

UNIVERSITE MOHAMED BOUDIAF - M'SILA
FACULTE DE TECHNOLOGIE



DÉPARTEMENT D'ELECTRONIQUE

MEMOIRE de fin d'étude

Présenté pour l'obtention du diplôme de MASTER

Domaine : Sciences et Technologies

Filière : ELECTRONIQUE

Spécialité : Instrumentation

Par :

HALITIM Ibtisem

BELAID Souaad

Intitulé

**Modélisation et identification de séries temporelles
par les réseaux de neurones et les algorithmes
méta-heuristiques**

Soutenu publiquement le : devant le jury composé de :

Dr. LADJAL Mohamed

Université de M'Sila

Encadreur

Dr. OUALI Mohamed Assam

Université de M'Sila

Encadreur

Dr. KHENNOUF Salah

Université de M'Sila

President

Dr. DJERIOUI Mohamed

Université de M'Sila

Examineur

Promotion : 2019 /2020

الله أكبر

ملخص ملخص

الهدف من مشروع نهاية الدراسة هذا هو توفير نموذج عصبي موثوق وفعال لتمثيل السلاسل الزمنية. تم تصميم النموذج الأساسي مبدئيًا باستخدام بيانات الإدخال / الإخراج الفعلية من السلسلة الزمنية قيد الدراسة. بعد ذلك، يتم نمذجة الخطأ بين ناتج النظام الديناميكي قيد الدراسة ومخرجات النموذج الأساسي لإنشاء نموذج الخطأ. يمثل النموذج الأخير حالات عدم يقين في النموذج الأساسي يمكن إزالتها بسهولة عن طريق طرح ناتج نموذج الخطأ من ناتج النموذج الأساسي.

الكلمات المفتاحية: السلسلة الزمنية، الشبكات العصبية، التحسين الحيوي استنادا إلى الجغرافيا الحيوية، الخوارزميات البحث المعقد.

Abstract Abstract

The objective of this end of study project is to provide a reliable and efficient neural model to represent time series. A primary model is initially designed using actual input / output data from the time series under consideration. Then, the error between the output of the dynamic system under consideration and the output of the primary model is modeled to create the error model. The latter model represents uncertainties in the primary model that can be easily removed by simply subtracting the output of the error model from the output of the Primary model.

Key words: time series, neural networks, BBO, meta-heuristic.

Résumé

L'objectif de ce mémoire de fin d'étude est d'offrir un modèle neuronal fiable et efficace pour représenter les séries temporelles. Un modèle primaire est initialement conçu à l'aide des données réelles d'entrée/ sortie de la série temporelle en considération. Ensuite, l'erreur entre la sortie du système dynamique en considération et la sortie du modèle primaire est modélisée pour créer le modèle d'erreur. Ce dernier modèle représente les incertitudes dans le modèle primaire qui peuvent être facilement supprimés par une simple soustraction de la sortie du modèle d'erreur de la sortie du modèle Primaire.

Mots clés : séries temporelles, réseaux de neurones, BBO, méta-heuristiques,.



REMERCIEMENTS

Remerciements

Nous remercions tout d'abord dieu pour nos avoir donné le courage et la force de mener à bien ce travail.

Nous remercions chaleureusement notre encadreur, **Dr. LADJAL Mohamed** et Nous remercions le Co- encadreur **Dr. OUALI Mohamed Assam** que nous lui souhaite Bonne chance dans ses études, pour son aide, son soutien et ses conseils, qui nous a motivés à accomplir cette tâche.

Nous remercions aussi **Dr. BRIK Youcef, Dr. OUDIRA Houcine et Dr. MEZAACHE AMAR** qui nous a aidé à terminer ce travail.

Nous voulons aussi à remercier les membres du jury pour avoir accepté d'évaluer ce travail.

Nous remercie tous nos enseignants du département d'électronique en particulier ceux de la filière INSTRUMENTATION pour leur orientation pendant notre formation pour toutes leurs remarques et critiques constructives.

Nous remercions également toutes les personnes présentes dans la salle.

Nous tenons à remercier tous nos professeurs depuis Primaire à Master 2.

Nos derniers remerciements vont à ma famille et *plus particulièrement à mes parents*. Pour tout ce qu'elle a fait pour moi et sans laquelle rien de ce qui est entre vos mains aujourd'hui n'aurait été réalisé.

Enfin, Je tiens à exprimer toute ma gratitude à tous ceux qui ont contribué, de près ou de loin, à la concrétisation de ce travail.

A tous je dis merci



DEDICACES

Dédicaces

Aux êtres les plus chers à mon cœur, ma mère et mon père, qui m'ont beaucoup soutenu durant mon cursus d'étude et à qui Je serai éternellement reconnaissant, qui m'ont permis de devenir Ce qui je suis aujourd'hui.

*Je dédie ce modeste travail A lou.ch
A mes frères et mes Sœurs Abdeldjabar, Zaïdi, Ahmed,
Noureddine, Fatna, Manouba, Zohra, et Nabila.*

Les femmes de mes frères Salha, Asya, souhila

À toute mon famille.

Mes amie Nabila, Souaad, et tous mes amis.

*A tous les enseignants qui m'ont aidé
De près ou de loin à obtenir mon master en électronique*

*Enfin, je remercie beaucoup tous ceux qui m'ont
encouragé et soutenu.*

Ibtisem

Dédicaces

*Tous d'abord je remercie Allah tout puissant qui m'a donné
la volonté et la force afin de réaliser ce travail, je dédie ce
travail :*

A mon père,

*Pour son grand amour, son encouragement, son sens du devoir
et ses sacrifices pour que je réussisse dans mes études.*

A ma mère,

*Pour son affection, sa patience, son encouragement
pendant les épreuves difficiles ainsi que ses prières qui
m'apportent bonheur et réussite.*

A mon seul cher sœur OUMAIMA.

*A mes frères IBRAHIM, DJAMEL, WALID, AMINE et
MOURAD.*

Qui sont toujours à mes côtés, prêts à m'aider.

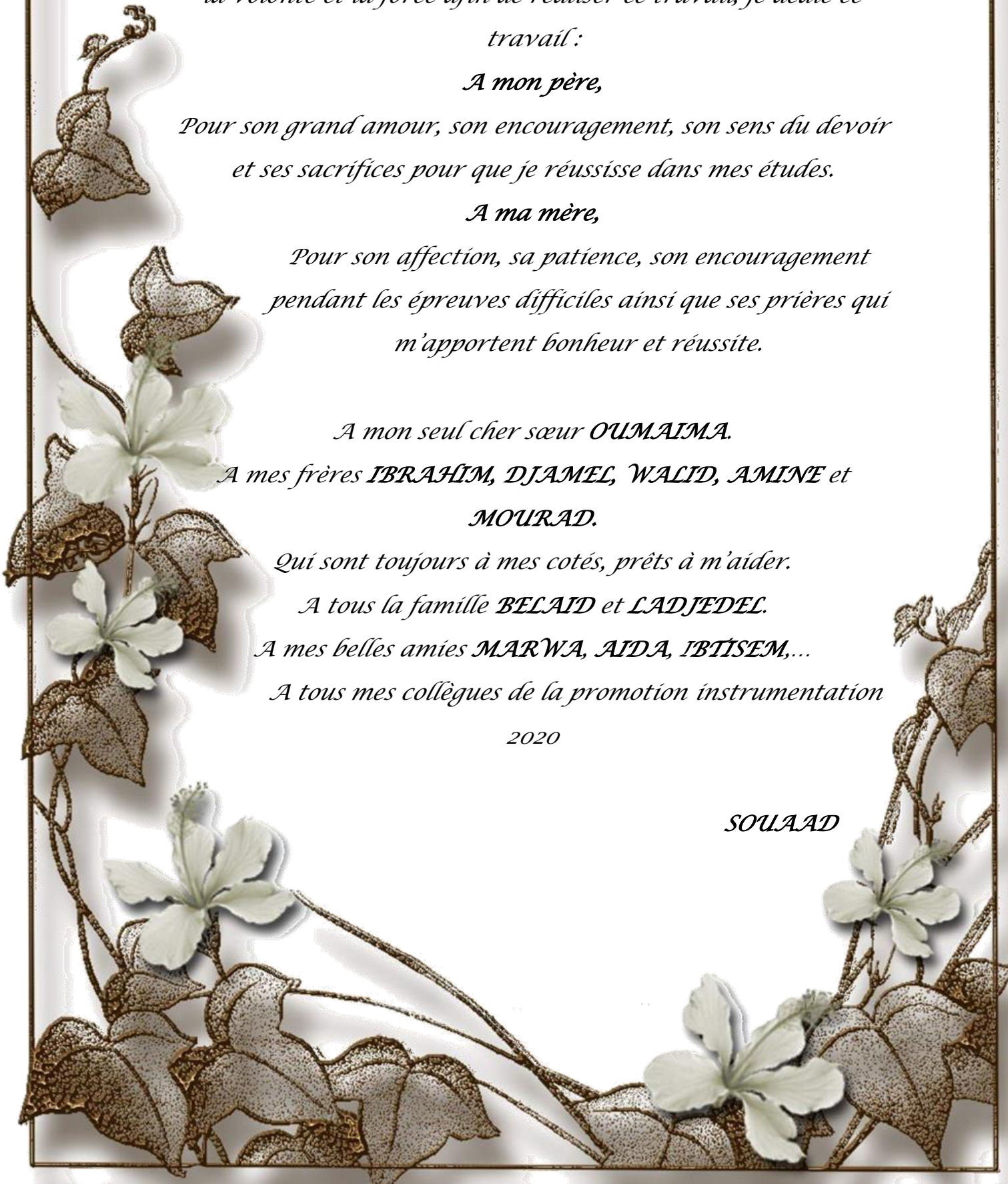
A tous la famille BELAID et LADJEDEL.

A mes belles amies MARWA, AIDA, IBTISEM,...

A tous mes collègues de la promotion instrumentation

2020

SOUAAD



A vertical scroll of aged, light brown parchment with a slightly textured surface and irregular, torn edges. The scroll is partially unrolled, showing a thick, cylindrical roll at the top and bottom. The word "SOMMAIRE" is printed in a large, bold, black, sans-serif font in the center of the unrolled section.

SOMMAIRE

SOMMAIRE

LES ABREVIATIONS.....	VII
LES SYMBOLES.....	IX
LISTE DES FIGURES	XII
LISTE DES TABLEAUX.....	XV

Introduction générale.....	2
----------------------------	---

CHAPITRE I

SERIES TEMPORELLES

1. Introduction.....	7
2. Types des séries temporelles.....	8
3. Composantes d'une Série temporelle	9
4. Analyse des séries temporelles.....	9
4.1. Objectifs de l'analyse des séries temporelles	10
5. Description schématique de l'étude complète d'une série chronologique	11
5.1. Correction des données	11
5.2. Observation de la série.....	12
5.3. Modélisation.....	12
5.4. Analyse de la série à partir de ses composantes.....	13
5.5. Diagnostic du modèle.....	13

6. Analyse de la stationnarité d'une série temporelle	14
7. Réseau de neurones artificiels pour la prédiction des séries temporelles	14
8. Conclusion.....	16

CHAPITRE II

RESEAUX DE NEURONES ARTIFICIELS

1. Introduction.....	19
2. Historique	19
3. Domaines d'application des réseaux de neurones artificiels	20
4. Présentation des Réseaux de Neurones.....	21
4.1. Définition.....	21
4.2. Neurone biologique	22
4.3. Neurone formel.....	23
4.4. Fonction d'activation.....	24
5. Architecture des réseaux de neurones.....	26
5.1. Les réseaux de neurones non bouclés	26
5.1.1. Réseaux de neurones monocouches	27
5.1.2. Réseaux de neurones multicouches.....	27
5.1.3. Réseaux de neurones à connexions locales	28
5.2. Les réseaux de neurones bouclés	29
6. Apprentissage.....	30
6.1. Apprentissage non supervisé	31
6.2. Apprentissage supervisé.....	31
6.3. Règles d'apprentissage	32
7. Applications des réseaux de neurones artificiels	33

A. Biogéographie	43
B. Principe de l'algorithme BBO.....	43
C. Optimisation et biogéographie.....	44
D. Étapes de l'optimisation basée sur la biogéographie.....	45
4. Conclusion.....	50

CHAPITRE IV

MODELISATION DES SERIES TEMPORELLES PAR *RNA* ET *BBO*

1. Introduction.....	53
2. Méthode proposée pour la modélisation des séries temporelles	54
2.1. Paramètres du modèle neuronal à ajuster	54
2.2. Fonction objectif.....	55
2.3. Identification du modèle primaire.....	55
2.4. Identification du processus d'erreur.....	57
2.5. Conception du modèle final	57
3. Résultats de simulation	59
3.1. Paramètres d'algorithme d'optimisation BBO	59
3.2. Modélisation d'un signal électrocardiogramme synthétique	59
3.3. Modélisation d'un signal électrocardiogramme réel.....	60
3.4. Modélisation de fonction de référence Mackey-Glass.....	63
4. Conclusion.....	64
Conclusion générale.....	66
Bibliographie.....	69



**LISTES
DES
ABREVIATIONS**

Abréviations	Notation
GARCH	Generalized Autogressive Conditional Heteroskedastici
ECG	ElectroCardioGraphe (E lectro C ardio G raph)
EMG	ElectroMyoGramme (E lectro M yo G ram)
EEG	ElectroEncephaloGramme (E lectro E ncephalo G ram)
IA	Intelligence Artificielle
AR	Autorégressif (A uto R egressive)
MA	Moyenne Mobile (M oving A verage)
ARMA	Moyenne Mobile Autorégressif (A utoregressive M oving A verage)
ARIMA	Moyenne Mobile Intégré Autorégressif (A utoregressive I ntegreted M oving A verage)
SARIMA	Moyenne Mobile Intégrée Autorégressif Saisonnière (S easonal A utoregressive I ntegreted M oving A verage)
RNA	Réseaux de Neurones Artificiels (A rtificial N eural N etworks)
PMC	Perceptron Multi Couches (M ulti- L ayer P erceptron)
RBF	Fonction à Base Radiale (R adial B asis F unction)
SOM	Carte Auto-Organisatrice (S elf O rganozing M ap)
ART	Théorie de Résonance Adaptative (A daptive R esonance T heory)
NP	Non déterministe Polynomial (N ondeterministic P olynomial time)
EA	Algorithmes Evolutionnaires (E volutionary A lgorithms)
DE	L'Evolution Différentielle (D ifferential E volution)
PE	Programmation Evolutionnaire (E volutionary P rogramming)
SE	Stratégies d'Evolution (E volution S trategies)
AG	Algorithmes Génétiques (G enetic A lgorithms)

PG	Programmation Génétique (Genetic Programming)
ACO	Optimisation par Colonies de Fourmis (Ant Colony Optimization)
BBO	Optimisation par Biogéographie (Biogeography-Based Optimization)
PSO	Optimisation par Essaims Particulaires (Particle Swarm Optimization)
AIS	L'Optimisation par Systèmes Immunitaires Artificiels (Artificial Immune Systems)
HSI	Indice d'Adéquation de l'Habitat (Habitat Suitability Index)
SIV	Variable d'Indice d'Adéquation (Suitability Index Variables)
MSE	l'Erreur Quadratique Moyenne (Mean Square Error)



**LISTES
DES
SYMBOLES**

Symboles	Notation
λ_i	Taux d'immigration d'un habitat (<i>BBO</i>)
μ_i	Taux d'émigration d'un habitat (<i>BBO</i>)
ε_t	Une suite de variables aléatoires
φ, ψ	Sont des fonctions non linéaires.
σ	l'écart type
a_i	Les amplitudes des Gaussienne.
b_i	Les largeurs des Gaussiennes
Cov	La covariance
E	La moyenne
E_p	Le processus d'erreur
\hat{E}_p	Le modèle neuronal d'erreur
e	la sortie du processus d'erreur
\hat{e}	La sortie du modèle neuronal d'erreur
f	fonction inconnue représentant la série temporelle
\hat{f}_p	Le modèle neuronal primaire
\hat{f}_F	Le modèle final
g_{max}	Nombre maximum de générations (<i>BBO</i>)
H	Habitat (<i>BBO</i>)
m	le taux de mutation
m_i	Probabilité de mutation (<i>BBO</i>)
m_{max}	Taux maximum de mutation (<i>BBO</i>)
N	La taille de l'essaim

P_i	Probabilité d'existence (BBO)
P_{max}	Probabilité maximale d'existence (BBO)
S_t	Une composante saisonnière
T	Le temps
Var	La variance
$V(t)$	La vitesse
W_i	Poids synaptiques
X	Entrée d'un réseau de neurones
$X(t)$	La position
$X_{CT,t}$	La série corrigée de la tendance
$X_{CVS,t}$	La série corrigée des variations saisonnières
$X_{SLP,t}$	La série lissée des prédictions
Y	Sortie d'un réseau de neurones
Z_0	La ligne de base du signal ECG
Z_t	La tendance
a_1, a_2	Coefficients autorégressifs
C_1, C_2	Coefficients d'accélération
r_1, r_2	nombres aléatoires
S_1, S_2	Solutions candidates
y_k, \hat{y}_k	sont la mesure réelle et son estimation, respectivement



**LISTES
DES
FIGURES**

La figure	Le titre	Page
Figure I.1	Exemples des séries temporelles	08
Figure II.1	Diagramme du réseaux de neurones	22
Figure II.2	Neurone biologique	23
Figure II.3	Neurone formel	24
Figure II.4	Les réseaux de neurones non bouclés	27
Figure II.5	Schéma d'un réseau de neurones monocouche	27
Figure II.6	Schéma d'un réseau de neurones Perceptron multicouches	28
Figure II.7	Schéma d'un réseau de neurones à connexions locales	29
Figure II.8	Schéma de réseau de neurones bouclé	29
Figure II.9	Forme canonique d'un réseau de neurones bouclé	30
Figure II.10	Apprentissage non supervisé	31
Figure II.11	Apprentissage supervisé	32
Figure III.1	Catégories d'algorithmes d'optimisations	38
Figure III.2	Migration des espèces	43
Figure III.3	Organigramme général de l'algorithme <i>BBO</i>	46
Figure III.4	Illustration de deux solutions candidates S1 et S2	49
Figure IV.1	Paramètres du réseau de neurones à optimiser	54
Figure IV.2	Identification du modèle primaire	55
Figure IV.3	Processus d'erreur	56
Figure IV.4	Identification du processus d'erreur	57
Figure IV.5	Modèle final \hat{f}_F	57

Figure IV.6	Résultats de simulation du signal électrocardiogramme synthétique : (a) Modèle Primaire, (b) Modèle Final, (c) Erreurs de Modélisation	61
Figure IV.7	Résultats de simulation du signal électrocardiogramme réel : (a) Modèle Primaire, (b) Modèle Final, (c) Erreurs de Modélisation	62
Figure IV.8	La série temporelle Macky Glass avec Tau=17	62
Figure IV.9	Le résultat de modélisation de la série temporelle Macky Glass	63



**LISTES
DES
TABLEAUX**

Le tableau	Le Titre	Page
Tableau II.1	Différentes fonctions d'activations utilisées dans les <i>RNA</i>	25
Tableau III.1	Terminologie de la <i>BBO</i> .	44
Tableau III.2	Paramètres de la <i>BBO</i>	47
Tableau IV.1	Paramètre du modèle d' <i>ECG</i> (4)	59



**INTRODUCTION
GENERALE**

INTRODUCTION GENERALE

La prévision des séries temporelles est un sujet de recherche très actif dans le domaine de la science et de l'ingénierie. L'objectif principal de l'analyse d'une série temporelle est la prévision de ses futures réalisations. Afin de réaliser cet objectif, une première étape de modélisation de la série est nécessaire. Cette étape consiste à sélectionner, parmi une famille de modèles correspondant à des approximations de la réalité, celui qui décrit le mieux la série en question.

Il existe deux approches différentes pour modéliser les séries temporelles en fonction de la théorie ou de l'hypothèse concernant la relation dans les données. Les méthodes traditionnelles telles que la régression par séries temporelles, le lissage exponentiel et le modèle autorégressif (**AR**) sont basées sur des modèles linéaires. En d'autres termes, ils supposent que la valeur future d'une série temporelle est liée linéairement aux observations passées.

Surtout le modèle autorégressif (**AR**) (représentatif des modèles linéaires), est un modèle de régression pour les séries temporelles dans lequel la série est expliquée par ses valeurs passées que par d'autres variables.

Il ya plusieurs façons de prédire une série temporelle, y'en a celles qui utilisent une méthode de décomposition. C'est une méthode qui a l'avantage d'être rapide à entraîner, d'être interprétable et qui s'applique à de nombreuses situations variées. Le concept de cette méthode est de simplifier au maximum la série initiale, en procédant à des transformations successives. Une fois la série simplifiée, ils utilisent des algorithmes d'apprentissage automatique. La dernière étape consiste à inverser les transformations pour remettre les prédictions dans le même contexte que la série initiale.

Un réseau de neurones artificiels est un système dont la conception est à l'origine schématiquement inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques.

Une méta-heuristique est un algorithme d'optimisation visant à résoudre des problèmes d'optimisation difficiles pour lesquels on ne connaît pas de méthode classique plus efficace. Les méta-heuristiques sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum global. Elles se comportent comme des algorithmes de recherche, tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution. Il existe un grand nombre de méta-heuristiques différentes, allant de la simple recherche Locale à des algorithmes complexes de recherche globale. Ces méthodes utilisent cependant un haut niveau d'abstraction, leur permettant d'être adaptées à une large gamme de problèmes différents. En d'autres termes, ces algorithmes se veulent des méthodes génériques pouvant optimiser une large gamme de problèmes différents, sans nécessiter de changements profonds dans l'algorithme employé.

L'objectif de ce mémoire de fin d'étude est de mettre en évidence en premier lieu les notions de base relatives aux séries temporelles et leurs modélisations. Puis nous allons présenter une structure de modélisation des séries temporelles à base de réseaux de neurones artificiels et les algorithmes d'optimisation

Pour ce faire, nous avons présenté ce manuscrit de la façon suivante : En plus d'une introduction générale et une conclusion générale, qui résume notre étude, le présent travail est subdivisé en quatre chapitres organisés comme suit :

- **Le premier chapitre** présente les séries temporelles, les différents types de séries temporelles et leur analyse, nous allons parler aux objectifs de leur analyse. Suivant la description schématique de l'étude complète d'une série temporelle. Nous terminerons par les composants d'une série temporelle et l'analyse de la stationnarité d'une série chronologique.
- **Dans le deuxième chapitre**, nous allons présenter les notions de base relatives aux réseaux neurones artificiels (RNA), en décrivant l'historique et domaines d'application des réseaux de neurones artificiels en suite la présentation des réseaux de neurones et leurs architectures.

- **Le troisième chapitre** est consacré à la présentation des notions de base relatives aux algorithmes d'optimisation.
- **Le quatrième et le dernier chapitre** est consacré à la simulation et la modélisation des séries temporelles à base des réseaux de neurones artificiels et l'algorithme d'optimisation à base de biogéographie.

L'objectif de notre travail est d'offrir un modèle neuronal fiable et efficace pour représenter les séries temporelles.



CHAPITRE I

SERIES

TEMPORELLES

CHAPITRE I

SERIES TEMPORELLES

Résumé

Une série temporelle, ou série chronologique, est une suite de valeurs numériques représentant l'évolution d'une quantité spécifique au cours du temps. Nous exposons dans ce premier chapitre, un aperçu détaillé des séries temporelles en incluant : leurs types, leurs méthodes d'analyses ainsi que les objectifs souhaités de l'étude des séries temporelles.

Sommaire

1. Introduction.....	7
2. Types des séries temporelles	8
3. Composantes d'une Série temporelle	9
4. Analyse des séries temporelles	9
5. Description schématique de l'étude complète d'une série chronologique	11
6. Analyse de la stationnarité d'une série temporelle.....	14
7. Réseau de neurones artificiels pour la prédiction des séries temporelles	14
8. Conclusion	16

1. Introduction

Une série temporelle est un ensemble d'observations où chacune est associée à un instant de temps unique t . Une série temporelle peut aussi être construite en rassemblant un nombre fini d'observations. Dans ce cas, on peut la représenter par un vecteur de n variables aléatoires où les réalisations sont les observations réelles. Il est possible de considérer le nombre d'observations comme un nombre infini, dans ce cas, on parle des processus stochastiques discrets. Dans la pratique, la première représentation est la plus utilisée, où les séries temporelles sont représentées par des vecteurs (indexés par le temps) d'observations successives.

Une série temporelle est une suite (x_1, \dots, x_n) de données indexées par le temps. Le nombre n est appelé la longueur de la série. Il est la plupart du temps bien utile de représenter la série temporelle sur un graphe construit de la manière suivante : en abscisse le temps, en ordonner la valeur de l'observation à chaque instant. Pour des questions de lisibilité, les points ainsi obtenus sont reliés par des segments de droite. Le graphe apparaît donc comme une ligne brisée (**Figure I.1**) [1].

Par exemple, les ventes quotidiennes d'un produit peuvent être représentées sous forme de série temporelle. Dans la distribution ou la fabrication, les séries temporelles sont importantes parce que ce sont les représentations les plus canoniques des flux de marchandises vendues ou produites. La représentation des données de l'entreprise sous forme de séries temporelles aide généralement les responsables à visualiser l'activité de leur entreprise.

Les signaux biologiques (**Figure I.1**) telles que le signal électrocardiogramme (*ECG*) qui représente l'activité électrique du cœur, le signal électromyogramme (*EMG*) qui représente l'activité électrique des muscles et le signal électroencéphalogramme (*EEG*) qui représente l'activité électrique du cerveau peuvent être considéré comme des séries temporelles.

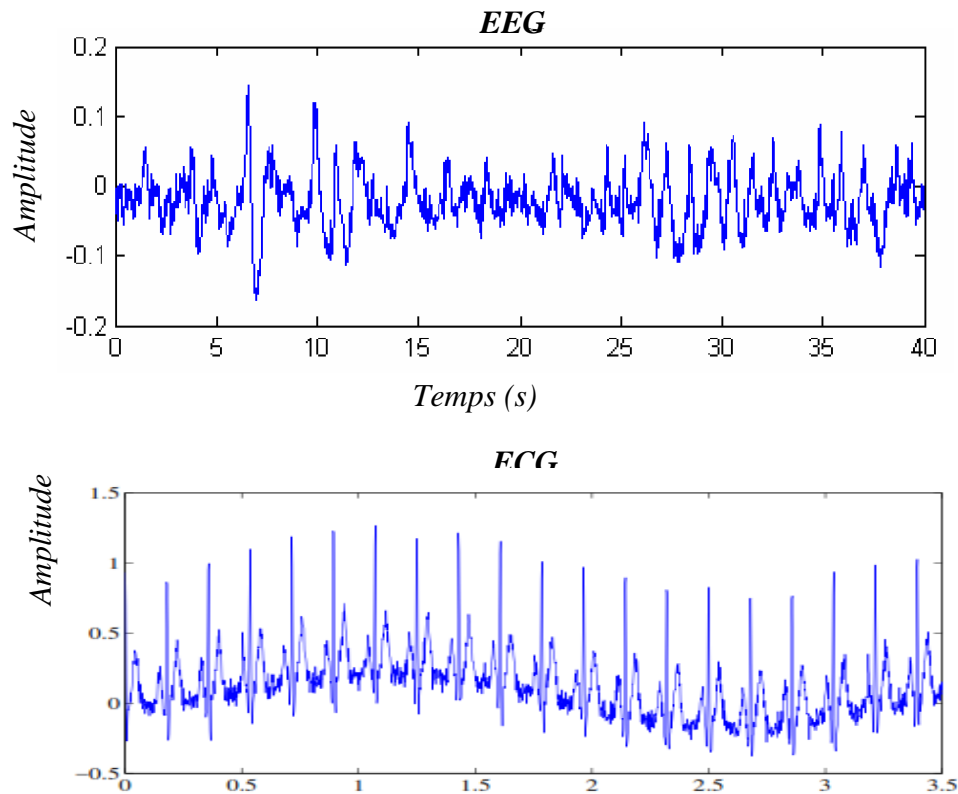


Figure I.1 : Exemples des séries temporelles.

2. Types des séries temporelles

Une série temporelle est un ensemble d'observations mesurées de façon séquentielle dans le temps. Ces mesures peuvent être continues dans le temps ou être prises comme un ensemble discret de points dans le temps. Par convention, ces deux types de séries sont appelés respectivement des séries temporelles continues et discrètes. Pour une série temporelle continue, la variable observée est généralement une variable continue enregistrée sur une ligne continue, comme la mesure de l'activité électrique du cœur enregistrée à partir d'un électrocardiographe (*ECG*). La méthode habituelle d'analyser une telle série est d'échantillonner (ou numériser) la série à des intervalles de temps égaux pour donner une série temporelle discrète. Peu ou aucune information n'est perdue par ce processus à condition que l'intervalle d'échantillonnage soit assez petit. En fait, les séries chronologiques discrètes peuvent être présentées par trois différentes manières :

- **Soit échantillonnées à partir des séries continues** (par exemple la température mesurée à un intervalle de temps).

- **Soit agrégées sur une période de temps** (par exemple les ventes totales des mois successifs).
- **Intrinsèquement comme une série discrète** (par exemple, le dividende des années consécutives versé par une entreprise).

Pour les trois types de séries temporelles discrètes, les données sont généralement échantillonnées à des intervalles de temps égaux [2].

3. Composantes d'une Série temporelle

La **tendance** représente l'évolution moyenne à long terme de la variable étudiée. La **composante saisonnière** est une composante périodique dont la période est généralement dépendante de notre environnement naturel (les saisons, alternance jour/nuit, ...). La **composante irrégulière** regroupe tout ce qui n'est pas pris en compte par les composantes précédentes. On considère qu'il s'agit d'une composante aléatoire qui correspond à un résidu. On ajoute parfois à cela, une **composante cyclique**, qui présente théoriquement un comportement périodique, dont la période est suffisamment importante pour être qualifiée de mouvement à moyen ou à long terme. Cette composante n'est pas nécessairement de période ni d'amplitude constante. (En pratique, elle est difficile à isoler, car on ne dispose en général pas de séries stables d'une longueur correspondant à plusieurs cycles). On notera que l'une ou l'autre des composantes citées ci-dessus peut éventuellement être absente de la série ; à l'inverse, certaines séries peuvent présenter plusieurs composantes du même type (par exemple, plusieurs composantes saisonnières) [3].

4. Analyse des séries temporelles

L'analyse des séries temporelles est un problème important en sciences naturelles et en sciences d'ingénierie, d'un point de vue prédiction et compréhension du comportement du système étudié. Il existe de nombreuses applications d'analyse des séries temporelles dispersées dans la littérature de l'économie, système d'identification, la chimie, les statistiques, la reconnaissance des formes et les réseaux de neurones [4].

4.1. Objectifs de l'analyse des séries temporelles

L'étude d'une série temporelle permet d'analyser, de décrire et d'expliquer un phénomène au cours du temps et d'en tirer des conséquences pour des prises de décision. L'un des objectifs principaux de l'étude d'une série temporelle est la prévision qui consiste à prévoir les valeurs futures X_{T+H} ($h = 1, 2, 3, \dots$) de la série temporelle à partir de ses valeurs observées jusqu'au temps : X_1, X_2, \dots, X_T , parmi les autres objectifs avoués de l'étude des séries temporelles, figure le problème de l'estimation d'une tendance où l'impact d'un événement sur une variable [5].

Les principaux objectifs de l'analyse des séries temporelles sont les suivants :

- ✓ **La description** : pour décrire les données en utilisant des méthodes statistiques et/ou des méthodes graphiques. Un graphe des données est particulièrement précieux [2].
- ✓ **La modélisation** : pour trouver un modèle statistique approprié dans le but de décrire le processus de génération des données, un modèle uni-varié d'une variable donnée est basé seulement sur les valeurs passées de cette variable. Par contre pour une variable donnée, un modèle à plusieurs variables peut se fonder, non seulement, sur les valeurs passées de cette variable, mais aussi, sur les valeurs présentes et passées d'autres variables (outil de prédiction). Dans le dernier cas, la variation d'une série peut aider à expliquer la variation d'autres séries. Bien sûr, tous les modèles sont des approximations et la construction du modèle est un art autant qu'une science [2].
- ✓ **La prédiction** : pour estimer la valeur future de la série, la plupart des auteurs utilisent les termes 'prévision' et 'prédiction' [2].
- ✓ **Le contrôle** : les bonnes prévisions permettent à l'analyste de prendre des mesures afin de contrôler un processus donné, s'il est un processus industriel, économique ou autre [2].

5. Description schématique de l'étude complète d'une série chronologique

Comme nous venons de le voir, l'un des objectifs principaux de l'étude d'une série temporelle est la prévision des valeurs futures de cette série. Pour cela, on a besoin de connaître ou tout au moins de modéliser le mécanisme de production de la série temporelle.

Notons que les variables X_t d'une série temporelle ne sont le plus souvent ni indépendantes (on peut s'attendre en effet à ce que des observations relativement proches dans le temps soient liées) ni identiquement distribuées (dans la plupart des cas, le phénomène évolue, se modifie au cours du temps ce qui entraîne que les variables le décrivant ne sont pas équidistribuées). Cela nécessite des méthodes statistiques de traitement et de modélisation spécifiques puisqu'en particulier dans un cadre standard (celui de la description d'un échantillon) les méthodes statistiques classiques sont basées sur des hypothèses d'indépendance. Schématiquement, les principales étapes de traitement d'une série chronologique sont les suivantes :

- ✓ Correction des données ;
- ✓ Observation de la série ;
- ✓ Modélisation ;
- ✓ Analyse de la série à partir de ses composantes ;
- ✓ Diagnostic du modèle - ajustement au modèle ;
- ✓ Prédiction (= prévision).

5.1. Correction des données

Avant de se lancer dans l'étude d'une série temporelle, il est souvent nécessaire de traiter, modifier les données brutes. Par exemple,

- évaluation de données manquantes, remplacement de données accidentelles.
- découpage en sous-séries.
- Standardisation afin de se ramener à des intervalles de longueur fixe. Par exemple, pour des données mensuelles, on se ramène au mois standard en calculant la moyenne journalière sur le mois (total des observations sur le mois divisé par le nombre de jours du mois).

- transformation des données : pour des raisons diverses, on peut être parfois amenés à utiliser des données transformées. Par exemple en économie, il utilise la famille de transformations de Box-Cox :

$$Y_t = \frac{1}{\lambda} [(X_t)^\lambda - 1] \quad \lambda \in \mathbb{R}^* \quad (1)$$

5.2. Observation de la série

Une règle générale en statistique descriptive consiste à commencer par voir les données avant d'effectuer le moindre calcul. Ainsi, une fois la série corrigée et pré-traitée, on trace son graphique, c'est-à-dire la courbe de coordonnées (t, x_t) . L'observation de ce graphique est souvent une aide à la modélisation de la série temporelle et permet de se faire une idée des différentes composantes de la série chronologique.

5.3. Modélisation

Modèle est une image simplifiée de la réalité qui vise à traduire les mécanismes de fonctionnement du phénomène étudié et permet de mieux les comprendre. Un modèle peut être meilleur qu'un autre pour décrire la réalité et bien sûr, plusieurs questions se posent alors : comment mesurer cette qualité ? Comment diagnostiquer un modèle ? Nous présentons dans cette section une petite liste qui sert à résumer et classer les différents modèles.

On peut distinguer trois types de modèle : les modèles d'ajustement, les modèles auto-projectifs, et les modèles explicatifs :

- ✓ **Les modèles d'ajustement** : la série est générée par un modèle du type $X_t = a + S_t + u_t$, où a est une constante inconnue représentant la tendance et S_t est une fonction périodique du temps, nulle en moyenne, qu'on appelle mouvement saisonnier : u_t , appelé partie irrégulière, est une variable aléatoire centrée. Cette partie irrégulière est souvent petite par rapport aux deux autres, mais pas pour autant négligeable ; en fait, c'est souvent dans cette partie que l'on trouve les fluctuations les plus. De manière plus générale, on peut proposer un modèle du type $X_t = f(t, u_t)$, où f est une fonction indexée par un nombre fini de paramètres inconnus et u_t est une variable aléatoire centrée. Les hypothèses faites sur les variables aléatoires u_t induisent des méthodes

d'estimation de la fonction f . Par exemple, dans le cas où les u_t ont même variances et sont non corrélés, on utilise la méthode des moindres carrés ordinaires.

- ✓ **Les modèles auto projectifs** : on suppose ici que X_t est fonction de ses valeurs passées et d'une perturbation aléatoire u_t : $X_t = f(X_{t-1}, X_{t-2}, \dots, u_t)$.

Une classe de tels modèles, utiles pour la prévision, sont les modèles ARIMA (*Auto Regressive Integrated Moving Average*).

- ✓ **Les modèles explicatifs** : la variable X_t est exprimée en fonction d'un vecteur de variables observables, dites exogènes, X_t^* et d'une perturbation aléatoire u_t : $X_t = f(X_t^*, u_t)$; X_t^* est soit déterministe, soit aléatoire et indépendant de u_t . Il existe deux types de modèles explicatifs, le modèle statique et le modèle dynamique.

- Dans le premier, les variables X_t^* ne contiennent pas de valeurs passées de X_t et les u_t sont indépendants entre eux.
- Un modèle explicatif peut être dynamique soit parce que les perturbations u_t sont auto-corrélées, soit parce que X_t^* contient des valeurs passées de X_t , c'est-à-dire des variables dites endogènes retardées.

5.4. Analyse de la série à partir de ses composantes

Une fois l'étape de modélisation effectuée, on étudie les composantes du modèle les une après les autres :

- Tendances, facteurs saisonniers et fluctuations irrégulières ;
- La série corrigée de la tendance ;
- La série corrigée des variations saisonnières ;
- La série lissée des prédictions.

5.5. Diagnostic du modèle

Une fois le modèle construit et ses paramètres estimés, on vérifie que le modèle proposé est bon, c'est-à-dire l'ajustement au modèle :

- en étudiant les résidus ;
- en faisant des tests.

5.6. Prédiction

Enfin, une fois ces différentes étapes réalisées, nous sommes en mesure de faire de la prédiction [6].

6. Analyse de la stationnarité d'une série temporelle

La modélisation des séries temporelles, nécessite, que ces dernières soient stationnaires. Autrement dit que la série ne comporte ni tendance, ni cycle et ni saisonnalité. Cette notion de stationnarité représente un point crucial dans l'économétrie des séries temporelles, où l'estimation des séries non stationnaires conduit à des régressions fallacieuses ou illusoire. Pour éviter ces estimations fallacieuses, les économètres procèdent à la stationnarisation des séries temporelles.

Une série temporelle est dite stationnaire si sa moyenne, sa variance et sa covariance sont constantes (ne sont pas affecté par le temps). Soit une série temporelle notée à valeurs réelles et en temps discret. S est dite stationnaire (stationnarité du second ordre) si :

$$\begin{cases} E(S_i) = \mu & \forall i = 1 \dots t \\ Var(S_i) = \sigma^2 \neq \infty & \forall i = 1 \dots t \\ Cov(S_i, S_{i-k}) = f(k) \forall i = 1 \dots t \forall k = 1 \dots t \end{cases} \quad (2)$$

Ces conditions stipulent que l'espérance, la variance et la covariance sont constantes au cours du temps. A noté que la variance doit être finie. Si l'une de ces conditions n'est pas vérifiée, on parle donc de non-stationnarité [7].

7. Réseau de neurones artificiels pour la prédiction des séries temporelles

Les réseaux de neurones artificiels (**RNA**) ont été utilisés avec succès pour résoudre un large éventail de systèmes. Le succès des applications des RNA est lié en réalité, au fait que ces réseaux soient correctement formés et configurés, ils deviennent capables d'approximer avec exactitude chaque fonction mesurable. Les neurones apprennent les patterns cachés dans les données et généralisent ces modèles, même en présence de bruit ou d'informations manquantes. Les prédictions sont effectuées par le RNA sur la base des données observées [8] [9], par exemple, mentionnent les inconvénients

suivants des modèles des séries temporelles statistiques que les modèles de réseaux de neurones pourraient résoudre :

- ✓ Sans expertise, il est possible de ne pas spécifier la forme fonctionnelle reliant les variables indépendantes et dépendantes, et ne pas parvenir à faire des transformations de données nécessaires.
- ✓ Les valeurs extrêmes peuvent conduire à des estimations biaisées des paramètres du modèle.
- ✓ Les modèles des séries temporelles sont souvent linéaires mais peuvent ne pas capturer le comportement non-linéaire [10].

L'observation des valeurs passées d'un phénomène afin d'anticiper son comportement futur constitue le cœur de la prévision [8]. Puisque le problème de prévision consiste à prédire la valeur future d'une variable ou d'un ensemble de variables basées sur les patterns passés, l'intérêt doit se focaliser sur le traitement temporel avec les RNA [11].

La prévision quantitative est basée sur les tendances de l'extraction du passé et leur projet dans le futur, en rendant le RNA très convenable, pour au moins deux raisons :

- **Premièrement**, il a été prouvé que le RNA est capable de réaliser une approximation numérique de toute fonction continue, avec la précision désirée. Ainsi, le RNA peut être considéré comme un modèle multi-varié, non linéaire et non paramétrique.
- **Deuxièmement**, le RNA utilise les méthodes « data-driven ». Ceci élimine le besoin d'élaborer un modèle mathématique et une estimation des paramètres.

À partir d'une paire de vecteurs d'entrée-sortie, les RNA sont capables d'établir une relation entre les entrées et les sorties et de la stocker dans les poids des neurones [12]. Une variété d'architectures de réseaux de neurones a été examinée pour résoudre le problème de la prédiction de séries temporelles. Ces architectures comprennent : le Perceptron Multi Couches (*PMC*) [13] [14] [15] [16] [17], les réseaux récurrents [17], les fonctions à base radiale (*RBF*) [18] [19].

8. Conclusion

Les séries temporelles représentent des données historiques mesurées d'une manière ordonnée. Ces dernières sont prises dans le temps, ce qui implique leur nature dynamique. Dans ce chapitre, nous avons présenté des généralités sur les séries temporelles, leurs types, leurs méthodes d'analyses ainsi que les objectifs de l'analyse et de l'étude des séries temporelles. Le prochain chapitre sera consacré aux réseaux de neurones artificiels et leurs applications.



CHAPITRE II

**RESEAUX DE
NEURONES**

CHAPITRE II

RESEAUX DE NEURONES ARTIFICIELS

Résumé

Les réseaux de neurones artificiels (Artificial neural networks en anglais) constituent une branche spécifique de la recherche en informatique et en neuro-informatique. Il existe différents types de réseaux de neurones artificiels, offrant différentes possibilités de traitement de l'information. Dans ce chapitre on va essayer de donner les notions essentielles pour comprendre les architectures, le fonctionnement et les applications des réseaux de neurones artificiels.

Sommaire

1. Introduction	19
2. Historique	19
3. Domaines d'application des réseaux de neurones artificiels	20
4. Présentation des Réseaux de Neurones	21
5. Architecture des réseaux de neurones	26
6. Apprentissage	30
7. Applications des réseaux de neurones artificiels	33
8. Conclusion	34

1. Introduction

Le cerveau humain a une fantastique puissance de traitement de l'information si l'on considère ses capacités à prendre en charge certaines tâches nécessaires pour obtenir un comportement intelligent. La nature de l'intelligence a longtemps été un sujet difficile et controversé. D'une certaine manière, chacun a une idée assez vague de ce qu'est l'intelligence : la capacité à observer, à comprendre, à se souvenir, à résoudre des problèmes, à apprendre, à créer, etc. Deux disciplines sont concernées par la définition et la modélisation du comportement intelligent : ce sont la psychologie cognitive et l'intelligence artificielle.

Les techniques de l'intelligence artificielle telles que les systèmes experts, la logique floue, les algorithmes génétiques et les réseaux de neurones artificiels (*RNA*) ont été largement utilisées dans le domaine de l'électronique.

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. Toute structure hiérarchique de réseaux est évidemment un réseau.

2. Historique

Les recherches sur les méthodes neuronales de traitement de l'information en vue de modéliser le comportement du cerveau humain ne sont pas récentes, en :

- **1890** : W. James, célèbre psychologue américain introduit le concept du mémoire associative et propose ce qui deviendra une loi de fonctionnement pour l'apprentissage sur les réseaux de neurones connus plus tard sous le nom de loi de Hebb.
- **1943** : J. Mc Culloch et W. Pitts laissent 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 (tout au moins au niveau théorique).
- **1949** : D. Hebb, physiologiste américain explique le conditionnement chez l'animal par les propriétés des neurones eux-mêmes. Ainsi, un conditionnement du type pavlovien tel que, nourrir tous les jours à la même heure un chien,

entraîne chez cet animal la sécrétion de salive à cette heure précise même en l'absence de nourriture. La loi de modification des propriétés des connexions entre neurones qu'il propose explique en partie ce type de résultats expérimentaux [20]. Le premier réseau de neurones artificiels apparaît en 1958, grâce aux travaux de Rosenblatt qui conçoit le perceptron. Ce dernier est inspiré du système visuel (en terme d'architecture neurobiologique) et possède une couche de neurones d'entrée une couche de sortie (décisionnelle). Ce réseau parvient à apprendre, à identifier des formes simples et à calculer certaines fonctions logiques [21].

- **1969** : M. Minsky et S. Papert publient un ouvrage qui met en exergue les limitations théoriques du perceptron. Limitations alors connues, notamment concernant l'impossibilité de traiter par ce modèle des problèmes non linéaires. Ils étendent implicitement ces limitations à tout modèle 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'**IA** (Intelligence Artificielle) et les systèmes à bases de règles [20].

Il faudra attendre le début des années 80 pour que l'intérêt pour ce domaine soit de nouveau présent [20]. En effet, Hopfield démontre en 1982 tout l'intérêt d'utiliser les réseaux récurrents (dits "feed-back") pour la modélisation des processus. Les réseaux récurrents constituent alors la deuxième grande classe de réseaux de neurones, avec les réseaux type perceptron (dit "feed-forward").

En parallèle des travaux de Hopfield, Werbos conçoit son algorithme de rétro-propagation qui ne sera pourtant popularisé qu'en 1986 par Rumelhart.

3. Domaines d'application des réseaux de neurones artificiels

Aujourd'hui, les réseaux de neurones artificiels ont de nombreuses applications dans des secteurs très variés :

- ✓ **Traitement d'images** : reconnaissance de caractères et de signatures, compression d'images, reconnaissance de formes, cryptage, classification, etc.
- ✓ **Traitement du signal** : filtrage, classification, identification de source, traitement de la parole, etc.

- ✓ **Contrôle** : commande de processus, diagnostic, contrôle de qualité, asservissement des robots, systèmes de guidage automatiques des automobiles et des avions, etc.
- ✓ **Défense** : guidage des missiles, suivi de cible, reconnaissance du visage, radar, sonar, compression de données, suppression du bruit, etc.
- ✓ **Optimisation** : planification, allocation de ressources, gestion et finances, etc.
- ✓ **Simulation** : simulation du vol, simulation de boîte noire, prévision météorologique, recopie de modèle, ... [22].

4. Présentation des Réseaux de Neurones

4.1. Définition

Un réseau de neurones peut être considéré comme un modèle mathématique de traitement réparti, composé de plusieurs éléments de calcul non linéaire (neurones), opérant en parallèle et connectés entre eux par des poids.

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.

Les neurones artificiels sont souvent utilisés sous forme de réseaux qui diffèrent selon le type de connexions entre les neurones, une cinquantaine de types peuvent être dénombrée. En guise d'exemples nous citons : le perceptron de roseblat, les réseaux de Hopfield, etc. Ces derniers sont les plus utilisés dans le domaine de la modélisation et de la commande des procédés. Ils sont constitués d'un nombre fini de neurones qui sont arrangés sous forme de couches. Les neurones de deux couches adjacentes sont interconnectés par des poids. L'information dans le réseau se propage d'une couche à l'autre, on dit qu'ils sont de type « feed-forward ». Nous distinguons trois types de couches :

- **Couche d'entrée** : les neurones de cette couche reçoivent les valeurs d'entrée du réseau et les transmettent aux neurones cachés. Chaque neurone reçoit une valeur, il ne fait pas donc de sommation.
- **Couches cachées** : chaque neurone de cette couche reçoit l'information de plusieurs couches précédentes, effectuent la sommation pondérée par les poids,

puis la transforme selon sa fonction d'activation qui est en général une fonction sigmoïde. Par la suite, il envoie cette réponse aux neurones de la couche suivante.

- **Couche de sortie** : elle joue le même rôle que les couches cachées, la seule différence entre ces deux types de couches est que la sortie des neurones de la couche de sortie n'est liée à aucun autre neurone [23].

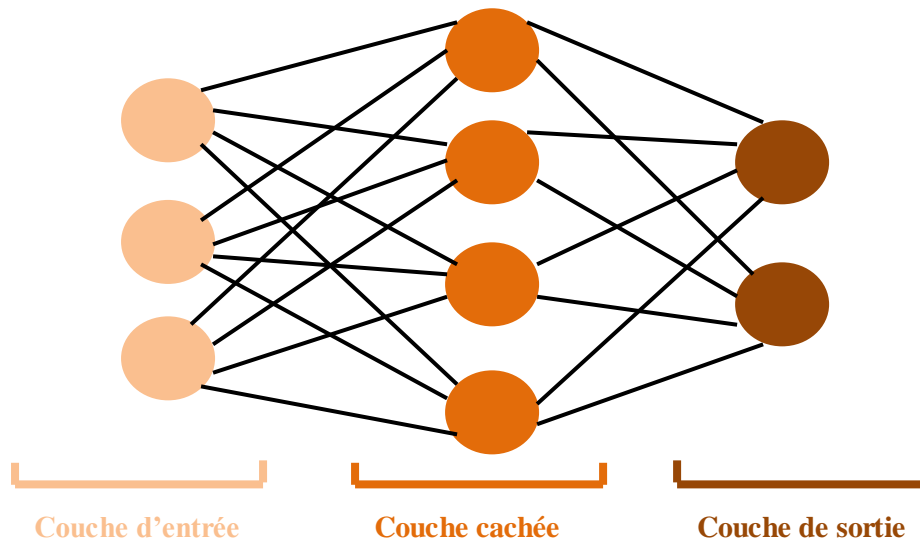


Figure II.1 Diagramme du Réseau de neurones.

4.2.Neurone biologique

Le cerveau humain possède deux hémisphères latéraux reliés par le corps calleux et d'autres ponts axonaux, il pèse moins de deux kilogrammes et contient mille milliards de cellules, dont 100 milliards sont des neurones constitués en réseaux.

Le neurone biologique (Figure II.2) est une cellule vivante spécialisée dans le traitement des signaux électriques. Les neurones sont reliés entre eux par des liaisons appelées axones. Ces axones vont eux-mêmes jouer un rôle important dans le comportement logique de l'ensemble.

Ils conduisent les signaux électriques de la sortie d'un neurone vers l'entrée (synapse) d'un autre neurone. Les neurones font une sommation des signaux reçus en entrée et en fonction du résultat obtenu vont fournir un courant en sortie [24].

Les neurones sont des cellules nerveuses décomposables en 4 parties principales (Figure II.2).

Les dendrites, sur lesquelles les autres cellules entrent en contact synaptique : c'est par les dendrites que se fait la réception des signaux. Le corps de la cellule ou noyau, c'est l'unité de traitement. L'axone est la partie où passent les messages accumulés dans le corps de la cellule. Enfin, à la sortie du neurone on trouve les synapses, par lesquelles la cellule communique avec d'autres neurones, ce sont des points de connexion par où passent les signaux de la cellule.

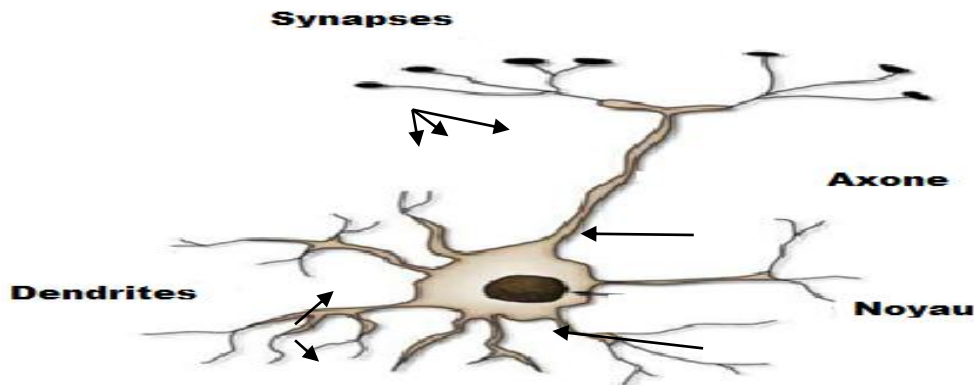


Figure II.2 : Neurone biologique [25].

Un neurone stimulé envoie des impulsions électriques ou potentielles d'action, à d'autres neurones. Ces impulsions se propagent le long de l'axone unique de la cellule. Au point de contact entre neurones, les synapses, ces impulsions sont converties en signaux chimiques. Quand l'accumulation des excitations atteint un certain seuil d'activation, le neurone en génère un potentiel d'action, pendant une durée de 1 ms. Le neurone émettant le signal est appelé le neurone pré-synaptique et celui recevant ce signal, neurone post-synaptique.

4.3. Neurone formel

Un RNA est un assemblage d'éléments de structure identique appelés cellules (ou neurones) interconnectées à l'instar des cellules du système nerveux des vertébrés. Chaque point de connexion (appelé coefficient ou poids) entre deux cellules joue le rôle d'une synapse, l'élément principal d'interaction entre les neurones. Ces connexions ou poids synaptiques ont un rôle primordial dans le fonctionnement parallèle et adaptatif des neurones d'où la notion de réseaux connexionnistes. La représentation mathématique du neurone est introduite par McCulloch et Pitts (1943). Le neurone formel est considéré comme étant la pierre angulaire de la structure des RNA et comprennent cinq éléments importants :

- **Les entrées** : les valeurs, considérées comme entrées du réseau, sont généralement normalisées.
- **Les poids et le biais** : les poids $W_{k0}, W_{k1} \dots W_{km}$ sont des valeurs reliant les neurones de deux couches différentes ou ceux de la même couche (selon le mode de connectivité). Le biais est représenté par x_0 .
- **La fonction de sommation (combinaison)** : est une fonction qui fait partie de la catégorie Maths et trigonométrie, calcule la somme des valeurs numériques uniquement.

$$Y = \sum_{i=0}^m x_i w_{ki} + x_0. \quad (1)$$

- **La fonction de transfert (d'activation)** : Est un simple effet de seuil.
- **La sortie** : le résultat obtenu Y [26].

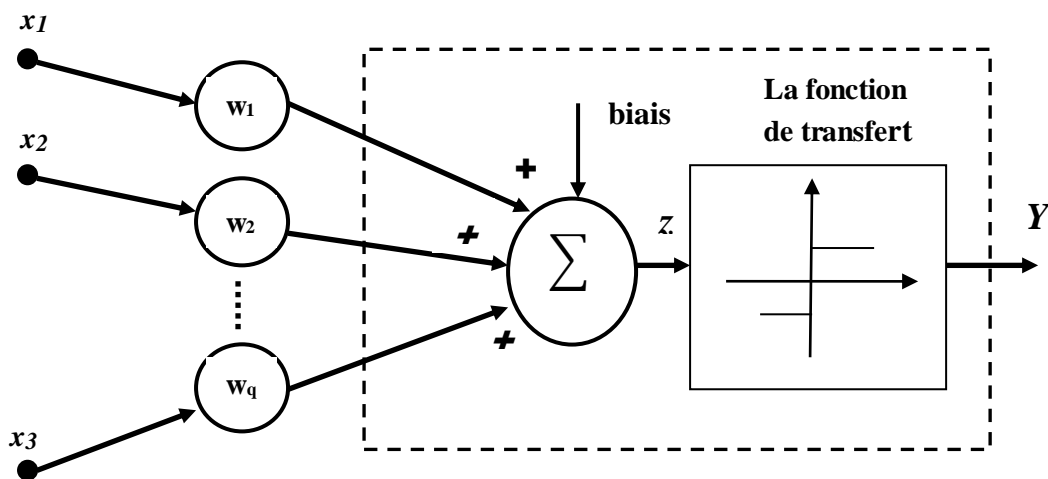


Figure II.3 : Neurone formel.



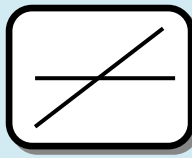
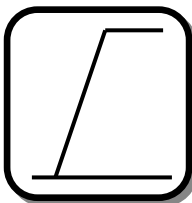
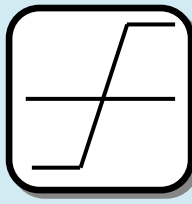

4.4. Fonction d'activation


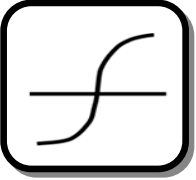
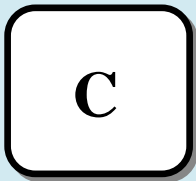
La fonction d'activation (ou fonction de seuillage, ou encore fonction de transfert) sert à introduire une non-linéarité dans le fonctionnement du neurone. Les fonctions de seuillage présentent généralement trois intervalles :

- ✓ En dessous du seuil, le neurone est non actif (souvent dans ce cas, sa sortie vaut 0 ou -1).
- ✓ Aux alentours du seuil, une phase de transition.

- ✓ Au-dessus du seuil, le neurone est actif (souvent dans ce cas, sa sortie vaut 1) dans sa première version, le neurone formel était implémenté avec une fonction à seuil, mais de nombreuses versions existent [27]. Ainsi, le neurone de McCulloch et Pitts a été généralisé de différentes manières, en choisissant d'autres fonctions d'activations, comme les fonctions énumérées dans le (Tableau II.1).

Tableau II.1 : Différentes fonctions d'activations utilisées dans les RNA [22].

Nom de la fonction	Relation entrée/sortie	graphe	Nom MATALB
Seuil	$y=0$ si $s < 0$ $y=1$ si $s \geq 0$		hardlim
Seuil symétrique	$y=-1$ si $s < 0$ $y=1$ si $s \geq 0$		hardlims
Linéaire	$y=s$		purelin
Linéaire saturée	$y=0$ si $s \leq 0$ $y=s$ si $0 \leq s \leq 1$ $y=1$ si $s \geq 1$		satlin
Linéaire saturée symétrique	$y=-1$ si $s < -1$ $y=s$ si $-1 \leq s \leq 1$ $y=1$ si $s > 1$		satlins
Linéaire positive	$y=0$ si $s \leq 0$ $y=s$ si $s \geq 0$		poslin

Sigmoïde	$y = \frac{1}{1+exp^{-s}}$		logsig
Tangente hyperbolique	$y = \frac{e^s - e^{-s}}{e^s + e^{-s}}$		tansig
Compétitive	$y=1$ si s maximum $y=0$ autrement		compet

5. Architecture des réseaux de neurones

L'architecture d'un réseau de neurones est l'organisation des neurones entre eux au sein d'un même réseau. Autrement dit, il s'agit de la façon dont ils ordonnent et connectés. La majorité des réseaux de neurones utilise le même type de neurones. Quelques architectures plus rares se basent sur des neurones dédiés. L'architecture d'un réseau de neurones dépend de la tâche à apprendre. Un réseau de neurones est en général composé de plusieurs couches de neurones, des entrées jusqu'aux sorties. On distingue deux grands types d'architectures de réseaux de neurones :

- ✚ Les réseaux de neurones non bouclés.
- ✚ Les réseaux de neurones bouclés [22].

5.1. Les réseaux de neurones non bouclés

Un réseau de neurones non bouclé réalise une (ou plusieurs) fonctions algébriques de ses entrées, par composition des fonctions réalisées par chacun de ses neurones. Un réseau de neurones non bouclé est représenté graphiquement par un ensemble de neurones "connectés" entre eux, l'information circulant des entrées vers les sorties sans « retour en arrière » ; si l'on représente le réseau comme un graphe dont les nœuds sont les neurones et les arêtes des « connexions » entre ceux-ci, le graphe d'un réseau non bouclé est acyclique. Le terme de "connexions" est une métaphore : dans la très grande

majorité des applications, les réseaux de neurones sont des formules algébriques dont les valeurs numériques sont calculées par des programmes d'ordinateurs, non des objets physiques (circuits électroniques spécialisés) ; néanmoins, le terme de connexion, issu des origines biologiques des réseaux de neurones, est passé dans l'usage, car il est commode quoique trompeur. Il a même donné naissance au terme du connexionnisme [22].

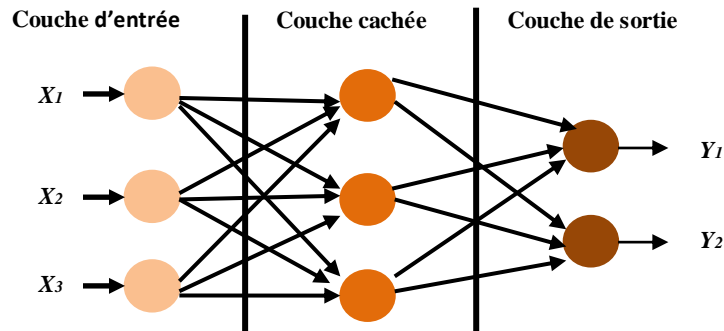


Figure II.4 : Les réseaux de neurones non bouclés.

5.1.1. Réseaux de neurones monocouches

La structure d'un réseau monocouche est telle que des neurones organisés en entrée soient entièrement connectés à d'autres neurones organisés en sortie par une couche modifiable de poids (figure II.5) [22].

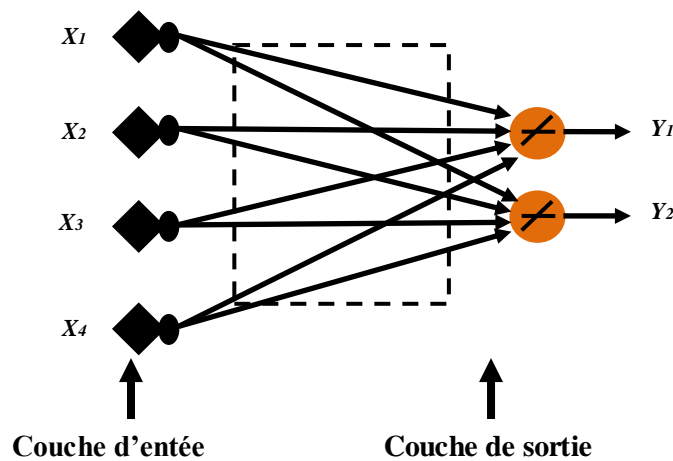


Figure II.5 : Schéma d'un réseau de neurones monocouche.

5.1.2. Réseaux de neurones multicouches

Les neurones sont arrangés par couches. Il n'y a pas de connexion entre neurones d'une même couche, et les connexions ne se font qu'avec les neurones de couches

avalent. Habituellement, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante et celle-ci seulement. Ceci nous permet d'introduire la notion de sens de parcours de l'information (de l'activation) au sein d'un réseau et donc définir les concepts de neurones d'entrée, neurone de sortie. Par extension, on appelle couche d'entrée l'ensemble des neurones d'entrée, couche de sortie l'ensemble des neurones de sortie. Les couches intermédiaires n'ayant aucun contact avec l'extérieur sont appelées couches cachées. La **Figure II.6** représente un réseau de neurones non bouclé qui a une structure particulière, très fréquemment utilisée : il comprend des entrées, deux couches de neurones cachés et des neurones de sortie. Les neurones de la couche cachée ne sont pas connectés entre eux. Cette structure est appelée Perceptron multicouche.

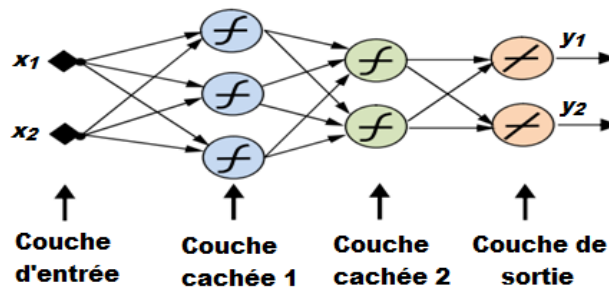


Figure II.6 : Schéma d'un réseau de neurones Perceptron multicouches.

On note aussi que les réseaux multicouches sont beaucoup plus puissants que les réseaux simples à une seule couche. En utilisant deux couches (une couche cachée et une couche de sortie), à condition d'employer une fonction d'activation sigmoïde sur la couche cachée, on peut entraîner un réseau à produire une approximation de la plupart des fonctions, avec une précision arbitraire (cela peut cependant requérir un grand nombre de neurones sur la couche cachée). Saufs dans des rares cas, les réseaux de neurones artificiels exploitent deux ou trois couches [22].

5.1.3. Réseaux de neurones à connexions locales

Il s'agit d'une structure multicouche, mais qui à l'image de la rétine conserve une certaine topologie. Chaque neurone entretient des relations avec un nombre réduit et localisé de neurones de la couche avalent. Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique (**Figure II.7**).

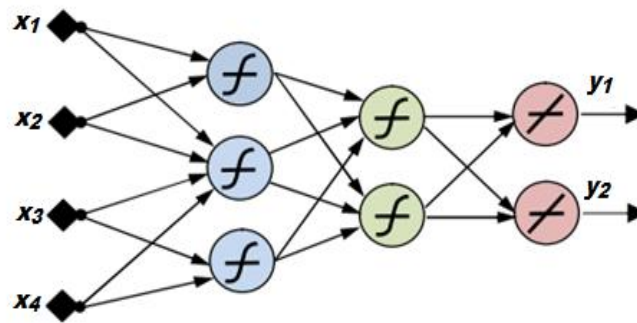


Figure II.7 : Schéma d'un réseau de neurones à connexions locales.

5.2. Les réseaux de neurones bouclés

Contrairement aux réseaux de neurones non bouclés dont le graphe de connexions est acyclique, les réseaux de neurones bouclés peuvent avoir une topologie de connexions quelconque, comprenant notamment des boucles qui ramènent aux entrées la valeur d'une ou plusieurs sorties. Pour qu'un tel système soit causal, il faut évidemment qu'à toute boucle soit associé un retard : 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 d'ordinateurs, on se place dans le cadre des systèmes à temps discret, où les équations différentielles sont remplacées par des équations aux différences. Il s'agit donc des réseaux de neurones avec retour en arrière (feedback network or recurrent network), (Figure II.8).

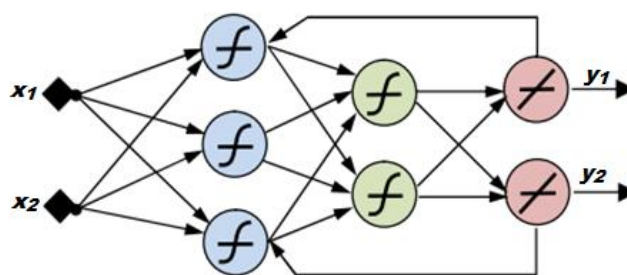


Figure II.8 : Schéma de réseau de neurones bouclé.

Un réseau de neurones bouclé à temps discret est donc régi par une (ou plusieurs) équations aux différences non linéaires, résultant de la composition des fonctions réalisées par chacun des neurones et des retards associés à chacune des connexions. La forme la plus générale des équations régissant un réseau de neurones bouclé est appelée **forme canonique** :

$$x(k+1) = \varphi[x(k), u(k)]; \quad (2)$$

$$y(k) = \psi[x(k), u(k)]; \quad (3)$$

Où φ et ψ sont des fonctions non linéaires réalisées par un réseau de neurones non bouclé (mais pas obligatoirement, un perceptron multicouche), et où k désigne le temps (discret). La forme canonique est représentée sur la **Figure II.9**. Tout réseau de neurones, aussi compliqué soit-il, peut être mis sous cette forme canonique, de manière complètement automatique [22].

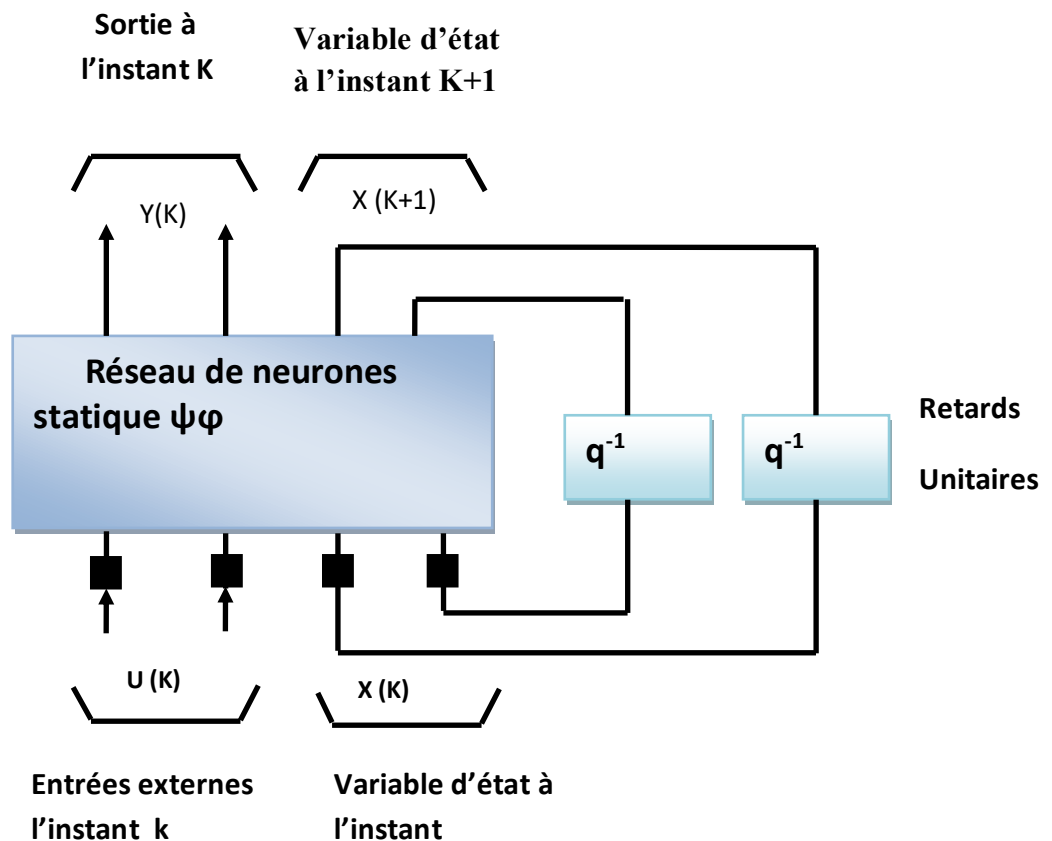


Figure II.9 : Forme canonique d'un réseau de neurones bouclé.

6. Apprentissage

Si le neurone est considéré comme étant la pierre angulaire du réseau de neurones artificiels, l'apprentissage peut être défini comme le processus le plus important pour la construction d'une structure performante du réseau de neurones en se déroulant sur l'adaptation des poids.

Pour cela, il existe deux méthodes d'apprentissage : l'apprentissage supervisé et non supervisé. Les deux méthodes précédentes, fournissent au réseau neuronal une base d'apprentissage qui représente les données sur lesquelles le réseau peut apprendre et réaliser la phase d'apprentissage [28].

6.1. Apprentissage non supervisé

Que signifie le fait de former un réseau de neurones sans supervision ? Comme indiqué précédemment, le réseau de neurones est muni d'une base d'apprentissage, qui est une collection de valeurs d'entrée bien définies. Le réseau de neurones non supervisé n'est pas muni de données attendues ou sorties anticipées. L'apprentissage non supervisé est généralement utilisé pour entraîner les réseaux de neurones utilisés pour la classification et le data mining [28].

La figure suivante résume la méthode d'apprentissage non supervisé.

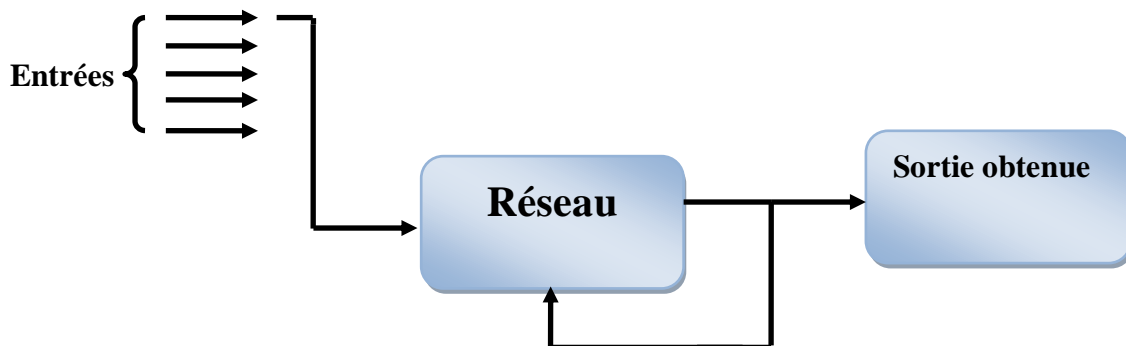


Figure II.10 : Apprentissage non supervisé.

6.2. Apprentissage supervisé

La méthode d'apprentissage supervisée est similaire à la méthode d'apprentissage non supervisé, dans laquelle les ensembles d'apprentissage sont fournis. Tout comme avec l'apprentissage non supervisé, les ensembles d'apprentissage spécifient les signaux d'entrée pour le réseau neuronal. La principale différence est que dans l'apprentissage supervisé, les résultats attendus sont fournis.

L'apprentissage supervisé est basé sur la comparaison directe entre l'entrée actuelle du RNA et la sortie désirée correcte, aussi connue comme la sortie cible. Il est souvent formulé comme la minimisation d'une fonction d'erreur comme l'erreur quadratique

moyenne totale entre l'entrée actuelle et la sortie désirée additionner à toutes les données disponibles. Une descente de gradient, basée sur l'algorithme d'optimisation telle que la rétro-propagation, peut alors être utilisée pour ajuster itérativement les connexions des poids dans le RNA afin de minimiser l'erreur [28].

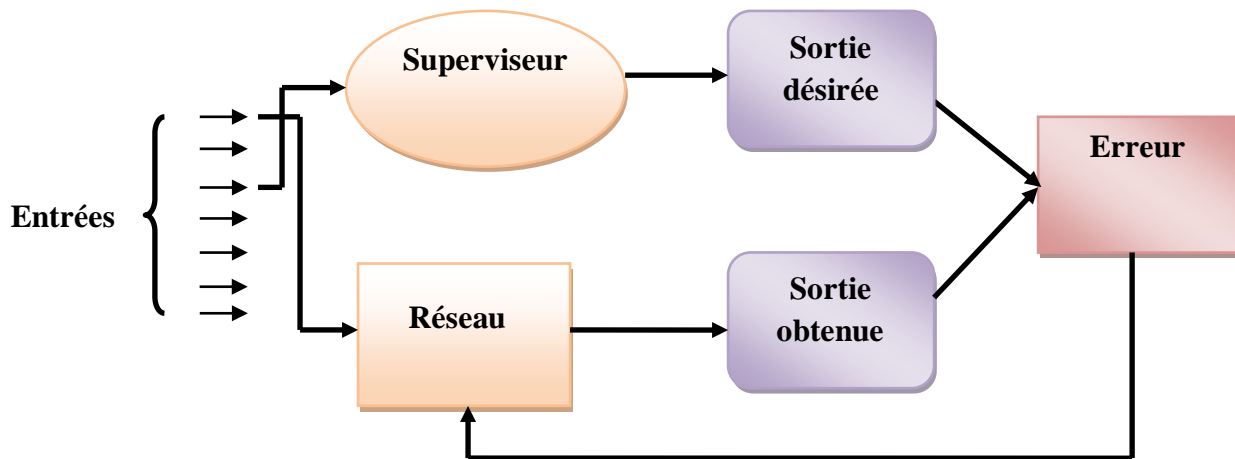


Figure II.11 : Apprentissage supervisé.

6.3. Règles d'apprentissage

L'apprentissage est une procédure adaptative par laquelle les connexions (ou synapses) des neurones sont ajustés face à une source d'information. Il existe au moins 23 types de règles d'apprentissage qui peuvent être regroupés en trois catégories : les règles d'apprentissage supervisé, non supervisé, sont renforcées. Mais l'objectif fondamental de l'apprentissage reste le même : soit la classification, l'approximation de fonction ou encore la prévision.

Dans tous les cas, la procédure adaptative de mise à jour des poids obéit essentiellement à un objectif : la recherche analytique du minimum de la fonction de coût (ou fonction d'erreur) dans un espace multidimensionnel. Il existe deux techniques de recherche du minimum de la fonction d'erreur : la première est une technique de recherche du minimal local basé sur la méthode du gradient décroissant ; la seconde est une recherche du minimum global fondée sur des approches statistiques ou génétiques [28].

7. Applications des réseaux de neurones artificiels

Les RNA sont considérés comme une nouvelle approche de traitement de l'information par apprentissage de cette information et la rend disponible à l'utilisation afin de résoudre un tel problème. Il existe de nombreux problèmes qui peuvent être résolus par un réseau de neurones. Toutefois, les réseaux de neurones sont généralement utilisés pour traiter des types particuliers de problèmes. Les quatre types de problèmes qui sont souvent résolus par les réseaux de neurones sont les suivants :

7.1. Classification

La classification est le processus de classement des entrées en groupes. Par exemple, une compagnie d'assurance peut vouloir classer les demandes d'assurance dans les différentes catégories de risques, ou une organisation en ligne peut vouloir de son système de messagerie de classer le courrier entrant dans des groupes de messages spam et non-spam [28].

7.2. Prédiction

La prédiction est une autre application des réseaux neurones. Étant donné une série temporelle de données d'entrée, un réseau de neurones peut prédire les valeurs futures. La précision de la prédiction dépend de nombreux facteurs, tels que la quantité et la pertinence des données d'entrée. Par exemple, les réseaux de neurones sont généralement appliqués à des problèmes de prédiction de l'évolution des marchés financiers [28].

7.3. Reconnaissance des formes (pattern recognition)

Reconnaissance des formes est l'une des utilisations les plus courantes des réseaux neuraux. Pattern recognition est une forme de classification et est tout simplement la capacité de reconnaître un motif. Le modèle doit être reconnu même s'il n'est pas clair. Exemple : la reconnaissance des visages [28].

7.4. Optimisation

Une autre application des réseaux de neurones est l'optimisation qui peut être appliquée à de nombreux problèmes pour lesquels une solution est recherchée. Le réseau de neurones peut ne pas toujours trouver la solution optimale mais il cherche à trouver une solution acceptable.

Les RNA sont utilisés dans le domaine des sciences cognitives où l'on cherche à développer des modèles capables de manifester des capacités d'apprentissage et d'adaptation à leur environnement (Blayo et Verleysen 1996). Les deux concepts de base à l'origine des différents types de modèles neuronaux sont l'architecture et l'apprentissage [28].

8. Conclusion

Les réseaux de neurones sont considérés comme des systèmes imitant le fonctionnement du cerveau humain. Dans ce chapitre, nous avons vu la différence entre le neurone biologique et neurone formel ainsi que les différentes fonctions d'activation. Comme les RNA se caractérisent selon le domaine d'utilisation et des problèmes qu'ils peuvent résoudre. Ils se caractérisent aussi par leurs différentes architectures, traitées dans ce chapitre.



CHAPITRE III

L'ALGORITHME D'OPTIMISATION A BASE DE BIOGEOGRAPHIE

CHAPITRE III

L'ALGORITHME D'OPTIMISATION A BASE DE BIOGEOGRAPHIE

Résumé

Dans ce mémoire de fin d'étude on va utiliser les réseaux de neurones artificiels et l'algorithme d'optimisation à base de biogéographie pour la modélisation des séries temporelles. Dans le deuxième chapitre on a exposé les notions liées aux réseaux de neurones artificiels. Nous exposons de ce troisième chapitre les notions liées aux algorithmes optimisations.

Sommaire

1. Introduction.....	37
2. Problème d'optimisation	37
3. Catégories d'algorithmes d'optimisations	38
4. Conclusion.....	50

1. Introduction

Un algorithme d'optimisation c'est une technique visant à résoudre des problèmes d'optimisation difficile (souvent issus des domaines de la recherche opérationnelle, de l'ingénierie ou de l'intelligence artificielle). Ces techniques sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum global, c'est-à-dire l'extremum global d'une fonction. Elles se comportent comme des algorithmes de recherche tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution. Il existe un grand nombre d'algorithmes d'optimisations, allant de la simple recherche locale à des algorithmes complexes de recherche globale. Ces techniques sont souvent inspirées par des mécanismes biologiques ou biogéographiques. L'analogie entre un problème d'optimisation et ces phénomènes biologiques et biogéographiques a été formalisée par plusieurs approches, à commencer par les algorithmes génétiques.

2. Problème d'optimisation

L'optimisation est un aspect fondamental de l'ingénierie et de la résolution de problèmes. L'objectif de l'optimisation est de chercher les valeurs d'un ensemble de paramètres pour maximiser ou minimiser les fonctions objectives soumises à certaines contraintes [29]. Un choix de valeurs, pour l'ensemble des paramètres, qui satisfont toutes les contraintes, est appelé une solution faisable. Les solutions faisables avec des valeurs de la fonction objective qui sont meilleures que les valeurs de toutes les autres solutions possibles, sont appelées les solutions optimales.

Un problème d'optimisation peut être formulé sous la forme d'un problème de minimisation ou d'un problème de maximisation. Parfois, nous essayons de minimiser une fonction et parfois, nous essayons de maximiser une fonction. Ces deux problèmes sont facilement convertis l'un à l'autre :

$$\min_x f(x) \leftrightarrow \max_x [-f(x)]; \quad (1)$$

$$\max_x f(x) \leftrightarrow \min_x [-f(x)]; \quad (2)$$

La fonction $f(x)$ est appelée la fonction objective, et le vecteur x est appelé la variable indépendante, ou la variable de décision. Notons, qu'en fonction du contexte, les termes « variable indépendante » et « variable de décision » se réfèrent parfois à

l'ensemble du vecteur x , et se réfèrent parfois à des éléments spécifiques dans le vecteur x . Les éléments de x sont aussi appelés les attributs de la solution. Le nombre des éléments de x est appelé la dimension du problème. Comme la montre les équations (1) et (2), tout algorithme conçu pour minimiser une fonction peut facilement être utilisé pour maximiser une fonction, et tout algorithme conçu pour maximiser une fonction peut être facilement utilisé pour minimiser une fonction :

- ✓ Quand nous essayons de minimiser une fonction, nous appelons la valeur de la fonction : *la fonction de coût*.
- ✓ Quand nous essayons de maximiser une fonction, nous appelons la valeur de la fonction : *la fitness*.

3. Catégories d'algorithmes d'optimisations

On distingue deux catégories d'algorithmes d'optimisation (Figure III.1) :

- *Algorithme Evolutionnaire.*
- *Algorithme Méta-heuristiques.*

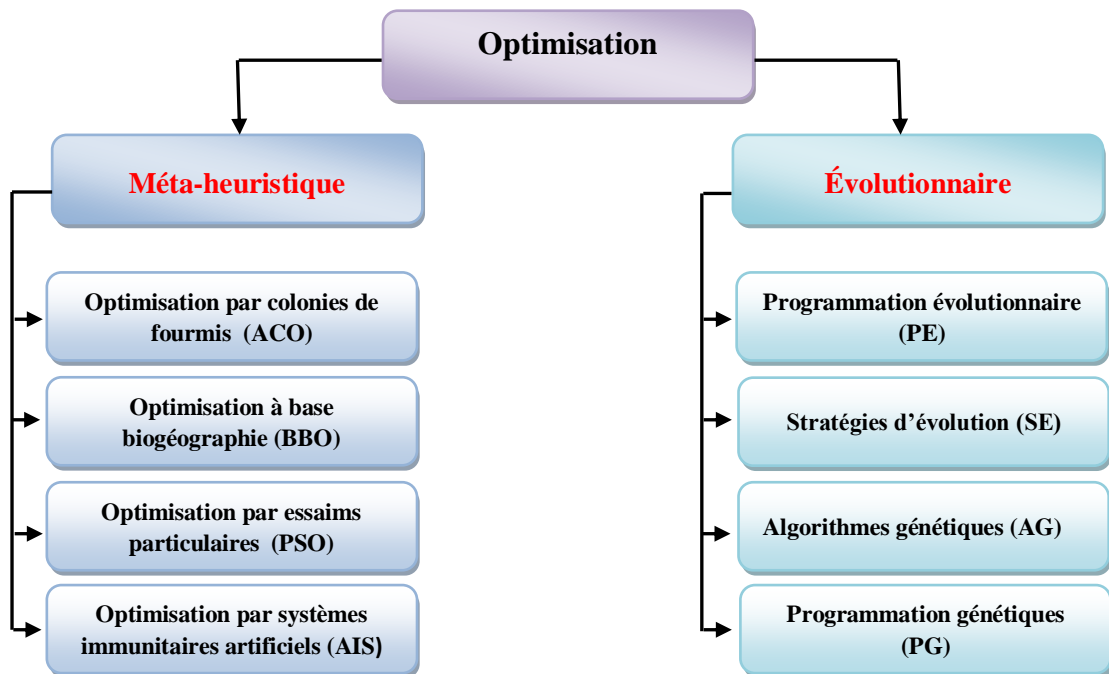


Figure III.1 : Catégories d'algorithmes d'optimisations.

3.1. Algorithme Evolutionnaire

3.1.1. Définition

L'appellation Algorithmes Evolutionnaires (en anglais *Evolutionary algorithms* (*EA*)) est un terme générique pour décrire les approches informatiques reposant sur les principes de l'évolution comme éléments-clefs de leurs fonctionnements. Ils sont inspirés du concept de sélection naturelle élaboré par Charles Darwin [30]. Il existe un grand nombre d'algorithmes évolutionnaires, toutes ces approches ont en commun la simulation de l'évolution d'une population d'individus divers (généralement tirée aléatoirement au départ) à laquelle on applique différents opérateurs (recombinaisons, mutations...) et que l'on soumet à une sélection, à chaque génération [31].

3.1.2. Différentes classes d'Algorithmes Evolutionnaires

Les Algorithmes Evolutionnaires comptent quatre principaux algorithmes :

3.1.2.1 La Programmation Evolutionnaire (PE)

Proposée par L.J. Fogel dans les années 60 également, et reprise par son fils D.B. Fogel dans les années 90, en Californie, USA. La Programmation Evolutionnaire est une des classes principales des Algorithmes Evolutionnaires. Elle se base sur le concept darwinien de l'évolution pour générer itérativement des solutions plus adaptées à leur environnement [32].

Bien que la Programmation Evolutionnaire ait été développée comme une méthode d'intelligence artificielle, son concept était loin des premières recherches en intelligence artificielle qui se basent sur la recherche des heuristiques simples. Au lieu de développer un ensemble complexe de lois de comportements dérivés de l'expertise humaine, la Programmation Evolutionnaire fait évoluer un ensemble d'individus faisant preuve d'un comportement optimal dans un environnement, statique ou dynamique, représenté par une fonction objective [32].

Dans sa forme de base, la Programmation Evolutionnaire utilise les principales procédures d'un Algorithme Evolutionnaire, soit : l'initialisation et la reproduction réduite à la mutation seule, la mutation fournit un moyen d'exploration de l'espace de recherche prévoyant toute convergence vers un optimum local. La fonction d'évaluation mesure la qualité de chaque individu. Enfin, la procédure de sélection favorise la survie des meilleures solutions suivant leurs adaptations [32].

3.1.2.2. Les Stratégies d'Evolution (SE)

Elles ont été développées par I. Rechenberg et H.P. Schwefel, 1965 à Berlin. Les Stratégies d'Evolution constituent une autre classe d'Algorithmes Evolutionnaires. À l'inverse des autres approches, les Stratégies d'Evolution ont été conçues, à l'origine, comme des méthodes d'optimisation numériques. Leurs premières applications étaient l'optimisation d'un corps dans un tunnel de vent [33].

Abordant les problèmes d'optimisation continue, les Stratégies d'Evolution sont très similaires à la Programmation Evolutionnaire. Ils partagent la même philosophie : le problème d'optimisation est considéré dans sa globalité et aucun partitionnement n'est effectué, ils n'obéissent pas à la théorie des schèmes. Les deux méthodes mettent l'accent sur la mutation comme principal opérateur de recherche et utilisent une mutation auto-adaptative [33].

La seule différence notable avec la Programmation Evolutionnaire est l'utilisation du croisement par les Stratégies d'Evolution. Le croisement est utilisé comme second opérateur de recherche, son rôle n'était pas la manipulation des blocs élémentaires, mais d'assurer plus de diversité dans la population.

3.1.2.3. Algorithmes Génétiques (AG)

Les Algorithmes Génétiques ont été développés par John Holland en 1975. Les Algorithmes Génétiques forment une des principales classes des Algorithmes Evolutionnaires, proposés et développés par Hollande. Ils se basent sur les théories d'évolution naturelle modernes et utilisent une combinaison de reproduction : croisement et mutation et de sélection, pour générer des individus de plus en plus adaptés à leur environnement, donc des solutions optimales [34].

Les Algorithmes Génétiques se basent sur la théorie des schèmes motivés par la conviction que les problèmes complexes sont plus faciles à traiter en les divisant en un ensemble de sous problèmes plus simples. Dans cette théorie une solution est représentée comme un ensemble de gènes (variables) composés d'allèles. Un schème peut être considéré comme une solution partielle dans laquelle seules quelques variables sont définies. Les schèmes fournissent ainsi un moyen de décomposition du problème de recherche en une hiérarchie de sous problèmes plus simples. Le cas le plus simple est d'assigner une valeur à un seul allèle [34].

3.1.2.4. La Programmation Génétique (PG)

Proposée par J. Koza en 1988 qui était auparavant un sous-groupe des AG. La spécificité de la PG est son espace de recherche, un espace de programmes le plus souvent représenté sous forme d'arbres. Cette technique cherche à atteindre un des vieux rêves des programmeurs : « *écrire le programme qui écrit le programme* » [35].

3.2. Algorithme Méta-heuristiques

3.2.1 Définition

Le mot méta-heuristique est un mot Grecque à l'origine composé en deux :

- **Heuristique** : trouver ;
- **Méta** : au-delà (dans un niveau supérieur).

✚ *Heuristique* :

Un algorithme heuristique permet de trouver au moins une solution réalisable dans un temps de calcul acceptable à un problème d'optimisation, mais sans garantir que cette solution soit optimale.

✚ *Méta-heuristique* :

Les méta-heuristiques forment une famille d'algorithmes d'optimisation utilisées pour résoudre approximativement des problèmes d'optimisation difficiles, dont on ne connaît pas de méthode classique plus efficace. Elles sont généralement utilisées comme des méthodes génériques pouvant traiter un grand nombre de problèmes différents, sans nécessiter de changements profonds dans l'algorithme employé.

Les méta-heuristiques sont classées en deux ensembles : les algorithmes à base de solution courante unique (recherche locale) et les méthodes à population (algorithmes évolutifs) [36].

3.2.2. Les Différentes classes d'Algorithmes Méta-heuristiques

3.2.2.1 Optimisation par colonies de fourmis (ACO)

L'optimisation par colonies de fourmis, (*Ant Colony Optimization (ACO)*), conçue par Dorigo s'inspire comme son nom l'indique du comportement des fourmis lorsque celles-ci cherchent de la nourriture et optimisent le chemin entre leur nid et la nourriture trouvée. En effet, les fourmis utilisent leur environnement pour communiquer entre elles, il s'agit d'un mécanisme dît stigmergique grâce auquel elles déposent des

phéromones sur le sol pour signifier aux autres fourmis le chemin qu'elles ont parcouru pour atteindre la nourriture. Ainsi, les autres pourront suivre la piste de phéromones pour retrouver la source de nourriture [37].

3.2.2.2. Optimisation par essais particulaires (PSO)

La méthode des essais particulaires, (*Particle Swarm Optimization (PSO)*), a été conçue en 1995. Le principe de la méthode provient de l'observation des comportements collectifs d'animaux, tels que le déplacement en vol des oiseaux, des insectes, ou des bancs de poissons, elle s'inspire des modélisations statistiques développées par Reynolds, Heppner et Grenander. Les solutions sont ici appelées particules, elles représentent des solutions potentielles qui parcourent l'espace de recherche, la population est donc vue comme un essaim particulaire. Concrètement chaque particule possède une position X_i , une vitesse V_i . Le but va donc être de déplacer les particules vers la meilleure solution possible, pour cela le déplacement de chaque particule va prendre en compte sa position courante vers la meilleure solution qu'elle a trouvée depuis le début de la recherche (notée P_i), et enfin elle va aussi tenir compte de la meilleure solution trouvée par l'essaim entier (notée P_g). Chacun de ces trois aspects est respectivement appelé des composantes physiques, cognitive et sociale [37].

3.2.2.3. Optimisation par systèmes immunitaires artificiels (AIS)

L'optimisation par systèmes immunitaires artificiels, (*Artificial Immune Systems (AIS)*), est née dans les années 1980 grâce au travail de Fermer, Packard et Perelson. L'AIS mime le fonctionnement du système immunitaire des êtres humains. En effet, ce dernier a pour but de protéger le corps d'agents pathogènes extérieur comme les bactéries ou les virus [37].

3.2.2.4. L'algorithme d'optimisation à base de biogéographie (BBO)

L'algorithme d'optimisation à base de biogéographie (*BBO : Biogeography-Based Optimization*) est un algorithme stochastique d'optimisation motivée par les mécanismes de la migration des écosystèmes. Elle est inspirée des modèles mathématiques de la biogéographie et elle est initialement développée par D. Simon [38]. Cet algorithme appartient à une famille des algorithmes évolutionnaires dont le but est d'obtenir une solution optimale à un problème d'optimisation par un processus évolutionnaire.

A. Biogéographie

La Biogéographie est l'étude de la distribution de la biodiversité dans l'espace et dans le temps, ce qui permet aux nombreuses espèces animales de migrer vers différents habitats ou îles pour leur survie et une meilleure vie. La biogéographie a été étudiée dès le 19^{me} siècle par Alfred Wallace et Charles Darwin.

Dans la science de la biogéographie, une île est définie comme la zone écologique habitée par des plantes particulières ou d'espèces animales et géographiques ment isolée d'autres habitats. Chaque île a ses caractéristiques telles que la disponibilité alimentaire, les précipitations, la température, la diversité des espèces, la sécurité, etc. (Figure III.2). La qualité d'une île est mesurée par son indice d'adéquation (*Suitability Index*). Les îles avec un indice élevé sont plus adaptées à la vie et ont donc une grande population [39].

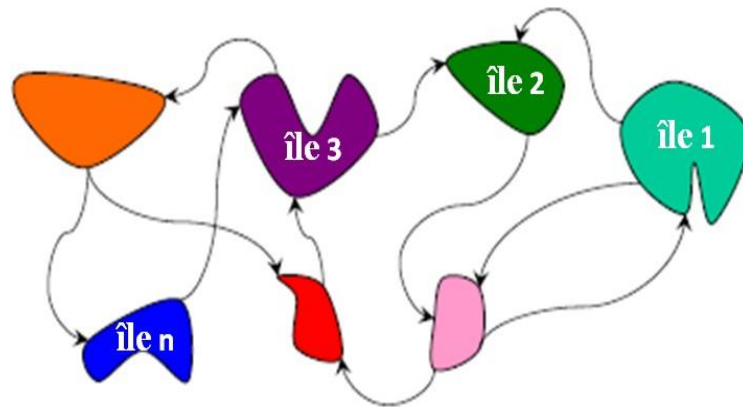


Figure III. 2 : Migration des espèces [38].

B. Principe de l'algorithme BBO

BBO, est un algorithme basé sur une population d'individus appelés île (ou habitats). Chaque île représente une solution possible au problème à résoudre. La « fitness » de chaque île est déterminée par son **HSI** (*Habitat Suitability Index*) qui est une mesure de la qualité d'une solution candidate, chaque île est représentée par des variables d'indice de qualité **SIVs** (*Suitability Index Variables*). Un HSI élevé d'une île signifie une bonne performance sur le problème d'optimisation, et un faible HSI signifie une mauvaise performance pour un problème d'optimisation [40].

Le fonctionnement de **BBO** est basé sur la migration et la mutation. La population initiale représente l'espace de recherche, elle est générée aléatoirement. L'évaluation de la population initiale engendre la migration de certains individus et les descendants vont

être mutés. La migration permet de créer un nouvel ensemble d'individus et la mutation permet de fixer la proportion de la population qui sera renouvelée à chaque génération. Les meilleurs individus trouvés sont conservés par l'élitisme (sélection). Les nouveaux des cédants remplacent les parents afin de former une nouvelle population.

C. Optimisation et biogéographie

L'application de la biogéographie dans l'optimisation est un exemple d'un modèle de processus naturel pour résoudre les problèmes d'optimisation. Ceci est similaire à ce qui s'est passé dans les dernières décennies avec les algorithmes génétiques, réseaux de neurones, les colonies de fourmis et d'autres domaines de l'intelligence informatique [41]. L'optimisation par la biogéographie (*BBO*) utilise un vocabulaire similaire à celui de la biogéographie (**Tableau III.1**) où chaque habitat est analogue à une solution du problème. Les caractéristiques des solutions (variables de décision) sont appelées variables d'indice d'adéquation (*SIV*). L'indice d'adéquation de l'habitat (*HSI*) est analogue au fitness qui permet de mesurer l'adéquation de l'individu. Les habitats avec un HSI élevé ont tendance à avoir un grand nombre d'espèces, un taux d'immigration faible et un taux d'émigration élevé. Inversement, les habitats avec un HSI faible ont un faible nombre d'espèces, un taux d'immigration élevé et un taux d'émigration faible [38]. L'algorithme BBO nécessite l'emploi de certains termes dont il est utile de préciser la définition.

Tableau III.1 : Terminologie de la BBO.

La Biogéographie	Algorithme BBO
Habitat ou île	Solution du problème
HSI	Qualité de la solution (fonction coût)
SIV	Les Variables du problème d'optimisation

✓ *Variable d'indice d'adéquation*

Une variable d'indice d'adéquation (*SIV: Suitability Index Variable*) est une variable entière, réelle ou booléenne qui caractérise l'habitabilité d'une île [38].

✓ **Habitat**

Un habitat H est une solution du problème représentée généralement par un vecteur initialisé aléatoirement par des variables d'indice d'adéquation. Il est noté [38] :

$$H = [SIV_1, SIV_2, \dots, SIV_k]. \quad (3)$$

✓ **Population** : une population est un ensemble de n habitats [38].

$$Pop = [H_1, H_2, \dots, H_n]. \quad (4)$$

✓ **Indice d'adéquation de l'habitat**

Un indice d'adéquation de l'habitat (**HSI**) équivalant au fitness, associe une valeur pour chaque individu. Cette valeur a pour but d'évaluer le degré d'adaptation d'un individu à son environnement [38].

$$HSI = f(H) = f([SIV_1, SIV_2, \dots, SIV_k]). \quad (5)$$

✓ **Taux d'immigration**

Le taux d'immigration $\lambda(H_i)$ est le taux d'entrée des variables (**SIV**) à un habitat [38].

✓ **Taux d'émigration**

Le taux d'émigration $\mu(H_i)$ est le taux de sortie des variables (**SIV**) d'un habitat [38].

D. Étapes de l'optimisation basée sur la biogéographie

L'organigramme fonctionnel présenté dans la **Figure III.3**, illustre la structure générale de l'algorithme **BBO**. L'algorithme commence par initialiser les paramètres et la population initiale, il modifie cette population par des opérateurs spécifiques en construisant de nouvelle population jusqu'à atteindre une qualité (**HSI**) meilleure qu'un seuil préfixé ou un nombre maximal de générations g_{max} .

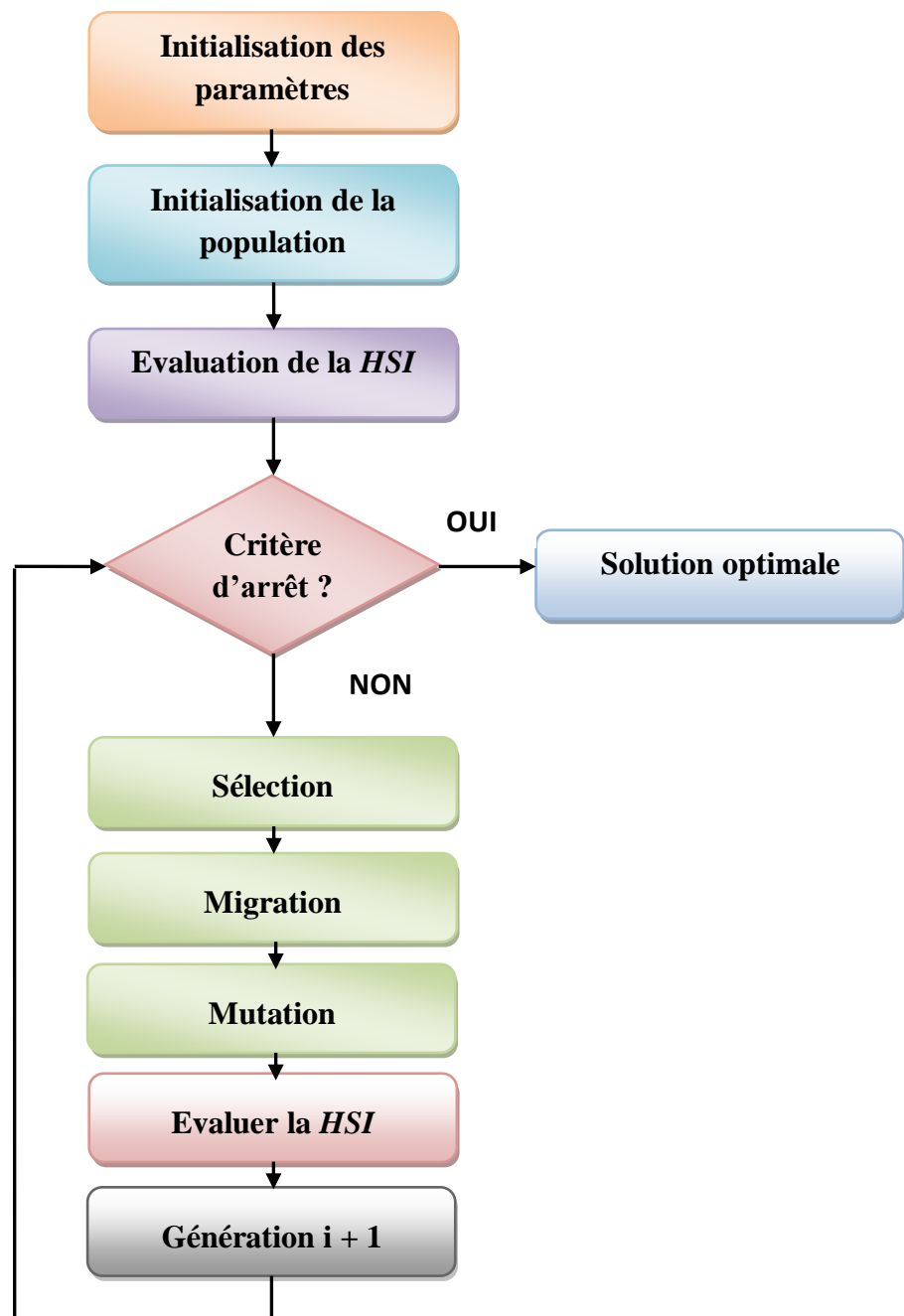


Figure III.3 : Organigramme général de l'algorithme *BBO*.

1. Initialisation

Dans cette étape, nous définissons les paramètres de contrôle (Tableau III.2) et nous générons aléatoirement une population initiale de n habitats. La taille de cette population reste constante tout au long de l'algorithme [42].

Tableau III.2 Paramètres de la *BBO*.

Paramètres	Notation
Taille de la population	N_p
Probabilité de mutation	P_m
Taille de la mémoire élite	n_{elit}
Taux d'immigration maximal	I
Taux d'émigration maximal	E
Nombre maximum de générations	g_{max}

2. Évaluation de la fonction *HSI*

Pour évaluer la pertinence d'une solution par rapport à une autre, nous calculons la valeur de la *HSI* correspondante à chaque solution candidate [38].

3. Sélection

Pour appliquer les opérateurs de l'algorithme BBO comme n'importe quel autre algorithme d'optimisation, nous devons sélectionner les habitats ou individus candidats à cet opérateur. La sélection est fondée sur la qualité des habitats, estimée à l'aide de fonction d'adaptation. Il existe plusieurs méthodes de sélection [42].

✚ Sélection par roulette

La population est représentée comme une roue de roulette, où chaque habitat est représenté par une portion qui correspond proportionnellement à sa valeur de *HSI* (fitness). La sélection d'un individu se fait en tournant la roue. L'un des inconvénients de ce type de sélection est de choisir presque toujours le même habitat s'il en existe un bien meilleur que tous les autres, ce qui cause une perte de diversité dans la population [43].

✚ Sélection par rang

La sélection par rang trie d'abord la population par *HSI*. Ensuite, chaque habitat se voit associé un rang en fonction de sa position. Ainsi, le plus mauvais habitat aura le rang 1, le suivant 2, et ainsi de suite jusqu'au meilleur habitat qui aura le rang n . La sélection par rang d'un habitat est la même que par roulette, mais les proportions sont en relation

avec le rang plutôt qu'avec la valeur de l'évaluation. Avec cette méthode de sélection, tous les habitats ont une chance d'être sélectionnés. Cependant, elle conduit à une convergence plus lente vers la bonne solution. Ceci est dû au fait que les meilleurs habitats ne diffèrent pas énormément des plus mauvais [43].

✚ Sélection par tournoi

La sélection par tournoi est l'une des sélections les plus utilisées dans les algorithmes évolutionnaires. Le principe consiste à choisir un sous-ensemble d'individus (S individus) aléatoirement dans la population, puis à sélectionner le meilleur individu dans ce groupe en fonction de son *HSI*. Ce processus est répété jusqu'à l'obtention du nombre d'individus requis [43]. Le nombre de participants à un tournoi, appelé la taille du tournoi, est utilisé pour faire varier la pression de cette sélection. Si ce nombre est grand, alors la pression sera forte et les faibles individus auront une petite chance d'être choisis. En général, un seul gagnant est choisi parmi les participants à un tournoi.

✚ Élitisme :

À la création d'une nouvelle population, il y a de grandes chances que les meilleurs habitats soient perdus après les opérations de migration et de mutation. Pour éviter cela, nous utilisons la méthode d'élitisme. Elle consiste à copier un ou plusieurs des meilleurs habitats dans la nouvelle génération. Ensuite, nous générons le reste de la population selon le mécanisme de reproduction usuel. L'opérateur de sélection est utilisé pour sélectionner des habitats pour appliquer les deux opérateurs d'évolution : la migration et la mutation [43].

4. Opérateur de migration

La migration est un opérateur probabiliste utilisé pour modifier chaque solution H_i en partageant des caractéristiques parmi les différentes solutions. L'idée de l'opérateur de migration est basée sur la migration en biogéographie, qui représente le mouvement des espèces entre les différents habitats. La probabilité qu'une solution est sélectionnée pour immigrer ou émigrer dépend de son taux d'immigration λ_i ou le taux d'émigration μ_j [41]. Le processus de migration est défini par la relation :

$$H_i(SIV_k) \Leftarrow H_j(SIV_k). \quad (6)$$

L'équation (6) représente comment une caractéristique ou *SIV* d'une solution H_i est remplacée par une caractéristique ou *SIV* d'une solution H_j par l'opération de

migration. Dans la **BBO**, chaque solution H_i a son propre taux d'immigration λ_i et taux d'émigration μ_i .

Ces deux taux sont calculés par l'équation (7) et l'équation (8) respectivement.

$$\lambda_i = I * (1 - \frac{k_i}{n}); \tag{7}$$

$$\mu_i = E * (\frac{k_i}{n}); \tag{8}$$

I et E représentent les taux maximaux possibles d'immigration et d'émigration respectivement. k_i représente le rang du i habitat après le tri de tous les habitats en fonction de leur **HSI**. n représente la taille de la population. La **Figure III.4** illustre deux solutions candidates S_1 et S_2 à un problème en utilisant des courbes d'immigration et d'émigration symétriques ($E = I$). S_1 représente une mauvaise solution et S_2 représente une meilleure solution. La probabilité d'immigration pour S_1 sera donc plus élevée que celle de S_2 tandis que la probabilité d'émigration pour S_1 sera inférieure à la probabilité d'émigration pour S_2 [38].

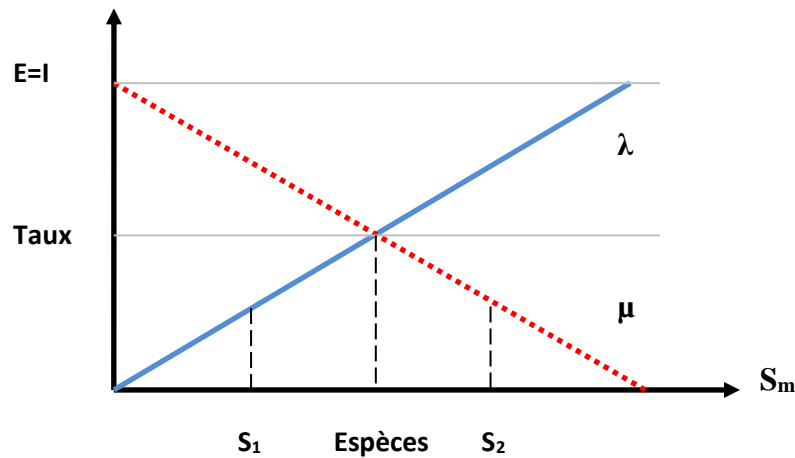


Figure III.4 : Illustration de deux solutions candidates S_1 et S_2 .

5. Opérateur de mutation

En plus de l'opérateur de migration, nous avons l'opérateur de mutation. La mutation est un opérateur probabiliste utilisé pour modifier un ou plusieurs **SIV** d'une solution sélectionnée aléatoirement en se basant sur sa probabilité d'existence P_{ig} pour la génération g . Cet opérateur aide à introduire de nouvelles caractéristiques et augmente la diversité dans la population. La probabilité de mutation m_i est calculée en fonction de la probabilité de la solution exprimée dans l'équation (9).

$$m_i = m_{max} \left(1 - \frac{P_{ig}}{P_{max}} \right); \quad (9)$$

Où :

m_i : le taux de mutation pour l'habitat i .

m_{max} : le taux maximum de mutation.

P_{max} : la probabilité maximale d'existence. Avant d'appliquer l'opérateur de la mutation, nous devons calculer P_{ig} , la probabilité d'existence de l'habitat i dans la génération actuelle g . Ce paramètre est calculé à partir de sa valeur dans la génération précédente par l'équation (10).

$$P_{ig} = P_{ig-1} + P_{ig}; \quad (10)$$

Les valeurs de P_i avant le début de l'algorithme sont initialisées à $1/n$ pour chaque habitat [38]. P_i est le changement de probabilité d'existence de l'habitat i , il est donné par l'équation (11) [38] :

$$P_i = \begin{cases} - (\lambda_i + \mu_i) P_i + \mu_{i+j} P_{i+1} & i = 1 \\ - (\lambda_i + \mu_i) P_i + \lambda_{i-1} P_{i-1} + \mu_{i+j} P_{i+1} & 2 \leq i \leq n. \\ - (\lambda_i + \mu_i) P_i + \lambda_{i+j} P_{i-1} & i = n \end{cases} \quad (11)$$

4. Conclusion

Ce chapitre a été dédié à la présentation des techniques d'optimisation et au juste l'algorithme d'optimisation à base de biogéographie, le fondement théorique de cette méthode a été explicitement présenté dans le but d'ouvrir la voie à l'algorithme de modélisation des séries temporelles qui sera développé dans le chapitre 4.



CHAPITRE IV

MODELISATION DES SERIES TEMPORELLES PAR RNA ET BBO

CHAPITRE IV

MODELISATION DES SERIES TEMPORELLES

PAR RNA ET BBO

Résumé

Dans ce chapitre nous allons discuter une technique de modélisation des séries temporelles à base des réseaux de neurones artificiels et l'algorithme d'optimisation à base de biogéographie. Cette technique intègre la capacité d'apprentissage des réseaux de neurones artificiels avec l'algorithme d'optimisation BBO pour approximer les séries temporelles.

Sommaire

1. Introduction.....	53
2. Méthode proposée pour la modélisation des séries temporelles	54
3. Résultats de simulation	59
4. Conclusion	64

1. Introduction

Les séries temporelles constituent une part importante des données produites et disponibles sur Internet dans de très différents domaines. Par exemple, dans le seul domaine médical, l'information enregistrée par les électroencéphalogrammes ou par des électrocardiogrammes, les données qui représentent l'expression de gènes [45], les données sur la croissance d'un individu..., sont des séries temporelles fréquemment traitées. Les séries temporelles se retrouvent de la même manière dans d'autres domaines tels que la finance, la météorologie, le son... Si depuis plus d'un siècle la communauté scientifique s'est penchée sur le traitement des séries temporelles [46].

Le sujet de la modélisation et de la prédiction de séries temporelles a donné lieu à une riche littérature depuis de nombreuses années en statistique et en apprentissage automatique. En statistiques, les approches linéaires classiques basées sur les modèles à moyenne mobile et autorégressifs ont été les plus utilisées. Ces modèles supposent que les séries temporelles sont stationnaires et qu'elles présentent des dépendances linéaires dans le temps [47]. En apprentissage automatique, des extensions non linéaires de ces modèles, basées essentiellement sur des réseaux de neurones, ont été proposées dès le début des années 1990, ouvrant la voie à de nombreuses extensions non linéaires comme les méthodes à noyaux [48].

Dans ce chapitre on va présenter une technique de modélisation des séries temporelles à base des réseaux de neurones artificiels et l'algorithme d'optimisation à base de biogéographie. Cette technique comprend une interconnexion parallèle de deux sous modèles neuronaux. Le premier sous modèle neuronal est le modèle primaire, qui représente un modèle ordinaire à faible résolution. Pour surmonter le problème de la résolution et obtenir un modèle avec une résolution plus élevée, nous avons introduit un deuxième sous-modèle neuronal appelé le modèle d'erreur, qui représente la modélisation d'erreur entre la sortie du modèle primaire et la série temporelle sous considération. Le modèle d'erreur représente l'incertitude dans le modèle principal, cette incertitude est minimisée par une simple soustraction de la sortie du modèle d'erreur de la sortie du modèle primaire, ce qui entraîne une interconnexion parallèle entre les deux modèles. Cette interconnexion nous emmène à un seul modèle final unique et complet possédant une résolution plus élevée.

2. Méthode proposée pour la modélisation des séries temporelles

Notre objectif est d'offrir un modèle neuronal fiable et efficace pour représenter les séries temporelles. Un modèle primaire est initialement conçu à l'aide des données réelles des séries temporelles. Ensuite, l'erreur entre les données réelles d'une série temporelle et la sortie du modèle primaire est modélisée pour créer le modèle d'erreur. Ce dernier modèle représente les incertitudes dans le modèle primaire qui peuvent être facilement supprimés par une simple soustraction de la sortie du modèle d'erreur de la sortie du modèle Primaire. Les détails de cette méthode seront présentés dans les sections suivantes.

Un modèle neuronal peut représenter ou modéliser n'importe quel système ou fonction inconnu, $y = f(x)$, en utilisant les données d'entrée/sortie. L'architecture des réseaux de neurones utilisée est les réseaux de neurones non bouclés discutée dans la section.5 du chapitre II, dont l'idée est de trouver une relation entre l'entrée et la sortie du modèle neuronal en ajustant au fur et à mesure ses paramètres en utilisant des mécanismes d'ajustement (algorithme d'optimisation) de telle façon une fonction objectif atteint son minimum. Pour avoir l'objectif voulu notre méthode sera composée de trois étapes :

- **Étape 1** : identification du modèle primaire;
- **Étape 2** : identification du processus d'erreur;
- **Étape 3** : conception du modèle final.

2.1. Paramètres du modèle neuronal à ajuster

La conception du modèle neuronal pour la modélisation comprend la détermination des paramètres inconnus qui sont les poids de différentes couches du la structure du réseau de neurones artificiels utilisée (voir [Figure IV.1](#)).

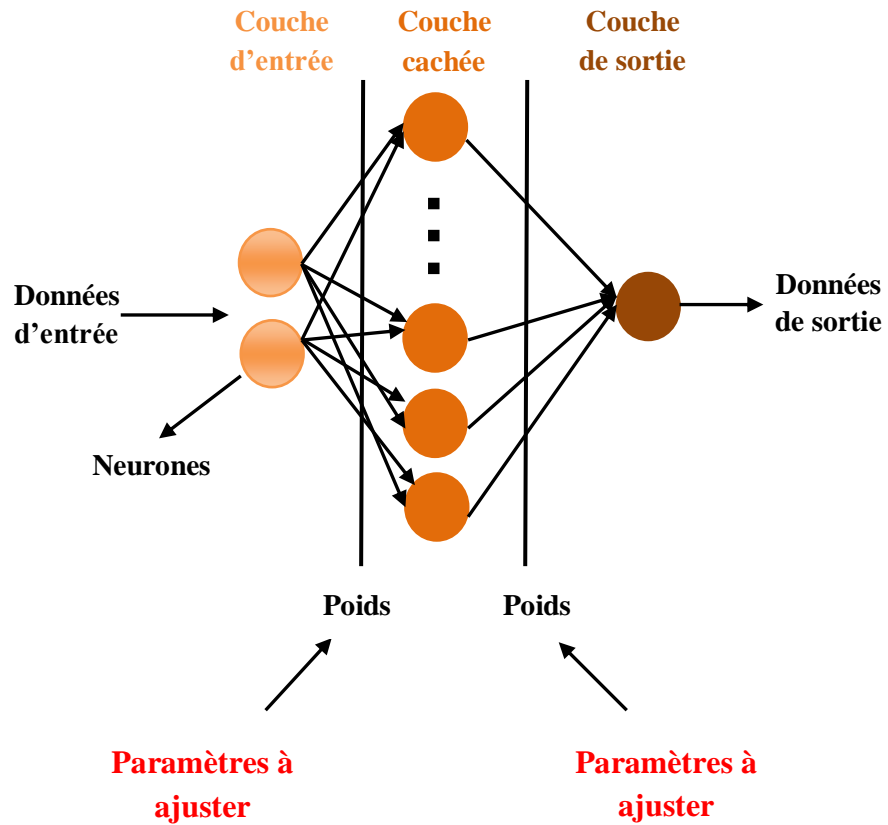


Figure IV.1 : Paramètres du réseau de neurones à optimiser.

2.2. Fonction objectif

Le terme fonction objectif ou la fonction de coût est utilisé en optimisation mathématique pour désigner une fonction qui sert de critère pour déterminer la meilleure solution à un problème d'optimisation. La fonction objectif utilisée est le critère d'erreur quadratique moyenne (*MSE* : *Mean Square Error*) qui combine les valeurs réelles et estimées comme suit :

$$MSE = \frac{\sum_{k=1}^N (y_k - \hat{y}_k)^2}{N} = \frac{\sum_{k=1}^N e^2}{N}. \quad (1)$$

Avec y_k et \hat{y}_k sont la mesure réelle et son estimation, respectivement, et N est la taille des données.

2.3. Identification du modèle primaire

Pour cette étape, l'ensemble de données d'entrée/sortie est utilisé pour déterminer le modèle neuronal primaire \hat{f}_p pour le processus qui est la série temporelle sous considération (voir [Figure IV.2](#)).

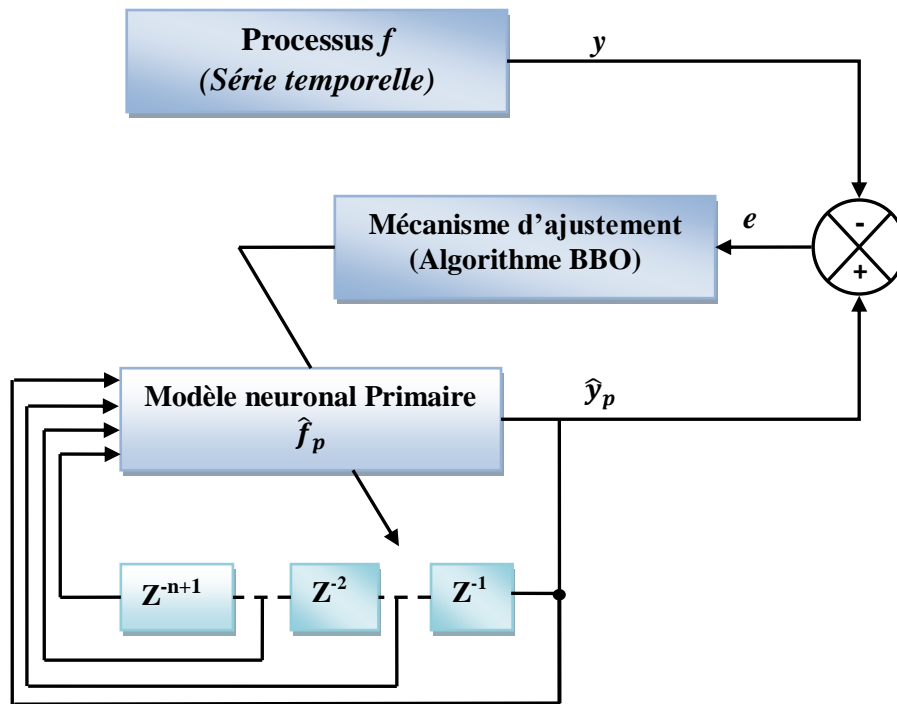


Figure IV.2 : Identification du modèle primaire.

Avec :

f : fonction inconnue représentant la série temporelle que nous souhaitons identifier, ayant y comme sortie.

\hat{y}_p : la sortie du modèle primaire.

$e = \hat{y}_p - y$: erreur de modélisation, utilisée dans la fonction du coût (1) de l'algorithme d'optimisation (pour exciter le mécanisme d'ajustement).

Le principe de l'identificateur neuronal consiste en une adaptation en temps réel de \hat{f}_p . Le bloc mécanisme d'ajustement représenté dans la Figure IV.2, est l'algorithme d'optimisation à base de biogéographie (BBO) discuté dans la section 3.2.3.4 du chapitre III qui va ajuster les paramètres du modèle neuronal \hat{f}_p de telle sorte que l'erreur e entre la sortie du processus y et la sortie du modèle primaire \hat{y}_p atteigne sa valeur minimale.

Pour identifier \hat{f}_p , on a utilisé un modèle autorégressif (AR) neuronal, qui est une technique de modélisation qui fait partie d'un groupe de formules de prédiction qui tentent de prédire la sortie d'un système en se basant sur les résultats précédents (ancienne sortie devient nouvelle entrée).

2.4. Identification du processus d'erreur

Le processus d'erreur (E_p) est défini comme une interconnexion parallèle entre le processus (série temporelle considéré) et le modèle neuronal primaire comme le montre la figure suivante :

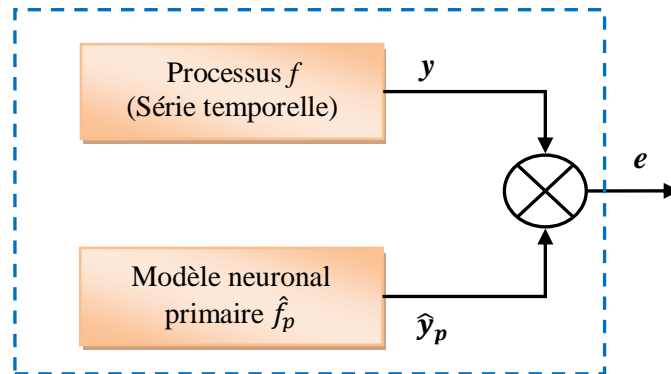


Figure IV.3 : Processus d'erreur.

La sortie e du processus d'erreur (E_p) est définie comme suit :

$$e = \hat{y}_p - y. \quad (2)$$

Une fois que la sortie du processus d'erreur (E_p) est obtenue (Equation. 2), un deuxième modèle neuronal peut être conçu pour la modélisation de l'erreur e . Ce modèle sera étiqueté par le modèle neuronal d'erreur. L'erreur e est aussi considéré comme une série temporelle, donc il convient de concevoir son modèle en utilisant un modèle neuronal autorégressif (**AR**), qui tente de prédire la nouvelle sortie en fonction des résultats précédents.

La structure de l'étape 2 est illustrée dans la Figure IV.4, où \hat{e} est la sortie du modèle neuronal d'erreur et e_1 est l'erreur de modélisation entre la sortie du processus d'erreur e et la sortie du modèle neuronal d'erreur \hat{e} .

La tâche maintenant est d'ajuster les paramètres du modèle neuronal d'erreur jusqu'à ce que l'erreur e_1 atteigne son minimum.

2.5. Conception du modèle final

Dans cette dernière étape, le modèle primaire \hat{f}_p et le modèle d'erreur \hat{E}_p sont interconnectés en une structure parallèle (voir Figure IV.5) afin d'obtenir le modèle final \hat{f}_F . Cette configuration nous permettra de compenser l'erreur résiduelle (erreur de modélisation) obtenu dans la première étape (modèle primaire), ce qui a comme

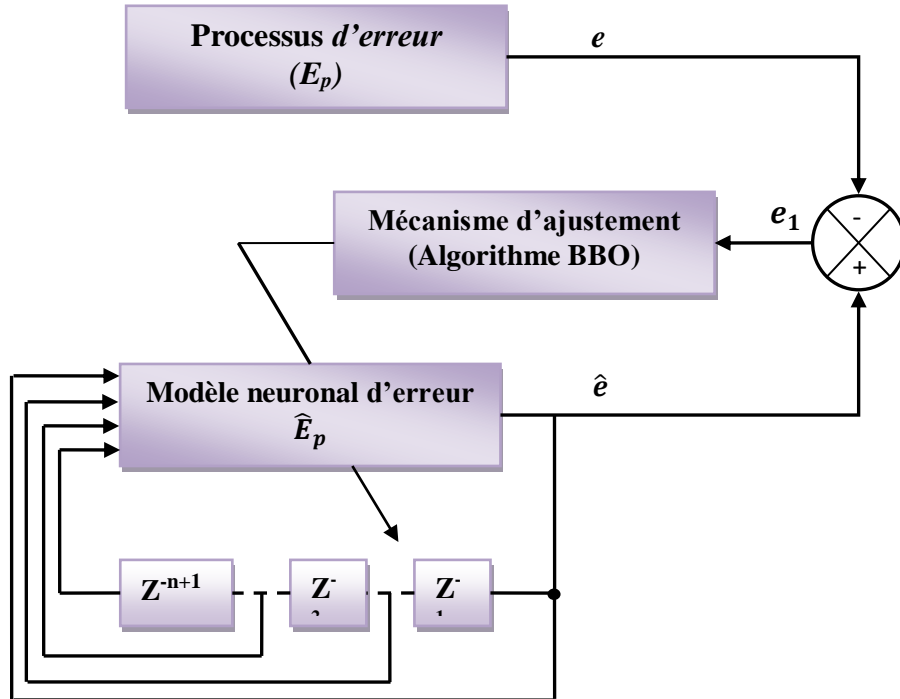


Figure IV.4 : Identification du processus d'erreur.

Conséquence la minimisation de l'erreur de modélisation globale. La sortie du modèle final peut être décrite comme suit :

$$\hat{y} = \hat{y}_p - \hat{e}. \quad (3)$$

Où \hat{y}_p et \hat{e} sont les sorties du modèle neuronal primaire et la sortie du modèle neuronal d'erreur, respectivement.

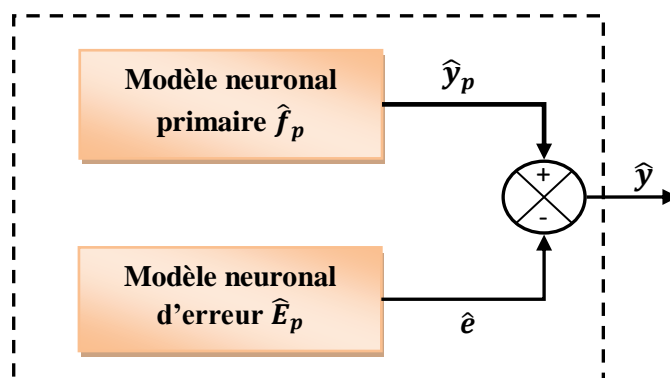


Figure IV.5 : Modèle final \hat{f}_F .

3. Résultats de simulation

Dans cette section nous allons présenter et discuter les résultats de simulation de notre méthode appliquée pour la modélisation des séries temporelles. Deux types de séries temporelles ont été considérés :

- 1- *Séries temporelles synthétiques* : générées à l'aide de modèles mathématiques à savoir :
 - ✓ Un signal électrocardiogramme synthétique [49].
 - ✓ La série temporelle Mackey-Glass qui est reconnu comme un problème de référence (*benchmark function*) [50].
- 2- *Séries temporelles réelles* : signaux électrocardiogrammes réels [51].

3.1. Paramètres d'algorithme d'optimisation BBO

La taille de la population utilisée dans les algorithmes d'optimisations est un facteur important dans la détermination de la solution optimale. Lorsque la taille de la population augmente, la solution dans l'espace d'exploration est améliorée ; cependant cela augmente le temps de calcul. Dans ce travail, la taille de population correspondante à l'algorithme BBO a été fixé à 200. Les différents paramètres d'algorithme d'optimisation BBO sont choisis comme suit : le taux de mutation $m = 0.01$ et le taux d'immigration et le taux d'émigration sont similaires aux courbes de migration linéaire (Figure III.4 du chapitre III).

3.2. Modélisation d'un signal électrocardiogramme synthétique

Dans cette section on va utiliser un modèle mathématique (dynamique) permettant de générer un signal électrocardiogramme [49]. Ce modèle peut être décrit par l'ensemble des trois équations différentielles suivantes :

$$\begin{cases} \dot{x}_1 = \alpha x_1 + \omega x_2 \\ \dot{x}_2 = \alpha x_2 + \omega x_1 \\ \dot{x}_3 = -\sum_{i \in \{P,Q,R,S,T\}} a_i \Delta \theta_i \exp\left(-\frac{\Delta \theta_i^2}{2b_i^2}\right) - (x_3 - z_0) \end{cases}; \quad (4)$$

Avec : $\Delta \theta_i = (\theta - \theta_i)$, $\theta = \arctan 2(x_2, x_1)$, a_i sont les paramètres représentant les amplitudes des Gaussiennes, les b_i représentent les largeurs des Gaussiennes et z_0 représente la ligne de base du signal ECG décrite comme suit :

$$z_0 = A \sin(2\pi f_r t); \quad (5)$$

Avec : $a = 0.15mv$ et $f_r = 0.25Hz$.

Le **Tableau IV.1** représente les paramètres du modèle (4) utilisés pour générer un signal électrocardiogramme synthétique.

Tableau IV.1 : Paramètre du modèle d'ECG (4).

Indice (i)	P	Q	R	S	T
<i>Temps (s)</i>	-0.2	-0.05	0	0.05	0.3
θ_i (Radians)	$-\pi/3$	$-\pi$	0	π	$\pi/2$
a_i	1.2	-0.5	30.0	-7.5	0.75
b_i	0.25	0.1	0.1	0.1	0.4

La technique de modélisation des séries temporelles discutée précédemment sera appliquée pour modéliser les données du signal électrocardiogramme synthétique. Les résultats de simulation sont illustrés dans les Figures 6-a, 6-b et 6-c, respectivement, telle que :

- ✓ **Figure. 6-a** : représente une superposition du signal ECG synthétique et la sortie du modèle neuronal primaire.
- ✓ **Figure. 6-b** : représente une superposition du signal ECG synthétique et la sortie du modèle neuronal final de ce signal.
- ✓ **Figure. 6-c** : représente une superposition de l'erreur de modélisation, le modèle de l'erreur de modélisation et l'erreur de modélisation finale.

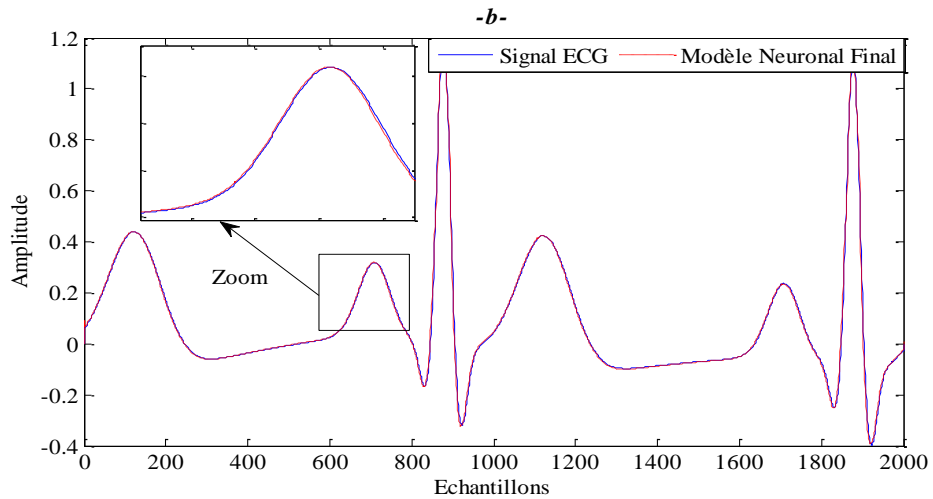
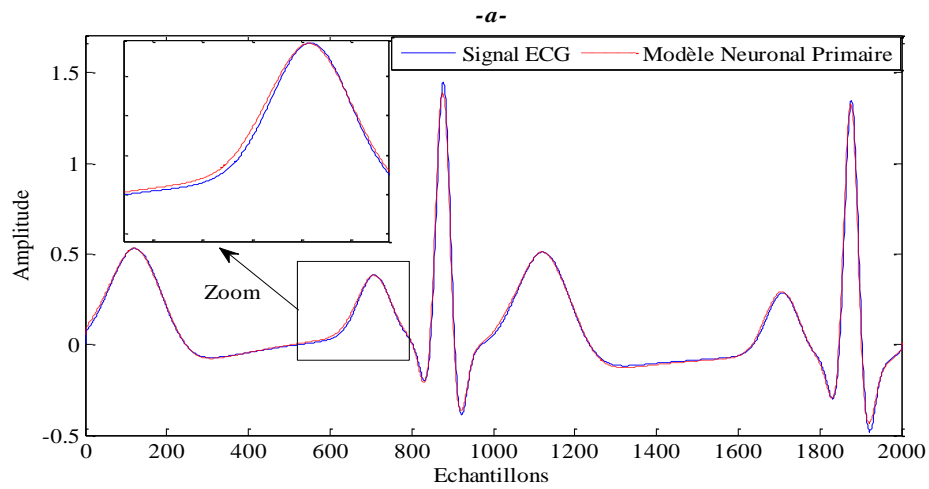
Par une simple inspection visuelle des Figures. 6-a, 6-b et 6-c, on peut remarquer que le modèle final est beaucoup mieux que le modèle primaire, où on constat que l'erreur de modélisation finale a été réduite ce qui permet d'améliorer efficacement le modèle final.

3.3. Modélisation d'un signal électrocardiogramme réel

Dans cette section on va considérer l'application de la technique de modélisation neuronale pour la modélisation d'un signal électrocardiogramme réel. Pour cette fin le signal électrocardiogramme 16539.dat pris de la base de données MIT-BIH Physio Bank [51] va être considéré comme une série temporelle inconnue que nous souhaitons de modéliser.

Les résultats de simulation sont illustrés dans les Figures 7-a, 7-b et 7-c, respectivement, telle que :

- ✓ **Figure. 7-a** : représente une superposition du signal ECG réel 16539.dat et la sortie du modèle neuronal primaire.
- ✓ **Figure. 7-b** : représente une superposition du signal ECG réel 16539.dat et la sortie du modèle neuronal final de ce signal.
- ✓ **Figure. 7-c** : représente une superposition de l'erreur de modélisation, le modèle de l'erreur de modélisation.



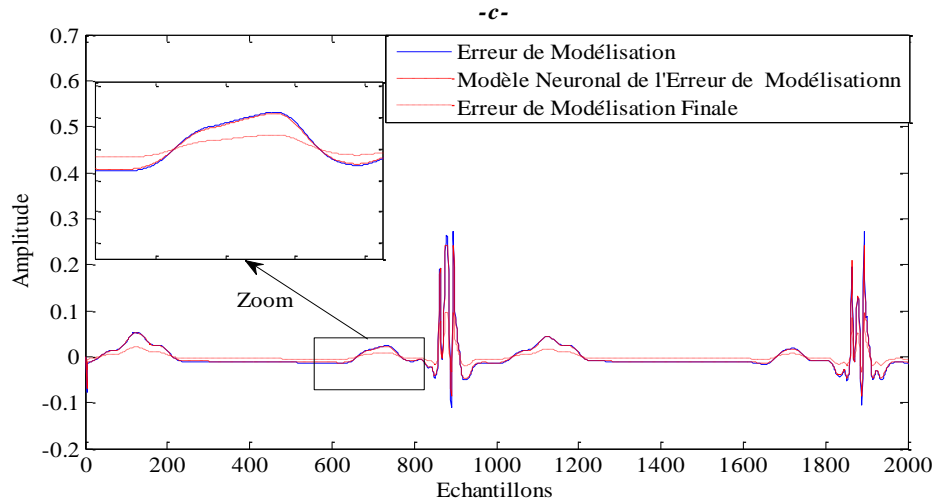
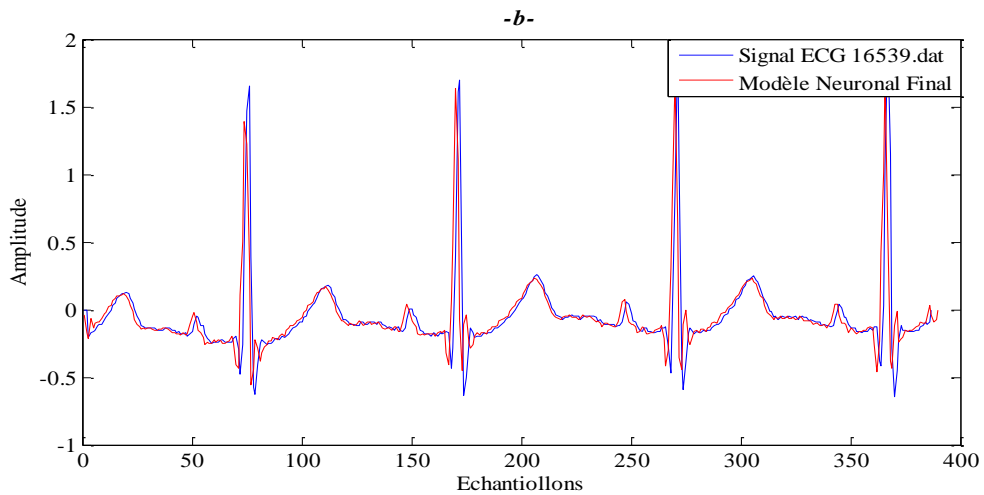
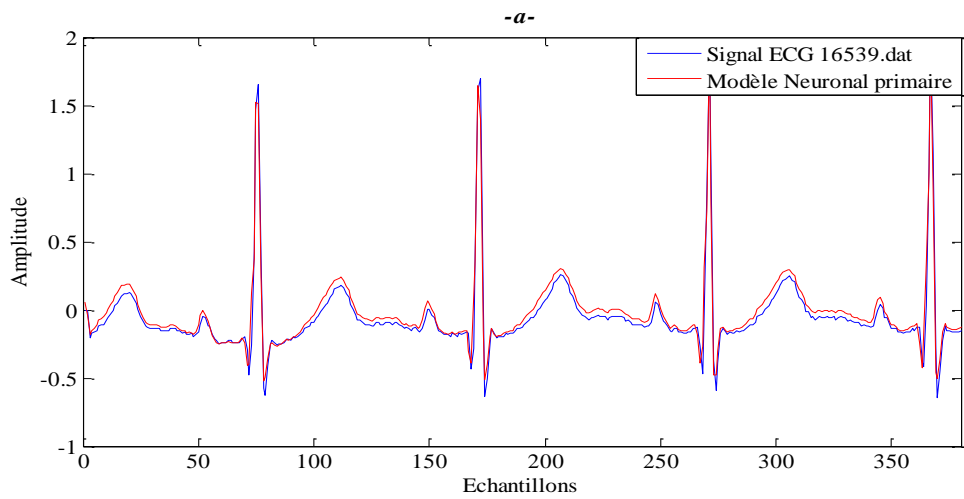


Figure IV.6 : Résultats de simulation du signal électrocardiogramme synthétique : (a) Modèle Primaire, (b) Modèle Final, (c) Erreurs de Modélisation.



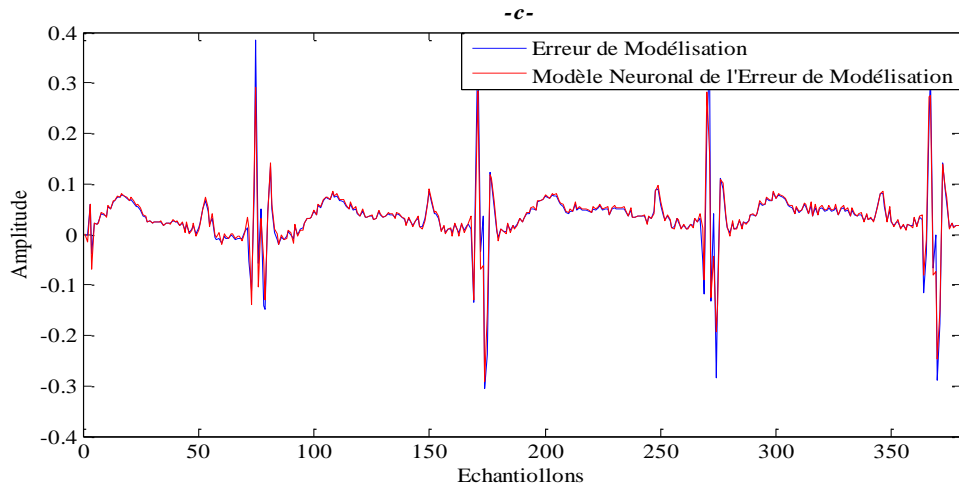


Figure IV.7 : Résultats de simulation du signal électrocardiogramme réel : (a) Modèle Primaire, (b) Modèle Final, (c) Erreurs de Modélisation.

3.4. Modélisation de fonction de référence Mackey-Glass

La série temporelle Mackey-Glass [50] est reconnue comme un problème de référence, qui est défini par l'équation différentielle suivante :

$$\frac{dx(t)}{dt} = \frac{0.2 * x(t-\tau)}{(1+x(t-\tau)^{10})} - 0.1 * x(t). \quad (6)$$

1200 échantillons de la série temporelle temporelles Mackey-Glass ont été générée sen utilisant l'équation 6 (Figure IV.8), et après notre méthode de modélisation a été appliquée pour modéliser ces données. Le résultat de modélisation et représenté dans la Figure IV.9

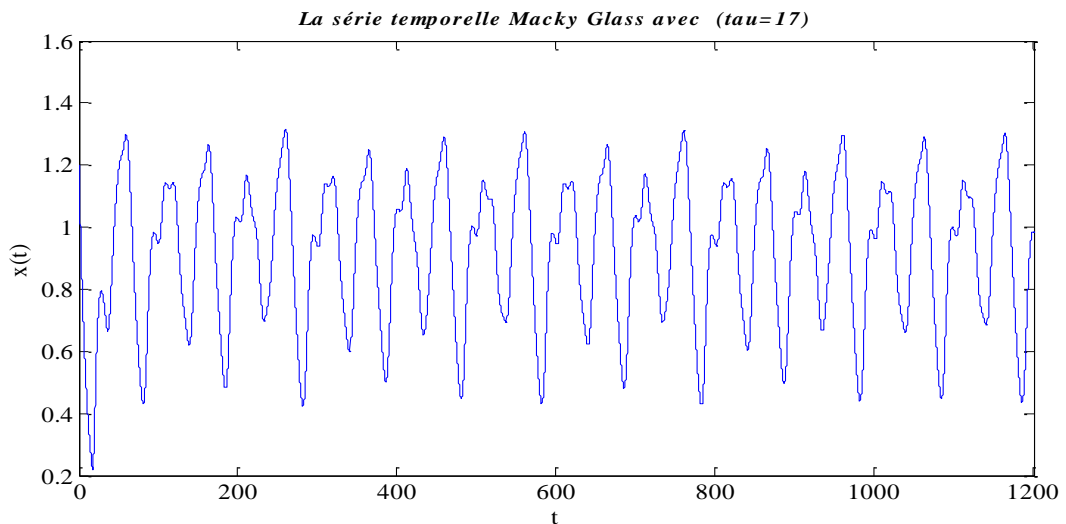


Figure IV.8 : La série temporelle Macky Glass avec Tau=17.

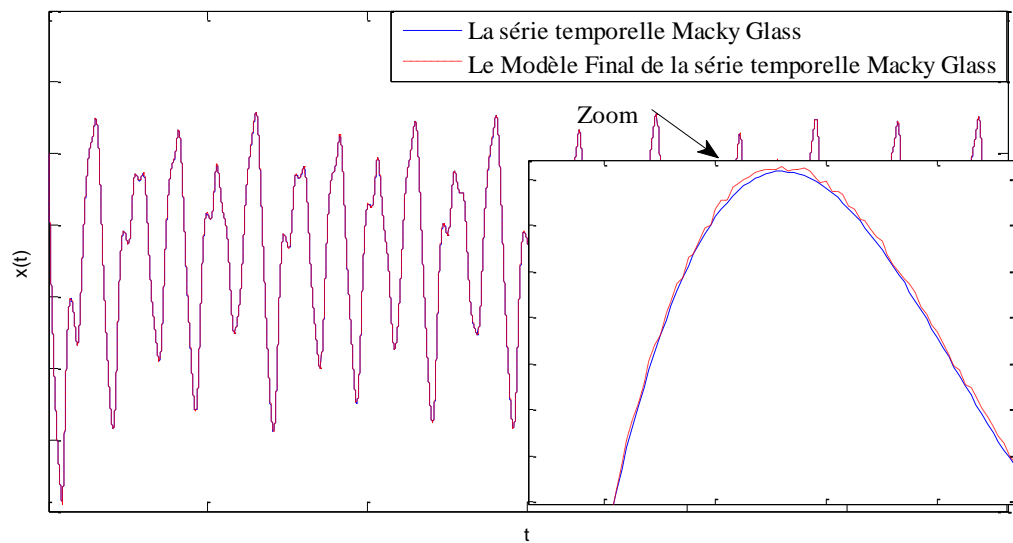


Figure IV.9 : Le résultat de modélisation de la série temporelle Macky Glass.

De tous ces résultats de simulation on peut remarquer l'efficacité de la structure proposée pour la modélisation des séries temporelles. La particularité ici est l'introduction d'un nouveau module s'appelle le modèle d'erreur qui permet de mieux laisser le modèle final de la série temporelle considéré. La deuxième particularité est l'utilisation de l'algorithme d'optimisation BBO pour ajuster les paramètres de la structure proposée à base des réseaux de neurones artificiels, ce qui élimine le problème du minimum local rencontré chaque fois en utilisant des méthodes d'optimisation du gradient.

4. Conclusion

Dans ce chapitre, nous avons présenté une méthode de modélisation des séries temporelles à base des réseaux de neurones artificiels et l'algorithme d'optimisation à base de biogéographie. L'approche permet de généraliser la notion d'identification en ajoutant un nouveau module d'identification, appelé modèle d'erreur. Le modèle neuronal d'erreur introduit a été utilisé comme un complément du modèle primaire pour améliorer la qualité du modèle, ce qui a fourni une bonne précision. L'optimisation des paramètres de la structure proposée a été assurée en utilisant l'algorithme d'optimisation BBO. Plusieurs résultats de simulations ont été introduits pour vérifier l'efficacité de notre méthode.



**CONCLUSION
GENERALE
ET
PERSPECTIVES**

CONCLUSION GENERALE ET PERSPECTIVE

Grâce aux développements dans le domaine de la technologie, nous avons pu obtenir de grandes quantités de données et les collecter également. Là où ces derniers peuvent être utilisés comme supports à la décision, ce qui conduit au développement d'outils capables de les analyser et de les traiter, les systèmes d'aide au diagnostic sont indispensables dans de nombreuses disciplines. L'objectif de ce travail est d'offrir un modèle neuronal fiable et efficace pour représenter les séries temporelles.

L'objectif de ce mémoire de fin d'étude a été de développer une technique de modélisation des séries temporelles à base des réseaux de neurones artificiels et l'algorithme d'optimisation à base de biogéographie. Cette technique comprend une interconnexion parallèle de deux sous modèles neuronaux. Le premier sous modèle neuronal est le modèle primaire, qui représente un modèle ordinaire à faible résolution. Pour surmonter le problème de la résolution et obtenir un modèle avec une résolution plus élevée, nous avons introduit un deuxième sous-modèle neuronal appelé le modèle d'erreur, qui représente la modélisation d'erreur entre la sortie du modèle primaire et la sortie réelle du système dynamique. Le modèle d'erreur représente l'incertitude dans le modèle principal, cette incertitude est minimisée par une simple soustraction de la sortie du modèle d'erreur de la sortie du modèle primaire, ce qui entraîne une interconnexion parallèle entre les deux modèles. Cette interconnexion nous emmène à un seul modèle final unique et complet possédant une résolution plus élevée.

L'approche développer dans le cadre de ce mémoire de fin d'étude permet de généraliser la notion d'identification en ajoutant un nouveau module d'identification, appelé modèle d'erreur. Le modèle neuronal d'erreur introduit a été utilisé comme un complément du modèle primaire pour améliorer la qualité du modèle, ce qui a fourni une bonne précision. L'optimisation des paramètres de la structure proposée a été

assurée en utilisant l'algorithme d'optimisation BBO. Plusieurs résultats de simulations ont été introduits pour vérifier l'efficacité de la structure développée.

Comme perspectives, nous proposons d'appliquer les autres techniques d'optimisation pour améliorer la qualité du résultat obtenue dont on peut citer :

- ✚ Les algorithmes génétiques (**PE, SE, AG, PG**).
- ✚ Les autres algorithmes méta-heuristiques (**ACO, AIS, PSO**).



BIBLIOGRAPHIES

Bibliographies

[1] Modèles linéaire pour les séries temporelles ULB 2014-STATS308 – économétrie

(Dehon-Van Bellegem) 194.

[2] Chatfield, C. (2000). *Time-series forecasting*. Chapman and Hall/CRC.

[3] Farouk, B. (2012). Méthodes Heuristiques pour la Prédiction des Séries Temporelles (Doctoral dissertation, Thèse présentée en vue de l'obtention du diplôme de Doctorat en Informatique, Université Badji Mohtar-Annaba, 2012. (Cité efinpage 30.)).

[4] Van Gestel, T., Suykens, J. A., Baestaens, D. E., Lambrechts, A., Lanckriet, G., Vandaele, B., ... & Vandewalle, J. (2001). Financial time series prediction using least squares support vector machines within the evidence framework. *IEEE Transactions on neural networks*, 12(4), 809-821.

[5] Lagnoux, A. (1996). Séries chronologiques. ISMAG, Master1-MI00141X, Université de Toulouse Le Mirail, 53p.

[6] Lagnoux, A. (1996). Séries chronologiques. *ISMAG, Master1-MI00141X, Université de Toulouse Le Mirail, 53p.*

[7] EL FAIZ Zakaria et ZOUIRI Sara (2016), master recherche en science Economique, UNIV. Mohammed V, FSJES Agdal. Analyse de la stationnarité d'une série chronologique : ADF Test.

https://zakariaelfaiz.blogspot.com/2016/02/analyse-de-la-stationnarite-dune-serie_24.html?m=1&fbclid=IwAR293uSEdWg5kxZLk6IXu8SD7NWmV9APd-mPZwjD-4lf7qdk8c4DIMfCzn4.

- [8] Ortiz-Arroyo, D., Skov, M. K., & Huynh, Q. (2005, November). Accurate electricity load forecasting with artificial neural networks. In International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06) (Vol. 1, pp. 94-99). IEEE.
- [9] Hill, T., O'Conner, M. et Remus, W. Neural network models for time series forecasts. *Management Science*, Vol. 42, No. 7, pp. 1082- 1092,1996.
- [10] Kajitani, Y., Hipel, K. W., & McLeod, A. I. (2005). Forecasting nonlinear time series with feed-forward neural networks: a case study of Canadian lynx data. *Journal of Forecasting*, 24(2), 105-117.
- [11] Bassi, D., & Olivares, O. (2006). Medium term electric load forecasting using TLFN neural networks. *International Journal of Computers Communications & Control*, 1(2), 23-33.
- [12] Buta, A., & Lustrea, B. (2005, November). Some Aspects Concerning Mid Term Monthly Load Forecasting Using ANN. In *EUROCON 2005-The International Conference on " Computer as a Tool"* (Vol. 1, pp. 253-256). IEEE.
- [13] Lisi, F., & Schiavo, R. A. (1999). A comparison between neural networks and chaotic models for exchange rate prediction. *Computational Statistics & Data Analysis*, 30(1), 87-102.
- [14] Faraway, J., & Chatfield, C. (1998). Time series forecasting with neural networks: à comparative study using the air line data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 47(2), 231-250.
- [15] Stern, H. S. (1996). Neural networks in applied statistics. *Technometrics*, 38(3), 205-214.
- [16] Lachtermacher, G., & Fuller, J. D. (1995). Back propagation in time-series forecasting. *Journal of forecasting*, 14(4), 381-393.
- [17] Jayawardena, A.W. et Fernando, D. Artificial neural networks in hydrometeorological modeling. In *Developments in Neural Networks and Evolutionary*

Computing for Civil and Structural Engineering, Topping BHV (ed.). Civil-Comp: Edinburgh, pp. 115- 120, 1995.

[18] Jayawardena, A. W., & Fernando, D. A. K. (1998). Use of radial basis function type artificial neural networks for runoff simulation. *Computer-Aided Civil and Infrastructure Engineering*, 13(2), 91-99.

[19] Hutchinson, J. M. (1993). A radial basis function approach to financial time series analysis.

[20] Touzet, C. (1992). Les réseaux de neurones artificiels, introduction au connexionnisme.

[21] H.Abdi. « *Les Réseaux de neurones* », édition presses universitaires de Grenoble, 1994.

[22] Youcef Djeriri (2017). Les réseaux de neurones artificiels, University of Sidi-bel-Abbes.

https://www.researchgate.net/publication/319939107_Les_Reseaux_de_Neurones_Artificiels.

[23] Ammar, M. Y. (2007). Mise en œuvre de réseaux de neurones pour la modélisation de cinétiques réactionnelles en vue de la transposition batch/continu (Doctoral dissertation).

[24] Coulibaly, P., Anctil, F., & Bobée, B. (1999). Prévission hydrologique par réseaux de neurones artificiels: état de l'art. *Canadian Journal of civil engineering*, 26(3), 293-304.

[25] SORIN, F., BROUSSARD, L., & ROBLIN, P. (2001). Régulation d'un processus industriel par réseaux de neurones. *Techniques de l'ingénieur. Informatique industrielle*, 2(S7582), S7582-1.

[26] Rivals, I., Personnaz, L., Dreyfus, G., & Ploix, J. L. (1995). Modélisation, classification et commande par réseaux de neurones: principes fondamentaux, méthodologie de conception et illustrations industrielles. *Les réseaux de neurones pour*

la modélisation et la commande de procédés, JP Corriou, ed.(Lavoisier Tec & Doc, 1995).

[27] Jayet, A. (2002). *Affective Computing: Apport des Processus Emotionnels aux Systèmes Artificiels*. Site.

[28] Dreyfus, G., Martinez, J. M., Samuelides, M., Gordon, M. B., Badran, F., Thiria, S., & Héroult, L. (2002). *Réseaux de neurones-Méthodologie et applications* (No. BOOK). Eyrolles.

[29] Karima BENLEMBAREK (2010), mémoire de MAGISTER UNIV- BADJI MOKHTAR- ANABA, *Système de Prédiction Basee Web: Applications aux Series Temporelles*.

<http://biblio.univ-annaba.dz/wp-content/uploads/2014/05/Memoire.pdf>.

[30] Michalewicz, Z., & Fogel, D. B. (2013). *How to solve it: modern heuristics*. Springer Science & Business Media.

[31] Godzik, A. (2005). *Approche évolutionnaires pour la robotique modulaire et anticipatoire* (Doctoral dissertation, Thèse de doctorat. Dept. Informatique., Paris XI Univ., Paris).

[32] Eiben, A. E., & Smith, J. E. (2003). *Introduction to evolutionary computing* (Vol. 53, p. 18). Berlin: springer.

[33] Bäck, T., Fogel, D. B., & Michalewicz, Z. (1997). *Handbook of evolutionary computation*. CRC Press.

[34] Bäck, T., Fogel, D. B., & Michalewicz, Z. (1997). *Handbook of evolutionary computation*. CRC Press.

[35] Eshelman, L. J. (1997). Genetic algorithms. *Handbook of Evolutionary Computation*.

[36] Bouchareb, A., & Laboudi, Z. (2012). Implémentation des algorithmes

[37] Hamlaoui, C., & Menassel, R. (2013). Optimisation d'un algorithme réversible de compression par l'utilisation des méta-heuristiques.

- [38] Ghoumari, A. (2018). *Métaheuristiques adaptatives d'optimisation continue basées sur des méthodes d'apprentissage* (Doctoral dissertation).
- [39] Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation*, 12(6), 702-713.
- [40] MacArthur, R. H., & Wilson, E. O. (1967). *The Theory of Island Biogeography*. Princeton (NJ): Princeton Univ.
- [41] Du, D., Simon, D., & Ergezer, M. (2009, October). Biogeography-based optimization combined with evolutionary strategy and immigration refusal. In *2009 IEEE International Conference on Systems, Man and Cybernetics* (pp. 997-1002). IEEE.
- [42] Ma, H., Simon, D., Fei, M., & Xie, Z. (2013). Variations of biogeography-based optimization and Markov analysis. *Information Sciences*, 220, 492-506.
- [43] Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.
- [44] Miller, B. L., & Goldberg, D. E. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex systems*, 9(3), 193-212.
- [45] ohnAach and George M Church. Aligning gene expression time series withtime warping algorithms. *Bioinformatics*, 17(6) :495–508, 2001.
- [46] rthur Schuster. On the periodicities of sunspots. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 206 :69–100, 1906.
- [47] Jan G De Gooijer and Rob J Hyndman. 25 years of time series forecasting. *International journal of forecasting*, 22(3) :443–473, 2006.
- [48] Klaus Robert Muller, Alexander J Smola, Gunnar Ratsch, Bernhard Scholkopf, Jens Kohlmorgen, and Vladimir Vapnik. Using support vector machines for time series prediction. *Advances in kernel methods—support vector learning*, 1999.
- [49] Patrick E McSharry, Gari D Clifford, Lionel Tarassenko, and Leonard A Smith. A dynamical model for generating synthetic electrocardiogram signals. *IEEE Transactions on Biomedical Engineering*, 50(3) :289–294, 2003.

[50] Marzi, Hosein, Ahmed Haj Darwish, and HumamHelfawi. "Training ANFIS using the enhanced Bees Algorithm and least squares estimation." *Intelligent Automation & Soft Computing* 23.2 (2017): 227-234.

[51] PhysioNet ECG database : The MIT-BIH Normal Sinus Rhythm Database.
[https ://physionet.org/physiobank/database/nsrdb/](https://physionet.org/physiobank/database/nsrdb/).