

**UNIVERSITÉ DE SAAD DAHLEB DE BLIDA**

**Faculté des Sciences de l'Ingénieur**

Département de Génie Civil

# **MEMOIRE DE MAGISTER**

Spécialité : Structures

## **APPLICATION DES RESEAUX DE NEURONES POUR LA PREDICTION DE LA PROPAGATION DES ROTULES PLASTIQUES**

Par

**Rafik TALEB**

Devant le jury composé de :

ABED Mohamed	Maître de Conférences, U.de Blida	Président
BENNOUAR Djillali	Professeur, U.Bab-Ezzouar	Examineur
BENSAIBI Mahmoud	Maître de Conférences, U.de Blida	Examineur
LAOUAMI Nacer	Maître de Recherche, CGS	Examineur
BELLAZOUGUI Mohamed	Directeur de Recherche, CGS	Invité
BOURAHLA Nouredine	Professeur, U.Blida	Rapporteur

Blida, mars 2005

## **REMERCIEMENTS**

Arriver au terme de mon travail, la coutume exige de remercier tous ceux qui on apporté leur aide de près ou de loin a l'avancement de se travail.

Particulièrement reconnaissant à mon directeur de recherche N.BOURAHLA, je le remercie avant tout pour m'avoir accepté sous sa direction, et aussi pour m'avoir guidé tout au long du chemin, d'avoir profité de ces conseilles et de son esprit scientifique.

Je tiens à remercier les membres de jury qui ont accepté d'évaluer et de critiquer mon travail.

Je joint a ces remerciement tout nos enseignants de graduation et de post-graduation qui ont assisté à notre formation.

## RESUME

Sous l'effet d'une action sismique, les constructions sont quasiment sollicitées au-delà de leurs limites élastiques. Ainsi, le comportement post-élastique exprimé en terme de ductilité et de capacité de dissipation d'énergie est d'une importance capitale dans la conception parasismique des structures. En pratique, ces notions sont prises en compte implicitement dans les codes sismiques par des coefficients d'amplification ou de réduction de la charge sismique. Il est cependant, intéressant de pouvoir prédire le comportement post-élastique au stade de la conception et intervenir éventuellement, durant cette phase, pour assurer une grande capacité de dissipation d'énergie et pourvoir une distribution uniforme de l'activité plastique, tout en évitant le risque d'une ruine prématurée. Ce travail examine l'application de la technique des réseaux de neurones dans l'analyse quasi-statique non linéaire des portiques pour prédire la distribution des nœuds plastiques au moment de la ruine. En premier lieu, une base d'exemples est construite servant comme référence pour l'apprentissage du réseau de neurone, ensuite, de nouveaux exemples sont introduits dans le réseau pour le valider. Les résultats obtenus démontrent le potentiel de cette technique comme outil d'évaluation du comportement post-élastique dans la pratique de la conception et le calcul parasismique des structures.

## ملخص

تحت تأثير القوى الزلزالية، فإن الأبنية تتحمل جهداً أكبر بكثير من قوة التحمل الخطية. لهذا، فإن دراسة تصرف الأبنية عن طريق اللدونة و قدرة امتصاص الطاقة من الأهمية بمكان في حسابات البنية المقاومة للزلازل. في الواقع، تأخذ بعين الإعتبار في قانون حساب البنية المقاومة للزلازل بطريقة غير مباشرة عن طريق معاملات تضخيم أو تقليل قوة الزلزال المستعملة في الحساب. في هذا المجال، يعتبر من المستحسن جداً، تكهن التصرف اللاخطي في المراحل الأولى للحسب الهندسي، وكذا التدخل من أجل ضمان أكبر قدر من امتصاص طاقة الزلزال و توزيع منتظم للنشاط البلاستيكي، مع تجنب ميكانيزمات انهيار قبل الأوان. هذا العمل يدرس إمكانية تطبيق تقنية الشبكات العصبية في الدراسة الستاتيكية اللاخطية لإطارات الأبنية من أجل تكهن توزيع العقد البلاستيكية قبيل الانهيار. أولاً، قمنا بتحضير قاعدة بيانات لاستعمالها في تمرين الشبكة العصبية، ثم عرض أمثلة جديدة من أجل تأكيد تعلم الشبكة. النتائج المحصل عليها أكدت على قوة هذه التقنية في تكهن النشاط البلاستيكي في مجال الحساب الهندسي للأبنية.

## ABSTRACT

*Under seismic loading, many structural elements of the constructions are stressed beyond their elastic limit. Therefore, the post elastic behaviour expressed in terms of ductility and capacity of energy dissipation is of paramount importance for the seismic design of constructions. In practice, these notions are taken in the account implicitly in seismic design codes by amplification or reduction coefficients of seismic loading. For that, it is very interesting to predict the post elastic behaviour at design stage, to ensure the performance capacity needed to dissipate energy and premise a uniform distribution of plastic deformations, at same, prevent fragile collapse mechanisms.*

*This work examine the application of neural networks technique in the nonlinear quasi-static analysis of plane moment resistant frames to predict the plastic hinges distribution at collapse moment. Firstly, an example base was constructed and used as reference to learn the neural network, then; the network was generalized by presenting inputs that are not shown previously. The results obtained improve the potential of this technique to evaluate post-elastic behaviour in structural seismic design practice.*

## TABLE DES MATIERES

RESUME

REMERCIEMENTS

TABLE DE MATIERE

LISTE DES ILLUSTRATIONS, GRAPHIQUE ET TABLEAUX

INTRODUCTION.....	9
1. GENERALITES SUR LES RESEAUX DE NEURONES.....	11
1.1. Introduction .....	11
1.2. Historique des reseaux de neurones artificiels.....	11
1.3. Neurones biologiques.....	13
1.4. Les reseaux de neurones artificiels ( modeles mathematiques).....	14
1.5. Reseaux multicouches a retropropagation.....	24
1.6. Les reseaux de neurones a contre propagation.....	32
1.7. Reseaux de neurones de hopfield.....	36
1.8. Reseaux de neurones a resonance adaptive.....	39
1.9. Conclusion.....	41
2. APPLICATION DES RESEAUX DE NEURONES EN GENIE CIVIL .....	42
2.1. Introduction.....	42
2.2. Principaux domaines d'application des reseaux de neurones.....	42
2.3. Application des reseaux de neurones en genie civil.....	43
2.4. Conclusion.....	49
3. COMPORTEMENT NON LINEAIRE ET MECANISMES DE RUINE .....	50
3.1. Introduction.....	50
3.2. Comportement non linear.....	50
3.3. Analyse dynamique non lineaire.....	51
3.4. Analyse pushove.....	52
3.5. Approche energetique.....	54
3.6. Comportement a la rupture.....	55
3.7. La redondance dans la conception parasismique.....	59
4. CONCEPTION DU RESEAU DE NEURONES.....	62
4.1. Introduction.....	62
4.2. Base d'apprentissage .....	62
4.3. Conception du reseau de neurones.....	63
4.4. Apprentissage du reseau de neurone.....	66
4.5. Test, validation et generalisation.....	69
CONCLUSION.....	72

APPENDICES.....	74
A. LISTE DES SYMBOLES.....	74
B. ANALYSE « PUSHOVER » AVEC SAP2000 NON-LINEAR .....	76
C. « NEURAL NETWORK TOOLBOX » DE MATLAB.....	81
D. ARTICLES PUBLIES.....	86

## LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

FIGURE 1.1 : UN NEURONE AVEC SON ARBORISATION DENDRITIQUE	13
FIGURE 1.2 : FONCTIONNEMENT AU NIVEAU SYNAPTIQUE	14
FIGURE 1.3 : STRUCTURE D'UN NEURONE ARTIFICIEL	15
FIGURE 1.4 : TYPES DE FONCTIONS DE TRANSFERT POUR LE NEURONE ARTIFICIEL	17
FIGURE 1.5 : DEFINITION DES COUCHES D'UN RESEAU MULTICOUCHE.	18
FIGURE 1.6 : RESEAU A CONNEXIONS LOCALES	19
FIGURE 1.7 : RESEAU A CONNEXIONS COMPLETE	19
FIGURE 1.8 : RESEAU A CONNEXIONS RECURRENTES	20
FIGURE 1.9 : ORGANIGRAMME D'APPRENTISSAGE DE HEBB	23
FIGURE 1.10 : TOPOLOGIE D'UN RESEAU DE NEURONE MULTICOUCHE A RETRO-PROPAGATION	24
FIGURE 1.11 : PROCESSUS D'APPRENTISSAGE DIRECT ET INDIRECT	29
FIGURE 1.12 : DISTRIBUTION UNIFORME DES DONNEES DE LA BASE D'EXEMPLE	30
FIGURE 1.13 : MINIMUM GLOBALE ET LES MINIMUMS LOCAUX	32
FIGURE 1.14 : TOPOLOGIE DES RESEAUX A CONTRE PROPAGATION	33
FIGURE 1.15 : TOPOLOGIE D'UN ELEMENT INSTAR	33
FIGURE 1.16 : MODIFICATION DU POIDS SYNAPTIQUE POUR UNE SEULE ENTREE	34
FIGURE 1.17 : TOPOLOGIE DE LA STRUCTURE OUTSTAR	35
FIGURE 1.18 : TOPOLOGIE DES RESEAUX DE NEURONES DE HOPFIELD	37
FIGURE 1.19 : SCHEMAS D'UNE FONCTION D'ENERGIE A DEUX DIMENSIONS	38
FIGURE 1.20 : CIRCUIT ANALOGIQUE DU RESEAU DE HOPFIELD	39
FIGURE 1.21 : TOPOLOGIE DES RESEAUX DE NEURONES A RESONANCE ADAPTIVE.	40
FIGURE 3.1 : COURBE DE COMPORTEMENT NON-LINEAIRE	50
FIGURE 3.2 : MODELISATION D'UN SYSTEME A PDDL PAR UN SYSTEME A IDDL	53
FIGURE 3.3 : FORME GENERALE DE LA COURBE DE CAPACITE D'UNE STRUCTURE	54
FIGURE 3.4 : SYSTEME PHYSIQUE INTERAGIE AVEC L'EXTERIEUR PAR UN CHANGEMENT D'ENERGIE	54
FIGURE 3.5 : PRINCIPE DE FORMATION D'UN NOEUD PLASTIQUE	55
FIGURE 3.6 : STRUCTURE TOTALEMENT RUINEE	58
FIGURE 3.7 : MECANISME DE RUINE PARTIEL	58
FIGURE 3.8 : SYSTEME STRUCTUREL INTACTE	59
FIGURE 3.9 : DETERMINATION DU DEGRE DE REDONDANCE SOUS CHARGEMENT STATIQUE LATERAL	60
COEFFICIENT DE REDONDANCE N : (C) N=4,(D) N=8,(E) N=8,(F) N=10	



FIGURE 4.1 : SCHEMA DU PRINCIPE DE FONCTIONNEMENT DU RESEAU DE NEURONE	62
FIGURE 4.2 : FACTEURS DE RIGIDITE ELEMENTAIRES.	63
FIGURE 4.3 : COURBE FORCE DEFORMATION ET NIVEAU DE PLASTIFICATION D'UN NŒUD PLASTIQUE	64
FIGURE 4.4 : EFFET DU NOMBRE DE NEURONES DANS LA COUCHE CACHEE SUR LE NOMBRE D'ITERATION (ALGORITHME DE LEVENBERG-MARQUARDT)	66
FIGURE 4.5 : EFFET DU NOMBRE DE NEURONES DANS LA COUCHE CACHEE SUR L'ERREUR MAXIMALE (ALGORITHME DE LEVENBERG-MARQUARDT)	67
FIGURE 4.6 : COURBE DE PERFORMANCE DE L'APPRENTISSAGE.	68
FIGURE 4.7 : DISTRIBUTION DES NŒUDS PLASTIQUES (EXEMPLE 1) A) MODELE REEL B) PREDICTION PAR RESEAU DE NEURONES	69
FIGURE 4.8 : COURBE DE CORRELATION (EXEMPLE 1)	69
FIGURE 4.9 : DISTRIBUTION DES NŒUDS PLASTIQUES (EXEMPLE 2) A) MODELE REEL B) PREDICTION PAR RESEAU DE NEURONES	70
FIGURE 4.10 : COURBE DE CORRELATION (EXEMPLE 2)	70
FIGURE B.2: COURBE DE FORCE-DEFORMATION D'UNE ROTULE PLASTIQUE	77
FIGURE B.3: PROPRIETES D'UNE ROTULE PLASTIQUE	78
FIGURE B.4 : ASSIGNER LES ROULES PLASTIQUES	79
FIGURE B.5: PARAMETRES D'UN CAS DE CHARGE PUSHOVER	80
FIGURE B.6: COURBE PUSHOVER	81
FIGURE B.7: DONNEES DE LA COURBE PUSHOVER SOUS FORME DE TABLEAU	81
FIGURE C.1: RESEAU A UNE SEULE COUCHE	87
FIGURE C.2: RESEAU A DEUX COUCHE	87
TABLEAU 2.1 : REPRESENTATION BINAIRE DES PARAMETRES D'OPTIMISATION (HAJELA, 1998)	47
TABLEAU 4.1 : COMPARAISON ENTRE LES ALGORITHMES D'APPRENTISSAGE	67
TABLEAU 4.2 : ERREUR DANS LA PREDICTION DU DEGRE DE PLASTIFICATION POUR CHAQUE NŒUD	70

## INTRODUCTION

### 1.1. Motifs et objectifs

Le phénomène de ruine structurel partiel ou total des constructions est largement observé lors des événements sismiques. Il représente l'un des dommages les plus redoutables. Souvent, il est engendré par l'apparition d'un mécanisme prématuré dans la structure dû au manque de redondance dans le système de contreventement. Il s'agit de la formation d'un certain nombre critique de nœuds plastiques dans les éléments de la structure qui entraîne la défaillance de la structure.

D'un point de vue structurel, une conception parasismique optimale, consiste à imposer la propagation ou la concentration des dommages dans une structure soumise à une sollicitation sismique, d'une manière à éviter la formation prématurée d'un mécanisme dans la structure portante et de différer l'état de ruine le maximum possible. Ce comportement donne à la structure un champ de ductilité suffisant pour dissiper le maximum d'énergie et d'augmenter sa capacité de résistance et de stabilité.

La propagation des nœuds plastiques dans une structure sous l'action d'une excitation sismique est un processus non-linéaire complexe. La procédure d'optimisation devient donc une tâche très difficile.

Le présent travail, présente un réseau de neurone capable de prédire la formation des nœuds plastiques et le mécanisme de ruine probable dans une structure sous l'action d'un système de force statique équivalent. Vu la complexité du problème, on s'est limité à des structures auto-stables planes à deux travées et trois étages.

L'approche neuronale présente l'avantage de surpasser l'étape algorithmique d'écriture du processus à suivre pour résoudre le problème qui s'avère très onéreux dans ce cas. Toutefois, les étapes concernant le choix de l'architecture et le type du réseau de neurones, la constitution de la base de données, l'apprentissage ainsi que la validation du modèle sont présentés.

### 1.2. Plan du travail

Dans le chapitre 2, on présente un aperçu théorique générale sur le fonctionnement des réseaux de neurones artificiel.

Le chapitre 3 sera consacré à tracer brièvement l'état d'art de l'application des réseaux de neurones dans quelques branches de génie civil. La branche de l'analyse des structures sera discutée un peu en détails.

Au chapitre 4, on présente quelques méthodes d'analyse du comportement non linéaire et mécanisme de ruine.

Les étapes de conception du réseau de neurone et son validation font l'objet du chapitre 5.

Et finalement, la conclusion générale récapitule les points essentiels de ce travail et propose des recommandations.

## **CHAPITRE 1**

### **GENERALITES SUR LES RESEAUX DE NEURONES**

#### 1.1. Introduction

L'informatique est la science du traitement automatique de l'information. Son développement est souvent confondu avec les machines de traitement : les ordinateurs, qui sont devenues de plus en plus puissants.

Cependant, cette augmentation de puissance ne permet pas toujours de résoudre les problèmes d'une application informatique dans un domaine particulier. Le développement de logiciels s'appuie sur deux approches. Deux parmi les plus utilisées sont l'approche algorithmique et l'approche basée sur la connaissance.

Une approche algorithmique nécessite l'écriture du processus à suivre pour résoudre le problème. Lorsque le problème est complexe, ce peut être une étape coûteuse ou impossible. D'autre part, les ordinateurs sont des machines complètement logiques qui suivent à la lettre chacune des instructions du programme.

C'est un avantage lorsque tous les cas ont été prévus à l'avance, ce qui n'est pas toujours possible.

Une autre approche du traitement automatique de l'information est proposer, où l'on cherche à s'inspirer du traitement de l'information effectué par le cerveau. L'hypothèse principale, à la base de l'essor des réseaux de neurones artificiels, est que le comportement intelligent est sous-tendu par un ensemble de mécanismes mentaux.

Dans une première partie nous examinons le passage des modèles des réseaux de neurones biologiques à des modèles mathématiques : les réseaux de neurones artificiels, ensuite nous décrivons le fonctionnement des réseaux de neurones artificiels, ainsi que leurs propriétés d'apprentissage et d'adaptation.

#### 1.2. Historique des réseaux de neurones artificiels [6]

*1880* : W. James, célèbre psychologue américain introduit le concept de mémoire associative, et propose ce qui deviendra une loi de fonctionnement de l'apprentissage sur les réseaux de neurones connue plus tard sous le nom de la loi de Hebb.

*1943* : J.McCulloch et W.Pitts laissent leurs noms à une modélisation du neurone biologique. Ce sont les premiers à montrer que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétique et symboliques complexes.

*1949* : D.Hebb, physiologiste américain explique le conditionnement chez l'animal par les propriétés des neurones eux-même. Ainsi, il à proposée la loi de modifications des propriétés des connexions entre neurones.

*1957* : F.Rosenblatt développe le modèle du Perceptron. Il construit le premier neuro-ordinateur basé sur ce modèle et l'applique a domaine de la reconnaissance de formes.

*1960* : B.Widrow, un automaticien, développe le modèle Adaline, qui ressemble dans ça structure au Perceptron, ce pendant la loi d'apprentissage est différent. Celle-ci est à l'origine de l'algorithme de rétro-propagation de gradient très utilisé aujourd'hui avec les Perceptrons multicouches.

*1969* : M.Minsky et S.Papert publient un ouvrage qui met en exergue les limitations théoriques du Perceptrone, notamment concernant l'impossibilité de traiter par ce modèle des problèmes non linéaires.

*1982* : J.J.Hopfield, un physicien à qui l'on doit le renouveau d'intérêt pour les réseaux de neurones artificiels. Il fixe préalablement le comportement à atteindre pour son modèle et construit à partir de là, la structure et la loi d'apprentissage correspondant au résultat espéré. Alors que les auteurs jusqu'alors tendent à proposer une structure et une loi d'apprentissage, puis à étudier les propriétés émergentes. Ce modèle d'Hopfield est aujourd'hui très utilisé pour des problèmes d'optimisation.

*1983* : La machine de Boltzmann est le premier modèle connu apte à traiter, de manière satisfaisante, les limitations recensées dans le cas du Perceptron. Mais l'utilisation pratique s'avère difficile, la convergence de l'algorithme étant extrêmement longue, les temps de calcul sont considérables.

1985 : la retropropagation de gradient apparaît. C'est un algorithme d'apprentissage adapté aux réseaux de neurones multicouches.

### 1.3. Neurones biologiques

#### 1.3.1. Modèle neurophysiologique

Le neurone est une cellule composée d'un corps cellulaire et d'un noyau. Le corps cellulaire se ramifie pour former ce que l'on nomme les dendrites.

C'est par les dendrites que l'information est acheminée de l'extérieur vers le corps du neurone. L'information traitée par le neurone chemine ensuite le long de l'axone pour être transmise aux autres neurones. La transmission entre deux neurones n'est pas directe, en fait, il existe une jonction entre chaque deux neurones appelée synapse (Figure 2.1).

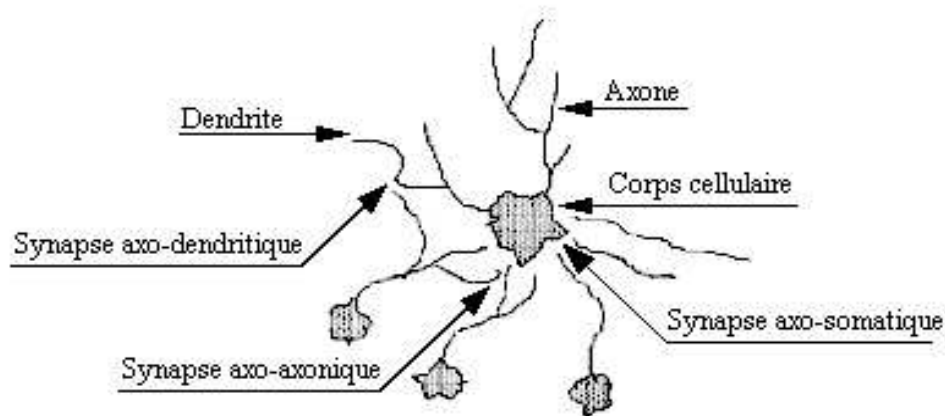


Figure 1.1 : Un neurone biologique avec son arborisation dendritique [6]

#### 1.3.2. Notion de potentiel d'action et d'habituation

A l'arrivée d'un potentiel d'action sur une synapse, un neuromédiateur est libéré dans l'espace synaptique. Il va ouvrir des canaux ioniques sur la membrane post-synaptique, créant ainsi une dépolarisation. Les dépolarisations unitaires sont sommées dans l'espace (toutes les synapses du neurone) et dans le temps et génèrent, éventuellement, un potentiel d'action sur le neurone post-synaptique (Figure 1.2).

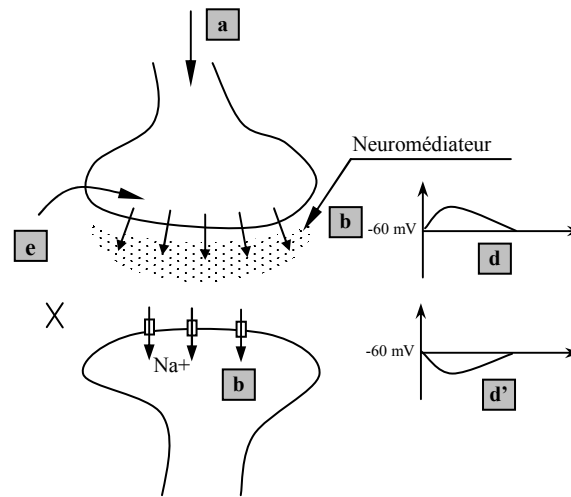


Figure 1.2 : Fonctionnement au niveau synaptique [6]

Sur la figure :

- Arrivée d'un potentiel d'action.
- Libération du neuromédiateur dans l'espace synaptique.
- Ouvertures des canaux ioniques dues au neuromédiateur
- Génération d'un potentiel d'action évoqué excitateur.
- Génération d'un potentiel évoqué inhibiteur. Les synapses inhibitrices empêchent la génération de potentiel d'action.
- Fermeture des canaux, élimination ou re-capture du neuromédiateur.

L'activation répétée d'un neurone ou d'une structure de neurones la réponse des neurones adjacente devient plus faible est va crée une sorte d'habitude : est c'est ce que l'on appelle l'apprentissage du réseau de neurones.

Chaque structure de neurones biologiques est dotée d'une fonction particulière, et ces structures adaptent leur comportement par des mécanismes d'apprentissage. L'apprentissage implique des modifications physiques des connexions entre neurones.

L'association entre plusieurs structures neuronales, dotée chacune d'une fonction précise, permet l'émergence d'une fonction supérieure pour l'ensemble.

#### 1.4. Les réseaux de neurones artificiels ( modèles mathématiques)

##### 1.4.1. Définition

C'est à partir de l'hypothèse que le comportement intelligent émerge de la structure et du comportement des éléments de base du cerveau que les réseaux de neurones artificiels se sont développés.

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires (neurone) fonctionnant en parallèle. Chaque neurone est doté d'un état interne, l'activation, par la quelle il influence les autres neurones du réseau, cette activation se propage dans le réseau par le long des liens synaptiques. La règle qui détermine l'activation d'un neurone en fonction de l'influence de ces pairs est appelée fonction d'activation.

L'intérêt d'un réseau de neurones est le calcul d'une valeur de sortie unique par chaque neurone de sortie sur la base des informations qu'il reçoit.

Les réseaux de neurones artificiels sont des modèles dont ils peuvent être décrit par leurs comportements, leurs variables descriptives et les interactions des composants.

#### 1.4.2. Composants d'un neurone artificiel

##### 1.4.2.1. Structure

Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre de variable d'entrées en provenance de neurones amont. A chacune de ces entrées est associée un poids  $w$  (abréviation de « *weight* » en anglais) représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurone aval. A chaque connexion est associée un poids (Figure 1.3).

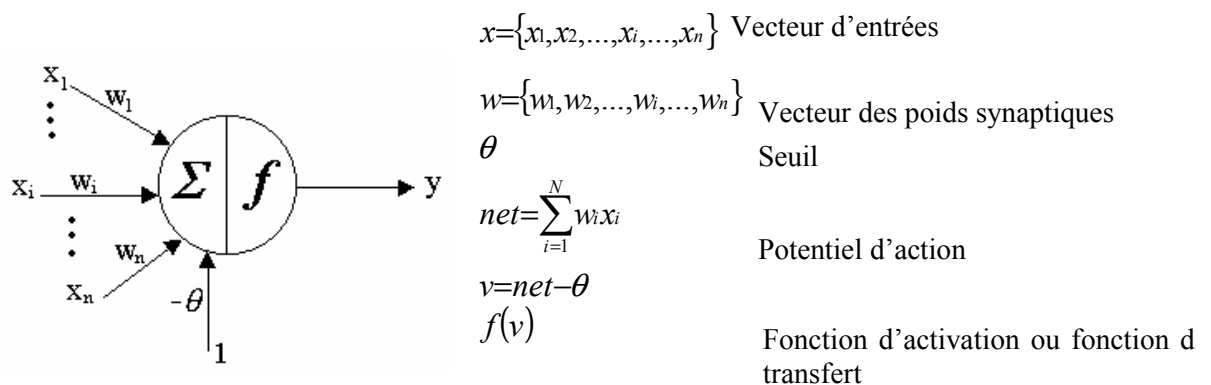


Figure 1.3 : Structure d'un neurone artificiel

Pour le neurone d'indice  $i$ , les entrées sur celui-ci sont de poids  $w_{ij}$ , alors que les connexions avals sont de poids  $w_{ki}$ .



### 1.4.2.2. Comportement

Le vecteur d'entrées  $x$  est multiplié par le vecteur des poids synaptiques, ce produit est sommé au niveau du neurone. Le neurone ne renvoi une valeur que si cette somme est accumulée avec la seuil  $\theta$  (appelée aussi biais). le biais est évaluée comme le poids synaptique d'une entrée additionnel de valeur 1.

A partir de cette valeur, une fonction de transfert calcule la valeur de l'état du neurone, la sortie du neurone est donc  $o = f(v) = f\left(\sum_{i=1}^N w_i x_i - \theta\right)$ . C'est cette valeur qui sera transmise aux neurones aval.

Il existe de nombreuses formes possibles pour la fonction de transfert (Figure 1.4). On remarquera qu'à la différence des neurones biologiques dont l'état est binaire, la plus part des fonctions de transfert sont continues, offrant une infinité de valeurs possibles comprises dans l'intervalle  $[0, +1]$  ou  $[-1, +1]$  selon la fonction utilisée. Les fonctions de transfert couramment utilisées sont listées ci-dessous :

a. Fonction linéaire :

$$f(v) = \beta v \quad \text{pour } \beta > 0$$

b. Fonction binaire :

$$f(v) = f(\text{net} - \theta) = \begin{cases} 1 & \text{pour } \text{net} > \theta \\ 0 & \text{pour } \text{net} \leq \theta \end{cases}$$

c. Fonction bipolaire :

$$f(v) = f(\text{net} - \theta) = \begin{cases} 1 & \text{pour } \text{net} > \theta \\ -1 & \text{pour } \text{net} \leq \theta \end{cases}$$

d. Fonction sigmoïde

$$f(v) = \frac{1}{1 + \exp(-\sigma v)} \quad \text{pour } \sigma > 0$$

$$\frac{df}{dv} = \sigma f(1-f)$$

e. Fonction sigmoïde bipolaire :

$$f(v) = \frac{1 - \exp(-\sigma v)}{1 + \exp(-\sigma v)} \quad \text{pour } \sigma > 0$$

$$\frac{df}{dv} = \frac{\sigma}{2} f(1-f)$$

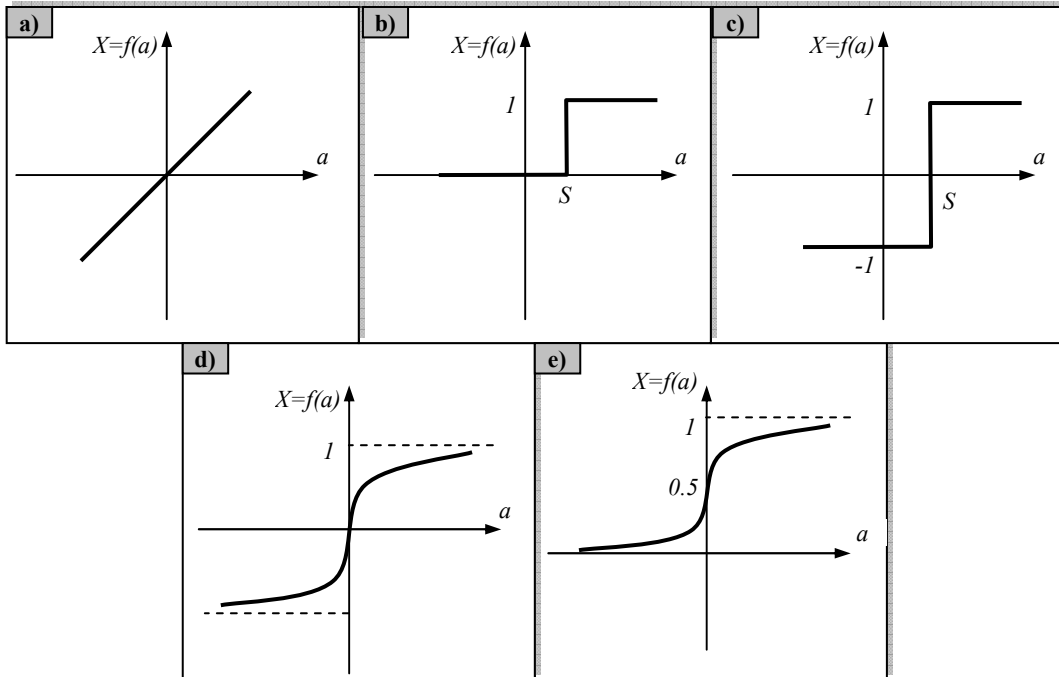


Figure 1.4 : Types de fonctions de transfert pour le neurone artificiel

a : linéaire , b : binaire, c : bipolaire, d : sigmoïde, e : sigmoïde polaire. [17]

### 1.4.3. Structures d'interconnexion

La capacité numérique d'un seul neurone est souvent très limitée. Cependant, lorsque plusieurs neurones sont connectés, formant ainsi un réseau de neurones, un comportement complexe peut être assuré. Les connexions entre ces neurones décrivent la topologie ou l'architecture du réseau. Dans les paragraphes qui suivent on donne une brève description des architectures de réseaux les plus utilisés.

### 1.4.3.1. Réseau multicouche

Les neurones sont arrangés par couche. Il n'y a pas de connexion entre neurones d'une même couche et les connexions ne se font qu'avec les neurones de la couche avale. (Figure 1.5). Habituellement, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante et celle-ci seulement. Ceci nous permet d'introduire la notion de sens de parcours de l'information (de l'activation) au sein d'un réseau et donc définir les concepts de neurones d'entrée, neurones de sortie. Par extension, on appelle couche d'entrée l'ensemble des neurones d'entrée, couche de sortie l'ensemble des neurones de sortie. Les couches intermédiaires n'ayant aucun contact avec l'extérieur sont appelées couches cachées.

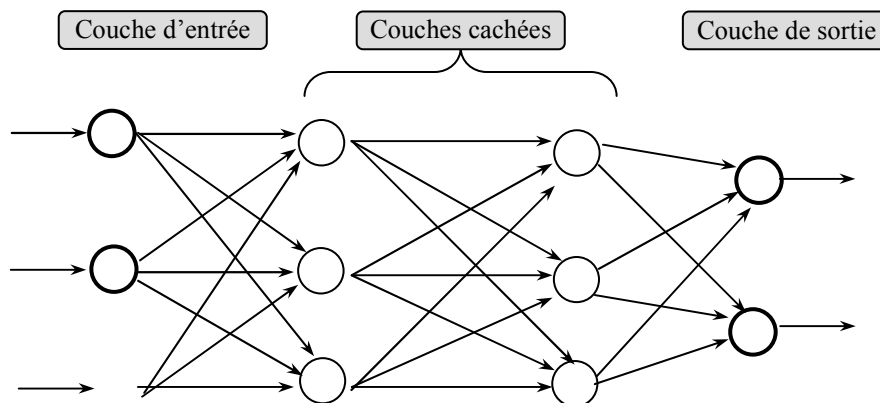


Figure 1.5 : Définition des couches d'un réseau multicouche [6].

### 1.4.3.2. Réseau à connexions locales

Il s'agit d'une structure multicouche, mais qui à l'image de la rétine, conserve une certaine topologie. Chaque neurone entretient des relations avec un nombre réduit et localisé de neurones de la couche avale (Figure 1.6). Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique.

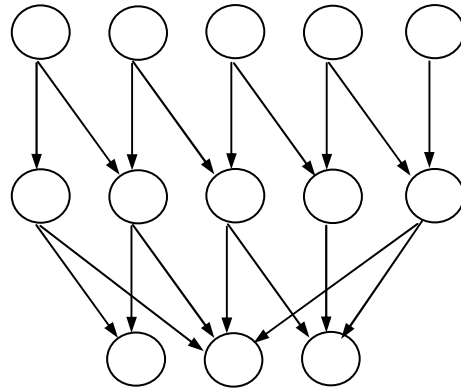


Figure 1.6 : Réseau à connexions locales [6].

#### 1.4.3.3. Réseau à connexions complète

C'est la structure d'interconnexion la plus générale (Figure 1.7). Chaque neurone est connecté à tous les neurones du réseau et à lui-même.

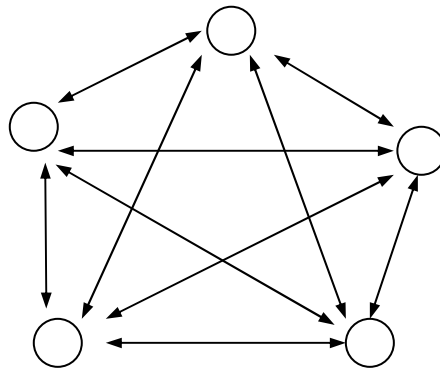


Figure 1.7 : Réseau à connexions complète [6].

#### 1.4.3.4. Réseau à connexions récurrentes

Les connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouche. Ces connexions sont le plus souvent locales (Figure 1.8).

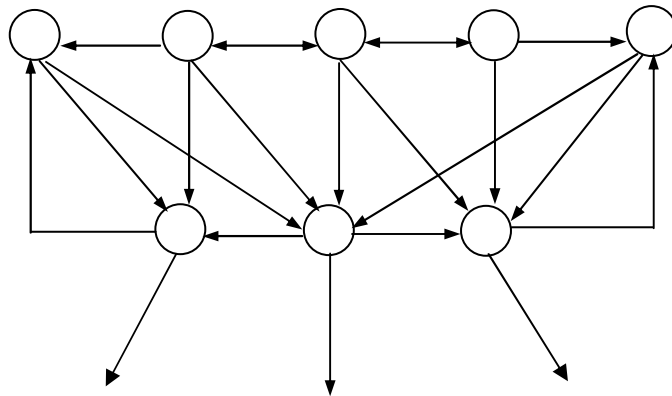


Figure 2.8 : Réseau à connexions récurrentes [6].

#### 2.4.4. le choix des poids synaptiques

Une fois l'architecture du réseau déterminée, il reste encore à choisir les valeurs des poids synaptiques des connexions que cette architecture prescrits.

De fait, la plupart des modèles de réseau de neurones disposent de mécanismes capables de modifier leurs poids synaptiques automatiquement ; ils sont dotés de règles d'apprentissage. Avec la fonction d'activation et les contraintes de connectivité, cette règle, ainsi que le choix de fonctions d'activations et de contrainte de connectivité, constituent les principaux éléments qui définissent un modèle de réseau de neurones.

Il en résulte que le réseau possède deux dynamiques : celle de l'activation de ces neurones et celle des poids de ces liens.

#### 2.4.5. Les entrées et les sorties du réseau

Tout l'intérêt d'un réseau de neurones réside en sa capacité de résoudre des problèmes pratiques. Pour l'utiliser, il est donc nécessaire de communiquer au réseau les données du problème en question, et de lui soutirer le résultat de ces calculs.

Les neurones d'un réseau sont donc divisés en trois ensembles :

- L'ensemble des neurones d'entrée qui perçoivent les données du problème.
- L'ensemble des neurones de sortie dont l'activation sera interprété comme la réponse du réseau.

- L'ensemble des neurones cachés qui ne sont ni à l'entrée ni à la sortie du réseau. Les neurones cachés n'ont aucun lien avec l'utilisateur et doivent donc agir par l'intermédiaire d'autres neurones. Un réseau disposant de neurones cachés est souvent plus puissant qu'un réseau sans neurones cachés.

Un mécanisme de traduction est nécessaire pour convertir les données en valeurs d'entrées, et pour exprimer les valeurs de sortie sous une forme plus compréhensible. Le mécanisme par lequel des informations provenant de l'environnement sont traduites en données d'entrée est appelé codage d'entrée. De même, un codage de sortie convertit les valeurs d'activation des neurones de sortie en information utilisable.

#### 2.4.6. Apprentissage des réseaux de neurones

##### 2.4.6.1. Définition

L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. L'apprentissage neuronal fait appel à des exemples de comportement.

Dans le cas des réseaux de neurones artificiels, on ajoute souvent à la description du modèle l'algorithme d'apprentissage. Dans la majorité des algorithmes actuels les variables modifiées pendant l'apprentissage sont les poids des connexions. L'apprentissage est la modification des poids du réseau dans l'optique d'accorder la réponse du réseau aux exemples et à l'expérience. A l'issue de l'apprentissage, les poids sont fixés : c'est alors la phase d'utilisation.

Certains modèles de réseaux sont improprement dénommés à apprentissage permanent. Dans ce cas il est vrai que l'apprentissage ne s'arrête jamais, ce pendant on peut toujours distinguer une phase d'apprentissage et une phase d'utilisation. Cette technique permet de conserver au réseau un comportement adopté malgré les fluctuations dans les données d'entrées.

##### 2.4.6.2. Classes d'apprentissage

Au niveau des algorithmes d'apprentissage, il a été défini deux grandes classes selon que l'apprentissage est dit supervisé ou non supervisé. Cette distinction repose sur la

forme des exemples d'apprentissage. Dans le cas de l'apprentissage supervisé, les exemples sont des couples (Entrée, Sortie associée) alors que l'on ne dispose que des valeurs d'entrées pour l'apprentissage non supervisé. Remarquons cependant que les modèles à apprentissage non supervisé nécessitent avant la phase d'utilisation une étape de labellisation effectuée par l'opérateur, qui n'est pas autre chose qu'une part de supervision.

a) *La loi de HEBB (Un exemple d'apprentissage non supervisé)*

La loi de *HEBB* s'applique aux connexions entre neurones, elle s'exprime de la façon suivante : « si deux neurones sont activés en même temps alors la force de la connexion augmente ». Cette loi peut être modélisée par les équations suivantes :

$$W_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \quad (W_{ij}(t+1) \text{ est le nouveau poids, } W_{ij}(t) \text{ est l'ancien})$$

$\Delta w_{ij}(t) = x_i \cdot x_j$  (la co-activation est modélisée comme le produit de deux valeurs d'activation).

L'algorithme d'apprentissage modifie de façon itérative les poids des neurones pour adapter la réponse obtenue à la réponse désirée. Il s'agit en fait de modifier les poids lorsqu'il y a erreur seulement.

Le processus d'apprentissage se fait selon l'organigramme ci-dessus.

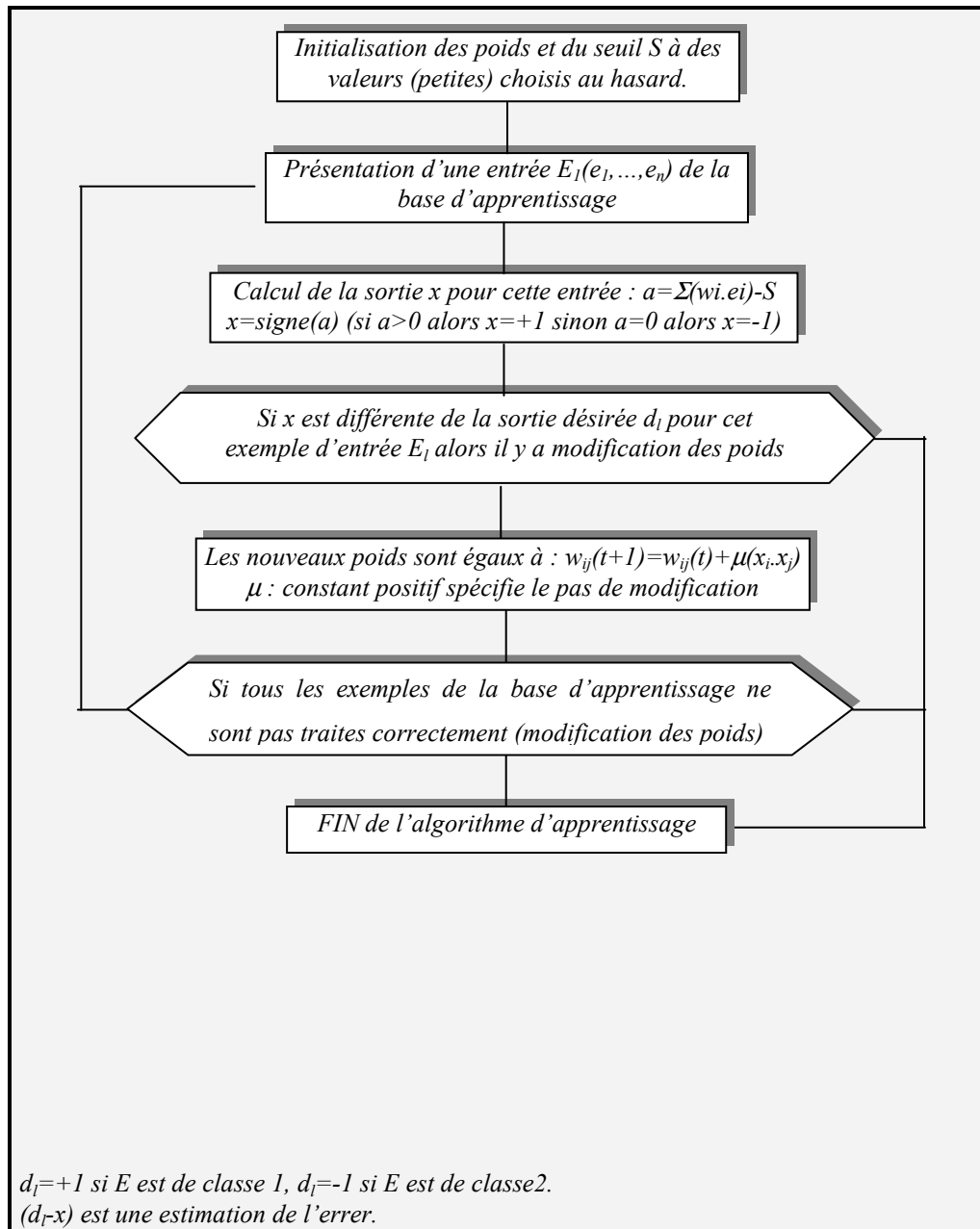


Figure 1.9 : Organigramme d'apprentissage de Hebb [6].

b) La règle d'apprentissage du Perceptron (un exemple d'apprentissage supervisé)

La règle de *HEBB* ne s'applique pas dans certain cas, bien qu'une solution existe. Un autre algorithme d'apprentissage a donc été proposé qui tient compte de l'erreur observée en sortie.

L'algorithme de l'apprentissage du Perceptron est semblable à celui utilisé pour la loi de *HEBB*. Les différences se situent au niveau de la modification des poids.



Dans les paragraphes qui suivent, on donne la topologie de quelques types de réseaux de neurones les plus répandus et les plus utilisés dans le domaine de la génie civil.

## 1.5. Réseaux multicouches à rétropropagation

### 1.5.1. Définition

Apparus en 1985, les réseaux multicouches sont aujourd'hui les modèles les plus employés.

Plusieurs couches de traitement leur permettent de réaliser des associations non linéaires entre l'entrée et la sortie. On savait que les possibilités de traitement des réseaux multicouches sont supérieures à celle du perceptron, cependant l'algorithme d'apprentissage manquait.

Après plusieurs travaux de recherche, on a pu découvrir le principe de la rétropropagation de gradient qui est une minimisation d'une fonction dépendante de l'erreur.

Une perception intuitive de cet algorithme consiste à considérer l'apprentissage comme la recherche sur la surface de coût de la position de coût minimal. À chaque configuration de poids correspond un coût. Le gradient est une estimation locale de la pente de la surface. La minimisation du gradient permet de parcourir cette surface orthogonalement aux courbes de niveau d'un pas fixé.

### 1.5.2. Structure et fonctionnement

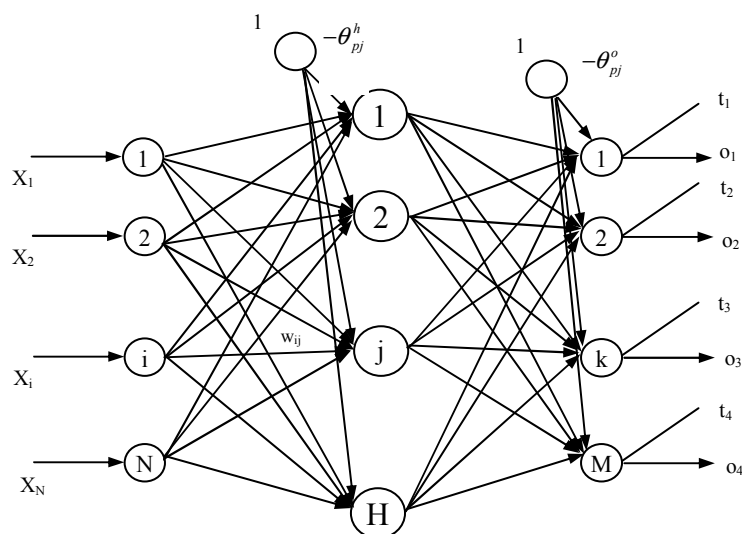


Figure 1.10 : Topologie d'un réseau de neurone multicouche à rétro propagation [35]

$x = \{x_1, x_2, \dots, x_i, \dots, x_N\}$  Vecteur des entrées du réseau de neurone de dimension  $N$  ;

$o = \{o_1, o_2, \dots, o_i, \dots, o_M\}$  Vecteur des sorties du réseau de neurone de dimension  $M$  ;

$t = \{t_1, t_2, \dots, t_i, \dots, t_M\}$  Vecteur cible du réseau de neurone de dimension  $M$  ;

$W_{ji}^h$  Poids synaptique connectant le nœud  $j$  dans la  $h^{i\text{ème}}$  couche cachée au nœud  $i$  de la couche cachée précédente ou la couche d'entrées ;

$W_{kj}^h$  Poids synaptique connectant le nœud  $k$  dans la couche de sortie au nœud  $j$  de la dernière couche cachée ;

$\theta_{pj}^h$  Valeur du seuil du  $p^{i\text{ème}}$  vecteur de la base d'apprentissage,  $j^{i\text{ème}}$  nœud de la  $h^{i\text{ème}}$  couche cachée ;

$\theta_{kj}^o$  Valeur du seuil du  $p^{i\text{ème}}$  vecteur de la base d'apprentissage,  $k^{i\text{ème}}$  nœud de la couche de sortie.

Le fonctionnement des réseaux de neurones à retropropagation peut être divisé en deux étapes : introduction des entrées et la rétro-propagation de l'erreur. Durant ce processus, les poids synaptiques sont constantes, à la fin de la première itération, le vecteur de sortie est comparée au vecteur voulu, et une valeur d'erreur est calculée pour chaque nœud de sortie. Ce signal d'erreur est propagée dans le sens inverse de la couche de sortie aux nœuds de la couche cachée adjacente, chaque neurone reçoit une part de cette erreur selon sa contribution au vecteur de sortie c'est à dire selon son poids synaptique. Ce processus est répétée à chaque itération, couche par couche, où les poids synaptiques vont être mis à jour, jusqu'à se que le réseau converge vers un état stable.

### 1.5.3. Apprentissage

L'apprentissage est supervisé : on associe une configuration d'entrée à une configuration de sortie. L'algorithme de la retro propagation est utilisé pour l'apprentissage du réseau, cet algorithme est un gradient itératif conçu pour minimiser un critère quadratique d'erreur entre la sortie obtenue et la sortie désirée. Cette minimisation est réalisée par une configuration des poids adéquate. L'erreur ( $e$ ) est la différence entre la valeur désirée ( $t$ ) pour le neurone de sortie et sa valeur calculée par propagation ( $o$ ).

### 1.5.3.1. Introduction des données

Lorsque le  $p^{ième}$  exemple de la base d'apprentissage est fourni à la couche d'entrées, la valeur du nœud  $j$  de la couche cachée est :

$$net_{pj}^h = \sum_{i=1}^N w_{ji}^h x_i - \theta_{pj}^h$$

la sortie de ce nœud est

$$\hat{o}_{pj}^h = f_j^h(net_{pj}^h)$$

de façon similaire, la sortie du nœud  $k$  de la couche de sortie est :

$$net_{pk}^o = \sum_{j=1}^M w_{kj}^h \hat{o}_{pj}^h - \theta_{pk}^o$$

$$\hat{o}_{pk}^o = f_k^o(net_{pk}^o)$$

où  $f_j^h$  et  $f_k^o$  représentent les fonctions de transfert du  $j^{ième}$  nœud de la couche cachée et nœud  $k$  de la couche de sortie. Ces fonctions doivent être continues et différentiable.

### 1.5.3.2. Calcul de l'erreur

L'erreur commise entre le vecteur de sortie du réseau de neurones et le vecteur désiré est calculée comme suit :

$$\delta_{pk} = t_{pk} - o_{pk}$$

$$E_p = \frac{1}{2} \sum_{k=1}^M \delta_{pk}^2 = \frac{1}{2} \sum_{k=1}^M (t_{pk} - o_{pk})^2$$

L'algorithme de la rétro propagation applique une correction  $\Delta W(p)$  aux poids synaptiques, qui est proportionnelle au gradient  $\partial E_p / \partial W(p)$ , selon l'équation suivante :

$$W(p+1) = W(p) + \Delta W(p) = W(p) - \mu \frac{\partial E_p}{\partial W(p)}$$

où  $0 < \mu < 1$  est un paramètre d'apprentissage. Puisque le gradient  $\partial E_p / \partial W(p)$  est différent entre les nœuds de la couche de sortie et la couche cachée, la formule de correction des poids synaptiques sera différente.

### 1.5.3.3. Modification des poids synaptiques liants la couche de sortie

Le gradient de  $E_p$  par rapport à  $w_{kj}^o$  est calculé comme suit :

$$\begin{aligned}\frac{\partial E_p}{\partial w_{kj}^o} &= -(t_{pk} - o_{pk}) \frac{\partial f_k^o}{\partial net_{pk}^o} \frac{\partial net_{pk}^o}{\partial w_{kj}^o} \\ &= -(t_{pk} - o_{pk}) \hat{\delta}_{pj}^h \frac{\partial f_k^o}{\partial net_{pk}^o}\end{aligned}$$

on posons

$$\delta_{pk}^o = (t_{pk} - o_{pk}) \frac{\partial f_k^o}{\partial net_{pk}^o}$$

Comme le gradient local du nœud  $k$ . Les valeurs des poids synaptiques des nœuds connectés avec la couche de sortie sont :

$$w_{kj}^o(p+1) = w_{kj}^o(p) + \Delta w_{kj}^o = w_{kj}^o(p) + \mu \delta_{pk}^o \hat{\delta}_{pj}^h$$

### 1.5.3.4. Modification des poids synaptiques liants la couche cachée

L'erreur totale,  $E_p$ , est reliée aux poids synaptiques liés à la couche cachée par la formule :

$$\begin{aligned}E_p &= \frac{1}{2} \sum_{k=1}^M \delta_{pk}^2 = \frac{1}{2} \sum_{k=1}^M (t_{pk} - o_{pk})^2 \\ &= \frac{1}{2} \sum_{k=1}^M \left( t_{pk} - f_k^o \left( \sum_{j=1}^M w_{kj}^o \hat{\delta}_{pj}^h - \theta_{pk}^o \right) \right)^2\end{aligned}$$

Le gradient de  $E_p$  par rapport aux poids synaptiques liés à la couche cachée est

$$\begin{aligned}\frac{\partial E_p}{\partial w_{ji}^h} &= \frac{1}{2} \sum_k \frac{\partial}{\partial w_{ji}^h} (t_{pk} - o_{pk})^2 \\ &= - \sum_k (t_{pk} - o_{pk}) \frac{\partial f_k^o}{\partial net_{pk}^o} \frac{\partial net_{pk}^o}{\partial \hat{\delta}_{pj}^h} \frac{\partial \hat{\delta}_{pj}^h}{\partial net_{pj}^h} \frac{\partial net_{pj}^h}{\partial w_{ji}^h} \\ &= - \sum_k (t_{pk} - o_{pk}) w_{kj}^o x_i \frac{\partial f_k^o}{\partial net_{pk}^o} \frac{\partial f_j^h}{\partial net_{pj}^h}\end{aligned}$$

$$= -\frac{\partial f_j^h}{\partial net_{pj}^h} x_i \sum_k \delta_{pk}^o w_{kj}^o$$

On pose

$$\delta_{pj}^h = \frac{\partial f_j^h}{\partial net_{pj}^h} \sum_k \delta_{pk}^o w_{kj}^o$$

Le gradient local du nœud  $j$  de la couche cachée. Donc,

$$w_{ji}^h(p+1) = w_{ji}^h(p) + \Delta w_{ji}^h = w_{ji}^h(p) + \mu \delta_{pj}^h x_i$$

Lorsque la fonction de transfert est une fonction sigmoïde binaire de dérivé  $f' = f(1-f)$ , le gradient local d'un nœud de la couche cachée et la couche de sortie sont écrits respectivement comme suit

$$\delta_{pj}^h = \frac{\partial f_j^h}{\partial net_{pj}^h} \sum_k \delta_{pk}^o w_{kj}^o = f_k^o (1 - f_k^o) \sum_k \delta_{pk}^o w_{kj}^o$$

$$\delta_{pj}^o = (t_{pk} - o_{pk}) \frac{\partial f_k^o}{\partial net_{pk}^o} = (t_{pk} - o_{pk}) f_k^o (1 - f_k^o)$$

Il est à noter que les termes d'erreurs pour les unités cachées doivent être calculée avant la mise à jour des poids synaptiques des unités de sorties.

#### 1.5.3.5. Processus d'apprentissage direct et indirect

Deux type de procédé d'apprentissage sont classée selon la manière de présentation des vecteurs de données : le procédé d'apprentissage directe et le procédé d'apprentissage indirecte. Pour le premier procédé, les poids synaptiques des couches sont mies à jour chaque fois qu'un vecteur d'entrées est présenté au réseau, par contre, pour le deuxième procédé, les poids synaptiques des couches sont mies à jour après que tout les vecteurs sont présentés au réseau. Dans ce dernier cas, l'erreur calculée est la somme des erreurs pour chaque présentation de vecteur d'entrées.

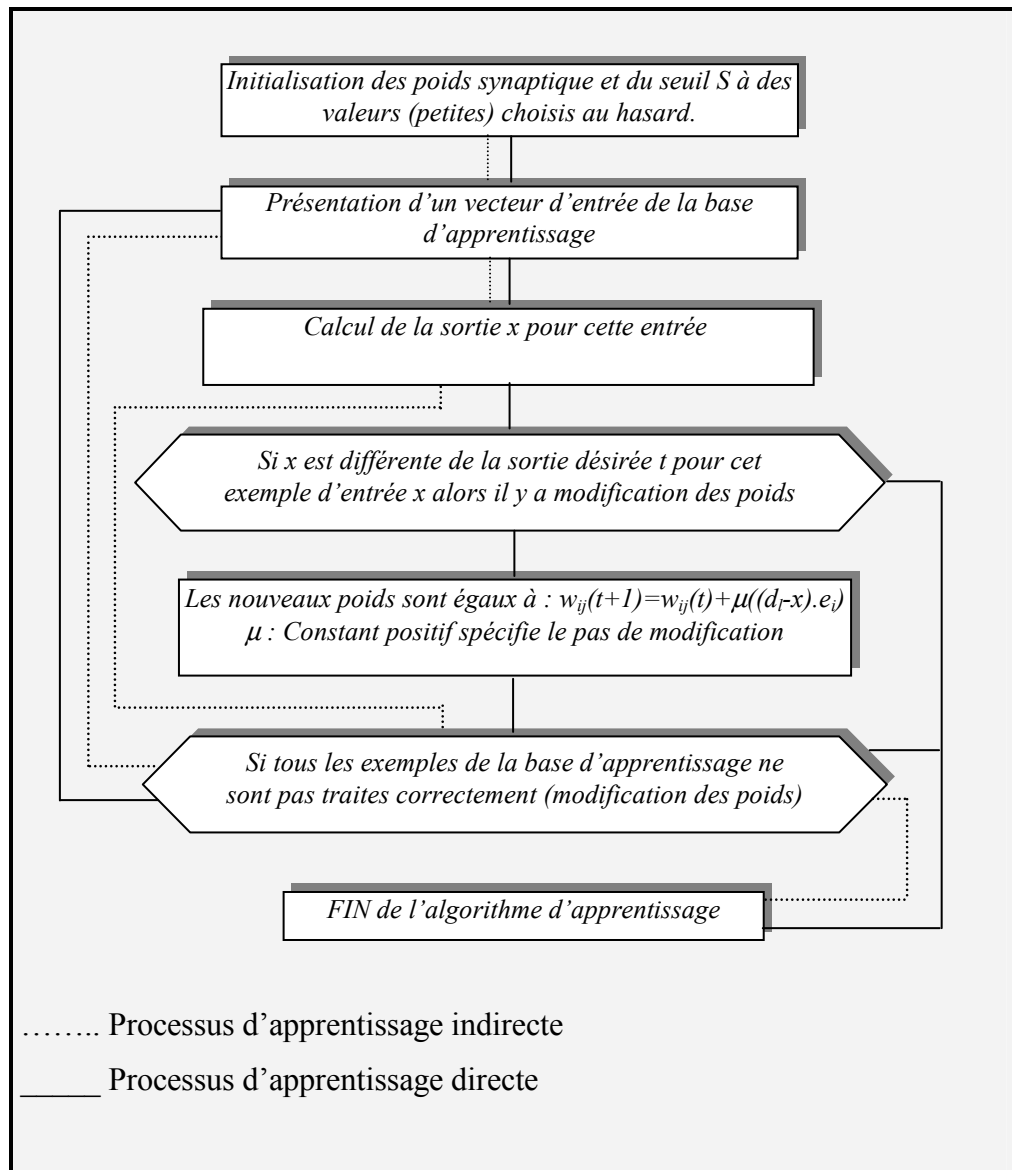


Figure 1.11 : Processus d'apprentissage direct et indirect [6]

#### 1.5.4. Considérations pratiques

Plusieurs facteurs influent la performance du réseau de neurones qui est décrite généralement par la vitesse d'apprentissage et la capacité de généralisation. Dans ce qui suit plusieurs facteurs sont discutés en détails :

### 1.5.4.1. Base d'exemple

On ne dispose pas de critères valables pour déterminer le nombre exact de vecteurs dans la base d'apprentissage. Cependant, certaines règles peuvent nous guider pour cette sélection :

- Puisque un réseau de neurones ne peut pas faire des extrapolations, donc la base d'exemple doit au moins englober les limites du problème traité selon toutes les dimensions.
- Les vecteurs de la base d'exemple doivent être distribués d'une manière uniforme dans le domaine du problème posé, si cela est impossible, l'apprentissage tend à se concentrer aux régions où ces exemples sont concentrés
- La progression de l'apprentissage peut être affaiblie si le processus d'apprentissage rencontre une région qui est sensible à certaines paramètres et peut être sensible aux autres. Dans ces cas, l'apprentissage va être rapide selon les paramètres sensibles, et ralenti selon les autres paramètres. La normalisation de la base d'exemple en assurant les proportions dimensionnelles du problème peut enlever ces difficultés.

Le nombre et la distribution des exemples pour l'apprentissage affectent la capacité de généralisation du réseau de neurones, qui devient efficace avec l'extension de la base d'exemples. Cependant, un réseau de neurones qui apprend trop de relation entrée/sortie va mémoriser tous les points, modère la généralisation dans les points intermédiaires.

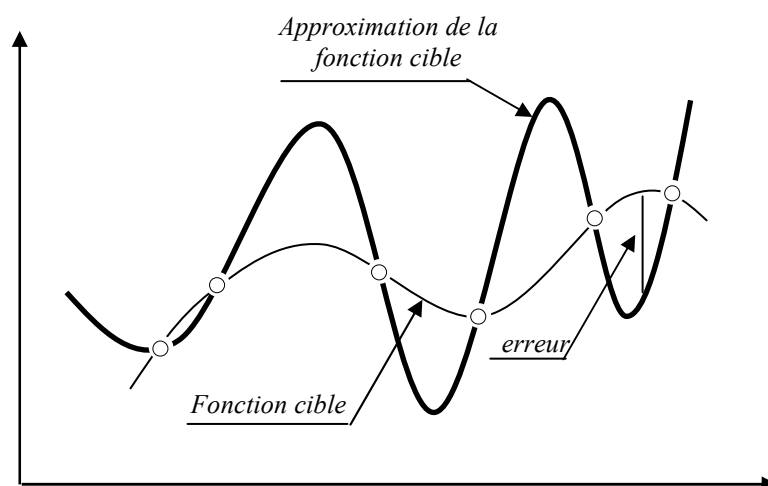


Figure 1.12 : Distribution uniforme des données de la base d'exemple [35]

#### 1.5.4.2. Nombre de couches cachées et de neurones cachés

La dimension de la couche d'entrée et de sortie est naturellement déterminée selon les paramètres du problème. Ce pendant, le nombre de couches cachées et le nombre de neurones ces couches n'est pas évident. On ne dispose pas de règles conte à leurs valeurs exactes.

Flood (1994) suggère que deux couches cachées assurent une grand flexibilité même pour modéliser des problèmes complexes, sans négligé que même avec une seule couche cachée on peut arrivée à un degré de performance semblable. Il y à deux exceptions à cette règle, la première, sont les réseau de neurones avec un seul neurone à la couche d'entrée, et la deuxième, sont les réseaux de neurones dont les entrées sont des valeurs binaires, pour ces deux cas, il est sure qu'avec une seul couche cachée la convergence est assurée.

Un dilemme existe conte à la détermination du nombre de neurones cachés. Un grand nombre de neurones cachés peut ralentir le fonctionnement du réseau de neurone, que se soit au cours de l'apprentissage ou en phase d'utilisation. D'un autre coté, un nombre réduit de neurones cachés ne peut pas englobée toute la surface des solutions.

De plus, Hajela (1991) suggère que le nombre de neurones dans les couches cachées soit entre la moyenne entre les neurones d'entrées et de sorties et la somme de ces neurones.

#### 1.5.4.3. Poids synaptiques et paramètres d'apprentissage

L'efficacité de l'apprentissage pour les réseaux de neurones à retropropagation et qualifier selon deux paramètre : la vitesse d'exécution, et éviter les zones de minimum local. La vitesse d'exécution est mesurée soit par le temps d'exécution de l'algorithme d'apprentissage ou bien le nombre d'itération envers la convergence selon l'erreur tolérée. La méthode du gradient descendant peut accidentellement tombée sur des minimaux locales au lieu du minimum globale, cela arrêt le gradient de recherche et gel la mise à jour des poids synaptiques. La Figure 1.13 illustre cette idée.



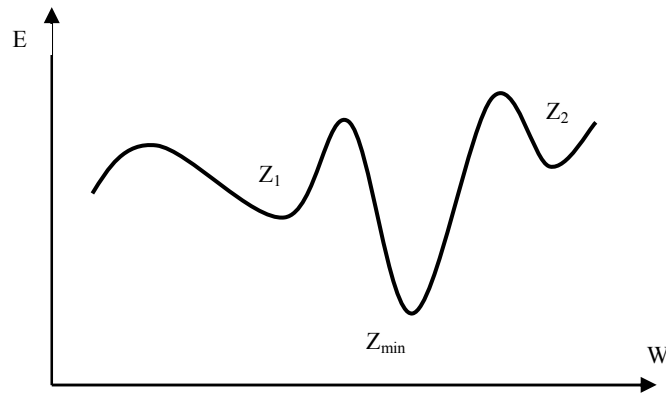


Figure 1.13 : Minimum globale et les minimums locaux

Le paramètre d'apprentissage  $\eta$  affecte considérablement les deux facteurs déjà mentionnés. Une valeur élevée de  $\eta$  peut omettre le minimum local. Toutefois, une valeur exagérée de  $\eta$  peut échouer la convergence. De plus, une valeur très réduite peut diminuer l'efficacité de l'apprentissage.

Les poids synaptiques sont généralement initialisés à des petites valeurs aléatoires, car si ces valeurs sont très grandes, le réseau de neurones peut devenir paralysé. Ce phénomène est observé lorsque les valeurs désirées correspondent à des valeurs de poids grandes, la dérivation des fonctions de transfert tend vers zéro, et par conséquent la variation des poids tend aussi vers zéro. L'apprentissage du réseau dans ce cas tend vers un point d'arrêt.

## 1.6. Les réseaux de neurones à contre propagation

### 1.6.1. Définition

Les réseaux de neurones à contre propagation, proposée pour la première fois par Hecht-Nielsen (1983), sont une combinaison de la carte auto-organisatrice de Kohonen et la structure outstar de Grossberg.

### 1.6.2. Principe des réseaux de neurones à contre propagation

La topologie des réseaux de neurones à contre propagation est composée de trois couches primaires : couche d'entrées, couche compétitive ou de Kohonen et couche d'interpolation ou de Grossberg (Figure 1.14)

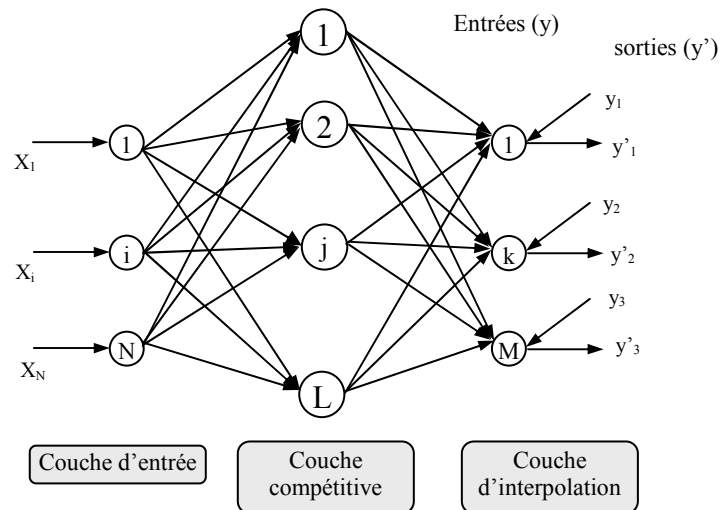


Figure 1.14 : Topologie des réseaux à contre propagation [35]

La construction des blocs du réseau à contre propagation est divisée en trois parties : la structure Instar, couche compétitive et la structure Outstar. Le mécanisme d'apprentissage et les modifications des poids synaptiques sont décrits dans ce qui suit.

#### 1.6.2.1. La structure Instar

Un élément Instar est un nœud unique comme montré sur la figure ci-dessous (Figure 1.15).

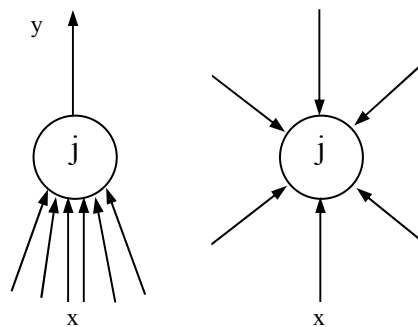


Figure 1.15 : Topologie d'un élément Instar

La modification du poids synaptique d'un nœud  $j$  peut être écrite comme suit :

$$\Delta w_{ji} = \alpha(x_{ji} - w_{ji})$$

$$w_{ji}(t+1) = w_{ji}(t) + \alpha[x_{ji} - w_{ji}(t)]$$

où  $\alpha < 1$  est un coefficient d'apprentissage. Cette modification de poids est représentée sur la figure suivante (Figure 1.16). Une fois l'apprentissage est achevé, l'élément Instar présente la valeur moyenne des entrées.

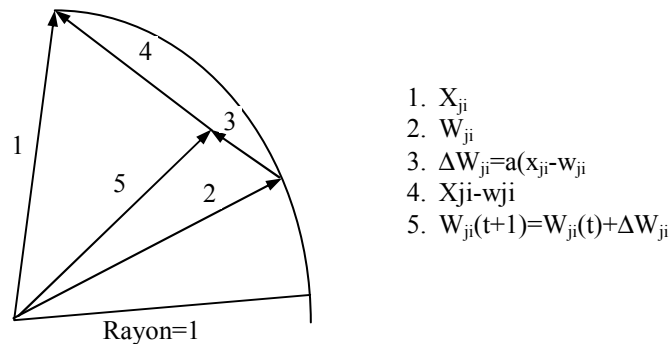


Figure 1.16 : Modification du poids synaptique pour une seule entrée [35]

#### 1.6.2.2. Réseau compétitive

La couche compétitive est composée de plusieurs éléments Instars. Le poids synaptique  $w_{ji}^c$  qui connecte un nœud  $j$  de la couche compétitive et un nœud  $i$  de la couche d'entrée, est ajusté selon la règle d'apprentissage de Kohonen (1988).

$$w_{ji}^c(t+1) = w_{ji}^c(t) + \alpha |x_{ji} - w_{ji}^c(t)| Z_j$$

où  $Z_j$  est la sortie du  $j^{\text{ème}}$  nœud gagnant de la couche compétitive, et est donnée par

$$Z_j = \begin{cases} 1.0 & \text{if } d_j < d_i \text{ pour tout } i \\ 0 & \text{cas contraire} \end{cases}$$

Lorsqu'un seul nœud gagne. Dans cette formule,  $d_j = \|W_j^c - X\|$ , où  $\|X\| = \sqrt{\sum_j x_j^2}$ , est la distance euclidienne entre le vecteur des poids  $W_j^c$  et le vecteur d'entrées  $X$ . Chaque nœud de la couche compétitive entre en compétition avec les autres nœuds, et le nœud avec la distance euclidienne la plus courte va gagné. Finalement, la sortie du nœud gagnant est 1 et la sortie des autres nœuds est 0.

### 1.6.2.3. La structure Outstar

La structure Outstar est composée de tous les nœuds de la couche d'interpolation ainsi que le nœud gagnant de la couche compétitive (Figure 1.17).

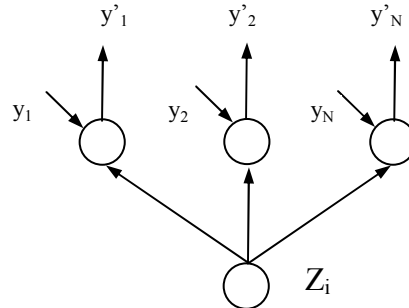


Figure 1.17 : Topologie de la structure Outstar [35]

Le poids synaptique  $w_{kj}^i$  connectant le nœud gagnant  $j$  de la couche compétitive ou le nœud  $k$  de la couche d'interpolation, est ajusté selon la règle d'apprentissage de Grossberg (1982)

$$w_{kj}^i(t+1) = w_{kj}^i(t) + \beta [y_k - w_{kj}^i(t)] Z_j$$

avec  $0 < \beta < 1$  et le coefficient d'apprentissage. La couche d'interpolation utilise comme fonction de transfert une fonction de sommation des poids, pour le  $k^{\text{ième}}$  élément du vecteur de sortie,  $y'_k$  est déterminée par

$$y'_k = \sum_j w_{kj}^i Z_j$$

Sommairement, les réseaux à contre propagation apprend en deux étapes. Premièrement, la couche de Kohonen est entraînée en utilisant des vecteurs sélectionnés de l'espace des entrées. La sortie de la couche de Kohonen est contrôlée de telle sorte qu'un seul élément est activé. Deuxième, une fonction d'approximation peut être implantée à travers le changement des poids de la couche de Grossberg par un apprentissage supervisée. Cette fonction d'approximation concerne seulement les neurones dans la sortie est différent de 0 de la couche de Kohonen. Cependant, comme les poids synaptiques entre la couche d'entrées et la couche de Kohonen sont entraînée pour converger aux valeurs

moyennes des entrées, les poids synaptiques entre la couche de Kohonen et de Grossberg sont aussi entraînée pour donner les valeurs moyen des sortie désirée.

### 1.6.3. Considération pratique

La sélection du nombre de nœud gagnant joue un rôle très important pour assurer la performance nécessaire du réseau à contre propagation. Diffèrent nombre de nœuds gagnants dans la couche de Kohonen implique différent degré de performance. Le nombre optimal de nœud gagnant est établi selon l'expérience de l'utilisateur.

L'initialisation des vecteurs des poids synaptiques de la couche de Kohonen se fait en assignant à chaque vecteur poids identiquement à l'un des vecteurs prototype pour chaque classe.

Freeman et Skapura suggèrent que le coefficient d'apprentissage  $\beta$  soit fixé initialement à une valeur très petite, et durant l'apprentissage, il peut être augmenté un peut.

## 1.7. Réseaux de neurones de Hopfield

### 1.7.1. Définition

L'apprentissage des réseaux de neurones à retropropagation et à contre propagation est un processus itératif pour trouver les poids synaptiques qui minimise l'erreur. Les poids synaptique sont donc fonction de temps d'un système dynamique.

Pour les réseaux de Hopfield, les poids synaptiques sont calculées d'avance et ne sont pas partie de ce système dynamique.

La topologie du réseau de Hopfield est schématisée sur la figure ci-après (Figure 1.18). Dans le réseau il n'y a pas de relation d'un nœud avec lui même et les poids synaptiques connectant d'autres nœuds sont symétriques.

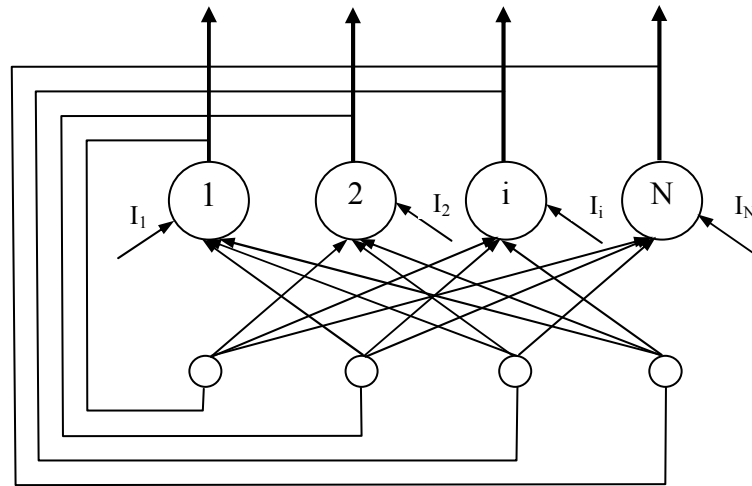


Figure 1.18: Topologie des réseaux de neurones de Hopfield [35]

Les réseaux de neurones de Hopfield sont classée en deux types : réseau de Hopfield discret et continue, dépendant de l'unité de sortie qui est une fonction discret ou continue des entrées.

### 1.7.2. Réseaux de neurones de Hopfield discrets

Le fonctionnement de ce réseau peut être décrit en deux étapes comme suit :

#### a) Stockage d'information :

Dans cette étape, le vecteur d'entrées binaire  $L$  présenté au réseau et les poids synaptiques peuvent être calculés d'avance comme suit :

$$w_{ij} = \begin{cases} \sum_{p=1}^L (2v_i^p - 1)(2v_j^p - 1) & \text{pour } i \neq j \\ 0 & \text{pour } i = j \end{cases}$$

#### b) Retrouvaille d'information :

Dans cette étape, lorsque un nouvelle entrée, qui pourrai être incomplet ou bruité, est présenté au réseau, la valeur d'entrée d'un nœud est donnée par :

$$u_i(t) = \sum_{j=1}^N w_{ij} v_j(t) + I_i$$

La valeur de sortie à l'instant  $t+1$  est donnée par :

$$v_i(t+1) = \begin{cases} 1 & \text{pour } u_i > U_i \\ v_i(t) & \text{pour } u_i = U_i \\ 0 & \text{pour } u_i < U_i \end{cases}$$

où  $U_i$  est la seuil. Dans les différents états stockés, le réseau cherche à déterminer la plus proche à l'entrée. Cette dynamique de recherche est conduite pour déterminer le minimum de la fonction d'énergie (Figure 1.19). Durant ce processus, les valeurs d'énergie vont être recalculées à partir de la valeur initiale jusqu'à l'arrivée au minimum le plus proche, tandis que les sorties du réseau change de l'état  $a$  à l'état  $b$ .

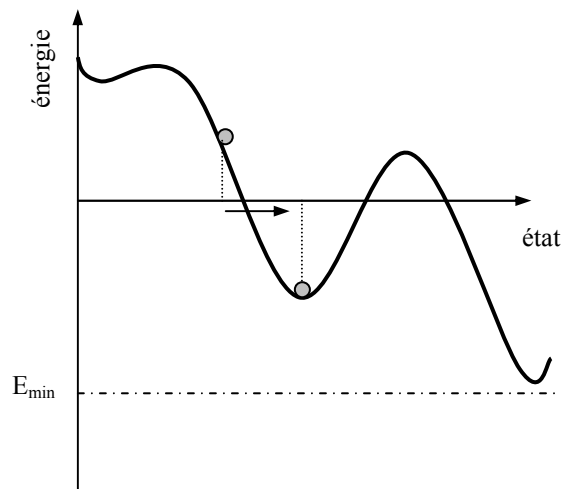


Figure 1.19 : Schémas d'une fonction d'énergie à deux dimensions [35]

La fonction d'énergie qui décrit l'état du réseau, est donnée, en terme des poids synaptiques et les entrées biais, comme suit :

$$E = -\frac{1}{2} \sum_j \sum_{j \neq i} v_i w_{ij} v_j - \sum_i I_i v_i + \sum_i U_i v_i$$

Cette fonction possède les caractéristiques suivantes :

1. tout changement de  $v$  durant le processus résulte une diminution de  $E$ .

2. la limite inférieure de  $E$  est  $E_{\min} = -\sum_i \sum_j |w_{ij}|$

Ces conditions assurent la stabilité du système.

### 1.7.3. Réseaux de neurones de Hopfield continus

La première caractéristique des réseaux de neurones de Hopfield continus est que leurs sorties sont une fonction continue des entrées au lieu des sorties binaires du modèle discret. La sortie d'un nœud est donnée par

$$v_i = f_i(u_i)$$

Dans le cas réel, entre deux neurones il y a un retard entre l'apparition des sorties du premier nœud et l'introduction de ces résultats comme données au nœud suivant. Ce retard est causé par la résistance et la capacité des membranes et la conductance finie de la synapse entre les cellules  $i$  et  $j$ .

La figure ci-dessus (Figure 1.20) montre l'idée de base du circuit du réseau de Hopfield. Sur cette figure le neurone est modélisé par un amplificateur avec une entrée résistance  $\rho_i$ , et une capacité  $C_i$ . La résistance entre la sortie du nœud  $j$  et l'entrée du nœud

$i$  est  $R_{ij} = \frac{1}{|T_{ij}|}$ , où  $T_{ij}$  est le poids synaptique.

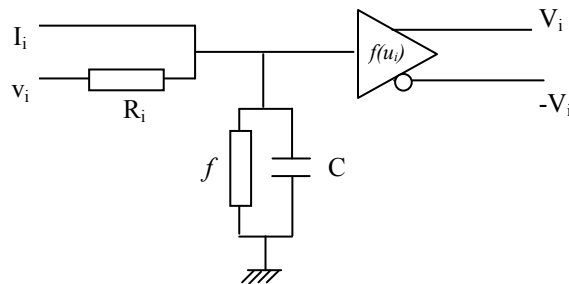


Figure 1.20 : Circuit analogique du réseau de Hopfield [35]

La fonction d'évolution du système s'écrit en termes de résistance et de capacité

$$C_i \left( \frac{du_i}{dt} \right) = \sum_{j=1}^N T_{ij} v_j - \frac{u_i}{R_i} + I_i$$

$$\text{Où } \frac{1}{R_i} = \sum_{j=1}^n \frac{1}{R_{ij}} + \frac{1}{\rho_i}$$

Pour une valeur initiale  $u_i(0)$  de chaque nœud, ces équations peuvent être résolues par des techniques numériques. La fonction d'énergie du système s'écrit

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N T_{ij} v_i v_j - \sum_{i=1}^N I_i v_i + \sum_{i=1}^N \frac{1}{R_i} \int_0^{v_i} g_i^{-1}(v) dv$$

Où  $g^{-1}(v) = u$  est l'inverse de la fonction  $v = g(u)$ .



### 1.8. Réseaux de neurones à résonance adaptive

La théorie de la résonance adaptive était mise au point par Grossberg et al (1987), ces réseaux de neurones sont composée de trois sous-systèmes : sous-système attentionnel, sous-système de control de gain et sous-système d'orientation (Figure 1.21).

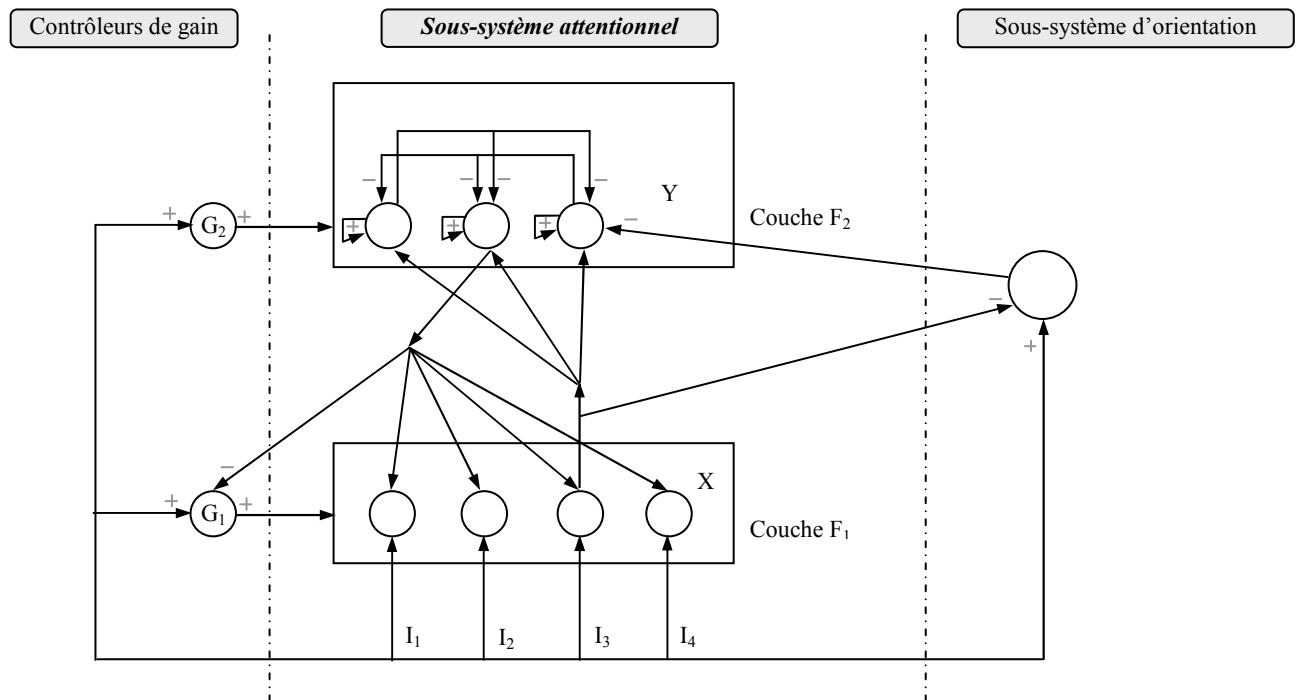


Figure 1.21 : Topologie des réseaux de neurones à résonance adaptive [35].

Le sous-système attentionnel est composé de la couche F<sub>1</sub> ou la couche de comparaison, et la couche F<sub>2</sub> ou la couche de réorganisation. Les entrées de la couche de comparaison sont:

1. les données d'apprentissage du réseau,  $I_i$ .
2. le contrôleur de gain  $G_1$

3. 
$$v_i = \sum_{j=1}^M u_j w_{ij}^d$$

Où :  $u_i$  est la sortie de la couche de réorganisation

$w_{ij}^d$  est le poids synaptique de F<sub>2</sub> vers F<sub>1</sub>

X est la fonction d'activation des neurones de la couche de comparaison.

Les entrées de la couche de réorganisation inclus:

$$1. \quad t_j = \sum_{i=1}^N s_i w_{ij}''$$

Où :  $s_i$  est la sortie de la couche de comparaison

$w_{ij}''$  est la poids synaptique de  $F_1$  vers  $F_2$

2. le contrôleur de gain  $G_2$

3. le vecteur d'entrée réinitialisé à partir du sous-système d'orientation

$Y$  est la fonction d'activation des neurones de la couche de réorganisation

Les neurones de la couche de réorganisation sont soumis à un mécanisme de ralentissement, dans le quel la sortie de chaque neurone est réentrée à ce même neurone avec un poids synaptique positif, et à tous les autres neurones avec un poids synaptique négatif. Seul un neurone va gagner et génère un vecteur sorti qui est comparée au vecteur d'entrée. Si ces deux vecteurs sont similaires, le vecteur d'entrée va être automatiquement classé. Faute de similitude, le processus va être bouclé.

La sortie du contrôleur de gain  $G_1$  est décidée selon les données d'apprentissage et la sortie de la couche de réorganisation.

$$G = \begin{cases} 1 & \text{si } I \neq 0 \text{ and } U = 0 \\ 0 & \text{si non} \end{cases}$$

Il joue un rôle central pour l'activation des neurones de la couche de comparaison.

Si le vecteur génère par la couche de réorganisation est différent du vecteur d'entrée, ce sous-système génère un vecteur réinitialisé, par le quel, le neurone déjà activé est désactivé dans la deuxième itération.

### 1.9. Conclusion

Ce chapitre constitue le background théorique nécessaire avant d'entamer l'aspect physique du problème, on a montre l'efficacité de l'approche neuronale, qui a la base constitue de nœuds élémentaires intervenant par des simple opérations arithmétiques, mais l'interconnexion de ces nœuds livre un réseau de neurones capable de modélisé des processus non linéaire assez complexe, difficilement attente par une approche algorithmique, voir impossible.

On a exposé aussi les étapes de conception d'un réseau de neurones (architecture, poids synaptique, entrées/sorties et l'algorithme d'apprentissage). En suite on à exhibé plus en détail quelque type de réseaux de neurones qui en trouvé leur applicabilité dans le domaine de la mécanique, ainsi que la pratique de leurs conception.

## **CHAPITRE 2**

### **APPLICATION DES RESEAUX DE NEURONES EN GENIE CIVIL**

#### 2.1. Introduction

La recherche connexionniste a pris une ampleur considérable dans la dernière décennie, à tel point qu'il est aujourd'hui illusoire d'espérer dominer le sujet entier. Un grand nombre d'auteurs utilisent le connexionniste pour résoudre des problèmes d'autres domaines, autant appliqués que théoriques.

Ainsi, une grande partie des travaux connexionnistes peuvent être qualifiés « d'applicatifs », car ils ont comme souci premier la résolution efficace d'un problème concret. Les applications les mieux résolues par des méthodes connexionnistes à l'heure actuelle sont des tâches dites « de bas niveau », où la méthode de résolution n'est pas connue ou difficile à formaliser, et où il n'est en général pas nécessaire d'effectuer des calculs séquentiels suivis. Dans ces domaines, les possibilités d'apprentissage automatiques, de généralisation, et de traitement d'informations incomplètes ou bruitées sont souvent citées comme les avantages prépondérants du connexionnisme.

#### 2.2. Principaux domaines d'application des réseaux de neurones

##### 2.2.1. La reconnaissance des formes

Dans ce problème, les données d'entrée représentent l'information recueillie par un ou plusieurs capteurs. Le but est de reconnaître le ou les objets perçus par les capteurs.

##### 2.2.2. La classification

Ce type d'application ressemble beaucoup à la reconnaissance des formes, en ce que son but est d'associer à une entrée donnée l'étiquette d'une forme connue. La différence réside dans la source des données.

### 2.2.3. La transformation

Plusieurs applications ont été proposées où le réseau doit apprendre une transformation mathématique entre l'entrée et la sortie. En traitement d'image, par exemple, il existe des modèles connexionnistes capables de calculer les contours d'objets et la compression de données.

### 2.2.4. La prédiction et le contrôle de processus

Le problème de la prédiction consiste à estimer l'état futur d'un processus à partir d'un historique de son comportement et des variables environnementales attenantes. En contrôle, il s'agit non seulement d'estimer passivement l'état futur, mais aussi d'intervenir de façon à atteindre un but donné. Dans ce cadre, un domaine où les réseaux ont un succès incontestable est dans le domaine de la finance.

D'une manière générale, une application potentielle qui peut être traitée par le concept connexionnisme est toute application qui se représente sous la forme d'une fonction réalisant une mise en correspondance de deux espaces, pour peu que l'on dispose d'exemple représentatif et en quantité suffisante du comportement de cette fonction. Il faut aussi spécifier le meilleur modèle de réseau susceptible de résoudre le problème et ses nombreux paramètres.

## 2.3. Application des réseaux de neurones en génie civil

La première application de la technique des réseaux de neurones remonte seulement à la fin des années 80 (Adeli, 1989). Ces applications incluent :

- L'établissement des paramètres d'entrées/sorties des relations non linéaires pour les matériaux et les structures.
- L'identification des dommages structurelle et le contrôle des structures vis-à-vis des charges dynamiques.
- Le design préliminaire des structures.
- L'analyse et optimisation de la conception des structures

- Ainsi que la dernière génération des systèmes experts associés avec la logique floue et le réseau de neurones pour l'établissement des plannings des projets, l'estimation des coûts, managements et maintenance des structures.

Ces applications ont profité de l'efficacité des réseaux de neurones pour résoudre des problèmes complexes en engineering.

### 2.3.1. Applications en analyse des structures

#### 2.3.1.1. Application des réseaux à rétro-propagation

L'application des réseaux de neurones à rétro-propagation en génie civil peut être classée selon deux (02) catégories. Due à l'efficacité de l'établissement de la capacité non linéaire avec des données incomplètes ou bruitées, les réseaux de neurones à rétro-propagation sont utilisés en analyse et conception des structures, comportement des matériaux et identification des dommages. Pour ce type d'applications, la base d'apprentissage est construite soit par analyse par éléments finis ou par exploitation directe des résultats expérimentaux. D'un autre côté, le caractère essai-erreur des réseaux de neurones à rétro-propagation conduit à prouver les techniques d'apprentissage du réseau. Dans le paragraphe qui suit, l'application des réseaux de neurones à rétro-propagation est décrite brièvement selon ces deux aspects.

Hjela et al (1991) a utilisé les réseaux de neurones à rétro-propagation pour représenter la relation force-déplacement en analyse statique des structures. Ces modèles de comportements fournissent des capacités efficaces pour l'analyse numérique, et apparaît qu'il est bien adapté pour l'optimisation et la conception des structures.

Jenkins (1995) a appliqué les réseaux de neurones en analyse approximative des structures surtout à titre de comparaison entre des systèmes structurels simples.

Chuang et al (1998) les a utilisées pour modéliser des relations non linéaires complexes entre les paramètres d'entrées associés à la capacité ultime d'un poteau en béton armé.

Messner (1994) a démontré qu'un système intelligent en réseau de neurones peut faire la décision pour le choix préliminaire des systèmes structurels.

Yeh (1999) présente une méthode d'optimisation des proportions d'éléments d'un béton à haute performance pour une ouvrabilité et une résistance définie d'avantage, utilisons les réseaux de neurones et la programmation non linéaire.

Kang et Yoon (1994) décrit la configuration et la technique d'apprentissage d'un réseau de neurones pour le dimensionnement des systèmes à treillis et exploitent le rôle possible des réseaux de neurones dans le problème de dimensionnement des structures.

Mukherjee et al (1996) a établie la relation entre le coefficient de flambement, le module d'élasticité longitudinale et la force critique de flambement d'un poteau. Puisque les paramètres d'entrées sont tirés directement de résultats expérimentaux, les facteurs affectant la force de flambement d'un poteau sont incorporées automatiquement dans le modèle.

Mukherjee et Deshpande (1995) ont développés un réseau de neurones pour le dimensionnement des poutres en béton armées à section rectangulaire.

Further, Ghaboussi et al. (1991) modélise le comportement du béton en contraintes planes sous chargement monotonique bi-axial et en compression mono-axiale cyclique. En outre, les réseaux de neurones en vu beaucoup d'application en domaine d'identification des dommages.

#### 2.3.1.2. Application des réseaux à contre propagation

Leur première application était élaborée par Adeli et Park (1995) pour l'analyse des structures. Le réseau établi comparé à un réseau à retro-propagation, à montrer la rapidité de convergence, c'est à dire un temps d'exécution réduit pour des résultats comparables. En plus, les réseaux à contre propagation on été appliquée pour deux exemples de dimensionnement des structures métalliques ou le réseau nécessite des milliers de connexions.

Hajela (1998) à utilisée ces réseau dans quelques application en dimensionnement des structures, incluant un outil d'analyse rapide pour un design optimale, et inversement l'établissement de la capacité dans les problèmes d'identification structurelle. L'utilisation de ces réseaux comme un outil d'analyse rapide à été testé pour l'optimisation des structures, qui on prouvée leur efficacité en donnant des résultats comparables, mais avec un temps d'exécution inférieure à 10% par rapport à l'optimisation par analyse exacte.

En plus de cette application, ils ont été utilisés pour l'évaluation des dommages structurels.

La réponse en déplacement statique sous l'effet d'un chargement monotonique a été utilisée pour évaluer la position et l'importance des dommages dans les éléments structurelles.

Arsian et Hajela (1997) ont décrit une approche où il est décomposé le problème d'optimisation structurelle en plusieurs niveaux, où les réseaux à contre-propagation ont été utilisés comme un outil de couplage entre ces différents niveaux du processus de conception. Cette approche a permis une représentation non linéaire bien précise de ce couplage, contrairement aux présentations linéaires basées sur la variation des paramètres du problème d'optimisation.

#### 2.3.1.3. Application des réseaux de Hopfield

Les réseaux de Hopfield ont aussi trouvés application pour les problèmes d'optimisation des structures. Les variables d'un problème d'optimisation sont représentés par l'état du réseau, et la fonction objective est considérée équivalente à la fonction d'énergie. L'évolution du réseau vers un niveau minimale d'énergie renvoie les valeurs optimales des variables de conception de la structure.

Hajela (1998) a appliquée les états binaires des RNH pour optimiser les systèmes de structures en treillis. Le principe d'optimisation se base sur la sélection de quelques éléments qui seront ensuite affectés à des positions particulières dans le but de réduire la section de ces éléments en assurant la stabilité nécessaire.

La valeur d'un variable de conception égale à l'unité quand l'élément est affecté à une position  $i$ , et zéro dans le cas contraire.

Zeng (1995) a traité le problème d'analyse des structures par les RNH continues où il a pris les coefficients de la matrice de rigidité comme étant les poids synaptiques, et les forces nodales des éléments comme les paramètres d'entrées du réseau. Ce réseau a été mis en place pour des systèmes en treillis à 4 barres, 7 barres, 15 barres ainsi que pour des structures en portiques auto-stables planes, et dans tout ces cas les résultats convergent à des états stables.

#### 2.3.1.4. Application des réseaux de neurones à résonance adaptative :

Hajela (1998) a appliquée ces réseaux dans les problèmes de conception des structures. Cette application est établit en deux processus distincts. La première contient une classe de structures où la géométrie de la structure est généralement connue et les charges et les conditions aux limites peuvent varier. Autre face du problème c'est où les charges et les conditions aux limites sont supposées donner, et l'objectif est d'optimiser la géométrie du système structurel. L'utilisation des réseaux à résonance adaptive peut être jugée comme très efficace lorsqu'on profite de leur capacité de mémorisation de leurs bases d'apprentissage, où lorsque on présente un nouveau exemple à traité il sera encore conservée à cette base.

L'application considérée par Hajela (1998) été d'optimisée la section d'une poutre chargée uniformément pour arrivée à un poids minimale. Le dimensionnement de la section se base complètement sur le moment de flexion en cette section, ou ce dernier est déterminé selon les conditions aux limites ainsi que le type de chargement appliqué. Le type de chargement appliquée et les conditions d'appuies qui sont très critiques sont représentée par une chaîne quatre chiffres binaires comme présenter dans le tableau suivant (Tableau 2.1) :

Tableau 2.1 : Représentation binaire des paramètres d'optimisation (Hajela, 1998) [35].

A	0	L'appui à droite est un appui simple
	1	L'appui à droite est un appui double
B	0	L'appui à gauche est un appui simple
	1	L'appui à gauche est un appui double
C	0	Le chargement appliqué est une force
	1	Le chargement appliqué est un moment
D	0	Le chargement appliqué est concentré
	1	Le chargement appliqué est uniforme

Dans cette application, l'influence du paramètre de vigilance des réseaux a été testée contre le degré de catégorisation. Il à été noté que, avec une limite supérieure de vigilance, des petites différences dans les vecteurs d'entrées sont traitées comme des paramètres très sensibles et donc groupées en différentes catégories. Une limite inférieure de vigilance résulte en la création de nombre réduit de catégories.



### 2.3.2. Application en géotechnique

Dans beaucoup de circonstances, notre compréhension actuelle du comportement des sols et des roches ne parvient définitivement pas à prévoir comment les terrains se comporteront réellement. Dans ces circonstances, le jugement de l'expert joue un rôle fondamental, c'est pour raison que les approches empiriques sont largement utilisées dans la conception géotechnique. Étant donné que les techniques d'intelligence artificielle (AI) peuvent manipuler la connaissance heuristique (principes de base) les modèles d'assortiments, de liaisons et rapprochements, même si elles sont incapables de résoudre un ensemble d'équations mathématiques, elles sont tout indiquées pour servir dans le domaine de la conception géotechnique.

Beaucoup de problèmes géotechniques ont connus l'application de la technique des réseaux de neurones en leurs études, entre autres, ils ont été développés pour la caractérisation des sites et la classification des sols et des roches et leur amélioration, l'étude des structures géotechniques, ainsi que l'évaluation des risques de liquéfaction. Zhou et Wu (1994) décrivent l'utilisation des réseaux neurologiques à cette fin. Leur système de réseau neurologique est employé pour caractériser la distribution spatiale des altitudes de rockhead.

Des applications semblables concernant la caractérisation des eaux souterraines sont décrites par Rizzo et Dougherty (1994) et Basheer et autres (1996). Basheer et al. expliquent comment des réseaux neurologiques peuvent être employés pour cartographier (tracer) la variation de la perméabilité afin d'identifier des frontières d'un remblai.

Agrawal et al. (1994) utilisent l'approche de réseau neurologique pour la prédiction de la cohésion et de l'angle de frottement pour l'argile limoneuse à partir de la densité sèche et de la teneur en eau.

Goh (1995c) a employé les réseaux de neurones pour modéliser des corrélations de sols. Le comportement en contrainte-déformation des sols a été également modélisé en utilisant les réseaux neurologiques. Penumadu et al. (1994) ont essayé de modéliser le comportement de contrainte-déformation des argiles, en incorporant le comportement à taux de chargement variable. Ellis et autres (1995) ont employé la distribution granulométrique et l'histoire des contraintes comme paramètres d'entrée afin de simuler des relations contrainte-déformation pour le sable. Ceci peut être comparé aux tentatives

semblables par Zhang et autres (1991), Millar et Clarici (1994) et Millar et Calderbank (1995) d'employer les réseaux neurologiques pour modéliser le comportement en contrainte-déformation de la roche.

TEH et al (1999) proposent trois modèles de réseaux à rétro-propagation dans l'estimation de la capacité statique des pieux.

NAWARI et al (1997) ont développé des modèles optimaux de réseau à rétro-propagation et à régression généralisés, en utilisant seulement des valeurs de SPT-N et les propriétés géométrique pour simuler le comportement des pieux axialement et latéralement chargés.

DERIN et HASAN (1998) propose des réseaux de neurones pour analyser le rapport complexe entre le sol et les paramètres sismiques dans l'analyse de liquéfaction.

### 2.3.3. Application en technologie des béton et matériaux

Dans ce domaine, plusieurs recherche et études concernant l'utilisation des systèmes de réseaux de neurones pour des applications en béton et matériaux ont été développées, dont la majorité se base sur le réseau multicouche à rétro-propagation. Ces application concernent: l'optimisation d'un mélange de béton, la prédiction des performances de l'auto-compaction d'un béton, la prédiction du degré d'hydratation de ciment, la prédiction des résistances à la compression des béton, avec et sans ajouts ou adjuvants chimiques la prédiction de durabilité du béton, la modélisation des résistances des béton a hautes performances ainsi que pour les béton cellulaires.

### 2.4. Conclusion

Ce chapitre montre le tracé des grands axes ou les réseaux de neurones on trouvés leurs applicabilité notamment dans les branches de génie civil.

Les réseaux de neurones n'ont affrontés le domaine de génie civil qu'au début des années 90. dans cette branche, et jusqu'à le temps actuel, les réseaux de neurones ont traité que des problèmes simples de calcul de structures, ou dans les problèmes d'optimisation des structures, et la le calcule neuronal intervient dans le processus comme une étape entre d'autres.

## CHAPITRE 3

### COMPORTEMENT NON LINEAIRE ET MECANISMES DE RUINE

#### 3.1. Introduction

ce chapitre présente les différentes approches analytiques utilisés en matière de dimensionnement des structures vis-à-vis d'une sollicitation sismique. L'attention ensuite est orientée vers les principes de l'analyse limite des structures est donc la détermination des mécanismes de ruine qui sont définie selon leurs conditions nécessaires et suffisantes de leurs apparition.

#### 3.2. Comportement non linéaire

##### 3.2.1. Non-linéarité mécanique

L'une des difficultés rencontrées dans une analyse non linéaire est la modélisation mathématique du comportement post-élastique des matériaux. Une approche simple du problème consiste à localiser la déformation non linéaire dans des points et idéaliser la courbe force-déformation du matériau par des modèles de comportement non linéaires.

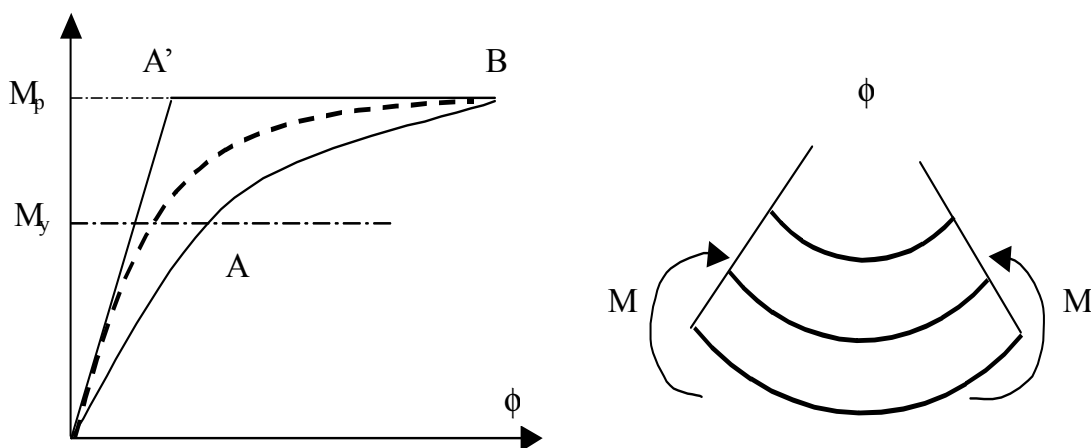


Figure 3.1 : Courbe de comportement non-linéaire

Pour une poutre simple, la courbe de moment-déformation pour une section homogène et de matériau isotrope est similaire à la courbe OAB (Figure 3.1). Si  $M < M_y$  la section se comporte élastiquement, et le tronçon OA est pratiquement une ligne droite. Au delà du point A le moment tend vers le moment plastique  $M_p$ , et la courbure  $f$  tend vers l'infinie.

Le rapport entre le moment plastique  $M_p$  et le moment élastique ultime est appelé le facteur de forme:

$$M_p/M_y = \nu$$

Le facteur  $\nu$  dépend de la forme de section (section IPE  $\nu=1.15$ , section rectangulaire  $\nu=1.15$ )

Cela est le fait que chaque type de section possède ces propres régions où le flux de plastification commence. Si l'on suppose que le facteur  $\nu=1$ , le diagramme de comportement devient élastique parfaitement plastique.

### 3.2.2. Non-linéarité géométrique

La petitesse des déformations constitue l'une des hypothèses de base de la résistance des matériaux. Ce principe n'est plus valable lorsque les éléments d'une structure subissent des déformations assez importantes ; provoquant une évolution des coefficients de rigidité géométrique.

Dans le domaine de calcul des structures on peut citer l'exemple de l'effet P- $\Delta$  : ce moment du second ordre résultant de l'excentrement des forces verticales entraîne des efforts supplémentaires qui doivent être repris par la structure.

### 3.3. Analyse dynamique non linéaire

Les méthodes d'analyses dynamiques linéaires élastiques supposent que le comportement des structures est linéaire, c'est à dire que leurs forces dynamiques résistantes étaient liées aux vecteurs d'accélération, de vitesse ou de déplacement par des coefficients d'influences linéaires.

L'équation fondamentale qui régit le mouvement vibratoire d'un oscillateur à plusieurs degrés de liberté s'écrit :

$$[M]\{\ddot{x}\}+[C]\{\dot{x}\}+[K]\{x\}=\{F(t)\}$$

$[M], [C], [K]$  : Respectivement la matrice de masse, d'amortissement et de rigidité et représentant les coefficients d'influences.

$\{\ddot{x}\}, \{\dot{x}\}, \{x\}$  : Respectivement le vecteur d'accélération, de vitesse et de déplacement.

Ce pendant, Il existe de nombreux cas où ces caractéristiques physiques ne peuvent pas être assimilées à des constantes lors de la réponse dynamique, Il est possible que les coefficients soit de masse, soit d'amortissement, ou de rigidité subit une évolution lors de la réponse dynamique. Toute variation de ce type modifie les caractéristiques des vibrations du système et le découplage des équations du mouvement ne sera plus valable.

La non-linéarité due à la variation de masse est d'une importance moindre par rapport aux autres sources de non-linéarité, de même La non-linéarité due à une variation de l'amortissement est souvent négligée, par conséquent leurs effets n'est pas considéré lors d'une analyse dynamique non linéaire.

La source considérable de non-linéarité des systèmes vibratoires est souvent due à la variation de la rigidité qui provient principalement des changements des propriétés géométriques ou mécaniques des éléments de la structure.

### 3.4. Analyse Pushover

Le comportement réel des structures soumises à une action sismique peut être adéquatement décrit par une analyse dynamique non linéaire. Cependant l'analyse dynamique non-linéaire est considérée comme une méthode rigoureuse pour le dimensionnement parasismique des structures. C'est pour cela qu'elle est généralement utilisée dans les études théoriques et non pas dans la pratique du dimensionnement.

L'analyse Pushover est considérée comme une méthode simplifiée pour évaluer la performance des structures parasismique

#### 3.4.1. Hypothèses de l'analyse Pushover

L'hypothèse fondamentale d'une analyse Pushover est que la réponse de la structure est principalement dominée par le premier mode. Beaucoup d'exemples illustrent que la réponse maximale de la structure, sous l'influence des modes supérieurs comparée au résultat d'une analyse dynamique temporelle. L'écart entre les résultats des deux méthodes devient de plus en plus important dans le cas des structures flexibles ; ou après la formation d'un mécanisme local, les propriétés dynamiques de la structure changent considérablement.

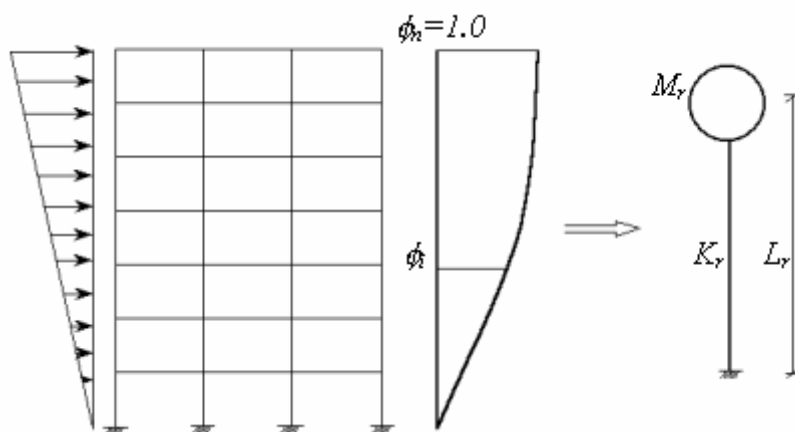


Figure 3.2 : Modélisation d'un système à pddl par un système à 1ddl

### 3.4.2. Principes de l'analyse Pushover

Une analyse Pushover nécessite que le modèle mathématique de la structure soit soumis à un système de force latéral. Ce système croît monotoniquement jusqu'à arrivé au déplacement ultime de la structure. Ce déplacement ultime de la structure est une estimation du déplacement maximale de la terrasse de structure durant un séisme majeur.

Les forces latérales sont appliquées pas à pas et les propriétés de rigidité de chaque élément de structure sont recalculées après chaque incrément. La relation entre l'effort tranchant à la base de la structure et le déplacement de la terrasse, appelé courbe de capacité, est le produit fondamental d'une analyse Pushover. Cette courbe caractérise la performance post-élastique globale de la structure. Les demandes en efforts et déformations de la structure sont calculées pour le déplacement prescrit de la terrasse et évalué ainsi les capacités acceptables en résistance et en déformations.

Les forces latérales appliquées au niveau de chaque étage durant une analyse Pushover peuvent avoir un effet significatif sur la distribution des activités de déformations plastiques. Ce système de forces latérales est admis pour représenter la distribution des forces d'inerties

qui se développe dans la structure durant un séisme. Les systèmes de forces latérales sont généralement classés en deux catégories : fixe ou variable.

Dans le cas d'un système de forces fixe, les forces d'inerties latérales sont supposées non changeable durant la réponse de la structure à une accélération de sol. Ce type de forces latérales ne tien pas compte du changement de la distribution des forces d'inertie qui s'impose quand la rigidité de la structure change durant le comportement non linéaire.

La forme générale d'une courbe de capacité est présentée sur la figure ci-dessous.

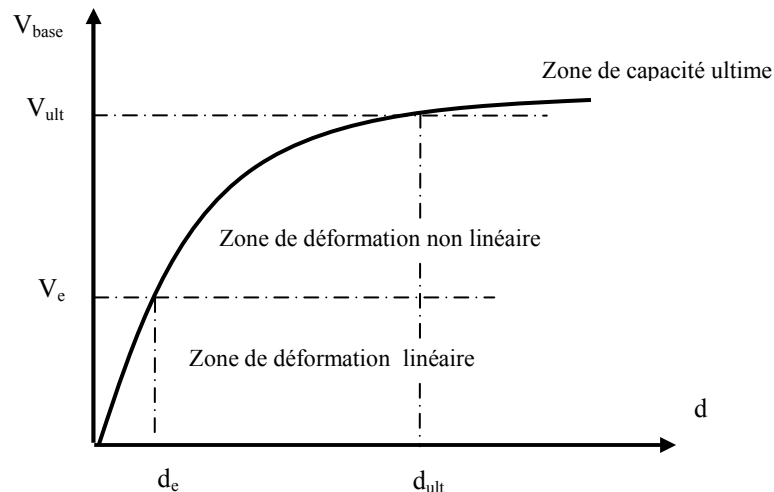


Figure 3.3 : Forme générale de la courbe de capacité d'une structure

### 3.5. Approche énergétique

Lors d'une excitation sismique, un échange d'énergie est effectué entre le sol et la structure, cet échange est traduit par le transfert d'une quantité d'énergie à la structure.

L'énergie reçue par la structure est évoluée en plusieurs types d'énergies : l'énergie cinétique qui cause le mouvement de la structure, l'énergie potentiel de déformation élastique des éléments, et l'énergie dissipée par amortissement et effet d'hystérésis.

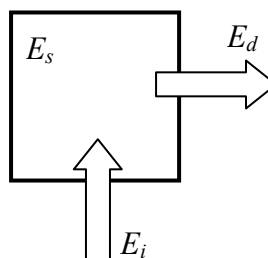


Figure 3.4 : Système physique interagie avec l'extérieur par un changement d'énergie [25].

$$E_i = E_s + E_d$$

où :

$E_i$  : l'énergie du séisme transmise à la structure par les fondations

$E_s$  : l'énergie emmagasiné dans la structure, peut être divisée en deux types :

$$E_s = E_e + E_k$$

où :  $E_e$  : énergie de déformation élastique

$E_k$  : énergie cinétique

$E_d$  : l'énergie dissipée par la structure. De même, elle peut être dissipé selon deux types :

$$E_d = E_h + E_v$$

où :  $E_h$  : énergie dissipée par effet d'hystérésis (déformation plastique)

$E_v$  : énergie dissipée par amortissement visqueux

On note que l'énergie est reliée à une force  $F$  qui dépend de la vitesse selon une loi du type :

$$F = c \times V^n$$

après c'est définition, on peut écrire l'équation générale suivante :

$$E_i = E_h + E_v + E_e + E_k$$

### 3.6. Comportement à la rupture

#### 3.6.1. notion de rotule plastique

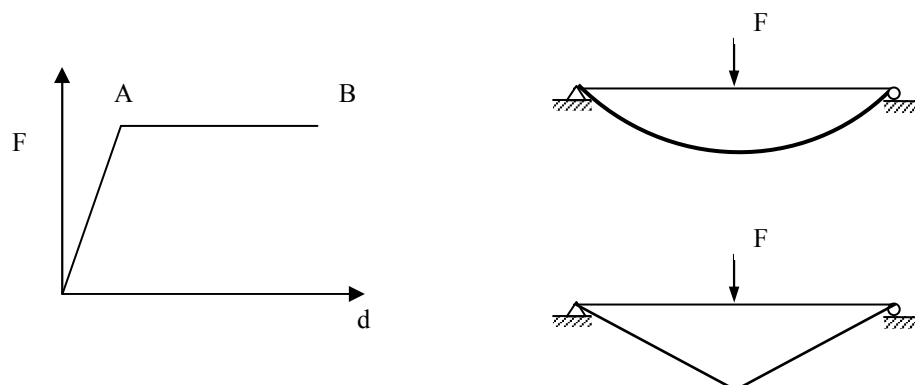


Figure 3.5 : Principe de formation d'un noeud plastique



Dans le cas d'une poutre simple comme représenté sur la figure [3.5], il est assez simple de décrire le comportement post-élastique, le moment de flexion sera en toute section moins que le moment plastique sauf à la section d'application de la charge, ou la déflexion augmente rapidement avec une stabilité du moment qui reste égale au moment plastique, en fait il y a formation d'une rotule plastique.

#### 4.6.2. Comportement limite d'un système structurel

Par illustration des exemples précédents de quelques cas simples, le comportement d'un système structurel peut être généralisé comme suit:

1. un élément structurel à comportement élastique parfaitement plastique, se comporte élastiquement jusqu'à apparition d'un nœud plastique à une section.
2. une force additionnelle peut être appliquée à ce stade si le nœud plastique formé permet la diffusion de cette force à d'autres parties stables du système structurel (redistribution des forces).
3. à chaque formation plastique, le moment reste constant est égale au moment plastique, contrairement à la déformation qui évolue avec la force appliquée.
4. le système structurel ruine lorsqu'il ne trouve pas d'éléments stables pouvant supporter un surplus de force
5. au moment de ruine, le système en entier ou une partie, forme un mécanisme.

#### 3.6.3. Définition d'un mécanisme de ruine

L'apparition d'une rotule plastique dans un système structurel, réduit le nombre de moments inconnus de un. D'où, si le nombre d'efforts inconnus est  $n$ , il faut produire  $n$  rotule pour que le système soit entièrement déterminé. L'apparition d'une autre rotule permet au système de se déplacer selon un degré de liberté. Donc un mécanisme se forme lorsque le nombre de rotule plastique est  $(n+1)$

Par exemple, le nombre d'efforts inconnus dans un système structurel à 2D est:

$(3m + r - 3j)$ , où:

$m$  est le nombre d'éléments structurels

$r$  est le nombre de réactions externes

$j$  est le nombre de joints

Le nombre de rotule formant un mécanisme est donc  $(3m + r - 3j + 1)$

Le cas d'une poutre continue sur plusieurs appuis, le degré d'inconnus est  $(2m + r - 2j)$  car les réactions horizontales sont nulles.

Ce critère applicable à chaque élément de même qu'au système en entier, car la ruine d'une partie du système entraîne en réalité le collapse totale.

#### 3.6.4. Conditions du collapse général

En définitive donc l'état de ruine d'un système structurel selon trois conditions nécessaires et suffisantes:

Lorsque la structure est juste au point du collapse:

a- Condition d'équilibre

Le système des moments de flexion doit être en équilibre avec les forces extérieures appliquées.

b- Condition du moment ultime

Le moment de flexion ne doit en aucune section dépasser le moment plastique de l'élément

c- Condition de mécanisme

Il doit y avoir suffisamment de nœuds plastiques pour former un mécanisme

Si le diagramme des moments de flexion satisfait ces trois conditions, le système définit donc l'état de ruine.

La rupture locale d'un élément structurel peut causer la rupture d'un autre élément du système, progressivement ces ruptures peuvent se propager résultant à la rupture d'une partie ou la totalité de la structure.

Chaque configuration de système structurel possède un niveau de sensibilité à la rupture progressive. Ces niveaux de sensibilité sont généralement négligés au stade du dimensionnement selon les approches en vigueur, qui mettent le point sur la résistance et la stabilité des éléments structuraux et non pas sur la configuration structurelle en entier, ou peu, sous forme de directive ou de conditions assez grossières.

A présent, les codes de dimensionnement ne sont pas aussi stricte sur ce point, et propose peut de règle évitant un mécanisme de ruine prématuré, et les ingénieurs en tendance à faire recours à leurs expériences.

### 3.6.5. Exemples de mécanismes de ruine

Les figures (figures 3.6, 3.7 et 3.8) présente quelques mode de ruine de structures affectées par le séisme de Boumerdes du 21 mai 2003. A part la qualité médiocre du béton et le non respect des détails et dispositions constructives, ces structures en était calcul selon un principe de résistance locale des éléments, mais le comportement limite et mode de ruine n'a pas été pris en compte dans la phase de calcul et qui nécessite un calcul non-linéaire.



Figure 3.6 : Structure totalement ruinée\*



Figure 3.7 : Mécanisme de ruine partiel\*



Figure 3.8 : Système structurel intacte\*

### 3.7. La redondance dans la conception parasismique

Selon le contexte d'étude et le problème posé, la redondance peut être définie de manières différentes. En résistance des matériaux la redondance d'une structure est équivalente au nombre de réactions (inconnues) en surplus des équations d'équilibre de la statique. Evidemment, cette définition ne révèle pas la capacité de la structure à préserver sa stabilité même avec un certains nombre d'éléments endommagés. En se basant sur ce concept, le règlement parasismique RPA99 [4], mesure la redondance par le nombre de files de contreventement parallèle à la direction de l'action sismique considérée.

Sous l'hypothèse d'une analyse pushover, Bertero et al [24], définissent explicitement la redondance en génie sismique comme étant le nombre de points (régions ou nœuds) critiques dont la défaillance (formation de nœuds plastiques) entraîne la ruine (formation d'un mécanisme) du système structurel. Par ailleurs, si une approche dynamique est sélectionnée pour l'analyse, le degré de redondance du système structurel est défini comme étant le nombre minimum de régions critiques ou de nœuds plastiques qui se rompent simultanément au moment de la ruine, et dans ce cas le degré de redondance ne dépend pas seulement des propriétés géométriques et mécaniques du système structurel, mais aussi du comportement dynamique de la structure, et de l'action sismique. Entre autre, l'apport essentiel de la redondance du système structurel et l'optimisation dans le dimensionnement des éléments, car en fait, un surdimensionnement même s'il assure la résistance, il pénalise la déformation plastique et la dissipation de l'énergie. Tout en sachant que la distribution de l'énergie dissipée en différentes régions évite la concentration des dommages.

Notons que malgré que les deux structures représentées sur la figure ci-dessous ont le même nombre de forces redondantes, elles n'ont pas le même degré de redondance lorsqu'elles sont sujettes à un système de forces latérales. Notons aussi que la redondance d'un même élément structurel peut être différente pour des dimensions, ferraillements et détails différents, et par conséquent le mécanisme de ruine est aussi différent.

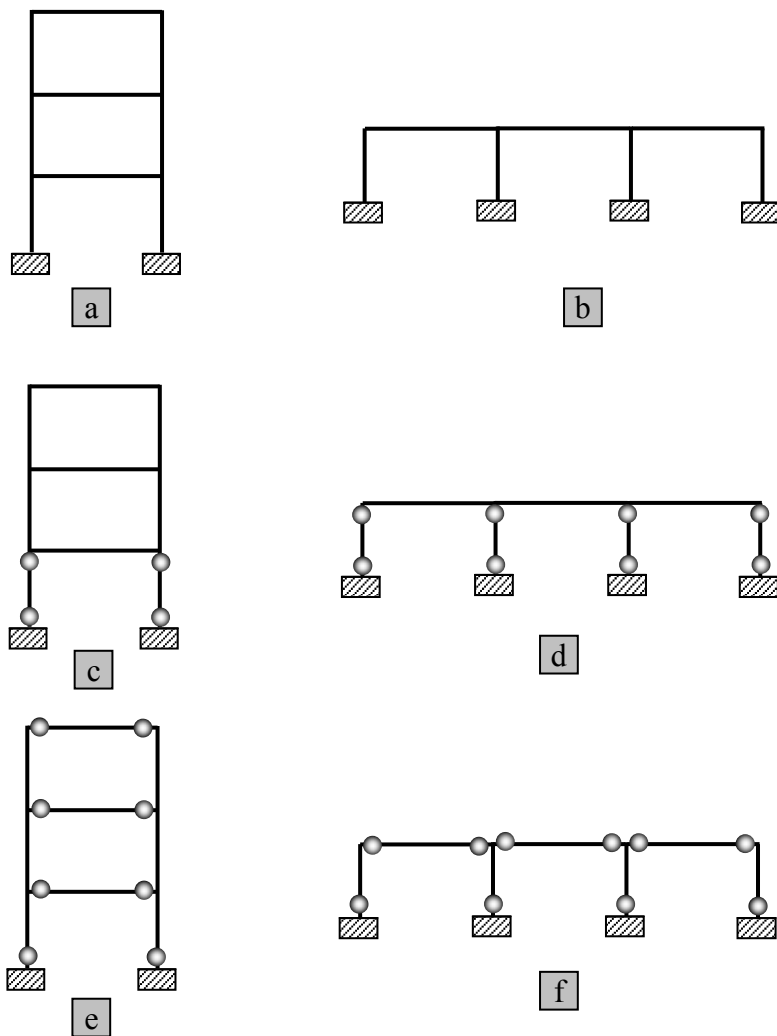


Figure 3.9 : détermination du degré de redondance sous chargement statique lateral

Coefficient de redondance  $n$  : (c)  $n=4$ , (d)  $n=8$ , (e)  $n=8$ , (f)  $n=10$ . [24]

## CHAPITRE 4

### PREDICTION DE LA DISTRIBUTION DES NŒUDS PLASTIQUES PAR RESEAU DE NEURONES

#### 4.1. Introduction

Dans cette étude, on établit un réseau de neurone capable de prédire la distribution des nœuds plastiques résultant d'une analyse Pushover juste avant le moment de rupture (Figure 5.1) c'est-à-dire le mécanisme dont il génère la ruine du système structural. Ce réseau est conçu pour des modèles de structures planes à 2 travées et 3 niveaux contreventées par des portiques auto-stables.

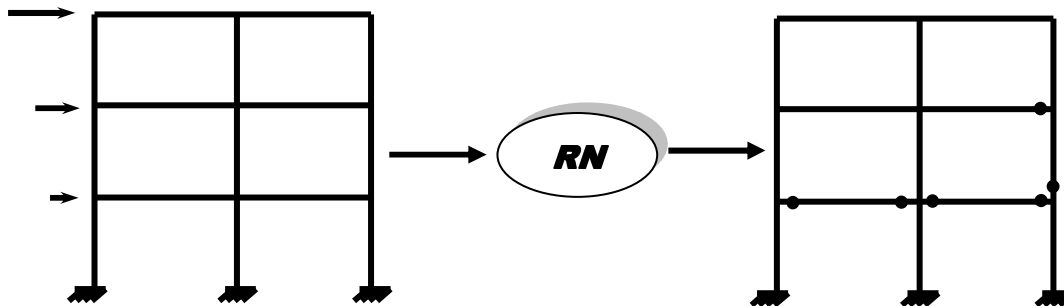


Figure 4.1 : Schéma du principe de fonctionnement du réseau de neurone.

#### 4.2. Base d'apprentissage

La construction de la base d'apprentissage est l'étape primaire et essentielle dans la conception du réseau de neurone. La base d'apprentissage pour cette application est constituée des résultats numériques d'une série d'analyses Pushover menées par le logiciel de calcul SAP2000 version Non-Linear.

L'analyse Pushover de ce système est menée en deux étapes : la première c'est l'application des forces gravitaires, la seconde c'est l'application progressive du système de force latérale jusqu'à la limite de la rupture. La loi de comportement est supposée élasto-plastique avec écrouissage.

Pour chaque configuration de structure il correspond une distribution des nœuds plastiques juste avant l'état de ruine, chaque nœud plastique est doté d'une couleur code représentant le niveau de plastification atteint, ces codes couleur sont ensuite codés numériquement.

### 4.3 Conception du réseau de neurones

#### 4.3.1 Architecture du réseau

L'architecture du réseau de neurone utilisé est le modèle multicouche à rétro-propagation avec une couche des entrées, une couche cachée et une couche de sorties. La **couche d'entrée** dont les variables sont les facteurs de rigidité flexionnelle des éléments résistants est constituée de 15 neurones, chaque neurone correspond au facteur de rigidité d'un seul élément. Les facteurs de rigidité élémentaire sont écrits sous leur forme la plus simple avec les paramètres physiques et géométriques, et sont extraits de la matrice de rigidité construite directement du modèle de la structure (Figure 4.2).

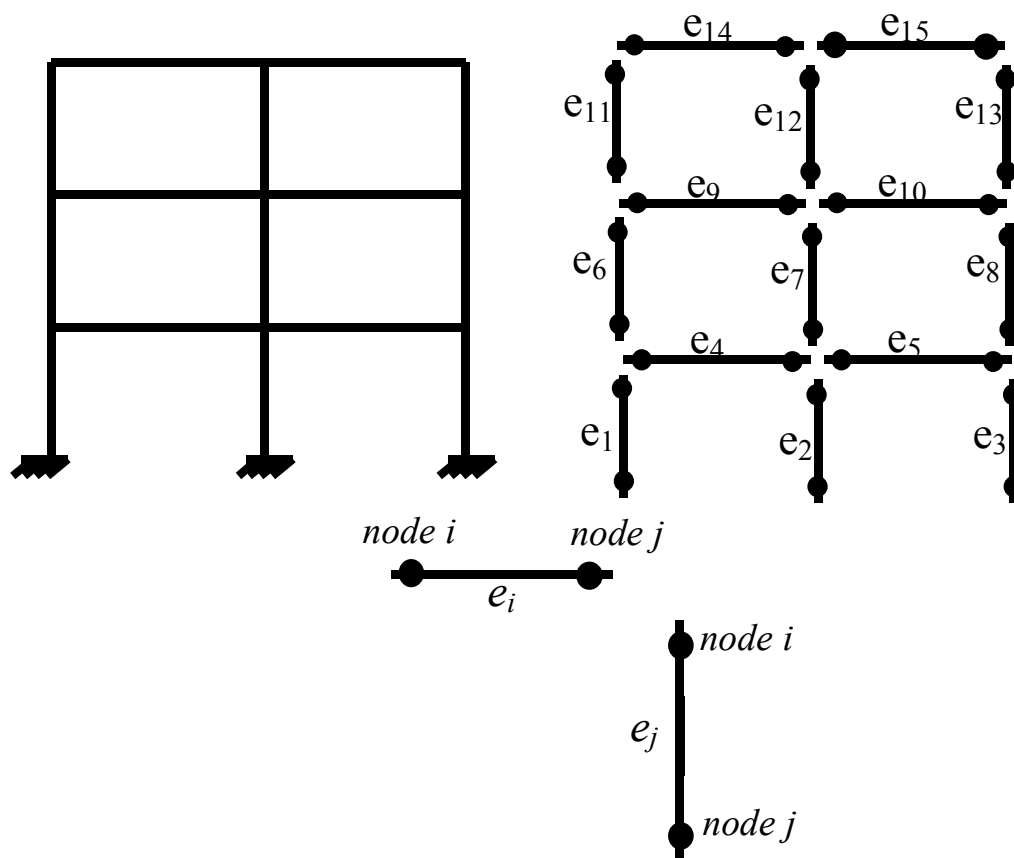


Figure 4.2 : Facteurs de rigidité élémentaires.



Matrice de rigidité extraite directement du modèle de structure:

$$\begin{bmatrix} k_{e14} & k_{e15} \\ k_{e11} & k_{e12} & k_{e13} \\ k_{e9} & k_{e10} \\ k_{e6} & k_{e7} & k_{e8} \\ k_{e4} & k_{e5} \\ k_{e1} & k_{e2} & k_{e3} \end{bmatrix}$$

$(k_{e1}, \dots, k_{e15})$

Facteurs de rigidité définie sous leur état la plus simple en sauvegardant le degré d'influence de chaque paramètre:

$$k_{ei} = \frac{E_{ei} \times I_{ei}}{l_{ei}^3} \quad (5)$$

$$k_{ej} = \frac{E_{ej} \times I_{ej}}{l_{ej}^3} \quad (6)$$

La **couche de sortie** du réseau donne la configuration des rotules plastiques dans la structure au moment de la ruine ainsi que le taux de plastification dans ces rotules, elle est constituée de 30 neurones donnant le taux de plastification des nœuds d'extrémités de chaque élément.

Les taux de plastification sont divisés en 05 niveaux selon le niveau de ductilité atteint (Figure 4.3) :

*Niveau initiale* : correspond au domaine élastique (pas de formation de rotule plastique).

*Niveau finale* : correspond à la rupture de l'élément où l'élément arrive à sa capacité maximale.

*Niveaux intermédiaires* : intervalle de l'activité plastique divisé davantage en trois niveau : plastification modéré, moyenne et importante.

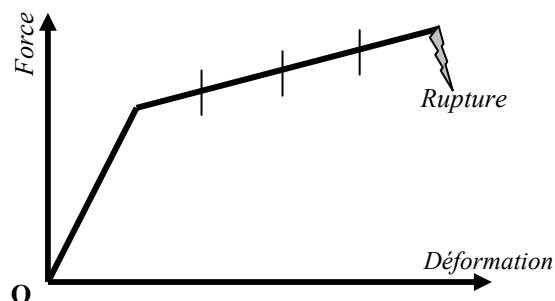


Figure 4.3 : Courbe force déformation et niveau de plastification d'un nœud plastique.

**Couche cachée** : Contrairement aux couches d'entrée et de sortie où leurs nombres de neurones est établi selon les paramètres définies, il n'existe pas de règles pour déterminer le nombre exact de nœuds dans la couche cachée, ou même le nombres de couches cachées à utiliser. Toutefois, le guide des utilisateurs de la boîte à outil des réseaux de neurones (MATLAB) suggère que les réseaux de neurones avec biais, ayant une fonction de transfert dans les nœuds de la couche cachée du type « sigmoïde », et dans la couche d'entrée une fonction de transfert du type « linéaire », sont capables d'imiter n'importe quelle fonction avec un nombre limité de discontinuités. Des travaux antérieurs suggèrent que le nombre de neurones dans les couches cachées soit entre la moitié des neurones d'entrées et de sorties et la somme de ces neurones. Néanmoins, une étude paramétrique du nombre de neurones cachés en fonction du nombre d'itération et de l'erreur commise est nécessaire.

#### 4.3.3. Codage numérique des données

##### 4.3.3.1. Codage des vecteurs d'entrées

Les valeurs numériques des entrées/sorties doivent être soigneusement choisies, et doivent dans la plupart des cas passer par une étape de codage, car l'utilisation des valeurs numériques sous leurs formes brutes peut générer le non apprentissage du réseau et la saturation des fonctions de transfert, donnant ainsi des résultats erronées.

Le codage ou la normalisation des paramètres consiste à transformer ces données (valeurs d'entrées et de sorties) en des valeurs qui peuvent être facilement traitées par les fonctions de transfert dans les neurones. Les intervalles les mieux adaptés sont :

$[-1, +1]$  ou  $[0, 1]$ .

Dans notre cas, la méthode adoptée pour le codage des facteurs de rigidité consiste à diviser ces facteurs par le facteur de rigidité le plus grand. L'intervalle des valeurs codées devient alors  $[0, 1]$ . Le vecteur des entrées s'écrit comme suit :

$$\left( 1, \frac{k_{e2}}{k_{e1}}, \frac{k_{e3}}{k_{e1}}, \dots, \frac{k_i}{k_{e1}}, \dots, \frac{k_{e15}}{k_{e1}} \right)$$

#### 4.3.3.2. Codage des vecteurs de sorties

Les taux de plastification sont normalisés dans l'intervalle  $[-0.5, 1]$ , la valeur  $-0.5$  correspond au domaine élastique, la valeur  $1$  à la limite de la rupture du noeud et les valeurs intermédiaires  $(0, 0.25, 0.75)$  aux 3 niveaux de plasticité.

La valeur  $0$  correspond au niveau de plastification initiale.

La valeur  $0.25$  correspond au niveau de plastification initiale.

La valeur  $0.75$  correspond au niveau de plastification initiale.

#### 4.4. Apprentissage du réseau de neurone

L'apprentissage utilisé est du type dit supervisé, où les couples entrées-sorties de la base d'apprentissage sont montrés au réseau d'une manière séquentiel. L'erreur commise (la différence entre la sortie désirée et celle donnée par le réseau) est rétro-propagée dans le réseau pour mettre à jour les poids synaptiques initiales. Il est parfois difficile de connaître l'algorithme d'apprentissage qui correspond le mieux au problème traité, car il dépend de plusieurs paramètres parmi les quelles la complexité du problème, la taille du vecteur d'entrée utilisé pour l'apprentissage, les poids synaptiques et le degré de performance voulu. Pour cet application, et après une étude comparative entre les différents algorithmes proposés par la "Neural Network Toolbox" du logiciel MATLAB, on a opté pour l'algorithme de Levenberg-Marquardt. La Figure 5.4 montre l'effet du nombre de neurones dans la couche cachée selon le nombre d'itération nécessaire à la convergence, selon la figure, 22 neurones dans la couche cachée correspondent au minimum de nombre d'itération.

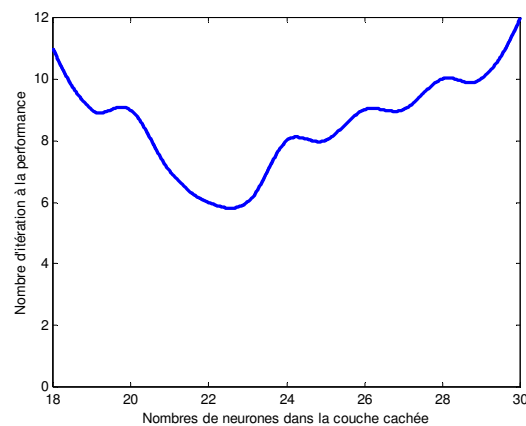


Figure 4.4 : Effet du nombre de neurones dans la couche cachée sur le nombre d'iteration (Algorithme de Levenberg-Marquardt)

La Figure 4.5 montre l'effet du nombre de neurones dans la couche cachée selon l'erreur commise par le réseau. Cette erreur est définie comme étant le maximum de la moyenne des valeur d'un vecteur de sortie. La figure montre qu'un réseau à 22 ou 28 neurones dans la couche cachée correspondent au minimum d'erreur commise. Un réseau à 22 neurones cachés est choisi donc pour l'architecture finale.

Dans notre cas, une architecture de réseau à deux couches cachées à l'inconvénient du temps d'apprentissage excessivement long.

Sur la figure, les couches cachées avec un nombre de neurones inférieur à 18 n'est pas présenté car la convergence n'est pas atteinte.

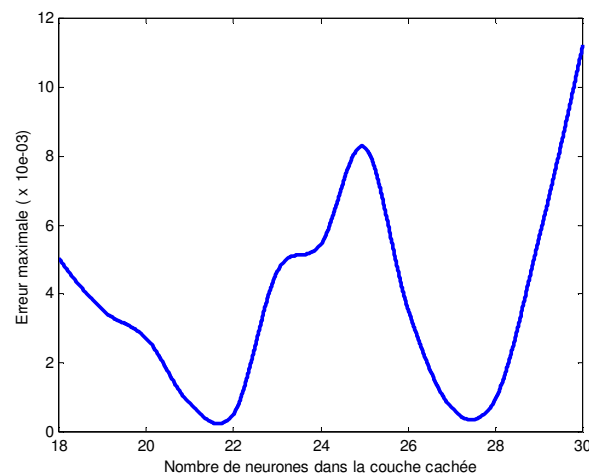


Figure 4.5 : Effet du nombre de neurones dans la couche cachée sur l'erreur maximale (Algorithme de Levenberg-Marquardt)

Le tableau ci-dessous (tableau 5.1) montre une comparaison entre différentes algorithmes d'apprentissages en terme de nombre d'itération maximale, temps nécessaire pour la convergence et degré de performance atteint. Dans ce cas on a fixé 1500 itération comme étant la limite du processus de convergence. Le seul algorithme doté d'une performance remarquable est celui de Levenberg-Marquardt.

Tableau 4.1 : Comparaison entre les algorithmes d'apprentissage

Algorithme	Technique	Nombre d'itération	temps	Performance
traindx	Variable Learning Rate	1500+	136.686	1.400E-02
trainrp	Rprop	1500+	134.784	5.621E-03
trainbfg	BFGS quasi-Newton	100+	1350.607	1.431E-02
trainscg	Scaled Conj. Grad.	1500+	160.341	1.357E-03
traincgf	Fletcher-Powell CG	1500+	163.966	3.560E-03
<b>trainlm</b>	<b>Levenberg-Marquardt</b>	<b>6</b>	<b>218.604</b>	<b>3.185E-08</b>
traincgp	Polak-Ribière CG	1500+	181.651	2.012E-03
traincgb	Powell-Beale CG	1500+	185.417	1.363E-03
trainoss	One-Step-Secant	1500+	166.700	4.691E-03

La courbe de performance de l'apprentissage représentée sur la Figure 5.6 montre l'efficacité de l'algorithme de Levenberg-Marquardt qui converge après 6 itérations.

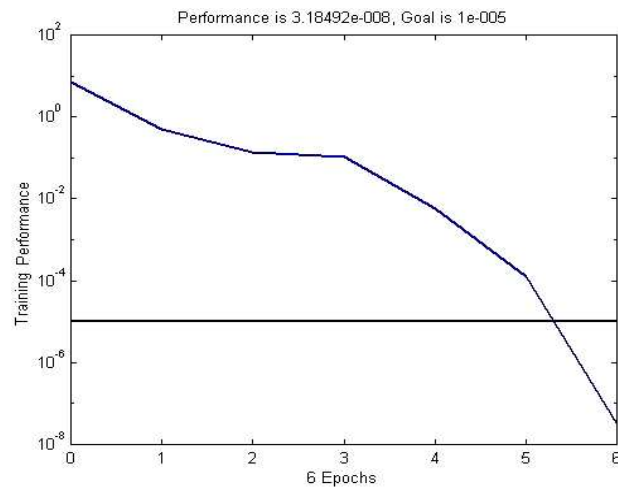


Figure 4.6 : Courbe de performance de l'apprentissage.

#### 4.5. Test, validation et généralisation

La phase d'apprentissage étant achevée, une série de testes de validation a été effectuée pour éprouver l'efficacité du réseau de neurone. Des modèles de structures autres que ceux déjà montrés au réseau pendant l'apprentissage ont été utilisés. A titre d'exemple, les Figures 5.7, 5.8, 5.9 et 5.10 illustre une comparaison entre une structure de la base d'exemples montrant la distribution réelle des nœuds plastiques avec leurs couleurs (nuances en gris) codes des niveaux de ductilité atteints (a), à droite on montre la distribution des nœuds plastique prédite par le réseau de neurones.

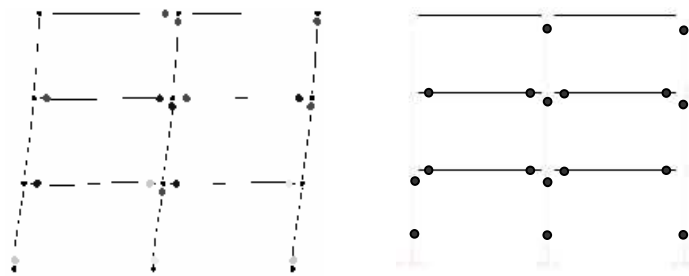


Figure 4.7 : Distribution des nœuds plastiques (exemple 1)

a) Modèle réel b) Prédiction par réseau de neurones

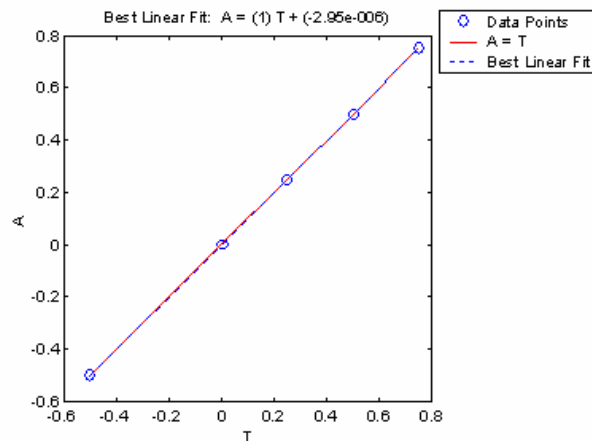


Figure 4.8 : Courbe de corrélation (exemple 1)

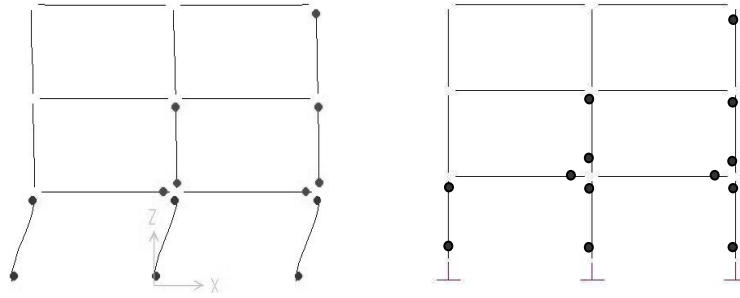


Figure 4.9 : Distribution des nœuds plastiques (exemple 2)

a) Modèle réel b) Prédiction par réseau de neurones

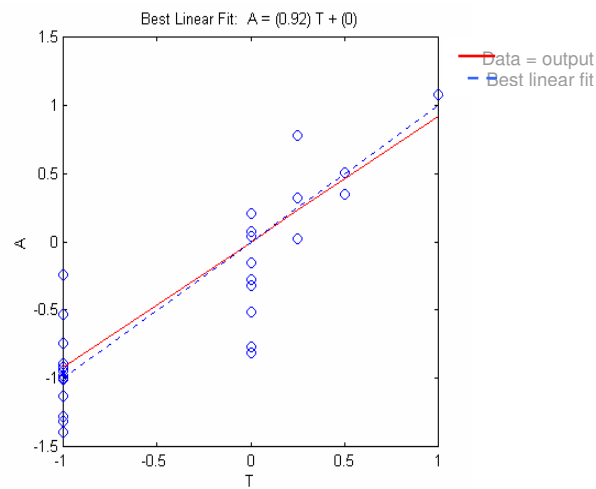


Figure 4.10 : Courbe de corrélation (exemple 2)

Une régression linéaire entre les résultats obtenus des analyses Pushover et les sorties du réseau de neurones a été effectuée. Le premier exemple présente une configuration favorable de l'activité plastique par apparition des nœuds plastique dans les poutres avec un niveau de ductilité élevé avant que la plastification apparaisse dans les poteaux. Par contre, le deuxième exemple est celui d'une distribution très défavorable, il y a apparition d'un mécanisme de ruine prématuré locale (effondrement du rez-de-chaussée).

Pour les deux exemples la distribution des nœuds plastique démontre une similitude exacte entre modèle réel et modèle prédit par le réseau. La prédiction des degrés de plastification est très bonne, avec un coefficient de corrélation de 0.99 pour le premier exemple et 0.85 pour le deuxième cas. Un coefficient de corrélation proche de l'unité, ce qui démontre une bonne corrélation entre les résultats du réseau et l'analyse Pushover. Cependant, comme le montre la Figure 5.7, le modèle manifeste des écarts plus importants pour les deux premiers niveaux de taux de plastification

(nœuds non plastifiés et nœuds légèrement plastifiés), tout en maintenant une erreur acceptable. Le tableau ci-dessous (tableau 5.2) exprime les rapports des écarts de prédiction des taux de plastification du deuxième exemple présenté.

Tableau 4.2 : Erreur dans la prédiction du degré de plastification pour chaque nœud

N° Nœud	Erreur	N° Nœud	Erreur	N° Nœud	Erreur
Nœud 1	<i>2.00E-04</i>	Nœud 11	<i>1.00E-04</i>	Nœud 21	<i>0.00</i>
Nœud 2	<i>1.00E-04</i>	Nœud 12	<i>2.00E-04</i>	Nœud 22	<i>2.00E-04</i>
Nœud 3	<i>1.00E-04</i>	Nœud 13	<i>2.00E-04</i>	Nœud 23	<i>2.00E-04</i>
Nœud 4	<i>2.00E-04</i>	Nœud 14	<i>2.00E-04</i>	Nœud 24	<i>3.00E-04</i>
Nœud 5	<i>0.00</i>	Nœud 15	<i>0.00</i>	Nœud 25	<i>1.00E-04</i>
Nœud 6	<i>2.00E-04</i>	Nœud 16	<i>0.00</i>	Nœud 26	<i>3.00E-04</i>
Nœud 7	<i>1.00E-04</i>	Nœud 17	<i>1.00E-04</i>	Nœud 27	<i>0.00</i>
Nœud 8	<i>1.00E-04</i>	Nœud 18	<i>2.00E-04</i>	Nœud 28	<i>2.00E-04</i>
Nœud 9	<i>4.00E-04</i>	Nœud 19	<i>2.00E-04</i>	Nœud 29	<i>2.00E-04</i>
Nœud 10	<i>2.00E-04</i>	Nœud 20	<i>2.00E-04</i>	Nœud 30	<i>0.00</i>

Il est clair que dans la pratique ces différences sont acceptables. Toutefois, il faut signaler que la généralisation de la méthode pour des systèmes structurels plus grands présente un inconvénient à cause de la taille du réseau de neurone qui devient très grande. Dans ce cas, le volume des entrées est très considérable ce qui conduit à des bases d'apprentissage importantes.



## CONCLUSION

Le risque tragique qui menace les structures sujettes aux sollicitations sismiques est sans doute la ruine totale ou partielle de la construction. Connaître le mécanisme de ruine d'une structure au stade de la conception est d'une utilité capitale. Ainsi, une technique de prédiction de la propagation des nœuds plastiques dans les portiques sous l'effet d'une sollicitation sismique a été établie.

Un réseau de neurone à rétro-propagation a été conçu pour déterminer la distribution des nœuds plastiques avec leurs taux de plastification au moment de la rupture. Les paramètres d'entrée du réseau sont les rigidités flexionnelles élémentaires. Vu que l'on ne possède aucune loi concrète sur les paramètres et l'architecture du réseau, une étude paramétrique est incontournable, cela est fait selon un critère de temps de convergence et d'erreur minimum. L'algorithme d'apprentissage joue le rôle primordial sur l'efficacité de l'apprentissage, et dans le cas de notre réseau l'algorithme de Levenberg-Marquardt est le plus efficace.

Ce réseau de neurone peut être utilisé comme un outil simple et rapide pour évaluer le comportement non linéaire du système structurel et permet au concepteur d'intervenir sur les caractéristiques géométriques de la structure pour optimiser la ductilité et la capacité de dissipation d'énergie de la structure, tout en évitant la formation d'un mécanisme de ruine prématuré. Quoique le dimensionnement de chaque élément à part pour qu'il assure la résistance et la stabilité, il est cependant nécessaire de penser sur l'intégrité du système structurel en entier et l'interaction des éléments entre eux.

L'étude réalisée ici ne permet pas de définir une règle générale de prédiction de mécanisme de ruine, mais contribue au profit de la potentialité de la technique des réseaux de neurones par des applications dans notre domaine.

Vu l'absence d'une base de donnée d'exemple d'apprentissage, on était obligé de la construire par un calcul éléments finis, qui lui-même ne traduit pas la réalité – surtout pour une analyse limite- et est élaborée sous certaines hypothèses.

Il est ainsi recommandée d'étaler l'applicabilité des réseaux de neurones à d'autres problèmes, pourvu qu'on dispose de résultats d'essais expérimentaux suffisamment

nombreux permettant la construction d'une base d'exemple, surtout si les essais aux laboratoire sont assez coûteuse à réalisée, et que l'on ne peut pousser l'étude à fin de cerné l'influence de tout les paramètres, dans ce cas, le recoure au réseaux de neurones est d'un intérêt incontournable.

**APPENDICE A**  
**LISTE DES SYMBOLES**

- $C$  : matrice d'amortissement
- $C_i$  : capacité
- $E$  : module de Young
- $E_d$  : énergie dissipée par la structure
- $E_e$  : énergie de déformation élastique
- $E_h$  : énergie dissipée par effet d'hystérésis (déformation plastique)
- $E_i$  : énergie du séisme transmise à la structure par les fondations
- $E_k$  : énergie cinétique
- $E_s$  : énergie emmagasinée dans la structure
- $E_v$  : énergie dissipée par amortissement visqueux
- $F_1$  : couche de comparaison,
- $F_2$  : couche de réorganisation
- $G_i$  : contrôleur de gain
- $I_{ei}$  : Inertie géométrique de l'élément  $e_i$
- $I_i$  : les données d'apprentissage du réseau
- $K$  : matrice de rigidité
- $K_{ei}$  : rigidité de l'élément  $e_i$
- $L_{ei}$  : longueur de l'élément  $e_i$
- $M$  : matrice de masse
- $M_p$  : moment plastique
- $M_y$  : moment résistant
- $R$  : coefficient de redondance
- $T_{ij}$  est le poids synaptique
- $Z_j$  : sortie du  $j^{ième}$  nœud gagnant de la couche compétitive
- $e_i$  : l'élément  $i$  du système structurel
- $j$  : le nombres de joints
- $m$  : le nombre d'éléments structurels

$net$  : abréviation anglaise de « network », somme des produits des entrées par poids

$r$  : le nombre de réactions externes

$w$  : poids synaptique d'une liaison

$w_{ij}$  : poids synaptique de la relation venant des neurones  $j$  vers le neurone  $i$

$x$  : entrée d'un neurone

$x$  : vecteur des déplacements

$\dot{x}$  : vecteur des vitesses

$\ddot{x}$  : vecteur des accélérations

$y$  : sortie d'un neurone

$\delta_{pk}$  : erreur commise entre vecteur d'entrées et de sorties

$\eta$  : paramètre d'apprentissage

$\square$  facteur de forme

$\theta$  : seuil

$\rho_i$  : résistance

## APPENDICE B ANALYSE PUSHOVER AVEC SAP2000 NON-LINEAIRE

L'analyse Pushover est une analyse statique, non-linéaire et dans la quelle, les forces appliquées sont augmentées progressivement. Cette analyse détermine les liaisons fragiles et le mode de ruine du système structurel.

L'effet réel du comportement cyclique dans le cas d'une analyse dynamique, est remplacée par une relation force-déformation monotonique (Figure B.1).

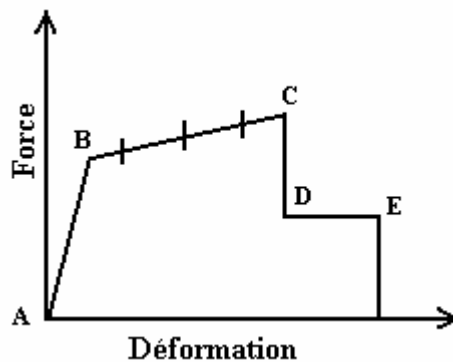


Figure B.1: Courbe de force-Déformation d'une rotule plastique [28]

Le ATC-40<sup>1</sup> et le FEMA-273<sup>2</sup> présente une procédure de modélisation de l'analyse Pushover. Dans ce qui suit, on présente les étapes à suivre pour mener une analyse Pushover avec le logiciel de calcul linéaire et non-linéaire SAP2000

1. Créer le modèle de base de la structure à étudier (sans les paramètres de l'analyse Pushover). Le menu graphique du SAP2000 facilite cette tâche, surtout pour des modèles complexes.
2. définir les critères et les critères à tolérer pour les rotules plastiques comme présenté à la figure B.2. Le programme présente plusieurs types par défaut de rotules plastiques qui sont basées sur les valeurs moyennes du ATC-40 pour les

<sup>1</sup> ATC, 1996. Seismic Evaluation and retrofit of Concrete Building, Volume 1, ATC-40 Report, Applied Technology Council, Redwood City, California

<sup>2</sup> FEMA, 1997. NEHRP Guidelines for the Seismic Rehabilitation of Buildings, Developed by the Building Seismic Safety Council for the Federal Emergency Agency (Report No. FEMA 273), Washington, DC.

éléments en béton armé, et les valeurs du FEMA-273 pour les éléments métalliques. Ces types par défauts sont préconisés pour des analyses préliminaires, l'utilisateur peut définir des propriétés propres à son cas d'analyse.

**Frame Hinge Property Data for FH1 - M3**  
Edit

Point	Moment/SF	Rotation/SF
E-	-0.2	-8
D-	-0.2	-6
C-	-1.25	-6
B-	-1	-1
A	0	0
B	1.	1.
C	1.25	6.
D	0.2	6.
E	0.2	8.

Graph showing the hinge behavior curve. The curve is symmetric about the origin and shows a yield point followed by a hardening region and a post-peak region.

Hinge is Rigid Plastic  
 Symmetric

**Scaling for Moment and Rotation**

	Positive	Negative
<input checked="" type="checkbox"/> Use Yield Moment	Moment SF	
<input checked="" type="checkbox"/> Use Yield Rotation	Rotation SF	

**Acceptance Criteria (Plastic Rotation/SF)**

	Positive	Negative
Immediate Occupancy	2.	
Life Safety	4.	
Collapse Prevention	6.	

OK  
Cancel

Figure B.2: propriétés d'une rotule plastique [2]

- Localisé les endroits d'affectation des rotules plastiques sur les éléments structurels. On sélectionne le ou les éléments sur le modèle et assigner les propriétés des rotules déjà définies (Figure B.3).

**Frame Hinges (Pushover)**

**Frame Hinge Data**

Hinge Property	Relative Distance
Default-M3	1
Default-M3	0
Default-M3	1

Add  
Modify  
Delete

OK  
Cancel

Figure B.3 : Assigner les rotules plastiques [2]

4. Définir les cas de charges Pushover. SAP2000 peut prendre en compte plusieurs cas de charge Pushover dans une même analyse. De même qu'un cas de charge Pushover peut débiter des conditions finales du cas Pushover précédent dans un même analyse. Typiquement le premier cas de charge Pushover est utilisé pour appliquer les forces gravitaires, en suite le cas de charge Pushover utilisé pour appliquer le système de force latérale qui commence de la fin du premier cas de charge. Les cas de charges Pushover peuvent être contrôlés selon la force, où les forces sont incrémentées jusqu'à une valeur prédéfinie, ou bien contrôlés par déplacement, c'est à dire jusqu'à un déplacement déjà spécifier, le point de référence des déplacements est généralement (sauf intervention de l'utilisateur) le dernier diaphragme. SAP2000 permet une distribution du système de force latérale basée sur une accélération uniforme dans une direction donnée, une déformé modale ou un cas de charge statique définie par l'utilisateur. La figure B.4 présente la fenêtre de définition des paramètres d'un cas de charge Pushover.

The screenshot shows the 'Static Pushover Case Data' dialog box. The 'Pushover Case Name' is 'PUSH1'. The 'Options' section includes:
 

- Push to Load Level Defined by Pattern
- Push to Displ. Magnitude: 0.36
- Use Conjugate Displ. for Control
- Monitor: U1 at Joint: 4
- Start from Previous Pushover: (dropdown)
- Save Positive Increments Only
- Minimum Saved Steps: 10
- Maximum Null Steps: 50
- Maximum Total Steps: 200
- Maximum Iterations/Step: 10
- Iteration Tolerance: 1.000E-04
- Event Tolerance: 0.01

 The 'Member Unloading Method' section has:
 

- Unload Entire Structure
- Apply Local Redistribution
- Restart Using Secant Stiffness

 The 'Geometric Nonlinearity Effects' section has:
 

- None
- P-Delta
- P-Delta and large Displacements

 The 'Load Pattern' section shows a table with two entries:
 

Load	Scale Factor
PUSH	1
PUSH	1

 Buttons for 'Add', 'Modify', 'Delete', 'OK', and 'Cancel' are also visible.

Figure B.4: Paramètres d'un cas de charge Pushover [2]

5. Lancer l'analyse statique ou dynamique, et en suite, lancer l'analyse statique non-linéaire Pushover.
6. Le résultat fondamental de l'analyse est la courbe Pushover illustrée sur la figure B.5, le menu « File » dans cette fenêtre d'affichage permet de visualiser ou d'imprimer un fichier, un tableau donnant les coordonnées de chaque pas de la courbe Pushover (Figure B.6)

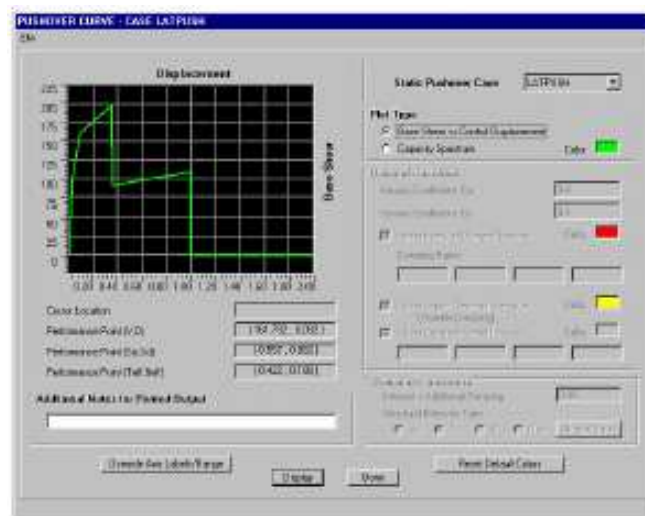


Figure B.5: Courbe Pushover [2]

Step	Displacement	Base Shear	A-B	B-IG	IG-L5	L5-CP	CP-C	C-D	D-E	>E	TOTAL
0	0.0000	0.0000	60	0	0	0	0	0	0	0	60
1	0.0224	100.4079	57	3	0	0	0	0	0	0	60
2	0.0421	157.4939	51	9	1	0	0	0	0	0	60
3	0.0978	161.1401	49	11	1	0	0	0	0	0	60
4	0.1401	198.9614	46	6	3	2	1	2	0	0	60
5	0.1401	160.7877	46	6	3	2	1	0	2	0	60
6	0.1513	161.2326	46	6	3	2	1	0	0	2	60
7	0.1513	142.7613	46	6	3	2	1	0	0	2	60
8	0.1530	144.4506	46	6	3	2	1	0	0	2	60
9	0.1564	144.9255	46	6	3	0	1	2	0	2	60
10	0.1565	108.0511	46	6	3	0	1	0	2	2	60
11	0.1596	108.2019	46	6	3	0	1	0	0	4	60
12	0.1596	89.6470	46	6	3	0	1	0	0	4	60
13	0.4538	93.9015	40	12	3	0	1	0	0	4	60
14	0.4538	95.5164	40	0	15	0	1	0	0	4	60
15	0.4538	105.1312	40	0	10	5	1	0	0	4	60
16	1.0031	109.3216	40	0	0	13	1	2	0	4	60
17	1.0031	-0.4043	40	0	0	11	1	0	0	4	60
18	1.2031	-0.4604	40	0	0	11	1	0	0	3	60
19	1.4031	-0.3965	40	0	0	11	1	0	0	3	60
20	1.6031	-0.3526	40	0	0	11	1	0	0	3	60
21	1.8031	-0.3086	40	0	0	11	1	0	0	3	60
22	2.0000	-0.2654	40	0	0	11	1	0	0	3	60

Figure B.6: Données de la courbe Pushover sous forme de tableau [2]



7. Revoir le mode de déformation est la séquence d'apparition es rotules plastiques étape par étape. Une apparition de rotule témoigne le début d'une déformation plastique, et les codes couleur pour le taux de plastification.
8. Revoir le changement des diagrammes des efforts internes étape par étape, ainsi que la progression des déformations, cela sur deux fenêtres séparées, ou sur le même graphisme.

## APPENDICE C

### Neural Network Toolbox de MATLAB [17]

#### C.1. Introduction

La « Neural Network Toolbox » (NNT) de MATLAB<sup>®</sup> est un environnement de développement de réseau de neurones, doté de plusieurs procédures système à travers lesquelles l'architecture et les propriétés de « l'objet réseau » sont construites. Essayer de cerner sur ces lignes tous ces propriétés

La première étape dans la conception d'un réseau de neurones est de créer un objet réseau soit par des fonction prédéfinie selon l'architecture voulu, comme par exemple les commande **newp** et **newff**, ou en créant un objet réseau vierge, et la ligne de commande qui permet de construire un tel objet est :

```
>> net = network;
```

Cela crée un réseau vierge, c'est à dire sans aucune propriétés prédéfinie. Ces propriétés seront le sujet des paragraphes suivants.

#### C.2. Couches du réseau

Le terme couche pour les réseaux de neurones définit une couche de neurones appartenant à la même sous ensemble du réseau, excepter la couche d'entre que l'on considère par abus de langage comme une couche.

Et donc, dans la terminologie de la boîte à outil réseaux de neurones l'architecture de la figure C.1 est à une seule couche, et celle de la figure C.2 à deux couches.

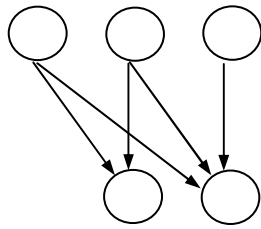


Figure C.1: Réseau à une seule couche

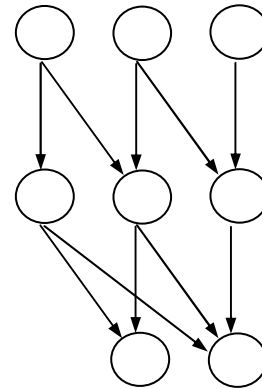


Figure C.2: Réseau à deux couche

chaque couche possède ces propre propriétés, dont la plus importante est la fonction de transfert des neurones de cette couche, et la fonction définissant l'entrée de chaque neurone qui donne les poids synaptiques et les sortie de la couche précédente

### C.3. Construction des couches

La boite à outil réseau de neurone support des réseaux à plusieurs couches d'entrées, la commande suivante définie cela :

```
>> net.numInputs = 1;
```

Le nombre de neurones dans la couche d'entrée est évident, il doit être égale à la dimension du vecteur d'entrées. Cette propriété est définie par commande **net.inputs{i}.size**, où i l'index de la couche d'entrées. Une couche d'entrée supportant un vecteur à 3 dimensions est donc déclarée par :

```
>> net.inputs{1}.size = 2;
```

Jusque là on a définie la couche d'entrée, les propriétés qui suivent sont **net.numLayers**, qui définie le nombre totale de couche dans le réseau, et **net.layers{i}.size**, qui donne le nombre de neurones dans la couche i. par exemple, les ligne ci-dessous définissent un réseau à 2 couches dont la première à 3 neurones, et la deuxième à 2 neurones :

```
>> net.numLayers = 2;
```

```
>> net.layers{1}.size = 3;
```

```
>> net.layers{2}.size = 2;
```

#### C.4. Connexion des couches

Premièrement, définir la couche qui est relié avec la couche d'entrées par la commande **net.inputConnect(i)** (généralement c'est la première couche, donc  $i = 1$ ).

Les connexions entre les autre couches est définie par la matrice de connectivité **net.layerConnect** qui peut avoir 0 ou 1. Si l'élément  $(i,j)$  est 1, les sorties de la couche  $j$  sont connectées aux entrées de la couche  $i$ .

On doit aussi définir quelle couche est la couche de sortie par attribution de la valeur 1 a **net.outputConnect(i)**, pour la couche  $i$  correspondante. Finalement, si on à un apprentissage supervisé, on doit définir la couche qui est liée aux valeurs désirés, (généralement, c'est la couche de sortie). Cela par initialisation de **net.targetConnect(i)** à 1. Par exemple :

```
>> net.inputConnect(1) = 1;
>> net.layerConnect(2, 1) = 1;
>> net.outputConnect(2) = 1;
>> net.targetConnect(2) = 1;
```

#### C.5. Attribution des fonctions de transfert

Chaque couche possède ça propre fonction de transfert, définie par la propriété **net.layers{i}.transferFcn**.

Les deux lignes de commande suivante définissent une fonction de transfert sigmoïde dans la première couche, et une fonction linéaire dans la deuxième.

```
>> net.layers{1}.transferFcn = 'logsig';
>> net.layers{2}.transferFcn = 'purelin';
```

Pour autres fonctions de transfert, on peut s'orienté à la documentation de Matlab.

#### C.6. Poids synaptiques et biais

On définit les couches qui ont des biais par la commande **net.biasConnect** dont on attribue 0 ou 1, une couche  $i$  possède des biais si **net.biasConnect(i) = 1**.

```
>> net.biasConnect = [ 1 ; 1];
```

Cette commande raccorde les deux couches du réseau par des biais.

En suite on initialise les poids synaptiques du réseau. La commande ci-dessous supprime toutes les valeurs des poids et biais existant pour permettre à l'utilisateur de choisir la procédure d'initialisation :

```
>> net = init(net);
```

La première des chose est le choie de la fonction d'initialisation par **net.initFcn**. La valeur 'initlay' est la plus appropriée, elle permet que chaque couche s'initialise avec ça propre routine. (L'utilisateur peut construire ça propre routine d'initialisation)

```
>> net.initFcn = 'initlay';
```

et cela se fait pour chaque couche par la propriété **net.layers{i}.initFcn**.

Pour le détail des routines d'initialisations, consulter la documentation Matlab.

## C.7. Fonctions et paramètres d'apprentissage :

### C.7.1. Différence entre "train" et "adapt":

Deux manières d'apprentissage peuvent être utilisée, l'apprentissage directe (adapt) ou indirect (train).

La différence demeure dans la manière de mise à jour des poids synaptique, l'apprentissage directe va mètre à jour les poids après chaque traitement d'un vecteur d'entrées, par contre dans le cas d'un apprentissage indirecte, les poids sont mis à jours après que tout les vecteurs d'entrées sont traités. (Dans certain cas « adapt » peut être un apprentissage indirect).

Par contre, l'avantage de « train » est le grand plage de choie des fonctions d'apprentissage (gradient descent, gradient descent w/ momentum, Levenberg-Marquardt, etc.) qui ont prouvées leurs efficacité.

### C.7.2. Fonctions de performance

Les deux options reconnues sont, l'erreur moyenne absolue (mae) et l'erreur moyenne des moindre carrée (mse).

La fonction de performance est définie par la propriété, par exemple:

```
>> net.performFcn = 'mse';
```

### C.7.3. Paramètres d'apprentissage

Chaque fonction d'apprentissage possède ces propres paramètres d'apprentissage, leurs valeurs sont définies par **net.trainParam**. Les paramètres présents dépendent de la fonction d'apprentissage choisie. Par exemple, si l'algorithme d'apprentissage choisie est le Gradient Descent w/ Momentum, on à donc :

```
>> net.trainFcn = 'traingdm';
```

Et les parametres sont initiés:

```
>> net.trainParam.lr = 0.1;
```

```
>> net.trainParam.mc = 0.9;
```

Voir la documentation Matlab pour d'autres éventuelles fonctions d'apprentissage.

Le paramètre **net.trainParam.epochs**, qui traduit le nombre maximal de fois que les vecteurs d'entrées sont utilisés pour l'apprentissage.

**APPENDICE D**  
**ARTICLES PUBLIES**

*Colloque International*  
*Alger, Algérie*

*11 et 12 Octobre 2003,*

Prédiction Des Mécanismes De Ruine Dans Les Portiques Sous L'action Sismique Par Les Réseaux De Neurones

N. BOURAHLA<sup>1</sup>, R. TALEB<sup>1</sup>

<sup>1</sup>Département de Génie Civil, Université Saad Dahleb de Blida, BP 270 Blida, Algérie, e-mail : nbourahla@yahoo.com

**RESUME** : *Sous l'effet d'une action sismique, les constructions sont quasiment sollicitées au delà de leurs limites élastiques. Ainsi, le comportement post-élastique exprimé en terme de ductilité et de capacité de dissipation d'énergie est d'une importance capitale dans la conception parasismique des structures. En pratique, ces notions sont prises en compte implicitement dans les codes sismiques par des coefficients d'amplification ou de réduction de la charge sismique. Il est cependant, intéressant de pouvoir prédire le comportement post-élastique au stade de la conception et intervenir éventuellement, durant cette phase, pour assurer une grande capacité de dissipation d'énergie et pourvoir une distribution uniforme de l'activité plastique, tout en évitant le risque d'une ruine prématurée. Cet article examine l'application de la technique des réseaux de neurones dans l'analyse quasi-statique non linéaire des portiques pour prédire la distribution des noeuds plastiques au moment de la ruine. En premier lieu, une base d'exemples est construite servant comme référence pour l'apprentissage du réseau de neurone, ensuite, de nouveaux exemples sont introduits dans le réseau pour le valider. Les résultats obtenus démontrent le potentiel de cette technique comme outil d'évaluation*

*du comportement post-élastique dans la pratique de la conception et le calcul parasismique des structures.*

Mots - clés : Mécanisme de ruine, Noeud Plastique, Réseaux de Neurones, Redondance, Analyse Push-over.

## 1- Introduction

Le phénomène de ruine structurel partiel ou total des constructions est largement observé lors des événements sismiques. Il représente l'un des dommages les plus redoutables. Souvent, il est engendré par l'apparition d'un mécanisme prématuré dans la structure dû au manque de redondance dans le système de contreventement. Il s'agit de la formation d'un certain nombre critique de noeuds plastiques dans les éléments de la structure qui entraîne la défaillance de la structure.

D'un point de vue structurel, une conception parasismique optimale, consiste à imposer la propagation ou la concentration des dommages dans une structure soumise à une sollicitation sismique, d'une manière à éviter la formation prématurée d'un mécanisme dans la structure portante et de différer l'état de ruine le maximum possible. Ce comportement donne à la structure un champ de ductilité suffisant pour dissiper le maximum d'énergie et d'augmenter sa capacité de résistance et de stabilité.

La propagation des noeuds plastiques dans une structure sous l'action d'une excitation sismique est un processus non - linéaire complexe. La procédure d'optimisation devient donc une tâche très difficile. Le présent travail, se limite dans une première phase, à prédire la formation des noeuds plastiques et le mécanisme de ruine probable dans une structure sous l'action d'un effort sismique en utilisant les réseaux de neurones. Cette approche d'analyse présente l'avantage de surpasser l'étape algorithmique d'écriture du processus à suivre pour résoudre le problème qui s'avère très

onéreuse dans ce cas. Toutefois, les étapes concernant le choix de l'architecture et le type du réseau de neurones, la constitution de la base de données, l'apprentissage ainsi que la validation du modèle sont présentés ci-dessous.



## 2- Mécanisme de ruine et redondance

Selon le contexte d'étude et le problème posé, la redondance peut être définie de manières différentes. En résistance des matériaux la redondance d'une structure est équivalente au nombre de réactions (inconnues) en surplus des équations d'équilibre de la statique. Evidemment, cette définition ne révèle pas la capacité de la structure à préserver sa stabilité même avec un certains nombre d'éléments endommagés. En se basant sur ce concept, le règlement parasismique RPA99, mesure la redondance par le nombre de files de contreventement parallèle à la direction de l'action sismique considérée Ref. [2].

Sous l'hypothèse d'une analyse push-over, Bertero et al Ref. [10], définissent explicitement la redondance en génie sismique comme étant le nombre de points (régions ou noeuds) critiques dont la défaillance (formation de noeuds plastiques) entraîne la ruine (formation d'un mécanisme) du système structurel. Par ailleurs, si une approche dynamique est sélectionnée pour l'analyse, le degré de redondance du système structurel est défini comme étant le nombre minimum de régions critiques ou de noeuds plastiques qui se rompent simultanément au moment de la ruine, et dans ce cas le degré de redondance ne dépend pas seulement des propriétés géométriques et mécaniques du système structurel, mais aussi du comportement dynamique de la structure, et de l'action sismique. Entre autre, l'apport essentiel de la redondance du système structurel et l'optimisation dans le dimensionnement des éléments, car en fait, un surdimensionnement même s'il assure la résistance, mais il pénalise la déformation plastique et la dissipation de l'énergie, ainsi que la distribution de l'énergie dissipé en différentes régions, évitant par cela la concentration des dommages.

## 3- Réseaux de neurones artificiels

Les premières applications des réseaux de neurones dans le domaine du génie civil remontent à plus de deux décennies. Ces dernières années, on constate que l'utilisation de cette technique pour la résolution de certaines catégories de problèmes dans les différentes branches du génie civil et particulièrement en génie sismique, est devenue courante.

Dans l'hypothèse que le comportement intelligent émerge de la structure et du comportement des éléments de base du cerveau, les réseaux de neurones artificiels se sont développés. Il s'agit de réseaux fortement connectés de processeurs élémentaires (neurone) fonctionnant en parallèle. Chaque neurone est doté d'un état interne, l'activation, par la quelle il influence les autres neurones du réseau, cette activation se propage dans le réseau par le long des liens synaptiques. La règle qui détermine l'activation d'un neurone en fonction de l'influence de ces pairs est appelée fonction d'activation Ref. [6]. L'intérêt d'un réseau de neurones est le calcul d'une valeur de sortie unique par chaque neurone de sortie sur la base des informations qu'il reçoit.

Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre de variable d'entrées en provenance des neurones amont. A chacune de ces entrées est associée un poids  $w$  représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurone aval. A chaque connexion est associée un poids (Fig.1).

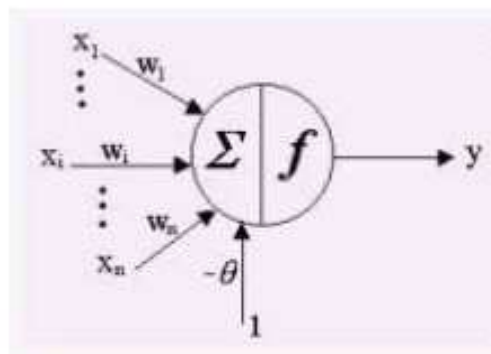


Figure 1 : Structure d'un neurone artificiel

$x = \{x_1, x_2, \dots, x_i, \dots, x_n\}$	Vecteur d'entrées	
$w = \{w_1, w_2, \dots, w_i, \dots, w_n\}$	Vecteur des poids synaptiques	
$\theta$	Seuil	
$net = \sum_{i=1}^N w_i x_i$	Potentiel d'action	(1)
$v = net - \theta$		(2)
$f(v)$	Fonction d'activation ou fonction de transfert	(3)

Le vecteur d'entrées  $x$  est multiplié par le vecteur des poids synaptiques, ce produit est sommé au niveau du neurone. Le neurone ne renvoi une valeur que si cette somme est accumulée avec la seuil  $\theta$  appelée aussi biais. Le biais est évalué comme le poids synaptique d'une entrée additionnel le de valeur 1. A partir de cette valeur, une fonction de transfert calcule la valeur de l'état du neurone, la sortie du neurone est donc :

$$o = f(v) = f\left(\sum_{i=1}^N w_i x_i - \theta\right) \quad (4)$$

C'est cette valeur qui sera transmise aux neurones avals. Il existe de nombreuses formes possibles pour la fonction de transfert. On remarquera qu'à la différence des neurones biologiques dont l'état est binaire, la plus part des fonctions de transfert sont continues, offrant une infinité de valeurs possibles comprises dans l'intervalle  $[0, +1]$  ou  $[-1, +1]$  selon la fonction utilisée.

La capacité numérique d'un seul neurone est souvent très limitée. Cependant, lorsque plusieurs neurones sont connectés, formant ainsi un réseau de neurones, un comportement complexe peut être assurée. Les connexions entre ces neurones décrivent la topologie ou l'architecture du réseau.

#### 4- Modèle de structure et analyse Push-over

Le développement récent dans le domaine de la conception parasismique des systèmes structurels a donné un statut pour la procédure de l'analyse push-over qui est devenue un moyen assez simple et rapide pour tester la performance d'un système structurel vis-à-vis d'une analyse dynamique non linéaire. L'analyse push-over est un calcul statique non linéaire, dont laquelle la magnitude d'un système de force est augmentée d'une manière incrémentale. Avec l'incrément progressif de ce système de force on peut détecter les régions faibles qui peuvent subir des déformations plastiques excessives ou les modes de ruine. L'analyse push-over est une tentative dans la phase de conception pour évaluer la capacité réelle d'un système structurel dans le cas des grandes déformations. Le concepteur peut ainsi intervenir dans cette phase pour éviter tout risque de rupture fragile ou de ruine prématuré Ref. [4].

Dans cette étude, on a établi un réseau de neurone capable de prédire la distribution des noeuds plastiques résultant d'une analyse push-over juste avant le moment de rupture (Fig. 2). Ce réseau est conçu pour des modèles de structures à 2 travées et 3 niveaux contreventées par des portiques auto-stables. L'analyse push-over de ce système est menée en deux étapes : la première c'est l'application des force gravitaire, la seconde c'est l'application progressive du système de force latérale jusqu'à la limite de la rupture. La loi de comportement est supposée élasto-plastique avec écrouissage.

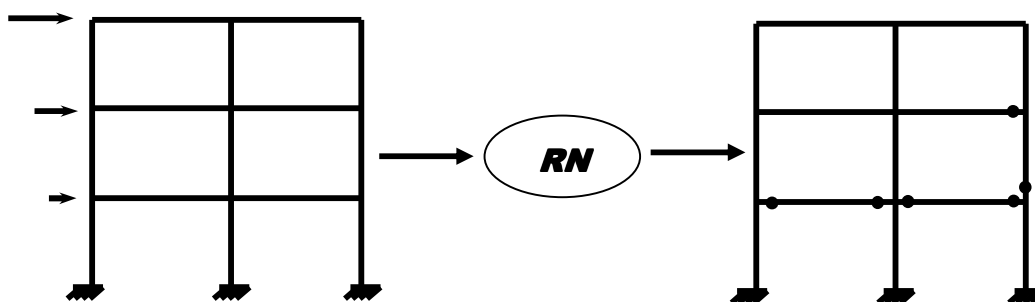


Figure 2 : Schéma du principe de fonctionnement du réseau de neurone.

## 5- Conception du réseau de neurones

### 5.1- Base de données

La construction de la base d'apprentissage est l'étape primaire et essentielle dans la conception du réseau de neurone. La base d'apprentissage pour cette application est constituée des résultats numériques d'une série d'analyse push-over menée par le logiciel de calcul SAP2000. Pour chaque configuration de structure il correspond une distribution des noeuds plastiques juste avant l'état de ruine, chaque noeud plastique est doté d'une couleur code représentant le niveau de plastification atteint.

### 5.2-Architecture du réseau

L'architecture du réseau de neurone utilisé est le modèle multicouche à rétro-propagation avec une couche des entrées, une couche cachée et une couche de sorties.

La couche d'entrée dont les variables sont les facteurs de rigidité flexionnelle des éléments résistants est constituée de 15 neurones, chaque neurone correspond au facteur

de rigidité d'un seul élément. Les facteurs de rigidité élémentaire sont écrits sous leur forme la plus simple avec les paramètres physiques et géométriques, et sont extraits de la matrice de rigidité construite directement du modèle de la structure (Fig. 3).

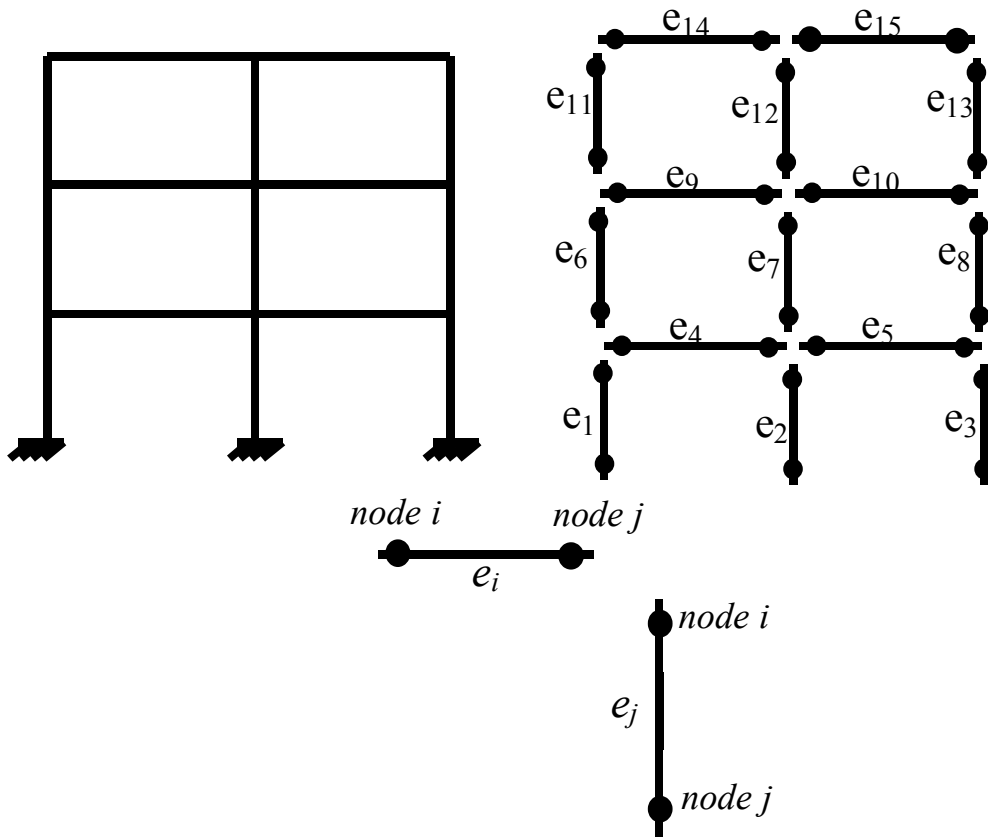


Figure 3 : Facteurs de rigidité élémentaires.

Matrice de rigidité extraite directement du modèle de structure:

$$\begin{bmatrix} k_{e14} & k_{e15} \\ k_{e11} & k_{e12} & k_{e13} \\ k_{e9} & k_{e10} \\ k_{e6} & k_{e7} & k_{e8} \\ k_{e4} & k_{e5} \\ k_{e1} & k_{e2} & k_{e3} \end{bmatrix}$$

$$(k_{e1}, \dots, k_{e15})$$

$$k_{ei} = \frac{E_{ei} \times I_{ei}}{l_{ei}^3} \quad (5)$$

$$k_{ej} = \frac{E_{ej} \times I_{ej}}{l_{ej}^3} \quad (6)$$

La couche de sortie du réseau donne la configuration des rotules plastiques dans la structure au moment de la ruine ainsi que le taux de plastification dans ces rotules, elle est constituée de 30 neurones donnant le taux de plastification des noeuds d'extrémités de chaque élément.

Les taux de plastification sont divisés en 05 niveaux selon le niveau de ductilité atteint (Fig. 4) :

Niveau initiale : correspond au domaine élastique (pas de formation de rotule plastique).

Niveau finale : correspond à la rupture de l'élément ou l'élément arrive à sa capacité maximale.

Niveaux intermédiaires : ou intervalle de l'activité plastique divisé davantage en trois niveau : plastification modéré, moyenne et importante.

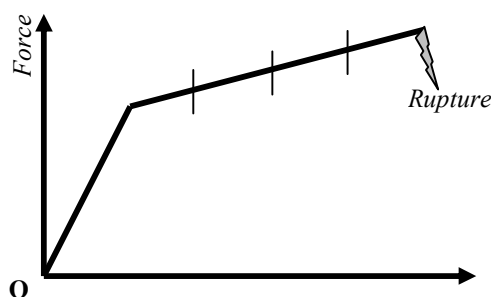


Figure 4 : Courbe force déformation et niveau de plastification d'un nœud plastique.

Couche cachée : Contrairement aux couches d'entrée et de sortie où leurs nombres de neurones est établi selon les paramètres définies, il n'existe pas de règles pour déterminer le nombre exact de noeuds dans la couche cachée, ou même le nombre de couches cachées à utiliser. Toutefois, le guide des utilisateurs de la boîte à outil des réseaux de neurones (MATLAB) suggère que les réseaux de neurones avec biais, ayant une fonction de transfert dans les noeuds de la couche cachée du type « sigmoïde », et dans la couche d'entrée une fonction de transfert du type « linéaire », sont capables d'imiter n'importe quelle fonction avec un nombre limité de discontinuités. Des travaux antérieurs Ref. [3] suggèrent que le nombre de neurones dans les couches cachées soit entre la moitié des neurones d'entrées et de sorties et la somme de ces neurones.

### 5.2- Codage (traitement) des données

Les valeurs numériques des entrées/sorties doivent être soigneusement choisies, et doivent dans la plupart des cas passer par une étape de codage, car l'utilisation des valeurs numériques sous leurs formes brutes peut générer le non apprentissage du réseau et la saturation des fonctions de transfert, donnant ainsi des résultats erronées.

Le codage ou la normalisation des paramètres consiste à transformer ces données (valeurs d'entrées et de sorties) en des valeurs qui peuvent être facilement traitées par les fonctions de transfert dans les neurones. Les intervalles les mieux adaptées sont :  $[-1, +1]$  ou  $[0 +1]$ .

Dans notre cas, la méthode adoptée pour le codage des facteurs de rigidité consiste à diviser ces facteurs par le facteur de rigidité le plus grand. L'intervalle des valeurs codées devient alors  $[0, 1]$ . Le vecteurs des entrées s'écrit comme suit :

$$\left( 1, \frac{k_{e2}}{k_{e1}}, \frac{k_{e3}}{k_{e1}}, \dots, \frac{k_i}{k_{e1}}, \dots, \frac{k_{e15}}{k_{e1}} \right)$$

Les taux de plastification sont normalisés dans l'intervalle  $[-0.5, 1]$ , la valeur  $-0.5$  correspond au domaine élastique, la valeur  $1$  à la limite de la rupture du noeud et les valeurs intermédiaires  $(0, 0.25, 0.75)$  aux 3 niveaux de plasticité.

### 5.3- Apprentissage

L'apprentissage utilisé est du type dit supervisé, où les couples entrées-sorties de la base d'apprentissage sont montrés au réseau d'une manière séquentiel. L'erreur commise (la différence entre la sortie désirée et celle donnée par le réseau) est rétro propagée dans le réseau pour mettre à jour les poids synaptiques initiales. Il est parfois difficile de connaître l'algorithme d'apprentissage qui correspond le mieux au problème traité, car il dépend de plusieurs paramètres parmi les quelles la complexité du problème, la taille du vecteur d'entrée utilisé pour l'apprentissage, les poids synaptiques et l'erreur maximale voulue. Pour cet application, et après une étude comparative entre les différents algorithmes proposés par le logiciel MATLAB Ref. [6], on a opté pour l'algorithme de Levenberg-Marquardt. L'erreur commise par le réseau de neurone durant la phase d'apprentissage est contrôlée par la méthode des moindres carrées.

La courbe de performance de l'apprentissage représentée sur la Fig. 5 montre l'efficacité de l'algorithme de Levenberg-Marquardt qui converge après 6 itérations.

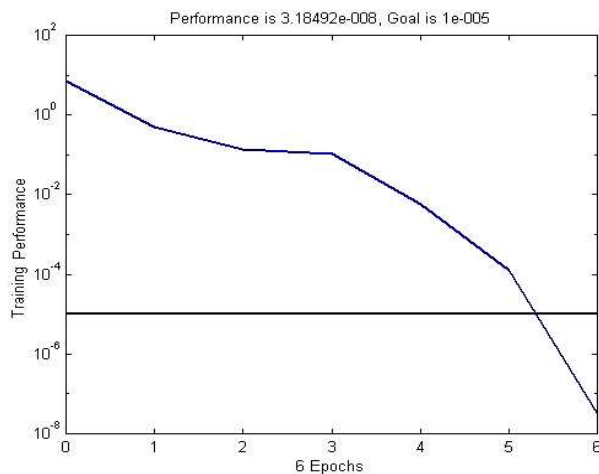


Figure 5 : Courbe de performance de l'apprentissage.

#### 5.4- Test et validation

La phase d'apprentissage étant achevée, une série de tests de validation a été effectuée pour éprouver l'efficacité du réseau de neurone. Des modèles de structures autres que ceux déjà montrés au réseau pendant l'apprentissage ont été utilisés. A titre d'exemple, Fig. 6 illustre une comparaison entre une structure de la base d'exemples montrant la distribution réelle des noeuds plastiques avec leurs couleurs (nuances en gris) codes des niveaux de ductilité atteints (a), à droite on montre la distribution des noeuds plastique prédite par le réseau de neurones qui démontre la similitude exacte.

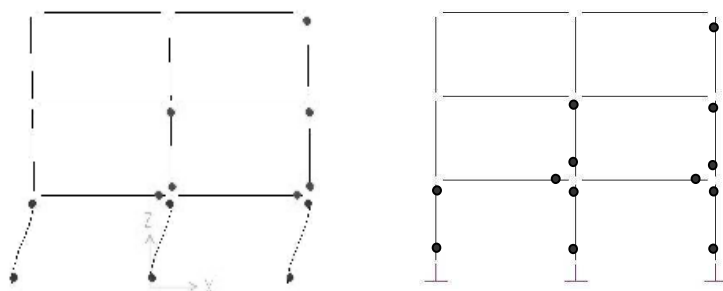


Figure 6 : Distribution des nœuds plastiques (exemple 2)

a) Modèle réel b) Prédiction par réseau de neurones



Une régression linéaire entre les résultats obtenus des analyses push-over et les sorties du réseau de neurones a été effectuée. Le coefficient de corrélation est proche de l'unité, ce qui démontre une bonne corrélation entre les résultats du réseau de neurones et l'analyse push-over. Cependant, comme le montre Fig. 7, le modèle manifeste des écarts plus importants pour les deux premiers niveaux de taux de plastification (noeuds non plastifiés et noeuds légèrement plastifiés).

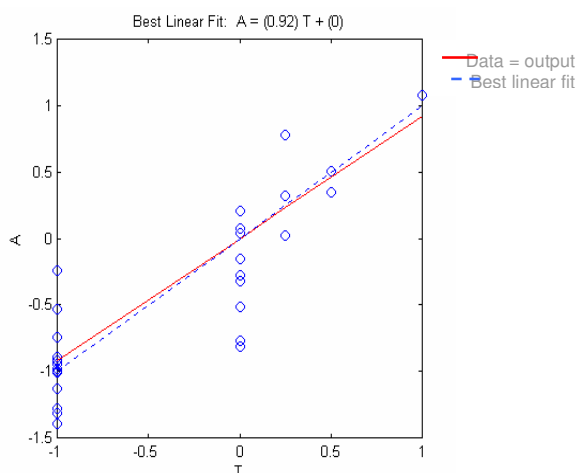


Figure 6 : Courbe de corrélation (exemple 2)

En terme d'erreurs, le tableau 1 exprime les rapports des écarts de prédiction des taux de plastification.

Tableau 1 : Erreur dans la prédiction du degré de plastification pour chaque nœud

N° Nœud	Erreur	N° Nœud	Erreur	N° Nœud	Erreur
Nœud 1	$2.00E-04$	Nœud 11	$1.00E-04$	Nœud 21	$0.00$
Nœud 2	$1.00E-04$	Nœud 12	$2.00E-04$	Nœud 22	$2.00E-04$
Nœud 3	$1.00E-04$	Nœud 13	$2.00E-04$	Nœud 23	$2.00E-04$
Nœud 4	$2.00E-04$	Nœud 14	$2.00E-04$	Nœud 24	$3.00E-04$
Nœud 5	$0.00$	Nœud 15	$0.00$	Nœud 25	$1.00E-04$
Nœud 6	$2.00E-04$	Nœud 16	$0.00$	Nœud 26	$3.00E-04$
Nœud 7	$1.00E-04$	Nœud 17	$1.00E-04$	Nœud 27	$0.00$

Nœud 8	<i>1.00E-04</i>	Nœud 18	<i>2.00E-04</i>	Nœud 28	<i>2.00E-04</i>
Nœud 9	<i>4.00E-04</i>	Nœud 19	<i>2.00E-04</i>	Nœud 29	<i>2.00E-04</i>
Nœud 10	<i>2.00E-04</i>	Nœud 20	<i>2.00E-04</i>	Nœud 30	<i>0.00</i>

Il est clair que dans la pratique ces différences sont acceptables. Toutefois, il faut signaler que la généralisation de la méthode pour des systèmes structurels plus grands présente un inconvénient à cause de la taille du réseau de neurone qui devient très grande. Dans ce cas, le volume des entrées est très considérable ce qui conduit à des bases d'apprentissage importantes.

### 5- Conclusion

Le risque tragique qui menace les structures sujettes aux sollicitations sismiques est sans doute la ruine totale ou partielle de la construction. Connaître le mécanisme de ruine d'une structure au stade de la conception est d'une utilité capitale. Ainsi, une technique de prédiction de la propagation des noeuds plastiques dans les portiques sous l'effet d'une sollicitation sismique a été établie.

Un réseau de neurone à rétropropagation est entraîné pour déterminer la distribution des noeuds plastiques avec leurs taux de plastification au moment de la rupture. Les paramètres d'entrée du réseau sont les rigidités flexionnelles élémentaires. Ce réseau de neurone peut être utilisé comme un outil simple et rapide pour évaluer le comportement non linéaire du système structurel et permet au concepteur d'intervenir sur les caractéristiques géométriques de la structure pour optimiser la ductilité et la capacité de dissipation d'énergie de la structure, tout en évitant la formation d'un mécanisme de ruine prématuré.

### **Références**

- [1] Ashraf Habibullah,, S. E. , and Stephen Pyle, S. E., "Practical Three Dimensional Nonlinear static Pushover Analysis", Structure Magazine, 1998
- [2] Centre National de Génie Parasismique, "Règlement Parasismique Algérien RPA99", 1999
- [3] Hajela, P. and Berke, L. ,"Neurobiological Computational Models in Structural Analysis and Design", Computers & Structures, Vol.41, No.4, pp.657-667. 1991

- [4] Hiratsuka K, and Shingu K, “Dynamic Response of Conical Shell Using Neural-Network Based Vibration Control”, Proceeding of 12<sup>th</sup> World Conference on Earthquake Engineering, Paper No 0450, 2000
- [5] Ian Flood, and Nabil Kartam, “Neural Network in Civil Engineering. I: Principles and Understanding”, Journal of Computing in Civil Engineering, Vol 8, No 2, pp. 131-148, 1993
- [6] Neural-Networks Toolbox User’s Guide: For use with MATLAB (1984-2001),
- [7] Papadrakakis M., Papadopoulos V., and Lagaros N. D., “Optimum Design of 3D Structures Under Static and Dynamic Loading”, Computational Methods for Shell and Spatial Structures, 2000
- [8] Papadrakakis M., Papadopoulos V., and Lagaros N. D., “Structural Reliability Analysis of Elasto-Plastic Structures Using Neurale-Networks and Monte Carlo Simulation”, Journal of Computer Methods in Applied Mechanics and Engineering, No 136, pp. 145-163, 1996
- [9] Rahal K. N., and Abu kiefia M. , “Neural-Network for Calculation of Shear Strength of Reinforced Concrete Beams”, Kuwait Journal of Science, Vol 26, No 3, pp. 239-251, 1999
- [10] Raul D. Bertero and Vitelmo V.Bertero, “Redundancy in Earthquake-Resistant Design”, Journal of Structural Engineering, Vol 125, No 1, pp. 81-88, 1999
- [11] Sameh S. F. Mehanny and Gregory G. Deierein, “Seismic Damage and Collapse Assessment of Composite Moment Frames”, Journal of Structural Engineering, Vol 127, No 9, pp. 1045-1053, 2001
- [12] Wei Lu, “Neural-Network Model for Distortional Buckling Behaviour of Cold-Formed Steel Compression Members”, Thesis for the Degree of Licentiate of Science in Technology, Helsinki University of Technology, 2000

**Paper 63**

©2003, Civil-Comp Ltd., Stirling, Scotland  
 Proceedings of the Seventh International Conference  
 on The Application of Artificial Intelligence to  
 Civil and Structural Engineering, B.H.V. Topping  
 (Editor), Civil-Comp Press, Stirling, Scotland.

## **Structural Response Extraction from Sound Vibration Measurements using Neural Networks**

**N. Bourahla, R. Taleb and T. Boukhemacha**  
**Department of Civil Engineering**  
**University of Blida, Algeria**

**Abstract**

In earthquake engineering accurate structural response measurements are of paramount importance for successful investigation of the behaviour of physical models. However in most cases, dynamic testing yields huge amount of data containing valuable information in clustery form which needs tedious processing. This paper presents a technique that helps interpreting measured signals during dynamic testing. For better illustrating this, simple experiments are conducted on fixed ended cables and use is made of neural networks to extract structural response parameters from the recorded acoustic signals emitted from the vibrating cables. The training is performed using a data base composed of recorded sound signals (input) and acceleration time-histories numerically calculated at a specific point of the specimen (target signals). The neural network model is validated using a wide range of data. The approach is interesting in situation in which fast demonstration for educational purposes are needed. It can be easily adapted to other applications of wave propagation through undefined media for which additional accuracy or noise extraction is sought.

**Keywords:** neural network, sound vibration, dynamic analysis, noise cancellation, dynamic testing.

**1 Introduction**

Earthquake experiments, whether on shaking tables, laboratory testing or instrumented field structures, can yield huge amounts of data containing valuable information on the performance of structures but often in cluttered form. In strictly

controlled conditions, measurements can be of high quality which makes the data reduction easier. Yet, the processing and interpretation phase is deemed to be a difficult task particularly for nonlinear behaviour of test specimens [1]. However, in situation where the measured response amplitudes are very low such as in most cases of ambient vibration testing where the signals may be highly altered by extraneous noise, then the processing task becomes more demanding [2].

This paper discusses the potential use of neural network and its application to the problems of data processing in seismic testing. Specific reference is made to structural information extraction from recorded signals. In fact, artificial neural networks have been trained to perform complex functions in various fields of application including pattern recognition, identification, classification, speech, vision and control systems. Civil engineering applications of neural network have grown in recent years and are now widely used in many fields, such as in structural, geotechnical and earthquake engineering [3, 4, 5]. In the particular context of experimental dynamic response of structures, the efficient use of artificial neural networks is attributed to the ability of the latter in extracting specific features from tremendous volume of data. To cite few of the many applications of neural networks in this field, it is worth mentioning the significant effort made in structural damage detection and health monitoring of structures [6, 7, 8]. Probabilistic neural networks have also been used to ‘perceive’ the intrinsic relationship between the seismic and soil parameters and liquefaction potential [9]. Further, neural networks were employed in an attempt to allow a certain degree of automatic traceability of the criteria used for the identification and timing of seismic events [10].

In the proposed paper, a multilayer feed-forward neural network is developed and trained to extract acceleration time histories from recorded sound signals emitted by a vibrating structure.

## **2 Description of the system structure**

The test specimens are six cables having diameters of 0.4 mm, 0.5 mm, 0.7 mm, 0.8 mm, 1.1 mm and 1.3 mm. The cables were attached at their ends to form fixed ended spans. Damped free vibrations were initiated by shifting the mid-span from its equilibrium position and releasing it. A standard PC microphone was used to record

the sounds emitted by the vibrating cables. A typical sound wave measurement is shown in Figure 1.

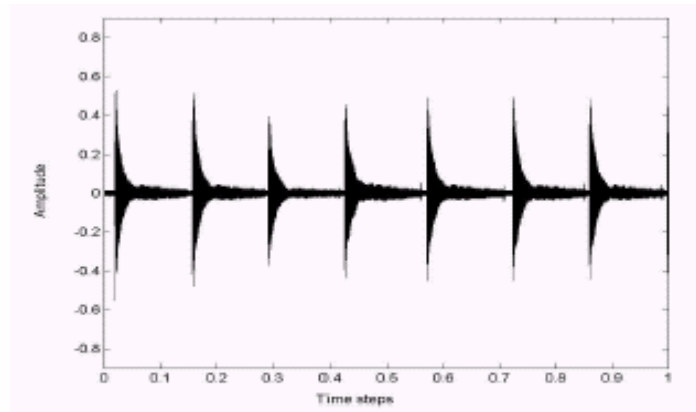


Figure 1 Typical sound wave record

For the purpose of this study, a simple finite element model of a fixed ended beam shown in figure 2a, has been elaborated using SAP2000NL program. The damped free vibrations in terms of acceleration time history responses at mid-span are calculated at three initial amplitudes for the six different cables. Thus a total of 18 responses are obtained. The damping factor is determined by matching the recorded wave pattern in the time domain. A typical acceleration time history dominated by the fundamental mode is shown in figure 2b.

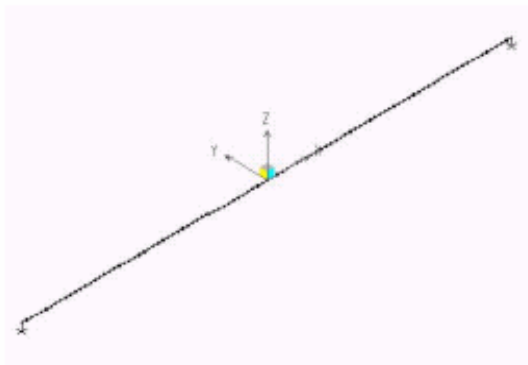


Figure 2a Finite element model

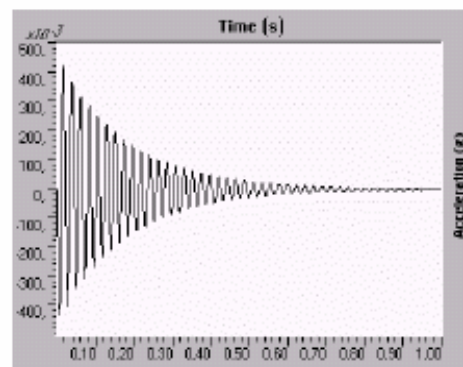


Figure 2b Acceleration time history at mid span of a fixed ended beam

### 3 Neural network analysis

In the above model, a complex relationship between the acceleration time history at mid-span and the recorded sound emitted by the vibrating cable is sought. The relationships encoded in a certain number of training patterns is an attractive

involved is crucial in the analysis as will be presented in the following example.

### 3.1 Network design

For an accurate extraction of a specific wave from a noisy background, we consider appropriate to use Elman network. Commonly, this network is a two-layer network with feedback from the first layer output to the first layer input. This recurrent connection allows the Elman network to both detect and generate time-varying patterns. A diagram representing an Elman network is shown in Figure 3. It has tangential sigmoid transfer functions in the neurons of its hidden (recurrent) layer, and pure linear transfer function in the neurons of its output layer. This combination is special in that two-layer networks with these transfer functions can approximate any function (with a finite number of discontinuities) with arbitrary accuracy. The only requirement is that the hidden layer must have enough neurons. More hidden neurons are needed as the function being fit increases in complexity. Note that the Elman network differs from conventional two-layer networks in that the first layer has a recurrent connection. The delay in this connection stores values from the previous time step, which can be used in the current time step. Thus, even if two Elman networks, with the same weights and biases, are given identical inputs at a given time step, their outputs can be different due to different feedback states [11].

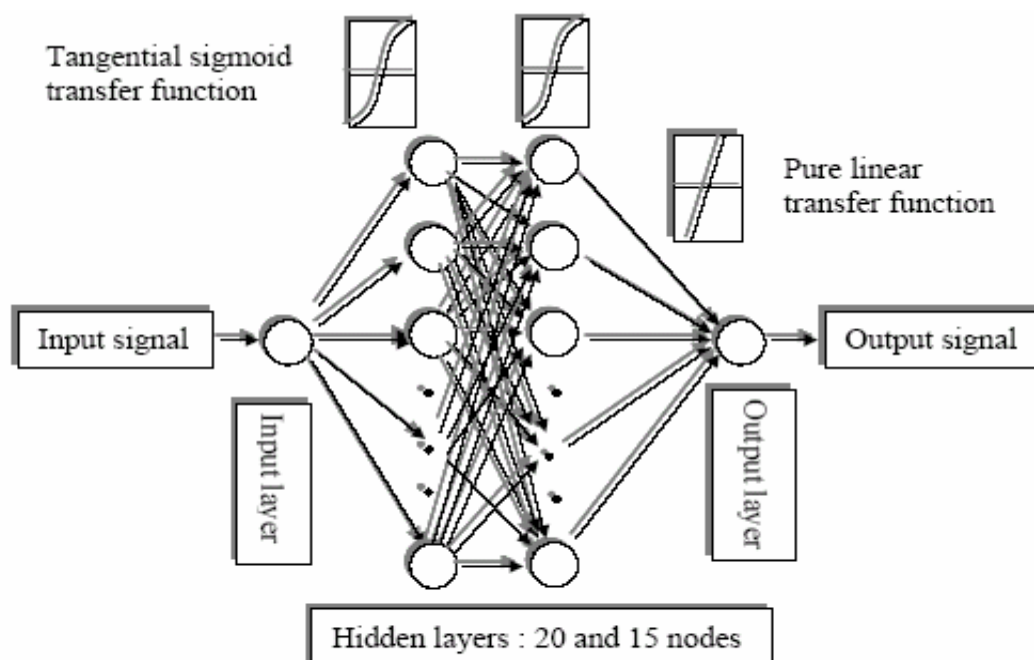


Figure 3 Architecture of the Elman network

As stated previously, the choice of the type of the neural network is dictated by the nature of the subject matter. The number of nodes in the hidden layer, however, is determined by experiments. With two hidden layers of 20 and 15 neurons, the network reached its optimal convergence limit. Additional neurons did not diminish further the error for most cases.

### 3.2 Input and output Data base

A relevant issue in the development of neural-network-based models is the choice of the input data set to be used during the training process. Since the neural network cannot be used to extrapolate, the training pattern should go at least to the edges of the problem domain in all dimensions. For structural response extraction, the ideal situation would be to train the neural network model with recorded sound signals that represent all experimental conditions. This is obviously impossible, as recording environments cannot be all known. However, it is reasonable to expect that the nature of distortions that affect the acquired signals in identical controlled recording conditions would be similar in an average sense. Within the scope of this work, the experiments were conducted in similar conditions. Hence, the input to the neural network was taken as segments of the sound signals recorded during the free vibration of the different specimen cables. The output of the neural network is the corresponding acceleration time histories calculated at mid-span of the cables. The data is input as cell array containing a sequence of inputs, and the network produced a cell array containing a sequence of outputs. Moreover, in order to improve the generalization performance of the neural networks, a dimensionless representation preprocessing steps were performed on the network inputs and targets. The training sets of input-output signals were randomly selected from the records representing strong, moderate and low vibration intensities with different degrees of noise alteration (signal-to-noise ratio). The segments vary from short signals of few periods with 140 samples to long signals having more than a hundred periods with up to 1400 samples. Typical input-output sets that will be further processed are presented in Figure 4, 5 and 6.



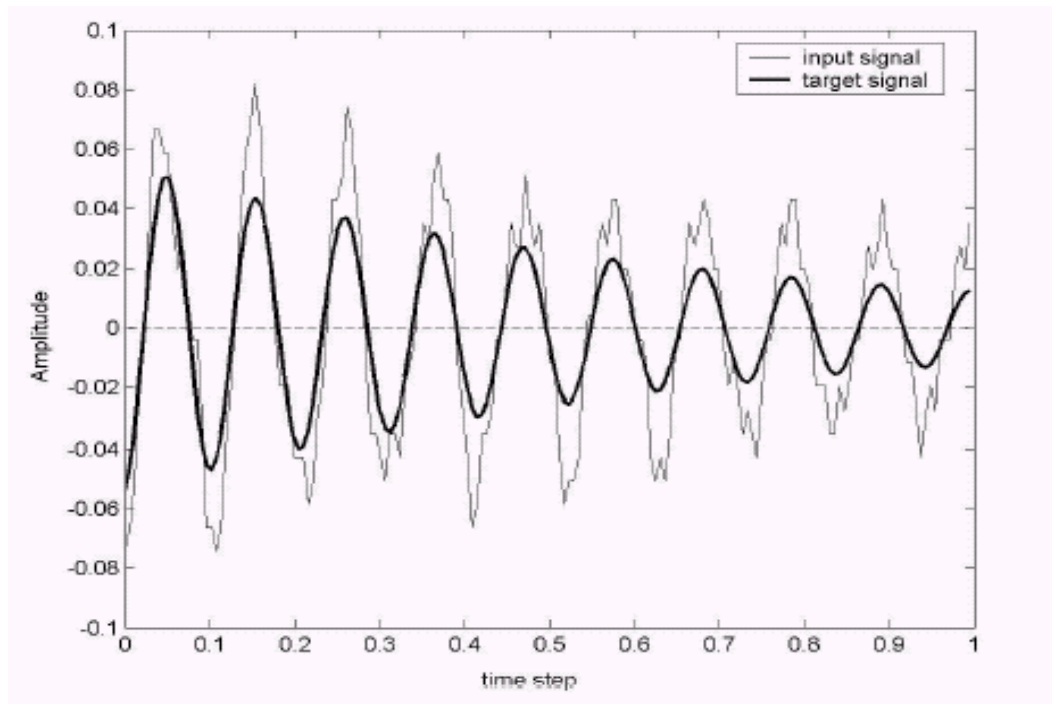


Figure 4 Typical short duration input – target signal

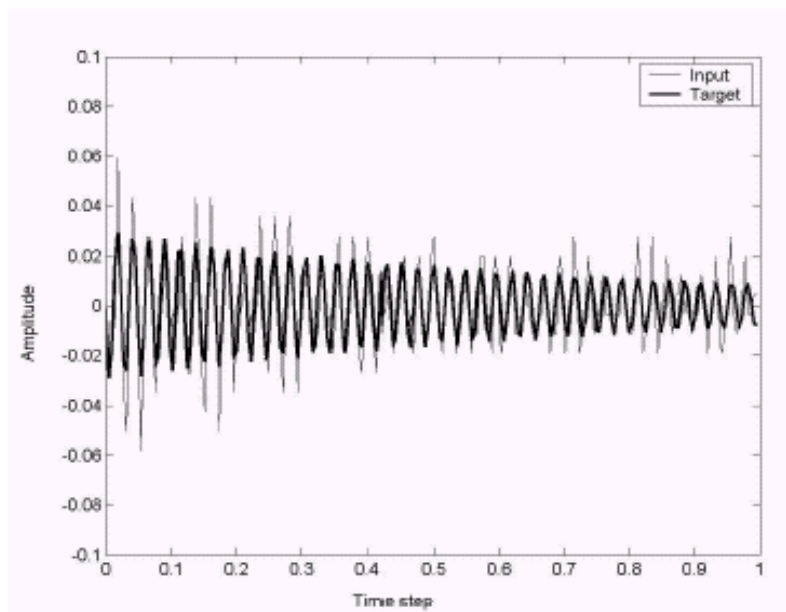


Figure 5 Typical medium duration input – target signal

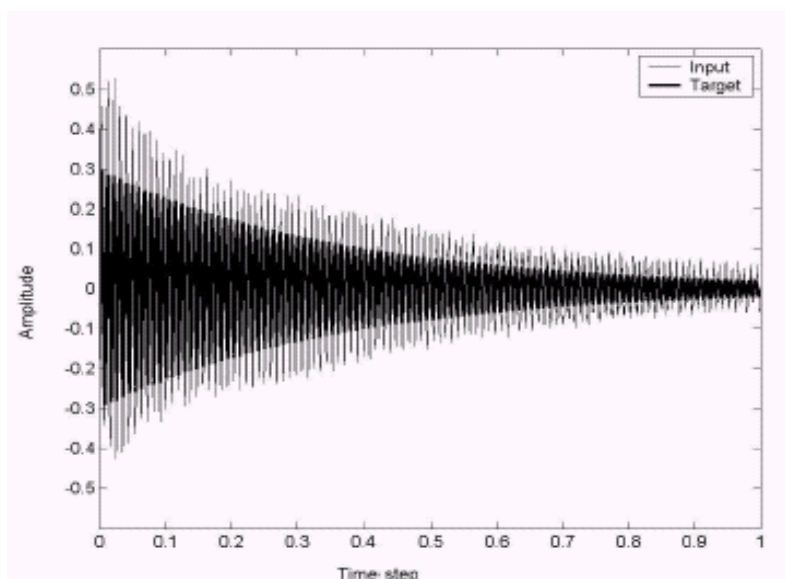


Figure 4 Typical long duration input – target signal

### 3.3 Training the neural network

It is difficult to know which training algorithm will be the fastest for a given problem. It will depend on many factors, including the complexity of the problem, the number of data points in the training set, the number of weights and biases in the network, and the error goal. For this particular application where accurate training is required, several test run were made for different training algorithm available in MATLAB [11] and found that Levenberg -Marquardt algorithm has the fastest convergence. The performance function that is used for training the neural network is the mean sum of squares of the network errors. It should be noted that no significant generalisation improvement is obtained in modifying the performance function by adding the mean of the sum of squares of the network weights and biases except that the network response became smoother.

The errors on the training sets are driven to relatively small values with a particular attention made to avoid overfitting. The entire input sequence is presented to the network, and its outputs are calculated and compared with the target sequence to generate an error sequence. For each time step the error is backpropagated to find *gradients* of errors for each weight and bias. This *gradient* is actually an approximation since the contributions of weights and biases to errors via the delayed recurrent connection are ignored. This gradient is then used to update the weights with the backpropagation training function. The example training curve shown in Figure 7

reveals that the performance of the network converges rapidly during the first few epochs and reached a limit after a small number of iterations.

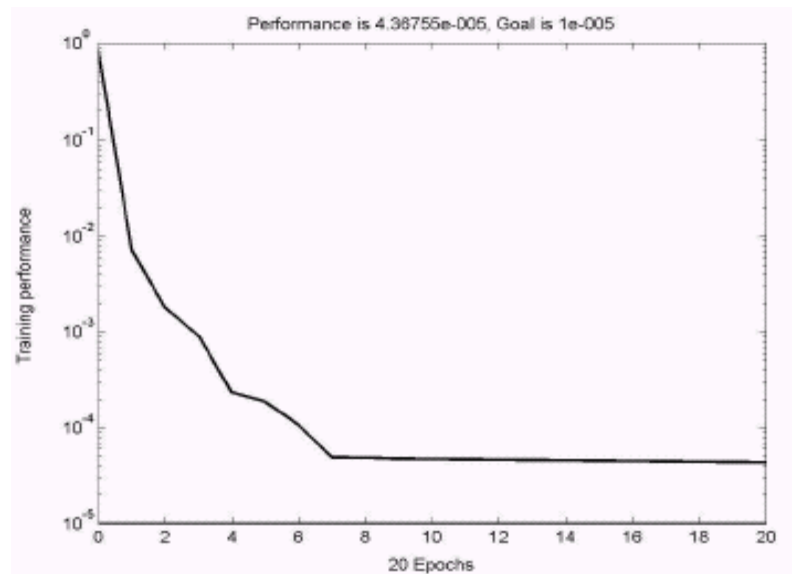


Figure 7 Training performance curve

### 3.4 Network testing and validation

Effectiveness of the proposed neural network model is evaluated on the basis of the capability of the network on simulating accurately the correlation between acceleration and the sound emitted from the vibrating structure. After the neural network is fully trained for a given set of input-output pairs, it is tested using signal that does not belong to the training set. Representative a validation results are presented and discussed hereafter. For short duration signals, the neural network performs remarkably well and simulates closely the target signal. The acceleration time history predicted by the neural network is plotted simultaneously with the recorded sound signal and the target signal in Figure 8.

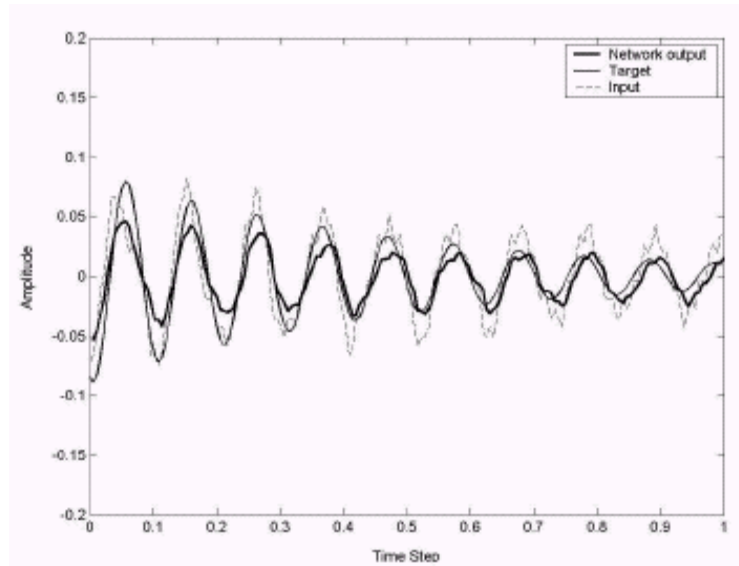


Figure 8 Acceleration time history predicted by the neural network

To check the quality of the network training a linear regression between the network response and the target is performed and a correlation coefficient (R value) between the network response and the target is computed. A plot of the best fit of all pairs of the network output (A) and the target (T) is presented in Figure 9. Perfect network prediction suggests that all data points are aligned along the diagonal ( $A = T$ ) and R value would be equal to one ( $R = 1$ ). For this example the best fit line is very close to the diagonal and the correlation coefficient approaches unity ( $R = 0.9$ ).

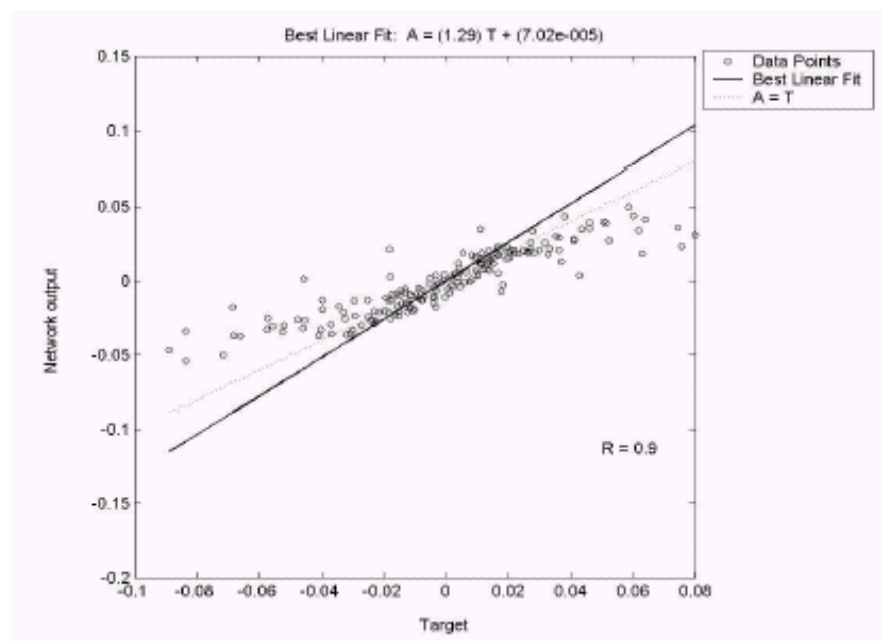


Figure 9 Plot of a linear regression between network response and target

As an example of long duration signal prediction, a graphical gauge of how well the neural network emulates the time history acceleration is presented in Figure 10. The target and the predicted signals are very similar for most instant of time. During portion of the histories shown in part (a), the network has successfully predicted the amplitudes of the acceleration vibration. In part (b), a discrepancy is noticeable as the predicted signal amplitudes are higher.

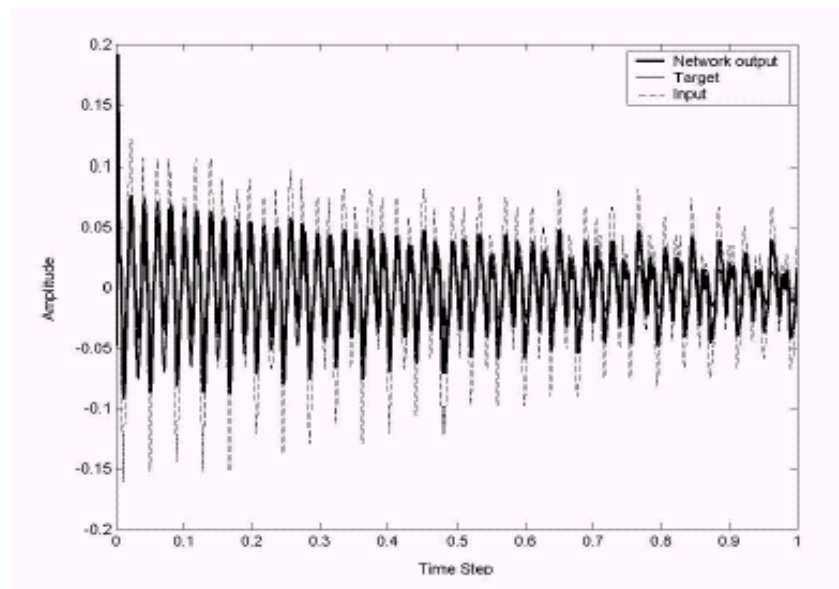


Figure 10 Acceleration time history predicted by the neural network

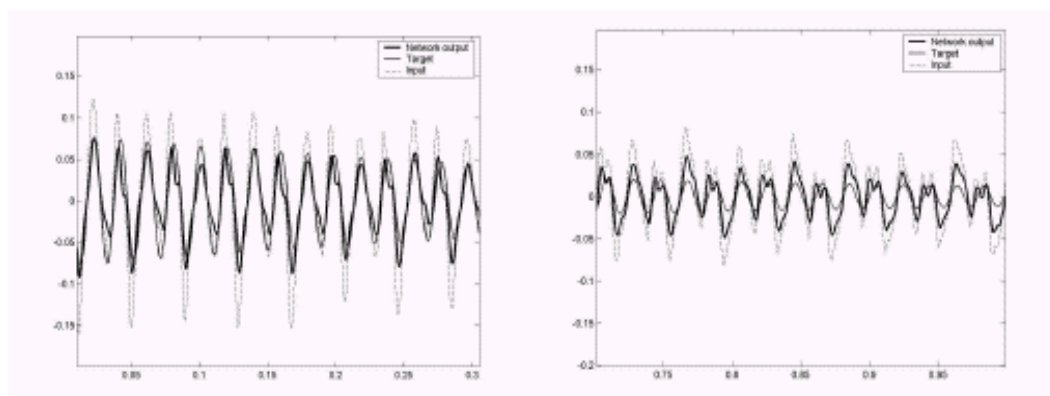


Figure 11 Part (a) of the predicted signal    Figure 12 Part (b) of the predicted signal

The correlation coefficient and the plot of the linear regression between the network response and the target are shown in Figure 13. In spite of the large number of points, the network performs reasonably well.

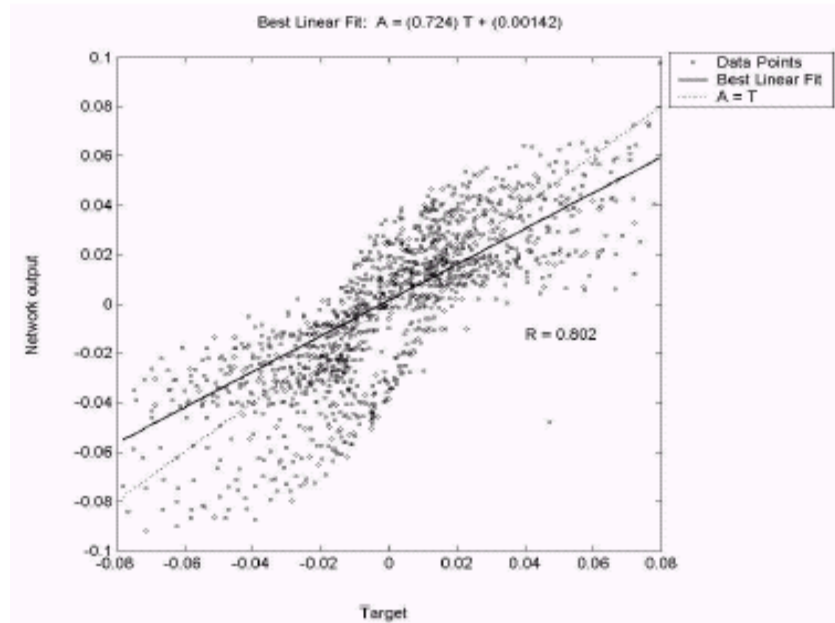


Figure 13 Plot of a linear regression between network response and target

In the previous validation examples, the network predicts accurately the phase; whereas in the following test, the predicted signal is not in phase with the target signal. The difference is clearly noticeable in part (b) of the time history because of the cumulated phase delay.

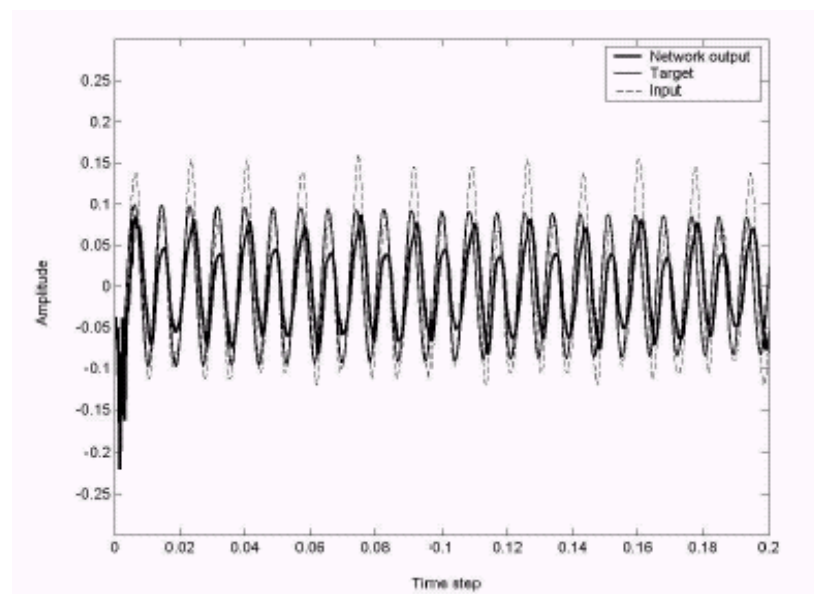


Figure 14 Part (a) of a predicted time history with phase delay

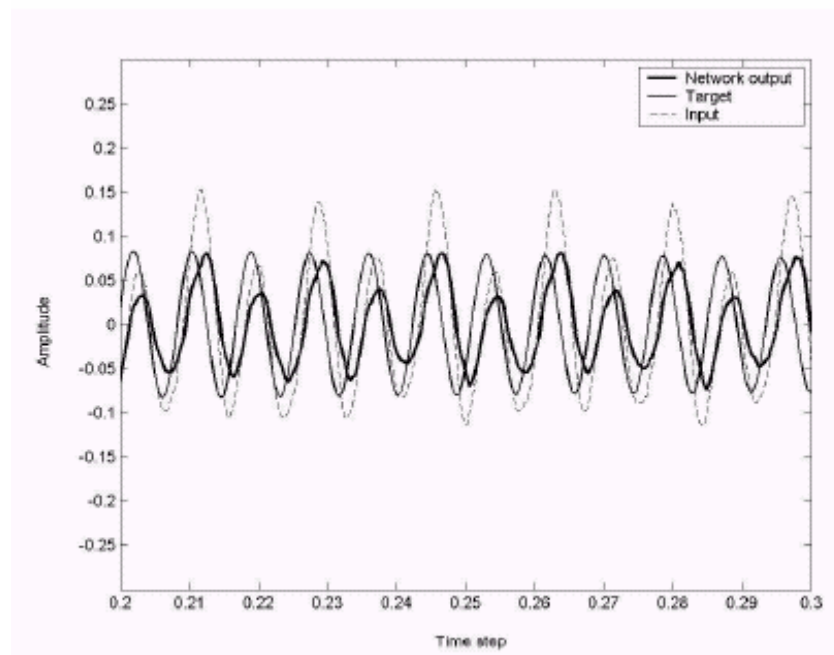


Figure 15 Part (b) of a predicted time history with phase delay

#### 4 Conclusion

The data reduction and accuracy of measurements are two issues of major concern in dynamic testing. This work presents a technique based on neural network analysis to extract specific information from recorded signal. For this purpose a simple physical model combined with numerical results is used to constitute a data base for training and validating a multi layer backpropagation neural network. The results although, corresponding to specific testing conditions, have shown an interesting potential, in which good agreement can be found between target and predicted signals. The experiment carried out in the present study is interesting in situation in which fast demonstration for educational purposes are needed. The approach, however, can be easily adapted to other applications of wave propagation through undefined media for which additional accuracy or noise extraction is sought.

#### Acknowledgements

This work has been supported by a grant CNEPRU No J 0901 / 04 / 01 / 2000. Part of the data processing has been carried out during the many visits of the first author to the University of Bristol, for which the author thanks Prof. C. Taylor for the simulating discussions and the cooperation.

## References

- [1] Williams, M. S., Blakeborough, A., Clement, D. and Bourahla, N., “ Seismic behaviour of knee braced frames”, Proc. of the Institution of Civil Engineers, Structures and Buildings, Vol. 152, No. 2, pp147-155, 2002.
- [2] Felber, A. and Cantieni, R., “Advances in ambient vibration testing: Ganter Bridge, Switzerland”, Structural Engineering International, 6(3), 187-190, 1996.
- [3] Adeli, H. and Park, H. S., “Counterpropagation Neural Networks in Structural Engineering”. Journal of Structural Engineering, Vol.121, No.8, pp. 1205-1212, 1995.
- [4] Adeli, H. and Asim, A., “Neural Network Model for Optimization of Cold-Formed Steel Beams”. Journal of Structural Engineering, Vol.123, No.11,pp.1535-1543, 1997.
- [5] Berrais, A., “Neural Networks in Structural Engineering: State of the art”,Advances in Computational Structures Technology, ed. B.H.V. Topping,Civil-Comp Press, Edinburgh, pp.93-101, 1996.
- [6] Wu, X., Ghaboussi, J. and Garrrrett, J. H., “Use of Neural Networks in Detection of Structural Damage”, Computers and Structures Vol. 42, No 4, pp. 649-659, 1992.
- [7] Marwala, T. and Hunt, H. E. M., “Fault identification using finite element models and neural networks”, Mechanical Systems and Signal Processing Vol.13, No 3, pp. 475-490, 1999.
- [8] Mukherjee, A., “A Fuzzy-neuro Health Monitoring Tool Using Transient Vibration Response”, Proceedings of VETOMAC-1, Bangalore, INDIA, pp. 25-27, 2000.
- [9] Goh, A. T., “Probabilistic Neural Network for Evaluating Seismic Liquefaction Potential”, Canadian Geotechnical Journal, Vol. 39, pp219-232, 2002.
- [10] Enescu, N., “Seismic Data Processing Using Nonlinear Prediction and Neural Networks”, Proc. IEEE NORISIG Symposium, Espoo, Finland, 1996.
- [11] Neural Network Toolbox User’s Guide: For Use with MATLAB (1984-2001), <http://www.mathworks.com>.



## REFERENCES

1. A. S. Moghadam and W. K. Tso "DAMAGE ASSESSMENT OF ECCENTRIC MULTISTORY BUILDINGS USING 3-D PUSHOVER ANALYSIS", 11th World Conference on Earthquake Engineering, paper 997, Mexico, 1996
2. Ashraf Habibullah, and Stephen Pyle, "PRACTICAL THREE DIMENSIONAL NONLINEAR STATIC PUSHOVER ANALYSIS", <http://www.csiberkeley.com> (Published in Structure Magazine, Winter 1998)
3. Bassam.A Izzuddin " CONCEPTUAL ISSUES IN GEOMETRICALLY NONLINEAR ANALYSIS OF 3D FRAMED STRUCTURES ",Computer Methods in Applied Mechanics and Engineering, Vol.191, pp.1029-1053, 2001
4. Centre National de Génie Parasismique (CGS), "REGLEMENT PARASISMIQUE ALGERIEN RPA99", 1999
5. C.Minowa, N.Ogawa, T.Hayashida, I.Kogoma, T.Okada "DYNAMIC AND STATIC COLLAPSE TESTS OF REINFORCED –CONCRETE COLUMNS" Engineering and Design, Vol. 156, pp. 269–276, 1995.
6. C. Touzet, " Les réseaux de neurons artificiels : introduction au connexionnisme », juillet 1992
7. C.ZAVALA, K.OHI, K.TAKANASHI "NEURO-HYBRID SUBSTRUCTURING ON-LINE TEST ON MOMENT RESISTANT FRAMES", 11th World Conference on Earthquake Engineering, paper 1387, Mexico, 1996
8. H.I. Hansen, S.R.K Nielson, and P.Thoft-Christensen "ACTIVE VIBRATION CONTROL OF HYSTERETIC OSCILLATORS USING NEURAL NETWORKS", 11th World Conference on Earthquake Engineering, paper 1266, Mexico, 1996
9. Hiratsuka K, and Shingu K, "DYNAMIC RESPONSE OF CONICAL SHELL USING NEURAL-NETWORK BASED VIBRATION CONTROL", 12th World Conference on Earthquake Engineering, Paper 450, New Zealand, 2000
10. G. Habibagahi, M. Taherian "PREDICTION OF COLLAPSE POTENTIAL VIA ARTIFICIAL NEURAL NETWORKS", Proceedings of the Sixth International Conference on the Application of Artificial Intelligence to Civil and Structural Engineering, paper. 33, Scotland, Sept 2003
11. Gian Michele CALVI And Martin RECLA, "ASSESSMENT OF CURRENT PREDICTION CAPACITIES OF THE RESPONSE OF EXISTING REINFORCED CONCRETE BUILDINGS", 12th World Conference on Earthquake Engineering, paper 2799, New Zealand, 2000

12. Jacob M EISENBERG, « GRAVITATIONAL SEISMIC COLLAPSE MECHANISM ANALYSIS IN VIEW OF CONCEPTUAL DESIGN », 12th World Conference on Earthquake Engineering, paper 245, New Zealand, 2000
13. Ju-Won Oh, In-Won Lee, Ju-Tae Kim, and Gyu-Won Lee "APPLICATION OF NEURAL NETWORKS FOR PROPORTIONING OF CONCRETE MIXES".ACI Materials Journal, Vol. 96, No. 1, pp. 61-67, Jan-Feb 1999
14. Khaldoon BANI-HANI, Jamshid GHABOUSSI And Stephen P SHNEIDER "EXPERIMENTAL STUDY OF PARAMETRIC AND NON-PARAMETRIQUE SYSTEM IDENTIFICATION USING NEURAL NETWORKS", 12th World Conference on Earthquake Engineering, paper 1387, New Zealand, 2000
15. Manolis Papadrakakis, Nikos D.Lagaros, Yiannis Tsompanakis "STRUCTURAL OPTIMIZATION USING EVOLUTION STRATEGIES AND NEURAL NETWORKS", Computer methods in applied mechanics and engineering. No. 156, pp. 309-333, 1998
16. Minehiro Nishiyama, Fumio Watanabe " CONTROLLED YIELD SEQUENCE OF REINFORCEMENT IN CONCRETE MEMBERS", International Workshop on Performance-Based Building Structural Design, JAPAN, November 2000
17. Neural-Networks Toolbox User's Guide: For use with MATLAB (1984-2001). <http://www.mathworks.com>.
18. O. Moselhi "NEURAL NETWORK APPLICATIONS IN CIVIL ENGINEERING" Proceeding of the Second International Conference in Civil Engineering on Computer applications, Research and Practice. pp 7-13.
19. Papadrakakis M., Papadopoulos V., and Lagaros N. D., "OPTIMUM DESIGN OF 3D STRUCTURES UNDER STATIC AND DYNAMIC LOADING", Computational Methods for Shell and Spatial Structures, pp 1-23, IASS-IACM 2000
20. Papadrakakis M., Papadopoulos V., and Lagaros N. D., "STRUCTURAL RELIABILITY ANALYSIS OF ELASTO-PLASTIC STRUCTURES USING NEURALE-NETWORKS AND MONTE CARLO SIMULATION", Journal of Computer Methods in Applied Mechanics and Engineering, No 136, pp. 145-163, 1996
21. P. Hajela, and L. Berke, "NEUROBIOLOGICAL COMPUTATIONAL MODELS IN STRUCTURAL ANALYSIS AND DESIGN", Computers & Structures, Vol.41, No.4, pp.657-667, 1991
22. Qingxiang Wang, Guofan Zhao, Liyan Lin " DUCTILITY OF HIGH STRENGTH REINFORCED CONCRETE COLUMNS" Engineering and Design, Vol. 156, pp. 75–81, 1995
23. Rahal K. N., and Abu kief M. , "NEURAL-NETWORK FOR CALCULATION OF SHEAR STRENGTH OF REINFORCED CONCRETE BEAMS", Kuwait Journal of Science, Vol. 26, No 3, pp. 239-251, 1999
24. Raul D. Bertero, and Vitelmo V. Bertero "REDUNDANCY IN EARTHQUAKE-RESISTANT DESIGN", Journal of Structural Engineering, pp. 81-88, Jan 1999

25. Renzo MEDEOT, "NEW DESIGN APPROACHES BASED ON ENERGY CONCEPTS AND SEISMIC HARDWARE", 12th World Conference on Earthquake Engineering, paper 1836, New Zealand, 2000
26. Sakai Junishi, Matsui Chiaki and Yushizumi Makoto "EFFECT OF COLLAPSE MODES ON EARTHQUAKE RESISTANT PROPERTIES FOR STEEL FRAMES", 11th World Conference on Earthquake Engineering, paper 779, Mexico, 1996
27. Sameh S. F. Mehanny and Gregory G. Deierein, "SEISMIC DAMAGE AND COLLAPSE ASSESSMENT OF COMPOSITE MOMENT FRAMES", Journal of Structural Engineering, Vol 127, No 9, pp. 1045-1053, 2001
28. SAP2000, Integrated Finite Elements Analysis and Design of Structures DETAILED TUTORIAL INCLUDING PUSHOVER ANALYSIS, Computers and Structures, Inc. Berkeley, California, USA. <http://www.csiberkeley.com>
29. Schilling G. D. " MODELING AIRCRAFT FUEL CONCEPTION WITH NEURAL NETWORKS", Thesis for the Degree of Master of Science in Civil Engineering, Virginia Polytechnic Institute and State University, August 1997
30. Sergio Lai and Mauro Serra "CONCRETE STRENGTH PREDICTION BY MEAN OF NEURAL NETWORK". Construction and Buildings Materials, Vol. 11, No. 2, pp. 93-98, 1997
31. S.M.R. EMAMI, Y. IWAO and T.HARADA "A METHOD FOR PREDICTION OF PEAK HORIZONTAL ACCELERATION BY ARTIFICIAL NEURAL NETWORK", 11th World Conference on Earthquake Engineering, paper 1238, Mexico 1996
32. Terrence F. Paret, Kent K. Sasaki, Daniel H.Eilbeck and Sigmund A. Freeman "APPROXIMATE INELASTIC PROCEDURES TO IDENTIFY FAILURE MECHANISMS FROM HIGHER MODE EFFECTS", 11th World Conference on Earthquake Engineering, paper 966, Mexico, 1996.
33. Tsuyoshi TAKADA And Takashi NAKANO, "SEISMIC LOAD EFFECT DIRECTLY LINKED TO SPECIFIED COLLAPSE MECHANISMS IN ULTIMATE LIMIT STATE DESIGN", 12th World Conference on Earthquake Engineering, Paper 2560, New Zealand, 2000
34. W.B. Kratzig, Y.S. Petryna, F. Stangenberg, "MEASURES OF STRUCTURAL DAMAGE FOR GLOBAL FAILURE ANALYSIS", International Journal of Solids and Structures, Vol. 37, pp. 7393-7407, 2000
35. Wei Lu, "NEURAL-NETWORK MODEL FOR DISTORTIONAL BUCKLING BEHAVIOUR OF COLD-FORMED STEEL COMPRESSION MEMBERS", Thesis for the Degree of Licentiate of Science in Technology, Helsinki University of Technology, 2000
36. Wing Ho Cheung "NEURAL NETWORK AIDED AVIATION FUEL CONSUMPTION MODELING", Thesis for the Degree of Master of Science in Civil Engineering, Virginia Polytechnic Institute and State University, August 1997

37. W.S. Choi, G.J. Park" STRUCTURAL OPTIMIZATION USING EQUIVALENT STATIC LOADS AT ALL TIME INTERVALS" computer .methods Applied mechanics and .engineering, vol.191, pp. 1077-2094, 2002
38. W. T. Yeung , " BRIDGE MONITORING STRATEGY USING MEASURED DYNAMICS RESPONSE AND NATURAL NETWORKS", Thesis for the Degree of Doctor of Philosophy, University of Bristol, UK, Dec 1999
39. Yixin LI, Shichun Zhao" SEISMIC DAMAGE ANALYSIS OF CONCRETE MEMBERS" Nuclear Engineering and Design, Vol. 160, pp. 261–266, 1996.
40. Bourahla.N Taleb.R & Boukhamacha.T « Structural Response Extraction from Sound Vibration Measurements using Neural Network ». Proceeding of 7th International Conference on The Application of Artificial Intelligence to Civil and Structural Engineering. Stirling, Scotland. Paper 63. (Sept 2003)
41. Bourahla.N & Taleb.R « Prédiction des Mécanismes de Ruine dans les Portiques sous Action Sismique par les Réseaux de Neurones ». Colloque international : Risque, Vulnérabilité & Fiabilité dans la Construction. Alger, Algérie. Paper 33. (Oct 2003)