



INSTITUT  
de MATHÉMATIQUES  
de TOULOUSE

# Introduction à l'apprentissage machine

**Laurent Risser**

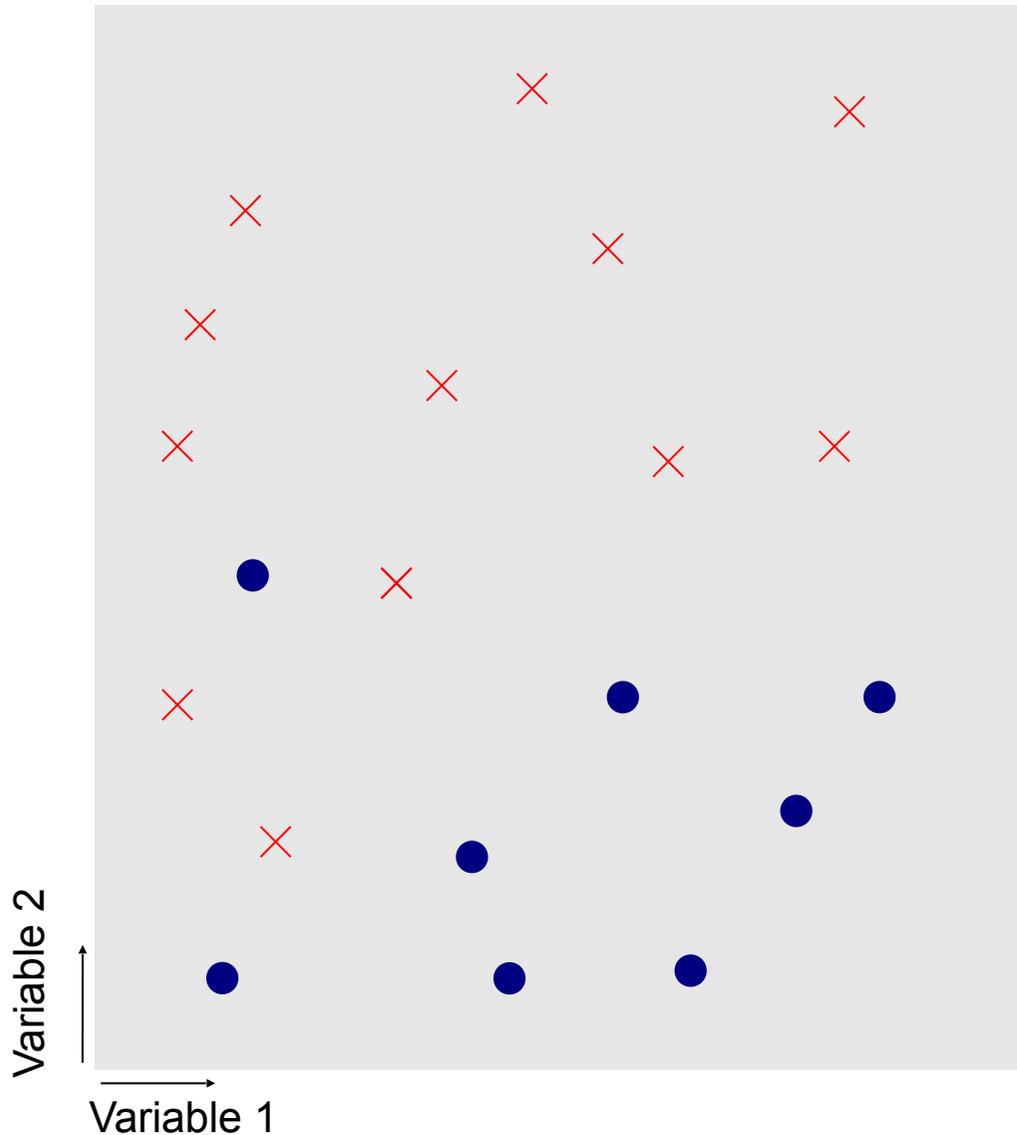
Ingénieur de Recherche à l'Institut de Mathématiques de Toulouse

[lrissier@math.univ-toulouse.fr](mailto:lrissier@math.univ-toulouse.fr)

# Exemples introductifs à l'apprentissage machine

# 0.a) Exemples introductifs — Apprentissage supervisé

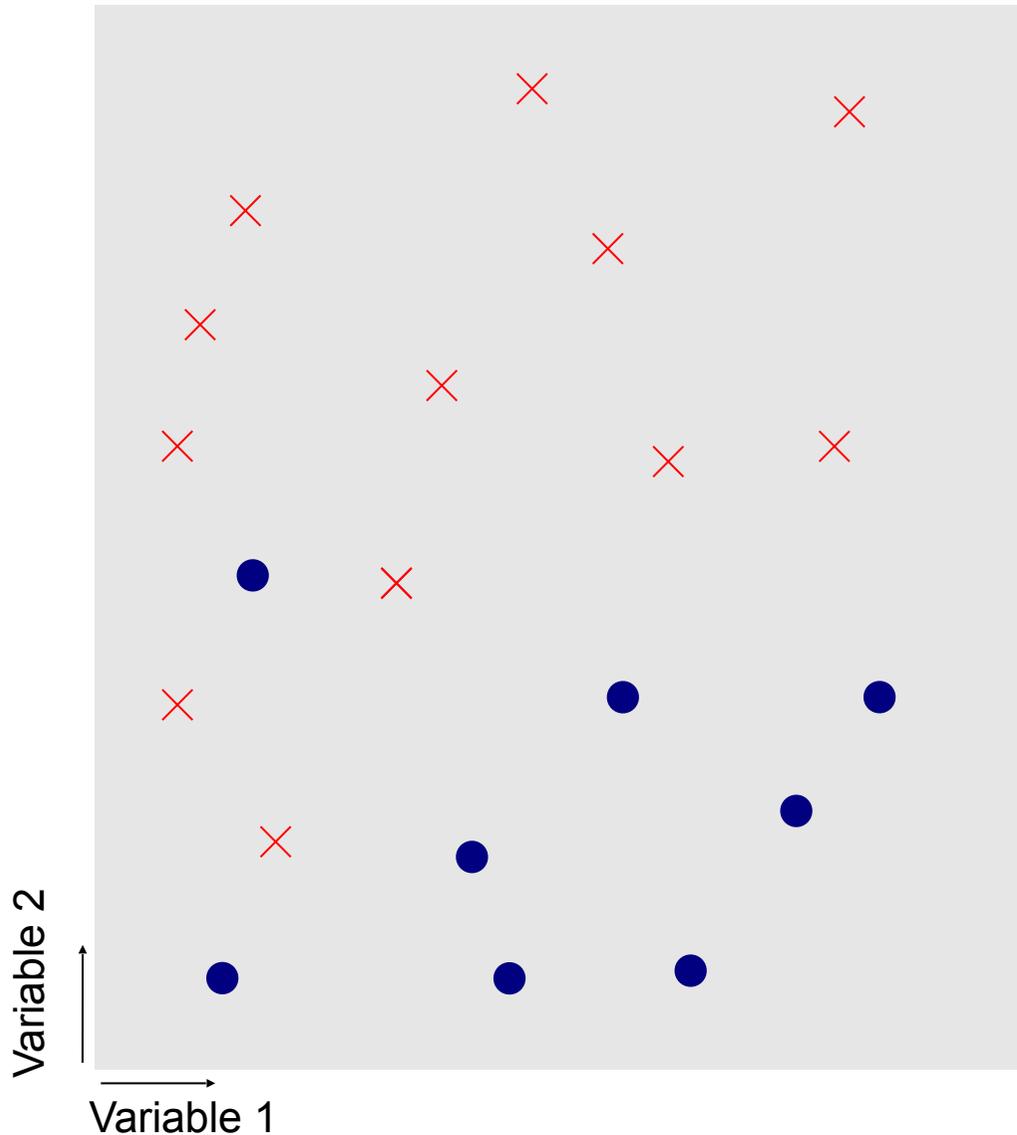
## Exemple introductif 1 : Apprentissage supervisé — *Jeu de données*



$n = 20$  observations  
 $p = 2$  variables (dimension)  
**Label à 2 états**

## 0.a) Exemples introductifs — Apprentissage supervisé

### Exemple introductif 1 : Apprentissage supervisé — *Jeu de données*

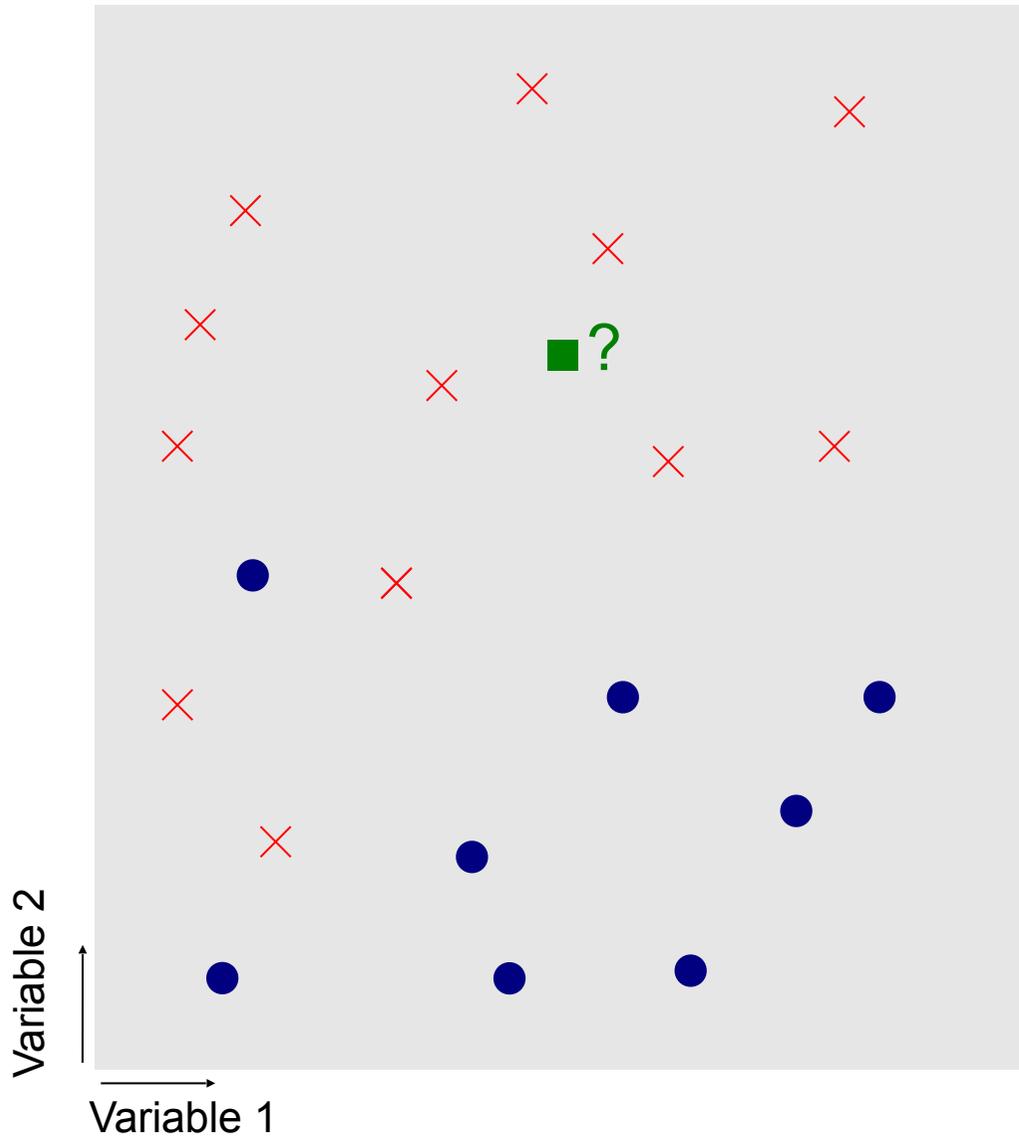


$n = 20$  observations  
 $p = 2$  variables (dimension)  
**Label à 2 états**

Exemple :  
Variable 1 = Age  
Variable 2 = Revenu annuel  
Etat = Achat d'un type de produit

# 0.a) Exemples introductifs — Apprentissage supervisé

## Exemple introductif 1 : Apprentissage supervisé — *Prédiction*

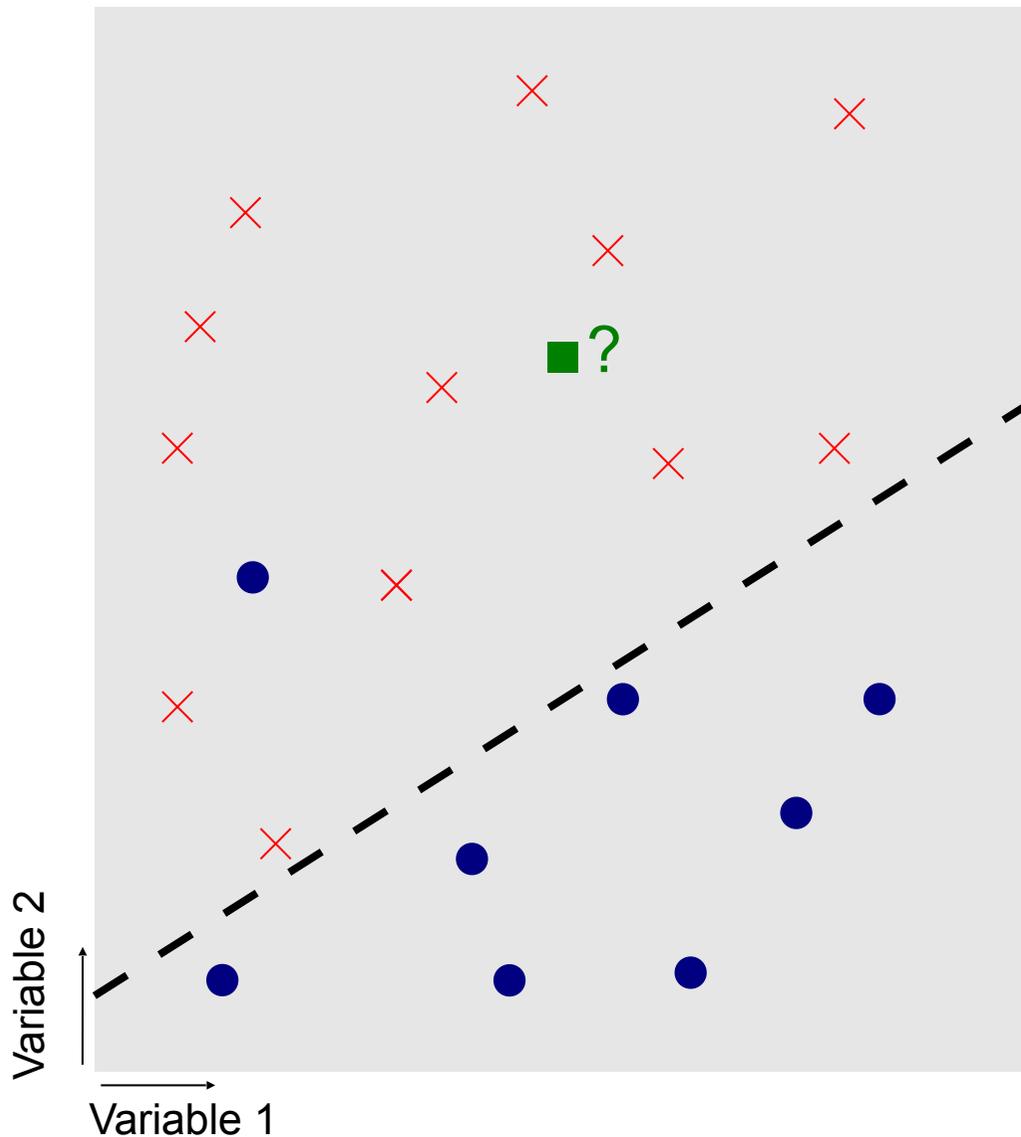


$n = 20$  observations  
 $p = 2$  variables (dimension)  
**Label à 2 états**

Etat le plus vraisemblable de ? ■

# 0.a) Exemples introductifs — Apprentissage supervisé

## Exemple introductif 1 : Apprentissage supervisé — *Apprentissage*



$n = 20$  observations  
 $p = 2$  variables (dimension)  
**Label à 2 états**

Etat le plus vraisemblable de ? ■

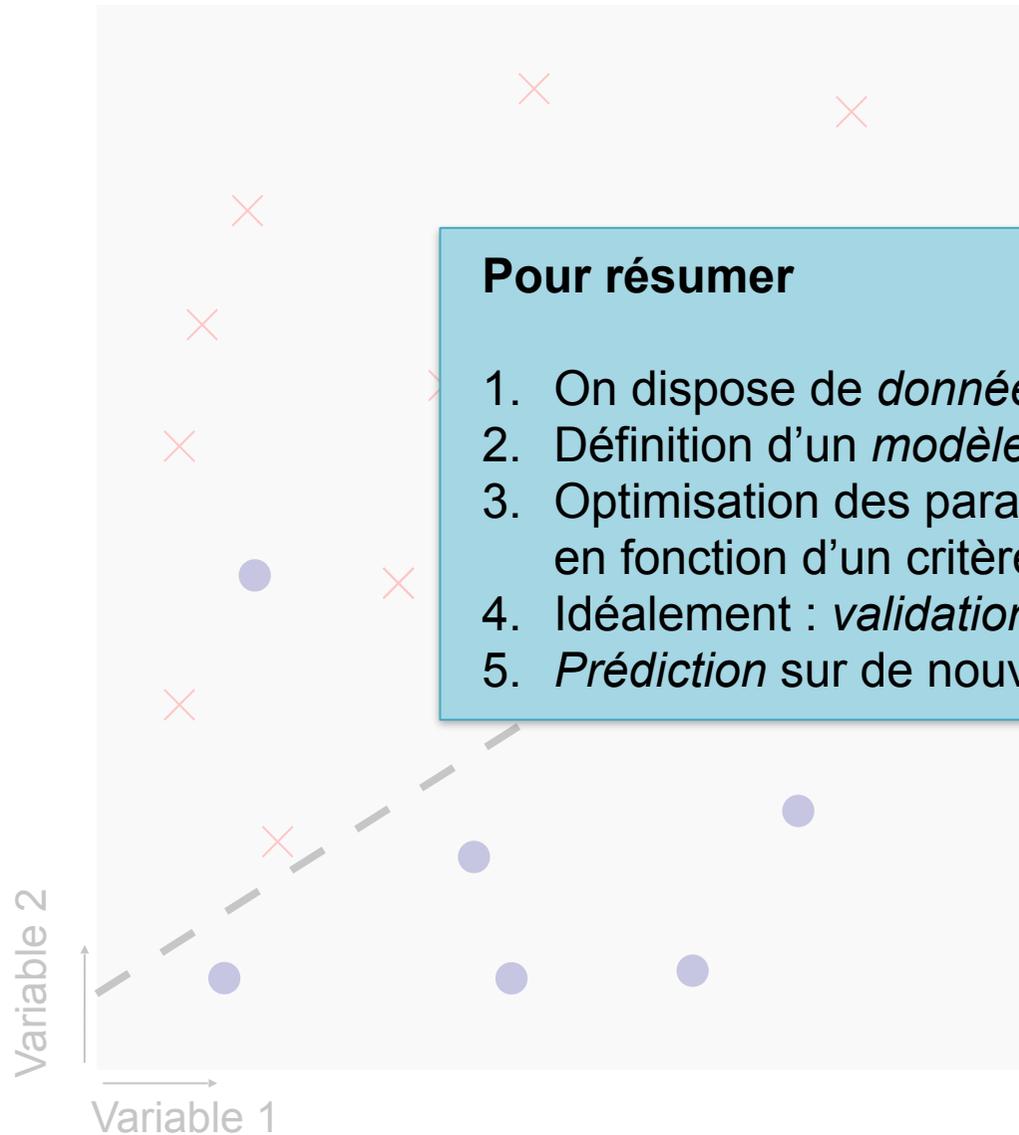
→ Apprentissage (ici par un modèle linéaire)  
puis décision/estimation

# 0.a) Exemples introductifs — Apprentissage supervisé

## Exemple introductif 1 : Apprentissage supervisé — *Apprentissage*

### Pour résumer

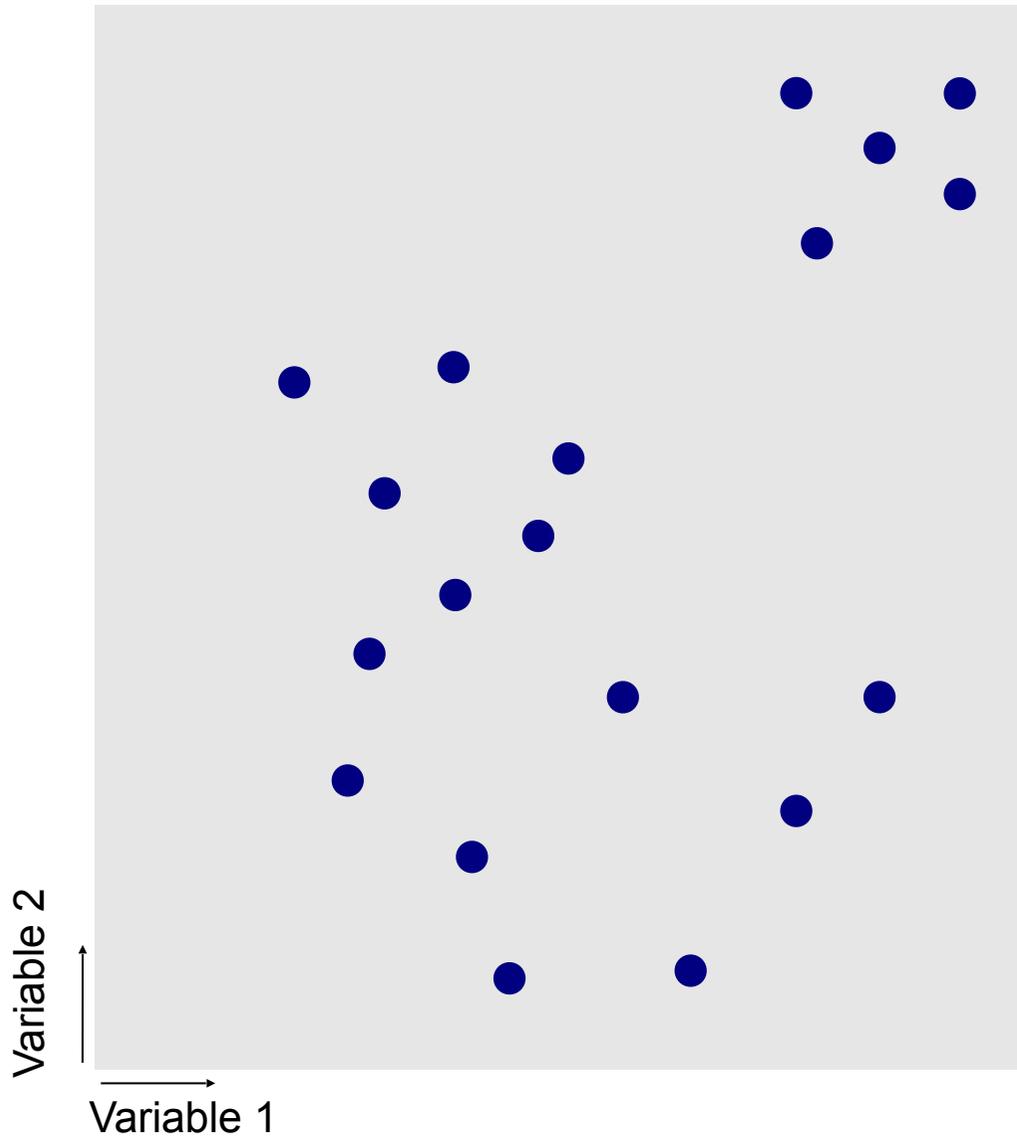
1. On dispose de *données d'apprentissage annotées*
2. Définition d'un *modèle* pour séparer les données
3. Optimisation des paramètres du modèle (*apprentissage*) en fonction d'un critère d'**erreur de prédiction**
4. Idéalement : *validation* du modèle sur données test
5. *Prédiction* sur de nouvelles observation



Etat le plus vraisemblable de ? ■

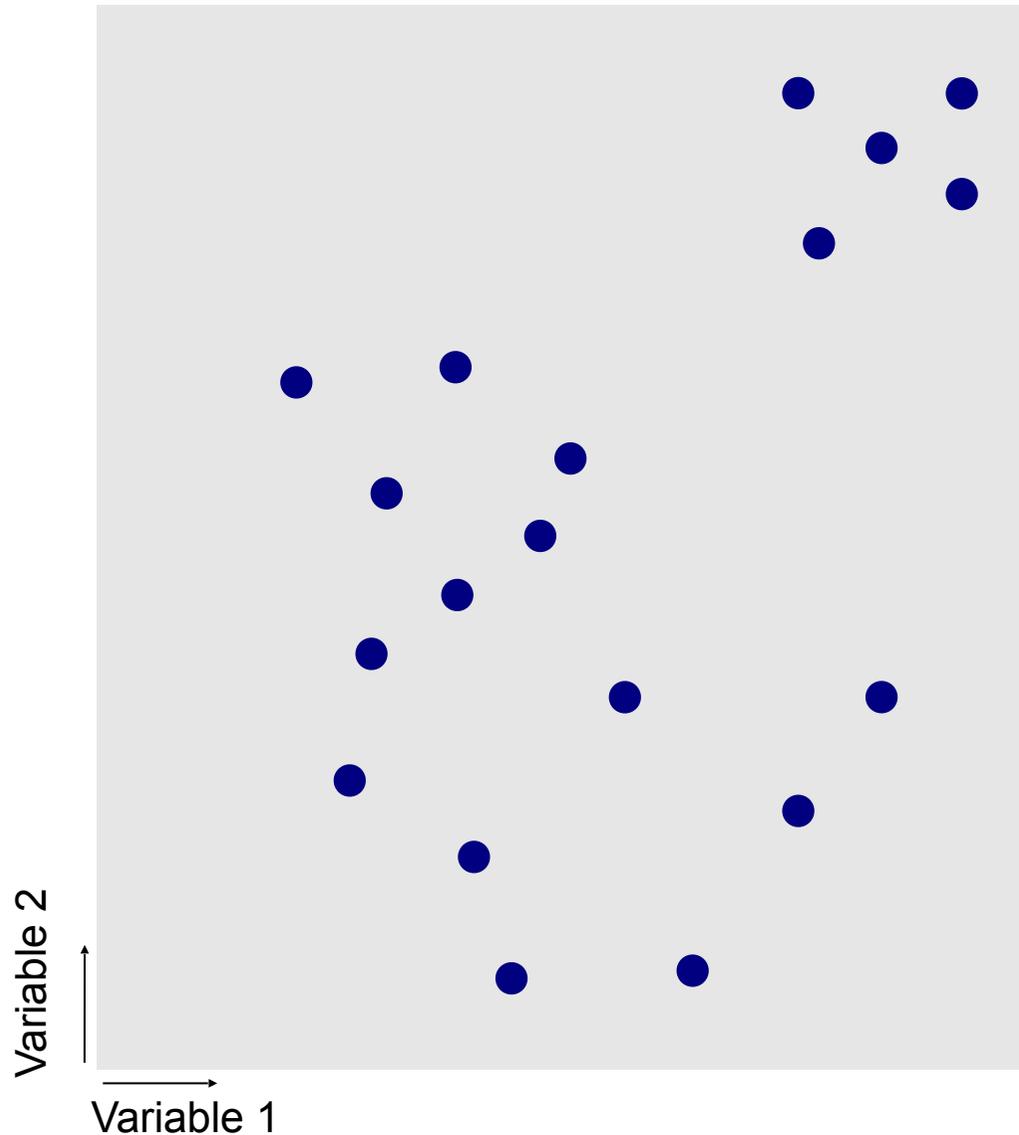
→ Apprentissage par un modèle linéaire puis décision/estimation

### Exemple introductif 2 : Apprentissage non-supervisé — *Jeu de données*



$n = 20$  observations  
 $p = 2$  variables (dimension)  
**Pas de label**

### Exemple introductif 2 : Apprentissage non-supervisé — *Jeu de données*

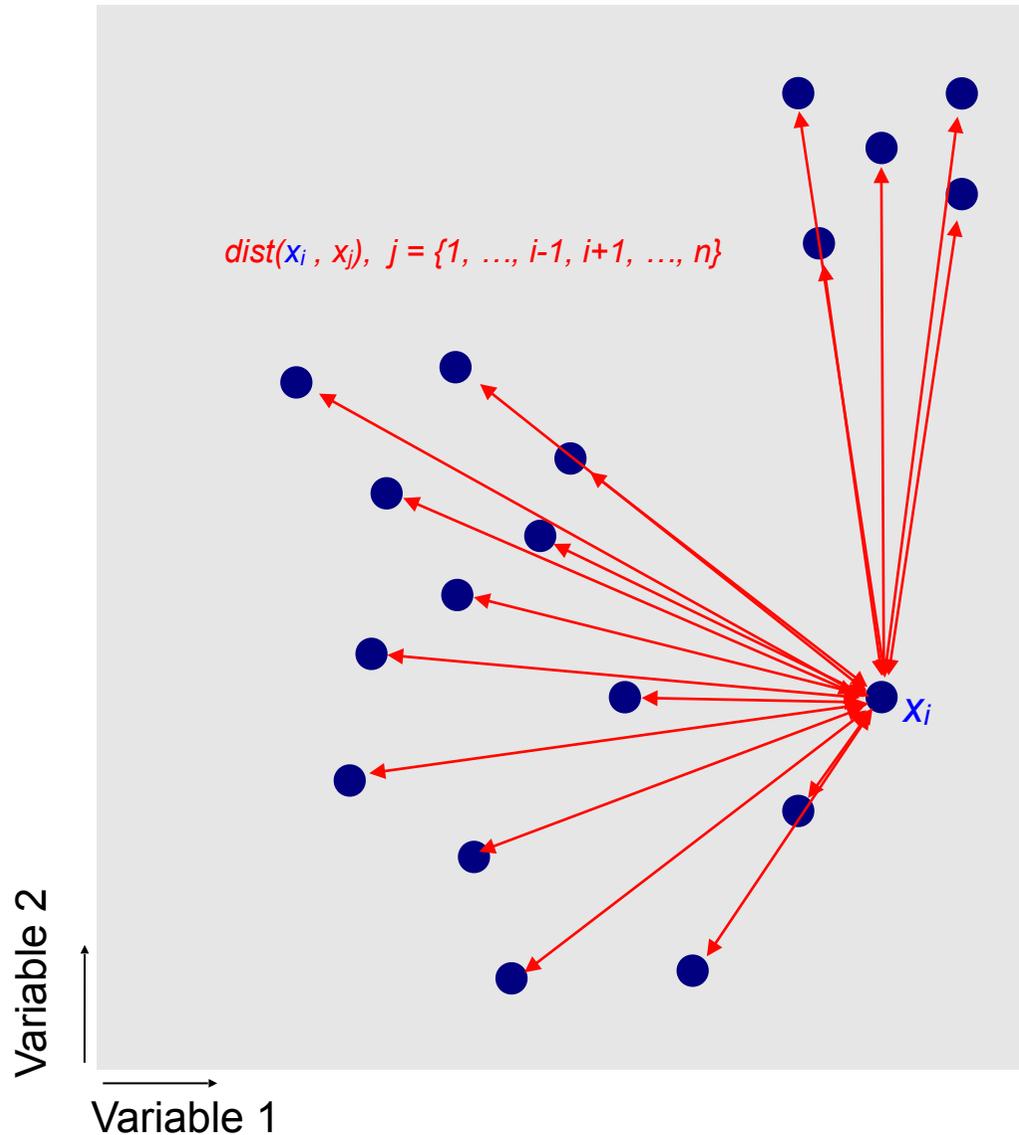


$n = 20$  observations  
 $p = 2$  variables (dimension)  
**Pas de label**

Peut-on raisonnablement distinguer plusieurs sous-groupes d'individus ?

## 0.b) Exemples introductifs — Apprentissage non-supervisé

### Exemple introductif 2 : Apprentissage non-supervisé — *Distance entre les observations*



$n = 20$  observations  
 $p = 2$  variables (dimension)  
**Pas de label**

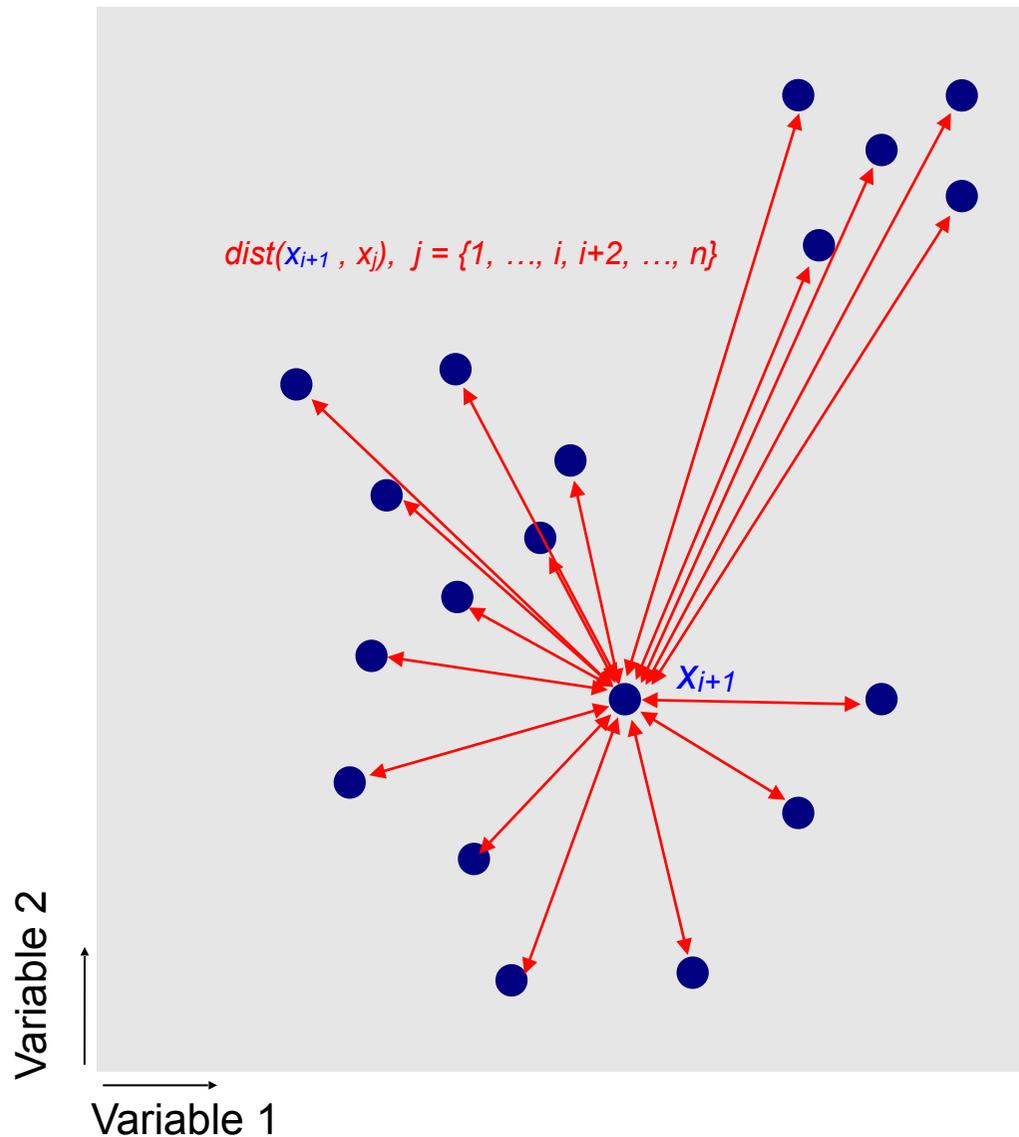
Peut-on raisonnablement distinguer plusieurs sous-groupes d'individus ?

Distance entre les observations

→  $dist(x_i, x_j), i, j = \{1, \dots, n\}^2$

## 0.b) Exemples introductifs — Apprentissage non-supervisé

### Exemple introductif 2 : Apprentissage non-supervisé — *Distance entre les observations*



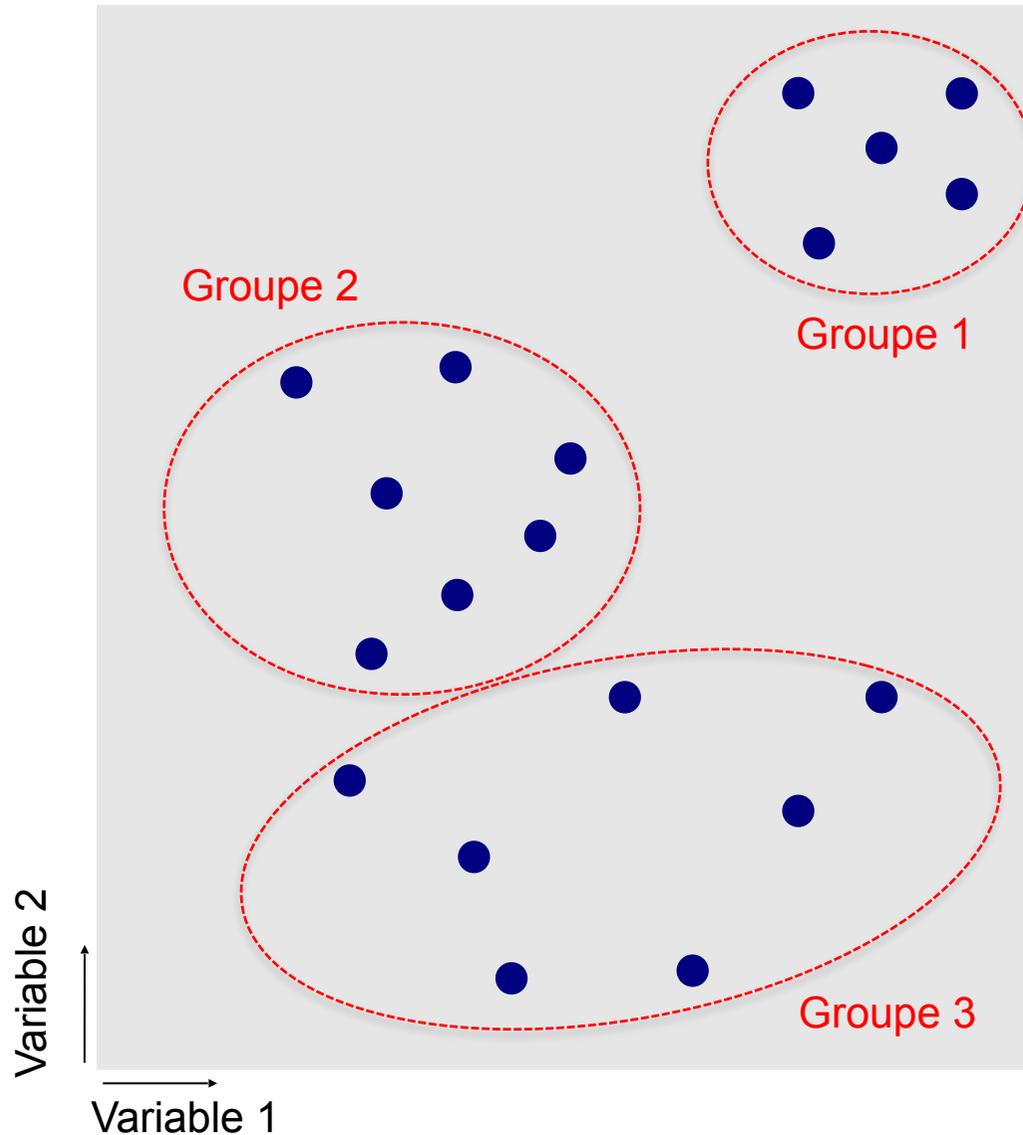
$n = 20$  observations  
 $p = 2$  variables (dimension)  
**Pas de label**

Peut-on raisonnablement distinguer plusieurs sous-groupes d'individus ?

Distance entre les observations

→  $dist(x_i, x_j), i, j = \{1, \dots, n\}^2$

## Exemple introductif 2 : Apprentissage non-supervisé — *Apprentissage*



$n = 20$  observations  
 $p = 2$  variables (dimension)  
**Pas de label**

Peut-on raisonnablement distinguer plusieurs sous-groupes d'individus ?

Distance entre les observations

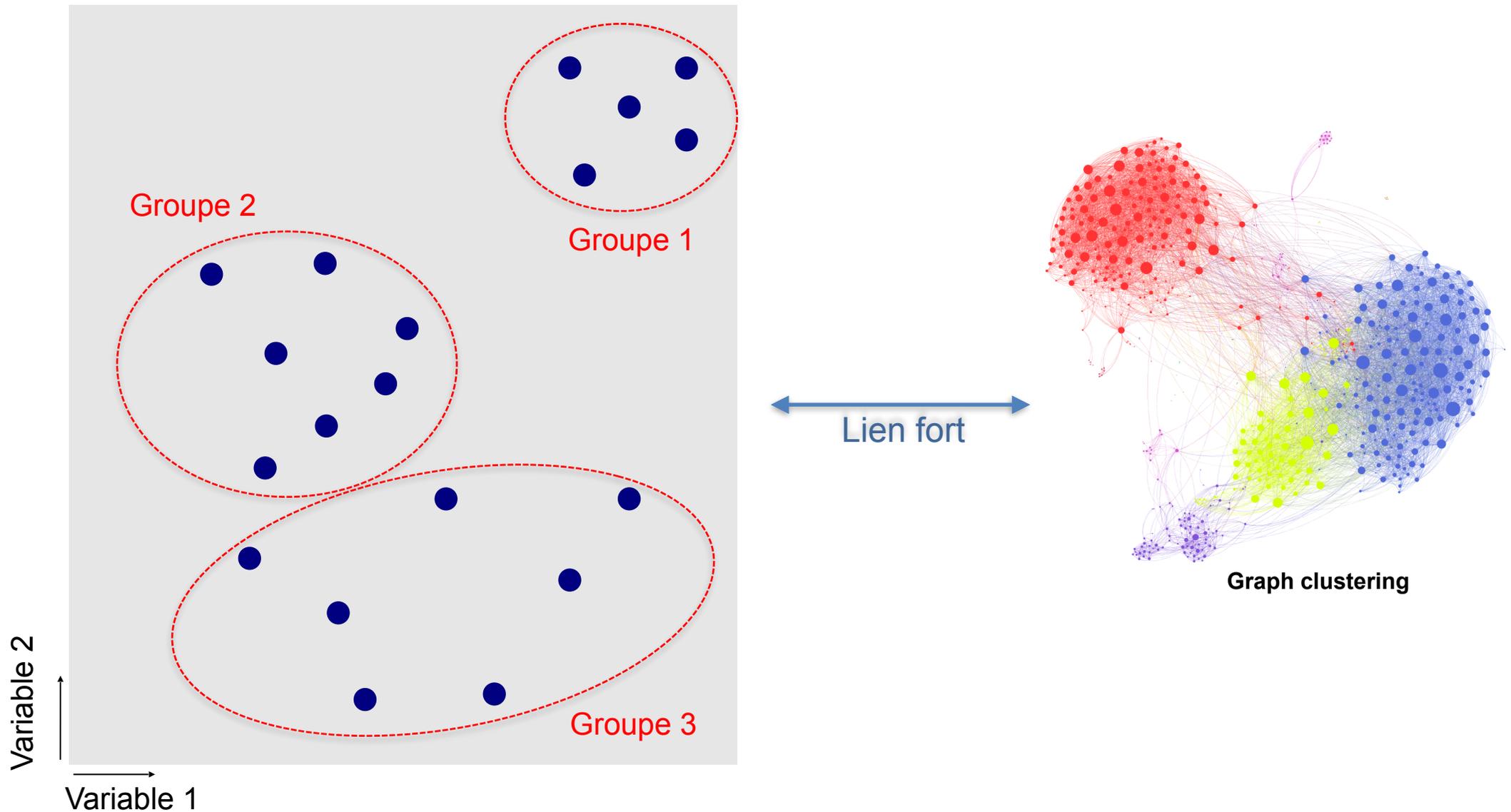
→  $dist(x_i, x_j), i, j = \{1, \dots, n\}^2$

Energie à minimiser en fonction de labels  $y_i$

→  $f(y_1, \dots, y_n, \{dist(x_i, x_j), i, j = \{1, \dots, n\}^2\})$

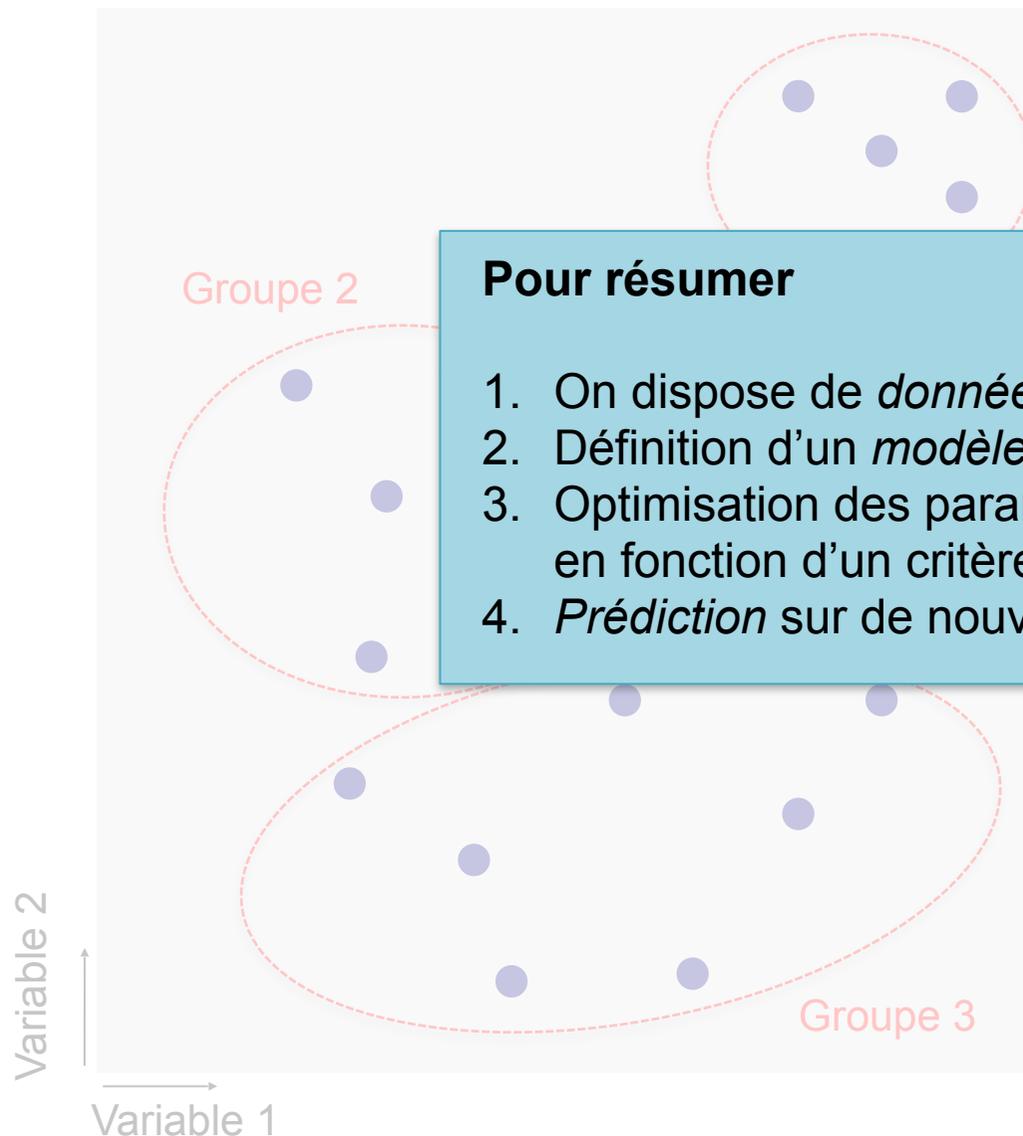
## 0.b) Exemples introductifs — Apprentissage non-supervisé

Exemple introductif 2 : Apprentissage non-supervisé — *Lien avec le clustering de graphe*



## 0.b) Exemples introductifs — Apprentissage non-supervisé

### Exemple introductif 2 : Apprentissage non-supervisé — *Apprentissage*



Peut-on raisonnablement distinguer plusieurs sous-groupes d'individus ?

→ Apprentissage en fonction d'une notion de distance entre les individus

## Plan de la présentation

- Exemples introductifs
  - Apprentissage supervisé
  - Apprentissage non-supervisé
- Evolution des tendances en science des données
- Algorithmes classiques en apprentissage machine
  - K-means
  - Arbres de décision et Random forests
  - SVM
  - Régression logistique
- Sur-apprentissage et validation croisée
  - Sur-apprentissage
  - Validation croisée
- Grande dimension et régularisation
  - Modélisation
  - Effet de la régularisation
- Réduction de dimension par ACP
- Apprentissage machine et GPU
  - Introduction
  - Réseaux de neurones profonds (deep learning)
  - Calcul GPU en deep learning
- Conclusion

# De la statistique classique à l'apprentissage machine

# 1) De la statistique à l'apprentissage machine

Historique<sup>1</sup>:

**1940-70** : *Statistiques classiques*. Question associée à une hypothèse expérimentalement réfutable avec  $n \approx 30$  observations et  $p < 10$  variables.

**1970s** : Généralisation des premiers outils informatiques. L'analyse de données explore des données plus volumineuses.

**1980s** : Les systèmes experts sont supplantés par l'apprentissage machine et les réseaux de neurones.

**1990s** : *1<sup>er</sup> changement de paradigme* : Les données ne sont plus planifiées mais sont préalablement acquises : *From Data Mining to Knowledge Discovery*.

**2000s** : *2<sup>eme</sup> changement de paradigme* : Le nombre de variables  $p$  explose, notamment avec les données omiques où  $p \gg n$ . La qualité de prévision devient plus importante que la réalité du modèle devenu *boîte noire*. Problématique du fléau de la dimension.

**2010s** : *3<sup>eme</sup> changement de paradigme* : Le nombre d'observations  $n$  explose dans le e-commerce, la géo-localisation, ... Bases de données structurées en *cloud* et moyens de calculs regroupés en *clusters* (*big data*). La notion de rapidité des algorithmes devient critique.

... explicabilité et interprétabilité des décisions des algorithmes

... données complexes (small data)

<sup>1</sup>Cours Apprentissage Statistique de P. Besse (INSA Toulouse): <http://www.math.univ-toulouse.fr/~besse/enseignement.html>

# 1) De la statistique à l'apprentissage machine

Historique<sup>1</sup>:

**1940-70** : *Statistiques classiques*. Question associée à une hypothèse expérimentalement réfutable avec  $n \approx 30$  observations et  $p < 10$  variables.

**1970s** : Généralisation des premiers outils informatiques. L'analyse de données explore des données plus volumineuses.

**1980s** : Les systèmes experts sont supplantés par l'apprentissage machine et les réseaux de neurones.

**1990s** : *1<sup>er</sup> changement de paradigme* : Les données ne sont plus planifiées mais sont préalablement acquises : *From Data Mining to Knowledge Discovery*.

**2000s** : *2<sup>eme</sup> changement de paradigme* : Le nombre de variables  $p$  explose, notamment avec les données omiques où  $p \gg n$ . La qualité de prévision devient plus importante que la réalité du modèle devenu *boîte noire*. Problématique du fléau de la dimension.

**2010s** : *3<sup>eme</sup> changement de paradigme* : Le nombre d'observations  $n$  explose dans le e-commerce, la géo-localisation, ... Bases de données structurées en *cloud* et moyens de calculs regroupés en *clusters* (*big data*). La notion de rapidité des algorithmes devient critique.

... explicabilité et interprétabilité des décisions des algorithmes

... données complexes (small data)

<sup>1</sup>Cours Apprentissage Statistique de P. Besse (INSA Toulouse): <http://www.math.univ-toulouse.fr/~besse/enseignement.html>

# 1) De la statistique à l'apprentissage machine

Historique<sup>1</sup>:

**1940-70** : *Statistiques classiques*. Question associée à une hypothèse expérimentalement réfutable avec  $n \approx 30$  observations et  $p < 10$  variables.

**1970s** : Généralisation des premiers outils informatiques. L'analyse de données explore des données plus volumineuses.

**1980s** : Les systèmes experts sont supplantés par l'apprentissage machine et les réseaux de neurones.

**1990s** : *1<sup>er</sup> changement de paradigme* : Les données ne sont plus planifiées mais sont préalablement acquises : *From Data Mining to Knowledge Discovery*.

**2000s** : *2<sup>eme</sup> changement de paradigme* : Le nombre de variables  $p$  explose, notamment avec les données omiques où  $p \gg n$ . La qualité de prévision devient plus importante que la réalité du modèle devenu *boîte noire*. Problématique du fléau de la dimension.

**2010s** : *3<sup>eme</sup> changement de paradigme* : Le nombre d'observations  $n$  explose dans le e-commerce, la géo-localisation, ... Bases de données structurées en *cloud* et moyens de calculs regroupés en *clusters* (*big data*). La notion de rapidité des algorithmes devient critique.

... explicabilité et interprétabilité des décisions des algorithmes

... données complexes (small data)

<sup>1</sup>Cours Apprentissage Statistique de P. Besse (INSA Toulouse): <http://www.math.univ-toulouse.fr/~besse/enseignement.html>

# 1) De la statistique à l'apprentissage machine

Historique<sup>1</sup>:

**1940-70** : *Statistiques classiques*. Question associée à une hypothèse expérimentalement réfutable avec  $n \approx 30$  observations et  $p < 10$  variables.

**1970s** : Généralisation des premiers outils informatiques. L'analyse de données explore des données plus volumineuses.

**1980s** : Les systèmes experts sont supplantés par l'apprentissage machine et les réseaux de neurones.

**1990s** : *1<sup>er</sup> changement de paradigme* : Les données ne sont plus planifiées mais sont préalablement acquises : *From Data Mining to Knowledge Discovery*.

**2000s** : *2<sup>eme</sup> changement de paradigme* : Le nombre de variables  $p$  explose, notamment avec les données omiques où  $p \gg n$ . La qualité de prévision devient plus importante que la réalité du modèle devenu *boîte noire*. Problématique du fléau de la dimension.

**2010s** : *3<sup>eme</sup> changement de paradigme* : Le nombre d'observations  $n$  explose dans le e-commerce, la géo-localisation, ... . Bases de données structurées en *cloud* et moyens de calculs regroupés en *clusters* (*big data*). La notion de rapidité des algorithmes devient critique.

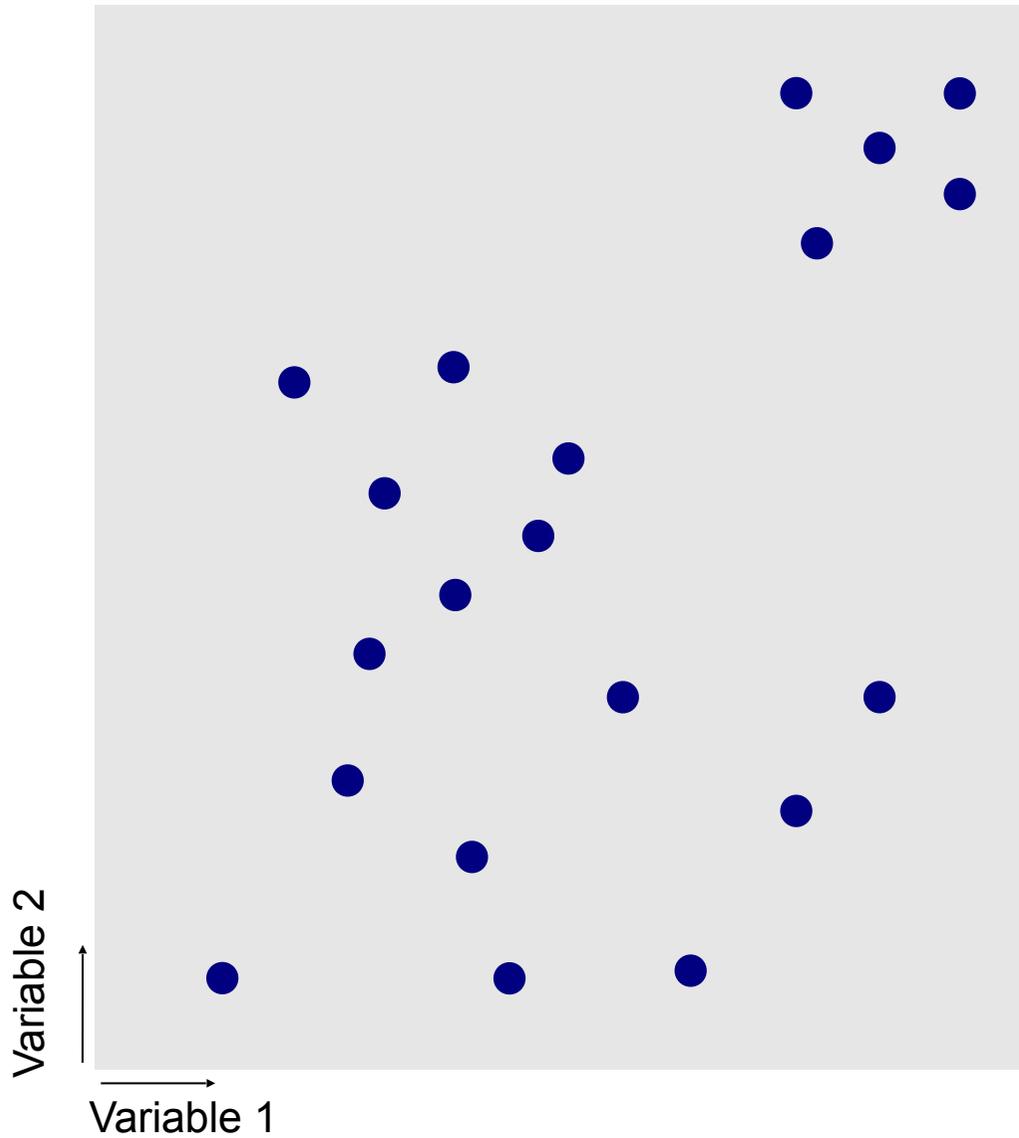
... explicabilité et interprétabilité des décisions des algorithmes

... données complexes (small data)

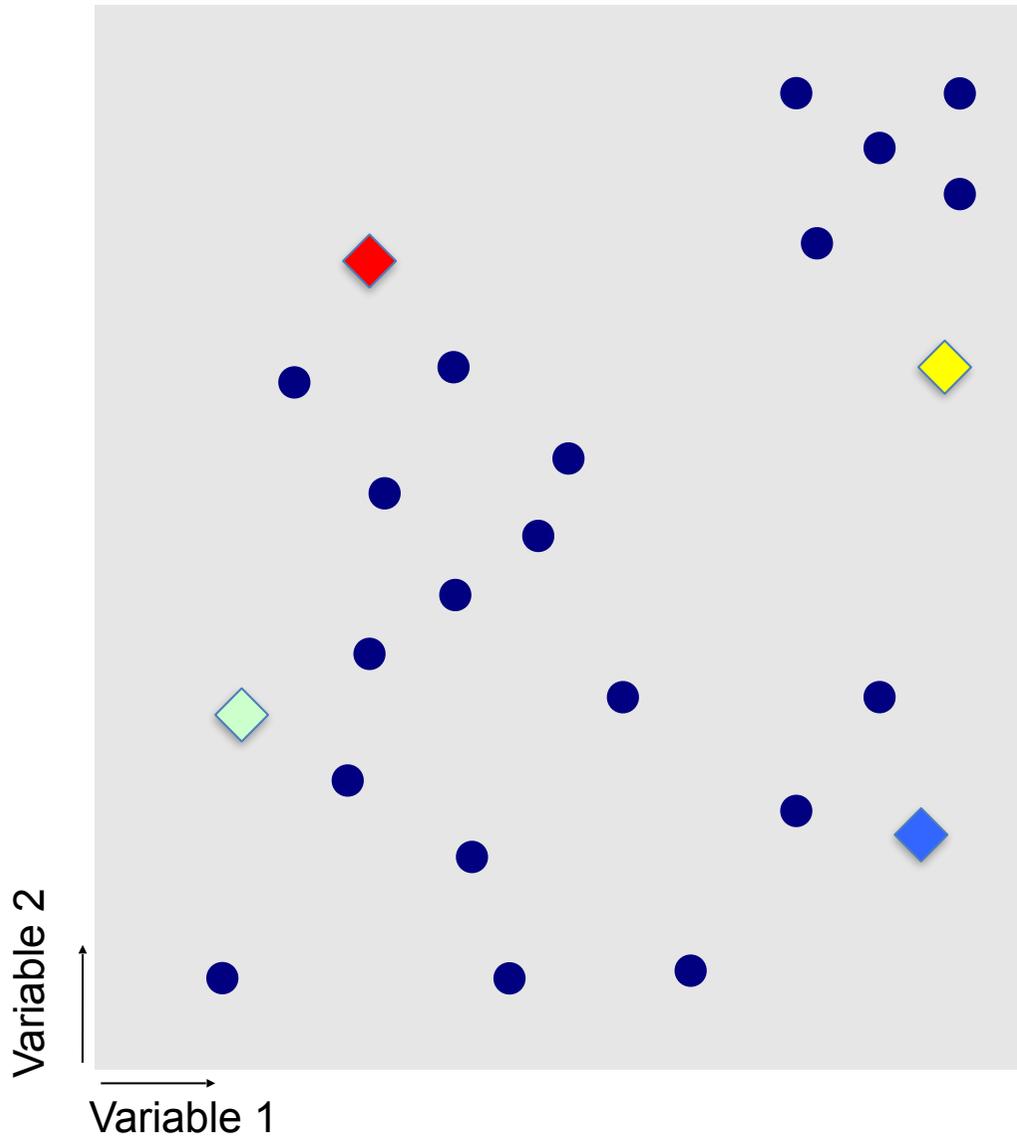
<sup>1</sup>Cours Apprentissage Statistique de P. Besse (INSA Toulouse): <http://www.math.univ-toulouse.fr/~besse/enseignement.html>

## **Algorithmes classiques en machine learning**

### Algorithme des K-means

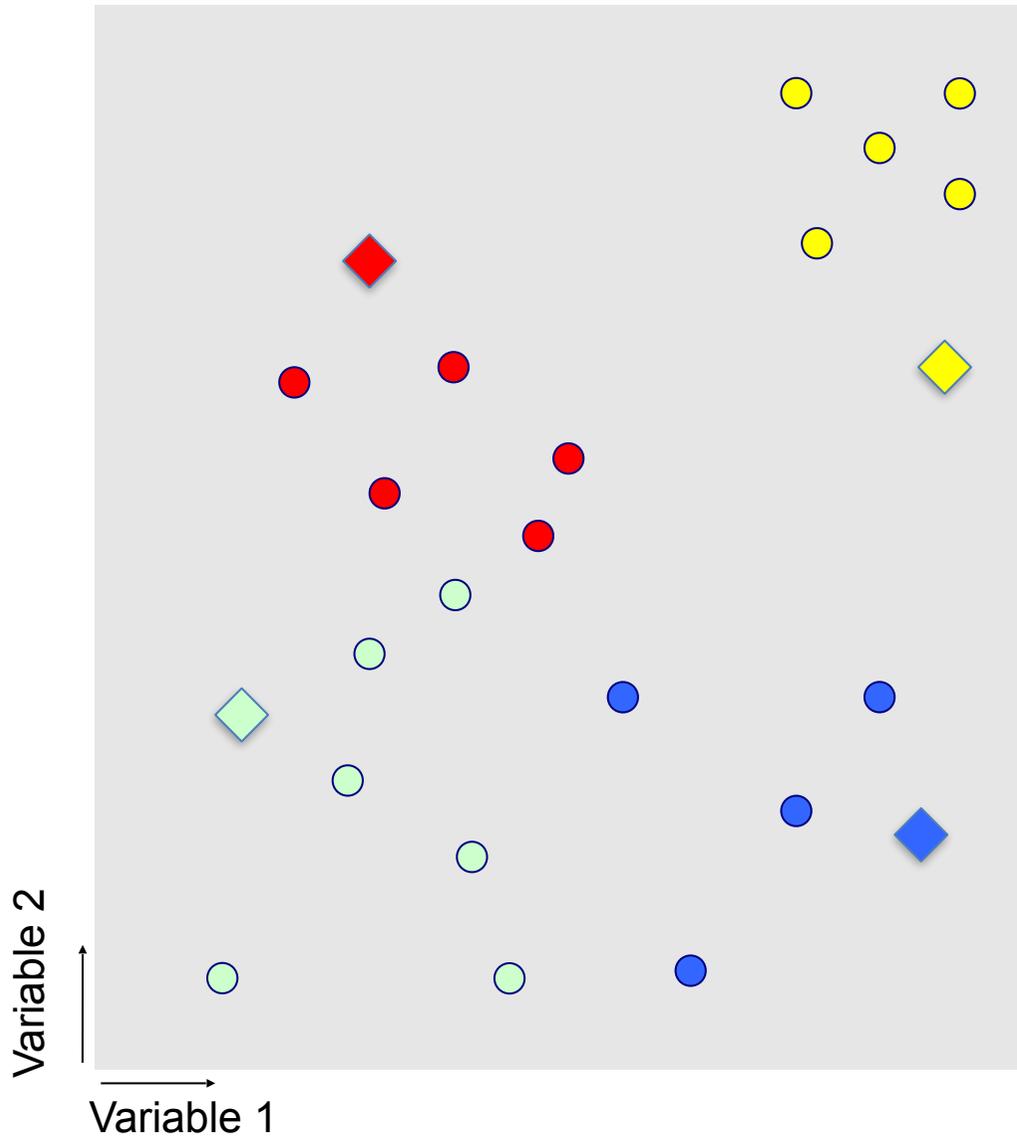


### Algorithme des K-means



On tire  $N$  graines au hasard  
(pour l'exemple  $N=4$ )

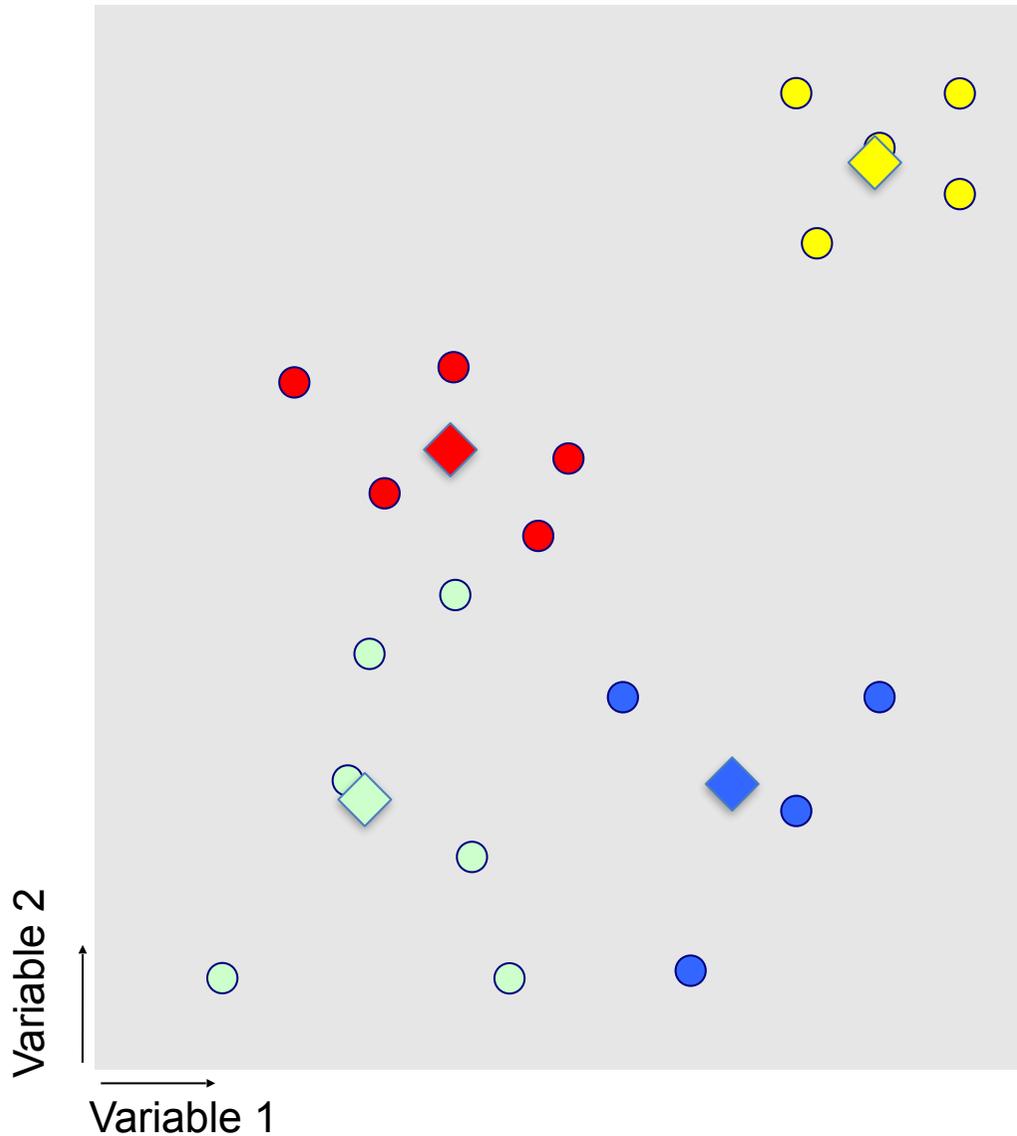
### Algorithme des K-means



Pour chaque observation, on cherche la graine la plus proche.

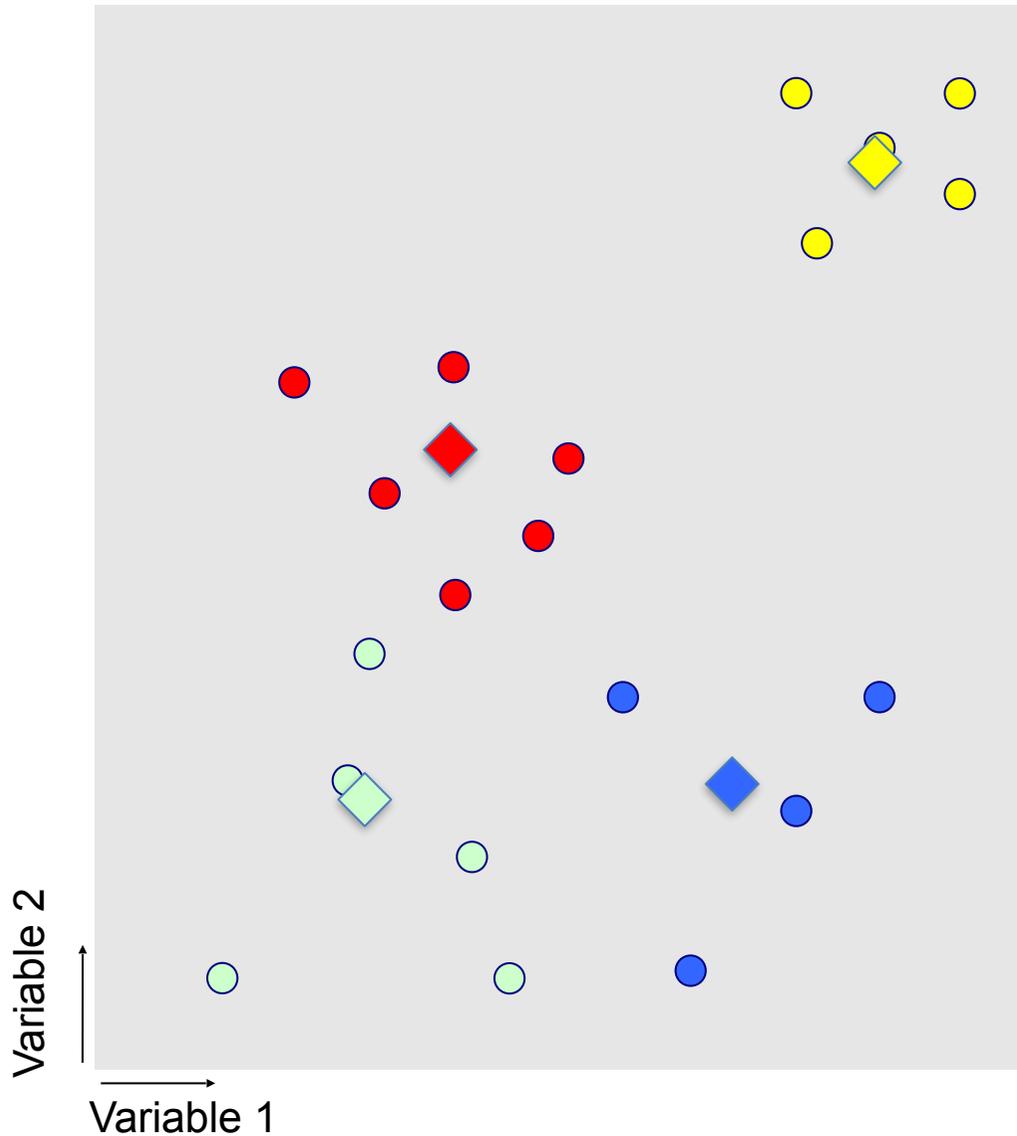
Remarque : Des distances Euclidiennes sont utilisées

### Algorithme des K-means



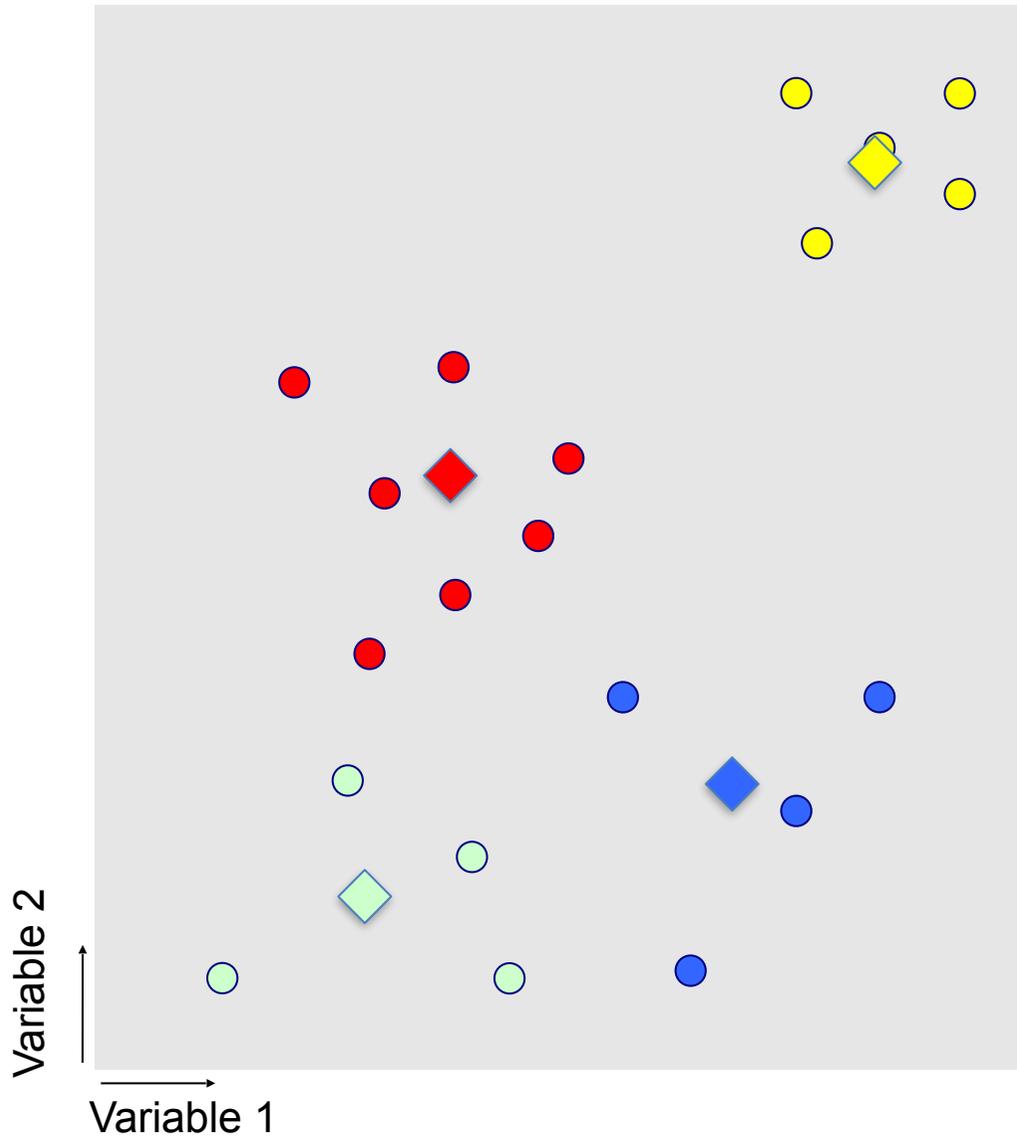
On centre les graines...

### Algorithme des K-means



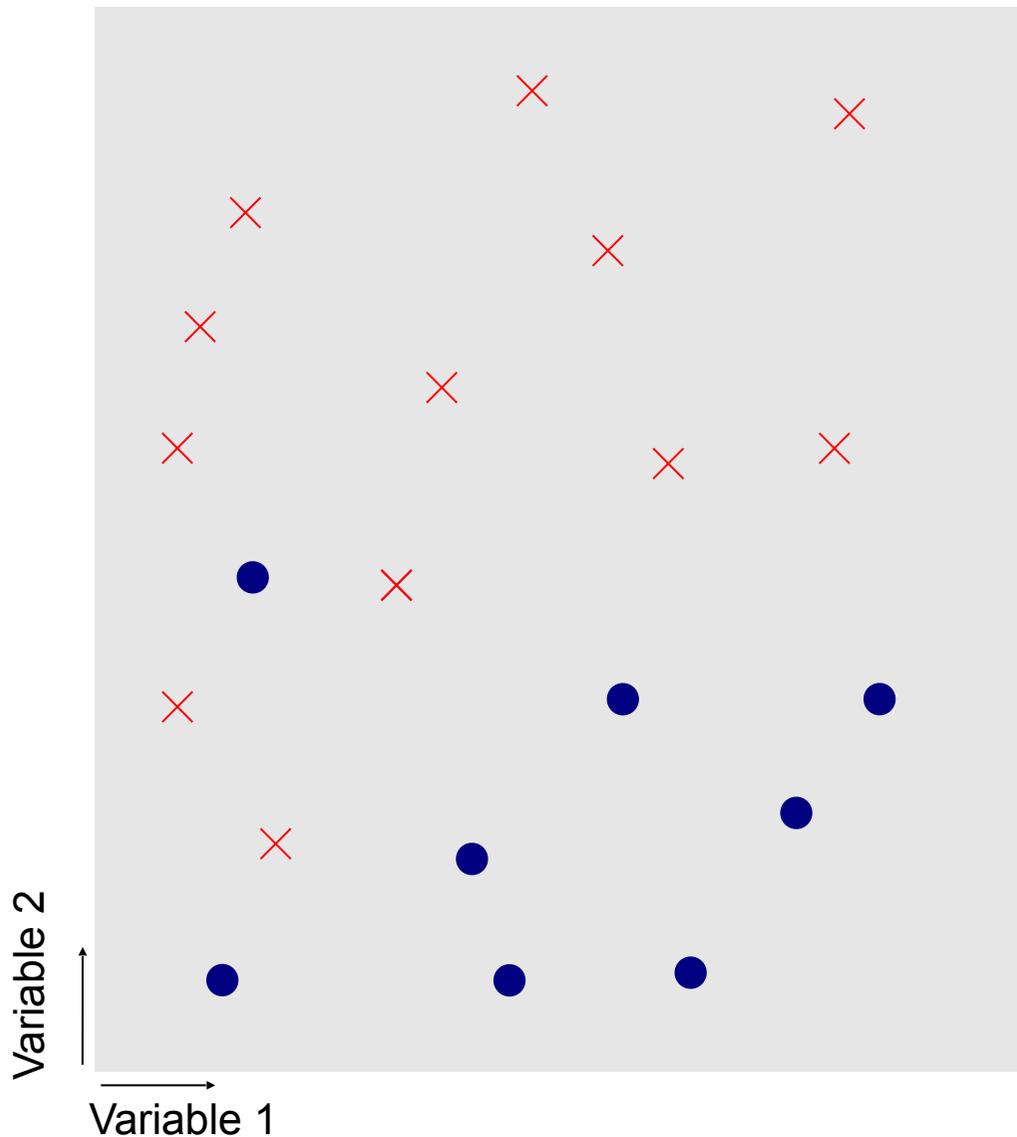
... pour chaque observation, on cherche à nouveau la graine la plus proche ...

### Algorithme des K-means



... et on recommence jusqu'à convergence.

### Arbres de décision

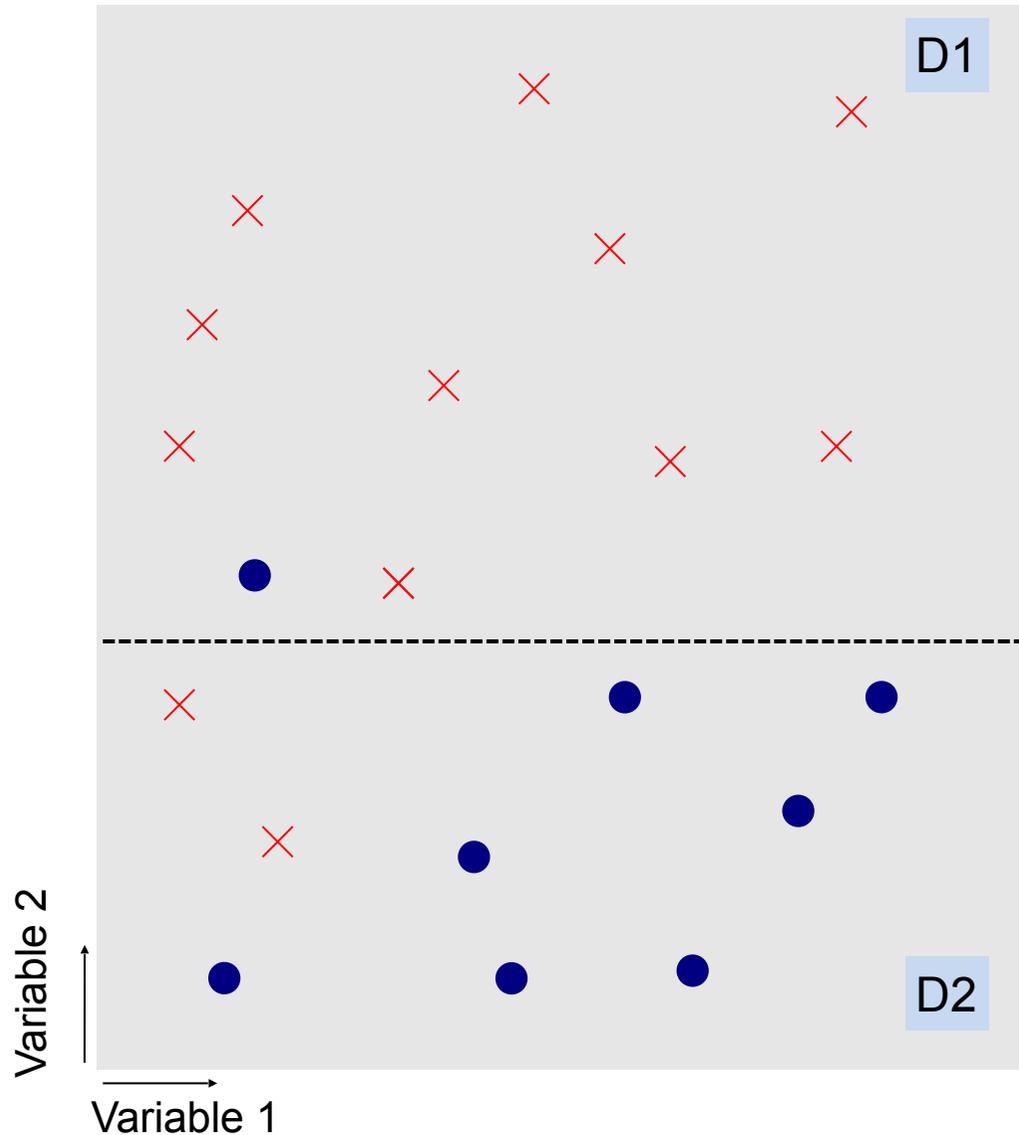


$x_1, x_2, \dots, x_N$  les observations  
(ici :  $x_i$  est la coordonnée du point en 2D)

$y_1, y_2, \dots, y_N$  les labels  
(ici :  $\times \rightarrow 1$  et  $\bullet \rightarrow -1$ )

On coupe le domaine en *sous domaines*  
pour minimiser la variance dans chaque  
sous domaine (CART).

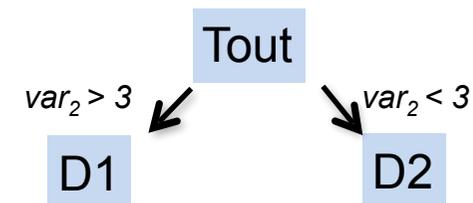
### Arbres de décision



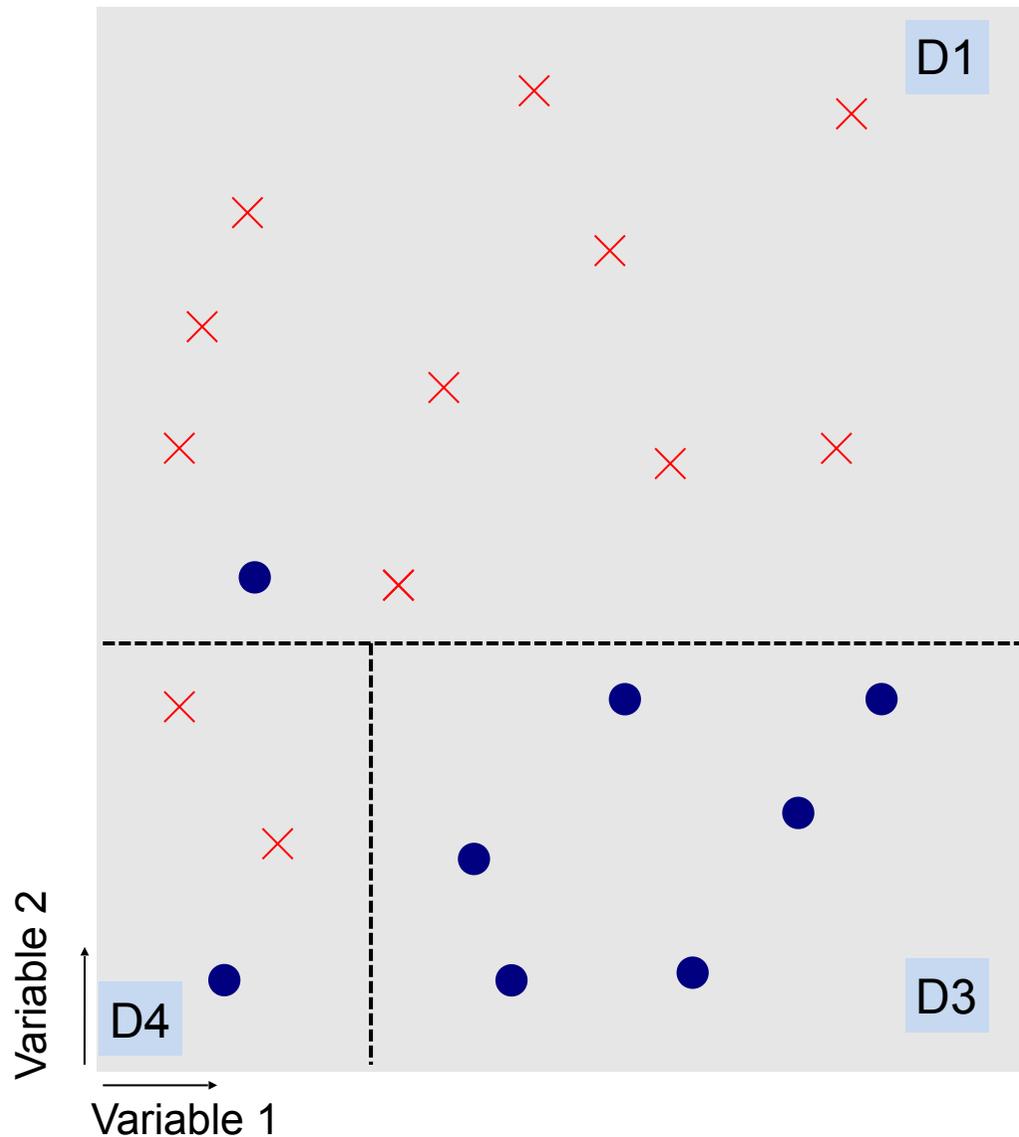
$x_1, x_2, \dots, x_N$  les observations  
(ici :  $x_i$  est la coordonnée du point en 2D)

$y_1, y_2, \dots, y_N$  les labels  
(ici :  $\times \rightarrow 1$  et  $\bullet \rightarrow -1$ )

On coupe le domaine en *sous domaines* pour minimiser la variance dans chaque sous domaine (CART).



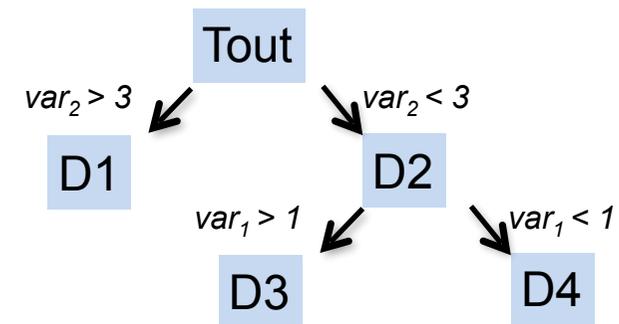
### Arbres de décision



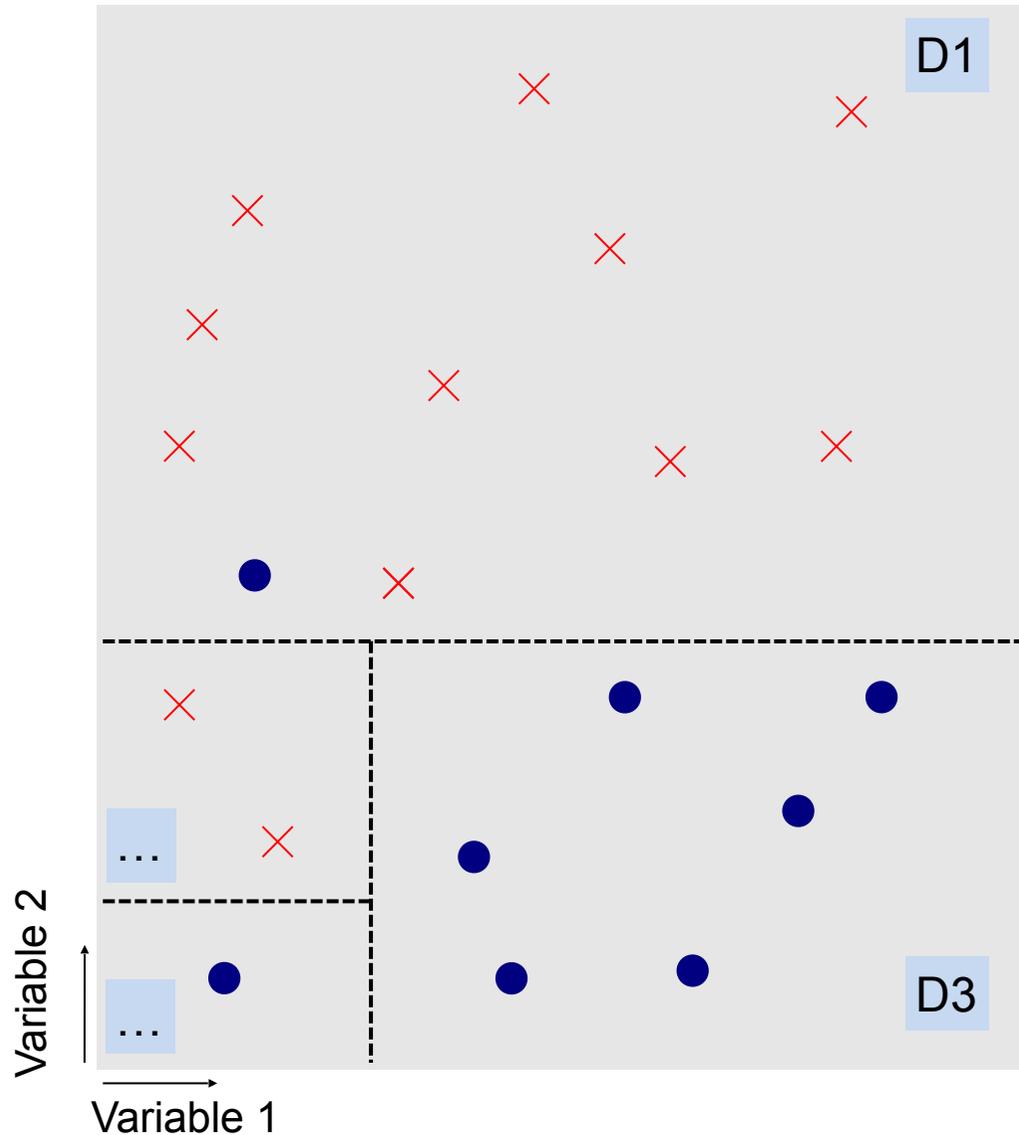
$x_1, x_2, \dots, x_N$  les observations  
(ici :  $x_i$  est la coordonnée du point en 2D)

$y_1, y_2, \dots, y_N$  les labels  
(ici :  $\times \rightarrow 1$  et  $\bullet \rightarrow -1$ )

On coupe le domaine en *sous domaines* pour minimiser la variance dans chaque sous domaine (CART).



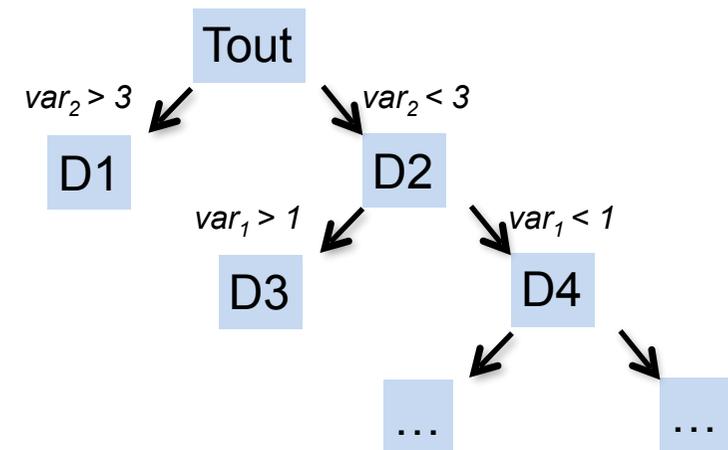
### Arbres de décision



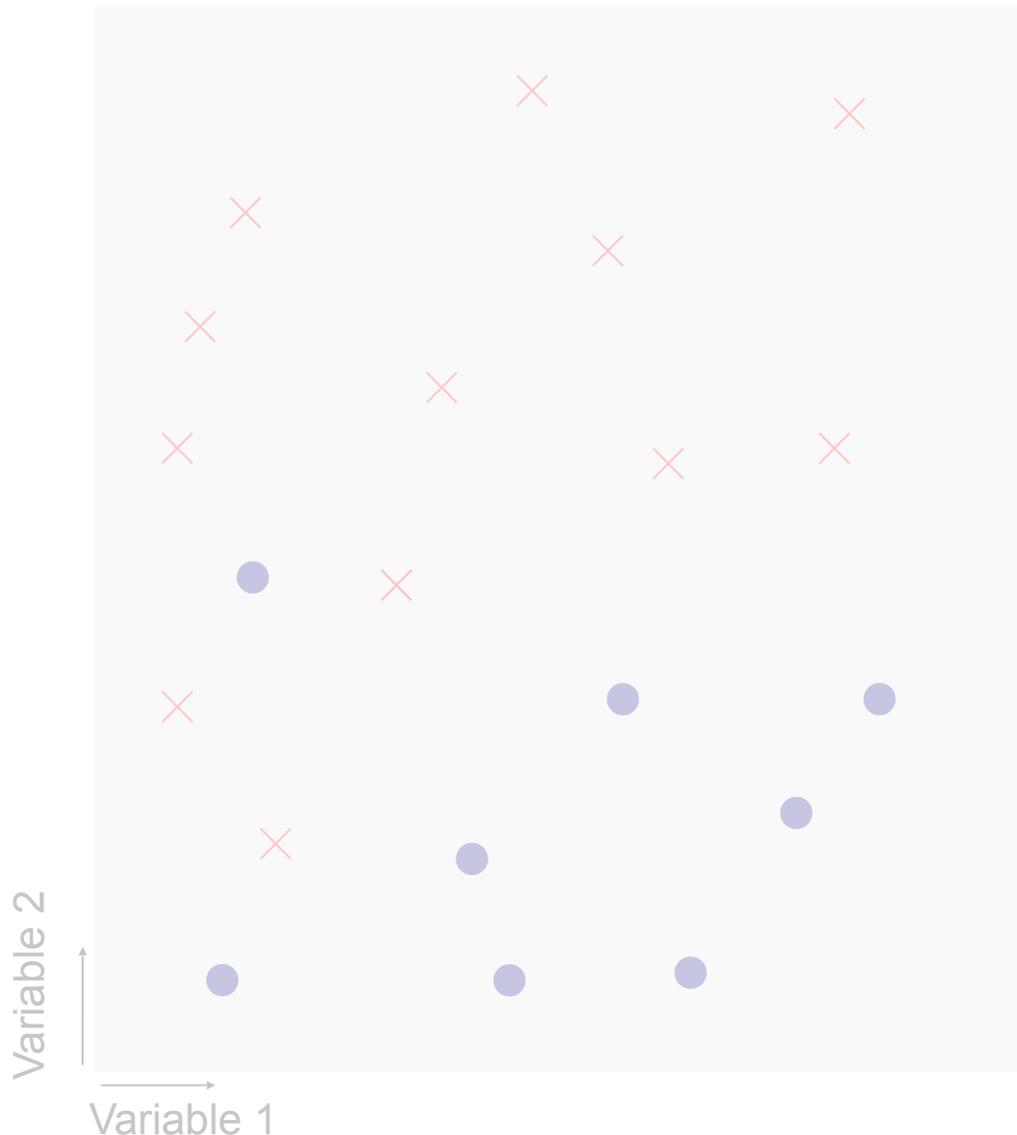
$x_1, x_2, \dots, x_N$  les observations  
(ici :  $x_i$  est la coordonnée du point en 2D)

$y_1, y_2, \dots, y_N$  les labels  
(ici :  $\times \rightarrow 1$  et  $\bullet \rightarrow -1$ )

On coupe le domaine en *sous domaines* pour minimiser la variance dans chaque sous domaine (CART).



### Random forest



$x_1, x_2, \dots, x_N$  les observations  
(ici :  $x_i$  est la coordonnée du point en 2D)

$y_1, y_2, \dots, y_N$  les labels  
(ici :  $\times \Rightarrow 1$  et  $\bullet \Rightarrow -1$ )

Contexte de grande dimension :  $p \gg 1$   
et pas  $p = 2$  comme ici.

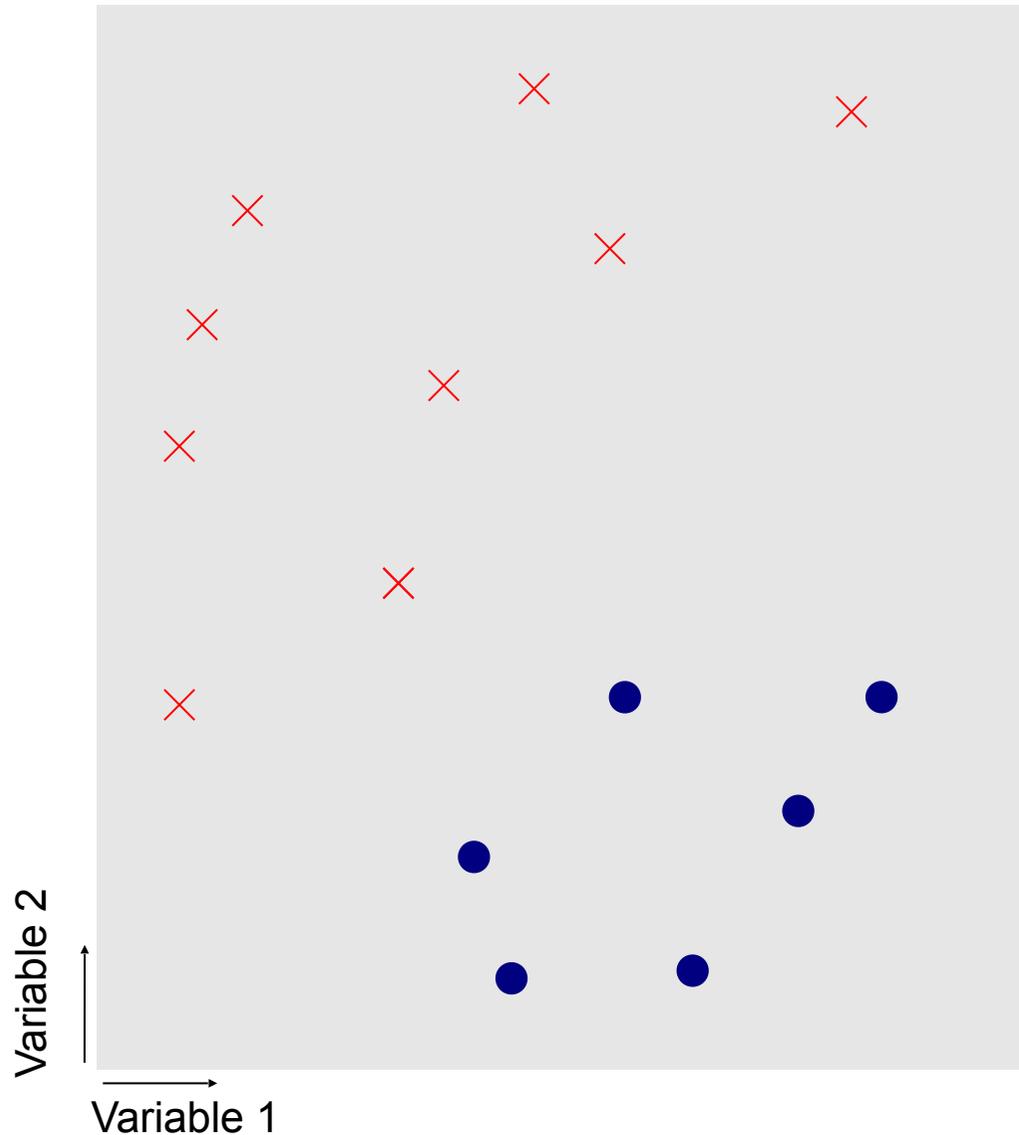
Apprentissage :

- Plusieurs arbres sont définis de manière indépendante.
- Les dimensions de coupes sont tirées au hasard.

Prédiction :

- Le label prédit en un point donné est celui prédit par la majorité des arbres (bagging).

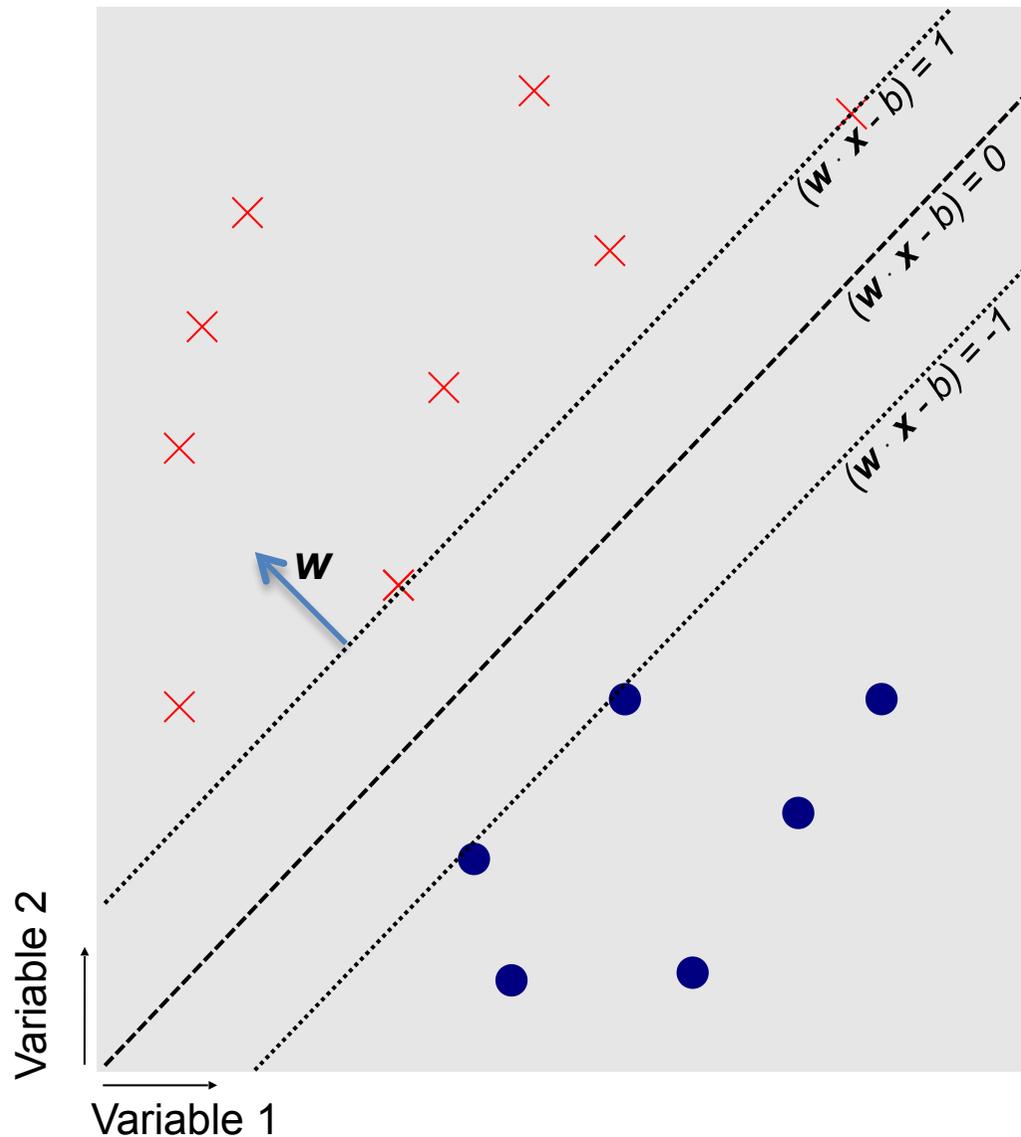
### Les Support Vector Machine (SVM) — Principe



$\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$  les observations  
(ici :  $x_i$  est la coordonnée du point en 2D)

$y_1, y_2, \dots, y_N$  les labels  
(ici :  $\times \rightarrow 1$  et  $\bullet \rightarrow -1$ )

### Les Support Vector Machine (SVM) — Principe



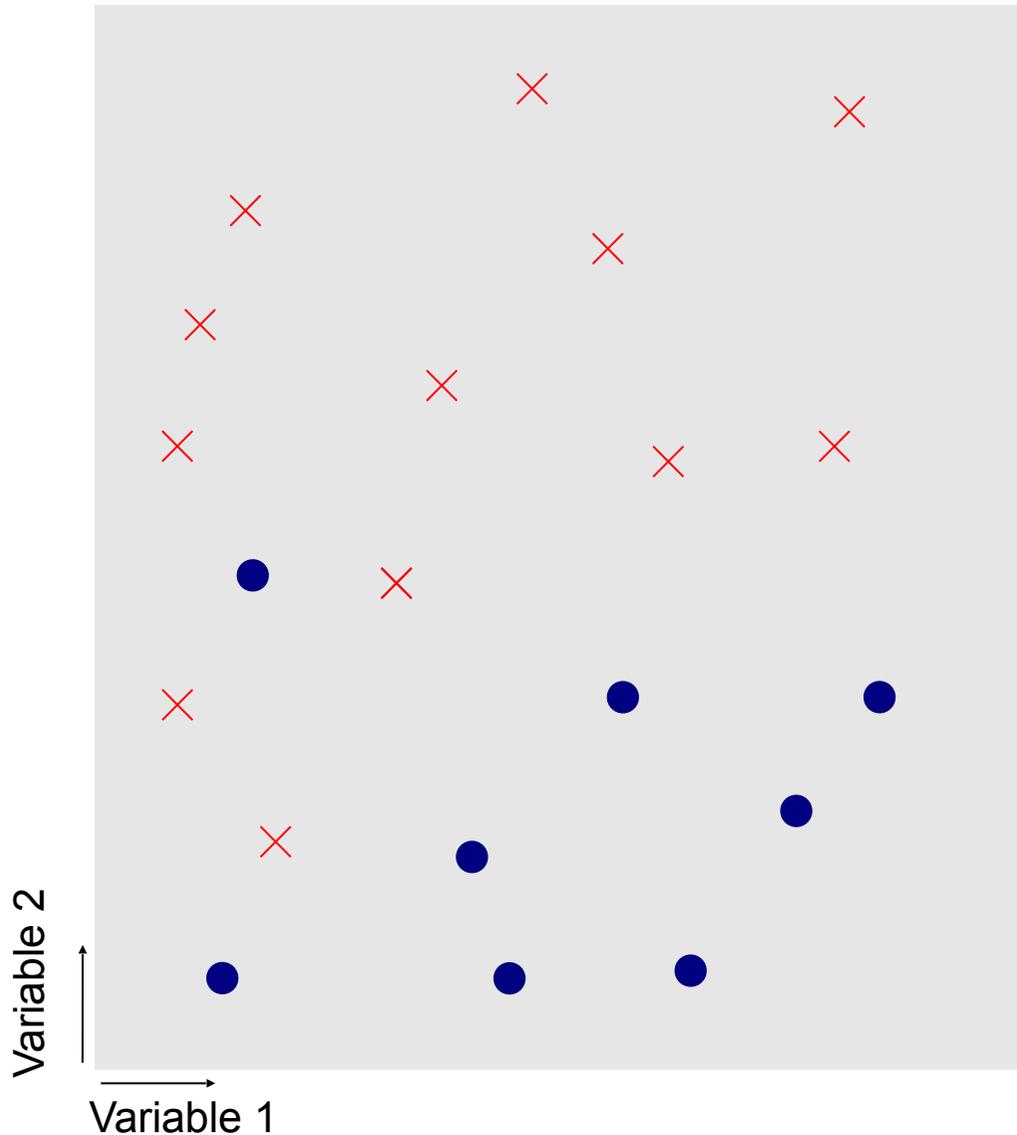
$\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$  les observations  
(ici :  $x_i$  est la coordonnée du point en 2D)

$y_1, y_2, \dots, y_N$  les labels  
(ici :  $\times \rightarrow 1$  et  $\bullet \rightarrow -1$ )

On optimise  $w$  et  $b$  tel que :

$$y_i (w \cdot x_i - b) \geq 1 \text{ pour tout } 1 \leq i \leq n$$

### Les Support Vector Machine (SVM) — Formulation

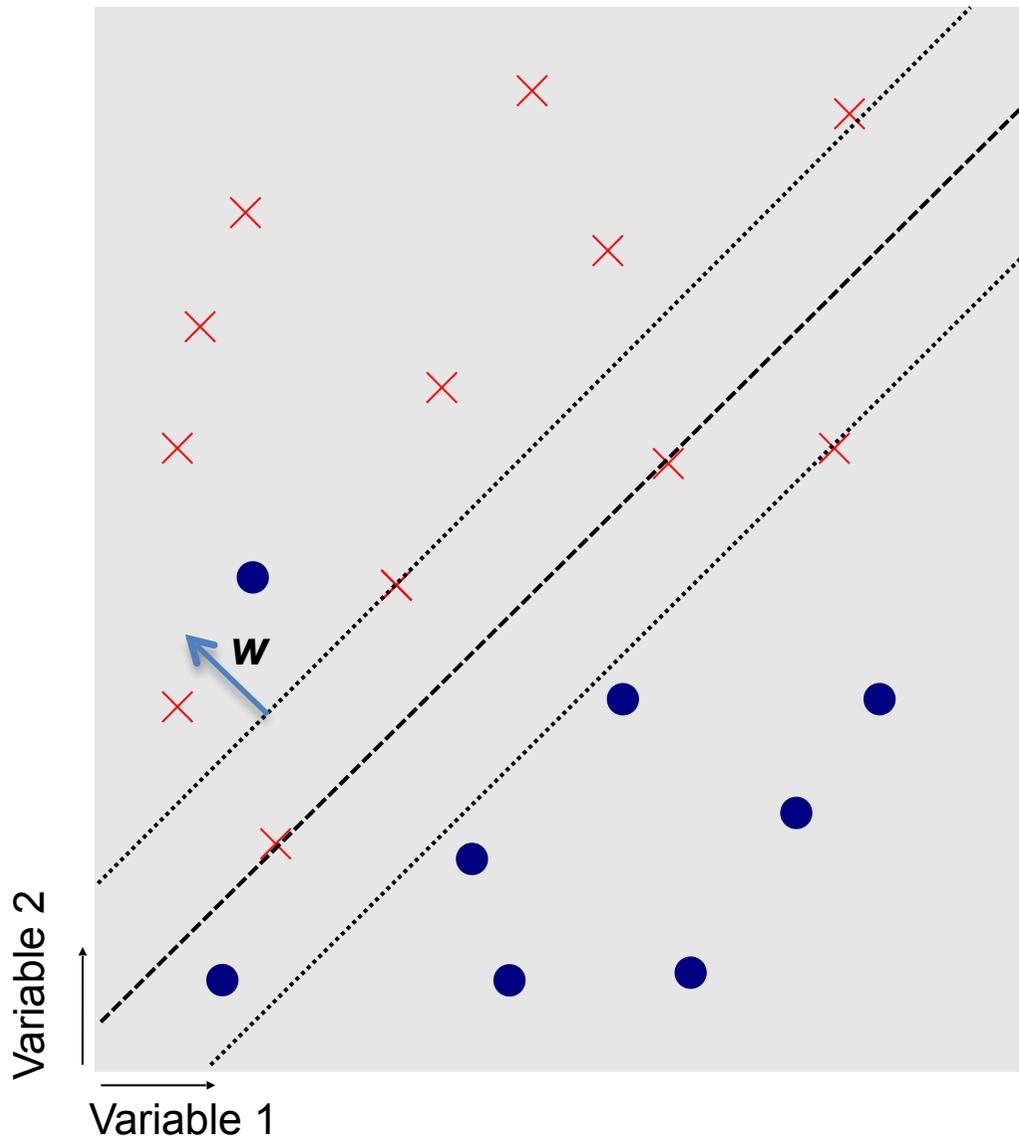


$\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$  les observations  
(ici :  $x_i$  est la coordonnée du point en 2D)

$y_1, y_2, \dots, y_N$  les labels  
(ici :  $\times \rightarrow 1$  et  $\bullet \rightarrow -1$ )

Que faire si il est impossible de séparer  
tous les points?

### Les Support Vector Machine (SVM) — Formulation



$\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$  les observations  
(ici :  $x_i$  est la coordonnée du point en 2D)

$y_1, y_2, \dots, y_N$  les labels  
(ici :  $\times \rightarrow 1$  et  $\bullet \rightarrow -1$ )

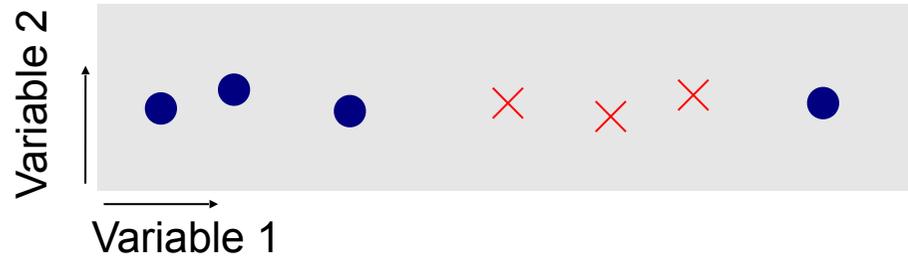
Que faire si il est impossible de séparer  
tous les points?

On cherche  $\mathbf{w}$  et  $b$  qui minimisent :

$$\left[ n^{-1} \sum_{i=1}^n \max(0, 1 - y_i (\mathbf{w} \cdot \mathbf{x}_i - b)) \right] + \lambda \|\mathbf{w}\|_2^2$$

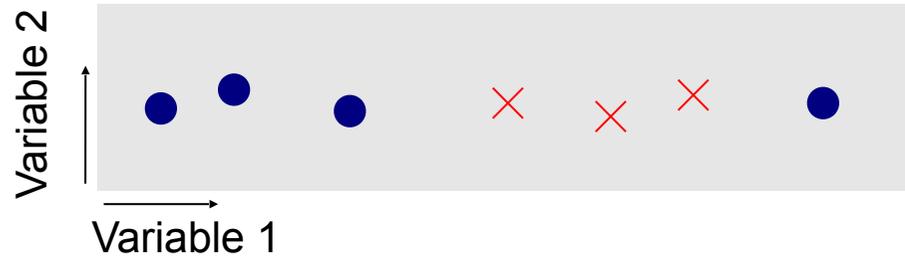
> 0 si  $\mathbf{x}_i$  est  
mal classé

### Les Support Vector Machine (SVM) — Méthodes à noyaux



Que faire dans ce cas là ?

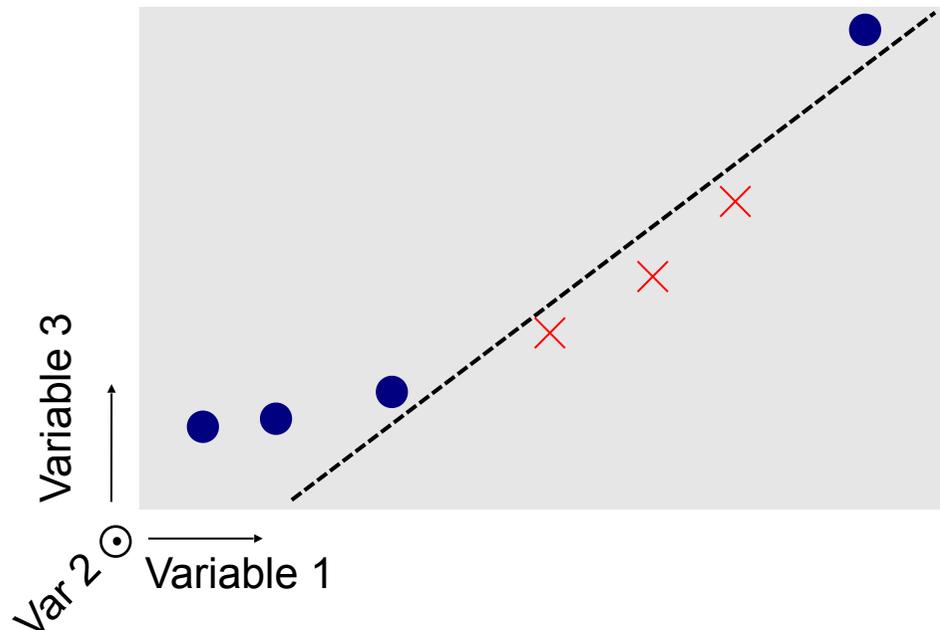
### Les Support Vector Machine (SVM) — Méthodes à noyaux



Que faire dans ce cas là ?

On note  $x_i = (x_i^1, x_i^2)$  une observation

On va séparer les  $\Phi(x_i) = (x_i^1, x_i^2, (x_i^1)^2)$  plutôt que les  $x_i$



### Régression logistique

Pour chaque observation  $i \in \{1, \dots, n\}$  :

- Variable explicative :  $x_i = (x_i^1, x_i^2, \dots, x_i^p)$  avec  $p \gg 1$
- Variable de réponse :  $y_i \in \{-1, 1\}$

Estimation de  $\beta = (\beta_0, \beta_1, \dots, \beta_n)$  tel que :

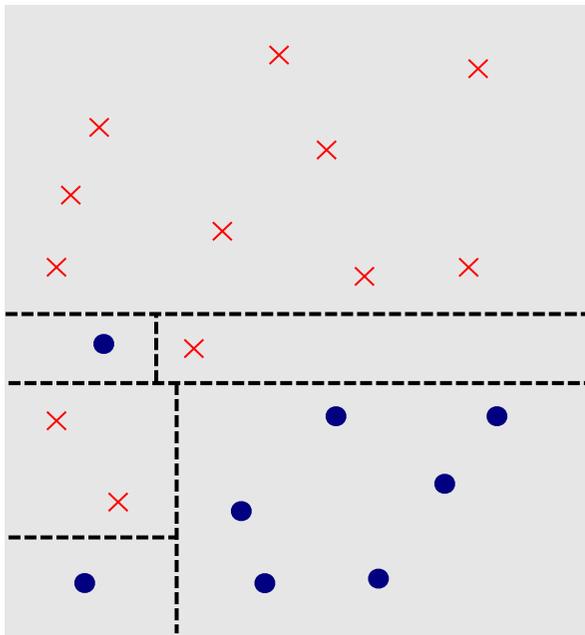
- $\beta_0 + \sum_{j=1}^p \beta_j x_i^j$  est positif si  $y_i = 1$
- $\beta_0 + \sum_{j=1}^p \beta_j x_i^j$  est négatif si  $y_i = -1$

#### Remarques :

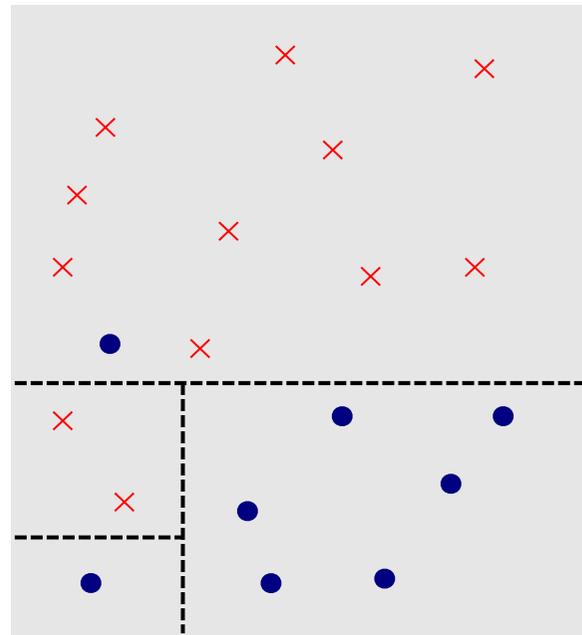
- Estimation au sens de la Log-vraisemblance en fonction d'un modèle logistique
- Contraintes possibles (et courantes) sur  $\beta$  en particulier quand  $p > n$
- Passe à l'échelle quand  $n$  est très grand

## **Sur-apprentissage et validation croisée**

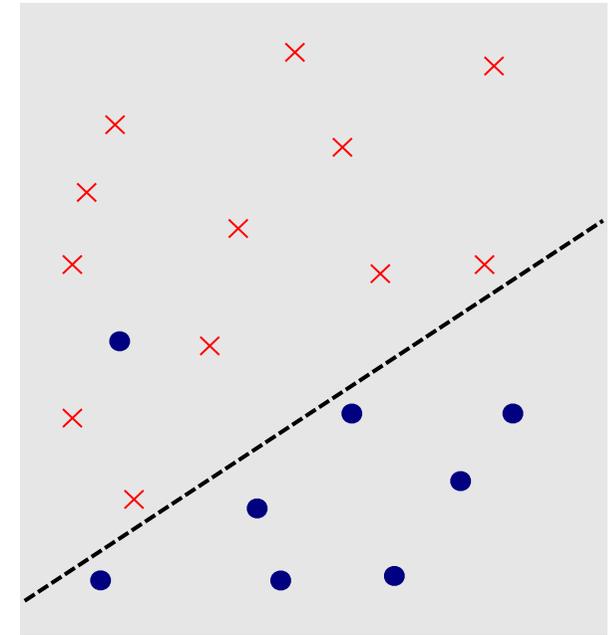
## Sur-apprentissage



Arbre de décision complet

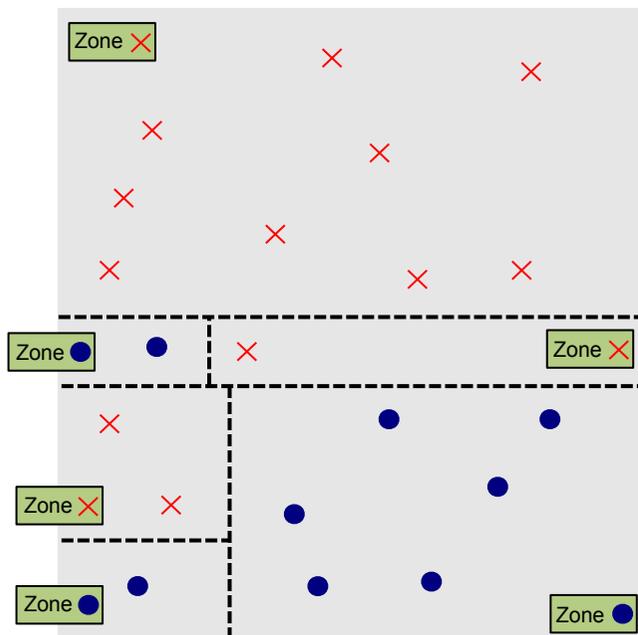


Arbre de décision incomplet

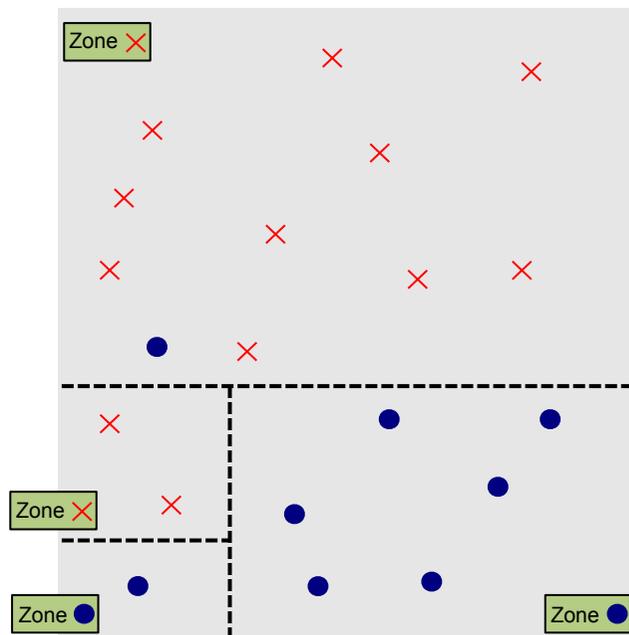


SVM linéaire

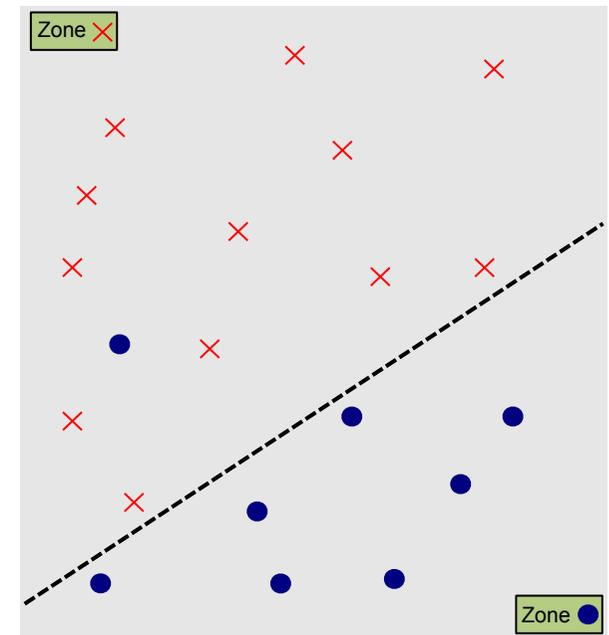
## Sur-apprentissage



Arbre de décision complet



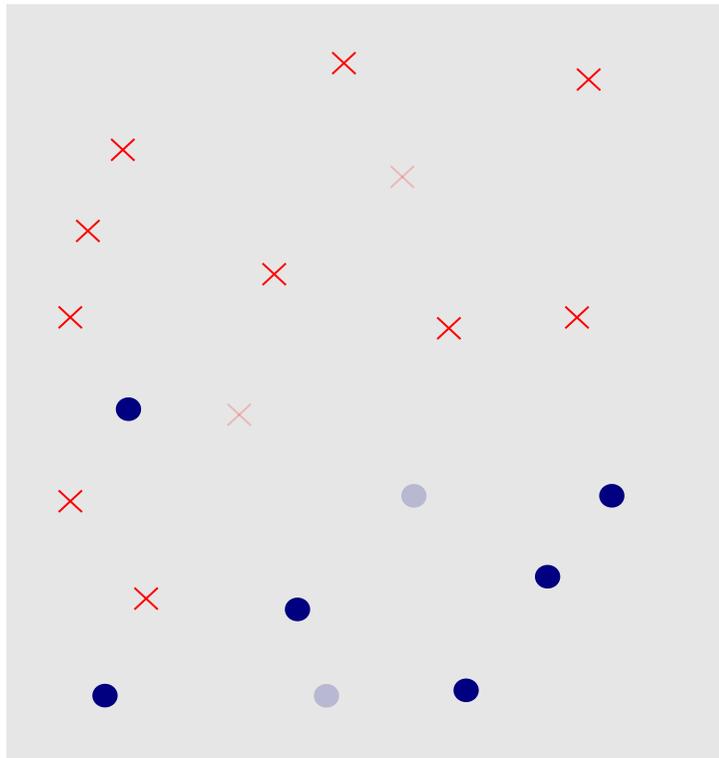
Arbre de décision incomplet



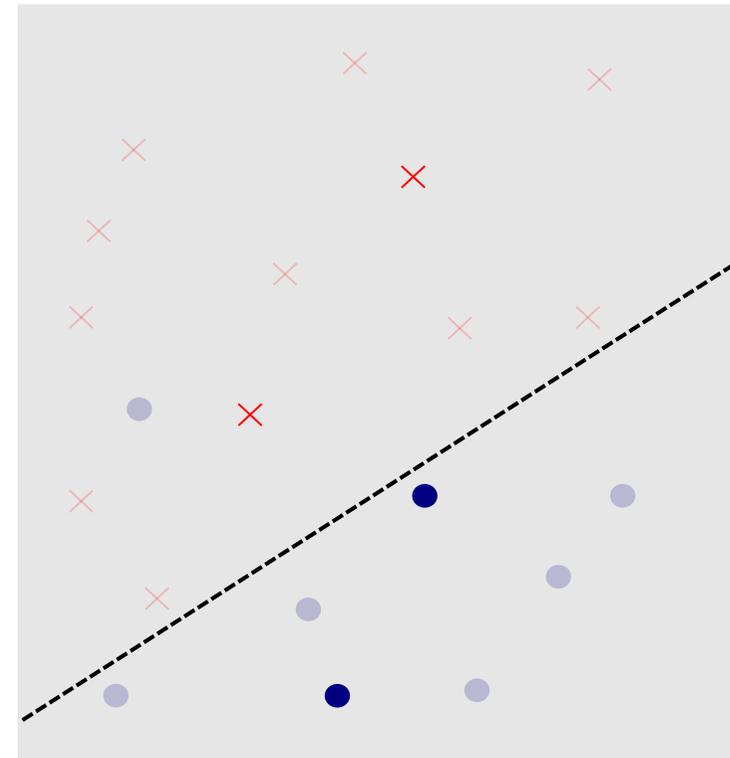
SVM linéaire

A quelle stratégie feriez-vous le plus confiance pour prédire le label d'une nouvelle observation ?

#### Séparation des données d'apprentissage et données test



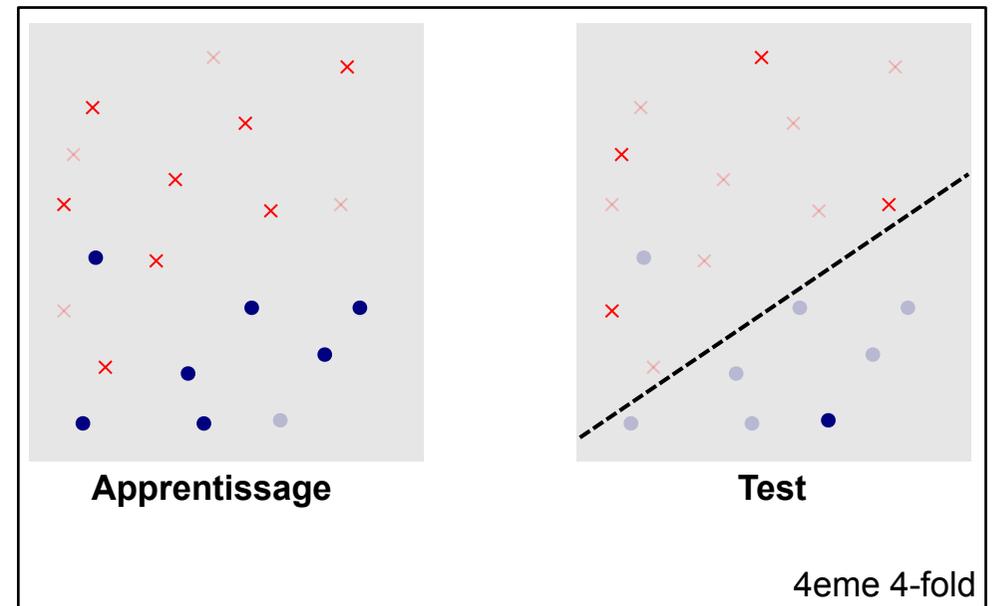
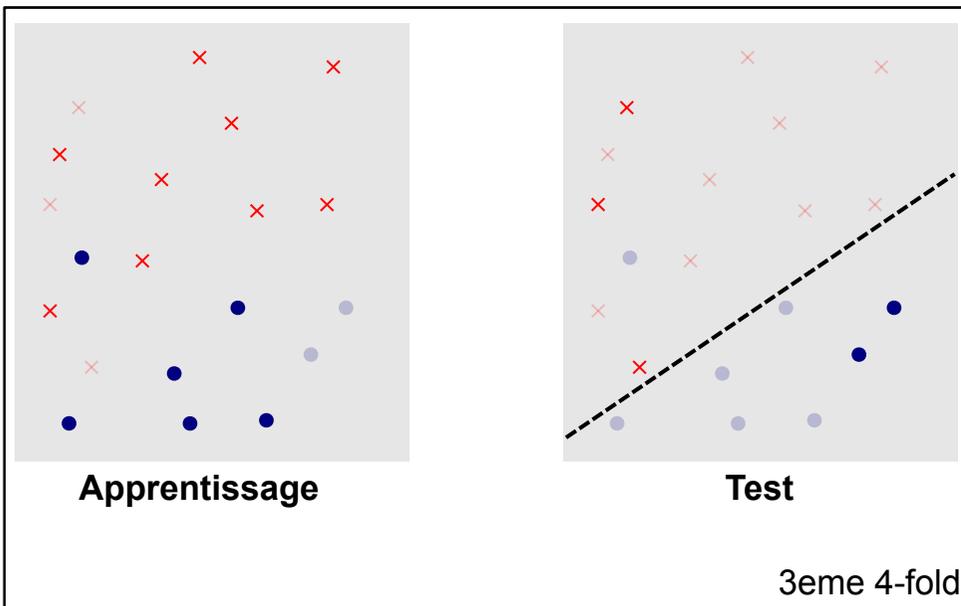
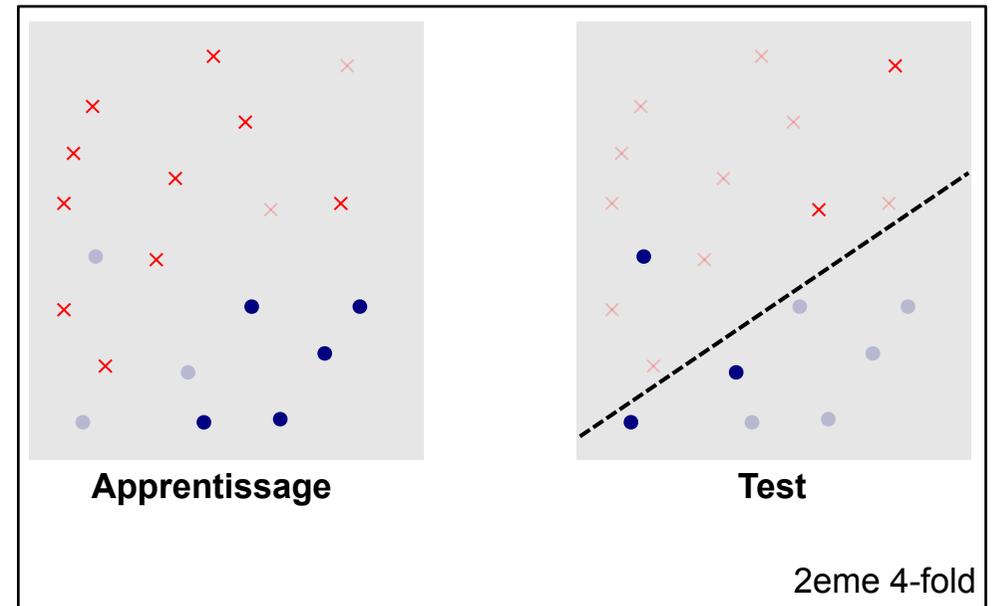
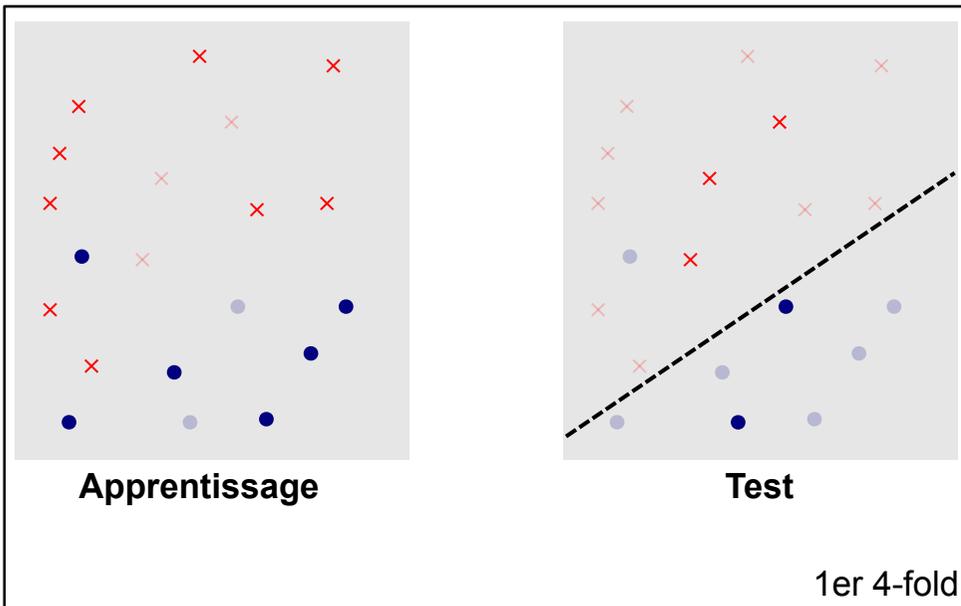
Apprentissage sur toutes les observations sauf 4



Test sur les 4 données enlevées

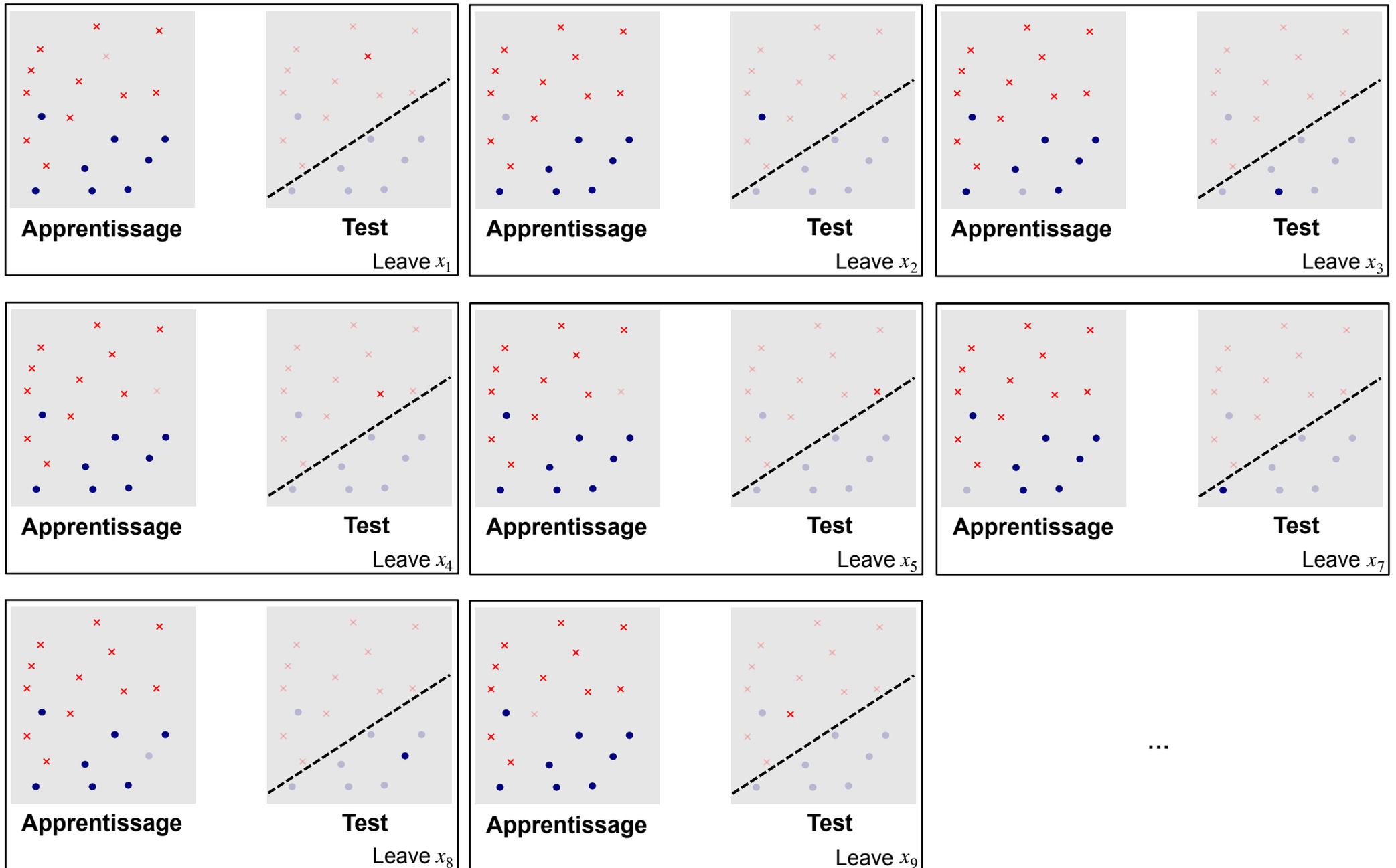
# 3.b) Sur-apprentissage et validation croisée — validation croisée

## K-folds



# 3.b) Sur-apprentissage et validation croisée — validation croisée

## Leave-1-out

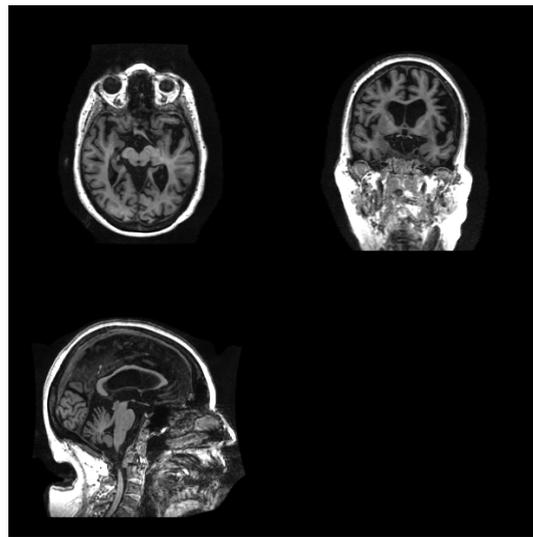


# **Grande dimension... régularisation et sélection de modèle**

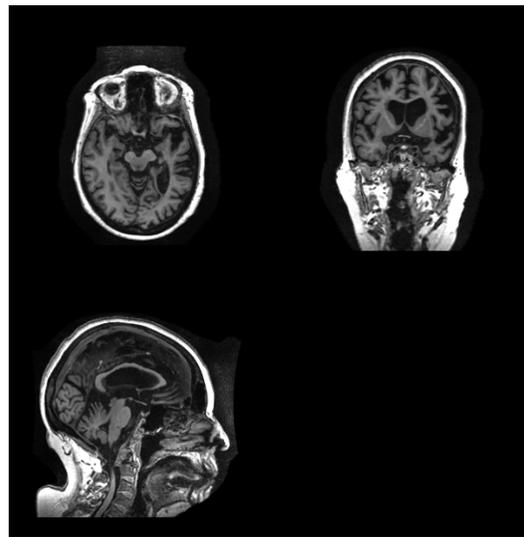
## 4.a) Grande dimension — exemple de données

### Context du projet :

- Observations = IRM du cerveau à différent temps d'acquisition (ADNI\*)
- Labels = Etat du patient (MCI/AD)
- Prédiction maladie d'Alzheimer en fonction de l'évolution morphométrique de l'hippocampe ?



*[Baseline]*



*[Baseline + 12 mois]*



*Hippocampe*

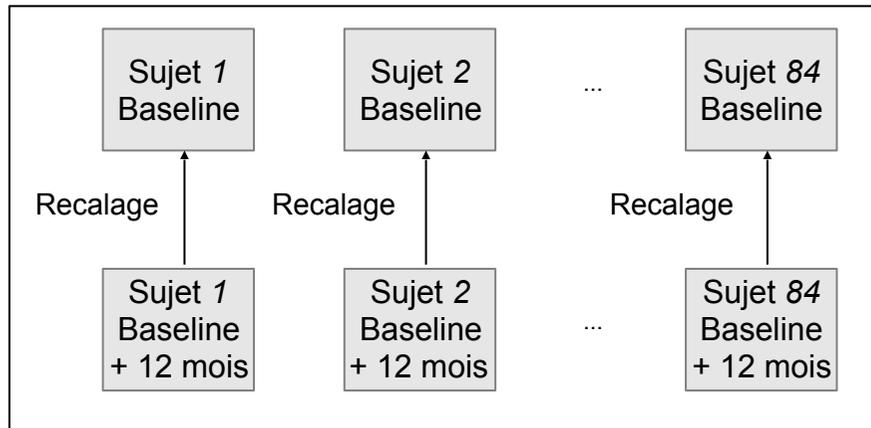
### Echantillon d'apprentissage :

- *[Baseline]* : 103 patients sont MCI
- *[Baseline + 12 mois]* : 84 patients sont MCI / 19 patients sont AD

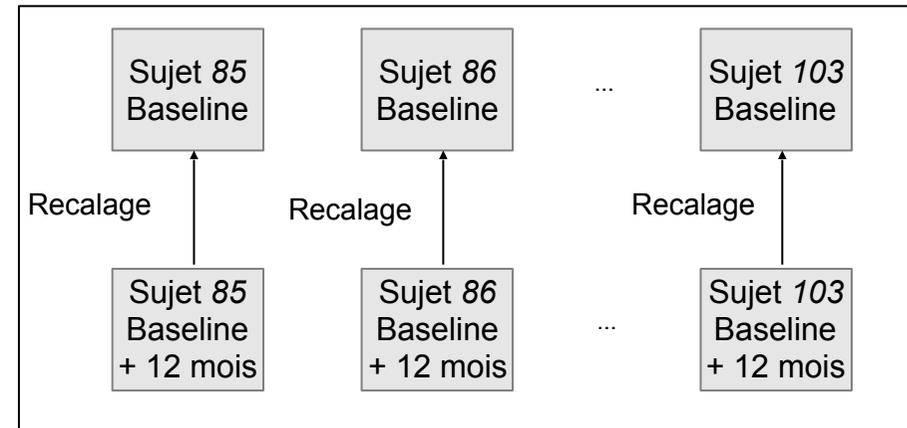
\* <http://adni.bmap.ucla.edu/>

## 4.a) Grande dimension — exemple de données

### Pré-traitement des données d'apprentissage :



Groupe des MCI



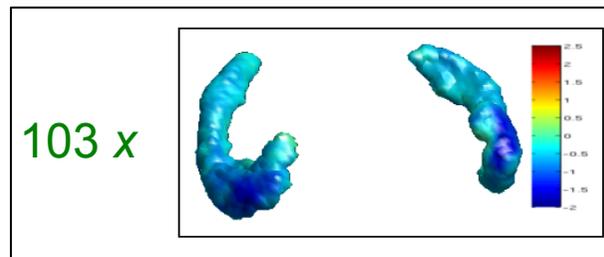
Groupe des AD

Suivi des déformations et transport des marqueurs d'évolution sur une forme *moyenne* [Vialard et al. IJCV, 2012]

### Données d'apprentissage :

Pour chacun des  $n = 103$  sujets :

- $\mathbf{x}_i$  : Observation de l'évolution de la forme sur environ  $p = 20000$  points
- $y_i$  : Etat AD ou MCI

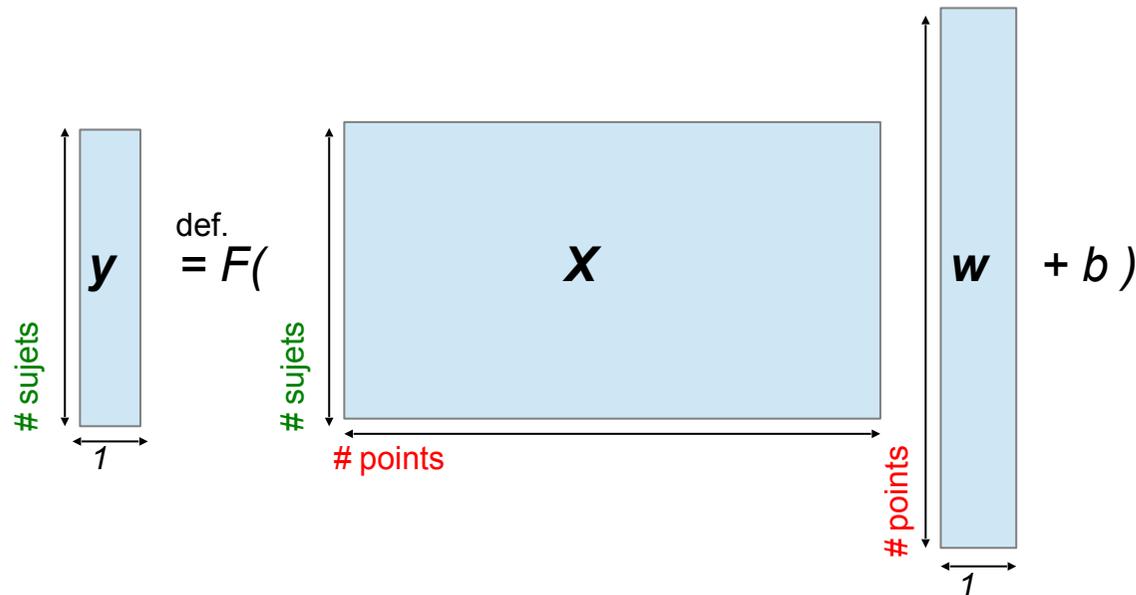


**Questions :** Discrimination possible ? Quels sont les points les plus discriminants ???

## 4.b) Grande dimension — modélisation

Modèle prédictif de régression logistique qui définit la probabilité des  $y_i$  en fonctions des  $\mathbf{x}_i$  :

$$p(y_i | \mathbf{x}_i, \mathbf{w}, b) \stackrel{\text{def.}}{=} \frac{1}{1 + \exp(-y_i (\mathbf{x}_i^T \mathbf{w} + b))}$$



Où :

$\mathbf{X} \in \mathbf{R}^{n \times p}$  : matrice des  $n = 103$  observations de dimension  $p = 20000$

$\mathbf{y} \in \{-1, 1\}^n$  : Etat ( $AD = -1 / MCI = 1$ )

$(\mathbf{w}, b) \in \mathbf{R}^p * \mathbf{R}$  : paramètres à estimer

Optimisation de la log-vraisemblance :

**Paramètre de régularisation obligatoire car  $p > n$**

Find  $(\hat{\mathbf{w}}, \hat{b})$  in  $\underset{\mathbf{w}, b}{\operatorname{argmin}} \mathcal{L}(\mathbf{w}, b) + \lambda J(\mathbf{w})$  où :

$$\mathcal{L}(\mathbf{w}, b) \stackrel{\text{def.}}{=} \frac{1}{n} \sum_{i=1}^n \log(1 + \exp(-y_i (\mathbf{x}_i^T \mathbf{w} + b)))$$

## 4.b) Grande dimension — modélisation

Find  $(\hat{\mathbf{w}}, \hat{b})$  in  $\operatorname{argmin}_{\mathbf{w}, b} \mathcal{L}(\mathbf{w}, b) + \lambda J(\mathbf{w})$  où :  $\mathcal{L}(\mathbf{w}, b) \stackrel{\text{def.}}{=} \frac{1}{n} \sum_{i=1}^n \log(1 + \exp(-y_i(\mathbf{x}_i^T \mathbf{w} + b)))$

Exploration de différents modèles de régularisation :

(1) Ridge :  $J(\mathbf{w}) = \|\mathbf{w}\|_2$

(2) LASSO :  $J(\mathbf{w}) = \|\mathbf{w}\|_1$

(3) Elastic net :  $J(\mathbf{w}) = \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \|\mathbf{w}\|_2$

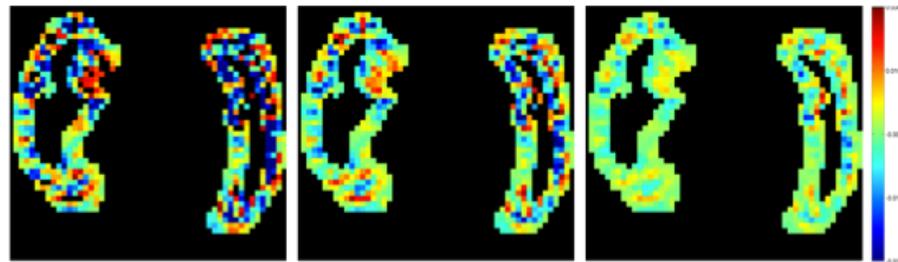
(4) Sobolev semi-norm:  $J(\mathbf{w}) = \sum_{\omega \in \Omega} |\nabla_{\Omega} \mathbf{w}(\omega)|$

(5) Total Variation :  $J(\mathbf{w}) = (\sum_{\omega \in \Omega} |\nabla_{\Omega} \mathbf{w}(\omega)|^2)^{1/2}$

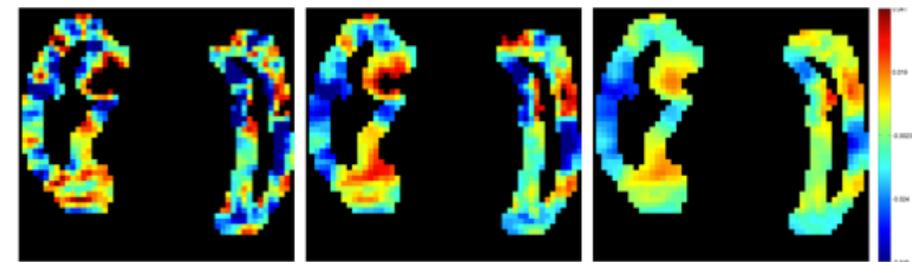
(6) Fused LASSO :  $J(\mathbf{w}) = \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \sum_{\omega \in \Omega} |\nabla_{\Omega} \mathbf{w}(\omega)|$

Minimisation de la log-vraisemblance en fonction de  $\mathbf{w}$  [Lewis & Overton. Math. Programming 2012]

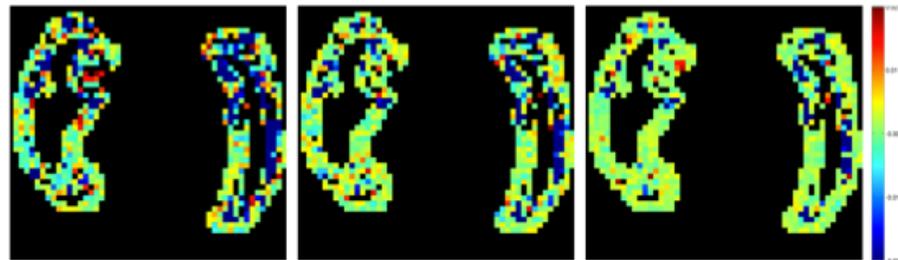
## 4.c) Grande dimension — Effet de la régularisation



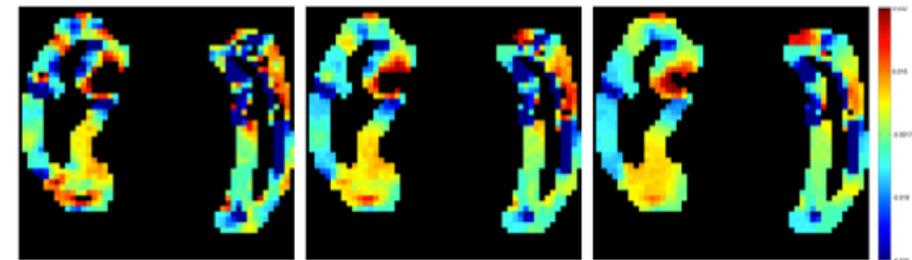
(1) Ridge



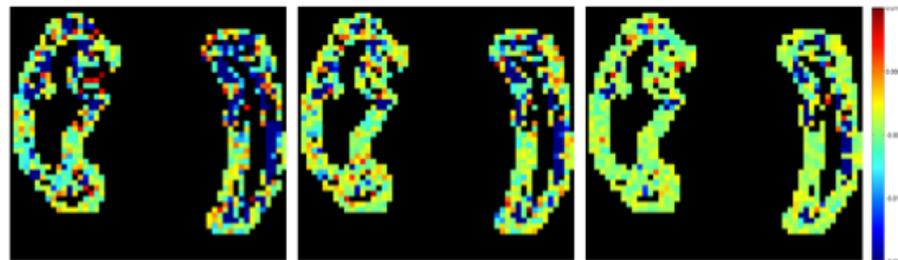
(4) Sobolev semi-norm



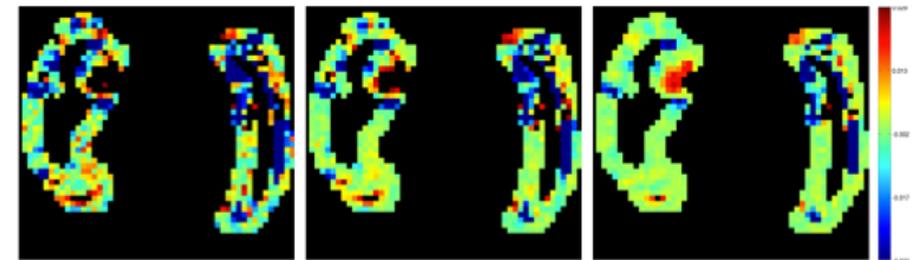
(2) LASSO



(5) Total Variation



(3) Elastic net



(6) Fused LASSO

Représentation de  $w$  pour trois  $\lambda$  sur un plan de l'hippocampe :

- **Bleu** et **rouge** : forte influence locale
- **Vert** : peu ou pas d'influence locale

## 4.c) Grande dimension — Effet de la régularisation

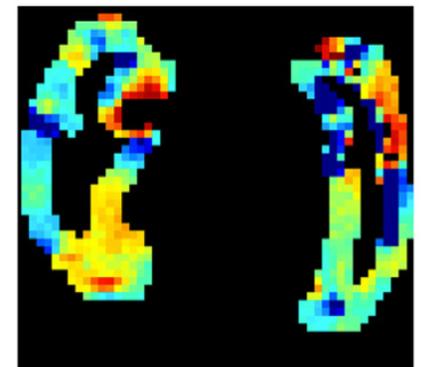
Résultats obtenus avec une méthode de cross validation (ici leave-10%-out) :

- Spec+Sens = 2 → bonne prédiction dans 100% des cas
- Spec+Sens = 1 → pile ou face aurait le même pouvoir prédictif

Regularization		$\lambda$ range	$\hat{\lambda}$ (optimal $\lambda$ )	Spec+ Sens
None		0	0	1.00
Standard	LASSO	$[10^{-9}, 10^0]$	0.01	1.04
	Ridge	$[10^{-9}, 10^0]$	0.001	1.06
	Elastic Net	$[10^{-9}, 10^0]^2$	$\begin{cases} \hat{\lambda}_1 = 0.01 \\ \hat{\lambda}_2 = 1 \end{cases}$	1.13
	Sobolev	$[10^{-9}, 10^7]$	$10^4$	1.17
Spatial	<b>Total Variation</b>	$[10^{-9}, 10^0]$	0.01	<b>1.31</b>
	<b>Fused LASSO</b>	$[10^{-9}, 10^0]^2$	$\begin{cases} \hat{\lambda}_1 = 0.01 \\ \hat{\lambda}_2 = 10^{-4} \end{cases}$	<b>1.32</b>

Meilleurs résultats avec une régularisation en pertinente avec les données :

- Tient compte de la distribution spatiale
- Permet quelques transitions franches



[Fiot J.B. et al., NeuroImage: Clinical, 2012]

**Réduction de dimension par  
Analyse en Composantes Principales (ACP)**

## 5) Réduction de dimension par ACP

Records nationaux (en secondes) de  $p = 9$  épreuves d'athlétisme pour  $n = 26$  pays

	100m	200m	400m	800m	1500m	5000m	10000m	SemiMarathon	Marathon
Australie	9.93	20.06	44.38	104.40	211.96	775.76	1649.73	3602	7671
Belgique	10.02	20.19	44.78	103.86	214.13	769.71	1612.30	3605	7640
Brésil	10.00	19.89	44.29	101.77	213.25	799.43	1648.12	3573	7565
RoyaumeUni	9.87	19.87	44.36	101.73	209.67	780.41	1638.14	3609	7633
Canada	9.84	20.17	44.44	103.68	211.71	793.96	1656.01	3650	7809
Chine	10.17	20.54	45.25	106.44	216.49	805.14	1670.00	3635	7695
Croatie	10.25	20.76	45.64	104.07	213.30	817.76	1704.32	3827	8225
Ethiopie	10.50	21.08	45.89	106.08	211.13	757.35	1577.53	3535	7439
France	9.99	20.16	44.46	103.15	208.98	778.83	1642.78	3658	7596
Allemagne	10.06	20.20	44.33	103.65	211.58	774.70	1641.53	3634	7727
Inde	10.30	20.73	45.48	105.77	218.00	809.70	1682.89	3672	7920
Iran	10.29	21.11	46.37	104.74	218.80	833.40	1762.65	4103	8903
Italie	10.01	19.72	45.19	103.17	212.78	785.59	1636.50	3620	7642
Jamaïque	9.58	19.19	44.49	105.21	219.19	813.10	1712.44	3816	8199
Japon	10.00	20.03	44.78	106.18	217.42	793.20	1655.09	3625	7576
Kenya	10.26	20.43	44.18	102.01	206.34	759.74	1587.85	3513	7467
Lituanie	10.33	20.88	45.73	106.64	220.90	797.90	1651.50	3851	7955
NouvelleZélande	10.11	20.42	46.09	104.30	212.17	790.19	1661.95	3732	7815
Portugal	9.86	20.01	46.11	104.91	210.07	782.86	1632.47	3665	7596
Russie	10.10	20.23	44.60	102.47	212.28	791.99	1673.12	3675	7747
AfriqueduSud	10.06	20.11	44.59	102.69	213.56	794.16	1649.94	3678	7593
Espagne	10.14	20.59	44.96	103.83	208.95	782.54	1634.44	3592	7562
Suède	10.18	20.30	44.56	105.54	216.49	797.59	1675.74	3655	7838
Suisse	10.16	20.41	44.99	102.55	211.75	787.54	1673.16	3686	7643
Ukraine	10.07	20.00	45.11	105.08	210.33	790.78	1679.80	3711	7635
USA	9.69	19.32	43.18	102.60	209.30	776.27	1633.98	3583	7538

Comment faire un classement général entre les pays ???

→ Somme pondérée des scores, puis classement en fonction du rang.

# 5) Réduction de dimension par ACP

Records nationaux (en secondes) de  $p = 9$  épreuves d'athlétisme pour  $n = 26$  pays

	100m	200m	400m	800m	1500m	5000m	10000m	SemiMarathon	Marathon
Australie	9.93	20.06	44.38	104.40	211.96	775.76	1649.73	3602	7671
Belgique	10.02	20.19	44.78	103.86	214.13	769.71	1612.30	3605	7640
Brésil	10.00	19.89	44.29	101.77	213.25	799.43	1648.12	3573	7565
RoyaumeUni	9.87	19.87	44.36	101.73	209.67	780.41	1638.14	3609	7633
Canada	9.84	20.17	44.44	103.68	211.71	793.96	1656.01	3650	7809
Chine	10.17	20.54	45.25	106.44	216.49	805.14	1670.00	3635	7695
Croatie	10.25	20.76	45.64	104.07	213.30	817.76	1704.32	3827	8225
Ethiopie	10.50	21.08	45.89	106.08	211.13	757.35	1577.53	3535	7439
France	9.99	20.16	44.46	103.15	208.98	778.83	1642.78	3658	7596
Allemagne	10.06	20.20	44.33	103.65	211.58	774.70	1641.53	3634	7727
Inde	10.30	20.73	45.48	105.77	218.00	809.70	1682.89	3672	7920
Iran	10.29	21.11	46.37	104.74	218.80	833.40	1762.65	4103	8903
Italie	10.01	19.72	45.19	103.17	212.78	785.59	1636.50	3620	7642
Jamaïque	9.58	19.19	44.49	105.21	219.19	813.10	1712.44	3816	8199
Japon	10.00	20.03	44.78	106.18	217.42	793.20	1655.09	3625	7576
Kenya	10.26	20.43	44.18	102.01	206.34	759.74	1587.85	3513	7467
Lituanie	10.33	20.88	45.73	106.64	220.90	797.90	1651.50	3851	7955
NouvelleZélande	10.11	20.42	46.09	104.30	212.17	790.19	1661.95	3732	7815
Portugal	9.86	20.01	46.11	104.91	210.07	782.86	1632.47	3665	7596
Russie	10.10	20.23	44.60	102.47	212.28	791.99	1673.12	3675	7747
AfriqueduSud	10.06	20.11	44.59	102.69	213.56	794.16	1649.94	3678	7593
Espagne	10.14	20.59	44.96	103.83	208.95	782.54	1634.44	3592	7562
Suède	10.18	20.30	44.56	105.54	216.49	797.59	1675.74	3655	7838
Suisse	10.16	20.41	44.99	102.55	211.75	787.54	1673.16	3686	7643
Ukraine	10.07	20.00	45.11	105.08	210.33	790.78	1679.80	3711	7635
USA	9.69	19.32	43.18	102.60	209.30	776.27	1633.98	3583	7538

→ Matrice  $M$

Comment faire un classement général entre les pays ???

→ Somme pondérée des scores, puis classement en fonction du rang.

Somme pondérée des scores est équivalente à une multiplication matrice x vecteur :

→ Vecteur contenant les scores =  $M \cdot w$

## 5) Réduction de dimension par ACP

Records nationaux (en secondes) de  $p = 9$  épreuves d'athlétisme pour  $n = 26$  pays

	100m	200m	400m	800m	1500m	5000m	10000m	SemiMarathon	Marathon
Australie	9.93	20.06	44.38	104.40	211.96	775.76	1649.73	3602	7671
Belgique	10.02	20.19	44.78	103.86	214.13	769.71	1612.30	3605	7640
Brésil	10.00	19.89	44.29	101.77	213.25	799.43	1648.12	3573	7565
RoyaumeUni	9.87	19.87	44.36	101.73	209.67	780.41	1638.14	3609	7633
Canada	9.84	20.17	44.44	103.68	211.71	793.96	1656.01	3650	7809
Chine	10.17	20.54	45.25	106.44	216.49	805.14	1670.00	3635	7695
Croatie	10.25	20.76	45.64	104.07	213.30	817.76	1704.32	3827	8225
Ethiopie	10.50	21.08	45.89	106.08	211.13	757.35	1577.53	3535	7439
France	9.99	20.16	44.46	103.15	208.98	778.83	1642.78	3658	7596
Allemagne	10.06	20.20	44.33	103.65	211.58	774.70	1641.53	3634	7727
Inde	10.30	20.73	45.48	105.77	218.00	809.70	1682.89	3672	7920
Iran	10.29	21.11	46.37	104.74	218.80	833.40	1762.65	4103	8903
Italie	10.01	19.72	45.19	103.17	212.78	785.59	1636.50	3620	7642
Jamaïque	9.58	19.19	44.49	105.21	219.19	813.10	1712.44	3816	8199
Japon	10.00	20.03	44.78	106.18	217.42	793.20	1655.09	3625	7576
Kenya	10.26	20.43	44.18	102.01	206.34	759.74	1587.85	3513	7467
Lituanie	10.33	20.88	45.73	106.64	220.90	797.90	1651.50	3851	7955
NouvelleZélande	10.11	20.42	46.09	104.30	212.17	790.19	1661.95	3732	7815
Portugal	9.86	20.01	46.11	104.91	210.07	782.86	1632.47	3665	7596
Russie	10.10	20.23	44.60	102.47	212.28	791.99	1673.12	3675	7747
AfriqueduSud	10.06	20.11	44.59	102.69	213.56	794.16	1649.94	3678	7593
Espagne	10.14	20.59	44.96	103.83	208.95	782.54	1634.44	3592	7562
Suède	10.18	20.30	44.56	105.54	216.49	797.59	1675.74	3655	7838
Suisse	10.16	20.41	44.99	102.55	211.75	787.54	1673.16	3686	7643
Ukraine	10.07	20.00	45.11	105.08	210.33	790.78	1679.80	3711	7635
USA	9.69	19.32	43.18	102.60	209.30	776.27	1633.98	3583	7538

→ Matrice  $M$

On peut aussi chercher le vecteur (de norme 1) qui maximise la variabilité entre les scores

→ Vecteur optimal = 1<sup>er</sup> vecteur propre ( $\mathbf{v}_1$ ) de l'ACP

→ Niveau de variabilité = 1<sup>ere</sup> valeur propre ( $\lambda_1$ ) de l'ACP

→ Vecteur de scores avec la plus grande variabilité possible =  $M \cdot \mathbf{v}_1$

# 5) Réduction de dimension par ACP

Records nationaux (en secondes) de  $p = 9$  épreuves d'athlétisme pour  $n = 26$  pays

	100m	200m	400m	800m	1500m	5000m	10000m	SemiMarathon	Marathon
Australie	9.93	20.06	44.38	104.40	211.96	775.76	1649.73	3602	7671
Belgique	10.02	20.19	44.78	103.86	214.13	769.71	1612.30	3605	7640
Brésil	10.00	19.89	44.29	101.77	213.25	799.43	1648.12	3573	7565
RoyaumeUni	9.87	19.87	44.36	101.73	209.67	780.41	1638.14	3609	7633
Canada	9.84	20.17	44.44	103.68	211.71	793.96	1656.01	3650	7809
Chine	10.17	20.54	45.25	106.44	216.49	805.14	1670.00	3635	7695
Croatie	10.25	20.76	45.64	104.07	213.30	817.76	1704.32	3827	8225
Ethiopie	10.50	21.08	45.89	106.08	211.13	757.35	1577.53	3535	7439
France	9.99	20.16	44.46	103.15	208.98	778.83	1642.78	3658	7596
Allemagne	10.06	20.20	44.33	103.65	211.58	774.70	1641.53	3634	7727
Inde	10.30	20.73	45.48	105.77	218.00	809.70	1682.89	3672	7920
Iran	10.29	21.11	46.37	104.74	218.80	833.40	1762.65	4103	8903
Italie	10.01	19.72	45.19	103.17	212.78	785.59	1636.50	3620	7642
Jamaïque	9.58	19.19	44.49	105.21	219.19	813.10	1712.44	3816	8199
Japon	10.00	20.03	44.78	106.18	217.42	793.20	1655.09	3625	7576
Kenya	10.26	20.43	44.18	102.01	206.34	759.74	1587.85	3513	7467
Lituanie	10.33	20.88	45.73	106.64	220.90	797.90	1651.50	3851	7955
NouvelleZélande	10.11	20.42	46.09	104.30	212.17	790.19	1661.95	3732	7815
Portugal	9.86	20.01	46.11	104.91	210.07	782.86	1632.47	3665	7596
Russie	10.10	20.23	44.60	102.47	212.28	791.99	1673.12	3675	7747
AfriqueduSud	10.06	20.11	44.59	102.69	213.56	794.16	1649.94	3678	7593
Espagne	10.14	20.59	44.96	103.83	208.95	782.54	1634.44	3592	7562
Suède	10.18	20.30	44.56	105.54	216.49	797.59	1675.74	3655	7838
Suisse	10.16	20.41	44.99	102.55	211.75	787.54	1673.16	3686	7643
Ukraine	10.07	20.00	45.11	105.08	210.33	790.78	1679.80	3711	7635
USA	9.69	19.32	43.18	102.60	209.30	776.27	1633.98	3583	7538

→ Matrice  $M$

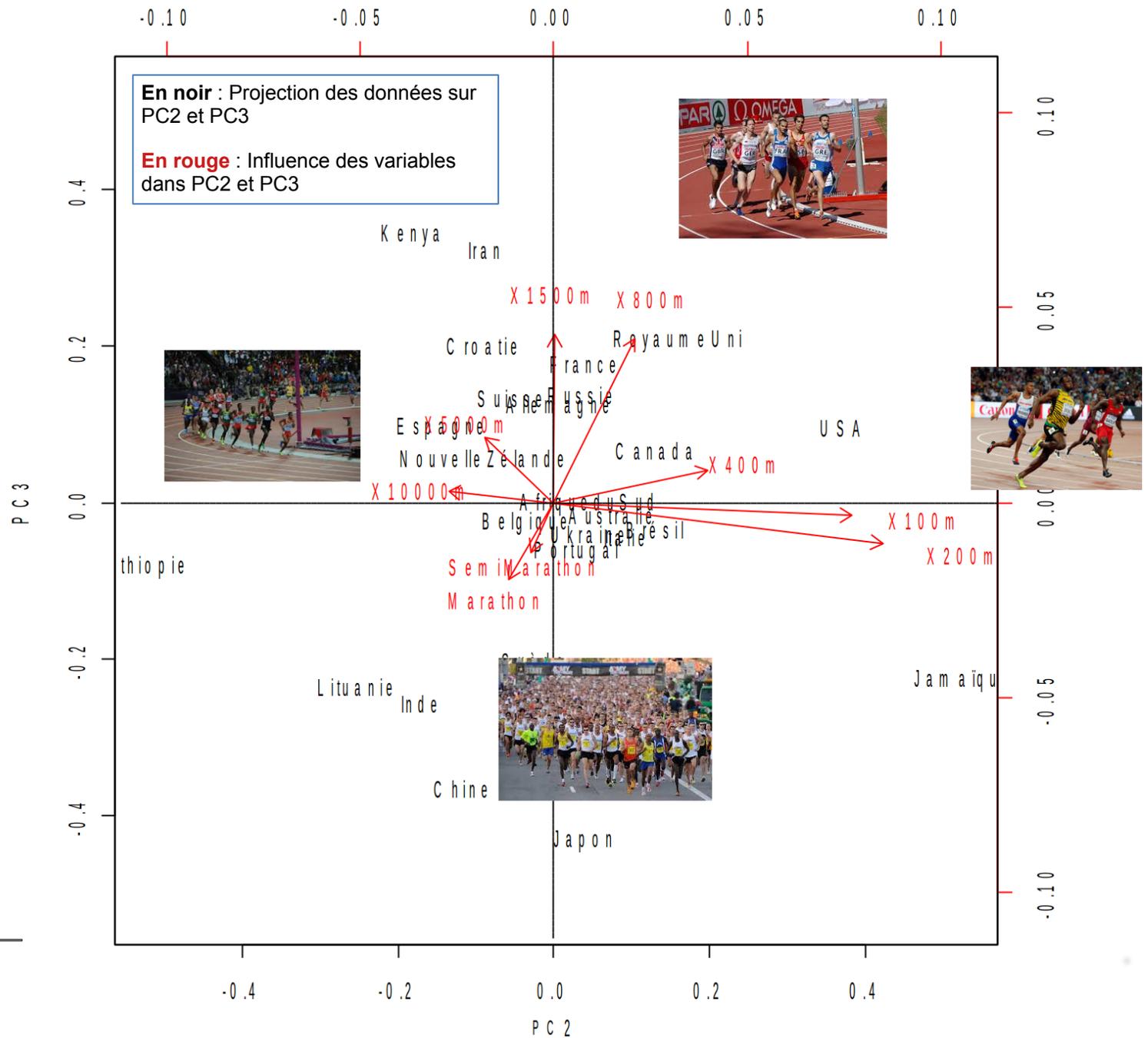
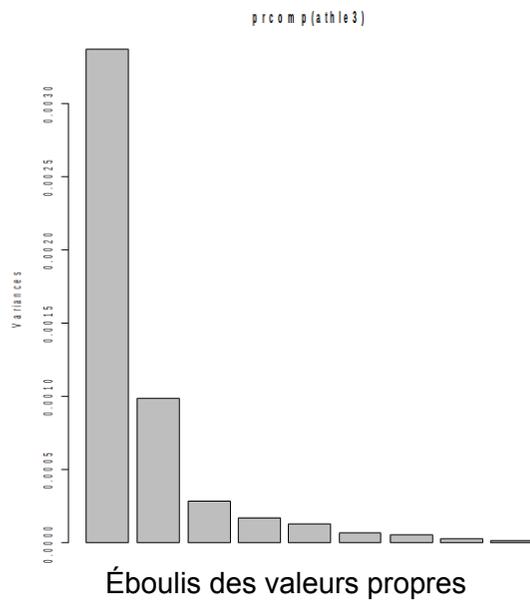
Une fois enlevée l'influence de  $v_1$ , on cherche le vecteur (de norme 1) qui maximise la variabilité

- Vecteur optimal = 2<sup>er</sup> vecteur propre ( $v_2$ ) de l'ACP
- Niveau de variabilité = 2<sup>ere</sup> valeur propre ( $\lambda_2$ ) de l'ACP

...

Calculable de manière analytique

# 5) Réduction de dimension par ACP



# Apprentissage machine et GPU

# 6.a) Apprentissage machine et GPU — Introduction

Deep learning

Caffe

K Keras

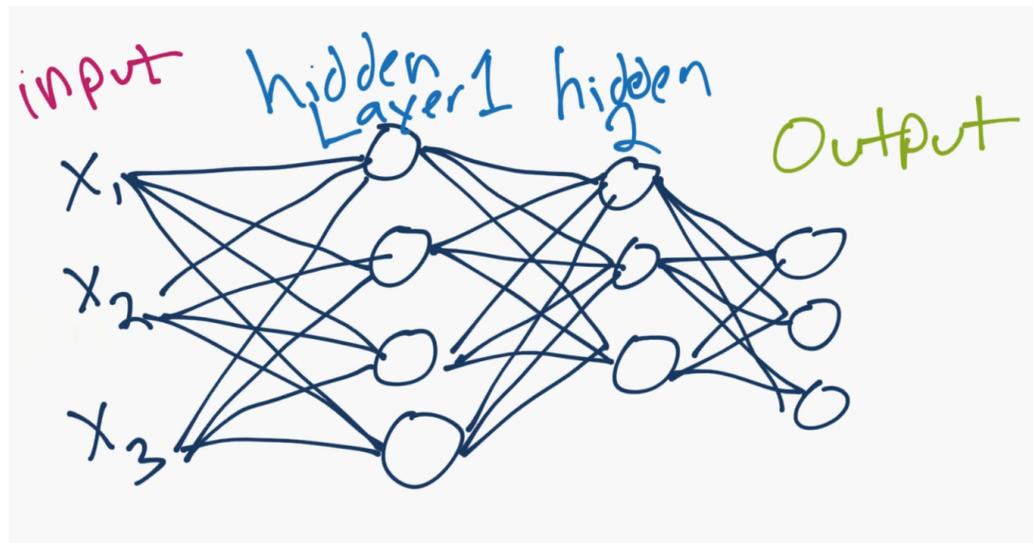
theano

PYTORCH



...

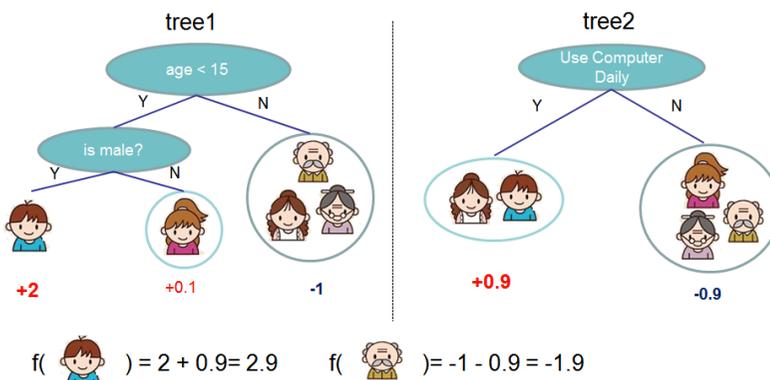
Connu Appris Connu



- Très efficace sur certains problèmes (signaux, images).
- Apprentissage nécessitant du calcul intensif mais prédiction rapide.
- Dimension du problème d'apprentissage très élevée.
- Nécessité d'avoir énormément de données annotées ou une structure de graphe adaptée aux données.

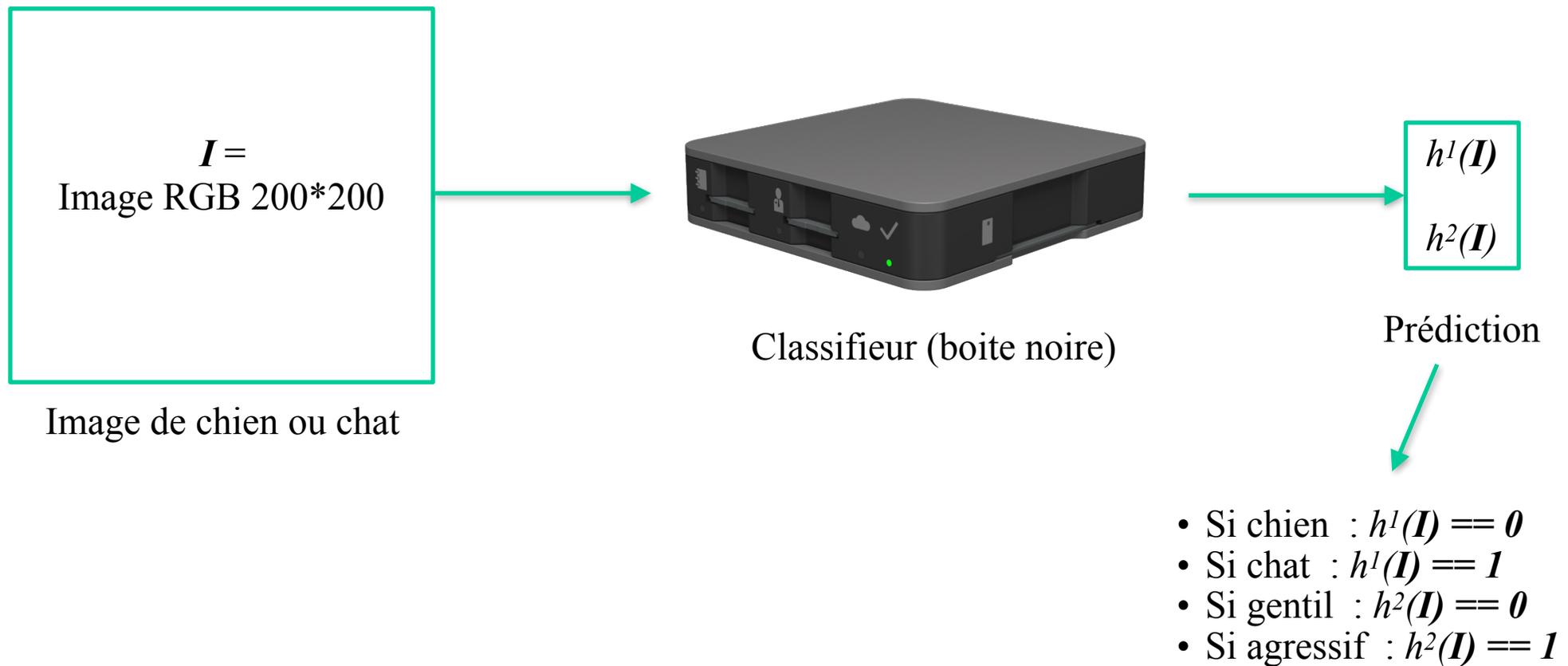
<https://pythonprogramming.net/neural-networks-machine-learning-tutorial/>

XGBoost XGBoost



- Méthode basée sur des arbres de décision.
- Très efficace dans le cas général.
- Paramètre à gérer (contrairement à Random Forest) et gros coût calculatoire.

## Prédiction



### Prédiction



Image de chien ou chat



Classifieur (boite noire)



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

- Si chien :  $h^1(\mathbf{I}) == 0$
- Si chat :  $h^1(\mathbf{I}) == 1$
- Si gentil :  $h^2(\mathbf{I}) == 0$
- Si agressif :  $h^2(\mathbf{I}) == 1$

### Prédiction



Image de chien ou chat



Classifieur (boite noire)



$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$

- Si chien :  $h^1(\mathbf{I}) == 0$
- Si chat :  $h^1(\mathbf{I}) == 1$
- Si gentil :  $h^2(\mathbf{I}) == 0$
- Si agressif :  $h^2(\mathbf{I}) == 1$

### Prédiction

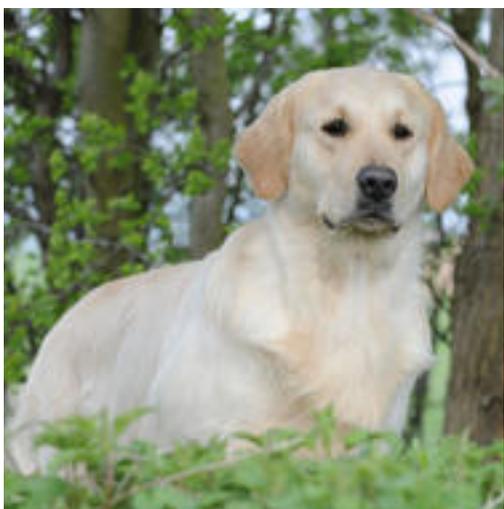


Image de chien ou chat



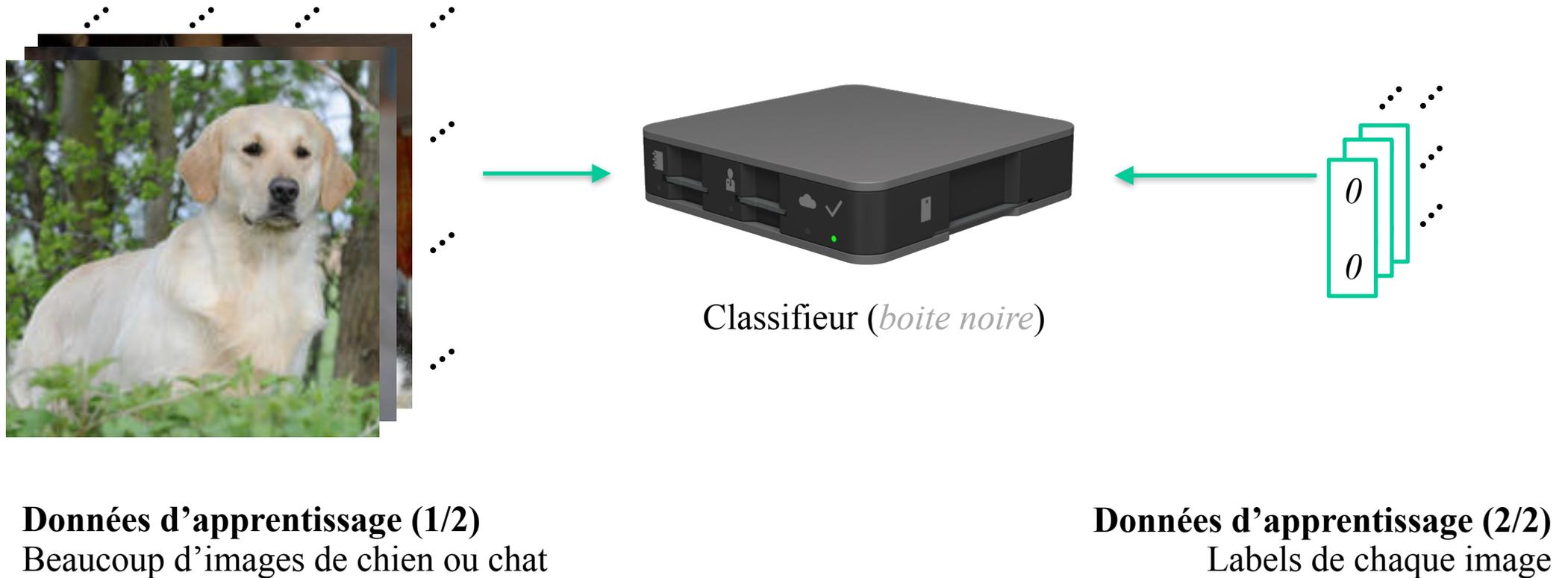
Classifieur (boîte noire)



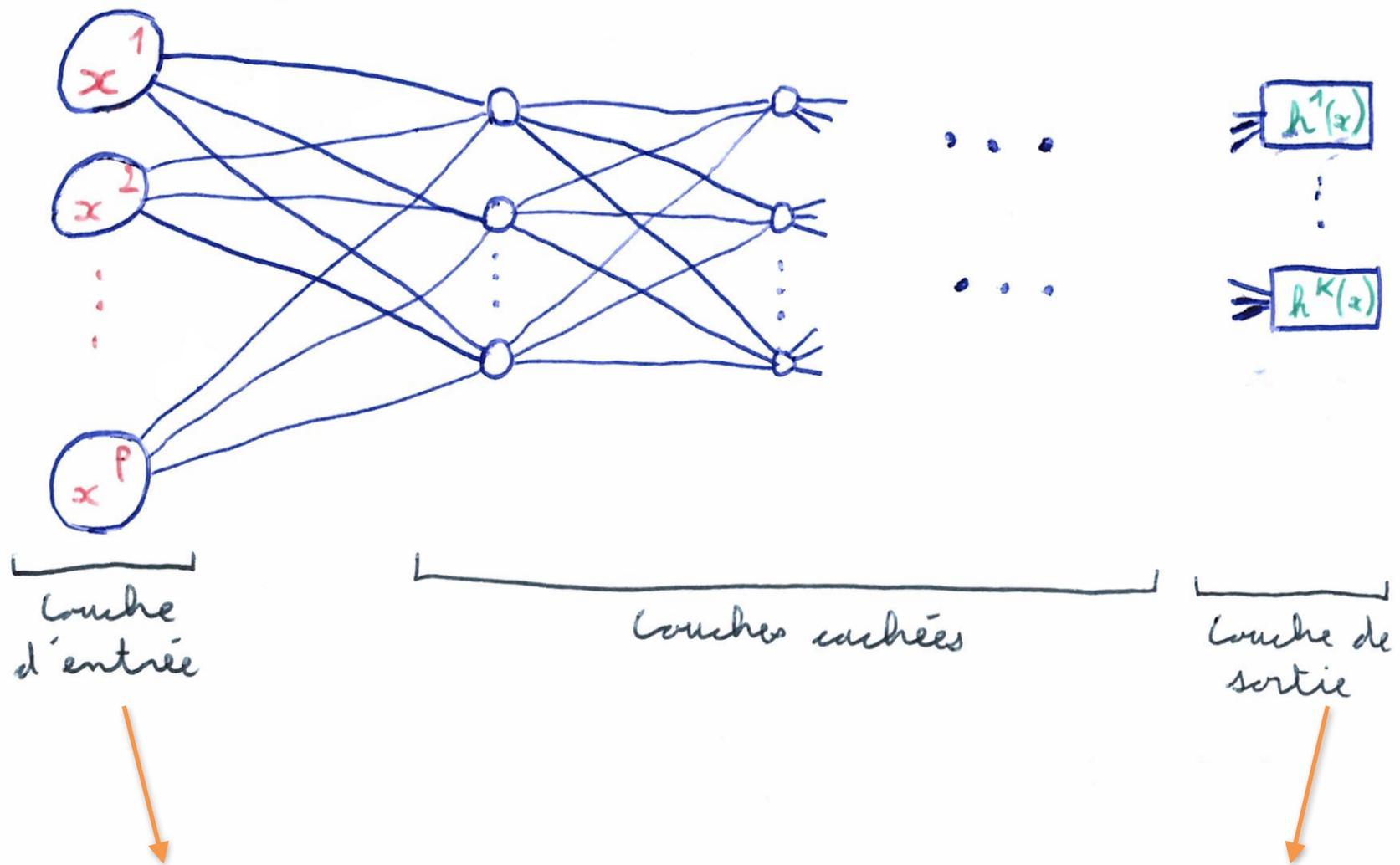
$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$

- Si chien :  $h^1(\mathbf{I}) == 0$
- Si chat :  $h^1(\mathbf{I}) == 1$
- Si gentil :  $h^2(\mathbf{I}) == 0$
- Si agressif :  $h^2(\mathbf{I}) == 1$

**Apprentissage** : Optimisation des paramètres pour obtenir les meilleures prédictions en moyenne



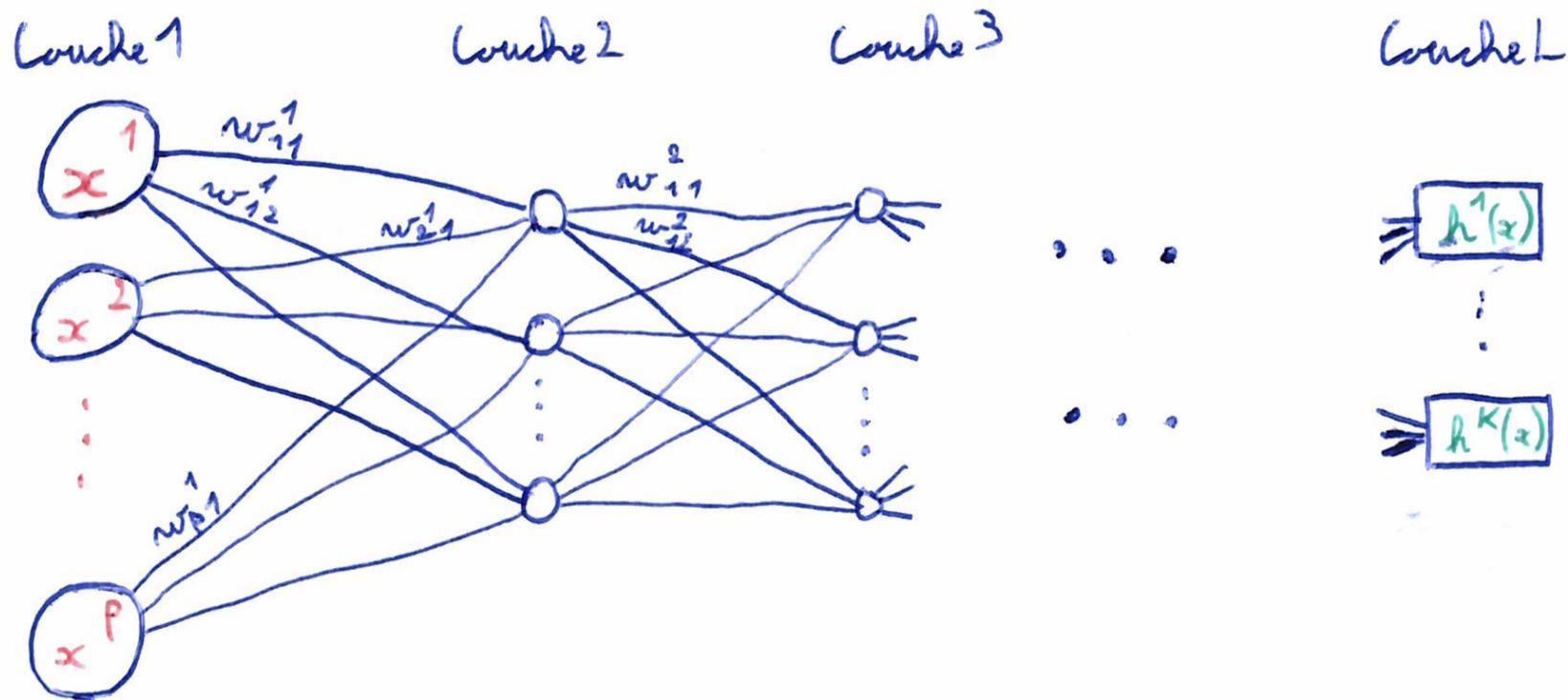
## 6.b) Apprentissage machine et GPU — Deep learning



Les  $x^i$  peuvent être les intensités de l'image  $I$  sur chaque canal RGB

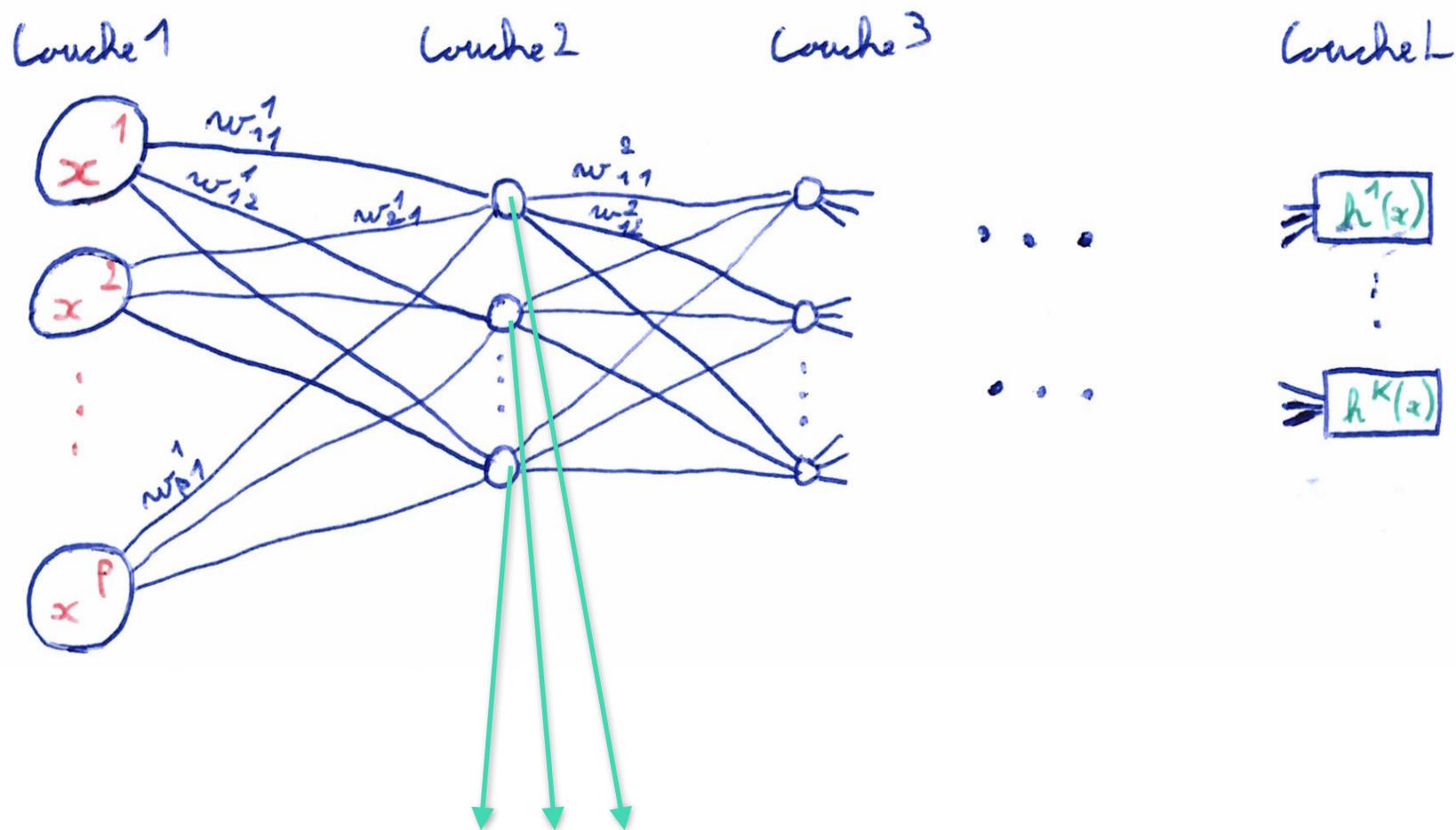
Labels prédits

# 6.b) Apprentissage machine et GPU — Deep learning



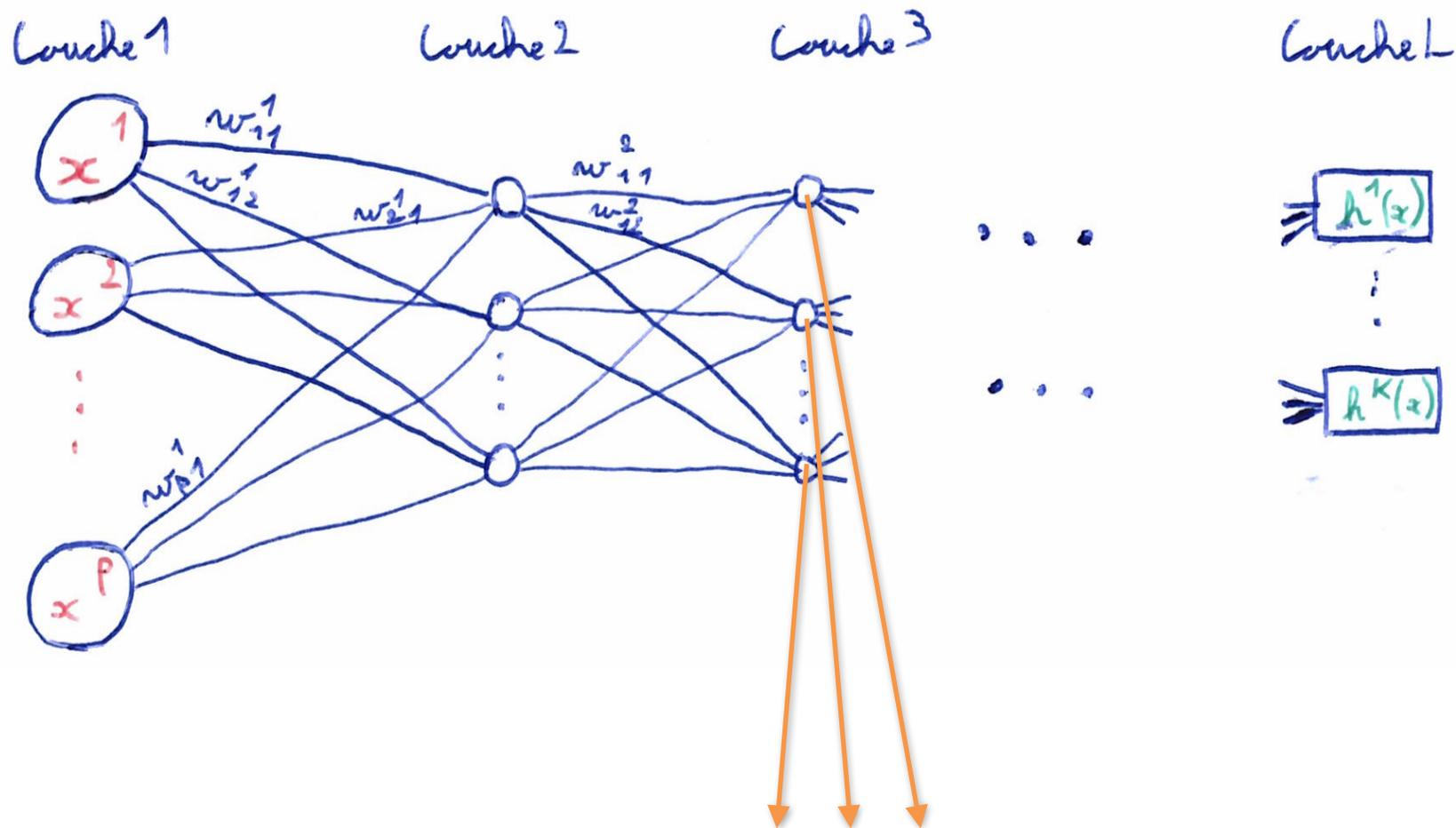
$$\sigma_k^l = f(\sigma_k^l) = f\left(\sum_{p \in N_{l-1}} w_{pk}^{l-1} \sigma_p^{l-1}\right)$$

# 6.b) Apprentissage machine et GPU — Deep learning



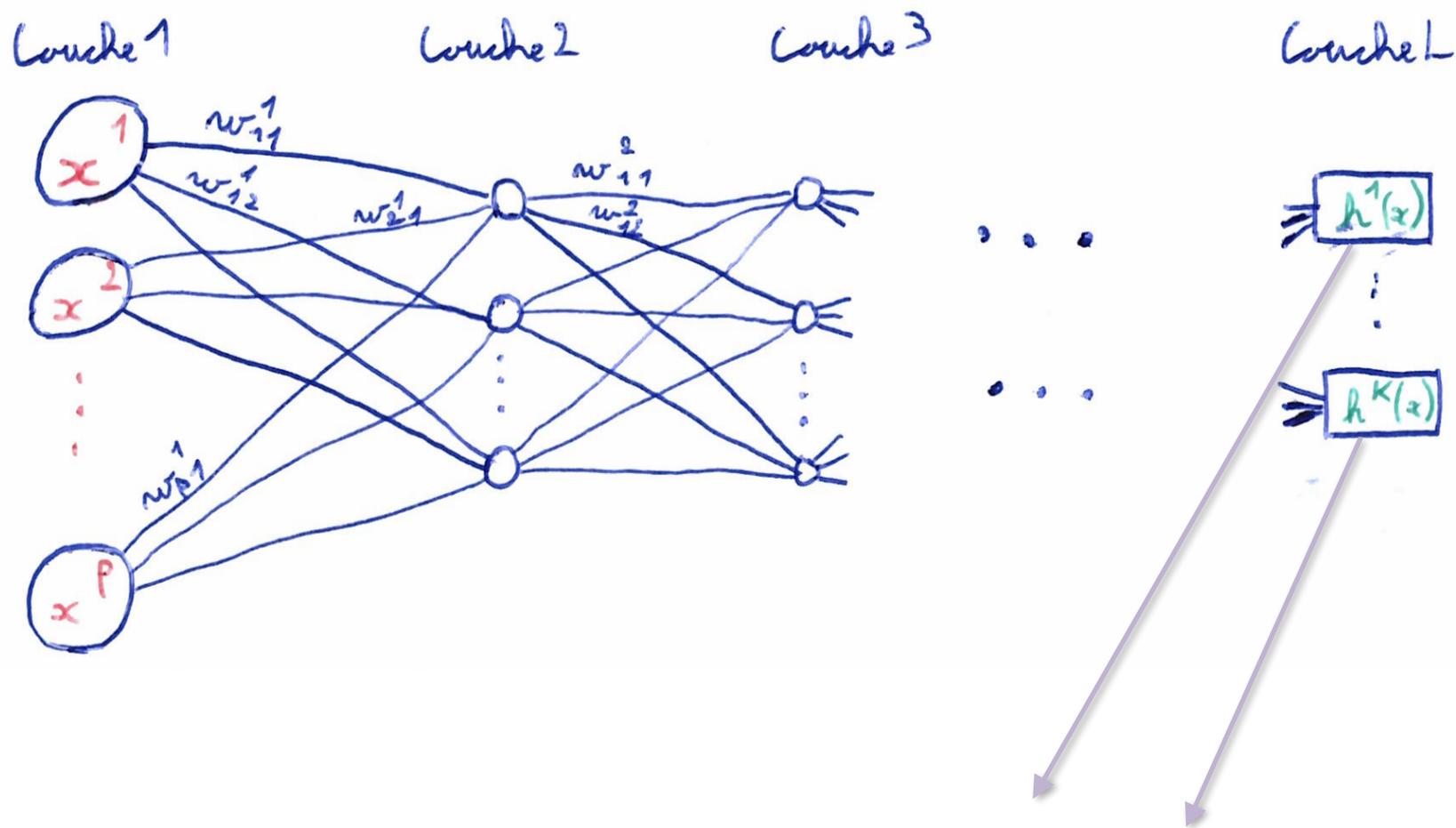
$$\sigma_k^l = f(\sigma_k^l) = f\left(\sum_{P \in N_{l-1}} w_{Pk}^{l-1} \sigma_P^{l-1}\right)$$

# 6.b) Apprentissage machine et GPU — Deep learning



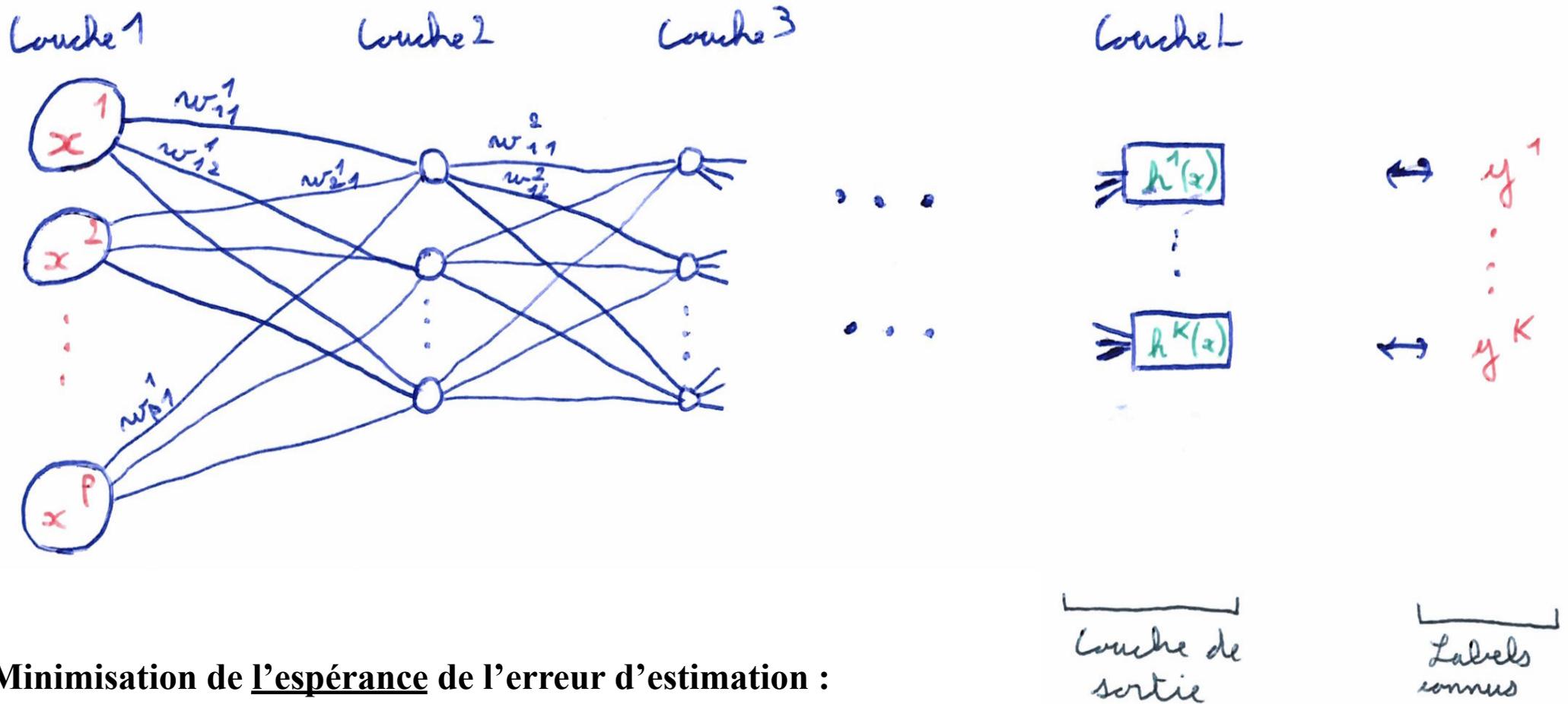
$$\sigma_k^l = f(\sigma_k^l) = f\left(\sum_{p \in N_{l-1}} w_{pk}^{l-1} \sigma_p^{l-1}\right)$$

# 6.b) Apprentissage machine et GPU — Deep learning



$$\sigma_k^l = f(\sigma_k^l) = f\left(\sum_{P \in N_{l-1}} w_{Pk}^{l-1} \sigma_P^{l-1}\right)$$

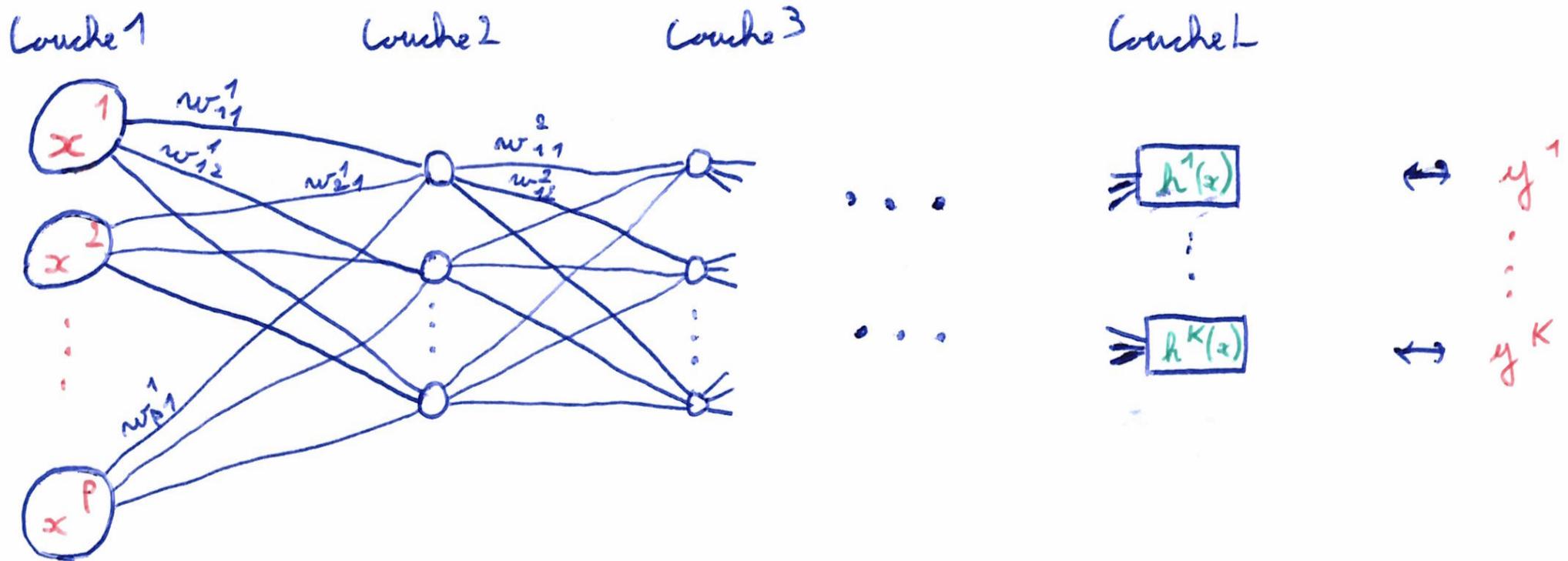
# 6.b) Apprentissage machine et GPU — Deep learning



Minimisation de l'espérance de l'erreur d'estimation :

$$E(\underbrace{x^1, \dots, x^P}_x) = \sum_{k=1}^K (h^k(x) - y^k)^2$$

## 6.b) Apprentissage machine et GPU — Deep learning

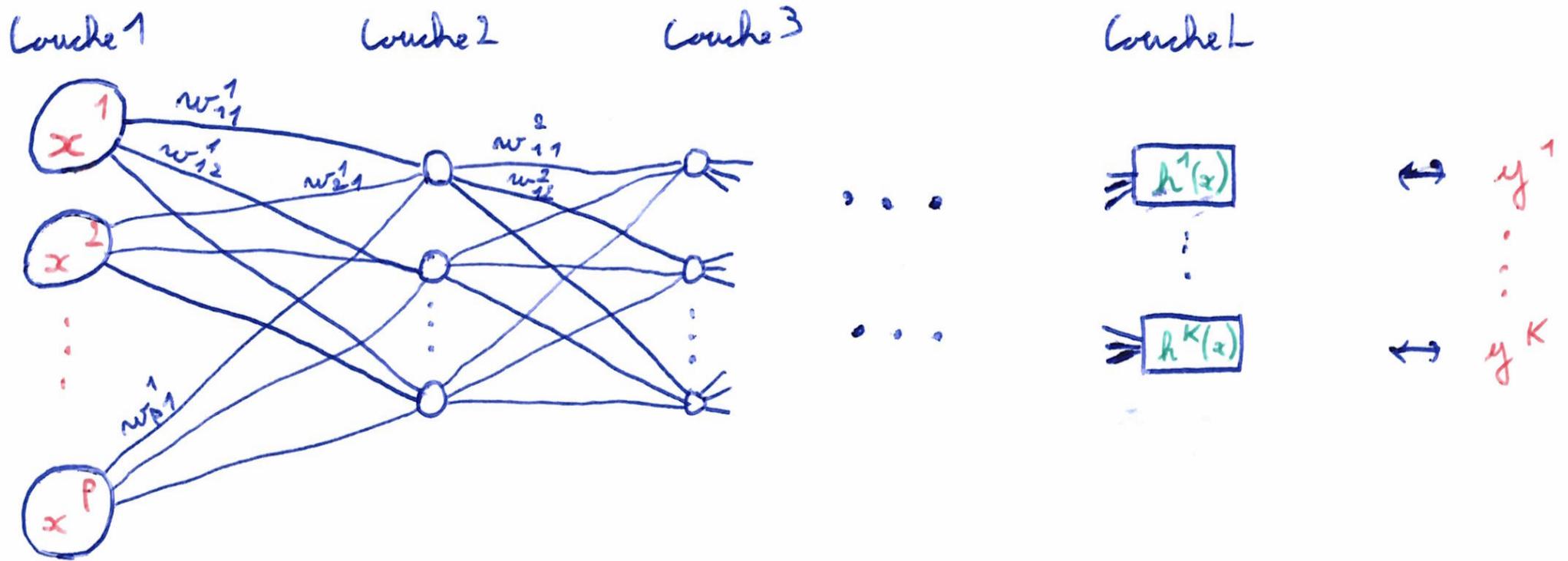


**Descente de gradient stochastique :**

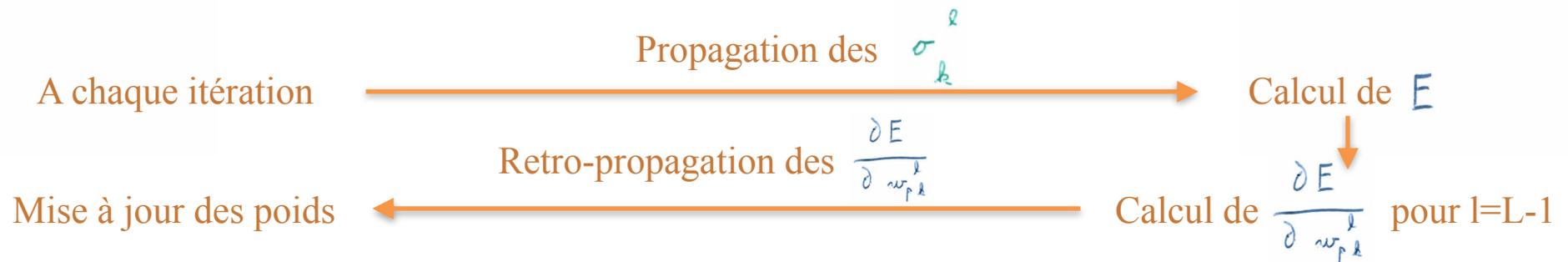
$$w_{pk}^l \leftarrow w_{pk}^l - \eta \frac{\partial E}{\partial w_{pk}^l} \quad \text{avec} \quad \frac{\partial E}{\partial w_{pk}^l}$$

- Facile à calculer analytiquement pour  $l=L-1$
- Calculable analytiquement pour  $l < L-1$  si connu à  $l+1$

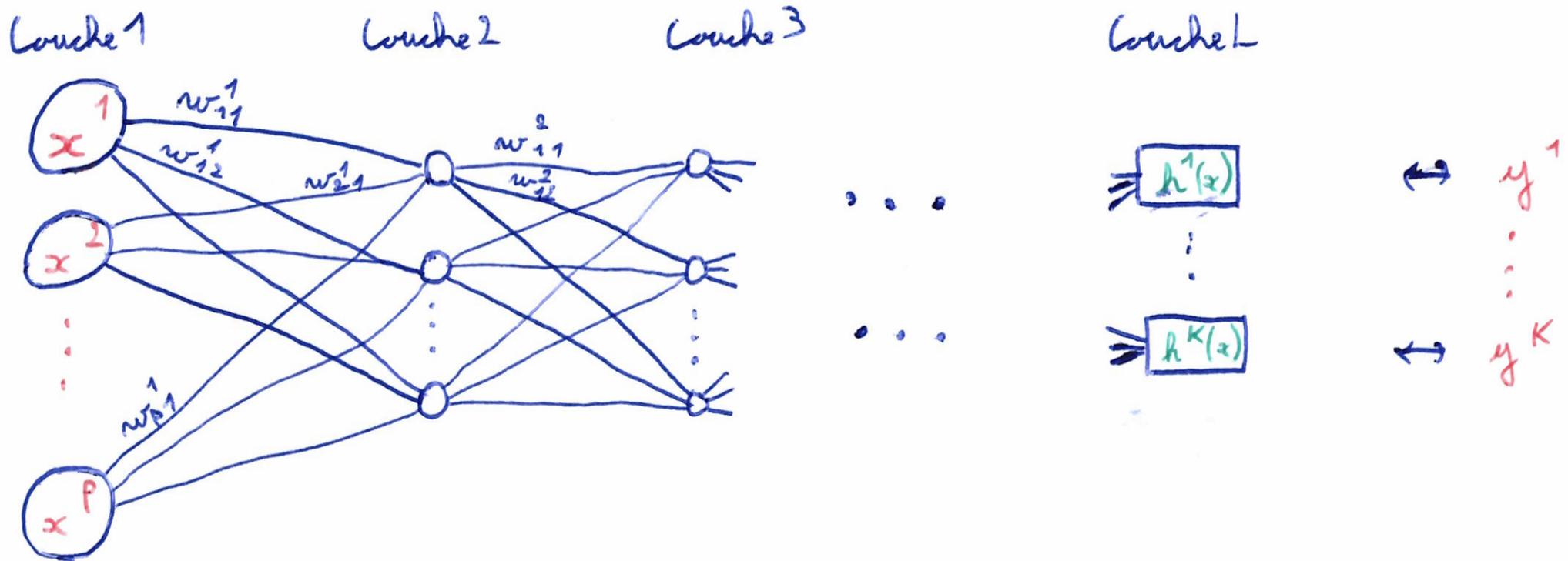
# 6.b) Apprentissage machine et GPU — Deep learning



## Descente de gradient stochastique :



## 6.b) Apprentissage machine et GPU — Deep learning



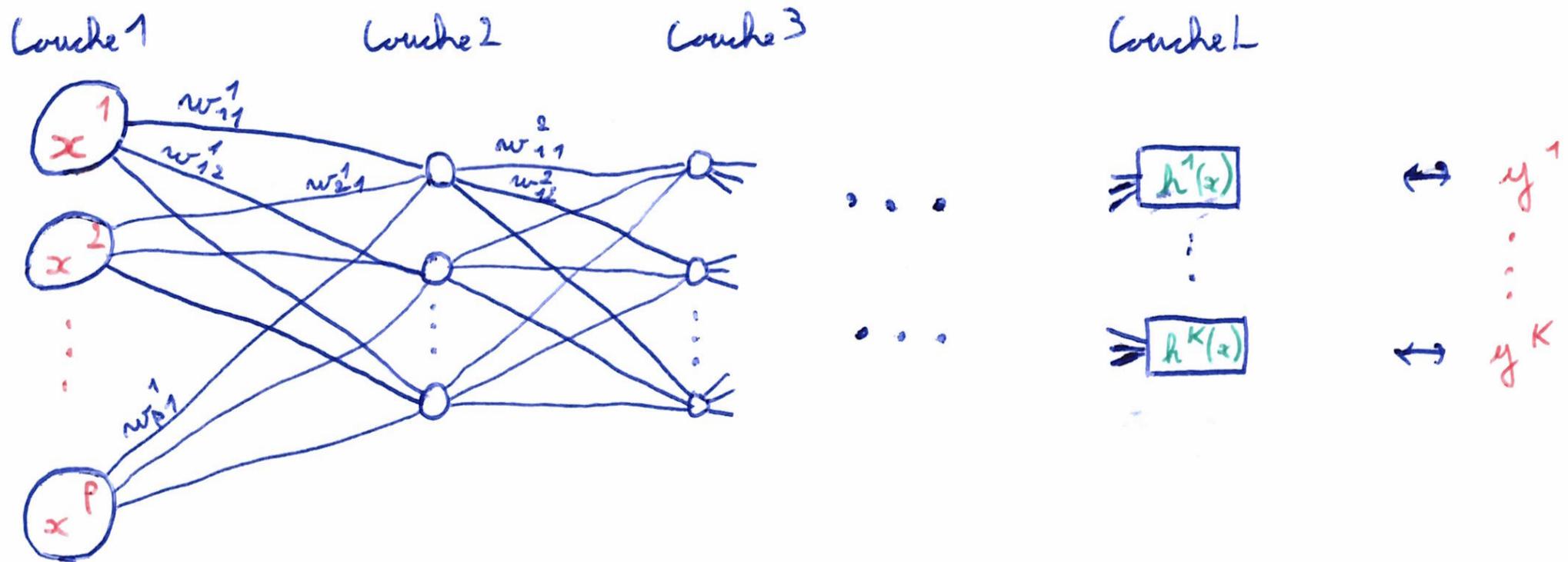
### En pratique :

- Différents types de couches
- Différents types d'architecture
- Différentes variantes de descente de gradient stochastique

### Dans tous les cas :

- Prédiction et apprentissage particulièrement simples à paralléliser sur GPU
- Librairie *Nvidia cuDNN* massivement utilisée et sous jacente à Caffe, Keras, TensorFlow, Theano...

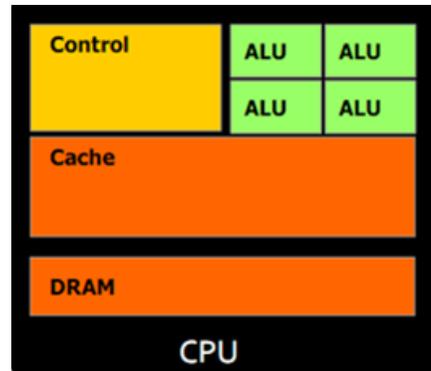
## 6.b) Apprentissage machine et GPU — Deep learning



Lien avec le GPU ???

```
#Create a simple model
import tensorflow as tf
x = tf.placeholder(tf.float32, [None, 784])
W = tf.Variable(tf.zeros([784, 10]))
y = tf.matmul(x, W)
```

## 6.c) Apprentissage machine et GPU — GPU



<http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU>

**DRAM** : Dynamic Random Access Memory (mémoire classique)

**Cache** : Mémoire (ici) interne au processeur. Rapide mais de taille limité

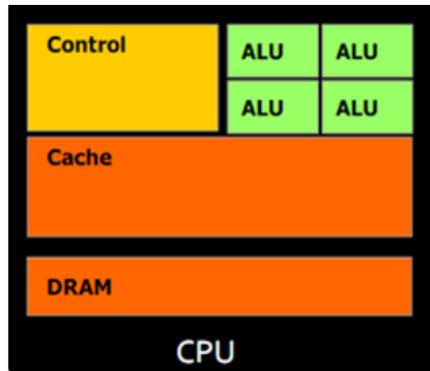
**ALU** : Arithmetic-Logic Unit. Effectue les calculs.

**Control** : Coordonne les ALU et la mémoire

### Illustration : Multiplication matrice / vecteur

$$\begin{pmatrix} 1 & 3 & \dots & \dots & -2 \\ 2 & 1 & \dots & \dots & 0 \\ -1 & -2 & \dots & \dots & 0 \\ \vdots & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ 0 & 1 & \dots & \dots & 3 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 0 \\ \vdots \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} ? \\ ? \\ ? \\ \vdots \\ \vdots \\ ? \end{pmatrix}$$

# 6.c) Apprentissage machine et GPU — GPU



<http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU>

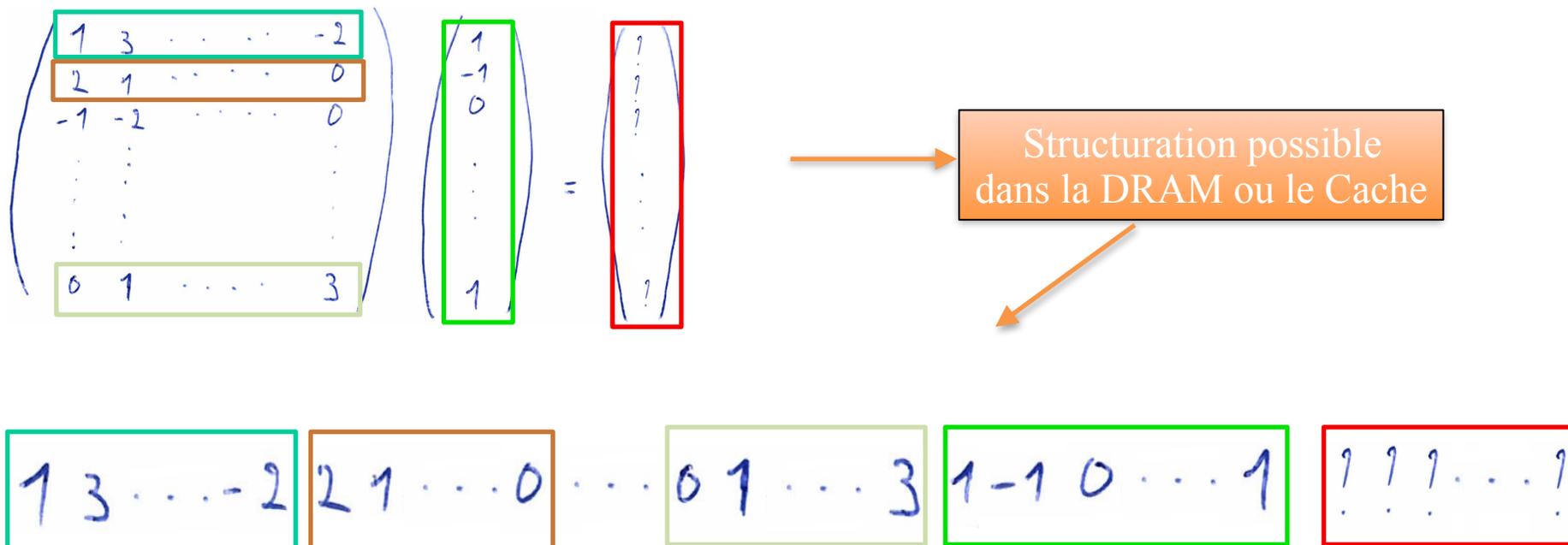
**DRAM** : Dynamic Random Access Memory (mémoire classique)

**Cache** : Mémoire (ici) interne au processeur. Rapide mais de taille limité

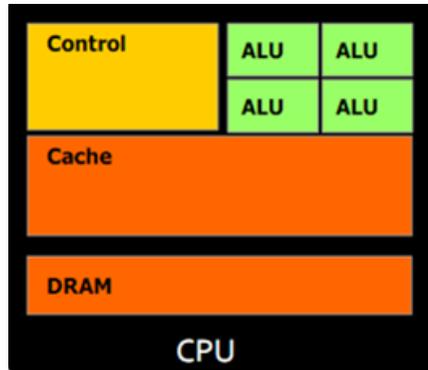
**ALU** : Arithmetic-Logic Unit. Effectue les calculs.

**Control** : Coordonne les ALU et la mémoire

## Illustration : Multiplication matrice / vecteur



## 6.c) Apprentissage machine et GPU — GPU



<http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU>

**DRAM** : Dynamic Random Access Memory (mémoire classique)

**Cache** : Mémoire (ici) interne au processeur. Rapide mais de taille limité

**ALU** : Arithmetic-Logic Unit. Effectue les calculs.

**Control** : Coordonne les ALU et la mémoire

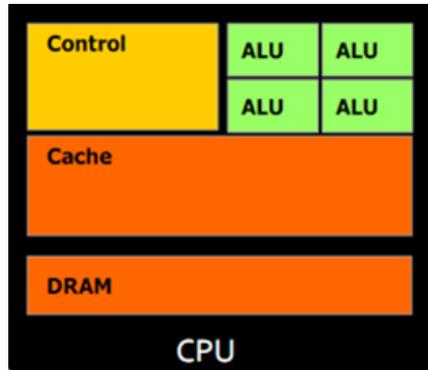
### Illustration : Multiplication matrice / vecteur

$$\begin{pmatrix} 1 & 3 & \dots & \dots & -2 \\ 2 & 1 & \dots & \dots & 0 \\ -1 & -2 & \dots & \dots & 0 \\ \vdots & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ 0 & 1 & \dots & \dots & 3 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 0 \\ \vdots \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} ? \\ ? \\ ? \\ \vdots \\ \vdots \\ ? \end{pmatrix}$$

Opérations + et \* sur différentes zones mémoires par le CPU

$$\boxed{1} \ 3 \ \dots \ -2 \ 2 \ 1 \ \dots \ 0 \ \dots \ 0 \ 1 \ \dots \ 3 \ \boxed{1} \ -1 \ 0 \ \dots \ 1 \ \boxed{?} \ ? \ ? \ \dots \ ?$$

# 6.c) Apprentissage machine et GPU — GPU



<http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU>

**DRAM** : Dynamic Random Access Memory (mémoire classique)

**Cache** : Mémoire (ici) interne au processeur. Rapide mais de taille limité

**ALU** : Arithmetic-Logic Unit. Effectue les calculs.

**Control** : Coordonne les ALU et la mémoire

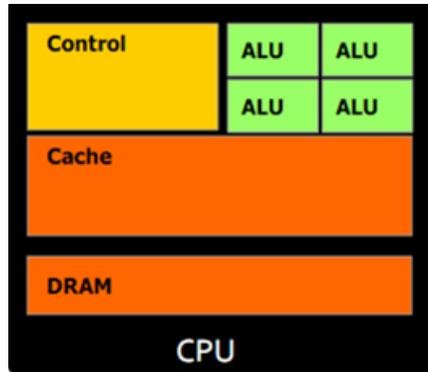
## Illustration : Multiplication matrice / vecteur

$$\begin{pmatrix} 1 & \boxed{3} & \dots & \dots & -2 \\ 2 & 1 & \dots & \dots & 0 \\ -1 & -2 & \dots & \dots & 0 \\ \vdots & \vdots & \dots & \dots & \vdots \\ \vdots & \vdots & \dots & \dots & \vdots \\ 0 & 1 & \dots & \dots & 3 \end{pmatrix} \begin{pmatrix} 1 \\ \boxed{-1} \\ 0 \\ \vdots \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} \boxed{?} \\ ? \\ ? \\ \vdots \\ \vdots \\ ? \end{pmatrix}$$

Opérations + et \* sur différentes zones mémoires par le CPU

$$1 \boxed{3} \dots -2 \quad 2 \quad 1 \dots 0 \dots 0 \quad 1 \dots 3 \quad 1 \boxed{-1} \quad 0 \dots 1 \quad \boxed{?} \quad ? \quad ? \dots ?$$

# 6.c) Apprentissage machine et GPU — GPU



<http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU>

**DRAM** : Dynamic Random Access Memory (mémoire classique)

**Cache** : Mémoire (ici) interne au processeur. Rapide mais de taille limité

**ALU** : Arithmetic-Logic Unit. Effectue les calculs.

**Control** : Coordonne les ALU et la mémoire

## Illustration : Multiplication matrice / vecteur

$$\begin{pmatrix} 1 & 3 & \dots & \dots & -2 \\ 2 & 1 & \dots & \dots & 0 \\ -1 & -2 & \dots & \dots & 0 \\ \vdots & \vdots & \dots & \dots & \vdots \\ \vdots & \vdots & \dots & \dots & \vdots \\ 0 & 1 & \dots & \dots & 3 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 0 \\ \vdots \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} ? \\ ? \\ ? \\ \vdots \\ \vdots \\ ? \end{pmatrix}$$

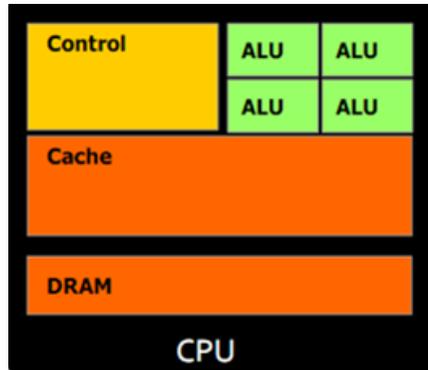
The diagram illustrates matrix-vector multiplication. The first matrix has a red box around the element -2 in the first row, fifth column. The second vector has a red box around the element 1 in the sixth row. The resulting vector has a red box around the first element, which is a question mark.

Opérations + et \* sur différentes zones mémoires par le CPU

$$1 \ 3 \ \dots \ -2 \ 2 \ 1 \ \dots \ 0 \ \dots \ 0 \ 1 \ \dots \ 3 \ 1 \ -1 \ 0 \ \dots \ 1 \ ? \ ? \ ? \ \dots \ ?$$

The diagram shows the dot product of the first row of the matrix and the vector. The elements -2, 1, and 1 are highlighted with red boxes. The result of the dot product is shown as a question mark.

# 6.c) Apprentissage machine et GPU — GPU



<http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU>

**DRAM** : Dynamic Random Access Memory (mémoire classique)

**Cache** : Mémoire (ici) interne au processeur. Rapide mais de taille limité

**ALU** : Arithmetic-Logic Unit. Effectue les calculs.

**Control** : Coordonne les ALU et la mémoire

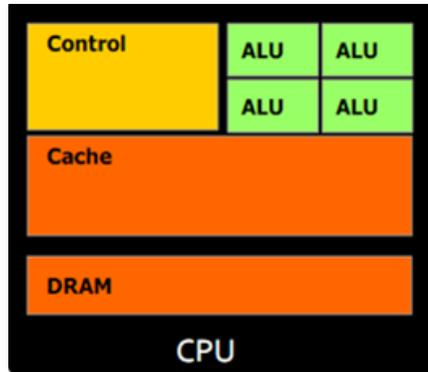
## Illustration : Multiplication matrice / vecteur

$$\begin{pmatrix} 1 & 3 & \dots & -2 \\ \boxed{2} & 1 & \dots & 0 \\ -1 & -2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \dots & 3 \end{pmatrix} \begin{pmatrix} \boxed{1} \\ -1 \\ 0 \\ \vdots \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} ? \\ \boxed{?} \\ \vdots \\ \vdots \\ \vdots \\ ? \end{pmatrix}$$

Opérations + et \* sur différentes zones mémoires par le CPU

$$1 \ 3 \ \dots \ -2 \ \boxed{2} \ 1 \ \dots \ 0 \ \dots \ 0 \ 1 \ \dots \ 3 \ \boxed{1} \ -1 \ 0 \ \dots \ 1 \ \boxed{?} \ \boxed{?} \ \dots \ ?$$

# 6.c) Apprentissage machine et GPU — GPU



<http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU>

**DRAM** : Dynamic Random Access Memory (mémoire classique)

**Cache** : Mémoire (ici) interne au processeur. Rapide mais de taille limité

**ALU** : Arithmetic-Logic Unit. Effectue les calculs.

**Control** : Coordonne les ALU et la mémoire

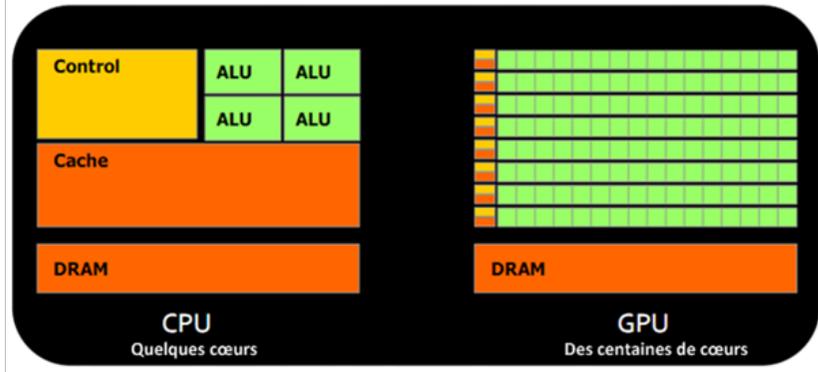
## Illustration : Multiplication matrice / vecteur

$$\begin{pmatrix} 1 & 3 & \dots & \dots & -2 \\ 2 & 1 & \dots & \dots & 0 \\ -1 & -2 & \dots & \dots & 0 \\ \vdots & \vdots & \dots & \dots & \vdots \\ \vdots & \vdots & \dots & \dots & \vdots \\ 0 & 1 & \dots & \dots & \boxed{3} \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 0 \\ \vdots \\ \vdots \\ \boxed{1} \end{pmatrix} = \begin{pmatrix} ? \\ ? \\ ? \\ \vdots \\ \vdots \\ \boxed{?} \end{pmatrix}$$

Opérations + et \* sur différentes zones mémoires par le CPU

$$1 \ 3 \ \dots \ -2 \ 2 \ 1 \ \dots \ 0 \ \dots \ 0 \ 1 \ \dots \ \boxed{3} \ 1 \ -1 \ 0 \ \dots \ \boxed{1} \ \ ? \ ? \ ? \ \dots \ \boxed{?}$$

# 6.c) Apprentissage machine et GPU — GPU



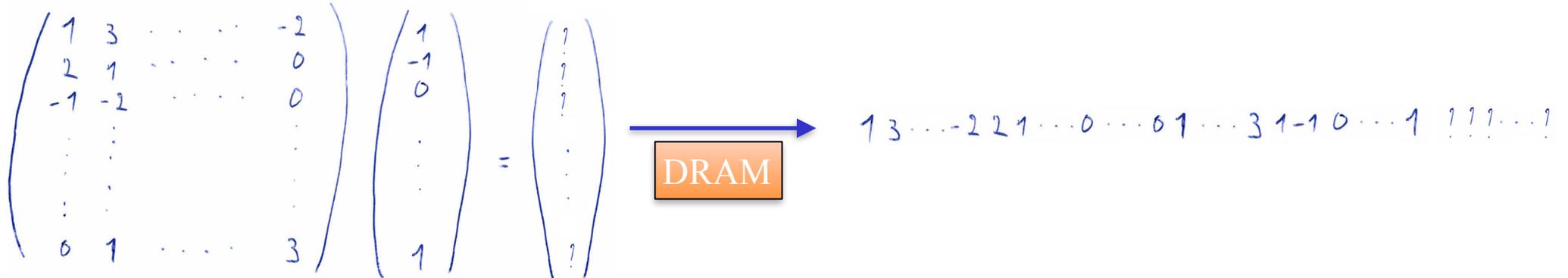
<http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU>

**DRAM** : Dynamic Random Access Memory (mémoire classique)

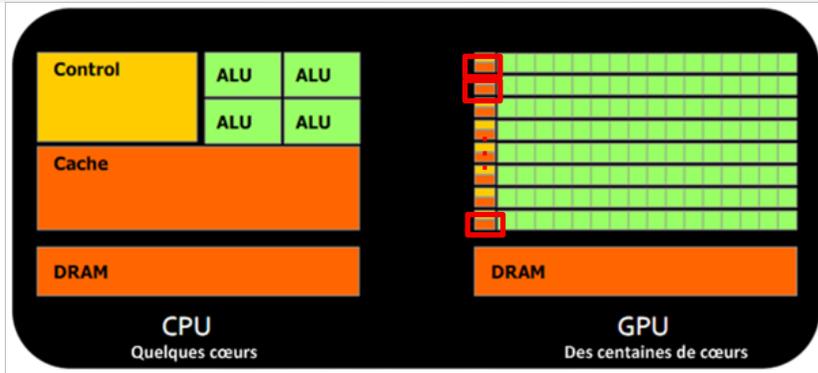
**Cache** : Mémoire (ici) interne au processeur. Rapide mais de taille limité

**ALU** : Arithmetic-Logic Unit. Effectue les calculs.

**Control** : Coordonne les ALU et la mémoire



# 6.c) Apprentissage machine et GPU — GPU



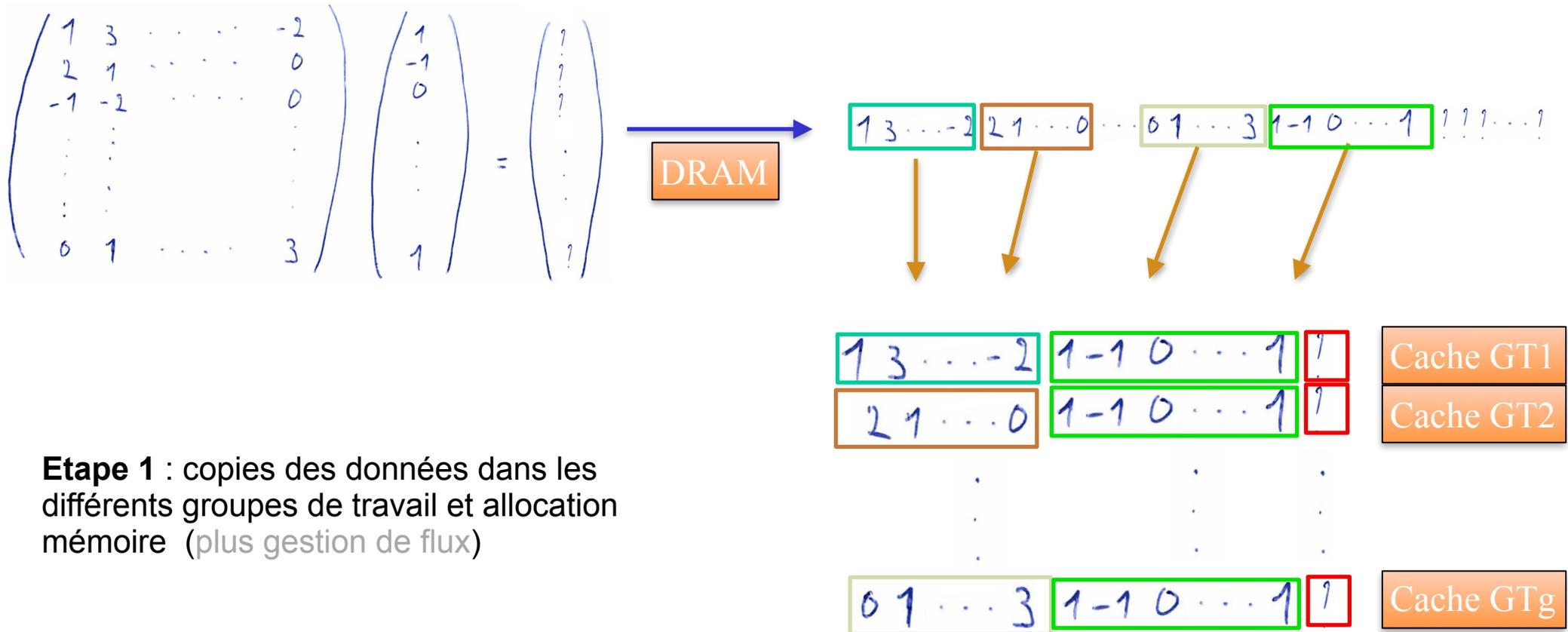
<http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU>

**DRAM** : Dynamic Random Access Memory (mémoire classique)

**Cache** : Mémoire (ici) interne au processeur. Rapide mais de taille limité

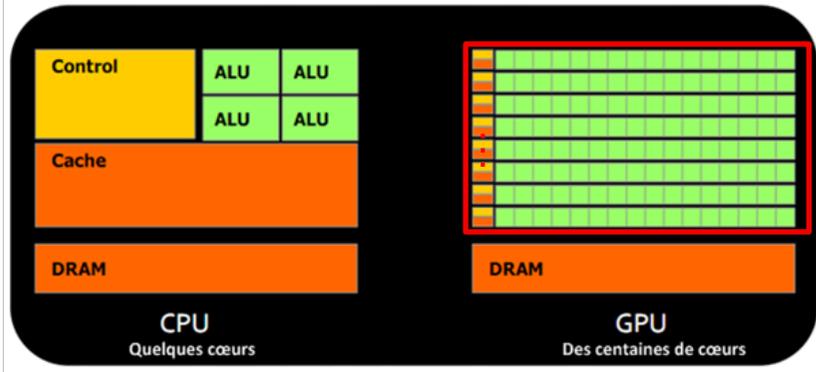
**ALU** : Arithmetic-Logic Unit. Effectue les calculs.

**Control** : Coordonne les ALU et la mémoire



**Etape 1** : copies des données dans les différents groupes de travail et allocation mémoire (plus gestion de flux)

# 6.c) Apprentissage machine et GPU — GPU



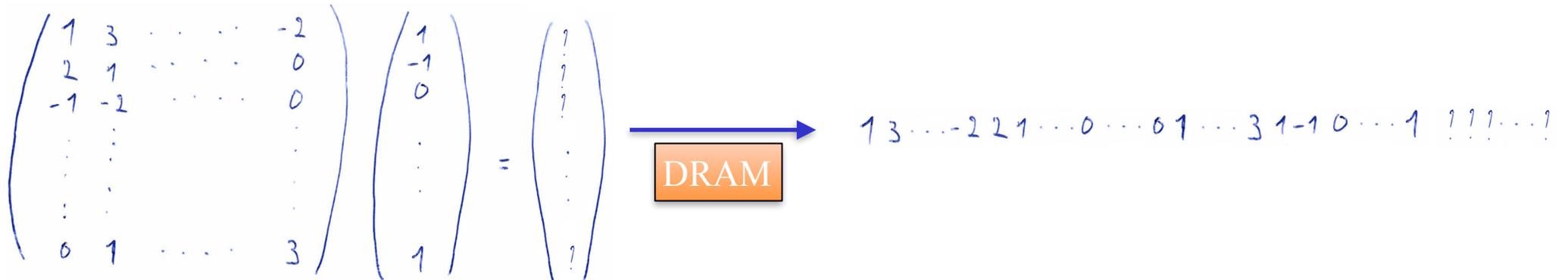
<http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU>

**DRAM** : Dynamic Random Access Memory (mémoire classique)

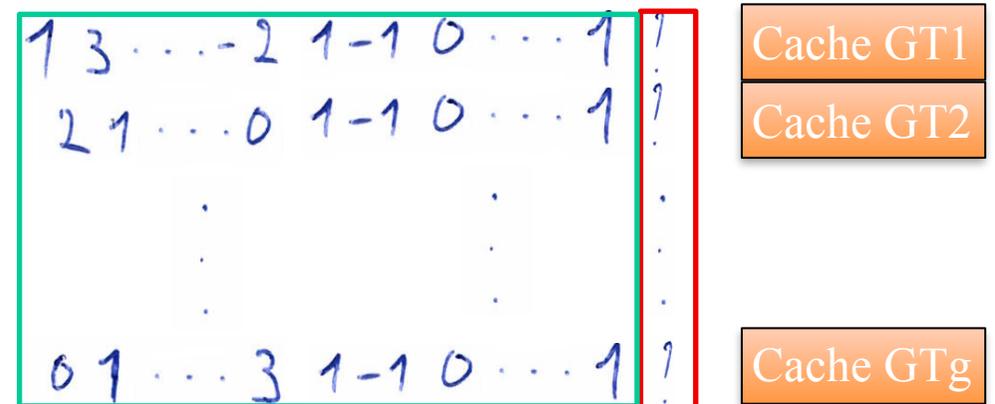
**Cache** : Mémoire (ici) interne au processeur. Rapide mais de taille limité

**ALU** : Arithmetic-Logic Unit. Effectue les calculs.

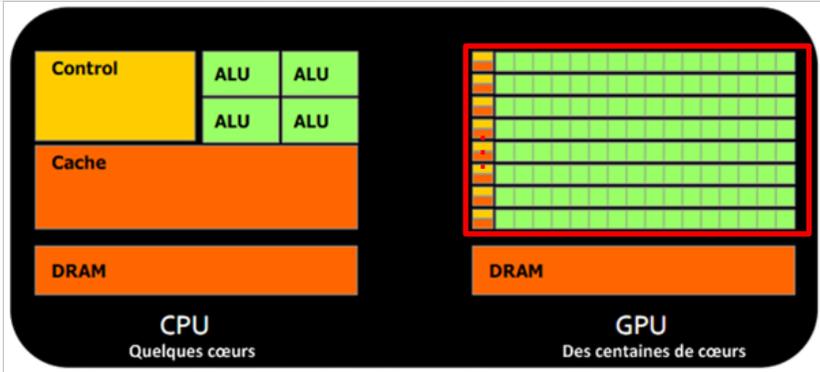
**Control** : Coordonne les ALU et la mémoire



**Etape 2** : Toutes les multiplications et additions sont faites en parallèle



# 6.c) Apprentissage machine et GPU — GPU



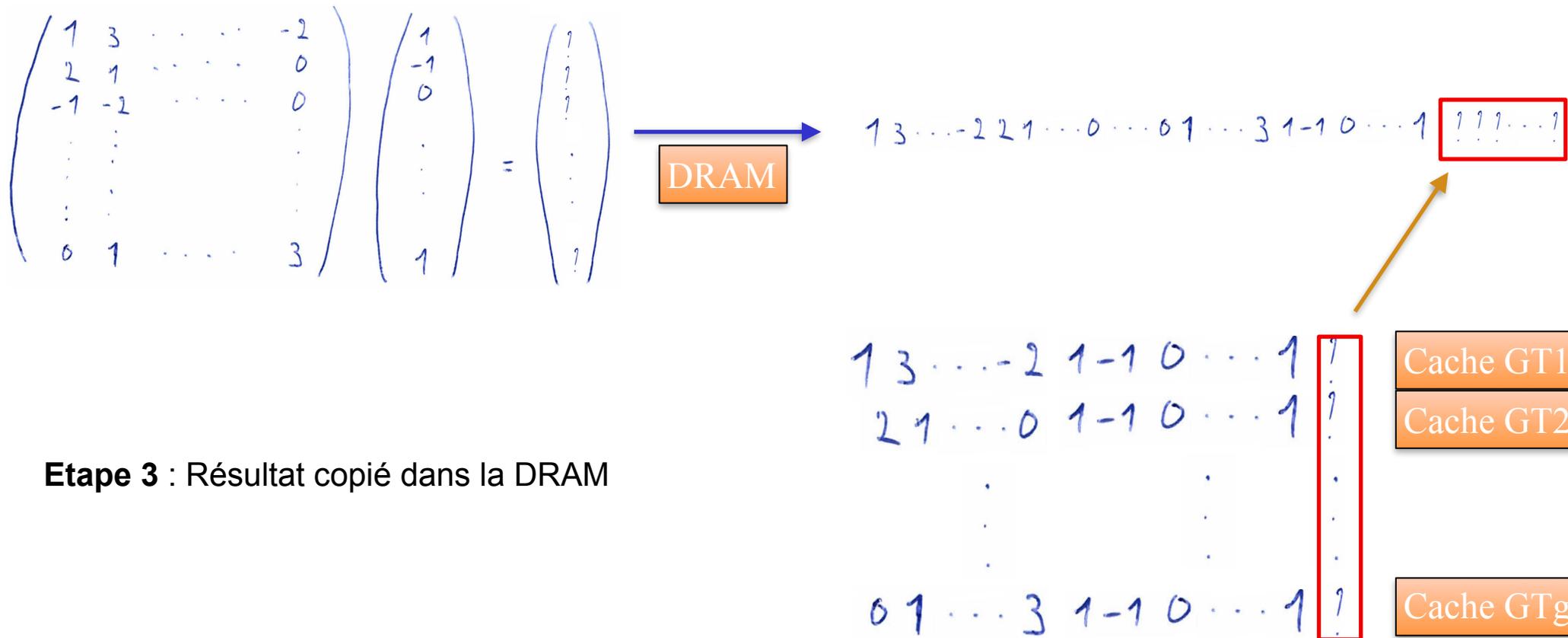
<http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU>

**DRAM** : Dynamic Random Access Memory (mémoire classique)

**Cache** : Mémoire (ici) interne au processeur. Rapide mais de taille limité

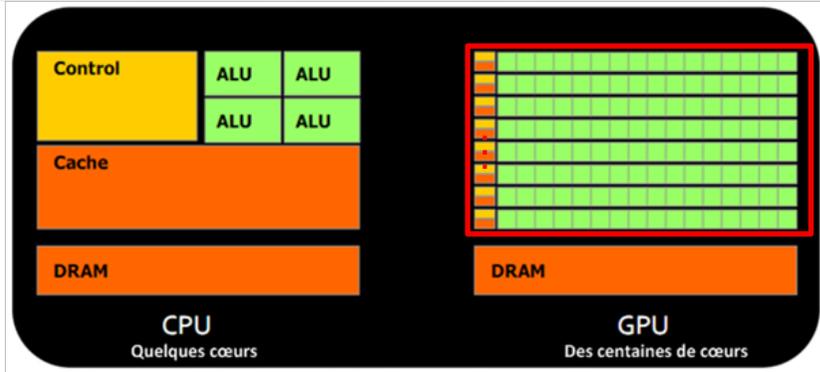
**ALU** : Arithmetic-Logic Unit. Effectue les calculs.

**Control** : Coordonne les ALU et la mémoire



**Etape 3** : Résultat copié dans la DRAM

## 6.c) Apprentissage machine et GPU — GPU



<http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU>

**DRAM** : Dynamic Random Access Memory (mémoire classique)

**Cache** : Mémoire (ici) interne au processeur. Rapide mais de taille limité

**ALU** : Arithmetic-Logic Unit. Effectue les calculs.

**Control** : Coordonne les ALU et la mémoire

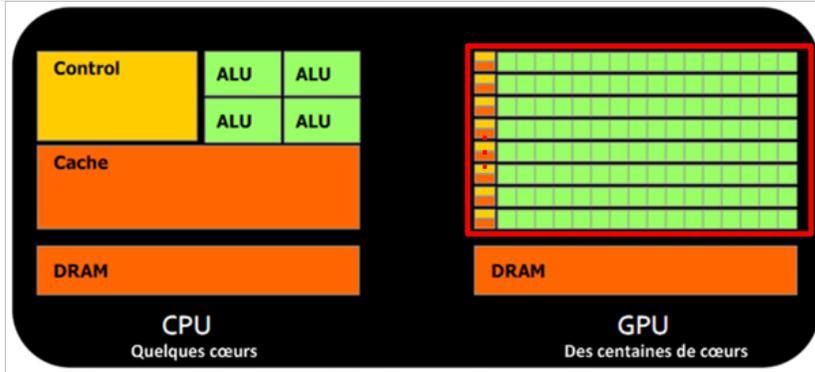
### Réflexions pour la programmation bas-niveau :

- $[\text{Gain de temps en parallélisant les calculs}] - [\text{perte de temps dû aux transferts de données}] > 0$  ?
- Peut-on rendre négligeable les temps de transferts (latence) ?

### Réflexions pour la programmation haut-niveau :

- Est-ce que les mêmes instructions (additions, multiplications, ...) vont s'effectuer sur de gros blocs de données ?
- Est-ce que ces blocs de données sont continus en mémoire ?

## 6.c) Apprentissage machine et GPU — GPU



<http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU>

**DRAM** : Dynamic Random Access Memory (mémoire classique)

**Cache** : Mémoire (ici) interne au processeur. Rapide mais de taille limité

**ALU** : Arithmetic-Logic Unit. Effectue les calculs.

**Control** : Coordonne les ALU et la mémoire

### Réflexions pour la programmation bas-niveau :

- [Gain de temps en parallélisant les calculs] - [perte de temps dû aux transferts de données] > 0 ?
- Peut-on rendre négligeable les temps de transferts (latence) ?

### Réflexions pour la programmation haut-niveau :

- Est-ce que les mêmes instructions (additions, multiplications, ...) vont s'effectuer sur de gros blocs de données ?
- Est-ce que ces blocs de données sont continus en mémoire ?

### Intérêt sur d'autres algorithmes de machine learning :

- Succès de XGboost, librairie Nvidia Rapids, ...
- Intérêt sur des graphes ou d'autres données peu régulières en espace ???

## 7) Conclusion

- Champ de recherche très actif et applications nombreuses
- Formalisme de description des données bien défini
- Lien avec le web des données ?

MERCI !!!