

CAHIER DES CHARGES  
PROJET C++  
Implémentation d'un réseau de neurones

Amina El Bachari, Camille Goujet,  
Israa Ben Sassi, Rafael Quilbier et Mehdi Helal

# Table des matières

<b>1</b>	<b>Objectifs et fonctionnalités</b>	<b>3</b>
1.1	Création et manipulation d'un réseau de neurones . . . . .	3
1.2	Appliquer un réseau de neurones à un exemple de la liste suivante : . . . . .	3
1.2.1	Identification des objets . . . . .	3
1.2.2	La reconnaissance d'image . . . . .	3
1.2.3	Les classifications de texte ou image : . . . . .	4
1.2.4	Reconnaissance vocale . . . . .	4
1.2.5	Prédiction de données . . . . .	5
<b>2</b>	<b>Bibliographie</b>	<b>5</b>
2.1	Définitions importantes . . . . .	5
2.1.1	Neurone dit formel ou artificiel : . . . . .	5
2.1.2	Réseau de neurones : . . . . .	6
2.1.3	Couche : . . . . .	7
2.1.4	Fonction d'activation : . . . . .	7
2.2	Fonctionnement d'un réseau de neurones . . . . .	8
2.3	Architecture des réseaux de neurones . . . . .	9
2.3.1	Réseaux de neurones feed forwarded . . . . .	9
2.3.2	Réseaux de neurones récurrents . . . . .	10
2.3.3	Réseaux de neurones à résonance . . . . .	11
2.3.4	Réseaux de neurones auto-organisés . . . . .	11
<b>3</b>	<b>Implémentation</b>	<b>11</b>
3.1	Fonctionnement du système . . . . .	11
3.2	Manuel d'utilisation préliminaire . . . . .	12

# 1 Objectifs et fonctionnalités

Tout au long de ce projet, nos objectifs seront les suivants :

## 1.1 Création et manipulation d'un réseau de neurones

## 1.2 Appliquer un réseau de neurones à un exemple de la liste suivante :

### 1.2.1 Identification des objets

Les applications reposant sur des modèles de détection d'objets se sont beaucoup développées dernièrement : comptage d'objets pour l'analyse du trafic routier, détection des panneaux de signalisation pour la voiture autonome, détection de défauts sur infrastructure. . .

L'objectif ici n'est plus de classer une image, mais de détecter les objets au sein de celle-ci, en dessinant un rectangle (on parle de «bounding box») entourant le plus précisément les objets présents.

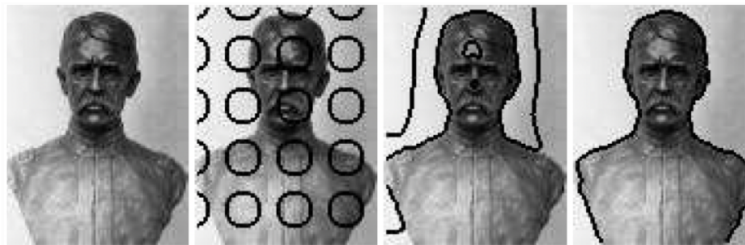


FIGURE 1 – Contours d'objet

### 1.2.2 La reconnaissance d'image

La reconnaissance d'image est de plus en plus utilisée de nos jours, que ce soit par exemple pour déverrouiller son téléphone, la sécurité et la surveillance, la géolocalisation visuelle ou la reconnaissance d'objets etc. . . Donc avec la quantité d'images qui s'accumulent sur internet, les scientifiques qui travaillaient dans le domaine de la vision par ordinateur ont saisi l'opportunité d'utiliser toutes ces bases de données pour créer des modèles de reconnaissance d'image.

Un modèle de réseau de neurones profond est capable aujourd'hui de reconnaître chaque élément d'une scène pourvu qu'il ait été entraîné pour cela. À partir de la sémantique de cette reconnaissance, un réseau de neurones profond peut même générer une légende à cette scène. Par exemple, aujourd'hui il est assez courant de pouvoir obtenir le résultat suivant automatiquement :

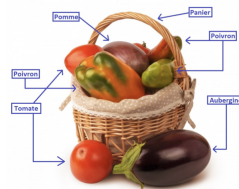


FIGURE 2 – Illustration avec des fruit et légumes

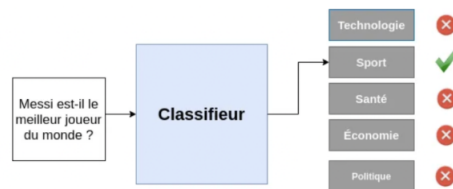


FIGURE 4 – Illustration avec le classifieur

### 1.2.3 Les classifications de texte ou image :

L'objectif est d'élaborer un système capable d'affecter à une image ou à un texte non-structuré, un tag qui correspond à une classe bien précise.

On cite deux exemples d'utilisation très importants.

D'une part, la classification de documents imprimés est une tâche cruciale dans de nombreuses chaînes de traitement ; par exemple, l'automatisation de tâches bureautiques afin de classer les documents imprimés selon des catégories telles que : lettres, publicités, plans et cartes, articles de presse, etc. . .

Il faut tout d'abord extraire les données textuelles ou les images utilisées pour ensuite les classer.

D'autre part, la reconnaissance des émotions, très utilisée sur les réseaux sociaux. Qu'il s'agisse de la détection de comportement violent à travers un texte ou une image (DeepBreath) ou de l'analyse de l'impact émotionnel sur des campagnes de marketing, etc. . .



FIGURE 3 – Illustration avec les visages

### 1.2.4 Reconnaissance vocale

Les réseaux de neurones sont également développés pour effectuer de la reconnaissance de signal vocal.

Pour faire simple, ils leur est possible de décomposer les audios en phrases puis en mots, aussi ils peuvent utiliser un modèle de langage pour calculer la probabilité d'une phrase donnée.

D'ailleurs, pour la saisie vocale depuis son clavier virtuel Gboard, Google a récemment déployé une nouvelle technologie entièrement pilotée par des réseaux de neurones récurrents *RNN* (que nous expliquerons dans la suite).

Elle prédit directement la sortie de caractères en fonction de l'entrée vocale de façon très précise.

D'autres recherches sur l'utilisation de réseaux de neurones dit «à décharges» sont menées. Selon une étude de Loïselle Stéphane de l'Université du Québec de Chicoutimi, ils devraient permettre d'effec-

tuer une reconnaissance vocale indépendamment du locuteur et sans avoir à réaliser une longue période d'apprentissage !

### **1.2.5 Prédiction de données**

Enfin les réseaux de neurones sont très utilisés pour mettre au point des modèles de prédictions à partir d'échantillons de données. La prédiction d'indicateurs financiers est un exemple très parlant. En effet « en 2011, 40 % des ordres donnés sur le CAC40 sont totalement automatisés à l'aide d'algorithmes informatiques ». Aujourd'hui, cet outil fournit d'excellents résultats dans le domaine de détection des entreprises en future difficulté.

Les réseaux de neurones peuvent aussi prédire des phénomènes qui nous semblent plus aléatoires et non linéaires tout en prenant en compte d'éventuelles imprécisions des données fournies en entrée. Une équipe de scientifiques s'est intéressée à la prévision des crues du bassin-versant de l'Eure à la station de Louviers en Normandie grâce à des réseaux de neurones. Le modèle pluie-débit en résultant a permis de prendre en compte l'imprécision des données fournies en entrée et ainsi d'établir des prévisions fiables en quelques secondes sur les grandes crues à venir dans les 48 heures. Cet exemple illustre parfaitement l'importance que peuvent avoir les réseaux de neurones dans la prédiction de phénomène non-linéaires.

## **2 Bibliographie**

### **2.1 Définitions importantes**

#### **2.1.1 Neurone dit formel ou artificiel :**

Le neurone formel est l'unité élémentaire des réseaux de neurones artificiels dans lesquels il est associé à ses semblables pour calculer des fonctions arbitrairement complexes, utilisées pour diverses applications en intelligence artificielle. Mathématiquement, le neurone formel est une fonction à plusieurs variables et à valeurs réelles.

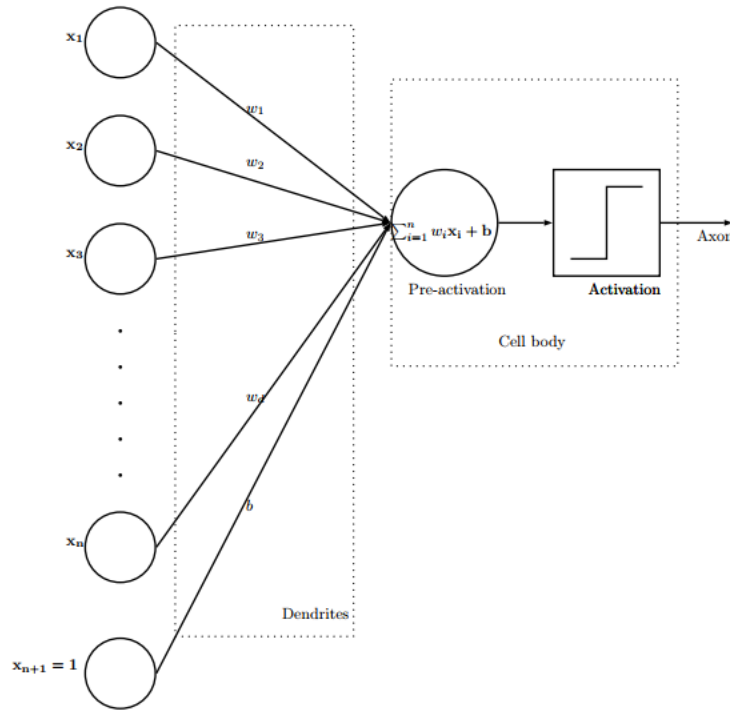


FIGURE 5 – Schématisation algorithmique d’un neurone artificiel

Cette représentation est appelée le perceptron, un algorithme d’apprentissage supervisé pour les classifications binaires linéaires. Cela fait beaucoup de mots inconnus alors arrêtons nous un instant sur chacun de ces termes car ils seront importants pour la suite !

Algorithme : le perceptron est une suite d’opérations et de calcul = la somme des entrées, leur pondération, la vérification d’une condition et la production d’un résultat d’activation.

Apprentissage : l’algorithme doit être “entraîné”, c’est à dire qu’en fonction d’une prédiction voulue, le poids des différentes entrées va évoluer et il faudra trouver une valeur optimale pour chacune.

Supervisé : l’algorithme trouve les valeurs optimales de ses poids à partir d’une base de données d’exemples dont on connaît déjà la prédiction. Par exemple on a une base de données de photos de banane et on “règle” notre algorithme jusqu’à ce que chaque photo (ou presque) soit classé comme banane.

Classification : l’algorithme permet de prédire une caractéristique en sortie et cette caractéristique sert à classer les différentes entrées entre elles. Par exemple, trouver toutes les bananes dans un panel de photos de fruits.

### 2.1.2 Réseau de neurones :

Un réseau de neurones est en général composé d’une succession de couches dont chacune prend ses entrées sur les sorties de la précédente. Chaque couche (i) est composée de  $N_i$  neurones, prenant leurs entrées sur les  $N_{i-1}$  neurones de la couche précédente.

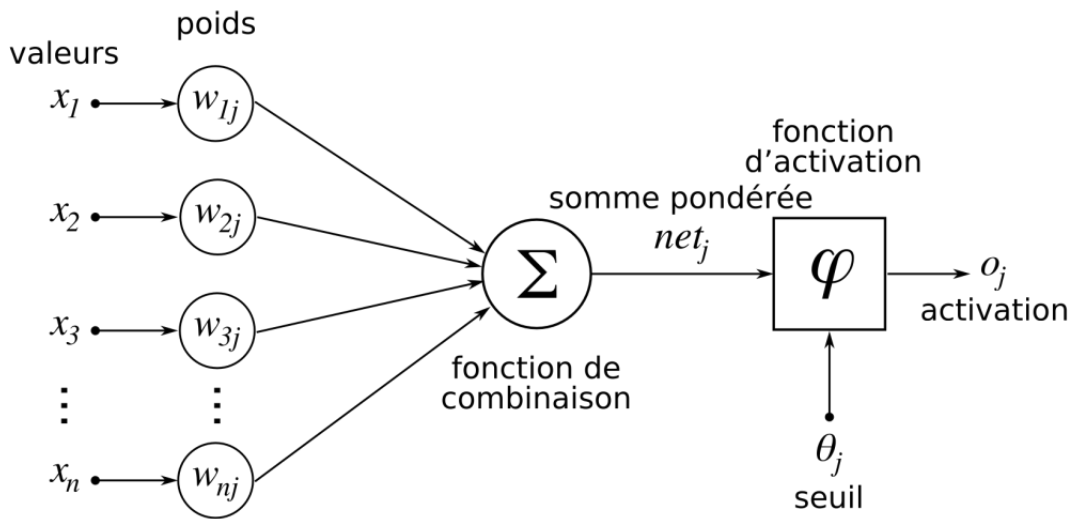


FIGURE 6 – Illustration d'un réseau de neurones simple

### 2.1.3 Couche :

En effet, un réseau de neurones est composé d'une couche d'entrées ("inputs layer"), d'une couche de sortie ("outputs layers") et d'au moins une couche cachée ("hidden layer") qui fait le lien entre entrée et sortie. Toutes ces couches sont composées de plusieurs neurones qui sont eux-mêmes reliés les uns aux autres par des poids.

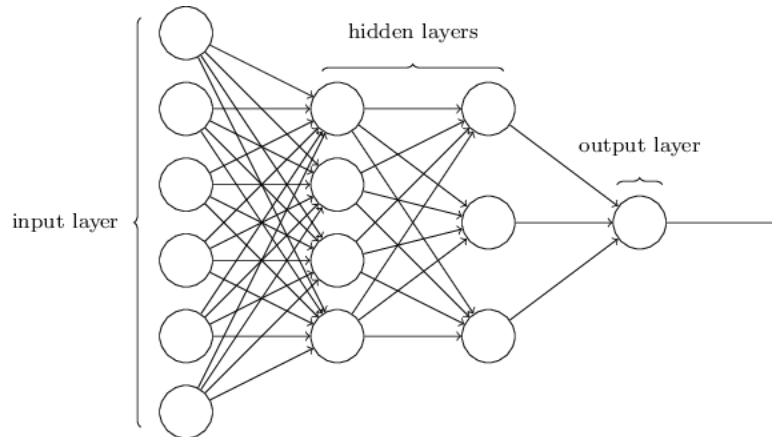


FIGURE 7 – Illustration des différentes couches

### 2.1.4 Fonction d'activation :

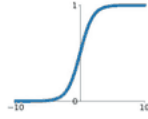
La fonction d'activation (ou fonction de seuillage, ou encore fonction de transfert) sert à introduire une non-linéarité dans le fonctionnement du neurone. Les fonctions de seuillage présentent généralement trois intervalles : en dessous du seuil, le neurone est non-actif (souvent dans ce cas, sa sortie vaut 0 ou -1) ; aux alentours du seuil, une phase de transition ; au-dessus du seuil, le neurone est actif (souvent dans ce cas, sa sortie vaut 1). Des exemples classiques de fonctions d'activation sont :

- La fonction sigmoïde.
- La fonction tangente hyperbolique.
- La fonction de Heaviside.

## Activation Functions

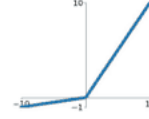
### Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



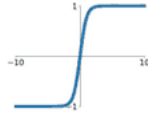
### Leaky ReLU

$$\max(0.1x, x)$$



### tanh

$$\tanh(x)$$

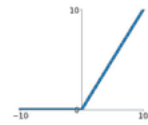


### Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

### ReLU

$$\max(0, x)$$



### ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

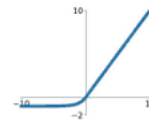


FIGURE 8 – Différentes fonctions d'activation

## 2.2 Fonctionnement d'un réseau de neurones

Pour comprendre son fonctionnement, étudions un modèle mathématique simple d'un neurone biologique : le modèle de McCulloch et Pitts (1943).

Considérons  $n$  entrées  $x_1, \dots, x_n \in \mathbb{R}$ . Un neurone fonctionne en 2 phases. Tout d'abord, il effectue la somme pondérée des entrées :

$$I = \sum_{i=1}^n \omega_i x_i$$

avec  $\omega_i \in \mathbb{R}$  le poids de la  $i^{\text{ème}}$  entrée.

Ensuite, la fonction d'activation  $f$  vérifie si la valeur calculée est supérieure au seuil requis et détermine si le neurone est actif ou non. Pour ce faire, on compare  $I$  à un seuil  $T$  : si  $I \geq T$  alors le neurone est actif et transmet le signal, sinon il est inactif. Dans le modèle de McCulloch et Pitts, on a :

$$f(I) = \begin{cases} 1 & \text{si } I \geq T \\ -1 & \text{sinon} \end{cases}$$

Notons que lorsque les neurones sont reliés uniquement par des connexions directes (i.e vers un neurone de la couche suivante), l'activation des différentes couches est réalisée de manière synchrone : la sortie de tous les neurones de la 1<sup>ère</sup> couche est calculée, puis celle de la 2<sup>e</sup>... Dans le cas d'un réseau possédant des connexions latérales (de haut en bas vers un neurone d'une même couche ou bien vers un neurone d'une couche précédente), l'ordre de mise à jour des sorties des différents neurones est important, car chaque nouvelle sortie va pouvoir modifier le calcul de la sortie d'un autre neurone. On peut mettre à jour des sorties de manière asynchrone et aléatoire (i.e les neurones sont choisis aléatoirement) ou bien de manière synchrone (toutes les sorties sont mises à jour en même temps).

Au fur et à mesure des itérations, les poids sont modifiés par apprentissage. Il les calcule en fonction des couples entrée/sortie désirées (on parle d'apprentissage supervisé) ou indépendamment d'une sortie désirée (on parle d'auto-organisation) par des petites adaptations successives.



Soit  $\omega_{ij}$  le poids de la connexion entre les neurones  $i$  et  $j$  à un instant donné discret  $t$ . Après une itération d'apprentissage, la nouvelle valeur est  $\omega_{ij}(t+1) = \omega_{ij}(t) + \Delta\omega_{ij}$ .

Il existe plusieurs règles d'apprentissage courantes comme la règle de Hebb, la règle d'apprentissage compétitif, etc que nous détaillerons dans notre projet lorsque nous les utiliserons.

## 2.3 Architecture des réseaux de neurones

On appelle architecture d'un réseau de neurones sa forme.

On distingue 4 types de réseaux de neurones :

- les réseaux de neurones Feed-forwarded
- les réseaux de neurones récurrents (RNN)
- les réseaux de neurones à résonance
- les réseaux de neurones auto-organisés.

Le choix de telle ou telle architecture est une question essentielle lors de la construction d'un réseau. Chacune possède ses forces et ses faiblesses. Ainsi, en fonction de l'application que l'on souhaite faire de notre algorithme, on se portera sur une architecture en particulier.

On peut aussi souligner que souvent, afin d'obtenir un meilleur résultat, plusieurs types d'architectures sont combinées. Nous avons préalablement cité 4 grandes familles d'architectures, mais il existe des dizaines de réseaux particuliers pour chacune d'elles. Voyons plus en détails ces 4 architectures :

### 2.3.1 Réseaux de neurones feed forwarded

L'appellation Feed-Forward signifie que l'information traverse le réseau de neurones de l'entrée à la sortie sans retour en arrière. En français, il est nommé : «réseau de neurones à propagation avant». Dans ce type de réseaux, on distingue les réseaux monocouches et les réseaux multicouches.

Dans les réseaux monocouches, le perceptron est dit «simple», car il est constitué que d'une couche d'entrée et une couche de sortie.

À l'inverse, dans les réseaux multicouches, les perceptrons sont qualifiés de multicouches, car ils disposent de plusieurs couches cachées entre l'entrée et la sortie.

Ces 2 types de réseaux de neurones ont des intérêts différents : le réseau multicouche est plus adapté pour traiter des informations non-linéaires.

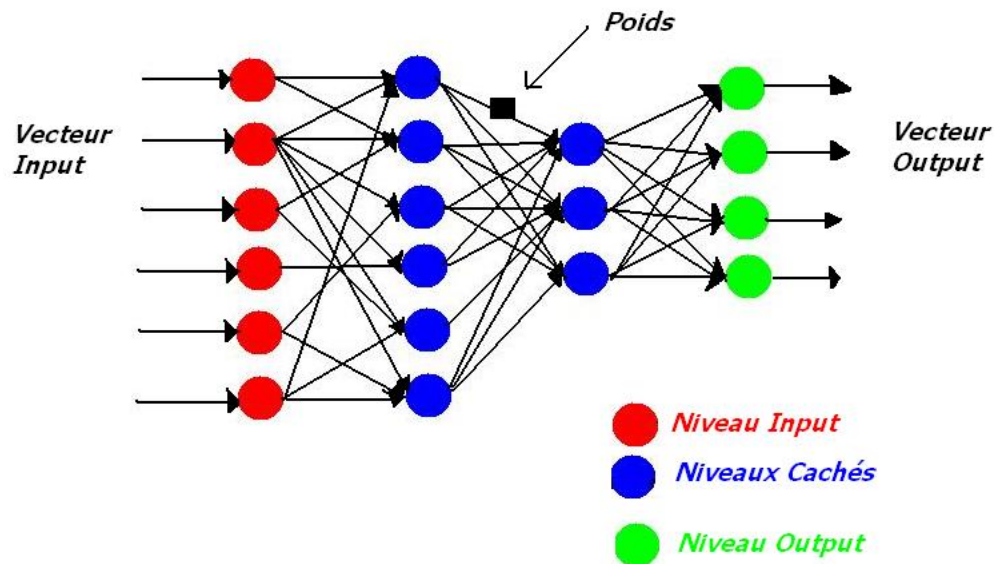


FIGURE 9 – Réseau Feed-Forwarded

### 2.3.2 Réseaux de neurones récurrents

Plus complexe et moins intuitif, les réseaux de neurones récurrents traitent les informations de manière circulaire. Un réseau est qualifié de récurrent si sa structure possède au moins un cycle. Contrairement l'architecture de type Feed Forwarded, dans les réseaux récurrents, l'information circule dans les deux sens. Ils peuvent posséder une couche ou plusieurs. L'intérêt est de conserver de l'information en mémoire et de la laisser accessible à tout instant ultérieur. C'est pourquoi les réseaux de neurones récurrents sont particulièrement bien adaptés aux applications faisant intervenir un contexte, comme la reconnaissance de forme.

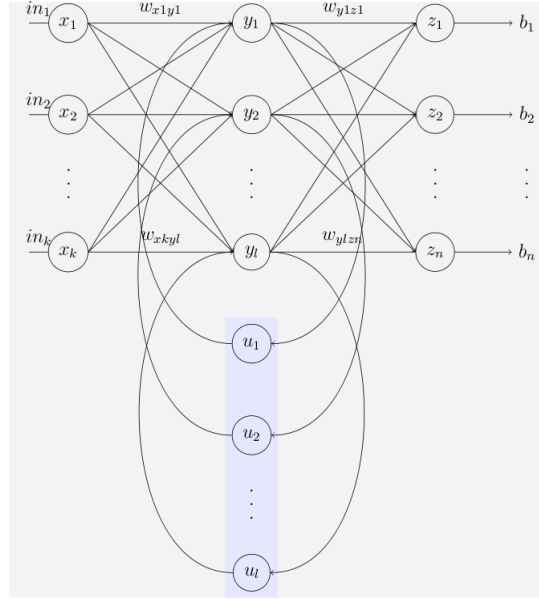


FIGURE 10 – Réseau récurrent de type Elman

### 2.3.3 Réseaux de neurones à résonance

L'appellation de ce type de réseaux fait référence à son fonctionnement. Lorsqu'un neurone est activé, son activation est renvoyée à tous les autres neurones du réseau. Cela provoque des oscillations, d'où l'emploi du terme «résonance».

Pour mieux comprendre l'objectif de cette architecture, étudions le modèle ART (Adaptative Resonance Theory) conçu par Gail Carpenter et Stephen Grossberg. Il existe de nombreux modèles ART, (ART1, ART2, fuzzy ART etc.) qui utilisent l'apprentissage supervisé, ou non supervisé. L'objectif général des modèles ART est de résoudre le dilemme entre stabilité et plasticité.

Selon un article du laboratoire d'Analyse Cognitive de l'information à Montréal : «la plasticité rend compte de la capacité du réseau à s'adapter aux informations nouvelles, et la stabilisation mesure la capacité du réseau à organiser les informations connues en ensembles stables.».

Finalement, le principe des modèles ART est d'apprendre de manière autonome, à s'adapter, et à se stabiliser en même temps. Ainsi, ces modèles sont capables de choisir entre une information pertinente à prendre en compte, et une information superflue, qui pourrait donner lieu à un surapprentissage.

### 2.3.4 Réseaux de neurones auto-organisés

Ce type de réseau de neurones est surtout utilisé dans le traitement d'informations spéciales. En effet grâce à des méthodes d'apprentissage non supervisé, ce type de réseau est capable de répartir en différentes classes de grands espaces de données.

## 3 Implémentation

### 3.1 Fonctionnement du système

La première étape consiste en la construction d'un réseau de neurones (construction des couches des perceptrons). Une fois les caractéristiques saisies par l'utilisateur, le programme doit construire un

réseau de neurones correspondant. Nous proposons le procédé suivant :

- construction des instances de la classe Neurone
- construction des instances de la classe Couche : on attribue les objets précédemment créés à une couche.
- construction des liaisons entre neurones. On pourra utiliser par exemple des matrices  $M$  avec  $M_{ij} = 1$  si le  $i$ -ème neurone est connecté au  $j$ -ième neurone et 0 sinon. L'information peut n'aller que dans un sens unique ; autrement dit, il est possible d'avoir  $(M)_{ij}=1$  et  $(M)_{ji}=0$  pour  $i, j$  fixé. On pourra éventuellement avoir une matrice par couche.
- attribution de poids aux liaisons : en notant  $m$  le nombre de liaisons, on pourra construire un vecteur de poids  $W$  de dimension  $m$ . On modifiera alors la matrice précédente en remplaçant les coefficients  $M_{ij} = 1$  par  $W_k$ .
- définition d'une fonction d'activation commune à tous les neurones
- regroupement des objets précédent dans une instance de classe Réseau

Une fois le réseau initial construit, il reste quelques étapes à effectuer afin que le réseau réalise son objectif :

- apprentissage du réseau : plusieurs méthodes pourront être définies suivant la fonction du réseau et/ou le type de réseau
- construction du vecteur des entrées : plusieurs méthodes pourront être définies suivant le type de données utilisé (image, signal...)
- construction du vecteur des sorties : utilisation d'une méthode qui prend en entrées le réseau à l'issu de l'apprentissage et le vecteur des entrées et qui retourne un vecteur des sorties.
- fournir une sortie à l'utilisateur : comme pour le vecteur des entrées, il y aura une étape d'interprétation du vecteur des sorties en fonction de la sortie attendue (image, signal, prédiction...)

### 3.2 Manuel d'utilisation préliminaire

- Choisir une application : l'utilisateur devra dans un premier temps choisir la fonctionnalité qui l'intéresse parmi une liste (voir la liste non exhaustive des applications précédentes).
- Choisir un type de réseau : cela permettra de comparer différents types de réseau sur une même application. Cela permettra aussi de simplifier la construction et la manipulation du réseau de neurones en utilisant l'encapsulation.
- Saisir les caractéristiques du réseau : l'utilisateur devra saisir le nombre de couches ainsi que le nombre de neurones par couches. De manière optionnelle, il pourra également choisir la fonction d'activation et les poids initiaux (de manière aléatoire, tous égaux, par ordre croissant...).
- Donner les entrées au programme : l'utilisateur indiquera le fichier sur lequel le réseau doit travailler. Le fichier doit correspondre à l'application choisie précédemment (par exemple, donner une image dans le cas d'une application en traitement d'image).

## Références

- [1] [http://benoit.decoux.free.fr/ENSEIGNEMENT/PROGRAMMATION/projet\\_RN\\_CPP.pdf](http://benoit.decoux.free.fr/ENSEIGNEMENT/PROGRAMMATION/projet_RN_CPP.pdf)
- [2] <https://www.juripredis.com/fr/blog/id-19-demystifier-le-machine-learning-partie-2-les-reseaux-de-neurones-artificiels>
- [3] <https://www.lebigdata.fr/perceptron-machine-learning>
- [4] <https://www.youtube.com/watch?v=sK9AbJ4P8ao>
- [5] <https://zestedesavoir.com/forums/sujet/6567/classe-generique-pour-des-reseaux-de-neurones-en-c/>

- [6] <https://openclassrooms.com/fr/courses/4470406-utilisez-des-modeles-supervises-non-lineaires/4730716-entraenez-un-reseau-de-neurones-simple>
- [7] <http://lexicometrica.univ-paris3.fr/jadt/jadt2000/pdf/47/47.pdf>
- [8] [http://nicolascormier.com/documentation/ia/cours\\_IA/node102.html](http://nicolascormier.com/documentation/ia/cours_IA/node102.html)
- [9] <https://dataanalyticspost.com/Lexique/reseaux-de-neurones-recurrents/>
- [10] <https://halshs.archives-ouvertes.fr/halshs-02096266/document>
- [11] <https://www.tandfonline.com/doi/pdf/>
- [12] <https://www.rncan.gc.ca/cartes-outils-publications/imagerie-satellitaire-photos-aer/tutoriels-sur-la-teledetection/analyse-interpretation-dimages/classification-et-analyse-des-images/9362>
- [13] <https://weave.eu/deep-learning-service-de-linformatique-affective/>
- [14] <https://ledatascientist.com/introduction-a-la-categorisation-de-textes/>
- [15] <https://www.aquiladata.fr/classification-dimages-et-detection-dobjets-par-cnn/>