

Lab Réseaux de Neurones (Part 1)

Construction d'un neurone à seuil

1. En python, construire la classe **neurone_a_seuil** contenant les trois attributs suivants :

Biais, poids et *fct_passage* (la fonction de calcul de la sortie en fonction de l'entrée)*.

Le constructeur de cette classe (**__init__**) prendra comme paramètre le nombre d'entrées que possède le neurone (entier).

* : La fonction **fct_passage** sera définie plus tard. Sa signature est **fct_passage(vect1,vect2)**

2. Ajoutez la méthode **Set_Weights** à cette classe qui prend en paramètre les nouveaux biais (entier ou réel) et poids(liste d'entiers ou de réels) du neurone.

Ajoutez la méthode **Set_Function** à cette classe qui prend en paramètre une fonction et qui l'assigne à **fct_passage**.

3. Ajoutez la méthode **get_output** qui prend en paramètre une excitation (une liste d'entier ou de réels de longueur égale au nombre d'entrées de la neurone) et retourne la sortie calculée du neurone.
Construisez cette méthode en fonction de **fct_passage**.

Réseaux de neurones linéaires à seuil

4. Dans un réseau de neurones linéaires à seuil, la sortie d'un neurone à n entrées est calculée de la manière suivante :

$$output = \begin{cases} 1 & \text{si } \sum_{i=0}^n w_i x_i \geq 0 \\ 0 & \text{sinon} \end{cases}$$

Avec,

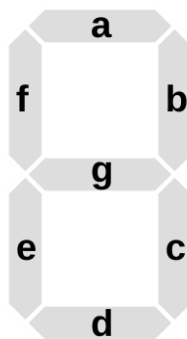
x_1, x_2, \dots, x_n les entrées et $x_0 = 1$.

w_1, w_2, \dots, w_n les poids et w_0 le biais.

5. Construisez la fonction **linear_output** qui prend en paramètre deux vecteurs (x_0, \dots, x_n) et (w_0, \dots, w_n) et calcule la sortie.
6. Instanciez un objet **neurone_a_seuil** avec 2 entrées. Changer son biais en 0.5, ses poids en [1,1] et affecter la fonction **linear_output** comme fonction de passage.
 Etudiez la sortie du neurone suite aux excitations suivante :
 $(0,0)$, $(0,1)$, $(1,0)$ et $(1,1)$.
 Quel fonction ce neurone effectue-t-il ?

Exemple

Soit un afficheur 7 segment qui affiche les nombre de 0 à 9 :



Soient le vecteur $X = (x_a, x_b, \dots, x_g)$ correspondant aux variables décrivant l'état des segments (a,b,...,g) ($x_a = 0$ si le segment a est éteint ,1 sinon).

Exemple :

Lorsque le chiffre 1 est affiché, le vecteur X est égal à $(0,1,1,0,0,0,0)$.

1. Construire un neurone à seuil à 7 entrées et initialisez ses poids à (1,1,1,1,1,1,1) et son biais à 1. Changez également sa fonction de passage à la fonction **linear_output**.
2. Construisez l'échantillon **S**, la liste de tous les cas états possibles de l'afficheur (S est alors un échantillon complet car il couvre tout les cas possibles)
3. On souhaite que le neurone construit remplisse la fonction de parité :

$$f(x) = \begin{cases} 1 & \text{si } x \text{ est impair} \\ 0 & \text{si } x \text{ est pair} \end{cases}$$

En utilisant l'**algorithme par correction d'erreurs** énoncé à la fin du document, modifier les poids du neurone afin que, selon ses entrées il retourne 1 ou 0 selon la parité du chiffre affiché sur le segment.
4. Tester les sorties du neurone afin de valider que ses poids ont été correctement choisis.

L'algorithme par correction d'erreurs

Pour X dans S faire

C = la sortie désirée du neurone avec X comme entrée

O = la sortie du neurone avec X comme entrée

Pour i de 0 à n faire

$$w_i = w_i + (c - o) * x_i$$

FinPour

FinPour

Bon travail