REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA

Faculté de **T**echnologie

Département du **G**énie **E**lectrique

Mémoire de fin d'études de Master

Filière : **Télécommunication**

Spécialité : **Système des Télécommunication**

Mise en œuvre d'une stratégie de stimulation Cochléaire basée sur les réseaux de neurones Artificiels

---

Présentés par :

Oulddali Thanaa

Klaleche lemya

Proposés et Dirigés par :

Mr : Essaid Billel

Promotion 2022



Remerciements

Nous tenons tout d'abord à remercier Allah le tout Puissant et miséricordieux qui nous a donné la force et la patience d'accomplir ce Modeste travail. Nous remercions tous ceux qui ont participé de près ou de loin à la réalisation de ce projet

Nous remercions notre encadreurs Mr Esaid Billel pour les efforts qu'il a déployés, pour nous aider, conseiller, encourager et corriger .

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions

Nous ne pouvons pas terminer sans remercier tous les enseignants, tous les collègues qui nous ont aidés et qui nous ont entièrement soutenus au cours des dernières années

Dédiace :

Premièrement Je remercie Allah le tout puissant de m'avoir donné la santé et la volonté d'entamer et de terminer ce mémoire.

Je dédier cette thèse à :

à la mémoire de mes deux grands-pères paternelle et maternelle

Mes chères père et mère pour leurs sacrifices, leur amour, leur tendresse, leur soutien leur prières tout au long de mes études.

Mon oncle Oulddali Mustapha et son épouse Amri Amel pour son soutien moral et ses précieuse directive

À mes amis Guttache Nora, Chikhi Farida, Imen, et mon binome Klaleche Lemya, Toga Asma.

Oulddali Thanaa

Dédicace

avec grande plaisir que je dédie ce modeste travail ;

A' l'être le plus cher de ma vie ,ma mère qui m'a soutenu et encouragé durant ces années d'étude .

A' l'homme, mon précieux offre du dieu ,ma réussite et tout mon respect mon père

A mes chères sœur ***nassima*** *et* ***warda*** *pour encouragements permanents*

A mes chers frères ***mohammed*** *et* ***rayan*** *,pour leur appui et leur encouragement*

A tout ma famille pour leur soutien tout au long de mon parcours universitaire

A tout mes amis ***,sarah*** *,* ***Imane*** *,* ***Noura*** *,* ***farida*** *,* ***Amina ,somaia,*** *et* ***mohemmed rafik***

A mon binôme ***Oulddali Thanaa*** *pour son soutien moral, sa patience et sa compréhension tout au long de ce projet .*

Sans oublier tout les professeurs que ce soit du primaire ,du moyen ,du secondaire ou de l'enseignement supérieur

Klaleche lemya

الملخص :

الصمم مشكلة صحية عامة. حيث يشجع انتشار هذا المرض الباحثين على تطوير تقنيات مساعدة لحل هذه المشكلة. غرس القوقعة هي جهاز طبي يسمح بإعادة تأهيل فقدان السمع العميق. من خلال الاستراتيجيات المنبهة، في هذا العمل قمنا بتطوير استراتيجية (ACE) التي تستخدم الشبكات العصبية. حيث أبانت عن نتائج واعدة.

الكلمات الاستدلالية : الصمم, غرس قوقعة, استراتجية(ACE), الشبكات العصبية.

Résumé :

La surdité est un problème de santé publique. La propagation de cette maladie encourage les chercheurs à développer des techniques auxiliaires pour résoudre ce problème. L'implant cochléaire est un dispositif médical qui permet la réhabilitation d'une perte auditive profonde. Grâce à des stratégies de stimulation, dans ce travail nous avons développé la stratégie (ACE) qui utilise les réseaux de neurone. Où les résultats obtenus sont prometteuses

Mots-clés : surdité, implant cochléaire**,** stratégies (ACE), les réseaux de neurone.

Abstract:

Deafness is a public health problem. The spread of this disease encourages researchers to develop auxiliary techniques to solve this problem. The cochlear implant is a medical device that allows the rehabilitation of deep hearing loss. Through stimulation strategies, in this work we develop the (ACE) strategy that uses neural networks. were the results are promised.

Keywords: Deafness, cochlear implant, strategies (ACE), neural networks.

Table de matière

Introduction générale 1
Chapitre I : Les implants cochléaires et les stratégies de stimulation 2
I.1. Introduction 2
I.2. Définition du son 2
I.3. Caractéristiques générales physiques du signal sonore 2
I.3.1. Types de signaux sonores 3
I.4. Un mouvement périodique : 4
I.5. Fréquences audibles par l'être humain 6
I.6. Anatomie de l'oreille 7
 L'oreille externe : 7
 L'oreille moyenne : 7
 L'oreille interne : 8
I.7. Le fonctionnement de l'oreille 9
I.8. Le traitement de l'information sonore à la périphérie de l'oreille 9
I.8.1. Le codage de l'intensité 9
I.8.2. Le codage de la fréquence 10
I.9. La surdité 13
I.10. L'implant cochléaire : 14
I.10.1. Les composant de IC : 15
I.10.2. Le fonctionnement de I C 15
I.10.3. Les avantage des ICs : 15
I.10.4. Les inconvénients des ICs : 16
I.11. Les stratégies de stimulation cochléaire 16
I.11.1. Stratégie d'échantillonnage entrelacé continu(CIS) 16
I.11.2. SPEAK 17
I.11.3. La stratégie ACE 18
I.11.4. La stratégie CA (Compressed Analog) 18
I.11.5. Stratégie F0F2 19
I.11.6. Stratégie F0F1F2 : 19
I.11.7. Stratégie multi-pics (MPEAK) 20

I.11.8. Stratégie SAS (Simultaneous Analog Stimulation) : 21
I.12. Les implants cochléaires commercialisés : 21
 Ineraid 21
I.13. Conclusion : 23
Chapitre II : Réseaux neurone artificielle et l'apprentissage profond 24
II.1. Introduction 24
II.2. Intelligence artificielle et les réseaux de neurones 24
II.3. Historique sur les réseaux de neurones : 25
II.4. Généralités sur les réseaux de neurones : 26
II.4.1. Neurone biologique : 26
II.4.2. Réseau formel (neurone artificiel) 27
II.5. Principe du neurone artificiel : 27
II.5.1. Modèle d'un neurone artificiel 28
II.5.2. Fonctions de transfert : 29
II.5.3. Types de topologie : 30
II.6. Architecture des réseaux de neurones : 32
II.6.1. Réseau de neurones non bouclé : 33
II.6.2. Réseau de neurones bouclé : 33
II.7. Avantages et inconvénients des réseaux de neurones 34
II.7.1. Avantage 34
II.7.2. Inconvénients 34
II.8. Le perceptron 35
II.8.1. Perceptron multicouches 35
II.9. Les différents types de RNA 36
II.10. Définition de machine Learning (ML) : 36
II.11. Méthodes d'apprentissage : 37
II.11.1. Apprentissage supervisé 37
II.11.2. Unsupervised Learning : 38
II.11.3. Différence entre apprentissage supervisé et non supervisé 40
II.12. Apprentissage semi-supervisé 40
II.13. Apprentissage par Renforcement 41
II.14. Le choix d'un type d'apprentissage automatique 42
II.15. Deep Learning 42

II.16. Applications du Deep Learning 47
II.17. Domaine du système deep Learning 47
II.18. Machine Learning Vs Deep Learning: 47
II.19. Pourquoi Deep learning 48
II.20. Traitement des données 49
II.21. Découpage des données 49
II.22. Fonction perte : 50
II.23. Optimiseurs : 50
II.23.3. AdaGrad : 52
II.24. Conclusion : 53
Chapitre III: simulation de la stratégie Deep ACE 53
III.1. Introduction : 53
III.2. Environnements et Outils de développement 53
III.2.1. Python 53
III.2.2. Tensorflow 53
III.2.3. Keras 54
III.2.4. Numpy 55
III.2.5. Matplotlib 55
III.2.6. Anaconda 56
III.2.7. Spyder 56
III.3. Conception de la stratégie de stimulation 57
III.4. La base de donnée audio (Dataset) : 58
III.5. Déclaration des variables : 59
III.6. Génération des données : 60
III.7. Le model du réseau de neurone 63
III.8.1. Apprentissage 65
III.8.2. Evaluation 67
III.9. Conclusion : 71
Conclusion générale : 74

Liste des figures

Figure I. 1 : Représentation graphique d'un son pur 3
Figure I. 2 : Représentation graphique d'un signal complexe. La parole (ou bruit) est un exemple très concret de son complexe. 4
Figure I. 3 : Le son pur aigu représenté par la sinusoïde du haut a une fréquence élevée soit beaucoup de périodes en une seconde. A l'inverse le son pur grave représenté par la sinusoïde du bas a une fréquence faible soit très peu de périodes dans une unité de temps. 5
Figure I. 4 : Ce schéma illustre sur différentes courbes la distance entre deux points consécutifs d'un cycle complet soit la longueur d'onde de chacune d'elle 5
Figure I. 5 : Fréquences audibles par l'homme. 7
Figure I. 6 : Schéma illustrant les vibration dans le conduit cochléaire déroulé. 8
Figure I. 7:Organisation tonotopique cochléaire (A) et réponse de la membrane basilaire aux stimulations acoustiques (B, C, D). Les sons de haute fréquence font vibrer uniquement la partie basale de la cochlée (B) alors que les sons de basse fréquence induisent une vibration jusqu'à l'apex (D) 12
Figure I. 8 : les composant de IC 15
Figure I.9 : Schéma bloc de la stratégie CIS. 17

Figure II. 1 : représente les élément d'un neurone biologique. 26
Figure II. 2: Schéma d'un neurone formel 27
Figure II. 3 : Modèle d'un neurone élémentaire. 28
Figure II. 4 : Réseau à propagation avant à une couche. 31
Figure II. 5: Réseau à propagation avant multicouche. 31
Figure II. 6 : Réseau récurrent avec neurones cachés. 32
Figure II. 7 : un réseau de neurones non bouclé 33
Figure II. 8 :un Réseau de neurones bouclé. 34
Figure II. 9 : Perceptron multicouches 35
Figure II. 10 : Les Types d'apprentissage en ML 37
Figure II. 11 : Apprentissage supervisé 38
Figure II. 12 :Apprentissage non-supervisé. 39
Figure II. 13: Regroupement (Clustering) 39
Figure II. 14 : Apprentissage semi-supervisé. 41
Figure II. 15 : Concept d'apprentissage par renforcement 42
Figure II. 16 :Les sous-bronches de l'intelligence artificiel. 43
Figure II.17: L'architecture des réseaux de neurones convolutifs. 44
Figure II.18 :schéma représenté l'architecture d'un autoencoder. 46
Figure II. 19 : ML vs DL 48

Figure III. 1 : Log du python 53

Figure III. 2 :Logo du TensorFlow 54
Figure III. 3 :Logo du Keras 55
Figure III. 4 :Logo du Numpy 55
Figure III. 5 :Logo du matplotlib 55
Figure III. 6 :Logo du Anaconda 56
Figure III. 7:Logo du Spyder 56
Figure III. 8 : L'environnement de travail du Spyder 57
Figure III. 9:Schéma bloc de la stratégie de stimulation 57
Figure III. 10 : Deep ACE 58
Figure III. 11 :l'appelle des bibliothèques 60
Figure III. 12 : Aucune entrée de table d'illustration n'a été trouvée. 61
Figure III. 13 :les deux modes train et valid 62
Figure III. 14 : l'encodage des signaux 62
Figure III. 15 : le décodage des signaux 63
Figure III. 16 :création des couches 64
Figure III. 17 :création des couches ..... Erreur ! Signet non défini.
Figure III. 18 : l'appelle des bibliothèques 65
Figure III. 19 :le programme générale 65
Figure III. 20 :L'évolution du MSE 66
Figure III. 21 :Contenu du fichier de sauvegarde des paramètres du model neuronal ... 67
Figure III. 22:les valeurs de chaque éléctrode 67
Figure III. 23 :Représentation des 22 valeurs de la fonction LGF 70
Figure III. 24 :Sélection des 8 valeurs de la fonction LGF parmi 22 71

Liste des tableaux

Tableau II.1 : tableau sur Fonction de transfert $y_{k=}\varphi$ (SP) ……. 30

Tableau II.2 : la Différence entre apprentissage supervisé et non supervisé ……..……40

Liste des abréviations

ABC : Advanced Bionic Coporation

ABS : Antweep Bionic Systems

ACE: Advenced combination encoder

AIS : Asynchronous Interleaved Stimulation

B : Biais.

C : célérité du mouvement vibratoire en (m/s)

CA : La stratégie Compressed Analog

CIS : Stratégie d'échantillonnage entrelacé continu

CNN : Convolutional Neural Networks

DL : Deep Learning

F : La fréquence

f : La fonction d'activation (ou de transfert).

GAN ; generative adversal networks

IA : L'intelligence artificielle

IC : implant cochléaire

ML : Machine Learning

MPEAK : Strategies multi-pics

MSP : Mini Speech Processor

MSE : Mean Square Error

PHI : Philips Hearing Implants

ReLU : Rectified linear unit

RNA : réseaux de neurones artificiels

RNN : Réseaux neuronaux récurrents

SAS : Simultaneous Analog Stimulation

T : période du mouvement vibratoire en (s)

W : Poids synaptique du neurone.

WSP-III : wearable speech processor

x : Entrée du neurone.

y : La sortie du neurone.

λ : longueur d'onde en (m)

Introduction générale

Les implants cochléaires (IC) permettent une restauration partielle de la fonction cochléaire en cas de surdité causée par des défauts cellulaires, et les fibres du nerf auditif sont préservées. L'implant consiste en une prothèse électronique insérée dans la cochlée, qui stimule le nerf auditif avec des électrodes. Le sons externes est captés par des microphone, analysé numériquement et traités dans un processeur, où ils sont convertis en signaux codés pour stimuler le nerf auditif.

Dans le premier chapitre de ce mémoire, nous présentons les caractéristiques d'un son, puis nous décrivons l'oreille humaine et les différents types de surdité, et finalement nous parlons sur l'implant cochléaire par la description de ses composants et son principe de fonctionnement, ses limites et les stratégies utilisée dans la stimulation cochléaire par les différents fabricants d'implants.

Dans le deuxième chapitre, nous parlons sur l'intelligence artificielle (IA) qui est largement utilisée dans la vie quotidienne dans différents domaines tels que l'industrie, la médecine, biologie, éducation et même les jeux vidéo, nous allons parler aussi sur les réseaux de neurones artificiels avec ses caractéristiques et ses modèles. Et nous allons également parler de l'apprentissage profond (Deep learning), est l'une des branches les plus importantes et les plus exploitées de l'apprentissage automatique. Grâce à son succès, il a réalisé un saut quantique dans le domaine de l'intelligence artificielle. Par conséquent, il est devenu le domaine le plus important dans le monde de l'informatique. L'apprentissage profond est basé sur l'apprentissage et l'acquisition d'expériences. Il simule la façon dont les humains pensent et apprennent.

Dans le troisième chapitre nous simulons la stratégie ***Deep ACE*** par le langage de programmation Python en utilisant le logiciel ***Anaconda*** (***Spyder***) dans le but de faire l'apprentissage du system.

Le présent mémoire se termine par une conclusion qui résumera les résultats obtenus dans cette étude suivie de perspectives.

Chapitre I : Les implants cochléaires et les stratégies de stimulation

Chapitre I : Les implants cochléaires et les stratégies de stimulation

I.1. Introduction

Dans ce chapitre nous allons parler premièrement de l'anatomie et le fonctionnement de l'oreille humaine, ensuite sur l'implant cochléaire qui est la solution des gens qui ont le problème de la surdité.

L'implant cochléaire est un dispositif électronique permet une meilleure compréhension possible de la parole, il est destiné aux enfants et aux adultes qui ont la surdité sensorielle qui représente un problème au niveaux de l'oreille interne (la cochlée), c'est-à-dire les minuscules cellules ciliées de l'oreille interne sont trop déficient ou sont absentes, en conséquent que le son même s'il est amplifié, il ne peut pas se transmit au cerveaux.

Dans cette partie nous allons également parler sur les composant de cette appareille qui a deux partie : la partie interne qui contient le faisceau d'électrode et un implant, et la partie externe qui se mettent et s'enlève, elle contient : un processeur de son et une antenne.

Ensuite nous allons étudier les diffèrent stratégies qui sont utilisé pour transformé le signal acoustique reçue au nerf auditif, comme SPEAK, MPEAK …etc.

I.2. Définition du son

Une onde sonore est la vibration d'une molécule autour de sa position d'équilibre (ou état de repos) et elle se propage avec des perturbations dans un milieu (généralement l'air), mais peut aussi être solide ou liquide. Cette vibration est captée par nos oreilles pour faire bouger le tympan, qui est le point de départ de la stimulation de l'oreille et de la perception des informations sonores .[1]

I.3. Caractéristiques générales physiques du signal sonore

Les ondes sonores sont caractérisées par plusieurs grandeurs physiques que nous décrirons en détail dans la première partie.

I.3.1. Types de signaux sonores

Distinguer deux types de signaux sonores :

- Le signal d'un son complexe.
- Le signal d'un son pur.

➢ Le son pur

Modélisé par une fonction sinusoïdale périodique qui transforme plusieurs Pour caractériser la quantité d'ondes sonores, **la Figure I.1** représente le son pur :

- une fréquence.
- une longueur d'onde.
- une amplitude ou intensité sonore.

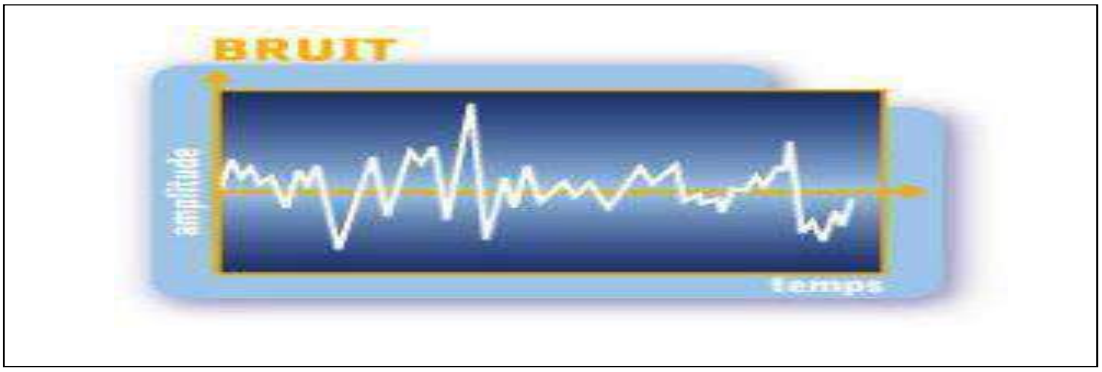


.

Figure I. 1 : Représentation graphique d'un son pur .

➢ Les sons composés périodiques

Il est généralement composé d'un grand nombre de sons purs superposés. Par conséquent, nous interprétons le signal complexe comme la somme d'un grand nombre de sinusoïdes. Toutes les voix ne peuvent pas tester cette théorie. Les notes d'un instrument tel qu'un piano sont un son complexe qui répond parfaitement à ce concept acoustique.



F**igure I.2 :** Représentation graphique d'un signal complexe. La parole (ou bruit) est un exemple très concret de son complexe.

I.4. Un mouvement périodique :

Le mouvement périodique est caractérisé par :

- **La fréquence**

La fréquence est le nombre de périodes dans une seconde, c'est-à-dire le nombre de fois qu'un phénomène périodique se reproduit par unité de temps. La période et la fréquence sont directement liées, donnant la fréquence **F** (Hertz) en fonction de période **T**(secondes) est :

$$F = 1/T \qquad (I.1)$$

Plus la fréquence est élevée, plus le son est aigu et plus la fréquence est basse, plus le son est faible. Plus précisément, les graves d'un piano auront des fréquences plus basses que les aigus de ce dernier. La Figure I.3 représente le son de haute pureté [2].

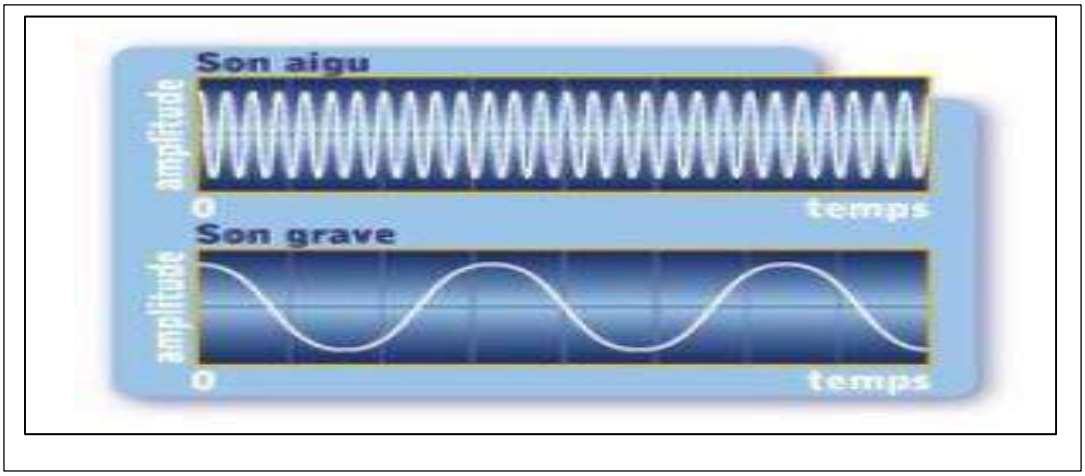


Figure I. 3 : Le son pur aigu représenté par la sinusoïde du haut a une fréquence élevée soit beaucoup de périodes en une seconde. A l'inverse le son pur grave représenté par la sinusoïde du bas a une fréquence faible soit très peu de périodes dans une unité de temps.

- **La longueur d'onde**

La longueur d'onde λ est la distance parcourue par l'onde pendant une période d'oscillation et s'exprime en mètre

$$\lambda = c\,T \qquad \textbf{(I.2)}$$

avec :

- λ : longueur d'onde en **(m)**
- **C** : célérité du mouvement vibratoire en **(m/s)**
- **T** : période du mouvement vibratoire en **(s)**

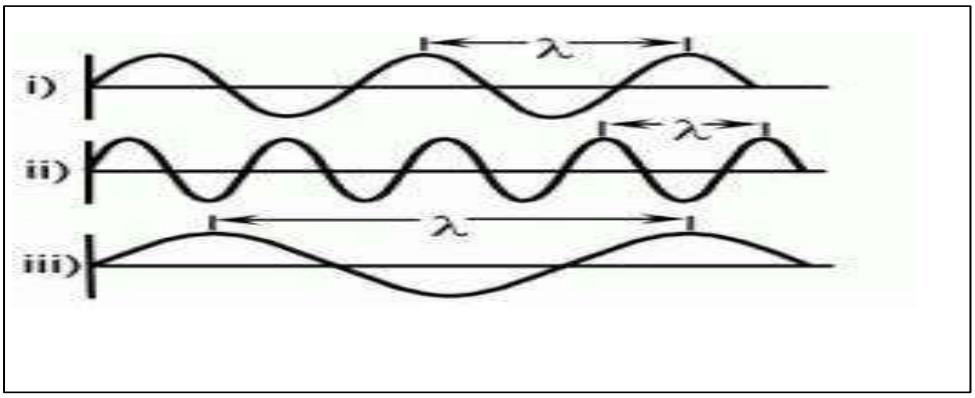


Figure I. 4 : Ce schéma illustre sur différentes courbes la distance entre deux points consécutifs d'un cycle complet soit la longueur d'onde de chacune d'elle.

- **La période**

Un phénomène est dit périodique s'il se reproduit identique à lui même au bout d'un certain temps. Elle s'exprime en seconde (s). [3]

$$T = \frac{1}{F} \qquad \textbf{(I.3)}$$

- **L'intensité**

L'intensité d'un son, également appelée volume, fait la distinction entre les sons forts et faibles. Elle correspond à l'amplitude de l'onde. L'amplitude est donnée par l'écart maximal de l'amplitude de l'onde caractéristique.

- **La vitesse**

La vitesse du son est la vitesse à laquelle se déplacent les ondes sonores. Elle est définie par la relation suivante : [4]

$$C = \omega / k \qquad \textbf{(I.4)}$$

- ✓ ω étant la fréquence de l'onde (en rad.s-1)
- ✓ k étant la norme de son vecteur (en rad.m-1)

I.5. Fréquences audibles par l'être humain

Les humains peuvent entendre des fréquences de 20 à 20 000 Hz. Ces chiffres peuvent varier selon l'âge et la personne. tous les sons avec moins de 20 vibrations par seconde sont appelés infrasons, tandis que les sons avec plus de 20 000 vibrations par seconde sont appelés ultrasons. [3]

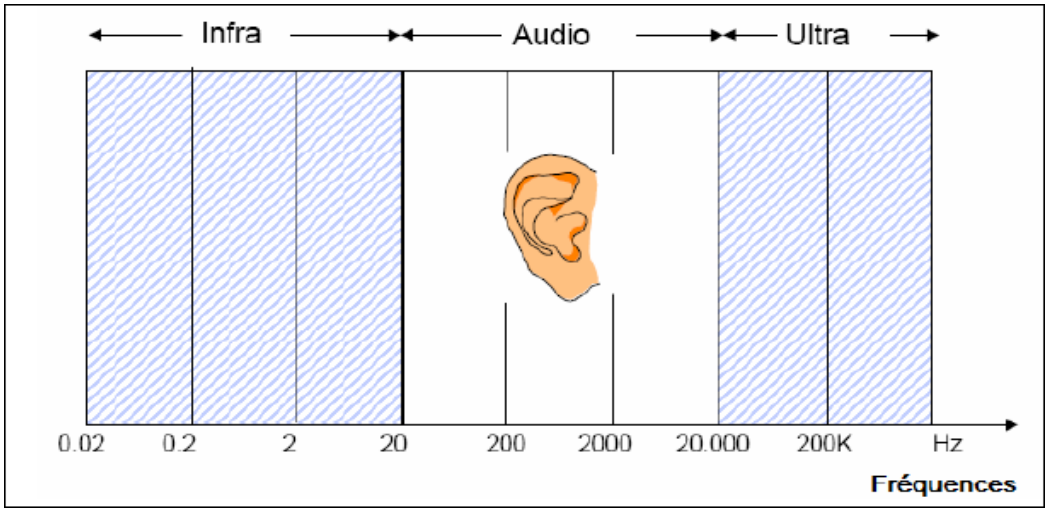


Figure I. 5 : Fréquences audibles par l'homme.

I.6. Anatomie de l'oreille

Le système auditif de l'oreille humaine se compose en trois partie qui sont : l'oreille externe, l'oreille moyenne et l'oreille interne :

- **L'oreille externe :**

L'oreille externe est constitué de : Pavillon, Canal Auditif (conduit auditif externe), Membrane tympanique

- Pavillon : qui recueille et transmet les sons au canal auditif.
- Canal Auditif : qui dirige les sons dans l'oreille.
- Les ondes sonores qui arrive du canal auditif provoque des vibrations au tympan.

- **L'oreille moyenne :**

L'oreille moyenne est constituée de : Tympan (membrane tympanique), Chaîne des osselets

- Tympan : transforme les sons en vibrations
- Chaîne des osselets : qui contient : marteau, enclume, étrier et qui est le responsable de la transmission des vibrations à l'oreille interne. [5]

Les vibrations de tympan sont transmises au osselets qui est composé de marteau qui transmet ces vibrations au enclume puis à étrier

- **L'oreille interne :**

L'oreille interne est constitué d'une cochlée, et un système de vestibulaire, et nerf auditif.

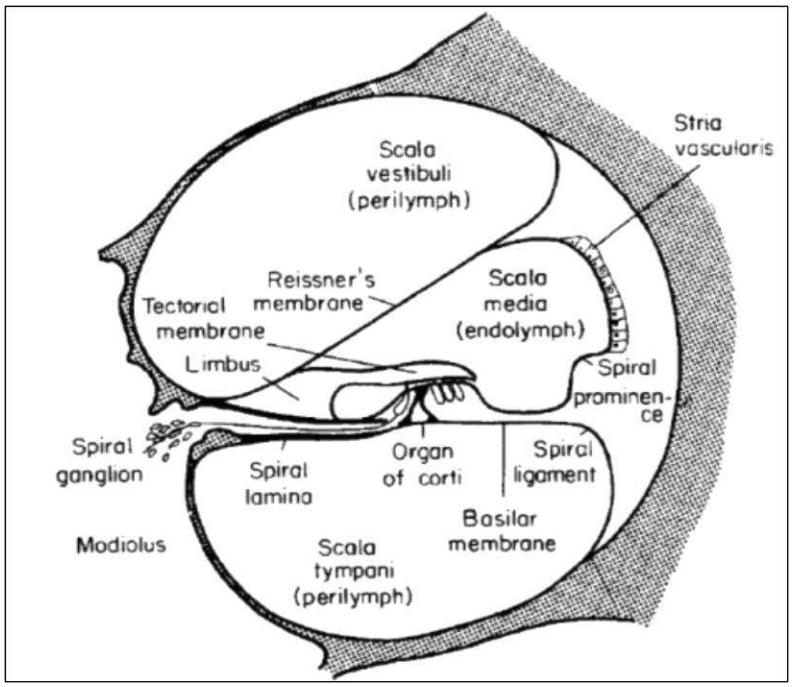


Figure I. 6 : Schéma illustrant les vibration dans le conduit cochléaire déroulé.

- Cochlée : est l'organe principale d'un système auditif, il contient des fluides et des cellules très sensibles qui s'appelle **des cellules ciliées**, avec des structures en forme de cils qui bougent grâce aux vibrations sonores.

La cochlée est une structure osseuse de 35 mm enroulée sur elle-même. Elle fait deux tours trois-quarts. Comme cela est montré schématiquement sur la **Figure I.6**, elle est divisée longitudinalement en trois espaces remplis de fluide : le canal vestibulaire (scala vestibuli), le canal tympanique (scala tympani), et le canal cochléaire (scala media). Les canaux externes (vestibulaire et tympanique) communiquent à son sommet par une ouverture, l'hélicotrème . Ils sont remplis de périlymphe qui est riche en ions sodium et pauvre en ions potassium.

Le canal cochléaire forme un compartiment interne clos, qui est séparé du canal vestibulaire par la membrane de Reissner et du canal tympanique par la membrane basilaire. Il contient de l'endolymphe, riche en potassium et pauvre en sodium. [6]

- Système Vestibulaire : elle contient entre 15-20 milles cellules cilié qui détecte les vibrations de liquide. [5]
- Nerf auditif : qui transmet l'information de la cochlée au cerveau[5]

I.7. Le fonctionnement de l'oreille

Le son capté par l'oreille se transmet à partir de la conduire auditif vers le tympan, ce dernier va convertir le son en vibration, cette vibration entrainera un mouvement de petite osselet (marteau, enclume, étrier) qui sont attachés directement derrière le tympan dans l'oreille moyenne, les osselets transmettant ces vibrations à la cochlée

La cochlée contient des cellules ciliées qui détecte le mouvement de liquide. Les ondes stimulent les cellules ciliées qui génèrent des signaux électriques, transmis ensuite au nerf auditif. Les cellules ciliées de la partie apicale de la cochlée envoient l'information portant sur les fréquences graves tandis que celles de la partie basale envoient l'information portant sur les fréquences aiguës. Et enfin Le nerf auditif transmet ces informations au cerveau. [5]

I.8. Le traitement de l'information sonore à la périphérie de l'oreille

Les sons sont des phénomènes vibratoires qui peuvent être apériodique (impulsionnel ou continu) ou périodique (simple ou complexe). Un son périodique simple peut être représenté par une courbe sinusoïdale, un son périodique complexe par une courbe complexe régulière. L'analyse fréquentielle par transformé de Fourier permet de décomposer une onde périodique complexe en ses composantes sinusoïdales, les harmoniques, caractérisés chacun par une fréquence ou hauteur (exprimée en hertz, Hz) et une amplitude ou intensité (exprimée en décibels, dB). Les fréquences des harmoniques sont des multiples entiers d'une fréquence de base : le fondamental.

I.8.1. Le codage de l'intensité

Le système auditif utilise plusieurs méthodes complémentaires pour coder les variations d'intensité des sons purs. Le taux moyen de décharge d'une fibre ou d'une

population de fibres coderait en partie l'intensité d'un stimulus. Lorsque pour une fibre donnée du nerf auditif on accroît l'intensité du stimulus, la fréquence de décharge des potentiels d'action augmente suivant une courbe sigmoïde pour toutes les hauteurs tonales

Le champ dynamique des fibres (l'augmentation de l'amplitude du stimulus nécessaire pour que le taux de décharge moyen augmente depuis son seuil jusqu'à saturation) est généralement compris entre 20-50 dB, alors que la dynamique acoustique est de 100 à 120 dB. Pour expliquer comment l'activité de ces fibres peut couvrir toute la dynamique acoustique, Liberman (1978) montre qu'il existe deux types de fibres de type I au niveau de la cochlée. Ils se différencient par leur diamètre et leur taux de décharge spontanée. D'un point de vue fonctionnel, les fibres ayant un petit diamètre et un taux de décharge spontanée élevé ont un seuil de déclenchement (défini comme étant l'intensité à laquelle le taux de décharge moyen augmente au-dessus du taux spontané) bas de 0 dB HL. Les fibres ayant un gros diamètre et un taux de décharge spontanée inférieur ont un seuil de déclenchement plus haut, vers 40-50 dB HL. Le premier type de fibres code les intensités entre 0 et 40-50 dB HL et le deuxième type de fibres code les intensités plus élevées, les fibres à seuils bas étant alors saturées. Le degré de coïncidence entre les décharges dans une population de fibres serait également utilisé pour coder l'intensité (Carney, 1994). [6]

Les caractéristiques temporelles de la réponse du nerf auditif en fonction de la vibration de la membrane basilaire ne doivent pas faire oublier cependant que ces vibrations sont différentes selon la position sur la membrane, ce qui génère également un codage tonotopique/spatial. Un modèle spatial permettrait donc également d'expliquer le codage de l'intensité. Une théorie propose en effet que l'intensité du stimulus soit codée par la dispersion spatiale de l'excitation neurale générée par le stimulus acoustique. [6]

I.8.2. Le codage de la fréquence

I.8.2.1. Le codage temporel

Même au repos (sans vibration de la membrane basilaire), les cellules ciliées internes ont une activité spontanée. Elles génèrent en moyenne 18 potentiels d'action par seconde et déchargent parfois même jusqu'à 100 fois par seconde. En présence d'une stimulation acoustique, la membrane basilaire vibre et les stéréocils des cellules ciliées internes se penchent en avant et en arrière comme cela est décrit plus haut. Cela fait varier la quantité de

neurotransmetteurs libérés de manière à ce que les potentiels d'action aient de plus grandes chances d'être générés pour une phase du stimulus plutôt que pour une autre. Ce principe de synchronisation en fonction d'une fréquence de prédilection ou fréquence caractéristique est appelé « phase-locking ».

Lorsque l'amplitude des vibrations de la membrane basilaire augmente, les quantités de neurotransmetteurs libérés augmentent également. Les taux de décharge moyens et la synchronisation des décharges évoluent à peu près en fonction de l'amplitude de la vibration. Le taux de décharge cependant n'augmente pas linéairement, parce que le processus de transduction n'est pas linéaire lui-même.

Dans les théories sur le codage temporel de la fréquence, il est fait l'hypothèse que les informations sur le stimulus acoustique sont codées par le rythme des réponses des fibres nerveuses. Par exemple, une théorie du codage fréquentiel temporel (pitch rate) propose que la fréquence soit codée par les intervalles inter-pics de chaque fibre ou d'une population de fibres (on parle alors de « volleying »). Cependant, il existe des périodes réfractaires qui limitent la rapidité avec laquelle une fibre qui vient de décharger peut se trouver à nouveau dans un état qui lui permette de décharger. Le taux de décharge est ainsi limité à un maximum de 1000 pics par seconde. Il est donc généralement reconnu que ce codage est limité aux fréquences de stimulation inférieures à 4000 Hz. [6]

I.8.2.2. Le codage tonotopique (spatial)

L'analyse fréquentielle est effectuée via des mécanismes passifs de la membrane basilaire selon la théorie de l'onde propagée décrite par von Bekesy (il reçu en 1961 le Prix Nobel de Physiologie en Médecine pour ses recherches). La distance parcourue par l'onde propagée le long de la membrane basilaire dépendrait de la fréquence du son. En effet, la structure de la membrane basilaire n'est pas identique à la base et à l'apex de la cochlée. A la base, la membrane est moins large, d'environ un facteur 5, mais également plus rigide, d'environ un facteur 100. Ainsi, lorsqu'un son de haute fréquence atteint l'oreille, la base de la membrane vibre beaucoup, en dépensant beaucoup d'énergie et l'onde ne se propage pas très loin. Pour les basses fréquences, l'onde peut se déplacer de la base jusqu'à l'apex de la membrane basilaire, avant que toute l'énergie soit épuisée. Ce mécanisme est décrit schématiquement sur la Figure I.2. La membrane basilaire peut ainsi être présentée comme

étant une série continue de filtres à bande passante, la fréquence centrale de chaque filtre diminuant de manière monotone avec la distance depuis la base. Cette relation qui lie la fréquence du son (de 20 à 20 000 Hz) et la localisation entre la base et l'apex du codage est nommée carte cochléaire fréquentielle par Liberman (1982). Il s'agit de l'organisation tonotopique de la cochlée: les fréquences aiguës sont localisées vers la base, les fréquences graves vers l'apex. Greenwood (1990) a proposé une carte de la cochlée associant position et fréquence : la carte qu'il a décrit contient 35 bandes critiques de 1 mm chacune et est souvent utilisée comme référence dans les articles scientifiques. [6]

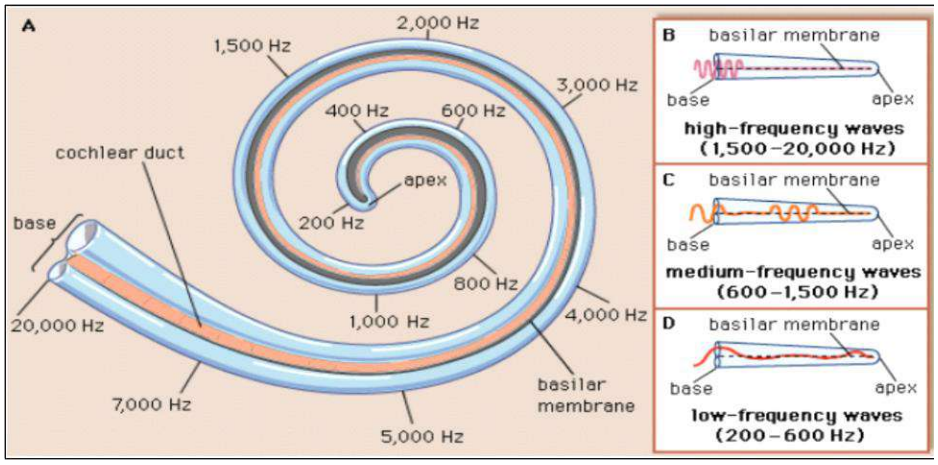


Figure I. 7: Organisation tonotopique cochléaire (A) et réponse de la membrane basilaire aux stimulations acoustiques (B, C, D). Les sons de haute fréquence font vibrer uniquement la partie basale de la cochlée (B) alors que les sons de basse fréquence induisent une vibration jusqu'à l'apex (D) .

I.8.2.3. Un codage mixte

En réalité, les théories temporelles et spatiales sont complémentaires et le codage des sons au niveau de la cochlée suit un mécanisme temps-place (Licklider, 1954). Un mécanisme temporel est particulièrement important pour le traitement des basses fréquences, alors qu'un mécanisme selon la place est essentiel pour les hautes fréquences (Wever, 1949).

Il est généralement reconnu que le codage selon la place est inadapté pour les fréquences en dessous de 300-400 Hz et voire même au-delà. En effet, le seuil de discrimination de fréquence à 2000 Hz est 0.2 % (Traux, 1999), alors que cette différence de fréquence perçue est égale à la distance entre deux cellules ciliées internes voisines. Le

codage par la place, selon lequel une fréquence active préférentiellement certaines cellules mais également les cellules voisines de manière moins spécifique, ne peuvent par conséquent expliquer à lui seul le codage de la fréquence des sons graves. D'un autre côté, le codage des fréquences supérieures à 4000-5000 Hz ne peut être temporel, car cela demanderait des décharges trop rapides aux neurones, par rapport à la durée de leur période réfractaire.

Loeb et coll. (1983) ont proposé un modèle impliquant trois types de mécanismes pour la perception de la fréquence. Le premier mécanisme utilise le codage spatial pour les fréquences allant de 5000 à 20 000 Hz, le deuxième utilise le « volleying » pour les fréquences entre 500 et 5000 Hz (Kiang, 1965), et le troisième le codage temporel pour les fréquences allant de 300 à 500 Hz où les nerfs sont capables de décharger à chaque cycle.

Une théorie, basée sur la latence de l'activité cochléaire et sur son expansion au niveau du cortex auditif, a été développée par Greenberg et coll. (1998). Elle intégrerait la notion de délai fréquence-dépendant lié au passage de l'onde acoustique dans la cochlée au codage par la place et le rythme. Les sons graves stimuleraient ainsi le tronc cérébral (Greenberg, 1980) et le cortex auditif (Woods et coll., 1993) plus tardivement que les sons aigus qui transitent très rapidement par la cochlée (le délai est de l'ordre de dizaines ou centaines de microsecondes pour des fréquences supérieures à 4 kHz). Le temps total pour aller de la base à l'apex serait de 10 ms au moins chez l'animal (Goldstein et coll., 1971 ; Ruggero & Rich, 1987). [6]

I.9. La surdité

La surdité est un affaiblissement ou une abolition complète du sens de l'ouïe. En cas d'affaiblissement, on parlera d'hypoacousie, l'abolition complète est qualifiée de cophose. Les termes « sourd » ou « malentendant » se disent de toute personne présentant un trouble de l'audition sans préjuger de l'importance de cette atteinte ni de son caractère uni ou bilatéral. Cinq à six millions de personnes sont malentendantes en France. Les pertes auditives touchent : trois personnes sur dix chez les plus de 60 ans, une personne sur six dans la tranche d'âge comprise entre 41 et 59 ans et une sur quatorze pour les personnes entre 29 et 40 ans. Au moins 1.4 millions d'enfants ont un problème auditif. Un enfant sur mille né sourd et un sur mille le devient. La surdité peut se décrire de différentes manières selon que l'on considère son degré de gravité, l'âge de la personne à son apparition, le délai de sa mise en place ou encore son origine anatomique.

Il y a 4 type de perte d'audition qui sont : surdité de transmission, surdité neurosensorielle, surdité mix et surdité rétro-cochléaire .

- ✓ **Surdité de transmission :**

C'est des problèmes concernant l'oreille externe ou l'oreille moyenne qui empêche la transmission correcte du son vers l'oreille interne, ces types de surdité sont temporaires[5]

- ✓ **Surdité neurosensorielle :**

Ce type de surdité et due à une perte des cellules cilié dans la cochlée, elle peut être légère, moyenne, sévère ou profonde.

Les surdités neurosensorielles (légères à sévère) se résout par des aides auditif ou implant d'oreille moyenne, tant que les pertes auditif (sévère à profond) peuvent être améliorées avec un implant cochléaire. [5]

- ✓ **Surdité mixte :**

C'est une combinaison d'une parte auditif de transmission et de neurosensorielle, elle représente un problème au niveau de l'oreille moyenne et l'oreille interne. [5]

- ✓ **Surdité rétro-cochléaire :**

C'est l'absence de nerf auditif ou le nerf auditif est endommagé, c'est une surdité profond et permanent. [5]

Notre sujet parle de la solution de la surdité neurosensorielle, qui est l'implant cochléaire.

I.10. L'implant cochléaire :

Un implant cochléaire est un dispositif électronique permet une perception sonore optimale et une meilleure compréhension de la parole. C'est la solution pour les enfants et les adultes qui ont la surdité sévère à profond auxquelles les prothèses auditives conventionnelle apportent une aide insuffisante, car les minuscules cellules ciliées de la cochlée sont très déficientes ou sont absentes, et ce dernier fait que le son n'arrive pas au cerveau. [7]

I.10.1. Les composant de IC :

L'implant cochléaire est fonctionné en court-circuitant les cellules ciliées pour que le son transmet de la chaine des osselets au nerfs auditif par des électrodes. Un implant cochléaire construit de deux partie : partie externe et partie implantée chirurgicalement.

La partie externe contient un microphone, un processeur de son, et une antenne, ce dernier peut s'enlève et se mettent facilement. Et la partie implantée chirurgicalement contient l'implant et le faisceau d'électrode, cette partie reste toujours invisible [7]

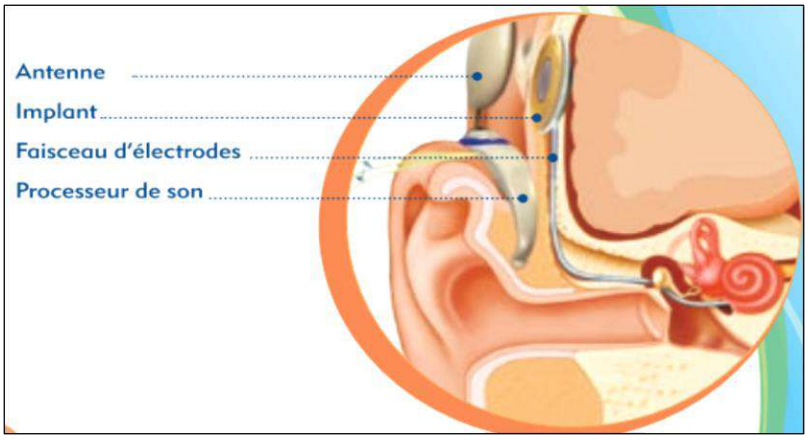


Figure I. 8 : les composant de IC .

I.10.2. Le fonctionnement de I C

Le microphone capture les sons environnants (exemple : la parole, la musique …etc.), ces dernières sont traitées par le processeur qui se trouve dans la partie externe du système qui transforment ces sons sous forme de signaux à travers la peau. L'implant conduit ces signaux au long du faisceau d'électrodes jusqu'à chaque électrode qui les apporte au nerf auditif. Et finalement ces signaux sont transmis au cerveau par le nerf auditif. [8]

I.10.3. Les avantage des ICs :

Les systèmes d'implant cochléaire (IC) peuvent offrir de nombreux avantages, tels qu'écouter la parole, les sons environnementaux et la musique. Les résultats obtenus avec les implants cochléaires montrent actuellement la compréhension de la parole pour la plupart des malades. Au sein de la communauté scientifique, les avantages d'un implant cochléaire sont mesurés par le pourcentage de parole compris par un implant cochléaire sans lecture labiale. Bien que comprenant les mots de la liste ouverte est un indicateur très important de la capacité auditive, ce n'est pas le seul bénéfice rapporté par les patients diffusés. La perception

des sons de l'environnement, comme les cloches, les signaux d'alarme, etc. peut faire une grande différence dans la vie d'un patient atteint d'une perte auditive sévère [7]

I.10.4. Les inconvénients des ICs :

L'inconvénient du système IC est que ce système n'est pas la solution pour tous les surdités, voici quelques facteurs a considéré :

- ✓ **Meilleur option :** lorsqu'une aide auditif classique permet une bonne compréhension de la parole cela signifie que cette option est meilleur que l'utilisation d'un système IC
- ✓ **Une surdité profonde pendant une très longue période :** lorsque le nerf auditif n'a pas été stimulé pendant une très long période, cela conséquent que ce nerf n'est pas capable de transmettre l'information correctement au cerveau.
- ✓ **La cochlée n'est pas la cause principale de la surdité :** si le problème ne se trouve pas au niveau de l'oreille interne donc le système IC n'aura aucun effet.
- ✓ **L'intervention n'a que peu de chance de réussir :** si la cochlée ne peut pas recevoir de port-électrodes, ou si le nerf auditif est endommagé, donc le IC ne donnera aucun résultat. [7]

Dans le plan de l'implant cochléaire il faut une technologie qui ramène le son jusqu'à le système auditif, et les facteurs principaux pour ce système est : **le concept de l'implant** et **la stratégie** de traitement du son.

I.11. Les stratégies de stimulation cochléaire

I.11.1. Stratégie d'échantillonnage entrelacé continu(CIS)

Cette stratégie stimule les électrodes les unes après les autres. Les électrodes délivrent des impulsion bi-phasique d'un rapport cyclique égale ½ avec un décalage temporel pour omettre tous les chevauchements entres les canaux. Ensuite l'enveloppe des impulsions résulte

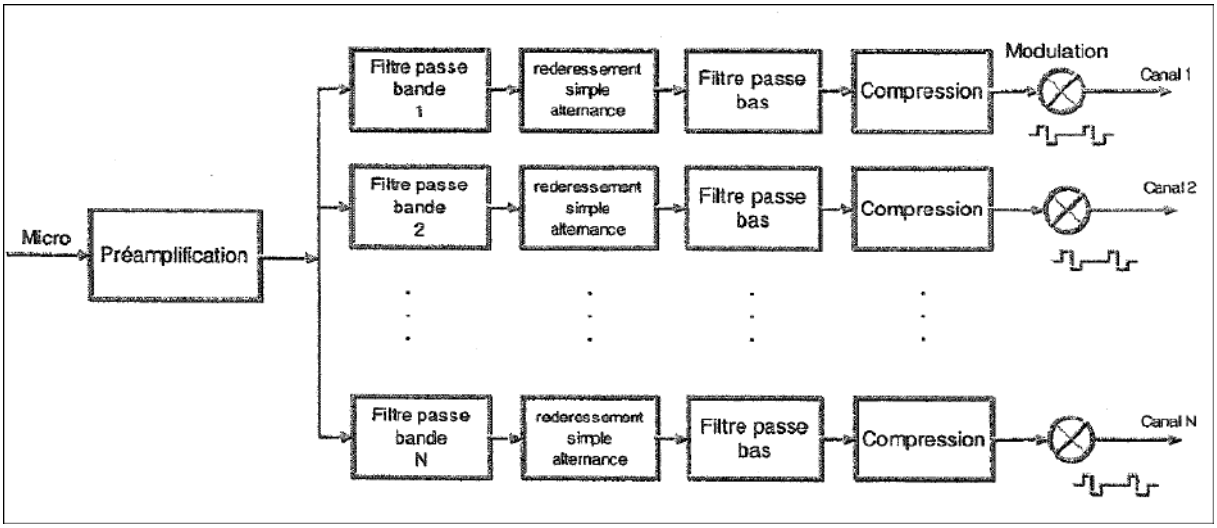


Figure I.9 : Schéma bloc de la stratégie CIS.

le signal résulte sera amplifier et passé par 5 ou 6 bandes de fréquence, les signaux de sortie de ces filtres sont extrait par redressent et filtrage passe bas de fréquence de coupure (200-400) Hz, ces derniers seront compressés par une loi non-linaire de type logarithme en générale

La stimulation se déroule à un taux constant indépendant de la nature de son (voisé ou non voisé), ce taux dépasse 800 impulsion par seconde pour chaque canal [10] , comme représente la **figure I.9** .

I.11.2. SPEAK

Pour évaluer le contenu de tout spectre de signal de la parole on utilise 16 filtres. après l'amplification , le signal est subdivisé en 16 bandes à cause des 16 filtres passe bande centré entre les fréquence 250 et 5400 Hz , ensuite chaque sortie du filtre est redressé puis filtré par des filtre basse bas a une fréquence de coupure 200 Hz pour déterminer l'amplitude maximal , ensuite la sélection des 6 sortie parmi les 16 filtres ayant les plus haut amplitude, ces amplitude vont transmettre vers les 6 canaux de stimulation correspondant à leur bande de fréquence sous forme des imputions diphasique à un taux de 250 imputions par seconde. [10]

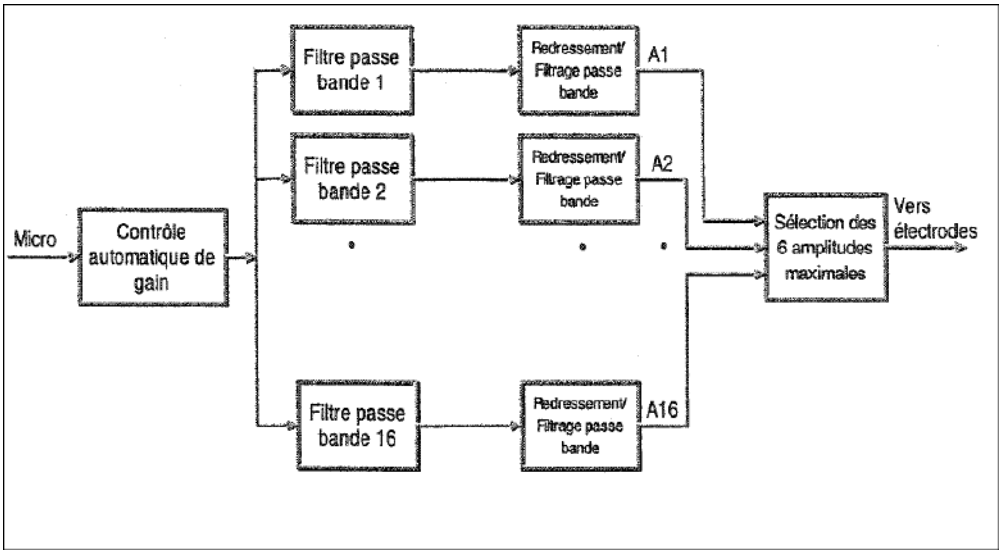


Figure I.10: Schéma bloc de la Stratégie SPEAK.

I.11.3. La stratégie ACE

C'est une stratégie combine les caractéristiques de SPEAK et CIS, le son est décomposé en 20 de bande de fréquence comme la stratégie SPEAK, et les électrodes sont stimulé l'un après l'autre comme la stratégie CIS.

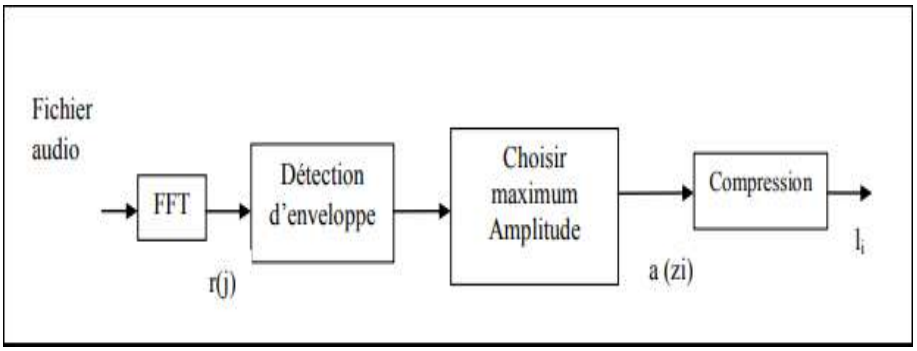


Figure I.11: schémas bloc de la stratégie ACE.

I.11.4. La stratégie CA (Compressed Analog)

Est faite par l'analyse du signal en fréquence par huit filtre, et le signal analogique obtenue va se distribuer de façon simultané aux électrodes.

I.11.5. Stratégie F0F2

Est la première stratégie développée, elle est basée sur l'extraction des caractéristiques des paroles. Elle est expérimenter par l'équipe de Clark à l'université Melbourne en 1978. Elle est basée sur l'extraction de fréquence de la parole F0 (fréquence fondamentale) et la fréquence du 2em formant F2.

F0 détermine le rythme de la stimulation, généralement, dans le cas des segments voisés de parole cette fréquence se trouve entre 50 et 300 HZ, ms dans le cas des segments non voisés, la stimulation est effectué à des intervalle pseudo-aléatoires à un rythme de100 pulsations par seconde. [10]

I.11.6. Stratégie F0F1F2 :

C'est l'amélioration de la stratégie précédente (F0F2), l'information si le formant F1 représente la concentration de l'énergie entre 300 et 800 HZ. L'amplitude de formant F1 est utilisé pour l'activation de la deuxième paires d'électrode. Le schéma ci-dessous explique la stratégie F0F1F2 .[10]

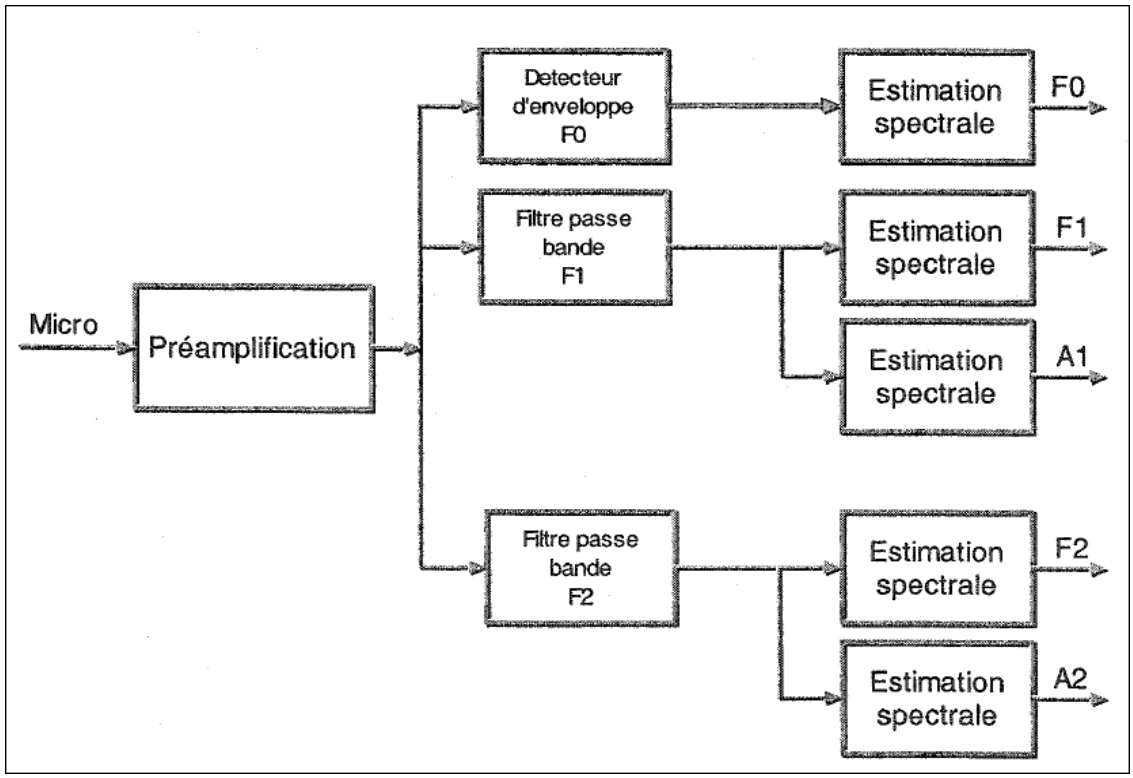


Figure I.12: Schéma bloc de la Stratégie F0F1F2**.**

La détection de l'enveloppe est réalisé grâce à un filtre « passe-bande » ou grâce à la transformation « Hilbert » qui fournit une enveloppe qui représente l'évolution temporelle de l'énergie dans la bande de fréquence du filtre.

I.11.7. Stratégie multi-pics (MPEAK)

C'est l'extension de la stratégie F0F1F2, ou en plus les formants F1 et F2, on évalue les 3 bandes d'information spectrale pour déterminer l'énergie du signal dans les hauts fréquence (2 - 2,8 Khz), (2,8 -4 Khz), (4 – 6 Khz)

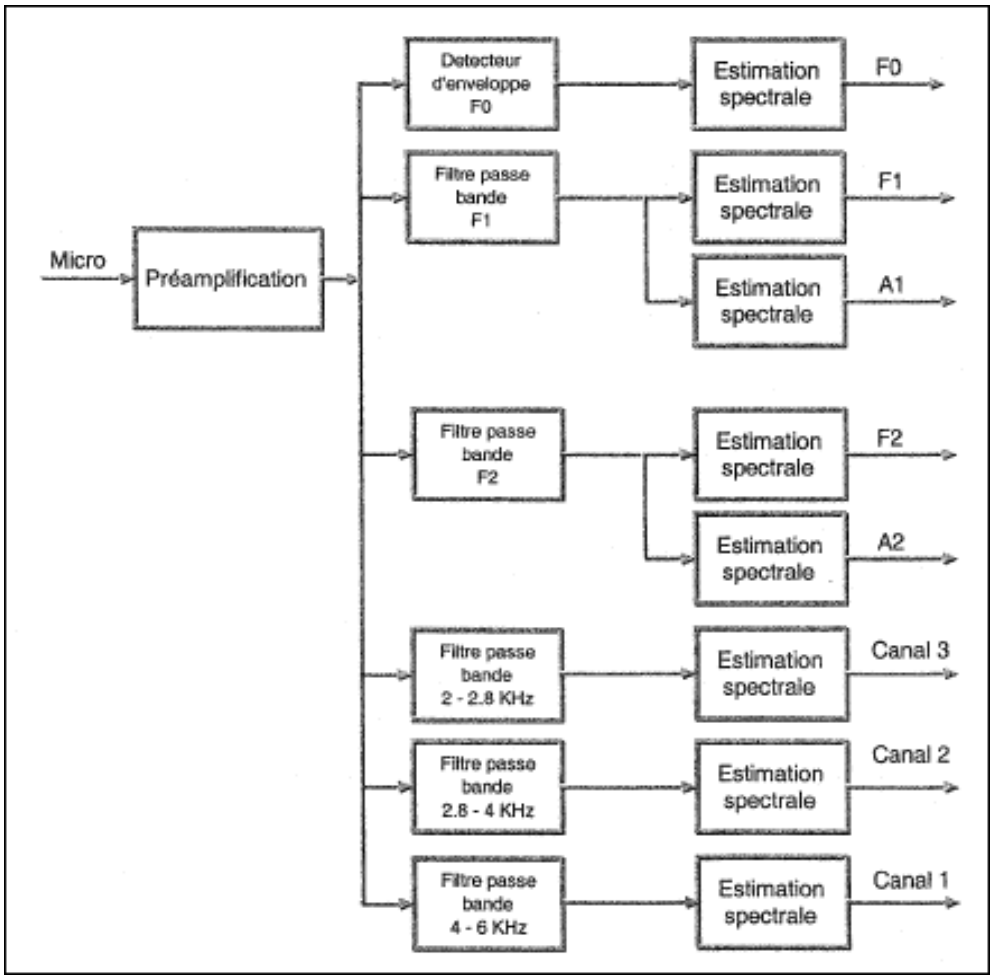


Figure I. 13: Schéma bloc de la Stratégie multi-pics (MPEAK)

Trois stimuli additionnel sont envoyés sur les électrode dans la région basale de la cochlée. Le stimulus est composé d'une sélection de 4 composant fréquentielle. La sélection

départ de la nature de signal de parole : si le son est voisé : les canaux représentant les formants F1 et F2, et les canaux de haut fréquence 2 et 3 seront stimulés, et canal 1 ne stimule pas car il y a peu d'énergie dans le spectre des sons voisé. Si le son est non voisé : les stimulation se font sur les canaux 1, 2 et 3 et sont déterminé par F2, et canal 1 ne stimule pas car il ya peu d'énergie dans le spectre des sons non voisé en dessous de 1KHZ [10] .

I.11.8. Stratégie SAS (Simultaneous Analog Stimulation) :

Cette stratégie est similaire a la stratégie CA tel que : le signal acoustique est subit de 8 filtre passe band qui résulte 8 bande de fréquence, les sorties de ces filtres sont compressé et transmis simultanément sur 8 paires d'électrode a un taux de 13000 imputions par seconde [10]

I.12. Les implants cochléaires commercialisés :

Chaque stratégie de stimulation doit être implémenté sur un système portable, ce dernier assure une bonne exécution de stratégie quel que soit le point de vue (algorithme de traitement de signal ou de génération de stimule dans l'oreille interne). [10]

- **Ineraid**

C'est le premier système qui a expérimenté la stratégie CA, ce système utilise 6 électrode intra cochléaire activé en mode mono polaire, et 2 électrodes pour le retour de la charge. La stratégie CA utilise 4 parmi les 6 électrodes. [10]

- **Nucleus**

Il a été développer à l'université Melbourne et expérimenté la première fois en 1978, le système Nucleus et le plus commercialisé actuellement. Il a passé par 3 génération : la première génération, cette prothèse base sur une seule source de courant, et contient 10 électrodes et un analyseur de parole désigné WSP-III (wearable speech processor), il est utilisé dans la stratégie F0F2 en estiment les fréquences à l'aide de détecteur de passage par zéro. La deuxième génération a été utilisé dans la stratégie F0F1F2. Ensuite le développement de l'analyseur analogique/numérique appelé MSP (Mini Speech Processor) a été utilisé dans la stratégie MPEAK, dans ce système le détecteur de passage a zéro qui estime les fréquences a été remplacé par une fonction de détection. La troisième génération utilise un système

complètement digital, il utilise un nouveau stimulateur contient 3 source de courant et qu'il est capable de généré une stimulation quasi-simultané sur 3 paires d'électrodes. [10]

- **Clarion :**

Il est le deuxième système le plus commercialisé et produit par la compagnie ABC (Advanced Bionic Coporation), il utilise 16 électrodes et permet une stimulation simultané et séquentielle, ce système est utilisé dans les stratégie CIS et SAS. [10]

- **Combi 40 :**

Cette prothèse est produite par la compagnie MED-EL, il contient 24 électrode regroupé en 12 paires d'électrode, l'analyseur de parole utilisé est conçus autour de processeur DSP56002 de Motorola, il comporte un étage analogique qui contient l'amplification, le contrôle du gain et le filtrage du signal, ensuite l'étage de la conversation analogique/numérique, le processeur utilisé est programmé pour l'exécution e la stratégie CIS.

- **Prothèse Laura :**

Il a été développé à l'université de l'Antweep en Belgique en 1980 et fabriqué et commercialisé par la compagnie PHI (Philips Hearing Implants) ensuite par la compagnie ABS (Antweep Bionic Systems), le dernier produit est appelé Laura Flex, contient 8 paires d'électrodes bipolaire. L'analyseur de parole est autour de 2 processeurs qui sont : le DSP et ADS2173 pour le traitement de la parole et d'un microcontrôleur. Ce système est utilisé dans la stratégie CIS. [10]

- **Prothèse Digisonic :**

La première version est appelée Chorimac produit par deux équipes de recherche français en 1984, elle est comporte 12 électrodes. Et la nouvelle prothèse appelé Digisonic produit par la compagnie MXM, elle comporte 15 électrodes et muni d'un analyseur qui utilise un algorithme basé sur la FFT dans le traitement de parole, cette analyseur extrait 64 valeurs d'énergie sur une bande de fréquence de 122 Hz dans la fenêtre de 100 à 7800 Hz. Ces valeurs extrait sont ensuite distribué sur 15 bandes de fréquence ensuite sont envoyé sur 15 électrodes en commençant par l'électrode qui a une haute fréquence. Les impulsions et les amplitudes sont représenter suivant les niveaux d'énergie.

En 1997 l'équipe Chouard evalut le système MXM en utilisant l'algorithme AIS (Asynchronous Interleaved Stimulation), cette dernière utilise plusieurs FFT de diffèrent largeurs sur le même segment de parole. [10]

I.13. Conclusion :

Dans ce chapitre nous avons parlé sur les composant de l'oreille humaine qui est composer de 3 niveaux et qui sont : l'oreille interne, l'oreille moyenne, et l'oreille interne, nous avons parlé ensuite sur le détail du fonctionnement de chaque niveau. Nous avons aussi décrit la problématique de surdité neurosensorielle, qui est une maladie dû d'un problème au niveau de l'oreille interne et exactement dans la cochlée, ce problème causé par le manque ou la diminution des cellules cilié qui sont une moyenne de travers des électrons, ces électrons sont produits de mouvement des cils qui est causé par le mouvement de liquide.

Et pour résoudre ce problème, les chercheurs sont produits une appareille qui compense l'oreille externe et l'oreille moyenne appelée « implant cochléaire ». Cette appareille compose de deux composant : un composant interne qui est placé chirurgicalement, et qui contient un fuseau d'électrode chaque électrode a une fréquence spécifique et une antenne de réception qui reçoive les signale émet par l'antenne de composant externe. Et un composant externe qui est placé derrière l'oreille, et qui est composer d'un microphone qui capte les signaux, un processeur qui traite ces signaux en appelant à une stratégie de codage spécifique comme : ACE, CIS, MPEAK…etc., et une antenne qui envoie les signaux traités vers l'antenne de composant interne.

Chapitre II : Réseau neurones artificiels

Chapitre II : Réseau neurones artificiels

II.1.Introduction

Dans ce chapitre nous allons parler de l'émergence du Machine Learning et du Deep Learning, les raisons de leur apparition ainsi que leur impact, puis on verra un exemple de réseau de neurones tout en illustrant son fonctionnement interne.

II.2.Intelligence artificielle et les réseaux de neurones

L'intelligence artificielle **(IA)** : c'est un domaine de l'information dont l'objectif et de créer des programmes qui exécutent des taches normalement assignées à l'intelligence humaine et de la simuler.

L'apprentissage automatique ou Machine Learning **(ML)** est une sous-section du domaine de l'intelligence artificielle ou **IA** en informatique, qui vise à apprendre aux machines à effectuer une tâche sans être explicitement programmé, et cela en utilisant un des algorithmes qu'on appellera modèles et de données.

Le concept d'intelligence artificielle est apparu dans les années 50 dans une assemblée rassemblant toute une flopée de savants célèbres en informatique et en mathématique dont **Alan Turing.** Ce savant de génie a aussi prédit le développement du Machine Learning tel qu'on le connaît.

Le ML a refait surface entre les années 70 à 80, l'idée derrière le concept était de créer des algorithmes ayant la capacité d'accumuler de l'expérience, et de la connaissance à partir de données sans être explicitement programmé pour effectuer cette tâche, et c'est vers la fin des années 80 qu'on a le retour des réseaux de neurones (créé plutôt 1943 par Walter Pitts et **Warren McCulloch** inventeurs du premier **Perceptron** qu'on appellera par la suite **Neurone artificiel.** L'idée était de reproduire un **neurone biologique** d'un cerveau humain de façon artificielle en utilisant d'opérations mathématiques, malheureusement cette approche fut très limitée dans la résolution des problèmes.

Vers la fin des années 80, on a vu le retour des réseaux de neurones, avec une réinvention de l'algorithme de **Rétropropagation** (**Backpropagation**), mais sans succès car le domaine sera finalement laissé à l'abandon faute de capacité de calcul et de stockage suffisantes.

Il fallait attendre le milieu des années 2000 pour le grand retour des réseaux de neurones avec le **Deep Learning**.

II.3.Historique sur les réseaux de neurones :

Le champ des réseaux neuronaux va démarrer par la présentation en 1943 par W. MCCulloch et W. Pitts du **neurone formel** qui est une abstraction du neurone physiologique. Le retentissement va être énorme. Par cette présentation, ils veulent démontrer que le cerveau est équivalent à une machine de Turing, la pensée devient alors purement des mécanismes matériels et logiques. Il déclara en 1955 "Plus nous apprenons de choses au sujet des organismes, plus nous sommes amenés à conclure qu'ils ne sont pas simplement analogues aux machines, mais qu'ils sont machine." **Mysterium Iniquitatis of Sinful Man Aspiring into the Place of God, repris in Embodiments of mind**. La démonstration de McCulloch et Pitts sera un des facteurs importants de la création de la cybernétique.

- En 1949, D. Hebb présente dans son ouvrage **"The Organization of Behavior"** une règle d'apprentissage. De nombreux modèles de réseaux aujourd'hui s'inpirent encore de la règle de Hebb.
- En 1958, F. Rosenblatt développe le modèle du Perceptron. C'est un réseau de neurones inspiré du système visuel. Il posséde deux couches de neurones : une couche de perception et une couche lié à la prise de décision. C'est le premier système artificiel capable d'apprendre par expérience
- Dans la même période, Le modèle de L'Adaline (ADAptive LINar Element) a été présenté par B. Widrow, chercheur américain à Stanford. Ce modèe sera par la suite le modèle de base des réseaux multi-couches.
- En 1969, M. Minsky et S. Papert publient une critique des propriétés du Perceptron. Cela va avoir une grande incidence sur la recherche dans ce domaine. Elle va fortement diminuer jusqu'en 1972, où T. Kohonen présente ses travaux sur les mémoires associatives. et propose des applications à la reconnaissance de formes.
- C'est en 1982 que J. Hopfield présente son étude d'un réseau complétement rebouclé, dont il analyse la dynamique.
- Aujourd'hui, les réseaux neuronaux sont utilisés dans de nombreux domaines (entre autres, vie artificielle et intelligence artificielle) à cause de leur propriété en particulier, leur capacité d'apprentissage, et qu'ils soient des systèmes dynamiques

[12]

II.4. Généralités sur les réseaux de neurones :

II.4.1. Neurone biologique :

Dans le cerveau humain, le neurone est l'élément de base. Il reçoit des signaux en provenance de neurones voisins, les traite, engendre, conduit et transmet l'influx nerveux à d'autres neurones. La **figure II.1** représente les éléments d'un neurone biologique

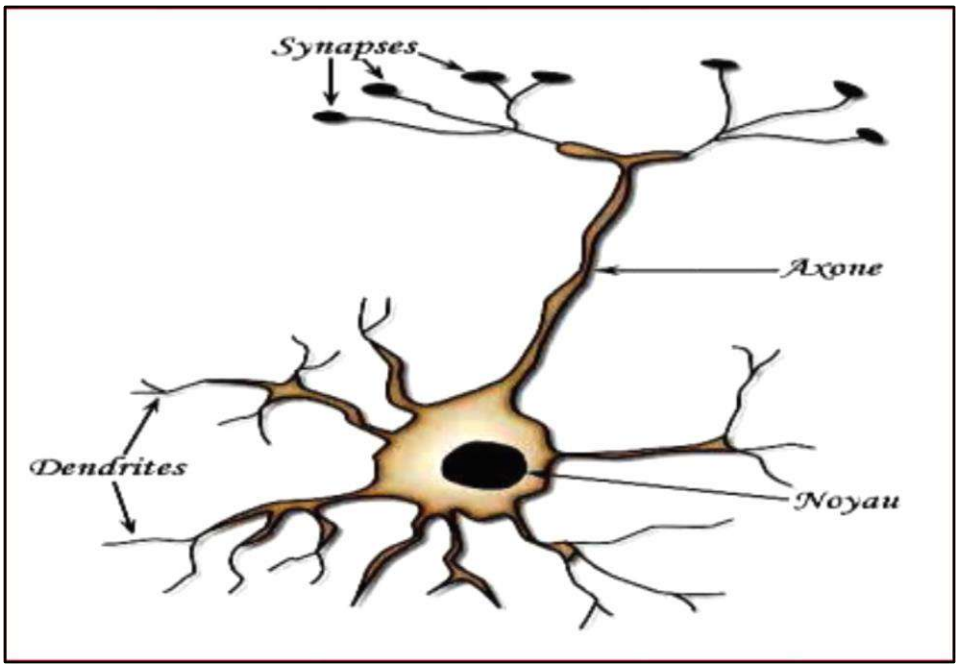


Figure II. 1 : représente les élément d'un neurone biologique.

Le neurone est constitué de :

- **Les dendrites** : ce sont les récepteurs principaux du neurone pour capter les signaux qui lui parviennent.
- **Le corps cellulaire :** Il contient le noyau. C'est un sommateur à seuil. Il effectue une sommation des influx nerveux transmis par ses dendrites. Si la somme est supérieure au seuil, le neurone répond par un influx nerveux ou potentiel d'action qui se propage le long de son axone. Si la somme est inférieure au seuil, il reste inactif.
- **L'axone :** Il sert de moyen de transport pour les signaux émis par le neurone. Il se ramifie à son extrémité, là où il communique avec d'autres neurones.

- **Les synapses :** Ils permettent aux cellules nerveuses de communiquer entre elles. Les synapses se rencontrent surtout entre les axones et les dendrites. [13]

II.4.2. Réseau formel (neurone artificiel)

Le neurone formel est une modélisation mathématique qui reprend les principes de fonctionnement du neurone biologique. Le premier neurone formel est proposé en 1943 par McCulloch et Pitts.

Le neurone représenté par la **Figure II.2** se compose d'une cellule possédant plusieurs entrées et une sortie. [14]

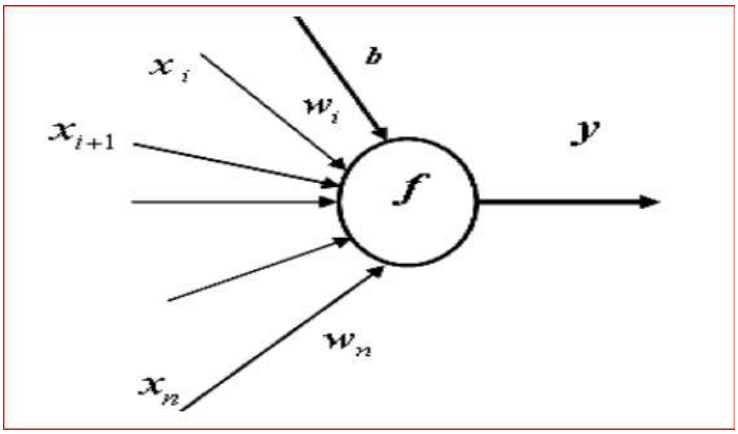


Figure II. 2: Schéma d'un neurone formel .

y : La sortie du neurone.

f : La fonction d'activation (ou de transfert).

x : Entrée du neurone.

W : Poids synaptique du neurone.

B : Biais

II.5. Principe du neurone artificiel :

Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones en amont ou des capteurs composant la machine dont il fait partie. A chacune de ses entrées est associé un poids représentatif de la force de la connexion.

Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones en aval. A chaque connexion est associé un poids.

II.5.1. Modèle d'un neurone artificiel

En 1943, McCulloch et Pitts ont proposé le premier modèle de base d'un neurone artificiel. Bien que très simple, il constitue l'élément de base des réseaux de neurones.
La **figure II.3** montre un modèle de neurone.

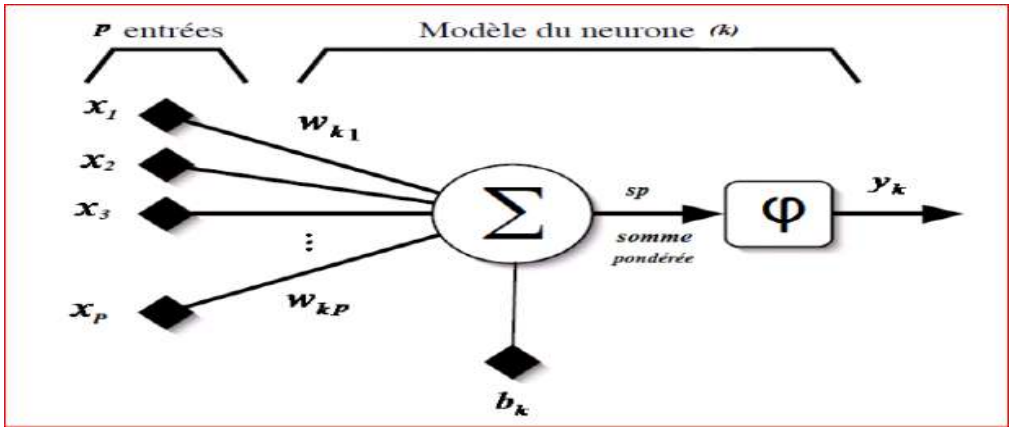


Figure II. 3 : Modèle d'un neurone élémentaire.

Les entrées au neurone sont notées par « xi », II a un ensemble de synapses caractérisées par un poids synaptique qui indique la force du lien entre l'entrée et le neurone. Ceux-ci sont notés par « *w*i » . Les signaux d'entrée sont multipliés par ceux-ci avant d'être additionnés. Un poids positif est associé à une synapse excitatrice; un poids négatif est associé à une synapse inhibitrice. Un additionneur est utilisé pour faire la somme pondérée des entrées plus un certain décalage constant de biais θ. Le résultat est ensuite passé à travers une fonction de transfert f () qu'on appelle fonction d'activation. Celle-ci limite l'amplitude de sortie du neurone dans un certain intervalle borné, usuellement de [0,1] ou bien encore de [-1,1].

Une phase d'apprentissage est alors utilisée pour ajuster ces coefficients. Soit un neurone on lui présente en entrée les signaux $\boldsymbol{X_1, X_2}$ ……$\boldsymbol{X_P}$ Ils sont respectivement multipliés par des coefficients, appelés poids synaptiques**,** $\boldsymbol{W_{K1}, W_{K2}, W_{K3}}$ …… $\boldsymbol{W_{KP}}$**.** On définit alors la somme pondéré $\mathbf{v_k}$.[15]

$$V_K = \sum_{j=1}^{p} W_{KJ} \bullet X_J \qquad \textbf{(II.1)}$$

Le neurone commence à devenir actif lorsque la sommation pondérée dépasse un seuil (ou biais) noté b_k . Il est additionné à cette grandeur, pour ensuite être transformé par la fonction d'activation notée **« ϕ»**[15]

$$y_k = \varphi\left(b_k + \sum_{j=1}^{p} W_{KJ} \bullet X_J\right) \qquad \textbf{(II.2)}$$

Pour

$$s_p = \left(b_k + \sum_{j=1}^{p} W_{KJ} \bullet X_J\right) \quad \textbf{On à :} \quad y_k = \varphi\,(SP) \qquad \textbf{(II.3)}$$

II.5.2. Fonctions de transfert :

Le choix de la fonction d'activation est important pour obtenir un modèle utile en pratique, car il influe sur les propriétés du neurone formel [16]

Différentes fonctions de transfert pouvant être utilisées comme fonction d'activation du neurone sont énumères au tableau (II.1)

Les plus utilisées sont les fonctions «lineaire » et «sigmoıde».

Tableau II.1 : tableau sur Fonction de transfert $y_k = \varphi(SP)$

Nom de la fonction	Relation d'entrée/sortie	Icône	Nom Matlab
seuil	$y_k = 0$ si $sp < 0$ $y_k = 1$ si $sp \geq 0$		hardlim
seuil symétrique	$y_k = -1$ si $sp < 0$ $y_k = 1$ si $sp \geq 0$		hardlims
linéaire	$y_k = sp$		purelin
linéaire saturée	$y_k = 0$ si $sp < 0$ $y_k = sp$ si $0 \leq sp \leq 1$ $y_k = 1$ si $sp > 1$		satlin
linéaire saturée symétrique	$y_k = -1$ si $sp < -1$ $y_k = sp$ si $-1 \leq sp \leq 1$ $y_k = 1$ si $sp > 1$		satlins
linéaire positive	$y_k = 0$ si $sp < 0$ $y_k = sp$ si $sp \geq 0$		poslin
sigmoïde	$y_k = \frac{1}{1+\exp^{-sp}}$		logsig
tangente hyperbolique	$y_k = \frac{e^{sp}-e^{-sp}}{e^{sp}+e^{-sp}}$		tansig
compétitive	$y_k = 1$ si sp maximum $y_k = 0$ autrement		compet

II.5.3. Types de topologie :

Plusieurs topologies et algorithmes d'apprentissage ont été pensés et conçus. Il existe trois principaux types d'architectures : le réseau à propagation directe à une couche, le réseau à propagation directe multicouche, le réseau récurrent .

le réseau à propagation directe, qui se caractérisent par le fait que le signal ne se propage que dans une seule direction, c'est-à-dire , du nœud d'entrée au nœud de sortie. Dans le cas d'un réseau monocouche, il n'y a qu'une seule couche de neurones en sortie, comme le montre la **figure II.4** C'est la forme la plus simple et a été utilisée pour la première fois jusqu'au milieu des années 1980.[17]

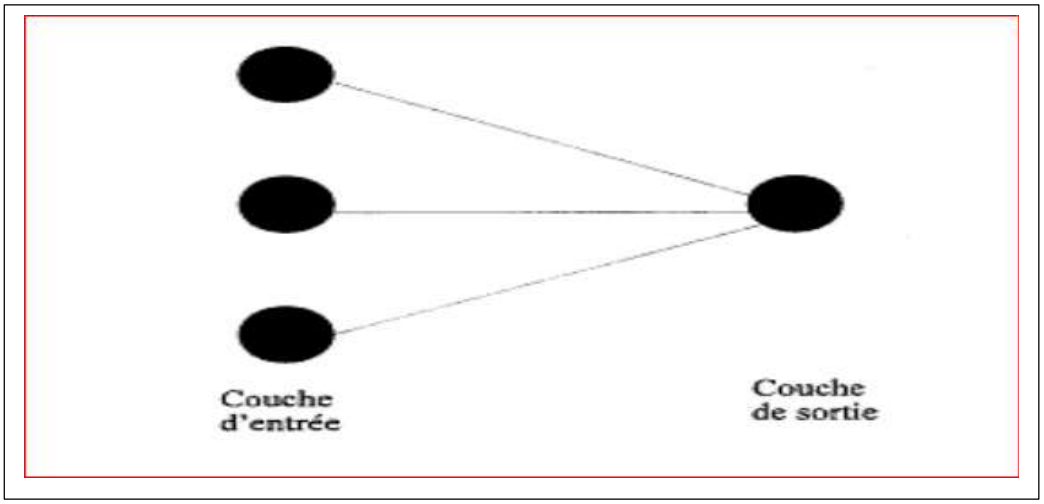


Figure II. 4 : Réseau à propagation avant à une couche.

Le deuxième type d'architecture, les réseaux à propagation directe, est de loin la configuration la plus populaire et la plus largement utilisée. Comme le montre la **figure II.5** il diffère par la présence d'une ou plusieurs couches cachées qui incluent des nœuds de calcul internes correctement nommés neurones cachés. Leur rôle est d'intervenir entre l'entrée et la sortie externes du réseau, chaque couche desservant des couches successives dans un sens. En ajoutant des couches cachées, le réseau peut extraire des statistiques d'ordre supérieur et augmenter son pouvoir de représentation. Cette architecture est devenue populaire au milieu des années 1980 lorsque Rumelhart et McClelland.[17]

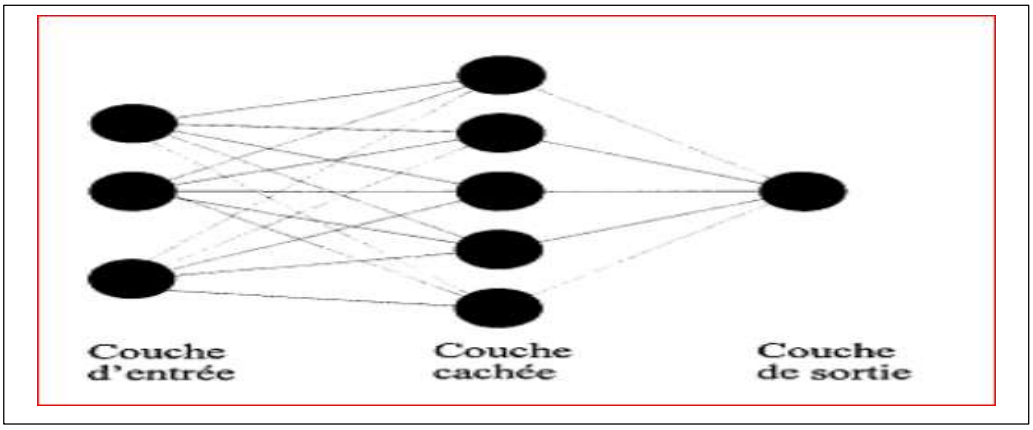


Figure II. 5: Réseau à propagation avant multicouche.

Couche d'entrée : elle reçoit les entrées du monde extérieur sur lesquelles le réseau va faire l'apprentissage, et le traitement.

Couche de sortie : elle répond aux informations durant son apprentissage de n'importe quelle tâche.

Couche cachée : elle se situe entre les couches d'entrée et de sortie ; son travail est de transformer l'entrée en information que l'unité de sortie peut utiliser. Chaque neurone caché est lié au neurone de sa couche précédente (entrée) et de la couche suivante (sortie).

La troisième classe d'architectures, soit les réseaux récurrents, se distingue des réseaux à propagation avant par le fait qu'elle contienne des boucles de rétroaction (ou liens récurrents). Une partie du signal de sortie peut revenir à l'entrée. On peut aussi imaginer de la rétroaction sur des cellules cachées comme on le voit à la **figure II.6.** [17]

Ce changement structurel a un impact profond sur la capacité d'apprentissage du réseau et son fonctionnement. De plus, les boucles de rétroaction utilisent des branches composées d'élément à retard unitaire (dénoté parz^{-1}) résultant en un comportement dynamique non linéaire. Ainsi, le réseau peut acquérir dès lors un comportement temporel proprement dit.[17]

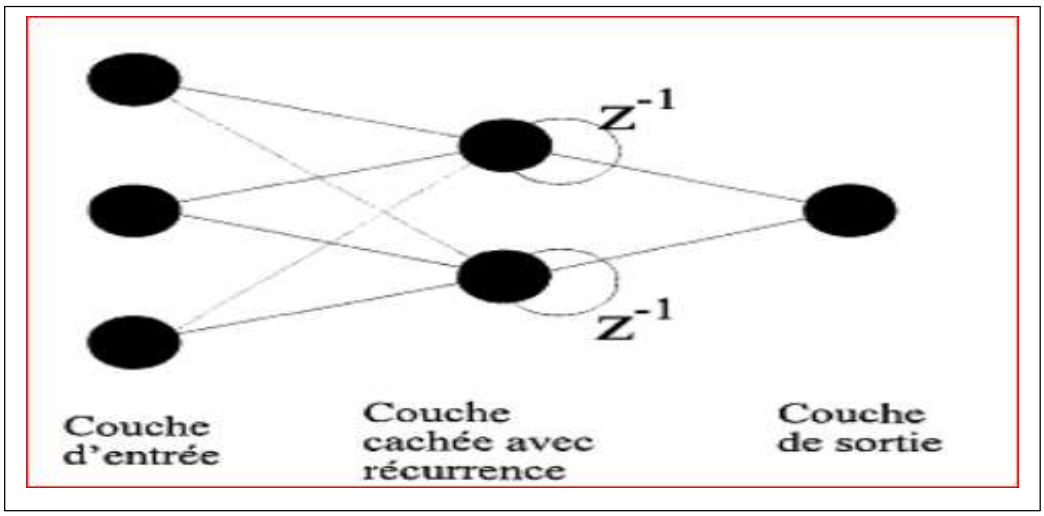


Figure II. 6 : Réseau récurrent avec neurones cachés.

II.6. Architecture des réseaux de neurones :

Selon la topologie de connexion des neurones, on peut les classer en deux grandes catégories: réseaux non bouclés (statique ou feedforward) et réseaux bouclés (dynamique, feedback ou récurrent).

II.6.1. Réseau de neurones non bouclé :

Un réseau de neurones non bouclé est représenté graphiquement par un ensemble de neurones "connectés" entre eux, l'information circulant des entrées vers les sorties sans "retour en arrière". Dans la très grande majorité des applications, les réseaux de neurones sont des formules algébriques dont les valeurs numériques sont calculées par des programmes d'ordinateurs, non des objets physiques [18]

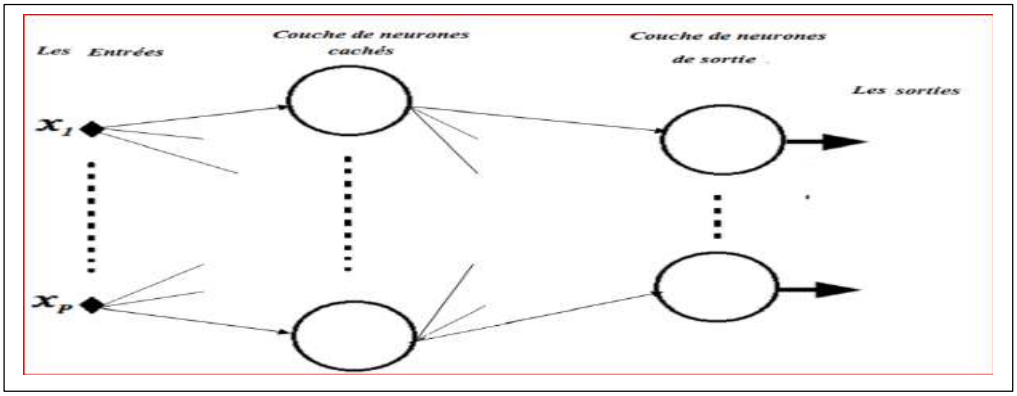


Figure II. 7 : un réseau de neurones non bouclé

Les réseaux de neurones non bouclés sont des objets statiques : si les entrées sont indépendantes du temps, les sorties le sont également. Ils sont utilisés principalement pour effectuer des tâches d'approximation de fonction non linéaire, de classification ou de modélisation de processus statiques non linéaires [18]

II.6.2. Réseau de neurones bouclé :

Les réseaux de neurones bouclés peuvent avoir une topologie de connexions quelconque, comprenant notamment des boucles qui ramènent aux entrées la valeur d'une ou plusieurs sorties.il faut évidemment qu'à toute boucle soit associé un retard : un réseau de neurones bouclé est donc un système dynamique [18]

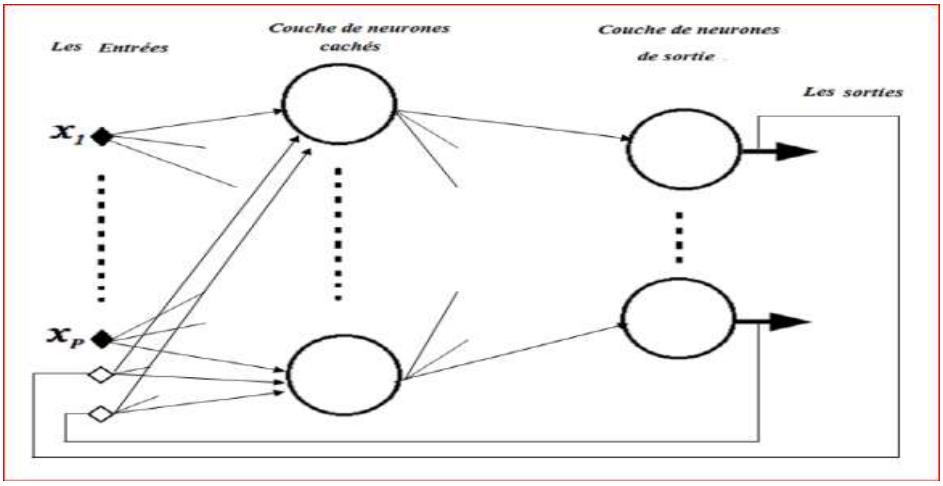


Figure II. 8 :un Réseau de neurones bouclé.

Les réseaux de neurones bouclés sont utilisés pour effectuer des tâches de modélisation de systèmes dynamiques, de commande de processus, ou de filtrage. [18]

II.7. Avantages et inconvénients des réseaux de neurones

II.7.1. Avantage

- Ils traitent des problèmes non structurés sur lesquels aucune information n'est disponible à l'avance. [19]
- Les réseaux de neurones fonctionnent sur des données incomplètes ou bruitées. Cette lacune d'information peut être complétée par l'ajout d'autres neurones à la couche cachée. [19]
- Les réseaux de neurones sont souples et génériques. Ils peuvent résoudre différents types de problèmes dont le résultat peut être : une classification, analyse de donné, etc.… [19]
- Les réseaux neuronaux se comportent bien parce que même dans des domaines très complexes, ils fonctionnent mieux que les arbres de statistique ou de décision. [20]

II.7.2. Inconvénients

Les réseaux de neurones ont aussi des inconvénients, tels que :

- La difficulté de choisir des valeurs initiales des poids de connexion ainsi qu'en cas d'erreur dans les résultats de sorties, l'utilisateur n'a aucune information sur le fonctionnement interne. [21]
- L'adaptation du pas d'apprentissage. [20]
- L'apprentissage au détriment de la généralisation. [20]
- La lenteur d'apprentissage. [21]

II.8. Le perceptron

Le perceptron est la forme la plus simple du réseau de neurones, et permet de classifier correctement des objets appartenant à deux classes linéairement séparables. Le perceptron est formé d'un seul neurone qui possède un seuil ainsi qu'un vecteur de poids synaptiques ajustables, tout comme le modèle de neurone de McCulloch & Pitts [22]

II.8.1. Perceptron multicouches

La mise en cascade de perceptrons conduit à ce qu'on appelle le perceptron multicouches. Lorsque le vecteur de caractéristiques d'un objet est présenté à l'entrée du réseau, il est communiqué à tous les neurones de la première couche. Les sorties des neurones de cette couche sont alors communiquées aux neurones de la couche suivante, et ainsi de suite. La dernière couche du réseau est appelée couche de sortie, les autres étant désignées sous le terme de couches cachées car les valeurs de sortie de leurs neurones ne sont pas accessibles de l'extérieur [22]

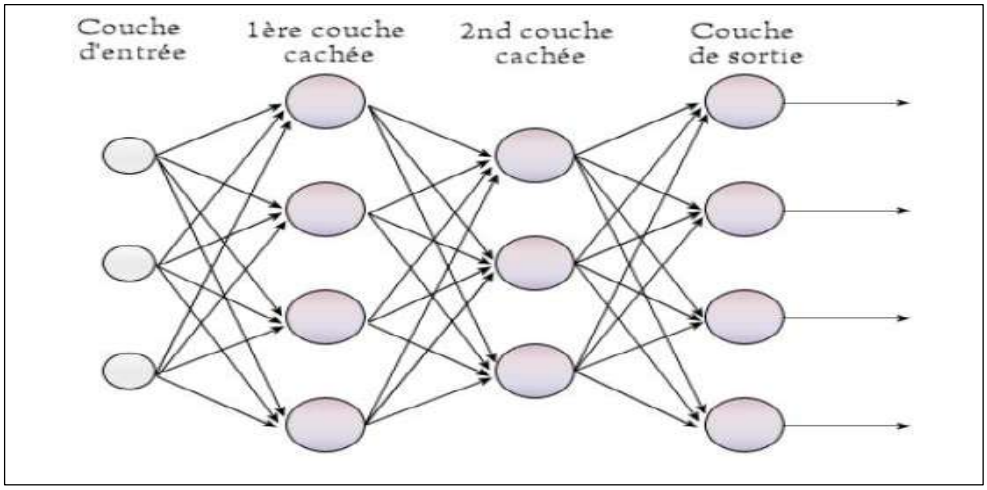


Figure II. 9 : Perceptron multicouches .

II.9. Les différents types de RNA

Il existe plusieurs types de réseaux de neurones artificiels (RNA), Selon chaque neurone et son fonctionnement en cerveau humain, ces types sont :

Feedback ANN

Dans ce type, la sortie retourne dans notre réseau afin d'obtenir des résultats meilleurs en interne ; cependant Selon le ‘’Lowell Center for AtmosphericResearch’’ de l'Université du Massachusetts, le réseau de rétroaction est correspond bien au réglage des problèmes d'optimisation et le renvoie des informations en lui-même, Les ANN de rétroaction sont utilisées par les corrections d'erreurs de notre système interne. [23]

FeedForward ANN

C'est un réseau de neurones simple composé d'une couche d'entrée, d'une couche de sortie et d'une ou plusieurs couches de neurones, il a une très bonne évaluation de sa sortie âpres examination de son entrée, nous remarquons la puissance du réseau dans le comportement de groupe des neurones connectés puis la sortie est définie. Il a comme avantage son apprentissage d'évaluation et sa reconnaissance des modèles d'entrée.[23]

Classification-Prédiction ANN

Le sous-ensemble de l'ANN à anticipation est appelé l'ANN de classification-prédiction, il est appliquée dans les explorations de données. il identifie des modèles particuliers et les classe en groupes spécifiques, après il les classent sous forme de « nouveaux modèles » pour le réseau.[23]

II.10. Définition de machine Learning (ML) :

Le machine Learning est quant à lui défini en 1959 par Arthur Samuel comme suit: « L'apprentissage automatique est la discipline donnant aux ordinateurs la capacité d'apprendre sans qu‟ils soient explicitement programmés.» [24]

L'apprentissage automatique est un sous-ensemble de l'intelligence artificielle « IA », il est axé sur la création des systèmes qui apprennent et améliorent les performances, en se

basant sur des données qu'ils traitent. Les algorithmes d'apprentissage automatique entrent en jeu pour optimiser, fluidifier, et sécuriser cette dernière [25]

II.11. Méthodes d'apprentissage :

Il existe de différents types d'apprentissage automatique. Selon la définition du type d'apprentissage est basée sur la réponse à ces deux questions suivantes :

- Est-ce que cet apprentissage compte sur la supervision humaine dans son entrainement et apprentissage ?
- Est-ce que ce type cet apprentissage utilise une base de données fournie par l'être humain ?
  - ✓ Si la réponse est oui pour les deux questions, nous avons un apprentissage supervisé.
  - ✓ Si la réponse est non pour la première question, et oui pour la deuxième question, nous parlons d'un apprentissage non supervisé.
  - ✓ Si la réponse est non pour les deux questions, le type de l'apprentissage est l'apprentissage par renforcement.[24]

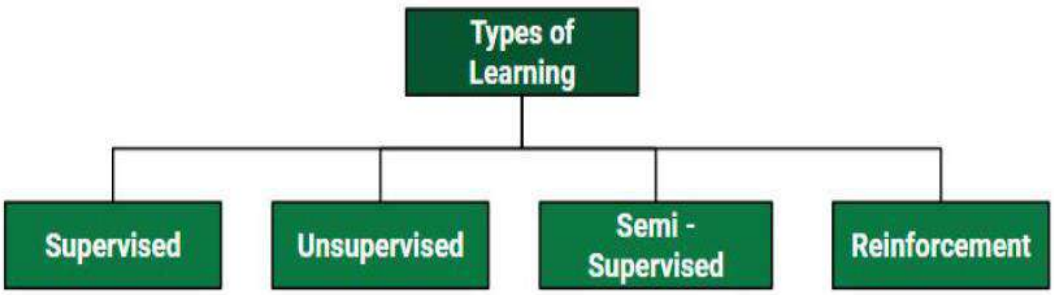


Figure II. 10 : Les Types d'apprentissage en ML .

II.11.1. Apprentissage supervisé :

Apprentissage supervisé ou « Supervised learning » prend des données qui sont étiquetées (possédant des étiquettes). Ces étiquettes nous renseignent sur le résultat souhaité, et crée un modèle à l'aide de ses données, afin d'effectuer des prédictions qui pourront généraliser sur de nouvelles données.

Dans l'apprentissage supervisé l'être humain aide l'algorithme pour apprendre. Le même cas lorsqu'on apprend un enfant pour identifier les fruits, en les mémorisant dans sa mémoire. Dans l'apprentissage supervisé, l'algorithme apprend grâce à un jeu de données déjà étiqueté et dont le résultat est prédéfini.[26]

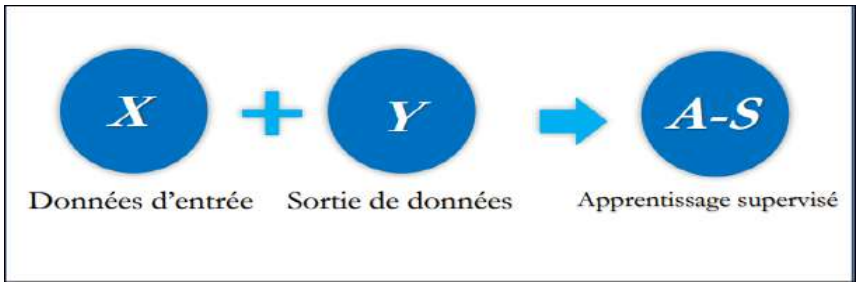


Figure II. 11 : Apprentissage supervisé .

Les problèmes dans l'apprentissage supervisé sont 2 types : **Classification**, **Régression**.

II.11.1.1. Classification

C'est une tâche d'apprentissage supervisé dont la sortie a des étiquettes définies (valeur discrète) ce qu'on appelle classes ou catégories. On peut avoir des classifications **Binaire** ou alors **Multi-classe**.

La **classification binaire** prédit pour un modèle (0 ou 1) vrai ou faux, dans le cas de **Multi-classes** le modèle va prédire plus d'une classe. Le but ici reste le même quel que soit le nombre de classes, **prédire** si les **valeurs discrètes** appartiennent à une classe particulière.

Exemple :

Gmail qui est en mesure de prédire si un mail donné est un spam ou mail normal.

II.11.1.2. Régression

Il s'agit d'une tâche d'apprentissage supervisé dont la sortie a une **valeur continue.** Le but ici est de prédire une valeur aussi proche de la valeur de sortie réelle que notre modèle le permet, puis l'évaluation est effectuée en calculant la valeur d'erreur. Plus l'erreur est petite, plus la précision de notre modèle de régression est grande.

II.11.2. Unsupervised Learning :

L'apprentissage non supervisé ou unsupervised est un type d'algorithme d'apprentissage automatique utilisé pour tirer des inférences à partir d'ensembles de données

constitués de données d'entrée sans réponses étiquetées. Ce type d'apprentissage est principalement utilisé quand on souhaite étudier un ensemble de données non labellisés.

Avec l'apprentissage non supervisé la machine n'a pas besoin de l'aide pour apprendre. Dans laquelle un ordinateur apprend à identifier des processus et des schémas complexes sans aucun guide, Il implique une formation basée sur des données sans étiquette, qui ne contiennent aucun résultat spécifique.[26]



Figure II. 12 : Apprentissage non-supervisé.

II.11.2.1. *Regroupement (Clustering) :*

Le clustering permet de séparer les données entrées en un ensemble ou groupe de données qui ont des traits similaires et de les affecter à un cluster. Contrairement à la classification dans l'apprentissage supervisé ces différents clusters ou groupe ne sont pas connus à l'avance, c'est l'algorithme lui-même qui va séparer les données aux nombres de clusters qu'il faut. Comme l'on peut le voir dans l'exemple **Figure II. 13**, les points de jeu de données donnés ont été divisés en groupes identifiables par les couleurs rouge, vert et bleu.[27]

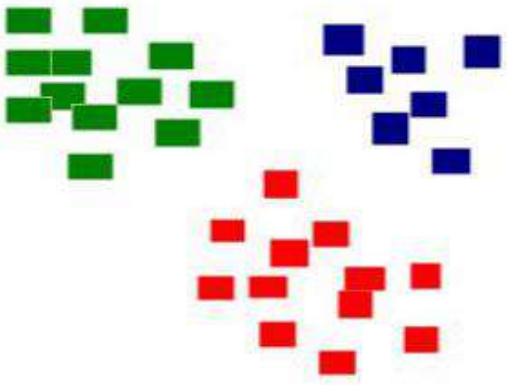


Figure II. 13: Regroupement (Clustering).

II.11.2.2. Réduction de la dimension

L'objectif est de simplifier les données sans perdre trop d'informations, à titre d'exemple, fusionner plusieurs caractéristiques en un seul caractère.[27]

II.11.3. Différence entre apprentissage supervisé et non supervisé

La majeure différence qu'on peut trouver entre ces 2 types d'apprentissages est la disposition des données d'entrée, mais il existe tout de même d'autres différences qu'on peut trouver dans le tableau ci-dessous.[27]

Ce tableau résume la différence entre apprentissage supervisé et non supervisé :

Tableau II.2 : la Différence entre apprentissage supervisé et non supervisé

	Supervised Learning	**Unsupervised Learning**
Les données d'entrée	Utilise des données connues et étiquetées en tant qu'entrée	Utilise des données inconnues en tant qu'entrée Non étiquetées
Nombre de Classes	Connues	Inconnues
Précision du résultat	Résultats précis et fiables	précision et fiabilité **Modéré**

II.12. Apprentissage semi-supervisé

Nous avons préalablement vue l'apprentissage supervisé et non supervisé, dont la majeure différence réside dans le fait que les données soient étiquetées ou non, et à cela s'ajoute les méthodes adéquates utilisées pour traiter ses données.[27]

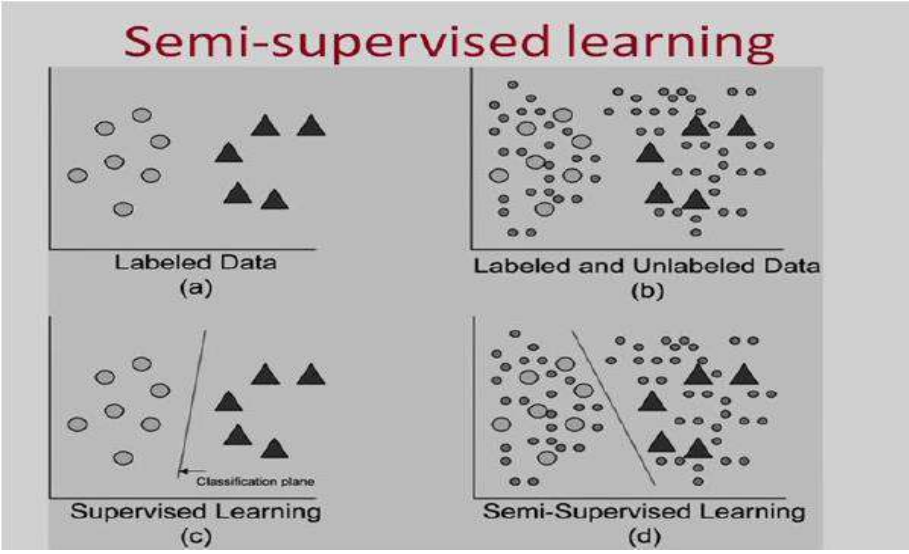


Figure II. 14 : Apprentissage semi-supervisé.

L'apprentissage semi-supervisé regroupe ses deux principes, il prend un ensemble réduit de données étiquetées avec un autre ensemble de données non étiquetées du même type. L'avantage de ce type d'apprentissage réside principalement dans le processus d'étiquetage des données prend beaucoup de temps et souvent coûteux. Donc paradoxalement le non étiquetage devient bénéfique pour le processus d'apprentissage, et la construction du modèle et moins coûteuse.[27]

II.13. Apprentissage par Renforcement

Entraînement par renforcement est une technique dans laquelle on immerge un agent dans un environnement où celui-ci interagit avec son environnement dans le but d'apprendre.

Un agent dans un environnement est muni de **sensors** pour capter les informations de son environnement, ainsi que d'actionneurs qui lui permettront d'agir dans son environnement. Les agents observent l'entrée, puis il effectue une action en prenant des décisions.

Une fois l'action réalisées, l'agent reçoit des récompenses en conséquence, ce qui renforce le modèle en stockant ses informations dans une base de données. La récompense peut être positive, négative ou nulle selon les actions effectuées.[26]

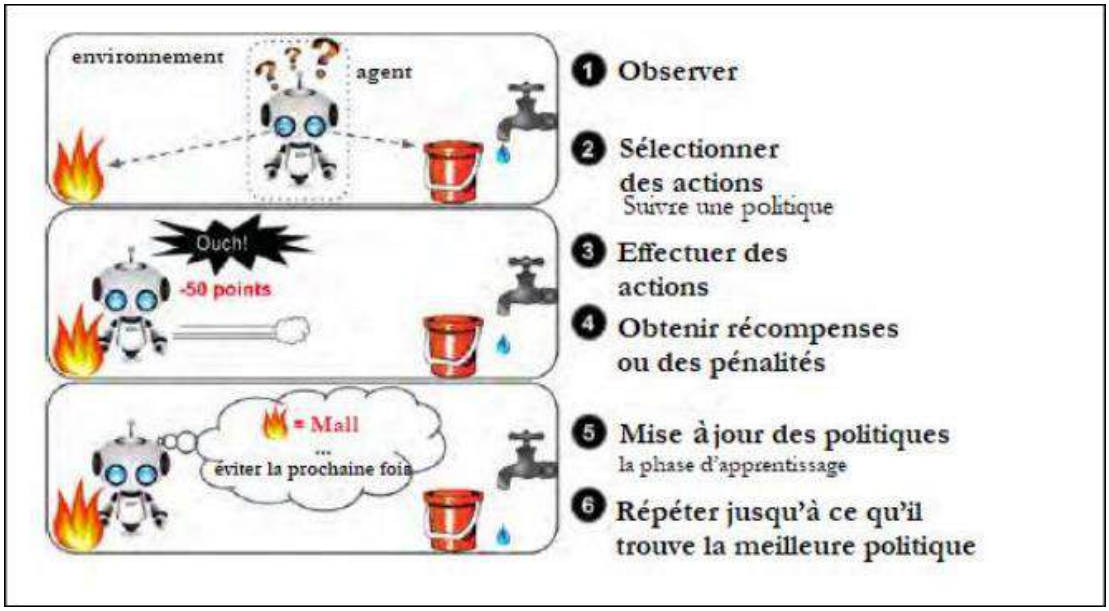


Figure II. 15 : Concept d'apprentissage par renforcement

Exemple : robot domestique ou encore IA dans les jeux vidéos

II.14. Le choix d'un type d'apprentissage automatique

Le choix du type d'apprentissage dans un problème donné est fonction des points suivants :

- La taille, la qualité et la nature des données.
- Le temps de calcul disponible.
- L'urgence de la tâche.
- Le but d'utilisation de ces données [26]

II.15. Deep Learning

Deep Learning ou apprentissage en profondeur ou **DL** est une branche du Machine Learning entièrement basée sur des **réseaux de neurones artificiels.**

Le concept d'apprentissage en profondeur existe depuis plusieurs années, mais il a été laissé à l'abandon faute de moyens nécessaires.

Dans le milieu des années 2000 le Machine Learning fait rage dans les compétitions de reconnaissance visuelle, en 2012 **deep mind** une startup dans le domaine de l'IA arrive dans la compétition avec un algorithme de **deep learning** qui bat largement tous les autres compétiteurs, l'année suivante tous les compétiteurs se sont tournés vers le deep learning au vu des résultats obtenus.

L'avancée du DL est dû à l'augmentation en exponentiel qu'ont connu les machines en capacités de calculs, et de stockages**,** ainsi que la disponibilité de données de masses **(big**

data), ses 3 ingrédients étaient nécessaires pour exploiter le potentiel du DL qui fût chose impossible dans les années 90.

Les pionniers qui ont soutenus le DL tel que **Geoffrey Hinton,** ou alors **Yoshua Bengio** qui a développé les réseaux **GAN (generative** adversal networks), **Yann leCun** qui est au cœur d'une avancée fulgurante dans le domaine de reconnaissance d'images avec les réseaux Convolutionnels **CNN** voir section**,** et son architecture **LetNet**. **Geoffrey Hinton** a prouvé que l'apprentissage profond pouvait résoudre des problèmes insolubles par d'autres approches.

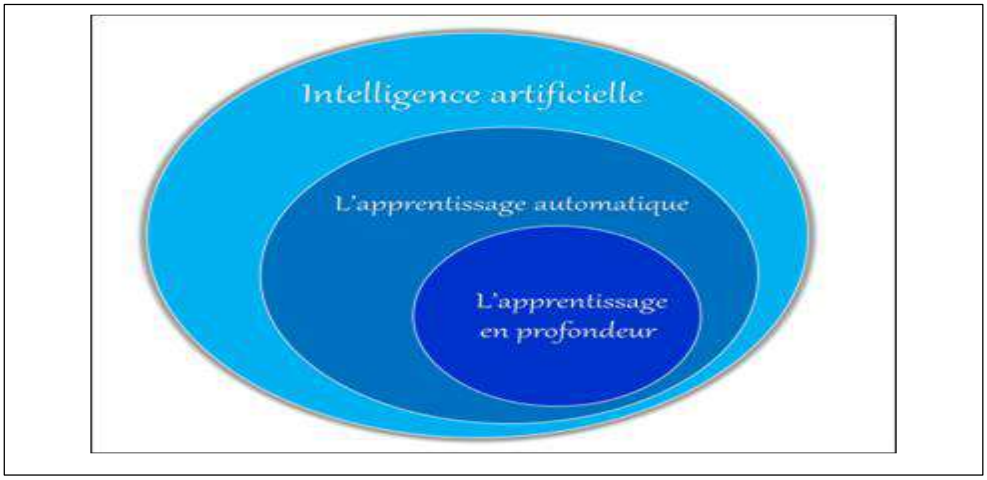


Figure II. 16 : Les sous-bronches de l'intelligence artificiel.

II.15.1. Architectures des modèles de deep learning

Il existe plusieurs types de réseaux de neurones profonds. Parmi les modèles les plus utilisés, on peut citer :

- Réseaux neuronaux convolutionnels (CNN).
- Réseaux neuronaux récurrents (RNN).

II.15.1.1. Les réseaux de neurones convolutifs (Convolutional Neural Networks, ConvNets ou CNN)

Les réseaux neuronaux convolutionnels CNNs sont un type spécialisé de réseau neuronal pour le traitement de données qui a une topologie connue, ils sont des réseaux supervisé et non supervisé, cela dépend de l'exigence et l'utilisation, cependant, principalement sont supervisé

L'architecture de ce réseau **figure II.16** est inspirée à partir de l'arrangement des neurones situés dans la région du cortex visuel des animaux. Cet arrangement correspond au chevauchement des régions lors du pavage du champ visuel Le fonctionnement de CNNs est inspiré à partir du fonctionnement du processus biologique, ils consistent en un empilage multicouche de perceptrons, dont le but est le prétraitement de petites quantités d'informations.[28]

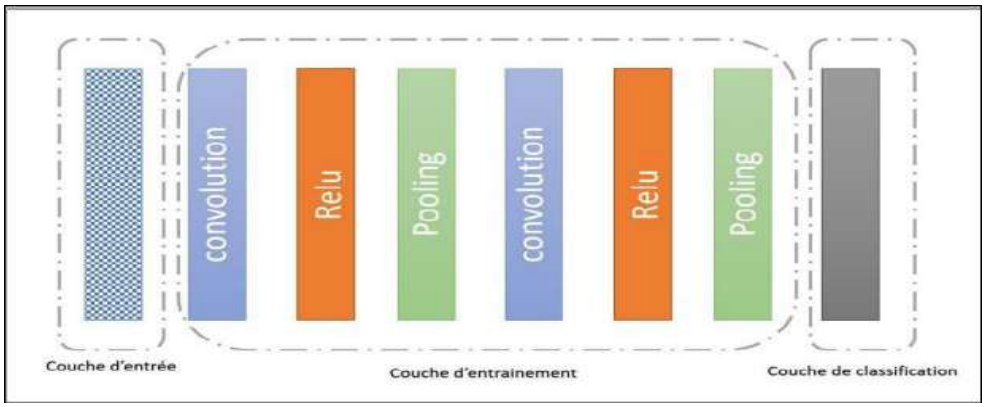


Figure II.17: L'architecture des réseaux de neurones convolutifs.

L' architecture des réseaux neuronaux convolutifs (CNNs) compose de trois majors couches :

- Couche d'entrée : la couche d'entrée accepte généralement l'entrée tridimensionnelle sous la forme (hauteur x largeur) de l'image et a une profondeur représentant les canaux de couleur (généralement trois pour les canaux de couleur RGB).
- Couche d'entrainement : Couche d'entrainement est construite de :

✓ Convolution couche : qui fait passer les images d'entrée par un ensemble de filtres convolutifs, chacun de ces filtres activant certaines caractéristiques des images ;

✓ la couche ReLU (Rectified linear unit) permet d'accélérer et d'optimiser l'apprentissage en remplaçant toutes les valeurs négatives reçues en entrées par des zéros. Elle joue le rôle de fonction d'activation ;

✓ la couche de pooling qui consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes.

- Couche de classification : Couche de classification produire des probabilités ou des scores de classe .[28]

II.15.1.2. *Réseaux neuronaux récurrents (RNN*)

les réseaux de neurones récurrents ou récurrent neural network (RNN) en anglais. Ils ont également été créés dans les années 1980. il existe non seulement des liens d'une couche vers les suivantes, mais aussi vers les couches précédentes. Ces architectures sont particulièrement adaptées au traitement de signaux séquentiels, tels que le texte. En théorie, les RNN peuvent utiliser des informations dans des séquences arbitrairement longues, mais dans la pratique, on les limite à regarder seulement quelques étapes en arrière.[28]

II.15.1.3. Autoencoder

Auto-encodeurs : les auto-encodeurs sont des algorithmes d'apprentissage non supervisés basés sur des réseaux de neurones artificiels qui peuvent créer de nouvelles représentations d'ensembles de données. Typiquement, celui-ci est plus compact et présente moins de descripteurs, ce qui permet de réduire la dimensionnalité du jeu de données. L'architecture d'un auto-encodeur se compose de deux parties : un encodeur et un décodeur. .[32]

Un encodeur est constitué d'un ensemble de couches de neurones qui traitent des données pour construire de nouvelles représentations dites "encodées". À leur tour, des couches de neurones dans le décodeur reçoivent ces représentations et les traitent pour tenter de reconstruire les données d'origine. La différence entre les données reconstruites et les données d'origine permet de mesurer les erreurs commises par l'auto-encodeur. L'apprentissage consiste à modifier les paramètres de l'auto-encodeur pour réduire l'erreur de reconstruction mesurée sur différents exemples du jeu de données. .[32]

L'architecture la plus simple d'un auto-encodeur est similaire à un perceptron multicouche. Cependant, selon les données traitées, différentes topologies de réseaux de neurones peuvent être utilisées. Par exemple, des couches convolutives pour l'analyse d'images ou des couches neuronales récurrentes pour le traitement de séries temporelles ou de séquences. .[32]

Notez que contrairement à un grand nombre de réseaux de neurones, les auto-encodeurs peuvent être entraînés de manière non supervisée, ce qui permet d'appliquer ces méthodes à des ensembles de données non annotés.[32]

- **Principe d'un autoncoder**

Ce principe d'auto encodeur a été généralisé à des tâches plus variées, en particulier dans les tâches de débruitage (on parle de denoising AE en anglais). Après avoir appris comment représenter de manière optimale les entrées, en fournissant à l'auto encodeur des entrées bruitées, il est capable de donner en sortie des données débruitées qui sont très proches de l'entrée originale.

- **Architecture d'un autoencoder**

Un autoencoder a une architecture très spécifique, car les couches cachées sont plus petites que les couches d'entrée. On appelle ce type d'architecture une architecture « bottleneck ». On peut décomposer un auto encodeur en deux parties à gauche et à droite de ce « bottleneck».

La partie gauche s'appelle l'encodeur. L'encodeur transforme l'entrée en une représentation dans un espace de dimension plus faible appelé espace latent. L'encodeur compresse donc l'entrée dans une représentation moins coûteuse.

La partie droite est appelée décodeur, car elle doit reconstruire à l'aide de la représentation latente de l'entrée, une sortie la plus fidèle à l'entrée .

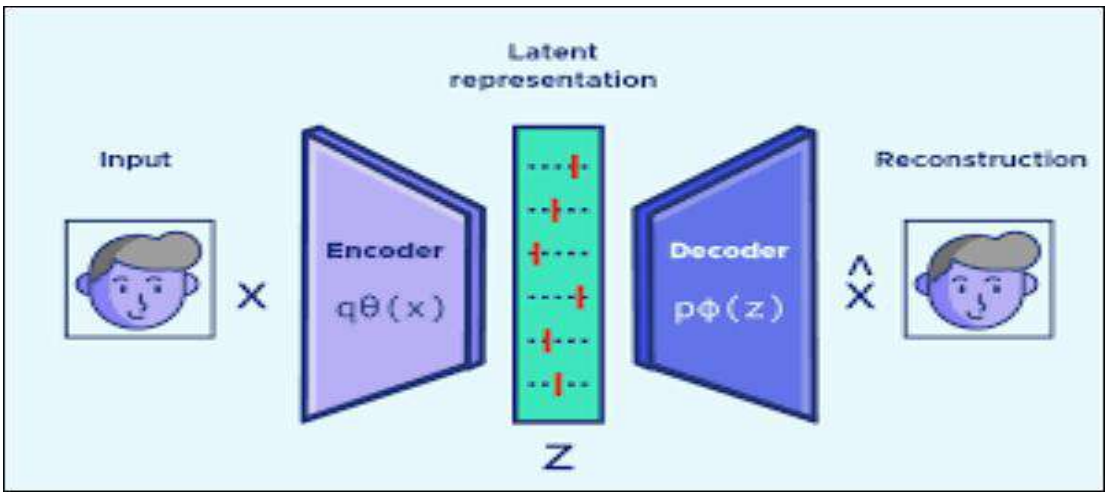


Figure II.18 :schéma représenté l'architecture d'un autoencoder.

II.16. Applications du Deep Learning :

- Classification des images
- Reconnaissance des actions humaines
- Reconnaissance des gestes de la main
- Analyse des scènes [29]

II.17. Domaine du système deep Learning :

il existe plusieurs domaines des systèmes utilisant le deep learning

- les télécommunications,
- les finances,
- la sécurité,
- les réseaux sociaux,
- les assurances et l'industrie automobile….[29]

II.18. Machine Learning Vs Deep Learning:

La majeure différence qu'on note entre ses 2 concepts provient de la manière dont les données sont présentées au système (modèle).

- Les algorithmes de ML nécessitent presque toujours des données structurées, alors que les réseaux d'apprentissage approfondis reposent sur des couches de réseaux de neurones artificiels (RNA).
- On voit aussi une différence au sein de l'architecture des modèles qui les composent, on note que les modèles type DL sont plus profond que les modèles type ML.
- Deep learning n'utilise que les réseaux de neurones, alors que pour le ML les réseaux de neurones sont qu'une approche de conception des modèles parmis tant d'autres.

En considérant le fait que le DL est la prochaine étape de l'évolution du ML inculquant aux machines la manière de prendre leurs décisions de façon précise sans l'intervention de l'expert humain.[27]

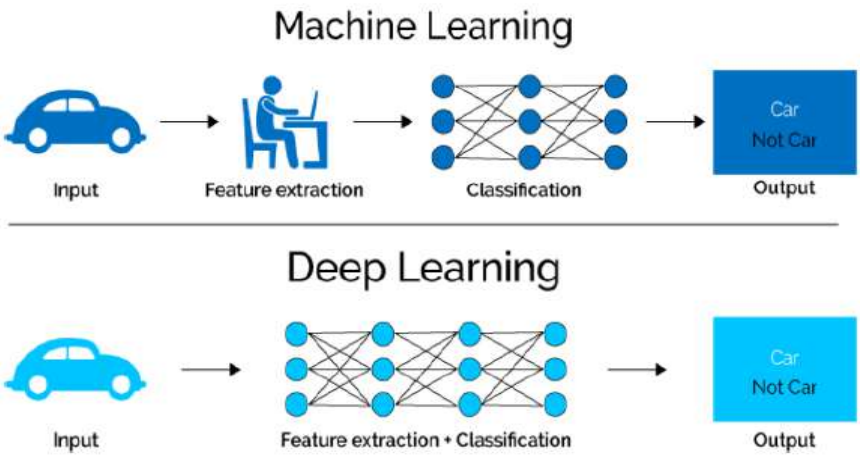


Figure II. 19 : ML vs DL.

II.19. Pourquoi Deep learning:

Il s'agit d'une combinaison de facteurs dont :

- L'**omniprésence des données** : Nous somme dans l'ère de l'informatisation (Internet Of Things), et le propre du Deep-Learning est de tirer parti d'une grande quantité de données pour en estimer une représentation abstraite et en tirer parti.
- La **puissance de calcul :** La théorie des réseaux de neurones existe depuis quelques décennies, mais c'est grâce à la puissance de calcul accessible aujourd'hui qui se démocratise, notamment depuis que les GPUs sont devenus la plateforme de choix pour le Deep-Learning.
- Des **besoins croissants dans le domaine de l'IA** : vision par ordinateur, reconnaissance vocale, traitement du langage, …etc.
- Un **effet de mode**. On a tendance à vouloir appliquer le Deep-Learning partout alors que ça reste un moyen et non une fin. Certains problèmes sont tout à fait solubles par d'autres méthodes d'apprentissage statistique. Cela dit, si beaucoup de gens sont prêts à investir dans le Deep-Learning, il est normal qui deviennent si populaire.
- **Capacité de Stockage :** qui sont devenu beaucoup plus accessible à prix raisonnable.

- Apparition de plateformes et communautés forte encouragent l'évolution de ce domaine, ainsi que sa démocratisation.

II.20. Traitement des données :

Les données sont le nerf de la guerre surtout quand il s'agit de Machine Learning, car ces dernières occupent une importance capitale pour construire un bon modèle.

Avant de découper ou fragmenter les données afin de procéder à l'apprentissage, une étape au préalable est nécessaire est c'est le prétraitement de données ou Data preprocessing**.**

Durant cette étape on va traiter les données en effectuant des transformations, le but est de transformer des données d'un état brut à un ensemble de données vierges exploitables par un modèle. Il y a certains aspects à prendre en compte comme le fait que les données prisent en entrées de chaque modèle sont différentes, mais aussi la nécessité de généraliser afin qu'on puisse tester plusieurs algorithmes pour la comparaison.[27]

Nous allons voir 3 méthodes de prétraitement des données pour le machine learning:

- **Rescale Data :** remettre à l'échelle la donnée, car il arrive que certaines données au seins même du dataset aient différentes échelles. Le redimensionnement des données en entrées est chose utile pour l'utilisation d'**algorithme** d'**optimisation** (ex: gradient descent).
- **Binarize data :** ou **Binarisation des données** c'est une méthode qui permet de transformer des entrées quelconque en valeur binaire, et cela est chose possible grâce au **seuil binaire** (**binary threshold**) à partir duquel on va comparer si la valeur de nos données sont supérieurs au seuil alors on marque d'un 1, et dans le cas contraire on marque d'un 0.
- **Standardize data** : Standardisation ou normalisation des données, c'est une méthode qui s'avère utiles pour transformer des attributs en distribution gaussienne et des moyennes et écart-types différents en une distribution gaussienne standard avec une moyenne de 0 et un écart-type de 1.[27]

II.21. Découpage des données :

L'ensemble des données ou Datase t à disposition est découpé selon 2 voire 3 parties :

- **Données d'apprentissage :** ou training set qui représente environ **80%** des données mise à disposition, elles serviront à entraîner notre modèle.

- **Données de test :** ou Test set représente **20%** des données globales elles serviront durant la phase de test du modèle.
- **Données de Validation :** représente une portion (15 à 20%) des données qui vont servir à tester le modèle, pour valider les résultats de l'entraînement. Ses données ne servent pas évaluer l'efficacité d'un modèle, mais plutôt des algorithmes entre eux.[27]

Remarque : La validation de l'entraînement n'est pas obligatoire.

II.22. Fonction perte :

Également appelé fonction d'erreur (Loss Function) dont le rôle de base est simple, c'est l'évaluation de nos prédictions effectuées sur le dataset, si elles sont bonnes ou mauvaises, dans le cas où les prédictions sont mauvaises voir même erronées alors elle générera une valeur supérieure, et dans le cas contraire elle produira une valeur inférieure.

Il y a de nombreuses fonctions permettant de calculer la perte (erreur), voici quelques-unes pour la classification. **[27]**

- **Mean Square Error : MSE :** c'est l'une des fonctions les plus utilisées, et plus simple à mettre en oeuvre. On prend la différence entre vos prédictions, et la vérité sur le terrain, vous la corriger et vous la calculer en moyenne sur l'ensemble du jeu de données.
- **Log Loss (Cross Entropy Loss) :** C'est aussi une fonction de perte populaire, et son utilisation principale réside dans les problèmes de **classifications** employant les **logarithmes.**

II.23. Optimiseurs

l'optimiseur implémente ce qu'on appelle l' algorithme de Rétropropagation de l'erreur, qui est l'algorithme central de l'apprentissage pro-fond pour ajuster les poids.

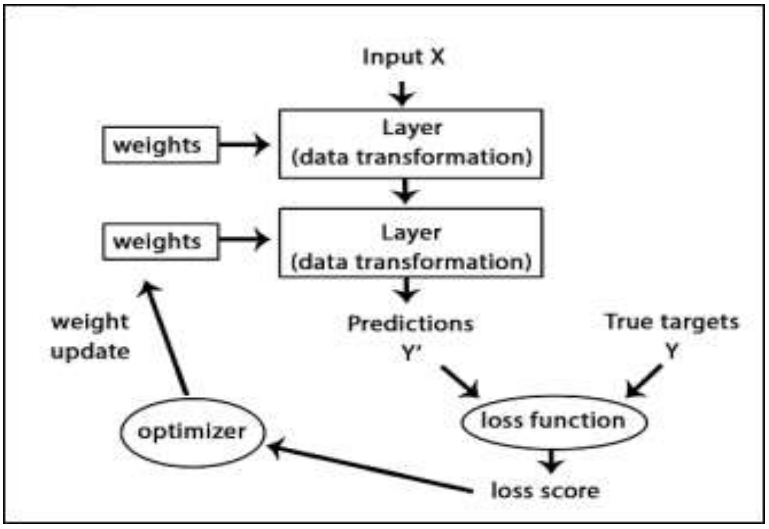


Figure II.20 : Utilisez le score de loss comme signal de retour pour ajuster les poids par l'optimiseur

II.23.1. Descente de gradient

La descente de gradient est l'algorithme de rétropropagation le plus basique, c'est un algorithme d'optimisation du premier ordre qui dépend de la dérivée du premier ordre d'une fonction de loss. Il calcule de quelle manière les poids doivent être modifiés pour que la fonction puisse atteindre un minimum, sa formule est :

$$\theta = \theta - \alpha \nabla J(\theta) \quad \textbf{(II.4)}$$

II.23.2. Momentum

Développé par Rumelhart et al. en 1986 .momentum est comme une balle qui descend une pente. La balle prend de l'élan en descendant la colline, ce qui contribue à accélérer la descente en gradient (GD) lorsque les surfaces présentent une courbe plus raide dans une direction que dans une autre

$$V(t) = \gamma V(t-1) + \alpha \nabla J(\theta) \quad \textbf{(II.5)}$$

$$\theta = \theta - V(t) \quad \textbf{(II.6)}$$

Où γ est un terme de quantité de mouvement (momentum term) généralement égal à 0.9

II.23.3. AdaGrad

Adagrad ou Algorithme de gradient adaptatif est une méthode de taux d'apprentissage adaptatif publiée en 2011 par Duchi et al. Adagrad adopte le taux d'apprentissage aux paramètres et effectue des mises à jour plus importantes pour les paramètres peu fréquents et des mises à jour plus petites pour les paramètres fréquents .

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\alpha}{\sqrt{gt.i+\varepsilon}} \cdot g_{t,i} \quad \textbf{(II.7)}$$

Où gt,i dérivée de la fonction de loss à un instant t donné.

II.23.4. RMSprop

est une méthode de taux d'apprentissage adaptatif proposée par GeoffHinton , elle utilise la magnitude des descentes de gradient récentes pour norma-liser le gradient, dans RMSProp le taux d'apprentissage est ajusté automatiquement et il choisit un taux d'apprentissage différent pour chaque paramètre.

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{(1-\gamma)g_t^2+\gamma g+ \varepsilon}} \cdot g_t \quad \textbf{(II.8)}$$

II.23.5. Adam

Développé par Kingma et al. en 2014 , Adam (Adaptive Moment Estimation) peut être considéré comme une combinaison d'AdaGrad, qui fonctionne bien sur les gradients sparse et de RMSprop qui fonctionne bien dans des paramètres en ligne et non stationnaires L'algorithme Adam met d'abord à jour les moyennes mobiles exponentielles du gradient mt et du gradient au carré vt qui sont les estimations du premier et du deuxième moment.

Les hyper-paramètres β1, β2 ∈ [0, 1) contrôlent les taux de décroissance exponentielle de ces moyennes mobiles comme indiqué ci-dessous :

$$m_{t=\frac{mt}{1-\beta^t}} \quad \textbf{(II.9)}$$

$$v_{t=\frac{v_t}{1-\beta_2^t}} \quad \textbf{(II.10)}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v_t+\varepsilon}}\, m_t \quad \textbf{(II.11)}$$

II.24. Conclusion

Dans ce chapitre, nous avons décrit c’est quoi un neurone, et comment ça fonctionne, ça nous a permis d’appuyer sur des fondements mathématiques qui nous permettent d'envisager un large éventail d'applications, et surtout dans le domaine de la fabrication, qui lui-même prend beaucoup de temps à travailler, sans oublier les domaines de la classification qui permettent une approche efficace et générale de résolution des problèmes.

Chapitre III : simulation de la stratégie Deep ACE

Chapitre III: simulation de la stratégie Deep ACE

III.1. Introduction

Dans ce chapitre, nous allons présenter la démarche de notre simulation ainsi que les résultats obtenus. Nous allons simuler notre modèle par le langage de programmation « Python » à l'aide d'un éditeur de code « Anaconda Spyder » en utilisant l'apprentissage profond, et à la fin nous allons travailler avec le logiciel « MATLAB » pour compléter les blocs de la stratégie de codage réalisée pour l'implant cochléaire.

III.2. Environnements et Outils de développement

Pour développer notre système, nous allons utiliser différents environnements et outils pour le langage de programmation y compris des API, bibliothèques, langage de balisage aussi la feuille de style avec différentes bibliothèques et cadres, travaillé sur de nombreux éditeurs, IDE etc ....

III.2.1. Python



Figure III. 1 : Log du python

Python est un langage de programmation interprété, de haut niveau et généralisé. Créé par Guido van Rossum et publié pour la première fois en 1991, la philosophie de conception de Python met l'accent sur la lisibilité du code avec son utilisation notable d'espaces blancs significatifs. Ses constructions de langage et son approche orientée objet visent à aider les programmeurs à écrire du code clair et logique pour des projets de petite et de grande envergure.[31]

III.2.2. Tensorflow

TensorFlow est une bibliothèque de logiciels libre et open-source pour le flux de données et la programmation différenciable sur une gamme de tâches. Il s'agit d'une bibliothèque mathématique symbolique, qui est également utilisée pour des applications

d'apprentissage automatique comme les réseaux neuronaux. Il est utilisé à la fois pour la recherche et la production chez Google.

TensorFlow a été développé par l'équipe de Google Brain pour une utilisation interne de Google. Il a été publié sous la licence Apache 2.0 le 9 novembre 2015.



Figure III. 2 :Logo du TensorFlow

III.2.3. Keras

Keras est une bibliothèque de réseau neuronal open-source écrite en Python. Il est capable de fonctionner sur TensorFlow, Microsoft Cognitive Toolkit, Theano, ou PlaidML. Conçu pour permettre une expérimentation rapide des réseaux neuronaux profonds, il se concentre sur une utilisation conviviale, modulaire et extensible. Il a été développé dans le cadre de l'effort de recherche du projet ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), et son auteur principal et responsable est François Chollet, un ingénieur de Google. Chollet est également l'auteur du modèle de réseau neural profond XCeption.

En 2017, l'équipe TensorFlow de Google a décidé de soutenir Keras dans la bibliothèque principale de TensorFlow.

Chollet a expliqué que Keras a été conçu pour être une interface plutôt qu'un cadre d'apprentissage automatique autonome. Il offre un ensemble d'abstractions plus intuitives et de plus haut niveau qui facilitent le développement de modèles d'apprentissage profond, quel que soit le moteur de calcul utilisé. Microsoft a ajouté un backend CNTK à Keras ainsi, disponible à partir de CNTK v2.0.



Figure III. 3 :Logo du Keras

III.2.4. Numpy

NumPy est une bibliothèque pour le langage de programmation Python, ajoutant le support pour les grands tableaux et matrices multidimensionnels, avec une grande collection de fonctions mathématiques de haut niveau pour fonctionner sur ces tableaux.



Figure III. 4 :Logo du Numpy

III.2.5. Matplotlib

Matplotlib est une bibliothèque de t:racés pour le langage de programmation Python et son numérique mathématique extension NumPy. Il fournit une API orientée objet pour intégrer des tracés dans des applications utilisant des outils GUI généraux comme Tkinter, wxPython, Qt ou GTK+.



Figure III. 5 :Logo du matplotlib

III.2.6. Anaconda

Anaconda est une distribution libre et open-source des langages de programmation Python et R pour l'informatique scientifique, qui vise à simplifier la gestion et le déploiement des paquets. Les versions des paquets sont gérées par le système de gestion des paquets conda.



Figure III. 6 :Logo du Anaconda

III.2.7. Spyder

Spyder est un environnement de développement pour Python. Libre et multiplateforme, il intègre de nombreuses bibliothèques d'usage scientifique : Matplotlib, NumPy, SciPy et IPython.



Figure III. 7:Logo du Spyder

Spyder est un environnement de développement intégré comme IDLE créé pour la programmation scientifique. Spyder est livré avec plusieurs fonctionnalités telles qu'un éditeur de code, une console interactive et finalement un explorateur de variables.

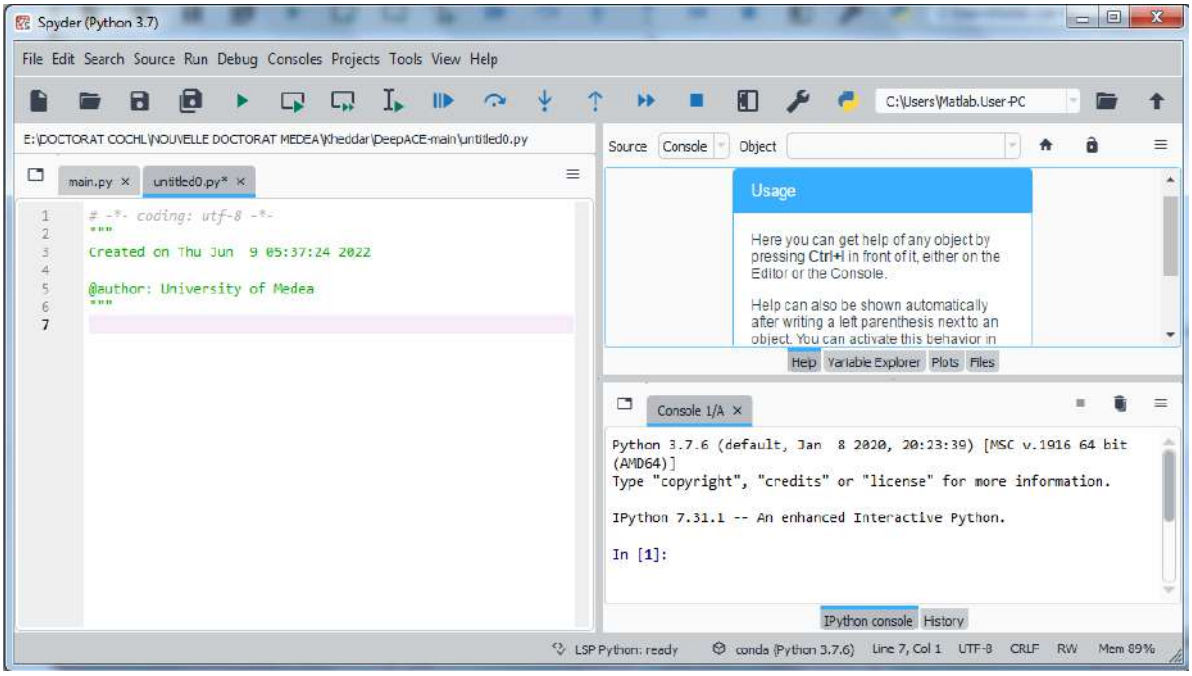


Figure III. 8 : L'environnement de travail du Spyder

III.3. Conception de la stratégie de stimulation

Dans cette section, nous allons décrire les étapes de conception et de la stratégie de stimulation cochléaire en utilisant le Deep Learning.

Le schéma bloc de la stratégie réalisée est donné par la figure suivante :

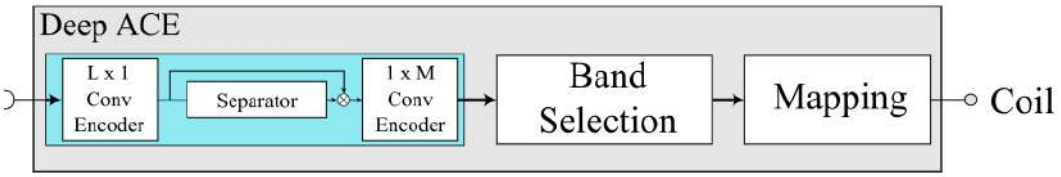


Figure III. 9:Schéma bloc de la stratégie de stimulation

La partie utilisant les réseaux de neurone est nommée « Deep ACE », elle est composée des blocs suivants :

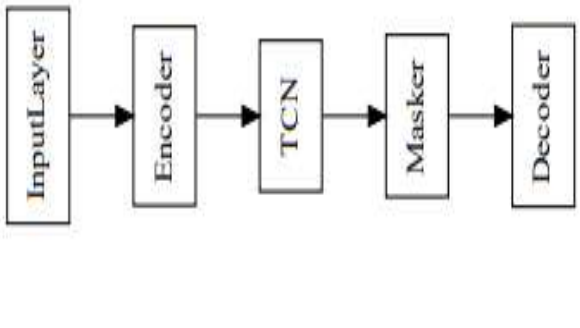


Figure III. 10 : Deep ACE

Dans cette partie, le réseau de neurone prend en entrée le vecteur du fichier audio, et estime en sortie la fonction LGF en prédiquant $p_k \in [0,1]$, avec :

$$p_k = \begin{cases} log(1+\rho((E_k - s)/(m-s))/log(1+\rho)) & si \quad s \leq E_k \leq m \\ 0 & ailleurs \end{cases} \quad \textbf{(III.1)}$$

Sachant que ***s*** est l'entrée d'un niveau de signal bas augmentée à la valeur de seuil, et ***m*** est la valeur d'entrée ou la sortie sera saturée. Le paramètre ρ contrôle la pente du ***LGF***.

La fonction d'activation utilisée dans « Deep ACE », est donnée par :

$$\emptyset(x) = \begin{cases} \alpha x & si\ x \geq 0 \\ -\beta x & ailleurs \end{cases} \quad \textbf{(III.2)}$$

Avec $\{(\alpha, \beta) \in R^{0+} \times R^{0+}\}$ sont des scalaires calculées dans la phase d'apprentissage, qui contrôlent la pente positive et négative du redresseur.

Cette fonction d'activation garantit que le signal codé est représenté par des nombres supérieurs à zéro.

Le bloc Deep ACE permet d'estimer la sortie LGF dans le domaine fréquentiel, prêt à l'utiliser pour la selection des bandes dans la strategie ACE standard.

III.4. La base de donnée audio (Dataset) :

L'utilisation des réseaux de neurones artificielles nécessite l'utilisation d'une base de donnée (dataset) pour l'apprentissage du réseau de neurone.

La basse de données audio a été fourni par le « 2nd Clarity Enhancement Challenge ». Il se compose de 6000 scènes dont 24 différents personnes qui parlent. La basse de données de développement, utilisé pour surveiller la performance des modèles neuronaux, se compose de 2500 scènes dont 10 parleurs cibles. Chaque scène correspond à un énoncé cible unique et à un segment unique de bruit provenant d'un brouilleur, mélangé à des SNR allant de -6 à 6 dB. Les réponses impulsionnelles binaurales dans une chambre (BRIR) ont été utilisées pour modéliser un auditeur dans un environnement acoustique réaliste. Les signaux audio pour les scènes sont générés par la convolution des signaux source avec les BRIR et la sommation. Des BRIR ont été générés pour les aides auditives situés de chaque côté d'écoute, offrant 3 canaux chacun (avant, milieu, arrière), à partir duquel seul le microphone avant a été utilisé. La base de donne téléchargée à partir du lien « *https://claritychallenge.github.io/clarity_CC_doc/docs/cec2/cec2_download* » a été sauvegardée dans un répertoire nommé *« data »*.[30]

III.5. Déclaration des variables :

Dans la première étape de notre programme, nous allons utiliser une bibliothèque « ***argparse*** », et pour attribuer des paramètres par la commande : ***parser.add_argument***.

Ces paramètres sont respectivement :

- Topologie du réseau de neurone de type « string » nommée « DeepACE »
- Le taux d'apprentissage : entier : 10-3
- « ***model_dir*** » est le chemin d'accès d'un répertoire nommé « ***models*** »
- « ***data_dir*** » est le chemin d'accès du répertoire nommé « ***data*** »
- La fréquence d'échantillonnage : entier : 16×10^3
- Le nombre d'itérations : entier = 100 itérations
- Taille des lots de données : entier = 2
- La durée : entier = 4 s
- N : entier : nombre de filtres dans ***autoencoder***= 64 filtres
- B : entier : Nombre de canaux dans le goulot d'étranglement 1 × bloc 1-conv = 64
- H : entier : Nombre de canaux dans les blocs convolutionnels = 128

- P : entier : Taille du noyau dans les blocs convolutionnels = 128
- X : entier : Nombre de blocs convolutionnels dans chaque répétition = 2
- R : entier : nombre de répétition = 2
- Taux de stimulation : entier : 1000
- La longueur de bloc : entier : 128
- Nombre d'électrodes = 22 électrodes

Avant d'écrire le code générant l'architecture du réseau neuronal, nous créant un programme permettant d'accéder et de prétraiter la base de donnée utilisée.

III.6. Génération des données

La première étape qu'on a fait est de crée une classe nommé ***DataGenerator*** après l'appelle des bibliothèque ***os***, ***sys***, ***glob***, ***librosa***, ***np*** (c'est l'abréviation du nom de bibliothèque numpy), ***tqdm***, ***sio*** (qui est l'abréviation du nom de bibliothèque ***scipy.io***), et la bibliothèque ***tensorflow*** (***tf***).

```
import os
import sys
import glob
import librosa
import numpy as np
from tqdm import tqdm
import scipy.io as sio
import tensorflow as tf
```

Figure III. 11 :l'appelle des bibliothèques

Le corps du programme ***DataGenerator*** contient les fonctions : ***__init__***, ***_float_list_feature***, ***fetch, _encode*** :

```
class DataGenerator():
    def __init__(self, mode, args):
        if mode != "train" and mode != "valid":
            raise ValueError("mode: {} while mode should be "
                             "'train'".format(mode))


        if not os.path.isdir(args.data_dir):
            raise ValueError("cannot find data_dir: {}".format(args.data_dir))

        self.wav_dir = os.path.join(args.data_dir, mode)
        self.tfr = os.path.join(args.data_dir, mode + '.tfr')
        self.mode = mode
        self.batch_size = args.batch_size
        self.sample_rate = args.sample_rate
        self.duration = args.duration
        self.M = args.n_electrodes
        self.block_shift = int(np.ceil(self.sample_rate/args.channel_stim_rate))
        self.n_frames = int(np.ceil(self.duration*self.sample_rate/self.block_shift))


        if not os.path.isfile(self.tfr):
            self._encode(self.mode)

    def _float_list_feature(self, value):
        return tf.train.Feature(float_list=tf.train.FloatList(value=value))


    def fetch(self):
        dataset = tf.data.TFRecordDataset(self.tfr).map(self._decode,
                                                        num_parallel_calls=tf.data.experimental.AUTOTUNE)
```

Figure III. 12 : Aucune entrée de table d'illustration n'a été trouvée.

- La fonction ***__init__*** : pour initialisé les valeurs, dans le mode ***train*** ou bien ***valid*** : si le mode n'est ni train ni valide un message d'erreur sera affiché. Nous vérifions aussi si le répertoire « ***data_dir*** » existe ou non, s'il n'existe pas le message « ***cannot find data_dir : { }*** » est affiché. Après nous joindrons les valeurs et les fichiers (***wav*** et ***tfr***).
- La fonction ***_ float_list_feature*** : c'est pour écrire les valeurs dans une liste.
- La fonction ***fetch*** : c'est pour convertir la base de donnée (fichiers ***.wav***) en forme numérique.

```
if self.mode == "train":
    dataset = dataset.shuffle(2000, reshuffle_each_iteration=True)
   # print("dataset=", dataset)

    train_dataset = dataset.batch(self.batch_size, drop_remainder=True)
    train_dataset = train_dataset.prefetch(tf.data.experimental.AUTOTUNE)
    return train_dataset

if self.mode == "valid":
    valid_dataset = dataset.batch(1, drop_remainder=True)
    valid_dataset = valid_dataset.prefetch(tf.data.experimental.AUTOTUNE)
    return valid_dataset
```

Figure III. 13 :les deux modes train et valid .

Dans la partie du code précédent, et dans la phase d'apprentissage et de validation :

- Les échantillons dans la base de donnée seront redimensionnés avec un nombre d'échantillons 2000.
- Nous créant des objet ***train_dataset*** et ***valid_dataset*** pour regrouper les échantillons en lot avec une taille de lot spécifique.

```
def _encode(self, mode):

    if self.mode == "train":
        print("\nSerializing training data...\n")

    if self.mode == "valid":
        print("\nSerializing validation data...\n")

    writer = tf.io.TFRecordWriter(self.tfr)

    mix_filenames = glob.glob(os.path.join(self.wav_dir, "*_mixed_CH1.wav"))

    target_filenames = glob.glob(os.path.join(self.wav_dir, "*_target_CH1_LGF.mat"))

    sys.stdout.flush()
    for mix_filename, target_filename in tqdm(zip(mix_filenames,
                                              target_filenames), total = len(mix_filenames)):
        mix, _ = librosa.load(mix_filename, self.sample_rate, mono = True)
        clean = sio.loadmat(target_filename)['lgf_clean']
        clean = clean.astype(mix.dtype)
```

Figure III. 14 : l'encodage des signaux .

Le but de la fonction précédente ***_encode*** est : de mélanger les audio pur avec les audio dans le répertoire ***interferent*** dans la base de donnée afin de les séparer prochainement (dans la fonction ***_decode***), et comme ça le système fait l'apprentissage : on prend par exemple un audio_1 pur et un autre audio contient un bruit seulement, le fichier contenant audio_1 pur + le bruit sera un signal d'entrée pour la fonction ***_decode***. Le système va apprendre comment obtenir en sortie le signal audio_1 pur seulement.

Targer_filenames est un fichier contenant des audio purs, et ***mix_filename*** contenant des audio bruités.

```
def _decode(self, serialized_example):
    example = tf.io.parse_single_example(
        serialized_example,
        features={
            "noisy": tf.io.VarLenFeature(tf.float32),
            "clean": tf.io.VarLenFeature(tf.float32)})

    noisy =  tf.sparse.to_dense(example["noisy"])

    clean =  tf.sparse.to_dense(example["clean"])

    clean = tf.reshape(clean, (self.n_frames, self.M))

    return noisy, clean
```

Figure III. 15 : le décodage des signaux

La fonction précédente permet de séparer le son pur et le son bruyant par la fonction « ***tf.separe.tp_dense*** ».

III.7. Le model du réseau de neurone

Notre réseau contient sept couche : deux couches cacher et cinq non, tel que chaque classe hérite ***tf.keras.layers.Layer*** après l'appelle de bibliothèque ***tensorflow*** représente une couche , et chaque couche a un but spécifique :

```
def call(self):

    inp  = tf.keras.Input(shape = (None,), name = "Input")

    enc_inp = self.encoder(inp)

    skp = self.TCN(enc_inp)

    masked = self.masker(skp, enc_inp)

    out = self.decoder(masked)


    model = tf.keras.Model(inputs = inp, outputs = out, name = self.model_name)


    for i, w in enumerate(model.weights):
        split_name = w.name.split('/')
        if len(split_name) > 1:
            new_name = split_name[0] + '_' + str(i) + '/' + split_name[1] + '_' + str(i) + '/' + split_name[-1]
            model.weights[i]._handle_name = new_name
        else:
            new_name = "new" + '_' + str(i) + '/' + "layer_name" + '_' + str(i) + '/' + split_name[-1]

    return model
```

Figure III. 16 :création des couches

Le but de cette fonction est de construite le model neuronal, par l'identification des entrées et les sorties de ce modèle, tel que l'entrer est nommé « ***inp*** », la couche ***encoder*** est nommé « ***enc_inp*** », la couche ***Masker*** est nommé « ***masked*** », et la sortie de ce modèle qui est la sortie de la couche ***Decoder*** est nommé « ***out*** ».

III.8. Le programme principal

Dans ce programme on fait appel aux autres fonctions pour simulé notre model. D'abord on va appeler les bibliothèques qu'on a besoin :

```
import os
import sys
sys.dont_write_bytecode = True
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

import tensorflow as tf

import json
import glob
import warnings
import scipy.io as sio
from utils import setup
from model import DeepACE
from datetime import datetime
import matplotlib.pyplot as plt
from collections import namedtuple
from data_generator import DataGenerator
import numpy as np
```

Figure III. 17 : l'appelle des bibliothèques

III.8.1. Apprentissage

La fonction de cout utilisée dans l'apprentissage est l'erreur quadratique moyenne (***MSE***), ainsi que l'optimiseur de ADAM.

```
def train(args):

    time_stamp = datetime.now().strftime("%d-%m-%Y_%H-%M-%S")

    model_dir = os.path.join(args.model_dir, (args.topology + '_' + 'model[{}]').format(time_stamp))

    train_ds = DataGenerator("train", args).fetch()

    valid_ds = DataGenerator("valid", args).fetch()

    opt = tf.optimizers.Adam()

    os.makedirs(model_dir)

    args.mode = "test"

    json.dump(vars(args), open(os.path.join(model_dir,'params.json'), 'w'), indent=4)

    log_path = os.path.join(model_dir, 'train_log[{}].csv'.format(time_stamp))

    checkpoint_path = os.path.join(model_dir, 'model_best[{}].h5'.format(time_stamp))

    # This callback will stop the training when there is no improvement in
    callbacks = [
        tf.keras.callbacks.EarlyStopping(monitor='loss', patience = 5),
        tf.keras.callbacks.ReduceLROnPlateau(monitor='loss', factor = .8, patience=3, min_lr = 0.),
        tf.keras.callbacks.CSVLogger(log_path),
        tf.keras.callbacks.ModelCheckpoint(checkpoint_path,
                                           monitor='loss', verbose=0, save_best_only=True,
                                           mode='auto', save_freq='epoch')]
```

Figure III. 18 :le programme générale

.

L'apprentissage sera réalisé pour un nombre d'***epoch*** maximal égale à 100.

La courbe donnant l'évolution du MSE en fonction de nombre d'itérations est donnée par la figure suivante :

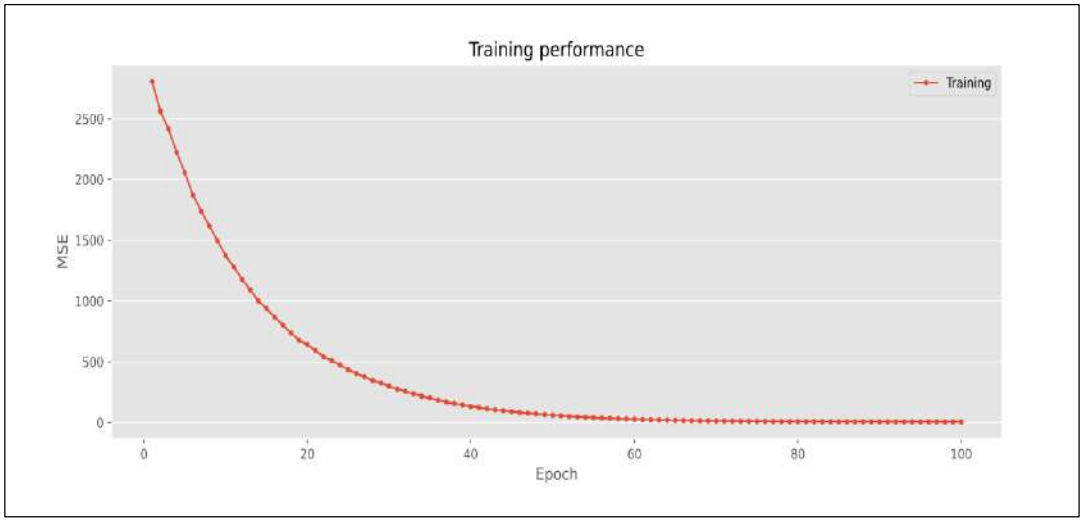


Figure III. 19 :L'évolution du MSE

D'après l'évolution du MSE, nous pouvons conclure que le modèle a bien entretenu et le MSE a été minimisé.

Les paramètres de notre model seront sauvegardés dans un fichier temporaire qu'on peut l'ouvrir et le consulter. Voici un exemple dans la figure suivante :

```
{
    "topology": "DeepACE",
    "mode": "test",
    "GPU": true,
    "learning_rate": 0.001,
    "model_dir": "./models",
    "data_dir": "./data",
    "sample_rate": 16000.0,
    "causal": true,
    "max_epoch": 100,
    "batch_size": 2,
    "skip": true,
    "duration": 4,
    "N": 64,
    "L": 32,
    "B": 64,
    "H": 128,
    "S": 32,
    "P": 128,
    "X": 3,
    "R": 2,
    "channel_stim_rate": 1000,
    "block_length": 128,
    "n_electrodes": 22
}
```


Figure III. 20 :Contenu du fichier de sauvegarde des paramètres du model neuronal

III.8.2. Evaluation

La sortie du système Deep ACE pour un fichier audio en entrée est les 22 valeurs de la fonction LGF, que l'on peut les sauvegarder dans un fichier temporaire d'extension .h5.

Le fichier précédant a été ouvert par un logiciel de lecture appelé ***HDFview***, comme le montre la figure suivante :

	0	1	2	3	4	5	6	7	8	9	10	11
0	0.07965...	0.11300...	0.10378...	-0.24387...	0.13219...	-0.04806...	-0.23740...	-0.17924...	0.18517...	0.23689...	0.12946...	-0.10181...

11	12	13	14	15	16	17	18	19	20	21
-0.10181...	0.03317...	-0.19256...	0.03698...	0.19391...	0.14104...	-0.24392...	0.07945...	0.15401...	0.24310...	-0.06340...

Figure III. 21:les valeurs de chaque éléctrode

Nous allons par la suite réaliser les deux blocs restants de notre stratégie de stimulation, à savoir le bloc de sélection des 8 canaux les plus énergétiques, et le bloc ***mapping***. Cette dernière tache a été réalisée en logiciel ***Matlab***. Voici le code écrit en ***Matlab*** en prenant les valeurs de la fonction ***LGF*** affichées dans le logiciel ***HDFview***.

```
clear
clc
ele=1:22;
lgf=[0.017116994
   -0.14328124
   -0.19362748
   -0.08883162
   -0.1537329
   0.19779453
   0.068906546
   -0.056180656
   0.11727902
   0.11365551
   -0.1948769
   0.2370002
   0.1783486
   -0.12462647
   -0.12169236
   -0.1316248
   -0.24152128
   -0.06301367
   -0.18753403
   -0.1619132
   0.23728833
   0.07763761];
lgf=lgf';
bar(ele,abs(lgf))
lgf_croiss=sort(abs(lgf))';
lgf_decroiss=lgf_croiss(22:-1:1)
electrode=[];
for i=1:22
   k=1;
   while k<=22
   if lgf_decroiss(i)==abs(lgf(k))
      electrode=[electrode k];
   end
   k=k+1;
   end
end
trie=[electrode' lgf_decroiss]
can_stim=trie(1:8,:)
figure
bar(can_stim(:,1),can_stim(:,2))
```

Après exécution du code ***Matlab***, nous obtenons les résultats suivants :

Premièrement nous trions les valeurs de la fonction ***LGF*** d'une manière décroissante comme le montre le résultat suivant :

```
lgf_decroiss =

    0.2415
    0.2373
    0.2370
    0.1978
    0.1949
    0.1936
    0.1875
    0.1783
    0.1619
    0.1537
    0.1433
    0.1316
    0.1246
    0.1217
    0.1173
    0.1137
    0.0888
    0.0776
    0.0689
    0.0630
    0.0562
    0.0171
```

La representation de ces valeurs en histogramme est donnée par la figure suivante :

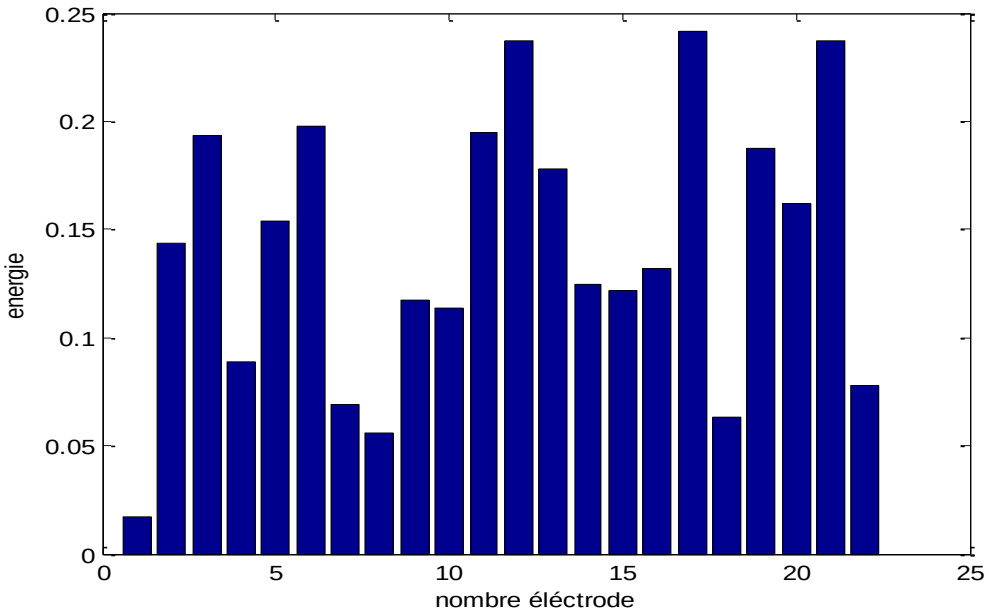


Figure III. 22 :Représentation des 22 valeurs de la fonction LGF

Dans le résultat suivant nous affectant chaque valeur de ***LGF*** a son électrode appropriée.

```
trie =

   17.0000    0.2415
   21.0000    0.2373
   12.0000    0.2370
    6.0000    0.1978
   11.0000    0.1949
    3.0000    0.1936
   19.0000    0.1875
   13.0000    0.1783
   20.0000    0.1619
    5.0000    0.1537
    2.0000    0.1433
   16.0000    0.1316
   14.0000    0.1246
   15.0000    0.1217
    9.0000    0.1173
   10.0000    0.1137
    4.0000    0.0888
   22.0000    0.0776
    7.0000    0.0689
   18.0000    0.0630
    8.0000    0.0562
    1.0000    0.0171
```

Généralement, dans la stratégie ACE ordinaire, 8 électrodes sont sélectionnées pour la stimulation parmi 22.

Le résultat suivant montre les 8 électrodes sélectionnés dans notre exemple de test.

```
can_stim =

   17.0000    0.2415
   21.0000    0.2373
   12.0000    0.2370
    6.0000    0.1978
   11.0000    0.1949
    3.0000    0.1936
   19.0000    0.1875
   13.0000    0.1783
```

Et voici la figure qui représente les 8 électrodes sélectionnés :

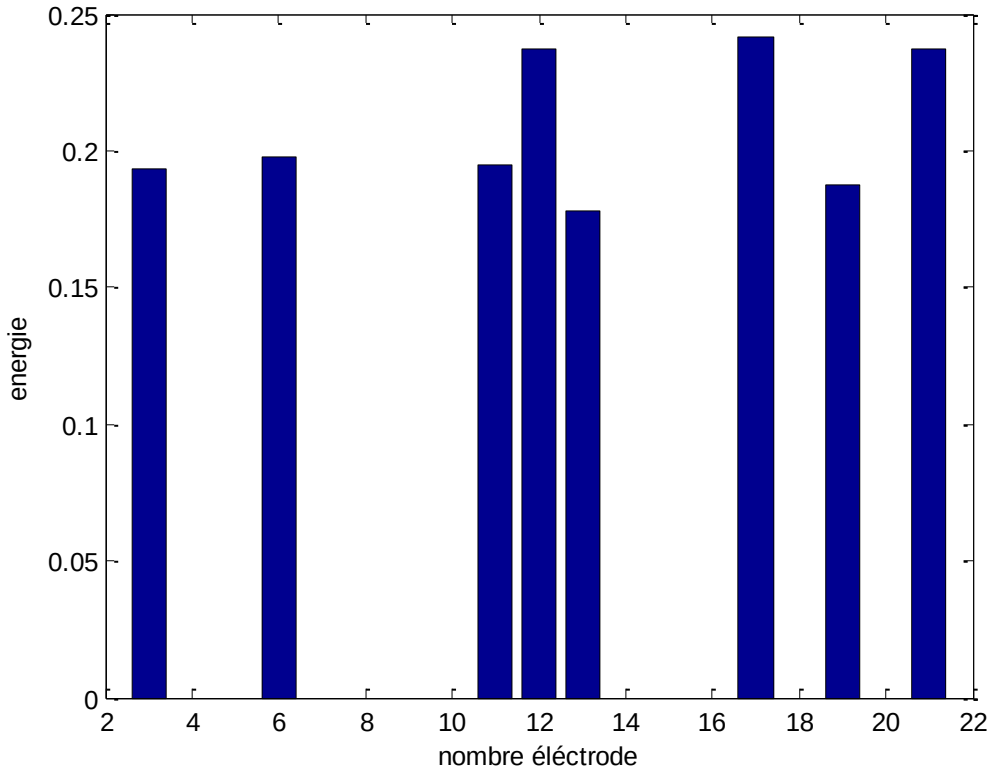


Figure III. 23 :Sélection des 8 valeurs de la fonction LGF parmi 22

.

Après ce dernier résultat, les électrodes sont près à stimuler la cochlée.

III.9. Conclusion

Ce chapitre présente la méthodologie qui a été suivie tout au long du développement de ce projet, il explique en détail comment la stratégie de stimulation cochléaire a été développée, tout en décrivant en détail la mise en œuvre de chacune des différentes étapes du système, commençons par l'entrée (audio) jusqu'à le résultat du model et finalement la vérification.

Conclusion générale

Dans ce mémoire nous avons décrit la méthode de résolution du problème de la surdité neurosensorielle. La surdité neurosensorielle est une maladie qui est due à un problème au niveau de l'oreille interne, exactement dans la cochlée à cause de l'absence ou le manque des cellules ciliés, qui présentent un moyen important de transmission de l'énergie électrique vers le nerf auditif. La solution est un appareil nommé « implant cochléaire ».

Dans le premier chapitre nous avons étudié le fonctionnement de l'oreille humaine, et ensuite nous avons parlé sur le dispositif permettant la résolution du problème des gens sourd, et les différentes stratégies de stimulation cochléaire.

Dans le chapitre 2 nous avons parlé sur l'IA, DL, ML, et sur les réseaux de neurones qui seront utilisés dans notre simulation.

Le 3eme chapitre est le chapitre de simulation. Nous avons présenté les résultats de simulation d'une méthode de stimulation cochléaire utilisant les réseaux de neurones artificielles. Ce chapitre est composé de deux partie ; une partie qui contient les outils utilisés pour réaliser le model neuronal, principalement les bibliothèques qui facilitent la construction de notre model, après la programmation en utilisant le langage Python, ensuite la présentation des résultats de simulation.

Comme perspective, nous proposons d'implémenter une tel stratégie sur des circuits numériques comme le DSP et le FPGA.

[1] Jean-Luc Puel .Qu'est ce que J'entends ? . *Dernière mise à jour : 25/05/2021 15:38* https://www.cochlea.org/ entendre.

[2] Jean Loup ,Tanguy." Le son et l'audition," [Online]. Available: http:// www.tpe- son.stlouis.over-blog.fr [Accessed 6 6 2014].

[3] Caractéristiques physique des sons .

http://acoustiquebatiment.e-monsite.com/pages/page.html

[4] Comprendre les Implants Cochléaires, MED-EL Headquarters Fürstenweg 77a 6020 Innsbruck, Austria office@medel.c

[5] Jeanne GUIRAUD .Effets de la surdité totale et de la réhabilitation auditive par l'implant cochléaire sur l'organisation fonctionnelle du système auditif. 227 p .Thèse de doctorat de Sciences cognitives : Université Lumière – LYON 2 : Présentée et soutenue publiquement le 1er Février 2007.

[6] Qu'est-ce qu'un implant cochléaire ? . 2007 Advenced Biorics SARL.rights reseved primted in France .8 p.

[7] Jean-louis BRZSSIENE. implant-brochure,systémes FM. 12 P. FFSB - Service Conseils en Aides Techniques - rue Van Eyck 11a Bte. 4 1050 Bruxelles Belgique. http://www.FFSB.BE

[8] F. Artières, G. Geffriaud, N. Gresillon, M. Mondain, J-P. Piron, S. Romdhane, M. Sicard, M. Sillon, V. Tesseydre, A. Uziel, V. Ventosa, A. Vieu : LES IMPLANTS COCHLÉAIRES CHEZ L'ENFANT ET L'ADULTE SOURD PROFOND OU SÉVÈRE : Document réalisé par les membres du GEORRIC Edité dans la monographie numéro 33 du CCA Groupe :2002 :13 p ,

[9] kais GADHOUMI ,mise en œuvre d'un algorithme de traitement de la parole basé sur la quantification vectorielle pour une prothèse cochléaire, 79 p, mémoire de maîtrisées science appliquées :canada , Université de SHERBROOKE : Avril 2021 ,

[10] Ben Huberman, Caitlin Kindig, « Towards data science – unsupervised learning with python », independent Medium publication, Oct 22, 2016

[11] https://www.futura-sciences.com/tech/dossiers/robotique-presentation-historique-reseaux-neuronaux-31/

[12] Simon HAYKIN, Neural Networks - A Comprehensive Foundation .Second Edition, Pearson Prentice Hall, India 2001: pp.823.

[13] McCulloch W.S. et Pitts W, "A Logical Calculus of the Ideas Immanent in Nervous Activity". Bulletin of Mathematical Biophysics Vol. 5, pp. 115-133

[14] BOUKHOBZA MOHAMED EL HABIB.Application des réseaux de neurones artificiels pour la classification automatique des tissus mammaires . Faculté des Sciences et de la technologie (FST), Université Abdelhamid Ibn Badis de Mostaganem l'UABM. Algérie. 2018

[15] SAVOYE, Gras. ALTERNATIVE NEURONALE EN TARIFICATION SANTE.

[16] GOURAA HANIA et DOUAOUDI SALIMA , Intitulé du sujet Modélisation d'un moteur BLDC par un réseau de neurones MLP'',85 p, MEMOIRE, MASTER EN ELECTRONIQUE, Université de Mostaganem , Année Universitaire 2019/2020

[17] DREYFUS, Gérard, et al. Les réseaux de neurones. Mécanique industrielle et matériaux, 1998, vol. 51.

[18] Wikiversite, Réseaux de Neurones: Avantages et ,18 Mai 2019.

[19] Amira, l, b. Khaoukha et l.adila, Identification et Commande des Systèmes Non linéaires. 2011.

[20] Hilali.h, Application de la Classification Textuelle pour L'extraction des Règles d'association Maximales. Thèse de Maitrise en Informatique, Université du québec à Troisrivières, Trois-rivières, 2009.

[21] Mohamed Assaad HAMIDA. Introduction aux Méthodes de Contrôle Intelligent .2014 .https://elearn.univouargla.dz/20132014/courses/INTRODUCTIONAUXMETHO/document/Cours_HAMIDA.pdf**.**

[22] B. Hadja et K .Fatiha, Modélisation d'un système solaire par Simulink et Réseaux de Neurones, 94 p, Université de Mostaganem, Année Universitaire 2020/2021.

[23] A. Géron, « Machine Learning avec Scikit-Learn », Dunod : Paris, 2017.

[24] Clayton, R. (2019). Qu'est-ce que le machine learning ? Retrieved Mars 08, 2020, from OracleAlgeria:https://www.oracle.com/dz/artificial-intelligence/what-is-machine-learning.html

[25] BENSIAH Oussama Akram, La proposition d'une nouvelle approche basée Deep Learning pour la prédiction du cancer du sain , 110p , Mémoire de fin d'étude de Master, Université L'arbi Ben M'hidi Oum El Bouagh ,2019/2020

[26] Ben Huberman, Caitlin Kindig, « Towards data science – unsupervised learning with python », independent Medium publication, Oct 22, 2016

[27] Patterson, J., & Gibson, A. (2017). Deep Learning : A Practitioner's Approach. Beijing: OReilly Media

[28] Goodfellow, I., Bengio, Y., & Courville, A. (2019). Deep Learning. MIT Press .

[29] MAOUDJ Salah Eddine, REMMOUCHE Brahim. Un Modèle CNN guidé SIFT pour la Reconnaissance des Classes d'Objet. 92 p . Faculté des Sciences Exactes et Informatique. Université Mohamed Seddik Benyahia Jijel. Promotion 2021.

[30] Tom Gajecki , Waldo Nogueira. AN END-TO-END DEEP LEARNING SOUND CODING STRATEGY FOR COCHLEAR IMPLANTS. 6 p .Department of Otolaryngology,Medical University Hannover and Cluster of Excellence Hearing4all,Hannover, 30625, Germany. · November 2021

[31] MERDOUD Kenza ,BOUSBAINE Karim. Détection de maladies par traitement d'image. 62 p. Master 02en : Génie Système Informatique. UNIVERSITE AKLI MOAND OULHADJE-BOUIRA. Année Universitaire 2017/2018.

[32] Auto envoder :

https://dataanalyticspost.com/Lexique/auto-encodeur/