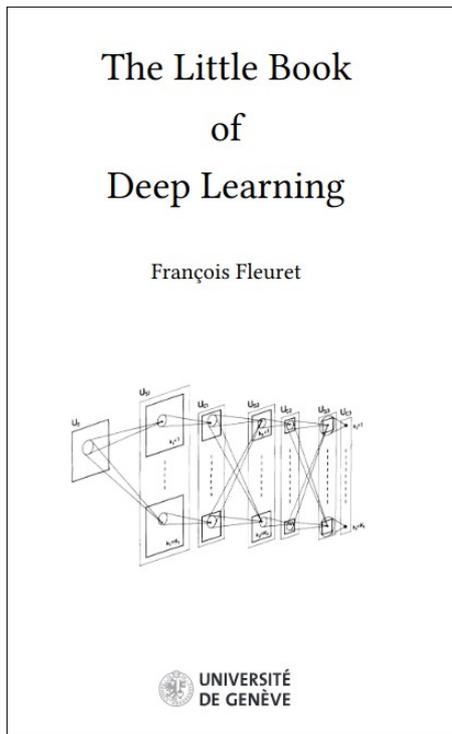


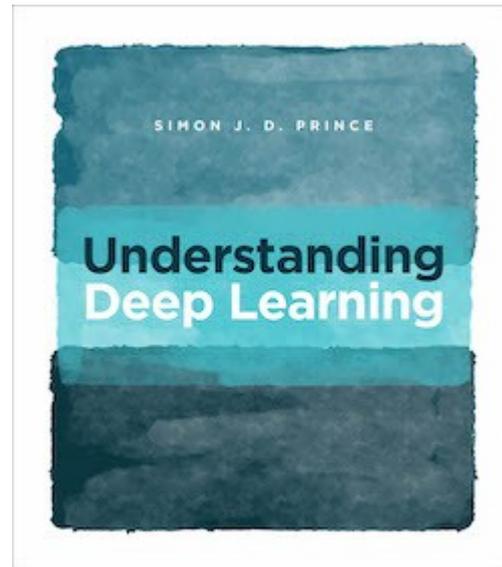
Introduction aux réseaux de neurones pour l'apprentissage supervisé

Guillaume Bourmaud

Livres conseillés



<https://fleuret.org/francois/lbdl.html>



<https://udlbook.github.io/udlbook/>

PLAN

I. Introduction

II. Apprentissage supervisé

III. Approches paramétriques

IV. Réseaux de neurones

V. Risques

I) Introduction

1)

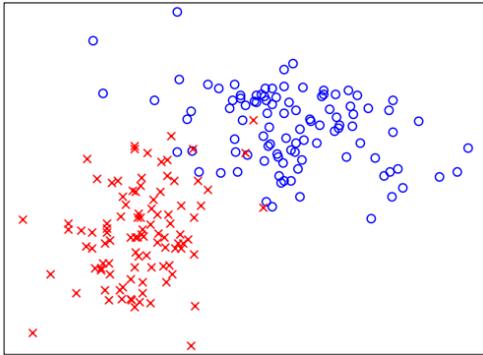
Apprentissage automatique (« Machine Learning »)

Laisser un ordinateur déduire des « **règles** »
d'un ensemble de **données numériques**.

Ex. d'application : recommandation de produits

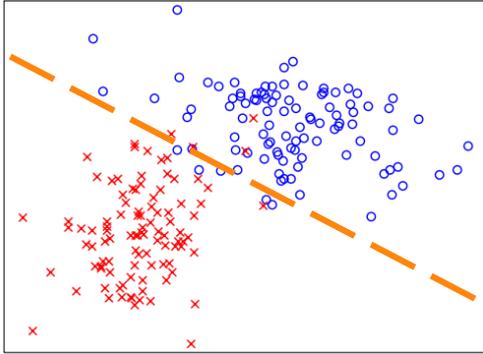
Apprentissage automatique (« Machine Learning »)

Supervisé



Apprentissage automatique (« Machine Learning »)

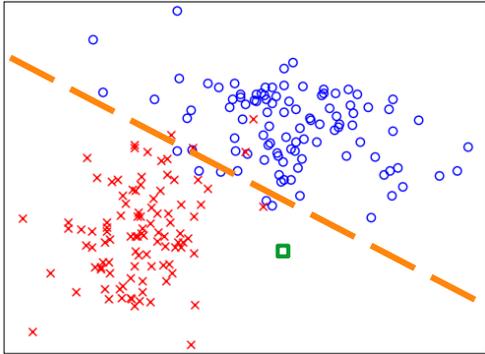
Supervisé



Déduire des « règles »
associant une étiquette à une
donnée.

Apprentissage automatique (« Machine Learning »)

Supervisé

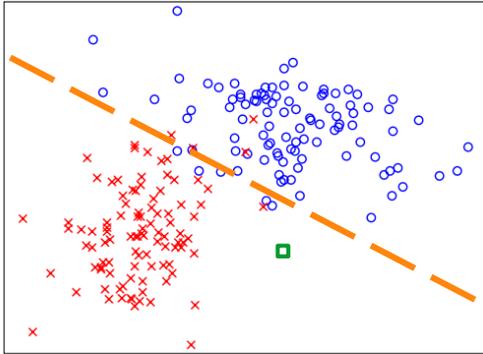


Déduire des « règles »
associant une étiquette à une
donnée.

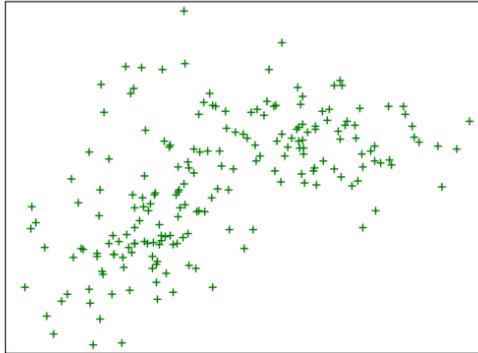
Utilisation la plus fréquente :
Classifier une nouvelle donnée

Apprentissage automatique (« Machine Learning »)

Supervisé



Non supervisé

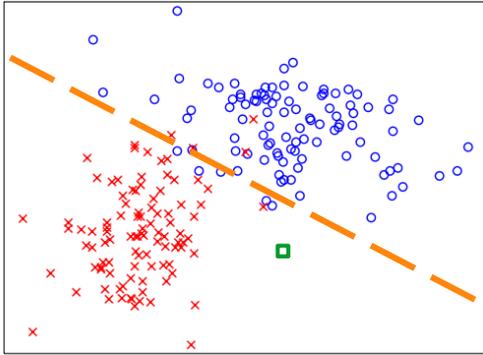


Déduire des « règles »
associant une étiquette à une
donnée.

Utilisation la plus fréquente :
Classifier une nouvelle donnée

Apprentissage automatique (« Machine Learning »)

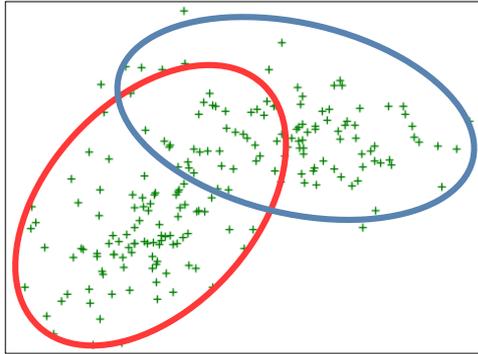
Supervisé



Déduire des « règles »
associant une étiquette à une
donnée.

Utilisation la plus fréquente :
Classifier une nouvelle donnée

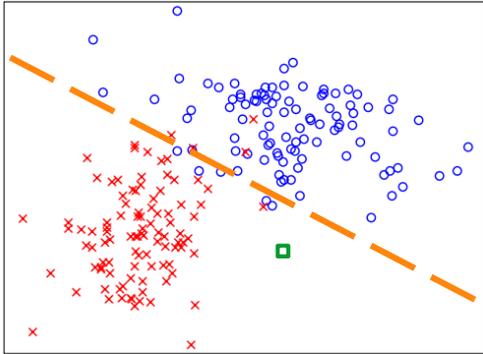
Non supervisé



Déduire des « règles »
structurant les données.

Apprentissage automatique (« Machine Learning »)

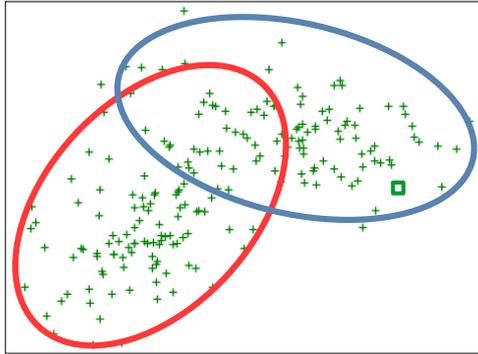
Supervisé



Déduire des « règles »
associant une étiquette à une
donnée.

Utilisation la plus fréquente :
Classifier une nouvelle donnée

Non supervisé

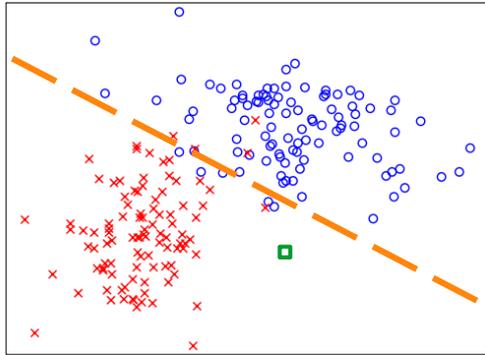


Déduire des « règles »
structurant les données.

Ex. d'utilisation :
- Réduction de dimension
- Générer une nouvelle donnée

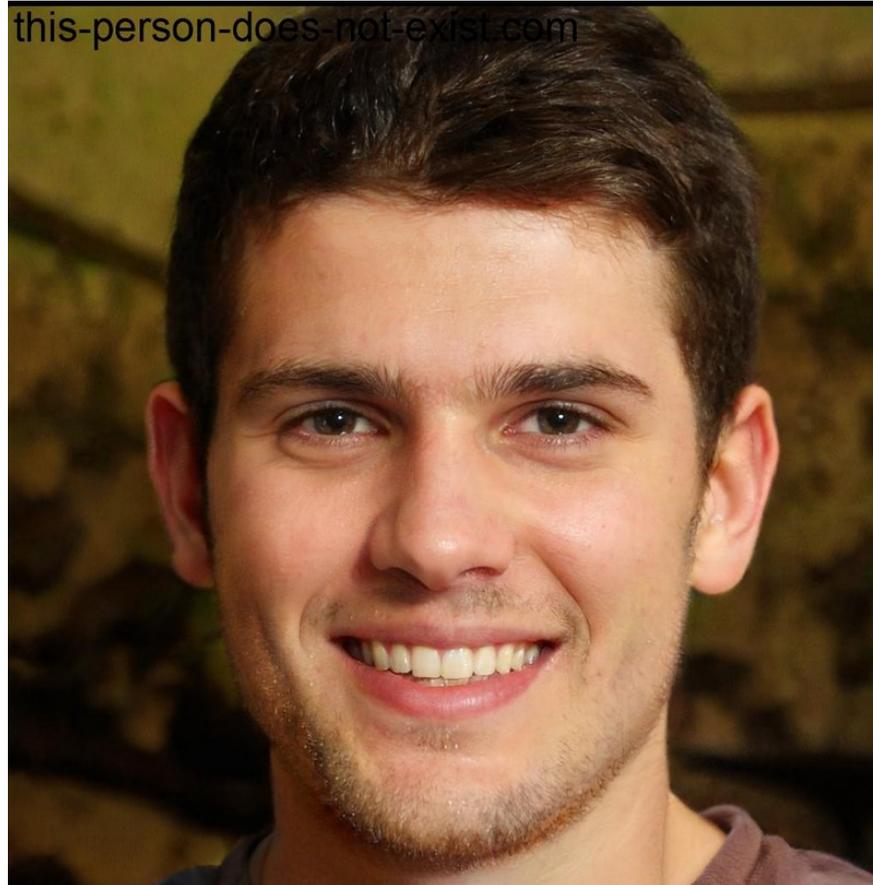
Apprentissage automatique (« Machine Learning »)

Supervisé



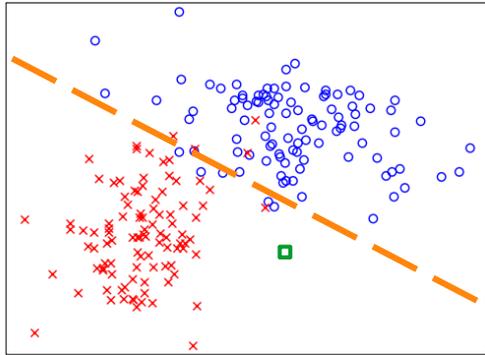
Déduire des « règles »
associant une étiquette à une
donnée.

Utilisation la plus fréquente :
Classifier une nouvelle donnée



Apprentissage automatique (« Machine Learning »)

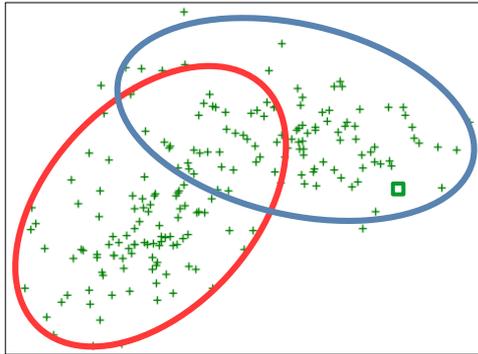
Supervisé



Déduire des « règles »
associant une étiquette à une
donnée.

Utilisation la plus fréquente :
Classifier une nouvelle donnée

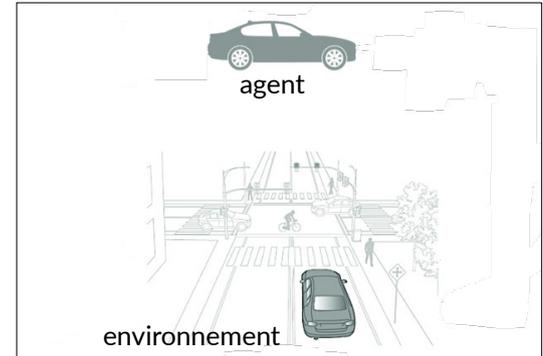
Non supervisé



Déduire des « règles »
structurant les données.

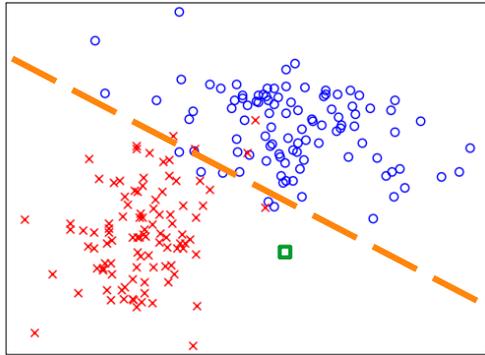
Ex. d'utilisation :
- Réduction de dimension
- Générer une nouvelle donnée

Par renforcement



Apprentissage automatique (« Machine Learning »)

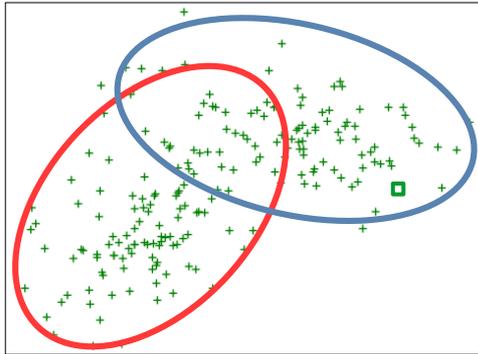
Supervisé



Déduire des « règles » associant une étiquette à une donnée.

Utilisation la plus fréquente :
Classifier une nouvelle donnée

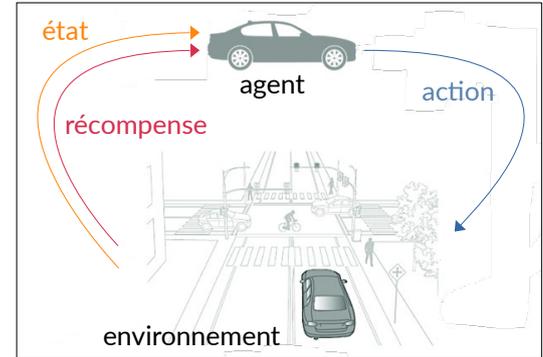
Non supervisé



Déduire des « règles » structurant les données.

Ex. d'utilisation :
- Réduction de dimension
- Générer une nouvelle donnée

Par renforcement

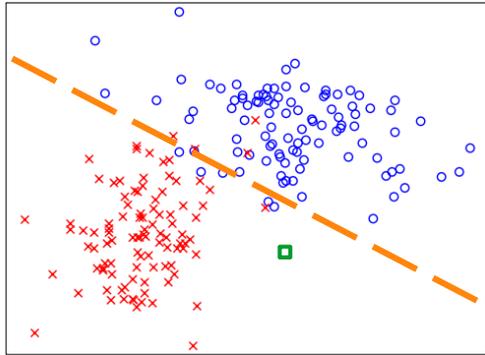


L'agent interagit avec l'environnement et déduit des « règles » (actions) permettant de maximiser une récompense.

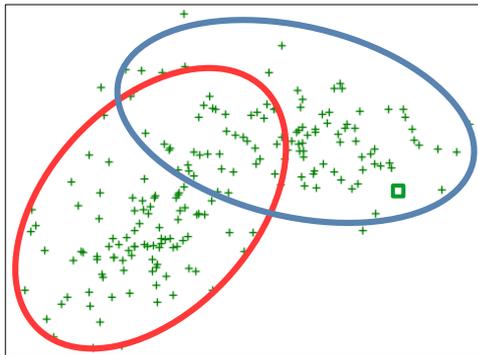
Ex. d'application :
Obtenir un véhicule autonome

Apprentissage automatique (« Machine Learning »)

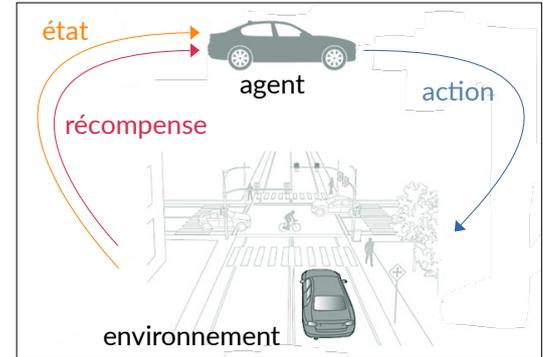
Supervisé



Non supervisé



Par renforcement



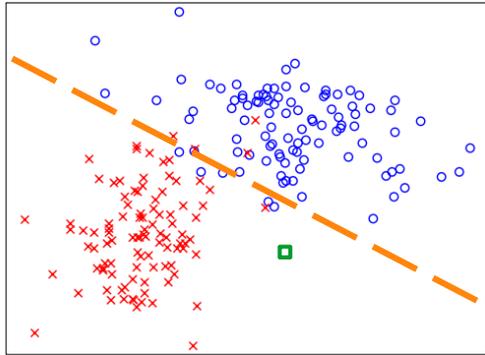
« Deep learning »

=

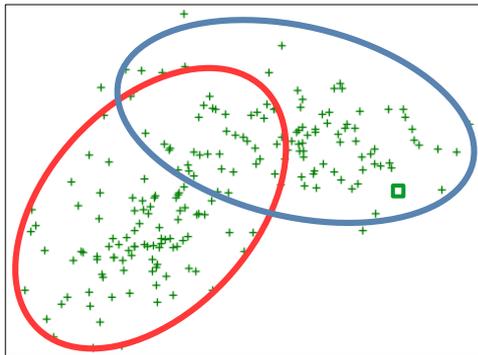
utilisation d'un réseau de neurones en apprentissage automatique

Apprentissage automatique (« Machine Learning »)

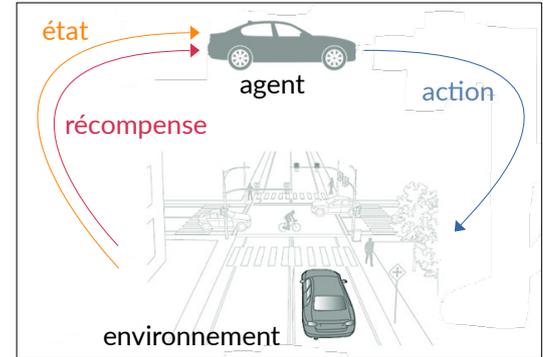
Supervisé



Non supervisé



Par renforcement



Aujourd'hui : le terme IA \approx « Deep learning »

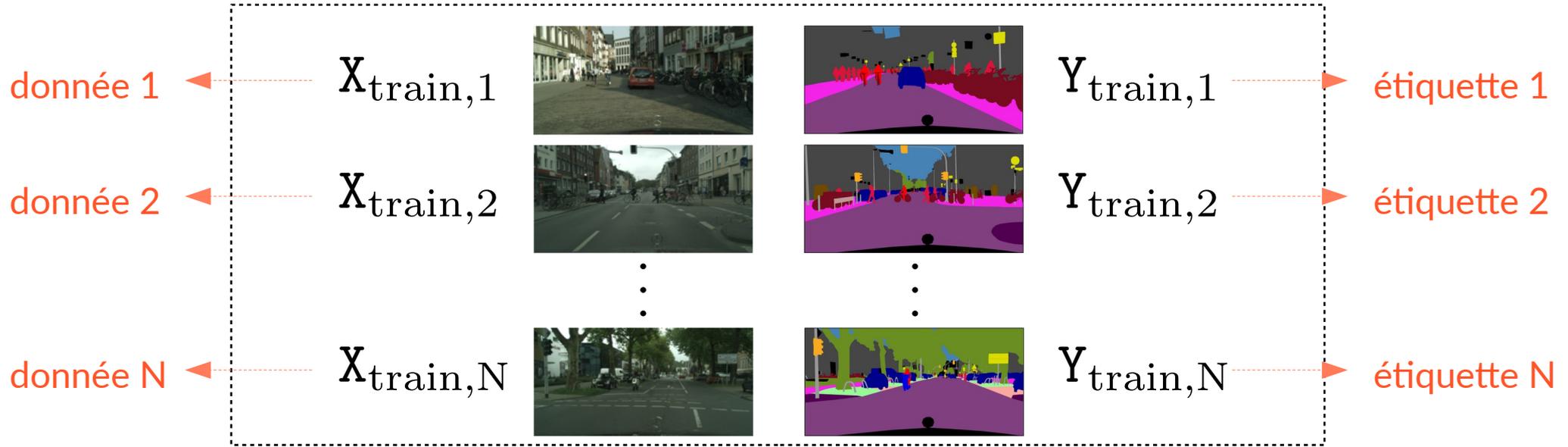
=

utilisation d'un réseau de neurones en apprentissage automatique

II) Apprentissage supervisé

II)

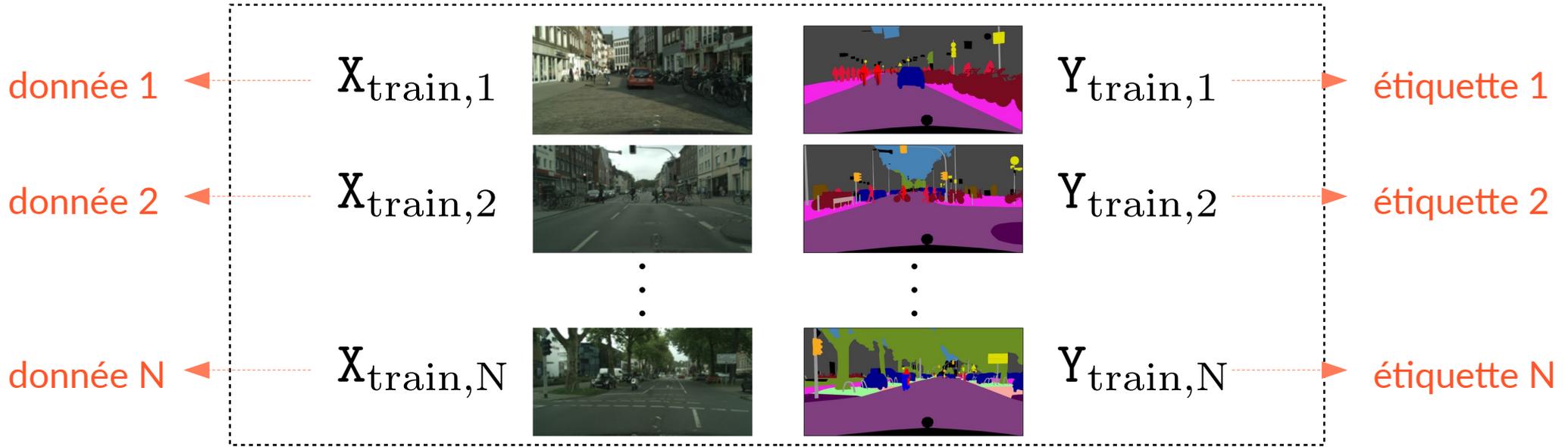
Apprentissage supervisé



Base de données étiquetées pour la *segmentation sémantique*

II)

Apprentissage supervisé

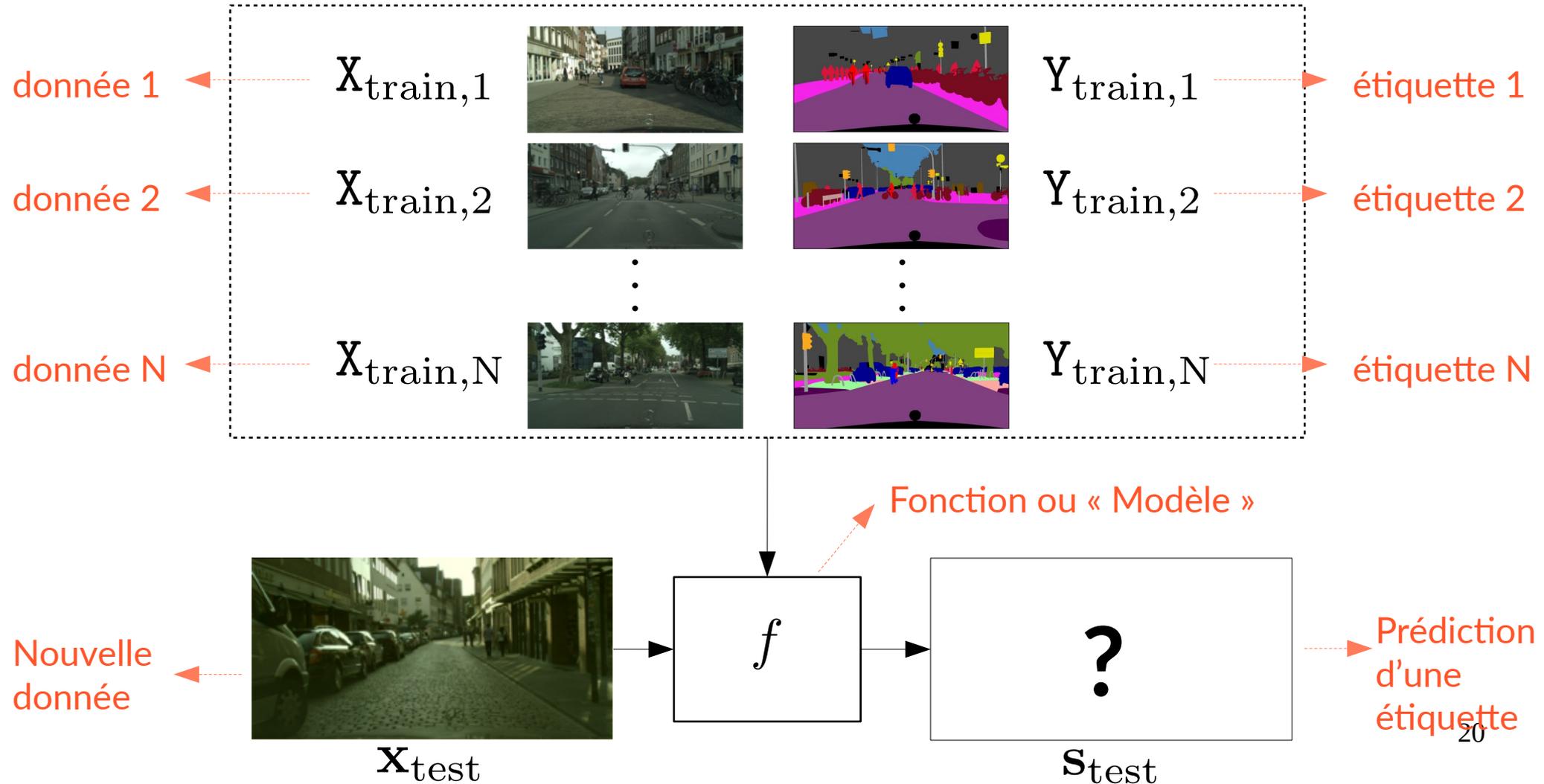


Base de données étiquetées pour la *segmentation sémantique*

- Objectif :
- déduire des « règles » de la base de données étiquetées
 - appliquer ces « règles » à une nouvelle donnée pour prédire une étiquette

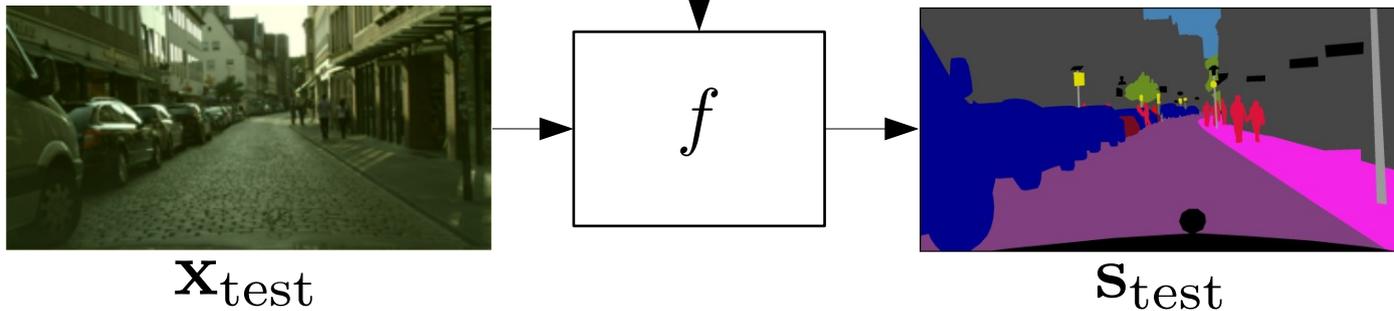
II)

Apprentissage supervisé



II)

Apprentissage supervisé

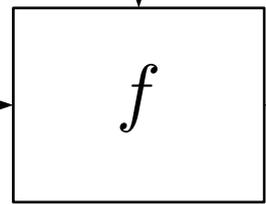
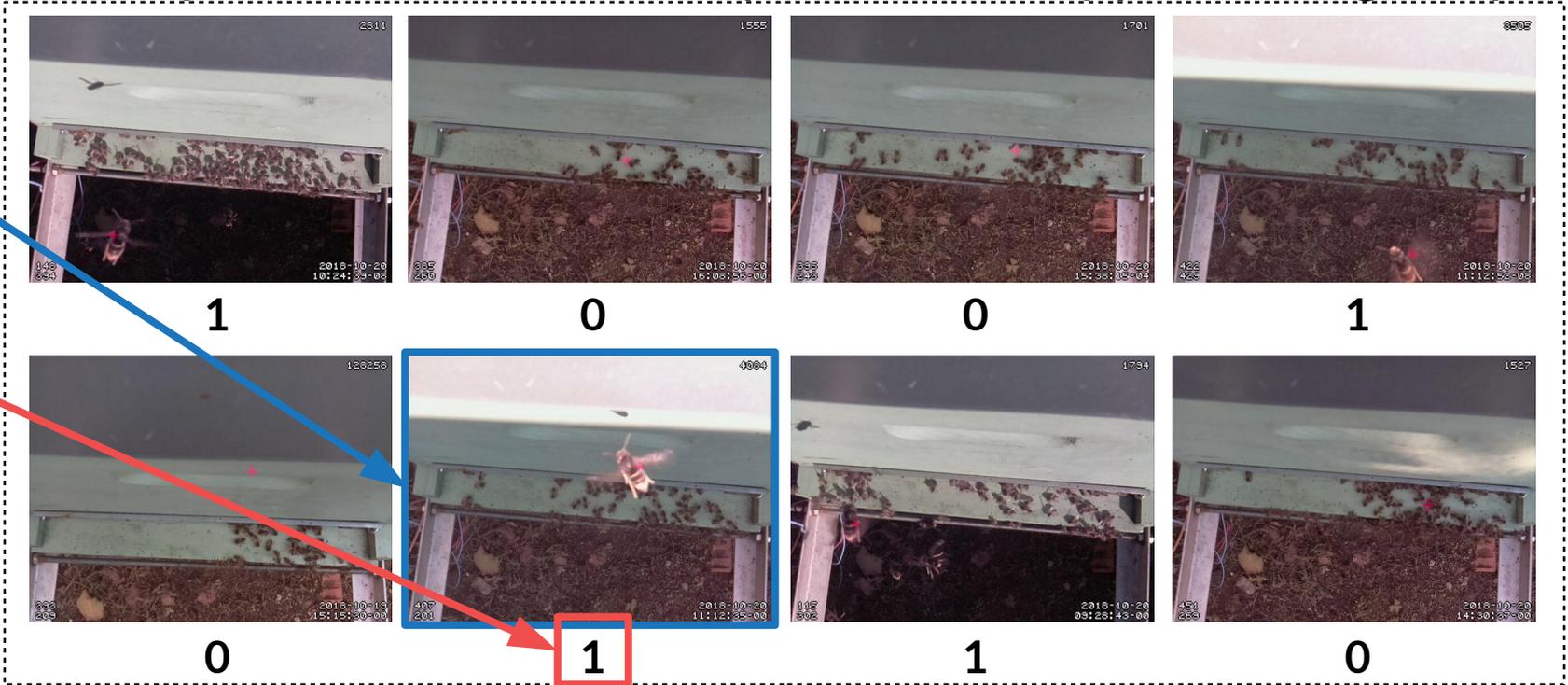


II)

Êtes-vous capable de résoudre ce problème d'apprentissage supervisé ?

Donnée $X_{\text{train},6}$

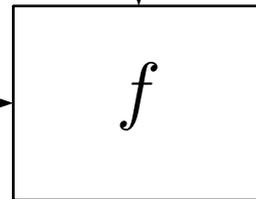
Étiquette $Y_{\text{train},6}$



0 ou 1 ?

II)

Mais de quel problème parle-t-on au juste ?



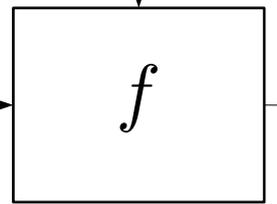
0 ou 1 ?

II)

Nous souhaitons que l'ordinateur apprenne à résoudre le problème suivant

Donnée $X_{\text{train},6}$

Étiquette $Y_{\text{train},6}$



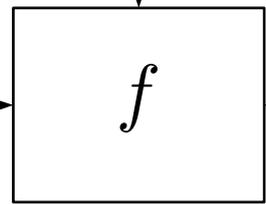
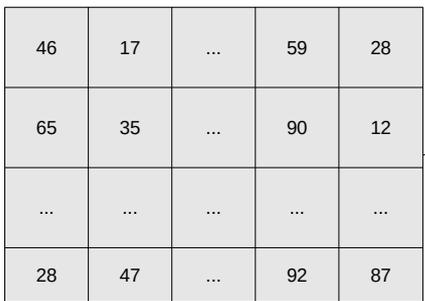
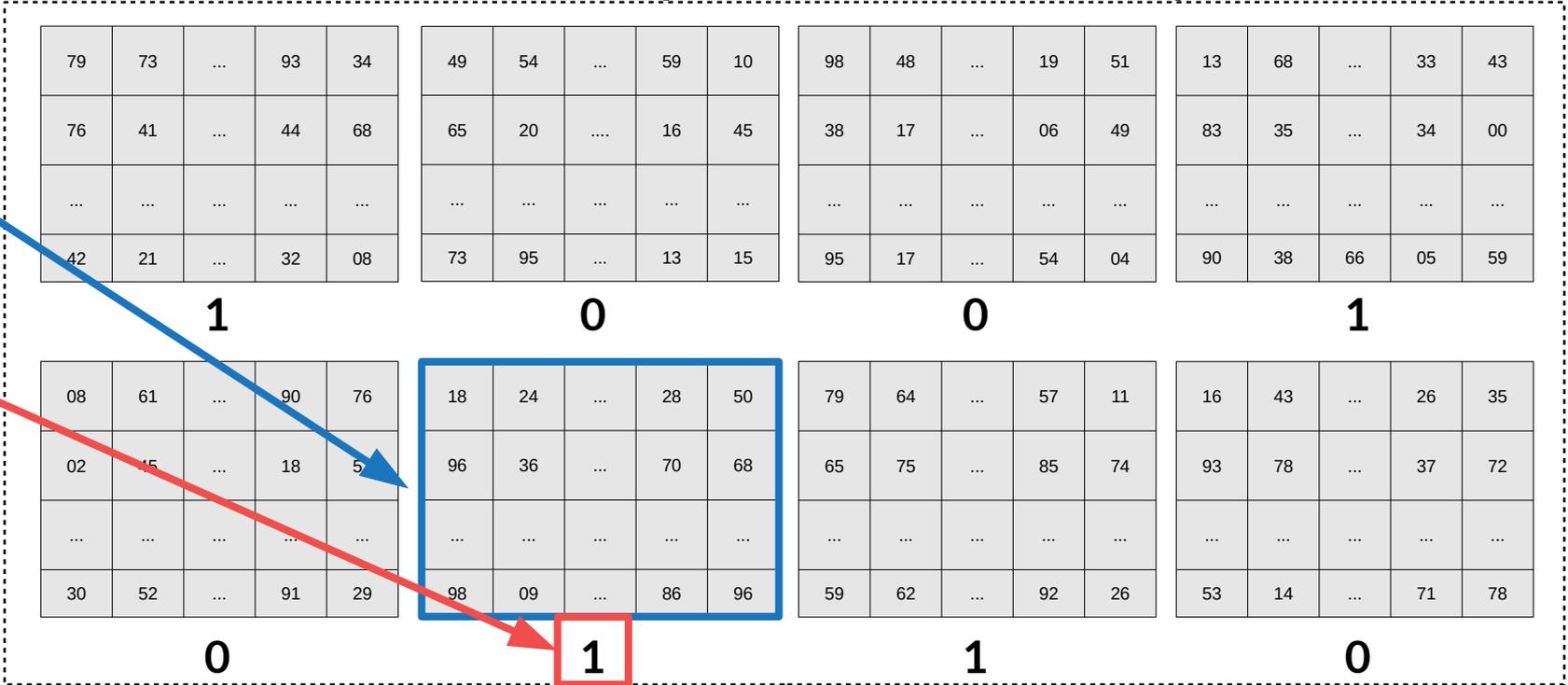
Présence d'au moins un frelon ?
ou
Absence de frelon ?

II)

Mais une fois mis en forme pour l'ordinateur le problème devient

Donnée
 $X_{\text{train},6}$

Étiquette
 $Y_{\text{train},6}$



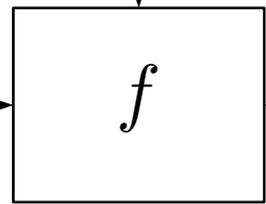
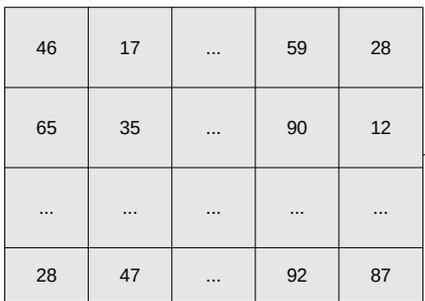
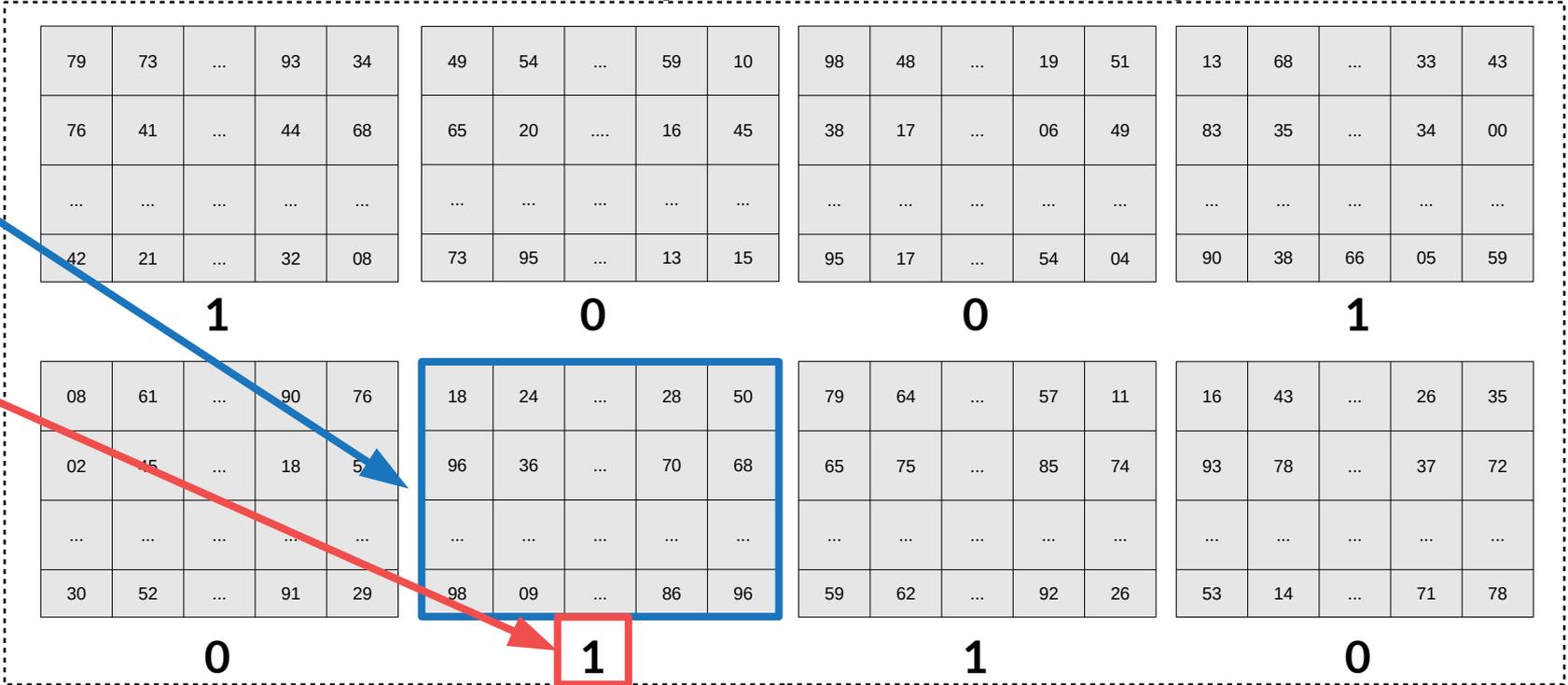
0 ou 1 ?

II)

Mais une fois mis en forme pour l'ordinateur le problème devient

Donnée
 $X_{\text{train},6}$

Étiquette
 $Y_{\text{train},6}$



0 ou 1 ?

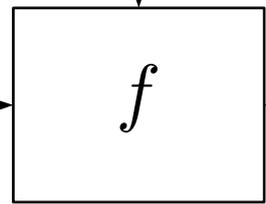
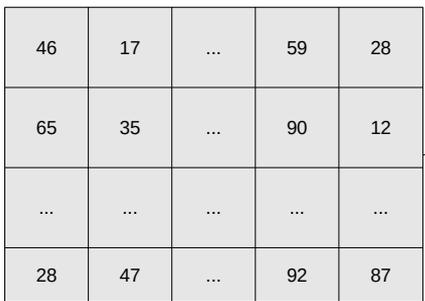
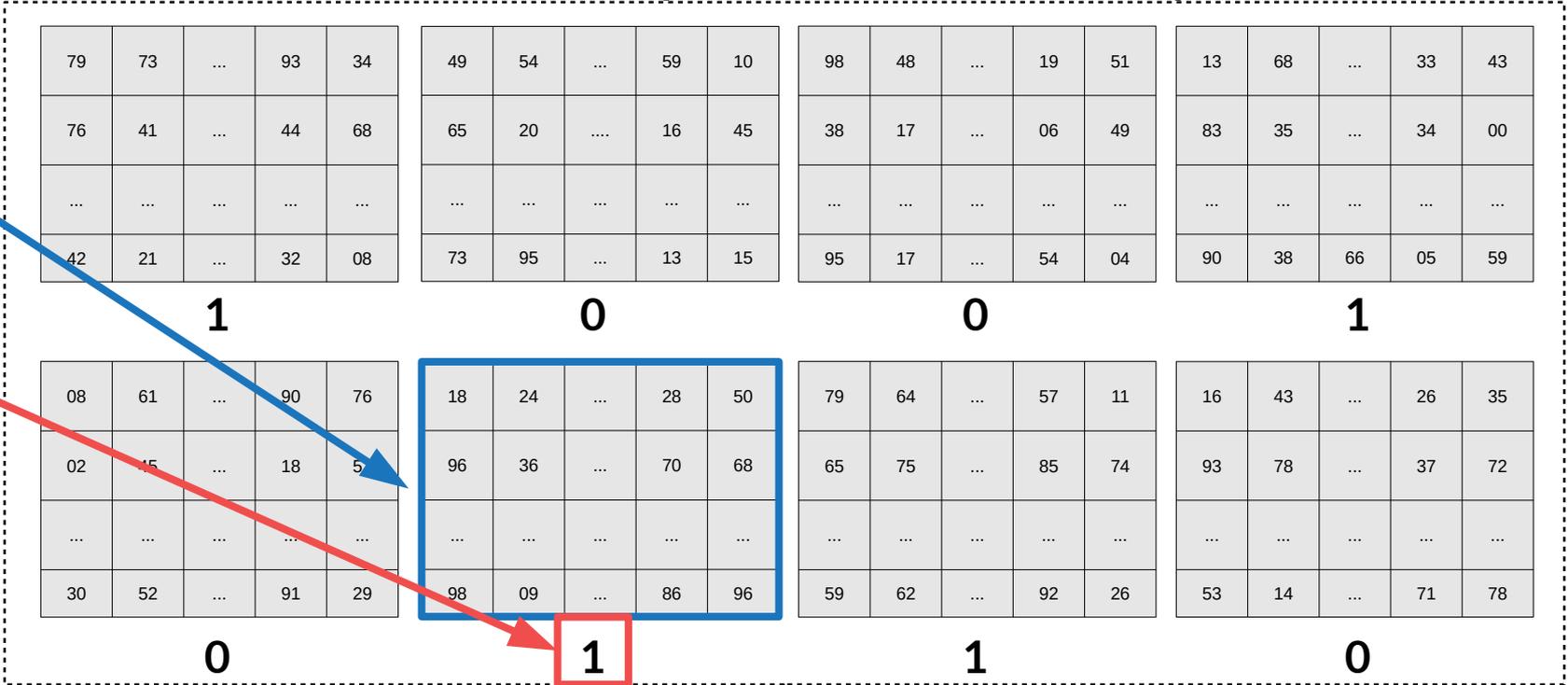
Les « règles » déduites ne seront (probablement) pas interprétables.
→ « Boîte noire »

II)

Mais une fois mis en forme pour l'ordinateur le problème devient

Donnée
 $X_{\text{train},6}$

Étiquette
 $Y_{\text{train},6}$



0 ou 1 ?

C'est nous qui interprétons :
« 0 » = présence frelon
« 1 » = absence frelon

II)

Résumé

- Des « règles » (souvent) non interprétables sont déduites de tableaux de valeurs numériques et d'étiquettes numériques.

II)

Résumé

- Des « règles » (souvent) non interprétables sont déduites de tableaux de valeurs numériques et d'étiquettes numériques.
- Ces « règles » sont appliquées à un nouveau tableau de valeurs numériques pour produire une étiquette numérique.

II)

Résumé

- Des « règles » (souvent) non interprétables sont déduites de tableaux de valeurs numériques et d'étiquettes numériques.
- Ces « règles » sont appliquées à un nouveau tableau de valeurs numériques pour produire une étiquette numérique.
- Nous interprétons cette étiquette numérique comme une étiquette « sémantique ».

II)

Résumé

- Des « règles » (souvent) non interprétables sont déduites de tableaux de valeurs numériques et d'étiquettes numériques.
- Ces « règles » sont appliquées à un nouveau tableau de valeurs numériques pour produire une étiquette numérique.
- Nous interprétons cette étiquette numérique comme une étiquette « sémantique ».
- Il ne s'agit que d'une interprétation...



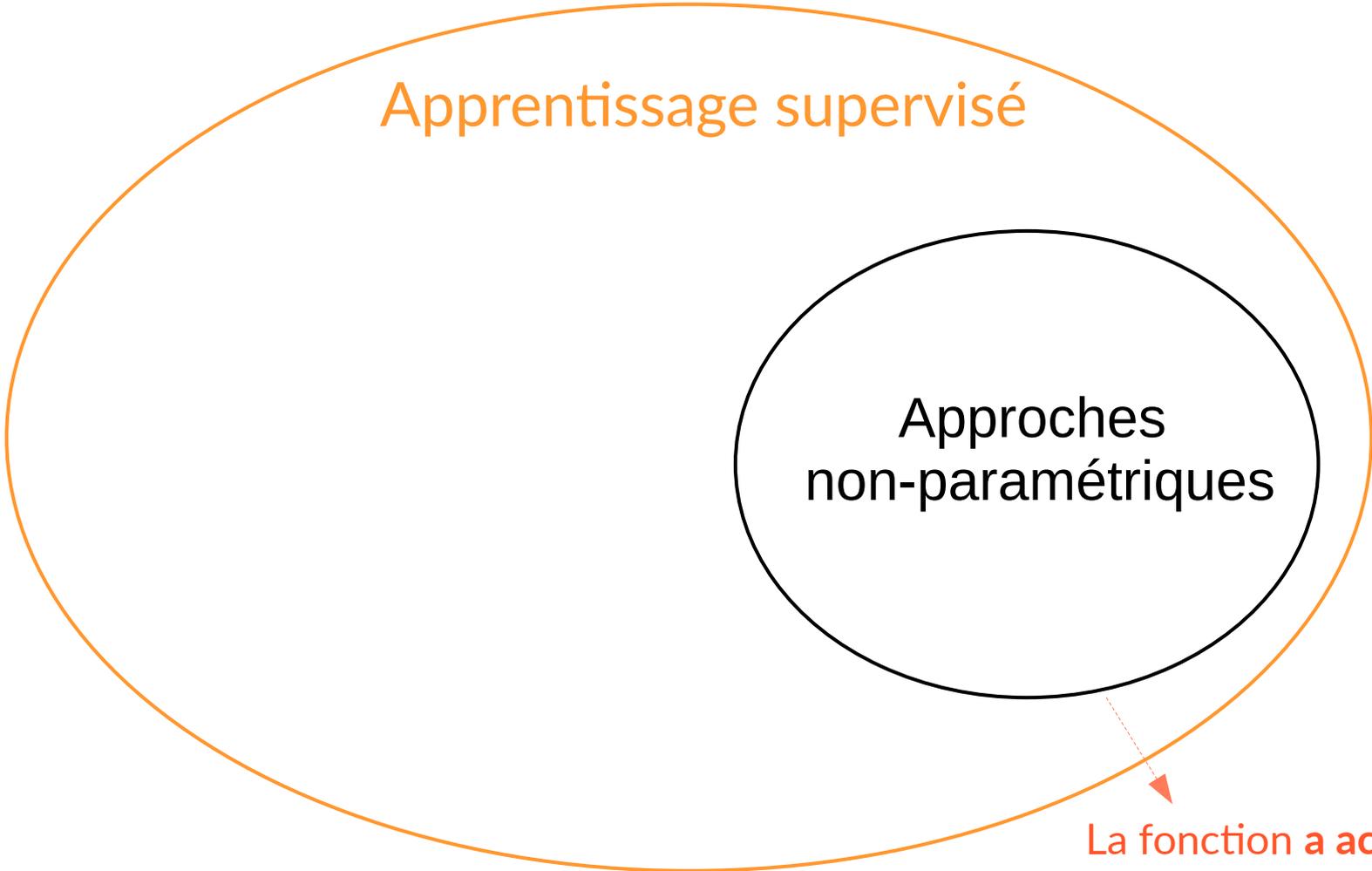
Tesla said autopilot was activated during a fatal Model X crash last week in California.

II)

Apprentissage supervisé

Approches
non-paramétriques

La fonction **a** accède à la base de données étiquetées pour traiter une nouvelle donnée (ex : processus gaussien).



II)

Apprentissage supervisé

Approches
paramétriques

Approches
non-paramétriques

La fonction n'a pas
directement accès à la base de
données étiquetées pour
traiter une nouvelle donnée.

La fonction a accès à la base
de données étiquetées pour
traiter une nouvelle donnée
(ex : processus gaussien).

III) Approches paramétriques

III)

Représentation d'une fonction paramétrique

Mathématique

$$s = f(\mathbf{x}; \boldsymbol{\theta})$$

Paramètres

III)

Représentation d'une fonction paramétrique

Mathématique

$$s = f(\mathbf{x}; \boldsymbol{\theta})$$

Paramètres



Informatique
(Python)

```
def f(x, theta):  
    ...  
    ...  
    s = ...  
    return s
```

III)

Représentation d'une fonction paramétrique

Mathématique

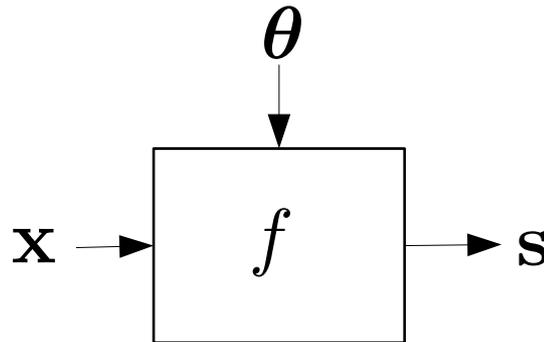
$$s = f(\mathbf{x}; \boldsymbol{\theta})$$

Paramètres

Informatique
(Python)

```
def f(x, theta):  
    ...  
    ...  
    s = ...  
    return s
```

Graphique
(Graphe de calcul)



III) Exemple de fonction paramétrique : transformation affine

Mathématique

$$\mathbf{s} = f(\mathbf{x}; \boldsymbol{\theta} = \{\mathbf{W}, \mathbf{b}\}) = \mathbf{W}\mathbf{x} + \mathbf{b}$$

III) Exemple de fonction paramétrique : transformation affine

Mathématique

$$s = f(\mathbf{x}; \theta = \{W, \mathbf{b}\}) = W\mathbf{x} + \mathbf{b}$$

Informatique
(Python)

```
def f(x, W, b):  
    s = x @ W + b  
    return s
```

III) Exemple de fonction paramétrique : transformation affine

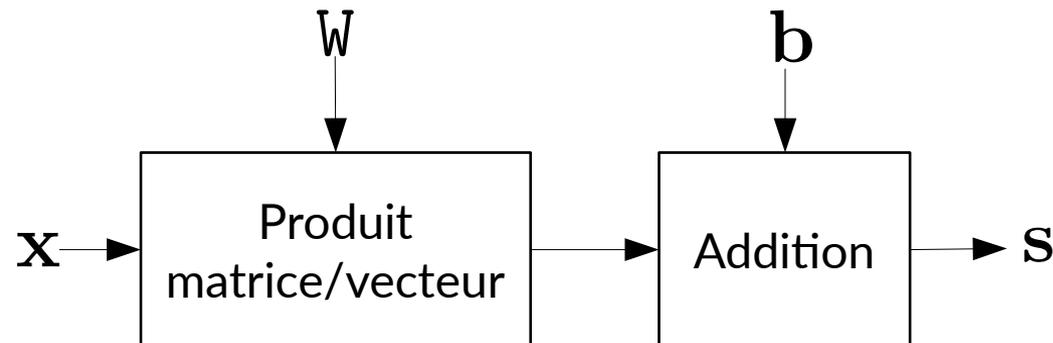
Mathématique

$$s = f(x; \theta = \{W, b\}) = Wx + b$$

Informatique
(Python)

```
def f(x, W, b):  
    s = x @ W + b  
    return s
```

Graphique
(Graphe de calcul)



III) Exemple de fonction paramétrique : modèle polynomial

Mathématique

$$s = f(x; \boldsymbol{\theta}) = \sum_{i=0}^d \boldsymbol{\theta}_i x^i = \boldsymbol{\theta}^\top \boldsymbol{\phi}(x) \quad \text{où} \quad \boldsymbol{\phi}(x) = [1 \quad x \quad \dots \quad x^d]^\top$$

III)

Exemple de fonction paramétrique : modèle polynomial

Mathématique

$$s = f(x; \theta) = \sum_{i=0}^d \theta_i x^i = \theta^\top \phi(x) \quad \text{où} \quad \phi(x) = [1 \quad x \quad \dots \quad x^d]^\top$$

Informatique
(Python)

```
def poly(x, theta):  
    phi_x = phi(x)  
    s = phi_x @ theta  
    return s
```

III)

Exemple de fonction paramétrique : modèle polynomial

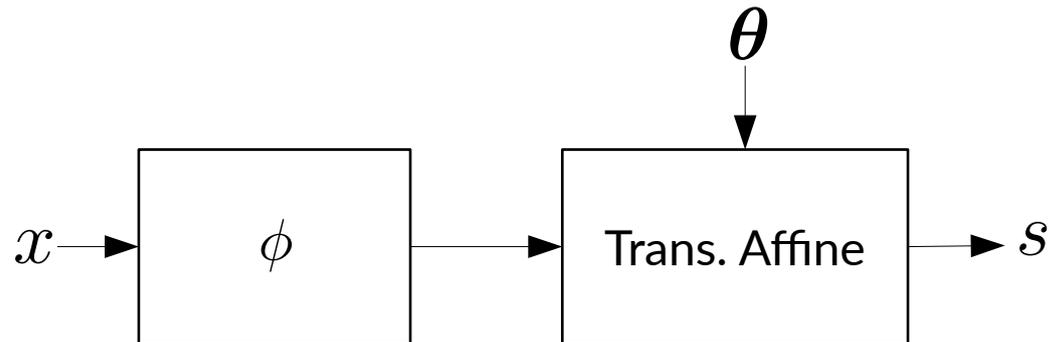
Mathématique

$$s = f(x; \theta) = \sum_{i=0}^d \theta_i x^i = \theta^\top \phi(x) \quad \text{où} \quad \phi(x) = [1 \quad x \quad \dots \quad x^d]^\top$$

Informatique
(Python)

```
def poly(x, theta):  
    phi_x = phi(x)  
    s = phi_x @ theta  
    return s
```

Graphique
(Graphe de calcul)

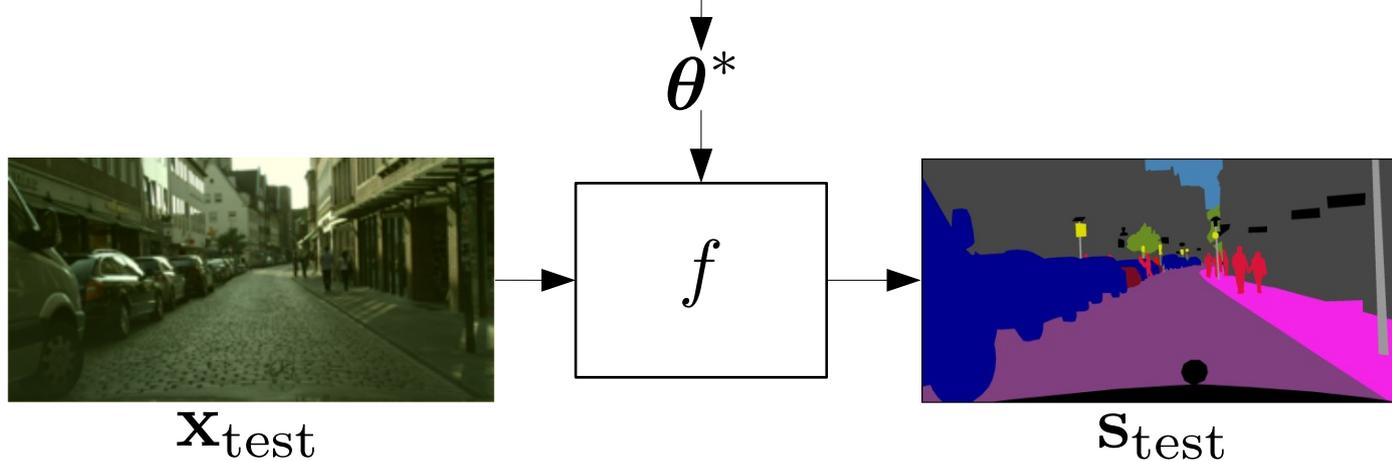


III)

Approches paramétriques



Les « règles » déduites de la base de données étiquetées sont (en partie) encodées dans la valeur des paramètres θ^* .

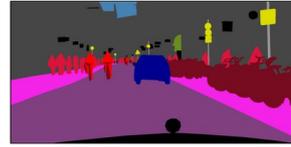


III)

Approches paramétriques

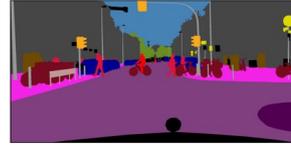
Apprentissage

$X_{\text{train},1}$



$Y_{\text{train},1}$

$X_{\text{train},2}$



$Y_{\text{train},2}$

⋮

⋮

$X_{\text{train},N}$

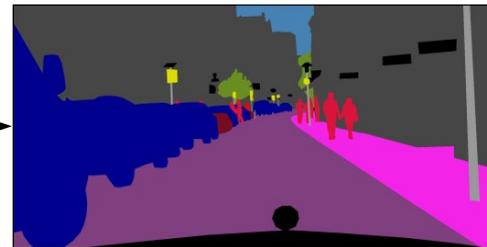
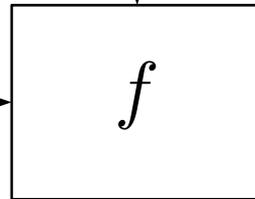


$Y_{\text{train},N}$

θ^*



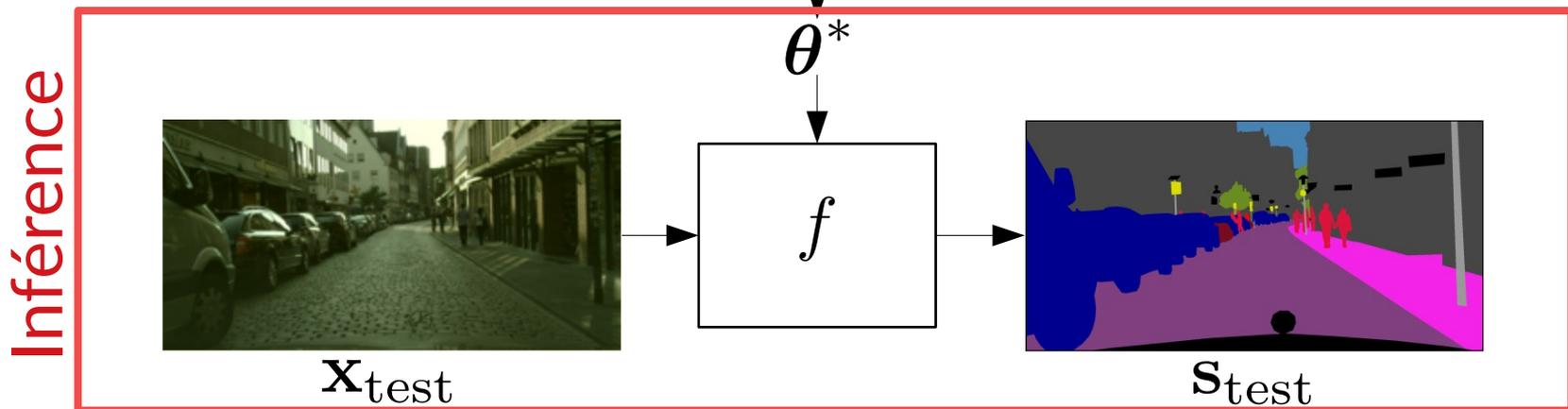
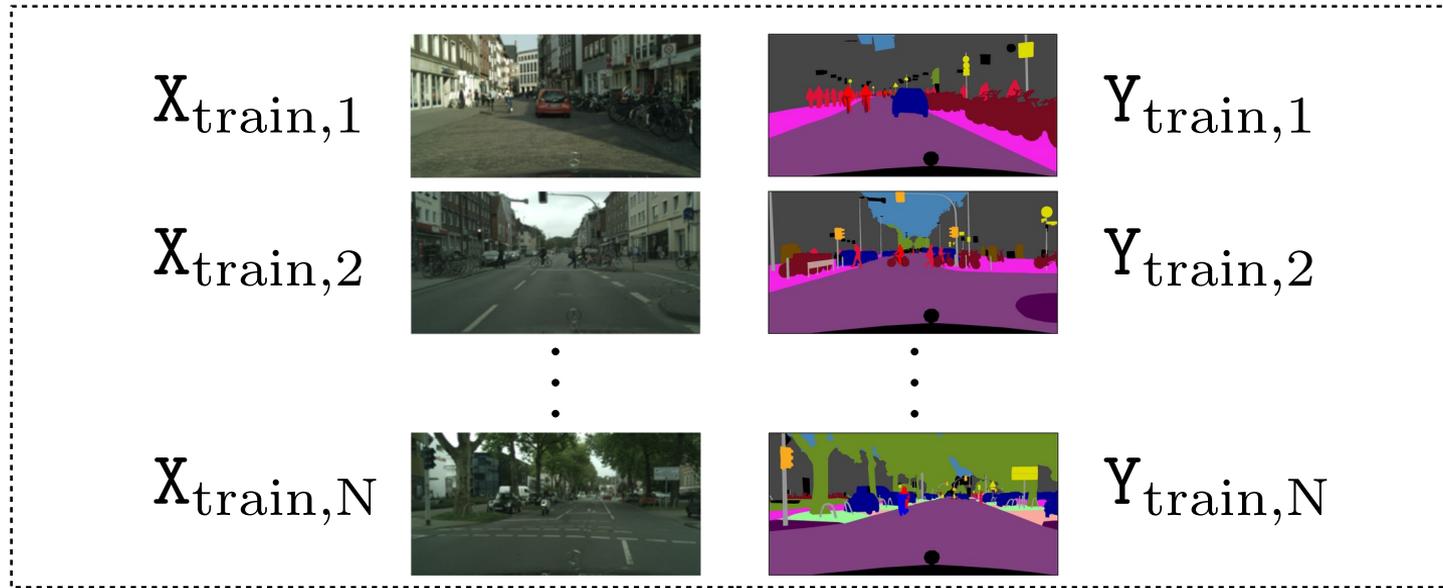
X_{test}



S_{test}

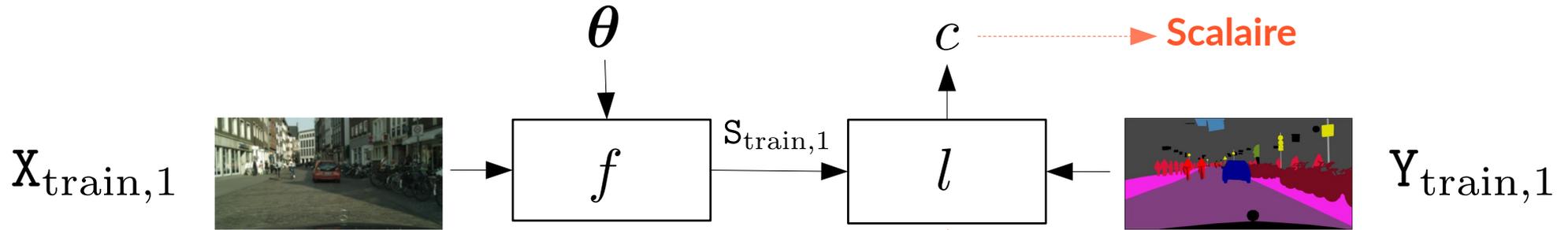
III)

Approches paramétriques



III)

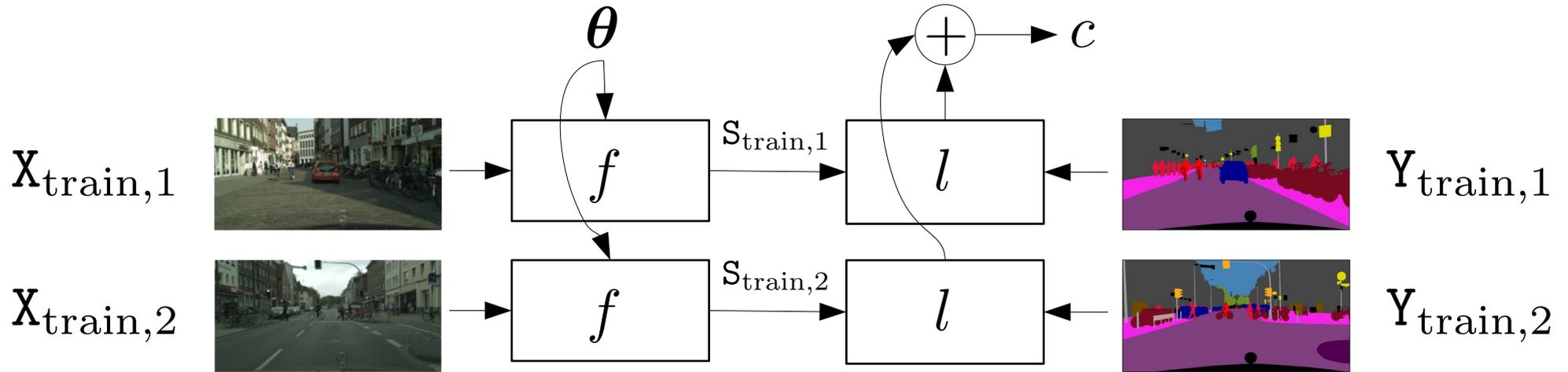
Étape d'apprentissage ("Training time")



Fonction de coût (« loss function ») : permet de comparer la prédiction du réseau à l'étiquette.

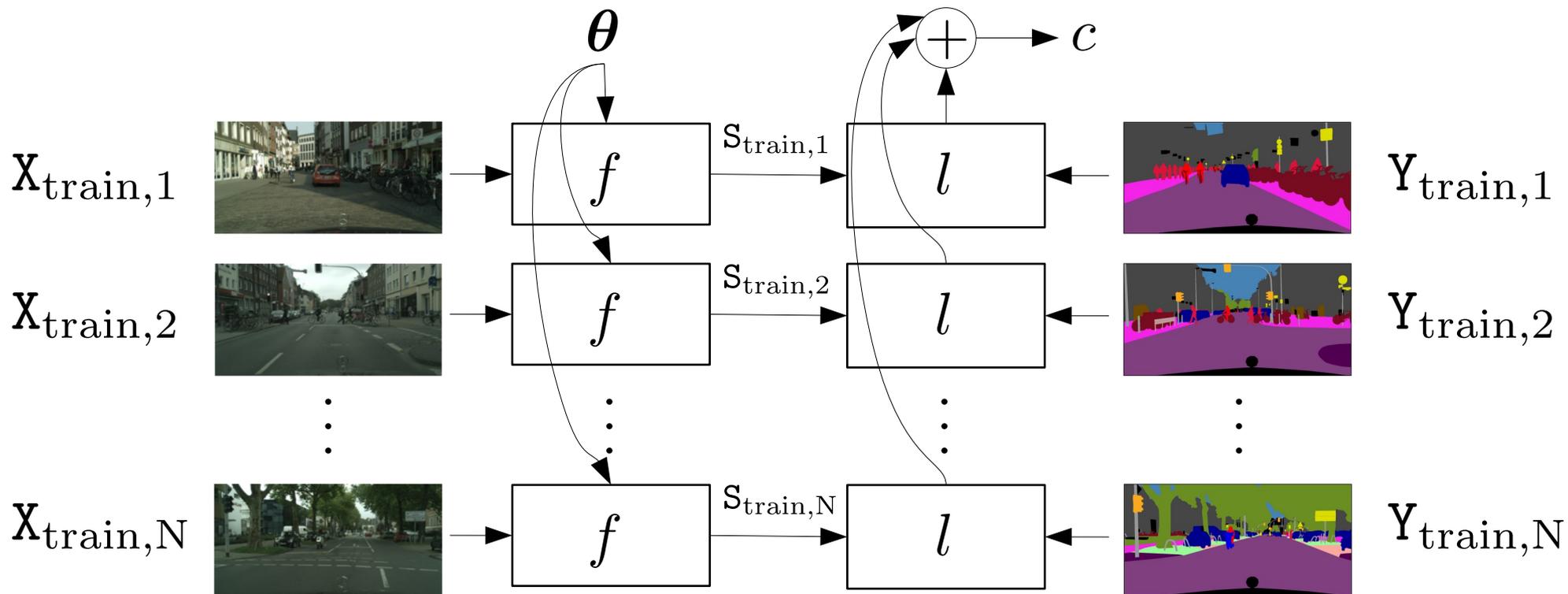
III)

Étape d'apprentissage ("Training time")



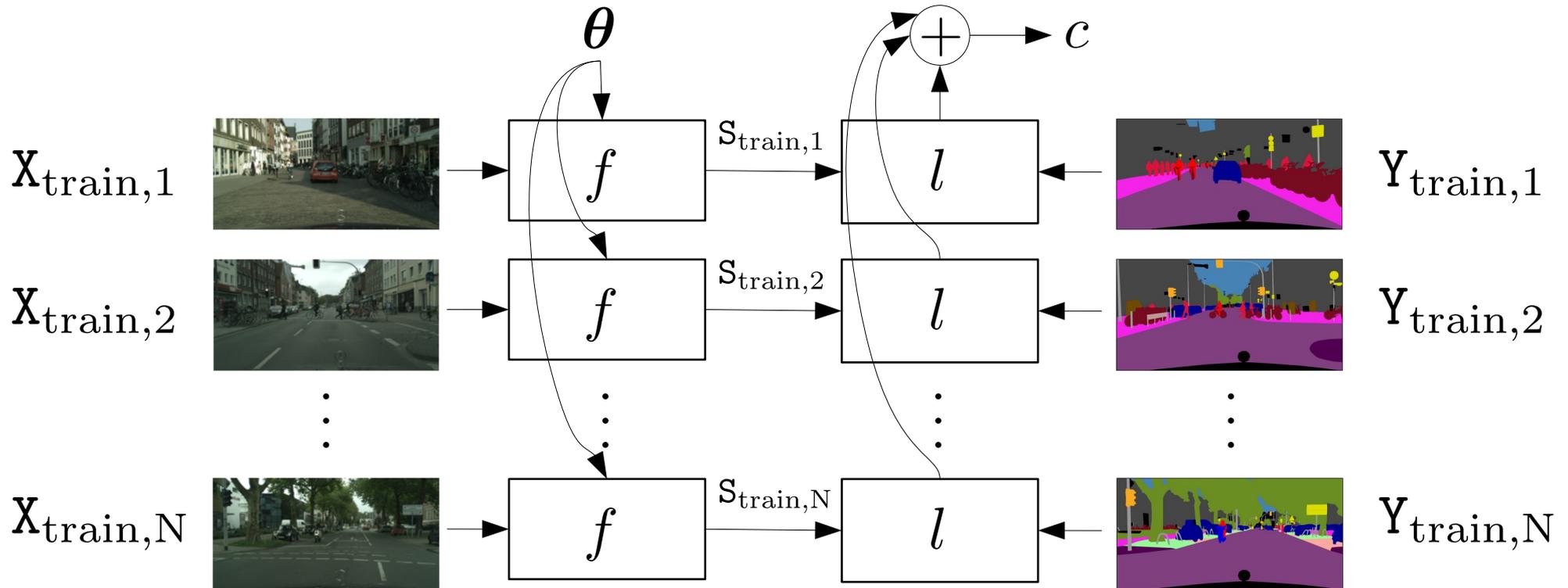
III)

Étape d'apprentissage ("Training time")



III)

Étape d'apprentissage ("Training time")

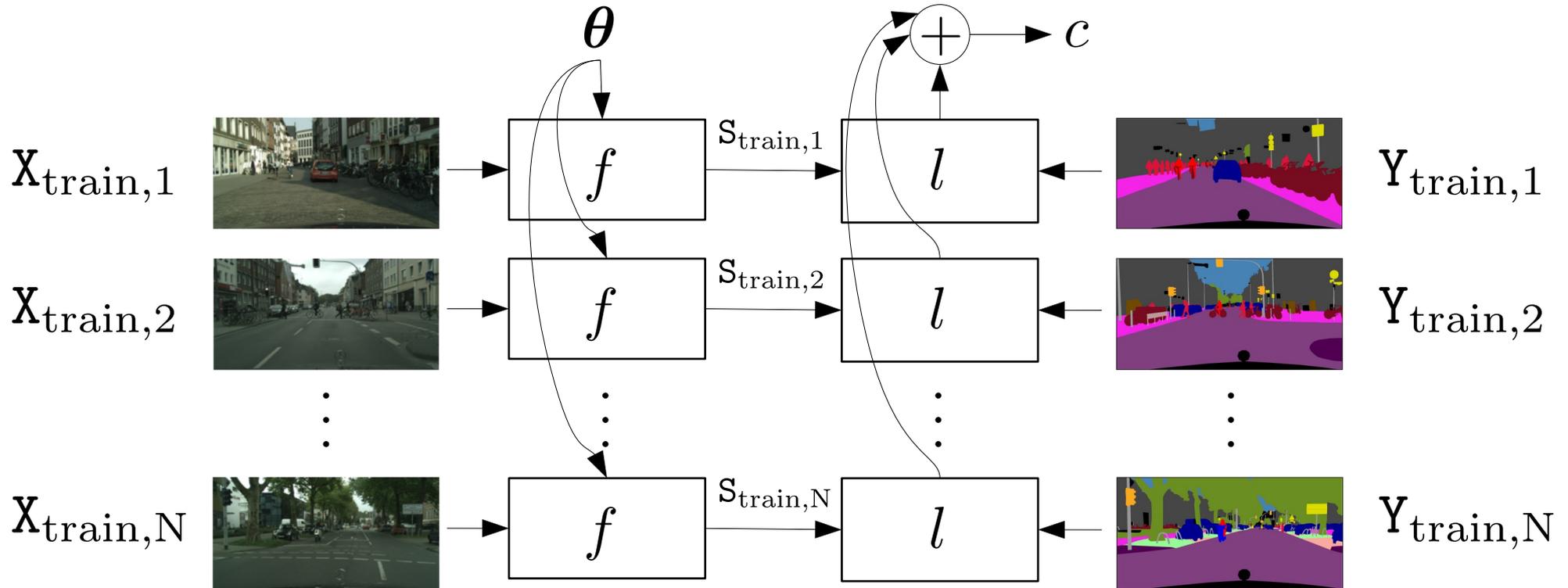


Coût d'entraînement
 (« Training loss »)

$$\sum_{i=1}^N l(Y_{\text{train},i}, f(X_{\text{train},i}; \theta))$$

III)

Étape d'apprentissage ("Training time")

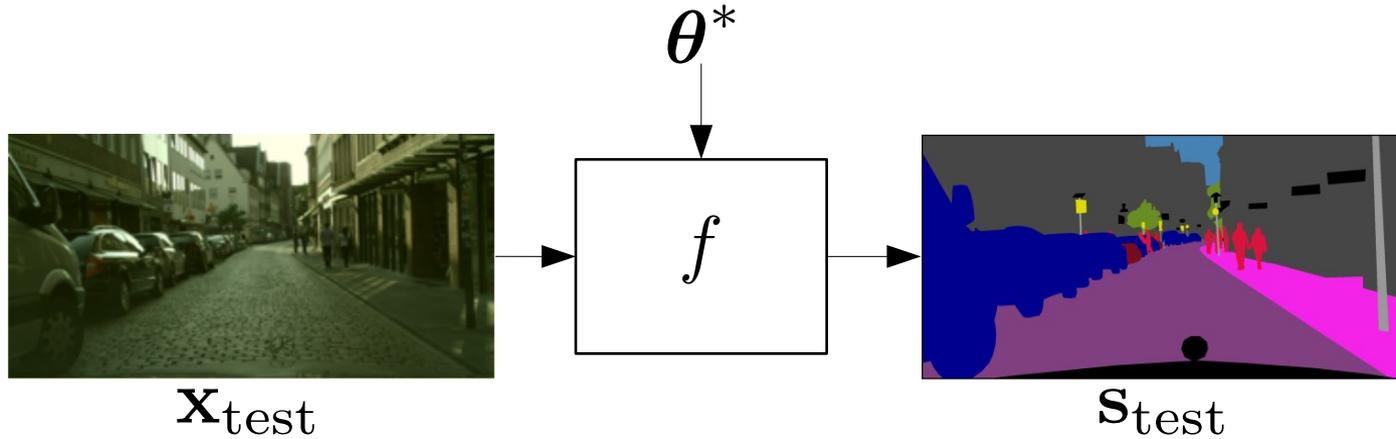


**Problème
d'optimisation**

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^N l(Y_{\text{train},i}, f(X_{\text{train},i}; \theta))$$

III)

Étape d'inférence ("Test time")

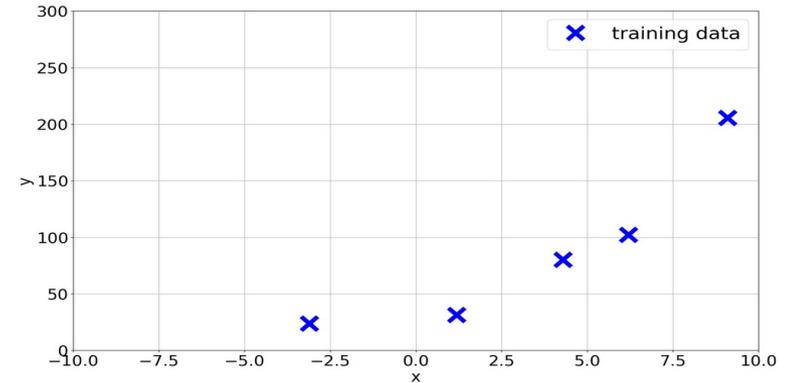


$$\mathbf{s}_{\text{test}} = f(\mathbf{x}_{\text{test}}; \theta^*)$$

III)

Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

$$\begin{array}{ccccccccc} X_{\text{train},1} = -3.1 & X_{\text{train},2} = 1.2 & X_{\text{train},3} = 4.3 & X_{\text{train},4} = 6.2 & X_{\text{train},5} = 9.1 \\ Y_{\text{train},1} = 23.7 & Y_{\text{train},2} = 31.3 & Y_{\text{train},3} = 79.9 & Y_{\text{train},4} = 101.9 & Y_{\text{train},5} = 205.5 \end{array}$$



III)

Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

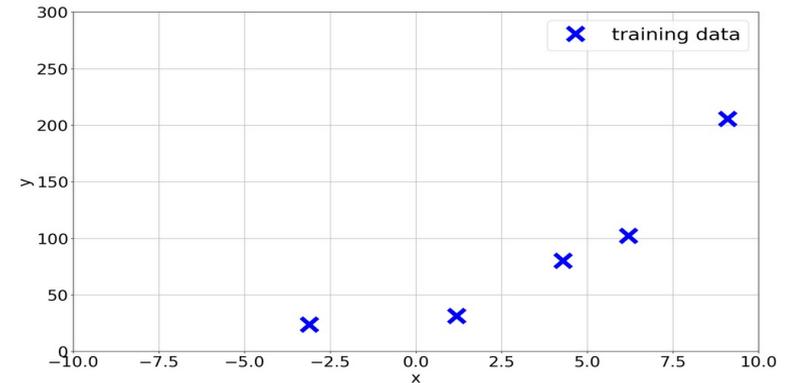
$$X_{\text{train},1} = -3.1 \quad X_{\text{train},2} = 1.2 \quad X_{\text{train},3} = 4.3 \quad X_{\text{train},4} = 6.2 \quad X_{\text{train},5} = 9.1$$

$$Y_{\text{train},1} = 23.7 \quad Y_{\text{train},2} = 31.3 \quad Y_{\text{train},3} = 79.9 \quad Y_{\text{train},4} = 101.9 \quad Y_{\text{train},5} = 205.5$$

Choix de la fonction $f(x; \theta) = \sum_{i=0}^d \theta_i x^i = \theta^\top \phi(x)$

où $\phi(x) = [1 \quad x \quad \dots \quad x^d]^\top$

hyper-paramètre : d \longrightarrow degré du polynôme



III)

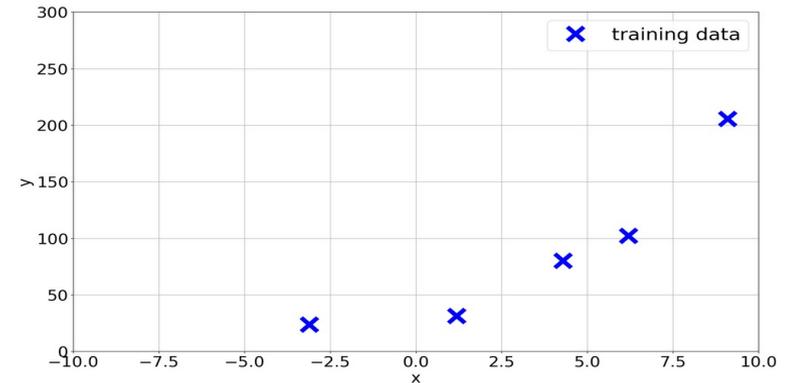
Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

$$\begin{array}{ccccc} X_{\text{train},1} = -3.1 & X_{\text{train},2} = 1.2 & X_{\text{train},3} = 4.3 & X_{\text{train},4} = 6.2 & X_{\text{train},5} = 9.1 \\ Y_{\text{train},1} = 23.7 & Y_{\text{train},2} = 31.3 & Y_{\text{train},3} = 79.9 & Y_{\text{train},4} = 101.9 & Y_{\text{train},5} = 205.5 \end{array}$$

Choix de la fonction $f(x; \theta) = \sum_{i=0}^d \theta_i x^i = \theta^\top \phi(x)$

où $\phi(x) = [1 \quad x \quad \dots \quad x^d]^\top$

hyper-paramètre : d \longrightarrow degré du polynôme



Quel est le nombre de paramètres du « modèle » ?

III)

Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

$$\begin{array}{ccccc} X_{\text{train},1} = -3.1 & X_{\text{train},2} = 1.2 & X_{\text{train},3} = 4.3 & X_{\text{train},4} = 6.2 & X_{\text{train},5} = 9.1 \\ Y_{\text{train},1} = 23.7 & Y_{\text{train},2} = 31.3 & Y_{\text{train},3} = 79.9 & Y_{\text{train},4} = 101.9 & Y_{\text{train},5} = 205.5 \end{array}$$

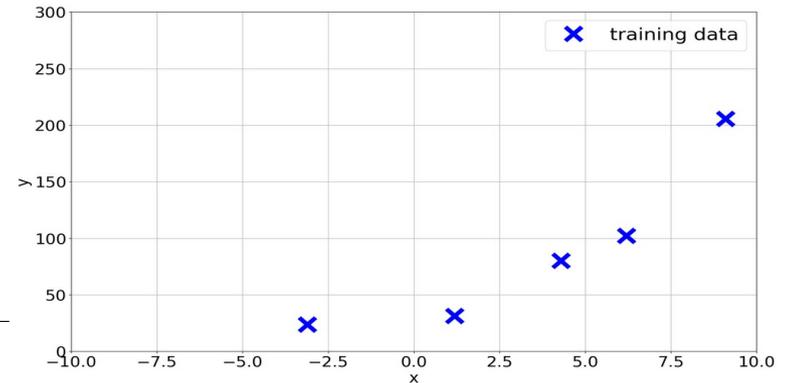
Choix de la fonction $f(x; \theta) = \sum_{i=0}^d \theta_i x^i = \theta^\top \phi(x)$

où $\phi(x) = [1 \quad x \quad \dots \quad x^d]^\top$

hyper-paramètre : d \longrightarrow **degré du polynôme**

Apprentissage

Choix du coût $l(y, s) = (y - s)^2$



III)

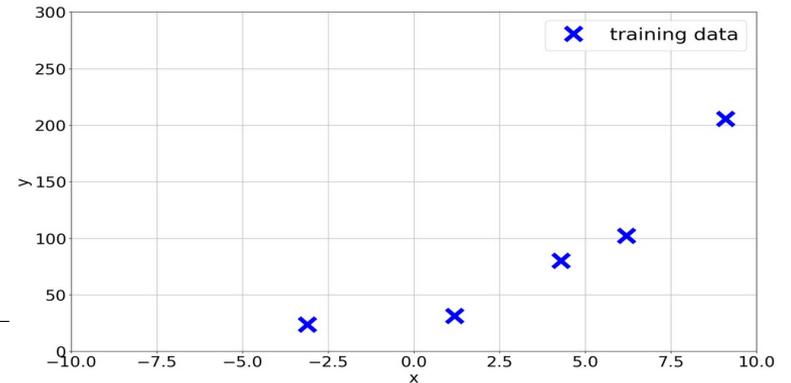
Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

$X_{\text{train},1} = -3.1$	$X_{\text{train},2} = 1.2$	$X_{\text{train},3} = 4.3$	$X_{\text{train},4} = 6.2$	$X_{\text{train},5} = 9.1$
$Y_{\text{train},1} = 23.7$	$Y_{\text{train},2} = 31.3$	$Y_{\text{train},3} = 79.9$	$Y_{\text{train},4} = 101.9$	$Y_{\text{train},5} = 205.5$

Choix de la fonction $f(x; \theta) = \sum_{i=0}^d \theta_i x^i = \theta^\top \phi(x)$

où $\phi(x) = [1 \quad x \quad \dots \quad x^d]^\top$

hyper-paramètre : d \longrightarrow **degré du polynôme**



Apprentissage

Choix du coût $l(y, s) = (y - s)^2$

Optimisation $\theta^* = \arg \min_{\theta} \sum_{i=1}^5 \left(Y_{\text{train},i} - \theta^\top \phi(X_{\text{train},i}) \right)^2 \longrightarrow$ **Moindres carrés linéaires**

III)

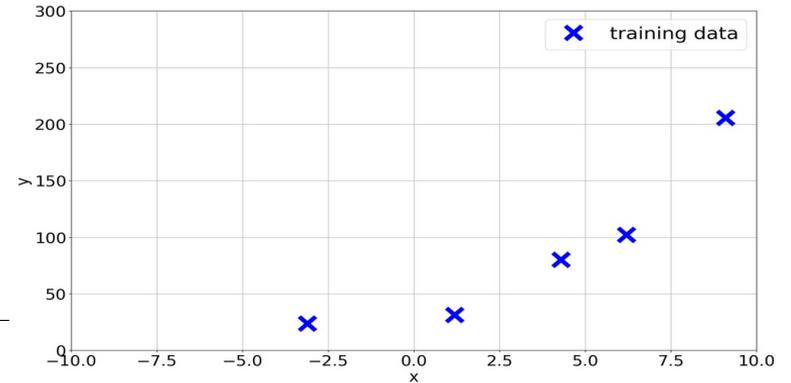
Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

$X_{\text{train},1} = -3.1$	$X_{\text{train},2} = 1.2$	$X_{\text{train},3} = 4.3$	$X_{\text{train},4} = 6.2$	$X_{\text{train},5} = 9.1$
$Y_{\text{train},1} = 23.7$	$Y_{\text{train},2} = 31.3$	$Y_{\text{train},3} = 79.9$	$Y_{\text{train},4} = 101.9$	$Y_{\text{train},5} = 205.5$

Choix de la fonction $f(x; \theta) = \sum_{i=0}^d \theta_i x^i = \theta^\top \phi(x)$

où $\phi(x) = [1 \quad x \quad \dots \quad x^d]^\top$

hyper-paramètre : d \longrightarrow **degré du polynôme**



Apprentissage

Choix du coût $l(y, s) = (y - s)^2$

Optimisation $\theta^* = \arg \min_{\theta} \sum_{i=1}^5 \left(Y_{\text{train},i} - \theta^\top \phi(X_{\text{train},i}) \right)^2 \longrightarrow$ **Moindres carrés linéaires**

Inférence $s_{\text{test}} = f(x; \theta^*) = \theta^{*\top} \phi(x_{\text{test}})$

III)

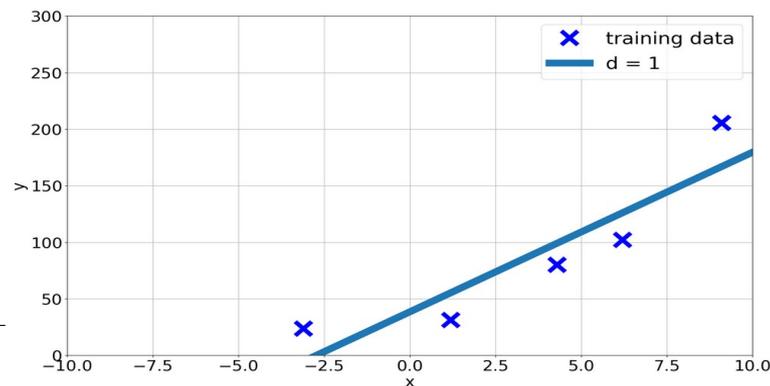
Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

$X_{\text{train},1} = -3.1$	$X_{\text{train},2} = 1.2$	$X_{\text{train},3} = 4.3$	$X_{\text{train},4} = 6.2$	$X_{\text{train},5} = 9.1$
$Y_{\text{train},1} = 23.7$	$Y_{\text{train},2} = 31.3$	$Y_{\text{train},3} = 79.9$	$Y_{\text{train},4} = 101.9$	$Y_{\text{train},5} = 205.5$

Choix de la fonction $f(x; \theta) = \sum_{i=0}^d \theta_i x^i = \theta^\top \phi(x)$

où $\phi(x) = [1 \quad x \quad \dots \quad x^d]^\top$

hyper-paramètre : d \longrightarrow **degré du polynôme**



Apprentissage

Choix du coût $l(y, o) = (y - o)^2$

Optimisation $\theta^* = \arg \min_{\theta} \sum_{i=1}^5 \left(Y_{\text{train},i} - \theta^\top \phi(X_{\text{train},i}) \right)^2 \longrightarrow$ **Moindres carrés linéaires**

Inférence $s_{\text{test}} = f(x; \theta^*) = \theta^{*\top} \phi(x_{\text{test}})$

III)

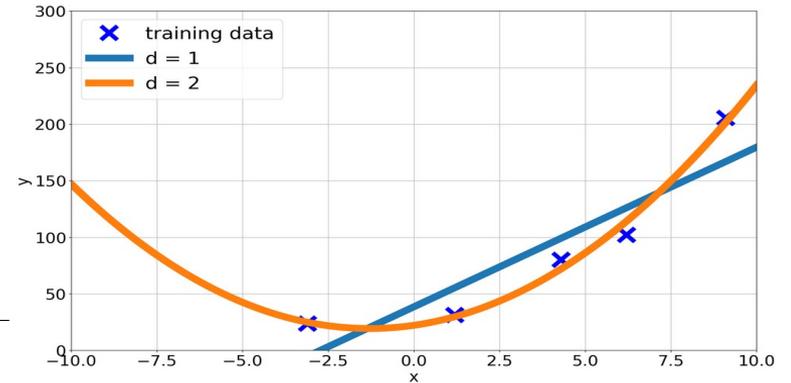
Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

$X_{\text{train},1} = -3.1$	$X_{\text{train},2} = 1.2$	$X_{\text{train},3} = 4.3$	$X_{\text{train},4} = 6.2$	$X_{\text{train},5} = 9.1$
$Y_{\text{train},1} = 23.7$	$Y_{\text{train},2} = 31.3$	$Y_{\text{train},3} = 79.9$	$Y_{\text{train},4} = 101.9$	$Y_{\text{train},5} = 205.5$

Choix de la fonction $f(x; \theta) = \sum_{i=0}^d \theta_i x^i = \theta^\top \phi(x)$

où $\phi(x) = [1 \quad x \quad \dots \quad x^d]^\top$

hyper-paramètre : d \longrightarrow **degré du polynôme**



Apprentissage

Choix du coût $l(y, o) = (y - o)^2$

Optimisation $\theta^* = \arg \min_{\theta} \sum_{i=1}^5 \left(Y_{\text{train},i} - \theta^\top \phi(X_{\text{train},i}) \right)^2 \longrightarrow$ **Moindres carrés linéaires**

Inférence $s_{\text{test}} = f(x; \theta^*) = \theta^{*\top} \phi(x_{\text{test}})$

III)

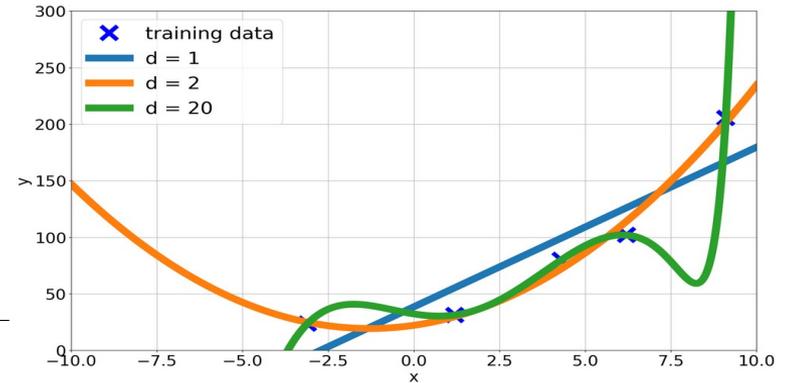
Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

$X_{\text{train},1} = -3.1$	$X_{\text{train},2} = 1.2$	$X_{\text{train},3} = 4.3$	$X_{\text{train},4} = 6.2$	$X_{\text{train},5} = 9.1$
$Y_{\text{train},1} = 23.7$	$Y_{\text{train},2} = 31.3$	$Y_{\text{train},3} = 79.9$	$Y_{\text{train},4} = 101.9$	$Y_{\text{train},5} = 205.5$

Choix de la fonction $f(x; \theta) = \sum_{i=0}^d \theta_i x^i = \theta^\top \phi(x)$

où $\phi(x) = [1 \quad x \quad \dots \quad x^d]^\top$

hyper-paramètre : d \longrightarrow **degré du polynôme**



Apprentissage

Choix du coût $l(y, o) = (y - o)^2$

Optimisation $\theta^* = \arg \min_{\theta} \sum_{i=1}^5 \left(Y_{\text{train},i} - \theta^\top \phi(X_{\text{train},i}) \right)^2 \longrightarrow$ **Moindres carrés linéaires**

Inférence $s_{\text{test}} = f(x; \theta^*) = \theta^{*\top} \phi(x_{\text{test}})$

III)

Avantage et inconvénients d'une approche paramétrique

Avantages

Inconvénients

III) **Avantage et inconvénients d'une approche paramétrique**

Avantages

- **Inférence efficace** → **pas d'accès** à la base de données étiquetées

Inconvénients

III) Avantage et inconvénients d'une approche paramétrique

Avantages

- **Inférence efficace** → **pas d'accès** à la base de données étiquetées
- **Temps d'inférence constant** → **ne dépend pas** de la taille de la base de données étiquetées

Inconvénients

III)

Avantage et inconvénients d'une approche paramétrique

Avantages

- **Inférence efficace** → **pas d'accès** à la base de données étiquetées
- **Temps d'inférence constant** → **ne dépend pas** de la taille de la base de données étiquetées

Inconvénients

- **Choix** de la fonction paramétrique et de ses **hyper-paramètres**

III)

Avantage et inconvénients d'une approche paramétrique

Avantages

- **Inférence efficace** → **pas d'accès** à la base de données étiquetées
- **Temps d'inférence constant** → **ne dépend pas** de la taille de la base de données étiquetées

Inconvénients

- **Choix** de la fonction paramétrique et de ses **hyper-paramètres**
- **Étape d'apprentissage** → souvent **longue** et **gourmande** en calculs

III)

Avantage et inconvénients d'une approche paramétrique

Avantages

- **Inférence efficace** → **pas d'accès** à la base de données étiquetées
- **Temps d'inférence constant** → **ne dépend pas** de la taille de la base de données étiquetées

Inconvénients

- **Choix** de la fonction paramétrique et de ses **hyper-paramètres**
- **Étape d'apprentissage** → souvent **longue** et **gourmande** en calculs
- **Difficile de modifier la base de données étiquetées** → nécessite de faire un **nouvel** apprentissage

IV) Réseaux de neurones

IV)

Choix de la fonction : Réseau de neurones

Réseau de neurones = Composition de fonctions paramétriques

Choix de la fonction : Réseau de neurones

Réseau de neurones = Composition de fonctions paramétriques

Mathématique
$$\mathbf{s} = f(\mathbf{x}; \boldsymbol{\theta}) = f_L(f_{L-1}(\dots f_2(f_1(\mathbf{x}; \boldsymbol{\theta}_1); \boldsymbol{\theta}_2) \dots); \boldsymbol{\theta}_{L-1}); \boldsymbol{\theta}_L)$$

Choix de la fonction : Réseau de neurones

Réseau de neurones = Composition de fonctions paramétriques

Mathématique $s = f(\mathbf{x}; \boldsymbol{\theta}) = f_L(f_{L-1}(\dots f_2(f_1(\mathbf{x}; \boldsymbol{\theta}_1); \boldsymbol{\theta}_2) \dots; \boldsymbol{\theta}_{L-1}); \boldsymbol{\theta}_L)$

Informatique
(Python)

```
def neuralNetwork_forward(x, theta, L):  
    x1 = f1(x, theta[0])  
    x2 = f2(x1, theta[1])  
    ...  
    s = fL(..., theta[L-1])  
    return s
```

IV)

Choix de la fonction : Réseau de neurones

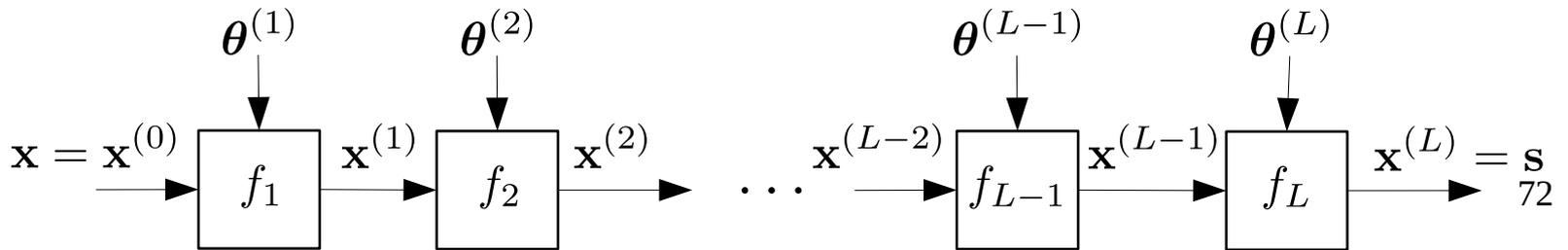
Réseau de neurones = Composition de fonctions paramétriques

Mathématique
$$s = f(\mathbf{x}; \boldsymbol{\theta}) = f_L(f_{L-1}(\dots f_2(f_1(\mathbf{x}; \boldsymbol{\theta}_1); \boldsymbol{\theta}_2) \dots; \boldsymbol{\theta}_{L-1}); \boldsymbol{\theta}_L)$$

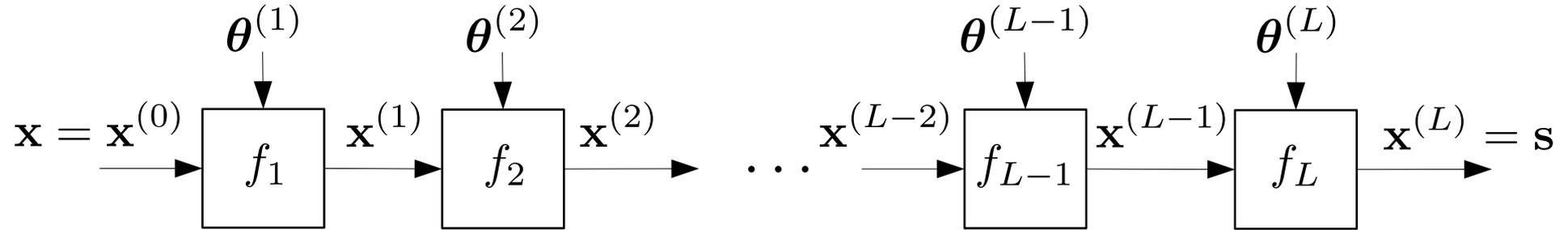
Informatique
(Python)

```
def neuralNetwork_forward(x, theta, L):  
    x1 = f1(x, theta[0])  
    x2 = f2(x1, theta[1])  
    ...  
    s = fL(..., theta[L-1])  
    return s
```

Graphique
(Graphe de calcul)

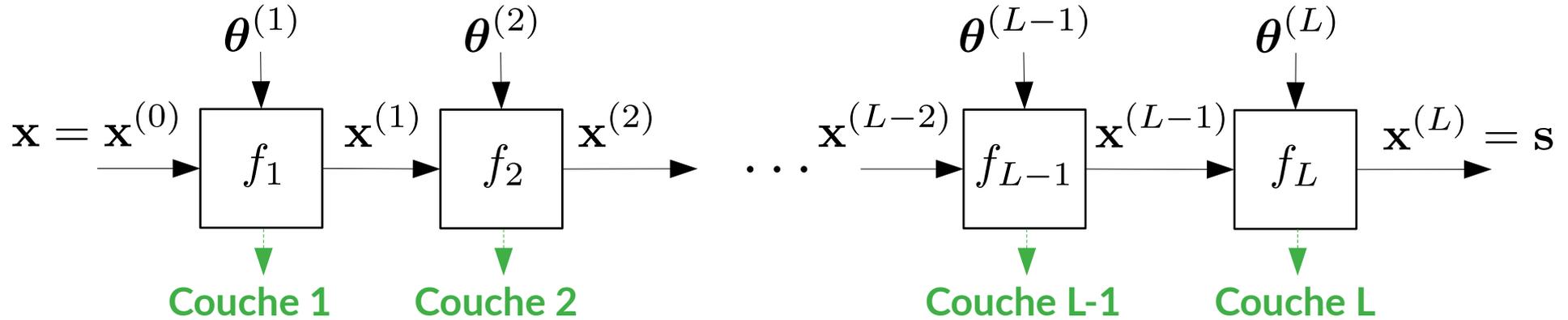


Réseau de neurones : Terminologie



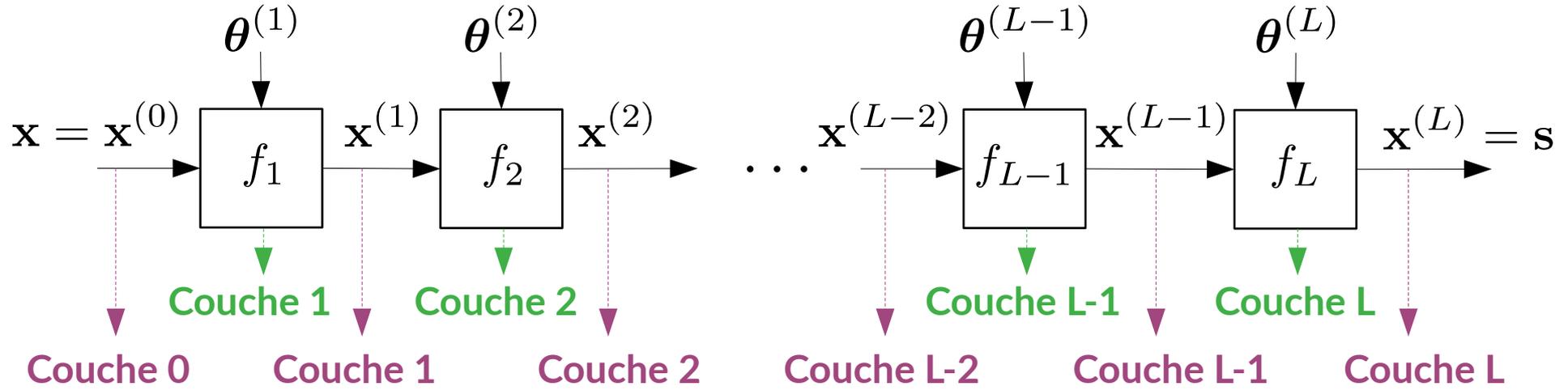
Couche (« layer » en anglais)

Réseau de neurones : Terminologie



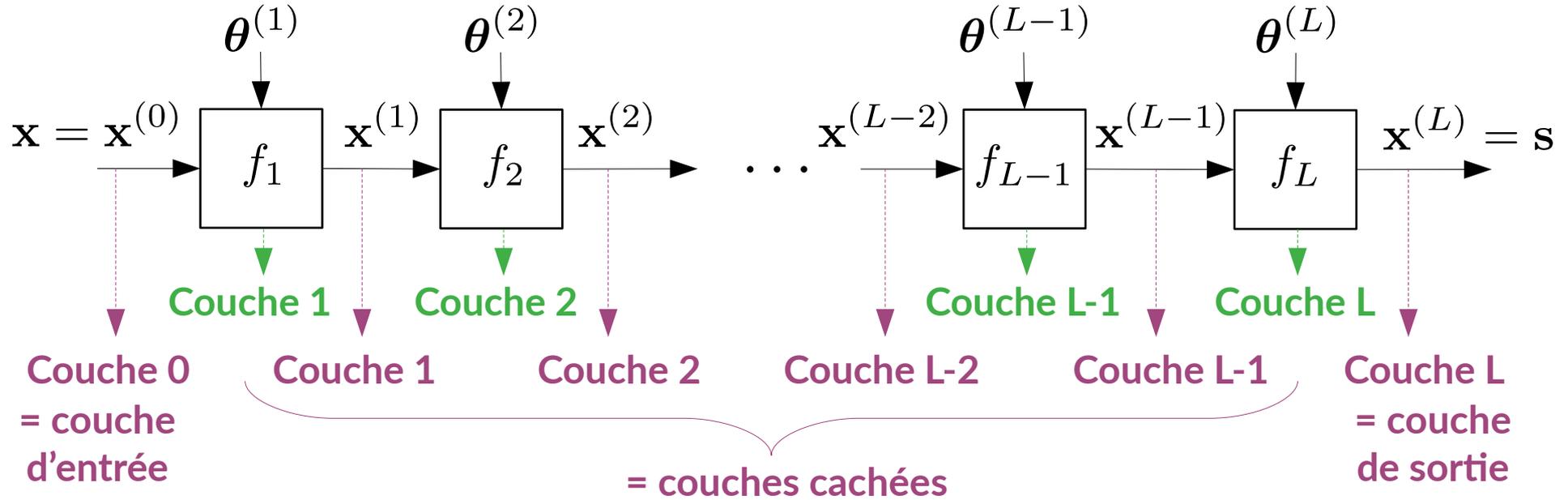
Couche (« layer » en anglais) $\xrightarrow{\text{Sens 1}}$ = une fonction paramétrique du réseau

Réseau de neurones : Terminologie



Couche (« layer » en anglais)
 Sens 1 → = une fonction paramétrique du réseau
 Sens 2 → = couche de « neurones » = un vecteur du réseau

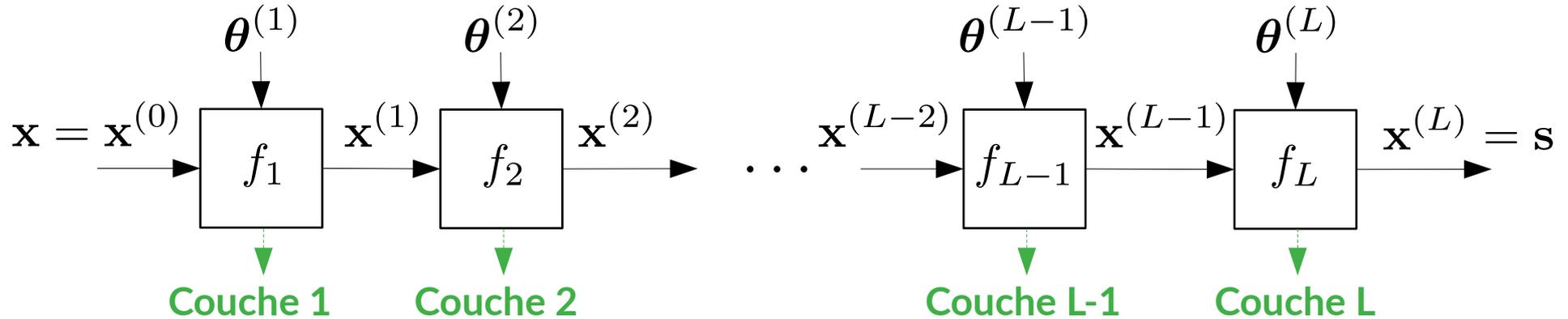
Réseau de neurones : Terminologie



Couche (« layer » en anglais)

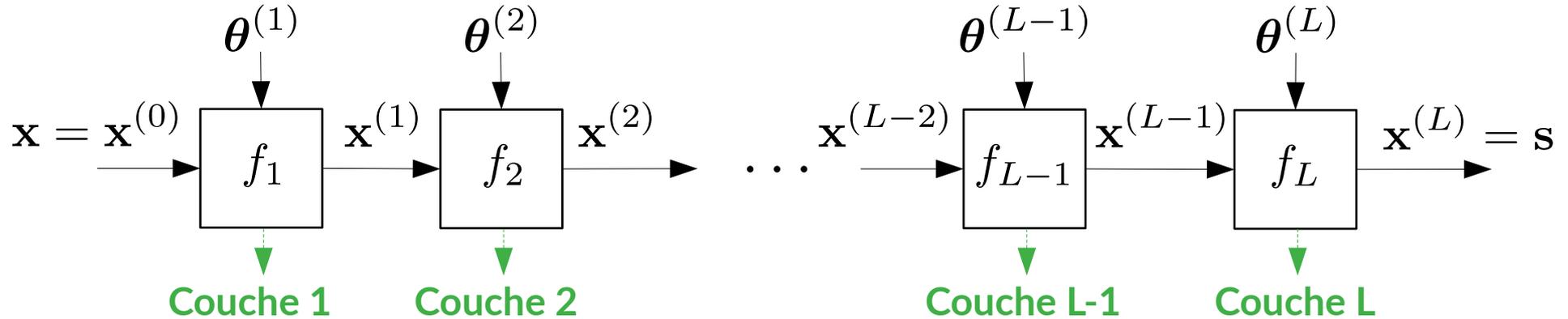
- Sens 1 → = une fonction paramétrique du réseau
- Sens 2 → = couche de « neurones » = un vecteur du réseau

Réseau de neurones : Terminologie (suite)



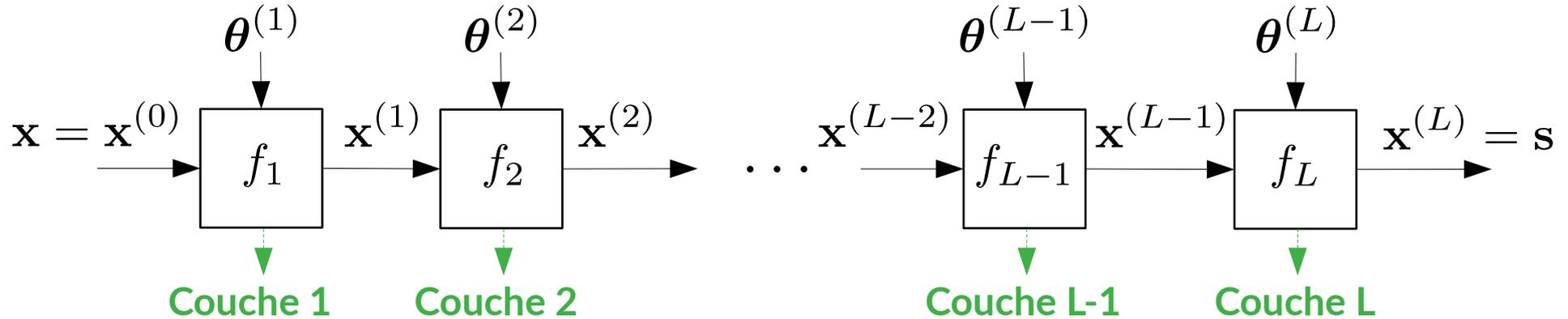
- **Profondeur** du réseau de neurones = nombre de couches

Réseau de neurones : Terminologie (suite)



- **Profondeur** du réseau de neurones = nombre de couches
- « **Deep Neural Network** » = réseau de neurones profond

Réseau de neurones : Terminologie (suite)



- **Profondeur** du réseau de neurones = nombre de couches
- « **Deep Neural Network** » = réseau de neurones profond
- **Architecture** du réseau de neurones = choix du nombre de couches, du type de chaque couche et de ses hyperparamètres, etc.

IV)

ImageNet Large Scale Visual Recognition Challenge 2012



ImageNet : 1.2 millions d'images annotées, 1000 classes

IV)

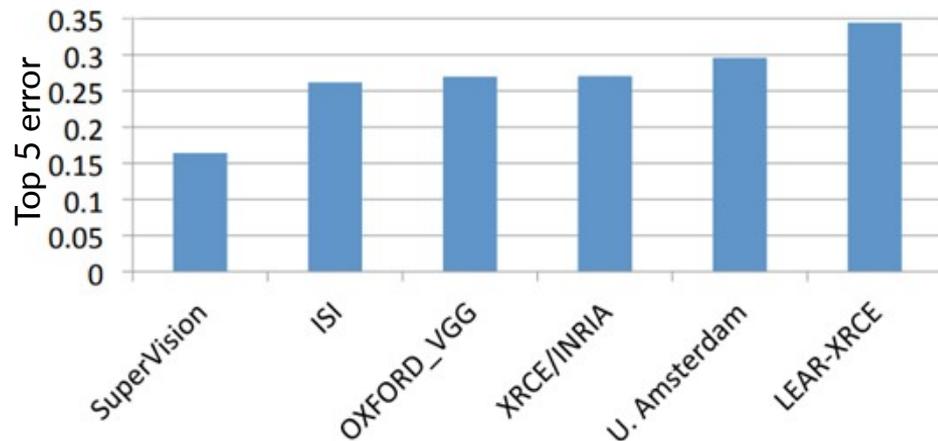
ImageNet Large Scale Visual Recognition Challenge 2012



ImageNet 1.2 millions d'images annotées, 1000 classes

Ingrédient 1 : grande base de données étiquetées

ImageNet Large Scale Visual Recognition Challenge 2012

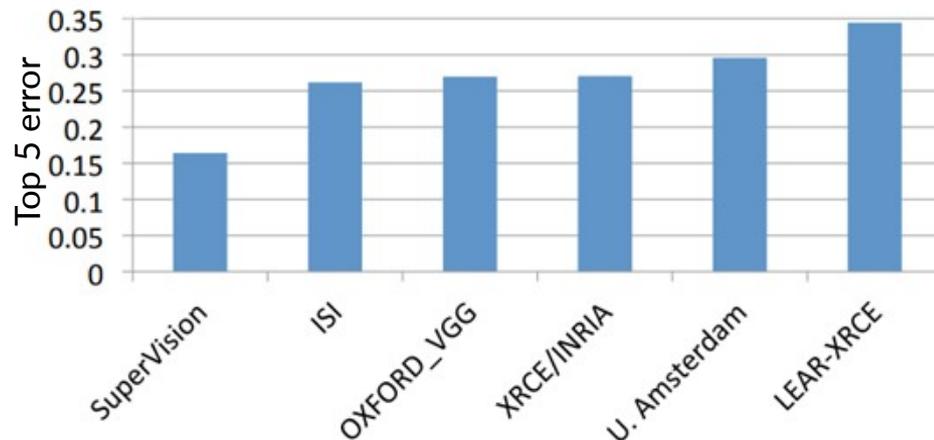


ImageNet : 1.2 millions d'images annotées, 1000 classes

SuperVision (A. Krizhevsky, I. Sutskever, G. Hinton, University of Toronto)

- Réseau de neurones à convolution (AlexNet)

ImageNet Large Scale Visual Recognition Challenge 2012



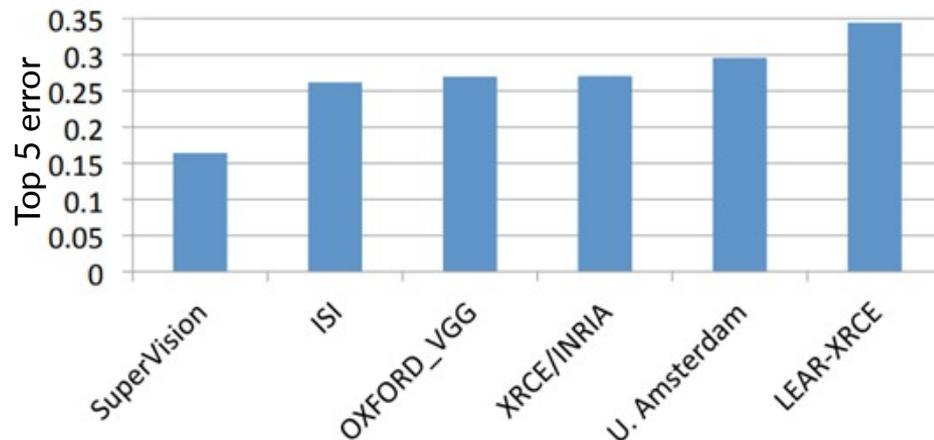
ImageNet : 1.2 millions d'images annotées, 1000 classes

SuperVision (A. Krizhevsky, I. Sutskever, G. Hinton, University of Toronto)

- Réseau de neurones à convolution (AlexNet)

Ingrédient 2 : fonction paramétrique de type « réseau de neurones » 83

ImageNet Large Scale Visual Recognition Challenge 2012

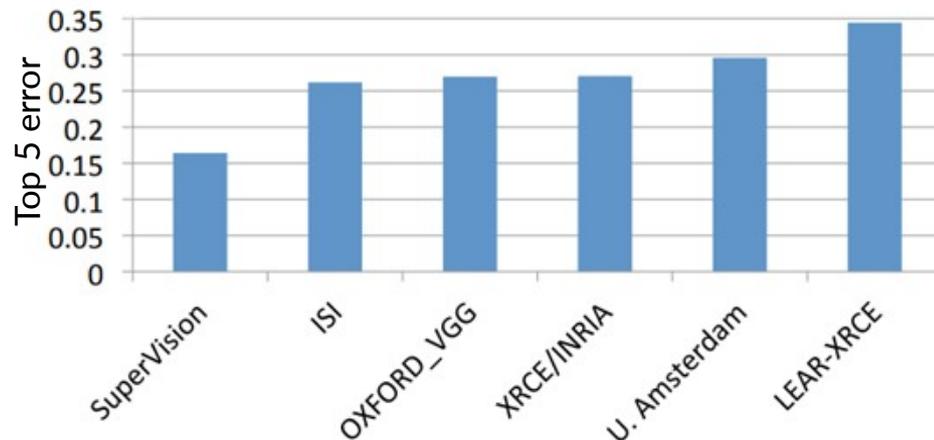


ImageNet : 1.2 millions d'images annotées, 1000 classes

SuperVision (A. Krizhevsky, I. Sutskever, G. Hinton, University of Toronto)

- Réseau de neurones à convolution (AlexNet)
- 62,3 millions de paramètres (pour information : GPT-3 → 175 000 millions de paramètres)

ImageNet Large Scale Visual Recognition Challenge 2012

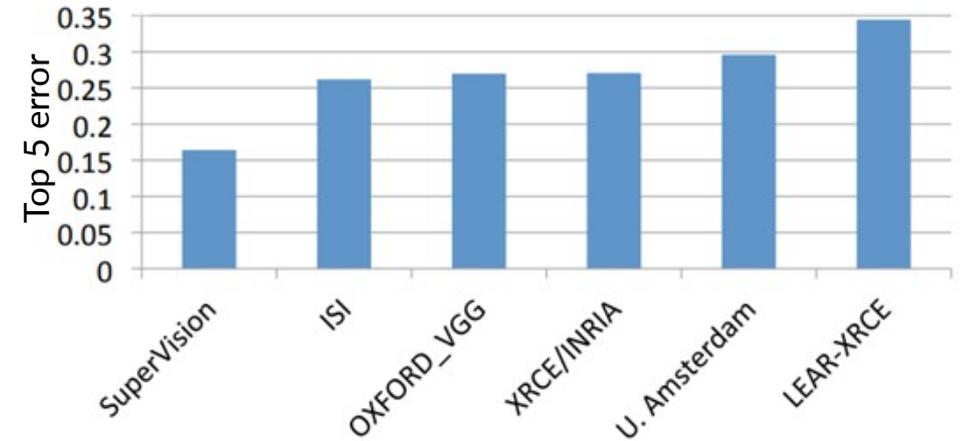


ImageNet : 1.2 millions d'images annotées, 1000 classes

SuperVision (A. Krizhevsky, I. Sutskever, G. Hinton, University of Toronto)

- Réseau de neurones à convolution (AlexNet)
- 62,3 millions de paramètres (pour information : GPT-3 → 175 000 millions de paramètres)
- 6 jours d'apprentissage sur 2 GPUs (GTX 580 3GB)

ImageNet Large Scale Visual Recognition Challenge 2012



ImageNet : 1.2 millions d'images annotées, 1000 classes

SuperVision (A. Krizhevsky, I. Sutskever, G. Hinton, University of Toronto)

- Réseau de neurones à convolution (AlexNet)
- 62,3 millions de paramètres (pour information : GPT-3 → 175 000 millions de paramètres)
- 6 jours d'apprentissage sur 2 GPUs (GTX 580 3GB)

Ingrédient 3 : grande capacité de calculs en parallèle

Résumé des ingrédients du « Deep Learning »

- 1) Grande base de données étiquetées
- 2) « Bonne » architecture de réseau de neurones profond
- 3) Grande capacité de calculs en parallèle (GPU)

Résumé des ingrédients du « Deep Learning »

1) Grande base de données étiquetées

2) « Bonne » architecture de réseau de neurones profond

↳ « Perceptron » multicouche, Réseau de neurones à convolution, Transformer

3) Grande capacité de calculs en parallèle (GPU)

Résumé des ingrédients du « Deep Learning »

1) Grande base de données étiquetées

2) « Bonne » architecture de réseau de neurones profond

↳ « **Perceptron** » **multicouche**, Réseau de neurones à convolution, Transformer

3) Grande capacité de calculs en parallèle (GPU)

IV)

Architecture : “Perceptron” multicouche (MLP)

Transformation affine = FC (“Fully Connected”)

$$\text{FC}(\mathbf{x}; \boldsymbol{\theta} = \{\mathbf{W}, \mathbf{b}\}) = \mathbf{W}\mathbf{x} + \mathbf{b}$$

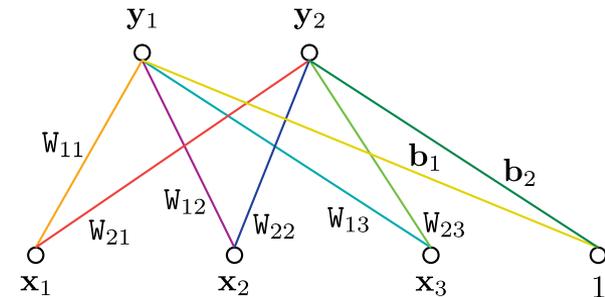
Architecture : “Perceptron” multicouche (MLP)

Transformation affine = FC (“Fully Connected”)

$$\text{FC}(\mathbf{x}; \boldsymbol{\theta} = \{W, \mathbf{b}\}) = W\mathbf{x} + \mathbf{b}$$

Pourquoi « Fully Connected » ?

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} & W_{13} & \mathbf{b}_1 \\ W_{21} & W_{22} & W_{23} & \mathbf{b}_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix}$$



Architecture : "Perceptron" multicouche (MLP)

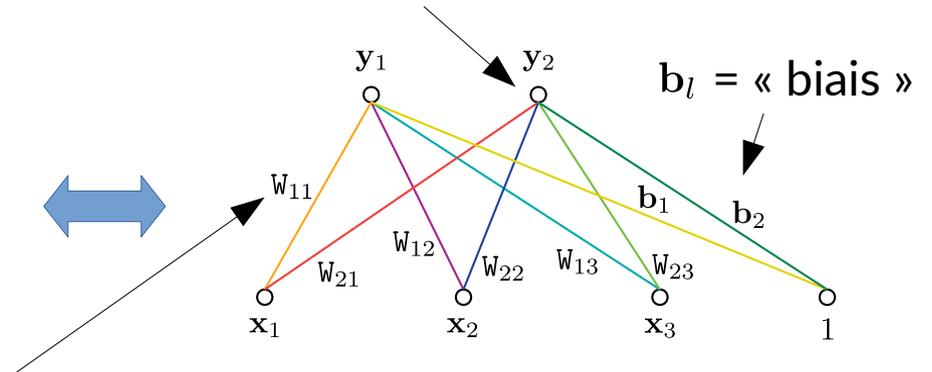
Transformation affine = FC ("Fully Connected")

$$\text{FC}(\mathbf{x}; \boldsymbol{\theta} = \{W, \mathbf{b}\}) = W\mathbf{x} + \mathbf{b}$$

Pourquoi « Fully Connected » ?

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} & W_{13} & b_1 \\ W_{21} & W_{22} & W_{23} & b_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix}$$

Élément d'un vecteur = « neurone »



W_{ij} = « poids » d'une connexion entre deux « neurones »

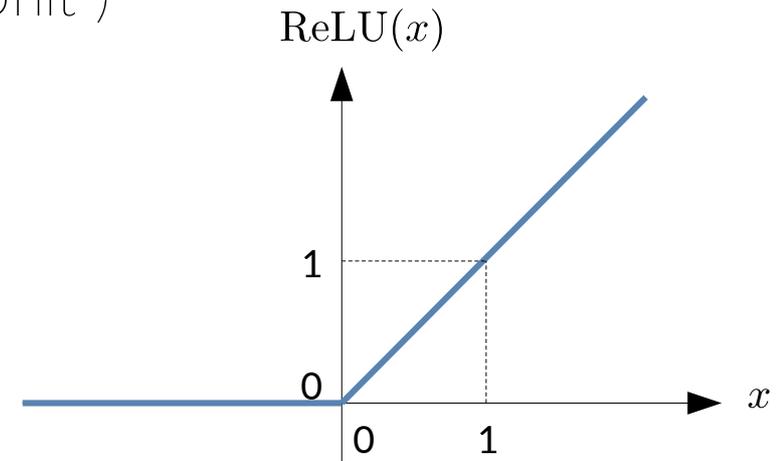
Architecture : “Perceptron” multicouche (MLP)

Transformation affine = FC (“Fully Connected”)

$$\text{FC}(\mathbf{x}; \boldsymbol{\theta} = \{\mathbf{W}, \mathbf{b}\}) = \mathbf{W}\mathbf{x} + \mathbf{b}$$

Non-linéarité = ReLU (“Rectified Linear Unit”)

$$\text{ReLU}(x) = \max(0, x)$$



Architecture : “Perceptron” multicouche (MLP)

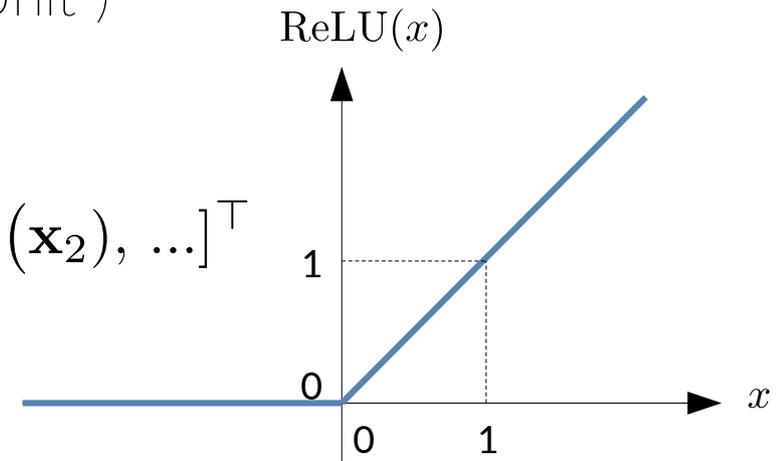
Transformation affine = FC (“Fully Connected”)

$$\text{FC}(\mathbf{x}; \boldsymbol{\theta} = \{W, \mathbf{b}\}) = W\mathbf{x} + \mathbf{b}$$

Non-linéarité = ReLU (“Rectified Linear Unit”)

$$\text{ReLU}(x) = \max(0, x)$$

Abus de notation : $\text{ReLU}(\mathbf{x}) = [\text{ReLU}(x_1), \text{ReLU}(x_2), \dots]^T$



Architecture : “Perceptron” multicouche (MLP)

Transformation affine = FC (“Fully Connected”)

$$\text{FC}(\mathbf{x}; \boldsymbol{\theta} = \{W, \mathbf{b}\}) = W\mathbf{x} + \mathbf{b}$$

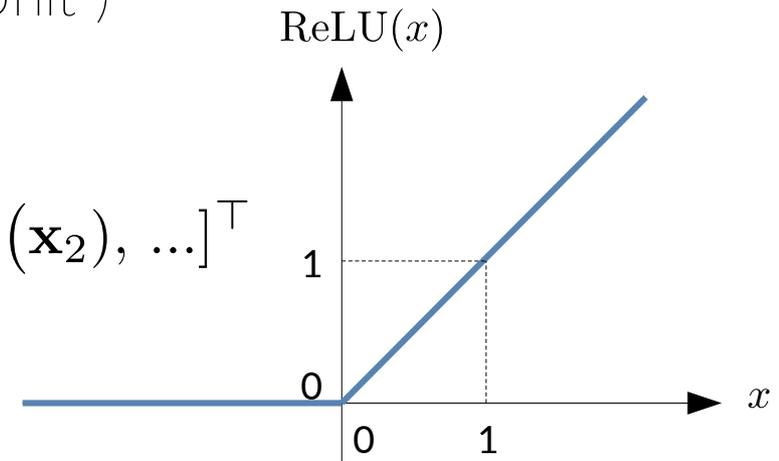
Non-linéarité = ReLU (“Rectified Linear Unit”)

$$\text{ReLU}(x) = \max(0, x)$$

Abus de notation : $\text{ReLU}(\mathbf{x}) = [\text{ReLU}(x_1), \text{ReLU}(x_2), \dots]^\top$



Terminologie : fonction d'activation



IV)

Architecture : “Perceptron” multicouche (MLP) (suite)

Mathématique

$$\mathbf{s} = \text{MLP}(\mathbf{x}; \boldsymbol{\theta}) = \text{FC}(\text{ReLU}(\dots \text{ReLU}(\text{FC}(\mathbf{x}; \boldsymbol{\theta}_1)) \dots); \boldsymbol{\theta}_L)$$

Architecture : “Perceptron” multicouche (MLP) (suite)

Mathématique

$$\mathbf{s} = \text{MLP}(\mathbf{x}; \boldsymbol{\theta}) = \text{FC}(\text{ReLU}(\dots \text{ReLU}(\text{FC}(\mathbf{x}; \boldsymbol{\theta}_1)) \dots); \boldsymbol{\theta}_L)$$

Informatique
(Python)

```
def MLP_forward(x, theta, L):  
    x1 = FC(x, theta[0])  
    x2 = ReLU(x1)  
    x3 = FC(x2, theta[2])  
    ...  
    s = FC(..., theta[L-1])  
    return s
```

IV)

Architecture : "Perceptron" multicouche (MLP) (suite)

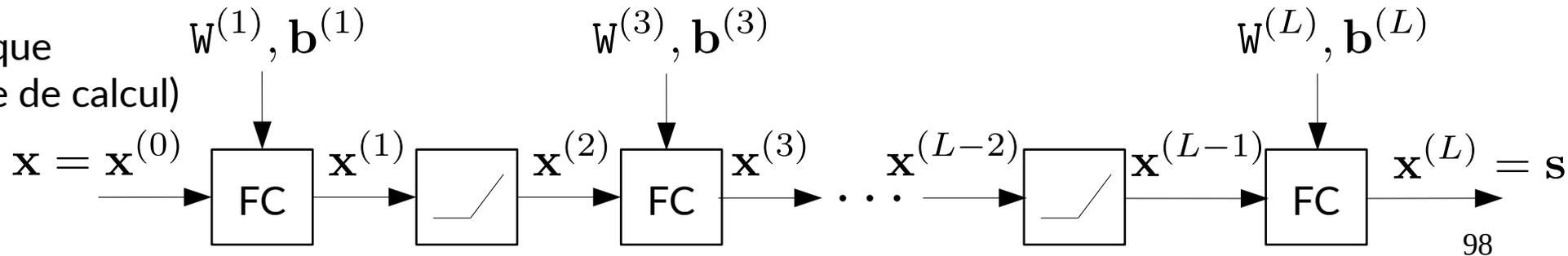
Mathématique

$$s = \text{MLP}(\mathbf{x}; \boldsymbol{\theta}) = \text{FC}(\text{ReLU}(\dots \text{ReLU}(\text{FC}(\mathbf{x}; \boldsymbol{\theta}_1)) \dots); \boldsymbol{\theta}_L)$$

Informatique
(Python)

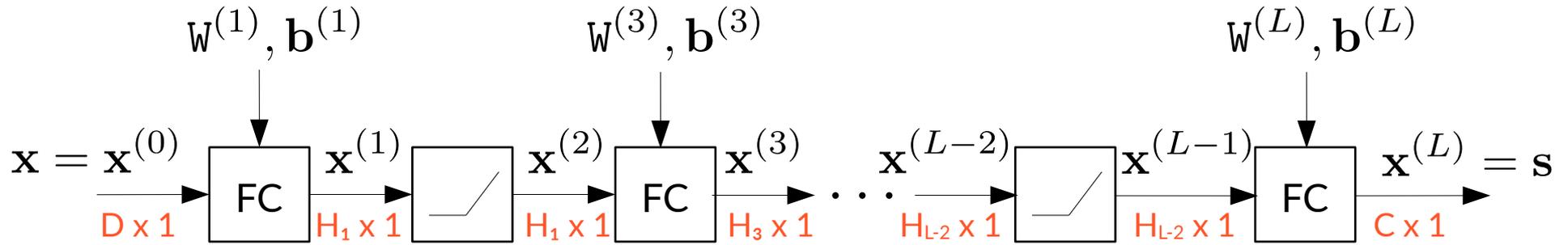
```
def MLP_forward(x, theta, L):  
    x1 = FC(x, theta[0])  
    x2 = ReLU(x1)  
    x3 = FC(x2, theta[2])  
    ...  
    s = FC(..., theta[L-1])  
    return s
```

Graphique
(Graphe de calcul)



IV)

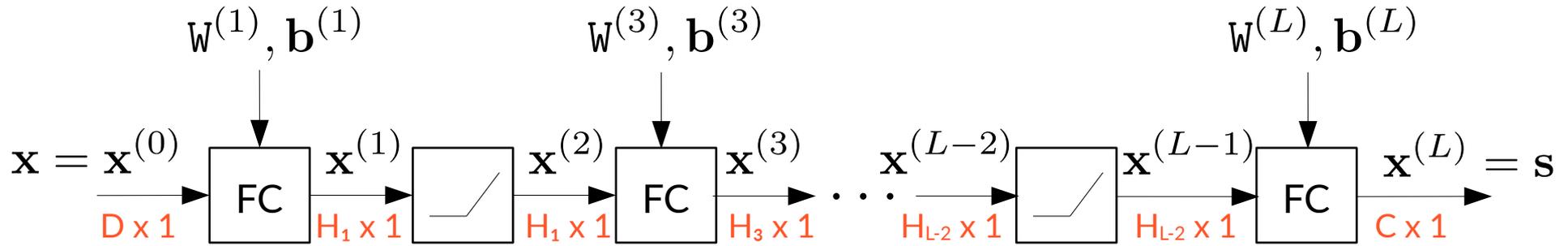
Architecture : "Perceptron" multicouche (MLP) (suite bis)



Paramètres : $\left\{ W^{(2j-1)}, \mathbf{b}^{(2j-1)} \right\}_{j=1, \dots, \lceil L/2 \rceil}$

IV)

Architecture : "Perceptron" multicouche (MLP) (suite bis)



Paramètres : $\left\{ W^{(2j-1)}, \mathbf{b}^{(2j-1)} \right\}_{j=1, \dots, \lceil L/2 \rceil}$

Hyper-paramètres : $L, \{ H_{2j-1} \}_{j=1, \dots, \lceil (L-2)/2 \rceil}$

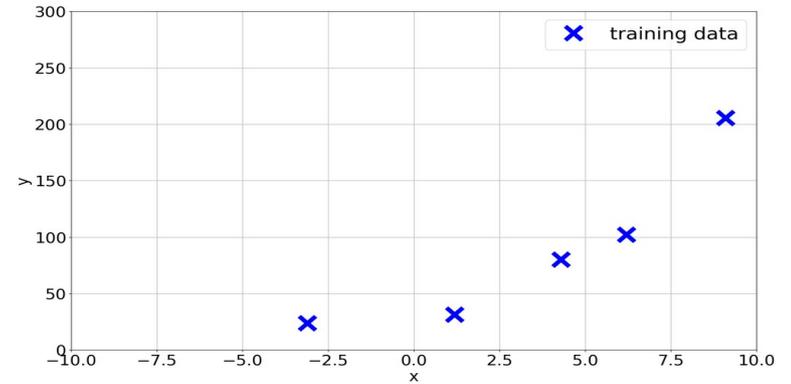
Nombre de couches

Dimension de chaque couche cachée

IV)

Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

$X_{\text{train},1} = -3.1$ $X_{\text{train},2} = 1.2$ $X_{\text{train},3} = 4.3$ $X_{\text{train},4} = 6.2$ $X_{\text{train},5} = 9.1$
 $Y_{\text{train},1} = 23.7$ $Y_{\text{train},2} = 31.3$ $Y_{\text{train},3} = 79.9$ $Y_{\text{train},4} = 101.9$ $Y_{\text{train},5} = 205.5$

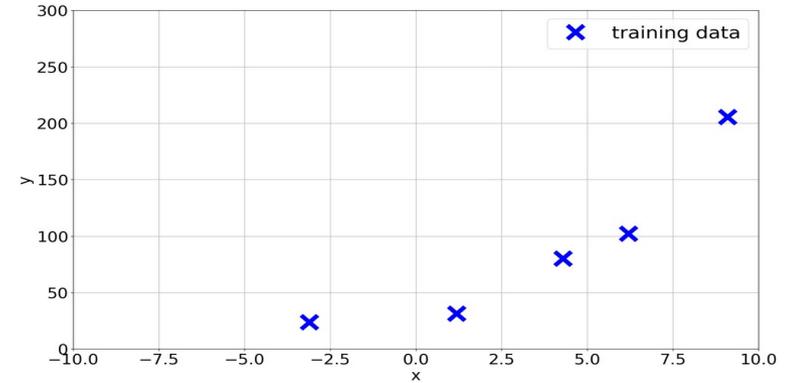
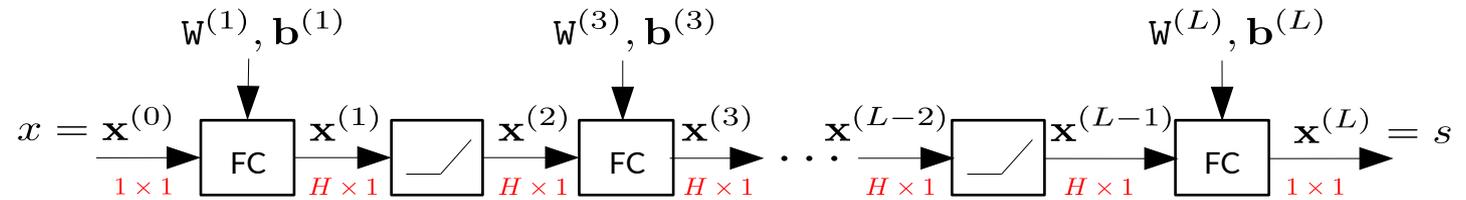


Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

$$X_{\text{train},1} = -3.1 \quad X_{\text{train},2} = 1.2 \quad X_{\text{train},3} = 4.3 \quad X_{\text{train},4} = 6.2 \quad X_{\text{train},5} = 9.1$$

$$Y_{\text{train},1} = 23.7 \quad Y_{\text{train},2} = 31.3 \quad Y_{\text{train},3} = 79.9 \quad Y_{\text{train},4} = 101.9 \quad Y_{\text{train},5} = 205.5$$

Choix de la fonction

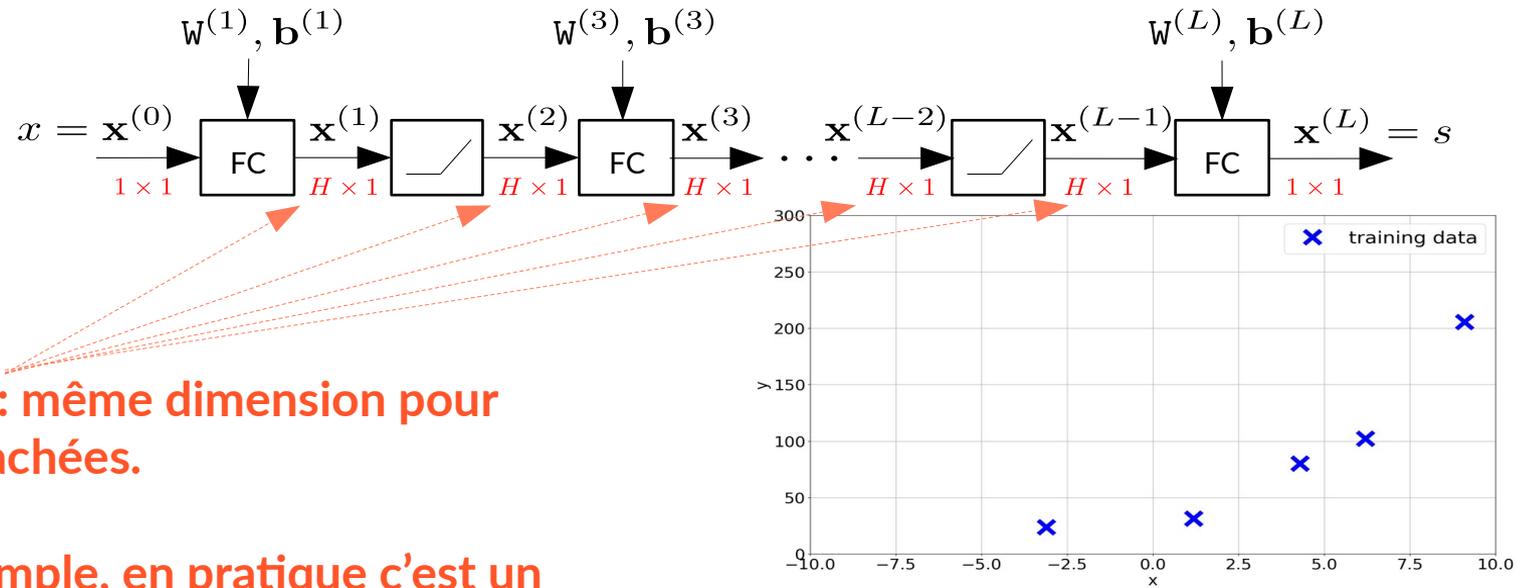


Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

$$X_{\text{train},1} = -3.1 \quad X_{\text{train},2} = 1.2 \quad X_{\text{train},3} = 4.3 \quad X_{\text{train},4} = 6.2 \quad X_{\text{train},5} = 9.1$$

$$Y_{\text{train},1} = 23.7 \quad Y_{\text{train},2} = 31.3 \quad Y_{\text{train},3} = 79.9 \quad Y_{\text{train},4} = 101.9 \quad Y_{\text{train},5} = 205.5$$

Choix de la fonction



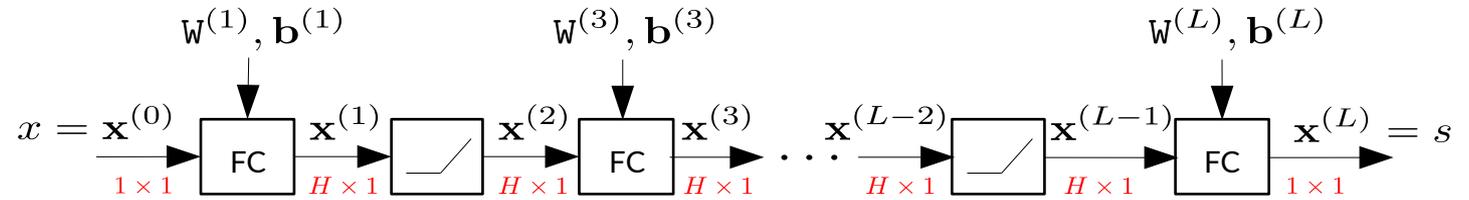
Choix d'architecture : même dimension pour toutes les couches cachées.

C'est juste pour l'exemple, en pratique c'est un mauvais choix.

Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

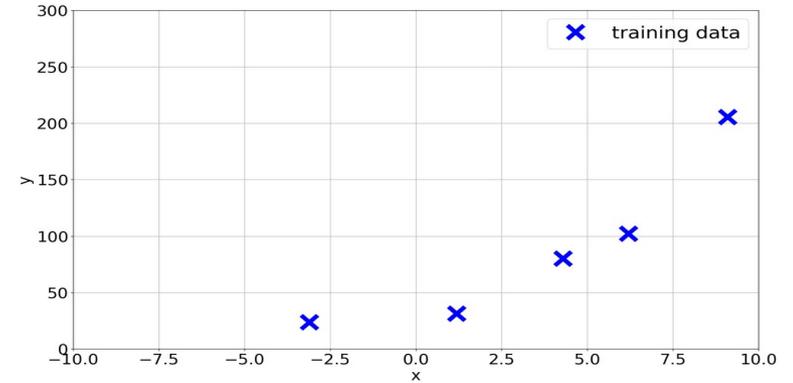
$$\begin{array}{cccccc}
 X_{\text{train},1} = -3.1 & X_{\text{train},2} = 1.2 & X_{\text{train},3} = 4.3 & X_{\text{train},4} = 6.2 & X_{\text{train},5} = 9.1 \\
 Y_{\text{train},1} = 23.7 & Y_{\text{train},2} = 31.3 & Y_{\text{train},3} = 79.9 & Y_{\text{train},4} = 101.9 & Y_{\text{train},5} = 205.5
 \end{array}$$

Choix de la fonction



$$f(x; \theta) = \text{MLP} \left(x; \theta = \{W^{(l)}, \mathbf{b}^{(l)}\}_l \right)$$

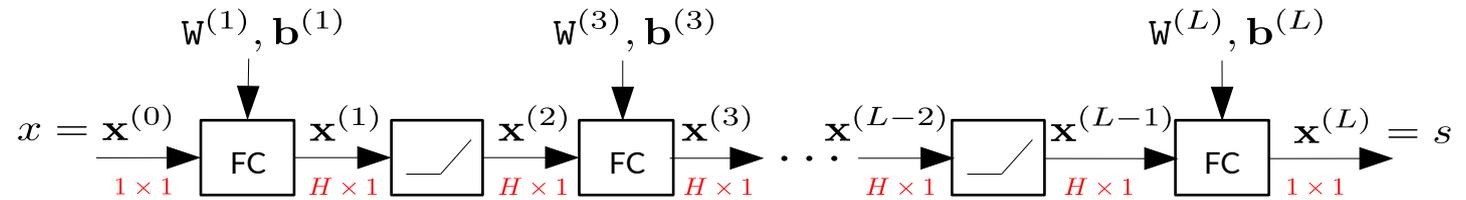
hyper-paramètres : L, H



Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

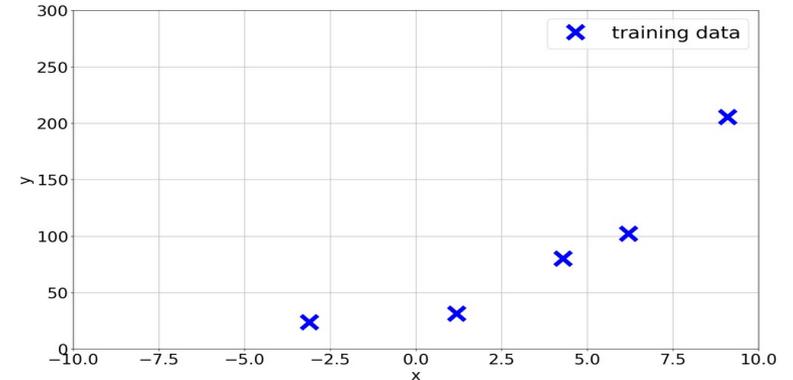
$$\begin{array}{ccccccccc} X_{\text{train},1} = -3.1 & X_{\text{train},2} = 1.2 & X_{\text{train},3} = 4.3 & X_{\text{train},4} = 6.2 & X_{\text{train},5} = 9.1 \\ Y_{\text{train},1} = 23.7 & Y_{\text{train},2} = 31.3 & Y_{\text{train},3} = 79.9 & Y_{\text{train},4} = 101.9 & Y_{\text{train},5} = 205.5 \end{array}$$

Choix de la fonction



$$f(x; \theta) = \text{MLP} \left(x; \theta = \{W^{(l)}, \mathbf{b}^{(l)}\}_l \right)$$

hyper-paramètres : L, H

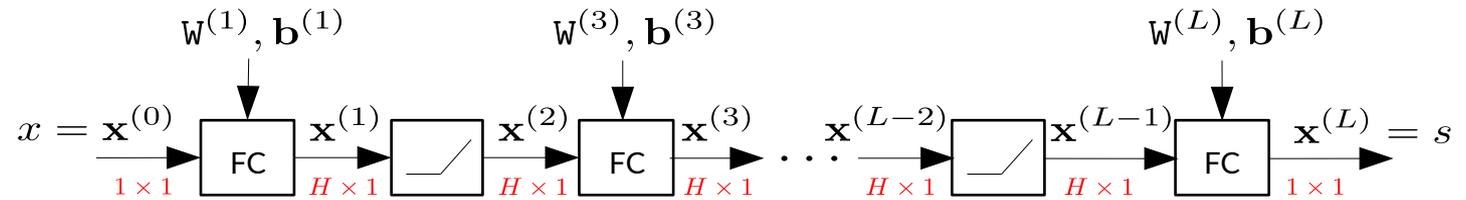


Quel est le nombre de paramètres du « modèle » ?

Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

$$\begin{array}{ccccccccc} X_{\text{train},1} = -3.1 & X_{\text{train},2} = 1.2 & X_{\text{train},3} = 4.3 & X_{\text{train},4} = 6.2 & X_{\text{train},5} = 9.1 \\ Y_{\text{train},1} = 23.7 & Y_{\text{train},2} = 31.3 & Y_{\text{train},3} = 79.9 & Y_{\text{train},4} = 101.9 & Y_{\text{train},5} = 205.5 \end{array}$$

Choix de la fonction

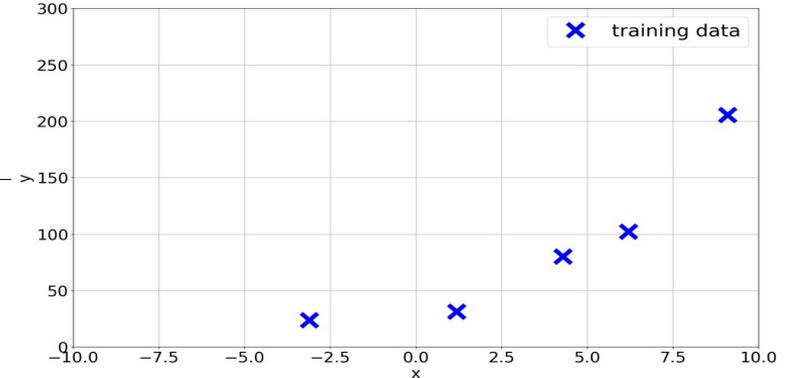


$$f(x; \theta) = \text{MLP} \left(x; \theta = \{W^{(l)}, \mathbf{b}^{(l)}\}_l \right)$$

hyper-paramètres : L, H

Apprentissage

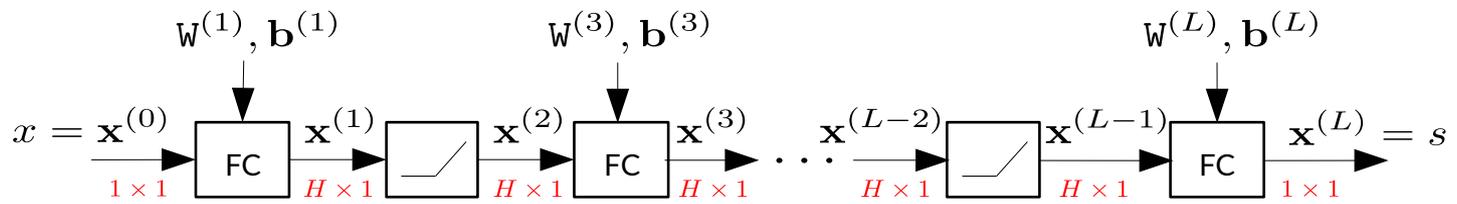
Choix du coût $l(y, s) = (y - s)^2$



Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

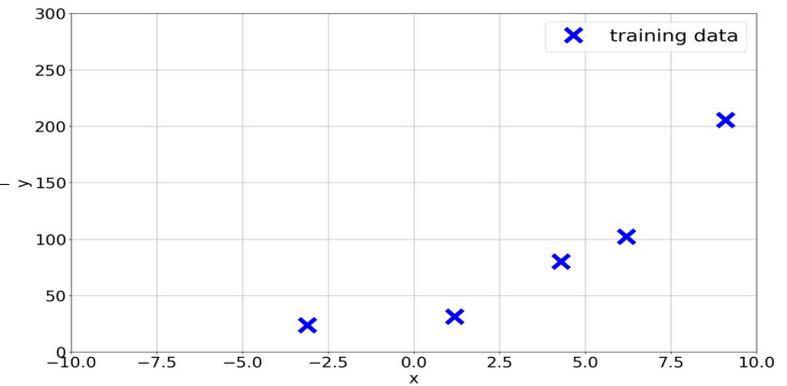
$X_{\text{train},1} = -3.1$ $X_{\text{train},2} = 1.2$ $X_{\text{train},3} = 4.3$ $X_{\text{train},4} = 6.2$ $X_{\text{train},5} = 9.1$
 $Y_{\text{train},1} = 23.7$ $Y_{\text{train},2} = 31.3$ $Y_{\text{train},3} = 79.9$ $Y_{\text{train},4} = 101.9$ $Y_{\text{train},5} = 205.5$

Choix de la fonction



$$f(x; \theta) = \text{MLP} \left(x; \theta = \{W^{(l)}, b^{(l)}\}_l \right)$$

hyper-paramètres : L, H



Apprentissage

Choix du coût $l(y, s) = (y - s)^2$

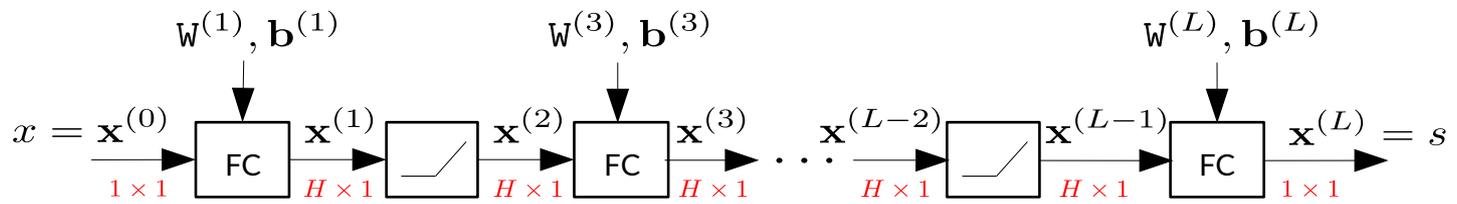
Optimisation $\theta^* = \arg \min_{\theta} \sum_{i=1}^5 (Y_{\text{train},i} - \text{MLP}(X_{\text{train},i}; \theta))^2$

Moindres carrés non linéaires

Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

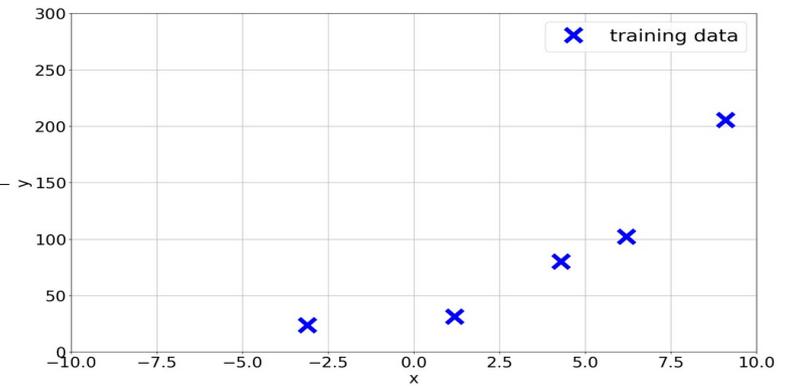
$X_{\text{train},1} = -3.1$ $X_{\text{train},2} = 1.2$ $X_{\text{train},3} = 4.3$ $X_{\text{train},4} = 6.2$ $X_{\text{train},5} = 9.1$
 $Y_{\text{train},1} = 23.7$ $Y_{\text{train},2} = 31.3$ $Y_{\text{train},3} = 79.9$ $Y_{\text{train},4} = 101.9$ $Y_{\text{train},5} = 205.5$

Choix de la fonction



$$f(x; \theta) = \text{MLP}(x; \theta = \{W^{(l)}, b^{(l)}\}_l)$$

hyper-paramètres : L, H



Apprentissage

Choix du coût $l(y, s) = \frac{(y - s)^2}{5}$

Optimisation $\theta^* = \arg \min_{\theta} \sum_{i=1}^5 (Y_{\text{train},i} - \text{MLP}(X_{\text{train},i}; \theta))^2$ ➔ Moindres carrés non linéaires

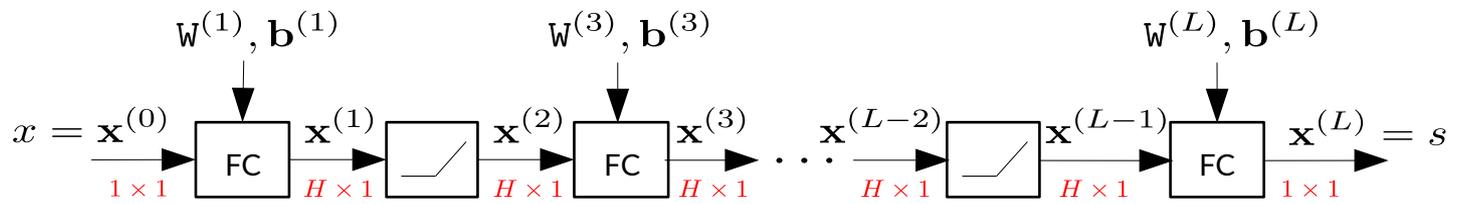
Inférence

$$s_{\text{test}} = \text{MLP}(x_{\text{test}}; \theta^*)$$

Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

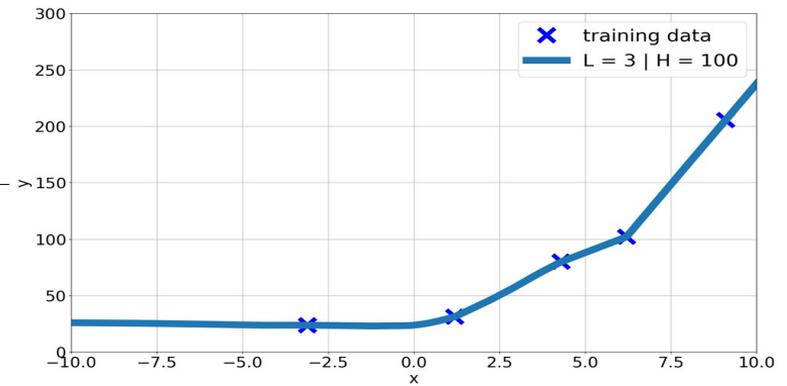
$X_{\text{train},1} = -3.1$ $X_{\text{train},2} = 1.2$ $X_{\text{train},3} = 4.3$ $X_{\text{train},4} = 6.2$ $X_{\text{train},5} = 9.1$
 $Y_{\text{train},1} = 23.7$ $Y_{\text{train},2} = 31.3$ $Y_{\text{train},3} = 79.9$ $Y_{\text{train},4} = 101.9$ $Y_{\text{train},5} = 205.5$

Choix de la fonction



$$f(x; \theta) = \text{MLP}(x; \theta = \{W^{(l)}, b^{(l)}\}_l)$$

hyper-paramètres : L, H



Apprentissage

Choix du coût $l(y, s) = (y - s)^2$

Optimisation $\theta^* = \arg \min_{\theta} \sum_{i=1}^5 (Y_{\text{train},i} - \text{MLP}(X_{\text{train},i}; \theta))^2$

Moindres carrés non linéaires

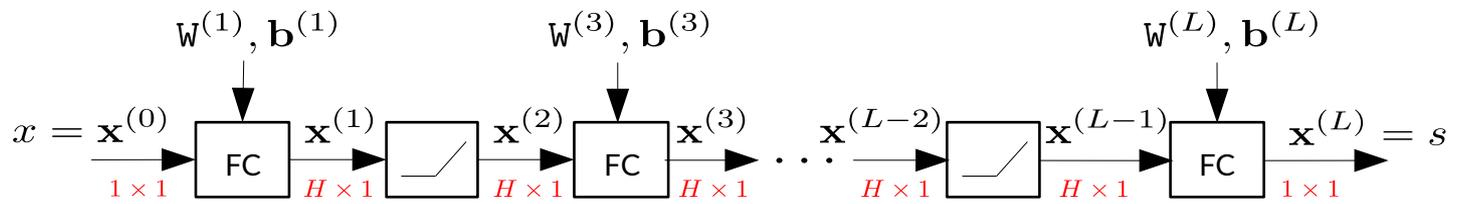
Inférence

$$s_{\text{test}} = \text{MLP}(x_{\text{test}}; \theta^*)$$

Exemple de régression : 5 données, $X \in \mathbb{R}$ et $Y \in \mathbb{R}$

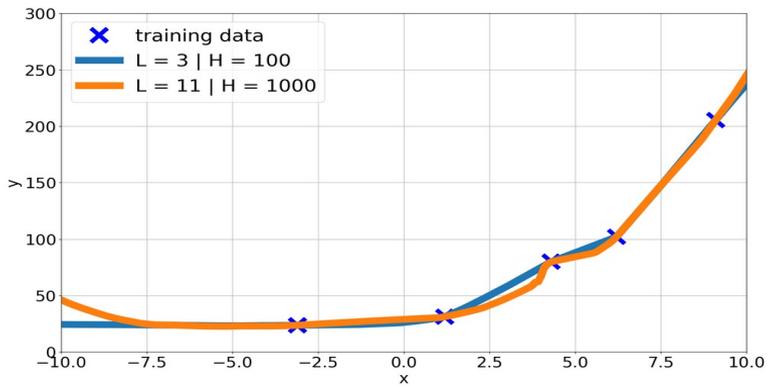
$X_{\text{train},1} = -3.1$ $X_{\text{train},2} = 1.2$ $X_{\text{train},3} = 4.3$ $X_{\text{train},4} = 6.2$ $X_{\text{train},5} = 9.1$
 $Y_{\text{train},1} = 23.7$ $Y_{\text{train},2} = 31.3$ $Y_{\text{train},3} = 79.9$ $Y_{\text{train},4} = 101.9$ $Y_{\text{train},5} = 205.5$

Choix de la fonction



$$f(x; \theta) = \text{MLP}(x; \theta = \{w^{(l)}, b^{(l)}\}_l)$$

hyper-paramètres : L, H



Apprentissage

Choix du coût $l(y, s) = (y - s)^2$

Optimisation $\theta^* = \arg \min_{\theta} \sum_{i=1}^5 (Y_{\text{train},i} - \text{MLP}(X_{\text{train},i}; \theta))^2$

Moindres carrés non linéaires

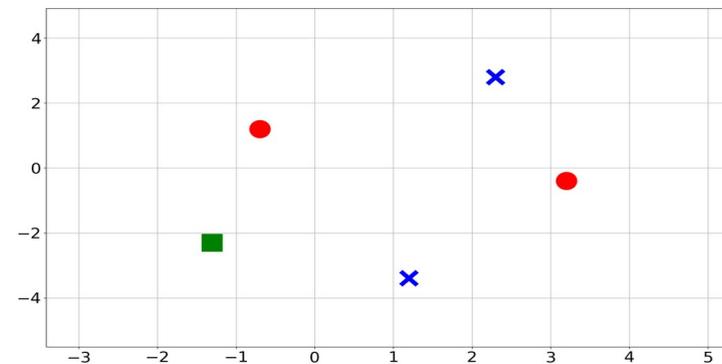
Inférence

$$s_{\text{test}} = \text{MLP}(x_{\text{test}}; \theta^*)$$

IV)

Exemple de classification : 5 données, $X \in \mathbb{R}^2$ et $Y \in \{0, 1, 2\}$

$$\begin{array}{lllll} X_{\text{train},1} = [1.2, -3.4] & X_{\text{train},2} = [2.3, 2.8] & X_{\text{train},3} = [-0.7, 1.2] & X_{\text{train},4} = [3.2, -0.4] & X_{\text{train},5} = [-1.3, 2.3] \\ Y_{\text{train},1} = 0 & Y_{\text{train},2} = 0 & Y_{\text{train},3} = 1 & Y_{\text{train},4} = 1 & Y_{\text{train},5} = 2 \end{array}$$

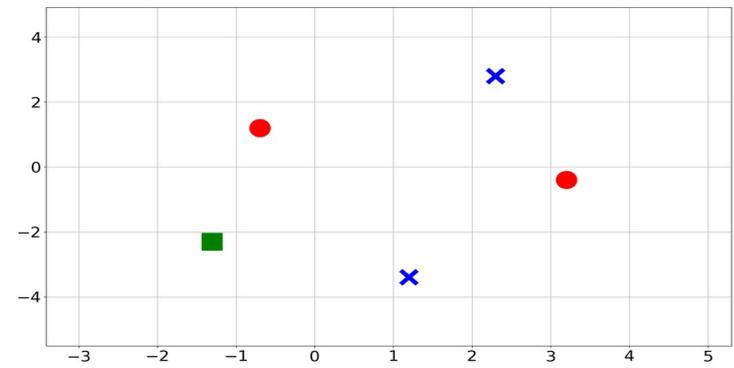
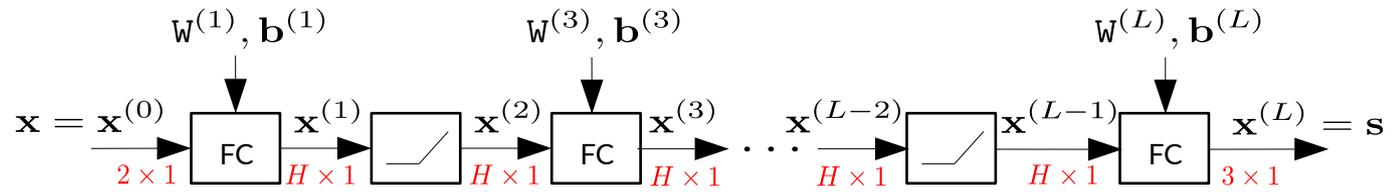


IV)

Exemple de classification : 5 données, $X \in \mathbb{R}^2$ et $Y \in \{0, 1, 2\}$

$X_{\text{train},1} = [1.2, -3.4]$ $X_{\text{train},2} = [2.3, 2.8]$ $X_{\text{train},3} = [-0.7, 1.2]$ $X_{\text{train},4} = [3.2, -0.4]$ $X_{\text{train},5} = [-1.3, 2.3]$
 $Y_{\text{train},1} = 0$ $Y_{\text{train},2} = 0$ $Y_{\text{train},3} = 1$ $Y_{\text{train},4} = 1$ $Y_{\text{train},5} = 2$

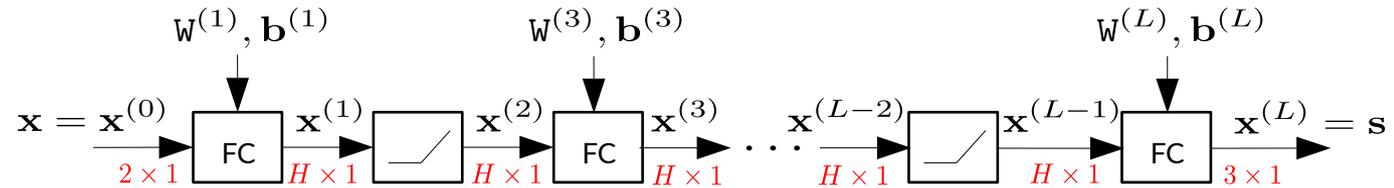
Choix de la fonction



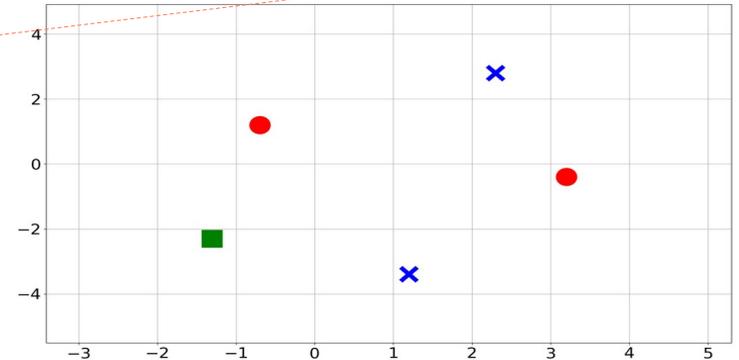
Exemple de classification : 5 données, $X \in \mathbb{R}^2$ et $Y \in \{0, 1, 2\}$

$$\begin{array}{lllll} X_{\text{train},1} = [1.2, -3.4] & X_{\text{train},2} = [2.3, 2.8] & X_{\text{train},3} = [-0.7, 1.2] & X_{\text{train},4} = [3.2, -0.4] & X_{\text{train},5} = [-1.3, 2.3] \\ Y_{\text{train},1} = 0 & Y_{\text{train},2} = 0 & Y_{\text{train},3} = 1 & Y_{\text{train},4} = 1 & Y_{\text{train},5} = 2 \end{array}$$

Choix de la fonction



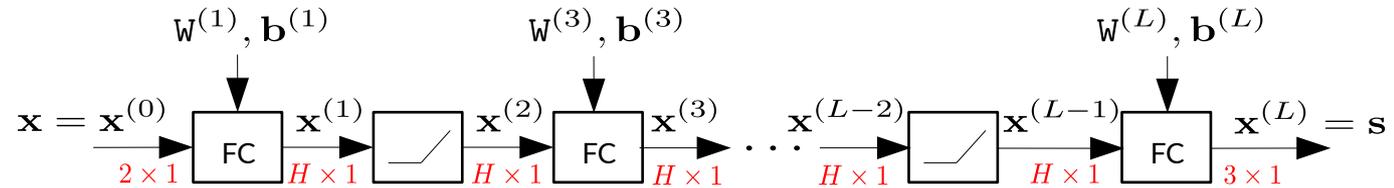
Problème : on souhaite prédire une valeur discrète, mais la sortie du MLP est continue.



Exemple de classification : 5 données, $X \in \mathbb{R}^2$ et $Y \in \{0, 1, 2\}$

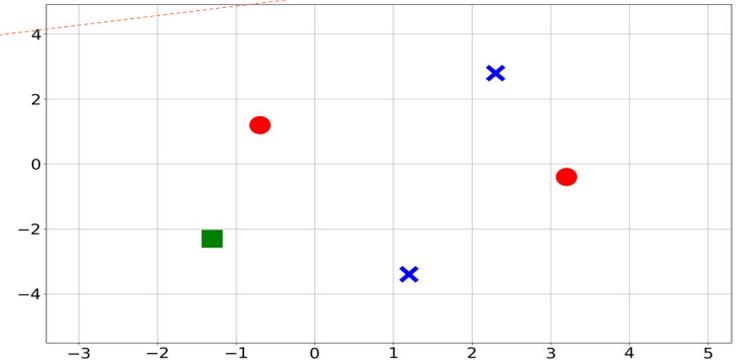
$$\begin{array}{lllll} X_{\text{train},1} = [1.2, -3.4] & X_{\text{train},2} = [2.3, 2.8] & X_{\text{train},3} = [-0.7, 1.2] & X_{\text{train},4} = [3.2, -0.4] & X_{\text{train},5} = [-1.3, 2.3] \\ Y_{\text{train},1} = 0 & Y_{\text{train},2} = 0 & Y_{\text{train},3} = 1 & Y_{\text{train},4} = 1 & Y_{\text{train},5} = 2 \end{array}$$

Choix de la fonction



Problème : on souhaite prédire une valeur discrète, mais la sortie du MLP est continue.

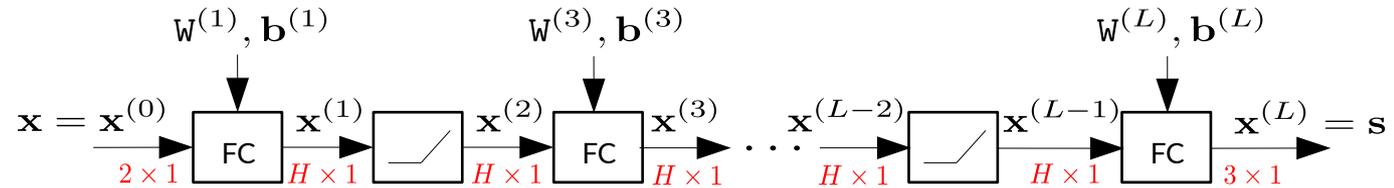
Solution : on prédit un score par classe, et la classe prédite sera celle du score maximum.



Exemple de classification : 5 données, $X \in \mathbb{R}^2$ et $Y \in \{0, 1, 2\}$

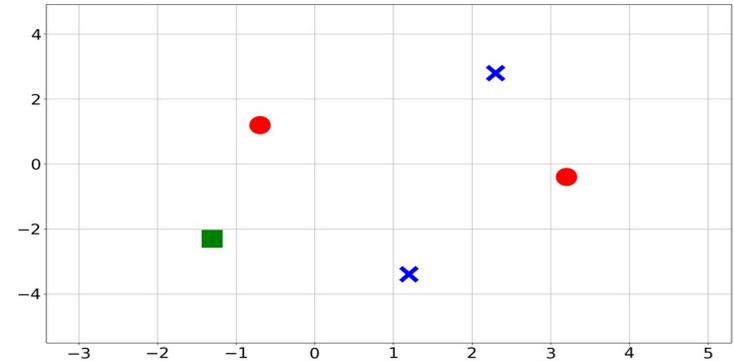
$$\begin{array}{ccccc} X_{\text{train},1} = [1.2, -3.4] & X_{\text{train},2} = [2.3, 2.8] & X_{\text{train},3} = [-0.7, 1.2] & X_{\text{train},4} = [3.2, -0.4] & X_{\text{train},5} = [-1.3, 2.3] \\ Y_{\text{train},1} = 0 & Y_{\text{train},2} = 0 & Y_{\text{train},3} = 1 & Y_{\text{train},4} = 1 & Y_{\text{train},5} = 2 \end{array}$$

Choix de la fonction



$$f(x; \theta) = \text{MLP} \left(x; \theta = \{W^{(l)}, \mathbf{b}^{(l)}\}_l \right)$$

hyper-paramètres : L, H

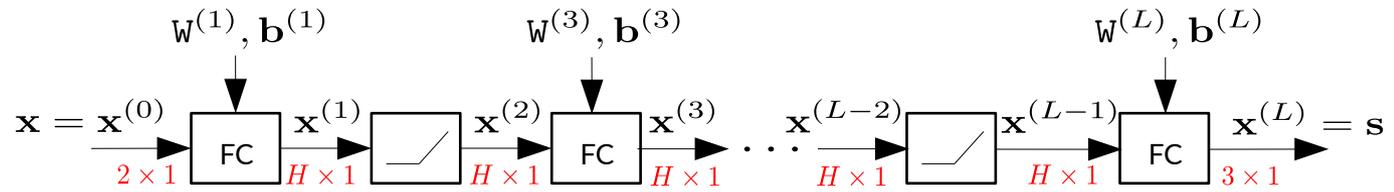


IV)

Exemple de classification : 5 données, $X \in \mathbb{R}^2$ et $Y \in \{0, 1, 2\}$

$X_{\text{train},1} = [1.2, -3.4]$	$X_{\text{train},2} = [2.3, 2.8]$	$X_{\text{train},3} = [-0.7, 1.2]$	$X_{\text{train},4} = [3.2, -0.4]$	$X_{\text{train},5} = [-1.3, 2.3]$
$Y_{\text{train},1} = 0$	$Y_{\text{train},2} = 0$	$Y_{\text{train},3} = 1$	$Y_{\text{train},4} = 1$	$Y_{\text{train},5} = 2$

Choix de la fonction



$$f(x; \theta) = \text{MLP} \left(x; \theta = \{W^{(l)}, \mathbf{b}^{(l)}\}_l \right)$$

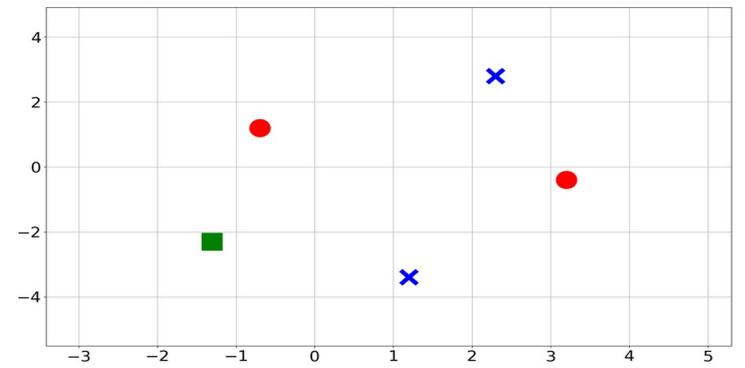
hyper-paramètres : L, H

Apprentissage

Choix du coût

$$l(y, \mathbf{s}) = -\ln(\text{softmax}(\mathbf{s})[y])$$

« Cross-entropy »

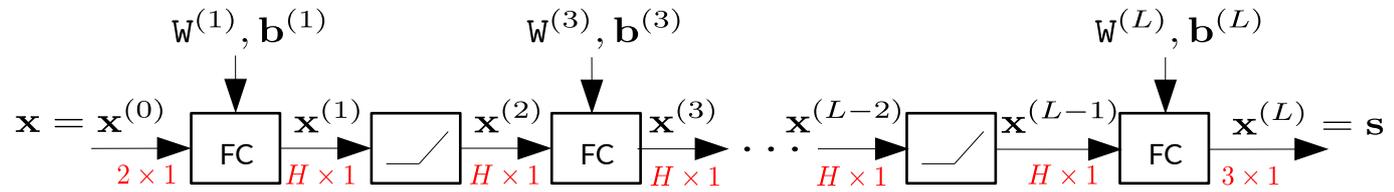


IV)

Exemple de classification : 5 données, $\mathbf{X} \in \mathbb{R}^2$ et $Y \in \{0, 1, 2\}$

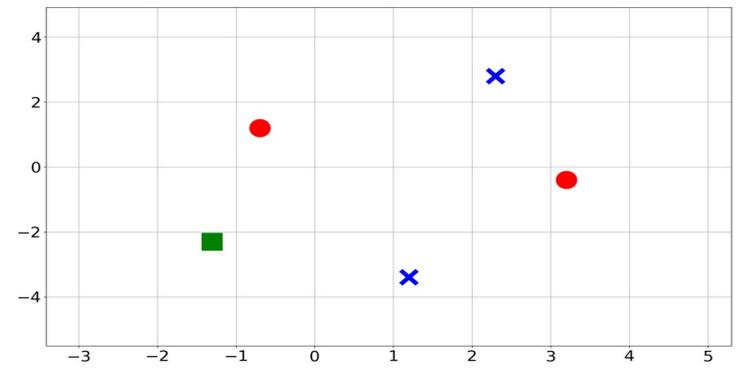
$\mathbf{X}_{\text{train},1} = [1.2, -3.4]$	$\mathbf{X}_{\text{train},2} = [2.3, 2.8]$	$\mathbf{X}_{\text{train},3} = [-0.7, 1.2]$	$\mathbf{X}_{\text{train},4} = [3.2, -0.4]$	$\mathbf{X}_{\text{train},5} = [-1.3, 2.3]$
$Y_{\text{train},1} = 0$	$Y_{\text{train},2} = 0$	$Y_{\text{train},3} = 1$	$Y_{\text{train},4} = 1$	$Y_{\text{train},5} = 2$

Choix de la fonction



$$f(x; \theta) = \text{MLP} \left(x; \theta = \{W^{(l)}, \mathbf{b}^{(l)}\}_l \right)$$

hyper-paramètres : L, H



Apprentissage

Choix du coût $l(y, \mathbf{s}) = -\ln(\text{softmax}(\mathbf{s})[y])$

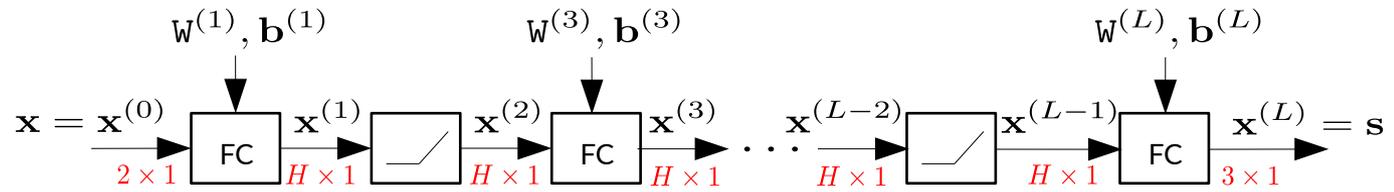
Optimisation $\theta^* = \arg \min_{\theta} \sum_{i=1}^5 -\ln(\text{softmax}(\text{MLP}(\mathbf{X}_{\text{train},i}; \theta)) [Y_{\text{train},i}])$

IV)

Exemple de classification : 5 données, $\mathbf{X} \in \mathbb{R}^2$ et $Y \in \{0, 1, 2\}$

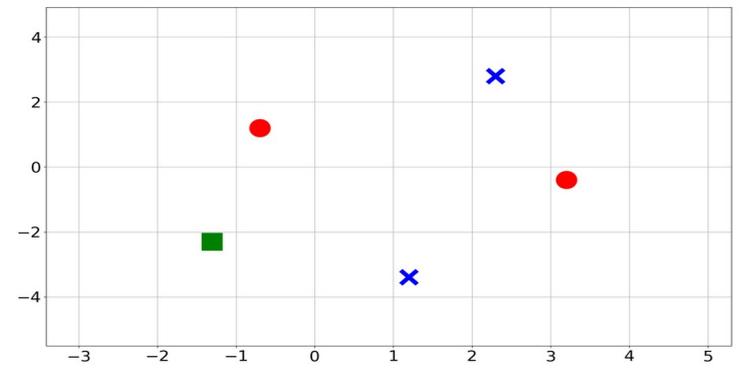
$\mathbf{x}_{\text{train},1} = [1.2, -3.4]$	$\mathbf{x}_{\text{train},2} = [2.3, 2.8]$	$\mathbf{x}_{\text{train},3} = [-0.7, 1.2]$	$\mathbf{x}_{\text{train},4} = [3.2, -0.4]$	$\mathbf{x}_{\text{train},5} = [-1.3, 2.3]$
$Y_{\text{train},1} = 0$	$Y_{\text{train},2} = 0$	$Y_{\text{train},3} = 1$	$Y_{\text{train},4} = 1$	$Y_{\text{train},5} = 2$

Choix de la fonction



$$f(x; \theta) = \text{MLP} \left(x; \theta = \{W^{(l)}, \mathbf{b}^{(l)}\}_l \right)$$

hyper-paramètres : L, H



Apprentissage

Choix du coût $l(y, \mathbf{s}) = -\ln(\text{softmax}(\mathbf{s})[y])$

Optimisation $\theta^* = \arg \min_{\theta} \sum_{i=1}^5 -\ln(\text{softmax}(\text{MLP}(\mathbf{x}_{\text{train},i}; \theta)) [Y_{\text{train},i}])$

Inférence

$$\mathbf{s}_{\text{test}} = \text{MLP}(x_{\text{test}}; \theta^*)$$

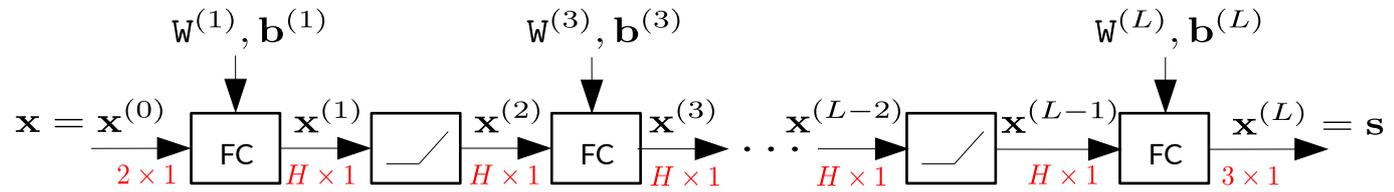
Classe prédite : $\arg \max \mathbf{s}_{\text{test}}$

IV)

Exemple de classification : 5 données, $\mathbf{X} \in \mathbb{R}^2$ et $Y \in \{0, 1, 2\}$

$\mathbf{x}_{\text{train},1} = [1.2, -3.4]$	$\mathbf{x}_{\text{train},2} = [2.3, 2.8]$	$\mathbf{x}_{\text{train},3} = [-0.7, 1.2]$	$\mathbf{x}_{\text{train},4} = [3.2, -0.4]$	$\mathbf{x}_{\text{train},5} = [-1.3, 2.3]$
$Y_{\text{train},1} = 0$	$Y_{\text{train},2} = 0$	$Y_{\text{train},3} = 1$	$Y_{\text{train},4} = 1$	$Y_{\text{train},5} = 2$

Choix de la fonction



$$f(x; \theta) = \text{MLP} \left(x; \theta = \{W^{(l)}, \mathbf{b}^{(l)}\}_l \right)$$

hyper-paramètres : L, H

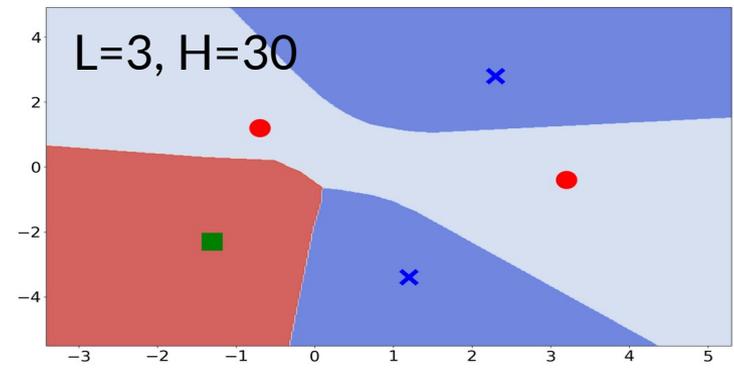
Apprentissage

Choix du coût

$$l(y, \mathbf{s}) = -\ln(\text{softmax}(\mathbf{s})[y])$$

Optimisation

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^5 -\ln(\text{softmax}(\text{MLP}(\mathbf{x}_{\text{train},i}; \theta)) [Y_{\text{train},i}])$$



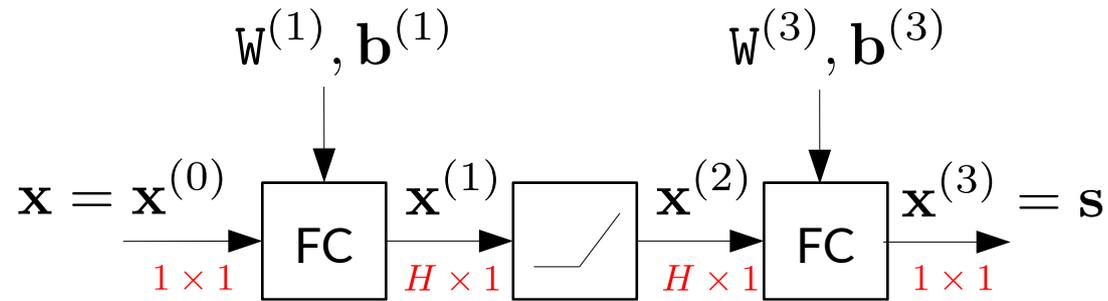
Inférence

$$\mathbf{s}_{\text{test}} = \text{MLP}(x_{\text{test}}; \theta^*)$$

Classe prédite : $\arg \max \mathbf{s}_{\text{test}}$

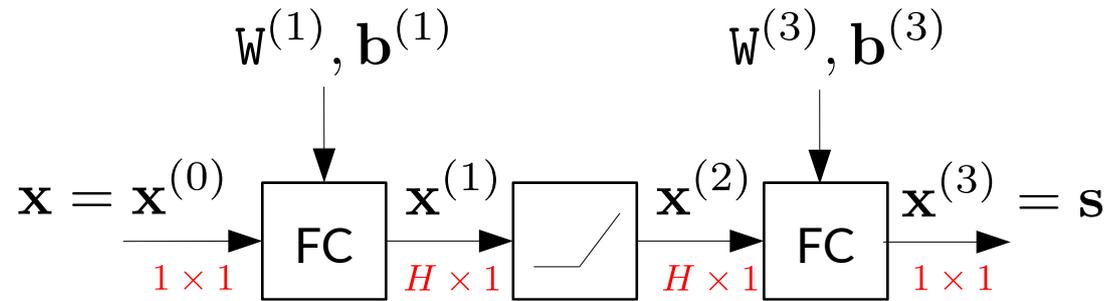
IV)

Étude du MLP à une « couche cachée » en 1D



IV)

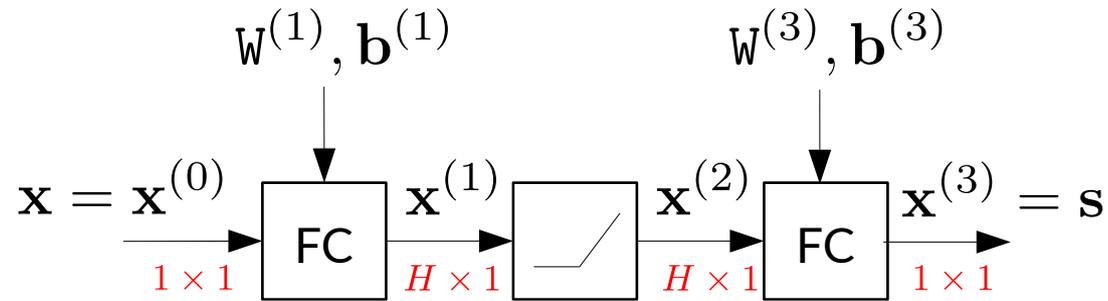
Étude du MLP à une « couche cachée » en 1D



Remarque : si on ignore la fonction ReLU, qui n'est pas paramétrique, alors il n'y a qu'une seule couche cachée $\mathbf{x}^{(1)}$.

IV)

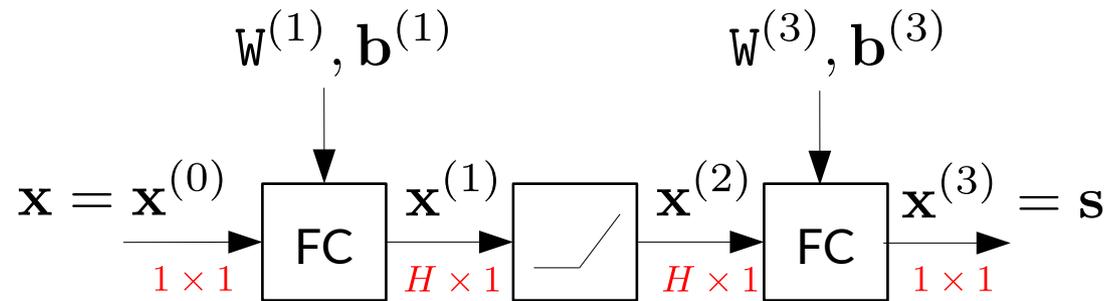
Étude du MLP à une « couche cachée » en 1D



H contrôle la largeur (« width ») du réseau

IV)

Étude du MLP à une « couche cachée » en 1D



$W^{(1)} \doteq \mathbf{w}_1$ Vecteur colonne $H \times 1$

$\mathbf{b}^{(1)} \doteq \mathbf{b}_1$ Vecteur colonne $H \times 1$

$W^{(3)} \doteq \mathbf{w}_3^\top$ Vecteur ligne $1 \times H$

$\mathbf{b}^{(3)} \doteq b_3$ Scalaire 1×1

$$f(x) = \mathbf{w}_3^\top (\text{ReLU}(\mathbf{w}_1 x + \mathbf{b}_1)) + b_3$$

IV)

Étude du MLP à une « couche cachée » en 1D (suite)

$$\begin{aligned} f(x) &= \mathbf{w}_3^\top \text{ReLU}(\mathbf{w}_1 x + \mathbf{b}_1) + b_3 \\ &= \sum_{j=1}^H \mathbf{w}_{3,j} \text{ReLU}(\mathbf{w}_{1,j} x + \mathbf{b}_{1,j}) + b_3 \end{aligned}$$



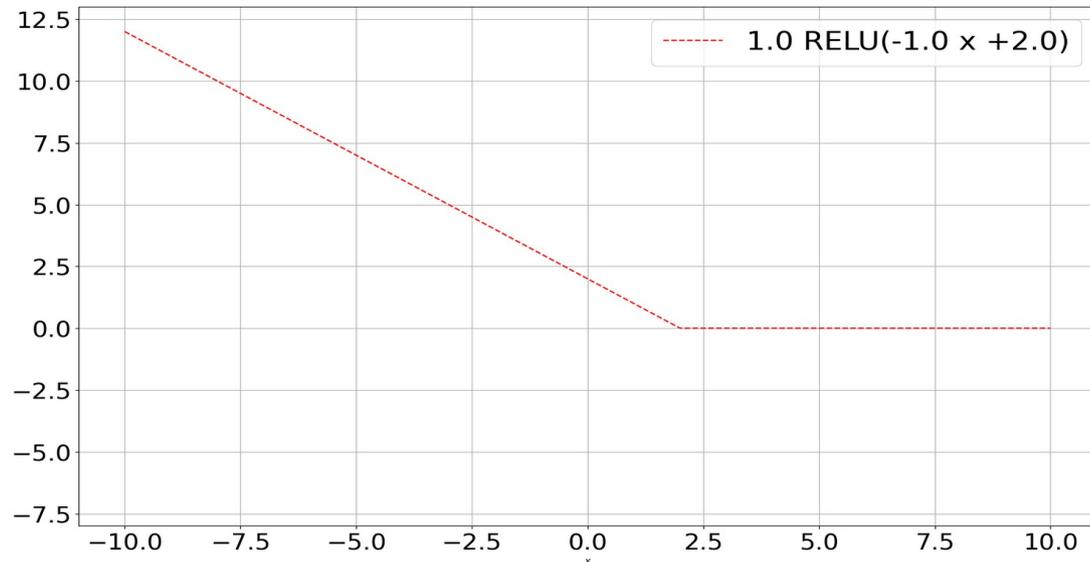
Somme pondérée de fonctions ReLU !

IV)

Étude du MLP à une « couche cachée » en 1D (suite)

$$\begin{aligned} f(x) &= \mathbf{w}_3^\top \text{ReLU}(\mathbf{w}_1 x + \mathbf{b}_1) + b_3 \\ &= \sum_{j=1}^H \mathbf{w}_{3,j} \text{ReLU}(\mathbf{w}_{1,j} x + \mathbf{b}_{1,j}) + b_3 \end{aligned}$$

Exemple
numérique avec
H=3

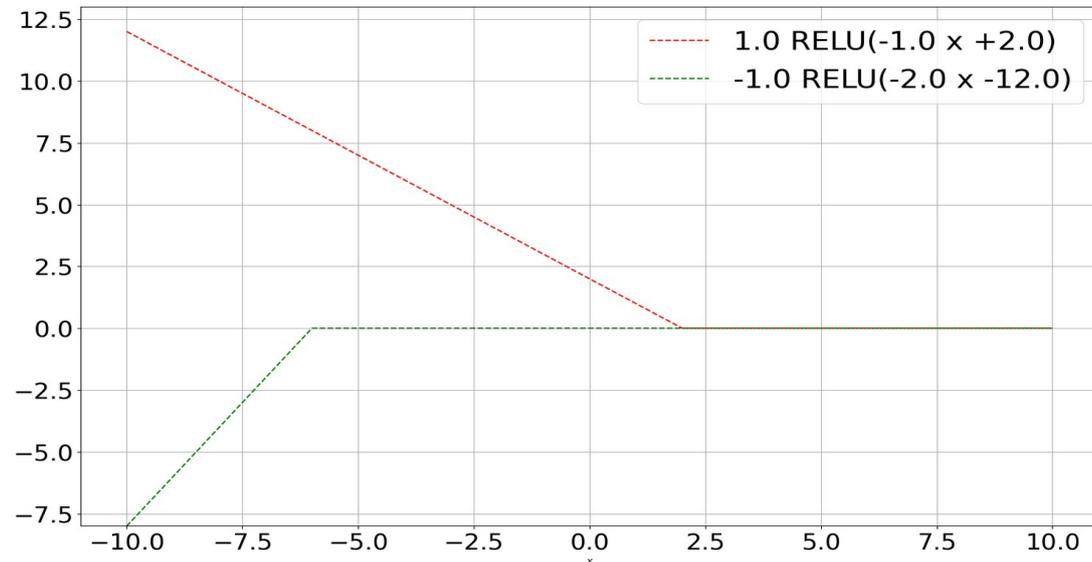


IV)

Étude du MLP à une « couche cachée » en 1D (suite)

$$\begin{aligned} f(x) &= \mathbf{w}_3^\top \text{ReLU}(\mathbf{w}_1 x + \mathbf{b}_1) + b_3 \\ &= \sum_{j=1}^H \mathbf{w}_{3,j} \text{ReLU}(\mathbf{w}_{1,j} x + \mathbf{b}_{1,j}) + b_3 \end{aligned}$$

Exemple
numérique avec
 $H=3$

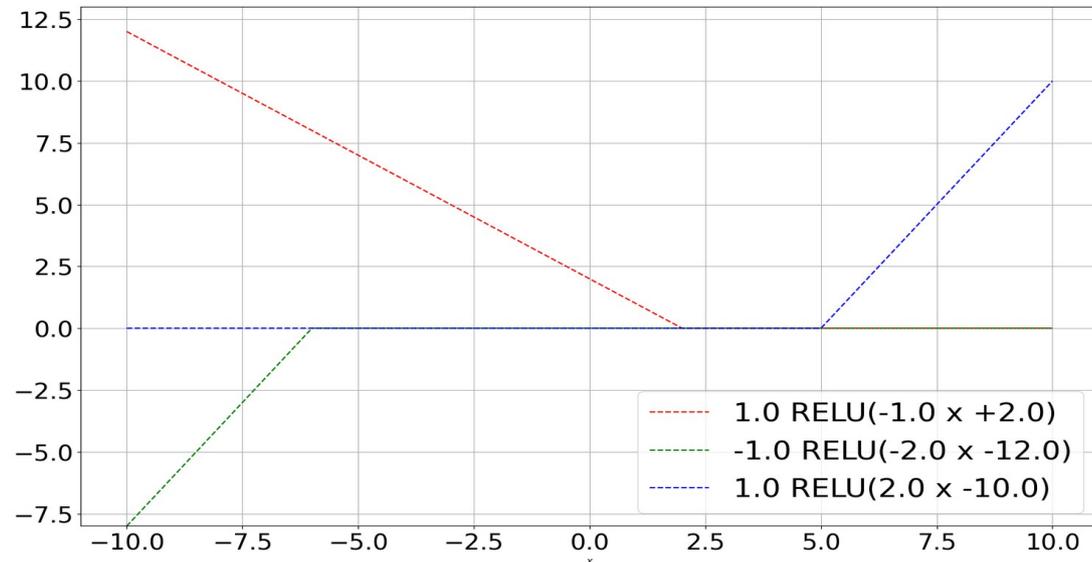


IV)

Étude du MLP à une « couche cachée » en 1D (suite)

$$\begin{aligned} f(x) &= \mathbf{w}_3^\top \text{ReLU}(\mathbf{w}_1 x + \mathbf{b}_1) + b_3 \\ &= \sum_{j=1}^H \mathbf{w}_{3,j} \text{ReLU}(\mathbf{w}_{1,j} x + \mathbf{b}_{1,j}) + b_3 \end{aligned}$$

Exemple
numérique avec
 $H=3$



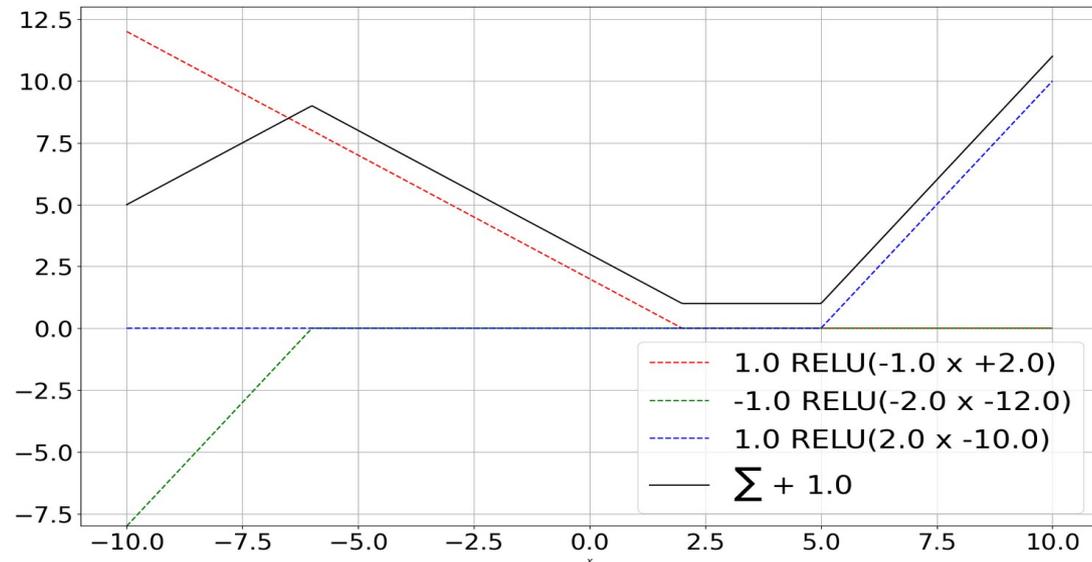
IV)

Étude du MLP à une « couche cachée » en 1D (suite)

$$f(x) = \mathbf{w}_3^\top \text{ReLU}(\mathbf{w}_1 x + \mathbf{b}_1) + b_3$$

$$= \sum_{j=1}^H \mathbf{w}_{3,j} \text{ReLU}(\mathbf{w}_{1,j} x + \mathbf{b}_{1,j}) + b_3$$

Exemple
numérique avec
H=3



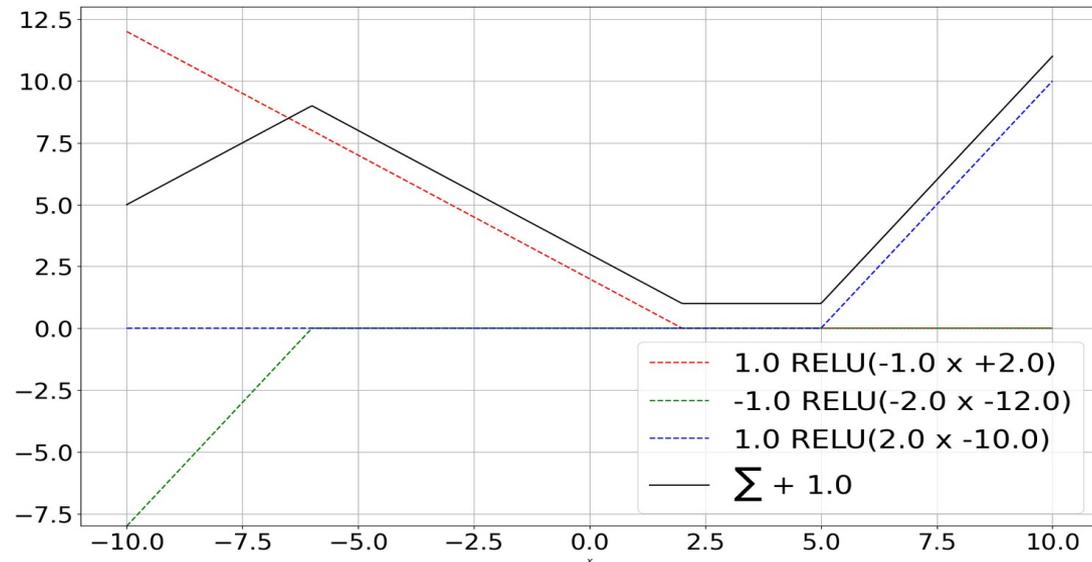
IV)

Étude du MLP à une « couche cachée » en 1D (suite)

$$f(x) = \mathbf{w}_3^\top \text{ReLU}(\mathbf{w}_1 x + \mathbf{b}_1) + b_3$$

$$= \sum_{j=1}^H \mathbf{w}_{3,j} \text{ReLU}(\mathbf{w}_{1,j} x + \mathbf{b}_{1,j}) + b_3$$

Exemple
numérique avec
H=3



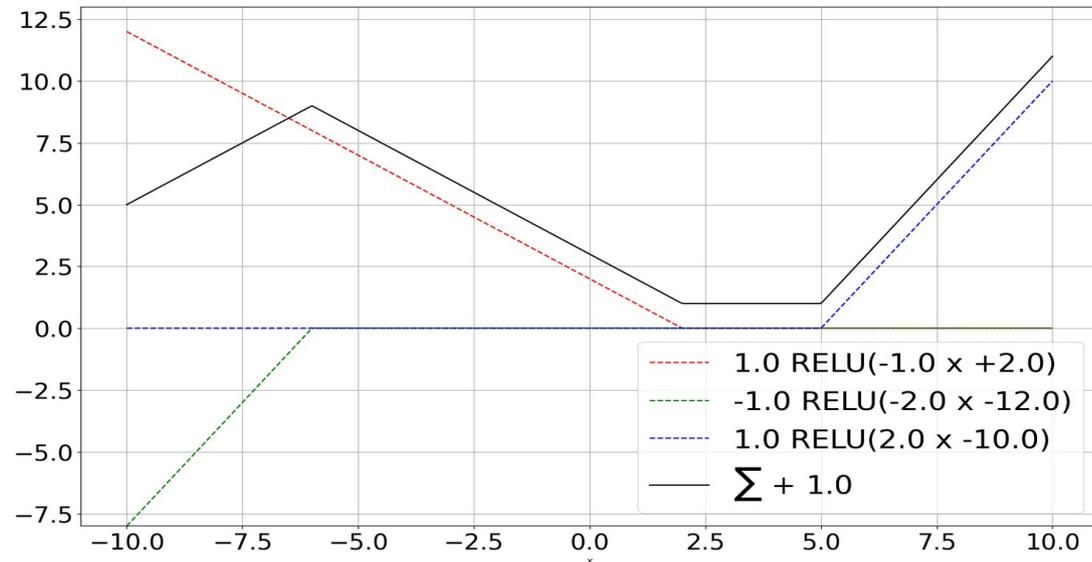
**Fonction
continue et
affine par
morceaux**

IV)

Étude du MLP à une « couche cachée » en 1D (suite)

$$\begin{aligned} f(x) &= \mathbf{w}_3^\top \text{ReLU}(\mathbf{w}_1 x + \mathbf{b}_1) + b_3 \\ &= \sum_{j=1}^H \mathbf{w}_{3,j} \text{ReLU}(\mathbf{w}_{1,j} x + \mathbf{b}_{1,j}) + b_3 \end{aligned}$$

Exemple
numérique avec
H=3



H
changements
de pente
(un par ReLU)
=
H+1
130
morceaux

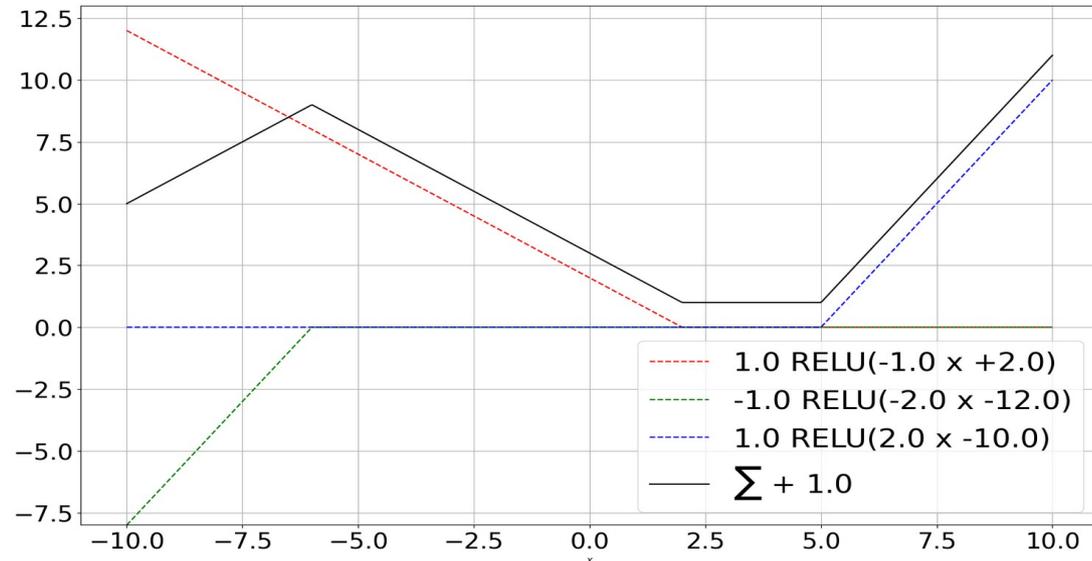
IV)

Étude du MLP à une « couche cachée » en 1D (suite)

$$f(x) = \mathbf{w}_3^\top \text{ReLU}(\mathbf{w}_1 x + \mathbf{b}_1) + b_3$$

$$= \sum_{j=1}^H \mathbf{w}_{3,j} \text{ReLU}(\mathbf{w}_{1,j} x + \mathbf{b}_{1,j}) + b_3$$

Exemple
numérique avec
H=3



**Augmenter H
accroît la
capacité du
réseau.**

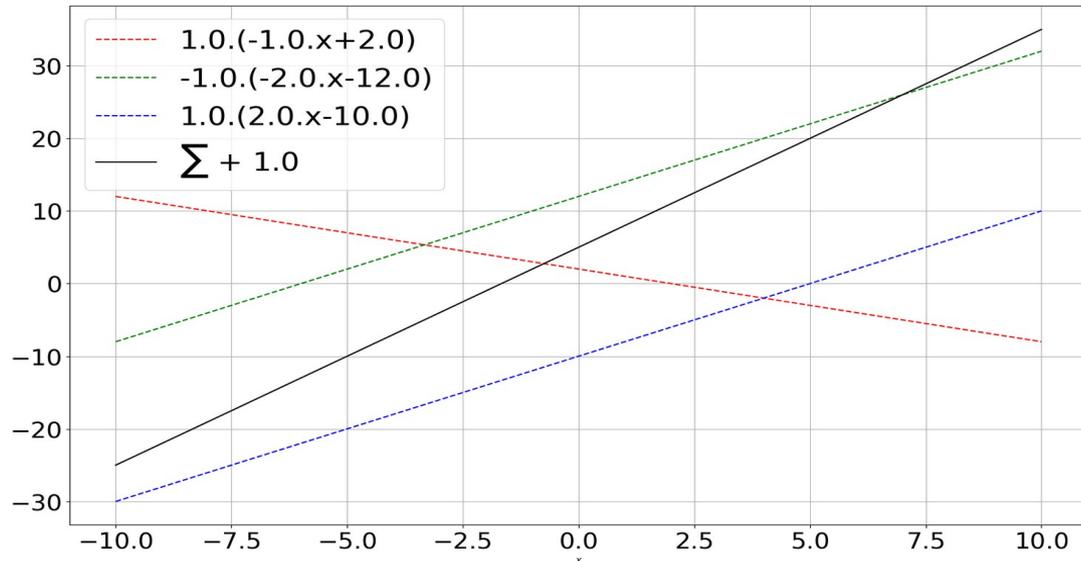
IV)

Étude du MLP à une « couche cachée » en 1D (suite)

$$f(x) = \mathbf{w}_3^\top \text{ReLU}(\mathbf{w}_1 x + \mathbf{b}_1) + b_3$$

$$= \sum_{j=1}^H \mathbf{w}_{3,j} \text{ReLU}(\mathbf{w}_{1,j} x + \mathbf{b}_{1,j}) + b_3$$

Et sans ReLU ?



Une somme de fonctions linéaires est une fonction linéaire...

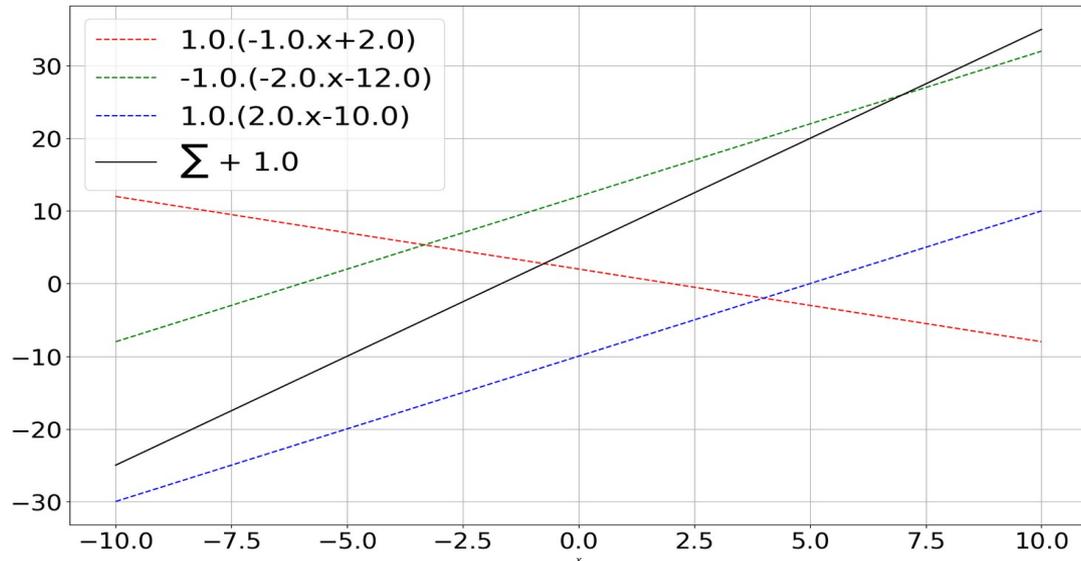
IV)

Étude du MLP à une « couche cachée » en 1D (suite)

$$f(x) = \mathbf{w}_3^\top \text{ReLU}(\mathbf{w}_1 x + \mathbf{b}_1) + b_3$$

$$= \sum_{j=1}^H \mathbf{w}_{3,j} \text{ReLU}(\mathbf{w}_{1,j} x + \mathbf{b}_{1,j}) + b_3$$

Et sans ReLU ?

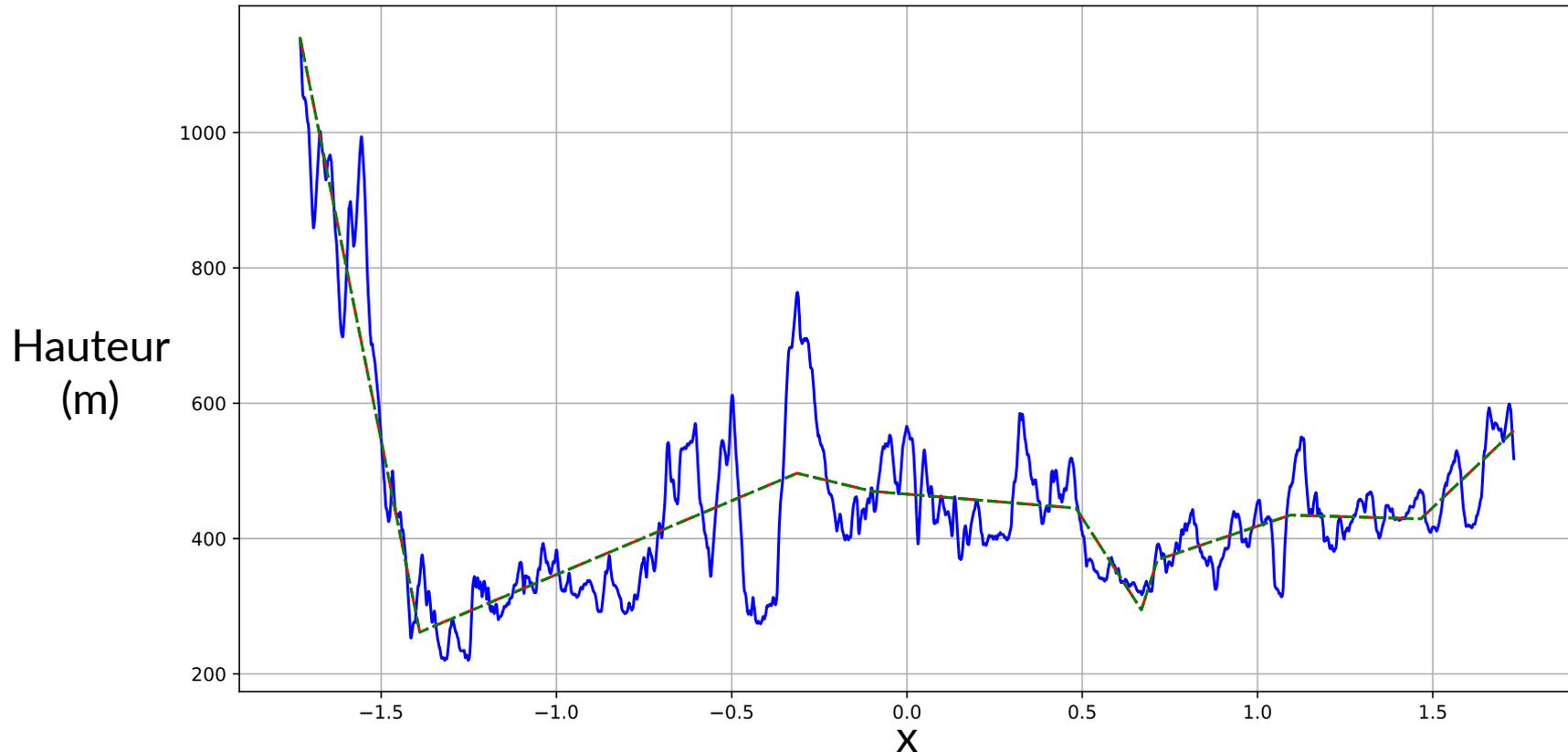


Augmenter H
n'accroît pas
la capacité du
réseau.

IV)

Étude du MLP à une « couche cachée » en 1D (suite bis)

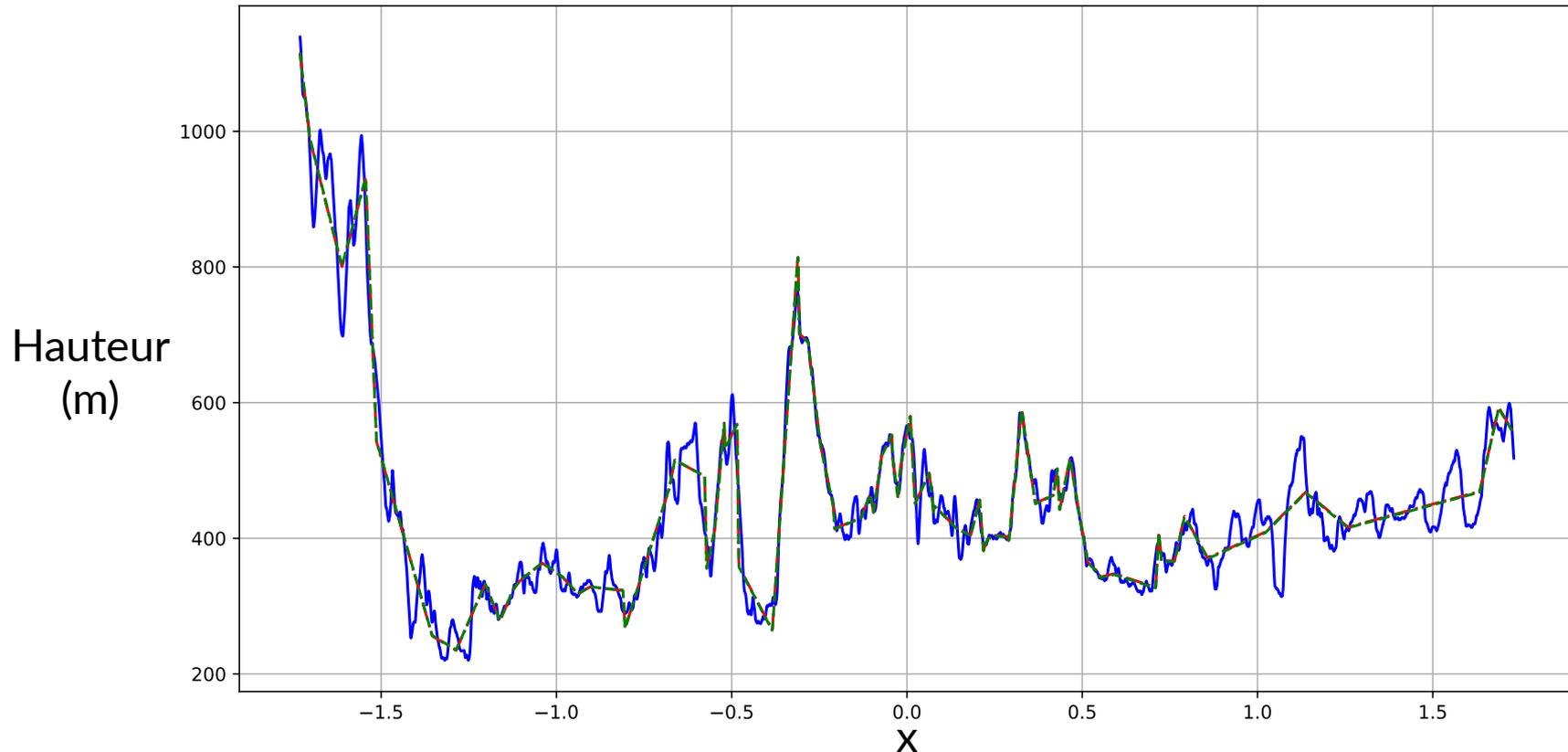
Optimisation d'un MLP sur un profil de terrain : $H=10$



IV)

Étude du MLP à une « couche cachée » en 1D (suite bis)

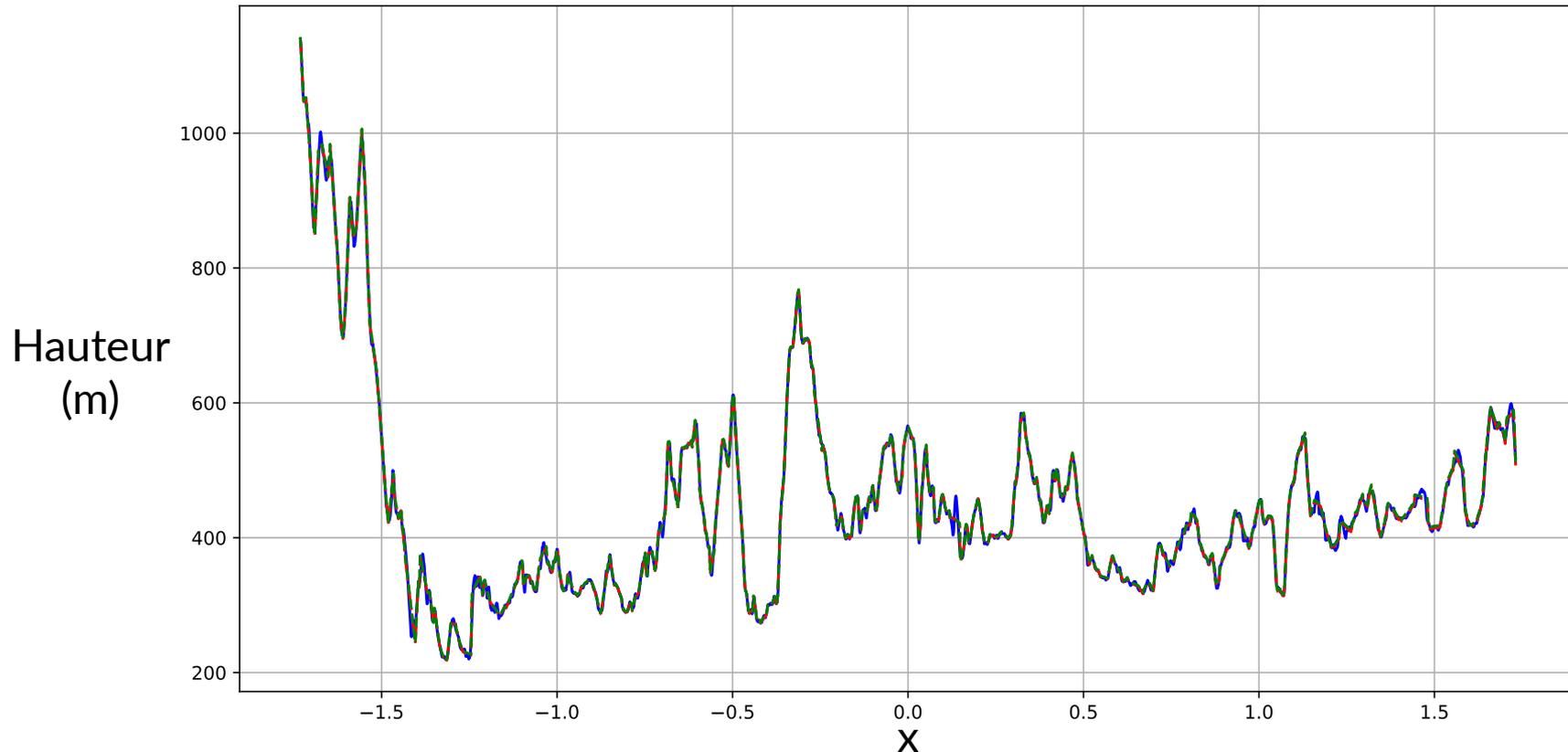
Optimisation d'un MLP sur un profil de terrain : $H=100$



IV)

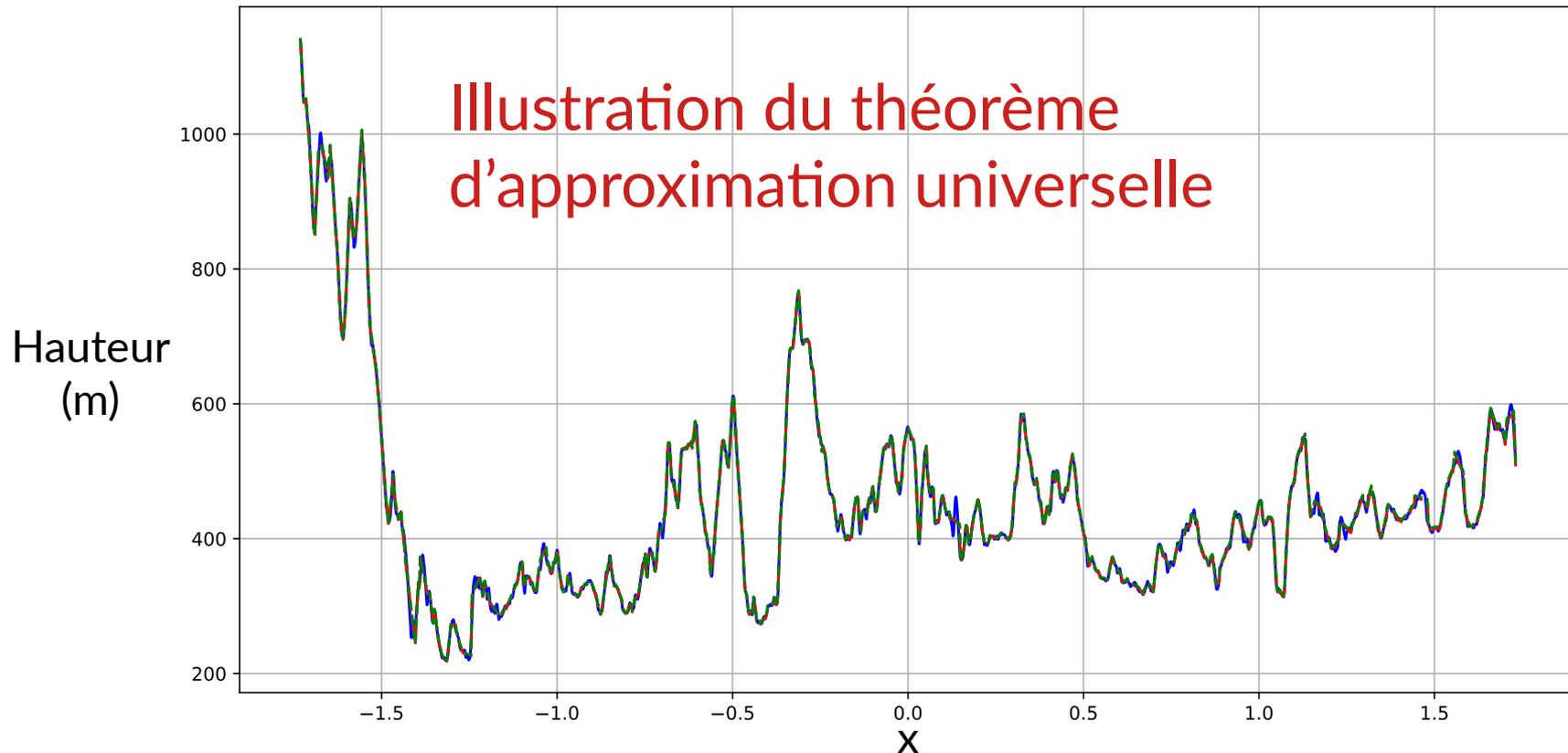
Étude du MLP à une « couche cachée » en 1D (suite bis)

Optimisation d'un MLP sur un profil de terrain : $H=1000$



Étude du MLP à une « couche cachée » en 1D (suite bis)

Optimisation d'un MLP sur un profil de terrain : $H=1000$



V) Risques

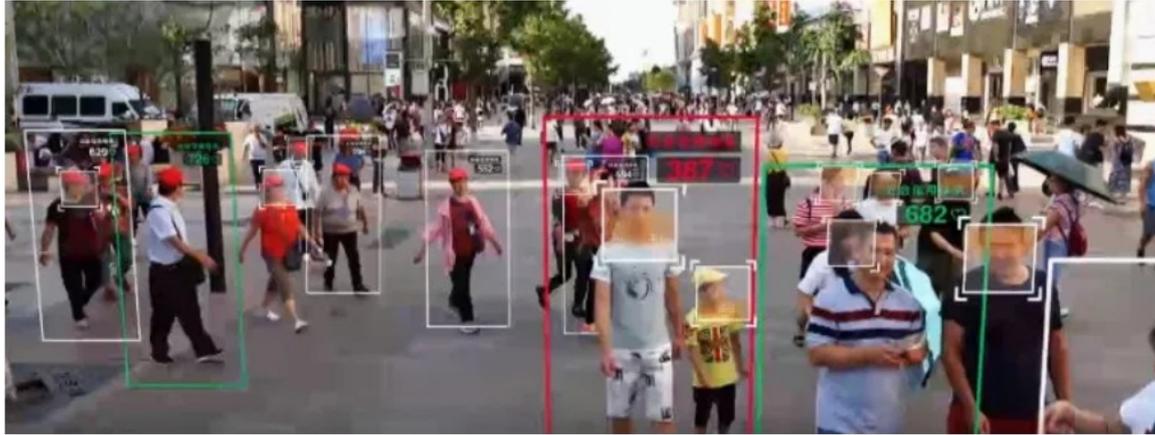
IA et vie privée

Are you ready? Here is all the data
Facebook and Google have on you
Dylan Curran

- Google knows where you've been
- Google knows everything you've ever searched – and deleted
- Google has an advertisement profile of you
- Google knows all the apps you use
- Google has all of your YouTube history
- The data Google has on you can fill millions of Word documents

<https://www.theguardian.com/commentisfree/2018/mar/28/all-the-data-facebook-google-has-on-you-privacy>

IA et vie privée (suite)



The Chinese state wants to control its citizens via a system of social scoring that punishes behavior it doesn't approve of. Image Credit: Telecoms

Le projet de smart city d'Alphabet à Toronto suscite l'inquiétude des experts

VU AILLEURS Le projet de smart city lancé à Toronto en 2017 par Sidewalk Labs, la branche d'innovation urbaine d'Alphabet, est vivement critiqué par la population locale et certains experts consultants du projet. La protection de la vie privée ne serait pas respectée.

v)

La machine au service de l'homme ?



Mme Tang Yu, PDG du chinois NetDragon Websoft et de ses 6000 employés, est le premier robot à être nommé à la tête d'une société. Disponible H24, elle ne touche aucun salaire. *NetDragon Websoft*

v)

IA et consommation énergétique

Common carbon footprint benchmarks

in lbs of CO2 equivalent

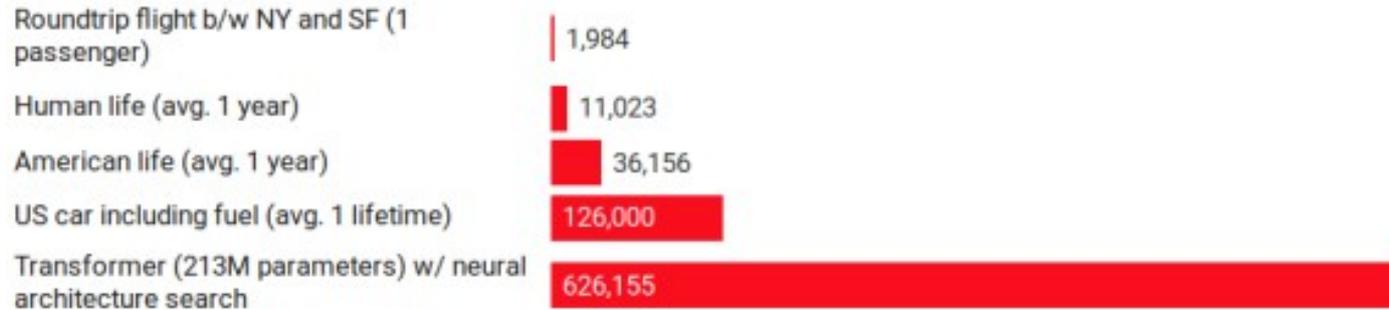


Chart: MIT Technology Review • Source: Strubell et al. • Created with Datawrapper

Training a single AI model can emit as much carbon as five cars in their lifetimes, MIT Press

Energy and Policy Considerations for Deep Learning in NLP, Strubell et al., 2019

v)

IA et transhumanisme

