Math A Data

Introduction à l'Intelligence Artificielle avec les Mathématiques du Lycée

ENS
École Normale Supérieure

PSL Research University Paris

Collège de France 1530

Académie de Lille
Liberté
Égalité
Fraternité

Ils nous soutiennent :

CFM Insight.Data.Clarity. | PR[AI]RIE PaRis Artificial Intelligence Research InstitutE | Fondation Collège de France | Agir pour l'éducation — Un enjeu scientifique pour la société | Fondation Covea | LVMH | Fondation Engie | INFRAVIA

Algorithme d'Intelligence Artificielle

Challenge: répondre à une question à partir d'informations contenues dans des données *D*

$$D = (d_1, d_2, \ldots) \longrightarrow \boxed{\textbf{Algorithme}} \longrightarrow \hat{R}$$

: estimation de la bonne réponse *R*
en faisant peu d'erreurs

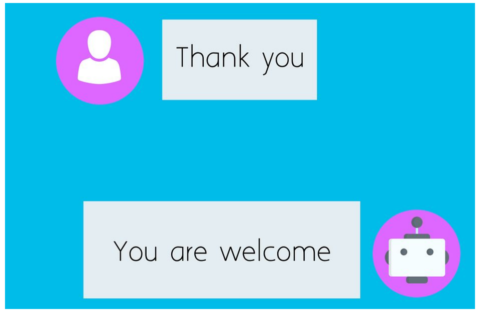
Classification/régression

Voiture, robotique...
Contrôle de systèmes

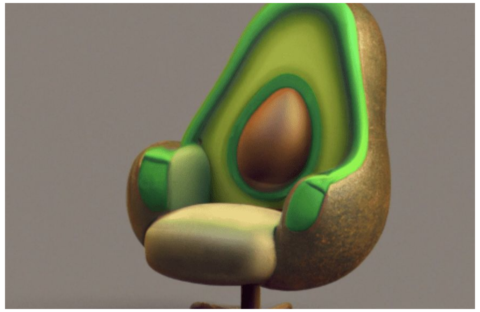
Reconnaissance,
diagnostic médical

Prédiction météo

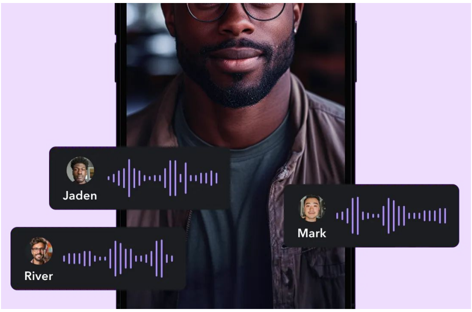
Génération de données



Textes : agent,
chatbot...



Images : vidéo, clip,
publicité...



Sons : musiques,
voix, doublage...

1- Algorithmes d’IA par apprentissage statistique

2- Comprendre l’IA avec les math du Lycée

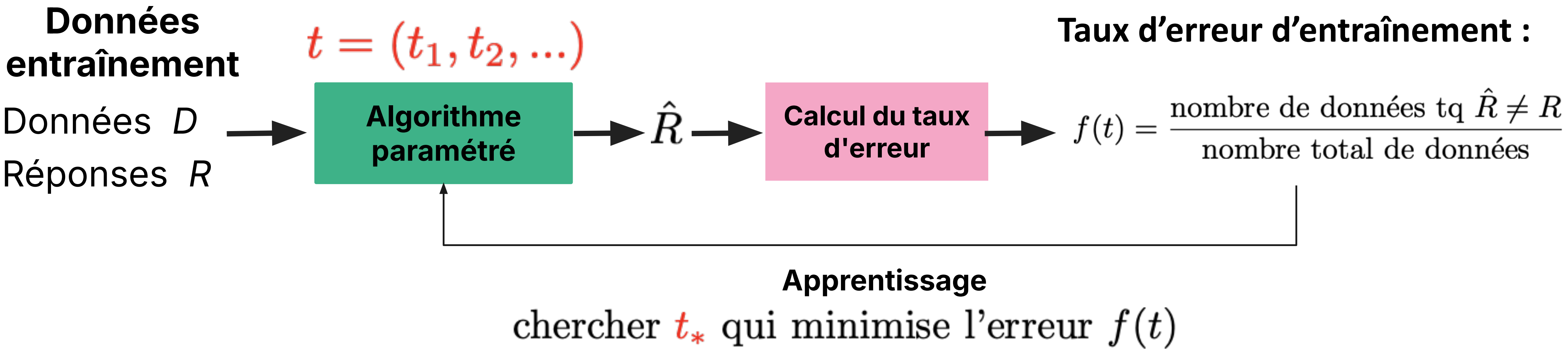
3- Les réseaux de neurones

4- Enseigner les maths avec des challenges d’IA

1- Algorithmes d'IA par apprentissage statistique

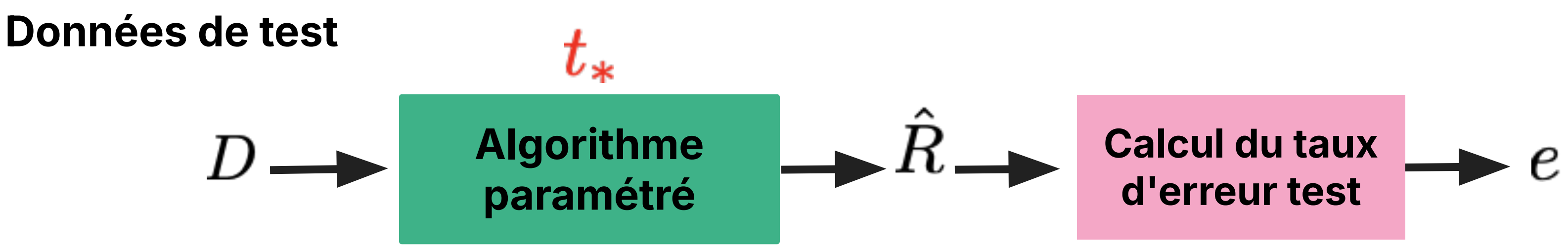
Estimer la réponse R à une question à partir de données numériques D en faisant peu d'erreurs.

Apprentissage des paramètres de l'algorithme avec une base de données d'entraînement



Test de généralisation de l'apprentissage

Test de l'algorithme sur des nouvelles données après l'apprentissage



L'algorithme généralise si les erreurs de test et d'entraînement sont similaires: $e \approx f(t_*)$,

Il faut suffisamment d'exemples d'entraînement (loi des grands nombres).

Sur-apprentissage si l'erreur est bien plus grande au test qu'à l'entraînement: $e \gg f(t_*)$

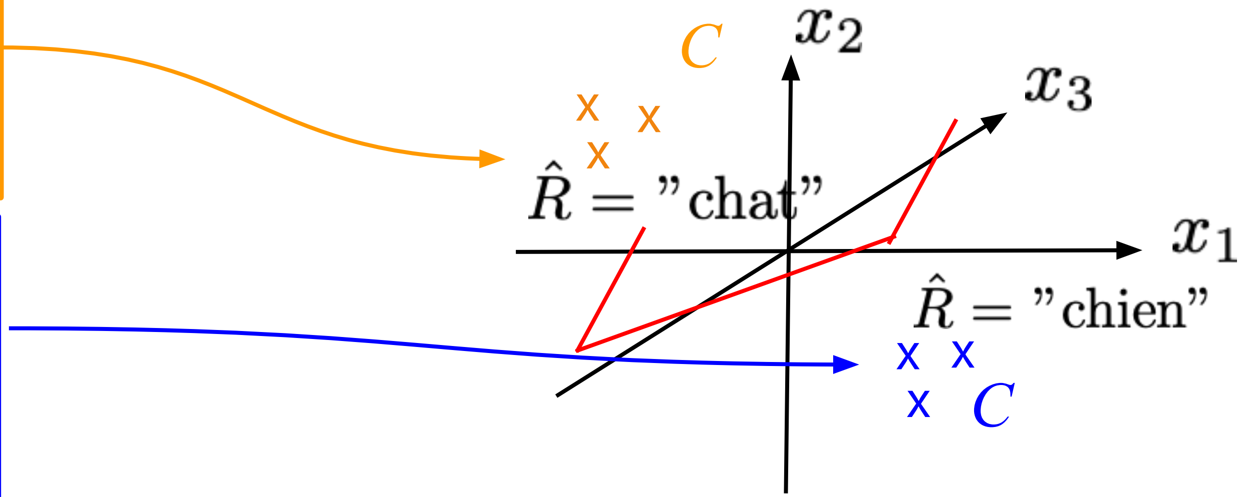
Pas assez d'exemples d'entraînement *(apprendre par coeur).*

2- Les mathématiques du lycée au coeur de l’IA

Estimer la réponse avec des caractéristiques

Images: données toutes très différentes

Caractéristiques $C = (x_1, x_2, \ldots)$ **similaires** parmi les chats et parmi les chiens et **différentes** entre les chats et les chiens.



Algorithme paramétré

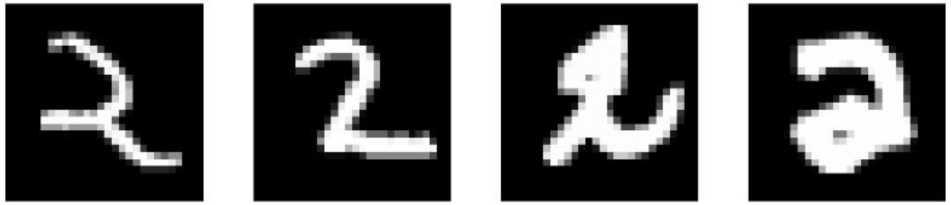
Donnée D → **Calcul des caractéristiques** → C → **Estimation** → $\hat{R}$

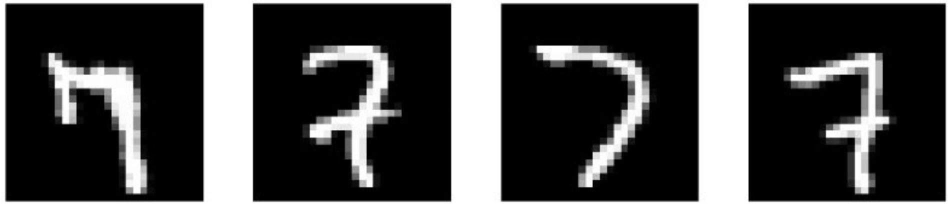
Caractéristique

Classification de chiffres manuscrits

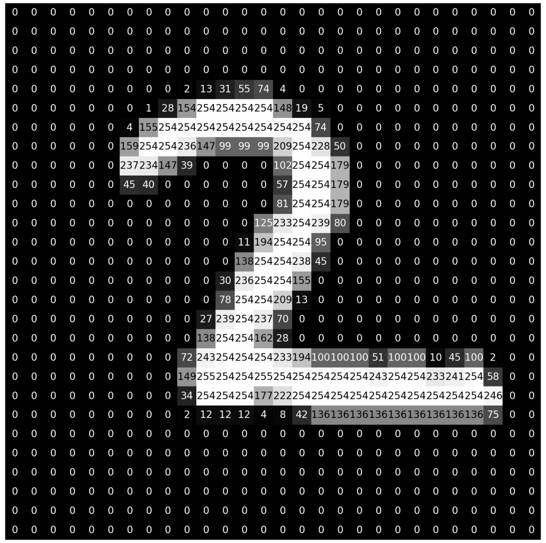
Challenge: Quel est le chiffre *R* écrit dans une image *D* ?

On commence par différencier deux classes de chiffres: 2 ou 7



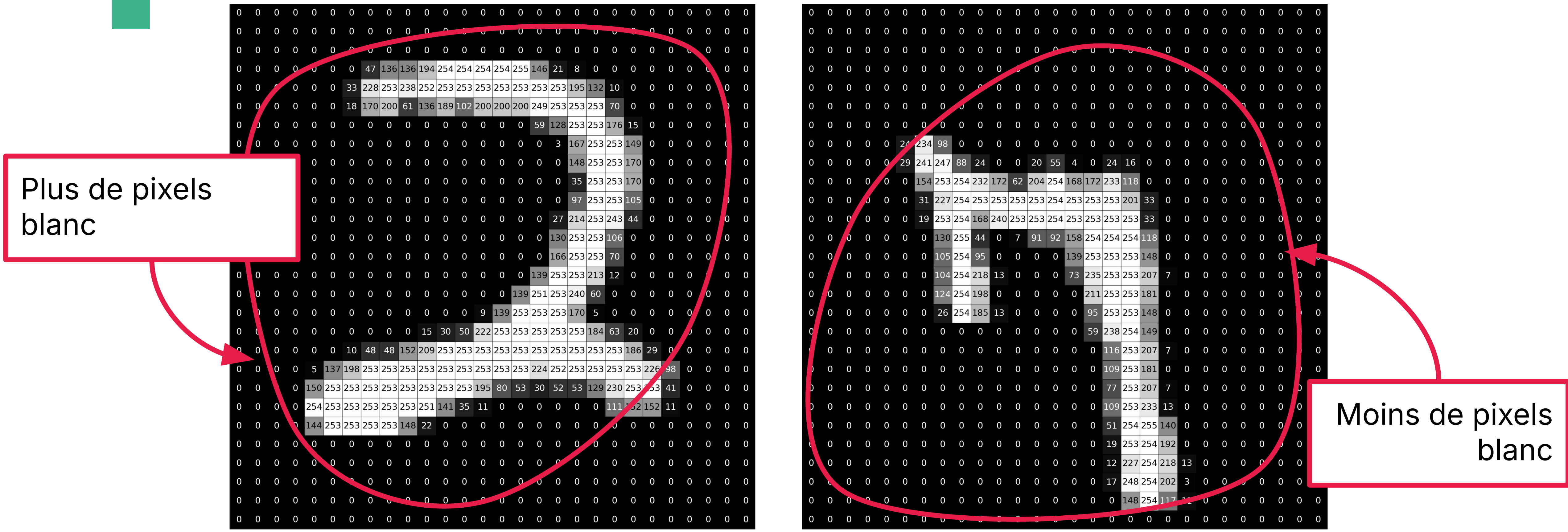


Donnée *D* **:** image, tableau de pixels $D = (d_1, d_2, ..., d_{784})$



valeurs des pixels

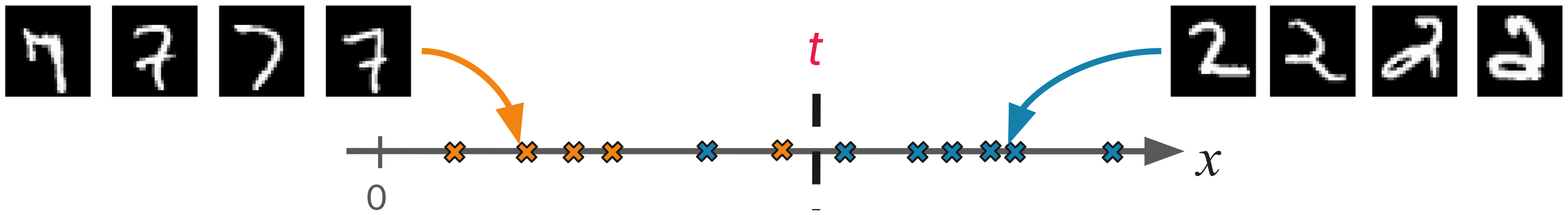
Quelle caractéristique choisir ?



Exemple d'une caractéristique $C = x$ qui différencie les 2 des 7:

x = valeur moyenne des pixels de l'image

Estimation avec 1 caractéristique



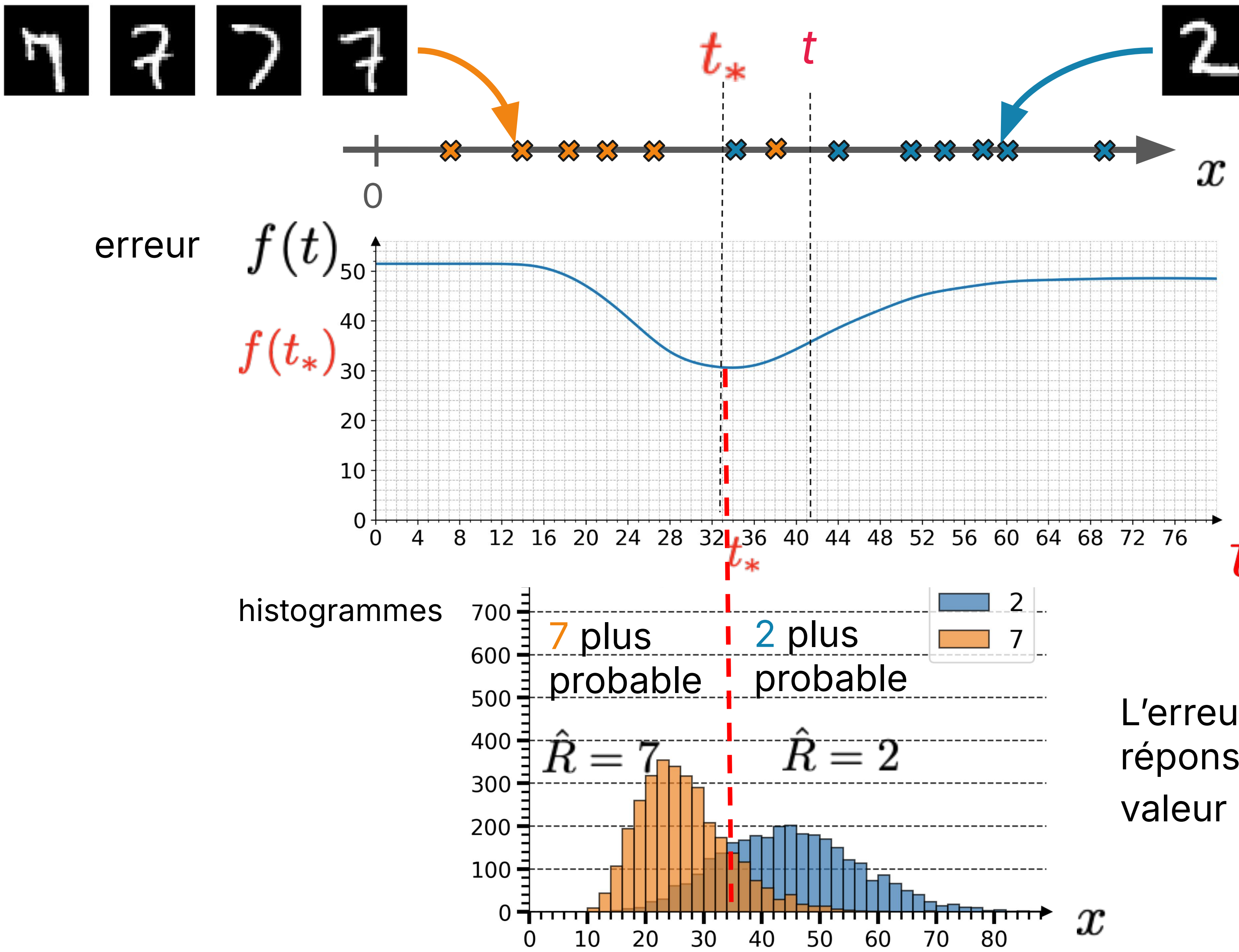
caractéristique des **données d'entraînement**

Estimation : $$\hat{R} = \begin{cases} 7 & \text{si } x < t \\ 2 & \text{si } x \geq t \end{cases}$$

Erreur moyenne : $$f(t) = \frac{\text{nombre de données tq } \hat{R} \neq R}{\text{nombre total de données}} = \hat{P}rob(\hat{R} \neq R)$$

Probabilité empirique que $\hat{R} \neq R$ calculée sur les données d'entraînement.

Apprentissage du paramètre



Apprentissage : chercher *t* pour minimiser *f(t)*

L'erreur est minimum si $\hat{R}$ est la réponse la plus probable sachant la valeur de la caractéristique $C = x$.

Apprentissage optimale et probabilité conditionnelle

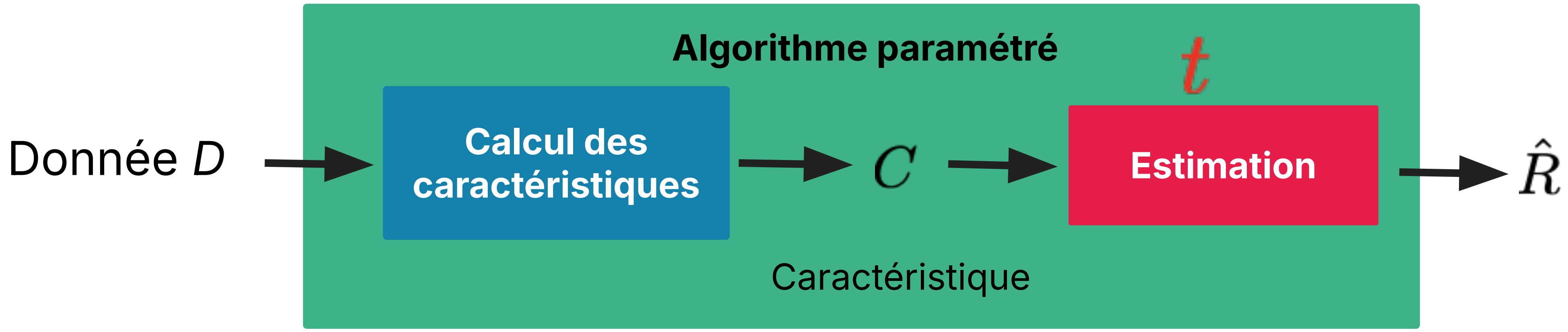
Minimiser l'erreur $\hat{P}rob(\hat{R} \neq R)$ $\Leftrightarrow$ Maximiser la précision $\hat{P}rob(\hat{R} = R) = 1 - \hat{P}rob(\hat{R} \neq R)$

$$\hat{P}rob(\hat{R} = R) = \sum_{C} \hat{P}rob(\hat{R} = R \,|\, C)\, \hat{P}rob(C)$$

La précision $\hat{P}rob(\hat{R} = R)$ est donc max si $\hat{P}rob(\hat{R} = R \,|\, C)$ est max.

L'erreur est donc minimum si $\hat{R}$ est la réponse la plus probable sachant C

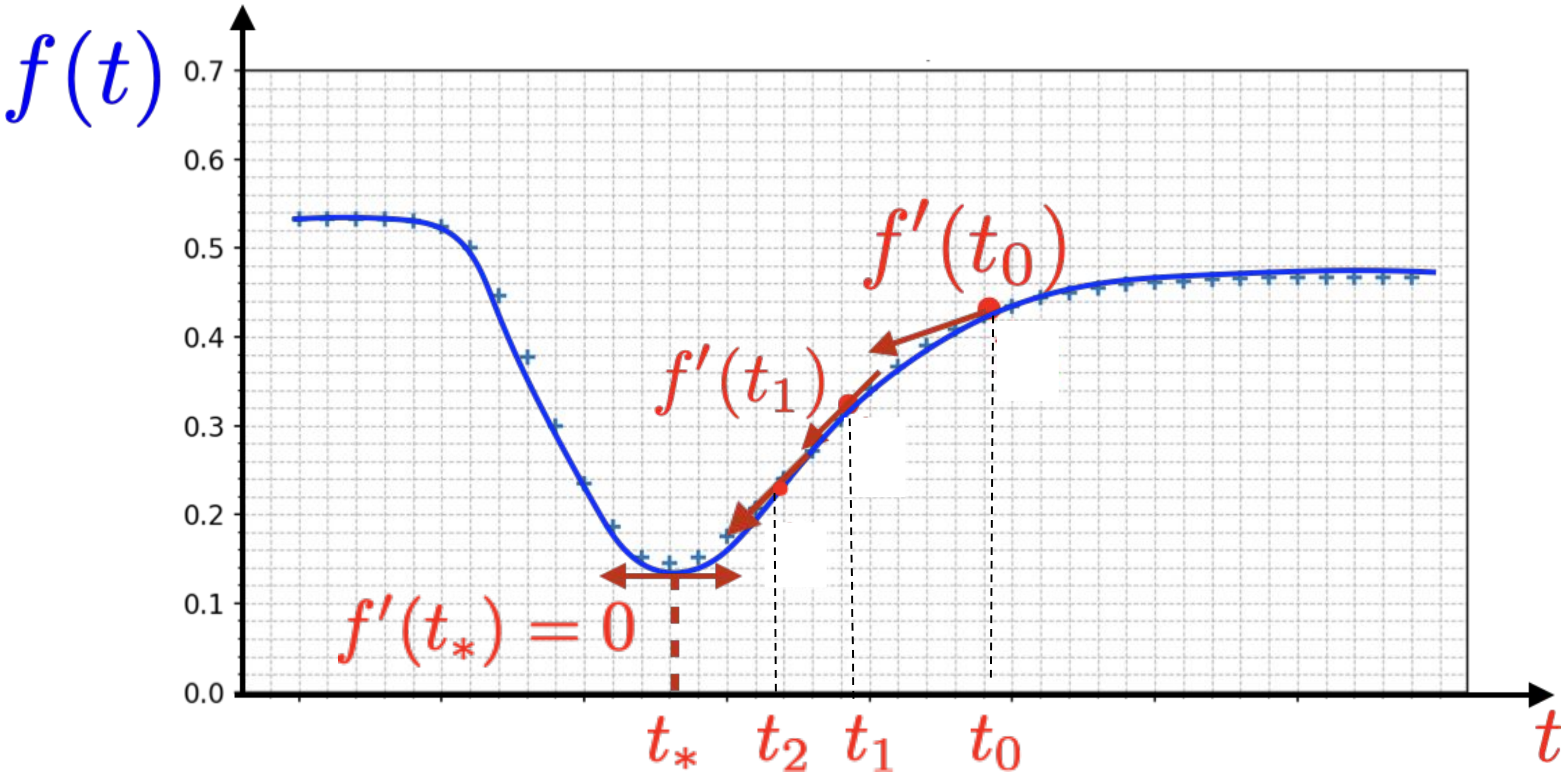
Optimisation d'un Algorithme d'IA



- **Apprendre:** chercher t_* qui minimise l'erreur $f(t)$ à l'entraînement.
- **Généralisation:** erreur du même ordre au test qu'à l'entraînement.
- **Modèle:** choix de caractéristique C qui réduit l'erreur minimum $f(t_*)$ peut être apprise *(réseaux de neurones)*.

Minimiser l'erreur par descente de dérivée

L'erreur *f(t)* est une moyenne sur les exemples d'entraînement: long à calculer.
On veut chercher le minimum de *f(t)* en faisant le moins d'essais possibles.



inconnu

Descente de dérivée: $t_{n+1} = t_n - \epsilon\, f'(t_n) \xrightarrow[n \to \infty]{} t_*$ si unique min local.

Généralisation et loi des grands nombres

À quelle condition un algorithme fera-t-il aussi peu d'erreurs sur des exemples de tests que sur des exemples d'entraînement ?

L'apprentissages minimise la moyenne empirique des erreurs sur m exemples d'entraînement qui sont supposés être **indépendants** *(problèmes des biais)*

Loi des grands nombres: si m est assez grand, les erreurs X_i des données ont une moyenne qui converge vers l'espérance (*ne dépend quasiment pas des exemples*):

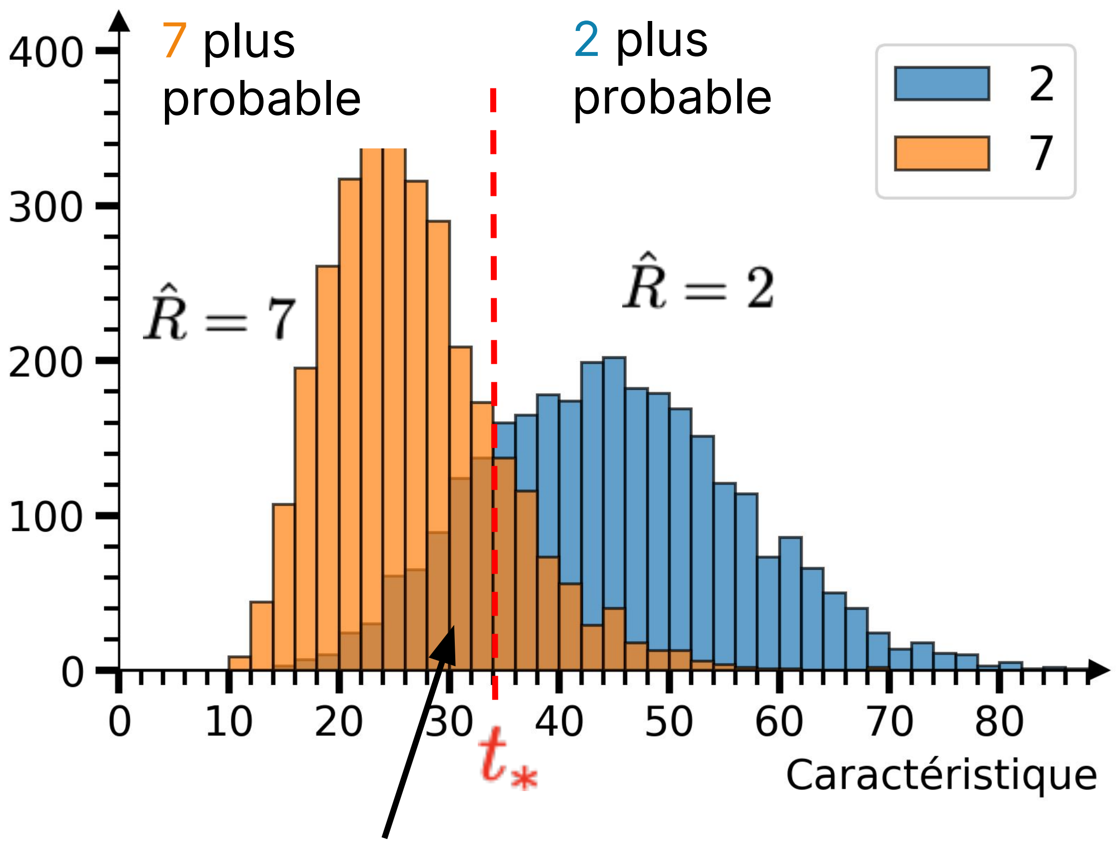
$$\frac{X_1 + \ldots + X_m}{m} \xrightarrow[m \to \infty]{} \mathbb{E}(X) \quad \text{en probabilité}$$

→ généralise si on a assez d'exemples d'entraînement indépendants: *m* grand.

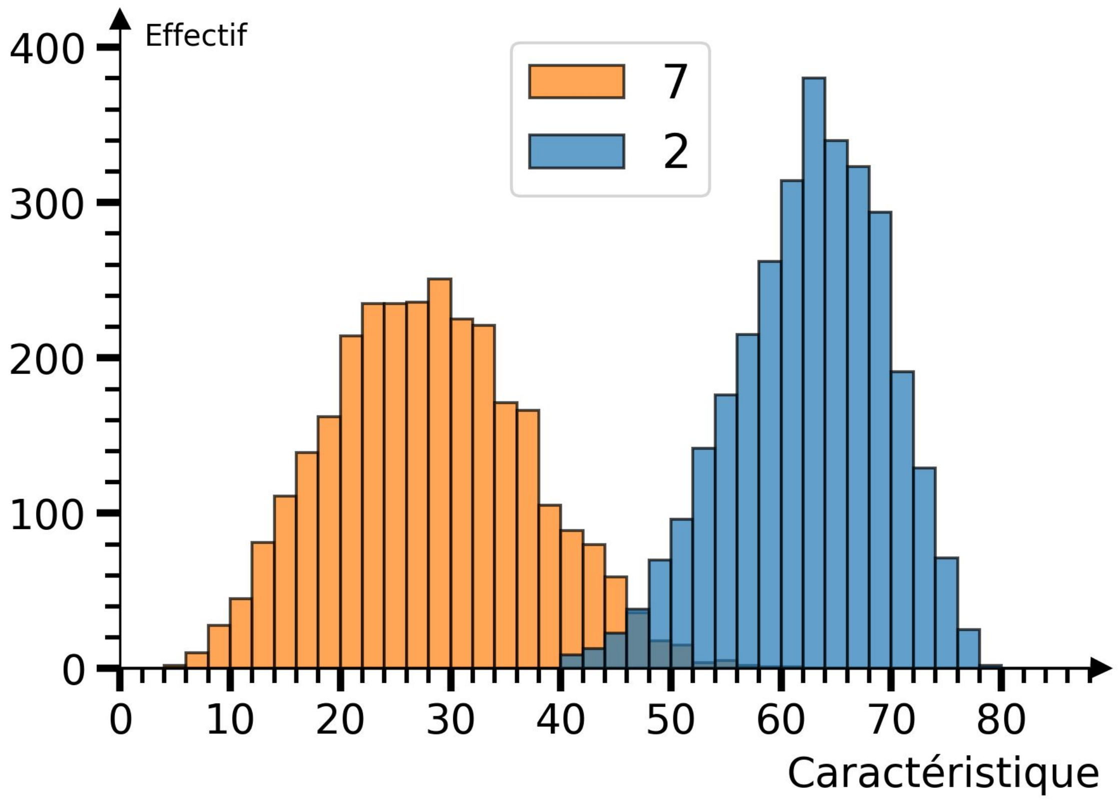
→ le nombre m d'exemples doit être grand relativement au nombre de paramètres ajustés lors de l'apprentissage.

Choix de la caractéristique

On choisit la caractéristique qui sépare mieux les classes



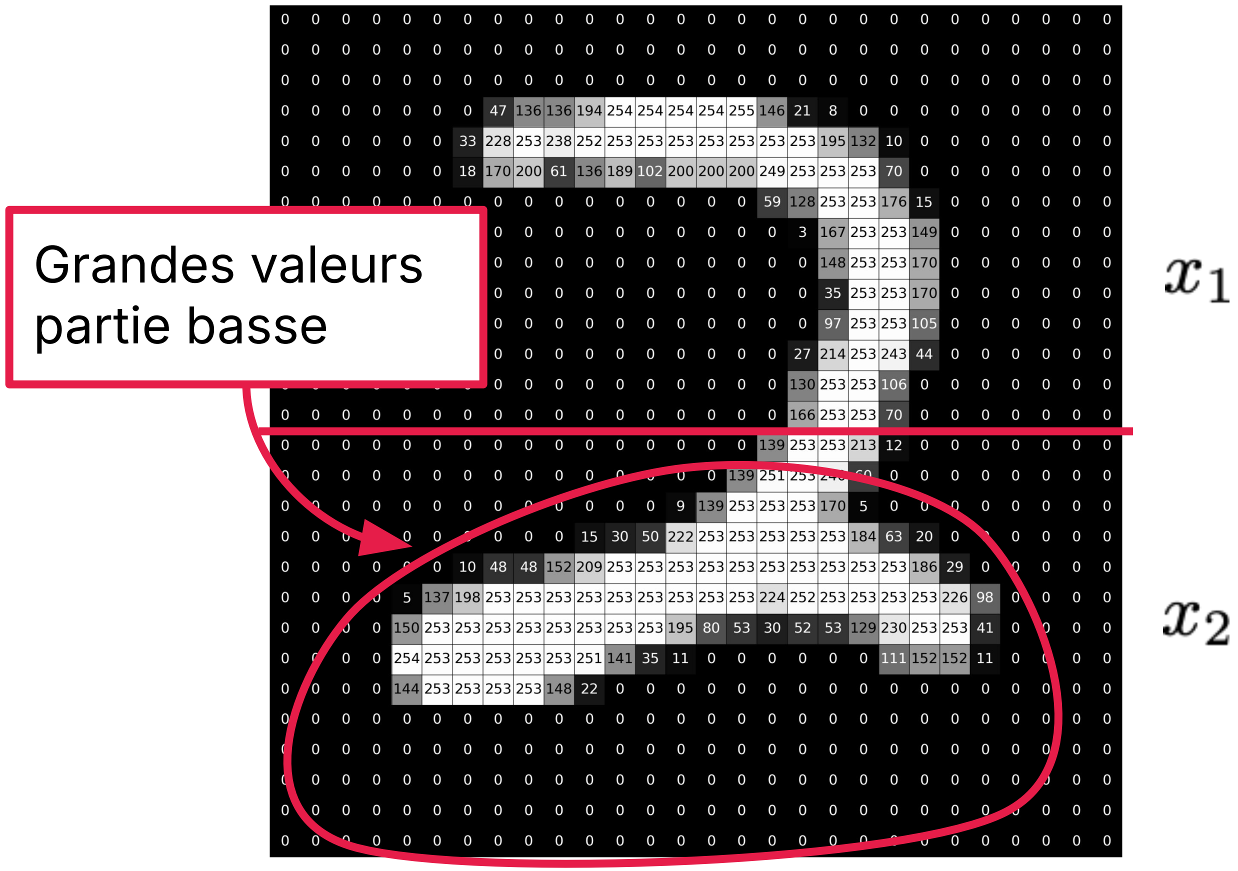
Erreur minimum = surface de recouvrement des histogrammes

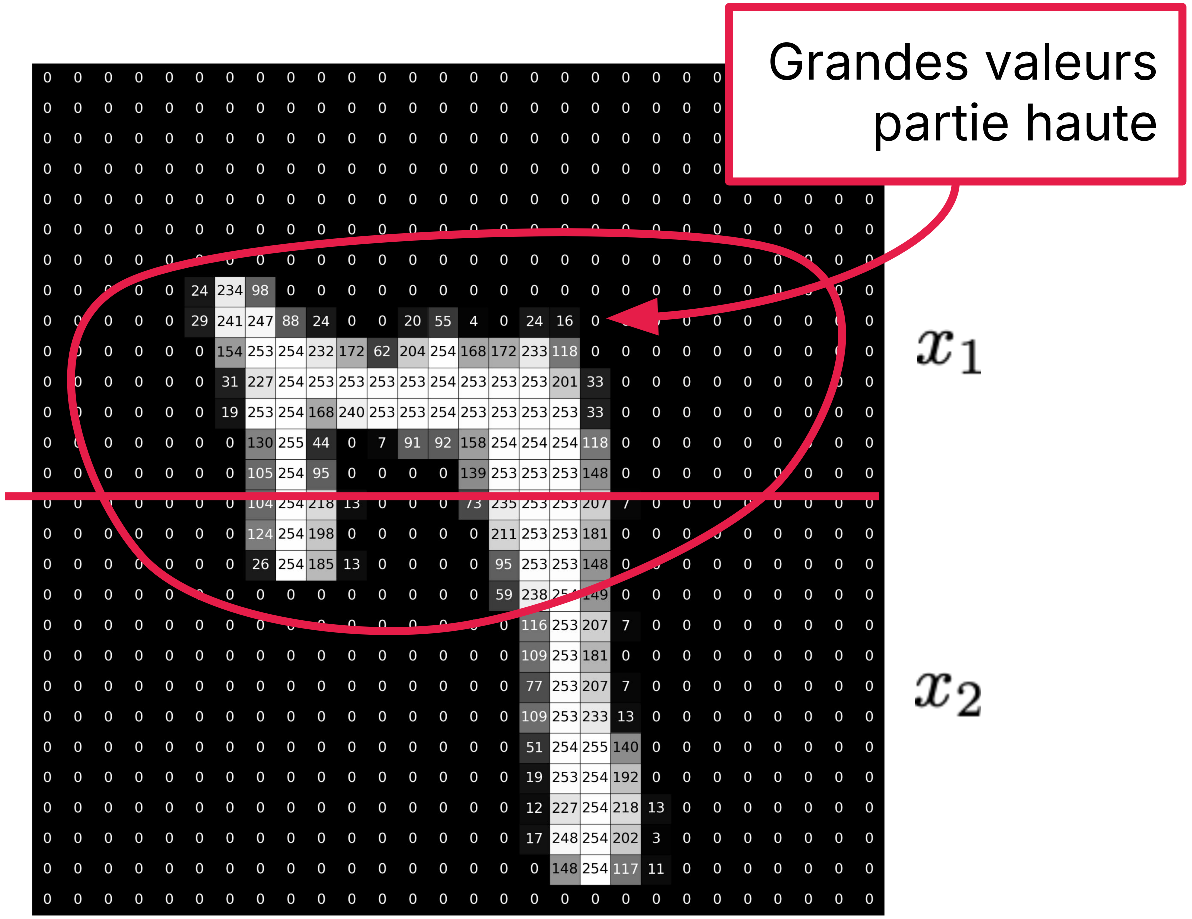


Erreur plus faible: distributions mieux séparées

Moins d'erreur avec 2 caractéristiques

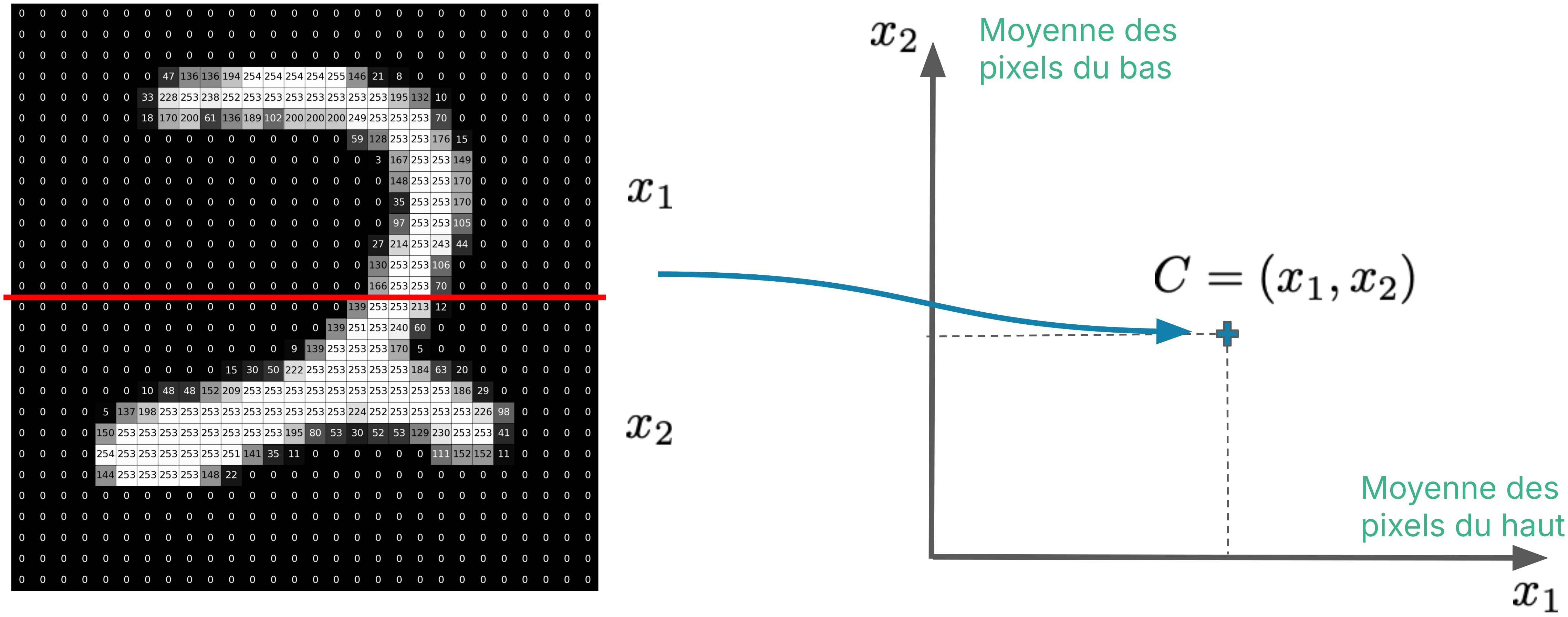
Mieux différencier les classes avec 2 caractéristiques complémentaires





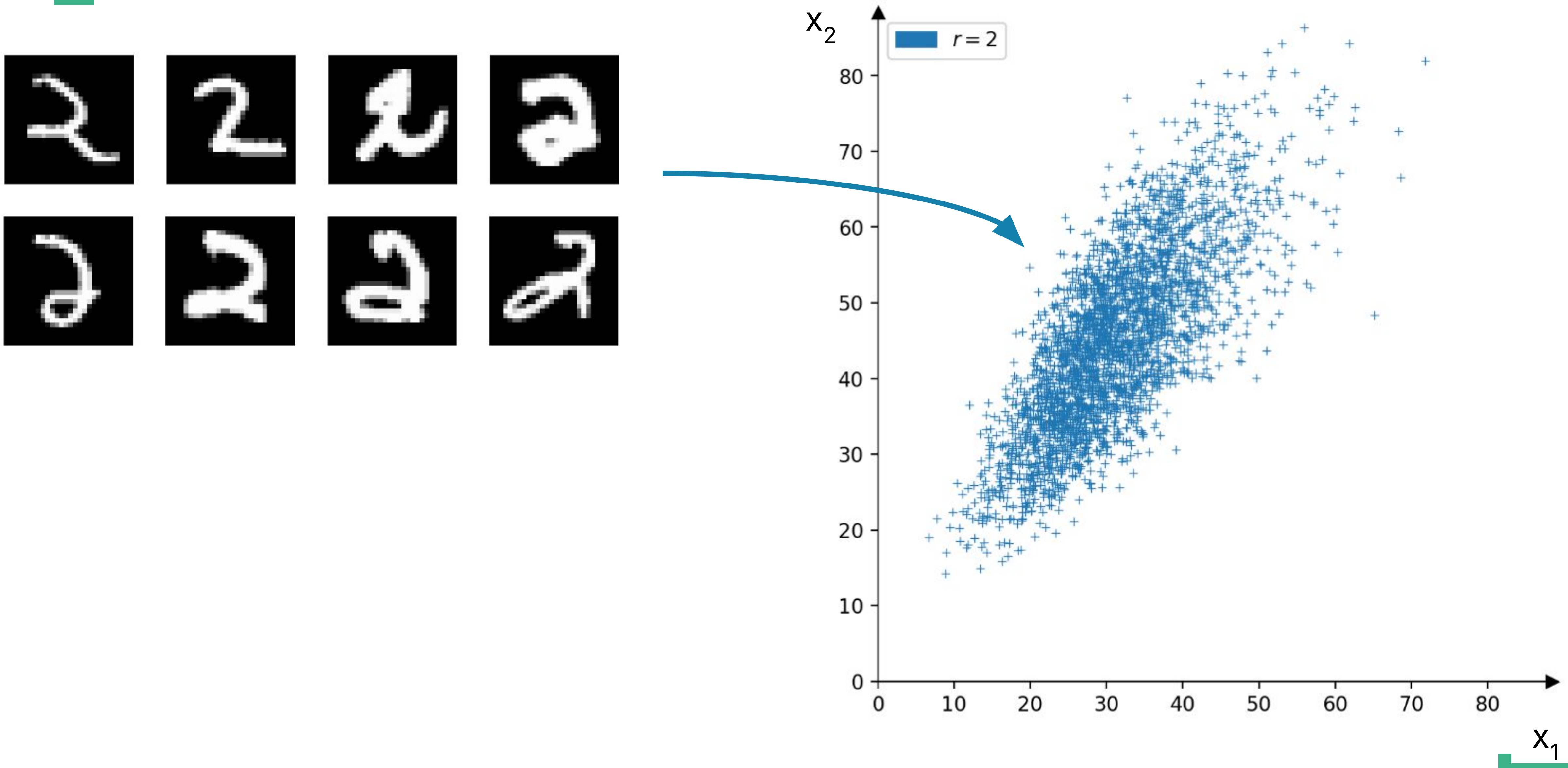
(x_1, x_2) : moyennes des pixels dans les parties haute et basse de l'image

Représentation d'une caractéristique dans un plan

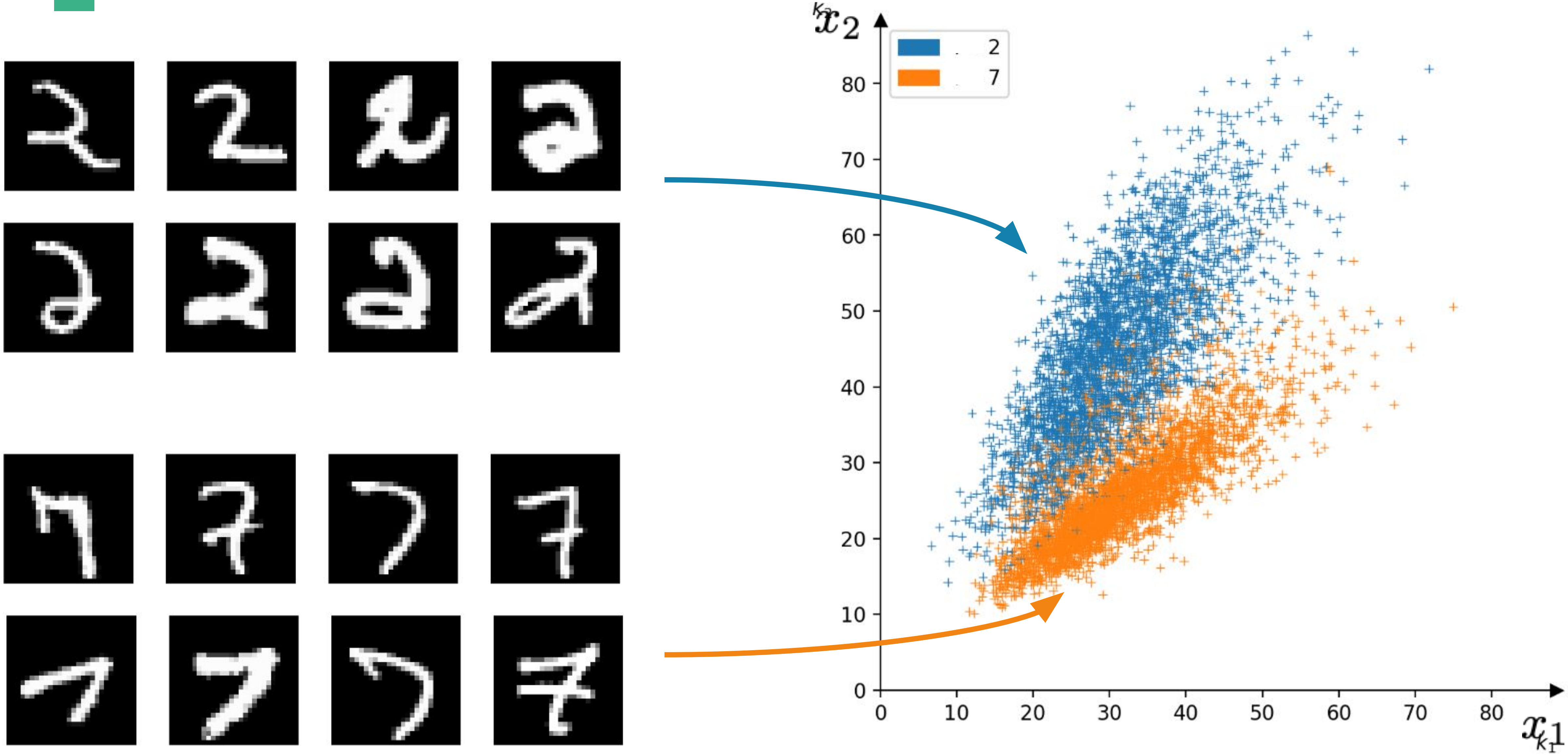


(x_1, x_2) : moyennes des pixels dans les parties haute et basse de l'image

Caractéristiques des images d'entraînement



Caractéristiques des images d'entraînement



Frontière des classes : droite

L'erreur est minimum si $\hat{R}$ est
la réponse la plus probable sachant C

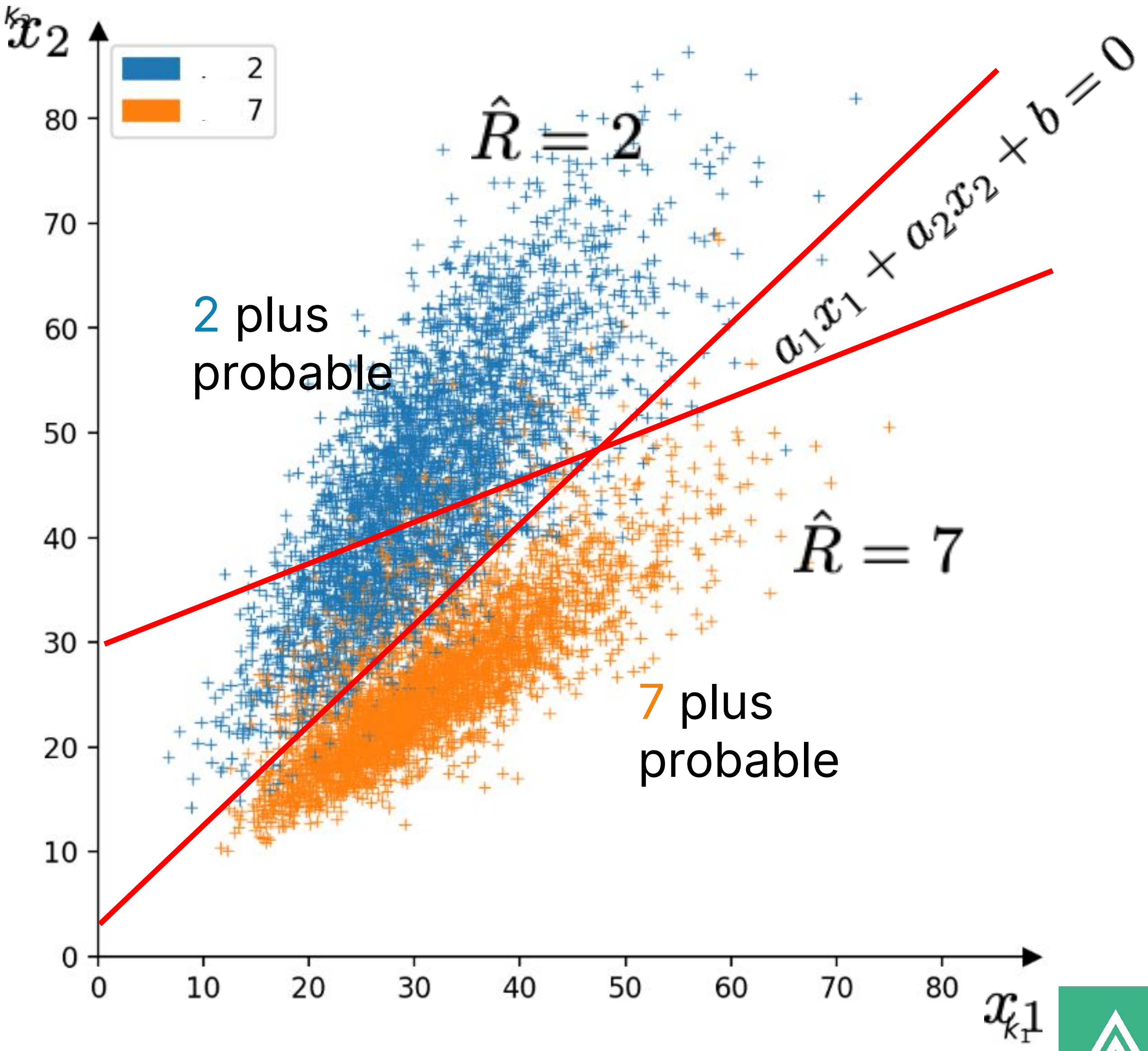
Frontière approximée par une droite

$$a_1 x_1 + a_2 x_2 + b = 0$$

Paramètres de la droite: $t = (a_1, a_2, b)$

L'erreur $f(t)$ dépend de $t = (a_1, a_2, b)$

Apprendre: chercher t qui minimise f(t).



Représentation vectorielle des classes

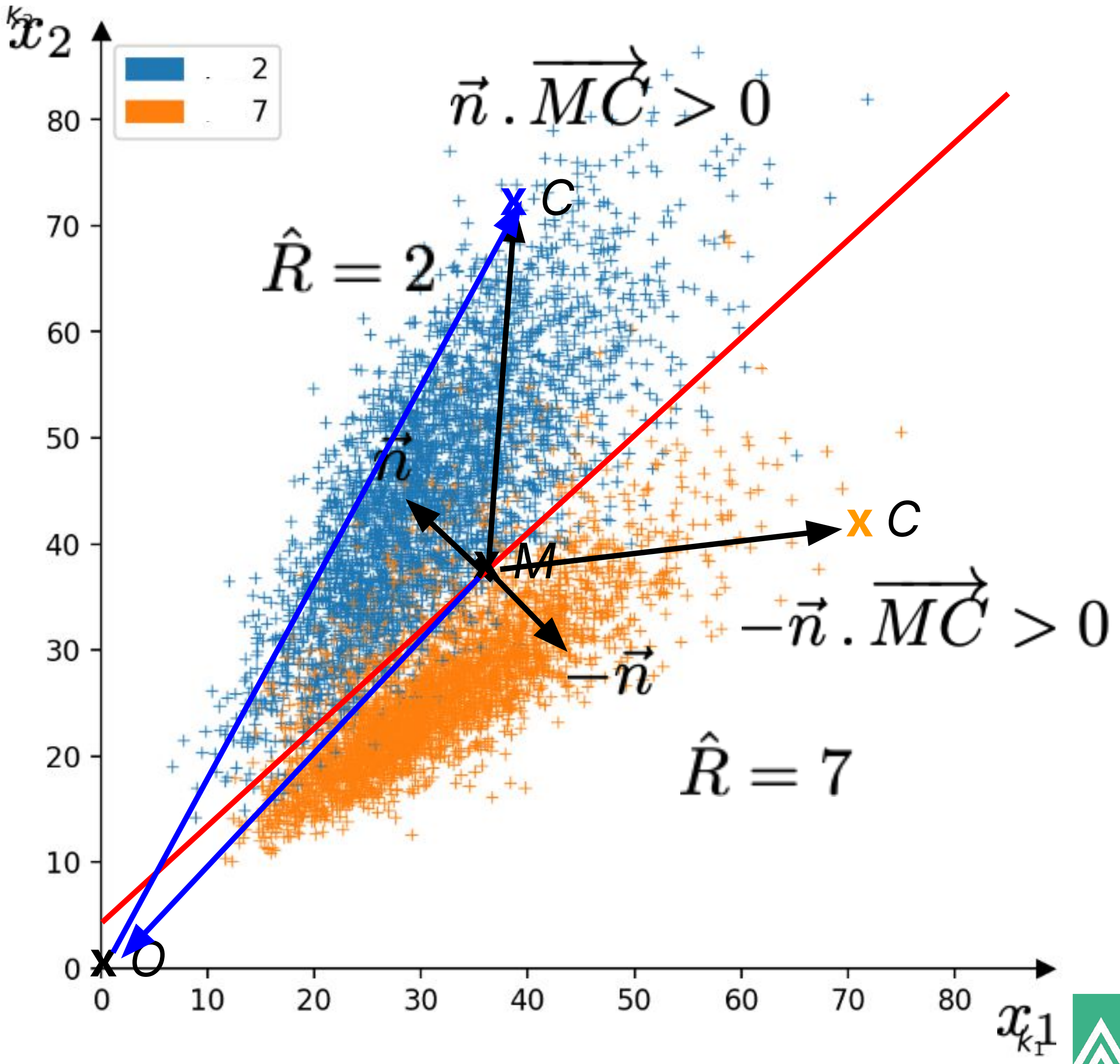
un point sur la droite : M

vecteur normal $\vec{n} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$

Pour une caractéristique $C = (x_1, x_2)$

$$\hat{R} = \begin{cases} 2 & \text{si } \vec{n} \cdot \overrightarrow{MC} > 0 \\ 7 & \text{si } -\vec{n} \cdot \overrightarrow{MC} > 0 \end{cases}$$

$$\begin{aligned} \vec{n} \cdot \overrightarrow{MC} &= \vec{n} \cdot \overrightarrow{MO} + \vec{n} \cdot \overrightarrow{OC} \\ &= b + a_1\, x_1 + a_2\, x_2 \end{aligned}$$



Plus de classes : découpage par morceaux de droites

Point M au centre des caractéristiques

Chaque $\vec{n}_k$ pointe vers la classe k

$$\boxed{\begin{array}{l}\hat{R} = k \text{ tel que} \\ v_k = \vec{n}_k \cdot \overrightarrow{MC} \text{ est maximum}\end{array}}$$

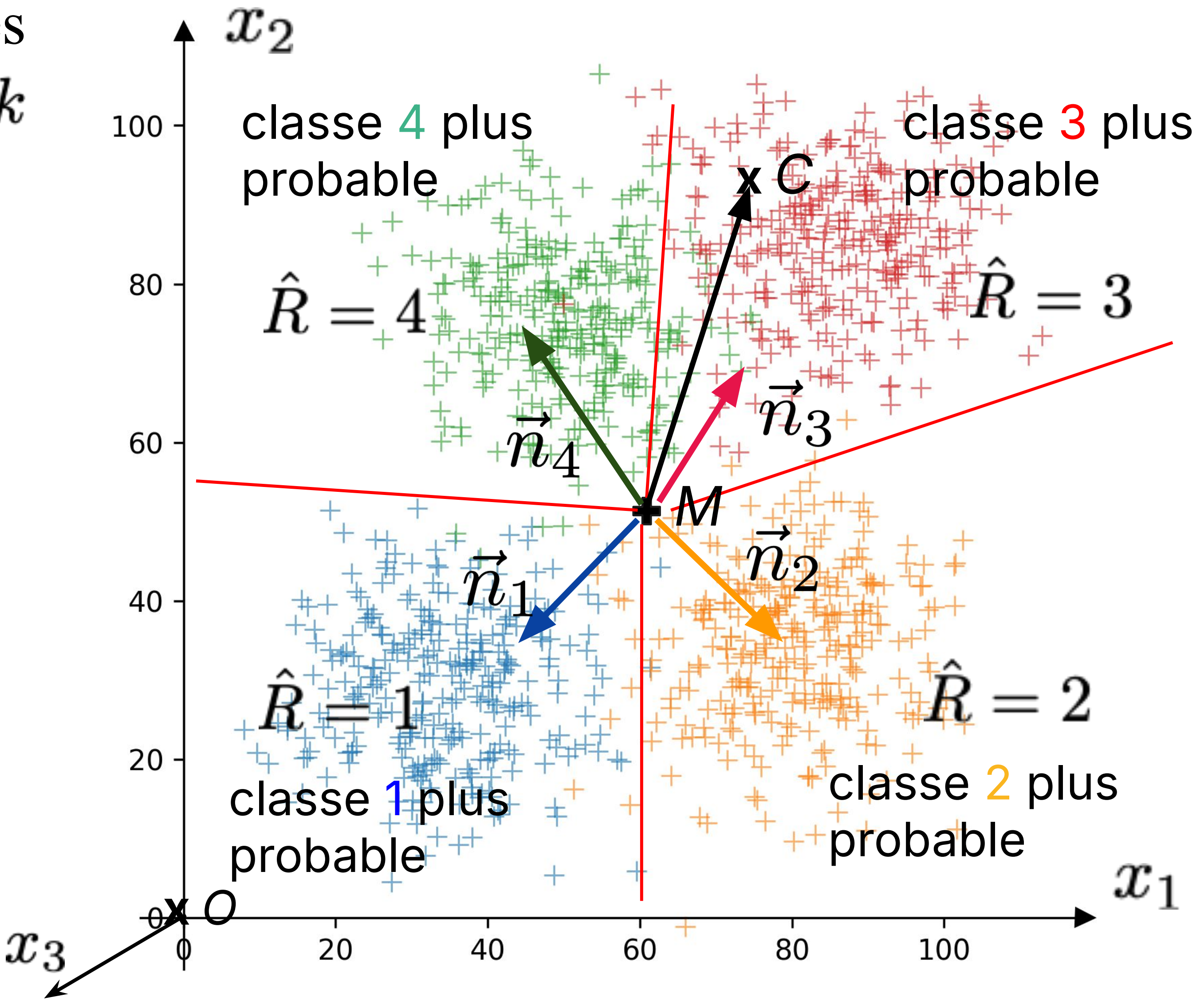
vraisemblance de la classe k

$$v_k = b_k + \vec{n}_k \cdot \overrightarrow{OC}$$

Apprendre: chercher

$$t = \Big((\vec{n}_1, b_1), (\vec{n}_2, b_2), (\vec{n}_3, b_3), (\vec{n}_4, b_4) \Big)$$

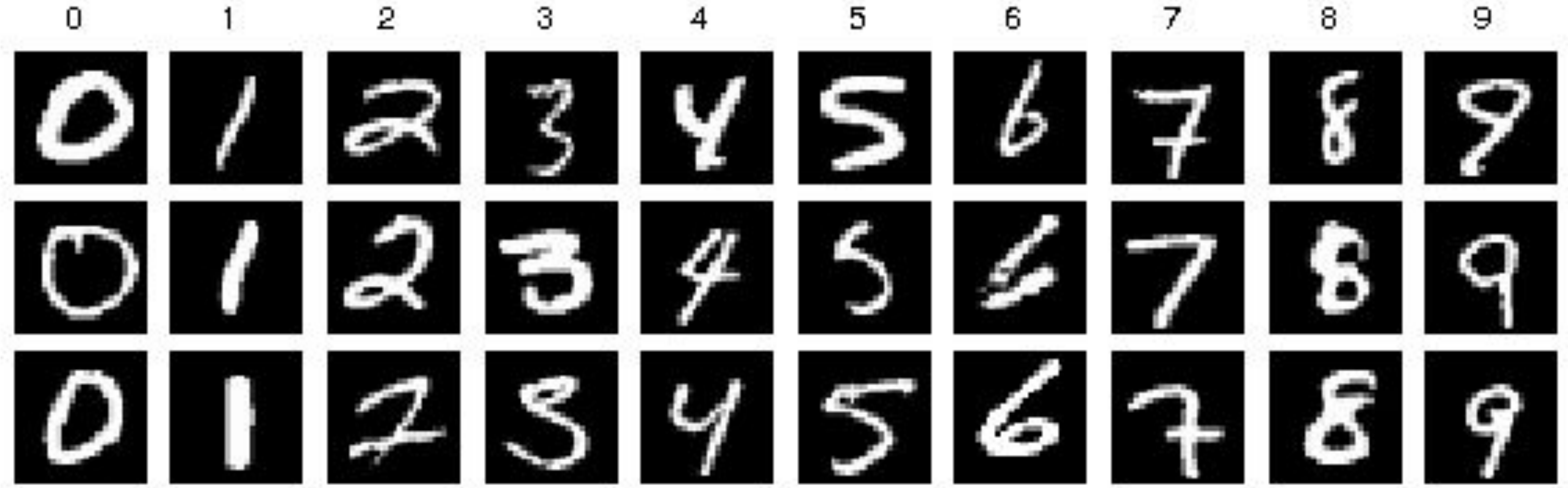
qui minimise l'erreur *f(t)*.



Si plus de caractéristiques $C = (x_1, x_2, x_3, \ldots)$ *alors* $\vec{n} = (a_1, a_2, a_3, \ldots)$ *: pareil.*

3- Réseaux de Neurones

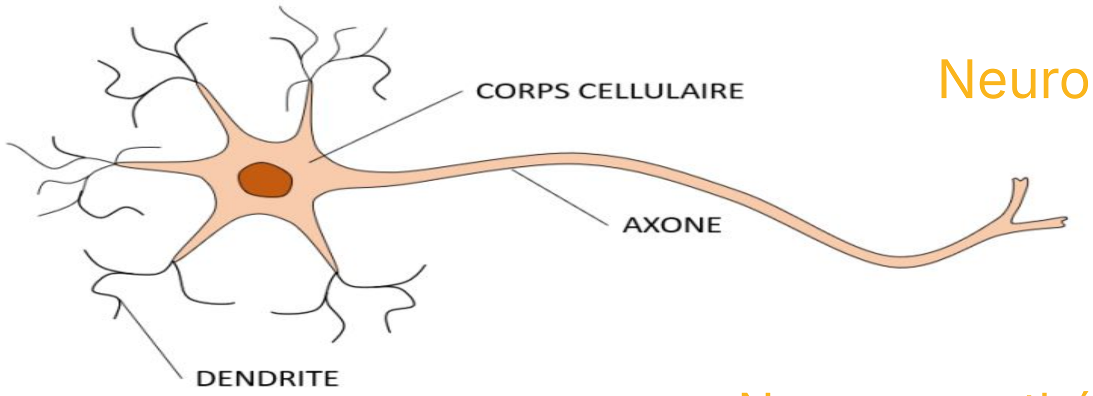
Challenge : classer les 10 chiffres



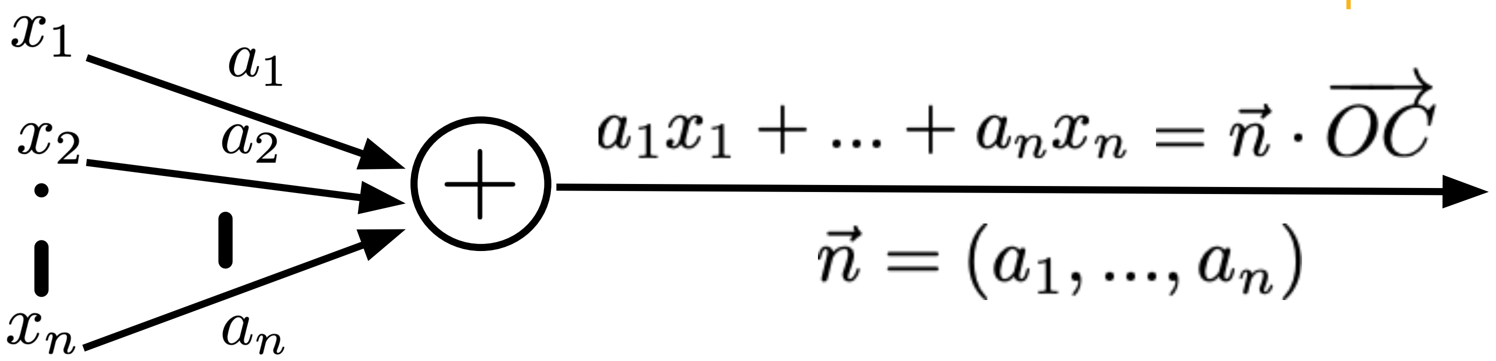
Il faut plus de caractéristiques $C = (x_1, x_2, ..., x_n)$ pour différentier ces classes.

Neurone Mathématique

Neurone biologique



Neurone mathématique



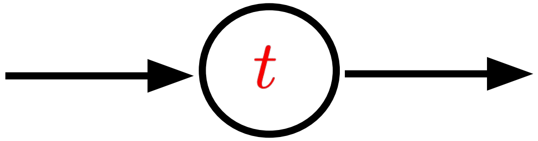
$$v = \max(\vec{n} \cdot \overrightarrow{OC} + b\,,\ 0)$$

paramètres du neurone: $t = (a_1, a_2, \ldots, a_n, b)$

$$t = (\vec{n}, b)$$

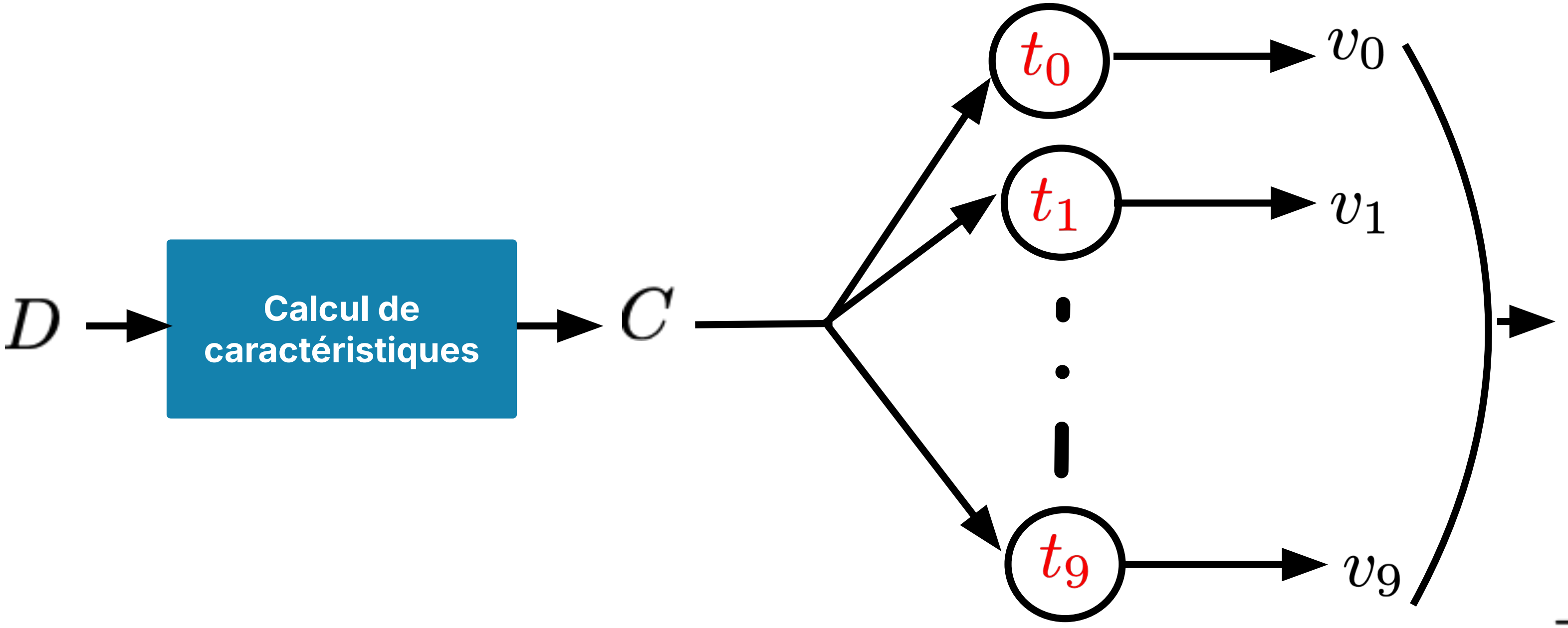
Estimation avec un réseau à 1 couche: perceptron

Un neurone :



Frank Rosenblatt (1957)

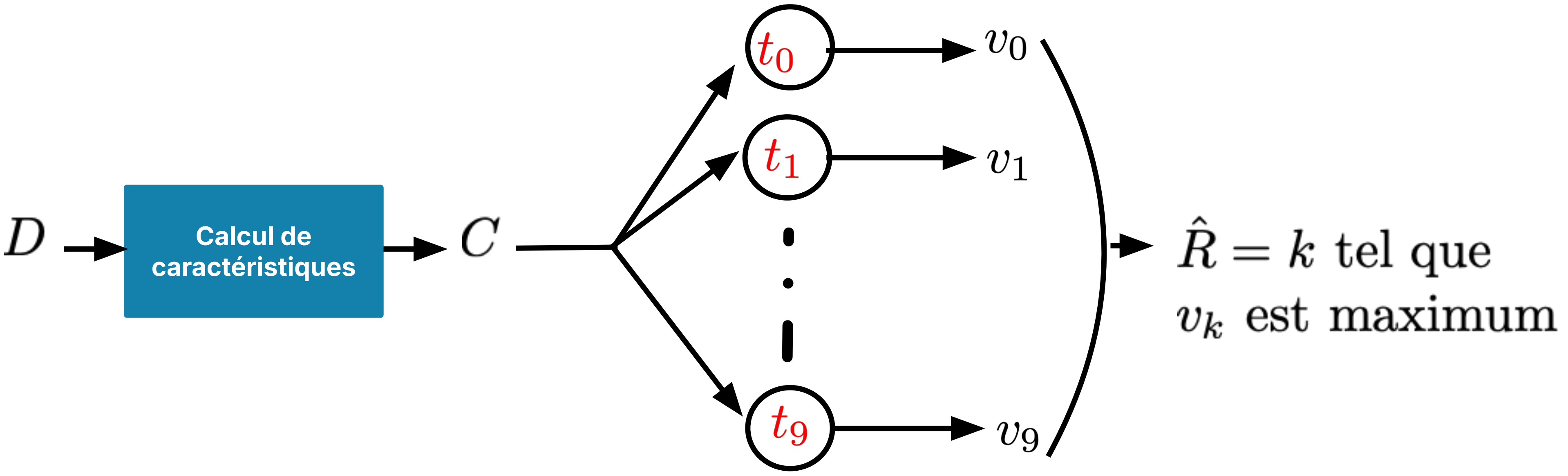
Estimation



$\hat{R} = k$ tel que v_k est maximum

$$t_k = (\vec{n}_k \, , b_k) \qquad v_k = \vec{n}_k \cdot \overrightarrow{OC} + b_k$$

Apprentissage des paramètres



Perceptron : ajuste itérativement les $t_k = (\vec{n}_k, b_k)$ avec une règle Hebb

Pour chaque caractéristique C d'une donnée d'entrainement

si $\hat{R} \neq R$ *(erreur)* alors

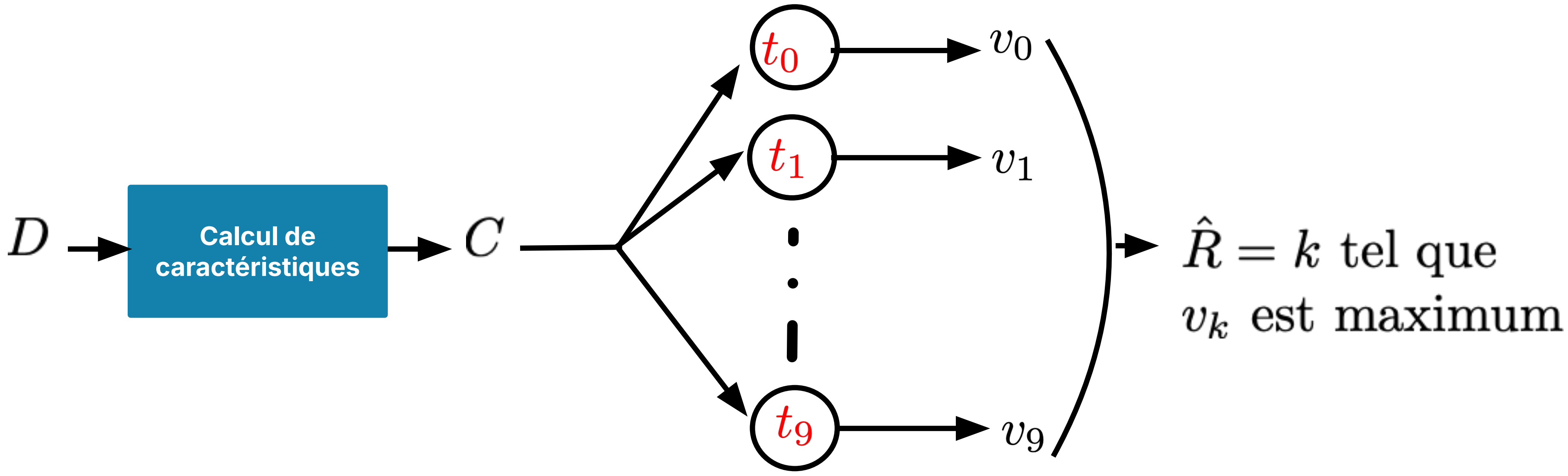
inhibition

pour $k = \hat{R}$ *(mauvaise classe)* $\vec{n}_k \to \vec{n}_k - \epsilon \overrightarrow{OC}$ et $b_k \to b_k - \epsilon$

pour $k = R$ *(bonne classe)* $\vec{n}_k \to \vec{n}_k + \epsilon \overrightarrow{OC}$ et $b_k \to b_k + \epsilon$

excitation

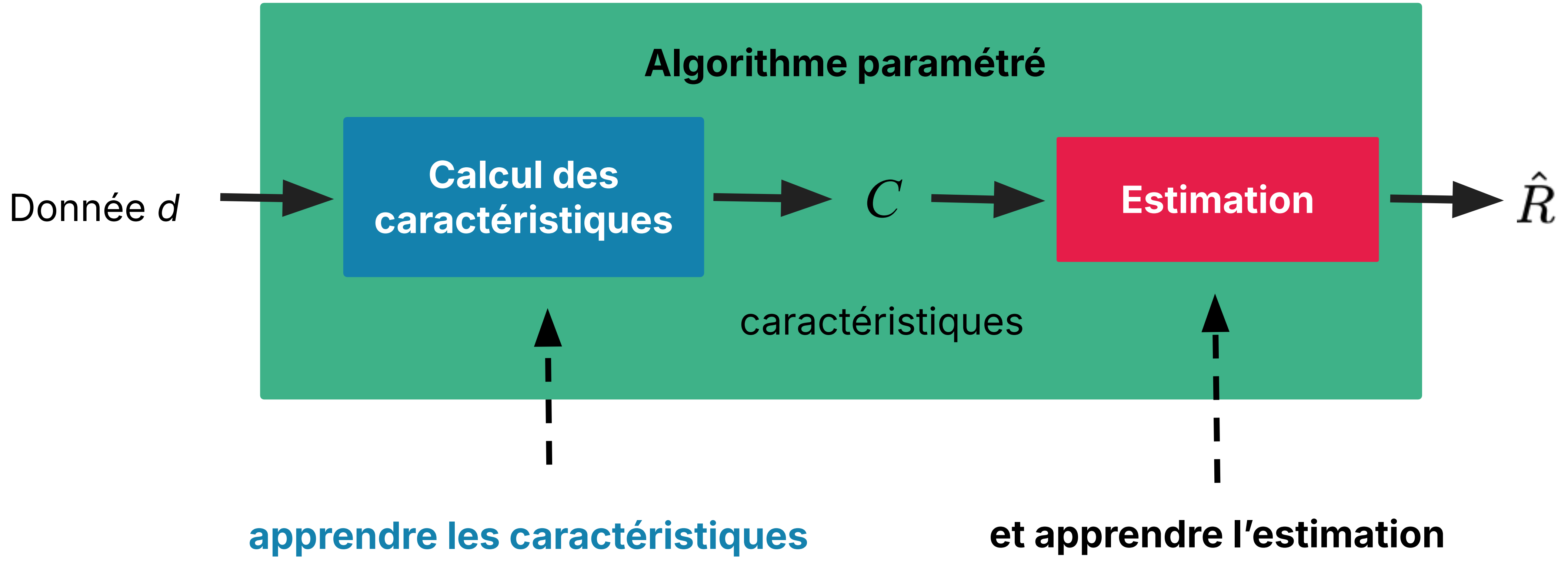
Apprentissage des paramètres



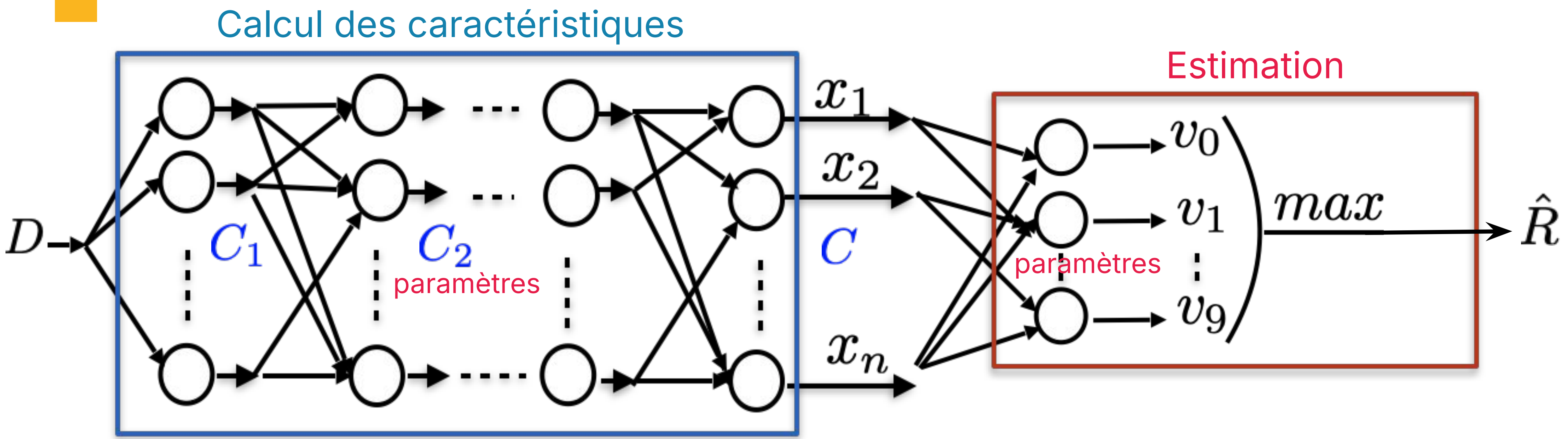
Perceptron : ajuste itérativement les $t_k = (\vec{n}_k, b_k)$ avec une règle Hebb

Descente de dérivée/gradient : sur l'erreur $f(t_0, t_1, ..., t_9)$

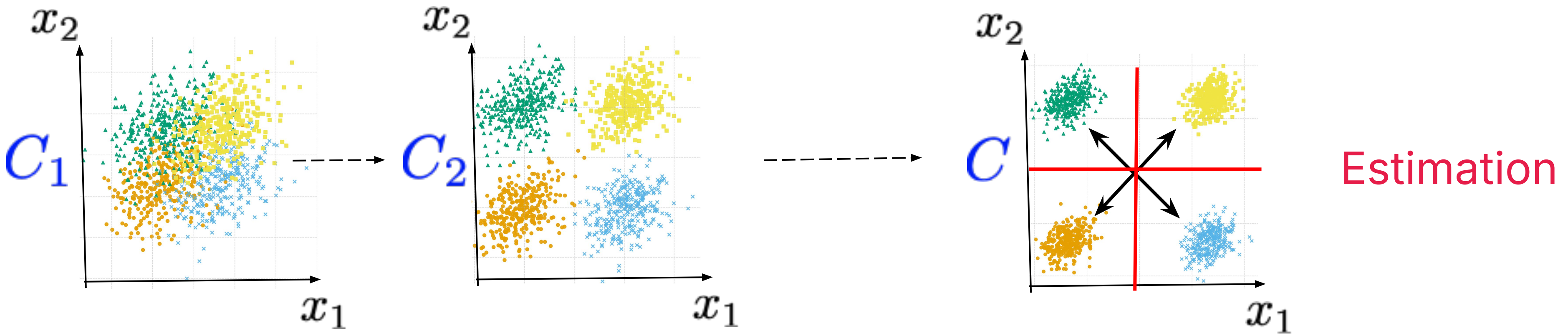
Apprentissage des caractéristiques



Apprendre les caractéristiques avec un réseau profond



Apprentissage des paramètres t par descente de dérivées/gradient pour minimiser l'erreur $f(t)$



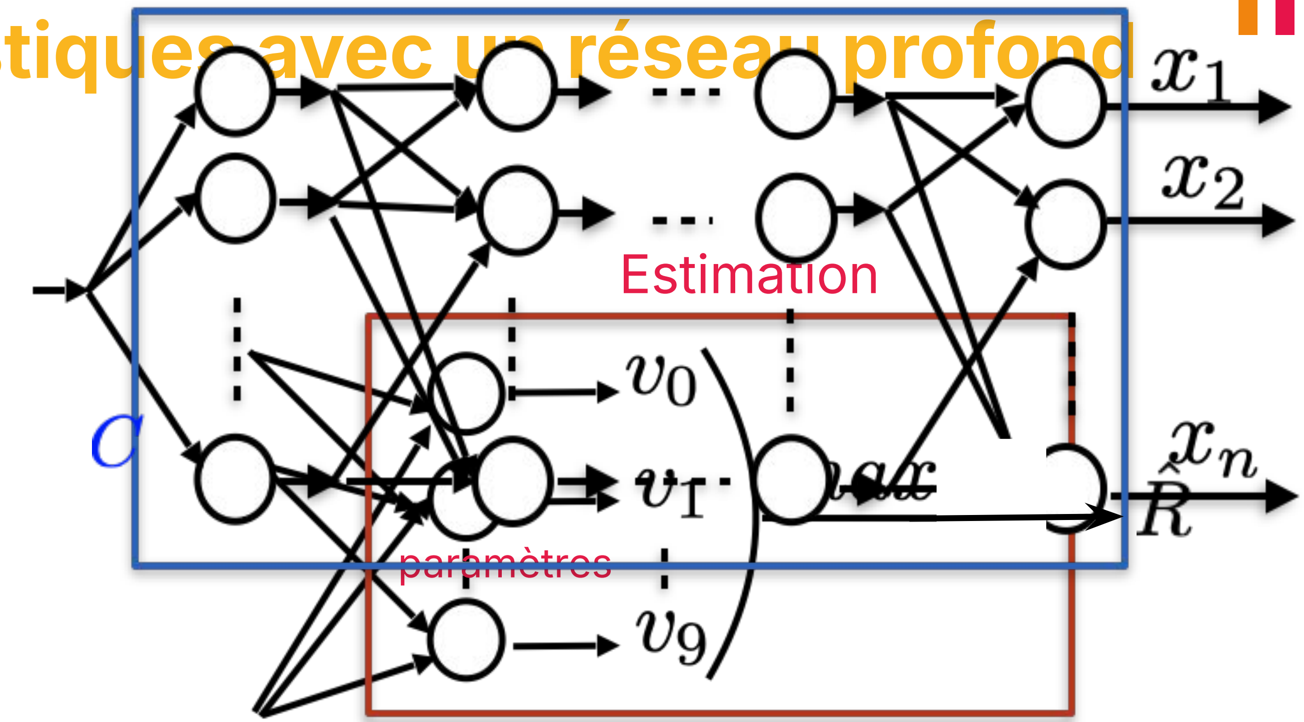
Après l'apprentissage le réseau sépare progressivement les caractéristiques: ***transport***.

Apprendre les caractéristiques avec un réseau profond

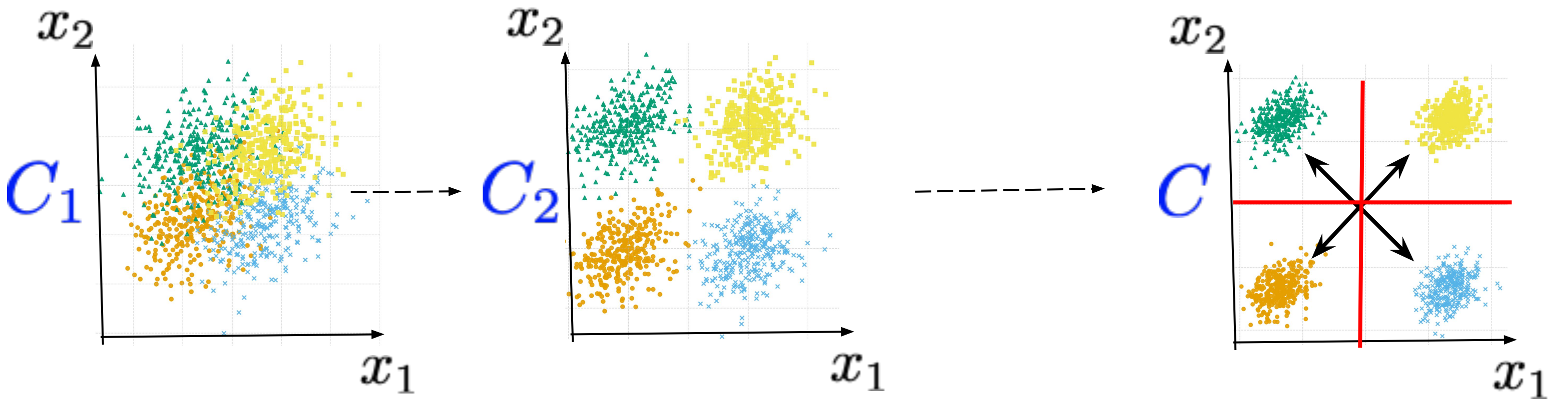
Calcul des caractéristiques

D C_1 C_2 C

paramètres



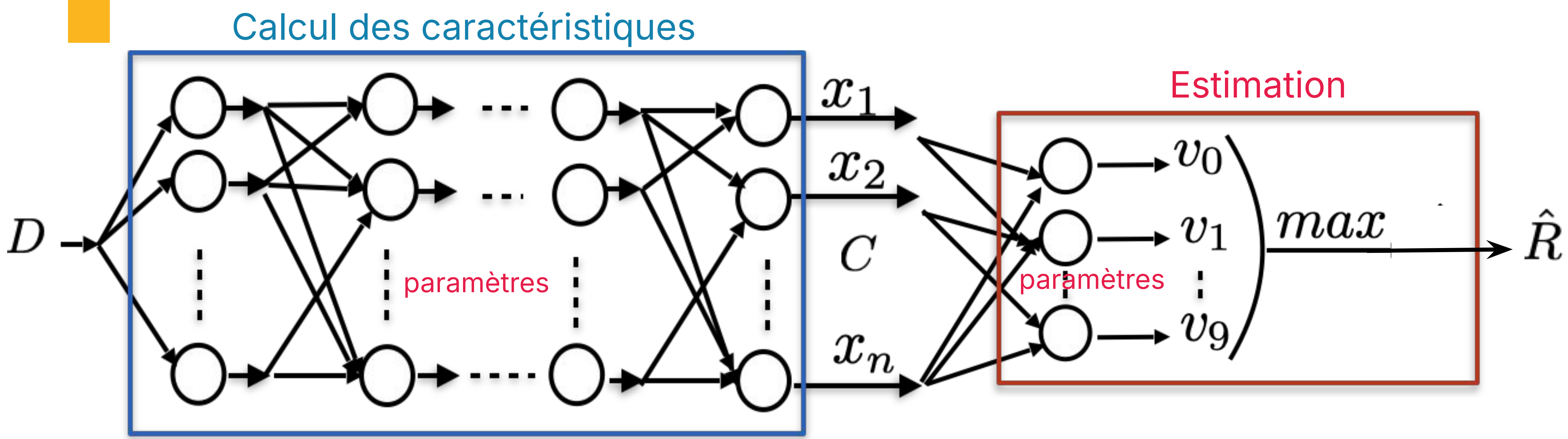
Apprentissage des paramètres *t* par descente de dérivées/gradient pour minimiser l'erreur *f(t)*



Estimation

Après l'apprentissage le réseau sépare progressivement les caractéristiques: ***transport*.**

Apprendre la réponse optimale avec un réseau profond



Énormément de paramètres donc énormément d'exemples pour généraliser.

Bonnes approximations de l'estimateur optimal de Bayes:

$$\hat{R} \text{ maximise } \hat{P}rob(\hat{R} = R|C) \approx Prob(\hat{R} = R|D)$$

Comment le réseau a-t-il évité la malédiction de la grande dimension ?

Grands modèles de langage

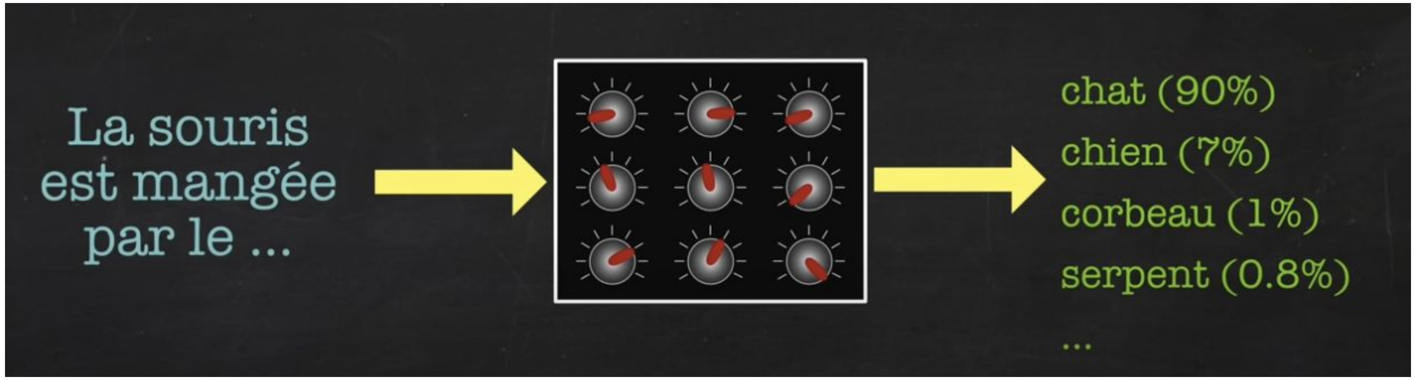
Modèle de Langage : réseau de neurone qui calcule itérativement le prochain mot $\hat{R}$ étant connu le prompt et les mots précédents (donnée *D*):

Transformer :
Mécanisme d'attention

x x x x x x x x x x x x x x x x x x x x x x x x x **?**

$\overset{\longleftrightarrow}{?}$ $\overset{\longleftrightarrow}{?}$ $\overset{\longleftrightarrow}{C}$ $\hat{R}$

Calcule $\hat{R}$ qui maximise $\hat{P}rob(\hat{R} = R|C)$ où C est un context.



→ peut faire mieux

Optimal: $\hat{P}rob(\hat{R} = R|C) \approx Prob(\hat{R} = R|D)$ si on a assez de context

4- Enseigner les maths avec des challenges d’IA

MathAData : Enseigner les maths avec des Challenges d'IA

La résolution d'un challenge d'IA met en jeux toutes les mathématiques du Lycée et Collège: géométrie, analyse, algèbre, statistiques, probabilités,...

Motiver et donner du sens aux mathématiques

à partir de challenges d'IA

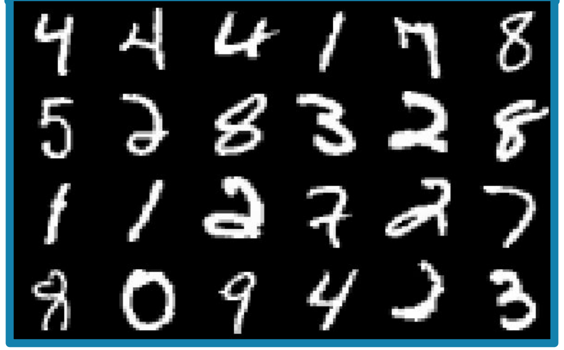
Comprendre les maths en manipulant et expérimentant

puis en pratiquant les maths sur des exercices.

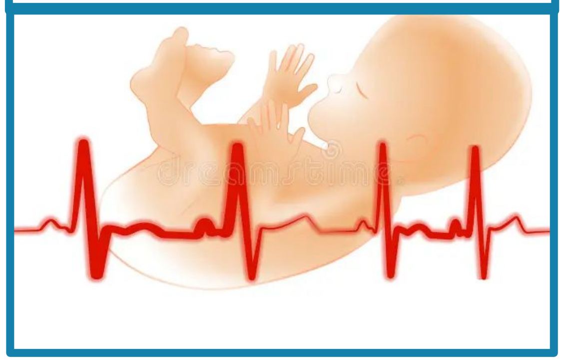
Des challenges d'IA pour les lycéens

Challenge : estimer la réponse *R* à une question à partir de données numériques *D*

Quel chiffre écrit dans l'image ?



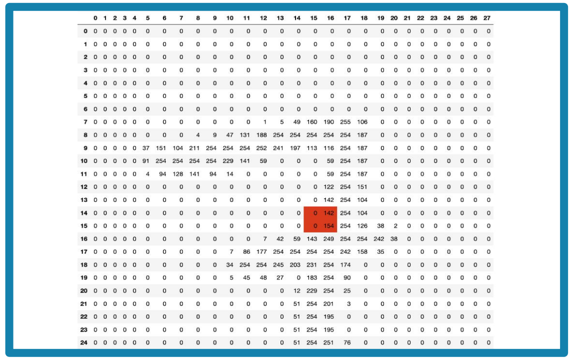
Le foetus est-il sain ou malade ?



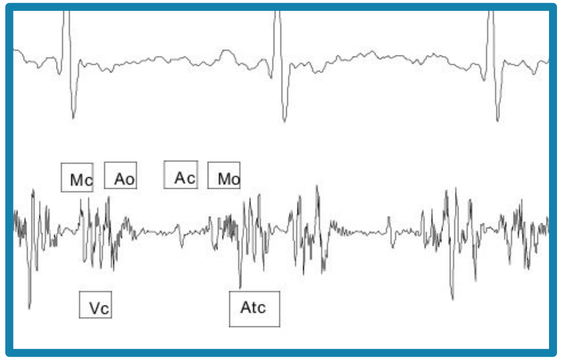
Corneille ou Molière a écrit ce texte ?

Décoder les chants des baleines

Données numérique ***d***



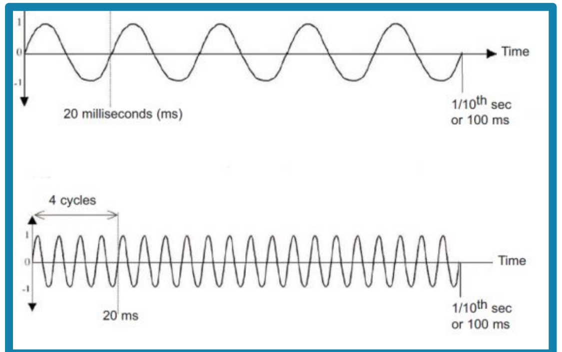
Valeurs des pixels de l'image



Électrocardiogramme.

Je n'aurais rien à craindre, si tout le monde vous voyait des yeux dont je vous vois, et je trouve en votre personne de quoi avoir raison aux choses que je fais pour vous. Mon coeur, pour sa défense, a tout votre mérite,

Mots du texte



Enregistrement du chant

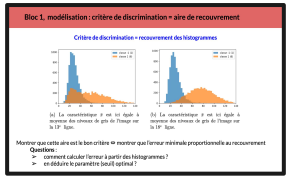
MathAData propose des séquences et outils pédagogiques pour enseigner les maths au programme avec des challenges d'IA.

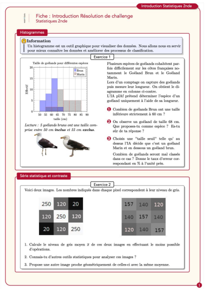
co-développés avec des professeurs de l'éducation nationale

ACADÉMIE DE LILLE
Liberté
Égalité
Fraternité

1. Cadre math et IA

DIAPO et FICHE INTRO





Intro du Challenge d'IA

Intro des concepts maths du programme

2. Pratique numérique

NOTEBOOK

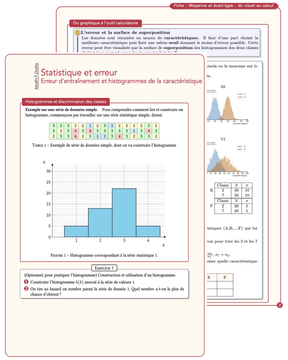
Résoudre le challenge

Manipuler les maths

Ludique

3. Pratiquer les maths

FICHES EXOS PAPIER



Exercices liés au challenge

Exercices du programme

mathadata.fr