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 μ, 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).

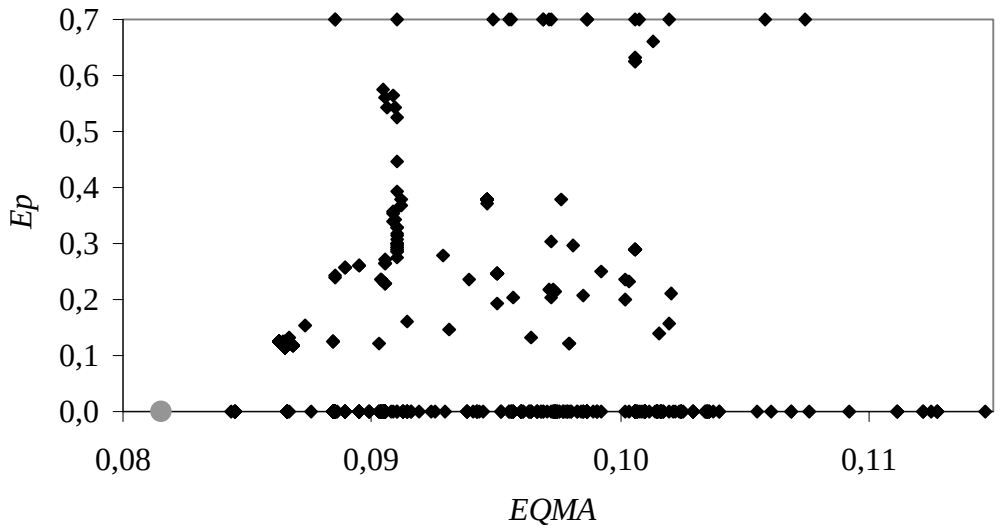

Figure 4.1 : Distribution des minima pour une architecture à 4 neurones cachés : cas de l'exemple $\frac{\sin(x)}{x}$

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.

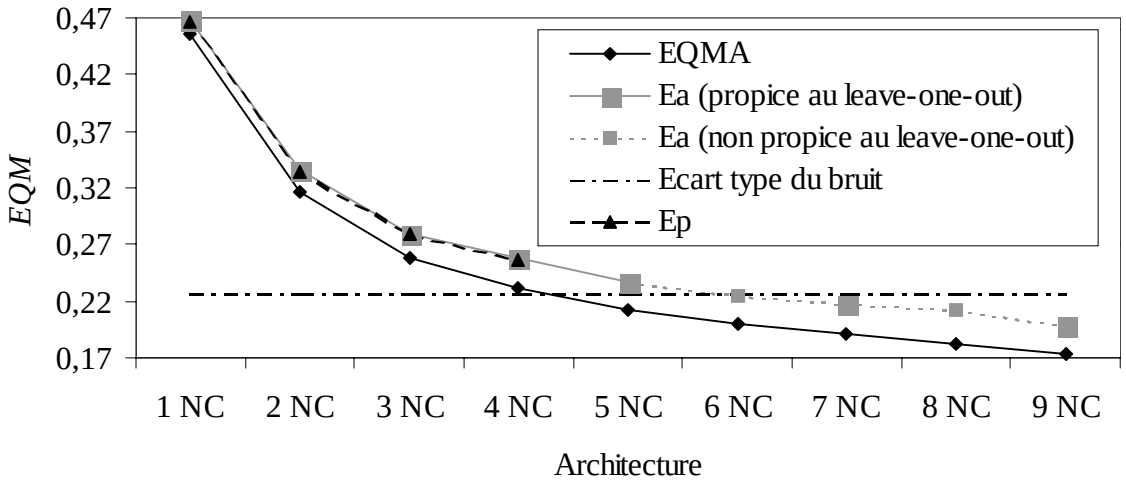

Figure 4.2 : Sélection de modèles sur la base de l'EQMA (problème maître élève)

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* !

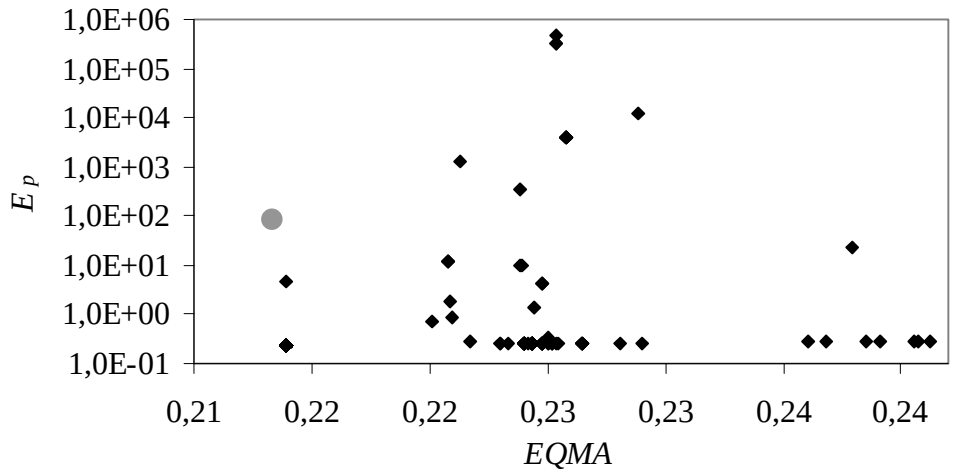

Figure 4.3 : Distribution des minima de rang plein pour une architecture à 5 neurones cachés : cas du problème maître-élève

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.

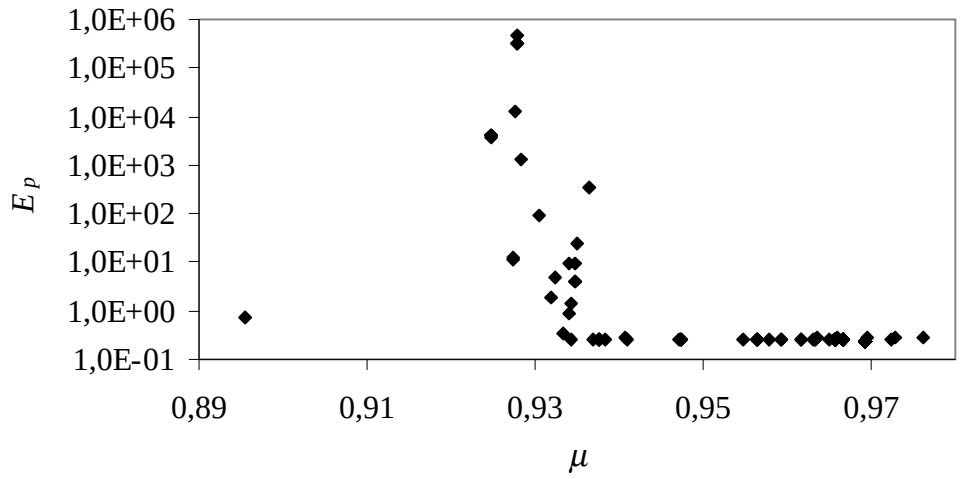

Figure 4.4 : Répartition du couple (μ, E_p) pour les minima de la figure 4.3

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 μ : 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 μ les plus élevées. Nous reviendrons en détail sur ce constat et sur l'utilisation du paramètre μ 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.

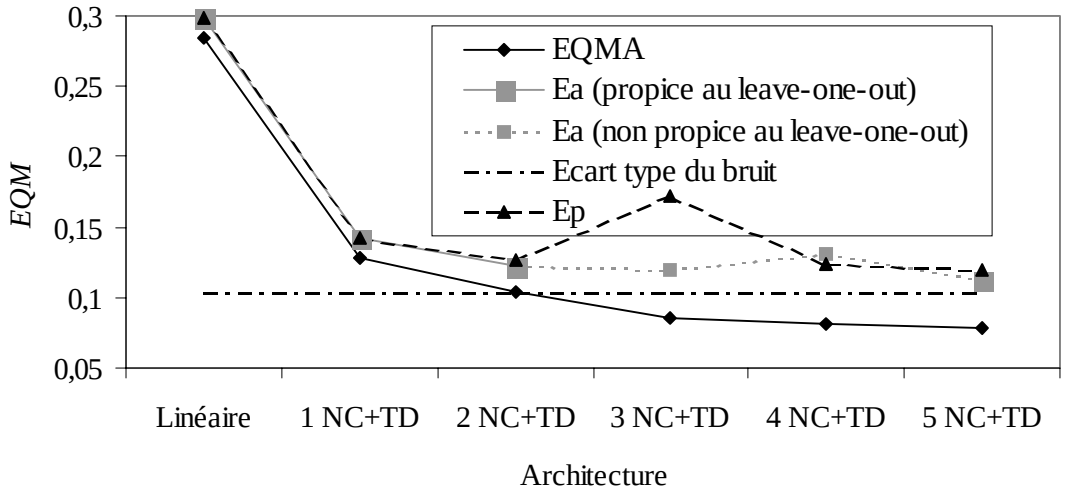

Figure 4.5 : Sélection de modèles sur la base de l'EQMA parmi les minima de rang plein (cas de l'exemple $\frac{\sin(x)}{}$)

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 E_p calculable	Z de rang non plein E_p non calculable
Propice au leave-one-out E_a calculable	E_p est une approximation de E_a	Surajustement avec déficience du rang E_p et E_a non comparables
Non propice au leave-one-out 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.

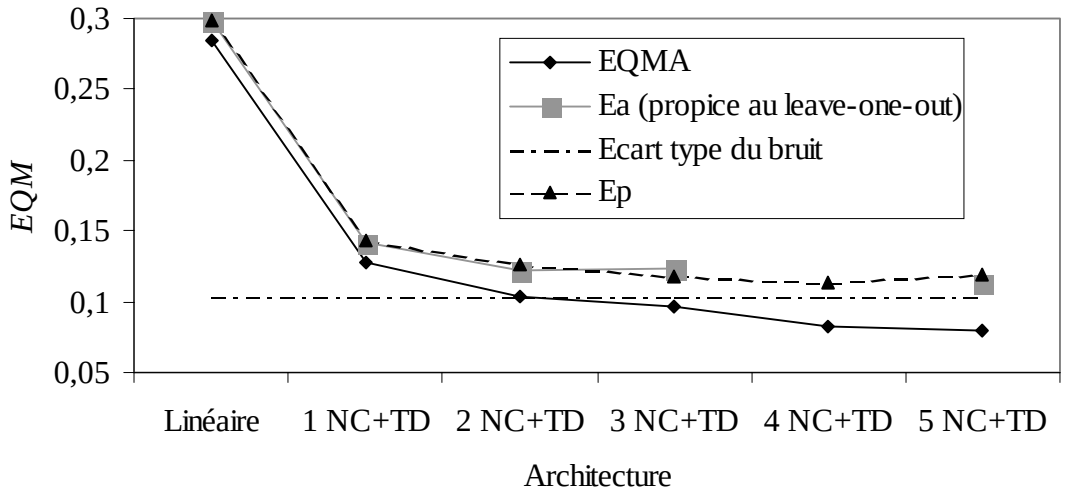

Figure 4.6 : Sélection de modèles sur la base de E_p (cas de l'exemple $\frac{\sin(x)}{x}$)

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.

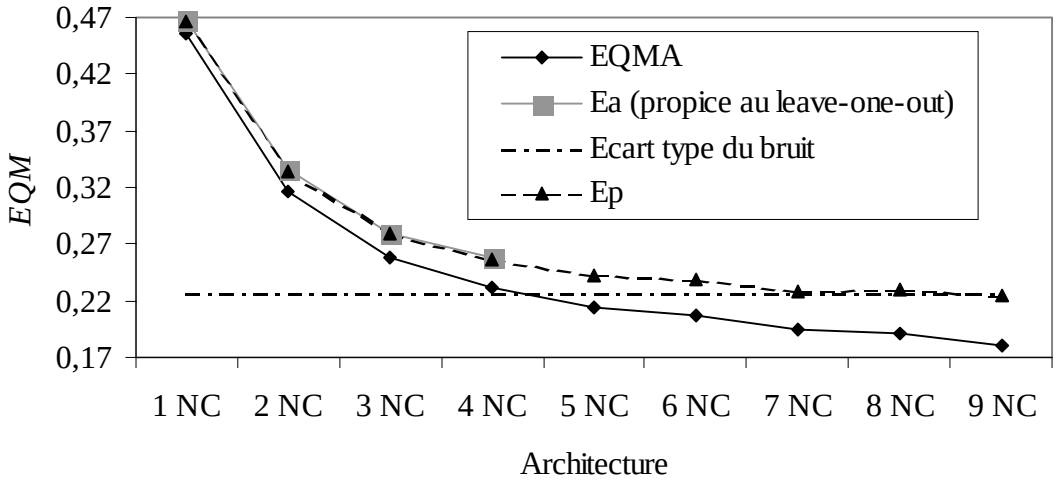

Figure 4.7 : Sélection de modèles sur la base de E_p (problème maître-élève)

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.

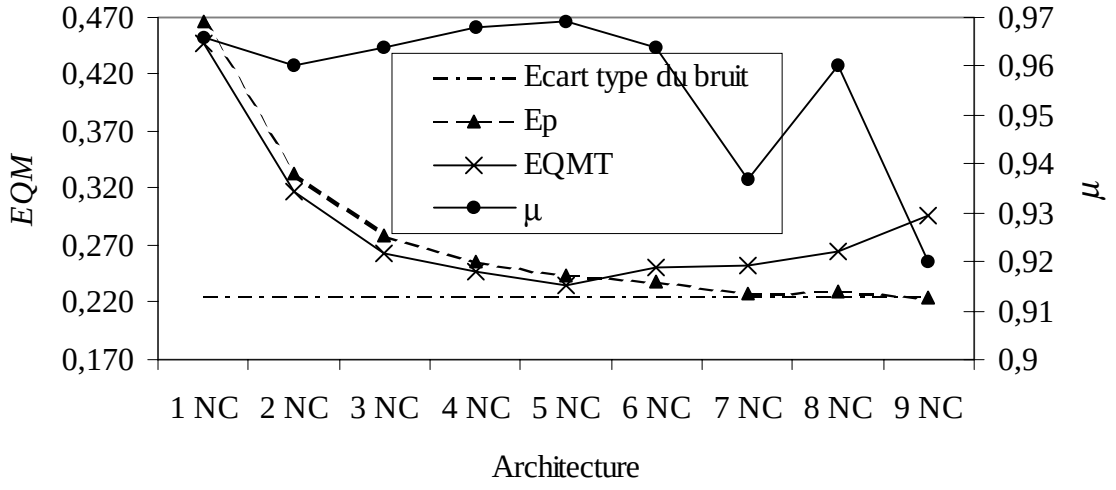

Figure 4.8 : Comparaison entre E_p et EQMT sur le problème maître-élève suite à la sélection des minima sur la base de E_p

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 μ. Remarquons que le minimum de l'*EQMT* correspond au maximum de μ : nous reviendrons sur ce point dans le paragraphe 4.5.

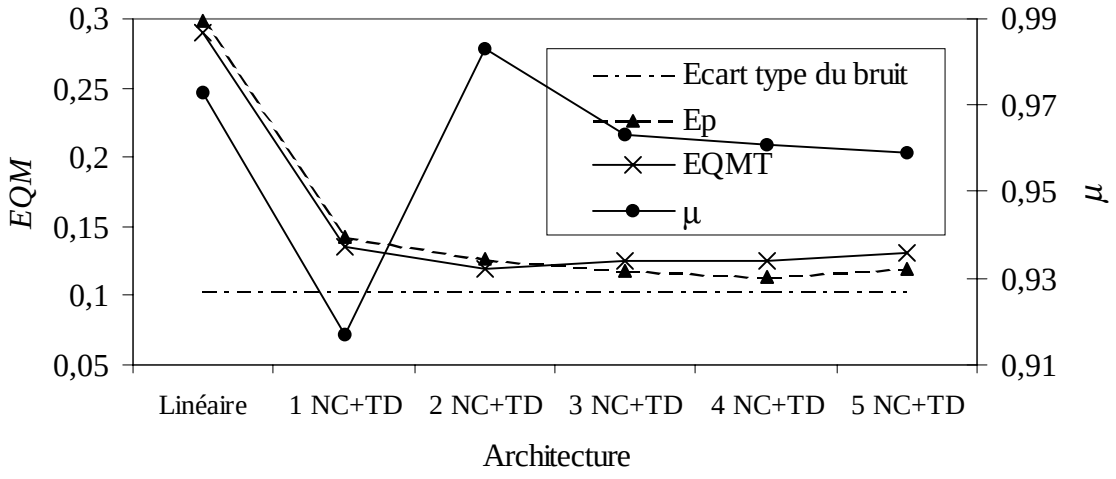

Figure 4.9 : Comparaison entre E_p et EQMT sur l'exemple $\frac{\sin(x)}{x}$ suite à la sélection des minima sur la base de E_p

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 μ 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 μ.

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

- pour l'architecture correspondant au maximum de μ, 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.

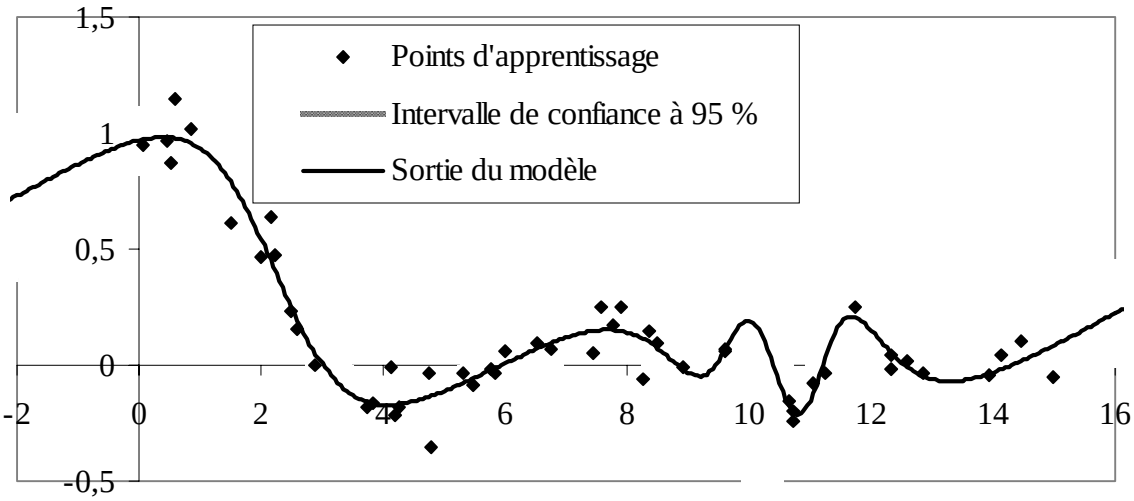

Figure 4.10 : Modèle à 4 neurones cachés sélectionné par E_p sur l'exemple $\frac{\sin(x)}{x}$

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.

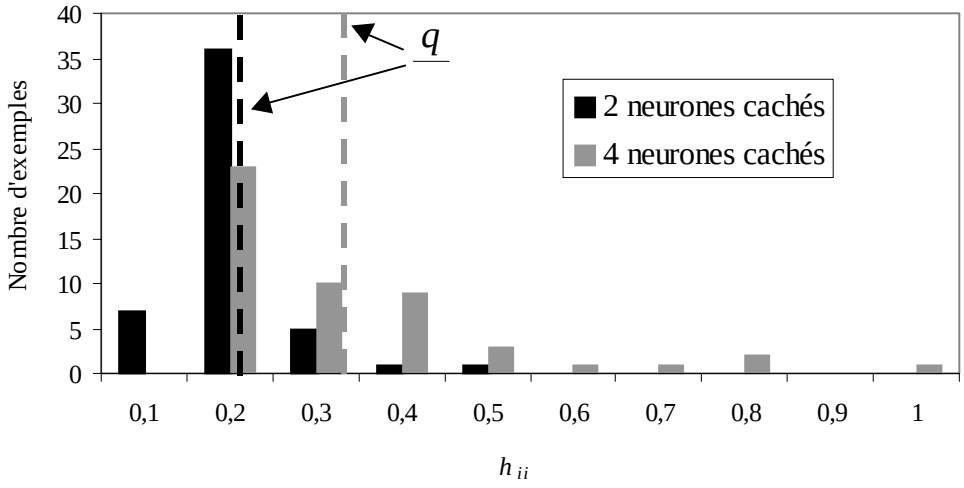

Figure 4.11 : Distribution des $\{h_{ii}\}_{i = 1, \ldots, N}$ *pour les modèles à* 2 *et* 4 *neurones cachés sur l'exemple* $\frac{\sin(x)}{}$

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 μ 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'α, s, N et q. Nous proposons d'utiliser μ 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 μ vaut 1, correspond ainsi à un maximum de μ. Plus μ 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 μ 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 μ 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 μ - 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).

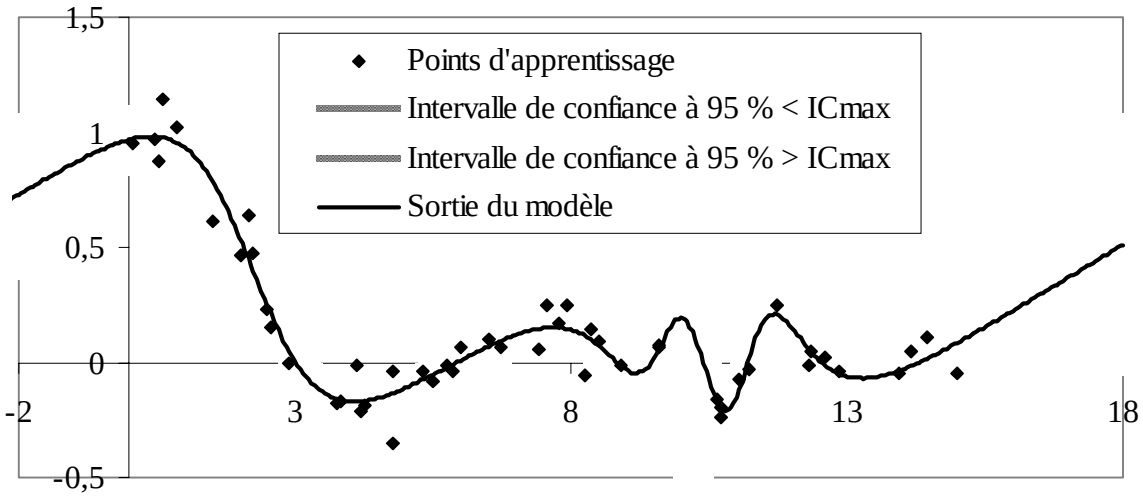

Figure 4.12 : Application du seuil IC_{max} au modèle de la figure 4.10

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 μ 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 μ sera le plus proche de 1.