UNIVERSITE KASDI MERBAH OUARGLA

Faculté des Sciences Appliquées

Département de Génie des procédés

Mémoire

MASTER ACADEMIQUE

Filière : Génie des procédés

Option : Génie chimique

Présenté par :

Nadjat HAKKOUM

Amira HAOUED MOUISSA

Thème :

Modélisation du comportement dynamique d'un système chimique par les réseaux de neurones

Soutenu le 04/06/2017

Devant le jury composé de :

Djamel MENNOUCHE	M.C.B	Président
Souad ZIGHMI	M.A.A	Examinatrice
Mokhtar Lati	M.A.A	Examinateur
DOUNIT Salah	Pr	Encadreur

Année Universitaire : 2016-2017

Dédicaces

A nos parents,

A nos frères et nos sœurs,

A nos familles,

Et a tous nos amis.

Nadjat et Amira.

Remerciement

Avant tout, merci pour notre dieu ALLAH qui nous a protégé et permis d'atteindre des succès dans notre vie.

Nous remercierons tout d'abord le Pr S.DOUNIT pour avoir accepté d'être notre encadreur de mémoire, pour ses critiques, ses conseils très précis sur nos travaux de recherche et pour avoir relu, corrigé et commenté très soigneusement ce travail. Nous espérons qu'il trouve dans cette mémoire l'expression de notre profond respect.

Nous remerciement les membres de jury qui ont eu la disponibilité et la bonne volonté pour examiner notre travail ainsi que pour évaluer sa valeur scientifique.

Nos plus vifs remerciements vont également à Melle S.ZIGHMI et Mr M.CHAOUKI qui nous a accordé un peu de son temps pour discuter autour de sujets diverses..

Nos remerciements vont également à Mr N.METIAZ, Mr Med.A.ZAATOUT et Melle Z.ZIDANI pour leurs aides efficaces.

Sommaire

Dédicaces I
Remerciement..... II
Liste des figures III
Liste d'abréviation.....VI
Introduction générale.....1
Chapitre I : Eléments de bibliographie.....3
Historique.....4
I. Les réseaux neuromimétique 5
I-1. Introduction 5
I-1-1. Définition 5
I-1-2. Le neurone formel 6
I-2. Les fonctionnements de RNA 6
I-2-1. Principe des réseaux de neurones artificiels 6
I-2-2. Les différents types de réseaux 8
I-2-3. Les réseaux de neurones non bouclés 9
I-2-4. Les réseaux de neurones bouclés. 10
I-3. Réseaux de neurones pour les systèmes dynamiques 10
I-3-1. Construction d'un modèle.....11
I-3-2. Modélisation de procédé 13
I-3-3. Identification par réseaux de neurones 14
I-3-4. Les avantages des réseaux de neurones 15
I-3-5. Les inconvénients des réseaux de neurones.....15
Chapitre II : Partie pratique et modélisation 16
II-1. Introduction.....17
II-2.Base et algorithme d'apprentissage 17
II-3.L'algorithme de rétropropagation. 18
II.3-1.Cas de la couche de sortie.....18
II.3-2.Cas d'une couche cachée 21
II.4.Présentation du système modélisé.....27
II.4-1. La première méthode 27

II.4-2. La deuxième méthode : 30
II-5.L'utilisation de logiciel Fortran 32
II-6. Synthèse du modèle neuromémitique du réacteur continu parfaitement agité .. 32
II-7.Présentation d'une simulation type 33
II-8. Analyse par le modèle des réseaux de neurones 34
II-8.1. Evolution de l'erreur d'apprentissage 34
II-8.2.Phase de test 34
II-8.3. Effet du nombre de neurone dans la couche cachée 35
II.8-4. Effet du nombre d'itérations 39
Conclusion générale 41
Références bibliographique 42

Liste des figures

Figure I.1 : Schéma simplifié d'un neurone de cerveau humain.....7
Figure I.2 : Le fonctionnement du neurone artificiel.....9
Figure I.3.Différents fonction de transfert: (a) du neurone « seuil » : (b) du neurone « Linéaire » et (c) du neurone « sigmoïde ». 9
Figure I.4 : les architectures des réseaux de neurone artificiel les plus populaires. 10
Figure I.5 : Architecture d'un réseau de neurone non bouclé. 10
Figure I.6Architecture d'un réseau de neurone bouclé. 11
Figure I.7 Identification d'un système dynamique non linéaire par le réseau de neurone. 15
Figure II.1Organigramme représentant l'algorithme de réseau de neurone du modèle.27
Figure II.2: Schéma de réacteur continu parfaitement agité de procédé. 28
Figure II.3 : Les courbes des équations continu et discrète du C_{AS}.....32
Figure II.4Architecture de réseau du modèle. 33
Figure II.5 Évolution de l'erreur quadratique moyenne d'apprentissage. 34
Figure II.6 Evolution de la sortie du procédé et celle du réseau. 35
Figure II.7 Résultat de simulation du 1ere Essais 36
Figure II.8 Résultat de simulation du 2ème Essais. 37
Figure II.9 Résultat de simulation du 3ème Essais. 37
Figure II.10 Résultat de simulation du 4ème Essais. 38
Figure II.11 Résultat de simulation pour 1000 itération. 39
Figure II.12 Résultat de simulation pour 7000 itération. 40

Liste d'abréviation

RNA : Réseaux de neurone artificiel.

PMC : Perceptron multicouche.

SISO : Signal input-signal output.

EQM : l'erreur quadratique moyenne.

$\mathbf{C_{A0}}$: La concentration de l'entrée de réacteur en régime transitoire.

$\mathbf{C_{AS}}$: La concentration de sortie de réacteur en régime transitoire.

$\widetilde{\mathbf{C_{A0}}}$: La concentration de l'entrée de réacteur en régime permanent.

$\widetilde{\mathbf{C_{AS}}}$: La concentration de sortie de réacteur en régime permanent.

$\mathbf{C'_{A0}}$ et $\mathbf{C'_{AS}}$: Les variables d'écart.

H(s) : Opérateur dit « zero order hold »

$\Delta\mathbf{t}$: le pas de temps considéré lors de la discrétisation.

Z : opérateur de retard d'un temps Δt

G(s) : la fonction de transfert analogique

V : Le volume de réacteur.

Q : le début volumique de réacteur.

$\boldsymbol{\tau}$: le taux de séjour de réacteur.

k : la constante de vitesse.

Introduction générale

Le grand développement de la science durant ces dernières décennies a permis de créer de nouvelles techniques pour résoudre des problèmes très complexes qui n'ont

pas trouvé de solutions en utilisant les méthodes traditionnelles. Parmi ces techniques on trouve les réseaux neuromimétique qui sont des mécanismes simulant le fonctionnement du cerveau humain c'est-à-dire plusieurs unités élémentaires de calcul interconnectées entre elles et appelées neurones artificiels.

Le neurone artificiel est capable d'apprendre automatiquement à partir d'un certain nombre d'exemples puis de généraliser d'une façon empirique sans requérir une description explicite de la connaissance. Ces réseaux partagent avec le cerveau des propriétés importantes une répartition distribuée de l'information sur l'ensemble du réseau, une programmation non explicite, etc.

A cause de la capacité de ces réseaux à traiter de l'information, ils trouvent de nombreuses applications en ingénierie comprenant la modélisation de processus dynamiques non linéaires (prédictions financières), la reconnaissance de formes (lecture des codes postaux), dans le domaine de la recherche en environnement, ils sont notamment utilisés en climatologie, dans les différentes opérations unitaires (distillation, extraction liquide -liquide, membrane liquide émulsionnée…). [1]

Les modèles paramétriques permettent de déterminer la réponse dynamique d'un processus quelle que soit la nature des signaux d'excitation. Ces modèles sont caractérisés par un nombre fini de paramètre qui interviennent dans la structure du modèle. On peut diviser ce type de modèle en : d'une part modèle de connaissance et d'autre part le modèle boite noire (réseau de neurones). Par contre les modèles non paramétriques ne dépendent pas de la structure du système. Ils sont donnés par des réponses temporelles ou fréquentielles du système.

Il est important de noter que le système existe indépendamment de tout modèle que l'on peut lui attribuer et que le modèle n'est souvent qu'une simplification de la réalité.[2]

La première partie de ce mémoire est consacrée à la présentation d'un aperçu général sur les réseaux de neurones c'est-à-dire leur architecture, leur fonctionnement, leur utilisation, et leur type…etc. Ensuite nous effectuerons une identification de notre système.

L'objectif de notre étude est de modéliser un système dynamique décrit par une équation différentielle par les réseaux de neurones. Nous avons choisi de modéliser le comportement dynamique d'un réacteur continu parfaitement agité dans lequel une réaction de premier ordre se déroule. On s'est proposé une architecture du réseau, et nous vérifions la capacité de ce dernier à modéliser un procédé chimique. Afin de construire notre base de données pour apprentissage et pour test, on procède à l'établissement de l'équation de bilan de matière sur le réactif A et ensuite, on la discrétise pour pouvoir générer des valeurs ponctuelles. Pour cela, nous utiliserons deux méthodes distinctes basées respectivement sur la transformée de Laplace (solution continue exacte du bilan de masse) et sur la transformée en Z (discrétisation de l'équation différentielle continue). Donc la partie pratique concerne l'apprentissage du réseau et à son test à partir d'un autre ensemble n'ayant pas servi à l'apprentissage.

Chapitre I : Eléments de bibliographie

Historique [3]

Les recherches menées dans le domaine du connexionnisme ont démarré avec la présentation en 1940 par W. MCCulloch et W. Pitts d'un modèle simplifié de neurone biologique communément appelé neurone formel. Ils montrèrent également théoriquement que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes.

En 1949, D.Hebb initie, dans son ouvrage "The Organization of Behavior", la notion d'apprentissage. Deux neurones entrant en activité simultanément vont être associés (c'est-à-dire que leurs contacts synaptiques vont être renforcés). On parle de loi de Hebb et d'associationnisme. En 1958, F. Rosenblatt développe le modèle du Perceptron. C'est un réseau de neurones inspiré du système visuel. Il possède deux couches de neurones : une couche de perception (sert à recueillir les entrées) et une couche de décision. C'est le premier modèle pour lequel un processus d'apprentissage a pu être défini.

S'inspirant du perceptron, Widrow et Hoff, développent, dans la même période, le modèle de l'Adaline (Adaptive LinearElement). Ce dernier sera, par la suite, le modèle de base des réseaux de neurones multi-couches. En 1969, Les recherches sur les réseaux de neurones ont été pratiquement abandonnées lorsque M. Minsky et S. Papert ont publié leur livre « Perceptrons » (1969) et démontré les limites théoriques du perceptron, en particulier, l'impossibilité de traiter les problèmes non linéaires par ce modèle. En 1982, Hopfield développe un modèle qui utilise des réseaux totalement connectés basés sur la règle de Hebb pour définir les notions d'attracteurs et de mémoire associative. En 1984 c'est la découverte des cartes de Kohonen avec un algorithme non supervisé basé sur l'auto- organisation et suivi une année plus tard par la machine de Boltzman (1985).

Une révolution survient alors dans le domaine des réseaux de neurones artificiels : une nouvelle génération de réseaux de neurones, capables de traiter avec succès des phénomènes non- linéaires : le perceptron multicouche (PMC) ne possède pas les défauts mis en évidence par Minsky. Proposé pour la première fois par réaliser à partir Werbos, le Perceptron Multi-Couche apparaît en 1986 introduit par Rumelhart et simultanément, sous une appellation voisine, chez Le Cun (1985). Ces systèmes

reposent sur la rétropropagation du gradient de l'erreur dans des systèmes à plusieurs couches, chacune de type Adaline de Bernard Widrow, proche du Perceptron de Rumelhart.

I. Les réseaux neuromimétique

I-1. Introduction

Les réseaux de neurones connaissent depuis quelques années un succès croissant dans divers domaines des sciences de l'ingénieur ; celui du génie des procédés n'échappe pas à cette règle. Malheureusement, la littérature fourmille d'exemples où la mise en œuvre des réseaux de neurones relève plus de la recette que d'une approche raisonnée. De plus, les connotations biologiques du terme « réseaux de neurones », et l'utilisation du terme d'apprentissage, ont introduit une grande confusion ; elles ont conduit à relier abusivement les réseaux de neurones à l'intelligence artificielle. Or l'intérêt des réseaux de neurones, dans le domaine des sciences de l'ingénieur, ne doit rien à la métaphore biologique : il est uniquement dû aux propriétés mathématiques spécifiques de ces réseaux.

Les réseaux de neurones ont prouvé leur efficacité dans plusieurs domaines : tels que le génie chimique, le traitement du signal, l'identification paramétrique, la commande des procédés non linéaires, l'estimation et la détection des défauts.

I-1-1. Définition

Un réseau neuromimétique est un ensemble d'unités de calcul connectées entre elles. Son utilisation dans le contrôle des procédés est tirée du fonctionnement du cerveau humain. En effet, le cerveau humain est constitué d'un grand nombre de cellules connectées entre elles et lorsqu'un signal provenant d'un neurone excite une cellule à travers des canaux dits ''dendrites'', celle-ci transmet le message à la cellule voisine. Cette transmission se fait par le biais d'un canal appelé ''axone'' et la jonction à travers laquelle le signal passe d'un neurone à un autre est appelé synapse. La figureI. 1 représentée un schéma simplifié d'un neurone de cerveau humain.[4]

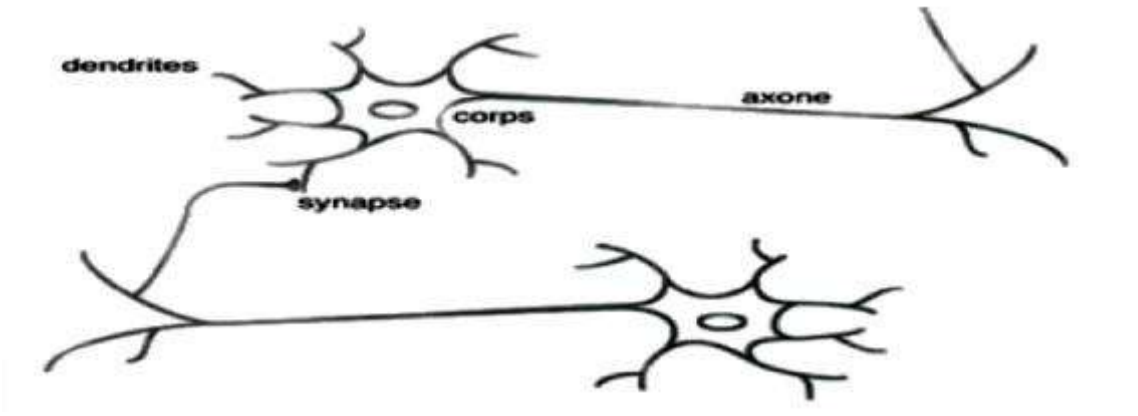


Figure I.1 : Schéma simplifié d'un neurone de cerveau humain

I-1-2. Le neurone formel [5]

Un neurone formel (artificiel) est une représentation mathématique et informatique d'un neurone biologique. Il possède généralement plusieurs entrées et une sortie qui correspondent respectivement aux dendrites du neurone biologique. Un neurone formel calcule la somme pondérée des entrées reçues, puis applique à cette valeur une fonction d'activation. La valeur finale obtenue est la sortie du neurone.

I-2. Les fonctionnements de RNA

I-2-1. Principe des réseaux de neurones artificiels

Chaque neurone peut être vu comme un automate se composant d'entrées, d'une sortie et d'une fonction de transfert dans le réseau, les neurones sont connectés entre eux afin de permettre l'échange d'information via ces connexions. Les neurones sont regroupés en couches. Chacune d'elles constitue un sous–groupe qui réalise une opération indépendante des opérations effectuées par les autres sous-groupes. Habituellement chaque neurone est connecté à tous neurones dans la couche précédente et la couche suivante, excepté dans les couches d'entrée et de sortie du réseau.

L'information donnée à un réseau de neurone est propagée couche par couche de l'entrée, à la sortie. Le réseau est construit grâce à un algorithme appelé algorithme d'apprentissage qui a pour but de calculer les différents paramètres. (Les poids associés aux différentes connexions).

Chaque neurone autre que ceux de la couche d'entrée se compose de deux éléments

- La jonction de somme qui additionne les signaux d'entrée après les avoir

Multipliés par la matrice des poids Wi.

- La fonction d'activation qui reçoit deux signaux : la sortie de la jonction de

Somme et la valeur ″ biaisée″.

Du point de vue mathématique, l'entrée du neurone i s'écrit :

$$e_i = \sum_{j=1}^{n} \mathrm{Wij} \,.\, \mathrm{Xi} \qquad \textbf{(I.1)}$$

Où

e_i : Entrée globale du neurone i

xi : Signaux d'entrée du neurone i

wij: Poids associé au signal reliant le neurone i au neurone j de la couche précédente.

Et sa sortie après transformation s'écrit :

$$y_i = f(e_i - \theta) \qquad \textbf{(I.2)}$$

θ : Seuil ou niveau d'activation du neurone.

f : Fonction d'activation (fonction de transfert).

La figure I.2 représente le fonctionnement du neurone artificiel.

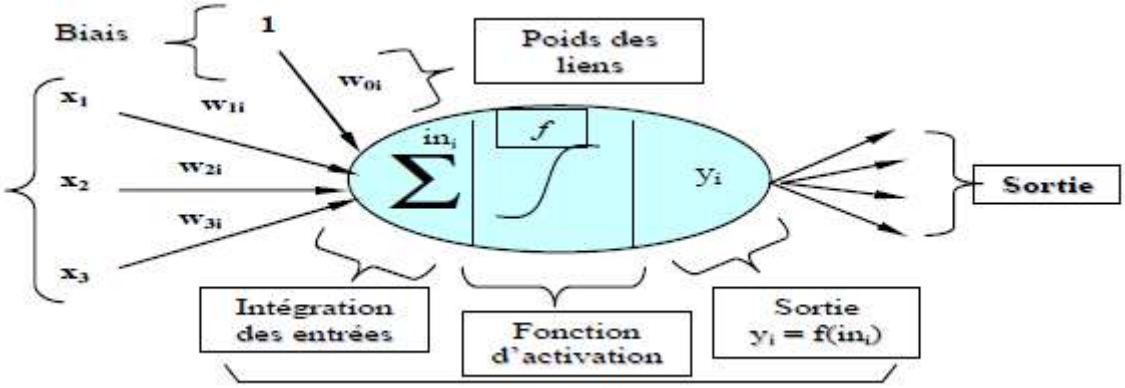


Figure I.2 : Le fonctionnement du neurone artificiel.

Le choix d'une fonction d'activation se révèle être un élément constitutif important des réseaux de neurones. Il existe plusieurs types de fonctions d'activation, mais les plus utilisées sont représentées par la figure I.3

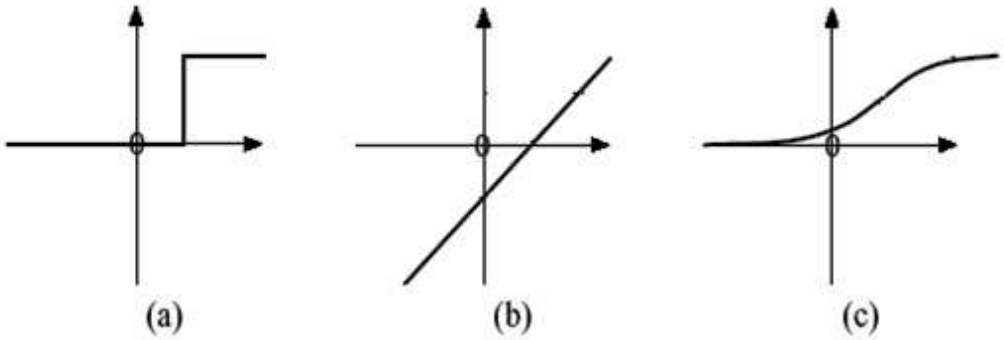


Figure I.3. Différents fonction de transfert: (a) du neurone « seuil » : (b) du neurone « Linéaire » et (c) du neurone « sigmoïde ».

I-2-2. Les différents types de réseaux [6]

On peut classer les RNA en deux grandes catégories:

- **Les réseaux "FEED-FORWARD"**

Appelés aussi "réseaux de type Perceptron", ce sont des réseaux dans lesquels l'information se propage de couche en couche sans retour en arrière possible. Parmi lesquels le perceptron (monocouche, multicouche), Les réseaux à fonction radiale, et Les réseaux "RandomVectorFunctional Link".

- **Les réseaux "FEED-BACK"**

Appelés aussi "réseaux récurrents", ce sont des réseaux dans lesquels il y a retour en arrière de l'information. Les cartes auto-organisatrices de Kohonen, les réseaux de Hopfield et les réseaux ART ("Adaptative ResonnanceTheory") sont quelques exemples. La Figure I.4 montre les architectures de quelques-uns des RNA qui existent.

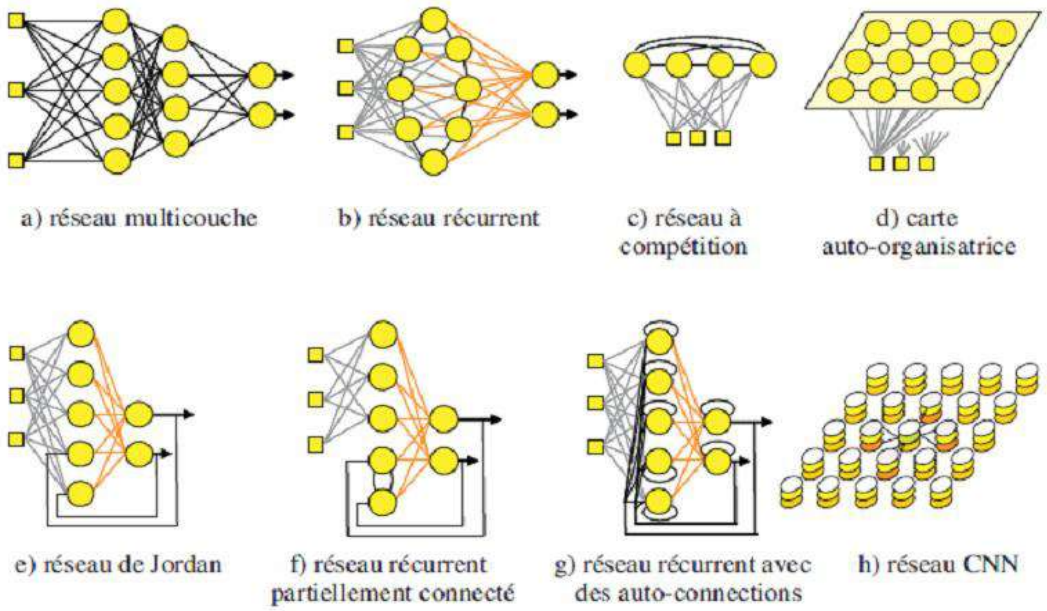


Figure I.4 : les architectures des réseaux de neurone artificiel les plus populaires.

I-2-3. Les réseaux de neurones non bouclés

Un réseau de neurone non bouclé est une composition de fonction réalisée par des neurones formels interconnectés entre eux. L'information circule de la couche d'entrée vers la couche de sortie par l'intermédiaire de connexions unidirectionnelles, sans bouclage. La figureI.5 explique la structure de réseau de neurone non bouclé. [7]

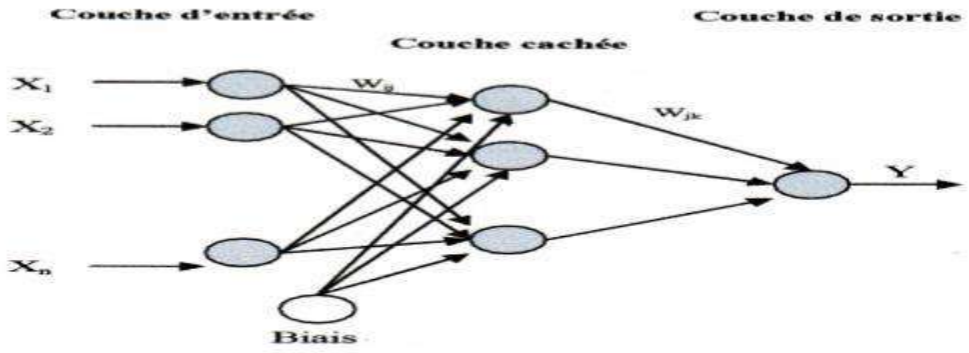


Figure I.5 : Architecture d'un réseau de neurone non bouclé.

I-2-4. Les réseaux de neurones bouclés.

Un réseau de neurones bouclé est un système dynamique, régi par des équations différentielles. Comme l'immense majorité des applications sont réalisées par des programmes d'ordinateurs, on se place dans le cadre des systèmes à temps discret, où les équations différentielles sont remplacées par des-équations aux différences. Un réseau de neurones bouclé à temps discret est donc régi par une (ou plusieurs) équations aux différences non linéaires, résultant de la composition des fonctions réalisées par chacun des neurones et des retards associés à chacune des connexions. [7]

Dans notre cas, le réseau bouclé est donc mieux adapté.La figure I.6 montre la structure de réseau de neurones bouclé.

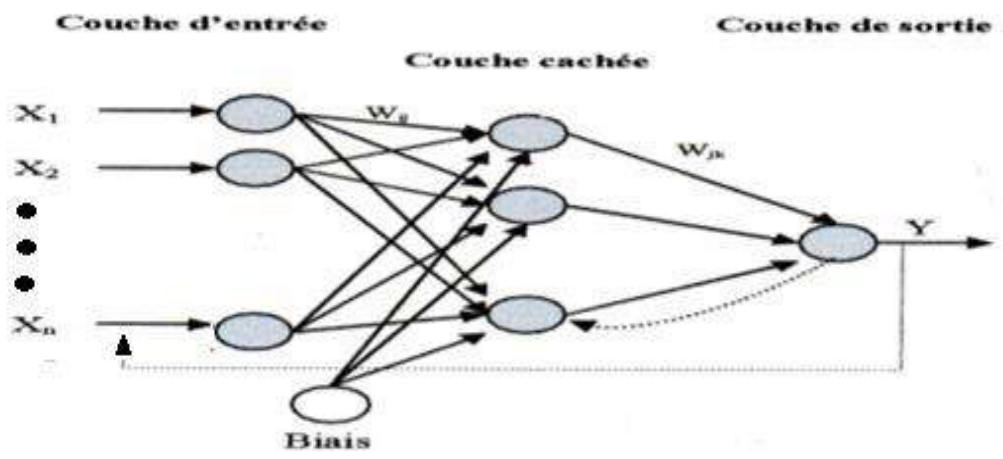


Figure I.6 Architecture d'un réseau de neurone bouclé.

I-3. Réseaux de neurones pour les systèmes dynamiques

Nous parlerons de systèmes chaque fois que l'on sera capable, sur une entité donnée de distinguer des entrées et des sorties liées par causalité. Par le terme de causalité, nous voulons simplement dire qu'une action sur les entrées engendre une réaction sur les sorties.

D'autre part, quand nous parlerons de modèle on peut dire que c'est une représentation abstraite permettant d'agréger l'ensemble des connaissances que l'on a du système. Les modèles considérés par la suite sont des représentations mathématiques, permettant de calculer l'évolution des sorties du système, lorsque l'on connaît l'évolution de ses entrées et les conditions initiales de cette évolution.[7]

La recherche d'un modèle mathématique du système étudié peut être abordée suivant deux approches distinctes : la première suppose qu'il est possible d'obtenir une description complète du système à l'aide d'un ensemble d'équations issues de l'application des lois de la physique en basant sur les bilans de matières et les bilans enthalpiques. Le modèle ainsi obtenu est dit de connaissance ou modèle boite blanche.

La seconde approche suppose que les mécanismes internes régissant le fonctionnement du système sont complètement inconnus. L'attitude consiste alors à rechercher un ensemble de relation rendant compte simplement du comportement entrée/sortie du système étudié c'est-à-dire qu'on ne s'intéresse pas aux phénomènes physique. Ce type de modèle est dit de représentation ou modèle boite noire, et c'est là que les réseaux de neurones trouvent leur place.[8]

I-3-1. Construction d'un modèle

La construction d'un modèle implique dans un premier temps le choix des échantillons des données d'apprentissage, de test et de validation. Le choix du type de réseau intervient dans un second temps.

Les quatre grandes étapes de la création d'un réseau de neurones sont détaillées comme suit.

- **Construction de la base de données et choix des échantillons** [9]

Le processus d'élaboration d'un réseau de neurones commence par la construction de la base de données et le choix des échantillons. Il faut donc déterminer les grandeurs qui ont une influence significative sur le phénomène que l'on cherche à modéliser. Cette étape est cruciale et aide à déterminer le type de réseau le plus approprié pour résoudre un problème donné. Afin de développer une application à base de réseaux de neurones, il est nécessaire de disposer de deux bases de données, une pour effectuer l'apprentissage et l'autre pour tester le réseau obtenu et déterminer ses performances.

Les valeurs d'entrées et de sortie sont normalisées dans un intervalle spécifique afin de donner à chaque paramètre la même influence statistique.

- **Définition de la structure du réseau**

La détermination de l'architecture optimale est restée longtemps un problème ouvert. Il existe actuellement diverses méthodes empiriques mettant notamment en jeu des tests

statistiques pour une vaste classe de réseaux.

Les réseaux multicouches statiques sont les plus utilisés à cause de la simplicité de leurs algorithmes d'apprentissage et leurs aptitudes à l'approximation et à la généralisation. Il n'existe pas de méthodes générales pour le choix du nombre de neurones sur chaque couche cachée ainsi que le nombre de ces dernières.

- **Apprentissage** [10]

La phase d'apprentissage consiste à calculer les coefficients synaptiques de telle manière que les sorties du réseau de neurones soient, pour les exemples utilisés lors de l'apprentissage, aussi proches que possibles des sorties ″ désirées ″. Celles-ci correspondent à la valeur de la fonction que l'on veut approcher ou à la sortie du processus que l'on veut modéliser. L'apprentissage correspond donc à la phase du développement du réseau durant laquelle les coefficients synaptiques sont modifiés jusqu'à l'obtention du comportement désiré.

Il existe de nombreux types de règles d'apprentissage qui peuvent être regroupées en trois catégories : les règles d'apprentissage supervisé, non supervisé et renforcé. Mais l'objectif fondamental de l'apprentissage reste le même : soit la classification, l'approximation de fonction ou encore la prévision.

- **Apprentissage supervisé**

Un apprentissage est dit supervisé lorsque l'on force le réseau à converger vers un état final précis, en même temps qu'on lui présente un motif. Ce genre d'apprentissage fait intervenir plusieurs exemples de types entrés- sorties. La procédure usuelle dans le cadre de la prévision est l'apprentissage supervisé qui consiste à associer une réponse spécifique désirée à chaque signal d'entrée. La modification des poids s'effectue progressivement jusqu'à ce que l'erreur entre les sorties du réseau et les résultats désirés soit minimisée.

- **Apprentissage renforcé**

L'apprentissage renforcé est une technique similaire à l'apprentissage supervisé à la différence qu'au lieu de fournir des résultats désirés au réseau, on lui accorde plutôt un grade qui est une mesure du degré de performance du réseau après quelques itérations. Les algorithmes utilisant la procédure d'apprentissage renforcé sont surtout utilisés dans le domaine des systèmes de contrôles.

- **Apprentissage non supervisé**

L'apprentissage non supervisé consiste à ajuster les poids à partir d'un seul ensemble d'apprentissage formé uniquement de données. Aucun résultat désiré n'est fourni au réseau. Dans ce cas l'apprentissage consiste à détecter les similarités et les différences dans l'ensemble d'apprentissage. Les poids et les sorties du réseau convergent, en théorie, vers les représentations qui capturent les régularités statistiques des données. Ce type d'apprentissage est également dit compétitif et /ou comparatif.

- **Validation**

Enfin, la dernière étape doit permettre d'estimer la qualité du réseau obtenu en lui présentant des exemples qui ne font pas partie de l'ensemble d'apprentissage. Elle permet à la fois d'apprécier les performances du système neuronal et de détecter le type de données qui posent problème. Si les performances ne sont pas satisfaisantes, il faudra, soit modifier l'architecture du réseau, soit modifier la base d'apprentissage.

I-3-2. Modélisation de procédé

La modélisation par les réseaux de neurones passe par des étapes :

- Le choix d'une architecture adéquate du réseau, c'est-à-dire le nombre de neurones dans la couche d'entrée et la couche cachée.
- La détermination des poids qui correspondent à une reproduction du comportement du procédé par le réseau. [4]

Nous déterminons le nombre de neurones dans la couche d'entrée de la façon suivante :

Considérons un processus qui contient une entrée u et une sortie réelle y. Le réseau neuromimétique génère une sortie future $\hat{y}(k+1)$ à partir des entrées présente et passées ainsi que les sorties présentes et passées. Pour obtenir un modèle décrivant ce processus nous déterminons une certaine fonction F relient la sortie à l'entrée, c'est-à-dire nous simulons cette fonction de transfert F du procédé par une équation de type :

$$\hat{y}(k+1) = \mathrm{F}[(u(k), u(k-1), u(k-2) \dots y(k), y(k-1), y(k-2) \dots]$$

$y(k)$: La sortie actuelle du système.

$u(k)$: L'entrée actuelle du système.

$F(.)$: Fonction non linéaire quelconque

On est donc ramené à l'approximation d'une fonction non linéaire F. L'erreur de prédiction (écart entre la sortie réelle et la sortie générée par le modèle) est alors définie par :

$$e(k+1) = y(k+1) - \hat{y}(k+1) \qquad \textbf{(I.3)}$$

I-3-3. Identification par réseaux de neurones

Identifier et modéliser un système revient à analyser et à comprendre son fonctionnement. Cela revient à connaitre ses états, à distinguer ses paramètres significatifs et à caractériser sa structure, cela se traduit par l'assimilation d'un système à un modèle paramétrique théorique. D'un point de vue plus pratique, on peut observer et mesurer certaines grandeurs qui en sont issues[11]. La figure I.7 montre l'identification d'un système non linéaire par le réseau des neurones. L'objectif de l'identification d'un processus est de construire un modèle capable, lorsqu'il est soumis aux mêmes entrées que le processus, de restituer une sortie suffisamment proche de celle du processus, au sens d'une certaine norme.

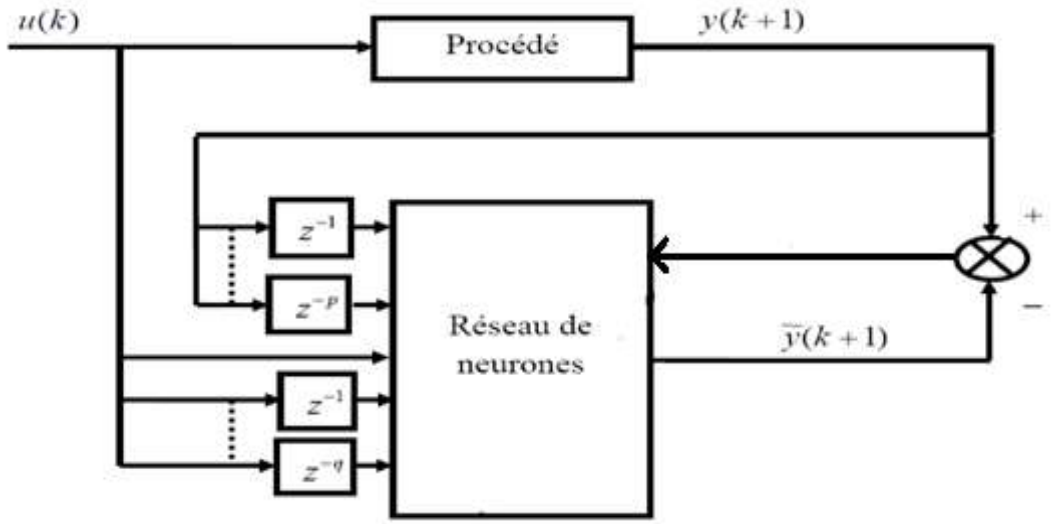


Figure I.7 Identification d'un système dynamique non linéaire par le réseau de neurone.

I-3-4. Les avantages des réseaux de neurones [7]

Il y a beaucoup d'avantages des réseaux de neurones parmi lesquelles :

- La capacité de modéliser des problèmes complexes.
- Une plus grande liberté et flexibilité que les approches conventionnelles.
- La capacité d'extraire l'information nécessaire à partir d'un nombre fini d'exemples.
- La capacité d'apprendre avec des données qui contiennent des bruits.
- La capacité de modéliser des problèmes qui présentent une grande variété de données multidimensionnelles.

I-3-5. Les inconvénients des réseaux de neurones [7]

- La capacité de prédiction n'est pas toujours améliorée avec plus d'apprentissage, les résultats obtenus avec lesquels il a été entraîné sont précis, mais la prédiction avec les données qui n'appartiennent pas à l'ensemble d'entraînement est faible (problème de sur - apprentissage)
- Il peut avoir une erreur moyenne très petite avec les données d'apprentissage, mais il peut ne pas justifier cette performance avec les données d'un test indépendant (problème de généralisation).

Chapitre II : Partie pratique et modélisation

II-1. Introduction

Les réseaux de neurones peuvent approximer n'importe quel fonction continue avec suffisamment de précision à mesure que le nombre de neurones augmente sans limite. En outre, ils ont été appliqués avec succès dans beaucoup de problèmes complexes y compris la reconnaissance de la parole, l'identification des chiffres manuscrits, le diagnostic médical et la modélisation de divers systèmes physiques, chimiques, économiques, ou autres. [12]

Dans cette partie, nous avons essayé de modéliser un système dynamique décrit par une équation différentielle. Il s'agit en l'occurrence d'un réacteur continu parfaitement agité. On s'est proposé une architecture du réseau, et nous vérifions la capacité de ce dernier à modéliser un procédé chimique.

II-2.Base et algorithme d'apprentissage

Les réseaux de neurones artificiels ont la capacité de stoker de la connaissance empirique et de la rendre disponible à l'usage. Les habiletés de traitement des réseaux vont être stockées dans les poids synaptiques, obtenus par des processus d'adaptation ou d'apprentissage.[13]

L'objectif de l'apprentissage est de fourni une méthode aux réseaux afin qu'il puisse ajuster ses paramètres lorsqu'on lui présente des exemples à traiter.

La base d'apprentissage comprend n entrées $\vec{e}$ et les n réponses souhaitées, donc l'algorithme d'apprentissage devient comme suivant :

- Initialiser les pois w_i avec des valeurs aléatoires.
- Présenter une entrée $\vec{e}$ de la base d'apprentissage.
- Calculer la valeur d'activation des neurones.
- Calculer l'erreur sur la sortie.
- Modifier les poids. (si la valeur d'activation du neurone est égale à la réponse désirée, les poids ne sont pas modifiés).
- Retourner à l'étape 2 jusqu'à ce qu'on obtient la réponse désirée

Plusieurs algorithmes ont été développés pour résoudre le problème donné, le plus connu étant l'algorithme de rétropropagation qui a pour objectif l'adaptation les poids des connexions du réseau de manière à minimiser la somme des erreurs sur tous les neurones de sortie.

II-3.L'algorithme de rétropropagation. [14]et [15]

Soit le couple $\vec{x}(n)$ et $\vec{d}(n)$ désignant la n^edonnée d'entrainement du réseau ou :

$$\vec{x}(n) =< x_1(n), \dots, x_p(n) > \text{ et } \vec{d}(n) = d_1(n), \dots, d_q(n) \quad \textbf{(II.1)}$$

Correspondent respectivement aux p entrées et aux q sorties désirées du système. Cet algorithme consiste, après avoir déterminé l'erreur entre la sortie obtenue par propagation vers l'avant des entrées $\vec{x}(n)$, c'est-à-dire $\vec{y}(n)$ et la sortie exigée $\vec{d}(n)$ exprimée par l'équation II.2, à rétropropager cette erreur à travers les couches du réseau en allant des sortie vers les entrées.

$$\vec{y}(n) =< y_1(n), \dots, y_q(n) > \quad \textbf{(II.2)}$$

II.3-1.Cas de la couche de sortie

L'algorithme de rétropropagation procède à l'adaptation des poids neurone par neurone en commençant par la couche de sortie. Soit l'erreur observée $e_j(n)$ pour le neurone de sortie j et la donnée d'en trainement n :

$$e_j(n) = d_j(n) - y_j(n) \quad \textbf{(II.3)}$$

Où $d_j(n)$ correspond à la sortie désirée du neurone j et $y_j(n)$ à sa sortie observée.

- La variable n représentera toujours la donnée d'entrainement c'est-`a-dire le couple contenant un vecteur d'entrées et un vecteur de sorties désirées.
- L'indice j représentera toujours le neurone pour lequel on veut adapter les poids.
- On voit que cette technique dérive de la méthode du gradient qui consiste à minimiser l'erreur globale sur toute la base de données par l'équation ci-dessous :

$$E(n) = \frac{1}{2}\sum_{j\in C} e_j^2\,(n) \quad \textbf{(II.4)}$$

Ou E(n)est la somme des erreurs quadratiques observées et C est l'ensemble des neurones de sortie.

La sortie $y_j(n)$ du neurone j est d´définie par :

$$y_j(n) = \varphi\left[v_j(n)\right] = \varphi\left[\sum_{i=0}^{r} w_{ji}\,(n) y_i(n)\right] \qquad \textbf{(II.5)}$$

Où

$\varphi[\,.\,]$ est la fonction d'activation du neurone

$v_j(n)$ est la somme pondérée des entrées du neurone j

$w_{ji}(n)$ est le poids de la connexion entre le neurone $i$de la couche précédente et le neurone j de la couche courante.

$y_i(n)$ est la sortie du neurone$i$.

On suppose ici que la couche précédente contient r neurone numérotés de 1 à r, que les poids $w_{j0}(n)$ correspond au biais du neurone j et que l'entrée $y_0(n)$ = 1.

Notons que l'indice $i$représentera toujours un neurone sur la couche précédente par rapport au neurone$j$.

En introduisant cette erreur, le réseau va agir sur les poids, puisqu'ils sont les seuls paramètres maniables, pour réduire au minimum l'erreur calculée.

La correction des poids se fait par la modification les poids $w_{ji}(n)$ dans le sens opposé au gradient $\frac{\partial E(n)}{\partial w_{ji}(n)}$ de l'erreur. Cette dérivée partielle représente un facteur de sensibilité.

Puisque il y a r neurone sur la couche précédent la couche de sortie, il y a aussi r poids à adapter.

Par la règle de chaînage des dérivées partielles, qui nous dit que :

$$\frac{\partial \mathrm{f(y)}}{\partial \mathrm{x}} = \frac{\partial \mathrm{f(y)}}{\partial \mathrm{y}} \times \frac{\partial \mathrm{y}}{\partial \mathrm{x}}$$

On obtient :

$$\frac{\partial \mathrm{E(n)}}{\partial \mathrm{w_{ji}(n)}} = \frac{\partial \mathrm{E(n)}}{\partial \mathrm{e_j(n)}} \cdot \frac{\partial \mathrm{e_j}}{\partial \mathrm{y_j}} \cdot \frac{\partial \mathrm{y_j}}{\partial \mathrm{v_j}} \cdot \frac{\partial \mathrm{v_j(n)}}{\partial \mathrm{w_{ji}(n)}} \qquad \textbf{(II.6)}$$

Et on exprime la variation de poids $\Delta w_{ji}(n)$sous la forme suivante :

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)} \qquad \textbf{(II.7)}$$

Avec$0 \leq \eta \leq 1$représentant un taux d'apprentissage ou gain de l'algorithme.

Evaluons maintenant chacun des termes du gradient.

$$\frac{\partial E(n)}{\partial e_j(n)} = \frac{\partial\left[\frac{1}{2}\sum_{k\in C} e_k^2(n)\right]}{\partial e_j(n)}$$

$$= \frac{1}{2} \cdot \frac{\partial e_j^2(n)}{\partial e_j(n)}$$

$$= e_j(n)$$

$$\frac{\partial e_j(n)}{\partial y_j(n)} = \frac{\partial\left[d_j(n) - y_j(n)\right]}{y_j(n)}$$

$$= -1$$

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \frac{\partial\left[\frac{1}{1+e^{-v_j(n)}}\right]}{\partial v_j(n)}$$

$$= \frac{e^{-v_j(n)}}{\left[1+ e^{-v_j(n)}\right]^2}$$

$$= y_j(n)\left[\frac{e^{-v_j(n)}}{1+ e^{-v_j(n)}}\right]$$

$$= y_j(n)\left[\frac{e^{-v_j(n)}+1}{1+e^{-v_j(n)}} - \frac{1}{1+e^{-v_j(n)}}\right]$$

$$= y_j(n)\left[1 - y_j(n)\right]$$

Et finalement :

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = \frac{\partial\left[\sum_{l=0}^{r} w_{jl}(n) y_l(n)\right]}{\partial w_{ji}(n)}$$

$$= \frac{\partial[w_{ji}(n)y_i(n)]}{\partial w_{ji}(n)}$$

$$= y_i(n)$$

Nous obtenons donc :

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -e_j(n)y_j(n)[1 - y_j(n)]y_j(n) \quad \textbf{(II.8)}$$

Et la règle dite ″delta″ pour la couche de sortie s'exprime par

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}} = \eta\delta_j(n)y_i(n) \quad \textbf{(II.9)}$$

Avec : $$\delta_j = e_j(n)y_i(n)[1 - y_j(n)] \quad \textbf{(II.10)}$$

Qui correspond à ce que l'on appelle ″gradient local″.
Jusqu'ici, nous avons traité seulement le cas de la couche de sortie. Il reste maintenant à faire l'adaptation des poids sur les couches cachées. Mais le problème est qu'on ne dispose plus de l'erreur observée.

II.3-2.Cas d'une couche cachée

Considérons maintenant le cas des neurones sur la dernière couche cachée (le cas des autres couches cachées est semblable).

La variable n désignera toujours la donnée d'entrainement c'est-à-dire un couple de vecteurs d'entrées et de sortie désirées.

L'objectif sera toujours l'adaptation des poids de la couche courante en minimisant la somme des erreurs sur les neurones de la couche de sortie. Les indices i et j désignent respectivement (comme précédemment) un neurone sur la couche précédente et un neurone sur la couche courante. L'indice k servira maintenant à désigner un neurone sur la couche suivante.

Reprenons l'expression de la dérivée partielle de l'erreur totale $E(n)$ par rapport à $w_{ji}(n)$ mais en ne dérivant plus par rapport à l'erreur $e_j(n)$ car celle-ci est maintenant inconnue :

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial y_j(n)} \cdot \frac{\partial y_{j(n)}}{\partial v_{j(n)}} \cdot \frac{\partial v_j(n)}{\partial w_{ji}(n)} \qquad \textbf{(II.11)}$$

Par rapport aux résultats obtenus pour la couche de sortie, les deux derniers termes de cette équation restent inchangés, seul le premier terme requiert d'être évalué :

$$\frac{\partial E(n)}{\partial y_j(n)} = \frac{\partial\left[\frac{1}{2}\sum_{k\in C} e_k^2(n)\right]}{\partial y_j(n)} \qquad \textbf{(II.12)}$$

Notre problème ici, contrairement au cas des neurones de la couche de sortie, est que tous les $e_j(n)$ dans la somme ci-dessus dépendent de $y_j(n)$.On ne peut donc pas se débarrasser de cette somme ! Néanmoins, nous pouvons écrire :

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_{k\in C}\left[e_k(n).\frac{\partial e_k(n)}{\partial y_j(n)}\right]$$

$$= \sum_{k\in C}\left[e_k(n).\frac{\partial e_k(n)}{\partial v_k(n)}.\frac{\partial v_k(n)}{\partial y_j(n)}\right]$$

$$= \sum_{k\in C}\left[e_k(n).\frac{\partial[d_k(n)-\varphi(v_k(n))]}{\partial v_k(n)}.\frac{\partial[\sum_l w_{kl}(n)y_l(n)}{\partial y_j(n)}\right]$$

$$= \sum_{k\in C}\left[e_k(n).(-y_k(n)[1-y_k(n)]).w_{kj}\right]$$

Et en substituant l'équation II.10 dans cette dernière on obtient :

$$\frac{\partial E(n)}{\partial y_j(n)} = -\sum_{k\in C}\delta_k(n)w_{kj}(n) \qquad \textbf{(II.13)}$$

Ensuite en substituant l'équation II.13 dans l'équation II.11, on obtient :

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -y_j(n)\left[1-y_j(n)\right]\left[\sum_{k\in C}\delta_k(n)w_{kj}(n)\right]y_i(n) \qquad \textbf{(II.14)}$$

Finalement,

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)} = \eta \delta_j(n) y_i(n) \quad \textbf{(II.15)}$$

Avec : $\delta_j(n) = y_j(n)\left[1 - y_j(n)\right]\left[\sum_{k \in C} \delta_k(n) w_{kj}(n)\right]$ **(II.16)**

On peut démontrer que les équations (II.15) et (II.16) sont valides pour toutes les couches cachées.

Noter bien, cependant, que dans le cas de la première couche cachée du réseau, puisqu'il n'ya pas de couche précédente de neurones, il faut substituer la variable $y_j(n)$ par l'entrée $x_i(n)$.L'algorithme de rétropropagation standard se résume donc à la série d'étapes suivantes :

1. Initialiser tous les poids à de petites valeurs aléatoires dans l'intervalle [0,1].
2. Normaliser les données d'entrainement.
3. Pour chaque donnée d'entrainement r:

a. Calculer les sorties observées en propageant les entrées vers l'avant.

b. Ajuster les poids en rétropropageant l'erreur observée comme suit :

$$w_{ji}(n) = w_{ji}(n-1) + \Delta w_{ji}(n) = w_{ji}(n-1) + \eta \delta_j(n) y_i(n) \quad \textbf{(II.17)}$$

Où le ″gradient local″ est défini par :

$$\delta_j(n) = \begin{cases} e_j(n) y_j(n)\left[1 - y_j(n)\right] & \text{Si } j \in \text{couche de sortie} \\ y_j(n)\left[1 - y_j(n)\right] \sum_k \delta_k(n) w_{kj}(n) & \text{Si } j \in \text{couche cachée} \end{cases} \quad \textbf{(II.18)}$$

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

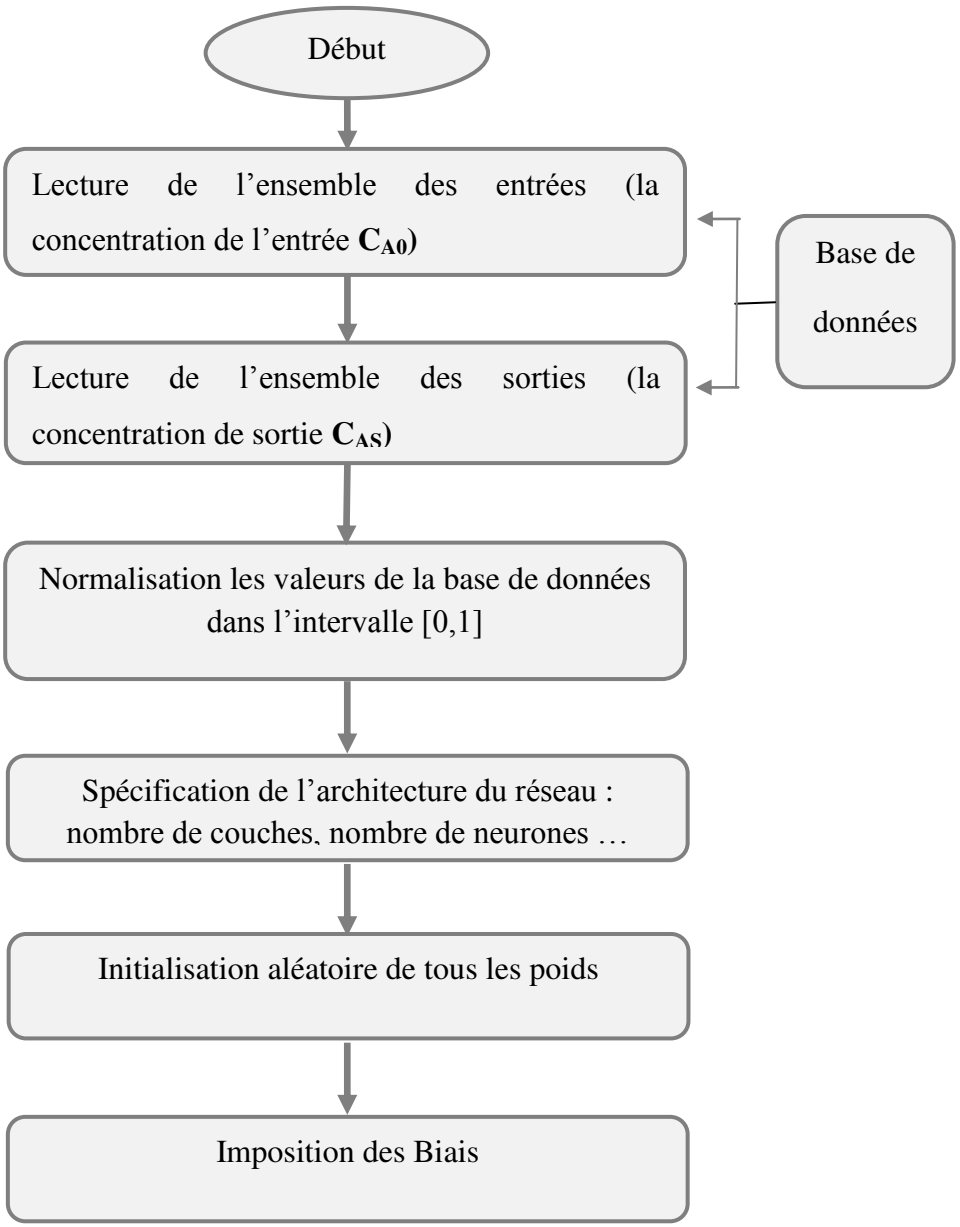
4. Répéter les étapes 3 et 4 jusqu'à un nombre maximum d'itérations ou jusqu'à ce que la racine de l'erreur quadratique moyenne (EQM) soit inférieure à un certain seuil.

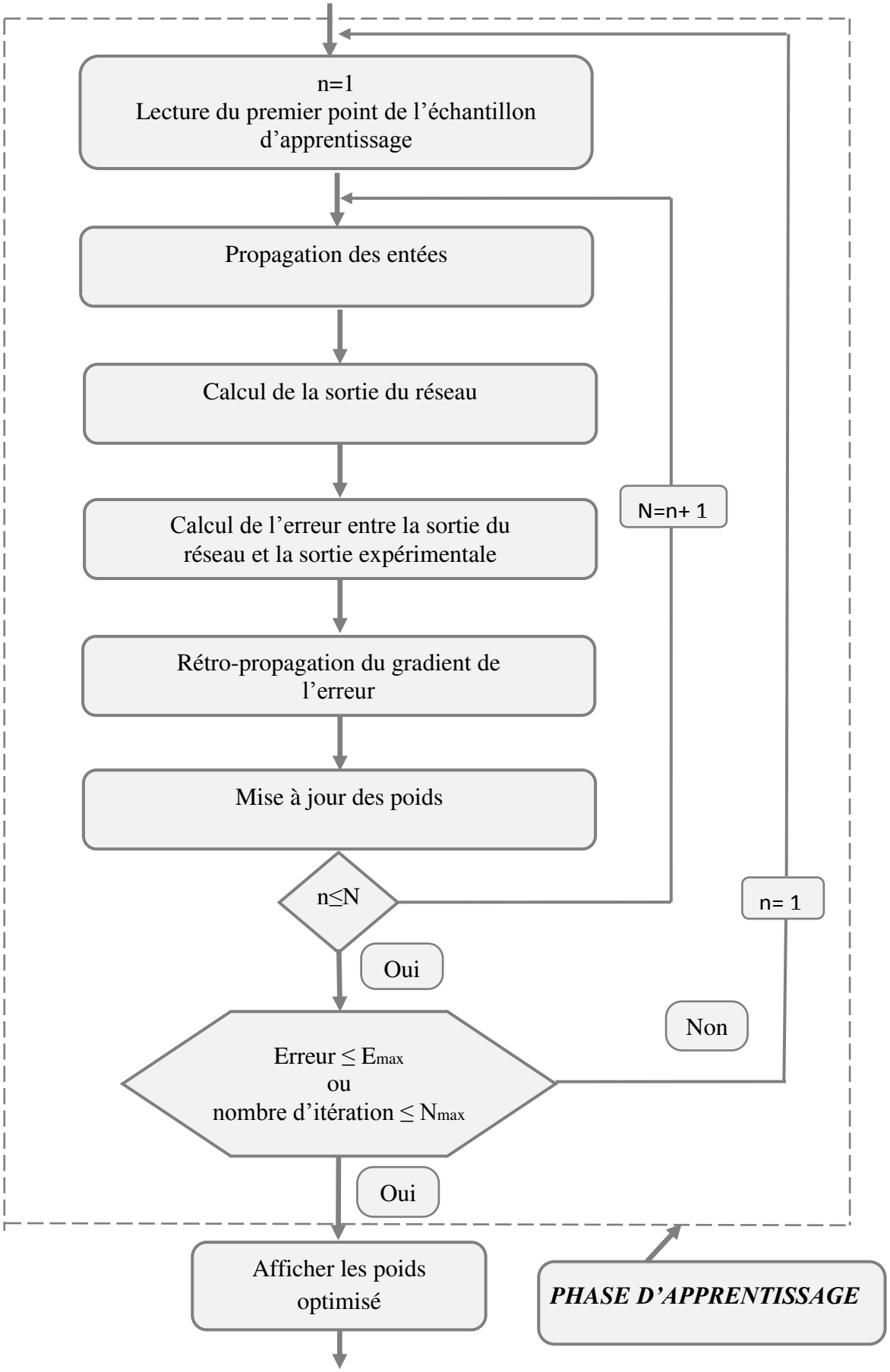
L'équation (II.17) décrit ce qu'on appelle la règle du ″delta″ pour l'algorithme de rétropropagation des erreurs. L'équation suivante, nommé règle du ″ delta généralisé″, décrit une autre variante de l'algorithme :

$$w_{ji}(n) = w_{ji}(n-1) + +\eta\delta_j(n)y_i(n) + b.\Delta w_{ji}(n-1) \qquad \textbf{(II.19)}$$

Ou $0 \leq b \leq 1$ un paramètre nommé ″momentum″ qui représente une espèce d'inertie dans le changement des poids.

La figure II-2 présent l'organigramme de notre modèle qui explique les étapes de calcul.




n=1
Lecture du premier point de l'échantillon d'apprentissage
Propagation des entées
Calcul de la sortie du réseau
Calcul de l'erreur entre la sortie du réseau et la sortie expérimentale
Rétro-propagation du gradient de l'erreur
Mise à jour des poids
n≤N
N=n+ 1
Oui
Non
n= 1
Erreur ≤ Emax
ou
nombre d'itération ≤ Nmax
Oui
Afficher les poids optimisé
PHASE D'APPRENTISSAGE

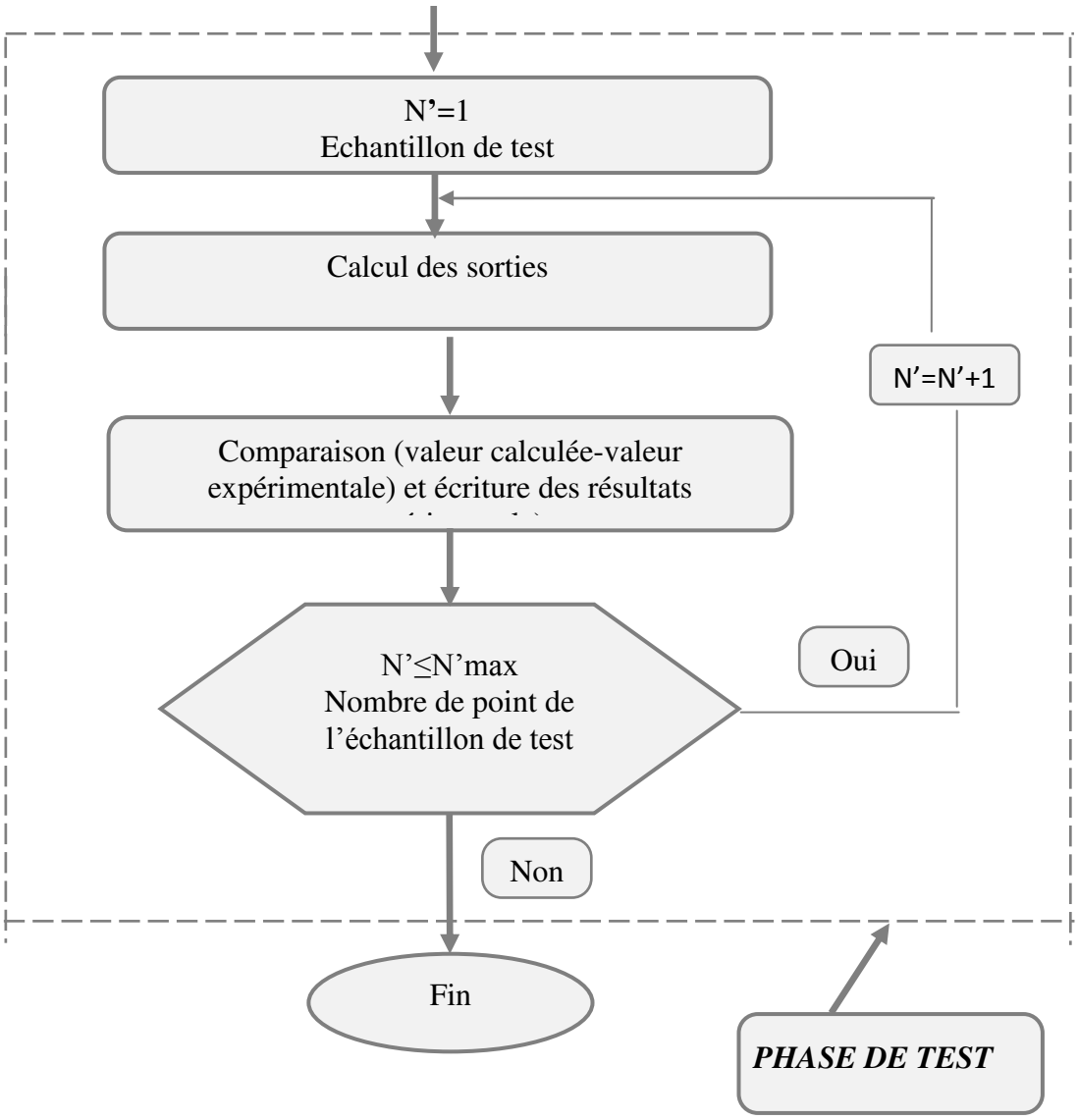


Figure II.1 Organigramme représentant l'algorithme de réseau de neurone du modèle.

II.4.Présentation du système modélisé

Nous avons choisi un réacteur continu parfaitement agité dans lequel une réaction chimique de premier ordre se produit. La réaction de type : $A \rightarrow B$ est supposée isotherme de premier ordre se produisant en phase liquide de sorte que le débit volumique soit constant. Il s'agit donc d'un système à une seule entrée (concentration de réactif à l'entrée du réacteur) et une seule sortie (la même concentration dans le fluide se sortie) (SISO). On cherche alors à modéliser le comportement dynamique de la concentration de sortie lorsque la concentration de l'entrée est soumise à une perturbation. Afin de construire notre base de données pour apprentissage et pour test, on procède à l'établissement de l'équation de bilan de matière sur le réactif A et ensuite, on la discrétise pour pouvoir générer des valeurs ponctuelles. Pour cela, nous utiliserons deux méthodes distinctes basées respectivement sur la transformée de Laplace (solution continue exacte du bilan de masse) et sur la transformée en Z (discrétisation de l'équation différentielle continue).

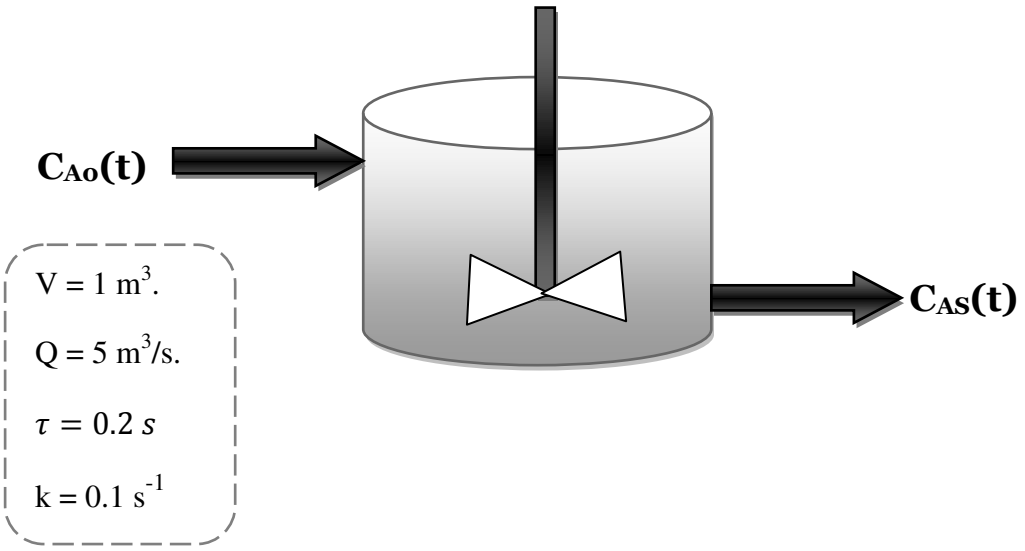


Figure II.2: Schéma de réacteur continu parfaitement agité de procédé.

II.4-1. La première méthode

Bilan de matière sur le réactif A:

$$QC_{A0} - QC_{AS} - kC_{AS} = V\frac{dC_{AS}}{dt}$$ **(II.20)** En régime transitoire.

$$Q\widetilde{C_{A0}} - Q\widetilde{C_{AS}} - k\widetilde{C_{AS}} = 0$$ **(II.21)** En régime permanent.

(II.20)_(II.21) → $Q\underbrace{\left(C_{A0}-\widetilde{C_{A0}}\right)}_{\mathbf{C'_{A0}}}-Q\underbrace{\left(C_{AS}-\widetilde{C_{AS}}\right)}_{\mathbf{C'_{AS}}}-k\underbrace{\left(C_{AS}-\widetilde{C_{AS}}\right)}_{\mathbf{C'_{AS}}}=V\frac{dC_{AS}}{dt}$ **(II.22)**

Donc
$$QC'_{A0}(t)-(Q+k)C'_{AS}(t)=V\frac{dC'_{AS}}{dt}$$

$$C'_{A0}(t)-(1+\frac{k}{Q})C'_{AS}(t)=\frac{V}{Q}.\frac{dC'_{AS}}{dt}$$

Alors **(II.22)** sera
$$C'_{A0}(t)-(1+\frac{k}{Q})C'_{AS}(t)=\tau.\frac{dC'_{AS}}{dt} \qquad \textbf{(II.23)}$$

Les nouvelles variables C'_{A0} et C'_{AS} sont des variables d'écart.

Considérons le cas où la concentration de l'entrée C'_{A0} change en fonction du temps selon la forme d'une fonction rampe :

$$\mathcal{F}\left(C'_{A0}\right)=C'_{A0}(t)=C'_{A0}(t=0)+0.01t \qquad \textbf{(II.24)}$$

La transformée de LAPLACE de **(II.24)** s'écrit :

$$\mathcal{L}\left[\mathcal{F}\left(C'_{A0}\right)\right]=\overline{C'_{A0}}(s)=\frac{C'_{A0}(t=0)}{s}+\frac{0.01}{s^2} \qquad \textbf{(II.25)}$$

D'autre part, la transformée de LAPLACE de **(II.23)** s'écrit :

$$\mathcal{L}\left[C'_{A0}(t)-(1+\frac{k}{Q})C'_{AS}(t)=\tau\frac{dC'_{AS}}{dt}\right]$$

$$\overline{C'_{A0}}(s)-(1+\frac{k}{Q})\overline{C'_{AS}}(s)=\tau s.\overline{C'_{AS}}(s)$$

$$\overline{C'_{AS}}(s)\left[(1+\frac{k}{Q})+\tau s\right]=\overline{C'_{A0}}(s)$$

Donc
$$\overline{C'_{AS}}(s)=\frac{1}{\left[1+\frac{k}{Q}+\tau s\right]}\cdot\overline{C'_{A0}}(s)$$

On simplifie $$\overline{C'_{AS}}(s) = \frac{1}{1+\frac{k}{Q}+\tau s}.\overline{C'_{A0}}(s)$$

$$\overline{C'_{AS}}(s) = \frac{1}{\frac{Q+k}{Q}+\tau s}.\overline{C'_{A0}}(s)$$

$$\overline{C'_{AS}}(s) = \frac{1}{\frac{Q+k+Q\tau s}{Q}}.\overline{C'_{A0}}(s)$$

Et aussi $$\overline{C'_{AS}}(s) = \frac{Q}{Q+k+Q\tau s}.\overline{C'_{A0}}(s)$$

Donc $$\overline{C'_{AS}}(s) = \frac{Q}{Q+k}\cdot\frac{1}{1+\frac{Q\tau}{Q+k}s}.\overline{C'_{A0}}(s)$$

On met : $$\frac{Q\tau}{Q+k} = \mu$$

Alors $\overline{C'_{AS}}(s)$ devient $$\overline{C'_{AS}}(s) = \frac{Q}{Q+k}\cdot\frac{1}{1+\mu s}.\overline{C'_{A0}}(s) \qquad \textbf{(II.26)}$$

La fonction de transfert est $$G(s) = \frac{\overline{C'_{AS}}}{\overline{C'_{A0}}} = \left[\frac{Q}{Q+k}\cdot\frac{1}{1+\mu s}\right]$$

On remplace **(II.25)** dans **(II.26)** : $$\overline{C'_{AS}}(s) = \frac{Q}{Q+k}\cdot\frac{1}{1+\mu s}.\left[\frac{C_{A0}}{s}+\frac{0.01}{s^2}\right]$$

Et finalement :

$$\overline{C'_{AS}}(s) = \frac{C_{A0}\,Q}{Q+k}\cdot\frac{1}{s(1+\mu s)}+\frac{0.01Q}{Q+k}\cdot\frac{1}{s^2(1+\mu s)}$$

Notre système dynamique de 1er degré est ainsi décrit par l'équation continue suivante après la transformation l'inverse de Laplace :

$$C_{AS}(t) = \frac{C_{A0}\,Q}{Q+k}\cdot\left[1-e^{-t/\mu}\right]+\frac{0.01Q}{Q+k}\cdot\left[t-\mu+\mu.e^{-t/\mu}\right]$$

II.4-2. La deuxième méthode :

Dans cette méthode on introduit la notion de l'opérateur dit « zero order hold » qui est défini comme un modèle mathématique de la reconstruction pratique du signal effectuée par un convertisseur numérique-analogique conventionnel, il décrit l'effet de convertir un signal de temps discret en un signal en temps continu en maintenant chaque valeur d'échantillon pour un intervalle d'échantillonnage.

Il s'exprime dans le domaine de Laplace comme suit :

$$\mathrm{H(s)} = \frac{1 - \exp((-\Delta \mathrm{t}).\mathrm{s})}{\mathrm{s}}$$

Δt: est le pas de temps considéré lors de la discrétisation.

On multiplie cet opérateur par la fonction de transfert analogique G(s) et on passe à la transformée en Z :

$$H(s).G(s) = \frac{1 - \exp((-\Delta t).\mathrm{s})}{s}.G(s)$$

$$Z[H(s).G(s)] = Z\left[\frac{1-\exp((-\Delta t).\mathrm{s})}{s}.G(s)\right]$$

Par ailleurs $Z(e^{-(\Delta t)s}) = Z^{-1}$

Avec Z : opérateur de retard d'un temps Δt.

Il s'ensuit alors :

$$\mathrm{Z[H(s).G(s)]} = (1 - Z^{-1}) \times Z\left[\frac{G(s)}{s}\right]$$

$$\mathrm{H(Z)} = (1 - \mathrm{Z}^{-1}) \times \mathrm{Z}\left(\frac{\left[\frac{\mathrm{Q}}{\mathrm{Q+k}}.\frac{1}{1+\mu \mathrm{s}}\right]}{\mathrm{S}}\right)$$

$$\mathrm{H(Z)} = (1 - \mathrm{Z}^{-1}) \times \frac{\mathrm{Q}}{\mathrm{Q+k}}.\mathrm{Z}\left(\frac{\frac{1}{\mu}}{\frac{1}{\mu}\cdot \mathrm{s}+\mathrm{s}^2}\right) \qquad \textbf{(II.27)}$$

La transformée en Z de (II.27)

$$H(Z) = (1 - Z^{-1}) \times \frac{Q}{Q+k} \cdot \frac{\left(1-e^{-\Delta t/\mu}\right)Z^{-1}}{\left(1-e^{-\Delta t/\mu}\ Z^{-1}\right).(1-Z^{-1})} \qquad \textbf{(II.28)}$$

La fonction de transfert discrète $H(Z) = \frac{\overline{\overline{C_{AS}}}}{\overline{\overline{C_{A0}}}}$ **(II.29)**

Qui est le rapport, dans le domaine Z de la sortie du procédé et de son entrée.

Après l'égalité entre les **(II.28)** et **(II.29),** nous obtenons

$$\frac{\overline{\overline{C_{AS}}}}{\overline{\overline{C_{A0}}}} = \frac{Q}{Q+k} \cdot \frac{\left(1-\exp\left(-\frac{\Delta t}{\mu}\right)\right).Z^{-1}}{\left(1-\exp\left(-\frac{\Delta t}{\mu}\right)Z^{-1}\right)}$$

Donc $\overline{\overline{C_{AS}}}.(Q+k).\left(1-e^{-\Delta t/\mu}.Z^{-1}\right) = \overline{\overline{C_{A0}}}.Q.\left(Z^{-1} - Z^{-1}.e^{-\Delta t/\mu}\right)$

Et aussi :

$$(Q+k)\cdot\left[\overline{\overline{C_{AS}}}(Z) - Z^{-1}\overline{\overline{C_{AS}}}(Z)e^{-\Delta t/\mu}\right] = Q.\left[Z^{-1}\overline{\overline{C_{A0}}}(Z) - Z^{-1}\overline{\overline{C_{A0}}}(Z).e^{-\Delta t/\mu}\right]$$

Alors :

$$(Q+k).\left[C_{AS}(k) - C_{AS}(k-1).e^{-\Delta t/\mu}\right] = Q.\left[C_{A0}(k-1) - C_{A0}(k-1).e^{-\Delta t/\mu}\right]$$

Notre système dynamique est ainsi décrit par l'équation discrète suivante après introduction des indices k correspondant aux étapes dans le temps :

$$C_{AS}(k) = \left(\frac{Q}{(Q+k)}\right).\left[C_{A0}(k-1)(1-e^{-\Delta t/\mu})\right] + C_{AS}(k-1).e^{-\Delta t/\mu}$$

La figure II.3 montre les courbes qui représentent les équations continu et discrète du concentration de sortie C_{AS}.

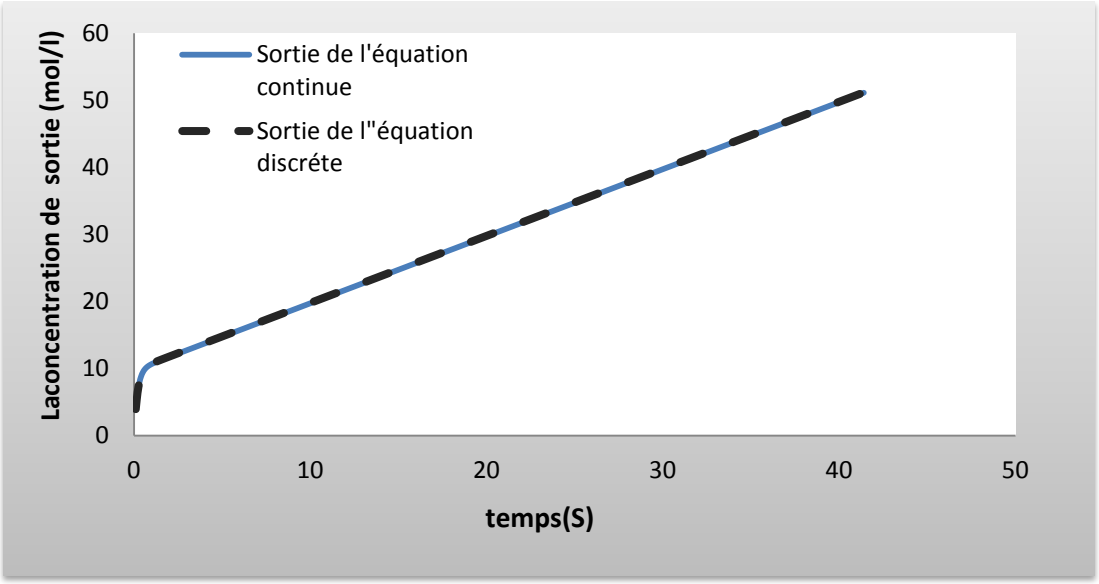


Figure II.3 : Les courbes des équations continu et discrète du C_{AS}

Notons que les deux courbes entièrement compatibles, ce qui signifie que la réponse dynamique fournie par l'équation discrète est exactement celle fournie par la solution exacte de l'équation continue. Ainsi, pour la suite, nous allons employer l'équation discrète pour générer notre base de données.

II-5.L'utilisation de logiciel Fortran

Dans notre partie pratique, pour effectuons les calculs décrits dans l'organigramme précédent, nous avons développé un programme écrit en Fortran 95. C'est un langage de programmation utilisé principalement pour le calcul scientifique. C'est le plus ancien langage de programmation. Pour la génération de la base de données, c'est le logiciel Microsoft Excel qui a été utilisé.

II-6. Synthèse du modèle neuromémitique du réacteur continu parfaitement agité

Nous allons passer à la synthèse du modèle décrivant son fonctionnement, ce serait après l'identification du processus.

Nous avons créé une base de données constituée de 199 exemples. Ces exemples ont été obtenus en donnant des valeurs aléatoires à la variable d'entrée et en calculant les variables de sorties correspondantes.

Notre réseau de neurones compte cinq entrées : u(k-2), u(k-1), u(k), y(k-1), y(k).

le réseau compte une seule sortie qui est y(k+1). Le nombre de neurones dans la couche cachée a été modifié et plusieurs simulations ont été effectuées.

Deux bases de données ont été utilisées pour le développement du modèle : une base d'apprentissage (199 points) et une base de test (90 points) pour la validation du modèle. La figure II.4 représente l'architecture du réseau effectué.

II-7.Présentation d'une simulation type

Rappelons que nous avons simulée l'équation différentielle discrète suivante

$$C_{AS}(k) = C_{A0}(k-1)(1-e^{-T/\tau}) + C_{AS}(k-1).e^{-T/\tau}$$

Pour ce faire, nous avons fixé l'architecture suivante :

- Nombre de neurone dans la couche d'entrée avec le biais : 06
- nombre de neurone dans la couche cachée avec le biais : 04
- nombre de neurone dans la couche de sortie : 01

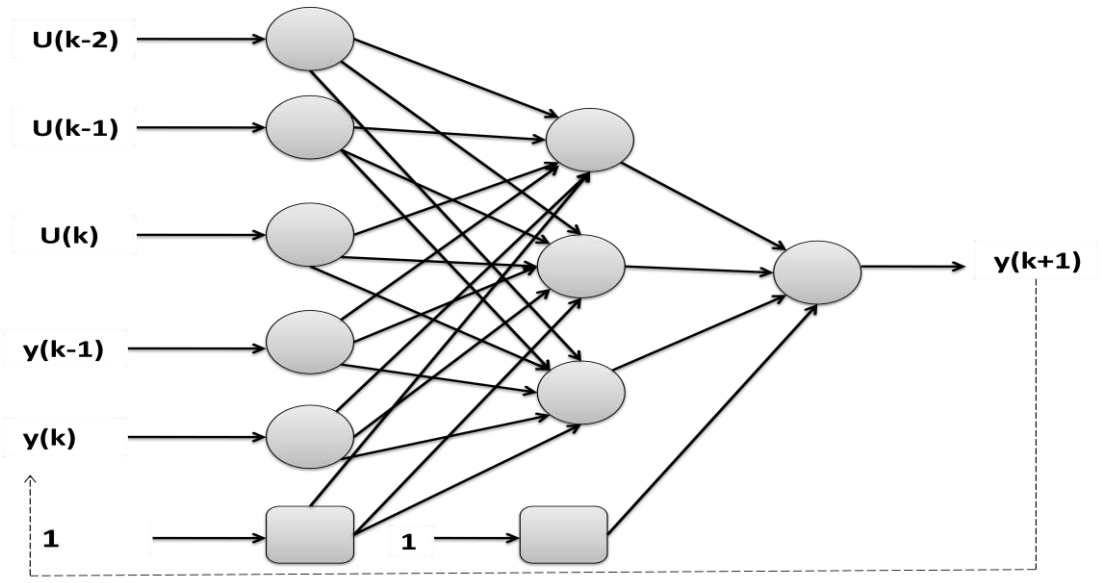


Figure II.4 Architecture de réseau du modèle.

II-8. Analyse par le modèle des réseaux de neurones

II-8.1. Evolution de l'erreur d'apprentissage

Dans la figure II.5 nous représentons les évolutions de l'erreur quadratique moyenne (EQM) calculée par le réseau par l'équation (II.4) et on l'a trace dans une fonction des cycles d'apprentissage.

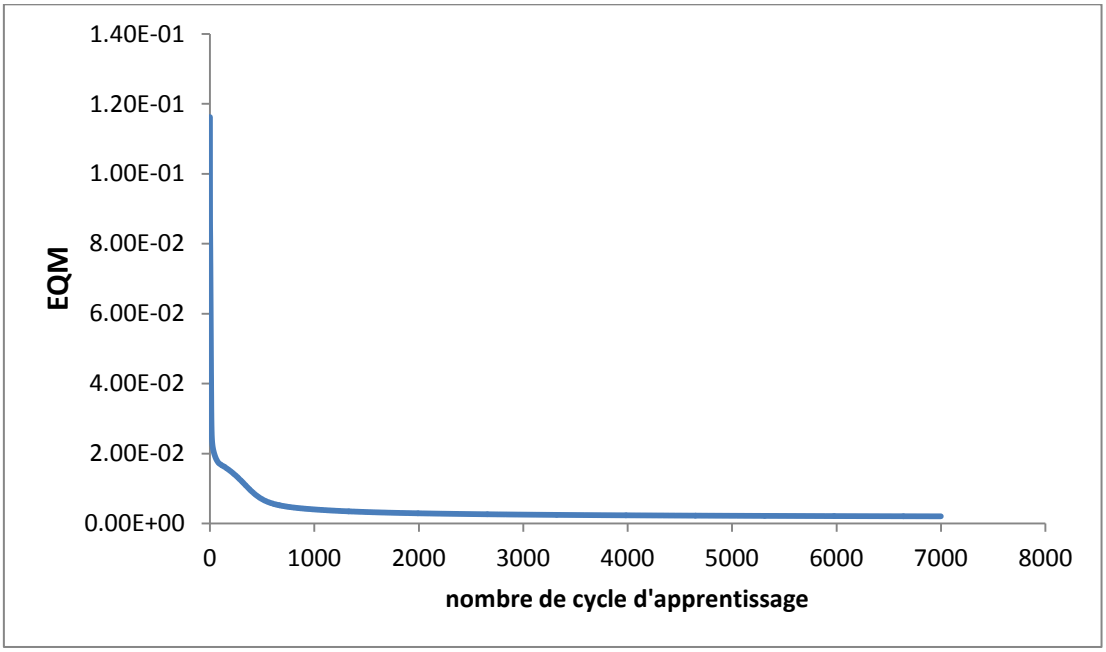


Figure II.5 Évolution de l'erreur quadratique moyenne d'apprentissage.

Commentaire

Nous remarquons que l'erreur quadratique tend à diminuer de façon continue, ce qui indique que les poids se rapprochent de leurs valeurs optimales qui minimisent l'écart entre la sortie du procédé et celle du réseau. Ceci indique que l'apprentissage du réseau se fait de façon correcte.

II-8.2.Phase de test

Nous avons généré une base de données constituée de 199 exemples. Dans un premier temps, nous considérons que la concentration de l'entrée change en fonction du temps selon la forme d'une fonction rompe, ensuite, nous utilisons l'équation discrète que l'on détermine à l'avance pour calculer la concentration exacte de sortie.

D'autre part, nous insérons les données d'entrée (les mêmes que ceux utilisés pour calculer la sortie exacte) dans le modèle développé. On calcul sa sortie puis on la trace en fonction du temps sur la même figure que celle de l'évolution de la sortie exacte. Les tracés se font tous en utilisant le logiciel Microsoft Excel comme le montre la figure II.6.Sur la même figure, nous avons tracé le signal imposé à l'entrée du réacteur.

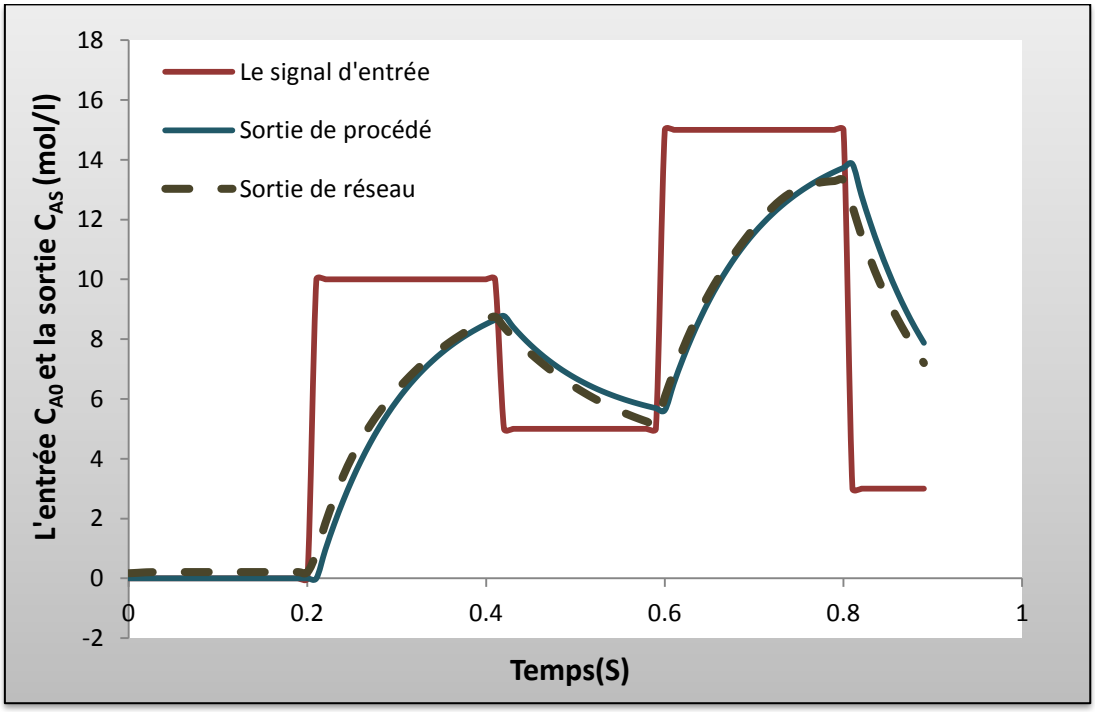


Figure II.6 Evolution de la sortie du procédé et celle du réseau.

Commentaire

On voit que les deux courbes représentant la sortie exacte et celle prédite par le modèle se rapprochent bien l'une de l'autre, ce qui signifie que le réseau simule bien le procédé. L'erreur entre les deux grandeurs est de l'ordre de 5.E-3.

II-8.3. Effet du nombre de neurone dans la couche cachée

La détermination du nombre optimal de neurones dans les couches cachés est nécessaire pour obtenir une approximation satisfaisante. Afin d'évaluer l'effet de ce paramètre, on procède à plusieurs simulations. Le signal d'entrée est identique à celui employé lors de la simulation type présentée précédemment.

On propose ainsi les architectures suivantes :

- **1er Essais**

• Nombre de neurone dans la couche d'entrée : 05

• nombre de neurone dans la couche caché : 03

• nombre de neurone dans la couche de sortie : 01

Résultat de 1er Essais

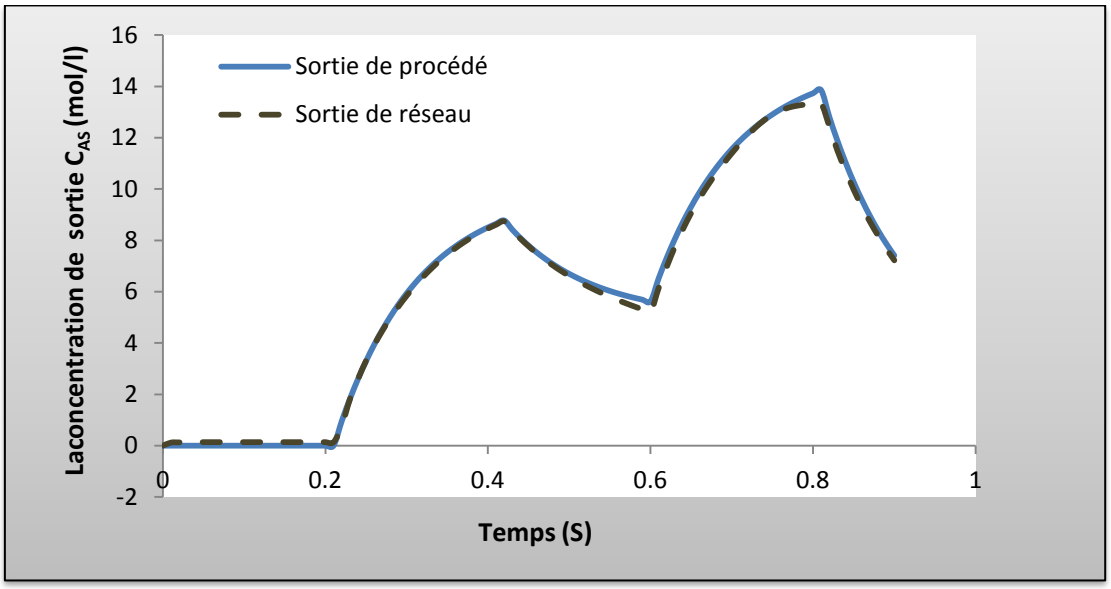


Figure II.7 Résultat de simulation du 1ere Essais

L'erreur quadratique moyenne : 2.043699879538E-03

- **2ème Essais.**

• Nombre de neurone dans la couche d'entrée : 05

• nombre de neurone dans la couche caché : 05

• nombre de neurone dans la couche de sortie : 01

Résultat de 2ème Essais

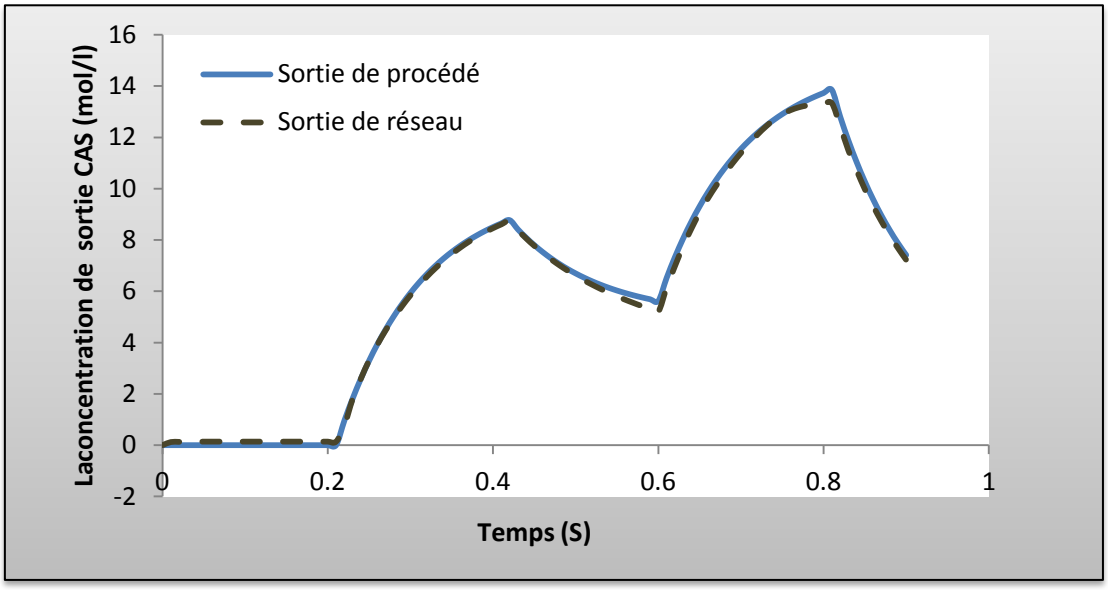


Figure II.8 Résultat de simulation du 2ème Essais.

L'erreur quadratique moyenne : 1.861690046449E-03

➢ **3ème Essais**

- Nombre de neurone dans la couche d'entrée : 05
- nombre de neurone dans la couche caché : 9
- nombre de neurone dans la couche de sortie : 01

Résultat de 3ème Essais

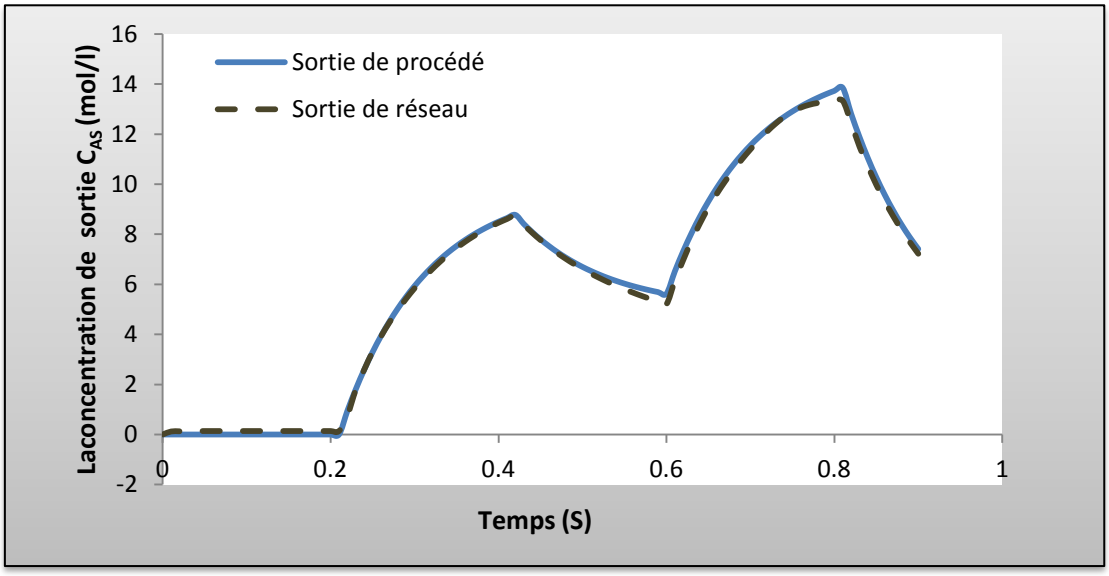


Figure II.9 Résultat de simulation du 3ème Essais.

L'erreur quadratique moyenne : 1.798911001428E -03

➢ **4ème Essais**

- Nombre de neurone dans la couche d'entrée : 05
- nombre de neurone dans la couche caché : 12
- nombre de neurone dans la couche de sortie : 01

Résultat de 4ème Essais

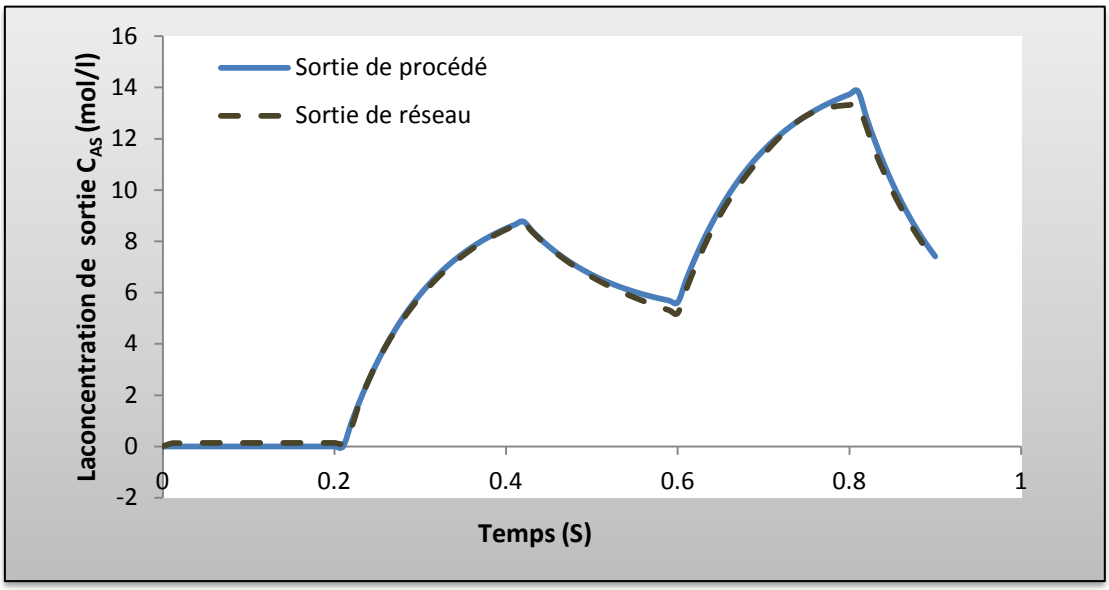


Figure II.10 Résultat de simulation du 4ème Essais.

L'erreur quadratique moyenne : 1.645914252206 E -03

Commentaire

Nous remarquons à travers les résultats que lorsque nous augmentons le nombre des neurones dans la couche cachée, l'erreur quadratique moyenne tend à diminuer. D'autre part, les courbes changent régulièrement et devient plus précis. Ceci nous permet de confirmer le fait que l'augmentation de nombre de neurone dans la couche cachée permet de mieux approximer le fonctionnement du procédé. Toutefois, la précision gagnée n'est pas très significative. Ainsi, dans notre cas, il apparait que 3 neurones dans la couche cachée soit un optimum.

II.8-4. Effet du nombre d'itérations

Parmi les paramètres qui influencent le fonctionnement des réseaux de neurones, on trouve le nombre d'itération dans la mesure où le processus d'apprentissage est un processus itératif. Comme il n'existe pas de règle générale pour l'estimation de ce paramètre, on procède par étapes pour trouver un optimum. Plusieurs simulations sont donc effectuées à différents nombre d'itérations et l'erreur de simulation est estimée à chaque fois. On garde pour ces simulations le même signal à l'entrée.

On fixe pour toutes les simulations l'architecture suivant :

- Nombre de neurone dans la couche d'entrée : 05
- nombre de neurone dans la couche caché : 03
- nombre de neurone dans la couche de sortie : 01

Cas I : pour 1000 itération.

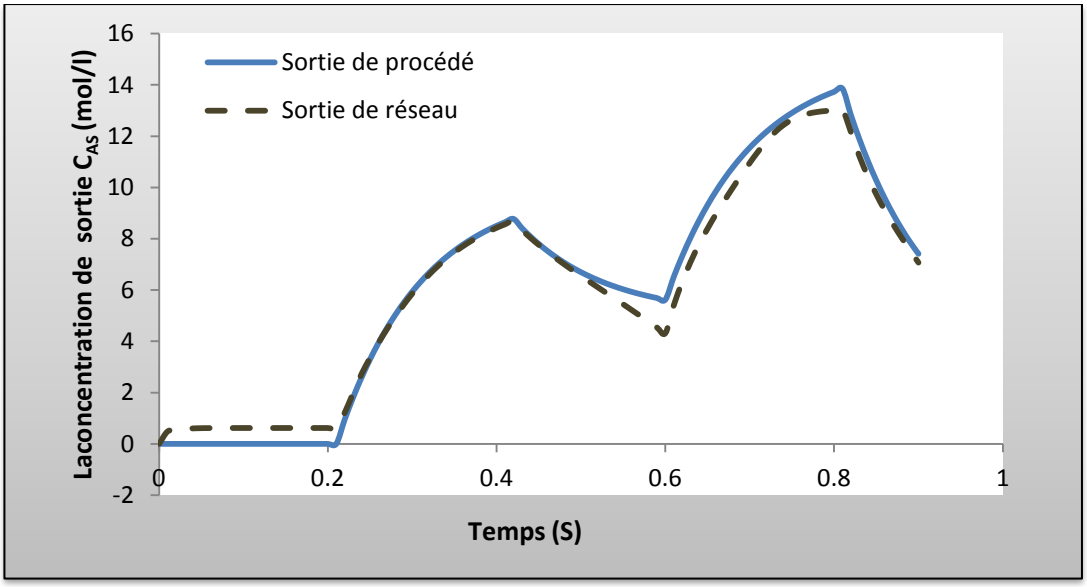


Figure II.11 Résultat de simulation pour 1000 itération.

Nous obtenu les même résultats pour l'itération 2000,3000,….jusqu'a 6000 itération.

- **Cas II :** Pour 7000 itération

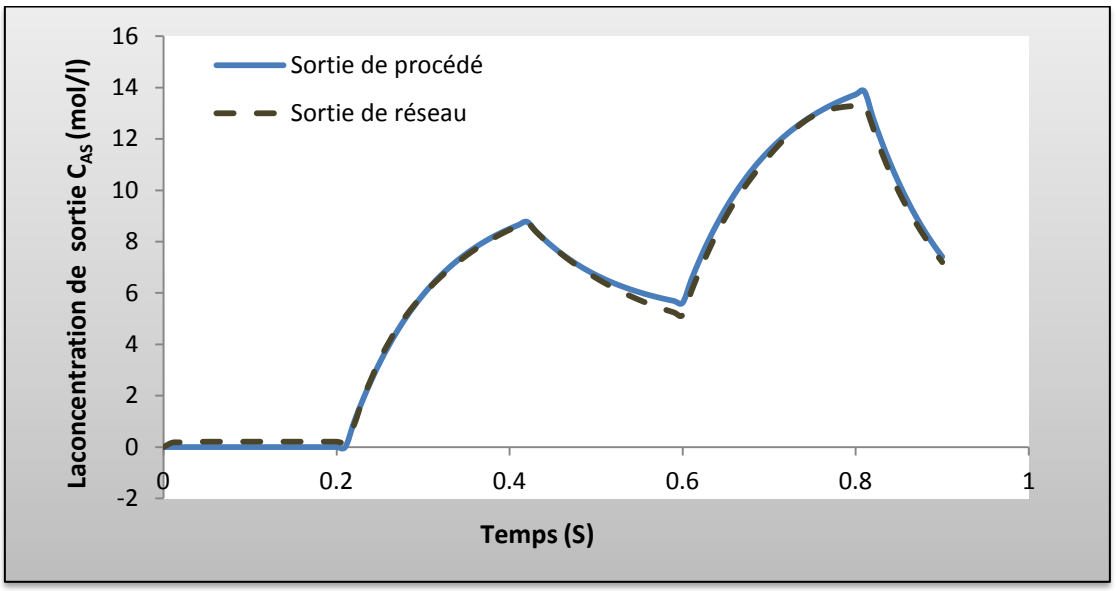


Figure II.12 Résultat de simulation pour 7000 itération.

On voit qu'un nombre d'itération de 1000 est suffisant pour effectuer les simulations car à 7000 itérations, l'erreur d'estimation de la sortie n'est pas très différente.

Conclusion

Dans ce chapitre nous avons présenté la modélisation dynamique d'un processus chimique, en l'occurrence un réacteur continu parfaitement agité où une réaction de premier ordre se déroule, à l'aide d'un réseau de neurones. Les résultats obtenus montrent que le modèle simule bien le procédé car les erreurs entre la sortie désirée et la sortie réelle du procédé est très faible. Une optimisation du nombre de neurones dans la couche cachée et du nombre d'itérations a été effectuée.

Conclusion générale

Ce travail est composé de deux parties :

La première partie, consacrée à l'étude des réseaux des neurones : leur fonctionnement, leur architecture, leur différent type…etc. a montré que les réseaux de neurones constituent un outil performent de modélisation de procédés chimiques. En deuxième lieu, nous avons identifié et modélisé un système dynamique. Cela vient après l'analyse et la compréhension de son fonctionnement. Cette identification qui est montrée par la figure I.7 a été faite pour un système chimique constitué d'un réacteur continu parfaitement agité soumis à une perturbation de l'entrée par un signal sous forme de rampe. L'idée est de trouver une équation algébrique issue du traitement de l'équation caractéristique du fonctionnement du réacteur, en l'occurrence une équation différentielle de premier ordre représentant le bilan de matière sur le réactif. Cette équation algébrique est ensuite employée pour la génération des sorties du procédé (exactes) pour différents signaux à l'entrée.

La seconde partie de ce travail était consacrée à l'application du modèle basé sur le réseau de neurones à la représentation du fonctionnement du réacteur. Les résultats des simulations effectuées ont montré qu'après l'apprentissage du réseau, la sortie calculée est très proche de la sortie réelle et les résultats obtenus sont satisfaisants. Ceci indique que ce type de modèle simule bien le processus chimique. Par ailleurs, nous avons étudié l'effet du changement de nombre de neurones dans la couche cachée ainsi que le changement du nombre d'itérations. Il est apparu clairement que plus le nombre de neurones augmente plus l'approximation et la précision sont meilleures.

Nos résultats obtenus montrent que le modèle simule bien le procédé car les erreurs entre la sortie désirée et la sortie réelle du procédé est très faible, Cela a été confirmé par la figure II.6 Ainsi, dans notre cas, il apparait que 3 neurones dans la couche cachée et 1000 itération soit un optimum.

Donc nous concluons que le réseau de neurones est un moyen approprié pour résoudre les problèmes très complexes qui n'ont pas trouvé de solutions en utilisant les méthodes traditionnelles. Ils ont amélioré et développé la modélisation et le traitement du signal ainsi que le contrôle des procédés chimique qui sont décrit par des systèmes d'équations non linéaires

Références bibliographique

[1] Corriou. J.P. , «Les réseaux de neurones pour la modélisation et la commande de procédées». Lavoisier Tec et Doc, 1995.

[2] Mohamed El Hossine. DAACHI . , «Identification paramétrique d'un système non linéaire par les méthodes classique et les réseaux de neurones». Mémoire de fin d'étude Tébessa, 2004-2005.

[**3] Zemouri.M.R**. , «Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques : Application à la e-maintenance ». Thèse de Doctorat. Université Franche-compté, 2003.

[4] Salah.Dounit. , «Commande prédictive (robuste) d'un procédé chimique ». mémoire DEA, Institut national polytechnique de Toulouse, 1996.

[5] Blayo. F, Verlysem. M. , « Réseaux de neurones artificiels ». 1er édition ,1996.

[6] Renard.J.-P. , «Réseaux neuronaux : une introduction accompagnée d'un modèle Java », Vuibert, 2001.

[7] Nabil.MESSIKH. , «Application des réseaux de neurones pour prédire le rendement de l'extraction liquide-liquide du phénol».Thèse de Doctorat, Université BADJI Mokhtar-Annaba ,2008.

[8] Amrani Mohamed. , **«** surveillance et diagnostic d'une ligne de production par les réseaux de neurones artificiels **»**. thèse de magister, université Mehamed Bougera Boumerdes (2001).

[9] Dan, W.P., «Artificial Neural Networks-Theory and Applications». Prentice Hall, USA, 1996.

[10] Victor.R.M.C. ,«Commande predictive adaptative des procédés chimiques ». Thèse de INP.Toulouse,1990.

[11] Christophe DOIGNON. , «Traitement du Signal ».Université Louis Pasteur de Strasbourg-France, 2008-2009.

[12] Ngac Ky Nguyen. , « Approches neuromimétiques pour l'identification et la commande des systèmes électriques : application au filtrage actif et aux actionneurs synchrones ». Cour, Engineering Sciences. Université de Haute Alsace - Mulhouse, 2010.

[13] Patrice Wira., «Approches neuromimétiques pour l'identification et la commande».Engineering Sciences. Université de Haute Alsace - Mulhouse, 2009.

[14] Marc Parizeau., «Le perceptron multicouche et son algorithme de rétropropagation des erreurs». Département de génie électrique et de génie informatique. Université Laval 10 septembre, 2004.

[15] Amira DJAMAI., « modélisation du procédé de la granulation en lit fluidisé par les réseaux neuro-mimétique». mémoire de fin d'étude université de Kasdi Marbah-Ouargla, 2016.

ملخص

تعتبر الخلايا العصبية المادية الداعم للذكاء الفيزيائي فهي تساعد على فهم ومعرفة كيفية استخدامه لتمكن من تحقيق أهداف لا يمكن تصورها. الشبكات العصبية الاصطناعية صنعت انطلاقا من الخلايا العصبية البيولوجية وهي مستوحاة من الجهاز العصبي, ويتم إعدادها لعمليات معالجة الإشارات والتحكم في العمليات, و تقوم بانجاز أسس الرقابة الذكية للأنظمة الفيزيائية المعقدة, الغير الخطية و الغير ثابتة. قدرة هذه الخلايا على التعلم تجعل هذه المهام والوظائف أكثر قوة واستقلال.

الهدف من دراستنا هي نمذجة نظام ديناميكي من قبل الشبكة العصبية حيث اخترنا لنمذجة السلوك الديناميكي مفاعل مستمر به تفاعل من الدرجة الأولى, بعد تحديد هذه العملية تم إنشاء قاعدتين الأولى للتعلم والثانية من اجل الاختبار. وأظهرت نتائج المحاكاة أن النموذج المطور يحاكي عملية مرضية جدا للمفاعل.

الكلمات المفتاحية :النمذجة ,الخلايا العصبية الاصطناعية , الخلايا العصبية المادية, الشبكات العصبية,التعلم.

Résume

Les neurones sont considérés comme le support de physique de l'intelligence. Ils fascinent car comprendre et savoir utiliser l'intelligence permet de réaliser des buts inimaginables. Les réseaux de neurone artificiels réalisé à partir de neurones formels sont ainsi inspirés du système nerveux, ils sont développés pour le traitement du signal et de contrôle des procédés.

L'objectif de notre étude est de modéliser un système dynamique par ce type de modèle. Nous avons choisi de modéliser le comportement dynamique d'un réacteur continu parfaitement agité dans lequel une réaction de premier ordre se déroule. Après identification du procédé, deux bases de données ont été générées : une pour l'apprentissage et une pour le test. Les résultats des simulations ont montré que le modèle développé simule de façon très satisfaisante le fonctionnement du réacteur.

Les mots clé: modélisation, neurones artificiels, neurone formel, réseaux, apprentissage.

Abstract

Neurons are considered as the physical medium of intelligence. They fascinate because understanding and knowing how to use intelligence makes it possible to achieve unimaginable goals. Artificial neural networks made from formal neurons are thus inspired by the nervous system, they are developed for signal processing and process control. The aim of our study is to model a dynamic system by this type of model. We have chosen to model the dynamic behavior of a perfectly stirred continuous reactor in which a first order reaction takes place. After identification of the process, two databases were generated: one for learning and one for testing. The simulation results showed that the model simulates the operation of the reactor very satisfactorily.

Key words: modeling, artificial neurons, formal neuron, networks, learning.