

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À  
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE  
DE LA MAÎTRISE EN ÉLECTRONIQUE INDUSTRIELLE

PAR  
SYLVIE LEGENDRE

ÉTUDE DE L'APPLICATION DES RÉSEAUX DE NEURONES MULTICOUCHES  
POUR L'ÉTALONNAGE ET LA RECONSTITUTION DE MESURANDES  
DANS UN SYSTÈME DE MESURE

DÉCEMBRE 1994

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

"À Daniel, mes parents, mon frère et ma sœur"

## RÉSUMÉ

Le traitement d'un signal mesuré dans un système de mesure se réalise en deux phases: conversion et reconstitution. La conversion dans un système de mesure électrique consiste en une série de transformations de signaux reçus d'un objet de mesure en signaux électriques, standards, préférablement numériques. La reconstitution de signaux à mesurer constitue un problème fondamental dans la métrologie. Elle consiste à estimer un signal  $x$ , qui n'est pas mesurable directement, à partir des résultats du mesurage d'un autre signal  $\tilde{y}$ , qui est lié avec le premier de façon causative. Pour effectuer la reconstitution, le système de mesure doit être étalonné. L'étalonnage consiste à l'identification du modèle de la relation entre la grandeur mesurée et les résultats de la conversion et à l'estimation des paramètres de ce modèle. Nous envisageons une méthode non analytique d'étalonnage d'un système de mesure et de reconstitution de mesurandes indépendante de la réponse impulsionnelle du système de mesure et basée sur les réseaux de neurones et leurs performances en reconnaissance de formes et classification.

L'application permettant de valider la méthode d'étalonnage et de reconstitution à l'aide d'un réseau de neurones est un système de mesure de pression, composé principalement de trois capteurs de pression à fibre optique et dont la caractéristique d'entrée/sortie est quasi-sinusoidale, et qui subissent une forte influence de la température. Parmi les nombreux types de réseaux de neurones, les réseaux multicouches ont été retenus car ils présentent une topologie qui s'accorde avec le type de données à traiter (vecteur à l'entrée, scalaire en sortie), leurs capacités d'apprentissage et de stockage sont élevées et ils sont relativement facile d'utilisation.

Afin d'améliorer la qualité de l'apprentissage du réseau de neurones et de disposer d'un réseau fonctionnel pour l'application envisagée, l'accent est mis sur la mise au point d'un nouvel algorithme d'apprentissage performant permettant un apprentissage rapide et robuste. Cet algorithme est obtenu à partir d'une version modifiée de la méthode standard de la rétropropagation du gradient déjà existante, cette version modifiée utilise le filtre de Kalman pour l'estimation des paramètres du réseau. Les trois algorithmes, soient la méthode de la rétropropagation du gradient et ses deux dérivées, sont testés et comparés en phase d'apprentissage.

Les résultats d'étalonnage et de reconstitution de mesurandes pour le système de mesure de pression à fibres optiques ont été réalisés pour des données synthétiques, à température constante et variable, et réelles. Les résultats obtenus montrent la flexibilité dont font preuve les réseaux de neurones comme outils d'étalonnage dans le cas d'un système de mesure fortement non-linéaire et sensible à des facteurs d'influence, la température dans le cas étudié. Nous montrons qu'il n'est pas nécessaire de mesurer la température et de la traiter comme donnée d'entrée, le réseau de neurones étant capable de tirer l'information sur la température directement des caractéristiques d'entrée/sortie des trois capteurs. En effet, durant la phase d'apprentissage, le réseau apprend l'évolution des données utilisées pour l'étalonnage du système de mesure et mémorise l'effet de la température sur cette évolution. Par la suite, en exploitant la capacité de généralisation des réseaux de neurones en phase de reconnaissance, il est possible de reconstituer des pressions à partir des tensions de sortie des capteurs de pression, sans avoir à mesurer la température.

Pour valider les résultats d'étalonnage et de reconstitution, une comparaison avec ceux obtenus avec la méthode basée sur les fonctions spline est proposée. Les points de comparaison concernent les résultats de reconstitution de la pression et l'implantation

matérielle dans un microcontrôleur MC68HC11 des deux méthodes en terme de précision de reconstitution, de temps de calcul et d'espace mémoire.

Les résultats de l'étude suggèrent de plus une possibilité intéressante de développement futur, soit que le même système de mesure, ayant la même implantation matérielle, pourrait être utilisé pour mesurer la température simultanément avec la mesure de la pression. Autrement dit, cela reviendrait à considérer la température comme un deuxième mesurande à reconstituer à partir des résultats de conversion, soient les trois tensions de sortie des capteurs de pression qui sont aussi des fonctions de la température.

## REMERCIEMENTS

Je tiens à remercier le Professeur A. Barwicz pour m'avoir donné la possibilité de travailler au Laboratoire de Systèmes de Mesures en acceptant de diriger ma maîtrise. Je le remercie pour son aide financière durant mon année de recherche au laboratoire.

Ma plus profonde gratitude s'adresse à Daniel Massicotte pour son aide constante, son soutien, ses encouragements tout au long de ce travail. Merci pour toutes les bonnes idées et les conseils dans les différentes étapes de ma recherche et qui m'ont permis de compléter ce travail.

Je tiens à remercier ma famille, particulièrement mes parents qui m'ont aidé durant toutes mes années d'études en France et au Québec. Je remercie également les professeurs de la filière E.E.A. de l'Université de Rouen, France, et qui m'ont donné l'opportunité de compléter mes études à l'Université du Québec à Trois-Rivières.

Je n'oublie pas mes collègues et amis du Laboratoire de Systèmes de Mesure, en particulier Pierre, Louis et Philippe pour leurs conseils et les bons moments passés au laboratoire.

# TABLE DES MATIÈRES

	<b>page</b>
RÉSUMÉ . . . . .	iii
REMERCIEMENTS . . . . .	vi
TABLE DES MATIÈRES . . . . .	vii
LISTE DES SYMBOLES ET ABRÉVIATIONS . . . . .	xi
LISTE DES FIGURES . . . . .	xviii
LISTE DES TABLEAUX . . . . .	xxii

## CHAPITRES

1. INTRODUCTION . . . . .	1
1.1 Problématique . . . . .	1
1.2 Objectifs . . . . .	2
1.3 Structure du mémoire . . . . .	3
2. SYSTÈME DE MESURE . . . . .	5
2.1 Introduction . . . . .	5
2.2 Définitions de base . . . . .	5
2.2.1 La mesure . . . . .	5
2.2.2 Le système de mesure . . . . .	7
2.3 Étalonnage statique d'un système de mesure . . . . .	9
2.3.1 Définition . . . . .	9
2.3.2 Méthodes d'étalonnage statique . . . . .	9



2.4	Reconstitution de mesurandes . . . . .	11
2.4.1	Définition . . . . .	11
2.4.2	Classification des méthodes de reconstitution . . . . .	13
2.4.3	Approche proposée . . . . .	16
2.5	Conclusion . . . . .	17
3.	SYSTÈME DE MESURE DE PRESSION À FIBRES OPTIQUES . . . . .	19
3.1	Introduction . . . . .	19
3.2	Capteur de pression à fibre optique . . . . .	19
3.2.1	Principe de fonctionnement . . . . .	19
3.2.2	Caractéristique d'entrée/sortie du capteur . . . . .	23
3.3	Système de mesure de pression . . . . .	25
3.3.1	Principe de mesure . . . . .	25
3.3.2	Description du système de mesure . . . . .	28
3.4	Conclusion . . . . .	29
4.	LES RÉSEAUX DE NEURONES . . . . .	31
4.1	Introduction . . . . .	31
4.2	Notions et définitions de base . . . . .	32
4.2.1	Fondements biologiques . . . . .	32
4.2.2	Le neurone formel . . . . .	33
4.2.3	Topologies des réseaux de neurones . . . . .	36
4.2.4	L'apprentissage des réseaux de neurones . . . . .	39
4.3	Principaux types de réseaux de neurones . . . . .	42
4.3.1	Le perceptron: 1958 . . . . .	43
4.3.2	Adaline et madaline: 1960 . . . . .	45

4.3.3	Le réseau de Hopfield: 1982	47
4.3.4	Le réseau de Kohonen: 1984	49
4.3.5	Les réseaux multicouches: années 1980	51
4.3.6	Les réseaux ART (Adaptive Resonance Theory): 1987	56
4.4	Propriétés et applications des réseaux de neurones	58
4.4.1	Propriétés des réseaux de neurones	58
4.4.2	Propriétés des modes d'apprentissage	59
4.4.3	Domaines d'applications des réseaux de neurones	60
4.5	Conclusion	62
5.	RÉSEAUX DE NEURONES MULTICOUCHES	65
5.1	Introduction	65
5.2	Notions de base	66
5.2.1	Structure des réseaux multicouches	66
5.2.2	Apprentissage et reconnaissance	68
5.2.3	Données d'apprentissage	70
5.3	Les algorithmes d'apprentissage	72
5.3.1	La rétropropagation du gradient (RETRO_GRAD)	74
5.3.2	Méthode d'optimisation basée sur le filtrage de Kalman (RETRO_KAL)	78
5.3.3	Nouvelle méthode proposée (RETRO_KAL+)	84
5.4	Conclusion	91

6.	RÉSULTATS D'ÉTALONNAGE ET DE RECONSTITUTION DE MESURANDES UTILISANT LES RÉSEAUX DE NEURONES . . . . .	93
6.1	Introduction . . . . .	93
6.2	Définition du problème . . . . .	94
6.2.1	Étalonnage et reconstitution de mesurandes . . . . .	94
6.2.2	Génération des signaux synthétiques . . . . .	96
6.2.3	Réalisation des tests . . . . .	98
6.3	Résultats d'étalonnage et de reconstitution de pression: données synthétiques . . . . .	101
6.3.1	Résultats à température fixe . . . . .	101
6.3.2	Résultats à température variable . . . . .	107
6.4	Résultats d'étalonnage et de reconstitution de pression: données expérimentales . . . . .	116
6.5	Implantation matérielle . . . . .	118
6.6	Conclusion . . . . .	121
7.	CONCLUSION . . . . .	125
	RÉFÉRENCES . . . . .	129
	ANNEXE A: Programme d'initialisation du réseaux de neurones . . . . .	133
	ANNEXE B: Programme d'apprentissage avec les fonctions: RETRO_GRAD, RETRO_KAL, RETRO_KAL+ . . . . .	135
	ANNEXE C: Programme de généralisation . . . . .	145
	ANNEXE D: Programmes d'implantation dans le MC68HC11 . . . . .	147

## LISTE DES SYMBOLES ET ABRÉVIATIONS

### Symboles/ Description Abréviations

$A$	constante dépendant de la géométrie et du type de fibre $HB$ ,
$A$	paramètre de génération des signaux synthétiques,
$A_i$	activation du neurone $i$ ,
$A/N$	Analogue/Numérique,
$ART$	Adaptive Resonance Theory,
$B$	indice de biréfringence,
$B$	paramètre de génération des signaux synthétiques,
$C$	paramètre de génération des signaux synthétiques,
$C_p$	la constante relative de photoelasticité,
$d$	sortie désirée,
$d_i$	sortie désirée du neurone $i$ ,
$d_i$	terme de correction (rétropropagation du gradient),
$DC$	tension continue,
$e^t$	fonction exponentielle de $t$ ,
$e(k)$	erreur quadratique entre la sortie obtenue $\hat{y}_{n,m}$ et la sortie désirée $y_{n,m}$ ,
$e_i^l(k)$	erreur de propagation au niveau du neurone $i$ de la couche $l$ à l'itération $k$ ,
$etal$	exposant signifiant une donnée d'étalonnage,
$eval$	exposant signifiant une donnée d'évaluation de la capacité de généralisation,
$E$	module de Young,
$E_i$	signal d'erreur (Adaline),

$E^l(k)$	erreur quadratique entre la sortie obtenue $\hat{y}_{n,m}$ et la sortie désirée $y_{n,m}$ ,
$E_t$	erreur totale (Adaline),
$f(\cdot)$	fonction de décision non linéaire,
$f(\cdot)$	fonction de transfert d'un capteur à fibre optique,
$f'(\cdot)$	fonction de décision dérivée,
$f^{-1}(\cdot)$	fonction de décision inverse,
$f_i(\hat{p}:T)$	caractéristique du capteur $s_i$ ( $i = 1, 2, G$ ),
$h(\cdot)$	réponse impulsionnelle du système de mesure,
$h_k$	réponse impulsionnelle échantillonnée,
$H$	transformée de Fourier de la réponse impulsionnelle $h$ ,
<b>HB</b>	Hautement Biréfringente,
$i$	indice de numérotation d'un neurone ou d'une couche,
$I$	intensité de la lumière propagée,
$I_0$	amplitude de l'intensité de la lumière propagée,
$I$	matrice identité,
$k$	indice d'itérations,
$k$	constante positive (Perceptron),
$k$	numéro d'itération,
$k^l(k)$	vecteur gain de Kalman pour $l = 2, \dots, L$ ,
$l$	exposant de numérotation d'une couche,
$L$	nombre de couches,
$L$	longueur de la fibre,
$L_B$	longueur de battement,
$m$	indice de numérotation des pressions,
$M$	fonction <i>mesure</i> , transformation du domaine $Q$ vers $N$ ,
<b>MPa</b>	Mégapascal, unité de mesure de la pression,

$n$	coefficient de réfraction,
$n$	nombre d'observations pour l'étalonnage statique
$n$	indice de numérotation de température,
$N$	ensemble de nombres $(n_1 \dots n_n)$ , avec $n_i \in \mathbb{R}$ ,
$N/A$	Numérique/Analogue,
$N_j$	nombre de neurones $j$ connectés au neurone $i$ ,
$N^l$	nombre de neurones par couches pour $l = 1, \dots, L$ ,
$N/N$	Numérique/Numérique
$N_P$	nombre de températures,
$N_T$	nombre de pressions,
$P_{11}, P_{12}$	coefficients de contrainte optique,
$p$	pression appliquée,
$P_{i,m}^{etal}$	pression d'étalonnage,
$P_{i,m}^{eval}$	pression d'évaluation,
$P_{max}$	valeur maximale de la pression $p$ ,
$P_{min}$	valeur minimale de la pression $p$ ,
$P_{n,m}$	pression générée à la température $m$ et la pression $n$ ,
$P_x$	pression mesurée,
$P_0$	pression atmosphérique,
$P$	pression,
$P$	ensemble de relations numériques définies dans l'ensemble des nombres $(p_1 \dots p_n)$ ,
$Q$	classe des entités extramathématiques $(q_1 \dots q_n)$ ,
$\check{r}_i^l(k)$	variance de l'erreur au niveau du neurone $i$ de la couche $l$ à l'itération $k$ ,
$\check{r}_i^l$	paramètres de l'équation de la variance estimée ( $i = 1, 2, 3, a, e$ ),
$R$	ensemble de relations empiriques existantes dans la classe $(r_1 \dots r_n)$ ,

$s$	sortie obtenue,
$s_i$	sortie du neurone $i$ ,
$\mathbf{s}$	vecteur des sorties obtenues (dimension $m$ ),
$S_G$	capteur guide de pression,
$S_i$	capteur de pression ( $i = 1, 2$ ),
$t$	variable d'intégration temporelle,
$t$	instant,
$T$	température,
$T_i$	température $i$ ,
$T_i^{etal}$	température $i$ d'étalonnage,
$T_i^{eval}$	température $i$ d'évaluation,
$T_m$	température indicée $m$ ,
$T_p$	période de la fonction d'intensité versus la pression,
$T_0$	température de référence,
$v$	tension de sortie d'un capteur à fibre optique,
$v_G$	tension de sortie du capteur guide,
$v_{G,n,m}$	tension de sortie du capteur $S_G$ ,
$v_{G \max}$	valeur maximale de la tension $v_G$
$v_{G \min}$	valeur minimale de la tension $v_G$ ,
$v_i$	tension de sortie du capteur $i$ ,
$\hat{v}_i$	valeur de la tension $v_i$ ( $i = 1, 2, G$ ) pour une pression $p$ mesurée,
$v_{i \max}$	valeur maximale de la tension $v_1$ ,
$v_{i \min}$	valeur minimale de la tension $v_1$ ,
$v_i^l(k)$	bruit d'observation au niveau du neurone $i$ de la couche $l$ à l'itération $k$ ,
$v_0$	l'amplitude de la tension de sortie,
$v_{1,n,m}$	tension de sortie du capteur $S_1$ ,

$v_{2,n,m}$	tension de sortie du capteur $S_2$ ,
$w_i$	poids synaptique (Kohonen),
$w_{ij}$	poids de connexion du neurone $j$ vers le neurone $i$ ,
$w_{ij}^l(k)$	poids de connexion entre le neurone $j$ de la couche $l-1$ et le neurone $i$ de la couche $l$ ,
$w_i^l(k)$	vecteur de poids de connexion associé au neurone $i$ pour $l = 2, \dots, L$ ,
$W$	matrice des poids de connexion,
$W^l(k)$	matrice de poids de connexion pour $l = 2, \dots, L$ ,
$x$	variable pour la définition des fonctions de décision,
$x$	moyenne (fonction gaussoidale),
$x$	mesurande ou objet de mesure,
$\hat{x}$	objet reconstitué,
$x_i$	mesurande à la $i^{\text{ème}}$ mesure,
$x_i$	entrée élémentaire du neurone $i$ ,
$x_i^l(k)$	sortie du neurone $i$ de la couche $l$ à l'itération $k$ ,
$x_i^0$	$i^{\text{ème}}$ donnée d'entrée du réseau ( $i = 1, 2, 3, 4$ ),
$x_k$	mesurande échantillonné,
$x(t)$	mesurande à l'instant $t$ ,
$x$	vecteur des entrées (dimension $n$ ),
$x_{n,m}^0$	vecteur d'apprentissage d'entrée,
$x^p$	vecteur prototype constitué des éléments $x_i^p$ (Hopfield),
$X$	transformée de Fourier du signal d'entrée $x$ ,
$y$	sortie de la conversion,
$y$	donnée de sortie du réseau,
$\tilde{y}$	sortie de la conversion bruitée,
$y_i$	sortie de la conversion à la $i^{\text{ème}}$ mesure,



$\tilde{y}_k$	sortie de la conversion échantillonnée et bruitée,
$y_{n,m}$	sortie désirée associée à $x_{n,m}^0$ ,
$\hat{y}_{n,m}$	sortie obtenue après propagation de $x_{n,m}^0$ ,
$y(t)$	sortie de la conversion à l'instant $t$ ,
$y$	vecteur des sorties désirées (dimension $m$ ),
$Y$	transformée de Fourier du signal de sortie $y$ ,
$z_i$	entrée totale du neurone $i$ ,
$z_i^l(k)$	entrée totale du neurone $i$ de la couche $l$ à l'itération $k$ ,
$\alpha$	nombre positif (fonction linéaire),
$\alpha$	pente de la fonction sigmoïdale,
$\beta$	niveau de seuil (fonction à seuils),
$\beta_i$	paramètre d'une fonction de régression,
$\beta(s)$	fonction d'oubli (Kohonen),
$\beta_x, \beta_y$	constantes de propagation,
$\gamma$	valeur de saturation (fonction à rampe),
$\gamma$	une constante liée au type de fibre,
$\delta$	niveau de seuil (fonction à seuils),
$\delta$	phase initiale,
$\delta_{jk}$	symbole de Kronecker,
$\bar{\delta}\hat{p}_n$	erreur relative limite de propagation,
$\delta\hat{y}(k)$	erreur relative entre la sortie obtenue $\hat{y}_{n,m}$ et la sortie désirée $y_{n,m}$ ,
$\Delta\hat{y}(k)$	erreur absolue moyenne entre la sortie obtenue $\hat{y}_{n,m}$ et la sortie désirée $y_{n,m}$ ,
$\Delta p$	intervalle de pression,
$\Delta p$	erreur de calcul sur la pression,
$\Delta p_{n,m}$	nombre pseudo-aléatoire simulant le bruit sur la pression,

$\bar{\Delta} \hat{p}_{n,m}$	erreur absolue limite,
$\Delta v_{i,n,m}$	nombre pseudo-aléatoire simulant le bruit sur les tensions,
$\Delta T_m$	nombre pseudo-aléatoire simulant le bruit sur la température,
$\bar{\Delta} \bar{T}^{etal}$	erreur absolue limite sur les données de référence de la température,
$\Delta v$	erreur de mesure sur la tension,
$\Delta w_{ij}$	modification apportée aux poids de connexion,
$\Delta y$	erreur quadratique,
$\varepsilon_i$	erreur de moyenne zéro,
$\theta$	seuil (fonction à seuils),
$\theta_i$	seuil du neurone $i$ ,
$\lambda$	variable d'intégration,
$\lambda$	longueur d'onde,
$\mu$	paramètre d'apprentissage (Hebb) ou pas de rétropropagation,
$\mu(\cdot)$	fonction de régression,
$\mu^l(k)$	pas de rétropropagation pour la couche $l$ à l'itération $k$ ,
$\mu_i^l$	paramètres de l'équation du pas de rétropropagation ( $i = 1, 2, 3, a, e$ ),
$\nu$	constante de Poisson,
$\nu$	variance (fonction gaussoidale),
$\nu_k$	bruit additif à l'échantillon $k$ ,
$\Sigma^l(k)$	matrice de covariance pour $l = 2, \dots, L$ ,
$\sigma^2$	variance,
$\partial$	symbole de dérivation partielle,
$\sum$	opérateur de sommation,
$\int$	opérateur d'intégration,
$*$	opérateur de convolution,
$\hat{x}$	notation d'une donnée $x$ exacte (non bruitée),

## LISTE DES FIGURES

	<b>page</b>
Figure 2.1: Fonction mathématique de mesure . . . . .	6
Figure 2.2: Schéma général d'un système de mesure . . . . .	8
Figure 2.3: Système à identifier où $n$ est le nombre d'observations . . . . .	10
Figure 2.4: Représentation de la partie conversion du système de mesure . . . . .	13
Figure 2.5: Application des réseaux de neurones à l'étalonnage d'un système de mesure (a) et à la reconstitution de mesurandes (b) . . . . .	16
Figure 3.1: Configuration du capteur de pression à fibre optique en transmission (a) et en réflexion (b) . . . . .	22
Figure 3.2: Caractéristique d'entrée/sortie du capteur pour deux longueurs de fibre . . . . .	23
Figure 3.3: Influence de la température sur la caractéristique d'entrée/sortie du capteur, $T_1$ en pointillé et $T_2$ en continu avec $T_1 < T_2$ . . . . .	24
Figure 3.4: Représentation de l'erreur de mesure de pression [LEM95] . . . . .	25
Figure 3.5: Caractéristiques des trois capteurs $S_1$ , $S_2$ et $S_G$ du système de mesure . . . . .	27
Figure 3.6: Régions de sensibilité des capteurs $S_1$ et $S_2$ en gras et erreur relative des deux capteurs déphasés de $90^\circ$ [LEM95] . . . . .	27
Figure 3.7: Diagramme bloc du système de mesure de pression . . . . .	28

Figure 4.1:	Neurone biologique (a) et sa formalisation mathématique (b) . . . . .	32
Figure 4.2:	Fonction d'activation linéaire (a), à seuils (b), à rampe (c), sigmoïdale (d) et gaussienne (e) . . . . .	36
Figure 4.3:	Classification des types de réseaux de neurones selon leur topologie . . . . .	39
Figure 4.4:	Règle de Hebb . . . . .	42
Figure 4.5:	Schéma général du perceptron . . . . .	44
Figure 4.6:	Réseau Adaline (ADaptive LINEar NEuron) . . . . .	46
Figure 4.7:	Réseau entièrement connecté: modèle de Hopfield . . . . .	48
Figure 4.8:	Réseau entièrement connecté par couches: modèle de Kohonen . . . . .	50
Figure 4.9:	Principe de propagation et rétropropagation dans un réseau multicouche . . . . .	53
Figure 4.10:	Résonance adaptative dans un réseau de neurones à deux couches . . . . .	57
Figure 5.1:	Schéma de principe du système de mesure . . . . .	67
Figure 5.2:	Structure du réseau de neurones à trois couches . . . . .	68
Figure 5.3:	Données d'apprentissage d'entrée (a) et de sortie (b) . . . . .	72
Figure 5.4:	Courbe d'apprentissage avec l'algorithme RETRO_GRAD avec $\mu^2 = \mu^3 = 0.2$ . . . . .	78
Figure 5.5:	Neurones formels modifiés de la couche cachée (a) et de la couche de sortie (b) . . . . .	79
Figure 5.6:	Courbe d'apprentissage avec l'algorithme RETRO_KAL: (a) $\mu^2$ variable et $\check{r}^2 = \check{r}^3 = 0.2$ , (b) $\check{r}^2 = \check{r}^3$ variables et $\mu^2 = 2$ . . . . .	83

- Figure 5.7: Courbes de variation de  $\mu^l(k)$  et  $\check{r}^l(k)$  en fonction de  $\Delta\hat{y}(k)$ ,  $\check{r}_1^l = 0.2$  et  $\mu_3^l = \check{r}_3^l = 0.1$  . . . . . 86
- Figure 5.8: Courbe d'apprentissage avec l'algorithme RETRO\_KAL+ :  
 (a)  $\mu_3^2 = 0.1$ , (b)  $\mu_3^2 = 1$ , (c)  $\mu_3^2 = 5$ , (d)  $\mu_3^2 = 10$  . . . . . 89
- Figure 6.1: Données d'entrée synthétiques:  $T_{\min} = 0^\circ C$ ,  $T_{\max} = 10^\circ C$ ,  
 $p_{\min} = 0MPa$  et  $p_{\max} = 25MPa$  . . . . . 99
- Figure 6.2: Courbes d'apprentissage (a) et de généralisation (b) pour un nombre variable de neurones sur la couche cachée avec  $\mu_e = 0.1$ ,  $\check{r}_a = 0.2$  et  $\check{r}_e = 0.01$  . . . . . 102
- Figure 6.3: Courbes d'apprentissage (a) et de généralisation (b) pour un nombre variable d'exemples d'apprentissage avec  $\mu_e = 0.1$ ,  $\check{r}_a = 0.2$  et  $\check{r}_e = 0.01$  . . . . . 103
- Figure 6.4: Courbes d'apprentissage (a) et de généralisation (b) pour différentes valeurs de  $\mu_e$  avec  $\check{r}_a = 0.2$  et  $\check{r}_e = 0.01$  . . . . . 104
- Figure 6.5: Courbes d'apprentissage (a) et de généralisation (b) pour différentes valeurs de  $\check{r}_a$  avec  $\mu_e = 0.1$  et  $\check{r}_e = 0.01$  . . . . . 105
- Figure 6.6: Courbes d'apprentissage (a) et de généralisation (b) pour différentes valeurs de  $\check{r}_e$  avec  $\mu_e = 0.1$  et  $\check{r}_a = 0.2$  . . . . . 106
- Figure 6.7: Courbes d'apprentissage de données synthétiques à température variable avec  $N_T = 4$  et  $N_P = 15$ , correspondant à trois réalisations de bruit . . . . . 110
- Figure 6.8: Variation de l'erreur relative limite  $\bar{\delta}\hat{p}_n$  pour différents niveaux de bruit sur la température  $\bar{\Delta}\bar{T}^{etal}$ , données synthétiques avec  $N_T = 4$  et  $N_P = 15$  . . . . . 112

- Figure 6.9: Variation de l'erreur relative limite  $\overline{\delta p}_n$  pour différents niveaux de bruit sur la température  $\overline{\Delta T}^{etal}$ , données synthétiques avec  $N_T = 4$  et  $N_P = 20$  . . . . . 112
- Figure 6.10: Variation de l'erreur relative limite  $\overline{\delta p}_n$  pour différents niveaux de bruit sur la température  $\overline{\Delta T}^{etal}$ , données synthétiques avec  $N_T = 2$  et  $N_P = 40$  . . . . . 113
- Figure 6.11: Variation de l'erreur relative limite  $\overline{\delta p}_n$  obtenue avec la méthode basée sur les fonctions spline, pour différents niveaux de bruit sur la température  $\overline{\Delta T}^{etal}$ , données synthétiques avec  $N_T = 4$  et  $N_P = 20$  . . . . . 115
- Figure 6.12: Données d'apprentissage d'entrée (a) et de sortie (b),  $T^{etal} = 15^\circ C$ ,  $p_{\min} = 6 MPa$  et  $p_{\max} = 24 MPa$ , interpolations avec les fonctions spline (—), données expérimentales (•) . . . . . 117
- Figure 6.13: Variation de l'erreur relative limite  $\overline{\delta p}_n$  pour des données expérimentales avec  $N_T = 4$  et  $N_P = 15$  . . . . . 118

## LISTE DES TABLEAUX

	page
Tableau 4.1: Qualités des différentes méthodes d'apprentissage . . . . .	60
Tableau 4.2: Classification des réseaux de neurones selon leurs domaines d'application . . . . .	62
Tableau 5.1: Valeurs des paramètres du modèle, équation 5.3, des capteurs $S_1$ , $S_2$ et $S_G$ . . . . .	71
Tableau 5.2: Influence des paramètres $\mu_3^l$ et $\check{r}_3^l$ sur les performances du réseau de neurones en phase d'apprentissage . . . . .	88

# CHAPITRE 1

## INTRODUCTION

### 1.1 Problématique

Le traitement d'un signal mesuré dans un système de mesure se réalise en deux phases: conversion et reconstitution. La conversion dans un système de mesure électrique consiste en une série de transformations de signaux reçus d'un objet de mesure en signaux électriques, standards, préférablement numériques. L'opération d'identification de la fonction de conversion du système de mesure consiste en l'étalonnage du système. La reconstitution de signaux à mesurer constitue un problème fondamental dans la métrologie. Elle consiste à estimer un signal  $x$ , qui n'est pas mesurable directement, à partir des résultats du mesurage d'un autre signal  $\tilde{y}$ , qui est lié avec le premier de façon causative. Les méthodes de reconstitution sont généralement basées sur certaines suppositions concernant le modèle mathématique de la relation entre les signaux, l'information *a priori* sur le signal reconstruit ainsi que le bruit qui entache les résultats de la conversion  $\tilde{y}$ . Pour effectuer la reconstitution, le système de mesure doit être étalonné. L'étalonnage consiste à l'identification du modèle de la relation entre la grandeur mesurée et les résultats de la conversion et à l'estimation des paramètres de ce modèle. La procédure d'étalonnage doit tenir compte de l'existence de facteurs d'influence qui agissent sur la conversion.

Les méthodes analytiques ou numériques de reconstitution de signaux présentent le désagrément d'être fortement dépendantes du modèle mathématique de la relation entre les signaux et l'information *a priori* du système de mesure. Dans le cas de résolution de problèmes mathématiquement dits "mal posés", la supposition simplificatrice que le



système de mesure est linéaire n'est pas toujours justifiée. Ce qui fait que plusieurs méthodes de résolution ne peuvent s'appliquer à des systèmes fortement non-linéaires. Compte tenu des problèmes inhérents aux méthodes de reconstitution analytiques, nous envisageons une méthode non analytique, indépendante du modèle mathématique de la relation entre les signaux du système de mesure, basée sur les réseaux de neurones et leurs performances en reconnaissance de formes et classification. L'idée de base est la suivante: un réseau de neurones peut être considéré comme une *boîte noire* accessible par ses entrées et ses sorties. Les signaux d'entrée sont introduits dans le réseaux, ils sont traités, et finalement un résultat nous est fourni par le réseau au niveau de sa sortie. Dans l'utilisation des réseaux de neurones nous avons deux phases, la première est la phase d'apprentissage qui consiste à faire apprendre aux réseaux des données déjà connues et la seconde est la phase de reconnaissance qui consiste à présenter au réseau des données inconnues et d'obtenir à sa sortie les informations pertinentes.

La connaissance des données d'entrées et des résultats obtenus à la sortie de la conversion permet de construire le réseau de neurones lors de sa phase d'apprentissage. Par la suite, la phase de reconnaissance qui consiste à présenter à l'entrée du réseau les signaux de conversion permet de reconstituer le mesurande. Un des points forts des réseaux de neurones est la généralisation. Elle permet de donner un résultat de reconstitution assez juste pour des données floues appliquées à ses entrées. La qualité de la reconstitution obtenue lors de la phase de reconnaissance est fonction de la richesse de l'étalonnage effectué.

## 1.2 Objectifs

Nous proposons d'étudier les possibilités d'application des réseaux de neurones aux problèmes d'étalonnage et de reconstitution de mesurandes statiques dans un système de

mesure. Le mode de fonctionnement des réseaux de neurones est basé sur l'utilisation de modèles d'apprentissage, et ce contrairement aux autres méthodes d'identification du modèle mathématique de la relation entre les résultats numériques de la conversion et la grandeur mesurée. L'objectif de ce travail est donc de faire le lien entre deux domaines de recherche distincts: les systèmes de mesure et les réseaux de neurones.

La méthode devra être appliquée dans un cas d'étalonnage et de reconstitution de mesurandes pour un système de mesure de pression avec des capteurs à fibres optiques. Ce système de mesure comprend trois capteurs polarimétriques dont les tensions de sortie sont des fonctions périodiques de la pression et sont sensibles aux variations de température. L'influence de la température sera prise en compte comme un élément important par la méthode d'étalonnage et de reconstitution de pression proposée.

### **1.3 Structure du mémoire**

Le travail présenté se divise en cinq sections distinctes qui permettent de définir le problème à résoudre, présenter les fondements de la nouvelle méthode proposée et finalement vérifier la validité de cette dernière dans un cas précis d'application, soit un système de mesure de pression à fibres optiques. Dans le chapitre 2, les problèmes d'étalonnage statique d'un système de mesure et de reconstitution de mesurandes sont étudiés afin de définir les contraintes qui doivent être satisfaites dès le début de cette recherche. Le chapitre 3 est réservé à la définition de l'application envisagée pour valider la méthode d'étalonnage et de reconstitution, soit le système de mesure de pression à fibres optiques.

La problématique posée, les différents réseaux de neurones existants sont étudiés, dont les réseaux utilisant les neurones analogiques qui devraient satisfaire à la résolution du problème. Ce qui est fait suite à une recherche bibliographique sur les réseaux de neurones et qui apporte les références pertinentes pour l'étude, les points les plus pertinents pour la suite du travail sont présentés au chapitre 4.

Le choix du réseau posé, de type multicouche, différents algorithmes d'apprentissage sont testés dans le but de mettre au point un réseau de neurones fonctionnel pour l'application envisagée, c'est-à-dire un réseau ayant une bonne capacité de stockage, un apprentissage rapide et robuste et une très bonne capacité de généralisation. Les résultats, obtenus avec trois algorithmes d'apprentissage basés sur la méthode standard de la rétropropagation du gradient, sont présentés au chapitre 5.

Le chapitre 6 met en application la méthode d'étalonnage et de reconstitution de mesurandes dans le cas du système de mesure de pression. Les résultats devront confirmer de l'applicabilité des réseaux de neurones pour l'étalonnage et la reconstitution de mesurandes dans le cas du système de mesure étudié, dans le cas de données synthétiques et expérimentales. Une comparaison avec une autre méthode d'étalonnage et de reconstitution, basée sur les fonctions spline, est proposée afin d'évaluer la qualité de la méthode mise au point dans ce travail. L'implantation matérielle de la méthode de reconstitution est aussi réalisée pour estimer la qualité, les temps de traitement et l'espace mémoire requis pour une reconstitution de pression. Le travail se termine par une conclusion, chapitre 7, qui est une synthèse du travail réalisé et des résultats obtenus.

## CHAPITRE 2

### SYSTÈME DE MESURE

#### 2.1 Introduction

L'objectif principal de ce travail est d'étudier les possibilités d'appliquer les réseaux de neurones aux systèmes de mesure, il convient donc dans une première étape de définir quelles opérations liées aux systèmes de mesure sont à considérer. Les réseaux de neurones peuvent être vus dans une première approche comme des *boites noires* qui reçoivent des données à leurs entrées et fournissent une réponse à cette sollicitation à leurs sorties après traitement. Il paraît donc évident que la première partie du travail va consister à définir la tâche que l'on va assigner aux réseaux de neurones et donc les données que l'on va lui fournir et les résultats que l'on souhaite obtenir en sortie. Le but de ce chapitre est donc de donner les bases fondamentales du travail en définissant les opérations qui régissent les systèmes de mesure, ce sont elles qui seront assignées aux réseaux de neurones par la suite, et de présenter succinctement les méthodes les plus couramment utilisées pour effectuer ces opérations et pour identifier les principaux problèmes inhérents à l'utilisation de ces méthodes. Concernant ces méthodes, l'accent sera mis sur la méthode variationnelle de spline qui sera par la suite utilisée à des fins de comparaison. Pour clore ce chapitre, nous introduirons les réseaux de neurones comme méthode potentielle à la réalisation des tâches préalablement décrites.

#### 2.2 Définitions de base

##### 2.2.1 *La mesure*

De manière très générale, la mesure peut être définie comme suit [FIN75]:



Figure 2.1 Fonction mathématique de mesure.

*La mesure est un procédé opérationnel d'application des nombres aux éléments d'une certaine classe des aspects ou des caractéristiques de l'univers suivant des règles bien définies.*

Les règles sont conçues de façon à ce que le nombre appliqué à l'entité la décrive. Une relation entre les nombres appliqués aux différents éléments de la classe correspond à une relation empirique entre les éléments auxquels ils sont appliqués. Mathématiquement cela se traduit par la relation représentée à la figure 2.1, où

- Q une classe bien définie des entités extramathématiques ( $q_1 \dots q_n$ ),
- R un ensemble de relations empiriques existantes dans la classe ( $r_1 \dots r_n$ ),
- N un ensemble de nombres, en général un sous-ensemble de l'ensemble des nombres réels ( $n_1 \dots n_n$ ),
- P un ensemble de relations numériques définies dans l'ensemble des nombres ( $p_1 \dots p_n$ ),
- M une transformation du domaine Q vers N, c'est un homomorphisme d'un système relationnel empirique (Q,R) et d'un système numérique relationnel (N,P).

Par la transformation M, procédé opérationnel bien défini:

- $n_i \in N$  devient l'image de  $q_i \in Q$  et il est noté par  $n_i = M(q_i)$ ,
- $n_i$  est appelé la *mesure* de  $q_i$ ,
- $q_i$  est la *valeur mesurée*,
- Q est la classe des valeurs mesurées.

### 2.2.2 *Le système de mesure*

La définition précédente nous permet de concevoir toutes les mesures pouvant exister et de mieux comprendre la complexité qui souvent résulte de l'action de mesurer une grandeur physique. Une opération de mesure nécessite, généralement, que l'information qu'elle délivre soit transmise à distance du point où elle est saisie, protégée contre l'altération par des phénomènes parasites, amplifiée, avant d'être traitée et exploitée par différents moyens. Ces fonctions sont assurées par des dispositifs ou éléments qui constituent un système dit *système de mesure*. La modélisation des systèmes de mesure a pour but de répondre à la question fondamentale *de quelle façon la représentation d'une grandeur fournie par un instrument ressemble-t-elle à la grandeur elle-même?* [PAR86]. Pour répondre à cette question, les systèmes de mesure sont décomposés en une chaîne d'éléments, chacun décrit par un modèle physique adéquat et mis sous forme d'une caractéristique de transfert.

Le système de mesure peut être représenté par le schéma de la figure 2.2, celui-ci nous permet de définir les deux opérations fondamentales associées à tout système de mesure: la *conversion* et la *reconstitution*. Soit la variable  $x$  sur le schéma une caractéristique empirique d'un objet de mesure, dont la détermination est le but de la mesure, soit  $\hat{x}$  le résultat obtenu en sortie de l'appareil de mesure. La transformation d'information de mesure  $x \rightarrow \hat{x}$  peut être décomposée en deux parties, la conversion et la reconstitution:

- la *conversion* consiste à transmettre l'information de mesure dans le domaine de phénomènes physiques qui sont faciles à interpréter ( $x \rightarrow \tilde{y}$ ),
- la *reconstitution* consiste à interpréter le résultat de conversion  $\tilde{y}$  ( $\tilde{y} \rightarrow \hat{x}$ ).

La *conversion* dans un système de mesure électrique consiste en un série de transformations de signaux reçus d'un objet de mesure, en signaux électriques, standards,

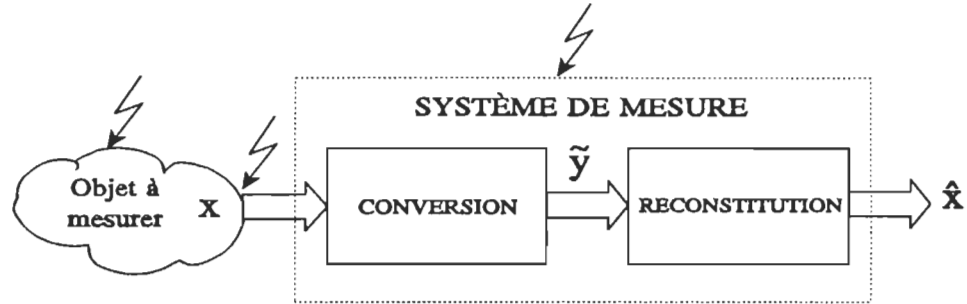


Figure 2.2 Schéma général d'un système de mesure.

préférentiellement numériques. Ces derniers sont remis au traitement afin d'obtenir des estimés des grandeurs à mesurer, cette dernière opération est appelée *reconstitution de mesurandes*.

Les performances d'un système de mesure électrique sont liées aux caractéristiques métrologiques des éléments qui le constituent, ces caractéristiques sont nombreuses et la liste suivante a pour but de présenter les principales [PAR86], [ASC87]:

- *sensibilité*: elle est définie comme le quotient de l'accroissement de la réponse par l'accroissement correspondant du signal d'entrée,
- *mobilité*: c'est l'aptitude d'un instrument de mesure à réagir aux petites variations du signal d'entrée,
- *finesse*: c'est l'aptitude d'un instrument de mesure à ne pas modifier la valeur de la grandeur mesurée,
- *justesse*: c'est l'aptitude d'un instrument de mesure à donner des indications, qui en moyenne, correspondent à la valeur vraie ou à la valeur conventionnellement vraie de la grandeur mesurée,
- *fidélité*: c'est l'aptitude d'un instrument de mesure à donner la même indication pour une même valeur de la grandeur mesurée,
- *rapidité*: un système est dit rapide s'il est capable de suivre l'évolution de la grandeur mesurée,

- *erreurs de mesure*: elles correspondent à l'écart entre la valeur mesurée et la valeur vraie, elles sont dues en particulier aux imperfections de la chaîne de mesure qui dégradent l'information du signal au cours de son traitement.

## 2.3 Étalonnage statique d'un système de mesure

### 2.3.1 Définition

L'étalonnage est l'opération qui permet d'identifier la relation empirique reliant les résultats fournis par un capteur et le mesurande. Dans le cas d'un étalonnage statique, le temps n'a pas d'action prépondérante sur le fonctionnement du système, par opposition à un étalonnage dynamique. En nous référant à la figure 2.2, résoudre un problème d'étalonnage statique revient à construire un modèle mathématique du bloc de conversion du système de mesure.

En situation pratique [PAR86], l'*étalonnage statique* est le relevé des performances d'un système de mesure dans les conditions spécifiées suivantes: toutes les grandeurs d'entrée, désirées, interférentes, modifiantes, sont maintenues constantes à l'exception de l'une d'entre elles. On fait varier cette dernière par paliers et la grandeur de sortie est relevée lorsque tout effet transitoire a disparu. L'énoncé d'une caractéristique de transfert statique n'a donc de valeur que lorsqu'il est accompagné des conditions spécifiées de l'étalonnage statique correspondant. En particulier, les valeurs des grandeurs maintenues constantes doivent être données.

### 2.3.2 Méthodes d'étalonnage statique

Considérons la figure 2.3 représentant le bloc conversion, partie à identifier, d'un système de mesure, l'étalonnage statique est le processus par lequel on va approximer la fonction reliant les  $y_i$  aux  $x_i$ . À partir d'un modèle simple nous allons présenter





Figure 2.3 Système à identifier où  $n$  est le nombre d'observations.

quelques méthodes d'étalonnage statique, deux classes distinctes sont considérées, les méthodes dites *paramétrées*, et les méthodes dites *non paramétrées*.

En considérant le système proposé à la figure 2.3, nous observons le comportement de la sortie  $y$  pour une entrée  $x = [x_1, x_2, \dots, x_n]^T$ , soit à chaque entrée  $x_i$  correspond une sortie  $y_i$  sur une plage de  $n$  observations. Le modèle d'étalonnage est le suivant:

$$y_i = \mu(x_i) + \varepsilon_i \quad \text{pour } i = 1, \dots, n \quad (2.1)$$

où  $\varepsilon_i$  est une erreur de moyenne zéro et de variance  $\sigma^2$ , la fonction  $\mu(\cdot)$  est appelée *fonction de regression* [EUB88].

Un modèle d'étalonnage est dit *paramétré* si la forme de la fonction de régression  $\mu(\cdot)$  est connue et qu'un nombre fini de paramètre est à déterminer. La dépendance du modèle en fonction de ces paramètres pouvant être linéaire:

$$\mu(t) = \sum_{i=1}^P \beta_i x_i(t) \quad (2.2)$$

où les  $x_i(t)$  sont des fonctions connues, et/ou non-linéaire:

$$\mu(t) = \beta_1 + \beta_2 e^{-\beta_3 t^{\beta_4}} \quad (2.3)$$

$$\mu(t) = \beta_1 \cos(\beta_2 t) + \beta_3 \sin(\beta_4 t) \quad (2.4)$$

où les  $\beta_i$  sont les paramètres à déterminer.

Les modèles paramétrés sont des fonctions qui permettent d'estimer les fonctions  $\mu(\cdot)$  [EUB88], ces fonctions ne sont pas uniques et les modèles sont donc très flexibles et s'ajustent très bien aux données. Cependant les méthodes paramétrées demandent un grand niveau de spécifications pour être estimées, ce niveau est diminué dans le cas de modèles d'étalonnage non paramétrés.

Un modèle d'étalonnage est dit *non paramétré* si la fonction de régression  $\mu(\cdot)$  est représenté par un élément d'un espace de fonctions connues. C'est-à-dire que pour construire un modèle de régression non paramétré, un espace de fonctions auquel la fonction  $\mu(\cdot)$  est supposée appartenir est choisi, ce choix est en général motivé par les propriétés de lissage que la fonction de régression est supposée avoir. Les données permettent alors de déterminer l'élément de l'espace de fonctions le plus représentatif de la fonction de régression  $\mu(\cdot)$ .

## 2.4 Reconstitution de mesurandes

### 2.4.1 Définition

Le problème de la reconstitution se pose comme un problème de restauration de l'entrée  $x$  d'un système linéaire et invariant. La mesurande  $x$  et le signal électrique  $y$  qui est le résultat de conversion sont, dans le cas d'appareils linéaires, liés par une équation intégrale de première espèce [DEM86], [MAS91]:

$$y(t) = \int_D h(t,\lambda)x(\lambda) d\lambda \quad (2.5)$$

Le noyau de cette intégrale,  $h(t,\lambda)$ , représente la fonction de transfert de l'appareil ou la réponse impulsionnelle. Dans les cas où la distorsion introduite par le système de mesure est invariante par translation et monodimensionnelle et où la transmission est causale, l'équation (2.5) s'écrit:

$$y(t) = x(t)*h(t) = \int h(t-\lambda)x(\lambda) d\lambda \quad (2.6)$$

ce qui correspond à une équation de convolution.

La reconstitution n'est en fait qu'un problème inverse particulier, et les difficultés rencontrées sont liées à la résolution d'une équation intégrale de première espèce, compliquée par la présence du bruit de mesure, ceci est représenté à la figure 2.4. En effet, le signal accessible  $\tilde{y}$  est entaché de bruit venant de perturbations inévitables. Ce qui fait que ce problème très simple et fort répandu offre le désagrément d'être mal posé. Ceci ne signifie pas qu'il soit mal formulé, mais simplement que sa résolution est très sensible aux inévitables erreurs dues à la modélisation. Nous obtenons l'équation numérique suivante:

$$\tilde{y}_k = \sum_{i=0}^k h_{k-i}x_i + v_k \quad (2.7)$$

où  $v_k$  est un bruit additionnel causant, même à de faibles valeurs, de grandes erreurs de reconstitution. Ainsi, une information *a priori* apportée sur la solution désirée fournissant une solution acceptable physiquement est appelée régularisation d'un problème mal posé.

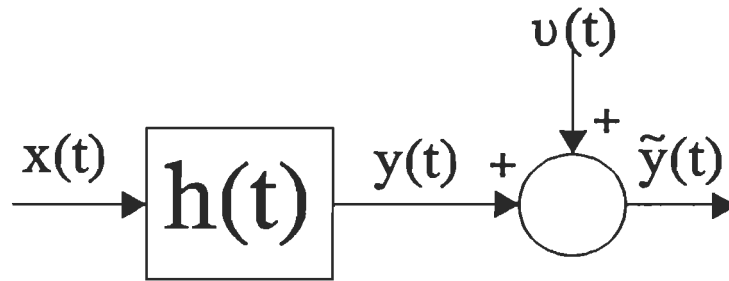


Figure 2.4 Représentation de la partie conversion du système de mesure.

La résolution de l'équation (2.7) devient un problème de *reconstitution* dans le cas où l'on cherche à reconstruire le signal d'entrée  $x$  et un problème d'*identification* ou d'*étalonnage* dans le cas où l'on cherche à identifier la réponse impulsionnelle du système  $h$ . D'une façon générale le modèle physique est une convolution et le problème à résoudre est une variété de problèmes inverses appelé pour cela *déconvolution*.

#### 2.4.2 Classification des méthodes de reconstitution

Les méthodes de reconstitution sont généralement basées sur certaines suppositions concernant le modèle mathématique de la relation entre les signaux et l'information *a priori* accessible et le bruit qui entache les résultats de conversion  $\tilde{y}$ . Il existe une diversité de méthodes pour résoudre le problème de la reconstitution ce qui démontre bien la difficulté présente [MOR89]. En fait, il n'existe pas de solution universelle. Le fait de ne pas avoir de solution unique pour résoudre l'équation (2.7) nous oblige à faire un choix parmi les diverses possibilités de résolution et pour chaque problème de reconstitution, il faut choisir ou développer une méthode adaptée à l'idée que l'on se fait *a priori* du signal à restaurer. La liste suivante, loin d'être exhaustive, a pour but de donner une idée générale sur quelques méthodes utilisées dans la reconstitution de mesurandes [MOR89]:

- *méthodes directes*, elles consistent à numériser le modèle de la relation liant l'entrée  $x$  et la sortie  $y$ . Le problème revient à résoudre une équation algébrique intégrale ou différentielle.

- *méthodes itératives*, elles consistent à reconstituer le mesurande par approximations successives à partir du signal de sortie  $\tilde{y}$ , cela revient à étendre l'opération de déconvolution en une série de déconvolutions. Nous pouvons citer la méthode de Jansson et la méthode de régularisation de Tikhonov.
- *méthodes stochastiques*, telle que la méthode de Kalman, ces méthodes stochastiques cherchent une estimation optimale du signal à reconstituer en déterminant le minimum de variance. La méthode de Kalman minimise la variance à chaque pas d'échantillonnage par une procédure récursive.
- *méthodes de transformation*, elles consistent à travailler dans le domaine des fréquences et sont basées sur une représentation algébrique de la convolution en transformées de Fourier (  $Y = HX$  ). La méthode de Wiener, qui est un filtrage inverse, fait partie de cette classe.
- *méthodes variationnelles*, telle que la méthode de spline qui consiste à déterminer des approximations polynomiales successives de la solution.
- *méthodes paramétriques*, elles consistent à trouver les paramètres des signaux à reconstituer.

Dans le cadre de notre travail, nous allons nous intéresser à l'une de ces méthodes, une méthode d'approximation basée sur les fonctions spline, pour pouvoir comparer les résultats d'étalonnage et de reconstitution de mesurandes obtenus avec notre méthode basée sur les réseaux de neurones (voir chapitres 6, 7 et 8). Cependant le but du travail n'est pas l'étude de ces méthodes et il est donc souhaitable de se référer à d'autres travaux réalisés précédemment sur le sujet [BEN91], [LEM95]. Nous dirons simplement que l'utilisation des fonctions d'approximation telles que les fonctions spline consiste à établir une approximation d'une fonction inconnue en se basant sur des séquences de données de mesure du système à l'étude. Ces données sont sujettes à des erreurs et parfois ne donnent

pas une approximation unique [SLI91]. Une fonction dite *spline* est une fonction formée de morceaux de polynômes qui se raccordent ainsi que certaines de leurs dérivées aux points de jonction [SCH81].

Les méthodes numériques de reconstitution de signaux sont dépendantes du modèle mathématique de la relation entre les signaux et l'information liée au système de mesure, tels que la fonction de transfert et des bruits qui le perturbent. La résolution est souvent sensible aux perturbations qui agissent sur le système de mesure.

Le problème de reconstitution, posé d'une façon abstraite, est bien conforme aux nombreuses situations pratiques dans les différents domaines de la technique de mesure et de commande. Des méthodes numériques de reconstitution sont profitablement utilisées pour:

- l'interprétation des données sismiques,
- l'amélioration de la résolution de l'analyse spectrométrique et chromatographique,
- la correction dynamique de capteurs,
- la mesure de la thermocinétique de réactions chimiques,
- l'accélération de mesurage quasi-statique,
- l'égalisation de canaux de télécommunication,
- le diagnostic médical basé sur les techniques de tomographie, etc.

Elles sont aussi utilisables dans le domaine de la robotique comme des moyens de d'étalonnage cinétique et d'estimation de paramètres dynamiques de manipulateurs. Les deux principales raisons de l'intérêt actuel pour l'étude des méthodes de reconstitution de signaux sont:

- L'amélioration de la précision des résultats fournis par un appareil sans pour autant atteindre les limites de résolution et ainsi travailler avec un appareil à moindre coût.
- La technologie actuelle nous permet la réalisation de processeur en traitement du signal pour ainsi atteindre des vitesses de reconstitution raisonnables.

### 2.4.3 Approche proposée

Compte tenu des problèmes inhérents aux méthodes de reconstitution analytiques, nous envisageons une méthode non analytique, indépendante du modèle mathématique du système de mesure, basée sur les réseaux de neurones et leurs performances en reconnaissance de formes et classification. L'idée de base est la suivante: un réseau de neurones peut être considéré comme une *boîte noire* accessible par ses entrées et ses sorties. Les signaux d'entrée sont introduits dans le réseaux, ils sont traités, et finalement un résultat nous est fourni par le réseau au niveau de sa sortie (voir chapitre 4). Avec les réseaux de neurones, nous disposons donc d'un système de traitement de signaux, leur application pour l'étalonnage d'un système de mesure et pour la reconstitution de mesurandes est schématisée à la figure 2.5, la méthode développée dans ce travail repose sur ce principe de base.

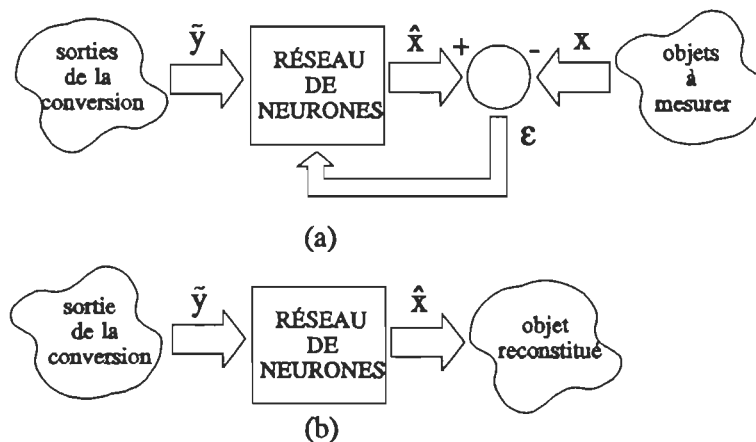


Figure 2.5 Application des réseaux de neurones à l'étalonnage d'un système de mesure (a) et à la reconstitution de mesurandes (b).

## 2.5 Conclusion

Dans ce chapitre nous avons défini en quoi consiste la tâche à réaliser, c'est-à-dire l'étalonnage d'un système de mesure et la reconstitution de mesurandes. Les notions de base ont été introduites sous formes de définitions, soient la *mesure*, le *système de mesure*, la *conversion*, l'*étalonnage* et la *reconstitution de mesurandes*. Les méthodes d'étalonnage et de reconstitution de signaux analytiques donnent de très bons résultats dans le cas où l'on dispose suffisamment d'informations sur le modèle mathématique du système de mesure et sur les bruits de perturbation. Cependant il est parfois impossible d'avoir ces informations et donc ces méthodes ne sont pas toujours applicables pour résoudre n'importe quel problème de reconstitution, c'est le cas des systèmes de mesure fortement non-linéaires ou si le bruit de perturbation est multiplicatif. Compte tenu des problèmes inhérents aux méthodes de reconstitution analytiques, nous envisageons une méthode non analytique, basée sur les réseaux de neurones et leurs performances de capacité d'apprentissage et de généralisation.



## **CHAPITRE 3**

### **SYSTÈME DE MESURE DE PRESSION À FIBRES OPTIQUES**

#### **3.1 Introduction**

Afin de démontrer l'applicabilité des réseaux de neurones aux systèmes de mesure comme méthode d'étalonnage statique et de reconstitution de mesurandes, nous avons utilisé un système de mesure précis, un *système de mesure de pression à fibres optiques*. Dans ce chapitre nous allons introduire les bases du fonctionnement de ce système de mesure de la manière suivante. Dans une première partie nous allons présenter le principe de fonctionnement et la caractéristique d'entrée-sortie d'un capteur de pression à fibre optique. La deuxième partie décrit le système de mesure de pression dans son ensemble, il est composé principalement de trois capteurs à fibre optique, d'un convertisseur analogique/numérique et d'un convertisseur numérique/numérique. Nous souleverons aussi un problème important lié au fonctionnement des capteurs de pression, ils présentent le désagrément d'être très sensibles aux variations de température et cet effet doit donc être pris en compte par la méthode d'étalonnage et de reconstitution de mesurande.

#### **3.2 Capteur de pression à fibre optique**

##### **3.2.1 Principe de fonctionnement**

Le capteur à fibre optique considéré, capable de mesurer des pressions allant jusqu'à 100 MPa, peut être vu comme une fibre interférométrique de type Mach-Zender, possédant deux modes de polarisation différents et dont les écarts de phase produisent différentes réponses à une pression extérieure [BOC89]. Les fibres optiques hautement biréfringentes sont capables d'éliminer l'influence de paramètres externes indésirables dans des

environnements instables et peuvent simultanément maintenir un état de polarisation linéaire quand la lumière est émise dans un plan parallèle à l'un des deux axes principaux de la fibre. Ces axes possèdent un indice de réfraction légèrement différent et par conséquent les deux plans perpendiculaires de polarisation ont différentes constantes de propagation. Pour une direction de polarisation différente des deux axes de polarisation, la fibre *HB* est caractérisée par un indice de biréfringence  $B$  et une longueur de battement  $L_B$  définis par:

$$B = \frac{\lambda}{L_B}, \quad L_B = \frac{2\pi}{\beta_x - \beta_y} \quad (3.1)$$

où  $\lambda$  est la longueur d'onde de la lumière,  $\beta_x$  et  $\beta_y$  sont les constantes de propagation pour les deux modes de polarisation linéaires et orthogonaux de la fibre.

L'état de polarisation tourne suivant la direction de propagation de la lumière. La longueur de battement  $L_B$  est définie par deux points d'égale phase sur la fibre. Si une pression hydrostatique est appliquée sur la fibre, la biréfringence de celle-ci augmente ou diminue et si la lumière suit un plan autre que l'un des deux plans de propagation de la fibre, l'intensité de la lumière devient sensible à la pression [BOC89]. L'équation de modulation de l'intensité de la lumière sous l'action d'une pression hydrostatique à la sortie du capteur est:

$$I = I_0 \cos^2 \left[ \frac{\pi L}{\lambda} (B + A \cdot C \cdot p) \right] \quad (3.2)$$

- où
- $I_0$  amplitude de l'intensité de la lumière propagée,
  - $L$  la longueur de la fibre,
  - $B$  l'indice de biréfringence (équation (3.1)),
  - $C = n^3(p_{11} - p_{12})(1 + \nu) / (2E)$ ,

- $n$  le coefficient de réfraction,
- $P_{11}, P_{12}$  les coefficients de contrainte optique,
- $\nu$  la constante de Poisson,
- $E$  le module de Young,
- $A$  une constante dépendant de la géométrie et du type de fibre *HB*,
- $p$  la pression appliquée.

L'intensité à la sortie du capteur observée avec un angle de  $45^\circ$  par rapport aux axes principaux de la fibre est une fonction périodique de la pression avec une période  $T_p$  (niveau de pression requis pour induire un déphasage de  $2\pi$  à la lumière polarisée à la sortie du capteur) calculée à partir de l'équation (3.2):

$$T_p = \frac{\lambda}{L \cdot A \cdot C} \quad (3.3)$$

on peut alors écrire l'équation (3.2) sous la forme:

$$I = I_0 \cos^2 \left[ \frac{\pi p}{T_p} + \delta \right] \quad (3.4)$$

où  $\delta = \pi p / L_B$  est la phase initiale.

Le capteur de pression peut être assemblé selon les deux configurations de base présentées à la figure 3.1, en *transmission* ou en *réflexion*, chacune ayant des avantages et des limites. Dans la configuration en transmission (figure 3.1(a)), le fonctionnement du capteur est le suivant: le laser envoie une lumière polarisée parallèlement à l'un des plans de polarisation de la fibre *HB* d'entrée afin de minimiser l'influence des paramètres extérieurs; une deuxième fibre, sensible et de longueur  $L$  est raccordée à la fibre d'entrée

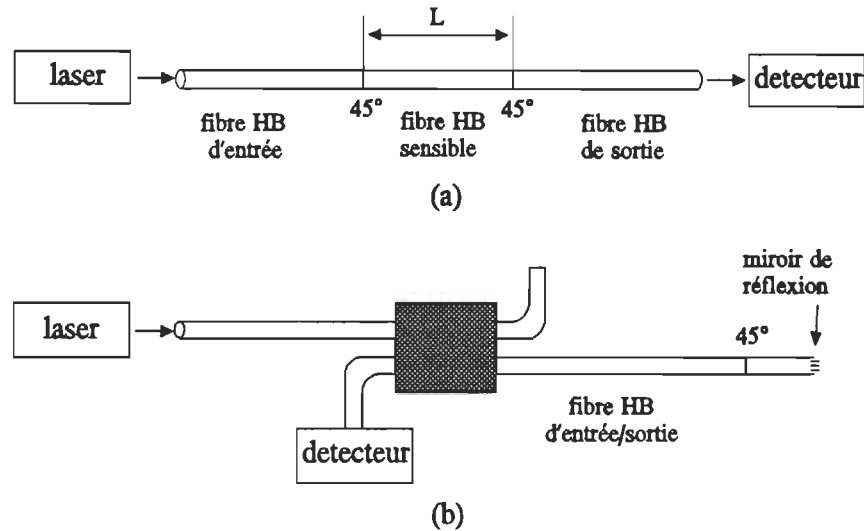


Figure 3.1 Configuration du capteur de pression à fibre optique en transmission (a) et en réflexion (b).

en formant un angle de  $45^\circ$  avec les axes de polarisation pour permettre à la lumière propagée d'exciter les deux modes de polarisation dans la partie sensible et ainsi d'avoir une modulation de l'intensité de la lumière en fonction de la pression exercée sur la fibre; pour récupérer le signal lumineux modulé par la pression, une troisième fibre *HB* de sortie est raccordée à l'élément sensible en formant un autre angle de  $45^\circ$ , l'immunité contre les paramètres d'influence extérieurs est alors préservée et il suffit de recueillir en sortie le signal par un détecteur. Le principal avantage de cette configuration est le niveau très élevé de lumière délivré en sortie pour un traitement ultérieur, le principal inconvénient est lié à la difficulté de fixer les angles de raccordement avec précision.

La deuxième configuration en réflexion (figure 3.1(b)) utilise une seule fibre *HB* d'entrée/sortie, raccordée à une extrémité à la fibre *HB* sensible avec un angle de  $45^\circ$ , un miroir de réflexion étant fixé à l'autre extrémité de cette dernière. Un coupleur bidirectionnel préservant la polarisation est ajouté pour séparer les signaux d'entrée et de sortie qui se propagent selon les deux modes de polarisation perpendiculaires de la fibre

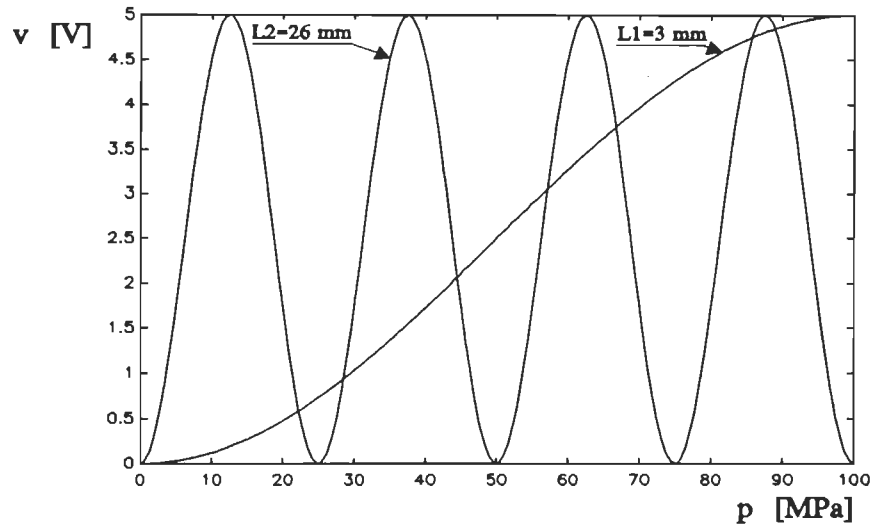


Figure 3.2 Caractéristique d'entrée/sortie du capteur pour deux longueurs de fibre.

*HB.* Le principal avantage de cette configuration est l'utilisation d'une seule fibre d'entrée/sortie et la possibilité relativement simple de multiplexage/demultiplexage [BOC89], le principal inconvénient est une baisse significative de l'intensité du signal lumineux disponible pour le traitement ultérieur, un rapport signal-bruit plus faible et une réduction du nombre de capteurs pouvant être multiplexés.

### 3.2.2 Caractéristique d'entrée/sortie du capteur

Nous avons vu dans la section précédente que la caractéristique d'entrée/sortie du capteur est une fonction périodique de la pression selon l'équation (3.2), comme le montre la figure 3.2 la variation du signal de sortie du capteur est une fonction quasi-sinusoidale de la pression dont la période est fonction de la longueur  $L$  de la fibre ( $v$  est la tension de sortie du capteur et  $p$  est la pression d'entrée). Il est très important de noter qu'une telle caractéristique d'entrée/sortie est obtenue à température fixe car le capteur est fortement influencé par celle-ci. La dépendance de la tension de sortie avec d'une part la pression et d'autre part la température est donnée par l'équation suivante:

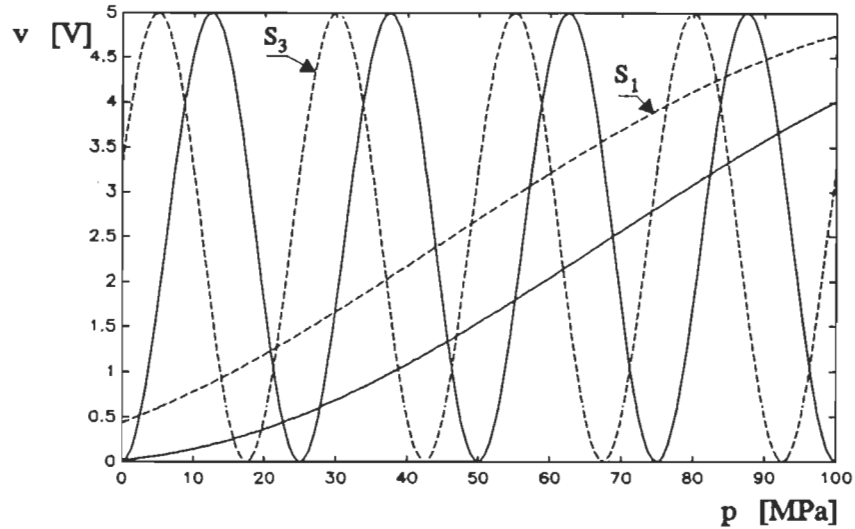


Figure 3.3 Influence de la température sur la caractéristique d'entrée/sortie du capteur,  $T_1$  en pointillé et  $T_2$  en continu avec  $T_1 < T_2$ .

$$v(p, T) = v_0 \cos^2 \left[ \frac{\pi L}{\lambda} (B - C_p \cdot \gamma \cdot \Delta T + |C| \cdot A \cdot \Delta p) \right] \quad (3.5)$$

- où -  $v_0$  l'amplitude de la tension de sortie,
- $C_p$  la constante relative de photoélasticité,
  - $\gamma$  une constante liée au type de fibre,
  - $\Delta T = T - T_0$  ,  
où  $T$  est la température ambiante et  $T_0$  est la température de référence,
  - $\Delta p = p_x - p_0$  ,  
où  $p_x$  est la pression mesurée et  $p_0$  est la pression atmosphérique.

L'effet de la température sur la caractéristique du capteur de pression est présenté à la figure 3.3. Le décalage induit par la température a été estimé expérimentalement à  $2 \text{ MPa}/^\circ\text{C}$  [BAR90].

### 3.3 Système de mesure de pression

#### 3.3.1 Principe de mesure

Avec le capteur de pression à fibre optique décrit dans la section précédente, on dispose d'un capteur très intéressant mais qui présente des problèmes qu'il faut résoudre avant de disposer d'un système de mesure fonctionnel. La fonction du système est de mesurer la pression, le capteur à fibre optique nous délivre une fonction du type  $v = f(p, T)$ , il faut donc inverser cette fonction pour obtenir la pression en fonction des tensions. Si nous observons la caractéristique du capteur à la figure 3.2, deux contraintes importantes apparaissent:

1. La caractéristique n'est pas monotone sur la plage des pressions envisagée, soit  $[0, \dots, 100] \text{ MPa}$ , il est donc impossible d'inverser la fonction  $v = f(p, T)$ .
2. La fonction  $v = f(p, T)$  possède des extrémums qui engendrent une grande erreur de calcul. Si la tension recueillie possède une erreur de mesure  $\Delta v$ , cette erreur de mesure engendre une erreur de calcul  $\Delta p$  sur la pression, d'autant plus élevée

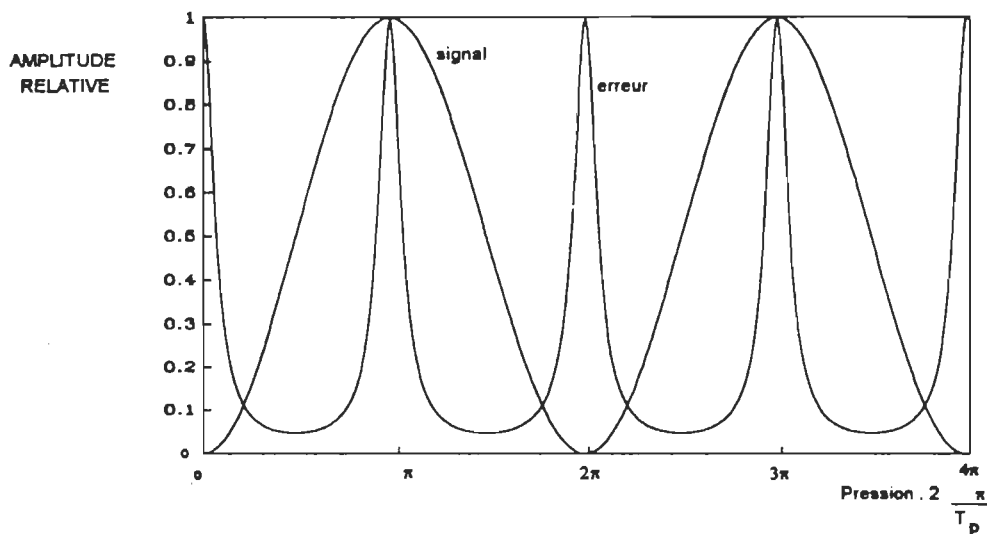


Figure 3.4 Représentation de l'erreur de mesure de pression [LEM95].

que la pente du signal est faible. La courbe d'erreur de calcul sur la pression est représentée à la figure 3.4, on constate qu'elle est maximum au niveau des extrémités du sinus.

En tenant compte de ces deux points, il apparait qu'un capteur unique ne peut pas être utilisable pour mesurer avec précision la pression. Le problème de la non-monotonie de la caractéristique du capteur peut être contourné théoriquement en jouant sur la période du signal qui est inversement proportionnelle à la longueur  $L$  de la fibre ou encore jouant sur la longueur d'onde de la lumière propagée dans la fibre. Dans ce cas la fonction est inversible, mais on perd de la précision et le problème des extrémums n'est pas résolu, de plus l'erreur de mesure de la pression n'est pas identique sur toute la plage de pression considérée. Ceci est inacceptable car l'erreur produite par un système de mesure doit être relativement constante sur la plage de mesure. La solution à ce problème se trouve dans l'utilisation de plusieurs capteurs déphasés les uns des autres et/ou de longueurs différentes.

Pour résoudre les problèmes de précision et d'extrémums, le système de mesure est composé de trois capteurs à fibre optique, leurs caractéristiques sont présentées à la figure 3.5. Le capteur  $S_G$ , appelé *guide*, permet d'obtenir une première approximation de la pression car sa caractéristique est monotone sur la plage de pression, matériellement il est constitué par une fibre sensible relativement courte, ce qui permet l'obtention d'un signal monotone. La pression qu'il fournit est sans ambiguïté mais peu précise. La précision est obtenue en utilisant un deuxième capteur  $S_1$  ayant une fibre sensible plus longue et fournissant ainsi une caractéristique non-monotone. Cependant l'erreur sur la pression aux extrémums de la caractéristique est plus élevée et la solution à ce problème consiste à enlever ces régions ou du moins à ne pas les utiliser lors de la détermination de la pression. Pour se faire, un troisième capteur de pression  $S_2$  ayant la même



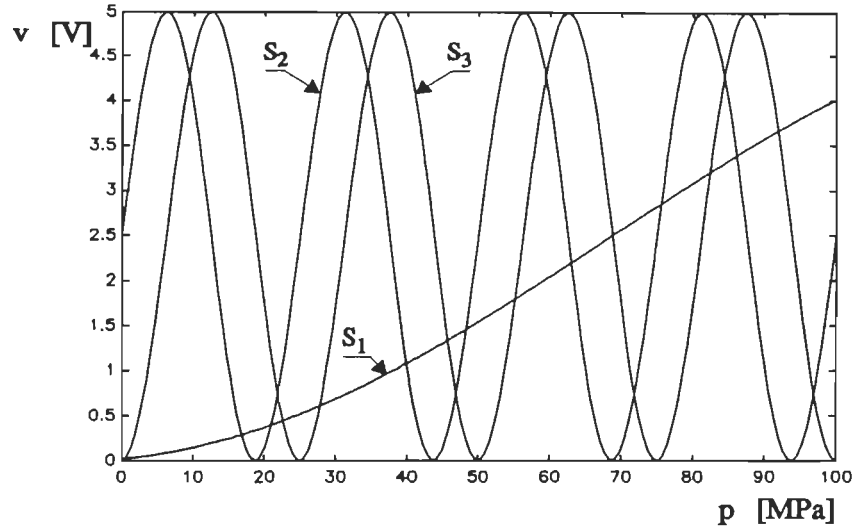


Figure 3.5 Caractéristiques des trois capteurs  $S_1$ ,  $S_2$  et  $S_3$  du système de mesure.

caractéristique que  $S_1$  mais déphasée de  $90^\circ$  est utilisé. Lors du calcul de la pression il suffit alors de choisir le capteur ayant la pente la plus élevée pour obtenir la meilleur précision sur la pression. Les régions sensibles des caractéristiques des capteurs  $S_1$  et  $S_2$ , ainsi que l'erreur relative de mesure sont présentées sur la figure 3.6.

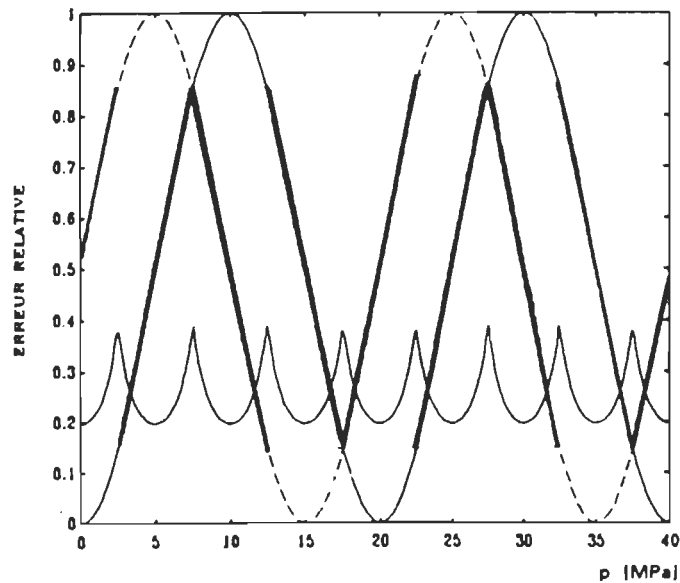


Figure 3.6 Régions de sensibilité des capteurs  $S_1$  et  $S_2$  en gras et erreur relative des deux capteurs déphasés de  $90^\circ$  [LEM95].

Un autre problème a été soulevé précédemment, l'influence de la température sur la caractéristique d'entrée/sortie du capteur (voir figure 3.3), il faudra donc tenir compte de la température pour étalonner le système de mesure et reconstituer les pressions.

### 3.3.2 Description du système de mesure

Le diagramme bloc du système de mesure de pression utilisant les trois capteurs de pression à fibre optique est proposé à la figure 3.7, il comprend les trois parties spécifiques suivantes [BAR90]:

1. *Conversion A/A*: cette partie convertit les signaux non électriques en signaux électriques et réalise le conditionnement des signaux, ce qui correspond à la conversion de la pression  $p$  en signaux lumineux par les capteurs et des signaux lumineux en tensions DC (variation comprise entre 0 et 5 volts).
2. *Conversion A/N*: cette partie convertit les signaux électriques en données numériques, ce qui correspond à la transformation des tensions DC en données numériques par un convertisseur A/N.
3. *Conversion N/N*: cette partie réalise le calcul du résultat de mesure à partir des données numériques et d'un modèle mathématique implanté dans sa mémoire.

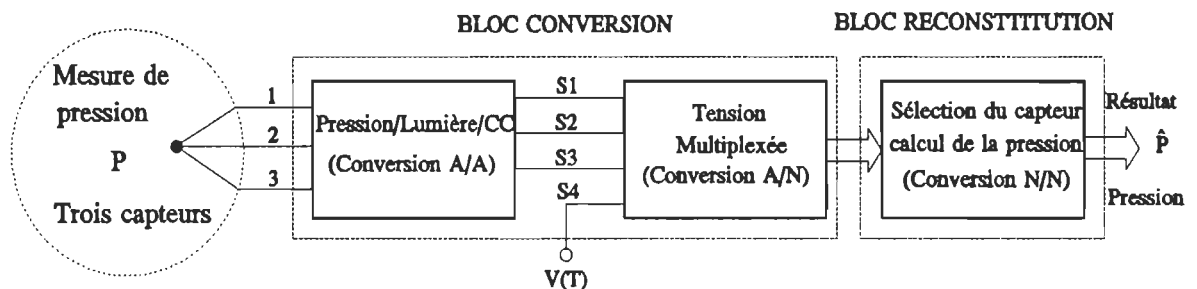


Figure 3.7 Diagramme bloc du système de mesure de pression.

Les parties 1 et 2 assurent l'opération de *conversion* du système de mesure, l'opération de *reconstitution de mesurande* est assurée par la partie 3 qui fournit la pression (voir figure 3.7).

Nous terminons ce chapitre sur le système de mesure de pression à fibre optique en présentant un algorithme général de mesure de la pression:

1. mesures successives des tensions de sortie  $v_1$ ,  $v_2$  et  $v_G$  des capteurs  $S_1$ ,  $S_2$  et  $S_G$  respectivement, mesure éventuelle d'une quatrième tension  $v_3$  comportant l'information sur la température ambiante lors de la mesure des tensions,
2. calcul de la première approximation de la pression en utilisant  $S_G$ , puis choix du capteur  $S_1$  ou  $S_2$  pour obtenir le maximum de précision.
3. calcul de la pression mesurée à partir des données numériques obtenues et du modèle inverse de la fonction  $v = f(p, T)$ , soit  $p = f(v, T)$ .

### 3.4 Conclusion

Dans ce chapitre nous avons présenté le système de mesure qui va être utilisé dans la suite du travail pour valider l'applicabilité des réseaux de neurones à l'étalonnage d'un système de mesure et à la reconstitution de mesurandes. Le système de mesure de pression à fibre optique peut être décomposé en trois blocs: les deux premiers servent à convertir la pression en trois (ou quatre) signaux numériques, c'est la partie conversion du système, traitables par le troisième bloc qui sert à reconstituer la pression d'entrée. Le premier bloc est composé de trois capteurs de pression sensibles à la pression qu'ils subissent et dont la caractéristique d'entrée/sortie est quasi-sinusoïdale. L'utilité de trois capteurs permet de profiter du maximum de sensibilité à la pression tout en résolvant les deux inconvénients engendrés par les capteurs eux-mêmes, soient la non-monotonie des caractéristiques et la

perte de précision en certaines zones des caractéristiques due à l'existence d'extrémums. Il reste toute fois un autre problème important qui est l'influence de la température sur la caractéristique d'entrée/sortie du capteur, il faudra donc tenir compte de la température pour étalonner le système de mesure et reconstituer les pressions.

Deux méthodes de reconstitution numériques ont été actuellement proposées, une méthode par approximation à l'aide d'une courbe sinusoïdale mise au carrée [BAR90] et une méthode d'interpolation polynomiale ajustée par segments appliquant la méthode des splines cubiques [MOR91]. Ces méthodes prennent en compte l'influence de la température sur les caractéristiques des capteurs en ajoutant un quatrième signal numérique au bloc de conversion  $A/N$  qui contient l'information sur la température. Dans les chapitres suivants, nous allons proposer une méthode différente d'étalonnage et de reconstitution de mesurandes en utilisant les réseaux de neurones, notre but est d'éviter le plus possible de dépendre de la mesure de la température pour mener à bien les deux opérations.

## **CHAPITRE 4**

### **LES RÉSEAUX DE NEURONES**

#### **4.1 Introduction**

La mise au point de systèmes intelligents capables de modéliser et de copier le fonctionnement du cerveau humain suscite un engouement général dans le monde scientifique depuis des années. Il n'est donc pas surprenant qu'une technologie basée sur des fondements biologiques tels que les réseaux de neurones retienne particulièrement l'attention. Cependant, avant d'appliquer les réseaux de neurones artificiels (ou formels) pour résoudre un problème complexe, il est indispensable de se familiariser avec les principales notions de base et ce qui est le but de ce chapitre. À la suite d'un rappel sur le système nerveux et les neurones biologiques, les différents types de neurones formels et les modes d'apprentissage sont présentés et ce par ordre chronologique de parution dans la littérature scientifique. Cette classification nous permet d'une part de mieux comprendre l'évolution considérable du domaine des réseaux de neurones formels ces dernières années, et d'autre part de présenter les exemples de réseaux de neurones les plus pertinents en présentant leurs structures, leurs algorithmes de fonctionnements, leurs qualités et de ce fait leurs domaines d'application, sans oublier leurs défauts et limites de performances. La conclusion de ce chapitre doit nous permettre de nous guider quant au choix du réseau de neurones à utiliser pour son application à l'étalonnage et à la reconstitution de mesurandes dans un système de mesure.

## 4.2 Notions et définitions de base

### 4.2.1 Fondements biologiques

Un bref rappel sur les neurones biologiques nous permettra de mieux comprendre l'origine des réseaux de neurones formels. Le système nerveux central est composé d'un ensemble de cellules nerveuses fortement interconnectées, leur nombre est estimé à une centaine de milliards. Les neurones sont des cellules particulières capables de recevoir, de traiter et de transmettre des messages sous formes d'impulsions électriques. Chaque neurone est constitué d'un corps cellulaire, d'une multitude de petites ramifications, appelées dendrites, et d'une ramification unique beaucoup plus longue appelée axone, ce qui est représenté à la figure 4.1(a). La connexion interneuronale se fait entre l'axone et les dendrites, par l'intermédiaire d'une synapse, c'est le nombre de neurones et les connexions interneuronales qui définissent le réseau de neurones.

Le mode de fonctionnement d'un neurone biologique est le suivant: il reçoit des impulsions électriques provenant des neurones auxquels il est connecté; à partir de tous ces signaux, le neurone peut ou non émettre un message, dans l'affirmative, il le fait par

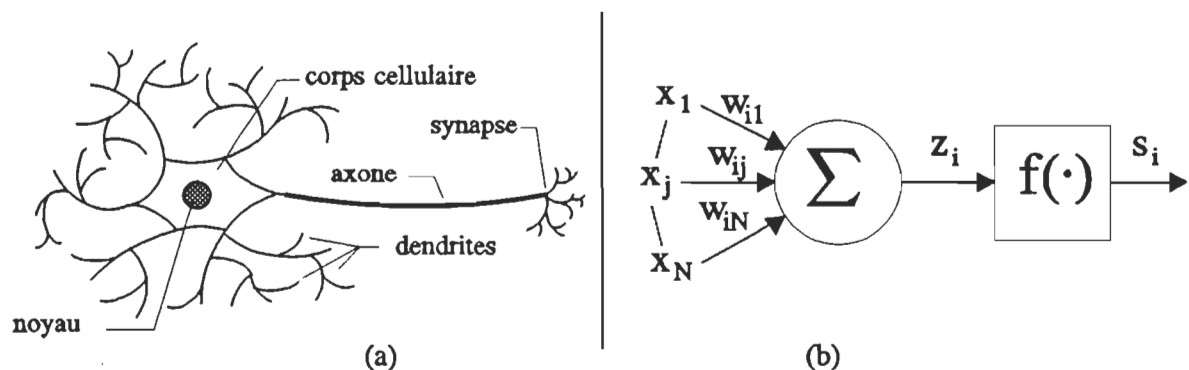


Figure 4.1 Neurone biologique (a) et sa formalisation mathématique (b)

l'intermédiaire de son axone et la synapse qui est le lieu de passage des impulsions. Une synapse peut donc être excitatrice ou inhibitrice. Le niveau d'excitation ou d'inhibition correspond à une pondération des influx nerveux arrivant des dendrites. On considère que le corps cellulaire du neurone réalise la somme pondérée de tous les influx nerveux, un signal sera émis si cette somme est supérieure à un seuil.

#### 4.2.2 *Le neurone formel*

En se basant sur les fondements biologiques du système nerveux, il est possible d'envisager un réseau artificiel qui serait lui aussi composé de cellules de calcul, appelées neurones artificiels ou formels et de connexions entre ces cellules. Intéressons nous dans un premier temps aux cellules de traitement. Par analogie avec les neurones biologiques, ces cellules doivent être aptes à réaliser les trois opérations suivantes: *collecter*, *traiter* et *transmettre* des messages. Pour satisfaire ces exigences, le neurone formel doit posséder plusieurs entrées pour collecter les messages provenant des autres neurones formels, pour que ceux-ci soient pondérés le neurone formel doit posséder des poids de pondération au niveau de chacune de ses entrées; une partie de traitement modélisée par une fonction mathématique appelée fonction de transition; une sortie unique qui transmet le message résultant du traitement des signaux d'entrée aux autres neurones formels. Nous avons alors le parallèle suivant entre le neurones formel et le neurone biologique: les dendrites sont des entrées, les synapses des poids de connexion et l'axone la sortie unique. Ceci est montré à figure 4.1 où l'on retrouve un neurone biologique (figure 4.1(a)) et son homologue formel (figure 4.1(b)).

Mathématiquement on peut transcrire cette modélisation sous la forme de l'équation suivante, les connexions se font dans le sens des neurones d'indice  $j$  vers le neurone d'indice  $i$ :

$$s_i = f(z_i) = f\left(\sum_{j=1}^{N_j} W_{ij}x_j - \theta_i\right) \quad (4.1)$$

- où
- $x_j$  une entrée élémentaire du neurone  $i$ ,
  - $z_i$  l'entrée totale du neurone  $i$ ,
  - $s_i$  la sortie du neurone  $i$ ,
  - $\theta_i$  le seuil du neurone  $i$ ,
  - $w_{ij}$  les poids de connexion,
  - $f(\cdot)$  la fonction de décision non linéaire,
  - $N_j$  le nombre de neurones  $j$  connectés au neurone  $i$

Dans un modèle plus général de neurone formel, on définit la fonction de décision comme étant constituée par une fonction d'entrée totale et une fonction d'activation.

Le premier modèle de neurone formel a été présenté en 1943 par Mc Culloch et Pitts, c'est un automate à seuil dont l'unité de sortie est binaire, la fonction de décision utilisée est une fonction à seuil. Ce neurone ne peut donc qu'être actif (sortie à 1) ou inactif (sortie à 0). D'autres modèles ont été obtenus en modifiant la fonction de décision, en général ces fonctions sont plus complexes et elles permettent d'obtenir des neurones plus performants. Les modèles les plus courants sont au nombre de six [DAV89]:

- *automates booléens* : la sortie est une fonction booléenne des entrées elles-mêmes booléennes.
- *automates à seuil* : la sortie est binaire, les entrées binaires ou réelles, la fonction d'entrée est affine, la fonction d'activation est la fonction signe ou de Heavyside.
- *automates linéaires* : la sortie et les entrées sont réelles, la fonction d'entrée est linéaire, la fonction d'activation est l'identité.



- *automates à saturation* : la sortie et les entrées sont comprises dans un intervalle  $[u,v]$ , les fonctions d'entrée et d'activation sont linéaires.
- *automates continus* : la sortie et les entrées sont réelles, la fonction d'entrée est linéaire ou affine, la fonction d'activation est sigmoïde.
- *automates probabilistiques* : la sortie est binaire, les entrées quelconques, la fonction d'entrée est linéaire ou affine, la fonction d'activation est stochastique.

Les différents modèles sont donc caractérisés par leur fonction de décision (fonctions d'entrée et d'activation), les cinq fonctions d'activation les plus fréquemment employées par les réseaux de neurones sont les suivantes [SÀN92]:

1. *fonction linéaire* (figure 4.2(a))

$$f(x) = \alpha x \quad (4.2)$$

où  $\alpha$  est un nombre positif.

2. *fonction à seuils* (figure 4.2(b))

$$f(x) = \begin{cases} \beta & \text{si } x \geq \theta \\ -\delta & \text{si } x < \theta \end{cases} \quad (4.3)$$

où les valeurs de  $\beta$  et  $\delta$  définissent les niveaux de seuil, la variable  $\theta$  est le seuil prédéfini.

3. *fonction à rampe* (figure 4.2(c))

$$f(x) = \begin{cases} \gamma & \text{si } x \geq \gamma \\ x & \text{si } |x| < \gamma \\ -\gamma & \text{si } x \leq -\gamma \end{cases} \quad (4.4)$$

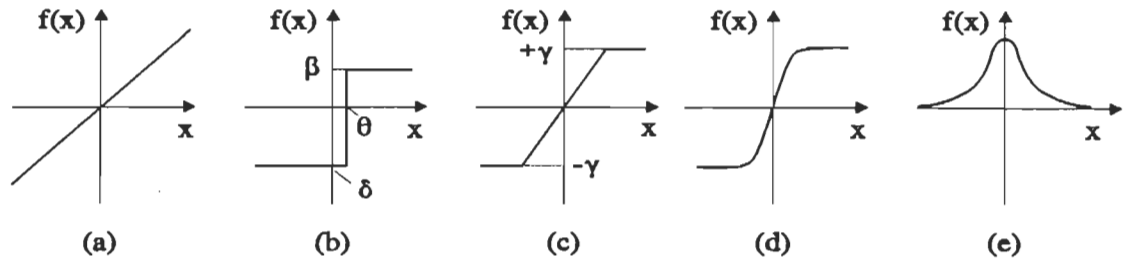


Figure 4.2 Fonction d'actication linéaire (a), à seuils (b), à rampe (c), sigmoïdale (d) et gaussoidale (e).

où  $\gamma$  est la valeur de saturation de la fonction et les points  $x = \gamma$  et  $x = -\gamma$  sont les points de discontinuité de la fonction.

4. *fonction sigmoïdale* (figure 4.2(d))

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \quad (4.5)$$

où  $\alpha > 0$ , en général  $\alpha = 1$ , ce qui maintient la sortie dans l'intervalle  $[0, 1]$ .

5. *fonction gaussoidale* (figure 4.2(e))

$$f(x) = e^{\left(\frac{-x^2}{v}\right)} \quad (4.6)$$

où  $x$  est la moyenne et  $v$  est la variance prédéfinie.

### 4.2.3 Topologies des réseaux de neurones

La topologie des réseaux de neurones est définie par les trois points suivants: l'arrangement dans l'espace des cellules de traitement (neurones formels); le réseau d'interconnexions entre ces neurones; les données traitées par le réseau [SÀN92]. Dans la section précédente nous avons défini les neurones, cellules élémentaires des réseaux de

neurones, il convient donc d'étudier maintenant les modes d'arrangements et d'interconnexions possibles entre eux. Tout comme le neurone formel est une modélisation mathématique du neurone biologique, le réseau formel d'interconnexions est basé sur le réseau de neurones biologiques formant le système nerveux central. S'il existe un parallèle entre une arborisation de neurones biologiques et un schéma d'interconnexions entre neurones formels, il est à noter que la densité en neurones est infiniment plus grande dans le cas biologique que dans le cas artificiel.

Nous avons donc introduit la notion de topologie des réseaux de neurones, définissons maintenant la terminologie associée à cette dernière:

1. *Couches*

Les réseaux de neurones sont organisés en couches de neurones formels qui correspondent à l'arrangement dans l'espace de ces derniers. D'un point de vue théorie des graphes, les réseaux à couches sont des graphes sans circuit.

2. *Connexions*

On distingue deux types de connexion dans les réseaux de neurones: les connexions *intracouches* relient les neurones d'une même couche, voire un neurone sur lui-même; les connexions *intercouches* relient les neurones de couches différentes. Les deux types sont compatibles et il est possible de voir des réseaux possédant les deux types de connexions.

3. *Boucles*

Une boucle est formée par un neurone connecté sur lui-même directement ou par l'intermédiaire de plusieurs neurones. D'un point de vue théorie des graphes, les réseaux bouclés constituent des graphes ayant au moins un circuit.

4. *Propagation et rétropropagation*

On distingue deux types de réseaux de neurones selon leur aptitude à *propager* une information dans un sens unique entre les neurones ou leur aptitude à *propager* une information dans un sens et *rétropropager* une information complémentaire dans l'autre sens. Dans ce dernier cas les neurones sont appelés à fonctionner dans les deux sens.

Les définitions précédentes nous permettent de classifier les réseaux de neurones selon leur topologie, cependant d'autres critères de classification sont intéressants pour parfaire la classification, tel que:

- le nombre de couches,
- le mode de connexion des réseaux de neurones à boucles: réseaux entièrement connectés et réseaux entièrement connectés par couches,
- le type de couches simples ou non: les réseaux à couches simples possèdent plusieurs couches et les connexions se font uniquement d'une couche  $i$  vers la couche  $i+1$ . S'ils sont non simples, les réseaux ont des connexions de n'importe quelle couche vers n'importe quelle autre.

L'arbre de la figure 4.3 permet de faire un synthèse des classifications envisagées sur les réseaux de neurones selon leur topologie. Le choix de classement a été fait d'une part sur l'existence ou non de boucles dans le réseau et d'autres part sur le nombre de couches, celles-ci pouvant être cachées ou non et les connexions entre couches simples ou non. Certaines classes de réseaux sont complétées par des exemples connus de réseaux de neurones, ceux-ci seront étudiés dans la suite du chapitre.

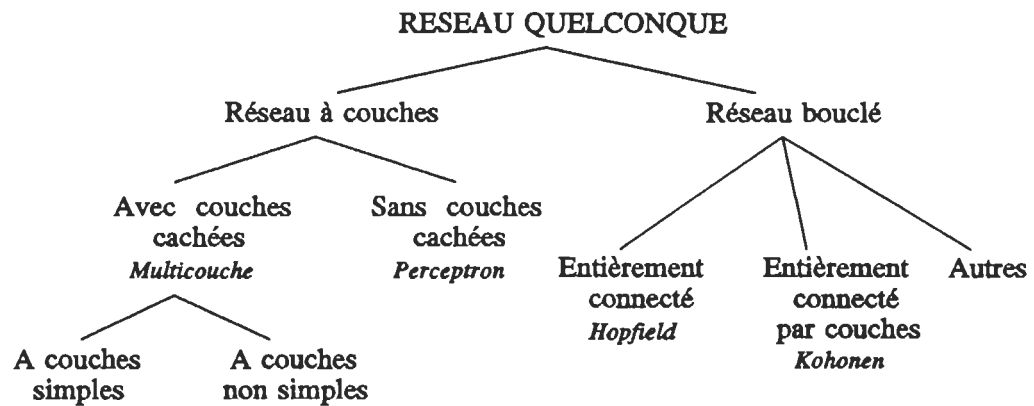


Figure 4.3 Classification des types de réseaux de neurones selon leur topologie.

#### 4.2.4 L'apprentissage des réseaux de neurones

La principale qualité des réseaux de neurones est leur capacité d'apprentissage, dans ce contexte, l'apprentissage est défini comme la capacité d'emmagasiner une information qui peut être rappelée par la suite. Les connaissances d'un réseau connexionniste sont mémorisées dans les poids de connexion, ces poids sont fixés lors d'un processus d'apprentissage. Il est important de comprendre que les poids de connexion sont totalement dépendants les uns des autres, en effet, pris séparément ils n'ont aucune signification. Ce sont les poids de connexion et l'ensemble du réseau qui définissent les capacités de ce dernier. L'apprentissage est fait en fonction des trois mécanismes suivants [RUM86a]:

1. *Développement de nouvelles connexions interneuronales.*
2. *Supression de connexions interneuronales inactives.*
3. *Modification des poids de connexion interneuronaux existants.*

À l'heure actuelle les points (1) et (2) ont été peu développés et peuvent être considérés comme des cas particuliers du cas (3) selon le changement apporté à la valeur

des connexions, pour le cas (1) une connexion initialement à "0" devient positive ou négative et dans le cas (2) une connexion initialement positive ou négative devient "0".

La phase d'apprentissage est capitale dans le fonctionnement d'un réseau de neurones puisqu'elle consiste à fixer les poids de connexion afin que le réseau de neurones devienne apte à réaliser la tâche qui lui est assignée. On distingue différents types d'apprentissage qui sont directement liés à la topologie du réseau de neurones:

- *supervisé*: des associations entrée/sortie sont présentées au réseau par un *professeur*, celui-ci doit déterminer, selon la nature du problème, la règle d'apprentissage la plus appropriée et le nombre de modèles à présenter.
- *non supervisé*: cet apprentissage ne concerne que les réseaux ne disposant que des informations d'entrée et qui les regroupent suivant un certain critère dans le but de créer des classes.
- *avec prototypes ou de nombreux exemples*: cet apprentissage concerne les réseaux qui nécessitent des algorithmes d'apprentissage itératifs. Les modifications des poids sont successives et dépendent du choix de présentation des modèles.
- *par empreinte ou par cœur*: dans ce cas la mémorisation est parfaite et définitive dès la première présentation de l'association à apprendre.
- *hors ligne*: le réseau est entraîné durant la phase d'apprentissage à l'aide d'un ensemble de données qui permettent de fixer les poids de connexion. Une fois la phase d'apprentissage achevée, le réseau est fonctionnel et il peut être utilisé sans modifications pour mener à bien la tâche qui lui a été assignée.
- *en ligne*: contrairement à l'apprentissage hors ligne, le réseau est en phase d'apprentissage continue, c'est-à-dire que toute action faite par le réseau est prise comme un élément d'apprentissage. Le réseau apprend et est apte à s'ajuster tout en réalisant la tâche qui lui est assignée.

En fonction de ces caractéristiques, de nombreuses règles d'apprentissage ont été proposées, chaque type de réseau possède une ou plusieurs règles d'apprentissage et compte tenu du nombre de types de réseaux de neurones et de la diversité des algorithmes d'apprentissage proposés dans la littérature, il est impensable de tous les étudier. Aussi nous allons nous restreindre aux réseaux et algorithmes les plus représentatifs des dernières décennies. Cependant la grande majorité des règles d'apprentissage sont basées sur la règle de Hebb proposée en 1949 [HEB49], cette règle est basée sur des études biologiques, selon le principe suivant : *Lorsqu'une connexion entre deux cellules est très forte, si la cellule émettrice est activée alors la cellule réceptrice l'est aussi. Pour lui permettre de jouer un rôle déterminant lors du mécanisme d'apprentissage, il faut donc augmenter le poids de cette connexion. En revanche, si la cellule émettrice est activée sans que la cellule réceptrice le soit, la connexion n'est pas prépondérante et on peut laisser un poids faible à cette connexion.*

La règle formelle de Hebb est alors : *Si deux neurones connectés sont activés en même temps, la connexion qui les relie doit être renforcée. Dans un cas contraire elle n'est pas modifiée.* Ce principe est illustré à la figure 4.4. La règle de Hebb peut être traduite mathématiquement de la façon suivante :

$$w_{ij}(t+1) = w_{ij}(t) + \mu A_i A_j \quad (4.7)$$

- où -  $w_{ij}$  le poids de connexion entre les neurones  $i$  et  $j$ ,
- $A_i$  l'activation du neurone  $i$ ,
  - $\mu$  un paramètre d'apprentissage.

D'autres règles d'apprentissage ont été proposées par la suite, on peut considérer qu'il existe deux grandes familles qui diffèrent par leur source d'inspiration :

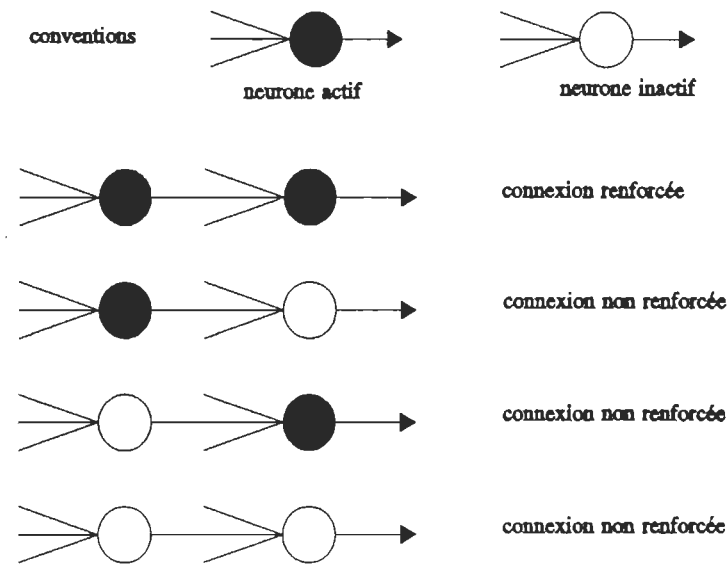


Figure 4.4 Règle de Hebb.

- *source biologique* : dans ce cas l'étude des modèles biologiques est à l'origine des modèles formels, par exemple les premières étapes des systèmes visuels animaux permettent de proposer des machines aptes à la reconnaissance de formes.
- *source mathématique* : des réseaux connexionnistes déjà existants sont considérés comme des fonctions de transfert des entrées, il faut alors calculer les paramètres de cette fonction. Les algorithmes qui en résultent sont basés sur des minimisations de fonctions de coûts ou des techniques d'algèbre linéaire.

### 4.3 Principaux types de réseaux de neurones

La formalisation d'un neurone biologique par Mc Culloch et Pitts dans les années 1940, puis le premier réseau de neurones formels fonctionnel de Rosenblatt appelé *perceptron* ont laissé entrevoir un potentiel formidable à exploiter des réseaux de neurones. Malgré les aléas accompagnant cette science nouvelle, les réseaux de neurones ont sans cesse intéressé de nombreux chercheurs qui ont proposé de nombreux types de réseaux, la plupart du temps basés sur le *perceptron* et la *règle de Hebb* pour améliorer et appliquer



les réseaux à de multiples domaines. Dans cette section dédiée aux différents types de réseaux de neurones et algorithmes d'apprentissage associés, nous allons dresser une liste chronologique et inévitablement très incomplète, des principaux réseaux qui ont le plus marqué et fait progresser la science des réseaux de neurones formels des années 1950 à aujourd'hui.

#### **4.3.1 *Le perceptron: 1958***

Le perceptron est le premier modèle de réseau de neurones, il a été présenté à la fin des années 1950 par Rosenblatt [ROS58]. Ce réseau est constitué de neurones formels de Mc Culloch et Pitts il met en application la règle de Hebb, c'est un réseau à une seule couche de poids adaptatifs. Il est donc à l'origine des premiers résultats probants sur les réseaux connexionnistes. L'organisation générale du perceptron est composée de trois éléments :

- *une rétine* : elle comprend les cellules sur lesquelles s'inscrit le stimulus.
- *une couche de cellules d'association* : chacune des cellules est connectée à des cellules de la rétine (dans le sens rétine vers couche d'association), ou à d'autres cellules d'association ou à des cellules de décision.
- *une couche de cellules de décision* : chacune des cellules est connectée à des cellules d'association (dans le sens couche d'association vers couche de décision), ou à d'autres cellules de décision.

Les cellules d'association ou de décision réalisent la somme des impulsions reçues et répondent suivant une loi du tout-ou-rien après une comparaison avec un seuil, ou elles donnent une réponse modulée. La figure 4.5 propose un schéma général du perceptron, par la suite nous nous limiterons à l'étude d'un perceptron simplifié. Dans ce cas les sorties de toutes les cellules sont binaires, les fonctions des cellules d'association sont booléennes,

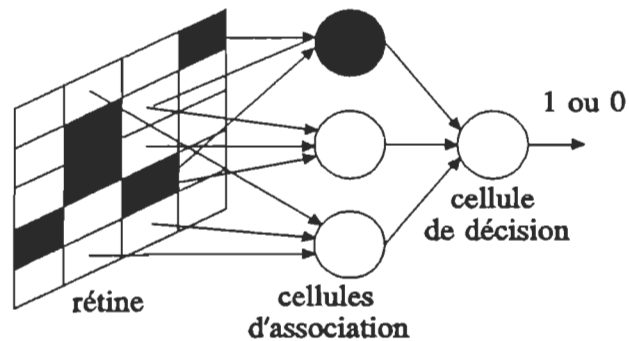


Figure 4.5 Schéma général du perceptron.

les cellules d'association sont des automates à seuil et les connexions se font uniquement d'une cellule d'association vers une cellule de décision.

L'apprentissage du perceptron est un apprentissage supervisé par correction d'erreur. L'erreur se rapporte à la différence entre la sortie obtenue  $s$  et la sortie désirée  $d$ . Mathématiquement la règle s'écrit :

$$w_{ij}(t+1) = w_{ij}(t) + k (d_i - s_i)x_j \quad (4.8)$$

- où -  $w_{ij}$  le poids de connexion  
 -  $k$  une constante positive  
 -  $x_j$  une entrée élémentaire

On vérifie que

$$\Delta w_{ij} = \begin{cases} =0 & \text{si } d_i = s_i \\ \neq 0 & \text{sinon} \end{cases} \quad (4.9)$$

Il est possible de montrer qu'un perceptron simplifié peut calculer une fonction booléenne. Le perceptron peut résoudre des problèmes linéairement séparables. Cependant, une étude mathématique réalisée par Minsky et Papert [MIN] a rigoureusement démontré

une limitation du perceptron. La démonstration est une extension de ce qui suit : *un perceptron n'ayant qu'un neurone de sortie réalise une séparation en deux classes des vecteurs d'entrée présentés*. La frontière est alors définie par l'équation d'un hyperplan:

$$\sum_j w_{ij}x_j = \text{seuil}_i \quad (4.10)$$

#### 4.3.2 Adaline et madaline: 1960

Proposés par B. Widrow en 1960, les modèles de réseaux de neurones *adaline* (ADAPtive LInear NEuron) et *madaline* (multi-adaline) sont des réseaux à deux couches [WID60], de structure analogue au perceptron, ayant respectivement un ou plusieurs neurones sur la deuxième couche (couche de sortie), un réseau adaline est proposé à la figure 4.6. Contrairement au perceptron qui délivre un signal binaire en sortie et pour lequel la correction d'erreur se fait par comparaison entre sorties, obtenue et voulue, binaires, l'adaline et le madaline comparent des sorties analogues, ce qui affine de beaucoup la correction des poids [CAR89].

L'équation de sortie d'un réseau adaline, résultant de la propagation d'un signal d'entrée dans les deux couches du réseau, est donnée par:

$$s_i = f(z_i) = f\left(\sum_j w_{ij}x_j + \theta_i\right) \quad (4.11)$$

- où -  $s_i$  la sortie binaire du réseau  
 -  $z_i$  la sortie analogue du réseau  
 -  $x_j$  une entrée élémentaire  
 -  $w_{ij}$  le poids de connexion  
 -  $\theta_i$  le seuil du neurone

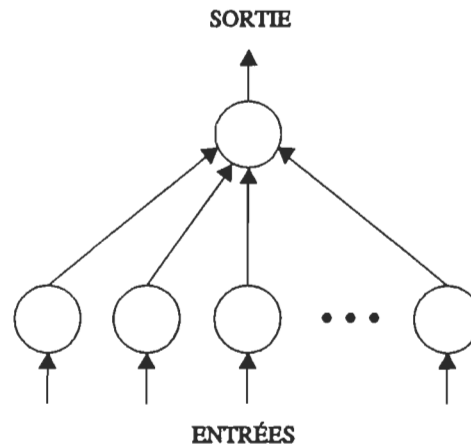


Figure 4.6 Réseau Adaline (ADaptive LInear NEuron).

- $f(\cdot)$  la fonction de transition (fonction seuil)

La règle de correction des poids, appelée règle de *Widrow-Hoff* du nom de ses inventeurs ou encore règle du *Delta*, se base sur un signal d'erreur  $E_i$  qui est la différence entre la sortie désirée et la somme pondérée des entrées non seuillées.

$$E_i = d_i - z_i \quad (4.12)$$

où  $d_j$  est la sortie désirée, la règle de modification des poids est alors:

$$\Delta w_{ij} = k x_j E_i \quad (4.13)$$

où  $k$  est une constante.

Ce choix de l'erreur permet d'obtenir d'une part une solution approchée aux problèmes non linéairement séparables et d'autre part une solution plus robuste car elle tend vers un minimum local de la fonction d'erreur. Si on définit l'erreur totale,

on obtient :

$$E_t = \sum_{\text{exemples}} \left( \sum_{\text{sorties}} \frac{k}{2} (E_i)^2 \right) \quad (4.14)$$

La règle de minimisation des poids réalise une minimisation d'erreur quadratique par descente du gradient.

### 4.3.3 *Le réseau de Hopfield: 1982*

La grande limitation d'utilisation des réseaux à deux couches, en partie due à l'incapacité de corriger les poids de connexion pour des couches cachées supplémentaires, est responsable d'une baisse d'intérêt général pour le domaine des réseaux de neurones dans les années 70. Au début des années 80, des modèles de réseaux bouclés constituent le renouveau dans le domaine des réseaux de neurones, ce sont respectivement le modèle de Hopfield qui est un réseau entièrement connecté et le modèle de Kohonen qui est un réseau entièrement connecté par couches [HOP82].

Le principe de base du modèle de Hopfield consiste à faire évoluer le réseau vers un état stable, à partir d'un état initial quelconque. Le réseau est susceptible de posséder de nombreux états stables et de créer des bassins attracteurs autour de ceux-ci. L'approche de Hopfield est basée sur le fait que le système nerveux recherche des états stables, attracteurs, dans son espace d'états. A ce moment, un état voisin tend à se rapprocher de l'état stable qui lui est le plus proche, ce qui permet la correction des erreurs et la capacité à compléter des informations manquantes. Les réseaux de Hopfield sont des mémoires adressables par leur contenu, ou mémoires auto-associatives, c'est-à-dire qu'un état mémorisé est retrouvé par une stabilisation du réseau. De plus si on supprime une fraction des neurones, les attracteurs sont peu modifiés. Cette robustesse rappelle celle du cerveau

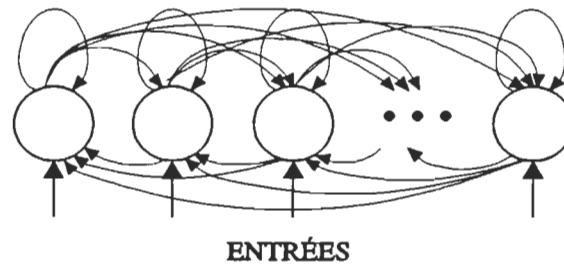


Figure 4.7 Réseau entièrement connecté: modèle de Hopfield.

et s'explique par le fait que l'information est répartie dans tout le réseau.

D'un point de vue structure, ce réseau est constitué de neurones formels de McCulloch et Pitts totalement interconnectés. Les poids de connexion  $w_{ij}$  entre un neurone  $i$  et un neurone  $j$  se présentent sous forme d'une matrice  $\mathbf{W}$ . Dans le cas d'un modèle simple, on considère que  $w_{ij} = w_{ji}$  et  $w_{ii} = 0$  (il n'y a pas de boucles sur le neurone lui-même). La fonction d'activation utilisée est la fonction signe, une modélisation du réseau est proposée à la figure 4.7.

D'un point de vue dynamique, le réseau est mis à jour de façon séquentielle aléatoire, A chaque intervalle de temps, un automate est pris au hasard et est modifié suivant la règle de Hebb qui peut être directement utilisée. En supposant que l'on veuille enregistrer  $n$  prototypes  $\mathbf{x}^p$  ( $\mathbf{x}$  est le vecteur constitué des sorties des  $n$  neurones), les poids sont fixés selon la relation suivante:

$$w_{ij} = \sum_{p=1}^n x_i^p x_j^p \quad \text{et} \quad w_{ii} = 0 \quad (4.15)$$

Ces états doivent être des états stables ayant un bassin attracteur le plus large possible, ainsi un état légèrement différent de  $\mathbf{x}^p$  doit converger et se stabiliser en  $\mathbf{x}^p$ .

Le modèle de Hopfield est intéressant car il est simple et de plus son fonctionnement est bien connu théoriquement. Cependant la capacité de ce modèle est limitée au niveau du nombre d'états mémorisables. Ainsi pour un système de  $n$  neurones, on ne peut enregistrer que  $0.14n$  prototype. De plus si on tente d'apprendre au réseau un modèle de plus, il risque de tout oublier, ce phénomène est appelé *oubli catastrophique*. Une autre difficulté provient du fait que certains états non appris, appelés *états poubelles*, peuvent être des attracteurs. Cependant, de nombreux résultats ont été obtenus à partir du réseau de Hopfield, et des applications utilisent cette structure de réseau dont la principale qualité est la simplicité.

Le modèle de Hopfield a aboutit à deux machines particulièrement intéressantes, la machine de Boltzman qui introduit des notions de probabilité et de température et la machine Gaussienne qui possède des neurones plus complexes et promet de bonnes performances.

#### 4.3.4 Le réseau de Kohonen: 1984

Ce modèle se distingue par l'utilisation d'un neurone formel spécifique, une structure de réseau relativement complexe et surtout un mode d'apprentissage de type *supervisé*. Le neurone formel utilisé est caractérisé par le fait qu'il évolue d'un état à un autre en passant par une série de transitions rapides [KOH84]. La stabilisation de l'état de sortie nécessite donc un temps plus important que le temps de variation des entrées. On a alors une limitation sur la rapidité de variation des états du réseau. La loi de fonctionnement du neurone formel est la suivante :

$$s = f\left(\sum_i w_i x_i\right) \quad (4.16)$$

où -  $s$  la sortie

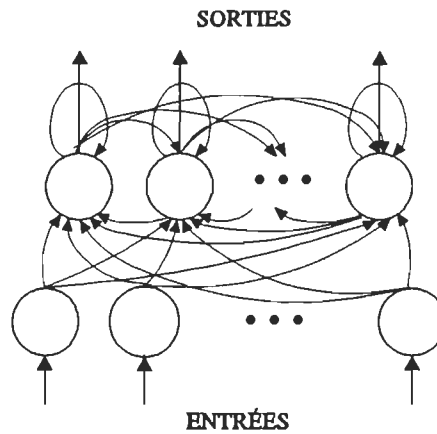


Figure 4.8 Réseau entièrement connecté par couches: modèle de Kohonen.

- $w_i$  un poids synaptique
- $x_i$  une entrée élémentaire
- $f(\cdot)$  une fonction sigmoïde

La structure du réseau est représentée à la figure 4.8, c'est un réseau entièrement connecté par couches à une seule couche. On peut considérer ce réseau comme un réseau entièrement connecté, relié en amont avec une couche de cellules d'entrée. On distingue dans ce modèle des connexions internes et externes, les poids de connexions sont stockés dans deux matrices distinctes. Le mécanisme d'apprentissage est le suivant :

- si le neurone  $i$  reçoit un potentiel d'action de la part d'un neurone  $j$  et qu'il s'active, alors la connexion entre ces deux neurones est renforcée.
- si le neurone  $i$  ne reçoit pas de potentiel d'action, ou qu'il ne s'active pas en ayant reçu un potentiel d'action, alors le poids de la connexion diminue.



L'équation d'apprentissage proposée par Kohonen est:

$$\frac{dw_i}{dt} = k s x_i - \beta(s) w_i \quad (4.17)$$

où le terme  $\beta(s) w_i$  est un terme d'oubli,  $\beta(s) \geq 0$  est la fonction d'oubli et  $k$  un paramètre en général supérieur à zéro. Ceci permet d'avoir un mécanisme d'apprentissage qui tient compte des phénomènes d'oubli qui sont proportionnels à l'inactivité des neurones.

La principale qualité de ce modèle est de posséder une ouverture sur le monde extérieur et d'avoir une possibilité de traitements internes grâce aux connexions modifiables entre les neurones. Nous pouvons citer quelques domaines d'applications du réseau de Kohonen, afin de comprendre l'importance de la structure utilisée.

- *Les cartes topologiques autoadaptatives* : ces réseaux s'inspirent de la modélisation des systèmes de perception chez les mammifères. Chaque neurone du réseau est incité à répondre à un type de stimulus particulier. En classification cela revient à associer un neurone à chaque classe.
- *Les filtres adaptatifs* : les réseaux sont considérés comme des filtres de fonction de transfert  $F$ . A un vecteur d'entrée  $E$  est associé un vecteur de sortie  $S$  tel que  $S = FE$  (notation matricielle).

#### 4.3.5 Les réseaux multicouches: années 1980

Comme nous l'avons dans la section 4.3.1, la limitation démontrée [MIN] des réseaux à deux couches a brusquement mis fin à l'engouement général pour les réseaux de neurones et ce pendant un certain temps. Il est apparu évident qu'il fallait disposer d'un algorithme d'apprentissage susceptible de s'appliquer à des réseaux multicouches (nombre de couches supérieur à deux) pour améliorer les performances de ces derniers et envisager

un renouveau dans le domaine. Le premier pas dans ce sens est du à Werbos qui en 1971 développa un algorithme qui est une généralisation de la règle d'apprentissage de Widrow-Hoff et s'applique aux réseaux multicouches. Cet algorithme appelé *rétropropagation du gradient* a été publié pour la première fois en 1974 comme thèse de doctorat par Werbos [WER74]. Cependant les travaux de Werbos ne sont pas passés à la postérité, malgré leur importance, et il faudra attendre 1982 pour que la méthode soit redécouverte par Parker qui publie en 1985 un rapport à M.I.T. sur ce nouvel algorithme d'apprentissage des réseaux multicouches [PAR82]. En 1986, donc peu de temps après la parution des travaux de Parker, une autre équipe de chercheurs, Rumelhart, Hinton et Williams, redécouvre aussi la méthode de Werbos, la perfectionne, la met en application et finalement réussit à la faire connaître [RUM86b]. Aujourd'hui le nom de Rumelhart et *al.* est en général associé à la méthode dite de *la rétropropagation du gradient* ou *backpropagation algorithm*. Suite à ce bref exposé historique, nous allons nous intéresser à la structure des réseaux multicouches et à l'algorithme de la rétropropagation du gradient.

Les réseaux de neurones multicouches sont caractérisés par une distribution en plusieurs couches des neurones qui les constituent, on distingue la couche d'entrée, la ou les couches cachées et la couche de sortie. Les connexions entre les neurones se font d'une couche vers une autre couche ou d'une couche sur elle-même. Dans le cas où les connexions sont unidirectionnelles et se font d'une couche vers la couche suivante immédiate, on parle de réseaux à couches simples, dans les autres cas on peut avoir des réseaux récurrents (neurones connectés sur eux-mêmes) et/ou des réseaux à couches cachées non simples (connexions vers n'importe quelle couche, y compris elle-même).

Du fait de leur structure, les réseaux présentent un sens de propagation de l'information, une entrée est présentée au réseau (sous forme de vecteur par exemple) et

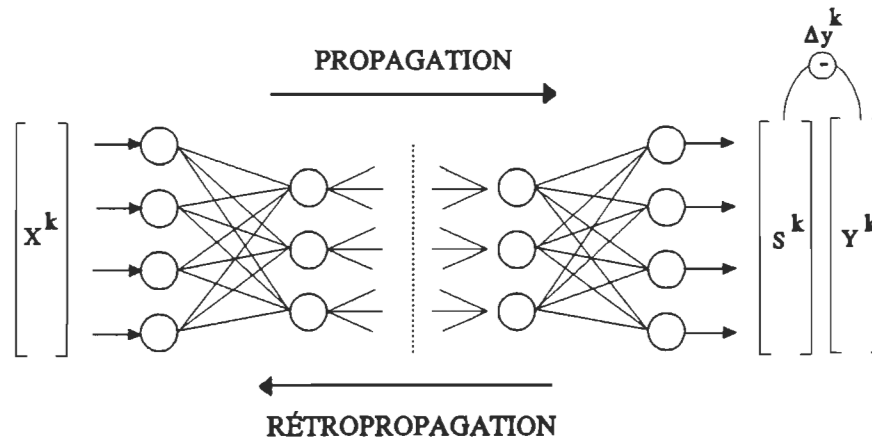


Figure 4.9 Principe de propagation et rétropropagation dans un réseau multicouche.

elle traverse le réseau, toutes les valeurs élémentaires de l'entrée subissent des transformations simultanément. En sortie on obtient un vecteur qui peut alors être comparé à une sortie désirée (figure 4.9). C'est cette comparaison entre le vecteur de sortie obtenu et celui désiré qui va permettre de corriger les poids du réseaux de neurones en rétropropageant l'erreur de propagation obtenue.

L'algorithme de la rétropropagation du gradient est une généralisation de la règle de Widrow-Hoff (voir section 4.3.2) pour les réseaux multicouches. L'idée est de trouver une fonction d'erreur qui ne travaille pas en tout-ou-rien. La solution consiste à changer le neurone formel en utilisant une fonction d'activation dérivable (en général sigmoïde) en remplacement de la fonction à seuil. Il faut alors trouver un mécanisme simple de calcul des dérivées composées. Le principe de fonctionnement du réseau est qu'un signal d'entrée est propagé vers la sortie et l'erreur obtenue entre la sortie désirée et la sortie obtenue est rétropropagée en suivant le chemin inverse, ce mécanisme est répété jusqu'à l'obtention d'un résultat satisfaisant. Pour réaliser la phase d'apprentissage complète, on dispose d'un ensemble d'exemples qui sont des couples *entrée/sortie désirée*, la rétropropagation est

appliquée à chaque modèle, à la fin du processus lorsque l'erreur totale est inférieure au seuil choisi, le réseau a convergé.

Le neurone applique une fonction sigmoïde à la somme pondérée de ses entrées, un exemple de fonction est présenté à la figure 4.2(d). Ce choix d'une fonction sigmoïde est motivé par la nécessité de pouvoir calculer la dérivée de l'erreur quadratique par rapport à chaque poids de pondération.

On pose :

- $\mathbf{x} = (x_1, x_2, \dots, x_n)$  vecteur des entrées (dimension  $n$ )
  - $\mathbf{y} = (y_1, y_2, \dots, y_n)$  vecteur des sorties désirées (dimension  $m$ )
  - $\mathbf{s} = (s_1, s_2, \dots, s_n)$  vecteur des sorties obtenues (dimension  $m$ )
- $\mathbf{s}$  correspond à la propagation de  $\mathbf{x}$  dans le réseau.

Les notations sont les suivantes:

- $f(\cdot)$  fonction sigmoïde,  $f'(\cdot)$  sa dérivée
- $x_i$  sortie du neurone  $i$
- $z_i$  entrée totale du neurone  $i$
- $\mu(k)$  pas du gradient à l'étape  $k$

Les grandes lignes de la formalisation mathématique sont données ci-dessous [DAV89].

L'erreur quadratique  $\Delta y$  est définie par :

$$\Delta y = \sum_{i=1}^m (s_i - y_i)^2 \quad (4.18)$$

La règle de modification des poids à la présentation numéro  $k$  de l'exemple  $\mathbf{x}$  est:

$$w_{ij}(k) = w_{ij}(k-1) - \mu(k) d_i x_j \quad (4.19)$$

où  $d_i$  est calculé de proche en proche de la couche de sortie vers la couche d'entrée.

On envisage deux cas :

1.  $i$  appartient est la couche de sortie

$$d_i = 2(s_i - y_i) f'(z_i) \quad (4.20)$$

$$z_i = \sum_j w_{ij} x_j \quad (4.21)$$

2.  $i$  appartient à une couche cachée

$$d_i = \sum_h d_h w_{hi} f'(z_i) \quad (4.22)$$

où  $h$  parcourt les neurones vers lesquels le neurone  $i$  envoie sa sortie.

Le principal avantage de cet algorithme est d'être local, c'est-à-dire que les calculs d'apprentissage sont effectués au niveau de chaque neurone indépendamment.

Un autre algorithme d'apprentissage dit *renforcé*, similaire aux algorithmes de correction d'erreur (perceptron, rétropropagation du gradient), s'applique aux réseaux de neurones multicouches. Il a été introduit en 1973 par Widrow, Gupta et Maitra et en 1986 Williams propose une version définitive [WIL86]. À la différence des méthodes de correction d'erreur, la méthode de renforcement des poids utilise une information d'erreur non spécifique pour estimer les performances du réseau. Alors que dans le cas des méthodes par correction d'erreur on dispose d'un vecteur d'erreur en sortie du réseau (différences entre le vecteur de sortie obtenu et celui désiré), dans la méthode de

renforcement une seule valeur est utilisée pour décrire la performance de la couche de sortie du réseau. Cette forme d'algorithme est particulièrement intéressante dans le cas d'application où une information spécifique n'est pas disponible alors que l'ensemble des informations sur la performance du réseau l'est, ce qui est le cas en prédiction et contrôle.

#### 4.3.6 Les réseaux ART (*Adaptive Resonance Theory*): 1987

Pour compléter cette section sur les types de réseaux de neurones, nous allons introduire un nouveau type de réseau apparu à la fin des années 80. Les réseaux ART (*Adaptive Resonance Theory*) sont considérés comme les réseaux les plus complexes actuellement utilisés. Sans entrer dans les détails de la théorie de la résonance adaptative [GRO76], nous allons voir les principes de base de fonctionnement de ces réseaux très prometteurs en terme de performances d'apprentissage, de rapidité et surtout d'implantation *hardware*. Les réseaux ART ont été introduits pour résoudre le problème de stabilité du réseau en phase d'apprentissage, ce problème est à l'heure actuelle un des principaux obstacles à franchir dans la recherche sur les réseaux de neurones. L'idée de base de la théorie de la résonance adaptative (ART) est la résolution du problème énoncé précédemment appelé *dilemne de la stabilité-plasticité*: le réseau doit être suffisamment stable pour conserver les connaissances résultant d'un apprentissage antérieur tout en restant suffisamment adaptable pour acquérir de nouvelles connaissances [CAU90].

Le principe de base de la résonance adaptative est un mouvement de va et vient de signaux entre deux couches d'un réseau entièrement connecté. Considérons le réseau de neurones à deux couches de la figure 4.10, la couche d'entrée reçoit les modèles à traiter, la couche de sortie produit la sortie correspondante. Les étapes de fonctionnement sont les suivantes:

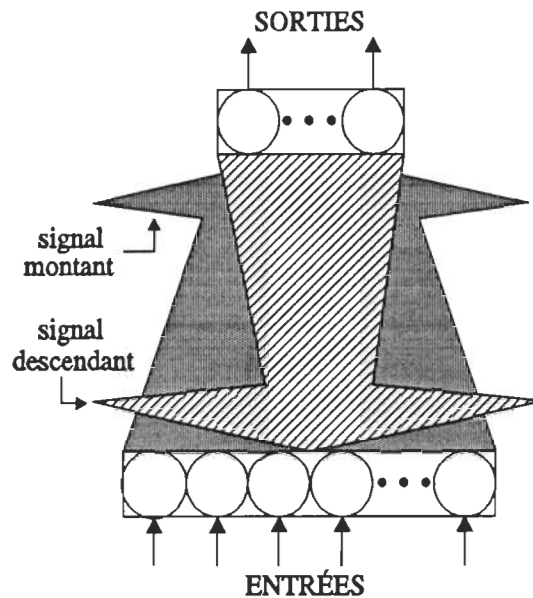


Figure 4.10 Résonance adaptative dans un réseau de neurones à deux couches.

1. Le modèle d'entrée est propagé à travers le réseau jusqu'à la couche de sortie qui doit reconnaître le signal reçu (signal d'entrée modifié par l'interconnexion entrée-sortie du réseau) appelé *signal montant*. La couche de sortie fait alors une estimation de la classe d'appartenance du *signal montant*.
2. L'estimation de la classe d'appartenance du *signal montant* est renvoyée à travers le réseau jusqu'à la couche d'entrée (signal de sortie modifié par l'interconnexion sortie-entrée du réseau), cette estimation est appelée *signal descendant*.
3. Le *signal descendant* est comparé au modèle d'entrée original, deux cas se présentent: l'estimation est correcte et le signal d'entrée original a donc bien été classé, les interconnexions sont renforcées; l'estimation est incorrecte, le signal d'entrée original a été classé de façon erronée et il est renvoyé dans le réseau pour une nouvelle estimation.

Lorsque le *signal descendant* et le modèle d'entrée original correspondent, le système est dit être en *résonnance adaptative*. Les réseaux ART mettent en application ce principe pour apprendre et reconnaître tout en résolvant le *dilemne de la stabilité-plasticité*.

## 4.4 Propriétés et applications des réseaux de neurones

Cette dernière section du chapitre a pour but de mettre en évidence les propriétés respectives des différents types de réseaux de neurones, les propriétés des différents modes d'apprentissage, ainsi que leurs domaines d'application respectifs et ce dans le but de choisir le type de réseau le plus approprié pour notre application en système de mesure.

### 4.4.1 Propriétés des réseaux de neurones

#### *Le parallélisme:*

Cette notion se situe à la base de l'architecture des réseaux connexionnistes qui sont considérés comme des ensembles d'entités élémentaires travaillant simultanément. Le parallélisme permet d'effectuer des calculs rapides, comparés aux méthodes séquentielles, mais il nécessite de repenser la définition du problème à résoudre.

#### *La capacité d'adaptation:*

Cette propriété se manifeste d'une part dans l'aptitude du réseau à apprendre sous de nouvelles contraintes et de nouvelles données et d'autre part dans la capacité de s'auto-organiser, ce qui assure la stabilité. Cette capacité permet de tenir compte de l'évolution du problème traité et/ou d'apprendre à le résoudre autrement.

#### *La mémoire distribuée:*

Dans les réseaux connexionnistes, la mémoire correspond à l'état d'activation de l'ensemble des neurones, elle est donc distribuée sur de nombreuses entités et permet une



grande résistance au bruit. La perte d'un élément ne correspond pas à la perte d'un fait mémorisé, cette propriété est appelée robustesse, de plus les mémoires distribuées permettent de renvoyer la carte d'activation neuronale, à partir d'une donnée bruitée, de la donnée non bruitée correspondante.

*La capacité de généralisation:*

Cette propriété se traduit par l'aptitude du réseau de neurones, à partir d'ensembles d'exemples, à apprendre à retrouver des règles sous-jacentes ou à mimer les comportements qui permettent de résoudre les problèmes. On parle alors de généralisation de règles à partir d'une base d'exemples.

**4.4.2 Propriétés des modes d'apprentissage**

Nous avons vu dans les précédentes sections que l'algorithme d'apprentissage d'un réseau de neurones est directement lié avec la topologie de celui-ci, les topologies étant très diverses, les algorithmes le sont aussi. Dans cette section nous allons revoir les principaux algorithmes qui ont été décrits en nous intéressant aux six propriétés fondamentales suivantes [SÀN92]:

1. *temps d'apprentissage*: permet de décrire comment le réseau stocke les informations d'apprentissage: rapidement, lentement, très lentement ou extrêmement lentement?
2. *en-ligne / hors-ligne*: l'apprentissage est-il une phase distincte de la phase d'utilisation du réseau (hors-ligne) ou simultanée avec la phase d'utilisation (en-ligne)?
3. *supervisé / non-supervisé*: la technique d'apprentissage nécessite-t-elle un superviseur ou non?

Algorithme d'apprentissage	temps d'apprentissage	en-ligne / hors-ligne	supervisé / non-supervisé	linéaire / non-linéaire	structurel / temporel	capacité de stockage
règle de Hebb	rapide	en-ligne	non-supervisé	linéaire	structurel	pauvre
compétitif (Kohonen, HART)	lent	en-ligne	non-supervisé	linéaire	structurel	bonne
correction d'erreur (Adaline)	lent	hors-ligne	supervisé	linéaire	les deux	bonne
rétropropagation (multicouche)	très lent	hors-ligne	supervisé	non-linéaire	les deux	très bonne
renforcé (multicouche)	extrêmement lent	hors-ligne	supervisé	non-linéaire	les deux	bonne
systèmes bouclés (Hopfield)	rapide	hors-ligne	supervisé	non-linéaire	structurel	bonne

Tableau 4.1 Qualités des différentes méthodes d'apprentissage.

4. *linéaire / non-linéaire*: la technique d'apprentissage est-elle capable d'apprendre des processus non linéaires?
5. *structurel / temporel*: l'algorithme d'apprentissage permet-il d'emmagasiner des informations structurelles, temporelles ou les deux?
6. *capacité de stockage*: la capacité de stockage est-elle bonne relativement au nombre de connexions dans le réseaux?

Le tableau 4.1 regroupe l'ensemble des méthodes d'apprentissage vues jusqu'à présent en comparant leurs qualités respectives.

#### 4.4.3 Domaines d'applications des réseaux de neurones

Les réseaux de neurones sont des systèmes de traitement d'information, les principales opérations qu'ils sont aptes à réaliser peuvent être classées de la manière suivante:

*La classification:*

Un modèle d'entrée est propagé à travers le réseau de neurones, celui-ci produit sa classe d'appartenance en sortie.

*L'association ou reconnaissance de formes:*

Un modèle d'entrée est propagé à travers le réseau de neurones, celui-ci produit le modèle de sortie correspondant.

*La complémentation de formes:*

Un modèle incomplet d'entrée est propagé dans le réseau de neurones, celui-ci produit un modèle de sortie qui est composé des portions manquantes du modèle d'entrée.

*Le filtrage:*

Un modèle d'entrée bruité est propagé à travers le réseau de neurones, celui-ci produit un modèle de sortie qui est une version filtrée (partiellement ou entièrement) du modèle d'entrée.

*L'optimisation:*

Un modèle d'entrée représentant les valeurs initiales d'un problème d'optimisation est propagé à travers le réseau de neurones, celui-ci produit un modèle de sortie qui représente la solution du problème.

*Le contrôle:*

Un modèle d'entrée représentant l'état actuel du contrôleur et la réponse désirée pour le contrôleur est propagé à travers le réseau de neurones, celui-ci produit en sortie la commande permettant l'obtention de la réponse désirée.

Type de réseau	classification	hétéroassociation / autoassociation	traitement d'images	optimisation	contrôle
Perceptron	*	hétéro			
Adaline, madaline	*	hétéro			
Hopfield	*	auto		*	
Kohonen	*	auto	*	*	
multicouches	*	hétéro	*		*
ART	*	hétéro	*		*

Tableau 4.2 Classification des réseaux de neurones selon leurs domaines d'application.

D'autres applications des réseaux de neurones peuvent être envisagées mais en général elles s'apparentent à l'une des classes précédentes. Le tableau 4.2 associe les types de réseaux de neurones et leurs domaines courants d'application.

## 4.5 Conclusion

Les premiers travaux dans le domaine des réseaux de neurones datent des années 1940 avec le neurone de Mac Culloch et Pitts, et la première réalisation de Rosenblatt : le perceptron. Cependant la mise en évidence de la limitation de ce modèle par Minsky et Papert a brutalement entraîné une baisse d'intérêt général. Le renouveau des réseaux de neurones au début des années 1980 est dû au progrès réalisé dans le domaine du connexionnisme.

Les principaux événements apparus résultent de nouvelles structures de réseaux, en particulier les réseaux multicouches et de règles d'apprentissage très performantes telle que la rétropropagation du gradient et ses variantes sans cesse plus rapides et performantes, et ce en grande partie grâce à la puissance des calculateurs aujourd'hui disponibles. Ces réseaux se révèlent très performants comme mémoires hétéroassociatives et système de

résolution de problèmes de classification, reconnaissance, filtrage... Des modèles monocouches ont aussi donné des résultats encourageants en tant que mémoires autoassociatives et système de résolution de problèmes d'optimisation (modèle de Hopfield et le modèle de Kohonen). Les réseaux ART quant à eux sont encore au stade de la recherche et promettent beaucoup en terme de performances de mémorisation, d'apprentissage en-ligne et surtout d'implantation matérielle, cependant ils sont au début de leur exploitation et présentent le désagrément d'être complexes.

Dans ce chapitre, les domaines actuels d'applications des réseaux connectionnistes ont été présentés, ils sont très variés et permettent de concevoir de nouvelles utilisations originales et qui exploitent les nombreuses qualités de ces systèmes. Pour l'application aux systèmes de mesures envisagée dans ce travail, notre choix concernant le type de réseaux de neurones s'est porté sur les réseaux multicouches avec algorithmes d'apprentissage basés sur la rétropropagation du gradient. Ce choix a été dicté par la topologie de ces réseaux qui s'accorde avec le type de données à traiter (vecteur à l'entrée, scalaire en sortie), la capacité d'apprentissage et de stockage élevée et leur facilité d'utilisation, cependant la phase d'apprentissage apparaît fastidieuse et le choix d'un algorithme d'apprentissage puissant et rapide va se révéler capital.

## CHAPITRE 5

### RÉSEAUX DE NEURONES MULTICOUCHES

#### 5.1 Introduction

Nous avons vu que le nombre de types de réseaux de neurones et d'algorithmes d'apprentissage est aujourd'hui très important et après le passage en revue des principaux, nous allons restreindre notre étude aux réseaux de neurones multicouches. Ce type de réseaux montre de très bonnes aptitudes à la reconnaissance de formes et à la classification, et ses qualités vont être à la base de la méthode d'étalonnage et de reconstitution de mesurandes dans le cas du système de mesure de pression à fibres optiques. Dans ce cas précis d'application, le problème à résoudre s'apparente à un problème de reconnaissance de formes, à partir des trois tensions délivrées par les trois capteurs à fibre optique du système de mesure, le réseau doit fournir à sa sortie une réponse qui est la pression reconstituée. Nous allons donc voir comment un réseau de neurones peut accomplir la tâche qui lui est assignée, et ce en distinguant deux phases d'opération: l'*apprentissage* du réseau à l'aide d'un algorithme approprié va correspondre à l'*étalonnage* du système de mesure et son utilisation en *reconnaissance* va correspondre à la *reconstitution de mesurande*. Ce chapitre se décompose de la façon suivante: suite à un rappel sur la structure des réseaux de neurones multicouches, nous étudierons trois algorithmes d'apprentissage afin d'évaluer leurs performances et opter pour le plus intéressant. Les tests d'évaluation seront faits en traitant des modèles simples générés par le système de mesure de pression à fibres optiques vu dans le chapitre 3.

## 5.2 Notions de base

### 5.2.1 Structure des réseaux multicouches

Nous limitons l'étude à des réseaux multicouches dont les couches cachées sont simples, c'est-à-dire qu'un neurone ne peut être connecté qu'avec des neurones de la couche suivante. Pour simplifier nous considérons que la connexion d'un neurone avec sa couche suivante est totale, donc chaque neurone est connecté avec l'ensemble des neurones situés en aval. Le réseau comporte une couche d'entrée qui permet de recevoir les données d'entrée; une couche de sortie qui émet le résultat de la propagation de l'entrée; un certain nombre de couches cachées qui transmettent l'entrée dans le réseau.

Le neurone utilisé est un automate continu, ses entrées et sa sortie sont réelles, la fonction de décision est sigmoïde, c'est donc une version "lissée" de la fonction à seuil. La fonction sigmoïde est définie par l'équation suivante:

$$f(z) = \frac{1}{1 + e^{-z}} \quad (5.1)$$

Les neurones de la couche d'entrée sont caractérisés par le fait qu'ils ne possèdent qu'une seule entrée qui n'est pas pondérée (le poids de pondération est pris égal à 1), ceux de la couche de sortie ont des sorties non connectées qui permettent de disposer du résultat de la propagation de l'information dans le réseau. L'équation (5.2) décrit mathématiquement le fonctionnement d'un neurone, les connexions se font dans le sens des neurones  $j$  vers le neurone  $i$  :

$$s_i = f\left(\sum_{j=1}^{N_j} w_{ij}x_j - \theta_i\right) \quad (5.2)$$

où -  $x_j$  une entrée élémentaire du neurone  $i$ ,

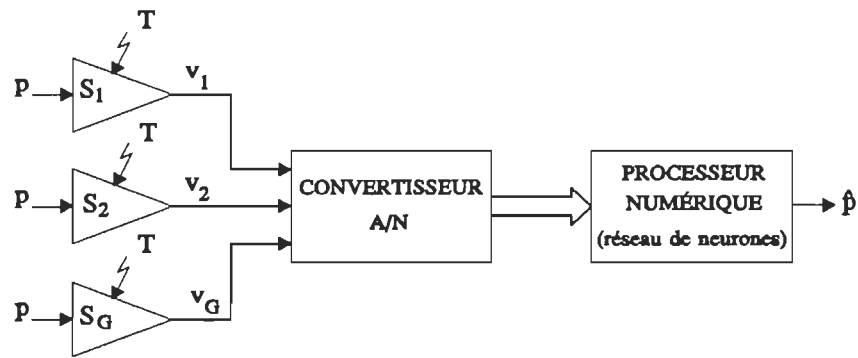


Figure 5.1 Schéma de principe du système de mesure.

- $s_i$  la sortie du neurone  $i$ ,
- $\theta_i$  le seuil du neurone  $i$ ,
- $w_{ij}$  le poids de connexion du neurone  $j$  vers le neurone  $i$ ,
- $f(\cdot)$  la fonction de transition sigmoïde
- $N_j$  nombre de neurones  $j$  connectés au neurone  $i$

Le nombre de neurones de chaque couche est variable, de même pour le nombre de couches cachées, cependant dans la majorité des cas on se limite à quelques couches cachées (inférieur à cinq). La couche d'entrée reçoit l'information et ne possède pas de poids adaptatifs (les poids sont pris égaux à un); les couches cachées sont caractérisées par des matrices de poids de connexion notées  $\mathbf{W}$  et contenant les poids  $w_{ij}$  (connexion dans le sens de la couche  $j$  vers la couche  $i$ ); la couche de sortie fournit le résultat.

Nous allons maintenant considérer l'application du réseau de neurones multicouche au système de mesure à fibres optiques afin de déterminer la topologie du réseau. Le système de mesure peut être schématisé tel que proposé à la figure 5.1: la sortie du convertisseur analogique/numérique fournit trois signaux numériques qui sont présentés au réseau de neurones, ce dernier aura donc trois entrées; la sortie du processeur numérique



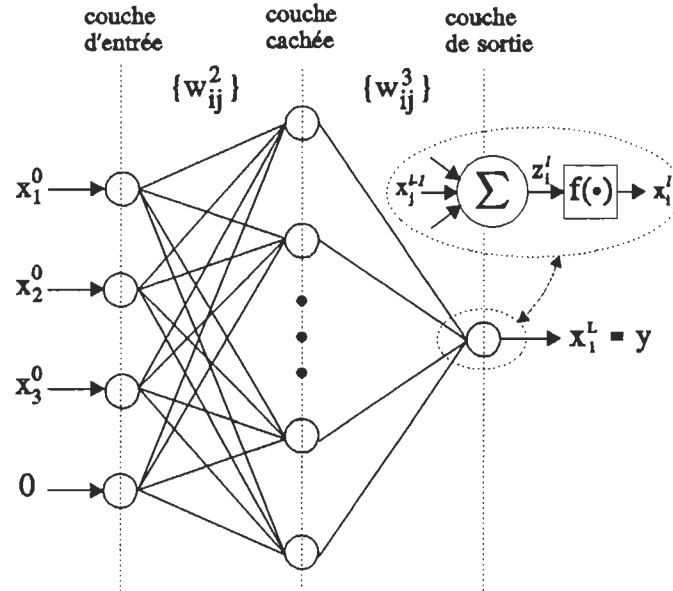


Figure 5.2 Structure du réseau de neurones à trois couches.

fournit la pression reconstituée, c'est le réseau de neurones qui joue ce rôle et donc il aura une sortie unique. Pour ce qui est du nombre de couches cachées, notre choix s'est restreint à une seule couche cachée pour les raisons suivantes: l'apprentissage d'un réseau à trois couches est plus aisé que pour des réseaux à quatre couches ou plus et les valeurs initiales des poids de connexion ont moins d'influence sur les performances finales du réseau [LIP87], [VIL92]. L'utilisation d'un réseau à trois couches semble donc un compromis acceptable en terme de complexité et de performances du réseau de neurones. La structure du réseau utilisé dans toute la suite du travail est présentée à la figure 5.2, il est à noter que ce réseau possède quatre entrées contrairement à ce qui a été écrit précédemment, en fait le quatrième neurone n'est introduit que dans le but de créer le décalage sur la fonction sigmoïde, son entrée est toujours nulle et son effet est équivalent à l'effet de  $\theta_i$  dans l'équation (5.2).

### 5.2.2 Apprentissage et reconnaissance

#### *La phase d'apprentissage*

Les réseaux multicouches sont de mémoires hétéroassociatives dont les connaissances sont mémorisées dans les coefficients synaptiques, ces poids sont fixés lors du processus d'apprentissage. L'un des principes de base des réseaux de neurones est que les coefficients synaptiques sont totalement dépendants les uns des autres, en effet, pris séparément ils n'ont aucune signification. Ce sont les poids de connexion et l'ensemble du réseau qui définissent les capacités de ce dernier.

L'apprentissage utilisant la méthode de rétropropagation du gradient de l'erreur et ses méthodes dérivées est de type supervisé avec de nombreux exemples, il se fait par modifications successives des poids de pondération. Le mode supervisé consiste à utiliser des associations entrée/sortie qui sont présentées au réseau par un professeur. La qualité de l'apprentissage est fortement liée au nombre d'associations présentées, il faut donc une bonne bases de données pour pouvoir traiter le plus possible de signaux inconnus. *L'apprentissage consiste à minimiser l'erreur quadratique commise sur l'ensemble des exemples, celle-ci est considérée comme un fonction de poids, par une approximation d'une descente de gradient, comme dans la règle de Widrow-Hoff [DAV89].*

#### *La phase de reconnaissance*

Cette phase consiste à présenter au réseau un signal inconnu  $x$ , il est propagé suivant l'équation (5.2), en sortie on obtient un signal  $s$ . Si la phase d'apprentissage a été correctement réalisée, le résultat obtenu en sortie doit correspondre au signal désiré. Il paraît évident que si le signal présenté en entrée est identique ou est très voisin d'un exemple appris, le résultat de reconnaissance doit être bon. Cependant, si nous considérons la capacité de *généralisation* des réseaux de neurones, celui-ci doit être capable de donner une réponse satisfaisante à une entrée qui ne fait pas partie des exemples à partir

desquels il a appris. Cette propriété est à la base des bonnes performances obtenues en reconnaissance de formes.

La phase de reconnaissance est donc une simple propagation d'un signal d'entrée dans le réseau, les calculs se font à partir des poids de pondération qui ont été fixés de manière optimale dans la phase d'apprentissage. Cette phase de calcul est très rapide car elle utilise la propriété de parallélisme des réseaux de neurones, les calculs sont effectués au niveau de chaque cellule élémentaire du réseau. Une fois la phase d'apprentissage effectuée, nous pouvons considérer que l'utilisation du réseau de neurones est très rapide.

### 5.2.3 Données d'apprentissage

Le but de ce chapitre est d'évaluer les performances de trois algorithmes d'apprentissage des réseaux de neurones multicouches pour pouvoir par la suite disposer d'un réseau fonctionnel et applicable au système de mesure. Pour ce faire nous avons tester les algorithmes en fonction du temps d'apprentissage nécessaire pour apprendre un ensemble de données synthétiques. Ces données sont générées selon la formule suivante [BAR92]:

$$v_i = V_0 \cos^2[(Ap + BT + C)\pi] \quad \text{pour } i = 1, 2 \text{ et } G \quad (5.3)$$

qui est un modèle idéalisé du capteur de pression à fibre optique et dont la justification est démontrée dans [BAR90]. Les valeurs des paramètres  $A$ ,  $B$ ,  $C$  et  $V_0$  pour les trois capteurs  $S_1$ ,  $S_2$  et  $S_G$ , soient les valeurs permettant de calculer  $v_1$ ,  $v_2$  et  $v_G$  respectivement, sont données par le tableau 5.1. Nous disposons alors d'un ensemble de données synthétiques d'apprentissage simulées selon l'équation (5.3) tel que:

	A	B	C	V <sub>0</sub>
S <sub>1</sub>	1/25	1/100	-1/4	5.0
S <sub>2</sub>	1/25	1/100	1/2	5.0
S <sub>G</sub>	1/300	1/400	25/48	5.0

Tableau 5.1 Valeurs des paramètres du modèle, équation (5.3), des capteurs S<sub>1</sub>, S<sub>2</sub> et S<sub>G</sub>.

$$\{ T_m, \{ v_{1,n,m}, v_{2,n,m}, v_{G,n,m}, P_{n,m} \mid n = 1, \dots, N_P \} \mid m = 1, \dots, N_T \} \quad (5.4)$$

où  $N_T$  et  $N_P$  sont respectivement les nombres de températures et de pressions utilisées pour la génération des données. Dans le cas présent nous avons choisi de générer les données avec:

$$\begin{aligned} N_T &= 1 && \text{avec} && T = 15 \text{ } ^\circ\text{C} \\ N_P &= 30 && \text{avec} && P \in [1,26] \text{ PMa} \end{aligned}$$

Les données synthétiques générées sont prétraitées avant d'être appliquées au réseau de neurones afin de travailler dans les plages de sensibilité de la fonction de transition sigmoïdale définie par l'équation (5.1). Les données d'entrée sont maintenues dans l'intervalle [0,1] et les données de sortie dans l'intervalle [0.25,0.75] en appliquant les transformations suivantes:

- données d'entrée

$$x_i^0 = \frac{v_i - v_{i \min}}{v_{i \max} - v_{i \min}} \quad \text{pour } i = 1, 2 \quad (5.5)$$

$$x_3^0 = \frac{v_G - v_{G \min}}{v_{G \max} - v_{G \min}}, \quad (5.6)$$

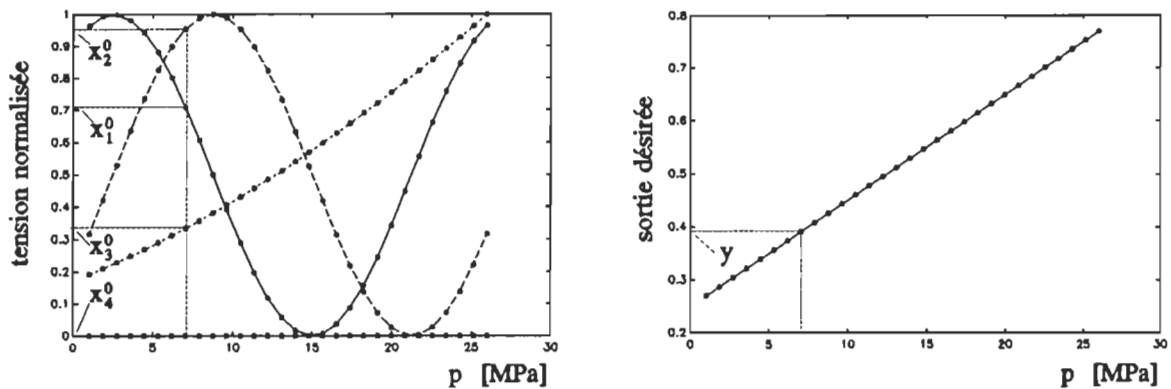


Figure 5.3 Données d'apprentissage d'entrée (a) et de sortie (b).

- données de sortie

$$y = \frac{P - P_{\min}}{2(P_{\max} - P_{\min})} + 0.25 \quad (5.7)$$

Les représentations graphiques de ses données d'apprentissage sont proposées à la figure 5.3.

### 5.3 Les algorithmes d'apprentissage

Trois algorithmes ont été testés: la *méthode de la rétropropagation* et deux variantes de cette méthode, dans les trois cas les notations sont les suivantes:

• données d'apprentissage et apprentissage

- $k$  indice d'itérations,  $k = 0$  correspond à l'état initial,
- $x_{n,m}^0$  vecteur d'apprentissage d'entrée,
- $y_{n,m}$  sortie désirée associée à  $x_{n,m}^0$ , c'est un scalaire car le réseau n'a qu'un neurone de sortie,
- $\hat{y}_{n,m}$  sortie obtenue après propagation de  $x_{n,m}^0$ ,
- $\delta\hat{y}(k)$  erreur relative entre la sortie obtenue  $\hat{y}_{n,m}$  et la sortie désirée  $y_{n,m}$ ,

- $\Delta\hat{y}(k)$  erreur absolue moyenne entre la sortie obtenue  $\hat{y}_{n,m}$  et la sortie désirée  $y_{n,m}$ ,
- $E^l(k)$  erreur quadratique entre la sortie obtenue  $\hat{y}_{n,m}$  et la sortie désirée  $y_{n,m}$ ,

- *réseau de neurones*

- $L$  nombre de couches,
- $N^l$  nombre de neurones par couches pour  $l = 1, \dots, L$ ,
- $W^l(k)$  matrice de poids de connexion pour  $l = 2, \dots, L$ ,
- $w_i^l(k)$  vecteur de poids de connexion associé au neurone  $i$  pour  $l = 2, \dots, L$ ,
- $k^l(k)$  vecteur gain de Kalman pour  $l = 2, \dots, L$ ,
- $\Sigma^l(k)$  matrice de covariance pour  $l = 2, \dots, L$ ,
- $\mu^l(k)$  pas de rétropropagation pour la couche  $l$  à l'itération  $k$ , si le pas est constant il est noté  $\mu^l$ ,

- *neurone*

- $x_i^l(k)$  sortie du neurone  $i$  de la couche  $l$  à l'itération  $k$ ,
- $z_i^l(k)$  entrée totale du neurone  $i$  de la couche  $l$  à l'itération  $k$ ,
- $w_{ij}^l(k)$  poids de connexion entre le neurone  $j$  de la couche  $l-1$  et le neurone  $i$  de la couche  $l$ ,
- $f(\cdot)$  fonction sigmoïde,  $f'(\cdot)$  sa fonction dérivée,  $f^{-1}(\cdot)$  sa fonction inverse,
- $\tilde{r}_i^l(k)$  variance de l'erreur au niveau du neurone  $i$  de la couche  $l$  à l'itération  $k$ ,
- $e_i^l(k)$  erreur de propagation au niveau du neurone  $i$  de la couche  $l$  à l'itération  $k$ ,
- $v_i^l(k)$  bruit d'observation au niveau du neurone  $i$  de la couche  $l$  à l'itération  $k$ ,

*Remarque:* Dans tous les tests d'apprentissage effectués dans ce chapitre, le réseau de neurones utilisé est défini par les caractéristiques suivantes:

$$L = 3, N^1 = 4, N^2 = 50 \text{ et } N^3 = 1.$$

### 5.3.1 La rétropropagation du gradient (RETRO\_GRAD)

Premier algorithme performant pour l'apprentissage des réseaux de neurones multicouches, l'apprentissage consiste à minimiser l'erreur quadratique commise sur l'ensemble des exemples, considérée comme une fonction des poids, par une approximation d'une descente de gradient. Mathématiquement cet algorithme utilise simplement des règles de dérivation composée et donc ne présente pas de difficulté majeure. La principale difficulté rencontrée avec les réseaux multicouches avant la mise au point de cet algorithme était d'effectuer la descente du gradient dans le réseau en calculant la dérivée de l'erreur quadratique par rapport à un poids donné. Ce problème a été résolu simplement en introduisant des neurones à fonction de transition dérivable, c'est le cas des fonctions sigmoïdes [DAV89].

L'algorithme de rétropropagation du gradient présenté ci-dessous est dédié à l'apprentissage des données synthétiques définies à la section 5.2.3. Le principe d'apprentissage est présenté ci-dessous, les connexions se font du neurone  $j$  de la couche  $l-1$  vers le neurone  $i$  de la couche  $l$  vers le neurone  $h$  de la couche  $l+1$ , les étapes d'apprentissage sont les suivantes:

1. Initialisation:  $L, N^1, N^2, N^3, W^2(0), W^3(0), \mu^2$ ,
2. Sélection d'une association entrée/sortie d'apprentissage  $(x_{n,m}^0, y_{n,m})$  en choisissant aléatoirement  $n, m$ ,

3. Propagation de l'entrée  $x_{n,m}^0$  à travers le réseau et obtention de la sortie  $\hat{y}_{n,m}$  selon:

$$x_i^1(k) = f(x_{i,n,m}^0) \quad \text{for } i = 1, \dots, N^1, \quad (5.8)$$

$$\left. \begin{aligned} z_i^l(k) &= \sum_{j=1}^{N^{l-1}} w_{ij}^{l-1}(k) x_j^{l-1}(k) \\ x_i^l(k) &= f(z_i^l(k)) \end{aligned} \right\} \quad \text{for } i = 1, \dots, N^l \text{ and } l = 1, \dots, L, \quad (5.9)$$

$$\hat{y}_{n,m} = x_1^L(k). \quad (5.10)$$

4. Calcul du gradient de l'erreur  $E^l(k)$  pour tous les  $w_{ij}^l(k)$ ,  $l = 2, \dots, L$  :

$$\frac{\partial E^l(k)}{\partial w_{ij}^l(k)} = \frac{\partial E^l(k)}{\partial z_i^l(k)} \cdot \frac{\partial z_i^l(k)}{\partial w_{ij}^l(k)} \quad (5.11)$$

et

$$\frac{\partial z_i^l(k)}{\partial w_{ij}^l(k)} = \frac{\partial}{\partial w_{ij}^l(k)} \left( \sum_{p=1}^{N^{l-1}} w_{ip}^l(k) x_p^{l-1}(k) \right) = x_j^{l-1}(k) \quad (5.12)$$

• pour la couche de sortie  $l = L$  :

$$\begin{aligned} \frac{\partial E^L(k)}{\partial z_i^L(k)} &= \frac{\partial}{\partial z_i^L(k)} \left( \sum_p (y_{p,n,m} - \hat{y}_{p,n,m})^2 \right) = \frac{\partial}{\partial z_i^L(k)} (y_{i,n,m} - \hat{y}_{i,n,m})^2 \\ &= 2(y_{i,n,m} - \hat{y}_{i,n,m}) \frac{\partial \hat{y}_{i,n,m}}{\partial z_i^L(k)} \end{aligned} \quad (5.13)$$



et

$$\frac{\partial \hat{y}_{i,n,m}}{\partial z_i^L(k)} = \frac{\partial x_i^L(k)}{\partial z_i^L(k)} = \frac{\partial}{\partial z_i^L(k)} (f(z_i^L(k))) = f'(z_i^L(k)) \quad (5.14)$$

donc

$$\frac{\partial E^L(k)}{\partial z_i^L(k)} = 2 (\hat{y}_{i,n,m} - y_{i,n,m}) f'(z_i^L(k)) \quad (5.15)$$

• pour les couches cachées  $l = 2, \dots, L-1$  :

$$\frac{\partial E^l(k)}{\partial z_i^l(k)} = \sum_{h=1}^{N^{l+1}} \frac{\partial E^l(k)}{\partial z_h^{l+1}(k)} \cdot \frac{\partial z_h^{l+1}(k)}{\partial z_i^l(k)} \quad (5.16)$$

avec

$$\frac{\partial z_h^{l+1}(k)}{\partial z_i^l(k)} = \frac{\partial}{\partial z_i^l(k)} \left( \sum_{p=1}^{N^l} w_{hp}^{l+1}(k) x_p^l(k) \right) = \frac{\partial}{\partial z_i^l(k)} (w_{hi}^{l+1}(k) x_i^l(k)) \quad (5.17)$$

et

$$\frac{\partial x_i^l(k)}{\partial z_i^l(k)} = \frac{\partial}{\partial z_i^l(k)} (f(z_i^l(k))) = f'(z_i^l(k)) \quad (5.18)$$

donc

$$\frac{\partial E^l(k)}{\partial z_i^l(k)} = \sum_{h=1}^{N^{l+1}} \frac{\partial E^l(k)}{\partial z_h^{l+1}(k)} \cdot w_{hi}^{l+1}(k) \cdot f'(z_i^l(k)) \quad (5.19)$$

5. Modification des poids de connexion,  $l = 2, \dots, L$

$$w_{ij}^l(k+1) = w_{ij}^l(k) - \mu^l \frac{\partial E^l(k)}{\partial w_{ij}^l(k)} \quad (5.20)$$

soit en utilisant les équations (5.11), (5.12), (5.15) et (5.19), on obtient le système d'équations suivant:

- pour la couche de sortie  $l = L$  :

$$w_{ij}^l(k+1) = w_{ij}^l(k) - 2\mu^l \cdot (\hat{y}_{i,n,m} - y_{i,n,m}) \cdot f'(z_i^l(k)) \cdot x_j^{l-1}(k) \quad (5.21)$$

- pour les couches cachées  $l = 2, \dots, L-1$  :

$$w_{ij}^l(k+1) = w_{ij}^l(k) - \mu^l \cdot \sum_{h=1}^{N^{l+1}} \frac{\partial E^l(k)}{\partial z_h^{l+1}(k)} \cdot w_{hi}^{l+1}(k) \cdot f'(z_i^l(k)) \cdot x_j^{l-1}(k) \quad (5.22)$$

6. Si le nombre de rétropropagations effectuées est  $k \cdot \text{modulo} \left( \frac{N_T N_P}{2} \right)$ , alors tous les exemples d'apprentissage sont propagés dans le réseau et l'erreur absolue moyenne est évaluée selon:

$$\Delta \hat{y}(k) = \sqrt{\frac{\sum_{m=1}^{N_T} \sum_{n=1}^{N_P} (\hat{y}_{n,m} - y_{n,m})^2}{N_P N_T}} \quad (5.23)$$

Si  $\Delta \hat{y}(k)$  est supérieur à un seuil préalablement fixé ou si le nombre maximum d'itérations n'est pas atteint alors retourner à l'étape 2., sinon fin de l'apprentissage.

Lorsque le processus d'apprentissage s'arrête et que l'erreur seuil a été atteinte, on dit que le réseau a convergé. Le choix de ne pas calculer l'erreur absolue moyenne sur

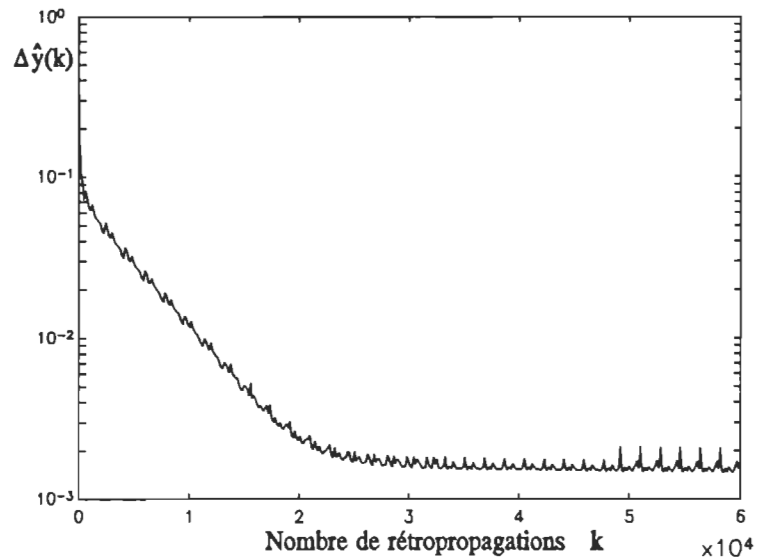


Figure 5.4 Courbe d'apprentissage avec l'algorithme RETRO\_GRAD avec  $\mu^2 = \mu^3 = 0.2$  .

tous les exemples après chaque rétropropagation nous permet de diminuer le temps d'apprentissage qui est déjà très long. Une courbe d'apprentissage du réseau de neurones utilisant la rétropropagation du gradient est présentée à la figure 5.4, l'arrêt de l'apprentissage a été obtenu quand un nombre maximum d'itérations a été atteint, il est important de noter que le réseau était en saturation. Le principal avantage de cette méthode est que la convergence du réseau est très stable, il n'y a pas d'oubli de la part du réseau de neurones, l'apprentissage est donc robuste, mais le temps d'apprentissage reste un inconvénient majeur qui limite de beaucoup l'emploi de cette méthode.

### 5.3.2 Méthode d'optimisation basée sur le filtrage de Kalman (RETRO\_KAL)

Cette méthode proposée dans [SCA92] utilise une forme modifiée de l'algorithme de rétropropagation du gradient et minimise l'erreur quadratique entre la sortie désirée et celle obtenue avec respect aux entrées et aux non-linéarités. Les signaux d'erreurs, générés par l'algorithme, sont utilisés pour estimer les entrées des non-linéarités sous forme de vecteurs à chaque couches respectives, les neurones modifiés permettant l'estimation des

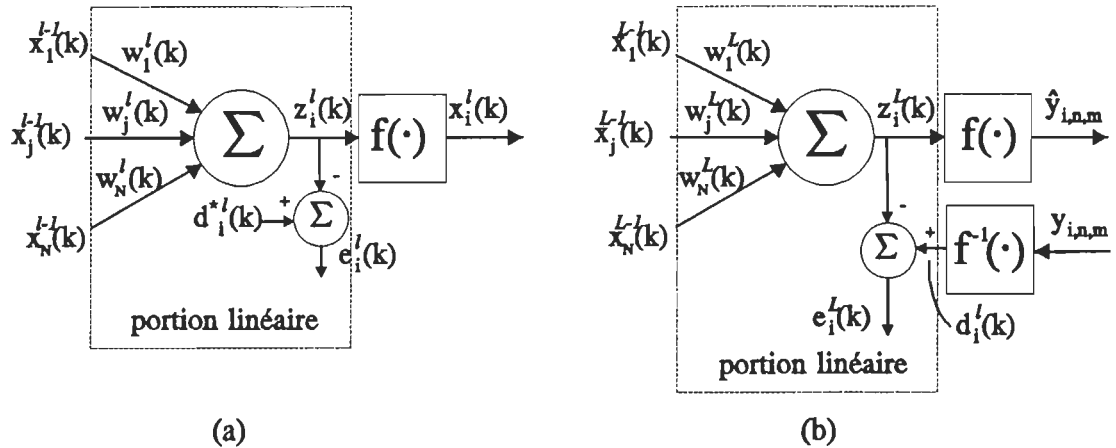


Figure 5.5 Neurones formels modifiés de la couche cachée (a) et de la couche de sortie (b).

erreurs sont proposés à la figure 5.5. Ces estimations sont utilisées pour corriger les poids de pondérations à travers un système d'équations linéaires à chaque couche. Ces systèmes d'équations linéaires sont résolues en utilisant le filtre de Kalman à chacune des couches. D'après [SCA92], cette méthode est très performante en terme de temps d'apprentissage de réseaux de neurones, ce qui est démontré dans le cas de réseaux à trois couches et de neurones traitant uniquement des données numériques, ce qui n'est pas l'application envisagée dans ce travail.

Considérons une portion linéaire du réseau de neurones pour le neurone  $i$  de la couche  $l$  représentée sous la forme du système discret suivant:

$$\mathbf{w}_i^l(k) = \mathbf{w}_i^l(k-1) \quad \text{for } k = 1, 2, 3, \dots \quad (5.24)$$

$$z_i^l(k) = (\mathbf{x}^{l-1}(k))^T \mathbf{w}_i^l(k) + v_i^l(k) \quad (5.25)$$

où le vecteur des poids de connexion  $\mathbf{w}_i^l(k)$  est l'état du système à l'instant  $k$ ,  $z_i^l(k)$  est la sortie du neurone  $i$  avant l'application de la fonction de transition et  $v_i^l(k)$  est le bruit d'observation. Supposons que l'état initial du système  $\mathbf{w}_i^l(0)$  est une variable aléatoire gaussienne de moyenne  $\bar{\mathbf{w}}_i^l(0)$  connue et de covariance  $\Sigma_i^l(0)$  connue, soit:

$$E[\mathbf{w}_i^l(0)] = \bar{\mathbf{w}}_i^l(0), \quad E\{[\mathbf{w}_i^l(0) - \bar{\mathbf{w}}_i^l(0)][\mathbf{w}_i^l(0) - \bar{\mathbf{w}}_i^l(0)]^T\} = \Sigma_i^l(0), \quad (5.26)$$

et que  $\mathbf{w}_i^l(0)$  est indépendant de  $v_i^l(k)$  quelque soit  $k$ , alors le procédé  $\{v_i^l(k)\}$  est un procédé gaussien et

$$E[v_i^l(0)] = 0, \quad E[v_i^l(j)v_i^l(k)] = r_i^l(k)\delta_{jk}, \quad (5.27)$$

pour tous les  $j$  et  $k$ , où  $\delta_{jk}$  est le symbole de Kronecker défini par

$$\delta_{jk} = \begin{cases} 1 & \text{si } j=k \\ 0 & \text{si } j \neq k \end{cases} \quad (5.28)$$

et où  $r_i^l(k)$  est la variance du procédé  $\{v_i^l(k)\}$ .

Sous ces hypothèses, le filtre de Kalman est appliqué pour déterminer un vecteur gain  $\mathbf{k}^l(k)$  pour chaque couche  $l$ , utilisé pour modifier les poids de connexion  $\mathbf{W}^l(k)$  en minimisant la trace de la matrice de covariance  $\Sigma^l(k)$  [AND79]. En utilisant le filtre de Kalman, la convergence du réseau de neurones en phase d'apprentissage est plus rapide qu'avec la méthode de rétropropagation du gradient standard [SIN89], [HUA91], [SCA92] et [IIG92].

L'algorithme d'apprentissage présenté ci-dessous est tiré de [SCA92] et est modifié pour notre application à l'apprentissage des données générées par le système de mesure de pression à fibres optiques. Les données d'apprentissage sont les mêmes que celles utilisées pour tester l'algorithme de la rétropropagation du gradient, voir section 5.2.3. Les étapes principales de l'algorithme sont les suivantes:

1. Initialisation:  $L, N^1, N^2, N^3, \mathbf{W}^2(0), \mathbf{W}^3(0), \Sigma^2(0), \Sigma^3(0), \mu^2, \check{r}^2, \check{r}^3$ ,

2. Sélection d'une association entrée/sortie d'apprentissage  $(x_{n,m}^0, y_{n,m})$  en choisissant aléatoirement  $n, m$ ,

3. Propagation de l'entrée  $x_{n,m}^0$  à travers le réseau et obtention de la sortie  $\hat{y}_{n,m}$  selon:

$$x_i^1(k) = f(x_{i,n,m}^0) \quad \text{for } i = 1, \dots, N^1, \quad (5.29)$$

$$\left. \begin{aligned} z_i^l(k) &= \sum_{j=1}^{N^{l-1}} w_{ij}^{l-1}(k) x_j^{l-1}(k) \\ x_i^l(k) &= f(z_i^l(k)) \end{aligned} \right\} \quad \text{for } i = 1, \dots, N^l \text{ and } l = 1, \dots, L, \quad (5.30)$$

$$\hat{y}_{n,m} = x_1^L(k). \quad (5.31)$$

4. Afin de diminuer l'erreur relative sur la pression reconstituée  $\hat{p}$  sur toute la plage de mesure, l'erreur relative entre la sortie obtenue  $\hat{y}_{n,m}$  et la sortie voulue  $y_{n,m}$  est rétropropagée dans le réseau pour modifier les poids de connexion:

$$\delta \hat{y}(k) = \frac{y_{n,m} - \hat{y}_{n,m}}{y_{n,m}} \quad (5.32)$$

5. L'erreur relative  $\delta \hat{y}(k)$  est rétropropagée en appliquant les équations du filtre de Kalman:

- calcul du vecteur gain de Kalman pour chaque couche  $l$

$$k^l(k) = \frac{\Sigma^l(k-1) x^{l-1}(k)}{\check{\gamma}^l + (x^{l-1}(k))^T \Sigma^l(k-1) x^{l-1}(k)} \quad \text{for } l = 2, \dots, L, \quad (5.33)$$

où la variance estimée  $\check{\gamma}^l$  est constante.

- modification de la matrice de covariance correspondante

$$\Sigma^l(k) = [I - k^l(k) (x^{l-1}(k))^T] \Sigma^l(k-1) (\tilde{r}^l)^{-1} \quad \text{for } l = 2, \dots, L, \quad (5.34)$$

- calcul des erreurs

$$e_i^L(k) = f'(z_i^L(k)) \delta \hat{y}_{n,m} \quad \text{for } i = 1, \dots, N^L \quad (5.35)$$

$$e_i^l(k) = f'(z_i^l(k)) \sum_{j=1}^{N^{l+1}} w_{ji}^l(k) e_j^{l+1}(k) \quad \text{for } i = 1, \dots, N^l \text{ and } l = 2, \dots, L-1 \quad (5.36)$$

- modification des poids de connexion

$$w_i^L(k) = w_i^L(k-1) + k^L(k) [f^{-1}(y_{n,m}) - z_1^L(k)] \quad \text{for } i = 1, \dots, N^L \quad (5.37)$$

$$w_i^l(k) = w_i^l(k-1) + k^l(k) e_i^l(k) \mu^l \quad \text{for } i = 1, \dots, N^l \text{ and } l = 2, \dots, L-1 \quad (5.38)$$

6. Si le nombre de rétropropagations effectuées est  $k \cdot \text{modulo} \left( \frac{N_T N_P}{2} \right)$ , alors tous les exemples d'apprentissage sont propagés dans le réseau et l'erreur absolue moyenne est évaluée selon:

$$\Delta \hat{y}(k) = \sqrt{\frac{\sum_{m=1}^{N_T} \sum_{n=1}^{N_P} (\hat{y}_{n,m} - y_{n,m})^2}{N_P N_T}} \quad (5.39)$$

Si  $\Delta \hat{y}(k)$  est supérieur à un seuil préalablement fixé ou si le nombre maximum d'itérations n'est pas atteint alors retourner à l'étape 2., sinon fin de l'apprentissage.

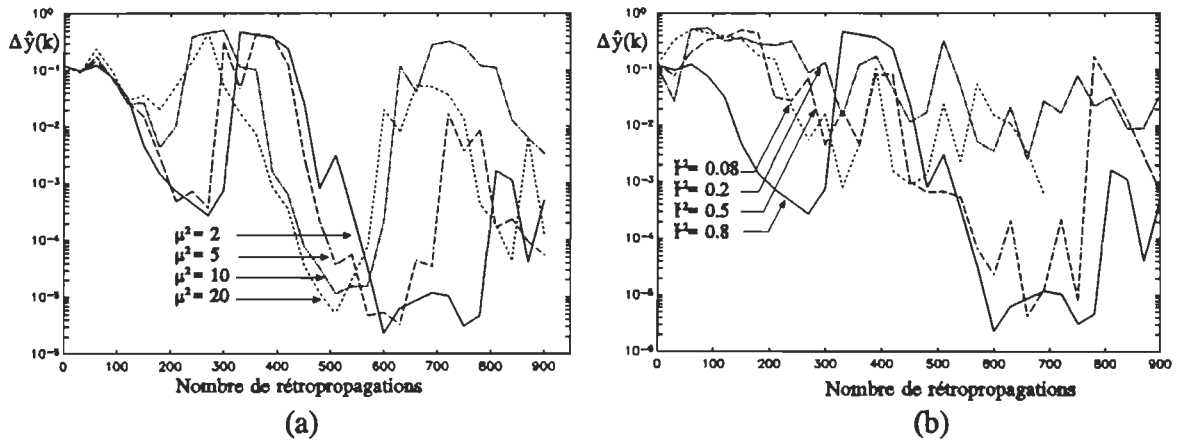


Figure 5.6 Courbe d'apprentissage avec l'algorithme RETRO\_KAL: (a)  $\mu^2$  variable et  $\check{r}^2 = \check{r}^3 = 0.2$ , (b)  $\check{r}^2 = \check{r}^3$  variables et  $\mu^2 = 2$ .

Le nombre d'itérations requis pour obtenir une convergence du réseau est de moins de 20% du nombre requis avec l'algorithme de rétropropagation du gradient standard et les performances du réseau de neurones sont moins affectées par le choix initial des poids de connexion [SCA92].

Différents tests d'apprentissage des données définies à la section 5.2.3 ont été réalisés pour étudier l'influence des paramètres  $\mu^l$  et  $\check{r}^l$  sur la méthode RETRO\_KAL. Les résultats pour différentes valeurs de  $\mu^l$  et  $\check{r}^l$  constants sont présentés à la figure 5.6(a), les résultats pour  $\mu^l$  constants et différentes valeurs de  $\check{r}^l$  à la figure 5.6(b). Les deux paramètres jouent un rôle sur les performances d'apprentissage du réseau et sont liés l'un à l'autre. La variance estimée  $\check{r}^l$  joue un rôle prépondérant sur la stabilité de l'apprentissage, celui-ci est d'autant plus stable que  $\check{r}^l$  est grand ( $\check{r}^l \in ]0, 1[$ ), le pas de rétropropagation  $\mu^l$  influe plus sur la rapidité de convergence ( $\mu^l \in [0, 1]$ ). De ces courbes il faut retenir le résultat suivant, si la convergence du réseau de neurones en phase d'apprentissage est très rapide, moins de 300 rétropropagations, elle est aussi très instable. Le réseau est alternativement en phase d'apprentissage où il converge et en phase d'oubli où il diverge, ce qui est un désavantage



important pour l'utilisation de cette méthode. Pour des performances d'apprentissage optimales, il faudrait pouvoir disposer d'une méthode aussi stable que la rétropropagation du gradient standard (RETRO\_GRAD) et aussi rapide que l'optimisation de cette dernière par le filtre de Kalman (RETRO\_KAL).

*Remarque:* Il est important de noter que l'instabilité de la phase d'apprentissage est constatée dans l'application que nous faisons de la méthode d'apprentissage basée sur le filtrage de Kalman. Une étude plus approfondie serait indispensable pour éventuellement généraliser ce résultat à d'autres applications.

### 5.3.3 Nouvelle méthode proposée (RETRO\_KAL+)

Dans la section 5.3.2, nous avons présenté une version modifiée de l'algorithme de la rétropropagation du gradient très performante en terme de vitesse de convergence, mais instable. Nous allons maintenant proposer une amélioration de cette méthode, notée RETRO\_KAL+, notre but est de stabiliser la convergence du réseau pour éviter le cycle d'apprentissage et d'oubli engendré par l'utilisation du précédent algorithme et de disposer ainsi d'un apprentissage plus robuste [MAS94]. La mise au point de cette méthode est le résultat de l'étude de l'influence des paramètres d'apprentissage  $\eta^l$  et  $\mu^l$  qui ont un effet prépondérant sur les performances du réseau de neurones en phase d'apprentissage. Voici les principales remarques, résultant de l'étude des courbes d'apprentissage obtenues avec les méthodes RETRO\_GRAD et RETRO\_KAL, qui nous ont permis d'améliorer la méthode RETRO\_KAL:

#### 1. Méthode RETRO\_GRAD:

- la convergence du réseau est extrêmement lente et stable,

- la convergence du réseau est d'autant plus lente que les modifications sur les poids sont faibles,
- la correction des poids est fonction de l'erreur de propagation (différence entre la sortie obtenue et la sortie voulue) et du pas de rétropropagation qui pondère la descente du gradient de l'erreur dans le réseau,
- le réseau diverge ou oscille si les modification sur les poids sont trop élevées, donc si le pas de rétropropagation  $\mu^l$  est trop grand,

## 2. *Méthode RETRO\_KAL:*

- la convergence du réseau est très rapide, mais instable et le réseau entre en cycle d'apprentissage et d'oubli,
- l'erreur d'apprentissage minimale est très faible, de l'ordre de  $10^{-5}$ ,
- dans la correction des poids de connexion, la variance estimée  $\hat{\sigma}^l$  est un diviseur, le pas de rétropropagation  $\mu^l$  un multiplicateur, leur rôle sont donc opposés quant à la modification des poids,
- si la variance estimée  $\hat{\sigma}^l$  est très faible, l'algorithme RETRO\_KAL tend vers l'algorithme RETRO\_GRAD,

De ces remarques, nous avons retenu que la convergence du réseau est directement liée à modification apportée aux poids de connexion, cette modification étant proportionnelle à l'erreur de propagation et au pas de rétropropagation  $\mu^l$  et inversement liée à la variance estimée  $\hat{\sigma}^l$ . La modification doit être suffisamment importante pour que la convergence du réseau soit rapide, mais elle ne doit pas entraîner un état oscillatoire du réseau. Nous proposons alors d'introduire dans l'algorithme RETRO\_KAL des paramètres  $\mu^l$  et  $\hat{\sigma}^l$  variables, fonction de l'erreur de propagation commise sur tous les exemples  $\Delta\hat{y}(k)$  à l'itération  $k$  et ce contrairement à [SCA92], selon les équations suivantes:

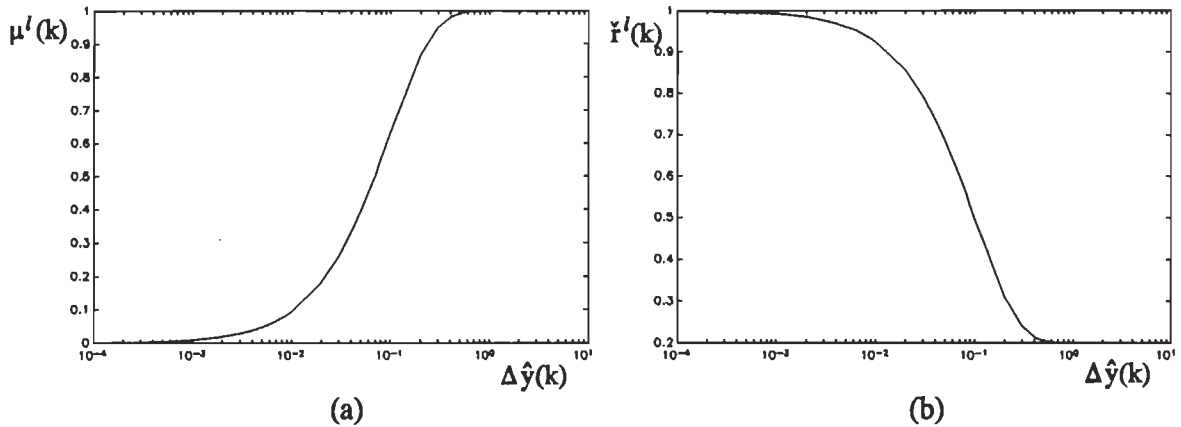


Figure 5.7 Courbes de variation de  $\mu^l(k)$  et  $\check{r}^l(k)$  en fonction de  $\Delta\hat{y}(k)$ ,  $\check{r}_1^l = 0.2$  et  $\mu_3^l = \check{r}_3^l = 0.1$ .

$$\mu^l(k) = \mu_1^l - \mu_2^l e^{-\frac{\Delta\hat{y}(k)}{\mu_3^l}} \quad (5.40)$$

$$\check{r}^l(k) = \check{r}_1^l + \check{r}_2^l e^{-\frac{\Delta\hat{y}(k)}{\check{r}_3^l}} \quad (5.41)$$

où  $\mu_1^l$ ,  $\mu_2^l$ ,  $\mu_3^l$ ,  $\check{r}_1^l$ ,  $\check{r}_2^l$  et  $\check{r}_3^l$  sont des constantes à déterminer. Dans la suite du travail nous avons fixé:

$$\mu_1^l = 1, \quad \mu_2^l = 1, \quad \check{r}_2^l = 1 - \check{r}_1^l,$$

Les variations de  $\mu^l(k)$  et  $\check{r}^l(k)$  en fonction de l'erreur  $\Delta\hat{y}(k)$  sont proposées à la figure 5.7.

L'algorithme d'apprentissage est présenté ci-dessous, il est identique à l'algorithme RETRO\_KAL pour les étapes 1. à 5., seules les variables  $\mu^l$  et  $\check{r}^l$  sont changées puisqu'elles deviennent variables en fonction des itérations, les notations  $\mu^l(k)$  et  $\check{r}^l(k)$ . L'application de l'algorithme correspond à l'apprentissage des données générées par le système de mesure de pression à fibres optiques, les données d'apprentissage sont

les mêmes que celles utilisées pour tester les algorithmes RETRO\_GRAD et RETRO\_KAL, voir section 5.2.3. L'étape 6. est modifiée et devient:

6. Si le nombre de rétropropagations effectuées est  $k \cdot \text{modulo} \left( \frac{N_T N_P}{2} \right)$ , alors tous les exemples d'apprentissage sont propagés dans le réseau et l'erreur absolue moyenne est évaluée selon:

$$\Delta \hat{y}(k) = \sqrt{\frac{\sum_{m=1}^{N_T} \sum_{n=1}^{N_P} (\hat{y}_{n,m} - y_{n,m})^2}{N_P N_T}} \quad (5.42)$$

Pour améliorer les performances du réseau et diminuer le temps d'apprentissage, le pas de rétropropagation  $\mu^l(k)$  et la variance estimée  $\check{r}^l(k)$  sont modifiés en fonction de l'erreur absolue moyenne  $\Delta \hat{y}(k)$

$$\mu^l(k) = 1 - e^{-\frac{\Delta \hat{y}(k)}{\mu_3^l}} \quad \text{pour } l = 2, \dots, L-1, \quad (5.43)$$

$$\check{r}^l(k) = \check{r}_1^l + (1 - \check{r}_1^l) e^{-\frac{\Delta \hat{y}(k)}{\check{r}_3^l}} \quad \text{pour } l = 2, \dots, L, \quad (5.44)$$

Si  $\Delta \hat{y}(k)$  est supérieur à un seuil préalablement fixé ou si le nombre maximum d'itérations n'est pas atteint alors retourner à l'étape 2., sinon fin de l'apprentissage.

Dans le cas de l'apprentissage des données définies à la section 5.2.3, de nombreux tests ont été effectués pour étudier l'influence de ces paramètres sur les performances du réseau de neurones en phase d'apprentissage, les résultats sont transcrits au tableau 5.2 et une sélection des courbes les plus représentatives à la figure 5.8. Pour ces tests, nous avons choisi de donner différentes valeurs à  $\check{r}_3^2$  et  $\check{r}_1^2 = 0.2$ . Les valeurs contenues

$\check{r}_3^l \backslash \mu_3^l$	0.1	0.5	1	2	5	10
1 E-3	2 E-6	1.3 E-6	2 E-6	3 E-6	8 E-6	3 E-6
5 E-3	4 E-6	2 E-6	3 E-6	2.6 E-6	1.7 E-6	2 E-6
8 E-3	7 E-6	6 E-6	8 E-6	7 E-6	5.7 E-6	5 E-6
1 E-2	2 E-6	1 E-5	1.3 E-5	1 E-5	1 E-5	9 E-5
2 E-2	9 E-6	3.3 E-5	4 E-5	4 E-5	3.7 E-5	3.3 E-5
8 E-2	7 E-5	1 E-4	7 E-5	7 E-5	1 E-4	7 E-5

Tableau 5.2 Influence des paramètres  $\mu_3^l$  et  $\check{r}_3^l$  sur les performances du réseau de neurones en phase d'apprentissage.

dans la table 5.2 sont les valeurs d'erreur d'apprentissage minimale, obtenues en fonction de  $\mu^2(k)$ ,  $\check{r}^2(k)$  et  $\check{r}^3(k)$ , nous avons convenu de prendre  $\check{r}^3(k) = \check{r}^2(k)$ . Les variations des paramètres sont données par les équations suivantes:

$$\mu^2(k) = 1 - e^{-\frac{\Delta \hat{y}(k)}{\mu_3^2}} \quad (5.45)$$

$$\check{r}^2(k) = \check{r}_1^2 + (1 - \check{r}_2^2) e^{-\frac{\Delta \hat{y}(k)}{\check{r}_3^2}} \quad (5.46)$$

Les cases ombragées foncées indiquent la présence de fortes oscillations sur la courbe d'apprentissage, les cases ombragées claires indiquent de faibles oscillations et les cases sans ombrage indiquent que l'apprentissage est stable et sans oscillations. De ces résultats obtenus avec la méthode RETRO\_KAL+, nous retenons les points suivants:

- la convergence du réseau est très rapide, plus qu'avec la méthode RETRO\_KAL,
- les valeurs minimales d'erreur d'apprentissage sont très faibles, du même ordre que celles obtenues avec la méthode RETRO\_KAL,

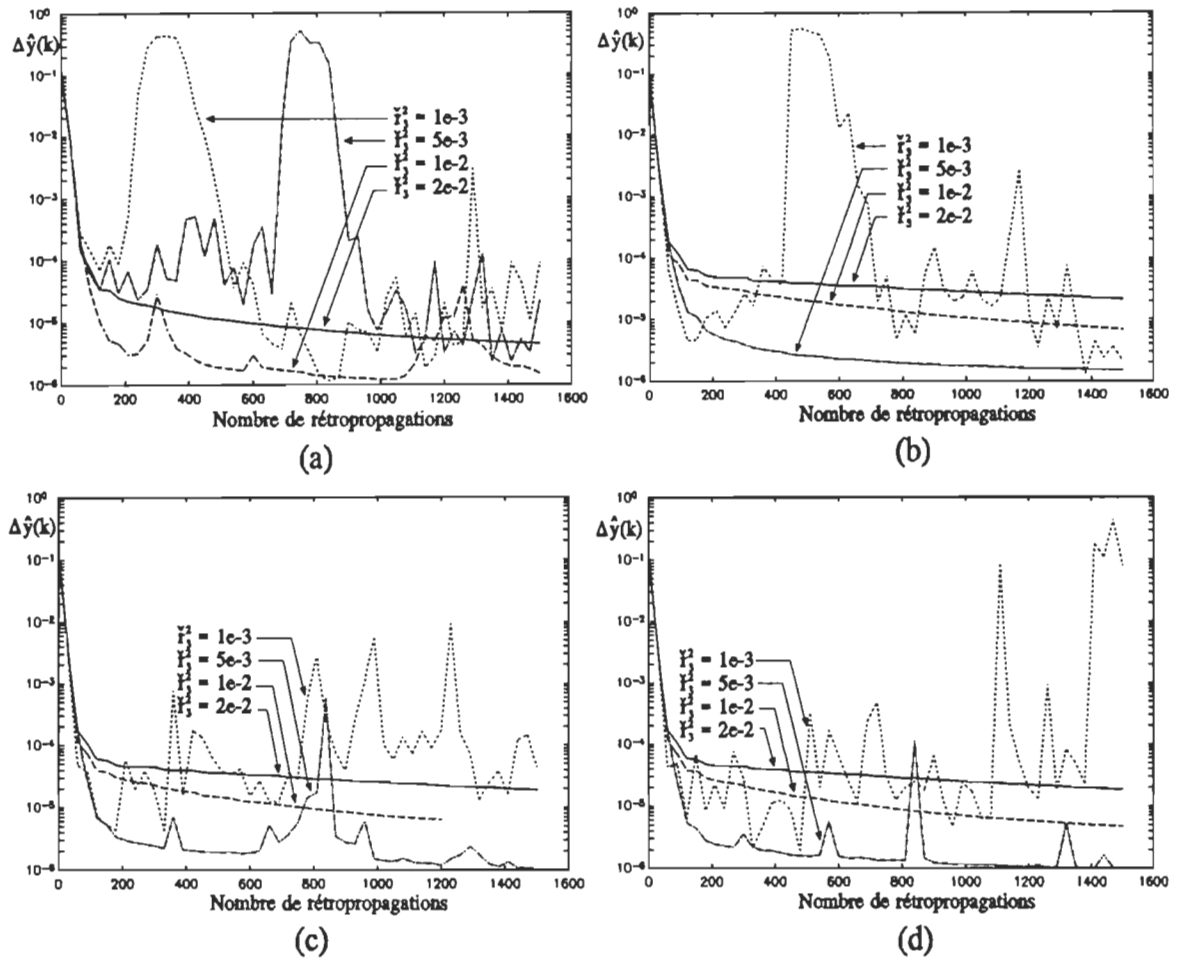


Figure 5.8 Courbe d'apprentissage avec l'algorithme RETRO\_KAL+: (a)  $\mu_3^2 = 0.1$ , (b)  $\mu_3^2 = 1$ , (c)  $\mu_3^2 = 5$ , (d)  $\mu_3^2 = 10$ .

- l'apprentissage du réseau peut être très stable si les paramètres régissant les variables  $\mu^l(k)$  et  $\tilde{\gamma}^l(k)$  sont bien ajustés,
- la stabilisation de l'apprentissage peut être obtenue en diminuant le pas de rétropropagation  $\mu^l(k)$ , c'est-à-dire en augmentant  $\mu_3^l$ , ou en augmentant la variance estimée  $\tilde{\gamma}^l(k)$ , c'est-à-dire en augmentant  $\tilde{\gamma}_1^l$  et/ou diminuant  $\tilde{\gamma}_3^l$ ,
- si l'une des variables  $\mu^l(k)$  ou  $\tilde{\gamma}^l(k)$  a ses paramètres constants (voir équations (5.43) et (5.44)), alors il est possible d'ajuster les paramètres de l'autre variable pour obtenir un apprentissage rapide et stable,

- il existe un dilemne entre l'erreur d'apprentissage obtenue après convergence et la présence ou non d'oscillations: la valeur minimale d'erreur d'apprentissage est obtenue dans le cas d'un apprentissage présentant des instabilités (phases d'oubli qui se traduisent par des oscillations), si les oscillations diminuent ou disparaissent alors l'erreur d'apprentissage augmente. Il existe donc des conditions optimales permettant d'obtenir une erreur d'apprentissage finale faible tout en conservant un apprentissage sans ou avec très peu d'oscillations,

Avec l'algorithme RETRO\_KAL+ nous disposons d'un algorithme d'apprentissage très performant en terme de rapidité et de stabilité, ce qui été le but de cette partie du travail. La nouvelle méthode garde la rapidité de la méthode RETRO\_KAL et elle peut être stabilisée après une étape de mise au point des paramètres régissant d'une part le pas de rétropropagation  $\mu^l(k)$  et d'autre part la variance estimée  $\hat{\sigma}^l(k)$ . Ces deux paramètres, constants dans RETRO\_KAL, sont modifiés au cours de l'apprentissage en fonction de l'erreur de propagation commise sur l'ensemble des données d'apprentissage, pour éviter des modifications trop importantes sur les poids de connexion qui peuvent engendrer une divergence (phase d'oubli) de l'apprentissage. En utilisant des paramètres  $\mu^l(k)$  et  $\hat{\sigma}^l(k)$  variables, l'algorithme RETRO\_KAL+ permet de corriger rapidement un début de divergence de l'apprentissage et ainsi de limiter l'amplitude des oscillations. Cependant deux points importants sont à prendre en compte dans l'utilisation de cette méthode: d'une part le dilemne entre l'erreur d'apprentissage obtenue après convergence et la présence ou non d'oscillations, en effet on ne peut pas avoir d'erreur minimale et une très bonne stabilité; d'autre part la détermination des constantes utilisées dans le calcul des variables  $\mu^l(k)$  et  $\hat{\sigma}^l(k)$  (équations (5.43) et (5.44)) qui peut être quelque peu fastidieuse.

## 5.4 Conclusion

Ce chapitre nous a permis de tester trois méthodes d'apprentissage de réseaux de neurones multicouches et d'étudier leurs qualités et inconvénients respectifs. Le choix de la tâche assignée au réseau de neurones en phase d'apprentissage a été fait en fonction de l'utilisation de celui-ci pour l'étalonnage et la reconstitution de mesurandes dans le cas du système de mesure de pression à fibres optiques. Les exemples d'apprentissage ont donc été générés par les équations des caractéristiques d'entrée/sortie des capteurs de pression, les entrées au nombre de trois sont les tensions de sortie normalisées des trois capteurs, la sortie est la pression reconstituée prétraitée. Compte tenu des résultats obtenus avec la méthode de la rétropropagation du gradient (RETRO\_GRAD) et la version modifiée par le filtre de Kalman (RETRO\_KAL) [SCA92], la première étant stable mais très lente et la deuxième étant très rapide mais instable, nous proposons une nouvelle méthode modifiée de RETRO\_KAL, appelée RETRO\_KAL+. La mise au point de cette méthode est le résultat de l'étude de l'influence des paramètres d'apprentissage  $\gamma^l$  et  $\mu^l$  qui ont un effet prépondérant sur les performances du réseau de neurones en phase d'apprentissage.

L'algorithme d'apprentissage RETRO\_KAL+ nous permet à ce niveau d'avancement du travail de disposer d'un réseau de neurones fonctionnel pour l'application envisagée au système de mesure de pression à fibres optiques. La phase d'apprentissage du réseau de neurones est réalisée très rapidement, moins de 400 rétropropagations, contrairement à la méthode RETRO\_GRAD qui nécessite un très grand nombre de rétropropagations, plus de 30000. Cette phase d'apprentissage est stable ou peut être stabilisée, contrairement à celle obtenue avec l'algorithme RETRO\_KAL, grâce à l'utilisation d'un pas de rétropropagation  $\mu^l(k)$  et d'une variance estimée  $\gamma^l(k)$  variables en fonction de l'erreur de propagation commise sur l'ensemble des données d'apprentissage  $\Delta\hat{y}(k)$ . Nous pouvons donc maintenant appliquer le réseau de neurones multicouche et cet algorithme d'apprentissage au système de mesure de pression, il va



falloir mettre l'accent sur les capacités de reconnaissance et de généralisation du réseau pour reconstituer les pressions, en limitant au maximum l'utilisation de la mesure de la température.

## **CHAPITRE 6**

# **RÉSULTATS D'ÉTALONNAGE ET DE RECONSTITUTION DE MESURANDES UTILISANT LES RESEAUX DE NEURONES**

### **6.1 Introduction**

Afin de vérifier l'applicabilité du réseau de neurones défini au chapitre 5 à l'étalonnage et à la reconstitution de mesurandes dans un système de mesure, nous allons maintenant présenter plusieurs résultats obtenus pour le système de mesure de pression à fibres optiques. Dans une première étape nous allons redéfinir de façon précise la tâche à effectuer par le réseau de neurones, c'est-à-dire l'étalonnage du système de mesure et la reconstitution de pressions, mais contrairement à ce qui a été exposé au chapitre 2, nous allons voir ces opérations dans le domaine des réseaux de neurones. Suite à cela, nous allons définir les signaux à traiter, soient les tensions de sortie des capteurs de pression, les pressions à reconstituer et l'influence de la température. Trois étapes d'évaluation seront réalisées, dans un premier temps avec des données synthétiques à température constante, dans un deuxième temps avec des données synthétiques à température variable et pour finir avec des données expérimentales. En fin de chapitre, nous verrons l'implantation matérielle de l'algorithme de reconstitution de mesurandes dans un microcontrôleur MC68HC11 [MOT90] et les difficultés rencontrées lors de cette implantation. Afin d'évaluer les différents résultats obtenus avec notre méthode, une comparaison est faite avec les résultats obtenus avec une méthode d'étalonnage et de reconstitution basée sur des approximations par les fonctions spline [LEM95], [BAR92].

## 6.2 Définition du problème

### 6.2.1 *Étalonnage et reconstitution de mesurande*

Le problème d'étalonnage et de reconstitution de mesurandes a été défini d'un point de vue analytique au chapitre 2, nous avons vu que la principale difficulté pour résoudre ces opérations est due au bruit de conversion inévitable. De plus les méthodes numériques de reconstitution de signaux sont dépendantes du modèle mathématique de la relation entre les signaux et l'information liée au système de mesure, tels que la fonction de transfert et des bruits qui le perturbent. La résolution est souvent sensible aux perturbations qui agissent sur le système de mesure. La méthode proposée dans ce travail n'est pas dépendante d'un modèle mathématique puisque le réseau de neurones utilisé agit comme une *boîte noire* dont seules les entrées et les sorties sont accessibles, l'algorithme d'apprentissage permet de fixer les poids de connexion du réseau pour répondre à l'application imposée au réseau.

La méthode d'étalonnage et de reconstitution de mesurandes à l'aide des réseaux de neurones se base sur les performances des réseaux de neurones en reconnaissance de formes. L'opération d'étalonnage du système correspond à la *phase d'apprentissage* du réseau et fait appel à un algorithme d'apprentissage performant (RETRO\_KAL+); la reconstitution de mesurandes correspond à une simple propagation d'un signal dans le réseau entraîné pour ce type de signal, cette opération fait appel à la capacité de *généralisation* du réseau. Cette capacité est définie comme l'aptitude à donner une réponse, bonne ou satisfaisante, à un signal qui n'a pas été appris, c'est donc la possibilité de généraliser ce qui a été appris en faisant face à une nouvelle situation. En exploitant cette capacité de généralisation de réseaux de neurones, une pression doit pouvoir être reconstituée à partir des trois tensions d'entrée, même si elles ne font pas partie des données d'apprentissage. C'est aussi cette capacité qui va permettre de tenir compte de

l'influence de la température sur les tensions de sortie des capteurs de pression et donc de reconstituer la pression sans avoir à mesurer la température.

Le cas du système de mesure de pression à fibres optiques a été présenté au chapitre 3, la principale difficulté est due à l'influence importante de la température qui crée un déphasage sur les caractéristiques d'entrée/sortie quasi-sinusoïdales des capteurs de pression. Dans les méthodes d'étalonnage et de reconstitution de mesurandes pour ce système de mesure proposées jusqu'à présent [BAR90] et [BAR92], l'effet de la température est pris en compte par l'ajout d'une quatrième tension qui vient en entrée du convertisseur, voir figure 3.3, la température est donc une donnée nécessaire pour réaliser l'étalonnage ou la reconstitution d'une pression. Dans notre méthode nous avons cherché à limiter au maximum le rôle de la température comme donnée d'entrée, en se basant sur l'hypothèse suivante: la caractéristique du capteur  $S_G$  étant monotone sur les deux plages de pression et de température, alors pour chaque pression  $\hat{p}$  la relation suivante est vérifiée:

$$\frac{\partial f_1(\hat{p}; T)}{\partial T} \neq \frac{\partial f_G(\hat{p}; T)}{\partial T} \quad \text{ou} \quad \frac{\partial f_2(\hat{p}; T)}{\partial T} \neq \frac{\partial f_G(\hat{p}; T)}{\partial T} \quad (6.1)$$

où  $f_1(\hat{p}; T)$ ,  $f_2(\hat{p}; T)$  et  $f_G(\hat{p}; T)$  sont respectivement les caractéristiques des capteurs  $S_1$ ,  $S_2$  et  $S_G$ . Ce qui équivaut au fait qu'il y a une et une seule paire  $\langle \hat{v}_1, \hat{v}_G \rangle$  ou  $\langle \hat{v}_2, \hat{v}_G \rangle$  correspondant à une pression donnée  $\hat{p}$ . Le symbole  $\hat{v}_i$  correspond à la valeur de la tension  $v_i$  ( $i = 1, 2$  ou  $G$ ) pour une pression  $p$  mesurée. Sous ces hypothèses, l'effet de la température peut être appris et mémorisé durant la phase d'apprentissage du réseau puisqu'il est contenu dans les données d'apprentissage, et nous supposons que le réseau est capable d'extraire de lui-même l'information sur la température par la suite. La phase d'apprentissage correspond à l'étalonnage du système de mesure et la phase de reconnaissance correspond à la reconstitution de mesurande, donc si l'effet de

la température est contenue dans les trois tensions  $v_1$ ,  $v_2$  et  $v_G$  alors la pression peut être reconstituée sans avoir à mesurer la température.

### 6.2.2 Génération des signaux synthétiques

Pour tester la validité de la méthode d'étalonnage et de reconstitution de mesurandes, deux ensembles de données sont à générer: les données d'apprentissage, notées avec l'exposant *etal*, et les données de généralisation, notées avec l'exposant *eval*. Dans les deux cas ces données sont constituées des paires tensions/pression  $(x_{n,m}^0, y_{n,m})$ , où  $x_{n,m}^0 = [x_{1,n,m}^0, x_{2,n,m}^0, x_{3,n,m}^0]^T$  est le vecteur constitué par les trois tensions de sortie des capteurs de pression prétraitées (voir équations (5.5) et (5.6)) et  $y_{n,m}$  est la pression correspondante prétraitée (voir équation (5.7)).

Les données d'étalonnage (ou d'apprentissage) sont générées de la façon suivante: on suppose que pour chaque valeur de température parmi les  $N_T$  différentes valeurs de température

$$T_{\min} \approx T_1^{etal} < T_2^{etal} < \dots < T_{N_T}^{etal} \approx T_{\max} \quad (6.2)$$

il y a  $N_p$  paires tensions/pression mesurées telle que

$$\left\{ (v_{1,1,m}^{etal}, v_{2,1,m}^{etal}, v_{G,1,m}^{etal}, p_{1,m}^{etal}), \dots, (v_{1,N_p,m}^{etal}, v_{2,N_p,m}^{etal}, v_{G,N_p,m}^{etal}, p_{N_p,m}^{etal}) \mid m = 1, \dots, N_T \right\} \quad (6.3)$$

où

$$P_{\min} \approx p_{1,m}^{etal} < p_{2,m}^{etal} < \dots < p_{N_p,m}^{etal} \approx P_{\max} \quad (6.4)$$

Afin de tenir compte des erreurs de mesure lors de l'étalonnage et des bruits de perturbation, les données exactes générées précédemment

$$\left\{ \tilde{T}_m^{etal}, \left\{ \tilde{v}_{1,n,m}^{etal}, \tilde{v}_{2,n,m}^{etal}, \tilde{v}_{G,n,m}^{etal}, \tilde{p}_{n,m}^{etal} \mid n = 1, \dots, N_P \right\} \mid m = 1, \dots, N_T \right\} \quad (6.5)$$

sont bruitées additivement à l'aide de nombre pseudo-aléatoires  $\{ \Delta T_m \}$ ,  $\{ \Delta p_{n,m} \}$ ,  $\{ \Delta v_{1,n,m} \}$ ,  $\{ \Delta v_{2,n,m} \}$  et  $\{ \Delta v_{G,n,m} \}$ , uniformément distribués autour de zéro. On introduit l'effet des erreurs de mesure dans l'étalonnage de la façon suivante:

$$\tilde{T}_m^{etal} = \dot{T}_m^{etal} + \Delta T_m \quad |\Delta T_m| \leq \overline{\Delta T}^{etal} \quad (6.6a)$$

$$\tilde{p}_{n,m}^{etal} = \dot{p}_{n,m}^{etal} + \Delta p_{n,m} \quad |\Delta p_{n,m}| \leq \overline{\Delta p}^{etal} \quad (6.6b)$$

$$\tilde{v}_{1,n,m}^{etal} = \dot{v}_{1,n,m}^{etal} + \Delta v_{1,n,m} \quad |\Delta v_{1,n,m}| \leq \overline{\Delta v}^{etal} \quad (6.6c)$$

$$\tilde{v}_{2,n,m}^{etal} = \dot{v}_{2,n,m}^{etal} + \Delta v_{2,n,m} \quad |\Delta v_{2,n,m}| \leq \overline{\Delta v}^{etal} \quad (6.6d)$$

$$\tilde{v}_{G,n,m}^{etal} = \dot{v}_{G,n,m}^{etal} + \Delta v_{G,n,m} \quad |\Delta v_{G,n,m}| \leq \overline{\Delta v}^{etal} \quad (6.6e)$$

où  $\overline{\Delta T}^{etal}$ ,  $\overline{\Delta p}^{etal}$ , et  $\overline{\Delta v}^{etal}$  sont respectivement les erreurs absolues limites de la mesure de la température, de la pression et de la tension.

L'erreur absolue limite de la mesure de la température  $\overline{\Delta T}^{etal}$  est sans effet sur l'étalonnage pour la méthode d'étalonnage proposée car la mesure de la température n'intervient pas dans l'algorithme d'étalonnage défini à la section 5.3.3. Cependant, pour vérifier l'effet des données exactes de référence de la température sur la qualité de l'étalonnage, ces données sont bruitées additivement à l'aide de nombres pseudo-aléatoires  $\{\Delta \tilde{T}_m\}$ , uniformément distribués autour de zéro

$$\dot{T}_m^{etal} = \bar{T}_m^{etal} + \Delta \bar{T}_m \quad |\Delta \bar{T}_m| \leq \bar{\Delta} \bar{T}_m^{etal} \quad (6.7)$$

où  $\bar{T}_m^{etal}$  est la donnée exacte de référence de la température gardée constante pour  $n = 1, 2, \dots, N_p$  et  $\bar{\Delta} \bar{T}_m^{etal}$  est l'erreur absolue limite sur la donnée exacte de référence de la température.

Les données d'évaluation de la capacité de généralisation correspondent à un autre ensemble de données générées

$$\left\{ \bar{T}_m^{eval}, \left\{ v_{1,n,m}^{eval}, v_{2,n,m}^{eval}, v_{G,n,m}^{eval}, p_{n,m}^{eval} \mid n = 1, \dots, 3N_p-3 \right\} \mid m = 1, \dots, 3N_T-3 \right\} \quad (6.8)$$

où les valeurs de pression et de température sont respectivement choisies approximativement égales à

$$\begin{aligned} &0.75 p_{n,m}^{etal} + 0.25 p_{n+1,m}^{etal} ; \quad 0.5 p_{n,m}^{etal} + 0.5 p_{n+1,m}^{etal} ; \quad 0.25 p_{n,m}^{etal} + 0.75 p_{n+1,m}^{etal} , \\ &0.75 \bar{T}_m^{etal} + 0.25 \bar{T}_{m+1}^{etal} ; \quad 0.5 \bar{T}_m^{etal} + 0.5 \bar{T}_{m+1}^{etal} ; \quad 0.25 \bar{T}_m^{etal} + 0.75 \bar{T}_{m+1}^{etal} . \end{aligned} \quad (6.9)$$

La figure 6.1 présente un ensemble de données synthétiques, seuls les triplets d'entrée  $\langle x_{1,n,m}^0, x_{2,n,m}^0, x_{3,n,m}^0 \rangle$  sont présentés pour  $N_T = 11$  et  $N_p = 30$ .

### 6.2.3 Réalisation des tests

Différents cas de tests sont réalisés afin d'évaluer les performances du réseau de neurones pour l'application à l'étalonnage et à la reconstitution de pression dans le système de mesure de pression à fibres optiques. Tous les tests sont effectués avec le réseau de neurones à trois couches présenté à la figure 5.2, l'algorithme d'apprentissage utilisé pour la modification des poids de connexion est l'algorithme RETRO\_KAL+ proposé à la section 5.3.3. Compte tenu des simplifications qui ont été faites sur les équations du pas

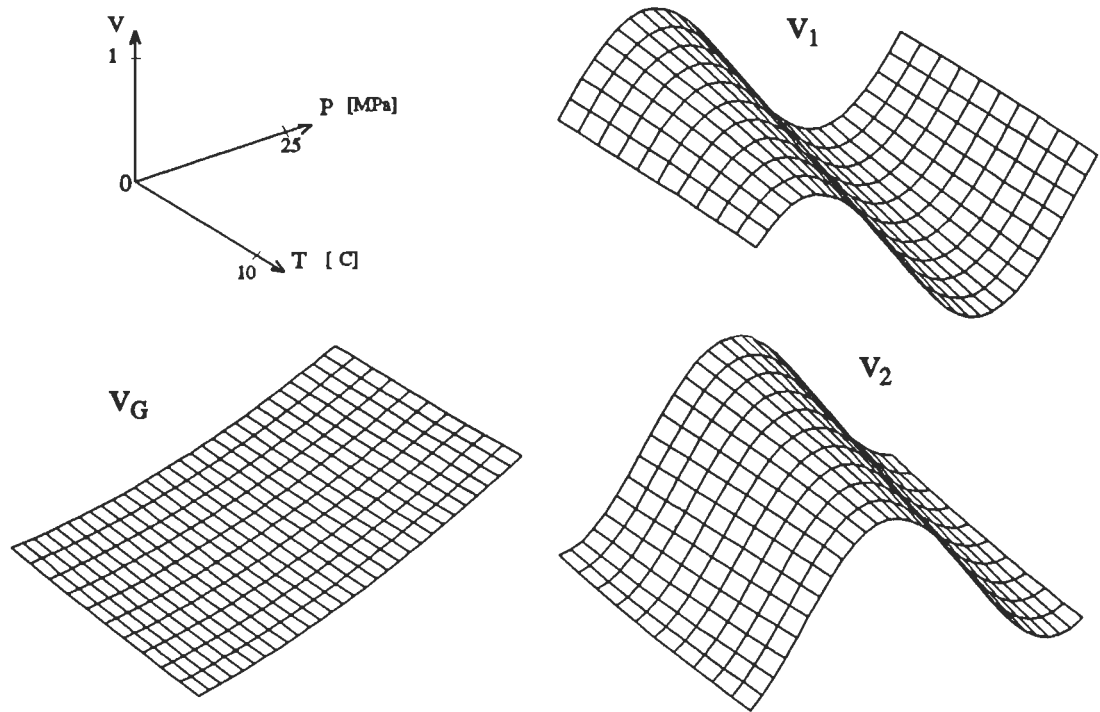


Figure 6.1 Données synthétiques  $T_{\min} = 0^{\circ}\text{C}$ ,  $T_{\max} = 10^{\circ}\text{C}$ ,  $p_{\min} = 0\text{ MPa}$ ,  $p_{\max} = 25\text{ MPa}$ .

de rétropropagation  $\mu^l$  (équation (5.45)) et de la variance estimée  $\check{r}^l$  (équation (5.46)), nous avons adopté les notations des paramètres suivantes:

$$\mu^l(k) = 1 - e^{-\frac{\Delta\hat{y}(k)}{\mu_c^l}} \quad \text{pour } l = 2 \quad (6.10)$$

$$\check{r}^l(k) = \check{r}_a^l + (1 - \check{r}_a^l) e^{-\frac{\Delta\hat{y}(k)}{\check{r}_c^l}} \quad \text{pour } l = 2,3 \quad (6.11)$$

Quatre parties principales vont être développées, chacune ayant son importance pour tester certains points de fonctionnement du réseau:

1. *Tests avec des données synthétiques et à température fixe:* ils ont pour objectif d'étudier le comportement du réseau, apprentissage et capacité de généralisation, en fonction du nombre d'exemples d'apprentissage, du nombre de neurones sur la



couche cachée et des paramètres régissant le pas de rétropropagation  $\mu^l(k)$  et la variance estimée  $\hat{\sigma}^l(k)$ .

2. *Tests avec des données synthétiques et à température variable:* ils constituent les principaux tests de mise au point de la méthode d'étalonnage et de reconstitution de mesurandes pour le système de mesure à l'étude. L'accent est mis sur l'influence de la température en phase d'étalonnage et sur la qualité des reconstitutions de pression. Une comparaison avec les résultats obtenus avec la méthode d'approximation basée sur les fonctions spline [BAR92] est proposée pour valider les résultats obtenus.
3. *Tests avec des données expérimentales:* ils sont réalisés avec les données expérimentales disponibles, ils permettent de valider la méthode dans un cas réel.
4. *Implantation matérielle:* seule les poids de connexion du réseau de neurones entraîné sont implantés, permettant ainsi de reconstituer des pressions à partir de tensions d'entrée. L'implantation est faite dans un micro-contrôleur MC68HC11, cette étape nous permet de vérifier la précision de reconstitution des pressions, le temps de traitement des données et l'espace mémoire nécessaire, une comparaison avec la méthode basée sur les fonctions spline est proposée.

L'évaluation des performances de réseau se font d'une part sur la phase d'apprentissage, selon la rapidité de convergence, la stabilité et la valeur minimale d'erreur atteinte; d'autre part sur la phase d'évaluation de la capacité de généralisation en fonction de l'erreur relative limite de propagation pour chaque pression  $\hat{p}^{eval}$  déterminée par

$$\bar{\delta}\hat{p}_n = \sup_m \left\{ \frac{\overline{\Delta}\hat{p}_{n,m}}{\hat{p}_{n,m}^{eval}} \mid m = 1, \dots, 3N_T-3 \right\} \quad (6.12)$$

où  $\hat{p}_{n,m}^{eval}$  est la pression désirée correspondant à la propagation du triplet  $\langle \hat{v}_{1,n,m}^{eval}, \hat{v}_{2,n,m}^{eval}, \hat{v}_{G,n,m}^{eval} \rangle$  et  $\overline{\Delta}\hat{p}_{n,m}$  est l'erreur absolue limite définie par

$$\overline{\Delta}\hat{p}_{n,m} = |\hat{p}_{n,m}^{eval} - \hat{p}_{n,m}| \quad (6.13)$$

où  $\hat{p}_{n,m}$  est la pression reconstituée,

Les trois premières parties sont réalisées à l'aide du logiciel mathématique MATLAB [MAT89] qui est performant pour le calcul matriciel, les réseaux de neurones font appel principalement à ce type de calcul, les programmes sont présentés à l'annexe C.

### 6.3 Résultats d'étalonnage et de reconstitution de pression: données synthétiques

#### 6.3.1 Résultats à température fixe

Le but de cette section est d'étudier le comportement du réseau dans différents cas de tests. À partir des résultats obtenus nous allons pouvoir mieux comprendre le fonctionnement du réseau pour nous guider dans la mise au point du réseau dans le cas où la température varie sur une plage. Les signaux d'apprentissage et de tests de capacité de généralisation sont générés tel que précisé à la section 6.2.2 avec:

$$\begin{aligned} N_T &= 1 & \text{avec} & & T &= 15^\circ C \\ N_P &= 30 & \text{avec} & & p_{\min} &= 1 \text{ MPa}, p_{\max} = 26 \text{ MPa} \end{aligned}$$

ces signaux sont non bruités et donc  $\overline{\Delta}T^{etal} = 0$ ,  $\overline{\Delta}p^{etal} = 0$ ,  $\overline{\Delta}v^{etal} = 0$  et  $\overline{\Delta}\bar{T}^{etal} = 0$ ; le nombre de neurones sur la couche cachée est  $N^2 = 50$  sauf pour les tests

où l'influence de ce paramètre est étudiée. Dans tous les cas d'apprentissage, l'arrêt du processus d'apprentissage est donnée par une erreur limite  $\Delta \hat{y}(k) = 10^{-4}$  ou par un nombre maximum de rétropropagations si l'erreur limite n'est pas atteinte.

### *Influence du nombre de neurones sur la couche cachée $N^2$*

Le premier groupe de simulations consiste en l'apprentissage des données définies précédemment avec un nombre variable de neurones sur la couche cachée du réseau. Ces tests ont pour but d'étudier le comportement du réseau lorsque le nombre de poids de connexion est variable, ce qui revient à une modification de la répartition des informations emmagasinées par le réseau. La figure 6.2(a) illustre le comportement du réseau en phase d'apprentissage pour différents cas, soient  $N^2 = 15$ ,  $N^2 = 20$ ,  $N^2 = 40$ ,  $N^2 = 50$  et  $N^2 = 60$ . Nous pouvons noter qu'un nombre élevé de neurones sur la couche cachée facilite l'apprentissage du réseau, celui-ci semble avoir plus de facilité à fixer les poids de connexion si ces derniers sont nombreux. Ce résultat s'explique par une répartition plus étalée des informations parmi les poids de connexion qui jouent le rôle de mémoires. La capacité de généralisation du réseau à des exemples propagés non appris,

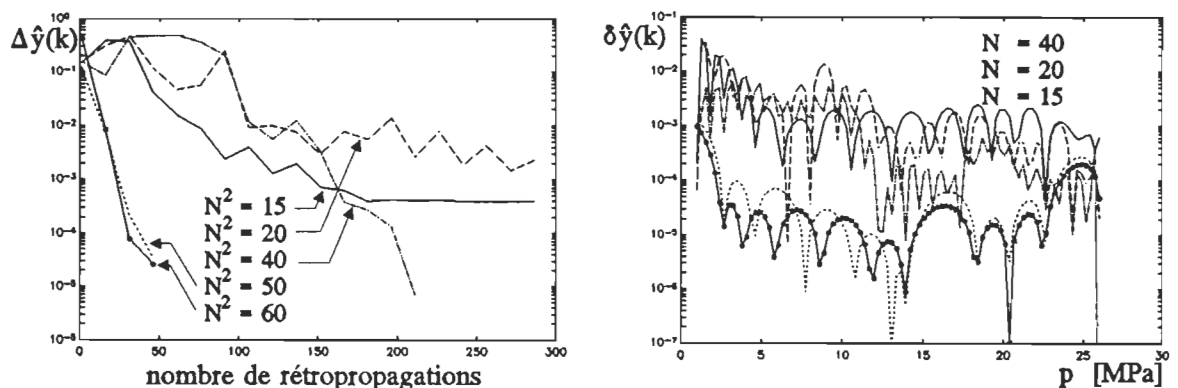


Figure 6.2 Courbes d'apprentissage (a) et de généralisation (b) pour un nombre variable de neurones sur la couche cachée,  $\mu_c = 0.1$ ,  $\check{r}_a = 0.2$  et  $\check{r}_c = 0.01$ .

présentée à la figure 6.2(b), est meilleure dans le cas où le nombre de neurones sur la couche cachée est grand ( $N^2 \geq 50$ ), mais les résultats sont comparables pour  $N^2 = 50$  et  $N^2 = 60$ . Il est intéressant de noter que dans les cas où  $N^2 \leq 40$  les courbes d'apprentissage sont similaires en nombre d'itérations et valeur minimale d'erreur d'apprentissage, ce qui incite à penser que les informations emmagasinées par le réseau peuvent se répartir sur un nombre limité de connexions.

### *Influence du nombre d'exemples d'apprentissage $N_T \cdot N_p$*

Le deuxième groupe de simulations consiste en l'apprentissage d'un réseau de neurones avec un nombre variable d'exemples d'apprentissage, soient  $N_p = 10$ ,  $N_p = 20$ ,  $N_p = 30$  et  $N_p = 40$  avec  $N_T = 1$ . L'objectif de ces tests est d'étudier la qualité de la généralisation avec des données non apprises, dans le cas d'apprentissage avec différents nombres de données d'apprentissage, ceci devant nous permettre d'estimer le nombre d'exemples nécessaires pour obtenir un réseau bien entraîné. Les résultats d'apprentissage sont présentés à la figure 6.3(a), l'apprentissage est plus rapide si le nombre d'exemples est grand, mais reste comparable pour  $N_p = 30$  et

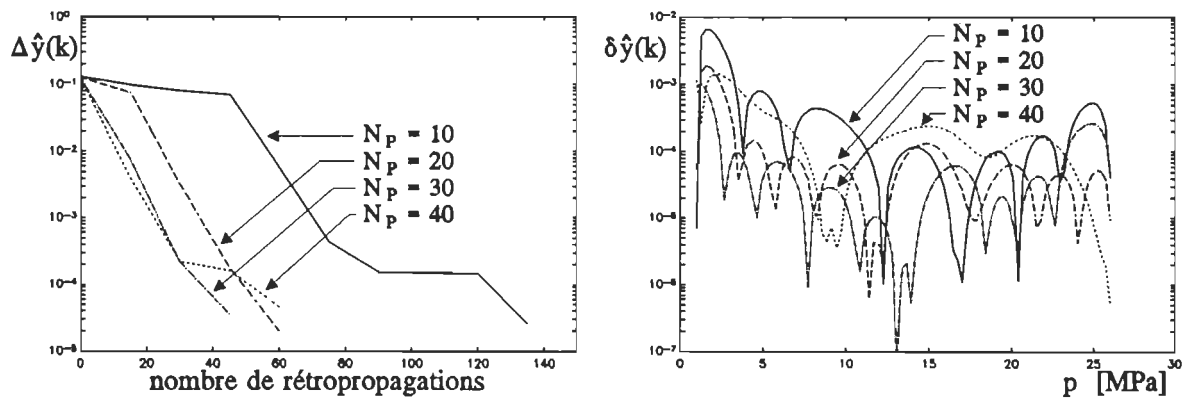


Figure 6.3 Courbes d'apprentissage (a) et de généralisation (b) pour un nombre variable d'exemples d'apprentissage,  $\mu_e = 0.1$ ,  $\tilde{r}_a = 0.2$ ,  $\tilde{r}_e = 0.01$ .

$N_p = 40$  , ce qui indique une bonne répartition des informations dans le réseau et montre la capacité de généralisation du réseau lors de l'apprentissage d'exemples voisins. Les meilleures conditions de généralisation sont obtenues pour  $N_p = 20$  et  $N_p = 30$  , voir figure 6.3(b), ce qui tend à montrer qu'un nombre trop élevé d'exemples d'apprentissage peut nuire à la généralisation. Il faut noter que dans le cas où  $N_p = 40$  , l'erreur minimale d'apprentissage est très faible et atteinte en très peu d'itérations, ce qui pourrait indiquer que la réseau à "trop" bien appris ses exemples d'apprentissage et n'est pas capable de généraliser ce qu'il a appris.

### *Influence du pas de rétropropagation $\mu^l(k)$*

Le troisième groupe de simulations consiste en l'étude de l'influence des paramètres des équations du pas de rétropropagation  $\mu^l(k)$  et de la variance estimée  $\check{\gamma}^l(k)$  (équations (6.10) et (6.11)), soient  $\mu_e$  ,  $\check{r}_a$  et  $\check{r}_e$  . La figure 6.4 présente les résultats d'apprentissage et de généralisation obtenus pour différentes valeurs de  $\mu_e$  , si  $\mu_e$  augmente on limite l'effet de correction apporté par l'erreur absolue moyenne  $\Delta\hat{y}(k)$  et donc le pas de rétropropagation  $\mu^2(k)$  augmente, l'apprentissage est plus rapide.

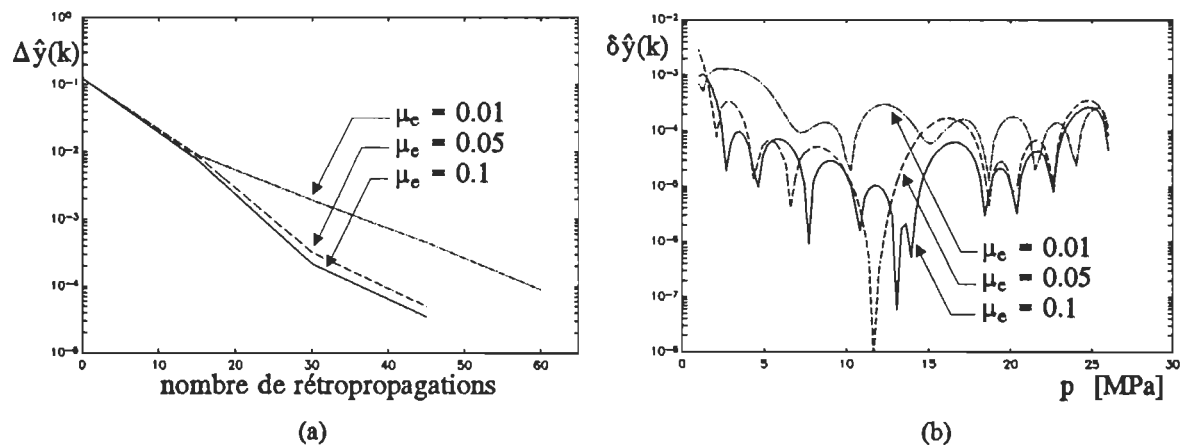


Figure 6.4 Courbes d'apprentissage (a) et de généralisation (b) pour différentes valeurs de  $\mu_e$ ,  $\check{r}_a = 0.2$  et  $\check{r}_e = 0.01$ .

Cependant il ne faut pas négliger le fait qu'un apprentissage trop rapide peut entraîner des oscillations (voir section 5.3). La généralisation est meilleure pour une valeur plus élevée de  $\mu_e$  car l'erreur minimale d'apprentissage atteinte est plus faible.

### *Influence de la variance estimée $\check{\gamma}^l(k)$*

La figure 6.5 présente les résultats d'apprentissage et de généralisation obtenus pour différentes valeurs du paramètre  $\check{\gamma}_a$  qui détermine les limites de variations de la variance estimée  $\check{\gamma}^l(k)$ . Ce paramètre joue aussi sur la rapidité de convergence du réseau en phase d'apprentissage, l'apprentissage sera d'autant plus rapide que le paramètre  $\check{\gamma}_a$  sera faible puisque la variance estimée agit comme diviseur dans le calcul du gain de Kalman (voir section 5.3) et donc agit comme multiplicateur sur le terme de correction des poids de connexion. Les courbes de généralisation sont comparables pour les trois valeurs de  $\check{\gamma}_a$ .

L'influence du paramètre  $\check{\gamma}_e$  est plus marquée puis qu'il agit directement sur la correction apportée par l'erreur absolue moyenne  $\Delta\hat{y}(k)$  sur la variance estimée  $\check{\gamma}^l(k)$ .

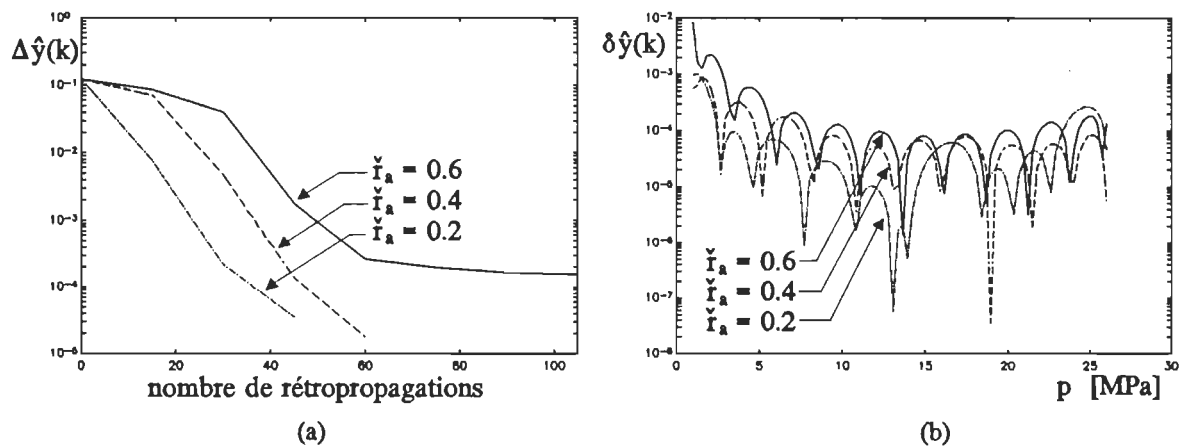


Figure 6.5 Courbes d'apprentissage (a) et de généralisation (b) pour différentes valeurs de  $\check{\gamma}_a$ ,  $\mu_e = 0.1$  et  $\check{\gamma}_e = 0.01$ .

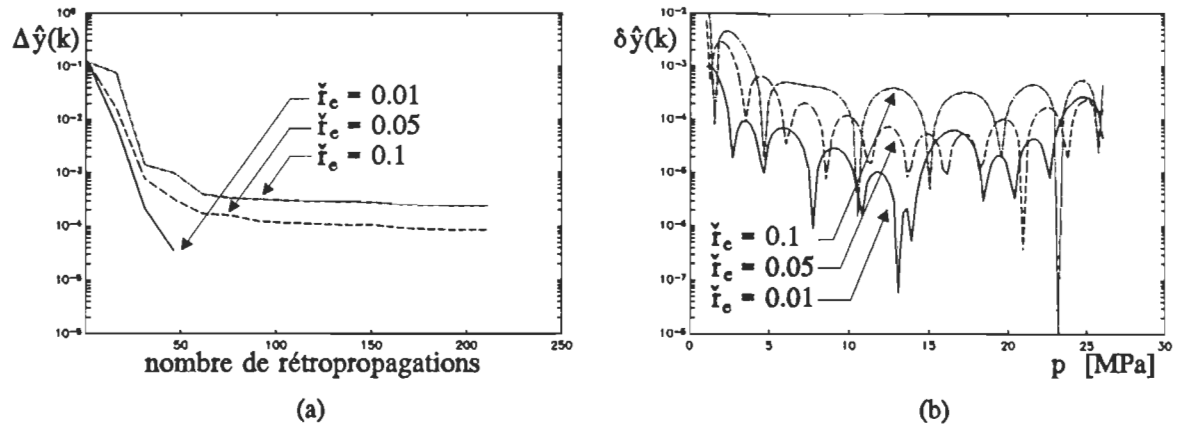


Figure 6.6 Courbes d'apprentissage (a) et de généralisation (b) pour différentes valeurs de  $\check{\gamma}_e$ ,  $\mu_c = 0.1$  et  $\check{\gamma}_a = 0.2$ .

Si  $\check{\gamma}_e$  diminue alors la variance estimée diminue, le gain de Kalman augmente et les corrections des poids sont plus importantes, la convergence du réseau est donc plus rapide. Cependant le risque d'oscillation reste présent si les corrections sur les poids sont trop fortes et donc le terme  $\check{\gamma}_e$  doit être bien choisi. Les courbes d'apprentissage et de généralisation pour différentes valeurs de  $\check{\gamma}_e$  sont présentées à la figure 6.6.

Les résultats d'apprentissage et de généralisation obtenus pour des données d'apprentissage synthétiques et à température fixe nous permettent de valider la méthode d'étalonnage et de reconstitution de pression pour le système de mesure de pression à fibres optiques. L'étalonnage s'effectue en un nombre faible d'itérations, moins de 60 rétropropagations, et les erreurs relatives de reconstitution de pression pour des données d'entrée non apprises sont faibles. Les meilleurs résultats, en termes de stabilité de la courbe d'apprentissage, de la valeur minimale de saturation atteinte et de qualité de généralisation (erreurs relatives de reconstitution) sont obtenus pour les conditions suivantes:

- nombre de neurones sur la couche cachée:  $N^2 = 50$  ,
- nombre d'exemples d'apprentissage:  $N_p = 30$  ,

- pas de rétropropagation:  $\mu_e = 0.1$  ,
- variance estimée:  $\check{r}_a = 0.2$  et  $\check{r}_e = 0.01$  .

L'erreur relative maximale commise sur les pressions reconstituées est alors de l'ordre de  $10^{-4}$  si on exclut les extrémités de la plage des pressions.

### 6.3.2 Résultats à température variable

Dans la section précédente, nous avons montré la validité de la méthode d'étalonnage et de reconstitution de pression dans le cas d'une température maintenue constante. Cependant cette situation est peu réaliste du fait de l'influence de la température sur les caractéristiques des capteurs de pression (voir chapitre 3), à moins que l'effet de la température ne soit compensé d'une autre façon. Nous allons maintenant nous intéresser à l'étalonnage et la reconstitution de pression dans le cas où la température varie sur une plage, l'objectif des tests suivants est de démontrer la possibilité d'étendre les capacités de généralisation du réseau de neurones à la température, c'est-à-dire que l'effet de la température sur les capteurs de pression va être appris par le réseau. Dans ce cas il est possible de reconstituer la pression sans mesurer la température, de plus lors de l'étalonnage du système la température n'est pas une donnée d'entrée, il suffit simplement de s'assurer que les données d'apprentissage ont bien été prises sur la plage de température considérée.

Les données synthétiques d'apprentissage permettant d'étalonner le système de mesure de pression et les données de généralisation permettant de tester les capacités de reconstitution de pression du réseau sont générées tel que décrit dans la section 6.2.2 avec

$$\begin{aligned} T_{\min} &= 0^{\circ}C, & T_{\max} &= 10^{\circ}C \\ p_{\min} &= 1\text{MPa}, & p_{\max} &= 26\text{MPa} \end{aligned}$$



Les signaux d'entrée sont bruités additivement à l'aide de nombres pseudo-aléatoires uniformément distribués autour de zéro  $\bar{\Delta}p^{etal} = 0.01 MPa$  ,  $\bar{\Delta}v^{etal} = 0.002 V$  et  $\bar{\Delta}T^{etal}$  prend successivement les valeurs  $0.001^\circ C$  ,  $0.01^\circ C$  ,  $1.0^\circ C$  et  $2.0^\circ C$  . Pour toutes les valeurs  $\bar{p}_{n,m}^{etal}$  correspondant aux triplets  $\langle \bar{v}_{1,n,m}^{etal}, \bar{v}_{2,n,m}^{etal}, \bar{v}_{G,n,m}^{etal} \rangle$ , les poids du réseau de neurones sont calculés avec l'algorithme d'étalonnage défini à la section 5.3.3. Pour étudier l'effet de l'erreur absolue limite de la mesure de la température  $\bar{\Delta}T^{etal}$  qui n'est pas comprise dans les triplets, nous appliquons une erreur absolue limite sur les données exactes de référence de la température selon l'équation (6.8) avec  $\bar{\Delta}\bar{T}^{etal}$  qui prend successivement les valeurs  $0^\circ C$  ,  $0.1^\circ C$  ,  $1.0^\circ C$  et  $2.0^\circ C$  . Une erreur importante sur les données de référence de la température nous permet de démontrer que l'étalonnage du système de mesure est peu dépendant de la précision sur la mesure de la température. Le prétraitement des données d'entrée/sortie est tel que les données d'entrée sont dans la plage  $[-1,1]$  afin d'améliorer la qualité de l'apprentissage et la capacité de généralisation. L'équation de transformation est alors:

$$x_i^0 = \frac{2(v_i - v_{i \min})}{v_{i \max} - v_{i \min}} - 1 \quad \text{pour } i = 1, 2, G \quad (6.14)$$

Le nombre de neurones sur la couche est  $N^2 = 60$  , les paramètres définissant le pas de rétropropagation  $\mu^l(k)$  et la variance estimée  $\check{r}^l(k)$  sont:  $\mu_e = 0.1$  ,  $\check{r}_a = 0.6$  et  $\check{r}_e = 0.15$  . Différentes valeurs du nombre de température  $N_T$  utilisé pour la génération des données d'apprentissage, ainsi que différents nombres de pression  $N_p$  pour chaque température ont été utilisés pour tester les performances du réseau de neurones.

Chaque test d'apprentissage et de généralisation de données synthétiques bruitées, réalisé pour des valeurs particulières de  $N_T$  et  $N_P$ , est répété pour 40 réalisations de bruits différentes (équation (6.6)) et ce sont les erreurs maximales qui sont retenues (équation (6.12)). Les tests de performances du réseau sont effectués selon les critères suivants:

- *phase d'apprentissage*: étude de la convergence du réseau en phase d'apprentissage en terme de rapidité et de stabilité, les résultats sont présentés en fonction de l'erreur absolue moyenne de propagation de l'ensemble des exemples appris, soit  $\Delta\hat{y}(k)$  où  $k$  est le nombre de rétropropagations.
- *phase de généralisation*: étude de la capacité du réseau de neurones à généraliser les informations apprises à des données inconnues. C'est ce qui permet de reconstituer les pressions dans notre application en fonction des tensions de sortie des capteurs à fibre optique. Les résultats sont présentés en fonction de l'erreur relative limite commise sur la pression reconstituée  $\hat{p}_n$ , soit  $\bar{\delta}\hat{p}_n$  définie par l'équation (6.12).

### *Phase d'apprentissage $\Delta\hat{y}(k)$*

Dans tous les tests d'apprentissage de données synthétiques réalisés présentement, l'influence de la température n'a pas été prise en compte. Nous avons montré que le réseau de neurones est capable d'apprendre facilement et rapidement un ensemble de données générées à température fixe, soit  $N_T = 1$ . Nous allons maintenant envisager le cas où  $N_T > 1$  afin de vérifier l'hypothèse émise à la section 6.2.1 (équation (6.1)), ce qui revient à montrer que le réseau est capable d'apprendre l'effet de la température sur les données d'apprentissage, tensions prétraitées, même si les données d'apprentissage sont très semblables dans certains cas. Trois exemples de courbes d'apprentissage de données synthétiques à température variable, correspondant à différentes réalisations de bruits

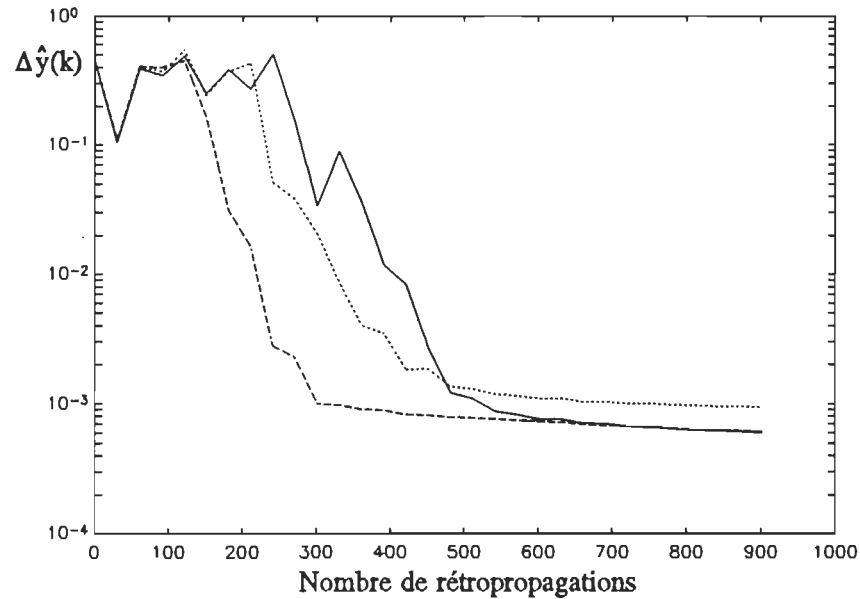


Figure 6.7 Courbes d'apprentissage de données synthétiques à température variable,  $N_T = 4$  et  $N_p = 15$ , correspondant à trois réalisations de bruit.

générés par le logiciel MATLAB avec  $\bar{\Delta p}^{etal} = 0.01 MPa$  ,  $\bar{\Delta v}^{etal} = 0.002 V$  , et  $\bar{\Delta T}^{etal} = 1.0^\circ C$  , sont présentés à la figure 6.7. Les données d'apprentissage sont générées pour

$$\begin{aligned} N_T &= 4 \\ N_p &= 15 \end{aligned}$$

À noter que la valeur de  $\bar{\Delta T}$  n'a pas d'importance puisqu'elle n'entre pas dans les calculs de l'algorithme d'apprentissage.

Le nombre d'exemples à apprendre est 60, ce qui est peu compte tenu de toutes les informations contenues dans ces exemples et que le réseau doit apprendre, on constate que la phase d'apprentissage est très rapide, moins de 400 rétropropagations et que les courbes de convergence  $\Delta\hat{y}(k)$  sont stables. Il est à noter que le nombre de rétropropagations nécessaires pour atteindre la convergence du réseau est variable selon la réalisation de

bruit, mais la valeur limite de l'erreur absolue moyenne d'apprentissage est toujours de l'ordre  $10^{-3}$ .

De nombreux tests d'apprentissage ont été effectués en variant les différents paramètres du réseau de neurones, le nombre d'exemples d'apprentissage, le niveau de bruit sur les exemples..., il serait fastidieux de présenter tous ces résultats. C'est pourquoi nous avons choisi de ne présenter que les résultats les plus significatifs et bien sûr les meilleurs. Dans la majorité des simulations réalisées il est toujours possible d'obtenir une convergence du réseau de neurones et on ne peut donc pas considérer ce résultat comme validation de la méthode proposée, nous allons donc nous intéresser aux performances du réseau de neurones pour la reconstitution de mesurandes, ce qui revient à tester sa capacité de généralisation à des données non apprises. De cette manière nous allons pouvoir montrer la validité de la méthode d'étalonnage et de reconstitution de mesurandes dans le cas du système de mesure à l'étude, l'accent est mis en particulier sur l'effet de la température sur les capteurs de pression.

#### *Influence du nombre d'exemples d'apprentissage $N_T.N_P$*

Considérons les deux cas suivants  $N_T.N_P = 4 \times 15 = 60$  et  $N_T.N_P = 4 \times 20 = 80$ , pour lesquels le nombre de températures utilisées pour la génération des signaux varie. Les résultats de généralisation pour les données définies selon les équations (6.8) et (6.9) sont présentés respectivement aux figures 6.8 et 6.9. Dans les deux cas les apprentissages pour les quarante réalisations de bruit se font en un nombre faible d'itérations (inférieur à 1000) et la valeur minimale de l'erreur absolue moyenne  $\Delta\hat{y}(k)$  est de l'ordre de  $10^{-3}$ . Les quatre valeurs différentes de bruits sur la température sont considérées, les quatre courbes obtenues sont présentées sur les figures. L'étude de ces cas d'apprentissage et de généralisation nous permettent de conclure qu'un

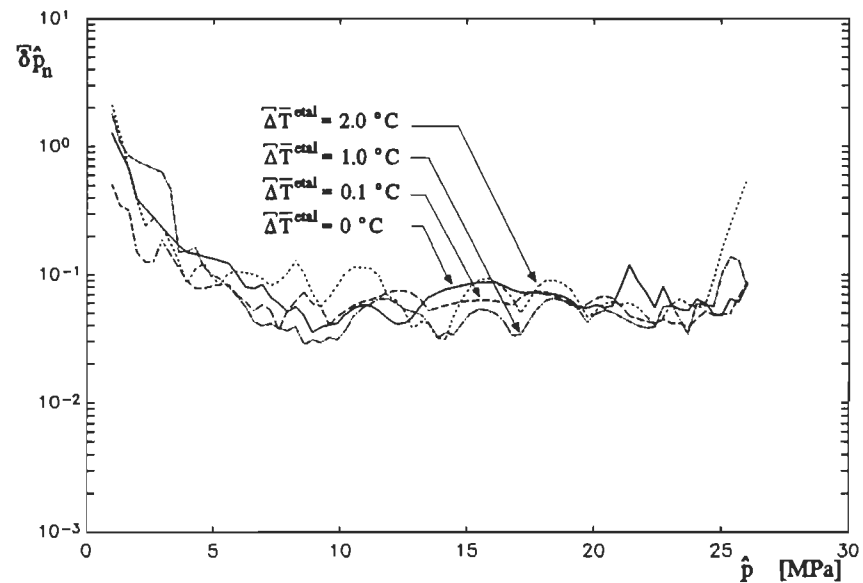


Figure 6.8 Variation de l'erreur relative limite  $\bar{\delta}_{\hat{p}_n}$  pour différents niveaux de bruit sur la température  $\bar{\Delta T}^{ctal}$ , données synthétiques avec  $N_T = 4$  et  $N_p = 15$ .

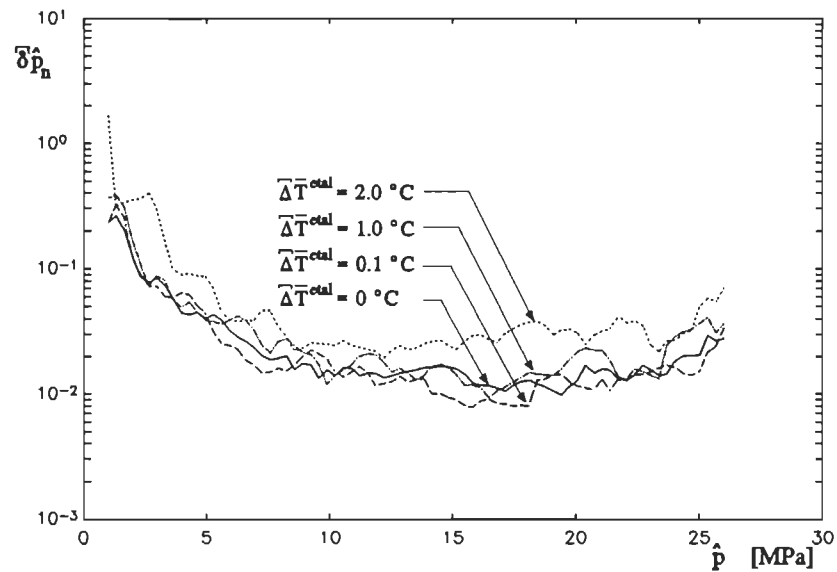


Figure 6.9 Variation de l'erreur relative limite  $\bar{\delta}_{\hat{p}_n}$  pour différents niveaux de bruit sur la température  $\bar{\Delta T}^{ctal}$ , données synthétiques avec  $N_T = 4$  et  $N_p = 20$ .

nombre plus élevé d'exemples d'apprentissage permet une meilleure qualité de généralisation, ce qui est logique, d'autant plus que le nombre de pression par température est plus grand, ce qui offre plus d'informations au réseau de neurones pour reconstituer des

pressions non apprises. Ce résultat rejoint ce qui a été vu dans la section 6.2.2 pour une température fixe.

### *Influence du nombre de valeurs de température $N_T$*

L'un des principaux objectifs de ce travail est de limiter le rôle de la température sur la méthode d'étalonnage et de reconstitution de pression pour le système de mesure de pression à fibres optiques qui présente l'inconvénient d'être très sensible à celle-ci. Dans ce deuxième groupe de simulations, nous avons considéré les cas suivants

$N_T \cdot N_p = 4 \times 20 = 80$  et  $N_T \cdot N_p = 2 \times 40 = 80$ , le nombre d'exemples est gardé constant, dans le premier cas les températures sont  $T^{etal} = 0^\circ C$ ,  $3.33^\circ C$ ,  $6.66^\circ C$  et  $10^\circ C$ , et dans le deuxième cas  $T^{etal} = 0^\circ C$  et  $10^\circ C$  ce qui correspond aux valeurs extrêmes non bruitées de la plage de température. Les quatre courbes d'apprentissage pour chaque cas sont présentées aux figures 6.9 et 6.10. Les meilleures reconstitution sont obtenues pour  $N_T = 4$  car l'information sur la température est plus

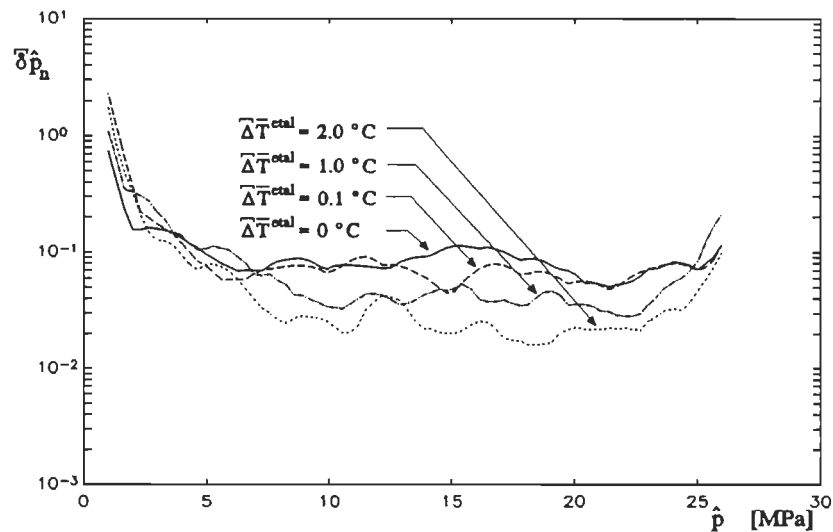


Figure 6.10 Variation de l'erreur relative limite  $\hat{\delta p}_n$  pour différents niveaux de bruits sur la température  $\Delta T^{etal}$ , données synthétiques avec  $N_T = 2$  et  $N_p = 40$ .

importante et donc plus facile à apprendre par le réseau de neurones. Cependant il faut noter que les résultats dans le cas où la température prend les valeurs extrêmes de la plage de température sont très intéressants car les erreurs de reconstitution ne sont pas trop détériorées, le réseau est capable de donner des résultats satisfaisants pour des températures éloignées de celles utilisées pour l'apprentissage (cas de  $T^{eval} = 5^{\circ} C$  ). Il est important de noter que le réseau de neurones généralise bien ses informations apprises car, mises à part les extrémités de la plage de pression, les résultats de reconstitution sont relativement constants sur toute la plage.

#### *Influence du bruit sur la température $\bar{\Delta}T^{etal}$*

L'erreur absolue limite de la mesure de la température  $\bar{\Delta}T^{etal}$  est sans effet sur l'étalonnage pour la méthode d'étalonnage proposée car la mesure de la température n'intervient pas directement dans l'algorithme d'étalonnage défini à la section 5.3.3.

#### *Influence du bruit sur les données exactes de références de la température $\bar{\Delta}\bar{T}^{etal}$*

Pour chaque cas de simulations dépendement des valeurs de  $N_T$  et  $N_P$  , quatre niveaux de bruits sur les données exactes de référence de la température ont été testés  $\bar{\Delta}\bar{T}^{etal} = 0^{\circ} C$  ,  $0.01^{\circ} C$  ,  $1.0^{\circ} C$  et  $2.0^{\circ} C$  afin d'étudier l'effet de la précision connue sur la mesure de la température lors de la phase d'apprentissage ou d'étalonnage. Nous rappelons que pour la reconstitution de pression, la température n'est pas requise, sous condition d'être dans la plage d'étalonnage considérée. Les figures 6.8 à 6.10 nous permettent de mettre en évidence le peu de précision requis sur la température pour réaliser l'étalonnage du système (phase d'apprentissage), les résultats de généralisation sont comparables pour les quatre niveaux de bruits. On peut donc conclure que l'étalonnage du système peut être fait sans grande précision sur la température de mesure des tensions de sortie des capteurs de pression, jusqu'à  $\bar{\Delta}\bar{T}^{etal} = 2^{\circ} C$  . Ce résultat très intéressant,

il est en effet souvent difficile de maintenir constante une température lors de la collecte des tensions de sortie des capteurs et de la mesurer avec précision, s'explique par le fait qu'une erreur importante sur le niveau de bruit de température permet de mieux répartir l'information traduisant son effet parmi toutes les données d'apprentissage. Ceci explique pourquoi la qualité de reconstitution est meilleure pour un niveau de bruit sur la température élevé dans le cas où  $N_T = 2$ .

### *Comparaison avec la méthode basée sur les fonctions spline*

Dans le but de comparer les résultats d'étalonnage et de reconstitution de mesurandes obtenus avec le réseau de neurones, un groupe de simulations annexe a été fait avec une méthode basée sur les approximations avec les fonctions spline, voir figure 6.11. Les conditions de simulations sont identiques à celles utilisées avec la méthode basée sur les réseaux de neurones à l'exception faite que nous nous sommes limités uniquement au

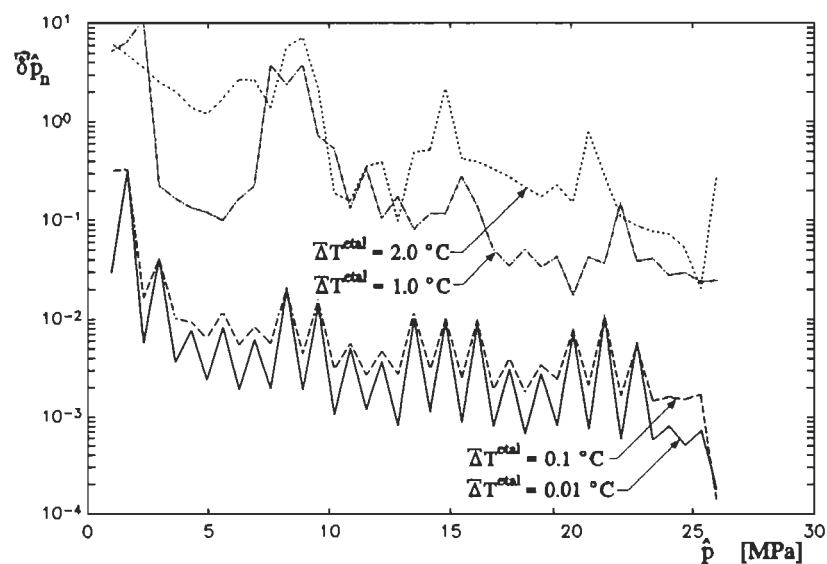


Figure 6.11 Variation de l'erreur relative limite  $\hat{\delta}_{\hat{p}_n}$  obtenue avec la méthode basée sur les fonctions spline, pour différents niveaux de bruit  $\Delta T^{etal}$ ,  $N_T = 4$  et  $N_p = 20$ .



cas où  $\bar{\Delta T}^{etal} = 0^\circ C$ . Les données d'apprentissage et de reconstitution sont générées pour  $N_T \cdot N_P = 4 \times 20 = 80$ ,  $N_T = 4$  étant le nombre minimal de température requis par l'algorithme d'étalonnage basé sur les fonctions spline [LEM95]. Pour comparer avec les figures 6.8 à 6.10 nous devons observer le résultat de l'erreur de reconstitution de la pression pour différentes valeurs d'erreur absolue limite de la mesure de la température  $\bar{\Delta T}^{etal}$  pour le cas où  $\bar{\Delta T}^{etal} = 0^\circ C$ .

Le point le plus important à retenir est que la mesure de la température est nécessaire aussi bien pour étalonner le système de mesure que reconstituer les pressions, de plus la méthode requiert une bonne précision sur la mesure de la température,  $\bar{\Delta T}^{etal} \leq 0.1^\circ C$ , ce qui est une contrainte importante. Dans le cas où la précision sur la température est bonne, la qualité de reconstitution des pressions est un peu meilleure que celle obtenue avec la méthode basée sur l'utilisation d'un réseau de neurones.

#### 6.4 Résultats d'étalonnage et de reconstitution de pression: données expérimentales

Les résultats d'étalonnage et de validation de la méthode avec des données expérimentales ont été obtenus pour des données mesurées sur les plages de température et de pression suivantes:

$$\begin{aligned} T_{\min} &= 15^\circ C, & T_{\max} &= 21^\circ C \\ p_{\min} &= 6 MPa, & p_{\max} &= 24 MPa \end{aligned}$$

Les données expérimentales disponibles, devant nous permettre d'étalonner le système de mesure de pression et de tester la validité de cet étalonnage en reconstituant des pressions non apprises, sont telles que:

$$\begin{aligned} N_T &= 7 \\ N_P &= 7 \end{aligned}$$

Il faut donc choisir parmi ces données celles qui serviront pour l'étalonnage et celles utilisées pour la reconstitution de pression. Nous avons choisi les températures suivantes pour les données d'apprentissage du réseau de neurones (étalonnage du système de mesure)

$$T^{etal} = 15^{\circ}C , 17^{\circ}C , 19^{\circ}C \text{ et } 21^{\circ}C .$$

En tenant compte du nombre limité de pressions mesurées pour chaque température, nous avons choisi de réaliser l'étalonnage du système avec des données synthétiques générées à partir des données réelles, aux températures choisies, dans le but de conserver le maximum de données expérimentales pour la validation de la méthode en reconstitution de mesurandes (capacité de généralisation). Ces données d'apprentissage ont été obtenues en utilisant la fonction d'interpolation spline proposée par le logiciel MATLAB et sont telles que

$$\begin{aligned} N_T &= 4 \\ N_p &= 15 \end{aligned}$$

un exemple des données ainsi obtenues est présenté à la figure 6.12 pour une des températures choisie pour l'étalonnage du système  $T^{etal} = 15^{\circ}C$  .

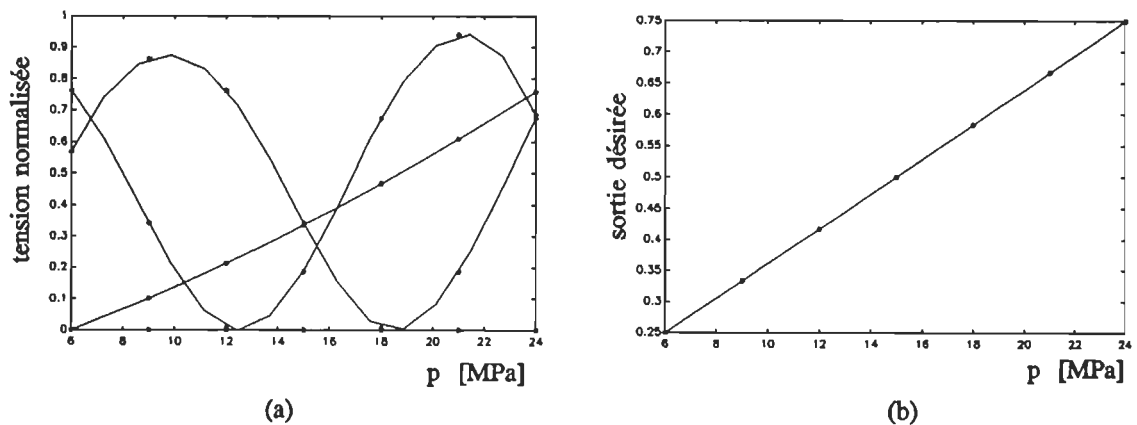


Figure 6.12 Données d'apprentissage d'entrée (a) et de sortie (b),  $T^{etal} = 15^{\circ}C$ ,  $p_{min} = 6$  MPa et  $p_{max} = 24$  MPa, interpolations avec des fonctions spline (—), expérimentales (\*).

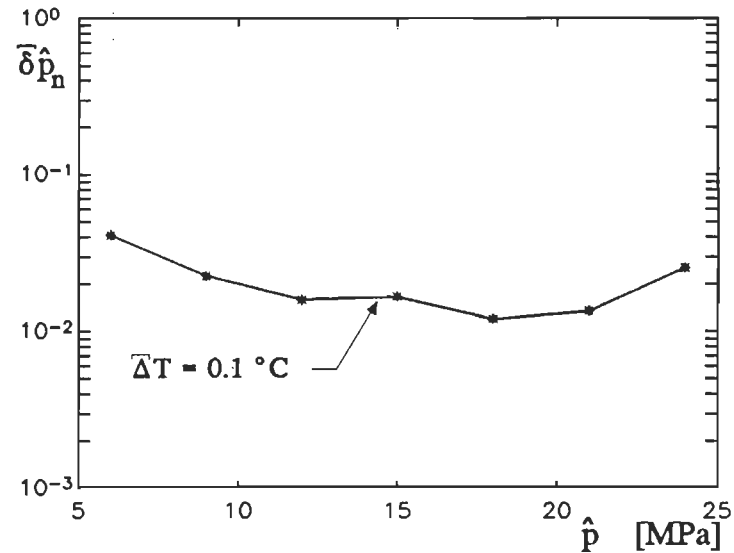


Figure 6.13 Variation de l'erreur relative limite  $\bar{\delta}_{\hat{p}_n}$  pour des données expérimentales avec  $N_T = 4$  et  $N_p = 15$ .

La phase d'étalonnage effectuée, c'est-à-dire quand le réseau de neurones a appris l'ensemble des données d'apprentissage, la capacité de généralisation du réseau est testée sur l'ensemble des données expérimentales. Les résultats de reconstitution de mesurandes sont présentés à la figure 6.13, l'erreur relative limite  $\bar{\delta}_{\hat{p}_n}$  obtenue est du même ordre que dans le cas des données synthétiques.

## 6.5 Implantation matérielle

L'implantation matérielle du programme de propagation, c'est-à-dire de la partie reconstitution de mesurandes de la méthode proposée, a été réalisée dans le microcontrôleur MC68HC11 de Motorola. Le but de cette implantation est de vérifier la précision obtenue sur une pression reconstituée et de mesurer le temps de calcul et l'espace mémoire requis. Le microcontrôleur supervise les opérations d'entrées/sorties telles que les données venant d'un ordinateur hôte et/ou d'un convertisseur A/N, les données d'affichage à un écran, ainsi que tous les traitements liés aux calculs de reconstitution de la pression. Son

horloge interne a une fréquence de 2 MHz, les trois tensions d'entrée  $\tilde{v}_1$ ,  $\tilde{v}_2$  et  $\tilde{v}_G$  sont fournies par un convertisseur A/N de tensions multiplexées avec une résolution de 12 bits. L'implantation logicielle a été faite en point flottant afin d'obtenir une précision satisfaisante pour la reconstitution.

L'implantation matérielle du programme de reconstitution de pression fait appel aux poids de connexions  $w_{ij}$  du réseau de neurones, calculés lors d'une phase ultérieure d'apprentissage à l'aide de l'algorithme RETRO\_KAL+. Le programme implanté a été écrit en langage C (voir annexe D, fichier NEU\_HC11.c); les poids de connexion sont stockés dans des fichiers de données, accessibles par le microcontrôleur, après avoir été convertis en valeurs hexadécimales (voir annexe D, fichier POIDS1.s); le programme fait appel à des sous-routines en langage assembleur disponibles avec le logiciel du cross-compileur pour langage C de Motorola [MOT90] et à des sous-routines d'acquisition de données et de calcul en point flottant [BRO91] (voir annexe D, fichier ROUTHFP.s).

Le traitement des données par le programme NEU\_HC11.c consiste en trois étapes:

1. acquisition et prétraitement des trois tensions d'entrée  $\tilde{v}_1$ ,  $\tilde{v}_2$  et  $\tilde{v}_G$  (équations (5.5) et (5.6)):

$$x_i^0 = \frac{v_i - v_{i \min}}{v_{i \max} - v_{i \min}} \quad \text{pour } i = 1, 2 \quad (5.5)$$

$$x_3^0 = \frac{v_G - v_{G \min}}{v_{G \max} - v_{G \min}}, \quad (5.6)$$

2. calcul de la pression reconstituée après propagation des tensions prétraitées dans le réseau de neurones, selon les équations (5.2) et (5.7):

$$s_i = f\left(\sum_{j=1}^{N_j} w_{ij}x_j - \theta_i\right) \quad (5.2)$$

$$y = \frac{P - P_{\min}}{2(P_{\max} - P_{\min})} + 0.25 \quad (5.7)$$

3. affichage de la pression reconstituée  $\hat{p}$  .

La précision de la pression reconstituée dépend principalement de la fonction de transition des neurones utilisée. Lors du calcul des poids de connexion, la fonction de transition est une fonction sigmoïdale décrite par l'équation (5.1), la fonction exponentielle n'est pas disponible avec le MC68HC11, elle doit donc être approximée. Deux cas ont été retenus: une approximation par une fonction polynomiale d'ordre 7; une combinaison d'une fonction linéaire et d'une fonction polynomiale d'ordre 2. Ces deux approximations nous ont permis d'évaluer l'influence de la précision sur la fonction de transition en terme de précision de reconstitution et de temps de calcul.

Le calcul des poids de connexion a été fait dans le cas de données synthétiques (voir section 6.3.2) avec un réseau de 50 neurones sur la couche cachée ( $N^2 = 50$ ), 80 exemples d'apprentissage ( $N_T.N_P = 4 \times 20$ ) et une précision sur la température  $\bar{\Delta T}^{etal} = 1^\circ C$ . L'approximation par une fonction polynomiale d'ordre 7 permet de reconstituer une pression avec la même précision que celle obtenue avec le programme utilisant le logiciel MATLAB, soit  $\delta \hat{p}_n \leq 5.10^{-2}$ . Le temps de calcul pour reconstituer une pression  $\hat{p}$  est approximativement de 3.5 secondes, la place mémoire nécessaire pour le stockage des données est de 1200 octets, c'est-à-dire 4 octets par poids de connexion en point flottant. C'est le nombre de poids de connexion du réseau implanté, contenus dans les matrices  $W^2$  et  $W^3$ , qui détermine l'espace mémoire requis. Une comparaison

avec les résultats de reconstitution de pression obtenus avec la méthode basée sur les fonctions spline a été faite en implantant l'algorithme dans le microcontrôleur MC68HC11, le temps de calcul par pression reconstituée est de 3.3 secondes et l'espace mémoire requis pour stocker les données est de 1280 bytes. Nous pouvons donc constater que les résultats avec les deux méthodes sont équivalents.

Dans ce cas, le temps de calcul pour reconstituer une pression est principalement dû à l'approximation de la fonction sigmoïde puisqu'elle nécessite 28 multiplications et 28 additions pour une seule évaluation. D'où l'idée d'utiliser la deuxième approximation par la combinaison d'une fonction linéaire et d'une fonction polynomiale d'ordre 2 qui nécessite au plus 3 multiplications et 3 additions. Un processus d'apprentissage du réseau de neurones a été refait avec cette nouvelle fonction de transition, les nouveaux poids calculés ont été implantés dans le microcontrôleur afin d'estimer la qualité de la reconstitution de pression. Le temps de calcul est diminué de manière significative, soit 1.3 secondes par reconstitution, mais la qualité de reconstitution s'en ressent puisque  $\delta \hat{p}_n \leq 10^{-1}$  est obtenu. L'espace mémoire requis est identique à celui estimé précédemment, soit 1200 octets, puisque le nombre de poids de connexion est inchangé. Nous remarquons l'importance du choix de la fonction sigmoïde sur le temps de calcul et l'erreur de reconstitution obtenue.

## 6.6 Conclusion

Dans ce chapitre, la méthode d'étalonnage et de reconstitution de mesurandes utilisant un réseau de neurones multicouches a été testée dans le cas d'un système de mesure de pression à fibres optiques [BOC90]. La méthode a été appliquée dans différents cas afin d'évaluer ses qualités, d'une part avec des données synthétiques et d'autre part avec des données expérimentales. Le principal avantage d'utiliser des données synthétiques

est la flexibilité qu'elles apportent en terme de quantité et de variété, il est possible de générer les données qui nous intéressent sans être limités par les contraintes imposées par la mesure dans le cas des données expérimentales.

Dans une première étape, le réseau de neurones mis au point à la section 5 a été testé pour le cas relativement simple où les données à traiter sont à température constante. Cette partie des tests nous a permis de mieux comprendre le comportement du réseau en fonction des différents paramètres de l'algorithme RETRO\_KAL+, de la structure du réseau et de l'ensemble des données d'apprentissage. La deuxième étape a permis d'ajouter les contraintes imposées par un facteur d'influence agissant directement sur les données à traiter par le réseau. Dans le cas du système de mesure de pression à fibres optiques, il faut prendre en compte l'effet de la température (chapitre 3). En exploitant la capacité de généralisation du réseau de neurones, nous avons montré qu'il est possible d'étalonner le système de mesure avec des informations limitées sur la température et le point le plus intéressant est la possibilité de reconstituer une pression sans aucune information sur la température. Dans les deux cas la température n'est pas une donnée d'entrée présentée au réseau de neurones. La capacité d'apprentissage de l'évolution des données permet donc au réseau de mémoriser l'effet du facteur d'influence contenu dans cette évolution et la capacité de généralisation permet de reconstituer des mesurandes dans le cas de données d'entrée bruitées. Les résultats obtenus ont été comparés avec la méthode basée sur les fonctions spline proposée dans [BAR92], les qualités de reconstitution de pressions sont comparables.

Le dernier groupe de traitement de données a permis de tester les performances du réseau dans le cas de données expérimentales. Malgré quelques difficultés dues en grande partie à la quantité limitée de données disponibles, nous avons montré que notre méthode

s'applique dans le cas de données réelles. Le système de mesure a été étalonné et les pressions sont reconstituées avec une précision satisfaisante. Pour compléter ce travail, l'implantation matérielle dans un microcontrôleur MC68HC11 de l'algorithme de reconstitution a été faite afin d'estimer la précision de la reconstitution, la place mémoire nécessaire et le temps de calcul requis.



## CHAPITRE 7

### CONCLUSION

L'objectif principal de ce travail consistait à étudier les possibilités d'application des réseaux de neurones aux problèmes d'étalonnage et de reconstitution de mesurandes statiques dans un système de mesure. Ce qui revient à faire le lien entre deux domaines de recherche: les systèmes de mesure et les réseaux de neurones. Ce lien devant nous permettre de proposer une nouvelle méthode basée sur les propriétés des réseaux de neurones et se présentant comme une alternative aux méthodes analytiques généralement utilisées pour résoudre les problèmes associés aux systèmes de mesure.

Le premier chapitre du travail est dédié au domaine des systèmes de mesure et a pour but d'en définir les notions de base. Deux opérations principales sont considérées, l'*étalonnage* du système et la *reconstitution de mesurandes*, elles constituent les deux problèmes à résoudre dans le cadre du travail. Quelques méthodes d'étalonnage et de reconstitution de mesurandes analytiques ont été présentées, elles donnent de très bons résultats dans le cas où l'on dispose suffisamment d'informations sur le modèle mathématique du système de mesure et sur les bruits de perturbation. Cependant il est parfois impossible d'avoir ces informations et donc ces méthodes ne sont pas toujours applicables pour résoudre n'importe quel problème de reconstitution, c'est le cas des systèmes de mesure fortement non-linéaires. Avant d'aborder le domaine des réseaux de neurones, nous avons choisi de présenter le système de mesure qui nous a permis de valider la méthode d'étalonnage et de reconstitution de mesurandes proposée dans ce travail. Ce système de mesure est un système de mesure de pression, il est composé principalement de trois

capteurs de pression à fibre optique et dont la caractéristique d'entrée/sortie est quasi-sinusoidale, et qui subissent une forte influence de la température. En effet celle-ci crée un déphasage sur les caractéristiques des capteurs, et il faut donc en tenir compte pour étalonner le système de mesure et reconstituer les pressions.

Suite à une introduction au domaine des réseaux de neurones faite au chapitre 4, où les notions de bases, les principaux types de réseaux de neurones, les algorithmes d'apprentissage et les domaines d'application sont passés en revue, nous avons pu choisir le type de réseau de neurones le plus approprié pour notre application. Les réseaux de neurones multicouches présentent une topologie qui s'accorde avec le type de données à traiter (vecteur à l'entrée, scalaire en sortie), leurs capacités d'apprentissage et de stockage sont élevées et ils sont relativement facile d'utilisation. Cependant leur phase d'apprentissage est apparue fastidieuse et une bonne partie du travail a consisté à mettre au point un algorithme d'apprentissage puissant et rapide. Le chapitre 5 fait état des différents résultats obtenus avec d'une part la méthode standard de la rétropropagation du gradient, appelée RETRO\_GRAD, la version modifiée à l'aide du filtre de Kalman [SCA92], appelée RETRO\_KAL, et finalement la méthode adoptée, obtenue en améliorant la vitesse de convergence de l'erreur de rétropropagation de la méthode RETRO\_KAL, appelée pour cela RETRO\_KAL+ [MAS94]. L'application du filtre de Kalman pour l'apprentissage du réseau de neurones permet de réduire de manière considérable le temps d'apprentissage en nécessitant un nombre relativement faible d'itérations, environ 20% du nombre d'itérations requis par la méthode de la rétropropagation du gradient [SCA92]. À la fin du chapitre 5, l'état d'avancement du travail est tel que nous disposons d'un réseau de neurones fonctionnel pour l'apprentissage de données générées par le système de mesure de pression à fibres optiques.

Les résultats d'étalonnage et de reconstitution de mesurandes pour le système de mesure de pression à fibres optiques ont été réalisés pour des données synthétiques et réelles pour différentes valeurs de température. Nous rappelons que l'objectif principal était de limiter au maximum l'utilisation de la mesure de la température qui est un facteur d'influence. Les résultats, présentés au chapitre 6, montre la flexibilité dont font preuve les réseaux de neurones comme outils d'étalonnage dans le cas d'un système de mesure fortement non-linéaire et sensible à des facteurs d'influence, la température dans le cas étudié. Contrairement aux méthodes analytiques déjà proposées, [BAR90] et [MOR91], et qui prennent en compte ce facteur d'influence en mesurant la température et en la traitant comme donnée d'entrée, la méthode proposée basée sur les réseaux de neurones a permis de tirer l'information sur la température directement des caractéristiques d'entrée/sortie des trois capteurs. En effet, durant la phase d'apprentissage, le réseau apprend l'évolution des données utilisées pour l'étalonnage du système de mesure et mémorise l'effet de la température sur cette évolution. Par la suite, en exploitant la capacité de généralisation des réseaux de neurones en phase de reconnaissance, il est possible de reconstituer des pressions à partir des tensions de sortie des capteurs de pression, sans avoir à mesurer la température. Puisque, l'effet de la température agit sur les triplets des tensions  $(\tilde{v}_1, \tilde{v}_2, \tilde{v}_G)$  selon l'équation (5.3) pour les données synthétiques, nous avons vérifié l'effet de l'erreur sur les données exactes de référence  $\overline{\Delta T}^{etal}$  que nous devons considérer lors de l'acquisition des données pour l'étalonnage. Une comparaison de résultats a été obtenue avec la méthode basée sur les fonctions spline [LEM95], [BAR90], les résultats de reconstitution sont comparables dans le cas d'une faible erreur de mesure sur la température lors de l'étalonnage et meilleurs sinon.

Pour conclure ce travail, nous terminons par quelques remarques concernant les résultats obtenus avec notre méthode [MAS94] selon les figures 6.8 à 6.13:

- Pour l'étalonnage du système, le fait de mesurer la température avec une faible précision permet de faciliter et de réduire le temps d'acquisition des données de référence. Cependant le temps de calcul pour entrainer le réseau à l'application visée, en fonction du nombre de pression  $N_p$  et de température  $N_T$ , est long et donc la phase d'étalonnage du système de mesure est plus longue en comparaison avec d'autres méthodes analytiques.
- Pour la reconstitution de pression, le fait de ne pas avoir à mesurer la température est un avantage indéniable, l'effet de la température a été appris et l'information sur celle-ci est extraite des tensions de sortie des capteurs de pression. Ce résultats est d'autant plus intéressant que pour des mesures ponctuelles de pression il peut être difficile de mesurer la température avec précision.
- La qualité de reconstitution de la pression  $p$  est pratiquement indépendante de la précision sur la mesure de la température lors de l'acquisition les données de référence utilisées pour l'étalonnage du système.
- L'implantation matérielle de notre méthode et de la méthode basée sur les fonctions splines [LEM95] donne des résultats comparables en terme de précision de reconstitution, de temps de calcul et d'espace mémoire.

Nous noterons comme remarque finale que les résultats de l'étude suggèrent une possibilité intéressante de développement futur, soit que le même système de mesure, ayant la même implantation matérielle, pourrait être utilisé pour mesurer la température simultanément avec la mesure de la pression. Autrement dit, cela reviendrait à considérer la température comme un deuxième mesurande à reconstituer à partir des résultats de conversion, soient les trois tensions de sortie des capteurs de pression qui sont aussi des fonctions de la température.

## RÉFÉRENCES

- [AND79] Anderson, B.D.O., Moore, J.B., Optimal Filtering, Prentice Hall, 1979.
- [ASC87] Asch, G., Les capteurs en instrumentation industrielle, Dunod, Paris, 1987.
- [BAR90] Barwicz, A., Bock, W.J., "An electronic high-pressure measuring system using a polarimetric fiber-optic sensor", IEEE Transactions on Instrumentation and Measurement, Vol 39, No 6, December 1990.
- [BAR92] Barwicz, A., Morawski, R.Z., Lemire, L., Bock, W.J., "Calibration of an Electronic Multisensor System for Measuring High Pressures", IEEE Transactions on Instrumentation and Measurement, Vol 41, No 2, April 1992.
- [BEN91] Ben Slima, M., Étude de l'application d'un processeur numérique de signaux (DSP) dans les systèmes de mesure électroniques, mémoire de maîtrise, Université du Québec à Trois-Rivières, 1991.
- [BOC89] Bock, W.J., Domanski, A.W., "Hight Hydrostatic Pressure Effects in Highly Birefringent Optical Fibers", Journal of Lightware Technology, Vol. 7, No 8, 1989, pp.1279-1283.
- [BRO90] Brouard, P., "Étude d'un cross-compileur du langage C", Rapport de projet de fin d'études, Université du Québec à Trois-Rivières, 1990.
- [CAR89] Carpenter, G.A., "Neural Network Models for Pattern Recognition and Associative Memory", Neural Networks, Vol. 2, 1989.
- [CAR91] Carpenter, G.A. and Grossberg, S., Pattern Recognition by Self-Organizing Neural Networks, MIT, 1991, p. 691.
- [CAU90] Caudill, M., Butler, C., Naturally Intelligent Systems, The MIT Press Cambridge, Massachussets London, England, 1990, 300 pages.
- [DAV89] Davalo, E., Naïm, P., Des réseaux de neurones, Eyrolles, Paris, 1989.
- [DEM86] Demoment, G., Déconvolution des signaux, Rapport interne No L2S 20/84 du Laboratoire des Signaux et Systèmes (CNRS/ESE) et Notes de cours de l'École Sup. d'Électricité, no 3086/85, 1986.
- [EUB88] Eubank, R.L., Spline Smoothing and Nonparametric Regression, New-York: Marcel Dekker, 1988.

- [FIN75] Finkelstein, L., "Fundamental concept of measurement: Definitions and scales", Measurement and Control, Vol. 8, March 1975.
- [GRO76] Grossberg, S., "Adaptive Pattern Classification and Universal Recoding, II: Feedback, Expectation, Olfaction, and Illusions", Biological Cybernetics, Vol. 23, pp. 187-202, 1976.
- [HEB49] Hebb, D., Organization of Behavior, New York: Wiley.
- [HOP82] Hopfield, J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", Proc. Nat. Acad. Sci. U.S.A., Vol. 79, pp. 2554-2558, 1982.
- [HUA91] Huang, T., Tsuyuki, Yasuhara, M., "A learning Algorithm of Neural Network Based on Kalman Filtering", IEICE Transaction, Vol.E74, No 5, pp.1059-1065, 1991.
- [IIG92] Iiguni, Y., Sakai. H., Tokumaru, H., "A Real-Time Learning Algorithm for a Multilayered Neural Network Based on the Extended Kalman Filter", IEEE Transaction on Signal Processing, Vol. 40, No 4, April 1992, pp. 959-966.
- [KOH84] Kohonen, T., Self-organization and Associative Memory, Berlin: Springer-Verlag, 1984.
- [LEM95] Lemire, L., Étalonnage statique d'un système de mesure à fibres optiques, Mémoire de maîtrise, Université du Québec à Trois-Rivières, 1994.
- [LIP87] Lippmann, R.P., "An Introduction to Computing with Neural Nets", IEEE Acoustics, Speech, and Signal Processing Magazine, pp. 4-22, April 1987.
- [MAS91] Massicotte, D., Reconstitution du mesurande basé sur le filtrage de Kalman: implantation sur la carte DSP56000, Rapport interne, Groupe de recherche en électronique industrielle, Laboratoire de systèmes de mesure, Université du Québec à Trois-Rivières, 1991, 33 pages.
- [MAS94] Massicotte, D., Legendre, S., Barwicz, A., "Calibration of a High-Pressure Measuring System Using Neural Networks, enabling Influence Factor-measurement-Free Pressure Reconstruction", soumis à IEEE Transactions on Instrumentation and Measurement, janvier 1994.
- [MAT89] MATLAB, User's Guide, The MathWorks Inc., p. 3.173, 1989.
- [MIN] Minsky, M., Papert, S., Perceptrons, The MIT Press, .
- [MOR89] Morawski, R.Z., "The Methods of Measurement Signal Reconstruction", Metrology and Measuring Systems, Monography No 1, 1989. (En Polonais)

- [MOR91] Morawski, R.Z., Lemire, L., Barwicz, A., Bock, W.J., "Calibration of Electronic Multisensor System for Measuring High Pressures", Record of the IEEE Instr. and Meas., Techn. Conf. IMTC'91 (Atlanta, May 14-16, 1991).
- [MOT90] Motorola, M68HC11 Family Cross C Compiler User's Manual for MS-DOS/PC-DOS, Motorola Inc, 1990.
- [PAR82] Parker, D., "Learning Logic", Invention Report, S81-64, File 1, Office of Technology Licensing, Stanford University, 1982.
- [PAR86] Paratte, P.-A. et Robert, P., Systèmes de mesure, Collection Traité d'électricité, Volume XVII, Presses Polytechniques Romandes, 1986, 368 pages.
- [PER90] Personnaz, L., Dreyfus, G., "Les réseaux de neurones, situation et perspectives", Revue Générale de l'Électricité, No 5, 1990, pp 27-34.
- [ROS58] Rosenblatt, F., "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain", Psychological Review, Vol. 65, pp 386-408, 1958.
- [RUM86a] Rumelhart, D.E., McClelland, J.L., Parallel Distributed Processing, The MIT Press, Vol. 1, 1986.
- [RUM86b] Rumelhart, D., Hinton, G., Williams, R., "Learning Representations by Backpropagating Errors", Nature, Vol. 323, pp. 533-536, 1986.
- [SÀN92] Sánchez-Sinencio, E., Lau, C., Artificial Neural Networks, IEEE Press, 1992.
- [SCA92] Scalero, R.S., Tepedelenlioglu, N., "A fast new algorithm for training feedforward neural networks", IEEE Transactions on Signal Processing, vol 40, No 1, January 1992, pp 102-210.
- [SCH81] Schumaker, L.L., Spline Functions - Basic Theory, New York, J. Wiley and Sons, Inc 1981.
- [SIN89] Singhal, S., and Wu, L., "Training Feed-Forward Networks with the Extended Kalman Algorithm", Proceeding IEEE International Conference on Acoustics, Speech, and Signal Processings, pp. 1187-1190, May 1989.
- [VIL92] Villiers (de), J., Barnard, E., "Backpropagation Neural Nets with One and Two Hidden Layers", IEEE Transaction on Neural Networks, Vol. 4, No. 1, pp. 136-141, January 1992.

- [WER74] Werbos, P., "Beyond Regression", PH.D. Dissertation, Harvard University, Cambridge, M.A., 1974.
- [WID60] Widrow, B., and Hoff, M.E., "Adaptive Switching Circuits", 1960 IRE WESCON Convention Record, Part 4, pp. 96-104, 1960.
- [WIL86] Williams, R., "Reinforced Learning in Connection to Networks: A Mathematical Analysis", Technical Reprot No. 8605, University of California, Institute for Cognitive Science, 1986.



## **ANNEXE A**

### **Programme d'initialisation du réseaux de neurones**

```

%=====
% PROGRAMME D'INITIALISATION
%
% NOM : nkalini3.m
% DATE : juin 1993
% PAR : Sylvie Legendre
%=====

%
% ENTREE DES DIMENSIONS DU RÉSEAU: L et Nc(i)
%
%L = 1 % Nombre de couches cachées
%Nc = [20 10 5] % Nombre de neurones par couche
%

L = input('Entrez le nombre de couches cachées, L + 2 ≤ 10 : ');
Nc = input('Entrez le nombre de neurones par couche de la première à la dernière: [ Nc(i) ] : ');

L = L + 2;

%
% INITIALISATION DES POIDS DE PONDÉRATION
% ET DES MATRICES DE CORRÉLATION
%

fact_init = 10;
init = 0.001;
rand('normal')

if L-2 >= 2
    W4 = fact_init*rand(Nc(4),Nc(3));
    R4 = init*eye(Nc(3));
end

if L-2 >= 1
    W3 = fact_init*rand(Nc(3),Nc(2));
    R3 = init*eye(Nc(2));
end

if L-2 >= 0
    W2 = fact_init*rand(Nc(2),Nc(1));
    R2 = init*eye(Nc(1));
end

k_retro = 0;
k_iter = 1;
pasretro = 1;
no_vec = 0;
a = 1;
b = ones(1,L);
mu = ones(1,L);
Nb_max = 500;
E_seuil = 0.001;
k_ex=1;
erreur = 0;

save neu_res

```

## **ANNEXE B**

### **Programme d'apprentissage**

**Fonctions: RETRO\_GRAD, RETRO\_KAL, RETRO\_KAL+**

```

%=====
% PROGRAMME D'APPRENTISSAGE DES EXEMPLES
%
% NOM : NKAL_ER1.m
% DATE : 2 juin 1993
% PAR : Sylvie Legendre
%
% SOUS-PROGRAMMES : nkalini3.m
%                   pro_er1.m
%                   ret_er1.m
%=====

rand('uniform')

[n,Nb_vec] = size(X);

%-----
% PROPAGATION LINEAIRE DES EXEMPLES
%-----

for no_vec = 1:Nb_vec

    %
    % Propagation du vecteur d'entrée
    %
    E1 = X(:,no_vec);
    pro_er1

end % Fin de la boucle for no_vec = 1:Nb_vec

%
% Calcul des erreurs
%
ERMoy(pasretro) = sqrt(sum( (Y-Yrec).^2 ) ./ (sum (Y.^2)) );
EAmoy(pasretro) = sqrt(sum(Eabs.^2)/Nb_vec);
EAsup(pasretro) = max( abs(Yrec - Y) );
ERSup(pasretro) = max( Erel );

%-----
% APPRENTISSAGE ALÉATOIRE
%-----

k_ex = 1;
while ERMoy(pasretro) > E_seuil & erreur == 0 & pasretro <= pas_max

    %
    % Choix aléatoire de l'exemple
    %
    no_vec = Ex((pasretro-1)*round(Nb_vec/2) + k_ex);
    k_ex = k_ex + 1;
    %
    % Propagation du vecteur
    %
    E1 = X(:,no_vec);
    pro_er1

    %
    % Rétropropagation de l'erreur selon différentes méthodes
    % ret_er1.m = RETRO_GRAD ou RETRO_KAL ou RETRO_KAL+
    %
    k_retro = k_retro + 1;
    ret_er1

    if rem(k_ex,fix(Nb_vec/2)) == 0
        k_ex = 1;
        %
        % Propagation de tous les exemples
        %
        pasretro = pasretro + 1;
    end
end

```

```

for no_vec = 1:Nb_vec
    %
    % Propagation du vecteur d'entrée
    %
    E1 = X(:,no_vec);
    pro_er1

end % Fin de la boucle for no_vec = 1:Nb_vec

%
% Calcul des erreurs
%
ERMoy(pasretro) = sqrt(sum( (Y-Yrec).^2 ) ./ (sum (Y.^2)) );
EAmoy(pasretro) = sqrt(sum(Eabs.^2)/Nb_vec);
EAsup(pasretro) = max( abs(Yrec - Y) );
ERsup(pasretro) = max( Erel );

%
% Affichage de la courbe d'apprentissage
%
semilogy(EAmoy)
title ('EAmoy')

end % Fin de la condition if rem(k_retro,fix(Nb_nec/2)) == 0
end % Fin de la boucle while ERMoy(pasretro) > E_seuil & pasretro*k_ex < 10000

```

```

%=====
% PROPAGATION D'UN EXEMPLE
%
% NOM : PRO_ER1.m
% DATE : 2 juin 1993
% PAR : Sylvie Legendre
%=====

%
% Propagation du vecteur d'entrée
%
if L-2 >= 0
    S1 = F(E1,a);
    E2 = W2 * S1;
end

if L-2 >= 1
    S2 = F(E2,a);
    E3 = W3 * S2;
end

if L-2 >= 2
    S3 = F(E3,a);
    E4 = W4 * S3;
end

end

%
% Recherche des vecteurs EL et SL
%
if L-2 >= 2
    EL = E4;
    SL = F(E4,a);
    S4 = SL;

elseif L-2 >= 1
    EL = E3;
    SL = F(E3,a);
    S3 = SL;

elseif L-2 >= 0
    EL = E2;
    SL = F(E2,a);
    S2 = SL;

end

Yrec(:,no_vec)=SL;

%
% Calcul de l'erreur absolue moyenne entre
% la valeur de sortie désirée et celle obtenue
%
Eabs(no_vec) = abs(Y(:,no_vec)-SL);

%
% Calcul de l'erreur relative
%
Rel(no_vec) = abs( ( Yrec(:,no_vec) - Y(:,no_vec) ) ./ Y(:,no_vec) );

erreur = 0;
if isnan(Eabs(no_vec)) == 1 | finite(Eabs(no_vec)) == 0
    erreur = 1
    break;
end

```

```

%=====
%  FONCTION SIGMOÏDE
%=====

function [FE] = F(E,a)
[m,n] = size(E);
for i = 1:m
    for j = 1:n
        FE(i,j) = 1/( 1 + exp(-a*E(i,j)) );
    end
end

%=====
%  FONCTION SIGMOÏDE DÉRIVÉE
%=====

function [F_der] = Fder(E,a)
[m,n] = size(E);

for i = 1:m
    for j = 1:n
        if E(i,j) > -1e2
            F_der(i,j) = a*exp(-a*E(i,j))/( 1 + exp(-a*E(i,j)) )^2;
        else
            F_der(i,j) = 0;
        end
    end
end

%=====
%  FONCTION SIGMOÏDE INVERSE
%=====

function [FE] = F_1(E,a)
[m,n] = size(E);
for i = 1:m
    for j = 1:n
        FE(i,j) = -log( 1/E(i,j) - 1 )/a;
    end
end

```

```

%=====
% FONCTION RETRO_GRAD
%
% NOM : ret_er1.m
% DATE : 02 novembre 1993
% PAR : Sylvie Legendre
%=====

%
% Pas de rétropropagation
%

mu(2) = m2_1;
mu(3) = m3_1;
% mu(4) = m4_1;

%
% Calcul du vecteur d'erreur a la couche de sortie
%

if L-2 >= 2
    ER4 = Fder(E4,a) .* ( Y(:,no_vec) - S4 );
elseif L-2 >= 1
    ER3 = Fder(E3,a) .* ( Y(:,no_vec) - S3 );
elseif L-2 >= 0
    ER2 = Fder(E2,a) .* ( Y(:,no_vec) - S2 );
end

%
% Calcul des vecteurs d'erreur
% aux couches cachées
%

if L-2 >= 2
    ER3 = ( W4' * ER4 ) .* Fder(E3,a);
end

if L-2 >= 1
    ER2 = ( W3' * ER3 ) .* Fder(E2,a);
end

%
% Modification des poids de la couche de sortie
%

if L-2 >= 2
    W4 = W4 + 2 * mu(4) * ER4 * S3';
elseif L-2 >= 1
    W3 = W3 + 2 * mu(3) * ER3 * S2';
elseif L-2 >= 0
    W2 = W2 + 2 * mu(2) * ER2 * S1';
end

%
% Modification des poids aux couches cachees
%
if L-2 >= 2
    W3 = W3 + mu(3) * ER3 * S2';
end

if L-2 >= 1
    W2 = W2 + mu(2) * ER2 * S1';
end

%
% Fin de la retropropagation
%
```



```

%=====
% FONCTION RETRO_KAL
%
% NOM : ret_er1.m
% DATE : 30 mai 1993
% PAR : Sylvie Legendre
%=====

mu(2) = m2_1;
mu(3) = m3_1;
b(2) = b2_1;
b(3) = b(2);

%
% Calcul des vecteurs gain Kj
% et des matrices de correlation Rj
%

if L-2 >= 0
    be(2) = b(2) + S1' * R2 * S1;
    K2 = ( R2 * S1 ) / be(2);
    R2 = ( eye(Nc(1)) - K2 * S1' ) * R2 / b(2);
end

if L-2 >= 1
    be(3) = b(3) + S2' * R3 * S2;
    K3 = ( R3 * S2 ) / be(3);
    R3 = ( eye(Nc(2)) - K3 * S2' ) * R3 / b(3);
end

if L-2 >= 2
    be(4) = b(4) + S3' * R4 * S3;
    K4 = ( R4 * S3 ) / be(4);
    R4 = ( eye(Nc(3)) - K4 * S3' ) * R4 / b(4);
end

%
% Calcul du vecteur d'erreur à la couche de sortie
%

if L-2 >= 2
    ER4 = Fder(E4,a) .* ( Y(:,no_vec) - S4 ) ./ Y(:,no_vec);
elseif L-2 >= 1
    ER3 = Fder(E3,a) .* ( Y(:,no_vec) - S3 ) ./ Y(:,no_vec);
elseif L-2 >= 0
    ER2 = Fder(E2,a) .* ( Y(:,no_vec) - S2 ) ./ Y(:,no_vec);
end

%
% Calcul des vecteurs d'innovation
% aux couches cachees
%

if L-2 >= 2
    ER3 = ( W4' * ER4 ) .* Fder(E3,a);
end

if L-2 >= 1
    ER2 = ( W3' * ER3 ) .* Fder(E2,a);
end

%
% Modification des poids de la couche de sortie
%

if L-2 >= 2
    ER4 = (F_1( Y(:,no_vec),a ) - E4);

```

```
W4 = W4 + mu(4) * ER4 * K4';  
elseif L-2 >= 1  
    ER3 = (F_1( Y(:,no_vec),a ) - E3);  
    W3 = W3 + mu(3) * ER3 * K3';  
elseif L-2 >= 0  
    ER2 = (F_1( Y(:,no_vec),a ) - E2);  
    W2 = W2 + mu(2) * ER2 * K2';  
end  
  
%  
% Modification des poids aux couches cachees  
%  
if L-2 >= 2  
    W3 = W3 + mu(3) * ER3 * K3';  
end  
if L-2 >= 1  
    W2 = W2 + mu(2) * ER2 * K2';  
end  
  
%  
% Fin de la retropropagation  
%
```

```

%=====
% FONCTION RETRO_KAL+
%
% NOM : ret_er1.m
% DATE : 30 mai 1993
% PAR : Sylvie Legendre
%=====

mu(2) = m2_1 + m2_2*exp(-EAmoy(pasretro)/m2_3);
mu(3) = m3_1 + m3_2*exp(-EAmoy(pasretro)/m3_3);
b(2) = b2_1 + b2_2*exp(-EAmoy(pasretro)/b2_3);
b(3) = b(2);

%
% Calcul des vecteurs gain Kj
% et des matrices de correlation Rj
%

if L-2 >= 0
    be(2) = b(2) + S1' * R2 * S1;
    K2 = ( R2 * S1 ) / be(2);
    R2 = ( eye(Nc(1)) - K2 * S1' ) * R2 / b(2);
end

if L-2 >= 1
    be(3) = b(3) + S2' * R3 * S2;
    K3 = ( R3 * S2 ) / be(3);
    R3 = ( eye(Nc(2)) - K3 * S2' ) * R3 / b(3);
end

if L-2 >= 2
    be(4) = b(4) + S3' * R4 * S3;
    K4 = ( R4 * S3 ) / be(4);
    R4 = ( eye(Nc(3)) - K4 * S3' ) * R4 / b(4);
end

%
% Calcul du vecteur d'erreur à la couche de sortie
%

if L-2 >= 2
    ER4 = Fder(E4,a) .* ( Y(:,no_vec) - S4 ) ./ Y(:,no_vec);
elseif L-2 >= 1
    ER3 = Fder(E3,a) .* ( Y(:,no_vec) - S3 ) ./ Y(:,no_vec);
elseif L-2 >= 0
    ER2 = Fder(E2,a) .* ( Y(:,no_vec) - S2 ) ./ Y(:,no_vec);
end

%
% Calcul des vecteurs d'innovation
% aux couches cachees
%

if L-2 >= 2
    ER3 = ( W4' * ER4 ) .* Fder(E3,a);
end

if L-2 >= 1
    ER2 = ( W3' * ER3 ) .* Fder(E2,a);
end

%
% Modification des poids de la couche de sortie
%

if L-2 >= 2
    ER4 = (F_1( Y(:,no_vec),a ) - E4);

```

```
W4 = W4 + mu(4) * ER4 * K4';  
elseif L-2 >= 1  
    ER3 = (F_1( Y(:,no_vec),a ) - E3);  
    W3 = W3 + mu(3) * ER3 * K3';  
elseif L-2 >= 0  
    ER2 = (F_1( Y(:,no_vec),a ) - E2);  
    W2 = W2 + mu(2) * ER2 * K2';  
end  
  
%  
% Modification des poids aux couches cachees  
%  
  
if L-2 >= 2  
    W3 = W3 + mu(3) * ER3 * K3';  
end  
  
if L-2 >= 1  
    W2 = W2 + mu(2) * ER2 * K2';  
end  
  
%  
% Fin de la retropropagation  
%
```

## **ANNEXE C**

### **Programme de généralisation**

```

%=====
% TEST DE LA CAPACITÉ DE GÉNÉRALISATION
%
% NOM : npro09.m
% DATE : 20 septembre 1993
% PAR : Sylvie Legendre
%=====

%
% Chargement du fichier contenant les poids
% de connexion du réseau entraîné
%
load reseau.neu

%
% Chargement du fichier contenant les exemples
% à propager dans le réseau
% contient: Xa, Ya
%
load donnees.pro

for i = 0:(NT-1) % où NT est le nombre de température

    i
    X = Xa(:,i*Np+1:i*Np+Np);
    Y = Ya(i*Np+1:i*Np+Np); % où Np est le nombre de pression

    [n,Nb_vec] = size(X);
    for no_vec = 1:Nb_vec
        %
        % Initialisation de la couche d'entrée
        %
        E1 = X(:,no_vec);

        %
        % Propagation du vecteur d'entrée
        %
        pro_er1

    end % Fin de la boucle for no_vec = 1:Nb_vec

    %
    % calcul de la pression reconstituée
    % et des erreurs de reconstitution
    %
    Prec(:,i+1) = 2*( max(P) - min(P) )*(Yrec' - 0.25) + min(P);
    EAg(:,i+1) = abs( P - Prec(:,i+1) );
    EAgmoy(1,i+1) = EAmoy*( max(P) - min(P) );
    ERg(:,i+1) = abs( ( P - Prec(:,i+1) ) ./ P );
    ERgsup(i+1) = max(ERg(1,:));
    YPrec(:,i+1) = Yrec';

end

save pro.neu EAg EAgmoy ERg ERgsup YPr

```

## **ANNEXE D**

### **Programmes d'implantation dans le MC68HC11**

```

/* =====
Programme de reconstitution d'une pression
à l'aide d'un réseau de neurones.
Microcontrôleur 68HC11.

NOM   : NEU_HC11.c
DATE  : août 1993
PAR   : Sylvie Legndre
=====*/

extern float W2[60][4], W3[60];

char p[14];
int lv1,lv2,lv3;
float LEC_V1,LEC_V2,LEC_VG,X[4],Y,PINT;

extern int lectad();
extern void trig();
extern void affich(char st[]);
extern void send(char st[]);
extern float fpabs(float x);
extern void putfp(char st[],float f);
extern float getfp(char st[]);
extern void attend();

float f(float x);
float puiss(float x, int n);
float nprohc11(float *vec, float W2[60][4], float *W3);

main()
(
/* Lecture des tensions des capteurs V1, V2 et Vg
appliquées au convertisseur A/N */

LEC_V1=0;
lv1=lectad();
trig();
attend();
lv2=lectad();
trig();
attend();
lv3=lectad();
LEC_V1=(float)lv1/819;
LEC_V2=(float)lv2/819;
LEC_VG=(float)lv3/819;

/* Normalisation des tensions V1, V2 et Vg */

X[0] = LEC_V1/5;
X[1] = LEC_V2/5;
X[2] = LEC_VG/(8.17468716e-1);
X[3] = 0;

/* Programme de propagation de V1_norm, V2_norm et Vg_norm
et récupération de la sortie du réseau */

Y = nprohc11(X, W2, W3);

/* Résultat de reconstitution de la pression P */

PINT = (Y - 0.25)*50;

/* Affichage du résultat de reconstitution de la pression P */

putfp(p,PINT);
send(p);
affich(p);
trig();

)

```



```
/* fonction de propagation d'un vecteur d'entrée X (dim 4)
dans le réseau de neurones. */
```

```
float nprohc11(vec, W2, W3)
float vec[4], W2[60][4], W3[60];
{
  int i,j;
  float vec1[4], vec2[60], vec3;
  for (i=0;i<4;i++)
    vec1[i] = f(vec[i]);
  for (i=0;i<60;i++)
  {
    vec2[i] = (float)0;
    for (j=0;j<4;j++)
      vec2[i] = vec2[i] + W2[i][j]*vec1[j];
    vec2[i] = f(vec2[i]);
  }
  vec3 = (float)0;
  for (i=0;i<60;i++)
    vec3 = vec3 + W3[i]*vec2[i];
  vec3 = f(vec3);
  return (vec3);
}
```

```
/* fonction sigmoïde approximée par un polynôme
du 7ème ordre */
```

```
float f(x)
float x;
{
  int i,j;
  float fx, c[8];
  c[0] = 0.5001959;
  c[1] = 0.2461736;
  c[2] = 0.01381953;
  c[3] = -0.0404859;
  c[4] = 0.01339151;
  c[5] = -0.002109816;
  c[6] = 1.680830e-4;
  c[7] = -5.443581e-6;
  if (x >= 7)
    fx = 1;
  else if ((x >= 0) && (x < 7))
  {
    fx = c[0];
    for (j=1;j<8;j++)
      fx = fx + c[j]*puiss(x,j);
  }
  else if ((x >= -7) && (x < 0))
  {
    x = (-x);
    fx = c[0];
    for (j=1;j<8;j++)
      fx = fx + c[j]*puiss(x,j);
    fx = 1 - fx;
  }
  else
    fx = 0;
  return (fx);
}
```

```
/* fonction puissance, retourne le nombre
réel x à la puissance entière n */
```

```
float puiss(x,n)
float x;
```

```
int n;  
{  
  int i;  
  float p;  
  p=1;  
  for (i=1;i<=n;i++)  
    p = p*x;  
  return (p);  
}
```

```

/*-----
SOUS-ROUTINES ASSEMBLEUR
NOM : ROUTFP.S
-----*/

```

```

                                psct
*****
*                                GETFPACx SUBROUTINE
*****
                                XDEF  GTFPAC1C
GTFPAC1C CLR  MANTSGN1
          LDD  0,%x
          BEQ  GETFP12C
          LSLD
          BCC  GETFP11C
          COM  MANTSGN1
GETFP11C SEC
          RORB
          INCA
          INCA
GETFP12C STD  FPACC1EX
          LDD  2,%x
          STD  FPACC1MN+1
          RTS
*
                                XDEF  GTFPAC2C
GTFPAC2C CLR  MANTSGN2
          LDD  0,%x
          BEQ  GETFP21C
          LSLD
          BCC  GETFP22C
          COM  MANTSGN2
GETFP22C SEC
          RORB
          INCA
          INCA
GETFP21C STD  FPACC2EX
          LDD  2,%x
          STD  FPACC2MN+1
          RTS
*
*****
*                                PUTFPACx SUBROUTINE
*****
                                XDEF  PTFPAC1C
PTFPAC1C EQU  *
          LDD  FPACC1EX
          DECA
          DECA
          LSLB
          LSRD
          TST  MANTSGN1
          BEQ  PUTFP11C
          ORAA #$80
PTFP11C STD  0,%x
          LDD  FPACC1MN+1
          STD  2,%x
          RTS
*
                                XDEF  PTFPAC2C
PTFPAC2C EQU  *
          LDD  FPACC2EX
          DECA
          DECA
          LSLB
          LSRD
          TST  MANTSGN2
          BEQ  PUTFP12C
          ORAA #$80
PTFP12C STD  0,%x
          LDD  FPACC2MN+2
          STD  2,%x

```

```

      RTS
*
*****
*      INTERFACE C/ASSEMBLEUR
*****
      xdef      .addf
.addf  pshx
      tsx
      ldab     #$04
      abx
      jsr      GTFPAC2C
      ldab     #$04
      abx
      jsr      GTFPAC1C
      jsr      FLTADD
      tsx
      ldd      2,%x
      std      6,%x
      ldd      0,%x
      std      4,%x
      ldab     #$08
      abx
      jsr      PTFPAC1C
      ins
      ins
      ins
      pulx
      rts
*
      xdef      .subf
.subf  pshx
      tsx
      ldab     #$04
      abx
      jsr      GTFPAC2C
      ldab     #$04
      abx
      jsr      GTFPAC1C
      jsr      FLTSUB
      tsx
      ldd      2,%x
      std      6,%x
      ldd      0,%x
      std      4,%x
      ldab     #$08
      abx
      jsr      PTFPAC1C
      ins
      ins
      ins
      pulx
      rts
*
      xdef      .mulf
.mulf  pshx
      tsx
      ldab     #$04
      abx
      jsr      GTFPAC2C
      ldab     #$04
      abx
      jsr      GTFPAC1C
      jsr      FLTMUL
      tsx
      ldd      2,%x
      std      6,%x
      ldd      0,%x
      std      4,%x
      ldab     #$08
      abx
      jsr      PTFPAC1C

```

```

ins
ins
ins
ins
puls
rts
*
.divf    pshx    xdef    .divf
        tsx
        ldab    #$04
        abx
        jsr    GTFPAC2C
        ldab    #$04
        abx
        jsr    GTFPAC1C
        jsr    FLTDIV
        tsx
        ldd    2,%x
        std    6,%x
        ldd    0,%x
        std    4,%x
        ldab    #$08
        abx
        jsr    PTFPAC1C
        ins
        ins
        ins
        ins
        puls
        rts
*
.cmpf    pshx    xdef    .cmpf
        tsx
        ldab    #$04
        abx
        jsr    GTFPAC2C
        ldab    #$04
        abx
        jsr    GTFPAC1C
        tsx
        ldd    0,%x
        std    8,%x
        ldd    2,%x
        std    10,%x
        ldab    #$08
        abx
        txs
        pshy
        ldaa    MANTSGN1
        cmpa    #$00
        bhi    condi1
        jmp    else1
condi1   ldaa    MANTSGN2
        cmpa    #$00
        bhi    condi2
        jmp    else2
condi2   ldx    #FPACC2EX
        ldy    #FPACC1EX
loop1    ldaa    0,%x
        cmpa    0,%y
        blo    cas11
        bhi    cas21
        beq    cas31
cas11    tpa
        anda    #%11111011
        oraa    #%00001000
        tap
        jmp
cas21    tpa
        anda    #%11110011
        tap

```

```

cas31  inx      jmp      fin
        iny
        cpy      #MANTSGN1
        beq      egal1
        jmp      loop1
egal1   tpa      oraa     #%00000100
        anda     #%11110111
        tap
        jmp      fin
else2   tpa      oraa     #%00001000
        anda     #%11110111
        jmp      fin
else1   ldaa    MANTSGN2
        cmpa     #$00
        beq      condi3
        jmp      else3
condi3  ldx      #FPACC1EX
        ldy      #FPACC2EX
loop2   ldaa    0,%x
        cmpa     0,%y
        blo     cas12
        bhi     cas22
        beq     cas32
cas12   tpa      anda     #%11110111
        oraa     #%00001000
        tap
        jmp      fin
cas22   tpa      anda     #%11110011
        tap
        jmp      fin
cas32   inx      iny
        cpy      #MANTSGN2
        beq      egal2
        jmp      loop2
egal2   tpa      anda     #%11110111
        oraa     #%00000100
        tap
        jmp      fin
else3   tpa      anda     #%11110011
        tap
        clv
        pulx
        pulx
        rts
*
.negf   pshx     xdef     .negf
        tsx
        ldab     #$04
        abx
        ldaa     0,%x
        lsll
        bcs     neg%1
        sec
        jmp      neg%2
neg%1   clc
neg%2   rora
        staa     0,%x
        pulx
        pulx
        rts
*
.ftoi   pshx     xdef     .ftoi
        tsx
        ldab     #$04

```

```

abx
jsr   GTFPAC2C
jsr   FLT2INT
tsx
ldd   0,%x
std   4,%x
ldd   2,%x
std   6,%x
ldab  #$04
abx
txs
ldd   FPACC1MN+1
pulx
rts

*
.itof  pshx  xdef  .itof
psha
pshb
ldd   #$0000
std   FPACC1EX
std   FPACC1MN+1
staa  MANTSGN1
pulb
pula
std   FPACC1MN+1
tsx
dex
dex
dex
dex
ldd   4,%x
std   0,%x
ldd   6,%x
std   2,%x
tsx
jsr   SINT2FLT
tsx
ldab  #$04
abx
jsr   PTFPAC1C
pulx
rts

*
fpabs  pshx  xdef  fpabs
tsx
ldab  #$04
abx
ldaa  0,%x
lsla
bcc   abs%1
clc

abs%1  lsra  staa  0,%x
xgdx
pulx
rts

*
putfp  pshx  xdef  putfp
tsx
pshb
psha
ldab  #$04
abx
jsr   GTFPAC1C
pulx
jsr   FLTASC
pulx
rts

*
xdef  getfp

```

```

getfp    pshx
          xgdx
          ldaa    0,%x
          cmpa   #$20
          bne    gfp%1
          inx
gfp%1    jsr    ASCFLT
          tsx
          dex
          dex
          dex
          dex
          ldd    4,%x
          std    0,%x
          ldd    6,%x
          std    2,%x
          txs
          ldab   #$04
          abx
          jsr    PTFPAC1C
          xgdx
          pulx
          rts
*
send      xdef send
          pshx
          xgdx
          ldab   #$0e
send1     ldaa   0,%x
          staa   $a800
          inx
          decb
          bne    send1
          pulx
          rts
*
affich    pshx
          xdef affich
          xgdx
          inx
          ldaa   0,%x
          cmpa   #'.'
          bne    aff1
          inx
          ldaa   0,%x
aff1      staa   $a802
          staa   $a001
          inx
          ldaa   0,%x
          cmpa   #'.'
          bne    aff2
          inx
          ldaa   0,%x
aff2      staa   $a802
          staa   $a000
          inx
          ldaa   0,%x
          cmpa   #'.'
          bne    aff3
          inx
          ldaa   0,%x
aff3      staa   $a800
          staa   $a001
          inx
          ldaa   0,%x
          cmpa   #'.'
          bne    aff4
          inx
          ldaa   0,%x
aff4      staa   $a800
          staa   $a000
          pulx
          rts

```



```

*
      xdef lectad
lectad  ldaa #00
      staa $8004
lectad1 ldaa $8004
      lsla
      bcs lectad1
      ldab $8000
      ldaa $8004
      rts
*
      xdef trig
      trig
trig    pshx
      psha
      pshb
      ldaa #$01
      staa $a800
      ldx $ffff
      ldd #$00
trl1    mul
      dex
      bne trl1
      ldaa #$00
      staa $a800
      pulb
      pula
      pulx
      rts
*
      xdef attend
attend  pshx
      psha
      pshb
      ldx $ffff
      ldd #$00
att1    mul
      dex
      bne att1
      pulb
      pula
      pulx
      rts
*
      END

```

```

/*=====
Fichier de données: poids du réseau calculés
durant la phase d'apprentissage, matrices W2 et W3.
Codage hexadécimal des valeurs

```

```

NOM : POIDS1.S
DATE : août 1993
PAR : Sylvie Legendre

```

```

=====*/

```

```

DSCCT
XDEF W2
W2
FCB $41,$80,$EA,$41,$C1,$14,$52,$08,$41,$49,$18,$83,$C1,$A8,$2D,$8D
FCB $3F,$ED,$70,$2C,$C0,$C1,$AD,$BC,$40,$3C,$58,$40,$3F,$18,$1A,$5D
FCB $C0,$8C,$C2,$19,$C1,$3B,$01,$1F,$40,$33,$63,$4F,$41,$7D,$80,$64
FCB $C1,$20,$6F,$44,$C1,$7C,$6C,$CF,$C1,$F3,$94,$0E,$42,$89,$FD,$E7
FCB $BF,$16,$D6,$AD,$41,$38,$91,$08,$C1,$81,$66,$8B,$40,$B2,$F2,$2E
FCB $40,$8A,$A7,$AC,$40,$8A,$29,$4B,$40,$F5,$08,$8B,$C1,$94,$3C,$A1
FCB $C1,$06,$81,$F0,$C1,$11,$82,$B7,$C1,$70,$2B,$0E,$42,$26,$5B,$37
FCB $40,$73,$82,$81,$41,$56,$6F,$7D,$41,$03,$FA,$D3,$C1,$F5,$18,$99
FCB $C0,$AF,$88,$62,$41,$00,$66,$5D,$41,$95,$C6,$DA,$C2,$01,$DC,$10
FCB $3F,$6C,$6A,$F4,$C0,$3B,$11,$D8,$41,$23,$CA,$F1,$C1,$6C,$28,$DA
FCB $40,$8A,$34,$3E,$C0,$DE,$F7,$57,$C1,$FC,$A0,$30,$42,$2E,$C7,$86
FCB $3D,$85,$9C,$C4,$41,$8D,$E1,$9F,$41,$D2,$21,$CE,$C2,$80,$3D,$FE
FCB $40,$BC,$C6,$41,$BF,$4D,$69,$0E,$BF,$6A,$7F,$96,$C0,$E3,$8A,$F6
FCB $41,$3A,$8C,$D1,$41,$1F,$2F,$85,$41,$53,$B2,$DF,$C2,$1A,$6A,$2D
FCB $40,$F6,$73,$CE,$C1,$AE,$E9,$4A,$C1,$8C,$F7,$22,$42,$37,$FC,$74
FCB $C1,$3B,$45,$2D,$41,$18,$5A,$DE,$41,$86,$02,$86,$C1,$B4,$C8,$94
FCB $40,$F3,$20,$AC,$41,$53,$8E,$1E,$41,$27,$90,$88,$C2,$22,$FC,$33
FCB $C0,$30,$B2,$F7,$BF,$FB,$7F,$F3,$C0,$08,$D5,$28,$41,$04,$2B,$02
FCB $C1,$37,$B9,$45,$C0,$84,$A8,$81,$C1,$10,$A3,$96,$41,$E1,$CA,$E4
FCB $3D,$12,$FA,$0B,$C1,$42,$C8,$94,$C1,$2A,$4D,$50,$41,$EB,$A0,$E6
FCB $BF,$FC,$7B,$F4,$40,$82,$C7,$97,$C1,$0D,$EE,$A8,$40,$CC,$03,$14
FCB $41,$2A,$BC,$3A,$40,$ED,$31,$2D,$41,$76,$46,$F6,$C2,$21,$DE,$99
FCB $41,$23,$55,$46,$C0,$AA,$D3,$37,$C1,$C4,$70,$E9,$41,$D2,$00,$8C
FCB $40,$EF,$80,$0C,$41,$6C,$EA,$91,$41,$69,$19,$8F,$C2,$1B,$61,$03
FCB $41,$39,$36,$96,$40,$92,$20,$58,$C1,$C7,$04,$E2,$41,$29,$91,$D6
FCB $C1,$58,$1D,$A6,$3F,$D5,$06,$4E,$41,$19,$15,$BB,$40,$92,$13,$2A
FCB $C0,$D6,$D6,$B3,$C1,$19,$7A,$7B,$C0,$80,$0D,$14,$41,$B7,$97,$69
FCB $40,$74,$69,$84,$41,$06,$46,$30,$42,$6F,$66,$79,$C2,$AE,$19,$3D
FCB $40,$08,$2B,$DE,$40,$DD,$A2,$F0,$C2,$27,$F6,$7A,$42,$09,$91,$62
FCB $3F,$82,$96,$F8,$C0,$F3,$5A,$93,$C1,$CC,$C5,$D6,$42,$1E,$F1,$5B
FCB $C0,$00,$AB,$C2,$41,$28,$E3,$DD,$41,$93,$9B,$C6,$C2,$03,$AD,$C7
FCB $41,$15,$2F,$3A,$C0,$EA,$92,$38,$41,$22,$E8,$F6,$C1,$87,$A0,$7C
FCB $40,$17,$38,$3E,$40,$0A,$7B,$70,$40,$F5,$A2,$D0,$C1,$85,$6D,$F5
FCB $41,$25,$1C,$DE,$41,$94,$11,$99,$41,$86,$BE,$3A,$C2,$5A,$B7,$6B
FCB $3F,$1D,$8E,$05,$40,$4E,$14,$CC,$C1,$CA,$44,$D9,$41,$EA,$4A,$19
FCB $C0,$CE,$E0,$BC,$BF,$4A,$EE,$36,$40,$EA,$C0,$58,$C0,$0B,$D0,$1E
FCB $40,$BD,$F2,$8C,$C0,$D5,$1D,$44,$C1,$AD,$4A,$AD,$41,$C7,$9F,$65
FCB $3F,$80,$B6,$F9,$41,$04,$BB,$F1,$40,$A3,$7B,$3E,$C1,$92,$26,$55
FCB $C0,$E1,$36,$39,$C0,$AE,$9F,$12,$C1,$48,$F4,$1E,$42,$07,$78,$22
FCB $C1,$62,$9A,$F4,$40,$ED,$DA,$0E,$41,$F6,$77,$D5,$C1,$DA,$32,$CF
FCB $C1,$35,$66,$DF,$C1,$13,$5E,$2B,$41,$8F,$9D,$8A,$40,$5B,$24,$3C
FCB $C0,$F9,$A6,$D8,$C0,$C8,$F5,$27,$BF,$1B,$EE,$C3,$41,$85,$64,$C6
FCB $C1,$4C,$B5,$FE,$C0,$A0,$F3,$57,$C1,$5B,$F1,$07,$42,$0B,$47,$C1
FCB $C0,$92,$7A,$4F,$40,$E0,$26,$8A,$40,$D4,$D2,$77,$C1,$77,$02,$09
FCB $C0,$41,$F2,$09,$C0,$06,$7F,$06,$41,$85,$28,$F4,$C1,$63,$4C,$84
FCB $BF,$97,$9A,$8E,$C0,$E6,$85,$F2,$C1,$2F,$1C,$E3,$41,$9E,$CB,$4E
FCB $C0,$72,$9C,$8D,$41,$20,$5A,$22,$41,$1F,$95,$CD,$C1,$9B,$AE,$AF
FCB $BF,$EC,$35,$15,$40,$91,$80,$8A,$41,$3F,$1B,$3E,$C1,$8D,$3A,$34
FCB $40,$D9,$BF,$0C,$41,$52,$A7,$4D,$41,$F7,$3D,$CC,$C2,$6F,$C5,$77
FCB $C0,$AA,$EE,$F5,$41,$13,$72,$66,$42,$25,$BF,$D2,$C2,$56,$1D,$12
FCB $C0,$DD,$68,$1F,$41,$12,$EC,$45,$41,$31,$A7,$4C,$C1,$83,$66,$29
FCB $40,$C3,$25,$A2,$BF,$97,$5C,$34,$C1,$FE,$6F,$25,$42,$12,$06,$EC
FCB $C1,$06,$1C,$A0,$C0,$83,$6F,$D6,$41,$D1,$5D,$44,$C1,$8B,$86,$2E
FCB $C0,$C3,$28,$80,$41,$2A,$63,$8F,$42,$2A,$9B,$42,$66,$6C,$6E
FCB $40,$D8,$B0,$69,$41,$0B,$5B,$8E,$C0,$4A,$4A,$1B,$C1,$1C,$BD,$5C
FCB $41,$A0,$B3,$EB,$C1,$0B,$C5,$FE,$3E,$28,$6B,$D8,$C1,$4C,$04,$DD
FCB $C1,$9C,$95,$71,$C0,$BC,$A0,$58,$C0,$FD,$63,$06,$42,$14,$8A,$DD
FCB $40,$36,$0E,$8C,$C0,$79,$34,$CD,$41,$87,$03,$2A,$C1,$DE,$2C,$16
FCB $40,$EC,$54,$82,$BE,$1B,$DE,$B2,$40,$5F,$48,$64,$C1,$2C,$ED,$FF
FCB $40,$DA,$3B,$E2,$40,$52,$51,$94,$C0,$7C,$C1,$D7,$C0,$89,$85,$6B
XDEF W3

```

W3   FCB \$BF,\$D7,\$C0,\$AA,\$40,\$21,\$EF,\$59,\$40,\$F1,\$D1,\$0C,\$BF,\$31,\$92,\$8B  
       FCB \$40,\$AD,\$16,\$7C,\$C0,\$50,\$89,\$F6,\$41,\$81,\$EC,\$FF,\$BE,\$80,\$25,\$55  
       FCB \$C0,\$8B,\$19,\$7E,\$3F,\$E1,\$0B,\$EB,\$C1,\$23,\$E4,\$6C,\$BF,\$8D,\$2B,\$8F  
       FCB \$41,\$9D,\$CA,\$8E,\$C0,\$D8,\$66,\$DA,\$40,\$43,\$84,\$27,\$C0,\$95,\$C1,\$51  
       FCB \$41,\$4C,\$64,\$05,\$40,\$A0,\$01,\$EC,\$C1,\$71,\$61,\$7B,\$C0,\$D2,\$59,\$F9  
       FCB \$40,\$38,\$BC,\$2F,\$BE,\$7F,\$F7,\$79,\$40,\$EC,\$2B,\$70,\$3F,\$D5,\$DA,\$54  
       FCB \$BF,\$EF,\$99,\$9D,\$C0,\$AE,\$BA,\$3C,\$C0,\$ED,\$10,\$B4,\$3E,\$8A,\$81,\$E6  
       FCB \$40,\$4C,\$5D,\$44,\$41,\$35,\$8C,\$4A,\$BF,\$A5,\$21,\$60,\$40,\$40,\$3A,\$90  
       FCB \$40,\$F4,\$2F,\$3A,\$40,\$A0,\$C1,\$5E,\$40,\$8A,\$97,\$B2,\$41,\$79,\$6B,\$F7  
       FCB \$C1,\$31,\$1C,\$0B,\$C1,\$87,\$6F,\$20,\$C0,\$65,\$8B,\$71,\$3F,\$DE,\$65,\$D3  
       FCB \$C0,\$1F,\$00,\$B0,\$C0,\$82,\$87,\$2F,\$C0,\$6F,\$E5,\$0C,\$41,\$C8,\$88,\$56  
       FCB \$41,\$A3,\$5A,\$5C,\$41,\$44,\$9C,\$AE,\$40,\$80,\$E3,\$F3,\$BE,\$8E,\$7A,\$BF  
       FCB \$C0,\$1A,\$21,\$4A,\$40,\$11,\$92,\$0B,\$40,\$88,\$64,\$4D,\$40,\$6F,\$A1,\$40  
       FCB \$C0,\$98,\$B4,\$A7,\$C0,\$1C,\$98,\$8E,\$C1,\$CB,\$6C,\$A4,\$3F,\$8A,\$D4,\$DB  
       FCB \$40,\$B1,\$8D,\$DD,\$3F,\$95,\$39,\$2C,\$C1,\$A3,\$28,\$F0,\$41,\$4B,\$F0,\$63  
       END