**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**INSTITUT SUPERIEUR POLYTECHNIQUE DE MADAGASCAR**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

****

Mini projet RESEAUX DE NEURONES ARTIFICIELS Octobre 2018

**PREDICTION ET MODELISATION DE SERIES TEMPORELLES PAR RESEAUX DE NEURONES ARTIFICIELS MULTICOUCHES**

Présentée par *:* **Mlle RAHARITSIRESY Dina Miharisoa**

*Option : Informatique de gestion, génie logiciel et intelligence artificielle (IGGLIA 4)*

Enseignant : Professeur RABOANARY Rolland

Déposé le mercredi 24 octobre 2018 au secrétariat de l'ISPM

Année universitaire : 2017 – 2018

Table des matières

[REMERCIEMENT 3](#_Toc528009336)

[AVANT PROPOS 4](#_Toc528009337)

[INTRODUCTION 5](#_Toc528009338)

[ Historique 5](#_Toc528009339)

[ Réseau de neurone 6](#_Toc528009340)

[ Série temporelle 7](#_Toc528009341)

[PARTIE I - ANALYSES ET METHODOLOGIES 8](#_Toc528009342)

[ Présentation de la problématique 8](#_Toc528009343)

[ Méthodologies 8](#_Toc528009344)

[PARTIE II - OUTILS UTILISES, IMPLEMENTATION ET RESULTAT 13](#_Toc528009345)

[ Outils utilisés 13](#_Toc528009346)

[ Extrait de code source 13](#_Toc528009347)

[ Résultat et présentation de l'application 16](#_Toc528009348)

[CONCLUSION 17](#_Toc528009349)

[RESUME 18](#_Toc528009350)

# REMERCIEMENT

Nous tenons à remercier Dieu Tout Puissant, car sans lui, tous nos efforts auraient été vains.

Aussi, ce mini-projet n’aurait pu être réalisé sans l’aide, l’assistance, le conseil, la compréhension, la collaboration et l’appui de toutes les personnes à qui nous adressons nos vifs remerciements entre autres :

* Le Professeur **RABOANARY Julien Amédée**, Fondateur et Recteur de l’Institut Supérieur Polytechnique de Madagascar.
* Le Professeur **RABOANARY Rolland**, notre formateur du cours Réseaux de neurones artificiels durant cette année universitaire 2014-2015.
* A tout le corps administratifs et professoraux de l’ISPM, de nous avoir éduqués dans le bon sens.

Enfin, pour réaliser ce document et le travail qu'il présente, nous avons largement bénéficié de l'aide de nombreuses personnes. Nous tenons à les remercier sincèrement.

# AVANT PROPOS

Depuis l’année universitaire 2008-2009, l’Institut Supérieur Polytechnique de Madagascar (ISPM) a adopté le système éducatif LMD (Licence Master Doctorat) et incite donc ses étudiants à effectuer des mini-projets dans le but de favoriser le travail personnel des étudiants.

De ce fait, nous avons effectué un mini-projet dans la matière Réseaux de Neurones Artificiels (R.N.A) enseigné par le professeur RABOANARY Rolland.

Ce test nous a été enrichissant dans la mesure où nous avons pu, élargir nos connaissances en classe et faire des découvertes en ayant fait beaucoup de recherche concernant le sujet.

# INTRODUCTION

## Historique

L’intelligence Artificielle, branche de l’informatique fondamentale s’est développée avec pour objectif la simulation des comportements du cerveau humain. Les premières tentatives de modélisation du cerveau sont anciennes et précèdent même l’ère informatique.

C’est en 1943 que Mc Culloch (neuro-physiologiste) et Pitts (logicien) ont proposé les premières notions de neurone formel. Ce concept fut ensuite mis en réseau avec une couche d’entrée et une sortie par Rosen blatt en 1959 pour simuler le fonctionnement rétinien et tacher de reconnaître des formes. C’est l’origine du perceptron. Cette approche dite connexionniste a atteint ses limites technologiques, compte tenu de la puissance de calcul de l’époque, mais aussi théoriques au début des années 70.

L’approche connexionniste à connaissance répartie a alors été supplantée par une approche symbolique qui promouvait les systèmes experts à connaissance localisée dont l’objectif était d’automatiser le principe de l’expertise humaine en associant trois concepts :

* une base de connaissance dans laquelle sont regroupées les connaissances d’experts humains sous forme de propositions logiques élémentaires ou plus élaborées en utilisant des quantificateurs (logique du premier ordre).
* une base de faits contenant les observations du cas à traiter comme, par exemple, des résultats d’examens, d’analyses de sang, de salive pour des applications biomédicales de choix d'antibiotique.
* un moteur d'inférence chargé d'appliquer les règles expertes sur la base de faits afin d'en déduire de nouveaux faits jusqu'à la réalisation d'un objectif comme le choix du traitement d'un infection bactérienne.

Au début des années 80, l'essor technologique et quelques avancées théoriques comme l'estimation du gradient par rétro-propagation de l'erreur, ont permis de relancer l'approche connexionniste. Celle-ci a connu au début des années 90, un développement considérable si l'on considère le nombre de publications et congrès qui lui ont été consacrés mais aussi les domaines d'applications très divers où elle apparaît.

Remis en veilleuse depuis le milieu des années 90 au profit d'autres algorithmes d'apprentissage machine, les réseaux de neurones connaissent un regain d'intérêt et même un énorme battage médiatique sous l'appellation d'apprentissage profond (deep learning).

## Réseau de neurone

Un réseau de neurone est l'association, en un graphe plus ou moins complexe, d'objets élémentaires.

De façon réductrice, un neurone biologique est une cellule qui se caractérise par :

* des synapses, les points de connexion avec les autres neurones
* des dendrites ou entrées du neurone
* les axones ou sorties du neurone vers d'autres neurones ou fibres musculaires
* le noyau qui active les sorties en fonction des stimulations d'entrée

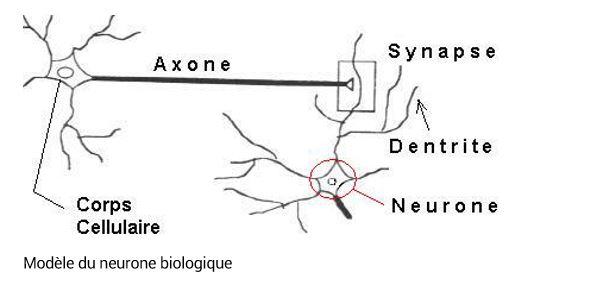


Figure 1 : Modèle d'un neurone biologique

Et un réseau de neurone artificiel se construit sur une architecture semblable, en première approximation, à celle du cerveau humain. Il reçoit les informations sur une couche réceptrice de "neurones" ou couches d'entrée, traite ces informations avec ou sans l'aide d'une ou plusieurs couches "cachées" contenant un ou plusieurs nœuds et produit un signal (ou plusieurs) de sortie. Chaque nœud, qu'il appartienne à la première couche (réceptrice), aux couches cachées ou à la couche de sortie, est lié aux autres neurones par des connexions (similaires aux synapses du cerveau) auxquelles sont affectés des poids (eux-mêmes assimilables aux potentiels synaptiques).

Il existe deux grands types d'apprentissage des réseaux dont le choix d'utilisation se fait en fonction des valeurs connues :

* L'apprentissage non supervisé : le réseau combine l'information d'entrée et l'analyse sans référence à une valeur de sortie.
* L'apprentissage supervisé suppose par contre une sortie connue. Chaque information d'entrée est couplée à une valeur de sortie et le réseau cherche à approcher la valeur attendue.

## Série temporelle

Une série temporelle est une suite de valeurs numériques représentant l'évolution d'une quantité spécifique au cours du temps.

La prévision des séries temporelles est un problème qui a été abordé depuis longtemps. On en trouve des applications dans de nombreux domaines, par exemple l’économie, la météorologie, la médecine, la communication, le traitement de la parole, etc. Théoriquement, la prévision des séries temporelles nécessite de modéliser le système qui a généré les données de la série. En disposant d’un système d’équations mathématiques et déterministes et connaissant les conditions initiales, il serait possible de prévoir l’évolution du système.

Ce type d'apprentissage correspond à des applications où l'on cherche à prédire une variable dépendante, que ce soit sous forme continue ou binaire (prix, taux d'intérêt, probabilité d'occurrence d'un événement, etc.).

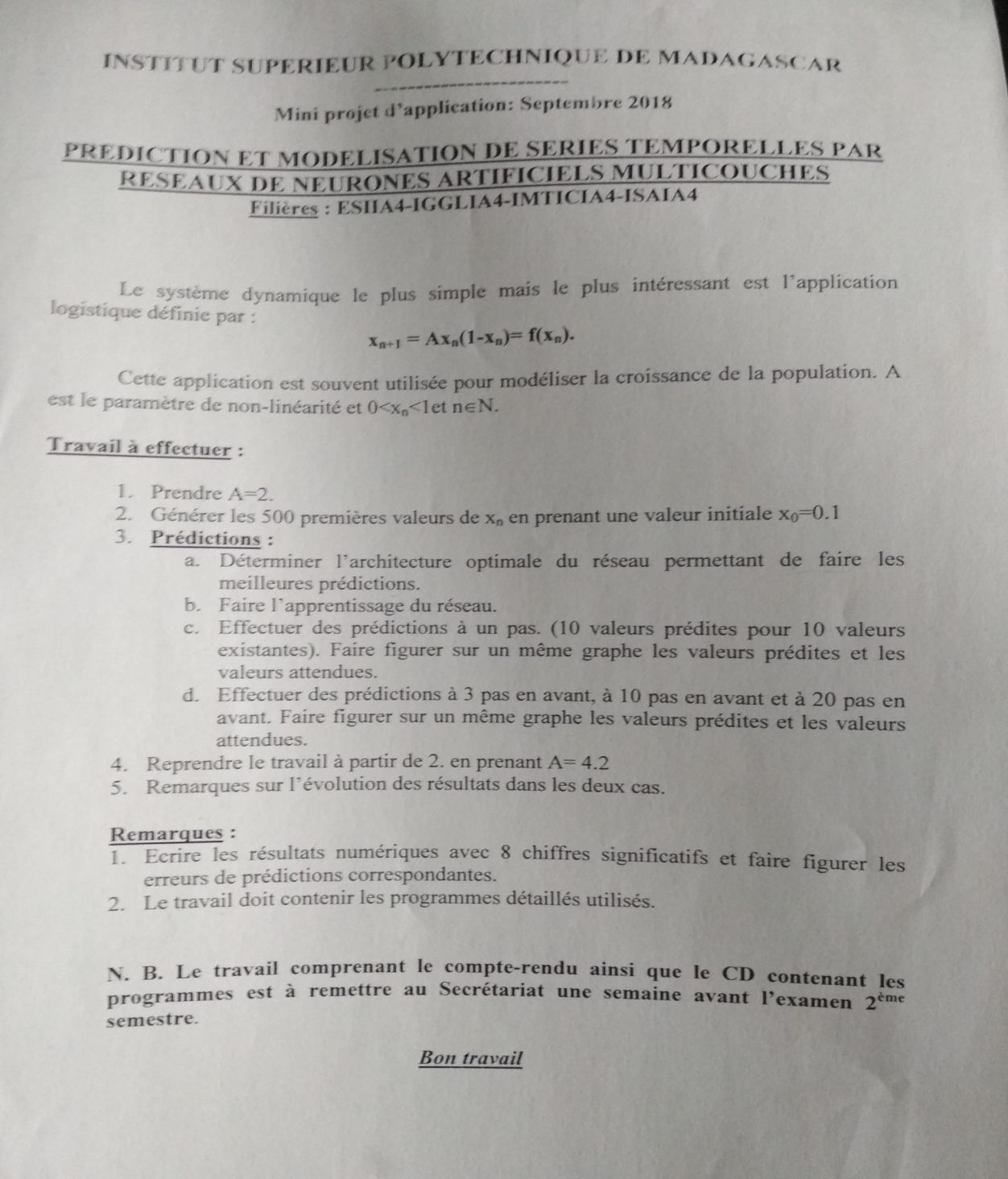
Dans cet ouvrage, nous allons vous présenter la prédiction d'une série temporelle par l'utilisation d'un réseau de neurone multicouche.

# PARTIE I - ANALYSES ET METHODOLOGIES

### Présentation de la problématique

Dans ce devoir, nous avons comme problème, la prédiction et modélisation de série temporelle par réseaux de neurones artificiels multicouches.

Cette série est obtenue par la suite



### Méthodologies

#### Détermination de l'architecture du réseau optimal

Pour un problème donné, il est nécessaire d'avoir un réseau optimal car trop peu de paramètres perdraient les informations de la série et trop de paramètres consommeraient beaucoup de temps.

Les méthodes que nous avons utiliser pour trouver notre architecture optimale sont les suivantes :

1. Nombre de couches

En fait, un réseau de neurone artificiel implémente un ensemble de fonction F où :

F : {x} y = F {x}

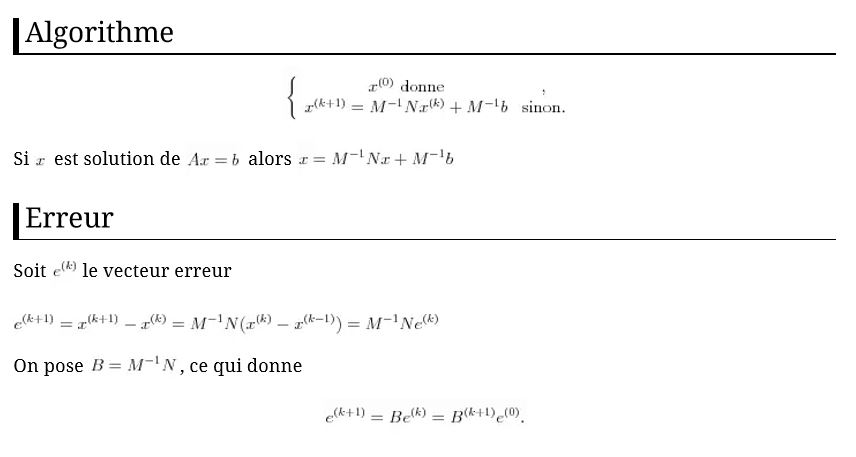
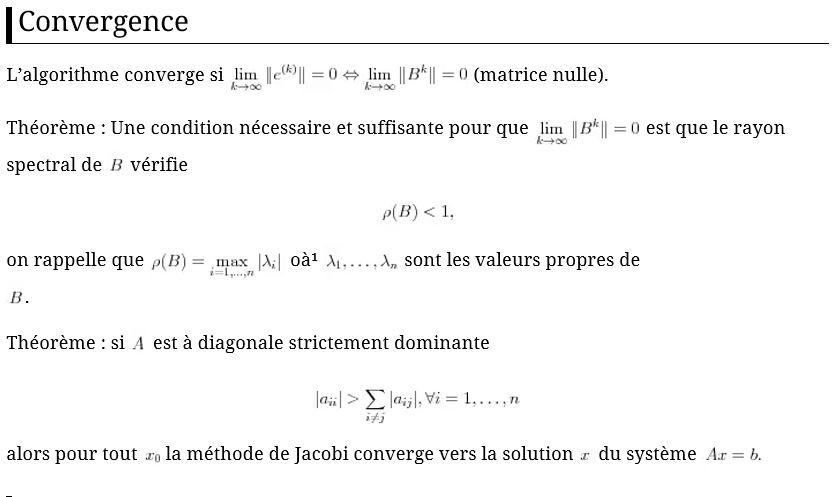
Entre la couche d'entrée et la couche cachée et entre les couches cachées elles-mêmes, on utilise la fonction sigmoïde g. Mais entre la dernière couche cachée et la couche de sortie, on utilise la fonction identité.

Afin de connaitre le nombre de couches nécessaires, on utilise le théorème de **Cybenko** qui montre qu'une seule couche cachée est suffisante pour approcher toutes fonctions continues.

1. Nombre d'unités d'entrées

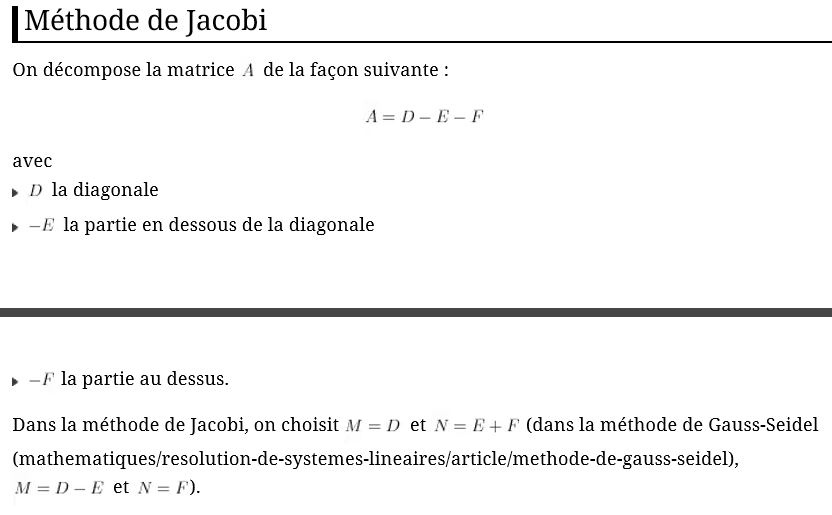
Le nombre d'unités d'entrées dépend du problème à traiter et il est donné par l'algorithme de **Takens** qui fait appel au système dynamique et physique.

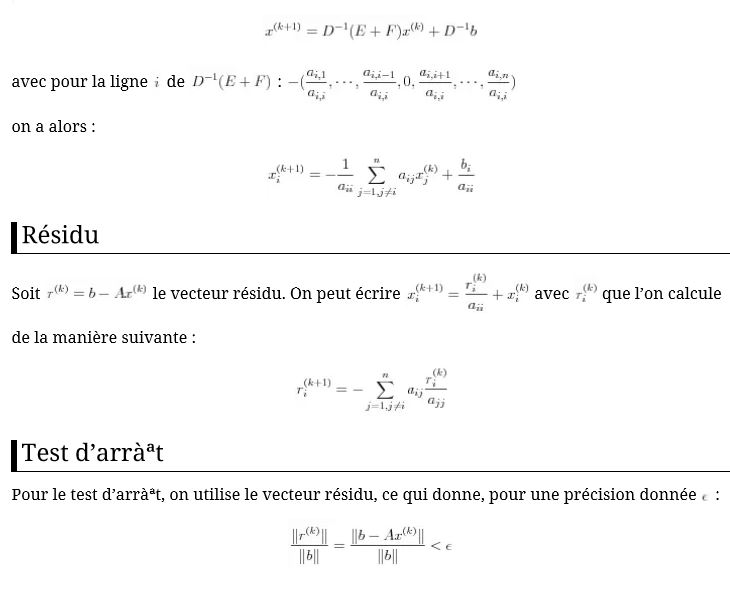
Algorithme de takens



Mais afin de pouvoir trouver les valeurs propres de la matrice de covariance dans l'algorithme de Takens, nous avons fait appel à la méthode itérative de Jacobi.

Algorithme de Jacobi





1. Nombre d'unité de sortie

Pour un problème de prédiction, ce qui est notre cas dans ce livre, nous avons besoin d'une seule unité de sortie.

1. Nombre d'unités cachées

De façon pratique, le nombre d'unité d'entrée et de sortie ayant été fixé ont fait varier la structure du réseau en donnant différentes valeurs au nombre d'unités cachées.

Par conséquent, le nombre d'unité caché du réseau optimal que l'on apportera, sera celui qui donnera l'erreur d'apprentissage la plus faible.

#### Apprentissage du réseau

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

Dans la majorité des algorithmes actuels, les variables modifiées pendant l'apprentissage sont les poids des connexions.

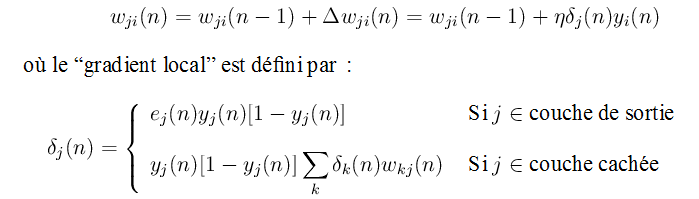
Au niveau des algorithmes d'apprentissage, il a été défini deux grandes classes : l'apprentissage supervisé et le non supervisé. Pour l'apprentissage supervisé, ce qui est notre cas dans ce mini projet, les exemples sont des couples (Entrée, Sortie associée).

Afin de réaliser l'apprentissage de notre modèle dans ce travail, nous avons employé l'algorithme de descente de gradient, qui est un algorithme itératif, ayant pour but d'ajuster les poids des connexions d'un réseau.

**Algorithme de Descente de gradient**

* Initialiser tous les poids à des petites valeurs aléatoires dans l'intervalle de [-0.5,0.5]
* Normaliser les donner d'entrainement
* Permuter aléatoirement les données d'entrainement
* Pour chaque donnée d'entrainement :

1. Calculer les sorties observées en propageant les entrées vers l'avant.
2. Ajuster les poids en rétro propageant l'erreur observée



Avec 0<n<1 représentant le taux d'apprentissage et yi(n) représentant soit la sortie du neurone *i* sur la couche précédente, si celui-ci existe, soit l'entrée *i* autrement.

* Répéter les étapes 3 et 4 jusqu'à un nombre maximum d'itérations ou jusqu'à ce que la racine de l'erreur quadratique moyenne soit inférieure à un certain seuil.

#### Prédiction de la série temporelle

Ici, la prévision des séries temporelles est réalisée par l'algorithme d'apprentissage supervisé. Cet apprentissage artificiel consiste à créer un modèle de prédiction à partir d'une base d'apprentissage comprenant les exemples d'entrée ainsi que les sorties désirées associées. Les paramètres du modèles vont ainsi s'adapter en comparant à chaque fois les sorties obtenues et les sorties désirées. Pour mesurer la performance, il est possible de calculer l'Erreur Quadratique Moyenne (EQM), en anglais Mean Square Error (MSE).

L'algorithme d'apprentissage permet de prédire une valeur cible étant donnée une ou des valeurs d'entrées.

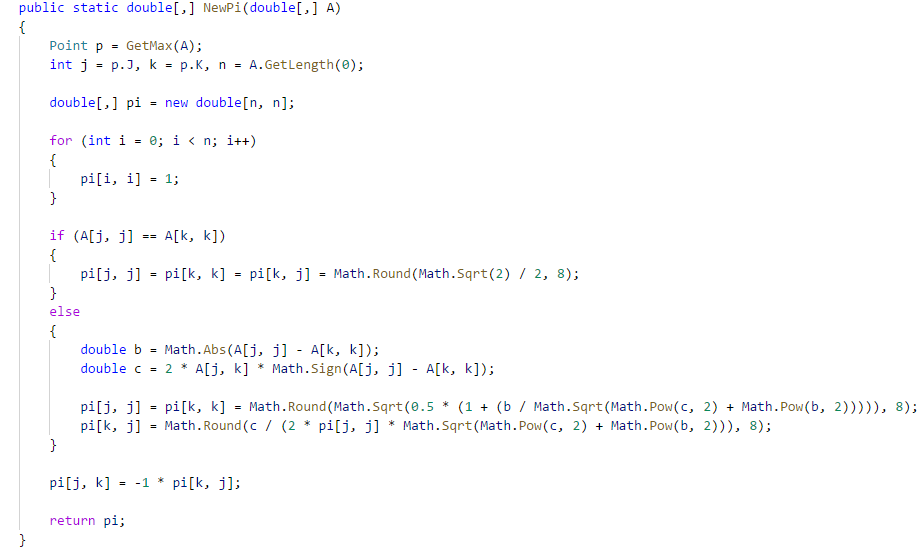
# PARTIE II - OUTILS UTILISES, IMPLEMENTATION ET RESULTAT

### Outils utilisés

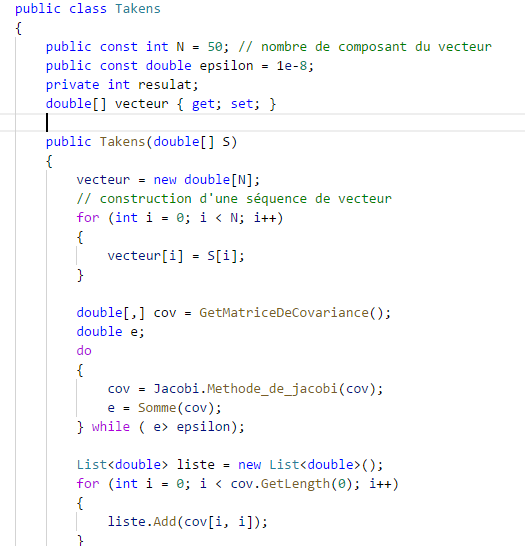
Pour la réalisation de ce mini projet, nous avons utilisé comme langage, le langage de programmation C# qui est un langage très puissant et enseigné dans notre institut. C'est aussi grâce à ce langage que nous avons pu élaborer les chartes graphiques permettant d'afficher les graphes montrant les résultats de notre travail.

### Extrait de code source

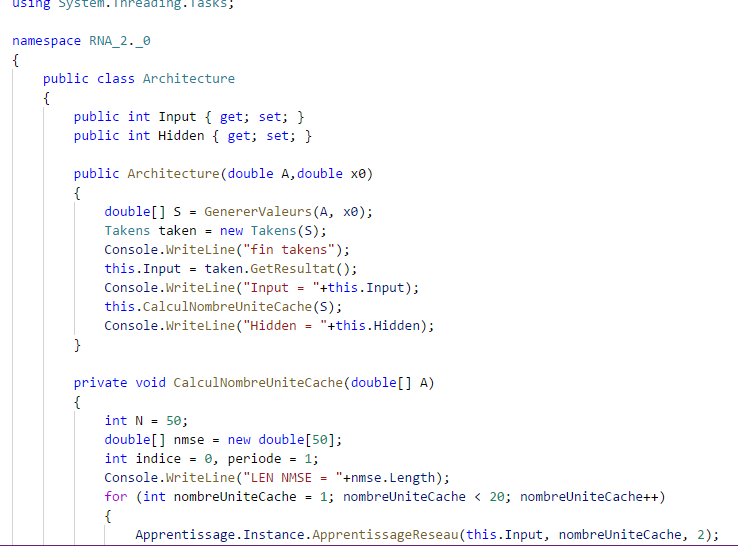
Extrait de l'implémentation de l'algorithme de Jacobi



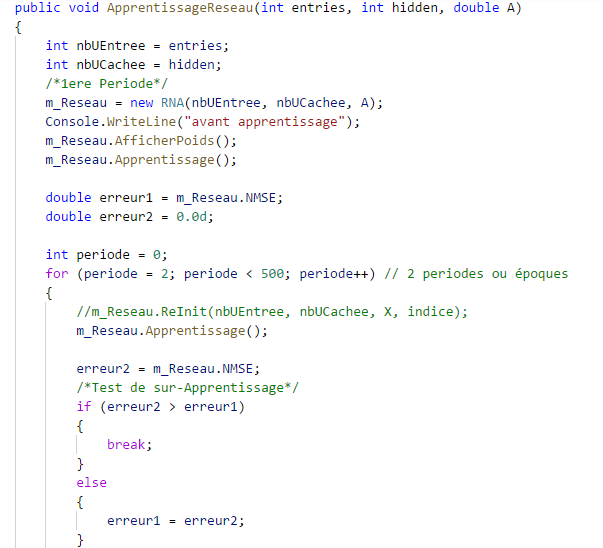
Extrait de code de l'algorithme de Takens



Extrait de code pour la définition de l'architecture optimal



Apprentissage du réseau



### Résultat et présentation de l'application

Tout d'abord, on arrive dans la page d'accueil de l'application, on y trouve l'aide sur l'utilisation du logiciel.

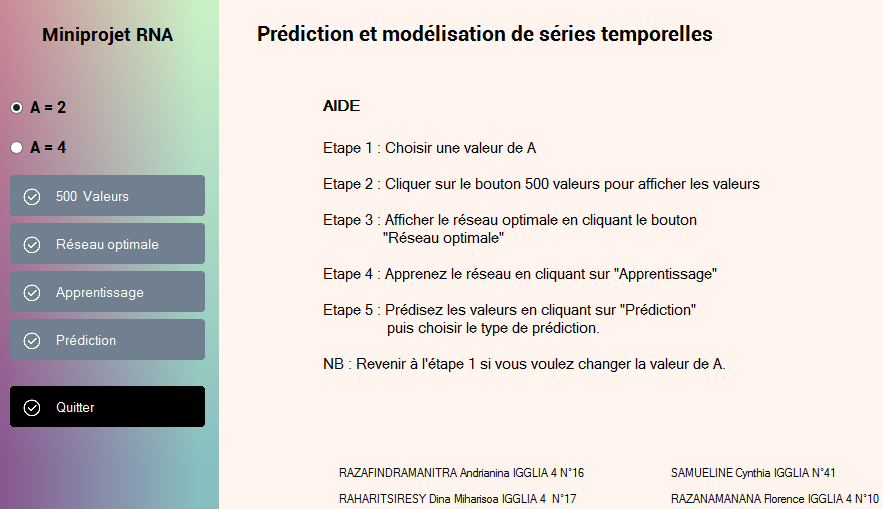


Figure 2 : Interface d'accueil du logiciel

On peut afficher la liste des 500 valeurs de notre série temporelle en cliquant sur le bouton "500 valeurs".

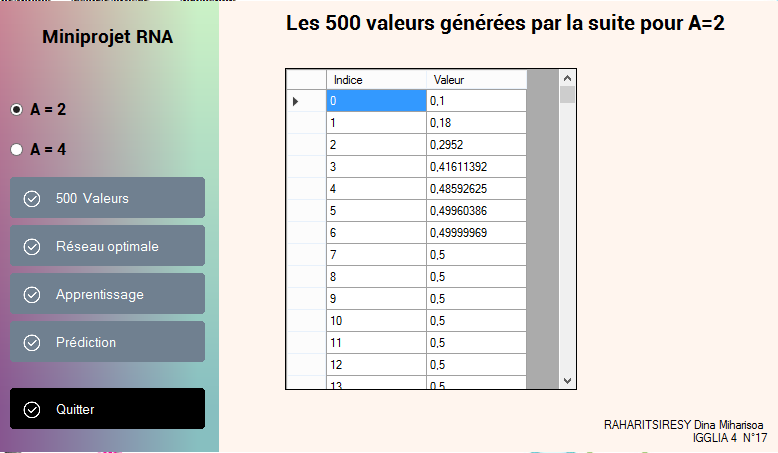


Figure : Affichage des 500 valeurs

Le bouton "Réseau optimal" permet de calculer puis afficher l'architecture optimale du réseau de neurone pour avoir une meilleure prédiction.

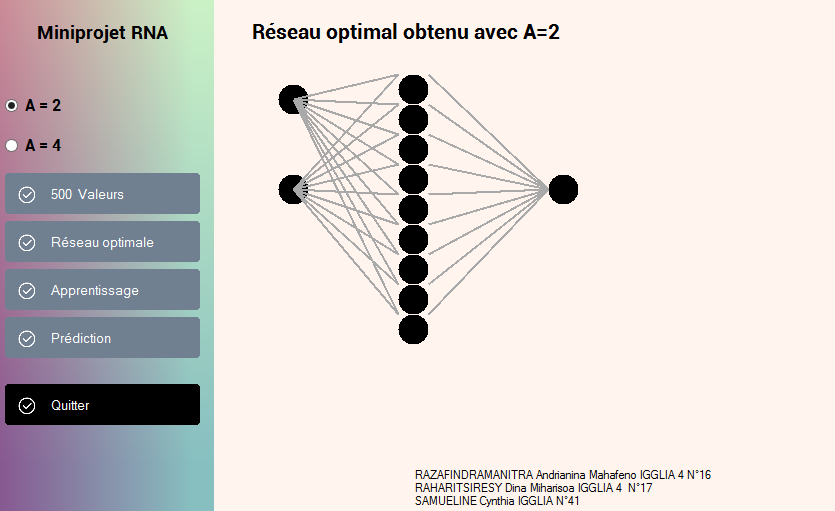


Figure 3 : Dessin de l'architecture optimale pour A=2

Comme on a obtenu le réseau optimal pour notre prédiction, on peut lancer l'apprentissage du réseau en cliquant su le bouton "Apprentissage", puis afficher les poids appropriés à notre réseau

.

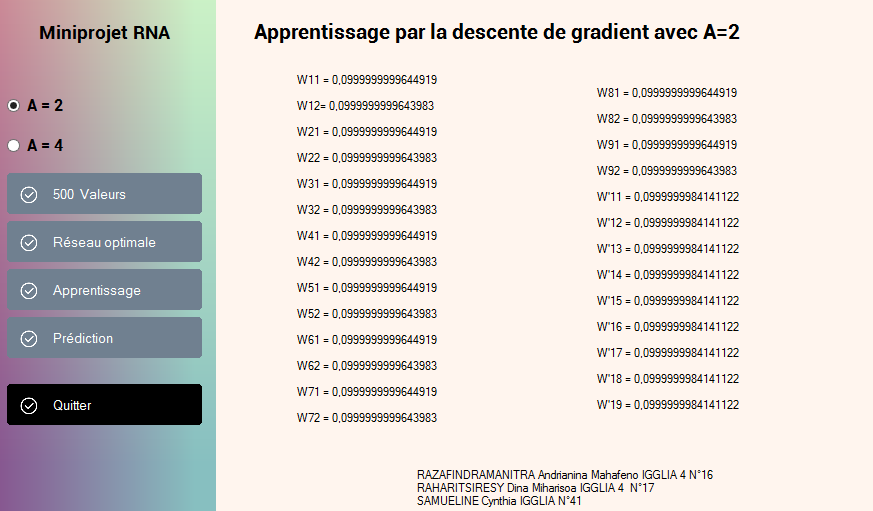


Figure 4 : Affichage des poids appropriés au réseau

Après la détermination du réseau optimal et l'obtention des poids appropriés, grâce aux méthodologies citées dans la partie II, nous pouvons maintenant procéder à la prédiction de notre série temporelle.

On fait notre choix de prédiction en sélectionnant une option parmi ceux présentées dans le menu déroulant, en couleur bleu, les courbes sur le repère s'actualisent automatiquement.

La courbe en jaune sur notre repère représente les valeurs attendues, et celle en noire montre les valeurs données par le réseau.

Enfin, on peut afficher le tableau des résultats contenant la sortie désirée, la sortie donnée par le réseau de neurone et l'écart entre ces deux sorties, en appuyant sur le bouton "Tableau résultat", en vert.

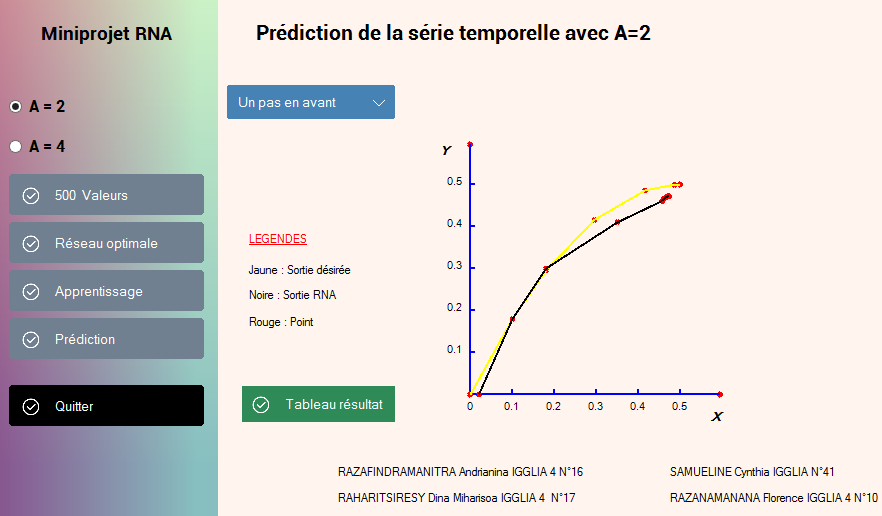


Figure 6 : Prédiction de la série temporelle

Après avoir cliqué le bouton "Tableau résultat", on peut voir la fenêtre ci-dessous, permettant de montrer les sorties désirées, les sorties données par notre réseau de neurones ainsi que les écarts entre ces sorties.

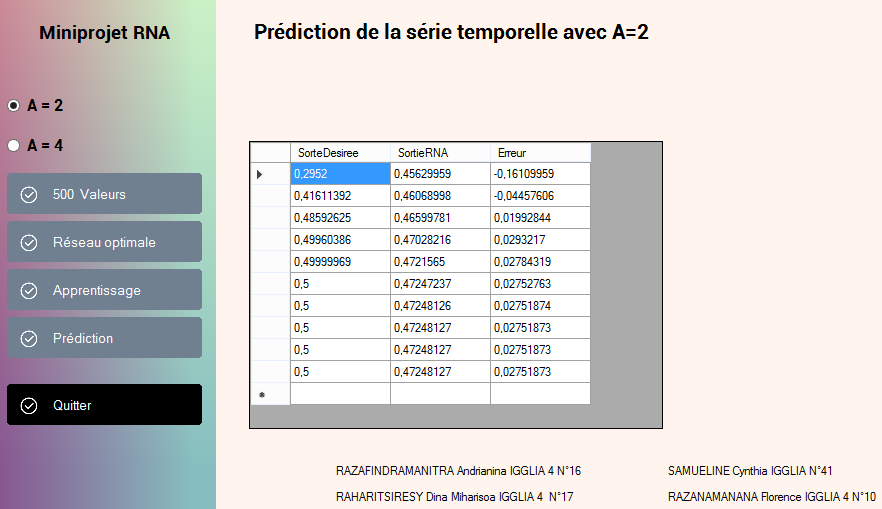


Figure : Tableau des résultats d'une prédiction

# CONCLUSION

Comme conclusion, nous avons pu voir dans ce présent livre la capacité des réseaux de neurones à prédire une série temporelle. Nombreuses sont les étapes passées ainsi que les algorithmes utilisés afin d'aboutir à ce fin. Comme solution à notre problématique, nous avons conçu un modèle avec l'architecture optimal, puis effectue l'apprentissage supervisé du réseau et enfin prédire les valeurs de la série temporelle en utilisant la prédiction à un et/ou plusieurs pas en avant. Nous remarquons que la prédiction avec A=2 est plus correcte que celle avec A=4. Mais dans les deux cas, les erreurs sont minimisées.

# RESUME

La prévision des séries temporelles est un problème qui est traité depuis de nombreuses années. On y trouve des applications dans différents domaines tels que : la finance, la médecine, le transport.

Dans cet ouvrage, on s’est intéressé aux méthodes appropriées pour la prédiction d'une série temporelle en utilisant les réseaux de neurones. On s’est également intéressé à l’intérêt des méthodes pour améliorer les performances des prédicteurs. Nous avons employé l’algorithme d’apprentissage supervisé des réseaux de neurones pour obtenir les poids appropriés du réseau.