Rivals I. & Personnaz L. "A statistical procedure for determining the optimal number of hidden neurons of a neural model", Second International Symposium on Neural Computation (NC'2000), Berlin, May 23-26 2000.

A statistical procedure for determining the optimal number of hidden neurons of a neural model

Isabelle Rivals and Léon Personnaz
École Supérieure de Physique et de Chimie Industrielles (ESPCI)
Équipe de Statistique Appliquée, 10 rue Vauquelin, 75231 Paris Cedex 05, France.
E-mail: Isabelle.Rivals@espci.fr, Léon.Personnaz@espci.fr

Abstract

This paper proposes a novel model selection procedure for neural networks based on least squares estimation and statistical tests. The procedure is performed systematically and automatically in two phases. In the first (bottom-up) phase, the parameters of candidate neural models with an increasing number of hidden neurons are estimated until they cannot be approved anymore, i.e. until the neural models become ill-conditioned. In the second (top-down) phase, a selection among approved candidate models using statistical Fisher tests is performed; the series of tests starts from an appropriate full model chosen with the help of computationally inexpensive estimates of the performance of the candidates, and ends with the smallest candidate whose hidden neurons all have a statistically significant contribution to the estimation of the regression. Large scale simulation experiments illustrate the efficiency and the parsimony of the proposed procedure, and allow a comparison to other approaches.

1. Introduction

We deal with modeling problems for the case of a non random (noise free) n-input vector $\boldsymbol{x} = [x_1\ x_2\ \dots\ x_n]^T$, and a measured scalar output y_p which is considered as the actual value of a random variable[1] $Y_p = Y_p \mid \boldsymbol{x}$ depending on $\boldsymbol{x}$. We assume that there exists an unknown function of $\boldsymbol{x}$, the regression $E(Y_p \mid \boldsymbol{x})$, such that for any fixed value $\boldsymbol{x}^a$ of $\boldsymbol{x}$:

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

where $W \mid \boldsymbol{x}^a$ is thus a random variable with zero expectation[2]. We consider families of parameterized functions of the form $\{f(\boldsymbol{x}, \boldsymbol{\theta}), \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{\theta} \in \mathbb{R}^q\}$. Such a family of functions contains the regression if there exists a value $\boldsymbol{\theta}_p$ of $\boldsymbol{\theta}$ such that $f(\boldsymbol{x}, \boldsymbol{\theta}_p) = E(Y_p \mid \boldsymbol{x})$. In real-world black-box modeling problems, a family of functions containing the regression is not known *a priori*, so that candidate families of various complexities must be put into competition; in this work, we consider neural networks with one layer of nonlinear hidden neurons and a linear output neuron. In order to estimate their parameters, a data set of input-output pairs $\{\boldsymbol{x}^k, y_p^k\}_{k=1 \text{ to } N}$ must be available, where the $\boldsymbol{x}^k = [x_1^k\ x_2^k\ \dots\ x_n^k]^T$ are the imposed inputs, and the y_p^k are the corresponding measurements of the process output. The goal is to select a model approximating the regression as accurately as possible within the input domain delimited by the data set and with a minimal number of hidden neurons, among the candidate models estimated with the data set. The selection procedure we propose is performed in two phases which can be outlined as follows:

a) a bottom-up estimation and approval phase: the parameters of candidate neural models with an increasing number of hidden neurons are estimated until the models, i.e. their Jacobian matrix, become ill-conditioned.

b) a top-down selection phase: a full model, i.e. a model that roughly estimates the regression, is chosen among the approved models according to an estimate of their performance, and Fisher tests are performed in order to

---

[1] We distinguish between random variables and their values (or realizations) 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 $\boldsymbol{x}^a$ and $\{\boldsymbol{x}^k\}$; non random matrices are denoted by light lowercase letters.

[2] In this work, we thus assume that the relevant inputs have already been selected, for instance by building a polynomial model and ordering its regressors according to their relevance with an orthogonalization procedure [Chen & Billings 1989].

establish wether all the hidden neurons of this full model are necessary, i.e. statistically significant.

Section 2 summarizes the practical and statistical framework of least squares estimation. Section 3 introduces the notion of model approval and its practical use. Section 4 presents the Fisher tests for the comparison of nested models, motivates the choice of an appropriate full model to start the tests from, and proposes to base this choice on performance estimates of the approved candidates. Section 5 presents simulated experiments illustrating the economic character of the proposed procedure, as well as its efficiency with respect to other approaches.

2. Least squares estimation

Preliminary to the selection, the parameters of the candidate models must be estimated. A least squares estimate $\boldsymbol{\theta}_{LS}$ of the parameters of a family of functions $\{f(\boldsymbol{x}, \boldsymbol{\theta}), \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{\theta} \in \mathbb{R}^q\}$ minimizes the cost function[3]:

$$J(\boldsymbol{\theta}) = \frac{1}{2} \sum_{k=1}^{N} \left(y_p^k - f\left(\boldsymbol{x}^k, \boldsymbol{\theta}\right)\right)^2 = \frac{1}{2} \left|\left| \boldsymbol{y}_p - \boldsymbol{f}(x, \boldsymbol{\theta}) \right|\right|^2 \quad (2)$$

where $x = \left[\boldsymbol{x}^1\ \boldsymbol{x}^2\ \ldots\ \boldsymbol{x}^N\right]^T$ is the (N, n) input matrix and $\boldsymbol{f}(x, \boldsymbol{\theta}) = \left[f\left(\boldsymbol{x}^1, \boldsymbol{\theta}\right) \ldots f\left(\boldsymbol{x}^N, \boldsymbol{\theta}\right)\right]^T$. The estimate $\boldsymbol{\theta}_{LS}$ is a realization of a least squares estimator $\boldsymbol{\Theta}_{LS}$. Efficient iterative algorithms are available for the minimization of cost function (2), for example the Levenberg-Marquardt algorithm used in this work. It modifies the parameter vector iteratively according to:

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i\text{-}1} + \left(z_{i\text{-}1}^T\, z_{i\text{-}1} + \lambda_i\, I_q\right)^{-1} z_{i\text{-}1}^T \left(\boldsymbol{y}_p - \boldsymbol{f}(x, \boldsymbol{\theta}_{i\text{-}1})\right) \quad (3)$$

where $z_{i\text{-}1}$ denotes the Jacobian matrix available at iteration i:

$$z_{i\text{-}1} = \left.\frac{\partial \boldsymbol{f}(x, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{i\text{-}1}} \quad (4)$$

and where the scalar $\lambda_i > 0$ is suitably chosen (see [Bates & Watts 1988]). All the following results and considerations are valid provided an absolute minimum of the cost function (2) is reached. Thus, in order to have a high probability to obtain such a minimum, several minimizations must be made with different initial conditions, the parameter value corresponding to the lowest minimum being kept.

The Jacobian matrix evaluated at $\boldsymbol{\theta}_{LS}$, simply denoted by z, plays an important role in the statistical properties of least squares estimation. As a matter of fact, if the family of functions contains the regression and if the noise W is homoscedastic with variance σ^2:

a) The covariance matrix of the least squares parameter estimator $\boldsymbol{\Theta}_{LS}$ is asymptotically (i.e. as N tends to infinity) given by $\sigma^2 \left(z^T z\right)^{-1}$.

b) The variance of the least squares estimator $f\left(\boldsymbol{x}^a, \boldsymbol{\Theta}_{LS}\right)$ of the regression for an input $\boldsymbol{x}^a$ is asymptotically given by $\sigma^2 \left(\boldsymbol{z}^a\right)^T \left(z^T z\right)^{-1} \boldsymbol{z}^a$, where $\boldsymbol{z}^a = \left.\frac{\partial f\left(\boldsymbol{x}^a, \boldsymbol{\theta}\right)}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{LS}}$.

c) The vector of residuals $\boldsymbol{R} = \boldsymbol{Y}_p - \boldsymbol{f}(x, \boldsymbol{\Theta}_{LS})$ is uncorrelated with $\boldsymbol{\Theta}_{LS}$ and has the asymptotic property $\frac{\boldsymbol{R}^T \boldsymbol{R}}{\sigma^2} \rightsquigarrow \chi^2(N\text{–}q)$. $S^2 = \frac{\boldsymbol{R}^T \boldsymbol{R}}{N - q}$ is an unbiased estimator of σ^2.

d) Hence an estimate of the $(1\text{-}\alpha)\%$ confidence interval for the regression for any input $\boldsymbol{x}^a$ of interest $f\left(\boldsymbol{x}^a, \boldsymbol{\theta}_{LS}\right) \pm g\left(1\text{–}\alpha/2\right) s \sqrt{\left(\boldsymbol{z}^a\right)^T \left(z^T z\right)^{-1} \boldsymbol{z}^a}$, where g is the inverse of the gaussian cumulative distribution.

Many approaches encountered in the neural literature perform the parameter estimation by minimizing a regularized cost function, or by stopping the training during the minimization of cost function (2) "early", i.e. before its minimum is reached, considering the mean square error on a independent set. Unfortunately, the regularization terms are often arbitrarily chosen, and early stopping lacks theoretical support (see [Anders & Korn 1999] for a discussion of this issue). We therefore recommend the classic least squares approach, which allows a sound statistical approach of the selection problem, the estimation of confidence intervals, etc.

3. Model approval

Selection is the process by which the best model is chosen; approval is a preliminary step to selection

[3] For a multilayer neural network, due to symmetries in its architecture (function-preserving transformations are neuron exchanges, as well as sign flips for odd activation functions like the hyperbolic tangent), the minimal value of the cost function can be obtained for several values of the parameter vector; but as long as an optimal parameter is unique in a small neighborhood, the following results remain unaffected.

which consists in rejecting unusable candidates. As a matter of fact, a candidate model may be unusable for the following reasons:

a) it overfits, i.e. it is unnecessarily complex given the data set;

b) it is numerically so ill-conditioned that relevant quantities such as confidence intervals (see [Rivals & Personnaz, to appear]) or approximate leave-one-out scores (see section 4.1) cannot be reliably computed;

c) some of its parameters are too large, i.e. may require too large a precision.

Unusable models in the above sense cannot be approved[4]. Luckily, these three features of a neural model are highly linked, and can be characterized with the conditioning of its Jacobian matrix z. Since the elements of z represent the sensibility of the model output with respect to the parameters, the ill-conditioning of z is naturally the symptom that some parameters are useless, i.e. that the model is too complex. A typical situation is the saturation of a "tanh" hidden neuron, a situation which generates in the matrix z a column of +1 or –1 that corresponds to the parameter between the output of the saturated hidden neuron and the linear output neuron, and columns of zeros that correspond to the parameters between the network inputs and the saturated hidden neuron[5] (see [Rivals & Personnaz 1998]). Further, the computation of confidence intervals involves that of the inverse of $z^T z$, thus necessitating that z itself be well-conditioned. This will also be necessary to compute approximate leave-one-out scores (see section 4.1). Finally, it is also clear that large parameters usually lead to the ill-conditioning of z, large values of parameters between the inputs and a hidden neuron driving this neuron into saturation.

In practice, we propose to perform a singular value factorization of z, and to compute its condition number $\kappa(z)$, that is the ratio of its largest to its smallest singular value, see for example [Golub & Van Loan 1983]. The matrix z can be considered as very ill-conditioned when $\kappa(z)$ reaches the inverse of the computer precision, which is of the order of 10^{-16}. Since $\kappa(z^T z) = (\kappa(z))^2$, only the neural candidates whose condition number is not much larger than 10^8 will be approved[6]. Usually, $\kappa(z)$ increases regularly with the number of hidden neurons, except for very small networks in the case of small and noisy data sets, where their neurons tend to saturate.

4. Model selection

Neural networks with one layer of hidden neurons and a linear output neuron are relatively easy to compare because they are nested models. We briefly recall the Fisher tests to select the simplest nested model needed to estimate the regression adequately, given the data set (see for example [Bates & Watts 1988]). An important issue is how to choose the most complex model where to start the tests from, the "full model", since the most complex approved model may give a bad estimate of the noise variance. We propose an economic choice of the full model based on approximate leave-one-out scores of the approved candidate models.

4.1. Choice of a full model

The full model must satisfy two conditions:

a) according to the assumptions needed to perform statistical tests, the family of functions defined by its architecture should be complex enough to contain a good approximation of the regression in the input domain delimited by the data set: there should be no "lack of fit" of the regression;

b) the full model should not be too complex, in order to

---

4 The term "validation" would also be appropriate, but it is unfortunately too tightly connected to cross-validation, and thus to performance estimation.

5 Such a situation might also correspond to a relative minimum; to check the conditioning of z thus also helps to discard neural networks trapped in relative minima, and leads to retrain the neural candidate with different initial conditions.

6 Previous studies of the ill-conditioning of neural networks deal with their training rather than with their approval, like in [Zhou & Si 1998] where an algorithm avoiding the Jacobian rank deficiency is presented, or in [Saarinen et al. 1993] where the Hessian rank deficiency is studied during training. In our view, rank deficiency is not relevant *during the training* since, with a Levenberg algorithm, the matrix to be "inverted" is made well-conditioned by the addition of a scalar matrix λI_q to the cross-product Jacobian $z^T z$.

avoid a bad estimation of the variance through $\frac{\boldsymbol{r}_q{}^T \boldsymbol{r}_q}{N-q}$, where $\boldsymbol{r}_q$ denotes the residuals of a model with q parameters.

In order to judge if these conditions are fulfilled, a performance estimate of the approved candidate is needed. We propose to use their leave-one-out scores, more precisely an economic approximation of them.

In the case of a linear model, the k-th leave-one-out error e^k can be directly derived from the corresponding residual r^k [Antoniadis et al. 1992] [Efron & Tibshirani 1993]:

$$e^k = \frac{r^k}{1 - [p_x]_{kk}} \quad k=1 \text{ to } N \tag{5}$$

where $p_x = x\left(x^T x\right)^{-1} x^T$ denotes the (N, N) orthogonal projection matrix on the range of the (N, n) input matrix x, and assuming that $[p_x]_{kk} < 1$. In the case of a nonlinear model, we have shown [Rivals & Personnaz, in press] that a useful approximation of the k-th leave-one-out error is:

$$e^k \approx \frac{r^k}{1 - [p_z]_{kk}} \quad k=1 \text{ to } N \tag{6}$$

where $p_z = z\left(z^T z\right)^{-1} z^T$ denotes the (N, N) orthogonal projection matrix on the range of the (N, q) Jacobian matrix, and assuming that $[p_z]_{kk} < 1$. Hence the approximate leave-one-out score:

$$ALOOS = \frac{1}{N} \sum_{k=1}^{N} \left(e^k\right)^2 \tag{7}$$

The *ALOOS* can be reliably computed only if the model is well-conditioned; if not, some of the diagonal terms $[p_z]_{kk}$ of the projection matrix may be outside their theoretical bounds $[1/N; 1]$. Another performance measure could be chosen (a 10-fold cross validation score, a mean square error on an independent set, etc.): the advantage of the *ALOOS* (7) is its economic computation, on the whole data set.

In [Rivals & Personnaz 1999], we have shown that the leave-one-out score alone is often not sufficient to perform a good selection. Here, we propose to use the *ALOOS* as a tool to perform the tests appropriately, i.e. to choose the full model as the most complex approved model before the *ALOOS* starts to increase significantly. The full model can then be considered as a good approximation of the regression if the ratio of its *ALOOS* to its mean square training error $\frac{2}{N} J(\boldsymbol{\theta}_{LS})$ (*MSTE*) is of the order of one[7].

4.2. Statistical tests

Let us suppose that the family of functions defined by the architecture of a given model with q parameters contains the regression; we call this model the unrestricted one. We are interested in deciding whether the family of functions defined by the architecture of a restricted model, i.e. a submodel with $q' < q$ parameter, also contains the regression. This decision problem leads to define the null hypothesis H_0, i.e. the hypothesis that the family of functions defined by the architecture of the restricted model contains the regression, and to build a statistical Fisher test. When H_0 is true, the following ratio b is the value of a random variable approximately Fisher distributed, with $q - q'$ and $N - q$ degrees of freedom:

$$b = \frac{\dfrac{\boldsymbol{r}_{q'}{}^T \boldsymbol{r}_{q'} - \boldsymbol{r}_q{}^T \boldsymbol{r}_q}{q - q'}}{\dfrac{\boldsymbol{r}_q{}^T \boldsymbol{r}_q}{N - q}} \tag{8}$$

where $\boldsymbol{r}_q$ and $\boldsymbol{r}_{q'}$ denote the residuals of the unrestricted model (with q parameters) and those of the restricted one (with q' parameters)[8]. The decision to reject H_0 with a risk α% of rejecting it while it is true will be taken when $b > f_{N-q}^{q-q'}(1-\alpha)$, where $f_{N-q}^{q-q'}$ is the inverse of the Fisher cumulative distribution. When $b \leq f_{N-q}^{q-q'}(1-\alpha)$, nothing in the data set allows to say that the family of functions defined by the architecture of the restricted model does not contain the regression.

In practice, a sequence of tests is performed starting with the full model as unrestricted model, the restricted model being then taken as new unrestricted model ($q_{new} = q'_{old}$) as long as the null hypothesis is not rejected. It is naturally interesting to use these tests to

---

7 When the data set includes replications, it advantageous to perform a test for lack of fit [Seber & Wild 1989].

8 Generally, $N-q$ is large (>100), so that the ratio:

$$b' = \frac{\boldsymbol{r}_{q'}{}^T \boldsymbol{r}_{q'} - \boldsymbol{r}_q{}^T \boldsymbol{r}_q}{\boldsymbol{r}_q{}^T \boldsymbol{r}_q}(N-q) = b\,(q-q')$$

is the value of a random variable approximately $\chi^2(N-q)$ distributed.

decide whether a restricted network with one or several hidden neurons less than the full model still gives a good approximation of the regression.

Each of these Fisher tests thus involves the estimation of the parameters of both the unrestricted model and the restricted one. It is also possible to perform tests (the Wald tests [Anders & Korn 1999] or more generally the so-called tests of "linear hypotheses" [Seber & Wild 1989]) involving only the unrestricted model, by testing whether some of its parameters are zero or not. Whereas in the case of linear models, the two tests are equivalent, they are not in the case of neural networks due to the interchangeability of their hidden neurons.

5. Illustrative examples

We consider three different processes. Process 1 is a single input process simulated with:

$$E\left(Y_p|\ x\right) = sinc\ \left(10\ (x+1)\right) \qquad (9)$$

where "sinc" denotes the cardinal sine function, the input values are drawn from an uniform distribution in [–1; 1], and the noise values from a gaussian distribution with $K\left(\boldsymbol{W}\right) = 2\ 10^{-3}\ I_N$.

Processes 2 and 3 are taken from [Anders & Korn 1999] in order to allow numerical comparisons of our method to those described in this very interesting paper. Process 2 is a three input process simulated with:

$$E\left(Y_p|\ \boldsymbol{x}\right) = 1 + tanh\left(2\,x_1 - x_2 + 3\,x_3\right) + tanh\left(x_2 - x_1\right) \qquad (10)$$

and process 3 is a two input process simulated with:

$$E\left(Y_p|\ \boldsymbol{x}\right) = -0.5 + 0.2\ \ (x_1)^2 - 0.1\ \ exp\,(x_2) \qquad (11)$$

whose input values are drawn from a gaussian distribution with unit variance, and the noise values from a gaussian distribution whose standard deviation equals 20% of the unconditional standard deviation of the output, i.e. $K\left(\boldsymbol{W}\right) = 0.2\ I_N$ for process 2 and $K\left(\boldsymbol{W}\right) = 5\ 10^{-3}\ I_N$ for process 3.

We build 1000 data sets of size N with different values of the noise (the inputs remaining unchanged); a large separate set of 500 samples is used for performance estimation. The neural models are trained with the Levenberg-Marquardt algorithm, each of them being trained q (the number of its parameters) times starting from different random initial parameters values, in order to maximize the probability to reach a global minimum.

The bottom-up estimation and approval phase is stopped when either the condition number of z reaches 10^8, or one of the diagonal terms $[p_z]_{kk}$ of the projection matrix needed for the computation of the *ALOOS* is outside the theoretical bounds [1/N; 1]. The full model is chosen as the largest approved model before the *ALOOS* starts to increase, and the top-down test phase is performed at the 5% risk level. In each case, i.e. a given process and a fixed data set size N, we give the average values on the 1000 data sets of:

a) the number of hidden units of the full model h_{full};

b) the number of hidden units of the selected model h_{sel};

c) the mean square training error *MSTE* of the selected model;

d) the approximate leave-one-out score *ALOOS* of the selected model;

e) the ratio of the *ALOOS* of the selected model to its *MSTE*, which we expect to indicate whether the selected model gives a good estimate of the regression:

$$\pi = \frac{ALOOS}{MSTE} \qquad (12)$$

f) the mean square performance error *MSPE*, i.e. the mean square error obtained on the large separate set;

g) the relative difference in *MSPE* between the true model, i.e. the model whose output is the regression, and the selected model:

$$\rho = \frac{MSPE - MSPE_{true\ model}}{MSPE_{true\ model}} \qquad (13)$$

This ratio[9] is used in order to perform comparisons with the results of [Anders & Korn 1999].

The results obtained for the modeling of process 1 are shown in Table 1. For large N, the procedure selects an average number of hidden neurons $\overline{h_{sel}}$ of 4.1 with an average *MSPE* ($\overline{MSPE}$) equal to the noise variance, performance which indicates that this selection is adequate. Both $\overline{MSPE}$ and $\overline{h_{sel}}$ remain almost constant until N becomes too small with respect to the number of parameters needed for the 4 hidden units network (q = 13): $\overline{h_{sel}}$ then drops to 3.1 for N = 50. Nevertheless,

9 Note that ρ carries the same information as the *MSPE* together with the noise variance σ^2. Note also that, when ρ = 100%, the *MSPE* is only twice as large as σ^2, a value which is quite satisfactory in the case of a small data set and a complex regression.

the performance is not bad ($\overline{MSPE}$ equals twice the noise variance), and the value of the ratio π, on the average equal to 1.63, helps the designer in individual cases to diagnose that the data set size N is too small to achieve a perfect modeling. Finally, we note that, whatever N, the full model is frequently selected ($\overline{h_{full}} \approx \overline{h_{sel}}$).

N	50	100	200	500
$\overline{h_{full}}$	3.2	4.2	4.3	4.3
$\overline{h_{sel}}$	3.1	4.1	4.2	4.2
$\overline{MSTE}$	$1.94\ 10^{-3}$	$1.77\ 10^{-3}$	$1.86\ 10^{-3}$	$1.94\ 10^{-3}$
$\overline{ALOOS}$	$3.08\ 10^{-3}$	$2.39\ 10^{-3}$	$2.15\ 10^{-3}$	$2.05\ 10^{-3}$
$\overline{\pi}$	1.64	1.35	1.15	1.06
$\overline{MSPE}$	$4.40\ 10^{-3}$	$2.56\ 10^{-3}$	$2.28\ 10^{-3}$	$2.14\ 10^{-3}$
$\overline{\rho}$	108.4%	15.9%	8.2%	3.1%

Table 1. Results[10] for process 1 ($\sigma^2 = 2\ 10^{-3}$).

The results obtained for the modeling of process 2 are shown in Table 2. The family of the neural networks with two hidden neurons contains the regression. It is interesting to note that $\overline{h_{sel}}$ almost does not vary with the data set size N: the model with the right architecture being almost always chosen. Like for process 1, we note that the preselection performed on the basis of the conditioning number of z and of the *ALOOS* almost always leads to the right model. Finally, the fact that a parameter from the input x_3 to one of the hidden neurons is not necessary does not lead to the ill-conditioning of 2 hidden units models with all their connections. The results obtained on this example can be compared to those presented in [Anders & Korn 1999], which studies how hypothesis tests, information criteria and cross-validation can guide model selection; it concludes that approaches based on statistical tests lead to the best results. These approaches proceed by constructing models with an increasing number of hidden neurons, until an additional one is shown to be statistically not significant; Wald tests are then performed on the input connections. The best method described in [Anders & Korn 1999] achieves a performance which is similar to ours ($\overline{\rho} = 3.3\%$), with roughly the same average number of hidden units of the selected model, in the case $N = 500$ (smaller training sets are not considered in their paper).

N	50	100	200	500
$\overline{h_{full}}$	2.1	2.2	2.2	2.2
$\overline{h_{sel}}$	2.0	2.1	2.1	2.2
$\overline{MSTE}$	$1.59\ 10^{-1}$	$1.73\ 10^{-1}$	$1.86\ 10^{-1}$	$1.94\ 10^{-1}$
$\overline{ALOOS}$	$2.69\ 10^{-1}$	$2.23\ 10^{-1}$	$2.10\ 10^{-1}$	$2.03\ 10^{-1}$
$\overline{\pi}$	1.75	1.30	1.10	1.05
$\overline{MSPE}$	$2.89\ 10^{-1}$	$2.23\ 10^{-1}$	$2.01\ 10^{-1}$	$1.92\ 10^{-1}$
$\overline{\rho}$	44.2%	18.3%	8.4%	3.1%

Table 2. Results for process 2 ($\sigma^2 = 2\ 10^{-1}$).

The results obtained for the modeling of process 3 are shown in Table 3. The regression being quite complex (see the regression surface in [Anders & Korn 1999]), the performance deteriorates when the size of the training set becomes too small. Again, the average numbers of hidden neurons of the full model and of the selected model increase as the data set size N increases, and the tests slightly reduce the size of the selected model as compared to that of the full model. In the case $N = 500$, the best statistical selection method described in [Anders & Korn 1999] obtains a poor performance ($\overline{\rho} = 30.9\%$) as compared to ours ($\overline{\rho} = 2.5\%$), but with roughly the same average number of hidden neurons (3.2). This difference may be due to the fact that their training algorithm is less efficient, or that their tests have lead to the suppression of too many connections from the inputs to the hidden neurons: on the average, 7.3 such connections remain in their selected network, whereas a completely connected 3 hidden units model possesses 9, and a 4 hidden units model possesses 11. They also report that 10-fold cross-validation achieves $\overline{\rho} = 53.6\%$ with 3.7 hidden units on the average (again with $N = 500$).

[10] In real life, the values of the *MSPE* and of ρ are not available, hence the gray background used for them in the result tables.

N	100	200	500
$\overline{h_{full}}$	3.2	3.3	3.4
$\overline{h_{sel}}$	3.1	3.2	3.3
$\overline{MSTE}$	4.48 10^{-3}	4.64 10^{-3}	4.82 10^{-3}
$\overline{ALOOS}$	7.20 10^{-3}	5.74 10^{-3}	4.15 10^{-3}
$\overline{\pi}$	1.6	1.2	1.1
$\overline{MSPE}$	1.38 10^{-2}	5.79 10^{-3}	4.41 10^{-3}
$\overline{\rho}$	178%	11.0%	2.5%

Table 3. Results for Process 3 ($\sigma^2 = 5\ 10^{-3}$).

We can now make two important statements:

a) the full model is always already quite small, i.e. the approval based on the conditioning of z together with the rule involving the $ALOOS$ lead to a good preliminary selection; unnecessarily complex models are not taken into account, as opposed to pruning approaches, thus sparing a lot of computation time.

b) the ratio $\pi = ALOOS/MSTE$ gives indeed a reliable indication of whether the selected model gives a good estimate of the regression, and thus if it makes sense to compute confidence intervals on its basis.

6. Conclusion

We have presented an economic and efficient automatic procedure for determining the optimal number of hidden neurons of a neural model, procedure which is performed in two phases. The first one, based on a bottom-up strategy, leads to the choice of a model, the full model, which is not too complex in the sense that it already gives a correct approximation of the regression and that its Jacobian matrix is sufficiently well-conditioned. The second phase, based on a top-down strategy, uses statistical Fisher tests to further refine the selection, that is to further reduce the complexity of the model. The whole procedure is economic since it only necessitates the computation of the condition number and of the approximate leave-one-out scores of the neural candidates in the bottom-up phase, and that of the Fisher ratios in the top-down phase. Finally, our procedure proves to be efficient as compared to many other approaches proposed in the literature.

References

Anders U., Korn O. (1999) "Model selection in neural networks", Neural Networks Vol. 12, 309-323.

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.

Chen S., Billings A., Luo W. (1989) "Orthogonal least-squares methods and their application to non-linear system identification", Int. J. of Control, Vol. 50, N°5, 1873-1896.

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

Golub G. H., Van Loan C. F. (1983) Matrix computations. Baltimore: The John Hopkins University Press.

Rivals I., Personnaz L. (1998) "Construction of confidence intervals in neural modeling using a linear Taylor expansion" *Proceedings of the International Workshop on Advanced Black-Box Techniques for Nonlinear Modeling*, 8-10 July 1998, Leuwen, 17-22.

Rivals I., Personnaz L. (1999) "On cross-validation for model selection", Neural Computation Vol. 11 N°4, 863-870.

Rivals I., Personnaz L. "Construction of confidence intervals for neural networks based on least squares estimation", Neural Networks, in press.

Saarinen S., Bramley R., Cybenko G. (1993) "Ill-conditioning in neural network training problems", *SIAM J. Sci. Stat. Comp.* **14**, 693-714.

Seber G.A.F., Wild C. J. (1989) *Nonlinear regression*, John Wiley & Sons.

Zhou G., Si J. (1998) "A systematic and effective supervised learning mechanism based on Jacobian rank deficiency", *Neural Computation* **10**, 1031-1045.