

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POUPLAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MENTOURI – CONSTANTINE-
FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT D'ELECTRONIQUE

MEMOIRE DE MAGISTER

Présenté par :

Mr. Belghobsi Abdelaziz

Option :

Traitement du Signal

Thème

**Réseaux de neurones appliqués à la
modélisation et à la compression du signal
ECG**

Soutenu le / /

Examiné par le jury :

Président	A. Charef	Professeur	Université de Constantine
Rapporteur	A. Bennia	Professeur	Université de Constantine
Examineur	M. Khamadja	Professeur	Université de Constantine
Examineur	A. Chaabi	Maître de conférence	Université de Constantine

Remerciements

Avant tout je voudrais remercier Allah qui m'a permis d'arriver à conclure avec succès ce modeste travail, puis je remercie mes parents pour toute l'éducation qu'ils m'ont inculquée et mes frères et sœurs de m'avoir encouragé et permis de réaliser ce mémoire.

Je tiens à remercier vivement la personne sans laquelle tout ceci n'existerait pas, j'exprime en effet toute ma profonde gratitude au professeur Abdelhak Bennia de m'avoir accueilli au sein de son équipe du laboratoire de traitement du signal. Je lui suis sincèrement reconnaissant pour son encadrement, ses conseils et ses orientations. Mais également pour toute la confiance qu'il a porté en moi et qui m'a permis d'acquérir une précieuse expérience en tant que jeune chercheur.

Je remercie Messieurs les membres du jury pour la caution qu'ils ont bien voulu apporter à ce travail. J'adresse mes remerciements à Abdelfatah Charef professeur à l'université de Constantine, je le remercie d'avoir accepté de présider le jury de soutenance, au professeur Mohamed Khamadja et au docteur Abelhafidh Chaabi de m'avoir honoré en acceptant d'être examinateurs.

Ces années de mémoire ont été pour moi l'occasion de connaître des personnes exceptionnelles qui m'ont tout simplement offert leur sincère amitié, et avec qui j'ai partagé d'agréables moments. Je remercie très chaleureusement mes collègues de post-graduation de l'option traitement du signal, ainsi que les membres du laboratoire de traitement du signal.

Sommaire

Introduction générale	1
Chapitre 1 : Morphologie de l'ECG	
1.1 Introduction.....	3
1.2. La structure interne du cœur.....	3
1.3. Les tuniques du cœur.....	4
1.4. La conduction électrique du cœur.....	5
1.5. Présentation de l'électrocardiogramme.....	7
1.6. Les dérivations d'un électrocardiogramme.....	7
1.6.1 Les dérivations frontales.....	7
1.6.2. Les dérivations précordiales.....	9
1.7. Techniques d'enregistrement de l'ECG.....	11
1.8. Terminologies et normes.....	12
1.8.1. Ondes enregistrées.....	12
1.8.2. Intervalles temporels.....	14
1.9. Artefacts dans l'électrocardiogramme.....	16
1.10. Le Holter cardiaque.....	17
1.10.1. Limitations.....	17
1.10.2. Différents types de Holters.....	18
Chapitre 2 : Les réseaux de neurones	
2.1. Introduction.....	19
2.2. Historique.....	20
2.3. Définition des réseaux de neurones.....	22
2.3.1. Le neurone artificiel.....	22
2.3.2. Les réseaux de neurones.....	23
2.4. Structure d'interconnexion.....	24
2.4.1. Les réseaux non bouclés.....	24
2.4.2. Les réseaux bouclés.....	25
2.4.2.1. Principales architectures de réseaux récurrents.....	26
a) TDNN.....	27
b) SRN.....	27
c) NARMA.....	27
d) Le réseau de Hopfield.....	27
e) Architecture de Jordan.....	27

f) Architecture d'Elman	28
2.5. Propriétés des réseaux de neurones.....	29
2.5.1. L'apprentissage.....	29
2.5.2. Généralisation.....	29
2.5.3.L'approximation parcimonieuse.....	30
2.6. Les différents algorithmes d'apprentissage.....	31
2.6.1. Règle de Delta.....	31
2.6.2. La rétropropagation.....	32
2.6.3. Algorithmes d'apprentissage pour réseaux récurrents.....	35
2.6.3.1. Rétropropagation dans le temps.....	35
2.6.3.2. Le RTRL (Real Time Recurrent Learning).....	37
2.7. Mise en œuvre des réseaux de neurones.....	38
2.8. Domaine d'application.....	38

Chapitre 3: Compression de l'ECG

3.1. Introduction.....	39
3.2. Algorithmes de compression existants.....	39
3.2.1. Méthodes de compression directes.....	40
3.2.2. Méthodes de compression par transformation.....	40
3.2.3. Méthodes de compression par extraction des paramètres.....	40
3.3. Evaluation des performances.....	41
3.4. Algorithme proposé.....	43
3.5. Principe de travail.....	44
3.5.1. Sous section non linéaire (PRNN).....	44
3.5.1.1. Linéarisation du signal d'entrée.....	46
3.5.1.2. L'algorithme d'apprentissage du PRNN	47
3.5.2. Sous section linéaire.....	49
3.5.3 Initialisation de l'algorithme d'apprentissage.....	50
3.6. Conception du réseau	52
3.7. Résultats.....	52
3.8. Conclusion	57
Conclusion générale.....	58
Références	59

Introduction générale

Durant des siècles l'homme n'a pas cessé de lutter pour mener une vie meilleure, ça était toujours la nécessité et le besoin qui l'ont motivé à chercher et à découvrir autour de lui tout ce que lui sert à surmonter les difficultés qui peuvent l'empêcher à aboutir à ses ambitions, la chaleur et le froid par exemple ont poussé l'homme à s'enquérir des vêtements qui l'abritent de froid ou de chaleur, la nécessité de sécurité l'ont incité à développer des armes qui lui servent à se défendre, mais dans tout ça la santé a occupé une place importante dans sa lutte insistante, la sensation de la douleur a poussé l'homme à donner l'intérêt à sa santé, alors durant les dernières décades la médecine n'a pas cessé de se développer.

Aujourd'hui, en s'alliant avec la technologie, la probabilité de guérison du malade s'est amplifiée, avec l'utilisation d'un matériel très sophistiqué on arrive à avoir des diagnostics plus fiables et une bonne compréhension des maladies. Citant l'exemple d'un organe noble qui est le cœur, ce dernier qui est en perpétuel mouvement jouant un rôle important dans la vie de l'être humain, ceci lui cause de temps en temps des faiblesses qui peuvent se convertir en maladies. Ce qui a motivé des scientifiques à élaborer des techniques pour diagnostiquer ces anomalies du cœur, cela a abouti à la naissance du tracé électrocardiographique (ECG) qui constitue un outil efficace pour l'analyse de l'activité électrique du cœur. Suite à cette importance qu'il occupe, il attire une grande attention. Plusieurs efforts ont été fournis dans le traitement du signal ECG ce qui a donné lieu à des algorithmes d'analyse automatique en améliorant la qualité des enregistrements avec l'emploi des techniques de traitement du signal tel que la suppression du bruit et la détection des caractéristiques de l'ECG. Néanmoins avec le développement actuel des nouvelles techniques d'information et de communication, on ne se contente pas d'enregistrer l'ECG et faire du diagnostic automatiquement par le biais de la détection de ces caractéristiques, mais on pense à le transmettre via les réseaux de communications. Ceci pose des problèmes de transmission à cause de la taille de l'enregistrement et le taux de bits à envoyer ce qui a motivé la recherche sur la compression de l'ECG, plusieurs méthodes de compression ont été proposées.

Dans ce mémoire qui comprend trois chapitres on va proposer une technique de compression du signal ECG, pour cette raison le premier chapitre est consacré à donner

l'essentiel sur la morphologie du cœur qui est la source du signal ECG ainsi que sur la morphologie du signal lui-même.

Dans le deuxième chapitre, on donnera une illustration générale de l'outil de travail qui sont les réseaux de neurones et enfin on terminera avec le troisième chapitre où on explique les différentes méthodes existantes et les différents critères d'évaluation de performance tout en insistant sur la technique proposée et les résultats obtenus et on terminera avec une conclusion générale.

1.1 Introduction

Il fut un temps où, pour le cardiologue, le seul outil complémentaire du stéthoscope était l'électrocardiogramme, ce dernier constituant un outil essentiel pour le diagnostic.

Ce chapitre met l'accent sur la morphologie de l'ECG, il décrit bien la source de ce signal qu'est l'activité électrique du cœur en donnant une explication de la structure du cœur ainsi que son processus de conduction.

1.2. La structure interne du cœur

Le cœur est un organe musclé constitué de quatre cavités. Les deux cavités supérieures sont appelées les oreillettes, tandis que les deux cavités inférieures sont appelées les ventricules. Le cœur est séparé en deux parties: la droite (cœur droit) et la gauche (cœur gauche).

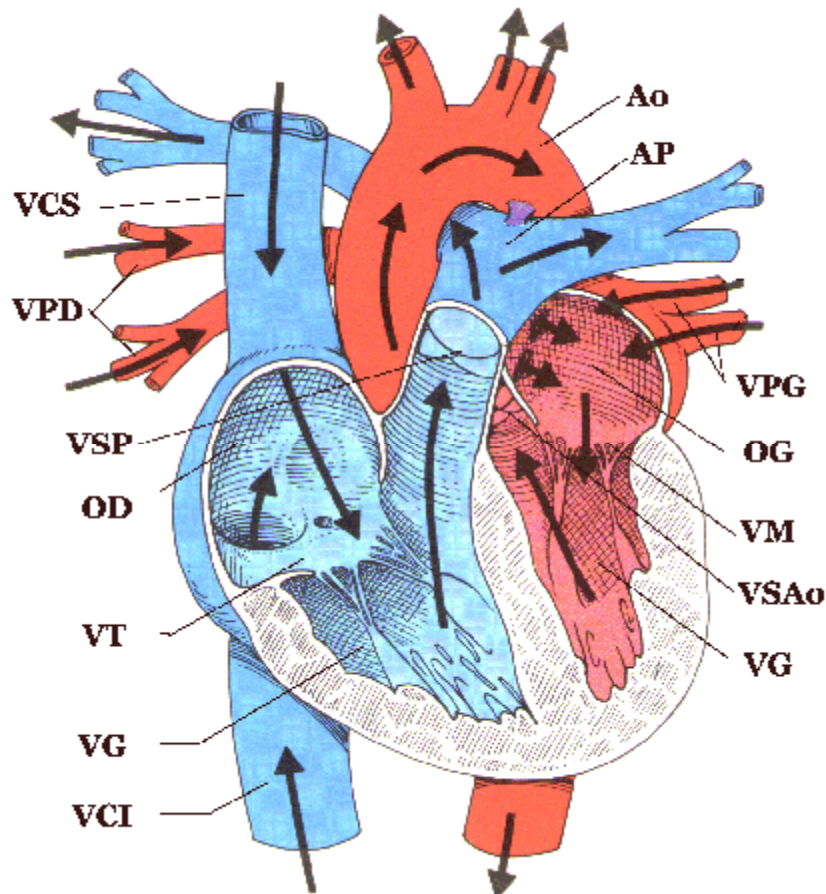


Figure (1.1) : La structure interne du cœur.

- **L'oreillette droite (OD)**

Reçoit de la veine cave supérieure (VCS) et de la veine cave inférieure (VCI) le sang veineux périphérique qui se jette ensuite dans le ventricule droit (VD).

- **Le ventricule droit (VD)**

Qui, en se contractant chasse le sang dans l'artère pulmonaire (AP). Le vestibule de l'artère pulmonaire est pourvu d'un "système anti-reflux", les trois valvules sigmoïdes pulmonaires (VSP). Entre le ventricule droit et l'oreillette droite se trouve la valvule tricuspide (VT), constituée de trois valves. Lors de la contraction ventriculaire, l'augmentation de la pression intraventriculaire ferme ces trois valves et empêche le reflux du sang dans l'oreillette droite (OD).

- **L'oreillette gauche (OG)**

Reçoit des veines pulmonaires droites (VPD) et gauches (VPG) le sang veineux pulmonaire qui se jette ensuite dans le ventricule gauche (VG).

- **Le ventricule gauche (VG)**

Qui en se contractant chasse le sang dans l'aorte (Ao), elle aussi pourvue d'un système valvulaire sigmoïde : les valvules sigmoïdes aortiques (VSAo).

Entre l'oreillette gauche et le ventricule gauche se greffe des feuillets qui forment la valvule mitrale (VM), empêchant le reflux du sang dans l'oreillette gauche lors de la contraction ventriculaire gauche.

1.3. Les tuniques du cœur

Il existe trois tuniques :

- **L'endocarde**

C'est une mince membrane qui tapisse la face interne des quatre cavités cardiaques et qui se prolonge par l'intima des gros vaisseaux.

- **Le myocarde**

C'est le tissu musculaire du cœur dont l'épaisseur dépend de la fonction des cavités, mince au niveau des oreillettes, il est particulièrement épais au niveau ventriculaire.

- **Le péricarde**

C'est une enveloppe séreuse externe du cœur constituée de deux feuillets :

L'un viscéral, adhérent au myocarde, l'autre pariétal, l'épicarde. L'espace péricardique entre les deux feuillets, contient une faible quantité de liquide (50 à 75 ml) pour faciliter les mouvements du cœur.

1.4. La conduction électrique du cœur

Il s'agit d'un réseau "électrique " constitué de cellules cardiaques, qui ont des propriétés différenciées pour la conductibilité et l'excitabilité.

1.4.1. Nœud de Keith et Flack, ou nœud sinusal

Cette structure épocardique de 15mm sur 5mm se situe à la jonction de la partie inférieure de la veine cave supérieure et de la face antérieure de l'oreillette droite. Il génère des décharges spontanées à la fréquence de 60 à 100 par minute, ce qui en fait le centre d'automatismes primaires.

1. Nœud d'Achoff Tawara

Il s'agit d'une structure de 6 mm sur 5 mm proche de la valve tricuspide, et de la cloison inter auriculaire à la base de l'oreillette droite. Il est constitué de deux voies, l'une à conduction lente (alfa), l'autre à conduction rapide (bêta). Il ralentit l'influx d'un dixième de seconde, protégeant ainsi les ventricules d'un rythme primaire trop rapide.

2. Les voies internodales

Elles font jonction entre le nœud sinusal, les oreillettes et le nœud auriculo-ventriculaire.

- faisceau internodal antérieur.
- faisceau de Bachman.
- faisceau internodal moyen de Wenchebach.
- faisceau internodal post de Thorel.
- les voies accessoires de Jame, Makaim et Kent.

3. Le faisceau de His

Long de un à deux centimètres, il est situé sous l'angle d'insertion des valves tricuspides et fait la jonction entre le nœud auriculo-ventriculaire et le ventricule par ses deux branches. Le faisceau de His est par ailleurs un centre d'automatismes

secondaire, car s'il propage, certes, l'influx de l'étage auriculaire à l'étage ventriculaire, il est aussi capable de décharger spontanément des impulsions à fréquence de 40 à 60 par minute. Il se sépare en deux branches :

a. La branche droite

Prolongement direct du faisceau de His, elle chemine le long du bord droit du septum inter ventriculaire se dispersant dans le ventricule droit.

b. La branche gauche

Elle chemine en avant et à gauche de la valve mitrale, se subdivise en faisceaux antérieur et postérieur, avec même quelquefois une branche septale.

4. Le réseau de Purkinje

Ce sont les ramifications terminales des branches droite et gauche du faisceau de His qui s'étendent sur toute la musculature ventriculaire pour propager l'influx. Mais c'est aussi un centre d'automatisme tertiaire, capable de décharger spontanément des impulsions à la fréquence de 20 à 40 par minute. Ce réseau électrique se superpose au réseau vasculaire dont il est dépendant.

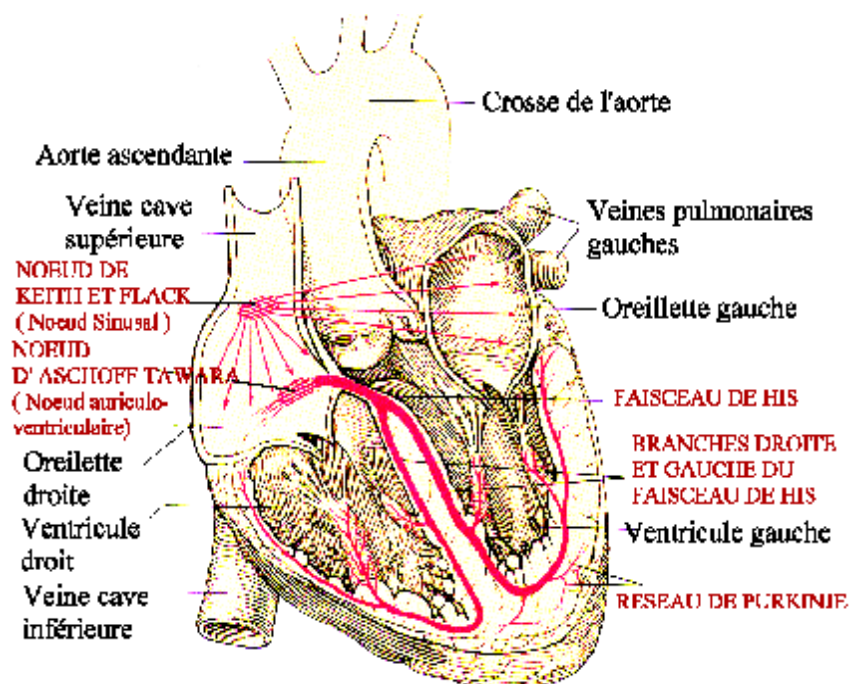


Figure (1.2) : Le système de conduction électrique du cœur.

1.5. Présentation de l'électrocardiogramme

L'électrocardiogramme est une projection graphique de l'activité électrique du cœur : c'est une image électrique de l'activité cardiaque. L'activité électrique est captée par des électrodes placées à la surface des téguments. Chaque électrode capte les ondes d'activation selon le plan du cœur qu'elle explore. Le tracé est effectué sur un papier millimétré et quadrillé.

L'électrocardiogramme est ainsi l'enregistrement sur papier des courants d'action cardiaque transmis à la surface du corps, on utilise deux électrodes de contact reliées par un fil à un galvanomètre mesurant l'intensité des courants électriques pour constituer une dérivation.

Par convention, le tracé utilise en abscisse l'échelle de temps qui correspond à la vitesse de déroulement du papier et en ordonnée le voltage, grâce au quadrillage, on appréciera l'amplitude des ondes enregistrées d'une part en durée, d'autre part en intensité. Ce quadrillage est par convention d'un millimètre sur un millimètre avec un trait renforcé tous les 5 mm :

1mm (1 petit carreau) horizontal correspond à 1 mvolt.

1mm (1 petit carreau) vertical correspond pour un déroulement de 25 mm/sec du papier à 0,04 seconde (soit 0,2 seconde par trait renforcé).

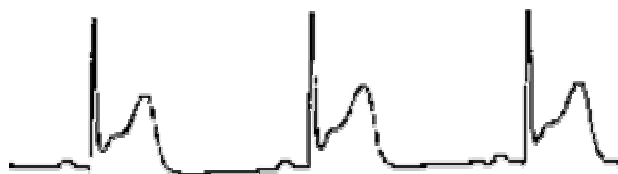


Figure (1.3) : exemple de tracé de l'ECG.

1.6. Les dérivations d'un électrocardiogramme

Il existe deux types de dérivations:

1.6.1 Les dérivations frontales

a. Loi d' Einthoven

Einthoven a imaginé les dérivations périphériques en disposant les électrodes sur le bras droit R, le bras gauche L et la jambe gauche F formant ainsi un triangle équilatéral dont le cœur occupe le centre géométrique : c'est le triangle d' Einthoven , figure (1.4).

D1 : entre bras droit (pôle -) et bras gauche (pôle +).

D2 : entre bras droit (pôle -) et jambe gauche (pôle +).

D3 : entre bras gauche (pôle -) et jambe gauche (pôle +).

D1, D2 et D3 décrivent le triangle d'Einthoven, Figure (1.5).

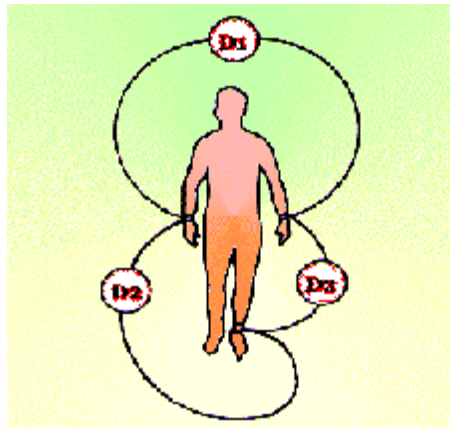


Figure (1.4) : D1, D2 et D3.



Figure (1.5) : triangle d'Einthoven.

b. Les dérivations unipolaires

aVR, aVL et aVF sont des dérivations unipolaires et correspondent au membre avec lequel elles sont connectées soit respectivement le bras droit, le bras gauche, et la jambe gauche. C'est la théorie de Wilson et Golberger, où l'électrode exploratrice positive correspond au membre appliqué. Le voltage est alors amplifié (d'où le préfixe a) pour obtenir un tracé de même amplitude que D1, D2 et D3.

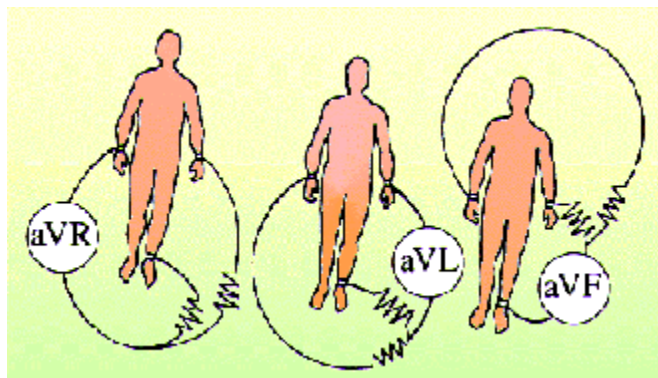


Figure (1.6) : aVR, aVL, aVF.

L'ensemble des dériviations uni et bipolaires projetées géométriquement représentent un double triaxe avec un centre schématique : le cœur.

On peut déjà apercevoir que les régions explorées par ces dériviations périphériques seront :

D1, aVL : paroi latérale du ventricule gauche.

D2, D3, aVF : paroi inférieure

aVR : intérieur des cavités du cœur.

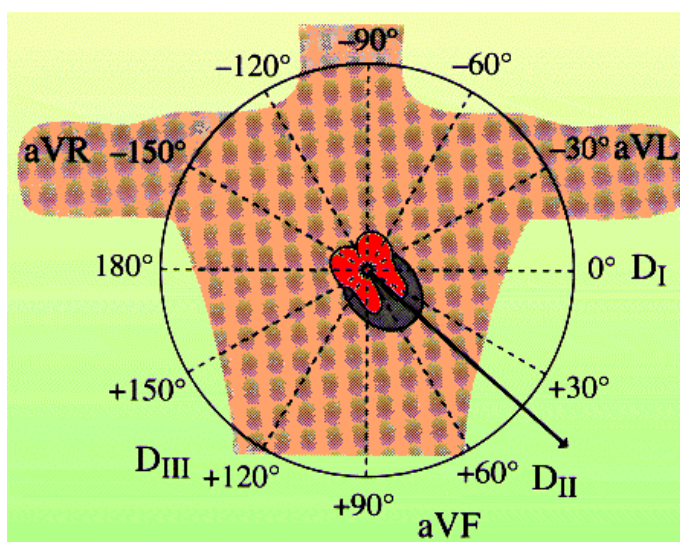


Figure (1.7) : Le triaxe des dériviations frontales.

1.6.2. Les dériviations précordiales

Ce sont des dériviations unipolaires fixées en des points définis sur la paroi thoracique désignés par Wilson.

On les nomme les dériviations standards : V1 à V6 :

- **V1** : est placée sur le 4^{ème} espace intercostal droit, au bord droit du sternum.
- **V2** : est placée sur le 4^{ème} espace intercostal gauche, au bord gauche du sternum.
- **V3** : est placée entre V2 et V4.
- **V4** : est placée sur le 5^{ème} espace intercostal gauche, sur la ligne médioclaviculaire.
- **V5** : est placée sur le 5^{ème} espace intercostal gauche, sur la ligne axillaire antérieure.
- **V6** : est placée sur le 5^{ème} espace intercostal gauche, sur la ligne axillaire moyenne.

Il est possible d'utiliser trois dérivations précordiales supplémentaires pour explorer la face postérieure du cœur : V7, V8, V9 qui sont à placer sur le 5^{ème} espace intercostal gauche, respectivement sur la ligne axillaire postérieure, sur la ligne médioscopulaire, et sur la ligne scapulo-vertébrale.

De même que pour les dérivations frontales, il est possible d'apercevoir les régions explorées par ces dérivations:

- V1 et V2 : les parois ventriculaires droite et septale.
- V3 et V4 : les parois antérieures du septum et du ventricule gauche.
- V5 et V6 : la paroi latérale du ventricule gauche.

La figure (1.8) montre ces dérivations et leur emplacement sur la paroi thoracique.

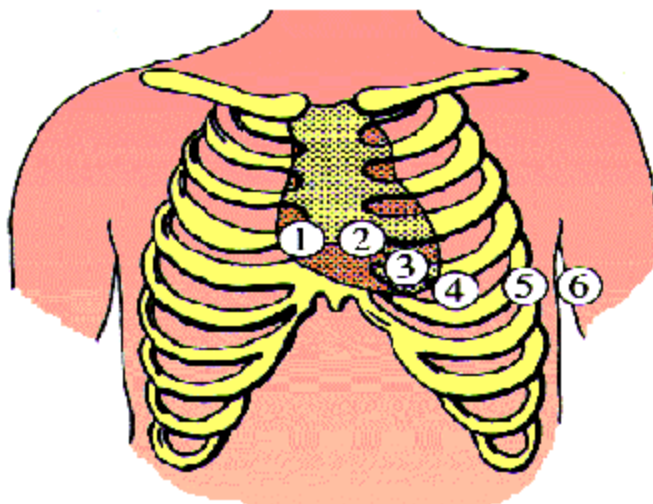


Figure (1.8) : Les dérivations précordiales.

1.7. Techniques d'enregistrement de l'ECG :

L'électrocardiographie sert à enregistrer l'activité électrique du cœur par l'intermédiaire des différentes électrodes exploratrices précédemment décrites, l'électrocardiogramme comprend 6 éléments principaux, Figure (1.9) [1] :

- Le circuit électrique constitué par l'ensemble des différentes électrodes exploratrices au contact du sujet, captant les différences de potentiel d'origine cardiaque.
- Un système amplificateur des impulsions nées dans le circuit précédent.
- Un galvanomètre.
- Un système inscripteur branché sur le galvanomètre, ayant une très faible inertie.
- Un fil de terre isolant, fixée sur le sujet à la jambe droite.
- Un fil de terre sur l'appareil qui élimine les courants parasites.

L'électrocardiogramme utilise du papier millimétré divisé en grandes divisions (5mm x 5mm) et petites divisions (1mm x 1mm). Une déflexion de 2 grandes divisions ou 10mm correspond à une tension de 1 mV. L'échelle du temps dépend de la vitesse de défilement du papier. Celle-ci est normalement de 25 mm/s, ce qui fait correspondre 0.20s à une grande division et 0.04s à une petite division. La vitesse de 50 mm/s est aussi utilisée pour mieux examiner les complexes QRS lorsque la fréquence cardiaque est trop élevée ou des détails particuliers de l'électrocardiogramme sont nécessaires. Les quatre électrodes des membres permettent d'obtenir les trois dérivations frontales D_I D_{II} l'électrode jaune correspond au bras gauche, la rouge au bras droit, la verte au pied gauche, la noire au pied droit. La plupart des appareils comportent un schéma, et certains ont les initiales anglaises des membres : RA (pour "right arm" ou bras droit), LA (pour "left arm" ou bras gauche), RL (pour "right leg" ou jambe droite), LL (pour "left leg" ou jambe gauche).

Que l'appareil soit manuel ou automatique, à une ou à trois pistes, les principes restent les mêmes, il faut :

1. régler le défilement sur 25 millimètres par seconde (vitesse par défaut).
2. enregistrer un étalonnage, qui trace un rectangle d'un centimètre de haut.
3. demander au patient de ne pas bouger et de respirer calmement.
4. enregistrer au moins trois complexes par dérivation, et un D_{II} long pour l'étude du rythme.
5. inscrire le nom, le prénom, la date de naissance ou l'âge du patient, et la date et l'heure de l'ECG.

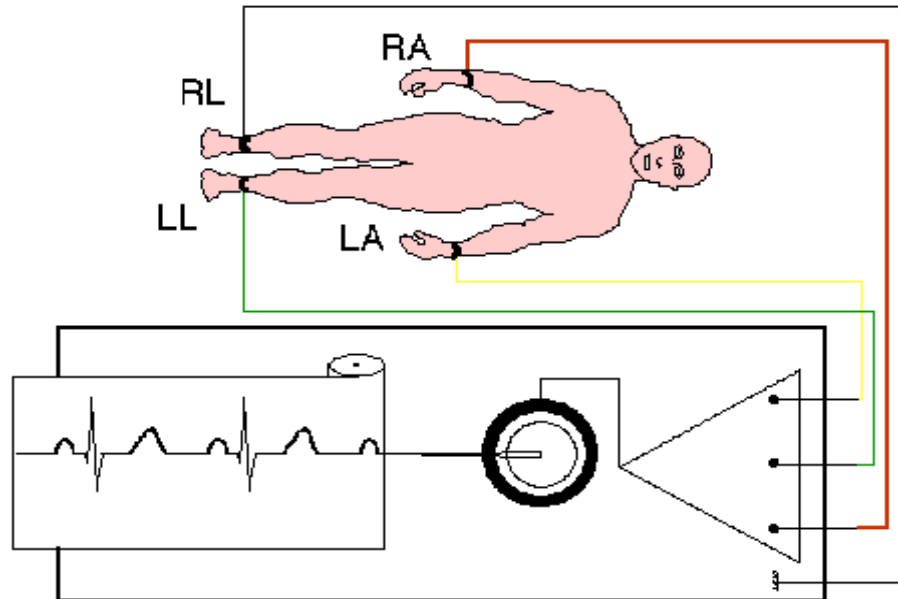


Figure (1.9) Système d'enregistrement d' ECG. Sur ce schéma, l' ECG se compose des trois dérivations bipolaires D_I , D_{II} et D_{III} .

1.8. Terminologies et normes

Sur un tracé électrocardiographique, le premier repère est la ligne isoélectrique. Elle est la ligne de base correspondant à l'absence de phénomène électrique. Au-dessus de celle-ci, on parle d'onde positive, en dessous, d'onde négative. Une onde peut être aussi diphasique si une partie de celle-ci se situe au-dessus et l'autre partie au-dessous de la ligne isoélectrique.

Toutes les ondes se mesurent du début de leur phase initiale, à la ligne isoélectrique.

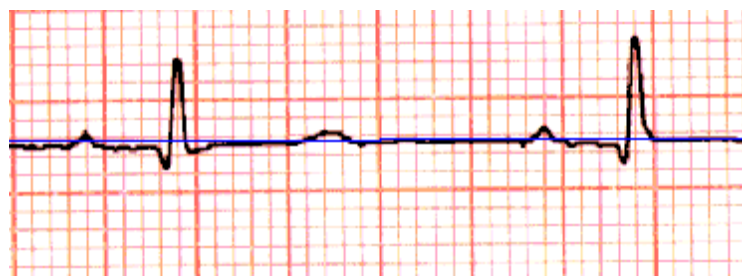


Figure (1.10) : la ligne isoélectrique.

1.8.1. Ondes enregistrées

- **L'onde P**

Elle est l'onde de dépolarisation auriculaire, le début de l'onde P correspond à la dépolarisation de l'oreillette droite, la fin de l'onde P correspond à la dépolarisation

de l'oreillette gauche. Elle est de forme arrondie, souvent positive, de faible amplitude (1 à 3 mvolts) et de moins de 0,12 seconde en D2.



Figure (1.11) : l'onde P.

- **Le complexe QRS**

Il correspond à l'activation et à la dépolarisation des ventricules de l'endocarde vers l'épicarde, il est constitué de trois segments :

L'onde Q : première déflexion négative : activation septale.

L'onde R : première déflexion positive : activation pariétale du VG.

L'onde S : déflexion négative qui suit l'onde R : activation basale du VG.

La durée de l'ensemble QRS varie de 0,06 à 0,1 seconde (3 à 5 petits carreaux) et se mesure du début du QRS jusqu'à la fin de l'onde S ou R, selon le cas. L'amplitude se mesure en mm et, par convention, une onde d'amplitude < 5 mm s'écrit en minuscules : q, r, s, cette convention permet de décrire différents aspects : qRS, QrS, QS, RS, rSr',..., etc.



Figure (1.12) : complexe QRS.

- **L'onde T**

Elle est la période de repolarisation ventriculaire, c'est l'inhibition de l'excitation ventriculaire de l'épicarde vers l'endocarde. Elle est asymétrique, d'une branche

ascendante légèrement oblique et d'une branche descendante plus abrupte. Son amplitude est inférieure à 2 mm, et ne témoigne d'aucun événement mécanique. C'est un phénomène purement électrique.

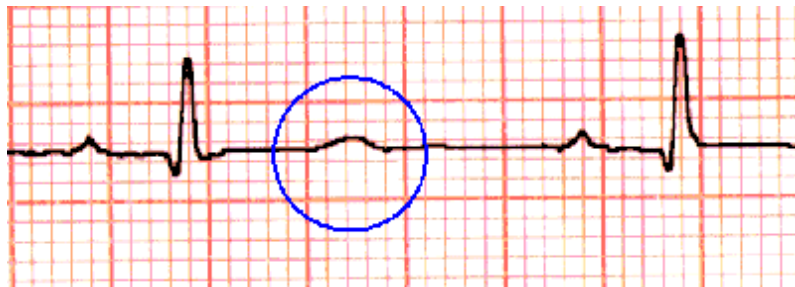


Figure (1.13) : L'onde T.

- **L'onde U**

C'est le témoin d'une repolarisation tardive de zones myocardiques d'amplitude inscrite entre celle de l'onde P et de celle de l'onde T. Elle est inférieure à $\frac{1}{4}$ de l'amplitude de l'onde T.



Figure (1.14): L'onde U.

1.8.2. Intervalles temporels

- **Le segment PR**

Il correspond à la pause d' $\frac{1}{10}$ de seconde entre l'activation auriculaire et l'activation ventriculaire, par le passage de l'influx du nœud auriculo-ventriculaire au faisceau de His, il se mesure de la fin de l'onde P jusqu'au début du QRS et correspond à 0.03 à 0.04 secondes (moins de 2 petits carreaux).



Figure (1.15) : Le segment PR.

- **Le segment ST**

Il correspond à la période d'excitation uniforme des ventricules jusqu'à la phase de récupération des ventricules. On le mesure de la fin de l'onde S ou R jusqu'au début de l'onde T. Il est normalement horizontal ou légèrement oblique plus au moins isoélectrique. Un sus-décalage ou un sous-décalage de plus d'1 mm par rapport à la ligne isoélectrique est anormal.



Figure (1.16) : Le segment ST.

- **L'intervalle PR**

C'est le temps de conduction auriculo-ventriculaire. C'est le temps nécessaire à l'influx pour dépolariser les oreillettes puis franchir le Nœud auriculo-ventriculaire et le tronc du faisceau de His., il se calcule à partir du début de l'onde P en allant jusqu'au début du QRS, il est de 0,12 à 0,23 secondes.



Figure (1.17) : L'intervalle PR.

- **L'intervalle QT**

C'est l'intervalle de dépolarisation (QRS), d'excitation (ST) et de repolarisation (T) des ventricules, il se mesure du début du QRS jusqu'à la fin de l'onde T.

Le QT est fonction de la fréquence cardiaque; c'est pourquoi il est préférable d'utiliser le **QT corrigé** (QTc), Qui est normalement inférieur ou égal à 0.44 seconde.

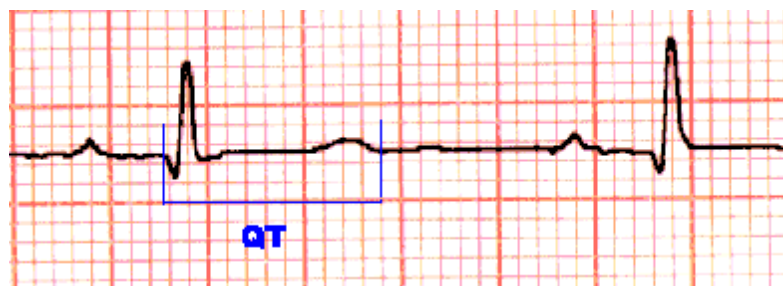


Figure (1.18) : L'intervalle QT.

1.9. Artefacts dans l'électrocardiogramme

En médecine, le terme artefact désigne tout signal indésirable enregistré simultanément avec l'électrocardiogramme par les électrodes. Les artefacts sont d'origines diverses [2]:

- Interférences avec le secteur 50 Hz (ou 60Hz suivant le pays) : ce type de bruit est caractérisé par la superposition à l'électrocardiogramme d'une composante de 50 Hz (ou de 60 Hz) et de ses harmoniques.
- Artefacts de mouvement : les artefacts de mouvement sont des changements transitoires (mais pas abrupts) de la ligne de base causés par la variation de l'impédance entre électrode et la peau. Lorsque cet impédance varie, l'amplificateur de l'enregistreur ECG trouve à son entrée une impédance de la source différente et par conséquent une tension d'entrée différente.
- Mauvais contact entre les électrodes et la peau : cette perturbation est due à une perte de contact entre l'électrode et la peau.
- Contraction musculaire (EMG) : ces signaux résultent de l'activité électrique des muscles.
- Saturation des amplificateurs.
- Bruit respiratoire : la respiration du patient se traduit par la superposition à l'ECG de variations de basses fréquences.

Les électrocardiogrammes Holter analogiques sont en plus parasités par autres artefacts :

- Artefacts dus au déroulement du support magnétique : ces artefacts résultent de la survenue d'une diminution de la vitesse de déroulement de la cassette. Ils donnent sur l'ECG l'aspect de fausse tachycardie. Plus rares sont les artefacts liés à une augmentation anormale de la vitesse de déroulement de la cassette qui donnent un aspect de fausse bradycardie.
- Artefacts liés au défaut du support magnétique : ces artefacts sont dus à une altération du support liée à une perte de substance magnétique.

1.10. Le Holter cardiaque

Son nom provient du Dr. *Norman Holter* qui créa cette technique en 1949, sous forme d'une valise portable d'un peu plus de 30kg. L'enregistrement du signal électrique cardiaque a été ensuite fait de manière analogique sur une simple cassette audio de 60 minutes, tournant au ralenti. La fin des années 80 vit apparaître les premiers enregistrements numériques, d'abord sur puces, puis sur cartes mémoires amovibles.

L'holter comporte deux parties :

- Un boîtier d'enregistrement confié au patient et relié à ce dernier par des électrodes (cinq le plus souvent, permettant d'enregistrer deux dérivations à peu près orthogonales, la dernière électrode faisant office de neutre) fixées à la peau par un adhésif. Ce boîtier comporte un bouton que le sujet peut actionner s'il ressent quelque chose durant l'enregistrement (douleurs thoraciques ou autre) : un repère sera inscrit alors sur l'enregistrement, permettant au médecin d'aller directement consulter le tracé à ce moment et en faire ainsi le diagnostic.
- Une console de traitement : il s'agit d'un ordinateur muni d'un logiciel permettant la visualisation et l'analyse semi-automatique de l'enregistrement.

Contrairement aux techniques d'imagerie médicale, il n'existe pas actuellement de norme informatique de stockage permettant l'échange de données entre différents systèmes.

1.10.1. Limitations

L'enregistrement n'excède pas quelques jours pour des raisons de taille mémoire du boîtier, mais aussi, de qualité du signal électrique qui se dégrade si les électrodes sont laissées longtemps en place.

Le problème essentiel reste la gestion informatisée des artefacts mal gérés par les logiciels actuels, et qui peuvent gêner considérablement l'interprétation de l'examen.

1.10.2. Différents types de Holters

Les Holters modernes peuvent enregistrer entre deux et douze pistes. On est dans ce dernier cas, limité par le nombre d'électrodes (dix) et par le câble devenant plus difficilement maniable.

Certains boîtiers comportent leur propre système d'analyse ce qui permet théoriquement de se passer de la console de lecture et d'avoir ainsi un résultat dès la fin de l'enregistrement. La fiabilité de ces systèmes reste cependant limitée.

L'holter implantable consiste en un boîtier métallique de la taille d'un petit briquet qui est implanté sous la peau, sous anesthésie locale. Elle permet un enregistrement pouvant aller jusqu'à un an.

2.1. Introduction

L'informatique est la science du traitement automatique de l'information, son développement est souvent confondu avec celui des machines de traitements : les ordinateurs. Depuis 1946 jusqu'à aujourd'hui, les ordinateurs sont devenus de plus en plus puissants, cependant cette augmentation de puissance ne permet pas de toujours résoudre les problèmes d'une application informatique dans un domaine particulier. La construction de logiciels s'appuie sur plusieurs approches, deux parmi les plus utilisées sont l'approche algorithmique et l'approche basée sur la connaissance, malheureusement, ces deux approches ne suffisent pas à répondre à tous les problèmes existants.

Les réseaux de neurones biologiques réalisent facilement un certain nombre d'applications telles que la reconnaissance de formes, le traitement du signal, l'apprentissage par l'exemple, la mémorisation et la généralisation. Ces applications sont pourtant, malgré tous les efforts déployés en algorithmique et en intelligence artificielle, à la limite des possibilités actuelles. C'est à partir de l'hypothèse que le comportement intelligent émerge de la structure et du comportement des éléments de base du cerveau que les réseaux de neurones artificiels se sont développés. Les réseaux de neurones artificiels sont des modèles basés sur un modèle simplifié du neurone, à ce titre ils peuvent être décrit par leurs composants, leurs variables descriptives et les interactions des composants. Ce modèle permet certaines fonctions du cerveau, comme la mémorisation associative, l'apprentissage, ...etc.

Dans ce chapitre, on va donner un aperçu général sur les réseaux de neurones et les différents modèles mathématiques proposés et leurs utilisations dans les systèmes non linéaires, en commençant par un brève historique qui résumera le développement des réseaux de neurones.

2.2. Historique

- **Apparition :**

Tout commence en 1943, lorsque deux biophysiciens de l'université de Chicago Mc.Culloch et Pitts, s'inspirant des récentes découvertes en neurobiologie, conçoivent le premier modèle du neurone biologique, baptisé neurone formel ou automate à seuil.

Ils donnent leurs noms à une modélisation du neurone biologique (un neurone au comportement binaire). Ce sont les premiers à montrer que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes.

Leurs travaux n'ont pas donné d'indication sur une méthode pour adapter les coefficients synaptiques. Un peu plus tard, un neurophysiologiste renommé, Donald Hebb propose en 1949 une formulation du mécanisme d'apprentissage, sous la forme d'une règle de modification des connexions synaptiques qui porte encore son nom.

- **Les premiers succès**

Les premiers succès de cette discipline remontent à 1957, lorsque F. Rosenblatt, combinant les idées de ses prédécesseurs, conçoit le Perceptron, un réseau de neurones artificiels inspiré du système visuel, possédant une couche de neurones « perceptive » et une couche de neurones « décisionnelle ». Ce réseau, qui parvient à apprendre à identifier des formes simples et à calculer certaines fonctions logiques, constitue le premier système artificiel exhibant une faculté jusque là réservé au vivant, la capacité d'apprendre par l'expérience; le premier réseau de neurones artificiel proprement dit.

Les travaux de Rosenblatt suscitent au début des années 60 un vif enthousiasme chez les scientifiques alors fortement impliqués dans la recherche sur l'intelligence artificielle, c'est alors qu'en 1960, l'automaticien Widrow développe le modèle Adaline (Adaptative Linear Element). Dans sa structure, le modèle ressemble au Perceptron, cependant la loi d'apprentissage est différente, celle-ci est à l'origine de l'algorithme de rétropropagation de gradient très utilisé aujourd'hui avec les Perceptrons Multi Couches.

Cet enthousiasme se voit pourtant brusquement refroidi en 1969 lorsque deux scientifiques américains de renom, Minsky et Papert, publient un livre qui, au terme d'une analyse mathématique approfondie, met à jour les limites intrinsèques du Perceptron, en particulier son incapacité à résoudre les problèmes non linéairement séparables, tels que le célèbre problème du XOR. Ils étendent implicitement ces limitations à tous modèles de réseaux de neurones artificiels, leur objectif est atteint, il y a abandon financier des recherches dans le

domaine (surtout aux U.S.A.), les chercheurs se tournent principalement vers l'Intelligence Artificielle et les systèmes à bases de règles.

- **L'ombre (1967-1982)**

Toutes les recherches ne sont, bien sûr, pas interrompues. Elles se poursuivent, mais déguisées, sous le couvert de divers domaines comme : le traitement adaptatif du signal, la reconnaissance de formes, la modélisation en neurobiologie, etc.

- **Le renouveau**

En 1982, J. J. Hopfield, physicien reconnu, donna un nouveau souffle au neuronal en publiant un article introduisant un nouveau modèle de réseau de neurones (complètement récurrent). Cet article eut du succès pour plusieurs raisons, dont la principale était de teinter la théorie des réseaux de neurones de la rigueur propre aux physiciens. Le neuronal redevint un sujet d'étude acceptable, bien que le modèle de Hopfield souffrait des principales limitations des modèles des années 1960, notamment l'impossibilité de traiter les problèmes non linéaires.

- **La levée des limitations**

Parallèlement aux travaux de Hopfield, Werbos conçoit un mécanisme d'apprentissage pour les réseaux multicouches de type perceptron : c'est l'algorithme d'apprentissage par rétropropagation de l'erreur "Back-propagation" qui fournit un moyen simple d'entraîner les neurones des couches cachées. Cet algorithme sera réellement popularisé en 1986 par Rumelhart dans un article de Nature et un livre ("Parallel Distributed Processing") qui a longtemps constitué la "bible" des connexionnistes.

A la même date, les approches algorithmiques de l'intelligence artificielle furent l'objet de désillusion, leurs applications ne répondant pas aux attentes. Cette désillusion motiva une réorientation des recherches en intelligence artificielle vers les réseaux de neurones (bien que ces réseaux concernent la perception artificielle plus que l'intelligence artificielle à proprement parler). La recherche fut relancée et l'industrie reprit quelque intérêt au neuronal.

Une révolution survient alors dans le domaine des réseaux de neurones artificiels : une nouvelle génération de réseaux de neurones, capables de traiter avec succès des phénomènes non-linéaires.

2.3. Définition des réseaux de neurones

2.3.1. Le neurone artificiel

Un neurone formel, ou neurone, est une fonction algébrique non linéaire et bornée, dont la valeur dépend de paramètres appelés coefficients ou poids w , abréviation de weight (poids). Les variables de cette fonction sont habituellement appelées «entrées» du neurone, et la valeur de la fonction est appelée «sortie». Un neurone est donc avant tout un opérateur mathématique, dont on peut calculer la valeur numérique par quelques lignes de programme informatique. Il est très rarement réalisé physiquement sous la forme d'un objet (circuit électronique par exemple). Il est cependant pratique de le représenter graphiquement.

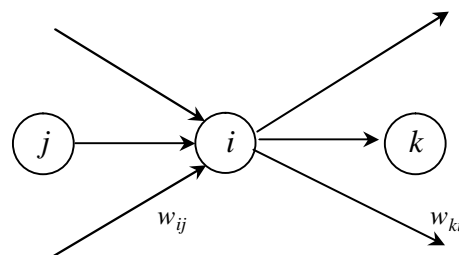


Figure (2.1) : Structure d'un neurone artificiel.

Pour le neurone d'indice i , les entrées sur celui-ci ont de poids w_{ij} alors que les connexions avals ont de poids w_{ki} .

Comme, nous avons mentionné auparavant que les réseaux de neurones ont été inspirés du comportement des éléments de base du cerveau, la figure suivante schématise la correspondance entre le neurone biologique et le neurone artificiel.

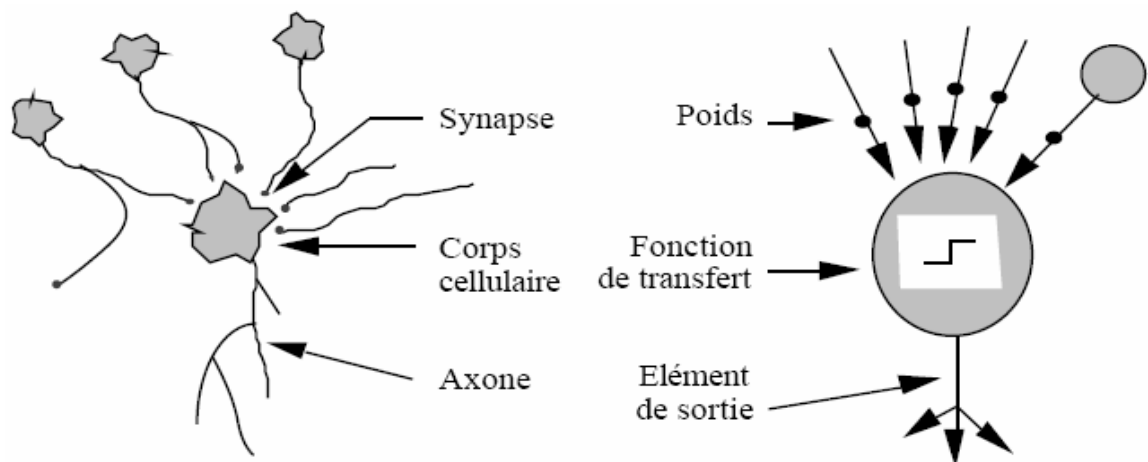


Figure (2.2) : Mise en correspondance neurone biologique / neurone artificiel.

Par analogie avec le neurone biologique, le comportement du neurone artificiel se compose de deux phases :

La première phase, appelée activation, représente le calcul de la somme pondérée des entrées selon l'expression suivante :

$$v_i = \sum_{j=1}^n w_{ij} x_j + b_i \quad (2.1)$$

Où :

x_j : entrée j du $i^{\text{ème}}$ neurone.

w_{ij} : pondération ou poids associée à l'entrée j du neurone i .

b_i : seuil interne ou biais du neurone i .

v_i : sortie du neurone i .

A partir de cette valeur, une fonction de transfert calcule la valeur de l'état du neurone, par la relation suivante :

$$y_k = f_k(v_i) \quad (2.2)$$

C'est cette valeur qui sera transmise aux neurones avals. Il existe de nombreuses formes possibles pour la fonction de transfert f , les plus courantes sont présentées sur la figure (2.3).

On remarquera qu'à la différence des neurones biologiques dont l'état est binaire, la plupart des fonctions de transfert sont continues, offrant une infinité de valeurs possibles comprises dans l'intervalle $[0, +1]$ ou $[-1, +1]$.

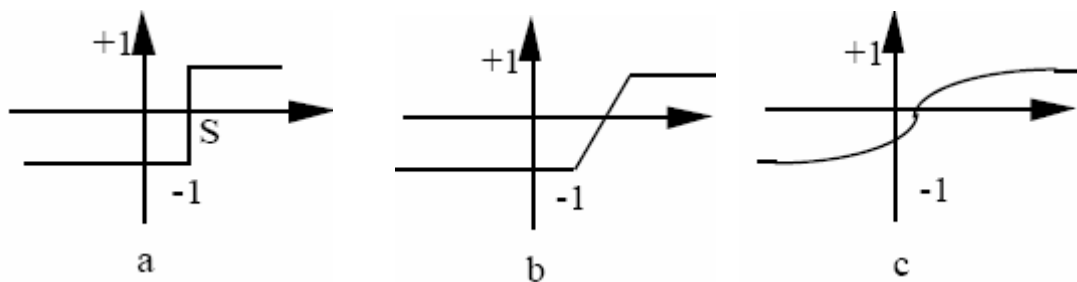


Figure (2.3) : Différents types de fonctions de transfert pour le neurone artificiel,
a : fonction à seuil(S, la valeur du seuil), b : linéaire par morceaux , c : sigmoïde.

2.3.2. Les réseaux de neurones

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie

unique sur la base des informations qu'il reçoit, dont le résultat reproduit ou prévoit aussi fidèlement que possible, le comportement de n'importe quel processus en fonction des facteurs qui déterminent ce comportement. On entend par «processus» tout système, naturel ou artificiel, et par «facteur» toute grandeur qui est susceptible d'avoir une influence sur le processus. Toute structure hiérarchique de réseaux est évidemment un réseau [3].

2.4. Structure d'interconnexion

Les connexions entre les neurones qui composent le réseau décrivent la topologie du modèle, elle peut être quelconque, mais le plus souvent il est possible de distinguer une certaine régularité, on distingue deux types de réseaux de neurones : les réseaux non bouclés et les réseaux bouclés.

2.4.1. Les réseaux non bouclés

Un réseau de neurones non bouclé est donc représenté graphiquement par un ensemble de neurones connectés entre eux, l'information circulant des entrées vers les sorties sans retour en arrière: si l'on se déplace dans le réseau, à partir d'un neurone quelconque, en suivant les connexions, on ne peut pas revenir au neurone de départ, les neurones qui effectuent le dernier calcul de la composition de fonctions sont les neurones de sortie ; ceux qui effectuent des calculs intermédiaires sont les neurones cachés [4].

Il n'y a pas de connexion entre neurones d'une même couche et les connexions ne se font qu'avec les neurones des couches avales, figure (2.4). Habituellement, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante et celle-ci seulement.

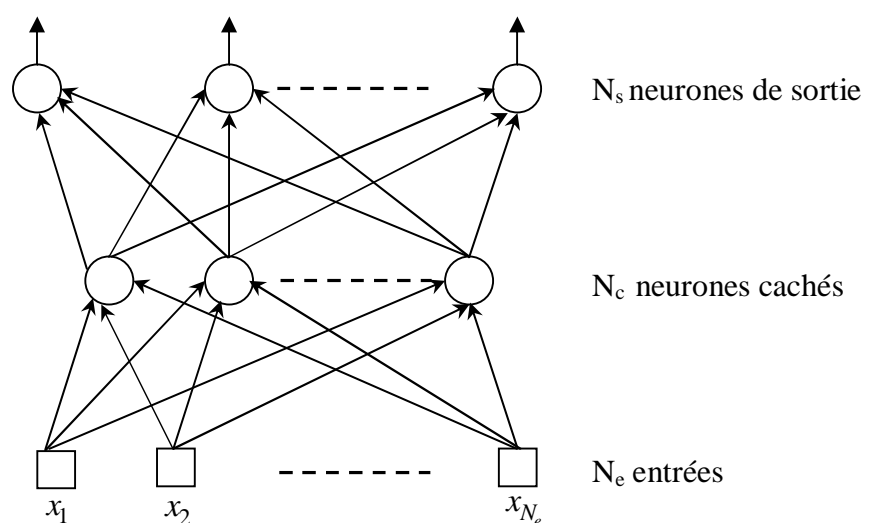


Figure (2.4) : réseau multicouche.

Les réseaux de neurones non bouclés à couches, dont les neurones cachés ont une fonction d'activation sigmoïde, sont souvent appelés perceptrons multicouche (ou MLP pour Multi Layer Perceptron), grâce à ces plusieurs couches de traitement leurs permettent de réaliser des associations non linéaires entre l'entrée et la sortie, ils sont ainsi capables de résoudre le cas de « XOR », on sait depuis les années soixante que les possibilités de traitement des réseaux multicouche sont supérieures à celles du Perceptron, cependant l'algorithme d'apprentissage manquait pour la couche de sortie, on peut appliquer l'apprentissage du Perceptron, mais comment modifier les poids pour les connexions qui ne sont pas en relation avec un neurone de sortie ? [5].

Les réseaux non bouclés sont 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 modélisation statique de processus non linéaires ou de classification par opposition aux réseaux bouclés ou « dynamiques ».

2.4.2. Les réseaux bouclés

Contrairement aux précédents, les réseaux de neurones bouclés peuvent avoir une topologie de connexions quelconque, la connectivité des unités dans les réseaux de neurones récurrents ne se limite pas, comme dans le cas des réseaux à propagation avant (feedforward), à des architectures dans lesquelles l'information se propage de l'entrée vers la sortie, couche après couche. Tout type de connexion est admis, c'est à dire d'un neurone à n'importe quel autre, y compris lui-même. En d'autres termes, lorsqu'on se déplace dans le réseau en suivant le sens des connexions, il est possible de trouver au moins un chemin qui revient à son point de départ. Un tel chemin est désigné sous le terme de cycle.

Une grandeur, à un instant donné, ne peut pas être fonction de sa propre valeur au même instant, par conséquent, tout cycle du graphe des connexions d'un réseau de neurones récurrent doit comprendre au moins une connexion de retard unitaire [4].

Un réseau de neurones bouclé est donc un système dynamique, régi par des équations différentielles. Comme l'immense majorité des applications sont réalisées par des programmes informatiques, on se trouve dans le cadre des systèmes à temps discret, c'est-à-dire des systèmes où les valeurs des variables ne sont connues qu'à des instants régulièrement espacés, les équations différentielles qui régissent les systèmes dynamiques à temps continu sont alors remplacées par des équations aux différences, ceci donne lieu à des comportements dynamiques fort intéressants mais qui peuvent être très complexes, contrairement aux réseaux de neurones statiques où cette dynamique est totalement absente.

Tout réseau de neurone bouclé, aussi complexe soit-il, peut être mis sous une «forme canonique», constituée d'un réseau non bouclé et de connexions qui relient une ou plusieurs sorties (dites sorties d'état) de celui-ci à une ou plusieurs entrées (dites entrées d'état, par opposition aux entrées externes qui sont des signaux provenant de l'environnement du processus, par exemple des signaux de commande), avec un retard d'une unité de temps.

Les réseaux de neurones bouclés sont utilisés pour effectuer des tâches de modélisation dynamique de processus non linéaires, de commande de processus ou de filtrage.

2.4.2.1. Principales architectures de réseaux récurrents :

Les réseaux récurrents ont des connexions feedback ou bien à partir des couches cachées ou de couche de sortie aux couches précédentes, ces feedbacks stockent l'information à partir des instants du temps précédents, donc ils peuvent être retardés et utilisés dans les instants de temps subséquents.

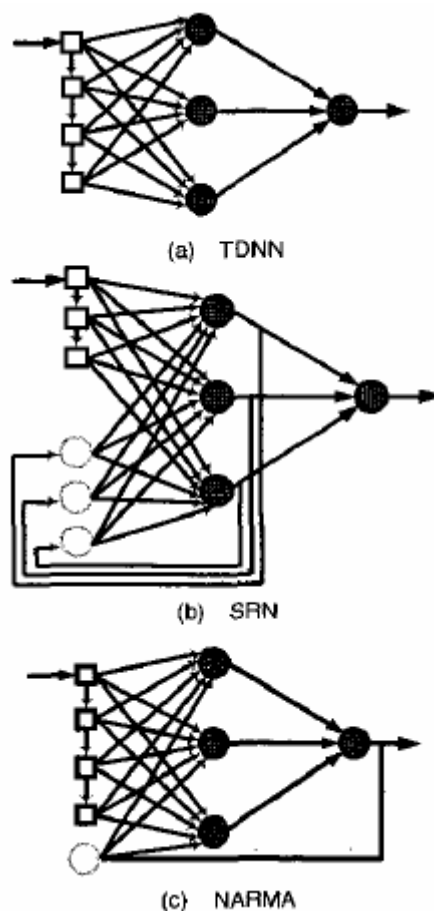


Figure (2.5) : architectures des réseaux neurones récurrents [6].

Donc ils peuvent être considérés comme une implémentation non-linéaire des modèles auto-régressifs à moyenne glissante (ARMA : autoregressive moving average), la figure (2.5) schématise trois architectures de réseaux de neurones récurrents :

a) TDNN (Time Delay Neural Network) : la mémoire des entrées précédentes est codée en utilisant une ligne de branchement de retard des entrées, ceci est une représentation du modèle auto-régressif.

b) SRN (Simple Recurrent network) : les connections feedback existent entre les couches cachées et le nœud récurrent, et ils sont entrés à la couche cachée elle-même à l'instant du temps suivant.

c) NARMA (Nonlinear ARMA) : les connections feedback sont fournies par la couche de sortie, laquelle alimente les sorties courantes de la couche cachée à l'instant suivant [6].

d) Le réseau de Hopfield (1982)

Le réseau de Hopfield consiste d'un ensemble de N neurones interconnectés, lesquels mettent à jour leurs valeurs d'activation en asynchrone et indépendamment des autres neurones ; tous les neurones sont des entrées et des sorties (réseau connecté complètement) [7].

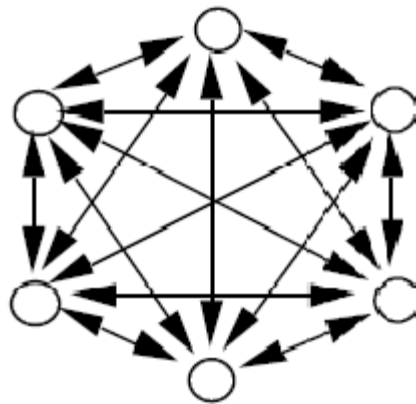


Figure (2. 6) : réseau de Hopfield.

e) Architecture de Jordan (Jordan, 1986)

Dans l'architecture proposée par Jordan, les unités de la couche de sortie sont dupliquées sur une couche appelée couche de contexte. Les unités de cette couche tiennent également compte de leur propre état à l'instant précédent. Cette connexion récurrente d'une unité de contexte à elle-même lui donne une dynamique ou une mémoire individuelle [6]. L'activation de chaque unité de cette couche est calculée selon l'équation suivante :

$$C_i(t+1) = aC_i(t) + O_i(t) \quad (2.3)$$

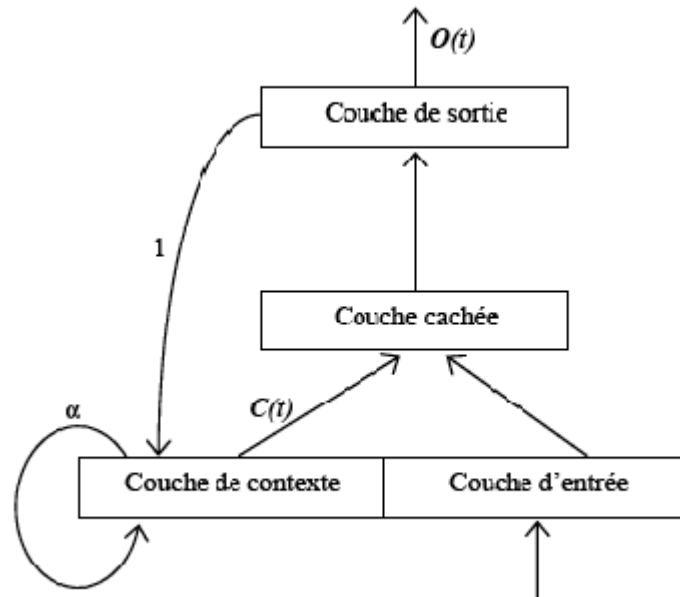


Figure (2.7) : architecture de Jordan [3].

f) Architecture d'Elman (Elman, 1990)

Elman s'est inspiré en grande partie du réseau de Jordan pour proposer son architecture montrée sur la figure (2.8), cette fois-ci, ce sont les unités de la couche cachée qui sont dupliquées dans la couche contexte avec un poids unitaire. L'apprentissage s'effectue par l'algorithme de rétropropagation et ne concerne que les poids de propagation avant [3].

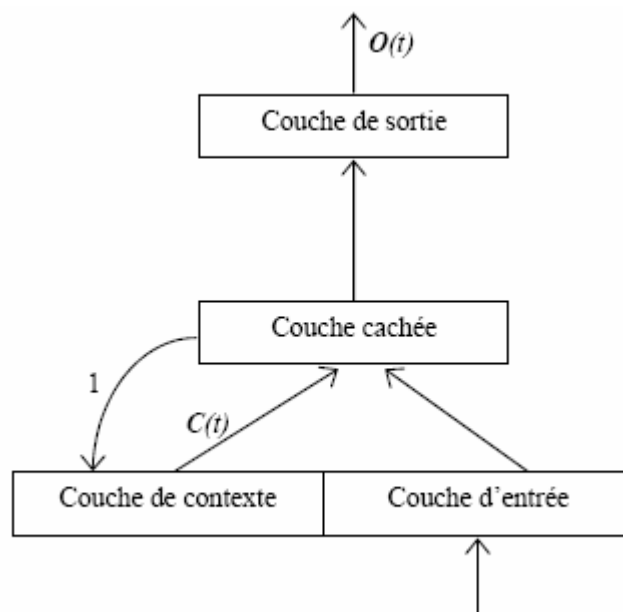


Figure (2. 8) : architecture d'Elman [3].

2.5. Propriétés des réseaux de neurones

Par analogie avec les réseaux de neurones biologiques, les réseaux de neurones artificiels tentent de reproduire les caractéristiques les plus importantes du comportement biologique, à savoir l'apprentissage, la généralisation et l'association.

2.5.1. L'apprentissage

L'apprentissage est vraisemblablement la propriété la plus intéressante des réseaux neuronaux ; c'est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré.

Au niveau des algorithmes d'apprentissage, il a été défini deux grandes classes selon que l'apprentissage est dit supervisé ou non supervisé.

- Dans le cas de l'apprentissage supervisé, un ensemble de données contenant une collection d'information représentant l'entrée et la valeur désirée ou sortie désirée. Cet ensemble d'apprentissage est présenté au réseau à maintes reprises jusqu'à ce que la différence entre la sortie désirée et la sortie actuelle du réseau atteigne une valeur prédéterminée.
- Dans le cas de l'apprentissage non supervisé, la valeur cible, ou réponse, n'est pas fournie et l'information dans l'ensemble d'apprentissage est présentée d'une façon continue jusqu'à ce qu'un certain critère de convergence est satisfait .

Le réseau de neurones construit des modèles internes qui capturent les régularités dans le modèle d'entrée.

2.5.2. Généralisation

La capacité de généralisation est l'une des raisons qui motivent l'étude et le développement des réseaux de neurones artificiels. Elle peut être définie par la capacité d'élargir les connaissances acquises après apprentissage à des données nouvellement rencontrées par le réseau de neurones, c'est de cette façon que les réseaux de neurones sont capables d'approximer une fonction uniquement à partir d'une partie des données, ou encore d'associer un vecteur d'entrée x qui n'a pas fait l'objet d'un apprentissage.

On peut distinguer deux types de généralisation : locale et globale.

- Dans l'approche locale, chaque neurone est associé à une région d'activation localisée dans l'espace des données. Seule une partie des neurones participe donc à la réponse du réseau.

- Par contre, dans l'approche globale, l'ensemble des neurones du réseau participe à l'élaboration de la sortie du réseau, l'information est donc distribuée dans le réseau tout entier. C'est le cas des réseaux de neurones de type Perceptron Multi Couches ou le modèle de Hopfield.

On parle de sur-apprentissage (over-fitting) quand le réseau a trop parfaitement appris les exemples proposés. Il sera donc incapable de généraliser, en pratique, on effectue un apprentissage sur un sous ensemble S de l'espace de données D , le réseau est alors testé sur un ensemble de test T ne faisant pas partie de l'apprentissage.

On calcule alors la moyenne des erreurs quadratiques sur l'ensemble S appelée « erreur base apprentissage » et sur l'ensemble de test T appelée « erreur base test ». Plus on agrandit l'ensemble S , plus l'erreur base apprentissage diminue, plus l'erreur base test augmente. Le réseau perd dans ce cas là ses capacités de généralisation. La Figure (2.9) illustre clairement ce compromis entre sur-apprentissage, sous-apprentissage et bonne généralisation.

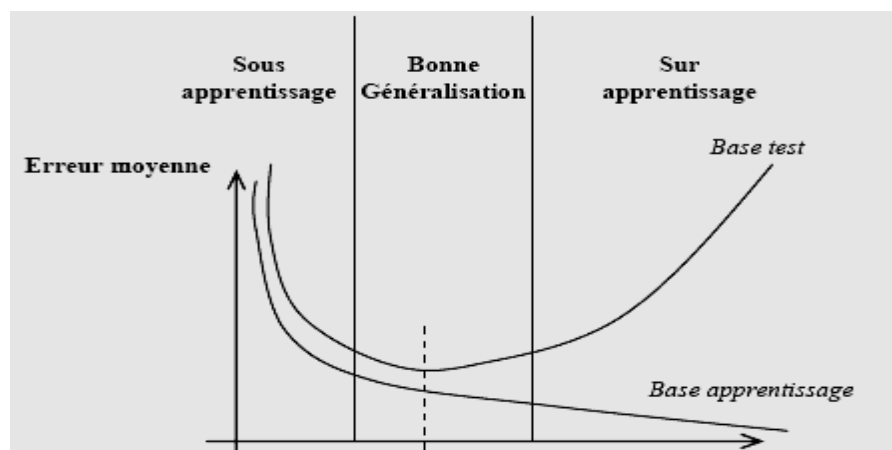


Figure (2.9) : compromis entre erreur apprentissage et erreur test [4].

2.5.3. L'approximation parcimonieuse

Un réseau de neurones, tel que nous l'avons définis plus haut, possède une propriété remarquable qui est à l'origine de son intérêt pratique: il est un « approximateur universel parcimonieux », qu'est capable d'imiter n'importe quel processus, après ajustement de ses paramètres par apprentissage; la propriété d'« approximation » est la traduction mathématique de cette capacité d'imitation. De plus, la « parcimonie » exprime le fait que le réseau a besoin d'un petit nombre de paramètres ajustables pour réaliser correctement sa tâche.

De manière plus précise, la propriété d'approximation peut être énoncée de la façon suivante: toute fonction bornée suffisamment régulière peut être approchée avec une

précision arbitraire, dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini, possédant tous la même fonction d'activation, et un neurone de sortie linéaire, cette propriété n'est pas spécifique des réseaux de neurones: bien d'autres familles de fonctions paramétrées possèdent cette propriété. La spécificité des réseaux de neurones réside dans le caractère «parcimonieux» de l'approximation: à précision égale, les réseaux de neurones nécessitent moins de paramètres ajustables que les autres approximateurs connus; plus précisément, le nombre de poids varie linéairement avec le nombre de variables de la fonction à approcher, alors qu'il varie exponentiellement pour la plupart des autres approximateurs. C'est cette remarquable parcimonie qui justifie l'intérêt pratique des réseaux de neurones: dès qu'un problème fait intervenir plus de deux variables, les réseaux de neurones sont, en général, préférables aux autres méthodes.

2.6. Les différents algorithmes d'apprentissage

2.6.1. Règle de Delta

Cette règle a été appliquée à l'adaline (adaptive linear element) qui est un neurone qui possède des valeurs d'activation continues et une fonction d'activation linéaire [4] :

$$v_i = \sum_j w_{ij} x_j + b_i \quad (2.4)$$

v_i : Activation du neurone i .

b_i : Biais du neurone i .

x_j : entrée extérieur j .

Pour n'importe quel échantillon d'entrée la sortie diffère de la valeur désirée par $(d_i - v_i)$ la règle delta maintenant utilise une fonction de coût basée sur ces différences pour ajuster les poids.

$$E = \sum_i E_i = \frac{1}{2} \sum_i (d_i - v_i)^2 \quad (2.5)$$

La règle delta trouve toutes les valeurs des poids qui minimisent la fonction d'erreurs par une méthode dite descente du gradient, l'idée est de faire changer les poids proportionnellement au négatif de la dérivée de l'erreur mesuré sur chaque modèle en fonction de chaque poids :

$$\Delta_i w_j = -h \frac{\partial E_i}{\partial w_j} \quad (2.6)$$

h : pas d'apprentissage.

La dérivée est :

$$\frac{\partial E_i}{\partial w_j} = \frac{\partial E_i}{\partial v_i} \frac{\partial v_i}{\partial w_j} \quad (2.7)$$

D'après l'équation (2.4)

$$\frac{\partial v_i}{\partial w_j} = x_j \quad (2.8)$$

et

$$\frac{\partial E_i}{\partial v_i} = -(d_i - v_i) \quad (2.9)$$

tel que

$$\Delta_i w_j = h d_i x_j \quad (2.10)$$

Où $d_i = d_i - v_i$ est la différence entre la sortie désirée et la sortie actuelle pour le modèle i .

2.6.2. La rétropropagation

La règle d'apprentissage de Widrow-Hoff (règle de Delta) ne peut s'appliquer que sur des réseaux à une seule couche de poids adaptatifs. C'est donc en étendant la règle de Widrow-Hoff que plusieurs équipes de chercheurs (Le Cun, 1985) et (Werbos, 1974) ont développé un algorithme d'apprentissage appelé rétropropagation du gradient de l'erreur, généralisé ensuite par l'équipe de Rummelhart en 1986. Cet algorithme fournit une façon de modifier les poids des connexions de toutes les couches d'un Perceptron Multi Couches (PMC) [3].

Soit le réseau à deux couches décrit par la figure (2.10) dans lequel les unités de sortie sont notées y_k , les unités cachées v_i et les unités d'entrée x_j , et d_k sont les sorties désirées.

Les connexions des unités d'entrée aux unités cachées sont notées w_{jk} et celles des unités cachées aux unités de sortie par w_{ij} .

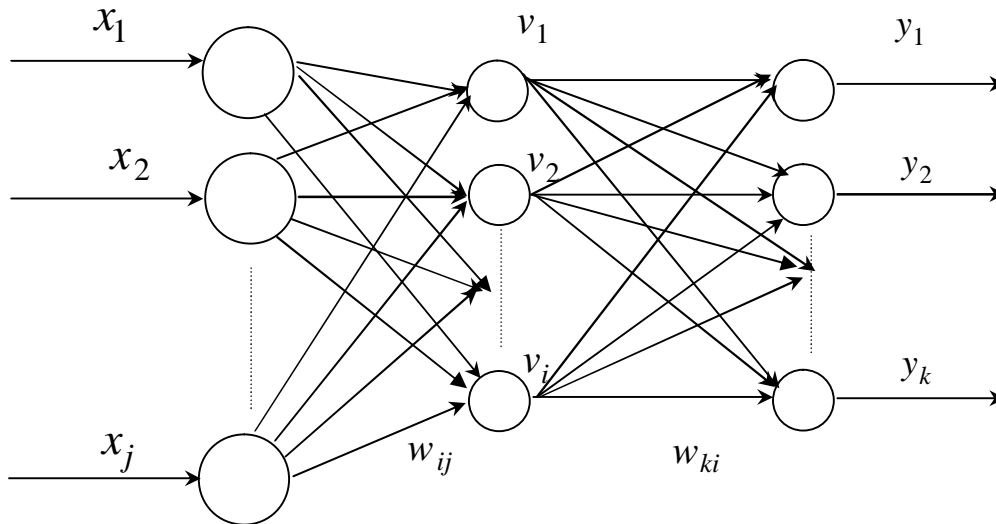


Figure (2.10) : Perceptron Multi Couches.

L'entrée j a pour valeur x_j^p lorsque le modèle de données p est présenté au réseau. Ces valeurs peuvent être binaires (0/1 ou +1/-1) ou continues, la valeur de sortie de l'unité cachée i est donnée par :

$$v_i^p = f_i(a_i^p) = f_i\left(\sum_j w_{ij}x_j^p\right) \quad (2.11)$$

Les unités de sortie ont comme valeur :

$$y_k^p = f_k(a_k^p) = f_k\left(\sum_i w_{ki}v_i^p\right) \quad (2.12)$$

Les fonctions d'erreurs partielles et globales sont alors définies par :

$$E^p = \frac{1}{2} \sum_i (y_k^p - d_k^p)^2 \quad \text{et} \quad E = \sum_p E^p \quad (2.13)$$

La minimisation de la fonction d'erreur globale va se faire par une descente de gradient, par conséquent, après la présentation de tous les vecteurs d'entrée de la base d'apprentissage, nous modifierons la valeur de chaque connexion par la règle :

$$\Delta w = -\eta \frac{\partial E}{\partial w} \quad (2.14)$$

Cette règle d'apprentissage est généralement appelée la règle de delta généralisée.

Dans l'expression (2.13), seule la sortie y_k dépend du paramètre w . Selon la position des poids des connexions, deux cas se présentent :

- Cas des connexions entre la couche cachée et celle de sortie w_{ki} :

Pour le cas des neurones de sortie, l'expression (2.14) devient fonction du paramètre w_{ki} qui influe uniquement sur la sortie du neurone d'indice k . Nous pouvons donc décomposer la dérivée de l'expression (2.14) par :

$$\frac{\partial E^p}{\partial w_{ki}} = \frac{\partial E^p}{\partial y_k^p} \frac{\partial y_k^p}{\partial a_k^p} \frac{\partial a_k^p}{\partial w_{ki}} = (y_k^p - d_k^p) f_k'(a_k^p) v_i^p \quad (2.15)$$

L'expression (2.15) devient alors :

$$\Delta w_{ki} = h \sum_p f_k'(a_k^p) (d_k^p - y_k^p) v_i^p \quad (2.16)$$

- Cas des connexions entre la couche d'entrée et la couche cachée w_{ij} :

Pour le cas des neurones cachés, l'expression (2.14) est fonction du paramètre w_{ij} qui influe non seulement sur la sortie du neurone i de la deuxième couche, mais aussi sur tous les neurones k de la couche de sortie (en aval) qui lui sont connectés. On obtient alors l'équation suivante :

$$\frac{\partial E^p}{\partial w_{ij}} = \frac{\partial E^p}{\partial v_i^p} \frac{\partial v_i^p}{\partial a_i^p} \frac{\partial a_i^p}{\partial w_{ij}} = \frac{\partial E^p}{\partial v_i^p} f_i'(a_i^p) x_j^p \quad (2.17)$$

Le premier terme de cette expression devient alors :

$$\frac{\partial E^p}{\partial v_i^p} = \sum_k \frac{\partial E^p}{\partial y_k^p} \frac{\partial y_k^p}{\partial v_i^p} = \sum_k \frac{\partial E^p}{\partial y_k^p} \frac{\partial y_k^p}{\partial a_k^p} \frac{\partial a_k^p}{\partial v_i^p} = \sum_k (y_k^p - d_k^p) f_k'(a_k^p) w_{ki} \quad (2.18)$$

On obtient alors la modification des poids :

$$\Delta w_{ij} = h \sum_p (f_i'(a_i^p) x_j^p (\sum_k (d_k^p - y_k^p) f_k'(a_k^p) w_{ki})) \quad (2.19)$$

Après avoir calculé la variation des poids des connexions pour tous les neurones de sortie (2.16), on calcule alors la variation des poids des connexions de la couche cachée (2.19), on met ainsi à jour les poids des connexions de la couche de sortie jusqu'à la couche d'entrée, on rétropropage ainsi le signal d'erreur. C'est de là que vient le nom de cet algorithme « rétropropagation du gradient de l'erreur ».

Du fait de sommer les Δw_{ij} pour tous les vecteurs p de la base d'apprentissage puis de remettre à jour les poids avec la variation totale ainsi calculée, l'algorithme est appelé gradient total.

L'algorithme de rétropropagation du gradient de l'erreur a permis de dépasser les limites du Perceptron simple. Il s'avère capable de résoudre un grand nombre de problèmes de classification et de reconnaissance des formes et a donné lieu à beaucoup d'applications. Cet algorithme souffre néanmoins de nombreux défauts, parmi lesquels [3] :

- Une des limitations importantes est le temps de calcul : l'apprentissage est très long .
- Une grande sensibilité aux conditions initiales, c'est-à-dire à la manière dont sont initialisés les poids des connexions ;
- De nombreux problèmes sont dus à la géométrie de la fonction d'erreur : minimums locaux.
- Le problème de dimensionnement du réseau. La rétropropagation apprend une base d'apprentissage sur un réseau dont la structure est fixée a priori. La structure est définie par le nombre de couches cachées, le nombre de neurones par couches et la topologie des connexions. Un mauvais choix de structure peut dégrader considérablement les performances du réseau.

2.6.3. Algorithmes d'apprentissage pour réseaux récurrents

2.6.3.1. Rétropropagation dans le temps «Back-Propagation Through Time(BPTT) »[3]

Soit un réseau de neurones récurrent entièrement connecté où chaque unité est connectée à n'importe quelle autre unité. Supposons que l'évolution du réseau soit menée de façon synchrone en temps discret et donc que chaque neurone du réseau ait pour équation de mise à jour :

$$y_k(t+1) = f(a_k(t)) = f\left(\sum_i (w_{ki} y_i(t)) + x_k(t)\right) \quad (2.20)$$

Dans laquelle $x_k(t)$ est l'entrée, si elle existe, du neurone k à l'instant t .

Supposons en outre que le comportement que l'on souhaite obtenir du réseau est de décrire un ensemble d'états donnés. En d'autres termes, en présence des entrées $x_k(t)$ pour chaque neurone et à chaque instant, on souhaite obtenir une réponse $d_k(t)$ pour certains neurones et à certains instants. Si l'horizon temporel sur lequel le réseau doit réaliser cette tâche est borné par une longueur maximale T , il est possible d'utiliser un artifice pour transformer un réseau récurrent en un réseau à propagation avant (feedforward). C'est cette idée qu'est utilisée avec l'algorithme de la rétropropagation qui a donné lieu à l'algorithme de la rétropropagation dans le temps. Cette transformation consiste à dupliquer les neurones sur

l'horizon temporel $t=1,2,\dots,T$ de façon à ce qu'une unité y_k^t représente l'état $y_k(t)$ du réseau récurrent équivalent.

La figure (2.11) illustre cette transformation pour un réseau récurrent comportant 2 unités et sur un horizon temporel de longueur $T=4$.

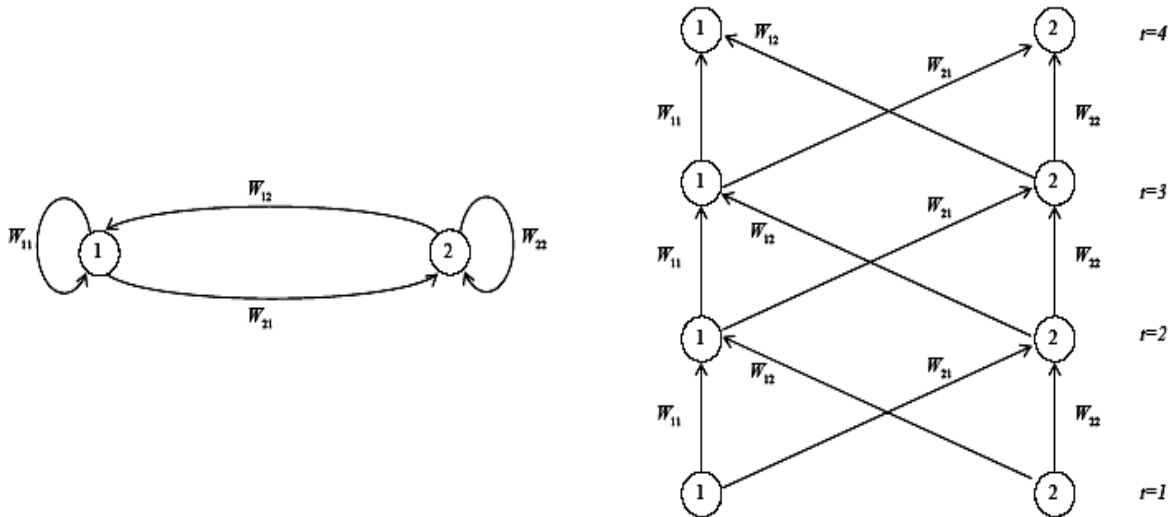


Figure (2.11) : un réseau récurrent et sa version dépliée [3].

Le réseau ainsi déplié est de type feedforward et peut faire l'objet d'un apprentissage par une version légèrement modifiée de l'algorithme de rétropropagation.

Chaque unité calcule ainsi la somme pondérée de ses entrées où, lorsqu'elle existe, l'entrée ou la cible de l'unité k à l'instant t est appliquée à l'unité y_k^t .

Dans le cas des unités de sortie, l'erreur doit être appliquée non seulement sur la dernière couche mais également sur toutes les autres et doit être propagée des couches où elle est apparue vers les couches inférieures.

Le calcul de cette erreur entre les bornes $[t_0, t_1]$ en considérant qu'on N neurones est le suivant:

$$E(t_0, t_1) = \frac{1}{2} \sum_{t=t_0}^{t=t_1} \sum_{k=1}^N E_k(t)^2 \tag{2.21}$$

avec

$$E_k(t) = \begin{cases} d_k(t) - y_k(t) & \text{si } k \text{ est un neurone de sortie} \\ 0 & \text{si ce n'est pas le cas} \end{cases} \tag{2.22}$$

La difficulté de l'algorithme vient du fait que toutes les copies des poids w_{ki} doivent rester identiques alors que l'application de l'algorithme de rétropropagation entraînerait un incrément de Δw_{ki} différent pour chaque copie.

La solution consiste à ajouter tous les incréments et à modifier les valeurs des copies des poids de la quantité obtenue.

$$\Delta w_{pq} = -h \frac{\partial E(t_0, t_1)}{\partial w_{pq}} = h \sum_{t=t_0}^{t_1} \sum_{k=1}^N E_k(t) \frac{\partial y_k(t)}{\partial w_{pq}} \quad (2.23)$$

Le principal problème que pose l'algorithme de rétropropagation du gradient dans le temps est qu'il nécessite beaucoup de ressources informatiques. Pour de longues séquences ou des séquences de longueur inconnue, il devient inutilisable. La duplication ne s'applique que pendant la phase d'apprentissage.

2.6.3.2. Le RTRL (Real Time Recurrent Learning)

L'algorithme de rétropropagation dans le temps représente une technique d'apprentissage hors-ligne : on attend que la totalité des données de l'horizon temporel soient présentées au réseau pour calculer (2.23).

Williams et Zipser [8] ont proposé un algorithme pour l'apprentissage des réseaux récurrents sans avoir à connaître la longueur de l'horizon temporel de la séquence d'apprentissage.

La mise à jour des poids s'effectue au fur et à mesure que les données sont présentées au réseau par une quantité $\Delta w_{pq}(t)$ et, ceci, sans avoir à dupliquer le réseau récurrent.

Leur principal désavantage est que le réseau de neurones doit être entièrement connecté et souffre d'une certaine lenteur du temps d'apprentissage.

$$\Delta w_{pq}(t) = -h \frac{\partial E(t)}{\partial w_{pq}} = h \sum_{k=1}^N E_k(t) \frac{\partial y_k(t)}{\partial w_{pq}} \quad (2.24)$$

avec $E_k(t)$ est l'erreur commise à chaque instant t par les neurones de sortie (équation (2.22)).

D'après l'équation (2.20), on obtient :

$$\frac{\partial y_k(t)}{\partial w_{pq}} = f'(a_k(t-1)) \left[d_{kp} y_q(t-1) + \sum_i w_{ki} \frac{\partial y_i(t-1)}{\partial w_{pq}} \right] \quad (2.25)$$

où d_{kp} est le symbole de Kronecker, notons que cette démarche est analogue à celle utilisée habituellement sur les réseaux à propagation avant, qui consiste à appliquer les modifications

aux poids après chaque exemple au lieu d'attendre la fin du cycle complet de présentation des données.

Par ailleurs, cette technique ne garantit pas le suivi du gradient total de l'erreur de toute une séquence d'apprentissage.

En effet, la trajectoire suivie par le réseau dans l'espace d'état dépend des modifications apportées aux poids à chaque instant. Cet effet peut être éliminé en prenant un taux d'apprentissage assez faible pour diminuer les variations $\Delta w_{pq}(t)$ de l'équation (2.24), [8].

2.7. Mise en oeuvre des réseaux de neurones

Les réseaux de neurones réalisent des fonctions non linéaires paramétrées ; ils peuvent approximer toute fonction de régression si l'on trouve un réseau de neurones qui rend minimum la fonction de coût des moindres carrés (somme des carrés des différences entre les valeurs calculées par le réseau de neurones et les valeurs mesurées).

La mise en oeuvre d'un réseau de neurones nécessite :

- La détermination des entrées pertinentes, c'est-à-dire les grandeurs qui ont une influence significative sur le phénomène que l'on cherche à modéliser.
- La collection des données nécessaires à l'apprentissage et à l'évaluation des performances du réseau de neurones,
- La désignation du nombre de neurones cachés nécessaires pour obtenir une approximation satisfaisante.
- L'estimation des valeurs des paramètres correspondant à un minimum de la fonction de coût, c'est-à-dire d'effectuer un apprentissage.
- L'évaluation des performances du réseau de neurones à l'issue de l'apprentissage.

En fonction des résultats obtenus, il peut être nécessaire d'effectuer plusieurs itérations de tout ou partie de cette procédure [4].

2.8. Domaine d'application

Les réseaux de neurones sont utilisés dans plusieurs domaines parmi eux, on cite :

- Traitement d'image.
- Optimisation.
- Systèmes experts.
- Reconnaissance des formes, des signaux, des événements.
- Radar.

3.1. Introduction

Durant les dernières années, le traitement du signal biomédicale a reçu une grande attention, notamment le signal ECG qui constitue un outil indispensable pour les cardiologues dans le diagnostic. Ce traitement du signal ECG a regroupé plusieurs opérations tel que la détection des complexes QRS, l'analyse automatique et la compression.

Cette dernière s'avère essentielle et primordiale pour résoudre plusieurs problèmes tel que la transmission en temps réel à un centre d'interprétation et le stockage efficace des grandes quantités d'enregistrements pour faire des études ou une consultation ultérieures.

Plusieurs efforts ont été fournis pour élaborer des techniques performantes pour la compression de l'ECG, ces efforts ont donné lieu à plusieurs techniques de compression qui peuvent être classées en trois catégories principales :

- Techniques de compression directes .
- Techniques de compression par transformation.
- Techniques de compression par extraction des paramètres.

La performance d'une technique est mise en évidence par la recherche d'un compromis entre la qualité du signal reconstruit après décompression et un taux de compression le plus élevé que possible en préservant l'information du diagnostic. Pour cela, plusieurs critères d'évaluation de performance ont été élaborés.

Dans ce chapitre, on va présenter ces différents critères d'évaluation ainsi expliquer les différentes méthodes de compression citées précédemment, ce qui s'agit de notre travail de compression de l'ECG est basé sur la troisième méthode en utilisant les réseaux de neurones récurrents, on va présenter l'algorithme proposé en le testant sur la base de données MIT-BIH, enfin nous commenterons les résultats obtenus.

3.2. Algorithmes de compression existants

Le principal but du n'importe quelle technique de compression est d'achever une réduction du volume de données en préservant les caractéristiques morphologiques du signal pour la reconstruction de celui-ci. La compression des données est le processus de détection et élimination des redondances ; Shannon a défini la redondance comme une fraction du message ou de données laquelle est non nécessaire et donc répétitive dans le sens où si elle est oubliée, le message reste essentiellement complet ou au moins il peut être complété.

La redondance dans le signal digital existe chaque fois que les échantillons adjacents du signal sont statistiquement dépendants ou les amplitudes du signal quantifié ne sont pas

présentés avec des probabilités égales[9], les méthodes de compression qu'existent peuvent être classés en trois catégories principales :

3.2.1. Méthodes de compression directes

Ces méthodes détectent les redondances par une analyse directe des échantillons actuels du signal, l'idée de base est d'extraire un sous ensemble des échantillons significatifs à partir d'un ensemble d'échantillons du signal original, selon un critère de sélection, la performance de l'algorithme repose sur la conception d'un critère intelligent de sélection des échantillons à extraire, le signal est reconstruit par un processus inverse, le plus souvent en dessinant des lignes droites entre les échantillons extrait.

Cette catégorie inclut : [10], [11], [12], [13], [14], ... etc.

3.2.2. Méthodes de compression par transformation

Ces méthodes utilisent principalement une analyse des distributions spectrales et énergétiques du signal pour détecter les redondances. Généralement ceci signifie que le traitement du signal d'entrée se fait par le biais de quelque transformée, et coder proprement la sortie transformée. La reconstruction du signal est achevée par un processus de transformation inverse. Généralement ces algorithmes procèdent d'abord à la segmentation du signal à l'aide d'un détecteur du complexe QRS puis à la représentation de chaque segment (en général l'intervalle RR) dans l'espace transformé.

Cette catégorie inclut, [15], la transformation de Walsh [16], la quantification vectorielle [17], les techniques sous bandes avec la transformée en cosinus discret (DCT) [18] ainsi que la compression basée sur la transformée d'ondelettes qui a reçu beaucoup d'attention durant les années récentes, spécialement pour les différentes applications du signal biomédical [19], [20], JPEG2000 [21].

3.2.3. Méthodes de compression par extraction des paramètres

Dans ces méthodes certaines caractéristiques sont extraites pour les utiliser ultérieurement pour reconstruire le signal. Ces méthodes incluent les méthodes de prédiction linéaire et la prédiction à long terme [22], [23] ainsi que les réseaux de neurones [24],[26], qui représentent le but de notre intérêt.

Différents articles ont été récemment publiés consacrés à la comparaison de toutes ces techniques [9], mais il faut à noter que cette tâche est très difficile due à plusieurs raisons parmi elles on cite :

- L'utilisation des bases de données différentes pour l'évaluation de la technique.

- L'utilisation des mesures d'erreurs différentes pour exprimer les dissemblances entre le signal original et le signal reconstruit.

Malgré ça il y'a des mesures d'évaluation des performances très utilisées dans les différentes techniques de compression.

3.3. Evaluation des performances

Les performances des techniques de compression de l'ECG sont évaluées essentiellement en terme de l'erreur de reconstruction introduite dans le signal reconstitué et du taux de compression.

- **PRD (Percent of Root-mean Difference)**

La PRD est utilisée comme critère d'évaluation pour la fidélité du signal reconstruit, elle est donnée par :

$$PRD = \sqrt{\frac{\sum_{n=1}^N (x(n) - \hat{x}(n))^2}{\sum_{n=1}^N x^2(n)}} \times 100 \quad (3.1)$$

Où:

$x(n)$: signal original.

$\hat{x}(n)$: signal reconstruit.

N : la longueur de la fenêtre sur laquelle le PRD est calculée.

La PRD a l'avantage de basse complexité et peu coûteuse pour la calculer, cependant elle a beaucoup de désavantages puisqu'elle dépend du niveau DC du signal original $x(n)$ lequel n'est pas inclus dans la PRD obtenue, il est a noter que si la moyenne du signal est très grande la PRD montre des petites résultats non appropriés.

- **PRD'**

Dans le signal ECG la partie importante n'est pas le niveau du signal mais la variation temporelle de l'ECG, en d'autre termes, la variabilité du signal autour de sa ligne de base c'est la chose qui doit être préservée et non pas la ligne de base elle même, donc la mesure de performance qui doit être utilisée pour révéler l'exactitude de l'algorithme doit être la variance de l'erreur par rapport à la variance du signal [20]:

$$PRD' = \sqrt{\frac{\sum_{n=1}^N (x(n) - \hat{x}(n))^2}{\sum_{n=1}^N (x(n) - \bar{x})^2}} \times 100 \quad (3.2)$$

où :

\bar{x} : est la moyenne du signal.

PRD' sera plus pertinente puisqu'elle est indépendante de la moyenne.

Il est établi que si la valeur du PRD' est entre 0 et 9%, la qualité du signal reconstruit est « très bonne » ou « bonne » [15].

- **RMS (Root Mean Square)**

Cette mesure est donnée par :

$$RMS = \sqrt{\frac{\sum_{n=1}^N (x(n) - \hat{x}(n))^2}{N}} \quad (3.3)$$

- **SNR (signal to noise ratio) :**

$$SNR = 10 \log \left(\frac{\sum_{n=1}^N (x(n) - \bar{x}(n))^2}{\sum_{n=1}^N (x(n) - \hat{x}(n))^2} \right) \quad (3.4)$$

- **Taux de compression de donnée (CDR)**

CDR en bits/seconde est utilisée comme critère d'évaluation pour l'efficacité du codage d'une méthode de compression de l'ECG [26].

$$CDR = \frac{R_{\text{échant}} * B_{\text{total}}}{L} \quad (3.5)$$

où :

$R_{\text{échant}}$: le taux d'échantillonnage.

B_{total} : le nombre total des bits compressés qui doivent être transmise ou stockés.

L : la taille de données.

- **Rapport de compression (CR)**

Cette mesure est donnée par :

$$CR = \frac{\text{nombre de bits utilisées par le signal original}}{\text{nombre de bits utilisées par le signal compressé}} \quad (3.6)$$

Une autre mesure **WDD** a été introduite par [25], cette mesure est basée sur la comparaison des caractéristiques du complexe PQRST des deux signaux ECG, l'original et reconstruit, la **WDD** mesure la préservation relative des informations du diagnostic dans le signal reconstruit.

Ces informations sont : la location, la durée, les amplitudes et formes d'ondes et complexes qui existent dans chaque battement (complexe PQRST), elles doivent d'être choisies avec l'aide d'un cardiologue expérimenté parce que la **WDD** exige l'extraction des caractéristiques.

3.4. Algorithme proposé

Il est bien connu que la prédiction adaptative est très utilisée dans la modélisation et le codage du signal parole, ce signal est non linéaire et qui présente des propriétés non stationnaires.

L'ECG aussi a des caractéristiques non linéaires et non stationnaires, alors l'idée de base était de faire un traitement sur l'ECG similaire à celui de la prédiction adaptative souvent utilisée pour le signal parole.

On a consulté les différentes techniques qui peuvent nous aider dans le filtrage adaptatif, on a trouvé que le réseau de neurones se manifeste comme le candidat le plus favorable pour cette tâche, à cause de ses capacités d'approximation des fonctions non linéaires, cette propriété peut être utilisée dans la prédiction des séries temporelles et la compression des signaux, ainsi que le filtrage adaptatif en utilisant les réseaux de neurones a les propriétés désirées suivantes [29] :

1. On a pas besoin d'approximations telles que, distribution gaussienne, ou de bruit additif.
2. On peut l'appliquer même si le modèle mathématique n'est pas disponible.
3. Le filtre neuronal résultant a la plus petite variance d'erreur pour une structure donnée.
4. Le filtre neuronal converge vers le filtre à variance minimale au fur à mesure que le nombre de neurones cachés augmente.

5. Le filtre neuronal convient pour le traitement en temps réel à cause de sa nature de calcul massivement parallèle.

Mais la méthode traditionnelle d'apprentissage supervisé est inadaptée à effectuer une prédiction non linéaire des signaux non stationnaires à cause de ses exigences d'entraînement hors ligne.

Haykin et Li [27] ont envisagé pour cette tâche la nécessité d'un réseau de neurones qui est capable d'apprendre en ligne c'est à dire apprendre à s'adapter aux statistiques des nouvelles séries temporelles pendant qu'il exécute son rôle de filtrage en même temps. William et Zipser ont proposé un algorithme dit algorithme d'apprentissage des réseaux de neurones récurrents en temps réel (RTRL) [8], cet algorithme est utilisé pour entraîner un réseau récurrent complètement relié, cependant il a une limitation majeure vue sa complexité de calcul $O(N^4)$ où N est le nombre total des neurones dans le réseau.

Haykin et Li [27], en se basant sur le principe de « diviser et conquérir » ont contourné cette complexité de calcul en proposant une architecture modulaire et récurrente dite réseau de neurones récurrent canalisé (PRNN : Pipelined Recurrent Neural Network) basée sur l'algorithme RTRL.

C'est cette architecture, qu'on va adopter pour notre travail. Tout d'abord on va la présenter avec son algorithme d'apprentissage en prenant en compte les modifications apportées par Chambers et Baltersee [28] qui ont permis de diminuer d'avantage la complexité des calculs.

3.5. Principe de travail :

Le prédicteur proposé par [27] est composé de deux sous sections , la première dite sous section non linéaire(PRNN) et la deuxième sous section linéaire représentée par un filtre conventionnel à ligne de retard branchée (TDL : Tapped-delay-line) .

Le prédicteur est basé premièrement sur la notion de linéarisation du signal d'entrée avec l'aide du PRNN puis faire avancer ces données traités dans le TDL, qui constitue un prédicteur linéaire pour nous donner une prédiction à un pas en avant du signal original.

Cette combinaison des deux traitements non linéaire et linéaire doit extraire les relations linéaire et non linéaire du signal original.

3.5.1. Sous section non linéaire (PRNN)

La figure (3.1) montre une structure détaillée du PRNN, impliquant M niveaux de traitement. Chaque niveau a un module neuronal et son propre comparateur, spécifiquement le module M consiste d'un réseau de neurones récurrent complètement reliés avec N neurones,

alors que pour les modules $\{M-1, \dots, 1\}$, un des signaux feedback est substitué par la sortie du premier neurone du module suivant.

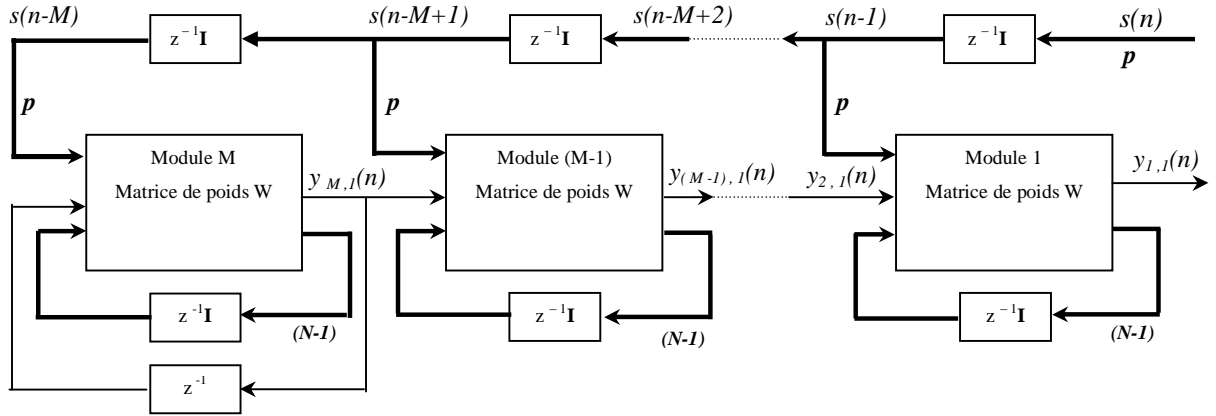


Figure (3.1) : section non linéaire : PRNN.

Chaque module i , en addition des N signaux feedback il y'a p entrées externes :

$$\mathbf{s}_i(n) = [s(n-i), s(n-(i+1)), \dots, s(n-(i+p-1))] \quad (3.7)$$

Avec p est l'ordre du prédicteur non linéaire, aussi chaque neurone a une entrée toujours fixe de +1 celle ci représente le biais.

Tous les modules opèrent de façon similaire, dans cela ils ont le même nombre des entrées externes et des signaux feedback. De plus ils ont la même matrice de poids $W: W(p+N+1) \times N$, laquelle est variante en temps c'est-à-dire $W=W(n)$, les entrées sont linéairement combinées et alors propager à travers une fonction d'activation non linéaire $f(\cdot)$ pour former la sortie de chaque neurone. La sortie totale du PRNN est la sortie du premier neurone du premier module c'est à dire $y_{1,1}(n)$ comme elle est montrée dans la figure (3.1).

Les équations suivantes donnent une description complète de la dynamique du PRNN :

Pour $1 \leq i \leq M-1$:

$$\mathbf{x}_i^T(n) = [s(n-i), \dots, s(n-i-p+1), 1, y_{i+1,1}(n), y_{i,2}(n-1), \dots, y_{i,N}(n-1)] \quad (3.8)$$

Pour $i=M$:

$$\mathbf{x}_i^T(n) = [s(n-i), \dots, s(n-i-p+1), 1, y_{i,1}(n-1), y_{i,2}(n-1), \dots, y_{i,N}(n-1)] \quad (3.9)$$

$$v_{i,k}(n) = \sum W_{k,l}(n) x_{i,l}(n) \quad (3.10)$$

$$y_{i,k}(n) = f(v_{i,k}(n)) \quad (3.11)$$

3.5.1.1. Linéarisation du signal d'entrée

La première opération du prédicteur non linéaire est la linéarisation des entrées externes $\{s(n)\}$ du signal d'entrée $x_i(n)$ en utilisant le PRNN, la linéarisation elle-même consiste en trois étapes :

1. **Prédiction** : après avoir donné à chaque module $i (i=1, \dots, M)$ les vecteurs d'entrée $x_i(n)$ la sortie de tous les neurones du réseau au temps n sont calculés. La sortie du module i est définie comme la sortie du premier neurone $y_{i,1}$, comme la configuration des modules dont laquelle ils sont connectés dans le PRNN est particulière la sortie totale du PRNN devient la sortie du module 1 c'est-à-dire $y_{1,1}$, ensuite le PRNN calcule l'erreur de prédiction à un pas $e_i(n)$ de chaque module i au temps n , laquelle est définie comme la différence entre la réponse désirée $s(n-i+1)$ du module i au temps n , et la sortie du module i au temps n :

$$e_i(n) = s(n-i+1) - y_{i,1}(n) \quad (3.12)$$

Après le calcul de ses erreurs, le PRNN calcule la fonction d'erreur $E(n)$ à travers tous les modules qu'est la somme des M erreurs quadratiques instantanés pondérés par un facteur d'oubli I , lequel détermine la pondération de chaque module.

$$E(n) = \sum_{i=1}^M I^{i-1} e_i^2(n) \quad (3.13)$$

2. **Mise à jour des poids** : l'algorithme d'apprentissage utilise le gradient de la fonction du coût $E(n)$ pour calculer la matrice de correction des poids $\Delta W(n)$, laquelle sera ajoutée à la matrice des poids $W(n)$ dans le but de construire la nouvelle matrice des poids $W(n+1)$.
3. **Filtrage** : le signal de sortie du PRNN à l'instant suivant $(n+1)$ est calculé, ceci est fait en recalculant tous les sorties des modules $y_{i,k}(n+1) (i=1, \dots, M, k=1, \dots, N)$ en utilisant la matrice des poids met à jour $W(n+1)$ et les vecteurs d'entrée met à jour $x_i^T(n+1)$, ces derniers sont formés en substituant le vecteur d'entrée externe $s_i(n) = [s(n-i), s(n-(i+1)), \dots, s(n-(i+p-1))]$ par $s_i(n+1)$ dans les vecteurs d'entrée $x_i(n)$ pour chaque module $i=1, \dots, M$.

Dans cette étape nous avons utilisé les modifications apportées par [20] qui consistent de la mise à jour des valeurs feedback dans le signal $x_i(n)$, les valeurs feedback à l'instant suivant ($n+1$) seront les sorties filtrés à l'instant n .

Dans le but d'implémenter le PRNN, il est nécessaire d'établir un algorithme d'apprentissage basé sur le gradient de la fonction du coût du PRNN pour calculer la correction $\Delta W(n)$ afin de met à jour la matrice de poids $W(n)$.

3.5.1.2. L'algorithme d'apprentissage du PRNN

L'idée de base de l'algorithme est de calculer la correction $\Delta W(n)$ dans la direction négative du gradient de la fonction de coût $E(n)$, donc le changement pour le l^{eme} poids du neurone k à l'instant n c'est à dire $\Delta w_{k,l}(n)$ peut être trouver par :

$$\begin{aligned}\Delta w_{k,l}(n) &= -h \frac{\partial}{\partial w_{k,l}(n)} \left(\sum_{i=1}^M I^{i-1} e_i^2(n) \right) \\ &= -2h \sum_{i=1}^M I^{i-1} e_i(n) \frac{\partial e_i(n)}{\partial w_{k,l}(n)}\end{aligned}\quad (3.14)$$

h : Le pas d'apprentissage.

Comme le signal d'entrée s ne dépend pas des éléments de W , la dérivée partielle de l'erreur instantanée du i^{eme} module dans le PRNN en fonction du poids $w_{k,l}$ devient :

$$\frac{\partial e_i(n)}{\partial w_{k,l}(n)} = - \frac{\partial y_{i,1}(n)}{\partial w_{k,l}(n)} \quad (3.15)$$

$$\frac{\partial y_{i,1}(n)}{\partial w_{k,l}(n)} = \frac{\partial y_{i,1}(n)}{\partial v_{i,1}(n)} \frac{\partial v_{i,1}(n)}{\partial w_{k,l}(n)} = f'(v_{i,1}(n)) \frac{\partial v_{i,1}(n)}{\partial w_{k,l}(n)} \quad (3.16)$$

Où f' est la dérivée première de la fonction d'activation f en fonction de son argument $v_{i,1}$, lequel est l'activité interne dans l' i^{eme} module, en substituant (3.8)-(3.11), dans (3.16) on trouve :

$$f'(v_{i,1}(n)) \frac{\partial v_{i,1}(n)}{\partial w_{k,l}(n)} = f'(v_{i,1}(n)) \left(\sum_{a=1}^{p+N+1} \left(\frac{\partial w_{1,a}(n)}{\partial w_{k,l}(n)} x_{i,l}(n) + \frac{\partial x_{i,a}(n)}{\partial w_{k,l}(n)} w_{1,a}(n) \right) \right) \quad (3.17)$$

Le premier terme dans la somme précédente est zéro à l'exception pour $k=1$ et $l=a$, et les seuls éléments du vecteur d'entrée x qui dépend des éléments de W sont les valeurs feedback

\mathbf{r} dans le vecteur d'entrée du module, si ils sont mise à jour avant la partie de filtrage, ils sont donnés par :

$$\mathbf{r}_i^T = [y_{i+1,1}(n), y_{i,2}(n-1), \dots, y_{i,N}(n-1)] ; \quad 1 \leq i \leq M-1 \quad (3.18)$$

$$\mathbf{r}_i^T = [y_{i,1}(n-1), y_{i,2}(n-1), \dots, y_{i,N}(n-1)] ; \quad i=M \quad (3.19)$$

Maintenant (3.17) devient :

$$\mathcal{F}(v_{i,1}(n)) \frac{\partial v_{i,1}(n)}{\partial w_{k,l}(n)} = \mathcal{F}(v_{i,1}(n)) \left(\sum_{a=1}^N \frac{\partial r_{i,a}(n)}{\partial w_{k,l}(n)} w_{1,a+p+1}(n) + \mathbf{d}_{kl} x_{i,l}(n) \right) \quad (3.20)$$

$$\mathbf{d}_{kl} = \begin{cases} 1, & k=l \\ 0, & k \neq l \end{cases} \quad (3.21)$$

Si les valeurs feedback dans $\mathbf{x}_i(n)$ ne sont pas mise à jour, l'index du temps n dans l'équation (3.20) doit être remplacé par $(n-1)$ et $(n-1)$ par $(n-2)$.

Ayant défini le vecteur \mathbf{r}_i^T , il est possible de formuler une relation récursive pour les dérivés partiales des éléments du vecteur feedback $\mathbf{r}_{i,j}$ en fonction des poids $w_{k,l}$.

Pour le cas lorsque $(1 \leq i \leq M \text{ et } 2 \leq j \leq N)$, (3.16) devient :

$$\begin{aligned} \frac{\partial r_{i,j}(n)}{\partial w_{k,l}(n)} &= \frac{\partial y_{i,j}(n-1)}{\partial w_{k,l}(n)} \\ &= \mathcal{F}(v_{i,j}(n-1)) \left(\sum_{a=1}^N \frac{\partial r_{i,a}(n-1)}{\partial w_{k,l}(n)} w_{j,a+p+1}(n-1) + \mathbf{d}_{kj} x_{i,l}(n-1) \right) \end{aligned} \quad (3.22)$$

Les dérivées partiales de la partie droite de l'équation précédente se posent puisque les modules des PRNN ont des feedback, par lequel les valeurs précédentes des échantillons de sortie dépendent des valeurs précédentes des poids, lesquelles sont reliées aux poids courantes via l'algorithme de mise à jour des poids.

Il est a noté que ces dérivées partiales sont fait en fonction des éléments courants de W , donc l'expression (3.22) n'est pas longtemps récursive.

Cependant, si le pas d'apprentissage h est choisit suffisamment petit, donc les poids sont supposés de s'adapter lentement, alors sous cette approximation $w_{k,l}(n+1) \approx w_{k,l}(n)$, (3.22)

peut être approximer par :

$$\frac{\partial r_{i,j}(n)}{\partial w_{k,l}(n)} \approx \mathcal{F}(v_{i,j}(n-1)) \left(\sum_{a=1}^N \frac{\partial r_{i,a}(n-1)}{\partial w_{k,l}(n-1)} w_{j,a+p+1}(n-1) + \mathbf{d}_{k,j} x_{i,l}(n-1) \right) \quad (3.23)$$

Pour le cas lorsque ($1 \leq i \leq M$ et $j=1$), aucune approximation nécessite d'être faite et la relation précédente devient :

$$\begin{aligned} \frac{\partial r_{i,l}(n)}{\partial w_{k,l}(n)} &= \frac{\partial y_{i+1,l}(n)}{\partial w_{k,l}(n)} \\ &= f'(v_{i+1,l}(n)) \left(\sum_{a=1}^N \frac{\partial r_{i+1,a}(n)}{\partial w_{k,l}(n)} w_{1,a+p+1}(n) + d_{kj} x_{i+1,l}(n) \right) \end{aligned} \quad (3.24)$$

Utilisant (3.20), (3.23) et (3.24) les gradients des sorties des neurones peuvent être exprimés à travers les relations récursives suivantes :

$$i = M \Rightarrow \frac{\partial y_{i,j}(n)}{\partial w_{k,l}(n)} \approx f'(v_{i,j}(n)) \left[\sum_{a=1}^N \frac{\partial y_{i,a}(n-1)}{\partial w_{k,l}(n-1)} w_{j,a+p+1}(n) + d_{kj} x_{i,l}(n) \right] \quad (3.25)$$

$$i \neq M \Rightarrow \frac{\partial y_{i,j}(n)}{\partial w_{k,l}(n)} \approx f'(v_{i,j}(n)) \left[\frac{\partial y_{i+1,j}(n)}{\partial w_{k,l}(n)} w_{j,p+2}(n) + \sum_{a=2}^N \frac{\partial y_{i,a}(n-1)}{\partial w_{k,l}(n-1)} w_{j,a+p+1}(n) + d_{kj} x_{i,l}(n) \right] \quad (3.26)$$

En outre, maintenant la procédure de linéarisation consiste seulement en deux étapes au lieu de trois :

- **Prédiction** : calculer les erreurs de prédiction à un pas (3.12) à partir du PRNN, et calculer la fonction du coût (3.13).
- **Mise à jour des poids** : utiliser l'algorithme de mise à jour sur la fonction du coût (3.13) pour calculer $\Delta W(n)$, lequel est ajouté à $W(n)$ pour former la matrice des poids met à jour $W(n+1)$.

Il n'y a aucun besoin de la procédure de filtrage explicite parce que la partie de prédiction de la procédure de linéarisation précédente à l'instant $(n+1)$ maintenant contient la partie filtrée de la procédure de linéarisation à l'instant n .

3.5.2. Sous section linéaire

Comme il était mentionner auparavant, la sous section linéaire consiste en un filtre conventionnel à ligne de retard branchée (TDL), lequel est montré sur la figure (3.2), la sortie du sous section non linéaire $y_{filt}(n)$ donnée par :

$$y_{filt}(n) = \{y_{filt}(n), y_{filt}(n-1), \dots, y_{filt}(n-q+1)\} \quad (3.27)$$

est entré dans le TDL pour accomplir le processus de prédiction à un pas en avant.

Le TDL contient q branchements, les entrées du branchements sont pondérés par le vecteur de poids donné par :

$$\mathbf{u} = [u_1, u_2, \dots, u_q]^T \quad (3.28)$$

$$\hat{s}(n+1) = \mathbf{u}^T \mathbf{y}_{\text{filt}}(n) \quad (3.29)$$

$\hat{s}(n+1)$ Constituer une prédiction à un pas, cette dernière est comparée au signal original $s(n+1)$ en calculant le signal erreur $e(n+1)$ donnée par :

$$e(n+1) = s(n+1) - \hat{s}(n+1) \quad (3.30)$$

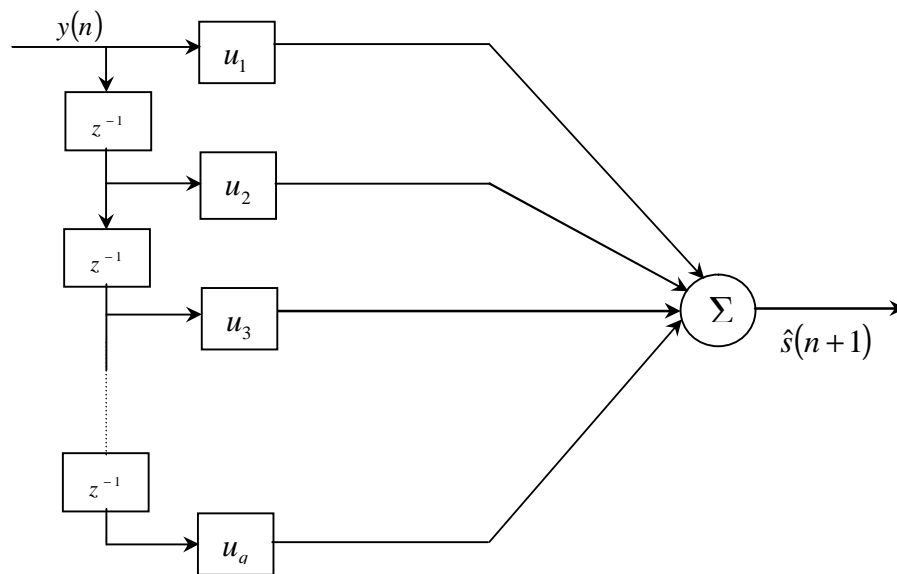


Figure (3.2) : section linéaire : TDL

La valeur de l'erreur est utilisée pour la mise à jour du vecteur de poids \mathbf{u} en utilisant l'algorithme des moindres carrés **LMS** (Least Mean Square) ce qui donne :

$$\Delta \mathbf{u} = \mathbf{m} \mathbf{y}_{\text{filt}}(n) e(n+1) \quad (3.31)$$

Avant de procéder au calcul de la prédiction, il faut faire une initialisation de la matrice des poids W du PRNN et du vecteur de poids du branchement w_i et ainsi que les valeurs feedback \mathbf{r} .

3.5.3. Initialisation de l'algorithme d'apprentissage

L'initialisation du vecteur de poids du branchement w_i suit la pratique habituelle de le mettre égale au vecteur nul ou de vecteur de petite valeurs distribuées aléatoirement, cependant l'initialisation du matrice des poids W et du vecteur feedback \mathbf{r} exige une intention spéciale, pour cette initialisation on utilise la méthode d'entraînement traditionnelle par

époque des réseaux de neurones récurrents, laquelle est appliquée simplement sur un seul module opérant avec N_0 échantillons du signal d'entrée, on peut la résumer comme suit :

1. entrer N_0 échantillons du signal $s(n)$, et obtenir un vecteur $\mathbf{s}(n) : (p \times 1)$, et un signal désiré $d(n)$, lorsque $1 \leq i \leq m$ et $m = N_0 - p$:

$$\mathbf{s}(n) = [s(n+(p-1)), \dots, s(n)]^T \quad (3.32)$$

$$d(n) = s(n+p) \quad (3.33)$$

2. la procédure de pré-entraînement commence avec $n=1$. Choisir un ensemble de petites valeurs aléatoires pour la matrice $W : W(p+N+1) \times N$, et un vecteur $(N \times 1)$ pour le vecteur feedback \mathbf{r} et obtenir $\mathbf{x}(n)$.

$$\mathbf{x}(n)^T = [s(n), \dots, s(n-p+1), 1, r(n,1), r(n,2), \dots, r(n,N)] \quad (3.34)$$

3. entrer $\mathbf{x}(n)$, $\mathbf{r}(n)$, et $d(n)$ à un module, et exécuter les calculs suivants :

$$v_k(n) = \sum_{l=1}^{p+N+1} W_{k,l}(n) x(n,l) \quad (3.35)$$

$$y_k(n) = f(v_k(n)) \quad (3.36)$$

$$e(i) = d(n) - y_1(n) \quad (3.37)$$

4. soit $n=n+1$, et mettre :

$$r(n) = y(n-1) \quad (3.38)$$

et retourner à l'étape (3).

5. répéter les calculs des étapes (3) et (4) jusqu'à $n=m$ et calculer la fonction du coût $E(m)$ définit par :

$$E(m) = \frac{1}{m} \sum_{n=1}^m e^2(n) \quad (3.39)$$

si la fonction du coût $E(m)$ est inférieure à une valeur permet ϵ , la procédure de pré-entraînement est terminée, autrement on exécute l'étape (6).

Approximativement la valeur permet est à peu près 1% de la moyenne de carrée du signal d'entrée externe $\mathbf{s}(n)$.

6. calculer la correction $\Delta W(n)$ en utilisant l'algorithme d'apprentissage montré auparavant.
7. mettre $n=1$ et $r(1) = y(m)$ et retourner à l'étape (3).

3.6. Conception du réseau

La conception du réseau a été soumise à un compromis entre la fonction du coût minimale avec le plus petit nombre d'itérations possible et cela a été obtenu expérimentalement.

a).Section non linéaire :

Le nombre de modules : $M = 4$.

Le nombre de neurones par chaque module : $N = 2$.

L'ordre du prédicteur : $p = 4$.

Le pas d'apprentissage : $\eta = 0.0001$.

b).Section linéaire :

Nombre de branchements du TDL : $q = 7$.

Le pas d'apprentissage : $\mu = 0.001$.

La longueur de la base d'apprentissage est de $T = 1024$ échantillons parmi lesquels on a choisit $N_0 = 200$ pour faire l'initialisation qui est essentielle pour assurer la convergence de l'algorithme.

3.7. Résultats

Le signal utilisé pour l'apprentissage est l'enregistrement 105 de la base de données MIT-BIH, ce dernier est échantillonné à une fréquence de 360 Hz et chaque échantillon est codé sur 11bits à l'issue de l'apprentissage on conserve que les $(M+p+1)$ premiers échantillons de la base d'apprentissage, la matrice de poids W section non linéaire, le vecteur de poids w section linéaire ainsi que le signal de l'erreur de quantification du signal d'erreur de prédiction de la section linéaire, c'est par exemple on quantifie le signal sur 256 niveaux ce qui implique un codage sur 8 bits cela résulte à un rapport de compression $CR = 1.375$.

Ce dernier peut être amélioré par un codage de longueur variable comme le fameux codage de Huffman.

Les résultats de reconstitution de quelques enregistrements sont schématisés sur les figures (3.3) à (3.8), elles représentent les enregistrements originaux et leurs enregistrements reconstruits correspondants. Ces figures montrent que les différentes composantes du signal ECG sont bien reproduites après décompression avec une grande fidélité qui peut être

comparable aux méthodes existantes actuellement comme le montre les deux tableaux suivants qui rapportent les résultats numériques du PRD et PRD1 de l'algorithme proposé et de d'autres algorithmes.

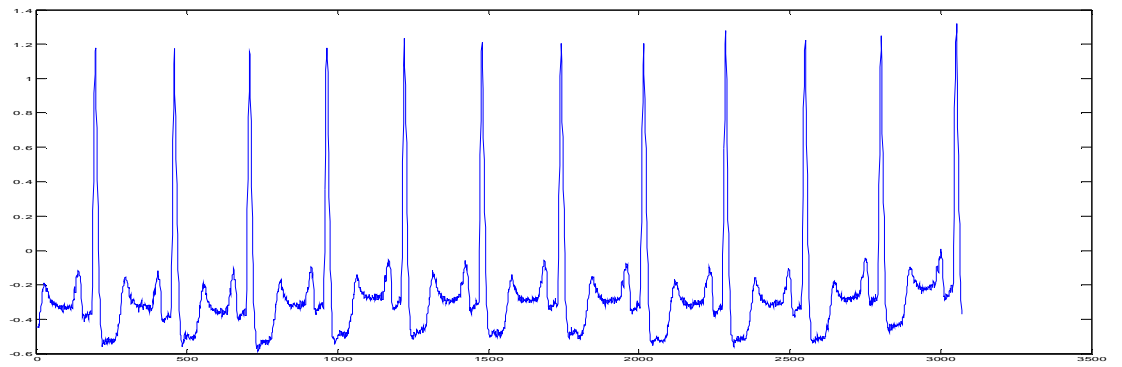
Enregistrements	Algorithme proposé	Référence [24]	Référence [15]
Enregistrement 103	3.54	4.10	-
Enregistrement 105	0.74	-	-
Enregistrement 108	1.62	-	-
Enregistrement 117	0.66	-	0.22
Enregistrement 119	1.86	1.72	1.17
Enregistrement 200	4.05	4.58	-

Tableau (3.1) : valeurs de PRD.

Enregistrements	Algorithme proposé	Référence [24]	Référence [15]
Enregistrement 103	4.44	-	-
Enregistrement 105	0.95	-	-
Enregistrement 108	3.48	-	-
Enregistrement 117	2.55	-	3.58
Enregistrement 119	1.72	-	9.54
Enregistrement 200	4.27	-	-

Tableau (3.2) : valeurs de PRD1.

a)



b)

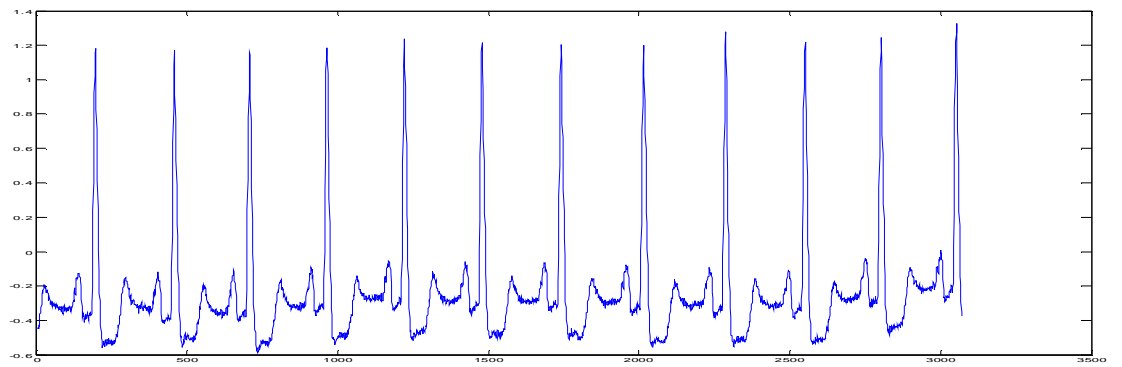
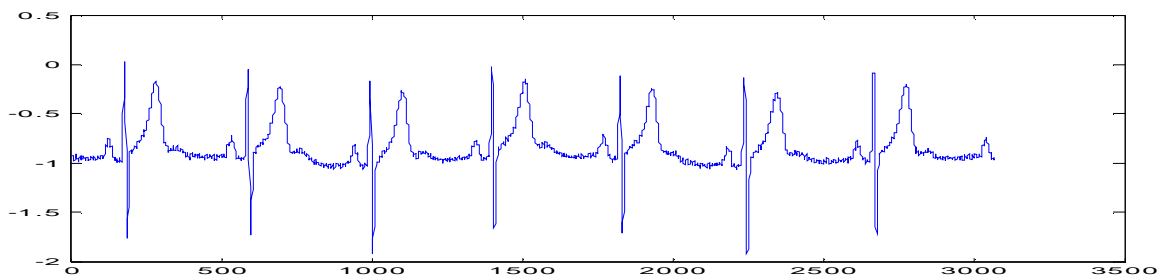


Figure (3.3) : a) signal original « enregistrement 105 ».b) signal reconstitué.

a)



b)

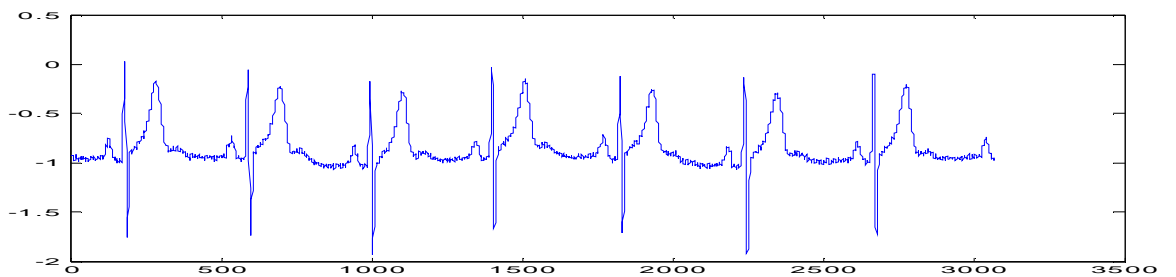
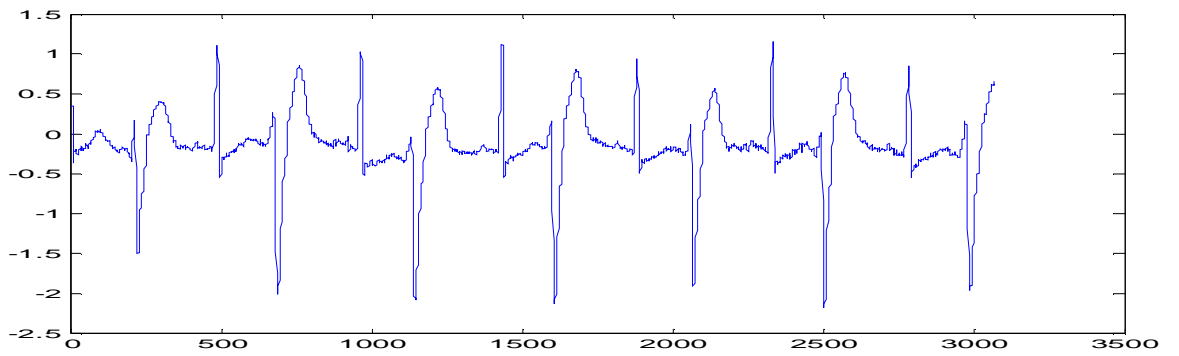


Figure (3.4) : a) signal original « enregistrement 117 ».b) signal reconstitué.

a)



b)

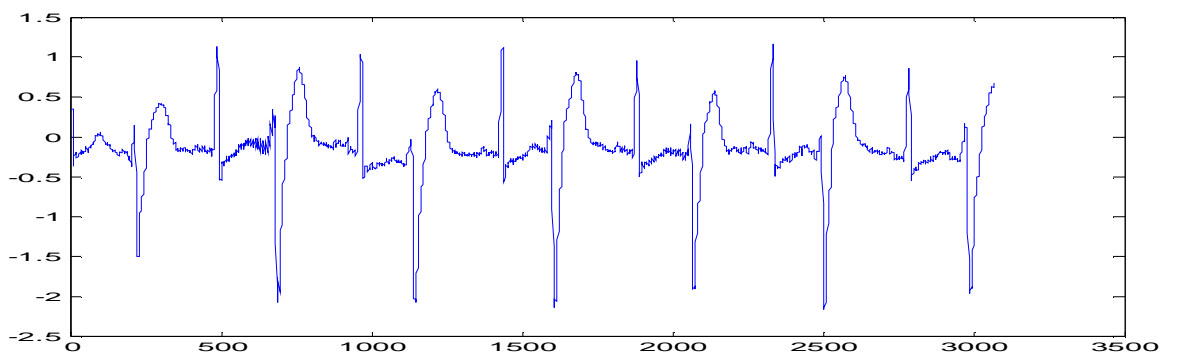
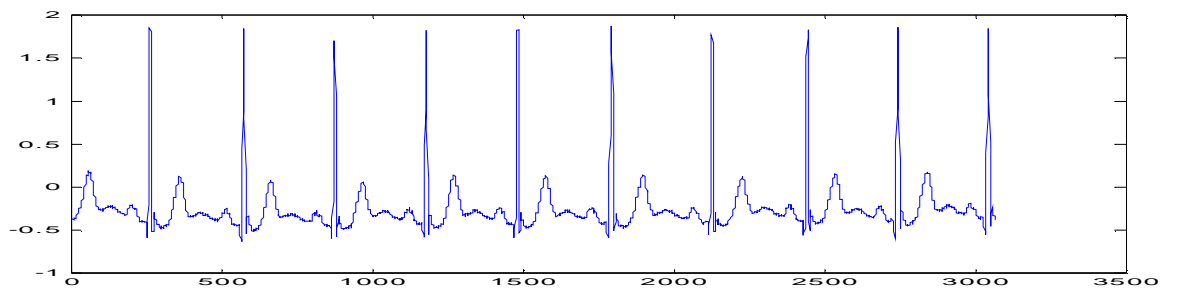


Figure (3.5) : a) signal original « enregistrement 200 ».b) signal reconstitué.

a)



b)

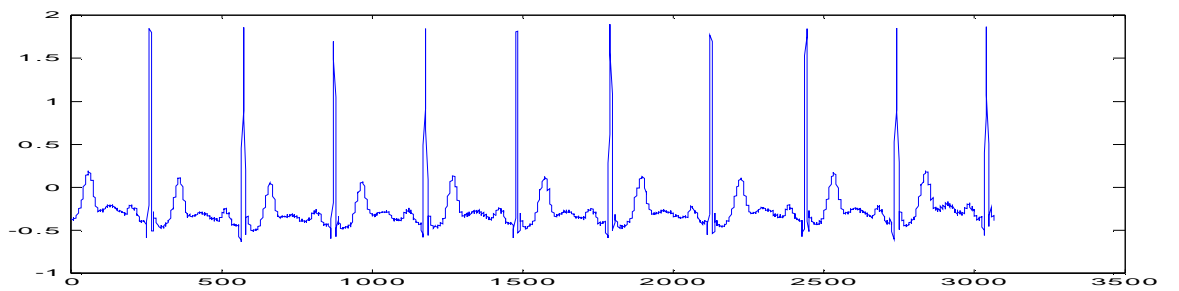
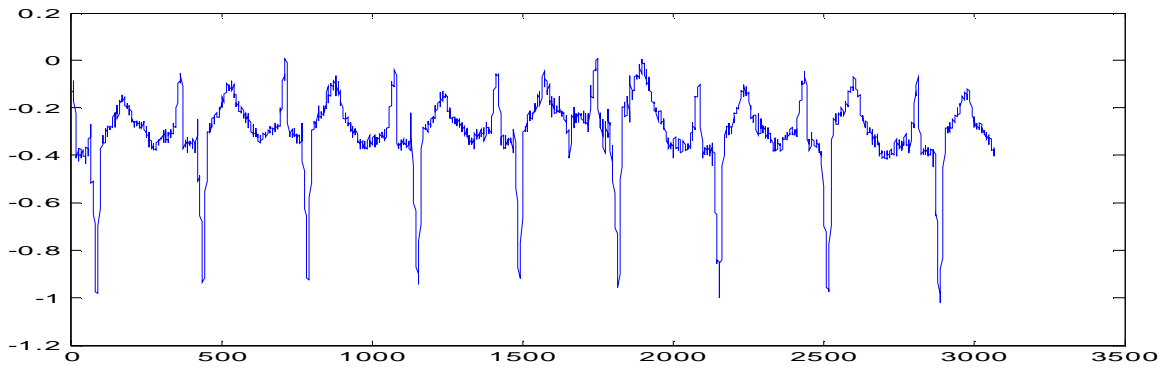


Figure (3.6) : a) signal original « enregistrement 103 ».b) signal reconstitué.

a)



b)

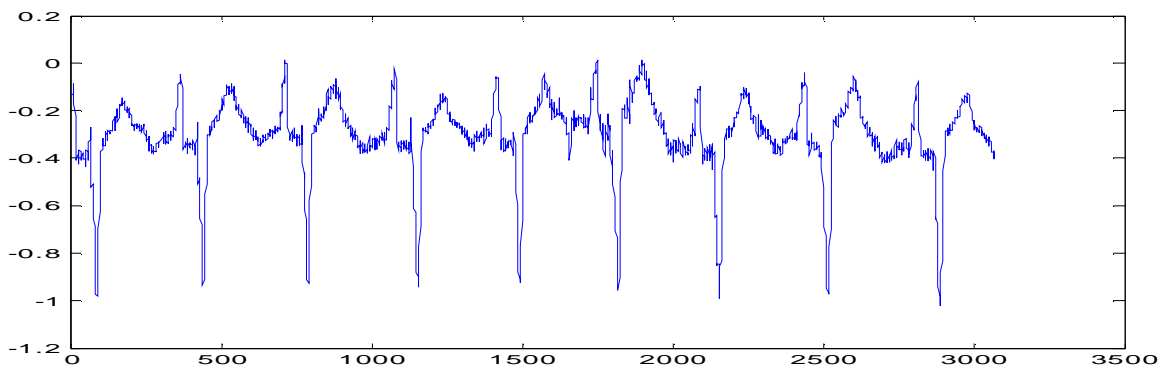
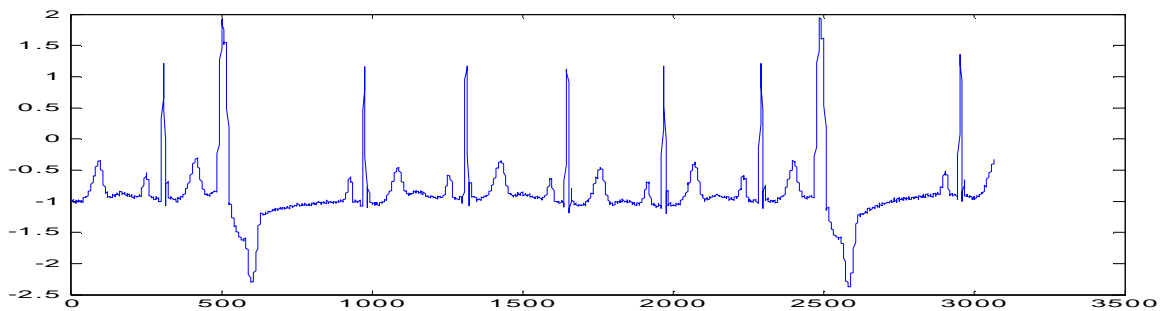


Figure (3.7) : a) signal original « enregistrement 108 ».b) signal reconstitué.

a)



b)

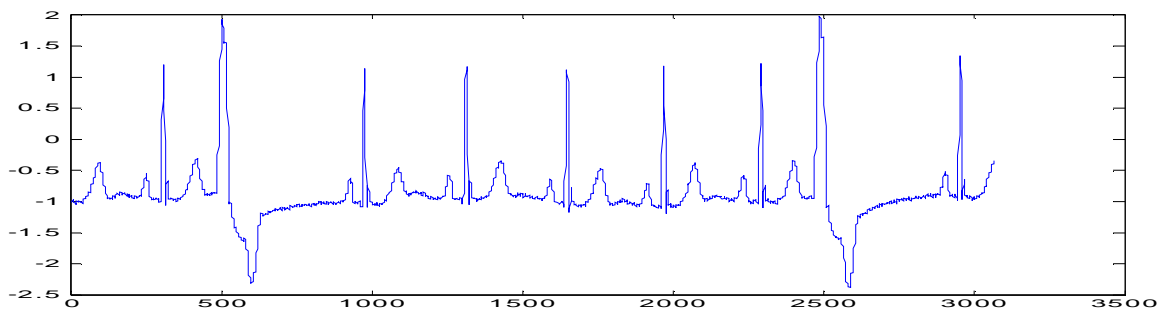


Figure (3.8) : a) signal original « enregistrement 119 ».b) signal reconstitué.

3.8. Conclusion

Dans ce chapitre nous avons proposé un algorithme de compression de l'ECG basé sur les réseaux de neurones récurrents constitué de deux sections, cet algorithme consiste à stocker la matrice de poids du section non linéaire et le vecteur de poids du section linéaire ainsi que le signal d'erreur de quantification du signal d'erreur de prédiction, qui aura une dynamique plus petite que le signal original.

Nous avons testé l'algorithme sur l'enregistrement 105 du MIT-BIH et avec la même matrice et le même vecteur de poids nous l'avons testé sur d'autres enregistrements de la même base de données sans faire l'apprentissage de ces derniers, malgré ça on a aboutit à des résultats encourageants.

Pour avoir un bon rapport de compression il faut faire un codage efficace du signal d'erreur , cependant un codage à longueur variable comme le fameux codage de Huffman améliore d'avantage le rapport de compression.

Conclusion générale

Dans ce mémoire, nous avons pris en charge le problème de la compression de l'ECG par une approche basée sur les réseaux de neurones artificiels.

Dans le premier chapitre, nous avons exposé l'essentiel de la morphologie du cœur pour comprendre un peu d'où vient le signal ECG ainsi que la morphologie du signal lui-même c'est à dire ses différents segments et ondes constituant l'ECG qui nous sont utiles pour évaluer la performance de l'algorithme de compression dans le troisième chapitre, ensuite nous avons exposé les artefacts qui peuvent affecter l'ECG. Enfin, on a donné quelques notions sur le Holter numérique qui constitue un appareil ambulateur d'enregistrement de l'ECG.

Dans le deuxième chapitre, on a illustré les principes des réseaux de neurones, leurs propriétés et les différentes architectures existantes tout en insistant sur les réseaux de neurones récurrents qui ont constitué notre outil de travail pour la compression, ensuite on a exposé les différents algorithmes d'apprentissage des réseaux de neurones, la règle delta et la rétropropagation pour les réseaux multicouches ainsi que la rétropropagation dans le temps et le RTRL pour les réseaux récurrents, ce dernier a été l'algorithme choisi pour l'apprentissage de notre algorithme de compression.

Enfin, le dernier chapitre a été consacré à l'exposition de l'algorithme proposé cependant on a commencé tout d'abord par montrer la nécessité d'un algorithme de compression dans le stockage ou la transmission du signal ECG, ainsi on a donné les différentes catégories d'algorithmes existants actuellement et les critères d'évaluation de la performance. Nous avons opté pour la catégorie d'extraction de paramètres pour sa simplicité d'acquisition des paramètres identifiant le réseau de neurones conçu.

Après l'exécution de l'algorithme sur l'enregistrement 105 du MIT-BIH, on a extrait les données nécessaires telle que la matrice des poids de la section non linéaire et le vecteur des poids de la section linéaire ainsi que le signal d'erreur, cela a permis de reconstruire fidèlement l'enregistrement ainsi que d'autres enregistrements qui n'ont pas été sujets d'apprentissage. Les valeurs de PRD et PRD1 ont montré que les résultats obtenus sont encourageants et en ce qui concerne le CR ça nécessite un codage efficace pour l'améliorer d'avantage.

Résumé

Aujourd'hui, en s'alliant avec la technologie la probabilité de guérison du malade s'est amplifiée, avec l'utilisation d'un matériel très sophistiqué on arrive à avoir des diagnostics plus fiable et une bonne compréhension des maladies, citant l'exemple d'un organe noble qui est le cœur, ce dernier qui est en perpétuel mouvement jouant un rôle très important dans la vie de l'être humain ceci lui cause de temps en temps des faiblesses qui peuvent se convertir en maladies. Ce qui a motivé des scientifiques à élaborer des techniques pour diagnostiquer ces anomalies du cœur, cela a abouti à la naissance du tracé électrocardiographique (ECG) il constitue un outil efficace pour le diagnostic des anomalies du cœur. Suite à cette importance qu'il a occupé il attire une grande attention, plusieurs efforts ont été fournis dans le traitement du signal ECG ce qui a donné lieu à des algorithmes d'analyse automatique en améliorant la qualité des enregistrements avec l'emploi des techniques de traitement du signal tel que la suppression du bruit et la détection des caractéristiques de l'ECG, c'est pour cela on a besoin d'un enregistrement de taille satisfaisante qui reflète bien l'état du cœur, néanmoins ceci résulte à des problèmes de stockage et encore avec le développement actuel des nouvelles techniques d'information et de communication on pense à le transmettre en temps réel via les réseaux de communications à des centres d'interprétation distant ce qui a motivé la recherche envers la compression, plusieurs méthodes de compression ont été proposées, ces méthodes peuvent être réparties en trois catégories principales :

- Méthodes de compression directes.
- Méthodes de compression par transformation.
- Méthodes de compression par extraction de paramètres.

Dans ce mémoire qui comprend trois chapitres un algorithme de compression via l'extraction des paramètres en utilisant les réseaux de neurones récurrents a été proposé due à sa simplicité d'acquisition des paramètres identifiant le réseau de neurones conçu.

Le principal but n'importe quelle technique de compression et d'achever une réduction du volume de données en préservant les caractéristiques morphologiques du signal pour la reconstruction de celui-ci. Pour cette raison le premier chapitre a été consacré à donner l'essentiel de la morphologie du cœur qui est la source du signal ECG ainsi que celle de l'ECG qui est utile pour l'évaluation de la performance des algorithmes de compression.

Récemment les réseaux de neurones ont pris une grande place dans le traitement du signal surtout les réseaux de neurones récurrents qui sont capables de s'adapter aux variations temporelles des signaux non linéaires et non stationnaires tel que le signal ECG.

Le deuxième chapitre, a été consacré à donner une illustration générale des réseaux de neurones, leurs propriétés et les différentes architectures existantes tout en insistant sur les réseaux de neurones récurrents qui ont constitués l'outil de travail pour la compression, ensuite les différents algorithmes d'apprentissage des réseaux de neurones ont été exposé la règle delta et la rétropropagation pour les réseaux de neurones multicouches ainsi que la rétropropagation dans le temps et l'RTRL(Real Time Recurrent Learning) pour les réseaux de neurones récurrents.

Enfin, le dernier chapitre a été consacré à l'exposition de l'algorithme proposé en commençant par une explication des différentes méthodes existantes et les différents critères d'évaluation de performance, le réseau de neurones utilisé est constitué de deux sections la première est non linéaire et l'autre est linéaire, ce réseau possède une architecture modulaire afin de contourner la complexité de calculs de l'algorithme d'apprentissage RTRL qu'est utilisé pour l'apprentissage, ensuite l'algorithme proposé a été exécuté sur l'enregistrement 105 du MIT-BIH, les données nécessaires tel que la matrice de poids du section non linéaire et le vecteur de poids du section linéaire de l'architecture du réseau de neurones utilisée ainsi que le signal d'erreur ont été extrait, cela a permis de reconstruire fidèlement l'enregistrement ainsi que d'autres enregistrements qui n'ont pas été sujet d'apprentissage. Les valeurs de PRD et PRD1 qui constituent des mesures de performances ont montrées que les résultats obtenus sont encourageantes et en ce qui concerne le rapport de compression CR ça nécessite un codage efficace de taille variable comme le fameux codage de Huffman pour l'améliorer d'avantage.

Abstract

The electrocardiogram (ECG) is the signal that represents time variations of the electric activity of the heart. It constitutes an effective tool for diagnostic of heart anomalies, for this reason we are in need for long recordings of ECG that reflect the state of the heart but this will cause problems of storing or transmission of the ECG to a distant interpretation center, this has motivated research works towards compression

Recently, Neural Networks have occupied a great place in signal processing, specifically Recurrent Neural Networks because they are capable to adapt to time variations of non linear and non stationary signals such as the ECG signal.

In this regard, a compression algorithm via parameters extraction using Recurrent Neural Networks has been developed and tested on electrocardiography signals of the “MIT-BIH Arrhythmia Data Base” and results obtained are presented, discussed and compared with those of some most recent algorithms of ECG compression.