Proceedings of the International Workshop on Advanced Black-Box Techniques for Nonlinear Modeling, 8-10 July 1998, Leuwen.

CONSTRUCTION OF CONFIDENCE INTERVALS IN NEURAL MODELING USING A LINEAR TAYLOR EXPANSION

Isabelle Rivals and Léon Personnaz
École Supérieure de Physique et de Chimie Industrielles,
10 rue Vauquelin, 75231 Paris Cedex 05, France.
Phone: 00 33 1 40 79 45 45 Fax: 00 33 1 40 79 44 25
E-mail: Isabelle.Rivals@espci.fr

Abstract: We introduce the theoretical results on the construction of confidence intervals for a nonlinear regression, based on the linear Taylor expansion of the corresponding nonlinear model output. The case of neural black-box modeling is then analyzed, and illustrated on an industrial application. We show that the linear Taylor expansion not only provides a confidence interval at any point of interest, but also gives a tool to detect overfitting.

Keywords: backpropagation algorithm, black-box modeling, bootstrap methods, confidence intervals, least squares estimator, linear Taylor expansion, neural networks, overfitting detection.

I. INTRODUCTION

In neural network modeling studies, generally only an average estimate of a neural model reliability is given through its mean square error on a test set. Yet, the problem of the estimation of a given model reliability has been investigated to a great extent in nonlinear regression theory (see for example [Bates & Watts 88] [Seber & Wild 89]). In this framework, section II presents the construction of confidence intervals (CIs) for a nonlinear regression based on the least squares (LS) solution applied to the linear Taylor expansion (LTE) of the corresponding nonlinear model output; an illustrative example using a simulated process is given. In section III, we exploit this method for practical modeling problems involving neural models. We show that the LTE not only provides a CI at any point of interest in the input space, but also gives a tool to detect overfitting, and thus to perform a good selection among candidate neural models. An illustration is given through an industrial application, the modeling of the elasticity of a complex material from some of its structural characteristics. Section IV finally discusses the advantages of the LTE based method with respect to the computationally intensive bootstrap methods and to second-order analytic methods.

We deal with static single-output models with a non random n-input vector $\boldsymbol{x} = [x_1\ x_2\ \dots\ x_n]^T$ and a noisy measured output y_p which is considered as the actual value of a random variable $Y_p \mid \boldsymbol{x}$ depending on $\boldsymbol{x}$. We assume that there exists an unknown regression function such that for any fixed value of $\boldsymbol{x}$:

$$Y_p \mid \boldsymbol{x} = E\left(Y_p \mid \boldsymbol{x}\right) + W \mid \boldsymbol{x} \qquad (1)$$

where $W \mid \boldsymbol{x}$ is a random variable with zero expectation depending on the input vector $\boldsymbol{x}$. A family of parameterized functions $\left\{f(\boldsymbol{x}, \boldsymbol{\theta}),\ \boldsymbol{x} \in \mathbb{R}^n,\ \boldsymbol{\theta} \in \mathbb{R}^q\right\}$ is called an *assumed model*. This assumed model is said to be *true* if there exists a value $\boldsymbol{\theta}_p$ of $\boldsymbol{\theta}$ such that $\forall\, \boldsymbol{x}\ \ f(\boldsymbol{x}, \boldsymbol{\theta}_p) = E\left(Y_p \mid \boldsymbol{x}\right)$. In the following, a data set consisting of N input-output pairs $\left\{\boldsymbol{x}^k, y_p^k\right\}_{k=1 \text{ to } N}$ will be available, where the $\boldsymbol{x}^k = \left[x_1^k\ x_2^k\ \dots\ x_n^k\right]^T$ are non random n-vectors, and the $\left\{y_p^k\right\}$ are the corresponding values of the random variables $\left\{Y_p^k = Y_p \mid \boldsymbol{x}^k\right\}$.

We will usually distinguish between random variables and their values by using upper- and lowercase letters, e.g. Y_p^k and y_p^k; all vectors are column vectors, and are denoted by boldface letters, e.g. the n-vectors $\boldsymbol{x}$ and $\left\{\boldsymbol{x}^k\right\}$; non random matrices are denoted by light lowercase letters.

II. APPROXIMATE CONFIDENCE INTERVALS FOR NONLINEAR MODELS

In this section, we consider a family of parameterized functions $\left\{f(\boldsymbol{x}, \boldsymbol{\theta}),\ \boldsymbol{x} \in \mathbb{R}^n,\ \boldsymbol{\theta} \in \mathbb{R}^q\right\}$ which contains the regression, i.e. the assumed model is known to be true; (1) can thus be rewritten as:

$$Y_p \mid \boldsymbol{x} = f(\boldsymbol{x}, \boldsymbol{\theta}_p) + W \mid \boldsymbol{x} \qquad (2)$$

where $\boldsymbol{\theta}_p$ is an unknown q-parameter vector. Let us consider the random data set $\left\{\boldsymbol{x}^k, Y_p^k\right\}_{k=1 \text{ to } N}$, and denote by $\boldsymbol{f}(x, \boldsymbol{\theta}_p)$ the vector with components: $\left[f(\boldsymbol{x}^1, \boldsymbol{\theta}_p) \dots f(\boldsymbol{x}^k, \boldsymbol{\theta}_p) \dots f(\boldsymbol{x}^N, \boldsymbol{\theta}_p)\right]^T$. We use the matrix notation:

$$\boldsymbol{Y}_p = \boldsymbol{f}(x, \boldsymbol{\theta}_p) + \boldsymbol{W} \qquad (3)$$

where x denotes the $N \times n$ values $\left\{x_i^k\right\}$ (k=1 to N, i=1 to n), and $\boldsymbol{Y}_p$ and $\boldsymbol{W}$ are random N-vectors with $E(\boldsymbol{W}) = \boldsymbol{0}$. Geometrically, this means that $E\left(\boldsymbol{Y}_p \mid x\right)$ belongs to the solution surface, the manifold $\mathcal{M} = \left\{\boldsymbol{f}(x, \boldsymbol{\theta}),\ \boldsymbol{\theta} \in \mathbb{R}^q\right\}$ of $\mathbb{R}^q$, which is assumed to be of dimension q.

II.1. The linear Taylor expansion of the nonlinear least squares solution

The LS parameter estimator $\boldsymbol{\Theta}_{LS}$ is the estimator whose value $\boldsymbol{\theta}_{LS}$ minimizes the empirical quadratic cost-function:

$$J(\boldsymbol{\theta}) = \left(\boldsymbol{y_p} - \boldsymbol{f}(x, \boldsymbol{\theta})\right)^T \left(\boldsymbol{y_p} - \boldsymbol{f}(x, \boldsymbol{\theta})\right) \qquad (4)$$

A global minimum of cost-function (4) can be obtained with efficient second-order algorithms. However, an analytic expression of a minimum, i.e. of $\boldsymbol{\theta}_{LS}$, which could be used to build CIs is not available (as opposed to the case of a linear assumed model $\boldsymbol{Y}_p = x\,\boldsymbol{\theta}_p + \boldsymbol{W}$, for which $\boldsymbol{\theta}_{LS} = \left(x^T x\right)^{-1} x^T \boldsymbol{y}_p$). Nevertheless, a linear

expansion of $\boldsymbol{\Theta}_{LS}$ is obtained by writing the LTE of $f(\boldsymbol{x}, \boldsymbol{\theta})$ around $f(\boldsymbol{x}, \boldsymbol{\theta}_p)$ for a given value $\boldsymbol{x}$ of the input:

$$f(\boldsymbol{x}, \boldsymbol{\theta}) \approx f(\boldsymbol{x}, \boldsymbol{\theta}_p) + \xi^T (\boldsymbol{\theta} - \boldsymbol{\theta}_p) \qquad (5)$$

where $\xi = \left.\dfrac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta} = \boldsymbol{\theta}_p}$.

In the case of multilayer neural network models, the global minimum of the cost-function can be obtained for different values of the parameter vector; but since the only function-preserving transformations are neuron exchanges and sign flips (for odd activation functions) [Sussman 92], we will legitimately consider the neighborhood of one of these values only. With the matrix notation, we obtain:

$$\boldsymbol{f}(x, \boldsymbol{\theta}) \approx \boldsymbol{f}(x, \boldsymbol{\theta}_p) + \xi (\boldsymbol{\theta} - \boldsymbol{\theta}_p) \qquad (6)$$

where $\xi = \left[\xi^1 \ \xi^2 \ \ldots \ \xi^N\right]^T$ and $\xi^k = \left.\dfrac{\partial f(\boldsymbol{x}^k, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta} = \boldsymbol{\theta}_p}$.

The (N,q) matrix ξ is the non random but unknown (since $\boldsymbol{\theta}_p$ is unknown) jacobian matrix of $\boldsymbol{f}$. Using (3) and (6), we obtain the following expansion of $\boldsymbol{\Theta}_{LS}$:

$$\boldsymbol{\Theta}_{LS} \approx \boldsymbol{\theta}_p + (\xi^T \xi)^{-1} \xi^T \boldsymbol{W} \qquad (7)$$

This means that, since the assumed model is true, the LS estimator is asymptotically (as N tends to infinity) unbiased. Let us denote by $\pi = \xi(\xi^T \xi)^{-1} \xi^T$ the orthogonal projection matrix on the range of ξ. The latter is tangent to the manifold $\mathcal{M}$ at $\boldsymbol{\theta} = \boldsymbol{\theta}_p$, and is denoted by $\mathcal{L}$; it is also assumed of dimension q. From (6) and (7), the LS estimator of $E(\boldsymbol{Y}_p \mid x)$ can be expanded as:

$$\boldsymbol{f}(x, \boldsymbol{\Theta}_{LS}) \approx \boldsymbol{f}(x, \boldsymbol{\theta}_p) + \pi \boldsymbol{W} \qquad (8)$$

i.e. it is approximately the sum of $E(\boldsymbol{Y}_p \mid x)$ and of $\pi \boldsymbol{W}$, the projection of $\boldsymbol{W}$ on $\mathcal{L}$, as illustrated in Figure 1.

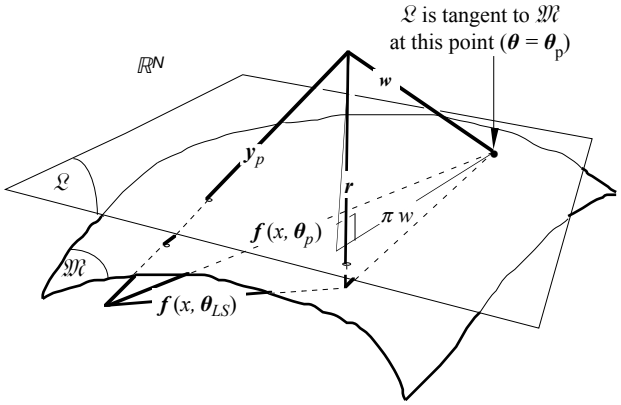

Figure 1. Representation of the nonlinear LS solution and of its LTE.

Let $\boldsymbol{R} = \boldsymbol{Y}_p - \boldsymbol{f}(x, \boldsymbol{\Theta}_{LS})$ denote the residual vector, thus:

$$\boldsymbol{R} \approx (I_N - \pi) \boldsymbol{W} \qquad (9)$$

where I_N denotes the (N, N) identity matrix. Under the assumption of appropriate regularity conditions on f, and for large N, the curvature of the solution surface $\mathcal{M}$ is small. Thus, if the assumption of homoscedasticity can be made i.e. $K(\boldsymbol{W}) = \sigma^2 I_N$, the variance $\sigma^2_Y(\boldsymbol{x})$ of the LS estimator $Y = f(\boldsymbol{x}, \boldsymbol{\Theta}_{LS})$ of the regression for a fixed input $\boldsymbol{x}$ can be approached with:

$$\sigma^2 \xi^T (\xi^T \xi)^{-1} \xi \qquad (10)$$

In the following, (10) is termed the LTE variance. Using (9), we obtain an estimator S^2 of σ^2:

$$S^2 = \frac{\boldsymbol{R}^T \boldsymbol{R}}{N - q} \qquad (11)$$

which is asymptotically unbiased. For future computations, the unknown matrix ξ in (10) may be approximated by:

$$z = \left.\frac{\partial \boldsymbol{f}(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\right|_{\boldsymbol{\theta} = \boldsymbol{\theta}_{LS}} \qquad (12)$$

Similarly, no vector $\boldsymbol{\xi}$ is available, and its value may be approximated by:

$$z = \left.\frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta} = \boldsymbol{\theta}_{LS}} \qquad (13)$$

Finally, from relations (10) to (13), an estimate of the variance $\sigma^2_Y(\boldsymbol{x})$ is:

$$\widehat{\sigma^2_Y(\boldsymbol{x})} = s^2 \, z^T (z^T z)^{-1} z \qquad (14)$$

where s is the value of the random variable S. In the following, (14) is termed the LTE variance estimate.

II.2. Approximate confidence intervals for the regression $E(Y_p \mid \boldsymbol{x})$

If $\boldsymbol{W} \rightsquigarrow \mathbf{N}(\mathbf{0}, \sigma^2 I_N)$, i.e. the gaussian assumption holds, it follows from the above relations and from linear LS statistical properties that *asymptotically* [Seber & Wild 89]:

$$\boldsymbol{\Theta}_{LS} \rightsquigarrow \mathbf{N}(\boldsymbol{\theta}_p, \sigma^2 (\xi^T \xi)^{-1}) \qquad (15)$$

$$\frac{\boldsymbol{R}^T \boldsymbol{R}}{\sigma^2} \rightsquigarrow \chi^2(N-q) \qquad (16)$$

and that $\boldsymbol{\Theta}_{LS}$ is statistically independent from $\boldsymbol{R}^T \boldsymbol{R}$.

Our goal is to build a CI for $E(Y_p \mid \boldsymbol{x})$, where $\boldsymbol{x}$ is any fixed input vector of interest. From (10) and the preceding results, it can be shown that *asymptotically*:

$$T = \frac{f(\boldsymbol{x}, \boldsymbol{\Theta}_{LS}) - E(Y_p \mid \boldsymbol{x})}{S \sqrt{\xi^T (\xi^T \xi)^{-1} \xi}} \rightsquigarrow \mathit{Student}(N-q) \qquad (17)$$

An *approximate* CI for $E(Y_p \mid \boldsymbol{x})$ with a level of significance $1 - \alpha$ is thus:

$$f(\boldsymbol{x}, \boldsymbol{\theta}_{LS}) \pm t_{N-q}(\alpha) \, s \sqrt{z^T (z^T z)^{-1} z} \qquad (18)$$

where $t_{N-q}(\alpha)$ is the corresponding Student value. (18) allows to compute a CI at any input vector $\boldsymbol{x}$ of interest. From a practical point of view, it only involves once and for all the computation of the $\{z^k\}_{k=1 \text{ to } N}$, i.e. the gradients of the model output with respect to the parameters at the data inputs $\{\boldsymbol{x}^k\}_{k=1 \text{ to } N}$, and that of $\boldsymbol{z}$, the gradient at the input $\boldsymbol{x}$ of interest. In the case of a neural model, these gradients are easily obtained with the backpropagation algorithm.

II.3. Quality of the approximate confidence intervals

Both the regularity of f and the size of N influence the curvature of the solution surface, i.e. the degree of bending and twisting of the manifold $\mathcal{M}$; measures of curvatures can be found in [Bates & Watts 88] [Seber & Wild 89] [Antoniadis et al. 92]. As a matter of fact, f is often regular enough for a first-order expansion to be satisfactory, provided that N is large.

If N is large enough, the curvature of the solution surface is small: (i) as in the linear case, the estimator of the noise variance S^2 is unbiased, and the difference

between s^2 and σ^2 is only due to the random character of the data set outputs; (ii) the variance $\sigma_Y^2(\boldsymbol{x})$ of $f(\boldsymbol{x}, \boldsymbol{\theta}_{LS})$ is globally small, and $\boldsymbol{\theta}_{LS}$ is likely to be close to $\boldsymbol{\theta}_p$: z and $\boldsymbol{z}$ are thus likely to be good approximations of respectively ξ and $\boldsymbol{\xi}$. A CI based on the LTE variance estimate (14) is thus reliable.

If N is too small, the curvature is large: (i) as opposed to the linear case, S^2 is biased; (ii) $\sigma_Y^2(\boldsymbol{x})$ is globally large, and $\boldsymbol{\theta}_{LS}$ is likely to differ from $\boldsymbol{\theta}_p$: z and $\boldsymbol{z}$ risk to be poor approximations of ξ and $\boldsymbol{\xi}$. Thus, the approximate CIs (18) are not reliable.

The above considerations are valid provided that a global minimum of cost-function (4) is reached. It is thus necessary to use an efficient second-order algorithm, to perform several trainings starting from different initial values of the parameters, and to keep the estimates corresponding to the lowest value of (4). In this work, we use the Levenberg algorithm as described for example in [Bates & Watts 88].

II.4. Illustrative example

We consider a single-input single-output process simulated by a neural network with one hidden layer of two *tanh* hidden neurons and a linear output neuron:

$$y_p^k = \theta_{p1} + \theta_{p2}\, tanh\left(\theta_{p3} + \theta_{p4}\, x^k\right) + \theta_{p5}\, tanh\left(\theta_{p6} + \theta_{p7}\, x^k\right) + w^k \quad (19)$$

with $\boldsymbol{\theta}_p = [1;\ 2;\ 1;\ 2;\ -1;\ -1;\ 3]^T$, N=50, and the $\{w^k\}$ are the values of independent gaussian variables with variance $\sigma^2=10^{-2}$. The $\{x^k\}$ are uniformly distributed in [–3; 3]. The true assumed model is considered, i.e. a neural network with the same architecture as that of the simulated process. The minimization of the cost-function with the Levenberg algorithm leads to $s^2=1.02\ 10^{-2}$. The true variance of the LS estimator of the regression is unknown, and the LTE variance (10) is a good approximation of it only if the curvature is negligible. In order to evaluate the accuracy of (10) with respect to the unknown true variance $\sigma_Y^2(\boldsymbol{x})$ of $f(\boldsymbol{x}, \boldsymbol{\theta}_{LS})$, i.e. the effect of the curvature, and the accuracy of the LTE variance estimate (14) from which the CIs are derived, a reference estimate of this true variance is computed using a large number M of other sets, whose inputs are the above $\{x^k\}$, and whose outputs are obtained by simulating different values of the $\{W^k\}$. The LS estimate $f\left(\boldsymbol{x}, \boldsymbol{\theta}_{LS}^{(i)}\right)$ of $E\left(Y_p \mid \boldsymbol{x}\right)$ is computed on each data set i (i=1 to M), and the reference estimate $\sigma_{ref}^2(\boldsymbol{x})$ of the true variance for any input $\boldsymbol{x}$ is computed as:

$$\langle f(\boldsymbol{x})\rangle = \frac{1}{M}\sum_{i=1}^{M} f\left(\boldsymbol{x}, \boldsymbol{\theta}_{LS}^{(i)}\right)$$
$$\sigma_{ref}^2(\boldsymbol{x}) = \frac{1}{M}\sum_{i=1}^{M}\left(f\left(\boldsymbol{x}, \boldsymbol{\theta}_{LS}^{(i)}\right) - \langle f(\boldsymbol{x})\rangle\right)^2 \quad (20)$$

Figure 2a) displays the reference estimate (20) computed over M=10000 sets, the LTE variance (10) computed with $\boldsymbol{\theta}_p$, and the LTE variance estimate (14) computed with $\boldsymbol{\theta}_{LS}$; the latter is the only estimate available in real life. Figure 2b) shows the regression, the data set, the model output and its 99% approximate CI (18).

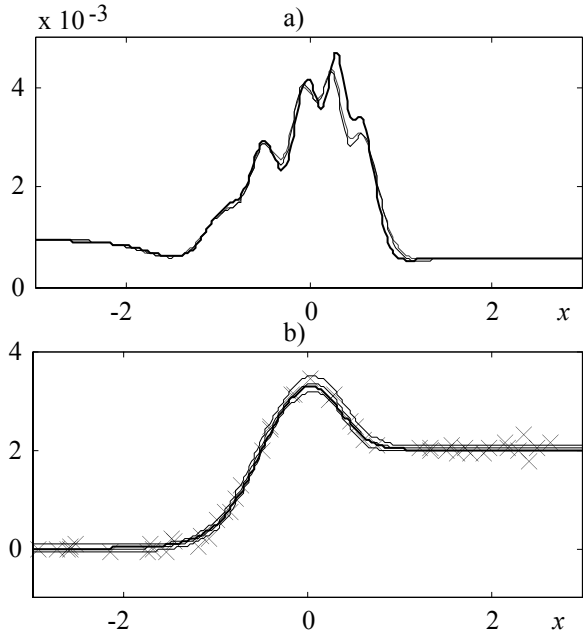

Figure 2. a) LTE variance (thin line), LTE variance estimate (thin dotted line), and reference variance estimate (thick line) of $f(\boldsymbol{x}, \boldsymbol{\Theta}_{LS})$; b) data set (crosses), regression (thick line), model output and 99% LTE CI (thin lines).

The variance of the noise being small, the model output is close to the regression (i.e. $\boldsymbol{\theta}_{LS}$ is close to $\boldsymbol{\theta}_p$); z and the $\boldsymbol{z}$'s are thus good approximations of ξ and of the $\boldsymbol{\xi}$'s. It follows that the LTE variance estimate (14) is close to the LTE variance (10). Moreover, the size N of the data set is relatively large with respect to the complexity of the regression, i.e. the curvature is small: we thus obtain a good estimation of the true variance through the LTE variance (10), so that the CIs are accurate.

III. CONFIDENCE INTERVALS USING NEURAL NETWORKS

For most real-world black-box modeling problems, a family of functions which contains the regression is unknown *a priori*, as opposed to the situation considered in the preceding example. Theoretically, the goal is to select the smallest family of functions which contains the regression to a certain degree of accuracy. In practice, several families of increasing complexity (i.e. neural networks with an increasing number of hidden units) are considered, and the data set is used both to estimate their parameters, and to perform the selection between candidate networks.

III.1. Overfitting detection for model selection

For example, the data set is partitioned in training and validation set or sets (cross-validation), and the smallest family of functions leading to the smallest mean square error on the validation set(s) is selected; a model of the selected family is then trained on the whole data set. If

the data set size N is small, the disadvantage of cross-validation is to use a training set which is even smaller: the best choice is then leave-one-out cross-validation, but with the drawback of requiring N successful trainings of the candidate models. Another possibility is to make use of statistical tests. With this method, one must choose a large model which is assumed to contain the regression (i.e. it is a true assumed model), and test if smaller models are sufficient [Antoniadis et al. 92] [Urbani et al. 94].

In any case, one must ensure that none of the considered candidate networks overfits, i.e. that all their neurons or parameters are useful. To detect overfitting, we propose, after the training of each neural network, to check the rank of z. As a matter of fact, if z is not of full rank, i.e. $rank\,(z) < q$, $\mathcal{M}$ is locally not of dimension q, which means that at least one of the parameters is not useful. This is usually the case when the model is too large with respect to the regression complexity and to the noise amplitude, and when overfitting has thus occurred. A typical situation is the saturation of one hidden neuron, a situation which generates in the matrix z a column of +1 or –1 that corresponds to the parameter between the saturated hidden neuron and the output, and columns of zeros that correspond to the parameters between the inputs and the hidden neuron.

In practice, one can for example perform a singular value decomposition of z, and decide whether all singular values are significantly different from zero [Nash 90], or test if the conditioning number (ratio of the smallest to the largest singular value) is too small. Nevertheless, it is often difficult to elaborate a tolerance to make the above decisions. We thus propose to perform two additional tests on values which are of interest for the computation of CIs, and are related to the orthogonal projection matrix p on the range of z:

$$p = z\,\left(z^T z\right)^{-1} z^T \qquad (21)$$

If z is of full rank, since p is a projection matrix, we should have:

$$trace\,(p) = \sum_{k=1}^{N} p_{kk} = rank\,(p) = q \qquad (22)$$

We should also have [Antoniadis et al. 92]:

$$\frac{1}{N} \le p_{kk} \le 1 \quad k=1\ to\ N \qquad (23)$$

It is of importance for our purpose that both (22) and (23) be verified since for an input $\boldsymbol{x}^k$ belonging to the training set, the expression of the CI (18) is precisely:

$$f\left(\boldsymbol{x}^k, \boldsymbol{\theta}_{LS}\right) \pm t_{N-q}(\alpha)\ s\ \sqrt{p_{kk}} \qquad (24)$$

If (22) or (23) are not satisfied, even if the network has not been diagnosed to overfit by a test on the singular values, the computation of the CIs is meaningless.

III.2. Quality of the confidence intervals

The quality of the selected model $f\,(\boldsymbol{x}, \boldsymbol{\theta}_{LS})$, and thus of the associated LTE CIs, depend essentially on the size N of the data set with respect to the complexity of the unknown regression function and to the noise variance σ^2:

1. N is large: it is likely that the selected family $\left\{f\,(\boldsymbol{x}, \boldsymbol{\theta}),\ \boldsymbol{\theta} \in \mathbb{R}^q\right\}$ contains $E\left(Y_p \mid \boldsymbol{x}\right)$, i.e. that the LS estimator is asymptotically unbiased, and that the model $f\,(\boldsymbol{x}, \boldsymbol{\theta}_{LS})$ is a good approximation of $E\left(Y_p \mid \boldsymbol{x}\right)$ in the domain of interest. In this case, reliable CIs can be computed with (18).
2. N is small: it is likely that either the selected family $\left\{f\,(\boldsymbol{x}, \boldsymbol{\theta}),\ \boldsymbol{\theta} \in \mathbb{R}^q\right\}$ is too small to contain $E\left(Y_p \mid \boldsymbol{x}\right)$ or that the selected model $f\,(\boldsymbol{x}, \boldsymbol{\theta}_{LS})$ overfits; in both cases, the CIs (18) are meaningless.

A very large validation mean square error (VMSE) as compared to the training mean square error (TMSE) is usually the symptom of situation 2.

III.3. Industrial application

The goal is to model a mechanical property of a complex material from structural descriptors. We have been provided with a data set of N=69 examples. Thanks to repetitions in the data, and assuming homoscedasticity, the following estimation of the noise variance could be made: $\hat{\sigma}^2$=3.38 10^{-2}, i.e. $\hat{\sigma}$=1.84 10^{-1}. Using this estimate, statistical tests established the significance of two inputs. The TMSE of a linear model with these inputs is 2.28 10^{-1} (i.e. s^2=2.38 10^{-1} and s=4.88 10^{-1}), hence the necessity of nonlinear modeling.

III.3.1. *Selection of the neural model architecture*

Neural network candidates with one hidden layer of sigmoidal hidden neurons of increasing size and a linear output neuron with direct connections from the inputs were trained with the Levenberg algorithm; several initializations of the weights were performed for each neural network, the final weights corresponding to the lowest TMSE being kept. For more than 4 neurons, overfitting was systematically detected through the tests proposed in III.1. The training data and the output of a model with 5 hidden neurons (q=23) are shown in Figure 3; the location of the inputs appears clearly on Figure 6.

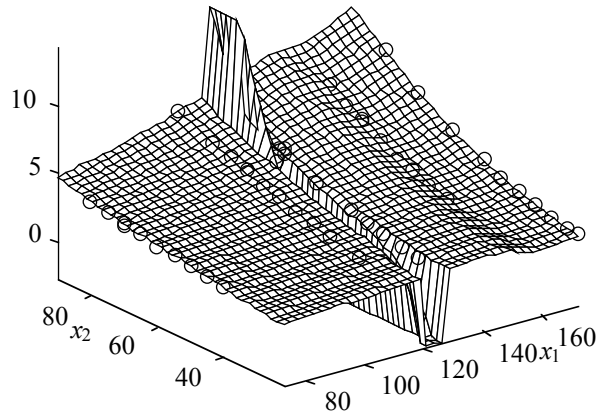

Figure 3. Examples of the data set (circles) and output of an overfitting neural model with 5 hidden units.

The network significantly overfits in a delimited input region containing no examples. The conditioning number equals 1.2 10^{-11}, a value which is not extremely small for a double precision computer. Nevertheless, the overfitting can be detected through the trace of p

which is larger than the expected value 23 (*trace* (*p*)=23.1), and through two diagonal elements which are larger than 1 ($p_{35\,35}$=1.002 and $p_{69\,69}$=1.02). Using statistical Fisher tests for the networks with up to 4 hidden neurons, a network with 2 hidden neurons is selected. Its TMSE equals 1.62 10^{-2} (s=1.39 10^{-1}). We also checked the leave-one-out VMSE of this network: it equals 2.50 10^{-2}, which is of the order of the TMSE. It is thus likely that we are in situation 1 according to the classification of section III.2, i.e. the selected model is large enough to contain the regression. The output of the selected network is shown on Figure 4.

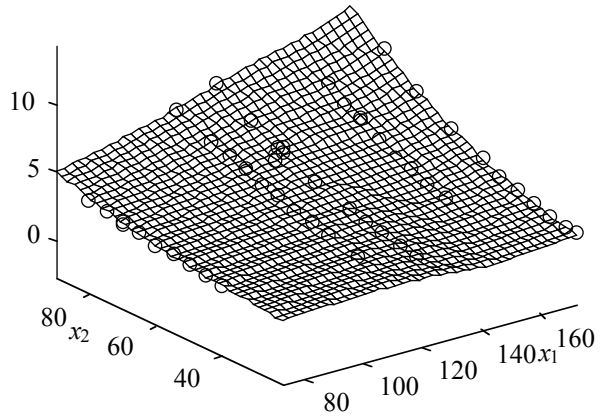

Figure 4. Examples of the data set (circles) and output of the selected neural model with 2 hidden units.

The output of the selected network is roughly the same as that of the overfitting one, except in the overfitted region, where its output is smooth.

III.3.2. *Construction of a confidence interval*

A CI with a level of significance of 95% is then computed using (18). The model error and the width of the 95% CI on the 69 examples of the data set (i.e. $\pm\, t_{58}(0.05)\, 0.139\, \sqrt{p_{kk}}$ for example k) are shown on Figure 5.

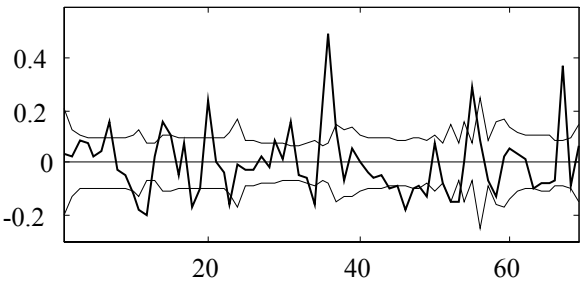

Figure 5. Model error (thick line) and width of the 95% CI (thin lines) on the data set.

In order to check the confidence which can be attached to the model, the variance of its output must be examined on the whole input domain of interest. Figure 6 shows the isocontours of the LTE estimate of the standard deviation of the model output on the input domain defined by the training set. The highest isocontour of Figure 6 corresponds to the estimate of the noise standard deviation s=1.39 10^{-1} obtained with the selected neural model. In two areas of the input domain (bottom left and top right), the variance of the model is higher than that of the noise itself, and no confidence can be attached to the model output in those areas.

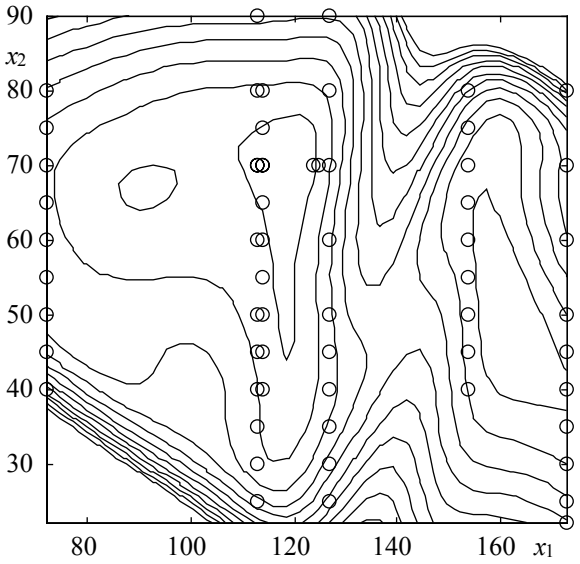

Figure 6. Isocontours of the LTE estimate of the standard deviation of the model output.

IV. DISCUSSION

So far we have established that the LTE approach to the computation of CIs is interesting because i) the LTE helps for the selection of a nonlinear model by detecting overfitting; ii) it is computationally inexpensive; iii) it is accurate if the curvature is small, as shown on the illustrative example of section 2. Point (i) is original, point (ii) and (iii) allow a comparison to other approaches to the construction of CIs.

IV.1. Comparison to bootstrap approaches

The LTE approach is computationally advantageous with respect to multi-training approaches like bootstrap methods. As a matter of fact, the bootstrap works by creating many pseudo replicates of the data set, the bootstrap sets, and reestimating the LS solution (retraining the neural network) on each bootstrap set; the variance of the neural model output, and the associated CI, are then computed over the trained networks, typically *a hundred* [Efron & Tibshirani 93]. In the "bootstrap pairs approach" for example, a bootstrap set is created by sampling with replacement from the data set. The first advantage of the LTE approach is to require only *one* successful training of the network (i.e. a global minimum is reached) on the data set to compute the LTE estimate of the variance of its output, whereas the bootstrap methods require *a hundred* successful trainings of the network on the different bootstrap sets. In fact, the bootstrap is especially suited to the estimation of the variance of estimators defined by a formula, like for example an estimator of a correlation coefficient [Efron & Tibshirani 93]: for each bootstrap set, an estimate is computed using the formula, and the estimate of the

variance is easily obtained. But the bootstrap is definitely not the best method if each estimation associated to a bootstrap set involves a nonlinear optimization like the training of a neural network, which is the case for the construction of CIs for a neural model.

However, if the data set is large enough, and if the training time is considered unimportant, the bootstrap pairs approach is a solution in the case of heteroscedasticity (i.e. $K(\boldsymbol{W})$ is diagonal but not scalar), whereas the LTE approach, as well as the "bootstrap residuals" approach [Efron & Tibshirani 93], are no longer valid.

IV.2. Comparison to second-order analytic approaches

Likelihood theory [Efron & Tibshirani 93], or a bayesian approach [Bishop 95], lead to an analytic expression of the variance of $f(\boldsymbol{x}, \boldsymbol{\Theta}_{LS})$ which contains the Hessian of the cost-function (4). As opposed to them, the LTE variance (10) brings in an approximation of the Hessian through $\xi^T\xi$: one could thus suspect the LTE approach described in this paper to be less accurate than the above second-order methods. But this argument does not hold since, in the same way that $\xi^T\xi$ must be replaced with $z^T z$, the components of the Hessian must be computed around $\boldsymbol{\theta}_{LS}$, instead of the unknown $\boldsymbol{\theta}_p$. In this context, trying to build second-order CIs is of little interest. Moreover, for many applications, the curvature of the solution surface is such that a first-order expansion is satisfactory.

V. CONCLUSION

We have given an original presentation and analysis of the LTE approach to the construction of CIs for a nonlinear regression using neural network models. We have stressed the fact that these CIs are meaningful only if the selection procedure has led to a good model. We have thus introduced LTE based tests to detect overfitting, precisely in order to perform a good model selection.

We have shown that, as opposed to the computationally intensive bootstrap methods, the LTE approach to the estimation of CIs is very economical in terms of computer power, since it only involves a few final backpropagation runs.

We have also shown that second-order analytic methods, which necessitate the computation of the Hessian, are of little interest with respect to the LTE approach. As a matter of fact, for many applications, the data set is large enough for a first-order expansion to be valid; on the other hand, when it is too small, the approximations needed because the true parameters are unknown make a second-order expansion meaningless.

Abbreviations

CI	confidence interval
LS	least squares
LTE	linear Taylor expansion
TMSE	training mean square error
VMSE	validation mean square error

REFERENCES

Antoniadis A., Berruyer J., & Carmona R. (1992). Régression non linéaire et applications. Paris: Economica.

Bates D. M., & Watts D. G. (1988). Nonlinear regression analysis and its applications. New York: Wiley.

Bishop M. (1995). Neural Networks for Pattern Recognition. Oxford: Clarendon Press.

Efron B., & Tibshirani R. J. (1993). An introduction to the bootstrap. New York: Chapman & Hall.

Nash J. C. (1990). Compact numerical methods for computers : linear algebra and function minimisation. New York: Adam Hilger.

Paass G. (1993). Assessing and improving neural network predictions by the bootstrap algorithm. In S. J. Hanson, J. D. Cowan, & C. L. Giles (Eds.), *Advances in Neural Information Processing Systems* **5** (pp. 186-203). Cambridge, MA: MIT Press.

Seber G. A. F., & Wild C. (1989). Nonlinear regression. New York: Wiley.

Sussman H. J. (1992). Uniqueness of the weights for minimal feedforward nets with a given input-output map. *Neural Networks* **5**, 589-593.

Tibshirani R. J. (1996). A comparison of some error estimates for neural models. *Neural Computation* **8**, 152-163.

Urbani D., Roussel-Ragot P., Personnaz L. & Dreyfus G. (1994). The selection of neural models of non-linear dynamical systems by statistical tests, Neural Networks for Signal Processing, Proceedings of the 1994 IEEE Workshop, pp. 229-237.