

République Algérienne Démocratique Et Populaire

Ministère de l'Enseignement Supérieur

et

de la Recherche Scientifique



Université d'Adrar

Faculté des Sciences

Et

De la Technologie

Département des Mathématiques

Et

Informatique

Mémoire de Fin d'Etude

Pour l'Obtention du Diplôme de Master en Informatique

*Option: Systèmes Intelligents*

Thème :

**L'application de l'approche neuronale pour la prévision du  
rayonnement solaire**

Présenté Par :

**LAISSAOUI LOUALI**

Encadré par :

**MR. BENATIALLAH DJELLOUL**

**Année Universitaire 2018/2019**

## المخلص

لقد اثر الذكاء الاصطناعي في شتى العلوم بتطبيقاته المختلفة ذات الدقة و السرعة في استنتاج النتيجة، حتى صار ملتجاً لكل الباحثين، و من خلال بحثنا هذا اطلعنا على الكثير من البحوث في تخصصات فيزيائية و إحصائية تلجا لخوارزميات الذكاء الاصطناعي و خاصة خوارزمية الشبكة العصبية للحصول على نتائج أكثر دقة عند التخمين أو التكهن، لكنها كانت جميعها تركز على إظهار النتائج دون التطرق إلى العمليات و الخطوات البرمجية المؤدية إلى تلك النتائج، و في بحثنا هذا أردنا أن نتدارك ذلك و نركز على الجانب العملي دون النظريات، و تناولنا التكهن بكمية الأشعة الشمسية اليومية، دون التطرق إلى ماهية هذه الأشعة و خصائصها الفيزيائية و التكوينية، و اعتبرناها مجرد بيانات تحتاج إلى نظام معالجة ذكية يحلها ثم يتمرن عليها ليتعلم كيف يتوقعها في المستقبل، بينما ركزنا على إظهار التقنيات المختلفة التي ينتهجها الذكاء الاصطناعي للتكهن، بالإضافة إلى أفضل أنواع الشبكات العصبية المختصة في التكهن، كما أبرزنا مجموعة من الدوال الجاهزة في برنامج "ماتلاب" يمكن للطلبة و الباحثين استخدامها لإنشاء شبكة عصبية من اجل التكهن أو التخمين.

الكلمات المفتاحية: الذكاء الاصطناعي، التعلم الآلي، الانحدار الخطي، الشبكات العصبية الاصطناعية، التدريب، التنبؤ

## Résumé

L'intelligence artificielle a influencé les diverses sciences par la rapidité et la précision de ses divers applications en conclusion des résultats, elle est devenue un refuge pour tous les chercheurs. Notre étude nous a permis de consulter de nombreuses recherches dans les disciplines physiques et statistiques utilisant des algorithmes d'intelligence artificielle notamment l'algorithme de réseau de neurones pour avoir des résultats plus précises lors de la conjecture ou la prévision, mais elles visaient toutes à montrer les résultats sans aborder les processus et les programmes menant à ces résultats. Par la présente recherche, nous voudrions remédier à ceci et nous concentrer sur le volet opérationnel sans le volet théorique, nous avons spéculé la quantité des rayonnements solaires quotidienne sans aborder la nature de ces rayonnements ni ses propriétés physiques et structurelles, que nous les avons considéré comme des données nécessitant un système de traitement intelligent qui les analysera puis les exercera pour savoir comment les prévoir dans le futur, notre recherche met en évidence les différentes techniques utilisées par l'intelligence artificielle pour conjecturer, les meilleurs types de réseaux de neurone spécialisé dans la prévision ainsi qu'un ensemble de fonctions prédéfinies du programme Matlab que les étudiants et les chercheurs peuvent les utiliser pour créer un réseau de neurones afin de prévoir ou de conjecturer.

**Mots clés** : L'intelligence artificielle, Machine Learning, Régression linéaire, Les réseaux de neurones artificiels, Training, la prévision.

## Abstract

Artificial intelligence had a big influence in the various sciences by the precision and speed of its deferent applications in the conclusion of the result, it has become a refuge for all researchers, and through our present research we had looked onto a big number of researches in physics and statistics disciplines taking refuge on artificial intelligence algorithms especially the neural network algorithm the most precise result while guessing or predicting, but they were all focused on showing the results without addressing the process and steps of the program leading to these results. And in this research, we wanted to cover this point and focus on the operational side without theories, and we have chosen the prediction of the amount of daily solar radiation without talking about the nature of this radiation or its physical and structural properties, and we have just considered it a data requiring an intelligent processing system that would analyze it and then exercise it to know how to predict for it in the future, while highlighting the different techniques used by artificial intelligence to predict in addition to the best kind of neural network specialized in prediction, we also highlighted a set of states ready in Matlab program that students and researchers may use to create a neural network to predict or guess.

**Key words**: Artificial intelligence, Machine Learning, Linear regression, Artificial neural networks, Training, prediction.

# Table des matières

## Table des matières

Liste des figures .....	1
Nomenclature.....	3
Glossaire des acronymes .....	4
Introduction générale.....	5
1. Problématique .....	5
2. Plan : .....	5
3. Méthodologie : .....	6
4. Limites de l'étude : .....	6
5. Les études précédentes : .....	6
Chapitre1 : la prévision.....	7
I-1 Introduction.....	7
I-2. L'intelligence artificielle.....	7
I-3. Applications d'intelligence artificielle .....	7
I-4. Machine Learning .....	9
I-5. Les techniques prédictives .....	10
I-6. Régression linéaire .....	10
I-6-1. Présentation formelle .....	11
I-6-2. Qualité de la prédiction.....	12
I-6.3. Avantage : .....	13
I-6.4. Inconvénients : .....	13
I-6.5. La police la plus appropriée : .....	14
I-7. L'apprentissage.....	16
I-7.1 Apprentissage supervisé : .....	16
I-7.1.1. Error Back Propagation (EBP)ou Gradual Regression(GR) .....	17
I-7.1.2. Descente graduelle (Gradient Descent) : .....	17
I-7.1.3. Descente de gradient avec élan (GDM).....	19
I-7.1.4. Conjugate Gradient BP (CGF) .....	19
I-7.1.5. Quasi- Newton (BFGS) .....	19

# Table des matières

I-7.1.6. Levenberg Marquardt (LM) .....	19
I-7.2 Apprentissage non supervisé .....	19
I-7.3 Apprentissage hybride.....	20
I-8. Conclusion : .....	20
Chapitre2 : Les réseaux de neurones artificiels.....	21
II-1 Introduction.....	21
II-1-1 Fondement biologique .....	21
II-1-2 Historique.....	22
II-2 Neurone formel.....	22
II-2- 1 Représentation mathématique.....	23
II-2-2 Fonctions d'activation .....	23
a. fonction binaire à seuil .....	24
b. Fonction linéaire.....	24
c. Fonction linéaire à seuil ou multi seuil.....	24
d. La fonction sigmoïde .....	24
e. La fonction de transfert de base radiale .....	25
II-3 Réseau de neurones artificiel (RNA).....	25
II-3-1 Définition.....	25
II-3-2 Architecture d'un réseau de neurones artificiel .....	26
II-3-2-1 Réseaux monocouche.....	26
II-3-2-2 Réseaux multicouche .....	26
II-3-2-2.1.Réseau multicouche classique.....	26
II-3-2-2.2.Réseau à connexions locales : .....	29
II-3-2-2.3.Réseau à connexions récurrentes :.....	29
II-3-3 L'apprentissage des réseaux de neurones.....	30
II-3-3-1 Règles d'apprentissage des réseaux de neurones artificiels .....	30
II-3-3-2 Algorithme d'apprentissage.....	31
II-3-4. Training .....	33
II-4 Conclusion .....	33
Chapitre 3. Notre approche.....	34
Introduction :.....	34

## Table des matières

III-1- Importation des bibliothèques et des modules .....	34
III-2- chargement des données.....	35
III-3- Traitement des données .....	37
III- 3-1- Initialisation de données par l'outil Excel: .....	37
III-3-2-Analyse de données avec les outils habituels de Matlab: .....	39
III-4- Définition de l'architecture du modèle .....	40
III-4-1. l'architecture du Feed Forward Neural Networks (FFNN) .....	40
III-4-2. l'architecture du Cascade Neural Network (CNN).....	40
III-4-3. l'architecture du Radial Basis Functions Network (RBFN).....	40
III-4- Adaptation du modèle aux données d'entraînement.....	41
III-5- Evaluation du modèle sur les données de test .....	44
III-6- prédire des valeurs.....	45
L'Interfaces système .....	46
III-7 Conclusion .....	47
Conclusion générale .....	48
Références Bibliographiques.....	49
Références françaises.....	49
Références anglaises .....	50
Références en arabe .....	50

## Liste des figures

Figure 1 : Applications d'intelligence artificielle (4) .....	8
Figure 2 : Branche de Machine Learning(6) .....	9
Figure 3 : les algorithmes de Machine Learning.....	11
Figure 4 : la meilleure ligne .....	12
Figure 5 : l'équation ligne de régression .....	13
Figure 6 : situations complexes relations non linéaires .....	13
Figure 7 : Les polices de prédiction linéaire .....	14
Figure 8 : les termes importants en prédiction linéaire.....	14
Figure 9 : réseau a apprentissage supervisé.....	16
Figure 10 : les valeurs de $\theta$ globale et les valeur locale .....	18
Figure 11 réseau a apprentissage non supervisé .....	20
Figure 12 : Neurone de Mac Culloch et Pitts.....	22
Figure 13 : Neurone formel.....	22
Figure 14 : Fonctions binaires a seuil.....	24
Figure 15 : Fonction linéaire .....	24
Figure 16 : Fonction linéaire a seuil.....	24
Figure 17 : Fonction sigmoïde .....	25
Figure 18 : Radial basis transfer function.....	25
Figure 19 : réseau de neurones monocouche .....	26
Figure 20: Réseau multicouche classique.....	26
Figure 21 : Architecture Perceptron Multicouche(MLP) .....	27
Figure 22 -FFNN .....	27
Figure 23 CNN.....	27
Figure 24 ENN .....	28

Figure 25 RBFN .....	28
Figure 26: Schéma d'une carte auto-organisatrice SOM.....	29
Figure 27: Réseau à connexions récurrentes .....	29
Figure 28 : Algorithme d'apprentissage du perceptron .....	31
Figure 29 : base de données brut .....	36
Figure 30 : Informations sur la base de données .....	36
Figure 31 : Traitement informatique de la quantité de rayonnement en fonction de la variable horaire .....	37
Figure 32 : Traitement de données de la quantité de rayonnement en termes de variable de temps quotidien	38
Figure 33 : régression linéaire de Quantité de rayonnement solaire pendant 40 et 400 jours .....	39
Figure 34 : Mise en œuvre de la prédiction linéaire des bibliothèques Matlab à l'aide de méthodes statistiques traditionnelles .....	39
Figure 35 : Feed-forward neural network .....	42
Figure 36 : Function fitting neural network(FitNet) .....	42
Figure 37 : Cascade-forward neural network .....	43
Figure 38 : Radial basis network.....	43
Figure 39 : Interface utilisateur .....	46

## Nomenclature

- $h(x)$  : la valeur attendue de l'équation linéaire =  $\theta_0 + \theta_1 X$
- $X$  : représente les données d'entrée pas moins de 2 et non spécifiées.
- $Y$  : la vraie valeur de la sortie.
- $\theta_0$  : transactions du  $x$ , c'est ordonnée à l'origine et la valeur de  $h$  lorsque  $x$  est 0
- $\theta_1$  : transactions du  $x$ , pente = taux de prédiction
- $J$  : la valeur de l'erreur
- $\alpha$  : Learning rate
- $a^i$  : la valeur d'activation est la sortie de toute cellule infinie et peut devenir une entrée dans la couche suivante.
- $\theta^i$  : c'est le coefficient  $x$  doit être trouvé.
- $\omega^i$  : les poids sont la matrice requise pour chaque couche.
- $z^i$  : est un multiplicateur de  $X * \theta^i$  par cellule et ses valeurs sont regroupées dans une grande matrice  $Z$ .
- $g$  est le résultat de l'entrée  $z$  dans la fonction sigmoïde ou d'une autre fonction et valeurs sont regroupées dans une grande matrice  $G$ .
- Layer : couche groupe de cellules de chaque niveau et au moins trois couches.
- $L$  : le nombre de couches de la cellule, y compris les couches d'entrée et de sortie.
- $S_l$  : nombre d'unités par couche.
- $h_{\theta}^x$  : la valeur attendue de la sortie du réseau et la symbolise  $y$  ou  $a$ .
- $\delta^l$  : est la différence entre la valeur réelle dans chaque élément,  $\delta^l = a^i - y^t$ .
- $\Delta_{i,j}^l$  : la différence totale entre l'attendu et le réel est si :  $\Delta_{i,j}^l = \sum \delta^l$ .
- $D_{i,j}^l$  : est la différence exacte, et la somme des valeurs  $\Delta_{i,j}^l$  après l'ajustement.
- $j(\theta)$  : Cost, la valeur de l'erreur.
- $\gamma$  : le coefficient de lissage est choisi en fonction des limites pour déterminer la vitesse de travail.
- $y_k^i$  :  $i$  (numéro de la couche),  $k$  (numéro de cellulaire).



### Glossaire des acronymes

GHI : Global Horizontal Irradiance

ANN : Artificielle Neural Network

IA : Intelligence Artificielle

MSE: Mean Squared Error (*MSE/RMSE* : erreur quadratique moyenne et sa racine carrée)

MAE: Mean Absolute Error performance function

SSE: Sum Squared Error performance function

RNA: Réseau de Neurones Artificiels

MLP: Multi layer Perceptron

SOM: Self Organised Maps

LVQ: Learning Vector Quantization)

QSAR: Quantitative Structure-Activity Relationship

PMC : Perceptron multicouche

RNN : Recurrent Neural Network

**RBFNN** : Radian Basis Function Neural Network

**MAE/MAPE** : Mean Absolute (pourcentage) Error, erreur absolue moyenne et sa forme  
Single Layer Networks

## Introduction générale

La prévision est l'un des outils de prise de décision les plus importants et l'élément le plus stratégique du processus de planification pour l'avenir. Pour chaque décision, il faut étudier le présent et le passé, puis prévoir les conséquences pour l'avenir.

Il existe plusieurs méthodes traditionnelles de prévision, mais elles étaient abandonnées et remplacées par les méthodes modernes, qui reposent sur une intelligence artificielle, en raison de sa précision, de sa rapidité et de son faible coût.

Le rayonnement solaire est l'une des plus importantes sources d'énergie renouvelable, la plus sûre, la moins chère, la plus disponible et la plus durable de notre région, de notre patrie, "l'Algérie" et de notre ville "Adrar" qui, selon la base de données observée au cours de la période 2009/2018, possède la plus longue durée de luminosité solaire grâce au graphique de série chronologique GHI(Global Horizontal Irradiance) ou RS(Rayonnement Solaire).

**Notre objectif** est de réaliser une application basée sur un réseau de neurones afin de prévoir la valeur de rayonnement solaire dans le futur.

## 1. Problématique

Dans de nombreuses études et statisticiens ont reconnu que l'algorithme de l'intelligence artificielle (réseau de neurones) est le meilleur et le plus précis moyen pour prévoir, estimer et catégoriser, mais le problème est de savoir quel est son type ? quels sont les modèles préférés ? et quels sont les algorithmes d'apprentissage utilisés ?.

D'autre part, est ce que l'algorithme de réseau neuronal est le meilleur algorithme on le comparent à d'autres algorithmes d'intelligence artificielle tels que l'algorithme génétique, l'Algorithme d'escadron, l'algorithme de fourmis, etc. ?

## 2. Plan :

Pour cela, nous avons adopté un plan comprenant une introduction et trois chapitres :

Le premier chapitre traite la prédiction dans l'intelligence artificielle et la branche de machine Learning,

Le deuxième chapitre traite les réseaux de neurones et leurs types, en particulier le type de prédiction,

Le troisième chapitre est un cas pratique : un état de la série temporelle du GHI ou R-S.

### 3. Méthodologie :

l'étude théorique adopte l'approche descriptive, la méthode d'application et la programmation inductive.

### 4. Limites de l'étude :

La présente recherche traite : le choix du type d'apprentissage, les types des réseaux de neurones artificielles, le nombre de couches qu'il contient, puis les valeurs du paramètre d'erreur.

Traitement d'une série chronologique a une variable, a savoir GHI ou R-S , la base de données est limitée a 2013/2018.

### 5. Les études précédentes :

1. Mémoire de magistère en systèmes d'information intitulé : ( استخدام الشبكات العصبونية )  
للتنبؤ في الأسواق المالية -سوق دمشق للأوراق المالية- إعداد عروة نادر)  
Le chercheur a installé deux modèles ANN pour accroître la précision des prévisions, l'un pour traiter les cours des actions, l'autre avec des indicateurs d'analyse technique, puis pour fusionner leurs résultats dans un troisième modèle ANN.
2. Mémoire de magistère en informatique intitulé : (التنبؤ بسرعة الرياح باستخدام الشبكات العصبونية -إعداد لمياء عبد اللطيف عمر العاني-)  
La chercheur a installé trois modèles l'un avec une couche cachée, le deuxième avec deux couches le troisième avec trois couches, pour prédire la vitesse du vent.
3. Mémoire de magistère en informatique intitulé : (comparaison de plusieurs méthodes pour la prédiction de la charge électrique nationale -préparé par : Yassine Hammouche)  
Ce travail s'inscrit dans le cadre de la prédiction des séries temporelles. Plus particulièrement, la prédiction de la charge électrique. Des techniques sont appliquées pour prédire la charge électrique Algérienne.  
Quatre modèles sont proposés dans le cadre de ce travail, les deux premiers se focalisent sur la prédiction d'un seul pas en avant, un modèle se base sur la méthode Box-Jenkins qui représente dans la littérature la base de comparaison de toutes autres techniques, les résultats obtenus par cette méthode sont comparés avec ceux obtenus par un modèle réseau de Neurones Artificiels de type Perceptron Multi Couches.

## Chapitre1 : la prévision

### I-1 Introduction

La prévision est le synonyme de prédiction, qui ressemble à la classification et à l'estimation mais dans une échelle temporelle différente, elle s'appuie sur le passé et le présent mais son résultat se situe dans un futur généralement précisé(1) . La seule méthode pour mesurer la qualité de la prédiction est d'attendre ! Les techniques les plus appropriées à la prédiction sont Les arbres de décision et les réseaux de neurones. Dans ce chapitre, nous voulons savoir qu'elle est la relation entre l'intelligence artificielle et la prévision ? Quelles sont les branches spécialisées dans la prévision ? Quel est l'algorithme qui donne la prédiction la plus précise ?

### I-2. L'intelligence artificielle

C'est le comportement et les caractéristiques spécifiques des programmes informatique capables de stimuler les capacités mentales humaines telles que la capacité a apprendre a déduire et a réagir a des situations qui n'ont pas été programmées dans la machine. John McCarthy l'a défini comme étant « la science et l'ingénierie de la fabrication de machines intelligentes », Andreas Kaplan et Michael Heinlin ont la défini : « la capacité du système a interpréter les preuves a en tirer des enseignements et a utiliser l'apprentissage pour réaliser des fonctions d'adaptation souples ». (2)

### I-3. Applications d'intelligence artificielle

Si l'on considère que la date de 1956 est la naissance ou l'émergence de l'intelligence artificielle, lorsque les participants à la conférence de Dartmouth introduisent des ordinateurs parlant anglais et résolvent des problèmes d'algèbre et de logique;

et que 1985 est l'histoire de la maturité et de la force du succès des systèmes experts ;

le 21ème siècle était l'ère de la révolution et des ramifications, où les branches de l'IA se divisent en sous-domaines profondément indépendantes, en raison de la spécialisation et de la technologie, comme le montre la figure1 ci-dessous. Parmi ces branche on trouve la branche de l'apprentissage automatique (Machine Learning), que nous adoptons pour étudier notre thème : «Prédiction en intelligence artificielle ou régression en IA». (3)

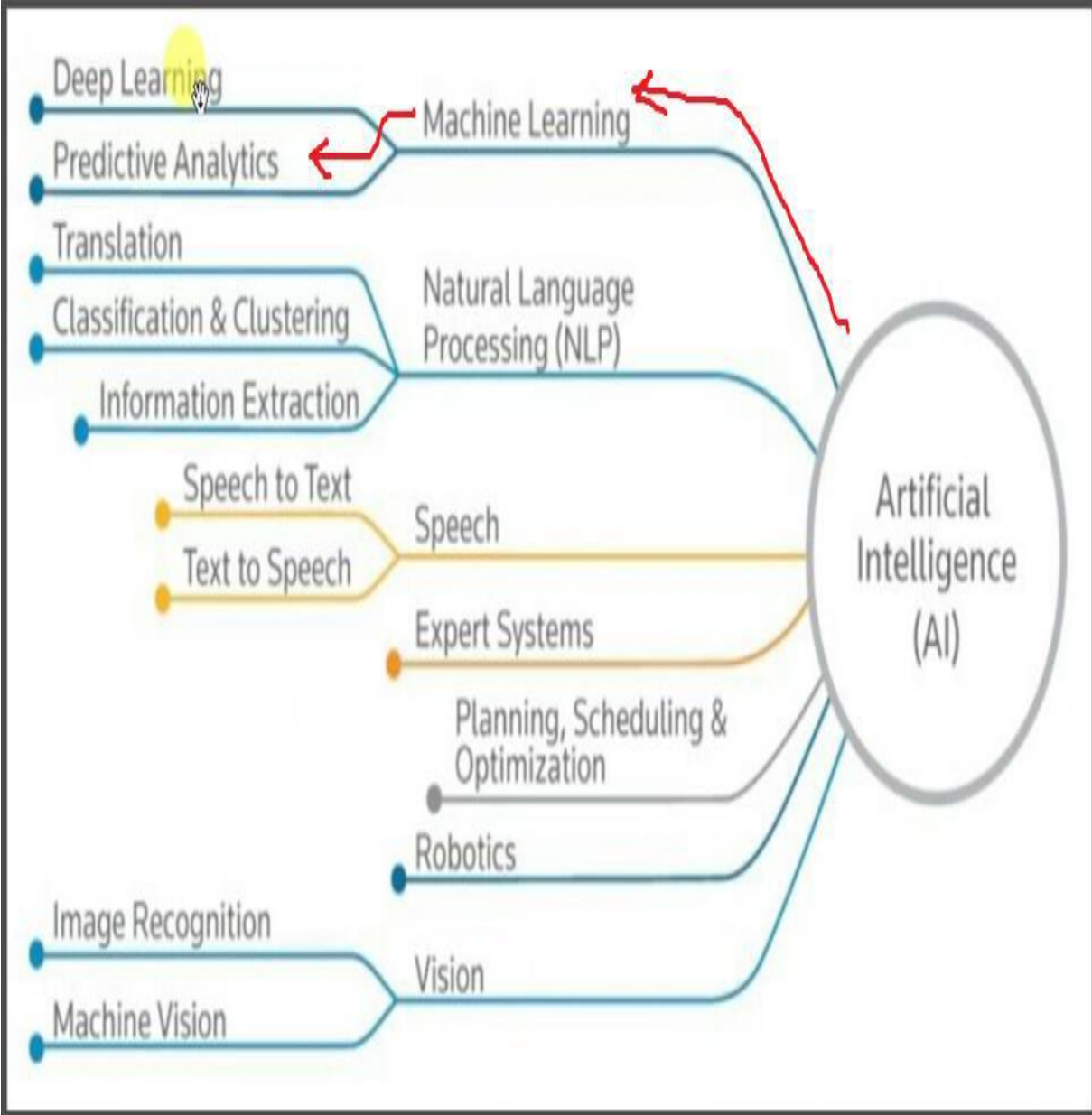


Figure 1 : Applications d'intelligence artificielle (4)

**I-4. Machine Learning**

Le Machine Learning est au cœur de la recherche sur l'intelligence artificielle, il comprend deux parties : l'apprentissage sans supervision et l'apprentissage sous supervision ;

la première partie s'applique à l'apprentissage en profondeur (Deep Learning) et la deuxième partie à la prévision analytique, comme schématisé à la figure2.

l'apprentissage sous supervision est subdivise, à son tour, en deux branches, une branche chargée de la classification et l'autre de la prédiction, la différence entre ces deux branches est que la première a comme objectif l'organisation des entrées en classes, alors que la seconde vise à trouver les valeurs réelles des entrées par rapport a celles apprises des entrées précédentes. (5)



Figure 2 : Branche de Machine Learning(6)

### I-5. Les techniques prédictives

Ce sont les méthodes utilisées pour expliquer et prévoir un ou plusieurs phénomènes observables et pour mesurer efficacement l'analyse de toutes les variables de la base de données à l'aide d'une classification avec apprentissage supervisé, basée sur un apprentissage automatisé. Ces techniques visent à extrapoler de nouvelles informations à partir des informations inconnues (telles que l'état du rayonnement solaire futur), auquel cas il existe une variable cible prévisible.

Les anciennes méthodes prédictives utilisent la science statistique, en particulier les modèles de Box et Genghis, alors que les modèles modernes utilisent soit : la logique (arbres décisionnels), soit les mathématiques tels que les réseaux neuronaux d'apprentissage supervisé (perception multicouches) soit la régression linéaire et les algorithmes génétiques.

L'apprentissage initialement supervisé du groupe de contrôle  $X$  du groupe {Jour} et les mesures {RS} appartenant à  $Y$ , nous essayons d'estimer les dépendances entre les groupes  $X$  et  $Y$ . par exemple: Nous essayons d'estimer les liens entre le temps et la quantité de rayonnement solaire,  $x_i$  est le jour et  $y_i$  est la quantité Rayonnement la veille. Et c'est l'apprentissage sous supervision en raison de l'existence du  $y_i$  qui peut guider le processus de prévision.

### I-6. Régression linéaire

De nombreux algorithmes d'apprentissage sont sélectionnés en fonction de l'objectif, des données et du problème présenté, comme le montre la figure3, mais nous avons pour objectif de conjecturer, nous choisirons donc l'algorithme de régression (car l'algorithme de RNA applique une bibliothèque qui utilise les fonctions de régression particulièrement « Régression linéaire »).

En statistiques et Machine Learning, Le modèle de régression linéaire est un modèle de régression qui cherche à établir une relation linéaire entre la variable présentée et la variable explicative, on l'appelle aussi modèle linéaire ou modèle de régression linéaire.

La régression linéaire est utilisée dans les statistiques pour déterminer l'effet d'une ou de plusieurs données variables sur l'évaluation linéaire, en tenant compte de paramètres aléatoires.

Le modèle de régression linéaire est souvent estimé à l'aide de la méthode des moindres carrés, bien qu'il existe de nombreuses façons de l'estimer. (7)

Pour prédire les phénomènes et les quantités, nous pouvons prévoir la Quantité RS de certaines valeurs de temps (Jour).

Dans Machine Learning, la méthode de régression linéaire est une méthode d'apprentissage supervisé utilisée pour conjecturer une variable quantitative, dans laquelle le modèle est généralement entraîné sur un modèle d'apprentissage, puis la performance prédictive du modèle est testée sur un échantillon test . (7)

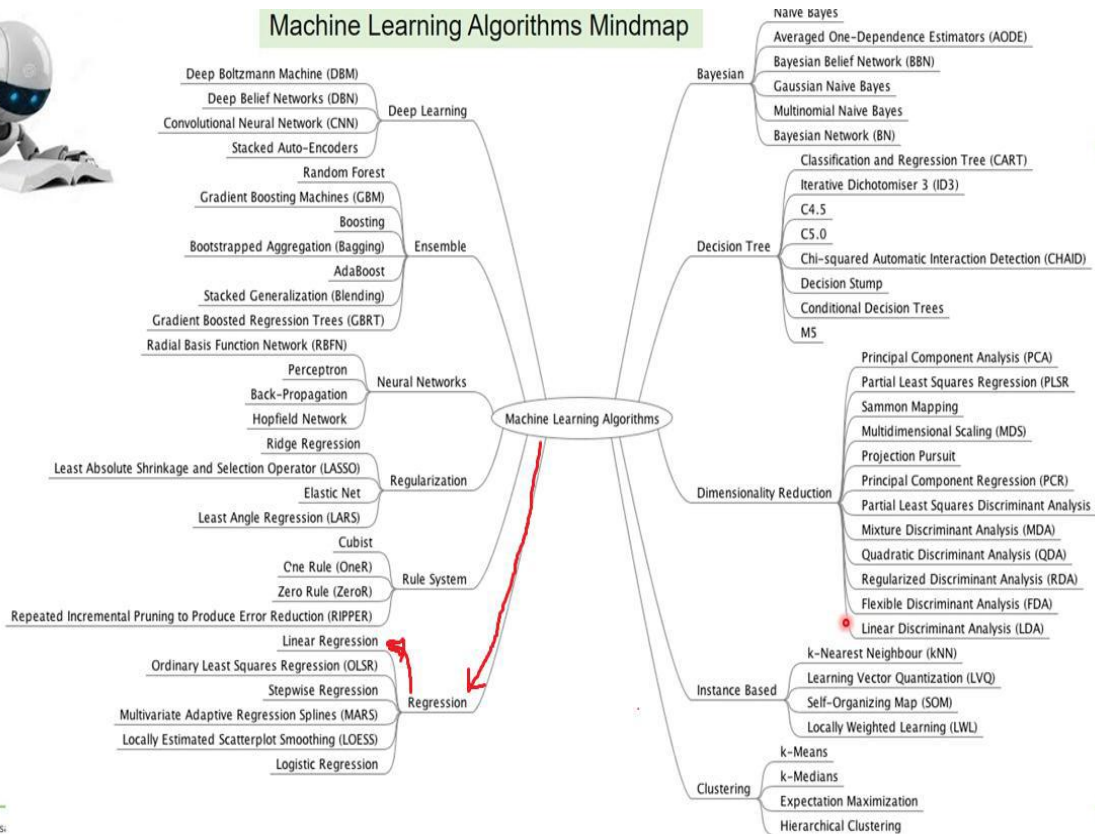


Image source: Choices galore for ML algorithms (<https://jixta.wordpress.com/2015/07/17/machine-learning-algorithms-mindmap/>)

Figure 3 : les algorithmes de Machine Learning

**I-6-1. Présentation formelle**

Il existe trois types de Présentation:

Présentation numérique  $RS = \theta_0 + \theta_1 J_1 + \dots + \theta_K J_K + \epsilon_i$

Présentation de vecteur  $RS = \theta_K J_K + \epsilon_i$

Présentation de la matrice  $[RS] = [\theta_K][J_K] + [\epsilon_i]$



### I-6-2. Qualité de la prédiction

Pour évaluer la qualité de la prédiction, on peut utiliser :

$SSE = \sum_{t=1}^n (\widehat{RS}_i - \overline{RS})^2$  est la variation expliquée par la régression (*Sum of Squares Explained*, en français SCE Somme des Carrés Expliquée [par la régression]).

$SSR = \sum_{t=1}^n (RS_i - \widehat{RS}_i)^2$  est la variation expliquée par les résidus (*Sum of Squares Residual*, en français SCR Somme des Carrés Résiduelle).

$SST = SSE + SSR = \sum_{t=1}^n (RS_i - \overline{RS})^2$  est la variation totale (*Sum of Squares Total*, en français SCT Somme des Carrés Totale).

Dépend de la création de la meilleure ligne comprenant la plupart des points de données, comme le montre la figure4.

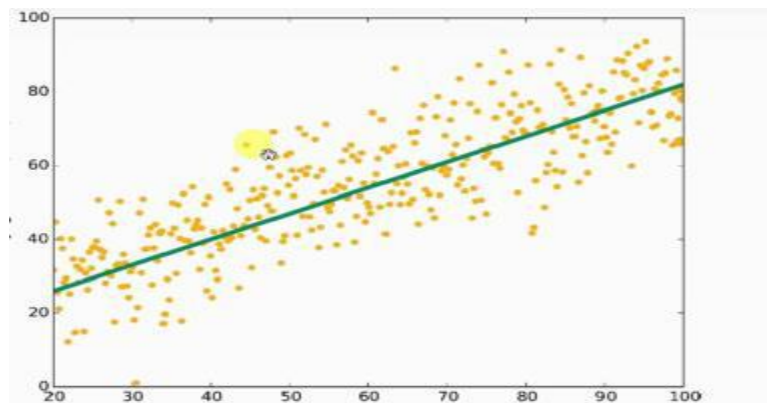


Figure 4 : la meilleure ligne

Il est utilisé pour prédire les valeurs réelles sur la base d'une variable constante telle que la série temporelle. Une relation entre les variables indépendantes et non indépendantes est évaluée par la meilleure structure de ligne appelée ligne de régression et est représentée par l'équation  $RS = \theta_0^1 + \theta_1^2 \text{Jour}_1$  (7), comme le montre la figure5.

<sup>1</sup>  $\theta_0$  : l'ordonnée à l'origine

<sup>2</sup>  $\theta_1$  : la pente de la droite de régression

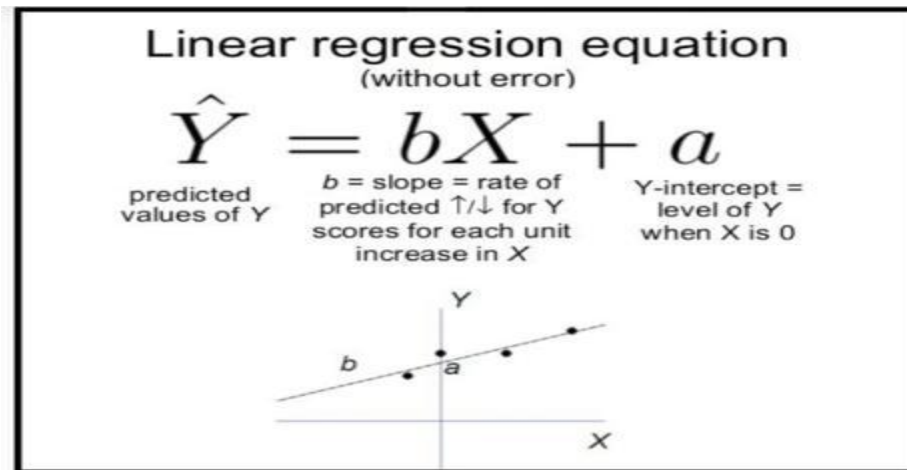


Figure 5 : l'équation ligne de régression

**I-6.3. Avantage :**

C'est l'un des algorithmes les plus célèbres en statistique et en apprentissage. Sa technologie est rapide et simple. Facile a utiliser et nécessite un ajustement minimal.

**I-6.4. Inconvénients :**

Il est limité aux relations linéaires et à la prédiction de sortie numérique. Sensible aux valeurs extrêmes et aux situations complexes, comme le montre la figure6. Ne gère que des données indépendantes.

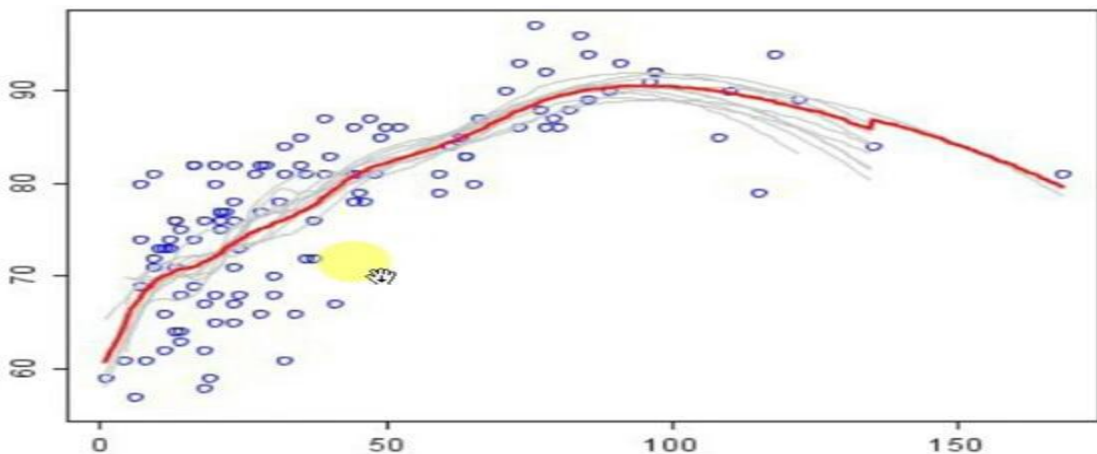


Figure 6 : situations complexes relations non linéaires

**1-6.5. La police la plus appropriée :**

Est la ligne sur laquelle la prédiction dépend et doit être tracée avant tout en trouvant un écart ( $\theta_1$  : la pente de la droite de régression) défini comme l'équation suivante :

$\theta_1 = (RS_2 - RS_1) / (\text{jour}_2 - \text{jour}_1)$ , comme les données sont indépendantes et sont souvent dispersées, la valeur de RS a plusieurs images dans Jours, nous devons donc considérer une valeur appropriée en les multipliant dans le  $\theta$  et en recherchant le point le plus approprié  $J(\theta_0, \theta_1)$ , défini en figure7 suivante :

Hypothesis:	$h_{\theta}(x) = \theta_0 + \theta_1 x$
Parameters:	$\theta_0, \theta_1$
Cost Function:	$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$
Goal:	minimize $J(\theta_0, \theta_1)$ $\theta_0, \theta_1$

Figure 7 : Les polices de prédiction linéaire

Avant d'appliquer cela a un exemple, il existe des termes importants en prédiction linéaire et en apprentissage en général, dans le tableau suivant :

<b>Input X</b>	Entrée
<b>Output Y</b>	Sortie
<b>Rows m</b>	<b>Lignes</b>
<b>Features n</b>	Caractéristique
<b>h(x)</b>	Valeur prédite
<b>Cost J</b>	La valeur de l'erreur
<b>Theta <math>\Theta</math></b>	Transactions du X

Figure 8 : les termes importants en prédiction linéaire

Exemple :

Dans cet exemple, les données sont les suivantes :

X	Y
1	7
2	8
2	7
3	9
4	11
5	10
5	12

L'équation de prédiction linéaire est la suivante :

Equation  $h(x) = 5 + 2x$

Le but est de réduire la différence entre la valeur de  $h(x)$ (la valeur attendue de l'équation linéaire) et la valeur de  $y$  ( la valeur réelle ) et de la diviser par  $2m$  pour lier la valeur de l'erreur au nombre de valeurs dans les données, puis a la fréquence jusqu'à trouver la valeur de  $\theta_0$  et  $\theta_1$  rendant  $J$  aussi bas que possible.

Theta0 = 5 , theta 1 = 2      Equation  $h(x) = 5 + 2x$

X	Y	h(x)	h(x) - y	(h(x) - y) <sup>2</sup>
1	7	7	0	0
2	8	9	1	1
2	7	9	2	4
3	9	11	2	4
4	11	13	2	4
5	10	15	5	25
5	12	15	3	9

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

$J = 1 / 14 ( 0+1+4+4+4+25+9 )$   
 $J = 47/14 = 3.3$

## I-7. L'apprentissage

La recherche des algorithmes d'apprentissage statistiques a mené, au cours des deux dernières décennies, à des importantes découvertes qui ont changé la manière de concevoir les problèmes et qui ont rendu les ordinateurs plus intelligents.

Un agent est intelligent parce qu'il a des connaissances opérationnelles qui lui permettent d'effectuer certaines tâches ou de répondre à certaines questions sur un domaine donnée. L'objectif de l'apprentissage est de fournir une méthode au réseau, afin qu'il puisse ajuster ses paramètres lorsqu'on lui présente des données à traiter.

On distingue trois types d'apprentissage : supervisé, non supervisé et hybride. En effet les réseaux de neurones ont été Appliqués avec succès à la tâche de classification, de prévision et d'estimation.

L'apprentissage est le processus d'adaptation des paramètres d'un système pour donner une réponse désirée à une entrée ou une stimulation quelconque. (9)

Et est utilisé pour cela plusieurs algorithmes, dont le plus important:

- Gradual Regression (GR) ou Error Back Propagation (EBP)
- Gradient Descent (GD)
- Gradient Descent with Momentum (GDM)
- Conjugate Gradient BP (CGF)
- Quasi-Newton (BFGS)
- Levenberg Marquardt (LM)

### I-7.1 Apprentissage supervisé :

Dans l'apprentissage supervisé, on présente à l'ordinateur des exemples sous la forme des paires (entrée, sortie désirée).

Il s'agit de la donnée à traiter et la sortie attendue.

L'apprentissage supervisé consiste à calculer les coefficients de telle manière que les sorties soient, pour les exemples utilisés lors de l'apprentissage, aussi proches que possibles des sorties 'désirées', qui peuvent être : la classe d'appartenance de la forme que l'on veut classer, la valeur de la fonction que l'on veut approcher ou de la sortie du processus que l'on veut modéliser, ou

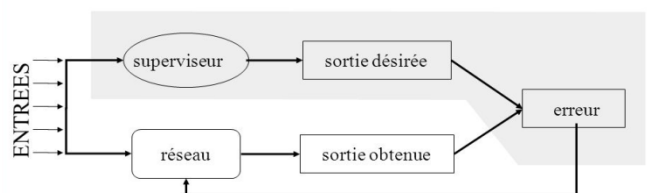


Figure 9 : réseau a apprentissage supervisé

encore la sortie souhaitée du processus à commander. L'apprentissage, est réalisé en général de façon itérative, en minimisant une mesure de l'erreur par les algorithmes suivants : (2)

### 1-7.1.1. Error Back Propagation (EBP) ou Gradual Regression (GR)

L'objectif de cet algorithme est de réduire la zone d'erreur globale jusqu'à ce que les poids appropriés soient obtenus pour la prédiction. Il se compose de trois étapes :

- Distribuer des échantillons en avant
- Distribuer l'erreur à l'envers
- Composition des poids

Utilisés dans la formation de réseaux de neurones à liaison complète avec alimentation frontale, multicouche et non linéaire. On l'appelle aussi un algorithme de régression progressive, où les poids sont ajustés par l'équation de régression progressive Ils sont (8):

$$(\theta_1 - \theta_0) = \alpha * J$$

(Où  $(\theta_1 - \theta_0)$ : Différence de poids ;  $\alpha$  : Learning Rate ;  $J$  : Coefficient de régression ) .

Où Il existe deux manières de calculer la régression progressive:

1: système incrémental

De cette façon, la pente est calculée et les poids sont ajustés après chaque revenu attribué à la grille. (12)

2: Système de paiement unique

Dans ce mode, le réseau fournit tous les rayons d'entrée avant la mise à jour des poids, ce qui permet d'ajuster les poids et déformations de cette méthode après la fourniture du réseau dans l'ensemble du groupe de formation. Les tendances calculées dans chaque exemple de formation sont additionnées pour déterminer les modifications de poids et de dégradations.

### 1-7.1.2. Descente graduelle (Gradient Descent) :

Dans celui-ci les poids se déplacent en face de la fonction active, Dans lequel la distribution inverse doit être conforme à la méthode de calcul de la régression (la pente), par l'équation suivante:  $\theta_1 = \theta_0 - \alpha * J$

Tant que nous recherchons des valeurs  $\theta_0$  et  $\theta_1$  qui réduiront la valeur de  $J$  au minimum, comme les équations suivant :

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m ((h_{\theta}(x_i) - y_i)x_i)$$

Nous imposerons les valeurs  $\theta_0=1$  et  $\theta_1=2$ , puis les réduirons progressivement jusqu'à atteindre la valeur la plus basse de  $J$ . mais nous voulons les valeurs de  $\theta$  globale pas les valeur locale, comme la fig9 suivant. (8)

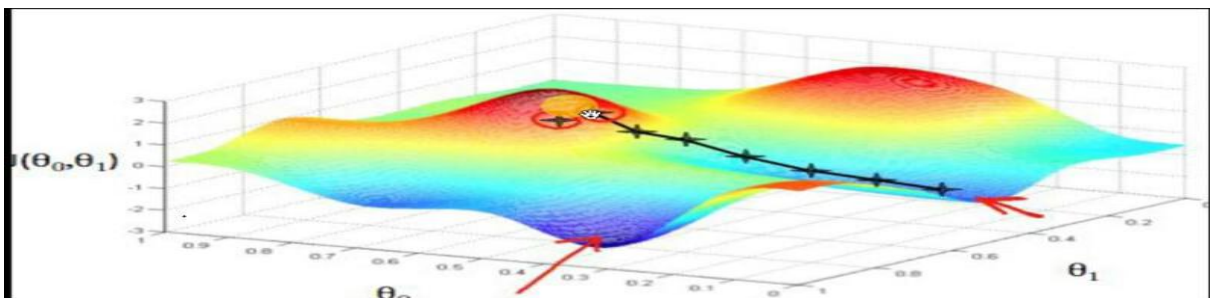


Figure 10 : les valeurs de  $\theta$  globale et les valeur locale

Exemple :

$X_i$	$Y$	$h(x)$	$h(x) - y$
100	300	301	1
95	285	286	1
90	270	271	1
80	240	241	1
80	235	241	6
70	200	211	11
70	205	211	6
60	180	181	1

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i)$$

ثابتنا 0  
ثابتنا 1  
المعادلة

$$h(x) = 1 + 3x$$

$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i)$   
 $\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m ((h_{\theta}(x_i) - y_i)x_i)$

Theta 0 = 1    Theta 1 = 3  
 Theta 0 = 0.993    Theta 1 = 2.48  
 Theta 0 = 0.991    Theta 1 = 2.46  
 ...  
 Theta 0 = 0.825    Theta 1 = 1.772

$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m ((h_{\theta}(x_i) - y_i)x_i)$

Theta 1 = 3 - ((0.002 / 8) \* (2095))  
 Theta 1 = 3 - 0.52 = 2.48

$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i)$

Theta 0 = 1 - ((0.002 / 8) \* (28))  
 Theta 0 = 1 - 0.007 = 0.993

### I-7.1.3. Descente de gradient avec élan (GDM)

Est identique à l'algorithme GD avec l'élan, Dans lequel la distribution inverse doit être conforme à la méthode de calcul de la régression (la pente), par l'équation suivante(II) :

$$\theta_1 = \theta_0 - \alpha * J + \mu(\theta_0 - \theta_{-1})$$

( où  $(\theta_1 - \theta_0)$ : Différence de poids ;  $\alpha$  : Learning Rate ;  $\mu$  : élan ;  $J$  : Coefficient de régression ) .

### I-7.1.4. Conjugate Gradient BP (CGF)

Il se concentre sur le changement de gradient brusque à chaque fréquence afin de tirer profit d'un temps court, Coefficient de régression doit être calculer par l'équation suivante(II) :

$$\theta_1 = \theta_0 - \alpha * J + \mu P_k \quad \text{Où } P_k = -J_k + \beta_k P_{k-1} \text{ et } \beta_k = \frac{J_k^2}{J_{k-1}^2}$$

### I-7.1.5. Quasi-Newton (BFGS)

Coefficient de régression doit être calculer par l'équation suivante(II) :

$$[\theta_{k+1}] = [\theta_k] - [\alpha]_k^{-1} * J_k$$

### I-7.1.6. Levenberg Marquardt (LM)

Coefficient de régression doit être calculer par l'équation suivante(II) :

$$[\theta_{k+1}] = [\theta_k] - [\alpha + \mu]_k^{-1} * J_k$$

Où  $\mu = 0$  donc :LM comme BFGS

## I-7.2 Apprentissage non supervisé

Une forme d'apprentissage effectuée à partir uniquement des données brutes, □ autrement dit L'apprentissage non-supervisé consiste à tirer de la valeur de données dans lesquelles l'attribut à prédire n'est prédéterminée.

Celui-ci est amené à découvrir la structure sous-jacente des données afin de les organiser en clusters.



Ainsi, l'ensemble d'entraînement ne contient que des entrées et ne définit pas explicitement la nature de la fonction (f) qui doit être retournée par l'algorithme d'apprentissage. C'est plutôt l'utilisateur qui doit spécifier le problème à résoudre.

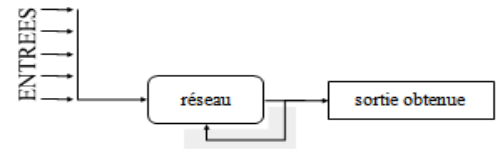


Figure 11 réseau a apprentissage non supervisé

Pour ce problème, (f) doit fournir une estimation de la fonction de densité ou de probabilité de la distribution ayant généré les éléments d'entrées. Pour le problème de la classification les cartes auto-organisatrices de Kohnen sont utilisées dans les réseaux de neurones artificiels. Dans l'apprentissage non supervisé, le réseau modifie ses paramètres en tenant compte seulement des informations locales. Ces méthodes n'ont pas besoins de sorties désirées prédéterminées. Les réseaux utilisant cette technique sont appelés réseaux à dynamique autonome et sont considérés comme des détecteurs de régularité, car le réseau apprend en détectant les régularités dans la structure des motifs d'entrée et produit la sortie la plus satisfaisante. (10)

### 1-7.3 Apprentissage hybride

Ce type d'apprentissage est rarement utilisé et mal exploré. Cette approche combine des méthodes numériques (réseaux de neurones, algorithmes génétiques) et des méthodes symboliques. Certains auteurs utilisent le terme d'apprentissage hybride pour parler d'un couplage 'supervisé non supervisé' ; dans ce cas, il s'agit d'un réseau qui met en parallèle ou en série un réseau entraîné en mode supervisé et un autre en mode non supervisé.

### 1-8. Conclusion :

La conclusion de ce qui précède est que l'IA est un programme intelligent, rapide et précis permettant de résoudre des problèmes mathématique et statistiques et de prévoir les meilleures décisions.

Il comporte de nombreuses branches, la plus importante est l'apprentissage automatique, qui s'intéresse, dans sa partie de la supervision, à la prédiction.

L'algorithme de régression linéaire est un algorithme statistique développé par machine Learning pour devenir la prédiction la plus précise.

Dans le chapitre suivant on essayera de vous présenter une des techniques prédictives, il s'agit des réseaux de neurones artificiels.

### Chapitre2 : Les réseaux de neurones artificiels

#### II-1 Introduction

Depuis une dizaine d'années, l'utilisation des réseaux de neurones artificiels (RNA) s'est développée dans de nombreuses disciplines (sciences économiques, écologie et environnement, biologie et médecine...), ils sont au cœur des progrès récents de l'intelligence artificielle. Ils utilisent notamment des principes mathématiques et des techniques d'analyse en provenance de l'apprentissage automatique (intelligence artificielle) et des bases de données.

Néanmoins, ils n'ont pas encore atteint leur plein développement pour des raisons plus psychologiques que techniques, liées aux connotations biologiques du terme et au fait qu'ils sont considérés, à tort, comme des outils d'Intelligence Artificielle.

Or l'intérêt des réseaux de neurones, dans de nombreuses disciplines ne doit rien à la métaphore biologique, il est uniquement dû aux propriétés mathématiques spécifiques de ces réseaux.

Les réseaux de neurones ont été développés pour résoudre des problèmes de contrôle, de reconnaissance de formes ou de mots, de décision, de mémorisation comme une alternative à l'intelligence artificielle, et en relation, plus ou moins, étroite avec la modélisation de processus cognitifs réels (capable de connaître ou faire connaître) et des réseaux de neurones biologiques. (13)

#### II-1-1 Fondement biologique

Dans le cerveau humain, un neurone typique recueille les signaux des autres à travers une multitude de structures fines appelées dendrites. Le neurone envoie des pics d'activité électrique à travers un long support mince connu sous le nom d'un axone, qui se divise en des milliers de branches. A la fin de chaque branche, une structure appelée synapse convertit l'activité de l'axone et compte les effets électriques qui inhibent ou excitent l'activité de l'axone et compte les effets électriques qui inhibent ou excitent l'activité dans les neurones connectés. Quand un neurone reçoit une entrée excitatrice qui est suffisamment grande par rapport à son entrée inhibitrice, il envoie un pic d'activité électrique vers le bas de son axone. (14)

### II-1-2 Historique

De façon générale, on situe le début des réseaux de neurones artificiels en 1943, avec les travaux de « McCulloch et Pitts » qui montrent qu'un réseau de neurones discret, sans contrainte de topologie, peut représenter n'importe quelle fonction booléenne et donc émuler un ordinateur. En 1958, « Rosenblatt » propose le premier algorithme d'apprentissage, qui permet d'ajuster les paramètres d'un neurone.

En 1969, Minsky et Papert publient le livre 'Perceptrons', dans lequel ils utilisent une solide argumentation mathématique pour démontrer les limitations des réseaux de neurones à une seule couche.

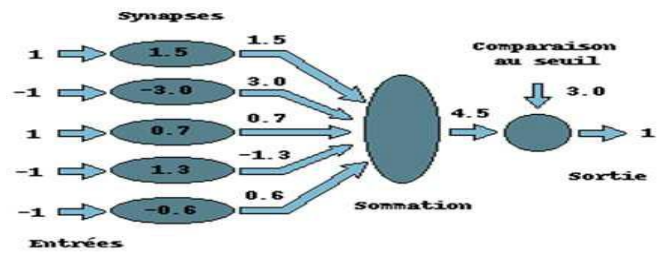


Figure 12 : Neurone de Mac Culloch et Pitts

En 1982, « Hopfield » propose des réseaux de neurones associatifs et l'intérêt pour les réseaux de neurones renaît chez les scientifiques.

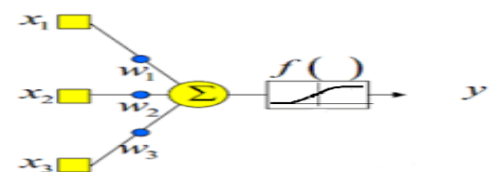
En 1986, « Rumelhart, Hinton et Williams » publient l'algorithme de la 'rétro propagation de l'erreur' qui permet d'optimiser les paramètres d'un réseau de neurones à plusieurs couches. À partir de ce moment, la recherche sur les réseaux de neurones connaît un essor fulgurant et, au cours des années 90, les applications commerciales de ce succès académique suivent. (14)

### II-2 Neurone formel

Le neurone artificiel (ou cellule) est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones appartenant à

Figure 13 : Neurone formel

un niveau situé en amont (on parlera de neurones "amont"). A chacune des entrées est associé un poids  $w$  représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique,



qui se ramifie ensuite pour alimenter un nombre variable de neurones appartenant à un niveau situé en aval (on parlera de neurones "avals"). A chaque connexion est associé un poids. (8)

## II-2- 1 Représentation mathématique

Les réseaux de neurones formels sont à l'origine d'une tentative de modélisation mathématique du cerveau humain (figure 11). Les premiers travaux datent de 1943 sont l'œuvre de MM. Mac Culloch et Pitts. Ils présentent un modèle assez simple pour les neurones et explorent les possibilités de ce modèle. La modélisation consiste à mettre en oeuvre un système de réseau neuronal sous un aspect non pas biologique mais artificiel. Un neurone est donc avant tout un opérateur mathématique, dont on peut calculer la valeur numérique par quelques lignes de logiciel. D'un point de vue mathématique, un neurone formel réalise donc une fonction linéaire. (5)

$$y = f(x_1, x_2, \dots, x_p; w_1, w_2, \dots, w_p)$$

Les  $\{x_i\}$  sont les variables (ou entrées), les  $\{w_j\}$  sont des paramètres (ou poids). Les entrées peuvent être booléennes, binaires (0,1), bipolaire (-1,1) ou réels. En règle générale, le calcul de la valeur de cette fonction peut se décomposer en deux étapes :

-Une combinaison linéaire des entrées

$$U = \sum_{i=1}^n \omega_i x_i - \theta$$

- La sortie du neurone est :

$$Y = \mathcal{F}(U) = \mathcal{F}\left(\sum_{i=1}^n \omega_i x_i - \theta\right)$$

$(\theta)$  est appelé biais, il peut être considéré comme la pondération de l'entrée 0 fixée à 1 ( $U$ ) est appelé potentiel du neurone.

La fonction  $(f)$  permet de définir l'état interne du neurone en fonction de son entrée Totale.

Citons à titre d'exemple quelques fonctions d'activation ou de transfert souvent utilisées.

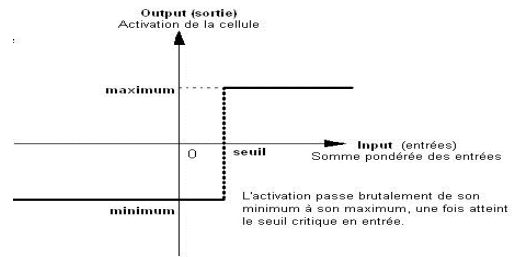
## II-2-2 Fonctions d'activation

Dans sa première version, le neurone formel était donc implémenté avec une fonction à seuil, mais de nombreuses versions existent. Ainsi le neurone de McCulloch et Pitts a été généralisé de différentes manières, en choisissant d'autres fonctions d'activations. La sortie du neurone dépend d'une fonction de transfert, dont les principales sont(15) :

**a. fonction binaire à seuil**

$$h(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases}$$

$$sgn(x) = \begin{cases} -1 & \text{si } x < 0 \\ 1 & \text{sinon} \end{cases}$$



Le seuil introduit une non-linéarité dans le comportement du neurone, cependant il limite la gamme des réponses possibles à deux valeurs.

**b. Fonction linéaire**

C'est l'une des fonctions d'activations les plus simples, sa fonction est définie par :  $F(x)=x$

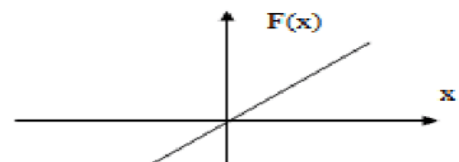


Figure 15 : Fonction linéaire

**c. Fonction linéaire à seuil ou multi seuil**

Cette fonction représente un compromis entre la fonction linéaire et la fonction seuil, entre ses deux barres de saturation, elle confère au neurone une gamme de réponses possibles. En modulant la pente de la linéarité, on affecte la plage de réponse du neurone

$$F(x) = \begin{cases} x & x \in [u, v] \\ v & \text{si } x \geq v \\ u & \text{si } x \leq u \end{cases}$$

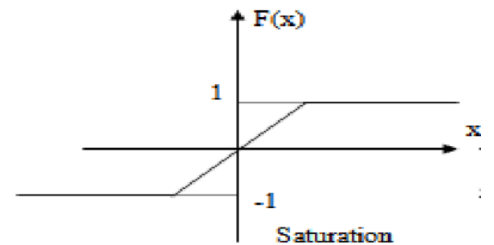
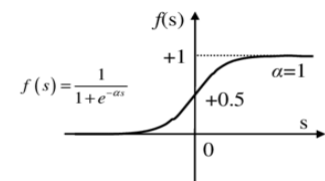


Figure 16 : Fonction linéaire a seuil

**d. La fonction sigmoïde**

Elle est l'équivalent continu de la fonction linéaire. Étant continu, elle est dérivable, d'autant plus que sa dérivée est simple à calculer, elle est définie par :

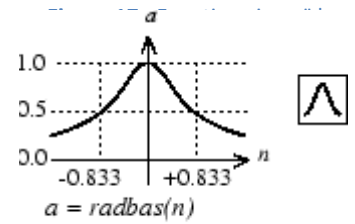
$$F(x) = \frac{1}{1 + e^{-x}}$$



### e. La fonction de transfert de base radiale

Radbasis (Radial basis transfer function) est une fonction de transfert neural. Les fonctions de transfert calculent la sortie d'une couche à partir de son entrée nette.

Syntaxe :  $a = \text{radbas}(n) = \exp(-n^2)$



Radial Basis Function

Figure 18 : Radial basis transfer function

Un neurone formel ne réalise donc rien d'autre qu'une somme pondérée suivie d'une non linéarité. C'est l'association de tels éléments simples sous la forme de réseaux qui permet de réaliser des fonctions utiles pour des différentes applications.

## II-3 Réseau de neurones artificiel (RNA)

### II-3-1 Définition

Un RNA (Réseau de Neurones Artificiels) est un ensemble de neurones formels (d'unités de calcul simples, de nœuds processeurs) associés en couches (ou sous-groupes) et fonctionnant en parallèle. Dans un réseau, chaque sous-groupe réalise un traitement indépendamment des autres et transmet le résultat de son analyse au sous-groupe suivant. L'information donnée au réseau va donc se propager couche par couche, de la couche d'entrée à la couche de sortie, en passant soit par aucune, soit par plusieurs couches intermédiaires (dites couches cachées). Habituellement (excepté pour les couches d'entrée et de sortie), chaque neurone dans une couche est connecté à tous les neurones de la couche précédente et de la couche suivante. Les RNA ont la capacité de stocker de la connaissance empirique et de la rendre disponible à l'usage. Les habiletés de traitement (et donc la connaissance) du réseau vont être stockées dans les poids synaptiques, obtenus par des processus d'adaptation ou d'apprentissage. (16)

## II-3-2 Architecture d'un réseau de neurones artificiel

La structure générale des réseaux de neurones artificiels comprend les composants de base suivants : Couche d'entrée, couche de sortie, couche cachée, interconnexion (poids), unités de traitement (neurones), fonction d'addition, Fonctions d'activation.

Où certains réseaux ne contiennent pas tous ces éléments

Les connexions entre les neurones qui composent le réseau décrivent la topologie du modèle qui peut être quelconque, mais le plus souvent il est possible de distinguer une certaine régularité (réseau à connexion complète).(17)

### II-3-2-1 Réseaux monocouche

La structure d'un réseau monocouche est telle que des neurones organisés en entrée soient entièrement connectés à d'autres neurones organisés en sortie par une couche modifiable de poids. Tels que la perception et le réseau Hopfield(9) .

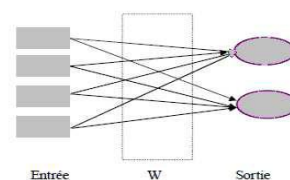


Figure 19 : réseau de neurones monocouche

### II-3-2-2 Réseaux multicouche

Les neurones sont arrangés par couche. Il n'y a pas de connexion entre les neurones d'une même couche, et les connexions ne se font qu'avec les neurones de couches avales. Habituellement, chaque neurone d'une couche est connecté seulement à tous les neurones de la couche suivante. Ceci nous permet d'introduire la notion de sens de parcours de l'information (de l'activation) au sein d'un réseau et donc définir les concepts de neurone d'entrée, neurone de sortie. Par extension, on appelle couche d'entrée l'ensemble des neurones d'entrée, couche de sortie l'ensemble des neurones de sortie. Les couches intermédiaires n'ayant aucun contact avec l'extérieur sont appelées couches cachées. (17)

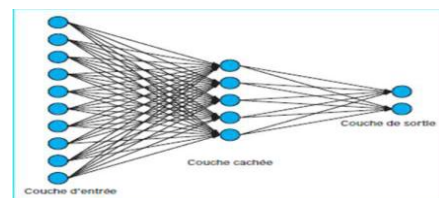


Figure 20: Réseau multicouche classique

#### II-3-2-2.1 Réseaux multicouche classique

Dans un réseau multicouche classique, il n'y a pas de connexion entre neurones d'une même couche et les connexions ne se font qu'avec les neurones de la couche aval. Tous les neurones de la couche amont sont connectés à tous les neurones de la couche aval. (5)

Tels que Perceptron Multicouche(MLP) est un réseau comportant (L) couches, chaque neurone d'une couche étant totalement connecté aux neurones de la couche suivante. Chaque neurone (k) est un automate linéaire généralisé dont la fonction de transfert est supposée sigmoïdale. L'algorithme d'apprentissage par rétro-propagation du gradient de l'erreur est un algorithme itératif qui a pour objectif de trouver le poids des connexions minimisant l'erreur quadratique moyenne commise par le réseau sur l'ensemble d'apprentissage. Cette minimisation par une méthode du gradient conduit à l'algorithme d'apprentissage de rétro propagation.

Cet algorithme, qui présente l'avantage d'exister, reste discutable dans la mesure où sa convergence n'est pas prouvée. Son utilisation peut conduire à des blocages dans un minimum local de la surface d'erreur. Son efficacité dépend, en effet, d'un grand nombre de paramètres que doit fixer l'utilisateur : le pas du gradient, les paramètres des fonctions sigmoïdes des automates, l'architecture du réseau ; nombre de couches, nombre de neurones par couche..., l'initialisation des poids.. (18).

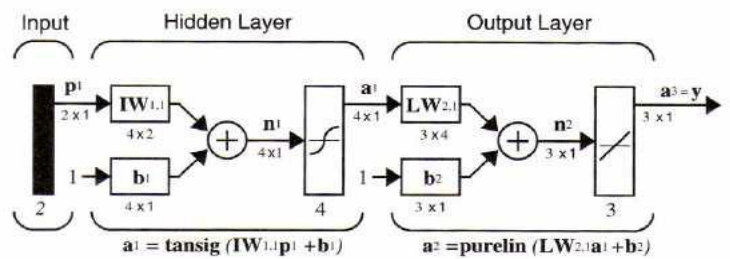


Figure 21 : Architecture Perceptron Multicouche(MLP)

Tels que Réseau d'alimentation avant ( Feed Forward Neural Networks -FFNN), Signifie que n'importe quelle cellule du couche d'entrée est envoyée à toutes les cellules du couche du milieu, Les cellules de la couche intermédiaire sont envoyées à chaque cellule de la couche en sortie.

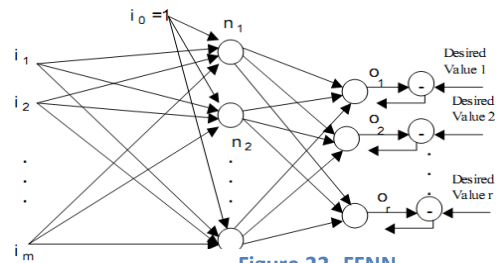


Figure 22 -FFNN

Aussi bien Le réseau de neurones en cascade (CNN) Mais il contient une liaison de poids de chaque couche à toutes les couches suivantes. Il s'agit d'un réseau d'alimentation directe qui utilise l'algorithme de rétro-propagation pour ajuster les poids ainsi que des fonctions log-sigmoid et tan-sigmoid pour atteindre l'état optimal.

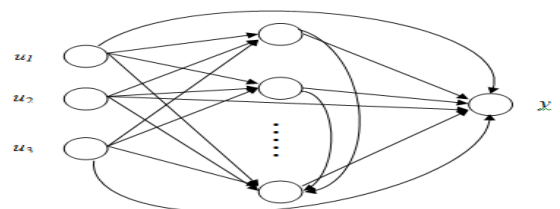


Figure 23 CNN



Aussi Réseau neuronal Elman (ENN), En ENN, toutefois, les cellules de la couche intermédiaire sont envoyées vers chaque cellule de la couche d'entrée ainsi que les cellules de la couche de sortie.

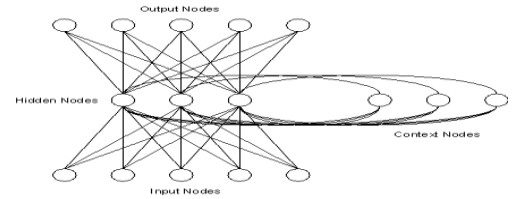


Figure 24 ENN

Aussi bien Réseau de fonctions de base radiale (RBFN), Mais il convertit les entrées de manière non linéaire via la Fonctions radiales gaussiennes, Bien qu'il s'agisse d'un réseau d'alimentation front-end, il compresse les données entre la couche d'entrée et la couche cachée et combine deux types

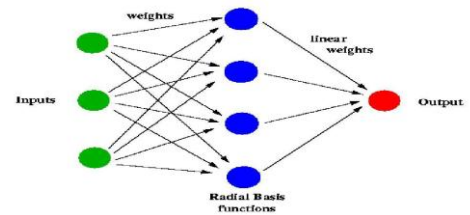


Figure 25 RBFN

d'apprentissage, Entre la couche d'entrée et la couche cachée, vous apprenez sans supervision, Entre la couche cachée et la couche de sortie, elle apprend sous supervision .(19)

### II-3-2-2.2.Réseau à connexions locales :

C'est aussi un réseau multicouche, mais tous les neurones d'une couche amont ne sont pas connectés à tous les neurones de la couche aval. Nous avons donc dans ce type de réseau de neurones un nombre de connexions moins important que dans le cas du réseau de neurones multicouche classique. (5) Tels que Les cartes auto-organisatrices de Kohonen Ce type de réseau, appelé aussi SOM (Self Organised Maps) est un réseau à apprentissage non supervisé qui établit une carte discrète, ordonnée topologiquement en fonction de patterns d'entrée. Le réseau forme ainsi une sortie de treillis, où chaque nœud du treillis est un neurone associé à un vecteur de poids. La correspondance entre chaque vecteur de poids est calculée pour chaque entrée. Par la suite, le vecteur de poids ayant la meilleure corrélation, ainsi que certains de ses voisins vont être modifiés afin d'augmenter encore cette corrélation. Les LVQ (Learning Vector Quantization) sont un cas particulier des SOM, où seul le nœud ayant la meilleure corrélation est adapté. On parle pour ce type de réseau, où seul le 'vainqueur' est sélectionné, de réseaux à compétition. Une carte auto-organisatrice est un procédé qui convertit un signal d'entrée complexe (plusieurs variables par exemple) en une nouvelle variables catégorielle c'est donc un procédé de classification (modélisation non-supervisée). Les SOM sont une généralisation de l'analyse en composantes principales. Elle fonctionne comme un réseau de neurones sans variable cible et avec plusieurs nœuds dans la couche de sortie. La carte structure les nœuds en sortie en classes de nœuds. (20)

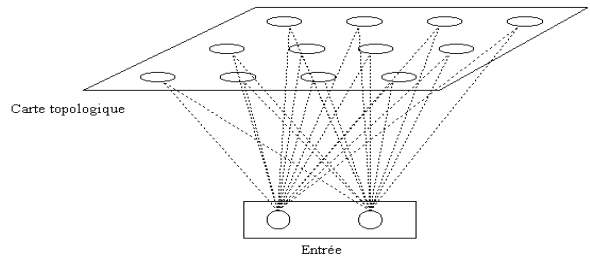


Figure 26: Schéma d'une carte auto-organisatrice SOM

### II-3-2-2.3.Réseau à connexions récurrentes :

Un réseau de ce type signifie qu'une ou plusieurs sorties de neurones d'une couche aval sont connectées aux entrées des neurones de la couche amont ou de la même couche. Ces connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouche. Les réseaux à connexions récurrentes sont des réseaux plus puissants car ils sont séquentiels plutôt que combinatoires comme

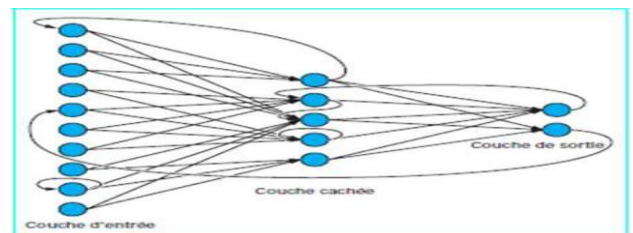


Figure 27: Réseau à connexions récurrentes

l'étaient ceux décrits précédemment. La rétroaction de la sortie vers l'entrée permet à un réseau de ce type de présenter un comportement temporel. (16)

### II-3-3 L'apprentissage des réseaux de neurones

La plupart des algorithmes d'apprentissage des réseaux de neurones sont des algorithmes d'optimisation : ils cherchent à minimiser, par des méthodes d'optimisation non linéaire, une fonction de coût, qui constitue une mesure de l'écart entre les réponses réelles du réseau et ses réponses désirées. Cette optimisation se fait de manière itérative, en modifiant les poids en fonction du gradient de la fonction de coût : le gradient est estimé par une méthode spécifique aux réseaux de neurones, dite méthode de rétro propagation, puis il est utilisé par l'algorithme d'optimisation proprement dit. Les poids sont initialisés aléatoirement avant l'apprentissage, puis modifiés itérativement, jusqu'à obtention d'un compromis satisfaisant entre la précision de l'approximation sur l'ensemble d'apprentissage et la précision de l'approximation sur un ensemble de validation, distinct du précédent. Contrairement à des affirmations maintes fois répétées, l'apprentissage des réseaux de neurones n'est pas spécialement lent : il existe des algorithmes d'optimisation non linéaire extrêmement rapides qui permettent de faire des développements industriels sur de simples PC. Le but fondamental de l'apprentissage est de bien généraliser à de nouveaux cas. (9)

#### II-3-3-1 Règles d'apprentissage des réseaux de neurones artificiels

L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. Cette modification consiste à modifier les poids des connexions entre les neurones en suivant certaines. La loi de Hebb (1949) s'applique aux connexions entre neurones, tel que :

(i) : le neurone amont, (j) : le neurone aval et ( $w_{ij}$ ) : le poids de la connexion.

Elle s'exprime de la façon suivante :

**"Si 2 cellules sont activées en même temps alors la force de la connexion augmente".**

La modification de poids dépend de la coactivation des neurones pré synaptique et post synaptique. (9)

$x_i$  et  $x_j$  sont respectivement les valeurs d'activation des neurones  $i$  et  $j$ .

$\partial w_{ij}$  (dérivée partielle du poids) correspond à la modification de poids réalisée.

La loi de Hebb peut être modélisée par les équations suivantes :

$w(t+1)$  est le nouveau poids

$w_{ij}(t)$  l'ancien

$$w_{ij}(t+1) = w_{ij}(t) + \partial w_{ij}(t)$$

$\partial w_{ij}(t) = x_i \cdot x_j$  (l'activité est modélisée comme le produit des deux valeurs d'activation)

### II-3-3-2 Algorithme d'apprentissage

L'algorithme d'apprentissage modifie de façon itérative (petit à petit) les poids pour adapter la réponse obtenue à la réponse désirée. Il s'agit en fait de modifier les poids en cas d'erreur seulement.

1/ Initialisation des poids et du seuil  $S$  à des valeurs (petites) choisies au hasard.

2/ Présentation d'une entrée  $E1 = (e_1, \dots, e_n)$  de la base d'apprentissage.

3/ Calcul de la sortie obtenue  $x$  pour cette entrée :  $a = \sum (w_i \cdot e_i) - S$

la valeur de seuil est introduite ici dans le calcul de la somme pondérée

4/ Si la sortie ( $x$ ) est différente de la sortie désirée ( $d$ ) pour cet exemple d'entrée  $E1$  alors modification des poids ( $\mu$  est une constante positive, qui spécifie le pas de modification des poids) :  $w_{ij}(t+1) = w_{ij}(t) + \mu \cdot (x_i \cdot x_j)$

5/ Tant que tous les exemples de la base d'apprentissage ne sont pas traités correctement (i.e. modification des poids), retour à l'étape 2.

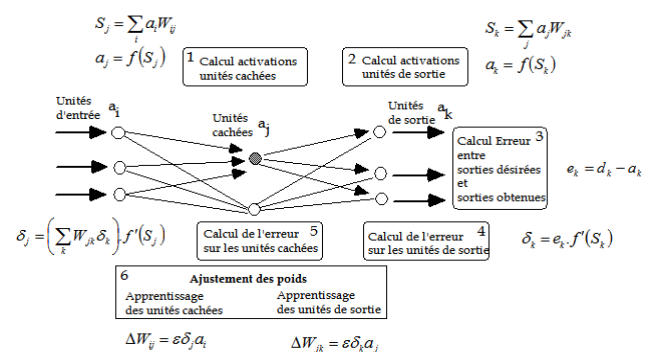


Figure 28 : Algorithme d'apprentissage du perceptron

### Exemple Algorithme d'apprentissage Perceptron Multicouche(MLP)

Les algorithmes d'apprentissage ont pour objectif de permettre à la machine d'acquérir des connaissances opérationnelles à partir d'exemples. Ils cherchent à découvrir la structure cachée d'un ensemble de signaux à partir des valeurs observées, les facteurs inhérents qui expliquent les variations et les dépendances observées. Ils sont utiles pour la prédiction, la prise de décision, et l'analyse de données.

a) Calcul d'activations dans les unités cachées  $S_j = \sum \omega_{ji} a_i$ ,  $a_j = f(S_j)$

b) Calcul d'activations dans les unités de sortie  $S_k = \sum \omega_{jk} a_j$ ,  $a_k = f(S_k)$

c) Calcul d'erreur entre les sorties désirées et les sorties obtenues  $e_k = d_k - a_k$

d) Calcul de l'erreur sur les unités de sortie

$$\delta_k = e_k f'(S_k)$$

**e)** Calcul de l'erreur sur les unités cachées  $\delta_k = \sum_k \omega_{jk} \delta_k f'(S_k)$

**f)** Ajustement des poids

-Apprentissage des unités cachées :  $\Delta W_{ij} = \epsilon \delta_j a_i$

-Apprentissage des unités de sorties :  $\Delta W_{jk} = \epsilon \delta_k a_j$

### Exemple Algorithme d'apprentissage SOM

lieu, la grille doit être initialisée de manière aléatoire. Un cycle d'apprentissage est constitué des étapes suivantes :

**a)** Présenter un vecteur d'entrée associé à un stimulus à la grille.

**b)** Trouver le noeud gagnant (ou *winner*). C'est l'unité dont le vecteur associé est le plus similaire au vecteur d'entrée.  $\| \text{entrée} - \text{neurone winner} \| = \text{Mini} \| \text{entrée} - \text{neurone } i \|$

**c)** Modifier les poids  $W_i$  du noeud gagnant, ainsi que ceux de son entourage, de manière à ce que les vecteurs associés (les vecteurs de poids) «se rapprochent d'avantage» du vecteur d'entrée présenté à la grille. La règle de modification est la suivante :

$$w_i(t+1) = w_i(t) + h(r,t)(p_i - w_i(t)) \text{ si } i \text{ voisinage}$$

$$w_i(t+1) = w_i(t) \text{ si } i \text{ voisinage}$$

$$\text{Avec } h(r,t) = (t) \cdot v(t)$$

$(t)$  : Le taux d'apprentissage.

$v(t)$  : la fonction de voisinage)

**d)** Faire décroître la taille de la zone de voisinage des noeuds gagnants (la zone qui contient les neurones subissant la transformation).

**e)** Faire décroître le coefficient d'apprentissage,  $(t)$ , qui contrôle l'importance des modifications appliquées aux vecteurs de poids.

**f)** Arrêter l'apprentissage si le coefficient d'apprentissage est nul, sinon, présenter un autre stimulus à la grille. La modification des vecteurs associés aux unités se fait de manière différente selon la position des noeuds par rapport à l'unité gagnante. Le noeud gagnant sera celui dont le vecteur subira le plus de modifications, tandis que les unités plus éloignées seront moins affectées. La fonction de voisinage  $v(r)$  va être maximale pour  $r=0$  et décroître quand  $r$  croît (quand on s'éloigne du noeud gagnant).

### II-3-4. Training

Le RNA est similaire au cerveau humain, il obtiens des connaissances sur l'entraînement et les stocke en utilisant des forces de connexion au sein des neurones appelées poids associatifs.

Après avoir déterminé les valeurs initiales de poids et de dégradations, le réseau les exerce d'une manière répétitive jusqu'à atteindre la valeur minimale du coût ou fonction dite de performance.

Les performances par défaut des réseaux à alimentation frontale correspondent à la zone d'erreur moyenne (mse). Les poids constituent la principale information utilisée par le réseau pour apprendre. Par conséquent, les poids doivent être mis à jour pendant la phase de Training moyennant des algorithmes d'apprentissage. (II)

### II-4 Conclusion

Actuellement, les réseaux de neurones sont solidement implantés dans diverses industries, entre autres : dans les milieux financiers, pour la prédiction des fluctuations de marché ; en pharmaceutique, pour analyser le 'QSAR' (Quantitative Structure-Activity Relationship) ; pour la détection de fraudes sur les cartes de crédit et le calcul de cotes de crédit; dans les départements de marketing de compagnies de diverses industries, pour prévoir le comportement des consommateurs ; en aéronautique, pour la programmation de pilotes automatiques.

Les applications du RNA sont nombreuses et partagent toutes un point commun essentiel : les processus pour lesquels on désire émettre des prédictions comportent de nombreuses variables explicatives et, surtout, il existe éventuellement des dépendances non linéaires de haut niveau entre ces variables qui peuvent servir à l'amélioration de la prédiction du processus, si elles sont découvertes et exploitées..

### Chapitre 3. Notre approche

#### Introduction :

Le présent chapitre est un cas pratique de tous ce qu'on déjà présenté dans les deux chapitres précédents

Afin de créer une fonction ou un programme, nous entrons une variable de temps et nous donnons ensuite la valeur réelle de la quantité de rayonnement solaire durant cette période. Comme l'entrée est une variable unique (un temps) et le travail requis consiste à prévoir et non à estimer, nous appliquerons donc les étapes d'apprentissage sous surveillance suivants :

1. Importation des bibliothèques et des fonctions
2. Chargement des données
3. Traitement des données
4. définition de l'architecture du modèle
5. Adaptation du modèle aux données d'entraînement
6. évaluation du modèle sur les données de test
7. prédire des valeurs

#### III-1- Importation des bibliothèques et des modules

Le langage de programmation utilisé est Matlab ID car il appelle des fonctions prédéfinies plutôt des bibliothèques comme certains langages tels que Python , C et R. Les plus importantes fonctions des réseaux de neurones sont:

- Fonctions pour créer un réseau de neurones par type.

```
feedforwardnet(hiddenSizes,trainFcn)
cascadeforwardnet(hiddenSizes,trainFcn)
elmannet(layerDelays,hiddenSizes,trainFcn)
fitnet(hiddenSizes,trainFcn)
linearlayer(inputDelays,widrowHoffLR)
newrb(P,T,goal,spread,MN,DF)
perceptron(hardlimitTF,perceptronLF)
selforgmap(dimensions,coverSteps,initNeighbor,topologyFcn,distanceFcn)
```

- Fonctions de formation

`train()` :Train neural network(Former un réseau de neurones)

`trainb()` : Entraînement par lots avec règles d'apprentissage du poids et des préjugés.

`trainc()` : Mise à jour incrémentielle d'ordre cyclique.

`Trainrp()` : Rétropropagation résiliente

traingd() : Gradient descent backpropagation

trainгда() : Gradient descent backpropagation avec règle d'apprentissage adaptatif

trainoss() : Une étape sécante backpropagation

trainr() : Entraînement incrémental d'ordre aléatoire avec fonctions d'apprentissage

trains() : Entraînement incrémental d'ordre séquentiel avec fonctions d'apprentissage

Fonctions de performance

mae(E,Y,X,FP) : Mean absolute error (Erreur absolue moyenne =  $\frac{1}{n} \sum_{t=1}^n e_t$  )

mse(E,Y,X,FP) : Mean squared error (Erreur quadratique moyenne =  $\frac{1}{n} \sum_{t=1}^n e_t^2$  )

sse(net.t,y,ew) : Sum squared error performance function (Fonction de performance d'erreur au total =  $\sum_{t=1}^n e_t^2$  )

Fonctions de traçage

plotep : Tracer le poids et la position de biais sur la surface d'erreur

ploterrcorr : Tracé, autocorrélation des séries temporelles d'erreur

plotregression : Tracé de régression linéaire

### III-2- chargement des données

Une base de données obtenue du site web : » [www.soda-pro.com/web-services/meteo-data](http://www.soda-pro.com/web-services/meteo-data) », comportant des données résultantes d'une surveillance de la durée de luminosité solaire durant la période allant du 01/01/2013 et du 31/10/2018.

La référence de temps en unité d'heures, ainsi 5 colonnes définissant respectivement : les valeurs de température, pression, vitesse du vent, direction du vent, précipitations et les rayonnements solaires

Notre objectif est d'établir un réseau de neurones que nous permettront de prédire l'ampleur du rayonnement solaire à tout moment.



# Date	Time	Temperature	Relative Humidity	Pressure	Wind speed	Wind direction	Rainfall	GH
01/01/2013	01:00	279.81	54.48	989.91	3.97	53.13	0.000000	0.0000
01/01/2013	02:00	279.24	56.30	989.30	3.87	59.55	0.000000	0.0000
01/01/2013	03:00	278.65	57.71	988.83	3.65	64.73	0.000000	0.0000
01/01/2013	04:00	278.07	58.69	988.53	3.42	68.93	0.000000	0.0000
01/01/2013	05:00	277.59	59.12	988.47	3.31	74.38	0.000000	0.0000
01/01/2013	06:00	277.19	59.45	988.73	3.25	76.67	0.000000	0.0000
01/01/2013	07:00	276.80	60.33	989.25	3.18	74.17	0.000000	0.4838
01/01/2013	08:00	278.17	54.86	989.91	3.32	69.23	0.000000	94.2078
01/01/2013	09:00	280.79	46.33	990.61	3.94	62.38	0.000000	292.4175
01/01/2013	10:00	283.61	39.00	991.03	2.81	53.80	0.000000	462.3661
01/01/2013	11:00	287.38	31.43	990.48	2.23	57.56	0.000000	512.4417
01/01/2013	12:00	288.88	28.00	989.50	2.06	41.32	0.000000	417.7521
01/01/2013	13:00	289.56	26.59	988.54	2.31	19.62	0.000000	384.5432
01/01/2013	14:00	290.07	25.81	987.75	2.96	7.19	0.000000	435.4280
01/01/2013	15:00	290.30	25.58	987.23	3.47	9.86	0.000000	434.9177
01/01/2013	16:00	289.92	26.42	987.00	3.80	14.69	0.000000	298.2065
01/01/2013	17:00	287.27	31.83	987.11	2.87	13.09	0.000000	100.8390
01/01/2013	18:00	283.99	40.55	987.43	3.49	12.23	0.000000	0.8875
01/01/2013	19:00	283.20	44.26	987.95	3.81	15.70	0.000000	0.0000
01/01/2013	20:00	282.72	47.35	988.33	3.95	20.12	0.000000	0.0000
01/01/2013	21:00	282.30	50.19	988.61	3.99	25.95	0.000000	0.0000
01/01/2013	22:00	281.84	52.93	988.83	3.95	32.30	0.000000	0.0000
01/01/2013	23:00	281.38	55.36	988.92	3.78	38.17	0.000000	0.0000
01/01/2013	24:00:00	280.97	57.20	988.66	3.64	43.69	0.000000	0.0000
02/01/2013	01:00	280.53	58.77	988.26	3.47	48.57	0.000000	0.0000
02/01/2013	02:00	280.11	59.77	987.89	3.31	51.66	0.000000	0.0000
02/01/2013	03:00	279.50	61.11	987.59	3.16	51.21	0.000000	0.0000
02/01/2013	04:00	278.81	62.82	987.37	3.01	45.38	0.000000	0.0000

Figure 29 : base de données brut

# Site latitude (positive means North)	27.880							
# Site longitude (positive means East)	-0.280							
# Site altitude (m)	293							
# Date beginning	01/01/2013							
# Date end	31/10/2018							
# Time reference (hour)	UT							
# Summarization (period of integration)	Hour (h)							
# Date - Instant	The date or instant given for each value corresponds to the end of the summarization							
# Columns:								
# Temperature (K)	Temperature at 2 m above ground							
# Relative humidity (%)	Relative humidity at 2 m above ground							
# Pressure (hPa)	Pressure at ground level							
# Wind speed (m/s)	Wind speed at 10 m above ground							
# Wind direction (deg)	Wind direction at 10 m above ground (0 means from North, 90 from East...)							
# Rainfall (kg/m2)	Rainfall (= rain depth in mm)							
# Short-wave irradiation (Wh/m2)	Surface incoming shortwave irradiation (broadband)							

Figure 30 : Informations sur la base de données

### III-3- Traitement des données

#### III- 3-1- Initialisation de données par l'outil Excel:

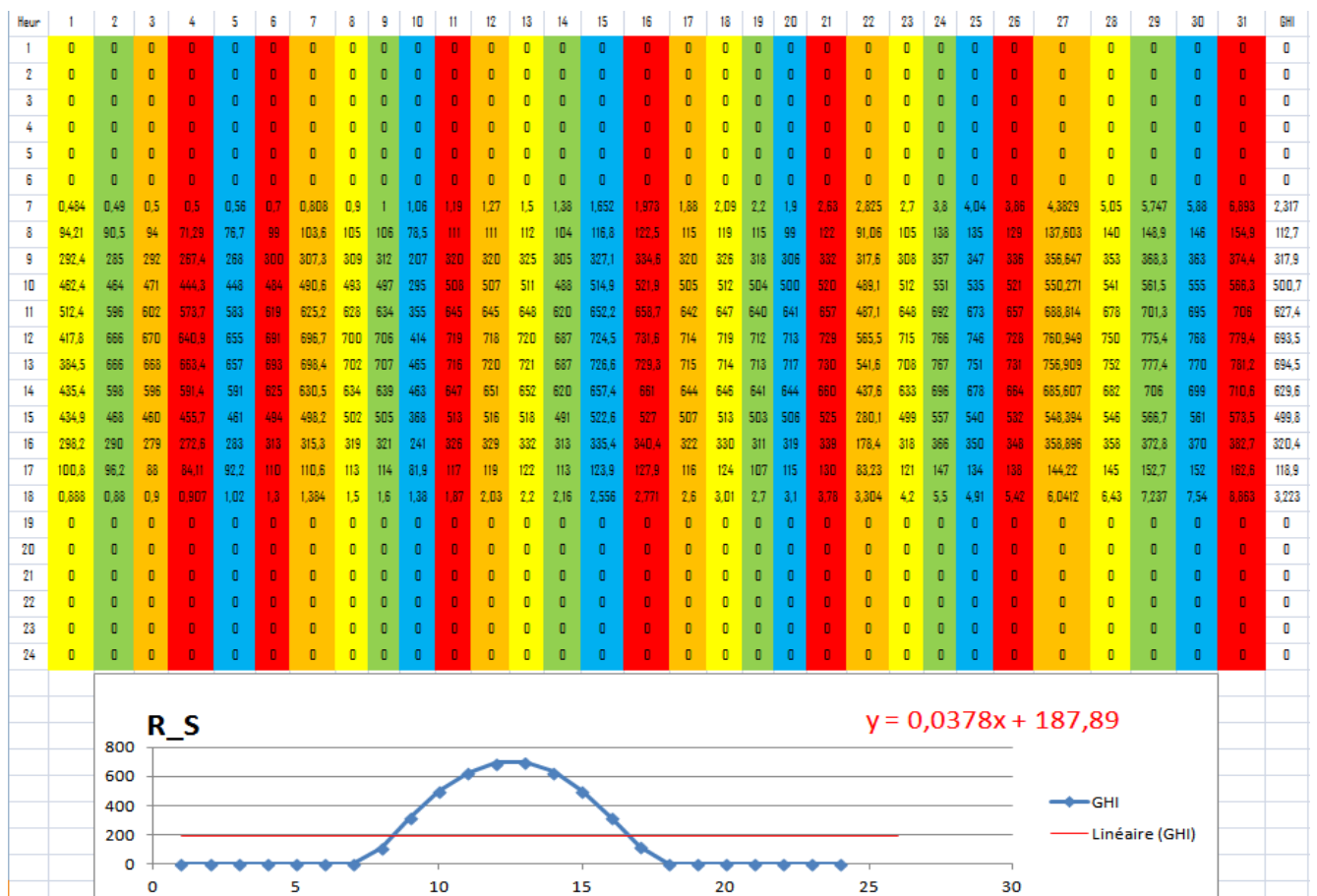


Figure 31 : Traitement informatique de la quantité de rayonnement en fonction de la variable horaire

Cette figure montre le processus de recherche de la moyenne des valeurs de R\_S(GHI) par heure. Nous avons calculé les valeurs horaires en janvier et nous les avons divisées par 31 : nombre de jours du mois concerné. La courbe est Convexe et non rectiligne, ce qui la rend la prévision difficile moyennant la technique de régression linéaire.

2  
0  
1  
3

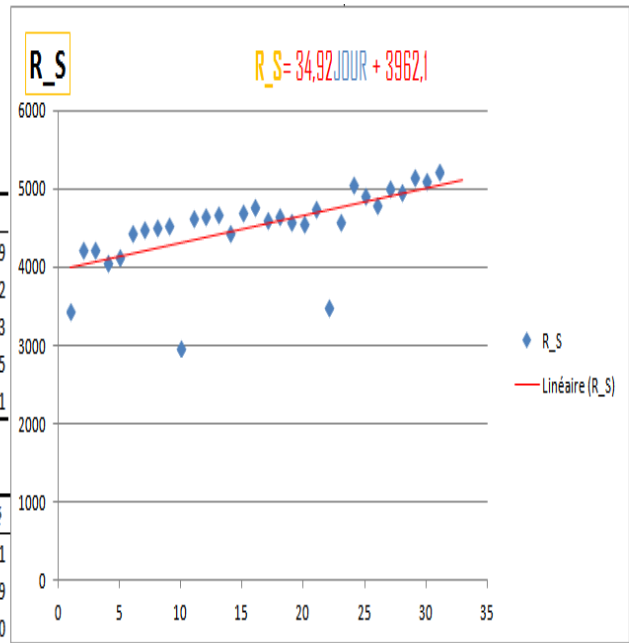
Janvier: R\_S = 140145,215

Jour	R_S
1	3434,4909
2	4220,5452
3	4220,7438
4	4066,1973
5	4116,4424
6	4429,7512
7	4478,6362
8	4506,2943
9	4541,8787
10	2970,6267
11	4626,5646
12	4640,3818
13	4664,1113
14	4430,6489
15	4705,5206
16	4759,6034
17	4603,9417
18	4653,673
19	4569,1956
20	4564,6021
21	4749,6386
22	3477,4726
23	4572,878
24	5046,9118
25	4899,2323
26	4794,7564
27	4998,7339
28	4956,423
29	5144,0616
30	5093,7122
31	5207,5453

Statistiques de la régression	
Coefficient de détermination multiple	0,632323819
Coefficient de détermination R^2	0,399833412
Coefficient de détermination R^2	0,379138013
Erreur-type	395,6374985
Observations	31

ANALYSE DE VARIANCE	
Degré de liberté	
Régression	1
Résidus	29
Total	30



	Coefficients	Erreur-type	Statistique t	Probabilité	Limite inférieure pour seuil de confiance = 95%	pour seuil		
19	4569,1956	Constante	3962,093393	145,6267796	27,2071758	3,40211E-22	3664,253191	4259,9331
20	4564,6021	Mois	34,92000044	7,944592151	4,395442809	0,000135502	18,67148533	51,168511

Figure 32 : Traitement de données de la quantité de rayonnement en termes de variable de temps quotidien

Cette figure montre la base de données de la collection de valeurs de R\_S pour chaque jour de janvier, la courbe étant en ligne droite, ce qui facilite la prévision par la technique de régression linéaire.

III-3-2-Analyse de données avec les outils habituels de Matlab:

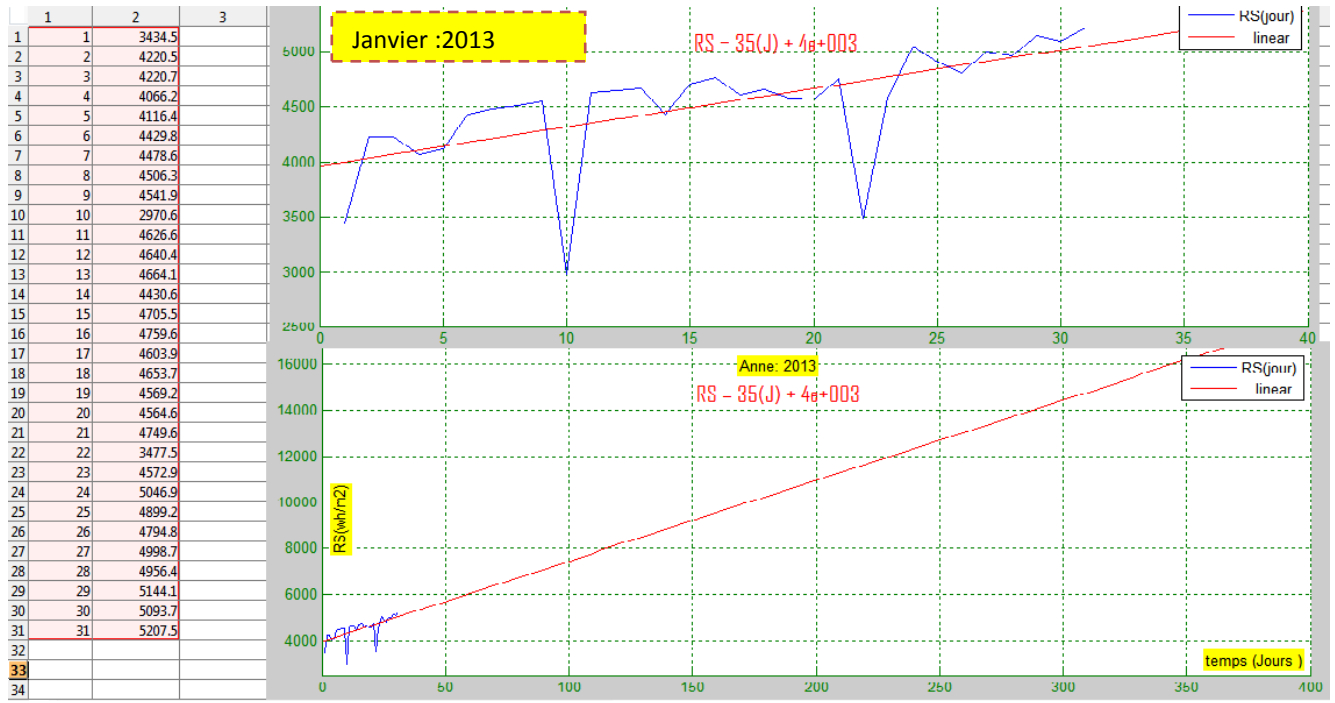


Figure 33 : régression linéaire de Quantité de rayonnement solaire pendant 40 et 400 jours

Cette figure montre le processus de prévision de la fonction linéaire du graphique. Le processus de prévision était à court terme et à long terme et a atteint le 31 décembre 2013(X=366jour) = près de 16 000, ce qui est une valeur extrêmement imaginaire.

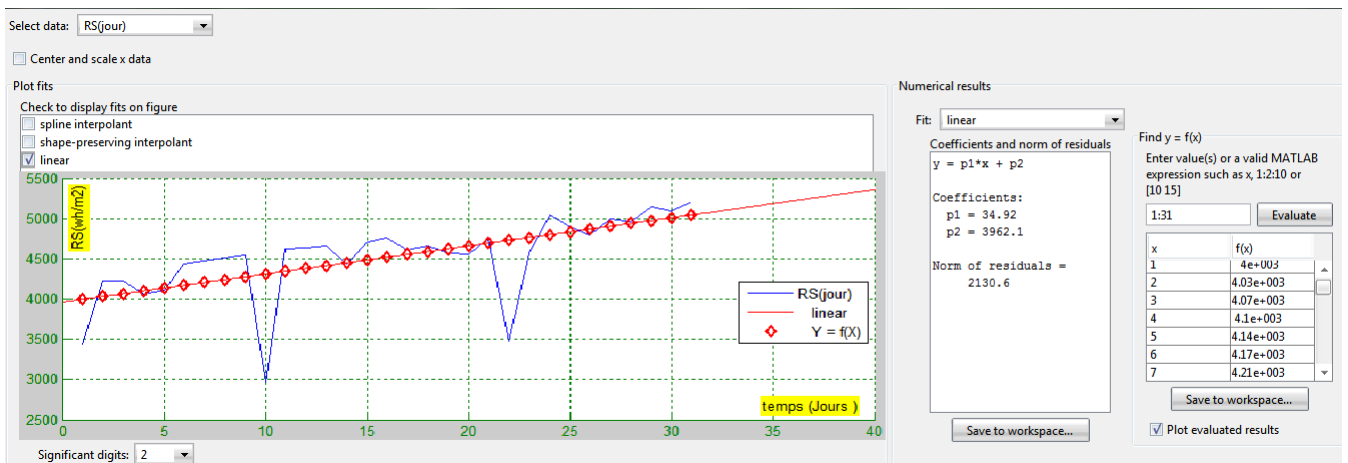


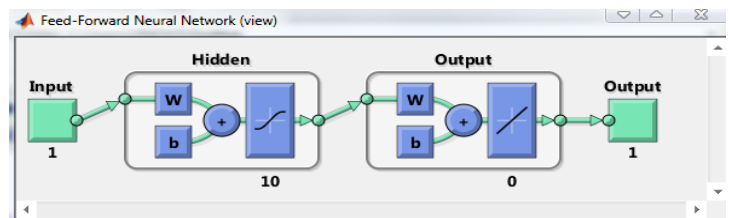
Figure 34 : Mise en œuvre de la prédiction linéaire des bibliothèques Matlab à l'aide de méthodes statistiques traditionnelles

## III-4- Définition de l'architecture du modèle

Puisque notre objectif est de prédire l'utilisation d'une base de données chronologiques, les modèles de réseaux de neurones les plus appropriés sont : FFNN, CNN, ENN, RBFN, tout en prenant en considération les caractéristique du support de travail car nous utilisons un ordinateur de type A 3 et RAM 4 GB qui ne peut pas les traité en parallèle.

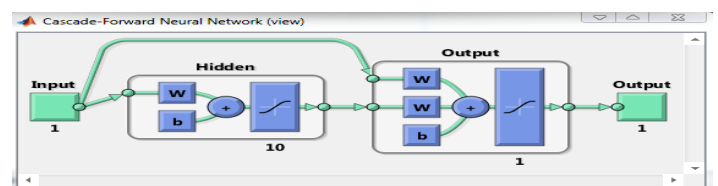
### III-4-1. l'architecture du Feed Forward Neural Networks (FFNN)

```
tab=xlsread('E:\proj2\Anne.xlsx','Feuil2','C2:D32');
[x,t] =size(tab);
FFN = feedforwardnet(10,'trainlm')
FFN = train(FFN,x,t);
view(FFN)
```



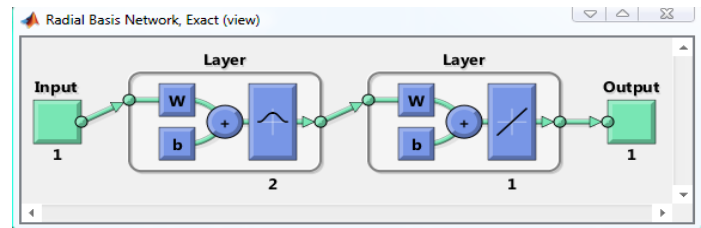
### III-4-2. l'architecture du Cascade Neural Network (CNN)

```
tab=xlsread('E:\proj2\Anne.xlsx','Feuil2','C2:D32');
[x,t] =size(tab);
CNN = cascadeforwardnet(10,'trainlm')
CNN = train(CNN,x,t);
view(CNN)
```



### III-4-3. l'architecture du Radial Basis Functions Network (RBFN)

```
tab=xlsread('E:\proj2\Anne.xlsx','Feuil2','C2:D32');
[x,t] =size(tab);
RBF = newrbe(x,t)
RBF = train(RBF,x,t);view(RBF)
```



## III-4- Adaptation du modèle aux données d'entraînement

Configuration de la division des données pour la formation, la validation et les tests, (`divideParam`). Cette propriété définit les paramètres et les valeurs du courant fonction de division de données.

```
FFN.divideParam.trainRatio = 60/100;
FFN.divideParam.testRatio = 20/100;
FFN.divideParam.valRatio = 20/100;
FFN.divideParam.lr = 0.01;
```

`trainRatio` : est le nombre relatif d'échantillons d'apprentissage à sélectionné parmi tous les échantillons par une fonction de division des données, comparé au nombre relatif d'échantillons de validation `valRatio` et d'échantillons d'essai `testRatio`

```
FFN.trainFcn='trainlm';
FFN.trainParam.min_grad =10^-10;
FFN.trainParam.showWindow =true;
FFN.trainParam.epochs =1000;
FFN.trainParam.time =120;
FFN.trainParam.goal=0;
FFN = train(FFN,1:k,R_S);
```

`trainFcn`= Cette propriété définit la fonction d'apprentissage pour mettre à jour le réseau, quand la fonction `train` est appelée.

'`trainlm`': est une fonction de formation en réseau qui met à jour les valeurs de poids et de biais en fonction de l'optimisation , `trainlm` est souvent l'algorithme de rétropropagation le plus rapide de la boîte à outils et il est vivement recommandé en tant qu'algorithme supervisé de premier choix, même s'il nécessite plus de mémoire que d'autres algorithmes.

`trainParam` : Cette propriété définit les paramètres de la fonction d'apprentissage actuelle.

`min_grad` : est le gradient de performance minimum avant d'arrêter l'entraînement. Lorsque le gradient de performance devient trop petit, il est peu probable que la formation continue produise des améliorations significatives.

`showWindow`: indique si la fenêtre d'apprentissage '`ntraintool`' devrait être affichée.

Epochs : est le nombre maximum d'itérations de formation avant l'entraînement est arrêté.

Time : est le temps maximum en secondes avant l'arrêt de l'entraînement.

Goal : est l'objectif de performance, en termes de fonction de performance.

`train(FFN,1:k,R_S)` : entraîne un réseau selon `net.trainFcn` et `net.trainParam`.

$\mu$  : est un facteur de mélange. Plus il est important, plus on accorde de poids à apprentissage de la descente de gradient et une petite étape. Le plus petit c'est la méthode de Newton donne plus de poids aux grandes marches.

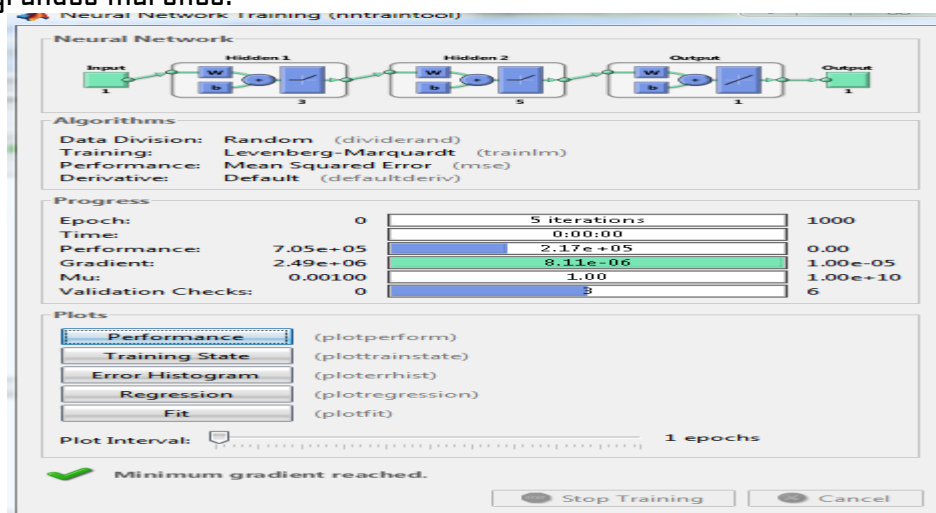


Figure 35 : Feed-forward neural network

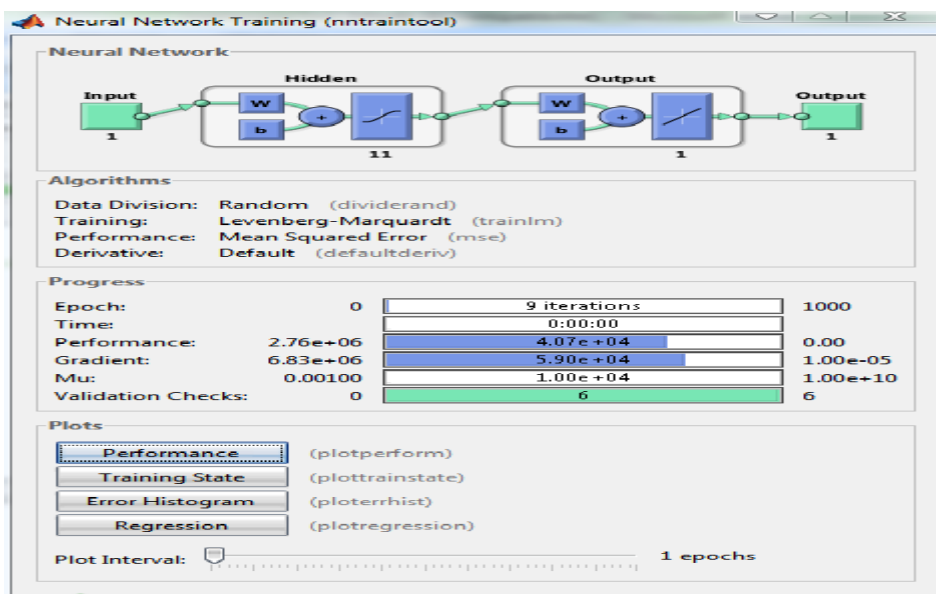


Figure 36 : Function fitting neural network(FitNet)

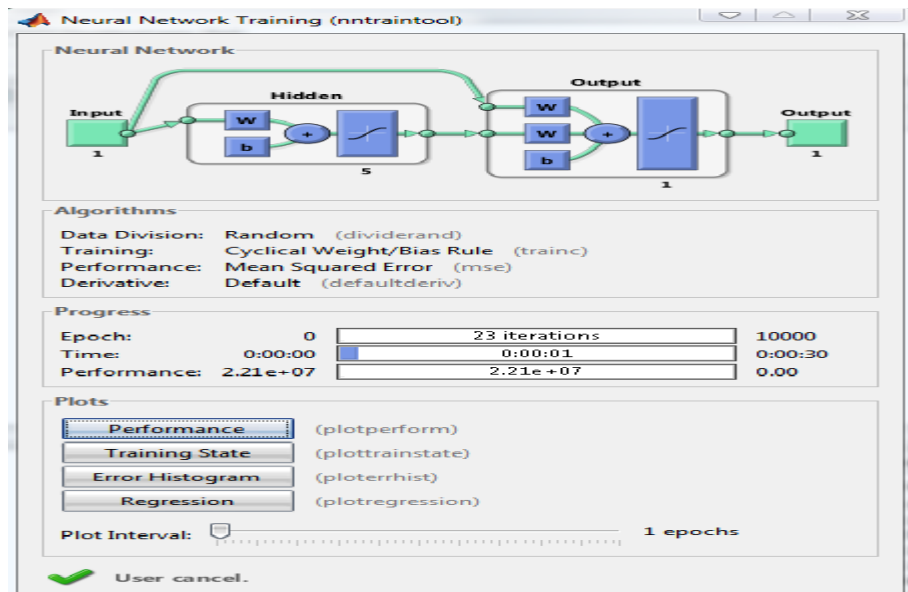


Figure 37 : Cascade-forward neural network

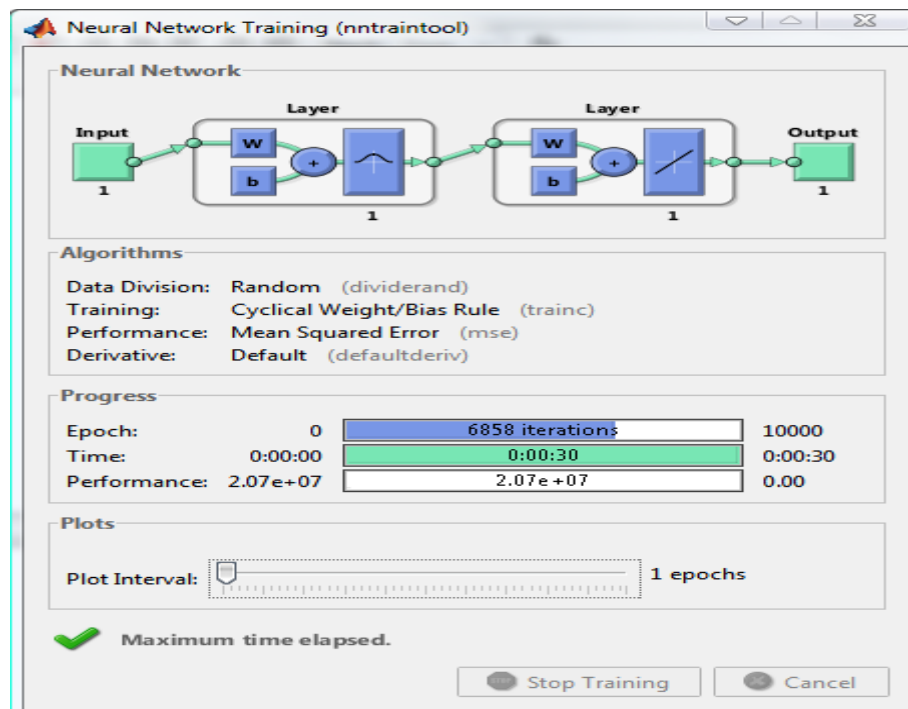


Figure 38 : Radial basis network



### III-5- Evaluation du modèle sur les données de test

Pour calculer la précision du réseau de neurones, trois fonctions sont donc prises en charge dans MATLAB, à savoir mse, mae, sse.

mse (e) : est une fonction de performance du réseau. Il mesure la performance du réseau en fonction de la moyenne des erreurs au carré.

mae (e) : est une fonction de performance du réseau. Il mesure les performances du réseau en tant que moyenne des erreurs absolues.

sse (e) : est une fonction de performance du réseau. Il mesure les performances en fonction de la somme des erreurs au carré.

```

y = FFN(x);
perf = perform(FFN,y,t) : Calculer les performances du réseau
e = t-y
MAE1 = mae(e) : MAE2 =mean(abs(e))
RMSE1=sqrt(mse(e))
RMSE2=sqrt(mean(e.^2))
MSE1 = mse(e) : MSE2 =mean(e.^2)
MBE1=mean(t-y);
MAPE1=100*MAE1/mean(t);
R1=corr(y',t') ;
SSE = sse(e)
    
```

Evaluation_FitNet	Evaluation_FFN	Evaluation_newff	Evaluation_CNN	Evaluation_RBFN	Evaluation_ENN
2.5086e+007	2.0033e+007	1.9388e+007	2.7791e+007	2.0420e+007	1.4226e+007
e =	e =	e =	e =	e =	e =
-5.0086e+003	-4.4759e+003	-4.4032e+003	-5.2717e+003	-4.5188e+003	-3.7718e+003
MAE =	MAE =	MAE =	MAE =	MAE =	MAE =
5.0086e+003	4.4759e+003	4.4032e+003	5.2717e+003	4.5188e+003	3.7718e+003
MSE =	MSE =	MSE =	MSE =	MSE =	MSE =
2.5086e+007	2.0033e+007	1.9388e+007	2.7791e+007	2.0420e+007	1.4226e+007
SSE =	SSE =	SSE =	SSE =	SSE =	SSE =
2.5086e+007	2.0033e+007	1.9388e+007	2.7791e+007	2.0420e+007	1.4226e+007
f_x >>	f_x >>	f_x >>	f_x >>	f_x >>	f_x >>

III-6- prédire des valeurs

```
>> net(365)
ans =
    5.1998e+003
>> net(31)
ans =
    5.1997e+003
>> net(21)
ans =
    4.3402e+003
>> net(11)
ans =
    4.5083e+003
>> net(1)
ans =
    3.8814e+003
>> 4267.7350
ans =
    4.2677e+003
```

	1	2
1	1	3.4345e+03
2	2	4.2205e+03
3	3	4.2207e+03
4	4	4.0662e+03
5	5	4.1164e+03
6	6	4.4298e+03
7	7	4.4786e+03
8	8	4.5063e+03
9	9	4.5419e+03
10	10	2.9706e+03
11	11	4.6266e+03
12	12	4.6404e+03
13	13	4.6641e+03
14	14	4.4306e+03
15	15	4.7055e+03
16	16	4.7596e+03
17	17	4.6039e+03
18	18	4.6537e+03
19	19	4.5692e+03
20	20	4.5646e+03
21	21	4.7496e+03
22	22	3.4775e+03
23	23	4.5729e+03
24	24	5.0469e+03
25	25	4.8992e+03
26	26	4.7948e+03
27	27	4.9987e+03
28	28	4.9564e+03
29	29	5.1441e+03
30	30	5.0937e+03
31	31	5.2075e+03

## L'Interfaces système

L'exploitation du système doit être simple intuitive et facile.

C'est dans ce but qu'en tant que concepteur, nous illustrons délibérément l'interface graphique du système et son utilisation.

Le schéma utilisé est une interface graphique pour montrer comment les outils graphiques interagissent dans l'environnement de développement visuel pour faciliter à l'utilisateur l'exploitation du programme.

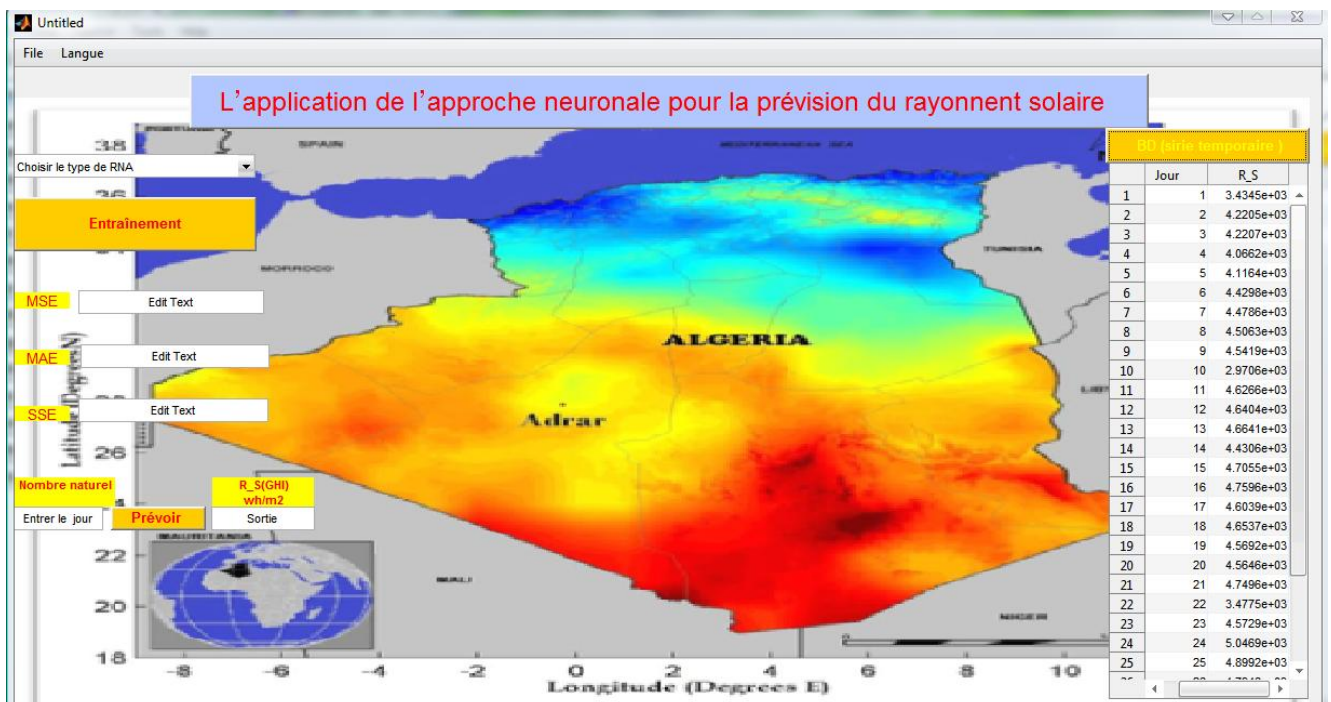


Figure 39 : Interface utilisateur

Initialement nous commençons par la liste des modèles de réseau de neurones (FFNN, FitNet, CNN, RBF), pour en choisir un pour l'entraînement, ensuite montrer la précision de l'entraînement moyennant l'évaluation MSE, MAE, SSE, puis nous pouvons effectuer le processus de prévision en introduisant le jour (la valeur à introduire doit être un nombre naturel et non pas une date) Nous appuyons sur le bouton "prévoir" pour produire le résultat dans la rangée suivante, qui correspond à la quantité de rayonnement solaire au cours de cette journée.

### III-7 Conclusion

Nous avons utilisé pour avoir la prévision solaire, une approche de réseau neuronal FFNN, FitNet, Newff, BRFN avec un flux de transmission direct, et une approche CNN,ENN avec un flux de transmission non direct

Le résultat conclu est que ENN est l'approche la plus précise, elle est suivie par la Newff, ce qui est acceptable, en prenant en considération l'environnement de travail qui ne répond pas au mode de fonctionnement des réseaux CNN, BRFN système parallèle.

## Conclusion générale

La prévision en intelligence artificielle est l'un des applications les plus importantes dans le domaine de Machine Learning, qui vient après clustering et de la classification en data mining, elle est appliquée dans des diverses disciplines : les réseaux sociaux pour prévoir les amis et les souhaits potentiels des clients, dans les marchés financiers, dans la navigation et la météorologie, ... etc. , parmi ses applications dans les prévisions, on prévoit la quantité de rayonnement solaire nécessaire pour obtenir de l'électricité.

Le réseau de neurones est l'un des algorithmes les plus fiables dans la prévision. Ils sont de différents types, ils sont sélectionnés en fonction des problèmes et en fonction de leur complexité.

Les plus importants types appliqués dans la prévision c'est ENN et Newff qui peuvent être sélectionnés et modifiés dans leurs couches et leurs fonctions selon la technique utilisée pour la prévision tel que la régression linéaire, la régression logistique et régression pas à pas.

### Recommandations:

Dans ce modeste travail, nous avons présenté la voie menant à la prévision par intelligence artificielle, tout en s'intéressant à une seule technique parmi des nombreuses techniques de Machine Learning : "la technique de régression linéaire", notre étude n'a pas prévu les autres types de réseaux de neurones tels que SOM , Nous vous recommandons donc de le faire dans des recherches ultérieures.

### Références Bibliographiques

#### Références françaises

- [1. LANGUE FRANCAISE. *LAROUSSE*. [En ligne] 15 06 2019. [Citation : 15 06 2019.] <https://www.larousse.fr/dictionnaires/francais/pr%C3%A9vision/63880#synonyme>.
7. **Besse, Philippe**. Régression linéaire. *wikipedia.org/Accueil\_principal*. [En ligne] cours de l'École des Mines de Nancy, 12 03 2019. [Citation : 15 06 2019.] [https://fr.wikipedia.org/wiki/Rgression\\_linaire](https://fr.wikipedia.org/wiki/Rgression_linaire).
8. **Ladjedel, Billal et Habiboullah, Beha Dine**. *Utilisation des réseaux de neurones artificiels pour la prédiction de la vitesse de vent*. M'sila, DEPARTEMENT D'ELECTRONIQUE : UNIVERSITE MOHAMED BOUDIAF - M'SILA, 2018. Master Académique.
10. **BESSAI, Fatma-Zohra, MOHAMED, Guergab, et MOHAMED, Guergab**. Utilisation des cartes auto-organisatrices de Kohonen dans la recherche documentaire. *Revue d'Information Scientifique et Technique (RIST)*. 2003. Vol. vol. 13, p. 01-14.
11. **HAMMOUCHE, Y**. *Comparaison de plusieurs méthodes pour la prédiction de la Charge Electrique Nationale*. Annaba, Université Badji Mokhtar : s.n., 2009. Mémoire de magister.
12. **KHEDAIRIA, Soufiane**. *Identification des Types de Jours Météorologiques et des Dépendances Météo-Paramètres de Pollution par RNA: Application à la Région d'Annaba*. Annaba, Département Informatique : UNIVERSITE BADJI MOKHTAR -ANNABA-, 2008. diplôme de magister.
13. **TAOUCHE, Rabah**. *Prévision du comportement mécanique d'alliages biphasés par algorithmes génétiques et réseaux de neurones. Application aux systèmes WC-cO*. 2010. CONSTANTINE, DEPARTEMENT DE PHYSIQUE : UNIVERSITE MENTOURI CONSTANTINE , 19 01 2010. DIPLOME DE DOCTORAT.
14. **OMARI, Abdallah**. *Approche hybride Réseau de neurones – Dempster Shafer*,. ORAN, Département d'Informatique : UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE D'ORAN Mohamed Boudiaf, 2013. diplôme de Magister.
16. **DREYFUS, Gérard**. *Les réseaux de neurones. Mécanique industrielle et matériaux, , vol. 51*. la Ville de Paris, École Supérieure de Physique et de Chimie Industrielles (ESPCI) : s.n., 1998.
18. **LEMAIRE, Vincent, BERNIER, Olivier, COLLOBERT, Daniel, et al**. Une nouvelle fonction de coût régularisante dans les réseaux de neurones artificiels: Applicationa la classification. In : Conférence d'Apprentissage, CAP99. <http://www. robo. jussieu. fr/lemai>.
20. **TOUZET, Claude**. . les réseaux de neurones artificiels, introduction au connexionisme. EC2, 1992.

### Références anglaises

3. **Jonathan Skillings** . archive.today. *CNET News Robotic*. [Online] 07 03, 2006 . [Cited: 06 15, 2019.] [http://news.com.com/Getting+machines+to+think+like+us/2008-11394\\_3-6090207.html](http://news.com.com/Getting+machines+to+think+like+us/2008-11394_3-6090207.html).
4. digital-analytics. *jixta.wordpress.com*. [Online] 07 17, 2015. [Cited: 06 15, 2019.] <https://jixta.wordpress.com/tag/machine-learning/>.
5. **Brownlee, jasonb**. How to Develop Convolutional Neural Network Models for Time Series Forecasting. *Machine Learning Mastery*. [Online] 11 12, November 12, 2018. [Cited: 06 15, 2019.] <https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/>.

### Références en arabe

2. الذكاء الاصطناعي. ويكيبيديا الموسوعة الحرة. [متصل] 10, 2018. [تاريخ الاقتباس: 06 15, 2019]. [https://ar.wikipedia.org/wiki/الذكاء\\_الاصطناعي](https://ar.wikipedia.org/wiki/الذكاء_الاصطناعي).
6. حسام الحوراني. قناة الذكاء الاصطناعي. 2018/02/10.
9. مقارنة أداء التنبؤ بين بعض الشبكات العصبية الاصطناعية ومنهجية (بوكس-جنكز) مع التطبيق.. مطر ظافر رمضان و صليوا رهاد عماد. vol. 16, no 28, p. 51-76, الوصل : المجلة العراقية للعلوم الاحصائية، 2019، المجلد 16.
15. نادر عروه. استخدام الشبكات العصبونية للتنبؤ في الأسواق المالية. دمشق، المعهد العالي للعلوم التطبيقية والتكنولوجيا، الجمهورية العربية السورية : اسم غير معروف، بعد 2011. رسالة ماجستير.
17. فعالية التنبؤ بمؤشر بورصة فلسطين باستخدام نموذج الشبكات العصبية الاصطناعية. درويش د. مروان جمعة. no 10، القدس فلسطين : مجلة جامعة القدس المفتوحة للبحوث الإدارية والاقتصادية، 2019، المجلد 3. vol.
19. سيد مجيد جمال صلاح الدين و إسراء زهير مجيد قبع. تخمين الجهد البرمجي باستخدام نموذج ال Cocomo التقييمي ا و شبكات العصبية. دمشق : كلية علوم الحاسوب والرياضيات، 2014.