AI Machine Learning & deep learning

SARI 6 Juin 2019

Powered by

python™ TensorFlow matplotlib NumPy scikit learn machine learning in Python Gym ANACONDA jupyter

Jean-Luc Parouty
Laboratoire SIMaP
http://simap.grenoble-inp.fr/

cnrs SIMaP

[intelligence]

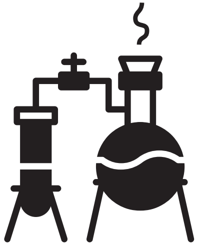
« Ability to perceive or infer information, and to retain it as knowledge to be appli towards adaptive behaviors within an environment or context »*

« 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 »

* Wikipedia

[Méthode scientifique]

1^{st} paradigm



Experimental science

2^{nd} paradigm

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

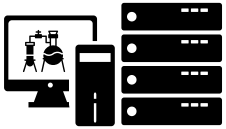
$$\nabla \times H = J + \frac{\partial D}{\partial t}$$

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

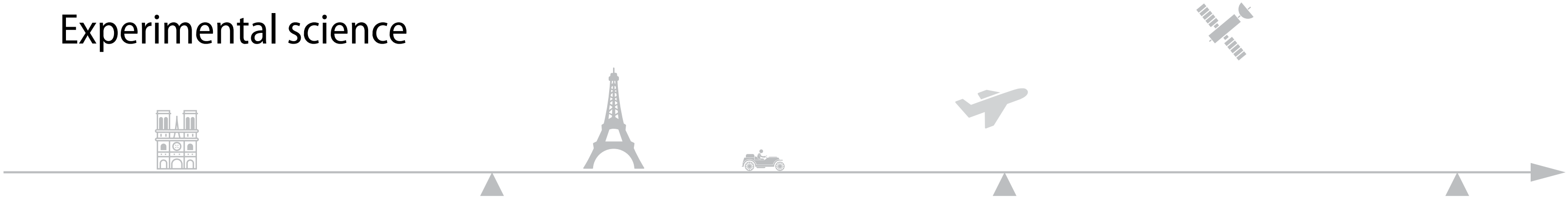
Theoretical science

3^{rd} paradigm

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



Computational science

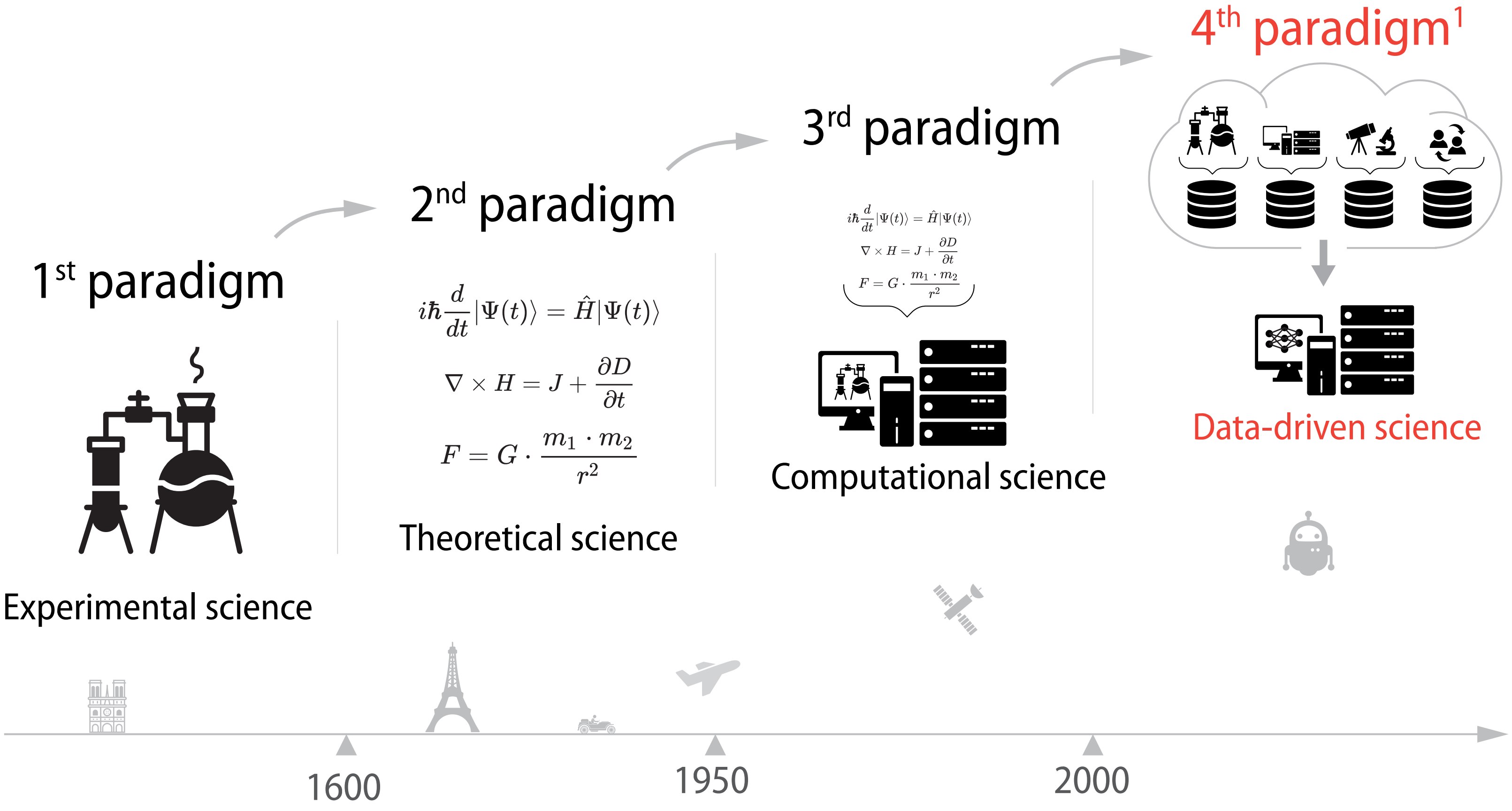


1600

1950

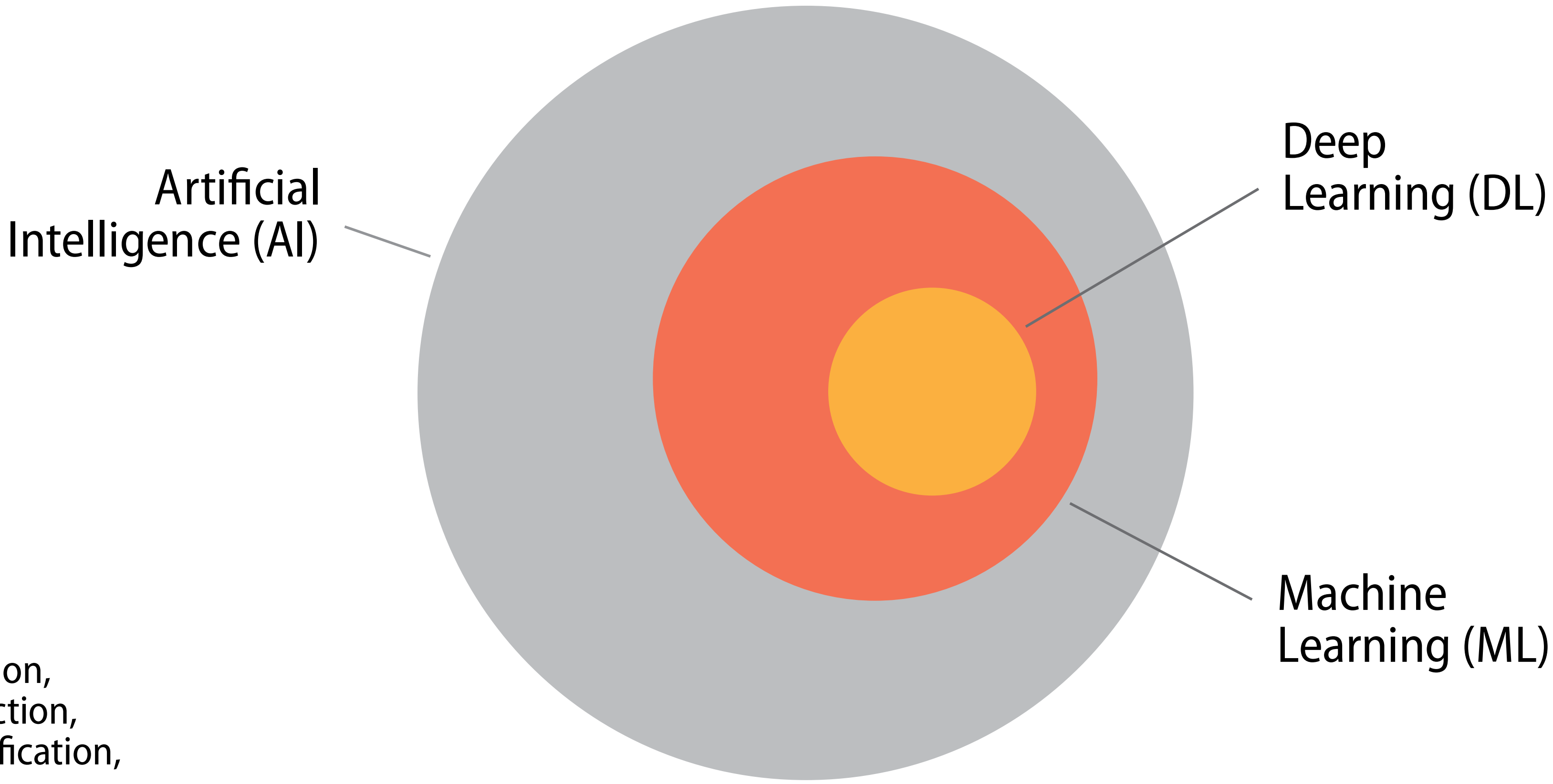
2000

[Méthode scientifique]

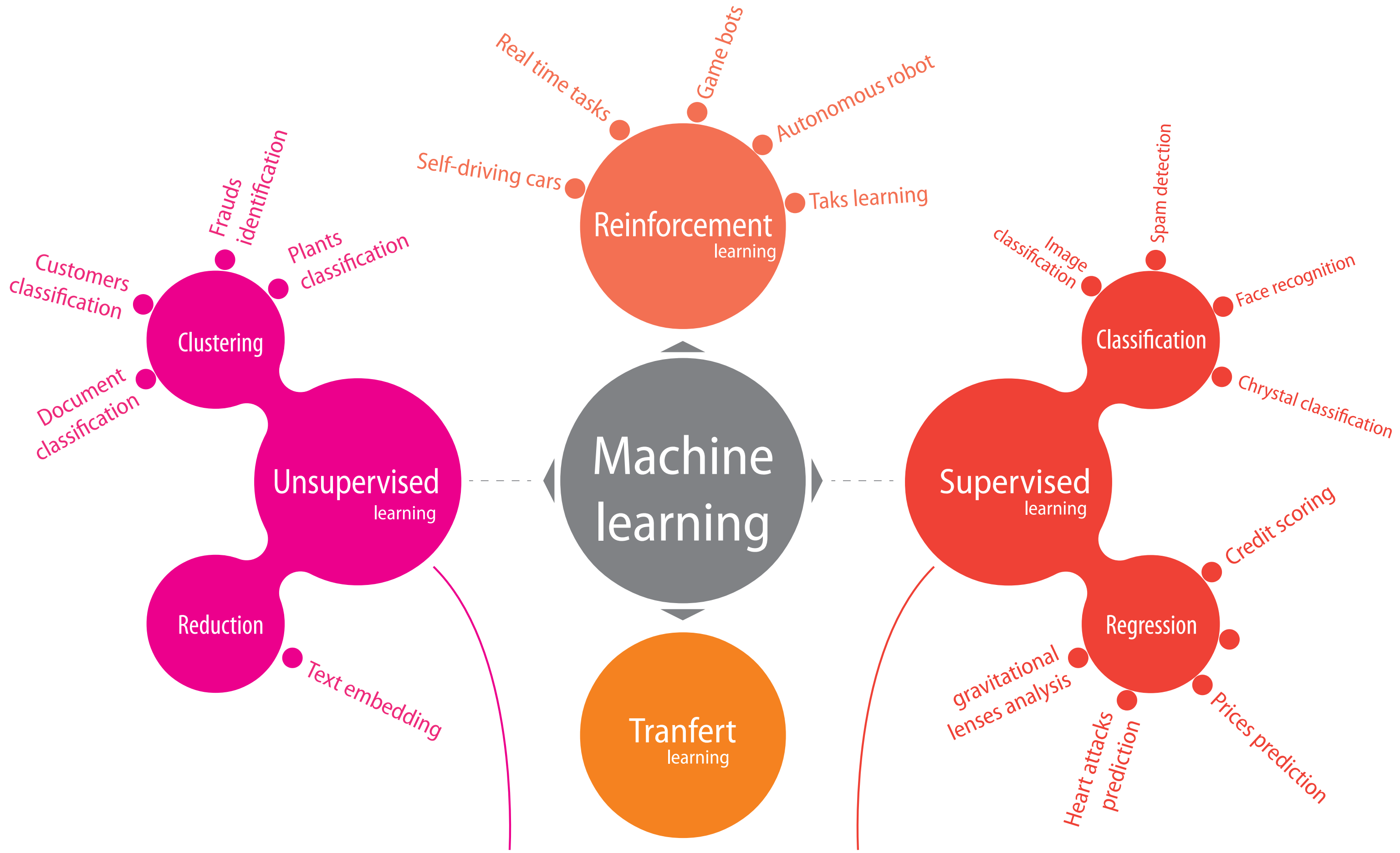


[1] Jim Gray, 2007 [GRAY]

[*-learning]

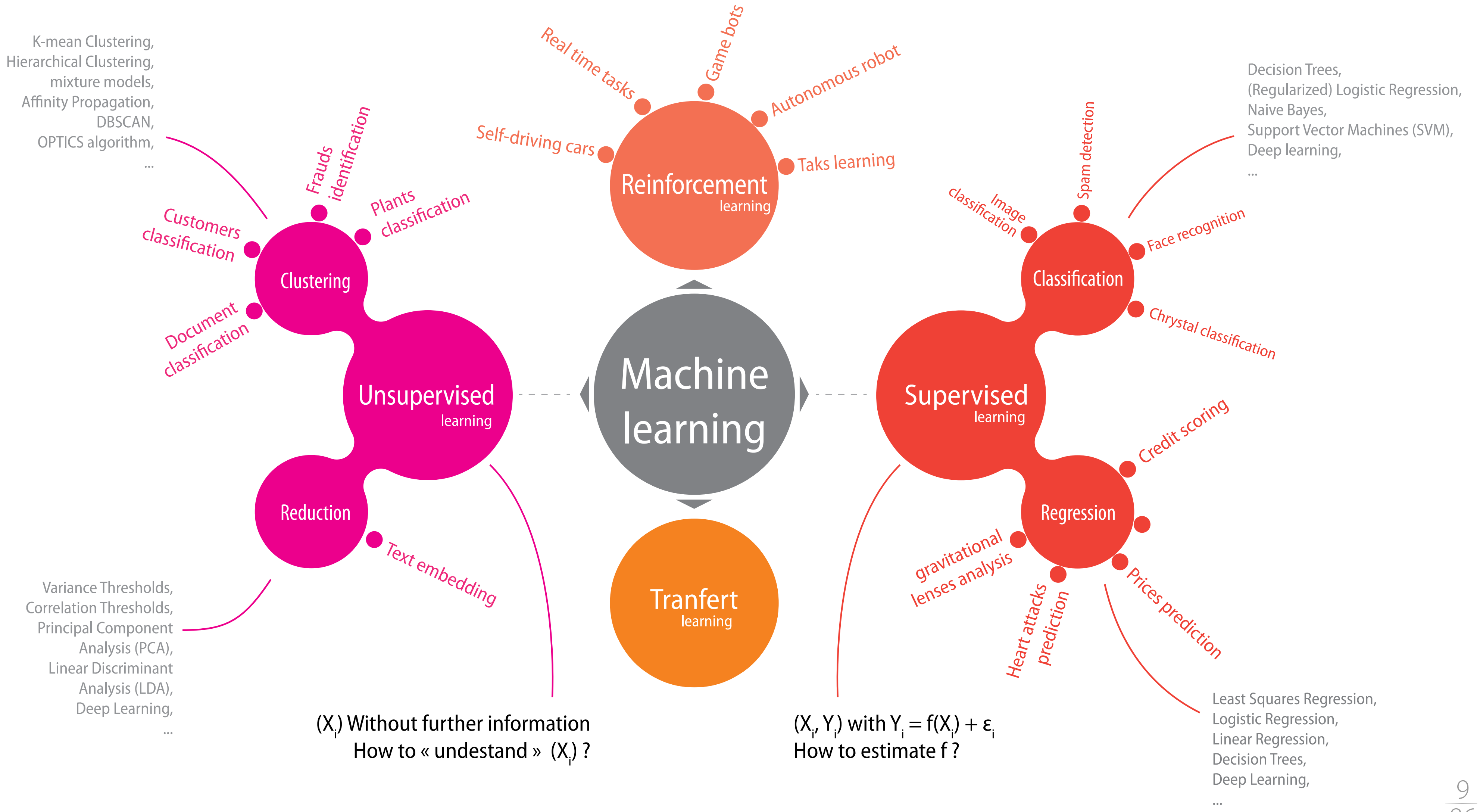


Decision,
Prediction,
Classification,
etc.



(X_i) Without further information
How to « undestand » (X_i) ?

(X_i, Y_i) with $Y_i = f(X_i) + \varepsilon_i$
How to estimate f ?



Machine learning
Unsupervised learning
Clustering
Reduction
Reinforcement learning
Supervised learning
Classification
Regression
Tranfert learning
Real time tasks
Game bots
Autonomous robot
Self-driving cars
Taks learning
Frauds identification
Plants classification
Customers classification
Document classification
Text embedding
K-mean Clustering,
Hierarchical Clustering,
mixture models,
Affinity Propagation,
DBSCAN,
OPTICS algorithm,
...
Variance Thresholds,
Correlation Thresholds,
Principal Component Analysis (PCA),
Linear Discriminant Analysis (LDA),
Deep Learning,
...
Image classification
Spam detection
Face recognition
Chrystal classification
Decision Trees,
(Regularized) Logistic Regression,
Naive Bayes,
Support Vector Machines (SVM),
Deep learning,
...
Credit scoring
gravitational lenses analysis
Heart attacks prediction
Prices prediction
Least Squares Regression,
Logistic Regression,
Linear Regression,
Decision Trees,
Deep Learning,
...
(X_i) Without further information
How to « undestand » (X_i) ?
(X_i, Y_i) with $Y_i = f(X_i) + \varepsilon_i$
How to estimate f ?

1/ From the linear regression to the first neuron

2/ Neural networks at the heart of a controversy

3/ Neurons & data

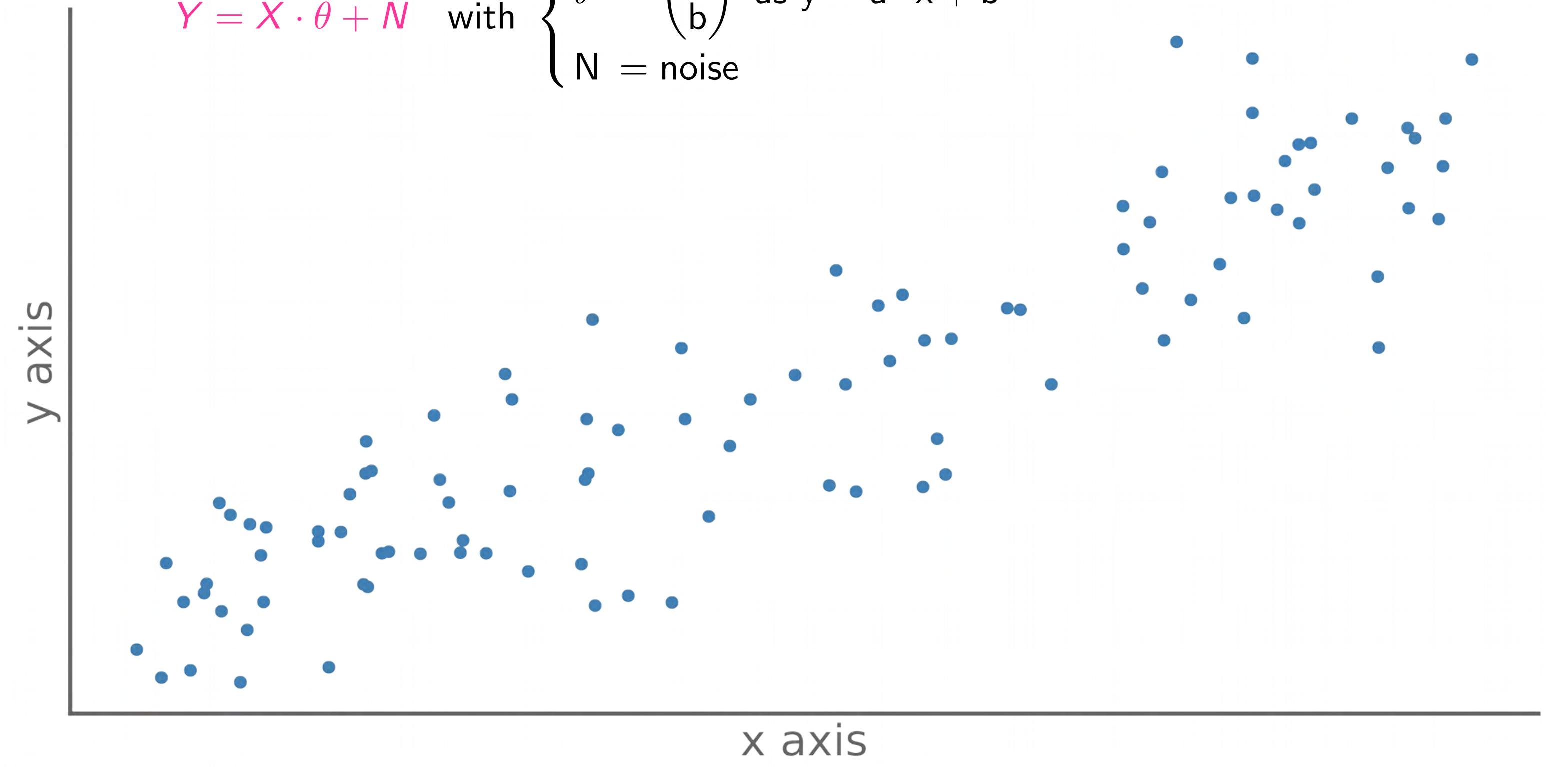
4/ Conclusion

1/ From the linear regression to the first neuron

...there's a little bit of math hidden behind the neurons...

Linear regression

$$Y = X \cdot \theta + N \quad \text{with} \quad \begin{cases} \theta = \begin{pmatrix} a \\ b \end{pmatrix} \text{ as } y = a \cdot x + b \\ N = \text{noise} \end{cases}$$

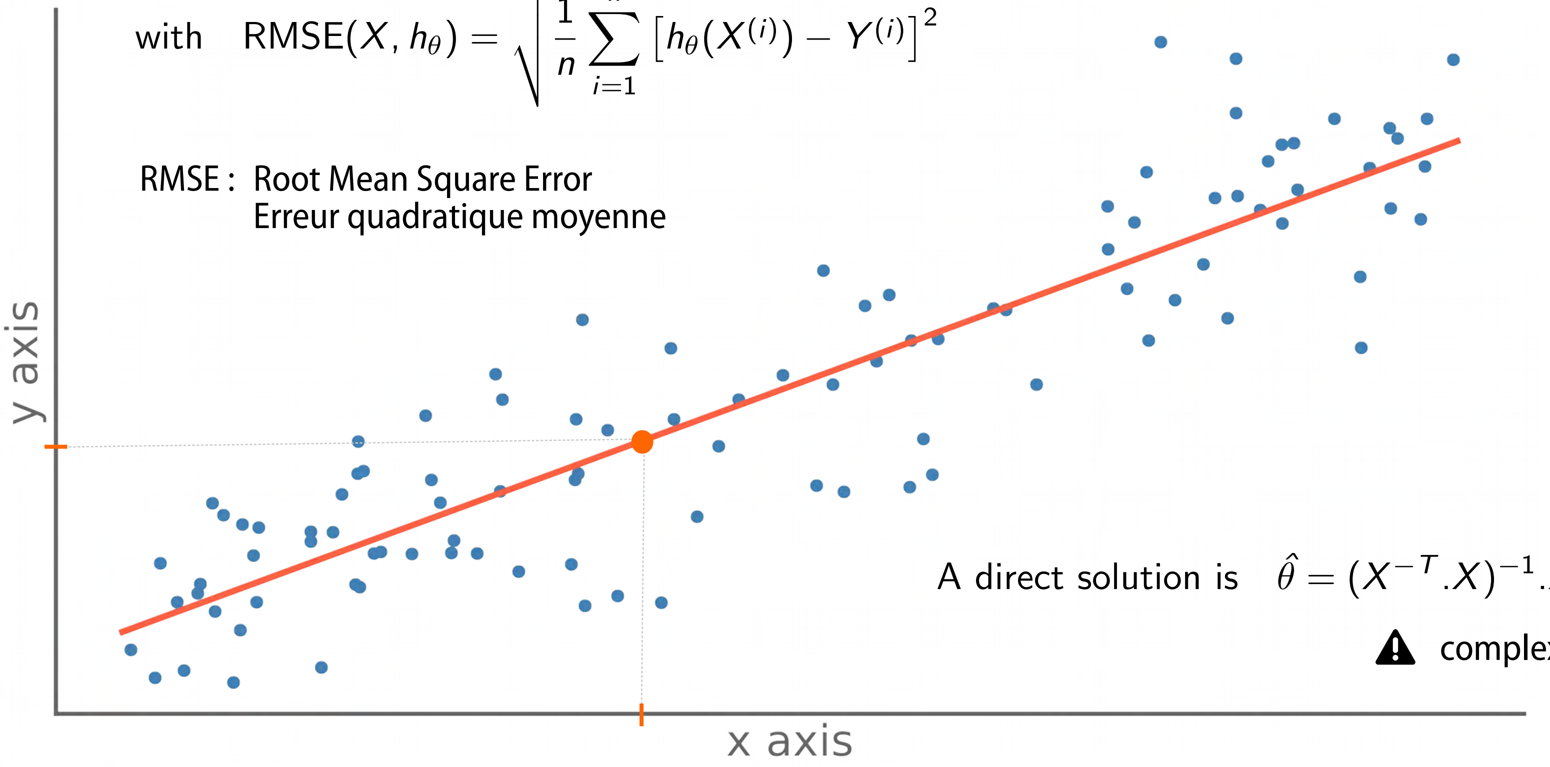


Linear regression

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

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

RMSE : Root Mean Square Error
Erreur quadratique moyenne

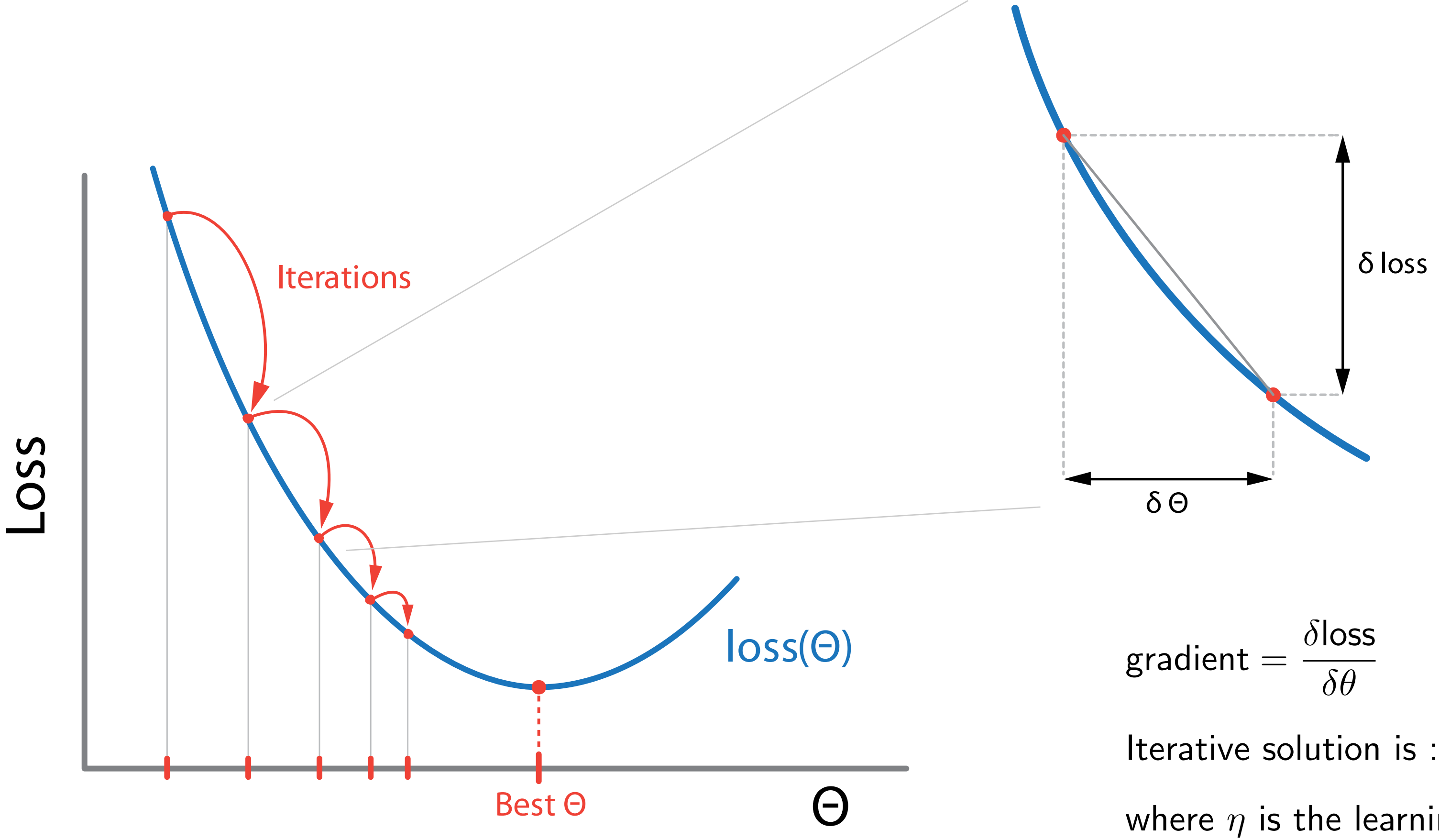


A direct solution is $\hat{\theta} = (X^{-T}.X)^{-1}.X^{-T}.Y$

⚠ complexity in n^3

Notebook [LAB1]

Gradient descent



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

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

where η is the learning rate

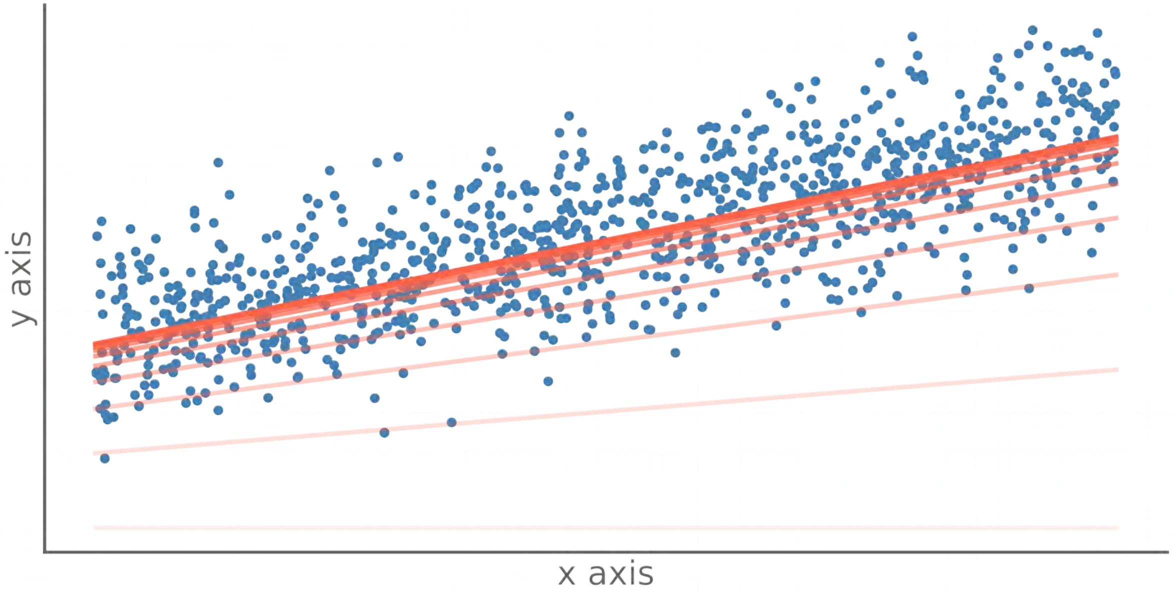
Gradient descent

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

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

Iterative solution is : $\Theta \leftarrow \Theta - \eta \cdot \nabla_\theta MSE(\Theta)$

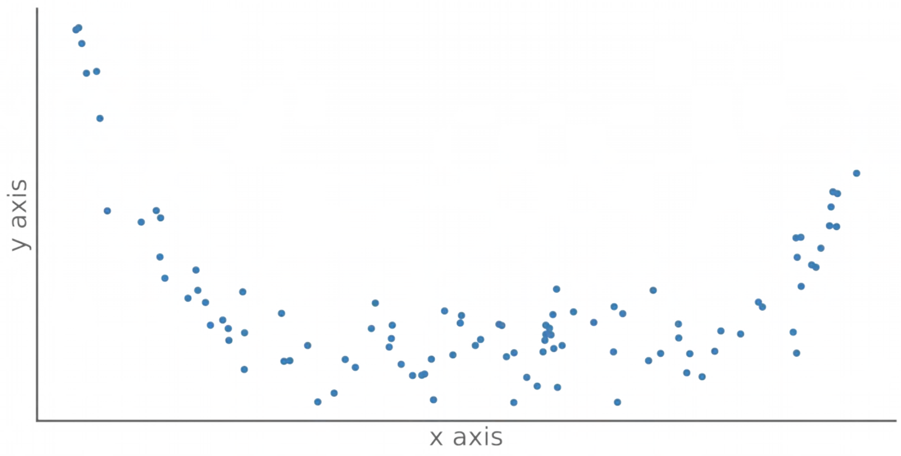
where η is the learning rate

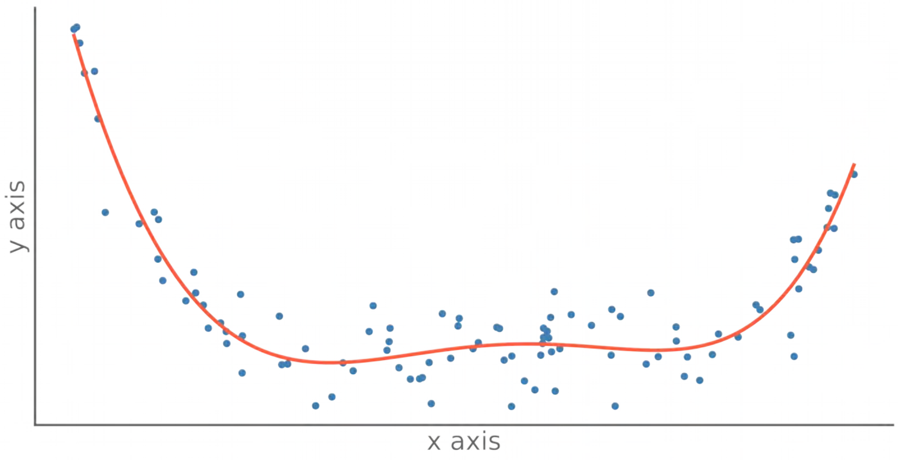


#i	Loss	Gradient		Theta	
0	+12.481	-6.777	-1.732	-3.388	+0.000
20	+4.653	-4.066	-1.039	-2.033	+0.346
40	+1.835	-2.440	-0.624	-1.220	+0.554
60	+0.821	-1.464	-0.374	-0.732	+0.679
80	+0.455	-0.878	-0.224	-0.439	+0.754
100	+0.324	-0.527	-0.135	-0.263	+0.799
120	+0.277	-0.316	-0.081	-0.158	+0.826
140	+0.260	-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

Notebook [LAB2]

Polynomial regression





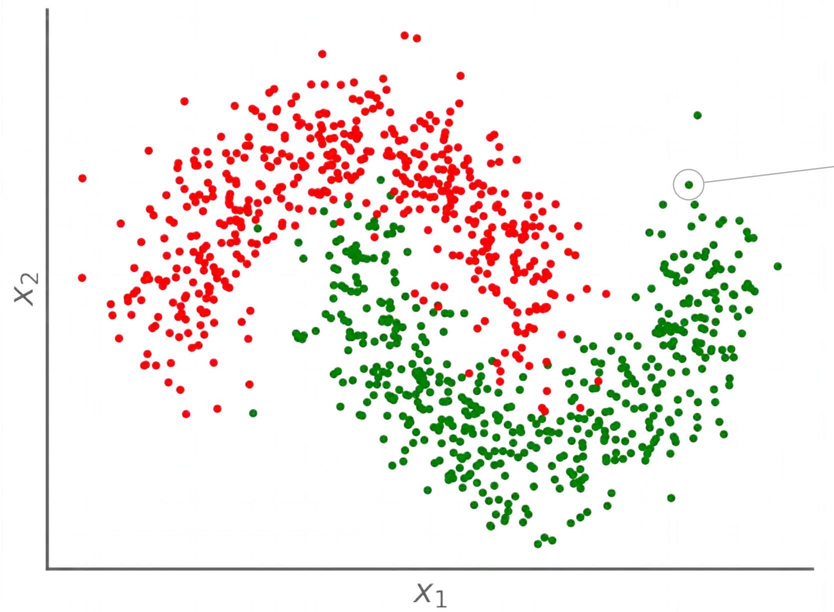
$$P_n(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \cdots + a_n \cdot x^n = \sum_{i=0}^{n} a_i \cdot x^n$$

Notebook [LAB3]

Logistic regression

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

Dataset : X characteristics
y probability of belonging



$$(X_i, y_i) \begin{cases} X_i = \begin{pmatrix} x_{i1} \\ \vdots \\ x_{in} \end{pmatrix} \\ y_i = \begin{cases} 1 & \text{belong to the class} \\ 0 & \text{don't belong} \end{cases} \end{cases}$$

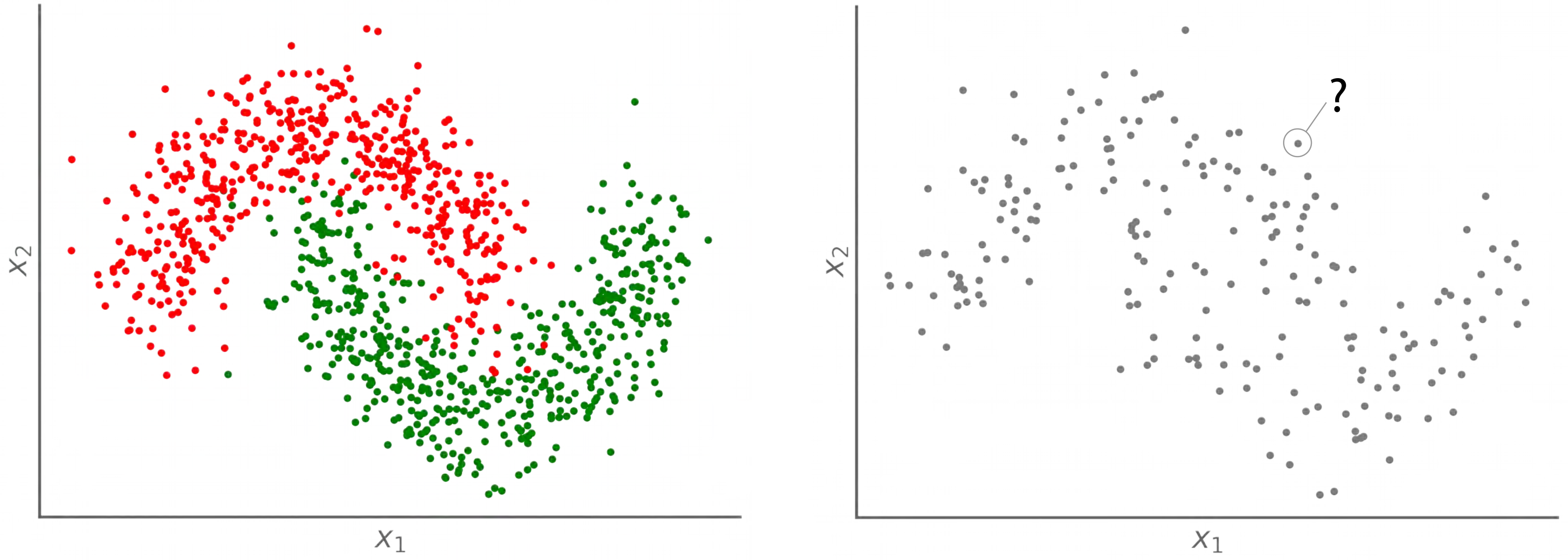
Notebook [LAB12]

Logistic regression

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

Dataset : X Observations
y Classe

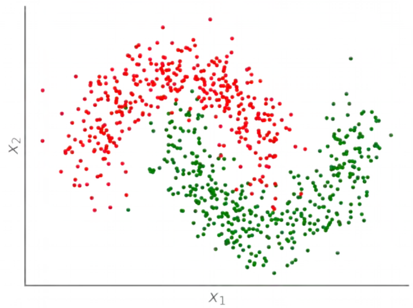
Objective : Predict the class
X given, we want to predict y



Notebook [LAB12]

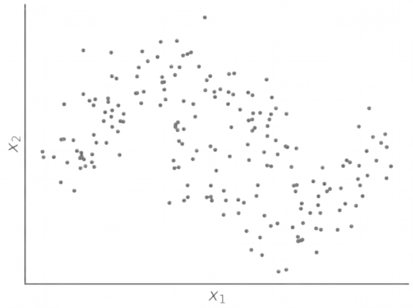
Logistic regression

80 %



Données
(X,y)

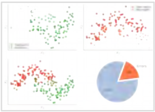
20 %



Learning phase

Evaluation phase

y_{test} vs $\hat{y}_{test}$



Determination of Θ by a minimisation of the log loss J(Θ)

$$\hat{y} = \begin{cases} 0 & \text{if } \hat{p} < 0.5 \\ 1 & \text{if } \hat{p} \geq 0.5 \end{cases}$$

where

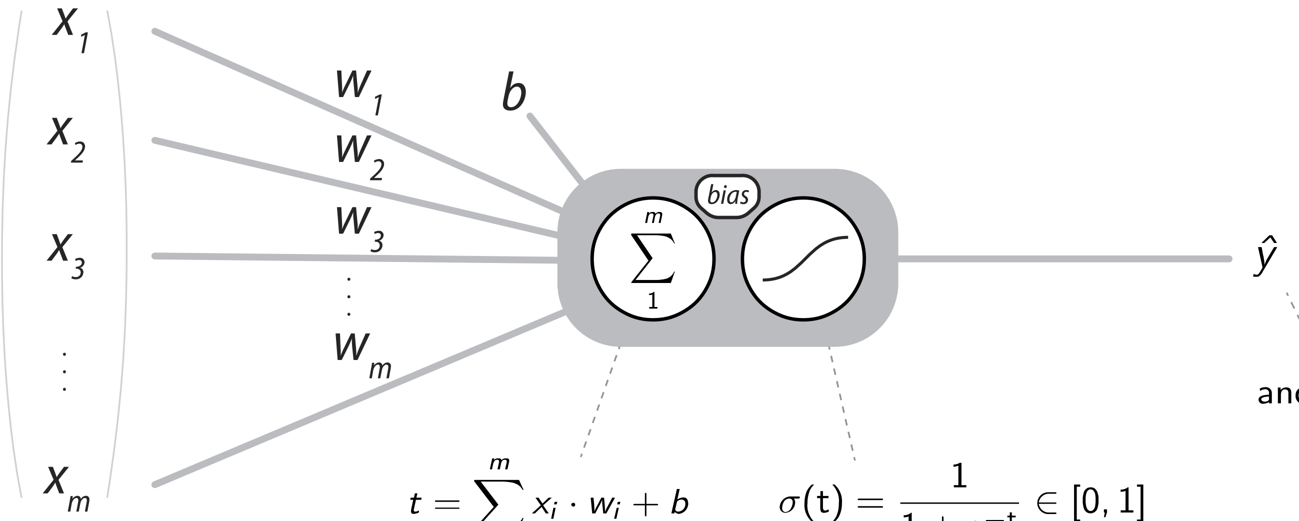
$$\hat{p} = h_\theta(X) = \sigma(\theta^T X + b)$$

and

$$\sigma(t) = \frac{1}{1 + \exp(-t)}$$

Notebook [LAB12]

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



and $\bar{y} = \begin{cases} 0 & \text{if } \hat{y} < 0.5 \\ 1 & \text{if } \hat{y} \geq 0.5 \end{cases}$

$$t = \sum_{1}^{m} x_i \cdot w_i + b \qquad \sigma(\mathrm{t}) = \frac{1}{1 + \mathrm{e}^{-\mathrm{t}}} \in [0, 1]$$

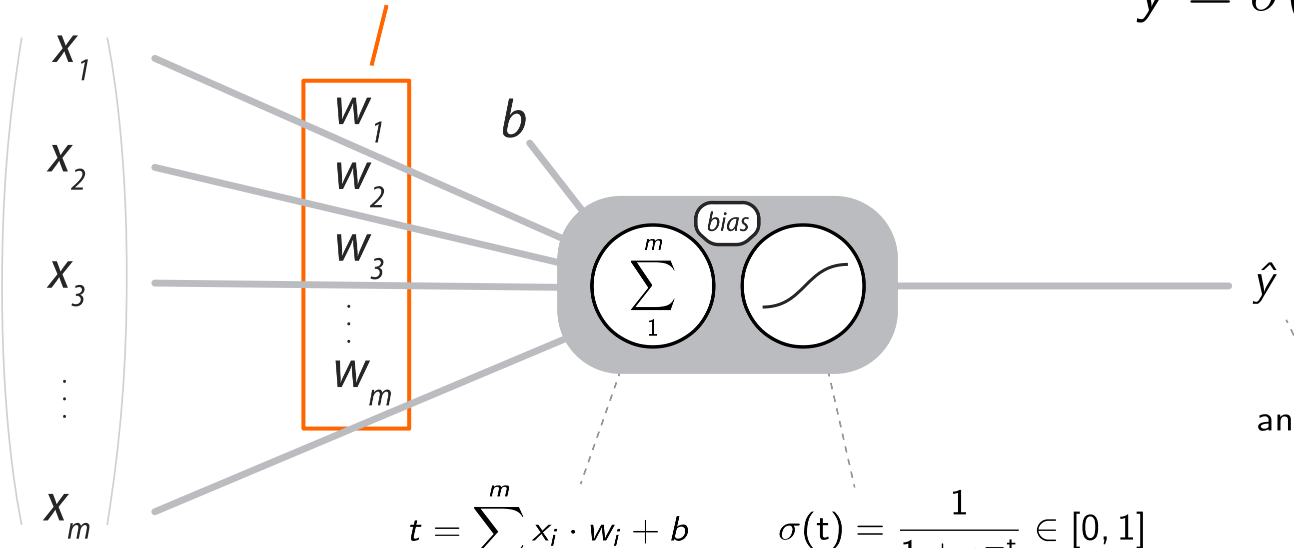
Input	**Bias / Weight**	**Activation function**	**Output**
X	Θ	$\sigma(t)$	$\hat{y}$

Notebook [LAB12]

Logistic regression

Determined by the minimisation of a cost function J(Θ)

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



$$\text{and } \bar{y} = \begin{cases} 0 & \text{if } \hat{y} < 0.5 \\ 1 & \text{if } \hat{y} \geq 0.5 \end{cases}$$

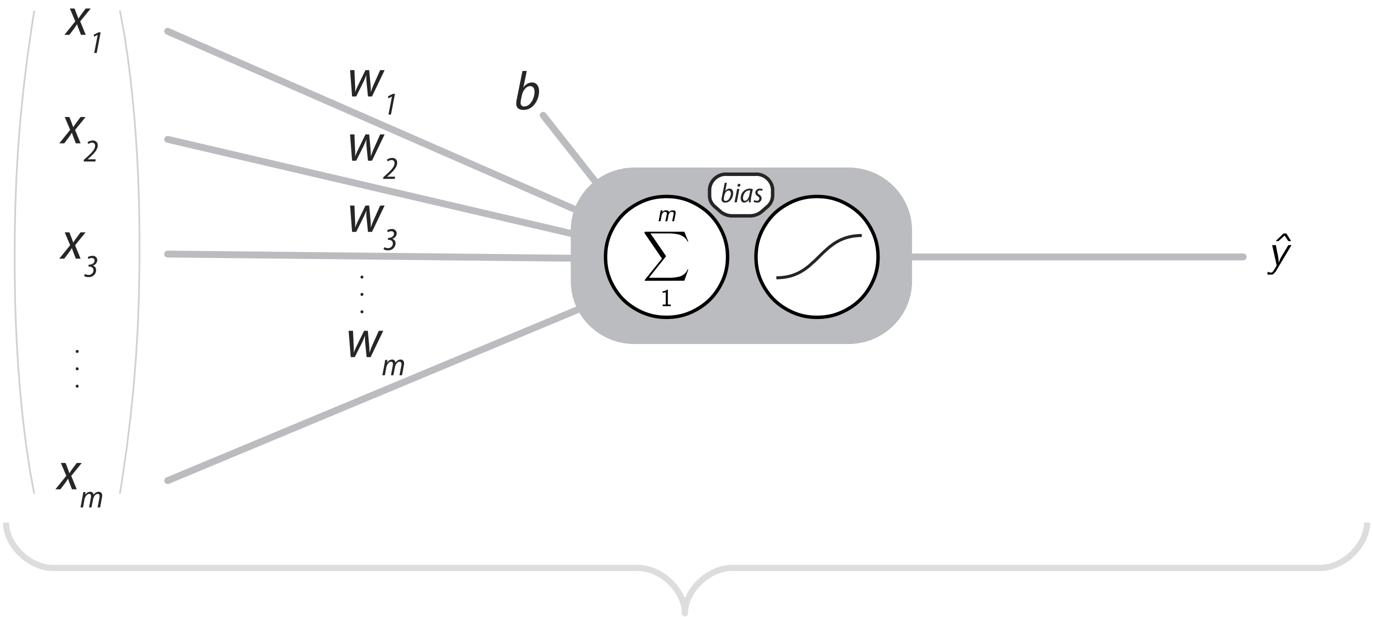
$$t = \sum_1^m x_i \cdot w_i + b \qquad \sigma(\mathrm{t}) = \frac{1}{1 + \mathrm{e}^{-\mathrm{t}}} \in [0, 1]$$

Input	**Bias / Weight**	**Activation function**	**Output**
X	Θ	$\sigma(t)$	$\hat{y}$

Notebook [LAB12]

Logistic regression

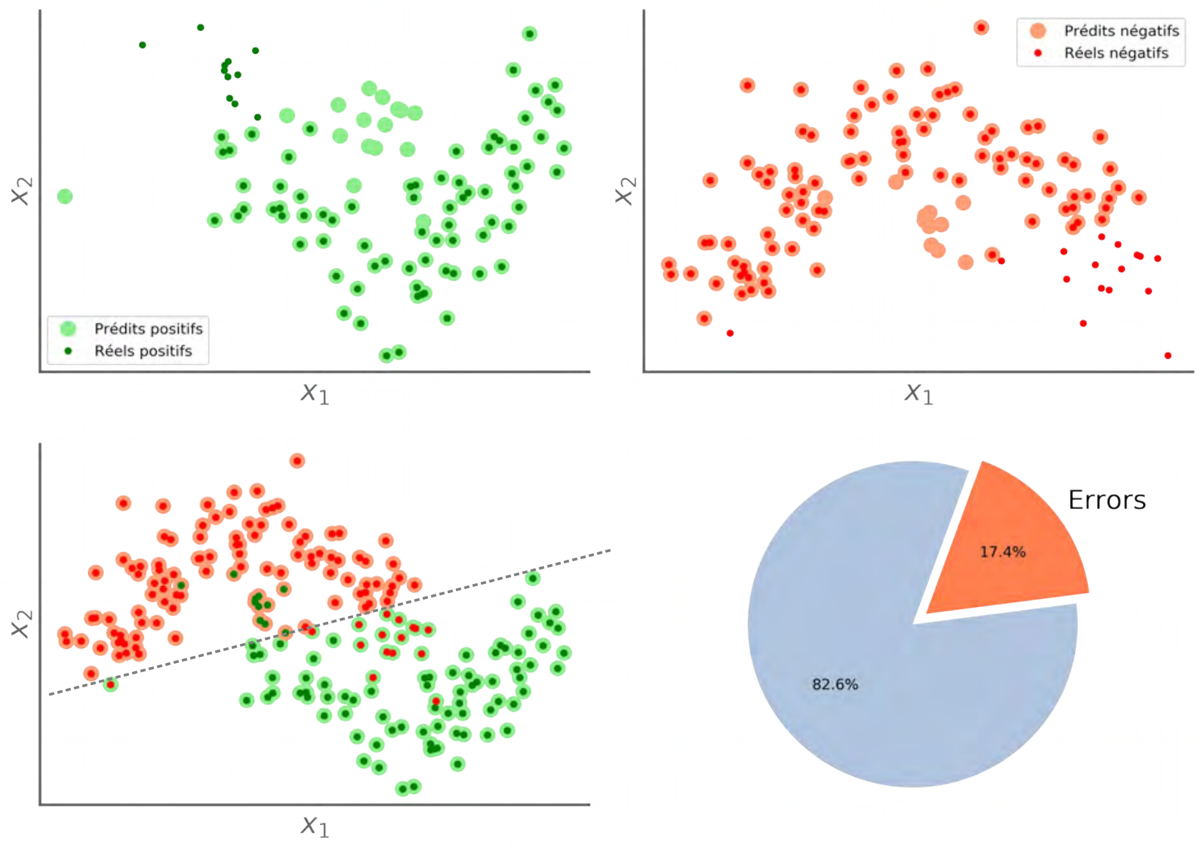
$$\hat{y} = \sigma(\Theta^T \cdot X + b)$$



That's an « artificial neuron » !

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

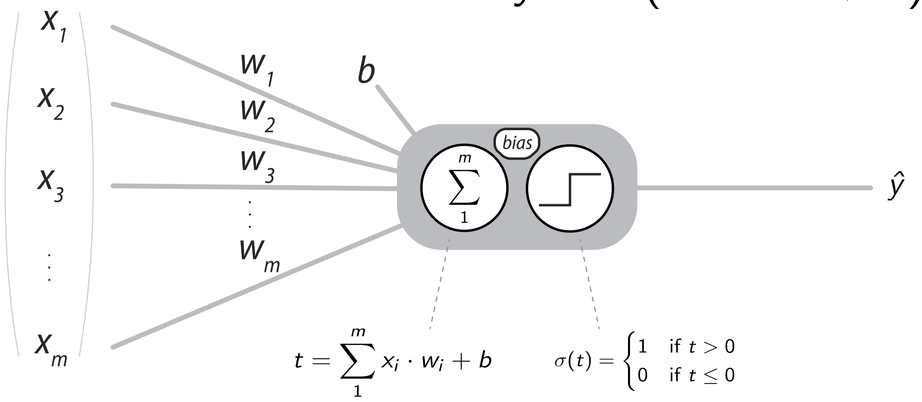
Logistic regression

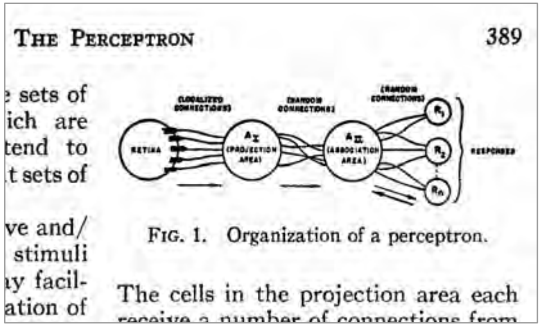


Notebook [LAB12]

Perceptron

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




THE PERCEPTRON 389

e sets of
ich are
tend to
t sets of

ve and/
stimuli
ly facil-
ation of

FIG. 1. Organization of a perceptron.

The cells in the projection area each

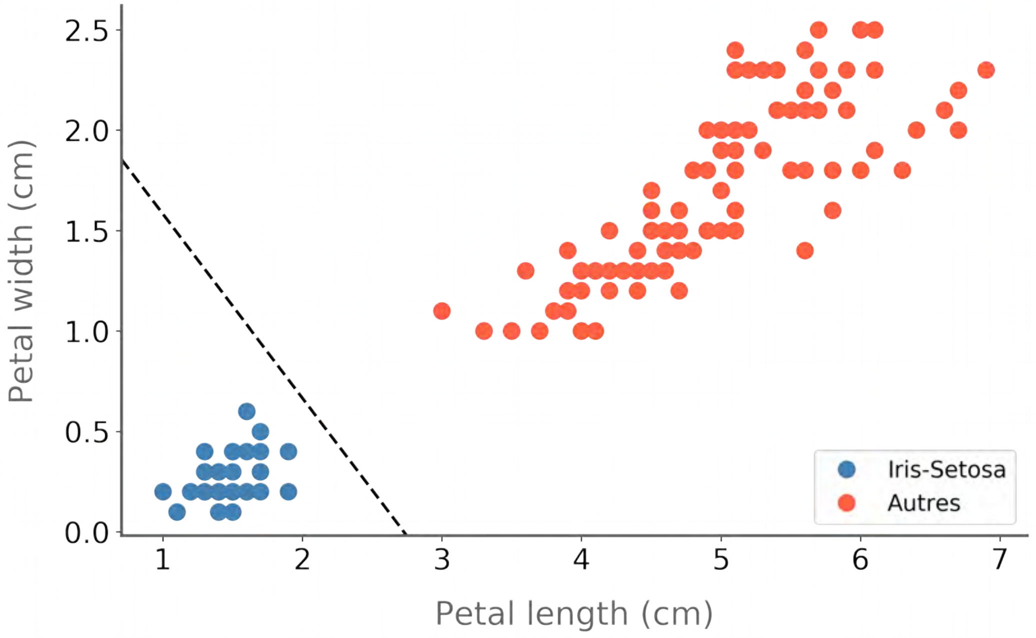
Perceptron
Frank Rosenblatt
1958

Linear and binary classifier

F. Rosenblatt, 1958 [FROS]

Iris plants dataset

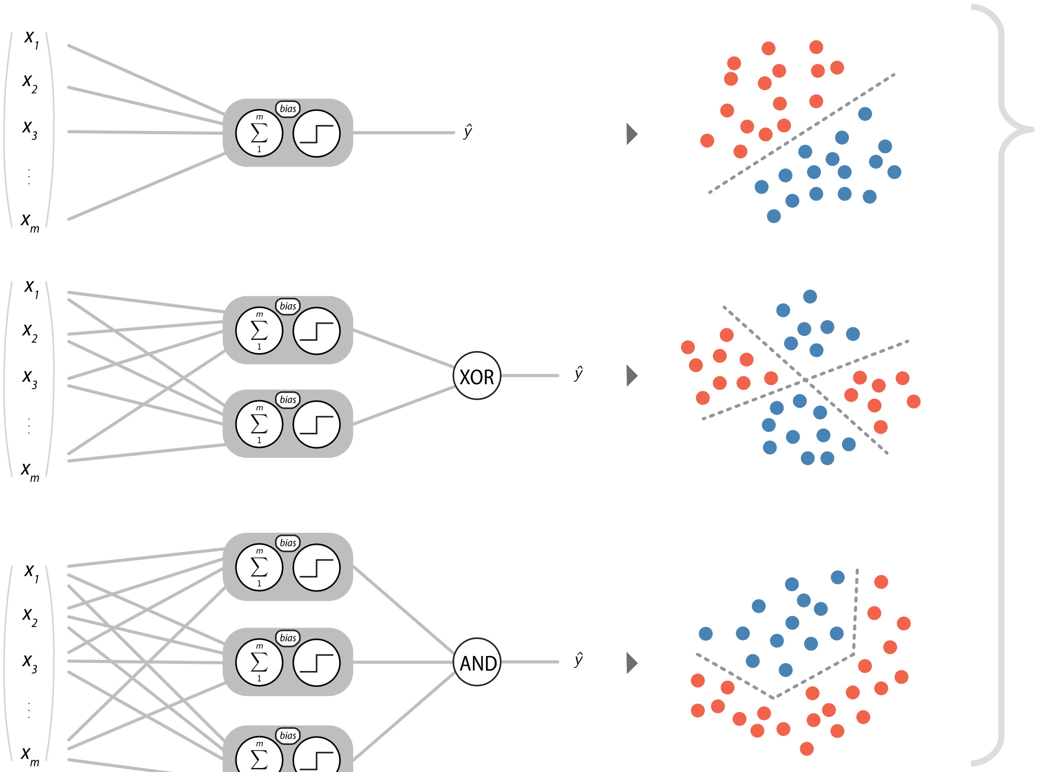
Dataset from : Fisher, R.A. "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188 (1936)



Length	Width	Iris Setosa (0/1)
x1	x2	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
(...)		

Notebook [LAB13]

Perceptron



Linear classifier...

1969

Marvin Minsky, Seymour Papert

« *Perceptrons : An Introduction to Computational Geometry* » [1]

First AI winter…

(for neural networks)

[1] Minsky, Marvin; Papert, Seymour, (1969) [MIPA]

2/ Neural networks at the heart of a controversy

The big Controversy

Modelling the brain :

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

L'information est un **signal** avant
d'être un code »[1]

Making a mind :

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

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

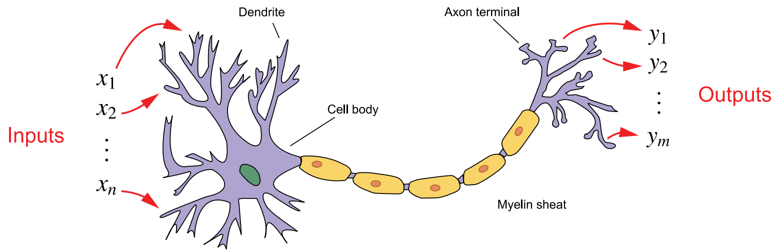
Connectionnism

vs

Symbolic

Modelling the brain
Modéliser le cerveau

Making a mind
Forger une opinion




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

Donc [Socrate] est [mortel]
```

[1] D Cardon, JP Cointet, A Mazieres, 2018 [LRDN]

The big Controversy

Inductive approach

Input

Output

Program
(model)

Deductive approach

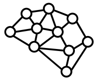
Input

Program

Output

Connectionnism vs **Symbolic**

Facts ▶ Rules and laws



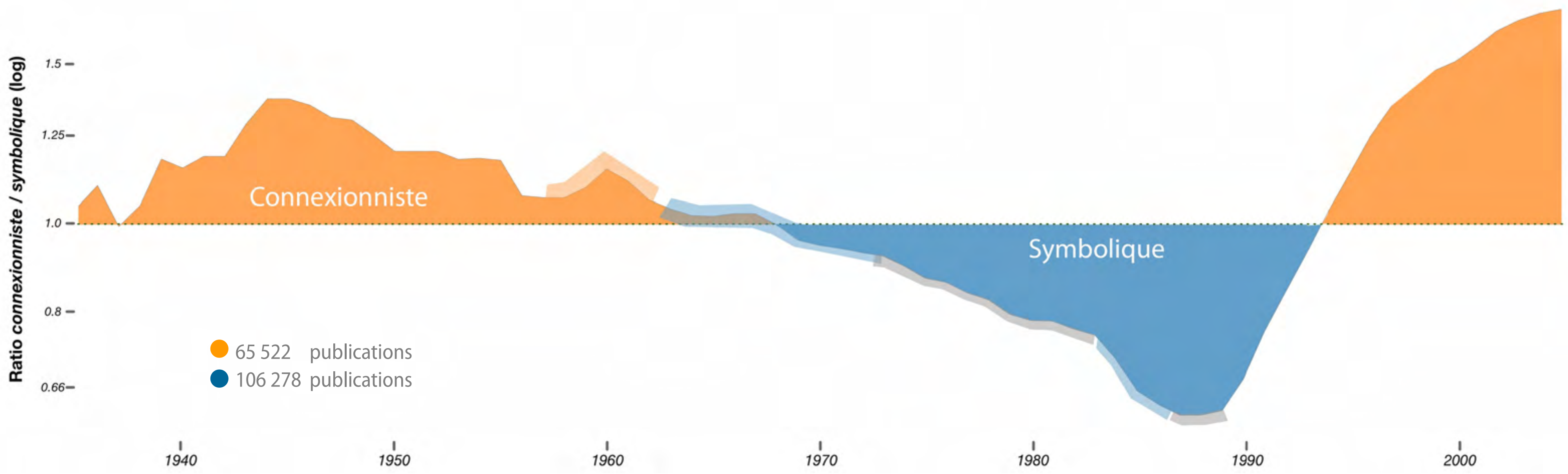
Model

Expert

Rules and laws ▶ Special case

Evolution of the academic influence of connexionist and symbolic approaches[1]

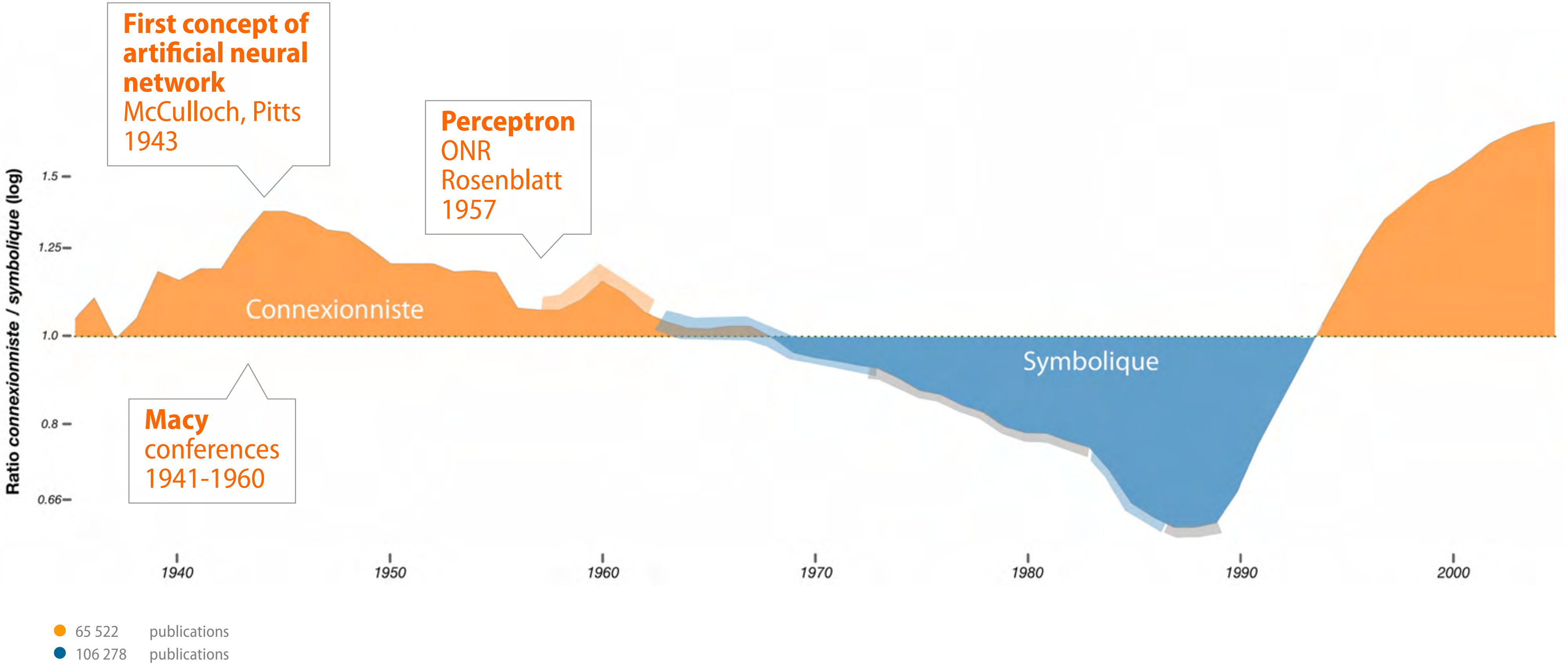
Ration of publications between connexionists and symbolists



[1] D Cardon, JP Cointet, A Mazieres, 2018 [LRDN]

The big Controversy

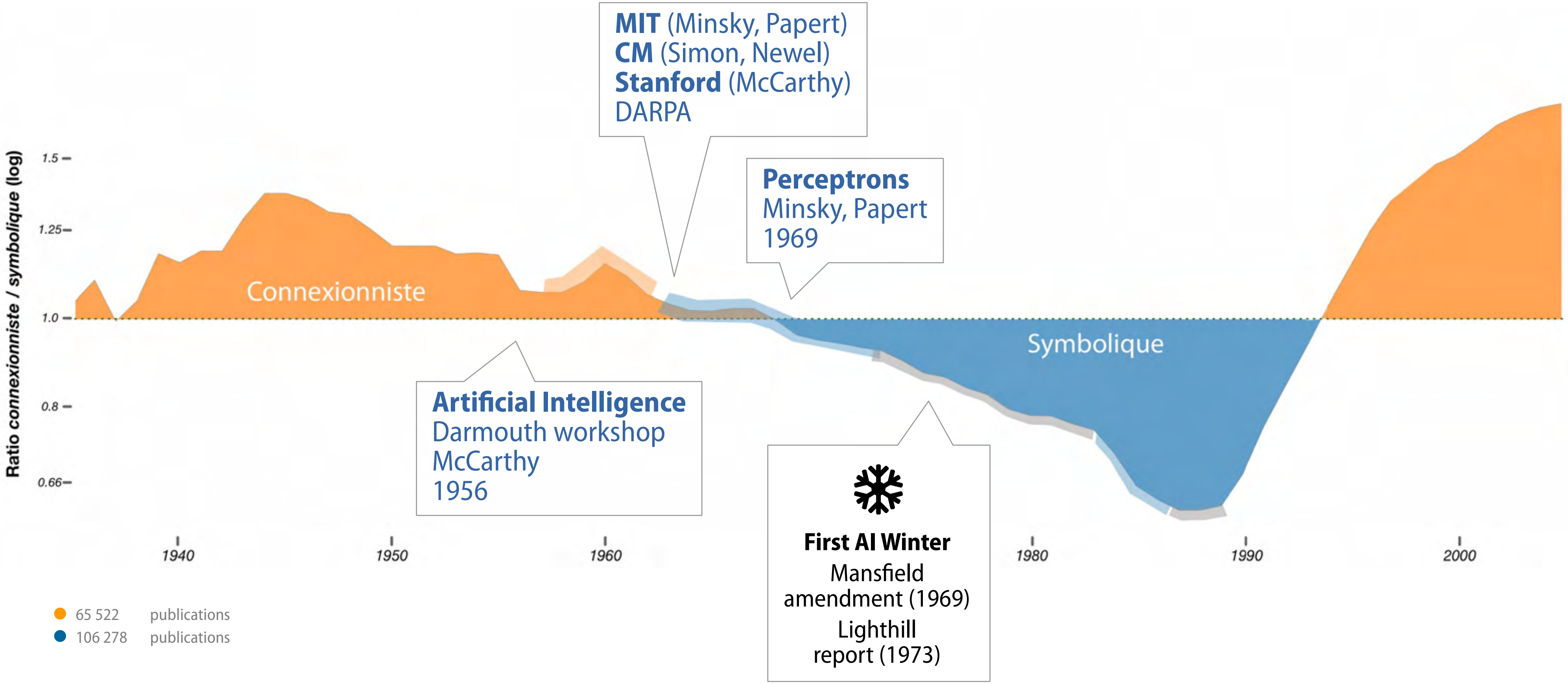
Evolution of the academic influence of connexionist and symbolic approaches[1]



[1] D Cardon, JP Cointet, A Mazieres, 2018 [LRDN]

The big Controversy

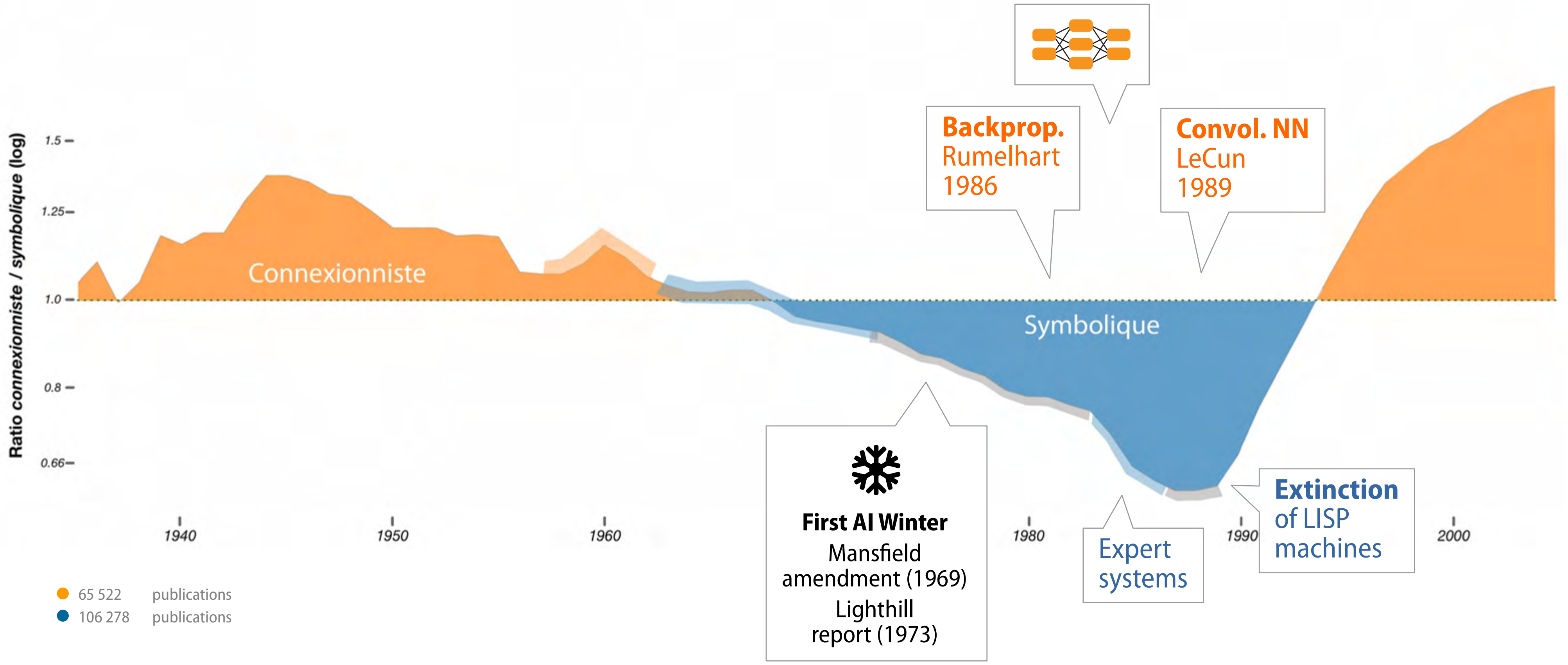
Evolution of the academic influence of connexionist and symbolic approaches[1]



[1] D Cardon, JP Cointet, A Mazieres, 2018 [LRDN]

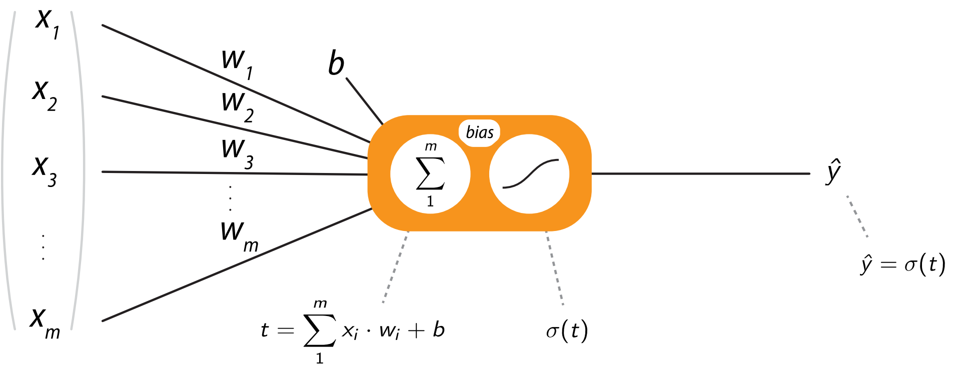
The big Controversy

Evolution of the academic influence of connexionist and symbolic approaches[1]

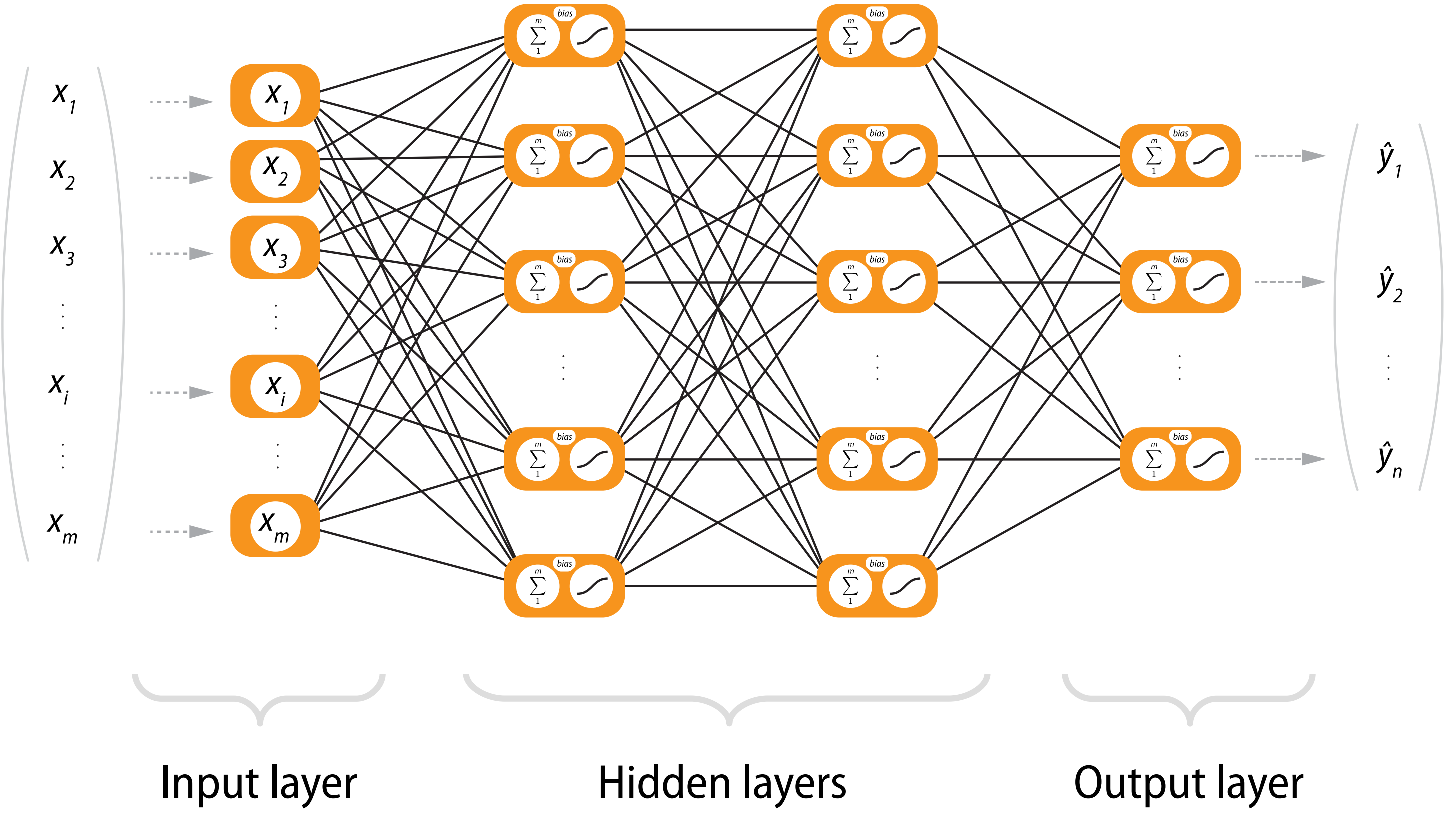


[1] D Cardon, JP Cointet, A Mazieres, 2018 [LRDN]

Deep Neural Networks



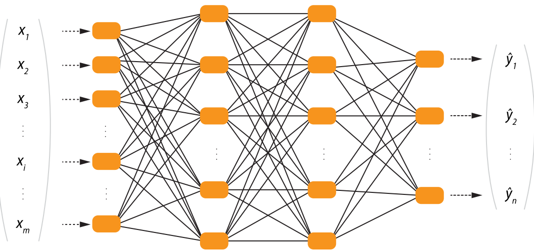
Deep Neural Networks



Observation

Feed forward

$X^{(i)}$



$$\forall X^{(i)} \in \left\{X^{(1)}, X^{(2)}, \ldots, X^{(m)}\right\}$$

Predicted output

$\hat{y}^{(i)}$

Expected output

$y^{(i)}$

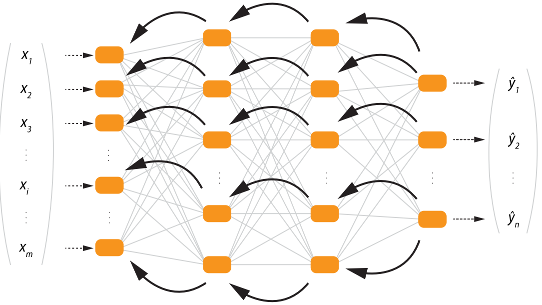
loss function

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

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

...

Repeat



Back-propagation

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

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

...

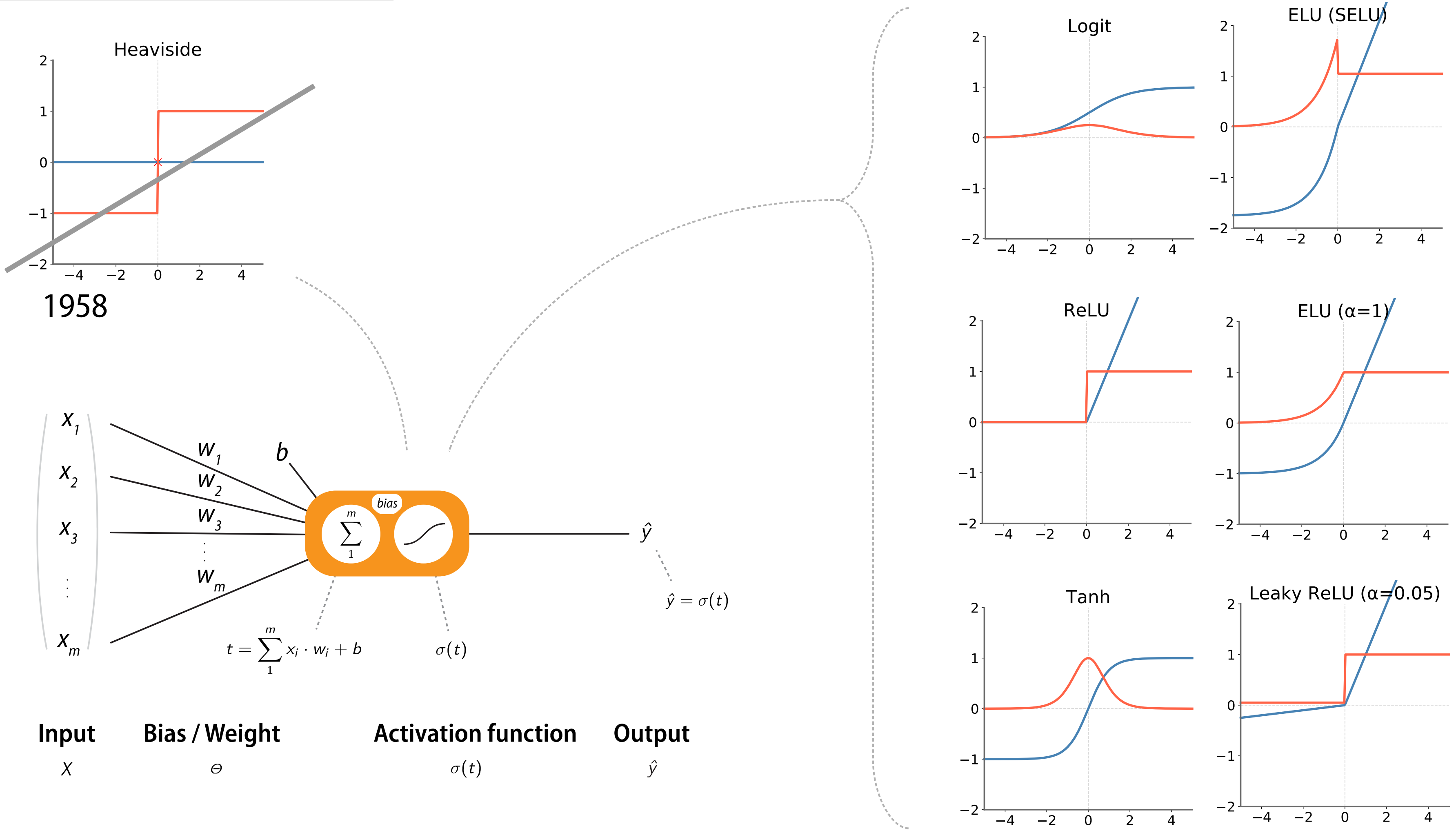
η : Learning rate
β : Momentum

Optimisations :
- Activation,
- Gradient descent,
- Dropout,
- Regularization,
- Etc.

Back-propagation

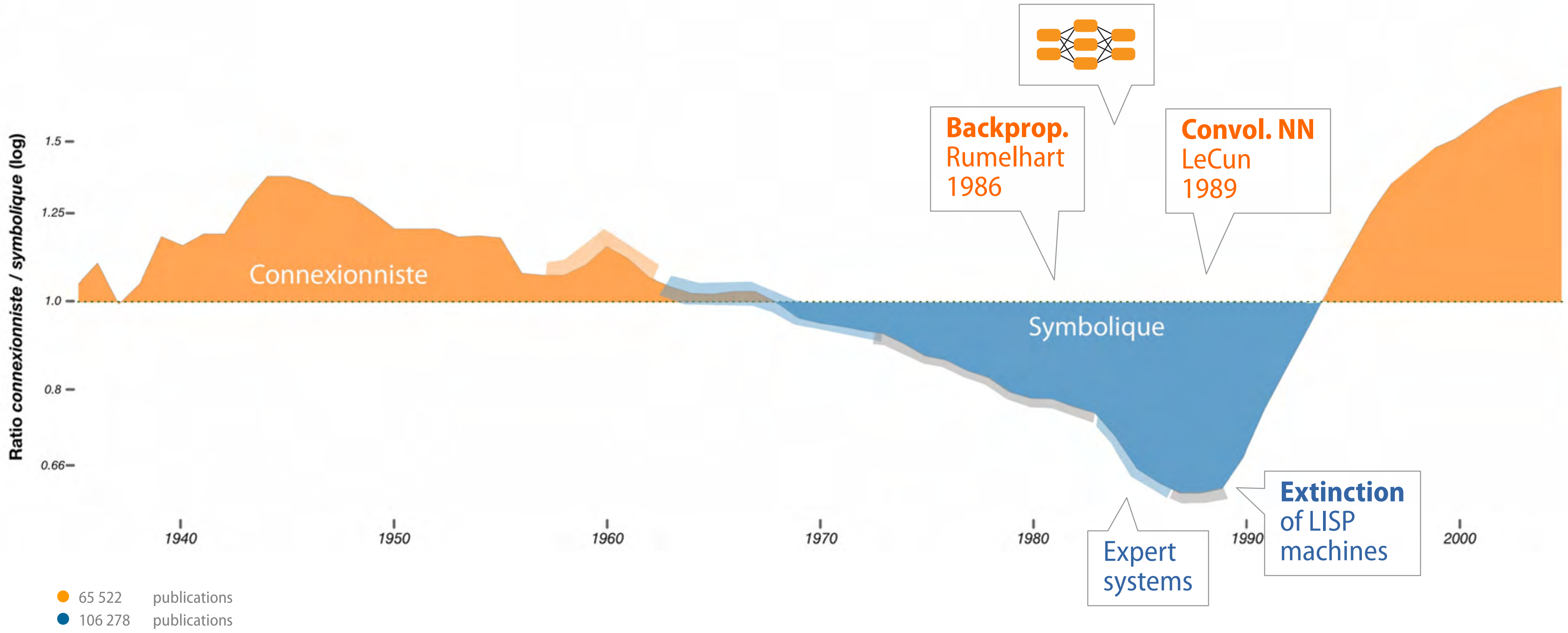
Learning process

Deep Neural Networks



The big Controversy

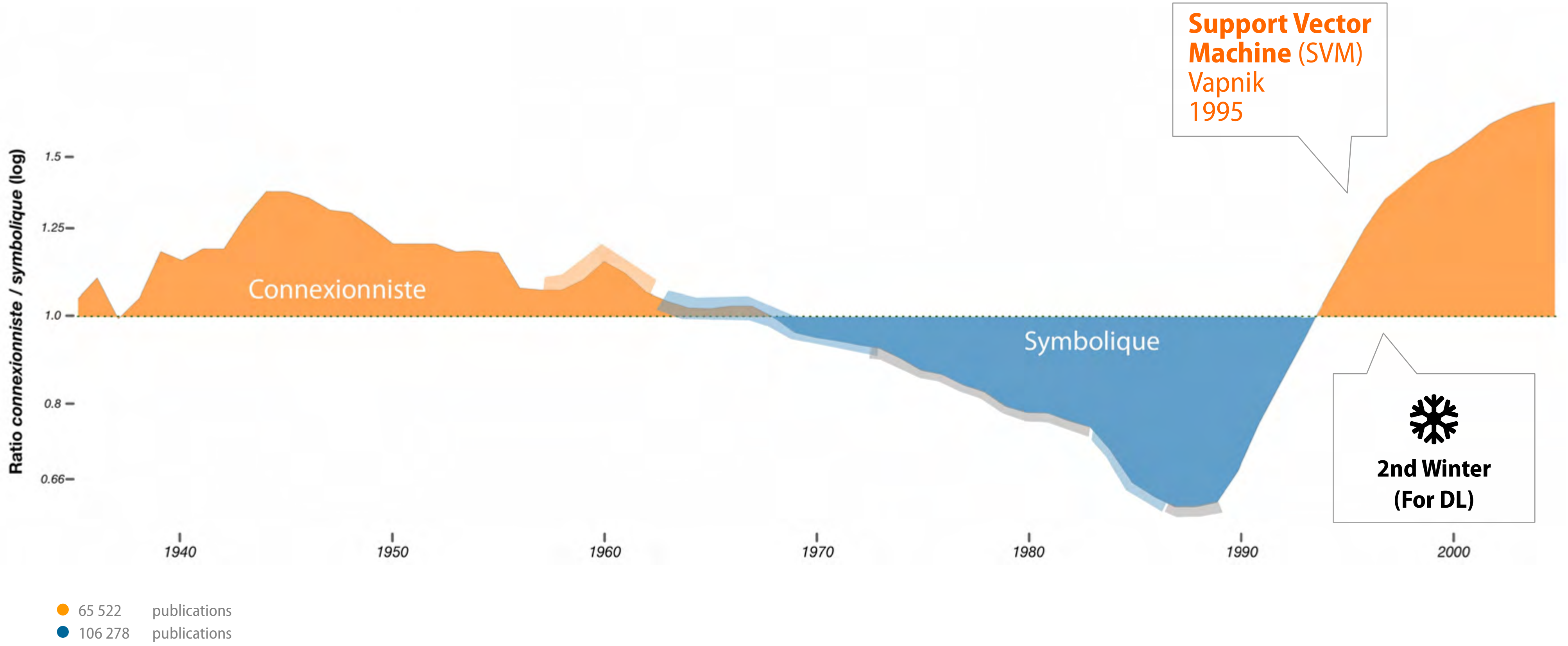
Evolution of the academic influence of connexionist and symbolic approaches[1]



[1] D Cardon, JP Cointet, A Mazieres, 2018 [LRDN]

The big Controversy

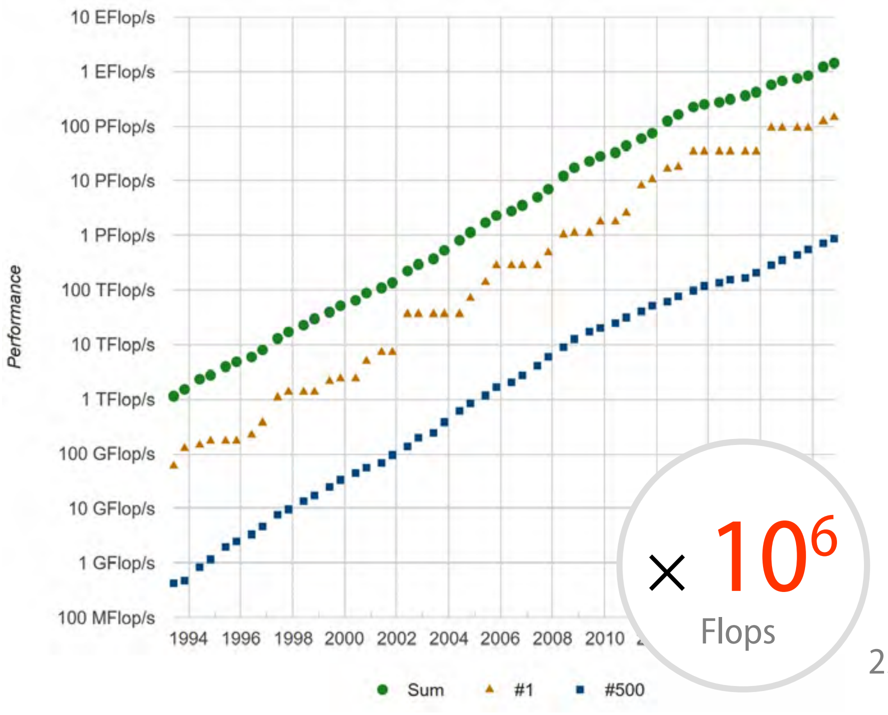
Evolution of the academic influence of connexionist and symbolic approaches[1]



[1] D Cardon, JP Cointet, A Mazieres, 2018 [LRDN]

The big Controversy

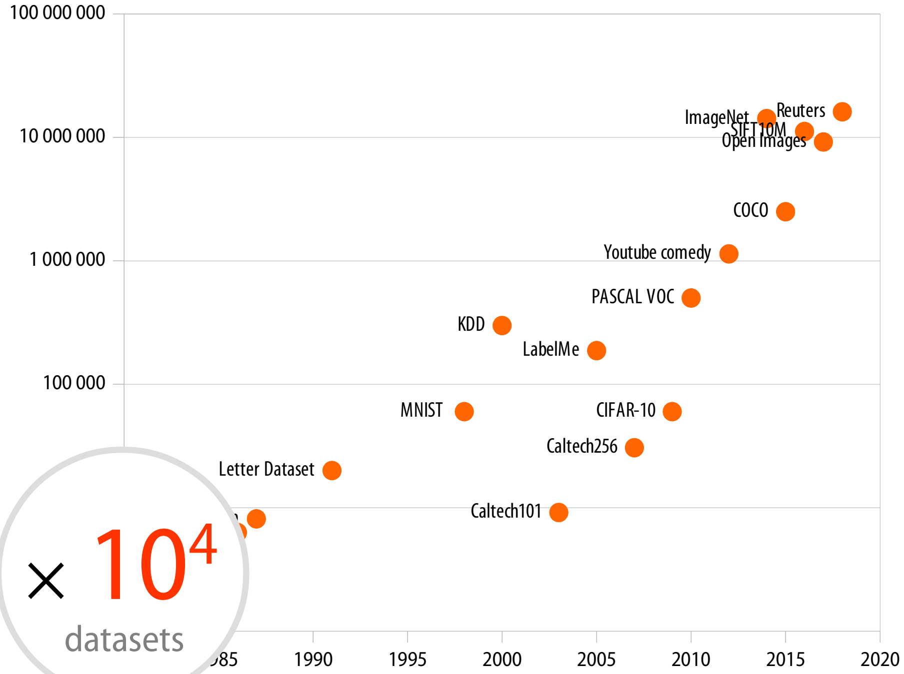
Performance Development[1]



$\times 10^6$
Flops

25 ans

Datasets for machine-learning[2]



$\times 10^4$
datasets

Laboratoire
Cas particulier

$\not\Rightarrow$

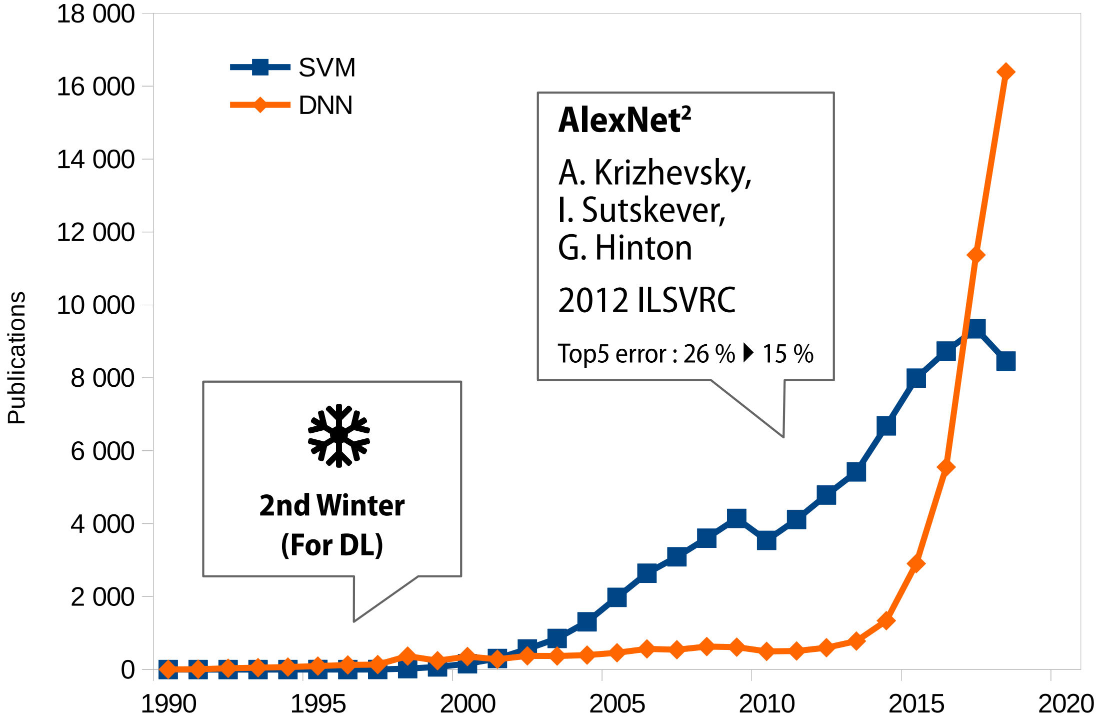
Monde réel

[1] TOP500 List [TOP500]

[2] Wikipedia [WKP1]

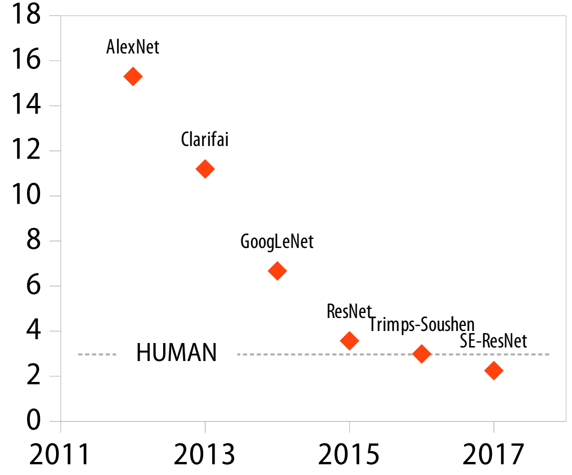
The big Controversy

Publications SVM vs DNN[1]



DN

Images classification Top 5 error at ILSVRC[3,4]



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

[1] Web of Science [WOS1][WOS2]
[2] AlexNet [ALEX]
[3] ImageNet Large Scale Visual Recognition [ILSVRC]
[4] Similar evolution in Natural language processing, translation, board games, etc. See : DeepL.com, AlphaGo, AlphaZero, ...

3/ Neurons & data

3/ Neurons & data

Generative Adversarial Network
GAN

Basic Classification
DNN

Reinforcement
learning

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

Sequences data
(Time data, ...)
RNN

Sparse data
(text, ...)
Embedding

Generative Adversarial Network
GAN

Basic Classification
DNN

Reinforcement learning

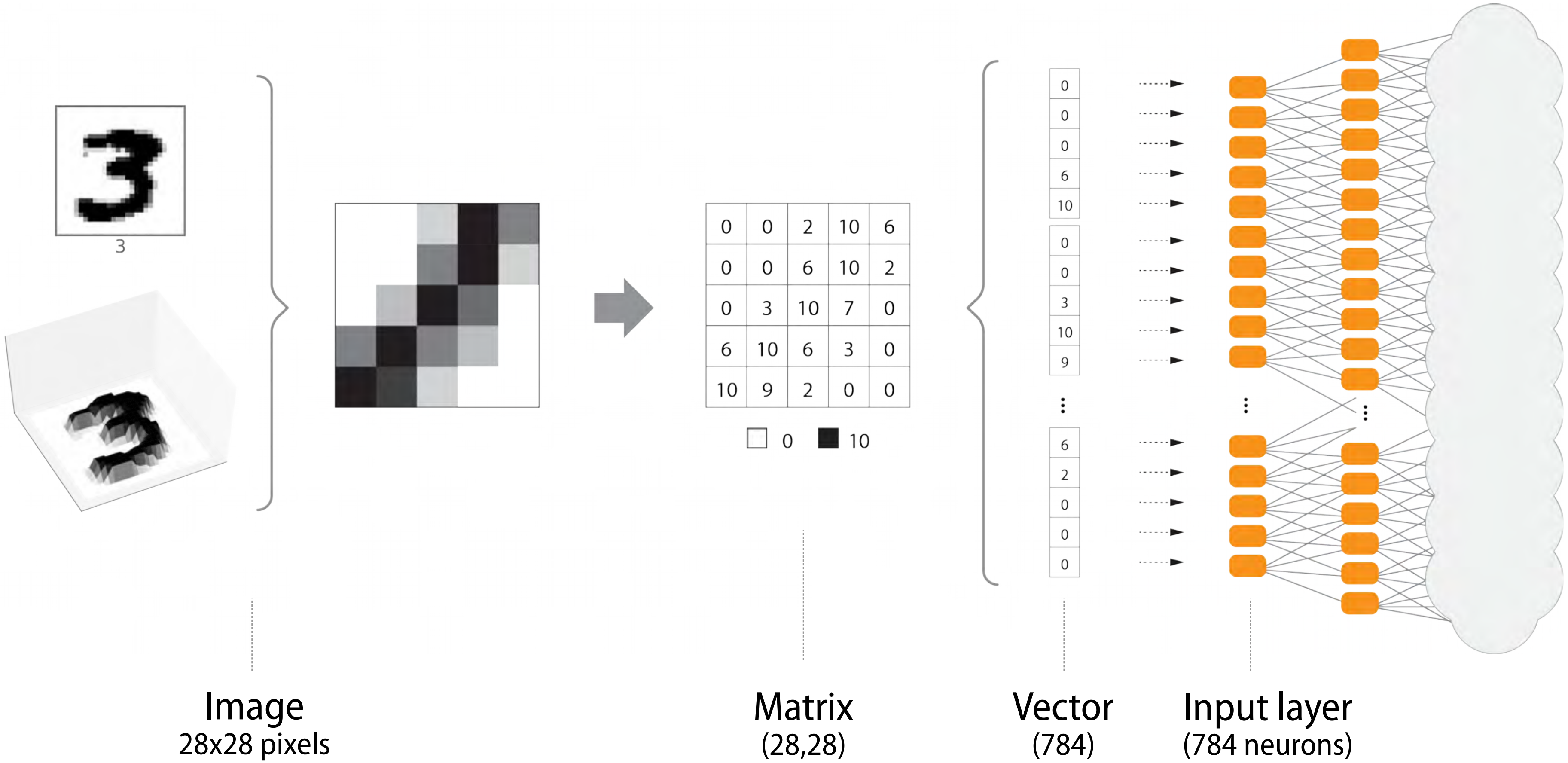
3/ Neurons & data

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

Sequences data
(Time data, ...)
RNN

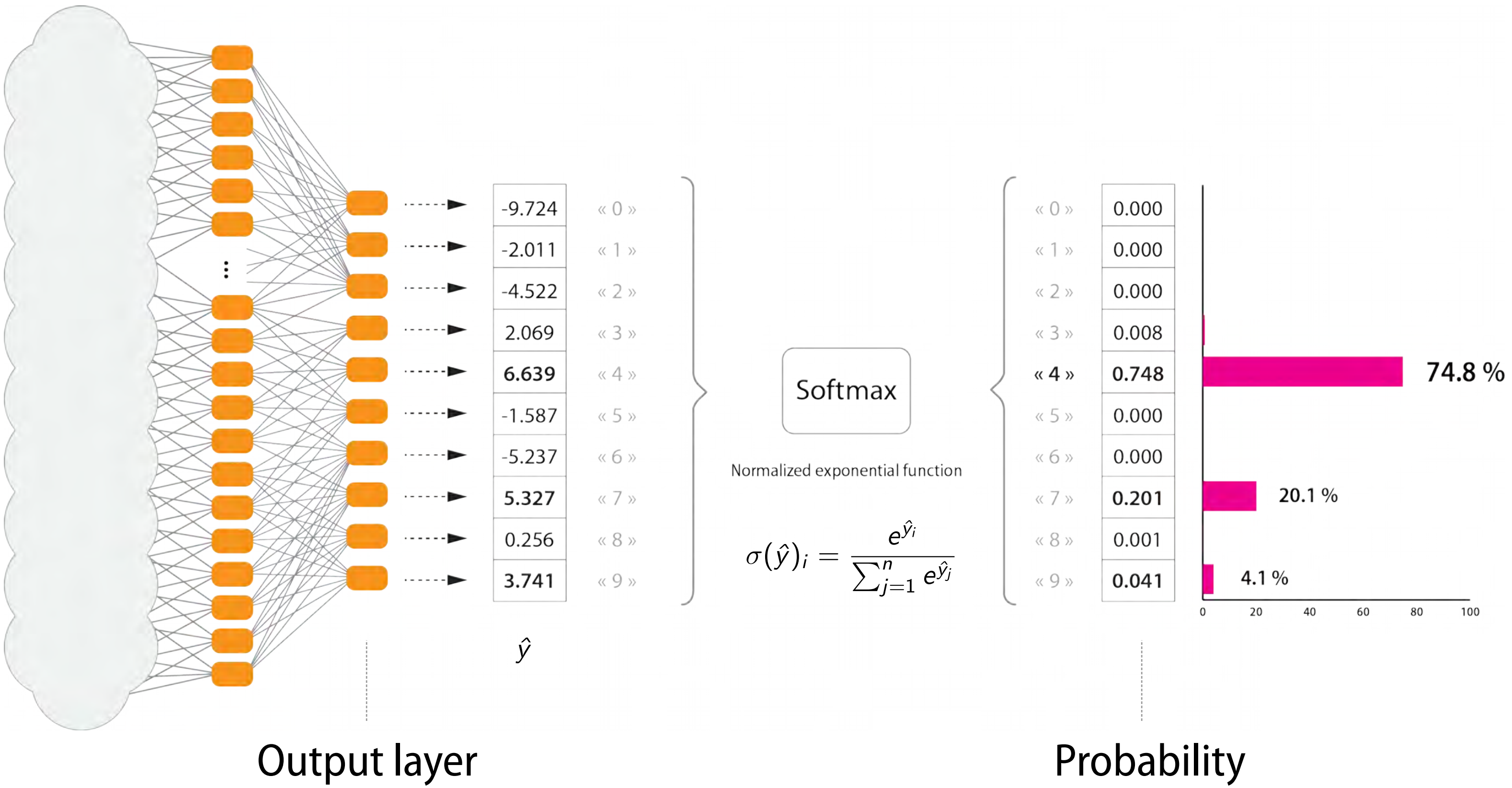
Sparse data
(text, . . .)
Embedding

Basic example / MNIST



Notebook [LAB14.1]

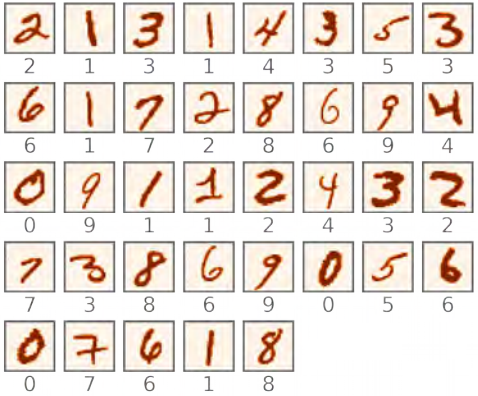
Basic example / MNIST



Notebook [LAB14.1]

Basic example
Handwritten Digits Recognition

MNIST dataset
Tensorflow, Jupyter lab



3/ Neurons & data

Generative Adversarial Network
GAN

Basic Classification
DNN

Reinforcement
learning

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

Sequences data
(Time data, ...)
RNN

Sparse data
(text, ...)
Embedding

Convolutional Neural Networks (CNN)

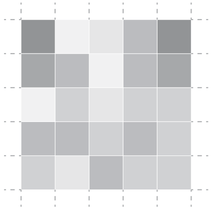
24 M pixels
(r,v,b) 3x8 bits

3 x 24 M neurons ?!

10 000	70 M
1 000 000	700 M
100 Mds	250 Mds

Convolutional Neural Networks (CNN)

"Straight ahead" by HarisDrako



5	2	1	3	5
4	3	2	3	4
0	2	1	2	2
3	3	2	3	2
2	1	3	2	2

Image piece

5	2	1
4	3	2
0	2	1

X

$\otimes$

Kernel 3x3

1	0	1
0	1	0
1	0	1

ω

=

10

y

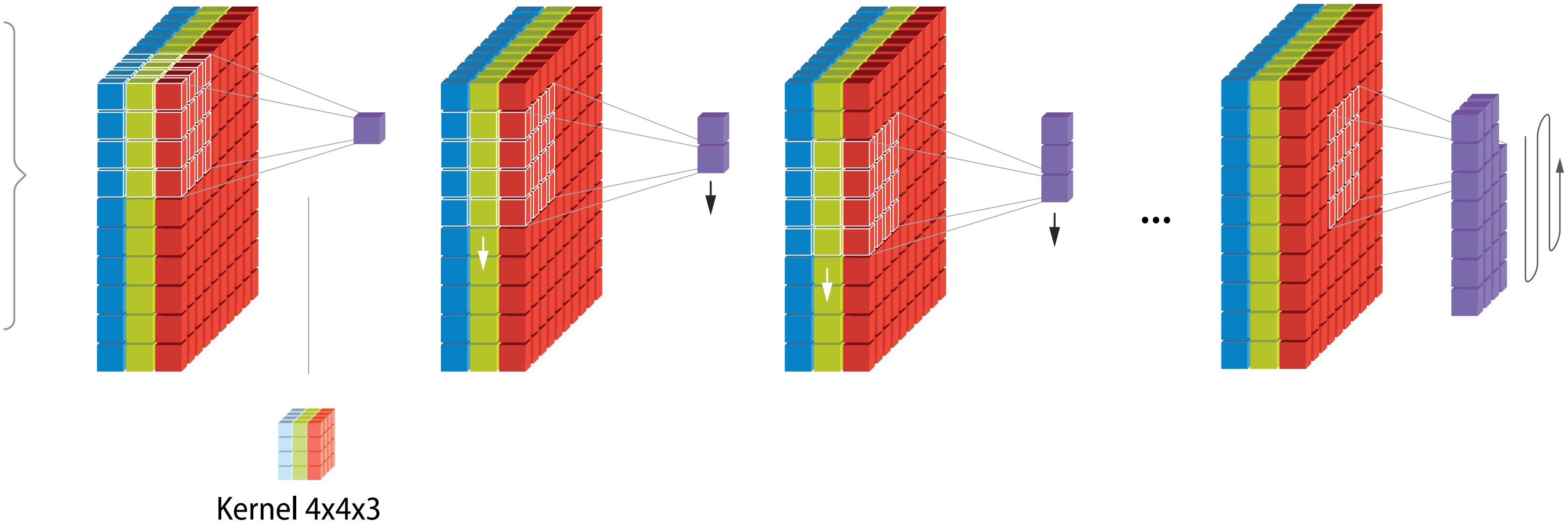
$$
\begin{aligned}
y &= 5\text{x}1 + 2\text{x}0 + 1\text{x}1 \\
&+ 4\text{x}0 + 3\text{x}1 + 2\text{x}0 \\
&+ 0\text{x}1 + 2\text{x}0 + 1\text{x}1 = 10
\end{aligned}
$$

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

2D convolution

Convolutional Neural Networks (CNN)

"Morondava - 28" by Olivier Lejade - CC BY-SA 2.0



3D convolution

Convolutional Neural Networks (CNN)

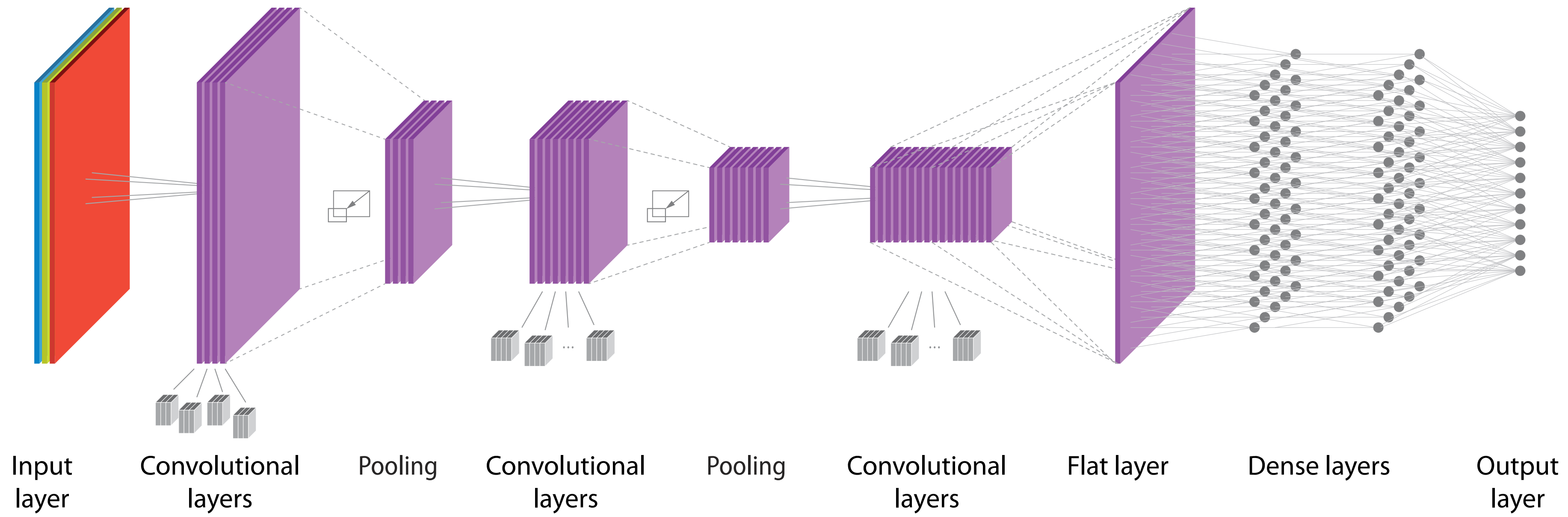


Image classification
with **MobileNet v1**

Trained model
TensorflowJS, Javascript

3/ Neurons & data

Generative Adversarial Network
GAN

Basic Classification
DNN

Reinforcement
learning

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

Sequences data
(Time data, ...)
RNN

Sparse data
(text, . . .)
Embedding

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

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Sparse matrix

$$\left.\begin{array}{lcl} \text{Dictionary} & = & 80\,000 \\ \text{Sentence} & = & 300 \end{array}\right\} \quad \text{Vectors} = 24\ \text{M}$$

Word Embedding

« movie»

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

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

« movie»

$$\begin{pmatrix} -2,03 \\ 15,12 \\ -13,08 \end{pmatrix}$$

Short dense vector based on context

T

word2vec[1]
- CBOW
- SG

GloVe[2]

...

ProtVec[3]

[1] 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.

[2] Jeffrey Pennington & all, (2014), [GLOVE]
Training is performed on aggregated global word-word co-occurrence statistics.

[3] Ehsaneddin Asgari, Mohammad R.K. Mofrad (2016), [PROTV]
Biological Sequences Representation

IMDB film review classification

Word Embedding
Keras, jupyter lab

87 %

3/ Neurons & data

Generative Adversarial Network
GAN

Basic Classification
DNN

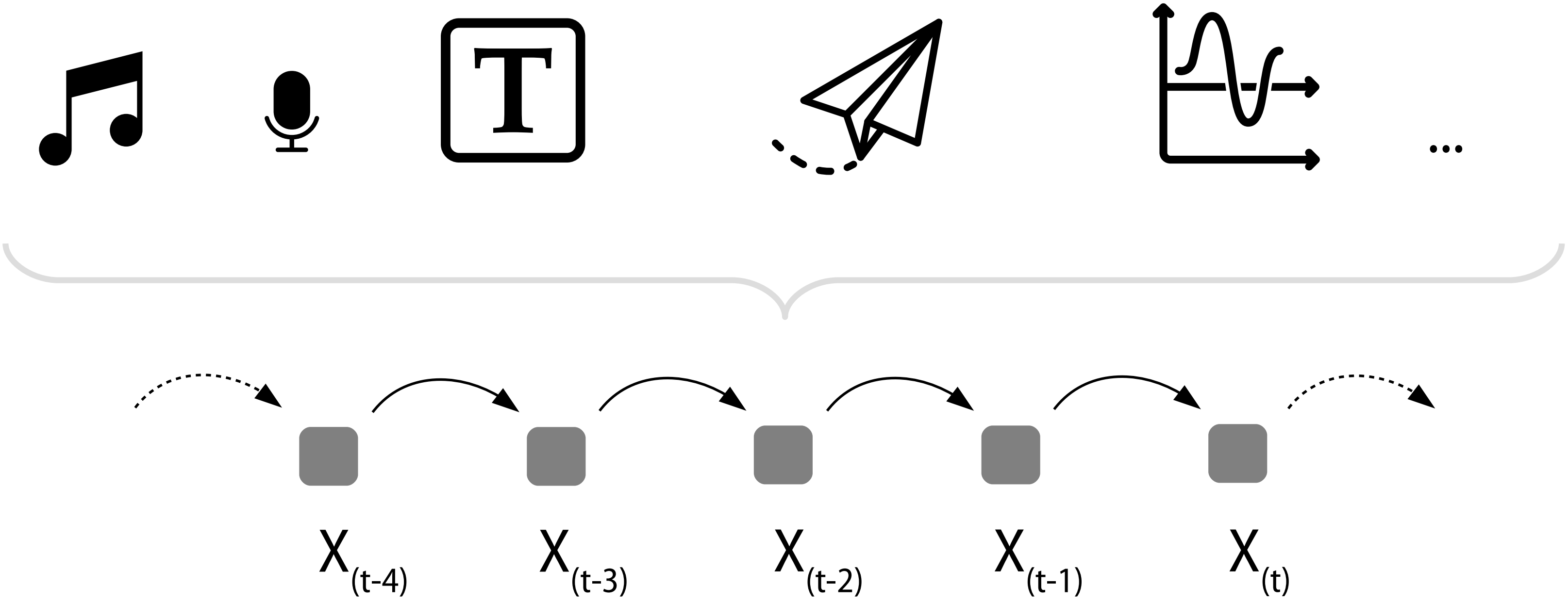
Reinforcement
learning

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

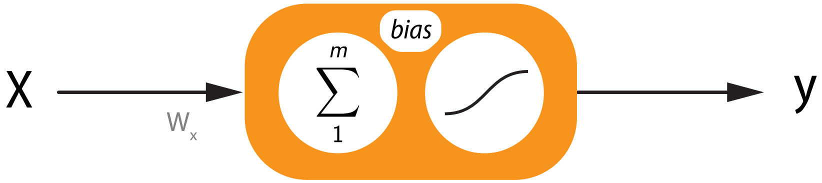
Sequences data
(Time data, ...)
RNN

Sparse data
(text, ...)
Embedding

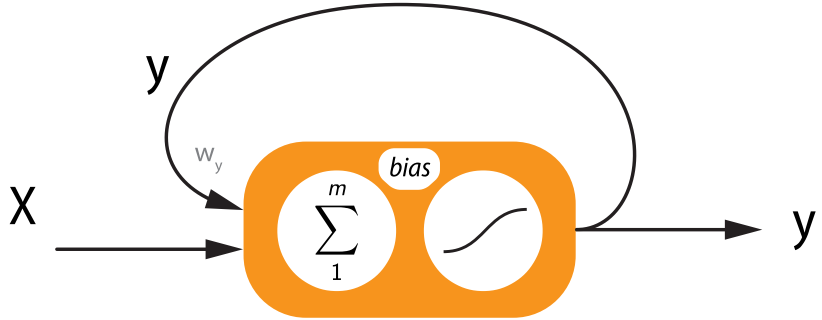
Reccurent Neural Network (RNN)



Reccurent Neural Network (RNN)

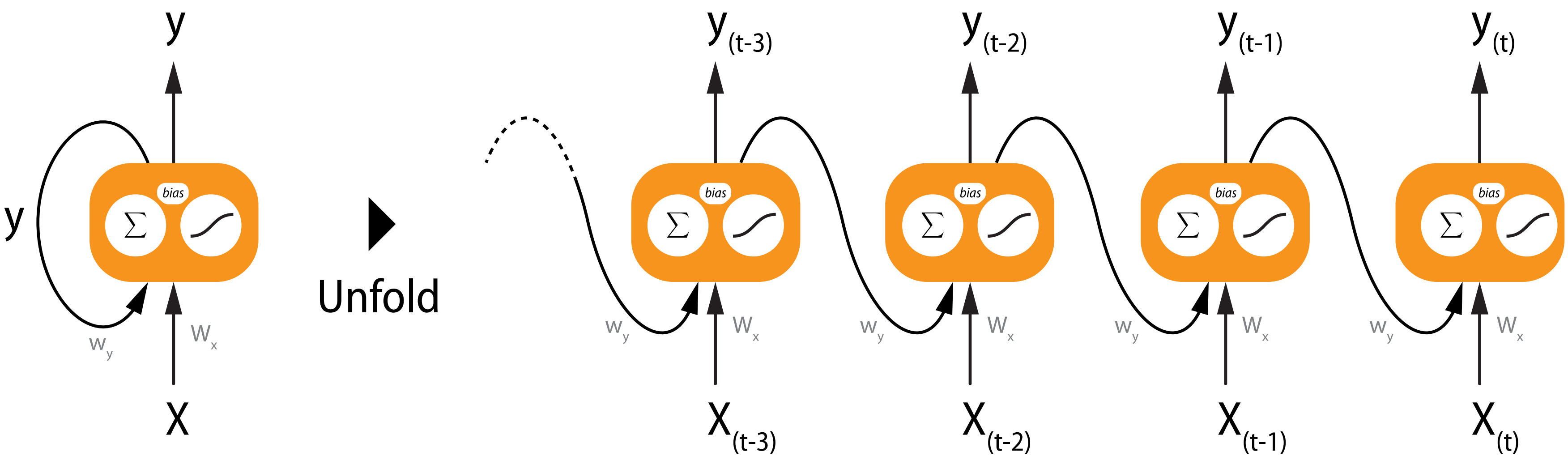


$$y = \sigma(W_x^T \cdot X + b)$$



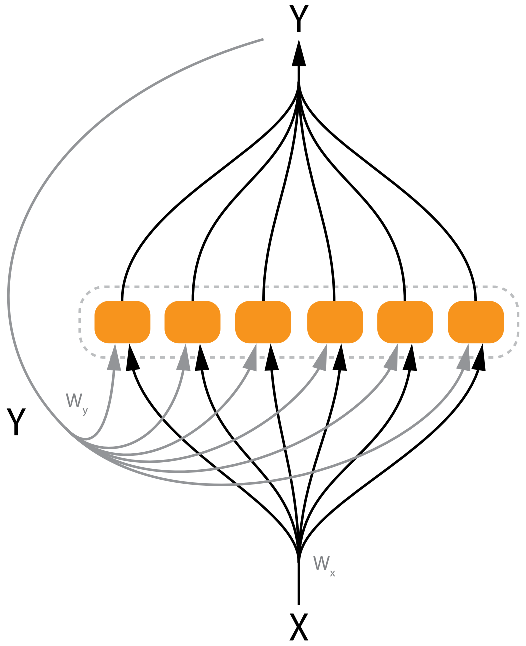
$$y_{(t)} = \sigma(W_x^T \cdot X_{(t)} + w_y \cdot y_{(t-1)} + b)$$

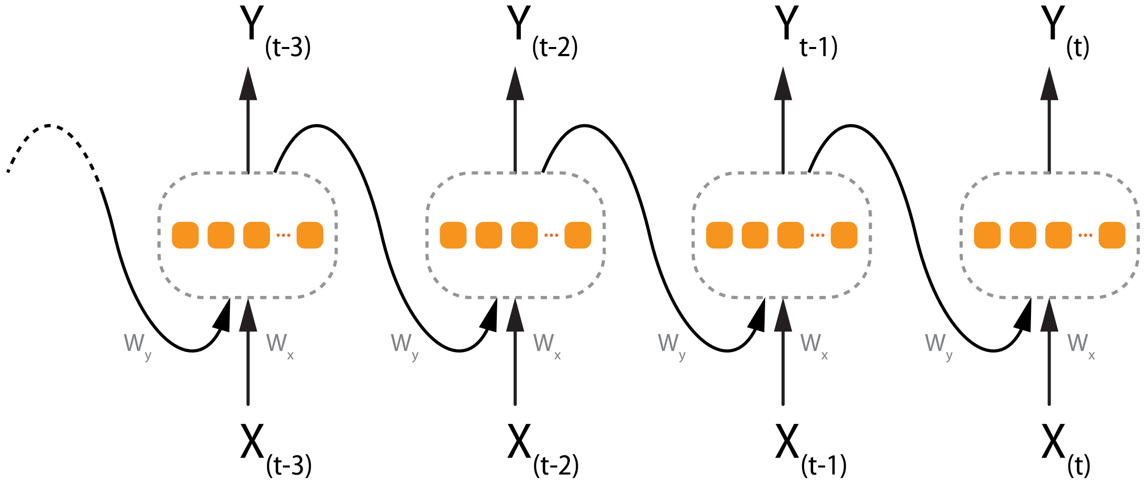
Reccurent Neural Network (RNN)



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

Reccurent Neural Network (RNN)



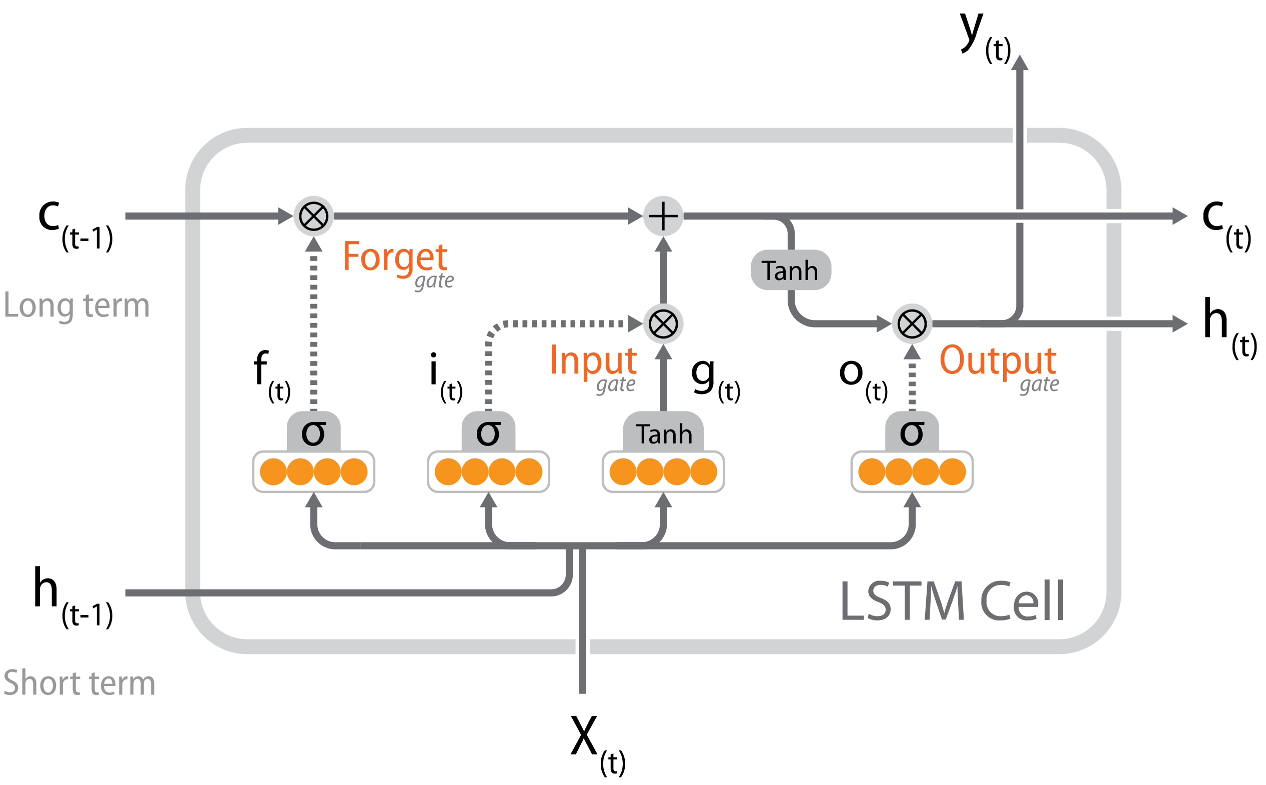


$$Y_{(t)} = \Phi\left(W_x^T \cdot X_{(t)} + W_y^T \cdot Y_{(t-1)} + b\right)$$

Recurrent neuron
Recurrent layer } « Cell »

Slow convergence,
Short memory,
Vanishing / exploding gradients

Reccurent Neural Network (RNN)



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

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
σ	sigmoid function
W_k	weights matrix
b_k	bias vector

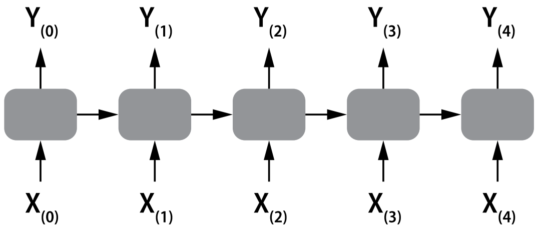
Long short-term memory (LSTM)[1]

Gated recurrent unit (GRU)[2]

[1] Sepp Hochreiter, Jürgen Schmidhuber, (1997) [LSTM]

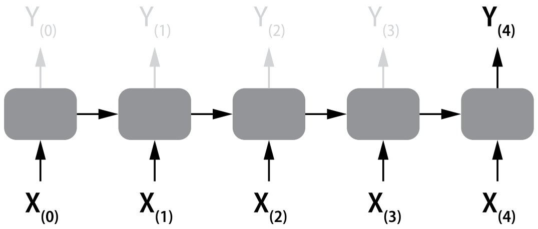
[2] Kyunghyun Cho et al, (2014) [GRU]

Reccurent Neural Network (RNN)



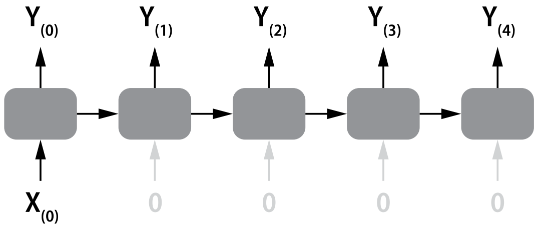
Serie to serie

Example : Time serie prediction



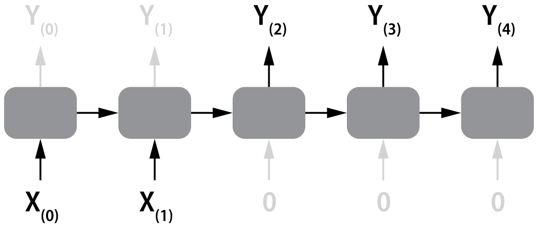
Serie to vector

Example : Sentiment analysis



Vector to serie

Example : Image annotation



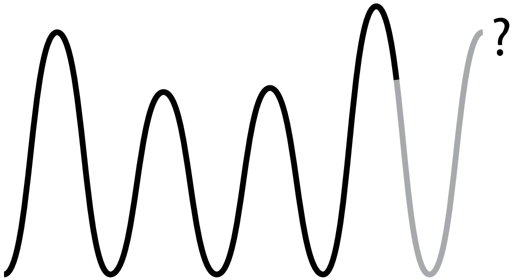
Encoder-decoder

Example : Language Translation

Reccurent Neural Network (RNN)

Time serie prediction

RNN with LSTM cell
Tensorflow, jupyter lab



3/ Neurons & data

Generative Adversarial Network
GAN

Basic Classification
DNN

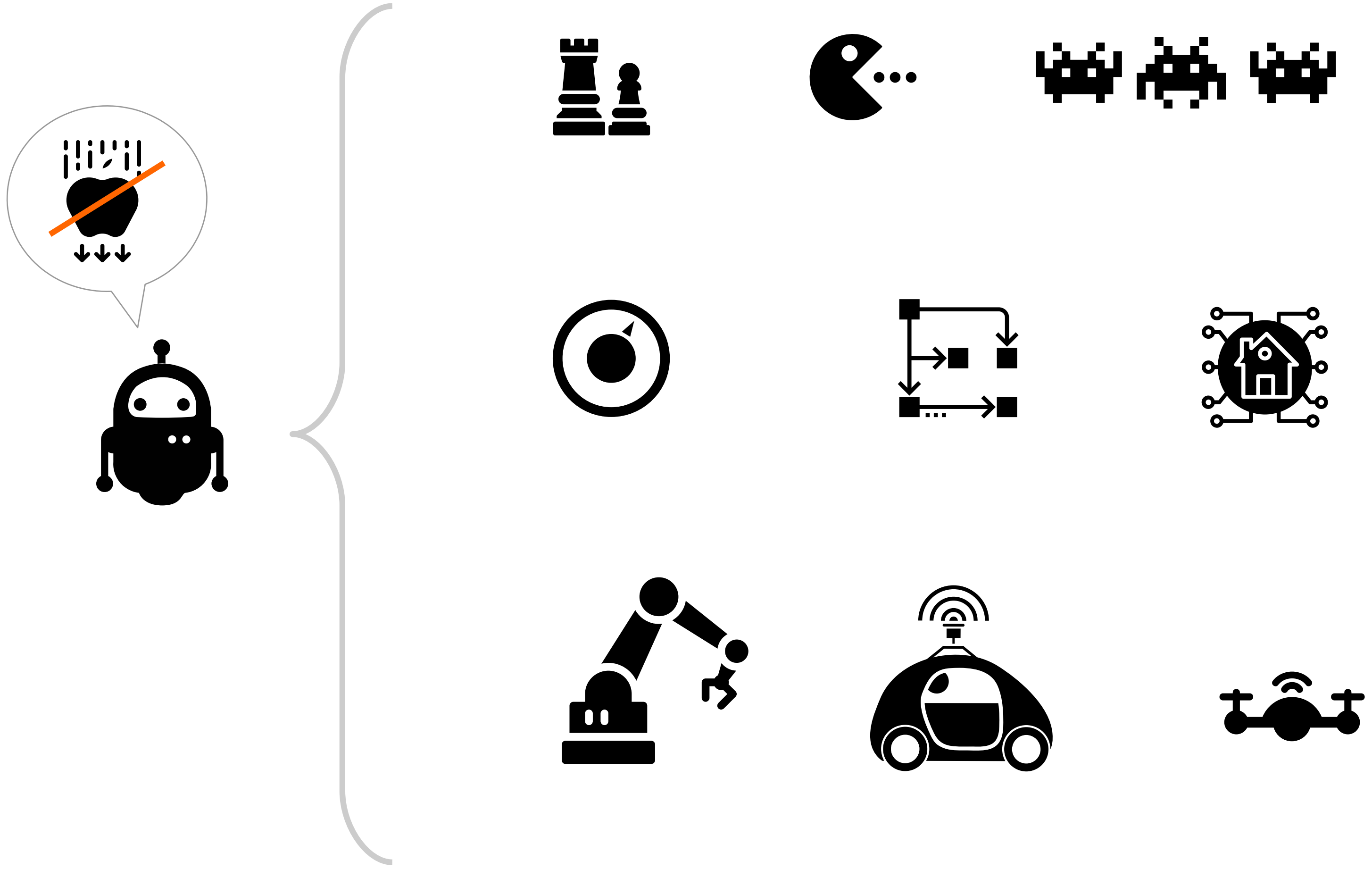
Reinforcement learning

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

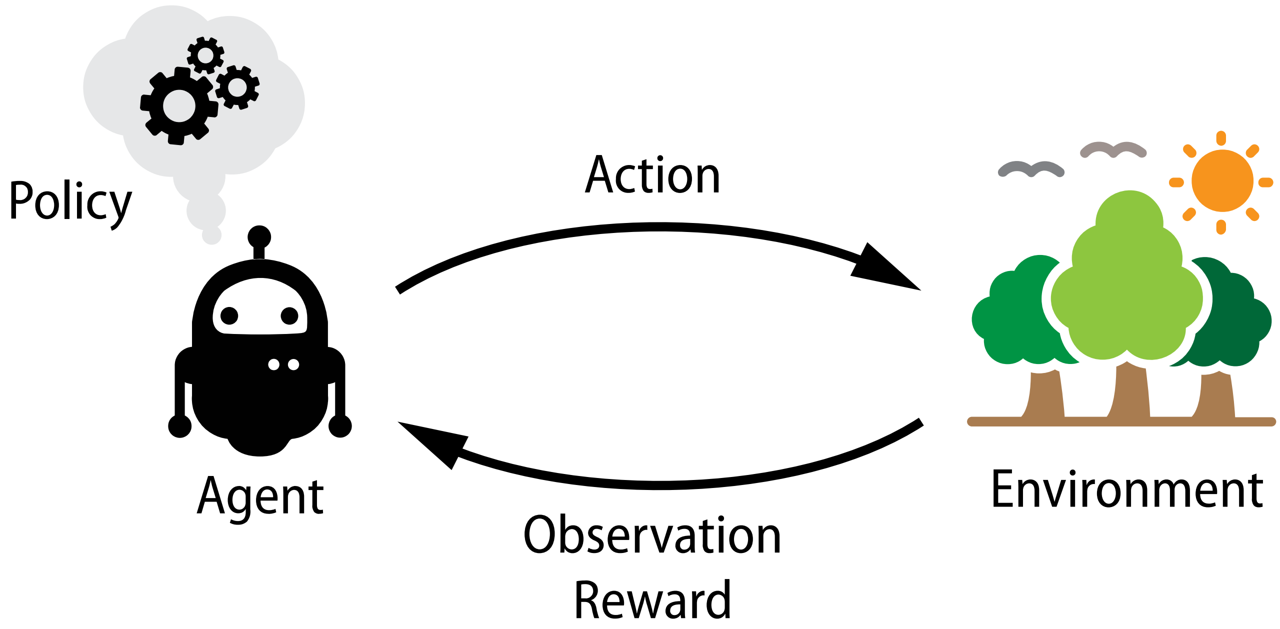
Sequences data
(Time data, ...)
RNN

Sparse data
(text, . . .)
Embedding

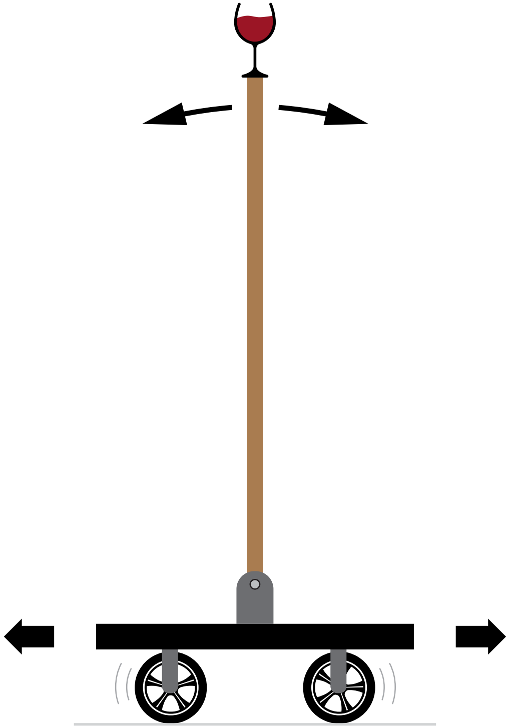
Reinforcement learning



Reinforcement learning



What actions can be taken to maximize rewards ?



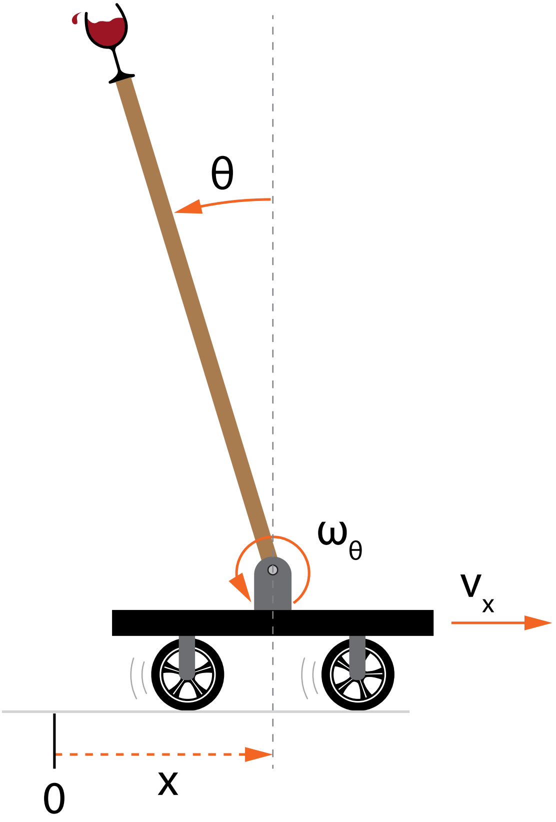
Inverted pendulum

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

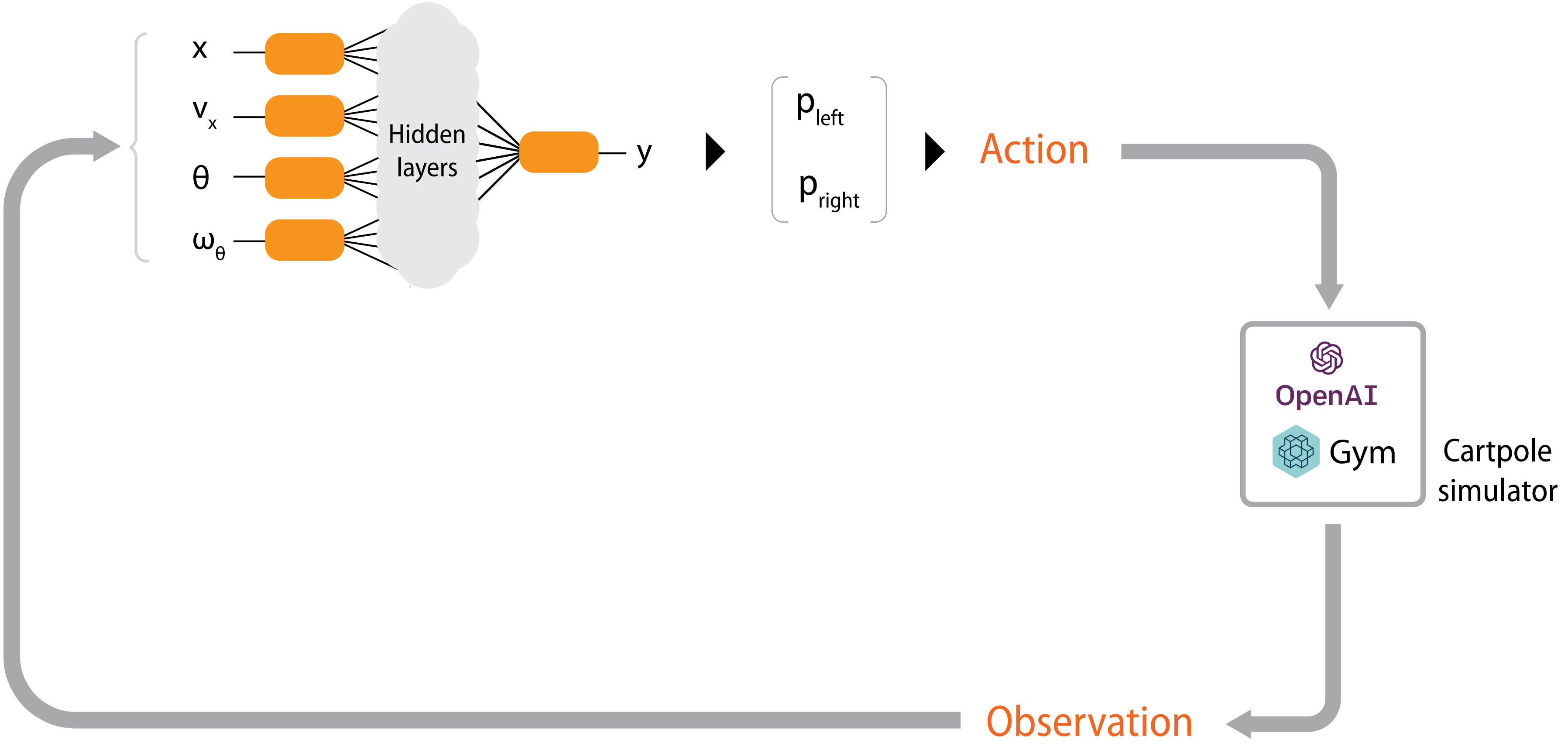
Observations :

x	Cart position
v_x	Cart velocity
Θ	Pole angle
ω_Θ	Pole angular velocity

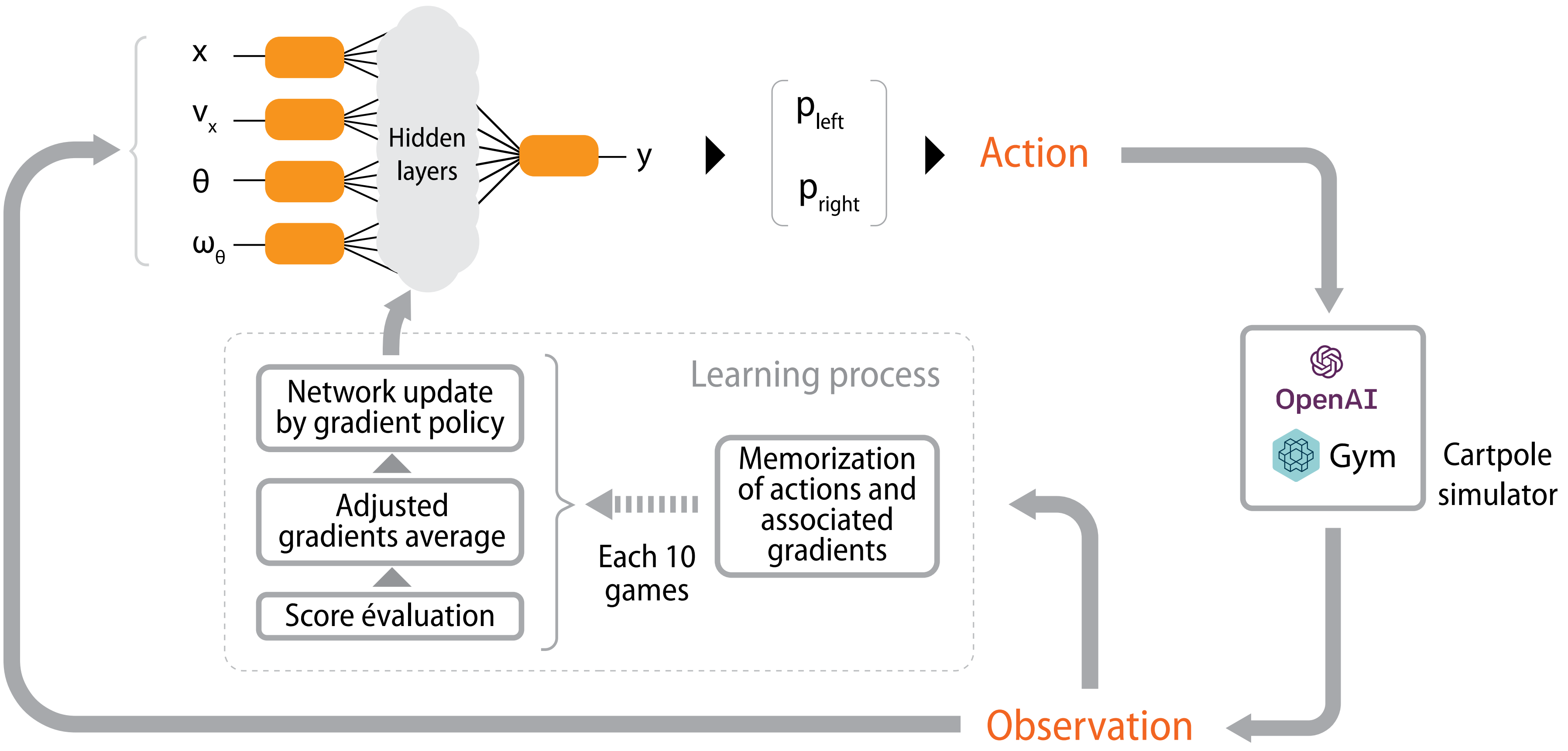
Rewards :

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

Reinforcement learning

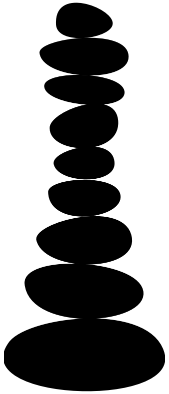


Reinforcement learning



Reinforcement learning

OpenAI/Gym Cartpole with gradient policy



3/ Neurons & data

Generative Adversarial Network
GAN

Basic Classification
DNN

Reinforcement
learning

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

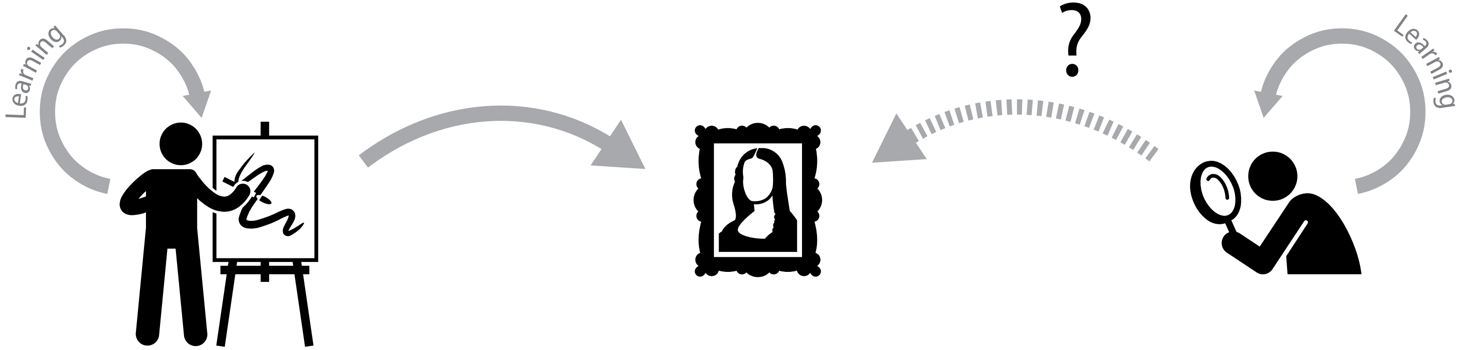
Sequences data
(Time data, ...)
RNN

Sparse data
(text, ...)
Embedding

Generative Adversarial Network

GAN[1] Use Cases :

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

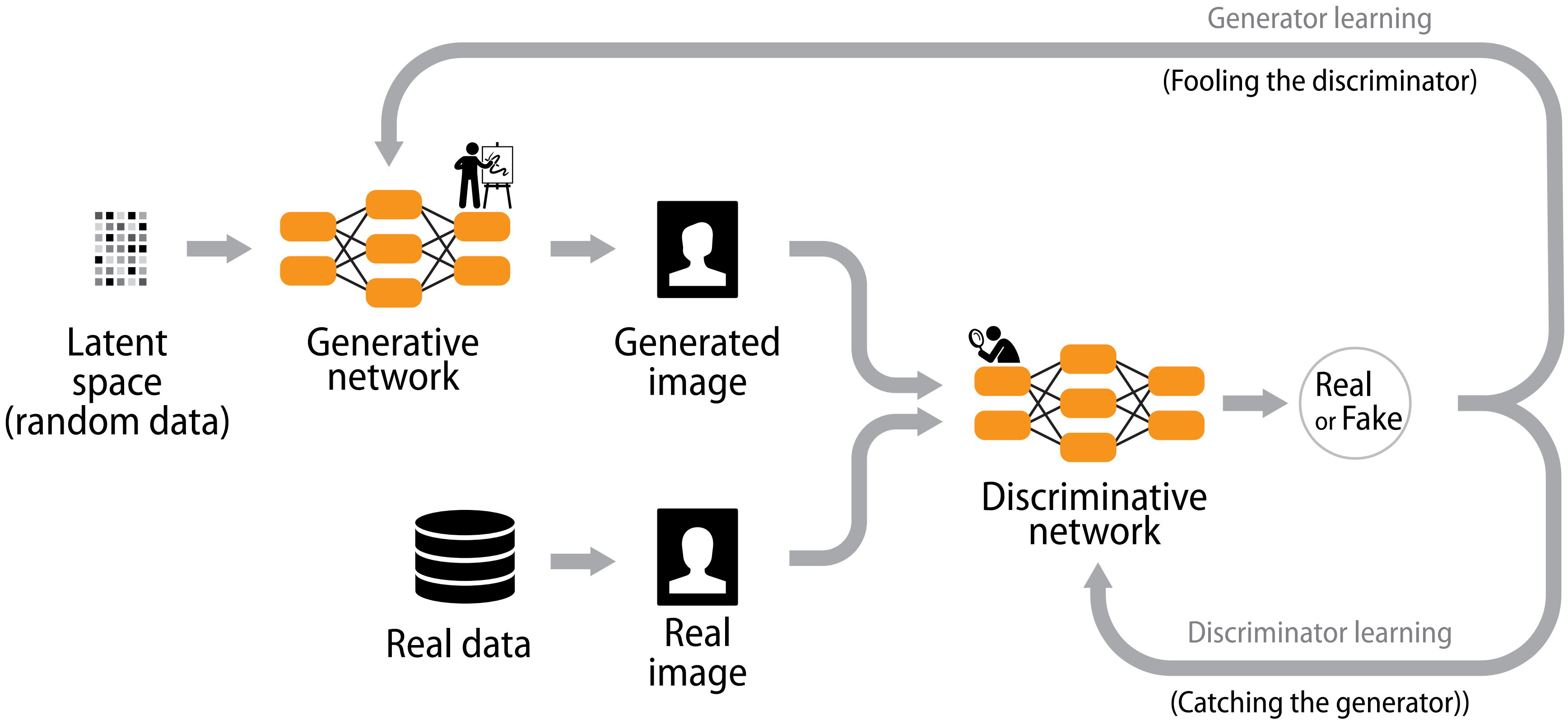


Counterfeiter
(Generator)

Expert
(Discriminator)

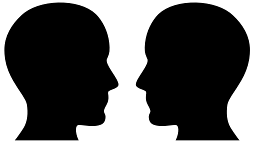
[1] Ian J. Goodfellow & all, (2014), « Generative Adversarial Networks » [GAN]

Generative Adversarial Network



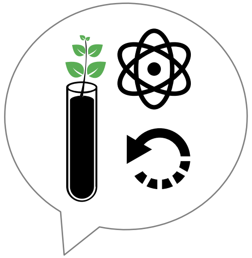
Generative Adversarial Network

Photorealistic generation



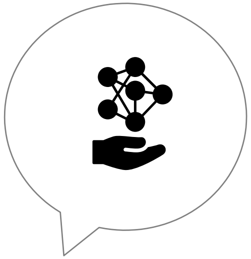
4/ Conclusion

Conclusion



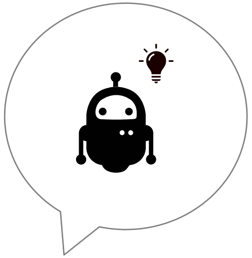
Great opportunities

it works !



Complex but accessible tools and techniques

Open — Science / Data / Source

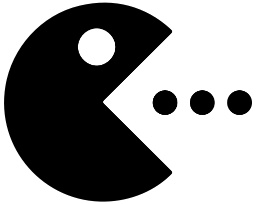


Very significant and rapid progress

AlphaFold won 13th Critical Assessment of Structure Prediction (CASP)
Prediction of the 3D structure of proteins from a given amino acid sequence as input.

« (…) Due to our concerns about malicious applications of the technology, we are not releasing the trained model.(…) »
OpenAI.com

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



Major societal impacts

« San Francisco Bans Facial Recognition Technology »
New York Times
May 14, 2019

COMMENT PERMETTRE À L'HOMME DE GARDER LA MAIN[1] ?

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 CONFIÉE PAR LA LOI POUR UNE RÉPUBLIQUE NUMÉRIQUE

[1] Report available on the CNIL website

Références

[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

Références

[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 Merrienboer, 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

[CNIL] 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

Notebooks

[LAB1] 01 Regression Linéaire.ipynb

[LAB2] 02 Descente de gradient.ipynb

[LAB12] 12 Regression Logistique.ipynb

[LAB1] Regression linéaire
Exemple de régression linéaire avec résolution directe

[LAB2] Gradient descent
Simple gradient descent example

[LAB12] Logistic Regression
Logistic Regression with Gradient Descent using TensorFlow

[LAB12.1] Activation functions
Example of activation functions

[LAB13] Simple Perceptron
IRIS classification with a simple perceptron, using sklearn

[LAB14.1] Deep Neural Network*
MNIST Example with Tensor Flow

[WEB1] Image classification with MobileNet v1*
Image classification with MobileNet using tensorflow js

[WEB2] Object detection with coco-ssd*
Object detection with coco-ssd/mobilenet using tensorflow js

Notebooks

[LAB22.2] Word Embedding – Basic
TripAdvisor CBOW Embedding with Gensim

[LAB22.3] Word Embedding – IMDB*
IMDB film review classification with Keras

[LAB21.3] Time series prediction with RNN*
Prediction of a time serie with LSTM RNN using Tensofflow

[LAB19.5] CartPole with Policy gradients*
CartPole game (from Gym) with gradient policy using Tensorflow

Illustrations

Illustrations from Wikimedia Commons, the free media repository.

"Morondava - 28" by Olivier Lejade is licensed under CC BY-SA 2.0

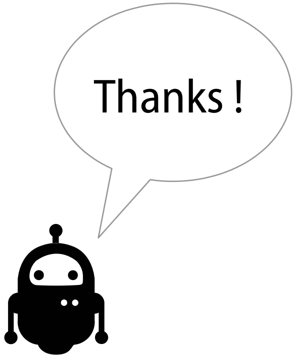
"straight ahead" by HarisDrako is licensed under CC BY-NC-ND 3.0

GitLab

https://gricad-gitlab.univ-grenoble-alpes.fr/talks/deeplearning

binder

https://bit.ly/2wDS3r6


Thanks !