

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



Département de Génie Civil

Laboratoire de Construction et Environnement (LCE)

Mémoire de fin d'études

en vue de l'obtention du diplôme de Master en Génie Civil

Thème :

Utilisation des réseaux de neurones pour la
simulation du comportement des matériaux :
Etude d'ajouts carbonatés à un mélange ciment
Portland ciment Alumineux.

Réalisé par :
BENAMEUR Mohamed Yacine
RAHAL Moncef Razine

Encadré par :
Pr. BALI Abderrahim
Dr. SILHADI Kamel

Promotion Juillet 2012

Ecole Nationale Polytechnique 10, Avenue Pasteur Hassen Badi BP182 El-Harrach
16200 Alger (Algérie)

Tel : 213 21 52 53 01/03 – Fax :213 21 52 29 73

www.enp.edu.dz

ملخص

الهدف من هذا العمل هو تطبيق الشبكات العصبونية للمساعدة في تطوير إسمنت ذات أخذ قابل للتعديل ويتكون من خليط من الاسمنت البورتلاندي والاسمنت الألومني مع اضافات كربونات. الخليط يتشكل من إسمنت من منطقة الجزائر العاصمة (الرايس حميدو) مع نسبة 35% من الأسمنت الألوميني و تم ترقية الخلائط مع نسب مختلفة من اضافات كربونات. وتألّف عملنا من التحقق من انطباق محاكاة مقاومة الاسمنت باستخدام شبكات عصبونية "FeedForward" بفضل "مجموعة أدوات الشبكات العصبونية" من MATLAB® نسخة B11. البيانات التجريبية المستخدمة في مشروعنا هي نتيجة للعمل على تطوير الاسمنت ذات أخذ قابل للتعديل الذي أجري في عام 1997 في المعهد الوطني لمواد البناء ببومرداس من طرف أحمد مصطفى يادي. كلمات البحث : الشبكات العصبونية، MATLAB، المحاكاة الإسمنت، إسمنت ذات أخذ قابل للتعديل، إضافات كربونات.

Résumé

Le but de ce travail est la mise en application des réseaux neuronaux pour l'aide à l'étude d'ajouts carbonatés à un ciment à prise réglable constitué d'un mélange de ciments Portland et de ciment Alumineux.

Les mélanges sont constitués d'un ciment de la zone d'Alger (Rais Hamidou) auquel a été ajouté 35% de ciment alumineux. Ce mélange a été amélioré avec des ajouts carbonatés à différents taux.

Notre travail a consisté en la vérification de l'applicabilité puis en la simulation des résistances des ciments à l'aide de réseaux neuronaux FEEDFORWARD par l'utilisation du «neural network toolbox» du logiciel MATLAB® version 11B.

Les données expérimentales utilisées au cours de notre projet sont le fruit d'un travail portant sur l'élaboration de ciments à prise réglable réalisé en 1997 à l'Institut National des Matériaux de Construction de Boumerdes par Ahmed El Mostefa YADI.

Mots clefs : Réseaux de neurones, MATLAB, Simulation, Ciments, Ciment alumineux, Prise réglable, Ajouts carbonatés.

Abstract

The purpose of this work is the use of the neural networks for the preparation of an adjustable setting time cement, constituted of a mixture of Portland cement and High Alumina cement and carbonate admixtures.

The blended mixtures are constituted by a cement from Algiers region (Rais Hamidou), with 35% of alumina cement. Those blended cements contained different percentage of carbonate admixtures.

Our work consisted on checking of the applicability and afterward the simulation of the strengths of the cements with the use of FEEDFORWARD neural networks simulated with the neural network tool box of MATLAB®, version 11B

The experimental data used during our work come from a thesis about the preparation of an adjustable setting time cement, done in 1997 at "Institut National des Matériaux de Construction, Boumerdes" by Ahmed El Mostefa Yadi.

Key words: Neural Networks, MATLAB, Simulation, Cements, High aluminates cement, setting time cement, carbonate additions.

Remerciements

Nous adressons nos vifs remerciements à nos encadreurs le Professeur Abderrahim BALI et le Docteur Kamel SILHADI pour l'appui scientifique et pour leur présence durant la préparation de notre projet de fin d'études ainsi que pour toutes les connaissances qu'ils nous ont transmises durant ces trois années.

Nous tenons à remercier le Professeur BENYOUCEF du département de génie chimique de l'Ecole Nationale Polytechnique pour l'aide précieuse qu'il nous a apporté lors de nos recherches sur les réseaux de neurones, de même nous adressons nos plus chaleureux remerciements à Monsieur CHAID, enseignant à l'Université Mohamed Bouguerra de Boumerdes pour sa disponibilité.

Une mention spéciale à Monsieur Ahmed El Mostefa YADI qui nous a soutenus et prodigué son aide précieuse lors de l'utilisation de ses résultats.

Que les membres du jury trouvent ici l'expression de toute notre gratitude pour nous avoir fait l'honneur d'accepter l'examen de notre projet.

Il nous serait impardonnable d'oublier de remercier tous les enseignants de l'Ecole Nationale Polytechnique d'Alger et plus particulièrement nos enseignants de spécialité et notre chef de département le Dr.MITICHE-KETTAB pour tout le savoir qu'ils nous ont transmis, pour leur disponibilité et leurs efforts.

Que tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste travail, que tous ceux qui ont su nous soutenir durant sa préparation, que toutes ces personnes trouvent ici l'expression de nos plus chaleureux remerciements.

Dédicace :

À mes parents, en témoignage de ma reconnaissance et mon amour, eux qui m'ont tant appris abondamment donné et même trouvé un prénom original, ils sont et seront toujours un modèle à suivre.

À ma chère sœur Narimane qui reste une « petite » sœur adorable.

À ma Mima et mon tonton Mani que dieu leur prête longue vie.

À YasMine, Yacine, Tahar, Soumeya, Sofiane, Othmane, Nadjib, Meriem, Mehdi, Lyes, Lotfi, Les Walids, Les Nazims B et S, Houssam, Ghalib, Dolly, Boulma, Amine, Anis. Amis ou camarades d'un jour, certains surement compagnons de toujours, pour tous les bons moments passés et à venir.

À la mémoire de mon grand-père et de ma grand-mère qu'ils reposent en paix.

Je dédie ce mémoire

Moncef Razine

À mes très chers parents qui ont toujours été là pour moi, et qui m'ont donné un magnifique modèle de labeur et de persévérance. J'espère qu'ils trouveront dans ce travail toute ma reconnaissance et tout mon amour.

À ma chère sœur Nassima, et mon cher frère Abdelkrim.

À toute ma famille, et tout particulièrement Mam, Tonton Mourad, Tata Lila, Selma et Mehdi.

À Selma, Amine, Dhikra, Ghalib, Meriem, Moncef, Sarah, Soumeya, Walid et tous mes amis qui m'ont soutenu de près ou de loin.

À mes camarades Houssam, Lotfi, Mehdi, Nadjib, Nazim et Walid pour les bons moments passés au cours de ces trois dernières années.

Je dédie ce mémoire.

Yacine

Table des matières

<i>Table des matières</i>	i
<i>Liste des tableaux</i>	iv
<i>Liste des figures</i>	vi
<i>Introduction générale</i>	1
<i>Chapitre I : Les réseaux de neurones artificiels</i>	
IV.1. Introduction	3
IV.2. Généralités.....	3
IV.3. Principe de base.....	3
IV.4. Parcimonie.....	3
IV.5. Analogie biologique	4
IV.6. Composants d'un RNA	5
IV.6.1. Les poids synaptiques	5
IV.6.2. Les fonctions d'activation.....	6
IV.7. Apprentissage d'un RNA	6
IV.7.1. L'apprentissage supervisé.....	7
IV.7.2. L'apprentissage non supervisé.....	7
IV.7.3. Les algorithmes d'apprentissage.....	7
IV.7.3.1. La règle delta (apprentissage par correction d'erreur).....	7
IV.7.3.2. La règle de Hebb.....	8
IV.8. Les différents types de RNA	9
IV.8.1. Les réseaux de neurones non bouclés (Feedforward).....	9
IV.8.1.1. Le perceptron	10
IV.8.1.2. Les perceptrons multicouches (PMC)	10
IV.8.1.3. Les réseaux de neurones à fonctions radiales de base (RBF).....	11
IV.8.1.4. Comparaison entre RBF et PMC	12
IV.8.2. Les réseaux de neurones bouclés (Feedback)	12
IV.9. Conclusion	13

Chapitre II : Modélisation des mélanges aux ajouts

II.1.	Introduction.....	14
II.2.	Démarche de AEM YADI.....	14
II.3.	Modélisation des nouveaux mélange.....	15
II.3.1.1.	Démarche suivie.....	15
II.3.1.2.	Simulation des mélanges.....	16
II.3.1.2.a.	Simulation RH35 pur.....	16
II.3.1.2.b.	Simulation de l'ajout C1.....	18
II.3.1.2.c.	Simulation de l'ajout C2.....	20
II.3.1.2.d.	Simulation de l'ajout C3.....	22
II.3.1.2.e.	Simulation de l'ajout C4.....	24
II.3.1.2.f.	Simulation de l'ajout C5.....	26
II.3.1.2.g.	Simulation de l'ajout C6.....	28
II.4.	Interprétation des résultats et conclusion.....	30
	Conclusion générale	31
	Références bibliographiques	32

Liste des tableaux

Chapitre I : Les réseaux de neurones artificiels

Tableau I- 1 Tableau récapitulatif des diverses fonctions d'activation.....6
Tableau I- 2 Tableau récapitulatif des divers algorithmes d'apprentissage.....8
Tableau I- 3 Tableau récapitulatif des différents types de RNA et de leurs domaines
d'utilisation.....13

Chapitre II : Modélisation des mélanges aux ajouts

Tableau II - 1 Résistances à la compression des mélanges aux ajouts..... 15
Tableau II - 2 Proportions des ajouts carbonatés 15
Tableau II - 3 Valeurs de temps prises en compte pour la modélisation des mélanges aux
ajouts 15
Tableau II - 4 Architecture du réseau utilisé pour la modélisation du mélange RH35 pur..... 16
Tableau II - 5 Architecture du réseau utilisé pour la modélisation du mélange RH35 + C1 .. 18
Tableau II - 6 Architecture du réseau utilisé pour la modélisation du mélange RH35 + C2 .. 20
Tableau II - 7 Architecture du réseau utilisé pour la modélisation du mélange RH35 + C3 .. 22
Tableau II - 8 Architecture du réseau utilisé pour la modélisation du mélange RH35 + C4 .. 24
Tableau II - 9 Architecture du réseau utilisé pour la modélisation du mélange RH35 + C5 .. 26
Tableau II - 10 Architecture du réseau utilisé pour la modélisation du mélange RH35 + C6 28
Tableau II - 11 Valeurs du coefficient de corrélation de la simulation des mélanges aux ajouts
..... 30

Liste des figures

Chapitre I : Les réseaux de neurones artificiels

Figure I- 1 Les différents composants d'un neurone biologique.....	4
Figure I- 2 Analogie entre les neurones biologiques et artificiels.....	5
Figure I- 3 Allure générale d'un réseau de neurones artificiels.....	5
Figure I- 4 Allure générale d'un réseau de neurones de type Feedforward.....	10
Figure I- 5 Schéma représentatif d'un perceptron simple.....	10
Figure I- 6 Schéma représentatif d'un perceptron multicouche.....	11
Figure I- 7 Schéma représentatif d'un réseauRBF.....	12

Chapitre II : Modélisation des mélanges aux ajouts

Figure II - 1 Courbe de résistance à la compression du mélange RH35	17
Figure II - 2 Courbe de régression linéaire du mélange RH35	17
Figure II - 3 Courbe de résistance à la compression du mélange RH35 + C1	19
Figure II - 4 Courbe de régression linéaire du mélange RH35 + C1	19
Figure II - 5 Courbe de résistance à la compression du mélange RH35 + C2	21
Figure II - 6 Courbe de régression linéaire du mélange RH35 + C2	21
Figure II - 7 Courbe de résistance à la compression du mélange RH35 + C3	23
Figure II - 8 Courbe de régression linéaire du mélange RH35 + C3	23
Figure II - 9 Courbe de résistance à la compression du mélange RH35 + C4	25
Figure II - 10 Courbe de régression linéaire du mélange RH35 + C4	25
Figure II - 11 Courbe de résistance à la compression du mélange RH35 + C5	27
Figure II - 12 Courbe de régression linéaire du mélange RH35 + C5	27
Figure II - 13 Courbe de résistance à la compression du mélange RH35 + C6	29
Figure II - 14 Courbe de régression linéaire du mélange RH35 + C6	29

Introduction générale

Composant essentiel des bétons, les ciments sont les principaux liants utilisés dans le domaine de la construction. Leur fabrication et leur commercialisation qui sont sujettes à diverses inflations en font un matériau d'une importance stratégique.

Les ciments se subdivisent en plusieurs catégories en fonction de leur domaine d'utilisation et des principales caractéristiques recherchées lors de leur mise en œuvre. Mise à part les ciments classiques plusieurs procédés de fabrication de ciments sont développés afin d'en améliorer les performances en tenant compte des impératifs locaux.

Le processus de création de nouveaux ciments fait intervenir une multitude de paramètres et nécessite une batterie de tests et d'essais dont les résultats ne peuvent pas être retrouvés en utilisant des lois analytiques ; l'expérience y est donc essentielle bien que laborieuse et onéreuse.

Le développement de nouveaux outils mathématiques et la naissance de méthodes de calcul numérique basés sur l'intelligence artificielle et les sciences cognitives a permis l'émergence de méthodes de calculs et de modélisation telles que les réseaux de neurones artificiels (RNA).

Ceux-ci ont vite fait l'unanimité dans les différentes disciplines des sciences et de l'ingénierie grâce à leur efficacité, fiabilité et robustesse pour la résolution de problèmes complexes non descriptibles sous forme analytique. Ceci ouvre la voie à l'utilisation de processus de développement se basant à la fois sur une expérimentation ciblée et moins coûteuse couplée à un travail de simulation, qui peut prendre en compte des résultats de travaux antérieurs.

Dans l'optique de mettre en application l'outil neuronal dans les phases d'amélioration d'un ciment, nous nous sommes proposés de procéder à la modélisation du comportement de ciments à prise rapide constitué de mélange de ciment Portland (RAIS Hamidou) et du ciment alumineux à 35% auquel on a adjoint des ajouts carbonatés.

Notre travail de Master vient en complément de notre sujet de PFE qui a traité de l'étude des mélanges ciment Portland ciment Alumineux.

Aussi, notre travail s'est basé sur les recherches menées en 1997 par Ahmed El Mostefa YADI lors de son travail effectué dans le cadre de son mémoire de magister qui a traité de l'élaboration d'un ciment à prise rapide en élaborant des mélanges de ciments Portland et de ciment alumineux.

Notre travail de Master se compose de deux chapitres, un qui rappelle les principes de base des réseaux de neurones et le second en l'étude des ajouts carbonatés aux mélanges de ciment alumineux optimal trouvé par M.Yadi.

Ces chapitres viennent en continuité des cinq chapitres traités dans notre thèse d'ingénieur et dont le premier avait pour but de présenter succinctement les principes fondamentaux en ce qui concerne les caractéristiques principales des ciments ordinaires et leur processus d'hydratation.

En second lieu nous nous sommes rapidement intéressés aux ciments alumineux en mettant en évidence la spécificité de leur hydratation et leur complémentarité en mélange avec des ciments portland ordinaires, chose nécessaire pour pouvoir appréhender la démarche d'AEM YADI que nous avons explicité dans le troisième chapitre.

Après nous être focalisé sur la compréhension de la partie ciment, il nous était nécessaire au travers du quatrième chapitre de présenter les caractéristiques et le fonctionnement des réseaux de neurones qui constituent l'essence même de notre projet.

Enfin, après s'être imprégnés de la théorie, nous sommes passés dans le dernier chapitre à la mise en application de celle-ci, dans le but de montrer l'applicabilité des RNA dans la simulation des résistances des mélanges que l'on a continué dans le cadre du Master.

I.1. Introduction

Les réseaux de neurones artificiels (RNA) sont devenus en l'espace de quelques années des outils extrêmement précieux dans les différents domaines de l'industrie et de l'ingénierie.

Notre objectif à travers ce chapitre est de présenter les principales définitions et caractéristiques de ces outils. Nous précisons par la suite les divers éléments qui les composent et leurs rôles dans le fonctionnement d'un réseau de neurone.

Nous commencerons par quelques définitions et principes de bases pour ensuite nous intéresser aux différents composants et opérations des RNA. Enfin, nous passerons en revue leurs principaux types.

I.2. Généralités

Selon Gérard Dreyfus, un "neurone formel" (ou simplement "neurone") est une fonction algébrique non linéaire et bornée, dont la valeur dépend de paramètres appelés coefficients ou poids. Les variables de cette fonction sont habituellement appelées "entrées" du neurone, et la valeur de la fonction est appelée sa "sortie". [1]

Les réseaux de neurones artificiels sont donc des réseaux fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. [2]

I.3. Principe de base

Le principe de base des réseaux de neurones consiste à injecter une ou plusieurs variables qui composent le vecteur (ou matrice) d'entrée dans le but d'obtenir la modélisation d'une grandeur donnée.

Ce principe dit de « boîte noire » consiste à introduire des entrées et récolter des sorties, sans pour autant tenir compte des étapes intermédiaires. Il permet au réseau, à partir d'une base de données dite « d'apprentissage » et d'un groupe de valeurs ciblées, toutes deux connues, d'établir une relation entre ces entrées et la grandeur à modéliser.

La notion d'apprentissage sera étudiée en détail plus loin dans ce document.

A partir de là, un réseau qui a effectué un apprentissage de bonne facture sera en mesure de généraliser cette relation à des entrées jusque là jamais rencontrées. Cette propriété en fait un approximateur universel particulièrement intéressant, de par son caractère *parcimonieux*.

I.4. Parcimonie

Le nombre de données nécessaires pour ajuster un modèle mathématique dépend directement du nombre de ses paramètres ajustables ; ce nombre croît exponentiellement avec le nombre de variables lors de l'utilisation d'approximateurs linéaires par rapport à leurs paramètres, alors qu'il croît linéairement pour les approximateurs non linéaires par rapport à leur paramètres, ce qui est le cas des RNA.

Cette propriété nommée « parcimonie » caractérise les réseaux de neurones et fait qu'au cours de l'apprentissage, le nombre d'exemples nécessaires à l'ajustement des paramètres du réseau, pour calquer la relation qui existe entre les entrées et les sorties soit moindre.

Les réseaux de neurones artificiels présentent donc un avantage indéniable par rapport aux techniques usuelles d'approximation. En effet, dans le cadre d'une modélisation non linéaire, les RNA parviennent à réaliser des modèles de précision très appréciable avec moins de données expérimentales.

Ces données étant généralement synonymes de tests, expériences ou processus industriel dont la collecte peut être longue et coûteuse, l'intérêt de cet aspect parcimonieux apparaît alors clairement au travers de cette économie temporelle et financière.

1.5. Analogie biologique

Il est clair que l'idée même de réseaux neuronaux artificiels est inspirée du traitement d'informations qui a lieu au niveau des neurones biologiques.

Dans le cerveau humain, l'apprentissage s'effectue au travers d'un réseau de neurones où chaque neurone biologique est une cellule comportant des ramifications appelées *dendrites* qui permettent aux informations de véhiculer de l'extérieur vers le corps du neurone (*soma*). Cette information est alors traitée par le neurone et transféré au travers de l'*axone* au neurone suivant. La jonction entre deux neurones est appelée *synapse*. (Figure IV-1)

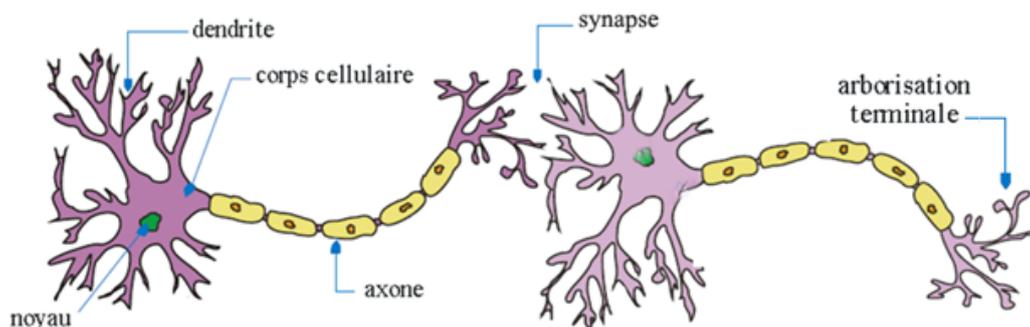


Figure I- 1 Les différents composants d'un neurone biologique

Une résistance synaptique variable affecte le flux d'informations entre deux neurones biologiques. Les dendrites, recueillent, via les synapses, les signaux électrochimiques ou les potentiels d'action venant d'autres cellules ou de l'extérieur. Si le potentiel somatique dépasse un certain seuil, il y a émission d'un potentiel d'action, qui correspond à une oscillation électrique très brève.

L'analogie avec le neurone artificiel est donc la suivante : il reçoit des données en entrée analogues aux impulsions électrochimiques reçues par les dendrites d'un neurone biologique. Ces entrées peuvent être les sorties d'autres neurones, ou des entrées de signaux extérieurs. La valeur de la sortie résulte du calcul de la somme des entrées, pondérées par des coefficients (dits poids synaptiques) et du calcul d'une fonction non linéaire bornée de cette somme pondérée. [3]

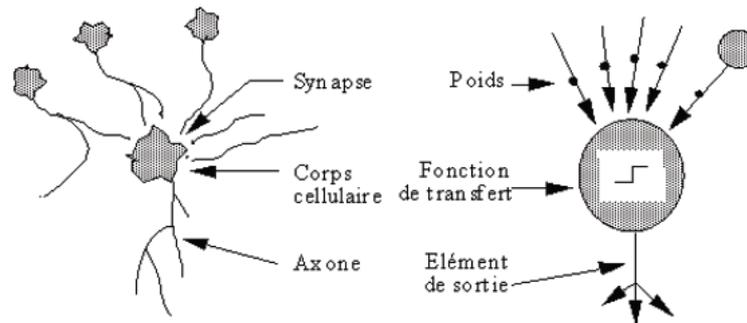


Figure I- 2 Analogie entre les neurones biologiques et artificiels

I.6. Composants d'un RNA

Un réseau de neurones artificiels typique se compose de manière générale d'une couche d'entrée, où sont injectées les valeurs d'entrée du réseau, d'une couche de sortie, qui donne les valeurs finales, ainsi qu'une ou plusieurs couches intermédiaires dites « couches cachées ».

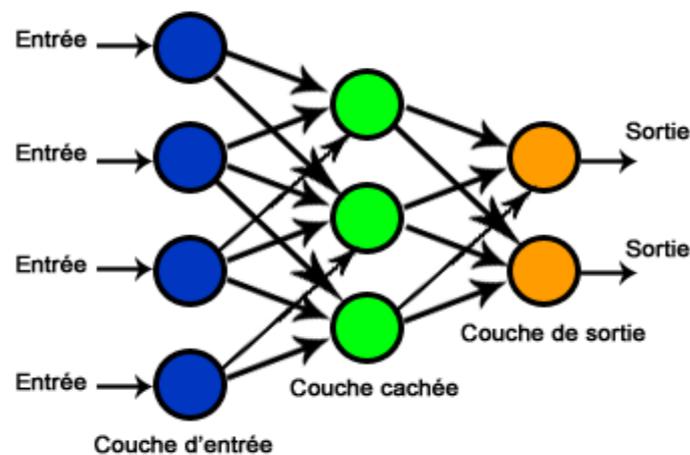


Figure I- 3 Allure générale d'un réseau de neurones artificiels

Au niveau de chaque connexion inter-neuronale, on retrouve les divers éléments constitutifs du réseau, tels que les poids synaptiques, les biais et les fonctions d'activation.

I.6.1. Les poids synaptiques

La notion de poids synaptiques implique le fait d'accroître la contribution de certains aspects d'un phénomène (ou d'une base de données) afin d'obtenir un résultat donné. Ceci peut être effectué en leur accordant « plus de poids » durant l'analyse, et ce au moyen de coefficients.

I.6.2. Les fonctions d'activation

La fonction d'activation (appelée également « fonction de seuillage », ou encore « fonction de transfert ») d'un neurone artificiel définit le rendement de ce neurone, donné à partir d'une entrée ou un ensemble d'entrées.

Diverses fonctions de transfert pouvant être utilisées comme fonction d'activation du neurone sont énumérées dans la figure ci-dessous. Les trois les plus utilisées étant les fonctions «seuil» (en anglais «hard limit»), «linéaire» et «sigmoïde». [4]

Tableau I- 1 Tableau récapitulatif des diverses fonctions d'activation

Nom de la fonction	Relation d'entrée/sortie	Icône	Nom Matlab
seuil	$a = 0$ si $n < 0$ $a = 1$ si $n \geq 0$		hardlim
seuil symétrique	$a = -1$ si $n < 0$ $a = 1$ si $n \geq 0$		hardlims
linéaire	$a = n$		purelin
linéaire saturée	$a = 0$ si $n < 0$ $a = n$ si $0 \leq n \leq 1$ $a = 1$ si $n > 1$		satlin
linéaire saturée symétrique	$a = -1$ si $n < -1$ $a = n$ si $-1 \leq n \leq 1$ $a = 1$ si $n > 1$		satlins
linéaire positive	$a = 0$ si $n < 0$ $a = n$ si $n \geq 0$		poslin
sigmoïde	$a = \frac{1}{1+\exp^{-n}}$		logsig
tangente hyperbolique	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		tansig
compétitive	$a = 1$ si n maximum $a = 0$ autrement		compet

I.7. Apprentissage d'un RNA

La phase d'apprentissage représente l'aspect le plus intéressant des réseaux de neurones artificiels. Bien qu'il existe des réseaux qui n'apprennent rien, ceux-ci ne présentent que très peu d'intérêt. En effet, ils se contentent de réaliser une combinaison d'informations provenant de plusieurs voies de traitement.

L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. [2]

Dans la plupart des algorithmes d'apprentissage utilisés, les paramètres modifiés sont les poids synaptiques des connexions (W_i). Le réglage de ces variables permet au réseau d'établir une relation entre ses entrées et ses sorties, en accordant les résultats du réseau aux valeurs cibles qu'on lui soumet (par exemple reconnaître une croix).

Cette opération peut être réalisée de deux façons. La première dite « apprentissage supervisé » consiste à effectuer une comparaison entre les valeurs de sortie (output) et celles ciblées

(target). La seconde, quant à elle appelée « apprentissage non supervisé » adapte la force de ses connexions en détectant les régularités statistiques des entrées.

Ces deux types d'apprentissage sont détaillés dans la partie suivante.

A l'issue de l'apprentissage, les poids sont fixés : on peut dès lors utiliser le réseau (il sait reconnaître une croix même si elle est bruitée ou détériorée).

I.7.1. L'apprentissage supervisé

Le principe de ce type d'apprentissage est simple et se présente sous deux phases.

La première est l'apprentissage proprement dit ; elle consiste à présenter au réseau de neurones un vecteur ou un ensemble de vecteurs d'entrée, en lui imposant d'autres vecteurs comme sortie ciblée (appelées valeurs cibles).

La seconde phase est la phase de test, pendant laquelle une nouvelle base de données est présentée au réseau. Ces valeurs n'ayant jamais été rencontrées auparavant, on pourra alors observer les performances du réseau et les évaluer par le pourcentage d'erreur.

L'intérêt des réseaux neuronaux réside précisément dans leur capacité intrinsèque à la généralisation, c'est-à-dire leur aptitude à bien se comporter sur des vecteurs qu'ils n'ont pas appris. [5]

I.7.2. L'apprentissage non supervisé

Il consiste à découper l'ensemble des vecteurs d'entrées en classes d'équivalence, sans qu'il soit nécessaire de donner au réseau neuronal les noms de classes pour chaque exemple. Cette classification automatique ressemble à certaines méthodes utilisées en analyse de données.

La séparation en classes d'équivalence s'opère par mesure de ressemblance entre les vecteurs proposés. La contrainte que l'on doit imposer au réseau neuronal est le nombre de classes d'équivalence. Pendant la phase d'apprentissage, le réseau neuronal construit une topologie de l'espace des vecteurs d'entrées. [25]

I.7.3. Les algorithmes d'apprentissage

I.7.3.1. La règle delta (apprentissage par correction d'erreur)

La correction d'erreur désigne la correction de l'erreur observée en sortie. L'apprentissage par correction des erreurs consiste à minimiser un indice de performance E basé sur les signaux d'erreur $e(\tau)_i$ qui est l'erreur entre ce qu'on obtient $y_k(x)$ et ce qu'on voudrait obtenir t_k , dans le but de faire converger les sorties du réseau avec ce qu'on voudrait qu'elles soient. Un critère très usuel est la somme des erreurs quadratiques E . Il faut changer les poids de réseau dans une direction qui diminue E , c'est-à-dire dans le sens opposé au gradient. On parle alors d'une direction de « descente » donnée par :

$$\Delta \mathbf{w}^{(\tau)} = -\eta \nabla E^{(\tau)} \quad (1)$$

Où η est appelée le taux d'apprentissage et $\eta \nabla E^{(\tau)}$ désigne le gradient de E par rapport à ces paramètres libres (les poids w) au temps τ , la règle (1) est dite de «descente du gradient». [6]

I.7.3.2. La règle de Hebb

On peut formuler l'énoncé de Hebb sous la forme d'une règle d'apprentissage donnée par

$$\Delta w_j^{(\tau-1)} = \eta p_j^{(\tau)} a^{(\tau)} \quad (2)$$

Où η est une constante positive qui détermine la vitesse de l'apprentissage, $p_j^{(\tau)}$ correspond à l'activité pré-synaptique (l'entrée j du neurone) au temps τ , et $a^{(\tau)}$ à l'activité post-synaptique (sortie du neurone) à ce même temps τ . [6]

Le tableau IV-2 résume les principales caractéristiques des algorithmes d'apprentissage de rétro propagation implantés dans MATLAB [7]

Tableau I- 2 Tableau récapitulatif des divers algorithmes d'apprentissage

Algorithme	Caractéristiques
Descente du gradient	
Descente du gradient simple (<i>traingd</i>)	Descente du gradient de base. Réponse lente; peut être utilisée pour apprentissage en mode séquentiel
Descente du gradient avec moment (<i>traingdm</i>)	Descente du gradient avec moment. Le moment permet à un réseau de répondre non seulement au gradient local, mais aussi aux tendances récentes dans la surface d'erreur. Sans moment un réseau peut être coincé dans un minimum local peu profond. Plus rapide que <i>traingd</i> , ne peut être utilisée que pour apprentissage en mode séquentiel
Descente du gradient avec taux d'apprentissage variable (<i>traingda</i>)	Taux d'apprentissage variable. Plus rapide que <i>traingd</i> , mais ne peut être utilisée que pour apprentissage en mode batch. Utilise des règles heuristiques pour la variation du taux d'apprentissage
Descente du gradient avec moment et taux d'apprentissage variable (<i>traingdx</i>)	Descente du gradient avec inertie (momentum) et taux d'apprentissage variable
Descente des gradients conjugués	
Descente des gradients conjugués avec la méthode de mise à jour de Fletcher-Reeves (<i>traingcf</i>)	Il possède la plus petite mémoire de stockage parmi les algorithmes de gradient conjugués.
Descente des gradients conjugués avec la méthode de mise à jour de Polak-Ribière (<i>traingcp</i>)	Nécessite une mémoire de stockage légèrement plus grande que <i>traingcf</i> . Convergence plus rapide sur quelques problèmes.
Descente des gradients conjugués avec la méthode de reprises Powell-Beale (<i>traingcb</i>)	Nécessite une mémoire de stockage légèrement plus grande que <i>traingcp</i> . Convergence Généralement plus rapide.
Scaled conjugate gradient (<i>traingcg</i>)	Le seul algorithme de gradients conjugués qui n'exige aucune recherche de ligne. Très bon algorithme d'apprentissage en général.
Quasi-Newton	

BFGS quasi-Newton (<i>trainbfg</i>)	Il exige le stockage d'une approximation de la matrice Hessienne et nécessite plus de calcul à chaque itération que les algorithmes de gradients conjugués, mais possède habituellement une convergence plus rapide.
Sécante à un pas (<i>trainoss</i>)	Cet algorithme exige moins de stockage et calcul par époque que l'algorithme BFGS. On peut considérer cela un compromis entre des algorithmes de quasi-Newton complets et des algorithmes de gradients conjugués.
Levenberg-Marquardt	
Levenberg-Marquardt (<i>trainlm</i>)	C'est l'algorithme d'apprentissage le plus rapide pour les réseaux de taille modérée. Il a la caractéristique de réduction de mémoire pour l'utilisation quand l'échantillon d'apprentissage est grand.
Régularisation	
Régularisation Bayésienne (<i>trainbr</i>)	Mise à jour des poids et des biais selon l'optimisation de Levenberg-Marquardt. Il réduit au minimum une combinaison d'erreurs quadratiques et des poids et détermine ensuite la combinaison correcte afin de produire un réseau avec une capacité de généralisation meilleure. Ce processus est appelé la régularisation Bayésienne. Il réduit la difficulté de déterminer l'architecture de réseau optimum.

I.8. Les différents types de RNA

Si l'on désire classifier les réseaux de neurones, on peut, dans un premier temps les regrouper en deux grandes familles, à savoir les réseaux de neurones *non bouclés* et les réseaux de neurones *bouclés*.

I.8.1. Les réseaux de neurones non bouclés (Feedforward)

Un réseau de neurones non bouclé réalise une (ou plusieurs) fonctions algébriques de ses entrées, par composition des fonctions réalisées par chacun de ses neurones. [8]

Ces réseaux dits « feedforward » sont représentés graphiquement par un ensemble de neurones "connectés" entre eux, l'information circulant des entrées vers les sorties sans "retour en arrière".

Si l'on représente le réseau comme un graphe dont les nœuds sont les neurones et les arêtes les "connexions" entre ceux-ci, le graphe d'un réseau non bouclé est *acyclique*. Le terme de "connexions" est une métaphore : dans la très grande majorité des applications, les réseaux de neurones sont des formules algébriques dont les valeurs numériques sont calculées par des programmes d'ordinateurs, non des objets physiques (circuits électroniques spécialisés) ; néanmoins, le terme de connexion, issu des origines biologiques des réseaux de neurones, est passé dans l'usage, car il est commode quoique trompeur. [8]

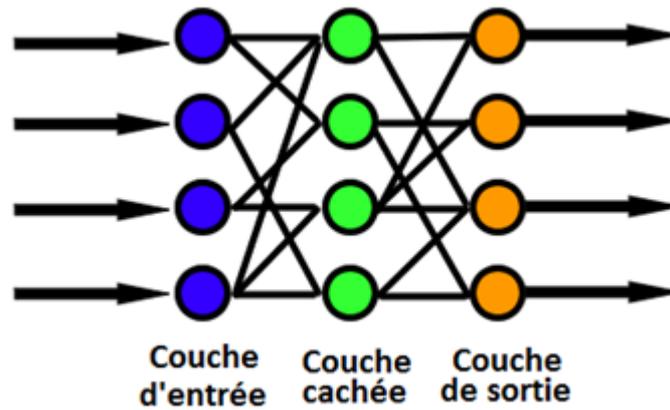


Figure I- 4 Allure générale d'un réseau de neurones de type Feedforward

1.8.1.1. Le perceptron

Il est possible de définir le perceptron comme étant le type de réseau de neurones le plus simple, il s'agit d'un classifieur linéaire composé uniquement d'une couche d'entrée et d'une couche de sortie.

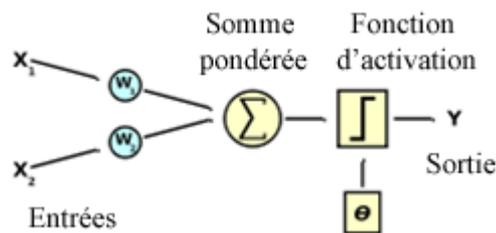


Figure I- 5 Schéma représentatif d'un perceptron simple

1.8.1.2. Les perceptrons multicouches (PMC)

Le perceptron multicouche est un réseau orienté de neurones artificiels organisé en couches où l'information voyage dans un seul sens, de la couche d'entrée vers la couche de sortie.

La figure IV-6 donne l'exemple d'un réseau contenant une couche d'entrée, deux couches cachées et une couche de sortie. La couche d'entrée représente toujours une couche virtuelle associée aux entrées du système. Elle ne contient aucun neurone. Les couches suivantes sont des couches de neurones. Dans l'exemple illustré, il y a trois entrées, trois neurones sur la première couche cachée, quatre neurones sur la deuxième et deux neurones sur la couche de sortie. Les sorties des neurones de la dernière couche correspondent toujours aux sorties du système.

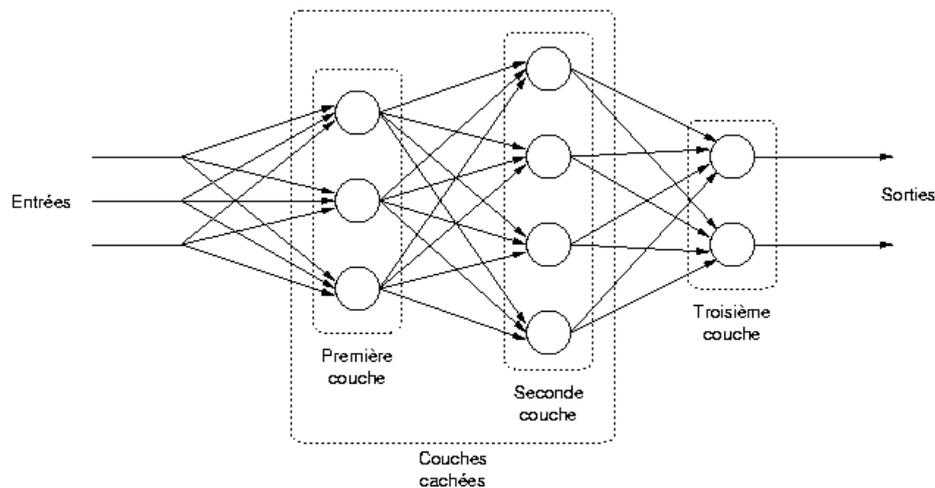


Figure I- 6 Schéma représentatif d'un perceptron multicouche

Dans le cas général, un perceptron multicouche peut posséder un nombre de couches quelconque et un nombre de neurones (ou d'entrées) par couche également quelconque.

Les neurones sont reliés entre eux par des connexions pondérées. Ce sont les poids de ces connexions qui gouvernent le fonctionnement du réseau et « programment » une application de l'espace des entrées vers l'espace des sorties à l'aide d'une transformation non linéaire.

La création d'un perceptron multicouche pour résoudre un problème donné passe donc par l'inférence de la meilleure application possible telle que définie par un ensemble de données d'apprentissage constituées de paires de vecteurs d'entrées et de sorties désirées. Cette inférence peut se faire, entre autre, par l'algorithme dit de rétropropagation. [9]

Les fonctions d'activation utilisées pour ce type de réseau sont principalement des fonctions sigmoïdes.

1.8.1.3. Les réseaux de neurones à fonctions radiales de base (RBF)

L'idée générale des réseaux RBF dérive de la théorie d'approximation des fonctions, ces réseaux sont une architecture Feedforward puissante. La raison de son application vient du fait que le réseau utilise des fonctions gaussiennes standard qui sont à symétrie radiale. Son apprentissage est basé sur l'algorithme K-means et l'algorithme des moindres carrés. [9]

Les réseaux de neurones RBFs, sont principalement utilisés pour résoudre des problèmes d'approximation de fonctions dans des espaces de grandes dimensions.

Pour des raisons de simplicité, nous avons décidé de faire une petite dualité entre le réseau RBF et le PMC, en précisant les ressemblances et les différences entre les deux types des réseaux. Ce choix est justifié par la popularité des PMCs et leur vaste utilisation dans les applications industrielles. Un réseau de neurone de type RBF est un PMC spécial, son architecture est identique à celle d'un PMC à une seule couche cachée (Figure IV-7).

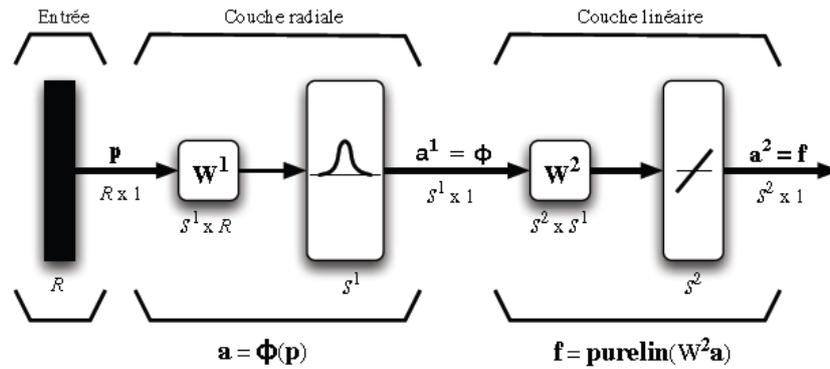


Figure I- 7 Schéma représentatif d'un réseau RBF

1.8.1.4. Comparaison entre RBF et PMC

Différences :

- **Le nombre de couches cachées :** Fixé à un seul dans le cas des réseaux RBF, ce nombre peut varier selon les cas étudiés avec l'utilisation des PMC, afin d'obtenir plus de précision.
- **La fonction d'activation :** Le réseau RBF utilise toujours une fonction dite à base radiale (fonction gaussienne).
- **Les poids synaptiques :** Les poids entre la couche d'entrée et la couche cachée dans les modèles neuronaux de type RBF sont toujours d'une valeur d'unité, c'est-à-dire que l'information inscrite sur la couche d'entrée sera retransmise sans distorsion vers les neurones de la couche cachée.

Similitudes :

- **La fonction de sortie :** Généralement une simple fonction linéaire qui renvoie une sommation pondérée des valeurs calculées par les neurones de la couche cachée. Bien sur, ce n'est pas toujours le cas, parfois l'utilisation d'autres fonctions pourrait être plus adéquate dans un problème donné.
- **Le sens des connexions :** Les deux réseaux sont de type Feedforward, les connexions suivent le même sens et ne sont pas récurrentes. Chaque neurone est entièrement connecté aux neurones de la couche suivante.
- **L'apprentissage :** L'apprentissage supervisé est utilisé aussi bien pour les RBF que pour les PMC pour le calcul des poids.

1.8.2. Les réseaux de neurones bouclés (Feedback)

Un réseau de neurone est donc dit bouclé ou récurrent lorsque des sorties de neurones d'une couche sont connectées aux entrées d'une couche d'indice inférieur. Autrement dit, lorsqu'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.

La sortie d'un neurone du réseau peut donc être fonction d'elle-même ; cela n'est évidemment concevable que si la notion de temps est explicitement prise en considération. Ainsi, à chaque connexion d'un réseau de neurones bouclé est attachée, outre un poids comme pour les réseaux non bouclé, un retard, multiple entier de l'unité de temps choisie. [3]

Les réseaux de neurones bouclés les plus couramment utilisés sont les cartes auto-organisatrices de Kohonen et les mémoires associatives de Hopfield.

Ces deux types de réseaux sont utilisés respectivement pour le codage d'informations et l'optimisation d'images.

Le tableau IV-3 récapitule les différents types de réseaux de neurones ainsi que leurs domaines d'utilisations respectifs :

Tableau I- 3 Tableau récapitulatif des différents types de RNA et de leurs domaines d'utilisation

	Types de réseaux	Domaine d'application
Feed-Forward	Perceptron monocouche	<ul style="list-style-type: none"> • Classification linéaire.
	PMC	<ul style="list-style-type: none"> • Classification non linéaire. • Approximation de fonction. • Contrôle. • Reconnaissance de formes et de paroles.
	RBF	<ul style="list-style-type: none"> • Classification non linéaire. • Approximation de fonction. • Contrôle. • Reconnaissance de formes et de paroles.
Feed-Back	Cartes auto-organisatrices	<ul style="list-style-type: none"> • Projection non linéaire des données. • Compression des données.
	Mémoires associatives	<ul style="list-style-type: none"> • Problèmes d'optimisation. • Reconnaissance de formes et de paroles.

I.9. Conclusion :

Utilisables pour résoudre une large gamme de problèmes d'ingénierie, les réseaux de neurones s'imposent comme des outils performants caractérisés par la non nécessité d'avoir des relations théoriques entre les différentes variables. En outre grâce à ce principe de boîte noire, on peut réaliser des approximations de précisions accrue pour peu que l'on ait à notre disposition une base d'apprentissage de taille et de pertinence adéquates.

Au travers cette initiation à l'utilisation des réseaux de neurones, il nous est apparu l'adéquation de la méthode avec notre problème. Les notions théoriques acquises au cours de ce chapitre nous ont donné les outils suffisants pour choisir l'architecture à utiliser lors de la modélisation.

CHAPITRE II

Modélisation des mélanges aux ajouts

II.1. Introduction

Le mélange choisi au terme de la phase de sélection montrant tout de mêmes certaines lacunes, il fut sujet à certaines améliorations.

Nous commencerons donc ce chapitre en rappelant la démarche suivie par M.YADI dans son travail de magister [10], en précisant les valeurs obtenues à la suite des essais qu'il avait suivi.

Par la suite, nous procéderons à la partie pratique où l'on effectuera la modélisation de ses résultats à l'aide de réseaux de neurones.

II.2. Démarche de AEM YADI

La deuxième partie du travail de M. YADI consistait en l'amélioration du mélange qu'il avait jugé comme optimal au terme de la phase de sélection, c'est-à-dire le mélange du ciment de Rais-Hamidou à 35% de ciment alumineux. Cette amélioration concerne les résistances à très court et long terme que l'auteur s'est proposé d'augmenter, notamment vis-à-vis du phénomène de conversion du ciment alumineux abordé dans le troisième chapitre de ce document.

Pour se faire, M.YADI a procédé à l'élaboration de nouveaux mélanges contenant du ciment résultant du mélange RH35, ainsi que des ajouts carbonatés, et ce à différents dosages allant de 2.5 à 5%. [10]

Son choix s'est basé sur divers travaux traitant de la réaction entre les aluminates hydratés et le carbonate de calcium, et qui montrent une atténuation de la perte de résistance mécanique due au phénomène de conversion. [11, 12, 13, 14, 15]

Les essais de résistance à la compression ont été réalisés sur les différentes compositions, et ce jusqu'à 120 jours. Les résultats obtenus figurent dans le Tableau II-1.

Nous utiliserons ces mêmes données pour réaliser nos modélisations.

Compositions Temps	RH35	C1	C2	C3	C4	C5	C6
18'	3.9	4.5	4.8	4.9	5.0	5.0	5.6
30'	6.0	7.5	7.8	7.4	10.0	10.2	10.4
1h	7.2	8.0	8.5	12.0	19.0	18.8	20.5
2h	9.8	10.0	10.5	12.5	20.8	20.0	21.6
3h	11.4	14.0	14.0	21.0	24.0	20.4	22.5
6h	18.1	20.0	21.0	24.0	28.0	24.8	28.0
1j	25.1	25.0	24.0	25.2	30.0	30.0	32.0
1.5j	27.0	28.0	28.0	26.0	31.2	30.4	32.0
2j	27.5	28.0	28.0	28.0	34.0	31.0	38.8
3j	28.1	28.0	30.0	34.0	40.0	45.0	41.5
7j	29.9	31.0	34.6	36.0	51.2	50.8	51.6
28j	31.0	43.2	39.0	58.0	60.0	57.6	57.6
90j	36.4	43.5	45.2	52.0	55.2	56.4	59.0
120j	37.0	45.6	48.3	57.8	65.1	64.0	66.0

Tableau II - 1 Résistances à la compression des mélanges aux ajouts

II.3. Modélisation des nouveaux mélanges

Le but de cette partie est donc de simuler le comportement de nouveaux mélanges réalisés à partir du mélange du ciment Rais-Hamidou à 35% de ciment alumineux (RH35) auquel on ajoute différentes proportions d'ajout carbonaté, dans le but d'améliorer ses performances initiales. Ces proportions sont exprimées en pourcentage de la masse totale du mélange de ciment dans le Tableau II-2 ci-dessous :

C1	C2	C3	C4	C5	C6
2.5%	3%	3.5%	4%	4.5%	5%

Tableau II - 2 Proportions des ajouts carbonatés

II.3.1. Démarche suivie

Nous avons procédé de la même manière que lors de la simulation des différents mélanges des ciments RH et MFT. Le taux de ciment alumineux étant cette fois-ci constant, la base d'apprentissage se compose donc des proportions d'ajout carbonaté et des valeurs de temps correspondant aux mesures de résistance. Les valeurs cibles sont quant à elles les valeurs de la résistance à la compression mesurées aux temps t.

L'intervalle de temps s'étend de 18 minutes à 120 jours, les valeurs sont réparties comme suit :

Temps (h)													
0.3	0.5	1	2	3	6	24	36	48	72	168	672	2160	2880

Tableau II - 3 Valeurs de temps prises en compte pour la modélisation des mélanges aux ajouts

Nous avons tour à tour retiré de la base d'apprentissage chacun des mélanges (suivant le pourcentage d'ajout) pour ensuite le simuler grâce à un réseau de neurones, présentant donc deux entrées (proportions d'ajouts et temps) et une sortie unique (résistance à la compression).

Compte tenu de la similitude des données à simuler et des bons résultats de simulation obtenus précédemment, nous avons opté pour la même architecture de réseau.

II.3.2. Simulation des mélanges

II.3.2.a. Simulation RH35 pur

Nous effectuons ici la simulation du mélange de ciment Rais Hamidou sans ajout carbonaté. De ce fait, la base d'apprentissage sera constituée en entrée de tous les dosages en ajouts C1, C2, C3, C4, C5, C6 avec les temps associés (Tableau II-3), et en cibles les résistances à la compression associées.

En utilisant la même fonction, nous trouvons que le réseau optimal est celui constitué de 2 neurones pour la première couche, et 8 pour la seconde.

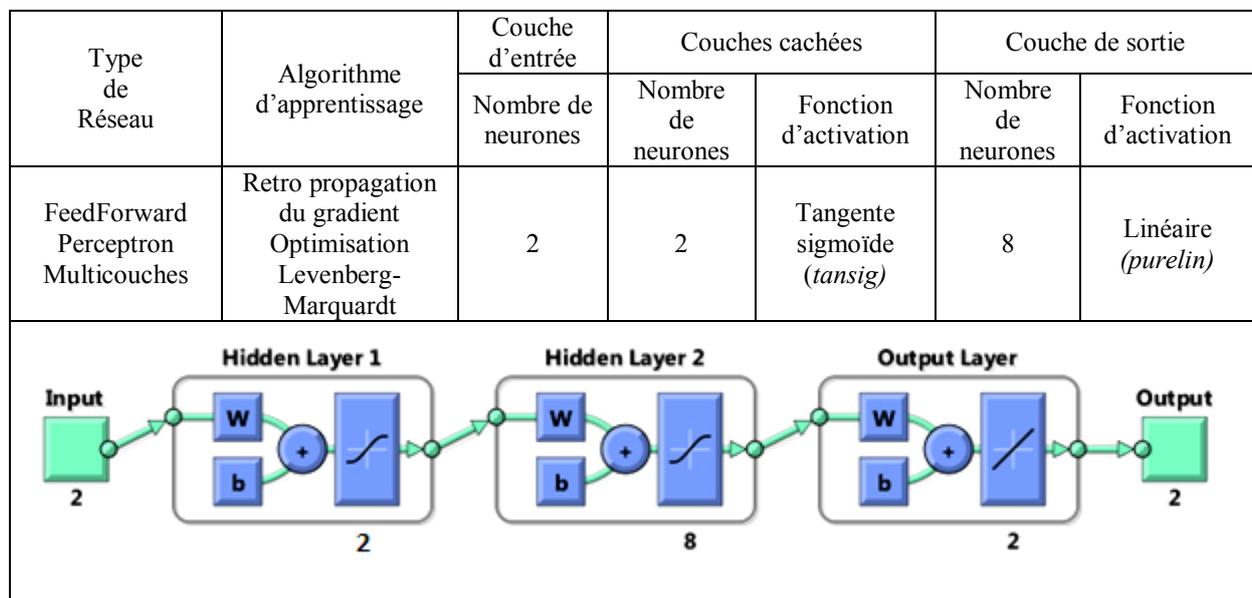


Tableau II - 4 Architecture du réseau utilisé pour la modélisation du mélange RH35 pur

Les figures VI-1 et VI-2 représentent respectivement la sortie (résistances à la compression) simulée avec le réseau de neurones comparée aux valeurs expérimentales de M.YADI, ainsi que la représentation de la corrélation sortie/cible avec la valeur du coefficient de corrélation R.

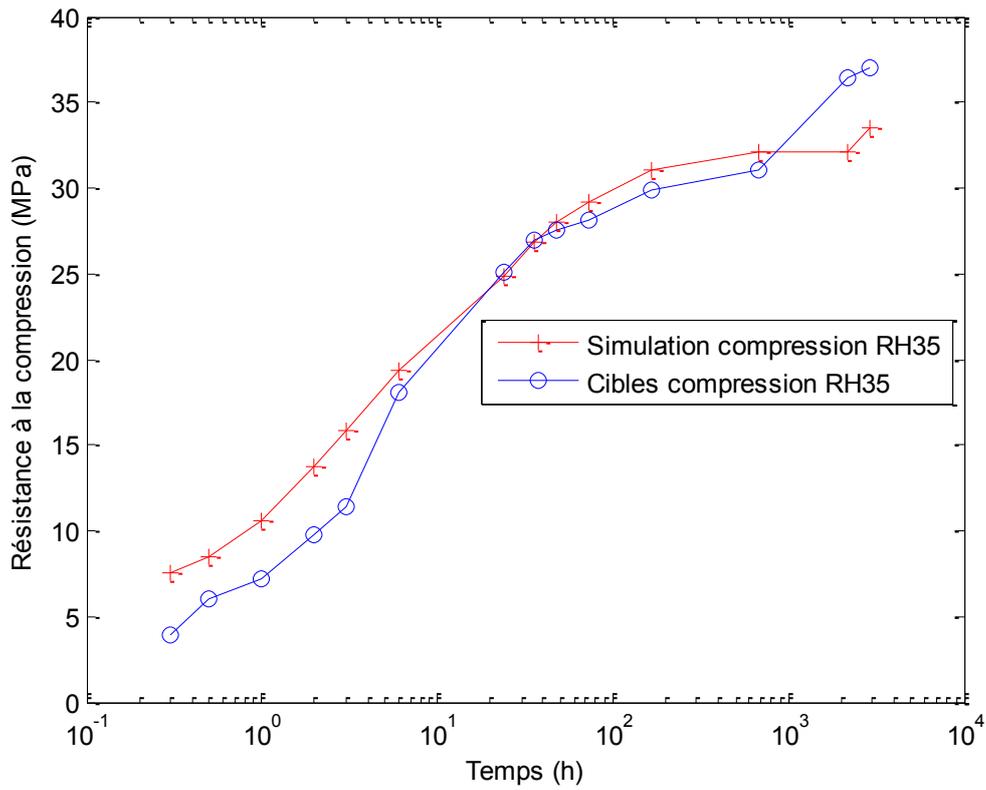


Figure II - 1 Courbe de résistance à la compression du mélange RH35

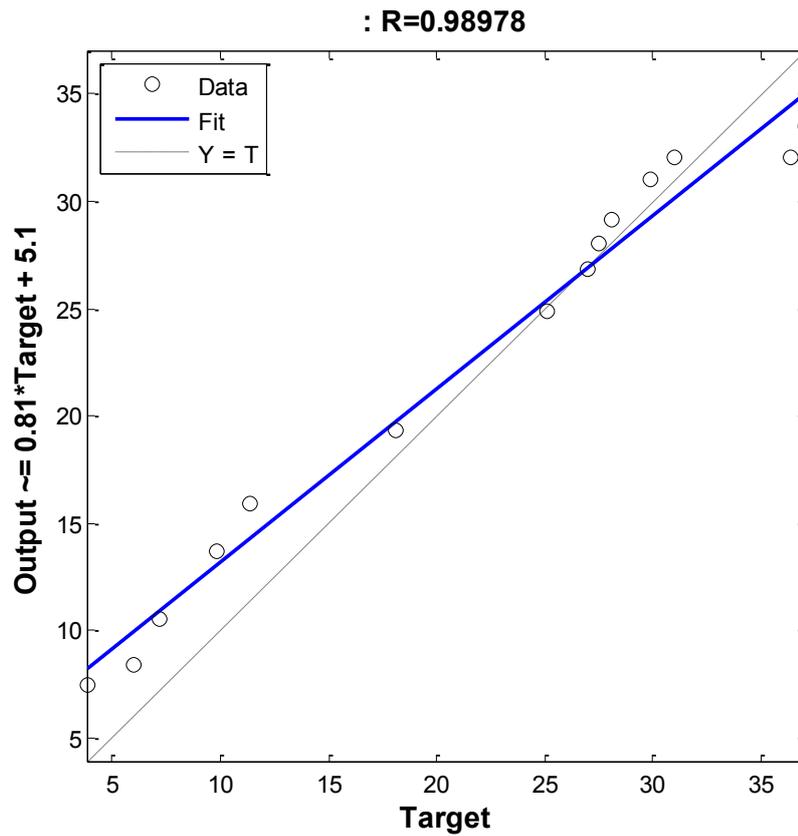


Figure II - 2 Courbe de régression linéaire du mélange RH35

II.3.2.b. Simulation de l'ajout C1

Pour la simulation du mélange RH35 avec un taux d'ajouts carbonatés C1, nous avons constitué notre base d'apprentissage à partir du RH35 pur ainsi que des proportions restantes, c'est-à-dire C2, C3, C4, C5, C6. Celles-ci, associées aux valeurs du temps constituent les entrées de la base d'apprentissage. Les valeurs cibles étant les résistances à la compression associées.

En utilisant la même fonction, nous trouvons que le réseau optimal est celui constitué de 5 neurones pour la première couche, et 4 pour la seconde.

Type de Réseau	Algorithme d'apprentissage	Couche d'entrée	Couches cachées		Couche de sortie	
		Nombre de neurones	Nombre de neurones	Fonction d'activation	Nombre de neurones	Fonction d'activation
FeedForward Perceptron Multicouches	Retro propagation du gradient Optimisation Levenberg-Marquardt	2	2	Tangente sigmoïde (<i>tansig</i>)	8	Linéaire (<i>purelin</i>)

Tableau II - 5 Architecture du réseau utilisé pour la modélisation du mélange RH35 + C1

Les figures VI-3 et VI-4 représentent respectivement la sortie (résistances à la compression) simulée avec le réseau de neurones comparée aux valeurs expérimentales de M.YADI, ainsi que la représentation de la corrélation sortie/cible avec la valeur du coefficient de corrélation R.

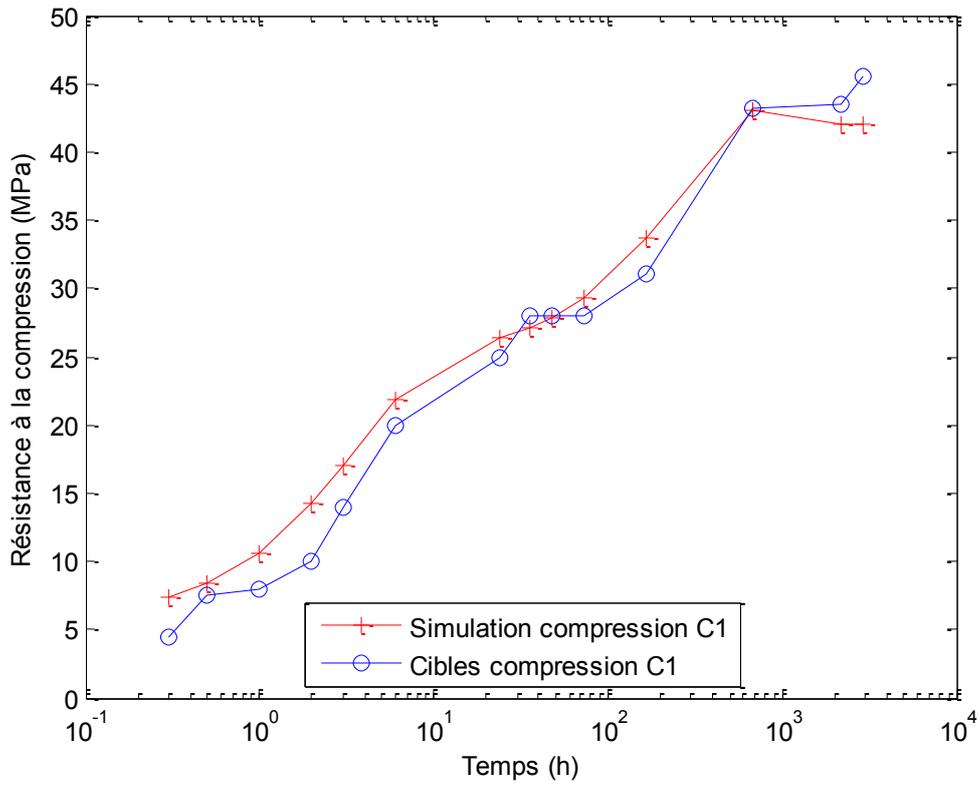


Figure II - 3 Courbe de résistance à la compression du mélange RH35 + C1

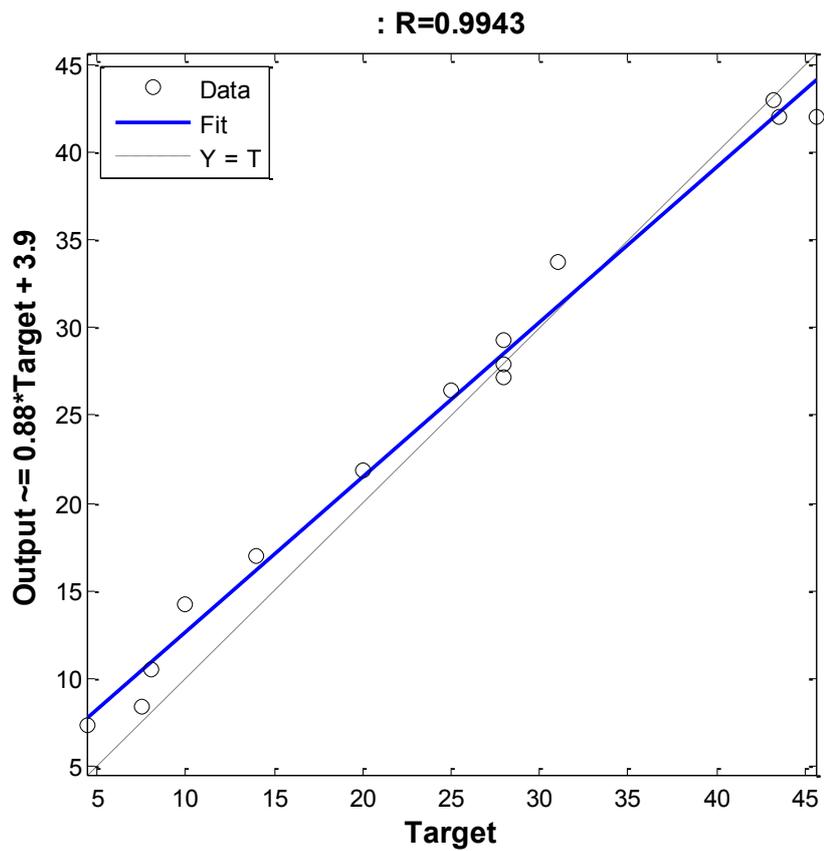


Figure II - 4 Courbe de régression linéaire du mélange RH35 + C1

II.3.2.c. Simulation de l'ajout C2

Pour la simulation du mélange RH35 avec un taux d'ajouts carbonatés C2, nous avons constitué notre base d'apprentissage à partir du RH35 pur ainsi que des proportions restantes, c'est-à-dire C1, C3, C4, C5, C6. Celles-ci, associées aux valeurs du temps constituent les entrées de la base d'apprentissage. Les valeurs cibles étant les résistances à la compression associées.

En utilisant la même fonction, nous trouvons que le réseau optimal est celui constitué de 4 neurones pour la première couche, et 3 pour la seconde.

Type de Réseau	Algorithme d'apprentissage	Couche d'entrée	Couches cachées		Couche de sortie	
		Nombre de neurones	Nombre de neurones	Fonction d'activation	Nombre de neurones	Fonction d'activation
FeedForward Perceptron Multicouches	Retro propagation du gradient Optimisation Levenberg-Marquardt	2	4	Tangente sigmoïde (<i>tansig</i>)	3	Linéaire (<i>purelin</i>)

Tableau II - 6 Architecture du réseau utilisé pour la modélisation du mélange RH35 + C2

Les figures VI-5 et VI-6 représentent respectivement la sortie (résistances à la compression) simulée avec le réseau de neurones comparée aux valeurs expérimentales de M.YADI, ainsi que la représentation de la corrélation sortie/cible avec la valeur du coefficient de corrélation R.

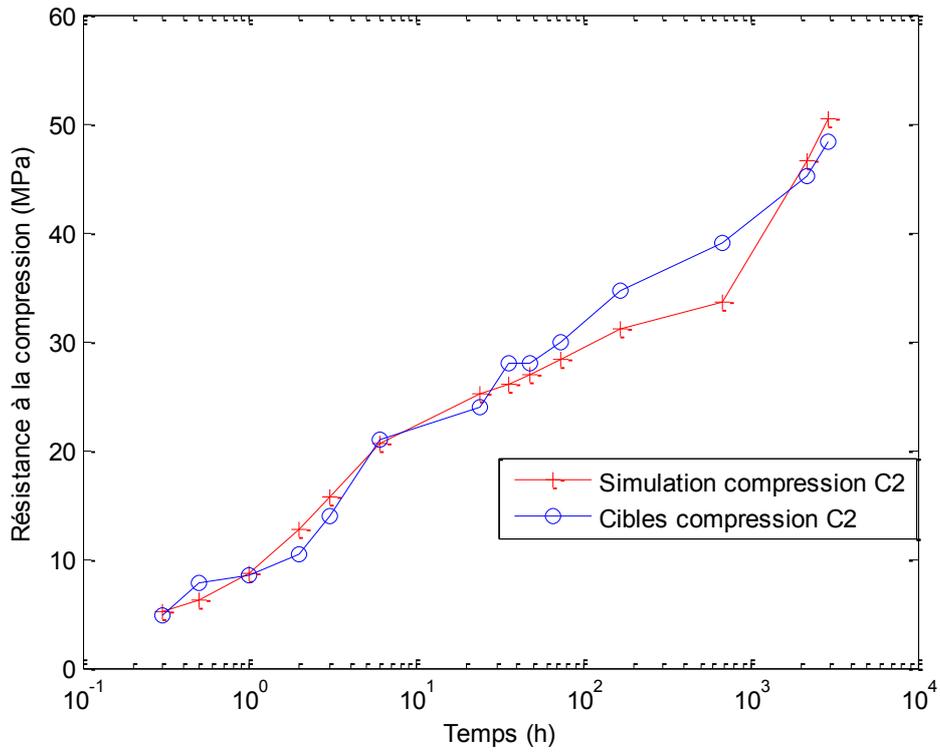


Figure II - 5 Courbe de résistance à la compression du mélange RH35 + C2

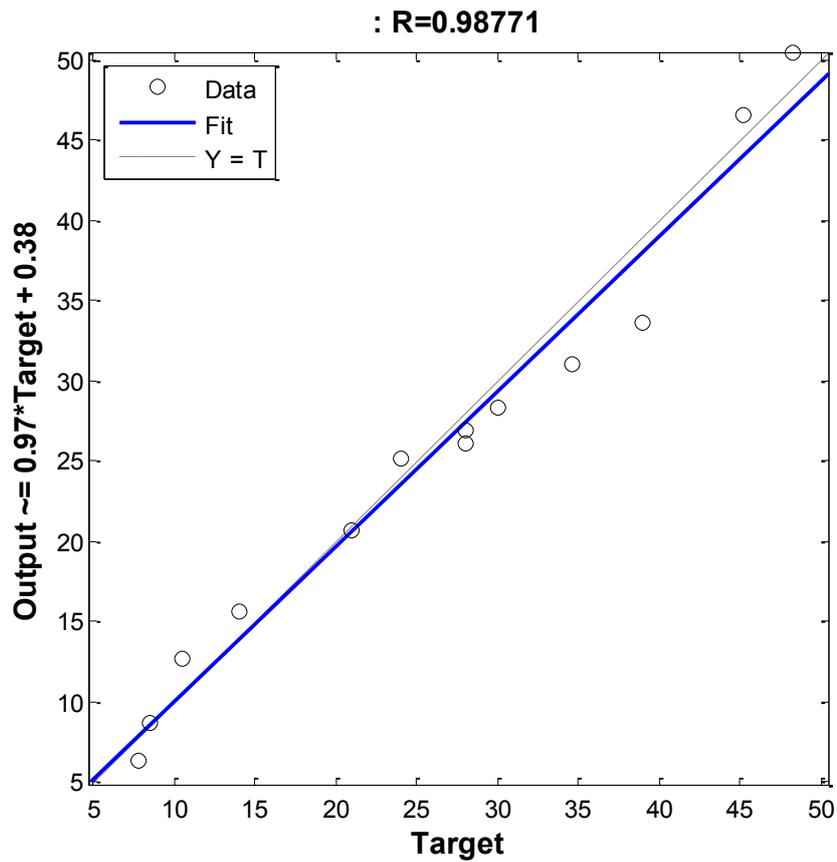


Figure II - 6 Courbe de régression linéaire du mélange RH35 + C2

II.3.2.d. Simulation de l'ajout C3

Pour la simulation du mélange RH35 avec un taux d'ajouts carbonatés C3, nous avons constitué notre base d'apprentissage à partir du RH35 pur ainsi que des proportions restantes, c'est-à-dire C1, C2, C4, C5, C6. Celles-ci, associées aux valeurs du temps constituent les entrées de la base d'apprentissage. Les valeurs cibles étant les résistances à la compression associées.

En utilisant la même fonction, nous trouvons que le réseau optimal est celui constitué de 7 neurones pour la première couche, et 7 pour la seconde.

Type de Réseau	Algorithme d'apprentissage	Couche d'entrée	Couches cachées		Couche de sortie	
		Nombre de neurones	Nombre de neurones	Fonction d'activation	Nombre de neurones	Fonction d'activation
FeedForward Perceptron Multicouches	Retro propagation du gradient Optimisation Levenberg-Marquardt	2	7	Tangente sigmoïde (<i>tansig</i>)	7	Linéaire (<i>purelin</i>)

Tableau II - 7 Architecture du réseau utilisé pour la modélisation du mélange RH35 + C3

Les figures VI-7 et VI-8 représentent respectivement la sortie (résistances à la compression) simulée avec le réseau de neurones comparée aux valeurs expérimentales de M.YADI, ainsi que la représentation de la corrélation sortie/cible avec la valeur du coefficient de corrélation R.

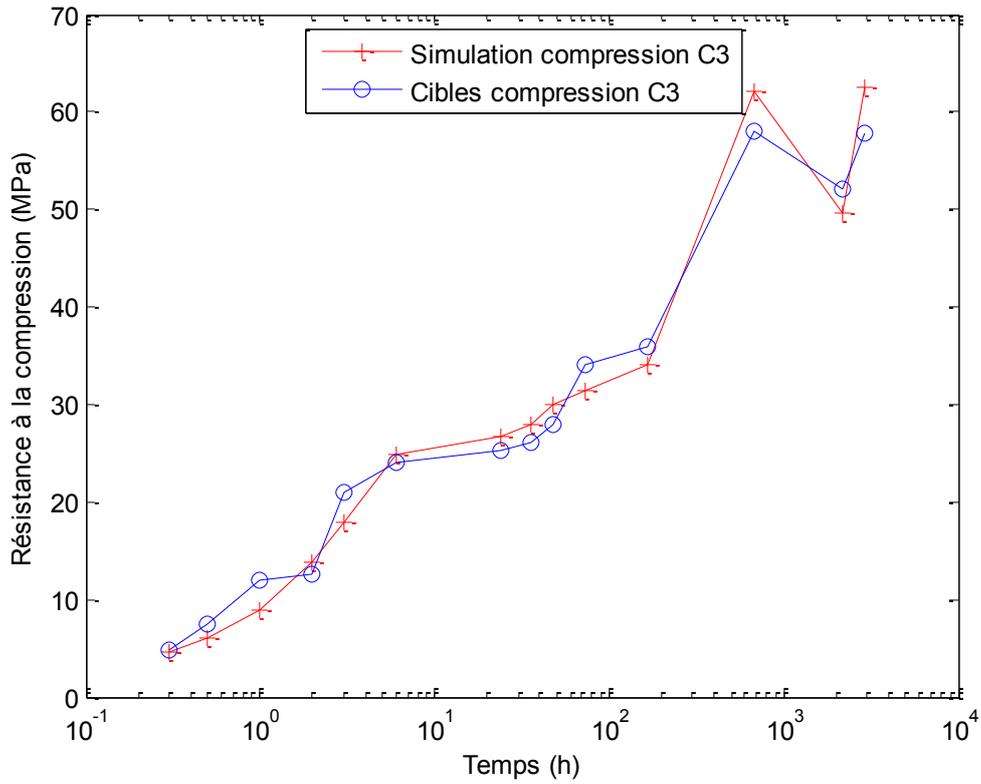


Figure II - 7 Courbe de résistance à la compression du mélange RH35 + C3

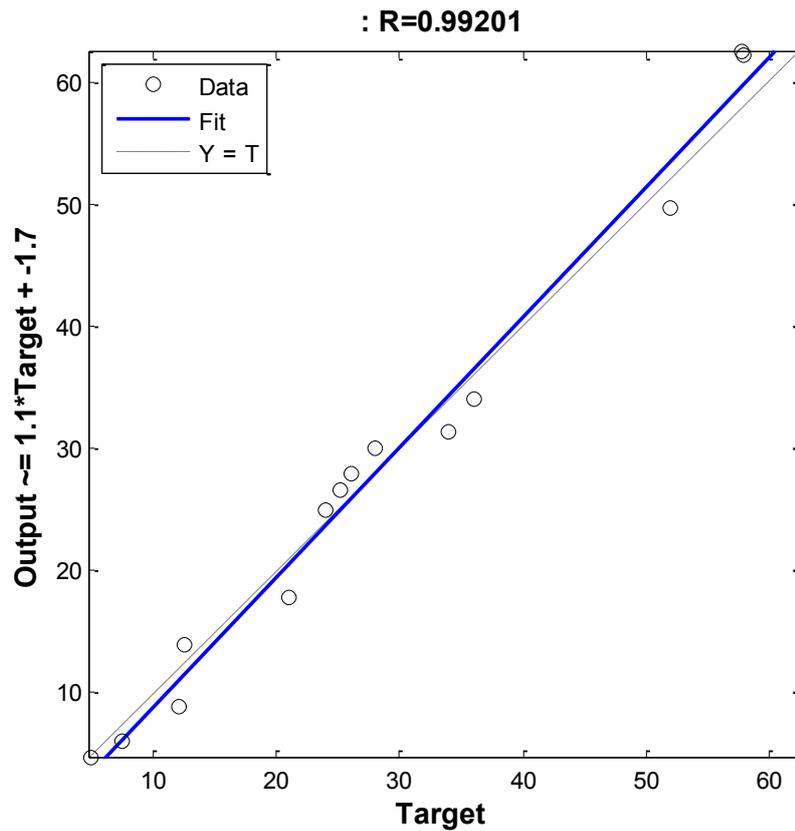


Figure II - 8 Courbe de régression linéaire du mélange RH35 + C3

II.3.2.e. Simulation de l'ajout C4

Pour la simulation du mélange RH35 avec un taux d'ajouts carbonatés C4, nous avons constitué notre base d'apprentissage à partir du RH35 pur ainsi que des proportions restantes, c'est-à-dire C1, C2, C3, C5, C6. Celles-ci, associées aux valeurs du temps constituent les entrées de la base d'apprentissage. Les valeurs cibles étant les résistances à la compression associées.

En utilisant la même fonction, nous trouvons que le réseau optimal est celui constitué de 7 neurones pour la première couche, et 7 pour la seconde.

Type de Réseau	Algorithme d'apprentissage	Couche d'entrée	Couches cachées		Couche de sortie	
		Nombre de neurones	Nombre de neurones	Fonction d'activation	Nombre de neurones	Fonction d'activation
FeedForward Perceptron Multicouches	Retro propagation du gradient Optimisation Levenberg-Marquardt	2	7	Tangente sigmoïde (<i>tansig</i>)	7	Linéaire (<i>purelin</i>)

Tableau II - 8 Architecture du réseau utilisé pour la modélisation du mélange RH35 + C4

Les figures VI-9 et VI-10 représentent respectivement la sortie (résistances à la compression) simulée avec le réseau de neurones comparée aux valeurs expérimentales de M.YADI, ainsi que la représentation de la corrélation sortie/cible avec la valeur du coefficient de corrélation R.

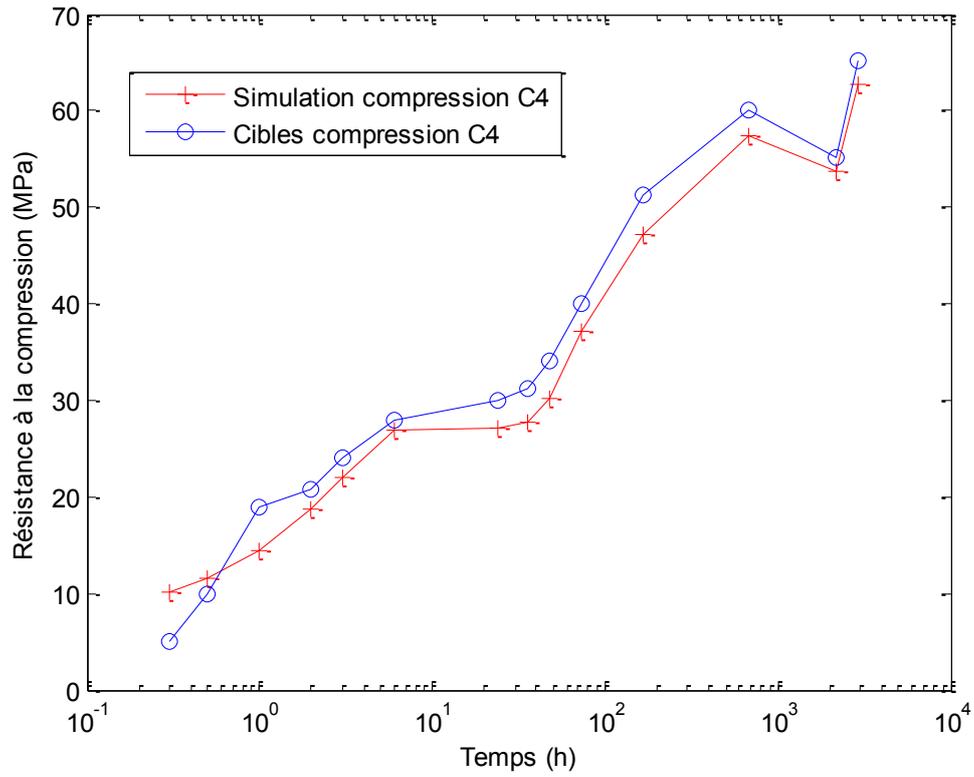


Figure II - 9 Courbe de résistance à la compression du mélange RH35 + C4

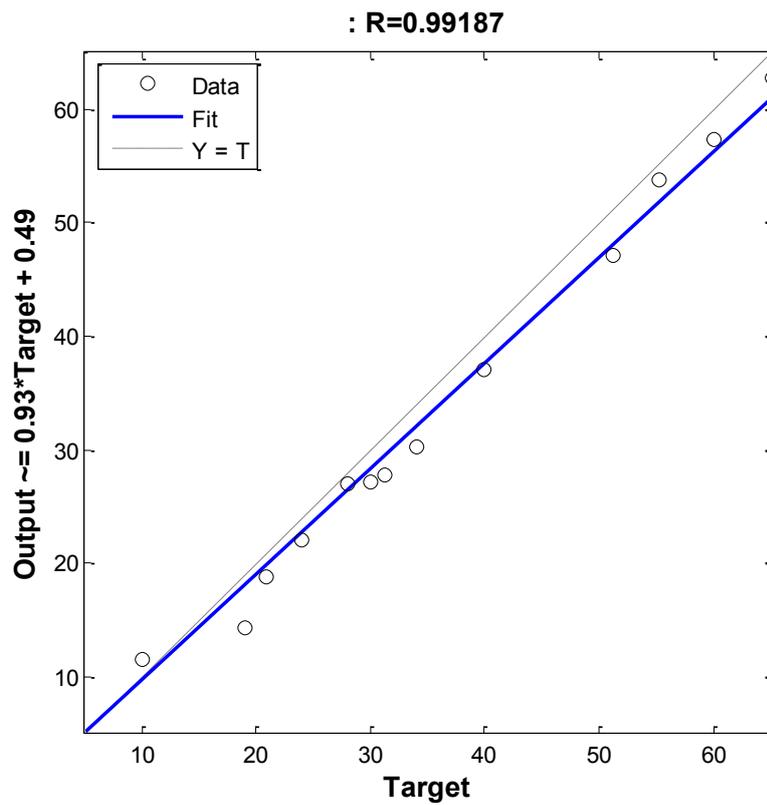


Figure II - 10 Courbe de régression linéaire du mélange RH35 + C4

II.3.2.f. Simulation de l'ajout C5

Pour la simulation du mélange RH35 avec un taux d'ajouts carbonatés C5, nous avons constitué notre base d'apprentissage à partir du RH35 pur ainsi que des proportions restantes, c'est-à-dire C1, C2, C3, C4, C6. Celles-ci, associées aux valeurs du temps constituent les entrées de la base d'apprentissage. Les valeurs cibles étant les résistances à la compression associées.

En utilisant la même fonction, nous trouvons que le réseau optimal est celui constitué de 4 neurones pour la première couche, et 3 pour la seconde.

Type de Réseau	Algorithme d'apprentissage	Couche d'entrée	Couches cachées		Couche de sortie	
		Nombre de neurones	Nombre de neurones	Fonction d'activation	Nombre de neurones	Fonction d'activation
FeedForward Perceptron Multicouches	Retro propagation du gradient Optimisation Levenberg-Marquardt	2	4	Tangente sigmoïde (<i>tansig</i>)	3	Linéaire (<i>purelin</i>)

Tableau II - 9 Architecture du réseau utilisé pour la modélisation du mélange RH35 + C5

Les figures VI-11 et VI-12 représentent respectivement la sortie (résistances à la compression) simulée avec le réseau de neurones comparée aux valeurs expérimentales de M.YADI, ainsi que la représentation de la corrélation sortie/cible avec la valeur du coefficient de corrélation R.

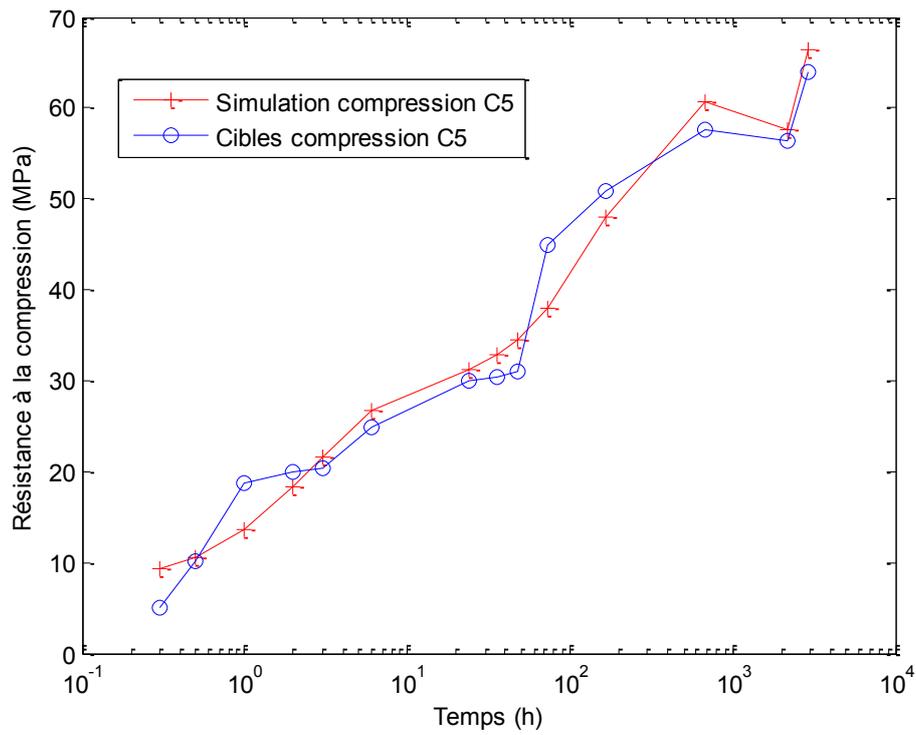


Figure II - 11 Courbe de résistance à la compression du mélange RH35 + C5

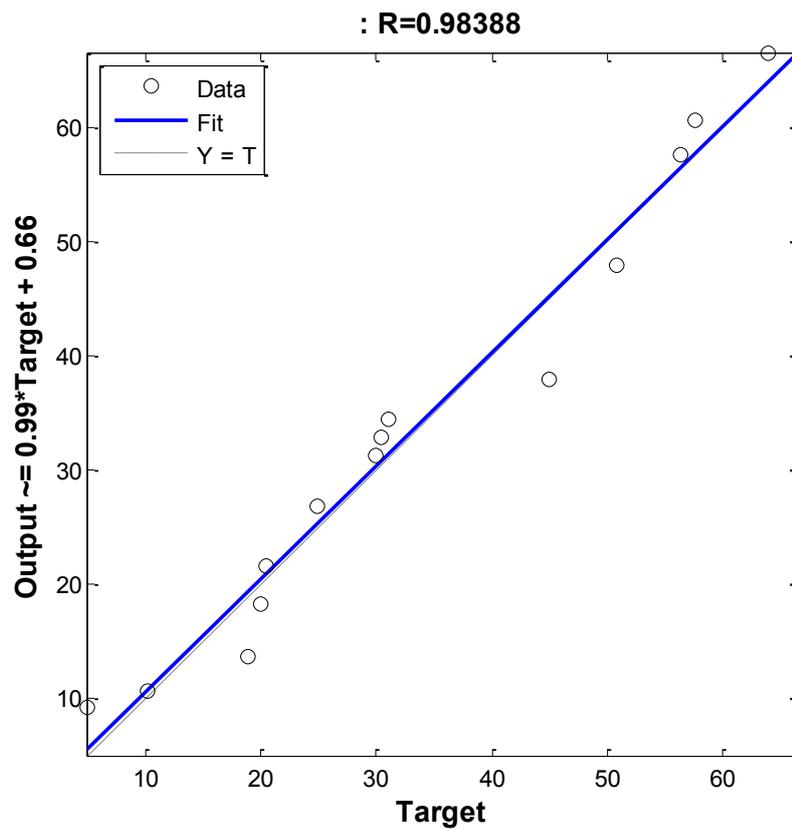


Figure II - 12 Courbe de régression linéaire du mélange RH35 + C5

II.3.2.g. Simulation de l'ajout C6

Pour la simulation du mélange RH35 avec un taux d'ajouts carbonatés C6, nous avons constitué notre base d'apprentissage à partir du RH35 pur ainsi que des proportions restantes, c'est-à-dire C1, C2, C3, C4, C5. Celles-ci, associées aux valeurs du temps constituent les entrées de la base d'apprentissage. Les valeurs cibles étant les résistances à la compression associées.

En utilisant la même fonction, nous trouvons que le réseau optimal est celui constitué de 3 neurones pour la première couche, et 3 pour la seconde.

Type de Réseau	Algorithme d'apprentissage	Couche d'entrée	Couches cachées		Couche de sortie	
		Nombre de neurones	Nombre de neurones	Fonction d'activation	Nombre de neurones	Fonction d'activation
FeedForward Perceptron Multicouches	Retro propagation du gradient Optimisation Levenberg-Marquardt	2	3	Tangente sigmoïde (<i>tansig</i>)	3	Linéaire (<i>purelin</i>)

Tableau II - 10 Architecture du réseau utilisé pour la modélisation du mélange RH35 + C6

Les figures VI-13 et VI-14 représentent respectivement la sortie (résistances à la compression) simulée avec le réseau de neurones comparée aux valeurs expérimentales de M.YADI, ainsi que la représentation de la corrélation sortie/cible avec la valeur du coefficient de corrélation R.

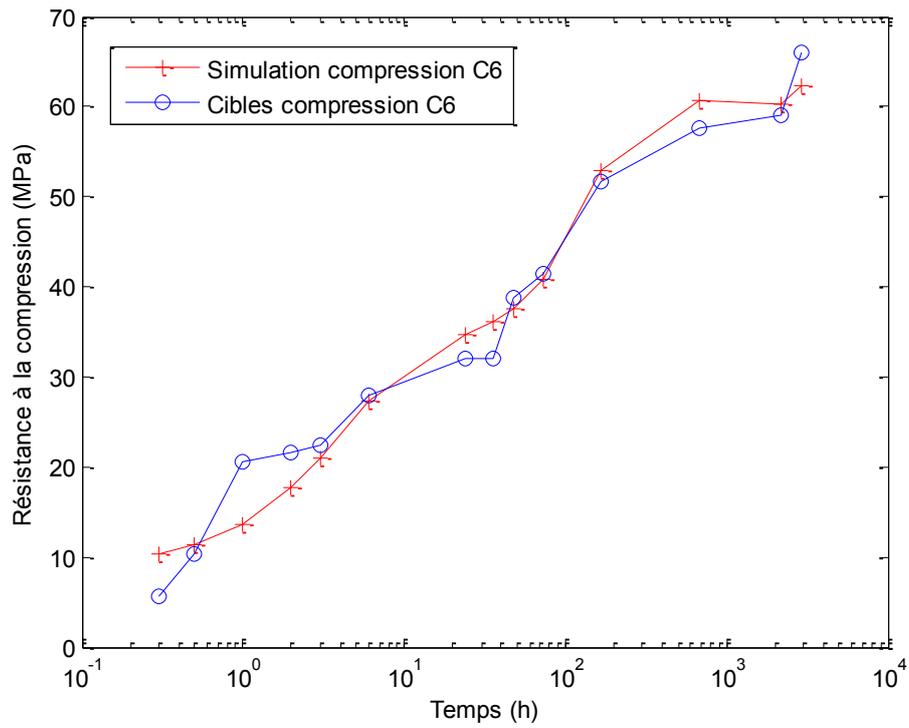


Figure II - 13 Courbe de résistance à la compression du mélange RH35 + C6

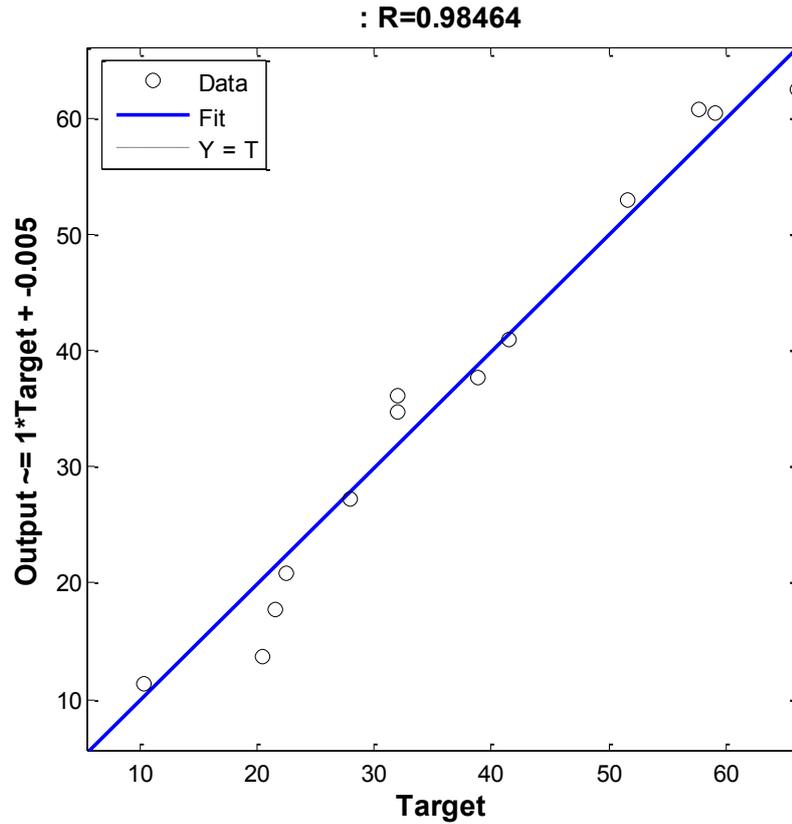


Figure II - 14 Courbe de régression linéaire du mélange RH35 + C6

II.4. Interprétation des résultats et conclusion

Au terme de nos multiples modélisations, nous pouvons effectuer les observations suivantes

- Les valeurs du coefficient de corrélation R sont satisfaisantes et se situent toutes entre 0.9839 et 0.9943. Sachant que ces valeurs peuvent être considérées comme admissibles à partir d'une valeur de $R=0.80$.

Les tableaux ci-dessous représentent les valeurs de ces coefficients R ainsi que de la valeur de R^2 correspondante.

Ajout	R	R^2
RH35 Pur	0.9897	0.980
RH35 + C1	0.9943	0.989
RH35 + C2	0.9877	0.976
RH35 + C3	0.9920	0.984
RH35 + C4	0.9919	0.984
RH35 + C5	0.9839	0.968
RH35 + C6	0.9846	0.970

Tableau II - 11 Valeurs du coefficient de corrélation de la simulation des mélanges aux ajouts

- Nous remarquons que les résultats obtenus lors de cette étape de modélisation présentent des coefficients de corrélation plus proches de la valeur unitaire. Nous justifions cela par le grand nombre d'échantillons testés et la bonne répartition temporelle des essais, ce qui nous a donc permis d'avoir une base d'apprentissage pertinente.

Conclusion générale

Conclusion générale

Au terme de notre travail qui a porté sur l'utilisation de réseaux de neurones pour la modélisation du comportement des ajouts carbonatés au mélanges de ciment Portland et de ciment alumineux, les résultats que nous avons obtenus au travers des diverses simulations montrent une corrélation plus que satisfaisante des données expérimentales. En effet, les coefficients de corrélation R obtenus grâce aux différentes étapes de modélisation étaient tous d'une valeur supérieure à 0.95.

Cette performance est étroitement liée à la qualité de la phase d'apprentissage des réseaux. La base de données expérimentale doit vérifier les conditions suivantes :

- Être assez fournie pour que le réseau puisse mettre au point une loi de comportement représentative du phénomène expérimental.
- Être homogènement répartie pour être en mesure de réaliser aussi bien des interpolations que des extrapolations.

Ceci s'est confirmé au travers des différentes phases de modélisation illustrées dans le dernier chapitre de notre projet de fin d'étude ainsi que dans ce document du master. En effet, le cinquième chapitre du PFE qui a porté sur la sélection du mélange optimal, contenait des données principalement concentrées au jeune âge, et qui handicapait la simulation au moyen et long terme contrairement à la partie des mélanges aux ajouts traitée dans le sixième chapitre qui a présenté de meilleurs résultats (avec une valeur minimale du coefficient de corrélation $R = 0.9839$). Cette bonne performance découle de la taille adéquate et de la bonne répartition de la base d'apprentissage.

Il est évident que ces deux paramètres sont d'une importance capitale pour le bon fonctionnement d'un réseau de neurones, mais celui-ci doit également présenter une architecture appropriée, dont la détermination ne s'appuie pas sur des lois et des relations, mais ne peut être effectuée qu'au travers de nombreux tâtonnements dans le but de choisir la meilleure combinaison possible de fonctions d'activations, d'algorithme d'apprentissage, de nombre de neurones et de couches cachées.

Nous avons pu contourner ce tâtonnement par la mise en place d'un processus itératif qui essaye toutes les combinaisons possibles dans un domaine prédéfini dont le choix prend comme critère de sélection l'erreur minimale de l'ensemble des itérations. Dans notre cas, nous avons utilisé l'erreur quadratique moyenne (MSE) comme référence pour parvenir à déterminer le nombre de neurones optimal pour chaque couche.

De par la bonne corrélation des résultats obtenus avec les valeurs expérimentales, notre travail confirme l'applicabilité de l'utilisation des réseaux de neurones pour l'élaboration de nouveaux types de ciments. Applicabilité qui peut certainement s'étendre particulièrement aux matériaux de construction et de façon générale au génie civil.

En tenant compte de cela, de nouvelles perspectives s'ouvrent pour leur mise au point. Il serait en effet intéressant dans le cadre de travaux similaires de se focaliser sur une bonne répartition et un plus grand nombre d'essais pour un nombre plus restreint de mélanges ce qui nous donnera une meilleure base d'apprentissage.

En optant pour des essais non destructifs pour la caractérisation des matériaux lors de l'élaboration de nouveaux ciments, une méthodologie alternative à celle suivie par AEM YADI nous apparait comme étant moins onéreuse tout en restant aussi précise et en nous donnant la possibilité d'alimenter constamment le réseau de neurones avec de nouvelles données tout en tirant profit de la disponibilité des éprouvettes pour d'éventuels investigations complémentaires.

Bibliographie:

- [1] G. DREYFUS. « Les réseaux de neurones », Mécanique Industrielle et Matériaux, 1998, n°51, 19p.
- [2] C. TOUZET. Les réseaux de neurones artificiels introduction au connexionnisme, 1992, 129p
- [3] C. SI-MOUSSA. Modélisation des équilibres de phases à pression élevées par les réseaux de neurones artificiels. Thèse de doctorat en génie chimique, Alger, Ecole Nationale Polytechnique, 2010, 249p.
- [4] M. PARIZEAU. Réseaux de neurones, Université Laval, 2004, PP 27-51.
- [5] S. AMROUCHE. Prédiction des variables climatologiques aux pas de temps mensuel et décadaire par les réseaux de neurones et l'Analyse en Composantes principales. Mémoire de fin d'études, Département d'Hydraulique, Ecole Nationale Polytechnique, Alger 2006.
- [6] I. OTMANI. L'analyse discriminante et le perceptron multicouche, Thèse de magister, Constantine, Université Mentouri , 2011, 110p.
- [7] H.DEMUTH & M.BEALE, Neural Network Toolbox, for use with MATLAB® User's Guide version 4, The MathWorks Inc, 1993.
- [8] C. BISHOP. Neural Networks for Pattern Recognition, USA, Oxford University Press, 1995, 504p.
- [9] M. PARIZEAU . « Le perceptron multicouche et son algorithme de rétropropagation des erreurs » Université Laval, Département de génie électrique et de génie informatique, 2004. 8p.
- [10] Ahmed El Mostefa YADI. Contribution expérimentale à l'élaboration d'un ciment à prise réglable. Thèse de Magister en génie des matériaux, Boumerdes , INMC, 1997, 101p.
- [11] MEHTA. « Successful prevention of loss of strength in concrete made with high alumina cement », Intern Cement Chemistry, 1969, n°2, p151-152.
- [12] A.NEGRO & al. « The hydratation of monocalcium aluminate in the presence of quartz and calcium carbonate», 1978, Il Cemento, n°3, p285-290.
- [13] M.COLLEPARDI & al. «The influence of $C_4A_3SO_3$ on the hydratation of monocalcium aluminate»,1972 , Il Cemento , n°2, p179-188
- [14] P.SCHIMMELWITZ. Influence du calcaire broyé sur les propriétés du clinker, Colloque RILEM, 1961, Prague.
- [15] H-E.SCHWIETE,U.LUDWIG , E.NIEL. Les ajouts calcaires au clinker , Seventh conference of silicates industry, 1965, Budapest.