

# I Introduction du problème

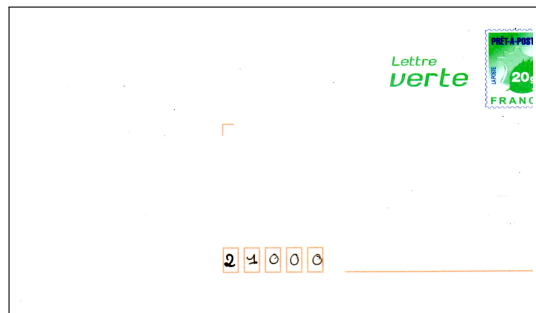
## A Position du problème

Comment automatiser la reconnaissance d'un code postal ?  
Comment une machine peut-elle apprendre ?

## B Restrictions du problème

- les enveloppes ont un format précis
- on possède un algorithme qui sait récupérer les chiffres sur l'enveloppe
- chaque image à un format de 28x28 pixels
- une image correspond à un chiffre
- elles sont traitées en niveaux de gris
- les images sont sauvegardées sous forme de vecteurs de taille 784

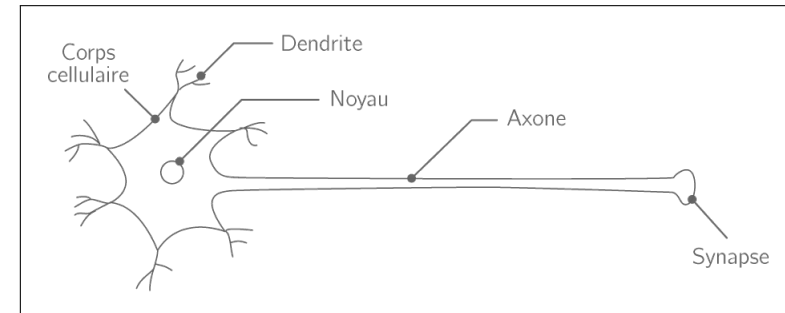
FIGURE 1 – Image d'une enveloppe



# II Approche Expérimentale

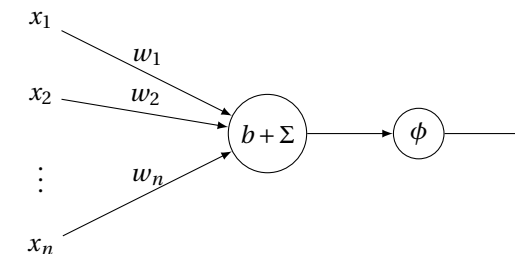
## A Analogie avec le Cerveau Humain

FIGURE 2 – Schéma d'un neurone biologique



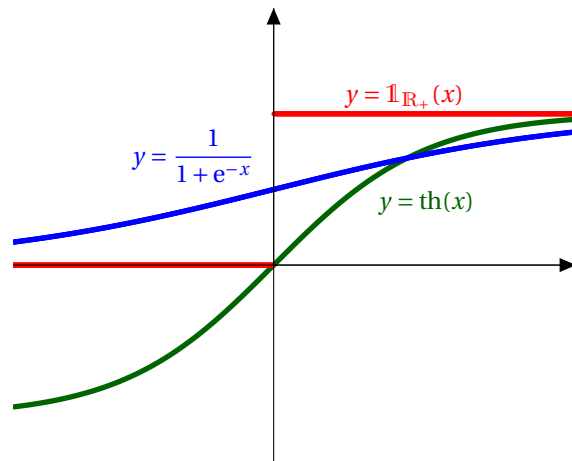
## B Neurone formel

FIGURE 3 – Représentation graphique d'un neurone formel



**Neurone :**  $\text{neur}(E, b, \vec{w}, \phi)(\vec{x}) = \phi(b + (\vec{x} | \vec{w}))$   
où  $E \in \mathcal{P}(\mathbb{R}) \setminus \{\emptyset\}$ ,  $\phi \in \mathbb{R}^{\mathbb{R}}$ ,  $b \in \mathbb{R}$  et  $\vec{w} = (w_1, \dots, w_n) \in \mathbb{R}^n$

FIGURE 4 – Représentation graphique des fonctions d'activation usuelle



### ③ Perceptron et séparabilité linéaire

FIGURE 5 – Représentation du OR

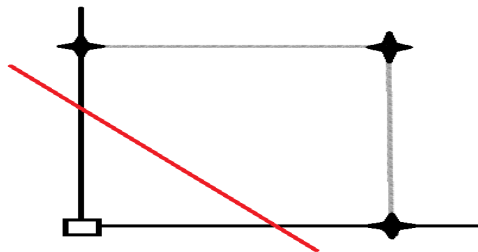
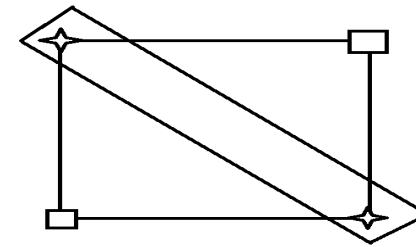
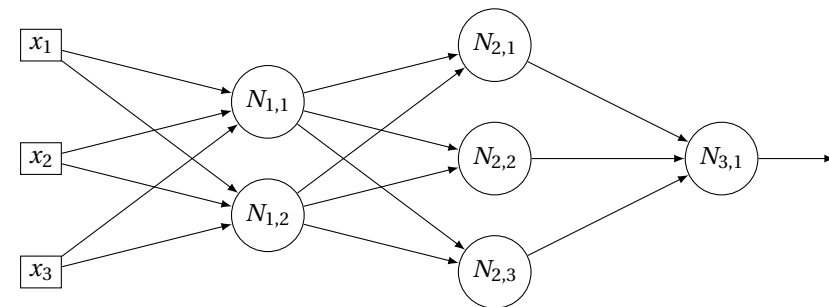


FIGURE 6 – Représentation du XOR



### ④ Réseau de neurones

FIGURE 7 – Représentation graphique d'un cerveau formel



### III Apprentissage des réseaux de neurones

#### A Lexique

Soit  $\mathcal{C}_P : E^n \rightarrow \mathbb{R}^p$  un cerveau formel.

- **Modèle** :  $M : B$  (base de donnée finie)  $\rightarrow \mathbb{R}^p$ .
- **Fonction de coût** :  $\delta : P$  (paramètres du cerveau)  $\mapsto \frac{1}{2} \sum_{\tilde{x} \in B} \|\mathcal{C}_P(\tilde{x}) - M(\tilde{x})\|_2^2$

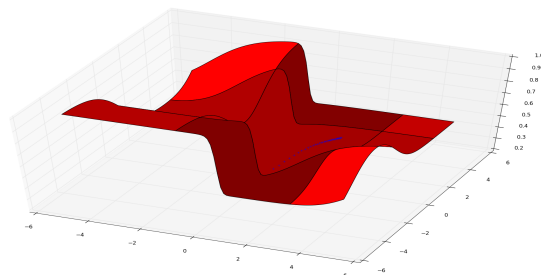
#### B Algorithme de WIDROW-HOFF

Pour l'apprentissage des perceptrons.

#### C Algorithme de descente du gradient

Pour l'apprentissage des réseaux multicouches. Cas particulier de la fonction d'activation  $f : x \mapsto \frac{1}{1 + e^{-x}}$ .

FIGURE 8 – Descente du gradient



#### Remarque

Au mieux on atteint un minimum local, le nombre d'itérations de l'algorithme peut devenir très important.

#### D Obstacles à l'apprentissage

- Pertinence du modèle, taille et obtention du modèle
- Sur-apprentissage
- Dés-apprentissage
- choix du coefficient de pondération de l'erreur.

Parti pris : étude empirique et résolution pratique, non-théorique.

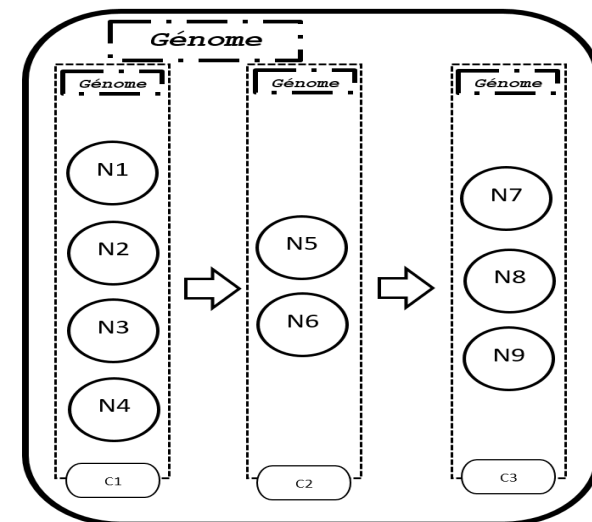
### IV Implémentation en Python

Choix d'une programmation orientée objet : création de classes.

#### A Classes

Genome, Neurone, Couche, Cerveau

FIGURE 9 – Schéma d'un Cerveau formel



## B Algorithmes d'apprentissages

Implémentation des fonctions apprentissageWH et apprentissageGradient

## C Côté expérimental

Implémentation de fonctions pour aider à la manipulation (sauvegarde, animation pour voir l'apprentissage en direct de fonctions mathématiques, logiciel pour créer un Cerveau et le voir sous forme de schéma)

FIGURE 10 – Exemple logiciel

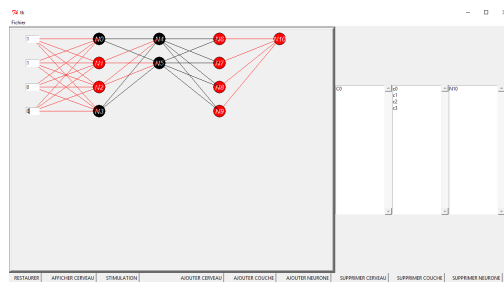
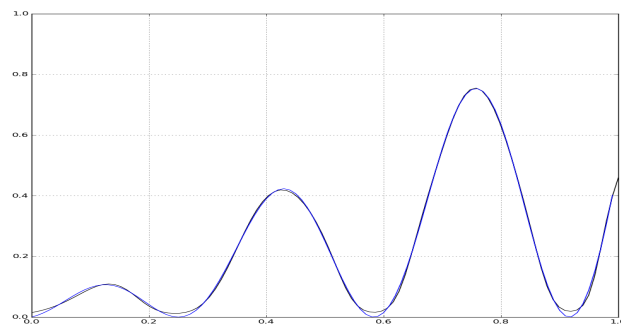


FIGURE 11 – Approximation de fonction



## V Lecture d'un code postal

### A Base de travail

Base de donnée : 50 000 images de chiffres trouvées sur internet. Image d'un chiffre :  $28^2$  pixels, contenant un nombre  $x \in [0 ; 1]$  représentant un dégradé de gris.

Modèle d'entraînement : 20 000 images.

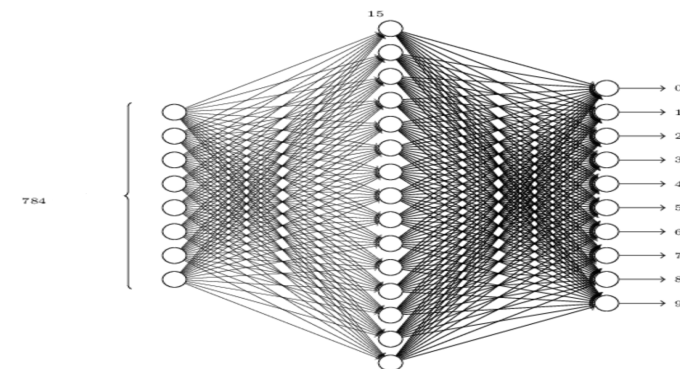
Modèle de vérification : 30 000 images

Réseau :  $\mathcal{R} = (C_1, C_2)$  avec  $28^2$  entrées, 15 neurones dans  $C_1$  et 10 dans  $C_2$ .

Sortie : 10-uplet.

**Exemple :** (1, 0, ..., 0, 0) correspond à 0. (0, 1, ..., 0, 0) correspond à 1.

FIGURE 12 – Réseau utilisé



### B Résultats finaux

On arrive au mieux à reconnaître 1 des 5 chiffres du code postal à 90%. On sait donc diriger 80% des lettres analysées vers le bon département.

On en conclue que ce type de réseau de neurone atteint rapidement ses limites. Néanmoins pour condamner ce type d'approximation il nous faudrait continuer l'investigation sur des structures plus "libres" (neurones récurrents, connexion d'un même neurone sur plusieurs couches).