N° d'ordre: 13/2010-M/MT

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'enseignement supérieur

Université des Sciences et de la Technologie Houari Boumediene

Faculté des Mathématiques
Département de Probabilités et Statistique

Mémoire Présenté Pour l'Obtention du Diplôme de

Magister en Mathématiques

Spécialité : *Probabilités et Statistique*

Par :

BENBOUTELDJA Mohamed Hamza

Thème

Prevision des

Series Temporelles par les Reseaux de Neurones et Architecture Optimale

Soutenu publiquement le:02/12/2010 devant le jury composé de :

Mme : Kh.DJABALLAH	Maître de conférences A	à l'USTHB	Présidente.
Mr : M.DJEDOUR	Professeur	à l'USTHB	Directeur de mémoire.
Mr : A.REBBOUH	Maître de conférences A	à l'USTHB	Examinateur.
Mme : H.SAGOU	Maître de conférences A	à l'USTHB	Examinatrice.

Remerciements

Ce projet de fin d'étude s'est déroulé au sein de l'Université de **U.S.T.H.B** (Université des Sciences et de la Technologie Houari Boumediene)

Louange à ALLAH, le miséricordieux, sans Lui rien de tout cela n'aurait pu être. Nous remerciement ALLAH qui nous a orienté au chemin du savoir et les portes de la science.

Je tiens tout particulièrement à exprimer ma plus profonde gratitude à Monsieur ***DJEDOUR Mohamed***, Professeur à l'Université des Sciences et de la Technologie Houari Boumediene de qui m'a accueilli au sein de son laboratoire, pour avoir dirigé ces travaux et m'avoir soutenu dans cette étude. Sa disponibilité et ses conseils ont été indispensables à la concrétisation de cette recherche.

Nous tenons à remercier vivement tous ceux qui nous ont aidés de prés ou de loin à l'élaboration de ce mémoire ; on pense particulièrement à :

Tous les professeurs qui m'ont aidé par leurs enseignements pendant toutes ces années d'étude. Grâce à eux j'ai été capable de finaliser ce projet de magister.

Enfin, j'adresse mes chaleureuses pensées à toute ma famille et à mes amis pour leurs soutiens et leurs encouragements tout au long de ces années de mémoire.

Table des Matières

Résumé— Ce mémoire concerne la prédiction de séries temporelles. Une série temporelle est un ensemble de données enregistrées séquentiellement. Les différentes méthodes de prédiction utilisées sont les modèles linéaires AR, MA et ARMA, et en ce qui concerne les modélisations non linéaires sont les réseaux de neurones.
Les simulations effectuées sur les séries temporelles avec ces modèles ont été satisfaisantes, sans plus. Ceci s'explique par la faible autocorrélation des échantillons utilisés. Une amélioration possible serait d'étudier les adaptations des modèles AR, MA, ARMA à des processus non stationnaires (modèle saisonnier, modèle GARCH, etc.). Dans le cadre de la prédiction non linéaire, différentes configurations de réseau ont été comparées. De façon surprenante, c'est le réseau de neurone le plus simple qui conduit à la meilleure prédiction. La qualité de la prédiction devient vraiment intéressante lorsqu'on ne considère les valeurs prédites que si elles sont comprises dans un intervalle donné. Une autre piste permettant d'améliorer sensiblement les performances de prédiction consiste à travailler sur les différences entre deux échantillons consécutifs plutôt que sur les échantillons eux-mêmes, ce qui conduit que le réseau à modéliser uniquement les relations non linéaires.

Introduction

L'objectif de la prévision est de connaître aujourd'hui la valeur future d'une variable, avec la meilleure précision et fiabilité possible. Ce qui permet l'utilisation la plus efficace des ressources disponibles, ainsi que de planifier l'acquisition de nouvelles ressources.
L'approche classique de prévision commence par formuler un modèle, en analysant la chronique afin d'extraire ses propriétés statistiques, ses paramètres seront ensuite estimés à l'aide des données fournies Néanmoins, cette modélisation linéaire s'avère insuffisante pour maîtriser certaines dynamiques pour lesquelles la relation entre la valeur à un instant donné de la série et les valeurs passées est de nature non linéaire.

C'est dans ce cadre qui est apparue la méthode des réseaux de neurones dans la prévision des séries temporelle après avoir réaliser de bon résultats dans plusieurs domaines tels que la classification, le traitement de langage et la reconnaissance des formes, celle-ci permet en fait de capter les relations non linéaire qui échappent aux méthodes classiques.

Notre travail tourne au autour d'une idée directrice, qui est la prévision d’une série temporelle à l'aide des modèles de réseaux de neurones de type multicouches, dont le principe consiste à utiliser les données et un critère à minimiser pour réaliser un modèle de type << **Boite noire** >>.

Le premier chapitre nous allons développer le nécessaire des outils de modélisation des réseaux de neurones artificiels, tout en approfondissant sur les modèles neuronaux utilisés tout particulièrement dans la prévision.
Le deuxième chapitre exposera un rappel des outils de base de l’analyse des séries temporelles, il mettra l’accent sur les modèles (S) ARIMA : les modèles auto régressives à moyenne mobiles, non stationnaires et saisonniers.
Le troisième chapitre présente le modèle neuronal le plus connu qui est le Perceptron multicouches (PMC). Nous rappelons dans le premier paragraphe les propriétés théoriques associées au PMC, ce sont elles qui expliquent le mieux pourquoi l'utilisation d'un PMC permet souvent d'obtenir des performances intéressantes.
Le second paragraphe introduit les PMC, l'algorithme d'apprentissage et les propriétés d'approximation universelle. L'apprentissage d'un PMC se ramène toujours à la minimisation d'une fonction de coût, les plus classiques étant celles des moindres carrés simples ou généralisés.
Le quatrième chapitre (partie empirique) contiendra deux sous sections, dans lesquelles nous allons essayer de déceler une relation entre l’analyse traditionnelle des séries temporelles et les réseaux de neurones artificiels, sachant que ces derniers peuvent représenter n’importe quel modèle (AR, ARMA, NAR, NARMA). Dans la première sous section nous allons appliquer la modélisation classique de Box-Jenkins, et on va tirer le modèle associé à notre étude de cas la série étudiée est celle présentant le produit pétrolier importé au port d'Alger.
Par la suite, au sein de la deuxième sous section, nous allons aborder la partie empirique par une justification de notre choix du modèle neuronal, par la suite nous allons comparer ses résultats à ceux fournis par les modèles linéaires classiques, en mettant l’accent sur la capacité de chaque modèle étudié de résister aux données bruitées et aux observations aberrantes.

I.1 Introduction

Le cerveau humain a une puissance de traitement de l'information très complexe, il possède des caractéristiques intéressantes absentes dans l'architecture des machines parallèles malgré leurs grandes capacités de calcul. Parmi ces caractéristiques, il existe en particulier son parallélisme, sa grande capacité d'apprentissage, de généralisation, d'adaptation …etc.

D'où l'idée des scientifiques de créer un modèle de calcul (les réseaux de neurones artificiels), qui tend à imiter le fonctionnement du cerveau (les réseaux de neurones biologiques), pour accroître les connaissances sur le mécanisme cérébral via l'élaboration de systèmes artificiels capables de reproduire des calculs complexes similaires à ceux effectué par le cerveau humain.

Le principe de base est une modélisation du comportement biologique des cellules nerveuses, ainsi que de leurs interconnexions, qui composent la partie la plus intéressante du corps humain qui est le cerveau, mais cette modélisation n'est que très pauvre au regard de la réelle complexité de fonctionnement d'une cellule nerveuse.

I.2 Historique des Réseaux de Neurones

Tout a commencé en 1943, lorsque deux biophysiciens **McCulloche et Pitts** [A Logical Calculus of ideas immanent in Nervous Activity], crée le premier modèle de neurone biologique, nommé neurone formel, s'inspirant des récentes découvertes en neurobiologie.

Six ans plus tard, **Donald Hebb** [the organization of behaviour], introduit sa brillante idée sur la notion des poids synaptiques. Il propose en 1949 une formulation du mécanisme d'apprentissage, sous la forme d'une règle de modification des connexions synaptiques qui porter encore sont nom.

Finalement c'est en 1958 que **Frank Rosenblatt** [principales of neurodinamics], combinant les idées de ses prédécesseurs, conçoit le premier réseau de neurones artificiels inspiré du système visuel possédant une couche de neurones " perceptive" et une couche de neurones « décisionnelle » nommé le Perceptron.

Ce réseau qui parvient à apprendre, à identifier des formes simples et à calculer certaines fonctions logiques, capacité d'apprendre par l'expérience jusqu'a là réservée aux vivants.

Les travaux de **Rosenblatt** suscitent au début des années 60 un vif enthousiasme attirant les scientifiques à la recherche sur l'intelligence artificielle.

D'autres chercheurs arrivèrent à des réseaux similaires ; **Widrow & Hoff** [adaptive switching circuits] invente un réseau dont l'algorithme plus rapide et plus précis ajuste les poids par rapport à la taille de l'erreur.

Cet enthousiasme se voit pourtant brusquement refroidi en 1969, lorsque deux scientifiques américains, **Minsky** et **Papert** montrent dans un livre [Perceptrons] toutes les limites de ce modèle, et soulèvent particulièrement l'incapacité du Perceptron à résoudre les problèmes non linéairement séparables, tels que le célèbre problème du XOR (OU exclusif).

Ces conclusions prolongent alors la recherche sur les réseaux de neurones artificiels dans une défaveur qui ne prendra fin que 15 ans plus tard.

Ecartés pendant prés de quinze ans, les chercheurs ont développé de nouvelles règles d'apprentissages, de nouvelles architectures qui rendait les réseaux de neurones bien plus puissants que les simples perceptrons des années 60. en 1972, **Kohonen** présente ses travaux sur les mémoires associatives, il a développé l'apprentissage neuronal non supervisé dans les réseaux de neurones. L'ouvrage de **Hinton & Anderson** (1981) [parallel models of associative memory] a constitué le signe d'une reprise. Les résultats des travaux de **John Hopfield** [neural networks and physical systems which Emergent collective Computation Abilities] (1982) ont également définis l'utilité des réseaux complètement connectés (la deuxième grande classe de réseaux de neurones).

Toutefois, le retour incontesté des réseaux neuronaux s'est marqué par la publication de **McClelland & Rumelhart** [learning publication by back-propagation error] (1986) donnant une variété d'architectures de fonction de transfert et d'algorithmes d'apprentissage pour les réseaux multicouches (la règle delta généralisée) permettant aux réseaux neuronaux de revenir au devant de la scène.

Parallèlement aux travaux de **Hopfield, Werbos** conçoit un mécanisme d'apprentissage pour les réseaux multicouches de type perceptron : c'est l'algorithme d'apprentissage nomé par " Back-propagation" (rétro propagation de l'erreur) qui fournit un moyen simple d'entraîner les neurones des couches cachées. Cet algorithme sera réellement popularisé en 1986 par **Rumelhart** dans le livre (" parallel distribution processing ").

Cet algorithme a eu un impact considérable : contrairement à son célèbre ancêtre, il ne souffre d'aucune limitation théorique et a pu être employé avec succès grandissant pour résoudre les problèmes complexes rencontrés dans de nombreux domaines à la fois scientifiques et techniques.

En pratique, les réseaux de neurones ont connu de très nombreuses applications, notamment en physique des particules[1]où leur utilisation qui a été envisagée pour la première fois en 1987 par **B. Denby** [neural networks and cellular automata appealer Experimental High Enrgy Physics] (1988).

[1] petit élément de matière physique : grain, poussière, etc.

Les réseaux de neuronaux connaissent depuis une dizaine d'années une forte utilisation car il s'agit avant tout d'un outil mathématique et algorithmique dont les domaines d'application sont variés.

Dés lors, la modélisation par les réseaux de neurones artificiels se retrouve en cardiologie.

I.3. Les Modèles de Réseaux de Neurones

I.3.1 Le Neurone Biologique

Le neurone biologique est une cellule nerveuse composée d'un corps cellulaire dit **soma** et de deux types de prolongements : les dendrites et l'axone.

- **Le soma** : est un corps cellulaire possédant un noyau. Le soma collecte les signaux électriques provenant des dendrites et les traites et effectue les transformations biochimiques nécessaires à la vie du neurone.
- **Les dendrites** : prolongement arborisé du cytoplasme d'une cellule nerveuse, permet au neurone de recevoir des signaux provenant des axones des autres neurones.
- **L'axone** : c'est la fibre nerveuse qui sert de transport pour les signaux émis par le neurone aux autres neurones.

Pour plus de précision sur le modèle biologique le lecteur peut consulter [1]. Sur la figure suivante, on montre un neurone au niveau biologique :

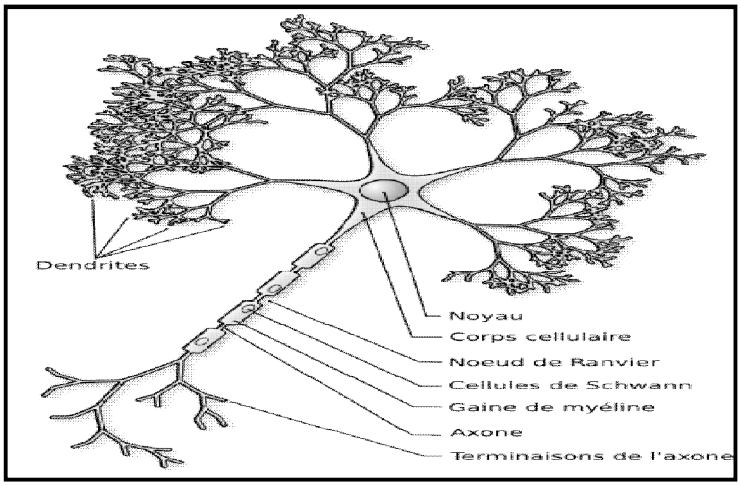


Fig I.1 Image : Neurone Biologique

Le neurone est le constituant de base du système nerveux dans le cerveau. On estime qu'il y a au moins cent milliard de neurones, présentant des formes très variables, qui ont la particularité d'être accolés les uns aux autres par points de jonction, les synapses[2]. L'ensemble de ces

[2] Synapse : région de contacte de deux neurones ie: la jonction entre deux neurones.

synapses forme un immense réseau dans lequel circulent simultanément des milliers d'informations qui sont des signaux électriques.

En règle générales, les signaux électriques arrivent à la cellule par les dendrites, ces signaux d'entrées sont traités par le soma puis transmis à un autre neurone par l'intermédiaire de l'axone. On observe une sorte de sommation des signaux au niveau d'un neurone, quand la somme dépasse un certain seuil. Le neurone émet à un signal électrique, mais si la somme ne dépasse pas le seuil il ne se passe rien.

I.3.2 Le Neurone Formel

Le modèle du neurone formel est un modèle mathématique très simple dérivé d'une analyse de la réalité biologique d'un neurone biologique. La première modélisation d'un neurone formel date de 1943 présentée par MC Culloch et Pitts, dont le but était de représenter l'activité électrique des cellules nerveuses du cerveau.

« Un *neurone formel est un opérateur algébrique qui effectue une somme pondérée de ses entrées, appelée potentiel. Sa sortie est une fonction de ce potentiel* **».**

Le neurone fait une somme pondérée des entrées, qui sont des potentiels d'actions provenant des autres neurones (chacun de ces potentiels d'actions est une valeur numérique qui représente l'état du neurone qui l'a émis).

Le neurone s'active suivant la valeur de cette somme pondérée : si la somme dépasse un certain seuil, le neurone est activé et transmet une réponse (sous forme de potentiel d'action) dont la valeur est celle de son activation, si le neurone n'est pas activé, il ne transmet rien.

La figure suivante montre une comparaison entre le neurone biologique et le neurone artificiel. Dans la cellule nerveuse humaine, la synapse correspond au poids d'un neurone artificiel, le corps cellulaire à une fonction de transfert et l'axone à un élément de sortie.

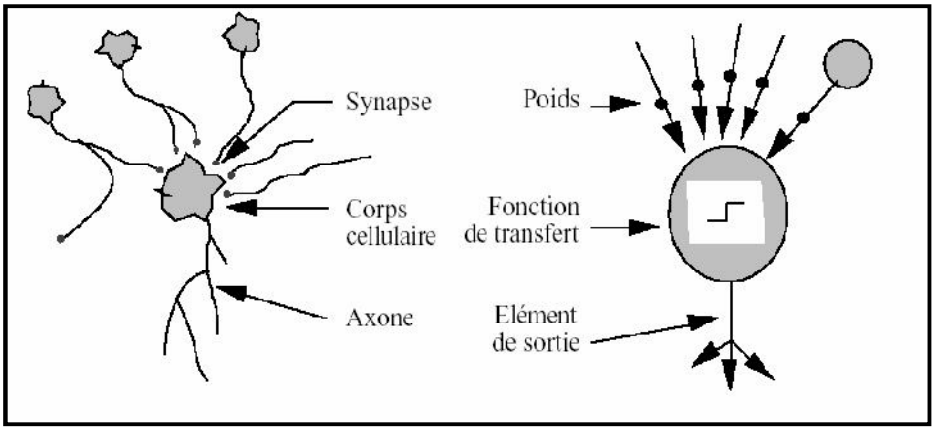


Fig I.2. La comparaison entre un neurone biologique et un neurone artificiel

D'une autre manière, le neurone formel est défini comme étant « un automate reproduisant la composée de plusieurs fonctions très simples chacun des P liaisons synaptiques entrant est affecté d'un poids $\theta_i, i \in \{1,2,...,P\}$, simulé par une entré réelle $x_i, i \in \{1,2,...,P\}$.Par convention, on ajoute aussi une entrée constante pondérée par un poids θ_0. L'opposé de θ_0 peut être vu comme une valeur seuil, au delà de laquelle le neurone est activé. »

I.3.2.1 Caractéristiques des Réseaux de neurones Artificiels

Selon ces définitions, le neurone effectue deux opérations de calcul :

- **Le potentiel** : la somme pondérée des entrées $\sum_{i=1}^{n} \theta_i x_i + \theta_o$
- **L'activation** : la transformation selon la fonction de transfert [3] (fonction d'activation)

$$f\left(\sum_{i=1}^{n} \theta_i x_i + \theta_o\right)$$

Où : x_i : représente la i[éme] entrée du neurone.

θ_i : représente le poids de la connexion entre la i[éme] entrée et le neurone [4].

θ_o : représente LE biais, il peut être vu comme une valeur seuil, au delà de laquelle le neurone est activé.

f : Fonction d'activation, qui peut être soit une fonction de seuillage, ou une fonction sigmoïde.

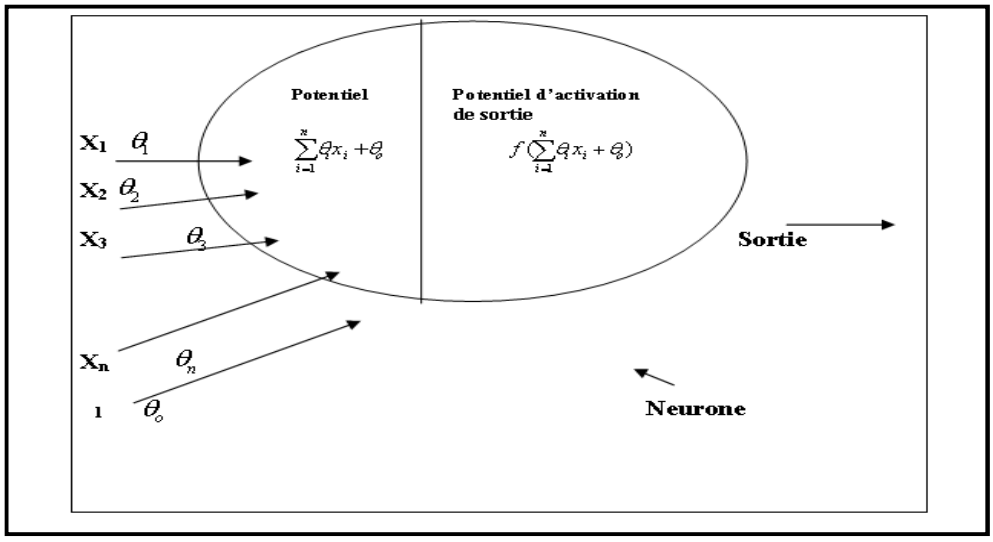


Fig I.3 le neurone formel [3]

[3] Fonction d'activation.
[4] Coefficient synaptique.

Le neurone formel, dés son apparition, suscita un vif intérêt parmi les pionniers du connexionnisme. La première appréciation notable était au début des années 60.

Fonctions de transferts :

- **Fonctions de seuillage**

1. **Fonction linéaire :** $f_1(x)= \lambda x$.
2. **Fonction binaire à seuil :** $f_2(x) = \begin{cases} 1 & \text{si } x>\theta . \\ 0 & \text{sinon .} \end{cases}$
3. **Fonction linéaire à seuil :** $f_3(x) = \begin{cases} 0 & \text{si } x< \theta_1 . \\ ax+b & \text{si } \theta_1< x< \theta_2 . \\ 1 & \text{si } \theta_2< x . \end{cases}$

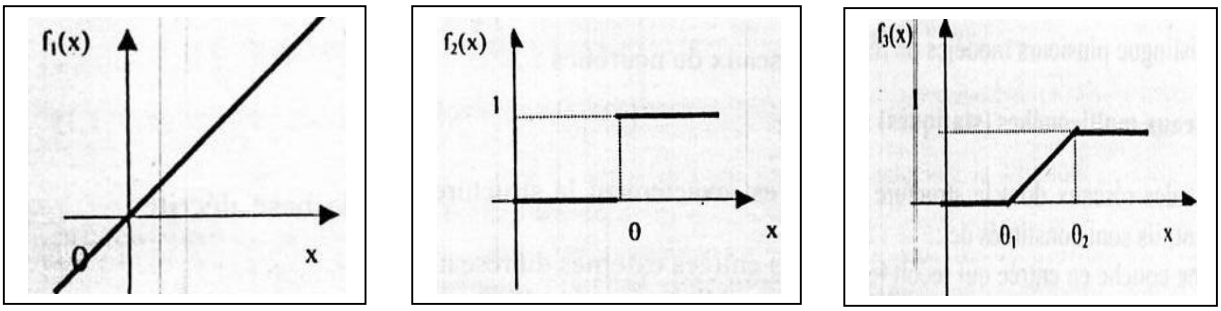


Fig I.4 : Les fonctions de seuillage f_1,f_2,f_3.

Fonctions sigmoïdes :

1. Fonction logistique : $f_4(x)=\left(1+e^{\frac{(x-\theta_1)}{\theta_2}}\right)^{-1}$ $\theta_1, \theta_2 \in R$ où : θ_1 : le seuil.

θ_2 : sa courbure.

2. Fonction tangente hyperbolique « tanh » : $f_5(x) = \dfrac{1-e^{-2x}}{1+e^{-2x}}$

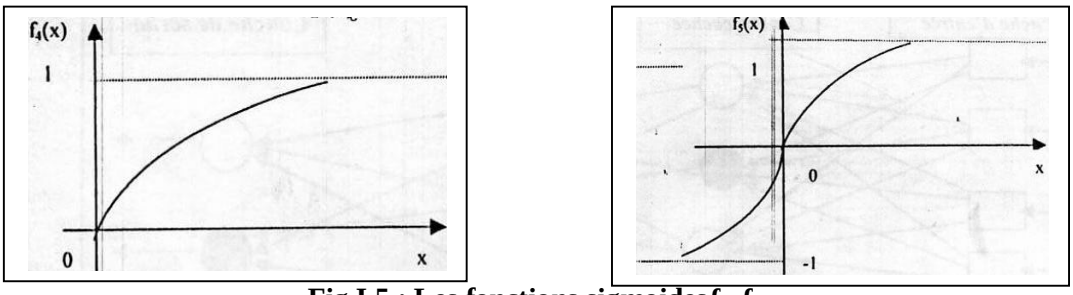


Fig I.5 : Les fonctions sigmoidesf_4,f_5.

La fonction f est en générale une fonction non linéaire, qui remplit les conditions des fonctions continues, bornées, dérivables. Telles que les fonctions citées précédemment[5] , ainsi « **Un neurone est une fonction non linéaire, paramétrée, à valeurs bornées ».**

Les éléments de ressemblance entre le neurone biologique et le neurone formel sont résumés dans le tableau ci-dessous :

Neurone biologique	Neurone formel
synapses	Poids des connexions
Axone	Signal de sortie
dendrite	Signal d'entrée
soma	Fonction d'activation

Tableau I.1 : éléments de ressemblances entre le neurone biologique et le neurone formel.

I.4 Les Différents Types des Réseaux de Neurones

I.4.1 Le Perceptron Simple

L'intérêt qui est porté tout particulièrement à l'étude du neurone formel atteint rapidement ses limites, du moins en ce qui concerne son utilisation comme unité unique de traitement, pour cela Rosenblatt a attribué en 1962 le nom de « **Perceptron** » à un ensemble de neurones formels connectés entre eux, dont le but était d'apprendre progressivement à ce modèle, de séparer un ensemble d'entrées en sous ensembles finis disjoints.

Ce Perceptron simple se caractérise par :

- **Sa structure** : les unités de la couche d'entée sont directement reliées aux unités de la couche de sortie.
- **La fonction d'activation** : fonction seuil pour les neurones de la couche de sortie.
- **Les poids de connexion :** sont de dimension n (on ajoute l'entrée constante).

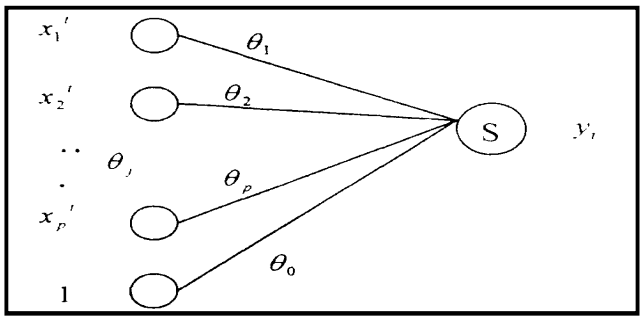


Fig I.6 : Le Perceptron Simple (S est la fonction signe)

[5]Sigmoïde, seuil, gaussienne.

Un neurone linéaire à seuil réalise une partition des vecteurs d'entrées qui lui sont soumis en deux classes. La frontière entre ces deux classes est définie par la condition: $\sum_{i=1}^{n} \theta_i x_i = \theta_o$,

Où θ_o représente seuil du neurone linéaire. En effet pour $\sum \theta_i x_i > \theta_o$ le neurone répond 1, (élément appartenant à la première classe), et il répond 0 pour $\sum \theta_i x_i \leq \theta_o$ (élément appartenant à la deuxième classe). Puisque un neurone linéaire à seuil réalise une partition des vecteurs d'entées en deux classes, donc le problème du partitionnement en plus de deux classes peut être résolu également, simplement en considérant plusieurs cellules de décision (p neurones linéaire à seuil), qui permettent d'effectuer une partition en 2^p classes.

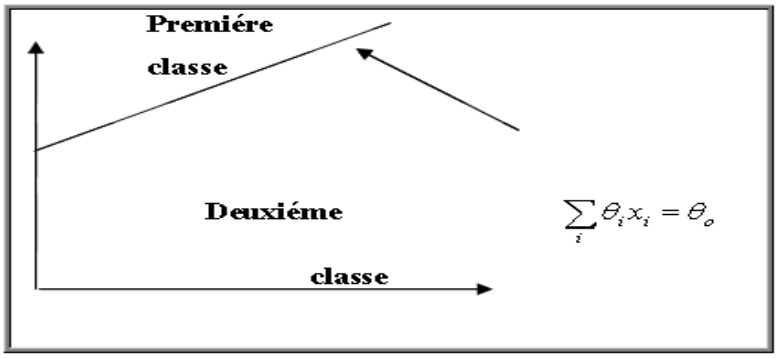


Fig I.7 : La séparation linéaire en deux classes.[6]

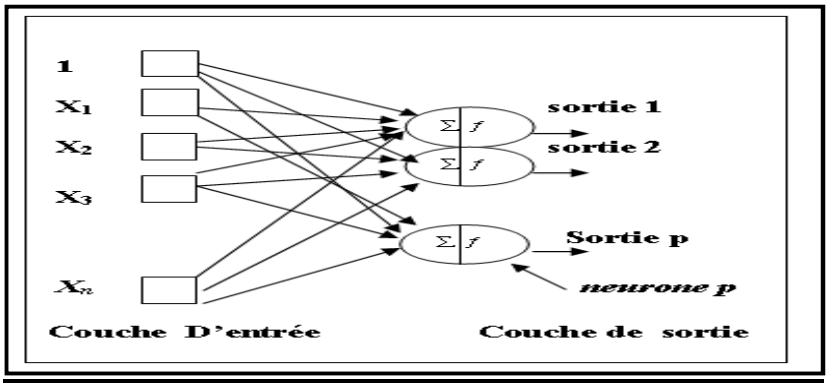


Fig I.8 : Perceptron Simple.

Toutefois, la publication d'un livre mathématique par Minsky et Papert a été cause du désespoir des chercheurs, celui-ci ayant démontré les limites théoriques du perceptron simple, qui s'avère incapable de séparer deux ensembles non linéairement séparables, la démonstration était illustrée par le célèbre exemple du « ou exclusif » (XOR), que le perceptron simple n'a pas

pu modéliser : $f:\{-1,1\}^2 \to \{-1,1\}$ avec $f(-1,-1)=-1$ $f(-1,1)=1$, $f(1,-1)=1$, $f(1,1)=-1$.

L'incapacité du Perceptron à opérer des séparations non linéaires a mené à un ralentissement des progrès de recherche dans le domaine.

La solution qui est plus tard apparue a permis de dépasser les limites théoriques du perceptron simple, c'était la structuration d'un réseau en composition de plusieurs perceptron simple, ce réseau porte le nom du réseau multicouches (MLP), pour lequel les chercheurs ont mis au point un apprentissage appelé l'algorithme de rétro propagation.

I.4.2 Le Perceptron Multicouches (MLP)

I.4.2.1 Description

Les réseaux de neurones artificiels ou les réseaux connexionnistes peuvent être définis comme un ensemble d'unités de calculs élémentaires (automates), structurés en couches successives, capables d'échanger des informations au moyen de connexions qui les relient, chaque unité effectue un traitement local de l'information. Il existe plusieurs types d'unités dans MLP.

- Les unités d'entrées ou cellules perceptives auxquelles sont transmises les données à traiter, en provenance des sources externes du réseau, ces unités n'effectuent aucun traitement sur les données en entrée.

- Les unités de sortie (neurones de sortie) qui contiennent l'information traitée et utilisable par d'autres systèmes connectés au réseau.

- Les unités cachées (neurones cachés), dont les entrées et les sorties sont reliées aux autres unités du réseau, sont non ''visibles'' pour les systèmes extérieurs. Ces unités cachées servent à traiter de façon interne au système les vecteurs d'entrées.

Un MLP muni d'une couche cachée de deux neurones est capable de résoudre le problème du (XOR).[2]

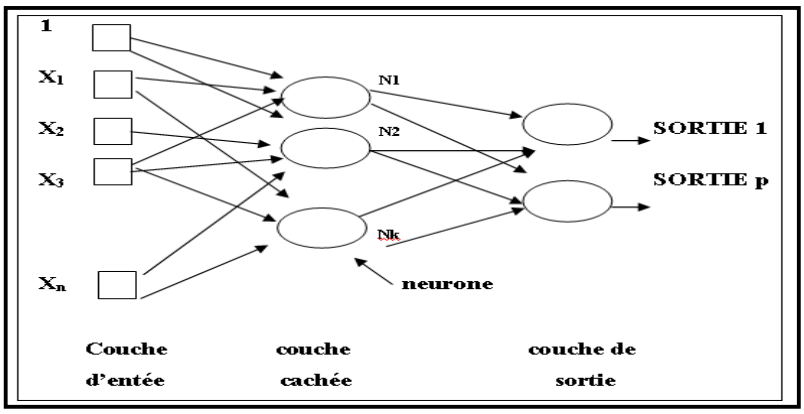


Fig I.8 : Perceptron multicouches.

L'intérêt des neurones réside dans les propriétés qui résultent de leurs associations en réseaux, c'est-à-dire de la composition des fonctions non linéaire réalisées par chacun des neurones.

Le perceptron multicouche comporte p unités en entrés recevant respectivement p variables $\{x_1,x_2,...,x_p\}$ et une seule unité de sortie qui produit la variable $y \in R^n$,si le réseau dispose de n neurones sur sa couche cachée, On note alors ce réseau PM(p,n,1). Un neurone seuil est aussi défini, il correspond a une entrée constante égale à 1.

$$Y_j = \sum_{i=1}^{n} \alpha_j \; f\left(\sum_{i=1}^{p} \beta_{ji} x_i + \beta_{j0}\right) + \alpha_0$$

Avec n : nombre de neurones de la couche cachée.

$\theta = \left\{ \left(\alpha_j\right)_{0\le j\le n} , \left(\beta_{ji}\right)_{1\le i\le p,\ 1\le j\le n} \right\} \in \mathrm{R}^{n\times(p+2)+1}$ est le vecteur des paramétres (poids)

Pour les neurones d'entrée, la fonction de transfert est l'identité. L'information passe sans modifications. Les neurones de la couche cachée ayant une fonction de transfert non linéaire, en général sigmoïde. En ce qui concerne la couche de sortie. La fonction, linéaire ou non, dérivable ou non, dépend du type de problème a résoudre, un MLP est un modèle de régression non linéaire paramétré par le vecteur θ.

I.4.3 Le Modèle Paramétrique NAR $_N$ (P) Basé sur Le Perceptron Multicouches

Un modèle autorégressif linéaire correspond à l'idée de la régression linéaire à chaque instant sur l'espace des observations passées, ce type de modèles pour lesquels la relation entre la variable qu'on va modéliser à un instant donné et la variable passée n'est pas linéaire.

Une extension non linéaire du modèle autorégressif est basée sur le perceptron multicouche. C'est une solution au problème de modélisation des relations de type non linéaire car le perceptron multicouche possède la propriété d'approximation universelle.
Un modèle autorégressif neuronal **NAR $_N$ (P)** est défini par l'équation :

$$x_t = \sum_{j=1}^{n} \alpha_j \delta(\sum_{i=1}^{p} \beta_{ij} x_{t-1} + \beta_{0j}) + \alpha_0 + \varepsilon_t \quad \varepsilon_t \text{ Bruit i.i.d.}$$

Le modèle autorégressif neuronal avec moyenne mobile **NARMA $_N$ (p, q)** est défmi par:

$$x_t = \sum_{j=1}^{n} \alpha_j \psi(\sum_{i=1}^{p} \beta_{ij} x_{t-1} + \sum_{k=1}^{q} \beta_{kj} \varepsilon_{t-k} + \beta_{0j}) + \alpha_0 + \varepsilon_t$$

I.5 L'Architecture du Réseau Neuronal

L'architecture d'un RNA est déterminée par le nombre de neurones qu'il contient et par la façon dont ils sont connectés, ces interconnexions sont caractérisées par des poids par analogie avec les neurones biologiques. Ie nombre de poids du réseau est donc égal au nombre total d'interconnexions entre les neurones. On distingue deux types de réseaux de neurones : **les réseaux non bouclés et les réseaux bouclés.**

I.5.1 Les réseaux non bouclés

« **Un réseau de neurones non bouclé réalise une (ou plusieurs) fonctions algébriques de ses entrées par composition des fonctions réalisées par chacun des neurones** »[7].

Dans ce type de réseaux l'information circule de l'entrée vers la sortie sans retour en arrière, car si on démarre d'un neurone donné suivant les connexions, on ne pourra jamais revenir au neurone de départ, ce qui fait que ces types de réseaux ont un graphe acyclique. Les réseaux de neurones non bouclés sont appelés : **les réseaux multicouches statiques,** puisque le temps ne

[7] G. Dreyfus. «Réseaux de neurones méthodologie et application ». p 4

joue aucun rôle fonctionnel dans ce type de modèle, car si les entrées sont constantes, les sorties le sont également, le temps de calcul de la fonction réalisée par chaque neurone est négligeable.

Dreyfus définit aussi, « **un réseau de neurones non bouclé à n entrées, Nc neurones cachés, et No neurones de sortie, réalise No fonctions non linéaires de ses n variables d'entrée par composition des Nc fonctions réalisées par ses neurones cachés, donc la sortie d'un réseau de neurones non bouclé est une fonction non linéaire de ses entrées et de ses paramètres** ».

Par contre un réseau de neurones non bouclé sans neurones cachés et un neurone de sortie linéaire réalise simplement une fonction linéaire de ses entrées. On peut considérer tout système linéaire comme un réseau de neurones, ce qui ne présente aucun intérêt, ni théorique ni pratique.

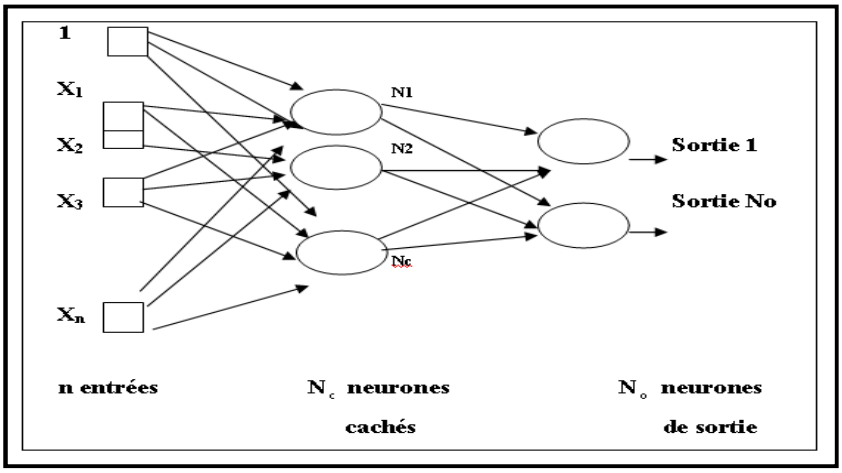


Fig I.11 : Un réseau de neurones à couche non bouclé.

I.5.2 Les réseaux de neurones bouclés (dynamiques)

« Un réseau de neurones bouclé à temps discret, réalise une (ou plusieurs) équations aux différences non linéaire, par composition des fonctions réalisées par chacun des neurones et des retards associés à chacune des connexions ».

L'architecture la plus générale pour un réseau de neurones est **les réseaux bouclés** dont le graphe des connexions est cyclique, car si on se déplace dans le réseau en suivant le sens des connexions, il est possible de trouver au moins un chemin qui revient à son point de départ. Un tel chemin est désigné sous le thème de cycle, donc la sortie d'un neurone du réseau peut être

fonctionne d'elle même, cela n'est évidemment concevable, que si la notion de temps est explicitement prise en considération.

Ainsi à chaque connexion d'un réseau de neurones bouclé est attaché outre le poids comme pour les réseaux non bouclés, un retard. C'est un multiple entier (éventuellement non nul) de l'unité de temps choisie, tel que tout cycle du graphe des connexions d'un réseau de neurones non bouclé doit comprendre au moins une connexion de retard non nul.

Toutefois, **« Tout réseau de neurones bouclé, aussi complexe soit il, peut être mis sous une forme canonique, comportant un réseau de neurones non bouclé dont certaines sorties sont ramenées aux entrées par des bouclages de retard unité »**.

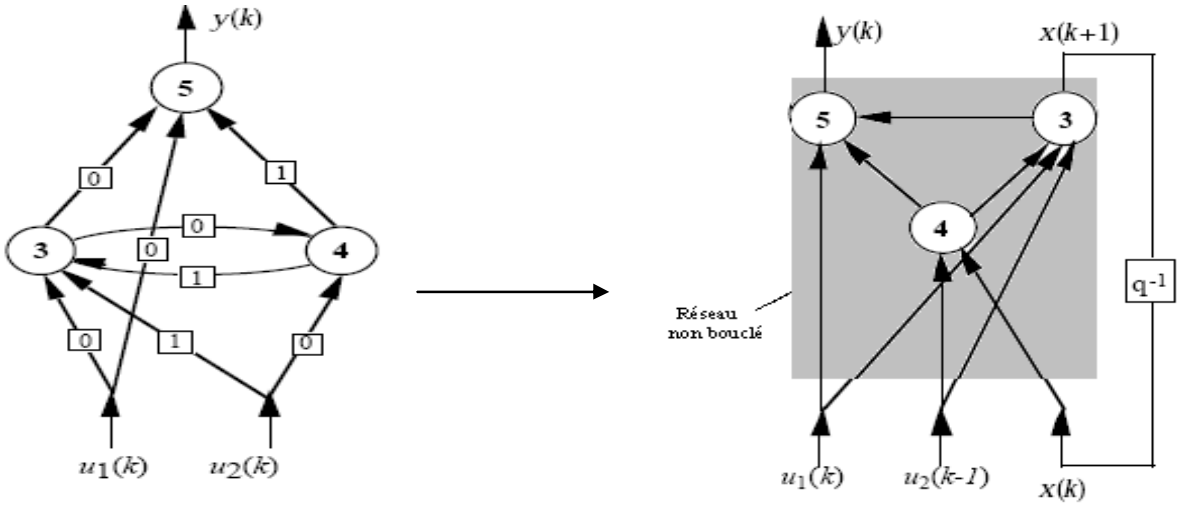


Fig I.12 : Un réseau de neurones bouclé.

Cette figure représente un exemple de réseau de neurones bouclé contenant un cycle qui part du neurone 3 et revient à celui-ci, en passant par le neurone 4, la connexion de 4 vers 3 ayant un retard d'une unité de temps.

Ce réseau bouclé peut être mis en forme canonique, comprenant un réseau de neurones non bouclé, dont les sorties sont ramenées à ses entrées par des bouclages de retard unité.

Dans les réseaux récurrents, les connexions sont affectées des poids (comme le 1er type), et d'un retard : multiple entier de l'unité de temps choisie, d'où la propriété : « tout cycle du graphe de connexions d'un réseau de neurones bouclé doit comprendre au moins une connexion de retard non nul.

Le choix de l'architecture joue un rôle important dans la performance du réseau, de plus influe plus sur l'apprentissage car une bonne architecture optimise la durée d'apprentissage.

Il n y a aucune théorie qui limite le choix du nombre de couches cachées ou même le nombre de neurones, un bon choix de l'architecture peut découler d'un savoir-faire et d'une expérience pratique.

Le moyen le plus sur consiste à essayer plusieurs combinaisons possibles et d'en choisir la meilleure.

Conclusion

Dans ce chapitre, on a présenté l'historique des réseaux de neurones qu'on pourrait diviser en trois parties, la première est la période des travaux basée sur les découvertes biologiques. La période de la défaveur après la démonstration mathématique des limites du perceptron, constitue la deuxième partie. Enfin la structuration du réseau multicouche a permis aux réseaux de neurone de revenir en avance.

Les trois notions de base des réseaux de neurones sont l'unité de base, l'architecture.

On a également défini le neurone formel qui constitue l'unité de base du calcul neuronal ainsi que son potentiel et sa fonction d'activation.

On a aussi essayé d'exposer les différents types de modèles neuronaux : le perceptron simple, perceptron multicouches jusqu'au modèle neuronal basé sur le modèle autorégressif.

II. Processus Aléatoires Stationnaires et Processus ARMA

II.1 Introduction

Un processus stochastique est une famille de variables aléatoires indexées par le temps dont l'objectif principal est la représentation des phénomènes aléatoires qui évoluent dans le temps.

Une trajectoire (ou une réalisation) prise par un processus aléatoire représentant certain phénomène (physique, économique, écologique, biologique,...), constitue une série chronologique dont l'analyse a pour but la description des principales propriétés du processus générateur de cette dernière.

Analyser une série chronologique revient à trouver un modèle mathématique adéquat décrivant le mécanisme ayant donné lieu à cette série temporelle.

Le modèle adéquat obtenu sera par la suite utilisé selon les objectifs désirés, tels que la prévision ou le contrôle.

On constate ainsi que le concept des processus stochastiques joue un rôle primordial dans la modélisation des séries chronologiques. Pour cela, allons présenter, dans un premier temps, les notions de base et les propriétés essentielles des processus aléatoires, en particulier celles de la famille des processus dits faiblement stationnaires ou encore de ceux qui peuvent être ramenés au cas stationnaire par le biais d'une transformation adéquate (ajustement d'une tendance déterministe, différence ordinaire, différence saisonnière, ...).

II.2 Définition d'un Processus Stochastique

Un processus stochastique est une suite de variables aléatoires réelles indexées par le temps $\{X_t,\ t \in N\}$.

Ici **t** appartient à un espace discret, ce qui définit un processus à temps discret. Un processus stochastique est donc une famille de variables aléatoires $\{X_t,\ t \in N\}$ c'est à dire de fonctions mesurables de l'espace S des échantillons à valeurs dans N. Pour chaque point **s** de l'espace des échantillons S, la fonction qui à t associe $X_t(s)$ est appelée : **la trajectoire du processus**. Les observations successives forment l'histoire (l'information) du processus.

II.3 Processus Stationnaires

La notion de stationnarité joue un rôle central dans la théorie des processus aléatoires, et particulièrement en analyse des séries chronologiques (Temporelles).

Dans plusieurs problèmes du monde réel, on rencontre des processus aléatoires qui évoluent dans un état d'équilibre statistique, dans le sens où les propriétés probabilistes et statistiques des processus ne changent pas dans le temps, de tels processus sont dits **stationnaires**.
On commence par donner la définition d'un processus stationnaire au sens strict, et ensuite celle de la stationnarité du second ordre.

- **Processus Strictement Stationnaire (Stationnarité Forte) :**

Grossièrement, un processus aléatoire est dit strictement stationnaire si sa loi de probabilité est invariante par translation dans le temps. Mathématiquement, le concept de stationnarité stricte est donné par la définition suivante :

Définition :

Un processus stochastique $\{X_t, t \in \mathbb{Z}\}$ est dit : strictement (ou fortement) stationnaire si pour tout $n \in \mathbb{N}^*$, et pour tout n-uples $(t_1, \ldots, t_n) \in \mathbb{Z}^n$, la distribution de probabilité conjointe du vecteur $(X_{t_1+h}, ..., X_{t_n+h})$ est la même que celle de $(X_{t_1}, ..., X_{t_n}), \forall h \in \mathbb{Z}$. Autrement dit, si on a :

$$P(X_{t_1} \le x_1, ..., X_{t_n} \le x_n) = P(X_{t_1+h} \le x_1, ..., X_{t_n+h} \le x_n), \forall (x_1, ..., x_n) \in R^n, \forall h \in \mathbb{Z}.$$

On note que toutes les caractéristiques (c'est à dire tous les moments) d'un processus strictement stationnaire si elles existent sont invariantes dans le temps. Cette définition de la stationnarité est cependant trop forte et très exigeante et repose sur la connaissance de la loi conjointe du processus qui ne peut être connue en pratique, sauf dans des cas très spéciaux. Toutefois, plusieurs propriétés essentielles des processus aléatoires peuvent être obtenus à partir des moments du premier et du second ordre.
La stationnarité de ces deux moments peut donc être suffisante pour expliquer la stationnarité du processus. Pour cette raison, on a besoin d'un concept de stationnarité moins fort et qui peut être rencontré dans la pratique.

- **Processus Faiblement Stationnaire (Second Ordre) :**

Considérons un processus stochastique de second d'ordre $\{X_t, t \in \mathbb{Z}\}$.

Définition Un processus est stationnaire du second ordre si :

$E(X_t) = E(X_{t+h}) = \mu$ (Moyenne constante).

$\forall t \in Z,\ E(X_t^2) < \infty$

$C\text{ov}(X_t, X_{t+h}) = E\left[(X_t - \mu_t)(X_{t+h} - \mu_{t+h})\right] = \gamma(h)$

La fonction $\gamma(h)$ est dite fonction d'autocovariance du processus.

Remarques

— 1. La fonction d'autocovariance d'un processus faiblement stationnaire dépend seulement de la différence des instants.

— 2. Dans la classe des processus du second ordre, il est clair que la stationnarité stricte implique la stationnarité faible (la réciproque n'est pas vraie, sauf pour les processus dits Gaussiens).

Un processus $\{X_t, t \in \mathbb{Z}\}$ est dit Gaussien si toute sous-famille finie du processus constitue un vecteur Gaussien. Autrement dit, pour tout $n \in N^*, (t_1,...,t_n) \in \mathbb{Z}^n$, le vecteur $(X_{t_1},..., X_{t_n})$ est Gaussien.

Processus Bruit Blanc (White Noise) Le plus simple processus stationnaire en analyse des séries temporelles est appelé : processus bruit blanc [8] $\{\boldsymbol{\varepsilon_t}, t \in \mathbb{Z}\}$ qui est une séquence de variables aléatoires non corrélées de moyenne nulle et de variance constante σ_ε^2 .

Le fait que les variables aléatoires $(\boldsymbol{\varepsilon_t})_t$ soient mutuellement non corrélées (hypothèse d'orthogonalité), nous permet de donner la fonction d'autocovariance de ce processus par :

$$\gamma_\varepsilon(h) = \text{cov}(\varepsilon_t, \varepsilon_{t+h}) = E(\varepsilon_t \varepsilon_{t+h}) = \begin{cases} \sigma_\varepsilon^2 & si \ \ h = 0 \\ 0 & \text{sinon.} \end{cases}$$

Par conséquent, la fonction d'autocorrélation est donnée par :

$$\rho_\varepsilon(h) = \begin{cases} 1 & si \ h = 0 \\ 0 & si \ h \neq 0 \end{cases}$$

II.3.1 Fonction d'Autocovariance

La suite de toutes les autocovariances d'une série stationnaire contient toutes les informations sur la mémoire de cette série. On l'estime au moyen de :

$$\hat{\gamma}(h) = \frac{1}{T}\sum_{t=1}^{T-h}\left(X_t - \bar{X}\right)\left(X_{t-h} - \bar{X}\right)$$

Avec

$$\overline{X} = \frac{1}{T}\sum_{t=1}^{T} X_t$$

On utilise T observations pour calculer la moyenne et la variance, alors que pour calculer $\gamma(h)$ on utilise seulement $T-h$ observations. Donc quand $h \to T$, l'estimateur de $\gamma(h)$ tend vers zéro

[8] Ce terme de la physique, faisant référence au spectre de la lumière blanche.

si le processus est stationnaire en covariance. Si cette condition de stationnarité est vérifiée, alors l'estimateur $\hat{\gamma}(h)$ est un estimateur consistent de $\gamma(h)$.

Propriétés

a) La fonction d'autocovariance $\gamma(h)$ satisfait la propriété suivante :

$\gamma(-h)=\gamma(h) \quad \forall h \in \mathbb{Z}$; (fonction paire).

Donc on peut, dans la pratique, se restreindre aux autocovariances aux retards positifs, c'est-à-dire que l'on peut, sans perte de généralité, prendre $h \in \mathbb{N}$.

b) On peut facilement, en utilisant l'inégalité de Cauchy Schwartz, vérifier la propriété suivante :

$|\gamma(h)| \le \gamma(0) = Var(X_t) \qquad \forall t, h \in \mathbb{Z}$.

II.3.2 Fonction d'Autocorrélation

La fonction d'autocorrélation de retard h : $\rho(h); \forall h \in \mathbb{Z}$, d'un processus du second ordre, faiblement stationnaire de moyenne $\mu = E(X_t)$ et de variance $Var(X_t) = \gamma(0)$; notée $\rho(h)$ est définie par :

$$\rho(h) = \frac{Cov(X_t, X_{t-h})}{\sigma_{X_t}\sigma_{X_{t-h}}} = \frac{\gamma(h)}{\gamma(0)} \quad \forall h \in \mathbb{Z}$$

Il est facile de vérifier que la fonction d'autocorrélation satisfait les deux propriétés suivantes, qui découlent directement des deux propriétés **a)** et **b)** de la fonction d'autocovariance.

Propriétés

1) $\rho(-h) = \rho(h)$; $\forall h \in \mathbb{Z}$.

Donc on peut dans la pratique se restreindre aux autocorrélations pour h =0: $\rho(0) = 1 \quad \forall h \in \mathbb{Z}$.

2) $|\rho(h)| \le 1; \forall h \in \mathbb{Z}$

- **Autocorrélation empirique**

L'estimateur de la fonction d'autocorrélation, $\hat{\rho}(h)$ est obtenu en remplaçant, dans l'expression de $\rho(h)$, $\gamma(0)$ et $\gamma(h)$ par leurs estimateurs $\hat{\gamma}(0)$ et $\hat{\gamma}(h)$, respectivement. En effet, on a :

$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)} \quad \forall h \in \mathbb{Z}.$$

Ce qui peut s'écrire, en tenant compte de la définition de l'estimateur empirique de la fonction d'autocovariance, sous la forme explicite suivante :

$$\hat{\rho}(h)=\frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}=\frac{T}{T-h}\frac{\sum_{t=1}^{t-h}\left(X_t-\overline{X}\right)\left(X_{t-h}-\overline{X}\right)}{\sum_{t=1}^{T}\left(X_t-\overline{X}\right)^2},\ \forall h\in\mathbb{Z}$$

II.3.3 Fonction d'Autocorrélation Partielle

Elle mesure la corrélation entre X_t et X_{t-h} , l'influence des variables X_{t-h+i} , ayant été étiré.

Soit la matrice des corrélations symétriques formées des (h-1) premières autocorrélations.

$$P_h=\begin{bmatrix}1 & \rho_1 & \cdots & \cdots & \rho_{h-1}\\ \rho_1 & 1 & \cdots & \cdots & \rho_{h-2}\\ . & & 1 & & \\ . & & & & \\ \rho_{h-1} & \rho_{h-2} & \cdots & \cdots & 1\end{bmatrix} \qquad h\in\mathbb{N}.$$

La fonction d'autocorrélation partielle est donnée par : $\rho_{hh}=\frac{\left|P_h^*\right|}{\left|P_h\right|}$

La fonction $\left|P_h^*\right|$ est le déterminant de la matrice P_h^* obtenue à partir de P_h , en remplaçant la dernière colonne de celle-ci par le vecteur $(\rho_1,...,\rho_h)$ ainsi :

$$P_h^*=\begin{bmatrix}1 & \rho_1 & \cdots\rho_1\\ \rho_1 & 1 & \cdots\rho_2\\ . & & \\ . & & \\ \rho_{h-1} & & \cdots\rho_h\end{bmatrix}$$

On peut se passer de ce calcul matriciel qui n'est souvent pas facile à faire ; pour cela on a recours à une écriture récurrente de ρ_{ii} tel que :

$$\rho_{ii}=\begin{cases}\rho_1 & si\quad i=1\\ \dfrac{\rho_i-\sum_{j=1}^{i-1}\rho_{i-1,j}\ \rho_{i-j}}{1-\sum_{j=1}^{i-1}\rho_{i-1,j}\ \rho_j} & i=2,.....,\ h\end{cases}$$

Avec

$\rho_{ij}=\rho_{i-1,j}-\rho_{ii}\ \rho_{i-1,i-j}$, $j=1,...,i-1$ et $i=2,...,h$.

Cet algorithme résolvant les équations de Yule-Walker de manière récursive est appelé algorithme de Durbin (1960).

Remarques

1. La représentation graphique de $\rho(h)$ est appelée : corrélogramme.

2. Si $\rho(h)$ décroît rapidement quand le nombre de retard augmente, cela signifie que la série est stationnaire, sinon elle est sans doute non stationnaire ou de mémoire longue.

II.4 Opérateurs

- **Opérateurs Retard (*Backward*)**

L'opérateur retard est un opérateur linéaire noté *B*, tel que : $BX_t = X_{t-1}$.

- **Opérateurs Avance (*Forward*)**

Par analogie, l'opérateur d'avance, noté *F* est tel que : $F\ X_t = X_{t+1}$.

Propriétés

1- Ces opérateurs sont inversibles tels que : $F^{-1} = B$ *et* $B^{-1} = F$.

2- $B^n X_t = X_{t-n}$ *et* $F^n X_t = X_{t+n}$

3- $(\sum_{i=1}^{n} a_i B^i) X_t = \sum_{i=1}^{n} a_i X_{t-i}$ Cette égalité décrit l'action sur le processus $\{ X_t, t \in \mathbb{Z} \}$ d'un polynôme en *B*, on peut évidemment déduire celui en F.

4- Ces opérateurs ont des propriétés qui permettent de les manipuler comme des séries entières habituelles, en particulier, on peut les sommer ou les composer entre eux.

- **Opérateur de différence ordinaire**

On note ∇ opérateur de différence ordinaire associé à un processus $\{ X_t, t \in \mathbb{Z}\}$ tel que :

$$\nabla X_t = X_t - X_{t-1} = (1 - B) X_t.$$

On définit le $d^{ème}$ opérateur de différence ordinaire par : $\nabla^d X_t = (1-B)^d\ X_t$

- **Opérateur de différence saisonnière**

On note ∇_S l'opérateur de différence saisonnière associé à un processus $\{ X_t, t \in \mathbb{Z}\}$ tel que :

$$\nabla_S X_t = (1 - B^S)\ X_t = X_t - X_{t-s}\ .$$

On définit le $D^{ème}$ opérateur de différence saisonnière par : $\nabla_S^D X_t = (1-B^S)^D X_t$

II.5 Classe des modèles ARMA

II.5.1 Processus Autorégressif d'ordre *p*

a- Définition

Le processus $(X_t)_t$ satisfait à une représentation AR d'ordre p, noté ***AR(p)***, s'il est solution de l'équation aux différences stochastique suivante :

$$\varepsilon_t = X_t - \sum_{j=1}^{p} \phi_j X_{t-j}$$

Ou encore

$\varepsilon_t = \phi(B) X_t$ Avec $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - ... - \phi_p B^p$ et $\phi_p \in \mathbb{R}^*$.

Où $\phi(B)$ représente le polynôme de retard et ε_t est un bruit blanc de moyenne nulle et de variance σ_ε^2.

b- Théorème (condition de stationnarité)

Une condition nécessaire et suffisante pour que le processus autorégressif soit stationnaire du second ordre est que les racines de l'équation caractéristique suivante $\phi(Z) = 0$ soient à l'extérieur du cercle unitaire.

c- Caractéristiques d'un processus *AR (p)*

- Le corrélogramme simple est caractérisé par une décroissance géométrique de ses termes.
- Le corrélogramme partiel à ses seuls p premières termes différents de zéro.

Remarque

On peut ajouter au modèle une constante qui ne modifie pas ses caractéristiques stochastiques et qui peut être utile pour expliquer quelques phénomènes économiques.

Cas particulier : soit le processus stationnaire ***AR (1)*** : $X_t = \phi X_{t-1} + \varepsilon_t$

Ce processus est dit processus de Markov car l'observation X_t dépend seulement de l'observation précédente X_{t-1}.

d- Les équations de Yule-Walker

Soit le processus autorégressif stationnaire d'ordre p suivant :

$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + \varepsilon_t$ … (I), avec ε_t bruit blanc de variance notée σ_ε^2.

En multipliant (I) par X_t on obtient : $X_t^2 = \sum_{i=1}^{p} \phi_i X_{t-i} X_t + \varepsilon_t X_t$ et en prenant l'espérance mathématique, on aura :

$$E\left[X_t^2\right] = \gamma(0) = \sum_{i=1}^{p} \phi_i \gamma(i) + \sigma_\varepsilon^2$$

D'où :

$$\gamma(0) = \frac{\sigma_\varepsilon^2}{1 - \sum_{i=1}^{p} \phi_i \rho(i)}$$

En multipliant maintenant (I) par $X_{t-h}, h > 0$ et prenant l'espérance, ensuite divisant par $\gamma(0)$, on obtient : $\rho(h) = \sum_{i=1}^{p} \phi_i \rho(h-i), \ \forall h > 0.$ En écrivant cette équation pour $h = 1, ..., p$ on obtient les équations dites de Yule-Walker suivantes :

$$\begin{pmatrix} \rho(1) \\ \rho(2) \\ . \\ . \\ \rho(p) \end{pmatrix} = \begin{pmatrix} 1 & \rho(1) & \rho(2) & . & . & \rho(p-1) \\ \rho(1) & 1 & \rho(1) & . & . & \rho(p-2) \\ . & & & & & . \\ . & & & & & . \\ \rho(p-1) & . & . & . & \rho(1) & 1 \end{pmatrix} \times \begin{pmatrix} \phi_1 \\ \phi_2 \\ . \\ . \\ \phi_p \end{pmatrix}$$

II.5.2 Processus Moyenne Mobile d'ordre *q* (*Moving Average)*

a- Définition

Le processus $\{ X_t, \ t \in \mathbb{Z} \}$ satisfait à une représentation moyenne mobile d'ordre *q*, noté : ***MA(q)***, s'il est solution de l'équation aux différences stochastique suivante :

$$X_t = \varepsilon_t + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j}$$

En introduisant le polynôme de retard nous obtenons

$X_t = \theta(B)\varepsilon_t$ Où $\theta(B) = 1 + \theta_1 B + \cdots + \theta_q B^q$ et $\theta_J \in \mathbb{R}^*$

ε_t est un bruit blanc de moyenne nulle et de variance σ_ε^2.

Remarque

Le modèle moyenne mobile d'ordre *q (MA (q))*, explique la valeur de la série à l'instant t par une moyenne pondérée d'aléas ε_t, jusqu'à la $q^{éme}$ période qui sont supposés être générés par un processus de type bruit blanc.

b- Théorème (condition d'inversibilité)

Une condition nécessaire et suffisante pour que le processus moyenne mobile soit inversible est que les racines de l'équation caractéristique suivante : $\theta(Z) = 0$ soient à l'extérieur du cercle unitaire.

Soit le processus *MA(1)* : $X_t = \theta \varepsilon_{t-1} + \varepsilon_t$

$$1-\theta Z=0 \Rightarrow Z=\frac{1}{\theta} \Rightarrow |Z|>1 \Rightarrow |\theta|<1$$

c- Caractéristiques d'un Processus *MA (q)*

Un processus moyen mobile d'ordre q est toujours stationnaire, car il est une combinaison linéaire finie d'un processus stationnaire $\{\varepsilon_t,\ t\in\mathbb{Z}\}$.

Pour le corrélogramme simple seuls ses q premiers termes sont différents de zéro. La fonction d'autocorrélation est dite tronquée au-delà du $q^{iéme}$ retard puisque la fonction d'autocorrélation est définie par :

$$\rho_k=\begin{cases}\dfrac{\sum_{i=0}^{q-k}\theta_i\theta_{i+k}}{\sum_{i=0}^{q}\theta_i^2} & \text{pour } k=0,\ldots,q\\ 0 & \text{pour } k>q\end{cases}$$

Le corrélogramme partiel est caractérisé par une décroissance exponentielle de ses termes.

Remarques

1- Un processus autorégressif est toujours inversible.

2- Il y a équivalence ente processus MA (1) et processus AR (p) avec p infini.

II.5.3 Processus Autorégressif Moyenne Mobile d'ordre *(p, q)*

a- Définition

Le processus *ARMA (p, q)* est généré par une combinaison des valeurs passées et des erreurs passées et présentes ; on l'exprime par l'équation : $\phi(B)X_t=\theta(B)\varepsilon_t$

Avec :

$\{\varepsilon_t, t\in T\}$: est un bruit blanc de variance σ_ε^2.

Et $\theta(B)=1+\theta_1B+\theta_2B^2+\cdots+\theta_qB^q$

$\phi(B)=1-\phi_1B-\phi_2B^2-\cdots-\phi_qB^q$

$\theta_i\in\mathbb{R},\ \forall i=1,....,q$ et $\phi_i\in\mathbb{R},\ \forall i=1,...,p.$

b- Théorème (condition de stationnarité et d'inversibilité)

1)- Une condition nécessaire et suffisante pour que le processus autorégressif moyenne mobile d'ordre *(p, q)* soit stationnaire est que les racines de l'équation caractéristique suivante : $\phi(Z)=0$ soient à l'extérieur du cercle unitaire.

2)- Une condition nécessaire et suffisante pour que le processus autorégressif moyenne mobile d'ordre *(p,q)* soit inversible est que les racine de l'équation caractéristique suivante: $\Theta(Z) = 0$ soient à l'extérieur du cercle unitaire.

c- Caractéristique d'un Processus *ARMA (p, q)*

Les corrélogrammes simple et partiel sont un mélange des fonctions exponentielles et sinusoïdales amorties. Cependant l'identification des paramètres p et q à partir de l'étude des fonctions d'autocorrélations empiriques s'avère plus délicate.

Remarques

1- Les processus AR, MA et ARMA ne sont représentatifs que de séries stationnaires en tendance et corrigées des variations saisonnières. De plus ces modèles ne tiennent pas compte d'une éventuelle variable exogène.

2- dans un contexte ultérieur, nous allons exposer une extension de la classe des modèles ARMA.

II.6 Processus aléatoire non stationnaire

La plupart des résultats et méthodes utilisés dans l'analyse des séries chronologiques sont basés sur l'hypothèse de la stationnarité du second ordre, lorsque cette hypothèse n'est pas satisfaite, ce qui est souvent rencontré en pratique dans diverses disciplines de recherche, en particulier, l'économie d'hydrologie, la météorologie…des transformations sont appliquées (différence ordinaire, différence saisonnière, différence mixte, transformation de Box-Cox …) pour assurer la stationnarité du second ordre.

Pour que ces transformations soient adéquates il faut à priori pouvoir détecter correctement la nature des variations de la série. Pour répondre à ce besoin, plusieurs techniques ont été mises au point afin de détecter la tendance, la saisonnalité, la rupture,… .

II.6.1 Composantes des séries temporelles

Avant le traitement d'une série chronologique, il convient d'en étudier ses caractéristiques stochastiques (son espérance et sa variance), si elles se trouvent modifiées dans le temps la série est considérée non stationnaire.

L'analyse des séries temporelles des phénomènes économiques permet de distinguer quatre types d'évolution des séries dans le temps appelées « composantes de la série temporelle » qui sont :

(a) Tendance

C'est un mouvement persistant dans un sens déterminé pendant un intervalle de temps assez long, il traduit l'allure globale du phénomène, qu'il soit à la baisse ou à la hausse, ce mouvement est fonction du temps.

(b) Saisonnalité

Elle se manifeste à travers des fluctuations périodiques plus au moins régulières (sous réserve de la variabilité du nombre de jours dans le mois, du nombre de jours fériés dans la semaine). Ce mouvement est donc une fonction du temps et est indépendant de la tendance.

Néanmoins, l'explication de ce mouvement se trouve dans des déterminismes extérieurs à l'activité économique elle-même (particularités du temps astronomique, rythme des saisons, rythme des activités sociales dont les caractères institutionnels s'imposent à l'activité économique).

(c) Cycle

Cette composante décrit un mouvement à moyen terme caractérisé à la fois par la périodicité et par la cyclicité, c'est-à-dire par la régularité de son amplitude comportant une phase croissante et une autre décroissante.

II.6.2 Méthode graphique

La représentation graphique de la chronique permet de détecter la présence d'une tendance, d'un cycle, d'une saisonnalité ou d'une modification de structure (rupture).

Aussi, l'étude de la fonction d'autocorrélation (corrélogramme) : qui consiste à analyser le corrélogramme simple permet de détecter si :

- Des pics marquants apparaissent aux retards S, 2S, 3S…, ce qui fait penser de la présence d'une saisonnalité de période S.
- La fonction d'autocorrélation ne décroît pas d'une manière rapide vers zéro, ce qui fait croire à la présence d'une tendance.

II.6.3 Méthodes Analytiques

II.6.3.1 Analyse de la Tendance

Certaines variables économiques peuvent avoir des évolutions analogues, dont il peut exister une corrélation entre ces variables sans que celle-ci exprime une quelconque liaison à caractère explicatif, donc il convient d'enlever cette tendance et voir si une telle liaison existe. En analyse des séries chronologiques, on distingue deux types de tendances : **déterministe** et **stochastique**. Dans cette optique, Nelson et Plosser (1982), ont développé deux sortes de processus non stationnaire : TS (Trend Stationnary) et DS (Differency Stationnary).

a- Processus TS (***T**rend **S**tationnary*)

Il arrive que les valeurs prises par les variables d'un processus stochastique décrivent une allure déterministe qui peut être représentée à travers une fonction polynomiale (linéaire ou non linéaire) du temps, de la façon suivante :

$$X_t = f_t + U_t.$$

Avec f_t : Fonction polynomiale par rapport au temps.

U_t : Processus stationnaire.

Un exemple simple est : $X_t = a + bt + U_t$

Avec : $Var(X_t) = \sigma_\varepsilon^2$ et $\text{cov}(X_t, X_s) = 0, s \neq t.$

Dans ce cas X_t est dit : **un processus non stationnaire de type déterministe**. Cela veut dire que l'effet produit par un choc (ou par plusieurs chocs aléatoires) à un instant t est transitoire, la chronique retrouve son mouvement de long terme dicté par les valeurs de la fonction f_t.

- Pour rendre stationnaire un tel processus, on doit estimer d'abord a *et* b par la méthode des moindres carrés ordinaires (MCO), puis retrancher de X_t la valeur estimée $\hat{a} + \hat{b}t$.

b- Processus DS (***D**ifferency **S**tationnary*)

C'est un processus non stationnaire de type aléatoire, dont un choc à un instant donné se répercute à l'infini sur les valeurs de la série, l'effet choc est donc permanent et va en décroissance. La stationnarisation de ce type de processus est réalisée par l'utilisation d'un filtre à la différence d'ordre d : $(1-L)^d X_t = \beta + \varepsilon_t$

Avec β :constante réelle.

ε_t : Processus stationnaire d'espérance nulle.

- En pratique on utilise souvent la différence d'ordre 1 :

$$\nabla X_t = \beta + \varepsilon_t \Leftrightarrow X_t = X_{t-1} + \beta + \varepsilon_t$$

On obtient, par substitution successive :

$$X_t = X_0 + \beta t + \sum_{i=1}^{t} \varepsilon_i$$

Le processus n'est pas stationnaire car on a :

$Var(X_t) = t\sigma_\varepsilon^2.$

$\text{cov}(X_t, X_s) = \sigma_t^2 \min(s,t)$, $s \neq t.$

En fait, on distingue deux types de processus

(a) Si $\beta = 0$ alors le processus est dit sans dérive, il s'écrit sous la forme suivante :

$$X_t = X_{t-1} + \varepsilon_t$$

Comme ε_t est un bruit blanc, le modèle porte le non de marche aléatoire (*random walk Model)*, il est fréquemment utilisé en analyse de l'efficience des marchés financiers.

Test de racine unitaire (test de Dickey-Fuller 1979)

Le choix d'un processus DS ou TS comme structure de la chronique n'est pas neutre ; pour cela les tests de Dickey et Fuller permettent non seulement de détecter l'existence d'une tendance (racine unitaire, *unit root test*) mais aussi de déterminer son type et par conséquent la bonne manière de stationnariser la chronique.

Les modèles servant de base à la construction de ces tests qu'on estime par la méthode des moindres carrés ordinaires sont les suivants :

Modèle [1] : $X_t = \rho X_{t-1} + \varepsilon_t$.modèle autorégressif d'ordre 1.

Modèle [2] : $X_t = \rho X_{t-1} + c + \varepsilon_t$. modèle autorégressif d'ordre 1 avec constante.

Modèle [3] : $X_t = \rho X_{t-1} + c + bt + \varepsilon_t$.modèle autorégressif d'ordre 1 avec tendance et constante.

Avec $\varepsilon_t \rightarrow iid(0,\ \sigma_\varepsilon^2)\quad (bruit\ blanc)$.

- Les hypothèses de test sont :

$H_0 : \rho = 1$ contre $H_1 : |\rho| < 1$

- Si dans l'un des trois modèles cités ci dessus l'hypothèse nulle est vérifiée, le processus est alors non stationnaire (le processus suit une marche aléatoire). Les observations présentes et passées ont la même importance, on détermine dans ce cas l'ordre d'intégration d.

- Sinon, la série est stationnaire (le processus est asymptotiquement stationnaire), i.e., l'observation présente est plus importante que les observations passées.

- Si $|\rho| > 1$ alors la série n'est pas stationnaire la variance augmente de façon exponentielle avec le temps, les observations passées ont un effet persistant sur les observations présentes et futures, dans ce cas le processus est dit explosif.

Donc Dickey et Fuller ont tabulé, à l'aide de simulation de Monte Carlo, les valeurs critiques pour des échantillons de différentes tailles[9].

Ce qui ne pose aucun problème puisqu'il est équivalent de tester comme hypothèse nulle :

$H_0 : \hat{\rho} = 1$ ou $H_0 : \hat{\rho} - 1 = 0.$

[9] Les valeurs théoriques sont données par les plupart des logiciels économétriques en particulier par (*EVIEWS*).

Le déroulement de test

On estime par la méthode des moindres carrés ordinaires le paramètre $\hat{\rho}$ et l'écart type pour chaque modèle, ce qui fournit $t_{\hat{\rho}}$ avec $t_{\hat{\rho}} = \dfrac{\hat{\rho} - 1}{\hat{\sigma}_{\varepsilon}}$

- Si $t_{\hat{\rho}} > t_{tabulée}$ alors H_0 est acceptée, c'est à dire, il existe une racine unitaire et le processus n'est pas stationnaire.

Test de Dickey-Fuller Augmenté (1981) (ADF)

Dans les tests de Dickey-Fuller simples les résidus sont supposés être des bruits blancs et donc non corrélés, ce qui n'est pas forcément le cas. Pour cela Dickey-Fuller (1981) ont proposé une généralisation de cette approche en considérant une représentation AR (p) de X_t.

Après transformation des modèles de base, les tests ADF sont fondés, sous l'hypothèse alternative $|\rho| < 1$, sur l'estimation par la méthode des moindres carrés ordinaires des modèles suivants :

Modèle [4] : $\nabla X_t = \hat{\phi} X_{t-1} - \sum_{j=1}^{p} \hat{\varphi}_j X_{t-j+1} + \varepsilon_t$ modèle autorégressif d'ordre p.

Modèle [5] : $\nabla X_t = \hat{\phi} X_{t-1} - \sum_{j=1}^{p} \hat{\varphi}_j X_{t-j+1} + c + \varepsilon_t$ modèle autorégressif d'ordre p avec constante

Modèle [6] : $\nabla X_t = \hat{\phi} X_{t-1} - \sum_{j=1}^{p} \hat{\varphi}_j X_{t-j+1} + bt + c + \varepsilon_t$ modèle autorégressif d'ordre p avec tendance et constante.

Où $\varepsilon_t \rightarrow iid(0,\ \sigma_{\varepsilon}^2)$ (*bruit blanc*).

Le déroulement des tests est identique au cas Dickey-Fuller sur les modèles [4], [5] et [6], seules les tables statistiques de Dickey-Fuller diffèrent[10].

Remarque :

Les retards X_{t-j} $(j = 1,..., p)$ participent dans l'explication du dynamisme du processus ce qui implique la baisse, en valeur absolue, des autocorrélations résiduelles, donc le nombre de retard p est choisi suffisamment grand pour éliminer les autocorrélations des résidus, pour se faire, on commence par estimer les modèles pour les premiers ordres de j et on l'augmente au fur et à mesure jusqu'à l'obtention des résidus qui forment un bruit blanc[11].

[10] On utilise le test « Portemanteau » habituel pour tester si les résidus sont blanchis.
[11] Voir GOURIEROUX : « séries temporelles et modèles dynamiques »

On note que le test ADF ne permet pas de tester directement si les résidus forment un bruit blanc, donc il convient de faire une estimation des modèles pour pouvoir observer les résidus et éventuellement les tester.

II.6.3.2 Analyse de la saisonnalité

Analyse de la variance et test de Fisher (test d'ANOVA)

Il s'agit de s'assurer que l'effet régulier que manifeste la série n'est pas une coïncidence due au seul fait du hasard, et qu'il ne s'agit pas aussi, d'oscillations plus au moins régulières d'un effet parasite dû à l'existence de corrélation non nulle entre les valeurs successives du processus (effet de Yule (1921), Slutsky (1937)).

Afin de ne pas se tromper dans l'interprétation de cette régularité un test dit d'ANOVA, basé (comme son nom l'indique) sur l'analyse de la variance des résidus a été mis au point.

- Soit une série chronologique mensuelle, trimestrielle, journalière, …, brute telle que :
- T = N×P : la taille de la série

Où

N : le nombre d'années.

P : le nombre d'observations dans l'année appelées période.

x_{ij} : l'observation de la série pour la $i^{ème}$ année et la $j^{ème}$ période, avec : i = 1…N et j = 1…P.

On suppose que la chronique est sans tendance ou la tendance a été retirée.

Le modèle s'écrit $x_{ij} = a_i + b_j + e_{ij}$

où : a_i : l'effet de la ième année ;

b_j : l'effet de la jème période ;

e_{ij} : résidus indépendants avec $e_{ij} \rightarrow N(0,\sigma^2)$.

Principe du test

On test ceux effets absents (par exemple : mois, année) contre deux effets significativement présents.

Si l'effet périodique (mois par exemple) est significatif alors la série est saisonnière, par contre si l'effet année est significatif, alors soit que la chronique n'a pas été transformée et de ce fait, elle possède des paliers horizontaux, ou bien la chronique a été transformée ce qui implique la présence de changements de tendance.

Déroulement du test

- Le calcul de la somme totale des carrés ajustés S_T :

$S_T = \sum_{i=1}^{N}\sum_{j=1}^{P}(x_{ij} - \bar{x})^2$ avec $\bar{x} = \frac{1}{N \times P}\sum_{i=1}^{N}\sum_{j=1}^{P} x_{ij}$ (La moyenne totale).

- Le calcul de la somme des carrés annuelle SA :

$S_A = p\sum_{i=1}^{N}\left(x_i - \bar{x}\right)^2$ avec $x_i = \frac{1}{p}\sum_{j=1}^{p} x_{ij}$ (La moyenne de de la i ème année).

- Le calcul de la somme des carrés périodique SP :

$S_P = N\sum_{j=1}^{N}(x_{.j} - \bar{x})^2$ avec $x_{.i} = \frac{1}{N}\sum_{j=1}^{N} x_{ij}$ (La moyenne de la j ème période)

Le calcul de la somme des carrés résiduels SR :

$S_R = \sum_{i=1}^{N}\sum_{j=1}^{P}(x_{ij} - x_{.j} - x_{i.} - \bar{x})^2$

Le calcul de la variance année : $Var_A = \frac{S_A}{N-1}$

- Le calcul de la variance périodique : $Var_P = \frac{S_P}{P-1}$

- Le calcul de la variance résidu : $Var_R = \frac{S_R}{(P-1)(N-1)}$

Le test de saisonnalité

Ce test est basé sur l'influence du facteur période et est construit à partir des hypothèses

Suivantes : $\begin{cases} H_0 : \text{Pas de saisonnalité.} \\ H_1 : \text{Il existe une saisonnalité.} \end{cases}$

On calcule la valeur de Fisher empirique $F^* = Var_p \big/ Var_R$ que l'on compare à la valeur de Fisher

tabulée $F^{\alpha}_{\nu 1 \nu 2}$ avec $\nu 1$=P-1, $\nu 2$=(N-1)(P-1) degré de liberté.

Si $F^* > F^{\alpha}_{v1,v2}$ on rejette l'hypothèse H_0, la série est saisonnière.

Si $F^* < F^{\alpha}_{v1,v2}$ on rejette l'hypothèse H_1, la série n'est pas saisonnière.

II.6.3.3 Méthode de dessaisonalisation

Pour stationnariser une série affectée d'une saisonnalité, on procède à la dessaisonalisation de la série par une différentiation.

Application de la différence saisonnière

Cette méthode consiste à considérer la série différenciée d'ordre S (S : période de la saisonnalité).

$\nabla_s X_t = X_t - X_{t-s}$

II.7 Extension des Modèles ARMA

L'objectif de cette extension est de tenir en compte des effets (tendance, saisonnalité) dans la modélisation de la chronique, sans avoir recours aux méthodes exposées ci-dessus (pour rendre la série stationnaire).

II.7.1 Processus Autorégressif Moyenne Mobile Intégré d'Ordre (p,d,q)

Ce sont des modèles ARMA intégrés notés ARIMA. Ils sont issus des séries stationnaires par l'application du filtre aux différences et ceci, bien entendu dans le cas des processus DS détectés par le test de Dicky-Fuller.

Le processus X_t suit un ARIMA (p,d,q), c'est-à-dire qu'il est solution d'une équation aux différences stochastique du type : $\phi(B)(1-B)^d X_t = \Theta(B)\varepsilon_t$.

II.7.2 Processus Autorégressif Moyenne Mobile Intégré Saisonnier

Il est possible de trouver que certaines séries chronologiques peuvent être caractérisées par une allure graphique périodique, pour cela il est important de les analyser en tenant compte de l'effet saisonnier. Box et Jenkins (1970) ont proposé une classe particulière de modèles appelée : classe de modèles ARIMA saisonniers.

II.7.3 Modèles Saisonniers Mixtes SARIMA

Ce sont des extensions des modèles ARMA et ARIMA. Ils représentent généralement des séries marquées par une saisonnalité comme c'est le plus souvent le cas pour des séries économiques voire financières. Ces séries peuvent mieux s'ajuster par des modèles saisonniers. Ce sont les SARIMA (p, d, q)(P,D,Q) qui répondent à la formulation :

$\phi(B)\phi_s(B^s)(1-B)^d(1-B^s)^D X_t = \theta(B)\theta_s(B^s)\varepsilon_t$ où :

$\phi(B) = 1-\phi_1 B-\phi_2 B^2-\cdots-\phi_p B^p$: Polynôme autorégressif non saisonnier d'ordre p.

$\phi(B) = 1-\phi_{1s} B-\phi_{2s} B^{2s}-\cdots-\phi_{ps} B^{ps}$: Polynôme autorégressif saisonnier d'ordre P.

$\theta(B) = 1+\theta_1 B+\theta_2 B^2+\cdots+\theta_q B^q$: Polynôme moyenne mobile non saisonnier d'ordre q.

$\theta(B) = 1+\theta_{1s} B+\theta_{2s} B^{2s}+\cdots+\theta_{Qs} B^{Qs}$: Polynôme moyenne mobile saisonnier d'ordre Q.

$(1-B)^d$: Opérateur de différence d'ordre d.

$(1-B^s)^D$: Opérateur de différence saisonnière d'ordre D. s : correspond à la saisonnalité.

$\varepsilon_t \rightarrow BB\left(0,\ \sigma_\varepsilon^2\right)$.

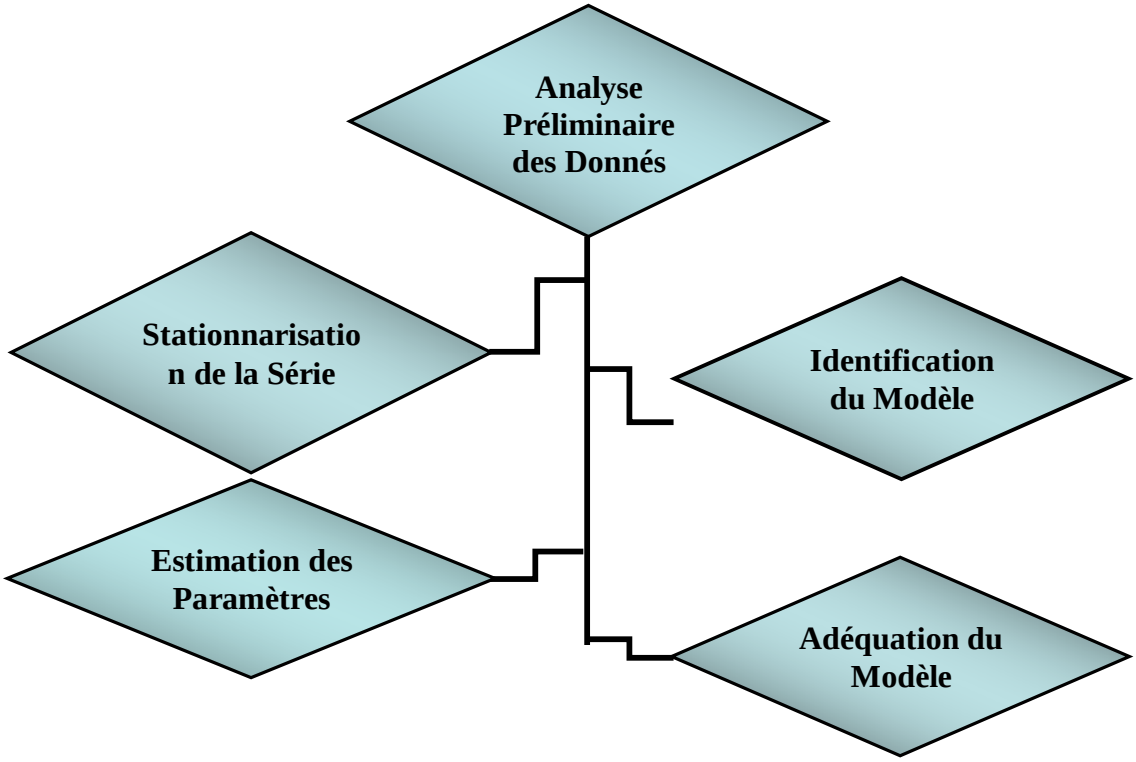
II.8 La Méthodologie de Box & Jenkins

Introduction

L'analyse des séries temporelles est un champ d'étude en perpétuelle évolution ces dernières années. D'énormes progrès ont été réalisés dans diverses disciplines, notamment en économie, finance,….En effet Wold (1938) est à la base du développement qu'a connu la classe des modèles autorégressifs moyennes mobiles (ARMA) univariés.

Cependant, les statisticiens George Box et Gwilym Jenkins ont contribué dans les années 70, à populariser la théorie des séries temporelles univariées par leur célèbre ouvrage. La modélisation univariée de Box & Jenkins concerne les processus *ARMA (p, q), ARIMA (p, d, q)* ou *SARIMA (p, d, q)(P, D ,Q).* Ces auteurs rassemblent tous les travaux dans une méthodologie itérative. Cette dernière englobe trois étapes essentielles à savoir : l'identification du modèle, l'estimation du paramètre et la validation à travers des tests. Une fois le modèle déterminé, nous pouvons faire des prévisions.

La première étape consiste à identifier le modèle qui pourrait engendrer la série. Elle consiste, d'abord à transformer la série afin de la rendre stationnaire. Le nombre de différentiations détermine l'ordre d'intégration d. Ensuite il s'agit d'identifier le modèle *ARMA (p, q)* de la série transformée avec l'aide du corrélogramme simple et du corrélogramme partiel. Le graphique des coefficients d'autocorrélation simple (corrélogramme simple) et d'autocorrélation partielle (corrélogramme partiel) donnent une information sur l'ordre du modèle *ARMA.* Après avoir choisi un ou plusieurs modèles *ARMA* théoriques, il faut estimer leurs paramètres en utilisant une méthode non linéaire (moindres carrés non linéaires ou maximum de vraisemblance). Ces méthodes sont appliquées en utilisant les degrés (*p, d,* q) et (*P, D, Q*) trouvés dans l'étape d'identification. Une fois les coefficients estimés, il s'agit de vérifier l'adéquation du modèle aux observations. Il existe plusieurs tests : tests graphiques de l'autocorrélation des résidus, test de Box-Ljung, et d'autres tests qui confirment la blancheur des résidus. Enfin, l'intérêt de l'approche de Box-Jenkins est qu'une modélisation *ARMA* conduit à des prévisions optimales si la variance de l'erreur de prévision est minimale. Cette approche se schématise comme suit :



- Organigramme de la méthodologie de Box et Jenkins -

II.8.1 Démarche de la méthode de Box et Jenkins

II.8.1.1 Analyse préliminaire

L'analyse préliminaire est une phase non coûteuse, elle permet avant tout test ou traitement statistique approprié, d'observer la représentation graphique de la série (les observations du processus $\{X_t, t \in \mathbb{Z}\}$ en fonction du temps).

En effet, parfois une simple visualisation du graphe permet de détecter ou soupçonner l'existence de plusieurs composantes (tendance, saisonnalité,…) donc il faut, bien évidemment, confirmer ou infirmer l'existence par des tests appropriés.

Cette étape, permet aussi de prendre des options sur les variables, tels que : corriger les données aberrantes, suppléer celles manquantes ou effectuer des transformations…etc.

II.8.1.2 Stationnariser la série

Les résultats de l'analyse des séries chronologiques reposent sur l'hypothèse de stationnarité du second ordre, mais souvent les caractéristiques stochastiques d'une série (moyenne, variance) se trouvent modifiées dans le temps, c'est le cas par exemple lorsque :

- On constate que la série est saisonnière par l'apparition des pics marquants de périodicité S dans la fonction d'autocorrélation simple ou dans la représentation graphique de la série.
- La chronique est affectée d'une tendance, dont il convient de déterminer la nature par les tests cités de Dickey-Fuller (voir chapitre 1).

II.8.1.3 L'identification du modèle adéquat

Cette étape consiste à identifier le modèle *ARMA* susceptible de représenter la série, c'est pour cela qu'il est important de se familiariser avec les données en examinant le graphe de la série chronologique (présence de saisonnalité, stationnarité,...) qui permet de faire une analyse préliminaire qui consiste par exemple à corriger les données aberrantes, transformer les données (transformation logarithmique, inverse, racine carrée,...) puisqu'il faut se ramener à un modèle *ARMA* stationnaire, le recours aux différences premières ordinaires, différences premières saisonnières, différences ordinaires et saisonnières. Le choix est dicté par l'allure graphique de la série. D'ailleurs le choix de la transformation des données est plus facile après avoir appliqué les opérateurs de différence adéquats. Il est conseillé de comparer les variances des différentes séries. La série avec la plus petite variance conduit souvent à la modélisation la plus simple. Ainsi un examen du corrélogramme s'impose. Cette phase est la plus importante et la plus difficile, elle consiste à déterminer, parmi l'ensemble des modèles ARMA(p,q), le modèle le plus représentatif du phénomène étudié, elle est fondée sur l'étude des corrélogrammes: simple et partiel; l'idée de base est que chaque modèle ARMA possède des fonctions d'autocorrélation (simple et partielle) théoriques spécifiques, le statisticien essaie donc, à l'aide de son expertise, de reconnaître et d'identifier, en comparant l'éventuelle similitude de ces fonctions théoriques et estimées. Il peut alors choisir un ou plusieurs modèles théoriques (en général, le choix porte sur trois modèles au plus) en se basant sur les propriétés suivantes :

- Si le corrélogramme simple n'a que ses q premiers termes différents de zéro et que les termes du corrélogramme partiel diminuent exponentiellement vers zéro ou d'une manière sinusoïdale amortie, nous pouvons pronostiquer un moyenne mobile d'ordre q : MA (q).
- Si le corrélogramme partiel n'a que ses p premiers termes différents de zéro et que les termes du corrélogramme simple diminuent exponentiellement vers zéro ou d'une manière sinusoïdale amortie, nous identifions un autorégressif d'ordre p : AR (p).

● Si les fonctions d'autocorrélation simple et partiel n'apparaissent pas tronquées, il s'agit d'un processus ARMA ; en fait dans ce cas il est très difficile d'identifier directement les vrais paramètres du modèle, il convient donc d'en proposer plusieurs pour éliminer, après des tests appropriés, ceux qui ne reflètent pas les variations du processus.

II.8.1.4 Estimation des paramètres du modèle

Une fois l'étape de l'identification terminée, il faut estimer les paramètres qui sont les coefficients des polynômes *AR* et *MA* ainsi que les polynômes saisonniers *SAR* et *SMA*, et la variance des résidus σ^2.

La méthode d'estimation la plus couramment utilisée est celle du maximum de vraisemblance ou bien la méthode des moindres carrés. Plus spécifiquement la technique consiste à construire une fonction appelée fonction de vraisemblance et à maximiser son logarithme par rapport aux paramètres Φ_i et θ_j (avec i := 1,.., p et j := 1,..., q) permettant de trouver la valeur numérique la plus vraisemblable pour ces paramètres. L'étape d'estimation achevée, l'étape suivante va nous permettre de valider le(s) modèle(s) estimé(s).

II.8.1.5 Validation

A l'étape de l'identification, les incertitudes liées aux méthodes employées font que plusieurs modèles en général sont estimés et c'est l'ensemble de ces modèles qui subit alors l'épreuve des tests, il en existe de très nombreux critères permettant de comparer les performances entre modèles ; nous pouvons citer les tests sur les paramètres et les tests sur les résidus.

II.8.1.6 Tests concernant les paramètres

Tous les coefficients du modèle retenu doivent être significativement différents de zéro, il convient donc d'utiliser le test de Student classique.

- ***Test de Student sur les paramètres***

Il s'agit dans cette étape de tester la significativité des paramètres Φ_i et θ_j (i=1…,p et j = 1…,q) dans la formulation obtenue. Nous rejetterons avec un risque 5% l'hypothèse que le paramètre est nul si :

$$\frac{\left|\hat{\Phi}_i\right|}{\sqrt{Var\left(\hat{\Phi}_i\right)}} > t_\alpha \quad \left(t_\alpha = 1{,}96\right)$$ (Même procédure pour les θ_j).

II.8.1.7 Tests sur les Résidus

Le processus estimé est évidemment de bonne qualité si la chronique calculée suit les évolutions de la chronique empirique. Les résidus entre les valeurs observées et les valeurs calculées par le modèle, doivent se comporter comme un bruit blanc. Pour montrer que les $\{\varepsilon_t,\ t \in \mathbb{Z}\}$ forment un bruit blanc, nous devons vérifier si :

-La moyenne des résidus est nulle, sinon il convient d'ajouter une constante au modèle.

-Le graphe des résidus en fonction du temps semble approximativement compatible avec une suite de variables aléatoires non corrélées. C'est ainsi que nous proposerons une multitude de tests concernant les caractéristiques du résidu souhaité.

- ***Test de Box-Ljung***

Lorsque le processus est bien estimé, les résidus entre les valeurs observées et les valeurs estimées par le modèle doivent se comporter comme un bruit blanc. Nous noterons par la suite $\hat{\varepsilon}_t$ le résidu d'estimation du modèle.

- ***Principe du test***

Ce test permet de savoir si les résidus forment un bruit blanc ou non, pour le réaliser : nous observons le corrélogramme des erreurs du modèle optimal, si tous les pics sont dans la bande de confiance de plus la probabilité de significativité est supérieure à 0.05 alors les résidus forment un bruit blanc.

Pour confirmer ce résultat nous testons

H_0 : « Les autocorrélations au pas K, (k=N/5) sont non corrélés » C'est-à-dire

H_0 : « $\rho_1 = \rho_2 = ... = \rho_K = 0$ » Contre H_1 : « $\exists\, \rho_j : j = \overline{1,k}$ tel que $\rho_j \neq 0$ ».

$Q = n(n+2)\sum_{i=1}^{K} \frac{\rho_i^2(\varepsilon)}{n-i}$ Statistique de BOX-LJUNG au pas *K* avec :

K : nombre de retard choisi.

N : Taille de la série brute.

n : nombre de résidus.

- ***Règle de décision***

Si $Q < \chi^2(K - p - q - P - Q)$, degrés de liberté nous acceptons l'hypothèse H_0 que les résidus sont non corrélés, Sinon les résidus ne forment pas un bruit blanc et le modèle est inadéquat.

- ***Test des Points de Retournements***

Nous dirons que la suite des données $\varepsilon_1, \varepsilon_2, ..., \varepsilon_n$ présente un point de retournement à la date

i, si $\begin{cases} \varepsilon_i < \varepsilon_{i+1} > \varepsilon_{i+2} \\ \varepsilon_i > \varepsilon_{i+1} < \varepsilon_{i+2} \end{cases}$ i = 1,2,…n-2

Soit la variable aléatoire $X_i = \begin{cases} 1 & si\ c'est\ un\ point\ de\ retournement \\ 0 & si\ non \end{cases}$

La variable X_i suit la loi de Bernoulli de paramètre $p = 2/3$

Le nombre total des points de retournement est $p = \sum_{i=1}^{n-2} x_i$

Nous avons : $E(p) = \sum_{i=1}^{n-2} E(x_i) = \frac{2}{3}(n-2)$;

$E(p^2) = E\left(\sum_{i=1}^{n-2} x_i\right)^2 = \frac{40n^2 - 144n + 131}{90}$ Donc $Var(p) = \frac{16n - 29}{90}$

Sous l'hypothèse que les (ε_i) forment une suite de variables aléatoires, indépendantes et identiquement distribuées.

La statistique $U = \frac{|p - E(p)|}{\sqrt{Var(p)}}$ suit la loi normale d'espérance nulle et de variance égale à 1 (U $\rightarrow$ N (0,1) ; (*n* (nombre d'observations) > 30).

Le principe de ce test est d'accepter l'hypothèse que les $(\varepsilon_i)_i$ (les résidus du modèle) forment un bruit blanc si U < 1.96, au seuil $\alpha = 0.05$.

- ***Test de la nullité de la moyenne des résidus***

Soit *T* le nombre de données disponibles (après avoir enlevé les retards correspondant aux termes *AR* et *MA*). Si le processus $\{\varepsilon_t, t \in Z\}$ est i.i.d. $(0, \sigma_\varepsilon^2)$, nous devons avoir :

$$\bar{\varepsilon}_t = \frac{1}{T}\sum_{t=1}^{T} \hat{\varepsilon}_t \quad \underset{T \to \infty}{\longrightarrow} 0$$

Par application du Théorème central limite, nous montrons que :

$$\frac{\bar{\varepsilon}_t}{\hat{\sigma}_{\varepsilon_t}} \sqrt{T} \quad \underset{T \to \infty}{\xrightarrow{L}} N(0,1)$$

Dés lors, nous pouvons tester la nullité de la moyenne des résidus en construisant l'intervalle de confiance sur $\bar{\varepsilon}_t$ au seuil standard de 95%.

$$P\left\{\bar{\varepsilon}_t \in \left[\frac{-1{,}96\cdot\hat{\sigma}_{\varepsilon_t}}{\sqrt{T}}, \frac{1{,}96\cdot\hat{\sigma}_{\varepsilon}}{\sqrt{T}}\right]\right\} = 0{,}95.$$

Le test basé sur la statistique de Student pour tester l'hypothèse

H_0 : « m=0 » contre H_1 : « m$\neq$ 0 ».

La statistique utilisée est : $t = \dfrac{\bar{\varepsilon}_t}{\sigma_\varepsilon / \sqrt{n-1}}$

H_0 Est acceptée si $|t| < t_{n-1}$ à 5% (=1,96) pour n >30, dans le cas contraire, il convient d'ajouter une constante au modèle.

- ***Tests de normalité***

Le test de Jarque & Bera *(1984)* peut s'appliquer pour tester la normalité des résidus.
Ce dernier est fondé sur la notion de Skewness (l'asymétrie de la distribution, moment d'ordre 3) et de Kurtosis (l'aplatissement qui se traduit en particulier par une épaisseur des queues de distribution, moment d'ordre 4). Soit μ_k le moment empirique d'ordre k du processus

$$\mu_k = \frac{1}{T}\sum_{t=1}^{T}\left(\hat{\varepsilon}_t - \bar{\varepsilon}\right)^k .$$

- ***Test de Skewness***

La Skewness est une mesure de l'asymétrie de la distribution de la série autour de sa moyenne.
Le coefficient de Skewness *(S_k ou encore β_1)* est défini par :

$$\left(S_k\right)^{1/2} = (\beta_1)^{1/2} = \frac{\mu_3}{\mu_2^{3/2}} \underset{T\to\infty}{\overset{L}{\rightarrow}} \mathrm{N}\left(0, \sqrt{\frac{6}{T}}\right)$$

La Skewness d'une distribution symétrique, telle que la distribution normale est nulle. Une Skewness positive signifie que la distribution a une queue allongée vers la droite et la Skewness négative signifie que la distribution a une queue allongée vers la gauche.

- ***Test de Kurtosis***

La Kurtosis mesure le caractère pointu ou plat de la distribution de la série. Le coefficient de Kurtosis *(k_u ou encore β_2)* est défini par :

$$k_u = \beta_2 = \frac{\mu_4}{\mu_2^2} \underset{T\to\infty}{\overset{L}{\rightarrow}} N\left(3, \sqrt{\frac{24}{T}}\right).$$

La Kurtosis de la distribution normale est 3. Si la Kurtosis est supérieure à 3, la distribution est plutôt pointue relativement à la normale ; si la Kurtosis est inférieure à 3, la distribution est plutôt plate relativement à la normale.

Nous construisons alors les statistiques centrées réduites correspondantes à $(S_k)^{1/2}$ et k_u que l'on compare aux seuils d'une loi normale centrée réduite

$$\gamma_1 = \frac{(S_k)^{1/2}}{\sqrt{\frac{6}{T}}} \underset{T\to\infty}{\overset{L}{\to}} N(0,1) \;\; et \;\; \gamma_2 = \frac{k_u - 3}{\sqrt{\frac{24}{T}}} \underset{T\to\infty}{\overset{L}{\to}} N(0,1)$$

Si la statistique centrée réduite de $(S_k)^{1/2}$ (γ_1) est inférieure au seuil 1,96 à 5%, nous acceptons l'hypothèse de symétrie. Si la statistique centrée réduite de k_u (γ_2) est inférieure au seuil 1,96 à 5%, nous acceptons l'hypothèse de queue de distributions non chargées (not weighted queues). La conjonction des deux conclusions nous fait accepter l'hypothèse de normalité.

- ***Test de Jarque-Bera***

La statistique de Jarque-Bera est une statistique de test pour examiner si la série est normalement distribuée. La statistique mesure la différence de la Skewness et de la Kurtosis de la série avec ceux de la distribution normale. La statistique est calculée comme suit :

$$JB = S = \frac{T}{6} S_k + \frac{T}{24}(k_u - 3)^2 \underset{T\to\infty}{\overset{\wp}{\to}} \chi^2(2)$$

Où S_k est la Skewness, k_u est la Kurtosis. Sous l'hypothèse nulle d'une distribution normale, la statistique de Jarque-Bera suit asymptotiquement une loi de χ^2 à deux degrés de liberté ; aussi, si $JB \geq \chi^2_{1-\alpha}(2)$ nous rejetons l'hypothèse H_0 de normalité des résidus au seuil.

- ***Test d'indépendance de Von-Neumann***

Ce test peut être effectué lorsque les résidus sont gaussiens.

Nous testons l'hypothèse nulle :

H_0 : « Les résidus sont indépendants et identiquement distribués » contre l'hypothèse

H_1 : « Au moins deux observations successives tendent à être corrélées ».

Les tests sont basés sur les deux estimateurs suivants de la variance σ_ε^2 des résidus :

$$D^2 = \frac{1}{n-1}\sum_{t=1}^{n}(\varepsilon_{t+1} - \varepsilon_t)^2 \;\; ; \;\; S'^2 = \frac{1}{n-1}\sum_{t=1}^{n}(\varepsilon_t - \bar{\varepsilon})^2$$

Sous H_0 : $E\left(\frac{D^2}{2S'^2}\right) = 1$ *et* $Var\left(\frac{D^2}{2S'^2}\right) = \frac{n-2}{n^2-1}$.

La statistique $U = \frac{\left(D^2/2S'^2\right)-1}{\sqrt{n-2/n^2-1}}$

sous l'hypothèse nulle, suit une loi Normale N (0,1).

La région critique est donnée par : $\{|U| > U_\alpha\}$, U_α est tel que $P[|U| > U_\alpha] = \alpha$.

- ***Test de Durbin-Watson***

Si les résidus $\{\varepsilon_t, t \in Z\}$ obéissent à un bruit blanc, il ne doit pas exister d'autocorrélation dans la série. Nous pouvons pour cela appliquer le test suivant :

Test de Durbin - Watson : repose sur l'hypothèse de normalité des résidus; test de l'autocorrélation d'ordre 1.

- ***Principe du test de Durbin – Watson***

Ce test permet de tester l'autocorrélation d'ordre 1 sous l'hypothèse que les résidus sont Gaussiens. Donc il teste l'hypothèse nulle

H_0 : « $\rho = 0$ » contre l'hypothèse alternative H_1 : « $\rho \neq 0$ ».Durbin et Watson ont proposé la statistique suivante :

$$DW = \frac{\sum_{t=2}^{n} (\varepsilon_t - \varepsilon_{t-1})^2}{\sum_{t=1}^{n} \varepsilon_t^2}$$

Le test de Durbin-Watson fait intervenir deux seuils critiques $d_{l,\alpha}$ et $d_{u,\alpha}$ ($d_{l,\alpha} < d_{u,\alpha}$) fonctions de n et du nombre de variables explicatives.

Ce test est utilisé pour tester trois hypothèses :

1- H_0 : « les résidus sont non corrélés » contre H_1 : « les résidus sont positivement corrélés »

Dans ce cas la règle de décision est :

i)- Si $DW < d_{l,\alpha}$: nous rejetons H_0.

ii)-Si $DW > d_{u,\alpha}$: nous acceptons H_0 .

iii)- Si $d_{l,\alpha} \leq DW \leq d_{u,\alpha}$: nous ne pouvons rien dire.

2- H_0 : « les résidus sont non corrélés » contre H_1 : « les résidus sont négativement corrélés »

la règle de décision est :

i)- Si (4-DW) < $d_{l,\alpha}$: nous rejetons H_0.

ii)- Si (4-DW) > $d_{u,\alpha}$: nous acceptons H_0 .

iii)-Si $d_{l,\alpha} \leq$ (4-DW) $\leq d_{u,\alpha}$: nous ne pouvons rien dire.

3- H_0 : « les résidus sont non corrélés » contre H_1 : « les résidus sont positivement ou négativement corrélés »

Dans ce cas :

i) - Si DW < $d_{l,\alpha/2}$ ou (4-DW) < $d_{l,\alpha/2}$: nous rejetons H_0.

ii) - Si DW > $d_{u,\alpha/2}$ ou (4-DW) > $d_{u,\alpha/2}$: nous acceptons H_0 .

iii) - Si $d_{l,\alpha/2} \leq$ DW $\leq d_{u,\alpha/2}$ ou $d_{l,\alpha/2} \leq$ (4-DW) $\leq d_{u,\alpha/2}$: nous ne pouvons rien dire.

- ***Test d'homoscédasticité***

L'hétéroscédasticité signifie que la dispersion des résidus a tendance à augmenter ou à diminuer en fonction des valeurs ajustées, plus généralement, elle se manifeste quand la dispersion des résidus varie en fonction des variables explicatives.

Non seulement L'hétéroscédasticité influence les tests de significativité mais surtout, elle fausse les intervalles de prévision.

Nous allons présenter, un test permettant de détecter une hétéroscédasticité éventuelle. Le test *ARCH* ou test du multiplicateur de Lagrange a été introduit par *Engle (1982).* Supposons que les résidus prévisionnels sont non corrélés et qu'ils obéissent à un modèle *ARCH* (dans la plupart des cas un modèle *ARCH* simple d'ordre *p).*Nous construisons alors une régression entre les résidus au carré et les résidus au carré décalés jusqu'à l'ordre p.

L'hypothèse nulle testée est celle d'homoscédasticité H_0:"$\alpha_1 = \alpha_2 = \ldots\ldots = \alpha_p = 0$" contre

L'hypothèse alternative d'hétéroscédasticité conditionnelle

H_1:"$\exists\, i\,,\ i = 1 \ldots p \quad tel\ que\ \alpha_i \neq 0$" .Si l'hypothèse H_0 est acceptée, la variance conditionnelle de l'erreur est constante $\sigma_\varepsilon^2 = \alpha_0$. En revanche, si l'hypothèse nulle est rejetée, les résidus suivent un processus *ARCH (p)* dont l'ordre *p* est à déterminer.

Le test est fondé soit sur un test de Fisher classique, soit sur le test du Multiplicateur de Lagrange *(LM).* La mise en œuvre du test est simple et peut s'effectuer en trois étapes :

- ***Etape1 :*** nous estimons l'équation de la moyenne. Nous récupérons les résidus estimés $\hat{\varepsilon}_t$ et nous calculons la série des $\hat{\varepsilon}_t^2$.
- ***Etape 2* :** Nous régressons $\hat{\varepsilon}_t^2$ sur une constante et sur ses *p* valeurs passées.
- ***Etape 3 :*** Nous calculons la statistique du Multiplicateur de Lagrange LM = n*R^2 où *n* est le nombre d'observations servant au calcul de la régression de l'étape 2 et R^2 est le coefficient de détermination de l'étape 2.

Sous l'hypothèse nulle d'homoscédasticité, la statistique TR^2 suit une loi de khi – deux à *p* degré de liberté. La règle de décision est :

- Si $LM < \chi^2(p)$, l'hypothèse nulle est acceptée : il n'existe pas d'effet ARCH.

- Si $LM \geq \chi^2(p)$, nous rejetons l'hypothèse nulle en faveur de l'hypothèse alternative d'hétéroscédasticité conditionnelle.

Une autre approche consiste à calculer le corrélogramme des résidus au carré du modèle initial. Si les premiers termes de ce corrélogramme sont significativement différents de zéro (0), alors nous pouvons conclure à un modèle de type *ARCH*. Sinon si tous les pics sont dans la bande de confiance, alors nous pouvons donc conclure que les résidus sont homoscédastiques.

II.8.2 Choix du Meilleur Modèle

Après les étapes précédentes, plusieurs formulations dans la vaste classe des modèles *ARMA* pourraient être retenus ; il faut donc choisir le meilleur modèle parmi ceux sélectionnés. Pour cela nous utilisons :

• Les Critères Standards

Ils sont fondés sur le calcul de l'erreur de prévision que l'on cherche à minimiser. Nous rappelons ici l'expression des trois critères les plus fréquemment utilisés.

-Erreur absolue moyenne (Mean Absolute Error) $\mathbf{MAE} = \frac{1}{N}\sum_t |\hat{\varepsilon}_t|$

-Racine de l'erreur quadratique moyenne (Root Mean Squared Error)

$$\mathbf{RMSE} = \sqrt{\frac{1}{N}\sum_t \hat{\varepsilon}_t^2}$$

-Ecart absolu moyen en pourcentage (Mean Absolute Percent Error)

$$\mathbf{MAPE} = 100 \cdot \frac{1}{N}\sum_t \left|\frac{\hat{\varepsilon}_t}{X_t}\right|$$

Où N est le nombre d'observation de la série X_t étudiée et $\hat{\varepsilon}_t$ désigne les résidus estimés. Plus la valeur de ces critères est faible, plus le modèle estimé est proche des observations.

• **Les Critères d'information**

L'idée sous – jacente consiste à choisir un modèle sur la base d'une mesure de l'écart entre la vraie loi inconnue et le modèle estimé. Cette mesure peut être fournie par la quantité d'information de Kullback. Les différents critères ont alors pour objet d'estimer cette quantité d'information. Il en existe plusieurs, Nous présentons ici les trois critères les plus fréquemment employés.

a) Critère d'information d'Akaike (AIC)(1969)

Le meilleur des modèles *ARMA (p, q)* est le modèle qui minimise la statistique :

$AIC(p, q) = n \, Log \, \hat{\sigma}_\varepsilon^2 + 2(p+q)$

b) Critère d'information Bayésien (BIC)

Ce critère présente l'avantage de pénaliser les modèles où les paramètres sont en surnombre comparativement à l'*AIC*. Il est donné par :

$$BIC(p, q) = n \, Log \, \hat{\sigma}_\varepsilon^2 - (n-p-q) \, Log\left[1 - \frac{p+q}{n}\right] + (p+q) \, Log\left[(p+q)^{-1}\left[\frac{\sigma_\varepsilon^2}{\hat{\sigma}_\varepsilon^2} - 1\right]\right]$$

c) Critère de Schwartz 1978

$SC(p, q) = n \, Log \, \hat{\sigma}_\varepsilon^2 + (p+q) \, Log \, n$

d) Critère de Hannan-Quin 1979

$$HQ(p, q) = Log \, \hat{\sigma}_\varepsilon^2 + (p+q) \, c \, Log\left[\frac{Log \, n}{n}\right]$$ Où c ($c > 2$) est une constante.

***Remarque* :** Le critère le plus utilisé est le critère AIC. Cependant *Hannan (1980)* a montré que seuls les estimations des ordres p et *q* déduits des critères *BIC* et *HQ* sont convergentes et conduisent à une sélection asymptotiquement correcte du modèle.

Nous cherchons à minimiser ces différents critères. Leurs applications nous permettent de retenir un modèle parmi les divers processus *ARMA* validés. Ainsi s'achève l'étape de validation. La dernière étape de la méthodologie de Box & Jenkins est celle de la prévision.

• **Principe de parcimonie**

Lorsqu'on veut modéliser une série chronologique, par un processus Stochastique et dans le cas où les critères d'information *AIC* et *BIC* de deux ou plusieurs modèles retenus seraient très proches ou contradictoires, nous faisons intervenir ce principe qui cherche à minimiser le

nombre de paramètre requis ; il est préférable de conserver un modèle qui est ″moins bon″, mais qui contient moins de paramètres.

II.8.3 Prévision

L'objectif de la modélisation de Box et Jenkins est la prévision de futures valeurs de la série chronologique. A l'horizon h la valeur de la prévision $\hat{X}(t+h)$ notée $\hat{X}_t(h)$ est l'espérance conditionnelle de *X(t+h)* telle que :

$$\hat{X}_t(h) = E(X_{t+h} / X_t, X_{t-1})$$

$\hat{X}_t(h)$ est la prévision (la valeur estimée de X_{t+h})

t : l'origine de la prévision.

h : l'horizon de la prévision

L'espérance conditionnelle de la prévision est définie par :

$$1 - E\{X(t-j)/X(t), X(t-1), \ldots, X(1)\} = X(t-j), \quad j \geq 0$$

$$2 - E\{X(t+j)/X(t), X(t-1), \ldots, X(1)\} = \widehat{X}_t(j), \quad j > 0$$

$$3 - E\{\varepsilon(t-j)/X(t), X(t-1), \ldots, X(1)\} = \varepsilon(t-j), \quad j \geq 0$$

$$4 - E\{\varepsilon(t+j)/X(t), X(t-1), \ldots, X(1)\} = 0, \quad j > 0$$

Nous considérons un processus ARIMA (p, d, q) défini par :

$$\phi(L)\nabla^d X_t = \theta(L)\varepsilon_t, \quad t > 0$$

$$où\ \phi(L) = \sum_{i=0}^{p} \phi_i L^i,\ \theta(L) = \sum_{i=0}^{q} \theta_i L^i \text{ avec } \theta_0 = \phi_0 = 1, \text{ et } \nabla^d = (1-L)^d$$

Nous pouvons écrire :

$$\left(1 - \sum_{i=0}^{q} \theta_i L^i\right)(1-L)^d = \left(1 - \sum_{i=0}^{q} \theta_i L^i\right) \sum_{j=0}^{d} \binom{d}{j} (-L)^{d-j} \quad **$$

$$X_t = \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \cdots + \varphi_p X_{t-p-d}$$

Où les φ_i sont obtenus à partir du développement de $**$

Pour un horizon *h*, l'équation s'écrit :

$$\widehat{X}_t(h) = E(X(t+j)/X(t), X(t-1), \ldots) = \sum_{i=1}^{p+q} \phi_i \widehat{X}_t(h) - \sum_{i=1}^{q} \theta_i \hat{\varepsilon}_t(h-i)$$

$$où\ \hat{\varepsilon}_t(h-i) = \begin{cases} 0 & \text{pour } i < h \\ \varepsilon_t(h-i) & \text{pour } i \geq h \end{cases}$$

- **L'erreur de prévision**

Soit e_{t+h} l'erreur de la prévision à l'instant t+h : $e_{t+h} = X_{t+h} - \hat{X}_{t+h}$

Nous notons que : E (ε_{t+h}) = 0, Var (ε_{t+h}) = $\left(1+\psi_1^2+\psi_2^2+\cdots+\psi_{h-1}^2\right)\sigma_\varepsilon^2$

Donc $X_{t+h} \sim N\left(\hat{X}_{t+h}, \left(1+\Psi_1^2+\Psi_2^2+...+\Psi_{h-1}^2\right)\sigma^2\right)$

D'où les intervalles de confiance de la prévision à (1- α)% donnée par :

$$X_{t+h} = \hat{X}_{t+h} \pm Z_{1-\frac{\alpha}{2}} \sigma \sqrt{\left(1+\Psi_1^2+\Psi_2^2+...+\Psi_{h-1}^2\right)};$$

où $Z_{1-\frac{\alpha}{2}}$ est le quantile d'ordre $\left(1-\frac{\alpha}{2}\right)$ de la loi normale centrée réduite. Pour α = 5% Nous avons :

$$X_{t+h} = \hat{X}_{t+h} \pm 1.96\sigma\sqrt{\left(1+\Psi_1^2+\Psi_2^2+...+\Psi_{h-1}^2\right)}$$

Les critères suivants sont utilisés pour juger la validité de la méthode de prévision :

1- ***L'erreur moyenne (Mean Error)*** **ME (ε) =** $\bar{\varepsilon} = \frac{1}{\mathbf{n}}\sum_{\mathbf{i}=1}^{\mathbf{n}} \varepsilon_{\mathbf{i}}$

2- ***La variance*** **Var (ε) =** $\frac{1}{n}\sum_{i=1}^{n}\left(\varepsilon_i - \bar{\varepsilon}\right)^2$

3- ***Le carré moyen des erreurs (Mean Square Error)*** **MSE (ε) =** $\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i^2$

Très utilisé, car il pénalise un biais éventuel dans la prévision.

4- ***Root mean square error*** **RMSE (ε) =** $\sqrt{\mathbf{MSE}(\varepsilon)} = \sqrt{\frac{1}{\mathbf{n}}\sum_{\mathbf{i}=1}^{\mathbf{n}}\varepsilon_{\mathbf{i}}^2}$

III.1 Introduction

La démarche du prévisionniste restera la même, il doit passer par l'Analyse, l'identification, l'estimation, la validation et enfin la prévision, sauf que les techniques utilisées se diffèrent d'une approche à une autre dans quelques étapes.

Nous allons découvrir une méthode de l'apprentissage supervisé. L'apprentissage est la caractéristique la plus importante des réseaux de neurones. Il est équivalent à l'estimation des paramètres du modèle neuronal qui ne sont rien d'autres que les poids synaptiques reliant les neurones entre eux, l'apprentissage ce n'est que la convergence d'un algorithme itératif (rétro propagation) vers un point stable (ou fixe). Pour tout ce qui est de la validation, l'analyse des résidus restent toujours un point commun entre les différentes approches utilisées.

III.2 Les Réseaux de Neurones et La Prévision des Séries Temporelles

Depuis la décennie 1990, les réseaux de neurones artificiels habituellement utilisés en physique appliquée ont fait leur entrée dans les sciences de gestion en tant que méthode quantitative de prévision, â coté des méthodes statistiques classiques.
Les techniques des réseaux neuronaux peuvent être classées dans le domaine de l'intelligence artificielle du fait de leur comportement très lié aux techniques d'apprentissage. A cette fin, on doit penser qu'ils peuvent s'adapter plus facilement aux techniques de prévision, où les modèles de prévisions sont réajustés de façon constante.

Les chercheurs se sont intéressés â cet outil pour les principales raisons suivantes:

- Contrairement aux méthodes statistiques, les réseaux de neurones ne nécessitent aucune hypothèse sur les variables.
- Ils sont adaptés pour traiter des problèmes complexes non structurés (des problèmes sur les quels il est impossible à priori de spécifier la forme des relations entre les variables utilisées).
- La prévision avec les réseaux de neurones est capable de «permettre aux gens possédant peu de connaissances dans le domaine de la prévision ou des réseaux de neurones de préparer dans un court espace de temps des prévisions raisonnables » [cf. HOPTROFF (l993)].
- Il n'est pas nécessaire de connaître la distribution de probabilité des variables, ce qui n'est pas le cas dans la plupart des modèles d'analyse statistique sauf s'il s'agit d'analyse non paramétrique.

- En découvrant eux-mêmes es relations entre les variables, ils sont adaptés pour traiter des problèmes non linéaires, aspect très intéressant car il n'oblige pas à s'interroger sur la forme de la fonction à estimer.
- les données incomplètes ou données bruitées ne posent pas de problèmes pour les réseaux de neurones « ils peuvent être pris en compte par l'ajout de neurones supplémentaires ».
- La robustesse du réseau, le processus itératif (apprentissage) ne s'arrête qu'après l'obtention du bon résultat sur l'échantillon de validation, donc seule l'information pertinente est intégrée (pas de bruit).
- Ils prennent en compte les variables qualitatives a travers des neurones qui reçoivent des entrée binaires.

La prévision des séries temporelles semble constituer aujourd'hui un champ d'investigation privilégié [cf. A,ZOFFI994}. Spécifiquement, dans les prévisions univariées, il s'agit d'utiliser le passé d'une variable afin d'en extraire des relations permettant de prédire sa future valeur, Le problème qui se pose est la détermination de la relation qui existe entre les deux valeurs, qui est souvent non linéaire.

Un autre point de vue, est celui de THIRIA SYLVIE3. Celle-ci considère que les réseaux de neurones, contrairement à l'approche classique qui consiste à commencer par poser un modèle en analysant le phénomène ayant produit la chronique, puis à estimer les paramètres de ce modèle à partir des données disponibles, forment une approche différente en privilégiant les données expérimentales. Ils consistent à utiliser les données et un critère à minimiser pour construire un modèle de prévision, un modèle flexible dont la complexité devra être ajustée à celle des données disponibles.

Le premier modèle neuronal utilisé pour traiter des séries temporelles est le perceptron multicouches [THACKER 1990, NARENDRA 1990, PORT 1990. JOHN 1990]. Les chercheurs qui s'intéressent aux réseaux de neurones comptent sur ses capacités de détecter les formes des relations dans la série et qui peuvent échapper aux méthodes statistiques traditionnelles.

En effet, les chercheurs s'interrogent « si nous avons là une progression ou ouverture au milieu statistique». L'écart entre «les croyants» et «les non croyants» en réseaux de neurones dans le domaine de la prévision est justifié par les différences d'approches qu'ils adoptent ; la première est l'algorithmique informatique et l'autre la statistique.

On pourrait synthétiser la différence en termes entre la statistique et les réseaux de neurones par :

Les Réseaux de neurones	La Statistique
Apprentissage	Estimation
Poids	Paramètres
Apprentissage supervisé	Régression
Apprentissage non supervisé	Estimation de densité
Réseau de neurone	Modèle
Ensemble d'apprentissage	Echantillon

Fig III.1 : Les RNA et la statistique

En ce qui concerne la prévision des séries temporelles un avantage des réseaux de neurone qui n'est pas facilement égalé dans la prévision traditionnelle est la capacité du réseau à «estomper progressivement les observations aberrante dans le reste des données des séries chronologique»

III.3 Les Procédures De Développement D'un Réseau de Neurones

Le cycle classique de développement peut être résumé en :

III.3.1 Collecte De Données

L'objectif de cette étape est de recueillir des données, à la fois pour développer le réseau de neurones, et pour le tester. Dans le cas d'application sur des données réelles, l'objectif est de rassembler un nombre de données suffisant pour constituer une base représentative des données susceptibles d'intervenir en phase d'utilisation du système neuronal.

III.3.2 Analyse Des Données

Il est souvent préférable d'effectuer une analyse des données de manière à déterminer les caractéristiques discriminantes pour détecter ou différencier ces données. Ces caractéristiques constituent l'entrée du réseau de neurones. Notons que cette étude n'est pas spécifique aux réseaux de neurones, quelque soit la méthode de détection utilisée, il est généralement nécessaire de présenter des caractéristiques représentatives. Une étude statistique sur les données peut permettre d'écarter celles qui sont aberrantes et redondantes.

III.3.3 Séparation De La Base de Données

Afin de développer une application à base de réseaux de neurones, il est nécessaire de disposer de deux bases de données : une base pour effectuer l'apprentissage, et une autre pour tester le réseau obtenu et déterminer ses performances.

Il est à noter que l'on ne doit pas entraîner le réseau complètement avec un ensemble d'entrées appartenant à la même classe, puis basculer vers une autre classe car le réseau oubliera l'entraînement original.

III.3.4 Mise en Forme des Données Pour un Réseau de Neurones

Bien que le choix des poids initiaux influe fortement sur la convergence de l'algorithme d'apprentissage, la manière de présenter les données au réseau joue aussi un rôle très important dans sa performance. Avant tout apprentissage, il est indispensable de faire subir aux bases de données un prétraitement, afin de les adapter aux entrées et aux sorties du réseau de neurones.

Ce prétraitement est la normalisation de toutes les variables d'entrées parce que « si des entrées ont des grandeurs très différente, celles qui sont petites n'ont pas d'influence sur l'apprentissage ».

III.3.5 Fixer le Nombre de Couches Cachées

Mise à part les couches d'entrées et de sorties, l'analyste doit décider le nombre de couches intermédiaires ou cachées, car il est presque impossible de spécifier une architecture qui satisfait à la résolution d'un problème quelconque, on doit construire cette architecture expérimentalement.

Le réseau est dit capable de résoudre le problème, s'il est capable d'apprendre à approximer la fonction avec une certaine précision.

La définition de l'architecture d'un réseau se fait d'une manière empirique, aucune théorie n'a pu trancher sur cette question.

III.3.6 Détermination du nombre de neurones par couches cachées

Un nombre très important de neurones permet de mieux coller aux données présentées en entrée. Mais diminue la capacité de la généralisation du réseau.

Là aussi, il n'existe pas de règles générales mais des règles empiriques. La taille de la couche doit être :

- soit égale à celle de la couche d'entrée (Wierenga et Kluytmans, 1994).
- soit égale à 75% de celle-ci (Venugopal et Baets, 1994).
- soit égale à la racine carrée du produit du nombre de neurones dans la couche d'entrée et de sortie (Shepard, 1990).

Notons que le dernier choix réduit le nombre de degré de liberté laissés au réseau, et donc la capacité d'adaptation sur l'échantillon d'apprentissage, au profit d'une plus grande stabilité (capacité de généralisation).

III.3.7 Choisir la Fonction D'activation

Souvent on utilise la fonction logistique pour le passage de la couche d'entrée à la couche cachée, le passage de cette dernière à la couche de sortie sera soit linéaire, soit logistique selon nos types de variables.

En générale, on choisit des tangentes hyperboliques pour lesquelles on peut choisir la fonction logistique à valeur entre 0 et 1.

Il faut savoir que l'utilisation des fonctions non linéaires ne permet pas au programme de trouver facilement le minimum local de la fonction coût par contre, par la fonction linéaire le minimum est vite trouvé.

III.3.8 Choisir L'apprentissage

L'apprentissage par rétro propagation nécessite la détermination du paramètre d'ajustement des poids synaptiques à chaque itération.

La détermination du critère d'arrêt est aussi cruciale dans la mesure où la convergence peut passer par des minima locaux.[12].

III.3.9 Valeurs Initiales des Poids et du Biais

Au lancement de l'apprentissage, les valeurs initiales des poids doivent être différentes de zéro pour que l'algorithme de rétro propagation puisse fonctionner, d'autre part l'utilisation de valeur élevée peut provoquer un phénomène de saturation prématurée qui contribue à diminuer la vitesse de convergence de l'apprentissage.

Ce type d'initialisation des poids n'est cependant pas facile à mettre en œuvre, c'est pour ça que les poids sont initialisés à de petites valeurs aléatoires entre [-0,5, 0,5].

Le biais qui est une entrée constante, ajouté au vecteur des entrées qu'on veut traiter, devra être traité comme autre poids. Ce biais est connecté à une unité imaginaire, et qui a toujours une sortie égale à 1. Ceci peut améliorer les propriétés de convergence du réseau.

III.3.10 Taux D'apprentissage

Le taux d'apprentissage doit être choisi avec soin, car celui-ci a un effet signifiant sur la performance du réseau. Généralement il doit être un petit nombre compris entre 0,05 et 0,25 pour assurer que le réseau atteint une solution. Une petite valeur signifie que le réseau devra faire un grand nombre d'itérations avant de converger. Il est possible de l'incrémenter durant

[12] Pour vérifier qu'un minimum est un minimum local, il faut que sa Héssienne soit positive.

l'apprentissage, quand l'erreur décroît est cela pour augmenter la vitesse de convergence, mais s'il devient trop grand, le réseau risquera de rebondir très loin du maximum.

III.3.11 Le Momentum

Pour augmenter la vitesse de convergence sans toute fois entraîner le risque d'instabilité du réseau, on utilise la technique de momentum. Quand la valeur de changement de poids est calculée, on rajoute une fraction du changement précédant, cet ajout devra tenir compte des changements des poids dans la même direction.

L'équation de changement des poids devient :

$$\theta_{ji}(t+1) = \theta_{ji}(t) + \varepsilon\delta_j x_i + \alpha\theta_{ji}(t-1)$$

En pratique, on prend α inférieur et voisin de 1.

III.3.12 Test D'arrêt de L'apprentissage

Théoriquement, on cherche à arrêter l'apprentissage dés que le minimum de l'erreur est atteint, ce qui correspond à un gradient nul. En pratique, ce minimum n'atteint jamais le zéro. La méthode consiste à arrêter l'apprentissage dés que : E_k est inférieure à un seuil E_{min}. Où E_{min} ne doit pas être petit (un grand nombre d'itération).

III.3.13 Problème de Minimaux Locaux

Il est possible que l'algorithme d'apprentissage, converge vers un minimum local. Dans ce cas, le processus d'apprentissage s'arrête et l'erreur sur les sorties du réseau peut être grande. (On rencontre rarement ce problème en pratique).

Si le réseau s'arrête d'apprendre avant d'atteindre une solution acceptable, le changement dans le nombre de cellules des couches cachées ou dans les paramètres d'apprentissage résoudra le problème.

III.3.14 Validation

Une fois le réseau de neurones entraîné (après apprentissage), il est nécessaire de le tester sur une base de données différente de celle utilisée pour l'apprentissage ou par la validation croisée. Ce test permet à la fois d'apprécier les performances du système neuronale, et de détecter le type de données qui pose problème, si les performances ne sont pas satisfaisantes, il faudra soit modifier l'architecture du réseau, soit modifier la base d'apprentissage (caractéristiques discriminantes, ou représentativité des données de chaque classe).

III.4 Estimation des Paramètres (d'un Modèle Neuronal)

III.4.1 Valeurs étranges et Prétraitement des Données

Les valeurs étranges obligent l'algorithme d'estimation à fournir une sortie incohérente. Un algorithme d'estimation au sens des moindre carrée donnera alors un poids encore plus important à l'erreur faite par le modèle en ce point.

Le but d'un prétraitement est de transformer les paramètres à travers une normalisation. De façon à pouvoir être utilisées convenablement pendant la phase d'estimation, soit de former une combinaison linéaire ou non linéaire a des variables originaires pour créer des nouvelles variables. Pour les séries temporelles, on pourrait même faire une étude des probabilités de répétition des schémas.

III.4.2 Sélection de Variables

La sélection de variables consiste à identifier les séries explicatives du processus que nous voulons modéliser, et aussi pour chacune de ces séries les retards dans le temps nécessaires. La sélection de variables représente une des difficultés majeures de l'estimation d'un modèle.

II faudra chercher les variables nécessaires afin d'expliquer convenablement le processus à modéliser, mais sans augmenter artificiellement la complexité du modèle en diminuant sa capacité de généraliser dans les prévisions futures. On distingue deux types de séries explicatives qui impliquent l'utilisation de modèles différents : le passé de la série à prévoir (appelée auto régressive), le passé des résidus du modèle (appelée moyenne mobile).

III.4.3 Normalisation Des Entrées du Réseau

Bien que le choix des poids initiaux influe fortement sur la convergence de l'algorithme d'apprentissage, la manière de présenter les données au réseau joue aussi un rôle très important dans sa performance.

Avant tout apprentissage, il est indispensable de normaliser toutes les variables d'entrée, **«**Si des entrées ont des grandeurs très différentes, celles qui sont petites n'ont pas d'influence sur l'apprentissage **»**

Il existe plusieurs méthodes pour la normalisation sur lesquelles le choix s'effectue.

Nous citons les deux méthodes les plus courantes :

- la première est exprimée par l'équation suivante :

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

avec x valeur de la série temporelle à normaliser, X_{min} valeur minimale de l'ensemble de données utilisé pour la normalisation, et X_{max} est la valeur maximale.

- La deuxième méthode de normalisation des données est exprimée par l'équation suivante :

$$X' = \frac{X - \overline{X}}{\sigma}$$

avec $\overline{X}$ est la valeur moyenne de l'ensemble des données à utiliser pour la prévision, et σ est l'écart type.

III.4.4 Traitement Logarithmique :

Pour de nombreuses séries temporelles, il est conseillé d'effectuer un prétraitement logarithmique, il suffit d'associer à chaque valeur de la série temporelle X_t une valeur X_t' tel que :

$$X_t' = Ln\ X_t$$

III.5 Problèmes d'estimation Des Paramètres

III.5.1 Problème des Minima Locaux

Dans le cas d'un modèle linéaire par rapport a ses paramètres. La fonction de coût minimisée lors de I 'apprentissage ne présente qu'un seul minimum. Par contre dans le modèle non linéaire, l'algorithme peut se trouver bloquer un minimum local et ne peut alors pas atteindre le minimum global. (Ceci est du au tan que les fonctions complexes non linéaires qu'on donne aux réseaux de neurones sont difficiles à optimiser au sens du critère MSE.

III.5.2 Problème de sur Apprentissage

Le problème du piège du sur ajustement s'applique au système de réseau de neurones, tout comme il s'applique aux autres méthodes statistiques. Dans les modèles statistiques traditionnels de régression, le problème du piège du sur ajustement se produit assez souvent et surtout avec des données de nature hautement non linéaire en cas ou nous introduisons trop de variables dans la régression.

Cette mauvaise adéquation entre le modèle et le phénomène à prévoir induit une augmentation de la variance de la perturbation aléatoire associée au phénomène.

Par conséquent, un tel modèle ne conduit pas toujours à faire de bonnes déductions concernant de futures données. Après un certain nombre d'itérations, on remarquera (pour la base d'apprentissage et la base du test) la croissance de la fonction de la somme des erreurs quadratique.

III.6 L'apprentissage

La notion d'apprentissage est claire et intuitive pour les humains ou les animaux, c'est une procédure cognitive qui doit faire en sorte que l'individu réalise de manière autonome une tâche donnée, typiquement cette procédure s'effectue à partir d'exemples, ainsi pour apprendre à lire à un enfant, on lui présente des exemples de lettres, de chiffres écrits avec des écritures et des formes différentes, à la fin de l'apprentissage, l'enfant pourra lire tous les chiffres et les lettres qu'il est susceptible de voir.

L'apprentissage numérique poursuit exactement le même objectif, il s'agit de faire en sorte, à l'aide d'une procédure numérique programmée, et exécutée sur un ordinateur, d'inférer un modèle d'un processus que l'on peut observer, et sur lequel on peut effectuer des mesures, c'est-à-dire, un ensemble d'équations qui décrivent le processus observé, et qui permettent de faire des prédictions concernant le comportement de celui-ci.

A supposer que notre processus puisse être décrit avec précision d'une (ou plusieurs) fonctions à paramètres, on ajuste ces derniers, pour que cette fonction soit la plus proche aux données, l'apprentissage est donc un algorithme permettant l'ajustement des paramètres, ou des poids des connexions du réseau, il nécessite des exemples de données désignés, appelés échantillon d'apprentissage.

Notons par ailleurs, que l'apprentissage est semblable à l'estimation pour les méthodes statistiques de modélisation, il emprunte aux statistiques un grand nombre de techniques, bien que la philosophie de l'apprentissage soit différente de celle des statistiques. Car dans le contexte d'apprentissage, on n'a aucune idée à priori sur le modèle, on choisit une forme d'équation aussi générale que possible, et l'on ajuste ses paramètres de manière à lui conférer la meilleure capacité de généralisation possible. On s'intéresse donc à la capacité de généralisation du modèle, et on cherche à estimer la qualité des prédictions que le modèle peut effectuer dans des situations qu'il n'a pas rencontrées dans l'apprentissage. Ainsi, l'objectif de l'apprentissage est différent de celui de l'estimation en statistique, mais les techniques développées par les statisticiens sont précieuses pour l'apprentissage.

On appelle **« apprentissage des réseaux de neurones, la procédure qui consiste à estimer les paramètres des neurones du réseau, à fin que celui-ci remplisse au mieux la tâche qui lui est affectée »**, donc 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. L'apprentissage neuronal fait appel à des exemples de comportements, il représente la propriété la plus intéressante des réseaux neuronaux, c'est l'indicateur de l'intelligence.

Dans le cas d'un problème de régression, il s'agit d'approcher une fonction continue, Cet apprentissage s'effectue grâce à la minimisation d'une fonction, appelée *fonction de coût*, calculée à partir des exemples de la base d'apprentissage et de la sortie du réseau de neurones , cette fonction détermine l'objectif à atteindre.
L'apprentissage est dit **supervisé** lorsque I' information précise sur la sortie désirée au réseau est fournie. Si aucune sortie désirée n'est donnée au réseau. L'apprentissage sera **non supervisé** et le réseau est censé produire une réponse correcte.

III.6.1 L'apprentissage Supervisé

Après initialisation des poids du réseau de façon aléatoire, des valeurs petites généralement comprises entre [-0.1, 0.1]. On présente au réseau les exemples constitués de couples de valeurs [entrée, sortie désirée] appelés **la base d'apprentissage**.

Pour un vecteur d'input donné, le réseau produit un vecteur d'output dont la valeur finale des composantes dépend de l'architecture du réseau, et de la valeur courante des poids.

Le vecteur d'output est comparé au vecteur de sortie désirée, ce qui permet le calcul un écart appelé l'erreur de prédiction. La fonction d'erreur la plus utilisée est la moyenne des carrés des écarts entre l'output et la sortie désirée **« MSE »,** cette erreur permet de modifier les poids du réseau, l'erreur totale commise par le système est ensuite rétro propagée de la couche de sortie à la couche d'entrée, dans le but d'ajuster les poids en fonction de leurs contribution à cette erreur (back propagation learning rule). Les poids sont ainsi corrigés, ce qui permet d'initialiser un nouveau calcul.

Dans l'objectif de minimiser la fonction d'erreur que l'on choisira à priori, le processus d'apprentissage se poursuit de manière itérative et convergente jusqu'à ce que le réseau se stabilise c'est-à-dire les poids ne changent presque plus à chaque itération, ou bien jusqu'à l'intervention de l'utilisateur quand l'erreur atteint le seuil fixé. Ce type d'apprentissage est surtout utilisé pour l'approximation de la fonction de régression et la classification.

III.6.2 L'apprentissage Non Supervisé

L'apprentissage est qualifié de non supervisé lorsque seules les valeurs d'entrée sont disponibles, le réseau est censé s'organiser par lui-même pour produire la réponse correcte. Donc ce mode d'apprentissage est moins intuitif, car lorsqu'on ne sait pas à priori déterminer ponctuellement, si la sortie est valable ou non, l'apprentissage repose alors sur un critère interne de conformité du comportement du réseau par rapport à des spécifications générales et non sur des observations externes, où il n'y a pas de maître pour contrôler de l'extérieur le déroulement de l'apprentissage, c'est pourquoi on l'appelle auto apprentissage, le réseau va essayer de s'adapter aux régularités statistiques des données d'entrées, il va donc automatiquement coder des classes dans sa représentation interne. Une variété de schéma existe pour ces réseaux tels que les réseaux de kohonen (1982) hopfield (1984).

III.6.3 L'apprentissage Forcé

L'apprentissage forcé est un cas particulier de l'apprentissage supervisé, mais au lieu d'employer un superviseur pour donner des sorties désirées, nous utilisons un critère pour évaluer seulement la qualité de la sortie du réseau neurologique correspondant à une entrée donnée.

III.6.4 Validation Croisée

La validation croisée consiste à extraire de la base d'apprentissage, une portion des observations qui servent non pas à l'apprentissage, mais à l'évaluation après apprentissage de l'erreur commise par le réseau.

La démarche est totalement similaire à celle adoptée en modélisation, elle est encore plus nécessaire dans la mesure où la complexité interne et le caractère encore fortement empirique des réseaux de neurones ne permettent pas aujourd'hui d'offrir un cadre théorique permettant d'extrapoler un apprentissage sur une probabilité.

III.6.5 Sur Apprentissage (Apprentissage Par Cœur)

Imaginons un protocole itératif, au cours duquel on améliore l'apprentissage au sens où l'erreur commise sur la base d'apprentissage est réduite à chaque itération. Si l'on procède à chaque itération à une validation croisée sur une base d'essai non utilisée par l'apprentissage, il peut arriver un moment où l'amélioration de l'erreur sur la base d'apprentissage conduit à une augmentation de l'erreur sur la base d'essai.

Ce phénomène est bien connu en modélisation, il correspond schématiquement au cas où l'on introduit trop de variables explicatives dans un modèle, il arrive alors que l'on n'explique plus le comportement global du système, mais les aléas spécifiques aux données de l'apprentissage (on modélise les résidus). On dit dans ce cas qu'il y a sur apprentissage, le modèle perd sa capacité de généralisation.

III.7 Les Algorithmes Des Différents Modèles

III.7.1 Loi d'apprentissage du perceptron

Soient ***p*** et ***t*** les vecteurs d'entrée et sortie cible utilisés pour l'apprentissage du perceptron et ***a*** est la réponse du perceptron. L'évolution de la valeur des poids W et des biais b du perceptron vont varier, à chaque fois (nombre de epoch) que les vecteurs d'entrée sont présentés au perceptron, selon la règle :

$$\Delta W = (t-a)\,p^t = ep^t \text{ et } \Delta b = (t-a)(1) = e,$$

donc on aura

$$\begin{Bmatrix} \mathrm{W}^{\,\mathrm{new}} = W^{\,\mathrm{old}} + ep^T \\ b^{new} = b^{old} + e \end{Bmatrix}$$

III.7.2 La Rétro Propagation ou Algorithme d'apprentissage de « Back Propagation »

La rétro propagation a été crée en généralisant la loi d'apprentissage de Widrow-Hoff à des réseaux de neurones multicouches constitués de fonctions de transfert différentiables. Les vecteurs d'entrées et les vecteurs cibles correspondant sont utilisés pour apprendre le réseau.

Les réseaux de neurones constitués de biais et de fonctions de transfert « sigmoïdale » et une couche de sortie constituée de fonctions de transfert linéaires sont capables d'approximer n'importe qu'elle fonction possédant un nombre fini de discontinuité.

La règle delta impose toujours $\Delta W = -\alpha \dfrac{\partial F}{\partial W}$.

La difficulté réside toujours dans le calcul de $\dfrac{\partial F}{\partial W}$. La rétro propagation standard est un algorithme de descente du gradient, comme la loi d'apprentissage de Widrow-Hoff, dans lequel les poids des réseaux sont ajustés dans le sens du gradient négatif de la fonction coût. Le terme de rétro propagation veut dire que le gradient est calculé pour des réseaux multicouches non linéaires. De nombreuses techniques existent, plus ou moins rapides, performantes et gourmandes en mémoire vive. Il apparaît que la technique de Levenberg-Marquardt est un algorithme très rapide.

L'algorithme de rétro-propagation a été développé en particulier par Rumelhart et Parkenet le Cun en 1985 [YOU 99]. Cet algorithme repose sur la minimisation de l'erreur quadratique entre les sorties calculées et celles souhaitées.

Le terme rétro-propagation du gradient provient du fait que l'erreur calculée en sortie est transmise en sens inverse vers l'entrée.

On considère un réseau de neurones recevant p vecteurs à N composantes. On note le k^{iéme} exemple de la base d'apprentissage : X=.$[x_1, x_2, x_3, ..., x_N]$. L'entrée de chaque neurone de la couche cachée est calculée par :

$$A_j = \sum_{i=1}^{N} \theta_{ji} x_i + \theta_j \text{ ..... (1) .}$$

Avec A_j : L'entrée d'un élément j de la couche cachée.

θ_{ji} : Poids de connexion entre le neurone j de la couche cachée et l'entrée i.

x_i : La sortie i de la couche d'entrée.

θ_j : est le terme du biais « seuil ».

La sortie du neurone j de la couche cachée est calculée par :

$$x_j = f(A_j) = \frac{1}{1+e^{-A_j}} \text{ ..... (2)}$$

tel que f est sigmoïde. L'entrée et la sortie de chaque élément de la couche de sortie s'effectuent de la même manière que celle du neurone de la couche cachée.

$$A_l = \sum_{j=1}^{n_s} \theta_{lj} x_j + \theta_l \text{ ..... (3)}$$

n_s : est le nombre de neurones dans la couche de sortie.

$$x_l = f(A_l) = \frac{1}{1+e^{-A_l}} \text{ ..... (4)}$$

Avec A_l : l'entrée d'un élément l de la couche de sortie.

x_l : La sortie du neurone l de la couche de sortie, elle représente le résultat obtenu par le réseau du k^{iéme} exemple.

III.7.2.1 Mise à jour des poids de la couche de sortie

Partant des poids aléatoires on présente au réseau un nombre d'exemples, puis on compare la sortie obtenue du réseau à celle désirée. L'écart calculé est minimisé en corrigeant les poids initiaux.

L'erreur commise sur le l^éme nœud de sortie est donnée par :

$e_l = d_l - x_l$

Par conséquent l'erreur totale (pour tous les nœuds) est :

$$E_k = \frac{1}{2}\sum_{l=1}^{n_s}(d_l - x_l)^2 \text{ ..... (5)}$$

Avec d_l : La sortie désirée d'un élément l .

x_l : La sortie obtenue à l'élément l de la couche de sortie.

Etant donné que le but de l'apprentissage est de minimiser cette erreur, on utilise la méthode de la descente du gradient. On corrige les poids tels que :

$$\Delta\theta_{lj} = -\varepsilon \frac{\partial E_k}{\partial \theta_{lj}} \text{ ..... (6) (la règle de delta).}$$

ε : Coefficient d'apprentissage, appartenant à $]0,1[$.

La méthode du gradient consiste à calculer θ_{lj} après avoir présenté au réseau les p exemples :

$$E = \sum_{k=1}^{p} E_k$$

$$\Delta\theta_{lj} = -\varepsilon\sum_{k=1}^{p}\frac{\partial E_k}{\partial \theta_{lj}} \text{ ..... (7)}$$

Le réseau effectue le calcul $\frac{\partial E_k}{\partial \theta_{lj}}$ de façon local de la manière suivante :

D'après les règles de dérivation des fonctions composées on écrit :

$$\frac{\partial E_k}{\partial \theta_{lj}} = \frac{\partial E_k}{\partial x_l}\frac{\partial x_l}{\partial A_l}\frac{\partial A_l}{\partial \theta_{lj}} \quad \text{..... (8)}$$

D'après l'équation (3)

$$\frac{\partial A_l}{\partial \theta_{lj}} = x_j \text{ ..... (9)}$$

On pose : $\delta_l = -\frac{\partial E_k}{\partial x_l}\frac{\partial x_l}{\partial A_l}$ ..... (10)

δ_l : le signal d'erreur de l'élément l , c'est-à-dire la contribution de l'entrée A_l de l'unité l à l'erreur quadratique constatée à la sortie.

$$\frac{\partial E_k}{\partial \theta_{lj}} = -\delta_l x_j \ldots\ldots (11)$$

La correction du poids θ_{lj} de la connexion reliant j à l'unité l est donnée par :

$$\Delta\theta_{lj} = \varepsilon\delta_l x_j \ldots\ldots (12)$$

Ainsi, à l'itération (t+1) la règle de modification des poids de la couche de sortie est donnée par :

$$\theta_{lj}(t+1) = \theta_{lj}(t) + \Delta\theta_{lj}$$

III.7.2.2 Mise à jour des poids des couches cachées

Appliquons ensuite la formule (7) pour l'ensemble des exemples. Le signal δ_l sera calculé par le principe du rétro propagation. Si on considère une unité de sortie l , on aura :

$$\frac{\partial E_k}{\partial x_l} = -(d_l - x_l) \ldots..(13)$$

Selon la fonction de transfert les unités du réseau (4) s'écrit :

$$\frac{\partial x_l}{\partial A_l} = f'(A_l) = f(A_l)\left[1 - f(A_l)\right] \ldots..(14)$$

En substituant (13) et (14) dans l'équation (10), le terme d'erreur devient :

$$\delta_l = f'(A_l)(d_l - x_l) = x_l(1 - x_l)(d_l - x_l) \ldots.. (15)$$

Si la couche de sortie contient n_s éléments, le calcul de l'erreur quadratique E_k relative à l'exemple k, dépend automatiquement de l'ensemble des unités l , ces dernières dépendent des éléments j de la couche cachée. Alors le signal d'erreur de l'unité j est :

$$\delta_j = -\frac{\partial E_k}{\partial x_j}\frac{\partial x_j}{\partial A_j} \ldots..(16)$$

Pour calculer $\frac{\partial E_k}{\partial x_j}$, on introduit les dérivées partielles $\frac{\partial E_k}{\partial x_l}$:

$$\frac{\partial E_k}{\partial x_j} = \sum_{n_s}\frac{\partial E_k}{\partial x_l}\frac{\partial x_l}{\partial x_j} \ldots..(17)$$

$$\frac{\partial E_k}{\partial x_j} = \sum_{n_s} \frac{\partial E_k}{\partial x_l} \frac{\partial x_l}{\partial A_l} \frac{\partial A_l}{\partial x_j} \text{ ..... (18)}$$

$$\text{donc } \frac{\partial E_k}{\partial x_j} = \sum_{n_s} -\delta_l \theta_{lj} \text{ .....(19)}$$

Le signal d'erreur à un élément j de la couche cachée est donc d'après les équations (14) et (16) est :

$$\delta_j = f'(A_J) \sum_{n_l} \delta_l \theta_{lj} \text{ .....(20)}$$

La formule (20) nous permet de calculer tous les termes d'erreurs des éléments de la couche cachée, à partir de ceux de la couche suivante (de la sortie). Le calcul de ce signal d'erreur se fait par couches successives en partant des neurones de sortie vers les entrées.

La correction du poids θ_{ji} de la connexion reliant i à l'unité j est donnée par :

$$\Delta \theta_{ji} = \varepsilon \delta_l x_j \text{ ..... (12)}$$

donc, à l'itération (t+1) la règle de modification des poids de la couche cachée est donnée par :

$$\theta_{ji}(t+1) = \theta_{ji}(t) + \Delta\theta_{ji} \text{ ..... (13)}$$

Ainsi, le choix des poids initiaux demande certaines précautions, il a une grande importance dans l'apprentissage et influe sur la convergence. Le moyen le plus simple consiste à prendre des poids initiaux de manière aléatoire, en leur donnant des valeurs suffisamment faibles pour éviter les problèmes de saturation, il ne faut surtout pas les prendre tous égaux, car cela entraînera une rétro propagation d'un terme d'erreur identique, et par conséquent un blocage de l'apprentissage.

III.7.2.3 Résumé de l'algorithme de rétro-propagation

1. Appliquer un vecteur d'entrée x_p aux nœuds d'entrées puis initialiser les poids du réseau ;

2. Exécuter l'échantillon d'apprentissage à travers le réseau ;

3. Calculer les termes d'erreur de signal de la couche de sortie et les couches cachées ;

4. Mise à jour les poids de la couche de sortie et couches cachées en utilisant;

5. Répéter ce processus jusqu'à ce que l'erreur E_k devienne acceptable (aller à 2).

III.7.2.4 Considérations Pratiques

- Les poids du réseau doivent être initialisés à de petites valeurs aléatoires.

- La valeur du taux d'apprentissage ε a un effet significatif sur les performances du réseau, si ce taux est petit l'algorithme converge lentement, par contre s'il est grand l'algorithme risque de générer des oscillations.
- Généralement, ε doit être compris entre 0 et 1 pour assurer la convergence de l'algorithme vers une solution optimale.
- Il n'existe pas de règles permettant de déterminer le nombre de couches cachées dans un réseau donné ni le nombre de neurones dans chacune d'elles.
- Théoriquement, l'algorithme ne doit se terminer dès que le minimum de l'erreur commise par le réseau sera atteint, correspondant à un gradient nul, ce qui n'est jamais rencontré en pratique. C'est pourquoi un seuil est fixé à priori afin d'arrêter l'apprentissage.

III.7.2.5 Accélération de L'algorithme Avec Le Momentum

La convergence du réseau par rétro-propagation est un problème crucial car il requiert de nombreuses itérations. Pour pallier à ce problème, un paramètre est souvent rajouté pour accélérer la convergence. Ce paramètre est appelé « le momentum ».

Le momentum est un moyen efficace pour accélérer l'apprentissage et aussi pour pouvoir sortir des minimums locaux.

La règle de mise à jour des poids devient alors :

$$\theta_{ji}(t+1) = \theta_{ji}(t) + \Delta\theta_{ji} + \Omega\left(\theta_{ji}(t) - \theta_{ji}(t-1)\right)$$

Ω : est la constante du momentum.

Conclusion :

La phase d'estimation du modèle neuronal présentée dans ce chapitre est celle de l'algorithme de rétropropagation du gradient, cet algorithme est utilisé pour déterminer l'ensemble des paramètres du modèle en minimisant la fonction d'erreur, l'écart entre les sorties du réseau et les sorties désirées.

Il existe plusieurs méthodes d'estimation dont l'algorithme de rétropropagation n'en est qu'un exemple, mais le calcul du gradient représente l'outil d'optimisation neuronale le plus utilisé, pour sa facilité d'utilisation et son efficacité pratique.

Les réseaux de neurones formels, tels que nous les avons définis, possèdent une propriété remarquable qui est à l'origine de leur intérêt pratique dans des domaines très divers : ce sont des approximations universelles parcimonieuses.

La propriété d'approximation peut être énoncée de la manière suivante : toute fonction bornée suffisamment régulière peut être approchée avec une précision arbitraire, dans un domaine fini de l’espace de ses variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini, possédant tous la même fonction d’activation, et un neurone de sortie linéaire, cette propriété de parcimonie est précieuse dans les applications industrielles.

Modélisation de la série Produits Pétroliers PPT

Modèle linéaire (Box et Jenkins)

Identification :

Nous considérons la série des valeurs mensuelles du trafic des produits pétroliers au port d'Alger, composées en grande partie des débarquements de carburants et d'hydrocarbures gazeux sur une certaine période de temps.

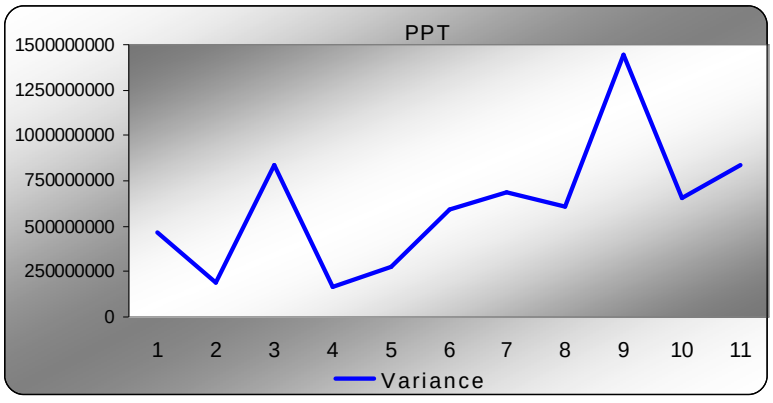
Les données de la série Produits Pétroliers (notée : **PPT**) s'étalent sur une période de onze ans, les observations sont mensuelles de janvier 1996 à décembre 2006. L'unité de mesure est la tonne.

I- Analyse graphique

Graphe de la moyenne et la variance de la série brute PPT

D'après les deux graphes ci-dessous, on remarque que la moyenne et la variance varient au cours du temps, donc on peut appréhender la non stationnarité de cette série.

Pour vérifier ceci, on va appliquer des tests statistiques juste après la présentation des corrélogrammes simple et partiel de la série brute.



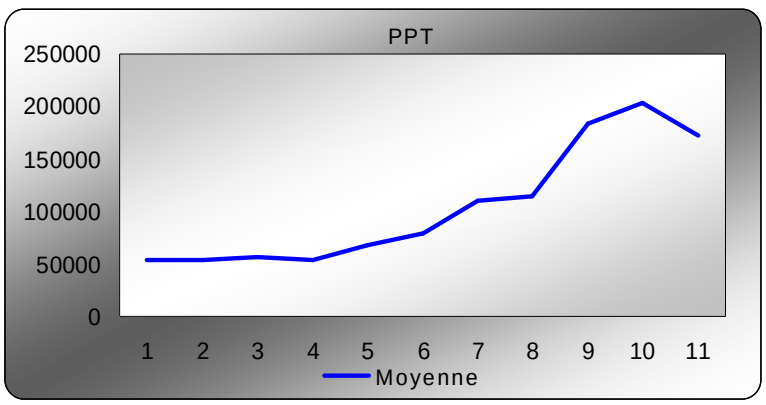
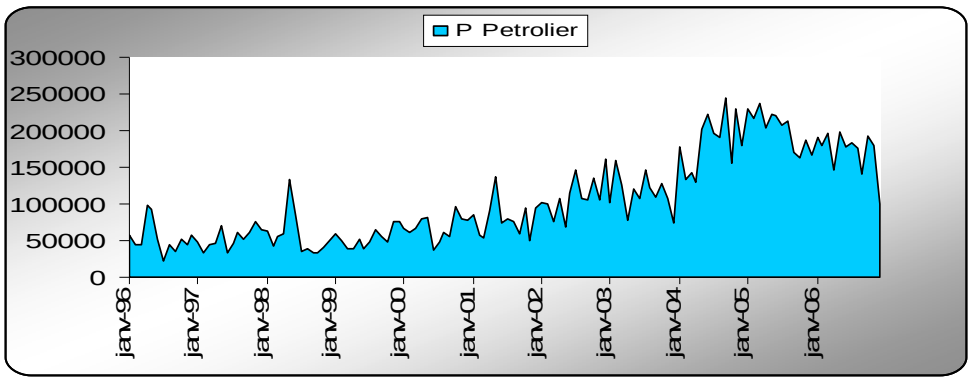


Diagramme séquentiel de la série brute PPT

On voit clairement sur le graphique de la série brute que ce processus est non stationnaire et cela provient tout naturellement la présence d'une tendance haussière.

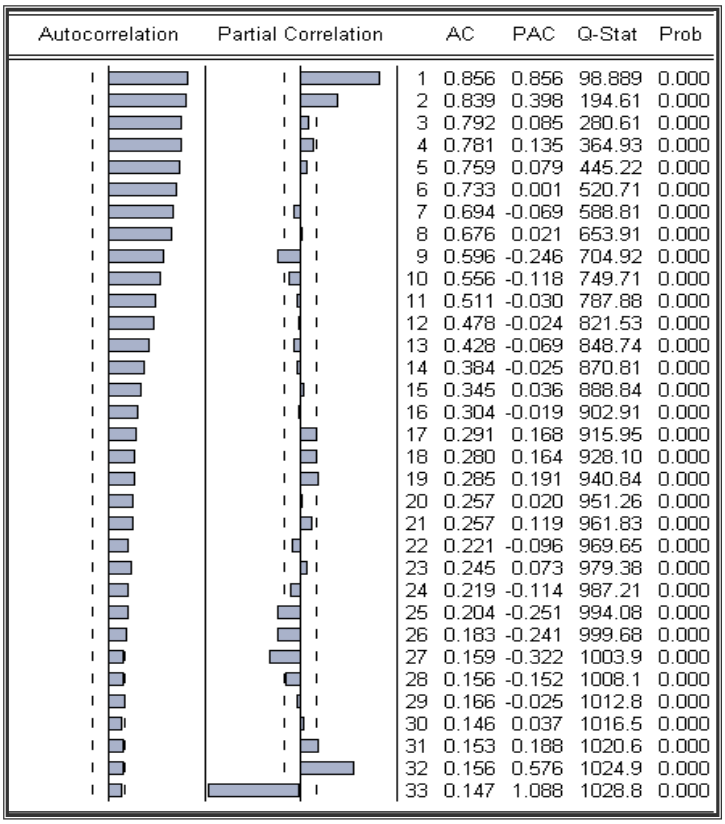


La représentation graphique fait ressortir une tendance qu'il faut confirmer ou infirmer à l'aide du test de l'analyse de variance et de Dickey-Fuller respectivement.

II -Analyse analytique

Corrélogramme de la série PPT

On remarque que le Corrélogramme simple présente une alternance entre une décroissance linéaire et un état constant, et le Corrélogramme partiel fait apparaître des pics significatifs aux retards « 1, 2, 9, 17, 18, 19… », Cela peut exprimer une non stationnarité due à l'existence d'une tendance et /ou d'une saisonnalité.



	AC	PAC	Q-Stat	Prob
1	0.856	0.856	98.889	0.000
2	0.839	0.398	194.61	0.000
3	0.792	0.085	280.61	0.000
4	0.781	0.135	364.93	0.000
5	0.759	0.079	445.22	0.000
6	0.733	0.001	520.71	0.000
7	0.694	-0.069	588.81	0.000
8	0.676	0.021	653.91	0.000
9	0.596	-0.246	704.92	0.000
10	0.556	-0.118	749.71	0.000
11	0.511	-0.030	787.88	0.000
12	0.478	-0.024	821.53	0.000
13	0.428	-0.069	848.74	0.000
14	0.384	-0.025	870.81	0.000
15	0.345	0.036	888.84	0.000
16	0.304	-0.019	902.91	0.000
17	0.291	0.168	915.95	0.000
18	0.280	0.164	928.10	0.000
19	0.285	0.191	940.84	0.000
20	0.257	0.020	951.26	0.000
21	0.257	0.119	961.83	0.000
22	0.221	-0.096	969.65	0.000
23	0.245	0.073	979.38	0.000
24	0.219	-0.114	987.21	0.000
25	0.204	-0.251	994.08	0.000
26	0.183	-0.241	999.68	0.000
27	0.159	-0.322	1003.9	0.000
28	0.156	-0.152	1008.1	0.000
29	0.166	-0.025	1012.8	0.000
30	0.146	0.037	1016.5	0.000
31	0.153	0.188	1020.6	0.000
32	0.156	0.576	1024.9	0.000
33	0.147	1.088	1028.8	0.000

Test de FISHER pour la série PPT

Pour détecter la saisonnalité de la série on a fait appel à l'analyse de la variance à deux facteurs sans répétition.

Table de l'ANOVA

Source des variations	Somme des carrés	DDL	Moyenne des carrés	F_c	Prob	Valeur critique pour F
Lignes	7307536571	11	664321506	**1,0906**	0,37549	**1,8767**
Colonnes	3,9365E+11	10	3,9365E+10	64,6251	2,1894E-41	1,9178
Erreur	6,7004E+10	110	609129884			

Règle de décision

Test d'influence du facteur Ligne, la période (mois : H_0 = pas d'influence)

Calcul de la statistique de Fisher $F_c = \frac{V_p}{V_R}$ que l'on compare à la valeur théorique lue dans la table.

$F^{\alpha}_{v_1;v_2}$ à $v_1 = p\text{-}1$ et $v_2 = (N\text{-}1)(p\text{-}1)$ degrés de liberté.

$F_c = 1{,}0906 < F^{\alpha}_{V_1V_2} = 1{,}8767$, donc on accepte l'hypothèse nulle, la série n'est pas saisonnière.

Test d'influence du facteur colonne, la tendance (H_0=pas d'influence du facteur Année)

$F_c' = \frac{V_S}{V_R}$ que l'on compare à la valeur théorique $F^{\alpha}_{v_3;v_2}$ à $v_3 = N\text{-}1$ et $v_2 = (N\text{-}1)(p\text{-}1)$ degrés de liberté. $F_c = 64,6251 > F^{\alpha}_{V_1V_2} = 1,9178$; donc on rejette l'hypothèse nulle la série est peut être affectée d'une tendance.

Désignation

V_p : La variance de la période, V_R : la variance résiduelle, V_S : la variance de la semaine, N: nombre de semaines, p : le nombre d'observations (périodicité) dans l'année (mois p=12).

Test de Dickey- Fuller Augmenté

On procède à l'estimation par la méthode des moindres carrés des trois modèles [3], [2] et [1] de Dickey-Fuller sur la série **PPT**.

Remarque

On choisit le retard (d=3) qui minimise les critères d'informations d'Akaike et Schwarz.

Etape [1]

Modèle [3]

$$\Delta PPT_t = \phi PPT_{t-1} + c + \beta t + \sum_{j=1}^{d} \phi_j \Delta PPT_{t-j} + \varepsilon_t .$$

Avec ε_t est un processus stationnaire.

Nous testons alors la présence d'une tendance dans le processus en testant la nullité du coefficient de la tendance β. Le résultat de l'affichage pour la série **PPT** est donné dans la table suivante :

Variable	Coefficient	Std. Error	t-Statistic	Prob.
PPT(-1)	-0.247784	0.092244	-2.686190	0.0082
D(PPT(-1))	-0.490627	0.110345	-4.446290	0.0000
D(PPT(-2))	-0.257433	0.111199	-2.315074	0.0223
D(PPT(-3))	-0.133779	0.092180	-1.451277	0.1493
C	4155.020	4972.810	0.835548	0.4050
@TREND(1996M01)	340.0369	140.7325	2.416194	0.0172

De la table ci-dessus nous constatons que la tendance n'est pas significativement différente de zéro puisque sa t-statistique **(2.416)** est inférieure aux valeurs critiques **3.53, 2.79 et 2.73** tabulées par Dickey-Fuller respectivement aux seuils **1%, 5% et 10%**. Nous sommes donc par la suite passés à l'étape [2].

Etape [2]

Modèle [2]

$$\Delta PPT_t = \phi PPT_{t-1} + \mathbf{c} + \sum_{j=1}^{d} \phi_j \Delta PPT_{t-j} + \varepsilon_t .$$

Où ε_t est un processus stationnaire.

Nous testons la présence d'une constante dans le processus en testant la nullité du coefficient de la constante. Le résultat de l'affichage pour la série **PPT** est donné dans la table suivante:

Variable	Coefficient	Std. Error	t-Statistic	Prob.
PPT(-1)	-0.048457	0.042074	-1.151700	0.2517
D(PPT(-1))	-0.633223	0.095053	-6.661823	0.0000
D(PPT(-2))	-0.353142	0.105928	-3.333799	0.0011
D(PPT(-3))	-0.186480	0.091307	-2.042352	0.0433
C	6337.260	4985.349	1.271177	0.2061

La constante n'est pas significativement différente de zéro puisque sa t-statistique **(1,271)** est inférieure aux valeurs tabulées par Dickey-Fuller aux seuils **1%, 5% et 10%** qui sont respectivement égales à **3.78, 3.11 et 2.38,** On passe alors à l'étape [3].

Etape [3]

Modèle [1]

$$\Delta PPT_t = \phi PPT_{t-1} + \sum_{j=1}^{d} \phi_j \Delta PPT_{t-j} + \varepsilon_t .$$

Où ε_t est un processus stationnaire.

Nous testons alors la présence d'une racine unitaire dans le processus en testant la nullité du paramètre ϕ à l'aide d'une statistique de Student, où $\hat{\phi}$ désigne l'estimateur des moindres carrés ordinaires (MCO). Le résultat de l'affichage pour la série **PPT** est donné dans la table suivante:

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-0.073018	0.6565
Test critical values:	1% level	-2.583153	
	5% level	-1.943344	
	10% level	-1.615062	

Nous procédons au test de racine unitaire, la valeur estimée de la statistique ADF est égale à (**-0,0730)** (voir la table ci-dessus). Cette valeur est supérieure aux valeurs critiques **-2, 5831, -1,9433 et -1,6150** aux seuils **1%, 5% et 10%.** Par conséquent, nous acceptons l'hypothèse nulle de racine unitaire tel que H_0: **(**il existe une racine unitaire ϕ=0**).**

Interprétation des résultats du test de Dickey-Fuller

D'après les résultats précédents nous concluons que la série **PPT** est non stationnaire de type **DS,** un des moyens pour la rendre stationnaire est de la différencier.

Soit la série : $DPPT_t = PPT_t - PPT_{t-1}$

Test de Dickey- Fuller Augmenté (DPPT)

Modèle [3]

$$\Delta DPPT_t = \phi DPPT_{t-1} + c + \beta t + \sum_{j=1}^{d} \phi_j \Delta DPPT_{t-j} + \varepsilon_t .$$

Avec ε_t est un processus stationnaire

Le résultat de l'affichage pour la série **DPPT** est donné dans la table suivante :

Variable	Coefficient	Std. Error	t-Statistic	Prob.
DPPT(-1)	-2.628183	0.304515	-8.630710	0.0000
D(DPPT(-1))	0.909340	0.251405	3.617033	0.0004
D(DPPT(-2))	0.466178	0.177528	2.625944	0.0098
D(DPPT(-3))	0.161181	0.092340	1.745521	0.0834
C	454.0910	4947.864	0.091775	0.9270
@TREND(1996M02)	14.49692	64.81767	0.223657	0.8234

De la table ci-dessus nous constatons que la tendance n'est pas significativement différente zéro puisque sa t-statistic **(0.2236)** est inférieure aux valeurs critiques **3.53, 2.79et 2.73** tabulées par Dickey-Fuller respectivement aux seuils **1%, 5% et 10%.** Nous avons donc par la suite passé à l'étape [2].

Modèle [2]

$$\Delta DPPT_t = \phi DPPT_{t-1} + \mathbf{c} + \sum_{j=1}^{d} \phi_j \Delta DPPT_{t-j} + \varepsilon_t .$$

Avec ε_t est un processus stationnaire.

Le résultat de l'affichage pour la série **DPPT** est donné dans la table suivante :

Variable	Coefficient	Std. Error	t-Statistic	Prob.
DPPT(-1)	-2.625642	0.303116	-8.662164	0.0000
D(DPPT(-1))	0.907720	0.250320	3.626235	0.0004
D(DPPT(-2))	0.465390	0.176800	2.632287	0.0096
D(DPPT(-3))	0.160770	0.091961	1.748230	0.0829
C	1423.152	2379.867	0.597996	0.5510

La constante n'est pas significativement différente de zéro puisque sa t-statistique (**0.5979**) est inférieure aux valeurs tabulées par Dickey-Fuller aux seuils 1%, 5% et 10% qui sont respectivement égales à **3.78, 3.11 et 2.38**. On passe alors à l'étape [3].

Modèle [1]

$$\Delta DPPT_t = \phi DPPT_{t-1} + \sum_{j=1}^{d} \phi_j \Delta DPPT_{t-j} + \varepsilon_t .$$

Avec ε_t est un processus stationnaire.

Le résultat de l'affichage pour la série **DPPT** est donné dans la table suivante :

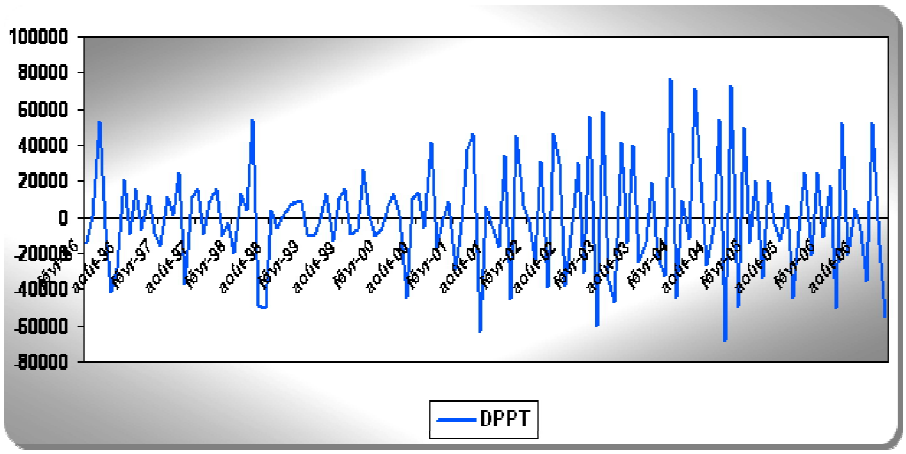
		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-8.672968	0.0000
Test critical values:	1% level	-2.583298	
	5% level	-1.943364	
	10% level	-1.615050	

Nous procédons au test de racine unitaire, la valeur estimée de la statistique ADF est égale à (**-8.6729)** (voir la table ci-dessus). Cette valeur est inférieure aux valeurs critiques (-2.5831, -1.9433 et -1.6150) aux seuils 1%, 5% et 10%. Par conséquent, nous rejetons l'hypothèse nulle de racine unitaire tel que H_0 : **(**il existe une racine unitaire ϕ =0**).**

Conclusion :

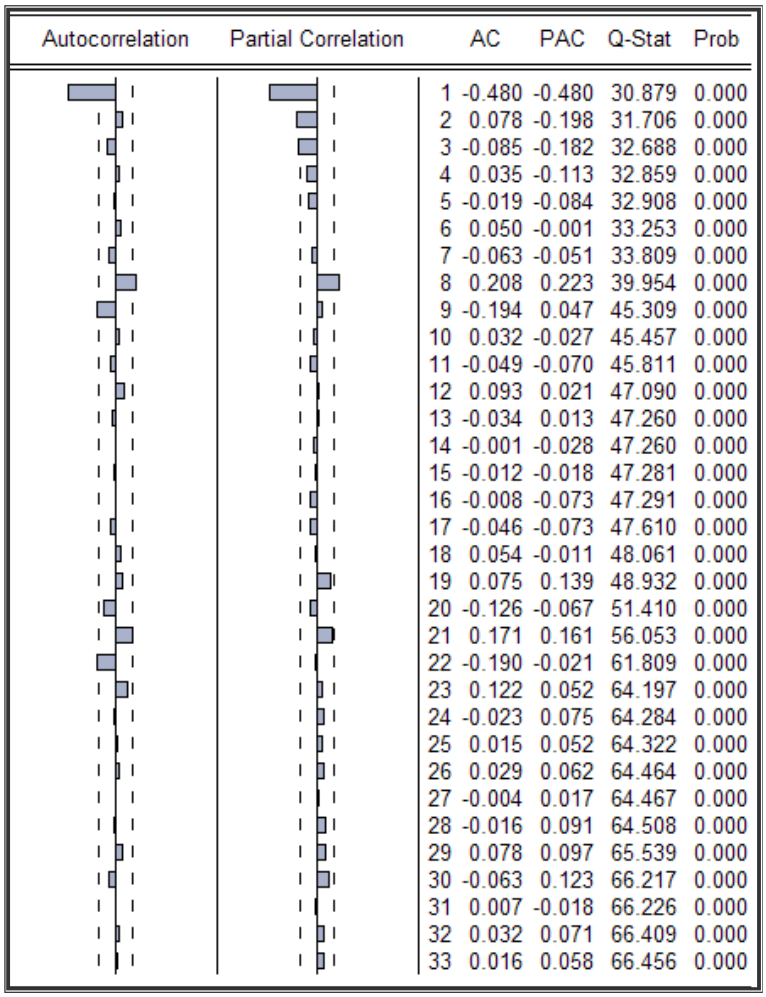
Selon le test **ADF** nous pouvons conclure que la série **DPPT** est **stationnaire.**

Diagramme séquentiel de la série DPPT



La série **DPPT** est stationnaire de type **I(0),** et le graphe de son diagramme séquentiel le confirme, ainsi que ses autocorrélations simples et partielle.

Corrélogramme de la série DPPT



Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	-0.480	-0.480	30.879	0.000
		2	0.078	-0.198	31.706	0.000
		3	-0.085	-0.182	32.688	0.000
		4	0.035	-0.113	32.859	0.000
		5	-0.019	-0.084	32.908	0.000
		6	0.050	-0.001	33.253	0.000
		7	-0.063	-0.051	33.809	0.000
		8	0.208	0.223	39.954	0.000
		9	-0.194	0.047	45.309	0.000
		10	0.032	-0.027	45.457	0.000
		11	-0.049	-0.070	45.811	0.000
		12	0.093	0.021	47.090	0.000
		13	-0.034	0.013	47.260	0.000
		14	-0.001	-0.028	47.260	0.000
		15	-0.012	-0.018	47.281	0.000
		16	-0.008	-0.073	47.291	0.000
		17	-0.046	-0.073	47.610	0.000
		18	0.054	-0.011	48.061	0.000
		19	0.075	0.139	48.932	0.000
		20	-0.126	-0.067	51.410	0.000
		21	0.171	0.161	56.053	0.000
		22	-0.190	-0.021	61.809	0.000
		23	0.122	0.052	64.197	0.000
		24	-0.023	0.075	64.284	0.000
		25	0.015	0.052	64.322	0.000
		26	0.029	0.062	64.464	0.000
		27	-0.004	0.017	64.467	0.000
		28	-0.016	0.091	64.508	0.000
		29	0.078	0.097	65.539	0.000
		30	-0.063	0.123	66.217	0.000
		31	0.007	-0.018	66.226	0.000
		32	0.032	0.071	66.409	0.000
		33	0.016	0.058	66.456	0.000

D'après les corrélogrammes simple et partiel de la série stationnaire **DPPT**, nous remarquons que la fonction d'autocorrélation simple (AC) possède des valeurs importantes aux retards (*q=1, 8, 9, et 22*) *e*t que la fonction d'autocorrélation partielle (PAC) possède des valeurs importantes aux retards (*p=1, 2,3 et 8*).

Par conséquent nous avons plusieurs modèles candidats parmi lesquels nous avons sélectionné deux modèles

Modèles	AIC	BIC
ARiMA (0,1,1)	**23.20**	**23.22**
ARiMA (1,1,2)	23.22	23.27

Nous avons choisi le modèle qui minimise les deux critères AIC et BIC et qui est le modèle **ARIMA (0,1,1).**

Estimation du Processus ARIMA (0,1,1)

L'estimation des Processus ARMA repose sur la méthode du maximum de vraisemblance. On suppose que les résidus suivent une loi normale de moyenne nulle et de variance σ_ε^2.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
MA(1)	-0.676150	0.066764	-10.12746	0.0000

R-squared	0.321659	Mean dependent var	327.8397
Adjusted R-squared	0.321659	S.D. dependent var	32025.16
S.E. of regression	26376.38	Akaike info criterion	23.20593
Sum squared resid	9.04E+10	Schwarz criterion	23.22788
Log likelihood	-1518.988	Durbin-Watson stat	2.002416

Inverted MA Roots	.68

L'étape d'estimation achevée, l'étape suivante permet de valider ou non le modèle estimé.

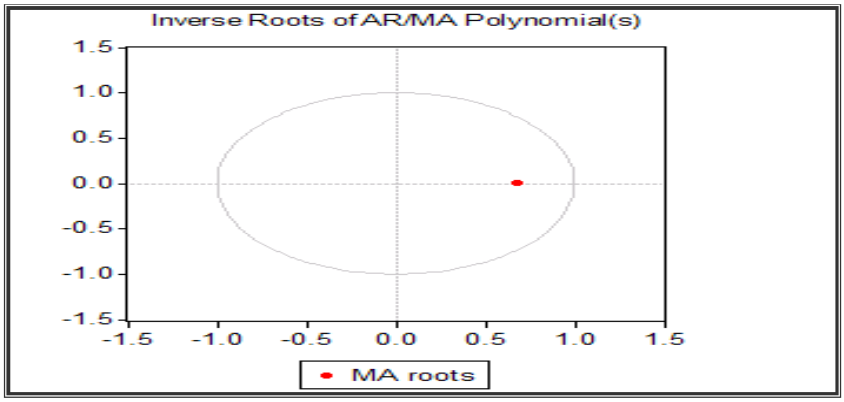
Validation du processus ARIMA (0,1,1)

Test sur les paramètres

Nous remarquons que le paramètre du modèle est significativement différent de zéro.

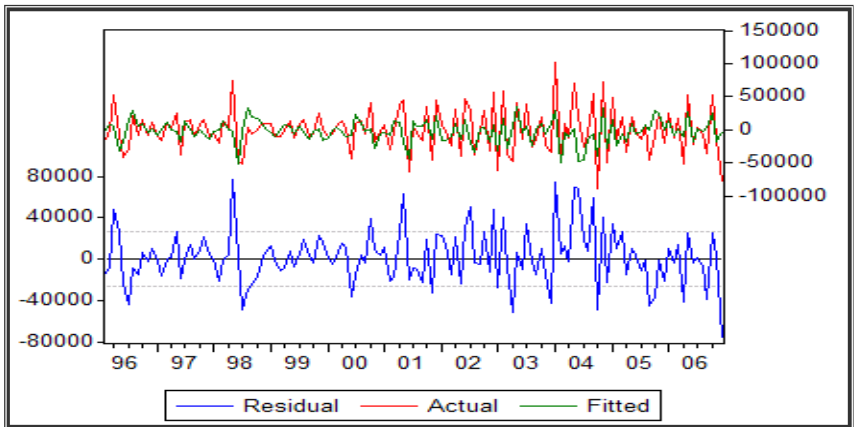
En effet le rapport du coefficient du modèle est en valeur absolue supérieure à *1.96*, ce qui est confirmé par la probabilité de nullité du coefficient qui est inférieure à *0.05* ***(voir tableau estimation du processus précédent*)**.

Graphique des inverses des racines



D'après la représentation graphique des inverses des racines des polynômes de retards moyenne mobile et autorégressif on s'aperçoit qu'il est supérieur à 1 en module (leurs inverses sont en module, inférieurs à 1*)*.

Graphique des séries résiduelles réelles et estimées



A partir de la représentation graphique des séries résiduelles réelles et estimées nous constatons que le modèle estimé ajuste parfaitement la série **DPPT**.

Il convient maintenant d'analyser les résidus à partir de leur fonction d'autocorrélation et d'appliquer une série de tests.

Tests sur les résidus

Ces tests ont pour objet de vérifier la blancheur des résidus estimés en appliquant des tests d'absence d'autocorrélation et des tests d'homoscédasticité**.**

Tests d'absence d'autocorrélation

Il existe un grand nombre de tests d'absence d'autocorrélation, les plus connus étant ceux de Box et Pierce (1970) et Ljung et Box (1978), test des points de retournements dont on rappelle brièvement le principe ci-dessous.

- **Test de Box – Ljung**

On a à tester l'hypothèse nulle **:**

H_0 : « Les autocorrélations jusqu'au pas K, (K=N/5) ne sont pas significatives » C'est-à-dire

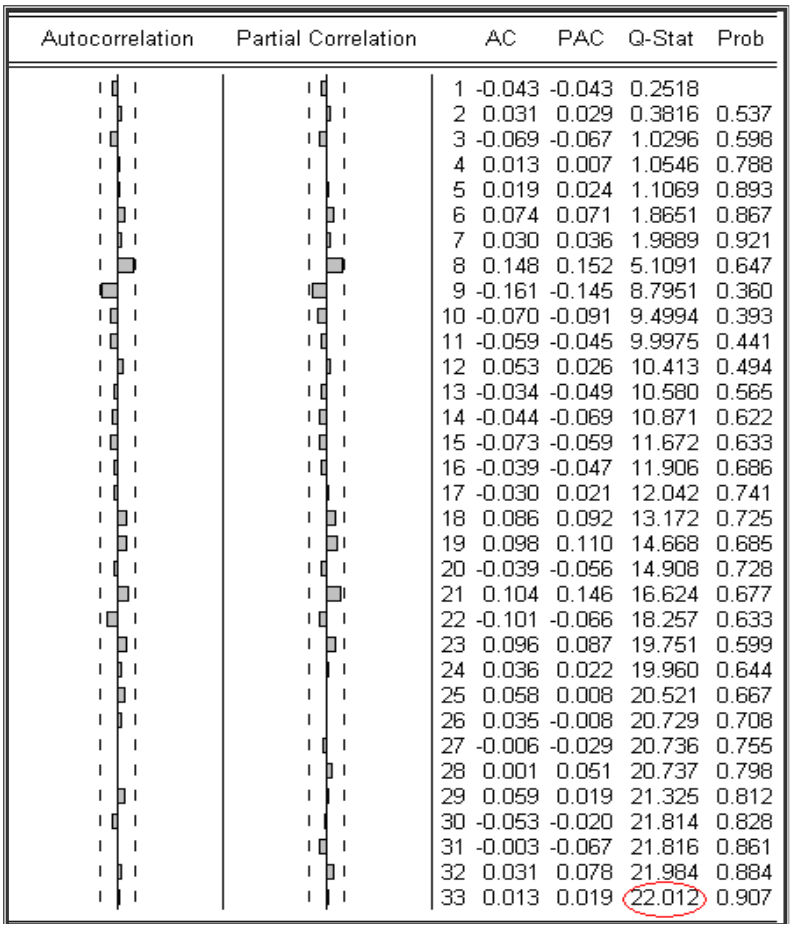
H_0 : « $\rho_1 = \rho_2 = ... = \rho_K = 0$ » Contre H_1 : « $\exists \rho_j \ : \ j = 1,\ldots,99$ tel que $\rho_j \neq 0$ ».

Ce test est basé sur la statistique de Box-Ljung au pas k : $Q = n(n+1)\sum_{k=1}^{K} \frac{\gamma_k^2}{n-k}$.

Si $Q < \chi^2_{0,95} \quad (K-p-q-P-Q)$, nous acceptons l'hypothèse H_0.

Les valeurs de la statistique de Box-Ljung ont de fortes probabilités. Ce qui entraîne à dire que les résidus forment un bruit blanc.

Corrélogramme simple et partiel des résidus



Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	-0.043	-0.043	0.2518	
		2	0.031	0.029	0.3816	0.537
		3	-0.069	-0.067	1.0296	0.598
		4	0.013	0.007	1.0546	0.788
		5	0.019	0.024	1.1069	0.893
		6	0.074	0.071	1.8651	0.867
		7	0.030	0.036	1.9889	0.921
		8	0.148	0.152	5.1091	0.647
		9	-0.161	-0.145	8.7951	0.360
		10	-0.070	-0.091	9.4994	0.393
		11	-0.059	-0.045	9.9975	0.441
		12	0.053	0.026	10.413	0.494
		13	-0.034	-0.049	10.580	0.565
		14	-0.044	-0.069	10.871	0.622
		15	-0.073	-0.059	11.672	0.633
		16	-0.039	-0.047	11.906	0.686
		17	-0.030	0.021	12.042	0.741
		18	0.086	0.092	13.172	0.725
		19	0.098	0.110	14.668	0.685
		20	-0.039	-0.056	14.908	0.728
		21	0.104	0.146	16.624	0.677
		22	-0.101	-0.066	18.257	0.633
		23	0.096	0.087	19.751	0.599
		24	0.036	0.022	19.960	0.644
		25	0.058	0.008	20.521	0.667
		26	0.035	-0.008	20.729	0.708
		27	-0.006	-0.029	20.736	0.755
		28	0.001	0.051	20.737	0.798
		29	0.059	0.019	21.325	0.812
		30	-0.053	-0.020	21.814	0.828
		31	-0.003	-0.067	21.816	0.861
		32	0.031	0.078	21.984	0.884
		33	0.013	0.019	22.012	0.907

L'analyse du Corrélogramme des résidus, montre que tous les termes sont à l'intérieur de l'intervalle de confiance (illustré par des pointillés sur le graphique) et les valeurs de la statistique de Box-Ljung ont de fortes probabilités. Ce qui nous entraîne à dire que les résidus forment un bruit blanc.

La valeur de la statistique de Box-Ljung quand K = 33, p = 0, q = 1 est égale à (22.012) est inférieure à $\chi^2_{0.,95}(32) = 44$. Nous concluons alors que les erreurs ne sont pas corrélées.

Conclusion

Les résultats du test de Box – Ljung sont identiques à ce qu'on a remarqué de visu sur les corrélogrammes simple et partiel.

Test des points de retournements

Il s'agit de tester l'hypothèse nulle :

H_0 : « Les ε_i sont aléatoires » **VS** H_1 : « Il existe une corrélation entre les ε_i i = 1,…, n ».

Après les calculs à l'aide du logiciel **MATLAB** nous avons obtenu les résultats suivants :

Le nombre des points de retournements égal à P = $\sum_{i=1}^{n-2} X_i$ = 87.

Nous avons : n = 131 donc $E(P) = \frac{2}{3}(n-2) = 86$ *et* $VAR(P) = \frac{16n-29}{90} = 22.97$

$\sqrt{VAR(P)} = 4.79$, $S = \frac{P - E(P)}{\sqrt{Var(P)}} = 0.21$.

Donc : $|S|$ =0.21< S (tabulée) = 1,96 alors, nous acceptons l'hypothèse H_0.

- **Test de la nullité de la moyenne des résidus :**

Pour tester l'hypothèse H_0 : « m=0 » contre H_1 : « m ≠ 0 » nous utilisons le test de Student basé sur la statistique : t = $\frac{\bar{\varepsilon}_t}{\sigma_\varepsilon / \sqrt{n-1}}$.

Si $|t| < t_{n-1}$ à 5% (=1,96) nous acceptons l'hypothèse de la nullité de la moyenne des résidus.

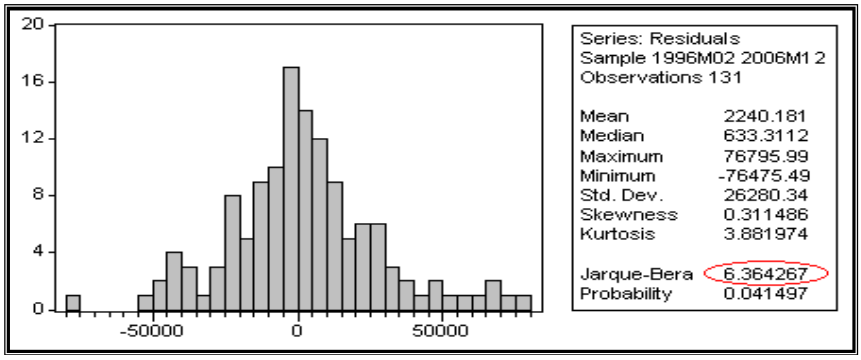
Test sur échantillon unique

	Valeur du test = 0					
	t	ddl	Sig. (bilatérale)	Différence moyenne	Intervalle de confiance 95% de la différence	
					Inférieure	Supérieure
RDPPT	,976	130	,331	2240,1807	-2302,43	6782,7876

D'après le **Tableau ci-dessus** nous avons : $|t|$ = 0.976 qui est inférieure à 1,96 ; Donc nous acceptons l'hypothèse H_0 : « la moyenne des résidus est nulle ».

- **Tests de normalité sur les résidus du modèle optimal :**

Les tests sont effectués à partir des valeurs empiriques des coefficients de Skewness, Kurtosis et la statistique de Jarque-Bera données par le logiciel EVIEWS 5.0. En utilisant le logiciel nous avons l'histogramme suivant :



- **Test de Skewness (Asymétrie) et de Kurtosis (Aplatissement) :**

Nous testons les hypothèses suivantes : $\begin{cases} H_0 : \gamma_1 = 0 \quad et \quad \gamma_2 = 0 \\ H_1 : \gamma_1 \neq 0 \quad ou \quad \gamma_2 \neq 0 \end{cases}$

Test de Skewness : $\gamma_1 = \left| \dfrac{\beta_1^{1/2} - 0}{\sqrt{\dfrac{6}{N}}} \right|$ **Test de Kurtosis** : $\gamma_2 = \left| \dfrac{\beta_2 - 3}{\sqrt{\dfrac{24}{N}}} \right|$

Où : $\beta_1^{1/2} = \dfrac{\mu_3}{\mu_2^{3/2}}$: est le coefficient de Skewness (l'indicateur d'asymétrie des résidus).

$\beta_2 = \dfrac{\mu_4}{\mu_2^2}$: est le coefficient de Kurtosis (le degré d'aplatissement de la loi des résidus).

Sous l'hypothèse H_0 et si le nombre d'observations est assez grand (N >30), nous avons :

$$\beta_1^{1/2} = \frac{\mu_3}{\mu_2^{3/2}} \xrightarrow[N\to\infty]{\wp} N\left(0, \sqrt{\frac{6}{N}}\right) \cdots\cdots\cdots (1) .$$

$$\beta_2 = \frac{\mu_4}{\mu_2^2} \xrightarrow[N\to\infty]{\wp} N\left(0, \sqrt{\frac{24}{N}}\right) \cdots\cdots\cdots\cdots (2).$$

Après calculs nous avons obtenu :

Test de Skewness: $\gamma_1 = \left| \dfrac{\beta_1^{1/2} - 0}{\sqrt{\dfrac{6}{N}}} \right| = 1.45 < 1, 96.$

Test de Kurtosis : $\gamma_2 = \left| \dfrac{\beta_2 - 3}{\sqrt{\dfrac{24}{N}}} \right| = 2.06 > 1,96.$

Alors : les résidus ne sont pas gaussiens. Ce qui est confirmé par le test de Jarque et Bera.

- **Test de Jarque et Bera :**

Nous définissons la statistique S par : $S = \frac{N}{6}\beta_1 + \frac{N}{24}(\beta_2 - 3)^2$

Sous (1) et (2) : $S \rightarrow \chi^2_{1-\alpha}(2)$

Nous testons H_0 : « accepter la normalité des résidus au seuil α=0,05 » contre

H_1 : « Il n'y a pas de normalité des résidus ».

Si $S > \chi^2_{1-\alpha}(2)$ nous rejetons L'hypothèse H_0 sinon nous l'acceptons.

D'après le tableau la statistique de Jarque et Bera notée (S) est égale à 6.364 ; elle est supérieure à $\chi^2(2) = 5{,}99$. Nous concluons que les résidus forment un bruit blanc non gaussien.

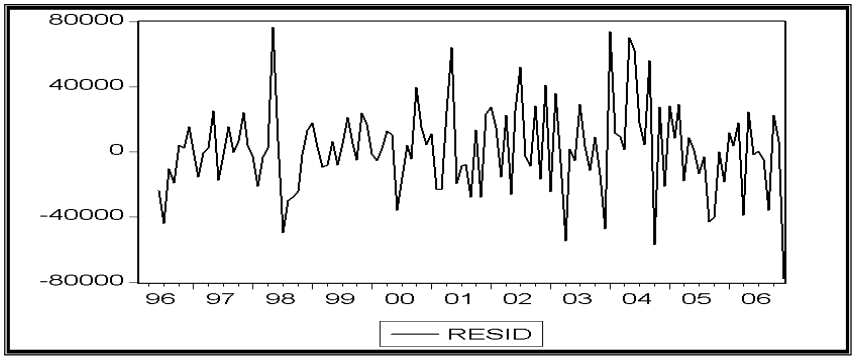
- **Remarque :**

  On n'a pas appliqué le test de Durbin-Watson et le test d'indépendance de Von Neumann à cause de non normalité.

- **Test d'homoscédasticité**

Test d'effet ARCH

Une première observation du graphe des résidus ci-dessous montre que la moyenne de cette série est constante alors que sa variance change au cours du temps. De plus le Processus étant non gaussien, on suspecte la présence d'un effet ARCH.



Soient les hypothèses :

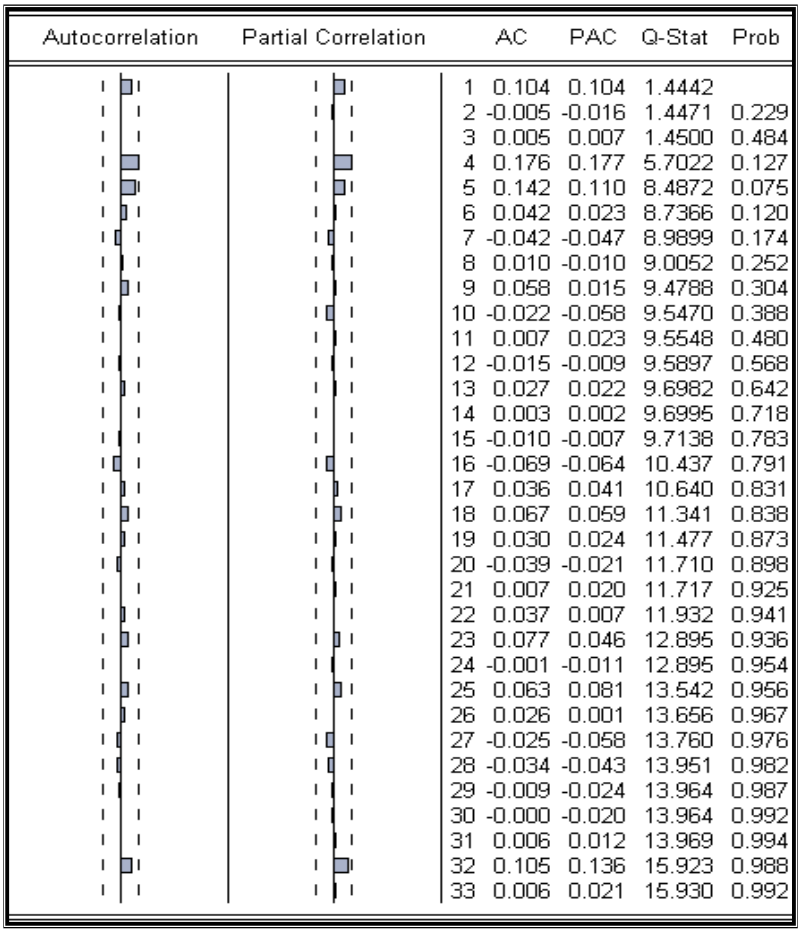
H_0 : « Les résidus sont homoscédastiques » **VS** H_1 : « Les résidus sont hétéroscedastiques ».

L'hypothèse nulle testée est celle d'homoscédasticité $H_0 : "\alpha_1 = \alpha_2 = \ldots\ldots = \alpha_p = 0"$ contre

L'hypothèse alternative d'hétéroscédasticité conditionnelle

H_1 :"$\exists\, i\,,\ i = 1 \ldots p \quad tel\ que\ \alpha_i \neq 0$" .Si l'hypothèse H_0 est acceptée, la variance conditionnelle de l'erreur est constante $\sigma_\varepsilon^2 = \alpha_0$. En revanche, si l'hypothèse nulle est rejetée, les résidus suivent un processus *ARCH (p)* dont l'ordre *p* est à déterminer.

Corrélogrammes Simple et Partiel des Résidus au Carré



	AC	PAC	Q-Stat	Prob
1	0.104	0.104	1.4442	
2	-0.005	-0.016	1.4471	0.229
3	0.005	0.007	1.4500	0.484
4	0.176	0.177	5.7022	0.127
5	0.142	0.110	8.4872	0.075
6	0.042	0.023	8.7366	0.120
7	-0.042	-0.047	8.9899	0.174
8	0.010	-0.010	9.0052	0.252
9	0.058	0.015	9.4788	0.304
10	-0.022	-0.058	9.5470	0.388
11	0.007	0.023	9.5548	0.480
12	-0.015	-0.009	9.5897	0.568
13	0.027	0.022	9.6982	0.642
14	0.003	0.002	9.6995	0.718
15	-0.010	-0.007	9.7138	0.783
16	-0.069	-0.064	10.437	0.791
17	0.036	0.041	10.640	0.831
18	0.067	0.059	11.341	0.838
19	0.030	0.024	11.477	0.873
20	-0.039	-0.021	11.710	0.898
21	0.007	0.020	11.717	0.925
22	0.037	0.007	11.932	0.941
23	0.077	0.046	12.895	0.936
24	-0.001	-0.011	12.895	0.954
25	0.063	0.081	13.542	0.956
26	0.026	0.001	13.656	0.967
27	-0.025	-0.058	13.760	0.976
28	-0.034	-0.043	13.951	0.982
29	-0.009	-0.024	13.964	0.987
30	-0.000	-0.020	13.964	0.992
31	0.006	0.012	13.969	0.994
32	0.105	0.136	15.923	0.988
33	0.006	0.021	15.930	0.992

L'analyse du Corrélogramme des résidus carrés, montre que tous les termes sont à l'intérieur de l'intervalle de confiance, ce qui nous entraîne à dire qu'il n y a pas d'effet ARCH.

Conclusion :

Nous pouvons conclure, d'après ces tests que les résidus forment bien **un bruit blanc non gaussien**. Finalement le modèle qui ajuste au mieux la série **DPPT** est **ARIMA (0,1,1)** Ainsi, le modèle qui ajuste au mieux la série **PPT** s'écrit sous la forme suivante :

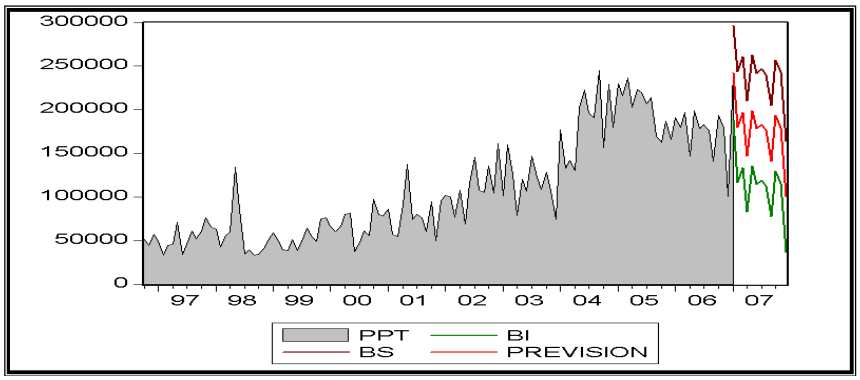
$$(1 - B)\,PPT_t = (1 + 0,6762 \times B)\,\varepsilon_t$$

- **Prévisions**

Pour faire des prévisions, on remplace t par t+h dans l'expression ci-dessus du modèle générateur de la série. Les prévisions de la série **PPT** sont calculées pour un horizon h =12 ; c'est-à-dire pour la période allant de **janvier 2007** à **Décembre 2007.**

- Le graphe suivant représente les réalisations et les valeurs prévues pour l'année 2007 pour la série **PPT**.
- L'unité de mesure est la tonne.

Graphique de la série réelle et prévision



Mois	Prévisions	Borne Inf	Borne Sup
Janvier	242604	188527,94	296679,94
Février	179726	116046,2	243405,8
Mars	197070	133390,2	260749,8
Avril	146443	82763,2	210122,8
Mai	198975	135295,2	262654,8
Juin	178297	114617,2	241976,8
Juillet	182990	119310,2	246669,8
Août	176159	112479,2	239838,8
Septembre	140998	77318,2	204677,8
Octobre	193259	129579,2	256938,8
Novembre	179032	115352,2	242711,8
Décembre	100280	36600,2	163959,8

RESULTATS :

Les prévisions des produits pétroliers pour l'année 2007 sont estimées globalement à plus de **2.1** millions de tonnes pour un volume réalisé de **2.0** millions de tonnes en 2006.

Modèle non linéaire (Perceptron multi couches)

La prédiction des séries temporelles peut être effectuée par une méthode non linéaire telle que les réseaux de neurones.

Nous considérons la série des valeurs mensuelles du trafic des produits pétroliers au port d'Alger, composées en grande partie des débarquements de carburants et d'hydrocarbures gazeux sur une certaine période de temps représentée à la figure1.

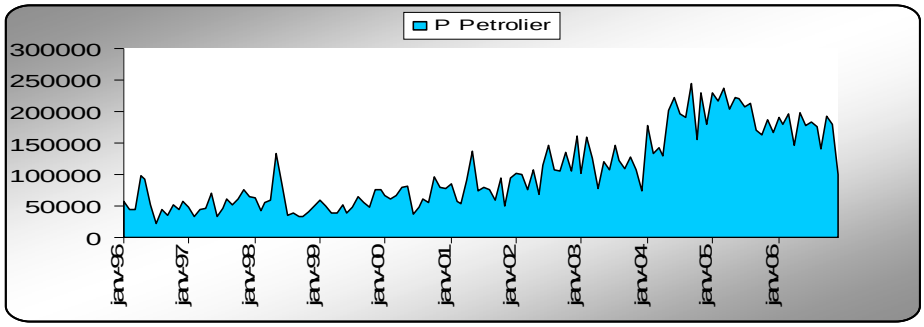


figure1

B. Le réseau utilisé

Pour dimensionner le réseau de neurones, nous pouvons d'abord déterminer son nombre d'entrées, autrement dit le nombre de valeurs antérieures sur lesquelles va se baser la prédiction de la valeur suivante.

Le but de notre travail est de :

- Présenter la manière par laquelle un réseau de neurone élabore la prévision des séries temporelles.
- Obtenir l'erreur la moins coûteuse en moyenne.

Pour ce faire, on va utiliser la même série décalée de la dernière année en tant que série explicative. Les données de la série seront divisées en deux parties (il existe plusieurs méthodes de division selon le type de données) : la première servira à l'estimation des paramètres du modèle, elle est appelée base du test, la deuxième sera un outil de validation des résultats obtenus relatifs à la première expérience.

La base du test : du 01 janvier 1996 au 31 décembre 2001.

La base de validation : du 01 janvier 2002 au 31 décembre 2007.

1. La normalisation des entrées

La normalisation des données sera effectuée de la manière suivante :

$$N(E) = \frac{E - \min}{\max - \min} \quad \textit{tel que} :$$

N (E) : la valeur normalisée.

E : la valeur brute.

Min : la valeur minimale enregistrée sur cette base du test pour chaque année.

Max: la valeur maximale enregistrée sur cette base du test pour chaque année.

2. Création d'un réseau de neurone

L'étape suivante est la création d'un réseau de neurones de type multicouche tel que :

- cinq neurones dans la couche d'input (Car il existe cinq ans dans la base du test).
- deux neurones dans la première couche cachée (choix arbitraire).
- Un neurone dans la deuxième couche cachée (choix arbitraire).

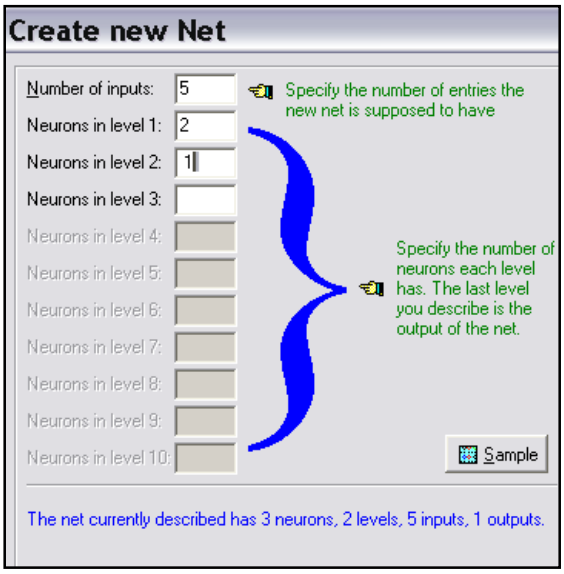


Figure 2 Création d'un MLP

- Un neurone constitue l'output (représente la dernière année de la base du test).

La définition de l'architecture doit être guidée par le souci de simplicité afin d'assurer une plus grande stabilité aux modèles.

Sur le plan pratique, on n'améliore pas l'efficacité d'un réseau en augmentant le nombre de couches intermédiaires.

L'architecture du réseau est présentée dans le graphique suivant :

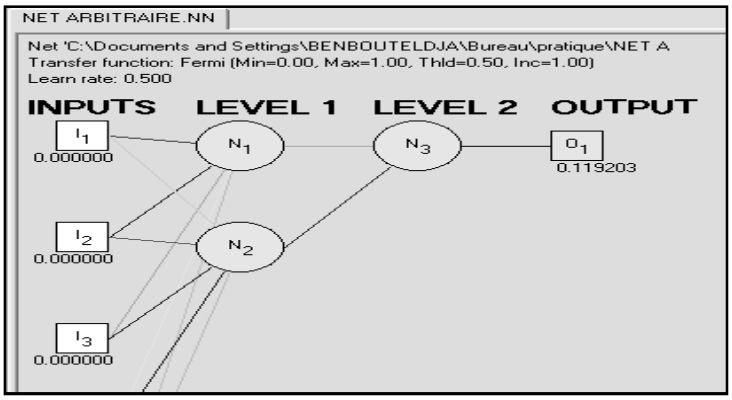


Figure3 : **architecture Perceptron multicouches**

Les poids sont initialisés aléatoirement, mais notre logiciel nous donne la possibilité de choisir l'intervalle d'initialisations. Pour notre cas, les poids appartiennent à]-1, +1].

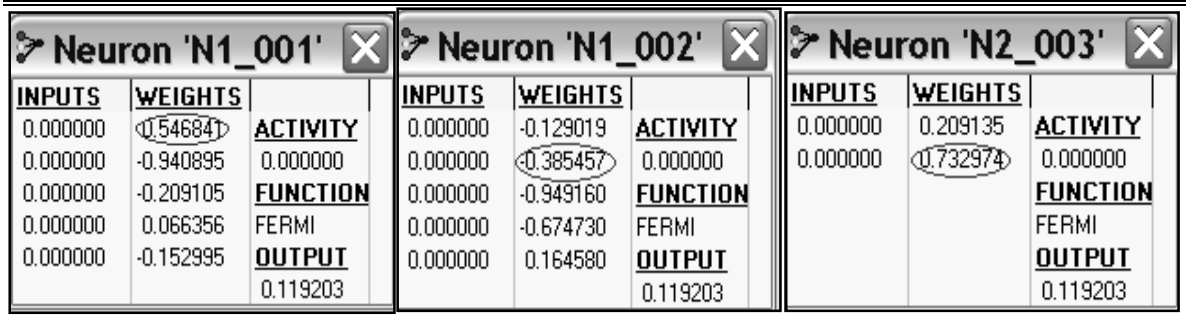


Figure4 : Initialisation des poids.

Le poids 0.5468 est celui de la connexion entre la première entrée et le premier neurone.

Le poids -0.3854est celui de la connexion entre la deuxième entrée et le deuxième neurone.

Le poids 0.7329 est celui de la connexion entre le deuxième neurone et le troisième neurone.

3. Estimation des paramètres du modèle

La phase d'apprentissage consiste à trouver le modèle (le vecteur des paramètres) qui minimisera le critère qu'on a choisi pour évaluer la performance de la prévision, celui-ci étant la moyenne des écarts des erreurs » MSE.

$$MSE = \frac{1}{N}\sum SQ.DV_E \quad \text{AVEC} \quad SQ.DV_E = \left[O1_E - O1(NET)_E\right]^2$$

$O1_E$: la vrai valeur de la série.

$O1(NET)_E$: la valeur calculée par le réseau de neurone.

L'algorithme de la rétropropagation du gradient ajustera les poids des connexions du réseau pour obtenir un modèle de prévision utilisable et le mieux adapté à notre série ce qui signifie minimiser l'écart entre les valeurs cibles et les valeurs prédites produite par le réseau de neurone.

Avant de lancer le processus de l'apprentissage du réseau nous devons régler les critères d'apprentissage à partir de la fenêtre de contrôle qui apparaît en **Figure 5**.

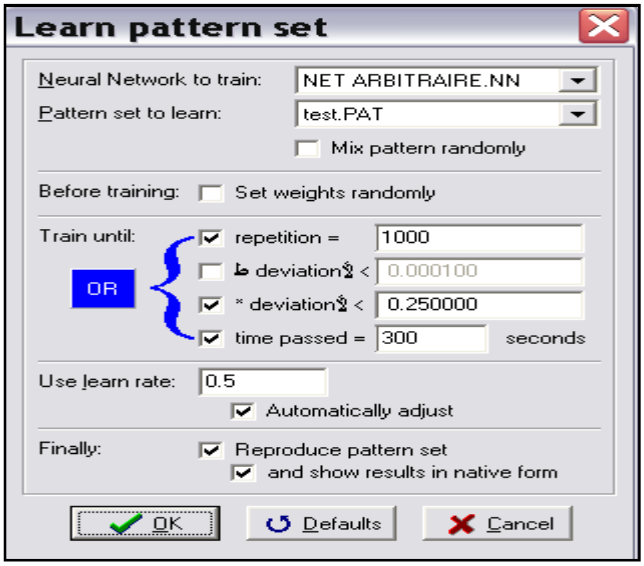


Figure5 : Boite de contrôle

MSE1 est la moyenne de tous les carrés des erreurs de prévision, c'est à dire les carrés des différences entre les prévisions et les valeurs cibles prises sur la série chronologique, Elle est désignée par ϕ ***deviation***^{2} (la déviation moyenne), et * ***deviation***^{2} (la déviation maximale dans l'ensemble de modèle) que l'on fixe inférieure à 0.25. Egalement, le nombre de répétitions de l'algorithme, qui signifie le nombre de fois qu'il ajustera les poids du modèle, est fixé à 1000 répétitions.

Le taux d'apprentissage conditionnant l'ampleur de variations des poids, une valeur trop petite de celui-ci conduit à des pas petits et donc à une convergence lente de l'algorithme. En revanche, un taux élevé induit de grandes variations et parfois des oscillations autour du minimum local de la fonction d'erreur. Le logiciel nous propose par défaut la valeur de 0.5.

Les résultats de la phase d'apprentissage sont :

N(test.PAT)

	I1	I2	I3	I4	I5	O1	O1(NET)	SQ DV
1	0.694176	0.365157	0.729520	1.000000	0.250726	0.720231	0.761038	0.001665
2	0.312289	0.000000	0.000000	0.356409	0.120118	0.000000	0.459475	0.211117
3	0.331741	0.264103	0.315628	0.027069	0.265537	0.426063	0.433090	0.000049
4	1.000000	0.069524	0.417781	0.000000	0.589837	0.853262	0.799500	0.002890
5	0.830025	0.879300	1.000000	0.487169	0.620922	1.000000	0.765747	0.054874
6	0.513359	0.025530	0.787533	0.013682	0.049329	0.428607	0.627266	0.039466
7	0.000000	0.290444	0.053141	0.017740	0.050937	0.565293	0.198105	0.134827
8	0.313347	0.650295	0.163574	0.202049	0.377181	0.723467	0.378329	0.119120
9	0.060158	0.437484	0.019898	0.236842	0.000000	0.078914	0.193595	0.013152
10	0.516277	0.639963	0.039315	0.021733	1.000000	0.688196	0.743647	0.003075
11	0.318840	1.000000	0.203658	0.877357	0.574138	0.558031	0.531918	0.000682
12	0.681532	0.524036	0.420901	0.911855	0.535851	0.696273	0.755865	0.003551

Average deviation of 'NET ARBITRAIRE.NN': 0.048706

Figure 6 : Apprentissage de données

La moyenne des écarts d'erreurs pour cette étape est de ***MSE1* = 0.048706**. Cette moyenne est obtenue à partir des calculs suivants :

La fonction d'activation des neurones devra être non-linéaire, pour que le perceptron multicouche implémente une transformation non-linéaire, et signée, pour pouvoir prédire des valeurs positives ou négatives. Nous optons donc pour la fonction d'allure sigmoïdale « *tansig* » en couche(s) cachée(s) et en couche de sortie.

La fonction de transfert est donnée par :

$$\boldsymbol{F}(\boldsymbol{X}) = \frac{1}{1+\exp\left[-4(\boldsymbol{X}-0.5)\right]}$$

- Pour le troisième neurone :

$$ACTIVITY = \sum_{1}^{2} INPUTS * WEIGHTS$$
$$= (0.9471 * 0.8238) + (0.0027 * 0.8187)$$
$$= 0.7825.$$
$$F(0.7825) = \frac{1}{1+\exp\left[-4*(0.7825-0.5)\right]}$$
$$= 0.7558.$$

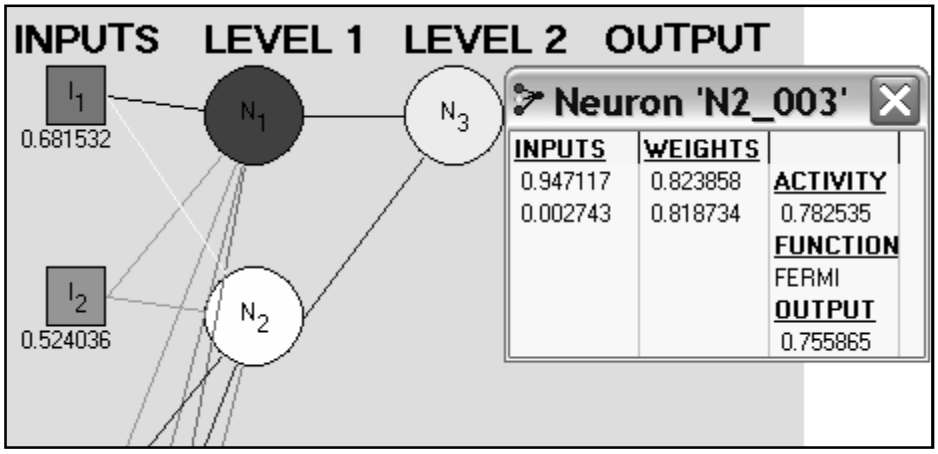


Figure7 : les poids après l'Apprentissage des données

La moyenne des carrés des erreurs après l'apprentissage est ***MSE1*= 0.048706,** selon le principe de l'algorithme de propagation du gradient, les poids se sont modifiés de manière à ce que l'erreur relative soit minimale.

4. La recherche de la meilleure structure (optimale)

A ce niveau, la moyenne ***MSE***1 est évaluée à **0.048706**, cette moyenne est censée diminuer car nous allons rechercher la meilleure structure du réseau.

Nous devons régler les paramètres de l'Algorithme d'optimisation tel que :

- ***deviation***^{2} doit être inférieure à 0.25
- le nombre de neurone doit être inférieur ou égal à 5
- la taille de la population : 50 réseaux de neurone par génération.
- le nombre de répétitions de l'algorithme (c'est le nombre maximum de génération avant que l'algorithme stoppe son travail) est fixé à 1000.

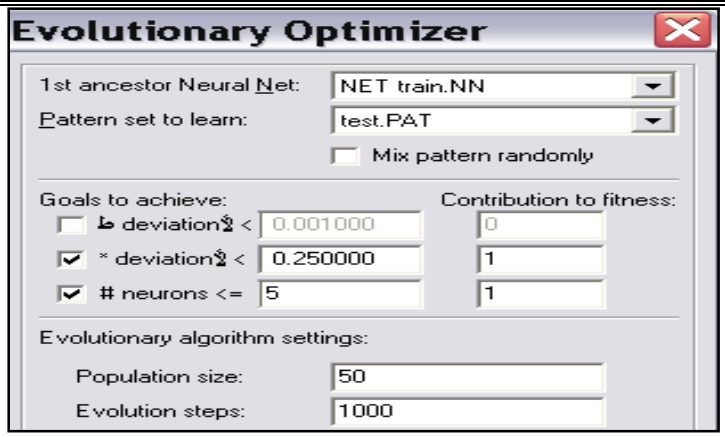


Figure 7 : Boite de contrôle

La procédure du travail de cet algorithme est la suivante :

- commencent par créer une génération de 50 réseaux de neurones dont il désigne l'architecture.
- Chaque réseau appartenant à la génération va être entraîné brièvement, sa convenance est déterminée suivant le but à atteindre qui est une moyenne inférieure à 0.25.
- La sélection s'effectuera jusqu'à ce que la génération atteint 50 membres. Le Logiciel va s'arrêter dès qu'il aura trouvé un réseau possédant **'fitness'** *de 100*. Sinon il terminera la recherche jusqu'à 1000 générations.

- On choisit la structure de réseau ayant la *déviation minimale ***0.000889***, celle-ci sera la meilleure structure, c'est le réseau numéro **35** ayant **5** neurones dont l'architecture est (**5,4,1**) (voir la figure 8).

No	Topology	Neurons	dev²	* dev²	Fitness
16	5,3,1	4	0.000520	0.003808	100.0000
17	5,4,4,1	9	0.000677	0.004664	77.77778
18	5,5,3,1	9	0.000757	0.002995	77.77778
19	5,5,3,1	9	0.000037	0.000262	77.77778
20	5,5,3,1	9	0.004157	0.039658	77.77778
21	5,3,1	4	0.009810	0.067700	100.0000
22	5,6,3,1	10	0.000226	0.001663	75.00000
23	5,6,3,1	10	0.000113	0.000667	75.00000
24	5,2,1	3	0.014479	0.093612	100.0000
25	5,2,1	3	0.016974	0.071610	100.0000
26	5,1	1	0.021384	0.129290	100.0000
27	5,1	1	0.021384	0.129290	100.0000
28	5,1	1	0.021384	0.129290	100.0000
29	5,1	1	0.021384	0.129290	100.0000
30	5,1	1	0.021384	0.129290	100.0000
31	5,1	1	0.021384	0.129290	100.0000
32	5,1	1	0.021384	0.129290	100.0000
33	5,1	1	0.021384	0.129290	100.0000
34	5,1	1	0.021384	0.129290	100.0000
35	5,4,1	5	0.000889	0.004790	100.0000

Ok Stop Cancel

Figure 8 : Résultat de l'algorithme d'optimisation

Résultat :

Le réseau possède cinq neurones sur la couche d'entrée, quatre neurones à la première couche cachée et un neurone à la deuxième couche cachée, enfin un neurone à la couche de sortie.

L'architecture du réseau et les poids associés sont :

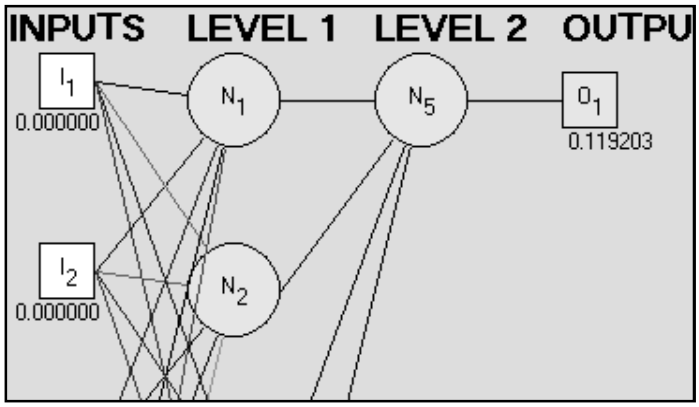


Figure 9: L'architecture du réseau optimal

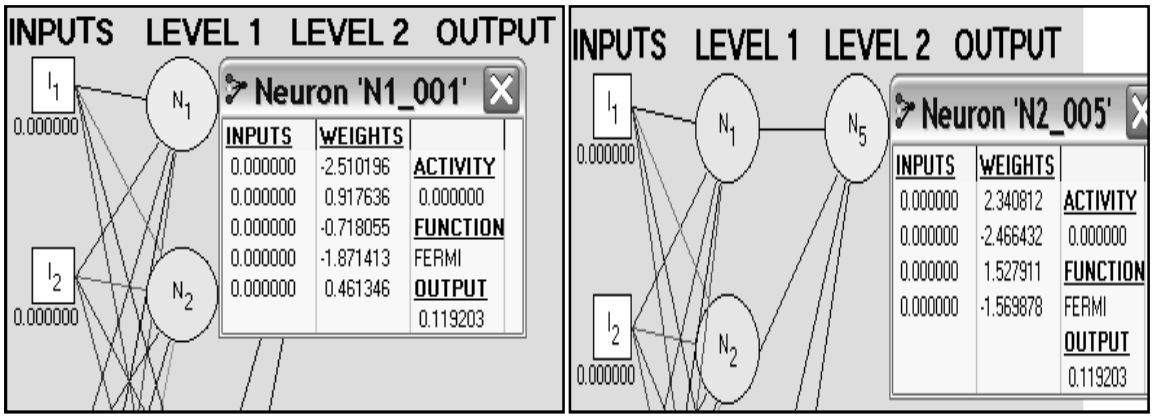


Figure10 : Nouvelle initialisation des poids

5. L'estimation des paramètres du réseau optimal

Il va falloir refaire l'étape de l'apprentissage à la nouvelle structure dans le but de minimiser de plus en plus la moyenne des carrés des erreurs. L'information, une fois introduite dans le réseau, se propage de l'entrée vers la sortie pour calculer la sortie du réseau. Sur la base de l'écart entre cette valeur et la vraie valeur, le poids sera modifié.

On règle les paramètres de la même manière précédemment décrite.

N(test.PAT)

	I1	I2	I3	I4	I5	O1	O1(NET)	SQ DV
1	0.694176	0.365157	0.729520	1.000000	0.250726	0.720231	0.743848	0.000558
2	0.312289	0.000000	0.000000	0.356409	0.120118	0.000000	0.002946	0.000009
3	0.331741	0.264103	0.315628	0.027069	0.265537	0.426063	0.495272	0.004790
4	1.000000	0.069524	0.417781	0.000000	0.589837	0.853262	0.877982	0.000611
5	0.830025	0.879300	1.000000	0.487169	0.620922	1.000000	0.966351	0.001132
6	0.513359	0.025530	0.787533	0.013682	0.049329	0.428607	0.448115	0.000381
7	0.000000	0.290444	0.053141	0.017740	0.050937	0.565293	0.565728	0.000000
8	0.313347	0.650295	0.163574	0.202049	0.377181	0.723467	0.718957	0.000020
9	0.060158	0.437484	0.019898	0.236842	0.000000	0.078914	0.104896	0.000675
10	0.516277	0.639963	0.039315	0.021733	1.000000	0.688196	0.695489	0.000053
11	0.318840	1.000000	0.203658	0.877357	0.574138	0.558031	0.594385	0.001322
12	0.681532	0.524036	0.420901	0.911855	0.535851	0.696273	0.729780	0.001123

Average deviation of 'NN OPT.NN': 0.000889

Figure11 : L'apprentissage pour le réseau optimal

Résultat

MSE2=0.00088 obtenue à partir de l'algorithme optimale a diminué, ce qui signifie que notre réseau a bien appris le comportement de la série chronologique.

Donc, Le deuxième modèle (réseau de neurones) optimal est plus performant que celui qu'on a définit aléatoirement selon le critère statistique *MSE* puisque *MSE1> MSE2*.

1.2.6/ Validation du modèle

Une fois effectué le cycle d'estimation des paramètres, il reste bien sûr à valider ces résultats en utilisant cette base de validation que le réseau ne connaît pas.

Cette étape consiste à présenter la deuxième partie des données appelées la base de validation. Ces données n'ont pas servi à l'estimation des paramètres du modèle**.**

On attend de notre réseau des résultats plus performants puisqu'il a appris suffisamment (1000 étapes) le comportement de la première partie de la série. On parle ici de la capacité du réseau à la généralisation, cette capacité sera traduite par la réponse correcte a des données non présentées au réseau auparavant.

On introduit les données de la base de validation, on doit utiliser la même procédure de normalisation.

N(test.PAT) N(validation.PAT)

	I1	I2	I3	I4	I5	O1	O1(NET)	SQ DV
1	0.250014	0.629037	0.173859	0.163593	0.628709	1.000000	--	--
2	0.434413	1.000000	0.000000	0.536680	0.604703	0.658634	--	--
3	0.336947	0.373766	0.236443	1.000000	0.755519	0.541339	--	--
4	0.357152	0.085723	0.215290	0.695026	0.537283	0.372300	--	--
5	0.000000	0.162908	0.476759	0.989780	1.000000	0.516978	--	--
6	0.701703	0.000000	0.603893	0.859618	0.780717	0.471543	--	--
7	0.508188	0.868268	0.773526	0.785237	0.671732	0.529415	--	--
8	0.551510	0.239015	0.708381	0.939223	0.531451	0.416859	--	--
9	0.500699	0.079902	1.000000	0.388917	0.220105	0.181162	--	--
10	0.890276	0.502350	0.531097	0.000000	0.882616	0.785641	--	--
11	0.879283	0.024234	0.928783	0.327958	0.590451	0.514947	--	--
12	1.000000	0.766116	0.568922	0.086623	0.000000	0.000000	--	--

Figure12 : normalisation des données

N(test.PAT) N(validation.PAT)

	I1	I2	I3	I4	I5	O1	O1(NET)	SQ DV
1	0.250014	0.629037	0.173859	0.163593	0.628709	1.000000	0.747556	0.063728
2	0.434413	1.000000	0.000000	0.536680	0.604703	0.658634	0.604874	0.002890
3	0.336947	0.373766	0.236443	1.000000	0.755519	0.541339	0.154674	0.149510
4	0.357152	0.085723	0.215290	0.695026	0.537283	0.372300	0.091733	0.078718
5	0.000000	0.162908	0.476759	0.989780	1.000000	0.516978	0.138086	0.143559
6	0.701703	0.000000	0.603893	0.859618	0.780717	0.471543	0.224364	0.061098
7	0.508188	0.868268	0.773526	0.785237	0.671732	0.529415	0.718605	0.035793
8	0.551510	0.239015	0.708381	0.939223	0.531451	0.416859	0.251247	0.027427
9	0.500699	0.079902	1.000000	0.388917	0.220105	0.181162	0.195794	0.000214
10	0.890276	0.502350	0.531097	0.000000	0.882616	0.785641	0.853488	0.004603
11	0.879283	0.024234	0.928783	0.327958	0.590451	0.514947	0.344171	0.029164
12	1.000000	0.766116	0.568922	0.086623	0.000000	0.000000	0.051237	0.002625

Average deviation of 'NN OPT.NN': 0.049944

Figure13 : validation du modèle optimale

A cette étape, on va voir le comportement du réseau en vers des données qui n'ont jamais servi aux calculs des paramètres.

La moyenne relative à la base de validation est **0.0499** (supérieure a celle de base de test), cela signifie que le modèle n'a pas validé toutes les valeurs du réseau.

Maintenant, on va calculer à partir des valeurs brutes les critères de performances des deux phases du réseau de neurones.

Les statistiques de comparaison sont données dans le tableau suivant :

Critères de Performance	Apprentissage	Validation
MAE	*950,532*	*11397,159*
MSE	*1457978,01*	*183098140*
MAPE	*1,2%*	*6,6%*

On remarque que le réseau est très performant. La valeur de l'écart absolu moyen en pourcentage est bien inférieure à 10 %.

6. Analyse Graphique

Pour vérifier visuellement les résultats obtenus, on a élaboré les graphiques suivants:

La figure 14 est caractérisée par une petite différence de niveau entre les vraies valeurs et celles calculées par le réseau optimal de la base de test.

X: les vraies valeurs.
O(NET) : valeurs obtenus par le réseau.

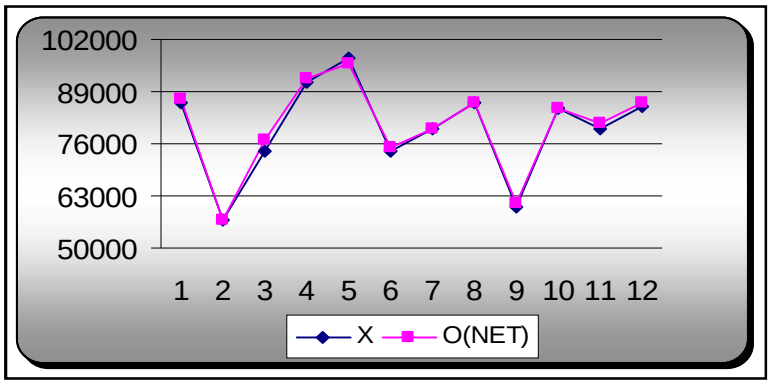


Figure14

Les valeurs de cible de la base de validation (Y) sont présentées au réseau de neurones dont on a estimé les paramètres, on remarque que les résultats des valeurs obtenus NET(Y) sont bien aussi ajustés aux vraies valeurs cibles, **(Voir Figure 15).**

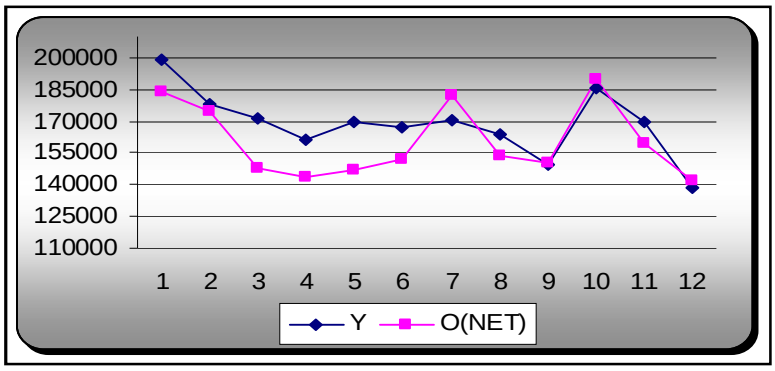


Figure15

Arrivé à ce stade d'analyse, l'erreur commise par le réseau optimal erreur 2 est inférieure a celle commise par le réseau initialisé aléatoirement erreur1.

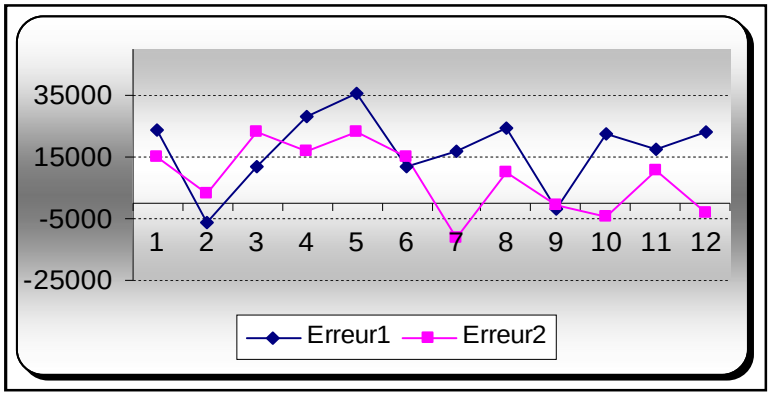


Figure16 : Comparaison des erreurs des deux réseaux

Analyse et comparaison :

Une méthode directe pour comparer l'efficacité des deux approches, consiste à comparer leur pouvoir prédictif, Il en résulte que le modèle **RN(5,4,1)** est plus performant que celui **ARIMA(0,1,1)** (voir le tableau suivant), ce jugement a eu lieu grâce aux critères MSE.

Modèles	**MSE** (L'erreur quadratique Moyenne)
ARIMA (0,1,1)	478533131
RN(5,4,1)	***183098140***

TABLEAU : Comparaison du pouvoir prédictif

Finalement on peut dire que les modélisations classiques offrent une source d'inspiration pour le choix d'architecture des réseaux de neurones, ce choix qui a resté une lacune, et qui a été considéré parmi les désavantages des réseaux de neurones dans l'étape de l'identification de l'architecture optimale.

Notre expérience pratique nous a permis de relever les résultats suivants :

D'un coté, plusieurs avantages des réseaux de neurones sont à noter :

- Des performances pouvant être supérieures aux modèles linéaires classiques.
- La détermination de l'architecture d'un réseau est un problème complexe d'optimisation, la structure influe sur la convergence de I 'algorithme d apprentissage, donc sur le résultat du réseau de neurone.
- Un outil de traitement de données simple et flexible permettant la manipulation visuelle des variables pour les gens non professionnels.
- Réalisation de la prévision de la série des produits pétroliers avec une MSE égale à (***183098140***).

D'un autre coté, on peut adresser aux réseaux de neurones les critiques suivants:

- L 'architecture optimale : il n'existe pas encore de théorie permettant de déterminer la structure optimale d'un réseau de neurone, en particulier, le nombre de couches cachées et le nombre de neurones sur chaque couche. L'intuition du prévisionniste et sa capacité à essayer plusieurs possibilités d'architecture sont le moyen pour réaliser de bons résultats.
- Le pouvoir explicatif des réseaux de neurones: l'aspect « boite noire» des réseaux de neurones ne permet pas d'extraire les relations pertinentes entre la variable explicative (les

entrées) et la variable à expliquer (la cible) donc il est difficile d'expliquer la non linéarité intuitivement.

- L 'intervenions de l'utilisateur : les réseaux de neurone font largement appel à l'intuition de l'homme et ses idées, en particulier pendant la phase d'apprentissage dont il convient de régler les paramètres manuellement. Mais le problème qui se pose est sur quelle base fixe-t-on le taux d'apprentissage (régler la vitesse de convergence de l'algorithme) sachant qu'un taux très grand peut aboutir à une oscillation (fluctuation et hésitation) du réseau. Par contre un taux faible se traduira par une convergence très lente, ce qui peut entraîner une longue durée de calcul. les chercheurs a partir de plusieurs expériences pratiques, ont suggéré un taux de 0.5 ou bien 0.9 pour conserver les capacités de généralisation du réseau. Cela reste insuffisant pour des modèles robustes comme les réseaux de neurones.
- Absence de test sur les résultats : contrairement aux méthodes statistiques classiques, les réseaux de neurones n'ont pas de tests permettant de tester les résultats obtenus.
- Absence de critère mesurant la non linéarité : on ne dispose pas de critère stable pour savoir le type de relation entre la variable dépendante est la variable indépendante.

Conclusion :

Le prévisionniste est toujours condamné à se tromper, son but est d'obtenir l'erreur la moins coûteuse possible en moyenne.

A l'aide d'une série des produits pétroliers comportant des données mensuels, nous avons tenté de construire un modèle de type réseau de neurone (boite noire). Pour ce faire, nous avons choisi le critère de performance prévisionnelle le plus utilisé qui consiste à minimiser la moyenne des écarts d'erreurs. Ce modèle se distingue par sa flexibilité, sa facilité de conception.

Parallèlement à ces atouts, nous avons relevé quelques faiblesses des réseaux de neurone. L'absence de test sur les résultats afin de pouvoir faire la vérification ainsi que les interprétations possibles, l'intervention de l'homme pour régler les paramètres par des valeurs qui résultent de plusieurs expériences. De plus, nous retrouvons le problème de l'architecture optimale qui nécessite plusieurs essais pour pouvoir la déterminer.

Conclusion générale

Nous nous sommes donc assignés, tout au long de notre travail, comme objectif de concevoir un modèle de prévision mieux adapté à la complexité du phénomène. Ceci a été effectué à travers un outil puissant de traitement des données : Les réseaux de neurones artificiels. Nous avons par la suite comparé ces résultats avec ceux obtenus par le biais d'un modèle linéaire prévisionnel avec la méthodologie de Box & Jenkins.

Dans le premier chapitre, nous nous sommes intéressés aux réseaux de neurones, à travers les principes de base qui leur sont associés pour mieux nous familiariser avec ce domaine. C'est en effet, par cette approche d'intelligence artificielle, que nous avons conçu notre second modèle de prévisions.
Ainsi, au chapitre II, a été détaillée la méthode de Box & Jenkins, jugée comme étant la méthode classique la plus appropriée a la série chronologique que nous devons traiter. Dans ce chapitre, nous avons abordés les différents concepts et définitions appropriés pour cette démarche. C'est dans cette optique que nous y avons détaillé les modèles ARMA associés a la méthodologie de Box & Jenkins.

Toute modélisation a des fins de prévision passe par plusieurs étapes. Pour cela, un prétraitement de cette série a été effectué lors du chapitre 3.
Par ailleurs, L'estimation des paramètres d'un réseau de neurones s'effectue à l'aide de l'algorithme de rétropropagation du gradient en utilisant la première partie des données normalisée. Cet algorithme consiste à modifier les poids initialisés aléatoirement des connexions dans le but de la minimisation de la moyenne des écarts d'erreur. C'est le critère mesurant la performance prévisionnelle.

En ce qui concerne notre expérience pratique, nous avons construit un modèle de prévision de type réseau de neurones multicouches possédant un neurone dans la couche d'entrée, deux neurones à la première couche, un neurone à la deuxième couche cachée et un neurone à la couche de sortie dont l'architecture résulte de la recherche par algorithmes de rétropropagation à l'aide des données relatives aux produits pétroliers.

principaux résultats auxquels on a abouti après notre expérience pratique sont :

- Une prévision dont la moyenne des écarts des erreurs est minimale.
- De bons résultats sur une base de données bien définit.
- Une efficacité des algorithmes en tant qu'un outil permettant la recherche d'une solution dans un espace de grande taille, en particulier pour les réseaux de neurones.
- Le réseau de neurones privilégie les données car le modèle qu'on a construit est peut être propre à nos données.

Parallèlement. On a constaté quelques limites telle que :

- I' intervention de l'utilisateur pou régler, manuellement les paramètres d'apprentissage.
- L'absence de tests permettant l'évaluation des résultats des réseaux de neurones.
- Les réseaux de neurones sont faibles en marche explicatifs du modèle.

En résumé, notre travail nous a permis de mettre en titre la principale différence qui existe entre la prévision à l'aide d'un modèle spécifié à priori et celle effectuée avec un modèle neuronal déterminé à partir de la série des données disponibles.

Nous noterons en définitive que ce modeste travail pourrait être amélioré par :

- Le choix d'une série explicative de nature différente à la série à prévoir pourvu qu'elles soient corrélées (on pourra appliquer une analyse en composante principale pour choisir la série la plus corrélée à la série à prévoir).
- La comparaison entre les résultats d'un réseau multicouches et ceux d'un réseau récurrent simple dans le cadre de la prévision des séries temporelles.

En ce qui nous concerne, cette étude nous a permis d'enrichir nos connaissances tant sur le plan théorique que sur la réalité du domaine de la prévision.
Les réseaux de neurones représentent, en effet, une alternative très prometteuse pour I' analyse des séries chronologiques. De plus, au vu des structures de réseaux et des procédures d'apprentissage utilisées lors de notre étude, nous pouvons penser que ce domaine présente un fort potentiel d'exploration. Ces outils puissants n'ont pas atteint leur plein développement.

Annexes

Annexe 01 : Tableau de données

Tableau. 01.1 - Données des produits pétroliers importés mensuellement au Port d'Alger

janv-96	67333	**janv-00**	65937	**janv-04**	147061
févr-96	53983	**févr-00**	60496	**févr-04**	132743
mars-96	54663	**mars-00**	66554	**mars-04**	152215
avr-96	78024	**avr-00**	80064	**avr-04**	150473
mai-96	72082	**mai-00**	81359	**mai-04**	172006
juin-96	61012	**juin-00**	57547	**juin-04**	182476
juil-96	43066	**juil-00**	57614	**juil-04**	196446
août-96	54020	**août-00**	71205	**août-04**	191081
sept-96	45169	**sept-00**	55492	**sept-04**	215097
oct-96	61114	**oct-00**	97151	**oct-04**	176481
nov-96	54212	**nov-00**	79410	**nov-04**	209232
déc-96	66891	**déc-00**	77815	**déc-04**	179596
janv-97	**58869**	janv-01	**86055**	janv-05	**179541**
févr-97	**43107**	févr-01	**56895**	févr-05	**195968**
mars-97	**54507**	mars-01	**74145**	mars-05	**216368**
avr-97	**46108**	avr-01	**91441**	avr-05	**202940**
mai-97	**81062**	mai-01	**97382**	mai-05	**215918**
juin-97	**44209**	juin-01	**74248**	juin-05	**210187**
juil-97	**55644**	juil-01	**79782**	juil-05	**206912**
août-97	**71177**	août-01	**86186**	août-05	**213692**
sept-97	**61991**	sept-01	**60090**	sept-05	**189462**
oct-97	**70731**	oct-01	**84758**	oct-05	**172338**
nov-97	**86272**	nov-01	**79488**	nov-05	**186778**
déc-97	**65727**	déc-01	**85085**	déc-05	**176152**
janv-98	82696	**janv-02**	101975	**janv-06**	180895
févr-98	52302	**févr-02**	111603	**févr-06**	179726
mars-98	65452	**mars-02**	106514	**mars-06**	187070
avr-98	69708	**avr-02**	107569	**avr-06**	176443
mai-98	93965	**mai-02**	88921	**mai-06**	198975
juin-98	85113	**juin-02**	125559	**juin-06**	188297
juil-98	54516	**juil-02**	115455	**juil-06**	182990
août-98	59117	**août-02**	117717	**août-06**	176159
sept-98	53131	**sept-02**	115064	**sept-06**	160998
oct-98	53940	**oct-02**	135405	**oct-06**	193259
nov-98	60787	**nov-02**	134831	**nov-06**	179032
déc-98	69838	**déc-02**	141134	**déc-06**	150280
janv-99	**78808**	janv-03	**131270**	janv-07	**199010**
févr-99	**59145**	févr-03	**139873**	févr-07	**178341**
mars-99	**49083**	mars-03	**125350**	mars-07	**171239**
avr-99	**48256**	avr-03	**118670**	avr-07	**161004**
mai-99	**63140**	mai-03	**120460**	mai-07	**169764**
juin-99	**48674**	juin-03	**116682**	juin-07	**167013**
juil-99	**48798**	juil-03	**136818**	juil-07	**170517**
août-99	**54429**	août-03	**122225**	août-07	**163702**
sept-99	**55492**	sept-03	**118535**	sept-07	**149431**
oct-99	**48920**	oct-03	**128332**	oct-07	**186031**
nov-99	**75061**	nov-03	**117244**	nov-07	**169641**
déc-99	**76115**	déc-03	**134449**	déc-07	**138462**

Tableau. 01.2 La base de test

In 01	In 02	In 03	In 04	In 05	Cible
67333	58869	82696	78808	65937	86055
53983	43107	52302	59145	60496	56895
54663	54507	65452	49083	66554	74145
78024	46108	69708	48256	80064	91441
72082	81062	93965	63140	81359	97382
61012	44209	85113	48674	57547	74248
43066	55644	54516	48798	57614	79782
54020	71177	59117	54429	71205	86186
45169	61991	53131	55492	55492	60090
61114	70731	53940	48920	97151	84758
54212	86272	60787	75061	79410	79488
66891	65727	69838	76115	77815	85085

Tableau. 01.3 La base de validation

In 01	In 02	In 03	In 04	In 05	Cible
101975	131270	147061	179541	180895	199010
111603	139873	132743	195968	179726	178341
106514	125350	152215	216368	187070	171239
107569	118670	150473	202940	176443	161004
88921	120460	172006	215918	198975	169764
125559	116682	182476	210187	188297	167013
115455	136818	196446	206912	182990	170517
117717	122225	191081	213692	176159	163702
115064	118535	215097	189462	160998	149431
135405	128332	176481	172338	193259	186031
134831	117244	209232	186778	179032	169641
141134	134449	179596	176152	150280	138462

Annexe 02 : Tableau de Calcule des Critères de Performances

Tableau. 02.1 La phase d'Apprentissage.

cible	out	Error	Error^2	Error absolue	Error absolue / cible

86055	87011,1835	-956,183548	914286,977	956,183548	0,01111131
56895	57014,2715	-119,271514	14225,6941	119,271514	0,00209634
74145	76947,0788	-2802,07884	7851645,81	2802,07884	0,03779188
91441	92441,8499	-1000,84993	1001700,57	1000,84992	0,01094531
97382	96019,6598	1362,3402	1855970,83	1362,3402	0,01398965
74248	75037,8259	-789,825899	623824,951	789,82589	0,01063767
79782	79799,6183	-17,61831	310,404847	17,61831	0,00022083
86186	86003,4258	182,574235	33333,3513	182,574235	0,00211837
60090	61141,9162	-1051,9162	1106527,7	1051,9162	0,01750568
84758	85053,2732	-295,273238	87186,2851	295,273238	0,00348372
79488	80959,8629	-1471,86291	2166380,42	1471,8629	0,01851679
85085	86441,5925	-1356,59247	1840343,13	1356,5924	0,01594397
///////	//////////	**********	**S=17495736,1**	**S=11406,3872**	**S=0,14436153**
**********	**********	**********	**MSE=1457978,01**	**MAE=950,53**	**MAPE=*1,2%***

Tableau. 02.2 La phase de validation

cible	Out (Prévisions)	Error	Error^2	Error absolue	Error absolue / cible
199010	183725,03	15284,97	233630308	15284,97	0,07680503
178341	175085,936	3255,06408	10595442,2	3255,06408	0,01825191
171239	147827,213	23411,7875	548111793	23411,7875	0,13671995
161004	144016,254	16987,7462	288583522	16987,7462	0,10551133
169764	146822,859	22941,1413	526295966	22941,1413	0,13513549
167013	152046,785	14966,2154	223987604	14966,2154	0,08961108
170517	181972,106	-11455,1062	131219458	11455,1061	0,06717867
163702	153674,5	10027,4995	100550747	10027,4995	0,06125459
149431	150316,959	-885,958597	784922,636	885,958597	0,00592888
186031	190139,018	-4108,0177	16875809,4	4108,0177	0,02208244

169641	159300,889	10340,1114	106917904	10340,1114	0,0609529
138462	141564,291	-3102,29099	9624209,37	3102,29099	0,02240536
/////////	//////////	/////////	**S= 2197177684**	**S=136765,909**	**S=0,80183765**
//////////	//////////////////	//////////////////	**MSE=183098140**	**MAE= 11397,15**	**MAPE =*6,6%***

Calcule du MSE (modéle linéaire)

Prévisions 2007	**Année 2007**	**Error**	**Error^2**
242604	199010	-43594	1900436836
179726	178341	-1385	1918225
197070	171239	-25831	667240561
146443	161004	14561	212022721
198975	169764	-29211	853282521
178297	167013	-11284	127328656
182990	170517	-12473	155575729
176159	163702	-12457	155176849
140998	149431	8433	71115489
193259	186031	-7228	52243984
179032	169641	-9391	88190881
100280	138462	38182	1457865124
			S=5742397576
			MSE=478533131

Bibliographie

❖ Références Bibliographiques

1. Bourbonnais .régis : « Econométrie », 4ème édition Dunod, Paris 2002.

2. Bourbonnais régis – Michel Terraza : « Analyse des séries temporelles », édition Dunod, Paris 2004.

3. Bourbonnais régis - Usunier Jean Claude : « Prévision des ventes », 3ème édition Economica, Paris 2001.

4. Bollerslev, T. et E. Ghysels, « Periodic AutoRegressive Conditional Heteroscedasticity », Journal of Business and Economics, 14(2) :139-52 (1996).

5. Bollerslev, T., R.Y. Chou et K. F. Kroner, « ARCH modeling in finance: A review of the theory and empirical evidence », Journal of Economics, 52:5-59.(1992).

6. Boullerslev, T. (1990), «Generalized Auto Regressive Conditional Heterosedasticity

7. Charpentier Arthur : « Cours de séries temporelles théorie et applications », édition université Paris DAUPHINE. Larousse illustré, édition librairie Larousse 1984.

8. Capera Philippe - Bernard van Cutsen « Méthodes et modèles en statistique non paramétrique », édition Dunod, Paris 1988.

9. *Dominick, (1986),* «Econométrie et statistique appliquée *», Série Schaum.*

10. DROSBERKE Jean jacques, BERNARD FICHET, PHILIPPE TASSI « MODELISATION ARCH », théorie statistique et applications dans le domaine de la finance, *(1990).*

11. *Gourieroux C., Montfort A. :« Séries temporelles et modèles dynamiques, Economica* », Paris 1999.

12. Hamilton James.D:«Time series analysis », Princeton University press, New Jersey 1994.

13. Mignon Valérie – Sandrine Lardig : « Econométrie des séries temporelles macro-économiques et financières », édition économica, Paris 2002.

14. *Wong, H et W. K. Li: «* On multivariate conditional heteroscedasticity model *», Biometrika, 8*4*, (1997).*

15. J.M.RENDERS «Algorithmes génétiques et réseaux de neurones». Hermes 1995.

16. S-THIRIA, Y-LECHEVALLIER, O-GUSCUEL, S-CANU <Statistique et méthodes neuronales>, DUNOD 1997.

17. Jean-François JODOUIN : <<Les réseaux de neurones : principes et définitions>, Hermes 1994.

18. Jean-Louis AMAT, Gérard YAHIAOUI :<< Techniques avancées pour le traitement de l'information : réseaux de neurones, logique floue, algorithmes génétiques>>. : Cépadués- éditions 1996.

19. J-HEROLAUT, C-JUTTEN <<Réseaux neuronaux et traitement du signal>>: Hermes 1994.

20. Jean-Michel RENDERS <<Algorithmes génétiques et réseaux de neurones>>. Hermes 1995.

21. J.F.Joduin <<Les réseaux neuromimétiques "Modèle et applications">>, Hermes 1994.

22. Hafedh El Ayech et Abdelwahed Trabelsi << Les réseaux de neurones artificiels pour la prévision du trafic aérien de passagers. Construction et comparaison avec l'analyse Box-Jenkins>> Institut Supérieur de Gestion de Tunis 2003.

23. M. Faignart et C. Hemptinne <<Data Mining de données financières>>, Faculté Polytechnique de Mons 2005-2006.

24. Julian Faraway and Chris Chatfield<<Time series forcasting with neural network >>: University of michigan, ann arbor, USA, University of Bath, UK, 1997.

25. Kaastra&Boyd << Designing a Neural Network for Forecasting Financial and economic Time Series >>: University of Manitoba Canada 1995.

26. M.Verleysen, E. de Bodt, A. Lendasse <<Forecasting financial time series through intrinsic dimension estimation and non linear data projection>>. Alicante (Spain), June 2-4, 1999, Springer.

27. Ruey Hwa Loh, R[1].H. - <<Time Series Forecast With Neural Network >> University of Queensland. (2003).

28. Yuehui Chena,Bo Yanga,b, Jiwen Donga Time-series prediction using a local linear wavelet neural network>: February 250022 Jinan, Wuhan, PR China 2005.

29. G. Dreyfus. «Réseaux de neurones méthodologie et application ».2^{éme} édition EYROLLES, Paris.

30. J. Rynkiewicz, M. Cottrell, M. Mangeas, J.F. Yao << Modèles de réseaux de neurones pour l'analyse des séries temporelles ou la régression : Estimation, Identification, Méthode d'élagage SSM >> SAMOS, Université de Paris1 2001.

31. R.Mekki << Complexité des algorithmes dans la modélisation stochastique par les réseaux neurones >>, USTHB: Université des Sciences et de la Technologie Houari Boumediene, Alger 2008.

❖ Références Webographiques

- Chapitre 3 : *Econométrie Appliquée Séries Temporelles.* Livre de Christophe HURLIN. (Www. dauphine.fr)
- Michel LUBRANO (Septembre 2004) *Cours Séries Temporelles.*
- (Www. Vcharite.univ-mrs.fr).
- **(**Www.spatial-econometrics.com)
- [1] http://fr.wikipedia.org/wiki/Neurone.htm
- [2] http://bebene.ifrance.com/inte.htm
- http://www.cict.fr/cictJpersonel/stpirre/réseax-neuronaux/nodel0.html