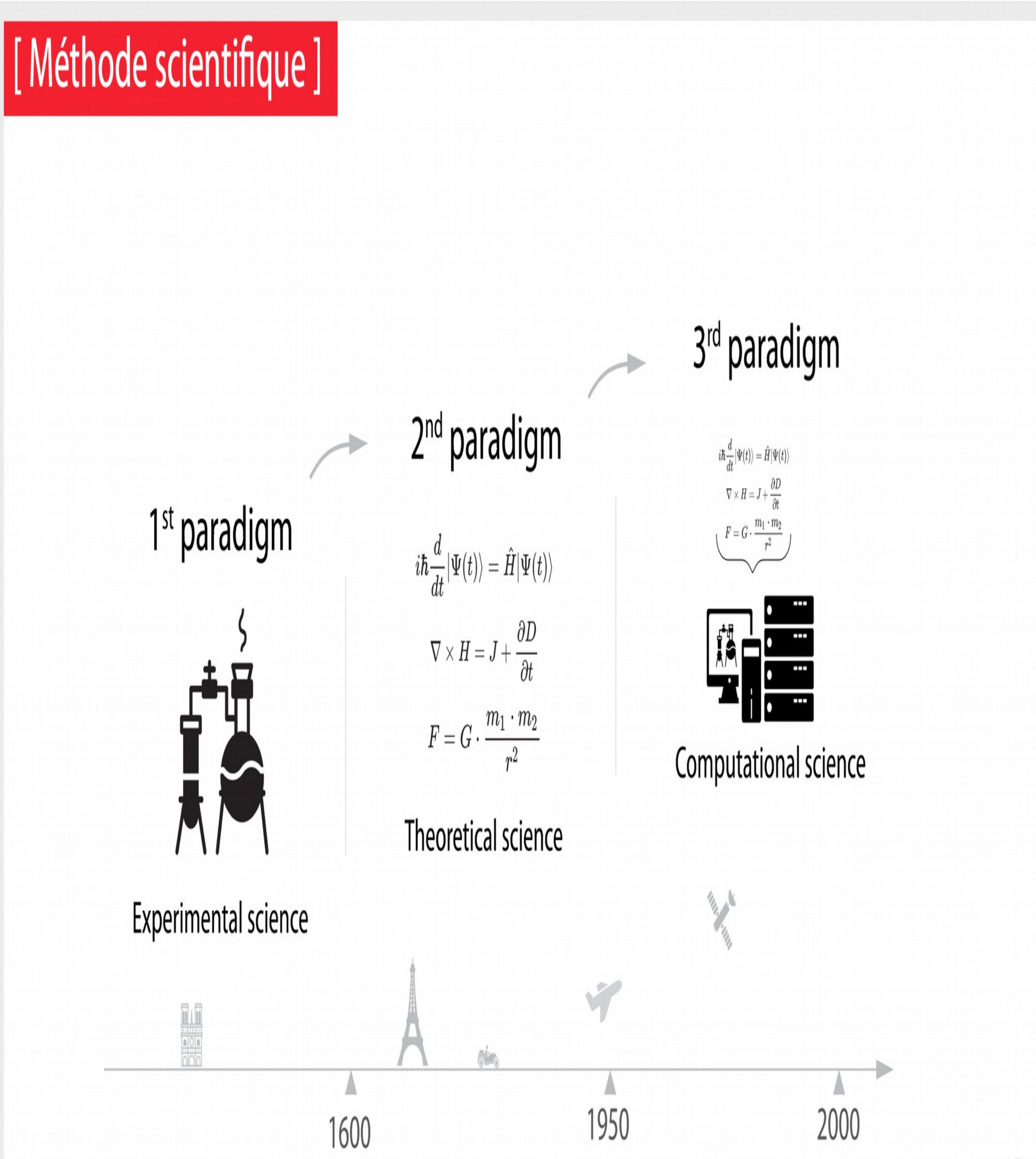
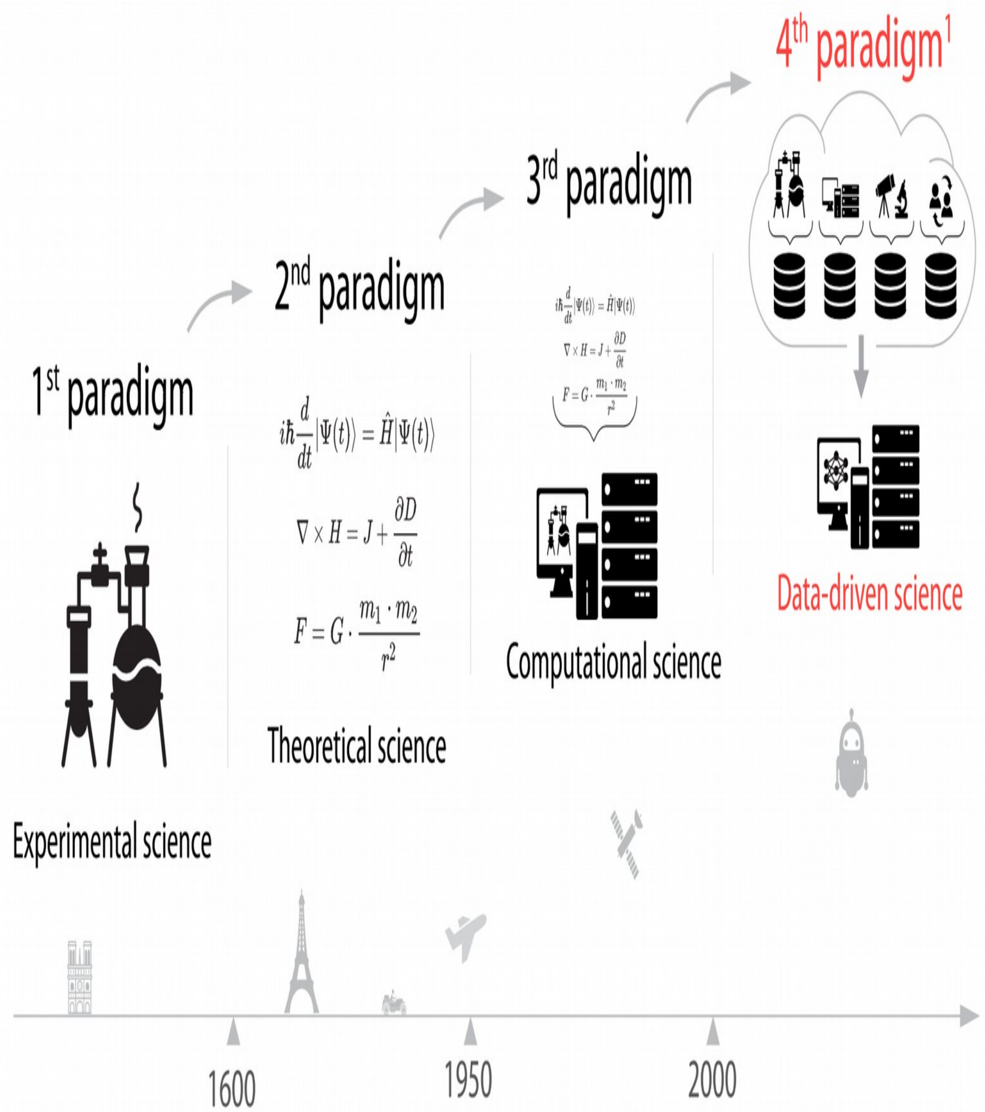
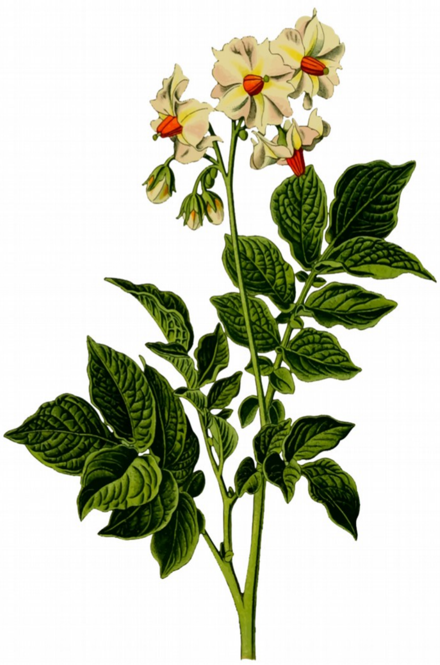
Introduction



[Méthode scientifique]



Intelligence, Intelligence artificielle ...Qu'est ce ? ….

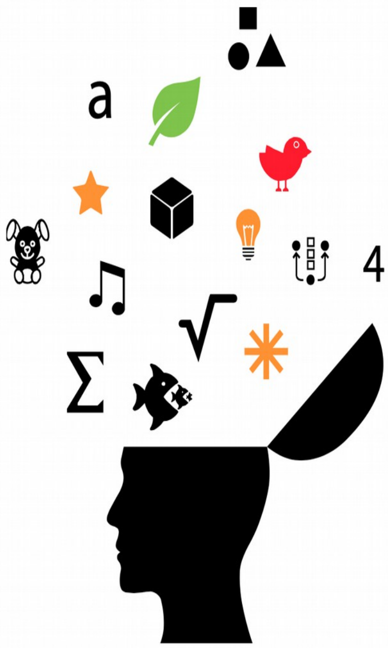


[intelligence]

« 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 appli towards adaptive behaviors within an environment or context »**

* Wikipedia - Illustration : [POTATO]



[intelligence]

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

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

* https://www.larousse.fr

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 :

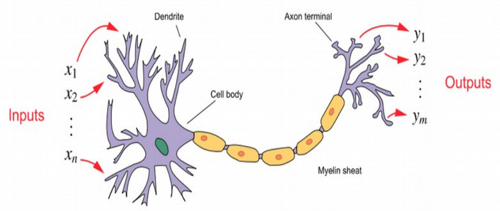
« Penser, c'est calculer des **symboles** qui ont à la fois une réalité matérielle et une valeur sémantique de représentation »[1]

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

Connectionnism vs **Symbolic**

Modelling the brain
Modéliser le cerveau

Making a mind
Forger une opinion

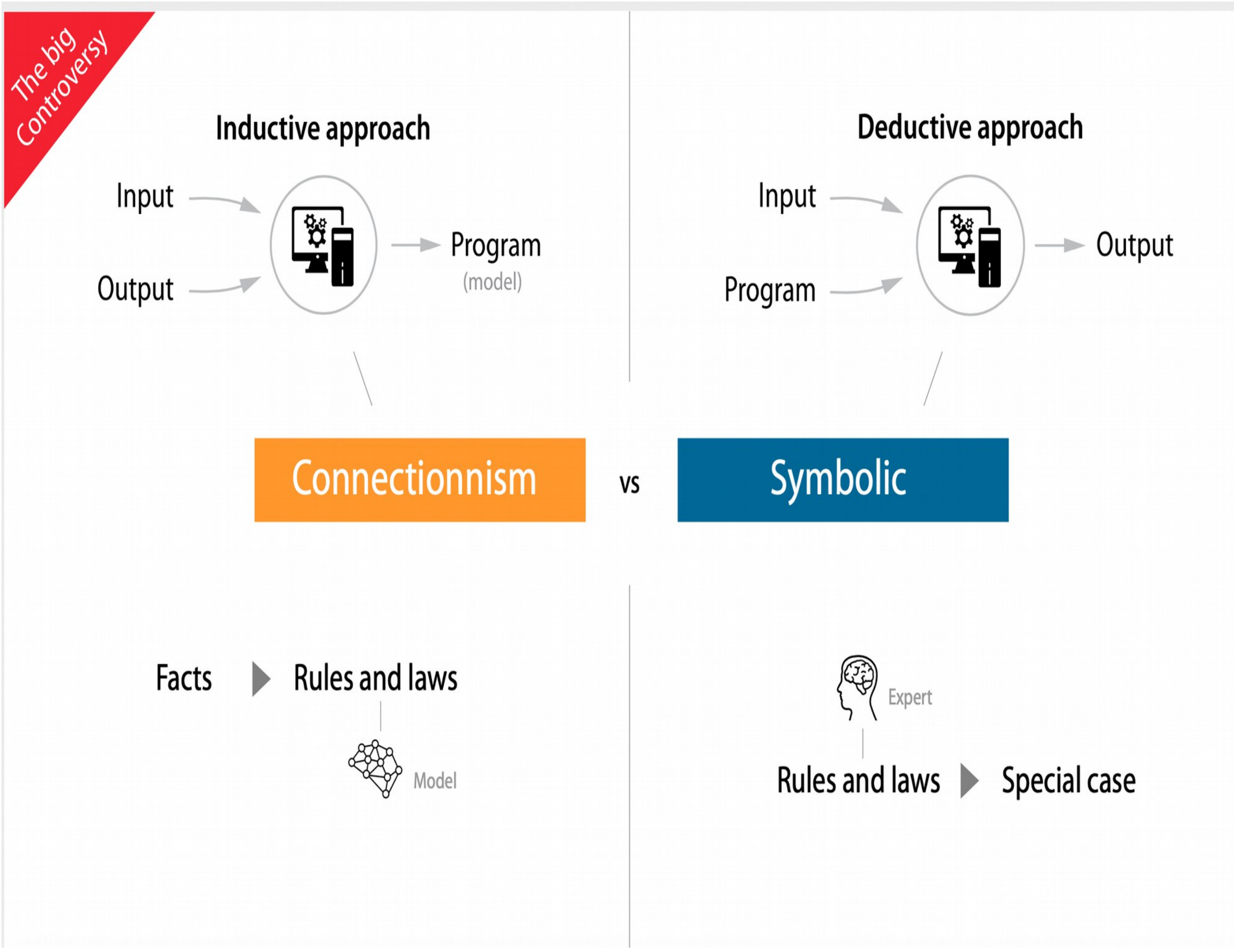


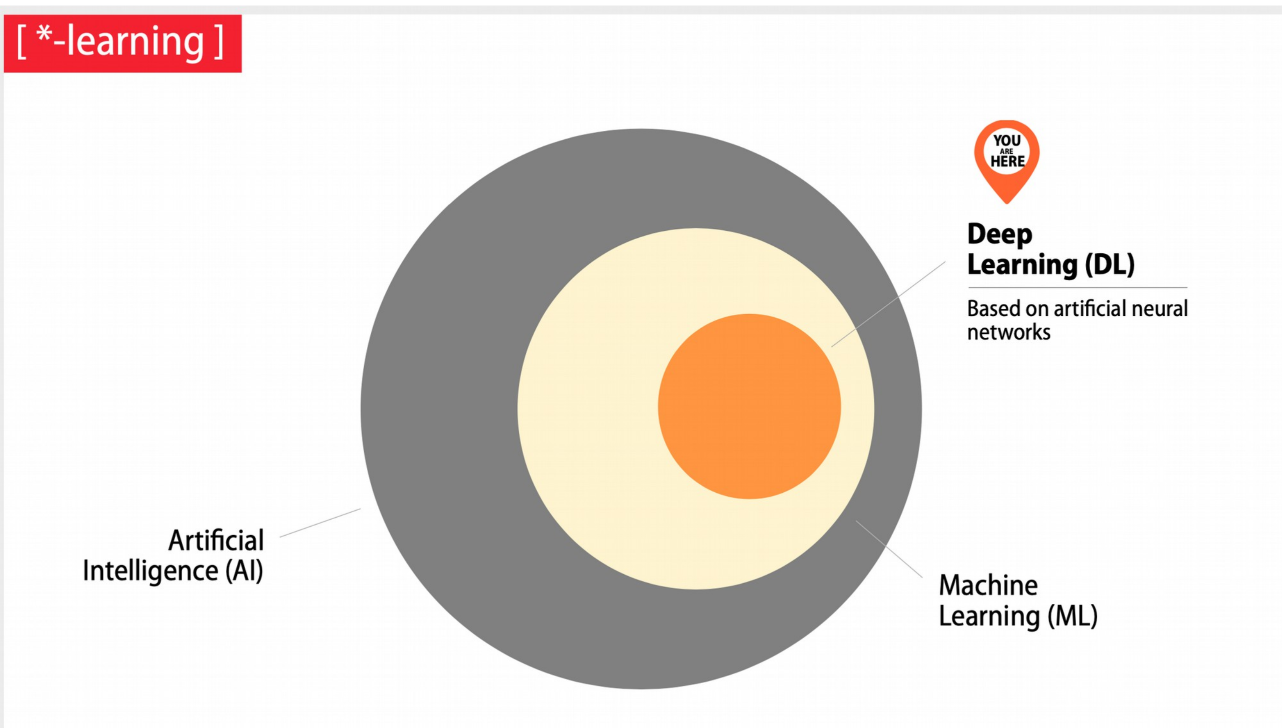

```
Tout [homme] est [mortel]
[Socrate] est un [homme]

Donc [Socrate] est [mortel]
```

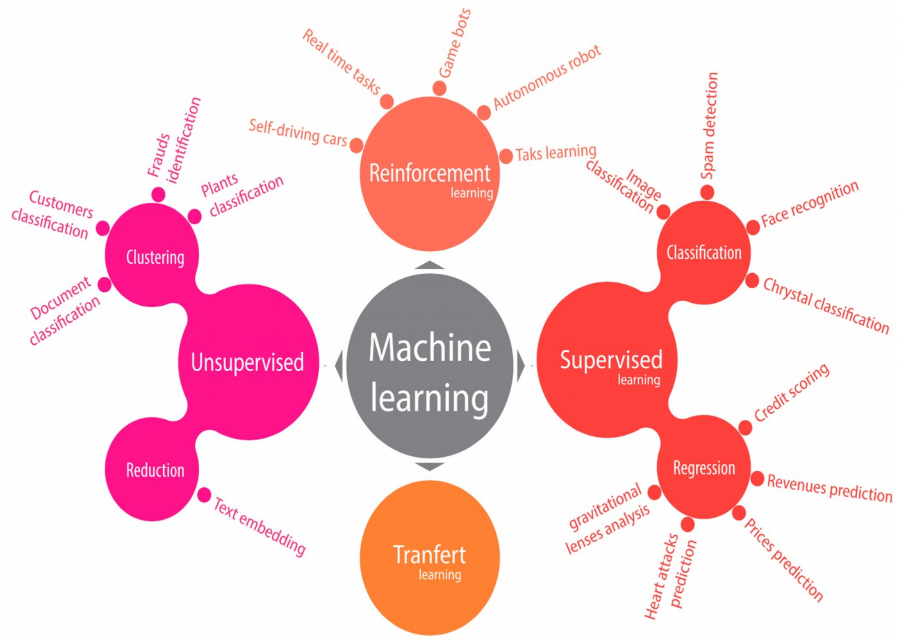
[1] D Cardon, JP Cointet, A Mazieres, 2018 [LRDN]

Illustration : [NEURON]


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

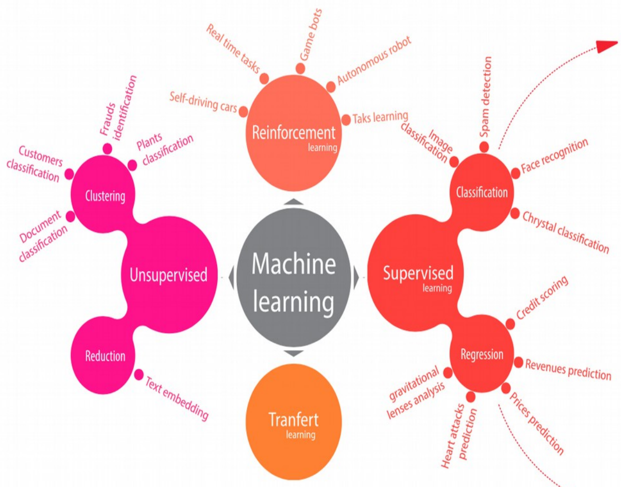

[*-learning]
YOU ARE HERE
Deep Learning (DL)
Based on artificial neural networks
Artificial Intelligence (AI)
Machine Learning (ML)

[*-learning]

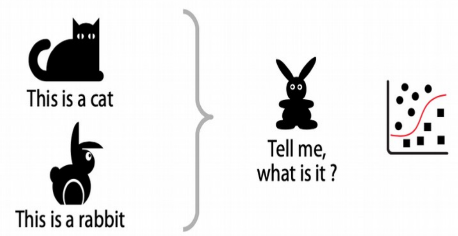


Supervised learning

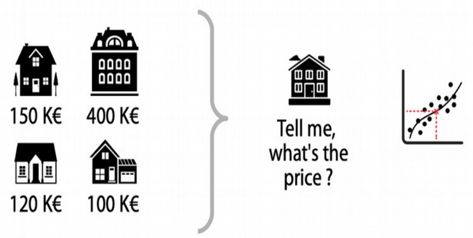
Learning from examples

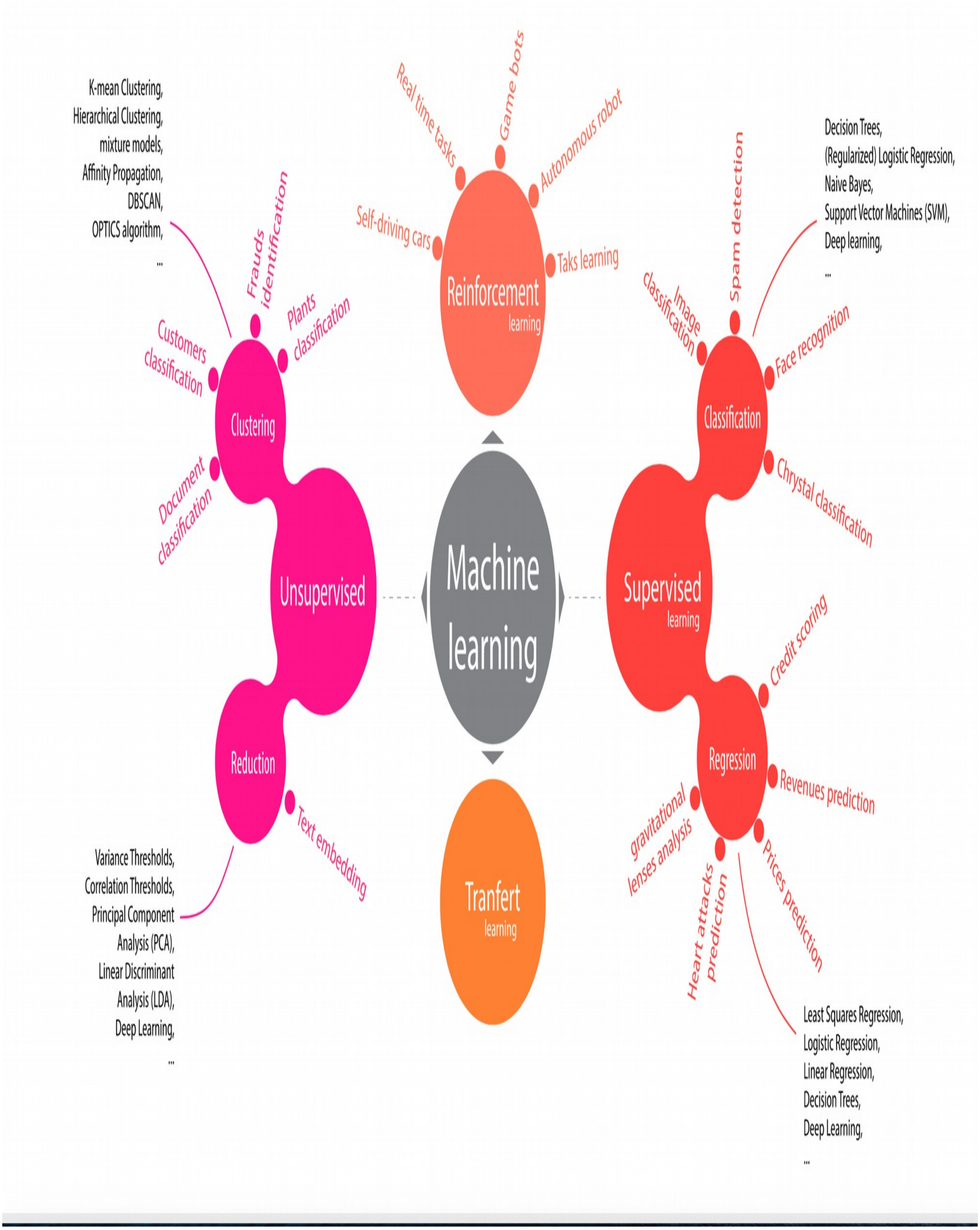


Classification :
Predict qualitative informations

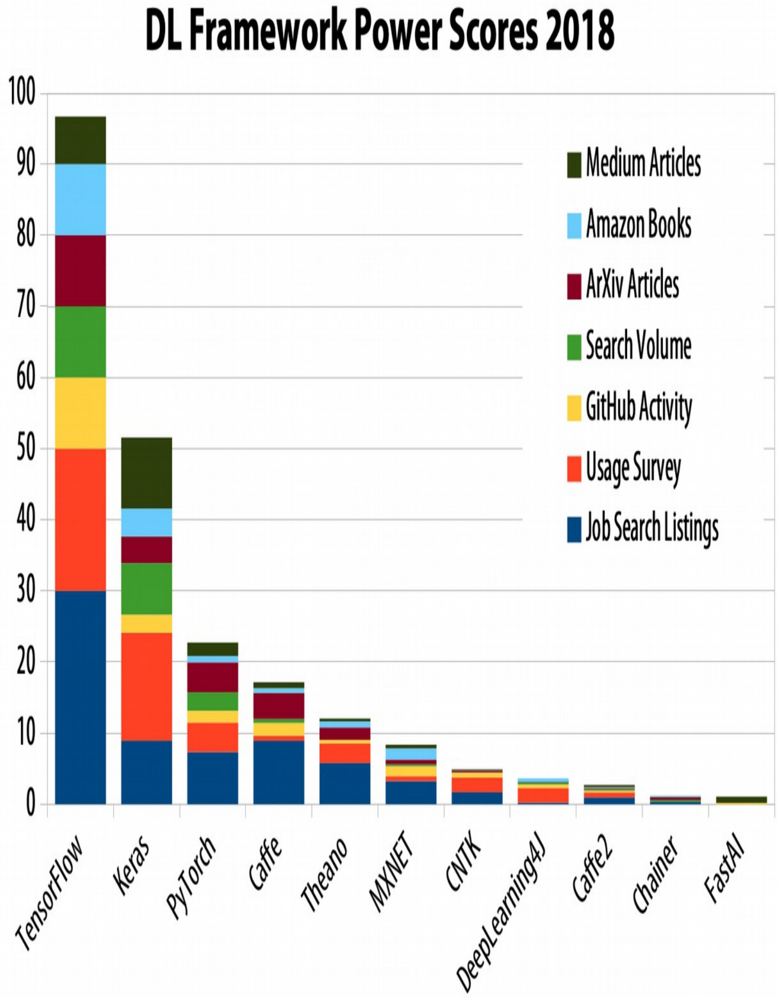


Régression :
Predict quantitative informations




Machine learning
Reinforcement learning
Real time tasks
Game bots
Autonomous robot
Self-driving cars
Taks learning
Unsupervised
Clustering
K-mean Clustering,
Hierarchical Clustering,
mixture models,
Affinity Propagation,
DBSCAN,
OPTICS algorithm,
...
Frauds identification
Plants classification
Customers classification
Document classification
Reduction
Text embedding
Variance Thresholds,
Correlation Thresholds,
Principal Component Analysis (PCA),
Linear Discriminant Analysis (LDA),
Deep Learning,
...
Supervised learning
Classification
Decision Trees,
(Regularized) Logistic Regression,
Naive Bayes,
Support Vector Machines (SVM),
Deep learning,
...
Spam detection
Image classification
Face recognition
Chrystal classification
Regression
Credit scoring
Revenues prediction
gravitational lenses analysis
Heart attacks prediction
Prices prediction
Least Squares Regression,
Logistic Regression,
Linear Regression,
Decision Trees,
Deep Learning,
...
Tranfert learning

A Python centered world



TensorFlow

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

Keras

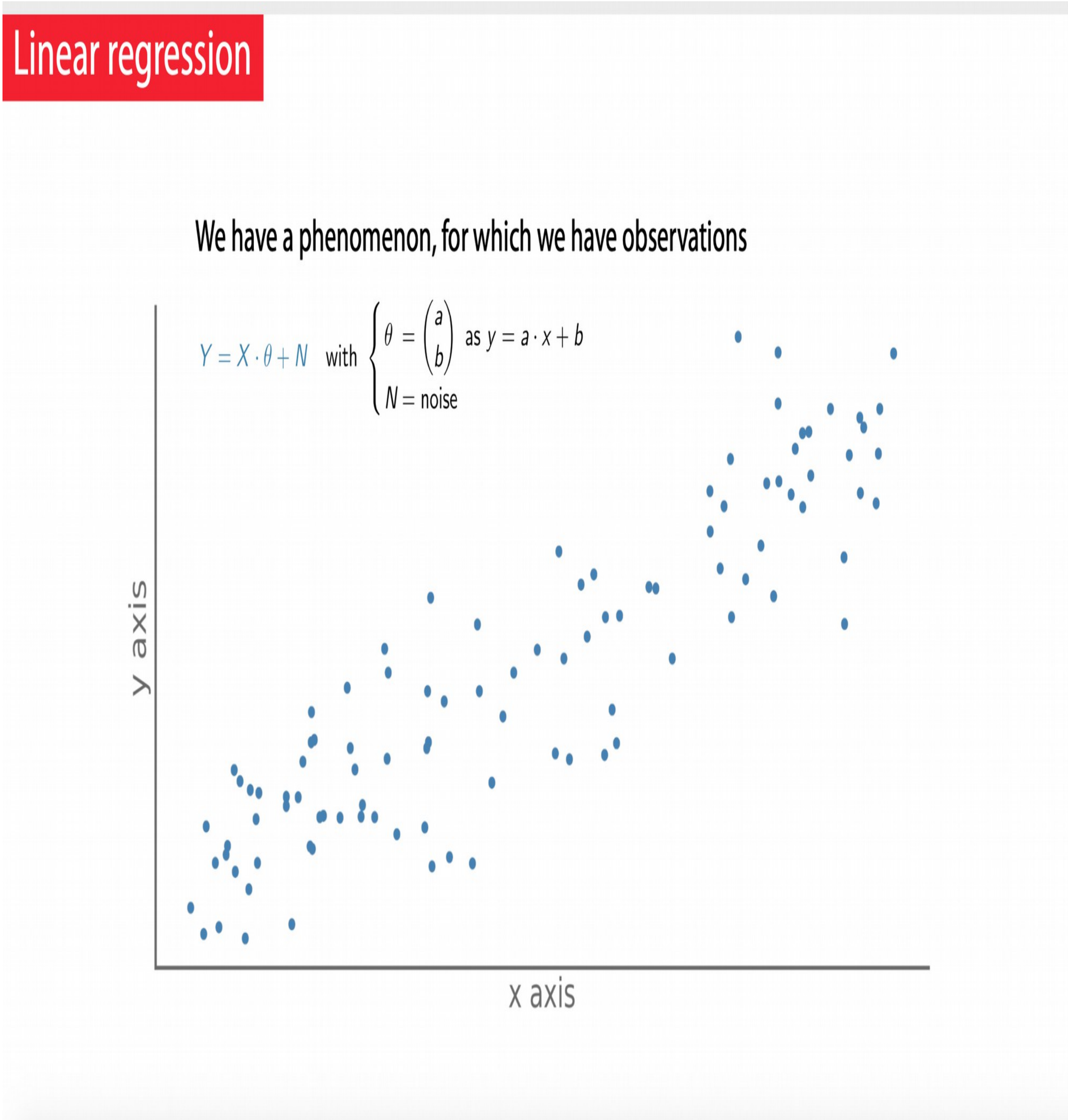
By François Cholet (Google)
Hight level API
Part on TensorFlow since 2017
MIT licence

PYTORCH

From Torch library
Supported by Facebook
BSD licence

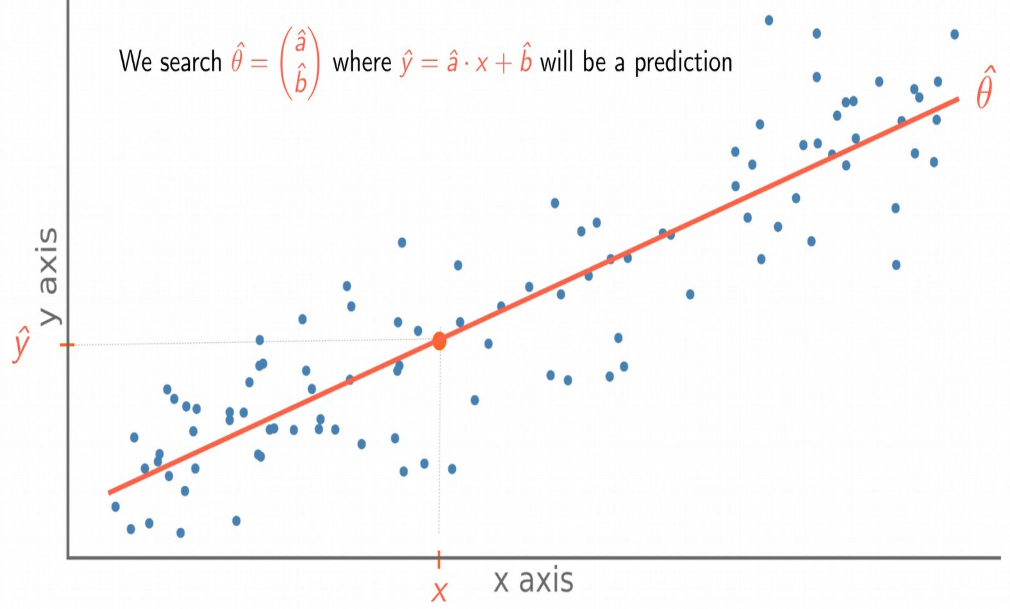
Source : Jeff Hale, « Deep Learning Framework Power Scores 2018 » [DLPW]

Introduction à l'apprentissage machine
L'exemple de la régression linéaire



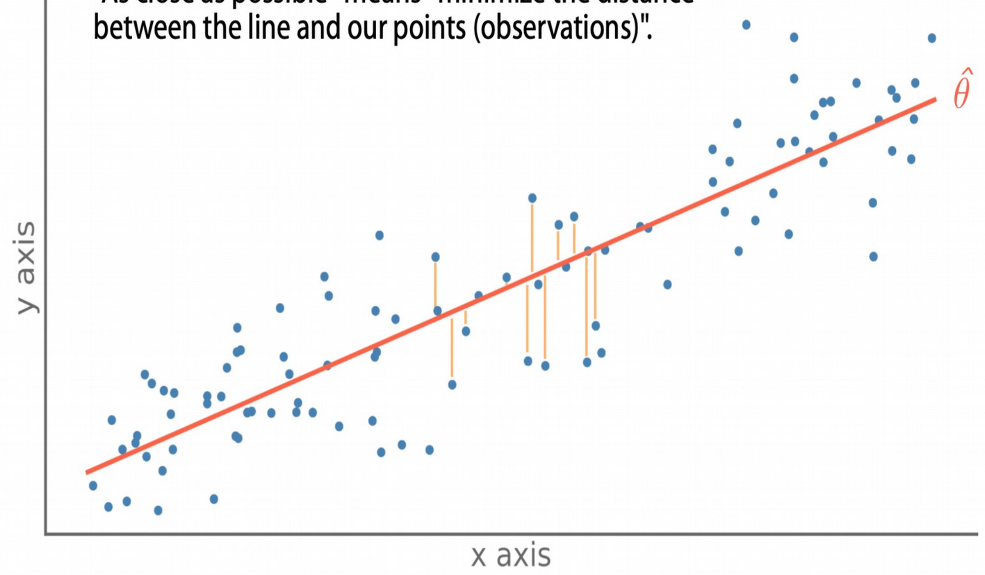
Linear regression

We are looking for a straight line that passes
« as close as possible » to our points.



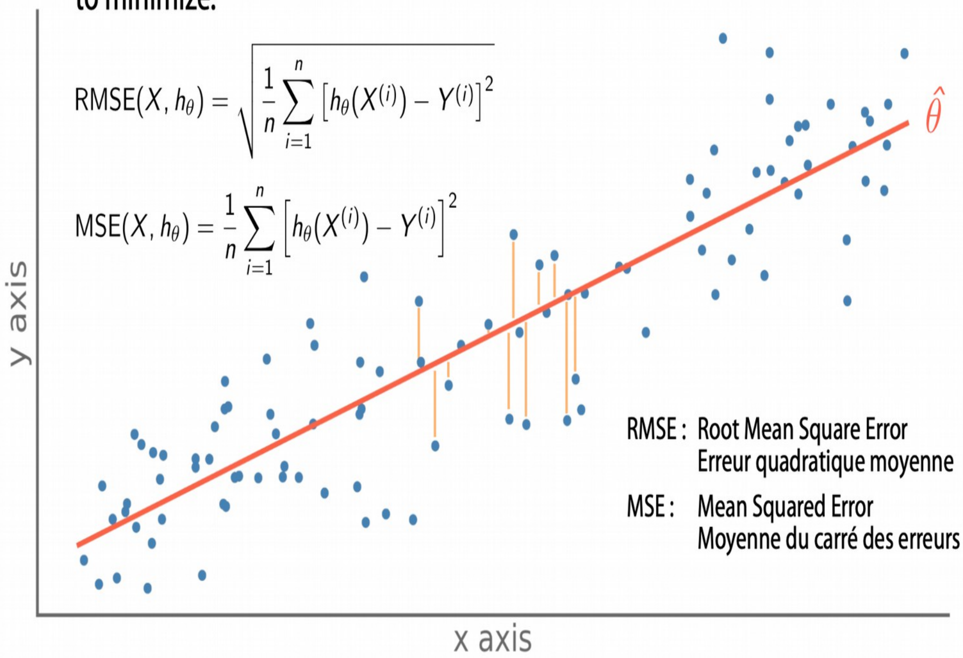
Linear regression

"As close as possible" means "minimize the distance between the line and our points (observations)".



Linear regression

For this, we will use an «loss function », which we will try to minimize.



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

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

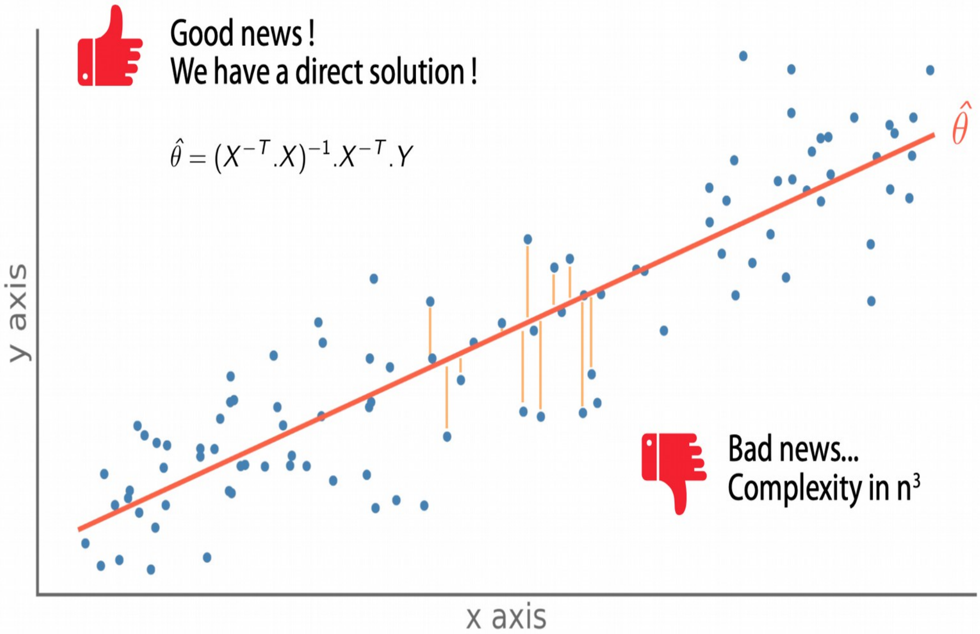
RMSE : Root Mean Square Error
Erreur quadratique moyenne

MSE : Mean Squared Error
Moyenne du carré des erreurs

Linear regression

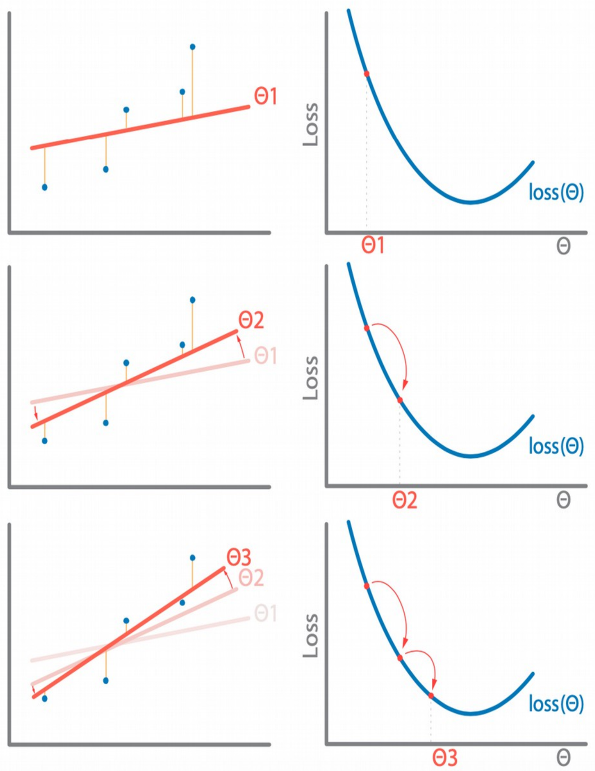
Good news !
We have a direct solution !

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



Bad news...
Complexity in n^3

Gradient descent



We will iteratively look for the best position of our line, by varying its parameters (Θ).

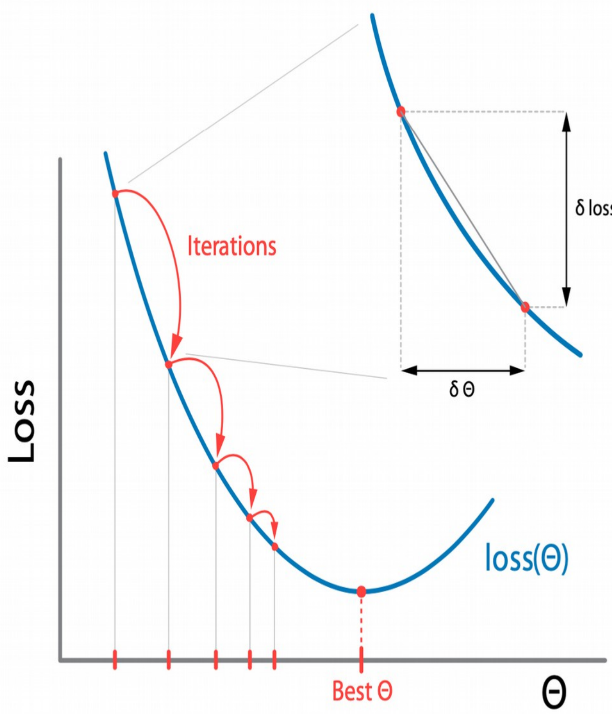
But how can we efficiently vary our parameters (Θ)?

Note : Loss functions could be :

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

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

Gradient descent



By changing Θ from δΘ
We improve loss(Θ) of δloss

The gradient is the slope we will follow to minimize our loss function.

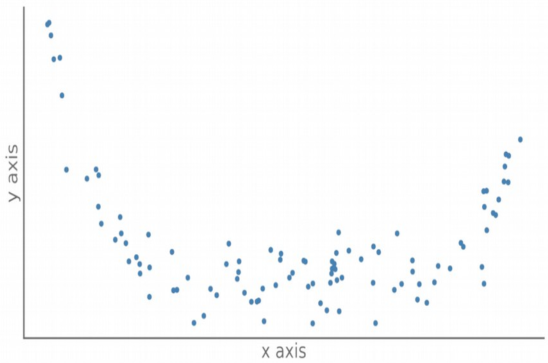
$$\text{gradient} = \frac{\delta \text{loss}}{\delta\theta}$$

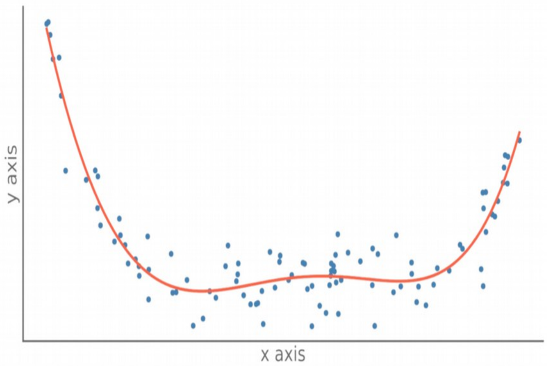
One iterative solution is : $\theta \leftarrow \theta - \eta \cdot \dfrac{\delta \text{loss}}{\delta\theta}$

where η is the learning rate

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

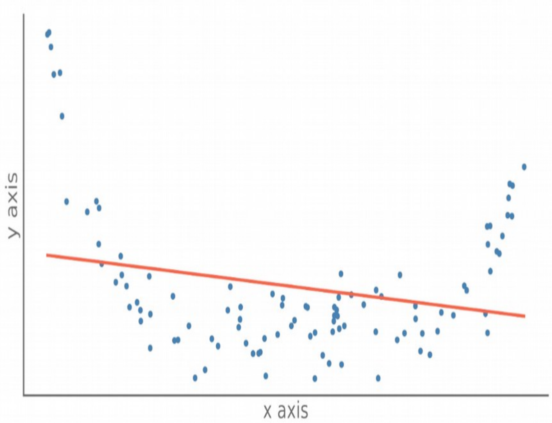
Polynomial regression



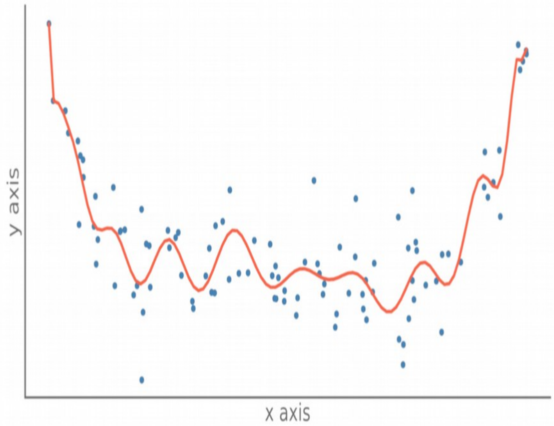


$$P_n(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \cdots + a_n \cdot x^n = \sum_{i=0}^{n} a_i \cdot x^n$$

Polynomial regression

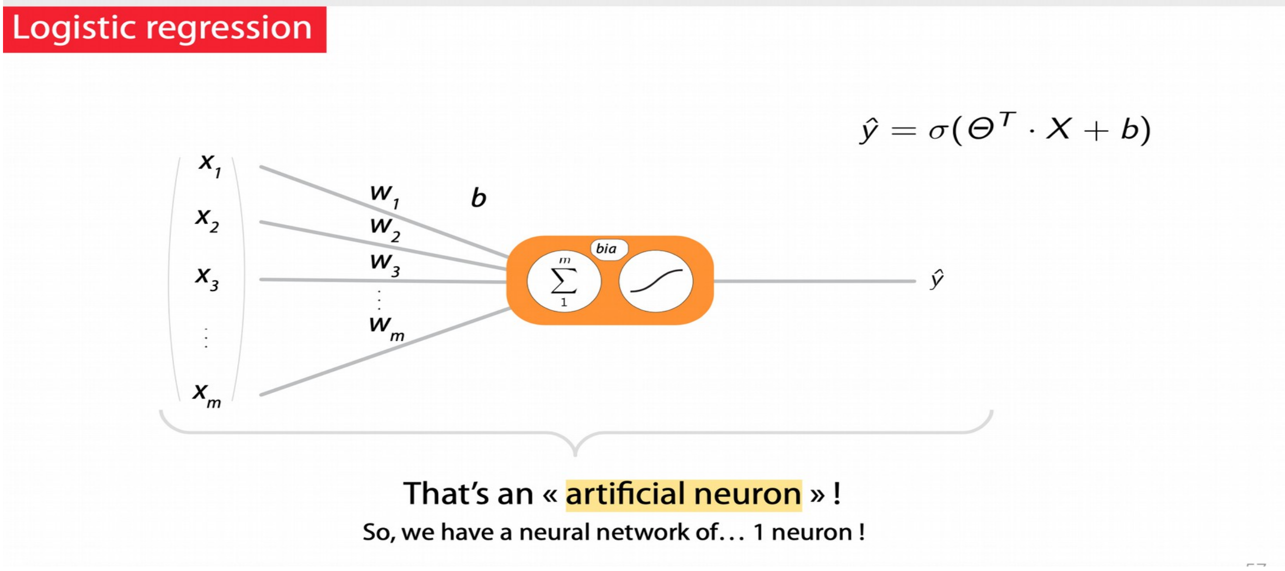


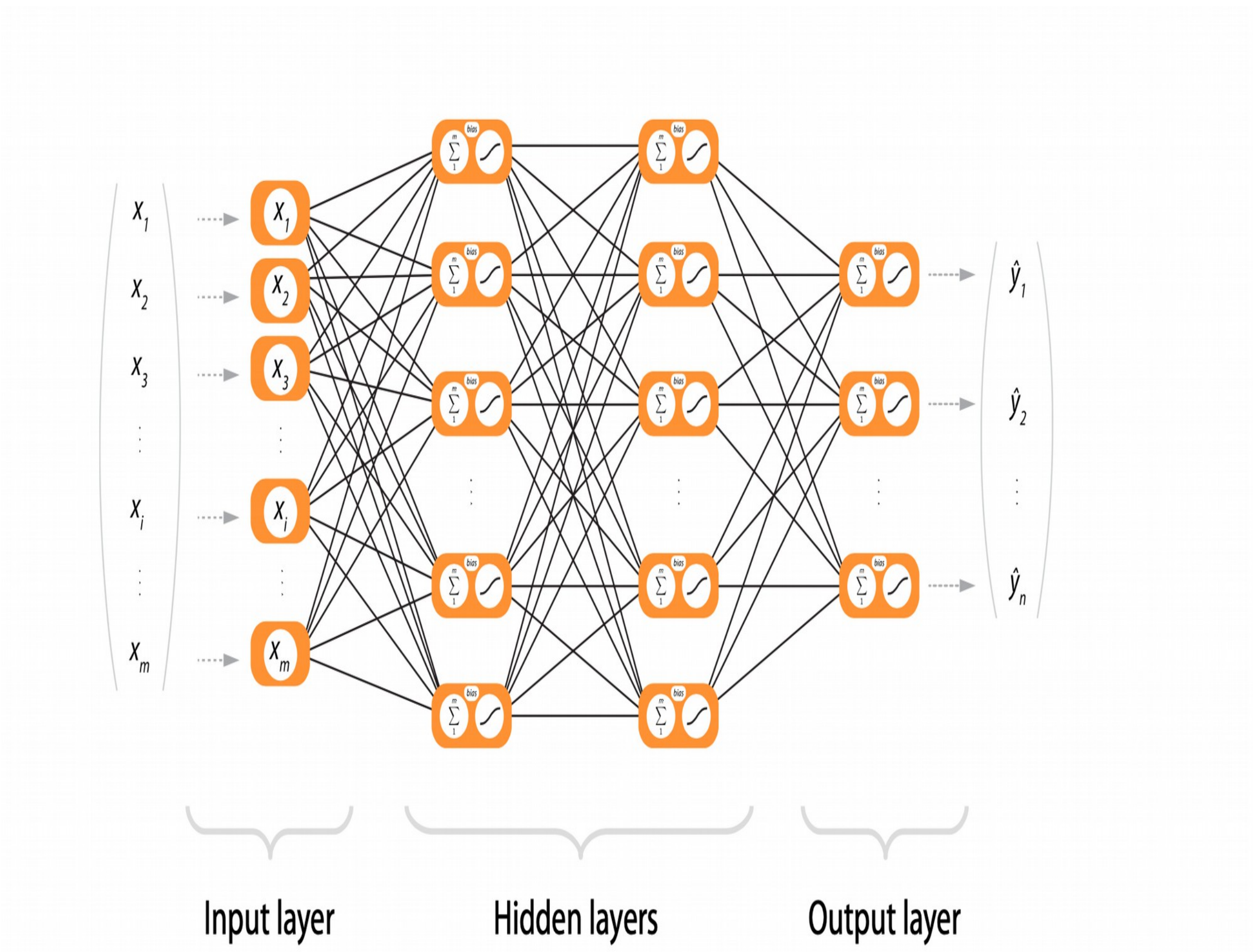
Underfiting



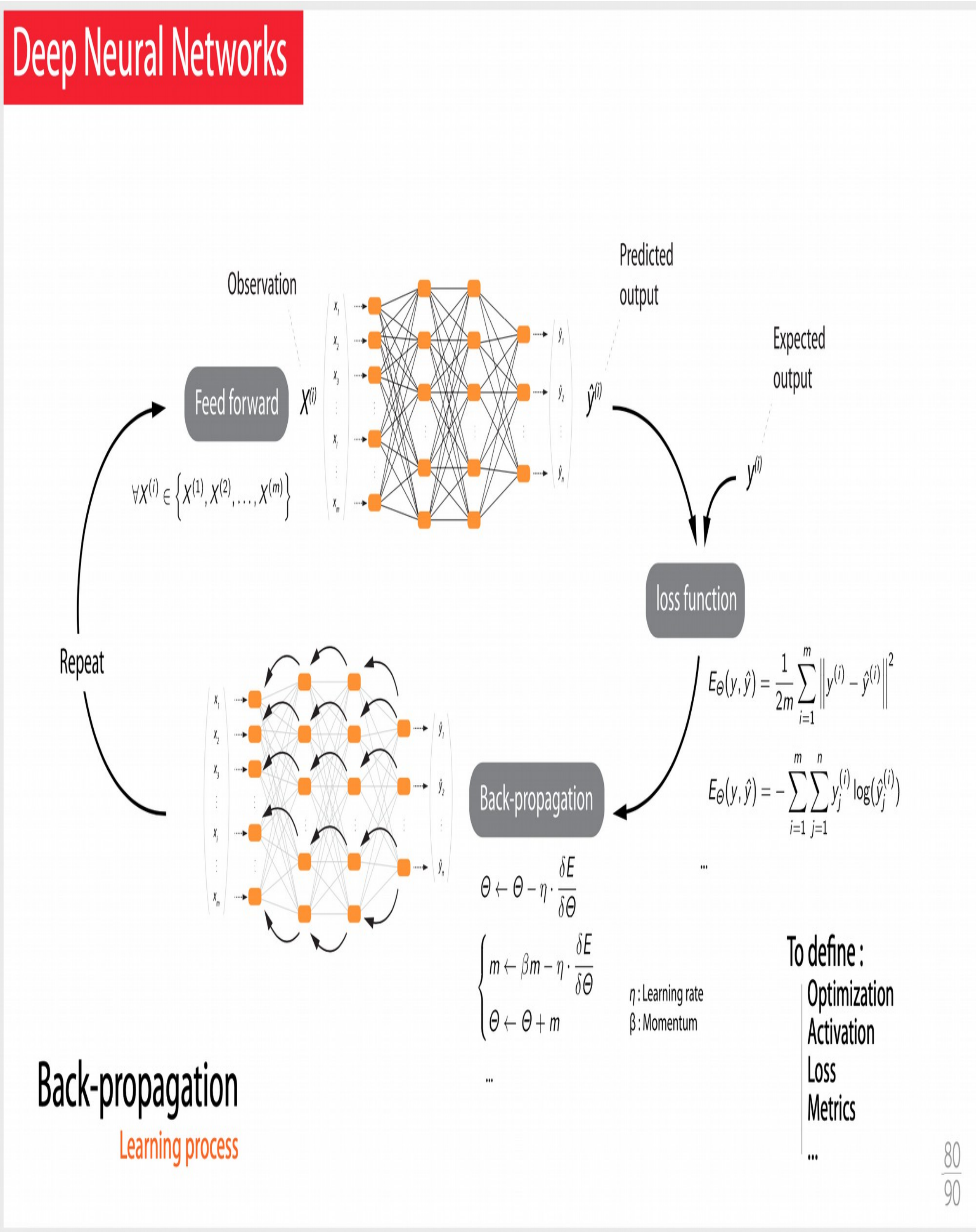
Overfiting

Architecture d'un réseau de neurone

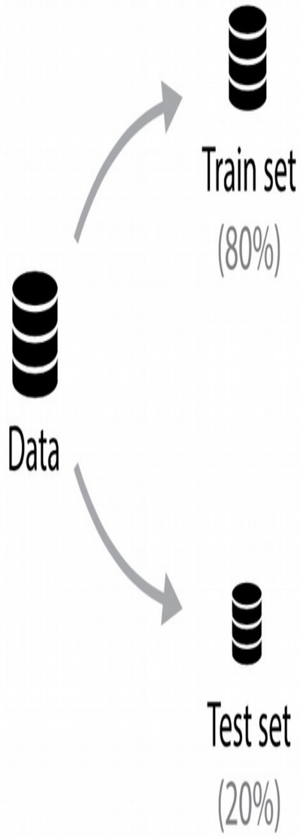




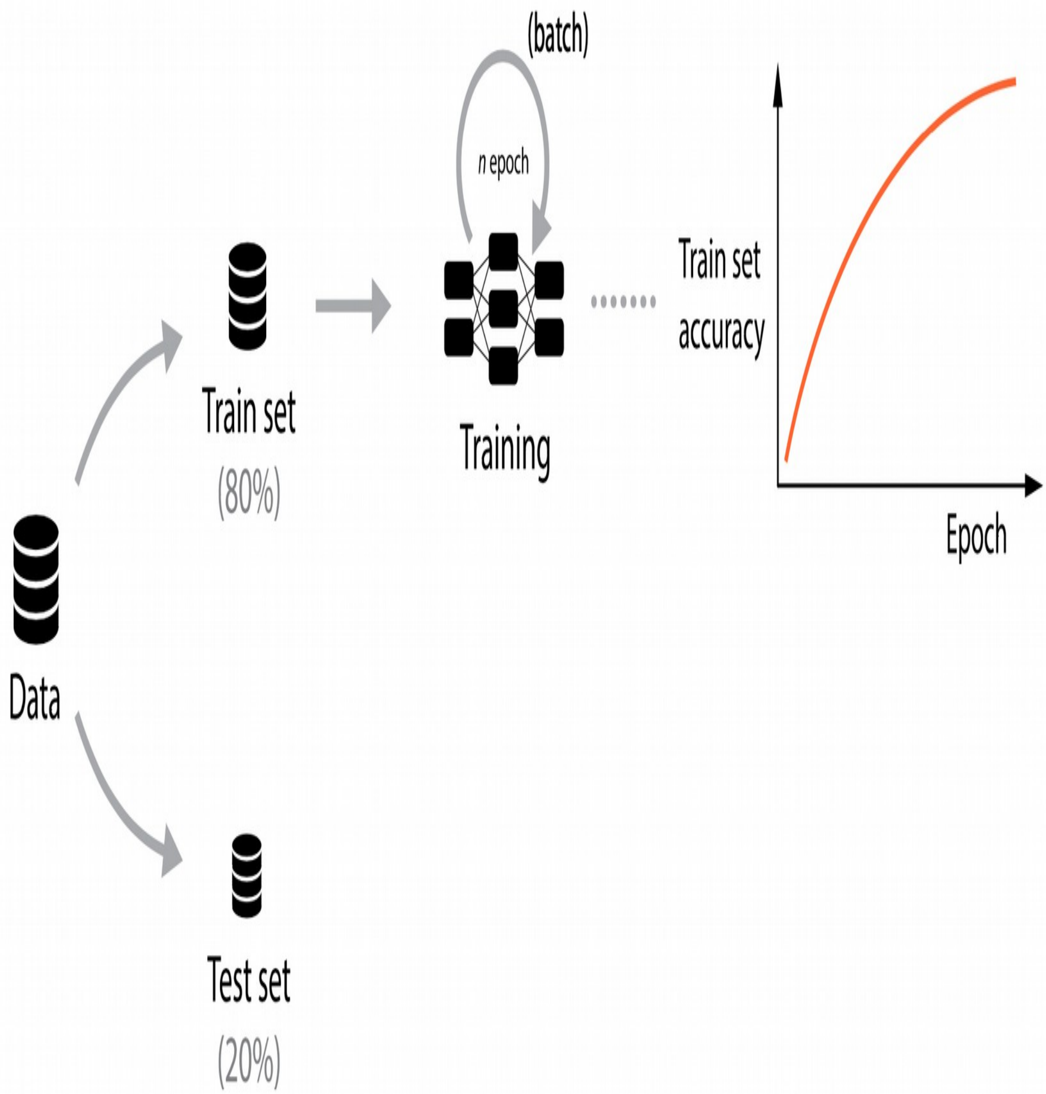
Apprentissage d'un réseau de neurone



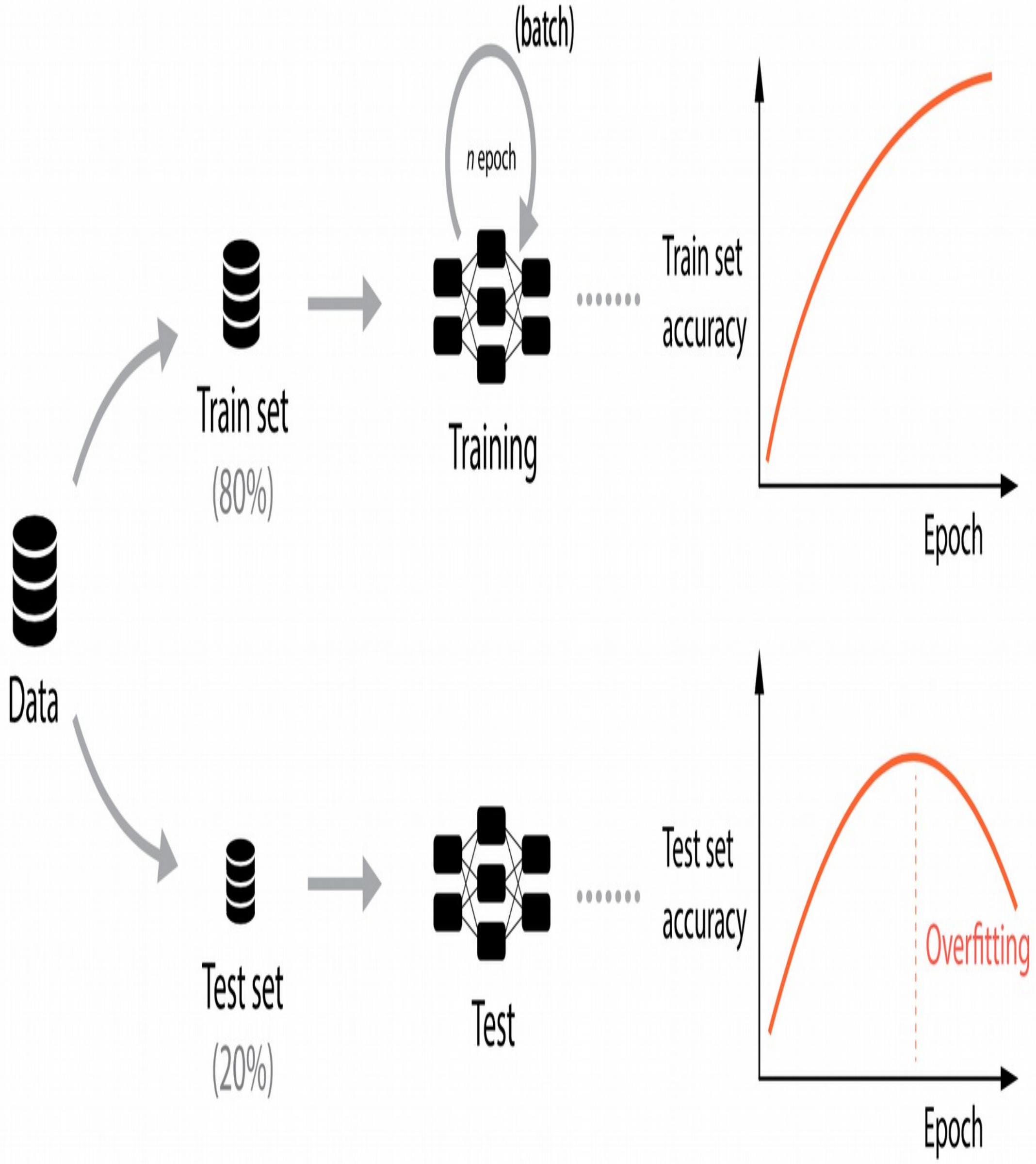
Training process - general



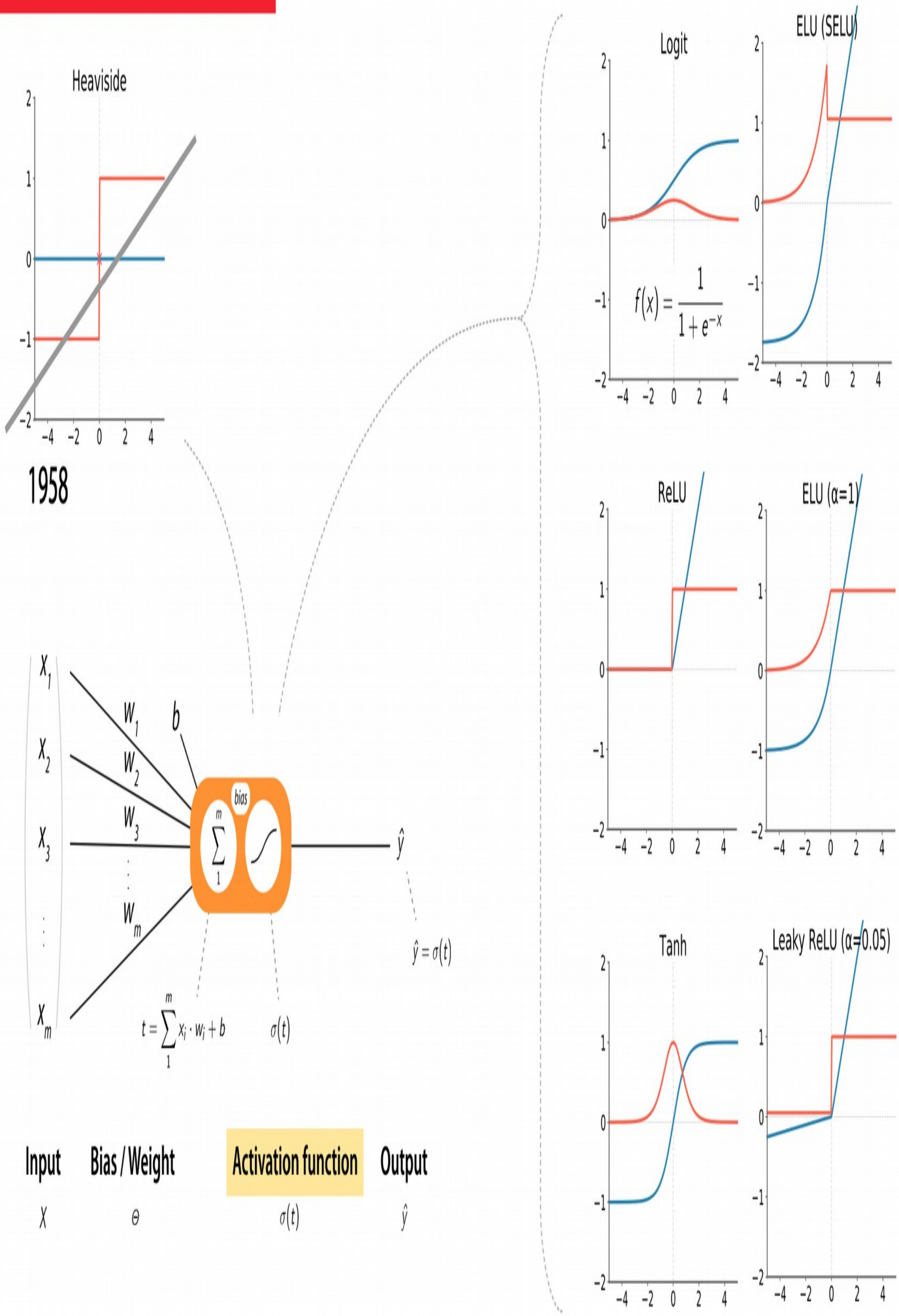
Training process - general



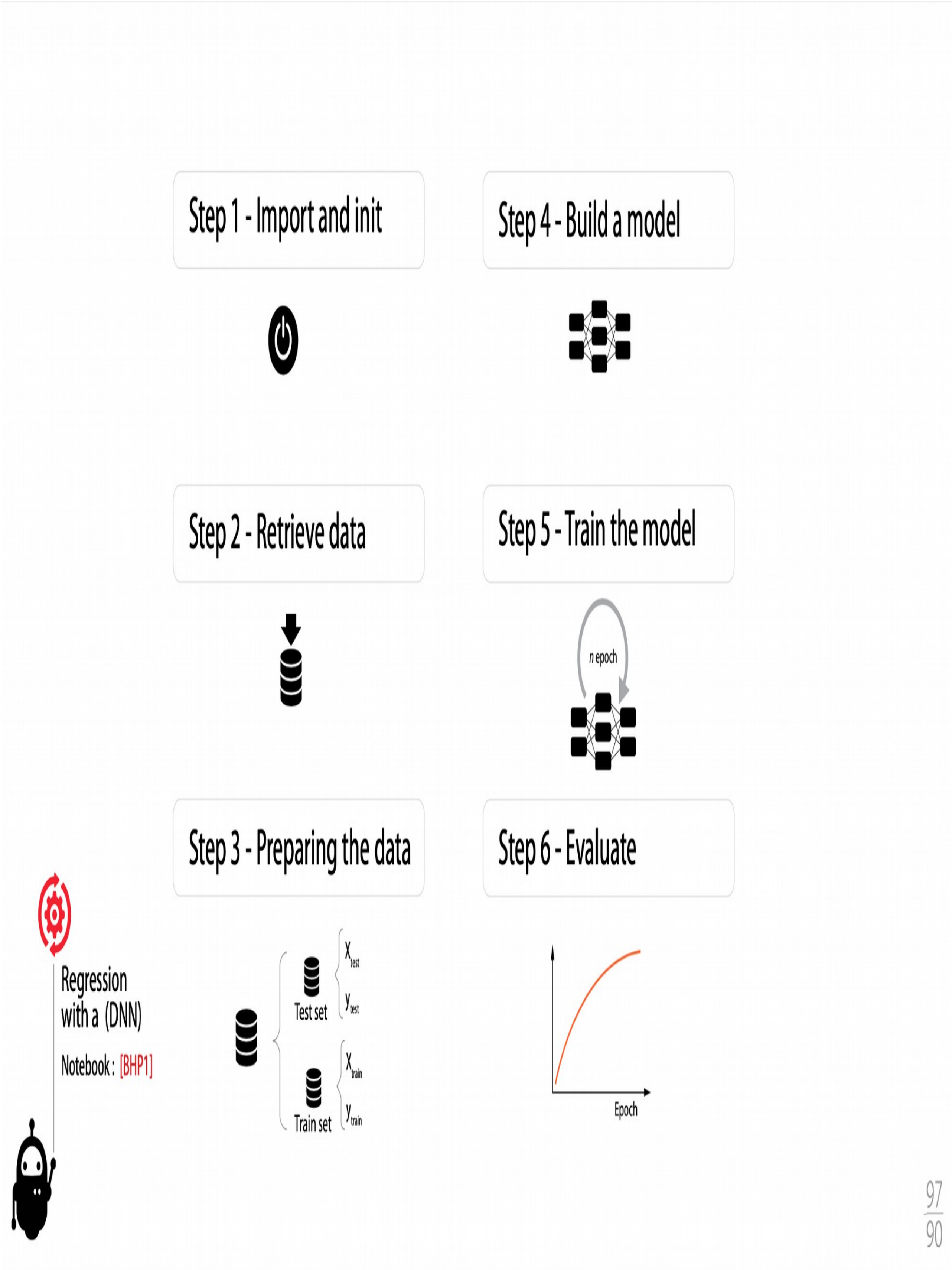
Training process - general



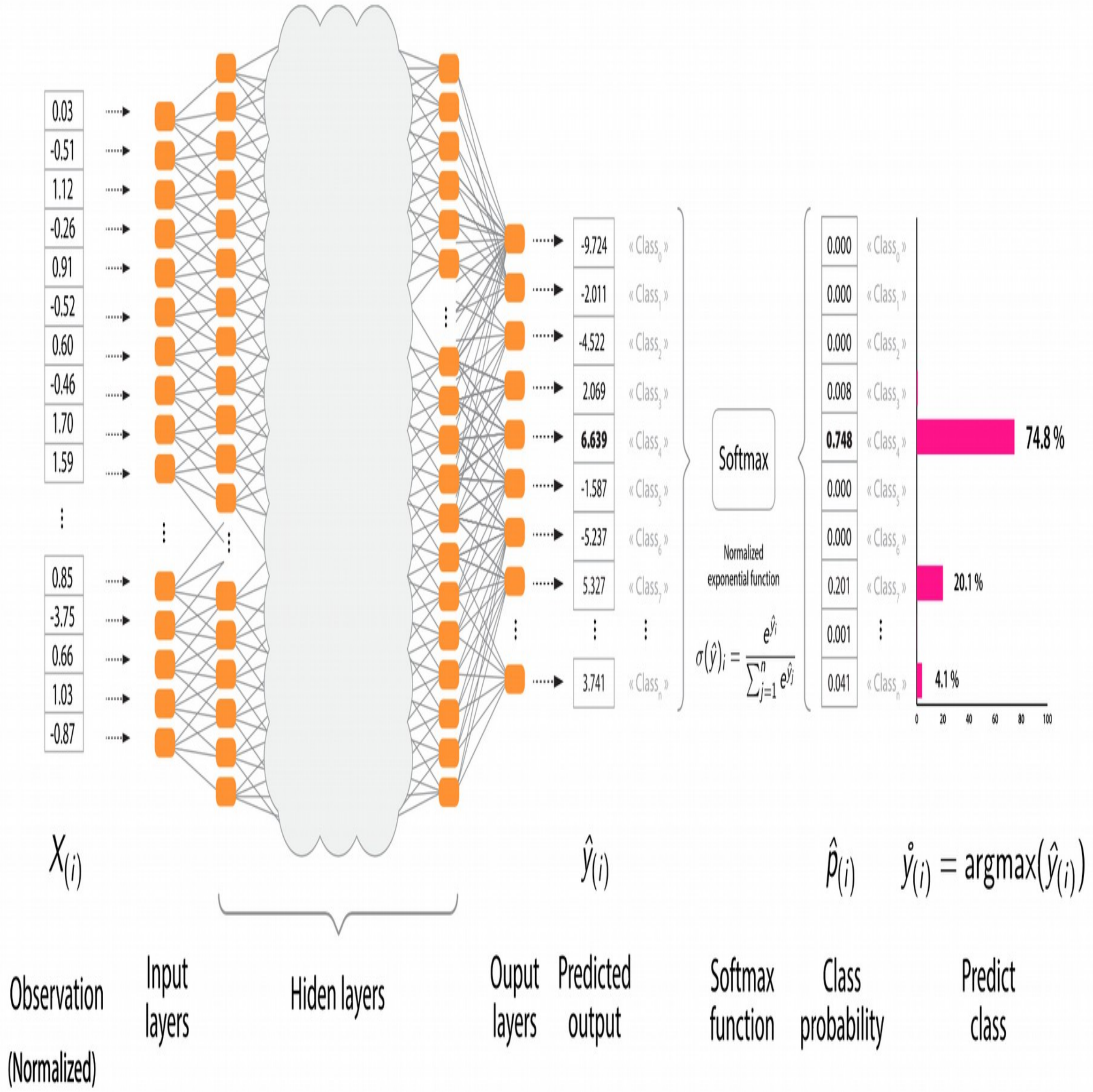
Deep Neural Networks



Etapes pour la mise en place d'un projet de machine Learning



Classification with a DNN



Hold-out evaluation

Validation simple