

الجمهورية الجزائرية الديمقراطية الشعبية  
وزارة التعليم العالي والبحث العلمي

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
*Ministère de l'enseignement supérieur et de la recherche scientifique*



جامعة عمار تليجي - الأغواط -

Université Ammar Telidji - Laghouat

Faculté de Technologie

Département de GENIE CIVIL

**Mémoire de Master**

Présenté par :

**Khaled Chorfi**

En vue de l'obtention du diplôme de Master en Hydraulique

Option : Ressources en Eau

**THÈME :**

**MODELE DE PREVISION DE DEBIT A L'AIDE D'UNE  
APPROCHE D'INTELLIGENCE ARTIFICIELLE HYBRIDE  
CONNEXIONNISTE NEURO-GENETIQUE.**

Soutenu le 08/06/2015. publiquement devant le jury composé de :

Mr. Mouattah K.  
Mr. Chettih M.  
Mr. Belkhiri M.  
Mr. Tadj W.

Président  
Encadreur  
Examineur  
Examineur

Année Universitaire 2014/2015

## ملخص

أصبحت الشبكات العصبية الاصطناعية خلال سنوات فقط آليات قيمة في مجالات متعددة. إلا هناك صعوبات في تحديد مكوناتها. التي تؤثر بشكل مباشر على كفاءة تلك الشبكات العصبية. وبهدف تحسين كفاءة هذه الشبكات، فإن استعمال طرقا هجينة التي تعمل على دمج الاقتراب الاستقرائي، الرمزي مع اقتراب الذكاء الاصطناعي الترابطي. وتشمل هذه الأساليب الخوارزميات الجينية والتي لديها العديد من المزايا. وتستند هذه الخوارزميات على فكرة مستمدة من الانتقاء الطبيعي. تركز دراستنا على تطوير نظام عصبي وراثي يمكن من التنبؤ بالتدفق اليومي. ان النتائج التي تم الحصول عليها من ثلاثة أحواض مائية مختلفة مشجعة للغاية وتعد أفضل من النتائج التي حُصل عليها من الشبكات العصبية.

**كلمات البحث:** شبكة عصبية اصطناعية، الخوارزمية الجينية، والتنبؤ، التدفق

## Résumé

Les réseaux de neurones artificiels sont devenus pendant quelques années des mécanismes précieux dans des domaines très divers. Néanmoins ils présentent des difficultés dans la détermination de leurs composants. Ces composants influencent directement sur la fiabilité de ces réseaux neuronaux. Afin d'améliorer l'efficacité de ces réseaux, des méthodes hybrides qui intègrent les approches déductives, symboliques et évolutionnaires avec les approches de l'intelligence artificielle connexionniste sont très rapidement développées. Parmi ces méthodes les algorithmes génétiques qui présentent de nombreux avantages. Ces algorithmes sont basés sur l'idée dérivée de la sélection naturelle. Notre étude porte sur l'élaboration d'un système Neuro-génétique permettant la prévision des débits journaliers. Les résultats obtenus sur trois bassins versants différents sont très encourageants et meilleurs que les résultats obtenus par les réseaux de neurones.

**Mots clés :** Réseau de neurones artificiels, Algorithme génétique, Prévision, Débit.

## Abstract

Artificial neural networks have become for some years valuable mechanisms in a variety of areas. Nevertheless they present difficulties in determining their components. These components directly affect the reliability of these neural networks. To improve the efficiency of these networks, hybrid methods that integrate deductive approaches, symbolic with evolutionary approaches to connectionist artificial intelligence are rapidly developed. These methods include genetic algorithms have many advantages. These algorithms are based on the idea derived from natural selection. Our study focuses on the development of a neuro-genetic system for forecasting flow. The results obtained on three different watersheds are very encouraging and better than the results obtained by the neural networks.

**Keywords:** artificial neural network, genetic algorithm, forecasting, flow.

# Introduction

Les événements hydrologiques exceptionnels tels que les crues constituent l'un des plus importants risques naturels. Les dégâts qu'elles engendrent, les vies qu'elles emportent et la peur qu'elles suscitent ont poussé les chercheurs de s'en prémunir par différentes techniques. Les modèles pluie-débit sont des outils essentiels pour le calcul des crues de projet ou la prévision des crues à court terme. Malgré les nombreuses recherches qui leur ont été consacrées, les modèles restent d'utilisation difficile et incertaine. Une des difficultés réside dans le fait que les processus de la transformation pluie-débit sont ignorés ou mal représentés, faute de données ou d'expérimentations suffisantes.

Au cours de ces dernières années, la puissance des moyens de calcul ont permis le développement de toute une panoplie de méthodes permettant de modéliser le comportement global d'un système complexe. A ce titre, le formalisme des réseaux d'automates permet une description semi-quantitative des systèmes complexes naturels. Il est aussi à la base d'applications technologiques concernant l'intelligence artificielle. Par conséquent, la combinaison de raisonnement probabiliste, logique floue, réseaux de neurones et algorithmes évolutionnaires ainsi que certaines méthodes d'analyse forme ce que nous appelons communément : le « Soft Computing », et constitue une nouvelle technique pour la construction de systèmes hybrides intelligents capables de raisonnement et d'apprentissage dans un environnement incertain et imprécis.

Les systèmes hybrides sont des systèmes qui intègrent les approches de l'intelligence artificielle déductives, symboliques et évolutionnaires avec les approches de l'intelligence artificielle connexionniste.

L'objectif de notre travail est d'élaborer un système hybride permettant de combiner les points forts de chaque approche, et d'obtenir ainsi des performances plus élevées et par conséquent une meilleure estimation des crues extrêmes.

Notre projet d'étude présente la possibilité d'appliquer des réseaux de neurones formels optimisés par un algorithme génétique pour la prévision des débits des cours d'eau.

Ce projet est élaboré selon le plan de travail suivant :

- Chapitre I : Réseau de neurones artificiels, dans lequel on donne des définitions et un bref historique et où on définit le réseau neuronal utilisé ;
- Chapitre II : algorithme génétique, ce chapitre donne un aperçu sur ces algorithmes et leur fonctionnement. Nous présentons aussi le principe d'hybridation neuro-génétique et son fonctionnement en donnant l'organigramme principal du système hybride neuro-génétique ;
- Chapitre III : Préviation des débits par l'approche hybride neuro-génétique, nous présentons dans ce chapitre et dans un premier temps, les résultats de l'application d'un exemple théorique de test pour valider les programmes de calcul. Par la suite, nous présentons les résultats obtenus sur trois bassins versants différents et nous commentons globalement les résultats ;
- Conclusion générale.

Chapitre I

**Réseaux de neurones  
artificiels**

## I.1 Introduction

Les réseaux neuronaux Artificiels peuvent être considéré en tant que procédures de recherche et assorties informatiques de modèle qui permettent des prévisions sans connaissance intime des processus physiques ou chimiques [1].

Inspirées par des observations de la biologie réelle, les réseaux des neurones artificiels sont des inventions mathématiques.

Cependant les réseaux neuronaux artificiels peuvent être décrit en tant que mappage d'un espace d'entrée à l'espace de sortie. Ce concept est analogue à celui d'une fonction mathématique. Le but d'un réseau de neurones consiste à mapper une entrée à une sortie souhaitée [2].

Si la modélisation résulte des interconnexions entre les neurones, Les réseaux neuronaux artificiels ne sont plus liés à de véritables neurones que les plumes sont liées à des avions modernes. Les deux systèmes biologiques, les neurones et les plumes, ne servent, dans les inventions de l'homme, que peu de ressemblance avec les systèmes biologiques qui ont engendré le processus créatif [2].

Nombreux termes sont utilisés dans la littérature pour désigner le domaine des réseaux de neurones artificiels, comme connexionnisme ou neuromimétique. Il faut associer à chacun de ces noms une sémantique précise. Ainsi, les réseaux de neurones artificiels ne désignent que les modèles manipulés ; ce n'est ni un domaine de recherche, ni une discipline scientifique. Connexionnisme et neuromimétique sont tous deux des domaines de recherche à part entière, qui manipulent chacun des modèles de réseaux de neurones artificiels, mais avec des objectifs différents. L'objectif poursuivi par les ingénieurs et chercheurs connexionnistes est d'améliorer les capacités de l'informatique en utilisant des modèles aux composants fortement connectés. Pour leur part, les neuromiméticiens manipulent des modèles de réseaux de neurones artificiels dans l'unique but de vérifier leurs théories biologiques du fonctionnement du système nerveux central [3].

## I.2 Bref historique

Nous citons ci-dessous la progression de la théorie des réseaux neuronaux [3]:

- 1890 :

W. James, célèbre psychologue américain introduit le concept de mémoire associative, et propose ce qui deviendra une loi de fonctionnement pour l'apprentissage sur les réseaux de neurones connue plus tard sous le nom de loi de Hebb.

- 1943 :

J. Mc Culloch et W. Pitts laissent leurs noms à une modélisation du neurone biologique (un neurone au comportement binaire). Ceux sont les premiers à montrer que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes (tout au moins au niveau théorique).

- 1949 :

D. Hebb, physiologiste américain explique le conditionnement chez l'animal par les propriétés des neurones eux-mêmes. Ainsi, un conditionnement de type pavlovien tel que, nourrir tous les jours à la même heure un chien, entraîne chez cet animal la sécrétion de salive à cette heure précise même en l'absence de nourriture. La loi de modification des propriétés des connexions entre neurones qu'il propose explique en partie ce type de résultats expérimentaux.

- 1957 :

F. Rosenblatt développe le modèle du Perceptron

. Il construit le premier neuro-ordinateur basé sur ce modèle et l'applique au domaine de la reconnaissance de formes. Notons qu'à cette époque les moyens à sa disposition étaient limités.

- 1960 :

B. Widrow, un automaticien, développe le modèle Adaline (Adaptative Linear Element). Dans sa structure, le modèle ressemble au Perceptron, cependant la loi d'apprentissage est différente. Celle-ci est à l'origine de l'algorithme de rétro-propagation de gradient très utilisé aujourd'hui avec les Perceptrons multicouches.

- 1969 :

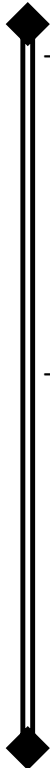
M. Minsky et S. Papert publient un ouvrage qui met en exergue les limitations théoriques du perceptron. Limitations alors connues, notamment concernant l'impossibilité de traiter par ce modèle des problèmes non linéaires. Ils étendent implicitement ces limitations à tous modèles de réseaux de neurones artificiels. Leur objectif est atteint.

- 1967-1982 :

Toutes les recherches ne sont, bien sûr, pas interrompues. Elles se poursuivent, mais déguisées, sous le couvert de divers domaines comme : le traitement adaptatif du signal, la reconnaissance de formes, la modélisation en neurobiologie, etc. De grands noms travaillent durant cette période tels : S. Grossberg, T. Kohonen, ...

- 1982 :

J. J. Hopfield est un physicien reconnu à qui l'on doit le renouveau d'intérêt pour les réseaux de neurones artificiels. A cela plusieurs raisons : Au travers d'un article court, clair et bien écrit, il présente une théorie du fonctionnement et des possibilités des réseaux de neurones. ; J. J. Hopfield fixe préalablement le comportement à atteindre pour son modèle et construit à partir de là, la structure et la loi d'apprentissage correspondant au résultat escompté. D'autre part les limitations du Perceptron mise en avant par M. Minsky ne soient pas levées par le modèle d'Hopfield, les recherches sont relancées.



- 1983 :

La Machine de Boltzmann est le premier modèle connu apte à traiter de manière satisfaisante les limitations recensées dans le cas du perceptron. Mais l'utilisation pratique s'avère difficile, la convergence de l'algorithme étant extrêmement longue.

- 1985 :

La rétro-propagation de gradient apparaît. C'est un algorithme d'apprentissage adapté aux réseaux de neurones multicouches (aussi appelés Perceptrons multicouches). Sa découverte réalisée par trois groupes de chercheurs indépendants indique que "la chose était dans l'air". Dès cette découverte, nous avons la possibilité de réaliser une fonction non linéaire d'entrée/sortie sur un réseau en décomposant cette fonction en une suite d'étapes linéairement séparables.

Avec le temps la théorie des réseaux de neurones artificiels devient respectable et plus utilisée par les physiciens dans divers domaines.

### I.3 Apprentissage numérique :

Généralement l'apprentissage numérique s'agit de faire à l'aide d'une procédure numérique programmée et exécutée sur un ordinateur, inférer un modèle d'un processus que l'on observe 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 [4].

D'autre part l'apprentissage numérique implique les mécanismes adaptatifs qui permettent d'apprendre de l'expérience, apprennent par exemple et analogie. Les possibilités d'étude peuvent améliorer l'exécution d'un système intelligent avec le temps. Les mécanismes d'apprentissage de machine forment la base pour les systèmes adaptatifs [5].

Le processus d'apprentissage numérique est basé sur l'hypothèse que le processus peut être décrit avec la précision désirée par une ou plusieurs fonctions

qui contiennent des paramètres, et l'on ajuste ces derniers pour que cette ou ces fonctions s'ajustent aux données. L'apprentissage est un algorithme qui permet d'ajuster ces paramètres [4].

Ayant donné un bref historique de progression de réseaux des neurones artificiels et définir l'apprentissage numérique, nous présenterons ci-dessous les approches les plus populaires d'apprentissage numérique qui sont les réseaux des neurones artificiels.

## I.4 Les neurones artificiels

### I.4.1 Le modèle neurophysiologique

Le cerveau se compose d'environ mille milliards de neurones, avec 1000 à 10000 synapses (connexions) par neurone [3]. Dans ce qui suit on donne la définition de cette cellule.

### I.4.2 Le neurone

Le neurone est une cellule composée d'un corps cellulaire et d'un noyau. Le corps cellulaire se ramifie pour former ce que l'on nomme les **dendrites**. Celles-ci sont parfois si nombreuses que l'on parle alors d'arborisation dendritique. C'est par les dendrites que l'information est acheminée de l'extérieur vers le **soma**, corps du neurone. (Fig. I-1 et I-2)

L'information traitée par le neurone chemine ensuite le long de l'**axone** pour être transmise aux autres neurones. La transmission entre deux neurones n'est pas directe. En fait, il existe un espace intercellulaire de quelques dizaines d'Angströms ( $10^{-9}$  m) entre l'axone du neurone afférent et les dendrites du neurone efférent. La jonction entre deux neurones est appelée la synapse [3].

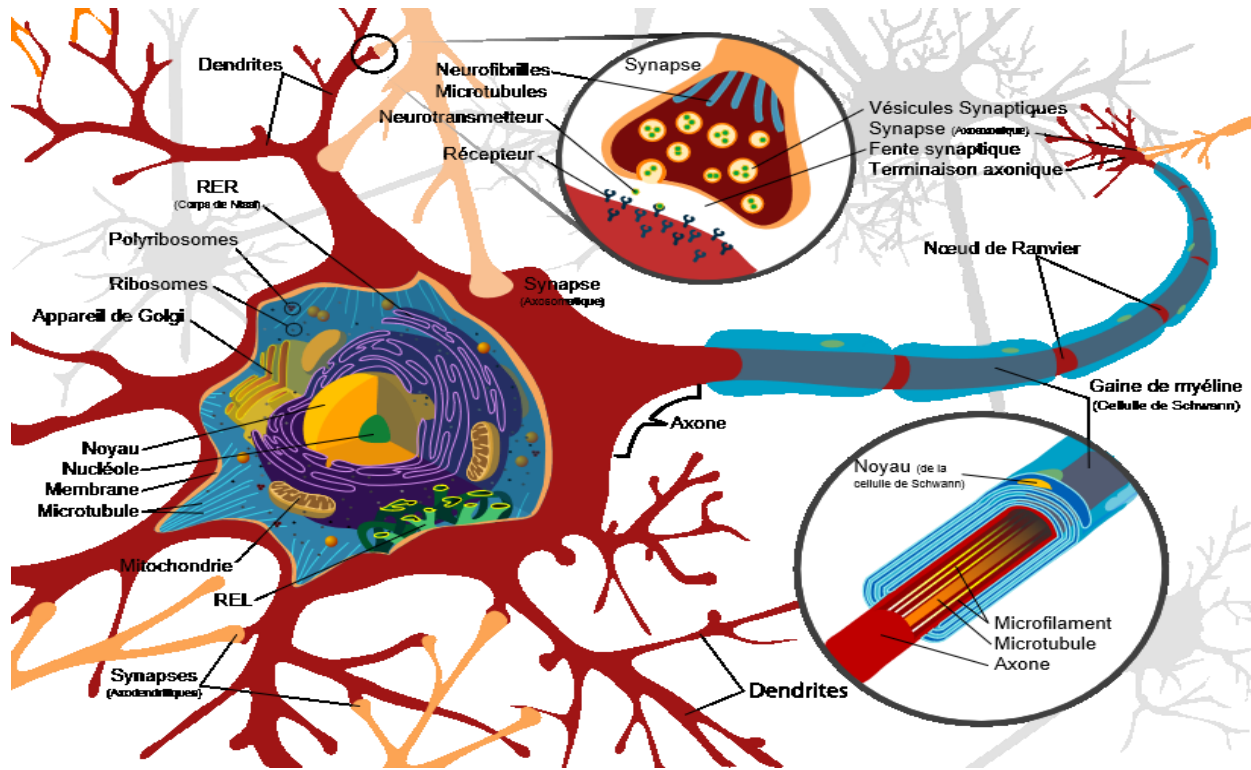


Fig. I-1 : Schéma complet d'un neurone [45]

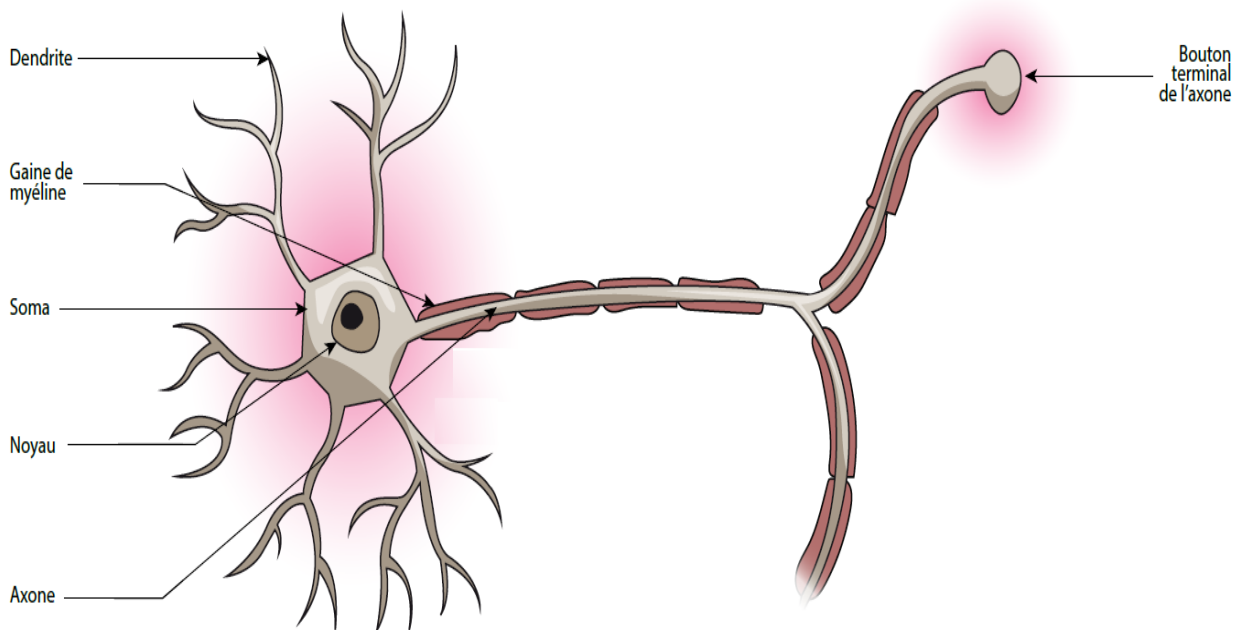


Fig. I-2 : Schéma simplifiée d'un neurone [46]

### I.4.2.1 Dendrite

Elle présente une sortie de ramification appelée arbre dendritique, collecte les signaux venant d'autres cellules ou de l'extérieur. La réception des signaux en provenance des autres neurones se fait par des points de contacts appelés synapses. Ces informations, qui ne sont rien d'autre que des impulsions électriques, sont par la suite acheminées vers le corps cellulaire [6].

### I.4.2.2 Soma

L'arbre dendritique fait converger vers le soma des influx nerveux venus d'une très large étendue autour du neurone. Le soma, outre son rôle concernant le métabolisme de la cellule, recueille et concentre les informations reçues et en fait une sommation dite « spatio temporelle ». Si le potentiel somatique dépasse un certain seuil, il y a émission d'un potentiel, appelé « spike » qui correspond à une oscillation électrique très brève (1 ms) [6].

### I.4.2.3 Axone

Il sert comme moyen de transport pour les signaux émis par le neurone (l'information traitée) vers l'extérieur, et répartie sur les synapses de neurones cibles grâce à l'arborescence terminale que possède l'axone. Il se distingue des dendrites par sa forme et par les propriétés de sa membrane externe, en effet il est généralement plus long que les dendrites et se ramifie à son extrémité, là où il communique avec d'autres neurones [6].

## I.4.3 Le modèle mathématique

### I.4.3.1 Le Neurone Formel :

Le neurone formel (artificiel) est l'élément de base du réseau neuronal artificiel. Sa conception et fonctionnalité sont dérivées des observations de neurone biologique<sup>1</sup>. Des similitudes dans la conception et les fonctionnalités est présentées dans la figure ci-dessous (Fig.I-3) où à gauche de la figure une représentation d'un neurone biologique avec son soma, dendrites et axone et à la

---

<sup>1</sup> La première définition formelle d'un neurone artificiel basée sur le modèle biologique a été formulée par McCulloch et Pitts.

droit de la figure une représentation d'un neurone artificiel avec ses entrées, poids, fonction d'activations, biais et sortie. [7]

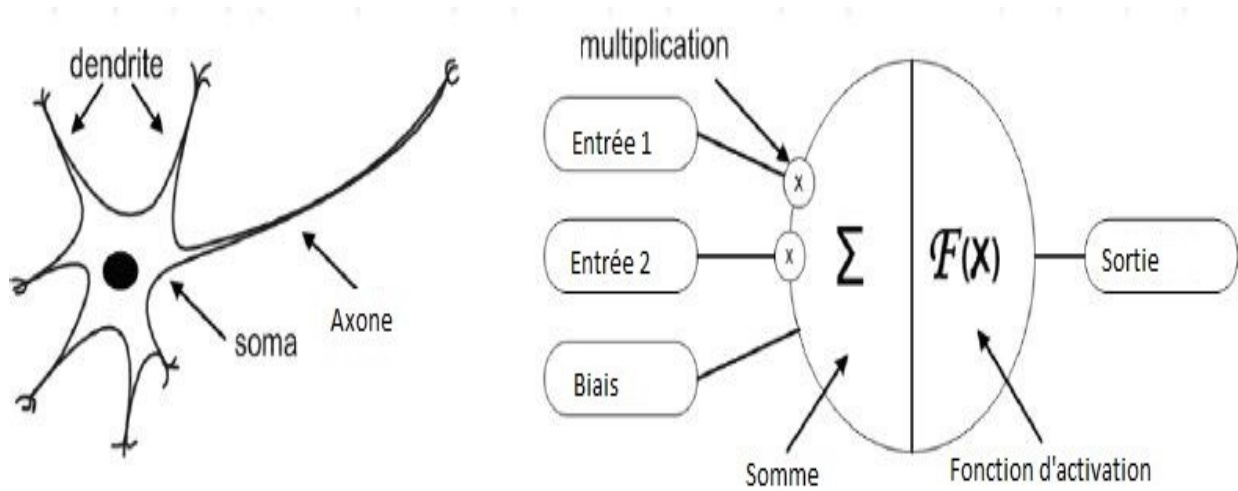


Fig. I-3 : Similitude entre le neurone biologique et artificiel [7]

Tableau I-1 : Analogie entre un neurone formel et biologique

Neurone formel	Neurone biologique
Entrées	dendrites
Poids	Synapse
Sommation et Fonction d'activation	soma
Sorties	axone

Mathématiquement on peut définir un neurone formel comme suit :

**Un neurone est une fonction non linéaire, paramétrée à valeurs bornées [4].**

Les variables sur lesquelles opère le neurone sont souvent désignées sous le terme d'entrées du neurone, et la valeur de la fonction sous le terme de sortie [4].

La fonction **F** peut être paramétrée de manière quelconque. Il existe deux types de paramétrage les plus utilisés [4] :

- **Type 1** : les paramètres sont attachés aux variables du neurone : la sortie du neurone est une fonction non linéaire d'une combinaison des variables  $X_i$  pondérées par les paramètres  $W_i$  (poids).
- **Type 2** : les paramètres sont attachés à la non-linéarité du neurone : ils interviennent directement dans la fonction  $F$ .

#### 1.4.3.2 Comportement

Le processus de traitement au niveau du neurone artificiel s'effectue en deux phases [6, 8]:

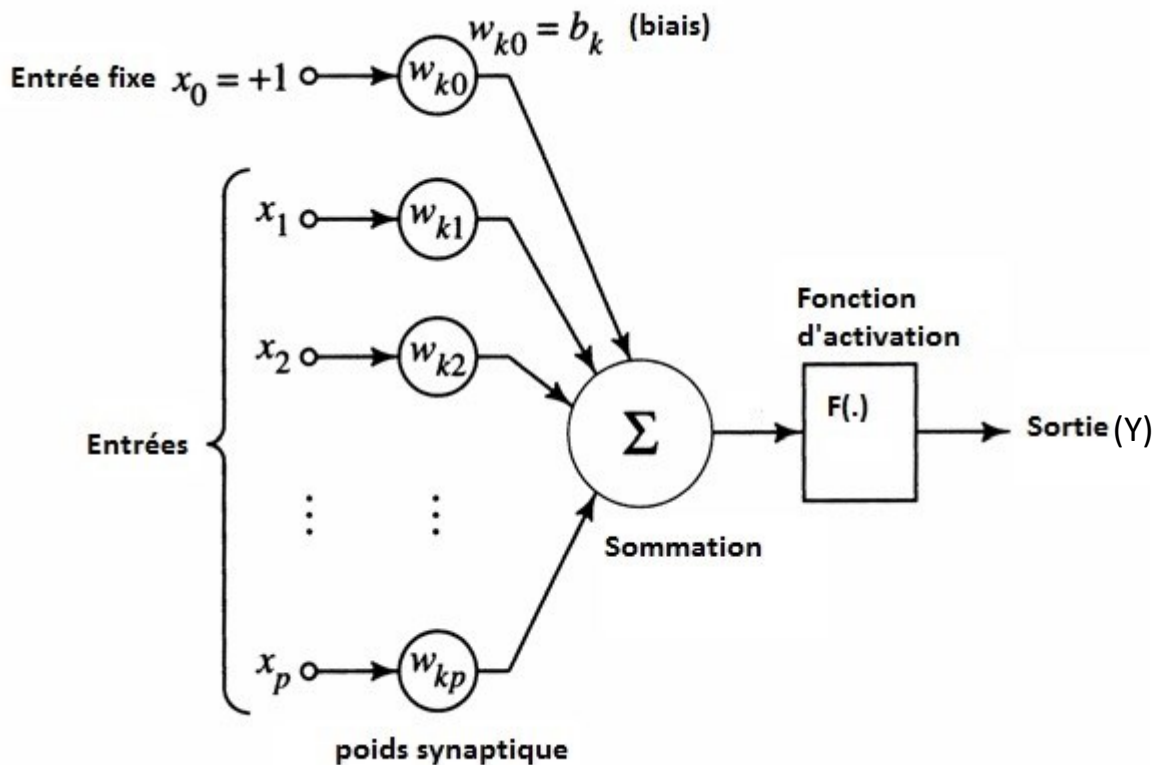


Fig. 1-4 : neurone a une fonction non linéaire [8]

Phase 1 :

- Chaque variable  $x_j$  de l'entrée  $p$  reliée au neurone  $k$  est multiplié par le poids synaptique  $w_{kp}$ . Il est important de faire attention de la manière dont les

indices de poids synaptique  $w_{kp}$  sont écrits. Le premier indice se réfère au neurone en question et le deuxième indice se réfère au variable d'entrée. Le poids  $w_{kp}$  est positif si la synapse est excitatrice; elle est négative si la synapse est inhibitrice.

- le neurone effectue une sommation des variables pondérées (principe de superposition) ; la valeur numérique obtenue représente l'état du neurone qui l'a émis, afin d'obtenir une stimulation résultante globale :

$$v = \sum_{j=0}^p w_{kj} x_j \quad \text{Équation 0-1}$$

Phase 2:

- A partir de cette valeur d'équation (I-1), une fonction de transfert calcule la valeur de l'état du neurone. C'est cette valeur qui sera transmise aux neurones aval.

$$y = F(v_k) \quad \text{Équation 0-2}$$

### I.4.3.3 Fonction d'activation

Il existe plusieurs formes de la fonction d'activation. Elle est en général une fonction non linéaire monotone croissante ; par ailleurs les fonctions de transfert ont des propriétés diverses : elles peuvent être déterministes, continues, discontinues ou aléatoires. La plupart des fonctions de transfert sont continues, offrant une infinité de valeurs possibles comprises dans l'intervalle  $[0, +1]$  ou  $[-1, +1]$ . On cite ci-dessous les plus couramment utilisés [3, 6]:

Tableau I-2 : formes de la fonction d'activation [9]

Nom de la fonction	Forme mathématique
Linéaire	$a = n$
Seuil	$a = \begin{cases} +1 & \text{si } n > n_1 \\ 0 & \text{si non} \end{cases}$
Sigmoïde	$a = \alpha \frac{e^{kn} - 1}{e^{kn} + 1}$
Tangente hyperbolique	$a = \frac{1 - e^{-2n}}{1 + e^{2n}}$
Fonction de base radiale	$a = e^{-n^2/\beta^2}$

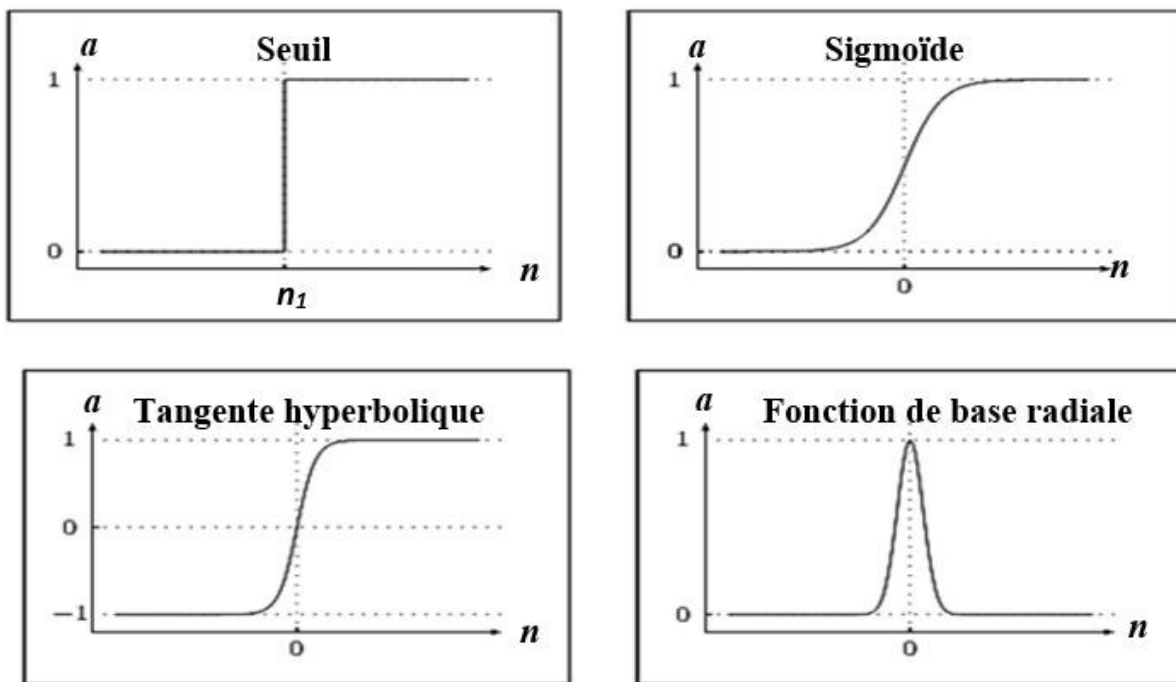


Fig. I-5 : présentation graphique des fonctions d'activations [9]

## I.5 Réseau de neurones artificiels

La combinaison de deux ou plusieurs neurones artificiels nous permet d'obtenir un réseau de neurones artificiels. En fait les réseaux de neurones artificiels sont capables de résoudre les problèmes de la vie réelle complexes [7].

### I.5.1 Topologie d'un réseau de neurones artificiels

La topologie, architecture ou le graphe d'un réseau neuronal artificiel se définit d'après la manière dont les neurones s'interconnectent entre eux. Le fait que l'interconnexion peut être faite de plusieurs façons en résulte de nombreuses topologies possibles qui sont divisés en deux grandes classes [4, 7]:

- les réseaux de neurones non bouclés ;
- les réseaux de neurones bouclés

### I.5.2 Les réseaux de neurones non bouclés

Un réseau de neurones non bouclé réalise une ou plusieurs fonctions de ses entrées par composition des fonctions réalisées par chacun des neurones.

Un réseau de neurones non bouclé peut donc être imaginé comme un ensemble de neurones connectés entre eux, l'information circulant des entrées vers les sorties sans retour en arrière.

Dans cette topologie, Les neurones sont classés dans des couches dont Les neurones qui effectuent le dernier calcul de la composition de fonctions sont les neurones de sortie et ceux qui effectuent des calculs intermédiaires sont les neurones cachés. La représentation de la topologie d'un réseau par un graphe est très utile. Pour ces réseaux, ils sont représenté par un graphe acyclique dont les nœuds sont les neurones et les arêtes sont les connexions entre ceux-ci (4; 8).

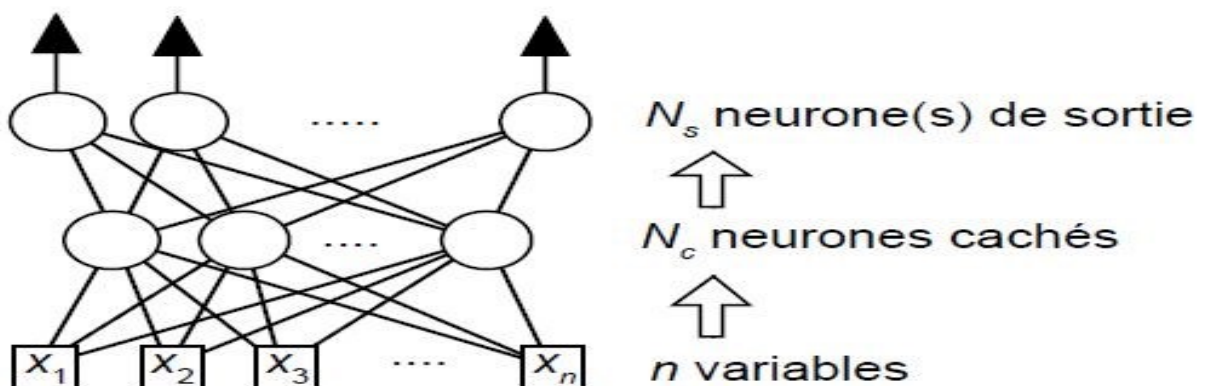


Fig. I-6 : graphe d'un réseau non bouclé [4]

### I.5.3 Les réseaux de neurones bouclés

L'architecture la plus générale, pour un réseau de neurones bouclés est présentée par un graphe cyclique, lorsque l'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 ce chemin est désigné sous le terme de cycle [4].

Ils sont connus sous la nomination d'un réseau à connexions récurrentes dont ces connexions ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouche. Ces connexions sont le plus souvent locales [3].

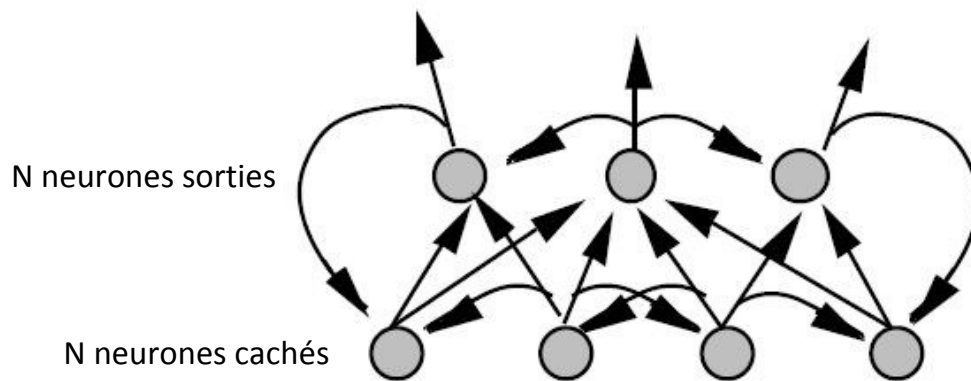


Fig. I-7 : graphe d'un réseau bouclé (ou récurrent) [3]

## I.6 L'apprentissage des réseaux de neurones

L'apprentissage est la procédure qui consiste à estimer les paramètres des neurones du réseau, afin que celui-ci remplisse au mieux la tâche qui lui est affectée [11].

Il y a trois règles majeures d'apprentissage, apprentissage supervisé, apprentissage non supervisé et l'apprentissage par renforcement. Habituellement, ils peuvent être utilisés par toute architecture de réseau neuronal artificiel. Chaque règle d'apprentissage a de nombreux algorithmes [7].

### I.6.1 Apprentissage supervisé

L'apprentissage supervisé est une technique d'apprentissage machine qui définit les paramètres d'un réseau neuronal artificiel à partir de données d'entraînement. On peut donc affecter à un tel réseau la tâche qui consiste à réaliser une fonction algébrique non linéaire ; celle-ci peut être :

- connue analytiquement : le réseau réalise alors une tâche d'approximation de fonction ;
- inconnue analytiquement, mais pour laquelle on dispose de valeurs, en nombre fini, qui sont entachées de bruit si ces valeurs résultent de mesures effectuées sur un processus physique, chimique, économique, biologique, etc. : le réseau réalise alors une tâche de modélisation statique ou une régression.

On connaît donc, en tous points ou seulement en certains points, les valeurs que doit avoir la sortie du réseau en fonction des entrées correspondantes : c'est en ce sens que l'apprentissage est supervisé [11].

### I.6.2 Apprentissage non supervisé

Apprentissage non supervisé est une technique d'apprentissage qui définit les paramètres d'un réseau de neurones artificiels basés sur des données fournies et une fonction de coût qui doit être minimisée. La fonction de coût peut être une fonction quelconque et elle est déterminée par la formulation de la tâche. L'apprentissage non supervisé est principalement utilisé dans des applications qui entrent dans le domaine des problèmes d'estimation tels que la modélisation statistique, la compression, le filtrage, la séparation aveugle de sources et l'agrégation. Dans l'apprentissage non supervisé, nous cherchons à déterminer comment les données sont organisées. Elle diffère de l'apprentissage supervisé et l'apprentissage par renforcement en ce que le réseau neuronal artificiel est donné seulement les exemples non étiquetés. Une forme commune de l'apprentissage non supervisé est l'agrégation où nous essayons de classer les données dans différents groupes par leur similitude. Les réseaux à apprentissage non supervisé les plus étudiés et utilisés sont les « cartes auto-organisatrices » ou « cartes de Kohonen » [7, 10].

### I.6.3 Apprentissage par renforcement

Apprentissage par renforcement est une technique d'apprentissage qui définit les paramètres d'un réseau de neurones artificiels, où les données sont généralement pas données, mais générées par les interactions avec l'environnement. L'apprentissage par renforcement est préoccupé par la façon dont un réseau neuronal artificiel devrait prendre des mesures dans un environnement de façon à maximiser une certaine notion de récompense à long terme. L'apprentissage par renforcement est fréquemment utilisé comme une partie de l'algorithme général d'apprentissage du réseau neuronal. Après la définition de la fonction de retour qui doit être maximisée, l'apprentissage par renforcement utilise plusieurs algorithmes pour trouver la méthode qui produit le rendement maximal. L'algorithme par renforcement calcule en première étape la fonction de retour pour chaque méthode possible et choisit la méthode avec le plus grand retour. La faiblesse évidente de cet algorithme est dans le cas de très grand nombre des méthodes possibles. Cette faiblesse peut être résolue par des approches de la valeur de la fonction ou l'estimation directe de la méthode. La valeur de la fonction d'approche tente de trouver une méthode qui maximise le retour par le maintien d'un ensemble d'estimations de rendements attendus pour une méthode. Ces méthodes convergent vers les estimations correctes pour une méthode fixe et peuvent également être utilisés pour trouver la politique optimale. La méthode d'estimation directe peut trouver la méthode optimale. En cherchant directement dans l'espace de la méthode ce qui augmente considérablement le coût de calcul.

Apprentissage a été appliquée avec succès à divers problèmes, y compris le contrôle de robot, les télécommunications, et les jeux comme les échecs et d'autres tâches séquentielles de prise de décision [7].

## I.7 Approximation parcimonieuse

Cette approximation est une propriété remarquable et fondamentale des réseaux de neurones artificiels qui est à l'origine de leur intérêt pratique dans des domaines très divers : ce sont des **approximateurs universels parcimonieux**.

On admet la définition énoncée par **G. DREYFUS** [11] :

« 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 »

La propriété d'approximation n'est pas une propriété spécifique aux réseaux de neurones artificiels. Mais la spécificité des réseaux de neurones réside dans le caractère parcimonieux de l'approximation. Les paramètres ajustables dans le réseau de neurones sont un peu nombreux que d'autres approximateurs universels (ondelette, fonction radiale,..... etc.), plus précisément, le nombre de poids varie linéairement avec le nombre de variables de la fonction à approcher, alors qu'il varie exponentiellement pour la plupart des autres approximateurs.

La propriété de parcimonie peut se comprendre dans le cadre qualitatif de la manière suivante :

Si l'approximation est une combinaison linéaire de fonctions élémentaires fixées dans le cas des gaussiennes par exemple, on ne peut ajuster que les coefficients de la combinaison, en revanche dans le cas d'un réseau de neurones, lorsque l'approximation est une combinaison linéaire de fonctions non linéaires à paramètres ajustables, on ajuste à la fois les coefficients de la combinaison et la forme des fonctions que l'on combine.

Le plus souvent dans la vie pratique, l'ingénieur dispose d'un ensemble de mesures de variables d'un processus de nature quelconque, et du résultat de ce processus ; il suppose qu'il existe une relation déterministe entre ces variables et ce résultat, et il cherche une forme mathématique de cette relation, valable dans le domaine où les mesures ont été effectuées, sachant que :

- les mesures sont en nombre fini ;
- elles sont certainement entachées de bruit ;
- toutes les variables qui déterminent le résultat du processus ne sont pas forcément mesurées.

En termes de statistiques, l'ingénieur cherche la fonction de régression de la grandeur à modéliser. Cette fonction est inconnue, mais on peut en chercher une approximation à partir des mesures disponibles. Si la fonction de régression cherchée est non linéaire, les réseaux de neurones sont donc de bons candidats pour cela.

L'approximation est obtenue en estimant les paramètres d'un réseau de neurones au cours d'une phase dite d'apprentissage. C'est ici que la propriété d'approximation parcimonieuse des réseaux de neurones est précieuse. En effet que les réseaux de neurones nécessite moins de données que les autres méthodes de modélisations [11].

## I.8 Quelques modèles des réseaux de neurones :

### I.8.1 Perceptron

C'est Un réseau de neurones à un seul neurone, qu'il est capable de l'apprentissage des problèmes linéairement séparables [4, 7].

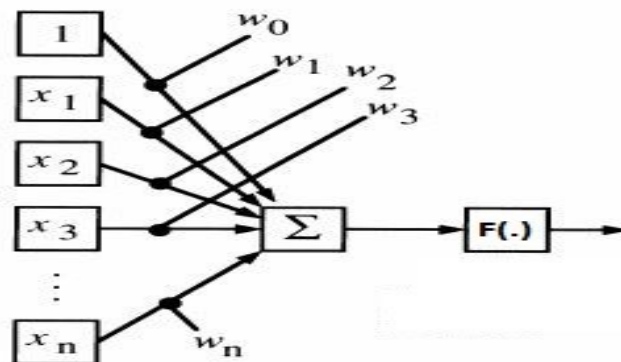


Fig. I-8 : Schéma d'un perceptron [8]

Les réseaux de neurones à plusieurs couches cachées au sein desquelles les informations circulent de la couche d'entrée vers la couche de sortie uniquement et les neurones cachés ont une fonction d'activation sigmoïde (Fig. I-9). Ce type de réseau neuronal est appelé perceptron multicouche [4].

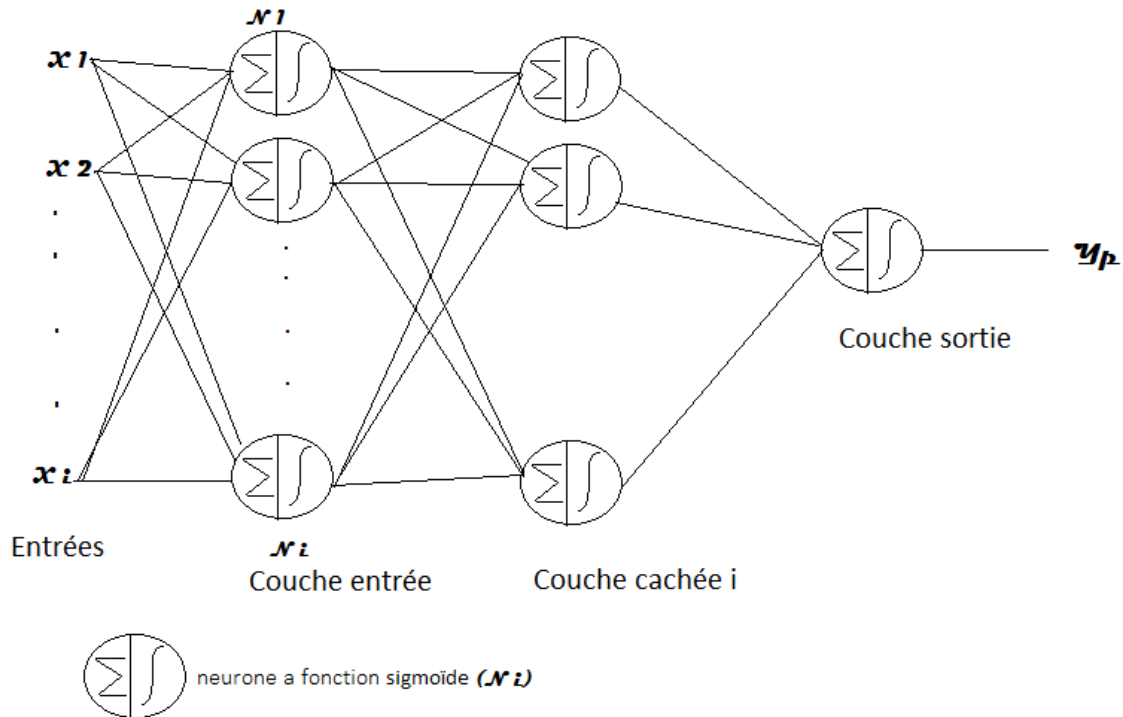


Fig. I-9 : Schéma simplifié d'un perceptron multicouche

### I.8.2 Réseau feedforward

Le réseau de neurones artificiel Feedforward, dont l'information doit circuler entre entrée et sortie dans une seule direction, sans être bouclée. Il n'y a aucune limitation sur le nombre de couches, type de fonction de transfert utilisée dans neurone.

Les perceptrons multicouche est un réseau feedforward, dont la fonction d'activation dans le neurone est sigmoïdale. L'exemple le plus simple d'un réseau neuronal feedforward est un seul perceptron.

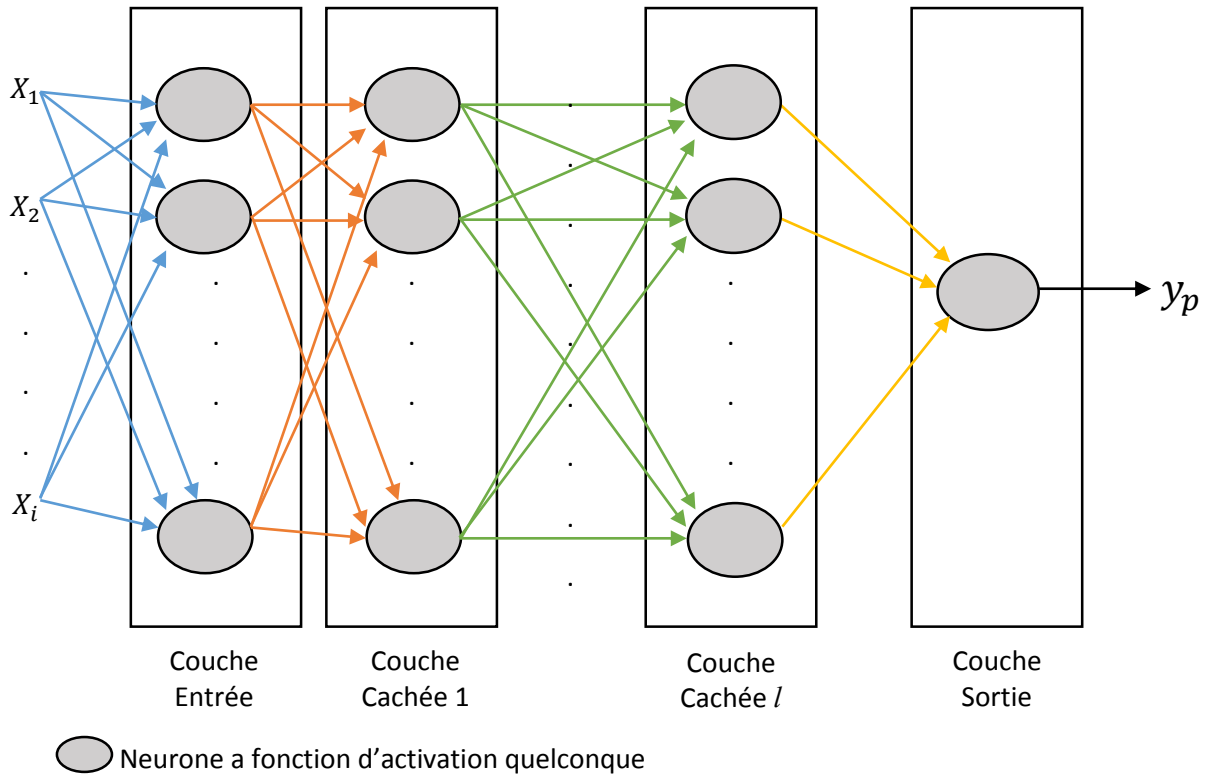


Fig. I-10 : Schéma simplifié d'un réseau feedforward à  $i$  entrées et  $l$  couche cachées

### I.8.3 Autre modèles des réseaux neuronaux :

Il existe nombreux modèles des réseaux neuronaux à différentes topologies et méthodes d'apprentissage. On cite quelques-uns dans ce que suit :

- Les réseaux neuronaux artificiels **Hopfield** (réseaux bouclé);
- Les réseaux neuronaux artificiels **Elman and Jordan** (réseaux bouclé);
- Les réseaux neuronaux artificiels **ART** (architecture Evolutive) ;
- Les réseaux neuronaux artificiels **Bi-directionnel** ;
- Les réseaux neuronaux artificiels **stochastiques** ;

## I.9 Algorithme de Levenberg-Marquardt

Cet algorithme a été développé par Kenneth Levenberg, puis publié par Donald Marquardt. C'est un algorithme qui peut être défini comme amélioration

de la méthode classique de Gauss-Newton dans la résolution des problèmes de régression non-linéaire des moindres carrés. L'algorithme de Levenberg-Marquardt est plus stable que celui de Gauss-Newton, il trouve une solution même s'il est démarré très loin d'un minimum.

### I.9.1 Description de l' algorithme LM

Le problème pour lequel l'algorithme LM fournit une solution est appelé la minimisation de moindre carré non linéaire. Cela implique que la fonction devant être minimisée est de la forme suivante [13] :

$$f(x) = \frac{1}{2} \sum_{j=1}^m r_j^2(x) \quad \text{Équation I-0-3}$$

où :

$x$  : Vecteur  $x_i$

$r_j$  : Fonction de  $R^n$  a  $R$

$r_j$  Sont connu sous le nom d'un résidu et il est supposé que  $m \geq n$ . D'une manière simplifiée,  $f$  est représenté comme un vecteur résiduel  $r$  de  $R^n \rightarrow R^m$  définit comme suit :

$$r(x) = r_1(x), r_2(x), \dots, r_m(x)$$

Maintenant,  $f$  peut-être récrit comme :

$$f(x) = \frac{1}{2} \|r(x)\|^2 \quad \text{Équation I-0-4}$$

La dérivée de la fonction  $f$  peut être écrire comme suit

$$J(x) = \frac{\partial r_i}{\partial x_i} \quad 1 \leq j \leq m, 1 \leq i \leq n$$

D'où :  $J$  est la matrice jacobienne

Dans le cas où toutes  $r_i$  sont une fonction linéaire, la matrice jacobienne est constante et on peut représenter  $r_i$  comme hyperplan dans un espace et  $f$  devient :

$$f(x) = \frac{1}{2} \|Jx + r(0)\|^2$$

et :  $\nabla f(x) = J^T (Jx + r)$

pour :  $\nabla^2 f(x) = J^T J$

d'autre part dans le cas non linéaire :

$$\nabla f(x) = \sum_{j=1} r_j(x) \nabla r_j(x) = J(x)^T r(x)$$

et

$$\nabla^2 f(x) = J(x)^T J(x) + \sum_{j=1}^m r_j(x) \nabla^2 r_j(x)$$

### I.9.2 Méthode de Levenberg entre Gauss-Newton et Gradient

Il est intéressant d'essayer de tirer le meilleur parti des deux approches, dont la méthode de la plus forte pente est généralement efficace loin d'un minimum là où une méthode de Newton diverge. Levenberg a proposé un algorithme basé sur cette observation, dont la règle de la mise à jour est un mélange des algorithmes mentionnés ci-dessus et est donné comme [14] :

$$x_{k+1} = x_k - (H + \mu I)^{-1} \nabla f(x_i) \quad \text{Équation I-0-5}$$

Où  $H$  la matrice hessienne évaluée à  $x_i$  ;  $\mu$  : paramètre d'algorithme

Si l'erreur va descendre suite d'une mise à jour, cela signifie que notre hypothèse quadratique sur  $f(x)$  est valable et nous réduisons  $\mu$  par un facteur de  $\mu_{\text{amor}}$  afin de réduire l'influence de la descente de gradient. D'autre part, si l'erreur augmente, nous tenons à augmenter  $\mu$  par un facteur  $\mu_{\text{aug}}$ .

La mise à jour de l'algorithme de Levenberg Marquardt se déroule comme suit [13] :

1. Faire une mise à jour comme indiqué par la règle ci-dessus ;
2. Évaluer l'erreur au nouveau vecteur ;
3. Si l'erreur a augmenté à la suite de la mise à jour, réinitialiser les poids à leurs valeurs précédentes et augmenter  $\mu$  par le facteur  $\mu_{aug}$ . Ensuite, allez à (1) et essayer une mise à jour à nouveau.
4. Si l'erreur a diminué à la suite de la mise à jour, garder les poids à leurs nouvelles valeurs et de diminuer  $\mu$  par un facteur de  $\mu_{amor}$ .

### I.10 L'étape d'initialisation des poids d'un réseau neuronal

L'initialisation des poids synaptiques d'un réseau de neurones artificiels dans l'apprentissage à rétro-propagation est la première étape à effectuer.

Le bon choix de ces poids peut aider énormément dans le fonctionnement des réseaux neuronaux. Les valeurs élevées peuvent provoquer un phénomène de saturation prématurée qui contribue à diminuer la vitesse de convergence de l'apprentissage. Elles doivent être différentes de zéro pour que l'algorithme de rétro-propagation puisse fonctionner [6, 8].

### I.11 Conclusion

Après avoir donné un bref historique et les principales définitions sur les réseaux de neurones artificiels, nous avons opté pour un réseau de neurones artificiels à topologie **Feedforward** (perceptron multicouche) avec un **apprentissage supervisé** basé sur l'algorithme de **Levenberg Marquardt**, le nombre des neurones sera déterminer en fonction du problème traité.

En profitant de l'hybridation neuro-génétique, on optimisera les poids et les paramètres de l'algorithme d'apprentissage par algorithme génétique, ce que nous présenterons dans le chapitre 2.

# Chapitre II

## Les algorithmes génétique

## II.1 Introduction

Les algorithmes génétiques entrent dans le cadre des algorithmes évolutionnaires. Cette programmation est un calcul évolutif, qui permet de résoudre automatiquement les problèmes sans que l'utilisateur connaisse ou précise la forme ou la structure de la solution à l'avance.

Plus simplement les algorithmes évolutionnaires sont un domaine des méthodes systématiques, qui donnent aux ordinateurs la possibilité de résoudre des problèmes automatiquement à partir d'une déclaration de ce qui doit être fait [15].

Les algorithmes génétiques appartiennent à la famille des algorithmes évolutionnistes. Ces algorithmes évolutionnaires constituent une approche originale : il ne s'agit pas de trouver une solution analytique exacte, ou une bonne approximation numérique, mais de trouver des solutions satisfaisant au mieux à différents critères, souvent contradictoires. S'ils ne permettent pas de trouver à coup sûr la solution optimale de l'espace de recherche, du moins peut-on constater que les solutions fournies sont généralement meilleures que celles obtenues par des méthodes plus classiques, pour un même temps de calcul [16].

Les algorithmes génétiques figurent parmi les méthodes non-déterministes. Ces méthodes font appel à des tirages de nombres aléatoires qui permettent d'explorer l'espace de recherche plus efficacement. Elles sont faciles à implanter pour le traitement des problèmes d'optimisation discrète, quand l'espace de recherche devient non-convexe ou lorsque les gradients sont discontinus [16].

Dans ce chapitre nous donnerons un bref historique et le principe d'un algorithme génétique ainsi qu'un aperçu sur l'optimisation des réseaux de neurones artificiels par ces algorithmes.

## II.2 Bref historique

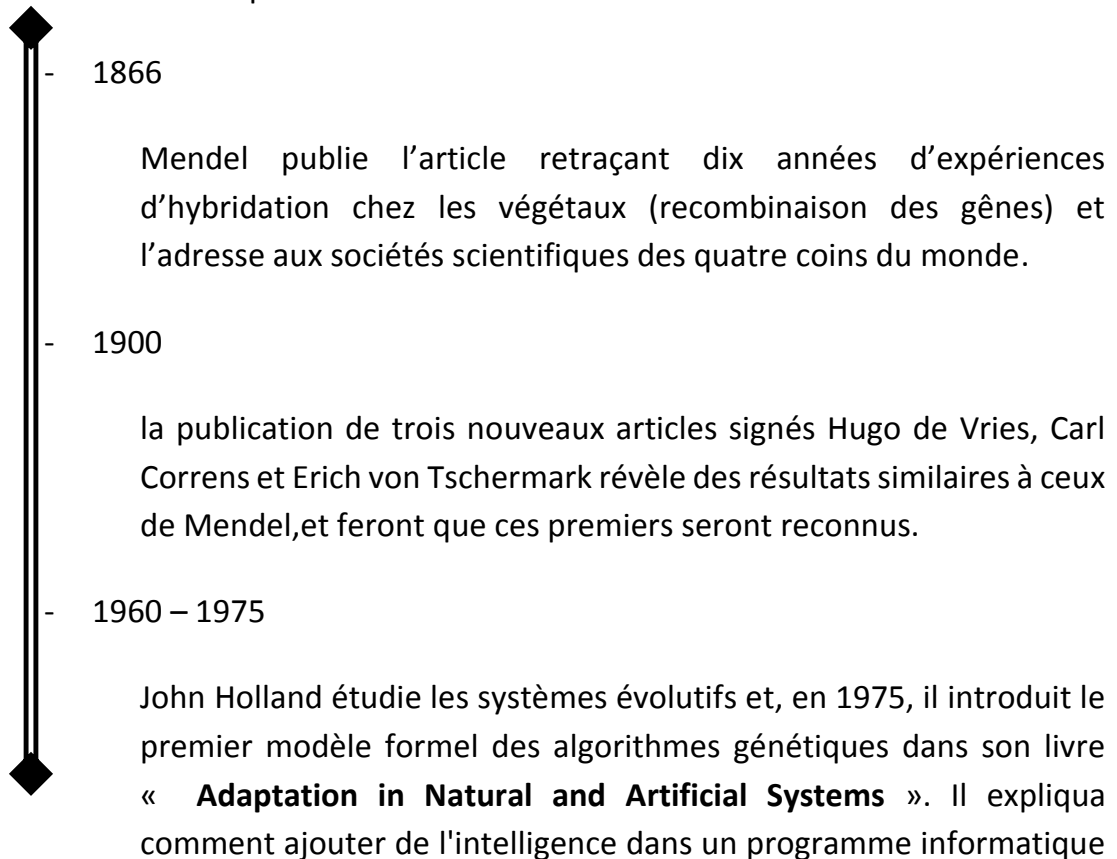
Charles Darwin publie son livre intitulé « **L'origine des espèces au moyen de la sélection naturelle ou la lutte pour l'existence dans la nature** » en 1860. Il


expose sa théorie de l'évolution des espèces : sous l'influence de contraintes extérieures, les êtres vivants se sont graduellement adaptés à leur milieu naturel au travers de processus de reproductions.

Les quatre lois mises en avant par la théorie proposée par Darwin qui clarifie l'évolution des espèces sont :

1. - La loi de croissance et de reproduction ;
2. - La loi d'hérédité qu'implique quasiment la loi de reproduction ;
3. - La loi de variabilité, résultant de condition d'existence ;
4. - La loi de multiplication des espèces qui amène la lutte pour l'existence et qui a pour conséquence la sélection naturelle.

Nous citons les travaux qui suivirent dans ce domaine et nous les citerons sur l'échelons historique ci-dessus :





avec les croisements (échangeant le matériel génétique) et la mutation (source de la diversité génétique) [16].

- 1989

Goldberg publie son ouvrage de vulgarisation des algorithmes génétiques, et ajouta à la théorie des algorithmes génétiques les idées suivantes :

- un individu est lié à un environnement par son code d'ADN.
- une solution est liée à un problème par son indice de qualité.

Le développement des algorithmes génétiques continuera et ils sont appliqués dans tous les domaines de recherche pour l'optimisation.

### II.3 Algorithmes évolutionnaires

Dans les années 60 et 70 du siècle passé, Les algorithmes évolutionnaires sont apparus. Ce sont le résultat de la recherche qui a abouti aux méthodes approchées.

Les méthodes approchées sont élaborés afin des résoudre des problèmes pour lesquels il est difficile, de trouver une solution en un temps limité. Il est alors utile de trouver une technique permettant la localisation rapide de solutions sous optimales, sachant que l'espace de recherche a une taille et une complexité suffisamment importantes pour éliminer toute garantie d'optimalité. Pour cela, un système capable de s'auto-modifier au cours du temps, tout en améliorant sa performance dans l'accomplissement des tâches auxquelles il est confronté, semble ouvrir la voie à une recherche intéressante [18].

Dans le calcul évolutionnaire, les mécanismes de l'évolution semblent bien adaptés pour certains problèmes informatiques les plus urgents dans de nombreux domaines. Beaucoup de problèmes de calcul nécessitent une recherche dans un grand nombre de possibilités de solutions [19].

Si La base des algorithmes évolutionnaires est la simulation d'évolution de la population des solutions, l'objectif de ces algorithmes évolutionnaires est de faire évoluer une population  $P$  dans le but de trouver l'optimum. Pour ce faire, à chaque génération  $g$ , les individus de la population sont mutés et croisés avec une probabilité et ce sont les plus aptes qui survivent pour la génération suivante. Ce processus est répété pendant un certain nombre de générations, en espérant que les solutions de la fonction **fitness** apparaissent dans la population [18].

## II.4 Principe d'un algorithme génétique

Les algorithmes génétiques fournissent des solutions aux problèmes n'ayant pas de solutions calculables en temps raisonnable de façon analytique ou algorithmique [16, 20].

Un algorithme génétique recherche les extrêmes d'une fonction définie sur un espace de données appelé population. Par analogie avec la génétique, chaque individu de cette population est un chromosome et chaque caractéristique de l'individu est un gène. Dans un cas simple, un gène sera représenté par un bit (0 ou 1), un chromosome par une chaîne de bits. Chaque gène représente une partie élémentaire du problème, il peut être assimilé à une variable et peut prendre des valeurs différentes appelées allèles. La position du gène dans le chromosome se nomme locus. Le génotype représente l'ensemble des valeurs des gènes du chromosome alors que le phénotype représente la solution réelle après transformation du chromosome (voir tableau II-1). Lors de la génération d'une nouvelle population, des opérateurs génétiques tels que la sélection, le croisement et la mutation sont nécessaires pour la manipulation des chromosomes [18].

Tableau II-1 : Equivalence terminologique biologie génétique/algorithme génétique [18]

Biologie génétique	Algorithme génétique
Chromosome	Chaîne
Gène	Trait, caractéristique
Allèle	Valeur de la caractéristique
Locus	Position dans la chaîne
Génotype	Structure Ensemble des valeurs des gènes
Phénotype	Ensemble de paramètres, structure décodée Evaluation d'un génotype

Le générique d'un algorithme génétique est comme suivant [18]:

- la représentation génétique du problème, c'est-à-dire un codage approprié des solutions sous la forme de chromosomes. Cette étape associe à chacun des points de l'espace de recherche une structure de données. Elle se place généralement après une phase de modélisation mathématique du problème traité. La qualité du codage des données conditionne le succès des algorithmes génétiques ;
- le mécanisme de génération de la population initiale. Ce mécanisme doit être capable de produire une population non homogène qui servira de base pour les générations futures. Le choix de la population initiale est important car il peut prendre plus ou moins rapidement la convergence vers l'optimum global. Dans le cas où l'on ne connaît rien sur le problème à résoudre, il est essentiel que la population initiale soit répartie sur tout le domaine de recherche ;
- la fonction d'adaptation (évaluation) pour mesurer la force de chaque chromosome ;
- le mode de sélection des chromosomes à reproduire ;
- les opérateurs permettant de diversifier la population au cours des générations et d'explorer l'espace de recherche. L'opérateur de croisement recompose les gènes d'individus existant dans la population, l'opérateur de mutation a pour but de garantir l'exploration de l'espace de recherche ;
- les valeurs pour les paramètres qu'utilise l'algorithme (taille de la population, nombre total de générations ou critère d'arrêt, probabilités de croisement et de mutation).

### II.4.1 Fonctionnement d'un algorithme génétique

L'AG débute par la génération d'une population initiale et l'évaluation de la fonction d'adaptation (évaluation) de tous les individus qui composent cette première population. Puis, des individus sont sélectionnés aléatoirement pour la reproduction selon le principe de la survie du plus adapté. Ensuite, des individus « enfants » (ou les descendants) sont générés en appliquant les deux opérateurs génétiques suivants : le croisement et la mutation. Ces enfants sont placés dans une nouvelle population  $P(g)$  et vont se substituer, en tout ou en partie, à la population de la génération précédente. De nouvelles populations d'individus vont ensuite se succéder, d'une génération ( $g$ ) à la génération ( $g+1$ ), chaque génération représentant une itération jusqu'à l'atteinte du critère d'arrêt [20, 16].

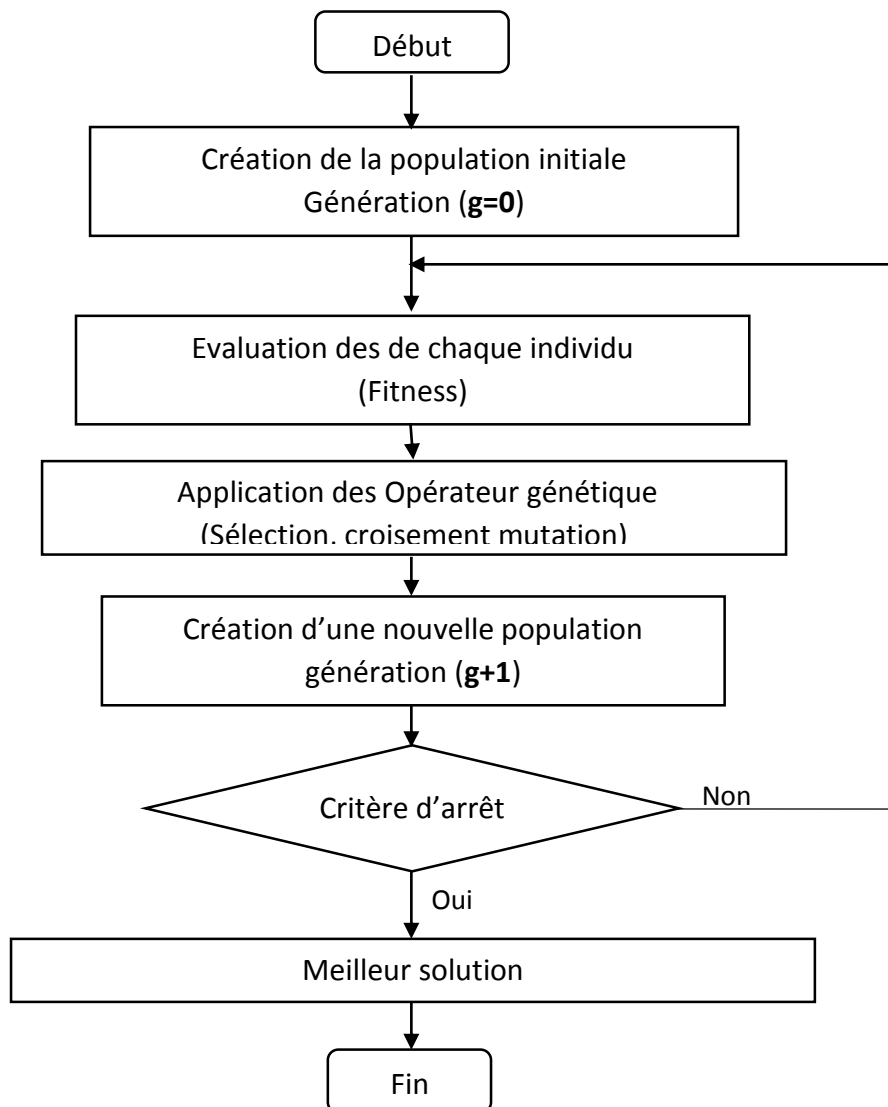


Fig. II-1 : Organigramme d'un algorithme génétique

## II.5 Les opérateurs génétiques

### II.5.1 Codage

Selon les chercheurs dans ce domaine, la méthode actuelle à appliquer dans la recherche d'un type de codage adapté consiste à choisir le codage qui semble le plus naturel en fonction du problème à traiter et à développer ensuite l'algorithme de traitement [22].

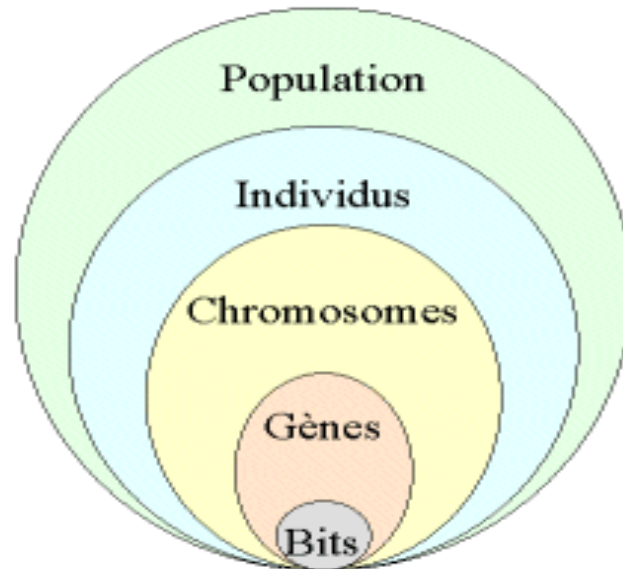


Fig. II-2 : Les cinq niveaux d'organisation d'un algorithme génétique [22]

Le codage est une modélisation d'une solution d'un problème donné sous forme d'une séquence de caractères appelée chromosome où chaque caractère, dit aussi gène dont le gène est une chaîne de bits (Fig. II-2), représente une variable ou une partie du problème. La tâche principale consiste à choisir le contenu des gènes qui facilite la description du problème et respecte ses contraintes [18].

Il existe plusieurs types de codage, on cite ci-dessous les plus couramment utilisés :

#### II.5.1.1 Codage binaire

Le codage binaire est un codage élémentaire dont le principe consiste à coder la solution selon une chaîne de bits. Une chaîne de bits est une suite de chiffres, chacun d'entre eux pouvant prendre la valeur 0 ou 1. Ce type de codage

est le plus utilisé. Il présente l'avantage d'être facile et permet de coder toutes sortes d'objets : des réels, des entiers, des chaînes de caractères, etc. [21, 16].

### II.5.1.2 Codage réel

Il est plus pratique d'utiliser un codage réel des chromosomes. Un gène est ainsi représenté par un nombre réel au lieu d'avoir à coder les réels en binaire puis de les décoder pour les transformer en solutions effectives [22].

### II.5.1.3 Codage sous forme d'arbre

Ce codage utilise une structure arborescente ; un arbre est une structure de données munie d'une racine de laquelle peuvent être issus un ou plusieurs fils (Fig. II-3). Un de ses avantages est qu'ils peuvent être utilisés dans le cas de problèmes où les solutions n'ont pas une taille finie. Les performances de ce type de codage, par rapport à des codages en chaînes sont encore mal connues [22].

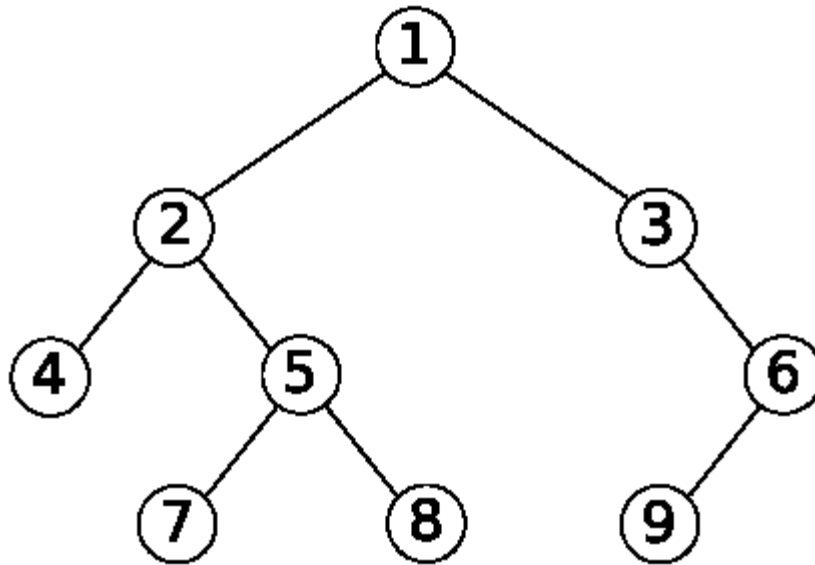


Fig. II-3 : simple arbre binaire [47]

## II.5.2 Population initiale

Après avoir choisi le type de codage, la création de la population initiale peut se faire par plusieurs mécanismes de génération. Le choix de l'initialisation se fera en fonction des connaissances que l'utilisateur a sur le problème. S'il n'a pas

d'informations particulières, alors une initialisation aléatoire, la plus uniforme possible afin de favoriser une exploration de l'espace de recherche maximum, sera la plus adaptée. Mais dans d'autres cas, il est possible d'utiliser d'autres mécanismes. Par ailleurs, cette étape présente un problème principal qui est celui du choix de la taille de la population. En effet une population trop grande augmente le temps de calcul et demande un espace mémoire considérable, alors qu'une population trop petite conduit à l'obtention d'un optimum local [18].

### II.5.3 Fonction d'évaluation

La fonction d'évaluation ou adaptation ou fitness mesure l'efficacité de la solution. Elle permet, à partir d'un chromosome, de calculer le coût d'un point de l'espace de recherche, de quantifier numériquement la validité de la solution qu'il représente et de mesurer la santé et le degré d'adaptation d'un individu à son environnement [22].

Toutefois, des problèmes liés à l'évaluation peuvent handicaper la convergence d'un algorithme génétique : si un génotype domine la population très rapidement en conférant à certains individus une adéquation élevée, l'exploitation de l'espace de recherche peut être inhibée car le croisement ne produit presque plus de nouvelles solutions et seule la mutation permet alors de diversifier la population, ce qui en général n'est pas suffisant, l'algorithme risque alors de converger prématurément vers un optimum local. Le problème opposé peut également survenir. Si tous les individus ont des adéquations très voisines, par exemple après un grand nombre de générations, la sélection ne parviendra pas à faire évoluer la population vers l'optimum car les meilleurs individus ne seront pas assez favorisés [6].

### II.5.4 Sélection

L'opérateur de la sélection permet d'identifier statistiquement les meilleurs individus de la population courante qui seront autorisés à se reproduire. La sélection ne crée pas de nouveaux individus mais identifie les individus sur la base de leur performance, estimée à l'aide de la fonction d'adaptation.

Ce procédé permet de donner aux meilleures chaînes, une probabilité élevée de contribuer à la génération suivante. Cet opérateur est bien entendu une

version artificielle de la sélection naturelle, la survie darwinienne des chaînes les plus adaptées [16, 20, 17].

Pour la sélection, il existe nombreuse technique. On citera ci-dessous les plus connues et utilisées :

#### II.5.4.1 Sélection par roulette

La sélection par roulette ou roulette artificielle (Roulette Wheel) consiste à créer une roue de loterie biaisée pour laquelle chaque individu de la population occupe une section de la roue proportionnelle à sa valeur d'évaluation. Donc plus les individus sont adaptés au problème, plus ils ont de chances d'être sélectionnés. Ainsi, même les individus les plus faibles ont une chance de survivre. [16, 18]

Cette technique de sélection est la plus utilisée mais elle présente des inconvénients, elle a une forte variance. Il n'est pas impossible que sur un nombre des sélections successives destinées à désigner les parents de la nouvelle génération, la quasi-totalité des individus sélectionnés soient des individus ayant une fitness mauvaise et donc que pratiquement aucun individu a forte fitness ne fasse partie des parents de la nouvelle génération. Ce phénomène est bien sûr très dommageable car cela va complètement à l'encontre du principe des algorithmes génétiques qui veut que les meilleurs individus soient sélectionnés de manière à converger vers une solution la plus optimale possible.

D'autre part, A l'inverse, on peut arriver à une domination écrasante d'un individu "localement supérieur". Ceci entraînant une grave perte de diversité. Imaginons par exemple qu'on ait un individu ayant une fitness très élevée par rapport au reste de la population, disons dix fois supérieure, il n'est pas impossible qu'après quelques générations successives on se retrouve avec une population ne contenant que des copies de cet individu. Le problème est que cet individu avait une fitness très élevée, mais que cette fitness était toute relative, elle était très élevée mais seulement en comparaison des autres individus. On se retrouve donc face à problème connu sous le nom de "convergence prématurée; l'évolution se met donc à stagner et on atteindra alors jamais l'optimum, on restera bloqué sur un optimum local [16].

#### II.5.4.2 Sélection par classement

Cette méthode ou technique de sélection consiste à ranger les individus de la population dans un ordre croissant (ou décroissant selon l'objectif) et à retenir un nombre fixé de génotypes. Ainsi, seuls les individus les plus forts sont conservés. L'inconvénient majeur de cette méthode est la convergence prématurée de l'algorithme génétique. Il est parfois nécessaire de garder quelques individus jugés faibles pour créer la diversité au niveau de la population. Une autre difficulté consiste à fixer une limite à la sélection ce qui empêche parfois de garder des bons candidats pour les futures générations [18].

#### II.5.4.3 Sélection par tournoi

Elle consiste à tirer aléatoirement deux ou plusieurs individus et à sélectionner l'individu qui a la performance la plus élevée. La nouvelle population avec une probabilité comprise entre 0.5 et 1. Le fait d'augmenter ou de diminuer la valeur de la probabilité permet respectivement de diminuer ou d'augmenter la pression de sélection. Cette méthode permet d'obtenir des résultats très satisfaisants [17, 16].

#### II.5.4.4 La méthode élitiste

Cette méthode consiste à sélectionner les  $n$  individus dont on a besoin pour la nouvelle génération en prenant les meilleurs individus de la population  $P$  après l'avoir triée de manière décroissante ou croissante selon la fitness de ses individus. Il est inutile de préciser que cette méthode est encore pire que celle de la loterie biaisée dans le sens où elle amènera à une convergence prématurée encore plus rapidement et surtout de manière encore plus sûre que la méthode de sélection de la loterie biaisée ; en effet, la pression de la sélection est trop forte, la variance nulle et la diversité inexistante, du moins le peu de diversité qu'il pourrait y avoir ne résultera pas de la sélection mais plutôt du croisement et des mutations.

#### II.5.5 Croisement ou reproduction

L'application de principe de croisement inspiré du phénomène naturelle, permet de générer de nouveaux couple individu appelés enfants (descendants) à partir d'un couple individus existants appelés parents.

C'est par analogie avec la notion d'hydratation de Darwin qu'a été conçu cet opérateur.

Dans le domaine des algorithmes génétiques on distingue selon le type de codage adapté et la nature du problème traité plusieurs opérateurs de croisement.

### II.5.5.1 Croisement binaire

L'opérateur de croisement le plus utilisé est celle de croisement à un point, dont ce point est choisi aléatoirement, puis la chaîne des individus parents après ou avant ce point sont inter-changées pour former les individus descendants. La figure ci-dessous illustre cette étape [18, 16].

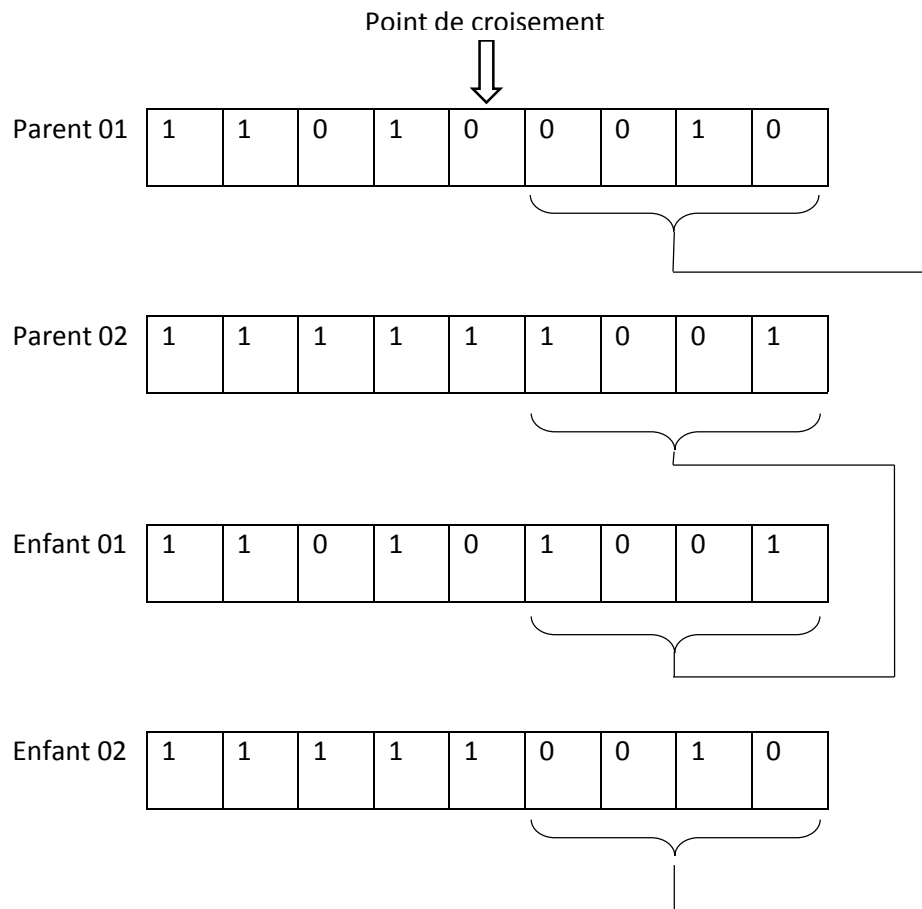


Fig. II-4 : Schéma d'un croisement à un point

L'opérateur peut généraliser pour deux ou plusieurs points de croisement en suivant les mêmes étapes et les points du croisement sont tirés aléatoirement.

### II.5.5.2 Croisement réel :

Ce type de croisement est conçu dans le cas où les individus ayant un codage réel. Plusieurs types de croisement sont à appliquer, croisement à Ordre de base cyclique, Croisement uniformément continu et Croisement d'ordre maximal. [18]

En effet L'opérateur de croisement favorise l'exploration de l'espace de recherche. Cet opérateur permet de créer des nouveaux individus descendant soient plus adapté que ses parents [16].

### II.5.6 Mutation

La mutation est définie dans l'évolution naturelle comme un processus aléatoire où un allèle d'un gène est remplacé par un autre pour produire une nouvelle structure génétique. Elle présente la modification aléatoire et occasionnelle d'une partie d'un chromosome.

L'opérateur de la mutation constitue une exploration aléatoire de l'espace des chaînes. Elle joue le rôle de bruit et permettent d'assurer une recherche aussi bien globale que locale [17, 16].

La mutation permet de limiter les risques d'une convergence prématurée et d'atteindre la propriété d'ergodicité qui garantissant que chaque point de l'espace de recherche puisse être atteinte [16].

#### II.5.6.1 Mutation binaire

Dans le cas du codage binaire, la mutation est la modification de la valeur d'un caractère de la chaîne d'un 1 en 0 et vice-versa. (Fig. II-5)

```
1 1 1 0 0 0 1 0 0 1 1 1 1
1 1 1 0 0 0 0 0 0 1 1 1 1
```

Fig. II-5 : schéma d'une mutation en code binaire

### II.5.6.2 Mutation réelle

Les opérateurs de la mutation les plus couramment utilisés dans le cas d'un codage réel sont les suivants [18]:

- La mutation d'inversion simple : consiste à choisir aléatoirement deux points de coupure et inverser les positions des bits situés au milieu ;
- L'opérateur d'insertion : consiste à sélectionner au hasard un bit et une position dans le chromosome à muter, puis à insérer le bit en question dans la position choisie ;
- L'opérateur d'échange réciproque : cet opérateur permet la sélection de deux bits et les inter-changés.

## II.6 Optimisation des réseaux des neurones artificiels par algorithme génétique

La création d'un système hybride neuro-génétique nous permet de profiter des avantages réseaux de neurones artificiels et des algorithmes génétiques.

Les réseaux neuronaux artificiels avec leur capacité d'apprendre et de généraliser, de plus les algorithmes génétiques avec leur caractère de résoudre automatiquement les problèmes sans connaître ou préciser la forme ou la structure de la solution à l'avance et la possibilité d'explorer un espace de recherche, nous donnent la possibilité d'avoir un système hybride plus fiable pour traiter les problèmes.

Dans notre travail nous optimisons un réseau neuronal Feedforward avec un apprentissage supervisé basé sur l'algorithme de Levenberg Marquardt par un algorithme génétique avec les opérateurs génétiques suivants :

- Codage binaire ;
- Sélection par la méthode élitiste;
- Croisement à un et deux points ;
- Mutation binaire.

### II.6.1.1 Les paramètres optimiser par algorithme génétique

Dans le cadre de ce travail on cherche à optimiser les paramètres liés au réseau neuronal (les poids synaptiques) et un paramètre lié à l'apprentissage (algorithme Levenberg Marquardt). Ce sont les suivants :

- Les poids synaptiques : l'optimisation de ces poids nous permet de résoudre le problème d'initialisation de ces derniers donc d'éviter le problème de saturation ;
- Le paramètre  $\mu$  (équation I-5) : ce paramètre est lié à l'algorithme de Levenberg Marquardt (chapitre I page 21).

### II.6.1.2 Organigramme de système hybride neuro-génétique

La figure ci-dessous représente l'organigramme d'optimisation du réseau des neurones artificiels par algorithme génétique en suivant les étapes suivant :

1. Définition des paramètres (maximums, minimums, nombre générations, nombre d'individus,..., etc.) ;
2. La création de de la population initiale ;
3. Evaluation des individus ;
4. Application des opérateurs génétiques (sélection, croisement, mutation) ;
5. Evaluation des individus (appel au réseau de neurones artificiels) ;
6. Sauvegarder l'individu le plus fiable ; Evaluation des individus ;
7. Refaire les étapes 4 à 6 jusqu'à la satisfaction du critère d'arrêt (nombre de génération ou valeur de fonction fitness);

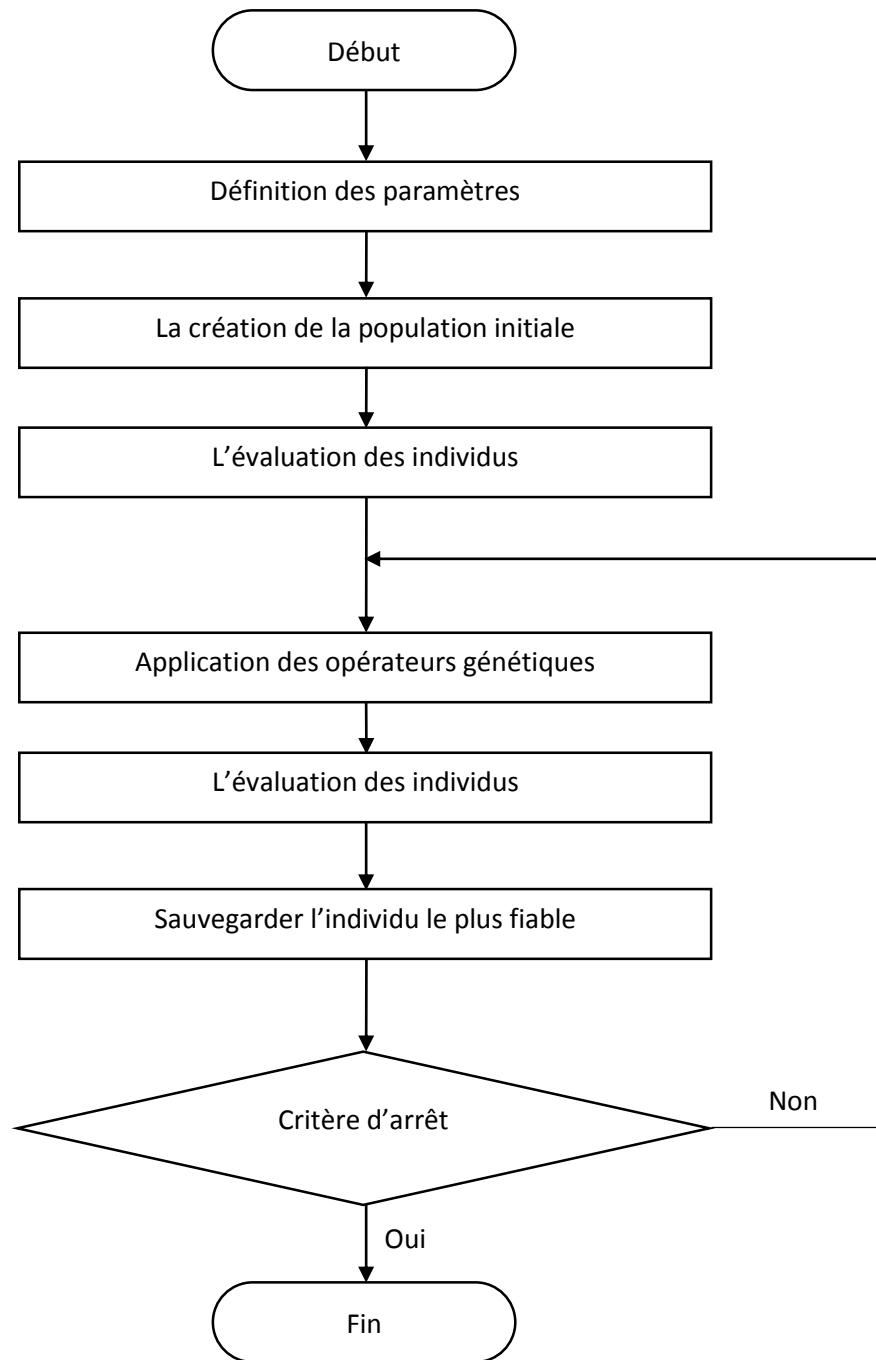


Fig. II-6 : organigramme du système hybride neuro-génétique

A la fin de ce chapitre 2, nous avons défini tous les composants de notre système hybride neuro-génétique et il ne nous reste que l'application de ce dernier, c'est ce que nous allons développer dans le chapitre 3.

# Chapitre III

## Prévision des débits par l'approche hybride neuro-génétique

Ce chapitre contient deux parties. Dans la première on présentera la structure du programme et ses paramètres, tandis que dans la deuxième partie nous présenterons des exemples d'application pour la prévision des débits dans des bassins hydrographiques et discuterons les résultats obtenus.

### III.1 Structure du programme et la mise en œuvre pour un exemple théorique

#### III.1.1 Réseau des neurones artificiels (RNA)

##### III.1.1.1 Les paramètres d'évaluation de performance du modèle

Les mesures qui nous permettent d'évaluer le modèle hybride neuro-génétique sont citées ci-dessous :

- **Erreur quadratique moyenne (MSE)**

Cette erreur est défini comme étant la moyenne arithmétique des carrés des écarts entre les prévisions et les observations. Sa formule est la suivante :

$$MSE = \frac{\sum_{i=1}^n (Obs_i - Pre_i)^2}{n} \quad \text{Équation III-1}$$

Où : **Obs<sub>i</sub>** : Valeurs observées ; **Pre<sub>i</sub>** : Valeurs prévues (simulées)

**n** : Nombre des valeurs.

- **Coefficient de détermination R<sup>2</sup>**

Ce coefficient est défini comme étant le carré du coefficient de corrélation selon Bravins-Person. Sa valeur peut calculer à partir de la formule suivante :

$$R^2 = \left( \frac{\sum_{i=1}^n (Obs_i - \overline{Obs})(Pre_i - \overline{Pre})}{\sqrt{\sum_{i=1}^n (Obs_i - \overline{Obs})^2} \sqrt{\sum_{i=1}^n (Pre_i - \overline{Pre})^2}} \right)^2 \quad \text{Équation III-2}$$

où :  $\overline{Obs}_i$  : moyenne des Valeurs observées ;  
 $\overline{Pre}_i$  : moyenne des Valeurs prévues (simulées).

La valeur du coefficient de détermination varie entre 1 et 0. La valeur 0 signifie qu'aucune corrélation n'existe entre les observations et les valeurs prévues alors que  $R^2 = 1$  indique qu'il existe une forte relation linéaire positive entre la prédiction et l'observation [23].

- **Coefficient d'efficience CE (Nash–Sutcliffe)**

Nash et Sutcliffe définissent ce coefficient comme **Un** moins la somme des carrés des différences absolues entre les valeurs prévues (simulées) et observées normalisées par la variance des valeurs observées au cours de la période considérée. Sa valeur peut être calculée par la formule suivante :

$$CE = 1 - \frac{\sum_{i=1}^n (Obs_i - Pre_i)^2}{\sum_{i=1}^n (Obs_i - \overline{Obs})^2} \quad \text{Équation III-3}$$

La plage des valeurs de CE se trouve entre 1 et  $-\infty$ . Un CE = 1 indique un ajustement parfait tandis que la valeur moyenne de la série chronologique observée aurait été un meilleur prédicteur que le modèle [23].

### III.1.1.2 Structure du réseau

Note réseau neuronal artificiel feedforward contient une seule couche cachée et une couche sortie. La couche cachée a un nombre des neurones qui sera déterminé ultérieurement, tandis que la couche sortie ne contient qu'un seul neurone.

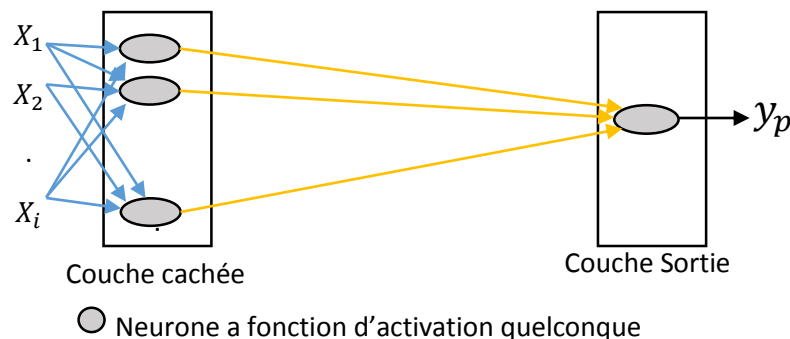


Fig. III-1 : Schéma du réseau neuronal du projet

### III.1.1.3 Fonction d'activation

Les fonctions de transfert ou d'activation utilisée dans notre projet sont les suivantes :

- Les neurones de la couche cachée ont une fonction sigmoïdale tangente hyperbolique bornée à valeurs réelles comprises entre -1 et +1;
- Le neurone de la couche sortie a une fonction linéaire où la fonction identité  $F(v_k) = v_k$  .

### III.1.1.4 Fonction de normalisation des données

Dans le but d'éviter la saturation pour la fonction d'activation sigmoïdale des neurones de la couche cachée et d'accélérer l'apprentissage on a admis une normalisation dans l'intervalle  $[-1,1]$ . La fonction utilisées est la fonction de normalisation MINMAX.

La normalisation a été faite selon la formule suivante :

$$X_i^{nor} = \frac{(\max(x_i) - \min(x_i))(x_i - \min(x_i))}{(\max(x_i) - \min(x_i))} + \min(x_i)$$

Ou :  $X_i^{nor}$  : valeur d'entrée normalisée ;  $x_i$  : valeur d'entrée

### III.1.1.5 Description d'exemple théorique de test

Les données de l'exemple théorique de test sont des données générées à partir de pluies réelles de Bejaia à l'aide d'une fonction non-linéaire régressive bruitées pour simuler les débits de l'oued Soummam correspondant à 13 009 valeurs journalières des pluie et débits.

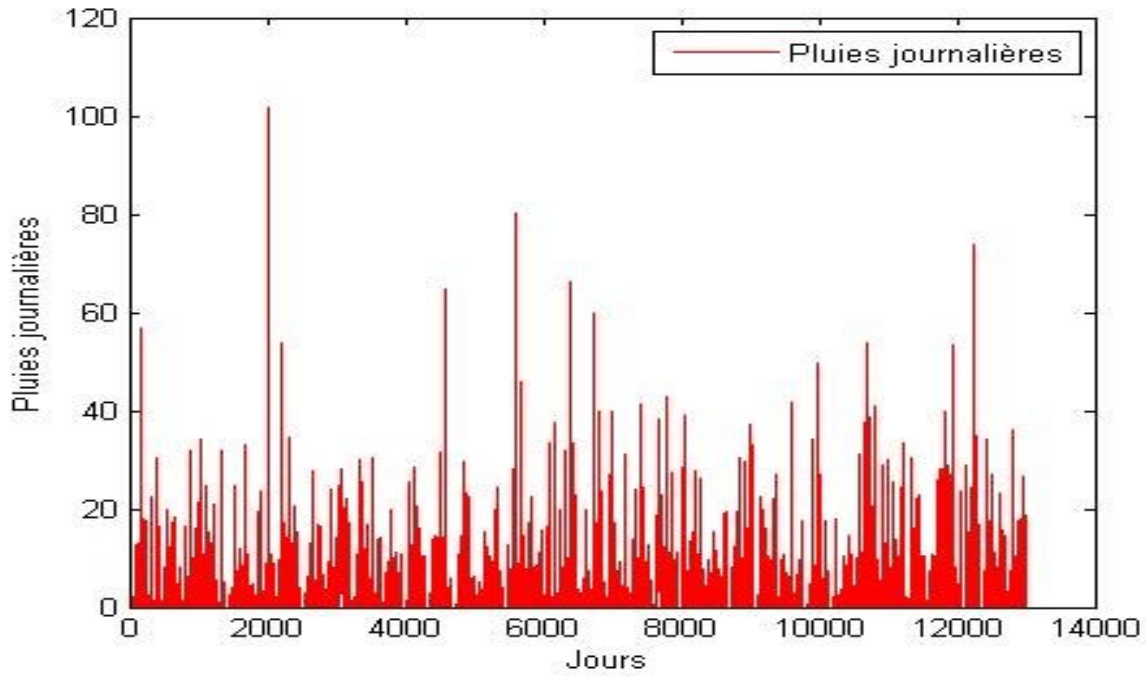


Fig. III-2 : les pluies journalières à la station de Bejaia

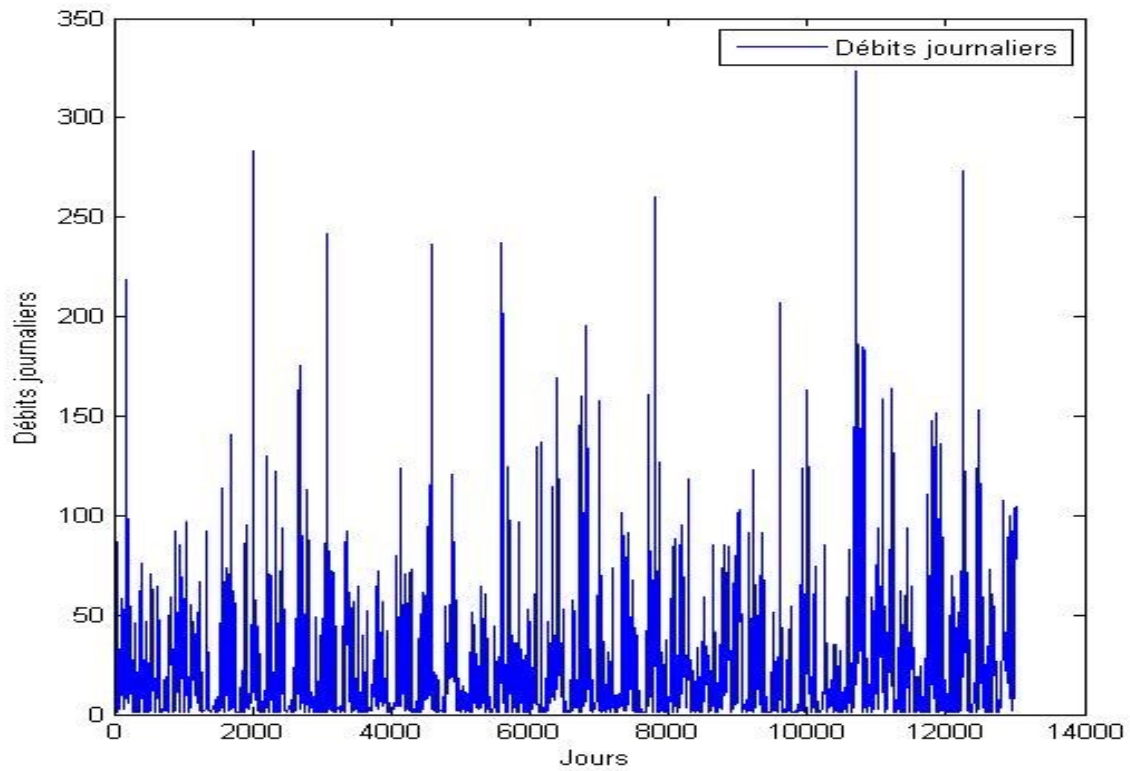


Fig. III-3 : séries des débits journaliers théoriques générés

Les paramètres statistiques de ces séries sont représentés dans le tableau suivant :

Tableau III-1 : les paramètres statistique des séries test

Paramètres statistiques	pluies	débits
Minimum	0	0.0990
Maximum	101.60	323.4867
Moyen	1.1777	24.6129
Ecart type	4.2830	32.5200
Coefficient de variation	3.6367	1.3213

#### III.1.1.6 Initialisation des poids et biais

Ce sont des paramètres liés aux neurones des réseaux. L'étape d'initialisation de ces paramètres est la première à effectuer, généralement, elle est faite aléatoirement.

Tandis que le choix des valeurs de ces paramètres influence tout le fonctionnement du réseau de neurones artificiels et pour mieux étudier l'efficacité de ces réseaux, il est primordial de déterminer les valeurs optimales de ces paramètres. On a utilisé un intervalle d'initialisation de -1 à 1 qui est le plus couramment utilisé.

Fixer les valeurs initiales des poids et biais permet d'éviter l'effet du tirage aléatoire de ces valeurs initiales ce qui est chiffré dans Tableau (III-2) et représenter graphiquement (Fig. III-4) en comparant les valeurs d'erreur quadratique moyenne (MSE).

L'exemple d'illustration est une prévision des débits à partir des données réelles de pluies que nous exposerons ultérieurement et les poids et biais sont fixés à une valeur égale à 1.

Tableau III-2 : comparaison entre l'initialisation aléatoire et fixée

Lancement du calcul	MSE	
	Initialisation aléatoire	Initialisation fixée à 1
01	26,32621	26,2164
02	26,3361	26,2164
03	26,3409	26,2164
04	26,3238	26,2164
05	26,3143	26,2164
06	26,3239	26,2164
07	26,3158	26,2164
08	26,5451	26,2164
09	26,3157	26,2164
10	26,3172	26,2164

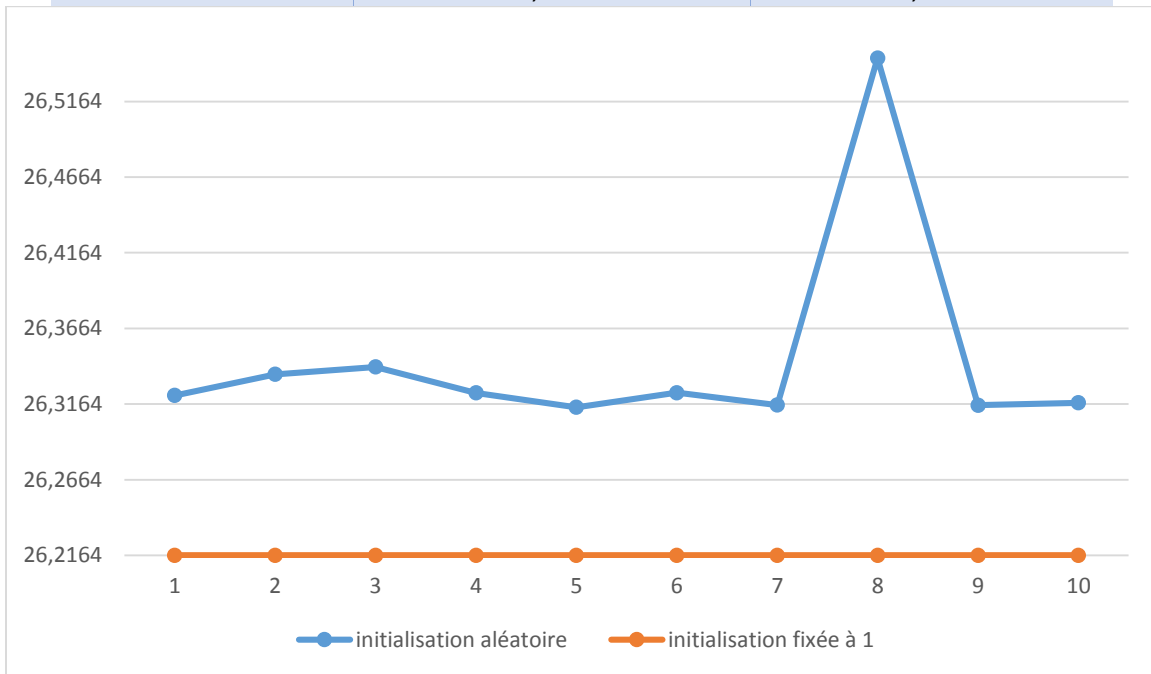


Fig. III-4 : représentation graphique de l'effet de l'initialisation aléatoire des poids et biais

### III.1.1.7 Modèle des données (Pluies, Débits)

Les débits prévus du jour suivant  $Q(t+1)$  sont les résultats de la modélisation pluies-débits par le réseau de neurones artificiels. La détermination du modèle le plus approprié est une étape essentielle afin de profiter au maximum de la plage des données observées. Les modèles que nous proposons illustrent l'effet de la pluie et le débit des jours précédents sur la prévision des débits pour le jour suivant  $Q(t+1)$ .

$$Q_{t+1} = RNA(P_{t-i}, \dots, P_t, Q_{t-i}, \dots, Q_t) \quad \text{Équation III-4}$$

où : **RNA** : signifie la modélisation par le réseau de neurones artificiels

Étant donnée la dynamique des bassins, on a admis des modèles allant du jour précédent P(t-1) au jour P(t) pour les pluies, alors que les débits allant jusqu'au jour Q(t-10). (Tableau III-3)

Tableau III-3 : les modèles des pluies-débits

Modèle
$Q_{t+1} = RNA(P_{t-1}, P_t, Q_{t-1}, Q_t)$
$Q_{t+1} = RNA(P_{t-1}, P_t, Q_{t-2}, Q_{t-1}, Q_t)$
$Q_{t+1} = RNA(P_{t-1}, P_t, Q_{t-3}, Q_{t-2}, Q_{t-1}, Q_t)$
$Q_{t+1} = RNA(P_{t-1}, P_t, Q_{t-4}, \dots, Q_{t-1}, Q_t)$
$Q_{t+1} = RNA(P_{t-1}, P_t, Q_{t-5}, \dots, Q_{t-1}, Q_t)$
$Q_{t+1} = RNA(P_{t-1}, P_t, Q_{t-6}, \dots, Q_{t-1}, Q_t)$
$Q_{t+1} = RNA(P_{t-1}, P_t, Q_{t-7}, \dots, Q_{t-1}, Q_t)$
$Q_{t+1} = RNA(P_{t-1}, P_t, Q_{t-8}, \dots, Q_{t-1}, Q_t)$
$Q_{t+1} = RNA(P_{t-1}, P_t, Q_{t-9}, \dots, Q_{t-1}, Q_t)$
$Q_{t+1} = RNA(P_{t-1}, P_t, Q_{t-10}, \dots, Q_{t-1}, Q_t)$

### III.1.1.8 Nombre des neurones optimal

La détermination de ce nombre permet d'avoir un bon apprentissage donc une bonne prévision. Pour notre étude la détermination de ce nombre se fait après l'initialisation fixée des poids et biais. On a exécuté une boucle allant de 1 à 100 pour déterminer le nombre optimal en tenant compte de la valeur de MSE minimale. L'organigramme est présenté dans la figure (III-6). Pour une valeur de MSE = 17.0631 Le réseaux de neurone doit avoir 50 neurones dans la couche cachés pour le modèle :

$$Q_{t+1} = RNA(P_{t-1}, P_t, Q_{t-9}, \dots, Q_{t-1}, Q_t) \quad \text{Équation III-5}$$

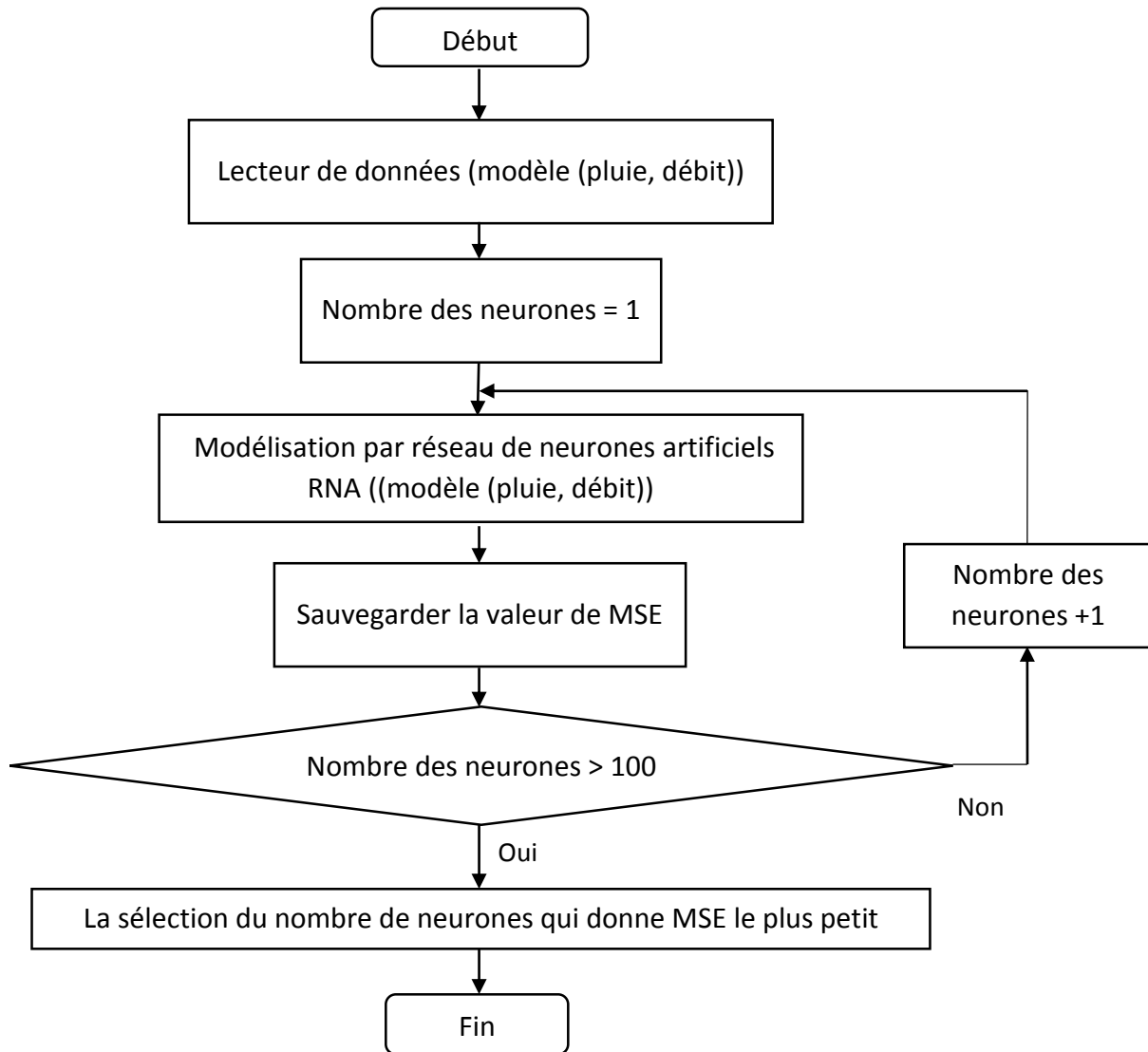


Fig. III-5 : Organigramme de détermination du nombre de neurones

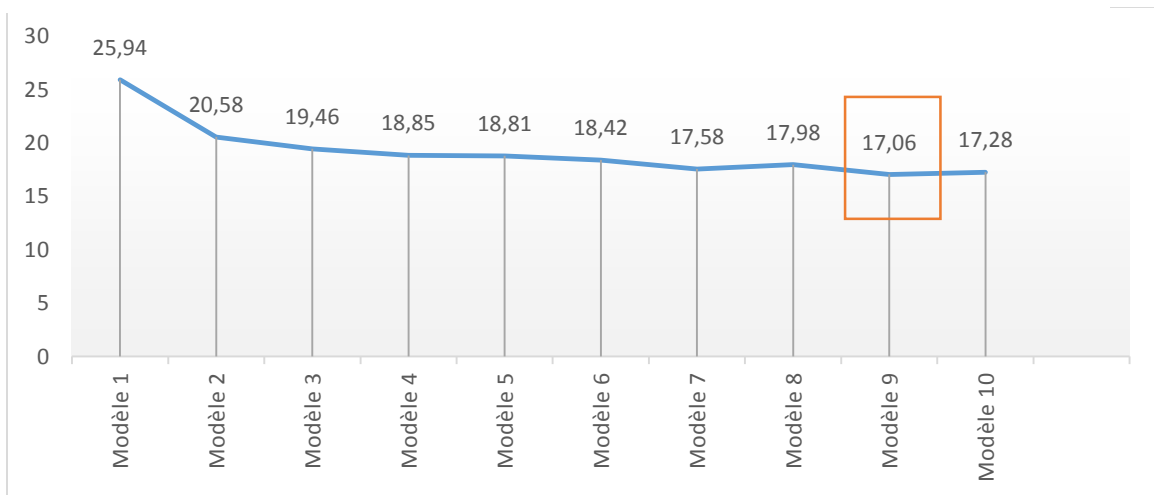


Fig. III-6 : Sélection du modèle pluie-débit en fonction MSE

### III.1.1.9 Mise en œuvre du programme RNA pour l'exemple de test

L'étape finale c'est l'application du programme RNA pour l'exemple de test (théorique) après avoir déterminé le nombre optimal des neurones de la couche cachée et le modèle pluies-débits le plus fiable. Les poids et biais sont initialisés à 1. Les données d'entrées sont divisées en trois parties, la première contient 70% de la plage des données pour l'apprentissage, 15% pour validation et 15% des données pour le test.

Les débits prévus sont représentés dans la figure (III-7 et III-8) en comparaison avec des débits observés. Le tableau (III-4) illustre une comparaison entre les paramètres statistiques des débits prévus et observés. Les résultats obtenus sont très satisfaisantes.

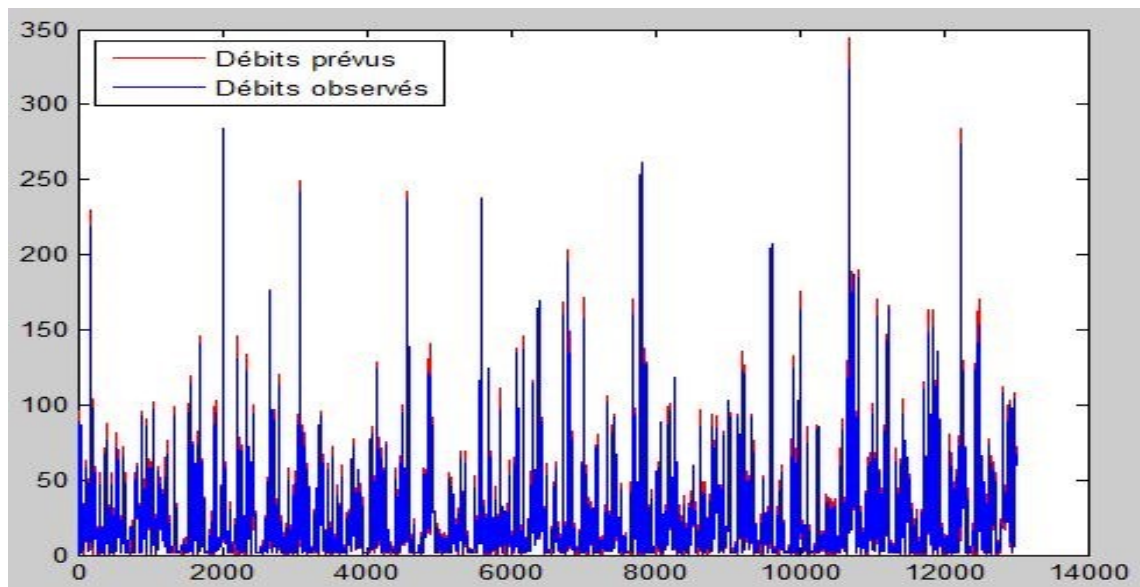


Fig. III-7 : comparaison débits prévus par RNA aux débits observés

Tableau III-4 : Comparaison des paramètres statistiques des débits prévus par RNA et observés

Paramètres statistiques	Débits prévus	Débits observés
<b>Minimum</b>	0.0166	0.0990
<b>Maximum</b>	330.2594	323.4867
<b>Moyen</b>	24.5429	24.5686
<b>Ecart type</b>	32.0664	32.4863
<b>Coefficient de variation</b>	1.3065	1.3223

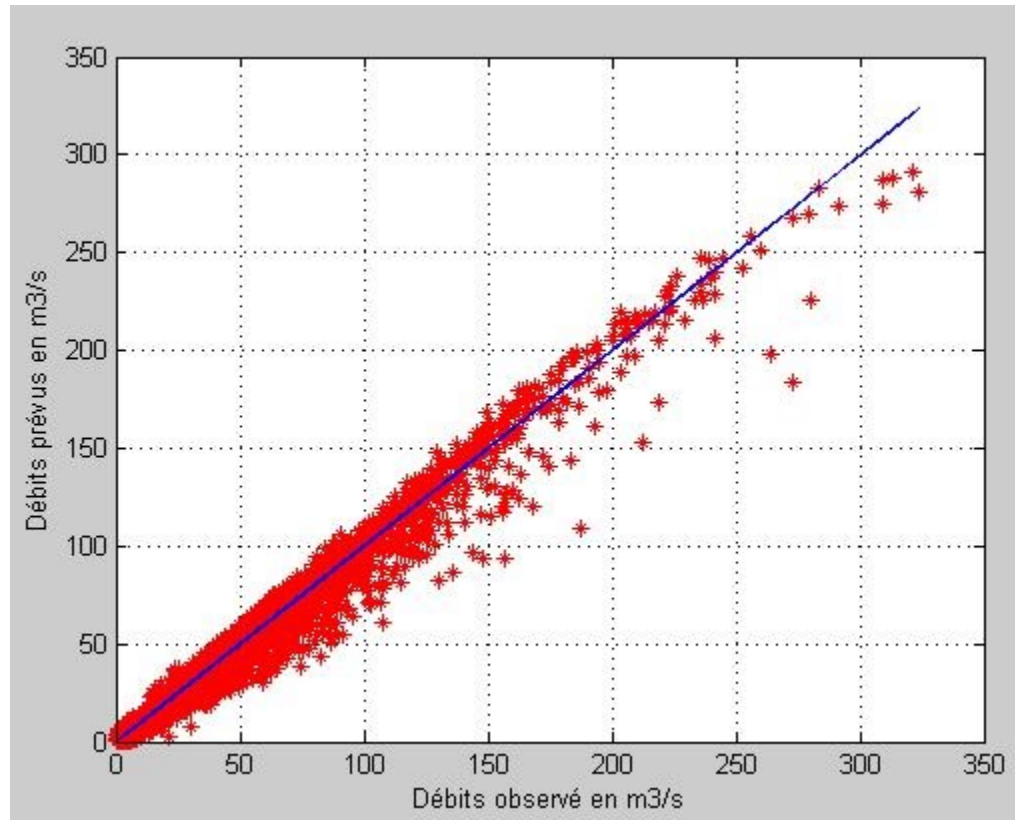
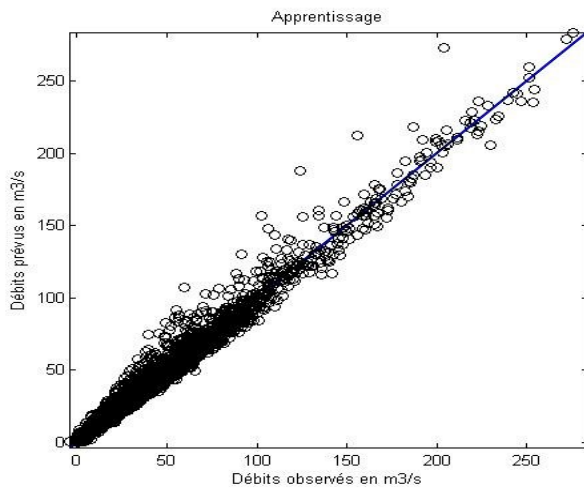


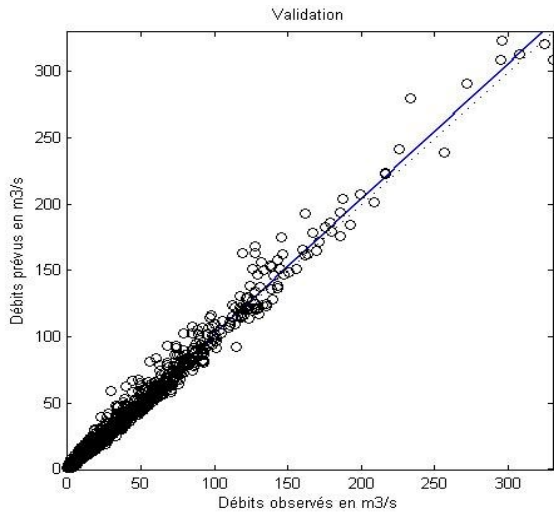
Fig. III-8 : Comparaison entre les débits prévus par RNA et observés

Les résultats obtenus au cours des trois phases sont représentées respectivement comme suit :



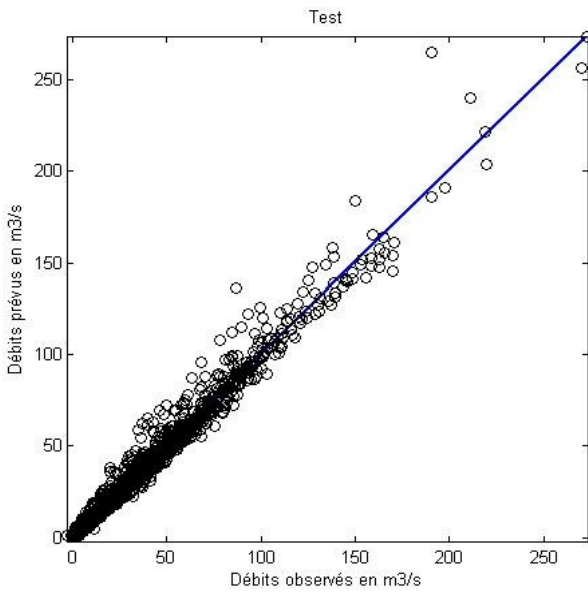
Paramètres statistiques	Débits prévus	Débits observés
<b>Minimum</b>	0.0219	0.0990
<b>Maximum</b>	275.9959	283.5000
<b>Moyen</b>	23.1875	23.1666
<b>Ecart type</b>	30.3179	30.5683
<b>Coefficient de variation</b>	1.3075	1.3195

Fig.III-9 : Comparaison les entre les débits prévus et débits observés et les paramètres statistique phase apprentissage



Paramètres statistiques	Débits prévus	Débits observés
Minimum	0.0166	1.1000
Maximum	330.2594	323.4857
Moyen	26.1305	26.3050
Ecart type	37.2010	38.2580
Coefficient de variation	1.4237	1.4544

Fig. III-10 : Comparaison entre les débits prévus et débits observés et les paramètres statistique phase de Validation



Paramètres statistiques	Débits prévus	Débits observés
Minimum	0.0307	1.1000
Maximum	272.6246	273.2429
Moyen	29.2811	29.3754
Ecart type	33.9024	34.3149
Coefficient de variation	1.1578	1.1681

Fig. III-11 : Comparaison entre les débits prévus et débits observés et les paramètres statistique phase de test

### III.1.2 Algorithme génétique (AGRNA)

#### III.1.2.1 Fonction d'évaluation

On prend l'erreur quadratique MSE comme une fonction de fitness pour chaque individu. La formule qui nous permet de calculer le MSE est donnée dans chapitre III équation (III-1).

#### III.1.2.2 Le nombre des paramètres à optimiser

Ce nombre est défini en fonction du nombre de neurones de la couche cachée et la couche sortie et les paramètres de l'algorithme Levenberg Marquardt ainsi que le modèle pluies-débits. Le nombre des variables à optimiser est calculé par l'équation suivante :

$$Npp = (NN * (C_m + 2) + 1) + 2$$

Où :  $Npp$  : Nombre des variables à optimiser

$NN$  : Le nombre de neurone de la couche cachée

$C_m$  : Nombre des colonnes du vecteur d'entrée

Cette équation est valable seulement sous les conditions suivantes :

- Le réseau a une couche cachée et une couche sortie contenant un seul neurone ;
- Seul Le paramètre d'augmentation et amortissement sont optimisés pour l'algorithme Levenberg Marquardt.

#### III.1.2.3 Création de la population initiale

La dimension de la population initiale est déterminée en fonction du nombre d'individus et du nombre de gènes alloué au variable à optimiser. On peut écrire cette population sous forme matricielle.

$$pop_{initiale} = \begin{bmatrix} gene_{1,1} & \cdots & gene_{1,n} & w_{1,1} & \cdots & w_{1,i} & \mu_{amor(1)} & \mu_{aug(1)} & MSE_1 \\ \vdots & & & & & \vdots & & & \vdots \\ gene_{ind,1} & \cdots & gene_{ind,n} & w_{1,1} & \cdots & w_{1,i} & \mu_{amor(ind)} & \mu_{aug(ind)} & MSE_{ind} \end{bmatrix}$$

### III.1.2.4 Le nombre des gènes

Pour déterminer ce nombre on a effectué l'optimisation en utilisant l'exemple de test avec un seul neurone dans la couche cachée et 100 individus par population. Le nombre de gènes se modifie d'un pas égal à 6 gènes et nous avons commencé par 06 gènes par variable et les résultats sont résumés dans le tableau suivant :

*Tableau III-5 : déterminer le nombre des gènes en fonction du MSE*

<b>Nombre de gènes</b>	<b>MSE après optimisation</b>
<b>6</b>	25.1885
<b>12</b>	26.1950
<b>18</b>	26.2077
<b>24</b>	26.2004

On a admis 06 gènes pour présenter une variable à optimiser. Ce nombre de gènes permet d'avoir des nouvelles valeurs après chaque opération de croisement ou mutation. Un nombre inférieur à 6 peut donner de bons résultats mais le problème de saturation s'impose à cause des valeurs nulles.

### III.1.2.5 Le nombre des individus de la population

On a effectué les mêmes étapes pour le même modèle que pour le nombre de gènes et le tableau suivant résume les résultats obtenus :

*Tableau III-6 : détermination du nombre des individus de la population en fonction de MSE*

<b>Nombre d'individus</b>	<b>MSE après optimisation</b>
<b>50</b>	26.1931
<b>100</b>	26.1885
<b>110</b>	26.1912
<b>300</b>	26.1920

Nous admettons 100 individus par population dans le but de minimiser MSE.

### III.1.2.6 Sélection par la méthode élitisme

La population se trie de manière croissante selon le fitness (MSE) de ses individus. On calcule la probabilité de chaque individu. On a choisi la sélection suivante pour la création de la nouvelle génération :

- Les individus ayant une probabilité inférieure à 0.30 dans la population sont conservés (il contient toujours le meilleur individu) ;
- Les individus ayant une probabilité entre 0.30 et 0.60 dans la population sont soumis à un croisement à un point ;
- Les individus ayant une probabilité entre 0.60 et 0.90 dans la population sont soumis à un croisement à deux points ;
- Les individus ayant une probabilité supérieure à 0.90 dans la population sont soumis à une mutation;

Nous avons profité des avantages de cette méthode qui permet de garder toujours le meilleur individu. On a également diminué l'effet du problème de l'inexistence de diversité par l'augmentation de la portion des individus subissant le croisement 60%, et 10% des individus de la population subiront la mutation à chaque génération. L'organigramme de cette sélection se présente comme suit :

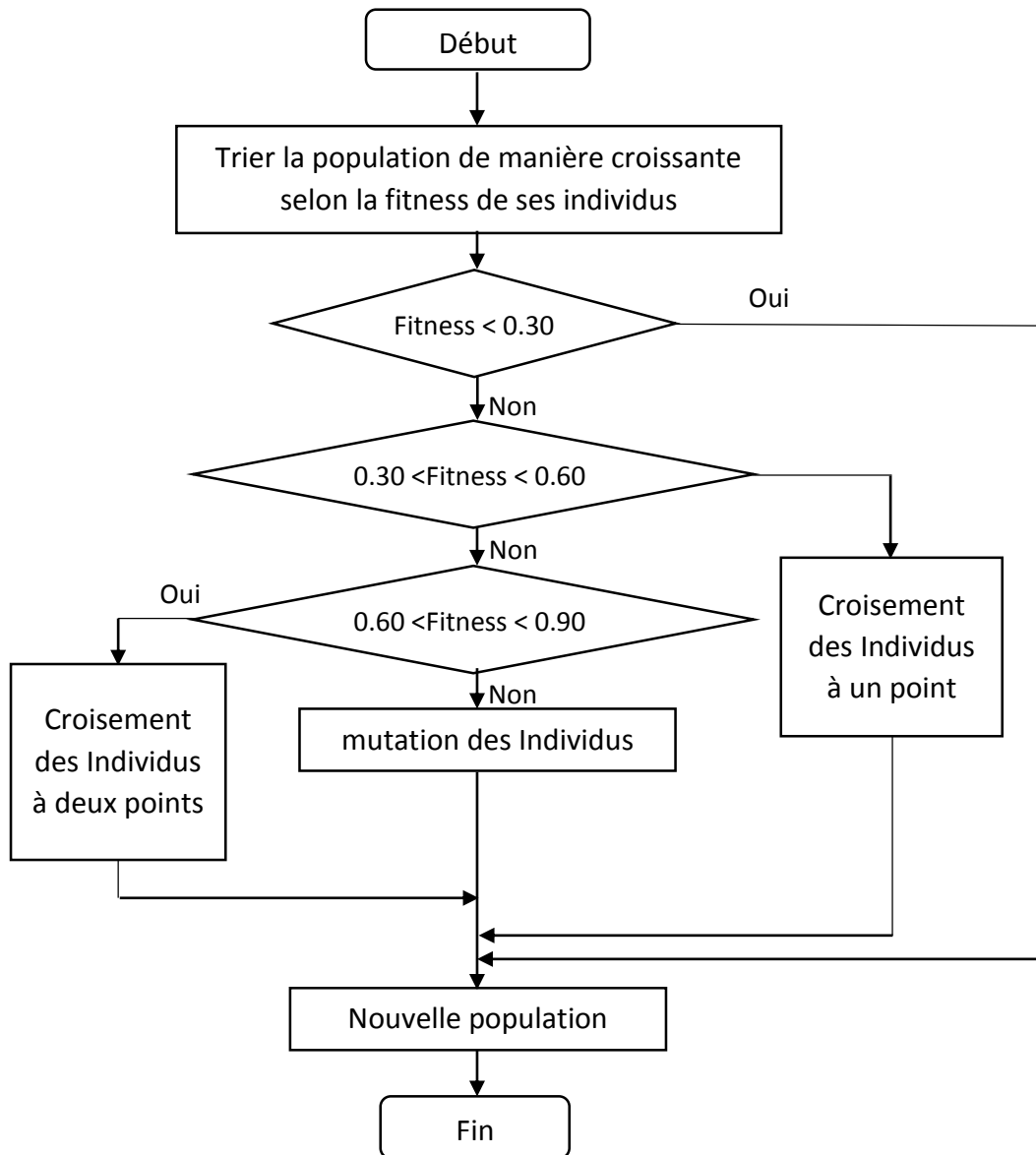


Fig. III-12 : Organigramme de la sélection par la méthode d'élite

### III.1.2.7 Mise en œuvre du système hybride neuro-génétique pour l'exemple théorique de test

L'application du réseau de neurones artificiels pour prédire les débits du jour suivant, en optimisant les poids et les biais ainsi que les paramètres de l'algorithme d'apprentissage pour 5 générations nous donne les résultats présentés dans la figure (III-13) et tableau (III-8) en comparaison avec les débits observés.

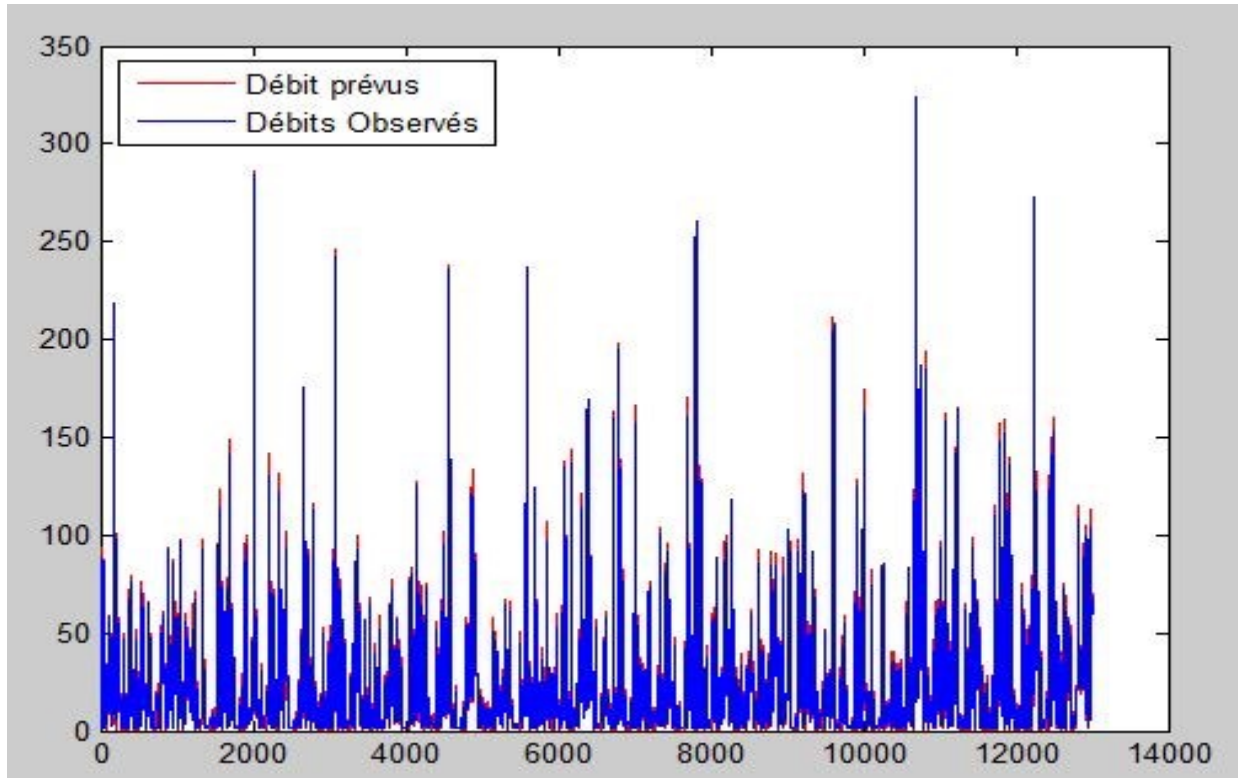
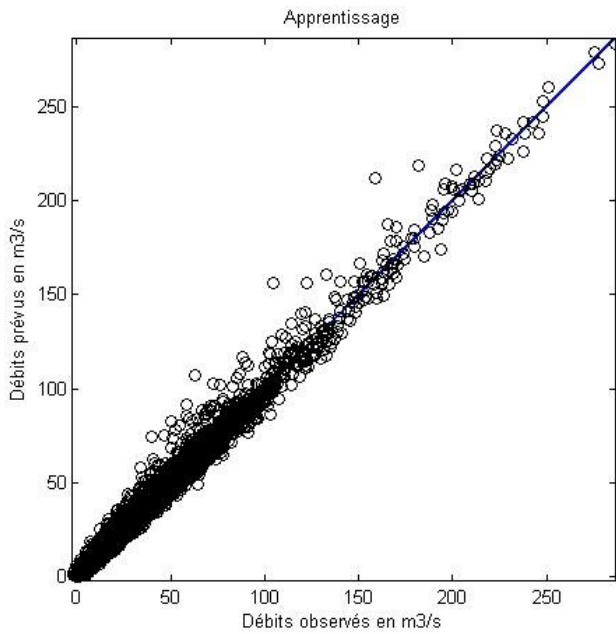


Fig. III-13 : Comparaison des débits prévus et débits observés

Tableau III-7 : les paramètres statistique des débits prévus et observés

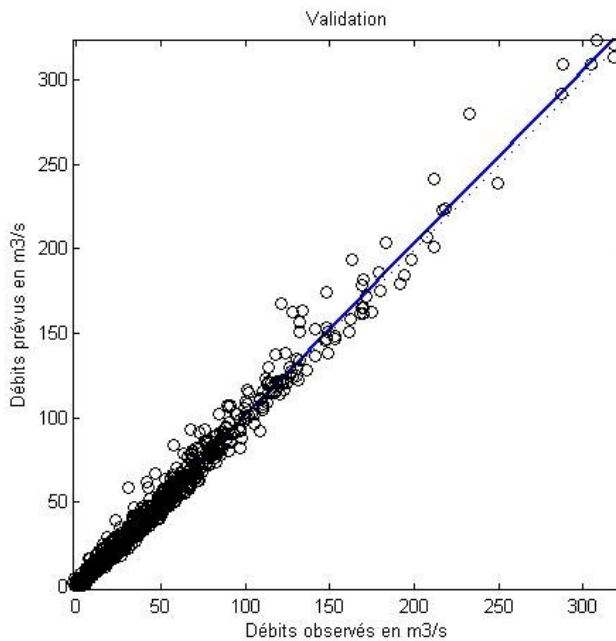
Paramètres statistiques	Débits prévus	Débits observés
<b>Minimum</b>	0.0044	0.0990
<b>Maximum</b>	318.0602	323.4867
<b>Moyen</b>	24.3967	24.6129
<b>Ecart type</b>	31.9247	32.5200
<b>Coefficient de variation</b>	1.3086	1.3213

Les résultats pendant les phases d'apprentissage, validation et test sont les suivants :



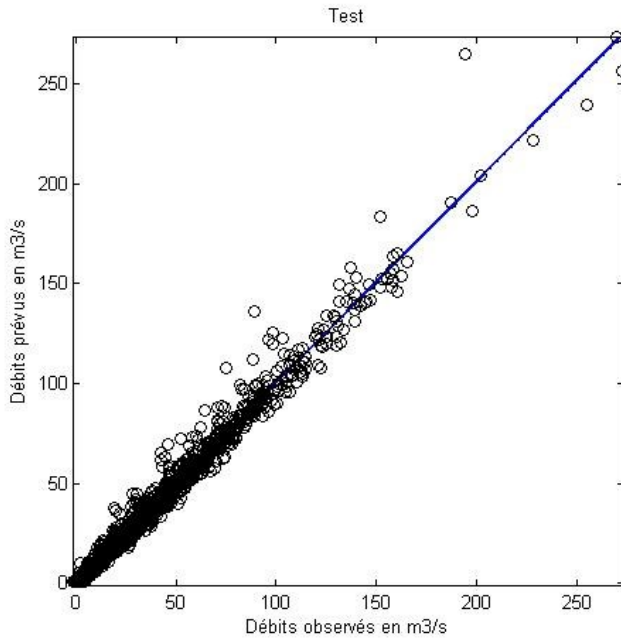
Paramètres statistiques	Débits prévus	Débits observés
<b>Minimum</b>	0.0044	0.0990
<b>Maximum</b>	286.3982	283.5000
<b>Moyen</b>	23.3844	23.1666
<b>Ecart type</b>	30.3529	30.5683
<b>Coefficient de variation</b>	1.2980	1.3195

Fig. III-14 : Comparaison entre les débits prévus par le système hybride neuro-génétique et débits observés et les paramètres statistiques phase d'apprentissage



Paramètres statistiques	Débits prévus	Débits observés
<b>Minimum</b>	0.0141	1.1000
<b>Maximum</b>	318.0602	323.4857
<b>Moyen</b>	26.2826	26.3050
<b>Ecart type</b>	37.2580	38.2580
<b>Coefficient de variation</b>	1.4176	1.4544

Fig. III-15 : Comparaison entre les débits prévus par le système hybride neuro-génétique et débits observés et les paramètres statistiques phase de validation



Paramètres statistiques	Débits prévus	Débits observés
<b>Minimum</b>	0.0099	1.1000
<b>Maximum</b>	273.1002	273.2429
<b>Moyen</b>	29.4432	29.3754
<b>Ecart type</b>	33.8481	34.3149
<b>Coefficient de variation</b>	1.1496	1.1681

Fig. III-16 : Comparaison entre les débits prévus par le système hybride neuro-génétique et débits observés et les paramètres statistiques phase de test

Tableau III-8 : Comparaison des paramètres d'évaluation entre les résultats de modélisation par RNA et AGRNA

Paramètre d'évaluation	Modélisation par RNA	Modélisation par AGRNA
<b>MSE</b>	17.0631	12.9768
<b>CE</b>	0.9838	0.9877
<b>R<sup>2</sup></b>	0.9839	0.9878

Tableau III-9 : les paramètres statistique des débits observés et prévus par RNA et AGRNA

Paramètres statistiques	Débits observés	Débits prévus par RNA	Débits prévus par AGRNA
<b>Minimum</b>	0.0990	0.0166	0.0044
<b>Maximum</b>	323.4867	330.2594	318.0602
<b>Moyen</b>	24.6129	24.5429	24.3967
<b>Ecart type</b>	32.5200	32.664	31.9247
<b>Coefficient de variation</b>	1.3213	1.3065	1.3086

### III.1.3 Discussion et interprétation

L'optimisation des poids, biais et paramètres de l'algorithme Levenberg Marquardt dans le réseau de neurones artificiels par algorithme génétique a montré que pendant 05 générations seulement une diminution nette de la valeur de MSE de 17.0631 à 12.9768 est observée.

On remarque que la valeur de la moyenne de débit simulé par RNA est supérieure à celle d'AGRNA avec une valeur de MSE grande pour RNA que AGRNA cela signifie une surestimation des valeurs dans la modélisation par RNA.

La valeur du coefficient de détermination pour la modélisation par AGRNA est un peu grand que celle de la modélisation par RNA qui augmente l'adaptation de décrire la distribution des points par l'équation de régression.

La valeur du coefficient d'efficience CE est plus intéressante à l'aide du modèle neuro-génétique que celle de réseau de neurones donc une meilleur capacité de précise les valeurs lointaines de la valeur moyenne.

Les résultats obtenus à l'aide du système hybride neuro-génétique pour l'exemple théorique de test sont très encourageants.

## III.2 Application sur des cas réels

On a choisi trois bassins versant pour tester le modèle hybride dont deux se situés dans l'Atlas Saharien et caractérisés par un climat semi-aride et un bassin situé au nord des côtier algérois.

### III.2.1 Description des bassins

#### III.2.1.1 Oued Seklafa

Oued Seklafa est situé au sud-est d'Aflou. Cet oued draine l'effluent le plus important d'oued M'zi. Il draine essentiellement les grès à dragées du Crétacé inférieur du synclinal d'Ed Dor – Kef El Gada, les grès massifs de Seklafa et les grès à dolomies de Kerakda. La délimitation du bassin drainé par oued Seklafa se fait par la détermination des lignes de partages des eaux, ces lignes sont délicates à déterminer car les formations perméables gréseuses dépassent parfois le bassin

versant. Ceci conduit à la détermination des contours présumés de l'impluvium du système.

Le débit d'oued Seklafa est contrôlé par une station hydrométrique implantés dans la région de Seklafa codée C0104 par l'ANRH.

Le bassin versant oued M'zi (Fig. III-17) fait partie des bassins endoréiques sahariens. Le réseau hydrographique de ce bassin versant est bien organisé en amont est constitué principalement par l'Oued M'Zi et l'Oued M'Saad qui prennent leurs sources au cœur du Djebel Amour et qui confluent juste à l'Exutoire d'El Fetaha, et rejoignent à l'aval l'Oued Djedi qui se déverse dans Chott El Melhrir.

La surface du bassin étant de 8895 Km<sup>2</sup>, pour un périmètre de 448 Km, l'indice de compacité de Gravellius est de 1.33, caractérisant ainsi un bassin compact de forme ramassée, par conséquent un temps de concentration de crues très court. Le sous-bassin de Seklafa, présente un l'indice de compacité de 1.23 pour une superficie de 783 Km<sup>2</sup>, et un indice de pente globale de 0.93 %, ce qui a pour effet d'accroître la rapidité de concentration des eaux à l'exutoire.

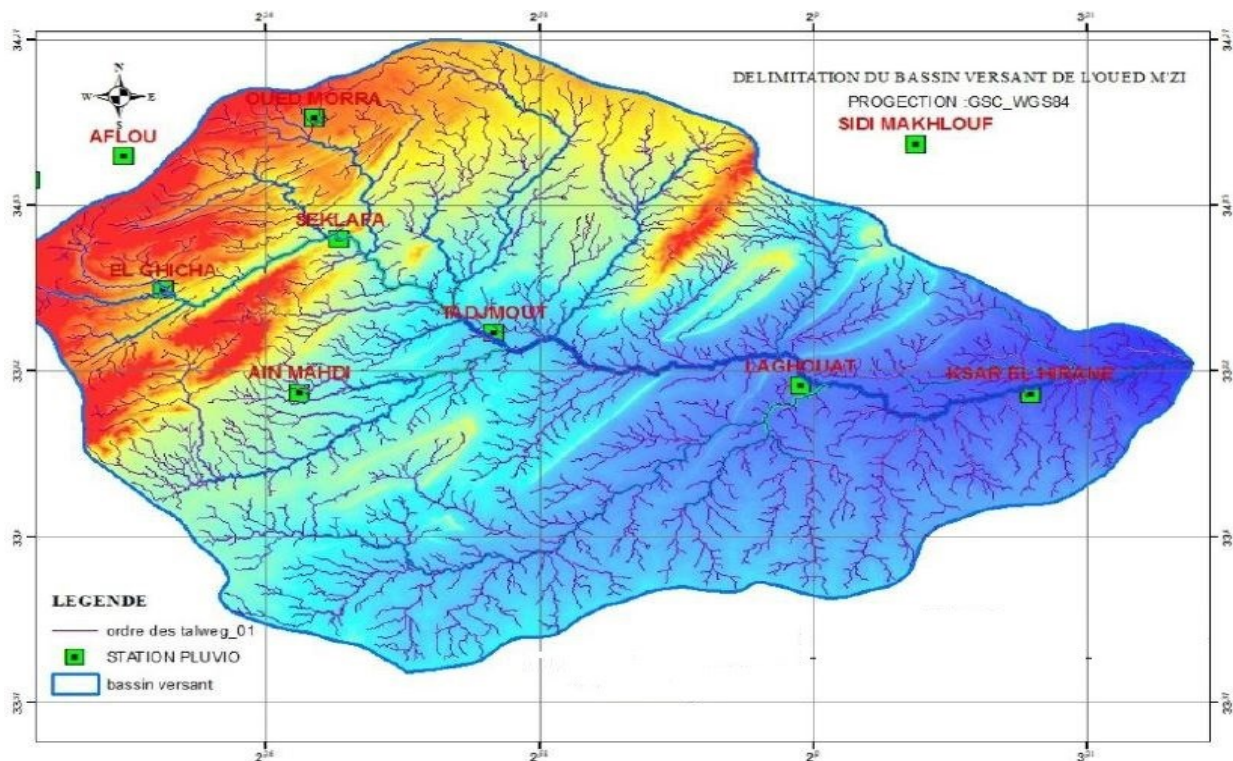


Fig. III-17 : Bassin versant oued M'zi [24]

Le bassin de l'Oued M'Zi est caractérisé par deux unités orographiques bien distinctes :

- Une première unité atlasique à reliefs escarpés où les sous-bassins présentent des altitudes moyennes supérieures à 1200 m ;
- Une deuxième unité saharienne ou de piémont saharien où l'altitude moyenne est inférieure à 950 m.

La densité de drainage pour le sous bassin de Seklafa est de  $0.75 \text{ Km}^{-1}$  et pour le bassin d'oued M'zi est de  $0.62 \text{ Km}^{-1}$ , cette valeur traduit la nature lithologique des formations géologiques et l'importance de l'intensité pluviale

Le coefficient de torrentialité de l'ensemble du bassin est de 0.13, il dépend de la pente pour les sous-bassins atlasiques, ce qui a pour effet de favoriser le ruissellement. En aval, le coefficient de torrentialité est plus faible et dépend du type d'écoulement, de la faible pente et de la typologie du réseau hydrographique ce qui a pour effet d'accroître l'infiltration au dépend du ruissellement.

#### III.2.1.2 Oued Mouilah

Cet Oued fait partie du système de Kerakda est situé dans le bassin versant de l'Oued Seggueur de Brezina appartenant au bassin versant n° 13 du Sahara. Dans l'Atlas Saharien, le bassin est représenté par les affluents à écoulement permanent constitué principalement par l'Oued Mouilah, l'Oued Rhassoul et l'Oued El Rhoul. La station R0321 draine une superficie de 3930 Km<sup>2</sup> pour un périmètre de 256 Km.

Les principales sources drainent les grès à dragées du Crétacé inférieur du synclinal de Kebeur Es Seguia, Djebel Rhoundjaïa, Dayet El Aouedj (Aïn Toualil, Aïn Marhsel, Aïn El Oudar, Aïn El Ksour), elles drainent également les grès du Kimméridgien au niveau de Djebel Alouat au Nord-Est. D'autres sources, drainent les grès de l'Oxfordien dans le secteur de Kerakda, Hassi Oumm Raneb et Rhassoul. Dans cette région, les principaux aquifères sont également connus dans les formations gréseuses fissurées du Crétacé inférieur et du Jurassique supérieur. Les conditions d'émergence sont identiques à celles du système précédent.

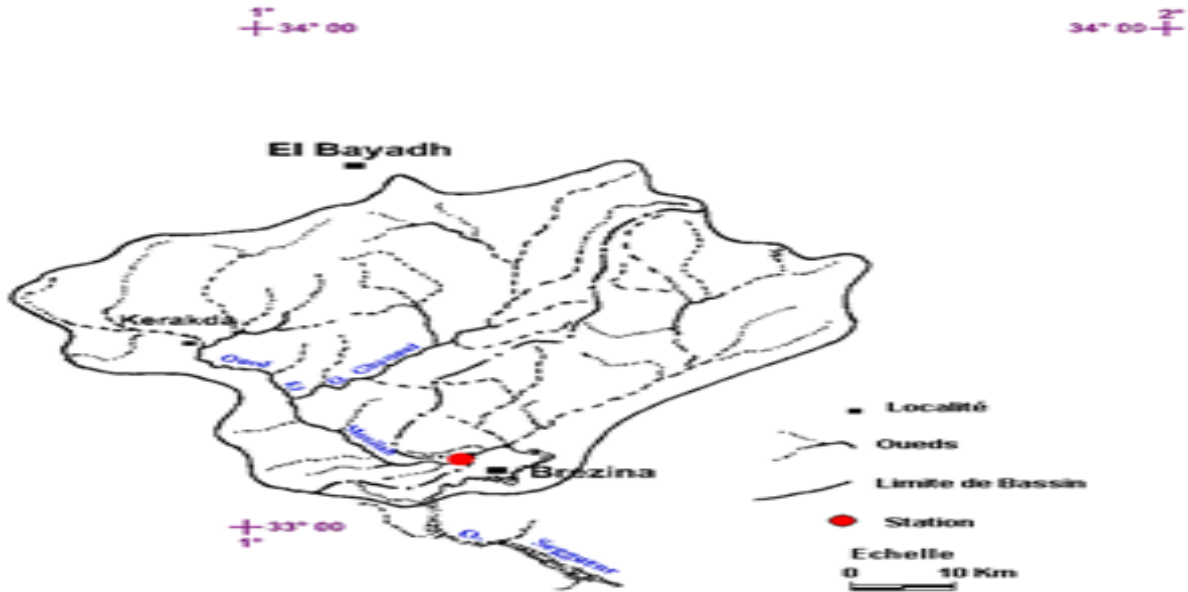


Fig. III-18 : Bassin versant de l'Oued El Mouilah.

### III.2.1.3 Oued El Hachem

Oued El Hachem draine un sous bassin versant du grand bassin côtier algérois situé dans le Nord-centre algérien et codé par l'ANRH par (02 03).

Pour ce bassin, la station pluviométrique de Sidi Amar (02 0310), offre les données pluviométriques qui fait sans lacunes de 1983 à 1989 (07 ans), la station hydrométrique de Bordj Ghobrini (02 0301) offre les données de débits (Fig. III-19).

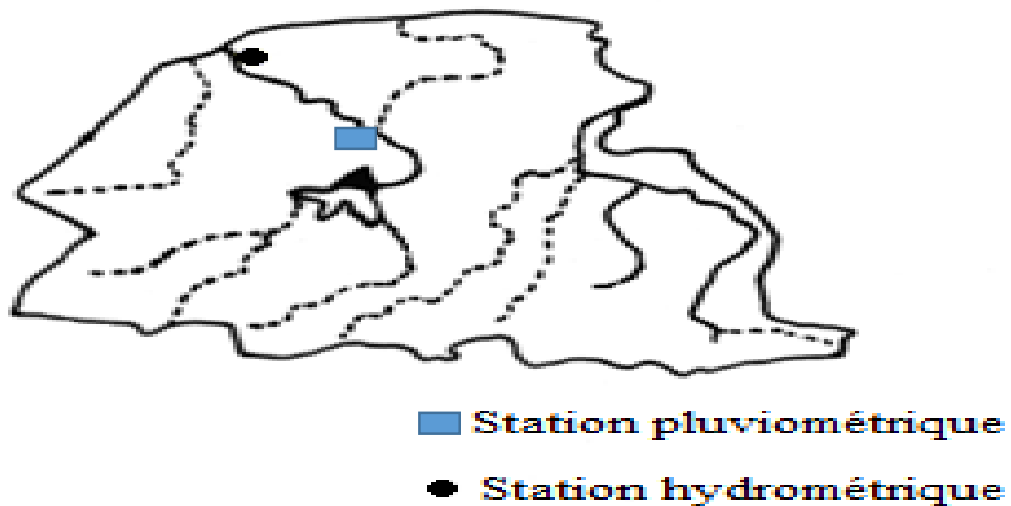


Fig. III-19 : Bassin versant de l'oued El Hachem

### III.2.2 Prévision des débits par réseau de neurones artificiels

#### III.2.2.1 Détermination du modèle pluie-débit et nombre de neurones

En admettant la même architecture du réseau de neurones artificiels, on déterminera que le nombre de neurones de la couche cachée et le modèle de pluie-débits pour chaque exemple. Les résultats obtenus sont résumés dans les tableaux suivants :

Tableau III-10 : Détermination de modèle et nombre de neurones pour les cas réels

Exemple	Modèles	Nombre des neurones	MSE
Oued Seklafa	$Q_{t+1} = RNA(P_{t-1}, P_t, Q_{t-5}, \dots, Q_{t-1}, Q_t)$	12	5.133
Oued Mouilah	$Q_{t+1} = RNA(P_{t-1}, P_t, Q_{t-6}, \dots, Q_{t-1}, Q_t)$	42	17.98
Oued El Hachem	$Q_{t+1} = RNA(P_{t-1}, P_t, Q_{t-1}, Q_t)$	81	6.375

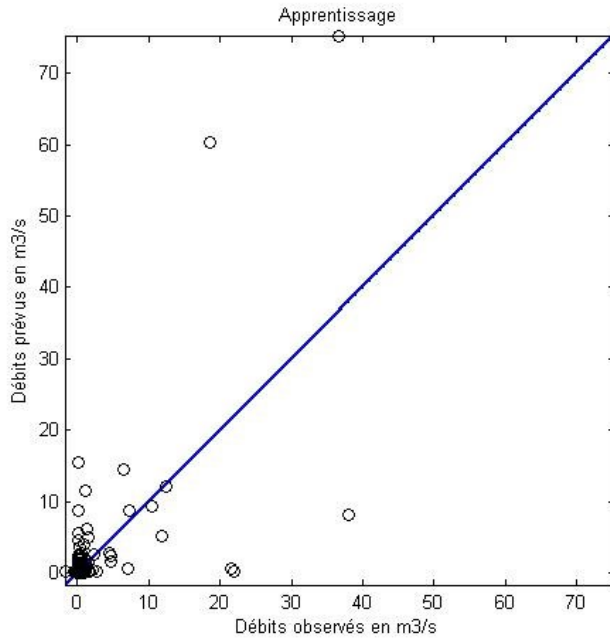
Tableau III-11 : paramètres d'évaluations de modélisation des 03 bassins par RNA

Exemple	MSE	CE	R <sup>2</sup>
Oued Seklafa	5.133	0.5990	0.6048
Oued Mouilah	17.977	0.7804	0.7820
Oued El Hachem	6.375	0.7529	0.7544

#### III.2.3 Application du système hybride neuro-génétique

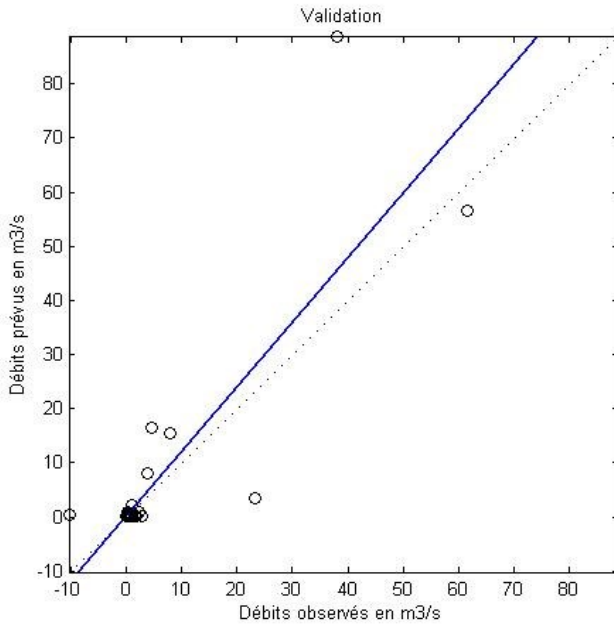
Les valeurs résultant de l'application du système hybride neuro-génétique pour prédire les débits du jour (t+1) pour les oueds pour les trois phases sont comme suit :

III.2.3.1 Oued Seklafa



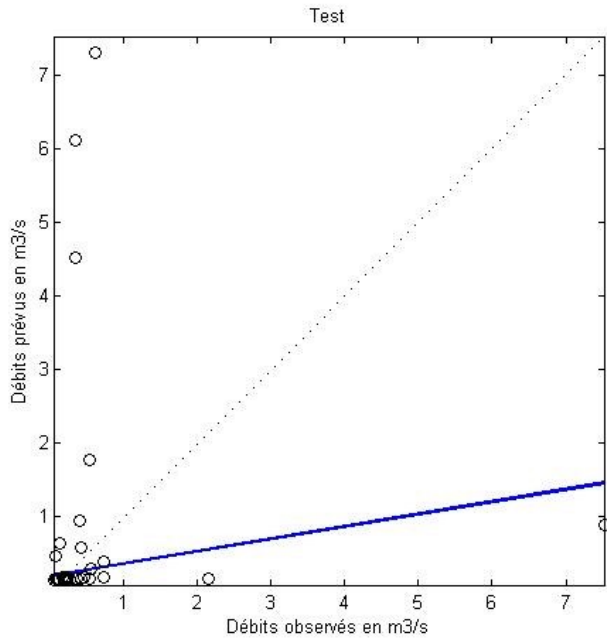
Paramètres statistiques	Débits prévus	Débits observés
<b>Minimum</b>	0.0385	0.0500
<b>Maximum</b>	38.0389	88.4700
<b>Moyen</b>	0.4012	0.4420
<b>Ecart type</b>	1.9077	3.4941
<b>Coefficient de variation</b>	5.4971	7.9048

Fig. III-20 : Comparaison entre les débits prévus par système hybride neuro-génétique et les débits observés et les paramètres statistiques phase d'apprentissage (oued Seklafa)



Paramètres statistiques	Débits prévus	Débits observés
<b>Minimum</b>	0.0656	0.0500
<b>Maximum</b>	61.5523	88.4700
<b>Moyen</b>	0.8054	0.4420
<b>Ecart type</b>	4.6509	3.4941
<b>Coefficient de variation</b>	5.7748	7.9048

Fig. III-21 : Comparaison entre les débits prévus par système hybride neuro-génétique et les débits observés et les paramètres statistiques phase de validation (oued Seklafa)



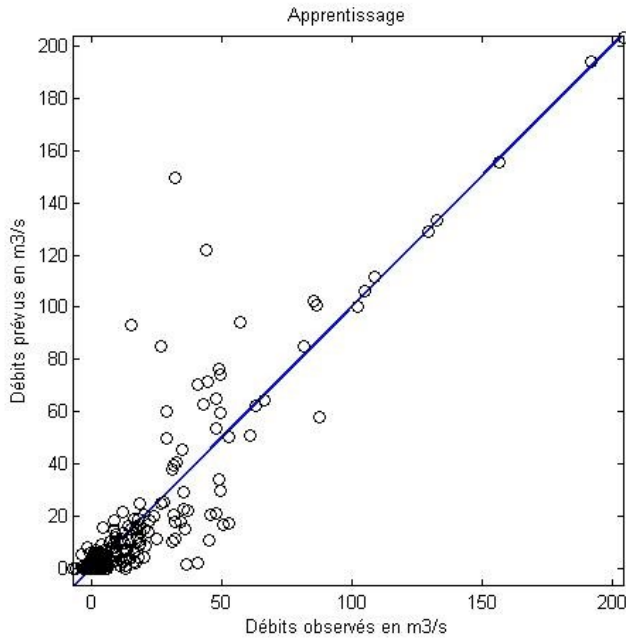
Paramètres statistiques	Débits prévus	Débits observés
<b>Minimum</b>	0.0659	0.0500
<b>Maximum</b>	7.5184	88.4700
<b>Moyen</b>	0.2525	0.4420
<b>Ecart type</b>	0.4641	3.4941
<b>Coefficient de variation</b>	1.8378	7.9048

Fig. III-22 : Comparaison entre les débits prévus par système hybride neuro-génétique et les débits observés et les paramètres statistiques phase de test (oued Seklafa)

Tableau III-12 : Comparaison des paramètres d'évaluation entre les résultats de modélisation par RNA et AGRNA (oued Seklafa)

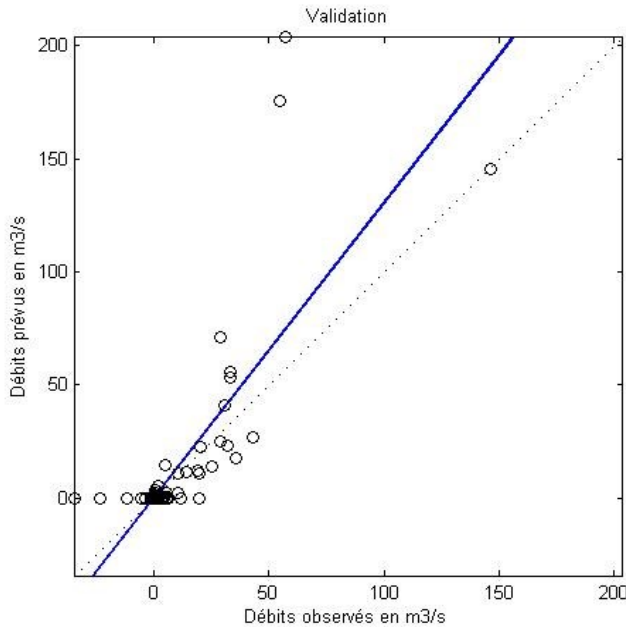
Paramètre d'évaluation	Modélisation par RNA	Modélisation par AGRNA
<b>MSE</b>	5.133	5.1199
<b>CE</b>	0.5990	0.5811
<b>R<sup>2</sup></b>	0.6048	0.5866

### III.2.3.2 L'exemple oued Mouilah



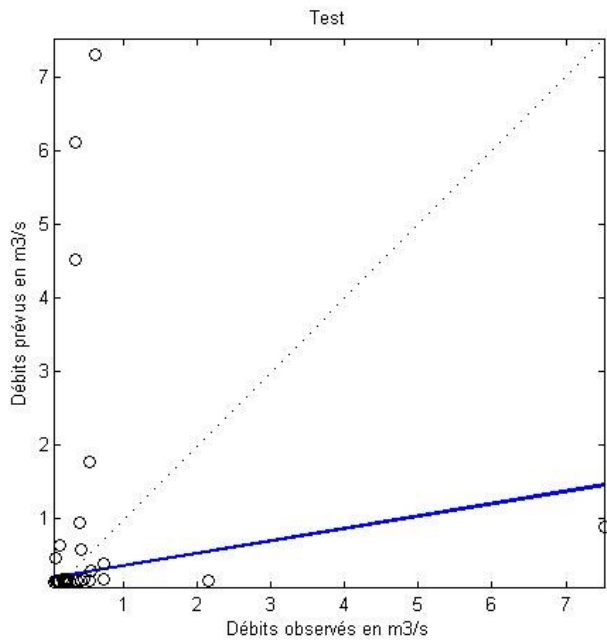
Paramètres statistiques	Débits prévus	Débits observés
<b>Minimum</b>	0.0016	0.0040
<b>Maximum</b>	203.9515	203.3000
<b>Moyen</b>	1.2822	1.1332
<b>Ecart type</b>	8.3236	9.1000
<b>Coefficient de variation</b>	6.4917	8.0304

Fig. III-23 : Comparaison entre les débits prévus par système hybride neuro-génétique et les débits observés et les paramètres statistiques phase d'apprentissage (oued Mouilah)



Paramètres statistiques	Débits prévus	Débits observés
<b>Minimum</b>	0.0103	0.0040
<b>Maximum</b>	146.4981	203.9000
<b>Moyen</b>	1.0967	1.1032
<b>Ecart type</b>	6.8034	11.1307
<b>Coefficient de variation</b>	6.2032	10.0898

Fig. III-24 : Comparaison entre les débits prévus par système hybride neuro-génétique et les débits observés et les paramètres statistiques phase de validation (oued Mouilah)



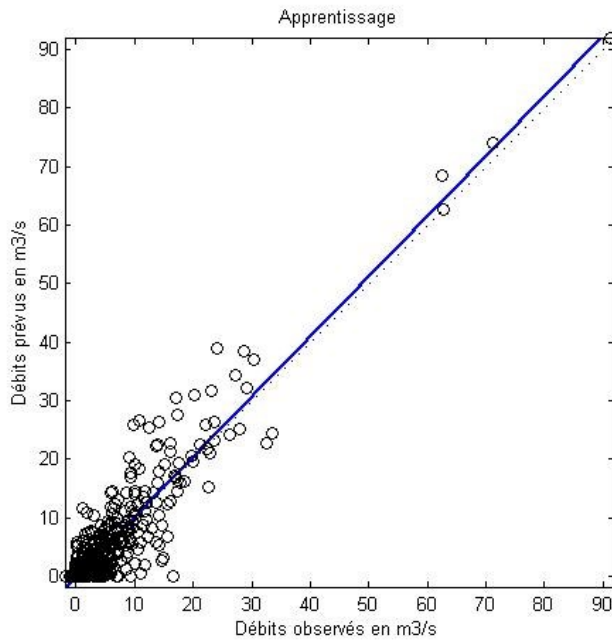
Paramètres statistiques	Débits prévus	Débits observés
<b>Minimum</b>	0.0044	0.0040
<b>Maximum</b>	97.0447	100.1000
<b>Moyen</b>	0.7789	0.5746
<b>Ecart type</b>	5.3125	5.9488
<b>Coefficient de variation</b>	6.8204	10.3525

Fig. III-25 : Comparaison entre les débits prévus par système hybride neuro-génétique et les débits observés et les paramètres statistiques phase de test (oued Mouilah)

Tableau III-13 : Comparaison des paramètres d'évaluation entre les résultats de modélisation par RNA et AGRNA (oued Mouilah)

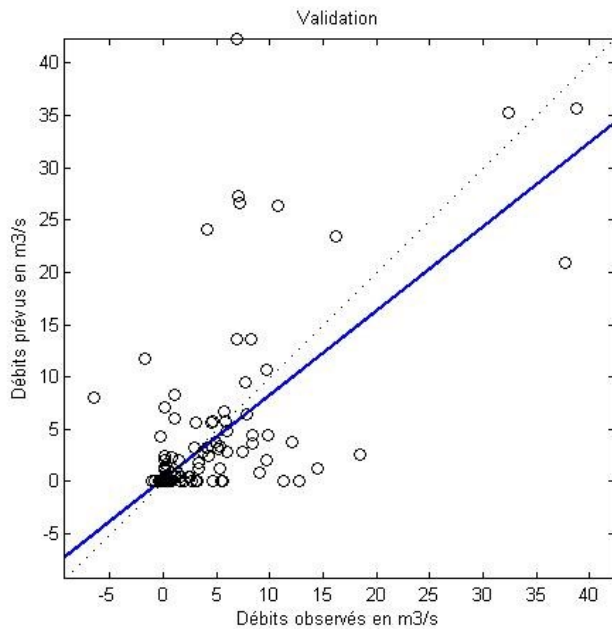
Paramètre d'évaluation	Modélisation par RNA	Modélisation par AGRNA
<b>MSE</b>	17.977	17.4967
<b>CE</b>	0.7804	0.7866
<b>R<sup>2</sup></b>	0.7820	0.7880

### III.2.3.3 L'exemple oued el Hachem



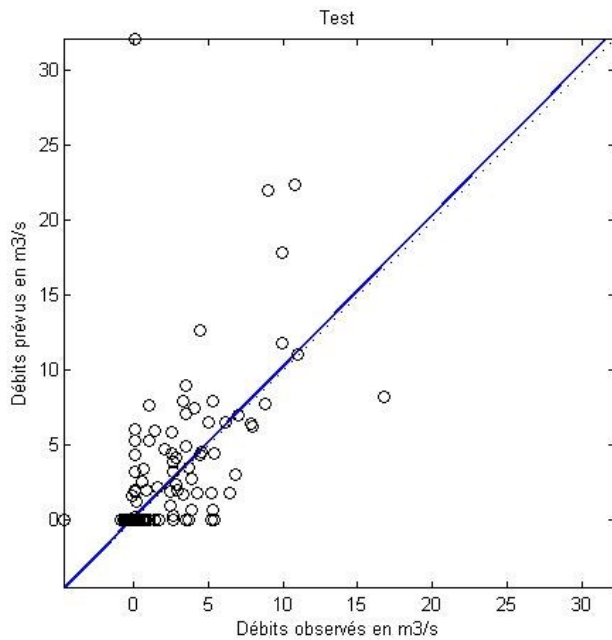
Paramètres statistiques	Débits prévus	Débits observés
<b>Minimum</b>	0.0025	0
<b>Maximum</b>	91.3045	91.9000
<b>Moyen</b>	1.5662	1.4574
<b>Ecart type</b>	5.0421	5.5019
<b>Coefficient de variation</b>	3.2193	3.7752

Fig. III-26 : Comparaison entre les débits prévus par système hybride neuro-génétique et les débits observés et les paramètres statistiques phase d'apprentissage (oued el Hachem)



Paramètres statistiques	Débits prévus	Débits observés
<b>Minimum</b>	0.0022	0.0170
<b>Maximum</b>	38.7714	16.7876
<b>Moyen</b>	1.3691	0.8498
<b>Ecart type</b>	4.0476	1.9610
<b>Coefficient de variation</b>	2.9565	2.3077

Fig. III-27 : Comparaison entre les débits prévus par système hybride neuro-génétique et les débits observés et les paramètres statistiques phase de validation (oued el Hachem)



Paramètres statistiques	Débits prévus	Débits observés
<b>Minimum</b>	0.0170	0
<b>Maximum</b>	16.7876	32.1000
<b>Moyen</b>	0.8498	0.9000
<b>Ecart type</b>	1.9610	3.0747
<b>Coefficient de variation</b>	2.3077	3.4163

Fig. III-28 : Comparaison entre les débits prévus par système hybride neuro-génétique et les débits observés et les paramètres statistiques phase de test (oued el Hachem)

Tableau III-14 : Comparaison des paramètres d'évaluation entre les résultats de modélisation par RNA et AGRNA (oued el Hachem)

Paramètre d'évaluation	Modélisation par RNA	Modélisation par AGRNA
<b>MSE</b>	6.375	5.1804
<b>CE</b>	0.7529	0.8059
<b>R<sup>2</sup></b>	0.7544	0.8602

### III.2.4 Discussion et interprétation

L'application du système hybride neuro-génétique pour la prévision des débits des trois bassins nous a permis de vérifier l'efficacité de ce système hybride pour simuler des problèmes complexes.

Les résultats obtenus sont encourageants et promoteurs:

L'exemple d'oued El Hachem présente des résultats Comme on s'y attendait une minimisation de MSE d'environ de 20% de 6.375 à 5.1804 qui applique une augmentation de 6% de la valeur de CE et R<sup>2</sup> qui signifie une bonne prédiction des

valeurs très différentes du moyen ainsi que l'augmentation de la fraction des débits prévus de 0.75 à 0.86 qui suit l'équation de corrélation.

D'autre part l'exemple d'oued Mouilah présente une petite diminution de la valeur de MSE et petite augmentation dans la valeur de CE et  $R^2$ . Ces changements prouvent les capacités du système hybride neuro-génétique à modéliser le problème.

Tandis que l'application sur l'exemple d'oued Seklafa donne des résultats différents. On a remarqué une petite minimisation dans la valeur de MSE accompagnée d'une diminution dans les valeurs de CE et  $R^2$  qui signifie que la modélisation par AGRNA a diminué la fraction des débits prévus qui peut être expliqué par corrélation ainsi qu'une mauvaise prédiction des valeurs loin du moyen.

# Conclusion générale

L'objectif de cette étude est d'élaborer un système Hybride Neuro-Génétique en commençant par la détermination d'un réseau neuronal artificiel le plus performant, puis l'optimisation de ce dernier à l'aide d'un algorithme génétique.

La détermination du réseau de neurones artificiels présente des difficultés dans la détermination du nombre de neurones des couches cachées et le choix de la topologie du réseau le plus adéquat ainsi que le choix de l'algorithme d'apprentissage.

D'un autre côté, l'algorithme génétique élaboré pour optimiser les poids, biais et même les paramètres de l'algorithme d'apprentissage du réseau de neurones artificiels nécessite une définition bien précise des opérateurs génétiques et des paramètres.

L'application de ce système neuro-génétique pour différents exemples de bassins versants dans le but de prévoir des débits journaliers nous donne généralement des résultats très encourageants.

Cependant, le système Neuro-Génétique serait moins performant pour les raisons suivantes :

- Un réseau neuronal mal dimensionné (nombre de neurones optimal, nombres des couches cachées, .... etc.) ;
- Le mauvais choix du modèle pluie-débit ;
- Le mauvais paramétrage : que ce soit pour le réseau neuronal (intervalle des poids, test d'arrêt, .... etc ) ou pour l'algorithme génétique (nombre des individus, nombre de générations, .....etc.).

En définitive, le système Neuro-Génétique permet de concilier entre deux approches et profiter des avantages des deux systèmes : réseaux de neurones artificiels (système boîte noire) et l'algorithme génétique (algorithme évolutionnaire stochastique). Malgré les résultats encourageants du système Neuro-Génétique, nous préconisons l'utilisation d'autres topologies et d'autres algorithmes d'apprentissage afin d'améliorer la prévision des débits extrêmes.



# Table des matières

INTRODUCTION .....	1
<b>CHAPITRE I : RESEAUX DE NEURONES ARTIFICIELS</b>	
<b>I.1 Introduction .....</b>	<b>3</b>
<b>I.2 Bref historique .....</b>	<b>4</b>
<b>I.3 Apprentissage numérique : .....</b>	<b>6</b>
<b>I.4 Les neurones artificiels .....</b>	<b>7</b>
I.4.1 Le modèle neurophysiologique .....	7
I.4.2 Le neurone .....	7
I.4.2.1 Dendrite .....	9
I.4.2.2 Soma .....	9
I.4.2.3 Axone .....	9
I.4.3 Le modèle mathématique .....	9
I.4.3.1 Le Neurone Formel : .....	9
I.4.3.2 Comportement .....	11
I.4.3.3 Fonction d'activation .....	12
<b>I.5 Réseau de neurones artificiels .....</b>	<b>13</b>
I.5.1 Topologie d'un réseau de neurones artificiels .....	14
I.5.2 Les réseaux de neurones non bouclés .....	14
I.5.3 Les réseaux de neurones bouclés .....	15
<b>I.6 L'apprentissage des réseaux de neurones .....</b>	<b>15</b>
I.6.1 Apprentissage supervisé .....	16
I.6.2 Apprentissage non supervisé .....	16
I.6.3 Apprentissage par renforcement .....	17
<b>I.7 Approximation parcimonieuse .....</b>	<b>18</b>
<b>I.8 Quelques modèles des réseaux de neurones : .....</b>	<b>19</b>
I.8.1 Perceptron .....	19
I.8.2 Réseau feedforward .....	20
I.8.3 Autre modèles des réseaux neuronaux : .....	21
<b>I.9 Algorithme de Levenberg-Marquardt .....</b>	<b>21</b>
I.9.1 Description de l' algorithme LM .....	22
I.9.2 Méthode de Levenberg entre Gauss-Newton et Gradient .....	23
<b>I.10 L'étape d'initialisation des poids d'un réseau neuronal .....</b>	<b>24</b>

<b>I.11 Conclusion.....</b>	<b>24</b>
-----------------------------	-----------

## CHAPITRE II : LES ALGORITHMES GENETIQUE

<b>II.1 Introduction .....</b>	<b>25</b>
<b>II.2 Bref historique .....</b>	<b>25</b>
<b>II.3 Algorithmes évolutionnaires.....</b>	<b>27</b>
<b>II.4 Principe d'un algorithme génétique .....</b>	<b>28</b>
II.4.1 Fonctionnement d'un algorithme génétique .....	30
<b>II.5 Les opérateurs génétiques .....</b>	<b>31</b>
II.5.1 Codage .....	31
II.5.1.1 Codage binaire.....	31
II.5.1.2 Codage réel.....	32
II.5.1.3 Codage sous forme d'arbre .....	32
II.5.2 Population initiale .....	32
II.5.3 Fonction d'évaluation .....	33
II.5.4 Sélection.....	33
II.5.4.1 Sélection par roulette .....	34
II.5.4.2 Sélection par classement.....	35
II.5.4.3 Sélection par tournoi .....	35
II.5.4.4 La méthode élitiste .....	35
II.5.5 Croisement ou reproduction .....	35
II.5.5.1 Croisement binaire .....	36
II.5.5.2 Croisement réel : .....	37
II.5.6 Mutation .....	37
II.5.6.1 Mutation binaire.....	37
II.5.6.2 Mutation réelle.....	38
<b>II.6 Optimisation des réseaux des neurones artificiels par algorithme génétique .....</b>	<b>38</b>
II.6.1.1 Les paramètres optimiser par algorithme génétique.....	39
II.6.1.2 Organigramme de système hybride neuro-génétique.....	39

## CHAPITRE III : PREVISION DES DEBITS PAR L'APPROCHE HYBRIDE NEURO-GENETIQUE

<b>III.1 Structure du programme et la mise en œuvre pour un exemple théorique .....</b>	<b>41</b>
III.1.1 Réseau des neurones artificiels (RNA).....	41
III.1.1.1 Les paramètres d'évaluation de performance du modèle.....	41
III.1.1.2 Structure du réseau .....	42
III.1.1.3 Fonction d'activation .....	43
III.1.1.4 Fonction de normalisation des données .....	43

III.1.1.5	Description d'exemple théorique de test.....	43
III.1.1.6	Initialisation des poids et biais .....	45
III.1.1.7	Modèle des données (Pluies, Débits) .....	46
III.1.1.8	Nombre des neurones optimal.....	47
III.1.1.9	Mise en œuvre du programme RNA pour l'exemple de test .....	49
III.1.2	Algorithme génétique (AGRNA).....	52
III.1.2.1	Fonction d'évaluation .....	52
III.1.2.2	Le nombre des paramètres à optimiser .....	52
III.1.2.3	Création de la population initiale .....	52
III.1.2.4	Le nombre des gènes.....	53
III.1.2.5	Le nombre des individus de la population .....	53
III.1.2.6	Sélection par la méthode élitisme.....	54
III.1.2.7	Mise en œuvre du système hybride neuro-génétique pour l'exemple théorique de test .....	55
III.1.3	Discussion et interprétation .....	59
<b>III.2</b>	<b>Application sur des cas réels .....</b>	<b>59</b>
III.2.1	Description des bassins.....	59
III.2.1.1	Oued Seklafa.....	59
III.2.1.2	Oued Mouilah .....	61
III.2.1.3	Oued El Hachem .....	62
III.2.2	Prévision des débits par réseau de neurones artificiels .....	63
III.2.2.1	Détermination du modèle pluie-débit et nombre de neurones .....	63
III.2.3	Application du système hybride neuro-génétique .....	63
III.2.3.1	Oued Seklafa .....	64
III.2.3.2	L'exemple oued Mouilah .....	66
III.2.3.3	L'exemple oued el Hachem .....	68
III.2.4	Discussion et interprétation .....	69
<b>CONCLUSION GENERALE .....</b>		<b>71</b>

## BIBLIOGRAPHIE

## Liste des figures

Fig. I-1 : Schéma complet d'un neurone [45] .....	8
Fig. I-2 : Schéma simplifiée d'un neurone [46].....	8
Fig. I-3 : Similitude entre le neurone biologique et artificiel [7] .....	10
Fig. I-4 : neurone a une fonction non linéaire [8].....	11
Fig. I-5 : présentation graphique des fonctions d'activations [9].....	13
Fig. I-6 : graphe d'un réseau non bouclé [4].....	14
Fig. I-7 : graphe d'un réseau bouclé (ou récurrent) [3] .....	15
Fig. I-8 : <i>Schéma d'un perceptron [8]</i> .....	19
Fig. I-9 : Schéma simplifié d'un perceptron multicouche .....	20
Fig. I-10 : Schéma simplifier d'un réseau feedforward à <b>i</b> entrées et <b>l</b> couche cachées.....	21
Fig. II-1 : Organigramme d'un algorithme génétique.....	30
Fig. II-2 : Les cinq niveaux d'organisation d'un algorithme génétique [22] .....	31
Fig. II-3 : simple arbre binaire [47] .....	32
Fig. II-4 : Schéma d'un croisement à un point.....	36
Fig. II-5 : <i>schéma d'une mutation en code binaire</i> .....	37
Fig. II-6 : organigramme du système hybride neuro-génétique.....	40
Fig. III-1 : Schéma du réseau neuronal du projet .....	42
Fig. III-2 : les pluies journalières à la station de Bejaia.....	44
Fig. III-3 : séries des débits journaliers théorique générées.....	44
Fig. III-4 : représentation graphique de l'effet du initialisation aléatoire des poids et biais .....	46
Fig. III-5 : Organigramme de détermination du nombre de neurones.....	48
Fig. III-6 : Sélection du modèle pluie-débit en fonction MSE.....	48
Fig. III-7 : comparaison débits prévus par RNA aux débits observés .....	49
Fig. III-8 : Comparaison entre les débits prévus par RNA et observés .....	50
Fig. III-9 : Comparaison les entre les débits prévus et débits observés et les paramètres statistique phase apprentissage.....	50
Fig. III-10 : Comparaison entre les débits prévus et débits observés et les paramètres statistique phase de Validation .....	51
Fig. III-11 : Comparaison entre les débits prévus et débits observés et les paramètres statistique phase de test .....	51
Fig. III-12 : Organigramme de la sélection par la méthode d'élite.....	55
Fig. III-13 : Comparaison des débits prévus et débits observés .....	56
Fig. III-14 : Comparaison entre les débits prévus par le système hybride neuro-génétique et débits observés et les paramètres statistiques phase d'apprentissage .....	57
Fig. III-15 : Comparaison entre les débits prévus par le système hybride neuro-génétique et débits observés et les paramètres statistiques phase de validation .....	57
Fig. III-16 : Comparaison entre les débits prévus par le système hybride neuro-génétique et débits observés et les paramètres statistiques phase de test .....	58
Fig. III-17 : Bassin versant oued M'zi [24].....	60
Fig. III-18 : Bassin versant de l'Oued El Mouilah. ....	62
Fig. III-19 : Bassin versant de l'oued El Hachem .....	62
Fig. III-20 : Comparaison entre les débits prévus par système hybride neuro-génétique et les débits observés et les paramètres statistiques phase d'apprentissage (oued Seklafa) .....	64
Fig. III-21 : Comparaison entre les débits prévus par système hybride neuro-génétique et les débits observés et les paramètres statistiques phase de validation (oued Seklafa) .....	64

Fig. III-22 : Comparaison entre les débits prévus par système hybride neuro-génétique et les débits observés et les paramètres statistiques phase de test (oued Seklafa) .....	65
Fig. III-23 : Comparaison entre les débits prévus par système hybride neuro-génétique et les débits observés et les paramètres statistiques phase d'apprentissage (oued Mouilah).....	66
Fig. III-24 : Comparaison entre les débits prévus par système hybride neuro-génétique et les débits observés et les paramètres statistiques phase de validation (oued Mouilah) .....	66
Fig. III-25 : Comparaison entre les débits prévus par système hybride neuro-génétique et les débits observés et les paramètres statistiques phase de test (oued Mouilah) .....	67
Fig. III-26 : Comparaison entre les débits prévus par système hybride neuro-génétique et les débits observés et les paramètres statistiques phase d'apprentissage (oued el Hachem).....	68
Fig. III-27 : Comparaison entre les débits prévus par système hybride neuro-génétique et les débits observés et les paramètres statistiques phase de validation (oued el Hachem).....	68
Fig. III-28 : Comparaison entre les débits prévus par système hybride neuro-génétique et les débits observés et les paramètres statistiques phase de test (oued el Hachem) .....	69

## Liste des tableaux

Tableau I-1 : Analogie entre un neurone formel et biologique.....	10
Tableau I-2 : formes de la fonction d'activation [9] .....	13
Tableau II-1 : Equivalence terminologique biologie génétique/algorithmes génétiques [18].....	28
Tableau III-1 : les paramètres statistiques des séries test .....	45
Tableau III-2 : comparaison entre l'initialisation aléatoire et fixée .....	46
Tableau III-3 : les modèles des pluies-débits.....	47
Tableau III-4 : Comparaison des paramètres statistiques des débits prévus par RNA et observés.....	49
Tableau III-5 : déterminer le nombre des gènes en fonction du MSE.....	53
Tableau III-6 : détermination du nombre des individus de la population en fonction de MSE .....	53
Tableau III-7 : les paramètres statistiques des débits prévus et observés .....	56
Tableau III-8 : Comparaison des paramètres d'évaluation entre les résultats de modélisation par RNA et AGRNA.....	58
Tableau III-9 : les paramètres statistiques des débits observés et prévus par RNA et AGRNA.....	58
Tableau III-10 : Détermination de modèle et nombre de neurones pour les cas réels .....	63
Tableau III-11 : paramètres d'évaluations de modélisation des 03 bassins par RNA .....	63
Tableau III-12 : Comparaison des paramètres d'évaluation entre les résultats de modélisation par RNA et AGRNA (oued Seklafa).....	65
Tableau III-13 : Comparaison des paramètres d'évaluation entre les résultats de modélisation par RNA et AGRNA (oued Mouilah).....	67
Tableau III-14 : Comparaison des paramètres d'évaluation entre les résultats de modélisation par RNA et AGRNA (oued el Hachem) .....	69

# Bibliographie

- [1] ROBERT J.ABRAHART , PAULINE E.KNEALE ,LINDA M.SEE, *Neural Networks for Hydrological Modelling*, Taylor & Francis Group plc, London, UK., 2004, p. 305.
- [2] Kevin L. priddy, Paul E. killer, *Artificial Neural Networks*, SPIE Publications; illustrated edition edition, 2005, p. 180.
- [3] C. TOUZET, *LES RESEAUX DE NEURONES ARTIFICIELS INTRODUCTION AU CONNEXIONNISME*, EC2 éd, 1992, p. 150 .
- [4] G. Dreyfus, J.-M. Martinez, M. Samuelides, M. B. Gordon, F. Badran, S. Thiria, *Apprentissage statistique troisième édition*, ÉDITIONS EYROLLES, 2008, p. 471.
- [5] M. Negnevitsky, *Artificial Intelligence A Guide to Intelligent Systems Second Edition*, Addison-Wesley, 2005, p. 435.
- [6] H. MOHAMMED, *Optimisation de la Structure des Réseaux de Neurones par Algorithmes Génétiques Mémoire de Magister*, UNIVERSITÉ DE BOUMERDÈS, 2005, p. 115.
- [7] Andrej Krenker, Janez Bešter and Andrej Kos, *Introduction to the Artificial Neural Networks*, InTech, 2011, p. 17.
- [8] M. Hajek, *NEURAL NETWORKS*, School of Computer Science , 2005, p. 114.
- [9] Z. M, *Modélisation pluie-débit à l'aide des systèmes d'inférence Neuro-flous adaptatifs de quelques bassins versants représentatifs d'Algérie. Mémoire de Magister*, Université Amar Telidji de Laghouat, 2012, p. 101.
- [10] G. Dreyfus, J.-M. Martinez, M. Samuelides, M. B. Gordon, F. Badran, S. Thiria, L Hérault, *Réseaux de neurones – Méthodologie et applications 2eme édition*, EYROLLES, 2004, p. 66.
- [11] G. DREYFUS, *LES RÉSEAUX DE NEURONES*, Laboratoire signaux, Modèles, Apprentissage statistique, 1998, p. 19.
- [12] A. Ranganathan, *The Levenberg-Marquardt Algorithm*, <http://ananth.in/Notes.html>, 2004, p. 5.
- [13] G. Laurent, *Optimisation sans contrainte de fonctions continues non linéaires*, Université de Franche-Comté, 2012, p. 27.
- [14] Riccardo Poli,William B. Langdon,Nicholas F. McPhee, John R. Koza, *A Field Guide to Genetic Programming*, Lulu Enterprises, UK Ltd, 2008, p. 250.
- [15] R. F.-G. Souquet Amédée, *ALGORITHMES GENETIQUES TE de fin d'année Tutorat de Mr Philippe Audebaud*, L'université Nice Sophia Antipolis , 2004, p. 50.

- [16] Z. ELMASKAOUI, *APPLICATION DE LA PROGRAMMATION ORIENTÉE OBJETS À L'OPTIMISATION DISCRÈTE SOUS CONTRAINTES DES STRUCTURES MÉTALLIQUES FORMÉES DE POUTRES VIA LES ALGORITHMES GÉNÉTIQUES*. Thèse doctorat, Académie Wallonie - Bruxelles, 2007, p. 153.
- [17] N. ZERARI, *LES ALGORITHMES GENETIQUES EN MAINTENANCE mémoire de Magister*, UNIVERSITE EL HADJ LAKHDAR BATNA, 2006, p. 97.
- [18] M. Melanie, *An Introduction to Genetic Algorithms*, A Bradford Book; Reprint edition, 1999, p. 221.
- [19] Rahul Malhotra, Narinder Singh & Yaduvir Singh, *Genetic Algorithms: Concepts, Design for Optimization of Process Controllers*, vol. 4, March 2011.
- [20] E. A. E. F. ABDELHAFID, *Master Sciences et Techniques Eau et Environnement*, 2012.
- [21] S. Ramkumar, *Algorithmes genetiques*, Université Libre de Bruxelles, 2007, p. 23.
- [22] P. Krause, D. P. Boyle, and F. Base, *Comparison of different efficiency criteria for hydrological model assessment*, vol. 5, *Advances in Geosciences*, 2005, p. 89–97.
- [23] A. Chenafi, *Carthographie et protection qualitative des eaux siuterraines en zone aride, cas de Milok-Metlili, Atlas saharien central (Laghouat-Algerie) Memoire de magister*, Université d'oran, 2013, p. 196.
- [24] Pierre Borne, Mohamed Benrejeb, Joseph Haggège, *Les réseaux de neurones Présentation et application*, TECHNIP, 2007.
- [25] Elena C.Carcano<sup>1</sup>, Paolo Bartolini<sup>2</sup>, Marco Muselli<sup>3</sup>, 1I.M.A.G.E. PADOVA, 2D.I.A.M. GENOVA; 3C.N.R.-I.E.I.I.T. GENOVA, *Recurrent Neural Networks in Rainfall –Runoff modeling at daily scale*.
- [26] M. G. i. Pons, *Algorithmes d'optimisation Cours de MATLAB*, université de Lausanne, 2009-2010, p. 14.
- [27] D. A. Coley, *An Introduction to Genetic Algorithms for Scientists and Engineers*, World Scientific, 1999, p. 211.
- [28] M. P. RAJURKAR, U. C. KOTHYARI, U. C. CHAUBE, *Artificial neural networks for daily rainfall-runoff modelling*, vol. 47, *Hydrologkal Sciences-Jo umai-des Sciences Hydrologiques*, 2002, pp. 865-877.
- [29] Kuok King Kuok and Nabil Bessaih<sup>1</sup>, *ARTIFICIAL NEURAL NETWORKS (ANNS) FOR DAILY RAINFALL RUNOFF MODELLING*, vol. 68, *Journal - The Institution of Engineers, Malaysia*, September 2007, pp. 31-41.
- [30] A. Bezzini, *Commande Prédictive Non Linéaire en Utilisant Les Systèmes Neuro-Flous et les Algorithmes Génétiques Mémoire de Magister*, Université Mohamed Khider – Biskra, 2013, p. 109.
- [31] Dr. Lubica Benuskova, Professor Nikola Kasabov, *Computational Neurogenetic Modeling 1st Edition*, Springer, 2007, p. 309.

- [32] A. Sedki a,\* , D. Ouazar a, E. El Mazoudi b, *Evolving neural network using real coded genetic algorithm for daily rainfall–runoff forecasting*, vol. 36, Elsevier, 2009, p. 4523–4527.
- [33] Andrew Chipperfield, Peter Fleming, Hartmut Pohlheim, Carlos Fonseca, *Genetic Algorithm TOOLBOX For Use with MATLAB*, University of Cheffield, p. 105.
- [34] R. Rojas, *Neural Networks - A Systematic Introduction*, Springer-Verlag, 1996, p. 502.
- [35] S. Hakin, *Neural Networks A comprehensive Foundations 2 edition*, Perason education, 1999, p. 823.
- [36] F. GIRARD, *OPTIMISATION DE STRATIFIEES EN UTILISANT UN ALGORITHME GEENEETIQUE de memoire maitrise*, Université LAVAL Québec, 2006, p. 182.
- [37] C. T. Leondes, *optimization Techniques 1 edition*, Academic Press;, 1997, p. 423.
- [38] S. E. H. Randy L. Haupt, *PRACTICAL GENETIC ALGORITHMS*, Wiley-Interscience, 2004, p. 272.
- [39] Jayawardena, A. W., N. Muttill and T.M.K.G. Fernando, *Rainfall-Runoff Modelling Using Genetic Programming*, Modelling and Simulation Society of Australia and New Zealand, pp. 1841-1847.
- [40] WU2, Y. J. CAO1 and Q. H., *TEACHING GENETIC ALGORITHM USING MATLAB*, vol. 36, Int. J. Elect. Enging. Educ Great Britain, 1999, p. 139–153.
- [41] D. W. Marquardt, *An Algorithm for Least-Squares Estimation of Nonlinear Parameters*, vol. 11, Journal of the society for Industrial and Applied Mathematics, 1963, pp. 431-441.
- [42] J. Carr, *An Introduction to Genetic Algorithms*, Université de Caen Basse-Normandie, 2014, p. 40.
- [43] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolutions Programs 3rd edition*, Soringer, 1998, p. 388.

### **site internet**

- [44] "<http://fr.wikipedia.org/wiki/Neurone>," [Online].
- [45] "<http://sante-medecine.commentcamarche.net/faq/13882-neurone-definition>," [Online].
- [46] "[http://fr.wikipedia.org/wiki/Arbre\\_binaire](http://fr.wikipedia.org/wiki/Arbre_binaire)," [Online].