UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Master DSAI
Advanced Deep Learning
2022-2023

From Artificial Intelligence to a first neuron model then to MLP, CNN and RNN

Frederic Precioso
(INRIA – CNRS – UCA, Team MAASAI)

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Disclaimer

If any content in this presentation is yours but is not correctly referenced or if it should be removed, please contact me and I will correct it.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Overview

- **Context & Vocabulary**
- Math Basics
- Simple Models
- Spatial Invariance (CNN)
- Sequential Invariance (RNN)

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Overview

- **Context & Vocabulary**
  - ***What is Artificial Intelligence?***
  - ***Machine Learning without Maths***
  - ***Machine Learning VS Data Mining***
  - ***Machine Learning VS Data Science***
  - ***Machine Learning VS Statistics***
- Math Basics
- Simple Models
- Spatial Invariance (CNN)
- Sequential Invariance (RNN)

UNIVERSITÉ
CÔTE D'AZUR

univ-cotedazur.fr

CONTEXT & VOCABULARY

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

What is Artificial intelligence?

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

How is Artificial intelligence defined?

- The term ***Artificial Intelligence***, as a research field, was coined at the conference on the campus of Dartmouth College in the summer of **1956**, even though the idea was around since Antiquity: Hephaestus built automatons of metal to work for him or protect others (as in the picture on the right *Talos* protecting goddess *Europa*), the Golem made of clay in Jewish folklore (on the left), Yan Shi has built humanoid automatons 10th century BC during Zhou dynasty, etc.

- Closer to the Dartmouth conference but still before, the first manifesto on Artificial Intelligence, an unpublished report **"*Intelligent Machinery*"**, written by Alan Turing in **1948**. He already distinguished two different approaches to AI, which may be termed ***"top-down"*** and ***"bottom-up"*** *(now more commonly called knowledge-driven AI and data-driven AI respectively).*

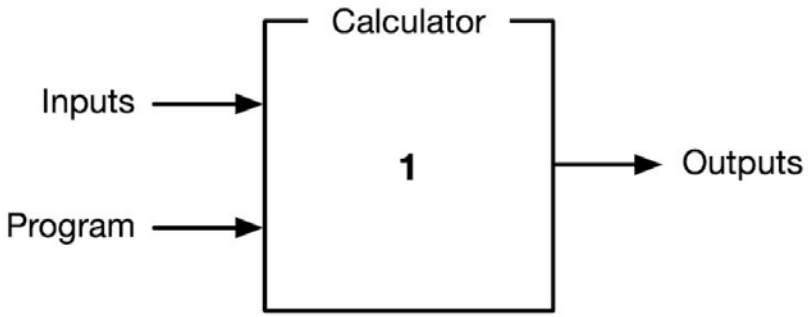
(sources: Wikipedia, https://www.greeklegendsandmyths.com/automatons.html *, Stanford Encyclopedia of Philosophy:* https://plato.stanford.edu/entries/artificial-intelligence/, http://www.alanturing.net/turing_archive/pages/Reference%20Articles/what_is_AI/What%20is%20AI02.html *)*

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

How is Artificial intelligence defined?

- ***"top-down"*** or ***knowledge-driven AI***
  - cognition = high-level phenomenon, independent of low-level details of implementation mechanism
  - Evolutionary Algorithms (1954,1957, 1960), Knowledge Representation, Reasoning (1959,1970), Expert Systems (1970), Logic, Automata, Intelligent Agent Systems (1990)…



(1) Hypothetical-deductive machines

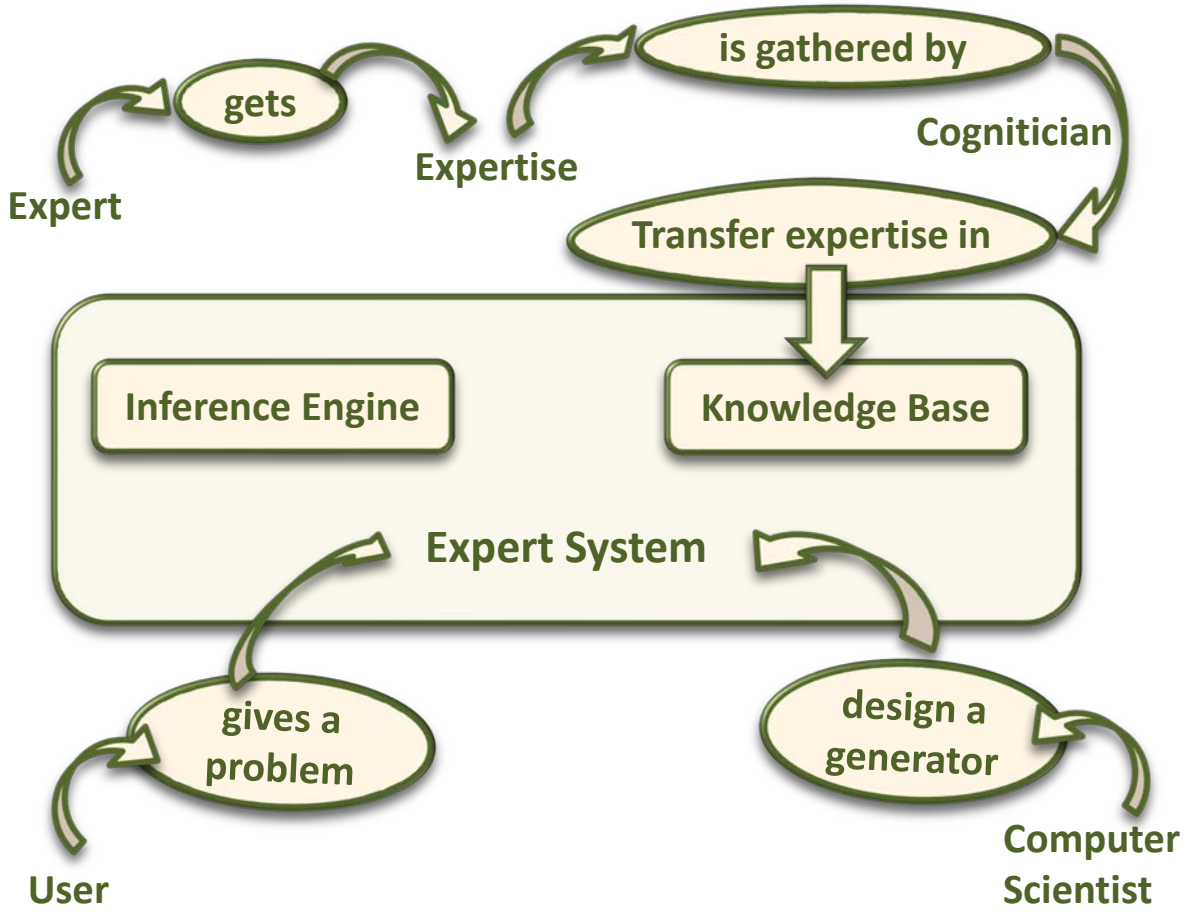
(Figure from: Neurons spike back The invention of inductive machines and the artificial intelligence controversy", D. Cardon, J.-P. Cointet, A. Mazières, Translated by Elizabeth Libbrecht In Réseaux Volume 211, Issue 5, 2018, pages 173 to 220)

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Artificial Intelligence, Top-Down

- **Example of an expert system:**

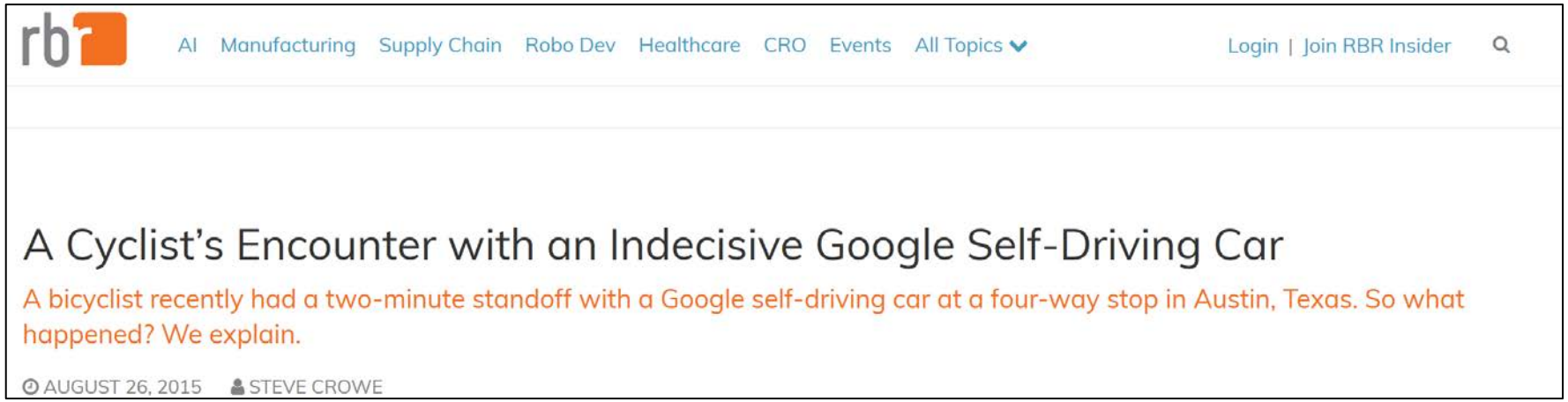


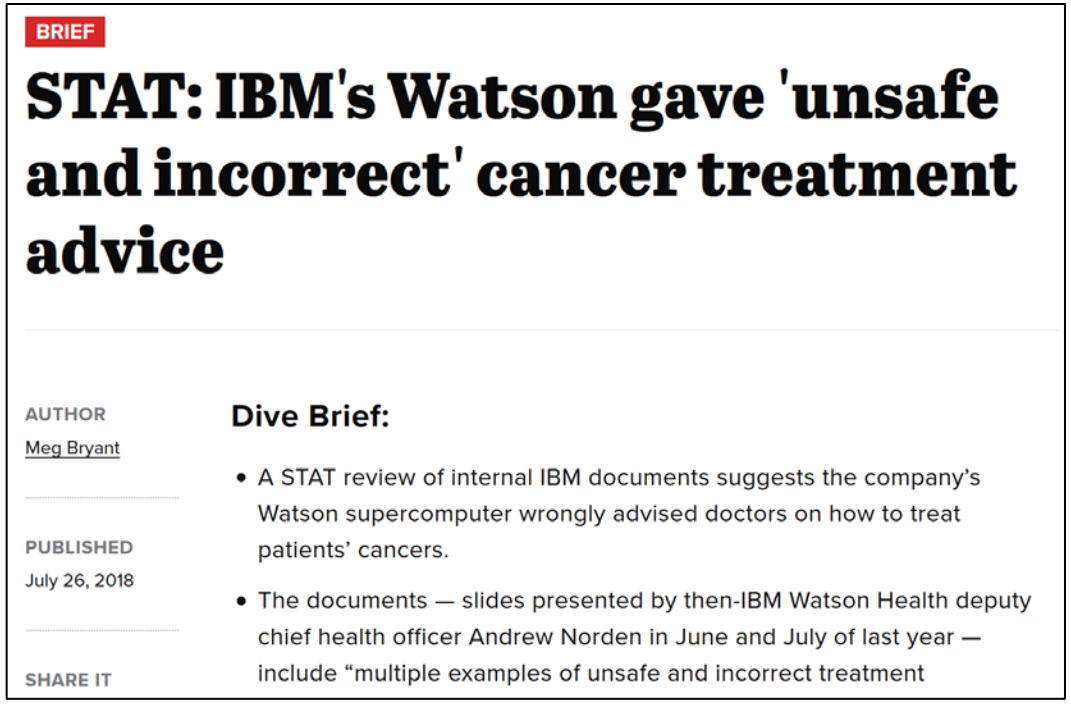
UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Artificial Intelligence, Top-Down

- **Expert system:**



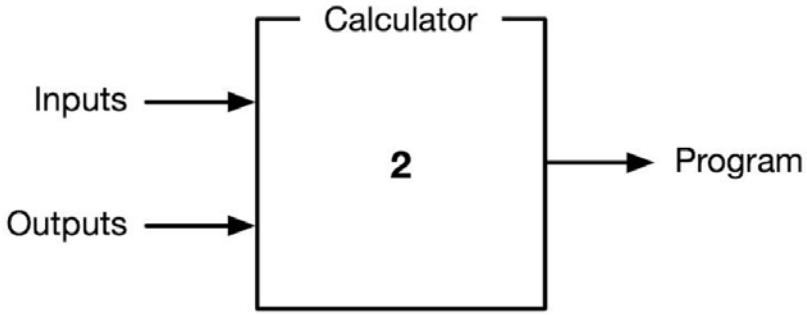


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

How is Artificial intelligence defined?

- ***"bottom-up" or data-driven AI***
  - opposite approach, start from data to build incrementally and mathematically mechanisms taking decisions
  - First neuron (1943), first neural network machine (1950), neucognitron (1975), Decision Trees (1983), Backpropagation (1984-1986), Random Forest (1995), Support Vector Machine (1995), Boosting (1995), Deep Learning (1998/2006)…



(2) inductive machines

(Figure from: Neurons spike back The invention of inductive machines and the artificial intelligence controversy", D. Cardon, J.-P. Cointet, A. Mazières, Translated by Elizabeth Libbrecht In Réseaux Volume 211, Issue 5, 2018, pages 173 to 220)

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

How is Artificial intelligence defined?

- *AI is originally defined in 1956, by Marvin Lee Minsky:*

  *"The construction of computer programs doing tasks, that are, **for the moment**, accomplished **more satisfyingly** by human beings because they require **high level mental processes** such as: learning. perceptual organization of memory and critical reasoning".*

- There are so the "artificial" side with the usage of computers or sophisticated electronic processes and the side "intelligence" associated with its goal to imitate the (human) behavior.

(sources: Wikipedia & http://www.alanturing.net & Stanford Encyclopedia of Philosophy)

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Why Artificial Intelligence is so difficult to grasp?

- Frequently, when a technique reaches **mainstream use**, it is **no longer considered as artificial intelligence**; this phenomenon is described as the ***AI effect***: "AI is whatever hasn't been done yet.“ (***Larry Tesler's Theorem***)
  -> e.g. Path Finding (GPS), Chess electronic game, Alpha Go…

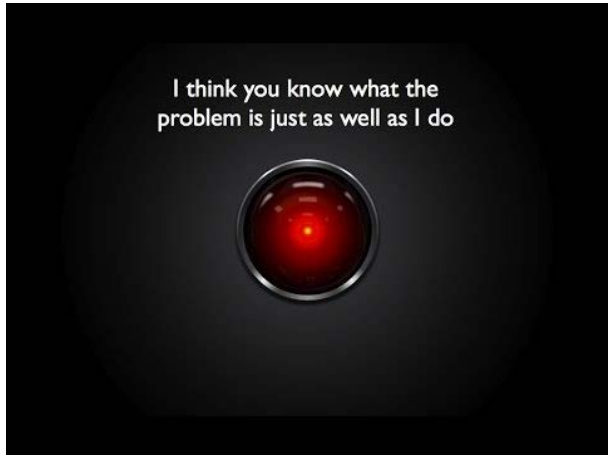
- Consequently, AI domain is continuously evolving and so very difficult to grasp.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

How is Artificial intelligence defined?

- The concept of ***strong artificial intelligence*** makes reference to a machine capable not only of producing intelligent behavior, but also to experience a feeling of a real sense of itself, “real feelings” (whatever may be put behind these words), and "an understanding of its own arguments”.



UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

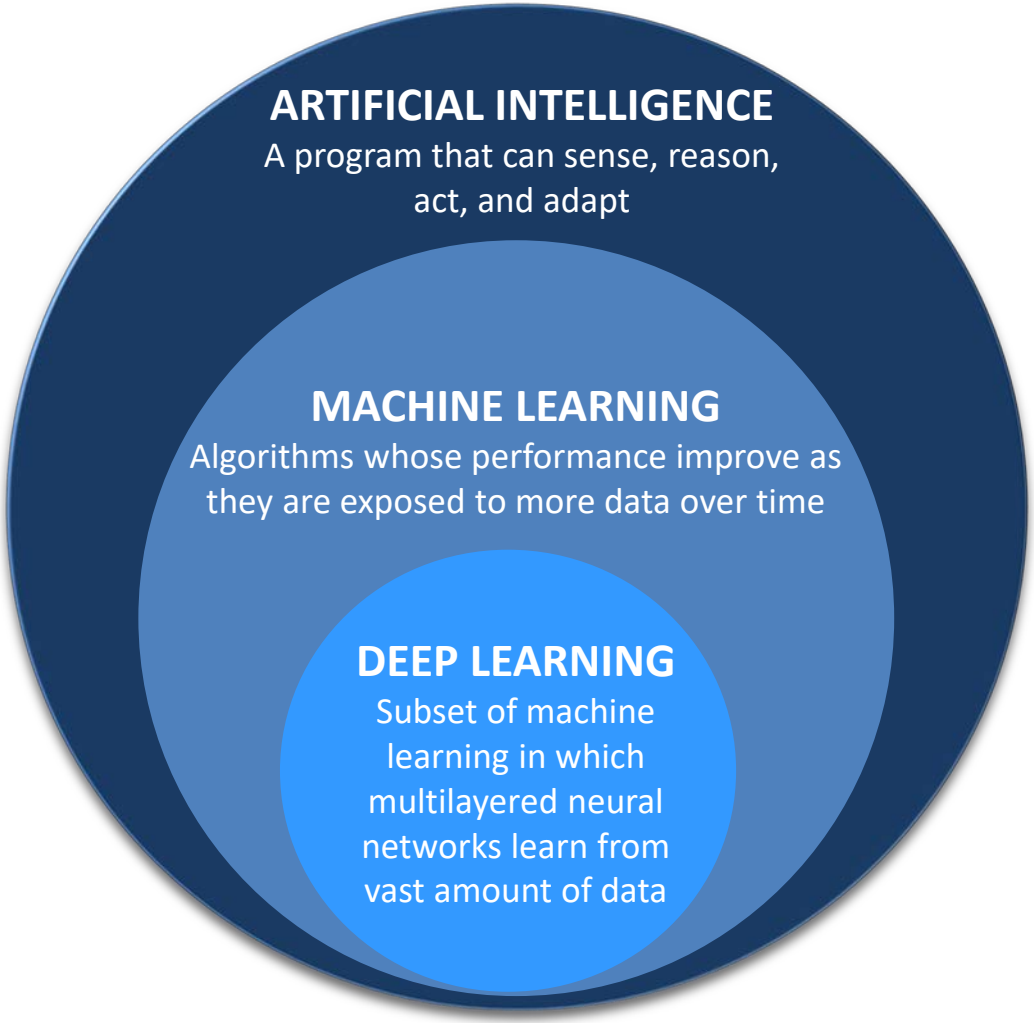
How is Artificial intelligence defined?

- The notion of ***weak artificial intelligence*** is a pragmatic approach of engineers: targeting to build more autonomous systems (to reduce the cost of their supervision), algorithms capable of solving problems of a certain class, etc. But this time, the machine *simulates* the intelligence, it seems to act as if it was smart.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

AI vs Machine Learning vs Deep Learning



But what is Machine Learning?

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Machine Learning

$$\left(\mathbf{x}\right) \xrightarrow{f(\mathbf{X},\boldsymbol{\alpha})\ ?} y$$

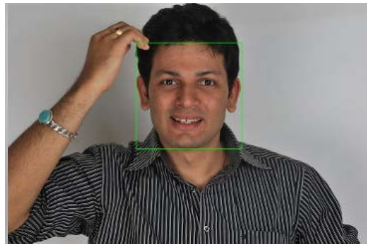
$\left(\mathbf{x}\right)$

Face detection

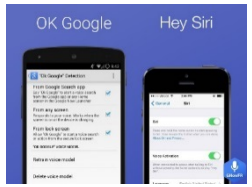
Scores prediction

Voice recognition

y



Sport bets

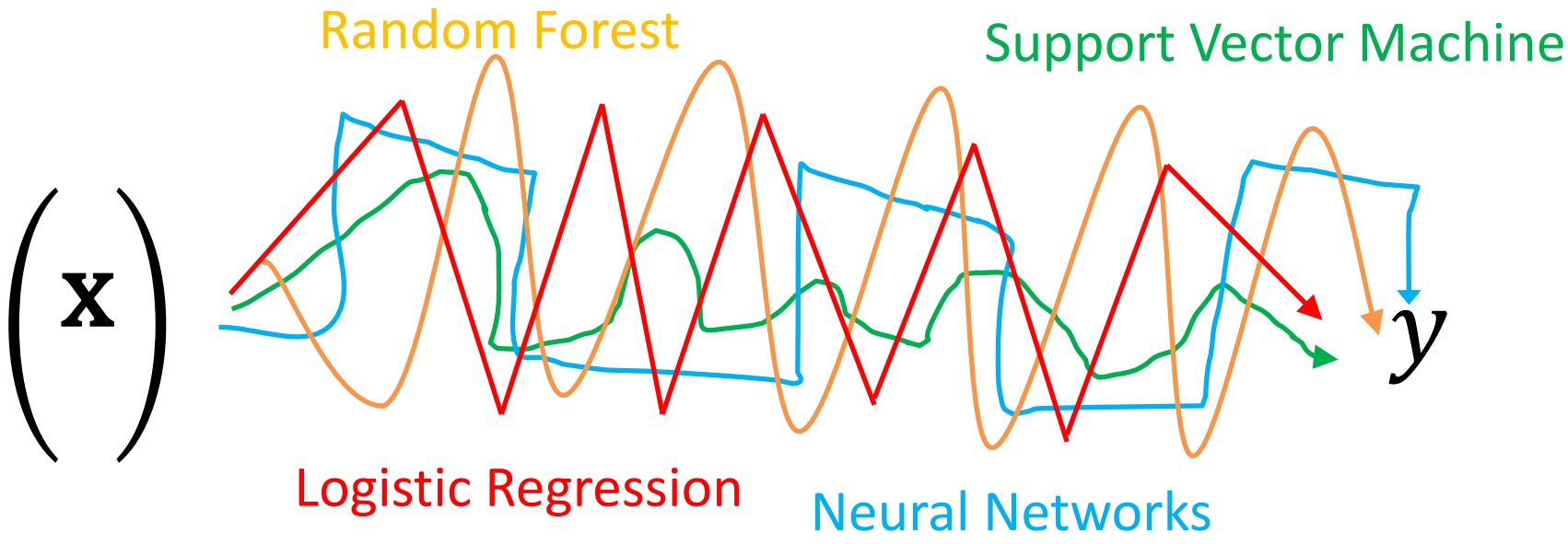


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Machine Learning

$$\left(\mathbf{x}\right) \xrightarrow{f(\mathbf{X},\alpha)\ ?} y$$

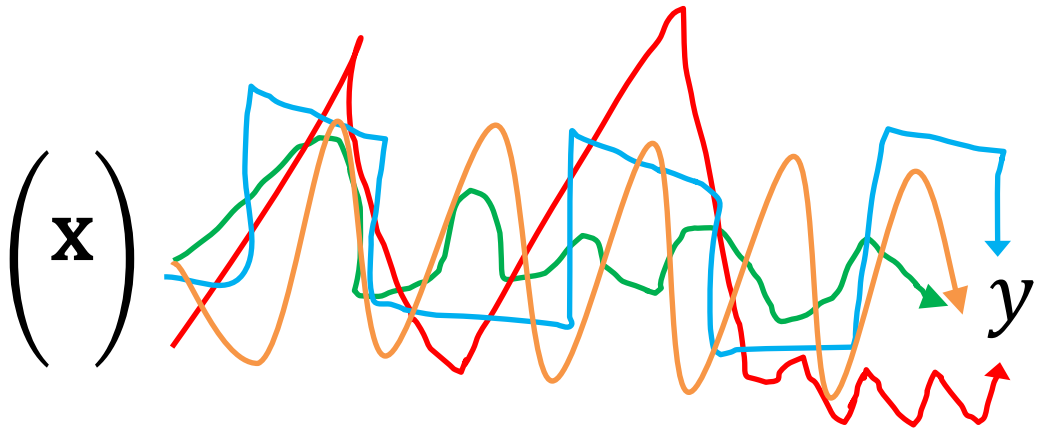


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Machine Learning

$$\left(\mathbf{x}\right) \xrightarrow{f(\mathbf{X},\alpha)\ ?} y$$



ML

$\Leftrightarrow$ "Weather Forcasting"

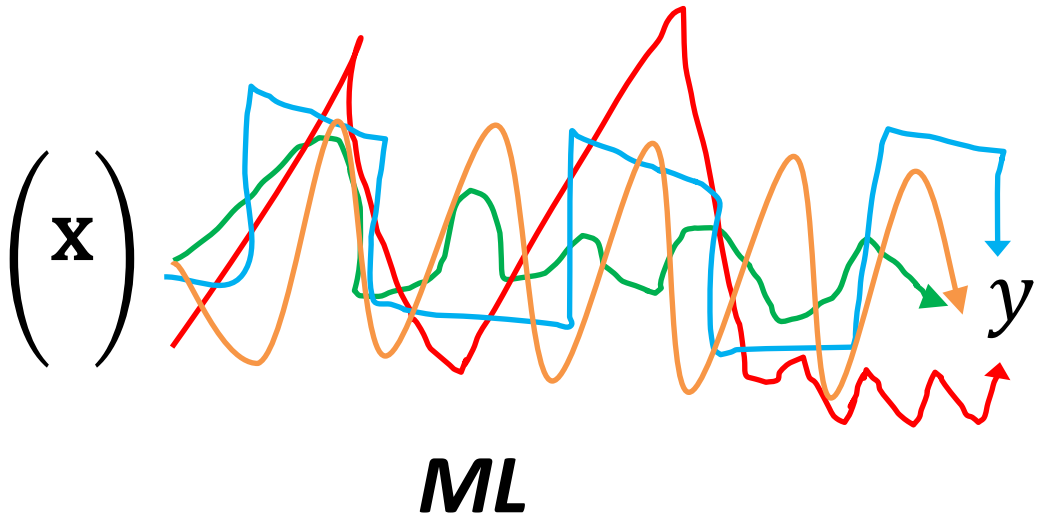
Cf. Question trolley dilemma

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Machine Learning

$$\left(\mathbf{x}\right) \xrightarrow{f(\mathbf{X},\alpha)\ ?} y$$



ML

$\neq$

AI

Francis Bach at *Frontier Research and Artificial Intelligence Conference*: **"Machine Learning is not AI"**
(https://erc.europa.eu/sites/default/files/events/docs/Francis_Bach-SEQUOIA-Robust-algorithms-for-learning-from-modern-data.pdf
https://webcast.ec.europa.eu/erc-conference-frontier-research-and-artificial-intelligence-25#)

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Beware of the diversion!

Trolley dilemma

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Back to weather forecast

- **The impact of summer weather on our economy!**
  - The weather influences the sales of products and impacts the revenues of companies in various sectors.
  - According to a study by Credoc, **1°C above seasonal norms leads to a 1.5% increase in tourist numbers**, which leaves professionals in this sector on the lookout for rain or shine.
  - More specifically, in the hotel sector, **a drop of only 1°C compared to seasonal norms in June, for example, will reduce sales by 8 to 10%.**

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Back to weather forecast

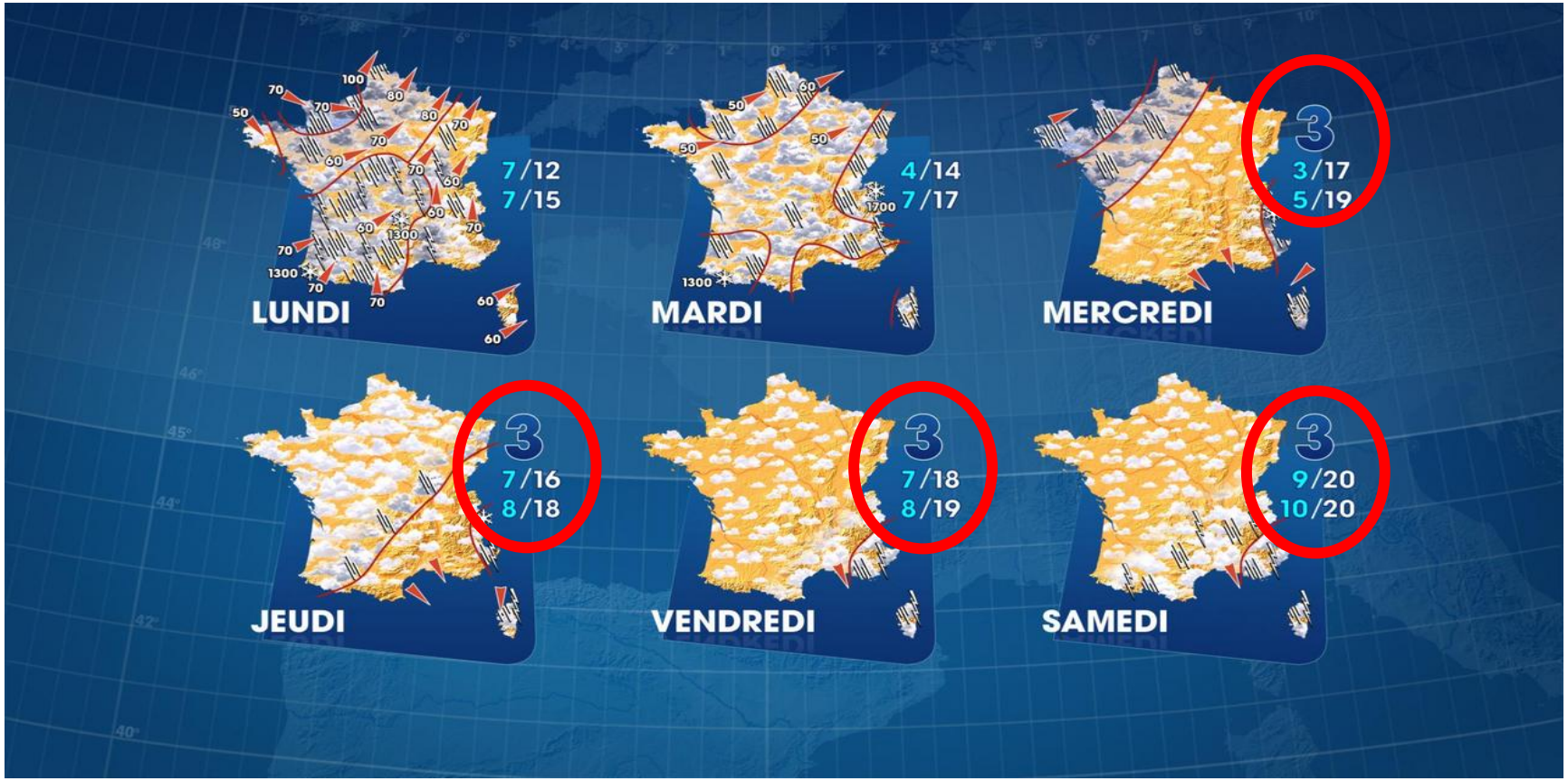
- Should we penalize algorithms that predict bad weather?
- Are they unethical?
- If in doubt, should we favor positive predictions for tourism? But then what about shops selling sweaters and coats?

- What is the solution?

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Back to weather forecast

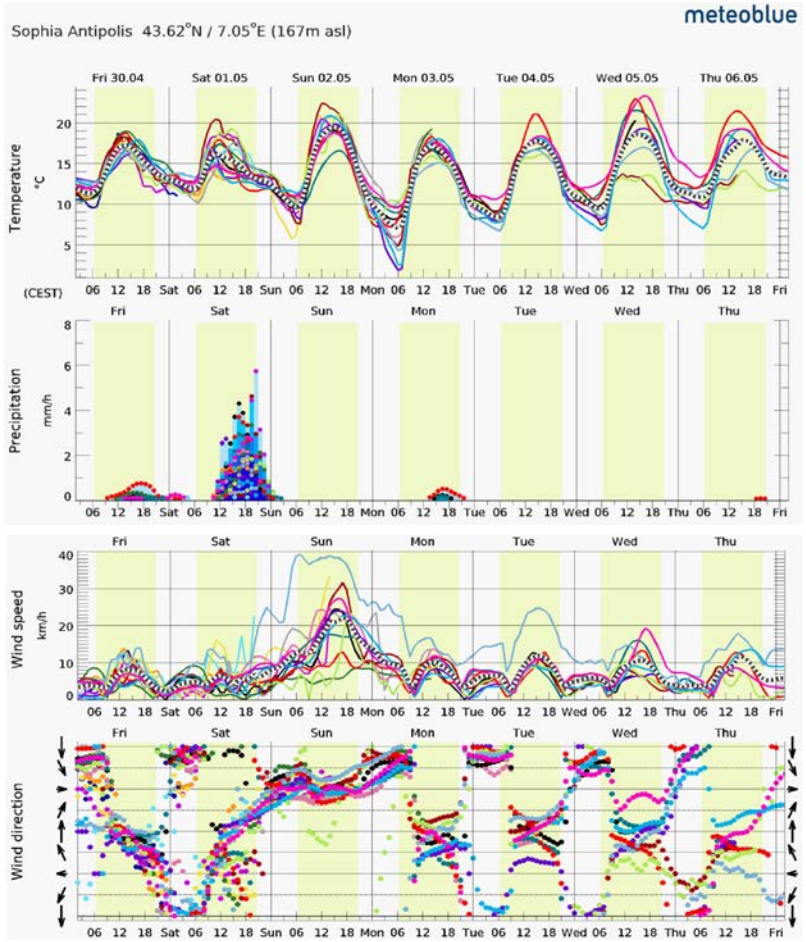


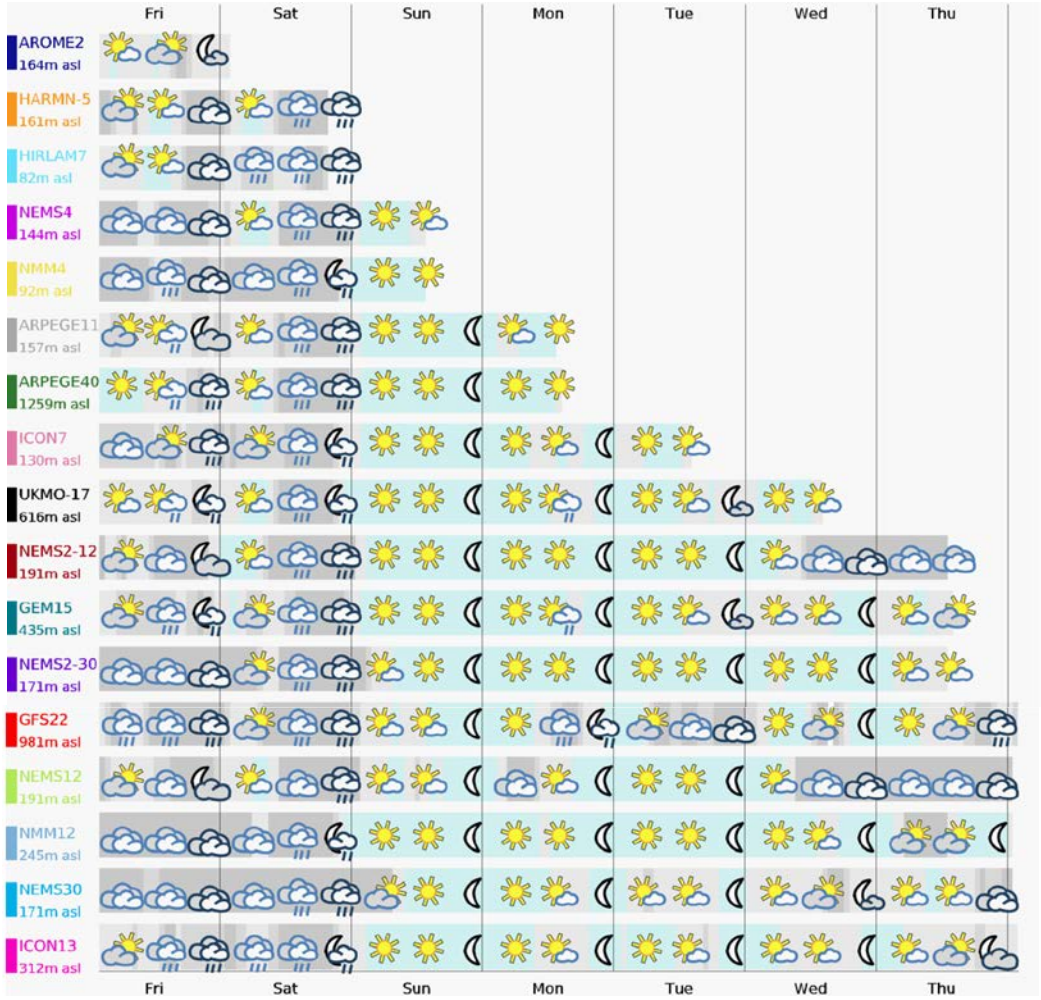
A solution: Defining a confidence indice

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Back to weather forecast





A solution: Combining different models

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Partial conclusions

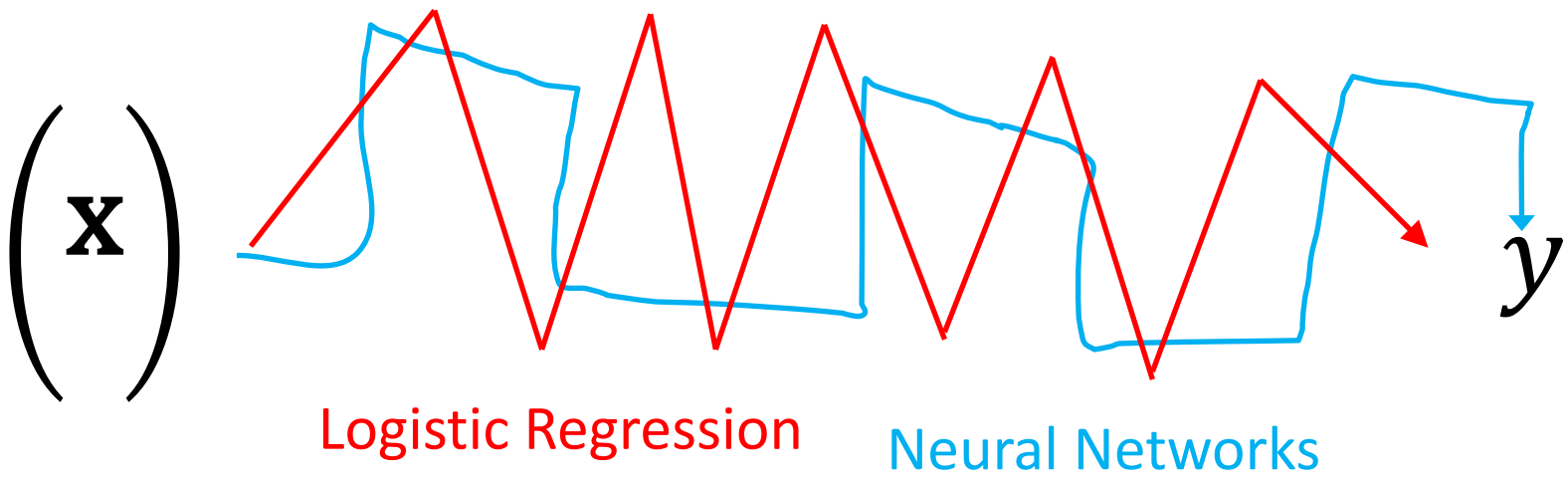
- A lot of what works today is not strictly speaking AI (compared to the original definition)
- It is the same as predicting the weather, predicting earthquakes or predicting the state of financial markets by very large mathematical models (hypothetico-deductive vs. inductive).
- Are we asking these models to be ethical?
- What does it mean that a mathematical function is ethical (or not)? Is $\sqrt{}$ more ethical than cosine? *(even if I have my own opinion on that point ;-))*
- We can only measure the model on examples for which we know the result!

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Machine Learning

$$\left(\mathbf{x}\right) \xrightarrow{f(\mathbf{X},\alpha)\ ?} y$$



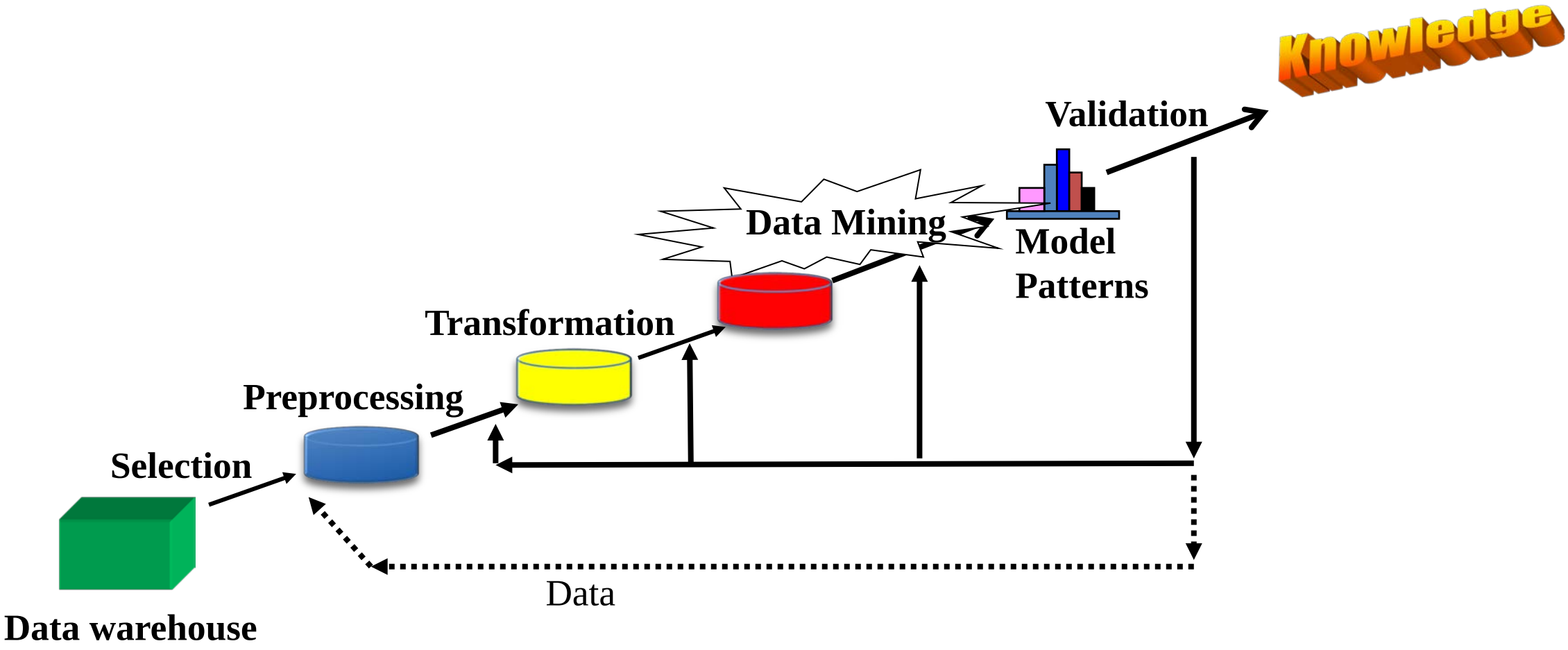
UNIVERSITÉ CÔTE D'AZUR

Machine learning VS Data Mining?

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

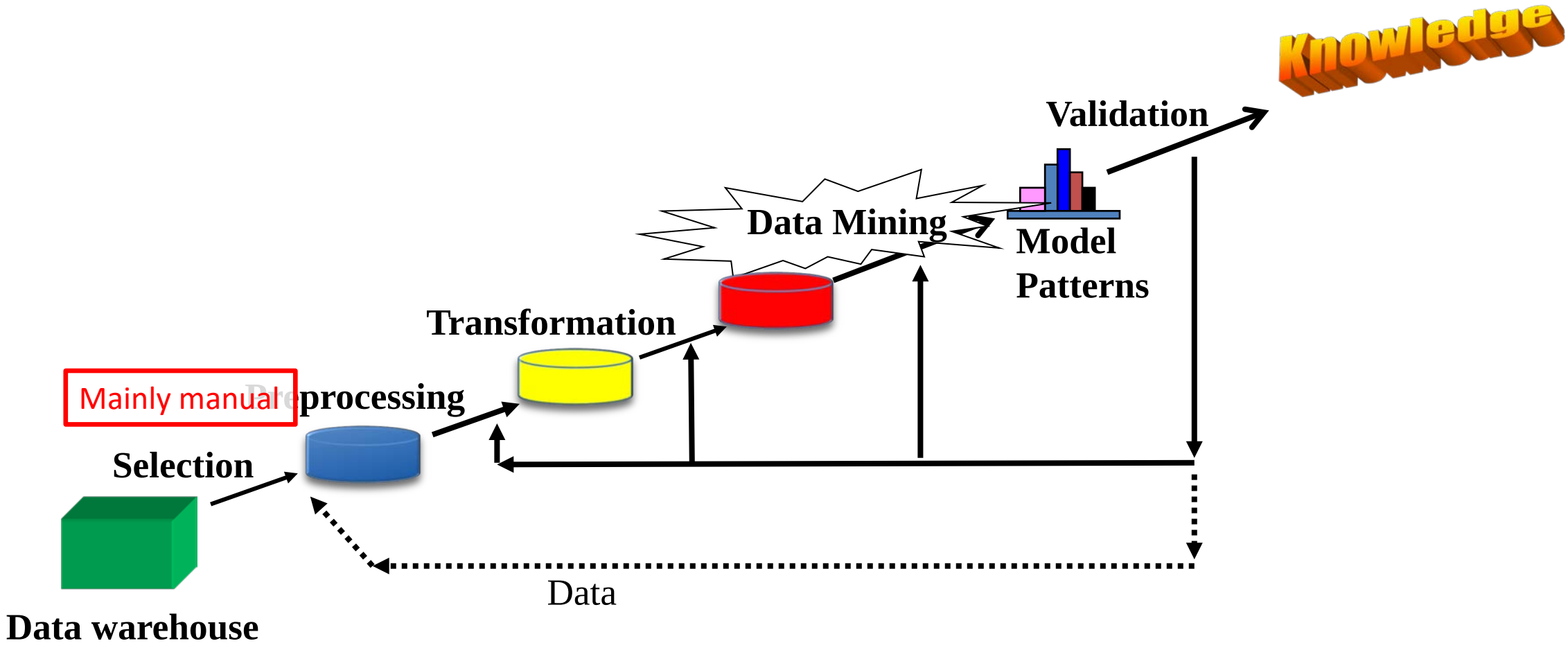
Data Mining Workflow



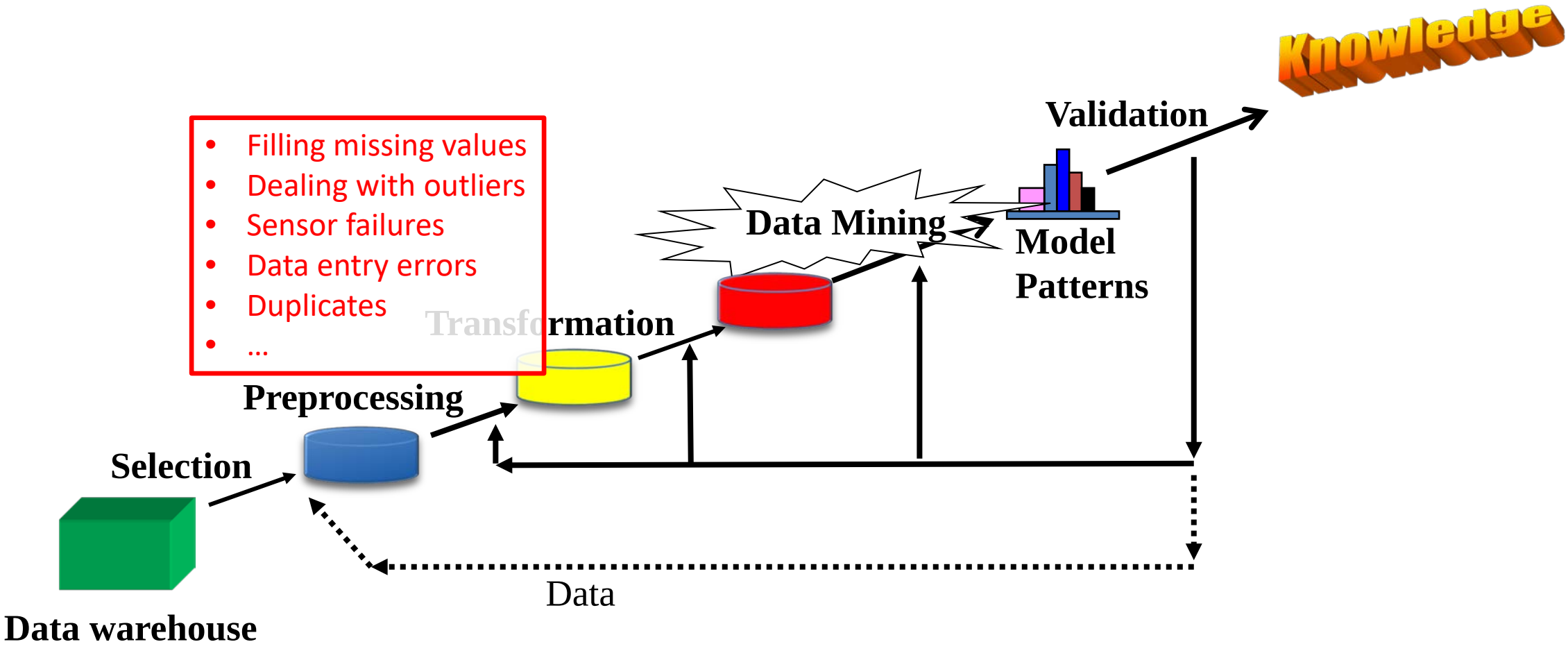
UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

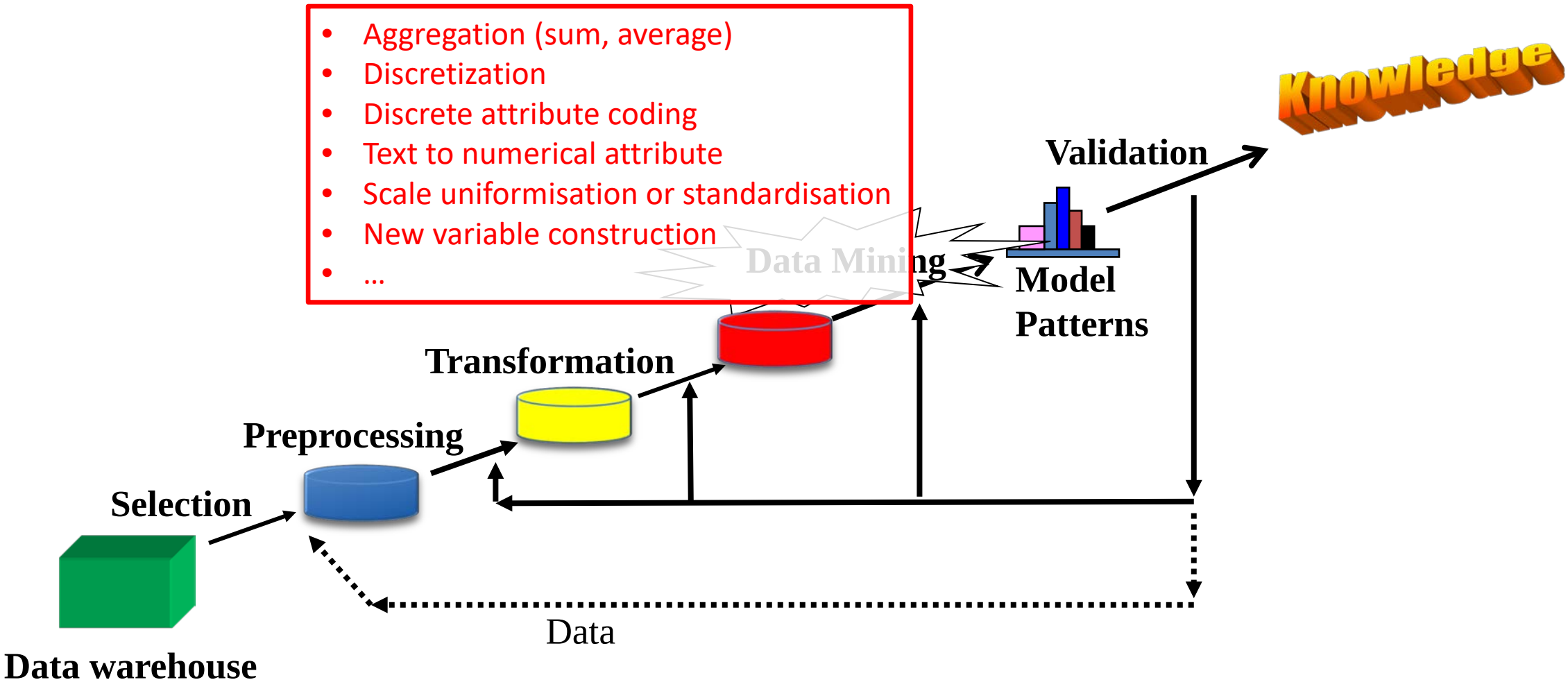
Data Mining Workflow



Data Mining Workflow



Data Mining Workflow

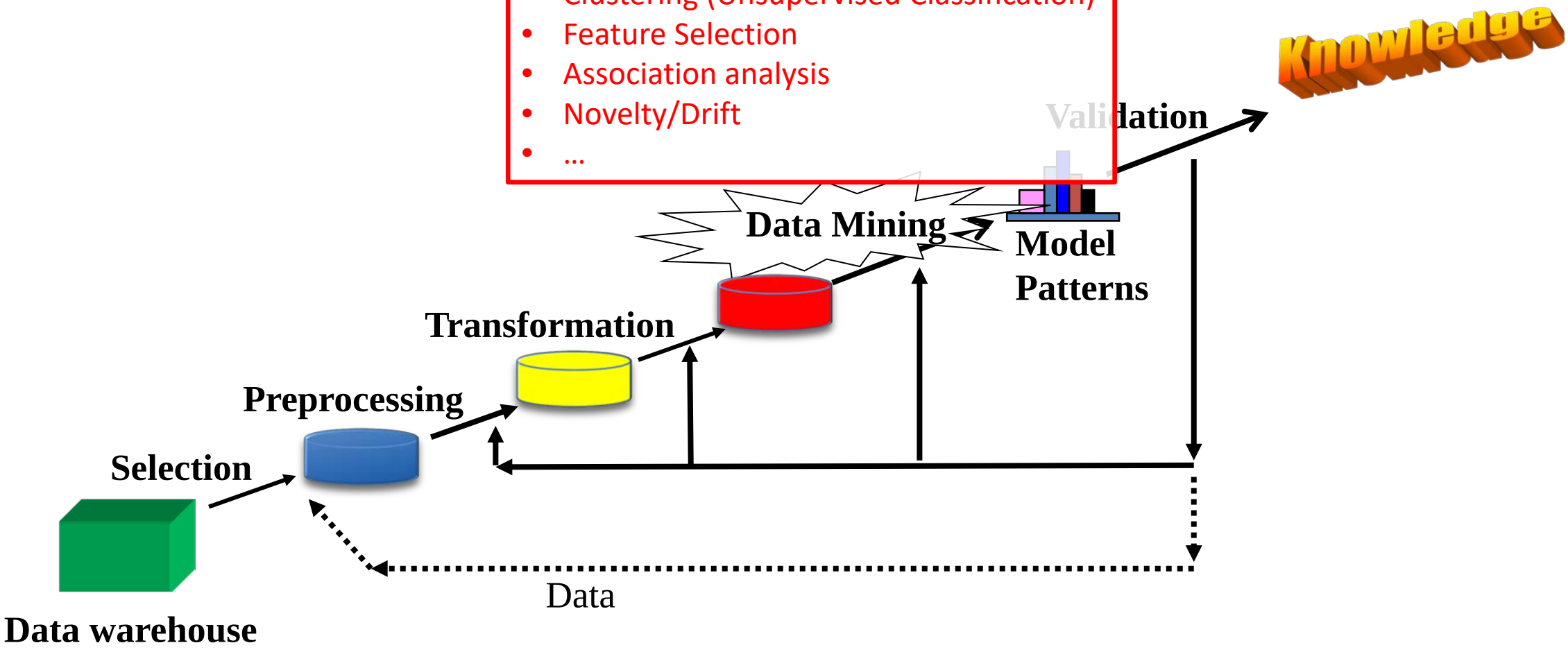


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

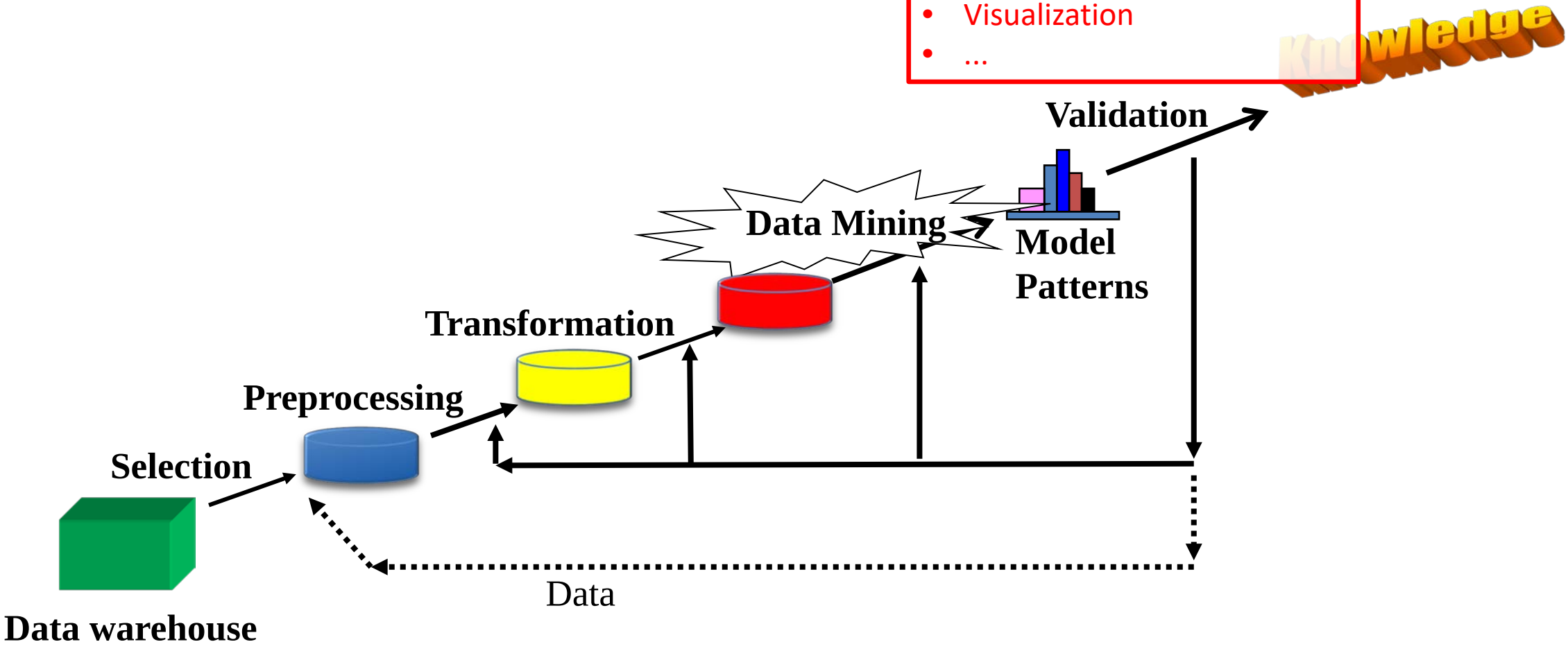
Data Mining Workflow

- Regression
- (Supervised) Classification
- Clustering (Unsupervised Classification)
- Feature Selection
- Association analysis
- Novelty/Drift
- …



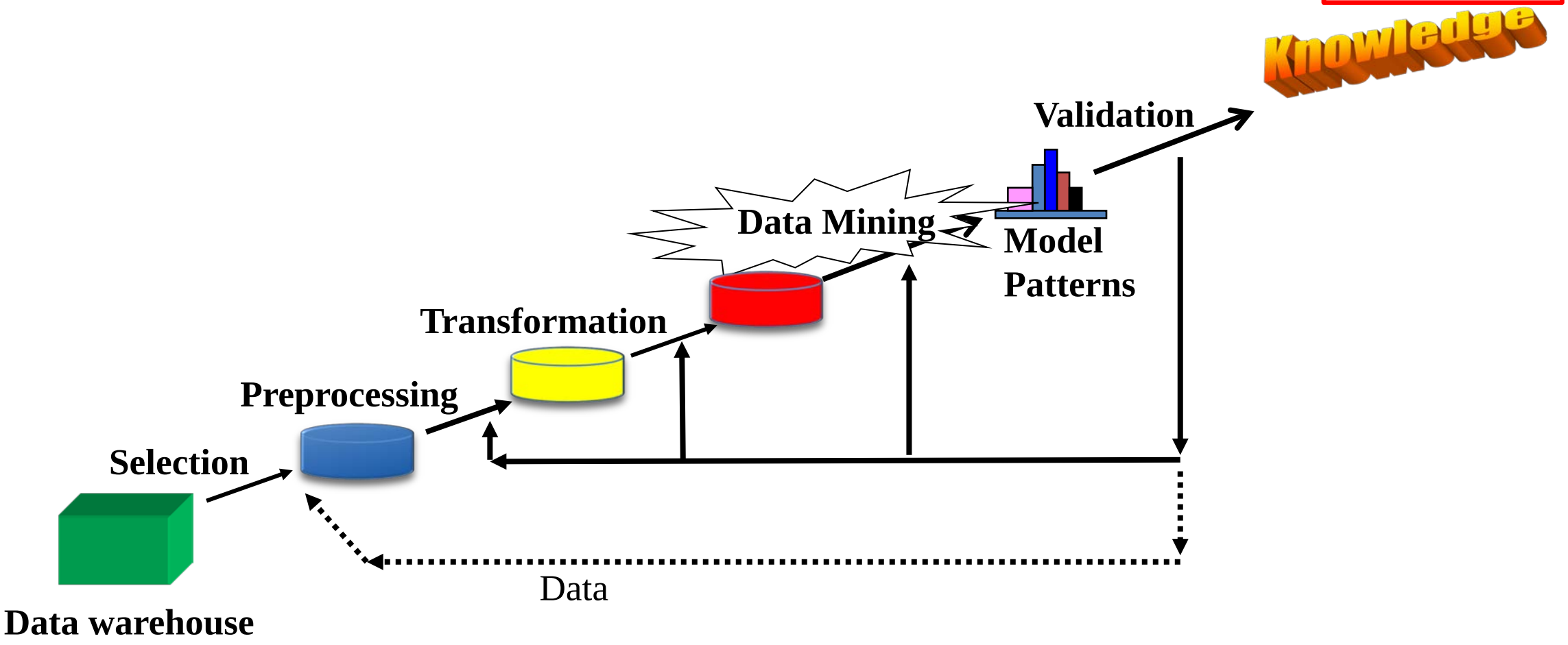
Data Mining Workflow

- Evaluation on Validation Set
- Evaluation Measures
- Visualization
- ...



Data Mining Workflow

- Visualization
- Reporting
- Knowledge
- …



UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Data Mining Workflow

Problems

- Regression
- (Supervised) Classification
- Density Estimation / Clustering (Unsupervised Classification)
- Feature Selection
- Association analysis
- Anomality/Novelty/Drift
- …

Possible Solutions

- Machine Learning
  - Support Vector Machine
  - Artificial Neural Network
  - Boosting
  - Decision Tree
  - Random Forest
  - …
- Statistical Learning
  - Gaussian Models (GMM)
  - Naïve Bayes
  - Gaussian processes
  - …
- Other techniques
  - Galois Lattice
  - …

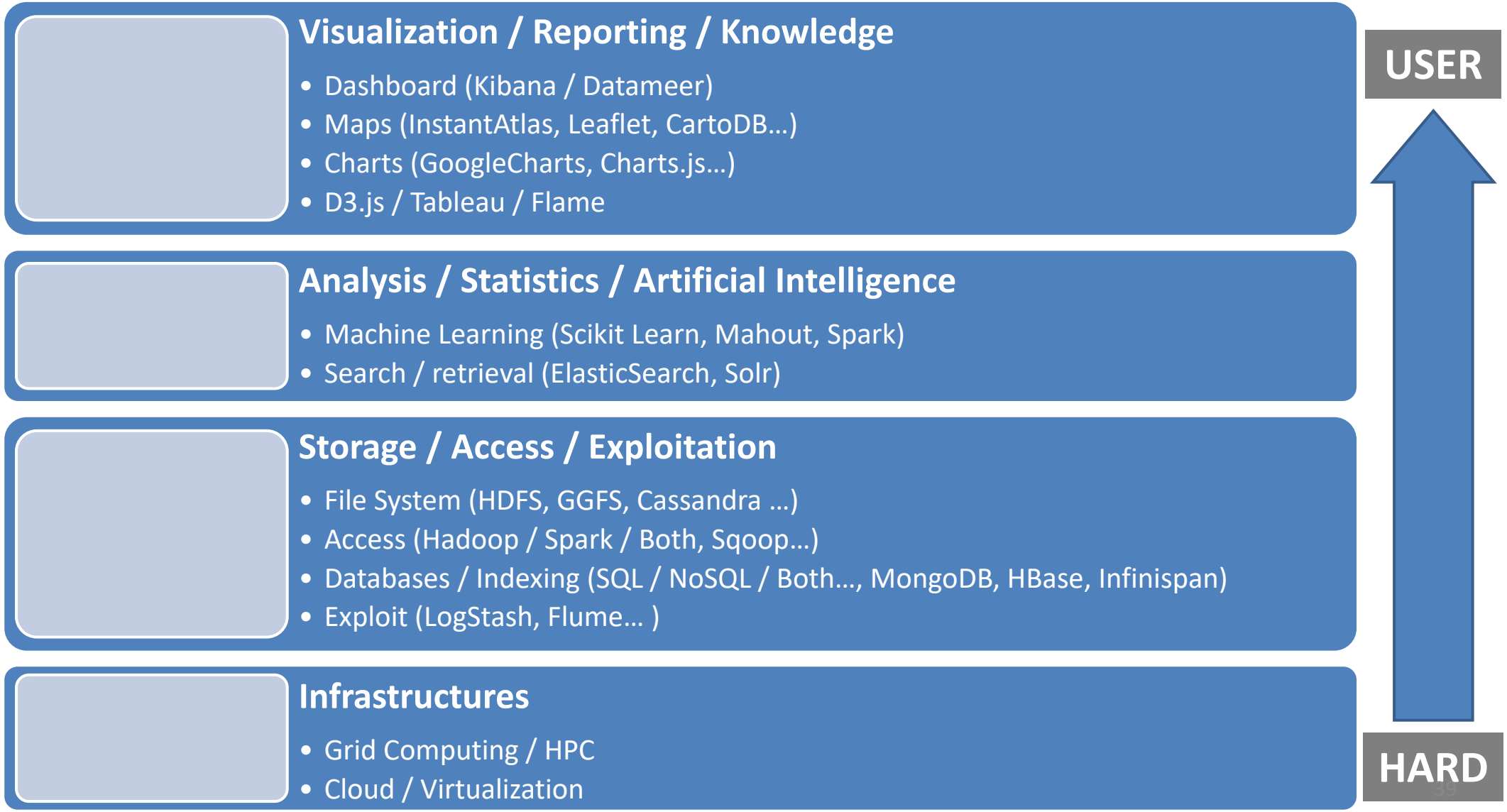
UNIVERSITÉ CÔTE D'AZUR

Machine learning VS Data science?

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Data Science Stack

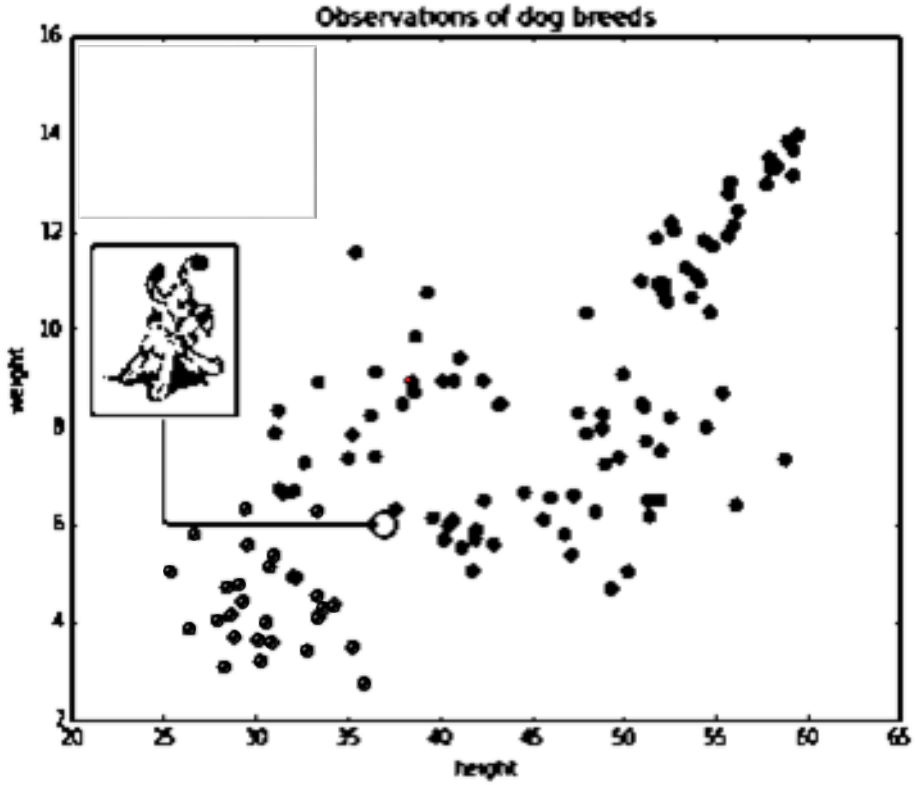


Machine Learning is Statistics?

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

What breed is that Dogmatix (Idéfix)?

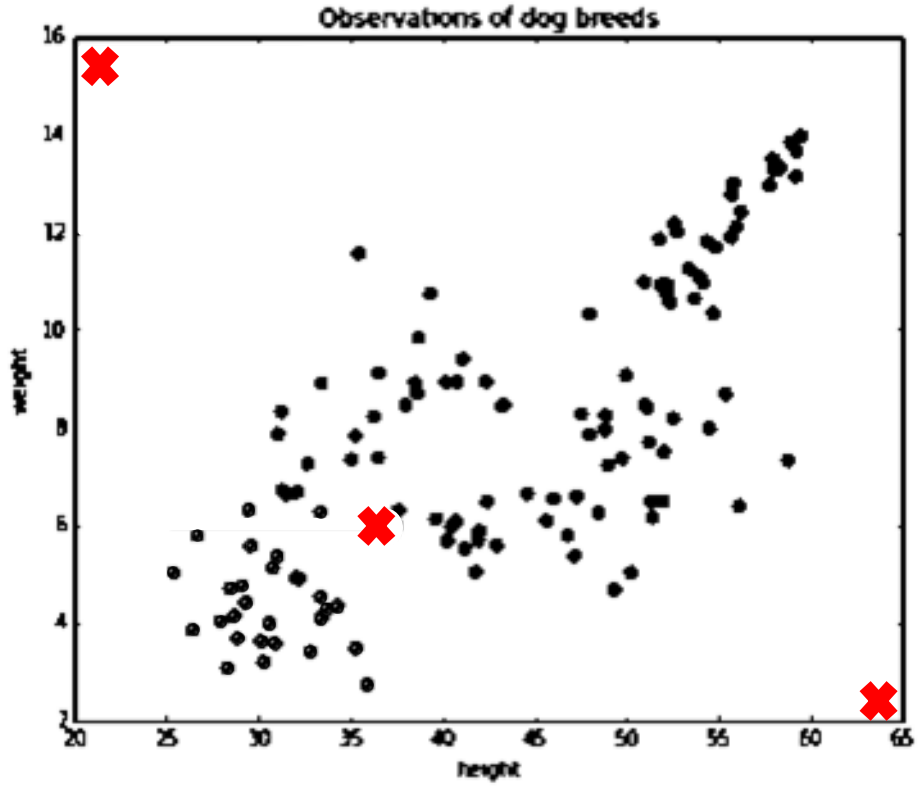


The illustrations of the slides in this section come from the blog "Bayesian Vitalstatistix: What Breed of Dog was Dogmatix?"

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Does any real dog get this height and weight?

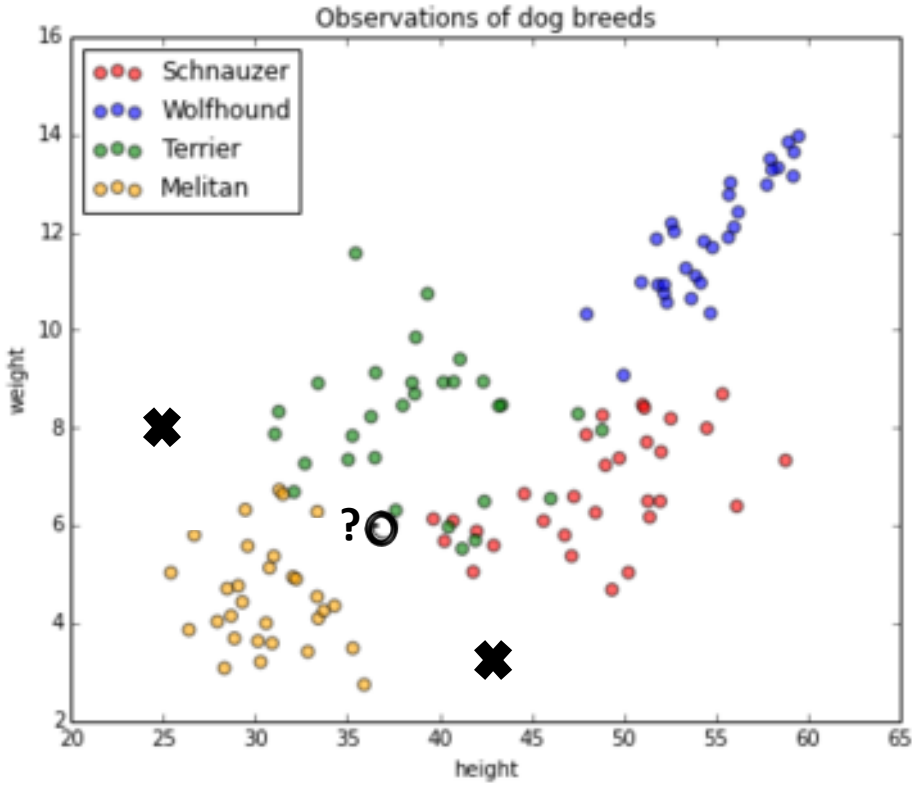


- Let us consider ***x***, vectors independently generated in $\boldsymbol{R}^d$ (here $\boldsymbol{R}^2$), following a probability distribution fixed but ***unknown*** $P(\boldsymbol{x})$.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

What should be the breed of these dogs?

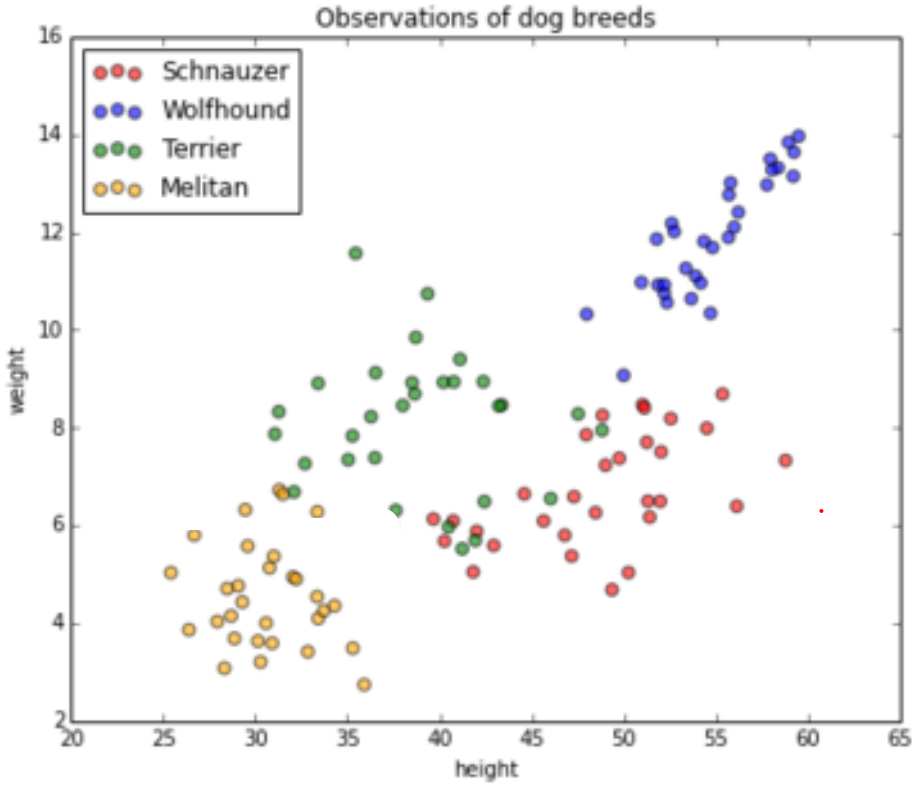


- An Oracle assigns a value *y* to each vector ***x*** following a probability distribution *P(y|**x**)* also fixed but ***unknown***.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

An oracle provides me with examples?

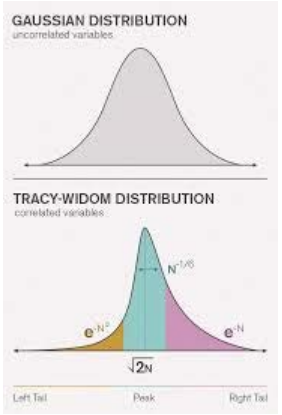


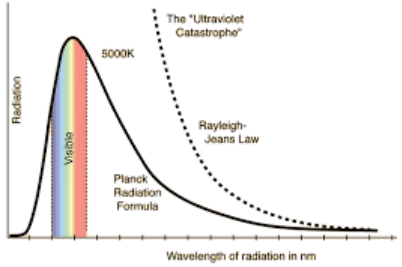
- Let S be a training set $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, \boldsymbol{y}_m)\}$, with m **training samples i.i.d.** which follow the **joint probability** $P(\boldsymbol{x}, y) = P(\boldsymbol{x})P(y|\boldsymbol{x})$.

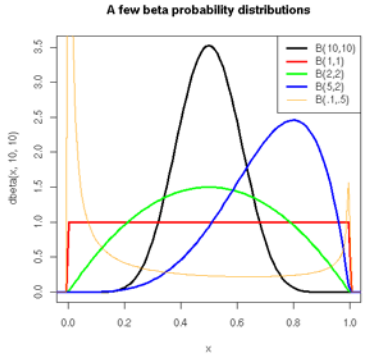
UNIVERSITÉ CÔTE D'AZUR

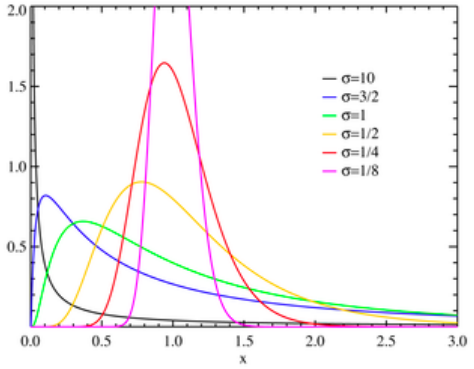
univ-cotedazur.fr

Statistical solution: Models, Hypotheses on data distribution…





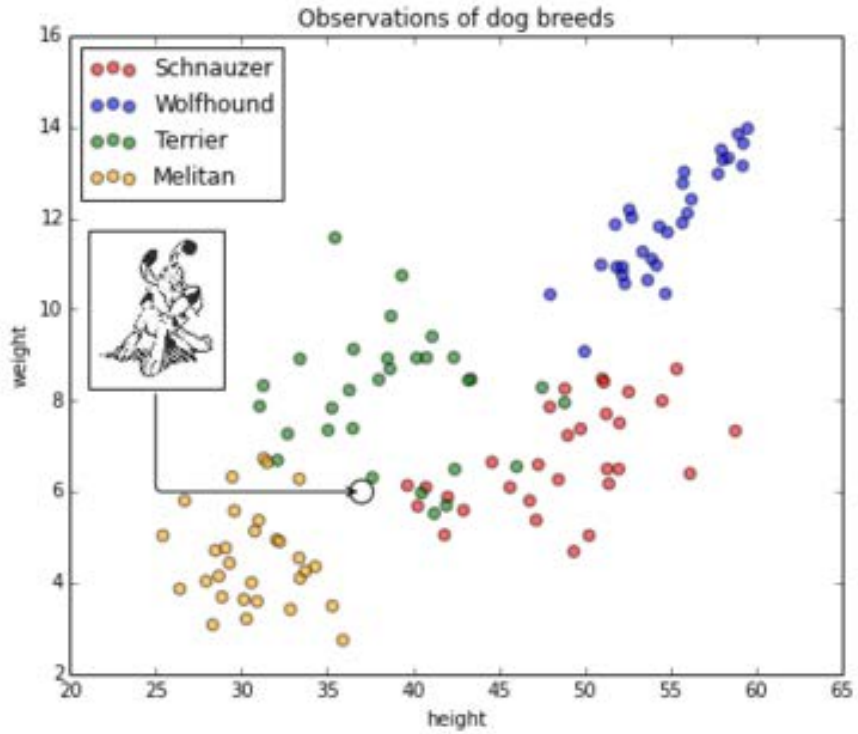


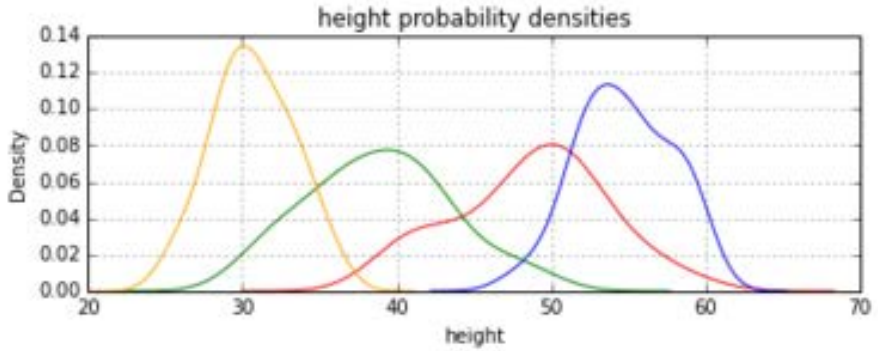


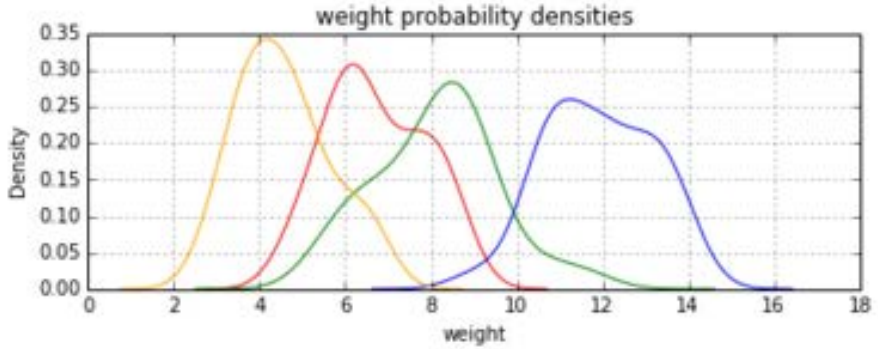
UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Statistical solution P(height, weight|breed)…



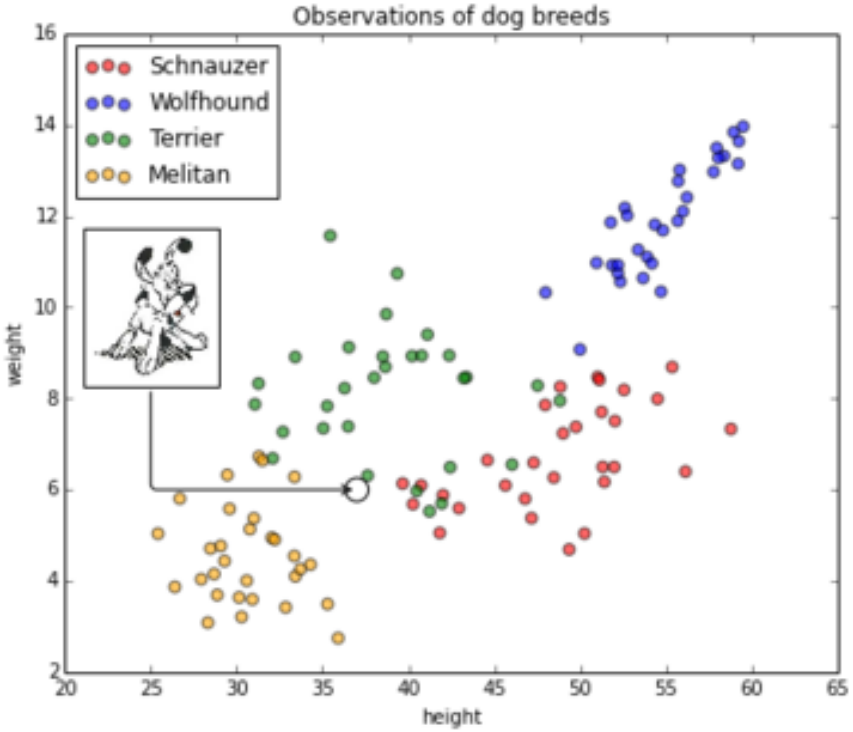


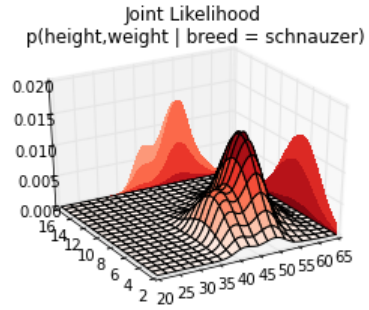


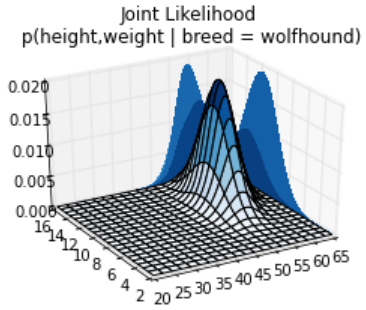
UNIVERSITÉ CÔTE D'AZUR

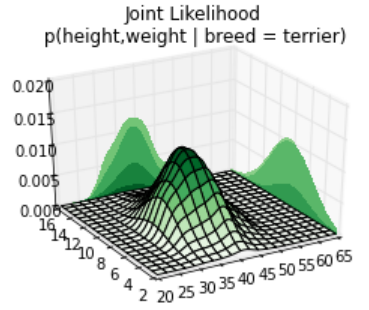
univ-cotedazur.fr

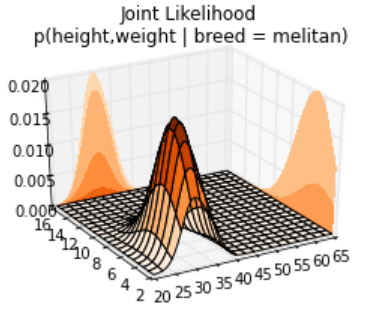
Statistical solution P(height, weight|breed)…







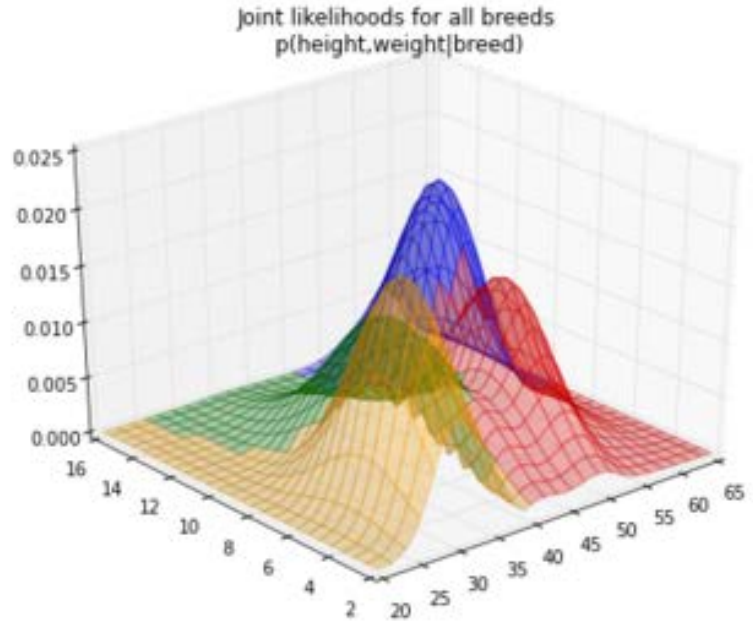


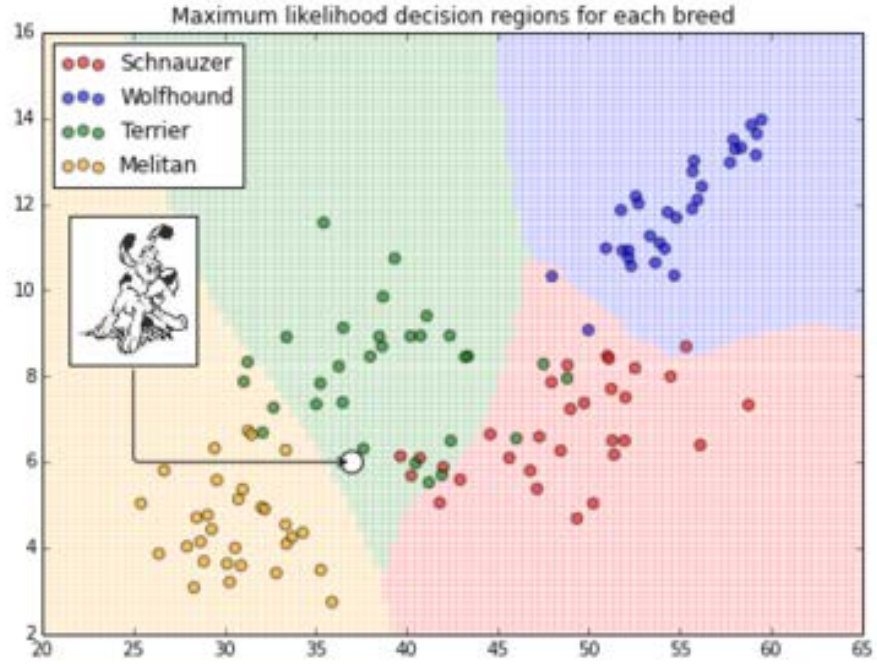


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Statistical solution P(height,weight|breed)…

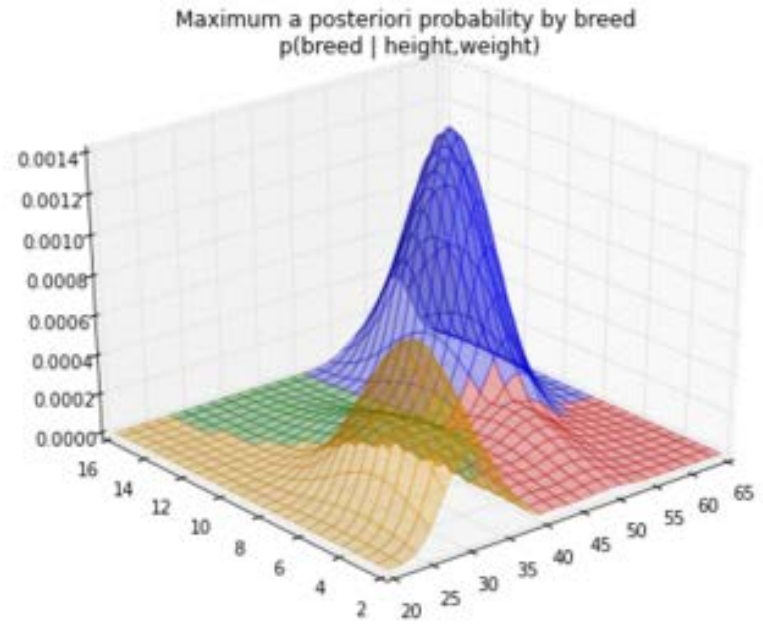


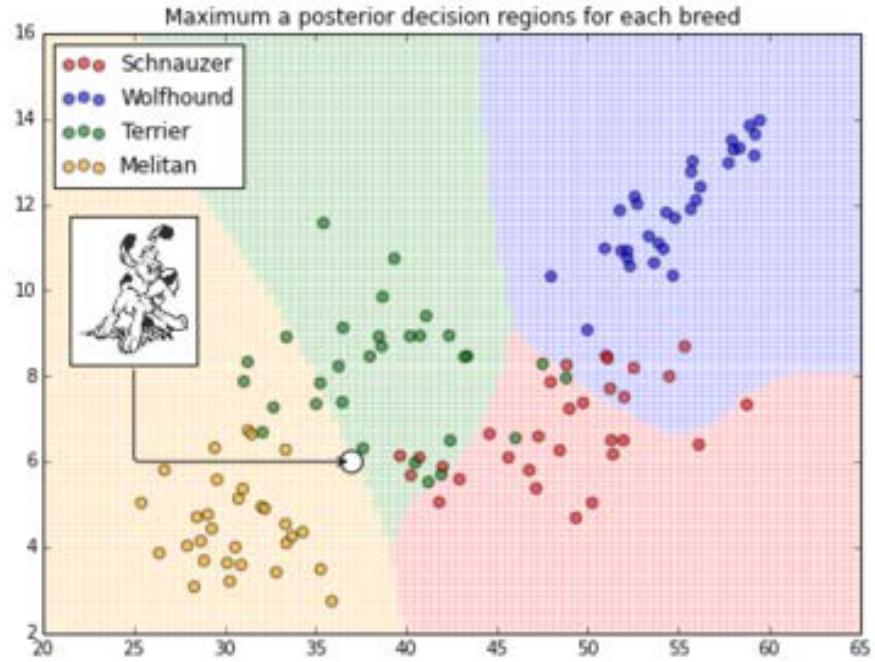


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Statistical solution: Bayes, P(breed|height, weight)…

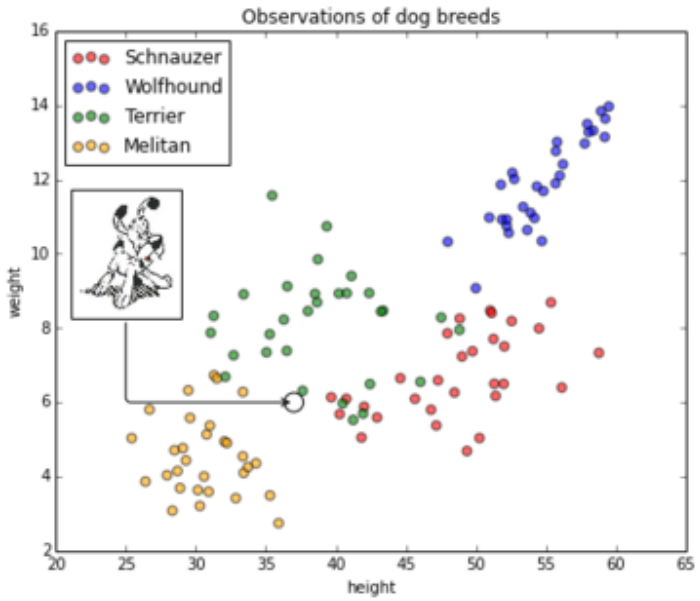




UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Machine Learning



- we have a learning machine (i.e. an algorithm) which can provide a family of functions *{f(**x**;α)}*, where *α* is a set of parameters.

$$\left(\mathbf{x}\right) \xrightarrow{f(\mathbf{X},\boldsymbol{\alpha})\ ?} y$$

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

The problem of (Machine) Learning

$$\left(\mathbf{x}\right) \xrightarrow{f(\mathbf{X},\boldsymbol{\alpha})\ ?} y$$

- The problem of learning consists in finding the function (among the $\{f(\boldsymbol{x};\alpha)\}$) which provides the best approximation $\hat{y}$ of the true label y given by the Oracle.

- "***best***" is defined in terms of minimizing a specific *error measure/cost/loss* ***related to your problem/objectives***

  $\boldsymbol{L}((\boldsymbol{x}, y), \alpha) \in [a; b]$.

UNIVERSITÉ CÔTE D'AZUR
univ-cotedazur.fr

The problem of (Machine) Learning

- Thus, the objective is to minimize the *(real)* ***Risk***, i.e. the expectation of the error cost:

$$R(\alpha) = \int \boldsymbol{L}((\boldsymbol{x}, y), \alpha) dP(\boldsymbol{x}, y)$$

where P(**x**, y) is unknown.

- The training set $\mathbf{S} = \{(\mathbf{x}_i, y_i)\}_{i=1,\dots,m}$ is built through an i.i.d. sampling according to P(**x** ,y). Since we cannot compute $R(\alpha)$, we look for minimizing the ***Empirical Risk*** instead:

$$R_{emp}(\alpha) = \frac{1}{m} \sum_{k=1}^{m} \boldsymbol{L}((\boldsymbol{x}_i, y_i), \alpha)$$

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Machine Learning fundamental Hypothesis

$S = \{(x_i, y_i)\}_{i=1,\dots,m}$ is built through an *i.i.d.* sampling according to $P(x, y)$.

Machine Learning $\Leftrightarrow$ ***Statistics***

Train through Cross-Validation

Machine Learning $\Leftrightarrow$ ***Statistics***

Training set & Test set have to be distributed according to the same law

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Vapnik learning theory (1995)

Vapnik had proven the following equation $\forall m$ with a probability at least equal to $1 - \eta$:

$$R(\alpha_m) \leq R_{emp}(\alpha_m) + (b - a)\sqrt{\frac{d_{VC}(ln(2m/d_{VC}) + 1) - ln(\eta/4)}{m}}$$

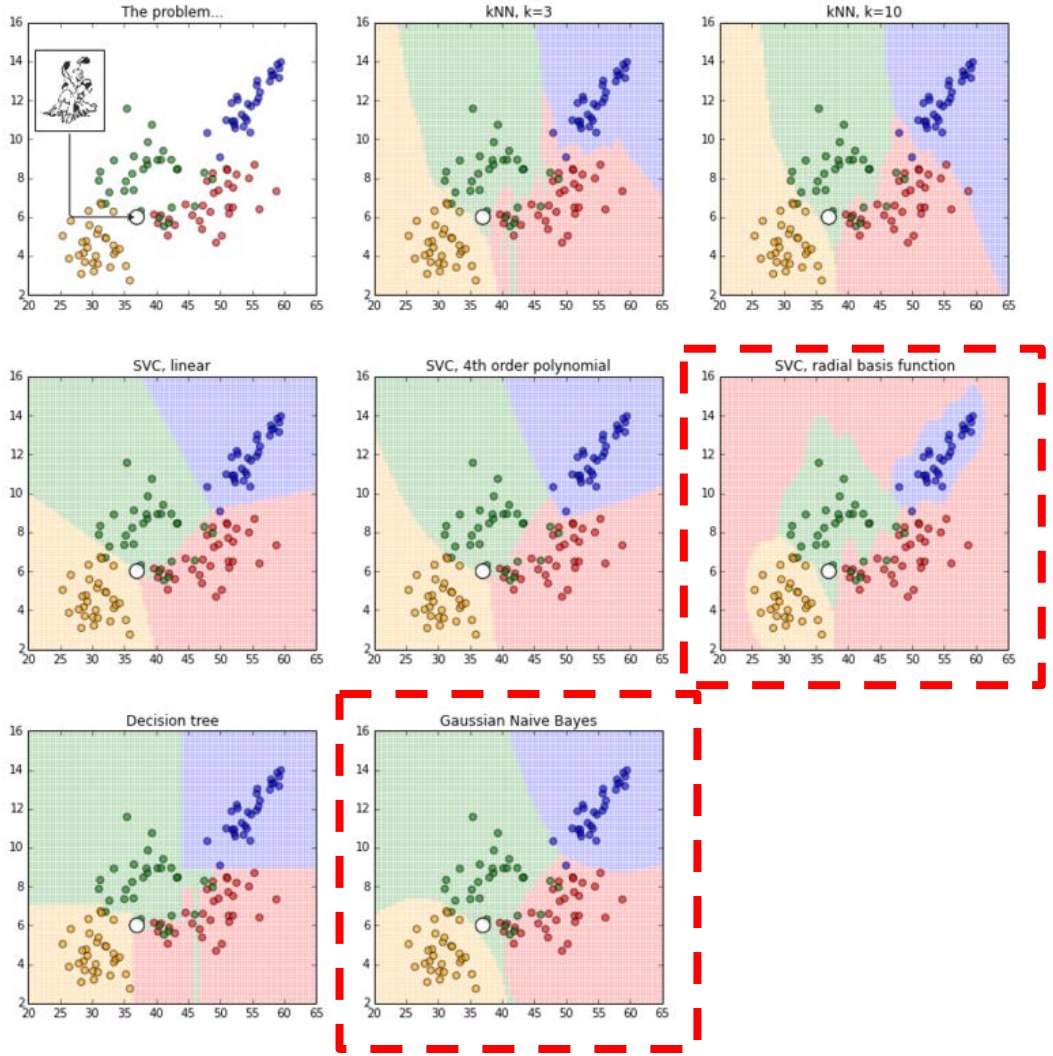
Training Error ***Generalization Error***

Thus minimizing the ***Risk*** depends on minimzing the ***Empirical Risk*** and the ***Generalization Error*** of the model which depends on *m* (the number of training sample), and d_{VC} (the complexity of the model family chosen, also called *Vapnik-Chervonenkis Dimension).*

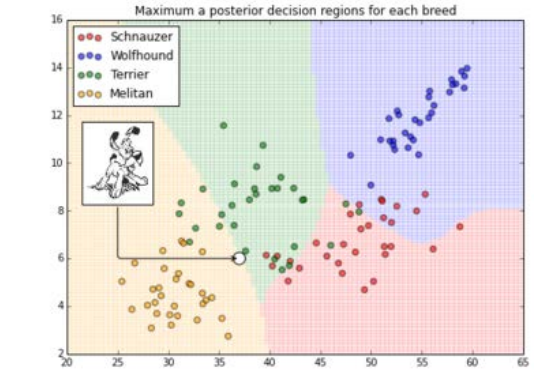
UNIVERSITÉ CÔTE D'AZUR

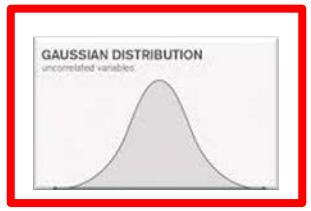
univ-cotedazur.fr

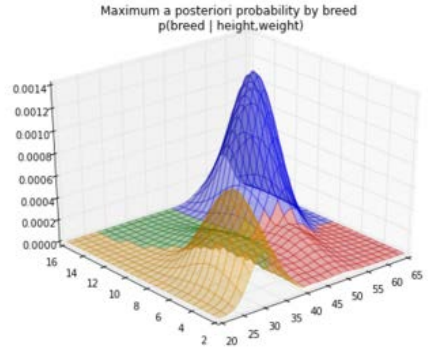
Machine Learning vs Statistics

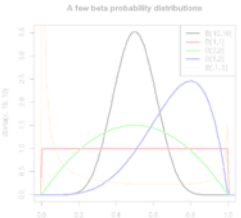


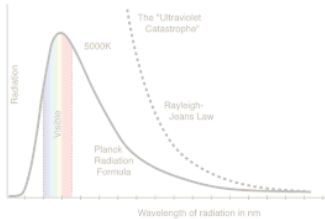
VS











UNIVERSITÉ CÔTE D'AZUR
univ-cotedazur.fr

Machine Learning vs Statistics

- Machine Learning and Statistics can be considered as two sides of the same coin: Statistical Learning
- In Statistics, you make explicit assumptions on the model driving the data
- In Machine Learning, you do implicit Statistics; you make very few very general assumptions on the data but no assumptions on the model itself

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

MATHEMATICAL BASICS

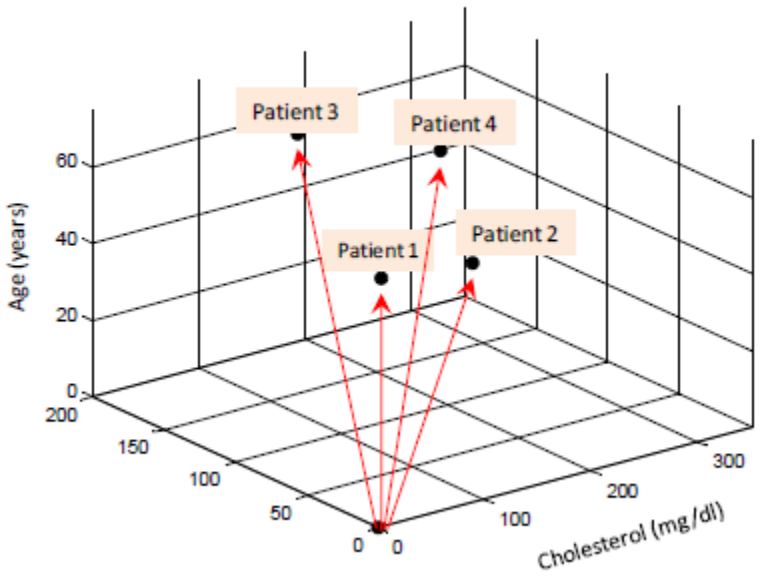
UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

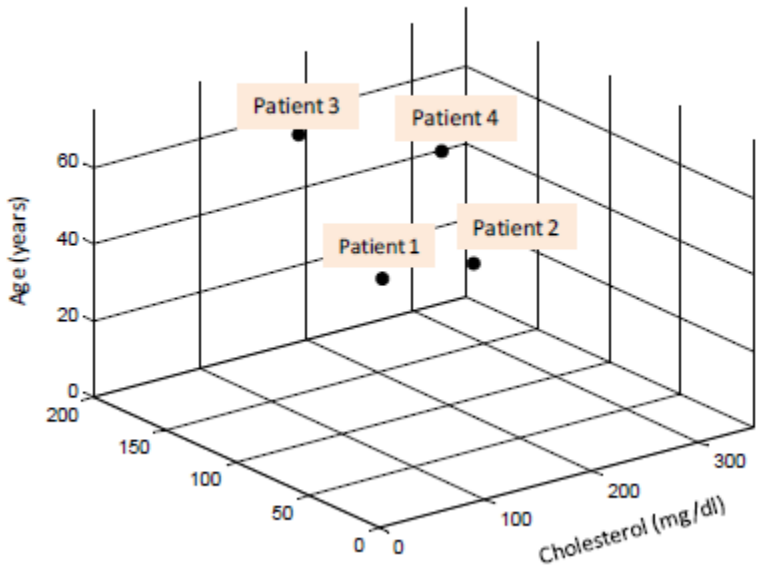
How to represent samples geometrically? Vectors & points in n-dimensional space ($\mathbb{R}^n$)

Patient id	*Cholesterol* (mg/dl)	*Systolic BP* (mmHg)	*Age* (years)	**Tail of the vector**	**Arrow-head of the vector**
1	150	110	35	(0,0,0)	(150, 110, 35)
2	250	120	30	(0,0,0)	(250, 120, 30)
3	140	160	65	(0,0,0)	(140, 160, 65)
4	300	180	45	(0,0,0)	(300, 180, 45)

Vector representation



Point representation



From "A Gentle Introduction to Support Vector Machines in Biomedicine", A. Statnikov, et al, AMIA 2010.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Basic operation on vectors in $\mathbb{R}^n$

1. Multiplication by a scalar

Consider a vector $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ and a scalar c

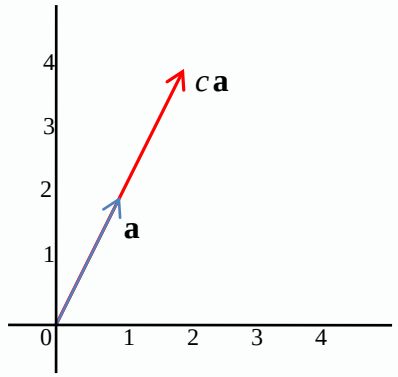
Define: $c\,\mathbf{a} = (ca_1, ca_2, \ldots, ca_n)$

When you multiply a vector by a scalar, you "stretch" it in the same or opposite direction depending on whether the scalar is positive or negative.

$\mathbf{a} = (1, 2)$

$c = 2$

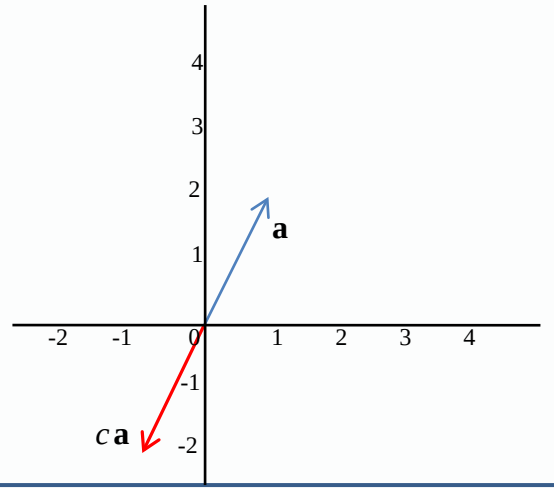
$c\,\mathbf{a} = (2, 4)$



$\mathbf{a} = (1, 2)$

$c = -1$

$c\,\mathbf{a} = (-1, -2)$



Basic operation on vectors in $\mathbb{R}^n$

2. Addition

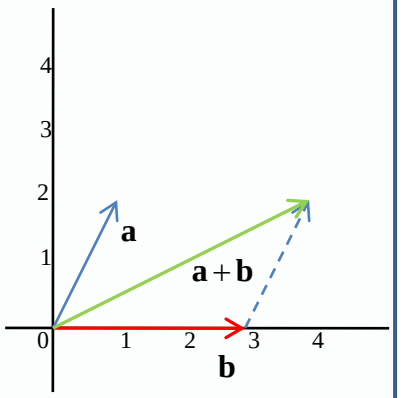
Consider vectors $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ and $\mathbf{b} = (b_1, b_2, \ldots, b_n)$

Define: $\mathbf{a} + \mathbf{b} = (a_1 + b_1, a_2 + b_2, \ldots, a_n + b_n)$

$\mathbf{a} = (1, 2)$

$\mathbf{b} = (3, 0)$

$\mathbf{a} + \mathbf{b} = (4, 2)$



Recall addition of forces in classical mechanics.

From "A Gentle Introduction to Support Vector Machines in Biomedicine", A. Statnikov, et al, AMIA 2010.

Basic operation on vectors in $\mathbb{R}^n$

3. Subtraction

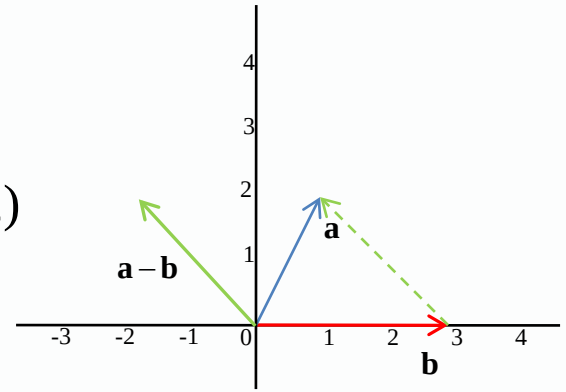
Consider vectors $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ and $\mathbf{b} = (b_1, b_2, \ldots, b_n)$

Define: $\mathbf{a} - \mathbf{b} = (a_1 - b_1, a_2 - b_2, \ldots, a_n - b_n)$

$\mathbf{a} = (1, 2)$

$\mathbf{b} = (3, 0)$

$\mathbf{a} - \mathbf{b} = (-2, 2)$



What vector do we need to add to $\vec{b}$ to get $\vec{a}$? I.e., similar to subtraction of real numbers.

From "A Gentle Introduction to Support Vector Machines in Biomedicine", A. Statnikov, et al, AMIA 2010.

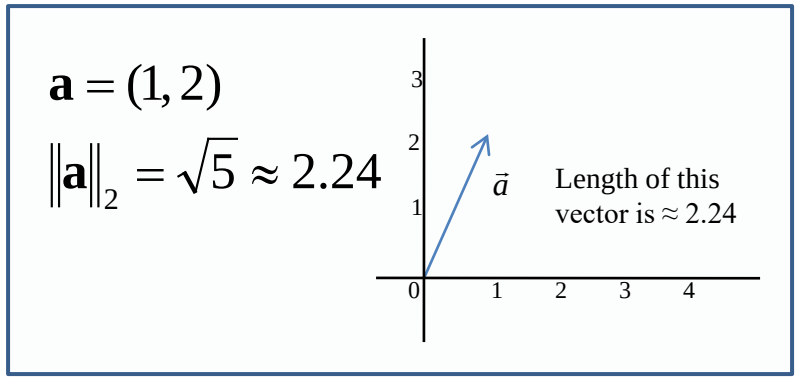
Basic operation on vectors in $\mathbb{R}^n$

4. Euclidian length or L2-norm

Consider a vector $\mathbf{a} = (a_1, a_2, \ldots, a_n)$

Define the L2-norm: $\|\mathbf{a}\|_2 = \sqrt{a_1^2 + a_2^2 + \ldots + a_n^2}$

We often denote the L2-norm without subscript, i.e. $\|\mathbf{a}\|$



L2-norm is a typical way to measure length of a vector; other methods to measure length also exist.

From "A Gentle Introduction to Support Vector Machines in Biomedicine", A. Statnikov, et al, AMIA 2010.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Basic operation on vectors in $\mathbb{R}^n$

5. Dot product

Consider vectors $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ and $\mathbf{b} = (b_1, b_2, \ldots, b_n)$

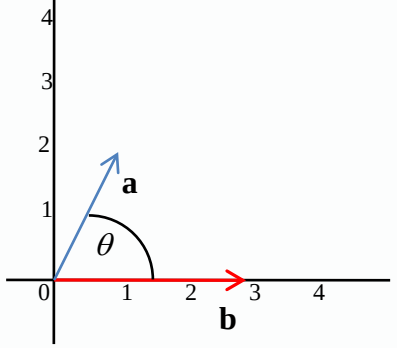
Define dot product: $\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 + \ldots + a_n b_n = \sum_{i=1}^{n} a_i b_i$

The law of cosines says that $\mathbf{a} \cdot \mathbf{b} = \| \mathbf{a} \|_2 \| \mathbf{b} \|_2 \cos\theta$ where θ is the angle between $\mathbf{a}$ and $\mathbf{b}$. Therefore, when the vectors are perpendicular $\mathbf{a} \cdot \mathbf{b} = 0$.

$\mathbf{a} = (1, 2)$

$\mathbf{b} = (3, 0)$

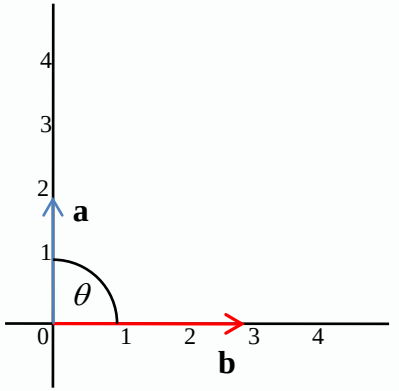
$\mathbf{a} \cdot \mathbf{b} = 3$



$\mathbf{a} = (0, 2)$

$\mathbf{b} = (3, 0)$

$\mathbf{a} \cdot \mathbf{b} = 0$



Basic operation on vectors in $\mathbb{R}^n$

5. Dot product (continued)

$$\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 + \ldots + a_n b_n = \sum_{i=1}^{n} a_i b_i$$

- Property: $\mathbf{a} \cdot \mathbf{a} = a_1 a_1 + a_2 a_2 + \ldots + a_n a_n = \|\mathbf{a}\|_2^2$
- In the classical regression equation $y = \mathbf{w} \cdot \mathbf{x} + b$

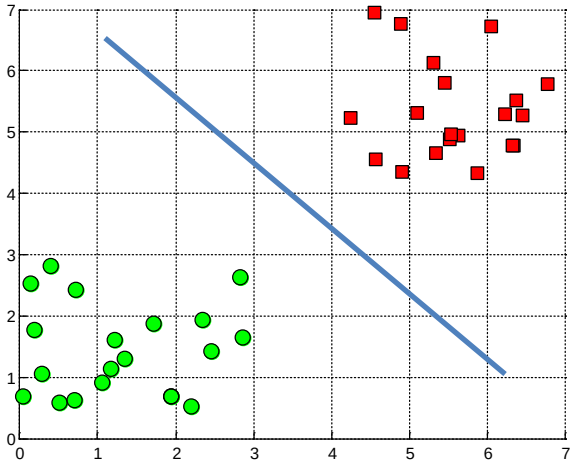
  the response variable *y* is just a dot product of the vector representing patient characteristics ($\mathbf{x}$) and the regression weights vector ($\mathbf{w}$) which is common across all patients plus an offset b.

From "A Gentle Introduction to Support Vector Machines in Biomedicine", A. Statnikov, et al, AMIA 2010.

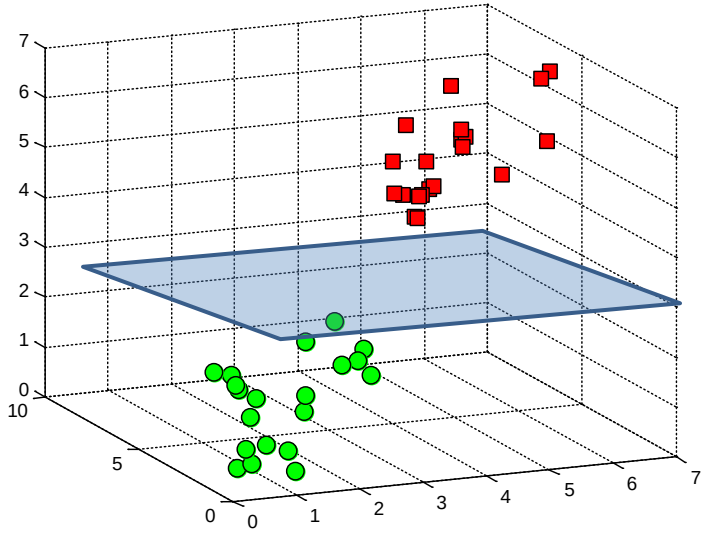
Hyperplanes as decision surfaces

- A hyperplane is a linear decision surface that splits the space into two parts;
- It is obvious that a hyperplane is a binary classifier.

A hyperplane in $\mathbb{R}^2$ is a line



A hyperplane in $\mathbb{R}^3$ is a plane

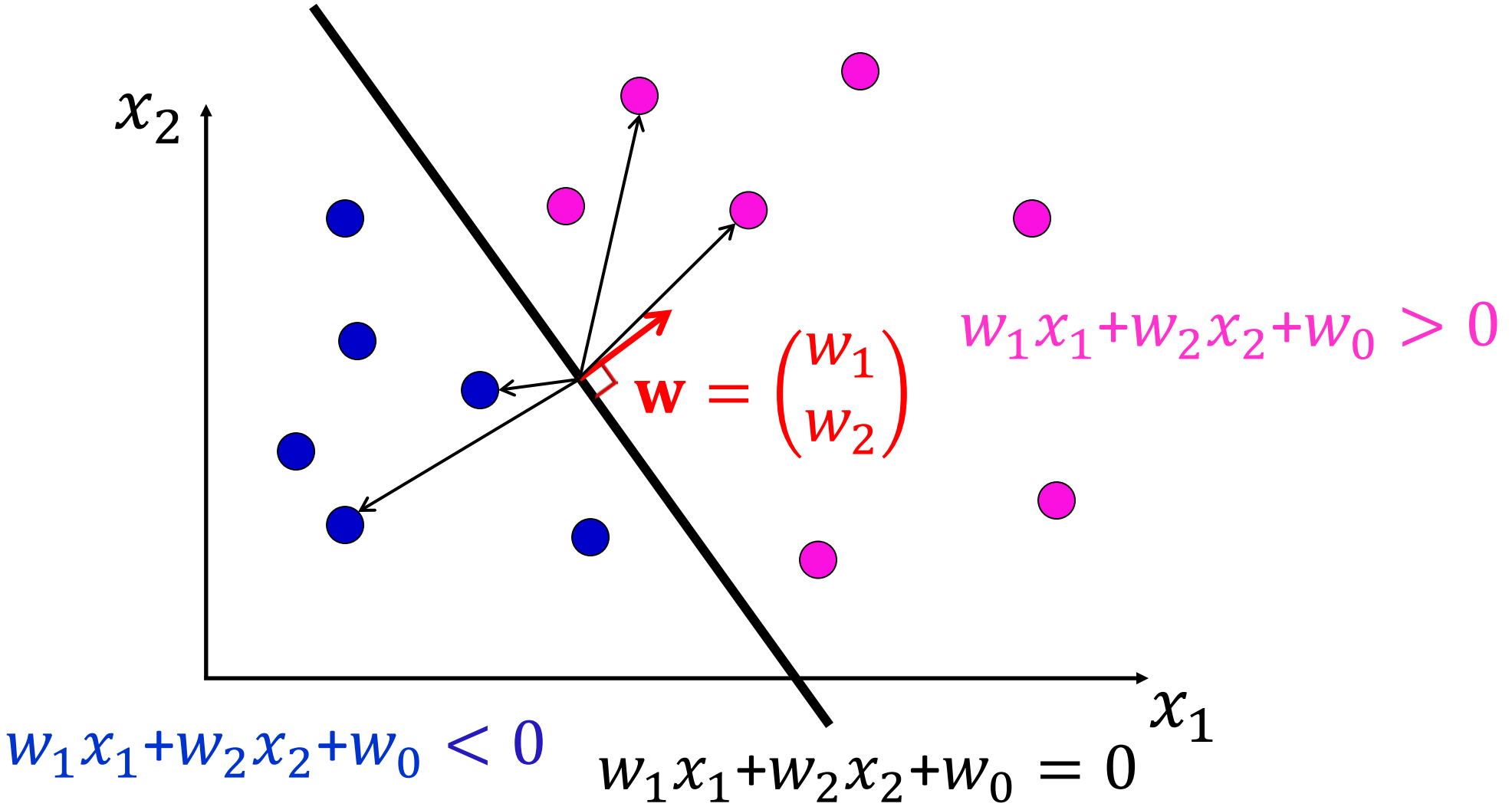


A hyperplane in $\mathbb{R}^n$ is an n-1 dimensional subspace

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

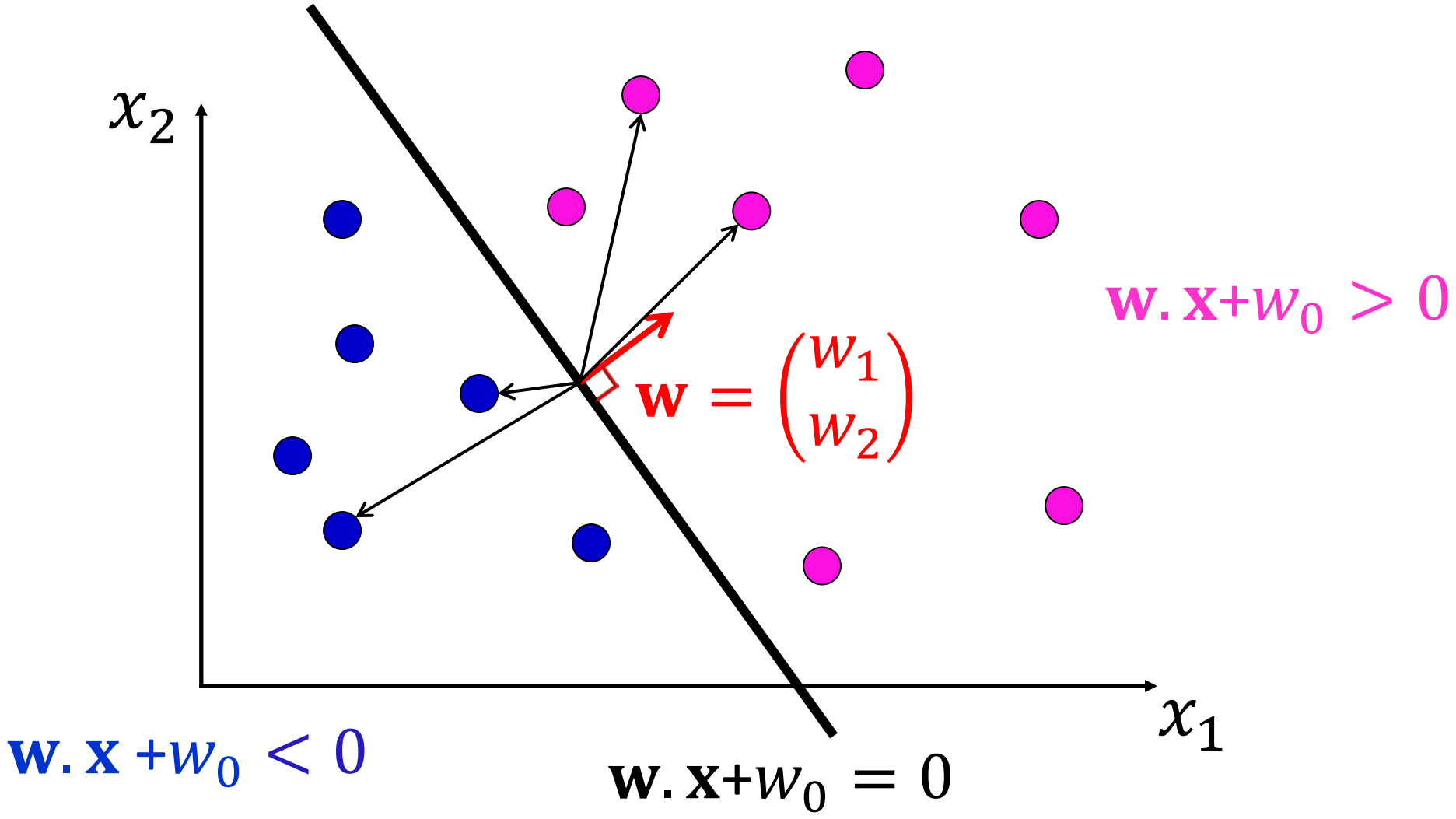
Geometry and Algebra



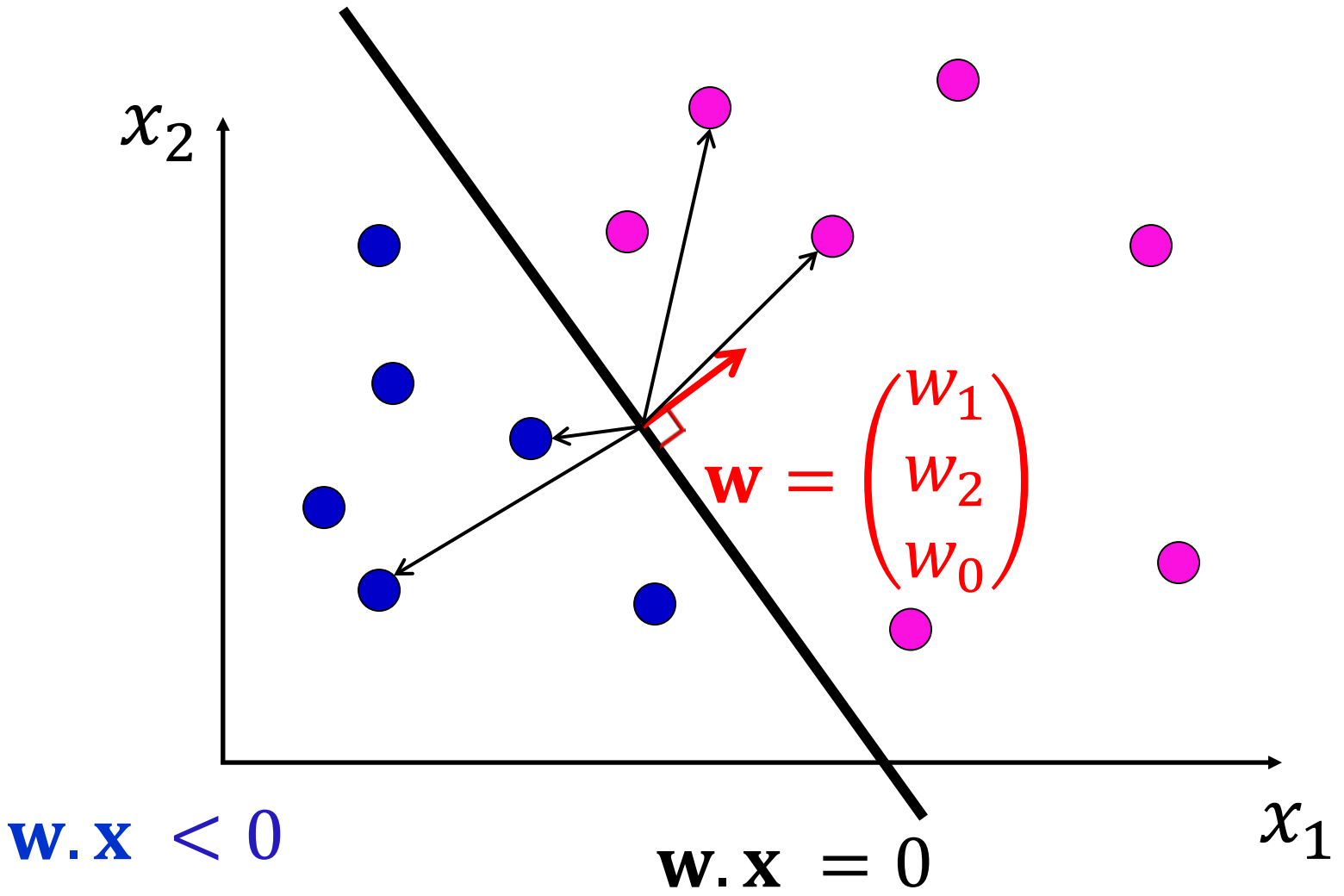
UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Geometry and Algebra



Simplification



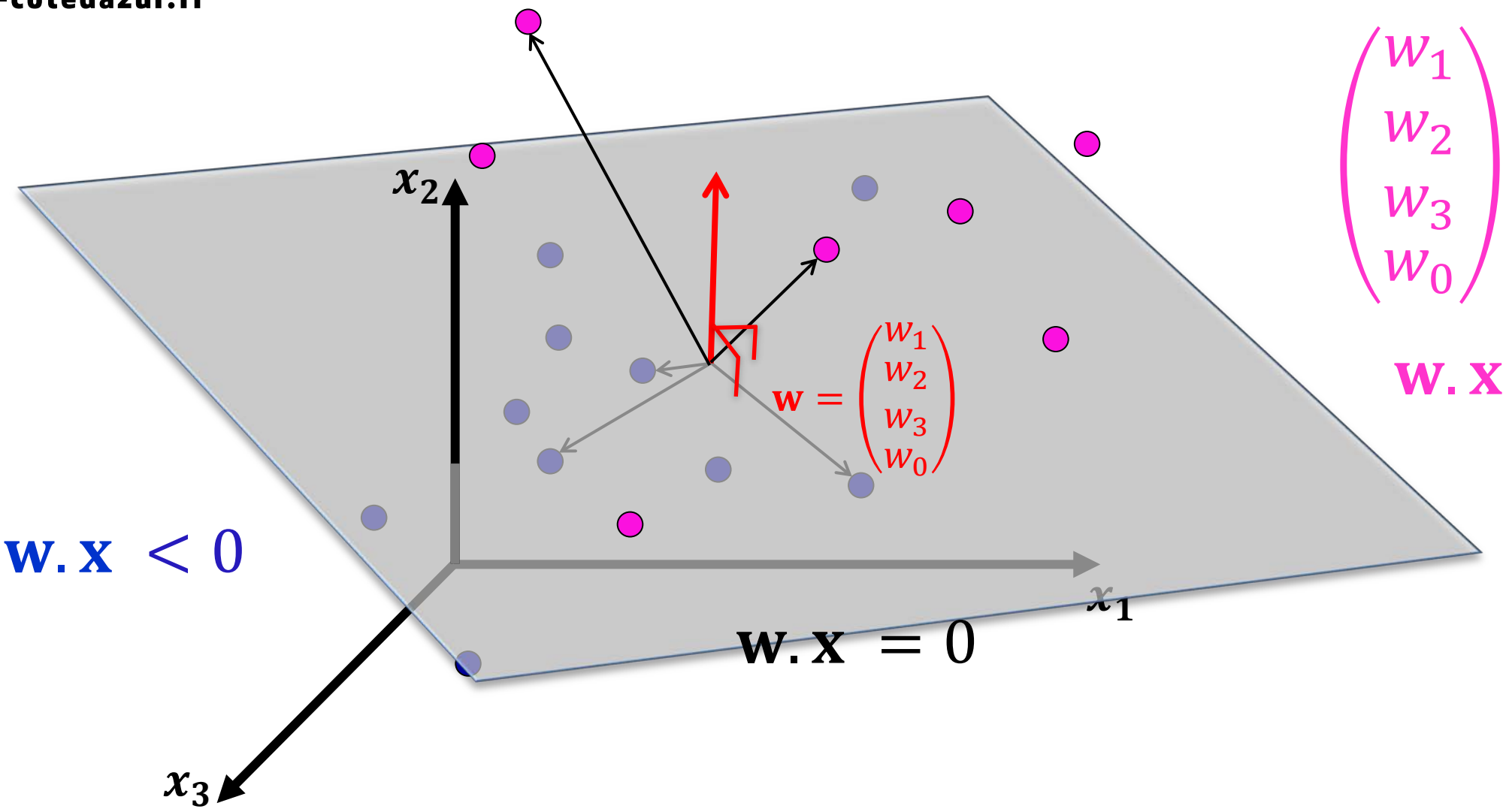
$$\begin{pmatrix} w_1 \\ w_2 \\ w_0 \end{pmatrix} . \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} > 0$$

$$\mathbf{w}.\mathbf{x} > 0$$

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Geometry and Algebra



$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} > 0$$

$$\mathbf{w}.\mathbf{x} > 0$$

Derivative

- Derivative of a scalar function (of one variable)
  - $(ax)' = a$
  - $(ax + b)' = a$
  - $(\mathrm{g}(\mathrm{f}(\mathrm{x})))' = \mathrm{g}'(\mathrm{f}(\mathrm{x}))\mathrm{f}'(\mathrm{x})$ **Chain rule**

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Gradient operator

- Gradient of a multivariate function ($\mathbf{x}$ is a vector)
  - $\nabla_{\mathbf{x}}(\boldsymbol{a}\mathbf{x}) = \boldsymbol{a}$
  - $\nabla_{\mathbf{x}}(\boldsymbol{a}\mathbf{x} + b) = \boldsymbol{a}$
  - $\nabla_{\mathbf{x}}(\mathrm{g}(\mathrm{f}(\mathbf{x}))) = \nabla_{\mathbf{x}}\mathrm{g}\,(\mathrm{f}(\mathbf{x}))\,\nabla_{\mathbf{x}}\mathrm{f}(\mathbf{x})$ **(Chain rule)**

  - $\nabla_{\mathbf{x_i}}(\mathbf{v}.\mathbf{x}) = \nabla_{\mathbf{x_i}}(v_1.x_1 + v_2.x_2 + \cdots + v_n.xn) = v_i$

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Overview

- Machine Learning vs Statistics
- Math Basics
- **Simple Model**
- From Simple to Complex

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

SIMPLE MODEL

UNIVERSITÉ CÔTE D'AZUR

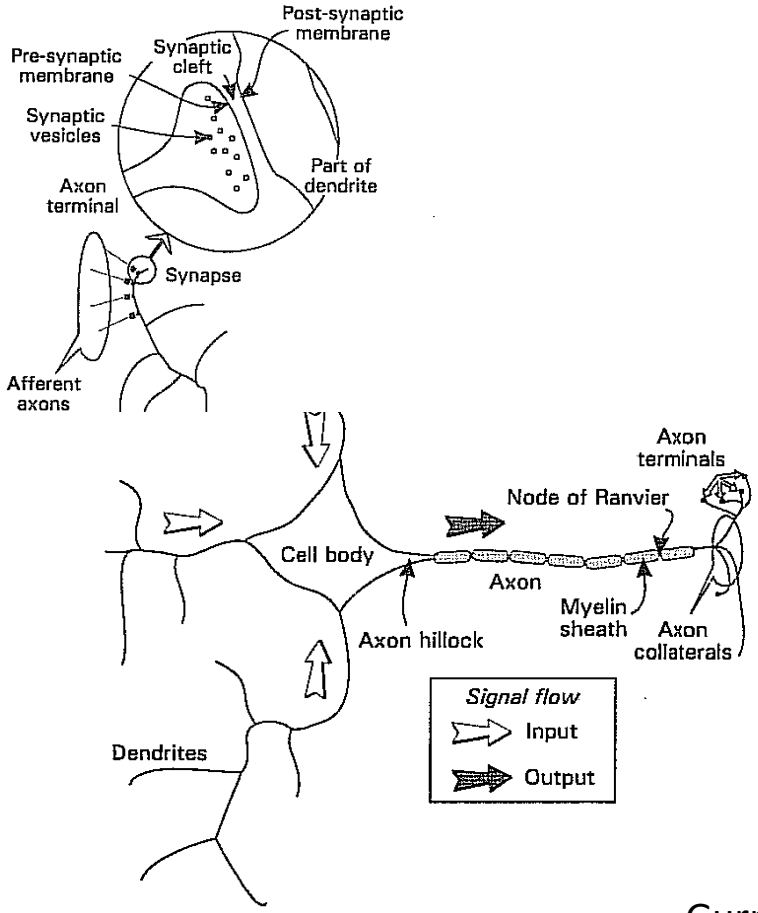
Initial Model: Perceptron

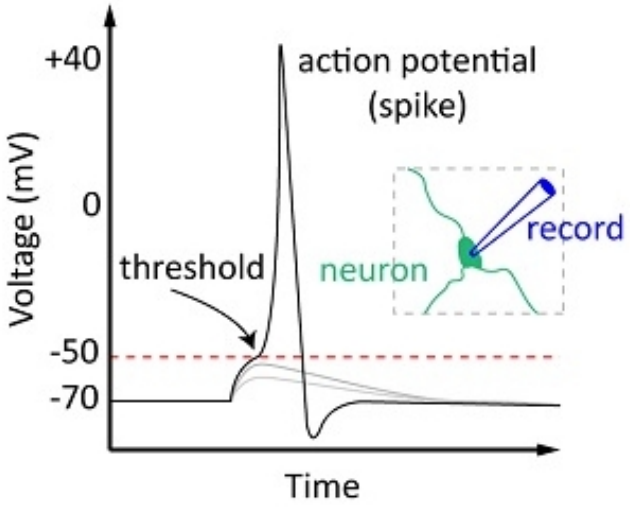
UNIVERSITÉ CÔTE D'AZUR

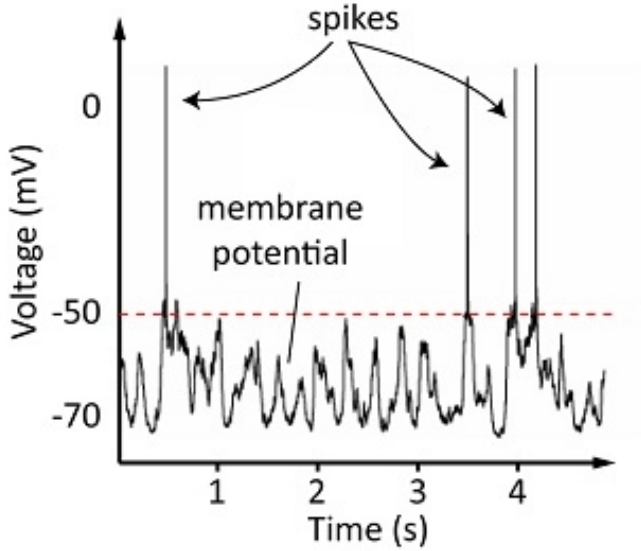
univ-cotedazur.fr

Biological neuron

- Before we study artificial neurons, let's look at a biological neuron







Gurney, K. (1997). *An introduction to neural networks*. CRC press.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

First, biological neurons

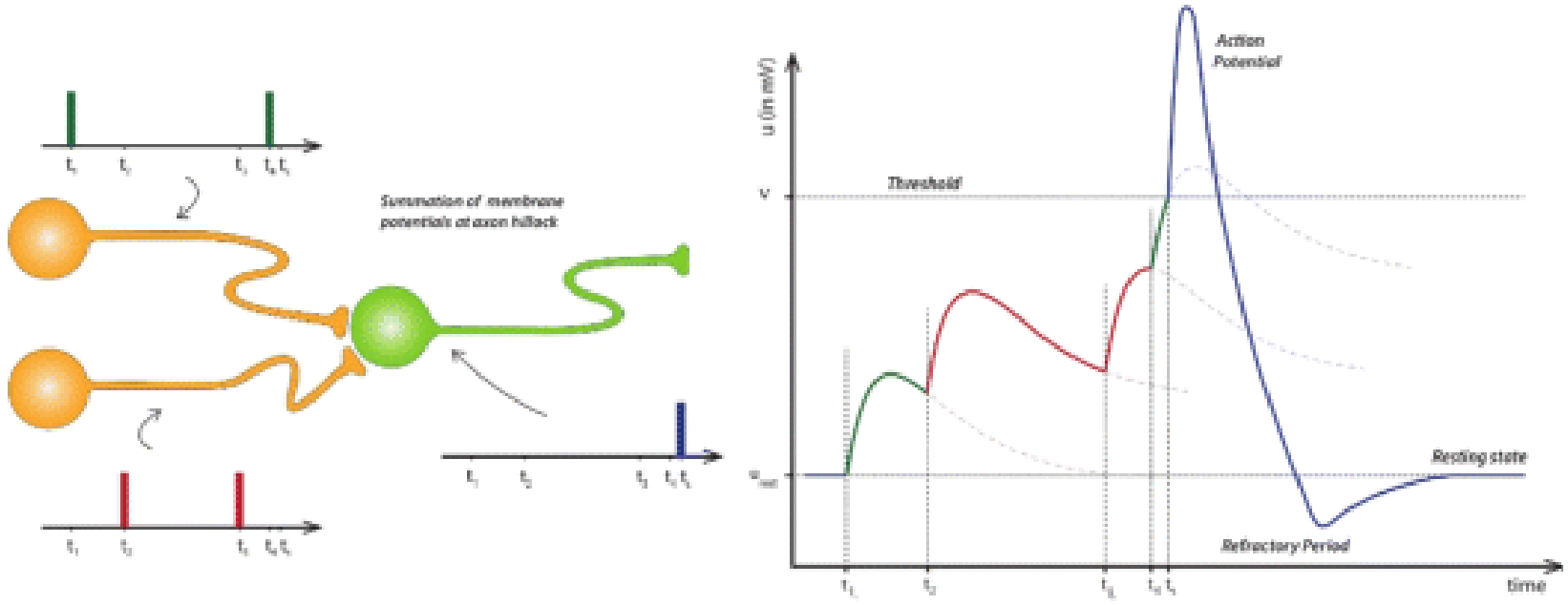
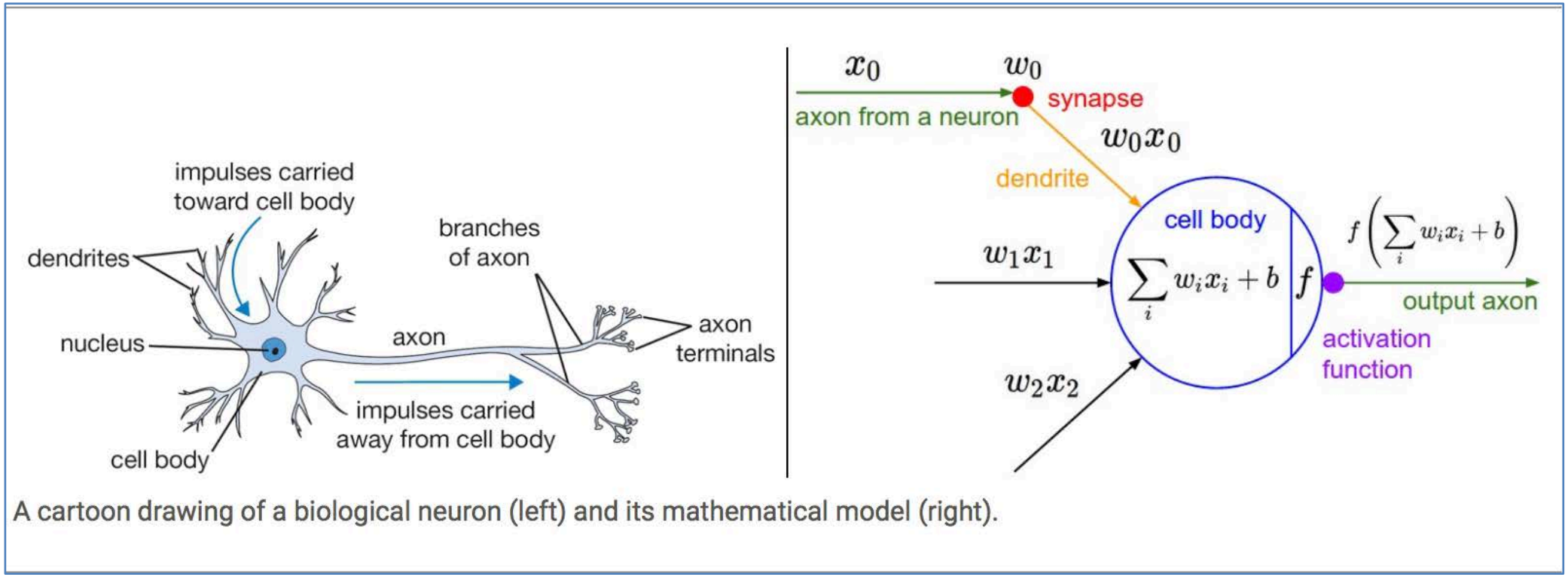


Figure from: Iakymchuk, T., et al. "Simplified spiking neural network architecture and STDP learning algorithm applied to image classification". In Journal of Image Video Proc. 2015, 4 (2015).

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Then, artificial neurons



A cartoon drawing of a biological neuron (left) and its mathematical model (right).

Pitts & McCulloch (1943), binary inputs & activation function *f* thresholding

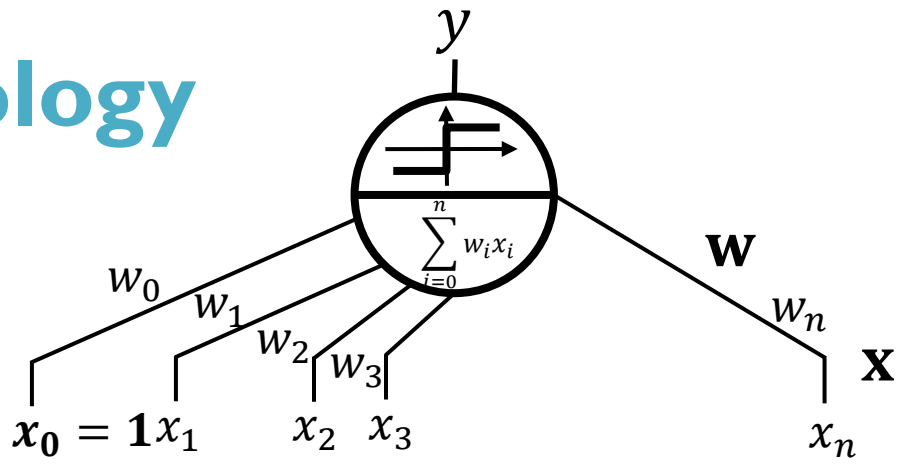
Rosenblatt (1956), real inputs & activation function *f* thresholding

(Schéma : Isaac Changhau)

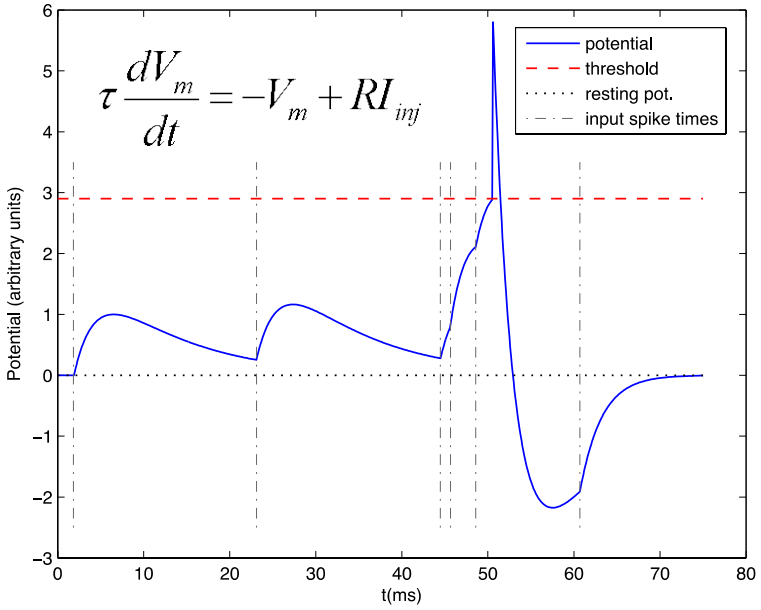
UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Artificial vs biology



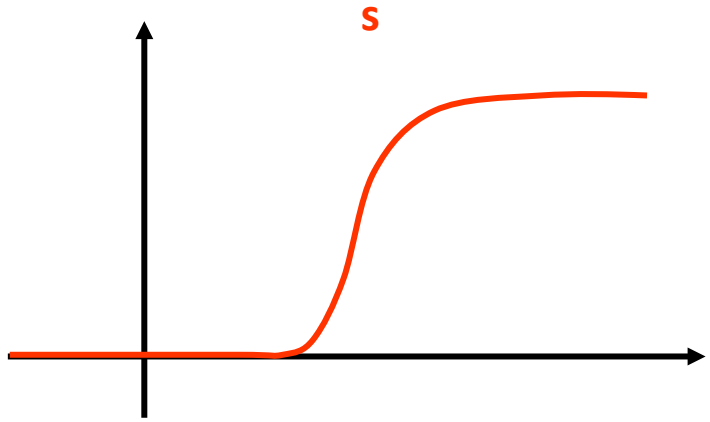
Spike-based description



Gradient descent: KO

Rate-based description

Steady regime



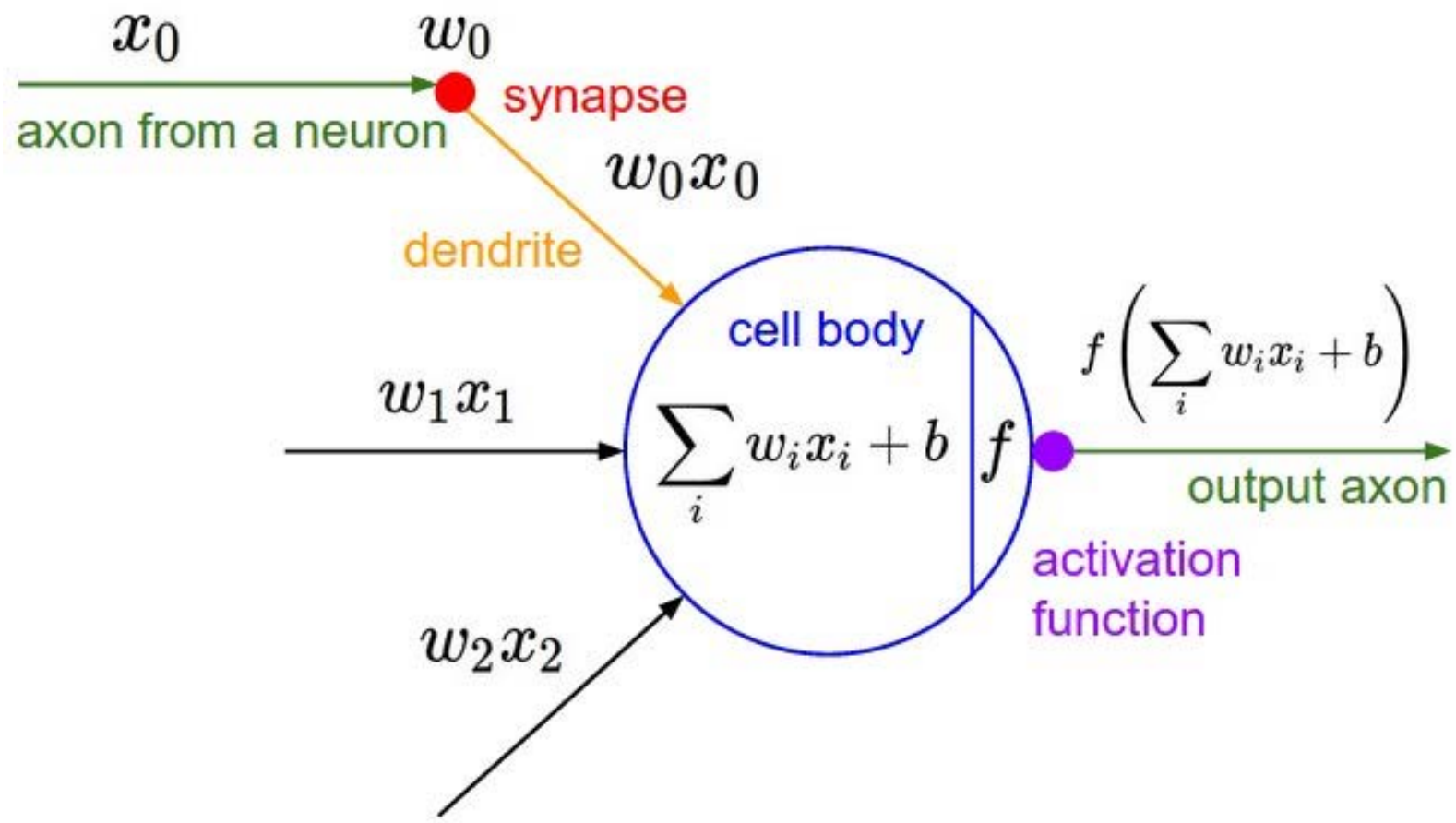
$$y = s(\Sigma w_i x_i)$$

Gradient descent: OK

UNIVERSITÉ CÔTE D'AZUR
univ-cotedazur.fr

A single artificial neuron

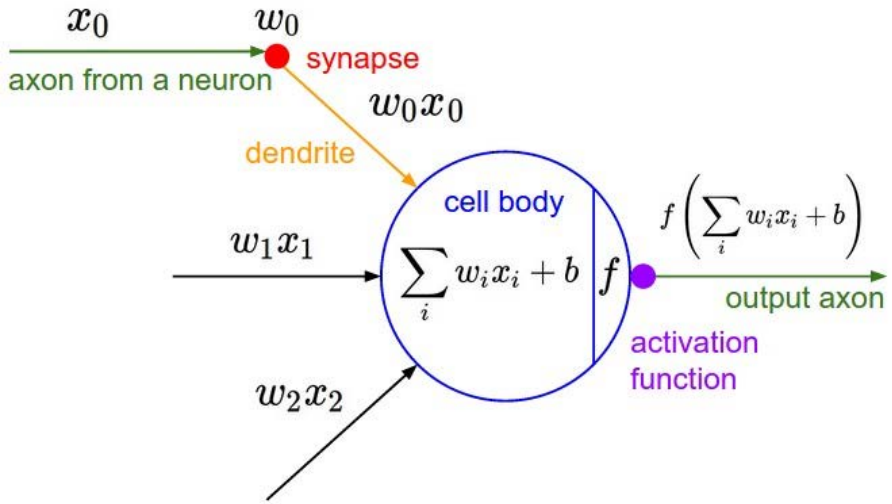
- It is a **very simple abstract** of a biological neuron (McCulloch & Pitts, 1943)



UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

A single artificial neuron

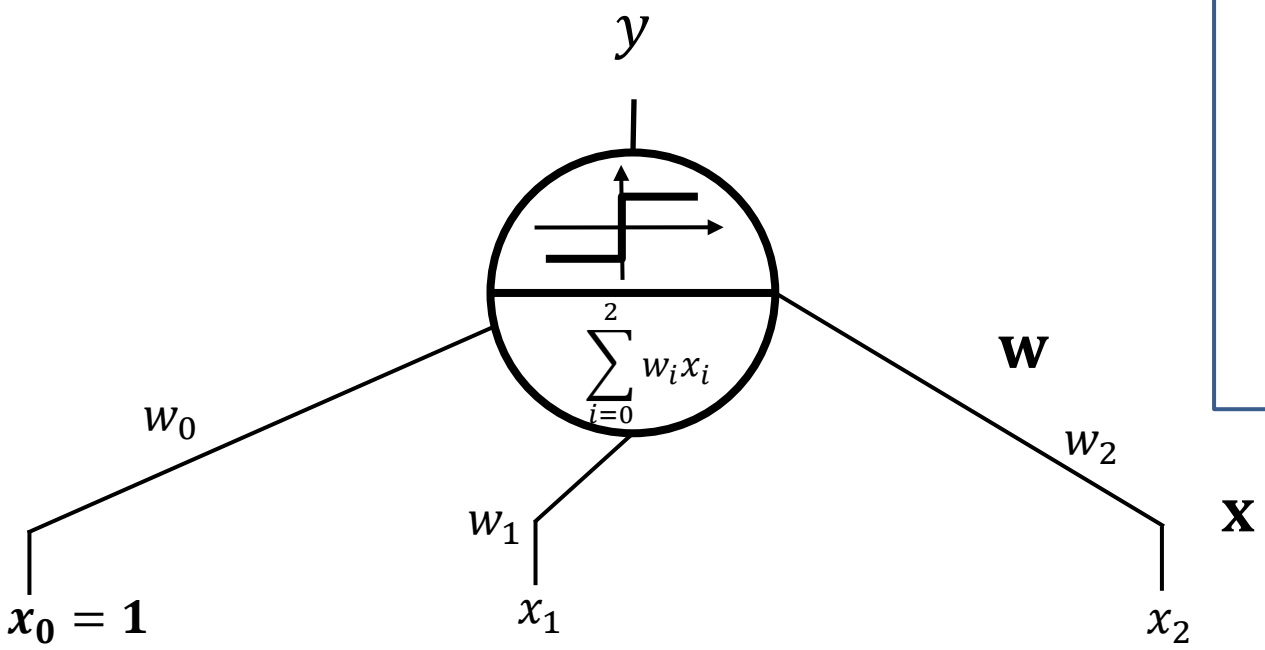


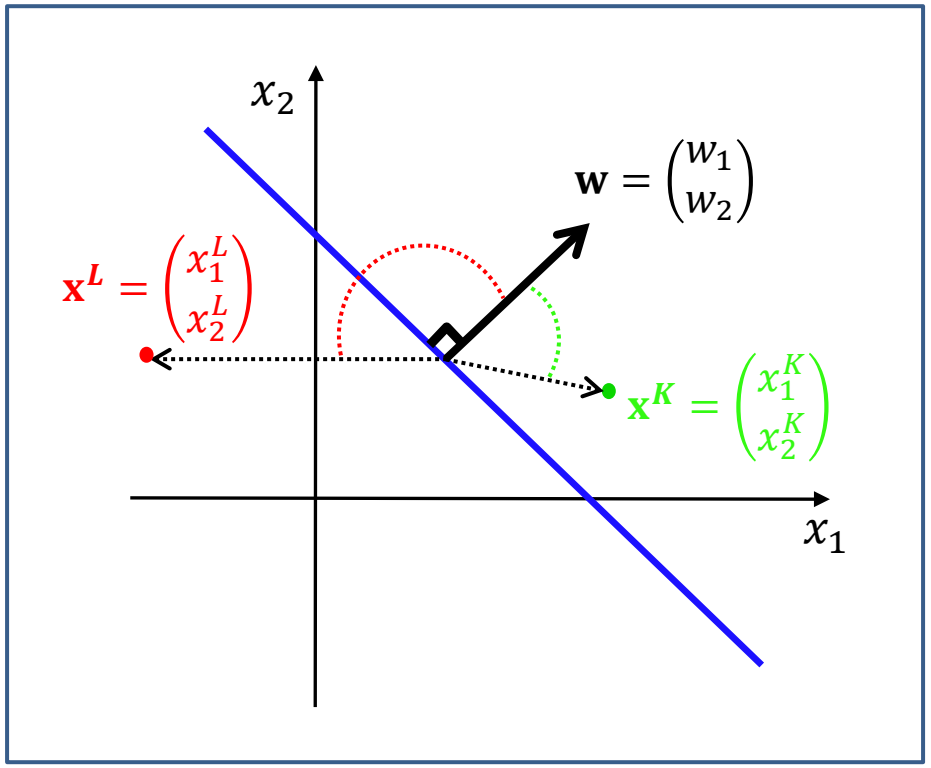
- Each input *x* has an associated **weight *w*** which can be modified
- Inputs *x* corresponds to signals from other neuron axons
  - x_0 - Bias are 'special' inputs, with weight w_0
- Weights ***W*** corresponds to synaptic modulation (i.e. something like strength/amount of neurotransmitters)
- The summation corresponds to 'cell body'
- The activation function corresponds to axon hillock - computes some function *f* of the weighted sum of its inputs
- So, output *y=f(z)*, corresponds to axon signal

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Artificial neuron = a linear classifier



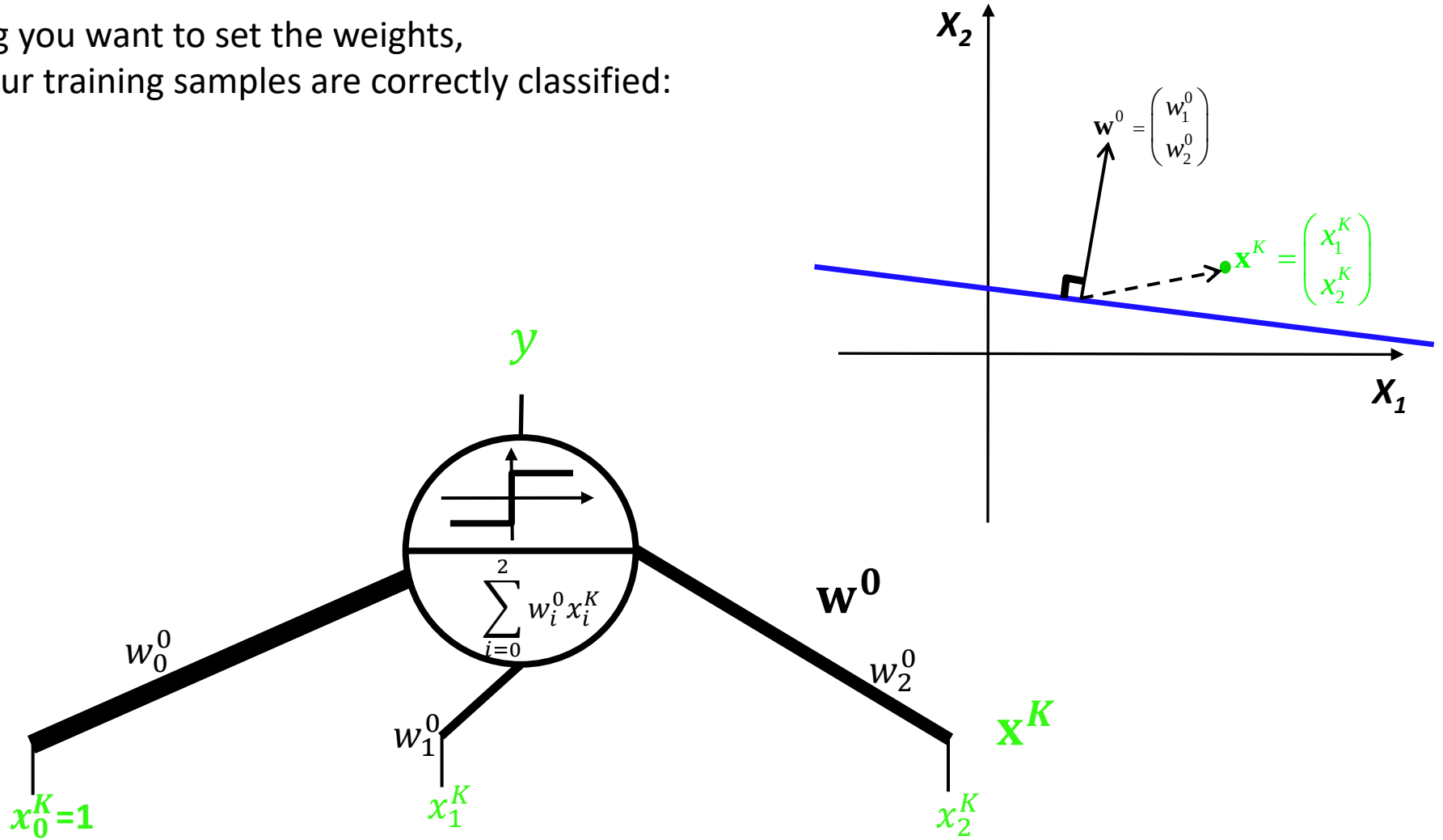


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

One artificial neuron (perceptron)

At training you want to set the weights,
so that your training samples are correctly classified:

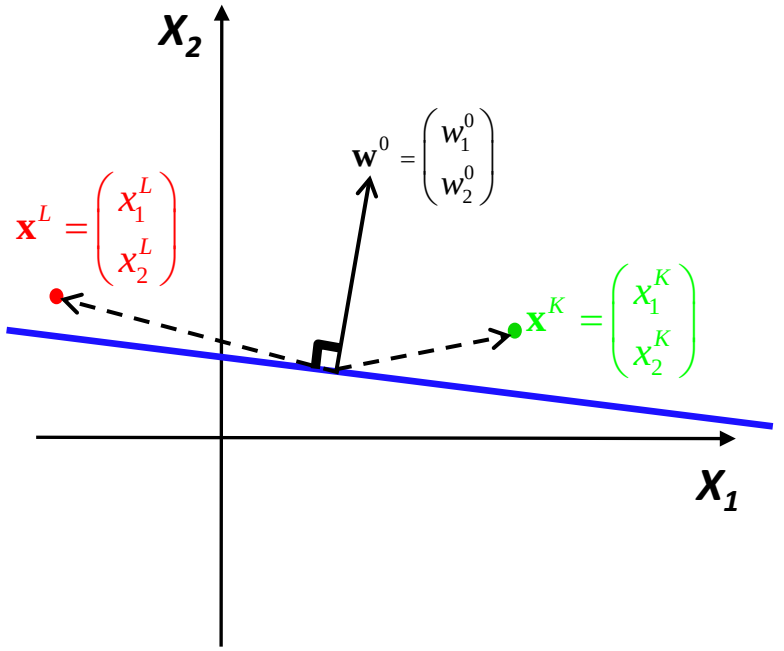


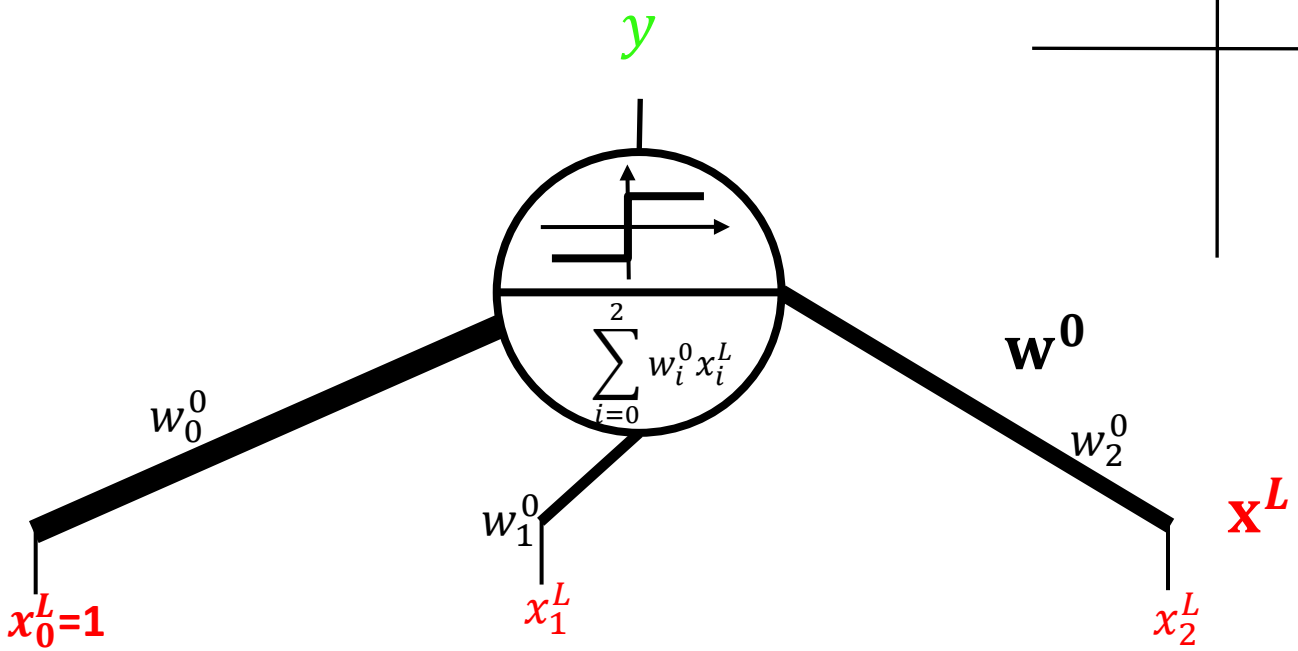
UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

One artificial neuron (perceptron)

At training you want to set the weights,
so that your training samples are correctly classified:



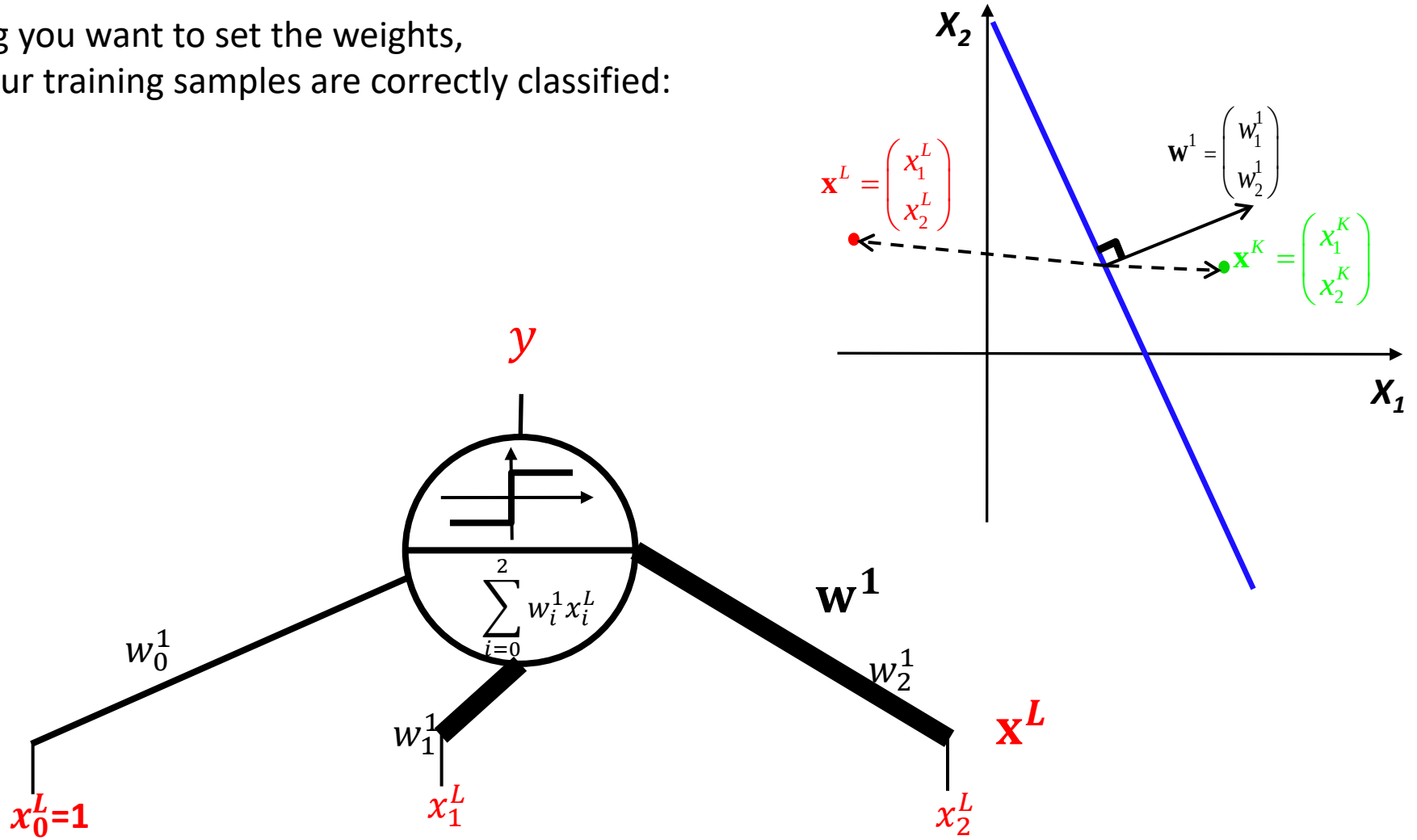


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

One artificial neuron (perceptron)

At training you want to set the weights,
so that your training samples are correctly classified:

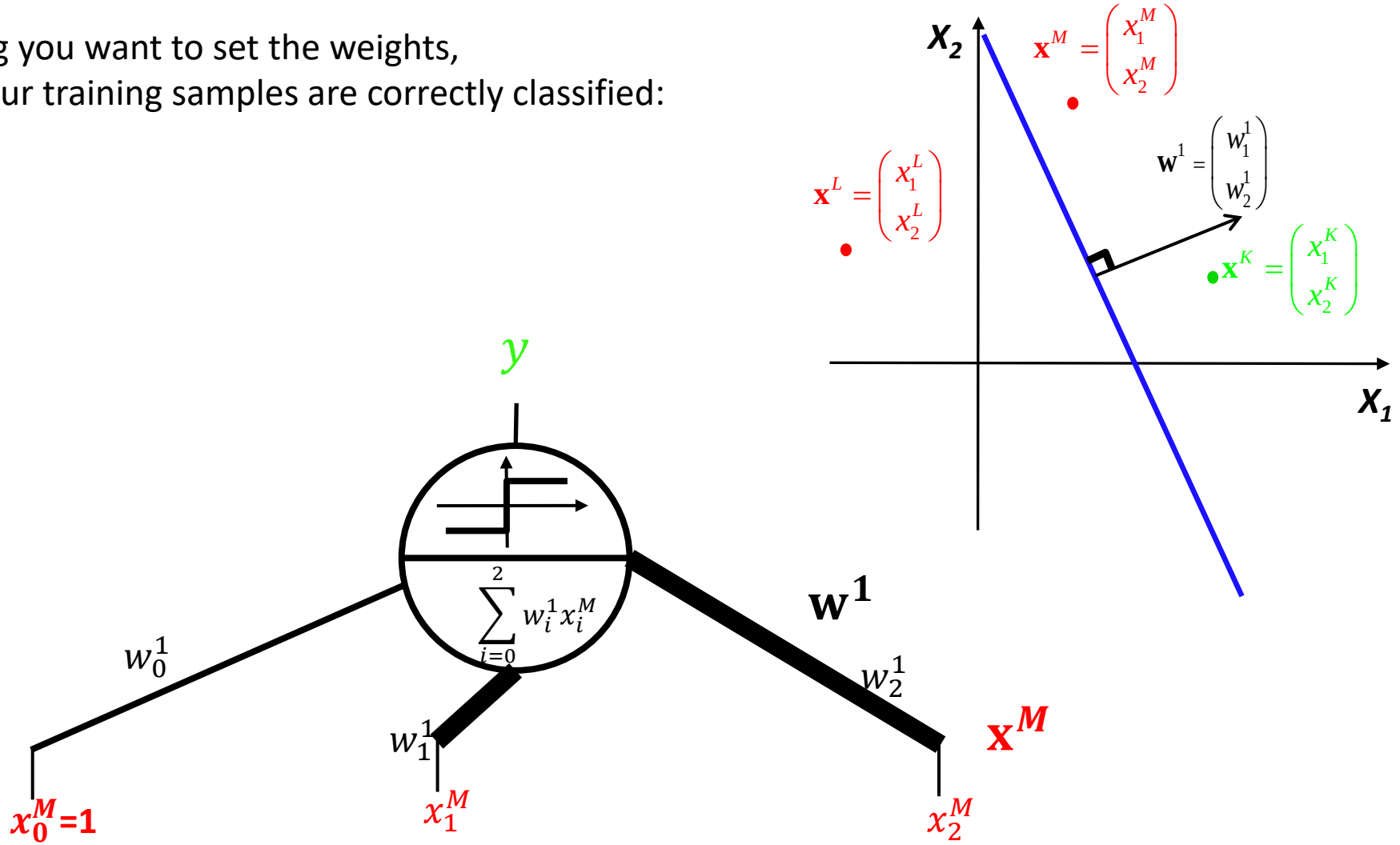


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

One artificial neuron (perceptron)

At training you want to set the weights,
so that your training samples are correctly classified:

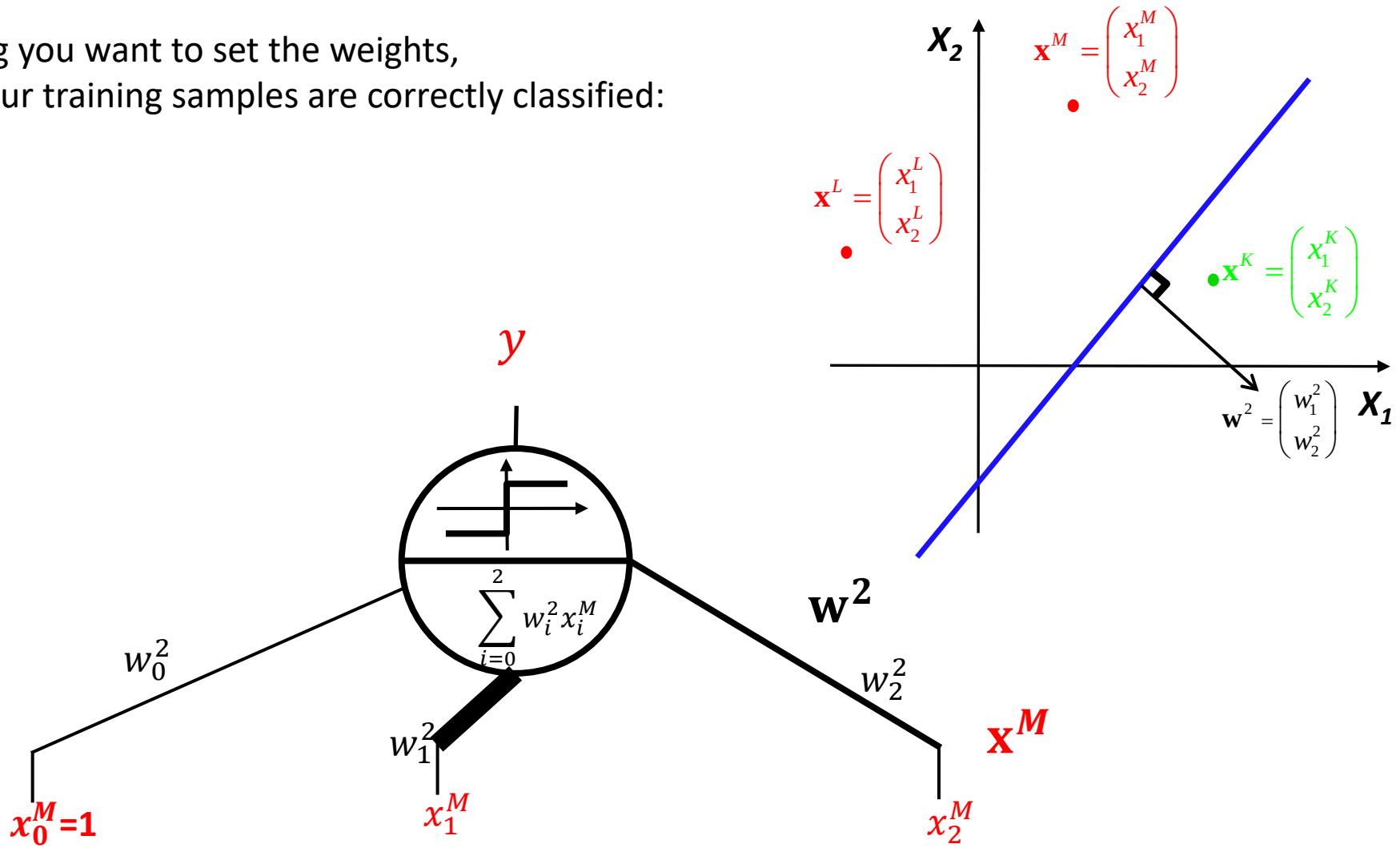


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

One artificial neuron (perceptron)

At training you want to set the weights,
so that your training samples are correctly classified:

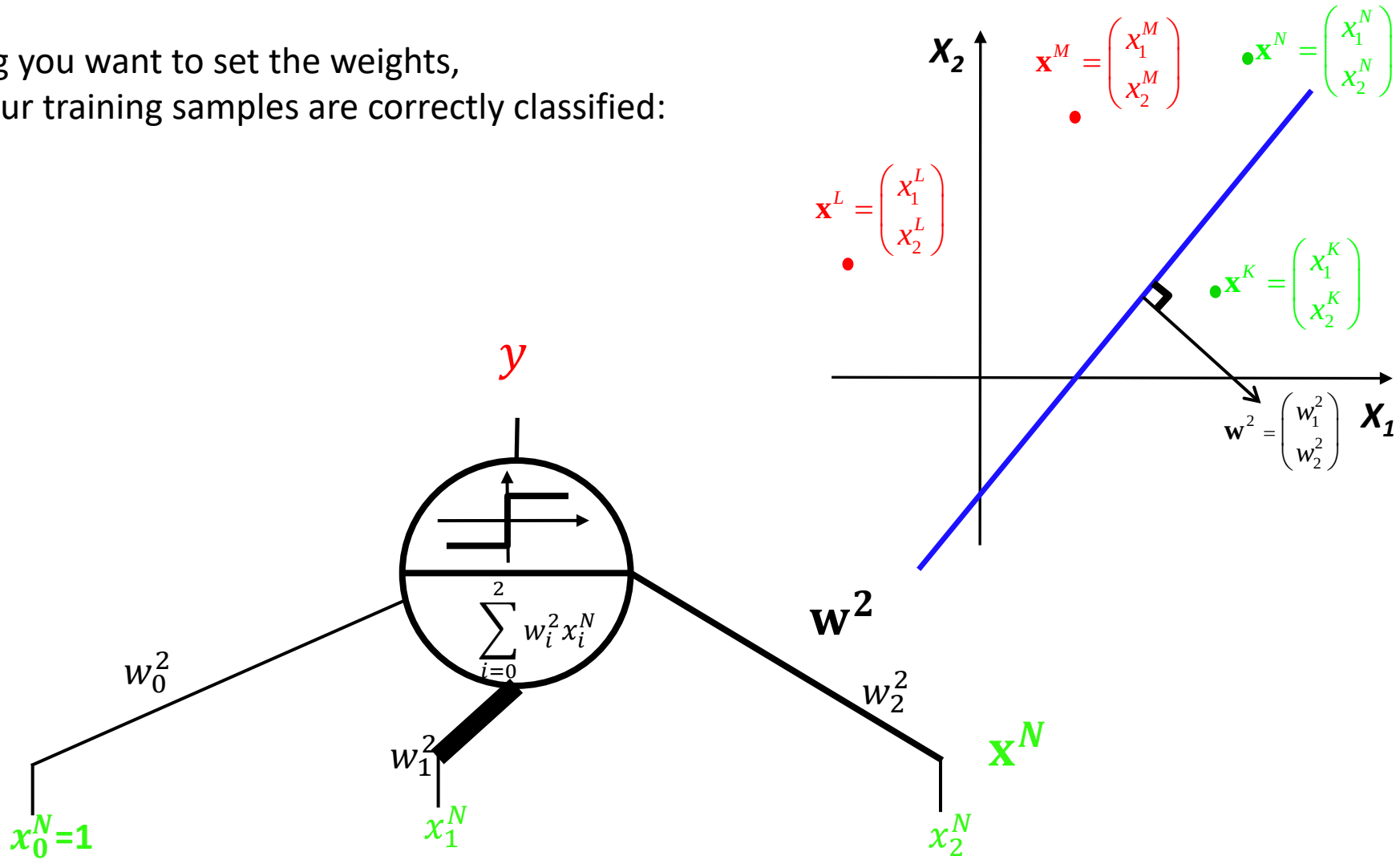


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

One artificial neuron (perceptron)

At training you want to set the weights,
so that your training samples are correctly classified:

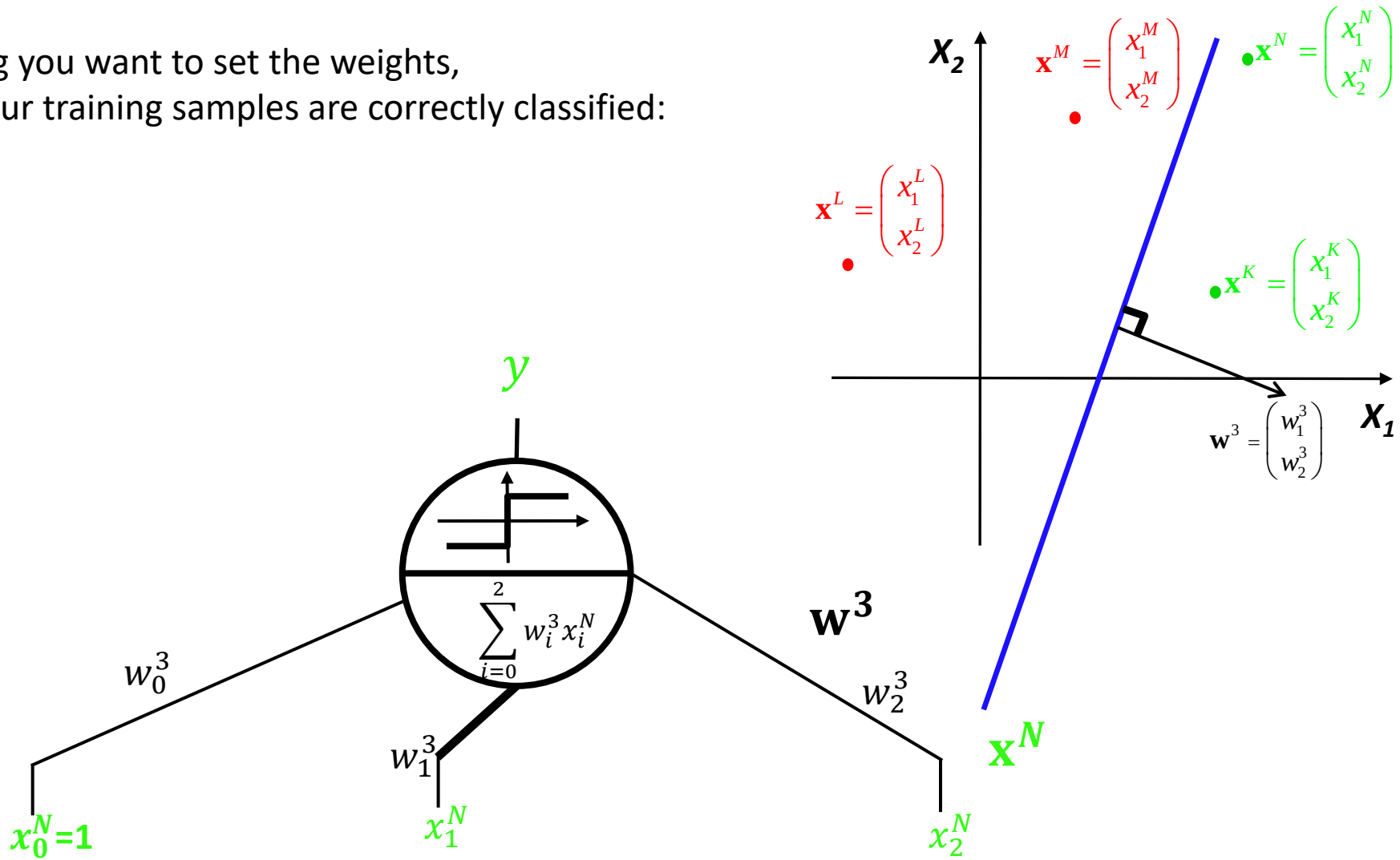


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

One artificial neuron (perceptron)

At training you want to set the weights,
so that your training samples are correctly classified:

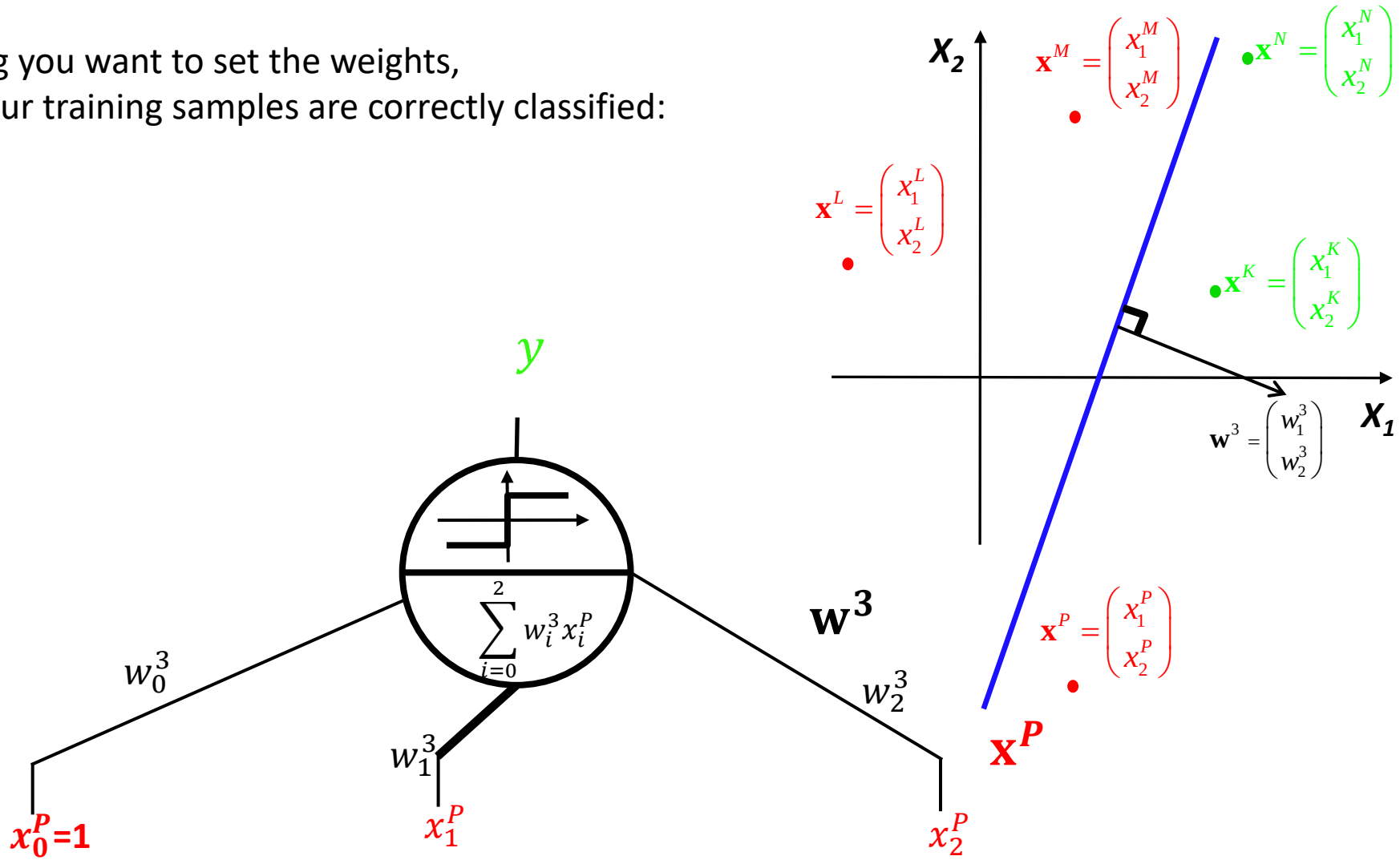


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

One artificial neuron (perceptron)

At training you want to set the weights,
so that your training samples are correctly classified:

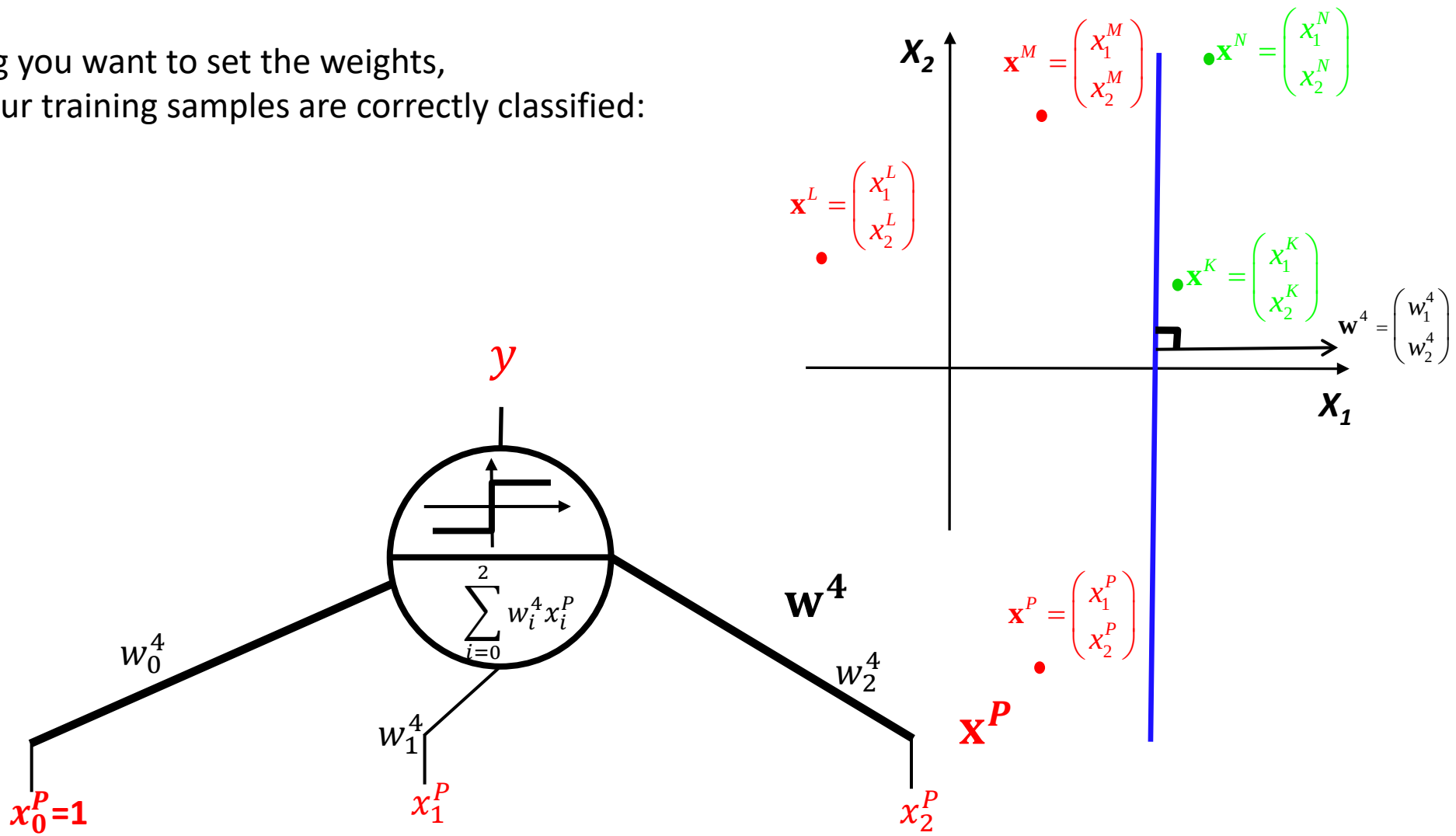


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

One artificial neuron (perceptron)

At training you want to set the weights,
so that your training samples are correctly classified:

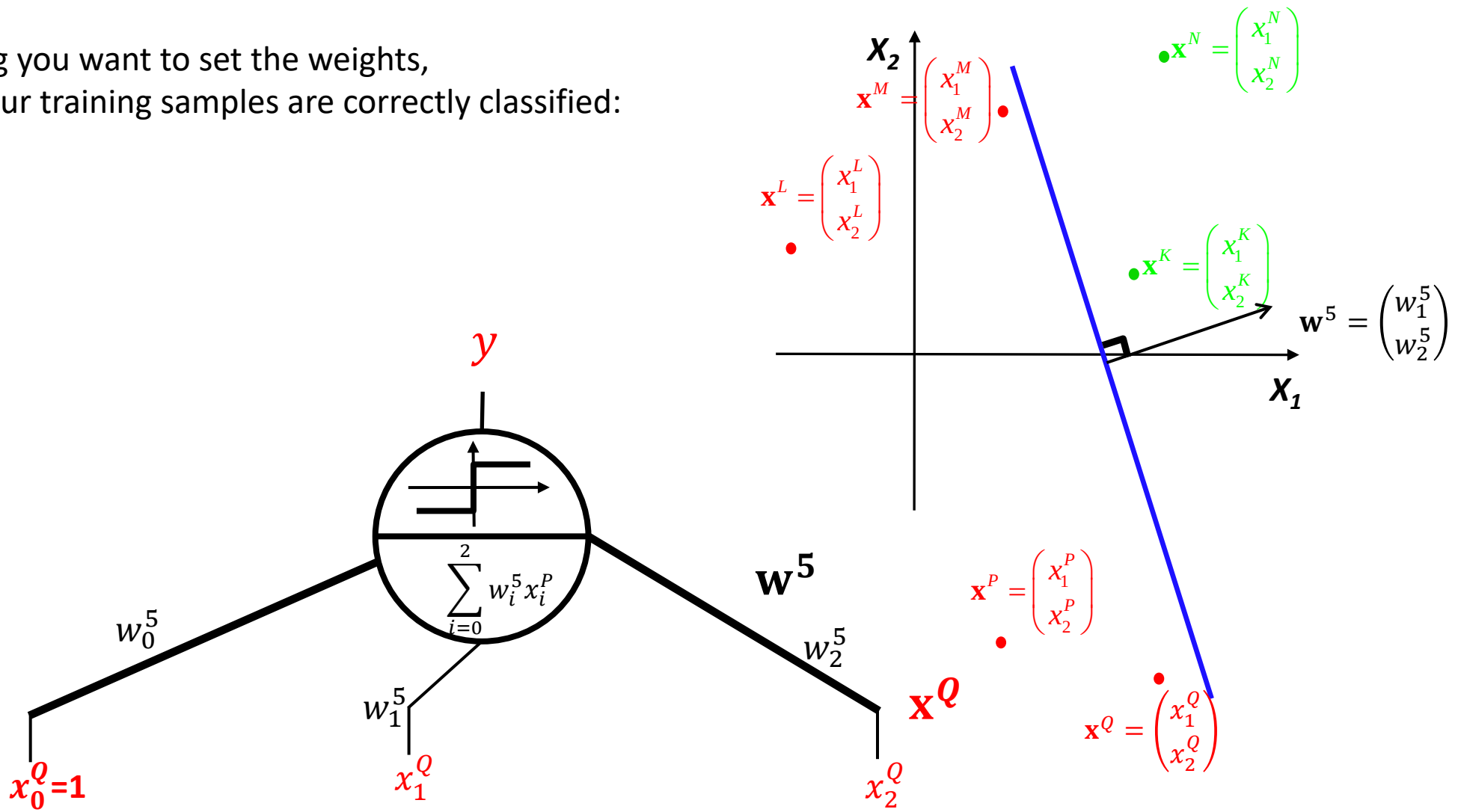


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

One artificial neuron (perceptron)

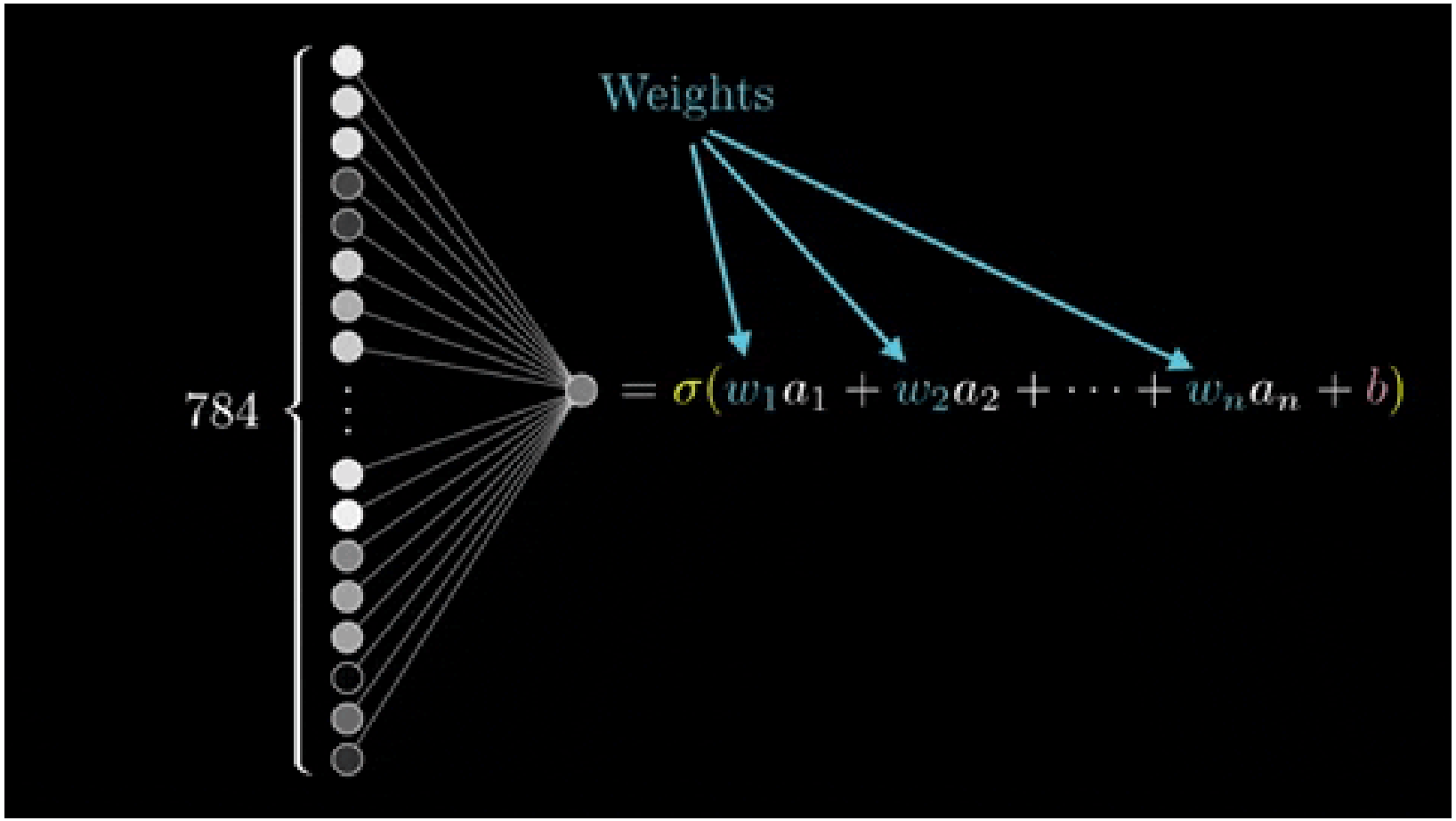
At training you want to set the weights,
so that your training samples are correctly classified:



UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

One neuron = simple linear decision



Source: https://www.3blue1brown.com/neural-networks

UNIVERSITÉ CÔTE D'AZUR

Perceptron: Rosenblatt's Algorithm (1956-1958)

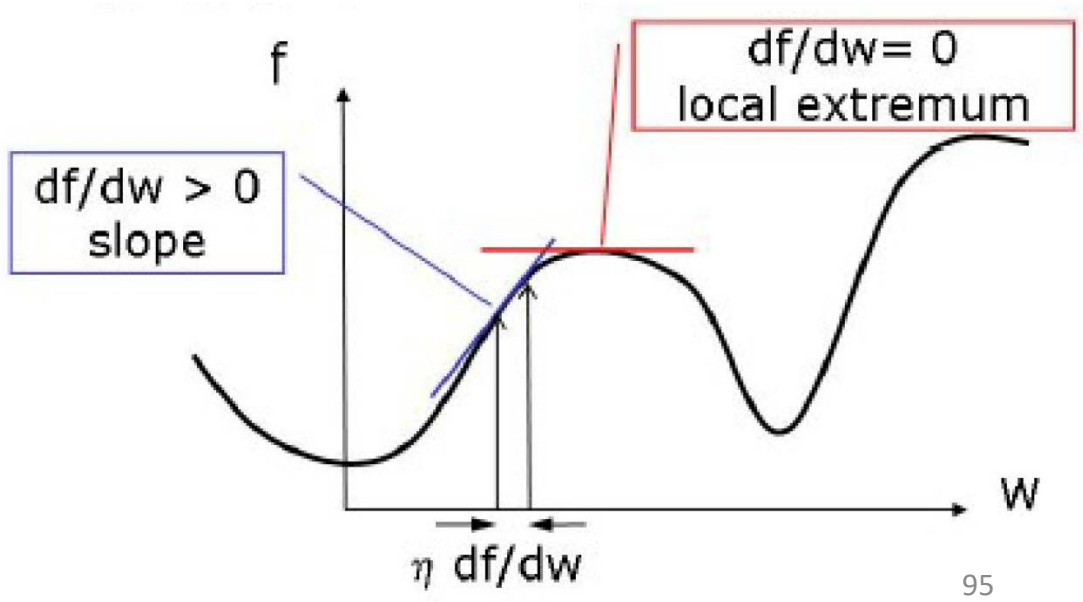
Perceptron Algorithm

- Pick initial weight vector (including w_0), e.g. (0, 0,…,0)
- Repeat until all points are correctly classified
  - *Repeat for each point*
    - Calculate $y^i\mathbf{w}\mathbf{x}^i$ for point i
    - If $y^i\mathbf{w}\mathbf{x}^i > 0$, the point is correctly classified
    - Else change the weights to increase the value of $y^i\mathbf{w}\mathbf{x}^i$; change in weight proportional to $y^i\mathbf{x}^i$

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6):386.

Gradient Ascent

- Why pick $y^i\mathbf{x}^i$ as increment to weights?
- To maximize scalar function of one variable $f(\mathbf{w})$
  - Pick initial $\mathbf{w}$
  - Change $\mathbf{w}$ to $\mathbf{w} + \eta\, df/d\mathbf{w}$ ($\eta > 0$, small)
  - Until f stops changing ($df/d\mathbf{w} \approx 0$)



Gradient Ascent

- To maximize a multivariate function $f(\mathbf{w})$
  - Pick initial $\mathbf{w}$
  - Change $\mathbf{w}$ to $\mathbf{w} + \eta \nabla f_{\mathbf{w}}$ ($\eta > 0$, small)
  - Until f stops changing ($\nabla f_{\mathbf{w}} \approx 0$)
- Find local maximum, unless function is globally convex

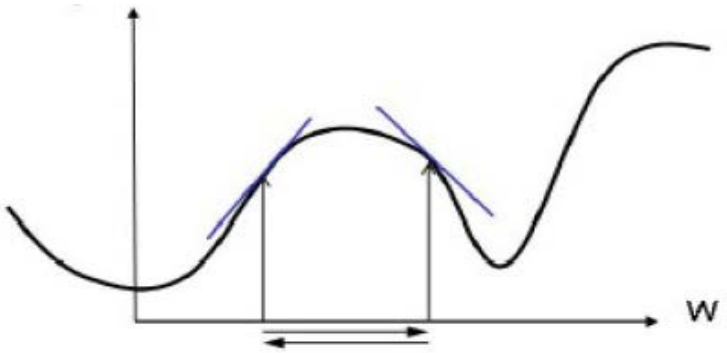
$$\nabla f_{\mathbf{w}} = \left[\frac{\partial f}{\partial \mathbf{w}_1}, \quad \ldots \quad , \frac{\partial f}{\partial \mathbf{w}_n} \right]$$

Gradient Ascent

- To maximize a multivariate function $f(\mathbf{w})$
  - Pick initial $\mathbf{w}$
  - Change $\mathbf{w}$ to $\mathbf{w} + \eta \nabla f_{\mathbf{w}}$ ($\eta > 0$, small)
  - Until f stops changing ($\nabla f_{\mathbf{w}} \approx 0$)

$$\nabla f_{\mathbf{w}} = \left[\frac{\partial f}{\partial \mathbf{w}_1}, \quad \ldots \quad , \frac{\partial f}{\partial \mathbf{w}_n} \right]$$

- Find local maximum, unless function is globally convex
- If f is non-linear, the learning rate η has to be chosen very carefully
  - Too small $\Rightarrow$ slow convergence
  - Too big $\Rightarrow$ osccilations



Gradient Ascent

- Maximize margin of misclassified points

$$f(\mathbf{w}) = \sum_{on\ \mathbf{i}\ misclassified\ points} y^i \mathbf{w}\mathbf{x}^{\mathrm{i}}$$

$$\nabla_{\mathbf{w}} f(\mathbf{w}) = \sum_{on\ \mathbf{i}\ misclassified\ points} y^i \mathbf{x}^{\mathrm{i}}$$

- Off-line training: Compute, at each iteration, the gradient as sum over all training points
- On-line training: Approximate gradient by one of the terms in the sum: $y^i \mathbf{x}^{\mathrm{i}}$ (principle of the *Stochastic Gradient Descent*, SGD)

UNIVERSITÉ CÔTE D'AZUR

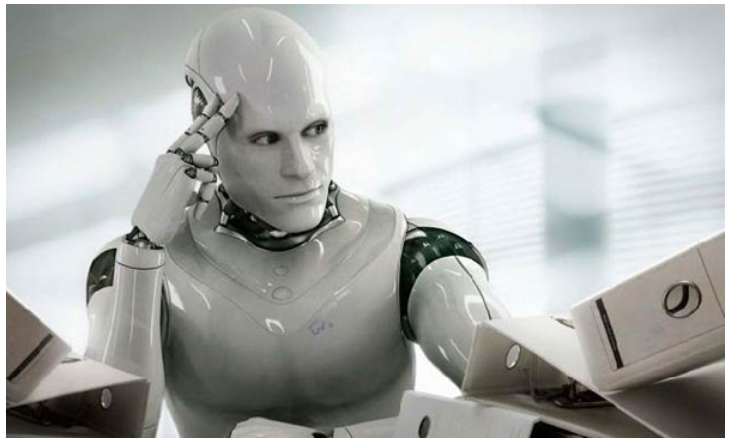
univ-cotedazur.fr

Gradient Descent

This is the “learning” of machines in deep learning ……

➡ Even alpha go using this approach.

What People imagine ……



In reality …..

Perceptron Algorithm

- Each change of **w** decreases the error on a specific point. However, changes for several points are correlated, that is different points could change the weights in opposite directions. Thus, this iterative algorithm requires several loops to converge.

- Guarantee to find a separating hyperplane if one exists – if data is linearly separable.

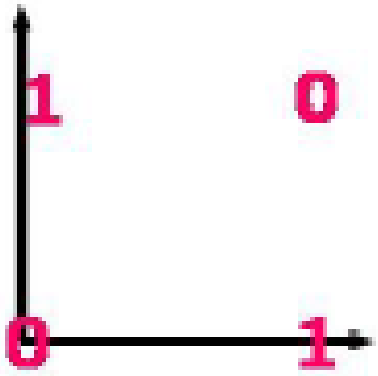
- If data are not linearly separable, then this algorithm loops indefinitely.

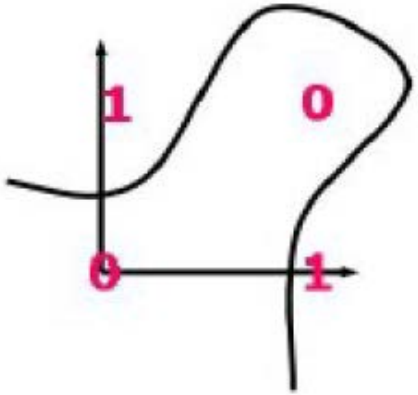
Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6):386.

UNIVERSITÉ CÔTE D'AZUR
univ-cotedazur.fr

Beyond Linear Separability

- Values of the XOR boolean function cannot be separated by a single perceptron unit
  [Minsky and Papert, 1969].





Minsky, M. and Papert, S. (1969). Perceptrons: An Introduction to Computational Geometry. MIT Press.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Overview

- Machine Learning vs Statistics
- Math Basics
- Simple Model
- **From Simple to Complex**

UNIVERSITÉ CÔTE D'AZUR

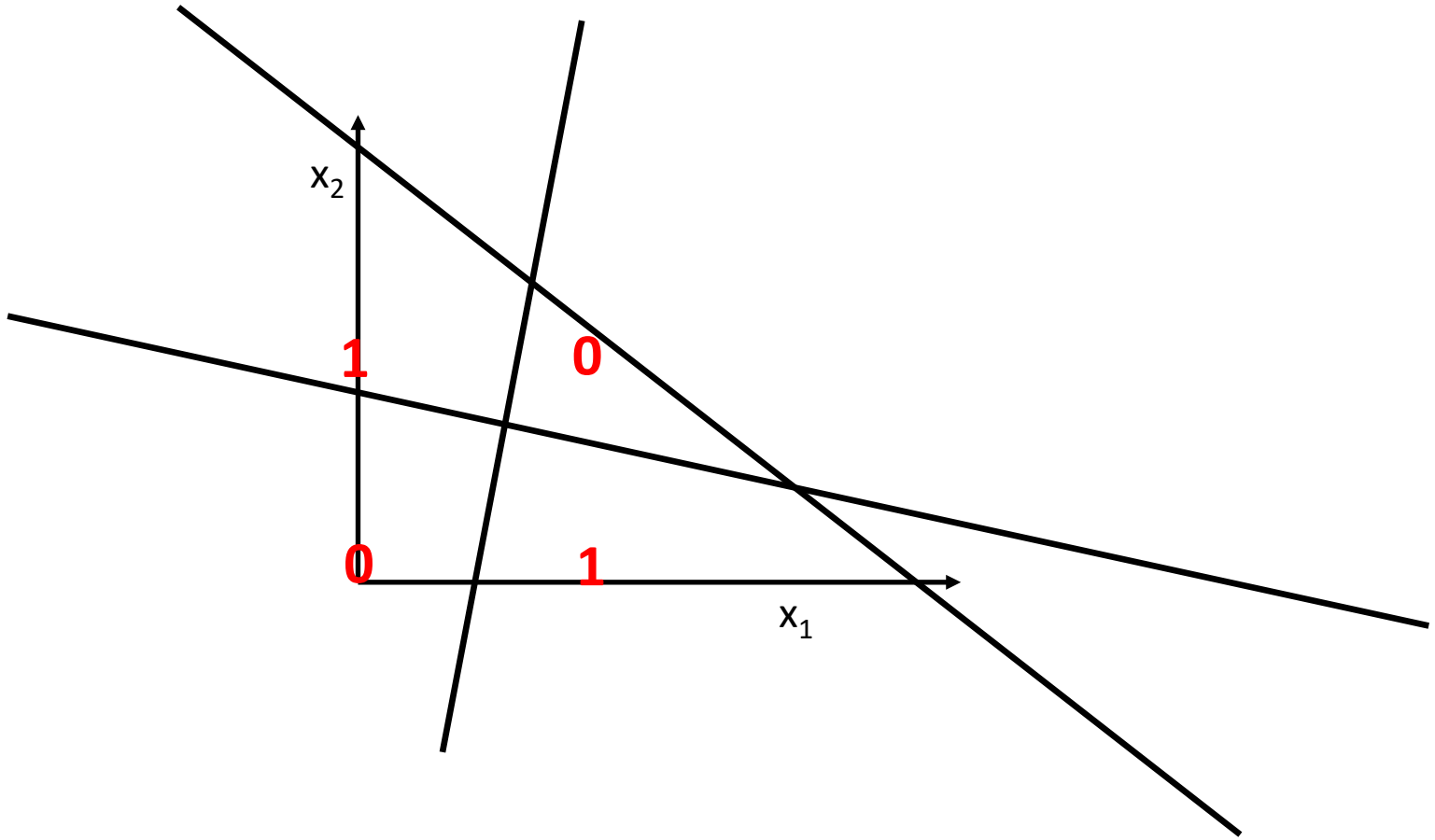
univ-cotedazur.fr

FROM SIMPLE TO COMPLEX

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

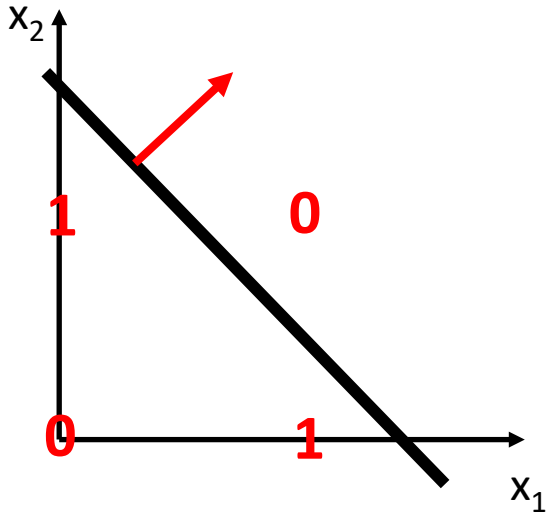
Problem which cannot be solved with a unique straight line

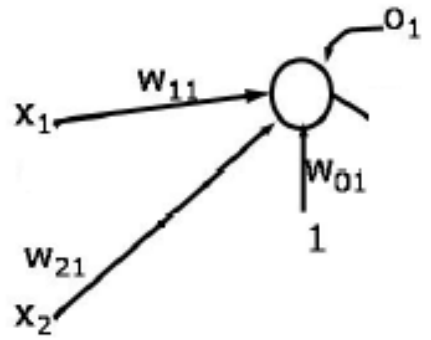


UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

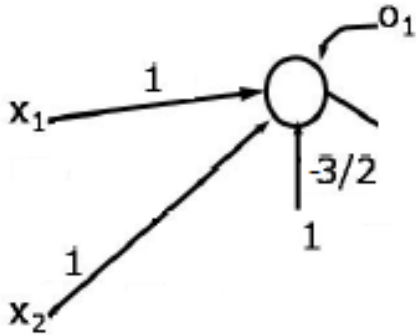
Problem which cannot be solved with a unique straight line





$w_{01} = -3/2 \quad w_{11} = w_{21} = 1$

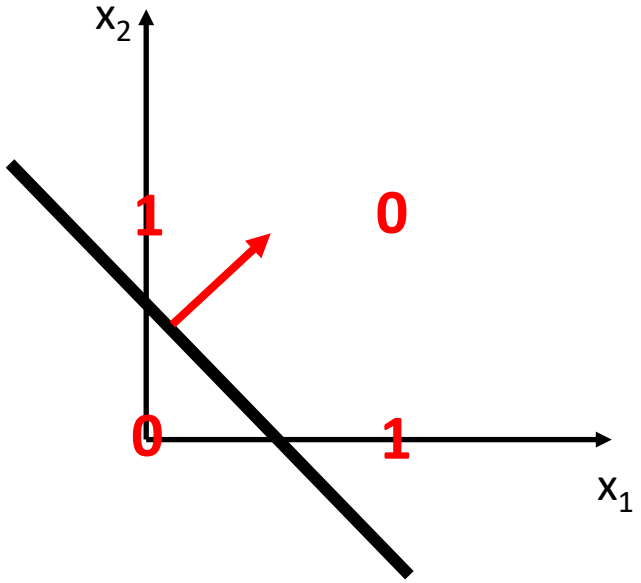
x_1	x_2	o_1
0	0	0
0	1	0
1	0	0
1	1	1



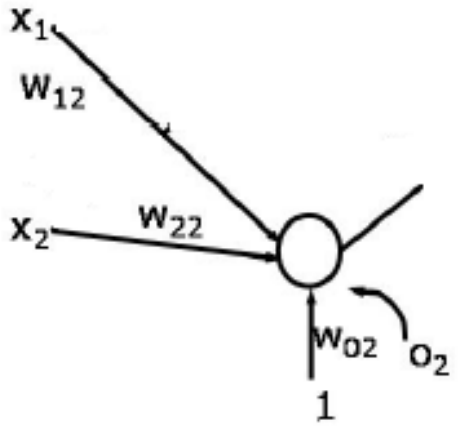
UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

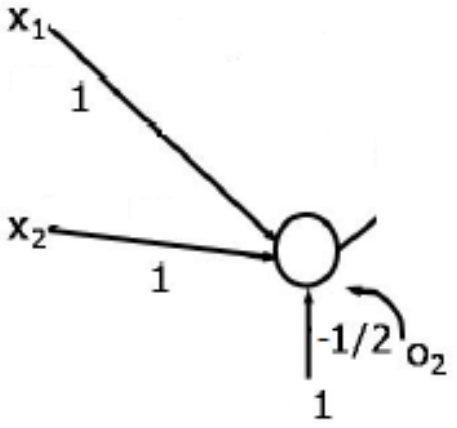
Problem which cannot be solved with a unique straight line



x_1	x_2	o_1	o_2
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1



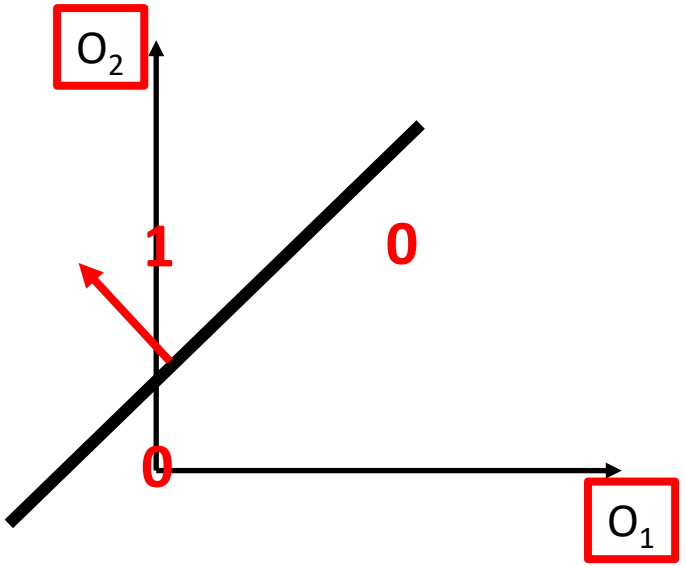
$w_{02} = -1/2 \quad w_{12} = w_{22} = 1$



UNIVERSITÉ CÔTE D'AZUR

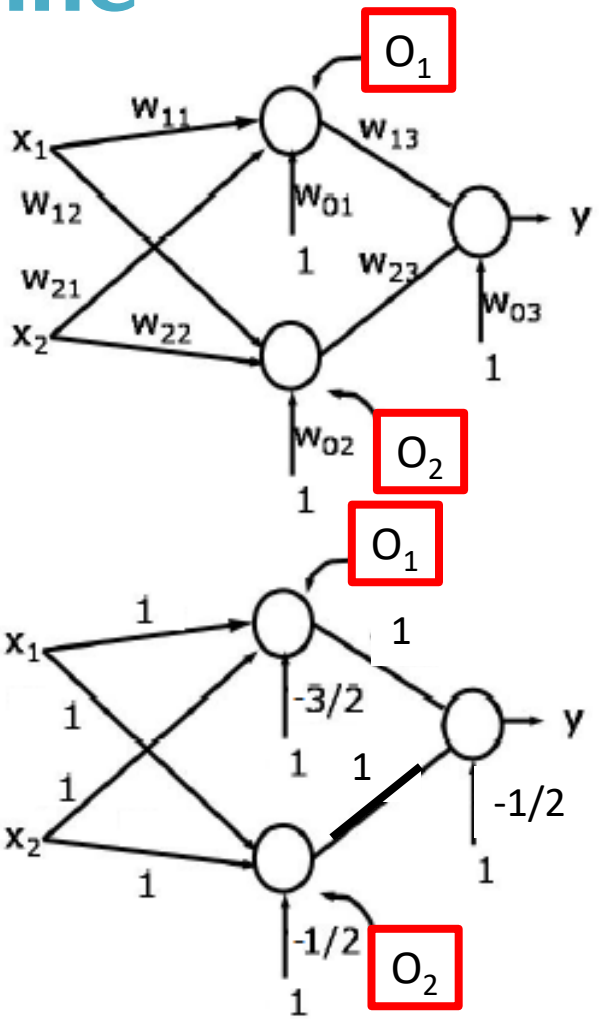
univ-cotedazur.fr

Problem which cannot be solved with a unique straight line



$O_2-O_1-1/2=0$

o_1	o_2	y
0	0	0
0	1	1
0	1	1
1	1	0



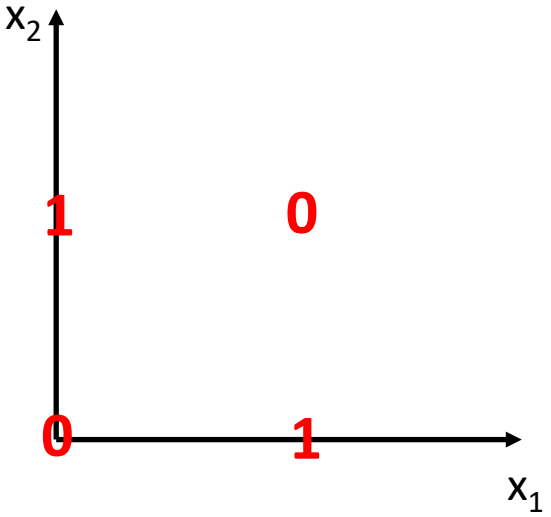
$w_{23}o_2+w_{13}o_1+w_{03}=0$

$w_{03}=-1/2$, $w_{13}=-1$, $w_{23}=1$

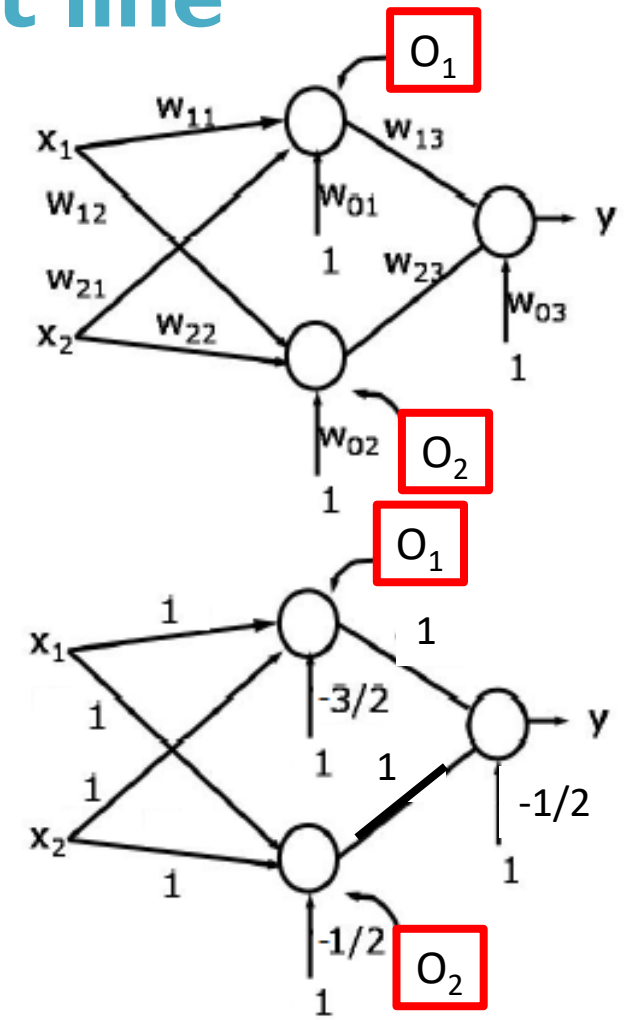
UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Problem which cannot be solved with a unique straight line



x_1	x_2	o_1	o_2	y
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0



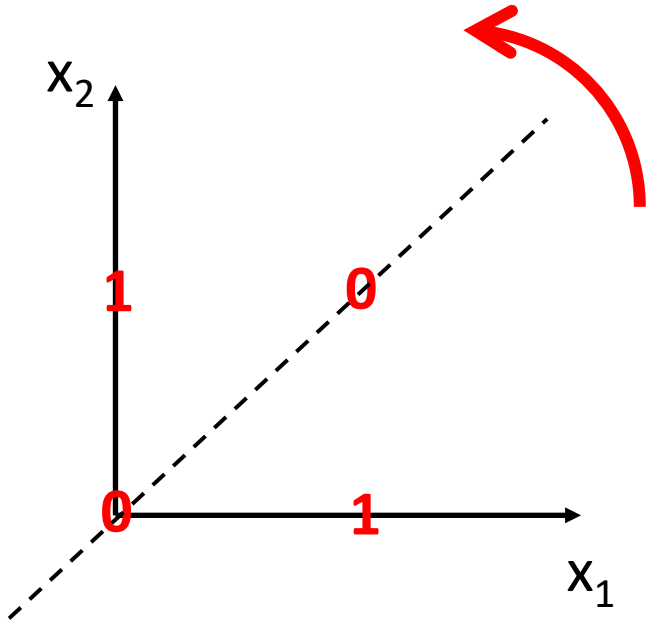
$w_{23}o_2+w_{13}o_1+w_{03}=0$

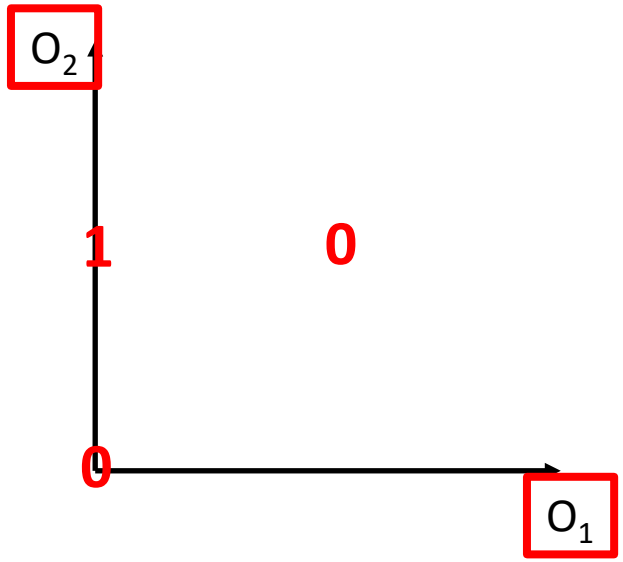
$w_{03}=-1/2$, $w_{13}=-1$, $w_{23}=1$

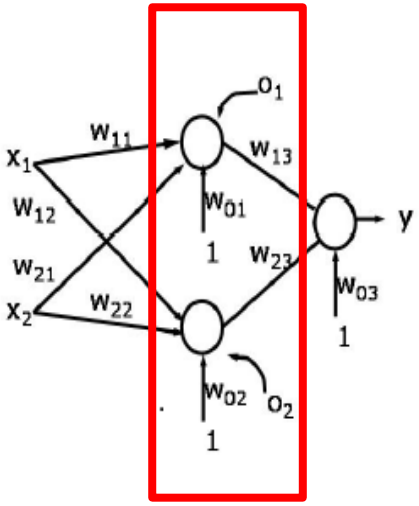
UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Problem which cannot be solved with a unique straight line





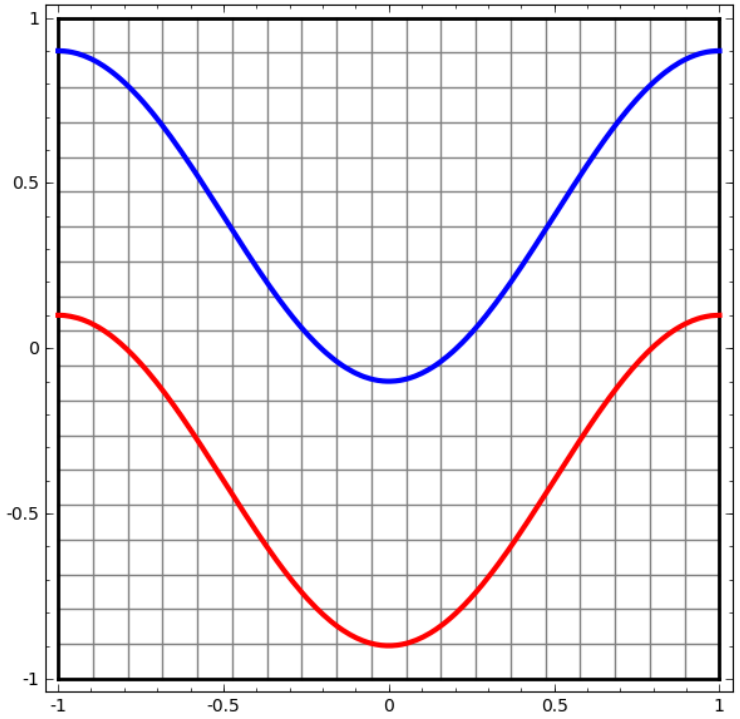


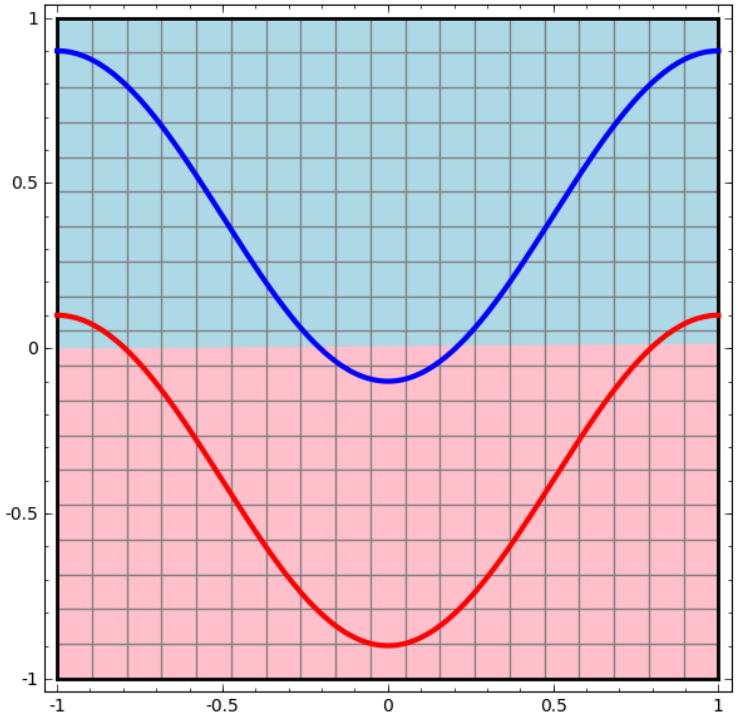
Adding a hidden layer of neurons has fold the input space

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

One perceptron



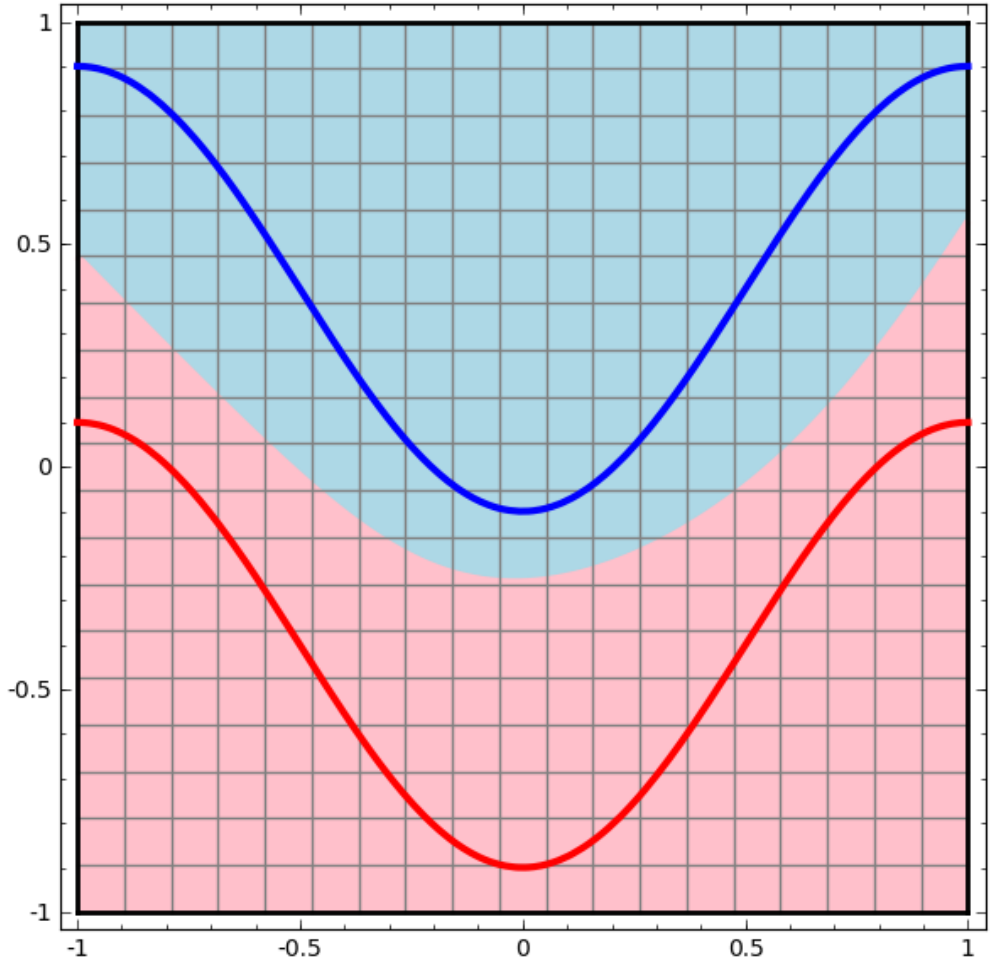


Illustrations from: http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Multi-Layer Perceptron, manifold disentanglement

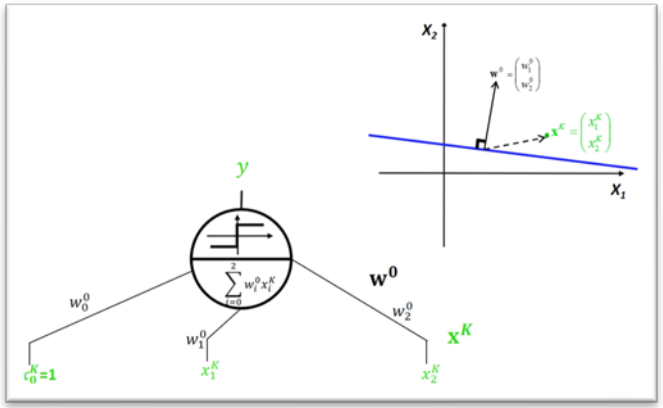


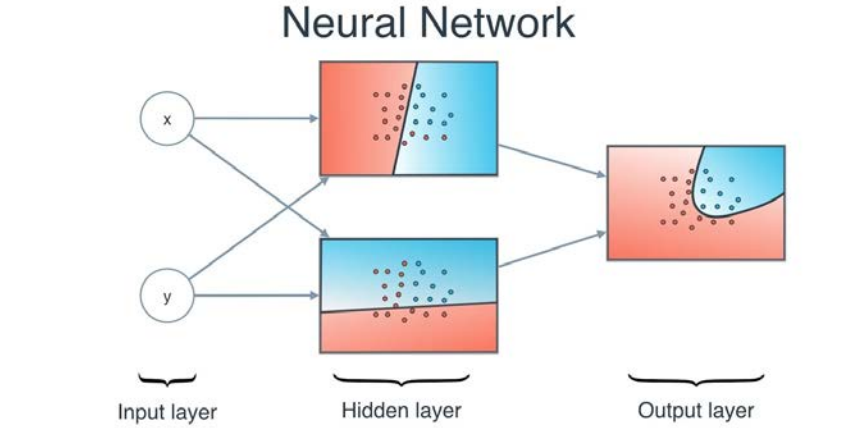
Illustrations from: http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

UNIVERSITÉ CÔTE D'AZUR

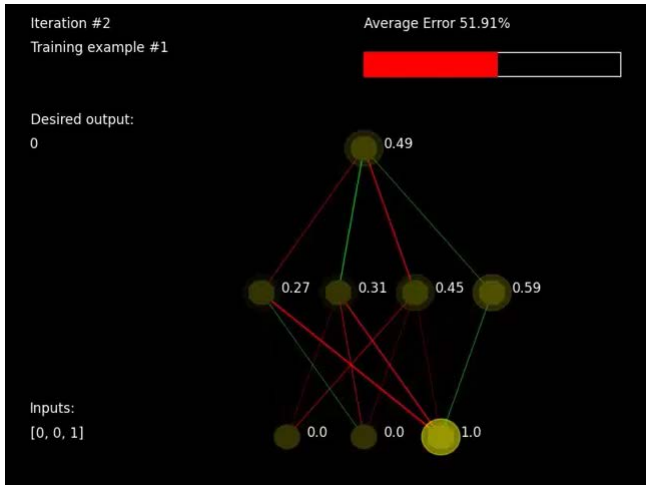
univ-cotedazur.fr

From perceptron to network





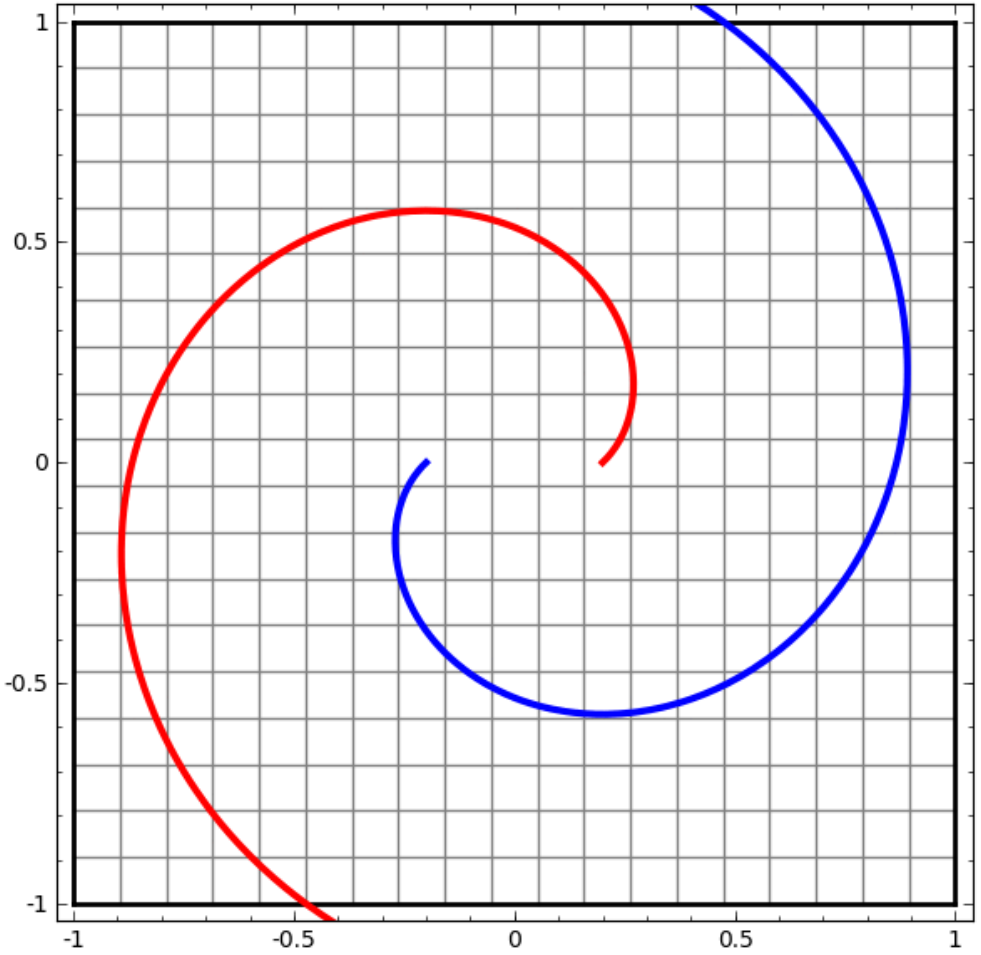
@tachyeonz: A friendly introduction to neural networks and deep learning.



UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Multi-Layer Perceptron, manifold disentanglement

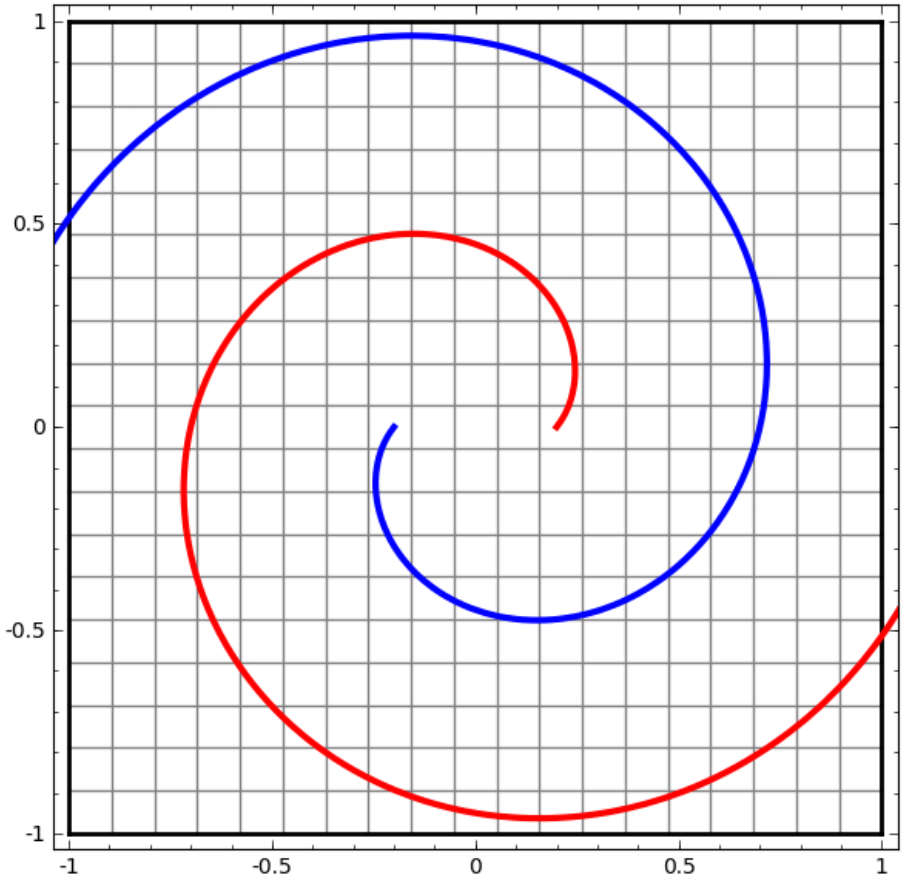


Illustrations from: http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Multi-Layer Perceptron, manifold disentanglement



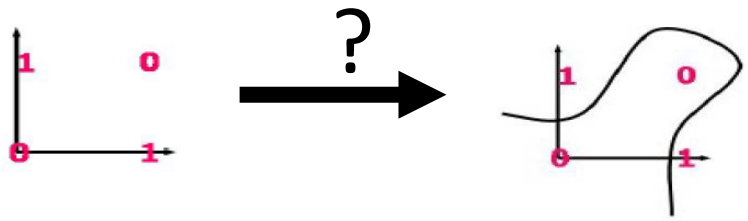
Illustrations from: http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

UNIVERSITÉ CÔTE D'AZUR

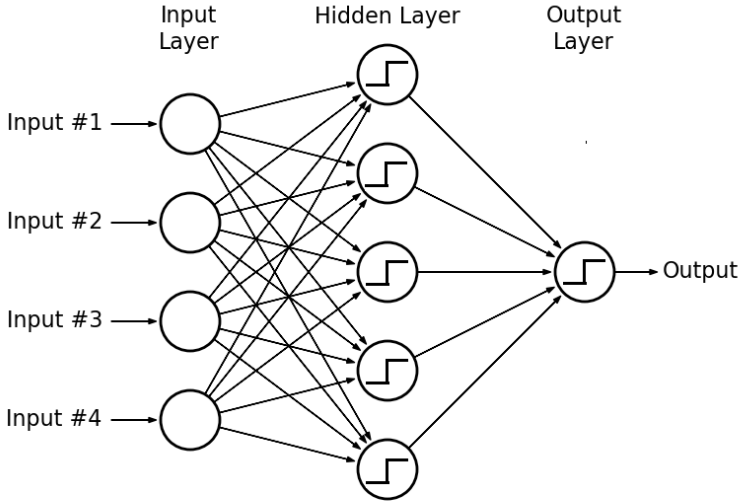
univ-cotedazur.fr

Single Perceptron Unit

- **Perceptron** only learns linear function [Minsky and Papert, 1969]



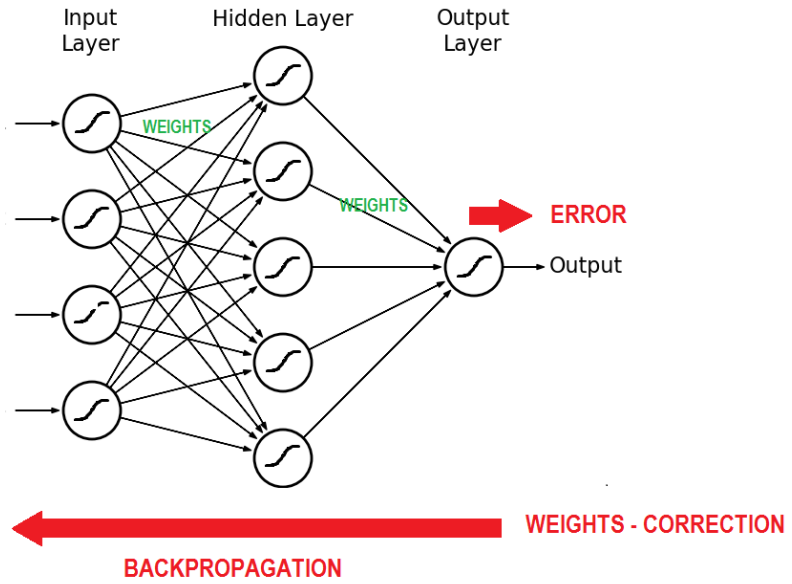
- Non-linear function needs layer(s) of neurons → Neural Network
- Neural Network = input layer + hidden layer(s) + output layer



Marvin Minsky, Seymour Papert, Perceptrons : An Introduction to Computational Geometry, MIT Press, 1969, 268 p.

Multi-Layer Perceptron

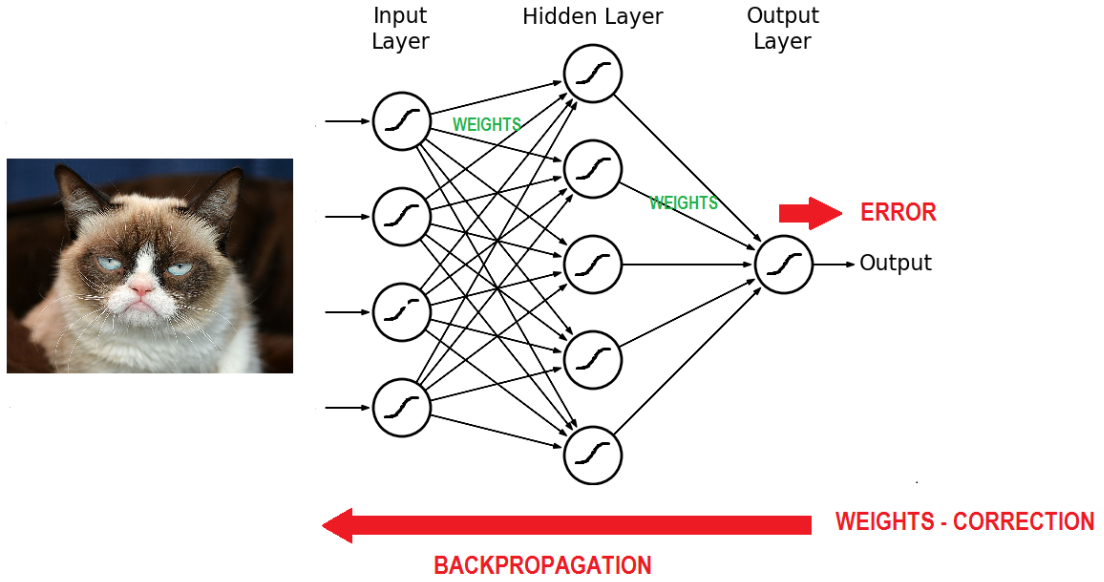
- Training a neural network [Rumelhart et al. / Yann Le Cun et al. 1985]
- **Unknown parameters: weights on the synapses**
- Minimizing a cost function: some metric between the predicted output and the given output



- Step function: non-continuous functions are replaced by a continuous non-linear ones

Multi-Layer Perceptron

- Minimizing a cost function: some metric between the predicted output and the given output



- Equation for a network of 3 neurons (i.e. 3 perceptrons):

$$y = s(w_{13}s(w_{11}x_1 + w_{21}x_2 + w_{01}) + w_{23}s(w_{12}x_1 + w_{22}x_2 + w_{02}) + w_{03})$$

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Multi-Layer Perceptron

Theorem (Cybenko (1989), Hornik & Stinchcombe & White (1989)) A neural network with one single hidden layer is **a universal approximator**: it can represent any continuous function on compact subsets of $\mathbf{R}^n$

- 2 layers is enough ... theoretically:

  "…networks with one internal layer and an arbitrary continuous sigmoidal function can approximate continuous functions with arbitrary precision providing that no constraints are placed on the number of nodes or the size of the weights"

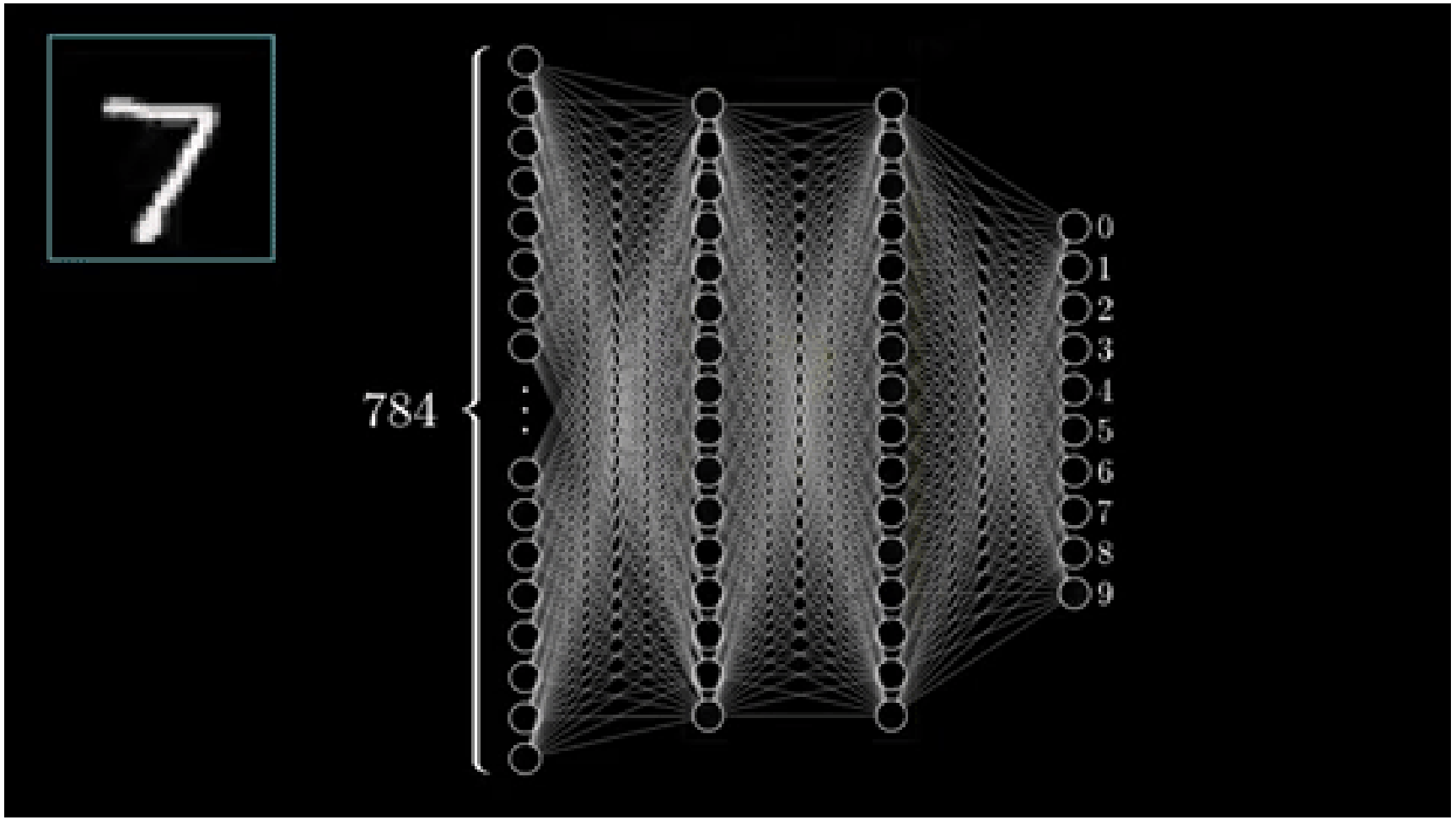
- **But *no efficient learning rule* is known and the size of the hidden layer is *exponential* with the complexity of the problem which is unkown beforehand.**

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems, 2(4), 303-314.
Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural networks, 2(5), 359-366.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Solution: Neural Network

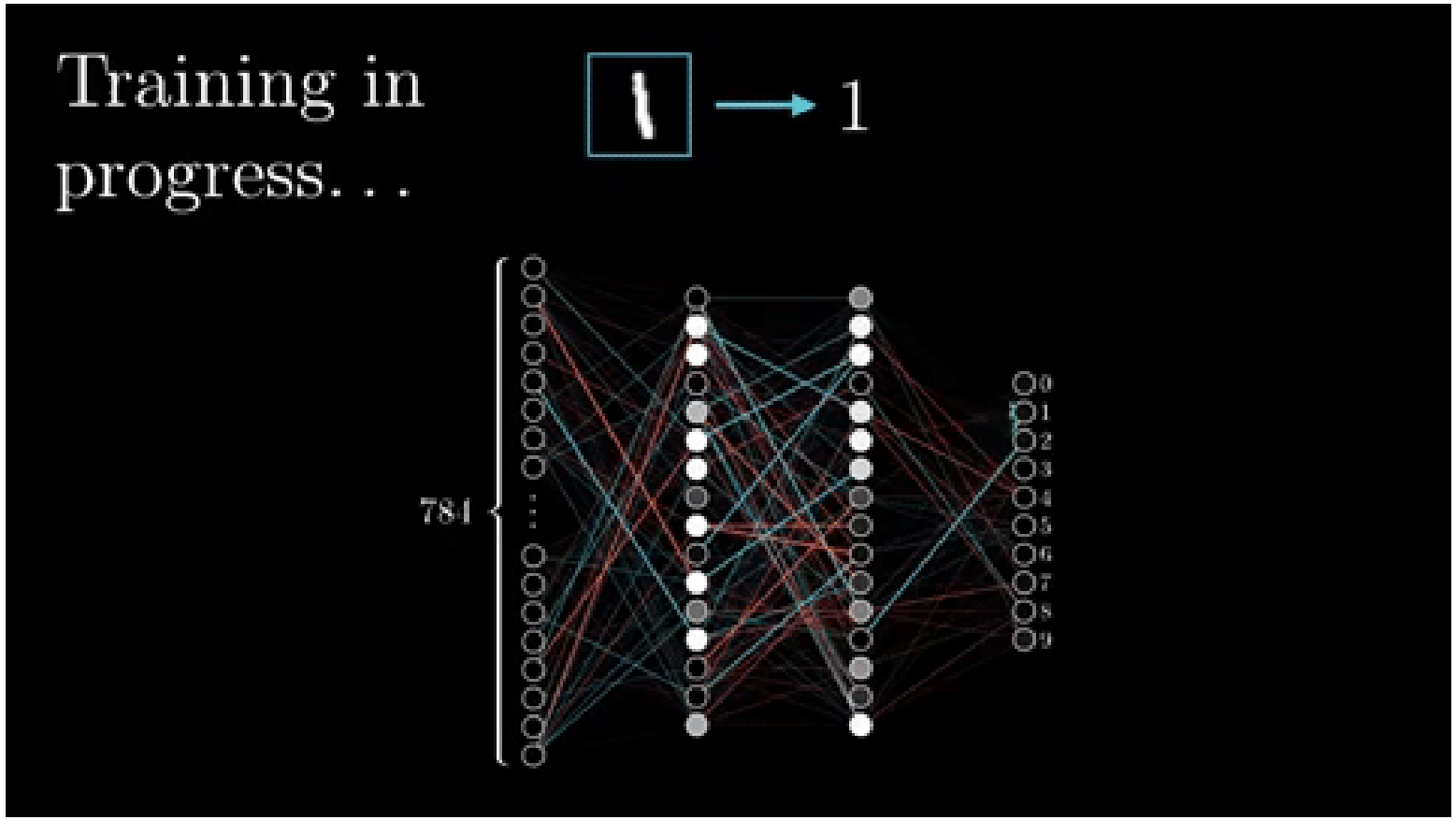


Source: https://www.3blue1brown.com/neural-networks

UNIVERSITÉ CÔTE D'AZUR

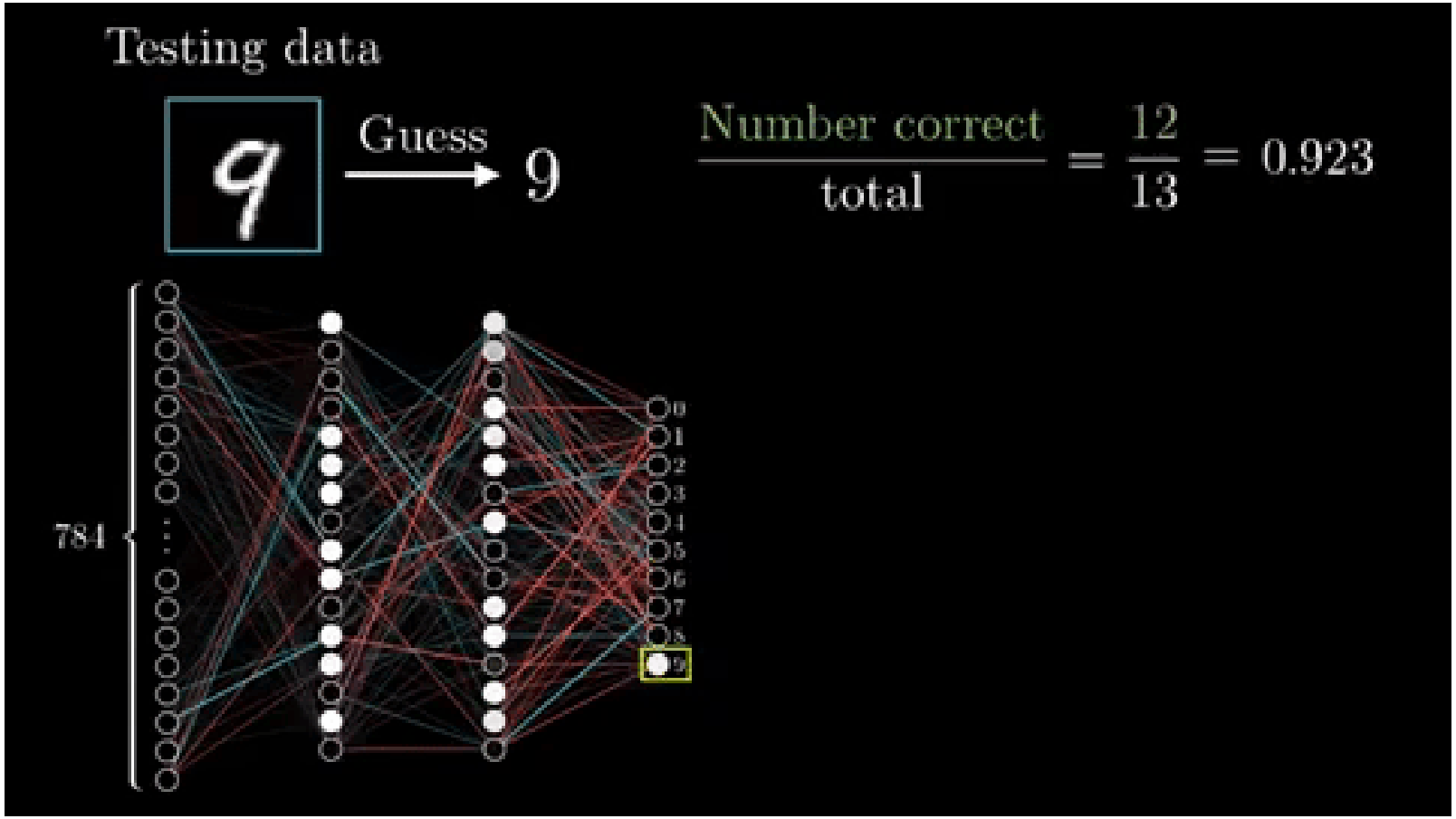
univ-cotedazur.fr

Training



Source: https://www.3blue1brown.com/neural-networks

Inference/Test



Source: https://www.3blue1brown.com/neural-networks

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

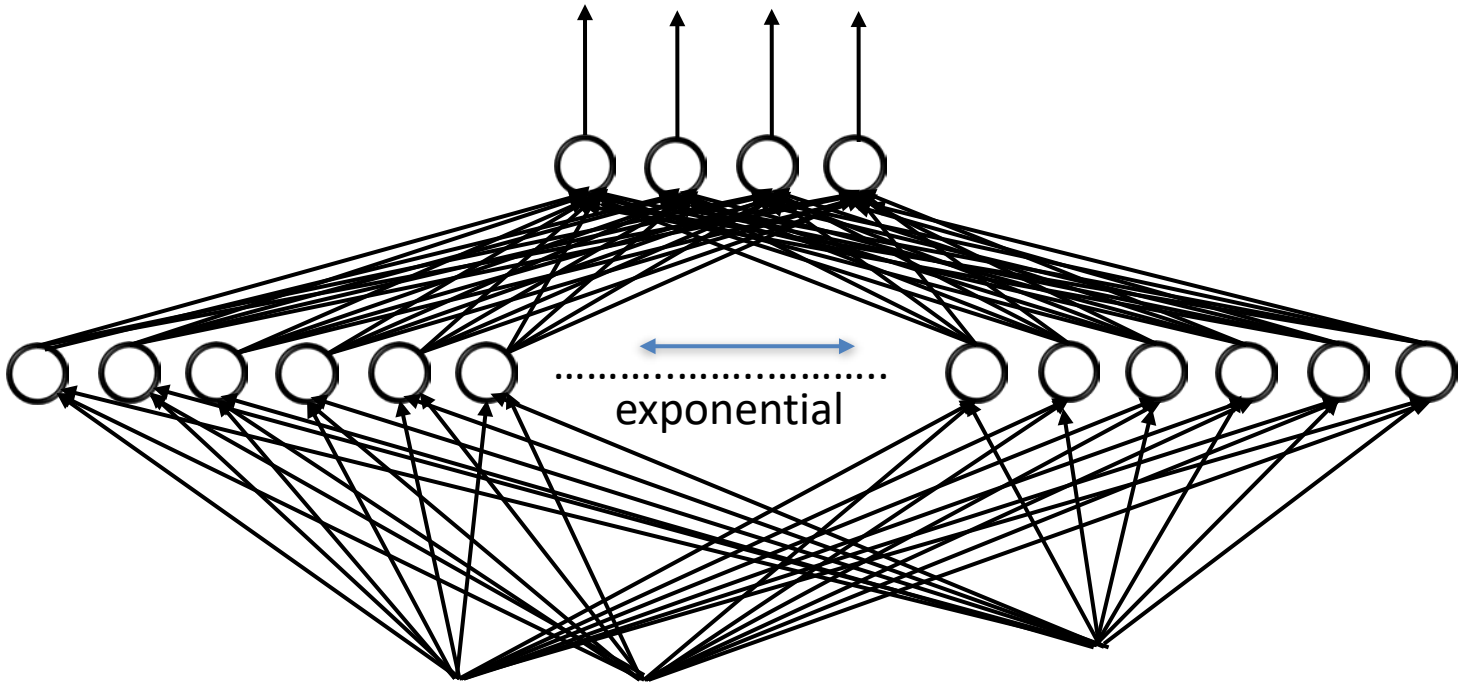
DEEP LEARNING PRINCIPLES

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Deep representation origins

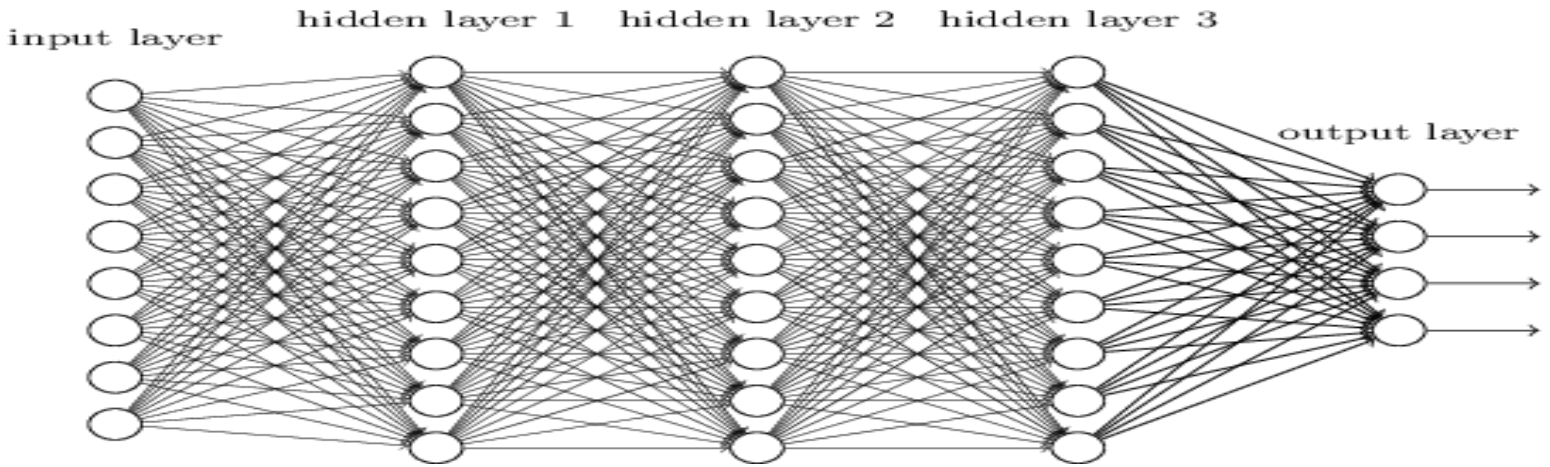
- **Theorems** (Cybenko (1989), Hornik & Stinchcombe & White (1989), Allan Pinkus (1999))
  A neural network with one single hidden layer is a "universal approximator", it can represent any continuous function on compact subsets of $\boldsymbol{R}^n \Rightarrow$ *1 hidden layer is enough…**but** the hidden layer size may be exponential for error* ε *(or even infinite for error 0), and there is no efficient learning rule known.*



Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems, 2(4), 303-314.
Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural networks, 2(5), 359-366.
Pinkus, Allan (1999). "Approximation theory of the MLP model in neural networks". Acta Numerica. 8: 143–195.

Deep representation origins

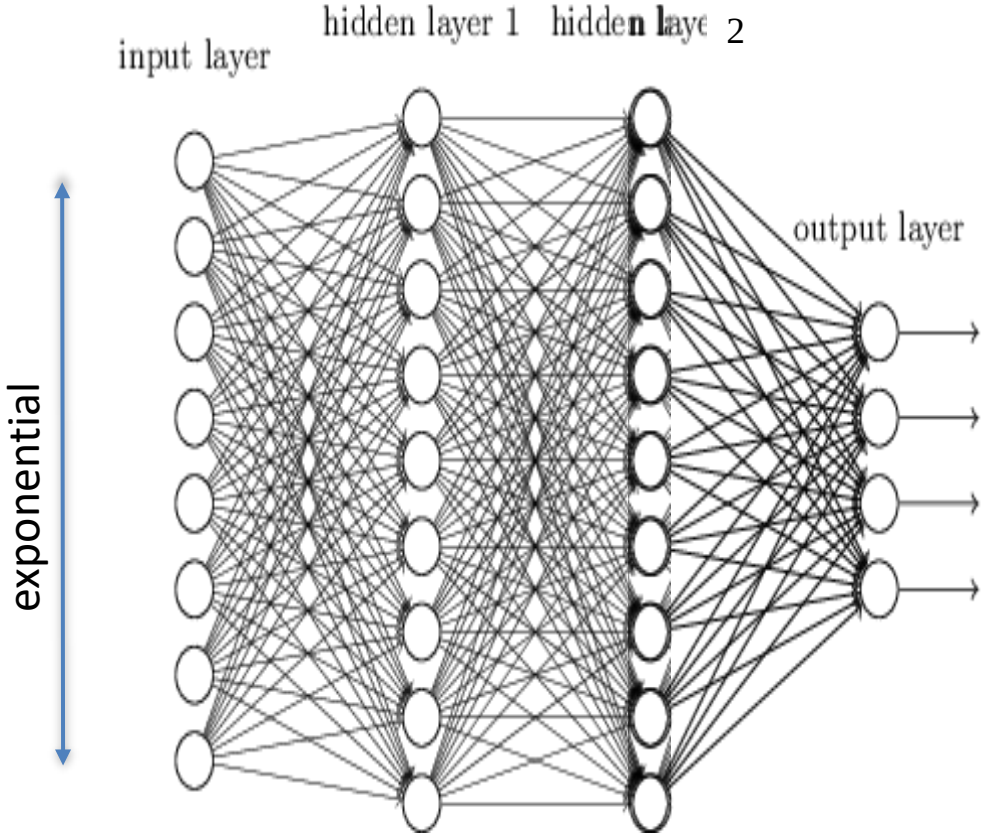
- **Theorem Hastad** (1986), **Bengio** et al. (2007) Functions representable compactly with *k* layers may require exponentially size with *k-1* layers



Johan T. Håstad. Computational Limitations for Small Depth Circuits. MIT Press, Cambridge, MA, 1987.
Bengio, Y., & LeCun, Y. (2007). Scaling learning algorithms towards AI. Large-scale kernel machines, 34(5), 1-41.

Deep representation origins

- **Theorem Hastad** (1986), **Bengio** et al. (2007) Functions representable compactly with *k* layers may require exponentially size with *k-1* layers



UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Deep Neural Network Approximation Theory (2021)

"…deep networks provide exponential approximation accuracy—i.e., the approximation error decays exponentially in the number of nonzero weights in the network—of the multiplication operation, polynomials, sinusoidal functions, and certain smooth functions.

Moreover, this holds true even for one-dimensional oscillatory textures and the Weierstrass function—a fractal function, neither of which has previously known methods achieving exponential approximation accuracy.

We also show that in the approximation of sufficiently smooth functions finite-width deep networks require strictly smaller connectivity than finite-depth wide networks."

Elbrächter, D., Perekrestenko, D., Grohs, P., & Bölcskei, H. (2021). Deep neural network approximation theory. IEEE Transactions on Information Theory, 67(5), 2581-2623.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

The Blessing of dimensionality: Thomas Cover's Theorem (1965)

Cover's theorem states: A complex pattern-classification problem cast in a high-dimensional space nonlinearly is more likely to be linearly separable than in a low-dimensional space
(repeated sequence of Bernoulli trials).

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

The Blessing of dimensionality: Thomas Cover's Theorem (1965)

Cover's theorem states: A complex pattern-classification problem cast in a high-dimensional space nonlinearly is more likely to be linearly separable than in a low-dimensional space.
(repeated sequence of Bernoulli trials)

The number of groupings that can be formed by (l-1)-dimensional hyperplanes to separate N points in two classes is:

$$O(N,l) = 2\sum_{i=0}^{l} \frac{(N-1)!}{(N-1-i)!i!}$$

Notice: The total number of possible groupings is 2^N

UNIVERSITÉ CÔTE D'AZUR

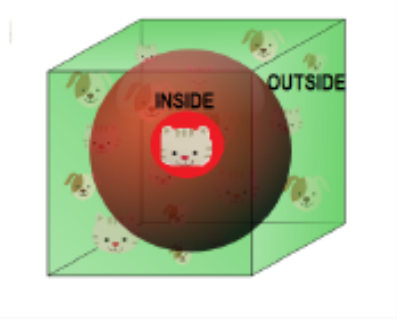
univ-cotedazur.fr

The curse of dimensionality [Bellman, 1956]

- Euclidian distance is not relevant in high dimension: d $\geq$10

1. look at the examples at distance at most r
2. the hypersphere volume is too small: practically empty of examples

$$\frac{volume\ of\ the\ sphere\ of\ radial\ r}{hypersphere\ of\ 2r\ width} \rightarrow_{d\rightarrow\infty} 0$$



3. need a number of examples exponential in d

Remark

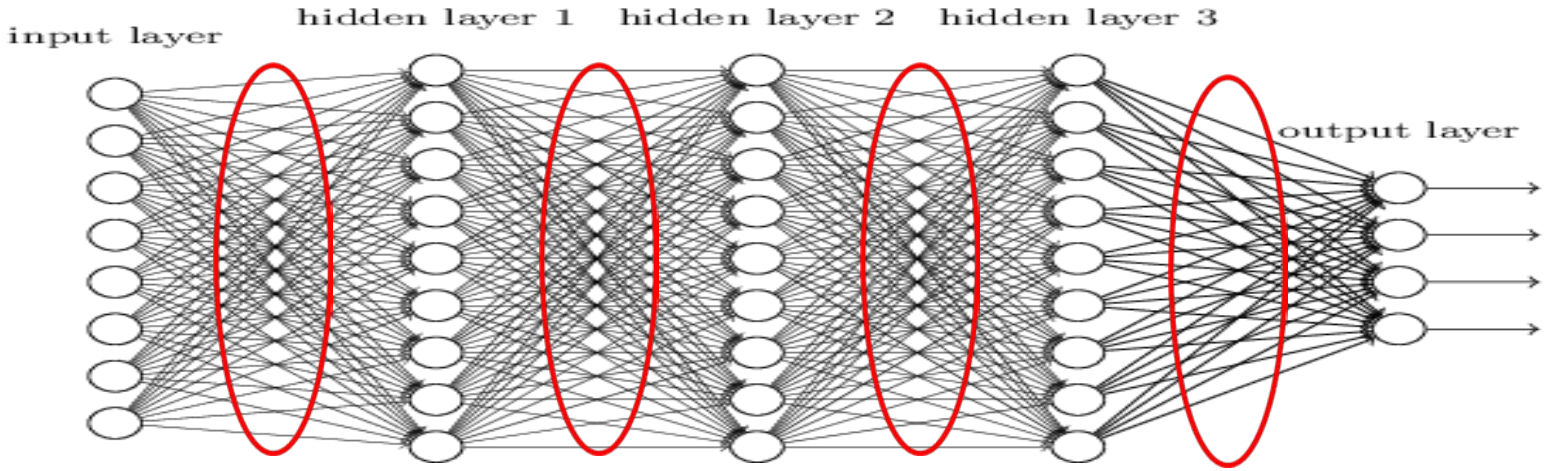
Specific care for data representation

Structure the network?

- Can we put any structure reducing the space of exploration and providing useful properties (invariance, robustness, sequentiality…)?

$$y = s(w_{13}\, s(w_{11}x_1 + w_{21}x_2 - w_{01}) + w_{23}\, s(w_{12}x_1 + w_{22}x_2 - w_{02}) - w_{03})$$

(z_1 over $w_{11}x_1 + w_{21}x_2 - w_{01}$; z_2 over $w_{12}x_1 + w_{22}x_2 - w_{02}$; z_3 over the whole expression)



Enabling factors

- Why do it now ? Before 2006, training deep networks was unsuccessful because of practical aspects
  - faster CPU's
  - parallel CPU architectures
  - advent of GPU computing
  - Advances in ML/Optim (1995 -> 2005)

- Hinton, Osindero & Teh « A Fast Learning Algorithm for Deep Belief Nets », *Neural Computation*, 2006
- Bengio, Lamblin, Popovici, Larochelle « Greedy Layer-Wise Training of Deep Networks », *NIPS'2006*
- Ranzato, Poultney, Chopra, LeCun « Efficient Learning of Sparse Representations with an Energy-Based Model », *NIPS'2006*

- Results…
  - 2009, sound, interspeech +~24%
  - 2011, text, +~15% without linguistic at all
  - 2012, images, ImageNet +~20%
  - 2020, molecules/graphs, AlphaFold, +~24%
    (sorry in French, https://www.youtube.com/watch?v=OGewxRMME8o)

Slide from Yoshua Bengio

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Invariance to spatial changes

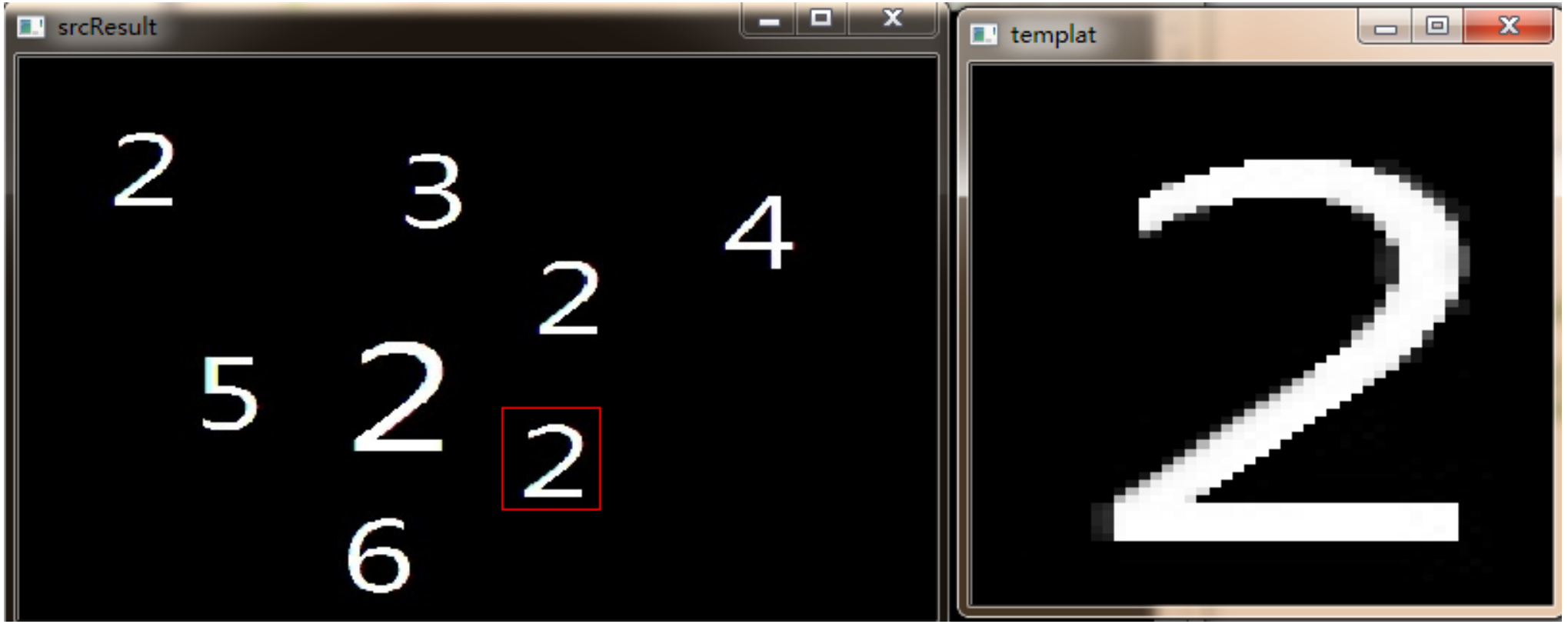
Convolutional Neural Networks (aka CNN, ConvNet)

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Do you know template matching?

Single target matching results:



I F

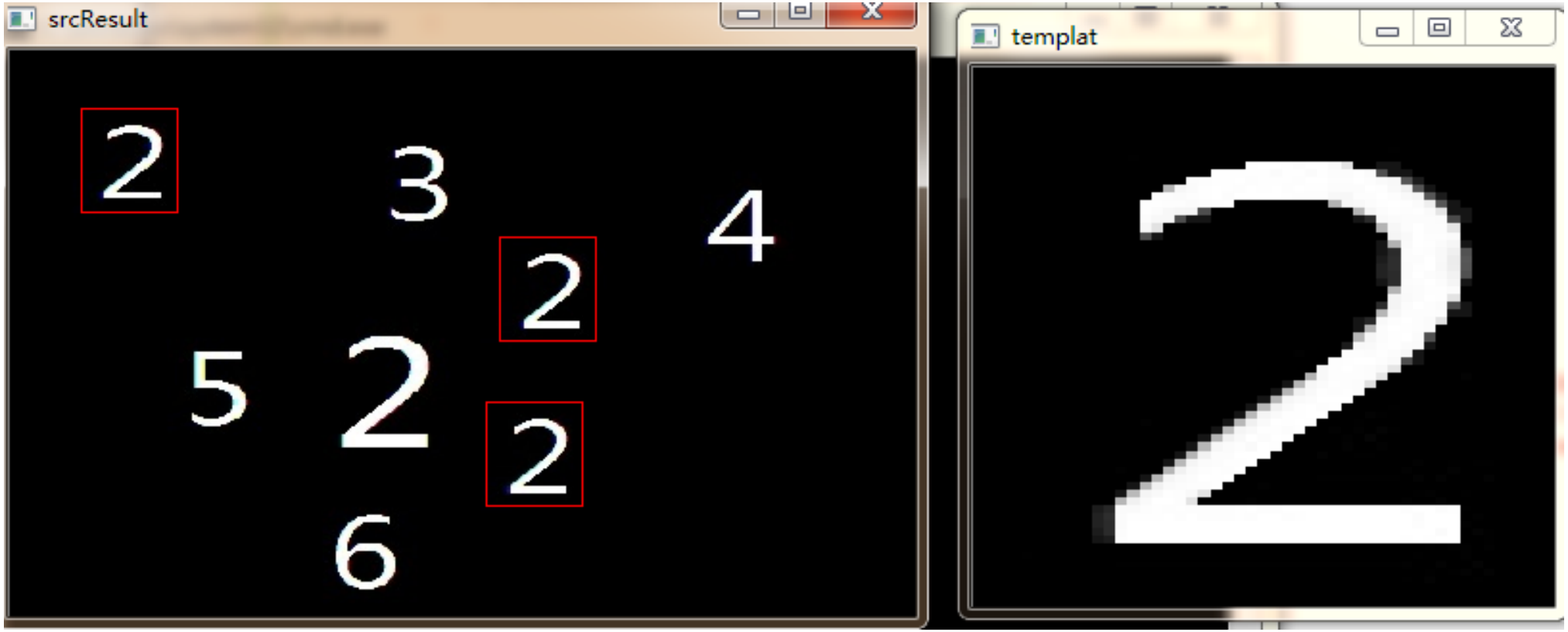
(picture from https://programmer.group/5ca5086fed10b.html)

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Do you know template matching?

Multi-objective matching result:



I F

(picture from https://programmer.group/5ca5086fed10b.html)

UNIVERSITÉ CÔTE D'AZUR
univ-cotedazur.fr

Do you know template matching?

How to compute Template matching? Cross-correlation!

$$F \circ I(x,y) = \sum_{j=-N}^{N} \sum_{i=-N}^{N} F(i,j) I(x+i, y+j)$$

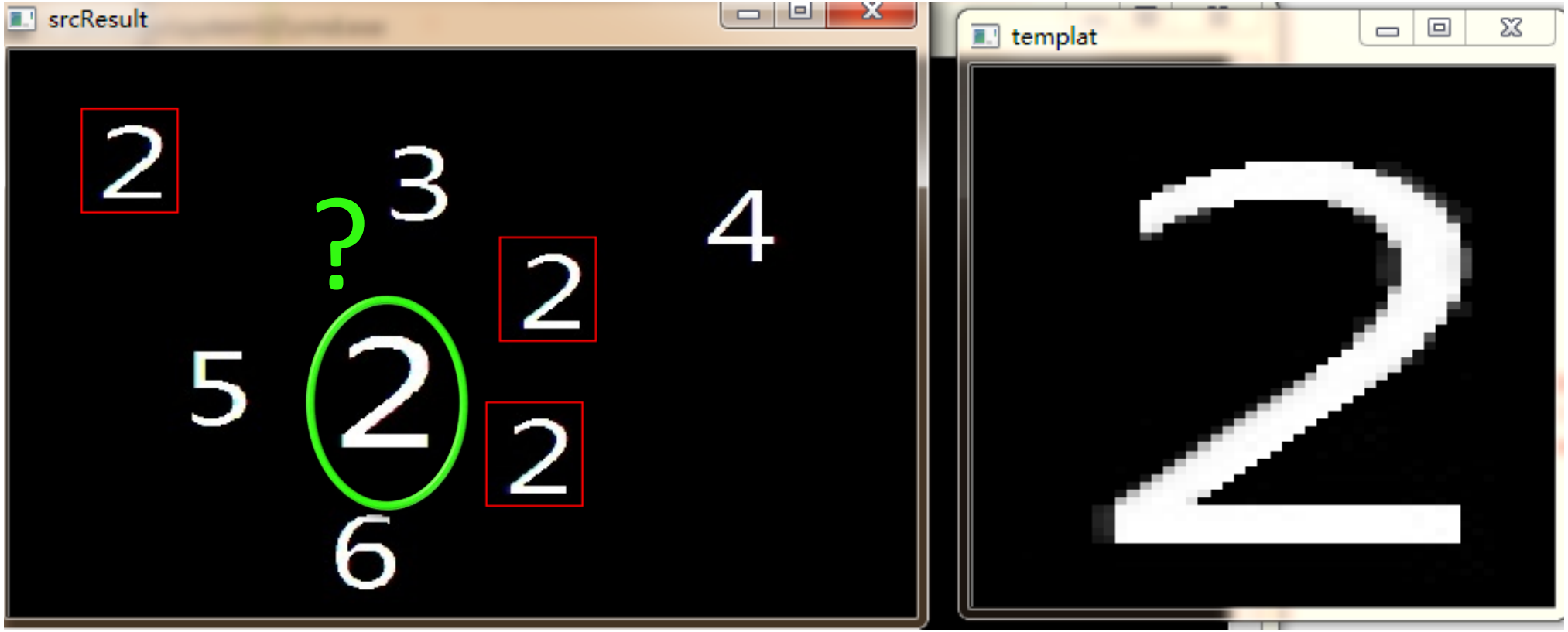
When this is maximal, the Filter is perfectly positioned with respect to the pattern to detect.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Do you know template matching?

Multi-objective matching result:



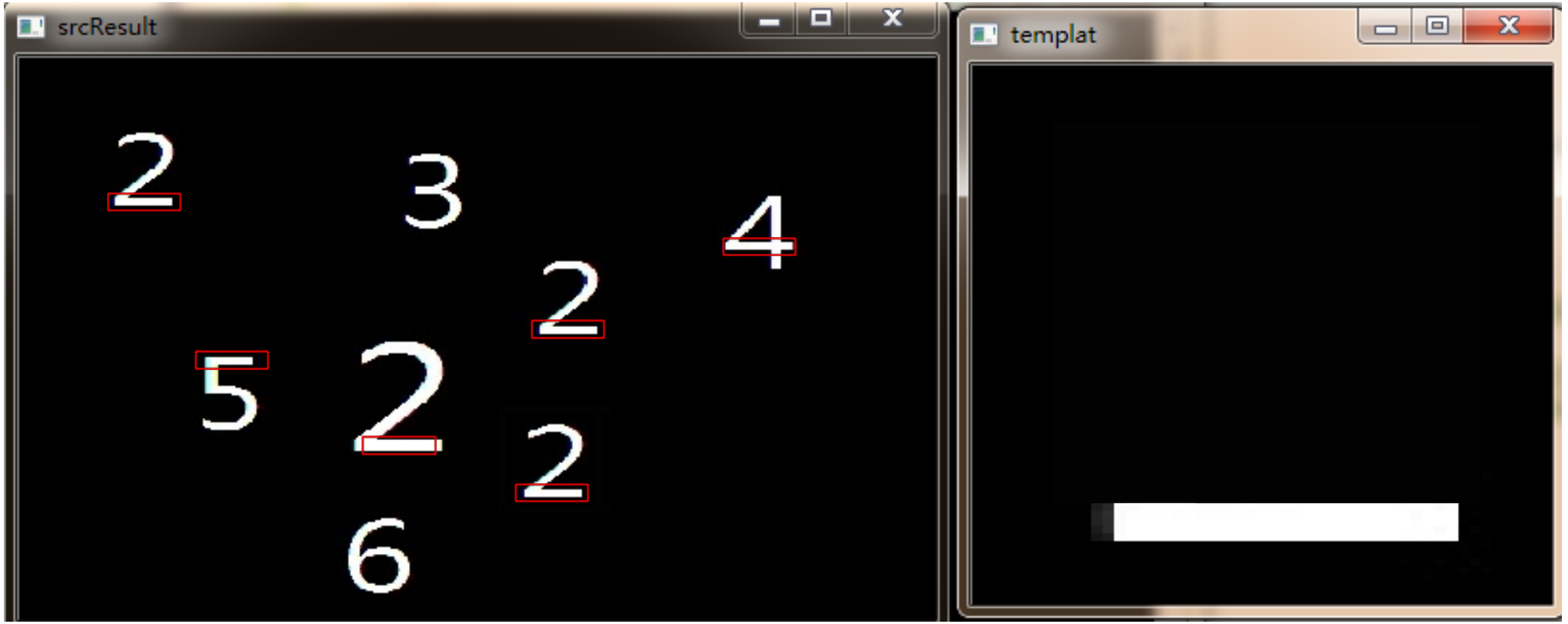
I F

(picture from https://programmer.group/5ca5086fed10b.html)

UNIVERSITÉ CÔTE D'AZUR
univ-cotedazur.fr

Are there more universal templates?

Multi-objective matching result:



I F

(picture from https://programmer.group/5ca5086fed10b.html)

UNIVERSITÉ CÔTE D'AZUR
univ-cotedazur.fr

Are there more universal templates?

Multi-objective matching result:

I F

(picture from https://programmer.group/5ca5086fed10b.html)

UNIVERSITÉ CÔTE D'AZUR
univ-cotedazur.fr

How to locate these templates?

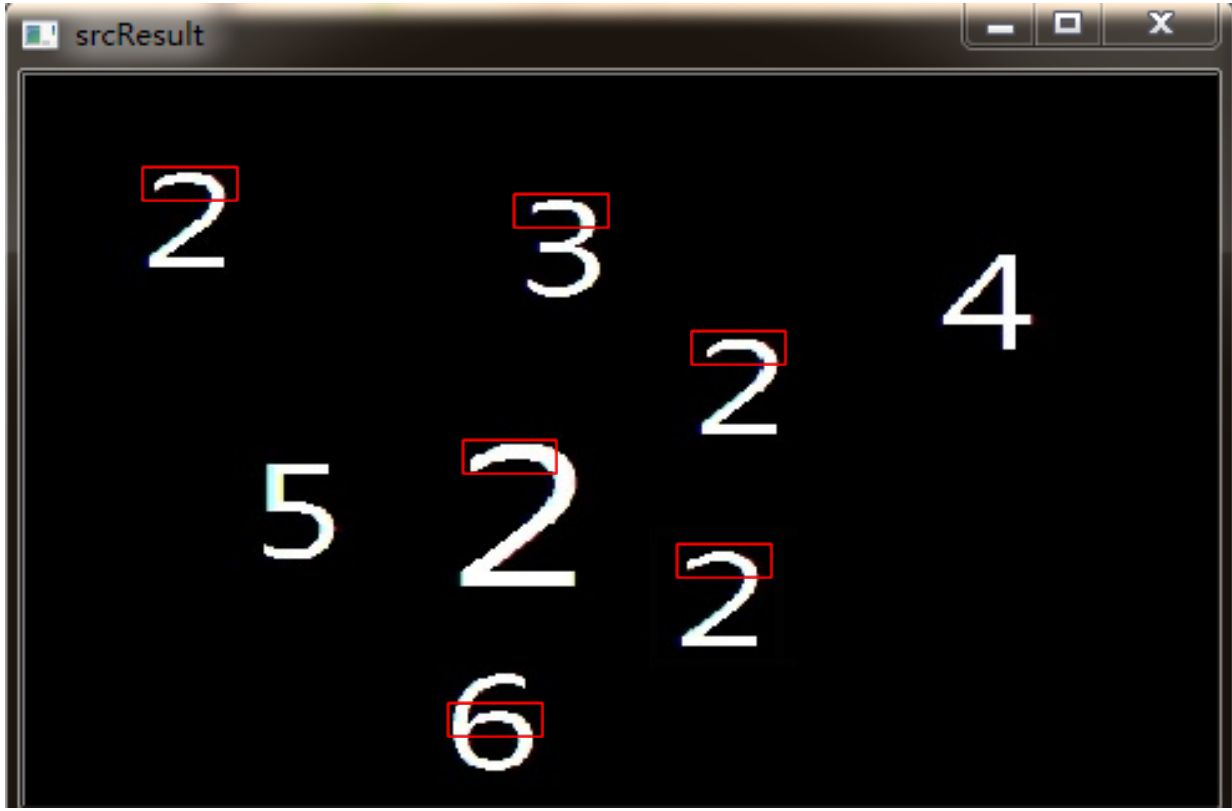
By scanning I, we acquire **translation invariance**

I F

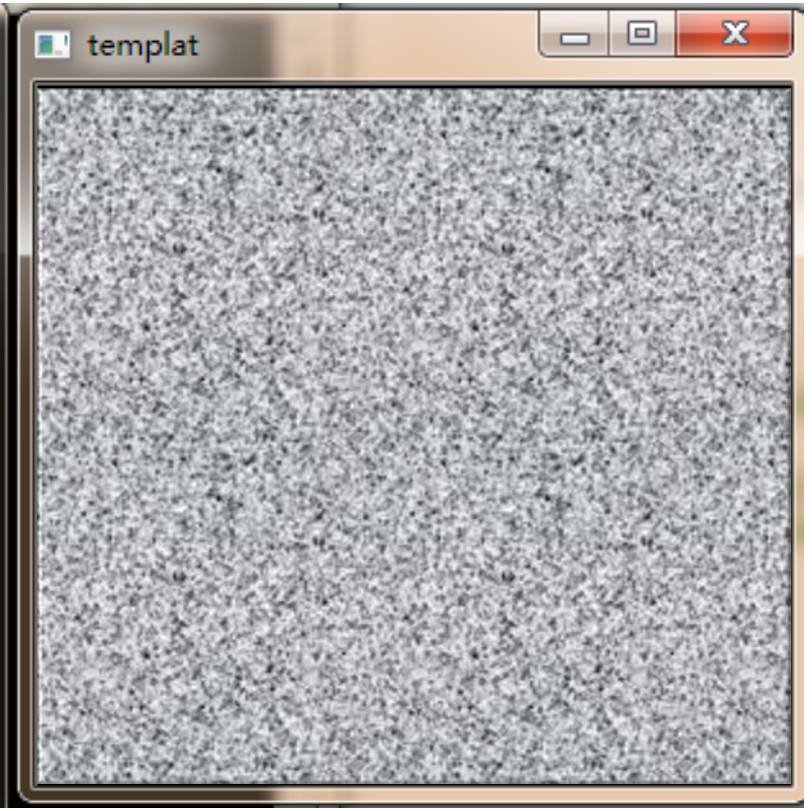
UNIVERSITÉ CÔTE D'AZUR
univ-cotedazur.fr

How to define these templates?

By learning what filter maximizes **local cross-correlations**



I



F

UNIVERSITÉ CÔTE D'AZUR
univ-cotedazur.fr

Cross-correlation or Convolution?

- How to define these templates?

By learning what filter maximizes **local cross-correlations**

$$F \circ I(x,y) = \sum_{j=-N}^{N} \sum_{i=-N}^{N} F(i,j) I(x+i, y+j)$$

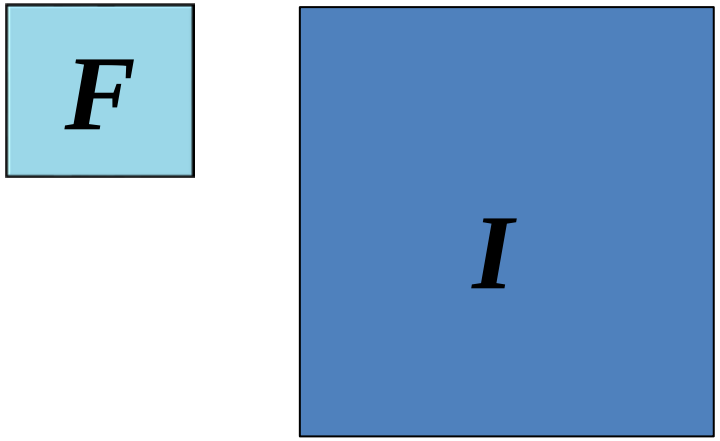
Or **local convolutions**

$$F * I(x,y) = \sum_{j=-N}^{N} \sum_{i=-N}^{N} F(i,j) I(x-i, y-j)$$

UNIVERSITÉ CÔTE D'AZUR
univ-cotedazur.fr

Cross-correlation or Convolution?

- Convolution:
  - Flip the filter in both dimensions (bottom to top, right to left)
  - Then apply cross-correlation



UNIVERSITÉ CÔTE D'AZUR
univ-cotedazur.fr

Cross-correlation or Convolution?

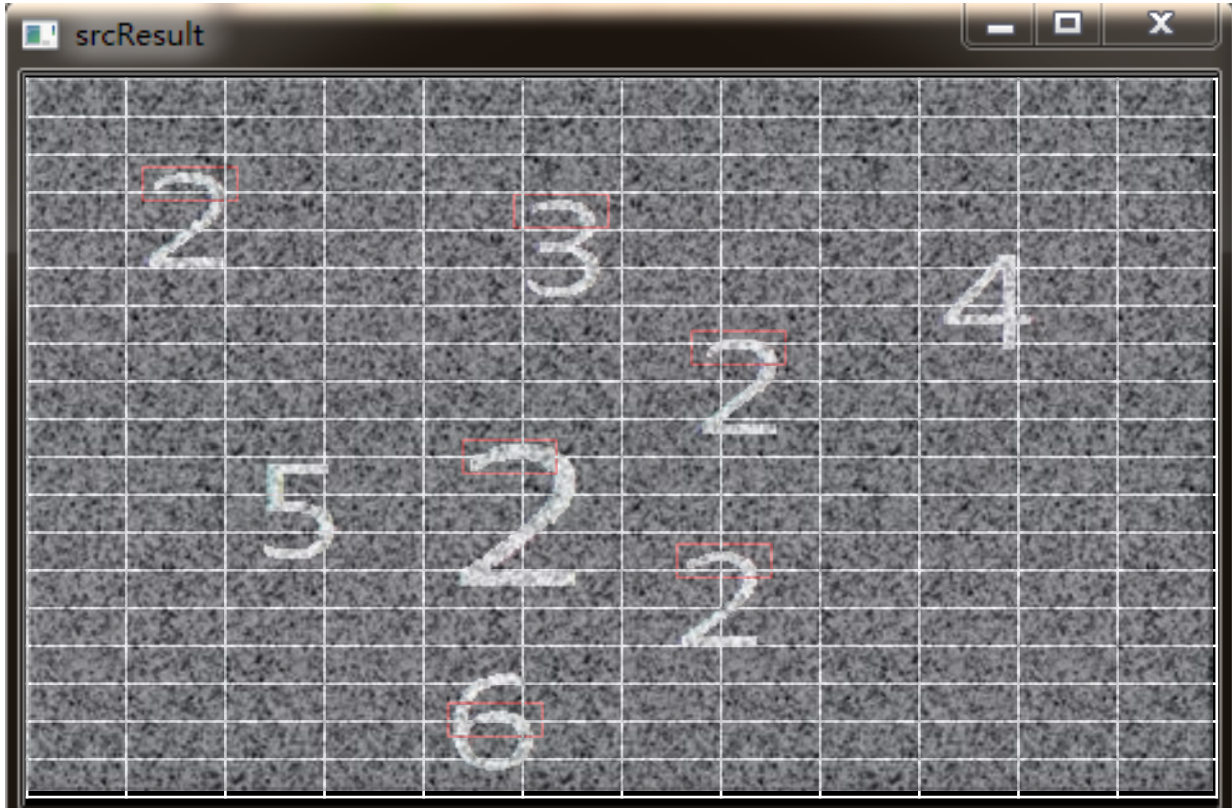
- Convolution:
  - Flip the filter in both dimensions (bottom to top, right to left)
  - Then apply cross-correlation

- Actually, The Convolutional operation widely used in Convolution Neural Network (CNN) is commonly and largely considered as a **misnomer**
  - The operation that is used is strictly speaking a correlation instead of convolution.
  - Both the operators have **translational invariance** and **locality**.

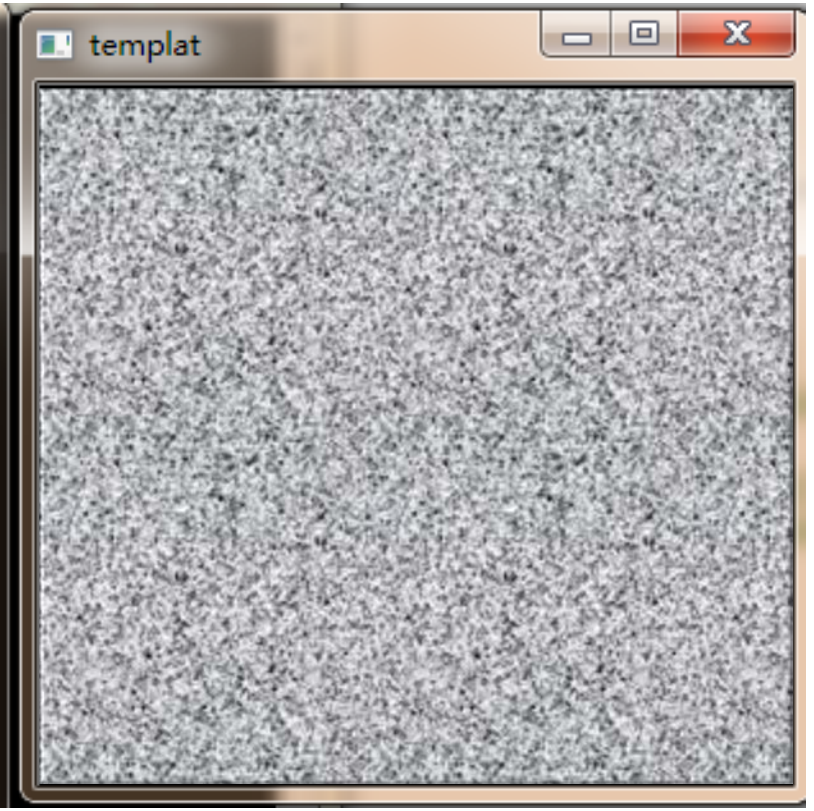
UNIVERSITÉ CÔTE D'AZUR
univ-cotedazur.fr

How to define these templates?

And instead of scanning I, let us just **duplicate** the template **into a grid**.



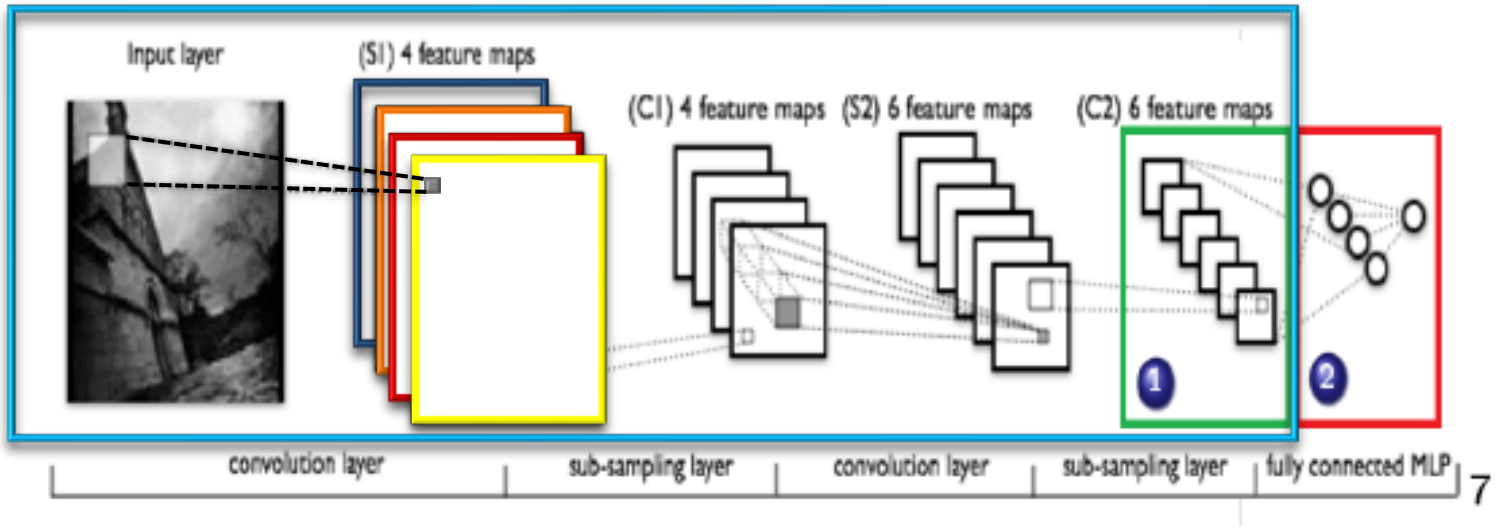
I



F

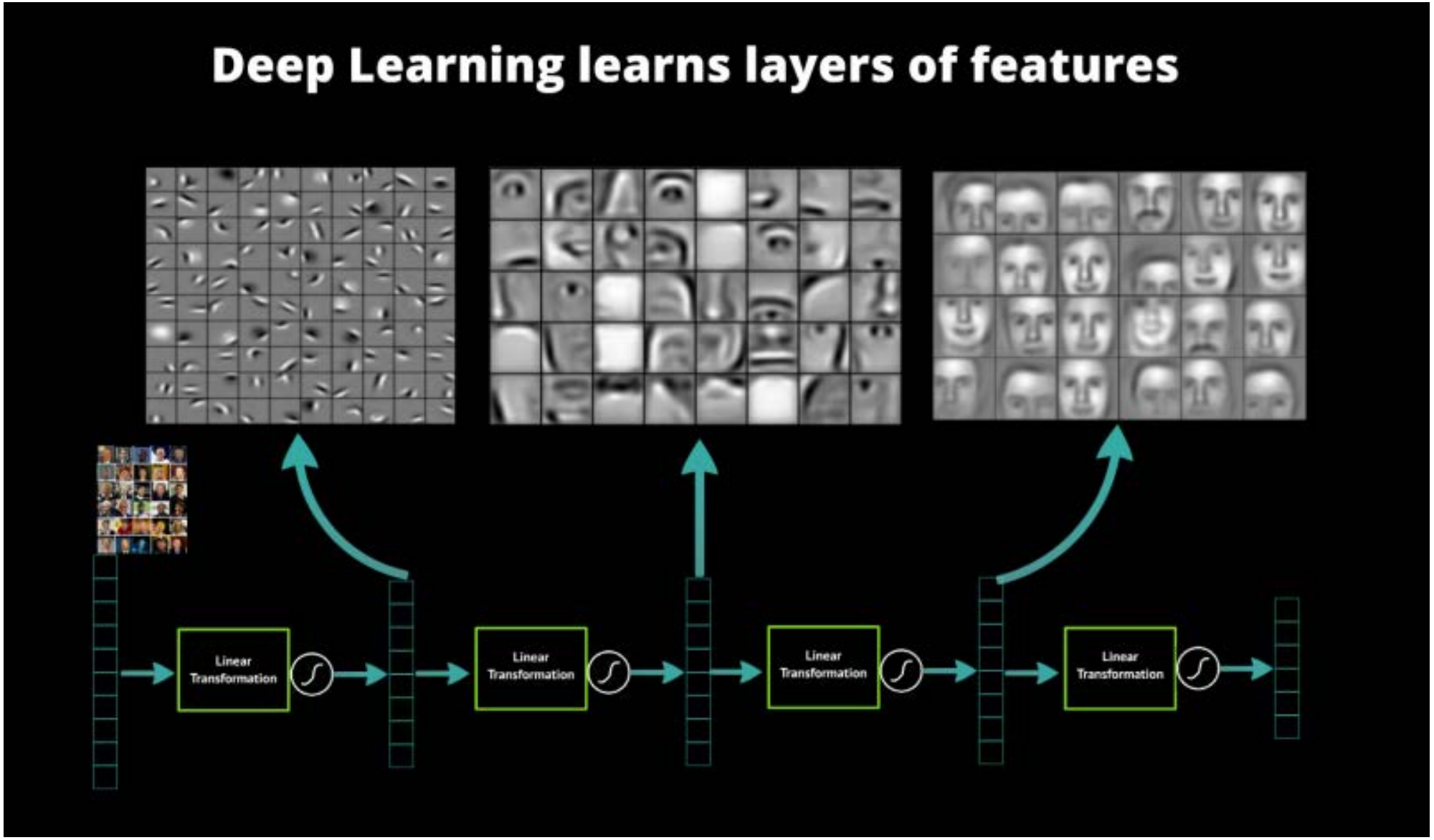
Deep representation by CNN

- feature map = result of the convolution
- convolution with a filter extract characteristics (*edge detectors*)
- extract parallelised characteristics at each layer

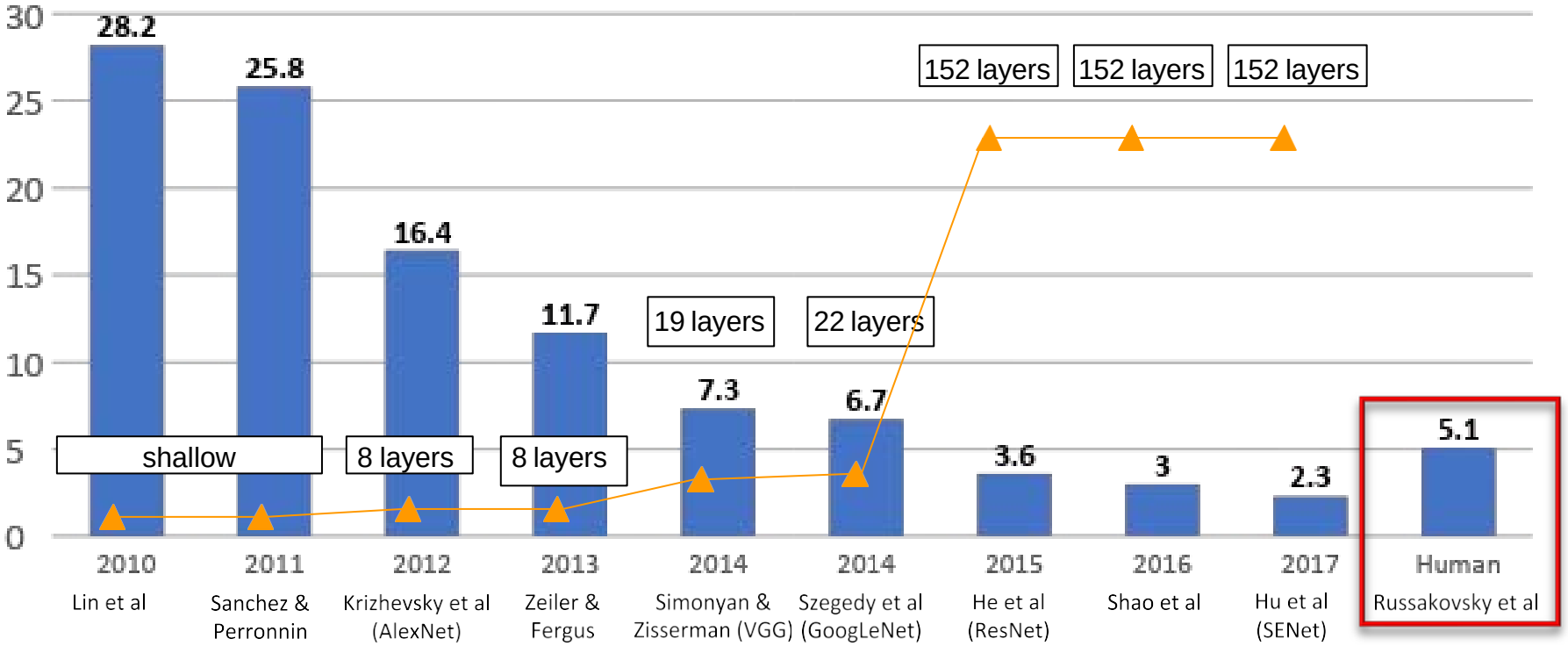


1. final representation of our data
2. classifier (MLP)

Deep representation by CNN



ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Invariance to sequential changes

Recurrent Neural Networks (aka RNN, LSTM / GRU)

Borrowed from COLAH's Blog, http://colah.github.io/posts/2015-08-Understanding-LSTMs/
From "Introduction to Deep Learning", by Professor Qiang Yang, from The Hong Kong University of Science and Technology
And from lecture on RNN of Prof Adriana Kovashka

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Some pre-RNN captioning results

This is a picture of one sky, one road and one sheep. The gray sky is over the gray road. The gray sheep is by the gray road.

This is a picture of two dogs. The first dog is near the second furry dog.

Here we see one road, one sky and one bicycle. The road is near the blue sky, and near the colorful bicycle. The colorful bicycle is within the blue sky.

Kulkarni et al., CVPR 2011

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Results with Recurrent Neural Networks

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

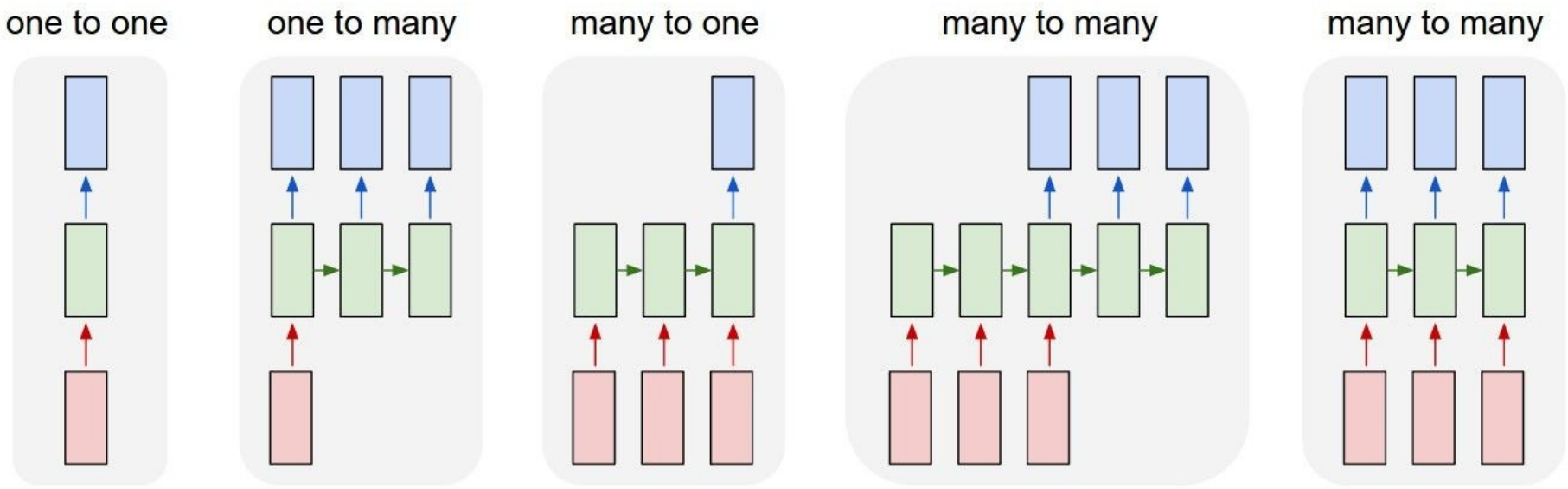
"boy is doing backflip on wakeboard."

Karpathy and Fei-Fei, CVPR 2015

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Recurrent Networks offer a lot of flexibility:



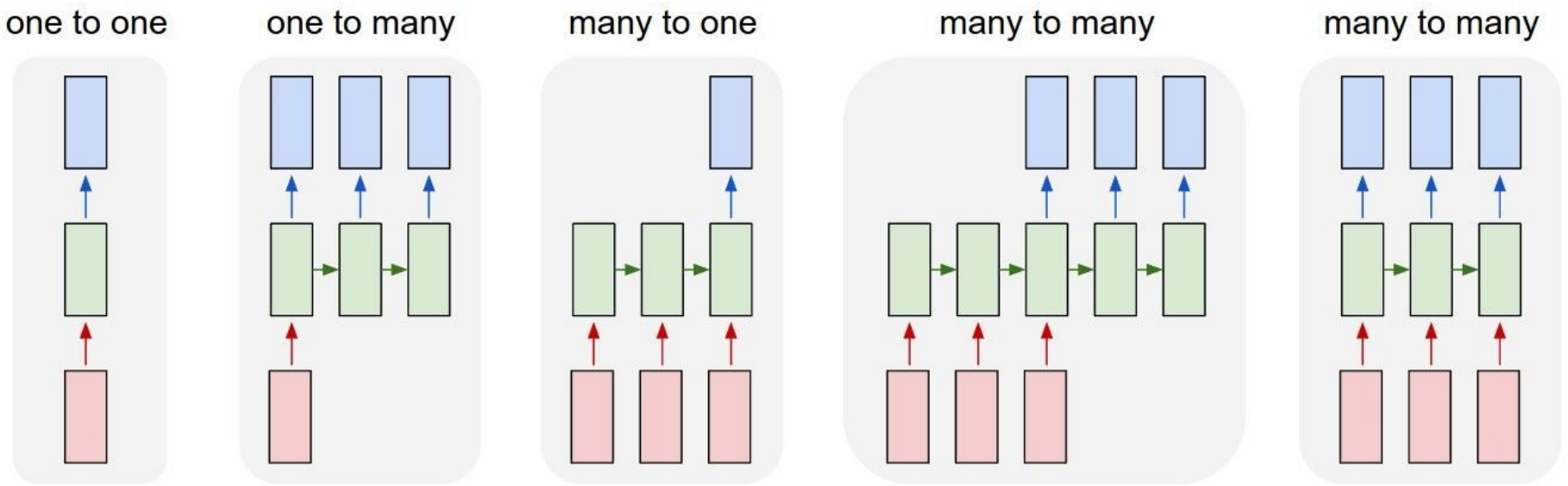
vanilla neural networks

Andrej Karpathy

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Recurrent Networks offer a lot of flexibility:



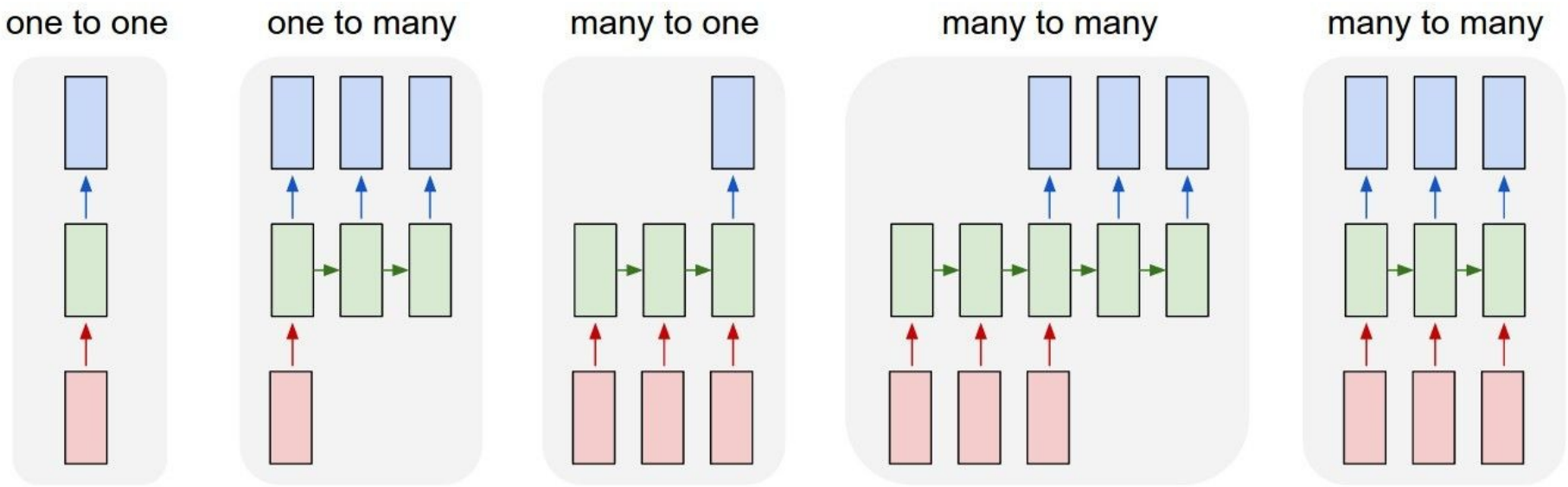
e.g. **image captioning**
image -> sequence of words

Andrej Karpathy

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Recurrent Networks offer a lot of flexibility:



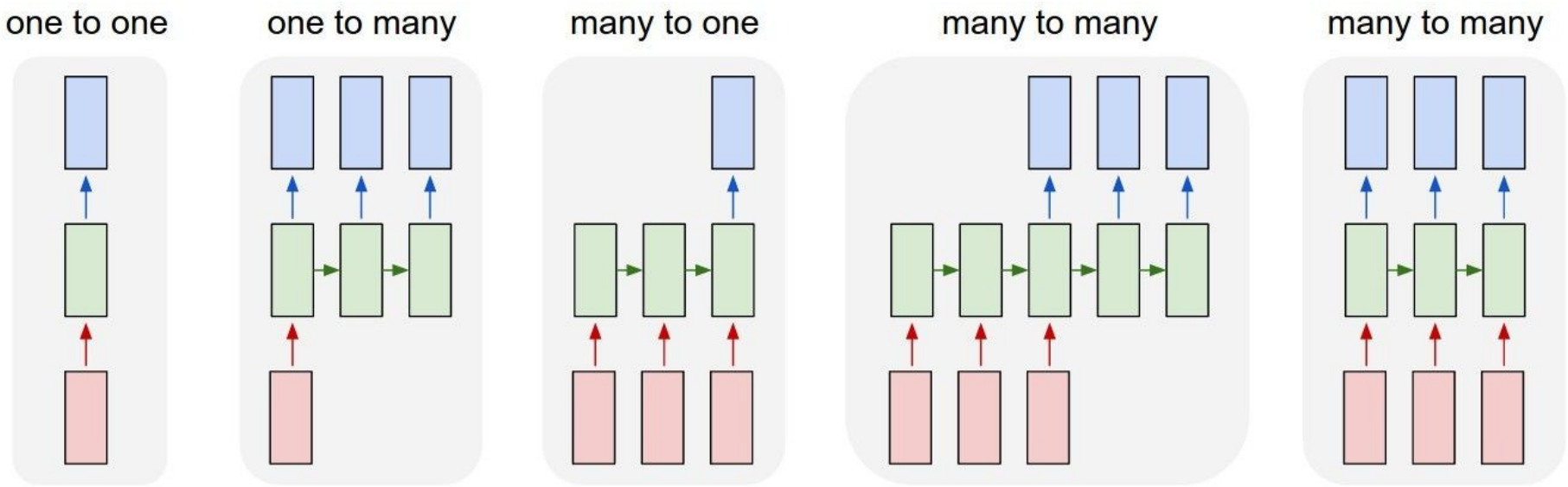
e.g. **sentiment classification**
sequence of words -> sentiment

Andrej Karpathy

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Recurrent Networks offer a lot of flexibility:



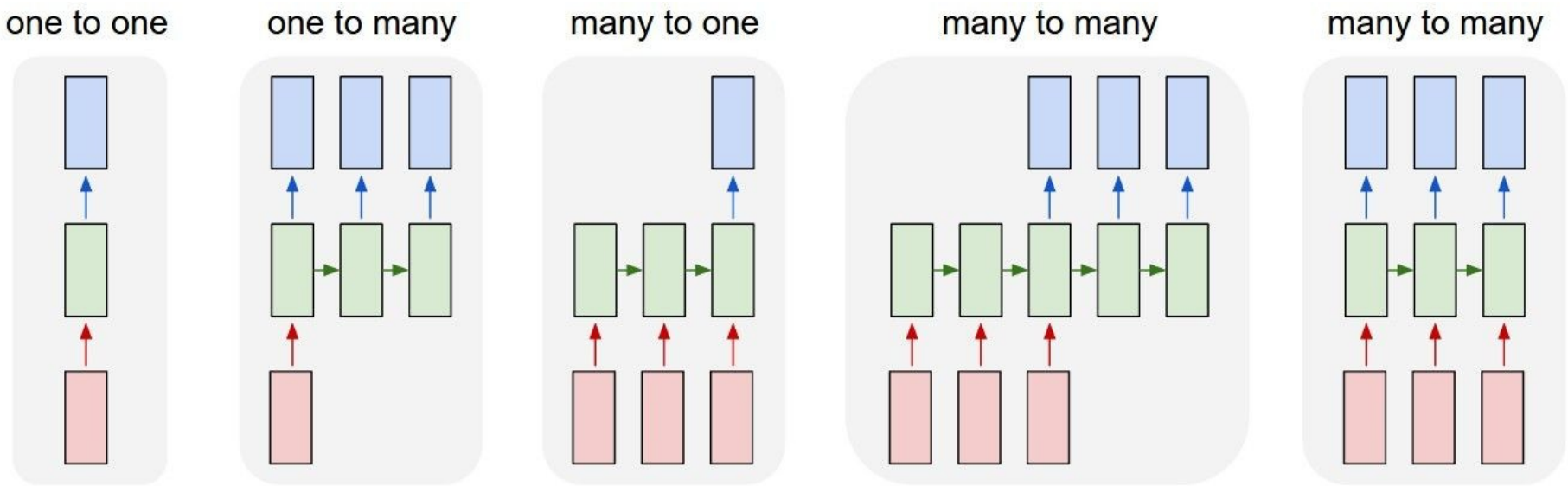
e.g. **machine translation**
seq of words -> seq of words

Andrej Karpathy

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Recurrent Networks offer a lot of flexibility:



e.g. **video classification on frame level**

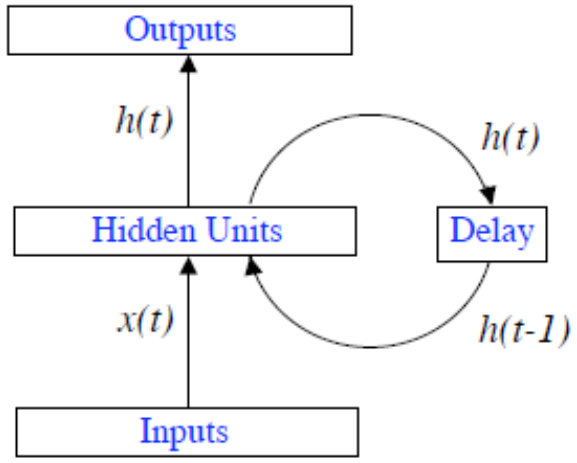
Andrej Karpathy

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

What are RNNs?

Recurrent neural networks (RNNs) are connectionist models with the ability to selectively pass information across sequence steps, while processing sequential data one element at a time.



The simplest form of ***fully recurrent neural network*** is an MLP with the previous set of hidden unit activations feeding back into the network along with the inputs

Allow a 'memory' of previous inputs to persist in the network's internal state, and thereby influence the network output

$$h(t) = f_H(W_{IH}x(t) + W_{HH}h(t-1))$$

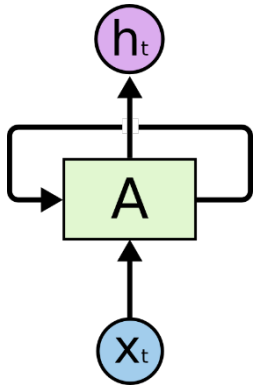
$$y(t) = f_O(W_{HO}h(t))$$

f_H and f_O are the activation function for hidden and output unit; W_{IH}, W_{HH}, and W_{HO} are connection weight matrices which are learnt by training

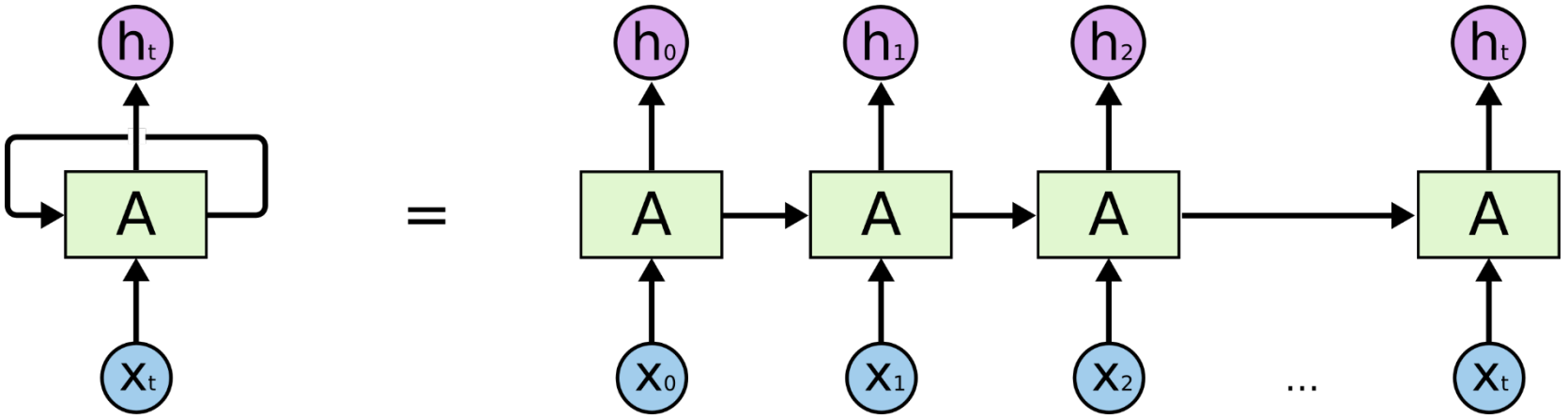
UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Recurrent Neural Networks have loops.



An unrolled recurrent neural network.



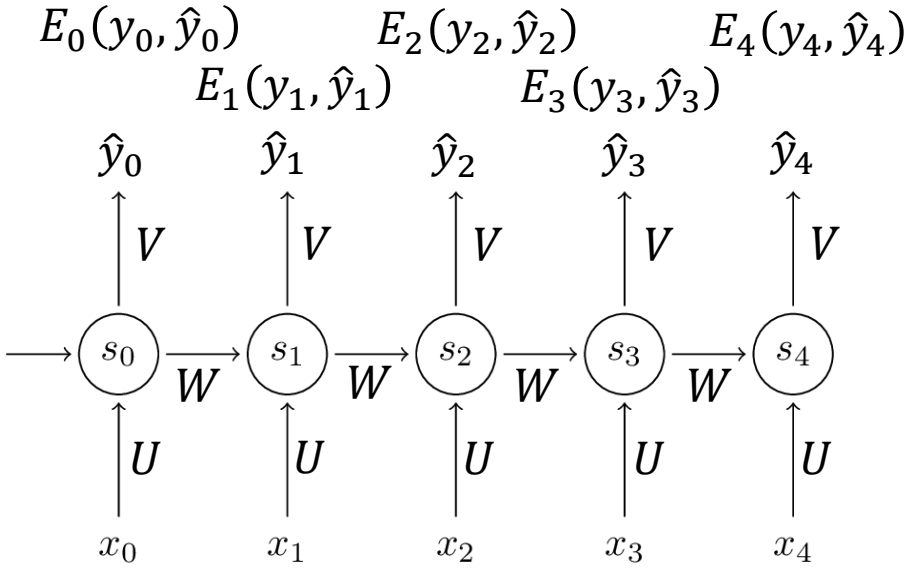
In the last few years, there have been incredible success applying RNNs to a variety of problems: speech recognition, language modeling, translation, image captioning…

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

What are RNNs?

- The recurrent network can be converted into a feed-forward network by ***unfolding over time***



An unfolded recurrent network. Each node represents a layer of network units at a single time step.
The weighted connections from the input layer to hidden layer are labelled 'W_{IH}', those from the hidden layer to itself (i.e. the recurrent weights) are labelled 'W_{HH}' and the hidden to output weights are labelled 'W_{HO}'.
Note that the same weights are reused at every time step. Bias weights are omitted for clarity.

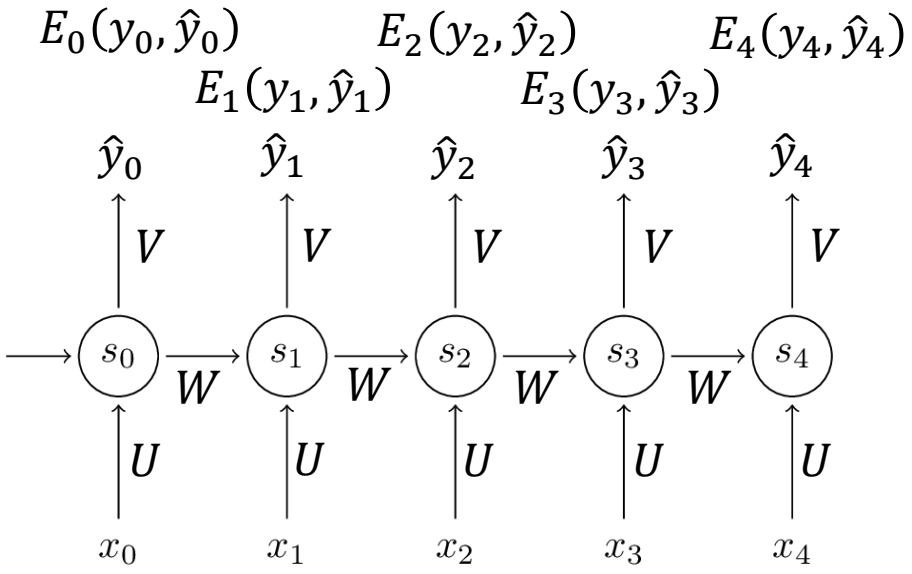
UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

What are RNNs?

- Training RNNs (determine the parameters)

  Back Propagation Through Time (BPTT) is often used to learn the RNN
  BPTT is an extension of the back-propagation (BP)



- The output of this RNN is $\hat{y}_t$

$$s_t = \tanh(Ux_t + Ws_{t-1})$$
$$\hat{y}_t = softmax(Vs_t)$$

- The loss/error function of this network is

$$E_t(y_t, \hat{y}_t) = -y_t \log \hat{y}_t$$ → The error at each time step

$$E(y, \hat{y}) = \sum_t E_t(y_t, \hat{y}_t)$$ → the total loss is the sum of the errors at each time step

UNIVERSITÉ CÔTE D'AZUR

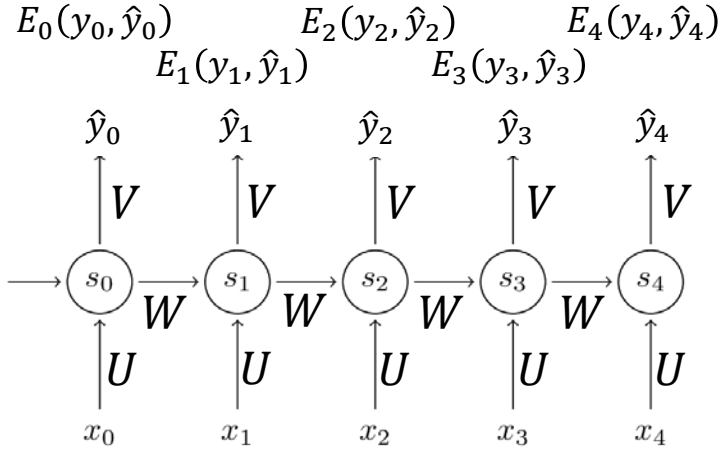
univ-cotedazur.fr

What are RNNs?

- Training RNNs (determine the parameters)
  - ✓ The gradients of the error with respect to our parameters

Just like we sum up the errors, we also sum up the gradients at each time step for one training example. For parameter W, the gradient is

$$\frac{\partial E}{\partial W} = \sum_t \frac{\partial E_t}{\partial W}$$



  - ✓ The gradient at each time step

we use time 3 as an example

$$\frac{\partial E_3}{\partial W} = \frac{\partial E_3}{\partial \hat{y}_3}\frac{\partial \hat{y}_3}{\partial s_3}\frac{\partial s_3}{\partial W}$$ → Chain Rule

$$s_3 = \tanh(Ux_3 + Ws_2)$$ → s_3 depends on W and s_2, we cannot simply consider s_2 a constant

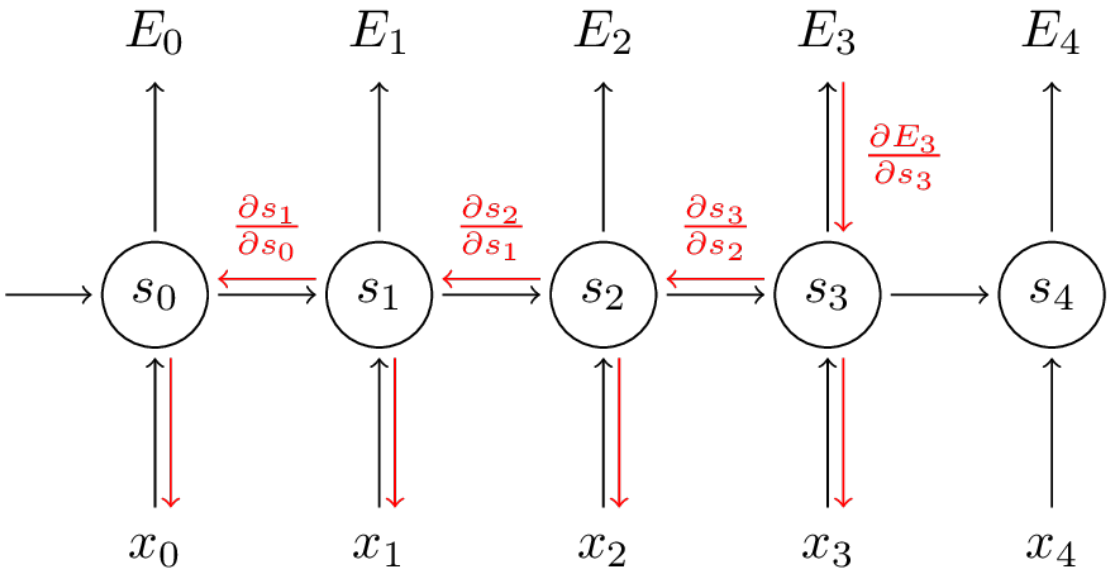
$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial \hat{y}_3}\frac{\partial \hat{y}_3}{\partial s_3}\frac{\partial s_3}{\partial s_k}\frac{\partial s_k}{\partial W}$$ → Apply Chain Rule again on s_k

What are RNNs?

- Training RNNs (determine the parameters)

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \frac{\partial s_3}{\partial s_k} \frac{\partial s_k}{\partial W} \longrightarrow \frac{\partial E_3}{\partial W} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \prod_{j=k+1}^{3} \frac{\partial s_j}{\partial s_{j-1}} \frac{\partial s_k}{\partial W}$$

Because W is used in every step up to the output we care about, we need to backpropagate gradients from $t = 3$ through the network all the way to $t = 0$



UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

What are RNNs?

- The vanishing gradient problem

To understand why, let's take a closer look at the gradient we calculated above:

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \underbrace{\prod_{j=k+1}^{3} \frac{\partial s_j}{\partial s_{j-1}}} \frac{\partial s_k}{\partial W}$$

Because the layers and time steps of deep neural networks relate to each other through multiplication, derivatives are susceptible to vanishing

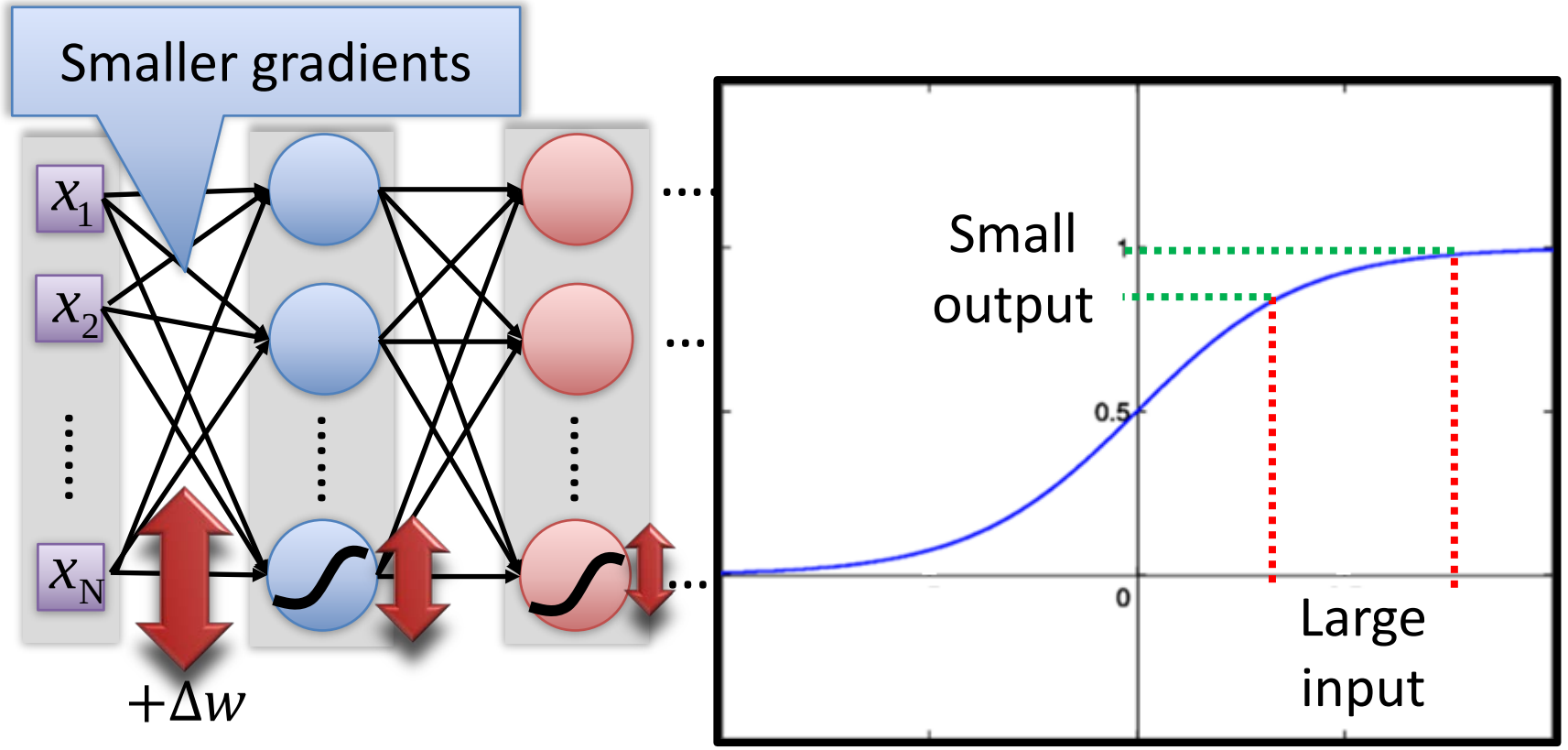
Gradient contributions from "far away" steps become zero, and the state at those steps **does not contribute** to what you are learning: You end up not learning long-range dependencies.

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Recipe of Deep Learning

- ***Vanishing Gradient Problem***



Intuitive way to compute the derivatives …

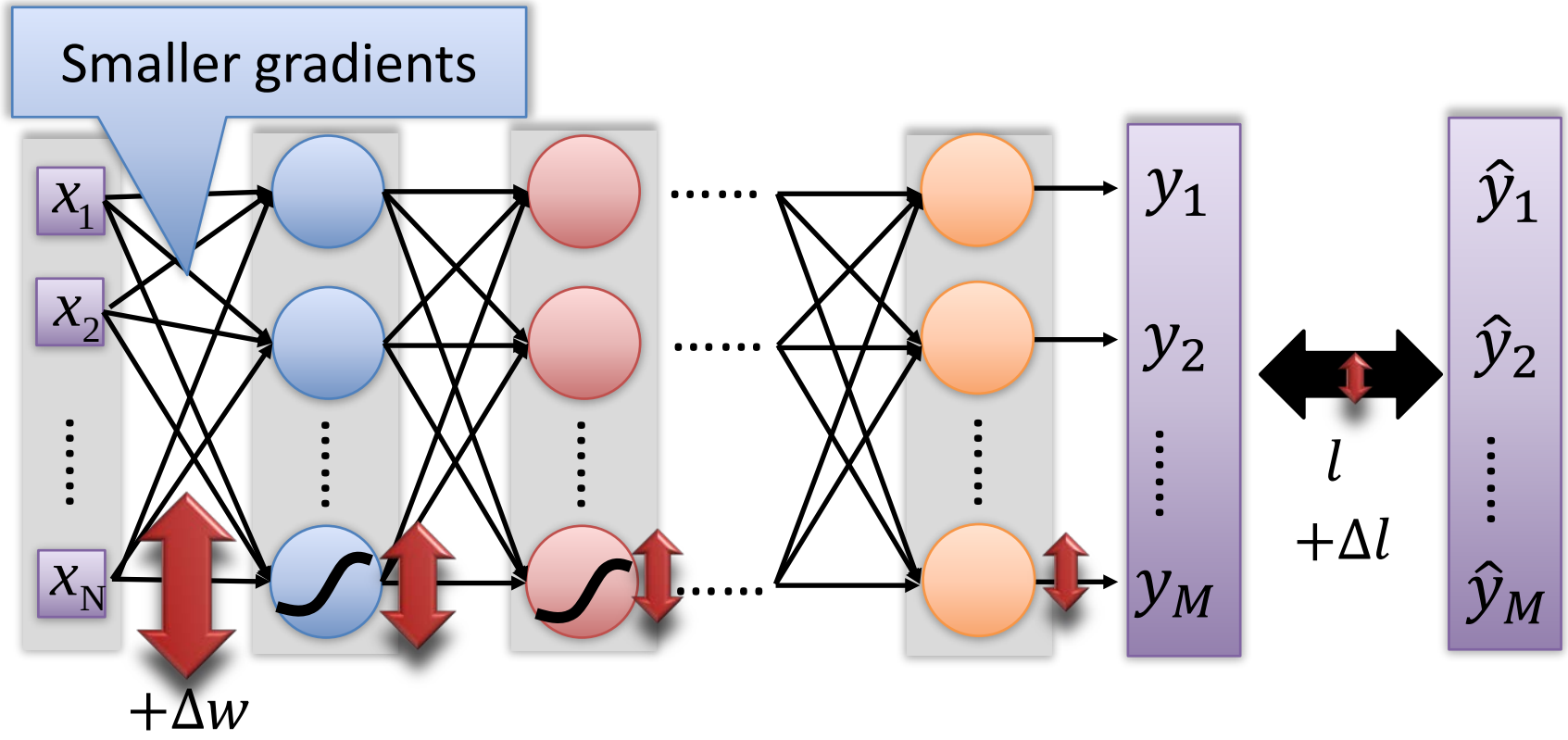
$$\frac{\partial l}{\partial w} =? \frac{\Delta l}{\Delta w}$$

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Recipe of Deep Learning

- ***Vanishing Gradient Problem***



Intuitive way to compute the derivatives …

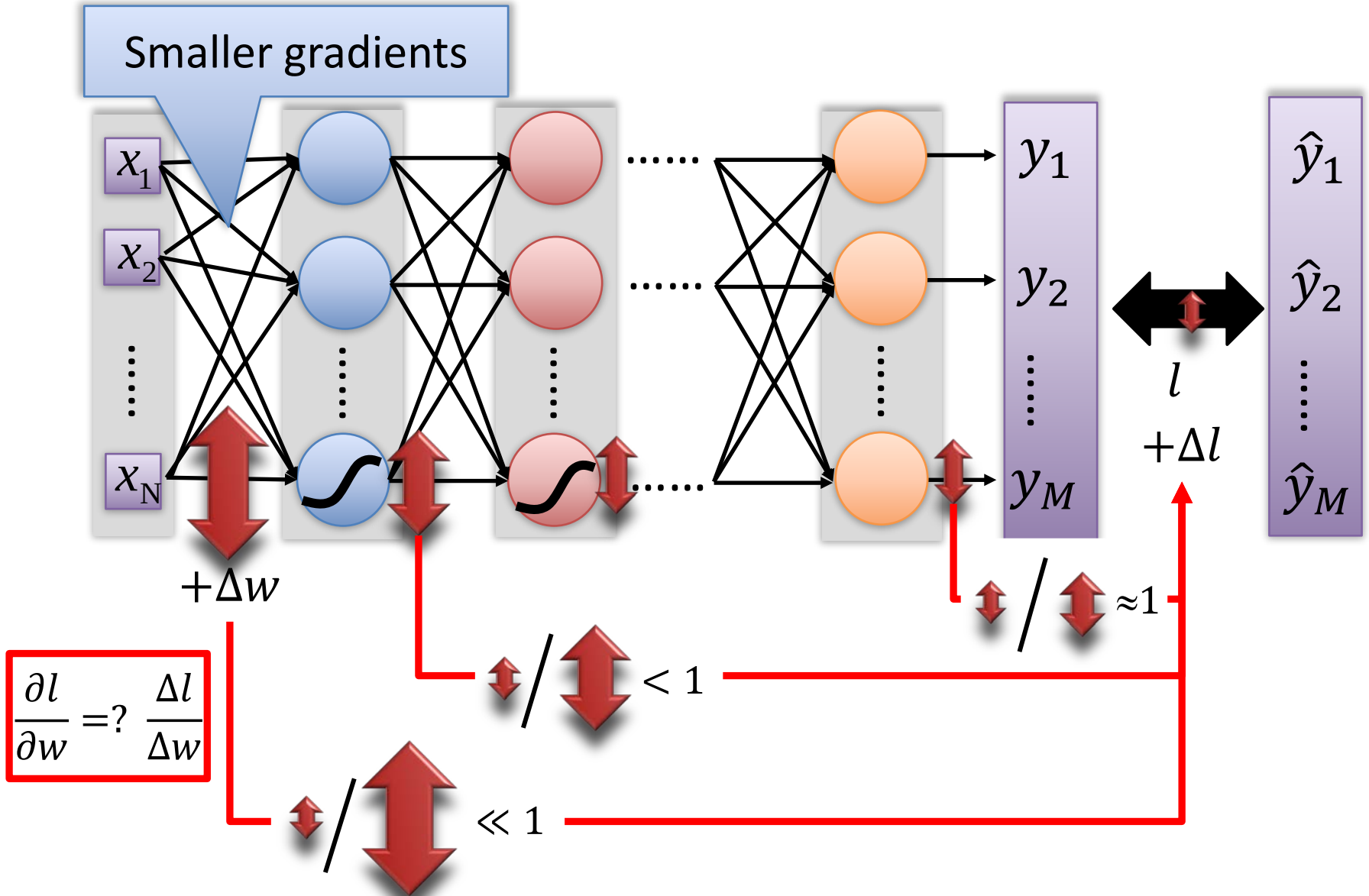
$$\frac{\partial l}{\partial w} =? \frac{\Delta l}{\Delta w}$$

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Recipe of Deep Learning

- ***Vanishing Gradient Problem***



UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

What are RNNs?

- The vanishing gradient problem

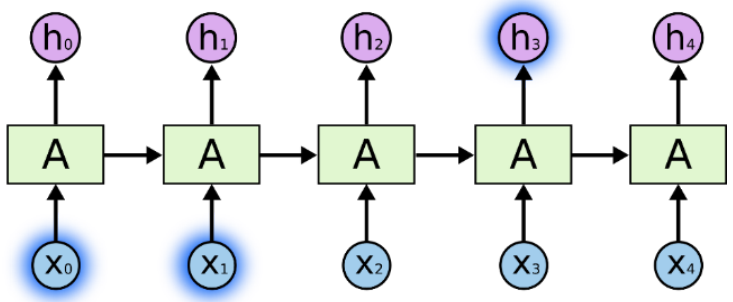
To understand why, let's take a closer look at the gradient we calculated above:

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \underbrace{\prod_{j=k+1}^{3} \frac{\partial s_j}{\partial s_{j-1}}} \frac{\partial s_k}{\partial W}$$

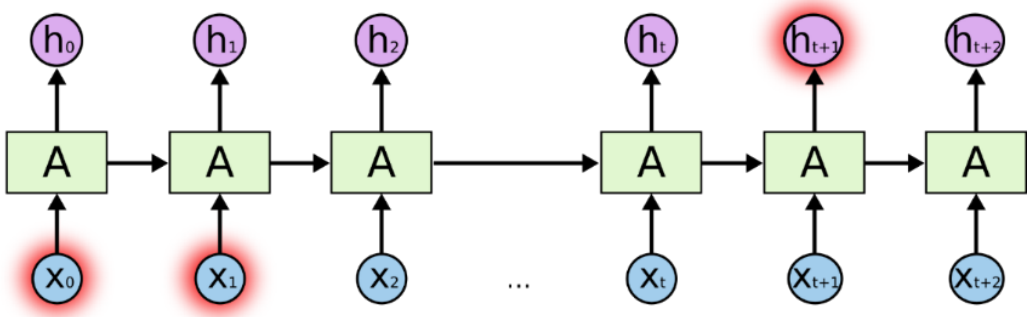
Because the layers and time steps of deep neural networks relate to each other through multiplication, derivatives are susceptible to vanishing

Gradient contributions from "far away" steps become zero, and the state at those steps **does not contribute** to what you are learning: You end up not learning long-range dependencies.

The Problem of Long-Term Dependencies

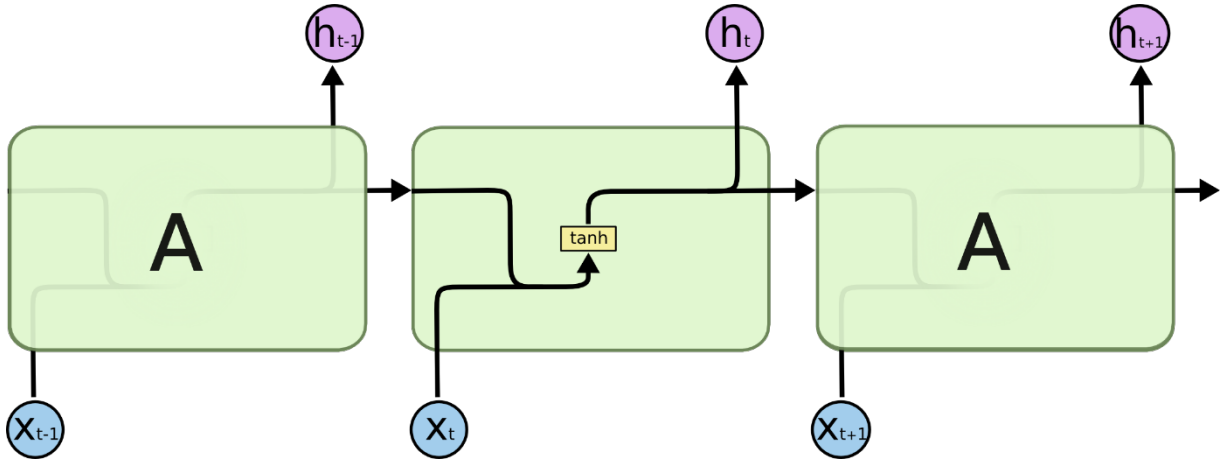


*If we are trying to predict the last word in "the clouds are in the **(sky)**," we don't need any further context – it's pretty obvious the next word is going to be sky.*



*Consider trying to predict the last word in the text "I grew up in France… I speak fluent **(French)**." Recent information suggests that the next word is probably the name of a language, but if we want to narrow down which language, we need the context of France, from further back.*

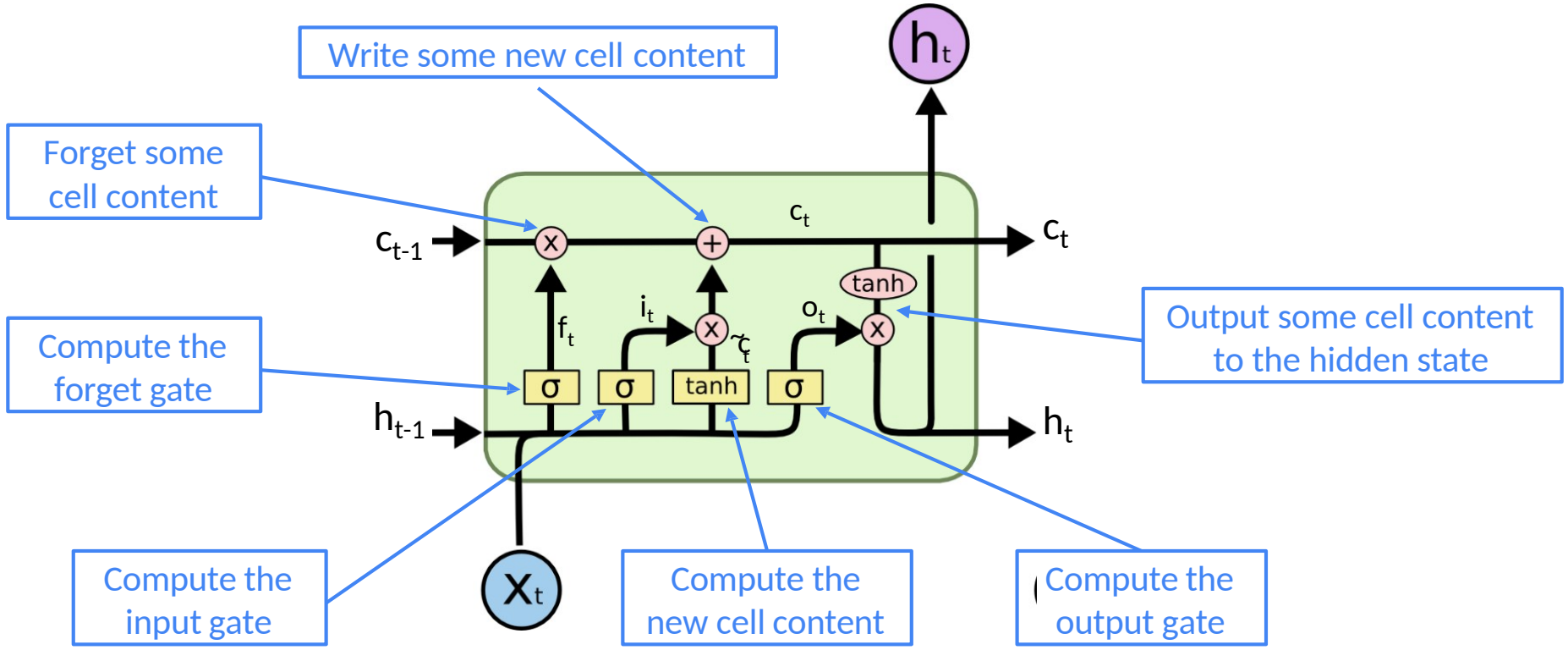
The repeating module in a standard RNN contains a single layer.



Unfortunately, as that gap grows, RNNs become unable to learn to connect the information (cf. vanishing gradients)
The problem was explored in depth by Hochreiter (1991) and Bengio, et al. (1994), who found some pretty fundamental reasons why it might be difficult.
Thankfully, LSTMs do not have this problem! (Hochreiter & Schmidhuber, 1997)

Review on your own: Long Short-Term Memory (LSTM)

You can think of the LSTM equations visually like this:



Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Abigail See

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Review on your own: Long Short-Term Memory (LSTM)

We have a sequence of inputs $\boldsymbol{x}^{(t)}$, and we will compute a sequence of hidden states $\boldsymbol{h}^{(t)}$ and cell states $\boldsymbol{c}^{(t)}$. On timestep t:

Sigmoid function: all gate values are between 0 and 1

Forget gate: controls what is kept vs forgotten, from previous cell state

$$\boldsymbol{f}^{(t)} = \sigma\left(\boldsymbol{W}_f \boldsymbol{h}^{(t-1)} + \boldsymbol{U}_f \boldsymbol{x}^{(t)} + \boldsymbol{b}_f\right)$$

Input gate: controls what parts of the new cell content are written to cell

$$\boldsymbol{i}^{(t)} = \sigma\left(\boldsymbol{W}_i \boldsymbol{h}^{(t-1)} + \boldsymbol{U}_i \boldsymbol{x}^{(t)} + \boldsymbol{b}_i\right)$$

Output gate: controls what parts of cell are output to hidden state

$$\boldsymbol{o}^{(t)} = \sigma\left(\boldsymbol{W}_o \boldsymbol{h}^{(t-1)} + \boldsymbol{U}_o \boldsymbol{x}^{(t)} + \boldsymbol{b}_o\right)$$

New cell content: this is the new content to be written to the cell

$$\tilde{\boldsymbol{c}}^{(t)} = \tanh\left(\boldsymbol{W}_c \boldsymbol{h}^{(t-1)} + \boldsymbol{U}_c \boldsymbol{x}^{(t)} + \boldsymbol{b}_c\right)$$

Cell state: erase ("forget") some content from last cell state, and write ("input") some new cell content

$$\boldsymbol{c}^{(t)} = \boldsymbol{f}^{(t)} \circ \boldsymbol{c}^{(t-1)} + \boldsymbol{i}^{(t)} \circ \tilde{\boldsymbol{c}}^{(t)}$$

Hidden state: read ("output") some content from the cell

$$\boldsymbol{h}^{(t)} = \boldsymbol{o}^{(t)} \circ \tanh \boldsymbol{c}^{(t)}$$

All these are vectors of same length n

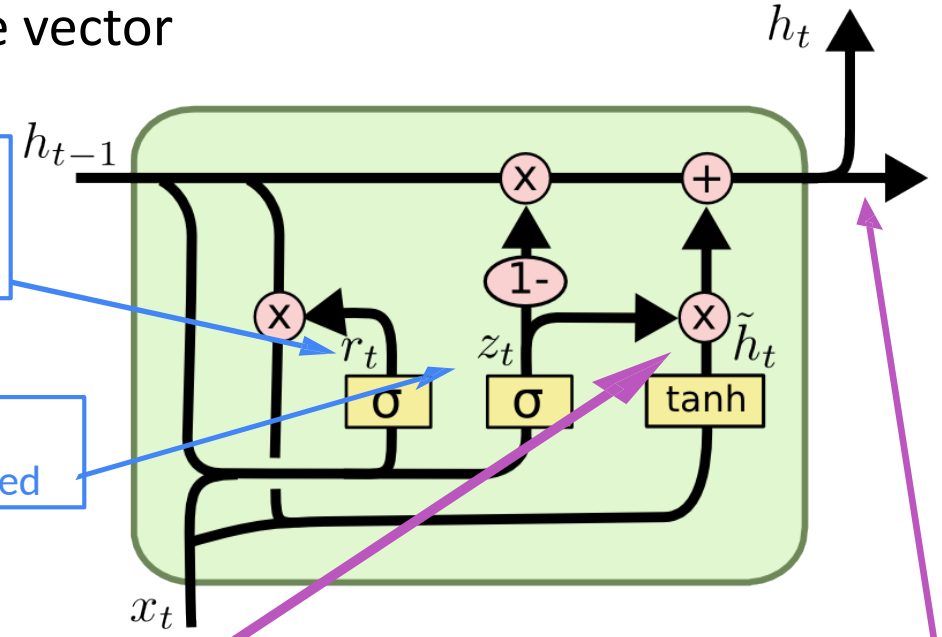
Gates are applied using element-wise product

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Gated Recurrent Unit (GRU)

- Alternative RNN to LSTM that uses fewer gates (Cho, et al., 2014)
  - Combines forget and input gates into "update" gate.
  - Eliminates cell state vector



Reset gate: controls what parts of previous hidden state are used to compute new content

Update gate: controls what parts of hidden state are updated vs preserved

New hidden state content: reset gate selects useful parts of prev hidden state. Use this and current input to compute new hidden content.

Hidden state: update gate simultaneously controls what is kept from previous hidden state, and what is updated to new hidden state content

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Review on your own: Gated Recurrent Units (GRU)

- Proposed by Cho et al. in 2014 as a simpler alternative to the LSTM.
- On each timestep *t* we have input $\boldsymbol{x}^{(t)}$ and hidden state $\boldsymbol{h}^{(t)}$ (no cell state).

Update gate: controls what parts of hidden state are updated vs preserved

$$\boldsymbol{u}^{(t)} = \sigma\left(\boldsymbol{W}_u\boldsymbol{h}^{(t-1)} + \boldsymbol{U}_u\boldsymbol{x}^{(t)} + \boldsymbol{b}_u\right)$$

Reset gate: controls what parts of previous hidden state are used to compute new content

$$\boldsymbol{r}^{(t)} = \sigma\left(\boldsymbol{W}_r\boldsymbol{h}^{(t-1)} + \boldsymbol{U}_r\boldsymbol{x}^{(t)} + \boldsymbol{b}_r\right)$$

New hidden state content: reset gate selects useful parts of prev hidden state. Use this and current input to compute new hidden content.

$$\tilde{\boldsymbol{h}}^{(t)} = \tanh\left(\boldsymbol{W}_h(\boldsymbol{r}^{(t)} \circ \boldsymbol{h}^{(t-1)}) + \boldsymbol{U}_h\boldsymbol{x}^{(t)} + \boldsymbol{b}_h\right)$$

Hidden state: update gate simultaneously controls what is kept from previous hidden state, and what is updated to new hidden state content

$$\boldsymbol{h}^{(t)} = (1 - \boldsymbol{u}^{(t)}) \circ \boldsymbol{h}^{(t-1)} + \boldsymbol{u}^{(t)} \circ \tilde{\boldsymbol{h}}^{(t)}$$

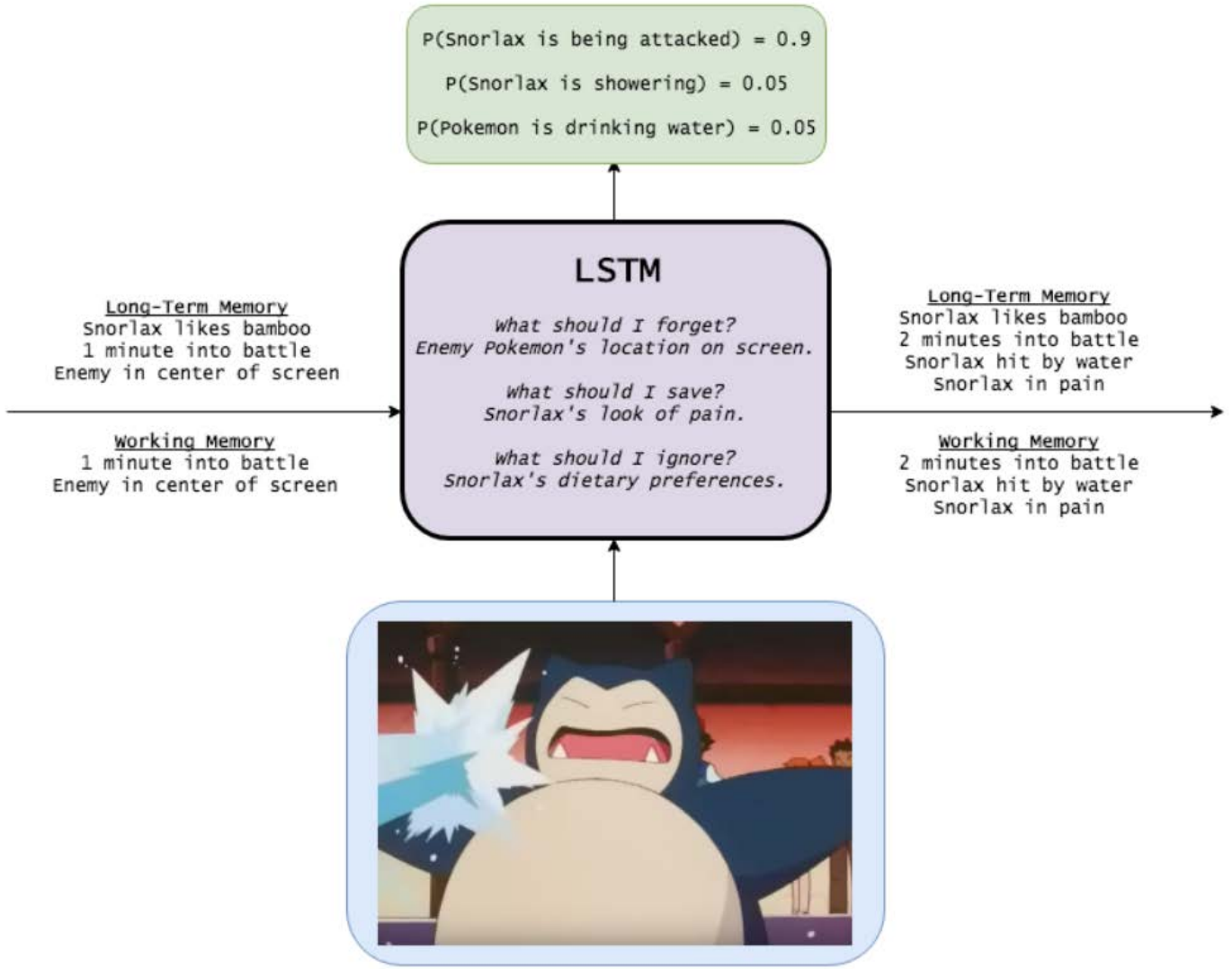
How does this solve vanishing gradient?
GRU makes it easier to retain info long-term (e.g. by setting update gate to 0)

"Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation", Cho et al. 2014, https://arxiv.org/pdf/1406.1078v3.pdf

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Cell State vs Hidden State



http://blog.echen.me/2017/05/30/exploring-lstms/

UNIVERSITÉ CÔTE D'AZUR

univ-cotedazur.fr

Activity

http://blog.echen.me/2017/05/30/exploring-lstms/

LSTM vs GRU

- Researchers have proposed many gated RNN variants, but LSTM and GRU are the most widely-used
- The biggest difference is that GRU is quicker to compute and has fewer parameters
- There is no conclusive evidence that one consistently performs better than the other
- LSTM is a good default choice (especially if your data has particularly long dependencies, or you have lots of training data)
- Rule of thumb: start with LSTM, but switch to GRU if you want something more efficient

LSTMs: real-world success

- In 2013-2015, LSTMs started achieving state-of-the-art results for sequence modeling
  - Successful tasks include: handwriting recognition, speech recognition, machine translation, parsing, image captioning
  - LSTM became the dominant approach

- Starting in 2019, other approaches (e.g. Transformers) became more dominant for certain NLP tasks (will discuss next lecture)
  - For example in **WMT** (machine translation competition):
  - In WMT 2016, the summary report contains ”RNN” 44 times
  - In WMT 2018, the report contains “RNN” 9 times and “Transformer” 63 times

Source: "Findings of the 2016 Conference on Machine Translation (WMT16)", Bojar et al. 2016, http://www.statmt.org/wmt16/pdf/W16-2301.pdf
Source: "Findings of the 2018 Conference on Machine Translation (WMT18)", Bojar et al. 2018, http://www.statmt.org/wmt18/pdf/WMT028.pdf

Adapted from Abigail See