# Sélection de modèles non linéaires par "leave-one-out": étude théorique et application des réseaux de neurones au procédé de soudage par points

Gaétan Monari

# THÈSE DE DOCTORAT DE L'UNIVERSITÉ PARIS 6

Spécialité

ÉLECTRONIQUE

présentée

par **Gaétan MONARI**

pour obtenir le titre de **DOCTEUR de l'UNIVERSITÉ PARIS 6**

Sujet de la thèse :

## Sélection de modèles non linéaires par leave-one-out

## Etude théorique et application des réseaux de neurones au procédé de soudage par points

Soutenue le 3 novembre 1999

devant le jury composé de :

| | | |
|---|---|---|
| Mme | S. THIRIA | Rapporteur |
| M. | S. CANU | Rapporteur |
| Mme | B. DORIZZI | Examinateur |
| M. | O. DIERAERT | Examinateur |
| M. | G. DREYFUS | Examinateur |
| M. | J. FRIGIÈRE | Examinateur |
| M. | P. GALLINARI | Président |
| M. | G. PRADÈRE | Invité |

# REMERCIEMENTS

*Le travail de recherche que décrit ce mémoire a été effectué dans le cadre d'une convention CIFRE entre la société SOLLAC (Groupe USINOR) et le Laboratoire d'Électronique de l'École Supérieure de Physique et de Chimie Industrielles de la Ville de Paris (ESPCI).*

*Ce projet n'aurait jamais vu le jour sans Monsieur Francis SAUVAGE - directeur du Centre de Recherche et de Développement Métallurgiques (CRDM) de SOLLAC-Dunkerque - dont l'intérêt pour les techniques d'ingénierie de la connaissance n'est plus à démontrer, et qui a su faire le pari d'une combinaison réussie entre réseaux de neurones et soudage par points. Monsieur Hugues OBERLÉ - ancien responsable du Service Soudage par Résistance du CRDM – s'est également approprié ce défi et, tant par sa connaissance approfondie du procédé que sa formidable capacité d'analyse, a pu orienter judicieusement mes travaux pendant la première moitié de ma thèse. Qu'ils trouvent ici l'expression de ma plus sincère gratitude.*

*Monsieur le Professeur Gérard DREYFUS, directeur du laboratoire d'Électronique de l'ESPCI, a assuré avec rigueur et disponibilité l'encadrement universitaire de cette thèse. Sans a priori particulier, il a su concilier son intérêt pour les questions fondamentales avec les exigences très pratiques de la modélisation du soudage par points. Je tiens à le remercier, tant pour sa contribution au développement et à la présentation des idées décrites dans ce mémoire, que pour son soutien et pour la confiance qu'il m'a témoignée tout au long de ces trois dernières années.*

*Je suis extrêmement reconnaissant à Monsieur Olivier DIERAERT, responsable du Service Soudage par Résistance du CRDM, d'avoir repris "au vol" le suivi de cette étude, bien que découvrant à la fois les réseaux de neurones et le soudage par points. Il m'a permis de poursuivre mes travaux de manière autonome et s'est toujours impliqué avec pertinence dès que la situation l'exigeait.*

*Madame Sylvie THIRIA, Professeur à l'Université de Versailles et Monsieur Stéphane CANU, Professeur à l'INSA de Rouen ont accepté de consacrer une partie de leur temps à la lecture de mon manuscrit et à la rédaction d'un rapport sur mes travaux. Je les remercie d'autant plus d'avoir accepté cette mission que leurs commentaires ont sans aucun doute contribué à clarifier mes idées et à préciser la validité de certains de mes résultats.*

*Monsieur Joël FRIGIÈRE, de la Direction de Informatique Scientifique et Avancée d'USINOR a contribué à la diversité de mon jury de thèse en y apportant sa grande expérience de l'intégration en milieu industriel des techniques d'Ingénierie des Connaissances. Je lui en suis très reconnaissant.*

*J'adresse mes plus vifs remerciements à Madame Bernadette DORIZZI, Professeur à l'INT d'EVRY, et Monsieur Patrick GALLINARI, Professeur à l'Université PARIS 6, qui ont accepté de faire partie de mon jury de thèse.*

*Je suis très sensible à l'honneur que m'a fait Monsieur Gérard PRADÈRE, ingénieur soudeur chez RENAULT, en acceptant mon invitation.*

*Je tiens à remercier Isabelle RIVALS et Léon PERSONNAZ, maîtres de conférences au laboratoire d'Électronique de l'ESPCI, pour avoir attiré mon attention sur l'utilisation du développement de Taylor, et notamment pour avoir démontré analytiquement, à partir d'une relation existant dans la littérature (formule 3.8), la relation (3.9) que j'avais établie empiriquement à partir de simulations.*

*André ELISSEEFF, étudiant en thèse à l'Université de Lyon 2, m'a permis de situer mes travaux par rapport aux "Vapnikeries" et autres études théoriques sur la validité de l'estimation des performances en généralisation d'un modèle. Je lui suis grandement reconnaissant pour la pertinence de ces remarques et pour sa disponibilité.*

*Je tiens par ailleurs à remercier l'équipe du laboratoire d'Électronique de l'ESPCI : Frédérique MONCET, Brigitte QUENET, Isabelle RIVALS, Yacine OUSSAR, Léon PERSONNAZ, Jean-Luc PLOIX, Pierre ROUSSEL, Mathieu STRICKER, sans oublier Madame DROUODE, pour m'avoir accueilli, conseillé et soutenu pendant ces trois années.*

*De même, j'adresse un grand merci à tout le service Soudage par Résistance du CRDM en réservant une mention particulière à Anne CLAD et Thomas DUPUY, qui ont bien voulu me faire profiter de leur expertise en soudage par points, et qui ont indéniablement contribué au développement et à la rédaction des idées présentées dans ce document.*

*Anne, dont je partage l'existence avec bonheur depuis bientôt dix ans, m'a considérablement soutenu tout au long de ce projet, depuis son origine jusqu'à la rédaction du manuscrit final. Je la remercie du fond du cœur pour son soutien, sa disponibilité et sa patience.*

*Je n'aurais sans doute pas pu me concentrer uniquement sur mes travaux de recherche sans le concours d'un certain nombre de personnes du CRDM, et en premier lieu sans Marie-Claude SMOCH, Marie-Noëlle SWAL et Christine DEGROOTE. Je me permets donc de leur adresser l'expression de ma plus profonde affection.*

*Enfin j'aimerais associer à ce moment particulier de ma vie mes parents, mes frères et tous mes amis…*

## TABLE DES MATIERES


INTRODUCTION 1

1 INTRODUCTION AUX RÉSEAUX DE NEURONES 3

2 GÉNÉRALISATION ET ESTIMATION DES PERFORMANCES 13

3 ETUDE THÉORIQUE DU LEAVE-ONE-OUT 31

4 UTILISATION DU LEAVE-ONE-OUT POUR LA SÉLECTION DE MODÈLES 47

5 UN NOUVEL ALGORITHME D'APPRENTISSAGE 65

6 INTRODUCTION AU SOUDAGE PAR POINTS 75

7 LA MODÉLISATION DU SOUDAGE PAR POINTS 87

8 DÉVELOPPEMENT D'UN MODÈLE NEURONAL DU SOUDAGE PAR POINTS 105

CONCLUSION 131

RÉFÉRENCES BIBLIOGRAPHIQUES 133

ANNEXE 1 : CALCUL DES $h_{ii}$ 141

ANNEXE 2 : SURAJUSTEMENT ET REDONDANCE DE COEFFICIENTS 143

ANNEXE 3 : DÉMONSTRATION DE LA RELATION (3.8) 147

ANNEXE 4 : PROCÉDÉ D'ORTHONORMALISATION DE GRAM-SCHMIDT 149

ANNEXE 5 : CARACTÉRISTIQUES DES PRODUITS UTILISÉS 151

## TABLE DES MATIERES DETAILLEE


**INTRODUCTION** **1**

**1 INTRODUCTION AUX RÉSEAUX DE NEURONES** **3**

**Résumé** **3**
**1.1 Introduction** **3**
1.1.1 Les neurones 3
1.1.2 Les réseaux de neurones non bouclés 5
**1.2 Propriété fondamentale des réseaux de neurones non bouclés** **7**
1.2.1 L'approximation universelle 7
1.2.2 La parcimonie 8
1.2.3 De l'approximation de fonction à la modélisation statistique 8
**1.3 Mise en œuvre des réseaux de neurones** **9**
1.3.1 la fonction de coût 10
1.3.2 Le calcul du gradient 10
1.3.3 L'algorithme d'optimisation 10

**2 GÉNÉRALISATION ET ESTIMATION DES PERFORMANCES** **13**

**Résumé** **13**
**2.1 Introduction** **14**
**2.2 Le dilemme biais / variance** **14**
**2.3 La validation croisée** **16**
**2.4 La régularisation** **18**
2.4.1 Early stopping 18
2.4.2 Pénalisation de la fonction de coût (weight decay) 19
**2.5 Le surajustement** **20**
2.5.1 Discussion : qu'est-ce que le surajustement ? 20
2.5.2 Détection du surajustement 21
**2.6 Les intervalles de confiance** **23**
2.6.1 Introduction 23
2.6.2 Différence entre performance du modèle et intervalle de confiance ? 24
2.6.3 Comment interpréter les intervalles de confiance ? 27
**2.7 Bornes sur les performances de généralisation** **29**

**3 ETUDE THÉORIQUE DU LEAVE-ONE-OUT** **31**

**Résumé** **31**
**3.1 Introduction** **31**
**3.2 Approximation locale de la solution des moindres carrés** **32**
**3.3 Effet du retrait d'un exemple de l'ensemble d'apprentissage** **33**
3.3.1 Effet du retrait d'un exemple sur sa prédiction 34
3.3.2 Effet du retrait d'un exemple sur l'intervalle de confiance de sa prédiction 35

3.3.3 Interprétation des $h_{ii}$ 36
**3.4 Validation des résultats de leave-one-out** **38**
3.4.1 Interprétation géométrique de l'estimation des performances en leave-one-out 39
3.4.2 Limite de l'approche : cas du retrait d'un exemple avec forte influence 42
**3.5 Conclusion** **46**

**4 UTILISATION DU LEAVE-ONE-OUT POUR LA SÉLECTION DE MODÈLES** **47**

**Résumé** **47**
**4.1 Introduction - définition du problème** **47**
**4.2 Sélection de modèle sur la base des performances d'apprentissage (pour une architecture donnée)** **49**
4.2.1 1ère méthode : choisir le modèle pour lequel l'EQMA est minimale 50
4.2.2 2ème méthode : choisir un "minimum de rang plein" de la fonction de coût 52
4.2.3 Conclusion 55
**4.3 Sélection de modèle sur la base de $E_a$ (pour une architecture donnée)** **56**
**4.4 Sélection des minima sur la base de $E_p$ (pour une architecture donnée)** **57**
4.4.1 Qualité de la sélection 57
4.4.2 Qualité de l'estimation des performances de généralisation 58
**4.5 Sélection de l'architecture optimale** **60**
4.5.1 Utilisation des intervalles de confiance 61
4.5.2 Amélioration progressive des modèles 63
**4.6 Conclusion** **64**

**5 UN NOUVEL ALGORITHME D'APPRENTISSAGE** **65**

**Résumé** **65**
**5.1 Introduction** **65**
**5.2 Algorithme pour la minimisation de $J^*$** **66**
5.2.1 Calculs du coût et du gradient 66
5.2.3 Modification des coefficients 67
*5.2.3.a Rappel : l'algorithme de Levenberg-Marquardt* *67*
*5.2.3.b Adaptation à la minimisation de $J^*$* *68*
5.2.4 En résumé 69
**5.3 Mise en œuvre de l'algorithme** **70**
5.3.1 Etude des contraintes 70
5.3.2 Mise en œuvre de l'algorithme 71
**5.4 Conclusion** **73**

**6 INTRODUCTION AU SOUDAGE PAR POINTS** **75**

**Résumé** **75**
**6.1 Généralités** **75**
6.1.1 Principe 76
6.1.2 Déroulement du cycle de soudage 76
6.1.3 Paramètres de soudage 77
6.1.4 Mécanisme de formation de la soudure 78

6.1.4 Géométrie d'un point soudé 80
**6.2 Caractérisation d'une tôle d'acier revêtues 80**
6.2.1 Le domaine de soudabilité 81
6.2.2 La dégradation des électrodes 83
**6.3 Conclusion 85**

**7 LA MODÉLISATION DU SOUDAGE PAR POINTS 87**

**Résumé 87**
**7.1 Introduction 87**
**7.2 Modélisation du soudage par points 88**
7.2.1 La soudabilité d'une tôle 88
7.2.2 Le point soudé 90
7.2.3 La commande du soudage par points 93
7.2.4 Conclusion 94
**7.3 Caractéristiques de la qualité de la soudure 94**
7.3.1 Introduction 95
7.3.2 Les signaux électriques 95
7.3.3 Les signaux mécaniques 100
7.3.4 Conclusion 104

**8 DÉVELOPPEMENT D'UN MODÈLE NEURONAL DU SOUDAGE PAR POINTS 105**

**Résumé 105**
**8.1 Introduction 106**
**8.2 Enjeux de la modélisation 107**
8.2.1 Domaine de validité souhaité 107
8.2.2 Incertitude sur la mesure du diamètre de bouton 109
**8.3 Principe de la modélisation 111**
8.3.1 Base d'apprentissage initiale 112
8.3.2 Sélection des entrées 113
8.3.3 Modélisation et utilisation des intervalles de confiance 115
8.3.4 Conclusion 116
**8.4 Application 116**
8.4.1 Contrôle de la qualité des soudures 116
*8.4.1.a Cas du produit GA 117*
*8.4.1.b Cas du produit GZ 2 120*
*8.4.1.c Conclusion et perspectives 122*
8.4.2 Dégradation des électrodes 124
*8.4.2.a Loi de commande utilisée 124*
*8.4.2.b Application aux produits GA et GZ 2 125*
*8.4.2.c Conclusion 127*
**8.5 Conclusion - perspectives industrielles 128**
8.5.1 Conclusion 128
8.5.2 Perspectives industrielles 129

**CONCLUSION** **131**

**RÉFÉRENCES BIBLIOGRAPHIQUES** **133**

**ANNEXE 1 : CALCUL DES $h_{ii}$** **141**

**ANNEXE 2 : SURAJUSTEMENT ET REDONDANCE DE COEFFICIENTS** **143**

**ANNEXE 3 : DÉMONSTRATION DE LA RELATION (3.8)** **147**

**ANNEXE 4 : PROCÉDÉ D'ORTHONORMALISATION DE GRAM-SCHMIDT** **149**

**ANNEXE 5 : CARACTÉRISTIQUES DES PRODUITS UTILISÉS** **151**

# INTRODUCTION

Bien qu'il soit universellement utilisé pour l'assemblage des carrosseries automobiles, le procédé de soudage par points n'est pas encore complètement maîtrisé. En effet, il n'existe pas de méthode fiable pour contrôler de manière non destructive la qualité de soudures, pourtant effectuées à réglages constants : la variabilité du procédé, et en particulier de l'état des électrodes de soudage, est telle que cette qualité change dans des proportions inacceptables.

D'un point de vue pratique, le manque de garantie sur la qualité de chaque soudure oblige à réaliser un nombre excessif de points soudés par pièce. Par ailleurs, la seule méthode dont disposent les industriels pour augmenter leur confiance en la qualité des soudures, est d'utiliser des intensités de soudage significativement plus élevées que nécessaire, ce qui accélère le processus de dégradation des électrodes, et conduit à des arrêts fréquents de la production.

Compte tenu du nombre de phénomènes physiques qui interviennent lors d'une soudure par point, de la rapidité et du caractère transitoire de ces phénomènes, il est exclu d'aborder ce problème par une approche utilisant les techniques de modélisation numérique. En revanche, des travaux récents montrent qu'il peut être avantageux d'avoir recours à des méthodes statistiques. Après examen, la non-linéarité du problème et le nombre de paramètres entrant en jeu font des réseaux de neurones d'excellents candidats à la résolution de cette question : avec des propriétés mathématiques bien établies et des algorithmes d'apprentissage très performants, les réseaux de neurones surpassent, lorsqu'ils sont convenablement mis en œuvre, les techniques classiques de modélisation non linéaire.

L'une des principales difficultés rencontrées par les utilisateurs de modèles statistiques non linéaires est le dimensionnement correct des modèles, de manière à ce qu'ils présentent des performances aussi bonnes lors de leur utilisation (sur une base de test) que lors de leur ajustement (sur une base d'apprentissage). L'une des méthodes utilisées pour procéder à la sélection de modèles est connue sous le nom de *leave-one-out*. Il s'agit d'estimer les performances de généralisation d'un modèle à partir des erreurs de prédiction commises sur un exemple lorsque celui-ci ne fait pas partie de la base d'apprentissage.

La première partie de ce mémoire, qui comprend 5 chapitres, est consacrée à l'étude théorique du *leave-one-out*, au cours de laquelle nous montrerons les limites de la mise en œuvre classique de cette méthode, puis nous définirons une manière plus fiable et plus rapide de procéder. Nous montrerons également la relation existant entre le *leave-one-out* et les intervalles de confiance sur la sortie d'un modèle.

La deuxième partie de ce mémoire, c'est-à-dire les chapitres 6 à 8, a pour but d'exposer les applications de l'étude théorique au contrôle de qualité et à la commande du procédé de soudage par points.

Les définitions relatives aux réseaux de neurones sont présentées dans le chapitre 1, ainsi que la propriété mathématique fondamentale des réseaux à une couche de neurones cachés sigmoïdaux, c'est-à-dire l'approximation parcimonieuse. Le principe de la mise en œuvre des réseaux de neurones par "apprentissage" est ensuite décrit.

Le chapitre 2 débute par une revue des deux principales méthodes employées pour faire face au dilemme biais / variance : la validation croisée, dont le leave-one-out est un cas particulier, et la régularisation. Nous introduisons ensuite les notions de surajustement et d'intervalles de confiance associés à la sortie d'un modèle, notions qu'il est indispensable de maîtriser lors de la mise en œuvre de modèles non linéaires paramétrés.

Dans le chapitre 3, nous voyons qu'il est possible, grâce à l'utilisation d'un développement de Taylor, de calculer l'effet, sur le modèle, du retrait d'un exemple de la base d'apprentissage et nous introduisons la notion d'influence d'un exemple sur l'estimation des paramètres d'un modèle. Dans certains cas, nous montrons pourquoi la méthode classique, par minimisation du coût quadratique, ne permet pas d'envisager raisonnablement le retrait d'un exemple de la base d'apprentissage.

La sélection de modèles sur le principe du leave-one-out est examinée en détail dans le chapitre 4. Nous prouvons que l'estimation des performances de généralisation d'un modèle, calculée d'après les résultats du chapitre 3, est à la fois plus fiable et plus rapide que celle qui est obtenue par la mise en œuvre traditionnelle du leave-one-out. Par ailleurs, nous définissons un critère de sélection de modèles à partir des intervalles de confiance sur la prédiction des exemples d'apprentissage.

Le chapitre 5, qui clôt la première partie de ce mémoire, présente un nouvel algorithme d'apprentissage fondé sur une fonction de coût définie d'après les résultats du chapitre 4. Nous montrons les perspectives qu'offre cet algorithme lorsqu'il est mis en œuvre après convergence de la minimisation classique du coût quadratique.

Les principes fondamentaux du soudage par points sont présentés dans le chapitre 6. Nous insistons plus particulièrement sur les notions de *domaine de soudabilité* d'un produit et de *durée de vie* des électrodes.

Le chapitre 7 débute par un exposé des différentes possibilités d'envisager la modélisation du soudage par points. Nous montrons que la prédiction de la qualité d'une soudure à partir des conditions opératoires nécessite la mesure, pendant le processus de soudage, de grandeurs pertinentes caractérisant le développement du noyau fondu. La deuxième moitié de ce chapitre est consacrée à l'étude de 4 signaux acquis lors d'une soudure, et à la définition de caractéristiques de la qualité du point.

Le chapitre 8 est divisé en deux parties bien distinctes :

- à partir des spécificités du procédé de soudage par points et des outils présentés dans les 5 premiers chapitres de ce mémoire, nous proposons une méthodologie précise permettant de construire et d'améliorer progressivement un modèle de prédiction du diamètre de bouton valable pour un produit donné dans un domaine de validité choisi,
- l'application de cette procédure sur deux types de tôles est ensuite présentée. Après avoir donné les performances des modèles ainsi obtenus, nous examinons l'impact de ces modèles, sur le processus de dégradation des électrodes, lorsque ceux-ci sont mis en œuvre dans le cadre d'une loi de commande optimisée de l'intensité de soudage.

# 2 GENERALISATION ET ESTIMATION DES PERFORMANCES

## Résumé

*Dans la pratique, l'objectif d'une modélisation statistique n'est pas d'ajuster finement un modèle sur un ensemble d'apprentissage, mais d'obtenir un bon compromis entre performances d'apprentissage et performances de généralisation. Il existe principalement deux manières de résoudre ce dilemme entre le biais et la variance de la famille de fonctions considérée :*

- *les méthodes de validation croisée, qui scindent la base de données disponibles de manière à estimer les performances de généralisation du modèle sur des données n'ayant pas servi à l'apprentissage, ce qui permet, a posteriori, d'éliminer les solutions surajustées,*
- *la régularisation qui, par arrêt prématuré de l'apprentissage (early-stopping), ou par ajout d'un terme de pénalisation à la fonction de coût (weight decay), permet de pénaliser a priori les modèles à forte variance.*

*En réalité, le surajustement se traduit par une influence trop importante de certains exemples sur l'estimation des coefficients du modèle, qui peut ainsi s'ajuster très précisément à ces exemples. Nous allons étudier dans ce mémoire, par une approche théorique du leave-one-out, une manière de résoudre à la base ce phénomène de surajustement, que la validation croisée classique et la régularisation traitent de manière indirecte.*

*Un cas de surajustement particulièrement simple à détecter concerne les modèles pour lesquels la matrice jacobienne Z n'est pas de rang plein. Nous nous proposons de le détecter numériquement en vérifiant les propriétés que doivent respecter les termes diagonaux $h_{ii}$ de la matrice de projection $Z\ ({}^tZZ)^{-1}\ {}^tZ$, et nous montrons qu'il correspond à des modèles pour lesquels certains coefficients sont sous-déterminés.*

*Par ailleurs, nous présentons la notion d'intervalle de confiance sur la sortie du modèle. Nous utilisons dans ce mémoire une expression classique de cet intervalle, fondée sur un développement de Taylor de la sortie du modèle au voisinage de la solution des moindres carrés. Par opposition à la performance de généralisation du modèle, qui peut s'interpréter comme un intervalle de confiance associé à la mesure de la sortie du processus, les intervalles de confiance sur la sortie du modèle dépendent des entrées de ce dernier. Ils permettent ainsi de déterminer les zones de l'espace des entrées où - par manque d'exemples - la confiance sur la prédiction du modèle est trop faible.*

*En revanche, la détermination d'intervalles de confiance (ou de bornes) sur l'erreur de généralisation empirique, par rapport à l'erreur de généralisation théorique, fait depuis plusieurs années l'objet de recherches sur la théorie de l'apprentissage. Cependant, compte tenu du cadre de cette thèse, ces bornes ne sont pas exploitables pour l'instant.*

## 2.1 Introduction

La propriété d'approximation présentée dans le chapitre précédent soulève, en corollaire, une difficulté bien connue des utilisateurs de réseaux de neurones (ou de tout modèle paramétré) : le dimensionnement des modèles.

Lorsque l'on doit traiter des données bruitées, ce que nous supposerons tout au long de cette thèse, l'objectif est de trouver le modèle optimal, présentant le meilleur compromis possible entre performance d'apprentissage et capacité de généralisation.

Après une introduction à la problématique et une description des principales méthodes utilisées pour y faire face, nous présenterons une discussion sur la nature exacte du phénomène de surajustement.

Nous introduirons ensuite la notion d'intervalle de confiance associé, d'une part, à la sortie d'un modèle, et, d'autre part à l'estimation des performances de généralisation de ce modèle.

## 2.2 Le dilemme biais / variance

Ce compromis a été formalisé en décomposant la performance moyenne d'un modèle - sur toutes les bases d'apprentissage possibles - en deux parties [Geman 92] : la première, appelée **biais**, rend compte de la différence moyenne entre le modèle et l'espérance mathématique de la grandeur à modéliser ; la seconde, appelée **variance**, reflète l'influence du choix de la base d'apprentissage sur le modèle.

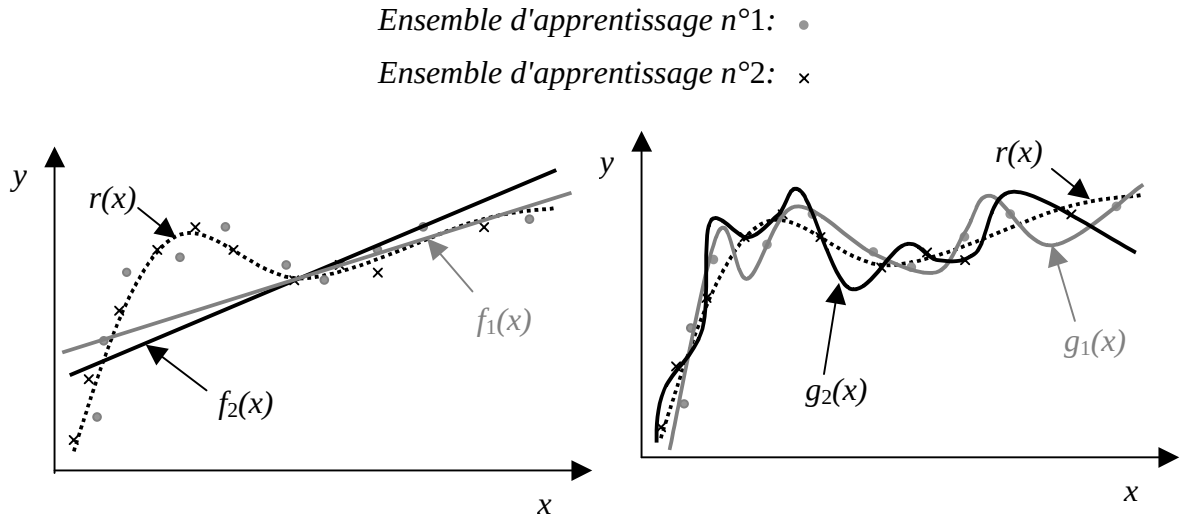


*Figure 2.1 : Exemples de familles de fonctions présentant - à gauche - un biais trop élevé - à droite - une variance trop élevée*

Sur la figure 2.1, considérons le problème de l'approximation d'une fonction $r(x)$ supposée inconnue, à partir de mesures bruitées de celle-ci. La famille des fonctions affines (notée $f$) présente un écart important avec le modèle idéal ; cependant cet écart dépend peu de la base d'apprentissage : cette famille de fonction possède donc un biais important et une variance faible. Pour la famille de fonction $g$, le phénomène inverse se produit : celle-ci est

suffisamment complexe pour s'ajuster au plus près aux points d'apprentissages ; en revanche sa forme varie beaucoup en fonction de ces derniers.

Idéalement, il faut chercher un modèle présentant un biais et une variance aussi faibles que possible. Un bon modèle doit donner ainsi une réponse moyenne satisfaisante tout en dépendant le moins possible des exemples dont on s'est servi pour le concevoir. Dans le cas de l'exemple précédent, la famille de fonctions *h* représentée sur la figure 2.2 réalise très bien ce compromis.

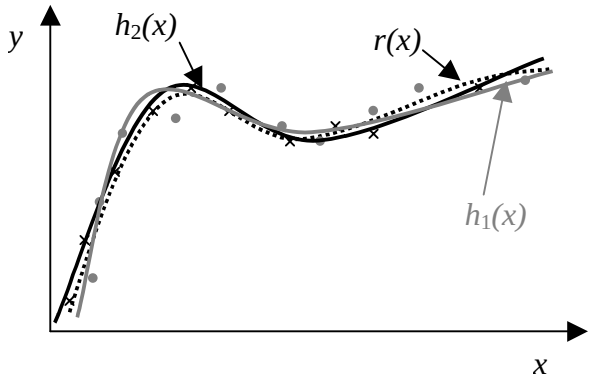


*Figure 2.2 : Exemple de famille de fonctions réalisant un bon compromis entre biais et variance*

Ce compromis peut certes s'obtenir en augmentant la taille de la base d'apprentissage, mais ce n'est malheureusement pas toujours possible. Dans la pratique, il existe principalement deux façons d'éviter le surajustement (cf. [Gallinari 97]), lorsqu'on dispose d'une base de donnée limitée :

- a posteriori, c'est-à-dire après apprentissage : le surajustement se détecte alors sur la base d'une estimation des performances de généralisation du modèle. La principale méthode utilisée dans le domaine des réseaux de neurones[2] est la validation croisée (voir [Stone 74]), fondée sur un ré-échantillonnage de la base de données,
- a priori, c'est-à-dire en cours d'apprentissage (voire avant celui-ci) : il s'agit des techniques de régularisation, qui visent à pénaliser l'obtention de modèles surajustés, mais qui ne dispensent pas de l'étape d'estimation des performances du modèle.

Dans ce chapitre comme dans les suivants, la question de la sélection de modèle n'est abordée que dans le sens de la sélection de l'architecture optimale, c'est-à-dire de la complexité de la famille de fonctions utilisée. Le problème de sélection des entrées n'est pas traité ici ; le lecteur intéressé pourra pour cela se référer à [Stoppiglia 97].

On suppose généralement que le processus à modéliser comporte plusieurs entrées non bruitées et une sortie bruitée.

---

[2] D'autres méthodes, moins utilisées, existent pour comparer différents modèles entre eux, par exemples les tests d'hypothèses ou les critères d'information (voir [Anders 99]).

Enfin, il est également important de noter que la question de la sélection de l'architecture optimale est étroitement liée à celle de l'estimation des performances de généralisation du modèle : idéalement, ces deux étapes devraient être effectuées simultanément, de manière à comparer des architectures entre elles sur la base de l'estimation des performances de généralisation.

En revanche, il peut être utile - dans certains cas - de séparer la sélection de la taille optimale du modèle, de l'apprentissage du modèle final. Pour ce faire, les notions de bases d'apprentissage et de validation concernent l'étape de sélection de la taille optimale. Dans un second temps, ces deux bases sont regroupées en une seule base d'apprentissage servant à concevoir le modèle final. Dans tous les cas, il est utile de disposer d'une base de test indépendante dont on se sert à la fin pour vérifier la validité et estimer les performances du modèle.

## 2.3 La validation croisée

Cette méthode repose sur une estimation des performances à partir d'exemples n'ayant pas servi à la conception du modèle. Pour ce faire, on scinde la base d'apprentissage en *D* parties de taille (approximativement) égale. On réalise alors *D* apprentissages du modèle, en laissant à chaque fois une des parties de côté pour le valider (cf. figure 2.3, tirée de [Bishop 97]). La performance du modèle s'obtient à partir des erreurs de validation constatées après les D apprentissages.

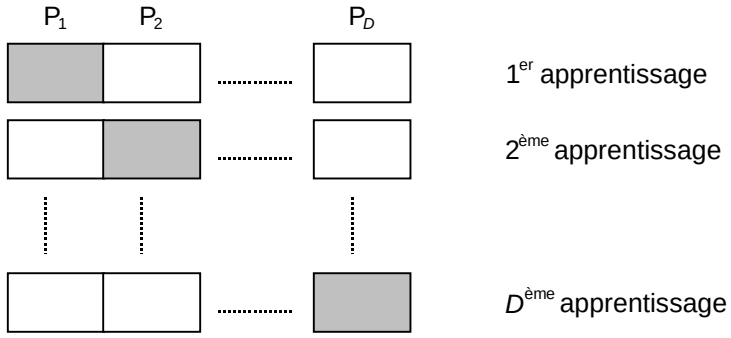


*Figure 2.3 : Principe de la validation croisée ; les parties grisées sont utilisées pour la validation et les autres pour l'apprentissage*

En utilisant la fonction de coût des moindres carrés, on procède généralement comme suit :

- pour chaque partie laissée de côté, on calcule l'erreur quadratique moyenne de validation (*EQMV*),
- à la fin, la performance de généralisation du modèle - appelée "score de validation croisée" - est estimée en réalisant la moyenne quadratique des *D* erreurs (*EQMV*) précédentes.

Dans le contexte de réseaux de neurones, la recherche de l'architecture optimale s'effectue souvent en partant d'un modèle linéaire et en augmentant progressivement le nombre de

neurones cachés. Le modèle optimal est alors défini comme étant celui qui présente le meilleur score de validation croisée.

La limite naturelle de la validation croisée correspond au cas où *D* est égal au nombre d'exemples dans la base d'apprentissage. Cette méthode est connue sous le nom de "leave-one-out" (voir [Plutowski 94]) car chaque apprentissage n'est validé que sur un seul exemple.

Les difficultés de cette méthode sont de deux ordres :

- le temps de calcul nécessaire, qui - pour une même base d'apprentissage est d'autant plus grand que *D* est élevé (il est donc maximum dans le cas du leave-one-out),
- des performances contrastées en termes de taille de l'architecture sélectionnée et d'estimation des performances. À ce niveau, deux cas sont à distinguer :
  - *le nombre d'exemples est grand au regard de la complexité de la fonction à approcher* (nombre d'entrées, non-linéarité) : dans ce cas, le phénomène de surajustement est difficile à mettre en évidence. La méthode donne certes de bons résultats - même avec un petit nombre de partitions - mais sans grand mérite car il y a peu de risque de surajustement.
  - *le nombre d'exemples est petit au regard de la complexité de la fonction à approcher* : on est obligé d'augmenter le nombre de partitions de façon à garder un nombre suffisant d'exemples pour réaliser l'apprentissage des *D* modèles. Les résultats montrent alors une tendance à la surestimation de la taille des modèles nécessaires et à la sous-estimation des scores de validation croisée. Ceci traduit un phénomène mis en évidence par [Breiman 96] : une petite modification des données d'apprentissage peut entraîner de grandes différences dans les modèles sélectionnés. Autrement dit, si l'on raisonne en termes de fonction de coût, les exemples dont on se sert pour estimer les paramètres d'un modèle peuvent grandement influencer les minima vers lesquels convergent les différents apprentissages. On parle alors d'instabilité vis-à-vis des données d'apprentissage : les *EQMV* calculées à partir des différentes partitions ne peuvent donc pas raisonnablement êtres moyennées pour estimer la performance de généralisation du modèle.

La littérature conseille généralement d'utiliser $D = 10$. Cependant, ne sachant pas a priori s'il dispose de "peu" ou de "beaucoup" d'exemples (au sens défini ci-dessus), le concepteur essaiera souvent différentes valeurs de *D*. Si l'on se rappelle qu'à partir d'une base d'apprentissage, il est recommandé de procéder à plusieurs initialisations des poids de façon à diminuer le risque de minima locaux, on arrive très vite à un nombre d'apprentissages très élevé. En soi, ceci n'est pas grave si les résultats de ces différents essais sont cohérents. Dans le cas contraire, le découragement peut rapidement intervenir…

Dans les chapitre 3 à 5, nous verrons comment remédier à ces difficultés par une approche originale du leave-one-out.

## 2.4 La régularisation

Ce terme désigne une série de techniques visant à privilégier les modèles les plus réguliers possible. Ceci part d'un constat simple : le surajustement se traduit le plus souvent par un modèle dont la sortie possède, aux endroits où celui-ci s'est ajusté au bruit, une courbure élevée.

La régularisation peut être effectuée de deux manières :

- en arrêtant l'apprentissage prématurément (early stopping), c'est-à-dire avant que ne soit atteint un minimum de la fonction de coût,
- en ajoutant un terme de pénalisation à la fonction de coût (par exemple par la méthode de "weight decay", présentée plus loin dans le paragraphe 2.4.2).

### *2.4.1 Early stopping*

Cette méthode consiste à arrêter l'apprentissage avant qu'il ne commence à s'ajuster au bruit contenu dans les points d'apprentissage, même si un minimum de la fonction de coût n'est pas atteint. Pour cela, on part :

- d'une architecture surdimensionnée, c'est-à-dire susceptible de conduire à un surajustement si on laissait l'apprentissage se poursuivre jusqu'à convergence,
- d'une partition des exemples disponibles en une base d'apprentissage et une base de validation.

On réalise alors l'apprentissage jusqu'au moment où les performances sur l'ensemble de validation atteignent un minimum. Ceci est représenté schématiquement sur la figure 2.4.

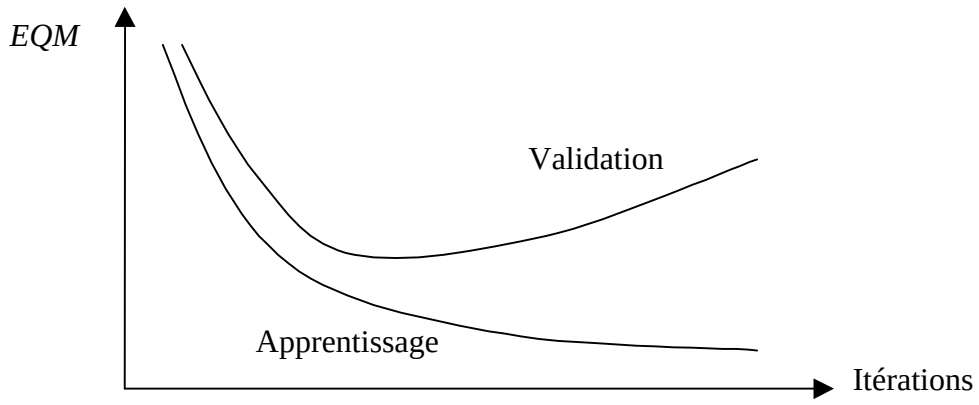


*Figure 2.4 : Évolution typique des performances d'apprentissage et de validation*

Les reproches que l'on peut faire à cette méthode, qui pourtant donne parfois de bons résultats, concernent à la fois sa mise en œuvre et sa "philosophie".

De même que pour la validation croisée, la meilleure façon de répartir les exemples disponibles en base d'apprentissage / de validation n'est pas définie, mais doit être étudiée au cas par cas, ce qui entraîne des problèmes de reproductibilité des résultats obtenus. Par ailleurs, il n'est plus question ici de procéder d'abord à la recherche de la taille d'architecture

optimale puis de regrouper les bases d'apprentissage et de validation pour avoir une meilleure estimation des paramètres du modèle.

Enfin, cette méthode - dans son principe même - va à l'encontre de la propriété de parcimonie des réseaux de neurones, puisqu'elle utilise des modèles surdimensionnés. Dans la littérature, il n'est pas rare de rencontrer des applications comportant beaucoup plus de paramètres ajustables que d'exemples (voir par exemple [Nelson 91]) ! Pour le concepteur qui cherche à utiliser un minimum de coefficients de manière à avoir un maximum de confiance sur leur estimation, une telle approche ne peut qu'éveiller des soupçons, mais elle est parfois incontournable, par exemple dans le domaine du traitement d'images.

En résumé, la régularisation de la fonction par early stopping est encore utilisée, car elle est rapide et simple à mettre en œuvre.

Dans [Sjöberg 92], les auteurs montrent que l'arrêt prématuré de l'apprentissage revient à utiliser un terme de pénalisation dans la fonction de coût, ce qui justifie la classification de l'early stopping parmi les méthodes de régularisation.

### *2.4.2 Pénalisation de la fonction de coût (weight decay)*

La deuxième façon d'influer sur la régularité du modèle consiste à introduire des contraintes dans la fonction de coût à minimiser. Nous ne détaillerons ici que la technique la plus connue (le *weight decay*), qui consiste à ajouter à la fonction de coût un terme proportionnel à la norme du vecteur des paramètres ajustables[3].

L'idée est ici de privilégier les modèles possédant de petits coefficients (en valeur absolue). La relation entre la taille des coefficients et la régularité de la fonction se comprend aisément en considérant la fonction réalisée par un neurone à une entrée :

$$y = \tanh(w_0 + w_1 x)$$

Le paramètre $w_0$ ainsi que la plage de variation de la variable $x$ étant fixés, cette fonction est d'autant plus régulière que la valeur absolue du paramètre $w_1$ est petite :

- si $w_1$ est très faible, la fonction sera quasiment linéaire,
- si $w_1$ est très grand, la fonction s'apparentera à un échelon.

On considère donc un terme $\Omega = \sum_i w_i^2$, pour lequel la somme s'effectue sur tous les coefficients du réseau, y compris les biais. Ce terme est le même que celui rencontré dans la méthode dite de *ridge regression* (voir [Saporta 90]), utilisée dans le cadre de régressions linéaires par rapport aux paramètres. Dans les deux cas, l'objectif est de limiter la variance d'un modèle en utilisant un estimateur biaisé.

---

[3] Une autre méthode consiste à pénaliser directement les fonctions à forte courbure par l'ajout - à la fonction de coût - de la norme du vecteur dérivée seconde de la sortie du modèle. Le lecteur intéressé par l'application de cette méthode aux réseaux de neurones pourra consulter l'article [Bishop 93].

Toute la difficulté du weight decay réside dans le dosage optimal entre la fonction de coût initiale et le terme de régularisation. Pour cela, on écrit la nouvelle fonction de coût sous la forme :

$$J^* = J + \nu \Omega \qquad (2.1)$$

Si l'on choisit $\nu$ trop grand, le modèle risque d'avoir un biais élevé. Inversement, si $\nu$ est trop petit, l'effet du terme de régularisation est trop faible, ce qui se traduit par une variance élevée. La grandeur $\nu$ devient donc en fait un paramètre, à estimer au même titre que les poids du réseau : elle est souvent désignée sous le nom d'hyperparamètre.

La seule véritable réponse apportée à l'estimation de cet hyperparamètre et de son interprétation est une approche bayesienne des réseaux de neurones décrite dans [MacKay 92] et d'autres publications du même auteur. Sans détailler les fondements théoriques ni la manière de mettre en œuvre les techniques bayesiennes, les avantages de cette approche sont :

- l'utilisation de tous les exemples d'apprentissage pour sélectionner et estimer les paramètres du modèle,
- l'accès immédiat aux intervalles de confiance.

Il s'agit cependant d'une technique assez compliquée à mettre en œuvre, et peu utilisée dans des applications pratiques.

## 2.5 Le surajustement

### *2.5.1 Discussion : qu'est-ce que le surajustement ?*

Après cette revue des outils les plus fréquemment utilisés dans le domaine des réseaux de neurones pour sélectionner le meilleur modèle tout en évitant les solutions surajustées, les questions suivantes se posent :

- quelle est l'origine exacte du surajustement ?
- quelle démarche faudrait-il mettre en œuvre pour régler ce problème à la base ?

Le surajustement caractérise une fonction dont la complexité - c'est-à-dire le nombre et la nature des degrés de libertés - est telle qu'elle est capable de s'ajuster exactement aux exemples d'apprentissage, même si ceux-ci sont entachés de bruit. Ce phénomène est donc - à l'origine - un phénomène local : dans certains domaines des entrées, la fonction utilise localement certains de ses degrés de liberté de manière à passer précisément par certains exemples.

Cette définition du surajustement suppose que tous les exemples ont la même importance et que l'on recherche effectivement une solution dont la réponse est - en moyenne - satisfaisante. Généralement, cette hypothèse est formalisée dans la fonction de coût choisie.

Ainsi, pour éviter le surajustement, il faudrait limiter l'influence de chaque exemple sur l'estimation des poids du modèle. Les valeurs de ceux-ci doivent être déterminées par

l'ensemble de la base d'apprentissage, et non par un exemple particulier (on retrouve ici le dilemme biais/ variance).

Dans cette optique, les méthodes précédentes ne s'attaquent pas directement au fond du problème :

- la validation croisée et l'early stopping sont des méthodes palliatives, qui n'empêchent pas le surajustement, mais permettent de le détecter : on se sert d'une base de validation pour détecter les zones de forte courbure de la sortie du modèle, entraînées par l'ajustement trop précis de la sortie du modèle aux exemples d'apprentissage,
- le weight decay est une heuristique : l'équivalence entre le fait que les poids du réseau soient "grands" et le surajustement n'a - à notre connaissance - jamais été démontrée ni dans un sens ni dans l'autre. Certes, elle se comprend intuitivement et [Bartlett 97] a montré que, pour avoir une bonne capacité de généralisation, la taille des poids est plus importante que la taille du réseau. Cependant, ceci pourrait très bien se révéler inexact dans tel ou tel cas particulier : cela dépend de la forme réelle de la fonction à approcher.

L'objectif que l'on a cherché à atteindre dans ce travail est le suivant : trouver un moyen de résoudre à la base le problème du surajustement. Pour cela, après quelques éléments sur la détection du surajustement, nous présenterons une étude détaillée de la théorie et de la mise en œuvre du leave-one-out.

### *2.5.2 Détection du surajustement*

La détection du surajustement se fonde sur la détermination du rang de la matrice $Z$. Cependant, il est difficile de vérifier - numériquement - si une matrice est de rang plein. Nous proposons donc de considérer la matrice $H = Z\ ({}^tZ\ Z)^{-1}\ {}^tZ$, qui est la matrice de projection orthogonale sur le sous-espace des solutions : si $Z$ est de rang plein, et dans ce cas seulement, $H$ doit vérifier les propriétés suivantes :

$$\forall i \in [1, ..., N]\ \ 0 \leq h_{ii} \leq 1 \text{ où } h_{ii} \text{ est le } i^{\text{ème}} \text{ terme diagonal de } H \tag{2.2}$$

$$\text{trace}\ (H) = \sum_{i}^{N} h_{ii} = \text{rang}\ (Z) \tag{2.3}$$

De plus, dans le cas où la matrice $Z$ possède une colonne dont tous les termes sont égaux, par exemple égaux à 1, c'est-à-dire si le modèle comporte un terme ajustable constant (appelé "biais" dans le contexte des réseaux de neurones) :

$$\forall i \in [1, ..., N]\ \ \frac{1}{N} \leq h_{ii} \tag{2.4}$$

Théoriquement, les termes $\{h_{ii}\}_{i\ =\ 1,\ ...,\ N}$ ne sont définis que dans le cas où ${}^tZ\ Z$ est inversible, c'est-à-dire dans le cas où $Z$ est de rang plein. Dans le cas contraire, le sous-espace de $R^N$ défini par les vecteurs colonnes de $Z$ (sous-espace appelé "sous-espace des solutions") n'est pas, au point correspondant à la solution des moindres carrés, de dimension $q$.

Ceci est caractéristique du surajustement : en effet, cela signifie que le modèle dispose, localement, de plus de paramètres ajustables que d'exemples[4]. Par analogie avec la résolution d'un système linéaire, le modèle dispose, localement, de plus d'inconnues que d'équations, ce qui conduit à une indétermination.

Contrairement à ce que l'on pourrait supposer, le surajustement ne traduit donc pas réellement la présence de paramètres "inutiles", mais plutôt la "sous-détermination" de certains paramètres.

De plus, il a été montré [Saarinen 93] qu'une déficience dans le rang de la matrice jacobienne a un effet négatif sur l'efficacité des algorithmes du second ordre. Récemment, [Zhou 98] a suggéré une procédure dans laquelle le rang de la matrice jacobienne est aussi réduit que possible, et le réseau est élagué lors de l'apprentissage : les résultats obtenus ainsi présentent une légère amélioration (en termes d'erreur quadratique moyenne sur un ensemble de test), par rapport à des modèles dont les paramètres sont estimés par des algorithmes conventionnels du second ordre. Nous proposons d'utiliser le même principe à un niveau différent.

D'après les relations (2.2), (2.3) et (2.4), trois cas sont à considérer :

1. (au moins) une des valeurs singulières de la matrice $Z$ est égale à zéro : alors rang ($Z$) < $q$ et les termes diagonaux $\{h_{ii}\}_{i = 1, ..., N}$ ne peuvent être calculés. Il y a donc au moins un des paramètres du modèle qui est sous-déterminé, ce qui se traduit par un surajustement.
2. aucune des valeurs singulières de la matrice $Z$ n'est égale à zéro, mais les valeurs calculées $\{h_{ii}\}_{i = 1, ..., N}$ ne satisfont pas les relations (2.2) à (2.4) : comme dans le premier cas, $Z$ n'est pas de rang plein. L'absence de valeurs singulières nulles résulte alors de problèmes de précision des calculs.
3. toutes les valeurs singulières sont non nulles et les $\{h_{ii}\}_{i = 1, ..., N}$ calculés satisfont les trois conditions (2.2) à (2.4).

Par abus de langage, nous désignerons les minima pour lesquels la matrice jacobienne est de rang plein sous le terme "*minima de rang plein*". Dans le cas contraire, nous parlerons de "*surajustement avec déficience du rang*".

Un calcul des $\{h_{ii}\}_{i = 1, ..., N}$ fondé sur une décomposition de $Z$ en valeurs singulières est présenté en Annexe 1.

Pratiquement, lors d'essais sur différents problèmes, nous avons effectivement constaté les phénomènes suivants :

- lorsque les $\{h_{ii}\}_{i = 1, ..., N}$ calculés ne satisfont pas aux relations (2.2) à (2.4), il y a au moins un neurone caché dont les poids sont très grands et ne contribuent donc pas à la constitution d'un modèle dont la sortie est régulière,

---

4 Prenons par exemple le cas de deux coefficients ne servant qu'à ajuster la valeur de la sortie d'un exemple particulier de la base d'apprentissage : les deux colonnes correspondantes de la matrice $Z$ sont composées de zéros, sauf pour la ligne correspondant à l'exemple en question, ce qui explique la déficience globale du rang de $Z$.

- lorsque l'erreur résiduelle sur un exemple *i* est très petite (de plusieurs ordres de grandeurs par rapport à l'écart-type du bruit), le $h_{ii}$ correspondant est systématiquement supérieur à 1, c'est-à-dire incompatible avec la relation (2.2).

Ceci confirme que les modèles pour lesquels les relations (2.2) à (2.4) ne sont pas satisfaites sont des modèles surajustés. Dans la pratique, pour une architecture donnée, on réalise plusieurs apprentissages, en partant à chaque fois d'initialisations aléatoires des poids, et l'on garde le modèle le plus satisfaisant pour le problème posé. Par conséquent, les conditions (2.2) à (2.4) peuvent être utilisées pour éliminer les modèles qui sont certainement surajustés. La figure 2.5 montre un exemple de modèle ne satisfaisant pas aux relations (2.2) à (2.4) : les $h_{ii}$ correspondant aux trois exemples désignés sont supérieurs à 1. Ceci correspond effectivement à un surajustement du modèle par rapport aux exemples d'apprentissage.

Tout cela repose sur l'hypothèse selon laquelle la déficience du rang de *Z* n'est pas due à une redondance de coefficients structurelle, due à l'architecture du modèle utilisé (voir Annexe 2).

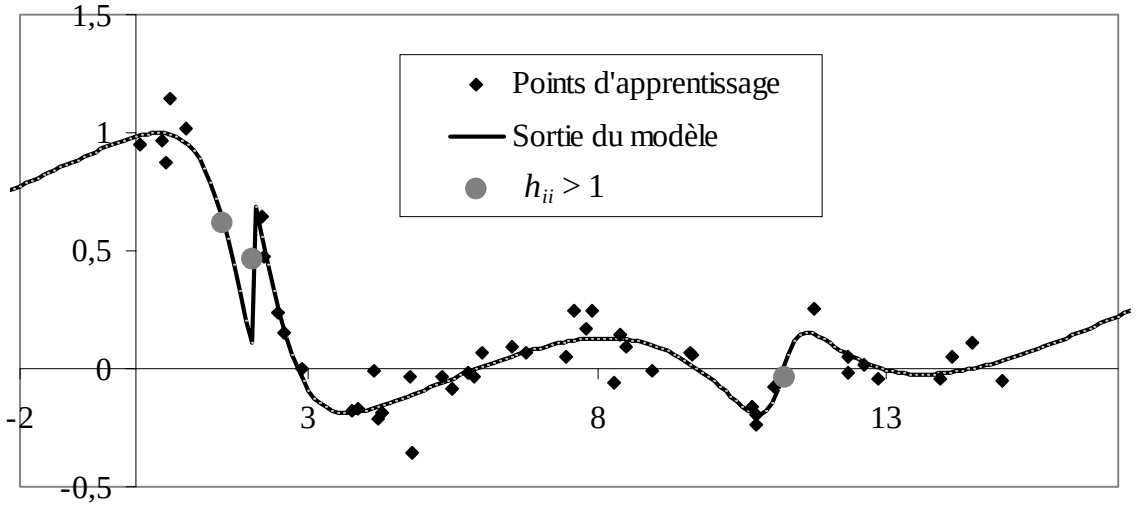


*Figure 2.5 : Exemple de modèle présentant trois $h_{ii}$ calculés supérieurs à* 1

Nous verrons dans les chapitres 3 et 4 qu'une déficience dans le rang de *Z* est une condition suffisante mais non nécessaire pour attester de la présence d'un surajustement : nous montrerons ainsi que les $\{h_{ii}\}_{i = 1, \ldots, N}$ peuvent être utilisés de manière plus subtile.

## 2.6 Les intervalles de confiance

### *2.6.1 Introduction*

La notion d'intervalle de confiance est étroitement liée à celle de variable aléatoire : établir un intervalle de confiance - au seuil de confiance 1 - $\alpha$ - pour une valeur aléatoire *Y*, c'est trouver un intervalle qui, avec une probabilité 1 - $\alpha$, contient la valeur de l'espérance mathématique de *Y*.

Il existe différentes manières d'estimer des intervalles de confiance pour la sortie d'un modèle non linéaire : [Tibshirani 96]. Il s'agit essentiellement des méthodes analytiques décrites dans

[Seber 89], des approches par ré-échantillonnage, type "boostrapping" [Efron 93] et des méthodes basées sur l'approche bayesienne et le weight decay [De Vaux 98]. Nous avons choisi, pour une raison sur laquelle nous reviendrons plus tard, d'utiliser la première de ces approches.

Il a été montré que, dans l'hypothèse d'un bruit de mesure gaussien, et si le modèle hypothèse est vrai (c'est-à-dire si la famille de fonctions considérée contient la fonction de régression inconnue), alors un intervalle de confiance approché pour l'espérance mathématique de la sortie $E(Y_p \mid \boldsymbol{x})$, avec un niveau de confiance 1-$\alpha$, est donné par :

$$E\left(Y_p \middle| \boldsymbol{x}\right) \in f\left(\boldsymbol{x}, \boldsymbol{\theta}_{LS}\right) \pm t_{\alpha}^{N-q}\, s \sqrt{{}^t\mathbf{z}\left({}^tZZ\right)^{-1}\mathbf{z}}\ , \tag{2.5}$$

expression dans laquelle :

- $\boldsymbol{\theta}_{LS}$ est la solution des moindres carrés obtenue par apprentissage,
- $t_{\alpha}^{N-q}$ désigne la valeur prise par une variable de Student à $N$ - $q$ degrés de liberté avec un niveau de confiance 1-$\alpha$,
- $s$ est une estimation de l'écart-type du bruit de mesure, estimation sur laquelle nous reviendrons dans le chapitre 4, paragraphe 4.4.

Cet intervalle de confiance suppose en outre que la matrice ${}^tZ\,Z$ est inversible, c'est-à-dire que la matrice jacobienne $Z$ est de rang plein. On ne peut donc pas calculer d'intervalles de confiance sur la sortie de modèles qui présentent un surajustement avec déficience de rang, argument supplémentaire pour ne pas considérer ces modèles.

Nous nous proposons ici de répondre à deux questions fréquemment posées au sujet des intervalles de confiance.

### *2.6.2 Différence entre performance du modèle et intervalle de confiance ?*

Pour répondre convenablement à cette question, il convient tout d'abord de rappeler ce que l'on entend par performance du modèle. La performance (sous-entendu "de généralisation") du modèle est une mesure de la différence entre la réalité et la prédiction du modèle, sur un ensemble représentatif de données. Pour cela, à partir de l'hypothèse (1.4), nous avions supposé que le bruit de mesure était gaussien. En caractérisant le modèle par son écart-type résiduel $s$, on obtient par conséquent une grandeur directement comparable à l'écart-type du bruit de mesure. Si le modèle hypothèse est vrai, alors la performance de généralisation est égale à l'écart-type du bruit de mesure de la sortie du processus.

Par une démonstration similaire à celle conduisant à la formule (2.5), on peut montrer que, pour tout $\boldsymbol{x}$ fixé, la variable aléatoire $\dfrac{y_p \middle| \boldsymbol{x} - E\left(Y_p \middle| \boldsymbol{x}\right)}{S}$ suit asymptotiquement une loi de Student à $N$ - $q$ degrés de liberté, et ainsi donner un deuxième intervalle de confiance pour $E(Y_p \mid \boldsymbol{x})$, centré cette fois-ci sur la sortie mesurée, avec un niveau de confiance de 1-$\alpha$ :

$$E\left(Y_p \middle| \boldsymbol{x}\right) \in y_p \middle| \boldsymbol{x} \pm t_{\alpha}^{N-q}\, s\,, \tag{2.6}$$

En résumé, la performance d'un modèle est définie par une estimation de l'écart type résiduel : il s'agit d'une grandeur moyenne caractérisant un modèle. En revanche, l'intervalle de confiance sur la prédiction du modèle est une grandeur qu'il faut calculer pour chaque vecteur d'entrée $\boldsymbol{x}$, puisque son expression fait intervenir la grandeur $\boldsymbol{z}(\boldsymbol{x}) = \left.\frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta} = \boldsymbol{\theta}_{LS}}$. Il s'agit en fait de la précision avec laquelle le modèle est capable d'ajuster **localement** sa sortie, compte tenu des exemples et des degrés de liberté disponibles.

Dans la représentation traditionnelle du plan (mesure, prédiction), les deux intervalles de confiance doivent donc être distingués et représentés perpendiculairement l'un à l'autre (figure 2.6).

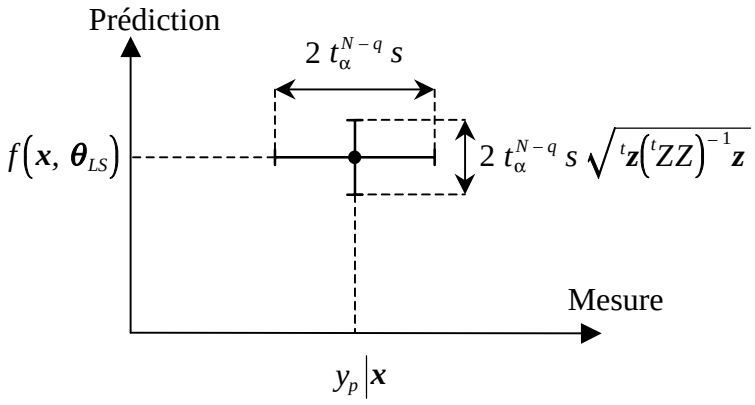


*Figure 2.6 : Intervalles de confiance sur la mesure et sur la prédiction*

De plus, il existe - pour les exemples d'apprentissage - un lien entre ces deux intervalles de confiance. En effet, nous avons montré dans le paragraphe 2.4.2 que :

$$\sqrt{{}^{t}\boldsymbol{z}^{i}\left({}^{t}ZZ\right)^{-1}\boldsymbol{z}^{i}} = \sqrt{h_{ii}} \leq 1 \qquad (2.7)$$

En résumé :

- la performance du modèle permet de définir un intervalle de confiance sur la mesure, intervalle qui est le même quel que soit le vecteur d'entrée $\boldsymbol{x}$,
- l'intervalle de confiance sur la prédiction du modèle doit être calculé pour chaque vecteur d'entrée $\boldsymbol{x}$. Il est égal à l'intervalle de confiance sur la mesure multiplié par un facteur qui, pour tous les exemples d'apprentissage, est inférieur ou égal à 1.

Considérons par exemple le modèle de la figure 2.7. Il s'agit de modéliser un processus à une entrée à partir des exemples représentés. La sortie du modèle est représentée par une ligne continue ; l'intervalle de confiance à 95 % sur la sortie du modèle par une zone grisée.

Le tracé (mesure, prédiction), pour les points d'apprentissage de l'exemple précédent, est représenté sur la figure 2.8. L'intervalle de confiance sur la mesure, qui est le même pour tous les points, n'y a été représenté qu'une seule fois afin de pas nuire à la lisibilité du graphique.

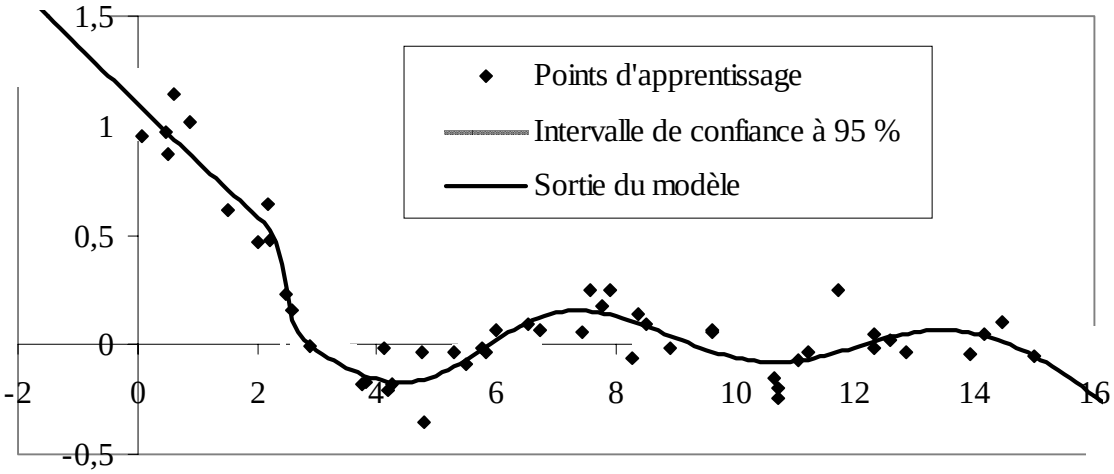


*Figure 2.7 : Exemple de modélisation avec intervalle de confiance associé*

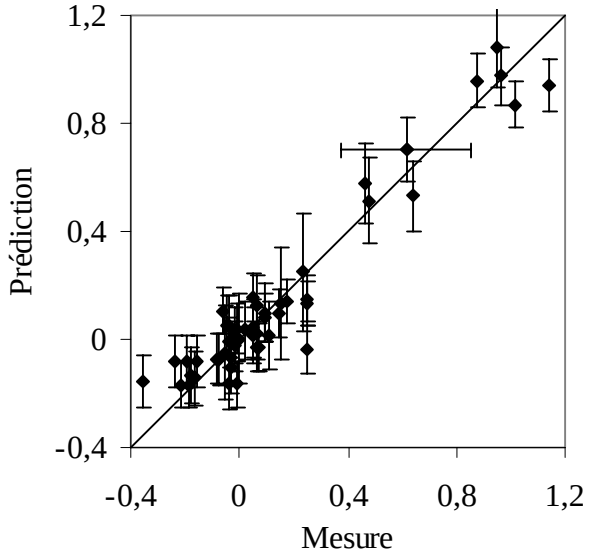


*Figure 2.8 : Intervalles de confiance sur la mesure et sur la prédiction pour le modèle de la figure 2.7*

Enfin, pour être plus complet, précisons qu'il est possible de combiner les deux expressions (2.5) et (2.6), en supposant qu'elles sont statistiquement indépendantes, afin d'obtenir un intervalle de confiance approché sur la réalisation de la variable aléatoire $Y_p \mid \boldsymbol{x}$, toujours en supposant le modèle hypothèse vrai. À un niveau de confiance 1 - $\alpha$, l'expression approchée de cet intervalle vaut (voir [Seber 89], page 193) :

$$Y_p \left| \boldsymbol{x} \in \; f\left(\boldsymbol{x}, \boldsymbol{\theta}_{LS}\right) \pm t_{\alpha}^{N-q}\, s \sqrt{1 + {}^{t}\mathbf{z}\left({}^{t}ZZ\right)^{-1}\mathbf{z}} \right. \tag{2.7}$$

Cependant - dans la pratique - sachant que ces deux notions ne sont pas indépendantes, il est plus prudent de considérer séparément l'incertitude sur la sortie du modèle et l'incertitude sur la mesure de la sortie du processus.

### *2.6.3 Comment interpréter les intervalles de confiance ?*

De manière générale, les intervalles de confiance sur la sortie d'un modèle permettent de statuer sur la validité d'une prédiction. Il est néanmoins utile de les interpréter avec plus de précision.

Comme nous venons de le montrer, l'intervalle de confiance est une grandeur locale, à calculer pour chaque vecteur $\boldsymbol{x}$ de l'espace des entrées. Cet intervalle correspond à la précision avec laquelle on est capable d'ajuster localement le modèle. Elle doit donc nous permettre de détecter les zones de l'espace des entrées où il n'y a pas assez d'exemples d'apprentissage.

Il ne s'agit pas pour autant (ou pas seulement) d'une mesure de la densité d'exemples au voisinage d'un vecteur d'entrée $\boldsymbol{x}$. Pour s'en convaincre, examinons l'exemple de la figure 2.9. Il s'agit d'une régression linéaire dont les paramètres sont estimés à partir d'un ensemble d'exemples répartis en deux groupe disjoints.

Dans le cas d'un modèle linéaire, il est facile de vérifier que ${}^t\boldsymbol{z}\ ({}^tZZ)^{-1}\ \boldsymbol{z}$ est une fonction quadratique des entrées du modèle. Ce comportement quadratique se retrouve effectivement sur la figure 2.9 et l'on constate que les intervalles de confiance sont plus faibles dans l'intervalle des entrées [3 ; 7], où il n'y a pas de points, qu'aux endroits où l'on dispose d'exemples. Ceci est tout à fait normal si la fonction de régression du processus est effectivement linéaire. Dans le cas contraire, par exemple si la fonction de régression présentait une bosse au milieu, il faudrait que l'on dispose d'exemples supplémentaires, de façon à justifier l'utilisation d'un modèle plus compliqué qu'un modèle linéaire.

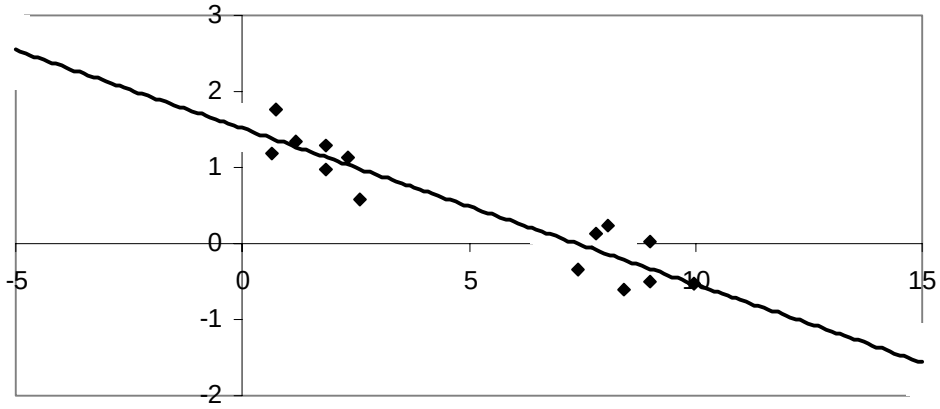


*Figure 2.9 : Intervalles de confiance sur une régression linéaire estimée à partir d'un ensemble d'exemples répartis en deux sous-groupes disjoints*

Considérons maintenant le cas de la figure 2.10, pour lequel l'intervalle où il n'y a pas de points est plus petit que précédemment, mais contient manifestement une non-linéarité de la fonction de régression. Le modèle considéré est un réseau de neurones à un neurone caché avec terme direct.

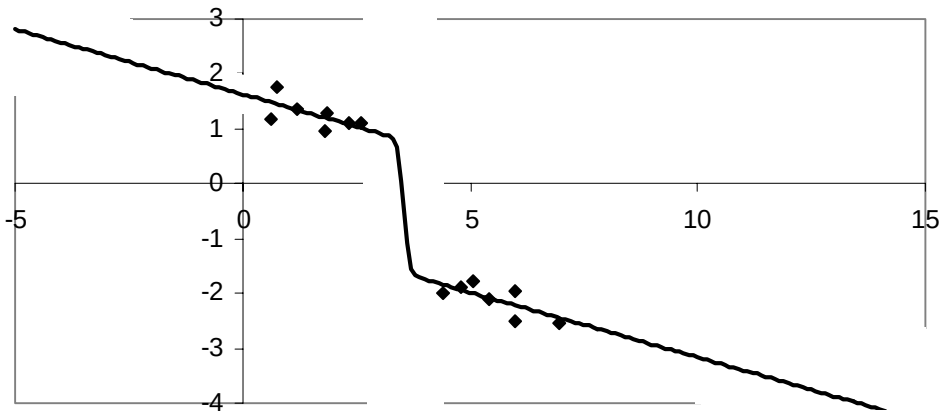


*Figure 2.10 : Intervalles de confiance traduisant l'incertitude sur le positionnement d'une non-linéarité*

Sur cet exemple, l'incertitude sur le positionnement de la non-linéarité - compte tenu des points disponibles - se traduit par des intervalles de confiance localement très élevés. Il suffit de rajouter quelques points (cf. figure 2.11) pour que cette incertitude soit levée.

Ici encore, le modèle de la figure 2.11 ainsi que les intervalles de confiance associés, ne sont exacts que si le modèle-hypothèse à 1 neurone caché et terme direct est vrai. Cependant, compte tenu des deux exemples ajoutés, il n'y a aucune raison de soupçonner, de la part du processus, un comportement autre que celui modélisé.

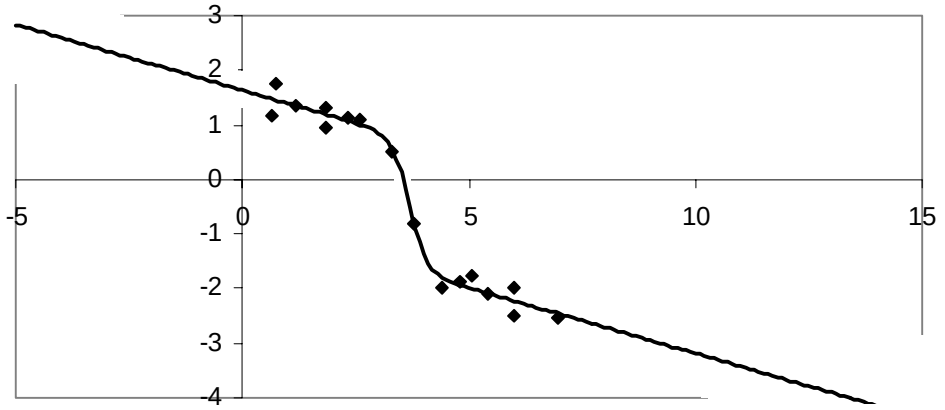


*Figure 2.11 : Intervalles de confiance après ajout de deux exemples dans la zone d'incertitude*

En résumé, les intervalles de confiance sont bien plus qu'une simple mesure de la densité d'exemples au voisinage d'un point de l'espace des entrées car ils tiennent également compte de la non-linéarité locale du modèle. En fait, les intervalles de confiance traduisent l'incertitude sur le positionnement d'une courbe moyenne (la sortie du modèle), compte tenu des exemples disponibles et de la complexité de la famille de fonctions utilisée. Ils s'avèrent donc constituer une aide indispensable. On trouve, dans la plupart des ouvrages sur la régression non linéaire ([Seber 89] ou [Bates 88]), diverses expressions des intervalles de confiance pour des modèles linéaires et non linéaires.

Il convient cependant de les utiliser avec précaution car des intervalles de confiance restreints ne garantissent pas forcément l'exactitude du modèle : ils peuvent provenir d'un manque d'exemples dans une région où l'interpolation la plus simple ne s'avère pas exacte. Pas plus que les autres méthodes de modélisation, les réseaux de neurones ne sont capables d'extrapoler des comportements non décelables sur les exemples d'apprentissage.

Nous reviendrons sur leur utilisation dans le cadre de la sélection de modèles au chapitre 4.

## 2.7 Bornes sur les performances de généralisation

L'obtention de bornes sur l'erreur de généralisation d'un modèle est un sujet ouvert qu'étudient beaucoup de théoriciens.

Le problème peut être posé de la manière suivante :

*Étant donné un modèle ou encore une famille de fonctions, un algorithme d'apprentissage pour cette famille, et une mesure empirique fondée sur une base de N exemples, est-il possible, pour toute fonction résultant de l'algorithme d'apprentissage, de borner la différence entre la performance réelle de généralisation et sa mesure empirique ?*

Les bornes obtenues sont valides avec une certaine probabilité sur l'ensemble d'apprentissage et permettent d'obtenir un intervalle de confiance sur l'estimation des performances de généralisation du modèle. Cet intervalle est alors valable quelle que soit la probabilité portant sur les données du problème et concerne toutes les fonctions du modèle. Les théorèmes concernant ces bornes sont donc des théorèmes de convergence uniforme en probabilité et c'est pour cette dernière raison qu'ils sont souvent cités comme étant des résultats au pire cas : ils sont valables pour toutes les fonctions du modèle et pas seulement pour la fonction obtenue par l'algorithme d'apprentissage.

Les bornes décroissent généralement avec le nombre d'exemples $N$ et conduisent alors à calculer le nombre minimal d'exemples à fournir pour être sûr à 1 - $\alpha$ % que notre estimation de la performance de généralisation ne diffère de la véritable performance de généralisation que d'une quantité $\varepsilon$ choisie préalablement.

Tous ces travaux se fondent de près ou de loin sur l'approche de Vapnik et Chervonenkis dont une présentation peut se trouver dans [Bottou 97], ou directement dans [Vapnik 82]. La notion de dimension de Vapnik-Chervonenkis (ou VC-dimension), qui caractérise la capacité d'un modèle (voir par exemple [Euvrard 93]), est centrale.

Dans la pratique, ces bornes sont extrêmement difficiles à obtenir et nécessitent un grand nombre de données (typiquement 100000) pour être intéressantes. Calculer la VC-dimension ou d'autres dimensions combinatoires pour utiliser les théorèmes de convergence uniforme est fastidieux et, souvent, seule une estimation assez lâche de ces dimensions est disponible. À l'heure actuelle, ces approches théoriques ne fournissent pas de résultats pratiques. Leur intérêt est essentiellement de donner des indications sur comment choisir un modèle par rapport à un autre en exhibant des caractéristiques utiles pour le contrôle de la performance de généralisation.

Concernant l'estimation des performances de généralisation par leave-one-out, il est nécessaire (voir [Kearns 97]), pour obtenir de telles bornes, que l'algorithme utilisé ait une certaine stabilité, c'est-à-dire que la solution obtenue en supprimant un exemple de la base d'apprentissage ne soit pas trop éloignée de la solution obtenue sur tous les exemples. Cette notion de stabilité a été formalisée de deux manières différentes par [Devroye 79] et [Kearns 97], dans les deux cas sous la forme "stable à $\delta$ près dans 1- $\beta$ % des cas".

En particulier, [Kearns 97] montre que, sous réserve de stabilité et par rapport à la vraie performance de généralisation, l'estimation obtenue par leave-one-out n'est jamais pire que l'estimation fournie par l'erreur empirique d'apprentissage.

En fait, toute la difficulté réside dans la détermination des propriétés de stabilité de tel ou tel algorithme. Dans le cas d'une minimisation de l'erreur quadratique, l'ajout d'un terme de pénalisation de la fonction de coût (weight decay) semble être une possibilité pour obtenir une certaine forme de stabilité (voir [Bartlett 97]), mais ceci reste à préciser.

Dans les chapitres suivants, nous ne considérerons pas cette question des bornes, dont l'utilisation pratique n'est pas envisageable pour l'instant. Tout au long de ce mémoire, nous utiliserons néanmoins l'erreur quadratique moyenne obtenue par leave-one-out pour estimer les performances de généralisation d'un modèle. Nous montrerons pourquoi, dans certains cas, cette estimation n'est pas raisonnable, ce qui nous amènera à définir une manière plus fiable de procéder.

# 3 ETUDE THEORIQUE DU LEAVE-ONE-OUT


## Résumé

*Dans le cas d'un modèle non-linéaire par rapport aux paramètres, il est possible d'obtenir une approximation de la solution des moindres carrés $\boldsymbol{\theta}_{LS}$ en effectuant un développement de Taylor au premier ordre de la sortie du modèle. On utilise également cette approche pour estimer la solution des moindres carrés $\boldsymbol{\theta}_{LS}^{(-i)}$ obtenue après avoir supprimé l'exemple i de la base d'apprentissage. En combinant ces deux expressions, on obtient une relation approchée entre $\boldsymbol{\theta}_{LS}$ et $\boldsymbol{\theta}_{LS}^{(-i)}$, expression qui est valable sous réserve que les deux développements de Taylor le soient, c'est-à-dire que la courbure du sous-espace des solutions soit suffisamment faible.*

*On utilise la relation précédente pour prédire l'effet du retrait d'un exemple de l'ensemble d'apprentissage. Ainsi, l'erreur de prédiction sur cet exemple est multipliée par un coefficient qui tend vers l'infini lorsque $h_{ii}$ tend vers* 1. *De même, l'intervalle de confiance sur la prédiction de l'exemple retiré de la base d'apprentissage devient infini lorsque $h_{ii}$ tend vers* 1. *Ceci nous amène à interpréter la grandeur $h_{ii}$ comme l'influence de l'exemple i sur l'estimation des paramètres du modèle : plus $h_{ii}$ (qui est positif) est grand et tend vers* 1, *plus cette influence est grande.*

*Afin de comparer l'effet du retrait d'un exemple, tel qu'il est prédit par ces formules, à celui obtenu par apprentissage, nous avons introduit une classification des modèles en modèles "propices au leave-one-out" ou non. Il s'agit en réalité de définir les modèles pour lesquels l'effet du retrait d'un exemple ne peut pas être raisonnablement estimé par apprentissage, et donc pour lesquels une comparaison avec les résultats des formules de linéarisation n'a pas de sens. Nous proposons une méthode géométrique qui permet de s'assurer - en partie seulement - qu'une solution est propice au leave-one-out.*

*Nous montrons enfin qu'un modèle est souvent non propice au leave-one-out à cause de la présence d'un exemple à forte influence sur les coefficients et dont le retrait fait converger l'apprentissage vers un autre minimum de la fonction de coût. Dans ce cas, la seule manière d'estimer l'effet du retrait de cet exemple sur le modèle est d'utiliser les formules fondées sur le développement de Taylor.*


## 3.1 Introduction

Lorsqu'un modèle est linéaire par rapport à ses paramètres, on obtient facilement la solution des moindres carrés en résolvant le système d'équations canoniques. Ce n'est pas le cas pour les modèles non linéaires par rapport à leurs paramètres ; cependant, en considérant une zone suffisamment petite de l'espace des paramètres, on peut trouver une expression approchée de la solution des moindres carrés en réalisant un développement limité au premier ordre de la sortie du modèle. Après un bref rappel de la démonstration de ce résultat classique (voir par

exemple [Seber 89]), nous allons montrer comment il peut s'appliquer pour détecter les modèles surajustés et prédire l'effet du retrait d'un exemple de l'ensemble d'apprentissage.

## 3.2 Approximation locale de la solution des moindres carrés

Notons $\boldsymbol{\theta}_{LS}$ la solution des moindres carrés, c'est-à-dire le vecteur des paramètres qui minimise la fonction de coût quadratique :

$$J(\boldsymbol{\theta}) = {}^t[\boldsymbol{y_p} - \boldsymbol{f}(X, \boldsymbol{\theta})]\ [\boldsymbol{y_p} - \boldsymbol{f}(X, \boldsymbol{\theta})] \tag{3.1}$$

avec $\boldsymbol{f}(X, \boldsymbol{\theta}) = {}^t[f(\boldsymbol{x}^1, \boldsymbol{\theta}), ..., f(\boldsymbol{x}^N, \boldsymbol{\theta})]$, où $X$ est la matrice ${}^t[\boldsymbol{x}^1, ..., \boldsymbol{x}^N]$ de dimensions $(N, n)$.

Si le modèle est linéaire par rapport aux paramètres, c'est-à-dire si $\boldsymbol{f}(X, \boldsymbol{\theta}) = {}^t[{}^t\boldsymbol{x}^1\boldsymbol{\theta}, ..., {}^t\boldsymbol{x}^N\boldsymbol{\theta}]$, alors la solution des moindres carrés s'écrit :

$$\boldsymbol{\theta}_{LS} = ({}^tX\,X)^{-1}\,{}^tX\,\boldsymbol{y}_p \tag{3.2}$$

Si le modèle n'est pas linéaire par rapport aux paramètres, il n'existe pas d'expression similaire. Néanmoins, on peut obtenir une solution locale approchée de la solution des moindres carrés à partir d'un développement limité de $\boldsymbol{f}$ au voisinage d'un point $\boldsymbol{\theta}^*$ de l'espace des paramètres.

Ce développement limité permet d'obtenir une expression approchée de $\boldsymbol{f}(X, \boldsymbol{\theta})$ et donc de $J(\boldsymbol{\theta})$. La solution des moindres carrés $\boldsymbol{\theta}_{LS}$ est ensuite obtenue, comme dans le cas linéaire, en annulant le gradient de $J$, mais après n'avoir conservé que les termes du premier ordre en $(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$.

Pour obtenir un développement limité cohérent de $\frac{\partial J}{\partial \boldsymbol{\theta}}$ au premier ordre en $(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$, il faut partir d'un développement limité au second ordre de $\boldsymbol{f}(X, \boldsymbol{\theta})$ au voisinage de $\boldsymbol{\theta}^*$ :

$$\boldsymbol{f}(X, \boldsymbol{\theta}) \cong \boldsymbol{f}(X, \boldsymbol{\theta}^*) + Z\,(\boldsymbol{\theta} - \boldsymbol{\theta}^*) + {}^t(\boldsymbol{\theta} - \boldsymbol{\theta}^*)\,\boldsymbol{S}\,(\boldsymbol{\theta} - \boldsymbol{\theta}^*) \tag{3.3}$$

Dans la formule précédente :

- $Z$ désigne la matrice jacobienne du modèle, définie par $Z = {}^t[\boldsymbol{z}^1, ..., \boldsymbol{z}^N]$ où $\boldsymbol{z}^i = \left.\frac{\partial f(\boldsymbol{x}^i, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta} = \boldsymbol{\theta}^*}$,
- $\boldsymbol{S} = \sum_{i=1}^{N} S(\boldsymbol{x}^i)\,\boldsymbol{e}_i$ est un tenseur d'ordre 3 dans lequel $S(\boldsymbol{x}^i)$ est une matrice de dimensions $(q, q)$ définie par $S(\boldsymbol{x}^i) = \left(\left.\frac{\partial^2 f(\boldsymbol{x}^i, \boldsymbol{\theta})}{\partial\theta_j\,\partial\theta_k}\right|_{\boldsymbol{\theta} = \boldsymbol{\theta}^*}\right)_{\substack{j = 1, ..., q \\ k = 1, ..., q}}$ et $\boldsymbol{e}^i$ est le $i^{\text{ème}}$ vecteur de la base orthonormale de $\Re^N$.

En utilisant (3.3) dans l'expression du coût, on obtient, après dérivation et en négligeant les termes d'ordre supérieur à 1 en $(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$ :

$$\frac{\partial J}{\partial \boldsymbol{\theta}} = \frac{\partial J}{\partial (\boldsymbol{\theta} - \boldsymbol{\theta}^*)} \cong -2\,{}^tZ\left(\boldsymbol{y}_p - \boldsymbol{f}(x, \boldsymbol{\theta}^*)\right) + \left\{2\,{}^tZZ - 2\sum_{i=1}^{N}\left(y_p^i - f(\boldsymbol{x}^i, \boldsymbol{\theta}^*)\right) S(\boldsymbol{x}^i)\right\}(\boldsymbol{\theta} - \boldsymbol{\theta}^*) \tag{3.4}$$

Or, la matrice de dimensions ($q$, $q$) située à l'intérieur des accolades n'est autre que le Hessien de la fonction de coût, définit par $H = \left( \frac{\partial^2 J(\boldsymbol{\theta})}{\partial \theta_j \, \partial \theta_k} \right)_{\substack{j = 1, \ldots, q \\ k = 1, \ldots, q}}$. Dans la pratique, la deuxième partie du Hessien peut être négligée, ce qui conduit à l'approximation suivante, dite de Levenberg-Marquardt :

$$H \cong 2 \, {}^t Z Z \tag{3.5}$$

Dans tout ce qui suit, on se place dans le cas où la matrice $Z$ est de rang plein, c'est-à-dire dans un cas où il n'y a ni surajustement avec déficience du rang, ni redondance de coefficients.

Finalement, en annulant le gradient de $J$, on obtient une approximation de $\boldsymbol{\theta}_{LS}$ sous la forme :

$$\boldsymbol{\theta}_{LS} \cong \boldsymbol{\theta}^* + ({}^t Z \, Z)^{-1} \, {}^t Z \, [\boldsymbol{y}_p - \boldsymbol{f}(X, \boldsymbol{\theta}^*)] \tag{3.6}$$

Cette expression est donc valable sous deux hypothèses :

1. le développement limité au second ordre de la fonction de coût est valable, c'est-à-dire que les termes du second ordre du développement limité de $\frac{\partial J}{\partial \boldsymbol{\theta}}$ sont négligeables par rapport aux termes du premier ordre,
2. le Hessien de la fonction de coût peut être approché par $2 \, {}^t Z Z$, comme discuté dans [Bishop 95].

Dans le cas d'un modèle linéaire par rapport à ses paramètres, la relation (3.6) n'est pas une approximation, mais une égalité. Cette approche a été utilisée par plusieurs auteurs dans des buts différents, y compris pour estimer des intervalles de confiances sur les paramètres et sur les prédictions du modèle (voir par exemple [Seber 89]).

Si ce résultat (formule 3.6) est classique, il est important d'insister sur sa démonstration, et sur les deux hypothèses permettant d'y arriver. En effet, il est possible d'arriver au même résultat en partant d'un développement de Taylor de $\boldsymbol{f}$ au premier ordre, ce qui conduit à oublier le second terme du Hessien de $J$ au lieu de le négliger (voir [Seber 89]).

## 3.3 Effet du retrait d'un exemple de l'ensemble d'apprentissage

Nous allons montrer comment l'approximation de la solution des moindres carrés peut être utilisée pour prédire l'effet du retrait d'un exemple de l'ensemble d'apprentissage.

Dans tout ce qui suit, toutes les grandeurs concernant les modèles dont l'apprentissage a été réalisé à l'aide de tous les exemples sauf le $i^{\text{ème}}$ seront munies d'un exposant $^{(-i)}$. Ainsi, $\boldsymbol{f}^{(-i)}(X, \boldsymbol{\theta})$ et $\boldsymbol{y_p}^{(-i)}$ sont des vecteurs de dimension $N$ - 1, de même que $Z^{(-i)}$ est une matrice de dimensions ($N$ - 1, $q$). Inversement, toutes les grandeurs sans exposant feront référence à des modèles ajustés sur les $N$ exemples. Par ailleurs, la différence entre la sortie mesurée et la prédiction d'un modèle sera appelée "résidu" si l'exemple correspondant fait partie de la base d'apprentissage, et "erreur" dans le cas contraire.

### *3.3.1 Effet du retrait d'un exemple sur sa prédiction*

Si l'on suppose que le retrait de l'exemple $i$ de la base d'apprentissage ne provoque qu'une légère modification de la solution des moindres carrés, alors on peut, de même que pour la relation (3.6), établir une expression approchée de $\boldsymbol{\theta}_{LS}^{(-i)}$ au voisinage de $\boldsymbol{\theta}^*$ :

$$\boldsymbol{\theta}_{LS}^{(-i)} \cong \boldsymbol{\theta}^* + \left({}^t Z^{(-i)}\, Z^{(-i)}\right)^{-1} {}^t Z^{(-i)}\, [\boldsymbol{y}_p^{(-i)} - \boldsymbol{f}^{(-i)}(X, \boldsymbol{\theta}^*)] \tag{3.7}$$

En combinant (3.6) et (3.7), on obtient le résultat suivant (voir par exemple [Antoniadis 92]) :

$$\boldsymbol{\theta}_{LS}^{(-i)} \cong \boldsymbol{\theta}_{LS} - \left({}^t Z\, Z\right)^{-1} \boldsymbol{z}^i \frac{R_i}{1 - h_{ii}} \tag{3.8}$$

dans lequel $\boldsymbol{z}^i$ est le vecteur dont les éléments constituent la $i^{ème}$ colonne de la matrice $Z$, et $R_i$ est le résidu du $i^{ème}$ exemple : $R_i = y_{pi} - f(\boldsymbol{x}^i, \boldsymbol{\theta}_{LS}) = y_{pi} - f(\boldsymbol{x}^i, \boldsymbol{\theta}^*) - {}^t\boldsymbol{z}^i\, \boldsymbol{\theta}_{LS}$ et $h_{ii}$ est la $i^{ème}$ composante de la projection, sur le sous-espace des solutions, du vecteur unité le long de l'axe $i$ : $h_{ii} = {}^t\boldsymbol{z}^i\, ({}^t Z\, Z)^{-1}\, \boldsymbol{z}^i$. La démonstration de la relation (3.8) est donnée en Annexe 3.

Ceci nous permet d'estimer l'erreur $R_i^{(-i)}$ sur la prédiction du $i^{ème}$ exemple quand celui-ci est retiré de la base d'apprentissage : $R_i^{(-i)} = y_{p\,i} - f(\boldsymbol{x}^i, \boldsymbol{\theta}^*) - {}^t\boldsymbol{z}^i\, \boldsymbol{\theta}_{LS}^{(-i)}$. On a donc : $R_i^{(-i)} \cong R_i + {}^t\boldsymbol{z}^i \left(\boldsymbol{\theta}_{LS} - \boldsymbol{\theta}_{LS}^{(-i)}\right)$. En utilisant la relation (3.8), on obtient la même relation que dans le cas linéaire :

$$R_i^{(-i)} \cong \frac{R_i}{1 - h_{ii}} \tag{3.9}$$

De même, en utilisant les formules précédentes, on trouve une approximation de la fonction de coût quadratique :

$$J(\boldsymbol{\theta}_{LS}^{(-i)}) \cong J(\boldsymbol{\theta}_{LS}) - \frac{R_i^2}{1 - h_{ii}} \tag{3.10}$$

Une idée analogue a été proposée par [Larsen 96] et [Sorensen 96]. Cependant, l'utilisation que ces auteurs ont faite de leur développement limité n'était pas correcte : pour s'en rendre compte, il suffit de remarquer que leurs résultats, contrairement aux formules (3.8) à (3.10), ne sont pas exacts dans le cas d'un modèle linéaire.

Les figures 3.1.b et 3.1.c permettent d'apprécier la précision des formules (3.9) et (3.10), dans le cas du modèle considéré sur la figure 3.1.a. La figure 3.1.b représente l'erreur de prédiction pour chaque exemple extrait de l'ensemble d'apprentissage, en fonction de l'erreur estimée par la relation (3.9). De même, la figure 3.1.c représente les valeurs de la fonction de coût obtenues après retrait de chaque exemple, par apprentissage et à l'aide de la relation (3.10).

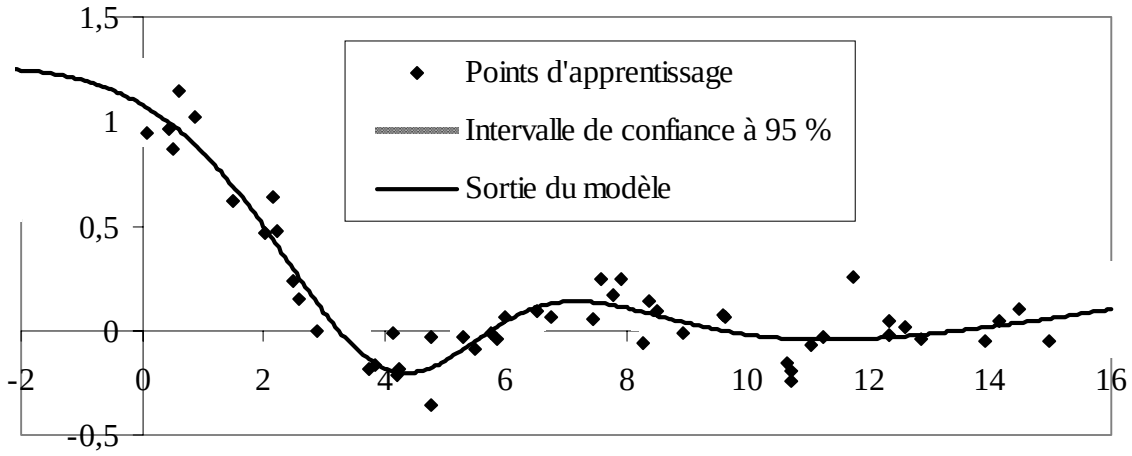


*Figure 3.1.a : Ensemble d'apprentissage et modèle considéré*

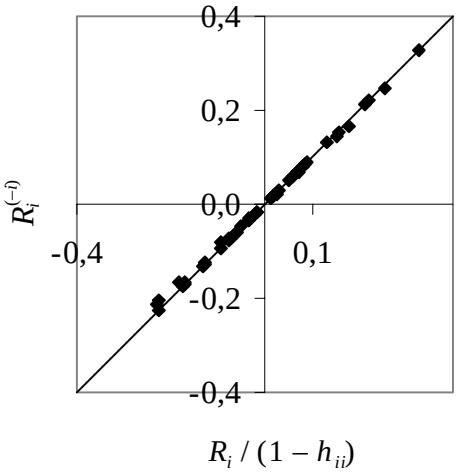


*Figure 3.1.b : Erreurs sur exemple retiré, par apprentissage et formule* (3.9)

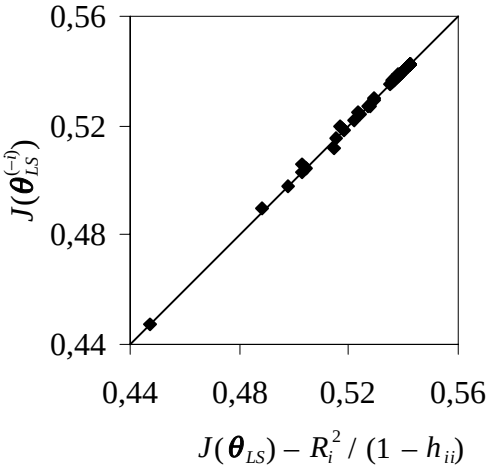


*Figure 3.1.c : Coûts avec exemple retiré, par apprentissage et formule* (3.10)

### *3.3.2 Effet du retrait d'un exemple sur l'intervalle de confiance de sa prédiction*

Dans le chapitre 2, nous avons indiqué l'expression approchée d'un intervalle de confiance sur la sortie du modèle, dans l'hypothèse d'un bruit de mesure gaussien, et en supposant le modèle-hypothèse vrai. Rappelons qu'à un niveau de confiance 1 - $\alpha$, l'intervalle de confiance approché pour $E(Y_p \mid \boldsymbol{x})$ est :

$$E\left(Y_p \middle| \boldsymbol{x}\right) \in f\left(\boldsymbol{x}, \boldsymbol{\theta}_{LS}\right) \pm t_\alpha^{N-q}\, s \sqrt{{}^t\boldsymbol{z}\left({}^tZZ\right)^{-1}\boldsymbol{z}}\ , \tag{3.11}$$

Pour l'exemple numéro $i$ de la base d'apprentissage, l'intervalle de confiance précédent s'écrit donc :

$$E\left(Y_p \middle| \boldsymbol{x}^i\right) \in f\left(\boldsymbol{x}^i, \boldsymbol{\theta}_{LS}\right) \pm t_\alpha^{N-q}\, s \sqrt{h_{ii}} \tag{3.12}$$

Ainsi, d'après la propriété (2.2), la demi-largeur de l'intervalle de confiance sur les exemples d'apprentissage est inférieure à $t_{\alpha}^{N-q}\, s$.

Par ailleurs, l'expression (3.12) montre que l'approche analytique des intervalles de confiance fait intervenir la même grandeur ($h_{ii}$) que les grandeurs associées au retrait d'un exemple de la base d'apprentissage, ce qui est normal car les deux approches utilisent le même développement limité. C'est la raison pour laquelle nous avons choisi cette expression des intervalles de confiance. Nous verrons dans le chapitre 4 comment exploiter cette similitude dans le cadre de la sélection de modèles.

Dans le paragraphe précédent, nous avons montré que l'on peut estimer à la fois la valeur de la fonction de coût (3.10), après avoir retiré un exemple de la base d'apprentissage, et l'erreur de prédiction (3.9) sur cet exemple. De la même façon, il est possible d'estimer les intervalles de confiance sur cette prédiction.

Étant donné un vecteur d'entrée $\boldsymbol{x}^i$, l'intervalle de confiance approché pour l'espérance mathématique de $\boldsymbol{Y}_p$, avec un niveau de confiance 1 - $\alpha$, est, pour le modèle obtenu après avoir supprimé l'exemple $i$ de la base d'apprentissage :

$$E^{(-i)}\left( Y_p \left| \boldsymbol{x}^i \right.\right) \in f\left(\boldsymbol{x}^i, \boldsymbol{\theta}_{LS}^{(-i)}\right) \pm t_{\alpha}^{N-q-1}\, s^{(-i)} \sqrt{{}^t\boldsymbol{z}^i \left( {}^tZ^{(-i)} Z^{(-i)} \right)^{-1} \boldsymbol{z}^i} \tag{3.13}$$

Notons $h_{ii}^{(-i)} = {}^t\boldsymbol{z}^i \left( {}^tZ^{(-i)} Z^{(-i)} \right)^{-1} \boldsymbol{z}^i$. D'après le lemme d'inversion matricielle présenté en Annexe 3, on montre facilement que :

$$h_{ii}^{(-i)} \cong \frac{h_{ii}}{1 - h_{ii}}, \tag{3.14}$$

et que cette relation est une égalité dans le cas d'un modèle linéaire. En combinant (3.9), (3.13) et (3.14), on obtient l'expression suivante :

$$E^{(-i)}\left( Y_p \left| \boldsymbol{x}^i \right.\right) \in f\left(\boldsymbol{x}^i, \boldsymbol{\theta}_{LS}\right) - \frac{h_{ii}}{1 - h_{ii}} R_i \pm t_{\alpha}^{N-q-1}\, s^{(-i)} \sqrt{\frac{h_{ii}}{1 - h_{ii}}}\ . \tag{3.15}$$

Dans l'expression précédente, la seule inconnue reste l'estimation $s^{(-i)}$ de la performance de généralisation du modèle ajusté sans l'exemple $i$. Dans le cas où l'on utilise la valeur de $J$ pour estimer cette performance, $s^{(-i)}$ se déduit de $s$ par la formule (3.10). Dans les autres cas, nous supposerons que $s^{(-i)} \approx s$.

### *3.3.3 Interprétation des $h_{ii}$*

En résumé, l'approximation de la solution des moindres carrés permet d'estimer l'effet du retrait d'un exemple sur toutes les grandeurs utilisées lors d'une modélisation non linéaire, notamment par réseaux de neurones, y compris sur les intervalles de confiance.

En examinant le récapitulatif du tableau 3.1, il apparaît que les $\{h_{ii}\}_{i\,=\,1,\,\ldots,\,N}$ jouent un rôle déterminant dans l'estimation de l'influence de chaque exemple sur le modèle.

| Base d'apprentissage | Tous les exemples | Exemple *i* retiré |
|---|---|---|
| Solution des moindres carrés | $\boldsymbol{\theta}_{LS}$ | $\boldsymbol{\theta}_{LS} - \left({}^t Z\, Z\right)^{-1} \boldsymbol{z}^i \frac{R_i}{1-h_{ii}}$ |
| Coût | $J(\boldsymbol{\theta}_{LS})$ | $J(\boldsymbol{\theta}_{LS}) - \frac{R_i^2}{1-h_{ii}}$ |
| Prédiction de l'exemple *i* | $f\left(\boldsymbol{x}^i, \boldsymbol{\theta}_{LS}\right)$ | $f\left(\boldsymbol{x}^i, \boldsymbol{\theta}_{LS}\right) - \frac{h_{ii}}{1-h_{ii}} R_i$ |
| Résidu / erreur sur l'exemple *i* | $R_i$ | $\frac{R_i}{1-h_{ii}}$ |
| Intervalle de confiance sur la prédiction de l'exemple *i* | $\pm\, t_\alpha^{N-q}\, s \sqrt{h_{ii}}$ | $\pm\, t_\alpha^{N-q-1}\, s \sqrt{\frac{h_{ii}}{1-h_{ii}}}$ |

*Tableau 3.1 : Résumé de l'influence du retrait d'un exemple de la base d'apprentissage*

D'un point de vue théorique, deux cas extrêmes sont à considérer :

- si l'axe *i* est orthogonal au sous-espace des solutions, défini par les vecteurs colonne de *Z*, tous ces vecteurs colonne ont leur $i^{\text{ème}}$ composante égale à zéro ; par conséquent $\boldsymbol{z}^i = 0$ et $h_{ii} = 0$. L'exemple *i* n'a pas d'influence sur le modèle, ce qui est confirmé par les relations (3.8) à (3.10). Rappelons que ce cas ne peut se produire que si le modèle ne possède pas de biais,
- si l'axe *i* se trouve dans le sous-espace des solutions, $h_{ii} = 1$ et $R_i = 0$. En d'autres termes, l'exemple *i* a été parfaitement appris, ce qui conduit à une indétermination dans les relations (3.8) à (3.10).

La grandeur $h_{ii}$ apparaît ainsi comme une véritable mesure de l'influence de l'exemple *i* sur l'estimation des paramètres du modèle. Plus $h_{ii}$ est proche de 1, plus son influence est grande : en effet, l'intervalle de confiance sur la prédiction de cet exemple, si on l'enlève de la base d'apprentissage, devient très grand, puisqu'il tend vers l'infini lorsque $h_{ii}$ tend vers 1. Cela signifie que le modèle a utilisé certains degrés de liberté spécifiquement, de façon à s'ajuster au plus près à cet exemple ($R_i$ très petit). Lorsque cet exemple est supprimé de la base d'apprentissage, le modèle "ne sait plus quoi faire" de ces degrés de libertés, ce qui se traduit, localement, par des intervalles de confiance très élevés.

L'indétermination provoquée par un exemple de forte influence, pour lequel $h_{ii}$ tend vers 1 et son résidu $R_i$ tend vers 0, peut être partiellement levée. En effet, on peut supposer - sauf cas pathologique - que le retrait de la base d'apprentissage d'un exemple à forte influence fait varier la valeur de sortie du modèle pour cet exemple. Cela signifie que le rapport $h_{ii} \frac{R_i}{1-h_{ii}}$ ne tend en général pas vers 0. On est donc sûr qu'il en va de même pour l'estimation de l'erreur de prédiction de cet exemple $R_i^{(-i)} \cong \frac{R_i}{1-h_{ii}}$. Cette remarque est très importante pour la suite car elle signifie que, pour un exemple à forte influence, le résidu $R_i$ tend en général vers 0 **moins rapidement** que $1 - h_{ii}$.

On peut utiliser le tableau précédent de deux manières :

- ces formules peuvent servir d'outil pour valider les résultats obtenus par une procédure conventionnelle de leave-one-out (qui implique la réalisation de $N$ apprentissages distincts si l'on dispose de $N$ exemples) ; nous expliquerons ceci dans le paragraphe suivant,
- elles peuvent également éviter au concepteur de réaliser le leave-one-out conventionnel ; la sélection de modèles se fonde alors sur *l'estimation* de l'effet du retrait d'un exemple de la base d'apprentissage. Cette application sera détaillée dans les chapitres 4 et 5 de ce mémoire.

## 3.4 Validation des résultats de leave-one-out

Nous allons montrer que les résultats présentés dans le paragraphe précédent peuvent s'avérer très utiles pour vérifier la validité de l'estimation des performance obtenues en effectuant la procédure conventionnelle de leave-one-out.

Pour ce faire, rappelons que l'hypothèse implicite de la procédure de leave-one-out est que la suppression d'un exemple de la base d'apprentissage n'affecte pas de manière importante l'estimation des paramètres d'un modèle, c'est-à-dire que les solutions des moindres carrés $\boldsymbol{\theta}_{LS}^{(-i)}$ sont très proches de la solution $\boldsymbol{\theta}_{LS}$ obtenue sur l'ensemble des données disponibles ; par conséquent, tous les minima des fonctions de coût $J^{(-i)}$ - obtenus à l'issue des $N$ apprentissages - devraient se trouver dans une petite région de l'espace des paramètres. En pratique, la validation de résultats en leave-one-out devrait obligatoirement passer par cette vérification.

Vérifier cette propriété en comparant entre elles les distances $\left\| \boldsymbol{\theta}_{LS} - \boldsymbol{\theta}_{LS}^{(-i)} \right\|_{i = 1, \ldots, N}$ nécessiterait la définition d'un seuil à partir duquel on considère que la solution $\boldsymbol{\theta}_{LS}^{(-i)}$ est trop éloignée de $\boldsymbol{\theta}_{LS}$. Une méthode plus satisfaisante consiste à vérifier qu'en remettant l'exemple $i$ dans la base d'apprentissage, et en poursuivant l'apprentissage à partir de $\boldsymbol{\theta}_{LS}^{(-i)}$, celui-ci converge de nouveau vers $\boldsymbol{\theta}_{LS}$. Dans la suite de ce mémoire, ceci tiendra lieu de définition.

Définition :

> Un minimum $\boldsymbol{\theta}_{LS}$, de l'erreur quadratique moyenne $J$ sur un ensemble de $N$ observations, est dit ***propice au leave-one-out*** si et seulement si $\forall i \in [1, \ldots, N]$, la poursuite de l'apprentissage à partir de $\boldsymbol{\theta}_{LS}$, en retirant l'exemple $i$ de la base d'apprentissage jusqu'à convergence vers un minimum $\boldsymbol{\theta}_{LS}^{(-i)}$, puis en réintégrant l'exemple $i$ dans la base d'apprentissage, fait revenir celui-ci à $\boldsymbol{\theta}_{LS}$.

Cette définition est illustrée graphiquement sur les figures 3.2 et 3.3.

Notons dès à présent un point fondamental : il ne faut pas confondre cette définition avec les notions de stabilité introduites pour obtenir des bornes sur l'estimation des performances de généralisation (voir paragraphe 2.7). En effet, la propriété de stabilité concerne un algorithme d'apprentissage, indépendamment des données utilisées, et donc des différents minima de la fonction de coût.

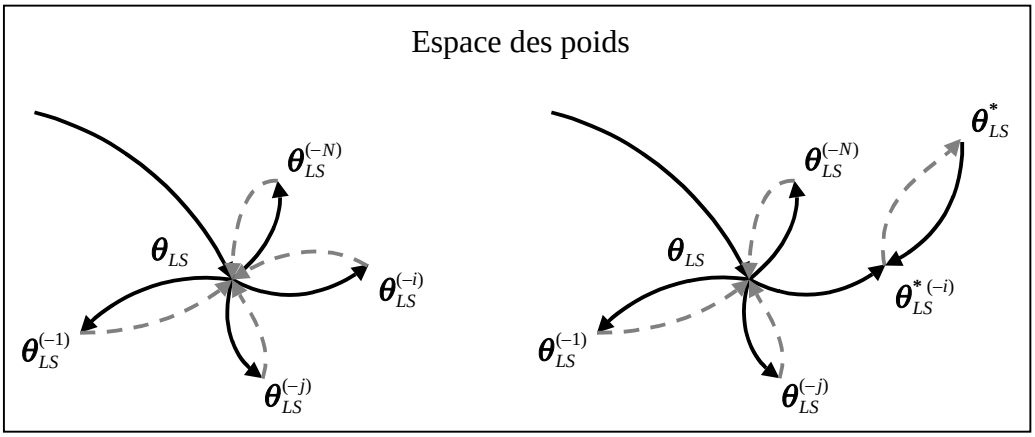


*Figure 3.2 : Minima respectivement propice (à gauche) et non propice (à droite) au leave-one-out*

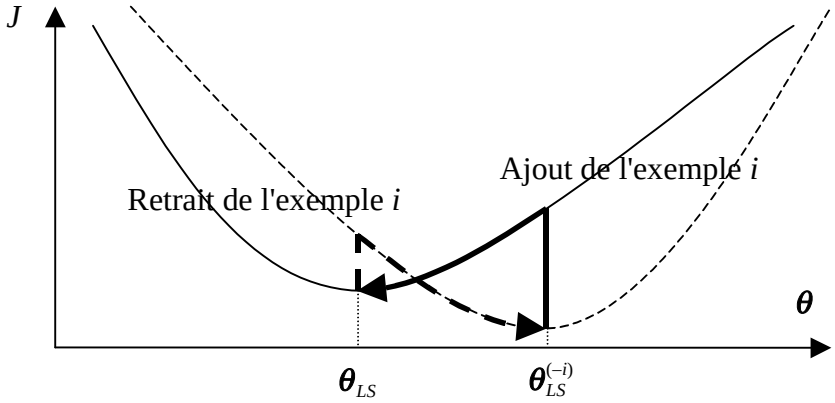


*Figure 3.3 : Retrait et ajout de l'exemple i de la base d'apprentissage : cas d'un minimum propice au leave-one-out*

Nous avons introduit cette définition pour déceler les minima pour lesquels il n'est pas raisonnable d'estimer des performances de leave-one-out par apprentissage. Néanmoins, ceci ne signifie pas que, pour un minimum propice au leave-one-out, l'erreur quadratique moyenne obtenue par leave-one-out est nécessairement une bonne estimation des performances de généralisation. Nous reviendrons sur ce point dans le paragraphe 4.4.2.

Nous allons montrer dans ce paragraphe :

- que tous les minima ne sont pas forcément propices au leave-one-out,
- que les inégalités (2.2) à (2.4), couplées aux formules (3.9) et (3.10), permettent en partie de s'assurer qu'un minimum est propice au leave-one-out,

### *3.4.1 Interprétation géométrique de l'estimation des performances en leave-one-out*

Si l'on suppose que le modèle possède un terme constant, ce qui est généralement le cas, les propriétés (2.2) à (2.4) s'écrivent :

$$\frac{1}{1-h_{ii}} \geq \frac{N}{N-1} \geq 0 \tag{3.16}$$

En combinant (3.16) aux relations (3.9) et (3.10), on obtient les deux inégalités suivantes :

$$J(\boldsymbol{\theta}_{LS}^{(-i)}) \leq J(\boldsymbol{\theta}_{LS}), \tag{3.17}$$

$$\left(R_i^{(-i)}\right)^2 \geq \frac{N}{N-1}\left(J(\boldsymbol{\theta}_{LS}) - J(\boldsymbol{\theta}_{LS}^{(-i)})\right). \tag{3.18}$$

Ces deux conditions sont illustrées graphiquement sur la figure 3.4, qui représente $\left(R_i^{(-i)}\right)^2$ en fonction de $J(\boldsymbol{\theta}_{LS}^{(-i)})$. Sur un tel graphe, à partir d'un modèle dont l'apprentissage a été effectué avec l'ensemble de la base d'apprentissage, les points $\{M_i\}_{i = 1, ..., N}$, représentant les $N$ modèles obtenus après suppression d'un exemple, devraient tous se situer dans le secteur angulaire représenté par la zone grisée.

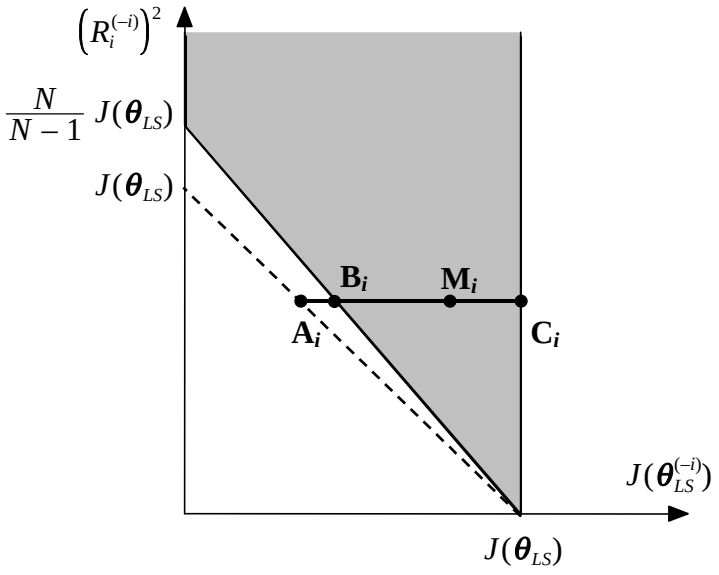


*Figure 3.4 : Localisation théorique des performances en leave-one-out*

Le graphique de la figure 3.4 permet également d'interpréter graphiquement les $\{h_{ii}\}_{i = 1, ..., N}$. On démontre en effet grâce à (3.9) et (3.10) la relation géométrique suivante :

$$\frac{\left|A_iM_i\right|}{\left|A_iC_i\right|} = \frac{1}{N} + \frac{\left|B_iM_i\right|}{\left|A_iC_i\right|} \cong h_{ii}. \tag{3.19}$$

En d'autres termes, $h_{ii}$ est le rapport suivant lequel le point $M_i$ partage le segment horizontal $[A_i\ C_i]$. Si le point $M_i$ est à la limite gauche du secteur angulaire, c'est-à-dire confondu avec $B_i$, alors $h_{ii} = \frac{1}{N}$.

Nous allons montrer que la présence de tous les points obtenus après suppression d'un exemple à l'intérieur d'un tel secteur angulaire est une **condition nécessaire** pour qu'un minimum soit propice au leave-one-out. En effet, en raisonnant par l'absurde, deux cas se présentent (cf. figure 3.5) :

1. supposons qu'il existe un exemple $i_0$ dont les performances d'apprentissage sont situées à droite du domaine : alors l'apprentissage correspondant n'a pas convergé vers le bon minimum. En effet, si l'on remettait $i_0$ dans la base d'apprentissage, la minimisation de la fonction de coût conduirait forcément à une solution pour laquelle la fonction de coût serait supérieure à $J(\boldsymbol{\theta}_{LS}^{(-i_0)})$.

2. supposons qu'il existe un exemple $i_1$ dont les performances d'apprentissages sont situées à gauche du domaine. Cela signifie qu'il existe un autre minimum $\boldsymbol{\theta}^*_{LS}$ tel que $J(\boldsymbol{\theta}^*_{LS}) < J(\boldsymbol{\theta}_{LS})$. En effet, considérons le modèle $f(\boldsymbol{x}, \boldsymbol{\theta}^*_{LS}) = f(\boldsymbol{x}, \boldsymbol{\theta}_{LS}^{(-i_1)}) + \frac{R_{i_1}^{(-i_1)}}{N}$. On montre facilement que la fonction de coût correspondante est majorée par $J(\boldsymbol{\theta}_{LS}^{(-i_1)}) + (N-1)\left(\frac{R_{i_1}^{(-i_1)}}{N}\right)^2 + \left((1-\frac{1}{N})\, R_{i_1}^{(-i_1)}\right)^2$. Le minimum $\boldsymbol{\theta}^*_{LS}$ vérifie donc :

$$J(\boldsymbol{\theta}_{LS}^{(-i_1)}) \le J(\boldsymbol{\theta}^*_{LS}) \le J(\boldsymbol{\theta}_{LS}^{(-i_1)}) + \frac{N-1}{N}\left(R_{i_1}^{(-i_1)}\right)^2 < J(\boldsymbol{\theta}_{LS}) \,. \tag{3.20}$$

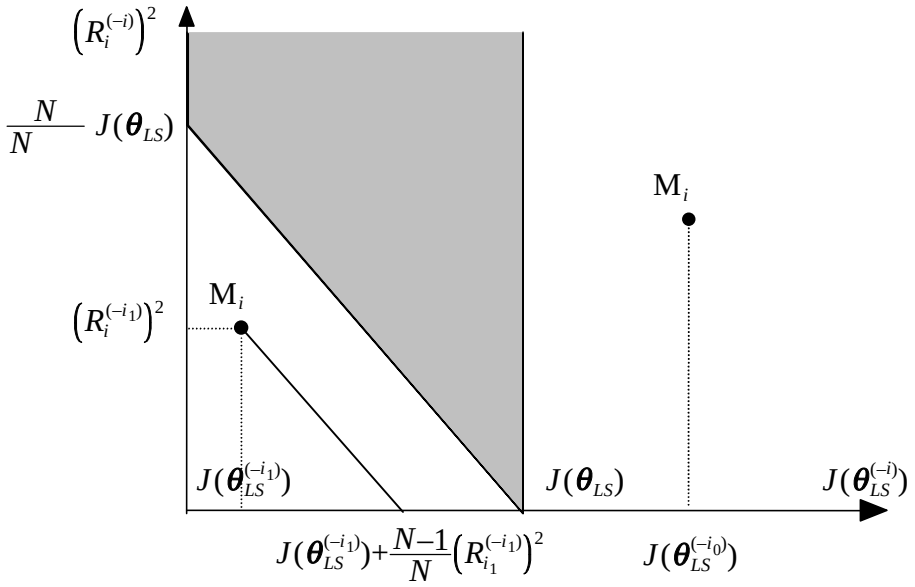


*Figure 3.5 : Exemples correspondant à des performances situées en dehors du secteur angulaire*

Cette interprétation géométrique montre qu'il est toujours possible, pour la meilleure solution trouvée $\boldsymbol{\theta}_{LS}$ (au sens du minimum de la fonction de coût), de placer toutes les performances obtenues après retrait d'un exemple dans le secteur angulaire de $J(\boldsymbol{\theta}_{LS})$. Il suffit pour cela de respecter les deux règles suivantes lors de l'application du leave-one-out :

Règle n°1 : démarrer les $N$ apprentissages à partir du meilleur minimum atteint sur l'ensemble des exemples d'apprentissage, en supposant qu'il s'agit du minimum global,

Règle n°2 : vérifier - graphiquement - que les performances des $N$ modèles obtenus se situent bien à l'intérieur d'un secteur angulaire tel que celui défini précédemment. Dans le cas contraire, c'est-à-dire si un point correspondant à un exemple $i_1$ se situe à gauche du secteur angulaire, redémarrer un apprentissage sur les $N$ exemples à

partir des poids $\boldsymbol{\theta}_{LS}^{(-i_1)}$. Nous venons en effet de démontrer que cet apprentissage convergera vers un autre minimum, situé plus bas que le précédent.

Dans [Moody 94], les auteurs affirment que la première de ces deux règles est suffisante pour s'assurer de la convergence des *N* apprentissages vers le "même" minimum. Ceci est inexact car - même en respectant également la deuxième règle - rien n'assure que l'apprentissage converge de nouveau vers $\boldsymbol{\theta}_{LS}$ en remettant chaque exemple dans la base d'apprentissage.

Remarques :

- nous avons fait l'hypothèse que le modèle possède un biais. Dans le cas contraire, la seule différence réside dans la pente de la droite délimitant - à gauche - le secteur angulaire : celle-ci est égale à 1 au lieu de $\frac{N}{N}$ ; tout le reste du raisonnement est identique.
- la démonstration précédente suppose qu'avec les algorithmes d'apprentissage utilisés, la fonction de coût est une fonction monotone décroissante du nombre d'itérations. Ceci est le cas des algorithmes utilisés dans le cadre de ce travail (Quasi-Newton et Levenberg-Marquardt), mais exclut des méthodes du type recuit simulé.

Cette interprétation géométrique est donc une condition **nécessaire** à l'estimation de la performance de généralisation d'un modèle par la méthode du leave-one-out. Si les *N* apprentissages d'un leave-one-out n'ont pas tous convergé vers des solutions situées à l'intérieur d'un tel secteur angulaire, l'estimation correspondante des performances n'a aucun sens. Cette condition n'est cependant pas **suffisante** pour assurer que l'on considère bien des solutions ayant convergé vers le même minimum. En effet, les secteurs angulaires correspondant à deux minima distincts $\boldsymbol{\theta}_{LS}$ et $\boldsymbol{\theta}^*_{LS}$ ont toujours une intersection non vide.

### *3.4.2 Limite de l'approche : cas du retrait d'un exemple avec forte influence*

Nous venons de montrer que, même en appliquant les deux règles précédentes de manière à ce que toutes les performances après retrait d'un exemple soient situées à l'intérieur d'un secteur angulaire, la seule façon de s'assurer qu'un minimum est propice au leave-one-out est, à partir de chaque solution $\boldsymbol{\theta}_{LS}^{(-i)}$, de remettre l'exemple *i* dans la base et de vérifier que la poursuite de l'apprentissage converge de nouveau vers $\boldsymbol{\theta}_{LS}$. Trois cas peuvent alors se présenter :

1. tous les apprentissages reviennent à $\boldsymbol{\theta}_{LS}$, auquel cas le minimum est propice au leave-one-out,
2. un apprentissage converge vers un minimum $\boldsymbol{\theta}^*_{LS}$ situé plus bas que $\boldsymbol{\theta}_{LS}$ ; ce dernier n'est donc pas propice au leave-one-out. Ce cas est le même que lorsque les performances correspondant au retrait d'un exemple se situaient à gauche du secteur angulaire, sauf que ceci ne peut pas se détecter graphiquement. De même qu'au paragraphe 3.4.1, il est possible de recommencer les apprentissages avec retrait d'un exemple à partir de ce nouveau minimum,

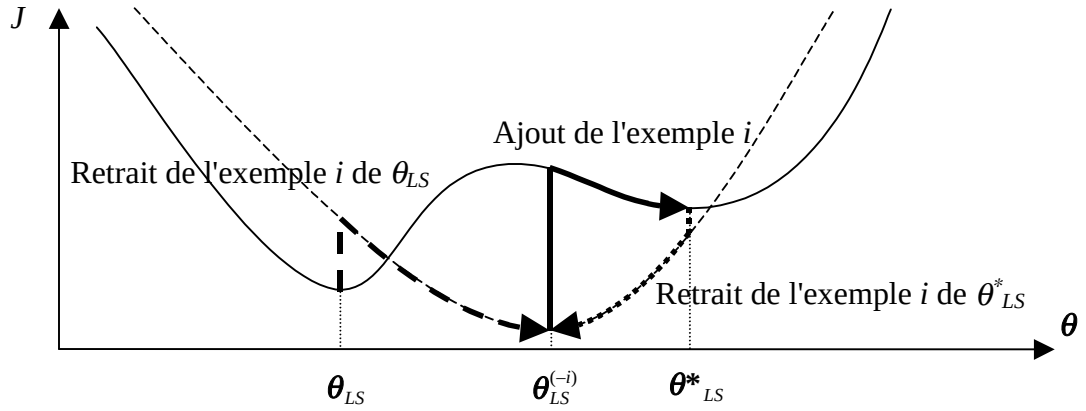


*Figure 3.6 : Retrait et ajout de l'exemple i de la base d'apprentissage : cas d'un minimum non propice au leave-one-out*

3. un apprentissage converge vers un minimum $\boldsymbol{\theta}^*_{LS}$ situé plus haut que $\boldsymbol{\theta}_{LS}$. Cela signifie que la solution $\boldsymbol{\theta}_{LS}$ n'existe que lorsque l'exemple en question se trouve dans la base d'apprentissage (voir figure 3.6). On ne peut donc pas connaître l'effet du retrait de l'exemple *i* sur la solution $\boldsymbol{\theta}_{LS}$ mais uniquement sur $\boldsymbol{\theta}^*_{LS}$ : $\boldsymbol{\theta}^{(-i)}_{LS}$ est en réalité $\boldsymbol{\theta}^{*\,(-i)}_{LS}$.Cette solution n'est donc pas propice au leave-one-out.

Intuitivement, on peut s'attendre à rencontrer ce dernier cas plus particulièrement lors du retrait d'un exemple dont l'influence sur les poids du modèle est grande. Illustrons ceci par un exemple : celui du modèle représenté sur la figure 3.7.a. L'exemple *i* est celui dont l'influence sur les poids du modèle est la plus grande, en l'occurrence $h_{ii} = 0.944$.

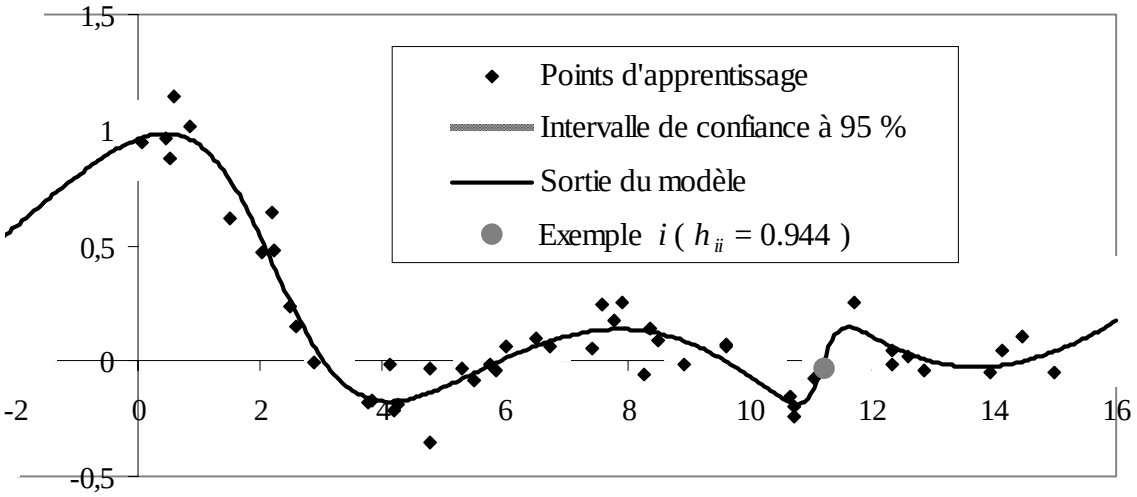


*Figure 3.7.a : Exemple de modèle avec grande influence de certains points*

Nous allons étudier en détail le retrait de cet exemple, suivant que ce retrait est calculé à partir des formules de linéarisation ou qu'il est effectué par la poursuite de l'apprentissage, en respectant les règles définies au paragraphe précédent.

Examinons, dans le tableau 3.2, les erreurs de prédiction sur cet exemple, obtenues respectivement par apprentissage et par utilisation de la formule (3.9) : l'erreur calculée est

environ trois fois plus grande que l'erreur constatée après apprentissage. Cette différence se reflète également sur la valeur de la fonction de coût : le coût obtenu par apprentissage est significativement plus élevé que celui prédit par la formule (3.10).

| | Poursuite de l'apprentissage | Formule de linéarisation |
|---|---|---|
| $\left(R_i^{(-i)}\right)^2$ | -0.277 | -0.770 |
| $J(\boldsymbol{\theta}_{LS}^{(-i)})$ | 0.358 | 0.333 |

*Tableau 3.2 : Comparaison entre poursuite de l'apprentissage et utilisation des formules de linéarisation pour l'exemple i sorti*

Cette différence sur les performances du modèle après retrait de l'exemple *i* est naturellement visible lorsque l'on considère (figure 3.7.b) les modèles obtenus respectivement :

1. par poursuite de l'apprentissage,
2. par linéarisation de la sortie du modèle au voisinage de $\boldsymbol{\theta}_{LS}$, c'est-à-dire :

$$f(\boldsymbol{x}, \boldsymbol{\theta}_{LS}^{(-i)}) \cong f(\boldsymbol{x}, \boldsymbol{\theta}_{LS}) + \left.\frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta} = \boldsymbol{\theta}_{LS}} \left(\boldsymbol{\theta}_{LS}^{(-i)} - \boldsymbol{\theta}_{LS}\right), \tag{3.21}$$

expression dans laquelle la différence $\boldsymbol{\theta}_{LS}^{(-i)} - \boldsymbol{\theta}_{LS}$ est calculée par la formule (3.8).

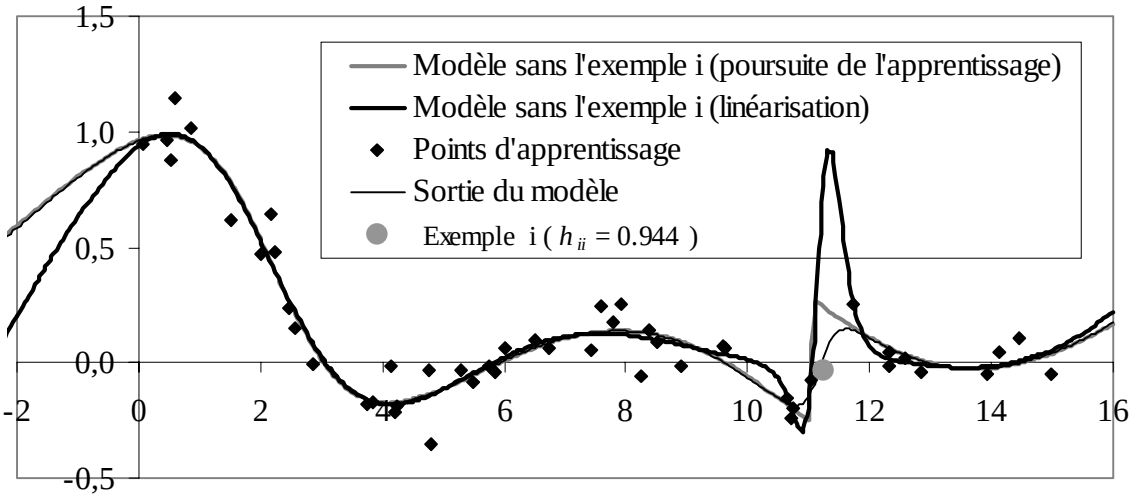


*Figure 3.7.b : Effet du retrait d'un exemple à forte influence sur les modèles obtenus respectivement par apprentissage et par linéarisation*

La figure 3.7.b permet de constater que la poursuite de l'apprentissage, après avoir éliminé l'exemple *i*, ne s'est traduite que par un ajustement local du modèle. En dehors de l'intervalle des entrées [10 ; 12], le modèle n'a pratiquement pas été modifié. En revanche, pour la solution obtenue par linéarisation, les modifications se situent certes au voisinage de l'exemple *i*, mais également en dehors de la plage des entrées définie par l'ensemble des points

d'apprentissage : ceci nous permet d'appréhender réellement ce qui se passe lorsqu'on "relâche" l'influence de cet exemple sur l'estimation de l'ensemble des poids du modèle.

Certes, cette différence est sans doute en partie due à une distance trop grande entre $\boldsymbol{\theta}_{LS}$ et $\boldsymbol{\theta}_{LS}^{(-i)}$, rendant imprécise l'utilisation du développement limité (3.21). Nous allons voir que ceci ne constitue pas l'explication principale des différences observées sur la figure 3.7.b.

En effet, considérons, sur la figure 3.7.c, la localisation des performances des modèles après retrait d'un exemple : celles-ci se situent effectivement dans un secteur angulaire tel que celui de la figure 3.4, ce qui est normal compte tenu du fait que nous avons procédé en respectant les règles définies au paragraphe 3.3.1.

Sur le graphique 3.7.c, nous avons en outre représenté l'effet du retour dans la base d'apprentissage de l'exemple qui en avait été retiré : ceci est illustré graphiquement par un trait horizontal qui part du point correspondant à l'exemple et dont la longueur est égale à $\left| J(\boldsymbol{\theta}_{LS}^{*}) - J(\boldsymbol{\theta}_{LS}^{(-i)}) \right|$. Ainsi, la valeur de la fonction de coût, pour le minimum atteint après retour d'un exemple dans la base d'apprentissage, se lit en abaissant la parallèle à l'axe des ordonnées passant par l'extrémité droite du trait. Pour des raisons de lisibilité du graphique, ce trait n'est représenté que dans le cas où le minimum atteint lors du retour diffère du minimum correspondant au secteur angulaire.

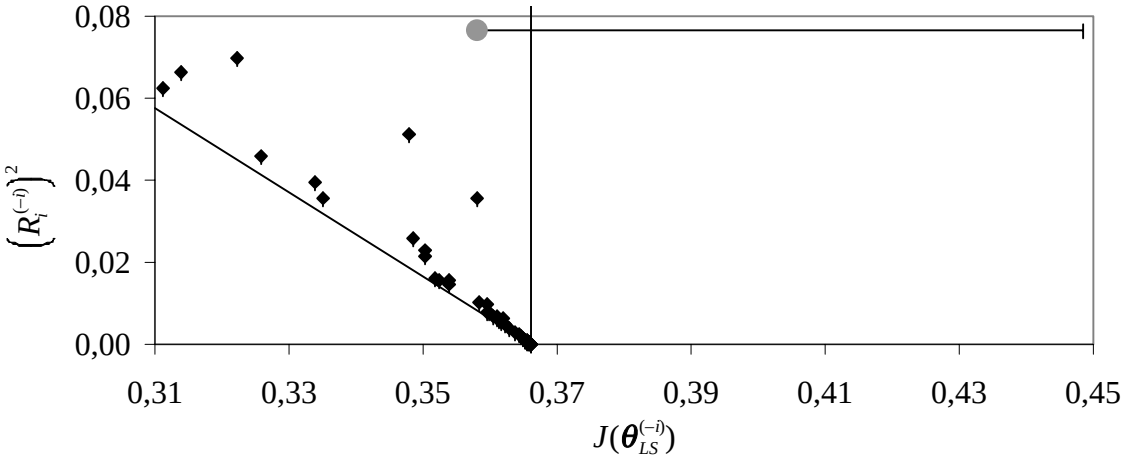


*Figure 3.7.c : Secteur angulaire et localisation des performances des N modèles après apprentissage*

Dans le cas du modèle précédent, il apparaît qu'il n'y a qu'un seul point dont le retour dans la base d'apprentissage provoque une convergence vers un autre minimum situé plus haut que le précédent. Cela signifie que cette solution n'est pas propice au leave-one-out. Or il s'avère que ce point correspond justement à l'exemple *i*, dont l'influence sur l'estimation des coefficients du modèle est maximale et pour lequel nous avions mentionné les différences dans le tableau 3.2.

Il apparaît ainsi que les différences entre les erreurs de prédiction d'un exemple à forte influence, lorsque celui-ci n'est pas dans la base d'apprentissage, ne suffisent pas remettre en cause la précision des formules de linéarisation (3.9) et (3.10). En effet, si le retrait d'un tel

exemple fait converger la minimisation du coût quadratique vers un autre minimum (local), les deux approches ne peuvent pas être comparées l'une à l'autre.

Cet exemple nous a permis de mettre en évidence l'existence de minima non propices au leave-one-out, ce qui se traduit par des différences entre les modèles obtenus après suppression d'un exemple, suivant que l'on supprime cette influence à partir des formules de prédiction ou par poursuite de l'apprentissage, même en validant ce dernier par le critère du secteur angulaire. Notre expérience montre que ceci se produit très souvent lors du retrait d'exemples qui ont une forte influence sur l'estimation des coefficients du modèle.

## 3.5 Conclusion

Nous avons montré qu'un développement de Taylor du premier ordre au voisinage de la solution des moindres carrés permet d'estimer l'effet du retrait d'un exemple à la fois sur la sortie du modèle (et donc sur son erreur de prédiction donnée par la formule (3.9)) et sur l'intervalle de confiance sur la sortie du modèle (3.15).

Ces estimations sont toutes fondées sur la grandeur $h_{ii}$, c'est-à-dire sur le terme diagonal de la matrice de projection orthogonale sur le sous-espace des solutions, qui n'est défini que lorsque le rang de la matrice jacobienne $Z$ est égal au nombre de paramètres ajustables du modèle. Nous avons en outre interprété $h_{ii}$ comme une mesure de l'influence de l'exemple $i$ sur l'estimation des paramètres du modèle.

Afin de pouvoir comparer les résultats obtenus par apprentissage et par utilisation des formules fondées sur le développement de Taylor, nous avons montré qu'en dépit du respect d'une procédure adéquate, certains minima de la fonction de coût quadratique n'étaient pas propices au leave-one-out. Il s'agit des minima qui ne sont pas conservés par le retrait d'un exemple à forte influence. Pour ces minima, l'utilisation des formules de linéarisation semble être le seul moyen d'estimer l'effet du retrait d'un tel exemple sur les paramètres du modèle.

Nous verrons dans le chapitre 4 comment tout ceci doit être pris en considération lors de la sélection de modèles neuronaux.

# 4 UTILISATION DU LEAVE-ONE-OUT POUR LA SELECTION DE MODELES

## Résumé

*Sauf dans le cas de modèles linéaires par rapport aux paramètres, sélectionner le modèle optimal ne se résume pas au choix de l'architecture, c'est-à-dire de la famille de fonctions paramétrées. En effet, le coût quadratique présentant plusieurs minima, il convient également de sélectionner l'initialisation aléatoire des coefficients conduisant au "meilleur" de ces minima.*

*Dans un premier temps, nous étudions donc - à architecture donnée - les différentes façons de procéder à ce choix de modèles dans le cadre du leave-one-out. Il apparaît que l'utilisation de l'EQMA et de l'erreur de généralisation $E_a$ obtenue en procédant au leave-one-out par apprentissage conduit dans de nombreux cas à la sélection de modèles dont nous avons prouvé le surajustement. La notion de minima propices ou non permet d'expliquer la défaillance de la mise en œuvre classique du leave-one-out.*

*Nous préconisons finalement d'utiliser comme critère de sélection l'erreur de généralisation $E_p$ obtenue en procédant au leave-one-out par utilisation des formules de prédiction de l'effet du retrait d'un exemple.*

*Dans un second temps, nous montrons que la sélection de modèles - à partir des modèles sélectionnés sur la base de $E_p$ pour chaque architecture - conduit également à de très bons résultats sur les deux exemples étudiés puisque $E_p$ se stabilise, à partir de l'architecture optimale, autour de l'écart-type du bruit de sortie.*

*L'utilisation des formules tirées du développement de Taylor permet ainsi de remplacer avantageusement la procédure classique de leave-one-out, à la fois en termes de résultats, puisque nous avons montré des situations d'échec de cette procédure, mais également en termes de temps de calcul, puisqu'il n'est plus nécessaire d'effectuer autant d'apprentissages que d'exemples.*

*En complément de l'erreur $E_p$, nous introduisons le paramètre $\mu$, à partir de la moyenne des grandeurs $\left\{\sqrt{h_{ii}}\right\}_{i=1,\ldots,N}$ auxquelles est proportionnel l'intervalle de confiance sur la sortie du modèle pour l'exemple i. Ce critère apparaît comme un excellent moyen de sélectionner, parmi les architectures pour lesquelles $E_p$ est du même ordre de grandeur, celle dont la performance de généralisation, estimée sur un ensemble de test indépendant, est la meilleure.*

*Nous montrons enfin comment la sélection de modèles légèrement surajustés permet, avec l'utilisation des intervalles de confiance, de compléter localement la base d'apprentissage, et ainsi d'augmenter les performances du modèle.*

## 4.1 Introduction - définition du problème

Le leave-one-out, en tant que méthode de validation croisée (cf. chapitre 2), doit permettre d'estimer la performance de généralisation d'un modèle, et ainsi de sélectionner le meilleur

modèle parmi un ensemble de candidats, possédant éventuellement des architectures différentes. A cet effet, on estime l'erreur de généralisation d'un modèle - construit à partir d'un ensemble de *N* exemples - par la relation :

$$E_a = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(R_i^{(-i)}\right)^2} \tag{4.1}$$

L'indice $_a$ sert à rappeler que cette grandeur est calculée à partir de l'erreur de prédiction $R_i^{(-i)}$ commise, après chaque apprentissage, sur l'exemple *i* inutilisé pendant cet apprentissage. Chacun de ces *N* apprentissages est effectué en respectant les règles définies au paragraphe 3.4.1, ce qui assure que les performances obtenues se situeront toutes dans un secteur angulaire adéquat. Rappelons que la quantité $E_a$ n'a pas de sens si le minimum $\boldsymbol{\theta}_{LS}$ de la fonction de coût n'est pas propice au leave-one-out, au sens défini dans le paragraphe 3.4.1.

L'utilisation de la formule (3.9), fondée sur une linéarisation de la sortie du modèle au voisinage de la solution des moindres carrés, permet de définir une autre estimation de l'erreur de généralisation :

$$E_p = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{R_i}{1-h_{ii}}\right)^2} \tag{4.2}$$

Rappelons que cette erreur, désignée par un $_p$ car fondée sur la prédiction de l'effet du retrait d'un exemple, n'est définie que si la matrice *Z* est de rang plein ; nous avons proposé, au paragraphe 2.5.2, de vérifier cette condition via les inégalités (2.2) à (2.4).

Nous désignerons souvent ces deux estimations de l'erreur de généralisation sous le terme "score de validation croisée".

Pour compléter la réflexion initiée au paragraphe 2.7 au sujet des bornes sur la différence entre erreur théorique et erreur empirique, il est important d'insister sur le fait que nous nous servons de ces scores de validation croisée pour effectuer la sélection de modèles, sans remettre en question l'hypothèse selon laquelle ils constituent une bonne approximation de la performance de généralisation théorique du modèle. Nous reviendrons dans le paragraphe 4.5 sur cette hypothèse.

Dans ce chapitre, afin d'illustrer notre propos, nous utiliserons deux exemples :

1. le problème à une entrée et une sortie utilisé dans le chapitre 3. Il s'agit d'une base de 50 exemples tirés de la fonction $y(x) = \frac{\sin(x)}{}$, à laquelle on a ajouté un bruit gaussien de moyenne nulle et de variance $\sigma^2 = 10^{-2}$. Les entrées proviennent d'une loi uniforme dans l'intervalle [0 ; 15].
2. un problème à 5 entrées et une sortie, où la fonction de régression est un réseau de neurones à une couche de 5 neurones cachés sigmoïdaux (dont la fonction d'activation est la fonction tangente hyperbolique) et un neurone de sortie linéaire sans connexion directe avec les entrées. Les poids de ce réseau, dit réseau "maître", ont été choisis suivant une loi uniforme dans l'intervalle [-1 ; +1]. Une base de données de 300 exemples a été créée de la façon suivante :

- les entrées proviennent d'une loi uniforme dans l'intervalle [-3 ; +3], ce qui garantit, compte tenu de la valeur des poids, que le domaine non linéaire des tangentes hyperboliques est utilisé,
- un bruit gaussien de moyenne nulle et de variance $\sigma^2 = 5.10^{-2}$ a été ajouté à la sortie du réseau maître.

Dans les deux cas, l'objectif est de sélectionner - à partir de la base d'apprentissage - le meilleur modèle, c'est-à-dire celui qui présente le meilleur compromis entre performances d'apprentissage et de généralisation. Bien entendu, les erreurs de généralisation estimées de ces modèles doivent être au moins égales à l'écart-type du bruit de mesure ; c'est pourquoi il est intéressant de travailler sur des exemples simulés, pour lesquels on connaît le niveau de bruit présent.

Rappelons qu'une fois que les entrées du modèle ont été choisies, la sélection de modèle se fait en deux étapes :

- à architecture fixée, (c'est-à-dire, dans le cas d'un modèle neuronal, pour un nombre de neurones cachés donné) il faut choisir l'initialisation aléatoire des poids conduisant au "meilleur" minimum (qui n'est pas forcément le minimum global de la fonction de coût). Ceci nécessite la définition d'un critère de classification des minima, et fait l'objet des paragraphes 4.2 à 4.4.
- à partir de la "meilleure" solution de chaque architecture, il faut déterminer l'architecture - ou famille de fonctions - optimale, c'est-à-dire le nombre optimal de neurones cachés. Cette étape est décrite au paragraphe 4.5.

Le but de ce chapitre est de déterminer la meilleure façon de procéder à cette double sélection dans le cadre d'une estimation des performances de généralisation effectuée sur la base du leave-one-out. A cet effet, nous partons de la méthode classiquement utilisée et l'améliorons en expliquant puis en éliminant progressivement les difficultés rencontrées.

Les architectures étudiées sont des réseaux à une couche de neurones cachés sigmoïdaux et un neurone de sortie linéaire. Une connexion directe entre l'entrée et la sortie est utilisée pour l'exemple de la fonction $\frac{\sin(x)}{x}$. Tous les résultats présentés ci-dessous sont obtenus en calculant le gradient de la fonction de coût par rétropropagation et en minimisant la fonction de coût par l'algorithme BFGS décrit, entre autres, par [Press 98].

## 4.2 Sélection de modèle sur la base des performances d'apprentissage (pour une architecture donnée)

La manière classique de mener une procédure de leave-one-out consiste, pour chaque architecture candidate, à :

- effectuer plusieurs apprentissages avec plusieurs initialisations différentes des paramètres, à l'aide de l'ensemble des $N$ données disponibles ; parmi les modèles ainsi obtenus, en

conserver un (que nous appellerons $M_0$) sur la base de l'Erreur Quadratique Moyenne d'Apprentissage (définition : $EQMA = \sqrt{\frac{1}{N}\sum_{i=1}^{N} R_i^2}$ )

- pour chaque exemple retiré de la base d'apprentissage :
  - effectuer un apprentissage à l'aide des $N$ - 1 exemples restants, en choisissant comme paramètres initiaux les paramètres du modèle $M_0$,
  - calculer, pour ce modèle, l'erreur de prédiction sur l'exemple retiré de la base d'apprentissage.

On estime ensuite, à l'aide de la quantité $E_a$ (définie par la relation (4.1)), la performance de généralisation de l'architecture considérée.

Les travaux présentés dans le chapitre précédent montrent que l'on peut envisager de remplacer cette procédure d'estimation de $E_a$ par le simple calcul analytique de la quantité $E_p$ définie dans le paragraphe 4.1, sous réserve que $E_p$ soit une approximation satisfaisante de $E_a$.

En tout état de cause, la première étape consiste à effectuer des apprentissages avec diverses initialisations des paramètres, et à conserver un seul modèle $M_0$ (dont on estime ensuite la performance) sur la base de l'*EQMA*. Pour réaliser ceci, nous avons mis en œuvre la procédure définie dans le paragraphe 3.4.1, car celle-ci nous permet de détecter la présence éventuelle d'un minimum de la fonction de coût situé plus bas que le minimum trouvé au préalable. Nous allons exposer, dans les paragraphes suivants, plusieurs méthodes possibles pour choisir ce modèle, et nous montrerons les avantages et les limitations de chacune d'elles.

### *4.2.1 1ère méthode : choisir le modèle pour lequel l'EQMA est minimale*

Cette méthode, appliquée à l'exemple de la fonction $\frac{\sin(x)}{}$, avec une architecture à 4 neurones cachés, donne les résultats suivants (figure 4.1).

Sur cette figure, nous avons choisi de représenter la distribution des minima atteints dans le plan (*EQMA*, $E_p$), dont les deux valeurs sont simples à calculer (par opposition à $E_a$ dont le calcul - pour chaque minimum - serait très long). En ce qui concerne $E_p$, nous avons choisi par convention de représenter les minima pour lesquels la matrice Z n'est pas de rang plein (et donc $E_p$ non calculable) sur l'axe des abscisses (c'est-à-dire comme si $E_p$ était nul). Par ailleurs, pour des raisons de lisibilité du graphique, $E_p$ pouvant atteindre $10^4$, nous avons borné $E_p$ à 0,7.

Ce graphique, réalisé avec 600 initialisations aléatoires des coefficients, permet d'illustrer la grande dispersion des minima de la fonction de coût quadratique pour la fonction considérée. Par ailleurs, moins d'un tiers des minima atteints sont - sur cet exemple - de rang plein. En l'occurrence, le modèle possédant la plus petite *EQMA* (désigné par un gros point) est un modèle avec déficience du rang de *Z*, c'est-à-dire manifestement surajusté : il s'agit en fait du modèle dont nous nous étions servi dans le chapitre 2 pour illustrer un cas de surajustement avec déficience du rang (figure 2.5). Or le surajustement n'est pas détectable si on calcule le score $E_a$ de ce modèle, qui n'est que très légèrement supérieur à l'écart-type du bruit (0.112 contre 0.104).

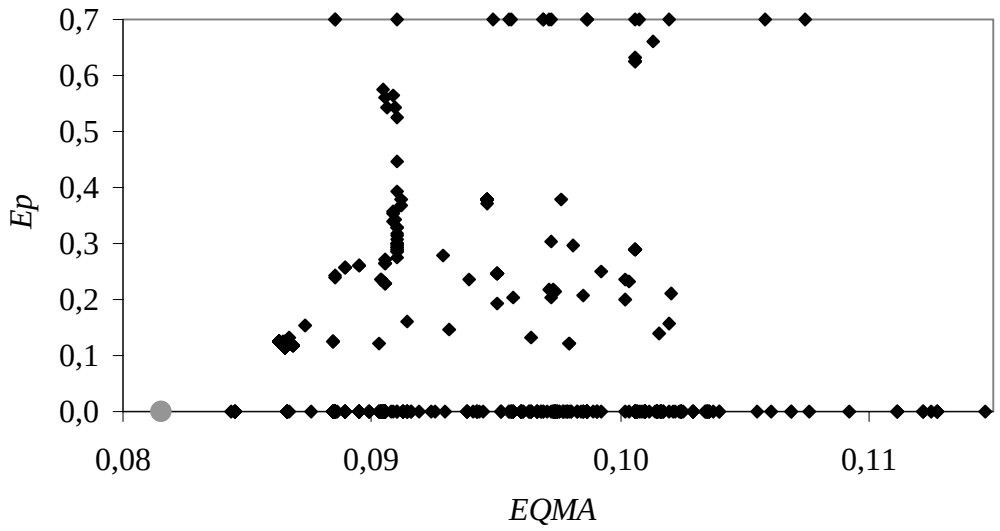


*Figure 4.1 : Distribution des minima pour une architecture à 4 neurones cachés : cas de l'exemple* $\frac{\sin(x)}{x}$

Dans le cas d'un modèle à 5 neurones cachés, le modèle possédant la plus petite *EQMA* présente également une déficience du rang. En revanche, $E_a$ (= 0.09) est significativement inférieur à l'écart-type du bruit.

Il s'avère en réalité que les deux minima précédents ne sont pas propices au leave-one-out, ôtant toute validité au calcul de $E_a$. Puisque $E_p$ n'est également pas défini, nous n'avons donc **aucun moyen de quantifier les performances de généralisation de ces modèles sur la base du leave-one-out**.

Sur le problème maître-élève, on constate le même phénomène : à partir de 5 neurones cachés, le modèle possédant la plus petite *EQMA* correspond à un minimum qui n'est pas de rang plein. Pour des architectures allant de 1 à 9 neurones cachés, la figure 4.2 présente les performances des modèles choisis sur la base de l'*EQMA* : $E_p$, dans les cas où la matrice Z est de rang plein et $E_a$, en distinguant les minima propices au leave-one-out des autres.

Cette figure montre que l'on commettrait de graves erreurs si l'on choisissait l'architecture optimale à partir des modèles sélectionnés sur la base de l'*EQMA*, en quantifiant leurs performances de généralisation par $E_a$, sans se préoccuper de savoir si les modèles en question sont propices ou non au leave-one-out. En effet, on choisirait une architecture à 9 neurones cachés (voire plus), alors que l'on sait que l'architecture optimale possède 5 neurones cachés, et notre estimation des performances de généralisation du modèle choisi serait beaucoup trop optimiste.

Par ailleurs, si l'on considère par exemple le modèle à 9 neurones cachés de la figure 4.2, qui est propice au leave-one-out malgré une déficience de rang, on constate que $E_a$ est significativement inférieur à l'écart-type du bruit (0.2 contre 0.23). Cela signifie que $E_a$ est manifestement une mauvaise estimation des performances de généralisation du modèle. Ceci ne doit pas nous surprendre : nous avons en effet introduit au paragraphe 3.4 la notion de "propice au leave-one-out" pour détecter les minima pour lesquels il n'était **pas** raisonnable de

considérer les performances obtenues par apprentissage selon le principe du leave-one-out. Nous avions bien précisé que ceci n'autorisait en aucun cas à garantir l'exactitude de l'estimation des performances dans le cas de minima propices au leave-one-out.

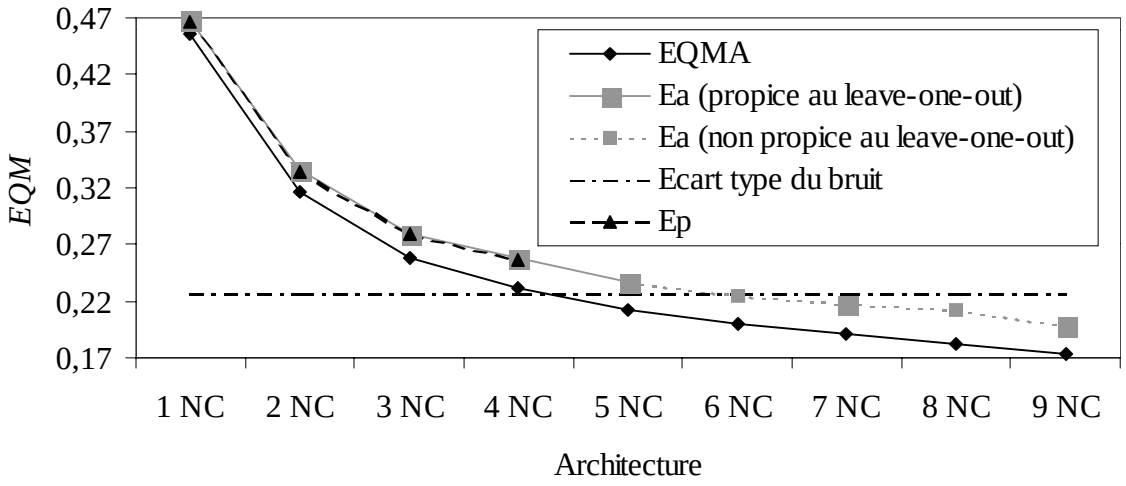


*Figure 4.2 : Sélection de modèles sur la base de l'EQMA (problème maître élève)*

Enfin, notons dès à présent que, sur l'exemple de la figure 4.2, dans le cas d'un minimum sans déficience de rang **et** propice au leave-one-out, $E_p$ est une excellente approximation de $E_a$.

Nous avons montré, sur ces deux exemples simples, deux motifs d'échec de la procédure de leave-one-out classique :

- application de la procédure à des minima pour lesquels la matrice jacobienne du modèle n'est pas de rang plein,
- application de la procédure à des minima non propices au leave-one-out.

Pour éliminer la première source d'échec, il est naturel de choisir le modèle parmi les minima pour lesquels la matrice jacobienne est de rang plein (nous les appellerons dans la suite "minima de rang plein").

### *4.2.2 2ème méthode : choisir un "minimum de rang plein" de la fonction de coût*

Comme nous l'avions fait sur la figure 4.1, nous avons représenté sur la figure 4.3 la distribution des minima de rang plein dans le cas du problème maître-élève, avec une architecture à 5 neurones cachés.

Sur cet exemple, pour lequel l'architecture du réseau "élève" correspond à celle du réseau "maître", et donc pour lequel on peut s'attendre à ne rencontrer que peu de situations de surajustement, plus de la moitié des initialisations aléatoires (sur un total de 500) ont convergé vers des minima avec déficience de rang (qui ont été éliminés de la figure 4.3).

Ici encore, notons la grande dispersion des minima atteints, surtout au niveau de l'erreur $E_p$, pour laquelle nous avons été amené à utiliser une échelle logarithmique. Sur la figure 4.3, le minimum de rang plein possédant la plus petite *EQMA* a été représenté par un gros point : il possède un score $E_p$ égal à $8.91\ 10^{1}$, soit plus de 400 fois supérieur à l'*EQMA* !

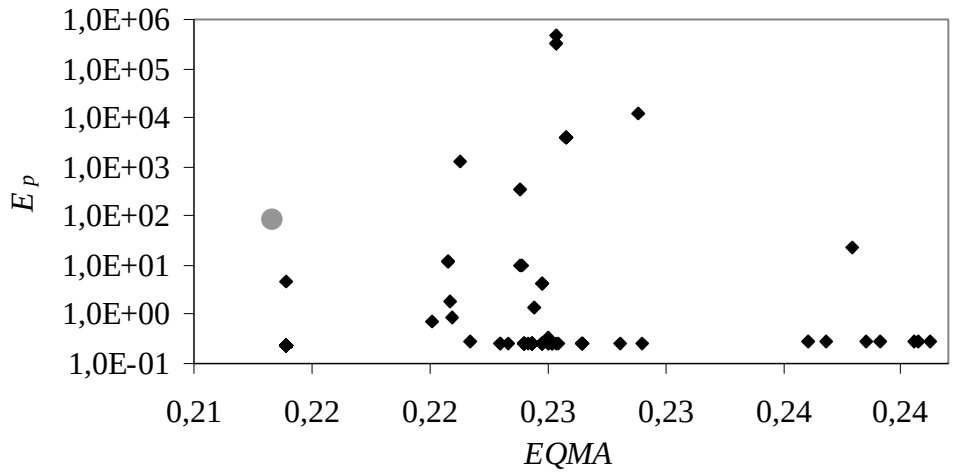


*Figure 4.3 : Distribution des minima de rang plein pour une architecture à 5 neurones cachés : cas du problème maître-élève*

Cela signifie que ce modèle est surajusté, avec des exemples à forte influence sur les coefficients ; en effet :

- certains poids du réseau sont très grands ($> 10^3$),
- plusieurs exemples présentent des résidus bien plus faibles que l'écart-type du bruit ($< 10^{-4}$) et une influence sur les poids du modèle très élevée ($h_{ii} > 0.9998$).

Tout ceci montre que, pour une architecture donnée, sélectionner les modèles qui possèdent les plus petites *EQMA* parmi les minima de rang plein ne suffit pas : il faut également prendre en considération l'amplitude des $\{h_{ii}\}_{i=1, \ldots, N}$ et se rappeler qu'un bon compromis entre biais et variance ne peut être obtenu qu'avec des modèles dont l'estimation des coefficients est influencée par l'ensemble des exemples d'apprentissage, et non pas uniquement par certains d'entre eux.

La distribution des $\{h_{ii}\}_{i=1, \ldots, N}$ ne peut pas être caractérisée par leur moyenne arithmétique, car celle-ci est toujours égale à $\frac{q}{N}$ (voir formule (2.3)). La première idée consiste à caractériser cette distribution par le moment d'ordre 2, c'est-à-dire par sa variance. Nous verrons dans le chapitre 4 qu'il est plus judicieux d'utiliser la moyenne des racines carrées.

Nous choisissons donc de caractériser la distribution des $\{h_{ii}\}_{i=1, \ldots, N}$ par la grandeur normalisée suivante :

$$\mu = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\frac{N}{q} h_{ii}} \qquad (4.3)$$

Dans le cas idéal où tous les exemples ont la même influence sur les poids du modèle, c'est-à-dire si $\forall i \in [1, ..., N] \;\; h_{ii} = \frac{q}{N}$, on a $\mu = 1$, quels que soient le nombre d'exemples $N$ et le nombre de paramètres $q$. Cette forme de normalisation permet de comparer des architectures de tailles différentes.

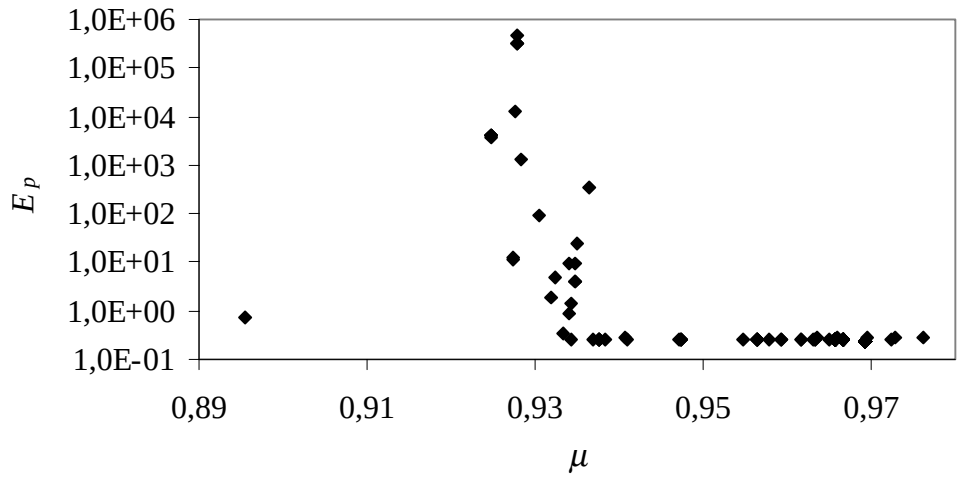


*Figure 4.4 : Répartition du couple ($\mu$, $E_p$) pour les minima de la figure 4.3*

Dans le cas des minima de la figure 4.3, nous avons représenté sur la figure 4.4 les valeurs de $E_p$ (échelle logarithmique) et du paramètre $\mu$ : les minima présentant des valeurs raisonnables de $E_p$ (c'est-à-dire du niveau de l'*EQMA*) sont ceux qui présentent les valeurs de $\mu$ les plus élevées. Nous reviendrons en détail sur ce constat et sur l'utilisation du paramètre $\mu$ à partir du paragraphe 4.4.2.

Sur l'exemple de la fonction $\frac{\sin(x)}{}$, le phénomène est moins marqué, en ce sens que les minima de rang plein possédant la plus petite *EQMA* ne présentent pas une erreur $E_p$ aussi considérable que l'exemple de la figure 4.5. Pour des architectures allant de 1 à 5 neurones cachés, la figure 4.5 présente les performances des modèles de rang plein choisis sur la base de l'*EQMA* : $E_p$ et $E_a$, en distinguant toujours les minima propices au leave-one-out des autres.

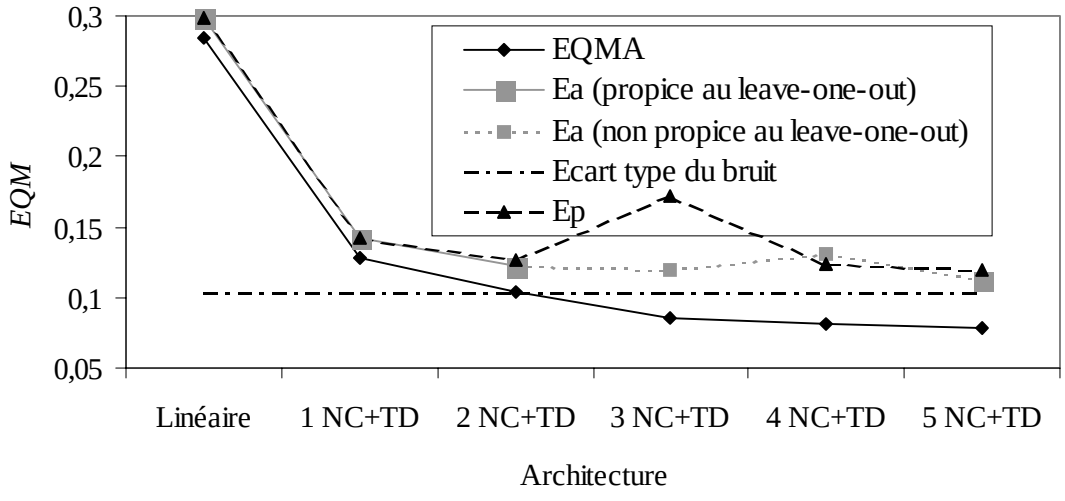


*Figure 4.5 : Sélection de modèles sur la base de l'EQMA parmi les minima de rang plein (cas de l'exemple $\frac{\sin(x)}{}$ )*

La figure 4.5 montre que si l'on devait choisir l'architecture optimale à partir des modèles de rang plein sélectionnés sur la base de l'*EQMA* et d'une quantification de leurs performances de

généralisation par $E_a$, sans se préoccuper de savoir si les modèles en question sont propices ou non au leave-one-out, on retiendrait une architecture parmi celles à 2, 3, 4 ou 5 neurones cachés pour lesquelles $E_a$ est du même ordre de grandeur.

Or les modèles à 3 et 4 neurones cachés ne sont pas propices au leave-one-out : $E_a$ n'a donc aucune signification pour ces solutions, ce qui explique, en particulier concernant le modèle à 3 neurones cachés[5], la différence avec le score estimé $E_p$.

Enfin notons ici encore que, pour les minima de rang plein et propices au leave-one-out, $E_a$ est une bonne approximation de $E_p$.

### *4.2.3 Conclusion*

La sélection de minima (de rang plein ou non) sur la base de l'*EQMA*, sans vérifier qu'ils sont propices au leave-one-out, conduit, en anticipant sur l'étape suivante, à savoir la sélection d'architecture sur la base du score de validation croisée obtenu par apprentissage ($E_a$) - aux phénomènes suivants :

- surestimation de la taille d'architecture nécessaire,
- estimation trop optimiste des performances de généralisation,
- sélection de modèles surajustés.

Ces trois problèmes, naturellement couplés, sont signalés dans la littérature ; néanmoins, [Breiman 96] ou [Moody 92]) se bornent à indiquer qu'une petite modification des données peut conduire à de grandes différences dans les minima atteints.

Nous sommes désormais capables d'en identifier plus précisément les causes :

1. une déficience éventuelle dans le rang de *Z*, qui peut conduire à un score $E_a$ très bon alors que le modèle est manifestement surajusté,
2. une mauvaise estimation de l'effet du retrait d'un exemple, lorsque celui-ci possède une forte influence sur les poids du modèle. Dans le cadre de la procédure conventionnelle de leave-one-out, une estimation correcte de l'effet du retrait successif de tous les exemples de la base d'apprentissage ne peut se faire que si le minimum est propice au leave-one-out, au sens défini dans le chapitre 3. A partir de maintenant, nous ne parlerons donc de l'erreur $E_a$ que si celle-ci est définie.

Il faudrait donc, parmi tous les minima de rang plein **et** propices au leave-one-out, chercher le modèle correspondant à l'*EQMA* la plus faible, voire directement à la plus petite valeur de $E_a$. C'est la solution que nous allons envisager dans le paragraphe suivant.

---

[5] Le modèle à 3 neurones cachés est celui présenté en exemple dans le paragraphe 3.3.2 sur les figures 3.7.a à 3.7.c. La différence entre $E_p$ et $E_a$ provient du fait que ce minimum n'est pas propice au leave-one-out, en raison de la présence d'un exemple de forte influence : l'erreur de prédiction obtenue en sortant cet exemple de la base d'apprentissage est sous-estimée par l'apprentissage, qui a convergé vers un autre minimum de la fonction de coût.

## 4.3 Sélection de modèle sur la base de $E_a$ (pour une architecture donnée)

Nous venons de montrer que l'on ne peut comparer les scores de validation croisée obtenus par apprentissage et utilisation des formules de prédiction que dans la mesure où ceux-ci sont définis. Les deux conditions à remplir, résumées dans le tableau 4.1, sont indépendantes (les exemples des figures 4.2 et 4.4 contiennent les quatre configurations possibles). En effet, le rang de *Z* est une propriété numérique de la famille de fonctions en un point de l'espace des paramètres, alors que ce sont la "topologie" de la fonction de coût au voisinage de ce point, ainsi que l'algorithme d'apprentissage utilisé, qui déterminent si un minimum est, ou non, propice au leave-one-out.

| Minimum | Z de rang plein<br>$E_p$ calculable | Z de rang non plein<br>$E_p$ non calculable |
|---|---|---|
| Propice au leave-one-out<br>$E_a$ calculable | $E_p$ est une approximation de $E_a$ | Surajustement avec déficience du rang<br>$E_p$ et $E_a$ non comparables |
| Non propice au leave-one-out<br>$E_a$ non calculable | $E_p$ et $E_a$ non comparables | Surajustement avec déficience du rang |

*Tableau 4.1 : Classification des minima d'une fonction de coût*

Il semble donc qu'il faille se restreindre aux minima de rang plein **et** propices au leave-one-out.

On peut alors se poser la question suivante : pourquoi - à architecture fixée - choisir les minima sur la base de l'*EQMA*, même en restreignant ce choix aux minima de rang pleins et propices au leave-one-out, alors que, *in fine*, la sélection d'architecture se fera à partir de l'estimation de leurs performances de généralisation. On pourrait envisager de sélectionner directement les minima sur la base de $E_a$, sous réserve que ces derniers soient propices au leave-one-out.

Malheureusement, en pratique, la recherche de minima propices au leave-one-out, à partir d'une taille d'architecture susceptible de provoquer un surajustement, devient quasiment impossible :

- les minima globaux, qu'ils soient de rang plein ou non, ne sont généralement pas propices au leave-one-out : en effet, le surajustement se traduit fréquemment par le fait que certains exemples ont une très grande influence, de sorte que le retrait puis l'ajout d'un exemple de forte influence conduit fréquemment l'apprentissage à converger vers un minimum situé plus haut que celui examiné, comme nous l'avons vu au paragraphe 3.4.2,
- les minima locaux posent également un problème, car le retrait d'un exemple fait souvent converger l'apprentissage vers un minimum situé plus bas que le précédent. De proche en

proche, on se dirige alors vers le minimum global, lui-même non propice au leave-one-out pour la raison indiquée ci-dessus.

En conclusion, cette condition est très difficile à respecter en pratique et conduit à une élimination injustifiée de la plupart des minima. Nous préconisons donc de pallier cette difficulté en estimant la performance de généralisation d'un modèle (c'est-à-dire d'un minimum) par le score de validation croisée $E_p$, les minima de rang non plein étant automatiquement rejetés. Nous avons en effet constaté, sur les deux exemples considérés, dans le cas de minima de rang plein propices au leave-one-out, que $E_p$ était une excellente approximation de $E_a$.

## 4.4 Sélection des minima sur la base de $E_p$ (pour une architecture donnée)

### *4.4.1 Qualité de la sélection*

Sur l'exemple de la fonction $\frac{\sin(x)}{}$, la sélection de modèles sur la base du score obtenu par utilisation des formules de linéarisation ($E_p$) donne d'excellents résultats (figure 4.6). Le score $E_p$ se stabilise à partir de deux neurones cachés autour d'une valeur légèrement supérieure à l'écart-type du bruit. Le modèle à deux neurones cachés est celui qui a été utilisé dans le paragraphe 3.3.1 comme illustration de la précision des formules de prédiction (figure 3.1.a).

Cette méthode donne également un résultat très satisfaisant dans le cas du problème maître-élève (figure 4.7) : $E_p$ se stabilise à partir de 5 neurones cachés et reste supérieure à l'écart-type du bruit de mesure. Par ailleurs, il est intéressant de noter que la solution choisie pour 5 neurones cachés est celle vers laquelle converge l'algorithme d'apprentissage lorsque l'on initialise les poids du réseau aux poids du réseau maître utilisé pour engendrer les données : c'est donc la meilleure solution possible. Ce modèle, ainsi que ceux qui sont sélectionnés pour les architectures de taille supérieure, correspondent tous à des minima locaux de la fonction de coût, en l'occurrence non propices au leave-one-out, ce qui ne pose ici aucun problème.

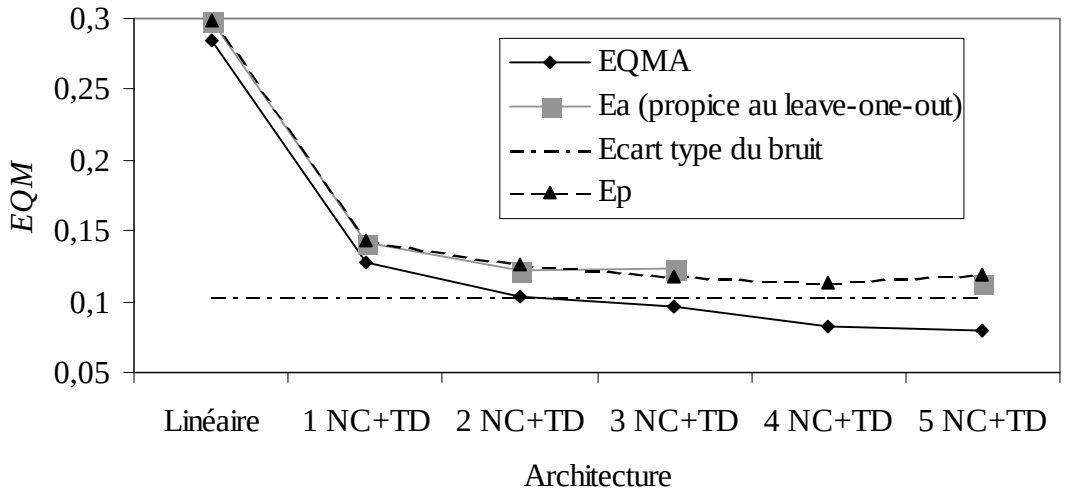


*Figure 4.6 : Sélection de modèles sur la base de $E_p$ (cas de l'exemple $\frac{\sin(x)}{x}$)*

La sélection sur la base de $E_p$ est donc efficace en ce qui concerne le choix des minima ; elle est également avantageuse en termes de temps de calcul puisqu'elle ne nécessite qu'un apprentissage sur l'ensemble des données disponibles. La seule contrainte supplémentaire est un calcul des $\{h_{ii}\}_{i = 1, \ldots, N}$, pour chaque minimum testé.

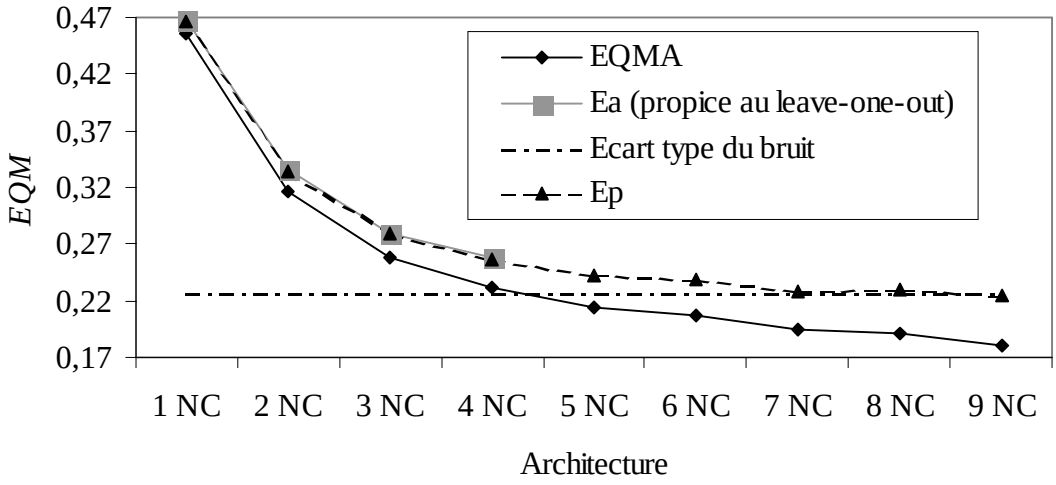


*Figure 4.7 : Sélection de modèles sur la base de $E_p$ (problème maître-élève)*

Sur les deux exemples précédents, sauf dans le cas de minima non propices au leave-one-out, $E_p$ est de nouveau une très bonne approximation de $E_a$.

Ceci est intéressant car on a consacré, pendant ces dernières années, beaucoup d'efforts pour développer des algorithmes d'apprentissage convergeant vers le minimum global de la fonction de coût (voir [Barhen 93]). Nous n'avions en effet que l'*EQMA* pour comparer les différents minima entre eux. Nous venons de montrer qu'une telle démarche n'est pas correcte, et qu'il faut, en fait, sélectionner les minima sur la base du score de validation croisée estimée. Cela signifie que les différentes initialisations des coefficients ne doivent pas servir à chercher le minimum global de la fonction de coût mais, parmi tous les minima, celui qui possède le plus petit $E_p$.

### *4.4.2 Qualité de l'estimation des performances de généralisation*

Nous l'avons annoncé au début de ce chapitre, notre objectif n'est pas de statuer, d'un point de vue théorique, sur la qualité de l'estimation des performances de généralisation par le score de validation croisée en leave-one-out. Nous avons vu dans le paragraphe 2.7 que ceci reviendrait à borner, à un certain niveau de confiance, la différence entre coûts théorique et empirique. Or, d'après l'état d'avancement actuel de ce type d'approche dans le cadre d'une estimation du coût empirique fondée sur le leave-one-out (voir [Kearns 97]), ceci nécessite de nombreuses d'hypothèses à la fois sur l'algorithme d'apprentissage, sur la fonction de coût et sur la famille de fonctions considérée.

Tout indique que, compte tenu du cadre que nous avons fixé pour ces travaux, de telles bornes ne peuvent exister sans hypothèses supplémentaires. Néanmoins, ceci ne doit pas nous empêcher d'étudier, sur les exemples présentés auparavant, les différences entre $E_p$ et une erreur quadratique moyenne de test (*EQMT*) calculée sur un ensemble de test représentatif.

Considérons par exemple - sur le problème maître-élève - un ensemble de test de 1000 exemples constitué dans les mêmes conditions que les exemples d'apprentissage (voir paragraphe 4.2). L'*EQMT*, calculée sur cette base de test dans le cas d'une sélection des minima sur la base de $E_p$, est représentée sur la figure 4.8.

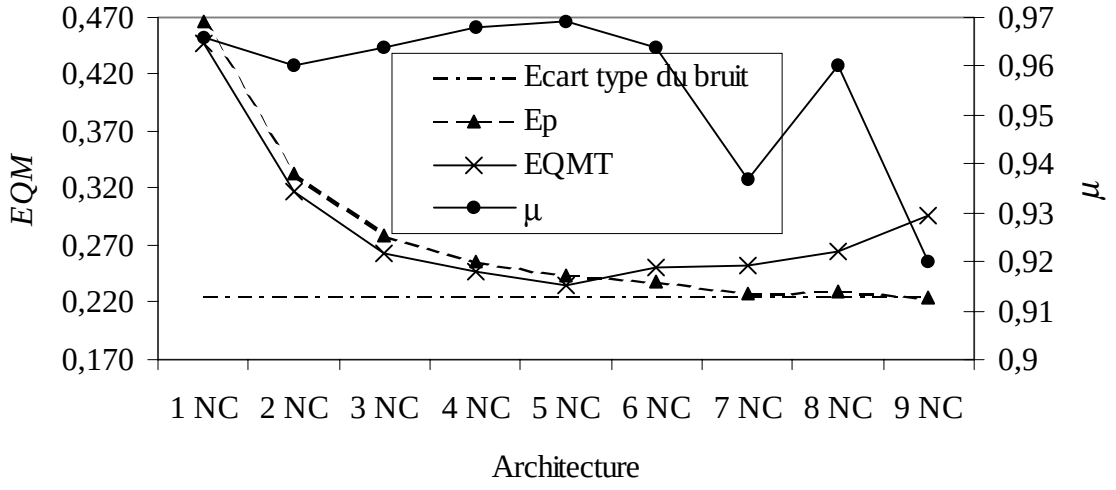


*Figure 4.8 : Comparaison entre $E_p$ et EQMT sur le problème maître-élève suite à la sélection des minima sur la base de $E_p$*

L'évolution de l'*EQMT* en fonction de l'architecture laisse apparaître clairement un minimum pour le réseau à 5 neurones cachés, c'est-à-dire pour l'architecture du réseau maître. Sur la figure 4.8 est également représenté, avec l'échelle de droite, le paramètre $\mu$. Remarquons que le minimum de l'*EQMT* correspond au maximum de $\mu$ : nous reviendrons sur ce point dans le paragraphe 4.5.

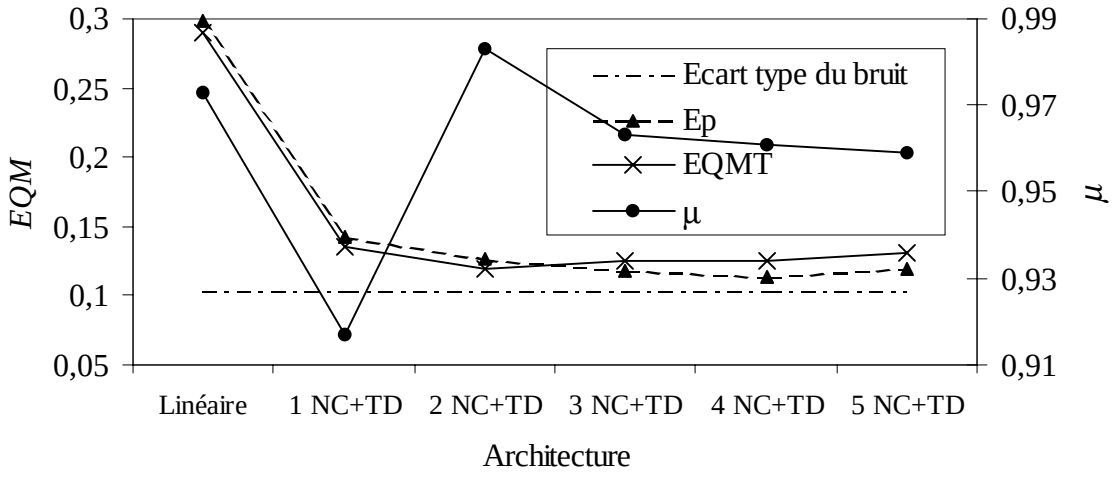


*Figure 4.9 : Comparaison entre $E_p$ et EQMT sur l'exemple $\frac{\sin(x)}{x}$ suite à la sélection des minima sur la base de $E_p$*

Cependant, pour des architectures plus grandes, $E_p$ continue de décroître très légèrement en se stabilisant autour de l'écart-type du bruit de mesure alors que les performances de test réelles se dégradent, indiquant la présence de surajustement.

Dans le cas de la fonction $\frac{\sin(x)}{x}$, reprenons les résultats de la figure 4.6 et complétons-les (figure 4.9) par une estimation de l'erreur théorique de généralisation et par la valeur de $\mu$ des modèles sélectionnés - à architecture fixée - sur $E_p$. La valeur de l'*EQMT* a été estimée à partir d'une base de test de 100 exemples constituée dans les mêmes conditions que la base d'apprentissage.

De même que sur le problème maître-élève, le minimum de l'*EQMT* correspond au maximum de $\mu$.

Finalement, sur la base de ces deux exemples, nous faisons les constatations suivantes :

- pour l'architecture correspondant au maximum de $\mu$, $E_p$ est une bonne estimation des performances réelles de généralisation du modèle. Dans le cas du problème maître-élève, nous savons qu'il s'agit effectivement de la meilleure solution possible puisque le modèle est celui vers lequel converge la minimisation du coût quadratique initialisée aux poids du réseau "maître",

- pour des architectures plus petites, $E_p$ est une estimation légèrement trop pessimiste des performances réelles de généralisation, qui se situent en général entre l'*EQMA* et $E_p$,

- pour des architectures plus grandes, $E_p$ est une estimation trop optimiste des performances réelles de généralisation. Cependant, $E_p$ reste une bonne approximation de l'écart-type du bruit de mesure.

Cette étude n'est pas seulement nécessaire pour savoir si $E_p$ approche correctement l'erreur réelle de généralisation ; c'est aussi la seule mesure raisonnable dont nous disposons pour estimer l'écart-type du bruit de mesure, nécessaire dans l'optique d'un calcul des intervalles de confiance sur la sortie du modèle.

Dans le paragraphe suivant, nous verrons comment, à partir de ces constatations, procéder à la sélection du modèle final à partir des intervalles de confiance.

## 4.5 Sélection de l'architecture optimale

Nous venons de voir que pour sélectionner un modèle parmi des candidats *d'une architecture donnée* à l'aide d'une procédure de validation croisée, l'utilisation du score $E_p$ constitue une méthode particulièrement efficace. En effet, elle permet, pour une architecture fixée, de quantifier la performance d'un modèle (obtenu en minimisant la fonction de coût) en pénalisant automatiquement les exemples dont l'influence sur l'estimation des poids du modèle est (trop) grande. En outre, il n'est pas nécessaire que les minima soient propices au leave-one-out pour pouvoir quantifier leurs performances de généralisation.

Dans ce mémoire, nous avons déjà présenté (paragraphes 2.6 et 3.2.2) la notion d'intervalle de confiance associé à la sortie d'un modèle, linéaire ou non. Cette notion peut naturellement s'appliquer à la sortie d'un réseau de neurones (voir [Rivals 98]). Nous présentons dans ce

paragraphe deux applications des intervalles de confiance, suivant qu'ils sont associés à la prédiction des exemples d'apprentissage ou d'autres exemples.

Dans un premier temps, nous allons montrer que les intervalles de confiance sur la prédiction des exemples d'apprentissage peuvent guider notre choix parmi les modèles *d'architectures différentes* qui ont été sélectionnés.

Ensuite, pour terminer ce chapitre, nous verrons que l'examen des intervalles de confiance pendant la phase d'utilisation - ou de test - d'un modèle permet d'améliorer progressivement les performances obtenues.

### *4.5.1 Utilisation des intervalles de confiance*

Face à des résultats tels que ceux de la figure 4.8 et 4.9, sans disposer de l'*EQMT*, le concepteur de modèle devra[6] choisir un modèle parmi tous ceux dont les performances estimées sont du même ordre de grandeur.

Par exemple, dans le cas de la figure 4.9, comment choisir parmi les modèles à 2 neurones cachés ou plus, dont les erreurs de généralisation estimées $E_p$ sont comprises entre 0.114 et 0.126 ? De même, sur le problème maître-élève (figure 4.8), $E_p$ varie peu (de 0.225 à 0.243) à partir de 5 neurones cachés.

Pour répondre à cette question, examinons (figure 4.10) le modèle à 4 neurones cachés et comparons-le au modèle à 2 neurones cachés déjà représenté sur la figure 3.1.a.

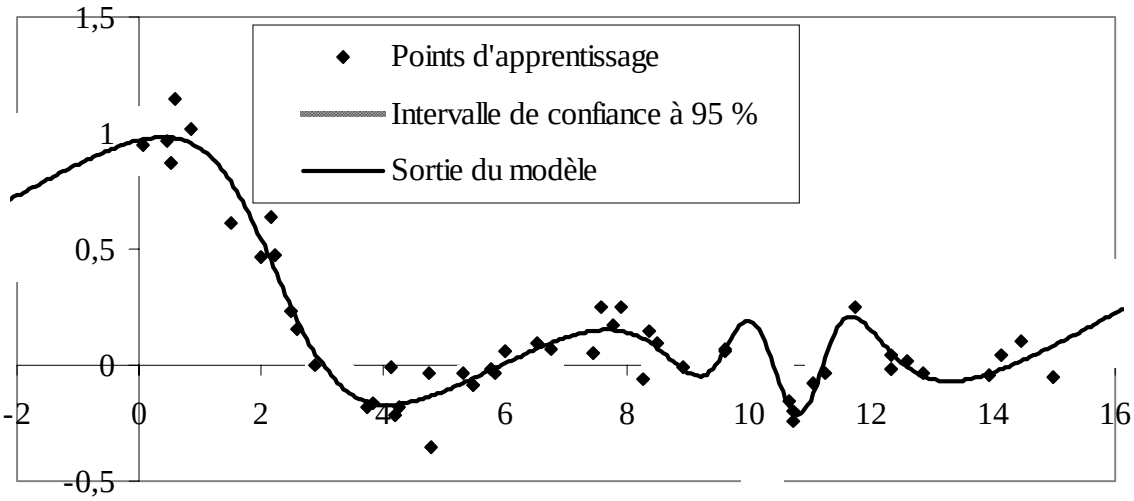


*Figure 4.10 : Modèle à 4 neurones cachés sélectionné par $E_p$ sur l'exemple* $\frac{\sin(x)}{x}$

Il apparaît que le réseau à 4 neurones cachés modélise, sur quelques exemples dans l'intervalle des entrées [9 ; 12], une tendance locale présente dans les points d'apprentissage. Cette tendance locale n'est pas prise en considération par le modèle à 2 neurones cachés. Le modèle

[6] sauf utilisation simultanée de plusieurs modèles sous forme de "comités de modèles", méthode que nous ne détaillerons pas ici.

de la figure 4.10 présente cependant une forme relativement "régulière" (au sens de la taille des coefficients du réseau), ce qui explique son bon score de validation croisée estimé $E_p$.

En considérant les intervalles de confiance, on s'aperçoit qu'ils sont - localement - significativement plus élevés pour le modèle à 4 neurones cachés. Ceci est tout à fait normal dans la mesure où les "oscillations" modélisées ne sont déterminées que par quelques points d'apprentissage.

Cette différence entre ces deux modèles est due à une répartition différente des $\{h_{ii}\}_{i = 1, \ldots, N}$, et donc de l'influence des exemples d'apprentissage, autour de leur moyenne $\frac{q}{N}$ comme l'indique la figure 4.11.

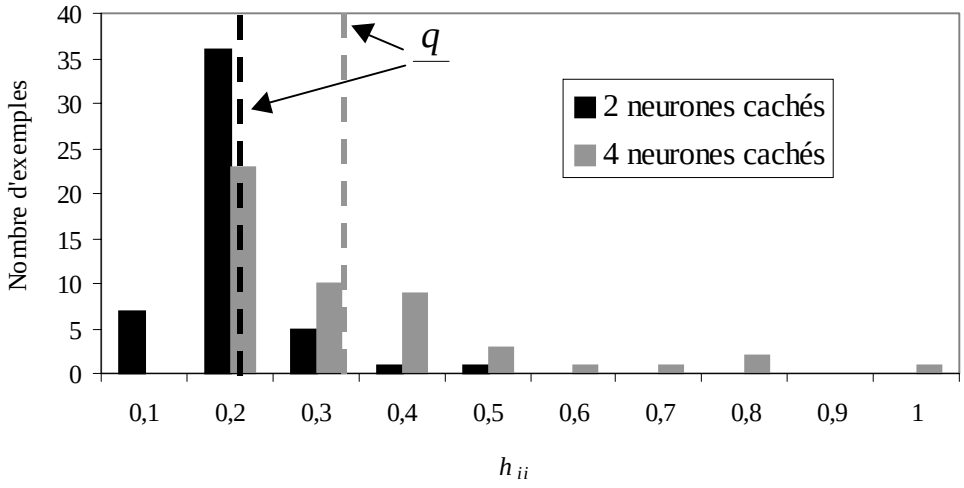


*Figure 4.11 : Distribution des* $\{h_{ii}\}_{i = 1, \ldots, N}$ *pour les modèles à* 2 *et* 4 *neurones cachés sur l'exemple* $\frac{\sin(x)}{}$

La seule façon de savoir si le modèle à 4 neurones cachés est surajusté ou non consiste à ajouter des points d'apprentissage dans le domaine des entrées où les intervalles de confiance sont jugés trop élevés. Deux cas sont à considérer :

- soit la tendance détectée sur les quelques points de la base initiale est effectivement présente dans la partie déterministe du processus à modéliser, auquel cas celle-ci sera confirmée par les points supplémentaires,
- soit cette tendance ne provenait que du bruit de mesure, auquel cas celle-ci sera infirmée par les points supplémentaires.

Dans le deuxième cas, $E_p$ était une estimation trop optimiste des performances de généralisation de ce modèle, mais, dans les deux cas, nous aurons amélioré localement la confiance sur la prédiction du modèle.

Sur le point d'apprentissage numéro *i*, nous avons montré dans le chapitre 3 (relation (3.12)), que l'intervalle de confiance sur la prédiction est proportionnel à $\sqrt{h_{ii}}$. La grandeur $\mu$ que nous avons introduite dans le paragraphe 4.2.2 correspond donc en réalité à la moyenne des intervalles de confiance - sur l'ensemble des exemples d'apprentissage - divisée par

$t_\alpha^{N-q}\, s \sqrt{q\, N}$ de manière à s'affranchir d'$\alpha$, $s$, $N$ et $q$. Nous proposons d'utiliser $\mu$ pour sélectionner le modèle correspondant le mieux à notre objectif.

Pour cela, on démontre, grâce à l'inégalité de Cauchy, la propriété suivante :

$$\forall \left\{h_{ii}\right\}_{i=1,\ldots,N} \in [0, 1] \text{ tels que } \sum_{i=1}^{N} h_{ii} = q \text{ , on a : } \sum_{i=1}^{N} \sqrt{h_{ii}} \leq \sqrt{N\, q} \qquad (4.4)$$

Le cas particulier où tous les exemples ont la même influence sur l'estimation des poids du modèle, donc pour lequel $\mu$ vaut 1, correspond ainsi à un maximum de $\mu$. Plus $\mu$ est proche de 1, plus la distribution des $\left\{\sqrt{h_{ii}}\right\}_{i=1,\ldots,N}$ est homogène et donc meilleure sera la répartition de l'influence des exemples d'apprentissage sur l'estimation des paramètres du modèle.

En conclusion, le choix entre deux modèles dont la performance estimée de généralisation est sensiblement la même, mais dont le nombre de paramètres ajustables est différent doit être effectué suivant la nature du problème à traiter :

1. si l'objectif est de continuer à améliorer la confiance sur la prédiction du modèle (et donc sa performance), et surtout si l'on a la possibilité de compléter la base d'apprentissage en fonction des intervalles de confiance, le concepteur de modèle aura intérêt à choisir un modèle légèrement trop grand avec un paramètre $\mu$ qui n'est pas maximal. Ceci lui permettra de confirmer ou d'infirmer certaines non-linéarités décelées localement sur les exemples d'apprentissage,
2. si le but est d'utiliser le modèle tel quel, soit parce qu'il est jugé suffisamment performant, soit parce qu'on ne peut pas disposer d'exemples supplémentaires, il vaut mieux choisir, à performances estimées similaires, le modèle dont la valeur $\mu$ se rapproche le plus de 1 et possédant le plus petit nombre de paramètres ajustables. $E_p$ est alors une bonne approximation de l'erreur de généralisation théorique.

### *4.5.2 Amélioration progressive des modèles*

Ainsi que nous venons de le signaler, si l'on a la possibilité lors du test ou de l'utilisation du modèle d'avoir accès à la mesure de sorties dont la prédiction serait jugée trop incertaine, il est extrêmement utile de compléter la base d'apprentissage par ces exemples.

Pour cela, nous venons de voir qu'il est alors préférable de sélectionner - à l'aide de $\mu$ - des modèles légèrement surajustés. Ensuite, cela demande de définir un seuil à partir duquel on estime qu'une prédiction est trop incertaine.

Nous proposons d'utiliser le seuil suivant :

$$IC_{max} = t_\alpha^{N-q}\, s \qquad (4.5)$$

D'après la formule (3.12), ce seuil correspond à la demi-largeur de l'intervalle de confiance sur la prédiction d'un exemple d'apprentissage dont l'influence sur les poids du modèle serait maximale ($h_{ii} = 1$).

Ainsi, dans le cas du modèle de la figure 4.10, il serait souhaitable de compléter la base d'apprentissage par des exemples situés dans les zones des abscisses correspondant à des intervalles de confiance représentés - sur la figure 4.12 - en gris plus foncé.

Il s'agit en fait du seuil maximal que l'on peut raisonnablement utiliser. En effet, au-dessus de celui-ci, la confiance sur la prédiction serait moins bonne que l'incertitude attribuée à la mesure (figure 2.6).

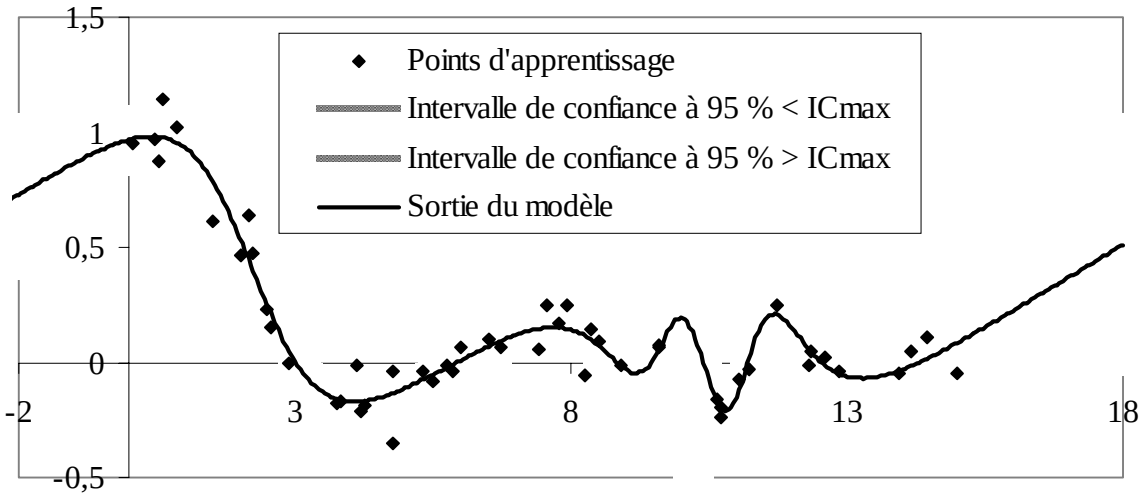


*Figure 4.12 : Application du seuil $IC_{max}$ au modèle de la figure 4.10*

Enfin, dans la pratique, rien n'interdit de réduire ce seuil, l'objectif étant d'atteindre une bonne homogénéité entre intervalles de confiance sur les bases d'apprentissage et de test.

## 4.6 Conclusion

Dans ce chapitre, nous avons montré, sur deux exemples, que, pour quantifier les performances de généralisation d'un modèle sur le principe du leave-one-out, il fallait sélectionner, à architecture donnée, les minima de la fonction de coût quadratique sur la base de l'erreur $E_p$. Nous avons montré clairement que les autres manières de procéder conduisaient souvent au choix de modèles surajustés.

Pour un problème donné, lorsque l'on considère des architectures de taille croissante, on observe une stabilisation de $E_p$ au niveau de l'écart-type du bruit. Nous proposons de sélectionner l'architecture optimale en combinant $E_p$ et la moyenne normalisée $\mu$ des intervalles de confiance sur les exemples d'apprentissage. Suivant qu'il cherche à améliorer progressivement le modèle ou bien qu'il désire l'utiliser tel quel, le concepteur de modèles aura en effet intérêt à choisir un modèle légèrement surajusté, de manière à cibler les zones de l'espace des entrées où rajouter des exemples, ou bien le plus petit modèle satisfaisant, pour lequel $\mu$ sera le plus proche de 1.

# 5 UN NOUVEL ALGORITHME D'APPRENTISSAGE


## Résumé

*Le score de validation croisée $E_p$ étudié dans le chapitre 4 peut être considéré comme la valeur prise - à la fin de l'apprentissage - par $\sqrt{J^* / N}$ où $J^*$ est obtenue à partir du coût quadratique en pondérant chaque résidu en fonction de son influence sur les poids du modèle. Nous étudions dans ce chapitre la minimisation itérative de $J^*$, qui n'est définie que sur un domaine de l'espace des coefficients, dont nous ne connaissons pas les propriétés.*

*Nous montrons, dans un premier temps, que, moyennant l'approximation selon laquelle les $\{h_{ii}\}_{i=1, \ldots, N}$ ne dépendent que faiblement des coefficients du modèle, le gradient de $J^*$ par rapport à ces derniers se calcule aisément. Ensuite, nous proposons une formule de modification itérative des coefficients directement inspirée de l'algorithme de Levenberg-Marquardt.*

*Puisque l'algorithme ne doit pas faire sortir les paramètres du domaine de définition de $J^*$, il faut considérer la minimisation de $J^*$ comme une poursuite de la minimisation de J, sous réserve que celle-ci ait convergé vers un minimum de rang plein.*

*Les résultats montrent surtout un gain très important au niveau de la dispersion des scores $E_p$ des minima atteints, ce qui permet d'augmenter la probabilité de trouver le vecteur des coefficients pour lequel $E_p$ est le plus faible.*

*Cette méthode s'avère être une façon de régulariser les modèles a posteriori, c'est-à-dire après minimisation du coût quadratique. Il s'agit donc d'un complément très intéressant aux travaux présentés dans le chapitre 4.*


## 5.1 Introduction

Dans le chapitre précédent, nous avons montré que, pour une architecture donnée, il était souhaitable de sélectionner le minimum $\boldsymbol{\theta}_{LS}$ de la fonction de coût quadratique $J(\boldsymbol{\theta}) = {}^t[\boldsymbol{y}_p - \boldsymbol{f}(X, \boldsymbol{\theta})]\,[\boldsymbol{y}_p - \boldsymbol{f}(X, \boldsymbol{\theta})]$ qui possède le plus petit score de validation croisée estimé $E_p = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{R_i}{1-h_{ii}}\right)^2}$, $R_i$ et $h_{ii}$ étant calculés à $\boldsymbol{\theta}_{LS}$. Ceci permet de limiter le surajustement, tout en utilisant tous les exemples disponibles.

Ceci revient à appliquer un algorithme d'apprentissage par rapport à une fonction $J$ alors que l'on cherche en réalité le minimum d'une autre fonction. Nous montrons dans ce chapitre que l'on peut directement chercher à minimiser la fonction :

$$J^*(\boldsymbol{\theta}) = {}^t[\boldsymbol{y}_p - \boldsymbol{f}(X, \boldsymbol{\theta})]\,[I - H^*(\boldsymbol{\theta})]^{-2}\,[\boldsymbol{y}_p - \boldsymbol{f}(X, \boldsymbol{\theta})], \qquad (5.1)$$

dans laquelle $H^*$ est la partie diagonale de la matrice de projection $H = Z\,({}^tZ\,Z)^{-1}\,{}^tZ$.

En notant $\boldsymbol{f}^*(X, \boldsymbol{\theta})$ le vecteur constitué par les $N$ fonctions obtenues en éliminant l'influence de chaque exemple d'apprentissage $\left\{ f^{(-i)}(\boldsymbol{x}^i, \boldsymbol{\theta}) = y_{pi} - \frac{y_{pi} - f(\boldsymbol{x}^i, \boldsymbol{\theta})}{1 - h_{ii}} \right\}_{i = 1, \ldots, N}$, $J^*$ peut également s'écrire [7] :

$$J^*(\boldsymbol{\theta}) = {}^t[\boldsymbol{y}_p - \boldsymbol{f}^*(X, \boldsymbol{\theta})]\,[\boldsymbol{y}_p - \boldsymbol{f}^*(X, \boldsymbol{\theta})]. \tag{5.2}$$

Une différence fondamentale entre ces deux fonctions de coût concerne leurs domaines de définition respectifs : contrairement à $J$, $J^*$ n'est pas définie sur tout $\Re^q$ mais uniquement dans le domaine (noté $\Omega^*$) où $Z$ est de rang plein (égal à $q$). Il faudra par conséquent veiller, si l'on envisage une minimisation itérative de $J^*$, à rester dans ce domaine.

L'autre caractéristique de $J^*$ est - compte tenu du fait que les $\{h_{ii}\}_{i=1, \ldots, N}$ sont tous compris entre 0 et 1 (cf. formule (2.2)) - d'être systématiquement supérieure à $J$.

Dans tout ce qui suit, afin de simplifier les notations, nous continuerons d'appeler $h_{ii}$ les éléments diagonaux de $H$, tout en précisant qu'ils sont définis pour tout $\boldsymbol{\theta}$ et non plus seulement en $\boldsymbol{\theta}_{LS}$.

La suite de ce chapitre est consacrée à l'étude de la minimisation de la fonction $J^*$ par adaptation de l'algorithme de Levenberg-Marquardt. Nous étudierons tout d'abord les calculs du coût et du gradient de cette fonction par rapport aux poids du modèle ainsi que la question de la modification itérative des coefficients. Ensuite, nous verrons comment mettre en œuvre cet algorithme de manière à en tirer profit.

## 5.2 Algorithme pour la minimisation de $J^*$

### *5.2.1 Calculs du coût et du gradient*

La particularité de la minimisation de $J^*(\boldsymbol{\theta})$ réside dans le fait que les $\{h_{ii}\}_{i=1, \ldots, N}$ sont eux-mêmes fonctions des poids du modèle (sauf dans le cas d'un modèle linéaire). Cependant, nous allons montrer qu'en négligeant cette dépendance, le gradient de $J^*$ par rapport aux coefficients se calcule simplement. En effet, notons :

$$J_i(\boldsymbol{\theta}) = \left(y_{pi} - f(\boldsymbol{x}^i, \boldsymbol{\theta})\right)^2 \tag{5.3}$$

et

$$J_i^*(\boldsymbol{\theta}) = \left(\frac{y_{pi} - f(\boldsymbol{x}^i, \boldsymbol{\theta})}{1 - h_{ii}(\boldsymbol{\theta})}\right)^2 \tag{5.4}$$

les contributions de l'exemple $i$ respectivement à $J$ et $J^*$. On a alors :

---

[7] Compte tenu de la définition (5.1), l'erreur $E_p$ utilisée dans le chapitre 4 représente la valeur prise par $\sqrt{\frac{J^*}{N}}$ à la fin de la minimisation du coût quadratique $J$.

$$\frac{\partial J_i}{\partial \boldsymbol{\theta}} = -2\left(y_{pi} - f(\boldsymbol{x}^i,\boldsymbol{\theta})\right)\frac{\partial f(\boldsymbol{x}^i,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \tag{5.5}$$

expression dans laquelle la dérivée de la fonction de régression par rapport au vecteur $\boldsymbol{\theta}$ s'obtient facilement, soit par calcul direct, soit - dans le cas d'un réseau de neurones - par rétropropagation.

En revanche, le calcul de $\frac{\partial J_i^*}{\partial \boldsymbol{\theta}}$ nécessiterait, en toute rigueur, celui du vecteur $\frac{\partial h_{ii}}{\partial \boldsymbol{\theta}}$. Dans tout ce qui précède, notamment dans le chapitre 3, nous avons toujours supposé qu'un développement au premier ordre dans l'espace des paramètres était valable ; en d'autres termes, nous avons supposé que les vecteurs qui définissent le sous-espace des solutions sont indépendants de $\boldsymbol{\theta}$. Or $h_{ii}$ est la $i^{\text{ème}}$ composante de la projection, sur le sous-espace des solutions, du vecteur unité le long de l'axe $i$ ; nous restons donc dans le même cadre d'approximation en supposant que les $\{h_{ii}\}_{i=1,\ldots,N}$ sont indépendants des paramètres :

$$\frac{\partial h_{ii}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cong 0 \tag{5.6}$$

L'approximation (5.6) permet ainsi d'obtenir :

$$\frac{\partial J_i^*}{\partial \boldsymbol{\theta}} \cong \frac{1}{(1-h_{ii})^2}\frac{\partial J_i}{\partial \boldsymbol{\theta}} \tag{5.7}$$

Nous disposons par conséquent d'une expression approchée du gradient de $J^*$ par rapport aux poids du modèle, expression qui se calcule très facilement à partir du gradient de $J$ et des valeurs $\{h_{ii}\}_{i=1,\ldots,N}$.

### *5.2.3 Modification des coefficients*

Après un rappel du principe et des fondements de l'algorithme de Levenberg-Marquardt, nous verrons comment adapter celui-ci à la minimisation de la fonction $J^*$.

#### 5.2.3.a Rappel : l'algorithme de Levenberg-Marquardt

Cet algorithme est une méthode itérative de minimisation de fonctions. Supposons que l'on cherche à minimiser une fonction $J$ par rapport à un vecteur de variables $\boldsymbol{\theta}$, que l'on se trouve, à l'étape $k$-1, au point $\boldsymbol{\theta}_{k-1}$ et que l'on cherche à se déplacer vers un point $\boldsymbol{\theta}_k$. Alors, si le déplacement $\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_k$ est suffisamment petit, on peut approcher le vecteur des résidus par un développement de Taylor au premier ordre :

$$\boldsymbol{y}_p - \boldsymbol{f}(X, \boldsymbol{\theta}_k) \cong \boldsymbol{y}_p - \boldsymbol{f}(X, \boldsymbol{\theta}_{k-1}) + Z_{k-1}(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1}) \tag{5.8}$$

L'idée de l'algorithme de Levenberg-Marquardt est de choisir $\boldsymbol{\theta}_k$ de manière à minimiser $J(\boldsymbol{\theta}_k)$ tout en limitant la distance entre $\boldsymbol{\theta}_{k-1}$ et $\boldsymbol{\theta}_k$. On écrit donc :

$$E(\boldsymbol{\theta}_k) = {}^t[\boldsymbol{y}_p - \boldsymbol{f}(X, \boldsymbol{\theta}_k)]\,[\boldsymbol{y}_p - \boldsymbol{f}(X, \boldsymbol{\theta}_k)] + \lambda\,{}^t[\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_k]\,[\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_k] \tag{5.9}$$

En utilisant (5.8), on obtient l'expression des poids tels que $\frac{\partial E(\boldsymbol{\theta}_k)}{\partial \boldsymbol{\theta}_k} = 0$ :

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \left( 2\, {}^t Z_{k-1}\, Z_{k-1} + \lambda_{k-1}\, I \right)^{-1} \left. \frac{\partial\, J}{\partial\, \boldsymbol{\theta}} \right|_{\boldsymbol{\theta} = \boldsymbol{\theta}_{k-1}} \tag{5.10}$$

Pour cela, le gradient total $\left. \frac{\partial\, J}{\partial\, \boldsymbol{\theta}} \right|_{\boldsymbol{\theta} = \boldsymbol{\theta}_{k-1}}$ est calculé en sommant les contributions des $N$ exemples d'apprentissages (formule (5.4)).

Cette méthode fait partie des algorithmes de second ordre. En effet, on reconnaît dans l'expression précédente une approximation classique du Hessien de la fonction de coût :

$$H = \left\{ \frac{\partial^2 J}{\partial\, \theta_i\, \partial\, \theta_j} \right\}_{i,\, j = 1,\, \ldots,\, N} \cong 2\, {}^t Z\, Z \tag{5.11}$$

Le paramètre $\lambda$, appelé pas de l'apprentissage, permet d'adapter l'algorithme à la forme de la fonction de coût et de réaliser un bon compromis entre la méthode de Newton ($\lambda$ nul), qui converge très rapidement au voisinage d'un minimum, et la méthode du gradient simple ($\lambda$ grand), efficace loin des minima.

Il existe plusieurs algorithmes d'asservissement automatique du pas ; nous avons mis en œuvre une technique simple et robuste :

- si la modification des paramètres provoque une diminution du coût, on accepte cette modification et on se rapproche de la direction de Newton en diminuant $\lambda$ (par exemple par un facteur 10),
- si la modification des paramètres provoque une augmentation du coût, on rejette cette modification et on se rapproche de la direction du gradient en augmentant $\lambda$ (par exemple par un facteur 10) ; on calcule une nouvelle modification des paramètres avec le nouveau pas.

La seule grandeur restant à fixer avant l'apprentissage est donc la valeur initiale du pas ([Bishop, 95] conseille $\lambda_0 = 0.1$).

#### 5.2.3.b Adaptation à la minimisation de $J^*$

Qualitativement, minimiser la fonction de coût $J^*$ consiste à calculer la contribution de chaque exemple à la modification du vecteur $\boldsymbol{\theta}$ en faisant comme si cet exemple n'appartenait pas à la base d'apprentissage.

Pour ce faire, il faut donc reconsidérer l'expression (5.8) et l'écrire sous la forme :

$$\begin{aligned} \boldsymbol{y}_p - \boldsymbol{f}^*(X,\, \boldsymbol{\theta}_k) &\cong \boldsymbol{y}_p - \boldsymbol{f}^*(X,\, \boldsymbol{\theta}_{k-1}) + \sum_{i=1}^{N} \left. \frac{\partial\, f^{(-i)}(\boldsymbol{x}^i, \boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \right|_{\boldsymbol{\theta} = \boldsymbol{\theta}_{k-1}} (\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1}) \\ &\cong \boldsymbol{y}_p - \boldsymbol{f}^*(X,\, \boldsymbol{\theta}_{k-1}) + Z^*_{k-1}\, (\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1}), \end{aligned} \tag{5.12}$$

Dans l'expression précédente, la matrice $Z^*_{k-1}$ représente, à l'étape *k*-1, la matrice (*N*, *q*) dont les lignes sont égales aux dérivées partielles par rapport à $\boldsymbol{\theta}$ des fonctions $\left\{ f^{(-i)}(\boldsymbol{x}^i, \boldsymbol{\theta}) \right\}_{i=1,\ldots,N}$.

Par analogie avec l'expression (5.10), on obtient donc :

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \left( 2\, {}^t Z^*_{k-1} Z^*_{k-1} + \lambda_{k-1} I \right)^{-1} \left. \frac{\partial J^*}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta} = \boldsymbol{\theta}_{k-1}} \tag{5.13}$$

Par ailleurs, par définition de $f^{(-i)}$ et d'après (5.6), on peut faire l'approximation :

$$Z^* = \left( I - H^* \right)^{-1} Z \tag{5.14}$$

La modification des paramètres à l'étape *k* s'écrit finalement :

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \left( 2\, {}^t Z_{k-1} (I - H^*_{k-1})^{-2} Z_{k-1} + \lambda_{k-1} I \right)^{-1} \left. \frac{\partial J^*}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta} = \boldsymbol{\theta}_{k-1}} \tag{5.15}$$

De même que précédemment, le gradient total $\left. \frac{\partial J^*}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta} = \boldsymbol{\theta}_{k-1}}$ s'obtient en sommant les *N* contributions partielles calculées par (5.7).

En ce qui concerne l'asservissement du pas $\lambda$, il faut également adapter la méthode présentée précédemment, en intégrant la contrainte concernant le domaine de définition de $J^*$, domaine duquel on ne doit pas sortir. Nous proposons de l'adapter de la manière suivante :

- si la modification des paramètres provoque une diminution du score de validation croisé $J^*$ **tout en conservant, numériquement, le rang de *Z***, on accepte cette modification et on se rapproche de la direction de Newton en diminuant $\lambda$,
- si la modification des paramètres provoque **soit une diminution du rang de *Z*, soit - si ce dernier est conservé -** une augmentation du score de validation croisée $J^*$, on rejette cette modification et on se rapproche de la direction du gradient en augmentant $\lambda$ ; on calcule une nouvelle modification des paramètres **avec le nouveau gradient total équivalent**.

### *5.2.4 En résumé*

Nous venons de voir qu'il est possible d'adapter l'algorithme de Levenberg-Marquardt de façon à minimiser la fonction de coût $J^*$. Les différences par rapport à la minimisation de *J* sont synthétisées dans le tableau 5.1.

Par conséquent, du point de vue du temps de calcul, la seule contrainte supplémentaire liée à la minimisation de $J^*$ est celle du calcul des $\{h_{ii}\}_{i=1,\ \ldots,\ N}$. Ce calcul nécessite une décomposition de *Z* en valeurs singulières à chaque essai du pas $\lambda$ (voir calcul des $\{h_{ii}\}_{i=1,\ldots,N}$ en Annexe 1). Rappelons néanmoins que la fonction $J^*$ n'est définie que dans un domaine $\Omega^*$.

| Contribution de l'exemple *i*… | $J$ | $J^*$ |
|---|---|---|
| …à la fonction de coût | $J_i(\boldsymbol{\theta}) = \left(y_{pi} - f(\boldsymbol{x}^i,\boldsymbol{\theta})\right)^2$ | $J_i^*(\boldsymbol{\theta}) = \left(\frac{y_{pi} - f(\boldsymbol{x}^i,\boldsymbol{\theta})}{1 - h_{ii}(\boldsymbol{\theta})}\right)^2$ |
| …au gradient | $\frac{\partial J_i}{\partial \boldsymbol{\theta}} = -2\left(y_{pi} - f(\boldsymbol{x}^i,\boldsymbol{\theta})\right)\boldsymbol{z}^i$ | $\frac{\partial J_i^*}{\partial \boldsymbol{\theta}} \cong -2\frac{y_{pi} - f(\boldsymbol{x}^i,\boldsymbol{\theta})}{\left(1 - h_{ii}(\boldsymbol{\theta})\right)^2}\boldsymbol{z}^i$ |
| …à la modification des poids | $\left(2\,{}^tZ\,Z + \lambda\, I\right)^{-1}\frac{\partial J_i}{\partial \boldsymbol{\theta}}$ | $\left(2\,{}^tZ\,(I - H^*)^{-2}\,Z + \lambda\, I\right)^{-1}\frac{\partial J_i^*}{\partial \boldsymbol{\theta}}$ |

*Tableau 5.1 : Différences entre la minimisation de J et de $J^*$*

## 5.3 Mise en œuvre de l'algorithme

Nous venons de montrer au paragraphe précédent qu'il était possible d'adapter l'algorithme de Levenberg-Marquardt à la minimisation de la fonction de coût $J^*$. Or, celle-ci n'étant définie que sur un domaine $\Omega^*$ de $\Re^q$, il convient, d'une part, d'initialiser les poids à l'intérieur de ce domaine, et, d'autre part, de s'assurer, à chaque modification des coefficients, que l'on reste bien à l'intérieur de $\Omega^*$.

Ce paragraphe débute par l'étude de ces contraintes et conduit à définir la manière dont nous avons mis en œuvre ce nouvel algorithme d'apprentissage. Ensuite, les résultats ainsi obtenus seront commentés.

### *5.3.1 Etude des contraintes*

Le domaine $\Omega^*$ est défini par les exemples d'apprentissage, et, bien entendu, par la famille de fonctions paramétrées considérée. Nous ne connaissons pas ses propriétés mathématiques, en particulier en termes de compacité. On peut néanmoins prévoir une difficulté pratique, résultant de l'initialisation des poids du réseau avant apprentissage : en effet, la procédure habituelle d'initialisation des poids d'un réseau consiste à choisir pour ceux-ci des valeurs aléatoires voisines de zéro, afin de ne pas saturer les fonctions de transfert sigmoïdales en début d'apprentissage. Or, il est facile de vérifier que l'origine de $\Re^q$ ne fait pas partie de $\Omega^*$. Il est donc très probable que ce type d'initialisation fasse démarrer l'apprentissage en-dehors du domaine recherché.

La figure 5.1 illustre le fait que, durant la minimisation de la fonction $J$ lors d'un apprentissage conventionnel, le point représentatif du réseau dans l'espace des paramètres parcourt successivement des zones où la matrice $Z$ est de rang plein, et des zones où elle n'est pas de rang plein, ce qui explique la discontinuité de $J^*$. Il s'agit d'un apprentissage réalisé sur les données issues de la fonction de régression $\frac{\sin(x)}{}$ avec une architecture à 2 neurones cachés avec terme direct, pourtant peu susceptible de présenter un surajustement. Le minimum atteint est de rang plein, mais l'apprentissage a traversé des zones avec déficience

de rang. Par ailleurs, on constate que bien que $J^*$ - lorsqu'il est défini - n'est soumis à aucune contrainte sinon à celle d'être supérieure à $J$.

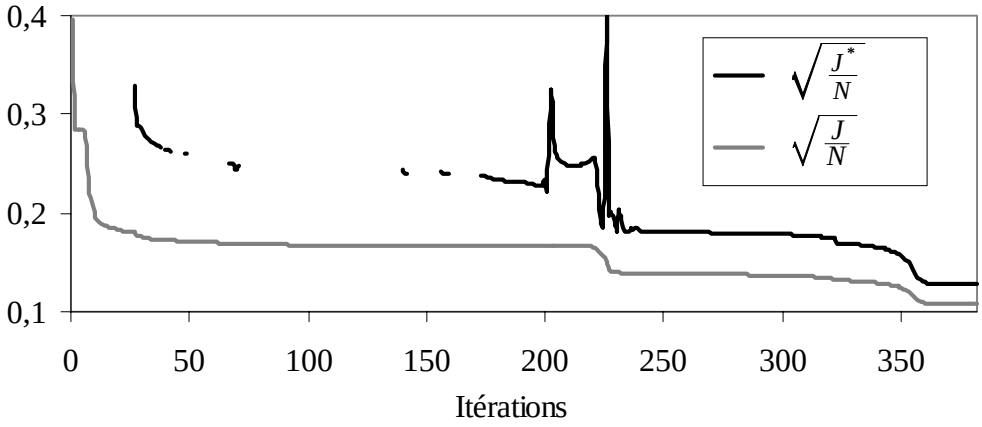


*Figure 5.1 : Exemple d'évolution, discontinue, de $J^*$ lors d'une minimisation itérative de J par l'algorithme de Levenberg-Marquardt*

De plus nous avons montré au paragraphe 3.3.3 que, pour un exemple à forte influence, le résidu $R_i$ tend en général vers 0 **moins rapidement** que $h_{ii}$ tend vers 1. En d'autres termes, il y a de fortes chances pour que la fonction de coût $J^*$ prenne de grandes valeurs, voire tende vers l'infini, à la frontière du domaine $\Omega^*$. Dans l'hypothèse où $\Omega^*$ serait formé d'un ou plusieurs domaines compacts, la surface représentée par $J^*$ serait constituée de "cuvettes" à bord fini ou infini, desquelles il serait donc difficile de sortir.

Dans ces conditions, il apparaît préférable d'utiliser la méthode de minimisation de $J^*$ comme une amélioration itérative de l'apprentissage classique, en laissant l'apprentissage sur $J$ se poursuivre jusqu'à son terme[8], et commencer celui sur $J^*$ à partir du vecteur final des coefficients, sous réserve que celui-ci se trouve dans $\Omega^*$.

### *5.3.2 Mise en œuvre de l'algorithme*

Afin de rendre compte des performances que l'on obtient par la méthode décrite précédemment, nous avons choisi de représenter les solutions atteintes (minima de $J$ ou de $J^*$) - pour diverses initialisations des poids - par le couple ($EQMA$, $E_p$) dont les coordonnées représentent les valeurs prises à la fin de l'apprentissage respectivement par $\sqrt{\frac{J}{N}}$ et $\sqrt{\frac{J^*}{N}}$.

La figure 5.2 présente dans le plan ($EQMA$, $E_p$) les résultats obtenus en poursuivant la minimisation du coût quadratique par la minimisation de $J^*$. Il s'agit ici d'une architecture à 4 neurones cachés plus terme direct, ajustée sur les données issues de la fonction de régression $\frac{\sin(x)}{x}$. Ces résultats sont comparés à ceux obtenus avec les mêmes initialisations aléatoires des poids (en l'occurrence 500) sans poursuite de l'apprentissage.

---

[8] C'est-à-dire jusqu'à satisfaction de différents critères d'arrêts portant sur la norme du gradient, le pas d'apprentissage, voire le nombre d'itérations.

Certains exemples d'évolution de la solution obtenue, entre la fin de l'apprentissage sur $J$ et la fin de celui sur $J^*$, sont présentés sur la figure 5.2 sous forme de flèches.

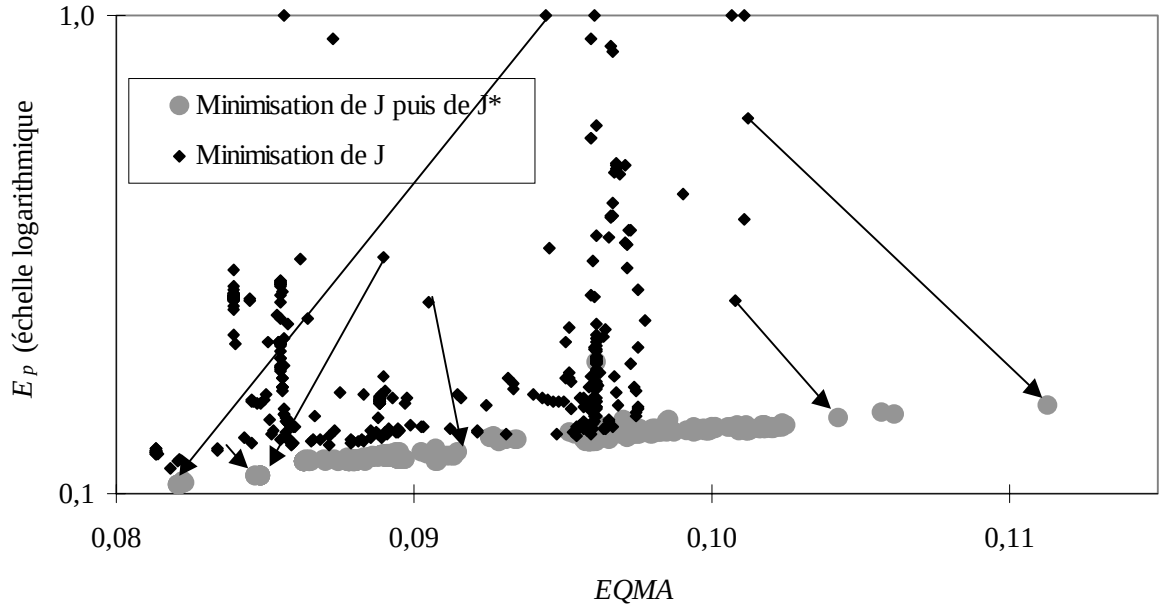


*Figure 5.2 : Exemple d'évolution des solutions atteintes grâce à la poursuite de l'apprentissage par la minimisation itérative de $J^*$*

L'analyse de cet exemple amène plusieurs commentaires :

1. comme nous l'avons montré dans le chapitre 4, beaucoup de minima de $J$ présentent - malgré une matrice $Z$ de rang plein - un surajustement certain. Ceci se traduit, dans le plan précédent, par une dispersion des performances des minima beaucoup plus importante sur $E_p$ que sur l'*EQMA*, ce qui nous a conduit à utiliser une échelle logarithmique pour l'axe des ordonnées[9],
2. cette dispersion a été considérablement réduite par le fait de poursuivre, par la minimisation de $J^*$, chaque apprentissage qui avait convergé vers un minimum de rang plein. A nombre d'initialisations aléatoires fixé, cette stratégie augmente donc la probabilité de détecter le minimum global de $J^*$,
3. la poursuite de l'apprentissage avec $J^*$ comme fonction de coût conduit à des réductions effective de $E_p$. Ceci est logique car il n'y aucune raison - surtout pour des "grandes" architectures - que les minima globaux de $J$ et de $J^*$ coïncident. Cependant, pour les minima présentant la plus petite valeur de l'erreur $E_p$, la réduction de cette dernière n'est pas forcément significative (sur cet exemple le meilleur minimum atteint est passé de $E_p = 0{,}113$ à $E_p = 0.104$),
4. la poursuite de l'apprentissage avec $J^*$ comme fonction de coût ne se traduit pas forcément par une augmentation de l'*EQMA*. La figure 5.2 présente deux exemples pour lesquels

[9] Pour des raisons de lisibilité du graphique, nous avons choisi de représenter les solutions telles que $E_p > 1{,}0$ comme si leur $E_p$ était égale à 1,0.

cette poursuite de l'apprentissage s'est traduite par la diminution conjointe de l'*EQMA* et de $E_p$.

Dans le cas du problème maître-élève, cette stratégie ne modifie pas les minima sélectionnés sur la base de $E_p$ jusqu'à 5 neurones cachés (figure 5.3). Néanmoins, même pour des architectures de taille supérieure, les erreurs $E_p$ des minima atteints ne sont que légèrement inférieures à celles atteintes par l'apprentissage classique (figure 4.8).

Dans ce cas, la principale amélioration apportée par la minimisation de $J^*$ consiste en une dispersion moindre des solutions atteintes dans le plan (*EQMA*, $E_p$), ce qui - à nombre d'initialisations égal - augmente la probabilité de trouver le minimum global de $J^*$.

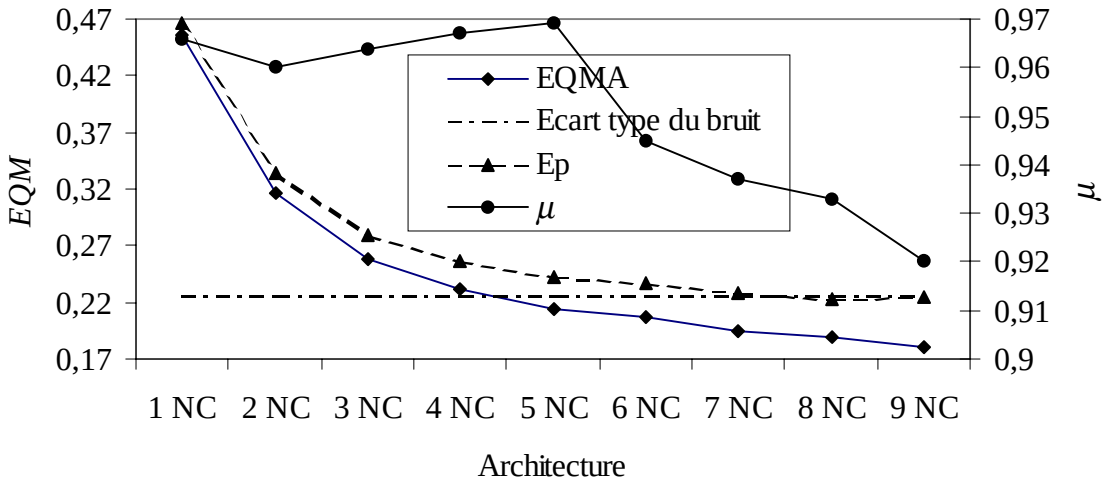


*Figure 5.3 : Performances des minima sélectionnés sur $E_p$ suite à la poursuite de l'apprentissage par la minimisation itérative de $J^*$ : cas du problème maître-élève*

Finalement, cette stratégie s'avère donc intéressante à plusieurs titres. Même si le gain - en termes de $E_p$ atteint - n'est pas toujours significatif, elle permet de diminuer largement la dispersion et la performance moyenne des minima atteints, quel que soit le nombre d'initialisations aléatoires utilisées. Lors du choix du ou des modèles définitifs dans le plan ($E_p$, $\mu$), ceci permet d'augmenter le nombre de modèles à la disposition du concepteur.

## 5.4 Conclusion

Nous avons montré qu'il est possible de considérer directement la fonction $J^*$ comme une fonction de coût, et d'adapter l'algorithme de Levenberg-Marquardt pour la minimiser. Compte tenu des caractéristiques de cette fonction, il est cependant conseillé de considérer l'apprentissage sur $J^*$ comme un prolongement de l'apprentissage sur $J$.

D'autres méthodes pourraient consister à "changer de fonction de coût" dès que la minimisation de $J$ est rentrée dans le domaine $\Omega^*$, ou à reprendre a posteriori l'apprentissage sur $J^*$ à partir des poids pour lesquels, lors de la minimisation de $J$, la valeur prise par $J^*$ était la plus faible. Outre le fait de nécessiter le calcul de $J^*$ et donc des $\{h_{ii}\}_{i=1, \ldots, N}$ lors de la première phase de l'apprentissage, ce qui ralentit significativement les calculs, ces autres

méthodes donnent - sur les deux exemples étudiés - de mauvais résultats (erreurs $E_p$ plus élevées et dispersion des minima, dans le plan ($EQMA$, $E_p$), plus grande qu'après minimisation de $J$).

Le fait de poursuivre l'apprentissage sur $J$ par la minimisation de $J^*$ peut s'interpréter comme une forme de régularisation a posteriori des solutions obtenues. En effet, on peut considérer $J^*$ comme la somme de $J$ et d'un terme de pénalisation. Cependant, ce type de pénalisation est différent de ceux couramment utilisés car, contrairement à - par exemple - la norme du vecteur des poids, le terme de pénalisation utilisé ici dépend explicitement des données utilisées. On ne pénalise donc pas a priori les solutions à forte variance, mais celles présentent un surajustement par rapport à une certaine base d'apprentissage.

Finalement, les travaux présentés dans ce mémoire montrent que, face au dilemme biais / variance, les approches a priori (par régularisation) et a posteriori (validation croisée) peuvent être utilisées dans un même contexte, à savoir le leave-one-out.

# 6 INTRODUCTION AU SOUDAGE PAR POINTS


## Résumé

*Le soudage par points sert à assembler localement deux tôles, en utilisant l'effet Joule. A cet effet, on comprime ces tôles à l'aide d'une paire d'électrodes, généralement en alliage de cuivre, et l'on fait passer par ces mêmes électrodes un courant électrique de forte intensité. La chaleur engendrée par ce courant à l'interface tôle-tôle fait fondre localement le métal, ce qui crée, après solidification, un point de soudure.*

*Une soudure est réalisée en une à deux secondes, avec un temps effectif de passage du courant de quelques dixièmes de secondes. Les phénomènes physiques entrant en jeu lors d'une soudure sont à la fois d'origine électrique, thermique, mécanique et métallurgique. La rapidité et la complexité de ces phénomènes en font un procédé extrêmement difficile à modéliser.*

*Conformément à la géométrie des électrodes, une soudure par point possède - dans le plan des deux tôles - une forme approximativement circulaire. Si la fonction principale d'une soudure est la tenue mécanique, on choisit généralement de caractériser sa qualité par son diamètre de bouton, c'est-à-dire par le diamètre moyen (en millimètres) du rivet restant sur l'une des deux tôles après un essai destructif appelé déboutonnage.*

*Parmi les différents paramètres de soudage, l'intensité du courant de soudage joue un rôle prépondérant, car elle conditionne directement la taille de la soudure. Le domaine de soudabilité d'un produit est défini comme la plage d'intensité permettant d'obtenir une soudure de qualité satisfaisante, tous les autres paramètres (effort mécanique, durées, etc.) étant fixés préalablement.*

*L'usage de tôles protégées contre la corrosion se généralise. Dans le cas où ce revêtement est zingué, on constate que le revêtement entraîne une dégradation des électrodes, au fur et à mesure de leur utilisation, ce qui se traduit par un décalage du domaine de soudabilité vers les intensités élevées. On a pris l'habitude de caractériser cette dégradation par le nombre maximal de points de qualité satisfaisante que l'on peut souder avec l'intensité de haut de domaine à électrodes neuves. Ce nombre est appelé la durée de vie des électrodes.*


## 6.1 Généralités

Le soudage par points fait partie de la famille des procédés de "soudage par résistance", au même titre que le soudage à la molette, par bossages, ou en bout. Il est utilisé pour assembler deux tôles (ou plus) dont l’épaisseur est typiquement comprise entre 0,5 et 10 mm. Signalons que ces deux tôles peuvent avoir - même si ce n'est généralement pas le cas au CRDM - des caractéristiques différentes (composition, revêtement), et qu'elles n’ont pas forcément la même épaisseur. Historiquement, ce fut l’Américain Thomson qui eut, en 1877, l’idée d’assembler deux tôles d’acier en utilisant, comme agent de chauffage, l’effet de la traversée de l’assemblage par un courant électrique de forte intensité.

Il s'agit depuis longtemps du procédé d'assemblage numéro un des carrosseries automobiles, une voiture nécessitant en moyenne 4000 points soudés.

### *6.1.1 Principe*

La figure 6.1 représente schématiquement le principe du soudage par points : les deux tôles sont prises en étau entre deux électrodes afin de maintenir l'ensemble en contact. Cet assemblage est ensuite traversé par un courant de forte intensité qui crée un noyau fondu au niveau de l'interface tôle-tôle. En refroidissant, ce noyau fondu fixe localement les deux tôles entre elles.

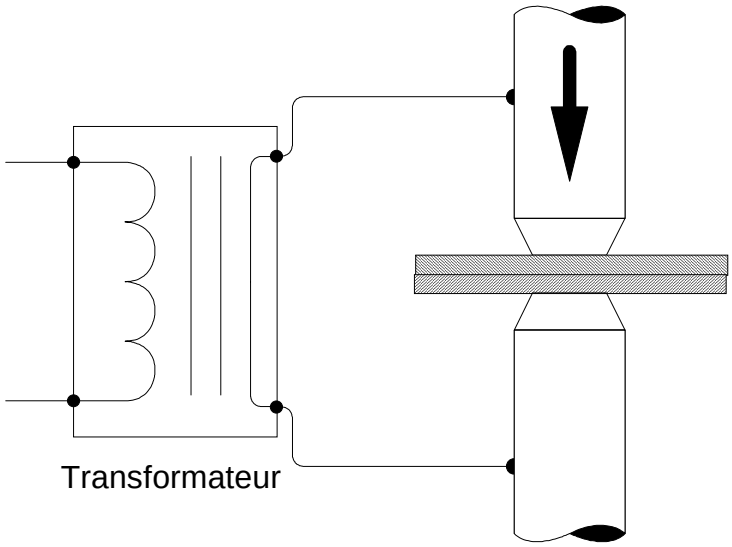


*Figure 6.1 : Principe du soudage par points*

### *6.1.2 Déroulement du cycle de soudage*

Un cycle de soudage se décompose en quatre phases (voir [Cazes 93]) :

- l'**accostage** : les électrodes se rapprochent et viennent comprimer les pièces à souder, à l'endroit prévu et sous un effort donné. Dans le cas des machines du CRDM, seule l'électrode supérieure se rapproche, l'autre étant fixe. Cette phase se termine quand la valeur d'effort nominale est atteinte,
- le **soudage** : le courant passe, déclenché par la fermeture du contacteur du circuit de puissance, et doit, par effet Joule, produire assez de chaleur à l'interface tôle-tôle pour qu'une zone fondue apparaisse,
- le **forgeage :** effectué avec maintien de l'effort mais sans passage de courant, il permet au noyau fondu de se refroidir et de se solidifier en restant confiné,
- la **remontée de l'électrode** : l'ensemble des deux tôles peut alors être translaté afin de procéder à la soudure d'un nouveau point.

Ces quatre phases, ainsi que les évolutions de l'effort mécanique et du courant de soudage tout au long d'un cycle, sont représentées sur la figure 6.2.

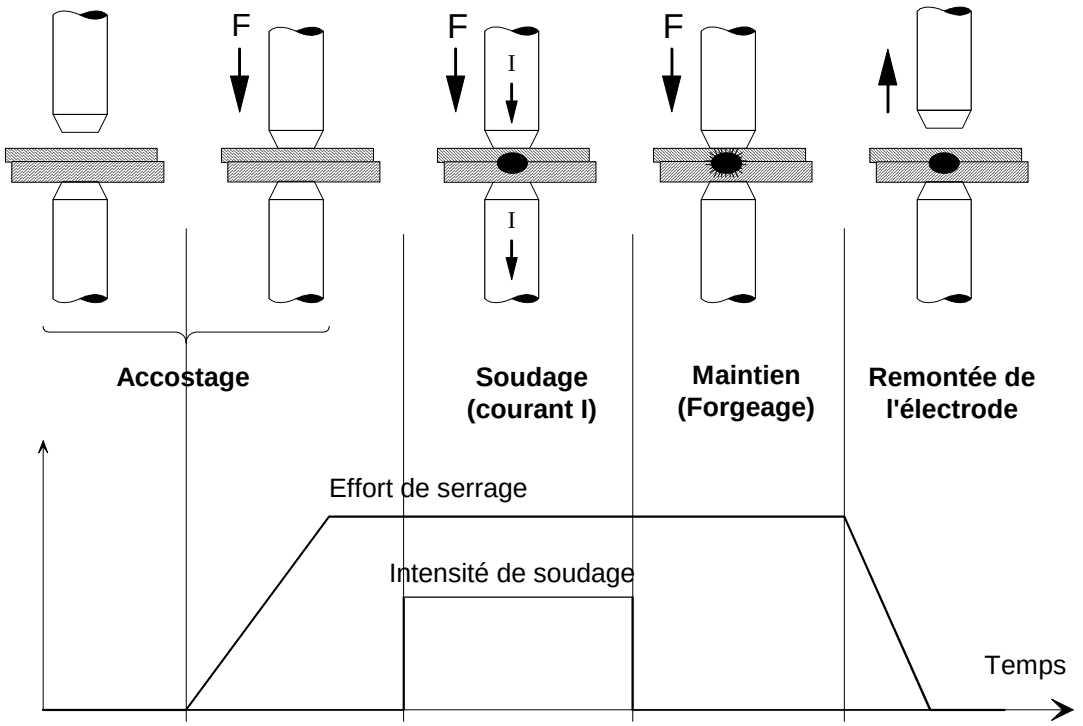


*Figure 6.2 : Les différentes phases d'un cycle de soudage*

L'allure des courbes d'effort et d'intensité représentées ici est purement qualitative. Dans la pratique, leurs évolutions ne sont jamais linéaires car les valeurs sont fortement perturbées par les évolutions des résistances mécaniques et électriques rencontrées.

### *6.1.3 Paramètres de soudage*

Les paramètres de soudage doivent être adaptés en fonction des caractéristiques des tôles à souder. Par exemple, on conçoit bien que des tôles plus épaisses nécessitent un apport d'énergie plus important, permettant de fondre plus de métal, donc de former un noyau plus gros.

Dans l'ordre chronologique, l'**effort de soudage** est la première variable entrant en jeu puisqu'elle intervient dès la phase d'accostage. La valeur à appliquer (de l'ordre de quelques centaines de daN pour des produits d'épaisseur inférieure à 1 mm) dépend essentiellement des caractéristiques mécaniques et de l'épaisseur des tôles à souder. Dans la pratique, la courbe de mise en effort n'est pas linéaire comme indiqué sur la figure 6.2, mais dépend des caractéristiques de la machine à souder.

Le **courant de soudage** est évidemment un paramètre décisif, car il intervient au carré dans l'énergie dissipée par effet Joule. L'intensité efficace à délivrer (typiquement entre 5 et 20 kA) dépend, là encore, des propriétés mécaniques, de l'épaisseur des tôles à souder ainsi que de la présence ou non de revêtement. On utilise généralement un courant alternatif monophasé à 50 Hz, dont la valeur moyenne efficace sur une soudure peut être régulée par rapport à la valeur de consigne, par un dispositif électronique adapté. Même si nous ne l'avons pas considéré dans ce travail, il est possible d'utiliser d'autres types de courants, notamment le courant continu, obtenu par redressement et filtrage, à partir d'un générateur 1000 Hz.

Le **temps de soudage** intervient au premier ordre dans l'énergie électrique dissipée. Ce paramètre, d'une valeur typique de quelques dixièmes de seconde, est donc, lui aussi, adapté aux propriétés des tôles à souder. Lorsque le temps de soudage désiré est particulièrement long, on le découpe en "pulsations" - ou "temps chauds" - séparés par des "temps froids".

Le **temps de forgeage** nécessaire à la solidification de la soudure est généralement du même ordre de grandeur que le temps de soudage.

La réalisation d'un point soudé nécessitant environ 1 à 2 secondes, le procédé est adapté aux cadences de production élevées de l'industrie automobile moderne.

### *6.1.4 Mécanisme de formation de la soudure*

La chaleur servant à faire fondre l'acier au niveau du contact tôle-tôle est créée par effet Joule durant le passage du courant dans les conducteurs. La quantité de chaleur dégagée pendant la durée *t* en fonction de l'intensité du courant et de la résistance électrique traversée est donnée par la relation :

$$Q = \int_{\tau_0}^{\tau_0 + t} R\, i^2\, d\tau \qquad (6.1)$$

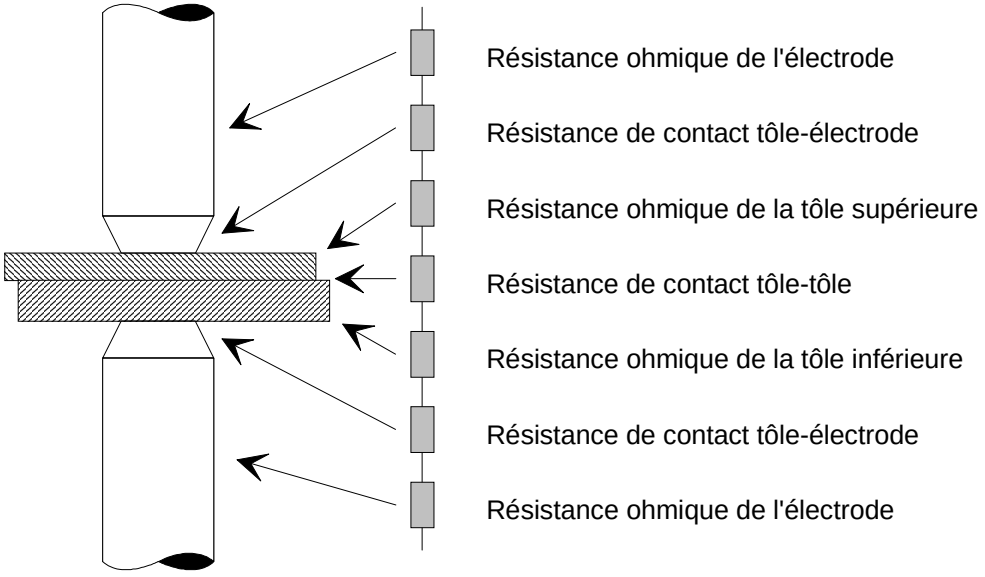


*Figure 6.3 : Circuit électrique équivalent d'un assemblage de deux tôles nues (d'après [Sauvage 94])*

Si l'on néglige les phénomènes capacitifs, l'équivalent électrique de l'assemblage de soudage par points est constitué de résistances en série (cf. figure 6.3). Le dégagement de chaleur est donc maximal à l'endroit où la résistance est la plus élevée.

Dans la pratique, les résistances de contact sont, au début d'un cycle de soudage, plus grandes que les résistances ohmiques des tôles et des électrodes. Dans le cas de produits revêtus, il est possible d'utiliser le même modèle que celui de la figure 6.3 (les résistances des revêtements étant alors incluses dans les résistances de contact), ou de compléter le modèle en ajoutant quatre résistances ohmiques du revêtement.

Ces diverses résistances, qui ont une influence directe sur les dégagements de chaleur, donc sur la constitution du noyau fondu, ne sont pas constantes au cours d'un cycle de soudage. Elles dépendent en effet fortement de la température :

- les résistances ohmiques augmentent en fonction de la température,
- les résistances de contact diminuent lorsque la température augmente. En effet, les contacts étant initialement "ponctuels" plutôt que surfaciques, ces résistances dépendent directement, à pression donnée, des caractéristiques mécaniques des matériaux en contact et de leur état de surface : la dureté des matériaux diminuant avec l'échauffement, les surfaces de contact augmentent. Il y a donc une diminution des résistances de contact avec la température.

En début de soudage, les dégagements de chaleur les plus importants sont situés au niveau des différentes discontinuités de l'assemblage :

- à l'interface tôle-tôle, cette chaleur sert à faire fondre le revêtement et l'acier afin de former le noyau fondu,
- aux interfaces électrode - tôle, cette chaleur - qui ne contribue pas à la formation de la soudure - est en partie évacuée par les électrodes, qui, outre une bonne conductivité électrique, doivent par conséquent avoir une conductivité thermique élevée.

La figure 6.4 représente les évolutions des différentes résistances au cours d'un cycle de soudage dans le cas de tôles non revêtues.

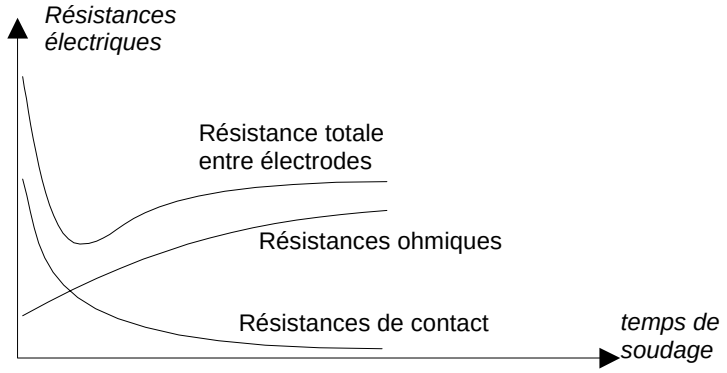


*Figure 6.4 : Évolution des résistances en cours de soudage sur tôles nues (voir [Sauvage 94])*

L'analyse des évolutions de ces résistances permet de mieux comprendre la cinétique de formation du point : l'échauffement est réalisé en premier lieu au niveau des interfaces, et l'électrode évacue les calories créées à l'interface tôle-électrode. Le noyau fondu s'initie à l'interface tôle-tôle et ne progresse que grâce aux résistances ohmiques des tôles.

Dans le cas de tôles revêtues, le graphe de la figure 6.4 se complique sensiblement, car il faut tenir compte de la résistance ohmique du revêtement ainsi que de sa température de fusion. Il est difficile de définir une tendance générale car certains revêtements ont une résistivité plus faible que celle de l'acier (cas des revêtements à base de zinc), tandis que d'autres ont une résistivité plus élevée (cas des revêtements organiques).

### *6.1.4 Géométrie d'un point soudé*

La géométrie d'un point soudé présente trois particularités (voir figure 6.5) :

- discontinuité de l'assemblage,
- présence d'une entaille concentrant les contraintes en cas de sollicitations mécaniques,
- indentation, par pénétration de l'électrode, des faces externes de l'assemblage.

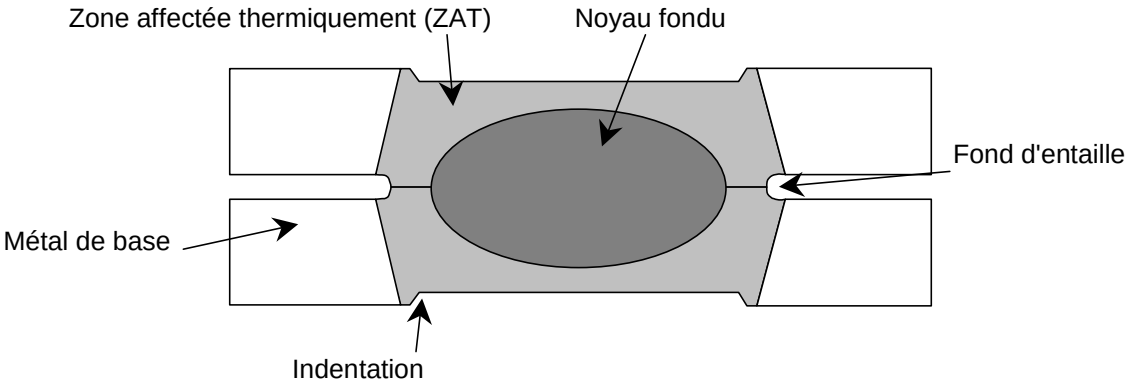


*Figure 6.5 : Coupe d'un point soudé*

Qualitativement, on constate que les caractéristiques mécaniques de la soudure sont principalement influencées par la taille du noyau fondu, et en particulier par son diamètre dans le plan des deux tôles.

Si la puissance électrique fournie est trop faible, le noyau fondu est trop petit, voire inexistant, et les caractéristiques mécaniques du point soudé risquent d'être insuffisantes. Plus on augmente la puissance fournie, plus la zone fondue est étendue et plus le point est résistant aux contraintes mécaniques. Cependant, passé un certain seuil, le noyau fondu atteint soit le fond d'entaille soit une des faces extérieures de la tôle : sous l'effet de la pression mécanique exercée par les électrodes, on assiste alors à une éjection de métal fondu (phénomène dit "d'expulsion") : la qualité du point soudé s'en trouve dégradée.

Ainsi, pour un type de tôle, une durée de soudage et un effort donnés, il existe une intensité minimale, en dessous de laquelle la tenue mécanique minimale définie par le cahier des charges de l'utilisateur du procédé n'est pas assurée, et une intensité maximale, au-dessus de laquelle il y a expulsion. Nous allons revenir dans le paragraphe suivant sur cette plage d'intensités acceptables, appelée "**domaine de soudabilité**" du produit.

## 6.2 Caractérisation d'une tôle d'acier revêtues

Par caractérisation d'une tôle, nous entendons ici l'étude de son aptitude au soudage par points dans certaines conditions. Au CRDM, ces études sont généralement faites selon la méthode préconisée par la norme [NF A 87-001]. Il s'agit de déterminer, pour une tôle donnée, son **domaine de soudabilité** et la **durée de vie des électrodes** de soudage.

Dans les deux cas, le critère de qualité d'un point soudé, en termes de résistance mécanique, est le **diamètre de bouton**. La définition du terme *bouton*, donnée dans [NF A 87-001] est le

"*rivet de déboutonnage restant sur l'une ou l'autre des tôles après essai destructif du point de soudure*" (voir figure 6.6).

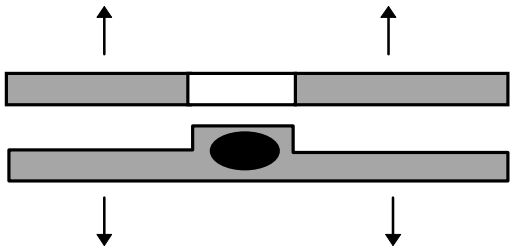

*Figure 6.6 : Déboutonnage d'un point soudé*

Ce test de déboutonnage peut être réalisé de plusieurs manières (traction pure, traction cisaillement, pelage, etc.).

Les diamètres minimal et maximal du bouton sont ensuite mesurés à l'aide d'un pied à coulisse. Le critère d'acceptabilité porte alors à la fois sur le diamètre minimal[10], et sur la moyenne des deux[11].

Cette approche suppose implicitement que le bouton a une forme elliptique et donc que ses diamètres minimal et maximal :

1. sont facilement détectables et mesurables,
2. caractérisent bien la taille de la soudure.

Nous reviendrons dans les chapitres suivants sur cette hypothèse.

Notons cependant que le test destructif ne met pas forcément en évidence un bouton mesurable : on peut observer une rupture en plan de joint ou un "déchirement" de ce bouton.

### *6.2.1 Le domaine de soudabilité*

La norme [NF A 87-001] définit le domaine de soudabilité comme la "*plage d'intensités efficaces du courant de soudage permettant d'obtenir un point de soudure de tenue mécanique satisfaisant aux critères définis auparavant*".

Comme nous l'avons déjà indiqué dans les paragraphes précédents, la taille du noyau fondu est fortement liée à la quantité d'énergie électrique fournie à la soudure, donc à l'intensité efficace utilisée. Si celle-ci est trop faible, le noyau fondu est trop petit voire inexistant, et le **déboutonnage** n'a pas lieu lors du test destructif : il y a **Rupture en Plan de Joint (RPJ)** de la soudure. Au-delà d'un certain seuil d'intensité, il y a déboutonnage, et le diamètre du bouton est caractéristique de la tenue mécanique du point soudé : plus on augmente l'énergie fournie, plus la zone fondue est large et plus le point est résistant aux contraintes mécaniques. Cependant, passé un certain seuil d'intensité, donc d'énergie électrique, le noyau fondu

[10] Qui doit être supérieur à 3 mm pour des tôles d'épaisseur inférieure à 1,25 mm, et supérieur à 5 mm pour des tôles plus épaisses.

[11] Qui doit être supérieure à 4 mm pour des tôles d'épaisseur inférieure à 1,25 mm et supérieure à 6 mm pour des tôles plus épaisses.

déborde la zone maintenue entre les électrodes : il y a expulsion, et la taille du noyau fondu s'en trouve diminuée.

En résumé, il s'agit d'étudier les variations du diamètre de bouton, donc de la tenue mécanique de la soudure, en fonction de l'intensité efficace. Ceci est schématisé sur la figure 6.7.

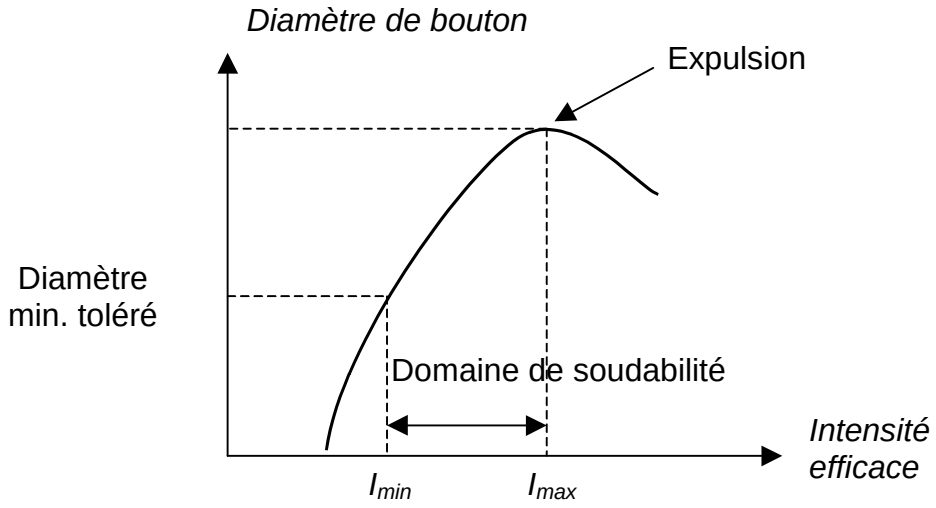


*Figure 6.7 : Définition du domaine de soudabilité*

Par domaine de soudabilité, on entend ainsi l'intervalle [$I_{min}$, $I_{max}$], où $I_{min}$ est la plus petite intensité donnant un diamètre acceptable, et $I_{max}$ est la plus grande intensité ne provoquant pas d'expulsion.

Pour mesurer le diamètre de bouton correspondant à une intensité donnée, on soude deux éprouvettes entre elles en les disposant comme indiqué sur la figure 6.8. Celles-ci sont maintenues dans cette position, avant et pendant le soudage, par un gabarit. La traction se fait en appliquant une force **F** croissante dans le temps jusqu'à rupture de la soudure. L'opérateur note alors, dans la mesure où un bouton apparaît sur l'une des deux tôles, ses diamètres minimal et maximal.

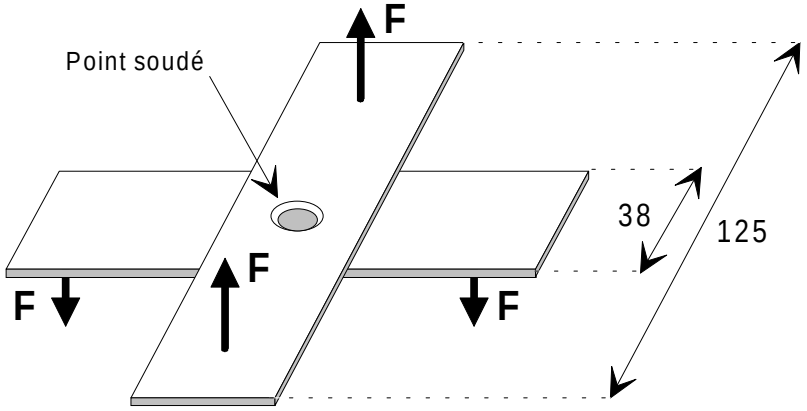


*Figure 6.8 : Éprouvettes de traction en croix*

Pour fixer les idées, les spécifications des clients de Sollac imposent en général un domaine de soudabilité d'au moins 1,5 kA. Dans le cas contraire, ils estiment que la mise en œuvre du produit est trop problématique.

### *6.2.2 La dégradation des électrodes*

Ce test vise à estimer le nombre de points soudés de bonne qualité (en termes de diamètre minimal acceptable) que l'on peut effectuer avec un jeu d'électrodes en gardant les mêmes réglages de paramètres, et en particulier l'intensité. En effet, on assiste à une dégradation des électrodes en fonction du nombre de points soudés, qui se traduit par une diminution de la qualité des soudures.

Cette dégradation des électrodes est due à une combinaison d'effets mécanique et chimique activés par les températures atteintes au niveau des interfaces électrode - tôle. L'usure des électrodes est particulièrement rapide sur produits revêtus zingués (avec - par ordre de vitesse de dégradation - les revêtements galvanisé-allié, électrozingué et galvanisé), pour lesquels on assiste à la création d'un alliage entre le cuivre des électrodes et le zinc du revêtement (formation de laiton). Cette dégradation entraîne un élargissement de la face active des électrodes et donc, à intensité de soudage égale, une diminution de la densité de courant. L'échauffement est donc de moins en moins localisé, ce qui a pour effet paradoxal de diminuer la taille du noyau fondu, de telle sorte que l'intensité nécessaire pour obtenir une taille de bouton donnée est de plus en plus élevée. Autrement dit, ***le domaine de soudabilité du produit se décale vers les hautes intensités***.

On définit alors la **durée de vie des électrodes**, pour certains réglages des paramètres, comme le "*nombre de points de qualité satisfaisante (selon le même critère que pour le domaine de soudabilité) que l'on peut souder avec un jeu d'électrodes en se plaçant à l'intensité haute du domaine de soudabilité correspondant aux électrodes neuves*".

La figure 6.9, qui illustre les notions de décalage du domaine de soudabilité et de durée de vie des électrodes, est tout à fait qualitative. Elle suppose en effet que l'état des électrodes peut être caractérisé par le nombre de points soudés ; or cela dépend également de l'intensité à laquelle ces points ont été soudés. Par ailleurs, l'évolution des limites basses et hautes n'est pas forcément linéaire - ni même monotone. En revanche, la tendance à l'élargissement du domaine de soudabilité est effectivement présente (voir par exemple [Gobez 94]), et résulte du fait qu'avec des électrodes usées, l'expulsion se produit pour des diamètres de boutons plus élevés.

Pour déterminer la durée de vie des électrodes, on utilise des électrodes neuves, et l'on soude des tôles appelées **bandes d'usure** en se plaçant à la limite supérieure du domaine de soudabilité. Suivant la durée de vie estimée a priori, on réalise tous les 200, 100 ou 50 points une **bande de contrôle** de 10 points destinée à subir des essais mécaniques.

En fonction des diamètres de boutons obtenus sur cette bande de contrôle et des mêmes critères de qualité que pour la détermination du domaine de soudabilité, l'opérateur décide soit de poursuivre l'essai en gardant la même intensité efficace, soit d'effectuer un **recalage** de cette intensité. Dans un contexte industriel, l'opération visant à compenser l'usure des

électrodes par une augmentation progressive de l'intensité de soudage est désignée sous le nom de **loi de déphasage**.

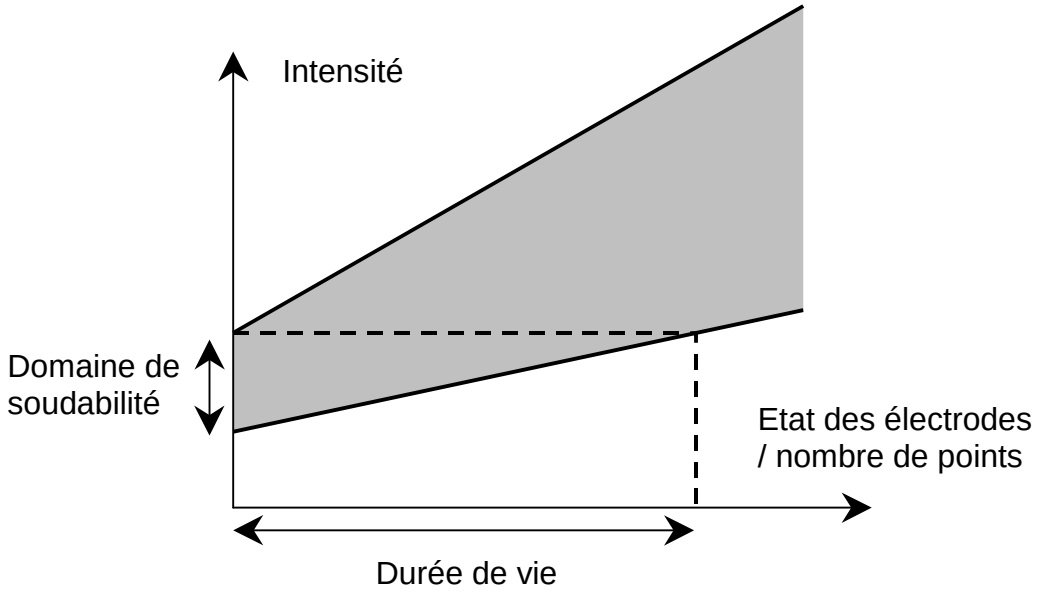


*Figure 6.9 : Décalage du domaine de soudabilité vers les intensités hautes avec la dégradation des électrodes*

Lors d'un recalage au sens de la norme [NF A 87.001], l'intensité utilisée correspond à la nouvelle limite supérieure du domaine de soudabilité. Il faut donc de nouveau déterminer l'intensité maximale sans expulsion avec les électrodes usées.

L'essai de durée de vie est arrêté - à l'initiative de l'opérateur - lorsqu'il juge, après un ou deux recalages, que le pourcentage d'augmentation du courant est trop élevé, voire que le nombre de points par incrémentation est trop faible, ou que la qualité du point (qu'il estime visuellement) est trop mauvaise.

Pour contrôler la qualité des points de la bande de contrôle, on utilise un dispositif permettant de déboutonner simultanément les 10 points soudés (voir [NF A 87-001]). Ce dispositif, communément appelé les "dents de la mer", est schématisé sur la figure 6.10.

Pour des raisons de symétrie évidentes, on ne considère pas les diamètres des deux points extrêmes comme significatifs. En effet, ils ne sont soumis que d'un seul côté à une force d'écartement. Il peut même arriver qu'ils ne soient pas totalement déboutonnés. Dans ce cas, les deux tôles sont séparées à l’aide d’un marteau et d’un burin.

Suivant le produit, la durée de vie des électrodes est, significativement différente. Ainsi, par exemple sur les produits GA et GZ 2 (Annexe 5) dont nous nous servirons au chapitre 8 pour illustrer les résultats de notre méthode, la durée de vie sans recalage est respectivement égale à 3500 et 400 points.

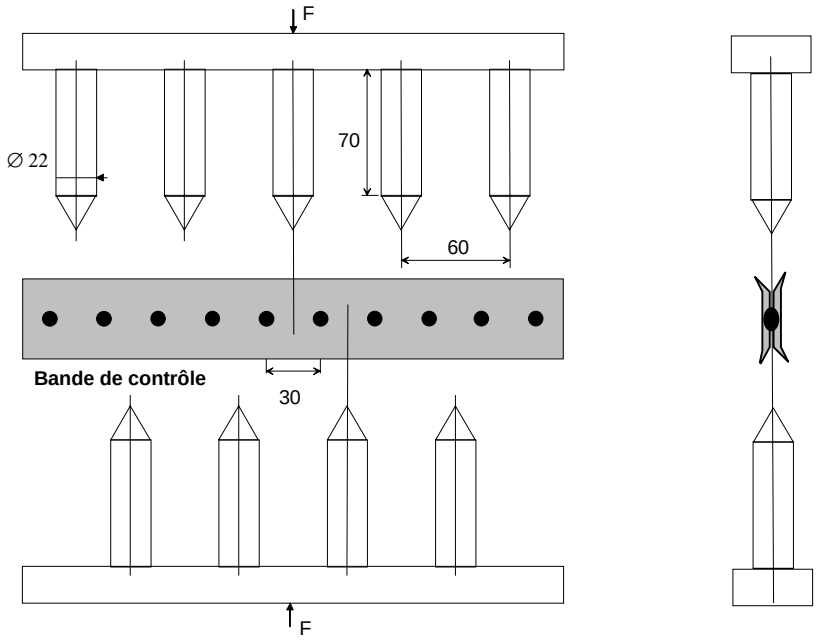


*Figure 6.10 : Dispositif de déboutonnage simultané de 10 points de soudure*

## 6.3 Conclusion

Les principes fondamentaux décrits dans ce chapitre d'introduction permettront au lecteur non spécialiste de se familiariser avec le procédé de soudage par points et de mieux appréhender les concepts qui seront présentés dans les deux chapitres suivants.

Précisons tout de même que cette présentation est fortement empreinte de la norme française [NF A 87.001], qui constitue la base de tous les essais standards effectués au CRDM. Il ne s'agit donc pas de la seule manière possible d'appréhender le procédé de soudage par points et la caractérisation de la soudabilité d'un produit.

Par exemple, [Waddell 97] souligne l'intérêt de suivre, en fonction de l'usure des électrodes, l'évolution des limites d'expulsion et de qualité minimale (4 mm). Ceci permet de caractériser plus finement la soudabilité d'un produit et d'estimer une pente de loi de déphasage linéaire : nous reviendrons sur ce point dans le chapitre 8.

Compte tenu du cadre que nous venons de fixer, nous allons étudier dans le chapitre suivant la signification et les enjeux d'une modélisation du soudage par points.

# CONCLUSION

Dans ce mémoire, nous avons montré l'intérêt que présentent les réseaux de neurones dans le cadre du contrôle non destructif et de la commande du procédé de soudage par points. Il en résulte une méthodologie, clairement définie, pour obtenir, tester et éventuellement améliorer un modèle de prévision du diamètre de bouton valable dans les conditions désirées.

Cette étude nous a amené à orienter nos recherches vers deux aspects théoriques concernant la mise en œuvre des réseaux de neurones, et plus généralement de tout modèle statistique :

1. *comment sélectionner le bon modèle parmi une série de modèles candidats ?* N'ayant pas d'idée a priori sur le nombre minimal de soudures à fournir pour obtenir de bons résultats, nous avons décidé d'axer nos recherches sur le leave-one-out, méthode de validation croisée réputée pour donner de bons résultats avec peu d'exemples,
2. *quel est le domaine de validité d'un modèle ?* Ceci nous a conduit à examiner la notion d'intervalle de confiance sur la sortie d'un modèle, linéaire ou non.

Finalement, il s'avère que ces deux aspects peuvent être abordés dans un même contexte, en utilisant le même outil, à savoir un développement de Taylor de la sortie du modèle au voisinage d'un point de l'espace des paramètres.

Nous avons montré que ce développement de Taylor permet d'estimer l'effet du retrait d'un exemple de la base d'apprentissage sur la solution des moindres carrés, la sortie du modèle, l'intervalle de confiance sur la prédiction de cet exemple, etc. Toutes ces estimations sont obtenues à partir de la grandeur $h_{ii}$, comprise entre 0 et 1, et qui représente l'influence de l'exemple $i$ sur l'estimation des paramètres du modèle.

Nous avons ensuite étudié l'estimation $E_p$ des performances de généralisation d'un modèle, fondée sur les erreurs de prédiction calculées précédemment. À architecture donnée, nous montrons que la sélection de modèles sur la base de $E_p$ est meilleure que celle résultant d'une mise en œuvre classique du leave-one-out, pour plusieurs raisons : elle évite la sélection de modèles surajustés, ne nécessite aucune hypothèse de stabilité algorithmique, et est bien plus rapide à effectuer. De plus, puisqu'il n'est plus nécessaire d'effectuer autant d'apprentissages qu'il y a d'exemples, cette méthode ne se limite plus aux problèmes où les exemples disponibles sont peu nombreux.

Pour comparer des modèles correspondant à différentes architectures, nous avons montré qu'il est judicieux de considérer la moyenne des intervalles de confiance sur la prédiction des exemples d'apprentissage, et nous avons défini un critère normalisé $\mu$. Nous sommes arrivés à la conclusion que le modèle définitif devait être choisi en fonction de l'objectif recherché : s'il s'agit de trouver le meilleur modèle, sachant que l'on ne dispose que de ces $N$ exemples d'apprentissage, il convient de choisir l'architecture présentant le critère $\mu$ le plus grand possible. En revanche, si l'on a l'intention et surtout la possibilité d'améliorer les performances du modèle, le choix d'un modèle légèrement surajusté, associé à l'utilisation des intervalles de confiance, permettra de compléter de manière ciblée la base d'apprentissage et d'augmenter localement la confiance du modèle.

Enfin, nous avons proposé une adaptation de l'algorithme de Levenberg-Marquardt, de manière à pouvoir minimiser directement la fonction de coût correspondant à l'erreur $E_p$. Nous avons montré qu'il est avantageux de rechercher un minimum de cette nouvelle fonction de coût, après la minimisation classique de la fonction de coût des moindres carrés.

Finalement, l'étude théorique du leave-one-out présentée dans ce mémoire nous a conduit à définir une méthode originale de sélection de modèles, qui considère directement l'origine du surajustement, à savoir la trop grande influence de certains exemples sur l'estimation des paramètres d'un modèle. L'utilisation conjointe de cette méthode et des intervalles de confiance permet à l'ingénieur de maîtriser à la fois la complexité et le domaine de validité de ses modèles, et ainsi d'en envisager l'amélioration progressive.

Une étude détaillée du procédé de soudage par points nous a permis de développer une application de cette approche au problème de la prédiction du diamètre de bouton. Nous avons tout d'abord analysé, d'un point de vue physique, les grandeurs susceptibles d'être pertinentes pour l'objectif de modélisation que nous nous étions fixé. Afin de sélectionner les quantités les plus pertinentes parmi les grandeurs candidates résultant de notre analyse, nous avons utilisé plusieurs méthodes de sélection qui ont donné des résultats concordants. Nous avons ensuite appliqué notre méthodologie de conception de modèles à deux aciers revêtus, et avons obtenu des résultats très proches de l'optimum : des prédictions du diamètre du bouton dont la précision est de l'ordre de l'incertitude sur la mesure de ce diamètre. Il convient de noter que, compte tenu de la nature de ce procédé, et en particulier de la différence entre paramètres de soudage et caractéristiques de la qualité du point, l'utilisation des intervalles de confiance a été déterminante dans le succès de l'application.

Les perspectives industrielles offertes par un modèle de contrôle non destructif de la qualité de soudures par points sont nombreuses. Il reste désormais à vérifier que cette approche pourra s'accommoder des contraintes en milieu de production.

Notre travail nous a donc permis de proposer une méthodologie originale, et d'usage très général, pour la conception de modèles non linéaires. Nous en avons montré l'efficacité lors de son application, dans le cas des réseaux de neurones, à un problème industriel réel dont l'enjeu est important.

## REFERENCES BIBLIOGRAPHIQUES

[Anders 99]
U. ANDERS & O. KORN
*"Model Selection in Neural Networks"*
Neural Networks 12, pp.309-323, 1999

[Antoniadis 92]
A. ANTONIADIS, J. BERRUYER & R. CARMONA
*"Régression non linéaire et applications"*
Paris : Economica, 1992

[Aro 94]
*"Micro 2x16 II Type "Bas-Plus" Logiciel 7B-8B"*
Notice d'utilisation, ARO S.A., 1994

[Barhen 93]
J. BAHREN, B. CETIN & J. BURDICK
*"Overcoming Local Minima in Neural Learning"*
Proceedings de la 6ème Conférence Internationale Neuro-Nîmes, 1993

[Bartlett 97]
P.L. BARLETT
*"For Valid Generalization, the Size of the Weights is more Important than the Size of the Network"*
M.C. Mozer, M.I. Jordan & T. Petsche (eds), Advances in Neural Information Processing Systems 9, Cambridge, MA : the MIT Press, pp. 134-140, 1997

[Bates 88]
D.M. BATES & D.G. WATTS
*"Nonlinear Regression Analysis and its Applications"*
Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, 1988

[Bishop 93]
C.M. BISHOP
*"Curvature-Driven Smoothing : a Learning Algorithm for Feedforward Networks"*
IEEE Transactions on Neural Networks4 (5), pp. 882-884, 1993

[Bishop 97]
C.M. BISHOP
*"Neural Networks for Pattern Recognition"*
Third Edition, Clarendon Press, Oxford, 1997

[Bottou 97]
*"La mise en œuvre des idées de V.N. Vapnik"*
Chapitre 16 de STATISTIQUE ET METHODES NEURONALES, S. Thiria, Y. Lechevallier, O. Gascuel & S. Canu éditeurs, DUNOD, 1997

[Breiman 96]
L. BREIMAN
*"Heuristics of Instability and Stabilization in Model Selection"*
Annal of Statistics 24, pp. 2350-2383, 1996

[Brown 97]
J.D. BROWN, M.G. RODD & N.T. WILLIAMS
"*Application of Artificial Intelligence to Resistance Spot Welding*"
International Institut of Welding, Doc. No. III - 1092, 1997

[Brugge 95]
M.H. BRUGGE, W.J. JANSEN, J.A.G NIJHUIS & L. SPAANENBURG
"*Non-destructive Test of Spot Welds Using a Neural Network*"
Proceedings ICAN '95, Paris, 1995

[Cazes 93]
R. CAZES
*"Le soudage par résistance"*
Les techniques de l'ingénieur, novembre 1993

[Chen 89]
S. CHEN, S.A. BILLINGS & W. LUO
*"Orthogonal Least Squares Methods & their Application to Nonlinear System Identification"*
International Journal of Control, Vol. 50, n° 5, pp. 1873-1896, 1989

[Cybenko 89]
G. CYBENKO
*"Approximation by Superpositions of a Sigmoidal Function"*
Mathematics of Control, Signals and Systems, Vol. 2, pp. 303-314, 1989

[De Vaux 98]
R.D. DE VAUX, J. SCHUMI, J. SCHWEINSBERG & L.H. UNGAR
*"Predictions Intervals for Neural Networks via Nonlinear Regression"*
Technometrics 40, pp. 273-282, 1998

[Devroye 79]
L.P. DEVROYE & T.J. WAGNER
*"Distribution-free Inequalities for the Deleted and Holdout Error Estimates"*
IEEE Transactions on Information Theory, IT-25 (2), pp. 601-604, 1979

[Dilthey 97]
U. DILTHEY & J. DICKERSBACH
"*Quality Assurance of Resistance Spot Welding by Employment of Neural Networks*"
International Institut of Welding, Doc. No. III - 1093, 1997

[Dreyfus 97]
G. DREYFUS, L. PERSONNAZ & G. TOULOUSE
*"Perceptrons, Old and New"*
Enciclopedia Italiana, in press.

[Dupuy 98]
T. DUPUY
*"La dégradation des électrodes lors du soudage par points de tôles d'acier zinguées"*
Thèse de Doctorat de Ecole des Mines de Paris, 1998

[Efron 93]
B. EFRON & R.J. TIBSHIRANI
*"An Introduction to the Boostrap"*
New-York : Chapman & Hall, 1993

[Euvrard 93]
G. EUVRARD
*"Estimation d'une régression non linéaire par réseaux de neurones ; application à un problème de robotique mobile"*
Thèse de Doctorat de l'Université Paris VI, 1993

[Funahashi 89]
K. FUNAHASHI
*"On the Approximate Realization of Continuous Mappings by Neural Networks"*
Neural Networks 4, pp. 349-360, 1991

[Gallinari 97]
P. GALLINARI
*"Heuristiques pour la généralisation"*
Chapitre 14 de STATISTIQUE ET METHODES NEURONALES, S. Thiria, Y. Lechevallier, O. Gascuel & S. Canu éditeurs, DUNOD, 1997

[Geman 92]
S. GEMAN, E. BIENENSTOCK & R. DOURSAT
*"Neural Networks and the Bias / Variance Dilemma"*
Neural Computation 4, pp. 1-58, 1992

[Gobez 94]
P. GOBEZ
*"Soudage des tôles revêtues"*
Les Techniques de l'ingénieur, B 7771, 1994

[Hansen 96]
L.K. HANSEN & J. LARSEN
*"Linear Unlearning for Cross-Validation"*
Advances in Computational Mathematics 5, pp. 269-280, 1996

[Hornik 94]
K. HORNIK, M. STINCHCOMBE, H. WHITE & P. AUER
*"Degree of Approximation Results for Feedforward Networks Approximating Unknown Mappings and their Derivatives"*
Neural Computation, Vol. 6, c°6, pp. 1262-1275, 1994

[Ivezic 99]
N. IVEZIC, J.D. ALLEN & T. ZACHARIA
*"Neural Network-Based Resistance Spot Welding Control and Quality Prediction"*
Proceedings of the Second International Conference on Intelligent Manufacturing and Processing of Materials (IPMM '99), Honolulu, 1999

[Jou 94]
M. JOU
"*An Intelligent Control System for Resistance Spot Welding Using Fuzzy Logic and Neural Networks*"
Thèse, Rensselaer Polytechnic Institute, Troy, 1994

[Jou 95]
M. JOU, R.W. MESSLER & C.J. LI
"*An Intelligent Control System for Resistance Spot Welding Using a Neural Network and Fuzzy Logic*"
IEEE Transaction, 1995

[Kearns 97]
M. KEARNS, D. RON
*"Algorithmic Stability and Sanity-Check Bounds for Leave-One-Out Cross Validation"*
Submitted to the Tenth Annual Conference on Computational Learning Theory

[Levenberg 44]
K. LEVENBERG
*"A Method for the Solution of Certain Non-linear Problems in Least Squares"*
Quartely Journal of Applied Mathematics II (2), pp. 164-168, 1944

[MacKay 92]
D.J.C. MACKAY
*"A Practical Bayesian Framework for Backpropagation Networks"*
Neural Computation 4 (4), pp. 448-472, 1992

[Marquardt 63]
D.W. MARQUARDT
*"An Algorithm for Least-squares Estimation of Non-linear Parameters"*
Journal of the Society of Industrial and Applied Mathematics 11 (2), pp. 431-441, 1963

[Matsuyama 97]
K.I. MATSUYAMA
"*Nugget Size Sensing of Spot Weld Based on Neural Network Learning*"
International Institut of Welding, Doc. No. III - 1081, 1997

[McQuarrie 98]
A.D.R MCQUARRIE & C.L. TSAI
*"Regression and Time Series Model Selection"*
Singapore : World Scientific, 1998

[Messler 94]
R.W. MESSLER, M. JOU & C.J. LI
*"A Fuzzy Logic Control System for Resistance Spot Welding Based on a Neural Network Model"*
Proceedings of Sheet Metal Welding Conference N° 6, AWS, 1994

[Moody 94]
J. MOODY
*"Prediction Risk and Neural Network Architecture Selection"*
From Statistics to Neural Networks : Theory and Pattern Recognition Applications, V. Cherkassky, J.H. Friedman, and H. Wechsler (eds), Springer-Verlag, 1994

[Nash 90]
J.C. NASH
*"Compact Numerical Methods for Computers : Linear Algebra and Function Minimisation"*
Ed. Adam Hilger, 1990

[Nelson 91]
M.C. NELSON & W.T. ILLINGWORTH
*"A Practical Guide to Neural Nets"*
Reading, MA : Addison Wesley, 1991

[NF A 87-001]
*"Caractérisation de la soudabilité par résistance par points de produits plats revêtus ou non"*
Comité de Normalisation de la Soudure - AFNOR, décembre 1994

[Osman 95]
K.A. OSMAN, A.M. HIGGINSON, H.R. KELLY, C.J. NEWTON & D.R. BOOMER
"*Monitoring of Resistance Spot-Welding Using Multi-Layer Perceptrons*"
Proceedings Autotech '95, Birmingham, 1995

[Osman 96]
K.A. OSMAN, A.M. HIGGINSON, H.R. KELLY, C.J. NEWTON & P. SHEASBY
"*Prediction of Aluminium Spot Weld Quality Using Artificial Neural Networks*"
Proceedings AWS '96, 1996

[Oukhellou 98]
L. OUKHELLOU, P. AKNIN, H. STOPPIGLIA & G. DREYFUS
*"A new Decision Criterion for Feature Selection : Application to the Classification of Non Destructive Testing Signature"*
European Signal Processing Conference (EUSIPCO'98), Rhodes, 1998

[Oussar 98]
Y. OUSSAR
*"Réseaux d'ondelettes et réseaux de neurones pour la modélisation statique et dynamique de processus"*
Thèse de Doctorat de l'Université Paris VI, 1998

[Plutowski 94]
M. PLUTOWSKI, S. SAKATA & H. WHITE
*"Cross-Validation Estimates IMSE"*
Advances in Neural Information Processing Systems 6, San Mateo, CA: Morgan Kaufmann Publishers, 1994

[Powell 76]
M.J.D. POWELL
*"Somme Global Convergence Properties of a Variable Metric Algorithm for Minimisation without Line Searches"*
Nonlinear Programming, London, 1986 SIAM-AMS Proceedings 9, R.W. Cottel & C.E. Lemke, Eds. Providence RI, 1976

[Press 92]
W.H. PRESS, S.A. TEUKOLSKY, W.T. VETTERLING & B.P. FLANNERY
*"Numerical Recipies in C : The Art of Scientific Computing"*
Second Edition, Cambridge University Press, 1992

[Rivals 98]
I. RIVALS, L. PERSONNAZ
*"Construction of Confidence Intervals in Neural Modeling Using a Linear Taylor Expansion"*
International Workshop on Advanced Black-Box Techniques for Nonlinear Modeling : Theory and Applications, Leuven-Belgium, 1998
*"Construction of Confidence Intervals for Neural Networks Based on Least Squares Estimation"*
Neural Networks, à paraître

[Rumelhart 86]
D.E. RUMELHART, G.E. HINTON & R.J. WILLIAMS
*"Learning Internal Representations by Error Propagation"*
Parallel Distributed Processing, MIT Press, Cambridge MA, pp. 318-362, 1986

[Saarinen 93]
S. SAARINEN, R. BRAMLEY & G. CYBENKO
*"Ill-Conditioning in Neural Network Training Problems"*
SIAM J. Sci. Stat. Comp. 14, pp. 693-714, 1993

[Saporta 90]
G. SAPORTA
*"Probabilités, analyse des données et statistique"*
Editions Technip, Paris, 1990

[Satoh 97]
T. SATOH, H. ABE & S. SUZUYAMA
"*A Trial of Quality Assurance in Resistance Spot Welding by Aid of Neural Network and Fuzzy Reasoning - Accomplished with Detection of Top Diameter of Electrode*"
International Institut of Welding, Doc. No. III - 1083, 1997

[Sauvage 94]
F. SAUVAGE, G. KAPLAN
*"Le soudage"*
Chapitre 38 du Livre de l'Acier, Technique & Documentation - Lavoisier, 1994

[Seber 89]
G.A.F. SEBER & C.J. WILD
*"Nonlinear regression"*
Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons (eds), New York, 1989.

[Sjöberg 92]
J. SJÖBERG & L. LJUNG
*"Overtraining, Regularization and Searching for Minimum in Neural Networks"*
Technical report LiTH-ISY-I-1297, Department of Electrical Engineering, Linkoping University, S-591 93 Linkoping, http ://www.control.isy.liu.se

[Sorensen 96]
P.H. SORENSEN, M. NORGARD, L.K. HANSEN & J. LARSEN
*"Cross-Validation with LULOO"*
Proceedings of the International Conference on Neural Information Processing - ICONIP '96, 1996

[Stone 74]
M. STONE
*"Cross-Validatory Choice and Assessment of Statistical Predictions"*
Journal of the Royal Statistical Society B 36, pp. 111-147, 1974

[Stoppiglia 97]
H. STOPPIGLIA
*"Méthodes statistiques de sélection de modèles neuronaux ; applications financières et bancaires"*
Thèse de Doctorat de l'Université Paris VI, 1997

[Thiria 97]
S. THIRIA, Y. LECHEVALLIER, O. GASCUEL & S. CANU
*"Statistique et méthodes neuronales"*
2ème cycle école d'ingénieur, S. Thiria, Y. Lechevallier, O. Gascuel & S. Canu éditeurs, DUNOD, 1997

[Tibshirani 96]
R.J. TIBSHIRANI
*"A Comparaison of Some Error Estimates for Neural Models"*
Neural Computation 8, pp. 152-163, 1996

[Thièblemont 92]
E. THIEBLEMONT
*"Modélisation du soudage par résistance par points"*
Thèse de doctorat, Institut National Polytechnique de Lorraine, 1992

[Vapnik 82]
V.N. VAPNIK
*"Estimation of Dependances Based on Empirical Data"*
Springer Verlag, New-York, 1982

[Waddel 97]
W. WADDEL & N. WILLIAMS
*"Control of Resistance Spot Welded Quality Using Multi-parameter Derived Algorithms for Zinc-Coated Sheets"*
Rapport technique EUR 17859 EN du contrat N° 7210-MB/805, Steel Research, 1997

[Waschkies 97]
E. WASCHKIES
*"Prüfen des Widerstandspunktschweißprozesses mit Ultraschall"*
Schweissen und Schneiden 1/97, pp. 17-19, 1997

[Wesgate 86]
S.A. WESTGATE
*"An Evaluation of Ultrasonic and Electrical Resistance Methods for the Non-destructive Testing (NDT) of Resistance Spot Welds in Low Carbon Steel Sheet"*
The Welding Institute, 1986

[Wolfe 69]
P. WOLFE
*"Convergence Conditions for Ascent Methods"*
S.I.A.M. Review 11, pp. 226-235, 1969

[Zhou 98]
G. ZHOU & J. SI
*"A Systematic and Effective Supervised Learning Mechanism Based on Jacobian Rank Deficiency"*
Neural Computation 10, pp.1031-1045, 1998

## ANNEXE 1 : CALCUL DES $h_{ii}$

Soit $Z$ une matrice de dimensions ($N$, $q$) (avec $N \geq q$), matrice jacobienne d'une fonction $\boldsymbol{f}(X, \boldsymbol{\theta})$ réalisée par exemple par un réseau de neurones à $q$ coefficients, ou par une fonction linéaire à $q$ - 1 régresseurs (dans ce cas, la matrice $Z$ est égale à la matrice des observations).

Définissons $Z = {}^t[\boldsymbol{z}^1, ..., \boldsymbol{z}^N]$, avec $\boldsymbol{z}^i = \left.\dfrac{\partial f(\boldsymbol{x}^i, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta} = \boldsymbol{\theta}^*}$. On cherche à calculer les termes diagonaux de la matrice de projection $H = Z ({}^tZ\, Z)^{-1}\, {}^tZ$ , définis comme :

$$h_{ii} = {}^t\boldsymbol{z}^i\, ({}^tZ\, Z)^{-1}\, \boldsymbol{z}^i \qquad \text{(A1.1)}$$

En tant qu'éléments diagonaux d'une matrice de projection orthogonale, les termes $\{h_{ii}\}_{i\, =\, 1,\, ...,\, N}$ ne sont définis que dans le cas où $Z$ est de rang plein, c'est-à-dire si ${}^tZ\, Z$ est inversible. Dans ce cas, ils vérifient les propriétés suivantes :

$$\forall i \in [1, ..., N] \quad 0 \leq h_{ii} \leq 1 \qquad \text{(A1.2)}$$

$$\text{trace}\ (H) = \sum^{N} h_{ii} = \text{rang}\ (Z) \qquad \text{(A1.3)}$$

On obtient $Z$ soit par calcul direct, à partir de la forme analytique de la fonction $\boldsymbol{f}$, soit - dans le cas de réseaux de neurones non bouclés - en rétropropageant une erreur égale à -1.

Une première méthode de calcul des $h_{ii}$ consiste à calculer la matrice ${}^tZ\, Z$, à l'inverser par une méthode classique (Cholesky, décomposition LU, ...), puis à la multiplier à droite et à gauche par les vecteurs $\boldsymbol{z}^i$ et ${}^t\boldsymbol{z}^i$. Cette méthode ne donne cependant de bons résultats que si la matrice ${}^tZ\, Z$ est suffisamment bien conditionnée pour que son inversion se déroule sans problème. Dans le cas contraire, ce calcul donne des valeurs supérieures à 1, voire négatives, c'est-à-dire ne satisfaisant pas à la relation (A1.2).

La solution que nous proposons consiste à décomposer la matrice Z sous la forme :

$$Z = U\, W\, {}^tV \qquad \text{(A1.4)}$$

avec :

- $U$ matrice ($N$, $q$) telle que ${}^tU\, U = I$,
- $W$ matrice ($q$, $q$) diagonale, dont les termes diagonaux, appelés valeurs singulières de $Z$, sont positifs ou nuls et classés par ordre décroissant,
- $V$ matrice ($q$, $q$) telle que ${}^tV\, V = V\, {}^tV = I$.

Cette décomposition, connue sous le nom de décomposition en valeurs singulières ou décomposition SVD (Singular Value Decomposition), est précise et très robuste, même si la matrice $Z$ est mal conditionnée ou de rang inférieur à $q$ (cf. [Press 92]).

On obtient donc :

$${}^{t}Z\,Z = V\,W\,{}^{t}U\,U\,W\,{}^{t}V = V\,W^{2}\,{}^{t}V \tag{A1.5}$$

puis

$$({}^{t}Z\,Z)^{-1} = V\,W^{-2}\,{}^{t}V \tag{A1.6}$$

Cette décomposition permet donc le calcul direct de la matrice $({}^{t}Z\,Z)^{-1}$, dont les éléments s'écrivent :

$$\left({}^{t}ZZ\right)^{-1}_{lj} = \sum_{k=1}^{q} \frac{V_{lk}\,V_{jk}}{W_{kk}^{2}} \tag{A1.7}$$

On peut alors calculer l'expression de $h_{ii}$ sous la forme :

$$h_{ii} = {}^{t}\boldsymbol{z}^{i}\left({}^{t}Z\,Z\right)^{-1}\boldsymbol{z}^{i} = \sum_{l=1}^{q}\sum_{j=1}^{q} Z_{il}\,Z_{ij}\left({}^{t}Z\,Z\right)^{-1}_{lj} \tag{A1.8}$$

En utilisant (A1.7) puis en inversant l'ordre des sommes, il vient :

$$h_{ii} = \sum_{k=1}^{q}\left(\frac{1}{W_{kk}}\sum_{j=1}^{q} Z_{ij}\,V_{jk}\right)^{2} \tag{A1.9}$$

Cette méthode permet ainsi le calcul des $\{h_{ii}\}_{i\,=\,1,\,...,\,N}$ sans avoir à calculer explicitement les termes de la matrice $({}^{t}Z\,Z)^{-1}$, ce qui est important au niveau de la précision du calcul, dans le cas de matrices mal conditionnées. D'un point de vue numérique, étant donné que les valeurs singulières de $Z$ sont classées par ordre décroissant, il est conseillé de calculer les $\{h_{ii}\}_{i\,=\,1,\,...,\,N}$ en faisant varier $k$ de $q$ à 1 et non pas de 1 à $q$.

Cette méthode de calcul fournit des termes systématiquement positifs ou nuls, ce qui permet de s'assurer en partie de la condition (A1.2).

## ANNEXE 2 : SURAJUSTEMENT ET REDONDANCE DE COEFFICIENTS

La méthode présentée au paragraphe 2.4.2 pour détecter le surajustement part du constat suivant : à partir d'une certaine taille d'architecture, la matrice $Z$ - dont le rang est normalement égal au nombre de coefficients ajustables du modèle - n'est plus de rang plein, traduisant ainsi un surajustement. Cette interprétation de la déficience du rang de $Z$ suppose toutefois que l'architecture utilisée ne présente aucune redondance parmi ses coefficients. Nous précisons ce point dans la présente annexe.

Considérons par exemple le réseau représenté sur la figure A2.1, réalisant une fonction $y = f(x, \boldsymbol{w})$, dans laquelle $\boldsymbol{w}$ est un vecteur de 6 paramètres. À partir de la forme algébrique de $f$, on montre la relation suivante :

$$\forall \left(x,\boldsymbol{w}\right) \qquad w_{43}\frac{\partial f\left(x,\boldsymbol{w}\right)}{\partial w_{43}} = w_{30}\frac{\partial f\left(x,\boldsymbol{w}\right)}{\partial w_{30}} + w_{31}\frac{\partial f\left(x,\boldsymbol{w}\right)}{\partial w_{31}}$$

Cette relation traduit une dépendance entre les colonnes correspondantes de la matrice $Z$, qui - quel que soit le vecteur des paramètres - sera de rang inférieur ou égal à 5.

La relation précédente peut également se comprendre de la façon suivante : en multipliant le paramètre $w_{43}$ par un réel quelconque $k$ et en divisant $w_{30}$ et $w_{31}$ par ce même facteur, la fonction $f$ reste inchangée. On pourrait ainsi fixer arbitrairement un de ces trois poids à une valeur non nulle (par exemple à 1) sans modifier la famille de fonctions paramétrées engendrée par ce réseau. Parmi les 6 poids utilisés, il n'y en a donc que 5 réellement utiles : l'un d'eux est redondant.

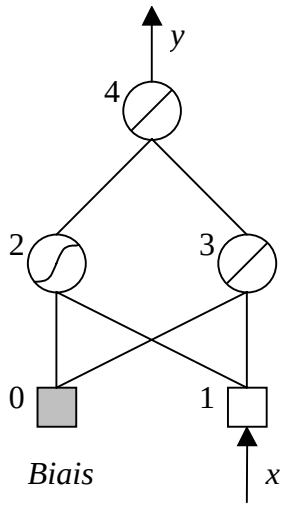


*Figure A2.1 : Exemple de modèle statique avec redondance de coefficients*

Dans le cas d'un modèle statique, ce cas ne se produit que lorsqu'on utilise plusieurs neurones linéaires en cascade, ce qui ne présente effectivement aucun intérêt. Il est donc difficile de ne pas se rendre compte de la présence de redondances dans l'architecture d'un réseau de neurones classique utilisé comme modèle statique.

En revanche, lorsque ces modèles sont bouclés de façon à modéliser des phénomènes dynamiques, ceci est moins évident à constater. Le modèle représenté sur la figure A2.2 permet de s'en convaincre. Il s'agit d'un modèle, comportant une entrée, une sortie et une variable d'état, réalisant une fonction du type :

$$\begin{cases} y(p+1) = f\left(x(p),\, y(p),\, s(p),\, \boldsymbol{w}\right) \\ s(p+1) = g\left(x(p),\, y(p),\, s(p),\, \boldsymbol{w}\right) \end{cases}$$

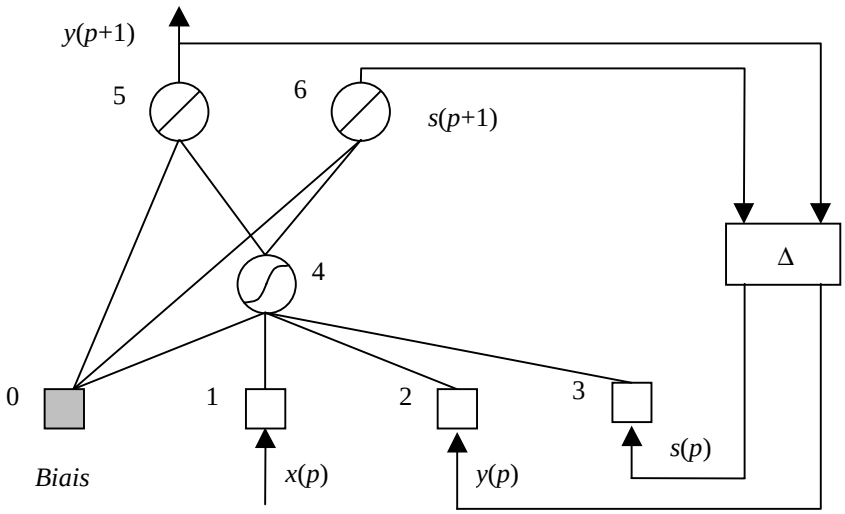


*Figure A2.2 : Exemple de modèle dynamique avec redondance de coefficients*

Lorsque l'on considère cette fonction sur un horizon fini, et que l'on en fixe l'état initial (c'est-à-dire $y(0)$ et $s(0)$), on peut écrire $y(p+1) = F(x(0), ..., x(p), \boldsymbol{w})$.

En l'absence de valeurs désirées pour la sortie d'état, et en appliquant le même type de raisonnement que dans le cas statique, on peut montrer que la matrice jacobienne de ce modèle, qui possède 8 coefficients, est toujours de rang inférieur ou égal à 5.

Pour ce faire, considérons les trois transformations linéaires suivantes, qui modifient la fonction $g$ tout en laissant la fonction $f$ invariante :

$\begin{pmatrix} w_{60} \to w_{60} + k \\ w_{40} \to w_{40} - k w_{43} \end{pmatrix} \Leftrightarrow s \to s + k$ : on peut donc fixer arbitrairement $w_{60}$ à 0,

$\begin{pmatrix} w_{60} \to k' w_{60} \\ w_{64} \to k' w_{64} \\ w_{43} \to w_{43} / k' \end{pmatrix} \Leftrightarrow s \to k' s$ : on peut donc fixer arbitrairement $w_{64}$ à 1,

$\begin{pmatrix} w_{60} \to w_{60} + k'' w_{50} \\ w_{64} \to w_{64} + k'' w_{54} \\ w_{42} \to w_{42} - k'' w_{43} \end{pmatrix} \Leftrightarrow s \to s + k'' y$ : on peut donc fixer arbitrairement $w_{42}$ à 0.

Pour chacune de ces transformations, il existe une relation linéaire entre les colonnes de la matrice jacobienne de $F$. Par exemple, pour la première de ces transformations, il existe une

relation linéaire, valable quels que soient $x(0), ..., x(p)$, entre $\frac{\partial y(p+1)}{\partial w_{40}}$, $\frac{\partial y(p+1)}{\partial w_{43}}$ et $\frac{\partial y(p+1)}{\partial w}$. L'expression de cette relation, dont les coefficients dépendent de $y(0)$, $s(0)$ et du vecteur $\boldsymbol{w}$, n'est pas aussi simple que dans l'exemple du modèle statique présenté plus haut.

Il y a donc 3 coefficients redondants dans l'architecture de la figure A2.2. Le modèle dynamique réalisant la même fonction *f*, sans aucune redondance, est représenté sur la figure A2.3 :

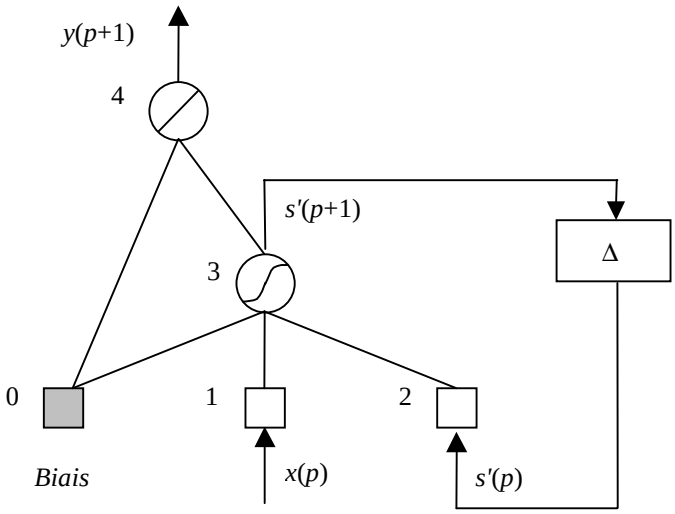


*Figure A2.3 : Modèle dynamique équivalent, sans coefficients redondants*

Cet exemple montre que le bouclage d'une sortie, ou d'une variable d'état, provenant d'un neurone linéaire peut conduire à des redondances de coefficients et ainsi à une déficience systématique dans le rang de *Z*. Il n'a cependant qu'une valeur illustrative : notre propos est simplement de mettre en garde contre certaines redondances "cachées" dans les modèles dynamiques.

La mise au point de règles concernant la définition d'architectures de modèles bouclés et de formes canoniques sans redondances demanderait sans doute une approche un peu plus poussée, qui sort du cadre de la présente thèse.

## ANNEXE 3 : DEMONSTRATION DE LA RELATION (3.8)

Nous présentons ici une démonstration de la relation (3.8), qui se fonde sur deux lemmes d'inversion matricielle. Elle est inspirée de [Antoniadis 92].

Considérons la matrice de dimensions ($q$+1, $q$+1) $M = \begin{pmatrix} {}^tZ\,Z & \boldsymbol{z}^i \\ {}^t\boldsymbol{z}^i & 1 \end{pmatrix}$ supposée inversible et notons son inverse $M^{-1} = \begin{pmatrix} A & \boldsymbol{b} \\ {}^i\boldsymbol{b} & c \end{pmatrix}$. La matrice $A$, le vecteur $\boldsymbol{b}$ et le réel $c$ vérifient donc les équations suivantes :

$${}^tZ\,Z\,A + \boldsymbol{z}^i\,{}^t\boldsymbol{b} = I \qquad \text{(A3.1)}$$

$${}^tZ\,Z\,\boldsymbol{b} + \boldsymbol{z}^i\,c = 0 \qquad \text{(A3.2)}$$

$${}^t\boldsymbol{b} + {}^t\boldsymbol{z}^i\,A = 0 \qquad \text{(A3.3)}$$

$$c + {}^t\boldsymbol{z}^i\,\boldsymbol{b} = 1 \qquad \text{(A3.4)}$$

**Lemme 1 :**

$$A = \left({}^tZ^{(-i)}\,Z^{(-i)}\right)^{-1} \qquad \text{(A3.5)}$$

Démonstration :

À partir de (A3.1) et (A3.3), on a ${}^tZ\,Z\,A - \boldsymbol{z}^i\,{}^t\boldsymbol{z}^i\,A = I$, soit $A = \left({}^tZ\,Z - \boldsymbol{z}^i\,{}^t\boldsymbol{z}^i\right)^{-1} = \left({}^tZ^{(-i)}\,Z^{(-i)}\right)^{-1}$

**Lemme 2 :**

$$A = \left({}^tZ\,Z\right)^{-1} + \frac{\left({}^tZ\,Z\right)^{-1}\boldsymbol{z}^i\,{}^t\boldsymbol{z}^i\left({}^tZ\,Z\right)^{-1}}{1 - h_{ii}} \qquad \text{(A3.6)}$$

Démonstration :

En résolvant le système linéaire constitué des équations (A3.1) à (A3.4), on obtient $c = \frac{1}{1 - h_{ii}}$, $\boldsymbol{b} = -\frac{\left({}^tZ\,Z\right)^{-1}\boldsymbol{z}^i}{1 - h_{ii}}$ et la relation (A.3.6). Ceci est un cas particulier d'un lemme d'inversion matricielle plus général.

**Démonstration de la relation (3.8) :**

En combinant (A3.5) et (A3.6), on obtient :

$$\left({}^{t}Z^{(-i)}\, Z^{(-i)}\right)^{-1} = \left({}^{t}Z\, Z\right)^{-1} + \frac{\left({}^{t}Z\, Z\right)^{-1} \boldsymbol{z}^{i}\, {}^{t}\boldsymbol{z}^{i} \left({}^{t}Z\, Z\right)^{-1}}{1 - h_{ii}} \qquad \text{(A3.7)}$$

Puis, à partir de la relation (3.3), restreinte au premier ordre, et de (3.6), il vient :

$$\boldsymbol{\theta}_{LS} - \boldsymbol{\theta}_{LS}^{(-i)} = \left({}^{t}Z^{(-i)}\, Z^{(-i)}\right)^{-1} {}^{t}Z^{(-i)} \left(\boldsymbol{y}_{\boldsymbol{p}}^{(-i)} - \boldsymbol{f}^{(-i)}\left(X, \boldsymbol{\theta}^{*}\right)\right) - \left({}^{t}Z\, Z\right)^{-1} {}^{t}Z \left(\boldsymbol{y}_{\boldsymbol{p}} - \boldsymbol{f}\left(X, \boldsymbol{\theta}^{*}\right)\right) \qquad \text{(A3.8)}$$

De plus, on peut facilement montrer que :

$${}^{t}Z^{(-i)} \left(\boldsymbol{y}_{\boldsymbol{p}}^{(-i)} - \boldsymbol{f}^{(-i)}\left(X, \boldsymbol{\theta}^{*}\right)\right) = {}^{t}Z \left(\boldsymbol{y}_{\boldsymbol{p}} - \boldsymbol{f}\left(X, \boldsymbol{\theta}^{*}\right)\right) - \boldsymbol{z}^{i} \left(y_{pi} - f\left(\boldsymbol{x}^{i}, \boldsymbol{\theta}^{*}\right)\right), \qquad \text{(A3.9)}$$

où $y_{pi}$ est la $i^{\text{ème}}$ composante du vecteur $\boldsymbol{y}_{p}$.

En combinant les équations (A3.7), (A3.8) et (A3.9), on démontre alors que :

$$\boldsymbol{\theta}_{LS}^{(-i)} - \boldsymbol{\theta}_{LS} = \frac{\left({}^{t}Z\, Z\right)^{-1} \boldsymbol{z}^{i}}{1 - h_{ii}} \left\{ \left(y_{pi} - f\left(\boldsymbol{x}^{i}, \boldsymbol{\theta}^{*}\right)\right) - {}^{t}\boldsymbol{z}^{i} \left({}^{t}Z\, Z\right)^{-1} {}^{t}Z \left(\boldsymbol{y}_{\boldsymbol{p}} - \boldsymbol{f}\left(X, \boldsymbol{\theta}^{*}\right)\right) \right\} \qquad \text{(A3.10)}$$

Le second terme à l'intérieur des accolades est la $i^{\text{ème}}$ composante de la projection de $\left(\boldsymbol{y}_{\boldsymbol{p}} - \boldsymbol{f}\left(X, \boldsymbol{\theta}^{*}\right)\right)$ sur le sous-espace des solutions ; par conséquent, la quantité à l'intérieur des accolades est la $i^{\text{ème}}$ composante du vecteur des résidus, c'est-à-dire le résidu de l'exemple *i*. Ceci prouve la relation (3.7).

# ANNEXE 4 : PROCEDE D'ORTHONORMALISATION DE GRAM-SCHMIDT

Nous présentons ici l'algorithme de Gram-Schmidt modifié, décrit en détail par exemple dans [Chen 89].

Le problème que nous considérons est le classement de $q$ entrées par ordre d'importance dans l'optique d'une régression linéaire par rapport aux paramètres.

Notons $N$ le nombre d'observations, $X = \begin{pmatrix} x_1^1 & \cdots & x_q^1 \\ \cdot & \cdots & \cdot \\ x_1^N & \cdots & x_q^N \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_q \end{pmatrix}$ la matrice des observations, avec $\boldsymbol{x}_i = \begin{pmatrix} x_i^1 \\ \cdot \\ x_i^N \end{pmatrix}$ le vecteur de la $i^{\text{ème}}$ entrée et $\boldsymbol{y} = \begin{pmatrix} y^1 \\ \cdot \\ y^N \end{pmatrix}$ le vecteur de la sortie du processus.

On suppose par ailleurs que les vecteurs $\boldsymbol{y}$ et $\{\boldsymbol{x}_i\}_{i = 1, \ldots, N}$ sont centrés, c'est-à-dire qu'on leur a appliqué une transformation affine de telle sorte que la moyenne de leurs composantes soit nulle.

On se place dans l'espace à $N$ dimensions engendré par les vecteurs $\boldsymbol{y}$ et $\{\boldsymbol{x}_i\}_{i = 1, \ldots, N}$.

La première itération consiste à chercher le vecteur d'entrée qui explique le mieux, au sens des moindres carrés, le vecteur de sortie. Pour cela, on se sert du carré du cosinus des angles entre le vecteur de sortie et les différents vecteurs d'entrée :

$$\cos^2\left(\boldsymbol{x}_i, \boldsymbol{y}\right) = \frac{\left({}^t\boldsymbol{x}_i \, \boldsymbol{y}\right)^2}{\left({}^t\boldsymbol{x}_i \, \boldsymbol{x}_i\right).\left({}^t\boldsymbol{y} \, \boldsymbol{y}\right)} \qquad \text{(A4.1)}$$

On sélectionne le vecteur d'entrée pour lequel cette quantité est maximale. Ensuite, on élimine la contribution de l'entrée sélectionnée en projetant le vecteur de sortie, et tous les vecteurs d'entrée restants, sur le sous-espace orthogonal au vecteur sélectionné.

La procédure se poursuit en choisissant, une nouvelle fois, le vecteur d'entrée projeté qui explique le mieux la sortie projetée. Elle se termine lorsque tous les vecteurs d'entrée ont été ordonnés.

Dans le cas d'un espace à 3 dimensions (c'est-à-dire avec 3 exemples) et 2 entrées, le principe de la première itération est représenté sur la figure A4.1.

Sur cet exemple, l'angle $\boldsymbol{\alpha}_1$ entre $\boldsymbol{y}$ et $\boldsymbol{x}_1$ étant plus petit que l'angle $\boldsymbol{\alpha}_2$ entre $\boldsymbol{y}$ et $\boldsymbol{x}_2$, l'entrée $\boldsymbol{x}_1$ est celle qui explique le mieux la sortie $\boldsymbol{y}$. On la sélectionne donc en premier et l'on projette les vecteurs $\boldsymbol{y}$ et $\boldsymbol{x}_2$ sur le plan perpendiculaire à $\boldsymbol{x}_1$, nommé $\boldsymbol{P}_1$. On recommence alors la procédure à partir des projections $\boldsymbol{p}_1(\boldsymbol{y})$ et $\boldsymbol{p}_1(\boldsymbol{x}_2)$ de ces vecteurs et $\boldsymbol{x}_2$ est alors naturellement la deuxième et dernière entrée sélectionnée.

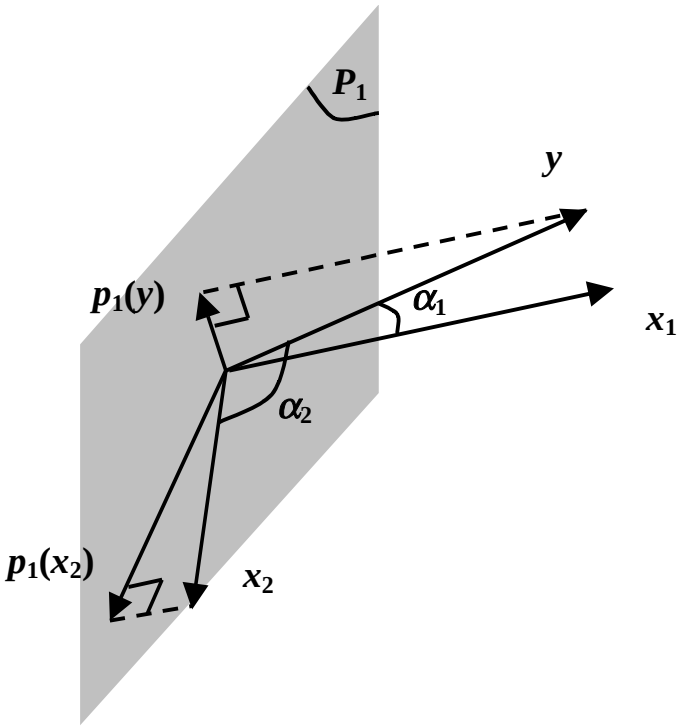


*Figure A4.1 : Principe de la méthode de Gram-Schmidt dans un espace à 3 dimensions*

## ANNEXE 5 : CARACTERISTIQUES DES PRODUITS UTILISES

Les caractéristiques et repérages des tôles d'acier revêtues utilisées dans ce travail sont résumés dans les tableaux suivants :

| Repère | Étude | Usine de provenance | N° de bobine | Nature de la tôle | Épaisseur nominale de la tôle | Nature du revêtement | Épaisseur nominale du revêtement |
|---|---|---|---|---|---|---|---|
| GZ 1 | 97058 | Montataire | 366105/2 | IF Ti | 0,8 mm | Galvanisé | 2 x 10 µm |
| GZ 2 | 98222 | Mardyck | Plusieurs (lot 98177) | IF Ti | 0,7 mm | Galvanisé | 2 x 10 µm |
| GA | 99088 | Mardyck | Plusieurs (lot 98977) | IF Ti | 1,2 mm | Galvanisé-allié | 2 x 10 µm |

*Tableau A5.1 : Repérages et caractéristiques des produits*

Pour les produits GZ 2 et GA, nous donnons ci-dessous la composition chimique et les caractéristiques mécaniques relevées sur une seule bobine, les autres bobines étant considérées comme ayant des propriétés comparables.

| Repère | C | Mn | P | S | Si | Al | Ni | Cr | Cu | Mo | Sn | Nb | V | Ti | B | Ca | N2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GZ 1 | 1,9 | 143 | 11 | 8,4 | 16 | 39 | 17 | 17 | 7 | <1 | 3 | 2 | 2 | 55 | <0,3 | <0,3 | 3,7 |
| GZ 2 | 2,4 | 94 | 6 | 6,7 | 5 | 30 | 20 | 15 | 8 | <1 | 1 | <1 | 2 | 55 | <0,3 | <0,3 | 2,8 |
| GA | 1,7 | 95 | 12 | 8,5 | 6 | 39 | 18 | 18 | 10 | <1 | 1 | <1 | 2 | 55 | 0,4 | <0,3 | 2,2 |

*Tableau A5.2 : Composition chimique des échantillons (teneur massique $10^{-3}$ %)*

| Repère | Limite élastique à 0,2 % (Mpa) | | Contrainte à rupture (Mpa) | | Allongement à rupture (%) | |
|---|---|---|---|---|---|---|
| | Sens long | Sens travers | Sens long | Sens travers | Sens long | Sens travers |
| GZ 1 | 170 | 152 | 304 | 299 | 42,8 | 44,2 |
| GZ 2 | 166 | 170 | 296 | 293 | 46,6 | 46,0 |
| GA | 182 | 187 | 310 | 308 | 44.4 | 39.8 |

*Tableau A5.3 : Caractéristiques mécaniques des échantillons*

| Repère | Effort (daN) | Durée de soudage (périodes) | Durée de forgeage (périodes) |
|---|---|---|---|
| GZ 1 | 230 | 10 | 10 |
| GZ 2 | 210 | 9 | 9 |
| GA | 300 | 14 | 14 |

*Tableau A5.4 : Paramètres de soudage suivant [NF A 87.001]*