



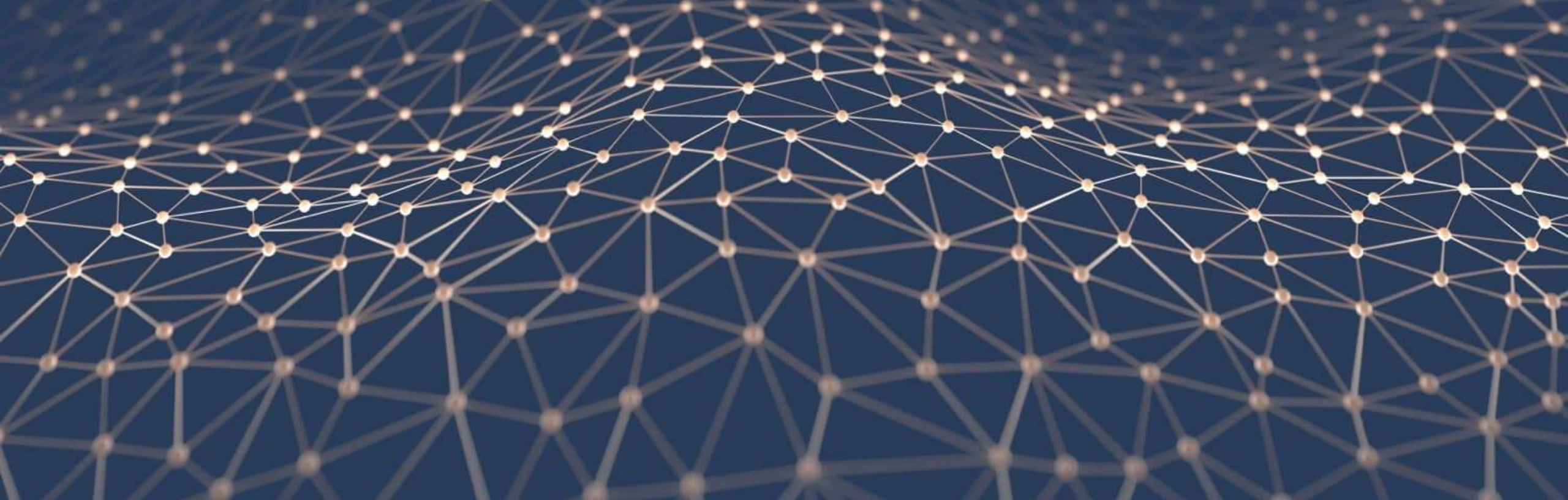
INTRODUCTION DATA INTELLIGENCE & DATA SCIENCE

MODULE 4 – INTRODUCTION AU DEEP LEARNING

Informatique – orientation IA – 1DA/IA

PLAN

- Introduction & définitions
- Le perceptron
- Application de l'algorithme
- Réseau multicouche
- Domaines d'application



INTRODUCTION & DÉFINITIONS



INTRODUCTION & DÉFINITIONS

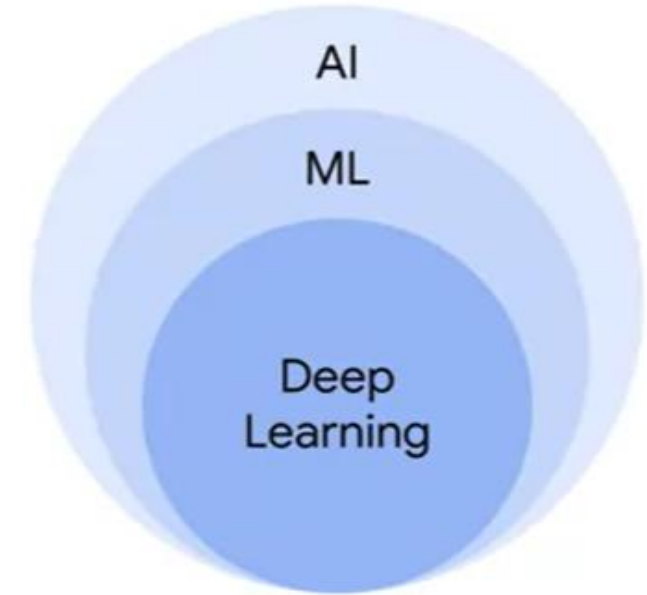
DEEP LEARNING

« Le deep learning est un sous-ensemble du machine learning (ML), où les réseaux neuronaux artificiels - des algorithmes conçus pour fonctionner comme le cerveau humain - apprennent à partir de grandes quantités de données. »

Oracle.com

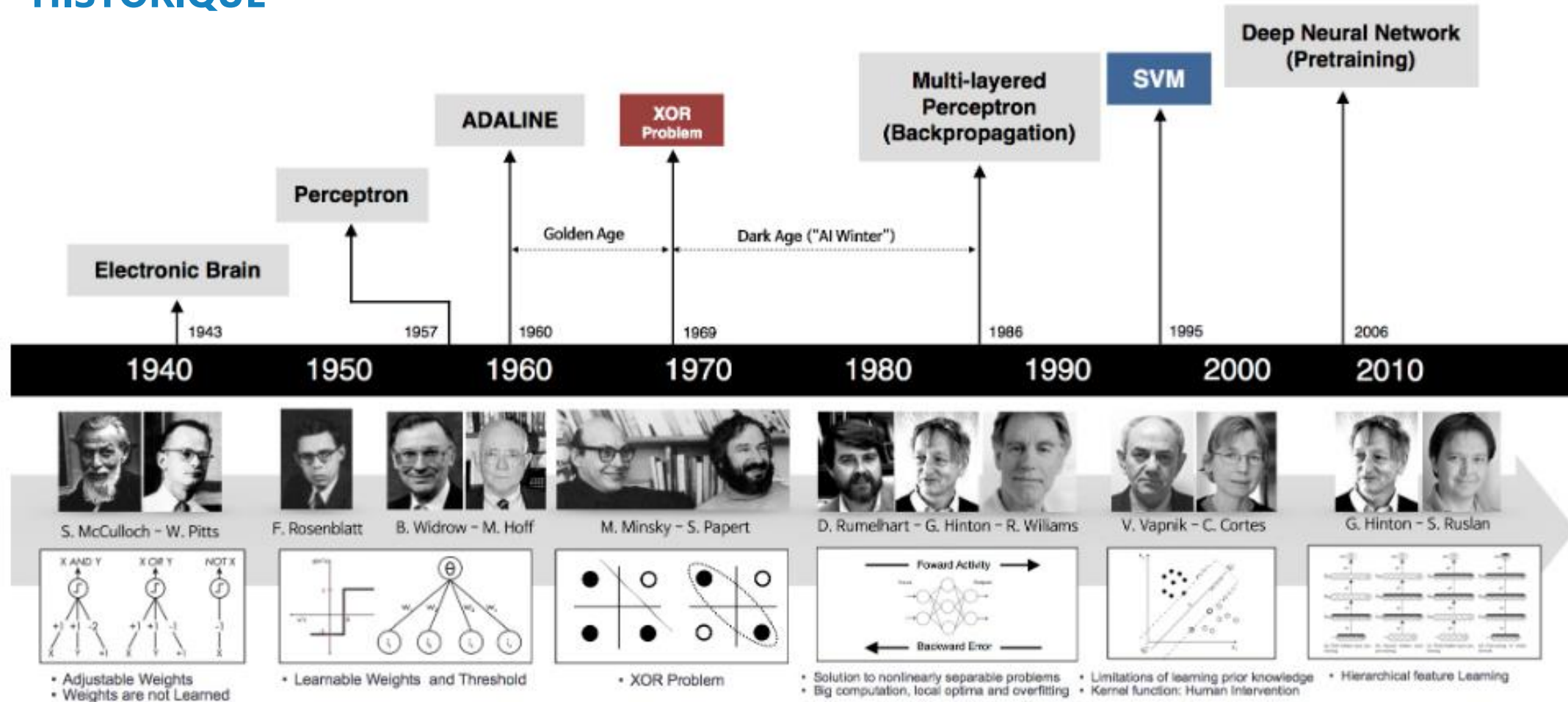
« L'apprentissage profond est un procédé d'apprentissage automatique utilisant des réseaux de neurones possédants plusieurs couches de neurones cachées. Ces algorithmes possédant de très nombreux paramètres, ils demandent un nombre très important de données afin d'être entraînés.»

Cnil.fr



INTRODUCTION & DÉFINITIONS

HISTORIQUE



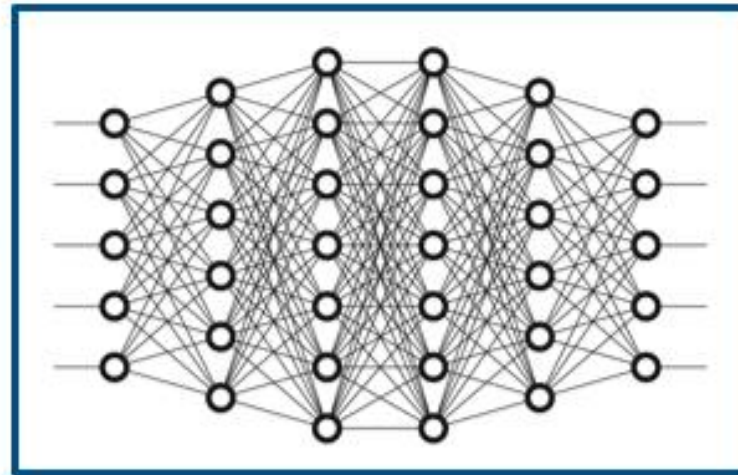
INTRODUCTION & DÉFINITIONS

PRINCIPE DE FONCTIONNEMENT D'UN RÉSEAU DE NEURONES

Phase d'entraînement

| | x1 | x2 | x3 | x4 | x5 | ... | y |
|----|----|----|----|----|----|-----|---|
| 1 | 4 | 9 | 2 | 16 | 20 | ... | 1 |
| 2 | 9 | 10 | 3 | 17 | 20 | ... | 1 |
| 3 | 19 | 10 | 8 | 9 | 24 | ... | 0 |
| 4 | 17 | 12 | 24 | 20 | 16 | ... | 0 |
| 5 | 4 | 1 | 2 | 11 | 11 | ... | 1 |
| 6 | 2 | 2 | 3 | 11 | 2 | ... | 1 |
| 7 | 10 | 15 | 16 | 16 | 8 | ... | 1 |
| 8 | 7 | 15 | 25 | 4 | 13 | ... | 1 |
| 9 | 11 | 15 | 5 | 12 | 1 | ... | 0 |
| 10 | 1 | 1 | 8 | 9 | 14 | ... | 1 |
| 11 | 2 | 12 | 6 | 1 | 9 | ... | 1 |
| 12 | 11 | 20 | 16 | 6 | 11 | ... | 1 |
| 13 | 6 | 16 | 16 | 16 | 16 | ... | 1 |
| 14 | 16 | 16 | 6 | 9 | 12 | ... | 1 |
| 15 | 14 | 14 | 12 | 2 | 11 | ... | 1 |
| 16 | 17 | 6 | 6 | 19 | 11 | ... | 0 |
| 17 | 16 | 19 | 3 | 5 | 6 | ... | 0 |
| 18 | 4 | 1 | 11 | 11 | 4 | ... | 1 |
| 19 | 6 | 1 | 10 | 11 | 16 | ... | 1 |
| 20 | 16 | 16 | 17 | 10 | 16 | ... | 1 |
| 21 | 1 | 11 | 1 | 11 | 8 | ... | 1 |
| 22 | 1 | 15 | 6 | 11 | 14 | ... | 1 |

Train set



Réseau de neurones



| |
|-----|
| w1 |
| w2 |
| w3 |
| w4 |
| w5 |
| ... |

Valeurs des poids



Apprentissage supervisé

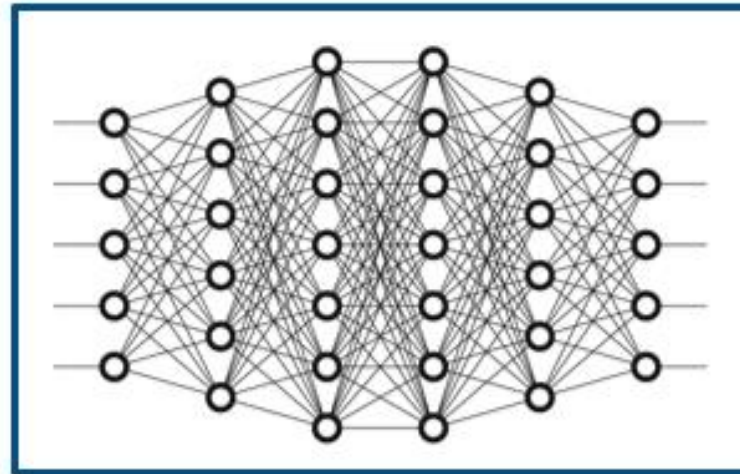
INTRODUCTION & DÉFINITIONS

PRINCIPE DE FONCTIONNEMENT D'UN RÉSEAU DE NEURONES

Phase d'utilisation

| |
|-----|
| x1 |
| x2 |
| x3 |
| x4 |
| x5 |
| ... |

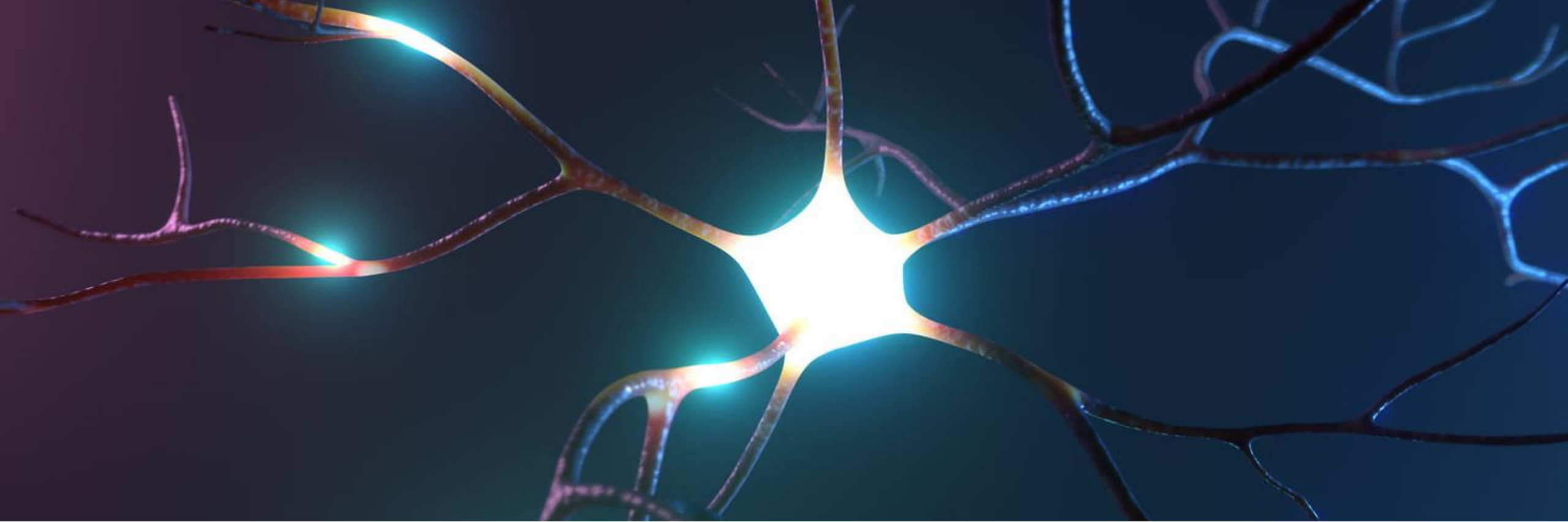
**Valeurs
d'entrée**



Réseau de neurones



Sortie estimée

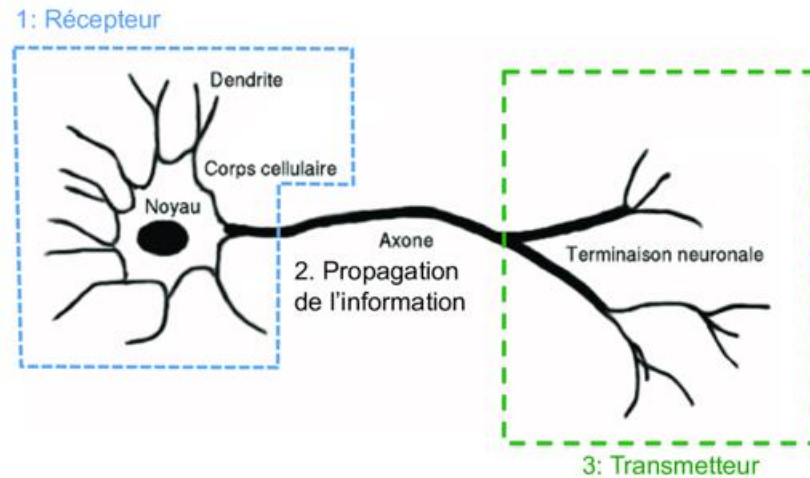


LE PERCEPTRON

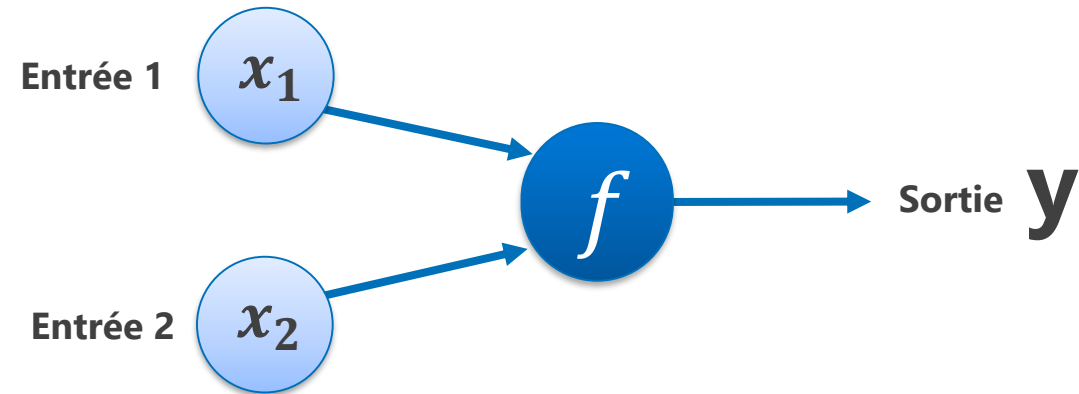
LE PERCEPTRON

INTRODUCTION

Le neurone formel s'inspire directement du neurone biologique.



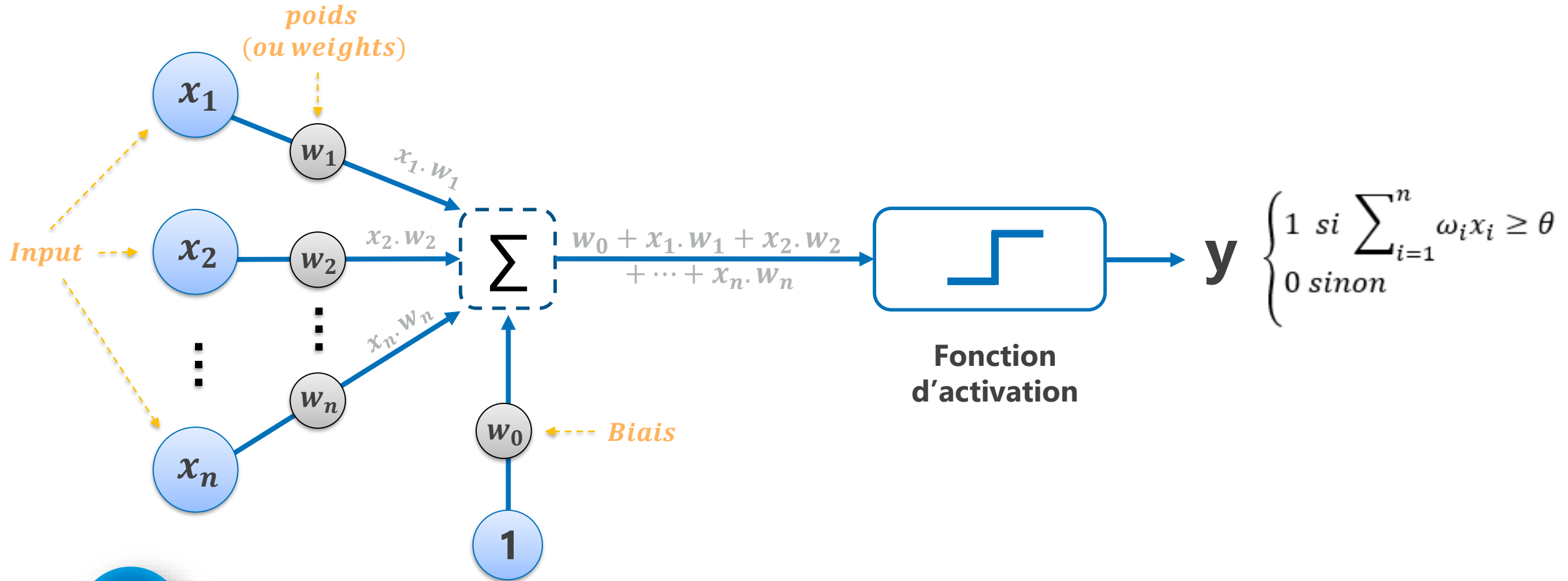
Neurone biologique



Représentation simplifiée

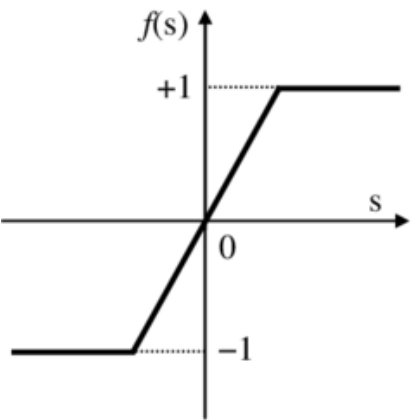
LE PERCEPTRON SIMPLE COUCHE

REPRÉSENTATION

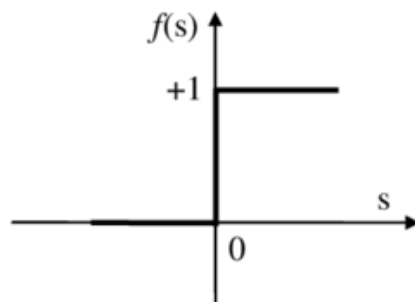


FONCTIONS D'ACTIVATION

FONCTIONS D'ACTIVATION LES PLUS COURANTES

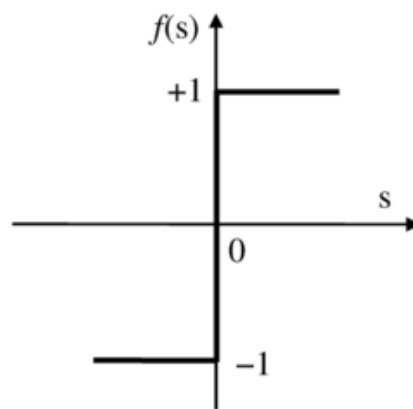


Fonction bipolaire
linéaire

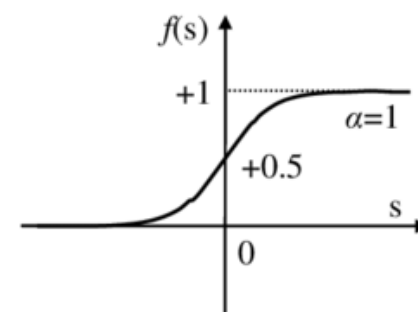


Fonction de seuil
(Heaviside)

$$\begin{cases} 1 & \text{si } f(s) \geq 0 \\ 0 & \text{sinon} \end{cases}$$

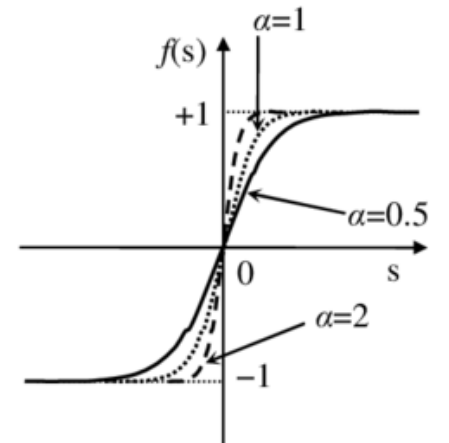


Fonction de seuil
bipolaire



Fonction sigmoïde

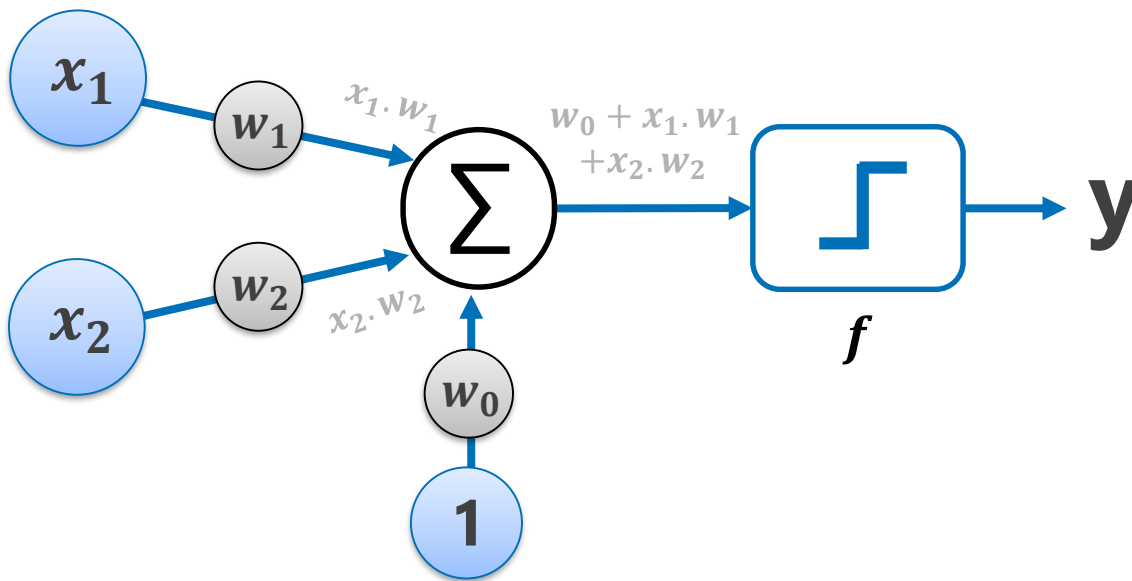
$$f(s) = \frac{1}{1 + e^{-as}}$$



Fonction tangente
hyperbolique

LE PERCEPTRON SIMPLE COUCHE

ETUDE DE CAS : 2 ENTRÉES



- **Sortie** : $y = f(w_0 + x_1 \cdot w_1 + x_2 \cdot w_2)$
- **Fonction d'activation** : fonction de seuil
- **N données d'entrée** : X_i , y_i avec i allant de 0 à n

X_1 : $x_{1,1}$, $x_{1,2}$ $\rightarrow y_1$

X_2 : $x_{2,1}$, $x_{2,2}$ $\rightarrow y_2$

X_3 : $x_{3,1}$, $x_{3,2}$ $\rightarrow y_3$

X_4 : $x_{4,1}$, $x_{4,2}$ $\rightarrow y_4$

...

X_n : $x_{n,1}$, $x_{n,2}$ $\rightarrow y_n$

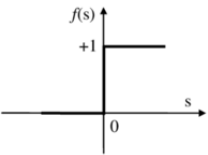
- **Inconnues** : w_0 , w_1 , w_2

LE PERCEPTRON SIMPLE COUCHE

FONCTION LOSS

Pour une donnée $(x_{1,1}, x_{1,2}, y_1)$ et des poids w_0, w_1, w_2 :

- Fonction d'activation f : **fonction de seuil**



- Valeur estimée de sortie : $y_{1\text{ estimé}} = f(w_0 + x_{1,1} \cdot w_1 + x_{1,2} \cdot w_2)$
- Valeur souhaitée de sortie : $y_{1\text{ réel}}$
- Calcul de l'erreur :

$$(y_{1\text{ estimé}} - y_{1\text{ réel}})(w_0 + x_{1,1} \cdot w_1 + x_{1,2} \cdot w_2)$$

Fonction LOSS
ou « **Cost Function** »

| $y_{1\text{ estimé}}$ | $y_{1\text{ réel}}$ | Erreur |
|-----------------------|---------------------|--|
| 1 | 1 | 0 |
| 1 | 0 | $w_0 + x_{1,1} \cdot w_1 + x_{1,2} \cdot w_2$ |
| 0 | 0 | 0 |
| 0 | 1 | $-(w_0 + x_{1,1} \cdot w_1 + x_{1,2} \cdot w_2)$ |

LE PERCEPTRON SIMPLE COUCHE

ALGORITHME - PRINCIPE

À partir de n données X_i, y_i : (avec $X_i = (x_{i,1}, x_{i,2})$)

- Déterminer les valeurs des paramètres w_0, w_1, w_2 de telle façon que :

- L'erreur soit minimum

→ Minimisation de la fonction LOSS

→ Calcul du gradient $\nabla : \left(\frac{\partial \text{LOSS}}{\partial w_0}, \frac{\partial \text{LOSS}}{\partial w_1}, \frac{\partial \text{LOSS}}{\partial w_2} \right)$

- $$\frac{\partial \text{LOSS}}{\partial w_1} = \frac{\partial ((y_{i \text{ estimé}} - y_{i \text{ réel}})(w_0 + x_{i,1} \cdot w_1 + x_{i,2} \cdot w_2))}{\partial w_1} = (y_{i \text{ estimé}} - y_{i \text{ réel}})x_{i,1} \quad (\text{En jaune : constantes})$$



En généralisant à w_j , on obtient :
$$\frac{\partial \text{LOSS}}{\partial w_j} = (y_{i \text{ estimé}} - y_{i \text{ réel}})x_{i,j}$$

LE PERCEPTRON SIMPLE COUCHE

ALGORITHME - PRINCIPE

Mise à jour des poids w_i :

$$w_i^{k+1} = w_i^k - \alpha(y_{i \text{ estimé}} - y_{i \text{ réel}})x_i$$

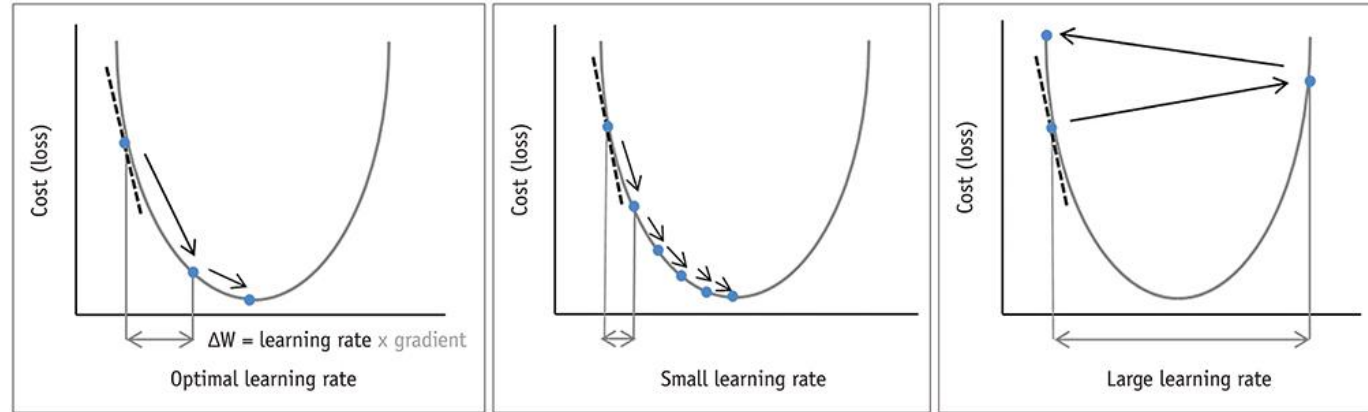
Le pas « α » (ou « learning rate ») :

- Définit la taille des ajustements ;
- Ne doit être ni trop grand, ni trop petit ;
- Plusieurs variantes possibles :

$$\alpha = \text{constante}$$

$$\alpha_k = \frac{\alpha_{k-1}}{1 + k}$$

... (Cnfr méthode du moment
ou l'accélération de Nesterov)



LE PERCEPTRON SIMPLE COUCHE

ALGORITHME - ETAPES

0. Données disponibles (data set) : $x_{i,1}, x_{i,2}, y_i$

1. Initialisation :

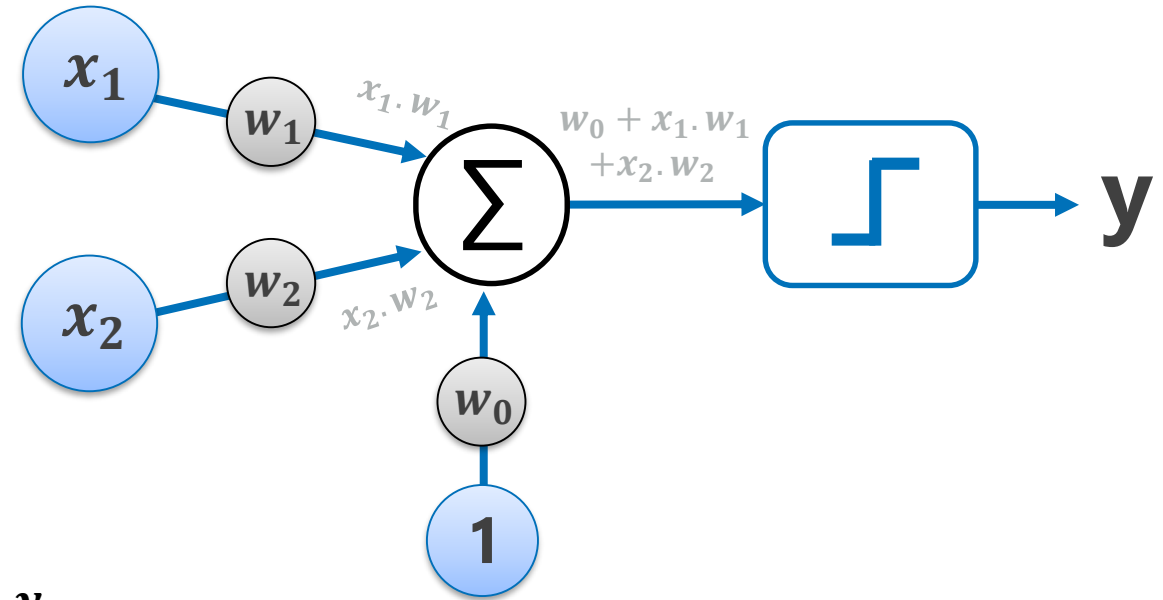
- Choix arbitraire de valeurs de w_0^1, w_1^1, w_2^1
- $k = 1$ (indice d'itération)

2. Sélection d'une nouvelle donnée disponible : $x_{k,1}, x_{k,2}, y_{k \text{ réel}}$

3. Calcul de $y_{k \text{ estimé}} = f(w_0^k + x_{k,1} \cdot w_1^k + x_{k,2} \cdot w_2^k)$

4. Mise à jour des poids (avec $\alpha = 0,1$) : $\longrightarrow \begin{cases} w_0^{k+1} = w_0^k - 0,1(y_{k \text{ estimé}} - y_{k \text{ réel}}) \cdot 1 \\ w_1^{k+1} = w_1^k - 0,1(y_{k \text{ estimé}} - y_{k \text{ réel}})x_{k,1} \\ w_2^{k+1} = w_2^k - 0,1(y_{k \text{ estimé}} - y_{k \text{ réel}})x_{k,2} \end{cases}$

5. Retour à l'étape 2



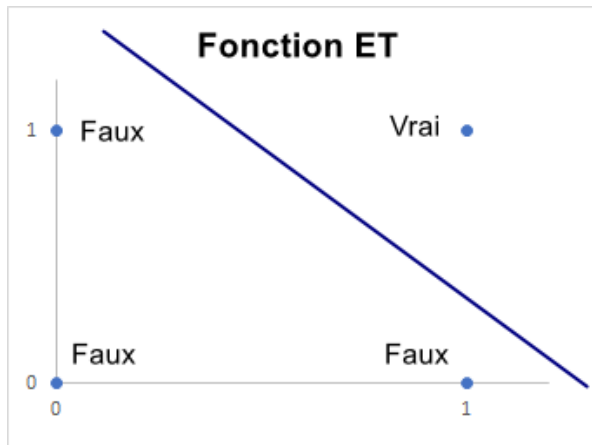
Arrêt lorsque les paramètres ne changent plus sur tout le set de données

LE PERCEPTRON SIMPLE COUCHE

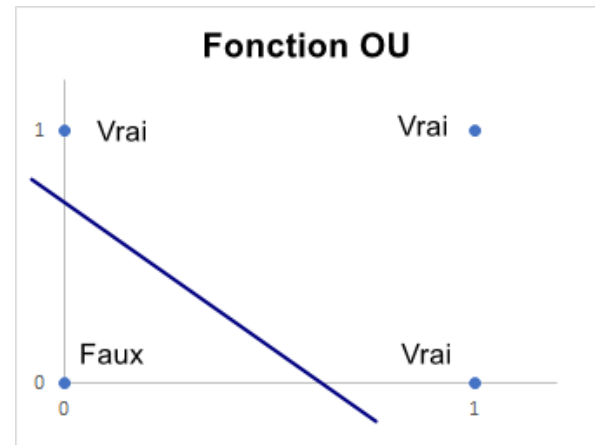
CONDITION D'UTILISATION

- Les données doivent être linéairement séparables.

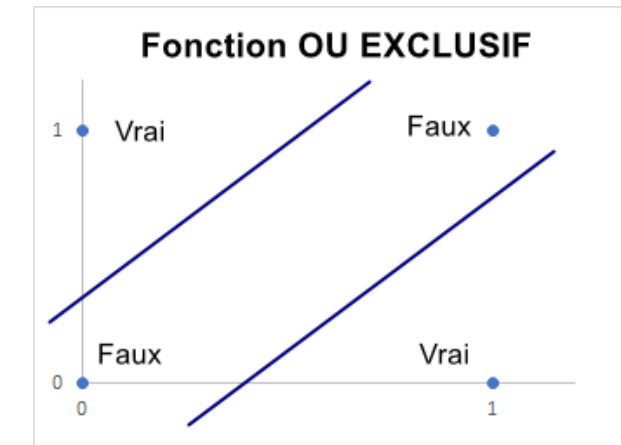
| ET | | |
|----|---|--------|
| A | B | A ET B |
| 0 | 0 | FAUX |
| 0 | 1 | FAUX |
| 1 | 0 | FAUX |
| 1 | 1 | VRAI |

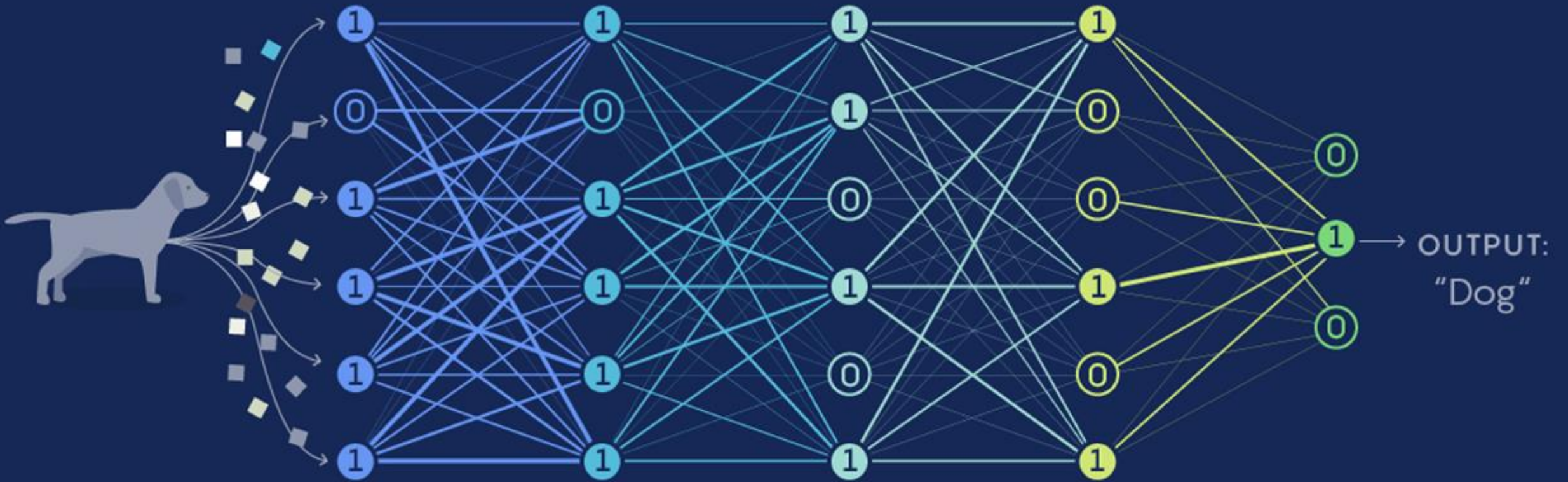


| OU | | |
|----|---|--------|
| A | B | A OU B |
| 0 | 0 | FAUX |
| 0 | 1 | VRAI |
| 1 | 0 | VRAI |
| 1 | 1 | VRAI |



| OU EXCLUSIF (XOR) | | |
|-------------------|---|---------|
| A | B | A XOR B |
| 0 | 0 | FAUX |
| 0 | 1 | VRAI |
| 1 | 0 | VRAI |
| 1 | 1 | FAUX |





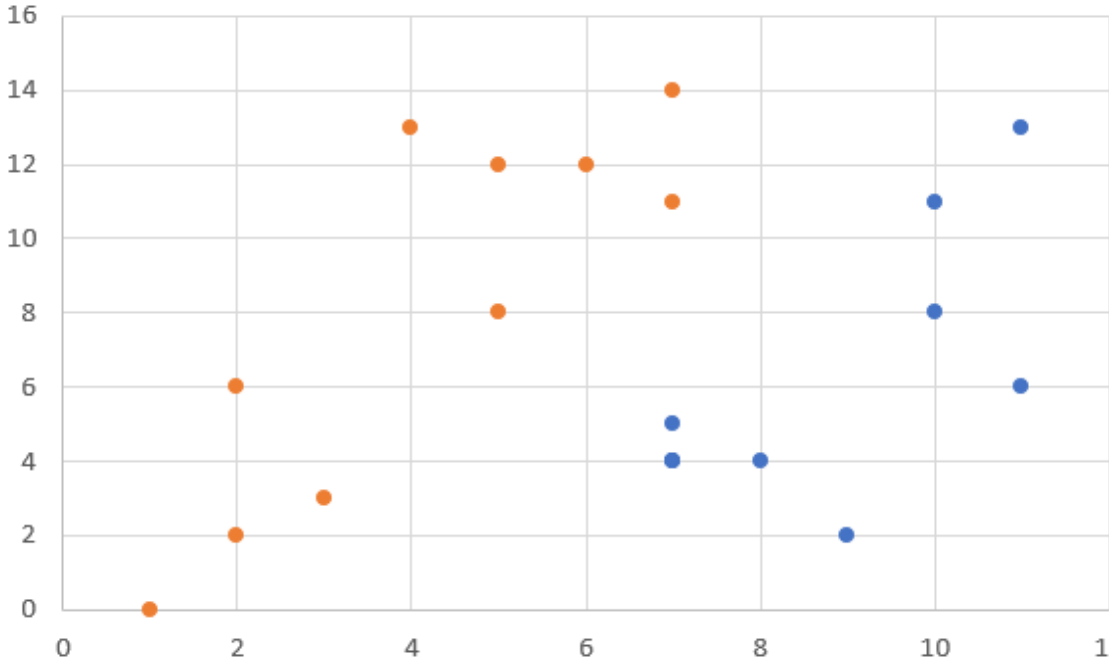
EXEMPLE D'APPLICATION

EXEMPLE D'APPLICATION

DONNÉES DE POLLUTION

- Les données représentent les coordonnées de sites pollués et non pollués

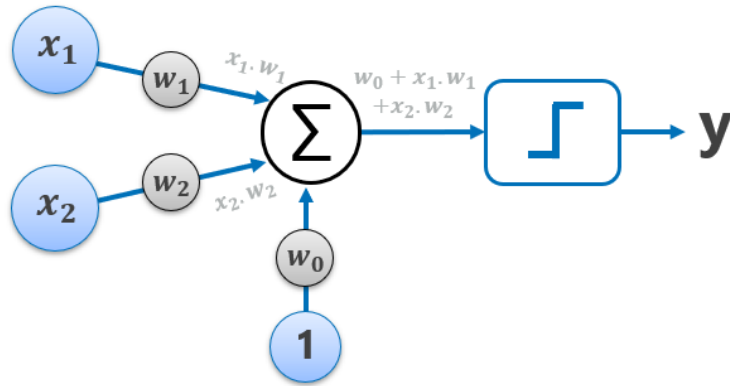
| Sites n° | Latitude | Longitude | Pollution |
|----------|----------|-----------|-----------|
| 1 | 2 | 6 | 1 |
| 2 | 7 | 11 | 1 |
| 3 | 7 | 4 | 0 |
| 4 | 10 | 8 | 0 |
| 5 | 1 | 0 | 1 |
| 6 | 7 | 5 | 0 |
| 7 | 6 | 12 | 1 |
| 8 | 10 | 11 | 0 |
| 9 | 11 | 6 | 0 |
| 10 | 4 | 13 | 1 |
| 11 | 7 | 4 | 0 |
| 12 | 3 | 3 | 1 |
| 13 | 9 | 2 | 0 |
| 14 | 5 | 8 | 1 |
| 15 | 7 | 14 | 1 |
| 16 | 5 | 12 | 1 |
| 17 | 7 | 4 | 0 |
| 18 | 11 | 13 | 0 |
| 19 | 8 | 4 | 0 |
| 20 | 2 | 2 | 1 |



EXEMPLE D'APPLICATION

PROBLÉMATIQUE

- But : Déterminer les valeurs des paramètres w_0 , w_1 , w_2 d'un perceptron simple couche



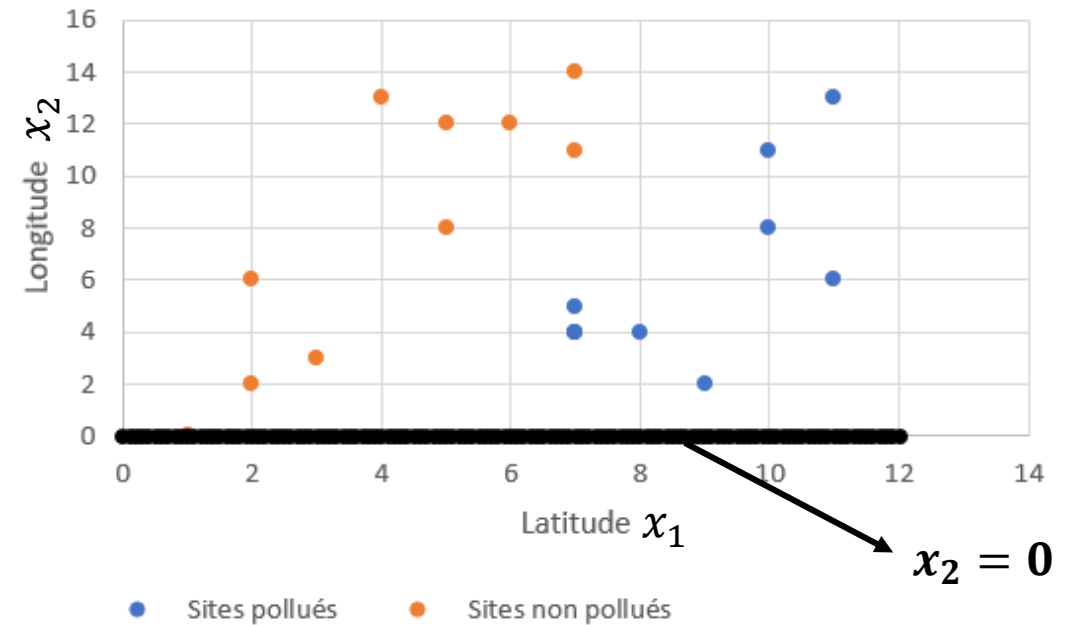
- Etape 1 : Initialisation des valeurs des poids et de α :

$$w_0^1 = 0$$

$$w_1^1 = 0$$

$$w_2^1 = 1$$

$$\alpha = 0,5$$



$$w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 = 0 + 0 \cdot x_1 + 1 \cdot x_2 = x_2$$

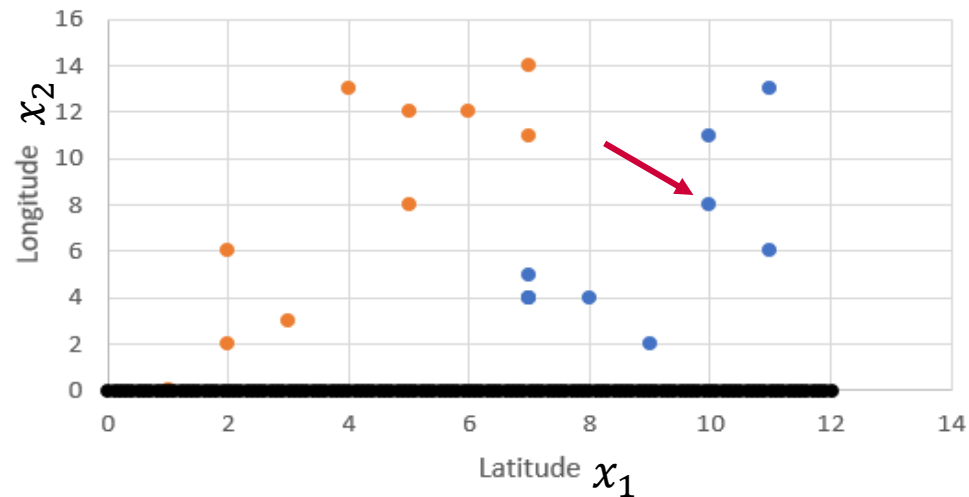
EXEMPLE D'APPLICATION

PROBLÉMATIQUE

- Etape 2 : Mise à jour des poids w_0 , w_1 , w_2

→

| Sites n° | Latitude | Longitude | Pollution |
|----------|----------|-----------|-----------|
| 4 | 10 | 8 | 0 |
| 8 | 10 | 11 | 0 |
| 18 | 11 | 13 | 0 |
| 17 | 7 | 4 | 0 |
| 3 | 7 | 4 | 0 |
| 6 | 7 | 5 | 0 |
| 9 | 11 | 6 | 0 |
| 11 | 7 | 4 | 0 |
| 13 | 9 | 2 | 0 |
| 19 | 8 | 4 | 0 |
| 7 | 6 | 12 | 1 |
| 10 | 4 | 13 | 1 |
| 14 | 5 | 8 | 1 |
| 16 | 5 | 12 | 1 |
| 2 | 7 | 11 | 1 |
| 5 | 1 | 0 | 1 |
| 12 | 3 | 3 | 1 |
| 15 | 7 | 14 | 1 |
| 20 | 2 | 2 | 1 |
| 1 | 2 | 6 | 1 |



$$w_0^2 = w_0^1 - \alpha(y_{i \text{ estimé}} - y_{i \text{ réel}}) = 0 - 0,5 \times 1 = -0,5$$

$$w_1^2 = w_1^1 - \alpha(y_{i \text{ estimé}} - y_{i \text{ réel}})x_1^1 = 0 - 0,5 \times 10 = -5$$

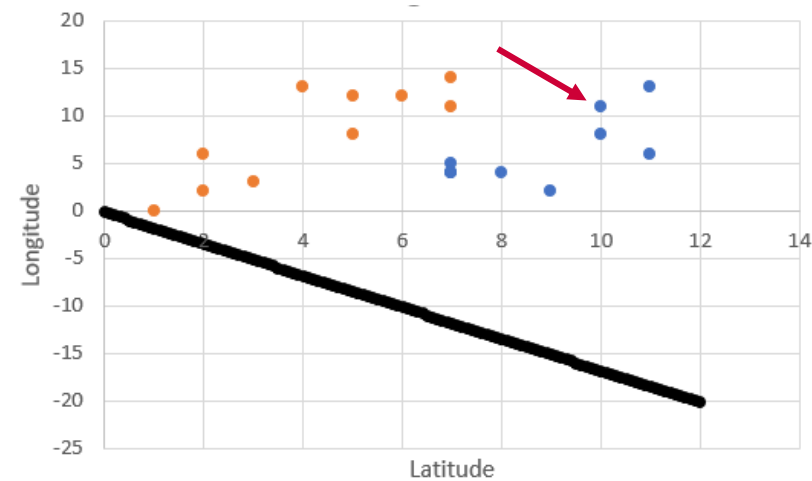
$$w_2^2 = w_2^1 - \alpha(y_{i \text{ estimé}} - y_{i \text{ réel}})x_2^1 = 1 - 0,5 \times 8 = -3$$

EXEMPLE D'APPLICATION

PROBLÉMATIQUE

- Etape 3 : Mise à jour des poids w_0 , w_1 , w_2

| Sites n° | Latitude | Longitude | Pollution |
|----------|----------|-----------|-----------|
| 4 | 10 | 8 | 0 |
| 8 | 10 | 11 | 0 |
| 18 | 11 | 13 | 0 |
| 17 | 7 | 4 | 0 |
| 3 | 7 | 4 | 0 |
| 6 | 7 | 5 | 0 |
| 9 | 11 | 6 | 0 |
| 11 | 7 | 4 | 0 |
| 13 | 9 | 2 | 0 |
| 19 | 8 | 4 | 0 |
| 7 | 6 | 12 | 1 |
| 10 | 4 | 13 | 1 |
| 14 | 5 | 8 | 1 |
| 16 | 5 | 12 | 1 |
| 2 | 7 | 11 | 1 |
| 5 | 1 | 0 | 1 |
| 12 | 3 | 3 | 1 |
| 15 | 7 | 14 | 1 |
| 20 | 2 | 2 | 1 |
| 1 | 2 | 6 | 1 |



$$w_0^3 = w_0^2 - \alpha(y_{i \text{ estimé}} - y_{i \text{ réel}}) = -0,5 - 0,5 \times 0 = -0,5$$

$$w_1^3 = w_1^2 - \alpha(y_{i \text{ estimé}} - y_{i \text{ réel}})x_1^2 = -5 - 0,5 \times 0 = -5$$

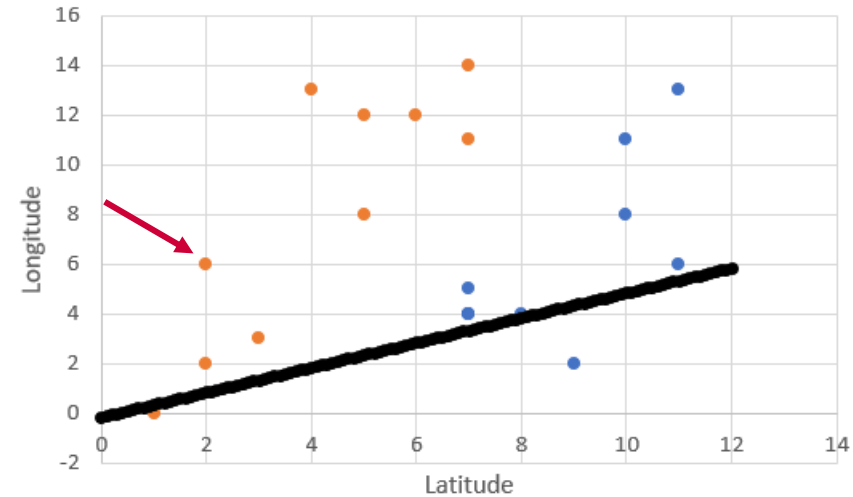
$$w_2^3 = w_2^2 - \alpha(y_{i \text{ estimé}} - y_{i \text{ réel}})x_2^2 = -3 - 0,5 \times 0 = -3$$

EXEMPLE D'APPLICATION

PROBLÉMATIQUE

- Etape 21 : Mise à jour des poids w_0 , w_1 , w_2

| Sites n° | Latitude | Longitude | Pollution |
|----------|----------|-----------|-----------|
| 4 | 10 | 8 | 0 |
| 8 | 10 | 11 | 0 |
| 18 | 11 | 13 | 0 |
| 17 | 7 | 4 | 0 |
| 3 | 7 | 4 | 0 |
| 6 | 7 | 5 | 0 |
| 9 | 11 | 6 | 0 |
| 11 | 7 | 4 | 0 |
| 13 | 9 | 2 | 0 |
| 19 | 8 | 4 | 0 |
| 7 | 6 | 12 | 1 |
| 10 | 4 | 13 | 1 |
| 14 | 5 | 8 | 1 |
| 16 | 5 | 12 | 1 |
| 2 | 7 | 11 | 1 |
| 5 | 1 | 0 | 1 |
| 12 | 3 | 3 | 1 |
| 15 | 7 | 14 | 1 |
| 20 | 2 | 2 | 1 |
| 1 | 2 | 6 | 1 |



$$w_0^{21} = w_0^{20} - \alpha(y_{i \text{ estimé}} - y_{i \text{ réel}}) = 0,5$$

$$w_1^{21} = w_1^{20} - \alpha(y_{i \text{ estimé}} - y_{i \text{ réel}})x_1^{20} = -1,5$$

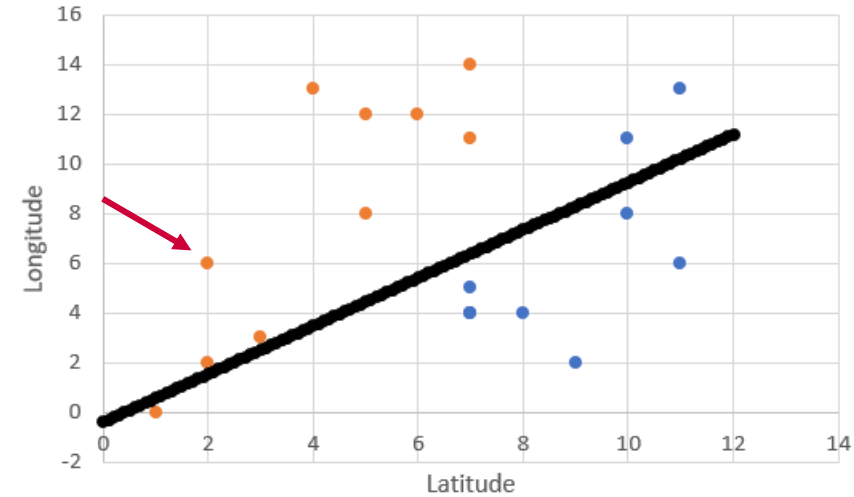
$$w_2^{21} = w_2^{20} - \alpha(y_{i \text{ estimé}} - y_{i \text{ réel}})x_2^{20} = 3$$

EXEMPLE D'APPLICATION

PROBLÉMATIQUE

- Etape 201 : Mise à jour des poids w_0 , w_1 , w_2

| Sites n° | Latitude | Longitude | Pollution |
|----------|----------|-----------|-----------|
| 4 | 10 | 8 | 0 |
| 8 | 10 | 11 | 0 |
| 18 | 11 | 13 | 0 |
| 17 | 7 | 4 | 0 |
| 3 | 7 | 4 | 0 |
| 6 | 7 | 5 | 0 |
| 9 | 11 | 6 | 0 |
| 11 | 7 | 4 | 0 |
| 13 | 9 | 2 | 0 |
| 19 | 8 | 4 | 0 |
| 7 | 6 | 12 | 1 |
| 10 | 4 | 13 | 1 |
| 14 | 5 | 8 | 1 |
| 16 | 5 | 12 | 1 |
| 2 | 7 | 11 | 1 |
| 5 | 1 | 0 | 1 |
| 12 | 3 | 3 | 1 |
| 15 | 7 | 14 | 1 |
| 20 | 2 | 2 | 1 |
| 1 | 2 | 6 | 1 |



$$w_0^{201} = w_0^{200} - \alpha(y_{i \text{ estimé}} - y_{i \text{ réel}}) = 1$$

$$w_1^{201} = w_1^{200} - \alpha(y_{i \text{ estimé}} - y_{i \text{ réel}})x_1^{200} = 5,5$$

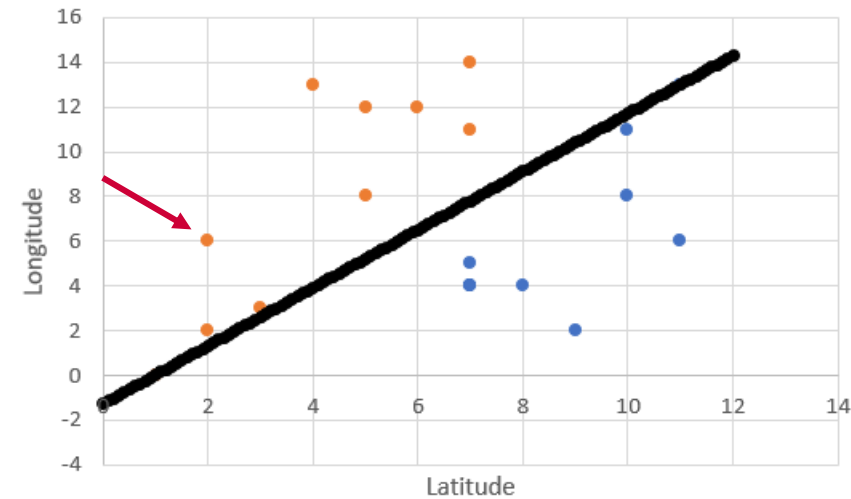
$$w_2^{201} = w_2^{200} - \alpha(y_{i \text{ estimé}} - y_{i \text{ réel}})x_2^{200} = -14$$

EXEMPLE D'APPLICATION

PROBLÉMATIQUE

- Etape 601 : Mise à jour des poids w_0 , w_1 , w_2

| Sites n° | Latitude | Longitude | Pollution |
|----------|----------|-----------|-----------|
| 4 | 10 | 8 | 0 |
| 8 | 10 | 11 | 0 |
| 18 | 11 | 13 | 0 |
| 17 | 7 | 4 | 0 |
| 3 | 7 | 4 | 0 |
| 6 | 7 | 5 | 0 |
| 9 | 11 | 6 | 0 |
| 11 | 7 | 4 | 0 |
| 13 | 9 | 2 | 0 |
| 19 | 8 | 4 | 0 |
| 7 | 6 | 12 | 1 |
| 10 | 4 | 13 | 1 |
| 14 | 5 | 8 | 1 |
| 16 | 5 | 12 | 1 |
| 2 | 7 | 11 | 1 |
| 5 | 1 | 0 | 1 |
| 12 | 3 | 3 | 1 |
| 15 | 7 | 14 | 1 |
| 20 | 2 | 2 | 1 |
| 1 | 2 | 6 | 1 |



$$w_0^{601} = w_0^{600} - \alpha(y_{i \text{ estimé}} - y_{i \text{ réel}}) = 23,5$$

$$w_1^{601} = w_1^{600} - \alpha(y_{i \text{ estimé}} - y_{i \text{ réel}})x_1^{600} = -24$$

$$w_2^{601} = w_2^{600} - \alpha(y_{i \text{ estimé}} - y_{i \text{ réel}})x_2^{600} = 18,5$$

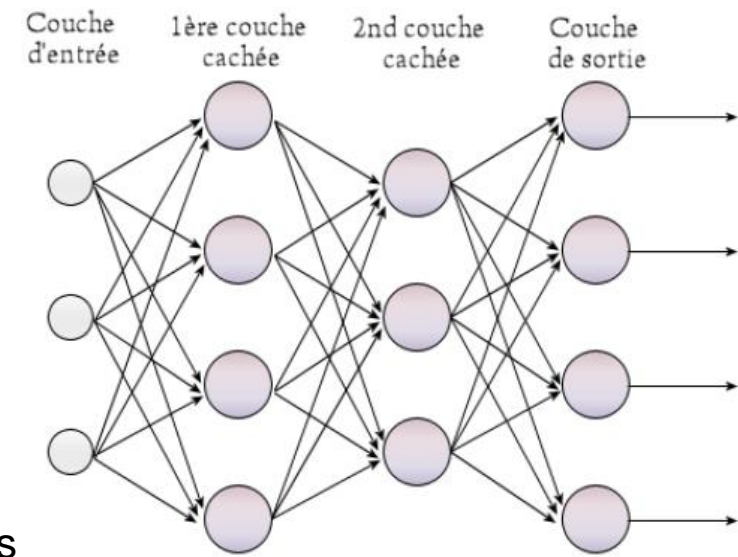


LE RÉSEAU MULTICOUCHE

LE PERCEPTRON MULTICOUCHE

PARAMÈTRES

- **Nombre de couches (Layers)** : dépend du contexte
- **Couche d'entrée** : nombre de neurones d'entrée dictée par les données
Exemple : 784 valeurs d'entrée pour une image 28x28
- **Couche de sortie** : nombre de neurones généralement par le nombre de possibilités
Exemple : 10 neurones de sortie pour une image qui représente un chiffre
- **Couches intermédiaires** : Le moins possible nécessaire pour obtenir des résultats pertinents
- **Nombre de neurones par couche** : Définit généralement par l'expérience
Exemple : nombre compris entre le nombre de neurones d'entrée et de sortie
- **D'autres éléments peuvent influencer les performances du réseau de neurones tels que :**
 - Les poids, biais ;
 - Les fonctions d'activation ;
 - Les taux d'apprentissage ;
 - La fonction LOSS ;
 - L'algorithme d'optimisation : classique, stochastique, ... ;
 - Les époques et taille de lot ;



DOMAINES D'APPLICATION

DEEP LEARNING - APPLICATIONS

| Industrie | Application |
|------------------|---|
| Banque | Prêts et scoring |
| Cartes de crédit | Détection des fraudes |
| Finance | Analyse d'investissements et de fluctuations des taux de change |
| Assurance | Couverture assurantielle et estimation des réserves |
| Marketing | Ciblage des prospections, mesures et comparaisons des campagnes et des méthodes |