

# Intelligence Artificielle



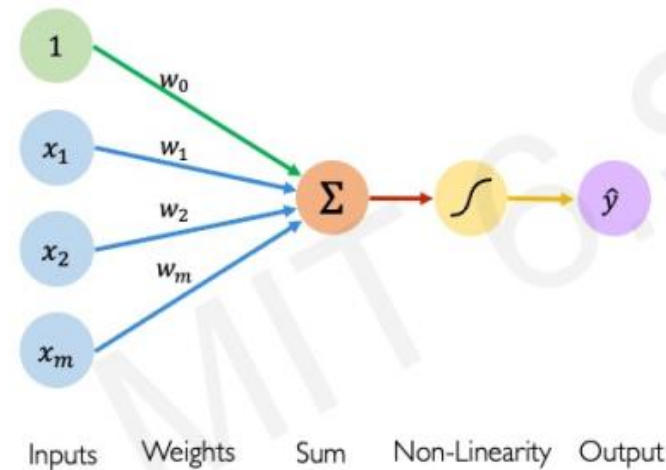
- Dispensé par **MWAMBA KASONGO Dahouda**
- Docteur en génie logiciel et systèmes d'information
- Machine and Deep Learning Engineer

Assisté par Ass. **Jason MOUSSA**

- E-mail : [dahouda37@gmail.com](mailto:dahouda37@gmail.com)
- Tel.: +243 99 66 55 265

Heure : 8H00 – 12H00

## The Perceptron: Forward Propagation

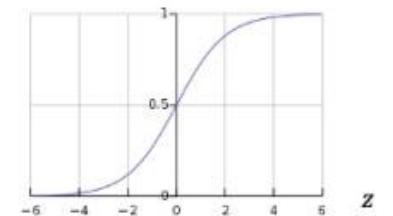


### Activation Functions

$$\hat{y} = g(w_0 + \mathbf{X}^T \mathbf{W})$$

- Example: sigmoid function

$$g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$



## PLAN DU COURS

### CHAPITRE 4 Introduction a l'Apprentissage Profond [Deep Learning]

- 4.1. Concepts de base
- Fonction d'activation
- Shallow Neural Network [Réseau neuronal superficiel]
- Implementation d'un Shallow Neural network



## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

### 4.1. Concepts de base

#### ❖ Qu'est-ce que le Deep Learning ?

Le Deep Learning est une branche de l'apprentissage automatique basée sur l'architecture d'un réseau neuronal artificiel.

Un **réseau neuronal artificiel** (ou RNA) utilise des couches de nœuds interconnectés appelés **neurones** qui fonctionnent ensemble pour traiter et apprendre à partir des données d'entrée.

À l'ère de l'intelligence artificielle, qui évolue rapidement, le Deep Learning est une technologie fondamentale qui révolutionne la façon dont les machines comprennent, apprennent et interagissent avec des données complexes.



## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

### 4.1. Concepts de base

#### ❖ Qu'est-ce que le Deep Learning ?

- ❑ Fondamentalement, l'IA basée sur le Deep Learning imite les réseaux neuronaux complexes du cerveau humain, permettant aux ordinateurs de découvrir des modèles de manière autonome et de prendre des décisions à partir de vastes quantités de données non structurées.
- ❑ Ce domaine transformateur a permis des avancées dans divers domaines, de la vision par ordinateur et du traitement du langage naturel aux diagnostics de santé et à la conduite autonome.



## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

### 4.1. Concepts de base

#### ❖ Qu'est-ce que le Deep Learning ?

Aujourd'hui, le Deep Learning est devenue l'un des domaines les plus populaires et les plus visibles de Machine Learning, en raison de son succès dans une variété d'applications, telles que la vision par ordinateur, le traitement du langage naturel et l'apprentissage par renforcement.

- ❑ L'IA peut être utilisée pour l'apprentissage automatique supervisé, non supervisé ainsi que pour l'apprentissage automatique par renforcement. Elle utilise une variété de moyens pour les traiter.
- **Apprentissage supervisé** : L'apprentissage automatique supervisé est la technique d'apprentissage automatique dans laquelle le réseau neuronal apprend à faire des prédictions ou à classer des données en fonction des ensembles de données étiquetés.



## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

### 4.1. Concepts de base

#### ❖ Qu'est-ce que le Deep Learning ?

- **Apprentissage non supervisé** : est la technique d'apprentissage automatique dans laquelle le réseau neuronal apprend à découvrir les modèles ou à regrouper l'ensemble de données en fonction d'ensembles de données non étiquetés.
- **Apprentissage par renforcement** : l'apprentissage automatique par renforcement est la technique d'apprentissage automatique dans laquelle un agent apprend à prendre des décisions dans un environnement pour maximiser un signal de récompense.  
L'agent interagit avec l'environnement en agissant et en observant les récompenses qui en résultent.



## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

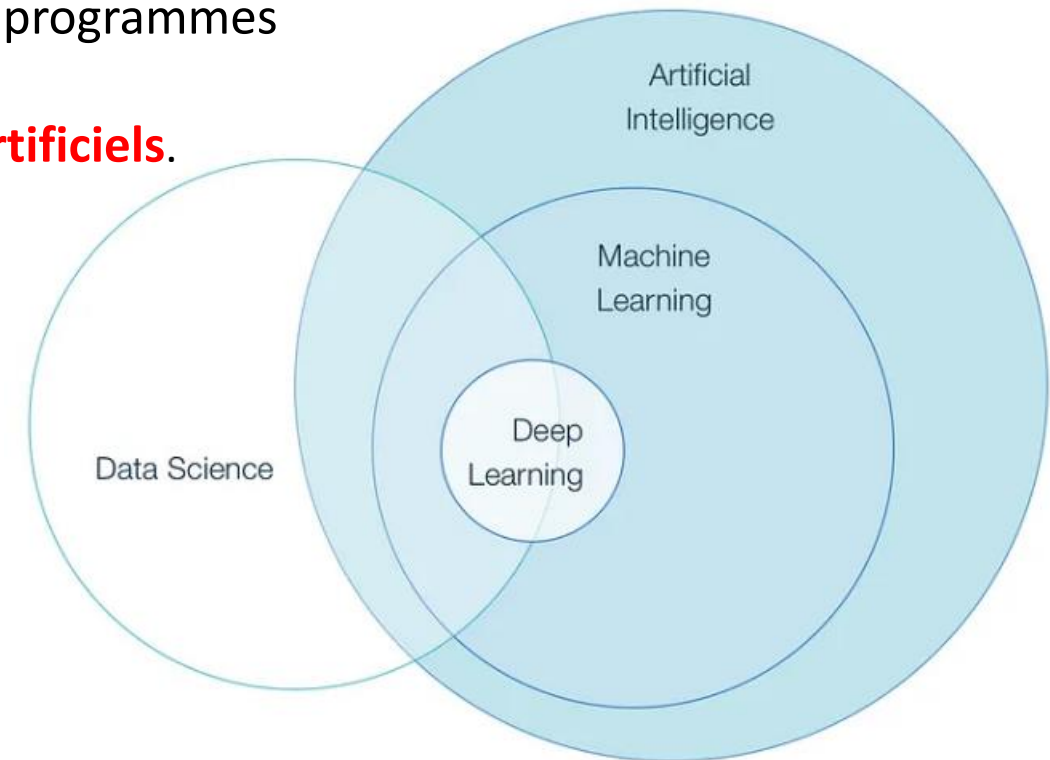
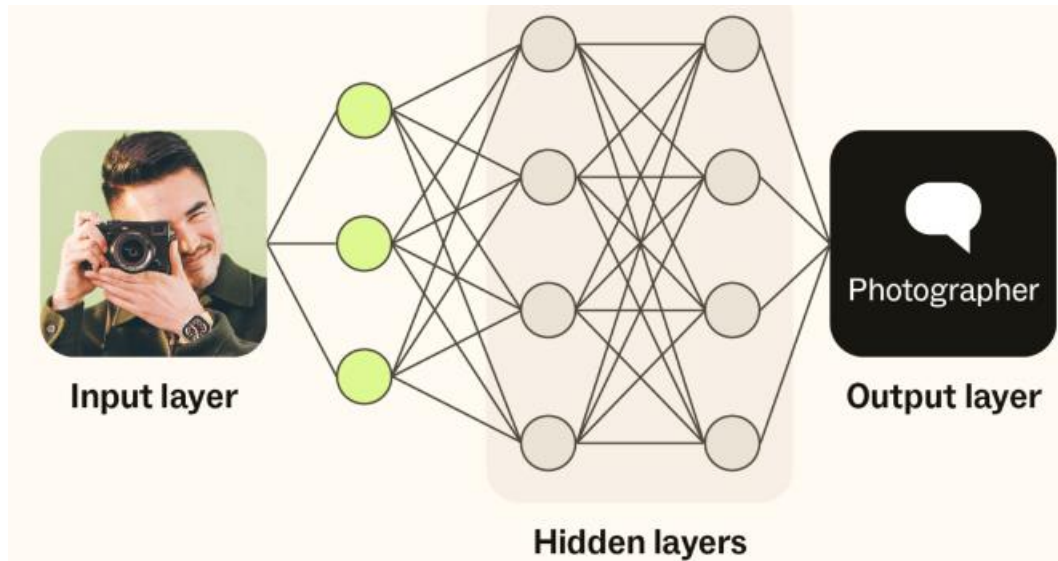
### 4.1. Concepts de base

#### ❖ Qu'est-ce que le Deep Learning ?

Le Deep Learning est une branche de l'apprentissage automatique basée sur l'architecture d'un réseau neuronal artificiel.

Il s'agit d'un sous-ensemble de l'apprentissage automatique, les programmes résolvent des tâches sans être explicitement programmés.

Il est plus spécifique car ils entraînent des **réseaux neuronaux artificiels**.





## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

### 4.1. Concepts de base

#### ❖ Qu'est-ce qu'un réseau neuronal [Neural Network]?



- ☐ Les réseaux neuronaux artificiels sont construits sur les principes de la structure et du fonctionnement des neurones humains.
- ☐ On les appelle également réseaux neuronaux.
- ☐ La couche d'entrée d'un réseau neuronal artificiel, qui est la première couche, reçoit des entrées provenant de sources externes et les transmet à la couche cachée, qui est la deuxième couche.
- ☐ Chaque neurone de la couche cachée obtient des informations des neurones de la couche précédente, calcule le total pondéré, puis le transfère aux neurones de la couche suivante.

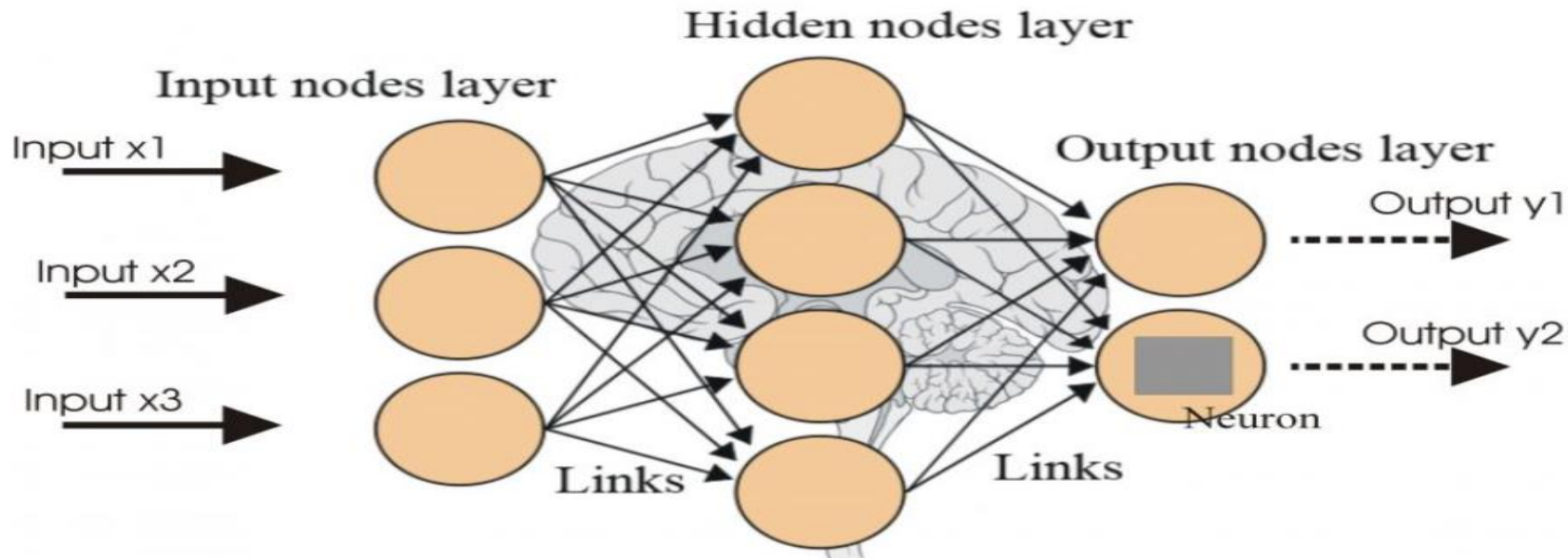


## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

### 4.1. Concepts de base

#### ❖ Qu'est-ce qu'un réseau neuronal [Neural Network]?

- ❑ Ces connexions sont pondérées, ce qui signifie que les impacts des entrées de la couche précédente sont plus ou moins optimisés en donnant à chaque entrée un poids distinct.
- ❑ Ces poids sont ensuite ajustés pendant le processus de formation pour améliorer les performances du modèle.



## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

### 4.1. Concepts de base

#### ❖ Perceptron dans l'apprentissage automatique

- ☐ Dans le domaine de l'apprentissage automatique et de l'intelligence artificielle, le terme Perceptron est le plus couramment utilisé.
- ☐ Il s'agit de la première étape de l'apprentissage des technologies de Machine Learning et de Deep Learning, qui consiste en un ensemble de pondérations, de valeurs d'entrée ou de scores et d'un seuil.
- ☐ Le Perceptron est un élément constitutif d'un réseau neuronal artificiel.
- ☐ Perceptron est un algorithme de machine learning pour l'apprentissage supervisé de diverses tâches de classification binaire.



## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

### 4.1. Concepts de base

#### ❖ Perceptron dans l'apprentissage automatique

- ☐ De plus, Perceptron est également considéré comme un **neurone artificiel** ou **une unité de réseau neuronal** qui aide à détecter certains calculs de données d'entrée dans la veille stratégique.
- ☐ Le modèle Perceptron est également considéré comme l'un des meilleurs et des plus simples types de réseaux neuronaux artificiels.
- ☐ Cependant, il s'agit d'un algorithme d'apprentissage supervisé de classificateurs binaires.



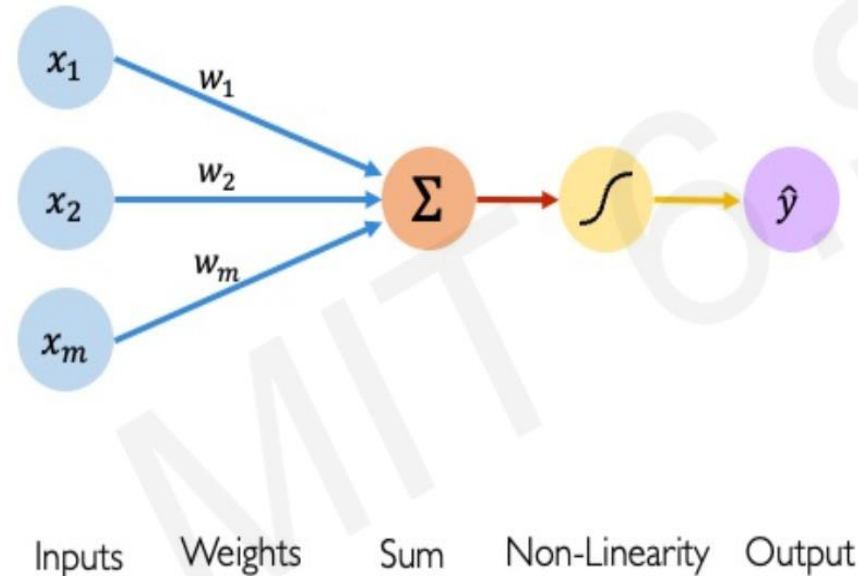
## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

### 4.1. Concepts de base

#### ❖ Perceptron dans l'apprentissage automatique

❑ M. Frank Rosenblatt a inventé le modèle du perceptron comme classificateur contient trois composants principaux à savoir:

- ✓ Valeurs d'entrée [Inputs],
- ✓ Poids et le biais [Weight et bias],
- ✓ Somme [Sum],
- ✓ Fonction d'activation [Activation Function].



Linear combination of inputs

$$\hat{y} = g \left( \sum_{i=1}^m x_i w_i \right)$$

Non-linear activation function

Output

## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING



### 4.1. Concepts de base

#### ❖ Perceptron dans l'apprentissage automatique

- 1. Nœuds d'entrée ou couche d'entrée [Inputs]** : Il s'agit du composant principal du Perceptron qui accepte les données initiales dans le système pour un traitement ultérieur. Chaque nœud d'entrée contient une valeur numérique réelle.
- 2. Poids et biais** : Le paramètre de poids représente la force de la connexion entre les unités. Il s'agit d'un autre paramètre très important des composants du Perceptron. Le poids est directement proportionnel à la force du neurone d'entrée associé pour décider de la sortie. De plus, le biais peut être considéré comme **la ligne d'interception** dans une équation linéaire.
- 3. Somme**: La somme pondérée pour améliorer les performances du modèle
- 4. Fonction d'activation** : Les composants finaux et importants qui aident à déterminer si le neurone va s'activer ou non. La fonction d'activation peut être considérée principalement comme une fonction en escalier

## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

### 4.1. Concepts de base

#### 4.1.1 Fonction d'activation

Une fonction d'activation est un élément clé d'un réseau neuronal qui détermine si un neurone doit être

- ✓ Elle introduit une non-linéarité dans le modèle, ce qui permet au réseau de capturer des modèles complexes.
- ✓ Sans fonctions d'activation, les réseaux neuronaux se comporteraient simplement comme des modèles linéaires, quelle que soit leur profondeur

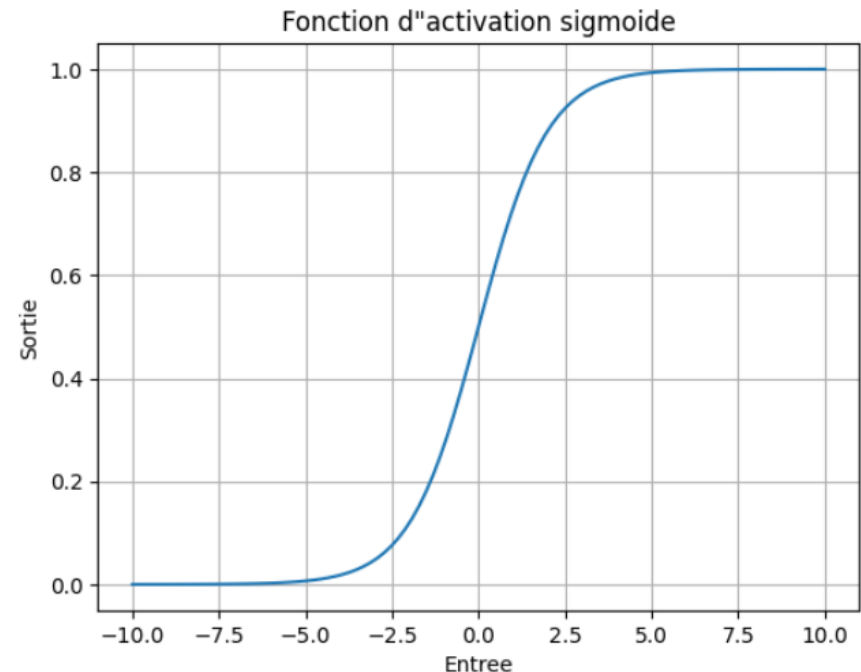
#### ❖ Fonction d'activation sigmoïde

➤ La fonction sigmoïde génère des valeurs comprises entre 0 et 1.  
Elle est couramment utilisée dans la couche de sortie pour les problèmes de classification binaire (par exemple, la régression logistique).

➤ Formule: 
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

➤ Caractéristiques :

- ✓ Plage : (0, 1)
- ✓ Cas d'utilisation : **classification binaire**, où vous souhaitez obtenir la sortie sous forme de probabilité.



## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

### 4.1. Concepts de base

#### 4.1.1 Fonction d'activation

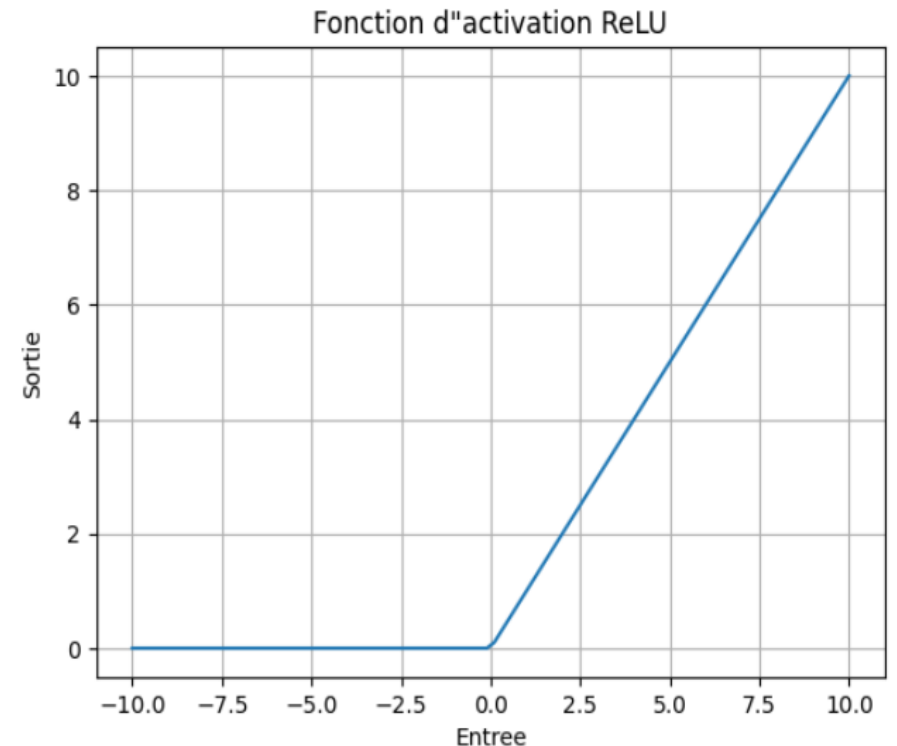
##### ❖ Fonction d'activation ReLU (Rectified Linear Unit)

- La fonction ReLU est la fonction d'activation la plus utilisée dans les architectures modernes de Deep Learning.  
Elle génère directement l'entrée si elle est positive, sinon elle génère zéro.

- Formule:  $f(x) = \max(0, x)$

##### ➤ Caractéristiques :

- ✓ Plage :  $[0, \infty)$
- ✓ Cas d'utilisation : le plus souvent utilisé dans les couches cachées des réseaux neuronaux à propagation directe et convolutionnels.





## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

### 4.1. Concepts de base

#### 4.1.1 Fonction d'activation

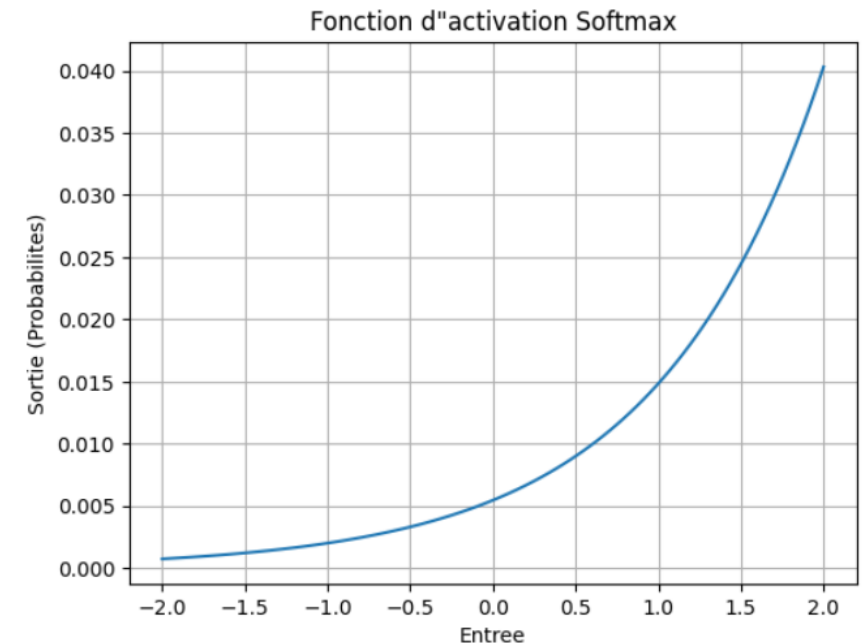
##### ❖ Fonction d'activation Softmax

➤ La fonction softmax convertit les logits en probabilités. Elle est généralement utilisée dans la couche de sortie pour les problèmes de classification multi-classes.

➤ **Formule:**  $\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$

➤ **Caractéristiques :**

- ✓ Plage : (0, 1) et la somme de toutes les sorties est égale à 1.
- ✓ Cas d'utilisation : **classification multi-classes** où vous souhaitez que la sortie représente les probabilités sur différentes classes.



## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

### 4.1. Concepts de base

#### 4.1.1 Fonction d'activation



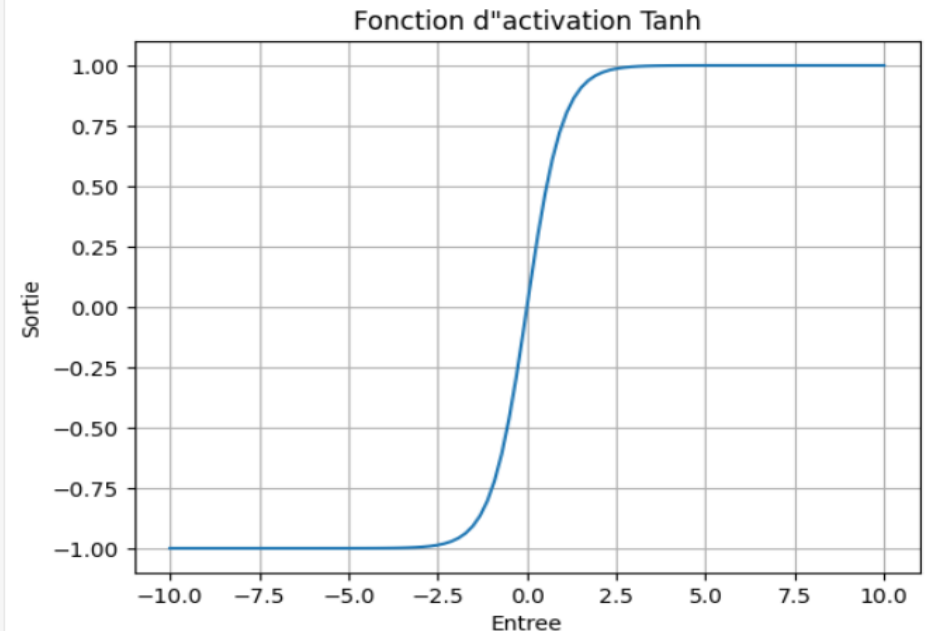
#### ❖ Fonction d'activation Tanh (Hyperbolic Tangent)

➤ La fonction tanh génère des valeurs comprises entre -1 et 1.  
Elle est souvent utilisée dans les couches cachées des réseaux neuronaux, en particulier dans les tâches où les valeurs négatives sont importantes.

➤ **Formule:**  $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

➤ **Caractéristiques :**

- ✓ Plage :  $(-1, 1)$
- ✓ Cas d'utilisation : utilisé lorsque vous souhaitez que la sortie soit comprise entre des valeurs négatives et positives.



## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

### 4.1. Concepts de base

#### 4.1.2 Shallow Neural Network



Un réseau neuronal superficiel [**Shallow Neural Network (SNN)**] est un réseau neuronal avec seulement une ou deux couches cachées.

❑ Il est plus simple que les réseaux neuronaux profonds [Deep Neural Network (DNN)], qui se composent de nombreuses couches cachées.

Bien que les réseaux superficiels puissent bien fonctionner sur des tâches plus simples, ils peuvent avoir des difficultés avec des ensembles de données plus complexes qui nécessitent des représentations approfondies.

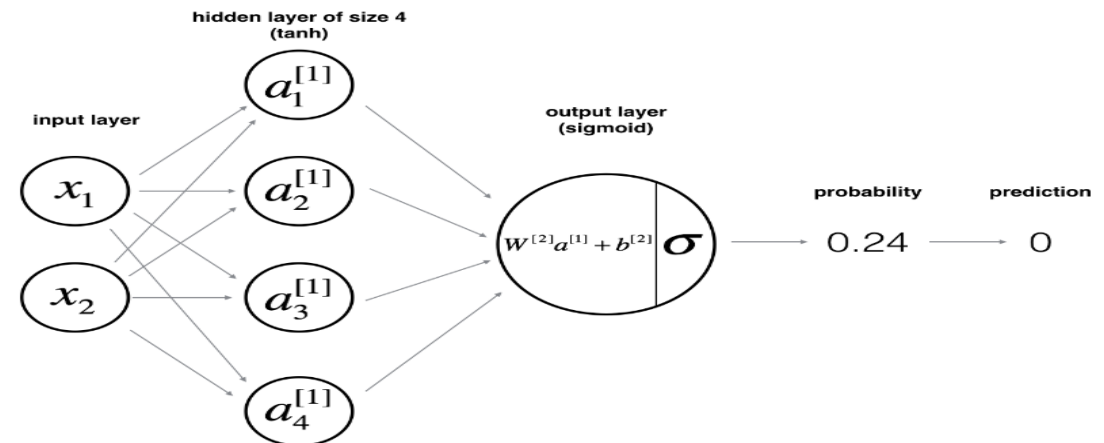
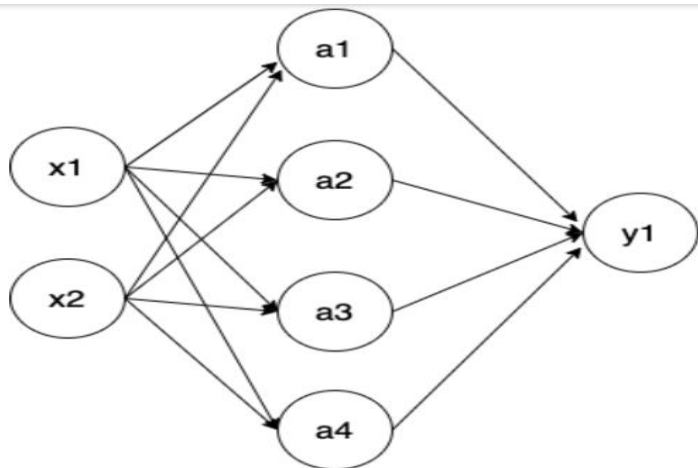
## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

### 4.1. Concepts de base

#### 4.1.2 Shallow Neural Network

□ La structure de base d'un réseau neuronal superficiel comprend :

1. **Couche d'entrée [Input layer]** : reçoit les données d'entrée.
2. **Couche cachée [Hidden layer]** : une seule couche avec des neurones qui appliquent des fonctions d'activation pour apprendre des modèles à partir des données.
3. **Couche de sortie [Output layer]** : fournit les prédictions finales (pour la classification ou la régression).



## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

### 4.1. Concepts de base

#### 4.1.2. Shallow Neural Network

Pour calculer la sortie d'un réseau neuronal superficiel, passons en revue les étapes à l'aide d'un exemple.  
Un réseau neuronal superficiel comporte généralement une couche cachée.

❖ Nous supposons ce qui suit

#### 1. Supposons les couches suivantes:

[Couche d'entrée] *Input Layer*:  $x_1$  et  $x_2$

[Couche cachée] *Hidden Layer*: 2 neurones

[Couche de sortie] *output Layer*: 1 neurone

#### 2. Paramètres du réseau neuronal superficiel

Nous utiliserons les éléments suivants :

$w_{11}, w_{12}$  pour le neurone 1 de la couche cachée

$w_{21}, w_{22}$  pour le neurone 2 de la couche cachée

#### 3. Biais pour la couche cachée

$b_1$  pour le neurone 1

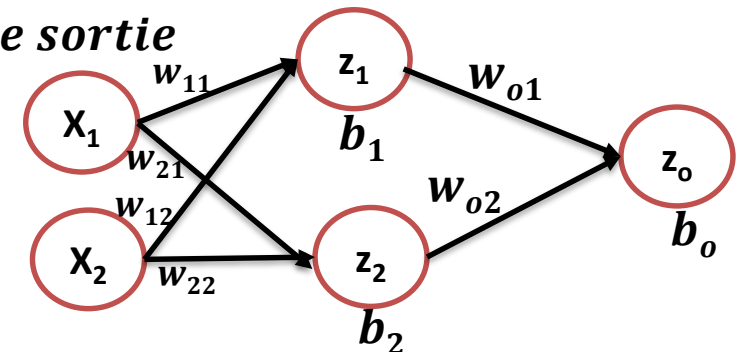
$b_2$  pour le neurone 2

#### 4. Weights pour la couche de sortie

$w_{o1}, w_{o2}$  sont les weights reliant les neurones de la couche cachée au neurone de sortie.

#### 5. Biais pour la couche de sortie

$b_0$  pour la couche de sortie



## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

### 4.1. Concepts de base

#### 4.1.2. Shallow Neural Network



##### ❖ Étape 1 : Calculer les sorties de la couche cachée

✓ Nous calculons d'abord l'entrée nette de chaque neurone de la couche cachée.

##### ▪ Entrée nette du neurone caché 1 :

$$Z_1 = w_{11}x_1 + w_{12}x_2 + b_1$$

##### ▪ Entrée nette du neurone caché 2 :

$$Z_2 = w_{21}x_1 + w_{22}x_2 + b_2$$

✓ Nous appliquons maintenant une fonction d'activation à ces entrées nettes.

✓ Supposons que nous utilisons la fonction d'activation ReLU :

$$a_1 = \text{ReLU}(Z_1) = \max(0, Z_1)$$

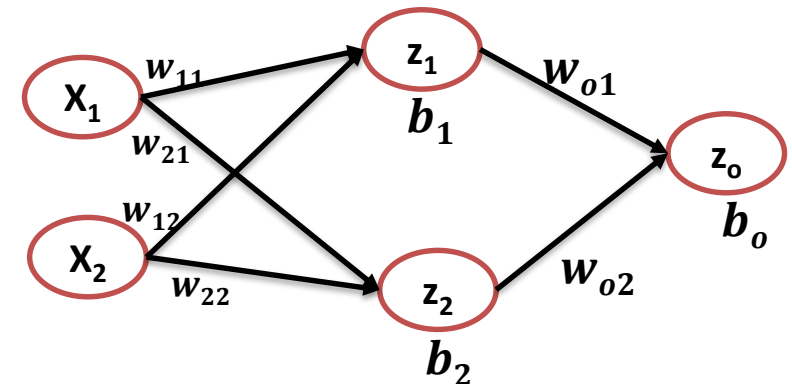
$$a_2 = \text{ReLU}(Z_2) = \max(0, Z_2)$$

##### ❖ Étape 2 : Calculer la couche de sortie

✓ Maintenant, calculez l'entrée nette de la couche de sortie :

$$Z_o = w_{o1}a_1 + w_{o2}a_2 + b_o$$

Le résultat final est simplement la valeur de  $Z_o$ , en supposant que nous utilisons une activation linéaire au niveau de la couche de sortie.



## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING



### 4.1. Concepts de base

#### 4.1.2. Shallow Neural Network

Pour calculer la sortie d'un réseau neuronal superficiel, passons en revue les étapes à l'aide d'un exemple.  
Un réseau neuronal superficiel comporte généralement une couche cachée.

❖ **Exemple 1** : Calcul d'un réseau neuronal superficiel avec des valeurs

1. Supposons les valeurs suivantes:

Inputs:  $x_1 = 1, x_2 = 2$

2. Weights

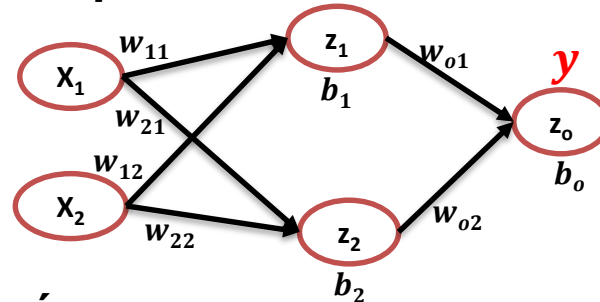
$$w_{11} = 0.5, w_{12} = 0.3$$

$$w_{21} = 0.4, w_{22} = 0.7$$

$$w_{o1} = 0.6, w_{o2} = 0.9$$

3. Bias

$$b_1 = 0.1, b_2 = 0.2, b_o = -0.1$$



❖ **Étape 2** : Calculer la couche de sortie

$$Z_o = w_{o1}a_1 + w_{o2}a_2 + b_o$$

$$Z_o = (0.6 * 1.2) + (0.9 * 2.0) + (-0.1)$$

$$Z_o = 0.72 + 1.8 - 0.1 \Rightarrow Z_o = 2.42$$

❖ **Résultat final** :  $y = 2.42$

❖ **Étape 1** : Calculer les entrées et sorties des neurones de la couche cachée

▪ **Entrée nette du neurone caché 1 :**

$$Z_1 = w_{11}x_1 + w_{12}x_2 + b_1$$

$$Z_1 = (0.5 * 1) + (0.3 * 2) + 0.1$$

$$Z_1 = 0.5 + 0.6 + 0.1 \Rightarrow Z_1 = 1.2$$

$$a_1 = \text{ReLU}(Z_1) = \max(0, Z_1)$$

$$a_1 = \text{ReLU}(1.2) = \max(0, 1.2) \Rightarrow a_1 = 1.2$$

▪ **Entrée nette du neurone caché 2 :**

$$Z_2 = w_{21}x_1 + w_{22}x_2 + b_2$$

$$Z_2 = (0.4 * 1) + (0.7 * 2) + 0.2$$

$$Z_2 = 0.4 + 1.4 + 0.2 \Rightarrow Z_2 = 2.0$$

$$a_2 = \text{ReLU}(Z_2) = \max(0, Z_2)$$

$$a_2 = \text{ReLU}(2.0) = \max(0, 2.0) \Rightarrow a_2 = 2.0$$



## 4.1. Concepts de base

### 4.1.2. Shallow Neural Network

### ❖ Exemple 2 : Calcul d'un réseau neuronal superficiel avec des valeurs

*Inputs:*    **x1 = 2, x2 = 1**

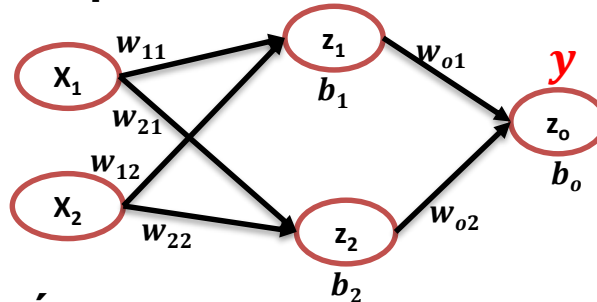
$$w_{11} = 0.2, w_{12} = 0.4$$

$$w_{21} = 0.3, w_{22} = 0.5$$

$$w_{o1} = 0.5 \quad w_{o2} = 0.4$$

### 3. Bias

$$b_1 = 0.2, b_2 = 0.3 \quad b_0 = -0.2$$



❖ **Étape 2 : Calculer la couche de sortie**

$$\mathbf{Z}_o = w_{o1}a_1 + w_{o2}a_2 + b_o$$

$$\mathbf{Z}_0 = (0.6 * 1.2) + (0.9 * 2.0) + (-0.1)$$

$$Z_0 = 0.72 + 1.8 - 0.1 \Rightarrow Z_0 = 2.42$$

❖ **Résultat final :  $y = 2.42$**

## Étape 1 : Calculer les entrées et sorties des neurones de la couche cachée

- **Entrée nette du neurone caché 1 :**

$$\mathbf{Z}_1 = \mathbf{w}_{11}x_1 + \mathbf{w}_{12}x_2 + \mathbf{b}_1$$

$$\mathbf{Z}_1 = (0.5 * \mathbf{1}) + (0.3 * \mathbf{2}) + 0.1$$

$$Z_1 = 0.5 + 0.6 + 0.1 \Rightarrow Z_1 = 1.2$$

$$a_1 = ReLU(Z_1) = \max(0, Z_1)$$

$$a_1 = \text{ReLU}(1.2) = \max(0, 1.2) \Rightarrow a_1 = 1.2$$

- **Entrée nette du neurone caché 2 :**

$$\mathbf{Z}_2 = \mathbf{w}_{21}x_1 + \mathbf{w}_{22}x_2 + \mathbf{b}_2$$

$$\mathbf{Z}_2 = (\mathbf{0.4} * \mathbf{1}) + (\mathbf{0.7} * \mathbf{2}) + \mathbf{0.2}$$

$$Z_2 = 0.4 + 1.4 + 0.2 \Rightarrow Z_2 = 2.0$$

$$a_2 = ReLU(Z_2) = \max(0, Z_2)$$

$$a_2 = \text{ReLU}(2.0) = \max(0, 2.0) \Rightarrow \mathbf{a_2 = 2.0}$$

## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING



### TD : Probleme de Classification Binaire

### Shallow Neural Network

Pour calculer la sortie d'un réseau neuronal superficiel, passons en revue les étapes à l'aide d'un exemple.  
Un réseau neuronal superficiel comporte généralement une couche cachée.

#### ❖ Exemple 3 : Calcul d'un réseau neuronal superficiel avec des valeurs

##### 1. Supposons les valeurs suivantes:

Inputs:  $x_1 = 2, x_2 = 1$

##### 2. Weights

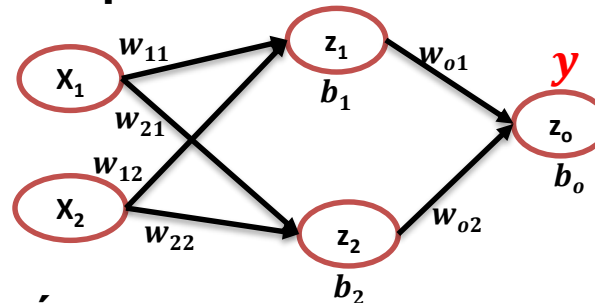
$$w_{11} = 0.2, w_{12} = 0.4$$

$$w_{21} = 0.3, w_{22} = 0.5$$

$$w_{o1} = 0.5, w_{o2} = 0.4$$

##### 3. Bias

$$b_1 = 0.2, b_2 = 0.3, b_o = -0.2$$



#### ❖ Étape 2 : Calculer la couche de sortie

$$Z_o = w_{o1}a_1 + w_{o2}a_2 + b_o$$

#### ❖ Résultat final :

#### ❖ Étape 1 : Calculer les entrées et sorties des neurones de la couche cachée

##### ■ Entrée nette du neurone caché 1 :

$$Z_1 = w_{11}x_1 + w_{12}x_2 + b_1$$

$$a_1 = \text{ReLU}(Z_1) = \max(0, Z_1)$$

##### ■ Entrée nette du neurone caché 2 :

$$Z_2 = w_{21}x_1 + w_{22}x_2 + b_2$$

$$a_2 = \text{ReLU}(Z_2) = \max(0, Z_2)$$

## CHAPITRE 4 INTRODUCTION AU DEEP LEARNING

### 4.1. Concepts de base

#### 4.1.3 Implementation d'un Shallow Neural network [Lab]

- ❖ Étape 1 : Importer les packages
- ❖ Étape 2 : Charger les données
- ❖ Étape 3 : Analyse de données
- ❖ Étape 4 : Prétraitement des données
- ❖ Étape 5 : Diviser les données : Train et Test
- ❖ Étape 6 : Construire le Shallow Neural Network
- ❖ Étape 7 : Visualiser l'évolution de l'entraînement du réseau de neurone
- ❖ Étape 8 : Evaluer le modèle
- ❖ Étape 9 : Sauvegarder le modèle entraîné

