

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Filière Électronique
Spécialité Instrumentation

présenté par

RAMDANE MOHAMED

&

DJARER MOHAMMED

Contrôle intelligent par réseaux de neurone pour la détection d'anomalie en radiothérapie

Proposé par : Mr .AllAM Abdelkrim & Co-encadré par Mr .GESSOUM

Année Universitaire 2018-2019

Remerciements

Avant tout, nous louons Allah le tout-puissant pour nous avoir donné la patience et la volonté afin de mener à bien ce modeste travail.

Nous exprimons nos profonds remerciements, notre vive reconnaissance et notre sincère gratitude à Messieurs **Allam A.**, qui nous ont permis de poursuivre notre projet au sein du **CDTA** et Messieurs **GESSOUM.A** pour avoir accepté de nous co-encadrer, pour leurs conseils et leurs précieuses orientations qu'ils n'ont cessé de nous apporter tout au long de ce travail

Nous remercions toutes les personnes qui ont contribué à la réalisation de ce travail.

Nous remercions les personnes qui ont contribué à la réalisation de ce travail

RAMDANE.M , DJARER. M.

Enfin, Nous tenons à remercier chaleureusement nos collègues, avec qui nous avons partagé l'apprentissage. Merci pour le sourire, l'aide et la gentillesse.

ملخص:

الهدف الرئيسي من المشروع هو الكشف عن الخلل في قسم العلاج الإشعاعي باستعمال اللذكاء الاصناعى. الجزء الأول من هذا العمل هو دراسة الشبكات العصبية و انواعها. أما الجزء الثاني من هذا العمل هو اخذ عينة لكل من درجة الحرارة الرطوبة و الضغط عن طريق المستشعر MS6503 ثم تدريب هذه العينة ووضع تصنيف لها عن شبكة عصبية احادية المسار ثم القيام باختبارها .

كلمات المفاتيح: الذكاء الصناعي ,الشبكات العصبية , المستشعر LabVIEW , MATLAB, MS6503

Résumé : L'objectif principal du projet est de détecter les défauts dans le département de radiothérapie en utilisant l'intelligence artificielle, la première partie de ce travail étant l'étude des réseaux de neurones et de leurs types. La deuxième partie de ce travail consiste à prélever un échantillon de la température, de l'humidité et de la pression à l'aide du capteur MS6503, puis à former cet échantillon et la classification d'un réseau de neurones à voie unique, puis à le tester.

Mots clés : l'intelligence artificielle, réseaux de neurone, capteurs MS6503 , MATLAB, LabVIEW.

Abstract :

The main objective of the project is to detect defects in the radiotherapy department using artificial intelligence, the first part of this work being the study of neural networks and their types. The second part of this work consists in taking a sample of the temperature, humidity and pressure using the MS6503 sensor, then forming this sample and the classification of a single-track neuron network, then to test it

Keywords : artificial intelligence, neural network ,sensors MS6503, MATLAB, LabVIEW.

Listes des acronymes et abréviations

AI: intelligence artificielle.

RNA : réseaux de neurones artificiels.

PMC: perceptrons multicouche.

RBF: radial basic function.

MLP: multilayer perceptron.

MATLAB: mathwork laboratory.

MSE: mean square error.

EQM: erreur quadratique moyenne.

NN: neurale network.

NET: network.

SIM: simulation.

NEWFF: neurale network feed-forward.

LABVIEW: Laboratory virtuel Instrument Engineering Workbench.

SVM: support vector machine.

USB: Universal Serial Bus.

AML: analytic and machine learning.

VI: Instruments virtuels.

LR: logistic regression.

C-SVM: coefficient d'un support vector machine.

NU-SVM: noyau d'un support vector machine.

T : température.

H : humidité.

P : pression.

Sommaire :

Introduction générale	1
Chapitre 1 : Réseaux de Neurone.....	3
1.1 Introduction	3
1.2 NEURONE	3
1.2.1 Modèle d'un neurone biologie	3
1.2.2 Neurone formel.....	4
1.2.3 Modélisation d'un RNA	5
1.3 Réseaux de neurones artificiels	6
1.3.1 Définition.....	6
1.3.1 FONCTIONNEMENT	6
1.3.2 Fonction d'activation	7
1.4 ARCHITECTURE DES RESEAUX DE NEURONES	9
1.4.1 Les réseaux de neurones non bouclés	9
1.4.2 Les réseaux de neurones bouclés	9
1.4.3 Les réseaux de neurones convolutifs(CNN)	9
1.5 Apprentissage	10
1.5.1 APPRENTISSAGE Supervisé.....	10
1.5.2 LE RENFORCEMENT	10
1.5.3 Apprentissage non supervisé	11
1.5.4 Apprentissage hybride	11
1.6 Types de réseaux à entraîner.....	11
1.6.1 Le perceptron multicoucheMLP.....	11
1.6.2 Réseaux à fonction radiale (RBF)	12
1.7 Algorithme de rétro propagation du gradient.....	12
1.7.1 Principe.....	12
1.7.2 Algorithme.....	13
a Définition du réseau	13
b Etapes de l'algorithme	13
1.8 APPLICATION.....	14
1.9 Conclusion.....	15

Chapitre 2	: Conception et implémentation	16
2.1	Introduction	16
2.2	OUTILS ET MATRIELS	16
2.3	base de données	17
2.4	Construction d'un réseau de neurone sur MATLAB	17
2.4.1	Collecte de données	17
2.4.2	Construire le réseau	17
a	Phase d'apprentissage	17
b	Phase d'entraînement	19
c	Simuler le réseau de neurone	22
2.5	Exécution	23
2.5.1	Architecture	23
2.5.2	Performance d'apprentissage :	24
2.5.3	Training :	24
2.5.4	Régression	25
2.5.5	Detection les sorties	26
2.5.6	Performance de test	26
2.6	ETUDE INFLUENCE DE NOMBRE DE COUCHE CACHENT	27
2.7	discussion générale	28
2.8	Conclusion	29
Chapitre 3	: Développement Labview	30
3.1	Introduction	30
3.2	Généralité sur LABVIEW	30
3.2.1	Les instruments virtuels (Vis)	31
3.2.2	Environnement LABVIEW	31
3.2.3	Commandes et Indicateurs	33
3.3	Toolkit analytic and machine learning(AML)	33
3.3.1	Définition	33
3.3.2	Caractéristiques et fonctionnalité d'un AML	34
3.3.3	Exemples prêts à l'emploi pour	34
3.3.4	Palette de fonctionnement	34

3.3.5	Rôles des palettes AML	34
3.4	Vis de classification	35
3.4.1	Déployer le modèle de classification VI	36
3.4.2	Évaluer le modèle de classification VI.....	36
3.4.3	Initialiser le modèle de classification (LR) VI.....	38
3.4.4	Initialiser le modèle de classification (NN) VI	40
3.4.5	Initialiser le modèle de classification (SVM) VI	41
a	La notion de marge maximale	42
b	Fonction noyau (kernel).....	42
3.4.6	Définir le modèle de classification VI	45
a	SVM	45
b	Réseau neuronal	46
c	Régression logistique	47
3.4.7	Modèle de classification de train VI.....	47
3.5	EXEMPLE D'Implémentation.....	48
3.5.1	Introduction :	48
a	Face diagramme.....	49
b	Face avant	49
c	Résultat	50
3.5.2	Discussion.....	51
3.6	Conclusion.....	52
	Conclusion générale.....	53
	Annexes.....	55
	Bibliographie	79

Liste des figures :

<i>Figure 1. 1</i> : le neurone biologique.	4
<i>Figure 1. 2</i> : neurone formel.	5
<i>Figure 1. 3</i> : Correspondance neurone biologique/neurone artificiel.	5
<i>Figure 1. 4</i> : Modèle général d'un neurone.	7
<i>Figure 1. 5</i> : Structure d'un réseau CNN.	10
<i>Figure 1. 6</i> : Structure d'un réseau MLP.....	12
<i>Figure 1. 7</i> : Structure d'un réseau RBF.....	12
<i>Figure 2. 1</i> : architecture de réseaux de neurone.....	23
<i>Figure 2. 2</i> : L'erreur quadratique moyenne (EQM) en fonction du nombre d'itération.	24
<i>Figure 2. 3</i> : training modèle réseaux de neurone.	25
<i>Figure 2. 4</i> : régression modèle réseaux de neurone.....	25
<i>Figure 2. 5</i> : classification des sorties.....	26
<i>Figure 2. 6</i> : erreur de classification.....	27
<i>Figure 3. 1</i> : Environnement LABVIEW 2017.....	31
<i>Figure 3. 2</i> : fenêtre de face-avant.....	32
<i>Figure 3. 3</i> : fenêtre de diagramme.	32
<i>Figure 3. 4</i> : palette des fonctions et palette des commandes.....	33
<i>Figure 3. 5</i> : palette AML.....	34
<i>Figure 3. 6</i> : les modèles de classifications.	35
<i>Figure 3. 7</i> : modèle de déploiement.....	36
<i>Figure 3. 8</i> : modèle d'évaluation.	36
<i>Figure 3. 9</i> : modèle de classification (LR).....	38
<i>Figure 3. 10</i> : modèle de classification (NN).	40
<i>Figure 3. 11</i> : Maximisation de la marge pour obtenir l'hyperplan optimal.....	42
<i>Figure 3. 12</i> : La transformation non linéaire.	43
<i>Figure 3. 13</i> : modèle de classification (SVM).	43
<i>Figure 3. 14</i> : modèle de définition SVM.	45
<i>Figure 3. 15</i> : modèle de définition NN.....	46
<i>Figure 3. 16</i> : modèle de définition LR.	47
<i>Figure 3. 17</i> : modèle de train.....	47
<i>Figure 3. 18</i> : face diagramme.....	49
<i>Figure 3. 19</i> : face avant.....	50
<i>Figure 3. 20</i> : Résultat de sortie.	50
<i>Figure 3. 21</i> : poids de neurone.	50
<i>Figure 3. 22</i> : Résultat de performance en sortie.	51

Liste des tableaux :

<i>Tableau 1. 1</i> : Analogie entre le neurone biologique et le neurone artificiel.	6
<i>Tableau 1. 2</i> : Fonctions d'activation.	8
<i>Tableau 2. 1</i> : influence de nombre de couche caché.	27
<i>Tableau 3. 1</i> : méthode de calcul moyenne.	37
<i>Tableau 3. 2</i> : exemple d'une matrice de confusion.	38
<i>Tableau 3. 3</i> : fonction d'activation de couche caché.	40
<i>Tableau 3. 4</i> : fonction d'activation de couche sortie.	41
<i>Tableau 3. 5</i> : les types modèles SVM.	43
<i>Tableau 3. 6</i> : les fonctions du noyau (kernel).	44

Introduction générale

La radiothérapie est une méthode de traitement locorégional des cancers, utilisant des radiations pour détruire les cellules cancéreuses en bloquant leur capacité à se multiplier. L'irradiation a pour but de détruire toutes les cellules tumorales tout en épargnant les tissus sains périphériques.

La salle de radiothérapie est influencée par des facteurs physiques comme température, humidité et pression, qui peuvent parfois entraîner des risques pour les patients s'ils ne sont pas bien programmés. Donc ce traitement nécessite d'attention et de rigueur dans les salles pour contrôler et éviter tout les dangers. Il améliore la qualité des services et augmente les soins médicaux tout en appliquant certaines procédures médicales, techniques et physiques. Ces procédures doivent être appliquées par le personnel du service de radiothérapie.

Le Centre de Développement des Technologies Avancées CDTA d'Alger et le centre de traitement de cancer "Centre Pierre et Marie Curie" (CPMC) d'Alger d'une part proposent plusieurs méthodes de contrôle comme éventuelle solution, l'une de ces méthodes est l'utilisation de la technique l'intelligence artificielle en utilisant les réseaux de neurones dans le processus de contrôle, soit dans la conception du modèle soit dans la conception du contrôleur. Ces derniers présentent plusieurs avantages notamment comme approximateurs globaux qui peuvent fournir des solutions efficaces aux problèmes de contrôle des systèmes non linéaires. La possibilité d'apprentissage peut réduire l'effort humain lors de la conception des contrôleurs et permet de découvrir des structures de contrôle plus efficace que celles déjà connues.

Les algorithmes des réseaux de neurones sont basés sur les entrées et les sorties des systèmes. Notre travail consiste à appliquer l'apprentissage supervisé et les

algorithmes de rétropropagation des réseaux de neurones afin d'obtenir un modèle valide du système à commander. Une fois cette étape réalisée, le modèle peut être intégré dans une approche de commande prédictive non linéaire.

PM Cest jugée très efficace pour aborder des problèmes de contrôle, grâce à l'habileté de prédiction donnée par le modèle encasté, ainsi que le maniement de la contrainte. C'est la raison pour laquelle nous avons choisi, dans notre travail, d'entreprendre une régulation numérique par la commande prédictive non linéaire.

Dans ce mémoire nous présentons le travail réalisé et il est organisé comme suit : dans le premier chapitre, nous présentons essentiellement les concepts fondamentaux des réseaux de neurones: propriétés mathématiques, architectures neuronales, types d'apprentissage ...etc., tout en se focalisant sur les réseaux multicouches MLP utilisés dans ce travail. Dans le second chapitre nous présentons la conception d'un apprentissage de base de données sur MATLAB. Le troisième chapitre a pour objectif la présentation et la description des AML en utilisant le logiciel LABVIEW.

En fin nous terminerons par une conclusion générale pour synthétiser le travail et les résultats obtenus.

Chapitre 1 : Réseaux de Neurone

1.1 Introduction

Le principe des réseaux de neurones artificiels est né dans les années 40 à partir d'une analogie avec le système nerveux humain. Ils sont basés sur la fonction neuronale, parce que les neurones sont identifiés comme éléments cellulaires responsables du traitement de l'information dans le cerveau. Les réseaux de neurones artificiels se sont donc basés sur l'hypothèse disant que le raisonnement intelligent des êtres humains a pour origine la structure de système nerveux et donc on peut l'inculquer à un ordinateur en lui implantant un réseau de neurones artificiels pour le rendre « intelligent », tout en sauvegardant sa puissance et sa rapidité d'exécution.

L'approche neuronale c'est développée avec l'expansion de l'utilisation des outils informatiques qui facilitent l'analyse numérique de l'information et la mise en œuvre de cette méthode.

1.2 NEURONE

1.2.1 Modèle d'un neurone biologique

Un neurone est une cellule nerveuse dont la fonction est de transmettre un signal électrique dans certaines conditions. Il agit comme un relai entre une couche de neurone et celle qui la suit. Sa structure se compose de quatre(04) régions principales (le corps cellulaire, les dendrites, l'axone, la synapse).

Le corps cellulaire : qui contient le noyau de la cellule nerveuse, c'est en cet endroit que prend naissance l'influx nerveux, qui présente l'état d'activité de neurone.

Les dendrites : ramifications tubulaires courtes formant une espèce d'arborescence autour du corps cellulaire, ce sont les entrées principales du neurone, qui captent l'information venant d'autres neurones.

L'axone : longue fibre nerveuse qui se ramifie à son extrémité, c'est la sortie du neurone et le support de l'information vers les autres neurones.

La synapse : qui communique l'information en la pondérant par un poids synaptique, à un autre neurone, elle est essentielle dans le fonctionnement du système nerveux.

Chaque neurone réalise une opération très simple, qui est en fait une somme pondérée de ses entrées. Le résultat est comparé à un seuil et le neurone devient excité si ce seuil est dépassé. L'information contenue dans le cerveau est représentée par les poids donnés aux entrées de chaque neurone [1].

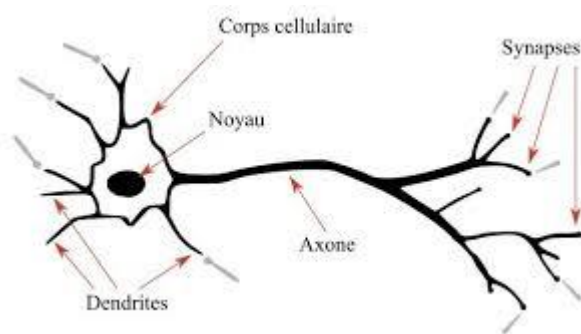


Figure 1. 1 : le neurone biologique.

1.2.2 Neurone formel

Est un processeur très simple imitant grossièrement la structure et le fonctionnement d'un neurone biologique, la première version du neural formel est celle de McCulloch et Pitts qui est largement utilisée [2].

Un neurone formel est un opérateur effectuant la somme pondérée de ses entrées externes puis elle s'active suivant la valeur de cette sommation pondérée puis modulée par une fonction d'activation (Figure1.2) [3].

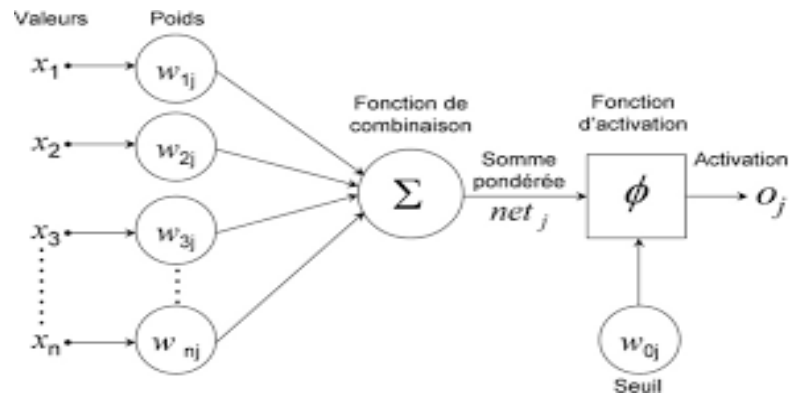


Figure 1. 2 : neurone formel.

1.2.3 Modélisation d'un RNA

La modélisation consiste à rassembler les connaissances que l'on a du comportement dynamique du processus par une analyse physique des phénomènes mis en jeu, et une analyses des données expérimentales, ces analyses conduisent à la définition des grandeur caractérisant le processus, c'est-à-dire ses entrées, ses variable d'état et ses sorties.

La modélisation consiste à mettre en œuvre un système de RN sous un aspect non pas biologique, on aura une correspondance pour chaque élément composant le neurone biologique, donc une modélisation pour chacun d'entre eux. [4]

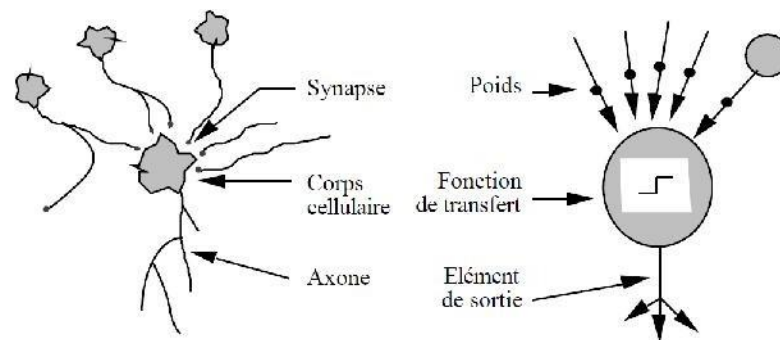


Figure 1. 3 : Correspondance neurone biologique/neurone artificiel.

On pourra résumer cette modélisation par (tableau 1.1), qui montre la transmission entre le neurone biologie et le neurone artificiel.

Neurone biologique	Neurone artificiel
Dendrite	Signal d'entrée
Synapses	Poids de connexions
Corps cellulaire	fonction d'activation
Axones	Signal de sortie

Tableau 1. 1 : Analogie entre le neurone biologique et le neurone artificiel.

1.3 Réseaux de neurones artificiels

1.3.1 Définition

Un réseau de neurones consiste en un ensemble de neurones reliés entre eux par des connexions pondérées. Il se caractérise principalement par le type des unités utilisées et par sa topologie. On distingue souvent deux types de neurones particuliers dans un réseau : les neurones d'entrée recevant les données du monde extérieur, et les neurones de sortie fournissant le résultat du traitement effectué. Les autres unités sont généralement qualifiées de cachées. Cette distinction n'est toutefois pas obligatoire et tous les neurones peuvent très bien communiquer dans les deux sens avec l'extérieur [5].

1.3.1 FONCTIONNEMENT

Un modèle de neurone formel est une règle de calcul qui permet d'associer aux x entrées une sortie: c'est donc une fonction à x variables et à valeurs réelles. Chaque entrée x est associée un poids synaptique, c'est-à-dire une valeur numérique notée de w_1 pour l'entrée 1 jusqu'à w_x pour l'entrée x . La première opération réalisée par le neurone formel consiste en une somme des grandeurs reçues en entrées, pondérées par les coefficients synaptiques, c'est-à-dire la somme [6] :

$$w_1 x_1 + w_2 x_2 + \dots + w_m x_m = \sum_{j=1}^m w_j x_j$$

Cette grandeur est comparée à un seuil θ . Le résultat est alors transformé par une fonction d'activation non linéaire F . La sortie associée aux entrées x_1 à x_m est ainsi donnée par : [6]

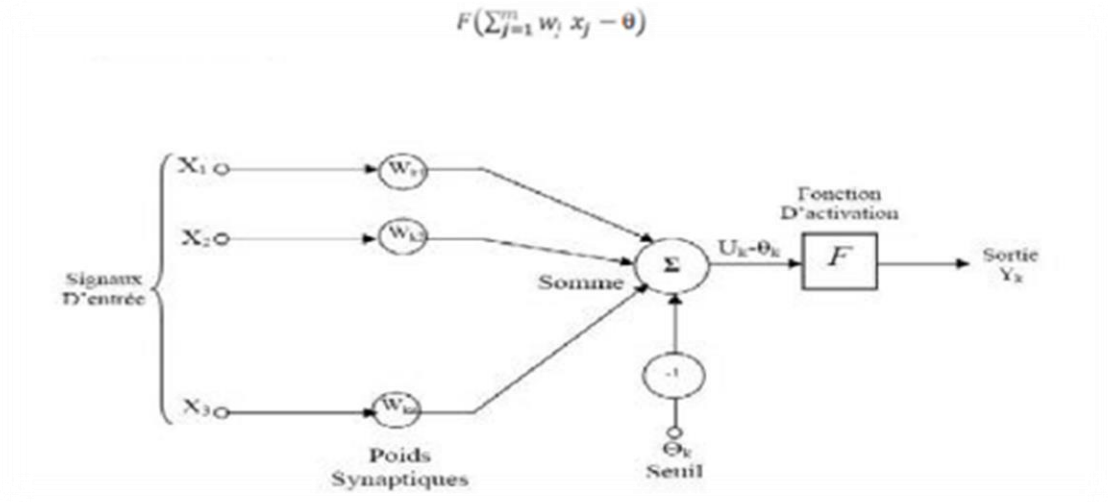


Figure 1. 4 : Modèle général d'un neurone.

1.3.2 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) [7].

(Le tableau 1.2) englobe l'ensemble des fonctions d'activations qui sont généralement utilisées.

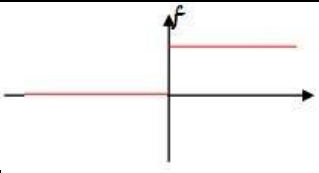
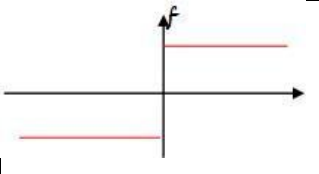
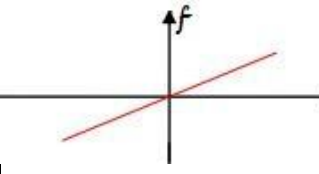
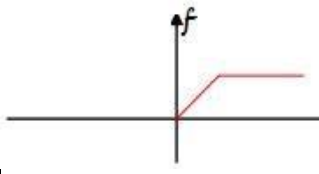
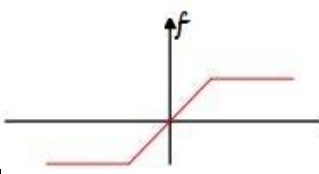
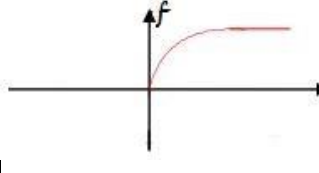
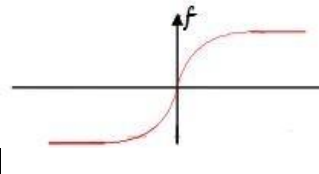
Nom de la fonction	Type	Equation	Allure	Nom Matlab
Seuil	Binaire (fonction de Heaviside)	$f(x) = 0$ si $x < 0$ $f(x) = 1$ si $x \geq 0$		Hardlim
	Signe	$f(x) = 1$ si $x > 0$ $f(x) = -1$ si $x \leq 0$		Hardlims
Linéaire	Identité	$f(x) = x$		Purelin
	Saturé positif	$f(x, k) = 0$ si $x < 0$ $f(x, k) = 1$ si $x \geq 1/k$ $f(x, k) = kx$ sinon		Satlin
	Saturé symétrique	$f(x, k) = 1$ si $x < -1/k$ $f(x, k) = -1$ si $x \geq 1/k$ $f(x, k) = kx$ sinon		Satlins
Sigmoïde	Positive (type logistique)	$f(x, k) = \frac{1}{1 + e^{-kx}}$		Poslin
	Symétrique (type tanh)	$f(x, k) = \frac{2}{1 + e^{-kx}} - 1$		Logsig

Tableau 1. 2 : Fonctions d'activation.

1.4 ARCHITECTURE DES RESEAUX DE NEURONES

La topologie ou architecture d'un RNA est la manière selon laquelle les neurones sont connectés et organisés. Généralement, les RNA peuvent être d'une connectivité totale ou chaque neurone est relié à tous les autres neurones du réseau, ou d'une connectivité locale dont les neurones ne sont liés qu'à leurs voisins. Alors, On distingue deux topologies des RNA : les réseaux de neurones non bouclés ou réseaux « feed-forward » et les réseaux de neurones bouclés ou réseaux « feed-back ». [8]

1.4.1 Les réseaux de neurones non bouclés

Appelés aussi réseaux « proactifs » ou « de type perceptron » ou « feed-forward ». Un Réseau non bouclé se caractérise par la propagation des signaux dans un seul sens, de la couche d'entrée vers la couche de sortie, sans utilisation d'aucune boucle de rétroaction. Il est généralement statique car les entrées et les sorties sont indépendantes du temps.

Ce type de RNA est utilisé principalement pour effectuer des tâches d'approximation de fonction non linéaire, de classification ou de modélisation de processus statiques non linéaires. [9]

1.4.2 Les réseaux de neurones bouclés

Les réseaux « récurrents » ou bien « feed-back » se caractérisent par la présence, au moins, d'une boucle de rétroaction des neurones de sorties vers les neurones d'entrée. Ces réseaux sont dynamiques ; ils régissent par des équations aux différences non linéaires à cause des retards associés aux connexions Ils sont utilisés fréquemment à l'affectation des tâches de modélisation de systèmes dynamiques, de commande de processus, ou de filtrage.[9]

1.4.3 Les réseaux de neurones convolutifs(CNN)

Les réseaux de neurones convolutifs (ConvNets) sont des outils largement utilisés pour l'apprentissage en profondeur. Ils sont particulièrement adaptés aux images en tant qu'entrées, bien qu'ils soient également utilisés pour d'autres applications telles que le

texte, les signaux et d'autres réponses continues. Ils diffèrent des autres types de réseaux de neurones de plusieurs façons [10]. Les réseaux de neurones convolutionnels sont inspirés de la structure biologique d'un cortex visuel, qui contient des arrangements de cellules simples et complexes [11]. Ces cellules s'activent en fonction des sous-régions d'un champ visuel. Ces sous-régions sont appelées champs réceptifs. Inspirés des résultats de cette étude, les neurones dans une couche convolutive se connectent aux sous-régions des couches avant cette couche au lieu d'être entièrement connectés comme dans d'autres types de réseaux neuronaux.

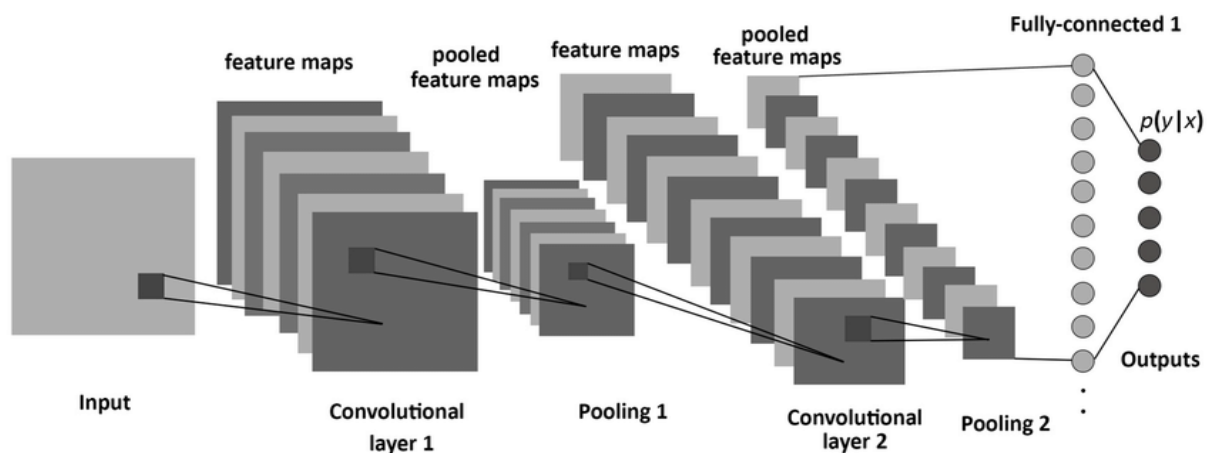


Figure 1.5 : Structure d'un réseau CNN.

1.5 Apprentissage

1.5.1 APPRENTISSAGE Supervisé

Les résultats corrects sont fournis au réseau (c'est-à-dire les valeurs que l'on désire que le réseau obtienne en sortie), si bien que celui-ci peut ajuster ses poids de connexions pour les obtenir. Après l'apprentissage, le réseau est testé en lui donnant seulement les valeurs d'entrée mais pas les sorties désirées, et en gardent si le résultat obtenu est proche du résultat désiré ou non.

1.5.2 LE RENFORCEMENT

Dans cette approche le réseau doit apprendre la relation échec/succès, le système étant capable ici de savoir si réponse qu'il fournit est correcte ou non, mais son connaître la bonne réponse.

1.5.3 Apprentissage non supervisé

On ne fournit pas au réseau les sorties que l'on désire obtenir, on le laisse évoluer librement jusqu'à ce qu'il se stabilise.

1.5.4 Apprentissage hybride

Une partie de poids va être déterminé par l'apprentissage supervisé et l'autre par le non supervisé.

1.6 Types de réseaux à entraîner

1.6.1 Le perceptron multicoucheMLP

Ce type de réseau est un des réseaux «propagation vers l'avant», c-à-d qu'en mode normale d'utilisation, l'information se propage dans un sens unique, des entrées vers les sorties sans aucune rétroaction. Son apprentissage est de type supervisé, par correction des erreurs. Dans ce cas uniquement, le signal d'erreur est rétro-propagé vers les entrées pour mettre à jour les poids des neurones. L'idée principale de ce type de RN est de grouper des neurones dans une couche. On place ensuite bout à bout plusieurs couches et connecte complètement les neurones de deux couches adjacentes. Les entrées des neurones de la deuxième couche sont donc en fait les sorties des neurones de la première couche. Les neurones de la première couche sont reliés au monde extérieur et reçoivent tous le même vecteur d'entrée. Ils calculent alors leur sorties qui sont transmises aux neurones de la deuxième couche,... etc. Les sorties des neurones de la dernière couche forment la sortie du RN. Deux neurones d'une même couche ne sont pas reliés entre eux [13]. (Figure 1.6).

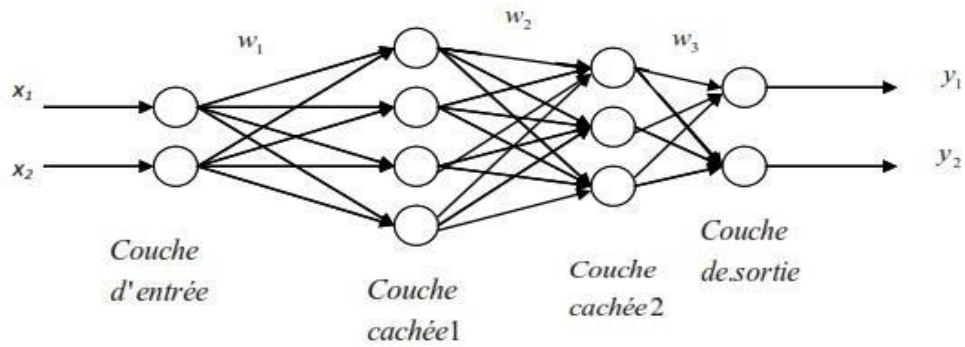


Figure 1. 6 : Structure d'un réseau MLP.

1.6.2 Réseaux à fonction radiale (RBF)

L'architecture de RBF (RADIAL BASIC FONCTIONS) est la même que pour le PMC, les fonctions de base utilisées ici sont des fonctions gaussienne, il se compose d'une seule couche intermédiaire. Les RBF seront donc employés dans les mêmes types de problèmes que le MLP. L'apprentissage le plus utilisé pour le RBF est le mode hybride et les règles sont soit la règle de correction de l'erreur soit la règle d'apprentissage par compétition [14].

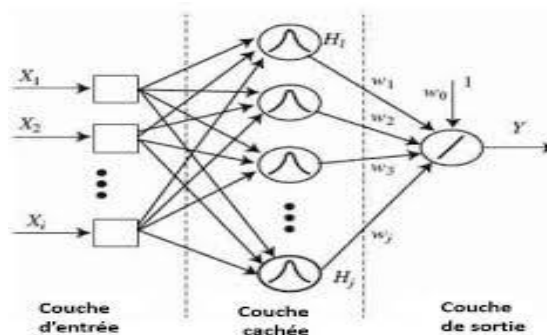


Figure 1. 7 : Structure d'un réseau RBF.

1.7 Algorithme de rétro propagation du gradient

1.7.1 Principe

La rétro propagation du gradient consiste à propager « à l'envers » (de la couche de sortie vers la couche d'entrée) l'erreur obtenue sur les exemples de la base d'apprentissage. On utilise pour cela l'erreur quadratique (le carré de la différence entre ce qu'on obtient et ce qu'on désire).

Si on calcule la dérivée partielle de l'erreur quadratique par rapport aux poids des connexions (gradient), il est possible de déterminer la contribution des poids à l'erreur générale, et de corriger ces poids de manière à se rapprocher du résultat souhaité. La correction par itération en corrige plus ou moins fortement les poids par l'intermédiaire d'un coefficient η . [15]

Après un certain nombre d'itérations, où on n'est satisfait du classement des exemples de notre base d'apprentissage, on fixe les poids qui constituent aussi des frontières entre les classes.

1.7.2 Algorithme

a *Définition du réseau*

Soit un réseau multicouche définit par:

Une couche d'entrée à m cellules d'entrées. Ces cellules ne sont pas des neurones mais simplement des entrées $x_i = e_i$ du réseau.

Une couche cachée qui contient n neurones qui ont une fonction d'activation y_i .

Une couche de sortie à p neurones qui ont une fonction d'activation z_k .

$n \times m$ connexions entre la couche d'entrée et la couche cachée, chacune pondérée par w_{ji} .

$m \times p$ connexions entre la couche cachée et la couche de sortie, chacune pondérée par w_{kj} . [16]

b *Etapas de l'algorithme*

ETAPE 1 : Initialisation au hasard ou aléatoire des poids des connexions w_{ji} et w_{kj}

ETAPE 2 : Propagation des entrées $x_i = e_i$.

On propage vers la couche cachée $= (\sum x_i * w_{ij} + x_0) = 1$

Puis de la couche cachée vers la couche de sortie $= (\sum y_j * w_{kj} + y_0) = 1$

Les valeurs x_0 et y_0 sont des Biais.

F : est la fonction d'activation qu'on a choisie où on a défini notre réseau MLP.

ETAPE 3 : rétro propagation de l'erreur pour chaque neurone de la couche de sortie on calcule l'erreur, c'est-à-dire la différence entre la sortie désirée s_k et la sortie réelle (obtenue) z_k .

Si :

La fonction d'activation des neurones cachés est sigmoïde alors on aura :

$$E_k = z_k(1 - z_k)(s_k - z_k)$$

On propage cette erreur sur la couche cachée ; l'erreur de chaque neurone de la couche cachée est donnée par $\delta_j = (1 - \sum_k w_{kj} * E_k) * \delta_j$

ETAPE 4 : Correction des poids des connexions : Il reste maintenant la modification des poids des connexions et aussi les biais.

o Entre la couche d'entrée et la couche cachée : $\Delta w_{ji} = \lambda y_j \delta_j$

$$\Delta x_0 = \lambda \delta_j$$

o Entre la couche cachée et la couche de sortie : $\Delta w_{kj} = \lambda z_k E_k$

$$\Delta y_0 = \lambda E_k$$

Où :

Est un coefficient de proportionnalité appartient à l'intervalle $]0,1[$, appelé pas d'apprentissage. BOUCLER à l'étape 2 jusqu'à un critère d'arrêt à définir. [17]

1.8 APPLICATION

On utilise les réseaux de neurones artificiels dans différents domaines (informatique, électronique, science cognitive, hydrologie, neurobiologie), l'étude des réseaux de neurones est une voie prometteuse de l'Intelligence Artificielle, qui a des applications dans de nombreux domaines:

Industrie : contrôle qualité, diagnostic de panne, corrélations entre les données fournies par différents capteurs, analyse de signature ou d'écriture manuscrite.

Finance : prévision et modélisation du marché (cours de monnaies...), sélection d'investissements, attribution de crédits.

Télécommunications et informatique : analyse du signal, reconnaissance de formes (bruits, images, paroles), compression de données.

Environnement : évaluation des risques, analyse chimique, prévisions et modélisation météorologiques et hydrologiques, gestion des ressources.

1.9 Conclusion

Dans ce chapitre nous avons présenté le fonctionnement, les différentes architectures et méthodes d'entraînement ainsi que les caractéristiques principales des réseaux de neurones artificiels qui sont utilisés dans ce travail afin de décrypter les données d'un capteur de température, d'humidité et la pression utilisé dans les salles de traitement de radiothérapie.

Chapitre 2 : Conception et implémentation

2.1 Introduction

Nous présentons dans ce chapitre notre contribution qui consiste à la détection d'anomalie en radiothérapie. Nous avons utilisées des RN pour réaliser notre modèle.

Nous présentons une base d'apprentissage pour exécuter nos expériences sur MATLAB.

Nous avons opté pour l'architecture de perceptron multicouche MLP type de réseaux de neurones.

2.2 OUTILS ET MATRIELS

MATLAB est un environnement informatique numérique et un langage de programmation. Il permet une manipulation facile de la matrice, la représentation graphique des fonctions et des données, la mise en œuvre d'algorithmes, la création d'interfaces utilisateur et l'interfaçage avec des programmes dans d'autres langues. La boîte à outils du réseau neuronal contient les outils MATLAB permettant de concevoir, mettre en œuvre, visualiser et simuler des réseaux neuronaux. Il fournit également une prise en charge complète de nombreux paradigmes de réseau éprouvés, ainsi que d'interfaces utilisateur graphiques (IHM) permettant à l'utilisateur de concevoir et de gérer des réseaux de neurones de manière très simple.[18]

Dans cette travail , MATLAB (R2015) est utilisé pour écrire des fichiers de script permettant de développer des modèles MLP ANN ainsi que des fonctions de performance permettant de calculer les statistiques d'erreur de performance de modèle telles que RMSE.

Le programme MLP commence par la lecture des données d'un fichier Excel (Training.xlsx et Testing.xlsx). La fonction "xlsread" est utilisée pour lire les données spécifiées dans le fichier Excel.

2.3 base de données

- Dans cette étude nous à pris un échantillon (Température, L'humidité et Pression) Pour classifier en quatre classe (0,1,2,3) à chacun échantillon.
- Ces échantillons étaient sauvegarder dans MATLAB, on utilise le toolkit NEURAL NETWORK TOOLBOX pour classifier.
- Cette base de données a été divisée en deux sous ensembles, le premier sert à effectuer l'apprentissage et le deuxième sous ensemble est utilisé pour tester les performances, c'est ce qu'on appelle l'ensemble test. Notons que l'ensemble d'apprentissage contient 274 échantillons et l'ensemble test contient 131 échantillons ce qui donne une période de Enregistrement de l'humidité, pression et de la température: décembre 2014 à janvier 2016.

2.4 Construction d'un réseau de neurone sur MATLAB

La robustesse des réseaux de neurones et la variété de ces méthodes dans la classification, nous avons choisi l'un des plus répondu qui est le perceptron multicouche (PMC).

La conception des modèles ANN suit un certain nombre de procédures systématiques:

2.4.1 Collecte de données

La collecte et la préparation des échantillons de données constituent la première étape de la conception des modèles ANN. Les données de mesure de la température, l'humidité et la pression pendant une période donnée par défaut.

2.4.2 Construire le réseau

a *Phase d'apprentissage*

Notre réseaux neurone multilayer perceptron(MLP) utilise dans l'apprentissage une base se donnée contient 274 échantillons.

Tous ses échantillons entrées complètement dans le MLP de 5 couches d'entrées et avec un nombre de couche caché fixé à une (1) neurone et le nombre de couche sortie est 1. En même temps en appliquant la propagation et la rétropropagation pour calculer et définir les poids de chaque entrée entre la couche entrée et la couche cachée et aussi entre la couche cachée et la couche de sortie et l'erreur quadratique moyenne.

MATLAB aide à concevoir le modèle MLP en utilisant la fonction intégrée «newff» qui crée un réseau de propagation rétroactif.

De plus, cette commande initialisera automatiquement les poids et les biais.

La fonction est appelée comme suit: [19]

newff: crée un objet réseau de feed-forward-backpropagation et initialise automatiquement le réseau.

```
net = newff (P, T, S);
```

```
net = newff (PR, [S1 S2... SNI], {TF1, TF2,..., TFNI}, BTF, BLF, PF);
```

```
net = newff (P, T, S, TF, BTF, BLF, PF, IPF, OPF, DDF);
```

La description:

La fonction prend les paramètres suivants :

P - Matrice $R \times Q_1$ de Q_1 représentant les vecteurs d'entrée d'élément R.

T - Matrice $S_N \times Q_2$ de vecteurs cibles d'élément S_N représentatifs Q_2 .

S - Tailles des N-1 couches cachées, S_1 à $S(N-1)$, valeur par défaut = [].

PR- = matrice $R \times 2$ des valeurs min et max pour les éléments d'entrée R.

S_i - Nombre de neurones (taille) dans la ième couche, $i = 1, \dots, N_1$.

N_1 - Nombre de couches.

TF_i - Fonction de transfert de la couche i. La valeur par défaut est 'tansig' pour les couches masquées et 'purelin' pour la couche en sortie. Les fonctions de transfert $TF \{i\}$

peuvent être toute fonction de transfert différentiable telle que TANSIG, LOGSIG ou PURELIN.

BTF - Fonction d'apprentissage de la rétropropagation, défaut = 'traingdx'.

BLF - Fonction d'apprentissage de la rétropropagation, default = 'learnngdm'.

PF - Fonction de performance, défaut = 'mse'.

Et retourne un réseau de propagation N-feed feed-forward de couche N.

Newff : utilise un générateur de nombres aléatoires pour créer les valeurs initiales des pondérations du réseau.

Si les neurones doivent avoir différentes fonctions de transfert, ils doivent être disposés en différentes couches.

Exemple:

```
net = newff (minmax (p), [5, 2], {'tansig', 'logsig'}, 'traingdm', 'learnngdm', 'mse');
```

Pendant le processus de formation, les poids sont ajustés afin de rendre les sorties réelles (prédites) proches des sorties cibles (mesurées) du réseau.

MATLAB fournit des fonctions de transfert intégrées qui sont utilisées dans cette étude. linéaire (puréline),Sigmoides Tangente Hyperbolique (logsig) et Sigmoides Logistique (tansig).

b Phase d'entraînement

Pour entraîner le réseau il faut déclarer divers paramètres, comme le nombre de cycle pendant l'apprentissage (itérations, epochs) la fréquence de vérification de l'erreur d'apprentissage vis-à-vis de l'erreur désirée (goal), le taux d'apprentissage (η : mu). Il faut aussi déclarer la division des données en trois parties selon les trois phases (l'apprentissage, test et la validation). Le cycle d'apprentissage signifie un passage complet par tous les neurones.

train: est utilisé pour former le réseau chaque fois qu'un train est appelé.

Syntaxe:

```
net1 = train (net, P, T) ;
```

La description:

La fonction prend les paramètres suivants :

net - le réseau MLP initial généré par newff.

Vector P - Vecteur d'entrée mesuré du réseau.

T - Cibles du réseau (vecteur de sortie mesuré), par défaut = zéro. Et revient

net1 - Nouvel objet réseau.

Les paramètres d'entraînement du réseau (net.trainParam) sont définis pour contenir les paramètres suivants:

o trainParam: cette propriété définit les paramètres et les valeurs de la fonction de formation actuelle.

o net.trainParam: les champs de cette propriété dépendent de la propriété de la fonction de formation.

o Le plus utilisé de ces paramètres (composants de trainParam).

net.trainParam.epochs : qui indique à l'algorithme le nombre maximum d'époques à former.

net.trainParam.show : qui indique à l'algorithme combien d'époques il doit y avoir entre chaque présentation de la performance.

• L'entraînement a lieu en fonction des paramètres d'entraînement trainlm, indiqués ici avec leurs valeurs par défaut:

net.trainParam.epochs 1000: Nombre maximum d'époques à former.

net.trainParam.show 25 : époques entre les progrès.

net.trainParam.goal 0 : Objectif de performance.

net.trainParam.time inf : Durée maximale d'entraînement en secondes.

net.trainParam.min_grad 1e-6 : Gradient de performance minimal.

net.trainParam.max_fail 5 : Échecs de validation maximum.

- En règle générale, une époque d'apprentissage est définie comme une présentation unique de tous les vecteurs d'entrée sur le réseau. Le réseau est ensuite mis à jour en fonction des résultats de toutes ces présentations.

- Chaque pondération et biais est mis à jour en fonction de sa fonction d'apprentissage après chaque époque (l'un passe à travers l'ensemble des vecteurs d'entrée).

- trainFcn: cette propriété définit la fonction d'entraînement Backpropagation.

net.trainFcn = 'trainlm';

- performFcn: cette propriété définit la fonction utilisée pour mesurer les performances du réseau. La fonction de performance est la fonction qui détermine à quel point l'ANN accomplit sa tâche.

net.performFcn

o fonctions de performance :

Mse : Fonction d'erreur quadratique moyenne.

Pour préparer un réseau personnalisé à former avec mse, définissez :

net.performFcn = 'MSE'.

Pour la régression linéaire, on utilise généralement la fonction de performance d'erreur quadratique moyenne MSE.

- La formation s'arrête lorsque l'une de ces conditions est remplie:

- ✓ Le nombre maximum d'époques (répétitions) est atteint.

- ✓ La performance a été réduite au minimum.

- ✓ Le temps maximum a été dépassé.
- ✓ Les performances de validation ont augmenté de plus de max_fail fois depuis sa dernière diminution (lors de l'utilisation de la validation).
- train appelle la fonction indiquée par net.trainFcn, en utilisant les valeurs du paramètre d'apprentissage indiquées par net.trainParam. [19]

c *Simuler le réseau de neurone*

Après la phase d'apprentissage, on passe à la simulation du réseau en utilisant la décomposition des données entre test et validation.

Une fois la formation terminée, les performances du réseau doivent être vérifiées. Par conséquent, les données invisibles (tests) seront exposées au réseau. Le processus de simulation de test s'appelle avec l'énoncé suivant:

sim : est utilisé pour simuler le réseau lorsque sim est appelé.

Syntaxe:

a = sim (net1, P)

La description:

La fonction prend les paramètres suivants :

net1 - objet MLP final.

P - vecteur d'entrée

a - sortie mesurée.

- Pour tester dans quelle mesure le net résultant MLP est proche des données, sim Command est appliqué. La sortie mesurée est une (sortie simulée du réseau MLP).[19]
- Différence d'erreur ($e = T - a$) à chaque point mesuré. La validation finale doit être effectuée avec des données indépendantes.
- afficher le nom et les propriétés des variables d'un réseau de neurones.

Affichage (net)

•vue: voir la structure du réseau.

voir (net);

2.5 Exécution

Chaque fois répéter cette opération jusqu'à obtenir le résultat désiré pour utilise comme entrée de la partie de test.

Après de Suite à une série d'essais on obtient les figures suivantes :

2.5.1 Architecture

Nous avons un réseaux qui compose de 5 couche en entrée et une couche caché de 20 neurone qui utilise un fonction sigmoide pour l'état d'activation et un seule couche en sortie qui utilise la fonction softmax pour l'activation.

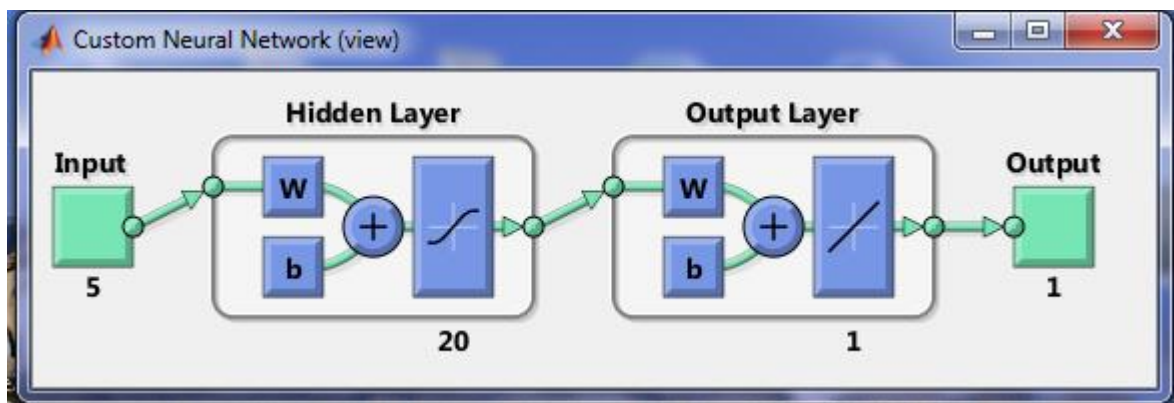


Figure 2. 1 : architecture de réseaux de neurone.

2.5.2 Performance d'apprentissage :

Nous à donnée où ils arrivent dans le processus de formation elle est proche de 10^{-3} . Où meilleur validation de performance EQM=0.0040731 à nombre itération est égale 6.

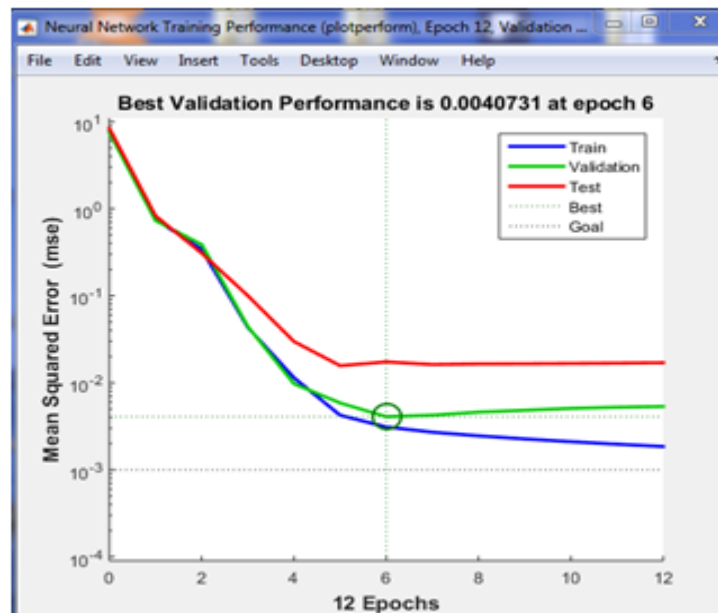


Figure 2. 2 : L'erreur quadratique moyenne (EQM) en fonction du nombre d'itération.

2.5.3 Training :

Le pas du gradient est un facteur déterminant dans la vitesse de convergence du réseau de neurones. Ou grad= 0.01201 à 12 itérations.

Dans le cadre de notre expérience le taux d'apprentissage est fixé à 0.001 et un nombre maximum d'itération est fixé à 1000 itérations. Durant l'apprentissage, le réseau est validé toutes les 12 itérations.

Val-fail : Échecs de validation maximum=6.

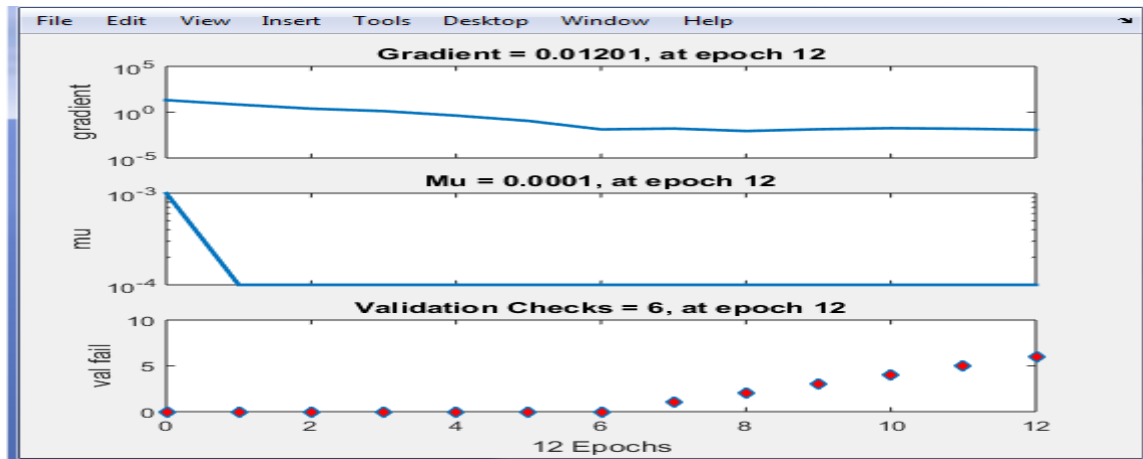


Figure 2.3 : training modèle réseaux de neurone.

2.5.4 Régression

Le modèle développé a donné le même coefficient de corrélation par approximation (0,99) durant les trois phases d'apprentissage, de test et de validation.

D'après ces résultats numériques, nous pouvons conclure que les modèles neuronaux ont des coefficients de corrélation presque égal à l'unité ($R \sim 1$). Ceci montre qu'il y a un grand rapprochement entre les valeurs mesurées et celles estimées par le modèle neuronal développé dans cette étude.

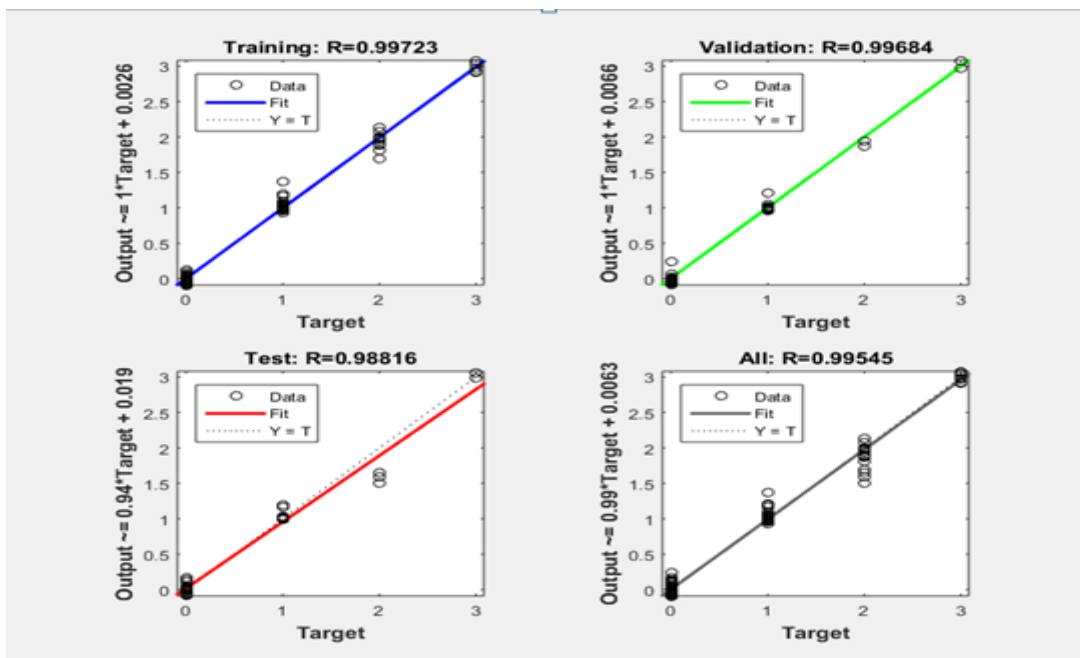


Figure 2.4 : régression modèle réseaux de neurone.

2.5.5 Detection les sorties

Après d'apprentissage le programme prédite des sortie estimés et met une classification selon le figure suivant (figure2.5) :

```
y =  
Columns 1 through 23  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
Columns 24 through 46  
0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 2 2 2 2 2 1 0 0  
Columns 47 through 69  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 1 1 2 2 1  
Columns 70 through 92  
1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 2 1 1 3 3 1  
Columns 93 through 115  
1 1 1 1 1 1 1 1 3 3 3 3 0 0 0 1 0 0 1 1 1 0 0  
Columns 116 through 131  
1 1 0 0 0 0 0 0 0 2 1 0 0 1 1 1
```

Figure 2. 5 : classification des sorties.

2.5.6 Performance de test

L'erreur quadratique moyenne EQM mesure l'écart entre les grandeurs physiques réelles (observations) et les grandeurs prévues par le modèle. (En comparant la sortie réelle du réseau avec la sortie désirée). Les résultats obtenue se montrent dans la (figure 2.6).

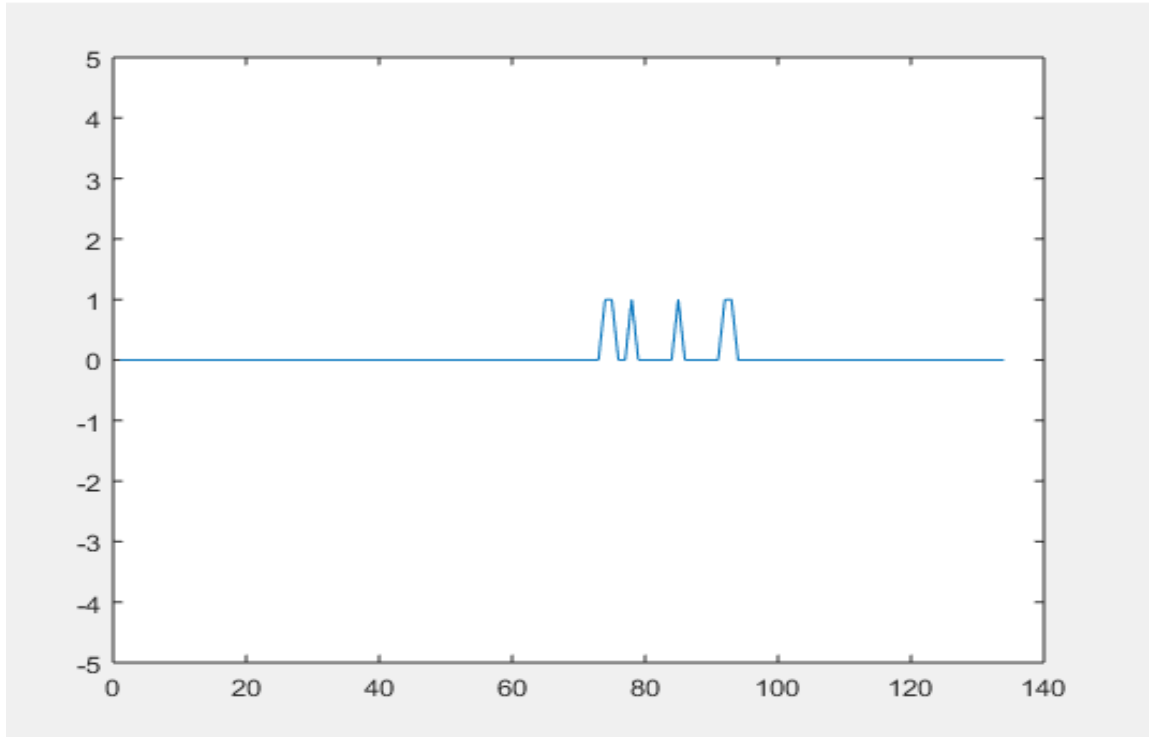


Figure 2. 6 : erreur de classification.

2.6 ETUDE INFLUENCE DE NOMBRE DE COUCHE CACHENT

On commence avec un réseau qui contient une seule couche cachée et on passe progressivement à un réseau qui contient une couche cachée de 25 neurones chacune.

Nombre de couche caché	Nbr itération	Mse d'apprentissage	Mse test
1	15	0.116	0.2090
10	20	0.00312	0.0896
20	26	0.00026	0.0448
25	97	5.03^{e-5}	0.0672

Tableau 2. 1 : influence de nombre de couche caché.

Les résultats obtenus montrent que si on augmente le nombre de couche caché le MSE d'apprentissage est diminué mais le MSE de test est aléatoire.

2.7 discussion générale

Nous avons opté donc pour le perceptron multicouche puisqu'il répond aux spécificités du problème de prédiction. Ce type de réseau apporte des résultats très significatifs et le taux de précision de la prédiction étant meilleur.

Dans le cadre de cette étude, nous a avons divisé, aléatoirement notre base de données en trois parties : phase d'apprentissage, phase de test et phase de validation.

La phase d'apprentissage ou d'entraînement a pour but de déterminer les paramètres de connexion du réseau à l'aide de la technique d'optimisation.

La phase de test consiste à vérifier le réseau déterminé lors de la phase d'apprentissage sur des données non utilisées lors de cette dernière, et à examiner la capacité du réseau à généraliser les exemples d'apprentissage (en comparant la sortie réelle du réseau avec la sortie désirée).

La phase de validation est effectuée sur la dernière partie des données.

L'apprentissage est stoppé lorsque l'erreur de validation devient minimum, ou bien le nombre d'itération fixé atteint le maximum.

Le taux d'apprentissage croissant très vite avec la complexité du réseau, il faut donc trouver un pas optimal. Plus le pas est petit, plus le nombre d'itérations de la base d'apprentissage sera important. Tandis que plus le pas est grand, plus ce nombre d'itérations nécessaire ne sera faible mais le réseau risque de diverger.

La prévision de futures sorties n'est pas obtenue par une linéarisation, mais à partir de l'utilisation récursive et successive d'un modèle non linéaire .permettant d'avoir commandes douces et des meilleures performances (l'erreur, temps derepense, rapidité et respect des contraintes).Cette commande présente aussi un avantage très intéressant réside dans l'élimination des effets de toute perturbation.

La formation n'est pas est une norme que le réseau à atteint son objectif. Ou la formation peut atteindre une étape avancée et très précis, mais le test est faible.

Nous ne pouvons pas observer exactement les mêmes chiffres ou les mêmes performances, ce n'est pas surprenant. La raison est que newff utilise un générateur de nombres aléatoires pour créer les valeurs initiales des pondérations du réseau. Par conséquent, le réseau initial sera différent même si exactement les mêmes commandes sont utilisées.

2.8 Conclusion

Dans ce chapitre nous avons présenté la conception d'un ANN en utilisant le perceptron multicouche (MLP) pour construire notre modèle par MATLAB et fait une discussion pour les résultats obtenue.

Chapitre 3 : Développement Labview

3.1 Introduction

L'apprentissage automatique (machine learning) est un type d'intelligence artificielle qui confère aux ordinateurs la capacité d'apprendre sans être explicitement programmés. Il consiste en la mise en place d'algorithmes ayant pour objectif d'obtenir une analyse prédictive à partir de données, dans un but précis.

L'organisation NI a établi et développé plusieurs logiciels du plus important et dans les dernières années programme LABVIEW qui traite le domaine d'apprentissage de machine.

3.2 Généralité sur LABVIEW

LABVIEW (Laboratory virtuel Instrument Engineering Workbench) est un environnement de développement de programme comme les langages C ou BASIC. Cependant, LABVIEW diffère de ces applications sur un point important. En effet, alors que les autres outils de programmation emploient des langages textuels, LABVIEW utilise un langage de programmation graphique pour créer des programmes sous la forme de diagrammes.

LABVIEW est un système de programmation complet comprenant différentes bibliothèques permettant d'effectuer toutes les fonctions des programmes traditionnels. De plus, LABVIEW comprend plusieurs bibliothèques avancées permettant, par exemple, de faire de l'acquisition de données et du contrôle d'instruments Pour l'acquisition de données et le contrôle d'instruments, LABVIEW supporte les standards RS-232/422, USB, IEEE488(GPIB) et VXI/PXI, ainsi que les cartes d'acquisition de données.[20]

3.2.1 Les instruments virtuels (Vis)

LABVIEW est fondé sur le langage de programmation graphique G. Ce type de langage, très convivial, simplifie les calculs et les interfaces de communication. Un programme LABVIEW est appelé VI (Virtual Instrument) à cause de sa ressemblance avec les instruments réels. Les VIs sont décomposées en trois parties: la face-avant, le diagramme et l'icône/connecteur.

- La face-avant est principalement utilisée pour les boutons de commandes, les graphiques et les différents types d'indicateurs.
- Le diagramme est le code source du programme. Comme les VIs sont modulaires, il est très facile de diviser le programme en petits modules qui seront reliés entre eux dans le diagramme.
- L'icône/connecteur permet de personnaliser son VI et de pouvoir l'utiliser en tant que sous-VI.

En somme, LABVIEW exploite au maximum la programmation modulaire. À la fin, le VI principal contiendra une série de sous-VI qui réalisent chacun une tâche bien précise.

3.2.2 Environnement LABVIEW

Lorsque nous lançons LABVIEW, la fenêtre suivante apparaît:

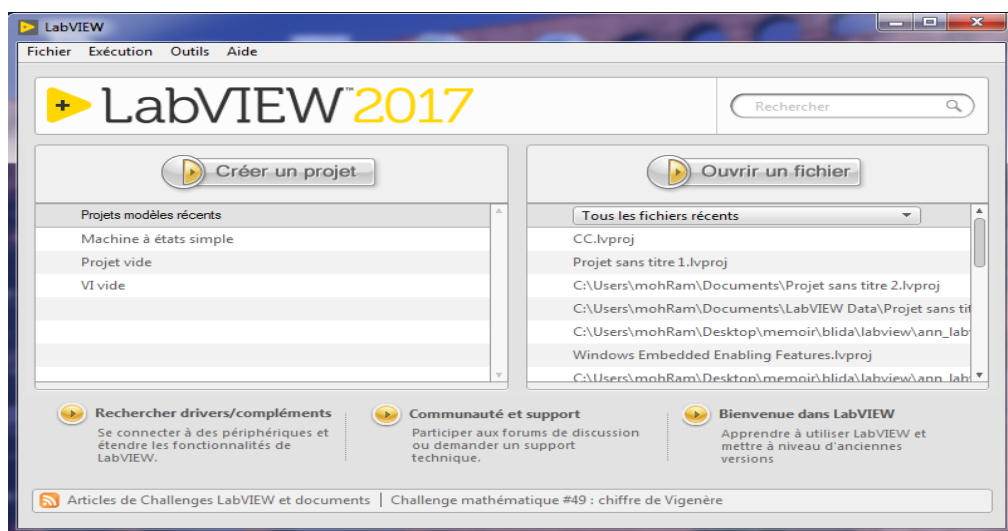


Figure 3. 1 : Environnement LABVIEW 2017.

Pour créer un nouveau VI, il suffit de cliquer sur New VI et deux fenêtres apparaîtront:

La fenêtre avec le fond gris est la face-avant:

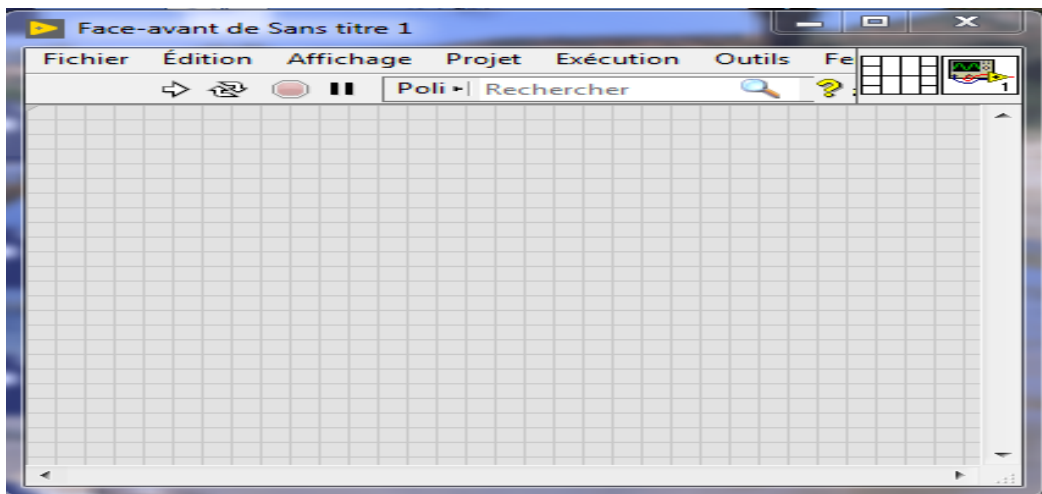


Figure 3. 2 : fenêtre de face-avant.

- La fenêtre avec le fond blanc est le diagramme:

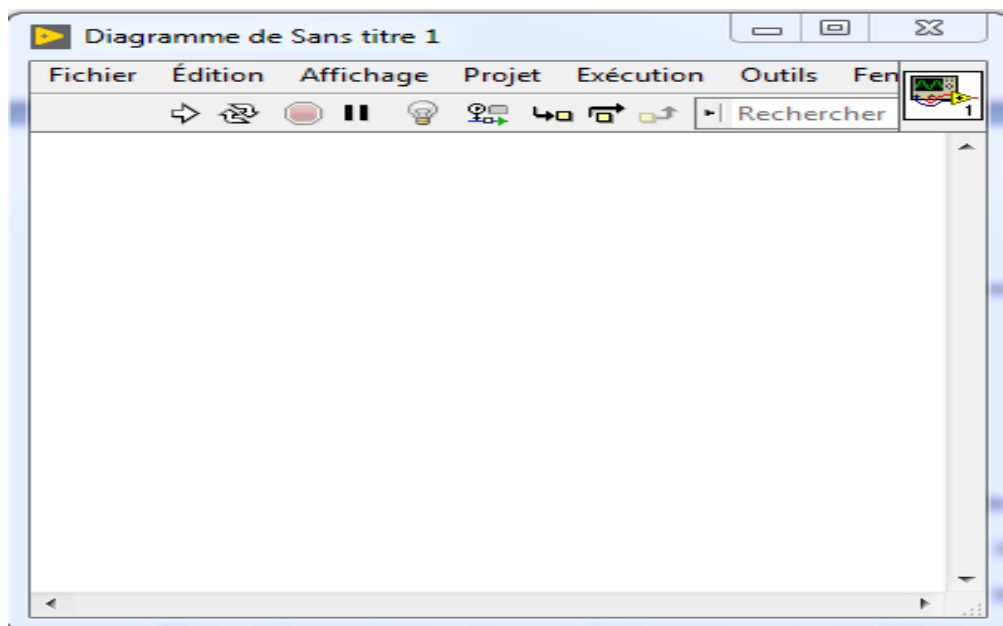


Figure 3. 3 : fenêtre de diagramme.

Nous pouvons également afficher la commandes palette ou la Fonctions palette lorsque et nous cliquons sur le bouton droit de la souris dans la face-avant ou dans le diagramme respectivement. Ces palettes referment toutes les fonctions et tous les contrôles nécessaires pour solutionner un problème. Voici à quoi ressemblent ces palettes:

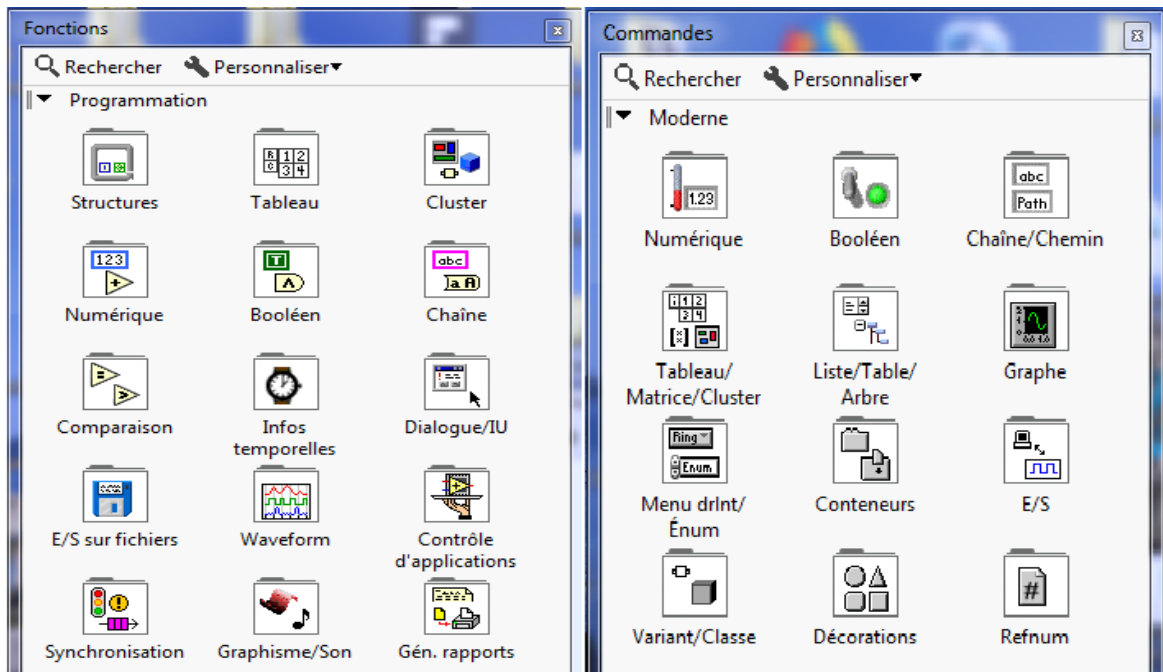


Figure 3. 4 : palette des fonctions et palette des commandes.

3.2.3 Commandes et Indicateurs

Les commandes et les indicateurs sont des objets qui placent dans la face-avant pour modifier ou visualiser les données du VI. Il est également possible de placer des commandes, des constantes et des indicateurs directement depuis le diagramme. Pour ce faire, il suffit de sélectionner l'outil fil de liaison dans la "tools palette", de se placer sur une entrée ou une sortie d'un objet.

3.3 Toolkit analytic and machine learning(AML)

3.3.1 Définition

Machine Learning (boîte à outils d'apprentissage automatique pour LABVIEW) est un produit primé conçu pour intégrer les possibilités d'apprentissage automatique dans LABVIEW et fournir un accès à diverses applications pour les développeurs LABVIEW.

Les programmes d'apprentissage automatique détectent des schémas dans les données et ajustent leur fonctionnement en conséquence.

3.3.2 Caractéristiques et fonctionnalité d'un AML

- ✓ Créer, former et déployer des réseaux de neurones profonds avec LABVIEW.
- ✓ Visualiser la topologie du réseau et afficher des métriques communes.
- ✓ Sauvegardez les réseaux formés et chargez-les pour le déploiement.
- ✓ Prise en charge de la cible temps réel de NI pour le déploiement.

3.3.3 Exemples prêts à l'emploi pour

- Classification des objets.
- Manipulation des fonctionnalités.
- Détection d'anomalies.
- Regroupement des données.
- Data.

3.3.4 Palette de fonctionnement

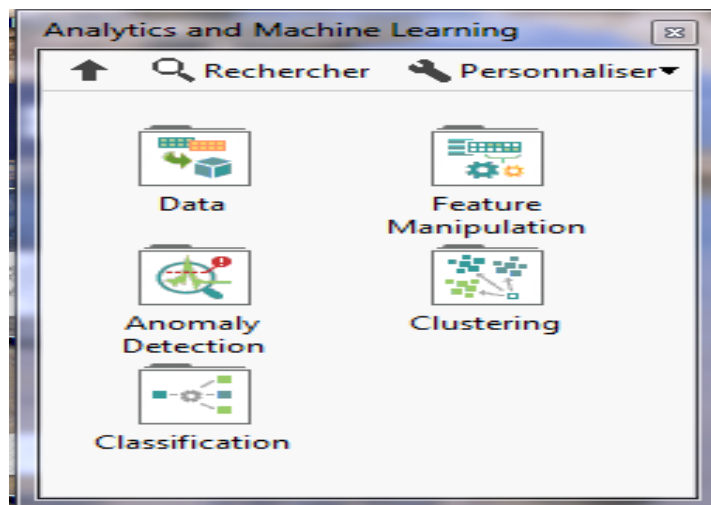


Figure 3. 5 : palette AML.

3.3.5 Rôles des palettes AML

Classification : classer les données.

Feature manipulation : Utilisez les VIs Manipulation de fonctionnalités pour former et déployer des modèles de réduction et de normalisation de fonctionnalités.

La réduction des fonctionnalités réduit la dimension des données afin que vous puissiez appliquer des algorithmes d'apprentissage automatique aux données d'apprentissage. La normalisation normalise la gamme de fonctionnalités pour les données d'apprentissage.

Anomaly detection: Utilisez les VIs Détection d'anomalies pour initialiser, former et déployer des modèles de détection d'anomalies non supervisés détectant des anomalies dans des données non étiquetées.

clustering : cluster pour initialiser, former, évaluer et déployer des modèles de mise en cluster qui regroupent des éléments de données dans des clusters. Les éléments de données dans le même cluster partagent plus de similitudes que ceux d'autres clusters.

Data : Utilisez les VIs de données pour enregistrer des données en vue de la formation ou du déploiement de modèles et pour afficher les modifications apportées aux données à partir des VIs de manipulation des fonctionnalités.

Nous sommes intéressées dans notre étude classification AML.

3.4 VIs de classification

La classification : la variable de sortie prend ses valeurs dans des classes.

On utilise les VIs de classification pour initialiser, former, évaluer et déployer des modèles de classification qui classifient des ensembles de données de catégories :

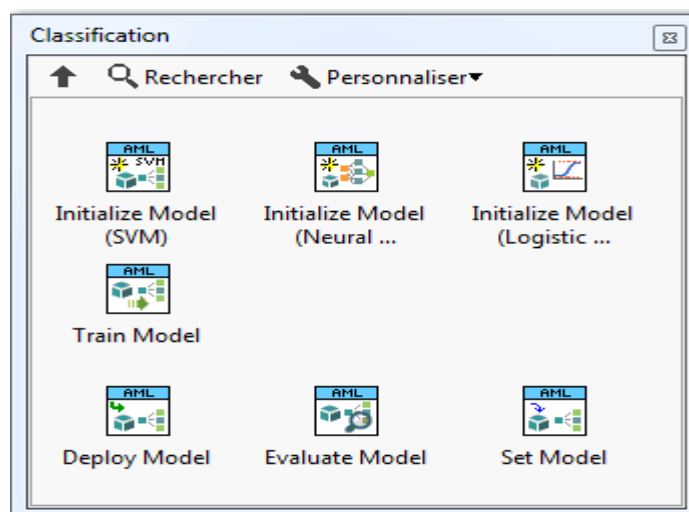


Figure 3.6 : les modèles de classifications.

3.4.1 Déployer le modèle de classification VI

Déploye un modèle de classification formé et renvoie les étiquettes prévues des données d'entrée, selon la palette suivant :

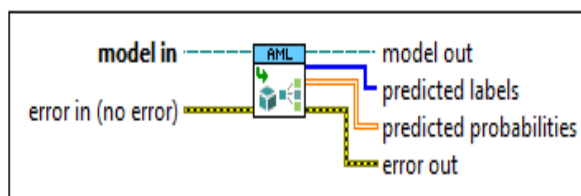


Figure 3. 7 : modèle de déploiement.

Définir les paramètres :

model in : spécifie les informations sur l'ensemble du flux de travail du modèle.

error in : décrit les conditions d'erreur qui se produisent avant l'exécution de ce nœud.

model out : renvoie les informations sur l'ensemble du flux de travail du modèle.

étiquettes prédites renvoie les étiquettes prédites des données pour le déploiement.

les probabilités prédites : renvoient la probabilité prédite pour chaque classe.

erreur en sortie : contient des informations d'erreur.

3.4.2 Évaluer le modèle de classification VI

L'évaluation d'un modèle de classification formé en utilisant de nouvelles données de test avec des étiquettes. Nous chargeons les nouvelles données de test à l'aide de l'instance de déploiement du VI.

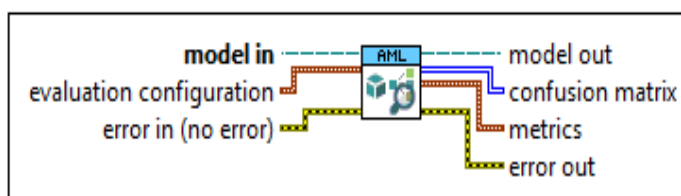


Figure 3. 8 : modèle d'évaluation.

Définir les paramètres :

Model in : spécifie les informations sur l'ensemble du flux de travail du modèle.

Configuration d'évaluation: spécifie la configuration pour la métrique d'évaluation.

La méthode moyenne spécifie la méthode de calcul de moyenne pour ce VI afin de calculer les valeurs métriques pour la classification mutilasse, selon le tableau suivant :

0	Micro (valeur par défaut) - Calcule les valeurs métriques pour chaque échantillon et renvoie la moyenne des valeurs métriques pour tous les échantillons.
1	Macro - Calcule les valeurs métriques pour chaque étiquette et renvoie la moyenne des valeurs métriques pour toutes les étiquettes.
2	Pondéré - Calcule les valeurs métriques pour chaque étiquette et renvoie la moyenne des valeurs métriques pondérées pour toutes les étiquettes. Le nombre de vrais cas dans une étiquette détermine le poids de la valeur métrique de l'étiquette.
3	Binary -(Binaire): calcule les valeurs de métrique de la classe spécifiée par une étiquette positive.

Tableau 3. 1 : méthode de calcul moyenne.

Étiquette positive spécifie la classe pour calculer les valeurs métriques. La Valeur par défaut est 0. Cette entrée est valide uniquement si la méthode moyenne est binaire.

error in : décrit les conditions d'erreur qui se produisent avant l'exécution de ce nœud.

Model out : renvoie les informations sur l'ensemble du flux de travail du modèle.

Matrice de confusion : renvoie la matrice de confusion à partir du résultat de l'évaluation. Une matrice de confusion décrit les performances d'un modèle de classification en signalant le nombre de vrais cas positifs, de vrais négatifs, de faux positifs et de faux négatifs. Chaque ligne d'une matrice de confusion représente la classe réelle et chaque colonne représente la classe prédite.

Par exemple, pour 100 échantillons, il existe deux classes possibles: positive et négative. Le tableau suivant est une matrice de confusion pour les deux classes.

—		Predicted Class	
		Positive	Negative
Actual Class	Positive	65	5
	Negative	19	11

Tableau 3. 2 : exemple d'une matrice de confusion.

La matrice de confusion contient 65 vrais cas positifs, 5 faux négatifs, 19 faux positifs et 11 vrais négatifs.

metrics : renvoie des métriques à partir du résultat de l'évaluation.

Erreur en sortie : contient des informations d'erreur.

3.4.3 Initialiser le modèle de classification (LR) VI

Régression logistique : Une autre méthode d'apprentissage supervisé que nous pouvons considérer est la régression logistique. Elle s'apparente à la régression linéaire car ses objectifs sont les mêmes : comprendre l'influence des variables indépendantes sur la variable dépendante et prévoir les valeurs futures de la variable dépendante à partir des variables indépendantes.

Pour Initialiser les hyperparamètres de l'algorithme de régression logistique (LR). Nous pouvons définir les hyperparamètres ou spécifier plusieurs valeurs pour chaque hyperparamètre.

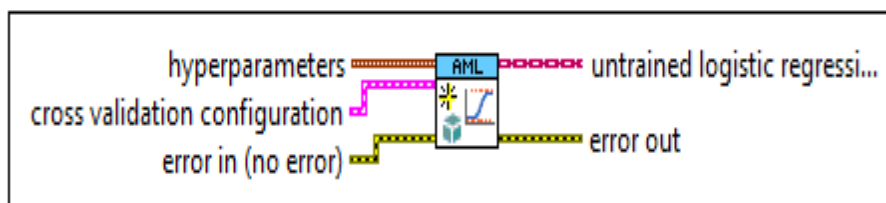


Figure 3. 9 : modèle de classification (LR).

Définir les paramètres :

hyperparameters : spécifie les paramètres du modèle de régression logistique. Ou la tolérance spécifie pour les critères d'arrêt. La valeur par défaut est 0,0001. L'ajustement du modèle s'interrompt si le changement d'erreur d'apprentissage entre deux itérations d'optimisation successives est inférieur à la tolérance. max itération spécifie le nombre maximal d'itérations d'optimisation pour les critères d'arrêt. La valeur par défaut est 1000. L'ajustement du modèle s'arrête si le nombre d'itérations d'optimisation atteint l'itération maximale.

La configuration de validation croisée : spécifie les paramètres de validation croisée.

Nombre de plis spécifie le nombre de sections dans lesquelles ce VI divise les données d'apprentissage. La valeur par défaut est 3. Une section est constituée de données de test et les autres sections sont des données d'apprentissage. Le VI Modèle d'évaluation de la classification répète le processus de validation croisée n fois, où n est égal au nombre de plis.

Configuration de la métrique spécifie la configuration de la métrique d'évaluation dans la validation croisée.

error in : décrit les conditions d'erreur qui se produisent avant l'exécution de ce nœud.

Modèle de régression logistique non entraîné renvoie le modèle de régression logistique initialisé pour la formation.

erreur en sortie: contient des informations d'erreur.

L'équation suivante définit le modèle de régression logistique:

$$p(\mathbf{x}) = p(y = 1|\mathbf{x}) = \frac{1}{1 + \exp(-(\beta_0 x_0 + \beta_1 x_1 + \dots + \beta_k x_k))}$$

Où :

x : est l'entrée avec k + 1 dimensions. x₀ est toujours 1.

Y : est la sortie avec une valeur binaire 0 ou 1.

B : est le vecteur de poids avec k + 1 dimensions.

P (y = 1 | x) : est la probabilité de y = 1 étant donné un exemple connu de x.

3.4.4 Initialiser le modèle de classification (NN) VI

Pour initialiser les hyperparamètres de l'algorithme de réseau neuronal (NN). Nous pouvons soit définir directement les hyperparamètres, soit spécifier plusieurs valeurs pour chaque hyperparamètre. Ce VI prend en charge uniquement les couches masquées.

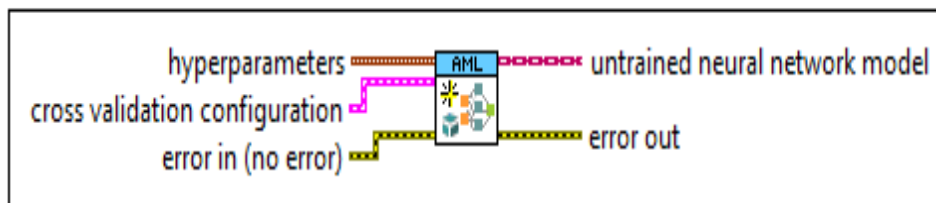


Figure 3. 10 : modèle de classification (NN).

Définir les paramètres :

Les hyperparameters spécifie les paramètres du modèle de réseau neuronal, Nombre de neurones cachés spécifie le nombre de neurones dans la couche cachée, Type de couche cachée spécifie le type de fonction d'activation dans les neurones de la couche cachée.

Function type	Definition	Description
Sigmoid function	$f(x) = \frac{1}{1 + e^{-x}}$	x is the activation value of the hidden neuron.
Tanh function	$f(x) = \tanh (x)$	x is the activation value of the hidden neuron.
Rectified linear unit function	$f(x) = \max (0, x)$	x is the activation value of the hidden neuron.

Tableau 3. 3 : fonction d'activation de couche caché.

Le type de couche de sortie spécifie le type de fonction d'activation dans les neurones de la couche de sortie.

Function type	Definition	Description
Sigmoid function	$f(x) = \frac{1}{1 + e^{-x}}$	x is the activation value of the output neuron.
Softmax function	$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{-x_j}}$	x is the activation value of the output neuron and n is the number of classes.

Tableau 3. 4 : fonction d'activation de couche sortie.

La tolérance spécifie pour les critères d'arrêt. La valeur par défaut est 0,0001. L'ajustement du modèle s'interrompt si le changement d'erreur d'apprentissage entre deux itérations d'optimisation successives est inférieur à la tolérance, max itération spécifie le nombre maximal d'itérations d'optimisation pour les critères d'arrêt. La valeur par défaut est 1 000. L'ajustement du modèle s'arrête si le nombre d'itérations d'optimisation atteint l'itération maximale.

La configuration de validation croisée spécifie les paramètres de validation croisée.

configuration de la métrique spécifie la configuration de la métrique d'évaluation dans la validation croisée.

La méthode moyenne spécifie la méthode de calcul de moyenne pour ce VI afin de calculer les valeurs métriques pour la classification multiclasse.

error in : décrit les conditions d'erreur qui se produisent avant l'exécution de ce nœud.

Modèle de réseau neuronal non entraîné renvoie le modèle de réseau neuronal initialisé pour la formation.

Erreur en sortie : contient des informations d'erreur.

3.4.5 Initialiser le modèle de classification (SVM) VI

Les SVM (support vector machine) sont un ensemble de techniques d'apprentissage supervisé de nouvelle génération basés sur les progrès récents de la théorie de l'apprentissage statistique, ils sont basés sur le principe de la maximisation de la marge (séparation des classes). [21]

Les SVM constituent la forme la plus connue parmi les méthodes à noyaux. Ce sont des classificateurs qui reposent sur deux idées clés : [22] [23]

a **La notion de marge maximale**

Il existe un unique hyperplan optimal permettant de résoudre ce problème. Cet hyperplan optimal, est défini comme étant l'hyperplan qui maximise la marge entre les échantillons et l'hyperplan séparateur. Ce choix est justifié théoriquement par le fait que la capacité des classes d'hyperplans séparateurs diminue lorsque leur marge augmente. La marge est la plus petite distance entre les échantillons d'apprentissage et l'hyperplan séparateur qui satisfait la condition de séparabilité. Ces échantillons sont appelés vecteurs supports.

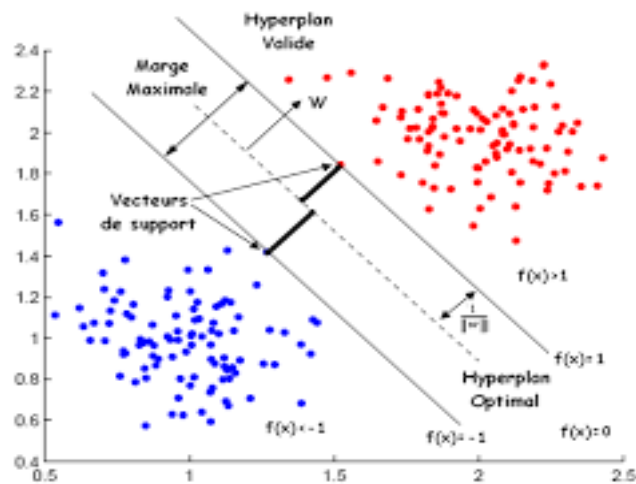


Figure 3. 11 : Maximisation de la marge pour obtenir l'hyperplan optimal.

b **Fonction noyau (kernel)**

La transformation de l'espace de représentation des données d'entrées en un espace de plus grande dimension F , dans lequel il est probable qu'il existe une séparatrice linéaire, afin de pouvoir traiter des cas où les données ne sont pas linéairement séparables. Les fonctions du noyau sont utilisées pour cartographier efficacement les données d'entrée qui peuvent ne pas être séparées linéairement en un espace de grande dimension, où des méthodes linéaires peuvent alors être appliquées.

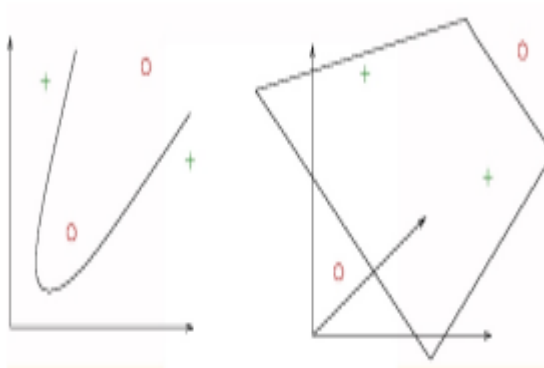


Figure 3.12: La transformation non linéaire.

Pour Initialiser les hyperparamètres de l'algorithme de machine à vecteurs de support (SVM). Nous pouvons soit définir directement les hyperparamètres, soit spécifier plusieurs valeurs pour chaque hyperparamètre.

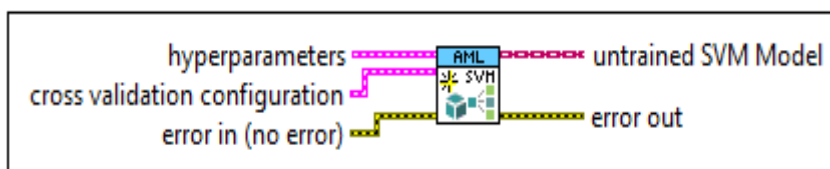


Figure 3.13 : modèle de classification (SVM).

Définir les paramètres :

hyperparameters spécifie les paramètres du modèle SVM.

Le type SVM spécifie le type du modèle SVM.

1	C_svm (valeur par défaut)- spécifie le modèle C-SVM.
2	NU_svm: spécifie le modèle nu-SVM.

Tableau 3.5 : les types modèles SVM.

Le paramètre c spécifie pour le modèle C-SVM. La valeur par défaut est 1. Plus la valeur de c est basse, plus le modèle SVM permet de sélectionner des échantillons d'apprentissage mal formés. Le paramètre c doit être supérieur à 0. Cette entrée est valide uniquement si le type de SVM est C.

Le paramètre nu est à la fois une limite inférieure pour le nombre d'échantillons qui sont des vecteurs de support et une limite supérieure pour le nombre d'échantillons qui se trouvent du mauvais côté de l'hyperplan. La valeur par défaut est 0.5. Le paramètre nu doit être compris dans la plage [0,1]. Cette entrée est valide uniquement si le type de SVM est NU.

Par exemple, si nu vaut 0,05, les échantillons d'apprentissage mal classés ne peuvent contenir plus de 5% de tous les échantillons d'apprentissage. En outre, au moins 5% des échantillons d'apprentissage sont des vecteurs de support.

les paramètres du noyau permettant de configurer la fonction du noyau.

Ce VI prend en charge les fonctions du noyau suivantes:

Kernel function	Definition	Description
Linear function	$K(x, y) = x^T y$	x and y are the input sample vectors.
Polynomial function	$K(x, y) = (\text{gamma} * x^T y + \text{coef0})^{\text{degree}}$	x and y are the input sample vectors; gamma , coef0 , and degree are the algorithm coefficients.
Radial basis function	$K(x, y) = \exp(-\text{gamma} * x - y ^2)$	x and y are the input sample vectors.
Sigmoid function	$K(x, y) = \tanh(\text{gamma} * x^T y + \text{coef0})$	x and y are the input sample vectors; gamma and coef0 are the algorithm coefficients.

Tableau 3. 6 : les fonctions du noyau (kernel).

Degré : spécifie le coefficient de degré de la fonction du noyau. La valeur par défaut est 3.

Gamma : spécifie le coefficient gamma de la fonction du noyau. La valeur par défaut est 1.

coef0 : spécifie le coefficient coef0 de la fonction du noyau. La valeur par défaut est 0.

c pondéré spécifie différentes pondérations du paramètre c pour différentes classes dans les données d'apprentissage. On Utilise cette entrée lorsque les données d'entraînement sont déséquilibrées. Dans les données non équilibrées, le nombre d'échantillons pour certaines classes est beaucoup plus grand que celui des autres classes.

La configuration de validation croisée spécifie les paramètres de validation croisée.

Le nombre de plis (couche) spécifie le nombre de sections dans lesquelles ce VI divise les données d'apprentissage.

La configuration de la métrique spécifie la configuration de la métrique d'évaluation dans la validation croisée.

error in : décrit les conditions d'erreur qui se produisent avant l'exécution de ce nœud.

modèle SVM non entraîné : retourne le modèle SVM initialisé pour la formation.

Erreur en sortie : contient des informations d'erreur.

3.4.6 Définir le modèle de classification VI

La Définition des propriétés d'un modèle de classification formé avant le déploiement, consisté sur la définition des trois VI (NN, SVM, LR) :

a SVM

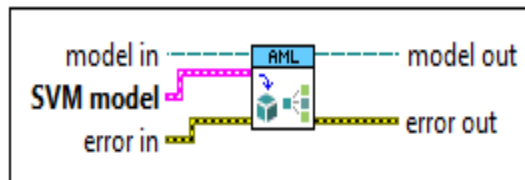


Figure 3. 14 : modèle de définition SVM.

Model in : spécifie les informations sur l'ensemble du flux de travail du modèle.

Le modèle SVM : spécifie les informations sur le modèle SVM formé.

L'équation suivante définit la fonction de décision du modèle SVM:

$$\text{sgn}\left(\sum_{i=1}^l y_i \alpha_i K(x_i, x) - \rho\right)$$

Où :

sgn : est la fonction de signe.

L : est le nombre de vecteurs de support.

y_i : est l'étiquette du vecteur support.

A_i : est le coefficient du vecteur support.

x_i : est le vecteur de support.

X : est la donnée d'entrée.

$K(x_i, x)$: est la fonction du noyau de x_i et x . Les types de fonctions du noyau disponibles sont les suivants: linéaire, polynomial, base radiale et sigmoïde.

P : est la constante de la fonction de décision

error in : décrit les conditions d'erreur qui se produisent avant l'exécution de ce nœud. Cette entrée fournit une erreur standard dans la fonctionnalité.

Modèle out renvoie les informations sur l'ensemble du flux de travail du modèle.

Erreur en sortie contient des informations d'erreur.

b Réseau neuronal

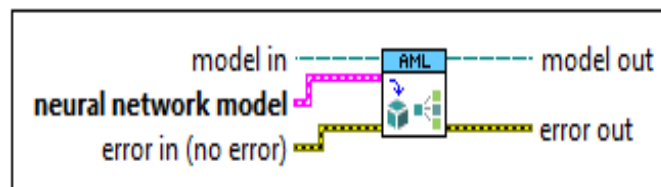


Figure 3. 15 : modèle de définition NN.

Model in : spécifie les informations sur l'ensemble du flux de travail du modèle.

Modèle de réseau neuronal : spécifie les informations sur le modèle de réseau neuronal formé.

error in : décrit les conditions d'erreur qui se produisent avant l'exécution de ce nœud.

Model out : renvoie les informations sur l'ensemble du flux de travail du modèle.

Erreur en sortie : contient des informations d'erreur.

c *Régression logistique*

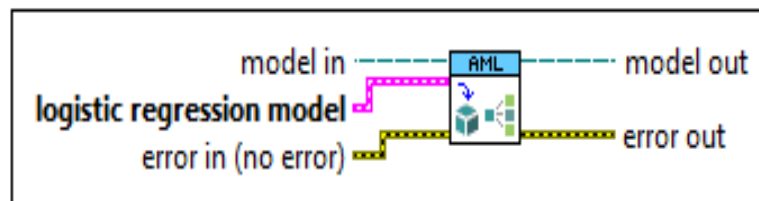


Figure 3. 16 : modèle de définition LR.

Modèle d'entrée : spécifie les informations sur l'ensemble du flux de travail du modèle.

Modèle de régression logistique : spécifie les informations relatives au modèle de régression logistique formé.

Erreur en entrée : décrit les conditions d'erreur qui se produisent avant l'exécution de ce nœud.

Model out : renvoie les informations sur l'ensemble du flux de travail du modèle.

Erreur en sortie : contient des informations d'erreur. Cette sortie fournit une fonctionnalité de sortie d'erreur standard.

3.4.7 **Modèle de classification de train VI**

Entraîne un modèle model in spécifie les informations sur l'ensemble du flux de travail du modèle.

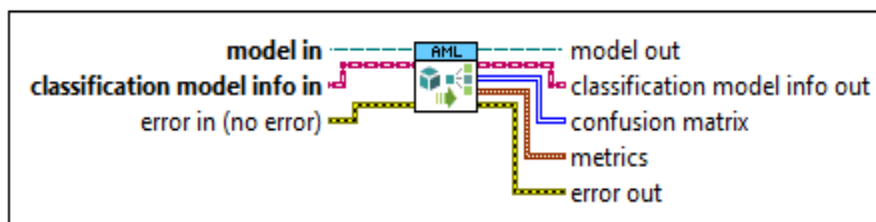


Figure 3. 17 : modèle de train.

Modèle de classification info in : spécifie le modèle de classification initialisé pour la formation.

Nous pouvons acquérir un modèle de classification initialisé à partir des VIs suivants:

-initialiser le modèle de classification (LR) VI.

-Initialiser le modèle de classification (NN) VI.

-initialiser le modèle de classification (SVM) VI.

error in : décrit les conditions d'erreur qui se produisent avant l'exécution de ce nœud.

Model out : renvoie les informations sur l'ensemble du flux de travail du modèle

Modèle de classification info out : renvoie le modèle de classification formé.

Matrice de confusion : renvoie la matrice de confusion à partir du résultat de l'évaluation.

Metrics : renvoie des métriques à partir du résultat de l'évaluation.

Erreur en sortie : contient des informations d'erreur. Cette sortie fournit une fonctionnalité de sortie d'erreur standard.

3.5 EXEMPLE D'Implémentation

3.5.1 Introduction

On prend un échantillon de la température composée de quatre valeurs et nous mettons une classification de leur sorties où cet échantillon est entrainer à l'aide d'un architecture de réseau de neurone artificielle, le type d'apprentissage est supervisée.

a **Face diagramme**

Dans le bloc diagramme nous présentons le code source de programme.

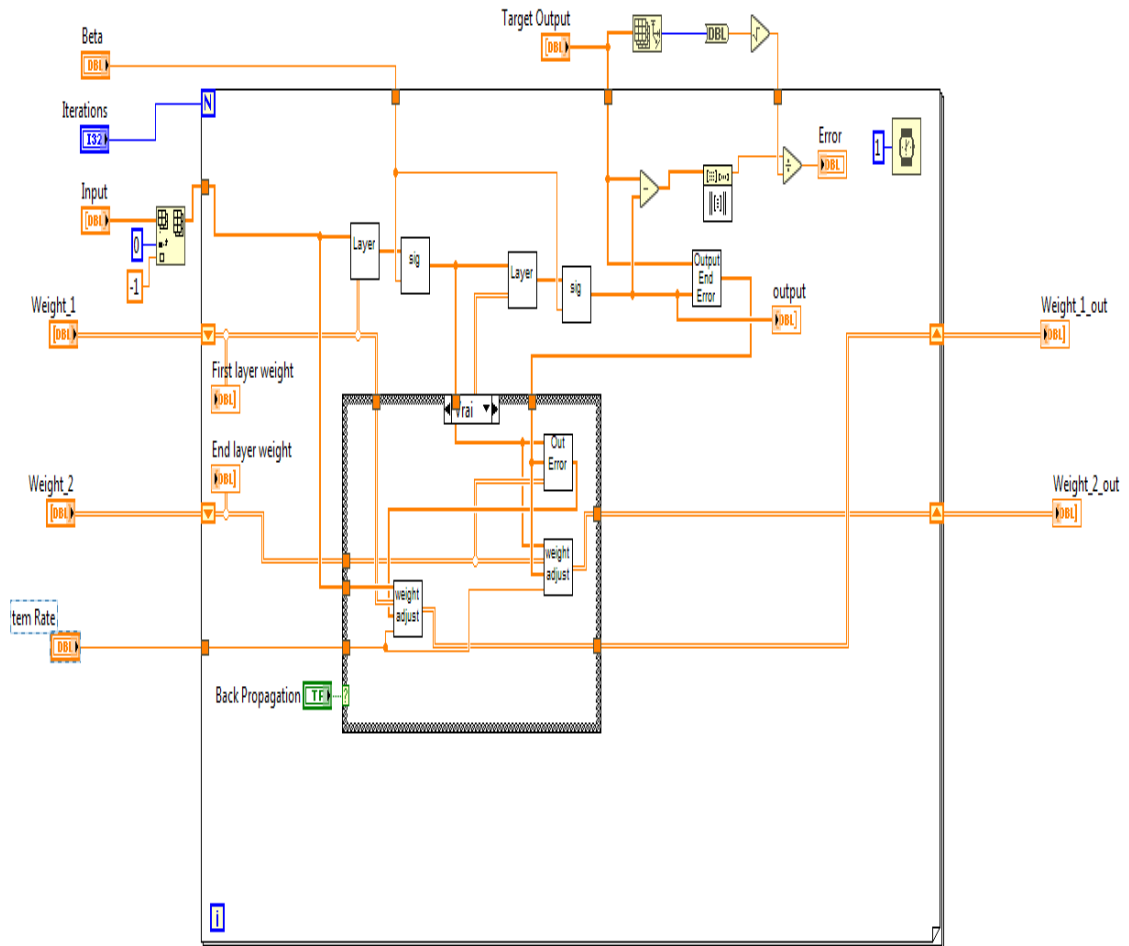


Figure 3. 18 : face diagramme.

b **Face avant**

Dans la face avant nous présentons les différents types des commandes et les indicateurs selon le figure 3.19 :

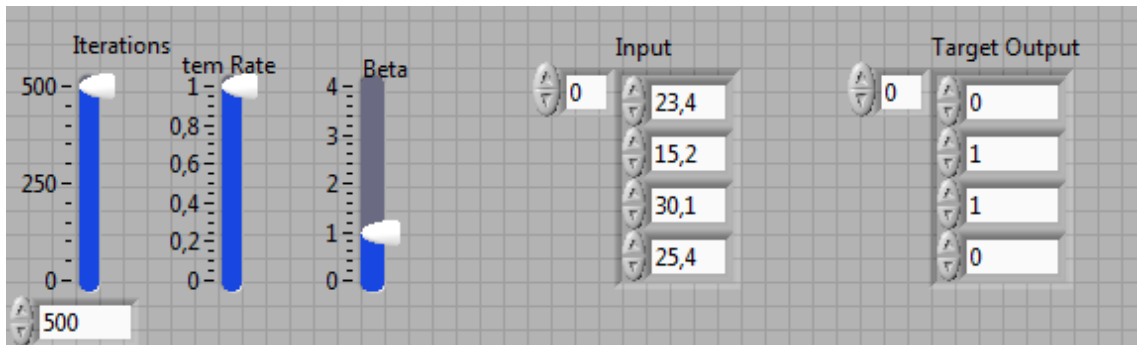


Figure 3. 19 : face avant.

c **Résultat**

Lorsqu'on exécute le programme on obtient les résultats suivantes :

Les sorties réelles :

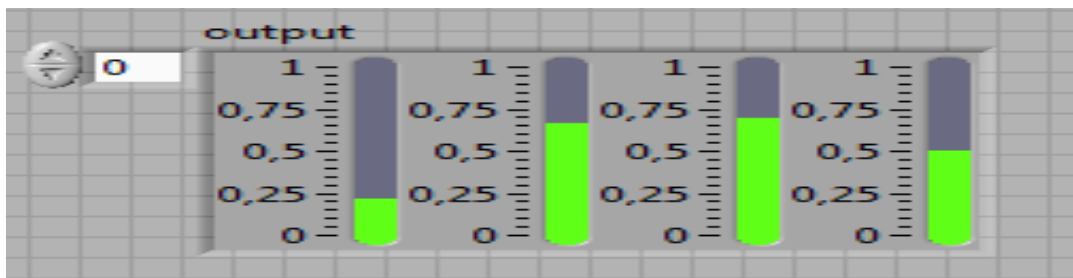


Figure 3. 20 : Résultat de sortie.

➤ Les poids :

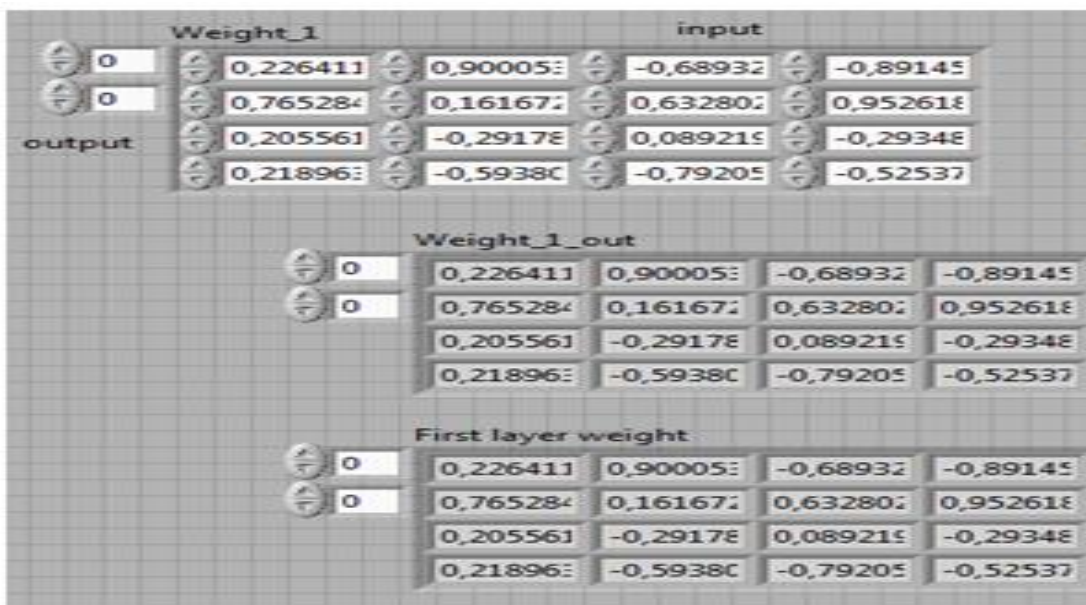


Figure 3. 21 : poids de neurone.

➤ Performance :



Figure 3. 22: Résultat de performance en sortie.

3.5.2 Discussion

Avant de construire une architecture d'un ANN il faut régler les paramètres du modèle de réseau neuronal (beta, nombre d'itération : max itération spécifie le nombre maximal d'itérations d'optimisation pour les critères d'arrêt, time rate, erreur en entrée qui décrit les conditions d'erreur qui se produisent avant l'exécution de ce nœud). Ce modèle contient 4 entrées d'une température d'une commande numérique et deux couches cachées chacune contient 4 neurones. Les poids sont les connexions forts entre les neurones dans les couches adjacentes, sont les poids associés à chacune des connexions entre les entrées à un neurone et le neurone suivant. Les entrées à un neurone sont pondérées selon le type de l'architecture neurale et les différents types d'entrées utilisées. S'appuie il existe 4 caractéristiques appliquées comme vecteur de données d'entrées et qu'il a 4 neurone cachés dans le net alors les poids entre la couche d'entrée et la couche caché sont représentés par une matrice de 4×4 . On applique une fonction d'activation de types sigmoïde dans les deux couches cachées. Ensuite on fournit un indicateur numérique dans la sortie (sortie désirée) pour valider l'apprentissage supervisée. Ou chaque sortie correspond une classe (0 ou bien 1) de leur entrée.

Lorsqu'on terminée la construire d'un architecture ANN nous lançons l'exécution du ce modèle on obtient des sorties réel une forme des classes. Ces sorties sont prennent des valeurs réelles allant entre 0 et 1 (0.25, 0.7, 0.7, 0.5), et aussi on obtient l'erreur en

sortie qui contient l'information de performance de test ou erreur est égale à 0.35 c'est très grand donc il y a un mauvais apprentissage pour cela il faut améliorer la performance par augmenter le nombre d'entrée.

3.6 Conclusion

Dans ce chapitre nous avons étudiée l'environnement et les toolkit de l'apprentissage automatique et son fonctionnement (classification, détection...), et enfin on a fait un exemple qui simuler les sortie d'une température en utilisant le modèle ANN sur LABVIEW.

Conclusion générale

Dans ce mémoire, nous avons présenté un travail sur la détection d'anomalies dans un service de radiothérapie, en utilisant le capteur MS6503 des trois grandeurs physique à savoir : température et l'humidité et la pression. Pour d'améliorer la qualité de leurs services de traitement de radiothérapie puisqu'il s'agit de la vie des patients, ce qui nécessite une vigilance et une surveillance permanentes. Cette méthode présentait certaines erreurs et n'était valide que dans un intervalle précis. Grâce à notre contribution, nous avons pu réduire encore plus l'erreur de détection en utilisant une méthode basée sur l'intelligence artificielle en décryptant les données par lesRN. Nous avons utilisée deux méthodes de contrôle MATLAB et LABVIEW.

Dans un premier temps, on à étudiée Les réseaux de neurones qui sont une nouvelle technique de traitement de l'information. Concrètement ils se traduisent par des algorithmes mettant en jeu des concepts associés à la nature du cerveau pour la notion d'apprentissage.

La deuxième tâche consiste à développer un d'algorithme d'apprentissage sous MATLAB (premier méthode) en utilisant un réseau de neurones de type MLP où l'information se propage de l'entrée vers la sortie sans retour en arrière. Étant donné que nous sommes intéressés à la méthode de prévision et classification selon les types d'anomalies. On a utilisé une base de données (T, H, P) Enregistré par l'ingénieur du CDTA, sur une période allant du décembre 2014 à janvier 2016.

Cette base de données a été divisée en deux sous ensembles, le premier sert à effectuer l'apprentissage, et le deuxième sous ensemble utilisé pour tester les performances, c'est ce qu'on appelle l'ensemble test.

Nous pouvons conclure que le MLP donne des résultats raisonnables quand on l'applique pour une tâche de classification.

La tâche principale de ce mémoire c'est le développement de notre programme par LABVIEW pour pouvoir classifier les données; température, humidité et la pression de la salle de traitement dans des sorties prédites, en utilisant le technique du RN.

Dans cette méthode utilisée (LABVIEW), nous avons trouvé certain difficultés concernant l'utilisation de logiciel LABVIEW à cause de manque la formation et les formateurs dans le domaine d'intelligence artificielle pour cela on a prend un simple exemple pour la classification de température en utilisant la structure du réseau de neurone.

Ce travail pratique été très bénéfique pour nous, et nous a permis d'apprendre plusieurs nouvelles approches telles que l'utilisation des réseaux de neurones dans le domaine de contrôle, l'utilisation du Labview ... etc. Ce travail nous a permis d'appliquer en pratique ce que nous avons appris en théorie durant toutes les années de formation et nous souhaitons que ce travail soit exploité par les futurs étudiants désirant travailler sur cet axe de recherche tel que la conception d'un système expert afin d'améliorer la qualité dans les services de traitement de cancer et même les autres services de la santé (projetCDTA).

Annexes

Ci-joint les données.

0 = pas d'anomalies

1 = anomalies graves (ne pas traiter)

0 limite = traiter avec anomalies (il faut vérifier ou réparer le système)

Capteur MS6503, humidité et température utilisé

- **Humidité**

Range	Résolution	Accuracy
0 – 100%RH	0.1%RH	+/- 2.5RH at 25°C

Spécification électrique

Transducteur utilisé : **HIH-3610 (Honeywell)**

- **Température**

Range	Résolution	Accuracy
-20°C ~ +60°C	0.1°C	+/- 0.7°C
-4°F ~ +140°F	0.1°F	+/- 1.4°F

Spécification électrique

Transducteur utilisé : **LM335Z (National Semiconductor Corporation)**

Maintenez la température de l'eau entre **15 et 26° C** (60 et 80 F) (page 273)

La température de l'eau interne ne doit pas dépasser 48°C.

- **Capteur de pression**

Mesure de la pression de l'eau de refroidissement des machines (accélérateurs linéaires)

La pression de l'eau ne doit pas dépasser **100 psi** (page 273).

les réseaux de distribution urbains conservent une pression variant entre **40 et 70psi** (pression statique). La plupart des systèmes d'alimentation privés maintiennent des pressions variant de **20 à 40psi**. Cela s'avère parfois insuffisant et un plombier peut parfois régler certains systèmes à la hausse. Cela ne fonctionne, bien entendu, que si la pompe peut offrir une plus haute pression. La moyenne est de l'ordre de **60 psi**.

Actions prendre

Causes possibles

- Le niveau d'eau dans le réservoir est insuffisant.
- La température du système d'eau dépasse 48±2°C.
- La température de l'eau dans l'échangeur de chaleur est trop élevée.
- L'eau courante (eau de ville) est coupée.
- Le contrôleur Barber Coleman est défectueux : cela peut provenir soit de l'unité de contrôle de la température, soit de l'asservissement de la vanne.
- Surcharge thermique de la pompe à eau.

Enregistrement de l'humidité, pression et de la température: décembre 2014 à janvier 2016

N°	Température		Humidité		Pression	A
	CPMC1	CPMC2	CPMC1	CPMC2		
1	24.3	23.7	37.3	39.3	50	0
2	25.4	24.2	34.5	37.4	49	0
3	25.5	24.3	34.7	37.2	49	0

4	25.5	24.4	34.8	37.0	49	0
5	25.5	24.5	34.9	37.1	48	0
6	25.3	24.5	38.6	40.6	48	0
7	26.1	24.6	38.9	41.9	45	0
8	26.1	24.9	38.9	41.6	45	0
9	23.9	22.9	52.4	54.6	50	0
10	23.9	23.0	52.4	54.0	51	0
11	23.9	23.0	52.8	53.8	52	0
12	24.0	23.3	52.6	52.7	52	0
13	24.1	23.4	52.4	51.5	55	0
14	25.0	23.8	51.0	51.5	55	0
15	25.0	24.2	50.7	51.2	56	0
16	24.9	23.2	49.3	52.2	55	0
17	24.9	23.8	49.3	51.7	55	0
18	24.8	23.9	49.6	51.2	54	0
19	24.8	23.9	49.5	51.2	54	0
20	24.9	24.0	49.4	51.3	53	0
21	25.2	23.7	48.6	52.0	51	0
22	24.7	24.2	45.8	46.8	50	0
23	24.7	24.2	45.7	47.2	52	0
24	25.0	24.4	45.1	46.5	49	0

25	25.1	24.5	45.1	46.4	49	0
26	25.3	24.7	44.7	46.2	49	0
27	25.4	24.7	44.6	46.4	48	0
28	25.3	24.5	42.7	45.0	48	0
29	25.5	24.6	42.4	44.6	48	0
30	25.6	24.7	42.4	44.6	47	0
31	25.7	25.1	42.1	44.1	47	0
32	25.7	25.1	41.9	44.1	46	0
33	25.7	25.2	42.0	44.2	45	0
34	25.7	25.2	42.0	44.3	45	0
35	25.6	25.2	42.2	44.1	45	0
36	21.0	21.4	48.9	47.1	55	0
37	21.0	21.5	49.1	46.9	55	0

38	21.0	21.6	48.6	46.3	58	0
39	21.4	21.5	48.8	47.5	58	0
40	21.4	21.6	48.5	46.8	58	0
41	21.5	21.4	47.0	46.4	59	0
42	21.5	21.6	47.3	46.2	59	0
43	21.9	21.5	46.2	46.0	59	0
44	21.9	21.9	46.2	45.5	60	0

45	21.9	22.0	45.8	44.9	60	0
46	22.2	22.4	47.1	45.5	57	0
47	22.2	22.4	47.0	45.6	57	0
48	22.2	22.5	46.6	45.2	58	0
49	22.3	22.5	46.4	45.5	56	0
50	22.3	22.3	46.5	45.4	57	0
51	22.7	22.3	46.1	46.0	58	0
52	22.7	22.6	46.0	45.5	58	0
53	22.7	22.3	46.0	45.5	58	0
54	22.7	22.8	46.1	45.4	58	0
55	20.5	20.3	77.5	64.4	60	0
56	20.5	20.5	77.2	64.3	62	0
57	20.7	20.6	77.2	64.1	61	0
58	20.9	20.8	77.2	63.7	62	0
59	21.6	21.1	75.7	64.0	59	0
60	21.5	21.1	75.8	63.9	59	0
61	21.5	21.1	76.3	63.7	59	0
62	21.5	21.1	76.5	63.6	58	0
63	21.4	21.2	76.7	63.6	58	0
64	21.4	21.2	76.8	63.4	57	0
65	21.7	21.5	74.5	62.1	58	0

66	21.7	21.5	74.7	62.0	58	0
67	21.7	21.5	74.4	61.5	57	0
68	21.7	21.5	74.3	61.5	57	0
69	21.8	21.6	73.9	61.2	56	0
70	21.8	21.6	74.1	61.3	55	0
71	21.8	21.6	74.0	61.3	55	0
72	21.8	21.6	73.9	61.0	56	0
73	21.8	21.6	73.8	61.1	54	0
74	21.8	21.6	73.9	61.1	54	0

75	21.8	21.6	73.8	61.1	57	0
76	21.0	20.5	66.1	58.5	59	0
77	21.1	20.7	65.8	58.1	58	0
78	21.3	21.0	65.2	57.3	58	0
79	21.3	21.1	64.9	57.2	58	0
80	21.4	21.1	64.8	57.2	57	0
81	21.4	21.2	64.8	57.2	56	0
82	21.5	21.4	63.7	56.3	58	0
83	21.7	21.5	63.7	56.2	57	0
84	21.8	21.6	63.5	56.2	58	0
85	21.8	21.7	63.6	56.4	58	0

86	21.9	21.8	63.4	55.8	56	0
87	22.1	21.4	63.0	57.2	57	0
88	22.1	21.8	63.0	56.5	54	0
89	22.3	22.1	62.7	55.8	53	0
90	22.3	22.2	62.8	55.4	54	0
91	22.5	22.4	62.5	55.4	53	0
92	22.6	22.6	62.3	55.1	52	0
93	22.6	22.0	62.0	56.5	52	0
94	22.7	22.4	61.9	55.5	52	0
95	22.8	22.8	66.2	54.5	53	0
96	22.9	22.8	61.0	54.6	51	0
97	22.9	22.7	61.3	54.6	51	0
98	22.9	22.8	61.3	54.8	50	0
99	22.9	22.8	61.2	54.6	50	0
100	23.0	22.8	60.9	54.4	49	0
101	23.0	22.9	60.7	54.4	49	0
102	23.0	22.8	60.7	54.7	48	0
103	23.0	22.8	60.6	54.7	47	0
104	23.0	22.9	61.0	54.4	48	0
105	23.0	22.9	60.6	54.4	49	0
106	23.0	22.9	60.6	54.4	47	0

107	23.1	23.0	60.1	54.3	45	0
108	23.1	23.0	60.5	54.4	45	0
109	23.2	23.1	60.4	54.1	44	0
110	23.2	23.1	60.0	54.1	43	0
111	23.2	23.1	60.0	54.1	43	0
112	21.8	21.6	73.8	61.1	57	0
113	21.0	20.5	66.1	58.5	58	0
114	21.1	20.7	65.8	58.1	58	0
115	21.3	21.0	65.2	57.3	57	0
116	21.3	21.1	64.9	57.2	57	0
117	21.4	21.1	64.8	57.2	56	0
118	21.4	21.2	64.8	57.2	54	0
119	21.5	21.4	63.8	56.4	54	0
120	21.6	21.4	63.8	56.5	53	0
121	23.2	23.1	59.9	54.1	51	0
122	23.4	23.3	57.4	52.8	50	0
123	23.4	23.3	57.6	52.9	50	0
124	23.4	23.3	57.7	53.0	49	0
125	23.5	23.4	57.3	52.8	49	0
126	23.4	23.4	57.5	53.0	48	0
127	23.4	23.3	57.8	53.1	48	0

128	23.6	23.4	57.5	52.9	48	0
129	23.6	23.4	57.1	53.0	47	0
130	23.6	23.4	57.4	53.0	47	0
131	23.7	23.6	57.2	52.6	46	0
132	23.7	23.5	56.0	52.6	46	0
133	23.7	23.5	56.2	52.6	46	0
134	23.8	23.7	56.0	52.0	46	0
135	23.8	23.7	55.4	52.1	46	0
136	23.8	23.6	55.7	52.2	45	0
137	23.8	23.6	55.9	52.4	45	0
138	23.8	23.5	55.9	52.4	44	0
139	23.8	23.5	55.8	52.4	44	0
140	23.8	23.7	57.0	53.1	43	0
141	23.8	23.6	57.5	53.3	43	0
142	23.8	23.6	57.0	53.0	43	0
143	22.5	22.1	57.4	54.0	47	0
144	22.5	22.2	57.2	53.8	48	0
145	22.3	22.3	57.5	53.8	48	0
146	22.5	22.3	57.2	53.7	47	0
147	23.0	22.7	55.9	53.1	41	0
148	23.0	22.8	61.0	55.8	41	0

149	23.0	22.8	61.6	56.0	45	0
150	23.0	22.8	61.5	55.8	44	0
151	23.0	22.8	61.3	55.8	48	0
152	23.0	22.8	61.4	55.7	48	0
153	23.0	22.8	61.2	55.8	47	0
154	23.0	22.8	61.3	55.7	44	0
155	23.0	22.7	60.3	55.3	43	0
156	23.0	22.7	59.8	55.1	42	0
157	23.1	22.9	59.6	55.1	43	0
158	23.3	23.0	60.5	55.8	42	0
159	23.4	23.0	60.5	55.8	42	0
160	23.4	23.2	61.7	56.1	41	0
161	23.4	23.2	61.7	56.0	42	0
162	23.4	23.2	61.0	55.8	42	0
163	23.4	23.1	60.6	55.8	41	0
164	23.6	23.4	59.7	54.8	40	0
165	23.6	23.4	59.6	54.9	40	0
166	23.6	23.4	59.5	55.0	39	0
167	23.5	23.3	59.6	55.1	38	0
168	23.5	23.3	60.0	55.2	40	0
169	23.5	23.3	60.7	55.5	40	0

170	20.5	20.1	48.2	47.8	62	0
171	20.6	20.2	47.7	47.5	64	0
172	20.7	20.3	47.6	47.4	65	0
173	20.8	20.4	47.4	47.3	64	0
174	20.8	20.4	47.4	47.3	64	0
175	20.8	20.4	47.3	47.2	62	0
176	20.9	20.5	47.2	47.2	61	0
177	20.9	20.5	46.9	47.0	60	0
178	20.9	20.5	47.0	47.0	60	0
179	21.0	20.6	46.5	46.6	58	0
180	21.1	20.7	46.3	46.9	59	0
181	21.1	20.7	45.8	46.4	59	0
182	21.1	20.7	45.5	46.5	58	0
183	21.5	21.3	45.2	44.3	57	0
184	21.6	21.5	44.9	44.1	56	0
185	21.7	21.5	44.3	43.7	55	0
186	21.9	21.6	44.0	43.1	62	0
187	21.9	21.7	44.1	43.2	60	0
188	22.0	21.8	43.4	42.6	59	0
189	22.0	21.8	43.0	42.1	58	0
190	22.1	21.9	43.0	42.3	57	0

191	22.3	22.0	42.1	41.6	56	0
192	22.7	22.2	41.7	41.3	55	0
193	22.5	22.3	39.2	39.3	54	0
194	22.6	22.5	39.7	39.2	56	0
195	22.8	22.6	39.6	39.3	56	0
196	22.8	22.6	39.3	39.1	56	0
197	22.9	22.6	38.9	38.9	57	0
198	22.4	22.3	34.1	34.7	58	0
199	22.2	22.1	34.4	34.5	59	0
200	21.9	21.9	35.0	34.8	57	0
201	21.8	21.7	35.8	35.5	59	0
202	21.6	21.6	36.3	35.8	60	0
203	21.5	21.6	36.4	35.8	59	0
204	21.4	21.5	37.2	36.8	58	0
205	21.3	21.5	38.0	37.4	55	0
206	21.2	21.4	38.3	37.6	59	0
207	21.1	21.1	37.1	36.8	57	0
208	21.0	21.1	37.6	37.1	60	0
209	21.1	21.2	38.4	37.5	59	0
210	21.0	21.1	38.6	37.9	58	0
211	21.0	21.1	38.2	37.7	60	0

212	20.9	21.1	38.0	37.5	59	0
213	20.9	21.0	38.1	37.4	60	0
214	20.9	21.1	38.2	37.5	58	0
215	20.9	21.1	38.6	37.9	59	0
216	20.9	21.1	38.7	37.9	60	0
217	20.8	21.0	38.8	38.1	59	0
218	20.7	20.9	38.9	38.1	59	0
219	20.7	20.9	38.3	37.7	60	0
220	20.7	20.8	38.3	37.9	60	0
221	20.7	20.8	38.2	37.6	59	0
222	20.7	20.9	38.2	37.2	59	0

223	24.2	23.8	61.6	61.0	51	0
224	25.0	24.8	63.1	62.8	49	0
225	25.8	25.2	62.8	62.1	57	0
226	26.4	26.1	62.7	62.0	57	0
227	26.9	26.3	62.4	59.8	55	0
228	27.1	26.8	63.0	62.7	54	0*(lim)
229	27.8	27.4	63.1	62.8	52	0*(lim)
230	28.2	27.9	62.9	62.2	50	0*(lim)
231	28.8	28.4	62.7	62.1	45	0*(lim)

232	29.1	28.7	62.2	59.9	44	0*(lim)
233	29.7	29.2	62.8	62.3	42	0*(lim)
234	30.0	29.8	62.5	62.0	40	0*(lim)
235	30.5	30.0	62.1	59.8	39	1
236	30.9	30.2	62.4	59.9	38	1
237	29.6	29.1	62.9	62.3	39	0*(lim)
238	28.9	28.4	62.6	62.0	39	0*(lim)
239	28.0	27.5	62.0	61.8	41	0*(lim)
240	27.8	27.2	62.2	61.9	42	0*(lim)
241	27.2	26.8	62.1	61.8	43	0*(lim)
242	26.9	26.5	61.9	61.2	49	0
243	26.4	26.0	61.8	61.1	48	0
244	28.2	27.9	67.2	66.8	45	0*(lim)
245	27.6	27.1	69.0	68.7	47	0*(lim)
246	24.9	24.2	67.5	67.0	57	0
247	24.2	23.8	62.5	62.0	54	0
248	22.0	21.8	63.9	63.1	59	0
249	22.8	22.1	62.0	61.7	56	0
250	22.2	21.9	61.9	61.4	57	0
251	21.9	21.4	59.9	59.4	58	0
252	21.7	21.2	59.4	59.0	59	0

253	21.5	21.0	59.7	59.1	59	0
254	20.8	20.5	59.6	59.2	59	0
255	20.7	20.4	59.4	59.1	60	0
256	20.6	20.1	59.3	59.0	60	0
257	20.4	20.0	59.4	59.1	60	0
258	20.3	20.0	59.3	59.0	60	0
259	20.2	19.9	59.2	58.8	60	0

260	20.2	19.9	40.5	39.5	62	0
261	20.9	20.0	41.5	40.6	60	0
262	19.8	19.1	38.5	37.5	62	0
263	19.0	18.5	45.8	44.1	63	0
264	19.2	18.4	44.2	43.4	61	0
265	18.7	18.0	49.5	48.2	62	0
266	18.1	17.5	50.4	49.8	60	0
267	17.5	16.8	51.1	50.9	61	0
268	17.0	16.4	52.7	51.8	60	0*(lim)
269	16.8	16.1	55.4	55.0	62	0*(lim)
270	16.2	15.8	56.2	55.9	61	0*(lim)
271	15.9	14.8	57.4	55.0	60	1
272	15.6	15.0	59.0	58.4	61	1

273	15.0	14.8	60.2	59.6	60	1
274	14.8	14.2	62.2	61.7	62	1
275	15.5	15.0	61.5	60.4	61	1
276	14.9	14.7	60.2	59.8	60	1
277	15.2	14.8	61.1	60.8	61	1
278	15.8	15.1	59.8	59.1	62	1
279	14.9	14.5	60.2	59.7	60	1
280	16.8	16.1	59.9	59.2	61	0*(lim)
281	16.9	16.5	60.3	59.8	62	0*(lim)
282	17.8	17.4	60.1	59.7	63	0
283	17.9	17.7	61.2	60.6	61	0
284	16.5	16.0	61.3	60.9	63	0*(lim)
285	18.1	17.8	62.1	61.7	62	0*(lim)
286	18.9	18.5	63.2	62.7	61	0
287	18.6	18.2	62.8	62.2	63	0
288	19.1	18.8	59.8	59.3	64	0
289	19.8	19.2	59.7	59.0	62	0
290	20.1	19.8	58.9	58.2	63	0
291	20.8	20.1	58.0	57.8	60	0
292	20.5	20.0	57.9	57.2	61	0
293	21.1	20.8	57.1	56.7	62	0

294	24.2	23.9	55.2	54.9	59	0
295	24.6	24.1	54.2	53.9	56	0
296	24.9	24.3	54.0	53.3	51	0
297	15.2	14.9	57.2	56.7	55	1
298	25.4	25.0	60.1	59.8	50	0
299	30.1	29.9	69.5	68.8	25	1
300	31.5	31.0	70.1	69.3	24	1
301	32.5	32.1	71.0	70.7	22	1
302	33.5	33.0	71.8	71.2	20	1
303	34.0	33.7	69.8	69.3	19	1
304	32.7	32.1	69.8	69.3	21	1
305	31.9	31.2	69.0	68.6	20	1
306	30.5	30.0	67.1	66.8	20	1
307	32.9	32.2	67.3	66.9	19	1
308	33.5	33.1	69.5	69.0	19	1
310	30.9	30.2	69.8	69.1	20	1
311	32.8	32.3	69.5	69.0	21	1
312	32.4	32.0	70.0	69.7	20	1
313	30.2	29.9	69.0	68.4	22	1
314	31.9	31.3	68.7	68.2	21	1
315	32.9	32.1	68.6	68.0	19	1

316	30.7	30.2	71.0	69.7	23	1
317	30.7	32.1	70.1	69.8	21	1
318	31.4	31.0	69.0	68.7	21	1
319	31.2	30.7	68.8	68.2	21	1
320	31.9	31.2	68.4	68.0	20	1
321	31.2	30.9	69.1	68.6	19	1
322	31.7	31.0	69.8	69.1	20	1
323	32.6	32.1	69.0	68.6	20	1
324	32.9	32.2	70.5	70.0	19	1
325	30.4	30.0	68.5	68.0	20	1
326	30.9	30.2	68.0	67.7	20	1
327	31.4	32.1	70.2	69.7	21	1
328	32.4	32.0	70.0	69.7	20	1
329	30.1	29.8	69.1	68.9	18	1
330	30.2	29.7	69.8	69.2	19	1
331	32.4	32.0	70.0	69.7	20	1
332	30.8	30.2	68.1	67.7	19	1
333	30.9	30.2	68.2	67.9	20	1
334	31.9	31.2	68.4	68.1	19	1
335	31.2	29.9	64.2	63.8	20	1
336	30.3	30.0	67.4	67.0	18	1

337	30.7	30.1	67.9	67.4	19	1
338	32.2	31.8	68.6	68.1	19	1
339	15.2	14.9	57.2	56.7	55	1
340	35.4	35.0	74.2	73.8	20	1
341	14.2	14.0	72.1	69.7	55	1
342	14.8	14.2	71.4	71.0	56	1
343	15.4	15.0	69.5	69.0	54	1
344	15.6	15.1	69.1	69.7	58	1
345	15.9	15.6	65.4	65.0	57	1
346	15.7	15.2	67.2	66.9	53	1
347	15.0	14.7	61.5	61.1	58	1
348	15.1	14.8	69.1	68.9	59	1
349	15.2	14.9	67.5	67.1	52	1
350	15.3	15.0	68.1	67.8	49	1
351	15.5	15.1	68.2	67.9	57	1
352	15.8	15.3	55.2	54.8	56	1
353	14.8	14.4	57.3	56.9	57	1
354	14.9	14.6	59.4	59.0	60	1
355	14.7	14.2	49.8	49.1	61	1
356	14.6	14.2	47.9	47.2	59	1
357	15.5	15.1	59.8	59.2	58	1

358	15.8	15.3	57.2	56.9	57	1
359	15.9	15.5	58.9	58.3	59	1
360	15.0	14.7	64.2	63.9	54	1
361	15.7	15.3	63.8	63.2	53	1
362	15.6	15.1	67.6	67.1	57	1
363	15.9	15.6	66.7	66.1	59	1
364	15.2	14.9	58.9	58.3	59	1
365	15.6	15.1	64.8	64.2	60	1
366	18.5	18.2	68.2	67.7	19	1*(Ps)
367	18.1	17.8	67.5	67.1	18	1*(Ps)
368	17.2	16.7	62.5	62.0	19	1*(Ps)
369	17.3	17.0	70.0	69.7	20	1*(Ps)
370	16.2	15.8	69.0	68.5	15	1*(Ps)
371	16.7	16.4	68.1	68.8	14	1*(Ps)
372	16.8	16.3	68.2	68.9	13	1*(Ps)
373	16.8	16.5	65.7	65.2	15	1*(Ps)
374	16.6	16.1	71.2	70.9	16	1*(Ps)
375	16.9	16.5	70.2	69.8	14	1*(Ps)
376	17.5	16.0	68.9	68.5	13	1*(Ps)
377	17.8	17.3	71.3	70.9	12	1*(Ps)
378	17.4	17.0	70.7	70.2	17	1*(Ps)

379	17.1	16.7	68.5	68.0	19	1*(Ps)
380	19.7	19.2	33.8	34.4	40	0
381	19.8	19.3	33.4	34.2	39	0
382	20.2	19.7	40.9	42.1	36	0
383	20.3	20.0	40.8	41.7	34	0
384	21.3	20.2	35.8	37.9	46	0
385	21.4	20.9	35.8	36.8	50	0
386	21.4	20.9	35.9	36.8	18	1
387	21.4	21.0	36.0	36.8	17	1
388	21.2	20.5	44.9	46.1	19	1
389	21.4	20.2	43.6	45.7	45	0
390	21.7	20.2	41.9	44.4	49	0
391	21.7	20.8	41.9	43.5	15	1
392	22.3	20.8	40.0	42.4	12	1
393	22.0	21.6	43.4	44.0	43	0
394	22.2	21.8	43.7	44.0	47	0
395	22.5	22.3	45.4	45.5	52	0
396	22.6	21.7	45.8	47.1	57	0
397	22.8	22.5	45.7	45.8	60	0
398	22.9	22.2	46.0	47.3	55	0
399	22.9	22.3	46.0	46.9	15	1

400	21.8	21.5	53.5	52.1	10	1
401	21.8	21.5	53.4	52.0	18	1
402	21.5	20.6	53.8	54.2	62	0
403	22.1	21.1	50.5	51.7	67	0
404	22.1	21.4	50.3	51.1	20	1
405	22.1	21.5	50.5	50.9	14	1
406	23.3	22.6	54.1	54.1	12	1

- **Pression**

Pression d'eau :

Unités de pression

Unités de pression

	<u>pascal</u>	<u>bar</u>	<u>atmosphère technique</u>	<u>atmosphère normale</u>	<u>torr</u>	<u>livre par pouce carré</u>
	Pa	bar	at	atm	Torr	psi
1 Pa	$\equiv 1 \text{ N/m}^2$	10^{-5}	$1,0197 \times 10^{-5}$	$9,8692 \times 10^{-6}$	$7,5006 \times 10^{-3}$	$1,450377 \times 10^{-4}$
1 bar	10^5	$\equiv 10^6 \text{ dyn/cm}^2$	1,0197	0,98692	750,06	14,50377
1 at	$0,980665 \times 10^5$	0,980665	$\equiv 1 \text{ kp/cm}^2$	0,9678411	735,5592	14,22334
1 atm	$1,01325 \times 10^5$	1,01325	1,0332	$\equiv p_0$	$\equiv 760$	14,69595
1 Torr	133,3224	$1,333224 \times 10^{-3}$	$1,359551 \times 10^{-3}$	$1,315789 \times 10^{-3}$	$\approx 1 \text{ mmHg}$	$1,933678 \times 10^{-2}$
1 psi	$6,8948 \times 10^3$	$6,8948 \times 10^{-2}$	$7,03069 \times 10^{-2}$	$6,8046 \times 10^{-2}$	51,71493	$\equiv 1 \text{ lb/po}^2$

- Le psi, de l'anglais *pound per square inch* (livre par pouce carré) est une unité anglo-saxonne très utilisée notamment en hydraulique, en oléohydraulique et en hydrostatique : 1 psi=6 894 Pa= 51,71 torr = 0.06894 bar

Programme sur MATLAB :

Code :

```
clc
```

```
close all;
```

```
% création d'un MLP:
```

```
a=xlsread('E1.xlsx');
```

```
b=xlsread('S1.xlsx');
```

```
net=newff(a,b,20);
```

```
% TRAIN MLP:
```

```
net.trainparam.epochs = 1000;
```

```
net.trainparam.goal = 0.001;
```

```
net.trainparam.lr = 0.01;
```

```
[net,tr]=train(net,a,b)
```

```
plottrainstate(tr);
```

```
plotperform(tr);
```

```
% architecture MLP:
```

```
view(net)
```

```
% test MLP:
```

```
ctest=xlsread('E2.xlsx');
```

```
y=sim(net,ctest);
```

```
y=round(y);
```

```
for i=1:length(y)
    if y(i)> 3
        y(i)=3
    end
end

for i=1:length(y)
    if y(i)< 0
        y(i)=0
    end
end

% error test

c=xlsread('S2.xlsx');

e=y-c;

e=round(e);

plot(e);

axis([0 140 -5 5])

% performance test:

performance=perform(net,y,c);
```

Bibliographie

- [1] Christophe LAMBERT : « Développement d'une méthode automatique fiable de modélisation de la structure tridimensionnelle des protéines par homologie et application au protéome de brucella multensis » facultés universitaires notre- dame de paix .2013
- [2] Gérard D. « Reducing the complexity of neural nets for industrial applications and biological models » Neuronal Information Processing - From Biological Data to Modelling and Applications (World Scientific, 1998).
- [3] Bernard G. « Application de réseaux de neurones artificiels a la reconnaissance au de caractères manuscrits »Thèse de Doctorat, Faculté polytechnique de Mons ,1996.
- [4] J.F. Jodouin, LES réseaux de neurones. Principes et définitions, édition Hermis,Paris1994.
- [5] B. Orsier. Etude et application de systèmes hybrides neurosymbolique. Thèse de doctorat, Universitié Joseph Fourier Grenoble, 1995.
- [6] H.Mezaache, "Les réseaux de Neurones formels Et Les systèmes Neuro-Flous Pour l'apprentissage par renforcement", Mémoire soumis en vue de l'obtention du Diplôme de Magister, Université El Hadj Lakhdar, Batna, 2008.
- [7] P. Campolucci,A circuit Theory Approach to Recurrent Neural Network Architecture and Learning Methods, These Doctorat, Université de Bologne, Février1998.
- [8] G.dreyfus, J.-M.martinez, M.Samuelides, M.B.Gordon, S.Thiria, L.Hérault, "réseaux de neurones, méthodologie et applications",2002.

- [9] Mourad ABIDI, 'Réalisation et implantation d'un réseau de neurones sur une architecture matérielle distribuée à base de réseau sur puce', mémoire de PFE d'ingénieur, Ecole Nationale d'Ingénieurs de Sousse, Juin 2014.
- [10] M.R. Alismail, N.Ourchani., "Fusion multimodale des scores pour la reconnaissance des personnes", Master 2, Université Mohamed Khider Biskra, 2011.
- [11] <https://www.mathworks.com/help/nnet/ug/introduction-to-convolutional-neural-networks.html>.
- [12] [chap2,2018]: « technique neuronales adaptées aux données spatio-temporelles, chapitre II ».
- [13] Tseng, Wen-Kung, Lu, and Chung-Sheng, "The system for appraisal of vehicle accident based on radial basis function neural networks," in *DOI: 10.1109/ICNC.2011.6022220*, 2011, pp. 869–872.
- [14] Frédéric Dardel, Francois Kepes ;« bioinformatique : génomique et post-génomique », éditions école polytechnique , juillet 2002.
- [15] H.Mezaache, "Les réseaux de Neurones formels Et Les systèmes Neuro-Flous Pour l'apprentissage par renforcement", Mémoire soumis en vue de l'obtention du Diplôme de Magister, Université El Hadj Lakhdar, Batna, 2008.
- [16] J. Hato, « intelligence artificielle et reconnaissance des formes. introduction », Technique de l'ingénieur II 1900. 1990.
- [17] Y. Hammouche , "Comparaison de plusieurs méthodes pour la prédiction de la Charge Electrique Nationale", Mémoire de magister, Université Badji Mokhtar , Annaba , 2009
- [18] (<http://www.mathworks.com/products/neuralnet>).
- [19] _____(<http://ch.mathworks.com/help/nnet/getting-started-with-neural-network-toolbox.html>).
- [20] National Instruments, Corporation ix Initiation à LabVIEW, Août 2006.

[21] P.V.M.J. Jones, Robust Real Time Face Detection, international journal of computer

vision kluwer academie publishers, 2004.

[22] M. Sorci, Modèle de choix discrets pour la reconnaissance des expressions faciales

statiques Ecole polytechnique générale de lausanne, 8 mars 2007.

[23] H.Tao and T.S. Huang, "A Piecewise Bezier Volume Deformation Model and Its Applications in Facial Motion Capture," in Advances in Image Processing and Understanding :

A Festschrift for Thomas S. Huang, edited by Alan C. Bovik, Chang Wen Chen, and Dmitry

B. Goldgof, 2002.