

# Comment une machine peut-elle apprendre à décider ?

## Découverte des réseaux de neurones artificiels

**Mise en situation.** Reconnaissance faciale, filtres anti-spam, recommandations de vidéos, voitures autonomes... Beaucoup de ces intelligences artificielles (IA) reposent sur des **réseaux de neurones artificiels**. Contrairement à ce que leur nom suggère, ce ne sont pas de véritables neurones biologiques, mais de simples **calculs mathématiques** inspirés (de très loin) du cerveau. Dans cette activité, vous allez découvrir comment un neurone artificiel calcule, comment on l'entraîne, et pourquoi assembler plusieurs neurones permet de résoudre des problèmes complexes.

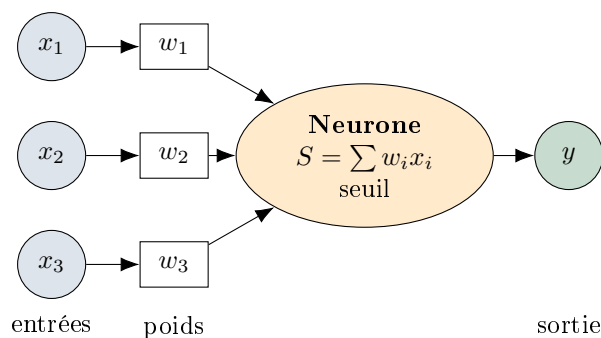
### 1. Qu'est-ce qu'un neurone artificiel ?

#### Définition

Un **neurone artificiel** reçoit une ou plusieurs **entrées** (des nombres), qu'il combine à l'aide de **poids** (des coefficients qui représentent l'importance donnée à chaque entrée). Il calcule ainsi une **combinaison linéaire** des entrées, puis renvoie un résultat en **sortie**.

Si les entrées sont  $x_1, x_2, \dots, x_n$  et les poids associés  $w_1, w_2, \dots, w_n$ , le neurone calcule :

$$S = w_1 \times x_1 + w_2 \times x_2 + \dots + w_n \times x_n$$



#### Remarque

Le neurone ne se contente pas de calculer  $S$  : il compare ensuite cette valeur à un **seuil** pour décider de sa réponse finale (par exemple « oui »/« non », ou une catégorie parmi plusieurs). Cette étape s'appelle la **fonction d'activation**. C'est elle qui permet au réseau de prendre de vraies décisions, et pas seulement de faire des calculs linéaires.

### 2. Activité 1 – Le neurone « thé ou café »

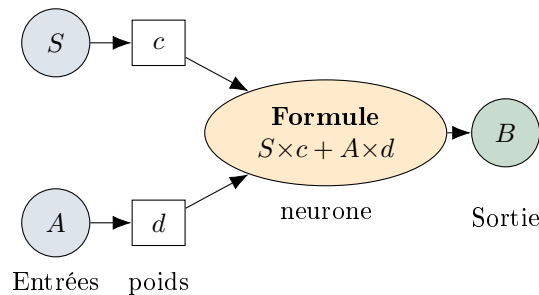
On s'intéresse à la préférence de consommation en matière de thé ou café. On code ainsi les données : la lettre  $A$  correspond à l'âge de la personne,  $S$  au sexe et  $B$  à la boisson :

| Sexe | Femme | Homme |
|------|-------|-------|
| $S$  | 1     | 2     |

| Boisson | Café | Thé |
|---------|------|-----|
| $B$     | 10   | 20  |

De plus, on fait intervenir deux coefficients  $c$  et  $d$  qui représentent l'importance attribuée au sexe ou à l'âge : c'est notre neurone, avec la formule  $S \times c + A \times d = B$ .

On dispose de deux données d'apprentissage : Madame T., 20 ans, préfère le thé, et Monsieur V., 50 ans, préfère le café.



### Remarque

Dans un vrai réseau de neurones, personne ne calcule les poids « à la main » : un **algorithme d'apprentissage** part de poids choisis au hasard, mesure l'erreur commise sur les données d'apprentissage, puis corrige les poids petit à petit pour réduire cette erreur, des milliers de fois de suite. Ici, on va simplement **terminer le travail de l'algorithme** : il a déjà trouvé le poids de l'âge,  $d = -3$ , et c'est à vous de trouver le poids du sexe,  $c$ .

### ► À vous de jouer !

#### Étape 1 : écrire les égalités d'apprentissage.

1. En remplaçant les données de Madame T. dans la formule «  $S \times c + A \times d = B$  », écrire une égalité liant  $c$  et  $d$ . On l'appelle égalité (1).
2. Faire de même avec les données de Monsieur V. pour obtenir l'égalité (2).

#### Étape 2 : trouver le poids manquant.

L'algorithme d'apprentissage a déjà trouvé  $d = -3$ .

3. Remplacer  $d$  par  $-3$  dans l'égalité (1), et montrer que l'égalité devient :  $c - 60 = 20$ .
4. En déduire la valeur de  $c$ .
5. **Vérifier** que les valeurs de  $c$  et  $d$  conviennent aussi pour Monsieur V., c'est-à-dire que l'égalité (2) est bien vraie. C'est essentiel : les poids du neurone doivent fonctionner pour **toutes** les données d'apprentissage, pas seulement pour la première !

#### Étape 3 : utiliser le neurone sur une nouvelle personne.

6. À l'aide de la formule et des valeurs de  $c$  et  $d$  trouvées, calculer  $B$  pour un homme de 46 ans.
7. La personne préfère le café si  $B$  est plus proche de 10, ou le thé si  $B$  est plus proche de 20. Quelle est la boisson préférée de cet homme de 46 ans ?
8. Que peut-on critiquer dans cette modélisation (nombre de données, choix des nombres 1/2, 10/20...) ?

### Coup de pouce

Pour Madame T. :  $S = 1$  (femme),  $A = 20$  et  $B = 20$  (thé). L'égalité (1) s'écrit donc  $1 \times c + 20 \times d = 20$ . À l'étape 2, souvenez-vous que  $20 \times (-3) = -60$ .

### 3. Compléter le programme Python

#### Exemple

Voici un programme qui calcule la valeur  $B$  à partir de  $S$ ,  $A$ ,  $c$  et  $d$  :

```
def calcule_B(S, A, c, d):
    B = S * c + A * d
    return B

print(calcule_B(2, 46, 80, -3))
```

#### ► À vous de jouer !

On souhaite écrire une fonction `preference(S, A)` qui renvoie la boisson préférée d'une personne, à partir de son sexe  $S$  et de son âge  $A$ . Elle doit :

- calculer  $B$  à l'aide de la formule (avec les valeurs de  $c$  et  $d$  trouvées en activité 1) ;
- renvoyer "café" si  $B$  est plus proche de 10, ou "thé" si  $B$  est plus proche de 20.

**Compléter** le programme ci-dessous (chaque ... doit être remplacé) :

```
def preference(S, A):
    B = S * ... + A * ...
    distance_cafe = abs(B - ...)
    distance_the = abs(B - ...)
    if distance_cafe < distance_the:
        return ...
    else:
        return ...

print(preference(2, 46))
```

#### Coup de pouce

Les valeurs de  $c$  et  $d$  sont celles trouvées dans l'activité 1. La distance entre deux nombres  $x$  et  $y$  s'obtient avec `abs(x - y)` : plus cette distance est petite, plus  $x$  est proche de  $y$ .

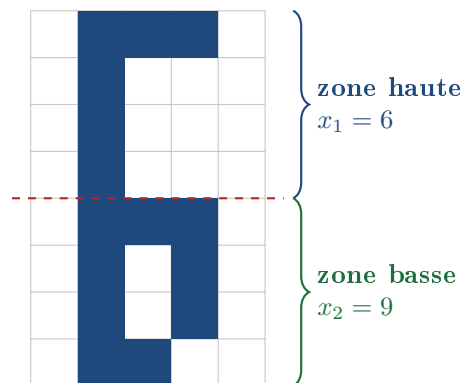
### 4. Activité 2 – Un neurone qui reconnaît des chiffres

#### Remarque

Cette activité s'inspire de l'atelier « *Connecte tes neurones* », conçu par **Olivier Druet** et **Nina Gasking** pour la Maison des Mathématiques et de l'Informatique. Dans l'atelier original, les élèves incarnent en groupe les neurones d'un réseau à plusieurs couches ; ici, vous allez effectuer individuellement les mêmes calculs qu'un neurone, pour comprendre comment il « voit » une image.

On souhaite construire un neurone capable de distinguer un chiffre « 6 » d'un chiffre « 9 » écrit à la main. Pour cela, on simplifie l'image en comptant le nombre de pixels (carreaux noircis) dans deux zones : la zone **haute** de l'image ( $x_1$ ) et la zone **basse** de l'image ( $x_2$ ).

## Chiffre « 6 » (pixellisé)



On donne au neurone les poids  $w_1 = 1$  et  $w_2 = -1$  (la zone basse compte donc « négativement ») et un seuil de  $S_0 = 0$ . Le neurone calcule  $S = w_1x_1 + w_2x_2$  et répond « 9 » si  $S > S_0$ , sinon il répond « 6 ».

## Exemple

Pour l'image ci-dessus, on compte  $x_1 = 6$  pixels en zone haute et  $x_2 = 9$  pixels en zone basse.

$$S = 1 \times 6 + (-1) \times 9 = 6 - 9 = -3$$

Comme  $-3 \leq 0$ , le neurone répond « 6 » : c'est correct !

## ► À vous de jouer !

On soumet au neurone les trois images ci-dessous, pixellisées sur la même grille :

Image A

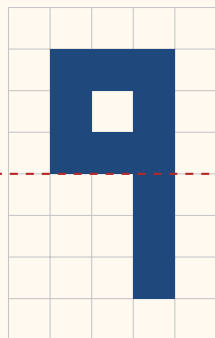


Image B

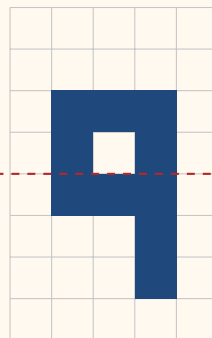
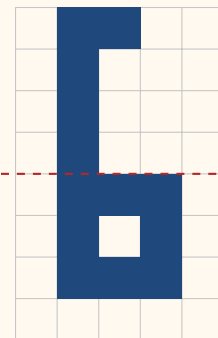


Image C



1. Compter, pour chaque image, le nombre de pixels en zone haute ( $x_1$ ) et en zone basse ( $x_2$ ), puis compléter les deux premières colonnes du tableau ci-dessous.
2. Calculer  $S = x_1 - x_2$  pour chaque image, puis en déduire la réponse du neurone (« 6 » ou « 9 »).

| Image   | $x_1$ (haut) | $x_2$ (bas) | $S = x_1 - x_2$ | Réponse du neurone |
|---------|--------------|-------------|-----------------|--------------------|
| Image A |              |             |                 |                    |
| Image B |              |             |                 |                    |
| Image C |              |             |                 |                    |

3. L'image B représente en réalité un « 9 »... mais écrit un peu trop bas dans la grille. Le neurone se trompe-t-il sur cette image ? Ce cas est-il inquiétant pour la fiabilité du modèle ? Justifiez.

- On souhaite modifier le seuil  $S_0$  pour que les **quatre** images de cette page (le « 6 » de l'exemple et les images A, B, C) soient toutes correctement reconnues. Proposer une valeur de  $S_0$  qui convient. Y a-t-il une seule valeur possible ?
- À votre avis, pourquoi utilise-t-on ici seulement 2 valeurs ( $x_1, x_2$ ) plutôt que les 40 pixels de la grille complète ? Quel est l'avantage pour un neurone ?

### Remarque

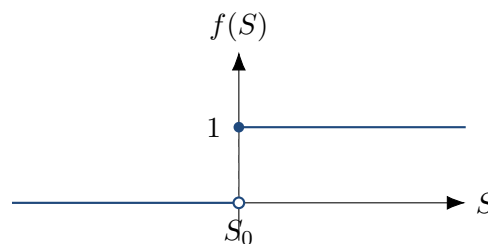
Un vrai réseau de neurones de reconnaissance de chiffres (comme celui utilisé par les banques pour lire les chèques) utilise directement les pixels de l'image et plusieurs couches de neurones, mais le principe reste le même : combiner des entrées avec des poids, puis appliquer un seuil.

## 5. Pourquoi le seuil est-il indispensable ?

### Définition

La fonction qui transforme la somme pondérée  $S$  en une décision finale s'appelle la **fonction d'activation** (ou fonction seuil). La plus simple est la fonction **seuil** (ou *fonction de Heaviside*) :

$$f(S) = \begin{cases} 1 & \text{si } S \geq S_0 \\ 0 & \text{si } S < S_0 \end{cases}$$



### Remarque

**À quoi sert le seuil ?** Sans seuil, un neurone ne sait faire qu'une chose : multiplier ses entrées par des poids, puis additionner. Or, si on branche plusieurs neurones de ce type les uns à la suite des autres, l'ensemble revient encore à « multiplier puis additionner » : un seul neurone aurait suffi ! Par exemple, si un premier neurone calcule  $2 \times x$  et qu'un second multiplie ce résultat par 3, les deux neurones réunis calculent simplement  $6 \times x$  : deux neurones pour rien... C'est le seuil, avec son passage brutal de 0 à 1, qui casse ce calcul trop simple. Grâce à lui, ajouter des neurones apporte réellement quelque chose, et le réseau devient capable de résoudre des problèmes compliqués (reconnaître une image, traduire une phrase, etc.).

### ► À vous de jouer !

On considère un neurone à deux entrées de poids  $w_1 = 2$  et  $w_2 = -3$ , avec un seuil  $S_0 = 1$ .

- Calculer  $S$  et la réponse  $f(S)$  pour les entrées  $(x_1, x_2) = (4, 1)$ .
- Calculer  $S$  et la réponse  $f(S)$  pour les entrées  $(x_1, x_2) = (2, 3)$ .
- Trouver un couple  $(x_1, x_2)$  pour lequel  $S$  est **exactement** égal au seuil  $S_0$ . Que répond alors le neurone ?

## ★ Question étoilée

Un **théorème d'approximation universelle** affirme qu'un réseau de neurones à plusieurs couches, muni d'une fonction d'activation non linéaire, peut approcher *n'importe quelle* fonction, aussi compliquée soit-elle, à condition d'avoir suffisamment de neurones. Cela explique pourquoi les réseaux de neurones sont utilisés dans des domaines aussi variés que la traduction automatique, la génération d'images ou la conduite autonome. Recherchez, en une phrase, un exemple d'application des réseaux de neurones qui n'a pas encore été cité dans cette activité.

## 6. Pour aller plus loin : assembler plusieurs neurones

## Remarque

Un seul neurone ne peut prendre que des décisions très simples (il sépare les données par une seule droite, ou un seul plan). En **assemblant plusieurs neurones en couches** (couche d'entrée, couches cachées, couche de sortie), un réseau devient capable de reconnaître des formes bien plus complexes : c'est le principe utilisé, par exemple, pour reconnaître n'importe quel chiffre manuscrit de 0 à 9, ou identifier un visage dans une photo. Ce fonctionnement en couches est détaillé dans l'atelier « *Connecte tes neurones* » d'Olivier Druet et Nina Gasking, où les élèves incarnent physiquement les neurones d'un petit réseau à trois couches.

## Définition

**Comment un réseau apprend-il ses poids ?** Les poids ( $c$ ,  $d$ ,  $w_1$ ,  $w_2...$ ) ne sont pas choisis à la main : ils sont ajustés automatiquement par un **algorithme d'apprentissage**, à partir de nombreux exemples déjà étiquetés (le **set d'entraînement**). On vérifie ensuite la qualité du réseau sur des exemples qu'il n'a jamais vus (le **set de test**). C'est le principe de l'**apprentissage supervisé**, illustré notamment par l'atelier « *Entrez dans la tête d'une IA* » de la Maison des Mathématiques et de l'Informatique.

## ► À vous de jouer !

1. Dans l'activité 1 (thé/café), quelles étaient les données d'apprentissage ? Combien y en avait-il ?
2. Pourquoi, à votre avis, un modèle entraîné avec seulement deux personnes (comme dans l'activité 1) risque-t-il de mal fonctionner sur de nombreuses autres personnes ?
3. Un réseau de neurones qui obtient 100% de bonnes réponses sur son set d'entraînement, mais seulement 60% sur son set de test, souffre d'un problème appelé **sur-apprentissage**. Formulez, en une phrase, une explication possible de ce phénomène.