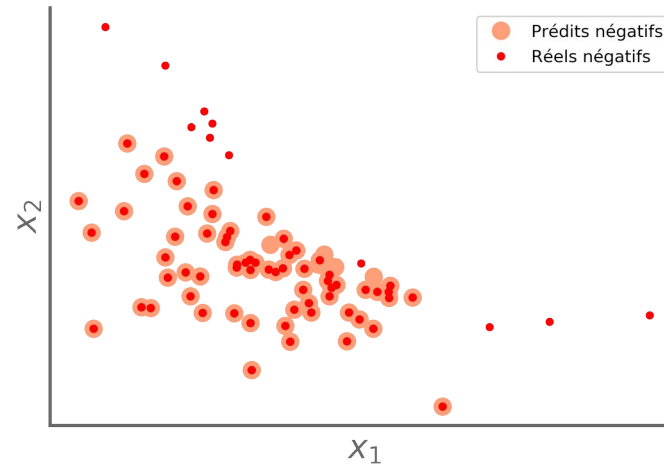
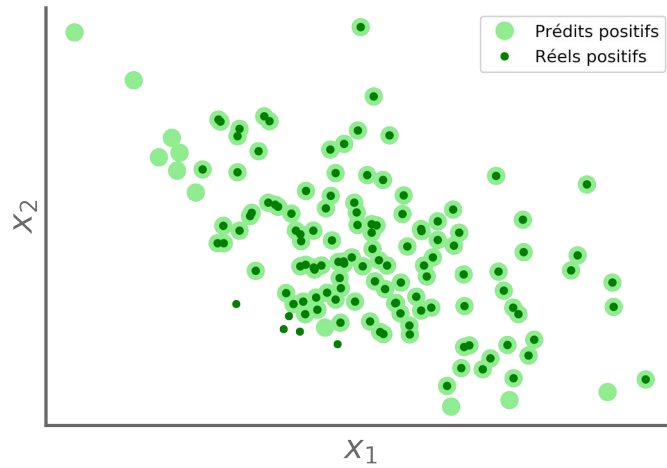
$$\hat{y} = \sigma(\Theta^T \cdot X + b)$$



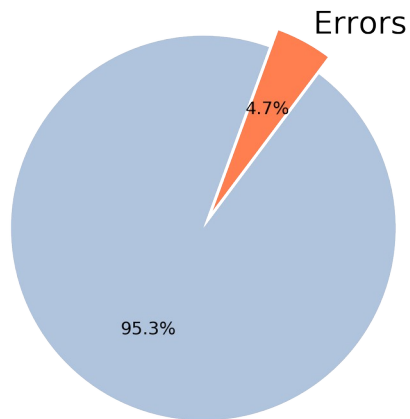
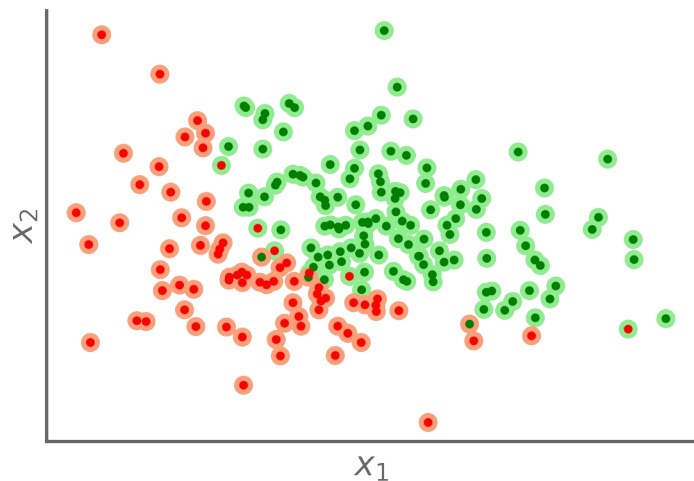
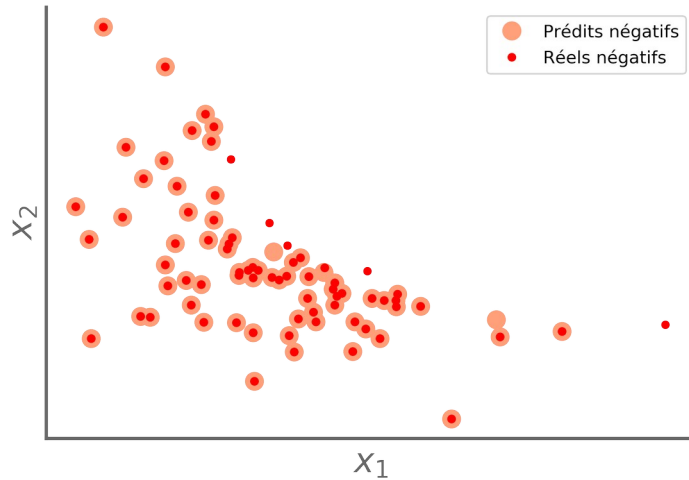
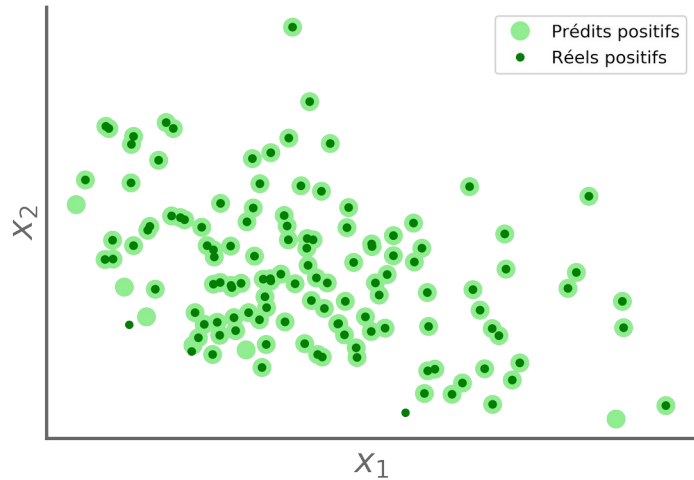
That's an « **artificial neuron** » !

So, we have a neural network of... 1 neuron !

# Logistic regression



# Logistic regression



Linear => Non linear

$\forall i \in [0, m]$ , we add :  $x_{i1}^2, x_{i2}^2, x_{i1}^3, x_{i2}^3$  to  $X_i$

so, for :

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} \\ \vdots & \dots & \vdots \\ 1 & x_{m1} & x_{m2} \end{bmatrix}$$

we have :

$$\hat{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{11}^2 & x_{12}^2 & x_{11}^3 & x_{12}^3 \\ \vdots & & & \dots & & & \\ 1 & x_{m1} & x_{m2} & x_{m1}^2 & x_{m2}^2 & x_{m1}^3 & x_{m2}^3 \end{bmatrix}$$

# Neurons at the heart of a controversy



[ intelligence ]



# [ intelligence ]

« Capacité de percevoir ou d'inférer l'information, et de la conserver comme une connaissance à appliquer à des comportements adaptatifs dans un environnement ou un contexte donné »

« Ability to perceive or infer information, and to retain it as knowledge to be applied towards adaptive behaviors within an environment or context »\*



# [ intelligence ]

« Ensemble des **fonctions** mentales ayant pour objet la connaissance **conceptuelle** et **rationnelle** »\*

*« Set of mental functions aimed at conceptual and rational knowledge »*

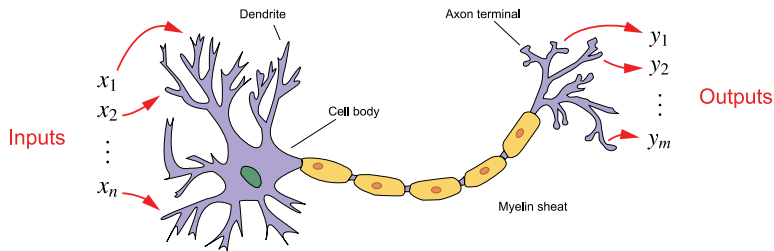
### Modelling the brain :

« Penser s'apparente  
à un calcul massivement parallèle de  
**fonctions élémentaires**.

L'information est un **signal** avant  
d'être un code »<sup>1</sup>

## Connectionnism

*Modelling the brain*  
*Modéliser le cerveau*



### Making a mind :

« Penser, c'est calculer des **symboles** qui  
ont à la fois une réalité matérielle et une  
valeur sémantique de représentation »<sup>1</sup>

L'information est une donnée  
symbolique de **haut niveau**.

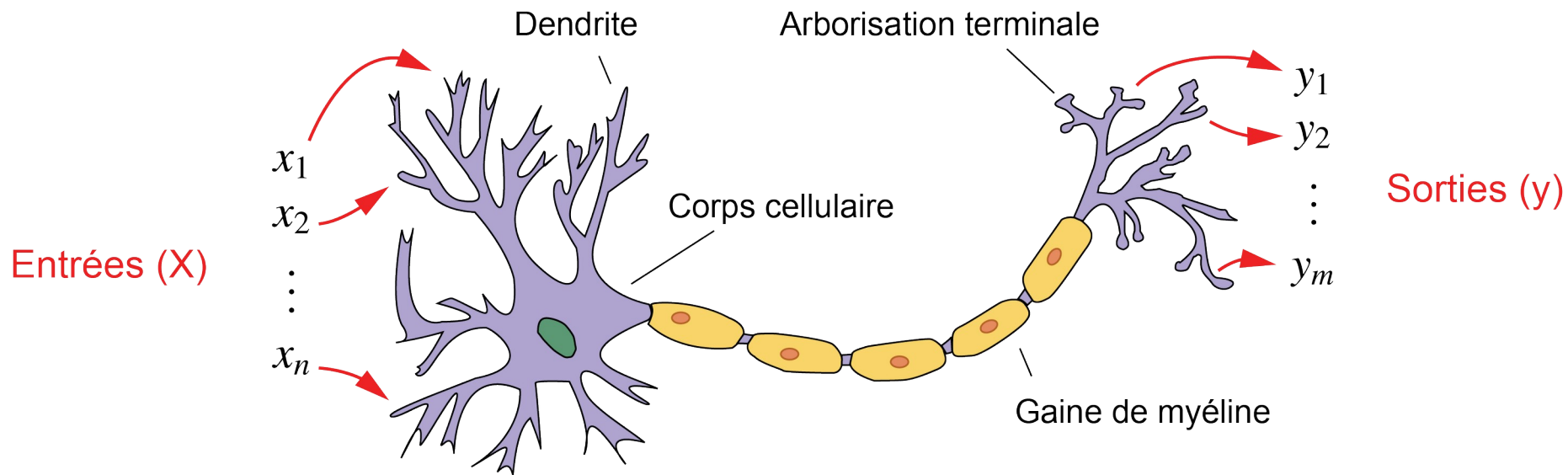
## Symbolic

*Making a mind*  
*Forger une opinion*

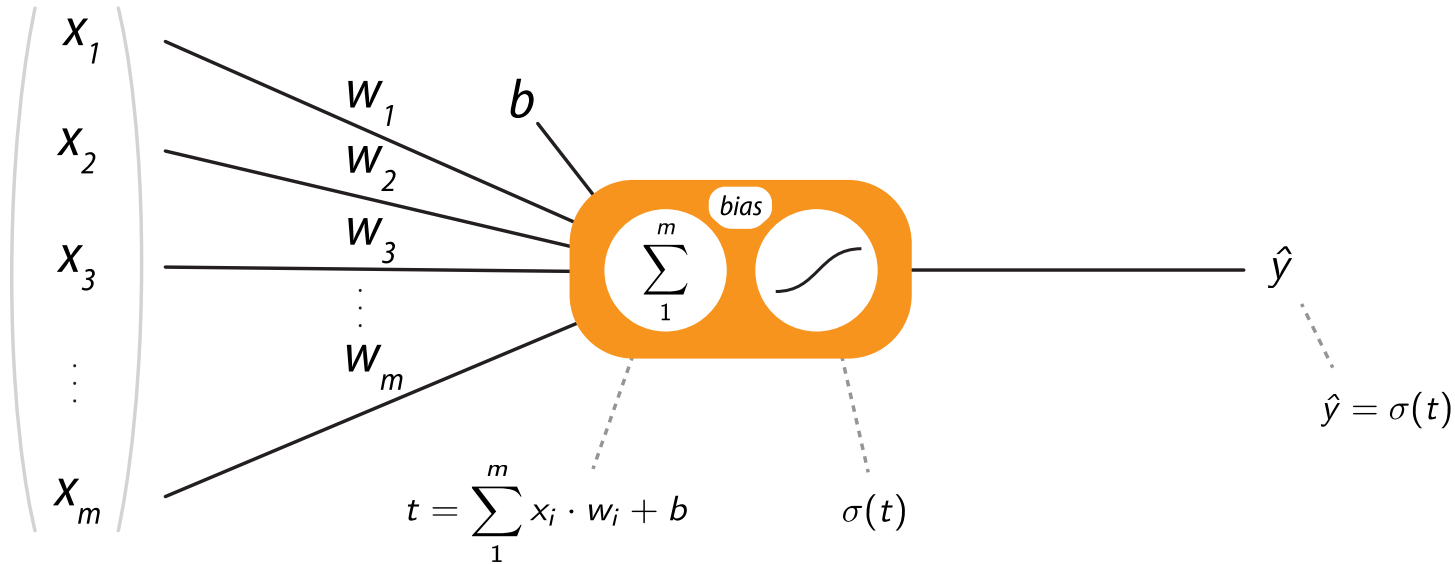
Tout [homme] est [mortel]  
[Socrate] est un [homme]  
Donc [Socrate] est [mortel]

<sup>1</sup> D Cardon, JP Cointet, A Mazieres, 2018 [LRDN]





$$\hat{y} = \sigma(\Theta^T \cdot X + b)$$



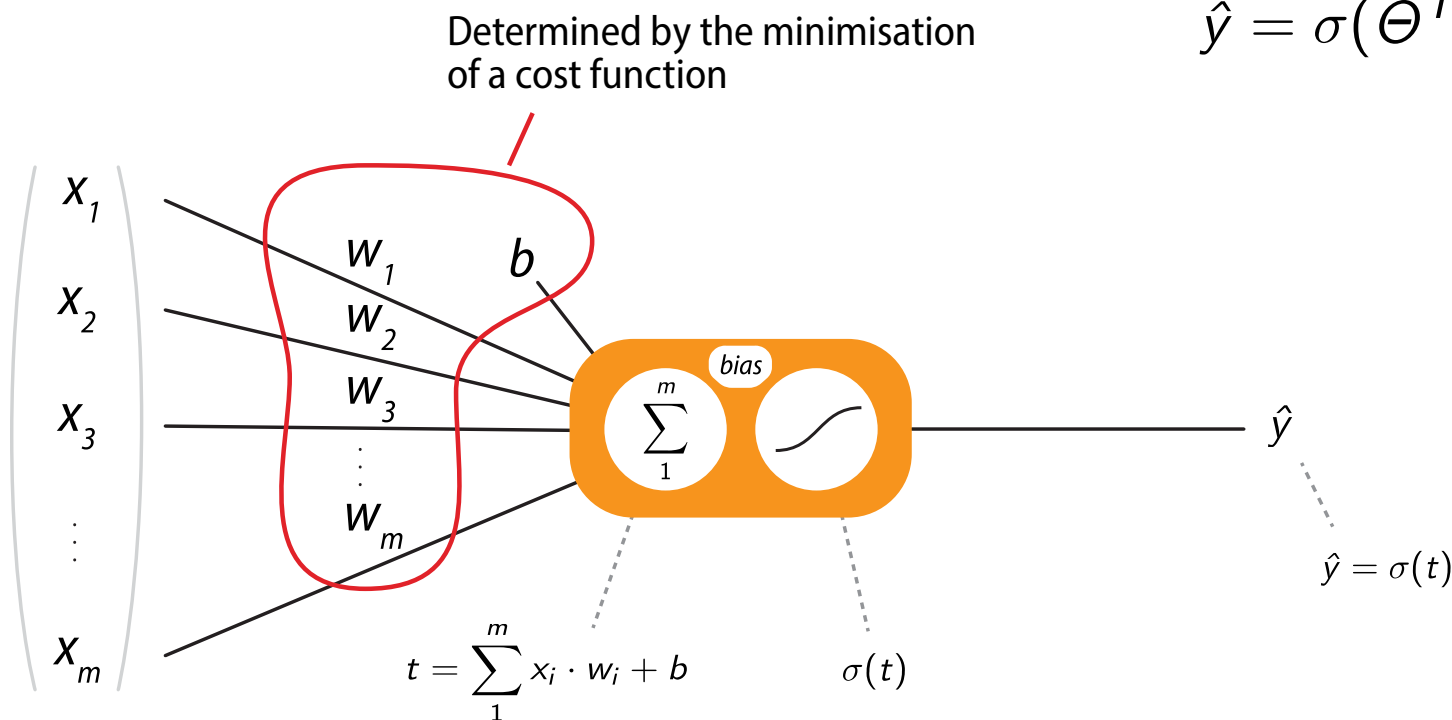
**Input**  
 $X$

**Bias / Weight**  
 $\Theta, b$

**Activation function**  
 $\sigma(t)$

**Output**  
 $\hat{y}$

$$\hat{y} = \sigma(\Theta^T \cdot X + b)$$



**Input**  
 $X$

**Bias / Weight**  
 $\Theta, b$

**Activation function**  
 $\sigma(t)$

**Output**  
 $\hat{y}$

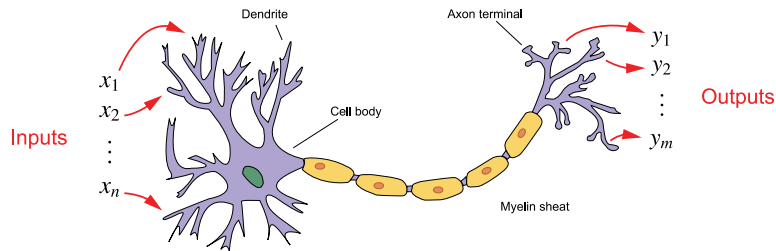
### Modelling the brain :

« Penser s'apparente  
à un calcul massivement parallèle de  
**fonctions élémentaires**.

L'information est un **signal** avant  
d'être un code »<sup>1</sup>

## Connectionnism

*Modelling the brain*  
*Modéliser le cerveau*



### Making a mind :

« Penser, c'est calculer des **symboles** qui  
ont à la fois une réalité matérielle et une  
valeur sémantique de représentation »<sup>1</sup>

L'information est une donnée  
symbolique de **haut niveau**.

## Symbolic

*Making a mind*  
*Forger une opinion*

Tout [homme] est [mortel]  
[Socrate] est un [homme]  
Donc [Socrate] est [mortel]

<sup>1</sup> D Cardon, JP Cointet, A Mazieres, 2018 [LRDN]

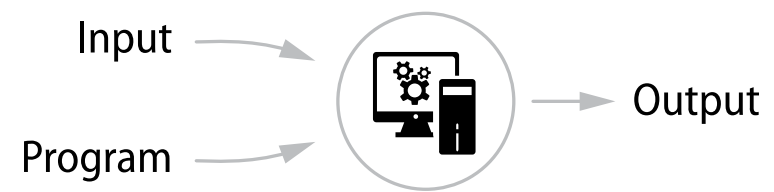
### Inductive approach



Connectionnism

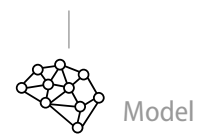
vs

### Deductive approach



Symbolic

Facts ► Rules and laws

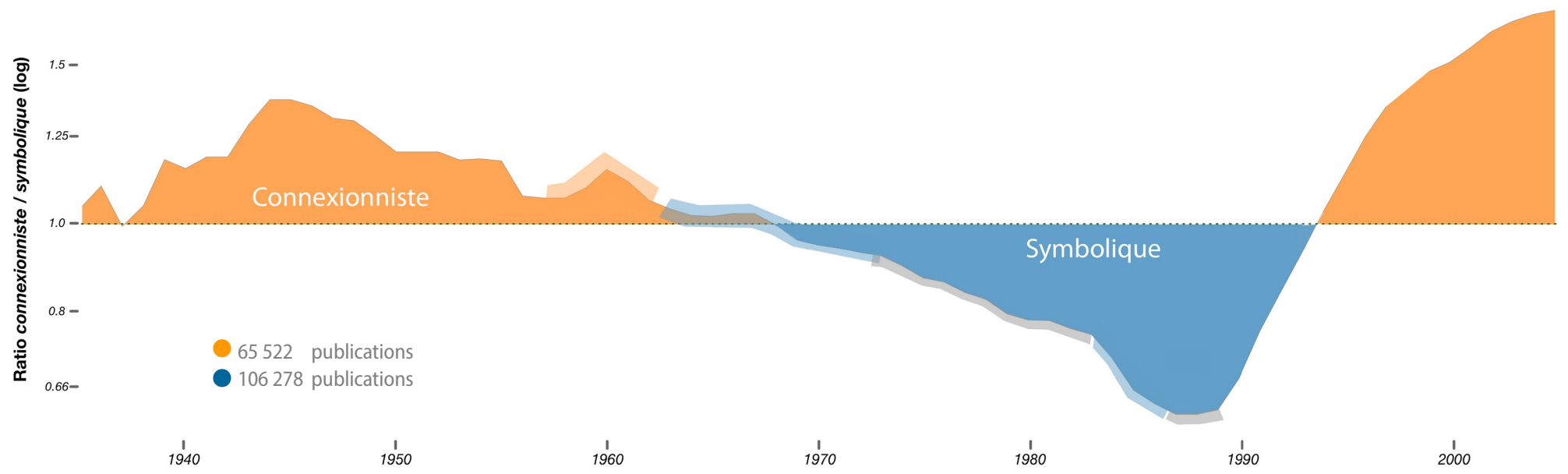


Expert

Rules and laws ► Special case

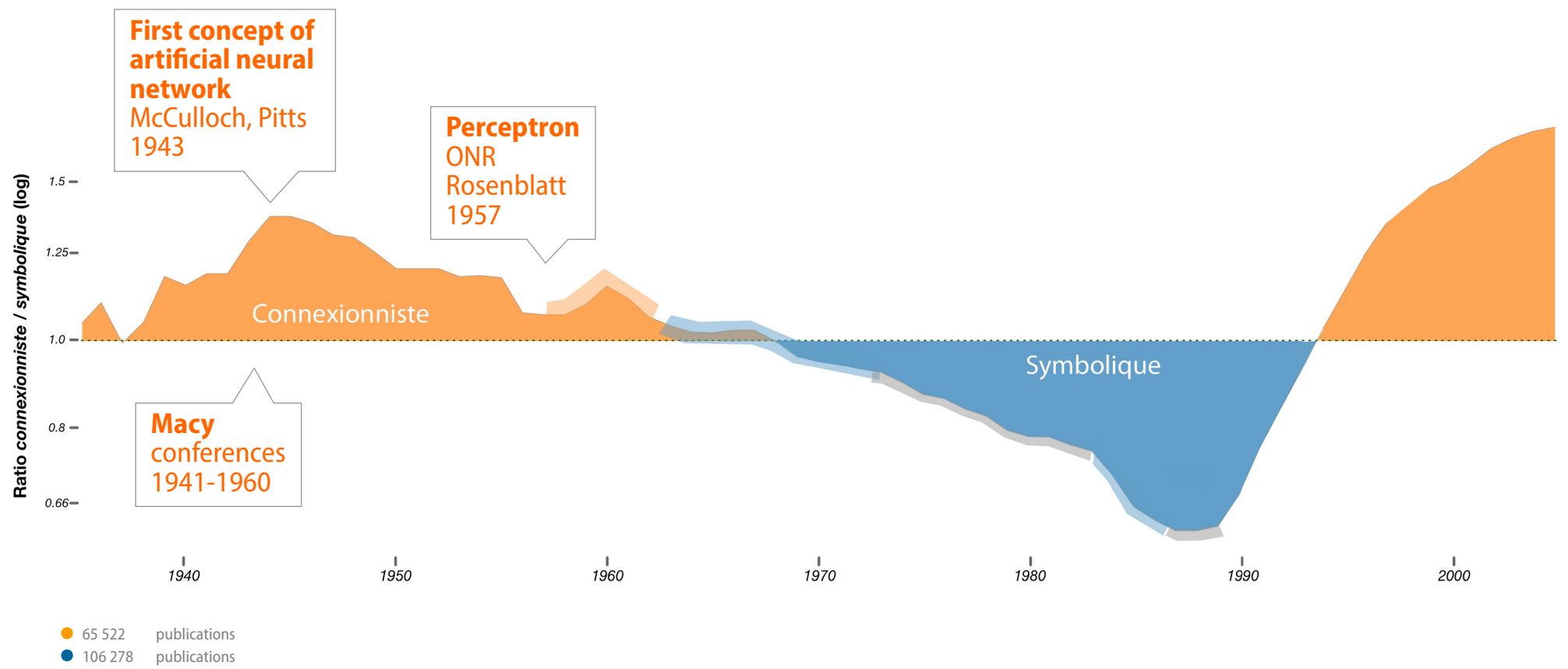
# Evolution of the academic influence of connexionist and symbolic approaches<sup>1</sup>

Ration of publications between connexionists and symbolists



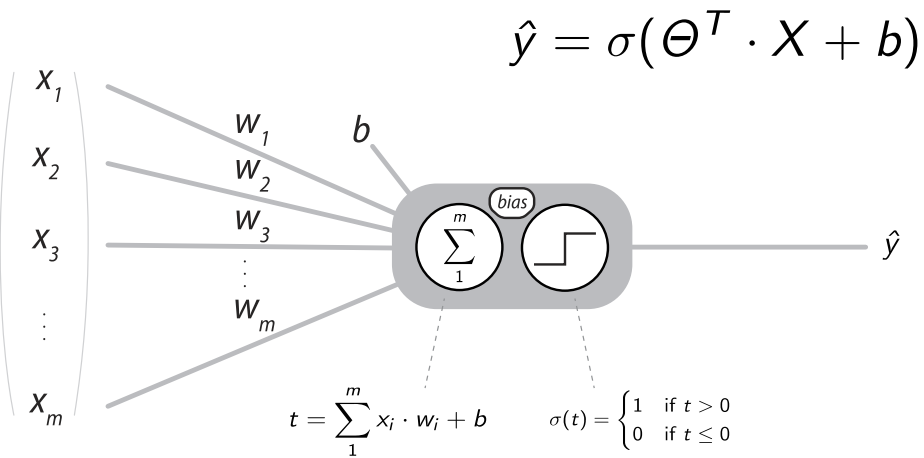
<sup>1</sup> D Cardon, JP Cointet, A Mazieres, 2018 [LRDN]

# Evolution of the academic influence of connexionist and symbolic approaches<sup>1</sup>



<sup>1</sup> D Cardon, JP Cointet, A Mazieres, 2018 [LRDN]

# Perceptron



Linear and binary classifier

## THE PERCEPTRON

389

sets of  
which are  
tend to  
t sets of

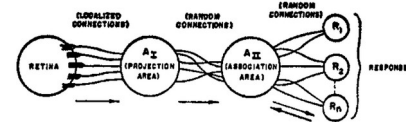


FIG. 1. Organization of a perceptron.

ve and/  
stimuli  
ay facili-  
ation of

The cells in the projection area each receive a number of connections from

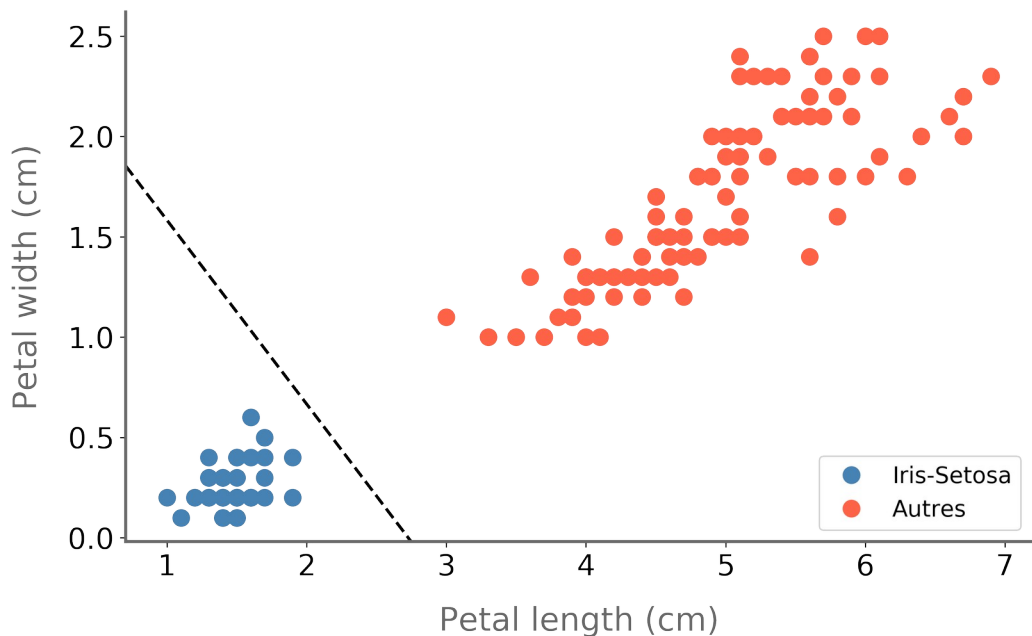
Perceptron  
Frank Rosenblatt  
1958





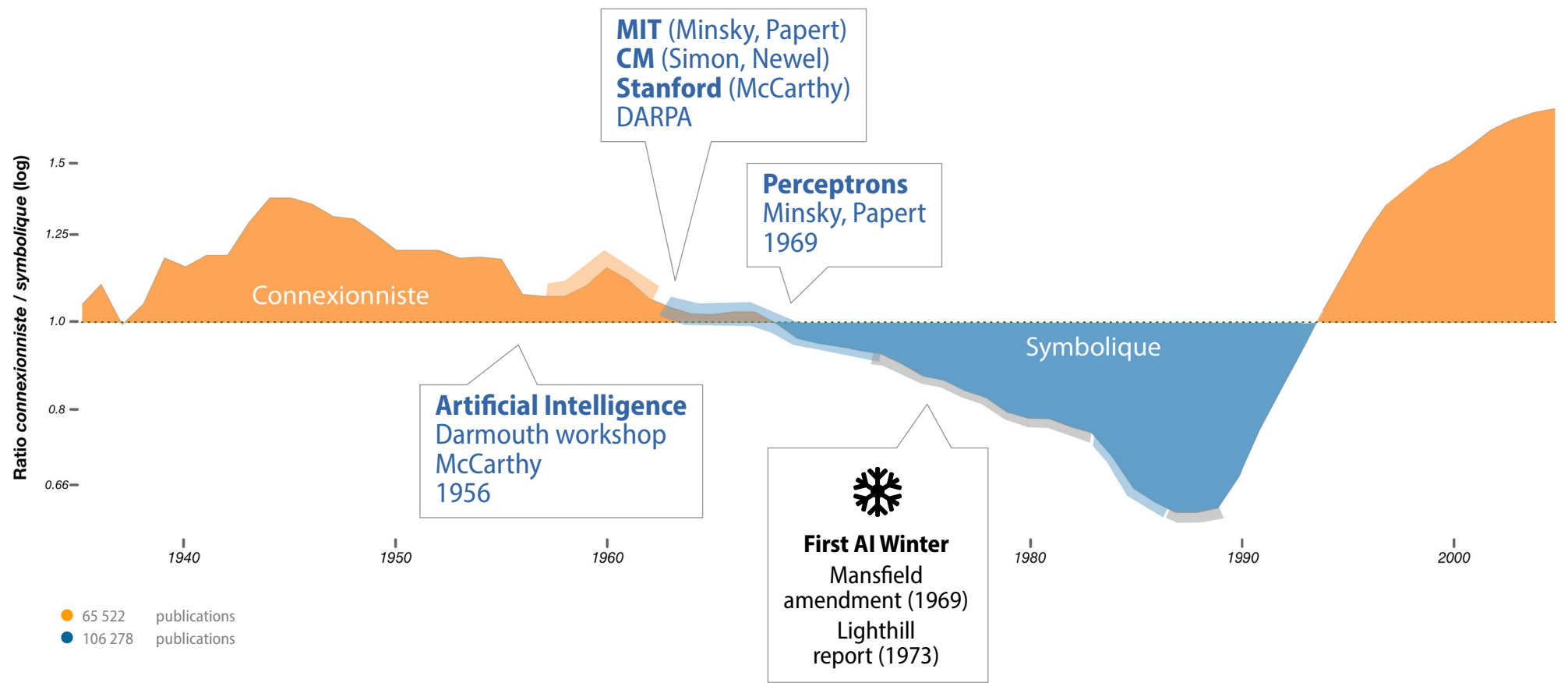
## Iris plants dataset

Dataset from : Fisher, R.A. "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188 (1936)



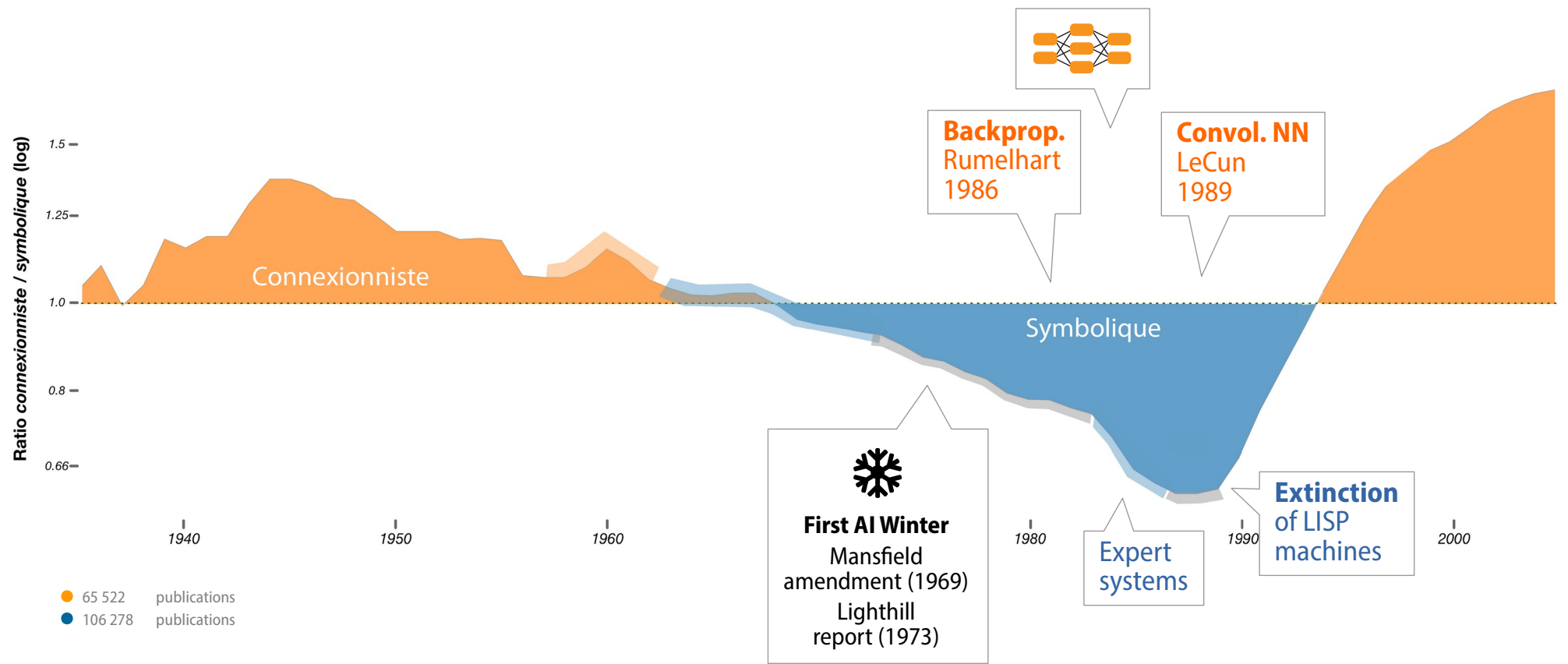
Length	Width	Iris Setosa (0/1)
$x_1$	$x_2$	$y$
1.4	1.4	1
1.6	1.6	1
1.4	1.4	1
1.5	1.5	1
1.4	1.4	1
4.7	4.7	0
4.5	4.5	0
4.9	4.9	0
4.0	4.0	0
4.6	4.6	0
(...)		

# Evolution of the academic influence of connexionist and symbolic approaches<sup>1</sup>



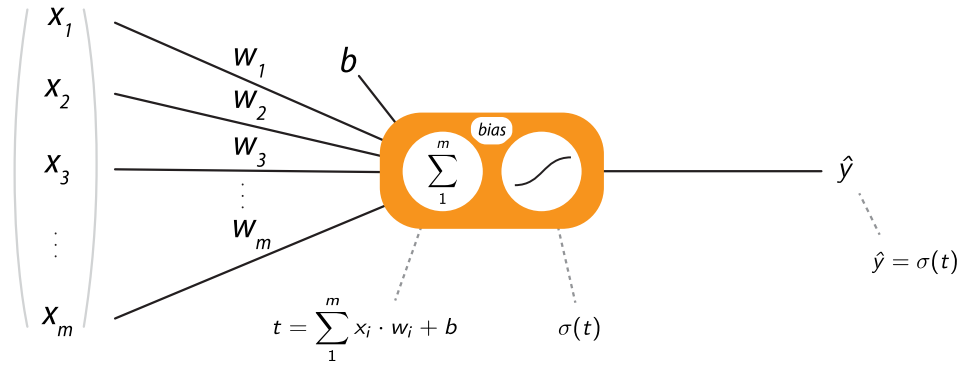
<sup>1</sup> D Cardon, JP Cointet, A Mazieres, 2018 [LRDN]

# Evolution of the academic influence of connexionist and symbolic approaches<sup>1</sup>

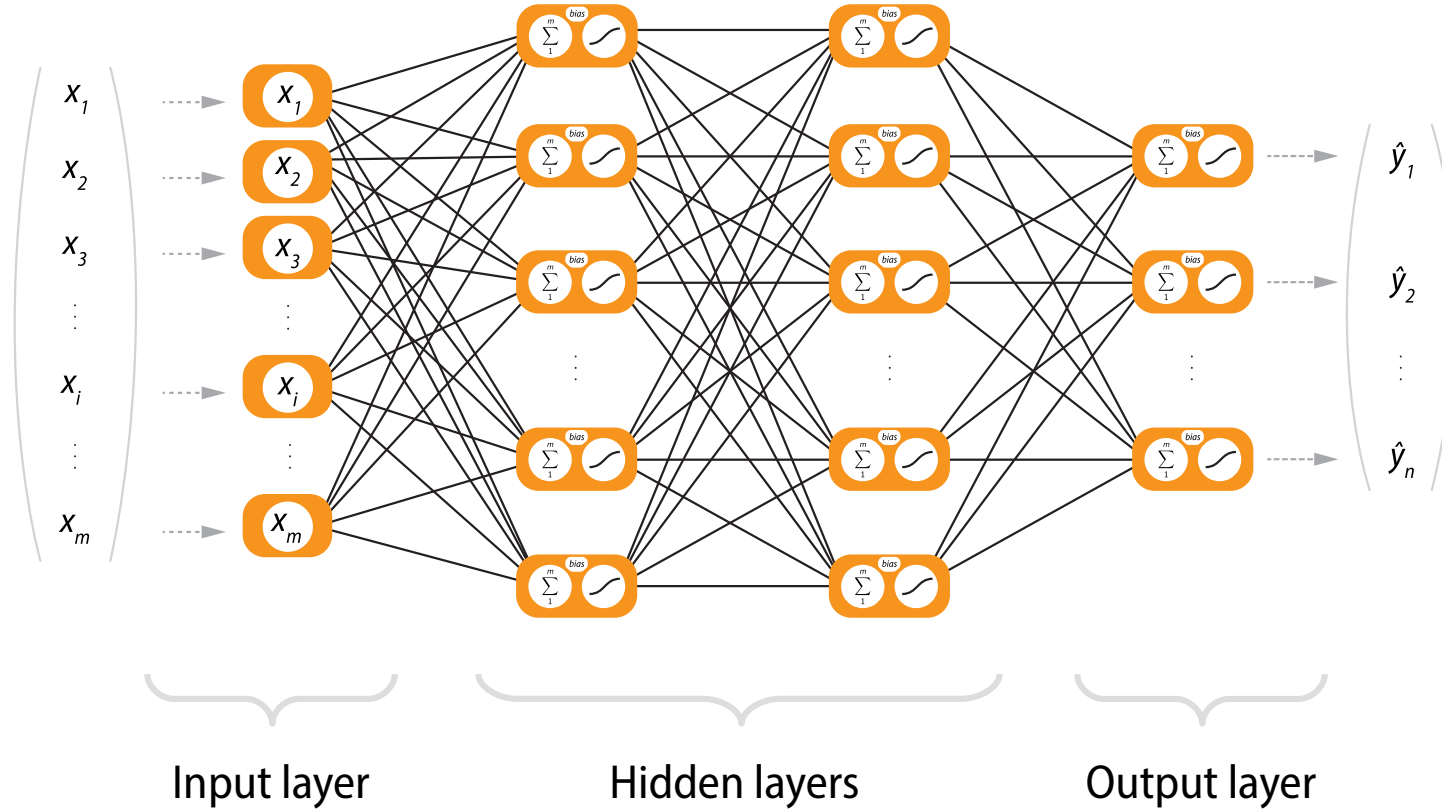


<sup>1</sup> D Cardon, JP Cointet, A Mazieres, 2018 [LRDN]

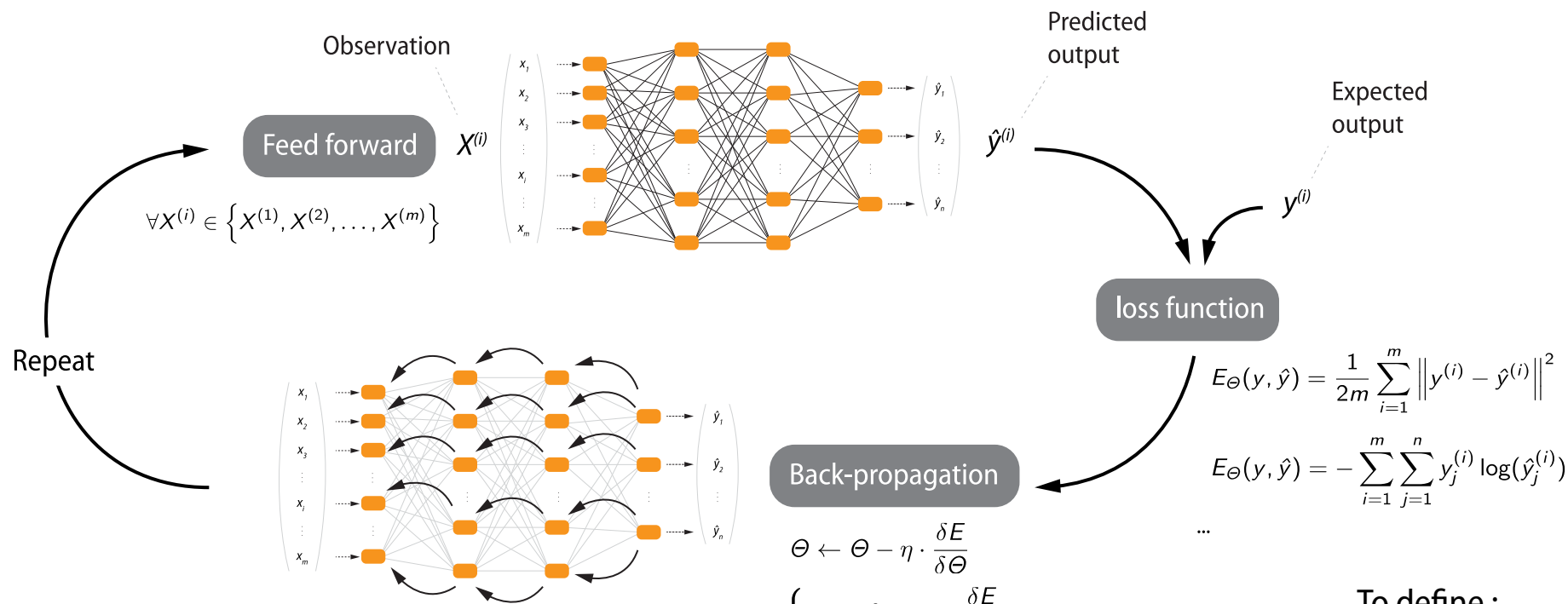
# Deep Neural Networks



# Deep Neural Networks



# Deep Neural Networks



$$E_{\Theta}(y, \hat{y}) = \frac{1}{2m} \sum_{i=1}^m \|y^{(i)} - \hat{y}^{(i)}\|^2$$

$$E_{\Theta}(y, \hat{y}) = - \sum_{i=1}^m \sum_{j=1}^n y_j^{(i)} \log(\hat{y}_j^{(i)})$$

...

$$\Theta \leftarrow \Theta - \eta \cdot \frac{\delta E}{\delta \Theta}$$

$$\begin{cases} m \leftarrow \beta m - \eta \cdot \frac{\delta E}{\delta \Theta} \\ \Theta \leftarrow \Theta + m \end{cases}$$

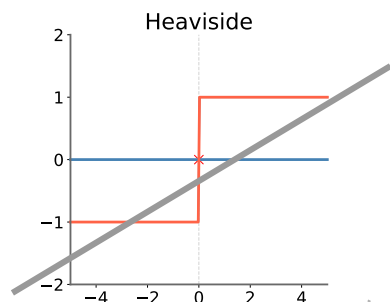
...

$\eta$  : Learning rate  
 $\beta$  : Momentum

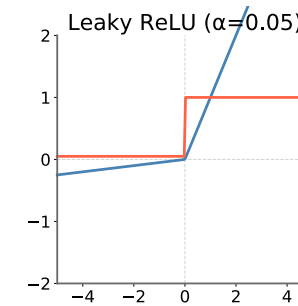
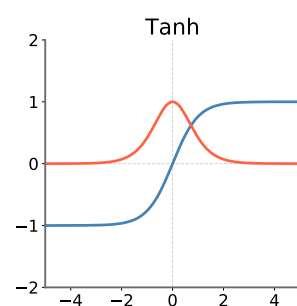
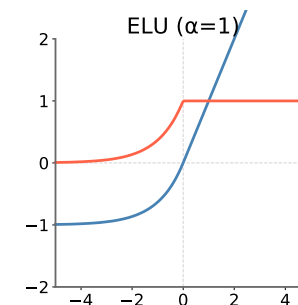
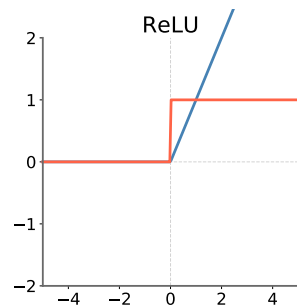
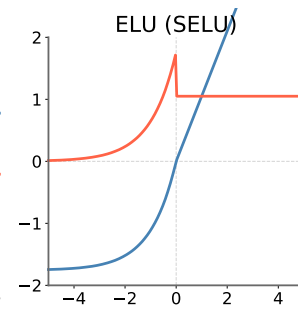
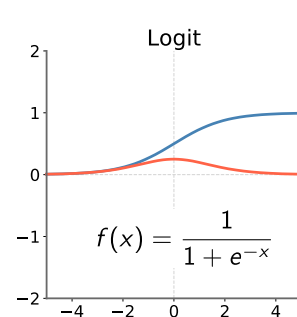
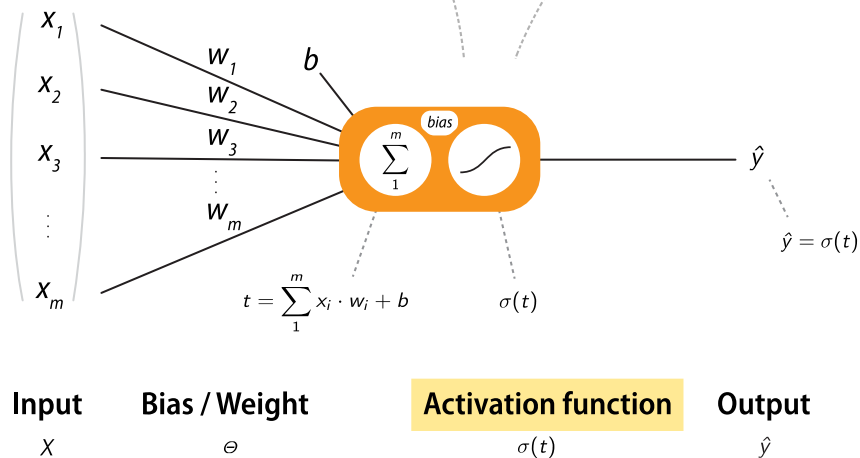
To define :  
Optimization  
Activation  
Loss  
Metrics  
...

Back-propagation  
Learning process

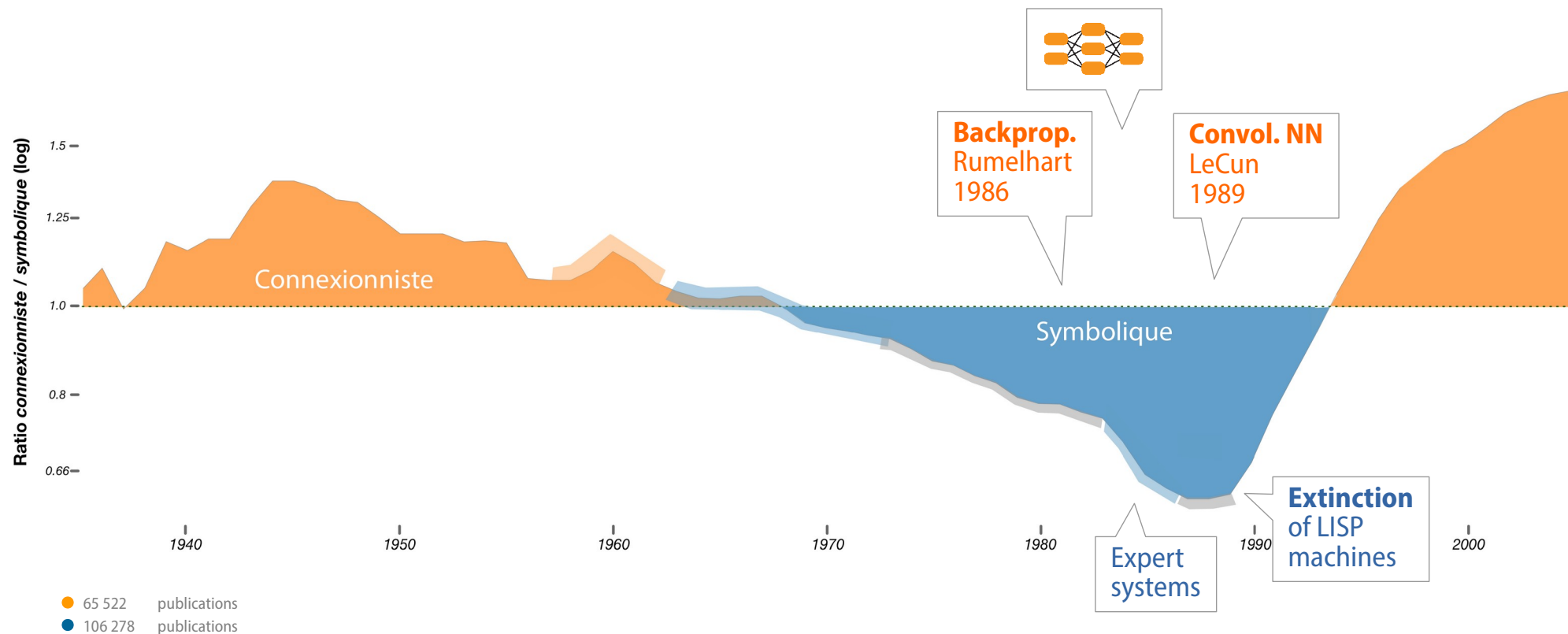
# Deep Neural Networks



1958



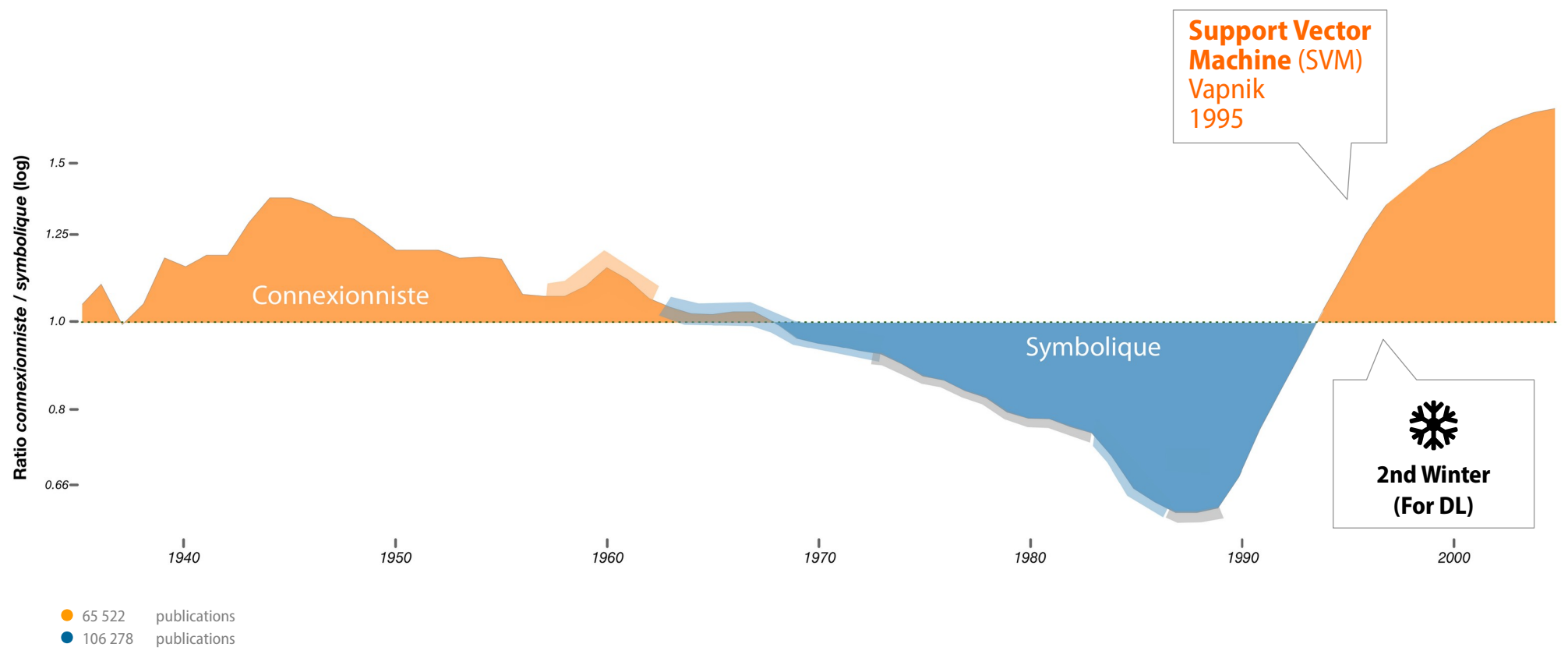
# Evolution of the academic influence of connexionist and symbolic approaches<sup>1</sup>



<sup>1</sup> D Cardon, JP Cointet, A Mazieres, 2018 [LRDN]

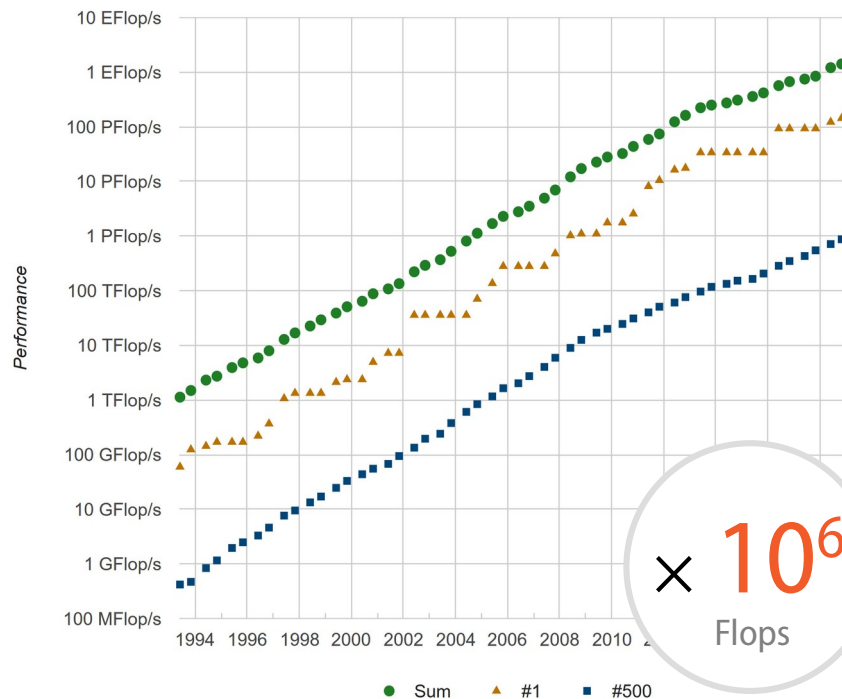


# Evolution of the academic influence of connexionist and symbolic approaches<sup>1</sup>



<sup>1</sup> D Cardon, JP Cointet, A Mazieres, 2018 [LRDN]

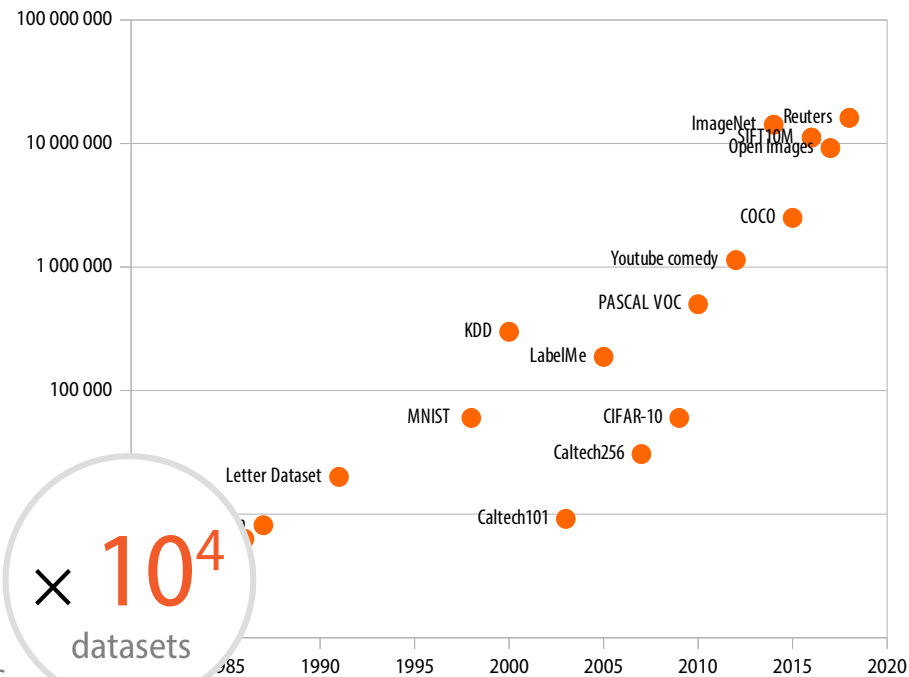
## Performance Development<sup>1</sup>



× 10<sup>6</sup>  
Flops

25 ans

## Datasets for machine-learning<sup>2</sup>



× 10<sup>4</sup>  
datasets

Laboratoire  
Cas particulier

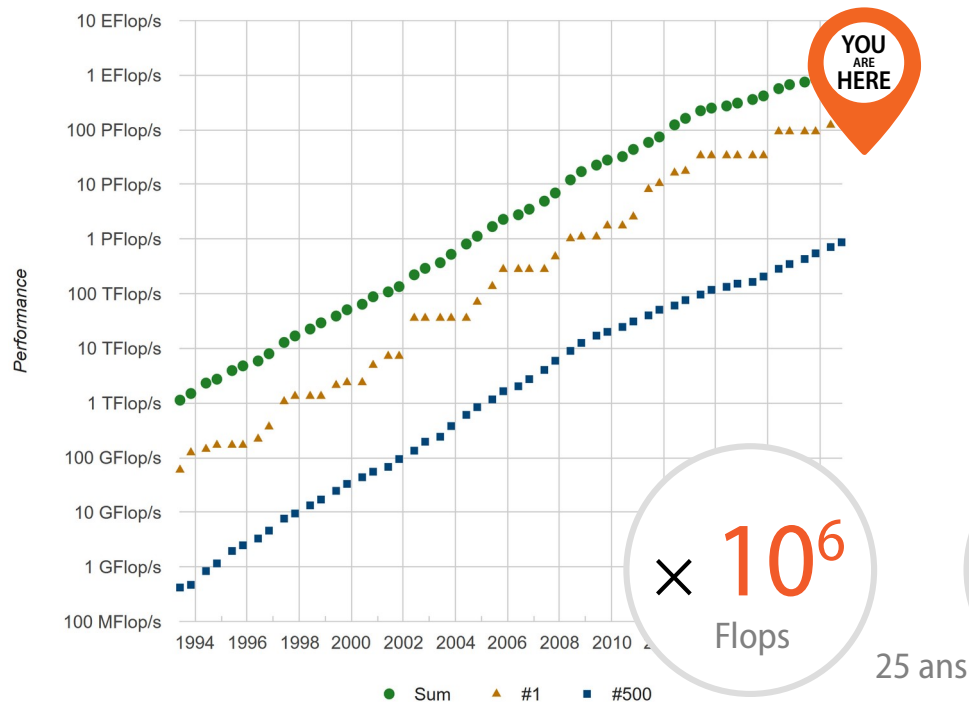


Monde réel

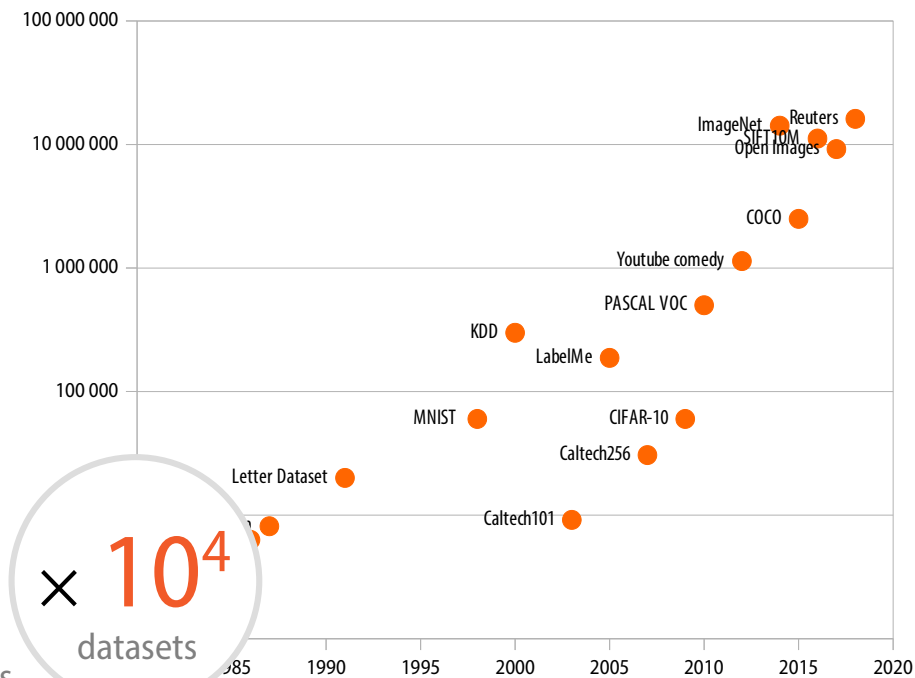
<sup>1</sup> TOP500 List [TOP500]

<sup>2</sup> Wikipedia [WKP1]

## Performance Development<sup>1</sup>



## Datasets for machine-learning<sup>2</sup>



Laboratoire  
Cas particulier

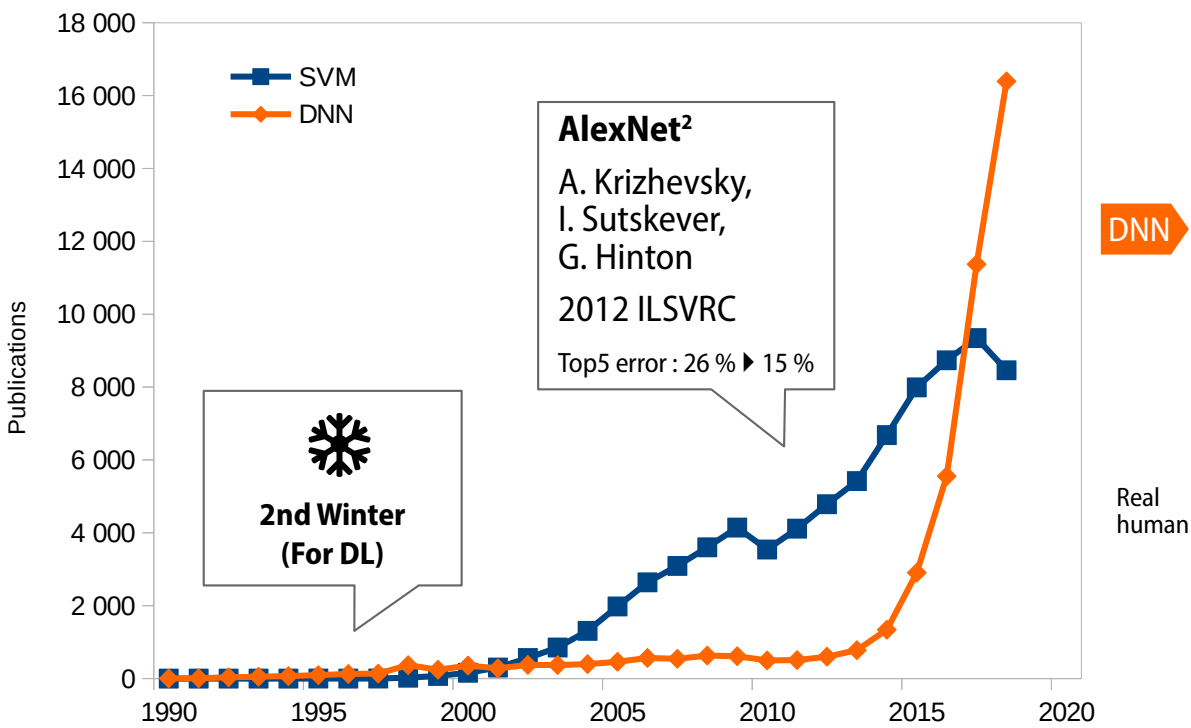


Monde réel

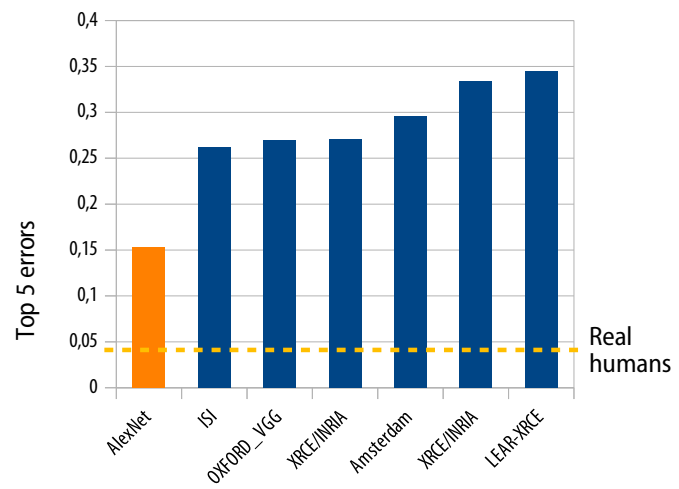
<sup>1</sup> TOP500 List [TOP500]

<sup>2</sup> Wikipedia [WKP1]

# Publications SVM vs DNN<sup>1</sup>



## Images classification Top 5 error at ILSVRC 2012<sup>3,4</sup>



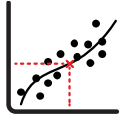
Without mathematical guarantee, DNN have proven to be more effective in the face of the **complexity of the real world** !

<sup>1</sup> Web of Science [WOS1][WOS2]  
<sup>2</sup> AlexNet [ALEX]  
<sup>3</sup> ImageNet Large Scale Visual Recognition [ILSVRC]

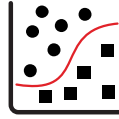
<sup>4</sup> Similar evolution in Natural language processing, translation, board games, etc.  
See : DeepL.com, AlphaGo, AlphaZero, ...

# Data and neurons





**Basic  
Regression**  
DNN



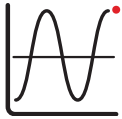
**Basic  
Classification**  
DNN



**Hight  
Dimensionnal Data**  
(images, ...)  
CNN



**Sparse data**  
(text, ...)  
Embedding



**Sequences data**  
(Time data, ...)  
RNN



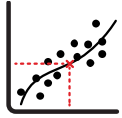
**Reinforcement**  
learning



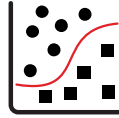
**Variational  
Autoencoder**  
VAE



**Generative  
Adversarial  
Network**  
GAN



**Basic  
Regression**  
DNN



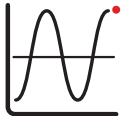
**Basic  
Classification**  
DNN



**Hight  
Dimensionnal Data**  
(images, ...)  
CNN



**Sparse data**  
(text, ...)  
Embedding



**Sequences data**  
(Time data, ...)  
RNN



**Reinforcement**  
learning

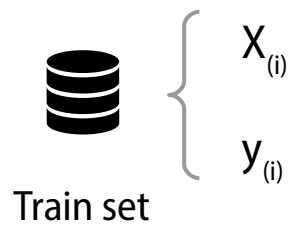


**Variational  
Autoencoder**  
VAE



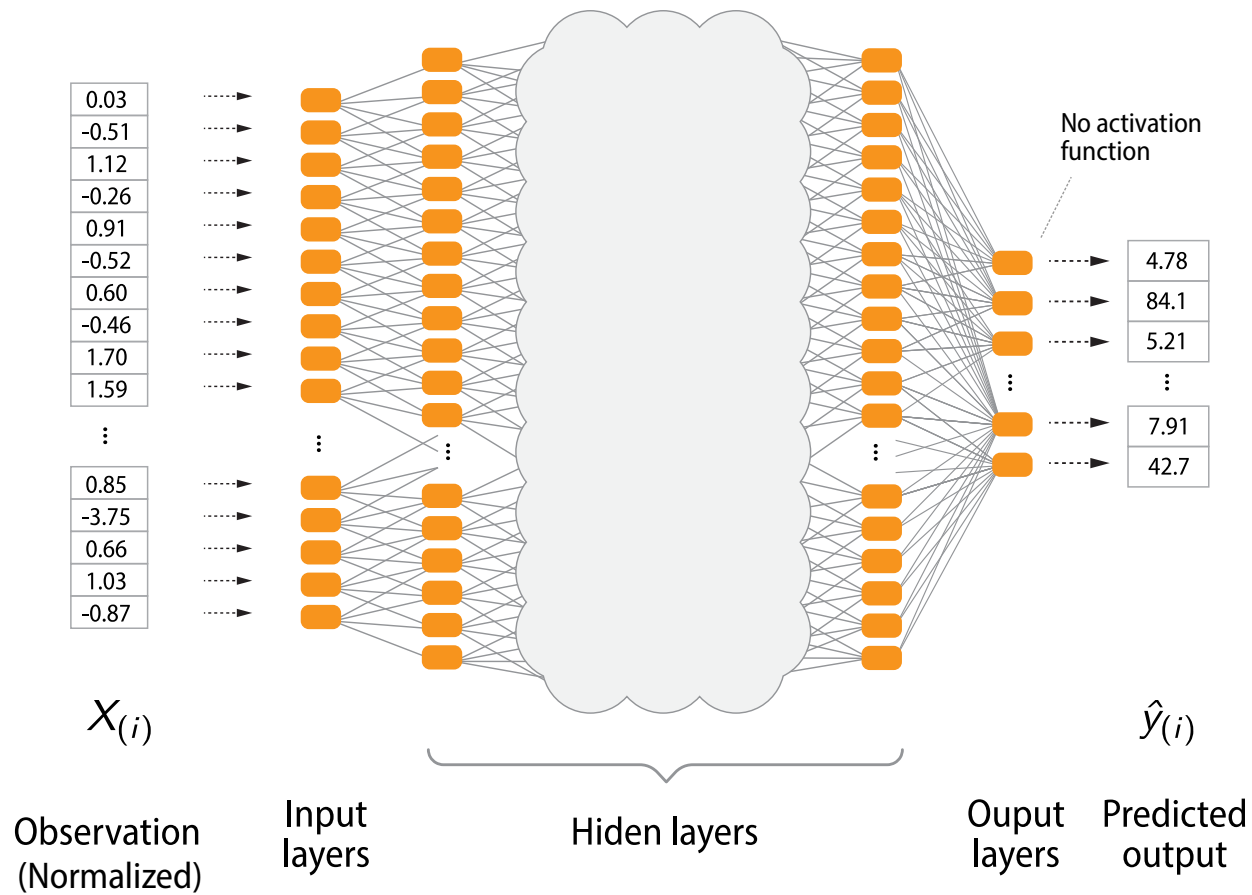
**Generative  
Adversarial  
Network**  
GAN

# Regression with a DNN



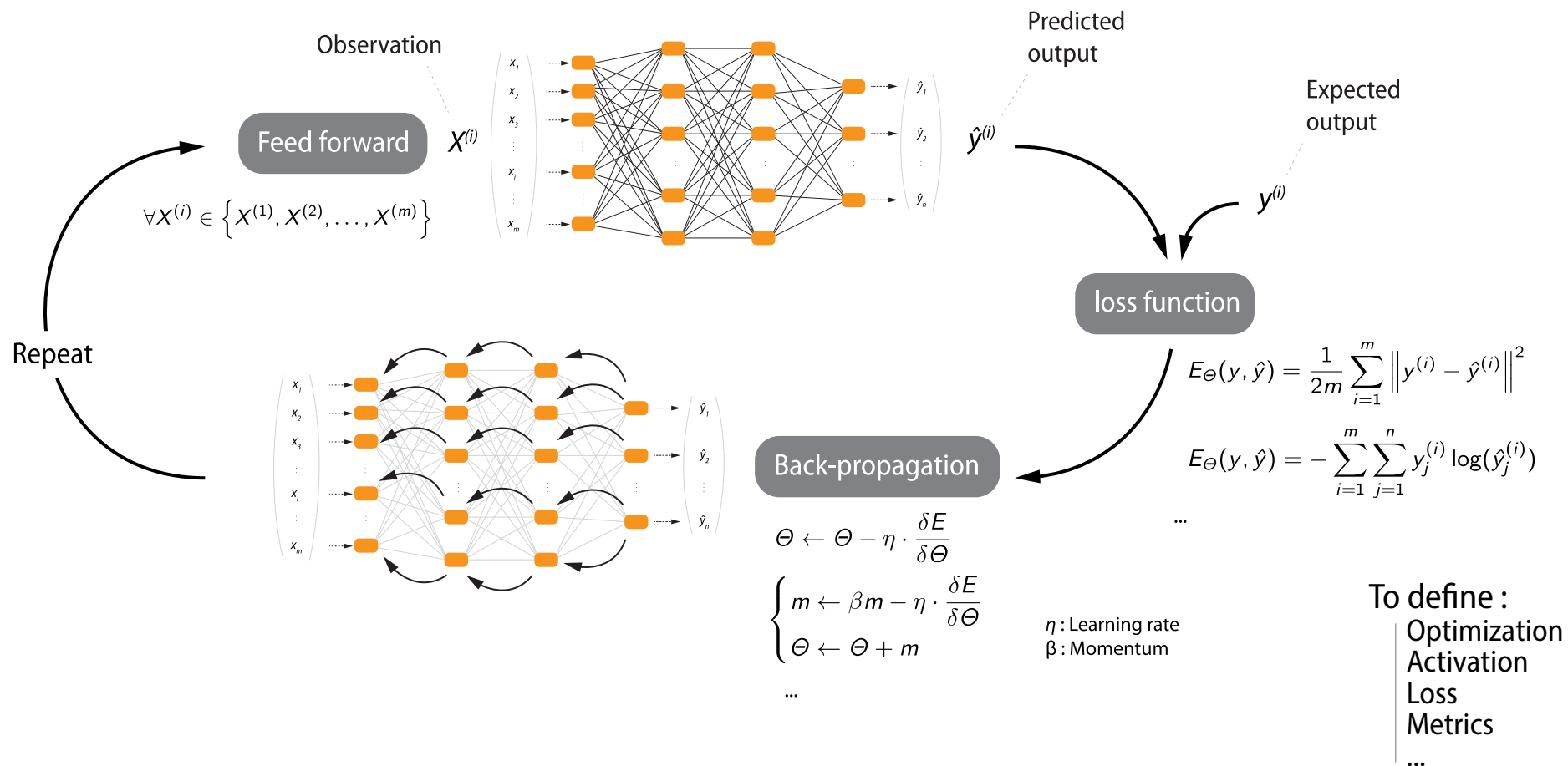
$X_{(i)}$  : Observations

$y_{(i)}$  : Expected output

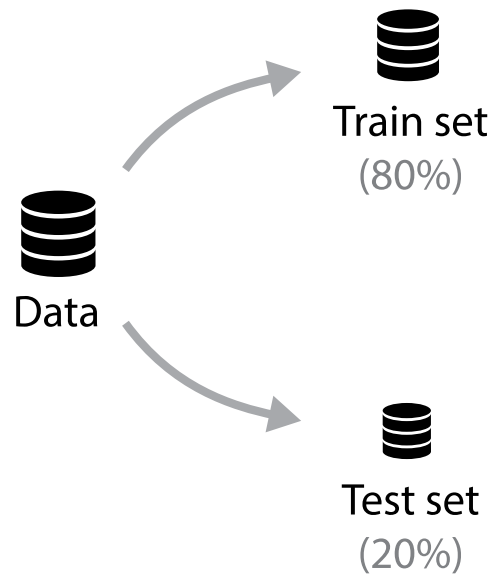




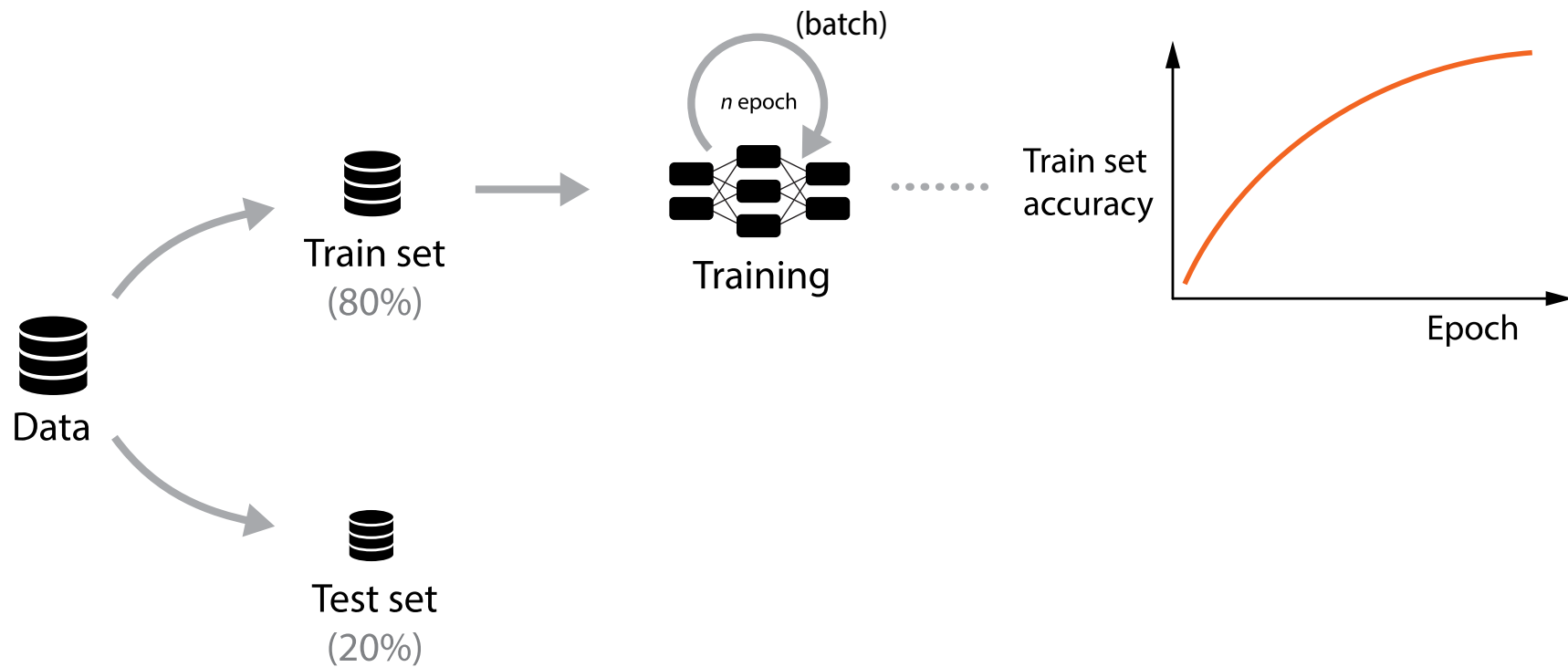
# Training process - general



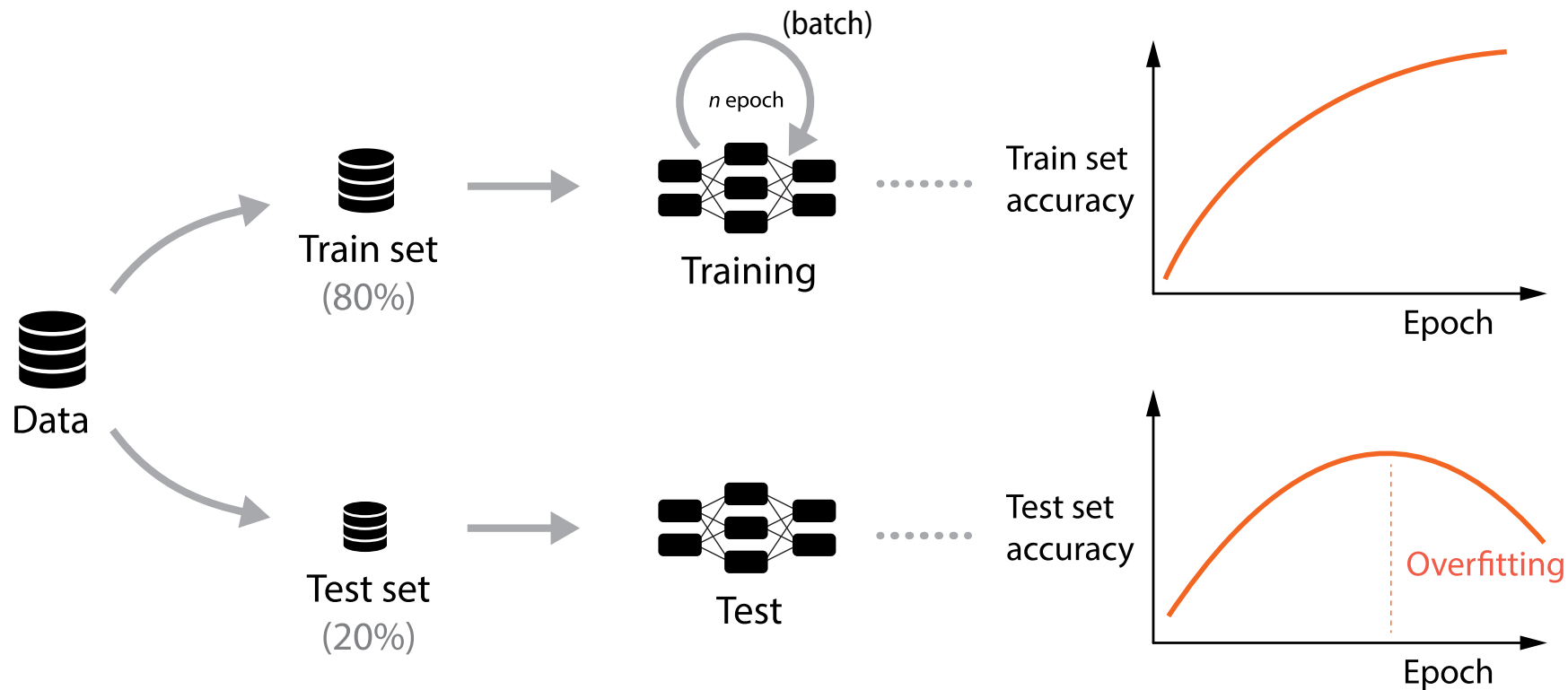
# Training process - general



# Training process - general



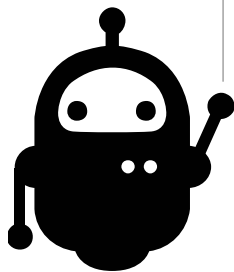
# Training process - general





## Regression with a Dense Network (DNN)

Notebook : [\[BHP1\]](#)



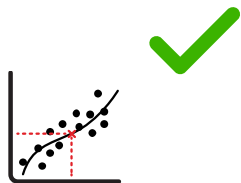
### **Objective :**

Predicts housing prices from a set of house features.

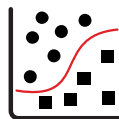
### **Dataset :**

Boston House Pricing Dataset (BHPD)





**Basic  
Regression**  
DNN



**Basic  
Classification**  
DNN



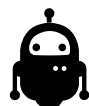
**Hight  
Dimensionnal Data**  
(images, ...)  
CNN



**Sparse data**  
(text, ...)  
Embedding



**Sequences data**  
(Time data, ...)  
RNN



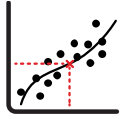
**Reinforcement**  
learning



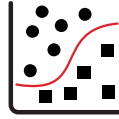
**Variational  
Autoencoder**  
VAE



**Generative  
Adversarial  
Network**  
GAN



**Basic  
Regression**  
DNN



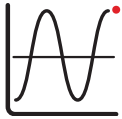
**Basic  
Classification**  
DNN



**Hight  
Dimensionnal Data**  
(images, ...)  
CNN



**Sparse data**  
(text, ...)  
Embedding



**Sequences data**  
(Time data, ...)  
RNN



**Reinforcement**  
learning

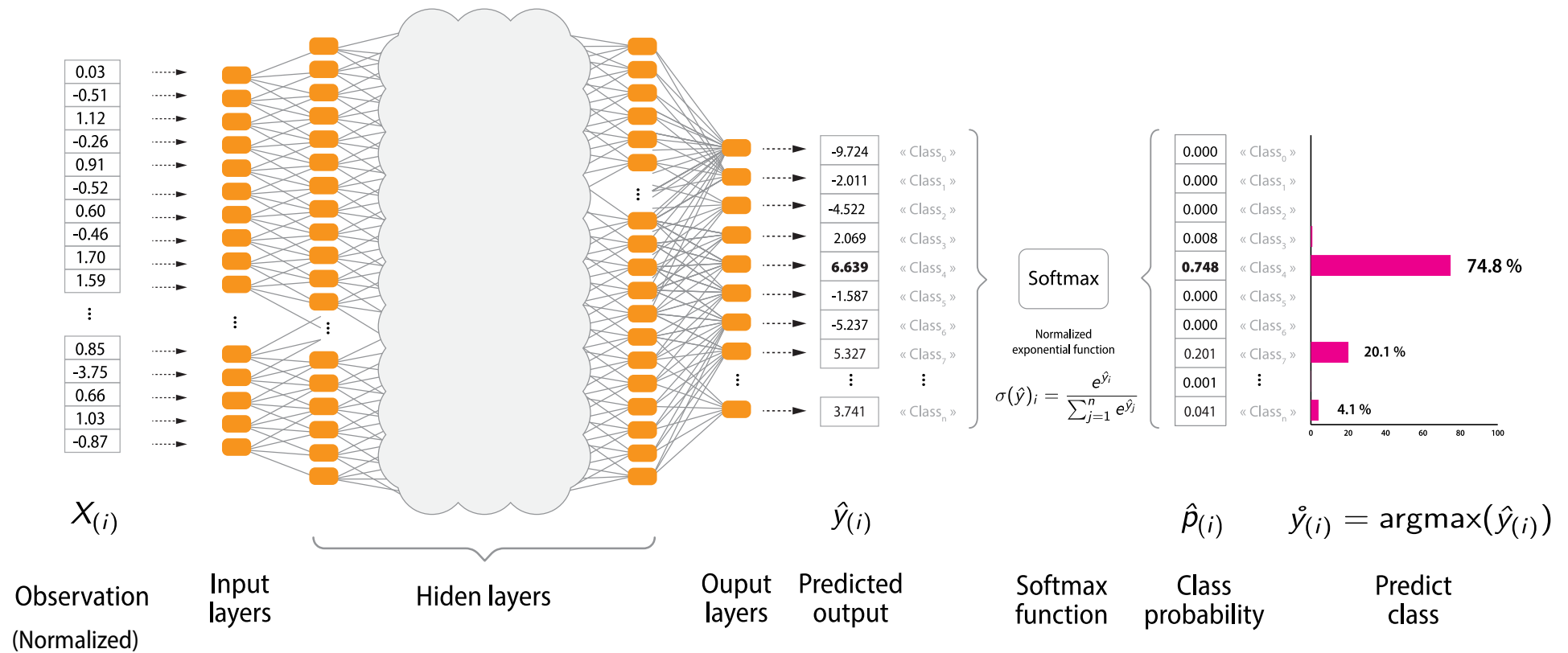


**Variational  
Autoencoder**  
VAE



**Generative  
Adversarial  
Network**  
GAN

# Classification with a DNN

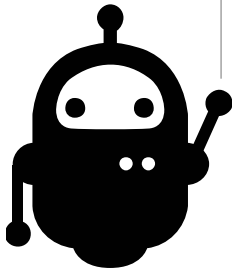






# Simple classification with DNN

Notebook : [\[MNIST1\]](#)



## **Objective :**

Recognizing handwritten numbers

## **Dataset :**

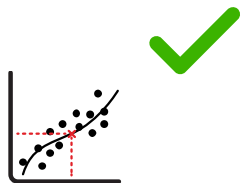
Modified National Institute of Standards and Technology (MNIST)



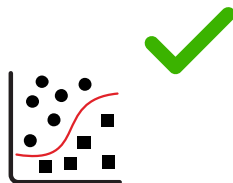


Little things and concepts to **keep in mind**

- Regression vs. Classification
- Data normalization
- Training and validation
- Epochs and Batches
- Activation functions
- Loss function
- Optimization functions – Gradient descent
- Metrics
- Softmax and Argmax function
- Numpy shape



**Basic  
Regression**  
DNN



**Basic  
Classification**  
DNN



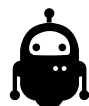
**Hight  
Dimensionnall Data**  
(images, ...)  
CNN



**Sparse data**  
(text, ...)  
Embedding



**Sequences data**  
(Time data, ...)  
RNN



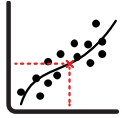
**Reinforcement  
learning**



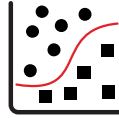
**Variational  
Antoencoder**  
VAE



**Generative  
Adversarial  
Network**  
GAN



**Basic  
Regression**  
DNN



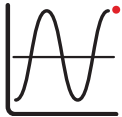
**Basic  
Classification**  
DNN



**Hight  
Dimensionnal Data**  
(images, ...)  
CNN



**Sparse data**  
(text, ...)  
Embedding



**Sequences data**  
(Time data, ...)  
RNN



**Reinforcement**  
learning



**Variational  
Autoencoder**  
VAE



**Generative  
Adversarial  
Network**  
GAN

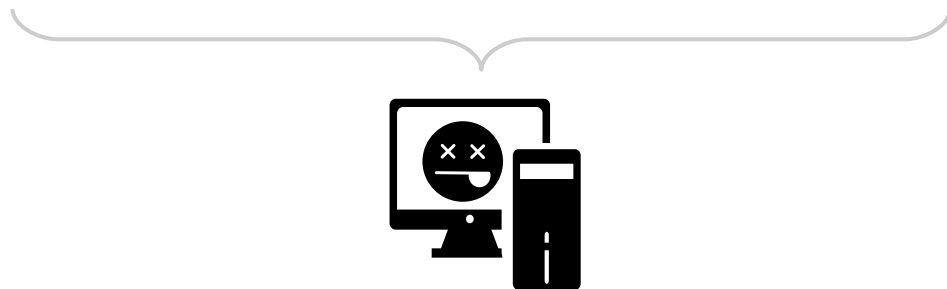
# Convolutional Neural Networks (CNN)



24 M pixels  
(r,v,b) 3x8 bits



3 x 24 M neurons ?!



10 000



70 M



100 Mds



1 000 000

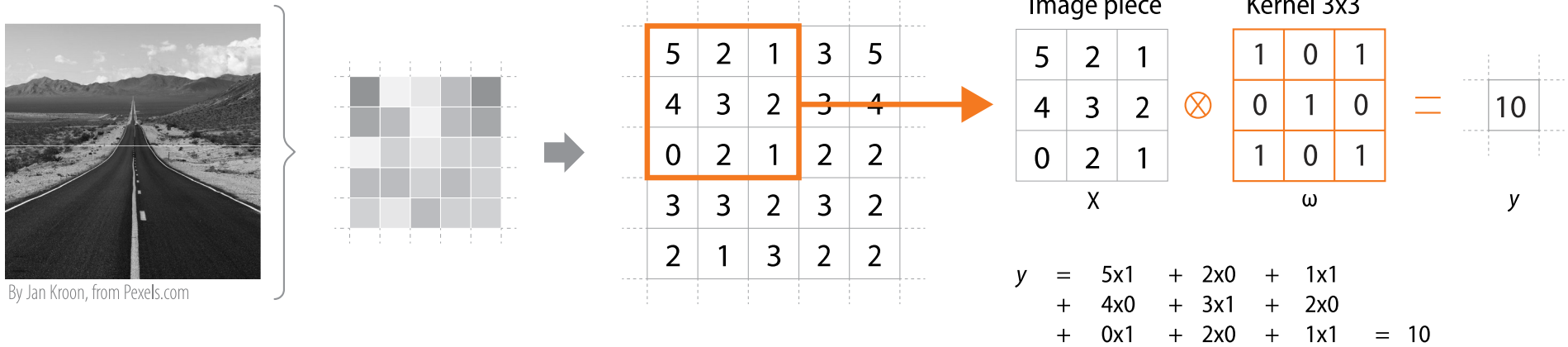


700 M



250 Mds

# Convolutional Neural Networks (CNN)

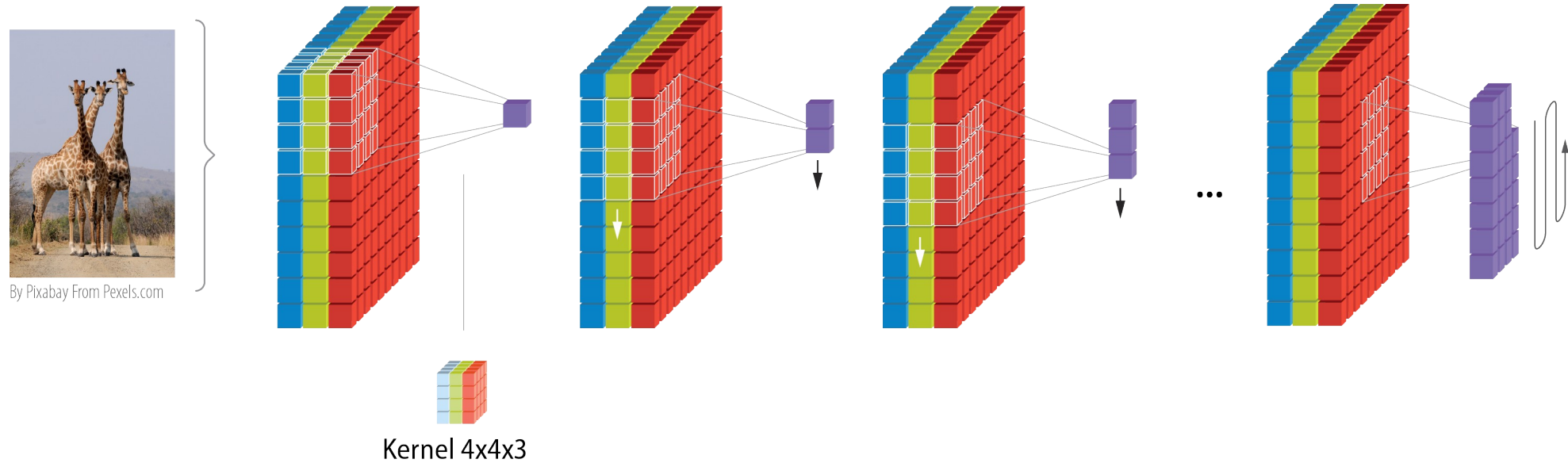


$$y = \sum_{i=1}^n \sum_{j=1}^m x_{i,j} \cdot \omega_{i,j} \quad \text{with} \quad \begin{cases} n & \text{kernel width} \\ m & \text{kernel height} \end{cases}$$

⊗ is Hadamard product

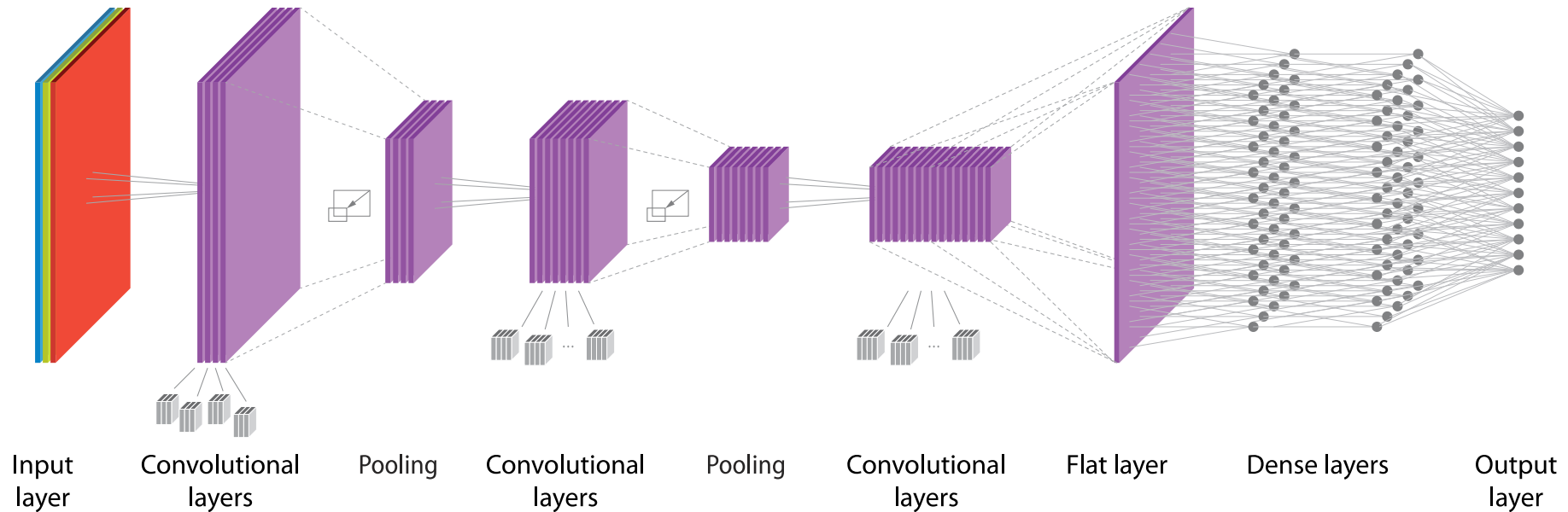
## 2D convolution

# Convolutional Neural Networks (CNN)



3D convolution

# Convolutional Neural Networks (CNN)

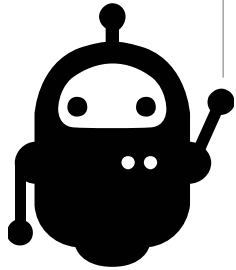






# CNN with GTSRB dataset

Notebook : [\[GTS1-7\]](#)



## **Objective :**

Recognizing traffic signs

## **Dataset :**

German Traffic Sign Recognition Benchmark (GTSRB) is a dataset with more than 50,000 photos of road signs from about 40 classes



# CNN with GTSRB dataset

Notebook : [\[GTS1-7\]](#)

Episode 1 : Data analysis and creation of a **usable dataset**

Episode 2 : First **convolutions** and first results

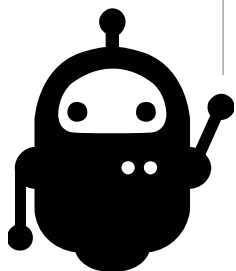
Episode 3 : **Monitoring** training, managing checkpoints

Episode 4 : Improving the results with **data augmentation**

Episode 5 : A lot of models, a lot of datasets and a lot of results.

Episode 6 : Run Full convolution notebook as a **batch**

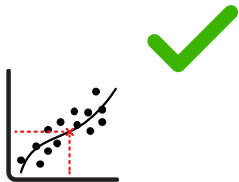
Episode 7 : Displaying the **reports** of the different jobs



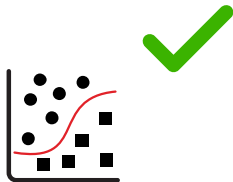


## Little things and concepts to **keep in mind**

- Understand the data !
- Organize and prepare our data
- Lots of small data = big problems
- Store our data, h5 files
- Finding the right model isn't easy
- Principle of hyperparameters
- Follow the training (Tensorboard...)
- Saving, retrieving and using recovery points
- Data augmentation
- Automate tests
- Batch mode submission



**Basic  
Regression**  
DNN



**Basic  
Classification**  
DNN



**Hight  
Dimensionnall Data**  
(images, ...)  
CNN



**Sparse data**  
(text, ...)  
Embedding



**Sequences data**  
(Time data, ...)  
RNN



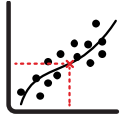
**Reinforcement  
learning**



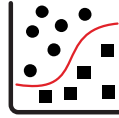
**Variational  
Antoencoder**  
VAE



**Generative  
Adversarial  
Network**  
GAN



**Basic  
Regression**  
DNN



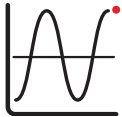
**Basic  
Classification**  
DNN



**Hight  
Dimensionnal Data**  
(images, ...)  
CNN



**Sparse data**  
(text, ...)  
Embedding



**Sequences data**  
(Time data, ...)  
RNN



**Reinforcement**  
learning



**Variational  
Autoencoder**  
VAE

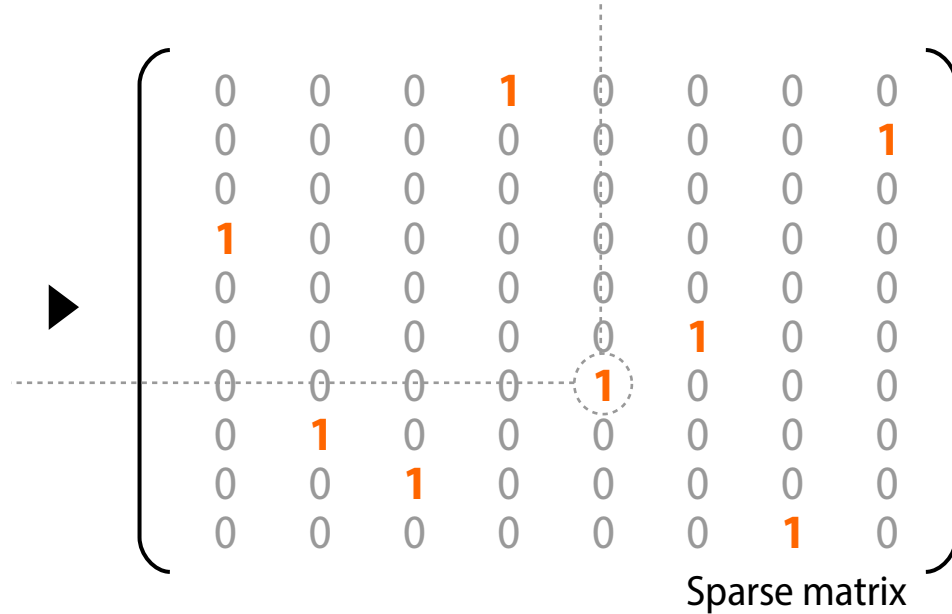


**Generative  
Adversarial  
Network**  
GAN

# Word Embedding

« I've never seen a movie like this before. »

1	a
2	before
3	fantastic
4	i've
5	is
6	like
7	movie
8	never
9	seen
10	this

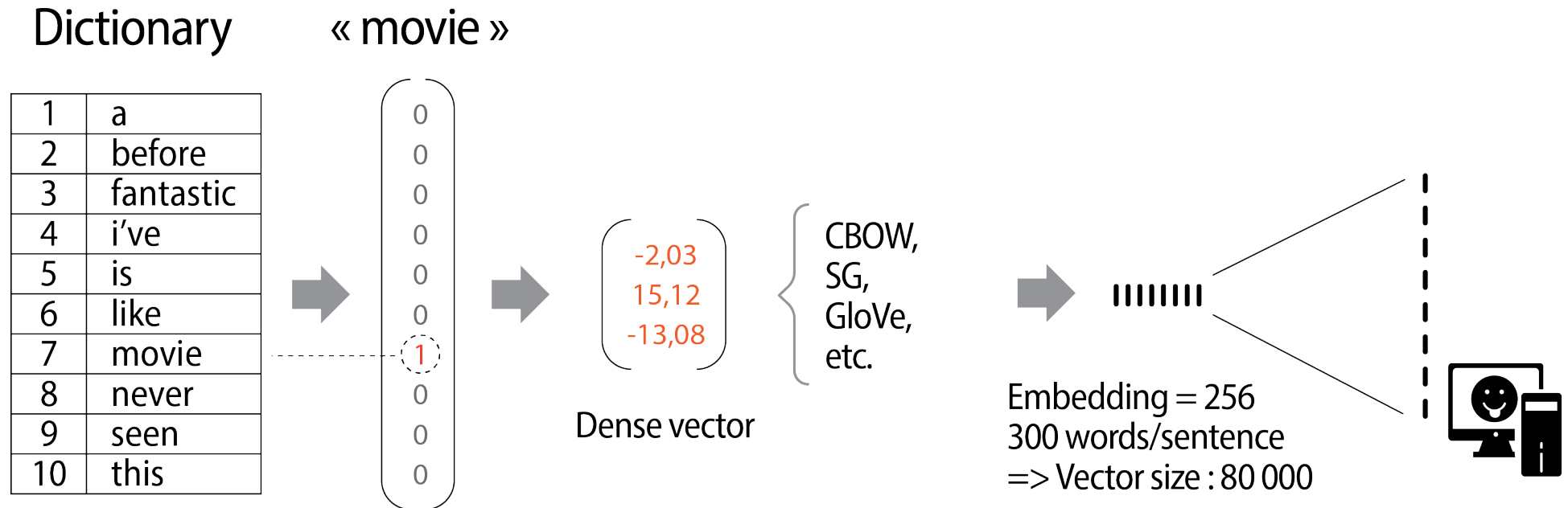


Dictionary = 80 000  
Sentence = 300

Vectors = 24 M



# Word Embedding



## Embedding layers in Keras



Utilisable comme une **simple couche**

Cette couche va constituer un **dictionnaire de vecteurs** qu'elle optimisera au cours de l'apprentissage, en **fonction du résultat** attendu et non de la sémantique pure.

L'embedding Keras est donc adapté à la **classification**, mais ne rendra pas compte, par exemple, de similarités sémantiques (comme identifier 2 phrases ayant un même sens sémantique).

La **sortie** de la couche est un **ensemble de vecteurs**



## Word2Vec<sup>1</sup>



Approche ayant pour objectif de constituer des **dictionnaires** dont la représentation vectorielle des mots est basée sur le **contexte** et donc de la **sémantique**.

Des dictionnaires construits à partir de gros corpus sont disponibles.

Deux modèles :

- **Continuous Bag-of-Words (CBOW)**,
- **Skip-Gram (SG)**.

<sup>1</sup> Tomas Mikolov & all, (2013), [W3VEC]

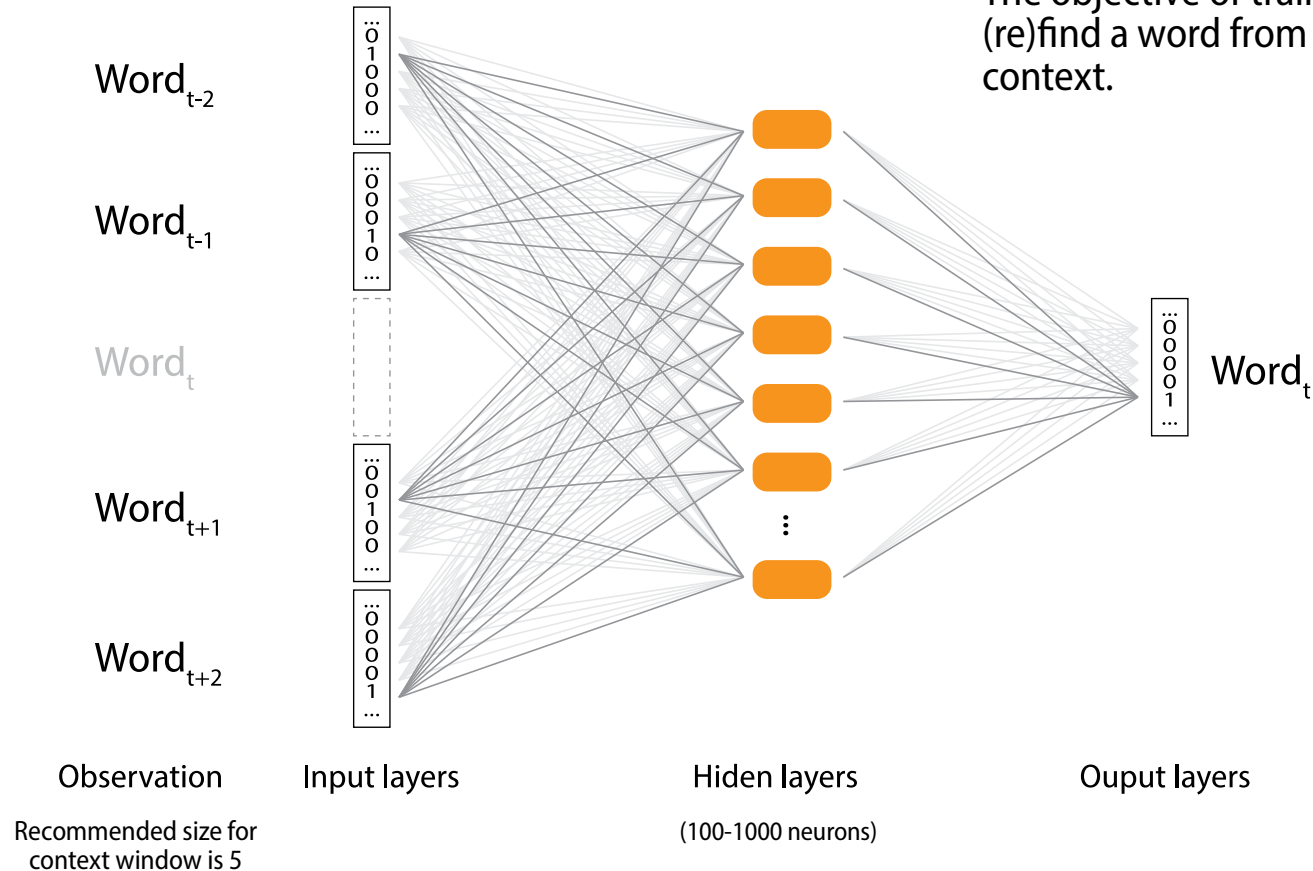
CBOW : Continuous Bag of Words - Embedding based on the prediction of the word according to its context.

SG : Skip-gram - Embedding based on context prediction from the word.

# Word Embedding

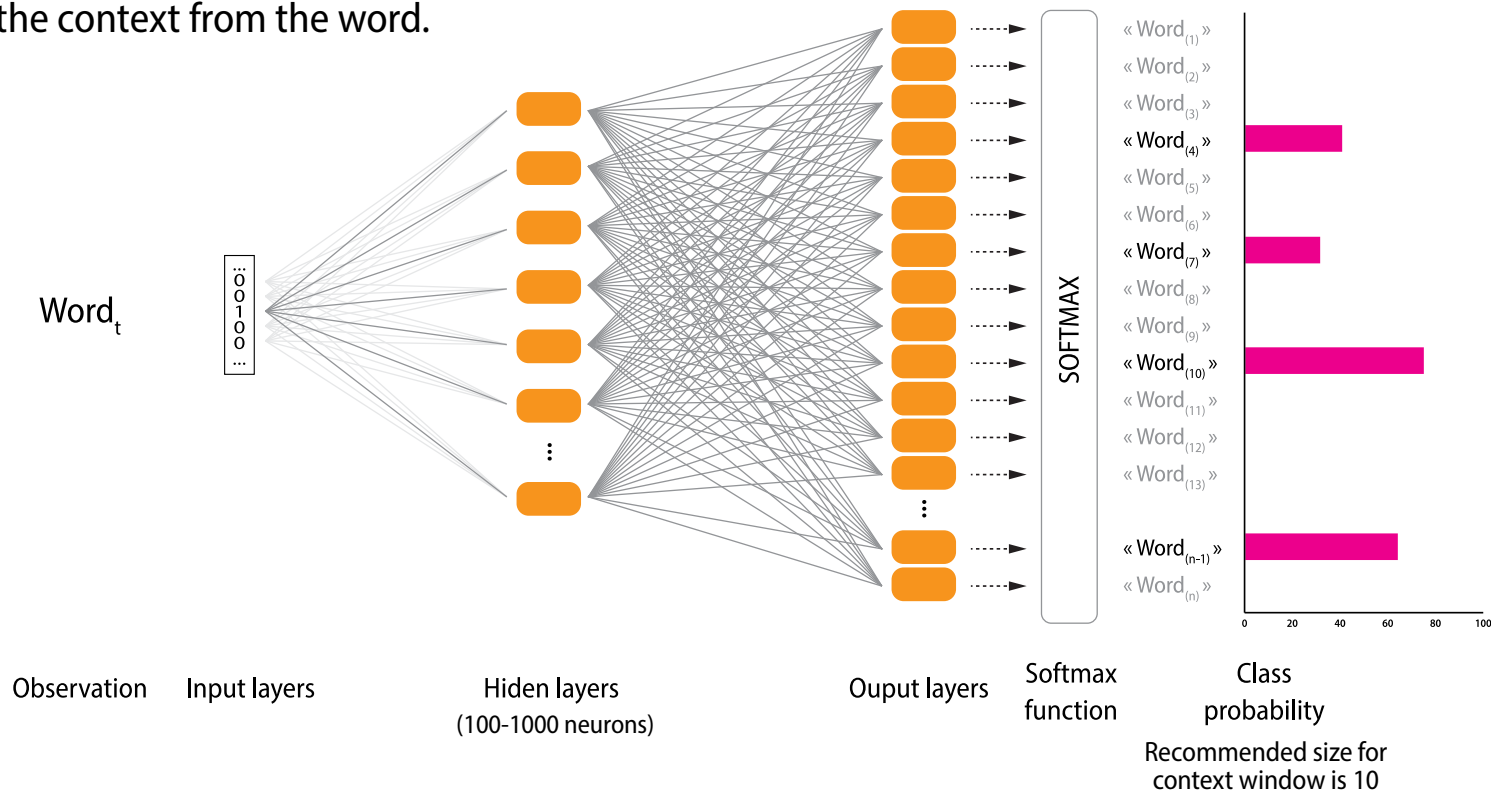
## Continuous Bag-of-Words (CBOW)

The objective of training is to (re)find a word from its context.



## Skip-Gram

The objective of training is to (re)find the context from the word.



## GloVe<sup>1</sup>



Contrairement à Word2vec, GloVe ne repose pas uniquement sur des statistiques locales (informations sur le contexte local des mots), mais intègre des **statistiques globales** (co-occurrence des mots) pour obtenir des vecteurs de mots.

<sup>1</sup> Jeffrey Pennington & all, (2014), [GLOVE]  
Training is performed on aggregated global word-word co-occurrence statistics

## (Flau)BERT<sup>1</sup>

2018



BERT est une représentation du langage proposée par Google en 2018, permettant de prendre en compte la dimension contextuelle du langage (« Avocat », peut être un fruit ou un juriste..).

FlauBERT<sup>2</sup> est une adaptation de l'algorithme au français.

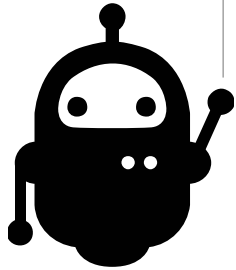
<sup>1</sup> BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.  
Jacob Devlin, Ming-Wei Chang, Kenton Lee,  
Kristina Toutanova  
<https://arxiv.org/abs/1810.04805>

<sup>2</sup> FlauBERT: Unsupervised Language Model Pre-training for French.  
Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux,  
Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé,  
Laurent Besacier, Didier Schwab  
<https://arxiv.org/abs/1912.05372>



# Text embedding with IMDB

Notebook : [\[IMDB1\]](#)



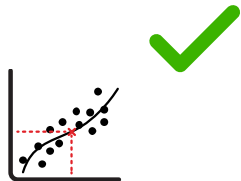
## **Objective :**

Guess whether a film review is positive or not based on the analysis of the text.

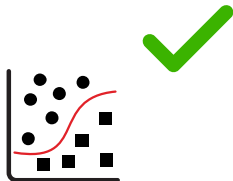
## **Dataset :**

The IMDB dataset is composed of 50,000 film reviews from the site of the same name.





**Basic  
Regression**  
DNN



**Basic  
Classification**  
DNN



**Hight  
Dimensionnall Data**  
(images, ...)  
CNN



**Sparse data**  
(text, ...)  
Embedding



**Sequences data**  
(Time data, ...)  
RNN



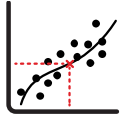
**Reinforcement  
learning**



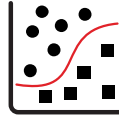
**Variational  
Antoencoder**  
VAE



**Generative  
Adversarial  
Network**  
GAN



**Basic  
Regression**  
DNN



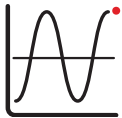
**Basic  
Classification**  
DNN



**Hight  
Dimensionnal Data**  
(images, ...)  
CNN



**Sparse data**  
(text, ...)  
Embedding



**Sequences data**  
(Time data, ...)  
RNN



**Reinforcement**  
learning



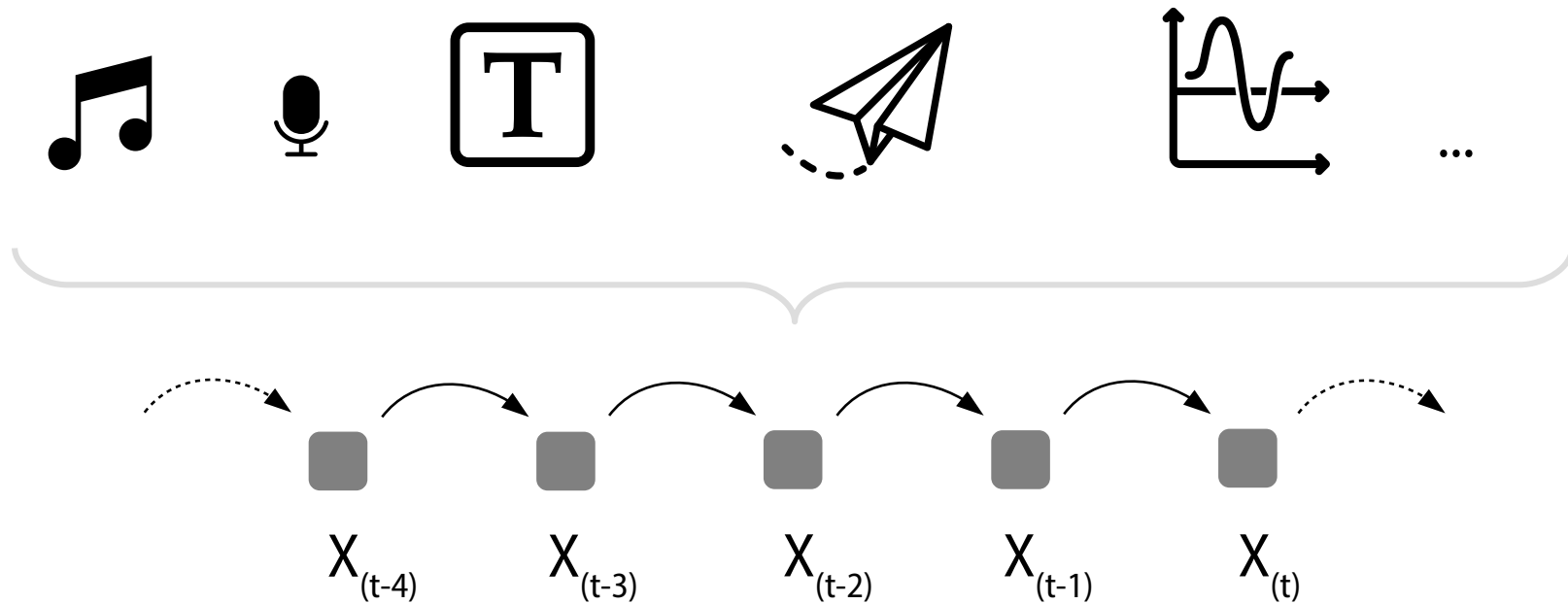
**Variational  
Autoencoder**  
VAE



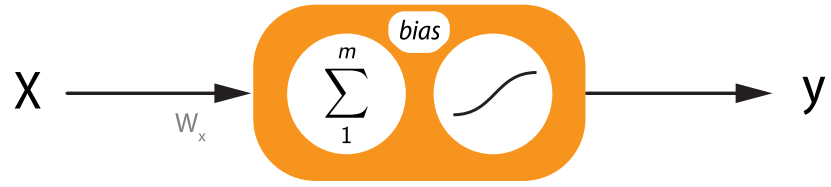
**Generative  
Adversarial  
Network**  
GAN



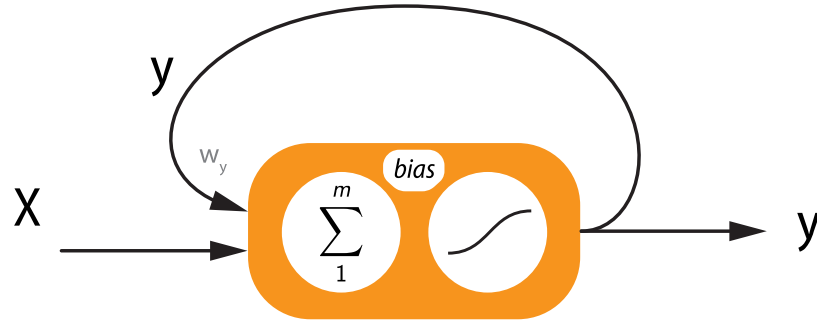
# Recurrent Neural Network (RNN)



# Recurrent Neural Network (RNN)

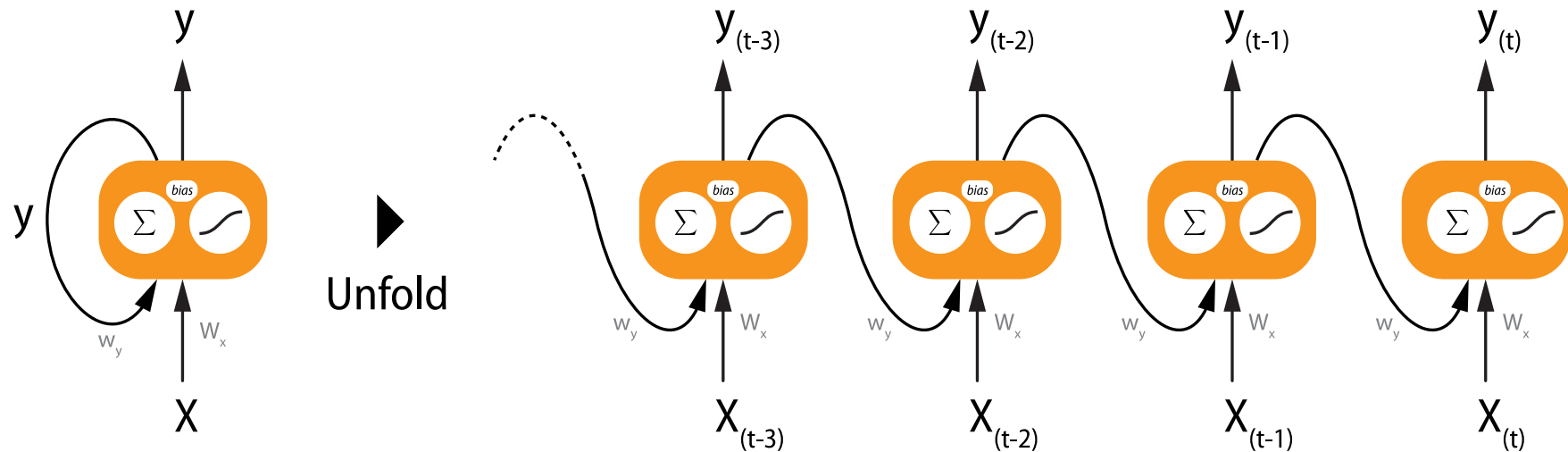


$$y = \sigma(W_x^T \cdot X + b)$$



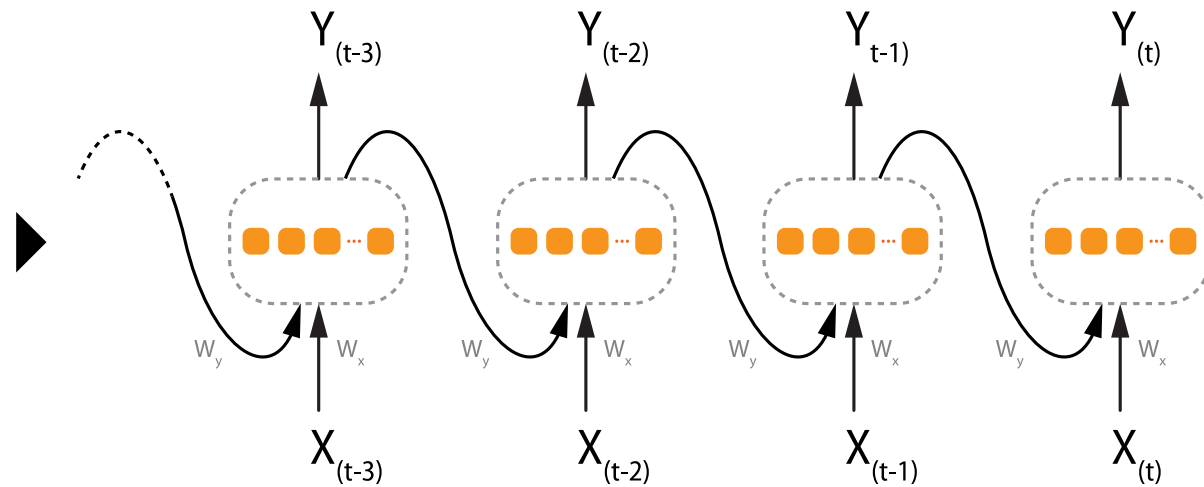
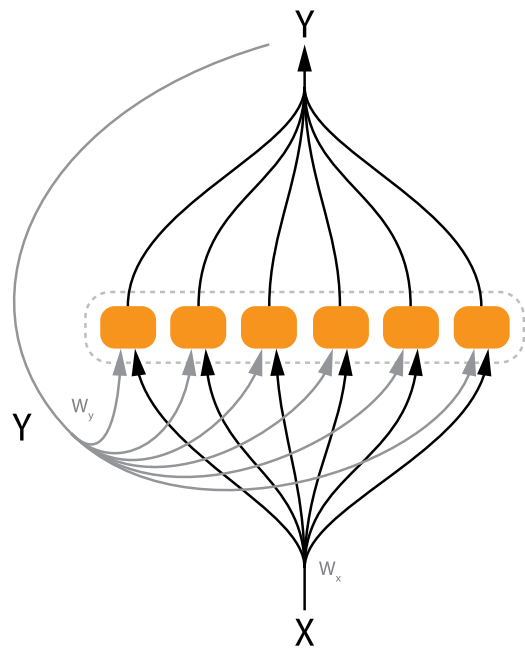
$$y_{(t)} = \sigma(W_x^T \cdot X_{(t)} + w_y \cdot y_{(t-1)} + b)$$

# Reccurent Neural Network (RNN)



$$y_{(t)} = \sigma(W_x^T \cdot X_{(t)} + w_y \cdot y_{(t-1)} + b)$$

# Recurrent Neural Network (RNN)



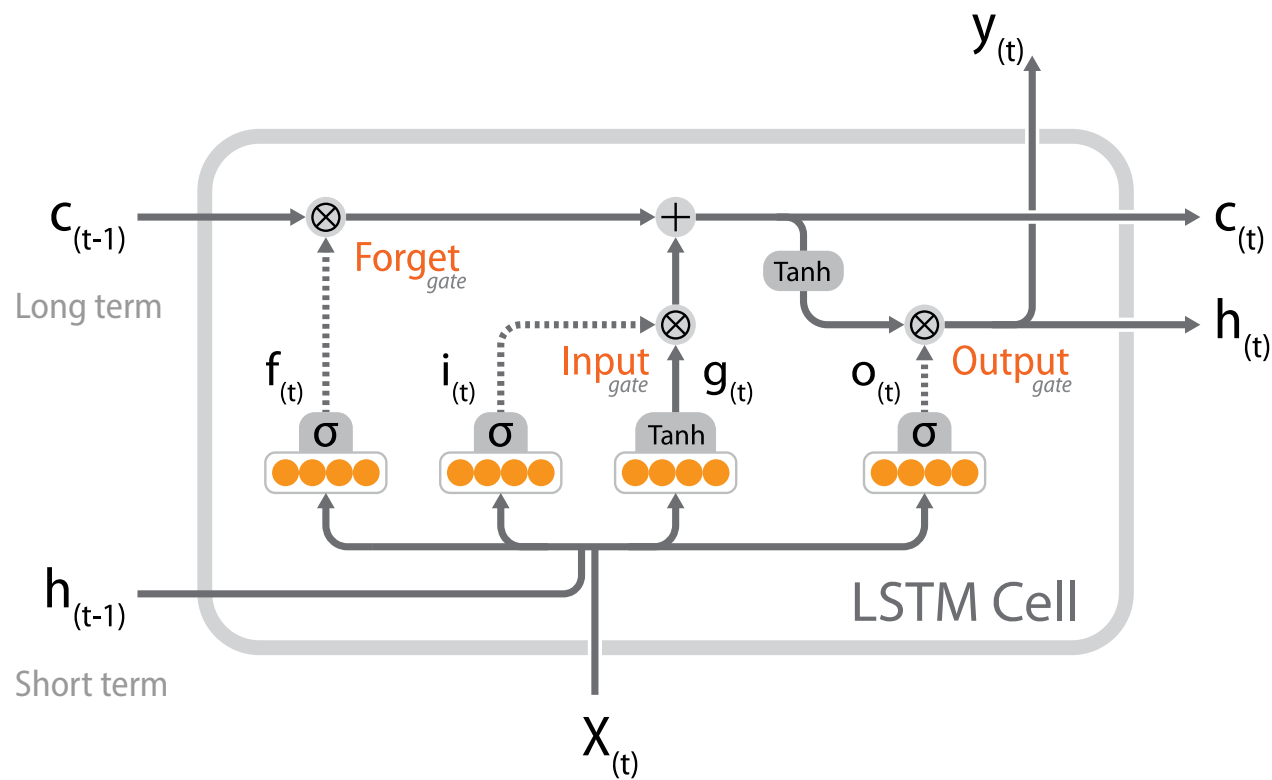
$$Y_{(t)} = \phi (W_x^T \cdot X_{(t)} + W_y^T \cdot Y_{(t-1)} + b)$$

**i** Recurrent neuron  
Recurrent layer } « Cell »



Slow convergence,  
Short memory,  
Vanishing / exploding gradients

# Reccurent Neural Network (RNN)



Long short-term memory (LSTM)<sup>1</sup>

Gated recurrent unit (GRU)<sup>2</sup>

$$\begin{aligned}
 f_{(t)} &= \sigma(W_{xf}^T X_{(t)} + W_{hf}^T h_{(t-1)} + b_f) \\
 i_{(t)} &= \sigma(W_{xi}^T X_{(t)} + W_{hi}^T h_{(t-1)} + b_i) \\
 g_{(t)} &= \tanh(W_{xg}^T X_{(t)} + W_{hg}^T h_{(t-1)} + b_g) \\
 o_{(t)} &= \sigma(W_{xo}^T X_{(t)} + W_{ho}^T h_{(t-1)} + b_o) \\
 c_{(t)} &= f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes g_{(t)} \\
 y_{(t)} &= h_{(t)} = o_{(t)} \otimes \tanh(c_{(t)})
 \end{aligned}$$

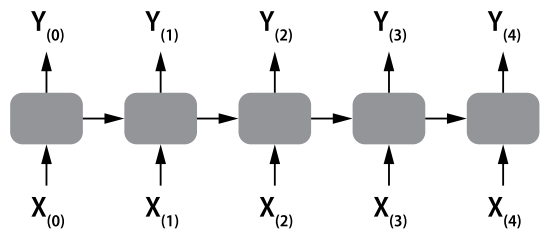
with :

$X_{(t)} \in \mathbb{R}^d$	input vector
$f_{(t)} \in \mathbb{R}^h$	forget gate's activation vector
$i_{(t)} \in \mathbb{R}^h$	input gate's activation vector
$o_{(t)} \in \mathbb{R}^h$	output gate's activation vector
$g_{(t)} \in \mathbb{R}^h$	current entry vector
$h_{(t)}, y_{(t)} \in \mathbb{R}^h$	hidden state or output vector
$c_{(t)} \in \mathbb{R}^h$	cell state vector
$\otimes$	Hadamard product
$\sigma$	sigmoid function
$W_k$	weights matrix
$b_k$	bias vector

<sup>1</sup> Sepp Hochreiter, Jürgen Schmidhuber, (1997) [LSTM]

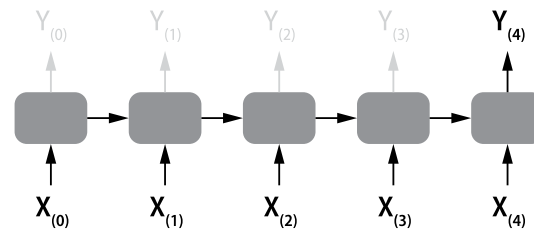
<sup>2</sup> Kyunghyun Cho et al, (2014) [GRU]

# Recurrent Neural Network (RNN)



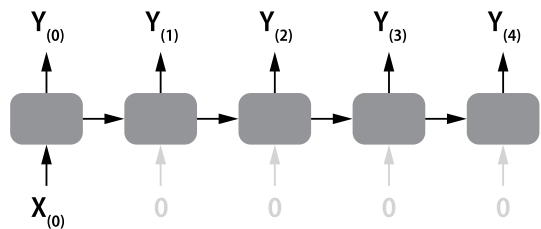
## Series to series

Example : Time series prediction



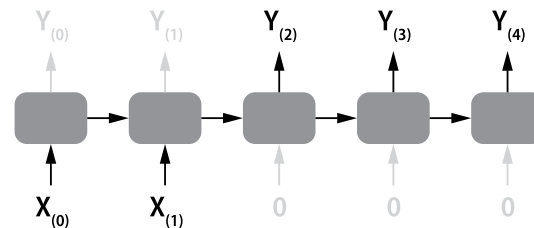
## Series to vector

Example : Sentiment analysis



## Vector to series

Example : Image annotation



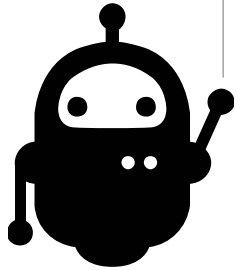
## Encoder-decoder

Example : Language Translation



# Time series with RNN

Notebook : [\[SYNOP1-3\]](#)



## **Objective :**

Guess what the weather will be like !

## **Dataset :**

SYNOP meteorological data.

Data from LYS airport for the period 2010-2020



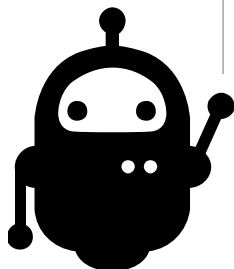
# Time series with RNN

Notebook : [\[SYNOP1-3\]](#)

Episode 1 : Data analysis and creation of a **usable dataset**

Episode 2 : **Training** session and **first predictions**

Episode 3: Attempt to **predict** in the **longer term**

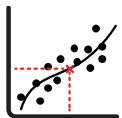




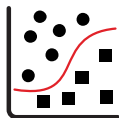


## Little things and concepts to **keep in mind**

- Understand the data, again and again !
- Beware of overfitting
- There are many sparses matrix
- Remember that Pandas is good for you !
- The json files are cool, too
- Preparing your data can cost 70% of the work
- Think about data generators
- Matplotlib (or seaborn) are also very good for you !
- There is a lot of sequential data
- Not everything can uses GPUs...



**Basic  
Regression**  
DNN



**Basic  
Classification**  
DNN



**Hight  
Dimensionnall Data**  
(images, ...)  
CNN



**Sparse data**  
(text, ...)  
Embedding



**Sequences data**  
(Time data, ...)  
RNN



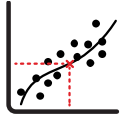
**Reinforcement  
learning**



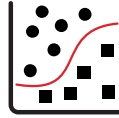
**Variational  
Antoencoder**  
VAE



**Generative  
Adversarial  
Network**  
GAN



**Basic  
Regression**  
DNN



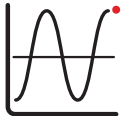
**Basic  
Classification**  
DNN



**Hight  
Dimensionnal Data**  
(images, ...)  
CNN



**Sparse data**  
(text, ...)  
Embedding



**Sequences data**  
(Time data, ...)  
RNN



**Reinforcement  
learning**

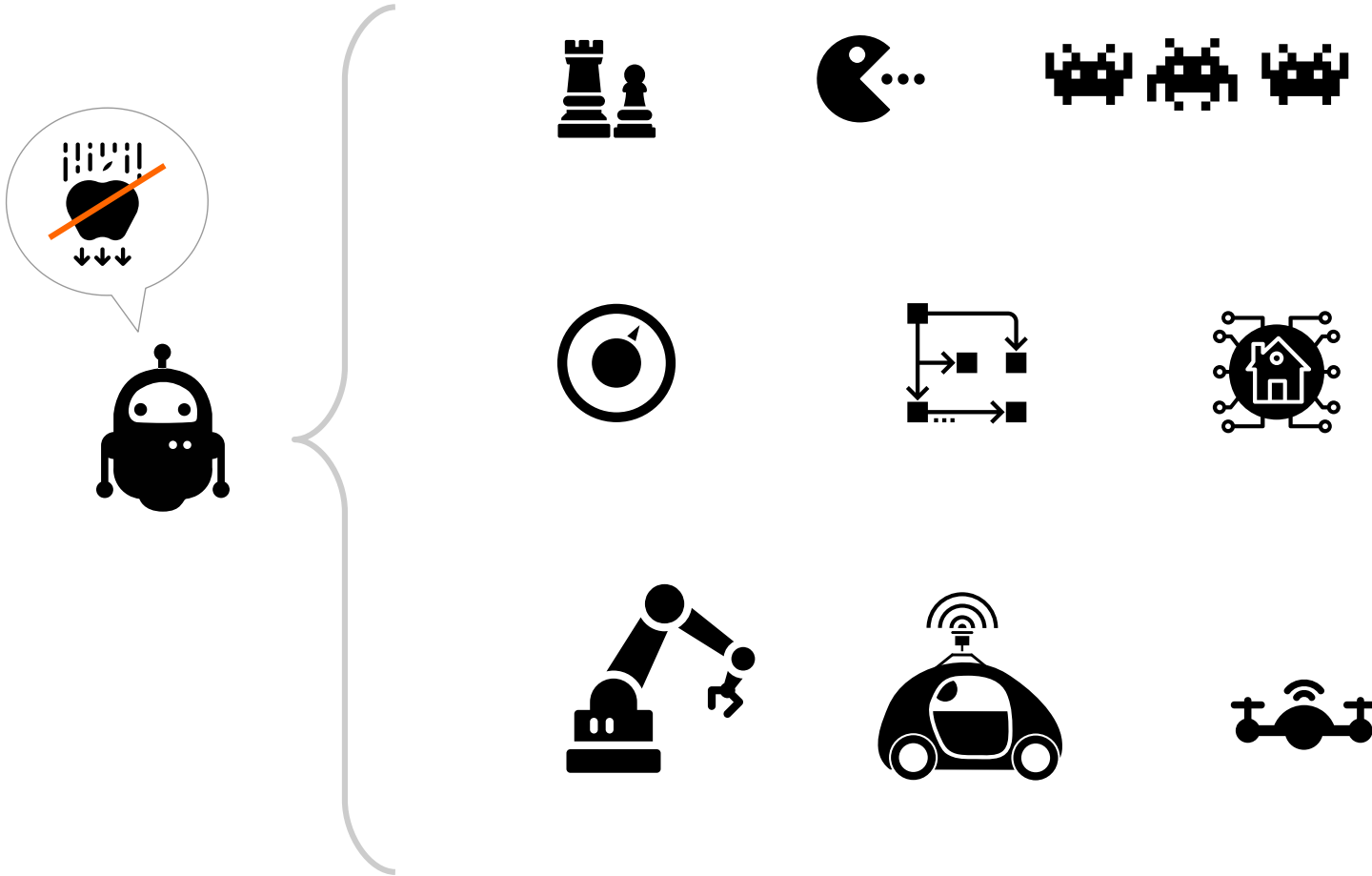


**Variational  
Autoencoder**  
VAE

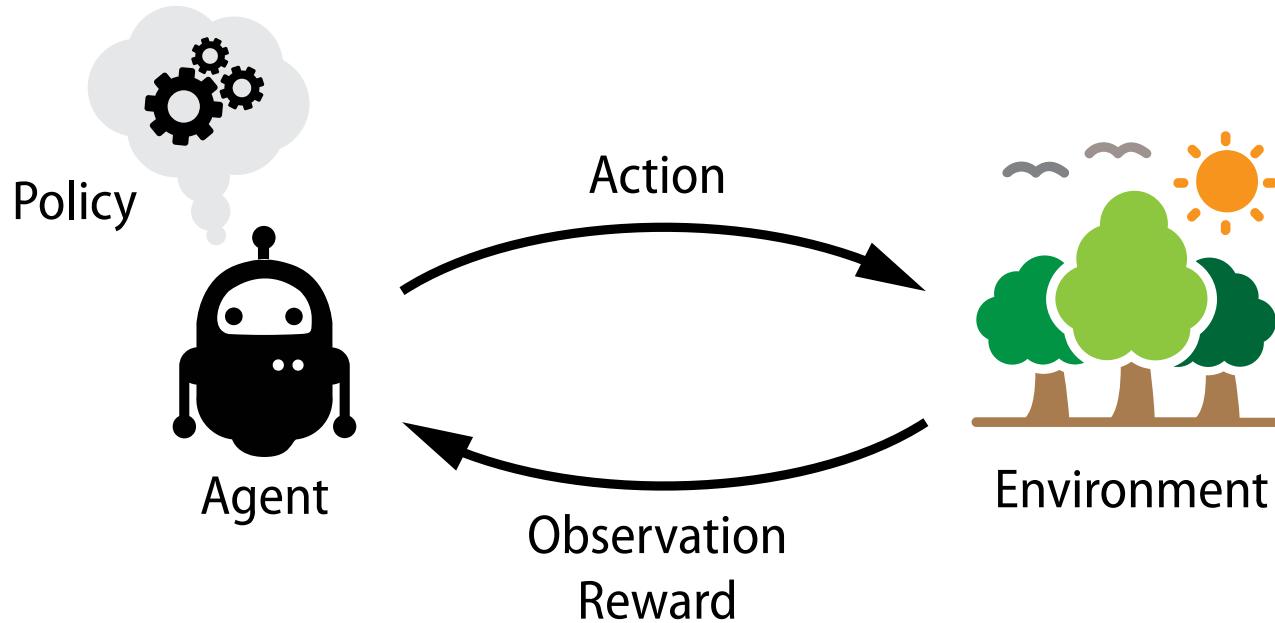


**Generative  
Adversarial  
Network**  
GAN

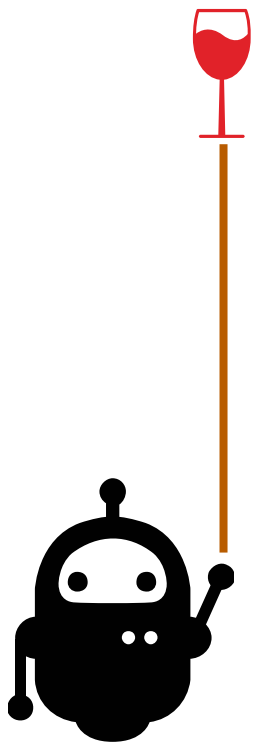
# Reinforcement learning



# Reinforcement learning



What actions can be taken to maximize rewards ?

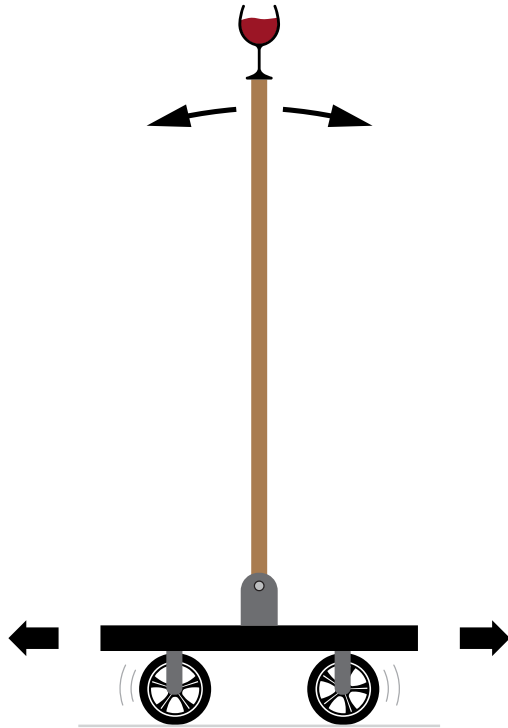


# OpenAI/Gym Cartpole with gradient policy

Implementation in TensorFlow 1.14  
Simulation avec  
Training on 200 epochs  
12'

About [Gym simulator](#)





## Inverted pendulum

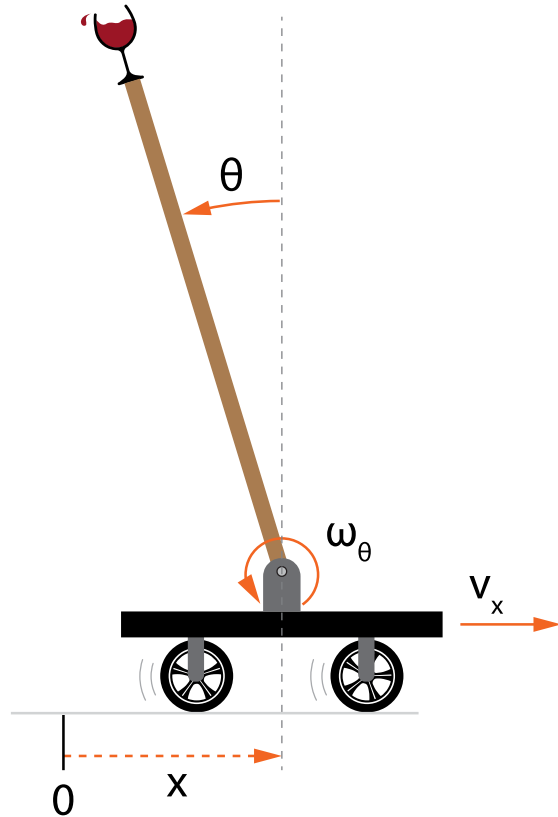
Objective :

Keep the pendulum in balance,  
in the centre of the stage

Actions :

Impulse to  
the **left** (-1)

Impulse to  
the **right** (+1)



## Inverted pendulum

### Observations :

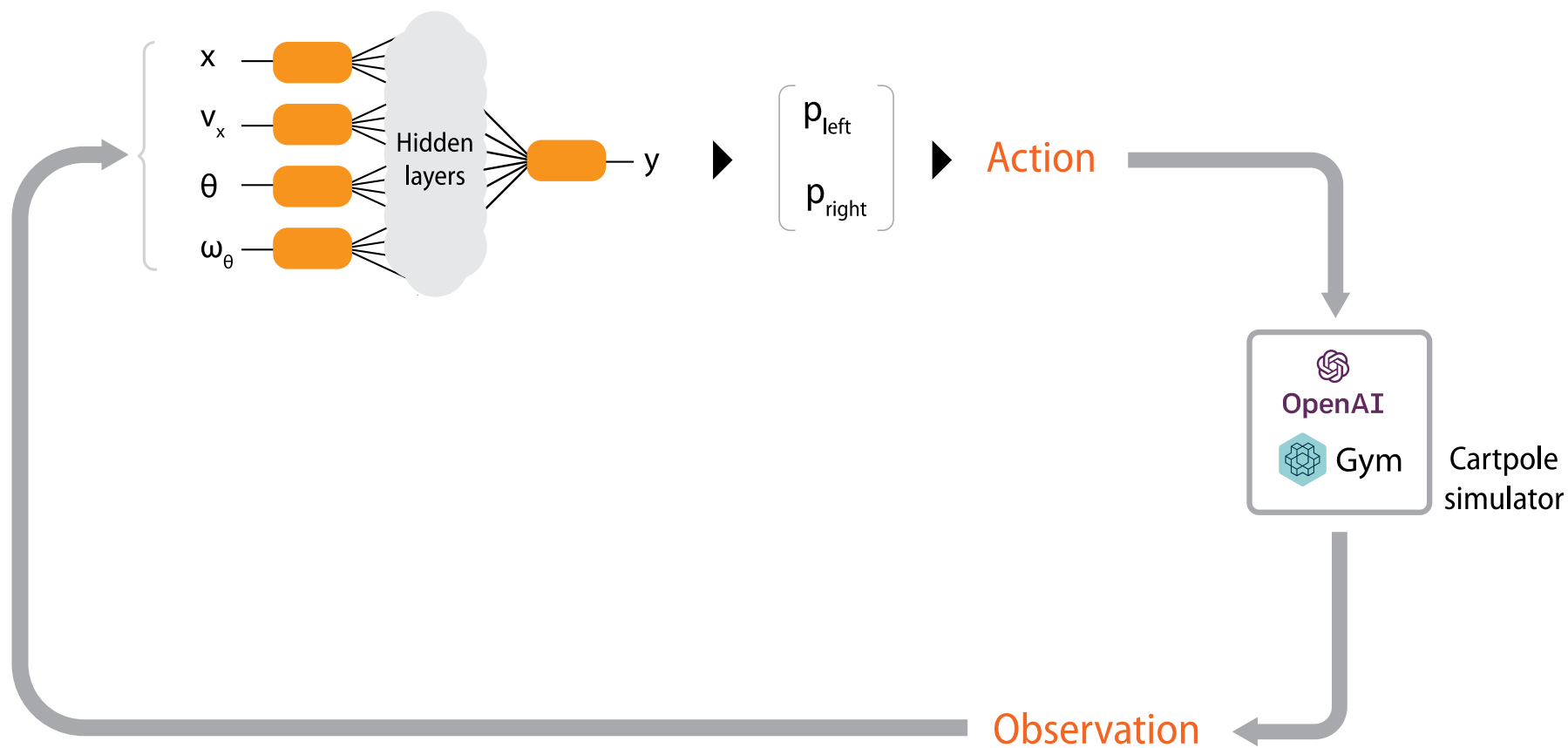
- $x$  Cart position
- $v_x$  Cart velocity
- $\theta$  Pole angle
- $\omega_\theta$  Pole angular velocity

### Rewards :

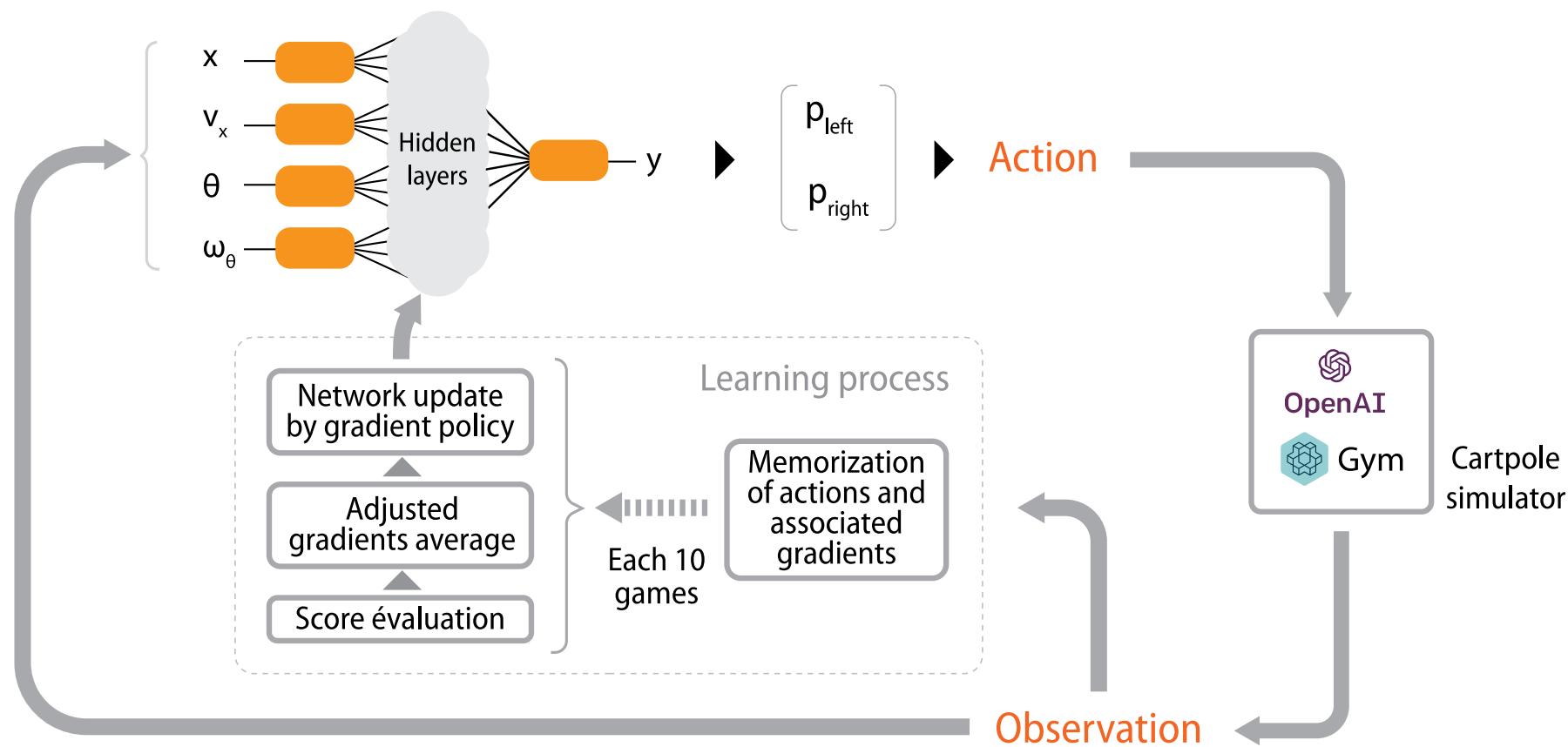
Based on keeping the bar in balance for as long as possible, while remaining in the centre of the stage



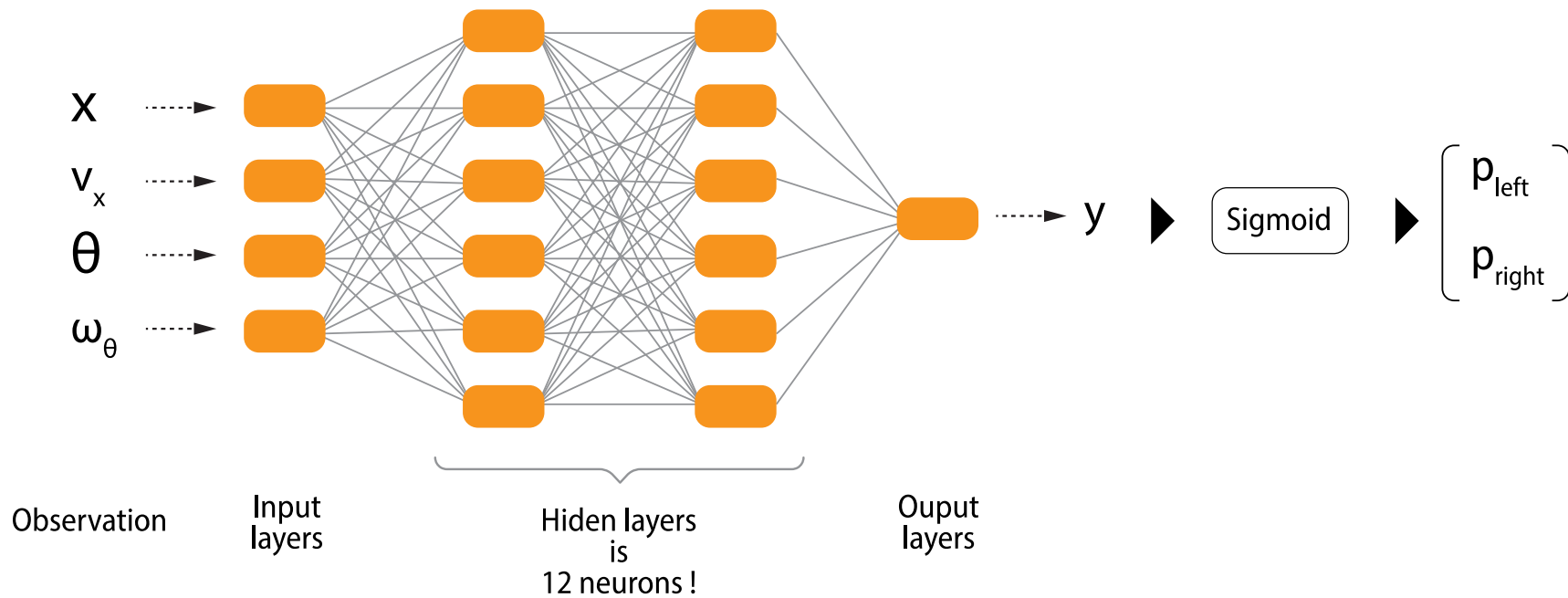
# Reinforcement learning



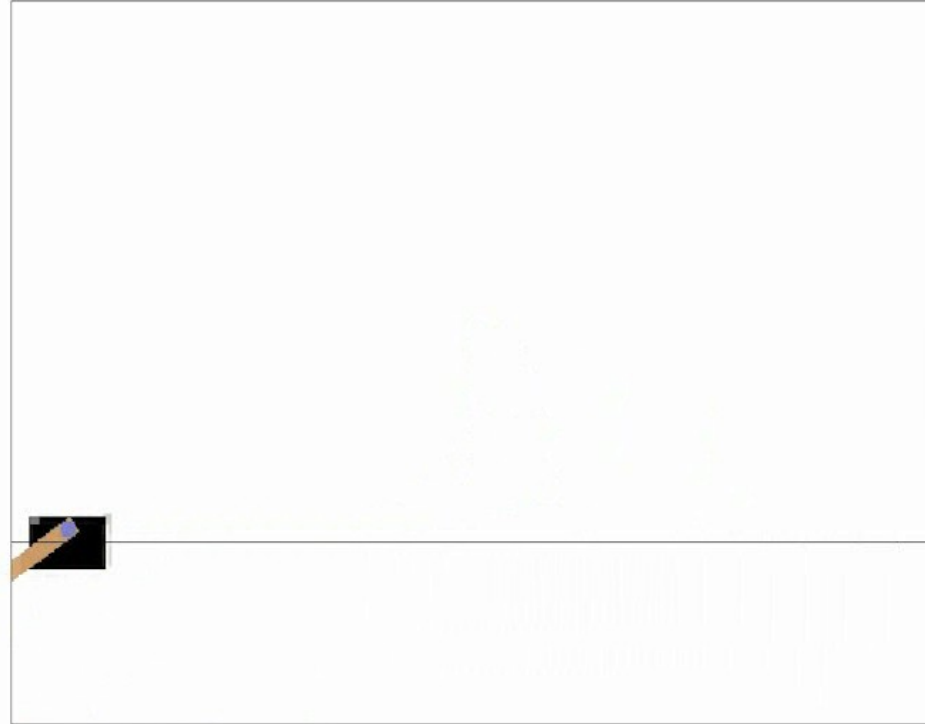
# Reinforcement learning



# Reinforcement learning



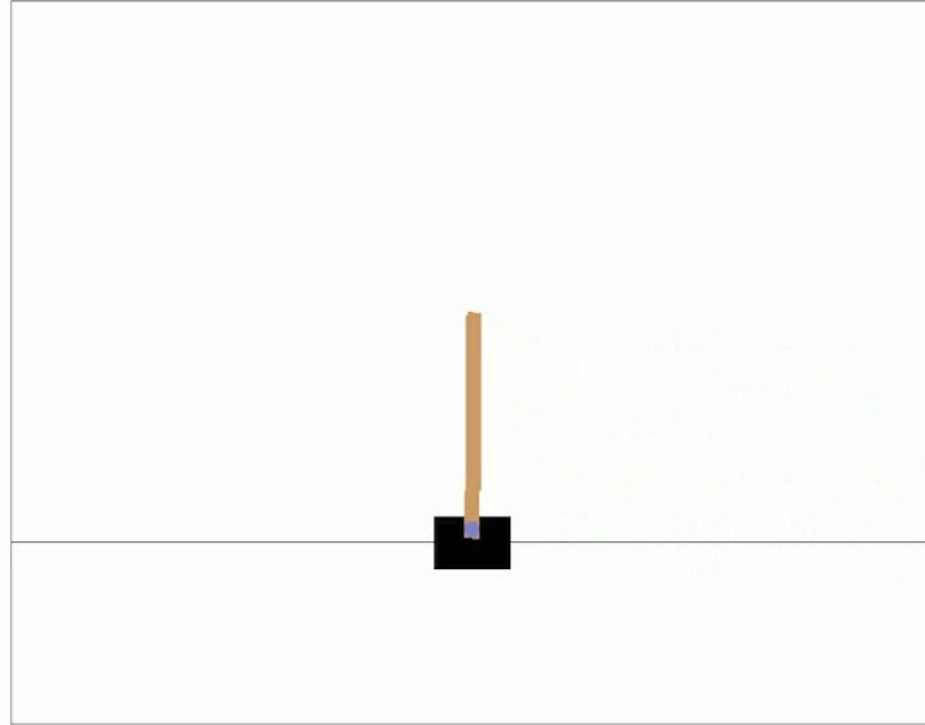
Epoch : 000 - 02:0062



Novice tu es né,  
Novice tu es...



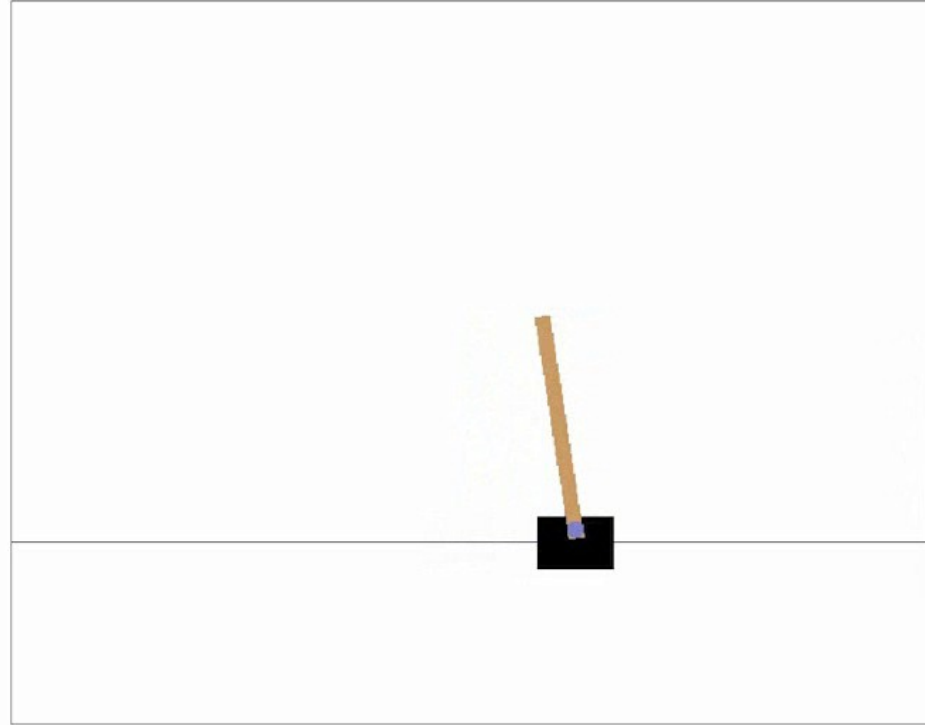
Epoch : 040 - 02:0002



Par le travail  
tu apprendras...



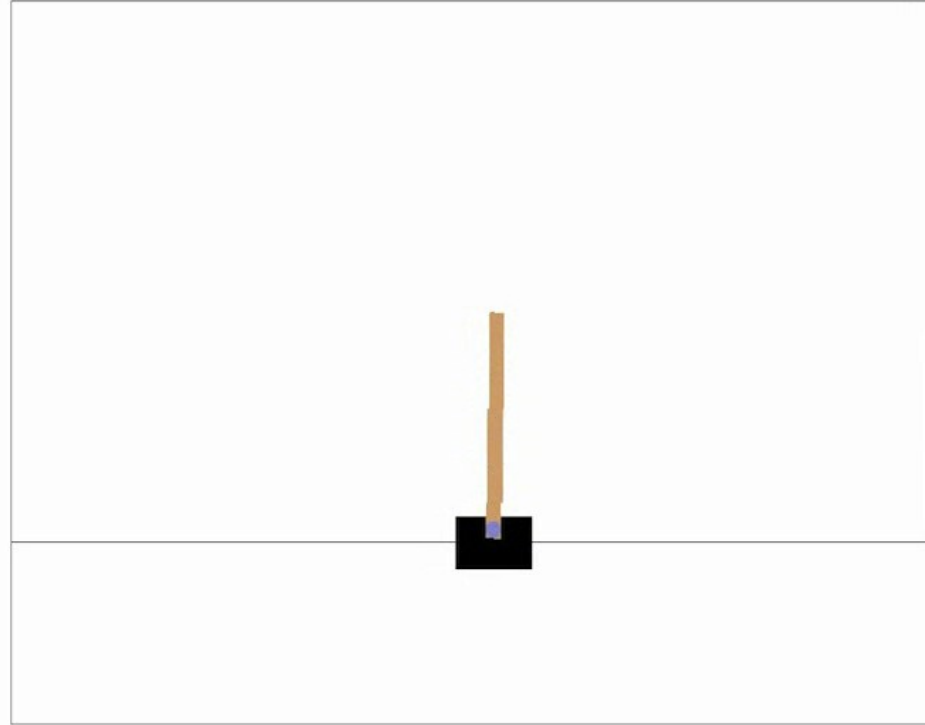
Epoch : 070 - 01:0150



Persévérer  
tu devras...

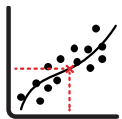


Epoch : 200 - 01:0150

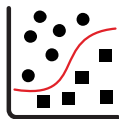


...et grand maître  
tu deviendras





**Basic  
Regression**  
DNN



**Basic  
Classification**  
DNN



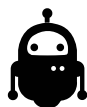
**Hight  
Dimensionnall Data**  
(images, ...)  
CNN



**Sparse data**  
(text, ...)  
Embedding



**Sequences data**  
(Time data, ...)  
RNN



**Reinforcement  
learning**

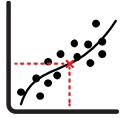


**Variational  
Antoencoder**  
VAE

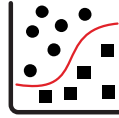


**Generative  
Adversarial  
Network**  
GAN





**Basic  
Regression**  
DNN



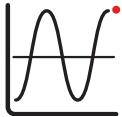
**Basic  
Classification**  
DNN



**Hight  
Dimensionnal Data**  
(images, ...)  
CNN



**Sparse data**  
(text, ...)  
Embedding



**Sequences data**  
(Time data, ...)  
RNN



**Reinforcement**  
learning

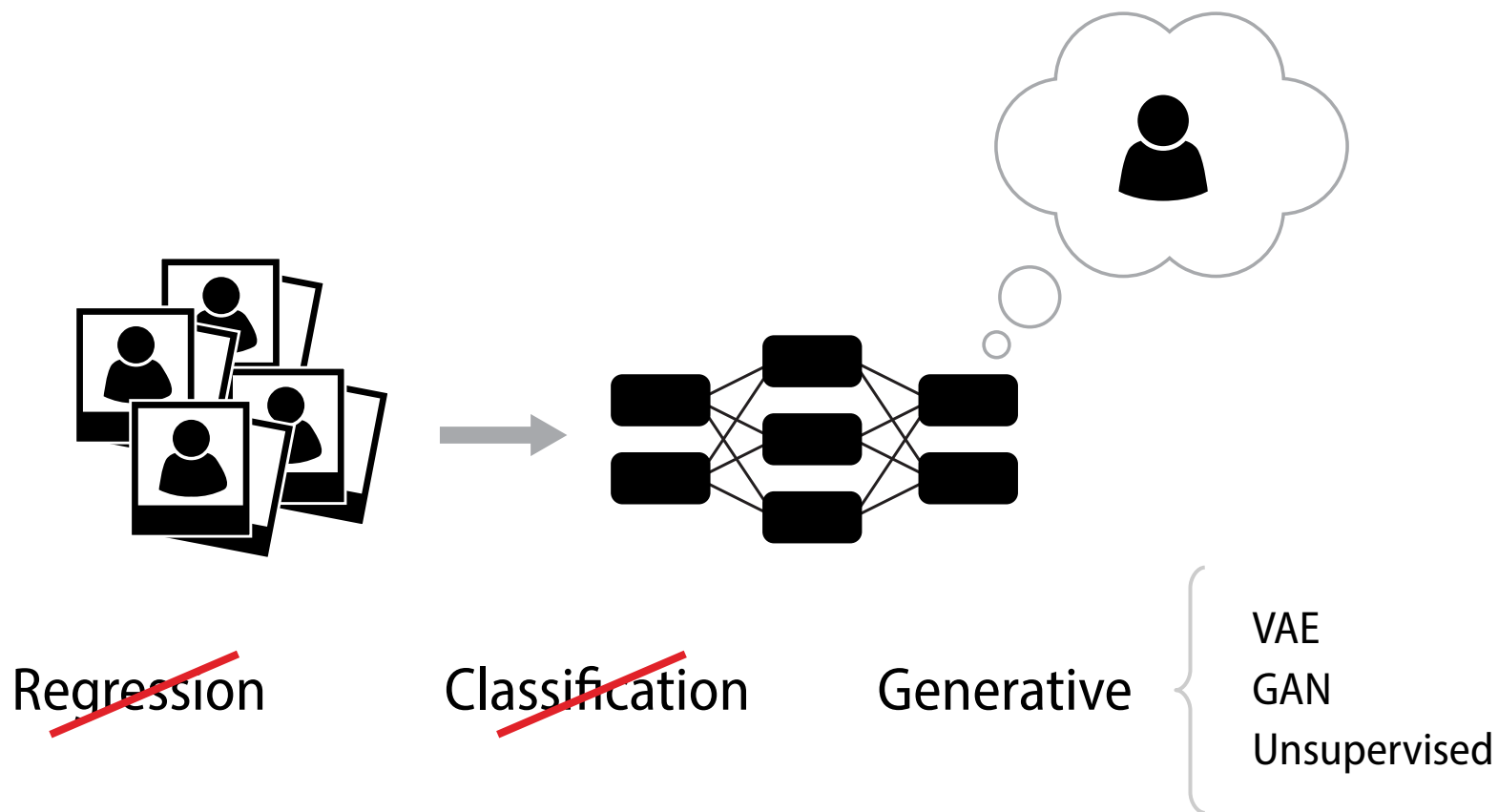


**Variational  
Autoencoder**  
VAE

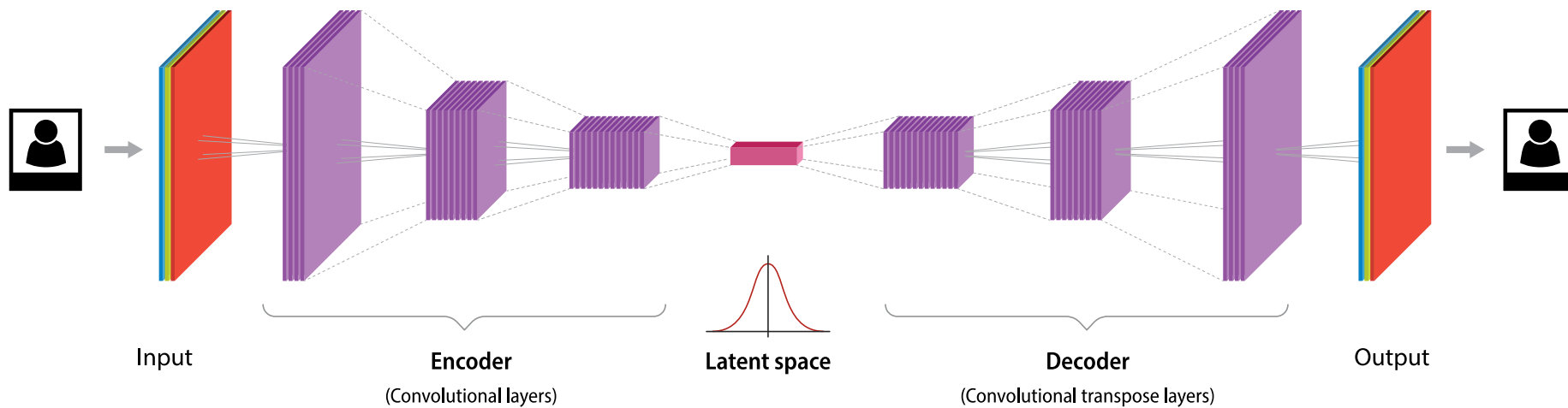


**Generative  
Adversarial  
Network**  
GAN

# Variational Autoencoder



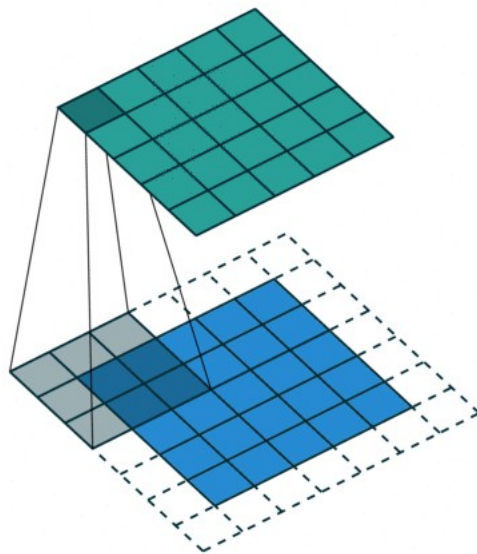
# Variational Autoencoder



Convolution layer



Original image



## Convolutions

`tf.keras.layers.Conv2D`

**Stride**

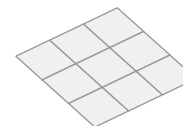
Step size  
(1)

**Padding**

Active or not  
(Active)

**Kernel**

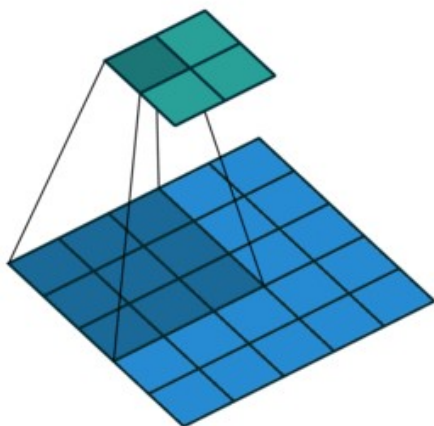
(3x3)



**Convolution layer**  
(2 x 2)



**Original image**  
(4 x 5)



## Convolutions

`tf.keras.layers.Conv2D`

**Stride = ?**

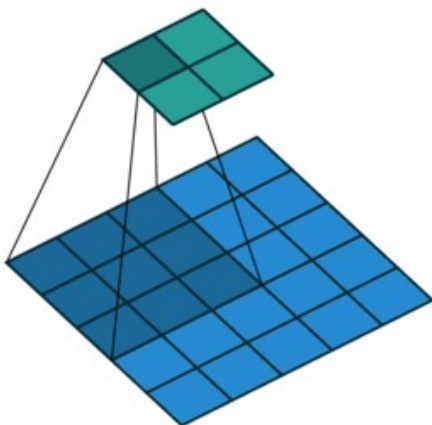
**Padding = ?**

**Kernel = ?**

**Convolution layer**  
(2 x 2)



**Original image**  
(5 x 5)



## Convolutions

`tf.keras.layers.Conv2D`

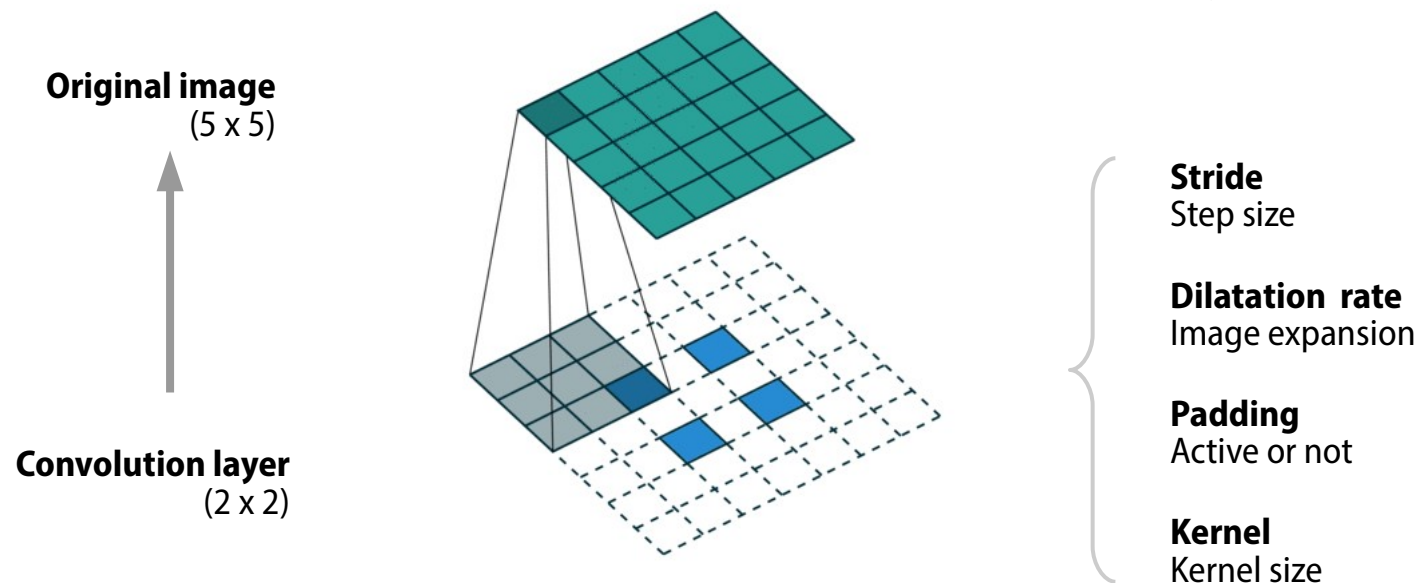
**Stride =** 2

**Padding =** Desactivated

**Kernel =** (3x3)

## Transposed Convolutions

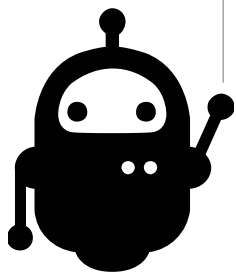
`tf.keras.layers.Conv2DTranspose`





## VAE with MNIST

Notebook : [\[VAE1-2\]](#)



### **Objective :**

First generative network experience with the MNIST dataset

### **Dataset :**

The eternal and essential MNIST dataset



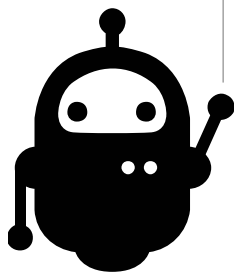


# VAE with MNIST

Notebook : [\[VAE1-2\]](#)

Episode 1 : Model construction and **Training**

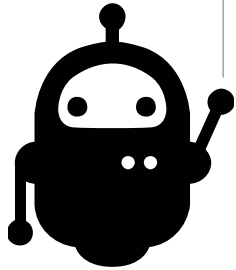
Episode 2 : Exploring our **latent space**





## VAE with CelebA

Notebook : [\[VAE3-8\]](#)



### **Objective :**

New VAE experience, but with a larger and more fun dataset !

### **Dataset :**

"CelebFaces Attributes Dataset (CelebA) is a large-scale face attributes dataset with more than 200K celebrity anotated images.



## VAE with CelebA

Notebook : [\[VAE3-8\]](#)

Episode 3 : About the **CelebA dataset**

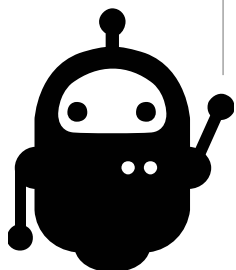
Episode 4 : Preparation of a **clustered dataset**

Episode 5 : **Checking** the clustered dataset

Episode 6 : **Variational AutoEncoder** (VAE) with CelebA (small res.)

Episode 7 : **Variational AutoEncoder** (VAE) with CelebA (medium res.)

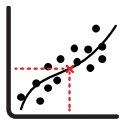
Episode 8 : Exploring our **latent space**



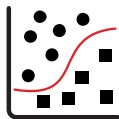


## Little things and concepts to **keep in mind**

- Unsupervised learning,
- Latent space concept
- Problem of large datasets composed of many elements,
- Importance of data centers and GPUs,
- Advanced programming model (classes,...)
- Implementation and use of data generators,
- Notebook and batch, it's possible and it's good !
- One notebook is good, 10 notebook is better !
- Notebooks can do real and serious things !



**Basic  
Regression**  
DNN



**Basic  
Classification**  
DNN



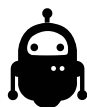
**Hight  
Dimensionnall Data**  
(images, ...)  
CNN



**Sparse data**  
(text, ...)  
Embedding



**Sequences data**  
(Time data, ...)  
RNN



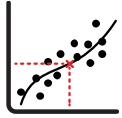
**Reinforcement  
learning**



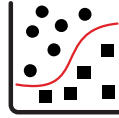
**Variational  
Antoencoder**  
VAE



**Generative  
Adversarial  
Network**  
GAN



**Basic  
Regression**  
DNN



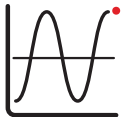
**Basic  
Classification**  
DNN



**Hight  
Dimensionnal Data**  
(images, ...)  
CNN



**Sparse data**  
(text, ...)  
Embedding



**Sequences data**  
(Time data, ...)  
RNN



**Reinforcement**  
learning



**Variational  
Autoencoder**  
VAE

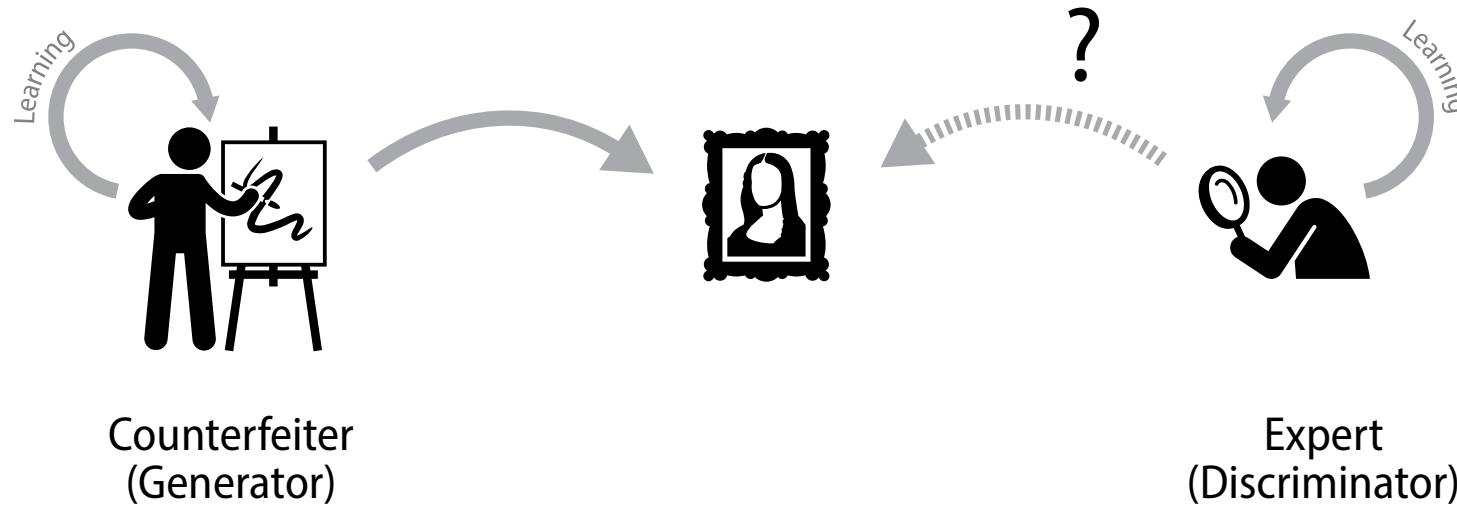


**Generative  
Adversarial  
Network**  
GAN

# Generative Adversarial Network

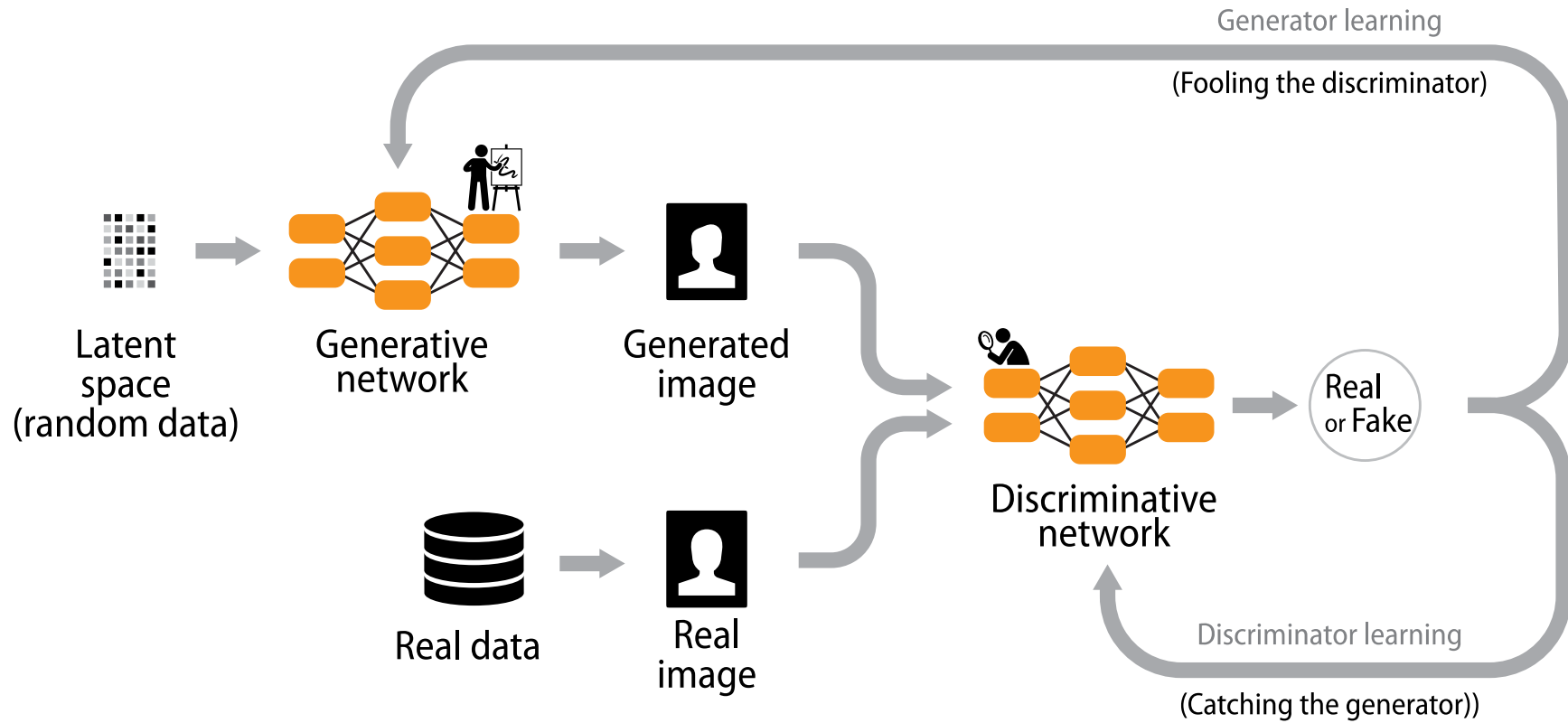
GAN<sup>1</sup> Use Cases :

- Photorealistic images generation
- Image to Image Translation
- Increasing Image Resolution
- Text to Image Generation
- Video / Frame prediction
- Etc.


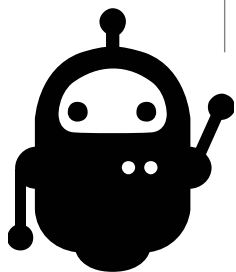


<sup>1</sup> Ian J. Goodfellow & all, (2014), « Generative Adversarial Networks » [GAN]

# Generative Adversarial Network







## Generative Adversarial Network

This X Does Not Exist

Which Face Is Real ?

Colorful Image Colorization

Image-to-Image Translation

Pixel Level Domain Transfer

DeOldify project

**GAN Lab**

**GAN Zoo**

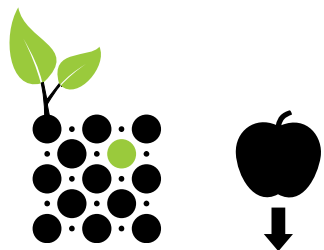


# Generative Adversarial Network

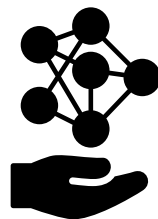
GAN - CGAN - LAPGAN - CatGAN - DCGAN - VAE-GAN - GRAN -  $S^2$ GAN - MGAN - BiGAN - GAN-CLS - ALI - CoGAN - f-GAN - Improved GAN - InfoGAN - SketchGAN - Context-RNN-GAN - EBGAN - IAN - iGAN - SeqGAN - SRGAN - VGAN - 3D-GAN - AC-GAN - AffGAN - GAWWN - b-GAN - C-RNN-GAN - CC-GAN - DTN - GMAN - IcGAN - LSGAN - MV-BiGAN - pix2pix - RenderGAN - SAD-GAN - SGAN - SSL-GAN - TGAN - Unrolled GAN - VGAN - AL-CGAN - MARTA-GAN - MDGAN - MPM-GAN - PPGN - PrGAN - SGAN - SimGAN - StackGAN - textGAN - AdaGAN - ID-CGAN - LAGAN - LS-GAN - SalGAN - Unim2im - ViGAN - WGAN - acGAN - ArtGAN - Bayesian GAN - BS-GAN - MalGAN - MaliGAN - McGAN - ST-GAN - WaterGAN - AEGAN - AM-GAN - AnoGAN - BEGAN - CS-GAN - CVAE-GAN - CycleGAN - DiscoGAN - GP-GAN - LR-GAN - MedGAN - MIX+GAN - RTT-GAN - SEGAN - SeGAN - SGAN - TAC-GAN - Triple-GAN - UNIT - DualGAN - FF-GAN - GoGAN - MAD-GAN - MAGAN - SL-GAN - Softmax GAN - TAN - TP-GAN - VariGAN - VAW-GAN - WGAN-GP -  $\hat{I}^2$ -GAN - Bayesian GAN - CaloGAN - Conditional cycleGAN - Cramér GAN - DR-GAN - DRAGAN - ED//GAN - EGAN - Fisher GAN - Flow-GAN - GeneGAN - Geometric GAN - IRGAN - MMD-GAN - ORGAN - Pose-GAN - PSGAN - RankGAN - RPGAN - RWGAN - SBADA-GAN - SD-GAN - VEEGAN - WS-GAN - ARAE - BCGAN - CAN - Chekhov GAN - crVAE-GAN - DeliGAN - DistanceGAN - DSP-GAN - Dualing GAN - Fila-GAN - GANCS - GMM-GAN - IWGAN - PAN - Perceptual GAN - PixelGAN - RCGAN - RNN-WGAN - SegAN - TextureGAN -  $\hat{I} \pm$ -GAN - 3D-IWGAN - AE-GAN - AlignGAN - APE-GAN - ARDA - DAN - I-GAN - LD-GAN - LeGAN - MMGAN - MoCoGAN - ResGAN - SisGAN - ss-InfoGAN - SSGAN - SteinGAN - VRAL - 3D-RecGAN - ABC-GAN - ASDL-GAN - BGAN - CDcGAN - CGAN - constrast-GAN - Coulomb GAN - DM-GAN - GAN-sep - GAN-VFS - MGGAN - PGAN - SN-GAN - SS-GAN - ViGAN - ARIGAN - CausalGAN - D2GAN - ExposureGAN - ExprGAN - GAMN - GraspGAN - LDAN - LeakGAN - MD-GAN - MuseGAN - OptionGAN - PassGAN - RefineGAN - Splitting GAN -  $\hat{I}''$ -GAN - CM-GAN - GAN-ATV - GAP - GP-GAN - Progressive GAN -  $PS\hat{A}^2$ -GAN - SVSGAN - TGAN - 3D-ED-GAN - ABC-GAN - ActuaL - AttGAN - AttnGAN - BCGAN - BicycleGAN - CatGAN - CoAtt-GAN - ConceptGAN - Cover-GAN - D-GAN - DAGAN - DeblurGAN - DNA-GAN - DRPAN - FIGAN - FSEGAN - FTGAN - GANDI - GPU - HAN - HP-GAN - HR-DCGAN - IfcVAEGAN - In2I - Iterative-GAN - IVE-GAN - iVGAN - KBGAN - KGAN - LGAN - MLGAN - ORGAN - Pip-GAN - pix2pixHD - Sobolev GAN - StarGAN - TGAN - tripletGAN - VA-GAN - XGAN - ZipNet-GAN - ACGAN - CA-GAN - ComboGAN - DF-GAN - Dynamics-Transfer GAN - EnergyWGAN - ExGAN - f-CLSWGAN - FusionGAN - G2-GAN - GAGAN - GAN-RS - GANG - GANosaic - IdCycleGAN - manifold-WGAN - MC-GAN - MIL-GAN - MS-GAN - PacGAN - PN-GAN - PPGAN - RAN - SGAN - SRPGAN - ST-CGAN - Super-FAN - TV-GAN - UGACH - UV-GAN - VGAN - weGAN - AdvGAN - CFG-GAN - CipherGAN - Cross-GAN - dp-GAN - ecGAN - FusedGAN - GeoGAN - GLCA-GAN - LAC-GAN - MaskGAN - SG-GAN - SketchyGAN - tempoGAN - UGAN - AmbientGAN - ATA-GAN - C-GAN - CapsuleGAN - DA-GAN - DP-GAN - DPGAN - First Order GAN - GC-GAN - LB-GAN - MAGAN - ND-GAN - PGD-GAN - RadialGAN - SAR-GAN - SCH-GAN - StainGAN - SWGAN - VoiceGAN - WaveGAN - Attention-GAN - B-DCGAN - BAGAN - BranchGAN - D2IA-GAN - DBLRGAN - E-GAN - ELEGANT - Fictitious GAN - GAAN - GONet - memoryGAN - MTGAN - NCE-GAN - NetGAN - OGAN - OT-GAN - PGGAN - Sdf-GAN - Social GAN - Spike-GAN - ST-GAN - Text2Shape - tiny-GAN - VOS-GAN - 3D-PhysNet - AF-DCGAN - BEAM - CorrGAN - D-WCGAN - Defo-Net - DSH-GAN - DTR-GAN - DVGAN - EAR - FBGAN - FusionGAN - Graphical-GAN - IterGAN - M-AAE - MelanoGAN - MGGAN - ModularGAN - NAN - PM-GAN - ProGANSR - PS-GAN - ReConNN - SAGA - sGAN - Sketcher-Refiner GAN - SyncGAN - TGANs-C - UT-SCA-GAN - AdvEntuRe - AVID - BourGAN - BRE - cd-GAN - cowboy - CSG - Defense-GAN - DialogWAE - DTLC-GAN - FairGAN - Fairness GAN - FakeGAN - FBGAN - FC-GAN - GAF - GAN - Q-learning - GAN-SD - GAN-Word2Vec - GANAX - GT-GAN - HAN - HiGAN - hredGAN - MC-GAN - MEGAN - MolGAN - N2RPP - PD-WGAN - POGAN - PSGAN - ReGAN - RegCGAN - RoCGAN - SAGAN - SG-GAN - speech-driven animation GAN - WGAN-CLS - Adaptive GAN - APD - BinGAN - BWGAN - CapsGAN - CR-GAN - DMGAN - EL-GAN - FrankenGAN - GAIN - GANG - GATS - IR2VI - IRGAN - JointGAN - JR-GAN - LCC-GAN - MedGAN - MMC-GAN - Modified GAN-CLS - PP-GAN - SeUDA - SN-DCGAN - SN-PatchGAN - SoPhie - SR-CNN-VAE-GAN - StarGAN-VC - table-GAN - tcGAN - TD-GAN - tempCycleGAN - VAC+GAN - acGAN - AlphaGAN - AMC-GAN - CE-GAN - ciGAN - CT-GAN - DE-GAN - Dropout-GAN - Editable GAN - FGGAN - GAIA - GAP - IntroVAE - ISGAN - LBT - Lipizzaner - MIXGAN - PIONEER - RaGAN - Resembled GAN - sAOG - Sem-GAN - SGAN - SiGAN - TequilaGAN - WGAN-L1 - BEGAN-CS - Bellman GAN - BridgeGAN - DOPING - GIN - GM-GAN - ISP-GPM - MinLGAN - Recycle-GAN - ScarGAN - Skip-Thought GAN - StepGAN - T2Net - TreeGAN - X-GANs - AE-OT - AIM - Bi-GAN - BubGAN - CinCGAN - ClusterGAN - DADA - DeepFD - ESRGAN - GAN Lab - GAN-AD - GANVO - GcGAN - GraphSGAN - IGMM-GAN - MeRGAN - SAM - SiftingGAN - SLSR - Twin-GAN - WaveletGLCA-GAN ...

# Conclusion

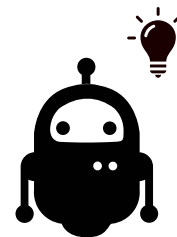
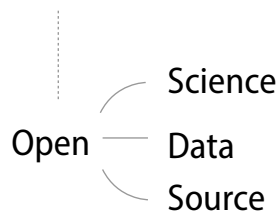




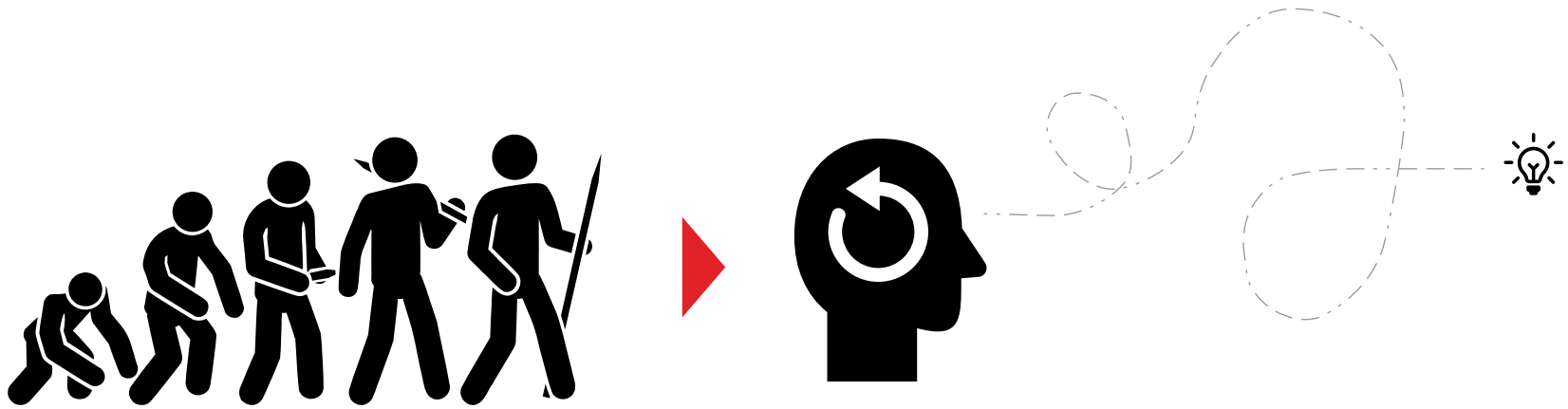
Many fields of  
application !  
...and **it works !**



Complex  
but more and more  
**accessible.**



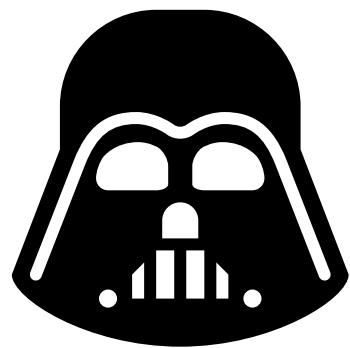
Very significant  
and **rapid progress...**  
...sometimes difficult to  
follow ;-|



**Change** in the apprehension of problems, tools and techniques

Generational fracture  
Infrastructure adaptation  
Competences development

...





Career / Jobs,  
Organization,  
Decision-making,  
...



Privacy,  
Surveillance,  
Censorship,  
...



Algorithmes, la bombe à retardement  
Editions Les Arènes  
Cathy O'Neil



Chair on the Legal and Regulatory Implications  
of Artificial Intelligence at MIAI Grenoble Alpes.  
<https://ai-regulation.com>

## RECONNAISSANCE FACIALE : POUR UN DÉBAT À LA HAUTEUR DES ENJEUX\*

15 novembre 2019

## COMMENT PERMETTRE À L'HOMME DE GARDER LA MAIN\* ?

Les enjeux éthiques des algorithmes et de l'intelligence  
artificielle

SYNTHÈSE DU DÉBAT PUBLIC ANIMÉ PAR LA CNIL DANS LE CADRE DE LA MISSION  
DE RÉFLEXION ÉTHIQUE CONFÉE PAR LA LOI POUR UNE RÉPUBLIQUE NUMÉRIQUE

## « SAN FRANCISCO BANS FACIAL RECOGNITION TECHNOLOGY »

New York Times  
May 14, 2019

\* See [CNIL1], [CNIL2]



- [JGRAY] Gray, J. (2001), from « The Fourth Paradigm: Data-Intensive Scientific Discovery » Tony Hey, Stewart Tansley, Kristin Tolle (2009). Published by Microsoft Research. ISBN: 978-0-9825442-0-4
- [MCPIT] McCulloch, Warren; Walter Pitts (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity". *Bulletin of Mathematical Biophysics*. 5 (4): 115–133. doi:10.1007/BF02478259
- [DHEBB] Hebb, D. O. (1949). « The Organization of Behavior: A Neuropsychological Theory. » New York: Wiley and Sons. ISBN 9780471367277.
- [FROS] Rosenblatt, Frank. (1958). « The perceptron: A probabilistic model for information storage and organization in the brain. » *Psychological Review*, 65(6), 386–408.
- [MIPA] Minsky, Marvin; Papert, Seymour. (1969). « Perceptrons : An Introduction to Computational Geometry », MIT Press
- [DRUM] Rumelhart, David E.; Hinton, Geoffrey E.; Williams, Ronald J. (1986). « Learning representations by back-propagating errors ». *Nature*. 323 (6088): 533–536. doi:10.1038/323533a0.
- [YLEC1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, « Backpropagation Applied to Handwritten Zip Code Recognition », AT&T Bell Laboratories
- [LRDN] Dominique Cardon, Jean-Philippe Cointet, Antoine Mazieres. (2018). « La revanche des neurones », *Réseaux, La Découverte*, 5 (211), <10.3917/res.211.0173>. <hal-01925644>
- [AMAZ] Antoine Mazieres (2016) Thèse : « Cartographie de l'apprentissage artificiel et de ses algorithmes » Université Paris 7 Denis Diderot, <hal-01771655>
- [TOP500] Statistics on top 500 high-performance computers. (2018) « Exponential growth of supercomputing power as recorded by the TOP500 list ». <https://www.top500.org>
- [WKP1] Wikipedia/en. (2018) « List of datasets for machine-learning research ». <https://en.wikipedia.org>
- [WOS1] Core database : TS=("support vector machine\*" OR ("SVM" AND "classification") OR ("SVM" AND "regression") OR ("SVM" AND "classifier") OR "support vector network\*" OR ("SVM" AND "kernel trick\*))
- [WOS2] Core database : TS=("deep learning" OR "deep neural network\*" OR ("DNN" AND "neural network\*") OR "convolutional neural network\*" OR ("CNN" AND "neural network\*") OR "recurrent neural network\*" OR ("LSTM" AND "neural network\*") OR ("RNN\*" AND "neural network\*"))
- [ALEX] A. Krizhevsky, I. Sutskever, G. Hinton. (2012). « ImageNet Classification with Deep Convolutional Neural Networks » doi: 10.1145/3065386
- [ILSVRC] ImageNet Large Scale Visual Recognition Challenges <http://image-net.org/challenges/LSVRC/<2012..2017>/results> <https://en.wikipedia.org/wiki/ImageNet>
- [MOBIN] Howard, Andrew G. et al. (2017) "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." <https://arxiv.org/abs/1704.04861>

- [W2VEC] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean (2013), « Distributed Representations of Words and Phrases and their Compositionality », <https://arxiv.org/abs/1310.4546>
- [GLOVE] Jeffrey Pennington, Richard Socher, Christopher D. Manning (2014) « GloVe: Global Vectors for Word Representation », <http://nlp.stanford.edu/projects/glove/>
- [P2VEC] Ehsaneddin Asgari, Mohammad R.K. Mofrad, (2016), « ProtVec: A Continuous Distributed Representation of Biological Sequences », <https://arxiv.org/abs/1503.05140>
- [LSTM] Sepp Hochreiter, Jürgen Schmidhuber, (1997), « Long Short-Term Memory, » <https://doi.org/10.1162/neco.1997.9.8.1735>
- [GRU] Cho, Kyunghyun; van Merriënboer, Bart; Gulcehre, Caglar; Bahdanau, Dzmitry; Bougares, Fethi; Schwenk, Holger; Bengio, Yoshua (2014), « Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation ». <https://arxiv.org/abs/1406.1078>
- [CARTP] AG Barto, RS Sutton and CW Anderson, (1983), « Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problem », IEEE Transactions on Systems, Man, and Cybernetics, 1983
- [GAN] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, (2014), « Generative Adversarial Networks » <https://arxiv.org/abs/1406.2661>
- [WOS3] Core database : TS=('material' and ('design' or 'discovery' or 'optimization') and ('deep learning' or 'machine learning' or 'neurons'))
- [AIDEX] AI Index. « A starting point for informed conversations about progress in artificial intelligence. The report aggregates a diverse set of metrics, and makes the underlying data easily accessible to the general public ». <https://aiindex.org>
- [DLPW] Jeff Hale, « Deep Learning Framework Power Scores 2018 » <http://bit.ly/33Wp14y> and <http://bit.ly/2NagcgH>
- [CNIL1] Comment permettre à l'homme de garder la main ? Synthèse du débat public animé par la cnil dans le cadre de la mission de réflexion éthique confiée par la loi pour une république numérique. <https://www.cnil.fr/fr/comment-permettre-lhomme-de-garder-la-main-rapport-sur-les-enjeux-ethiques-des-algorithmes-et-de>
- [CNIL2] Reconnaissance faciale : pour un débat à la hauteur des enjeux 15 novembre 2019 <https://www.cnil.fr/fr/reconnaissance-faciale-pour-un-debat-la-hauteur-des-enjeux>

# Illustrations

[POTATO]	From <i>Die Giftpflanzen Deutschlands</i> , Peter Esser, 1910, via <a href="#">iconspng.com</a>
[CONVO]	An Introduction to different Types of Convolutions in Deep Learning <a href="https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d">https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d</a>
[NEURON]	Wikimedia Commons, the free media repository.
Photos	<a href="#">pixels.com</a>
Icons	<a href="#">thenounproject.com</a>



<https://gricad-gitlab.univ-grenoble-alpes.fr/talks/fidle>



Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0)

<https://creativecommons.org/licenses/by-nc-nd/4.0/>

