Réseaux de neurones

LIONEL PREVOST

HEAD OF LEARNING, DATA & ROBOTICS LAB – ESIEA

lionel.prevost@esiea.fr

Organisation du module

Face à face : 24h

- Dont cours : 12h
- Dont TD sur machine (python/sklearn) : 12h
- Dont tests sur machine (notés)

Exam : 1h30 QC + Exos (sans doc)

Annonces

Présence

Plan détaillé du cours

Enseignants de Cours et de TDs

Cours	TP	Enseignant	Adresse Email
TC1		Lionel Prevost	lionel.prevost@esiea.fr
	41	Lionel Prevost	lionel.prevost@esiea.fr
	42	Lionel Prevost	lionel.prevost@esiea.fr
TC2		Lionel Prevost	lionel.prevost@esiea.frr
	43	Siba HAIDAR	siba.haidar@esiea.fr
	44	Siba HAIDAR	siba.haidar@esiea.fr
TC3		Antoun YAACOUB	antoun.yaacoub@esiea.fr
	45	Antoun YAACOUB	antoun.yaacoub@esiea.fr
	46	Antoun YAACOUB	antoun.yaacoub@esiea.fr

Plan du cours

Plan détaillé disponible sur Moodle

https://learning.esiea.fr/mod/page/view.php?id=52308

INTRODUCTION

DECIDER SANS APPRENDRE

INTELLIGENCE ARTIFICIELLE

NEURONE ARTIFICIEL ET APPRENTISSAGE

RESEAUX DE NEURONES MONOCOUCHES

RESEAUX DE NEURONES MULTI COUCHES

BONNES PRATIQUES

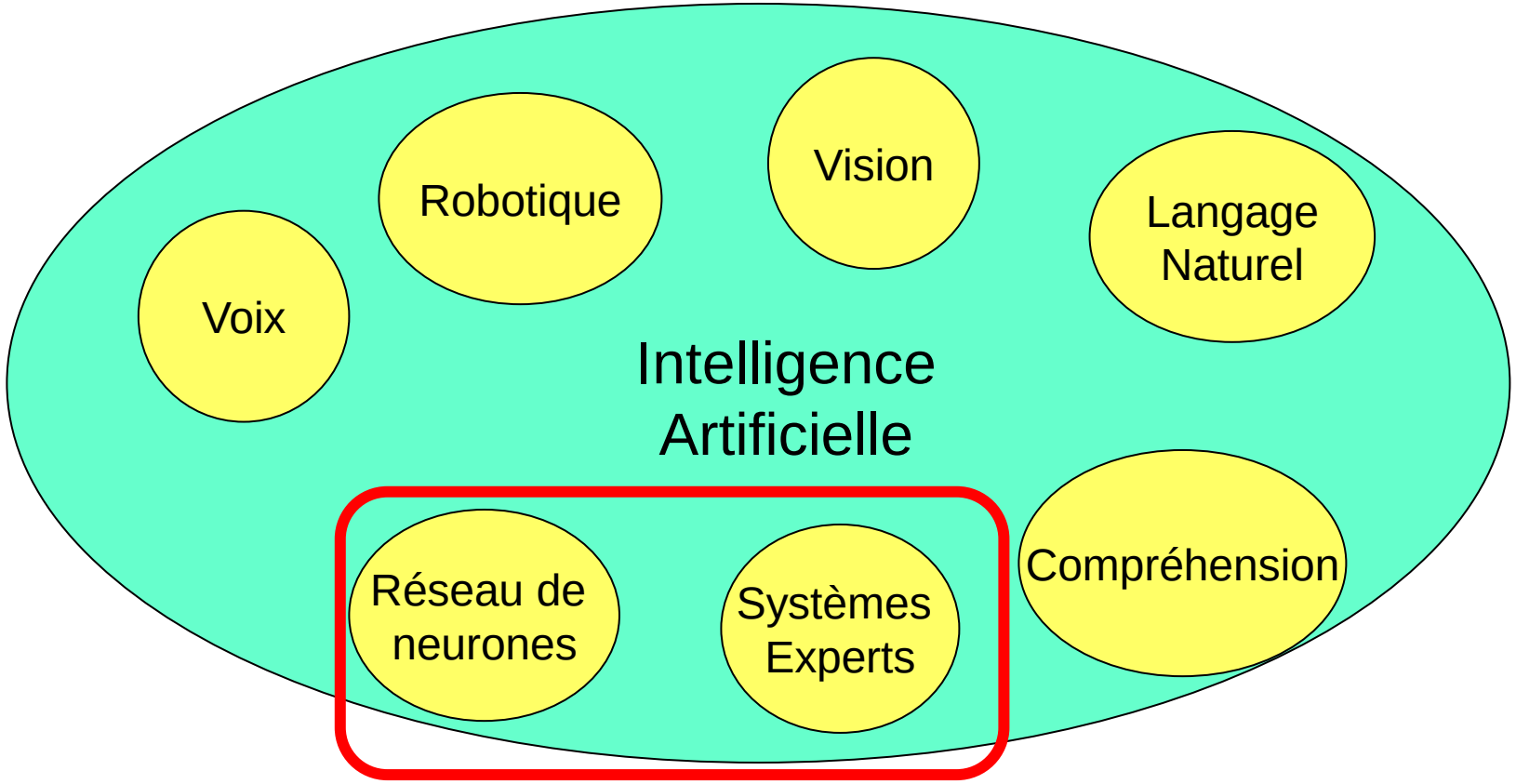
MESURE DE PERFORMANCES

TECHNIQUES CLASSIQUES DE TRAITEMENTS D'IMAGES

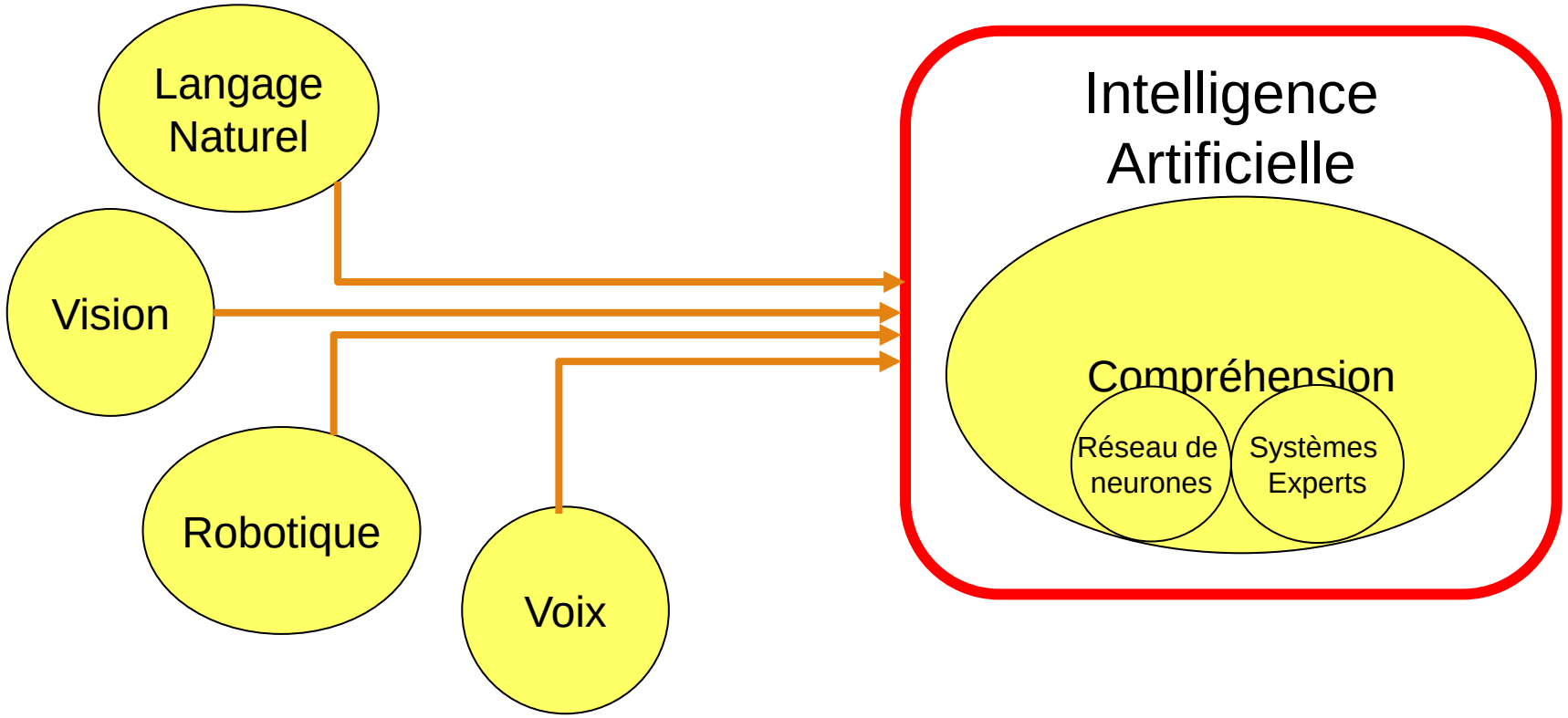
RESEAUX A CONVOLUTION

Introduction

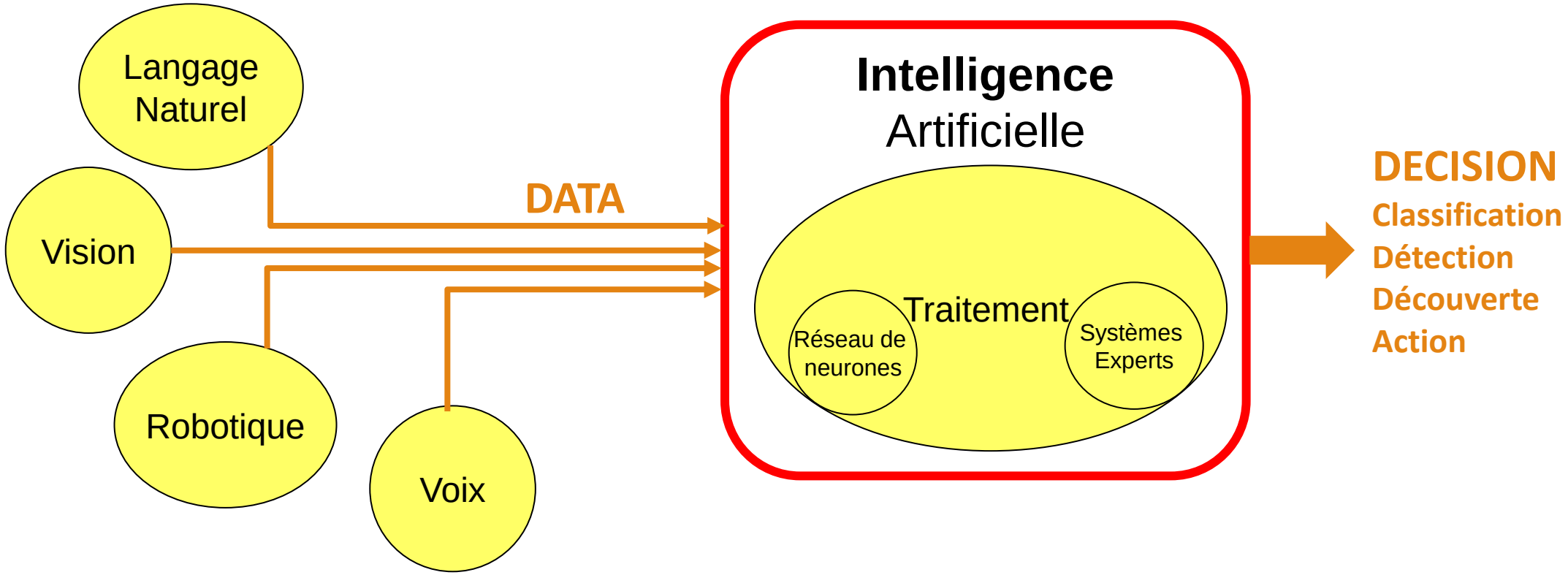
L'intelligence artificielle



(...)


Langage Naturel
Vision
Robotique
Voix
Intelligence Artificielle
Compréhension
Réseau de neurones
Systèmes Experts

(...)


Langage Naturel
Vision
Robotique
Voix
DATA
Intelligence Artificielle
Traitement
Réseau de neurones
Systèmes Experts
DECISION
Classification
Détection
Découverte
Action

Qu’est ce qu’être intelligent ?

Savoir Apprendre

- élaborer un système de connaissances **à partir des données**
- pouvoir **intégrer de nouvelles données** (connaissances)
- supervisé (par un professeur) ou non supervisé (par association)

Savoir Raisonner (déduire, anticiper)

- à partir du système de connaissances **et des données** de l’expérience pouvoir produire
  - Une décision
  - de nouvelles connaissances

Qu’est ce qu’être intelligent ?

Posséder une histoire (mémoire court/long terme)

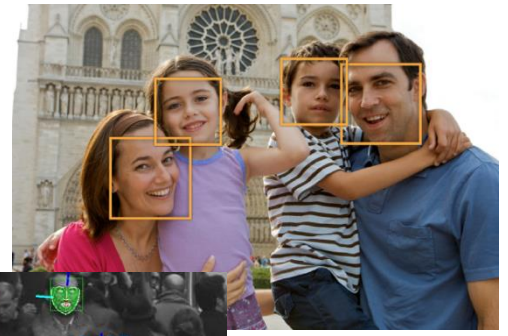
Posséder des sentiments

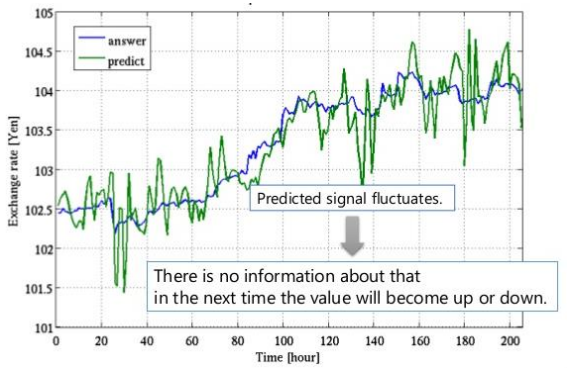
(= ressentir (percevoir)/exprimer une émotion ?)

Posséder une conscience

Données ?

- Signal (parole …)
- images (RGB, IR, Z …)
- Vidéo (RGB + t)
- Séries temporelles





Voir les entrepôts de données :

- [UCI machine learning repository]
- [Kaggle]

Exemples

Breast cancer dataset

Data Set Characteristics:	Multivariate	**Number of Instances:**	699	**Area:**	Life
Attribute Characteristics:	Integer	**Number of Attributes:**	10	**Date Donated**	1992-07-15

Complete attribute documentation:
1. Sample code number: id number
2. Clump Thickness: 1 - 10
3. Uniformity of Cell Size: 1 - 10
4. Uniformity of Cell Shape: 1 - 10
5. Marginal Adhesion: 1 - 10
6. Single Epithelial Cell Size: 1 - 10
7. Bare Nuclei: 1 - 10
8. Bland Chromatin: 1 - 10
9. Normal Nucleoli: 1 - 10
10. Mitoses: 1 - 10
11. Class: (2 for benign, 4 for malignant)

(…)

Arrythmia dataset

Abstract: Distinguish between the **presence and absence** of cardiac arrhythmia and classify it in one of the **16** groups.

Data Set Characteristics:	Multivariate	**Number of Instances:**	452	**Area:**	Life
Attribute Characteristics:	Categorical, Integer, Real	**Number of Attributes:**	279	**Date Donated**	1998-01-01

Complete attribute documentation:
1 Age: Age in years , linear
2 Sex: Sex (0 = male; 1 = female) , nominal
3 Height: Height in centimeters , linear
4 Weight: Weight in kilograms , linear
…

15 Heart rate: Number of heart beats per minute ,linear

…

(...)

Auto MPG (miles per gallon) dataset

Data Set Characteristics:	Multivariate	**Number of Instances:**	398	**Area:**	N/A
Attribute Characteristics:	Categorical, Real	**Number of Attributes:**	8	**Date Donated**	1993-07-07

Complete attribute documentation:
1. mpg: continuous
2. cylinders: multi-valued discrete
3. displacement: continuous
4. horsepower: continuous
5. weight: continuous
6. acceleration: continuous
7. model year: multi-valued discrete
8. origin: multi-valued discrete
9. car name: string (unique for each instance)

Formalisation

Exemple = Observation + Signification → apprentissage **supervisé**

Observation : vecteur aléatoire $X_j=(x_1, x_2,\dots x_D)$

- x_i : caractéristiques, descripteurs, *features,* variables explicatives
  - *Qualitatives (nominales / ordinales)*
  - *Quantitatives (continues / discrètes)*
- D : dimension de l'espace de représentation

(…)

Signification : y_{dj}

- $y_{di} \in \mathbb{R}$: variable quantitative → **régression** variable à expliquer/prédire
- $y_{di} \in \mathbb{N}$: variable qualitative → **classification** vérité terrain, label, classe
- K : dimension de l'espace de décision (== nombre de classes)

→ **Fonction de classification** : soit X l'espace des entrées (représentation) et Y, celui des sorties (décision)

$h : X \rightarrow Y$ telle que $h(X_j) \approx y_{dj}$

Base de données

Matrice de dimension (DxN) (#dimension x #exemples)

X_1	X_2	X_3	…				X_N
x_{11}	x_{21}	x_{31}	…				x_{N1}
x_{12}	x_{22}	x_{32}	…				x_{N2}
…							
x_{1D}	x_{2D}	x_{3D}					x_{ND}

Signification :

y_1	y_2	y_3					y_N

Exemple : analyse de trafic routier

Discrimination camion/autres véhicules

Véhicules : 2 descripteurs
- x_1 : longueur (m)
- x_2 : bruit (dB)

Véhicule	x_1	x_2	y_d
Camion1	20	8	1
Camion2	15	20	1
Car	16	10	-1
Voiture1	5	15	-1
Voiture2 (+remorque)	16	6	-1
Moto	2	20	-1

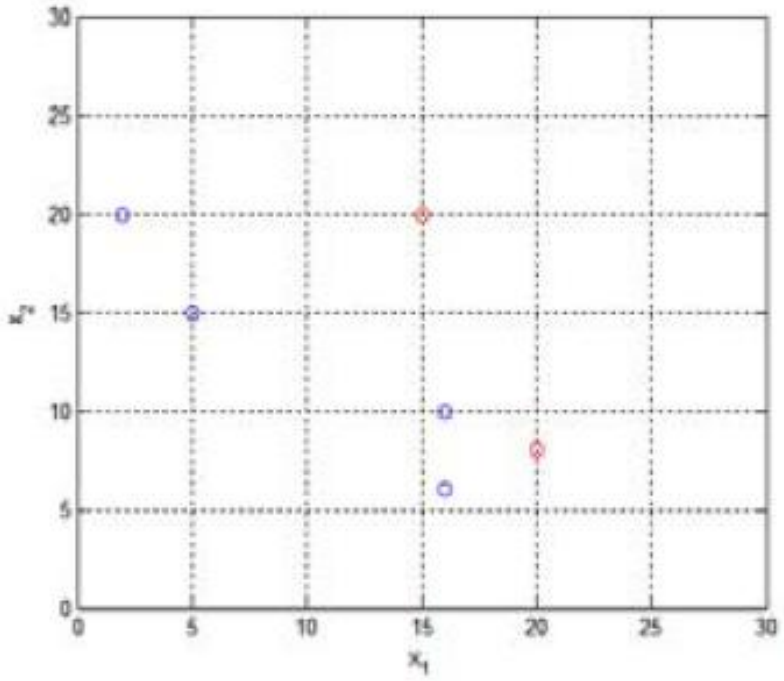
1 exemple

(…)

Discrimination camion/autres véhicules

Véhicules : 2 descripteurs
- x_1 : longueur (m)
- x_2 : bruit (dB)

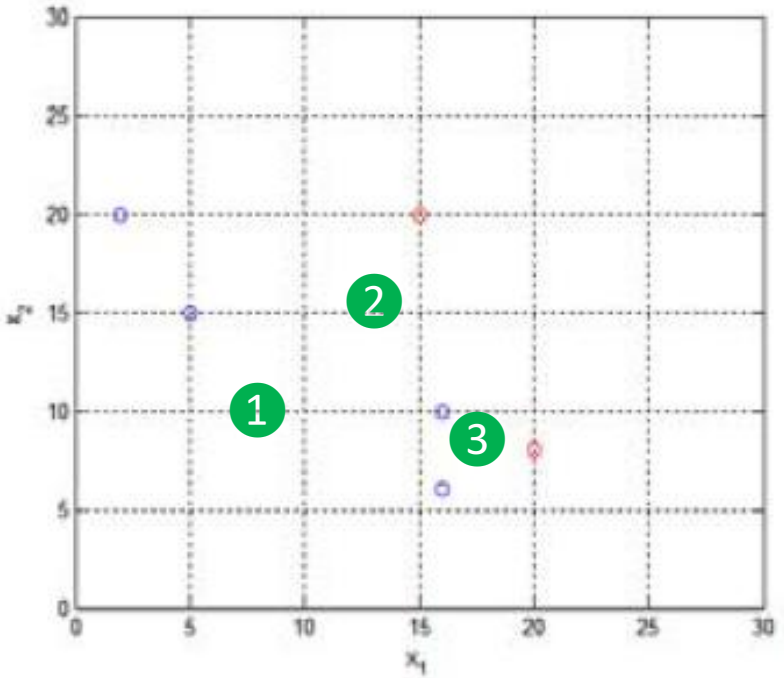
Véhicule	x_1	x_2	y_d
Camion1	20	8	1
Camion2	15	20	1
Car	16	10	-1
Voiture1	5	15	-1
Voiture2 (+remorque)	16	6	-1
Moto	2	20	-1



Fun time

Q : « **intuitivement** », comment classer un nouveau véhicule observé ?

	x1	x2
1	8	10
2	13	15
3	17	9



Décider SANS apprendre : plus proche voisin

- Données de départ :

  Un ensemble d'exemples de référence E de vecteurs étiquetés X_i avec y_{di} la classe associée

- Objectif :

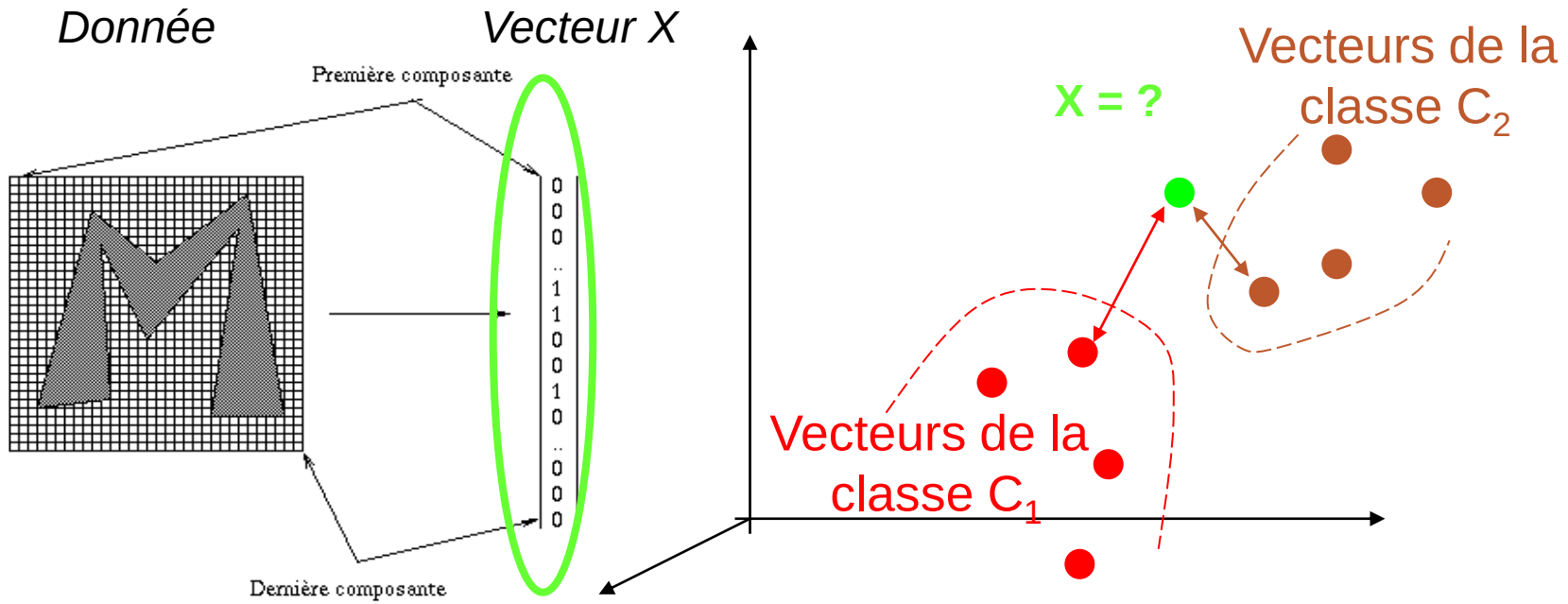
  Décider de la classe d'un vecteur inconnu X

- Méthode :

  Calculer les distances entre X et l'ensemble des vecteurs X_i de E

  Attribuer à X la classe de son plus proche voisin (distance minimum)

Calcul de distance (variables quantitatives)



→ distance euclidienne (norme L_2) : $d(X_1 , X_2) = \sqrt{\sum_{i=1}^{i=D}(x_{1i} - x_{2i})^2}$

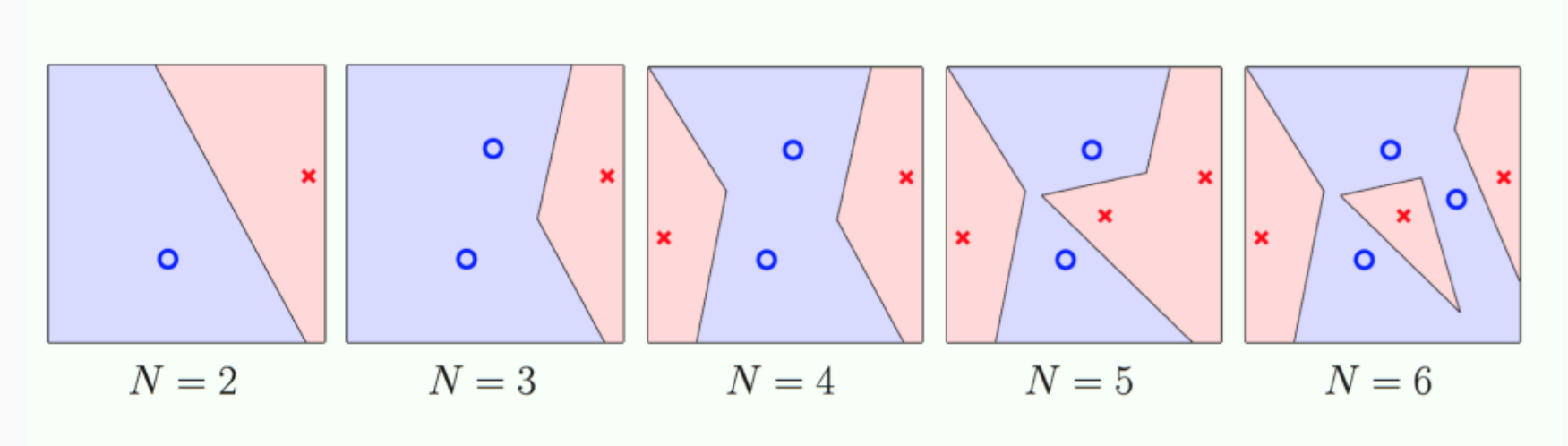
Fun time

Q déterminer la frontière de décision définie par deux exemples de classe différente X_1 et X_2

- En 1D : $X_1 = x_1$, $X_2 = x_2$

- en 2D : $X_1 = (x_1 , y_1)$ et $X_2 = (x_2 , y_2)$

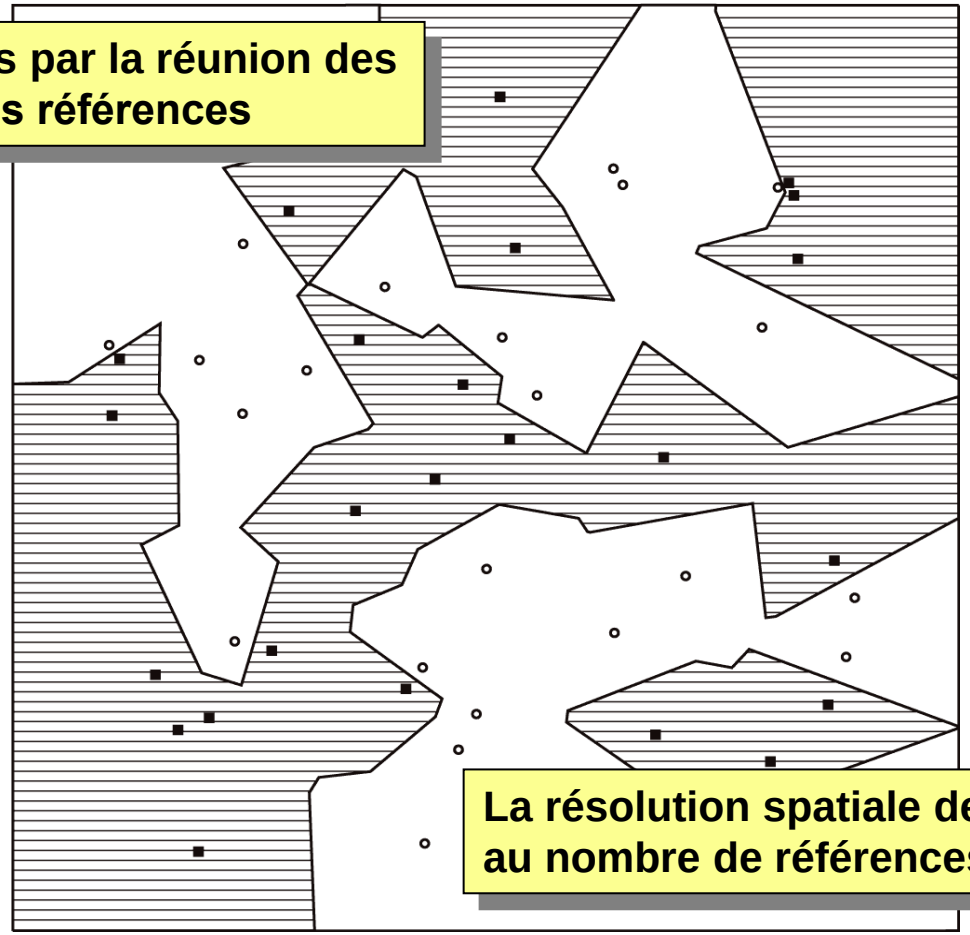
Interprétation géométrique



→ Frontières « linéaires par morceaux »

(…)

Les classes sont définies par la réunion des domaines d'influence des références



La résolution spatiale des frontières est liée au nombre de références et à leur densité

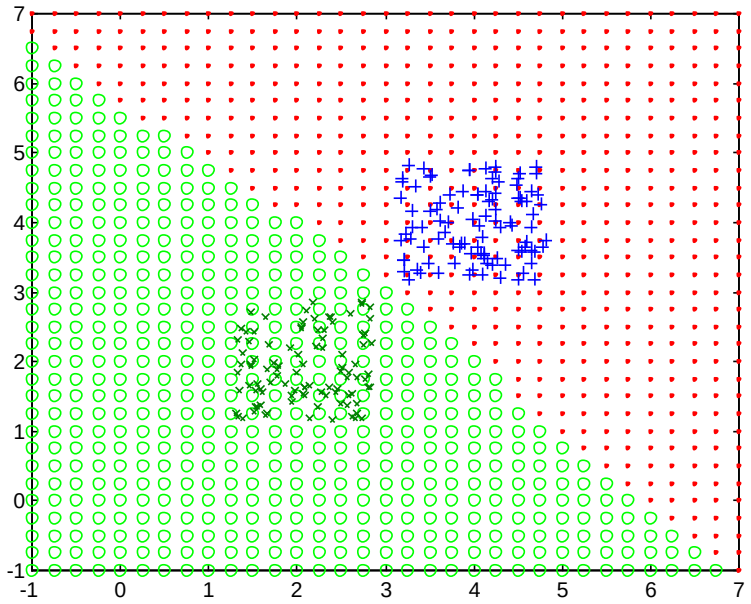
Extension : k-plus-proches-voisins

1. Calculer la distance entre X et tous les exemples de la base de référence

2. Déterminer les k vecteurs PPV(X) de la base les plus proches

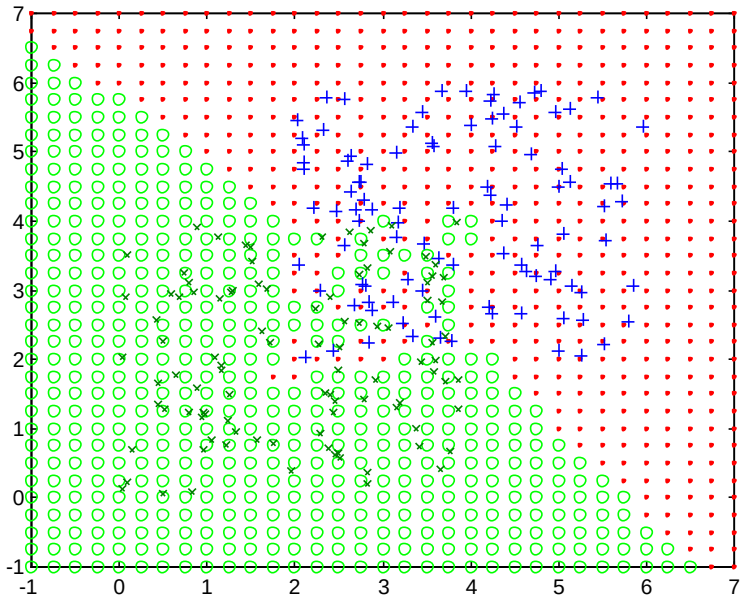
- **k = 1** : C_i = argmin d(X_i, X)
- **k ≠ 1** : C_i = classe majoritaire de {PPV(X)}
- **Variante** : unanimité sinon rejet, prise en compte de la distance

Exemple : 1-ppv



100 exemples par classe

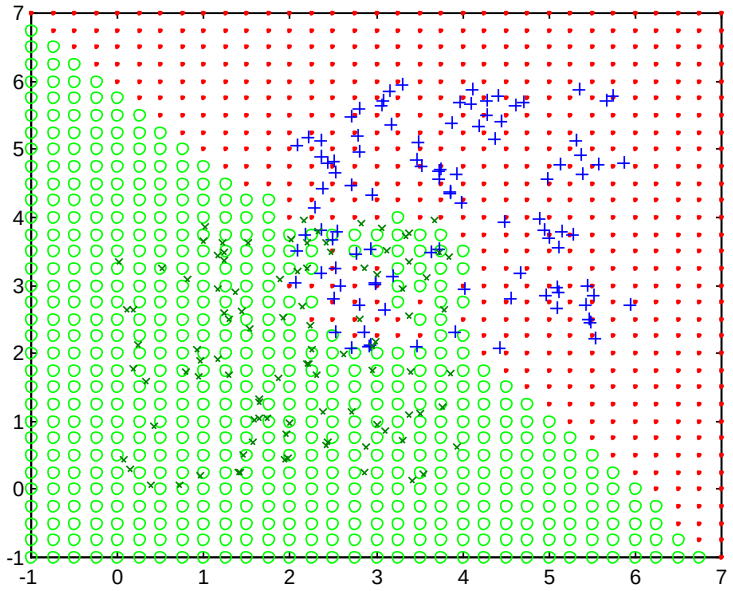
K = 1, disp = 0.3



100 exemples par classe

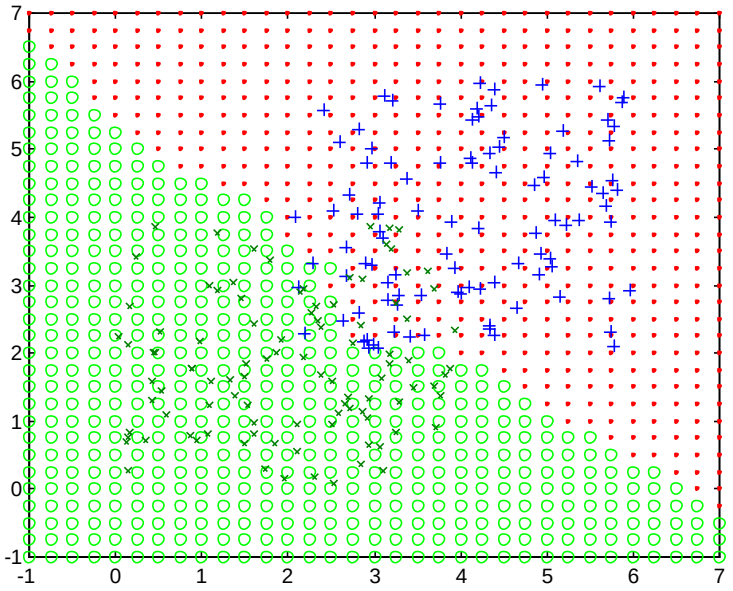
K = 1, disp = 0.7

Exemple : k-ppv



K = 11

disp = 0.7

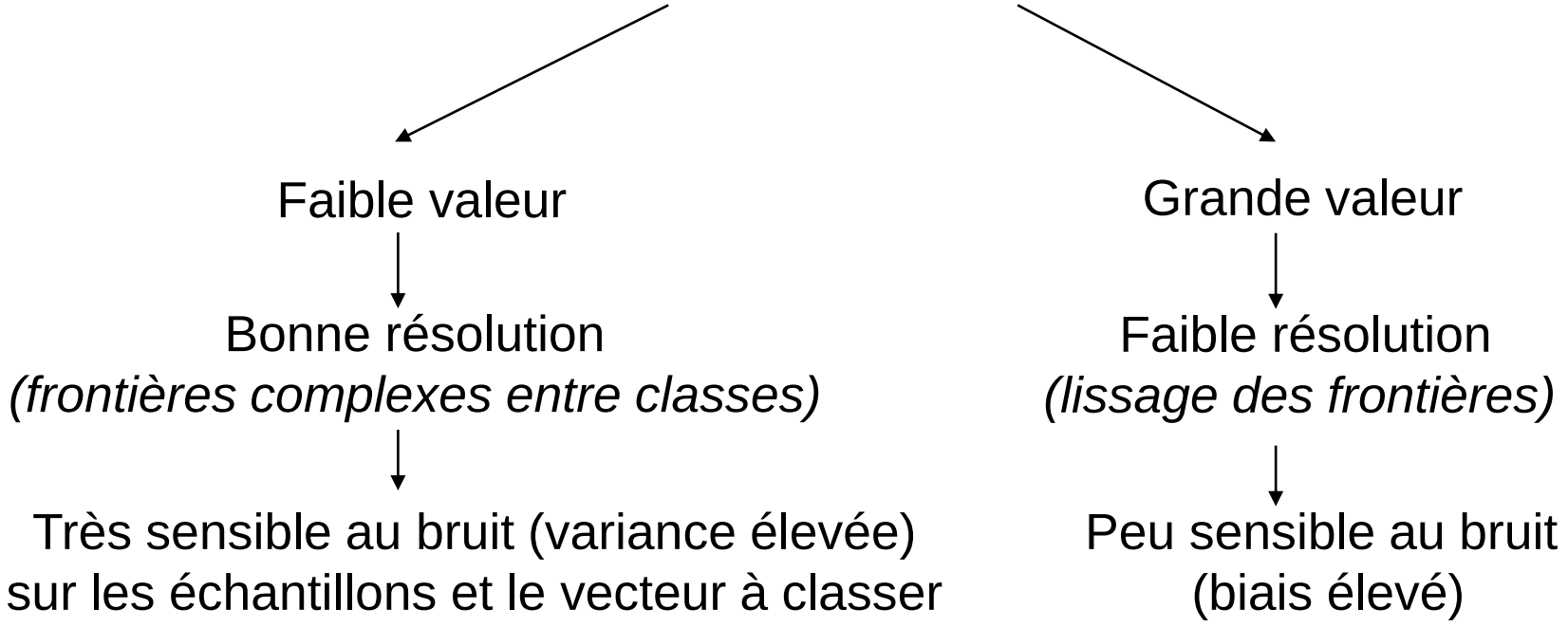


K = 21

disp = 0.7

Dilemme biais-variance

Le choix du paramètre k a une influence directe sur les propriétés du classifieur



L'algorithme parfait ?

Avantages :

- Très simple à mettre en œuvre (*lazy learning*)
- Naturellement multiclasse
- Incrémental
- Tend asymptotiquement (N→∞) vers l'erreur optimale ($\varepsilon_B < \varepsilon_{ppv} < 2\varepsilon_B$)
- S'adapte facilement à la régression : $y = \frac{1}{k}\sum_{i=1}^{i=k} y_{di}$ sur les PPV(X)

(…)

Inconvénients :

- Stockage des références
- Quantité de calculs proportionnelle au nombre de référence
- Pas d'extraction d'information utile
- Réglage du paramètre k

Bilan

Algorithme « baseline » pour évaluer la complexité d'un problème, la qualité des données …

Accélérateurs : diminuer D ou N (module datamining)

→ réduction de dimension

→ catégorisation (clustering)

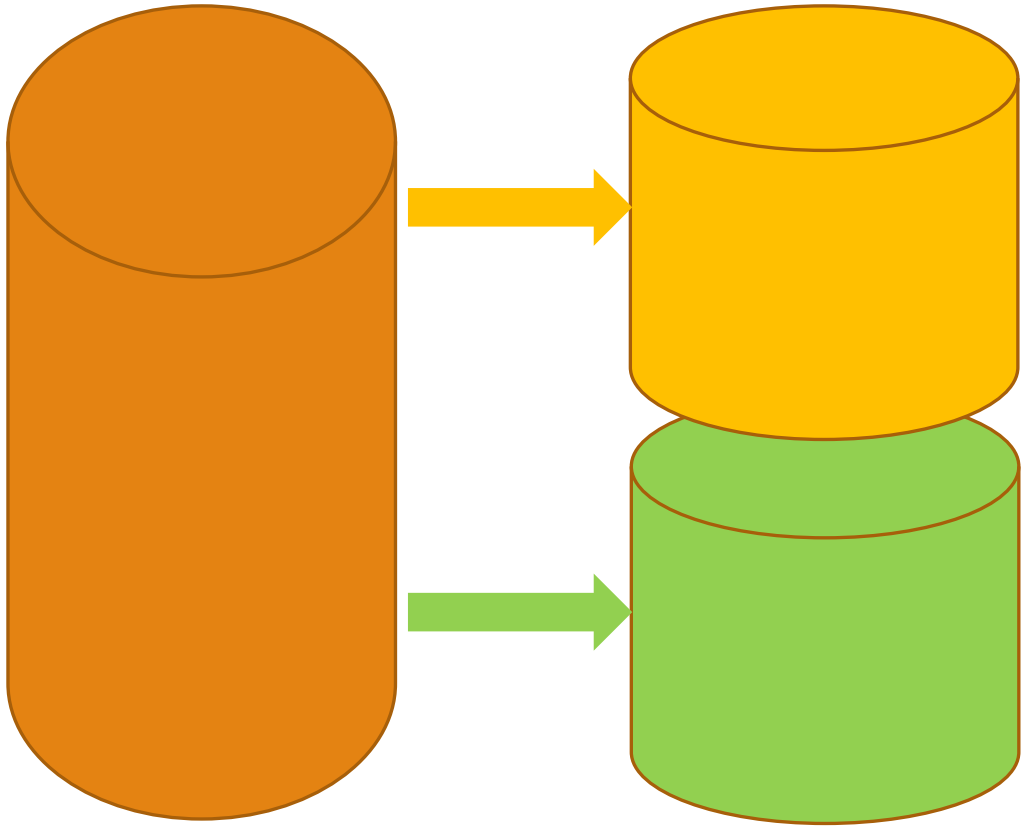
TD (correction en TP)

- Analyser les BDD des transparents 12-13-14 pour déterminer leurs caractéristiques (dimension, nombre d'exemples …)
- Coder les algorithmes 1-ppv et k-ppv (∀ langage*)
- Savoir déterminer graphiquement des frontières en 2D

(*) on dispose des fonctions :

- d=distance(ex1, ex2) : calcul de la distance entre deux exemples
- sort(d) : tri des distances par ordre croissant

Mesures de performance

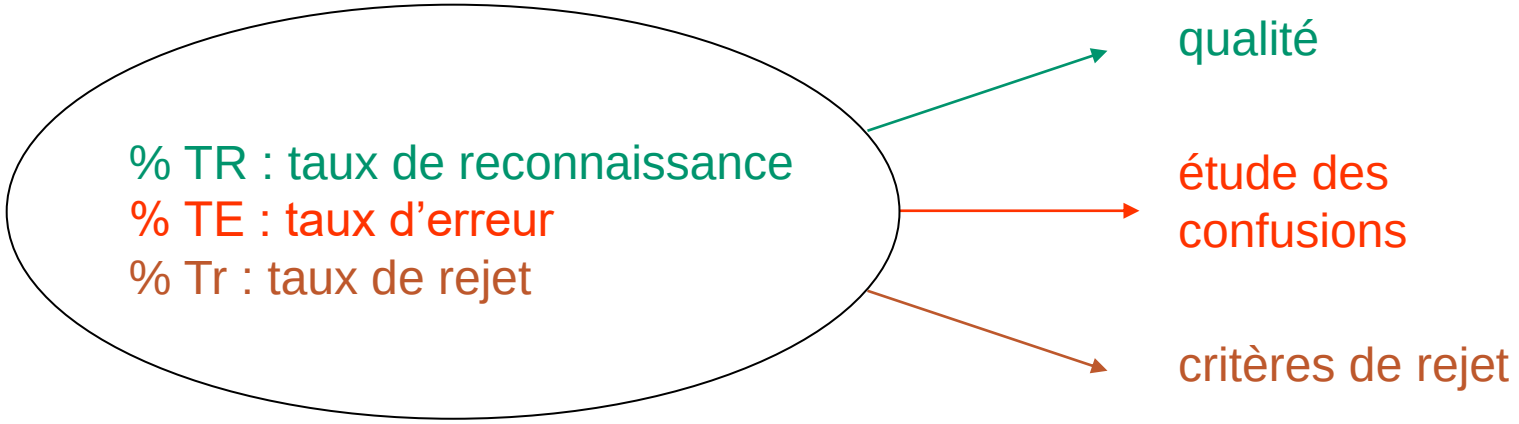


Base d'apprentissage/de référence

Base de test/de généralisation = capacité à classer des données nouvelles, inconnues

→ **Performances opérationnelles**

Cas multiclasse



$$TR = \frac{\#(X \backslash y = y_d)}{N}$$

TE = 1 - TR

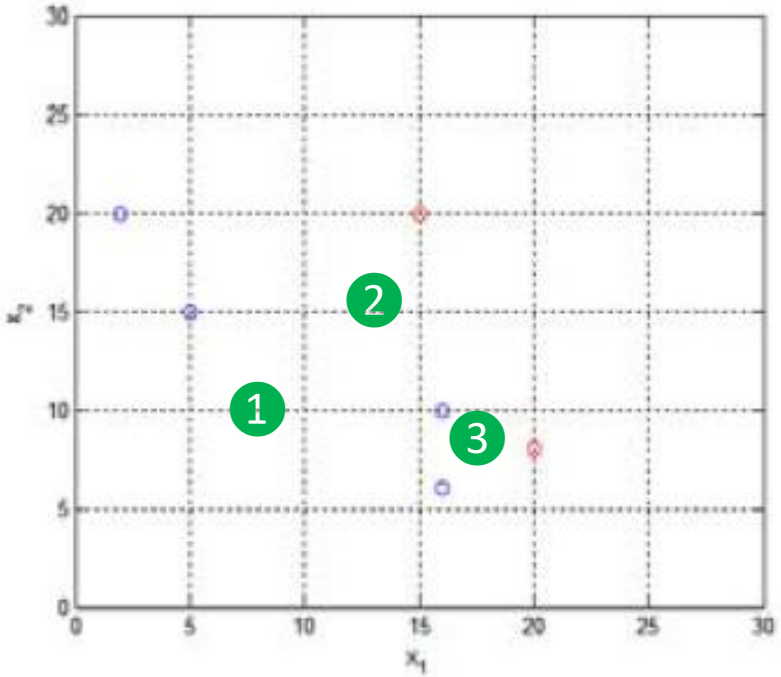
Matrice de confusion :

y / y_d	1	2	3
1	# (X\y=y_d=1)	# (X\y=2&y_d=1)	# (X\y=3&y_d=1)
2		# (X\y=y_d=2)	
3	# (X\y=1&y_d=3)		# (X\y=y_d=3)

Fun time

Q : utiliser l'algorithme du ppv pour classer un nouveau véhicule observé ?

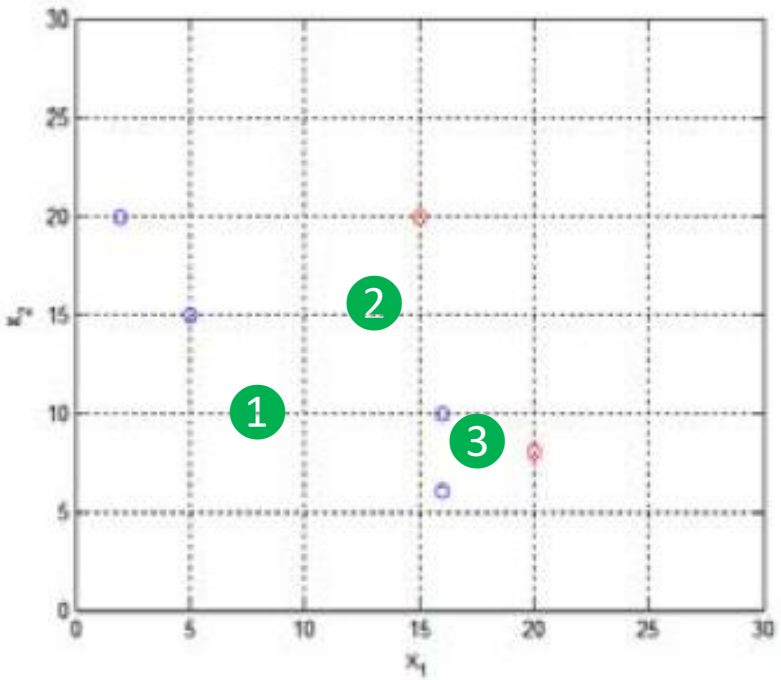
	x1	x2
1	8	10
2	13	15
3	17	9



Fun time

Q déterminer le taux de reconnaissance et la matrice de confusion du 1-ppv et du 3-ppv ?

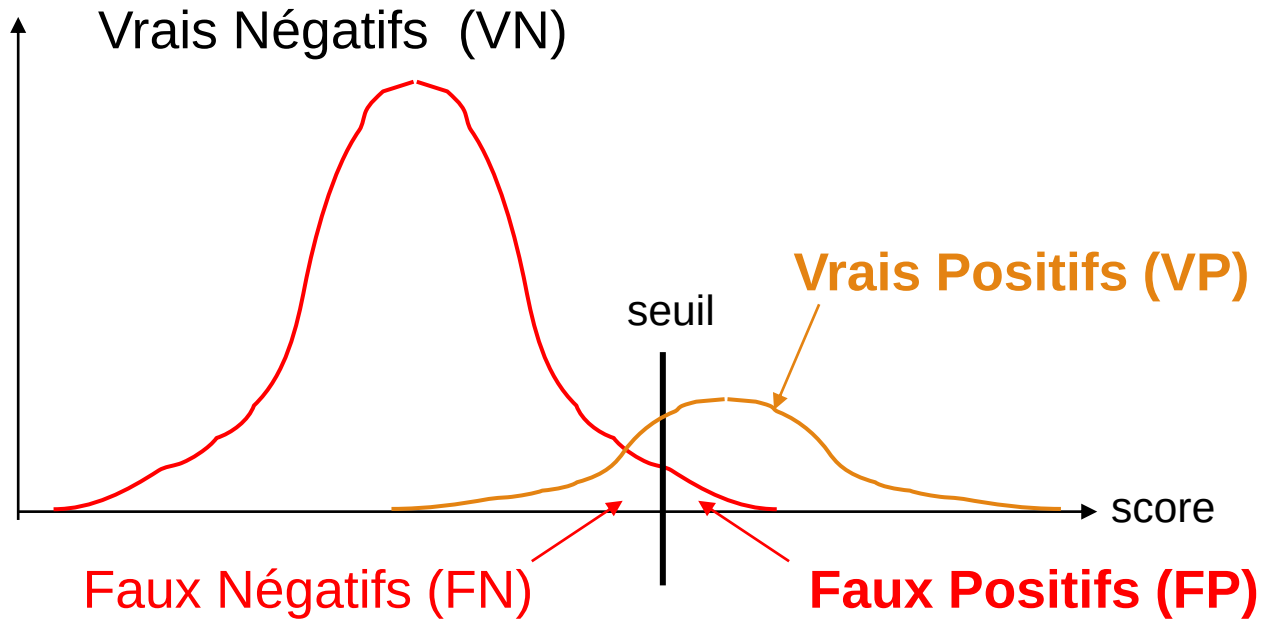
	x1	x2	**yd**
1	8	10	**-1**
2	13	15	**-1**
3	17	9	**-1**



Cas binaire

Problèmes de détection (visuelle, diagnostic, biométrie …)

	+	-
+	VP	FN
-	FP	VN



(…)

Taux de vrais positifs

$$T1 = \frac{\text{positifs bien classés}}{\text{total positifs}}$$

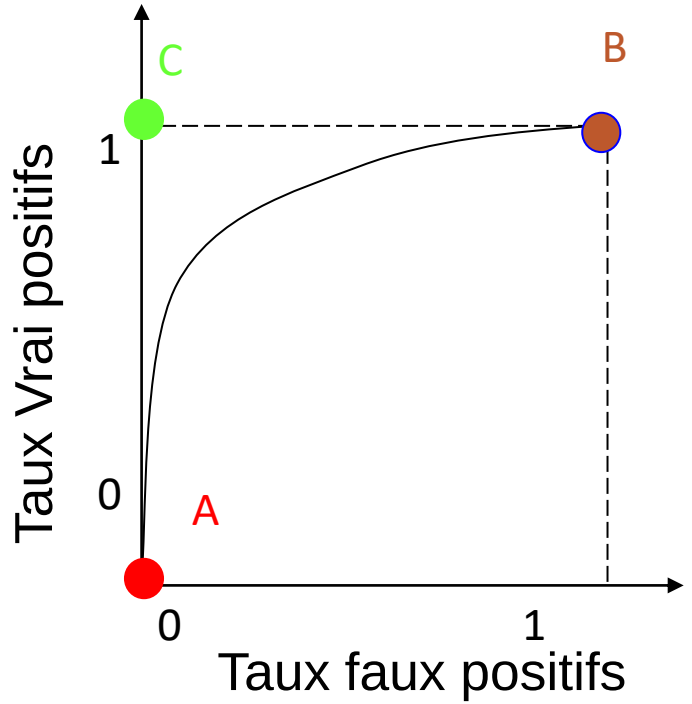
Taux de faux positifs

$$T2 = \frac{\text{Négatifs mal classés}}{\text{total négatifs}}$$

→ **Courbe ROC**
(Receiving Operator Caracteristic)

→ **AUC : comparaison de classifieurs**
(area under the curve)

→ **Voir aussi : courbe précision/rappel**



A : prédit toujours négatif
B : prédit toujours positif
C : Point idéal

STIA (Système Tutoriel Intelligent et Affectif)

Projet de Recherche au sein du Labo LDR

COLLECTE DE DONNÉES

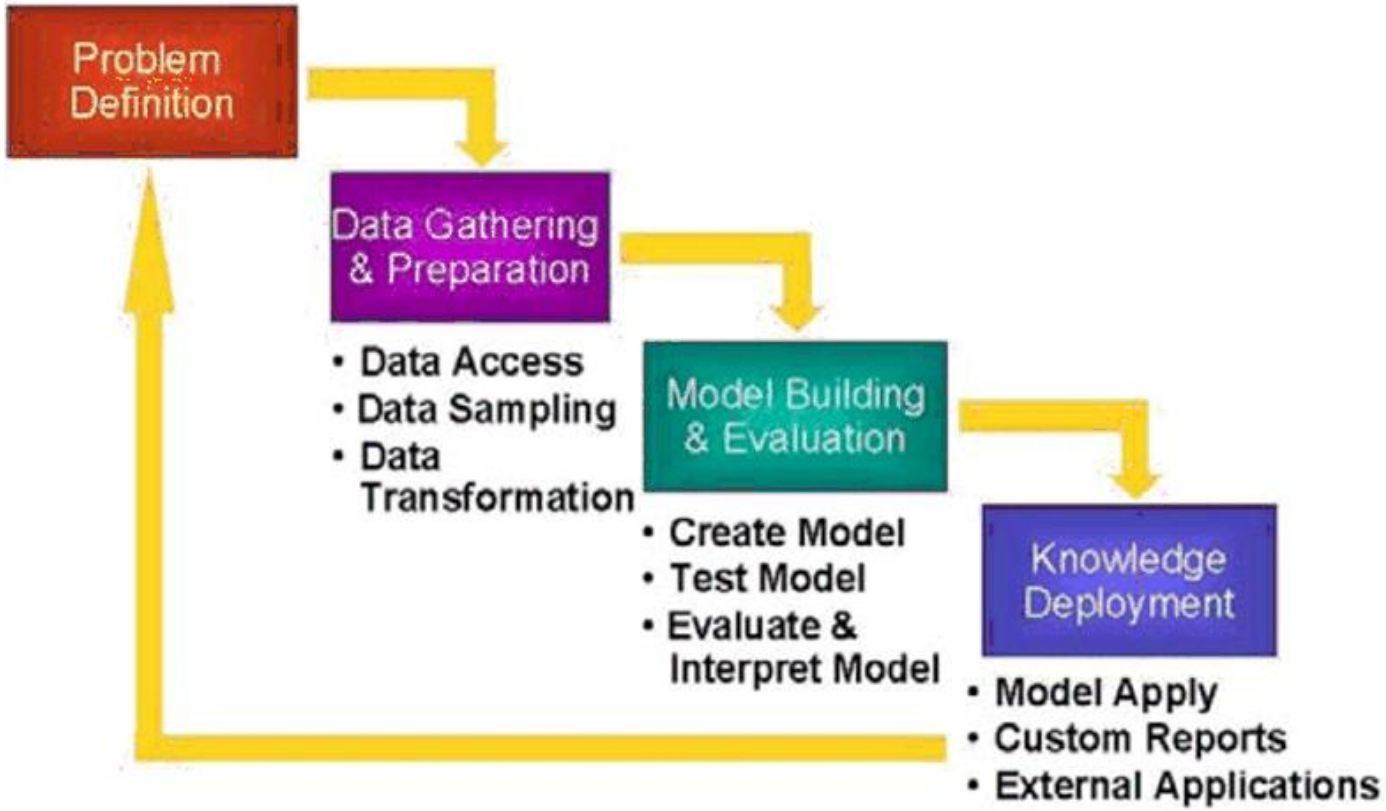
PRÉSENTATION DE NOTRE APPRENTI : EDOUARD NADAUD

Réseaux de neurones

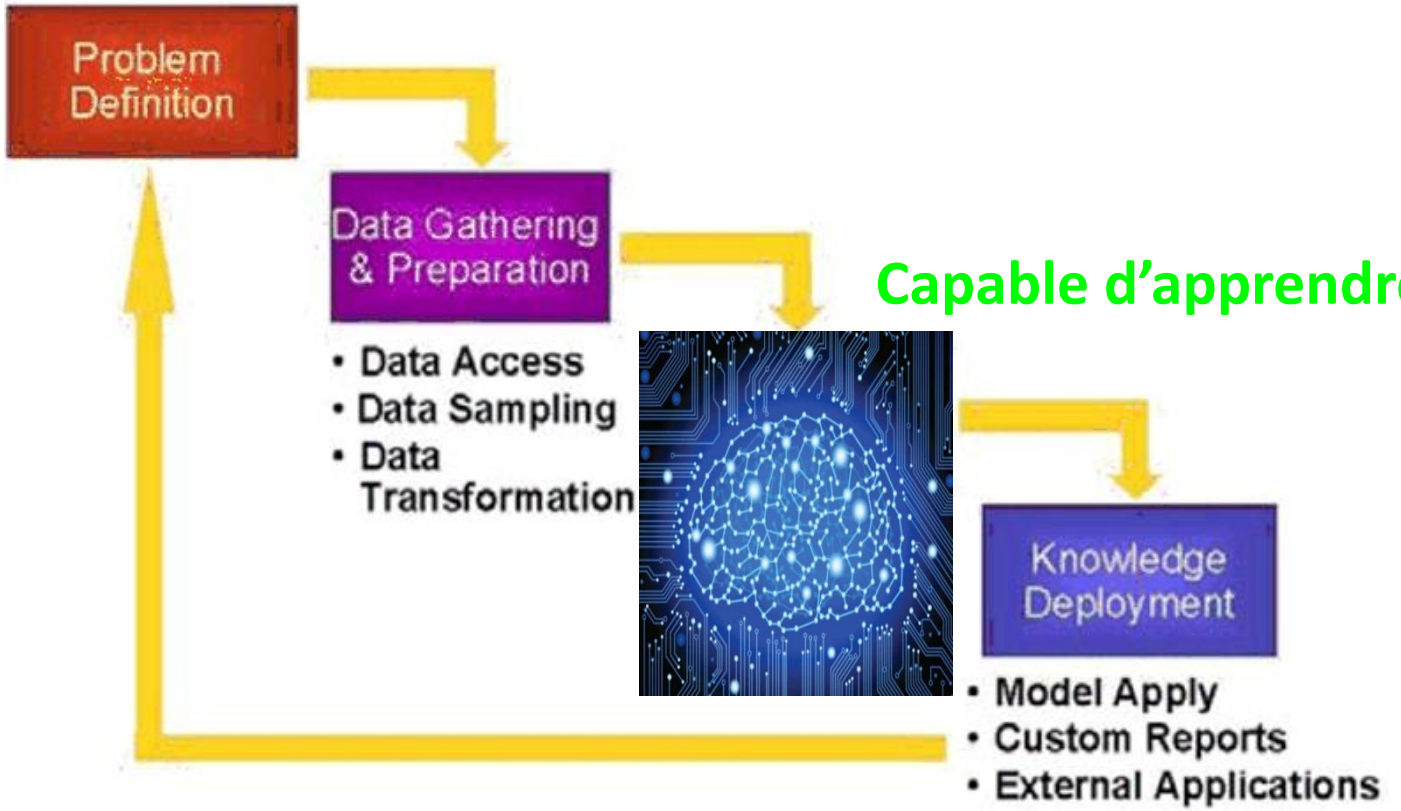
Pourquoi apprendre ? les 4V du Big Data

- Volume
  - twitter : 12 To/jour
- Vélocité
  - 5 million de transactions commerciales par jour à authentifier
- Variabilité
  - données structurés ou non, sorties capteurs, audio, vidéo, logs …
- Véracité

Data science



(…)



Capable d'apprendre, raisonner …

Les réseaux de neurones : pourquoi ?

Caractéristiques du cerveau humain	réseaux artificiels
une capacité d'apprentissage	☺
une capacité de généralisation	☺ (précautions à prendre)
une capacité d'adaptation	☹ (si beaucoup d'exemples)
une faible consommation énergétique	☹
une architecture massivement parallèle	☺

1888 : Neurone naturel

Fondements de neuro-physiologie :

Cerveau humain :

Nombre de neurones : 10^{10} à 10^{11}

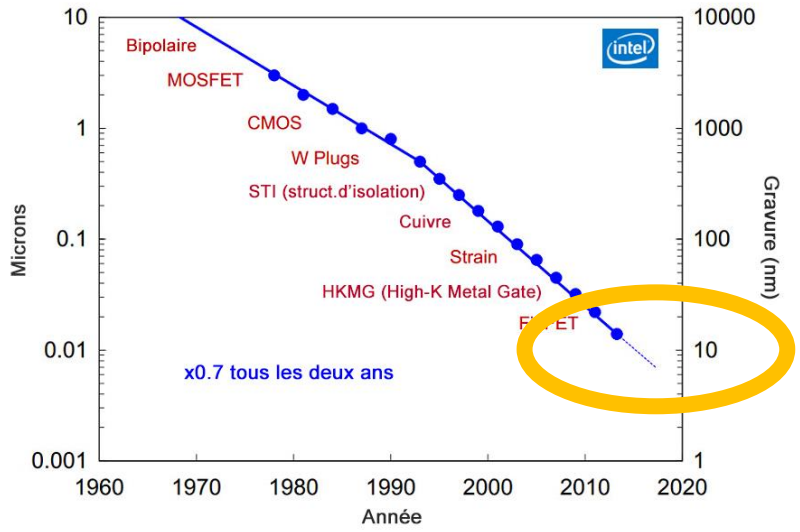
Nombre de connexions par neurone 10^{3} à 10^{5}

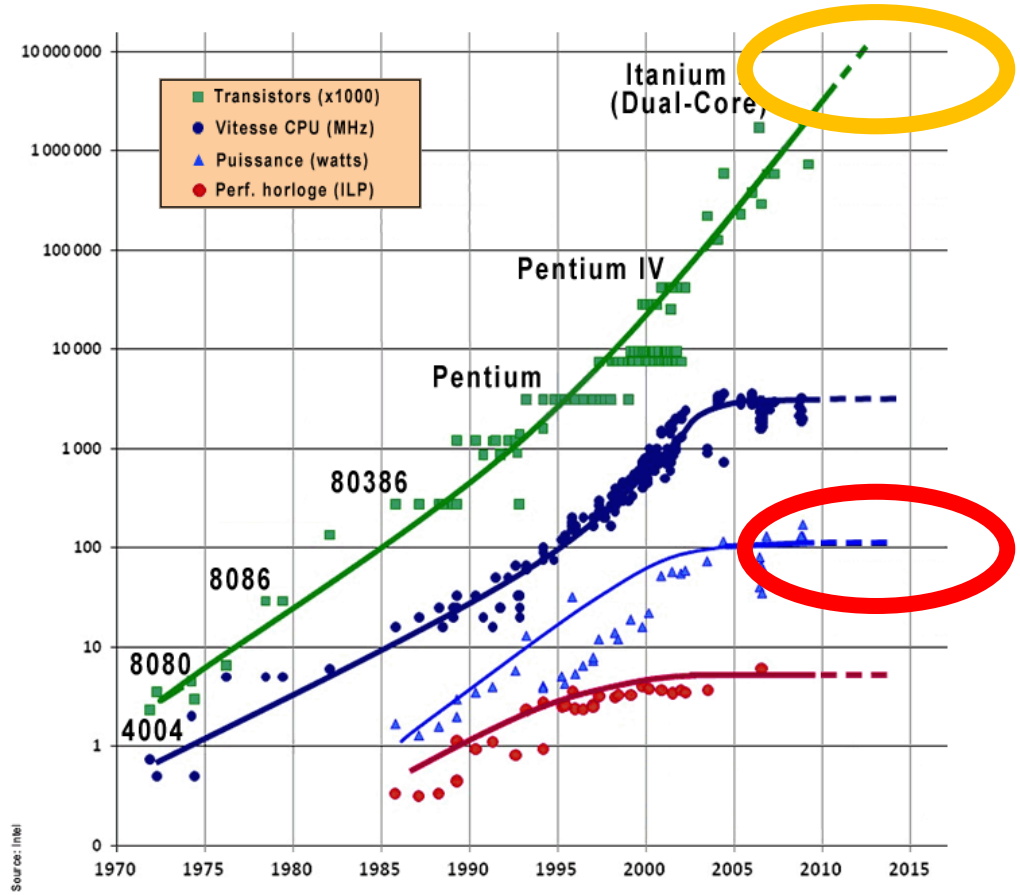
→ 10^{13} à 10^{16} connexions

→ 10^{16} (10 000 000 000 000 000) connexions

Loi de Moore

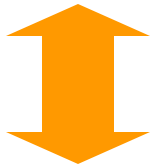
→ 10^{16} (10 000 000 000 000 000) connexions





(…)

MOTIVATIONS DES INGENIEURS :
S’inspirer de la neurobiologie pour construire des machines capables d’apprentissage et aptes à remplir des taches spécifiques : classification, prédiction, contrôle …



MOTIVATIONS DES BIOLOGISTES :
Utiliser des outils issus des mathématiques et de l’informatique en vue de construire des modèles plausibles du fonctionnement du système nerveux

Différents types d'apprentissage

Les 4 grandes familles de problèmes d'apprentissage

- Supervisé (observations X, classe y_d)
- Non supervisé (observations X)
- Semi-supervisé (mixte)
- Renforcement

Apprentissage supervisé

PART 1

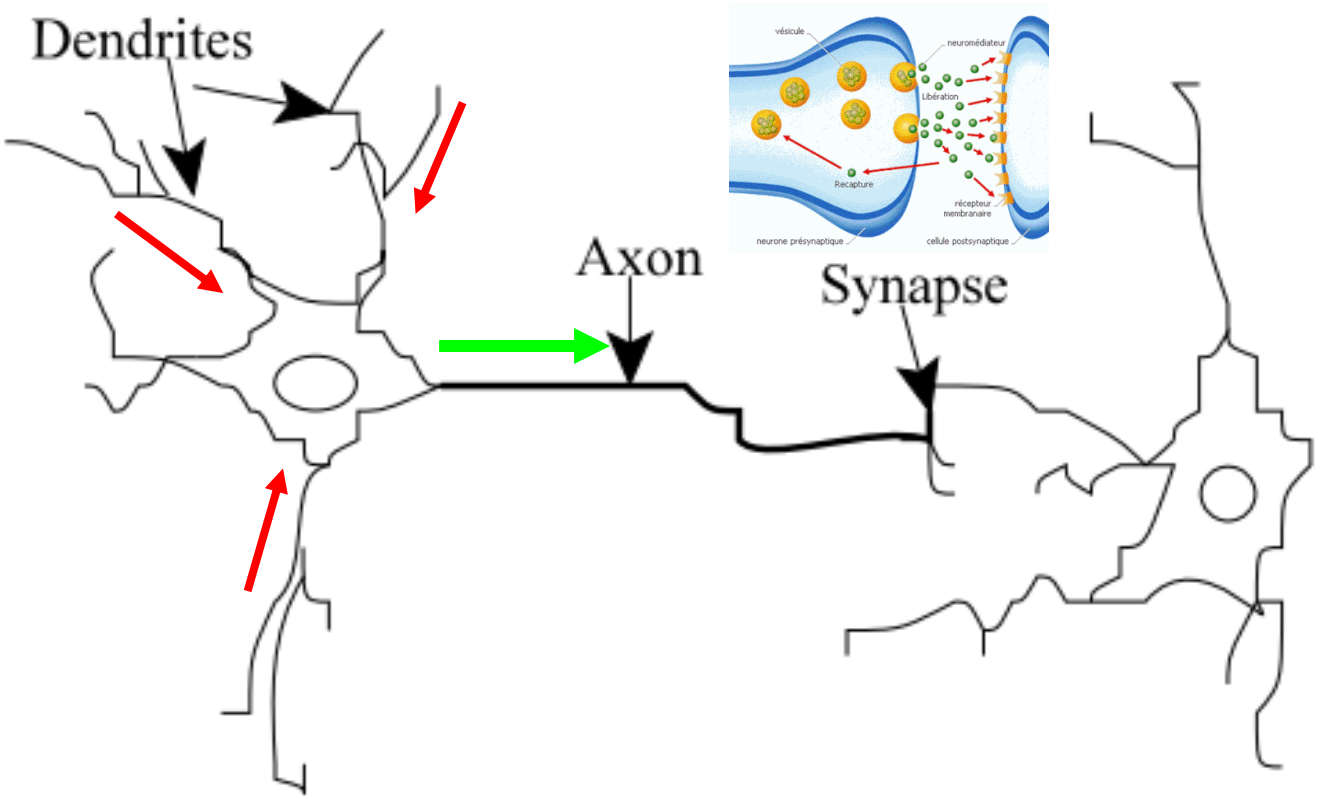
Sommaire

Réseaux de neurones et apprentissage supervisé

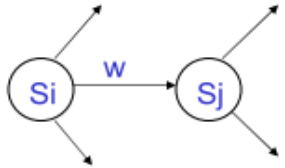
- **Du neurone naturel au neurone artificiel**
- **Le perceptron**
- **Les réseaux de neurones monocouches**
- **L'Adaline**
- **Les réseaux multicouches**
- **La rétro-propagation**

Propagation de l'influx nerveux

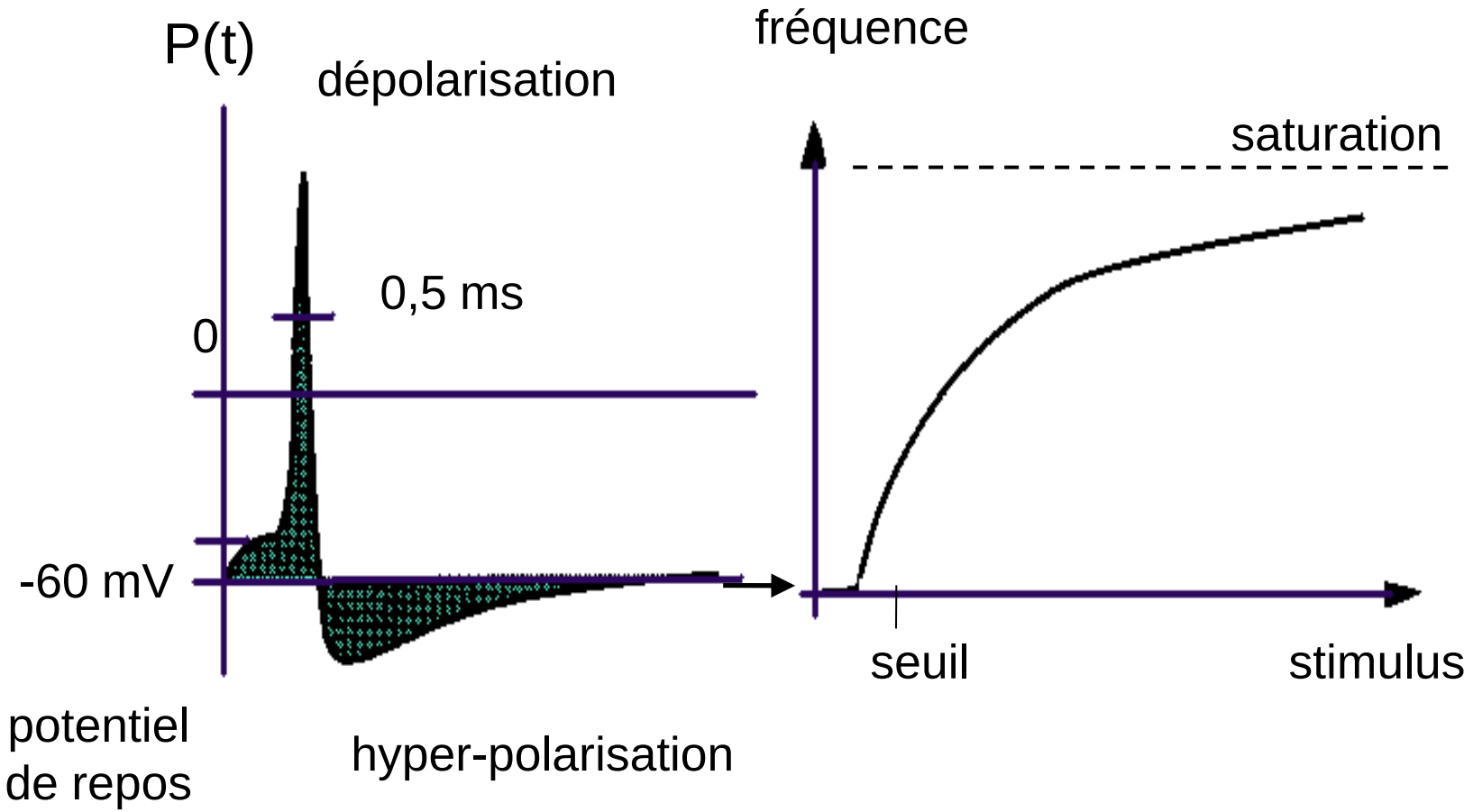
Fondements de neuro-physiologie :



Plasticité synaptique (règle de Hebb) :
dw/dt = SiSj



(...)


P(t)
dépolarisation
fréquence
saturation
0,5 ms
0
-60 mV
seuil
stimulus
potentiel
de repos
hyper-polarisation

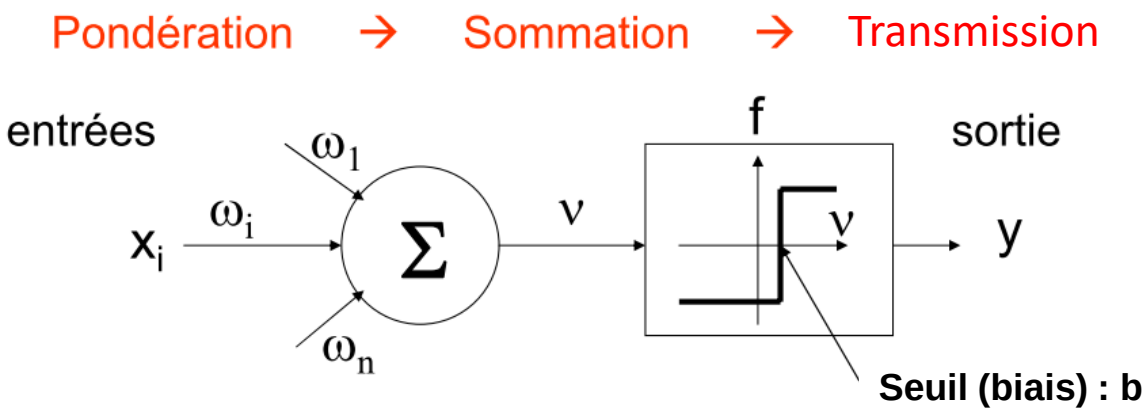
1945 : Neurone artificiel

Modélisation :

(1) Propagation de l'information (influx nerveux) des axones vers les dendrites **pondéré par des coefficients** synaptiques positifs (excitateurs) ou négatifs (inhibiteurs)

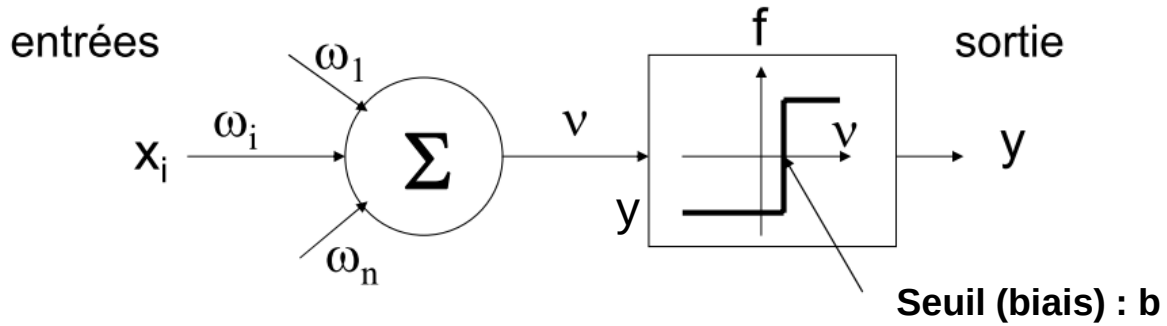
(2) Potentiel (ν) = **Somme des influx** (« entrées ») au niveau du corps cellulaire

(3) Transmission (« sortie ») **si la somme dépasse un seuil**



(…)

Pondération → Sommation → Transmission



$$y = f(\nu) = f(\sum_{i=1}^{D} \omega_i x_i + b)$$

!!

$$\left.\begin{array}{l} \omega_0 = b \\ x_0 = 1 \end{array}\right\} \rightarrow y = f(\nu) = f(\sum_{i=0}^{D} \omega_i x_i)$$

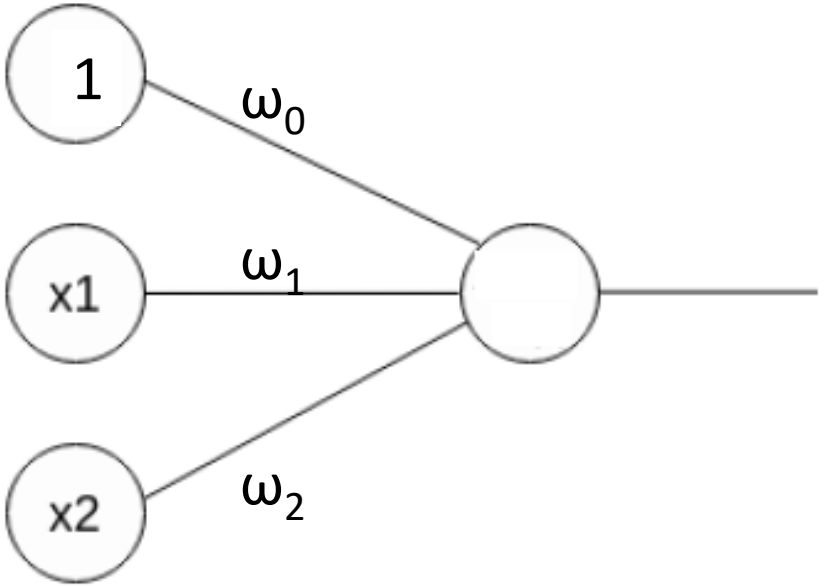
Notation : $\mathbf{X} = [1\ x_1 \dots x_D]^T$ le vecteur des **entrées**

$\mathbf{W} = [\omega_0\ \omega_2 \dots \omega_D]^T$ le vecteur des **poids**

$$y = f(\mathbf{W}^T.\mathbf{X})$$

Neurones à deux entrées

neurone à 2 entrées (x_1 , x_2) :



$$y = \text{signe}(W^T.X)$$

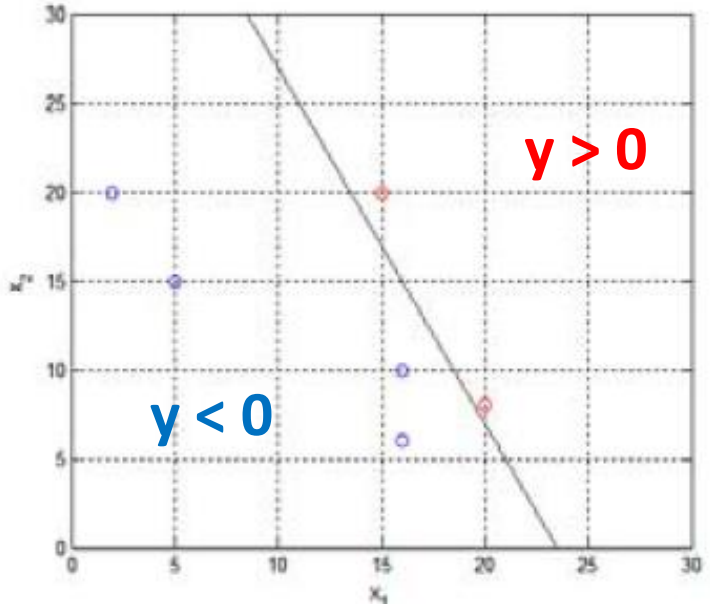
$$= \text{signe}(\omega_0 + \omega_1 x_1 + \omega_2 x_2)$$

Interprétation géométrique

Pour un neurone à 2 entrées :

L'équation $\omega_0 + \omega_1 x_1 + \omega_2 x_2 = 0$

est une droite qui sépare le plan en deux parties distinctes dont le signe est donné par la sortie du neurone :



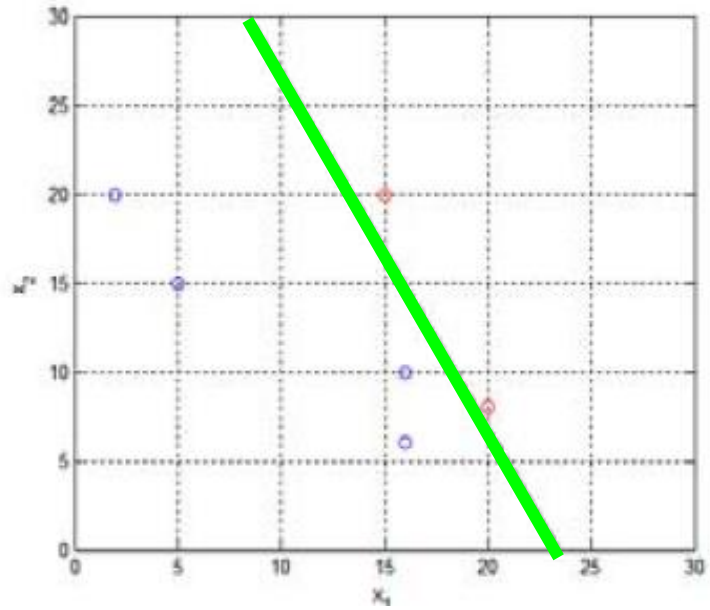
$$y = signe(W^T.X)$$

Aide à la decision : analyse de trafic

Discrimination camion/autres véhicules

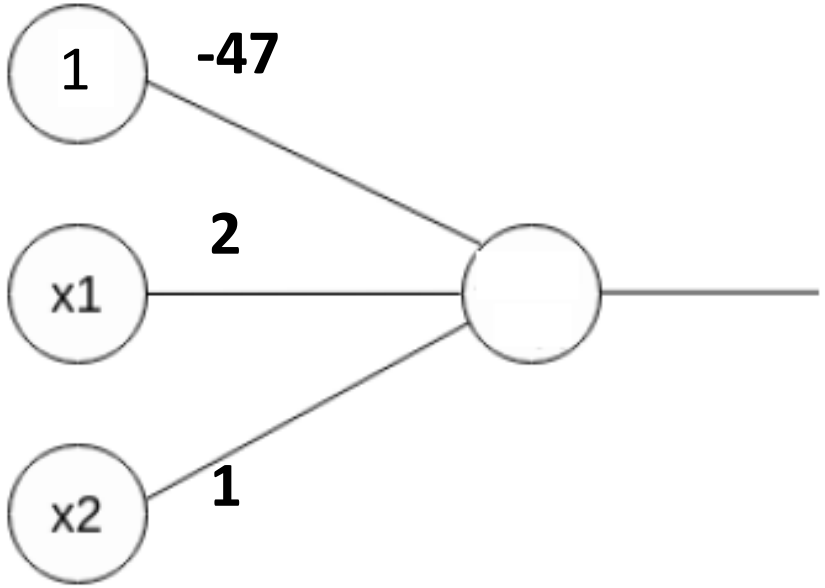
Véhicules : 2 descripteurs
- x_1 : longueur (m)
- x_2 : bruit (dB)

Véhicule	x_1	x_2	y_d
Camion1	20	8	1
Camion2	15	20	1
Car	16	10	-1
Voiture1	5	15	-1
Voiture2 (+remorque)	16	6	-1
Moto	2	20	-1



$-47 + 2x_1 + x_2$

Solution



Pb : comment trouver automatiquement les valeurs (optimales) des poids?

→ Apprentissage par l'exemple

Apprentissage

→ Modification des poids du réseau

(1) Le réseau est stimulé par l'environnement → **Présentation d'un exemple**

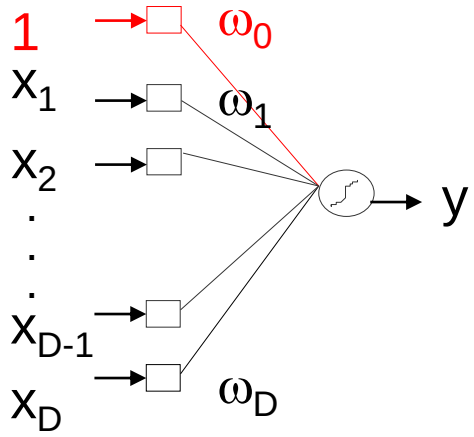
(2) Le réseau subit des changements en réponse à cette stimulation → **Modification des poids**

(3) Le réseau réagit différemment suite aux modification de sa structure interne → **Meilleure décision**

Formalisation du problème

Apprentissage supervisé : On dispose d'une base de données **étiquetées**

base d'apprentissage : couples {observation, classe} (X, y_d)



1) Observations décrites par *D features*

→ Vecteurs d'entrée de dimension *(1+D) : X = (1 x_1 x_2 … x_D)*

→ **un neurone à *(1+D)* entrées et une sortie y**

2) Problème à deux classes C_1 , C_2 (binaire)

→**valeurs désirées** $y_d = +1$ si $X \in C_1$

$y_d = -1$ si $X \in C_2$

(…)

3) Objectif :

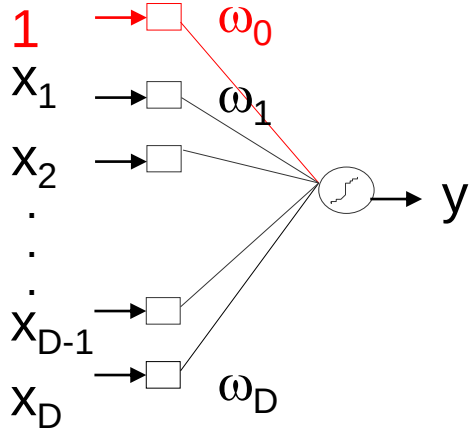
Trouver le vecteur des poids $\omega = (\omega_0\ \omega_1 \dots \omega_D)$ tel que

$W^T.X > 0 \quad \forall X \in C_1 \qquad (y_d = +1)$

$W^T.X < 0 \quad \forall X \in C_2 \qquad (y_d = -1)$

Si $X \in C_1$ est mal classé : $W^T.X < 0$ et $y_d\ W^T.X < 0$

Si $X \in C_2$ est mal classé : $W^T.X > 0$ et $y_d\ W^T.X < 0$



Règle de Hebb : lorsque deux neurones sont excités conjointement, leur connexion se renforce :

$$\Delta W_j = \lambda y^* x_j \text{ avec } \lambda \text{ le pas d'apprentissage}$$

1955: Algorithme du Perceptron

Initialiser **W(0)** aléatoirement

- Apprentissage :
  - (1)Tirer au hasard un exemple X de la base d'apprentissage
  - (2) Si y_d $\mathbf{W}^T(t).X < 0$ c'est à dire **si X est mal classé**

    Modifier les poids ω (pour que X soit bien classé) suivant la relation :

    $$\mathbf{W(t+1) = W(t) + \lambda \Delta W \text{ avec } \Delta W = y_d X}$$

    λ : pas d'apprentissage

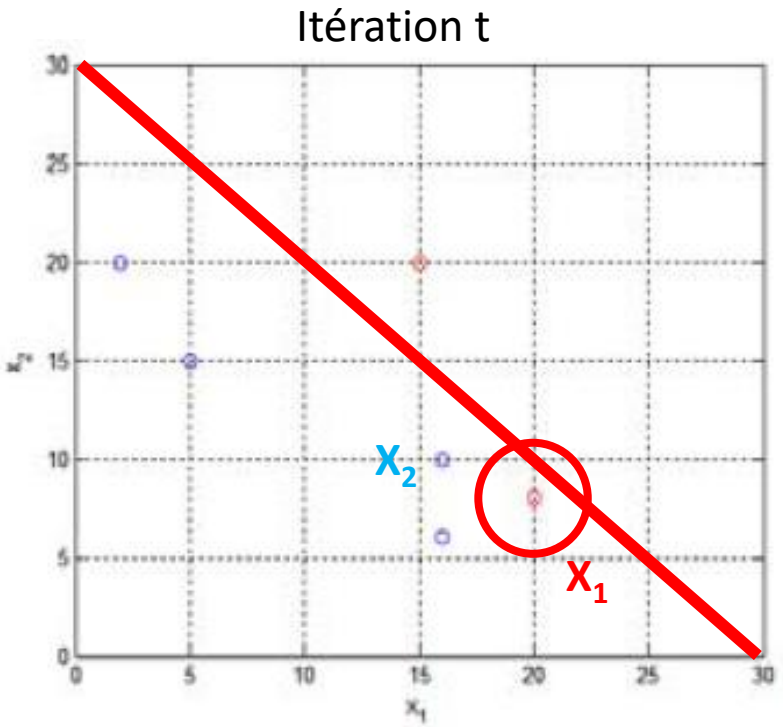
    Incrémenter le compteur de mises à jour t=t+1
  - (3) critère d'arrêt :

    Si tous les exemples sont bien appris (classés)

    Fin

    Sinon : retour en (1)

Fun time



1) Déterminer graphiquement W(0)

2) Vérifier que l'exemple X_1 est mal classé

3) Appliquer la règle de mise à jour du vecteur W(0) avec :

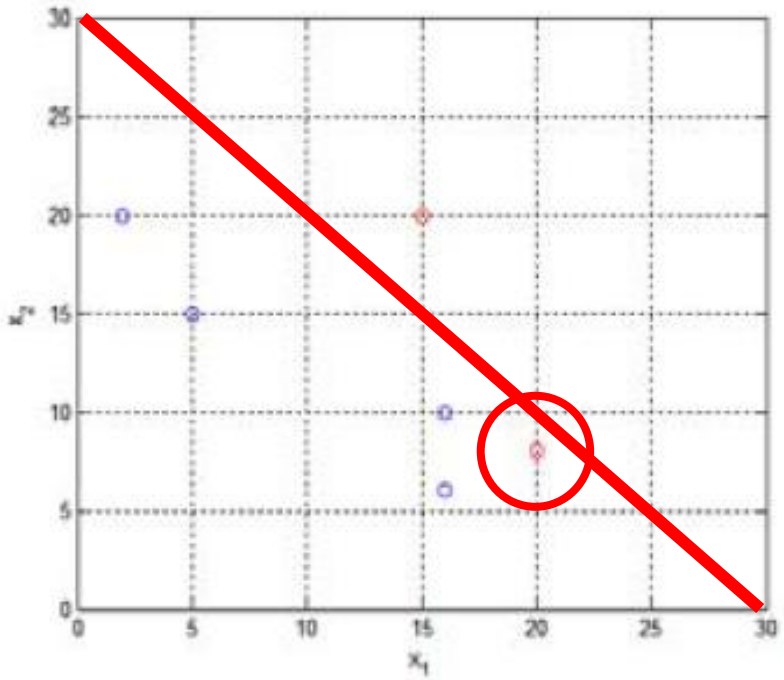
- $\lambda = 0,1$
- $\lambda = 0,05$
- $\lambda = 0,01$

Tracer dans chaque cas la frontière de décision et évaluer le taux de reconnaissance

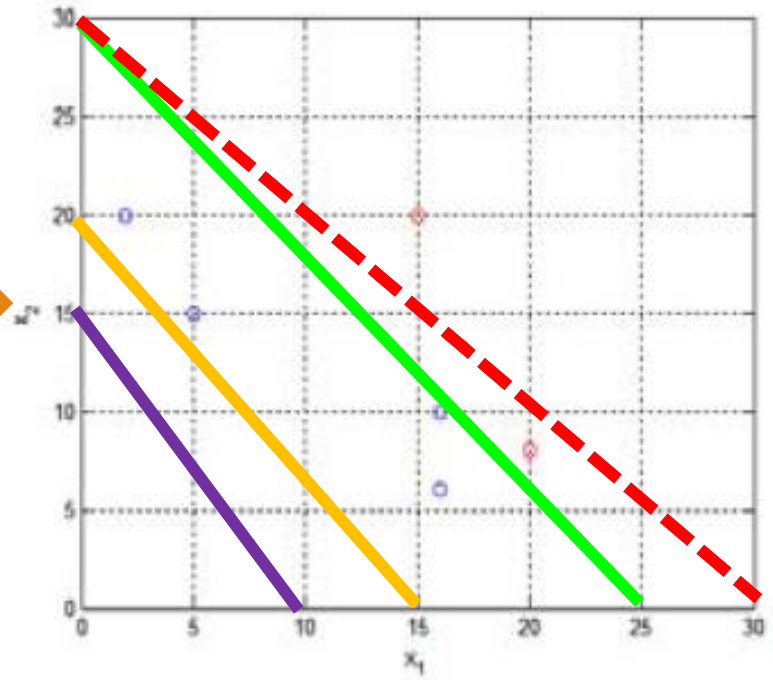
Dans le dernier cas, vérifier que X_1 et X_2 sont bien classés

(…)

Itération t



Itération (t+1)



→ $W(0) = (-30\ 1\ 1)^T$

- $\lambda = 0,1$

→ $W(1) \approx (-30\ 3\ 2)^T$

- $\lambda = 0,05$

→ $W(1) \approx (-30\ 2\ 1,5)^T$

- $\lambda = 0,01$

→ $W(1) \approx (-30\ 1,2\ 1)^T$

Preuve de convergence

$X \in C_1$ mais $WX < 0$

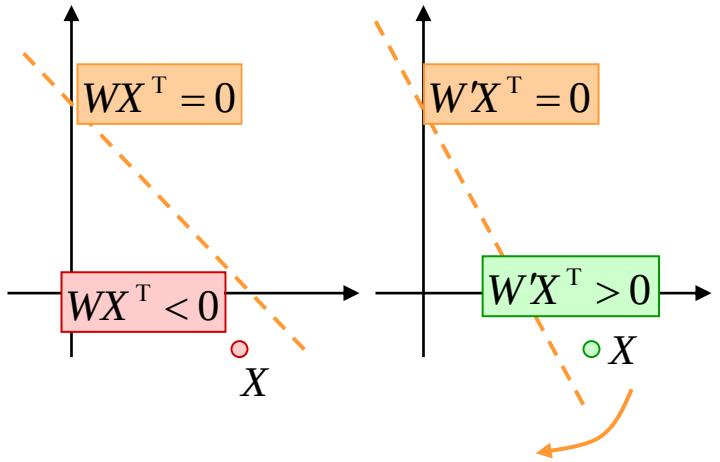
on cherche ΔW tel que $W'X^T = (W + \Delta W)X^T > 0$:

On a : $W'X^T = (W + \lambda.1.X)X^T = \left(WX^T + \lambda \|X\|^2\right) > WX^T$

$X \in C_2$ mais $WX^T > 0$

on cherche ΔW tel que $W'X^T = (W + \Delta W)X^T < 0$:

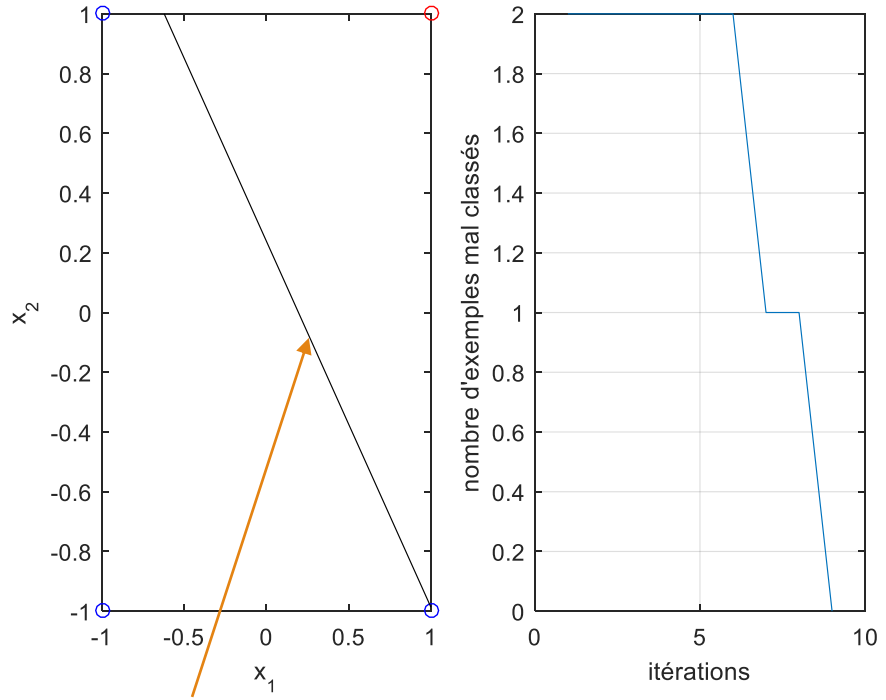
On a : $W'X^T = (W + \lambda.(-1).X)X^T = \left(WX - \lambda \|X\|^2\right) < WX^T$



Quels que soient l'ensemble de données en entrée et leur classification désirée, l'algorithme d'apprentissage du perceptron, **convergera** vers un **ensemble correct de poids**, et ceci en un **nombre fini d'opérations** si un tel ensemble existe [Rosenblatt,1962].

Opérateurs AND, OR et XOR

AND



$W = (-0.145 \quad 0.746 \quad 0.609)^T$

XOR