

**Cours-TD d'introduction
à l'Intelligence Artificielle
Partie II**

Le Neurone Formel

Simon Gay

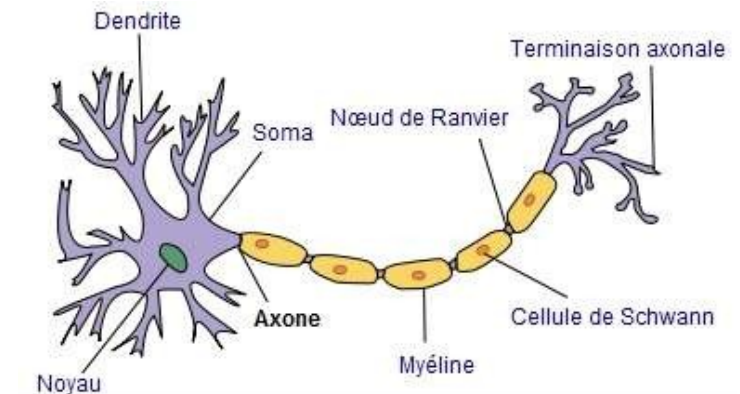
Introduction à l'Intelligence Artificielle

- **Menu :**
 - Théorie :
 - le neurone formel
 - Pratique :
 - implémentation d'un neurone formel
 - Création d'outils pour l'entraînement et le test de neurones
 - Apprentissage sur une base d'images

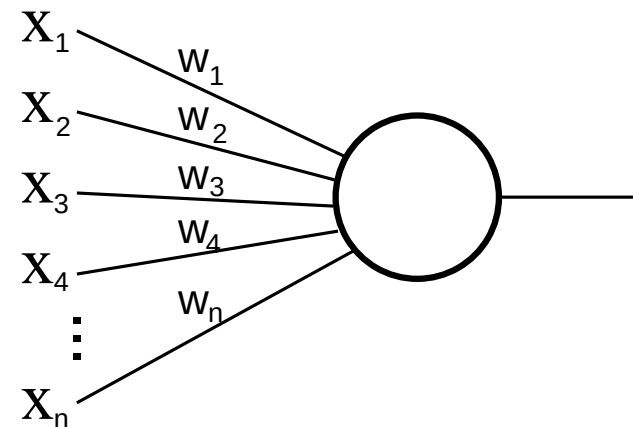
Introduction à l'Intelligence Artificielle

- **Vers un modèle simplifié du neurone :**

- Le neurone biologique :
 - Des dendrites qui collectent les signaux d'entrée
 - Un corps qui intègre les signaux
 - Un axone pour transmettre le signal
 - Un fonctionnement par impulsion (fréquence)



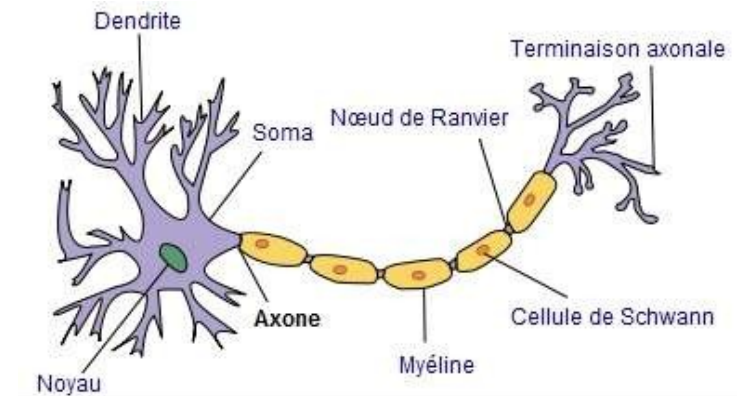
- Il faut simplifier le modèle et l'adapter au fonctionnement d'un ordinateur
 - On remplace la fréquence par une valeur réelle
 - En entrée : un vecteur de valeurs
 - Vecteur $X = \{x_1, x_2, x_3, \dots, x_n\}$
 - Un vecteur de réel pour les synapses
 - Vecteur $W = \{w_1, w_2, w_3, \dots, w_n\}$



Introduction à l'Intelligence Artificielle

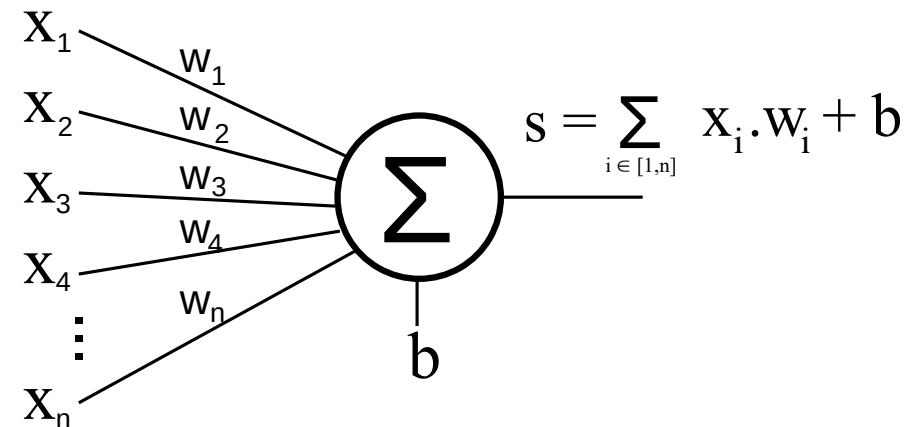
- **Vers un modèle simplifié du neurone :**

- Le neurone biologique :
 - Des dendrites qui collectent les signaux d'entrée
 - Un corps qui intègre les signaux
 - Un axone pour transmettre le signal
 - Un fonctionnement par impulsion (fréquence)



- Il faut simplifier le modèle et l'adapter au fonctionnement d'un ordinateur

- Pour l'intégration des entrées :
 - Somme des entrées pondérées par les poids synaptiques
 - Un biais b

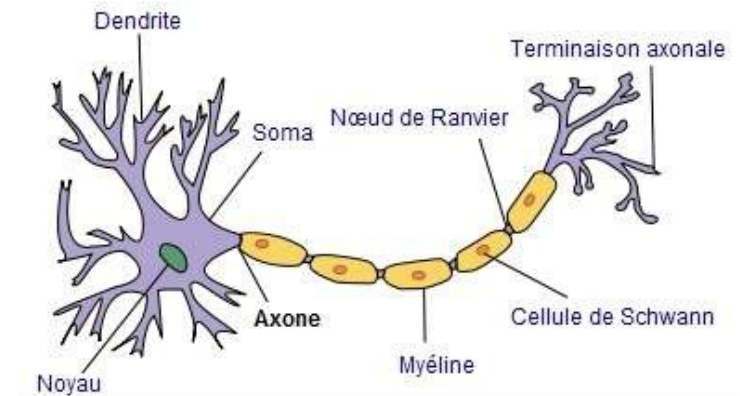


Introduction à l'Intelligence Artificielle

- **Vers un modèle simplifié du neurone :**

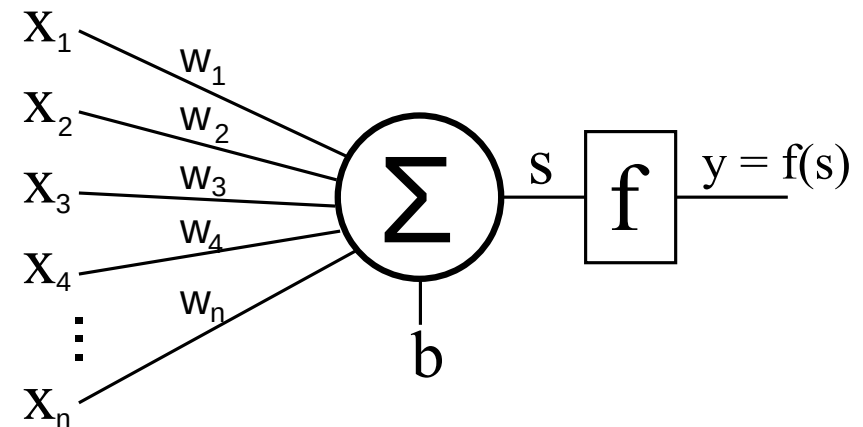
- Le neurone biologique :

- Des dendrites qui collectent les signaux d'entrée
 - Un corps qui intègre les signaux
 - Un axone pour transmettre le signal
 - Un fonctionnement par impulsion (fréquence)



- Il faut simplifier le modèle et l'adapter au fonctionnement d'un ordinateur

- La sortie passe par une fonction d'activation
 - Permet de restreindre la valeur



Introduction à l'Intelligence Artificielle

- **Vers un modèle simplifié du neurone :**

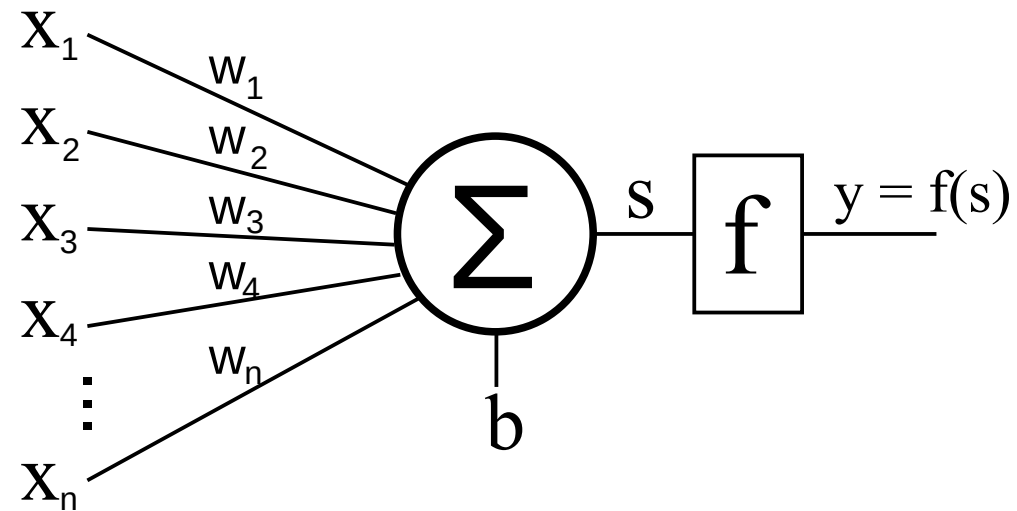
- Le neurone formel

- Un vecteur d'entrée X
 - Un ensemble de poids W
 - On ajoute un biais b
 - Équivalent à une entrée à 1

$$s = \sum_{i \in [1,n]} X_i \cdot W_i$$

- Une fonction d'activation
 $Y = f(S)$

On reviendra sur les fonctions d'activation



Introduction à l'Intelligence Artificielle

- **Méthode d'apprentissage supervisé**

- Méthode dérivée de la loi de Hebb
 - Règle de Widrow-Hoff : le poids est modifié proportionnellement à l'erreur
- La sortie est une valeur réelle
- La valeur recherchée est une valeur de $\{ 0 ; 1 \}$ pour un vecteur X donné
 - on cherche à modifier les poids (vecteur W) pour que la valeur de sortie du neurone se rapproche de la valeur désirée
- Idée : modifier chaque poids w_i proportionnellement à :
 - la valeur d'entrée (plus l'entrée a une valeur élevée, plus elle influe sur le résultat)
 - la différence entre le résultat et la sortie attendue (plus l'écart est grand plus il faut modifier les poids)

Introduction à l'Intelligence Artificielle

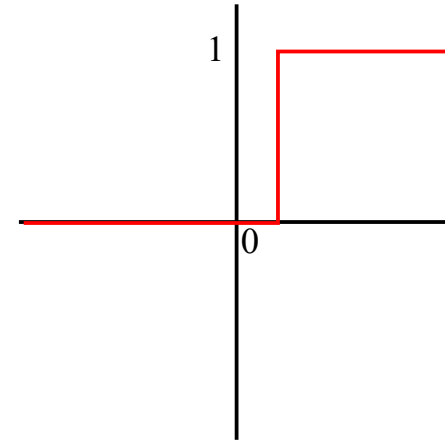
- **Méthode d'apprentissage supervisé (Widrow-Hoff)**

- On calcule l'écart entre le résultat r voulu et le résultat y du neurone :
 - $\Delta = r - y$
(à noter : on utilise parfois le carré de la différence, en conservant le signe)
- On met chaque poids à jour :
 - $w_i^{t+1} = w_i^t + \alpha \cdot x_i \cdot \Delta$ (note : loi de hebb : $w_i^{t+1} = w_i^t + \alpha \cdot x_i \cdot r$)
 α est le *taux d'apprentissage*, limitant la variation de chaque apprentissage même mise à jour pour le biais b
- Et c'est tout !
- L'apprentissage s'effectue sur un (très) grand nombre d'exemples

Introduction à l'Intelligence Artificielle

- Quelle fonction d'activation ?

- Fonction seuil : $S = 1$ si $y > \text{seuil}$, $S=0$ sinon



(note : la valeur du seuil n'a pas d'importance grâce au biais)

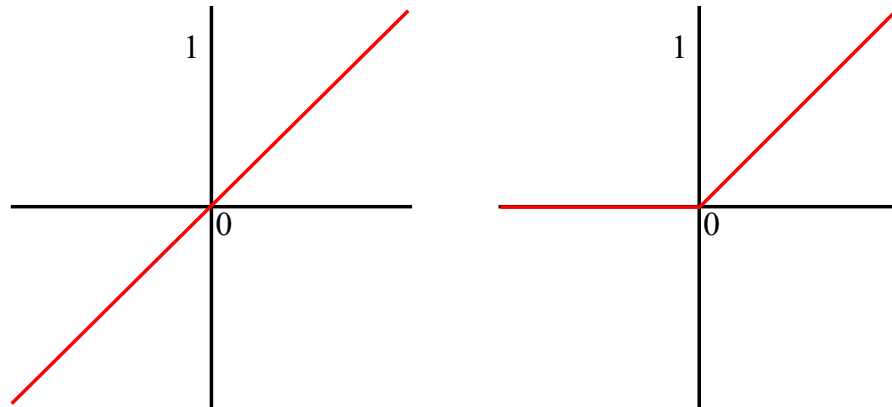
- Fonction utilisée sur les premiers modèles de neurones formels (proposé par Warren McCulloch et Walter Pitts en **1943**)
- Quelques limitations :
 - On ne peut pas comparer les valeurs de plusieurs neurones
 - Les valeurs de l'information ne peuvent pas être « transmises »

Introduction à l'Intelligence Artificielle

- Quelle fonction d'activation ?

- Fonctions linéaires et ReLU (Rectified Linear Unit)

- $y = x$
 - $y = \max(0, x)$



- Fonctions très simples

- ReLU permet de supprimer les valeurs négatives

- très utilisée dans les approches de deep learning

- La valeur de sortie peut dépasser 1 (la somme pondérée doit être égale à 1)

- Gradient constant (une variation du poids a un même effet sur la sortie)

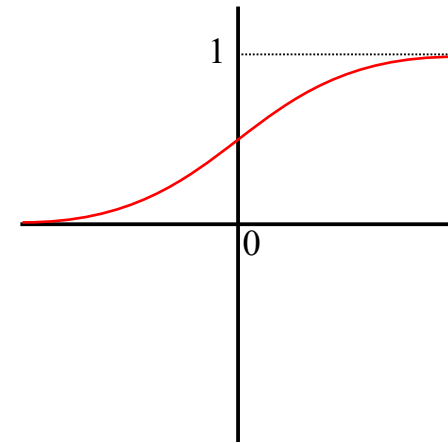
Introduction à l'Intelligence Artificielle

- **Quelle fonction d'activation ?**

- Fonction sigmoïde

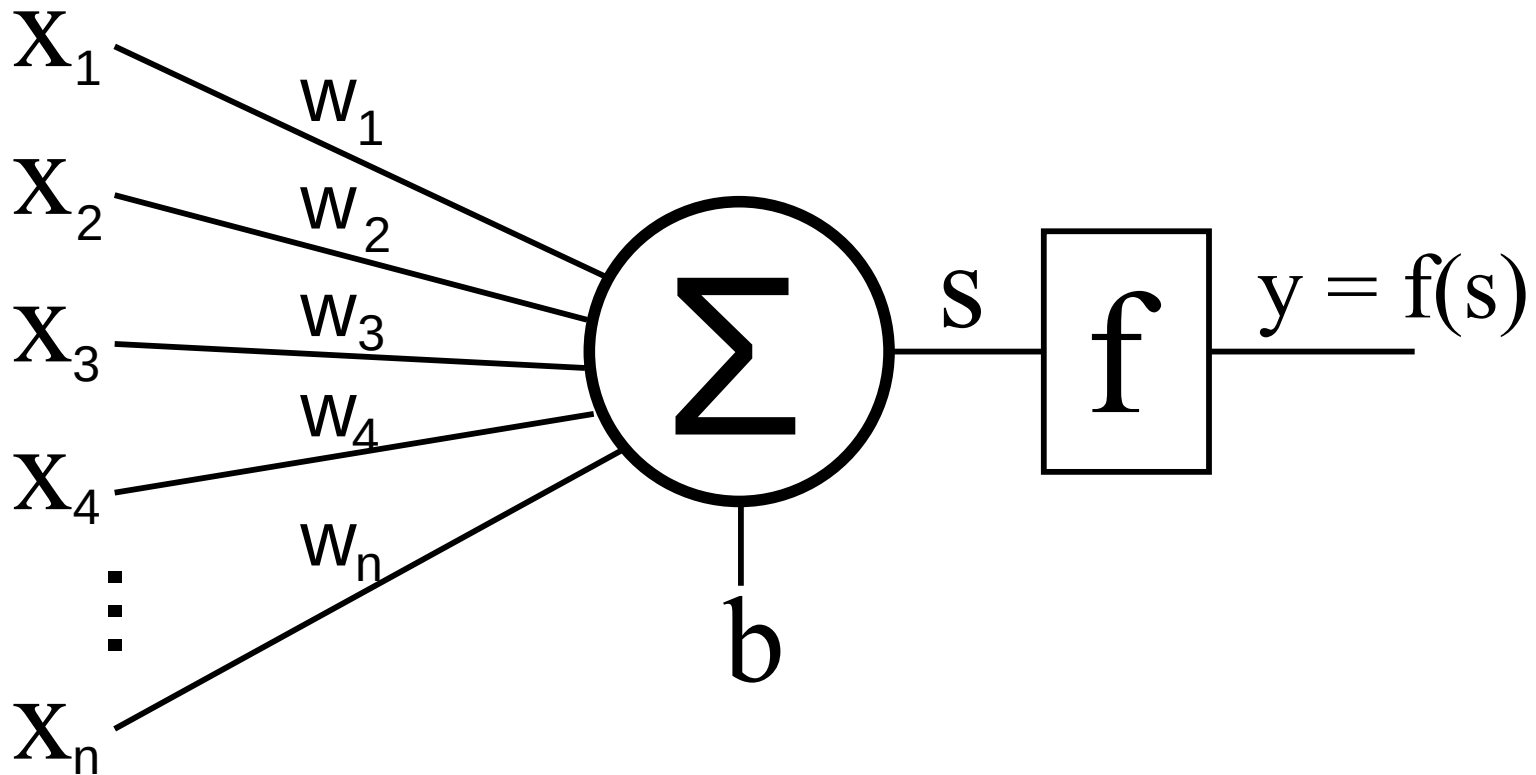
- $y = \frac{1}{1 + e^{-x}}$

(note : la fonction tanh est aussi utilisée)



- Fonctions plus complexe
- Toute grande valeur donne un résultat proche de 1 (ou de 0 en négatif)
- Le gradient tend vers 0 pour les grandes valeurs
 - Stabilisation des poids (variations de plus en plus faible, mais toujours existantes)
 - Mais « perte » du gradient (problématique avec un grand nombre de couches)
- Autre avantage que nous verrons plus tard

Introduction à l'Intelligence Artificielle



Passons à la pratique !