

Découverte de l'intelligence artificielle



Notion d'intelligence artificielle

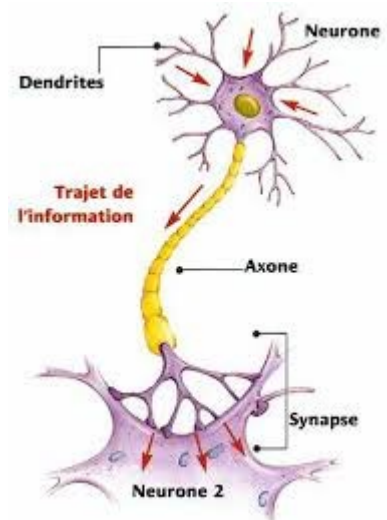
L'intelligence artificielle est l'ensemble des théories et techniques qui permettent de faire réaliser par une machine des tâches que l'homme accomplit en utilisant son intelligence.

Ce terme apparait en 1956.

1891 Waldeyer invente le mot « neurone »

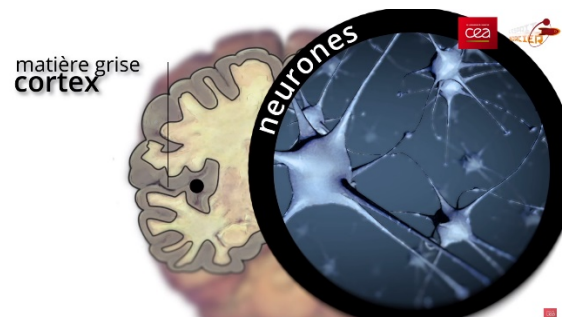
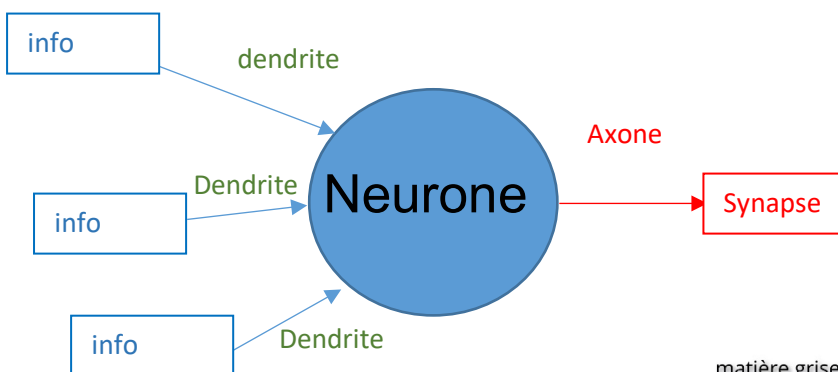
Synapse est un élément de liaison entre :

- Un neurone et un deuxième neurone
- Ou un neurone et une autre cellule



Un neurone reçoit des informations via ses dendrites et transmet une information via son axone à un autre neurone.

Voici un modèle biologique simplifié d'un neurone.



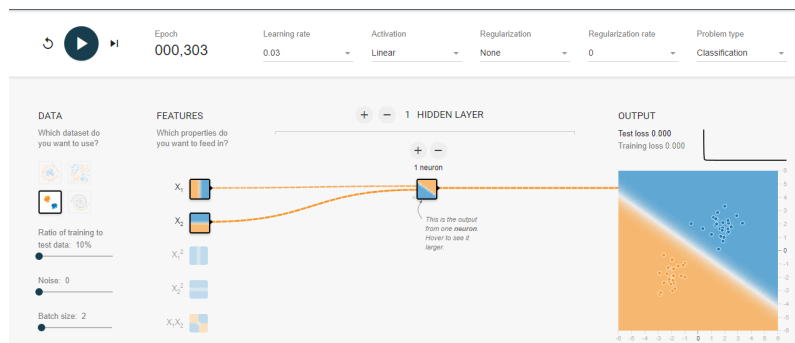
Application de l'intelligence artificielle

- Décision sur les marchés financiers
- Soutien au diagnostic médical
- Analyse et reconnaissance d'image
- Gestion à l'intérieur d'un jeu
- Fonction d'aide à la conduite automobile
- Siri



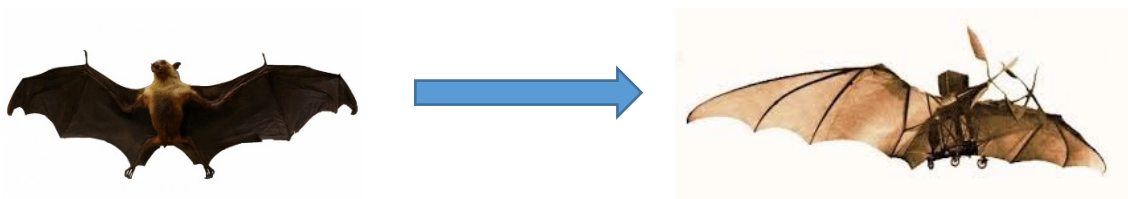
Modélisation

Sur ce site, tu peux simuler la phase d'apprentissage d'un neurone. On voit bien la zone de séparation entre les deux populations de couleurs différentes. Donc si on présente les caractéristiques x et y associées à un point, le neurone pourra alors nous dire si ce point fait partie de la population 1 ou 2.

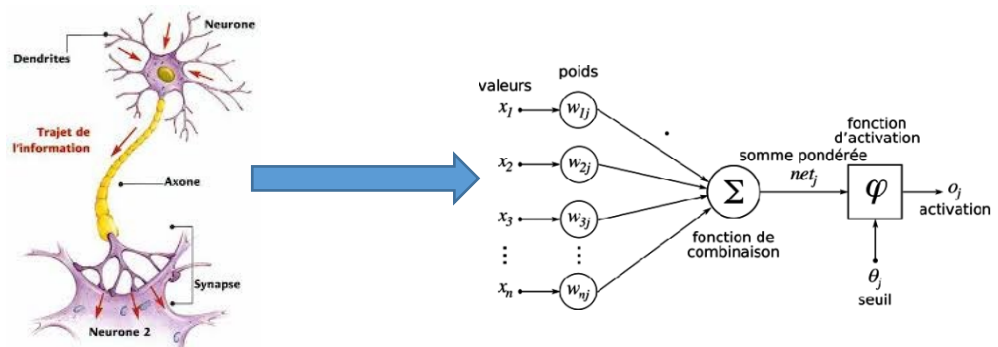


Modélisation mathématique d'un neurone

En aviation, les premiers inventeurs se sont basés sur le monde réel, comme les oiseaux. Le premier avion ressemblait à un oiseau.



Pour le neurone on essaie de faire la même chose, mais de façon mathématique.



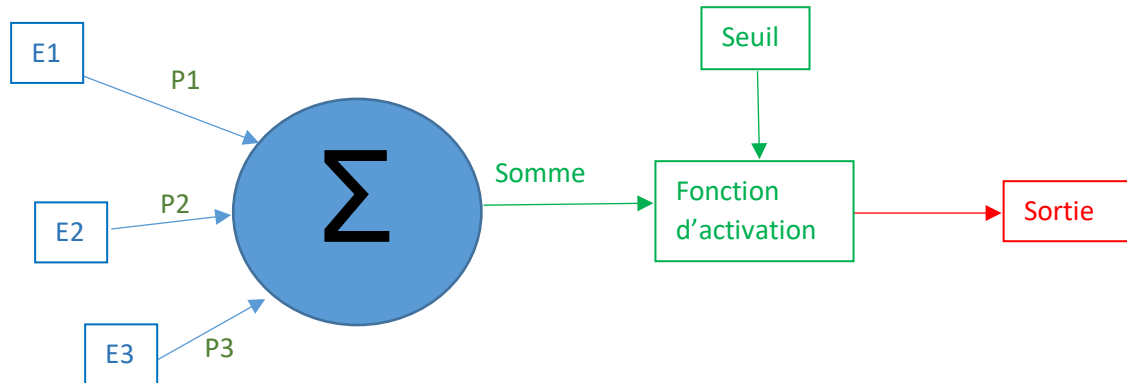
Modèle mathématique d'un neurone biologique.

Entrée E1, E2, E3

Poids synaptiques P1, P2, P3

Seuil de la fonction d'activation du neurone.

Sortie du neurone



$$\text{Somme} = E1 \times P1 + E2 \times P2 + E3 \times P3$$

Somme s'appelle la fonction de combinaison du neurone

La fonction d'activation du neurone renverra 0 ou 1 si la somme est en dessous ou au-dessus du seuil.

Problématique

Pour comprendre ce modèle on souhaite créer un neurone qui permettrait de reconnaître une image (noir et blanc) composée de 3 pixels spécifiques parmi plusieurs images différentes. Dans le cas présent, il y a $2^3 = 8$ possibilités avec 3 pixels



qui correspond alors à $e1=1, e2=1, e3=0$



qui correspond alors à $e1=0, e2=1, e3=0$



qui correspond alors à $e1=0, e2=1, e3=1$

Parmi toutes ces possibilités on souhaite rechercher l'image suivante automatiquement :



qui correspond alors à $e1=0, e2=0, e3=1$

A chaque pixel est alors associé une entrée du neurone ($e1, e2, e3$). A chaque entrée est associée un poids synaptique. Puis on fixe un seuil.

Programmation d'un neurone en Python

1. Création de la classe neurone

Création d'une classe d'objet (famille ou modèle d'objets types).

Nous allons créer une classe appelée Neurone

La classe va être associée à 4 attributs (variables dans l'objet). Un attribut est un paramètre d'un objet.

Les poids de chaque synapse (p_1 , p_2 , p_3) ainsi que le seuil de décision (seuil) du neurone.

Les poids synaptiques seront par exemple définis dans la classe, alors que l'attribut seuil pourra être passé en paramètre d'entrée à l'objet neurone lors de son instanciation comme ceci :

`neurone1 = Neurone(0,7)` ce qui permettra d'instancier l'objet neurone avec un seuil de 0.7 modifiable à chaque instanciation d'un objet neurone.

2. Création de la méthode `somme_ponderee()`

Dans cette classe Neurone, tu vas créer une méthode qui va permettre de calculer la fonction de combinaison du neurone (somme pondérée).

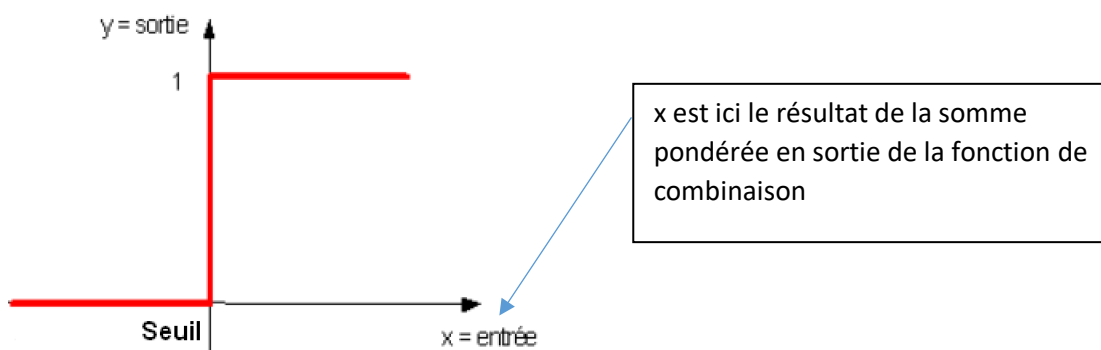
$$S = E1 \times P1 + E2 \times P2 + E3 \times P3$$

Cette méthode aura comme paramètres d'entrée e_1 , e_2 , e_3 . Attention dans cette méthode pour avoir accès au attribut de poids de la classe, il faudra taper par exemple `self.p1` qui permet de dire que l'on veut l'attribut p_1 de l'objet en cours (self) !

3. Création de la méthode `f_activation_neurone()`

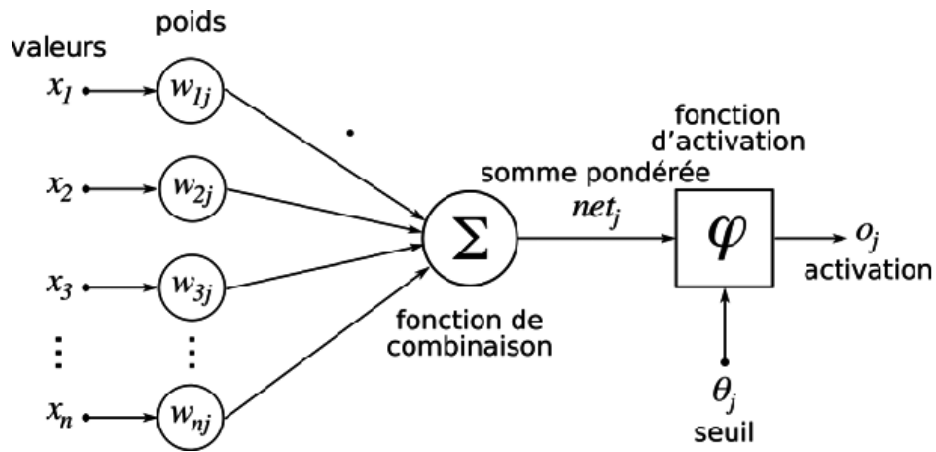
Tu vas créer une deuxième méthode dans la classe neurone qui va correspondre à la fonction d'activation du neurone qui renverra son information de sortie y .

Ici la fonction d'activation est simple et va correspondre au schéma suivant



Si le résultat de la somme pondérée est supérieur au seuil, le neurone est actif, la sortie de la fonction d'activation sera alors égale à 1, sinon la sortie sera égale à 0

Au final, on obtiendra alors bien le schéma voulu suivant :



4. Instanciation d'un neurone

Créer un objet (instance) neurone1 avec un seuil de 0.7

5. Poids synaptiques

Fixer les nouveaux poids synaptiques à $p1=-3$, $p2=-3$, $p3=1$

6. Affichage résultat

Afficher le résultat de la fonction d'activation pour différentes informations d'entrée et vérifier que le neurone fonctionne bien.

Seul le neurone1(0,0,1) renverra 1, toutes les autres combinaisons renverront 0