Neural Computation 16: 401-418 (2004).

Jacobian conditioning analysis for model validation

Isabelle Rivals and Léon Personnaz
Équipe de Statistique Appliquée
École Supérieure de Physique et de Chimie Industrielles
10, rue Vauquelin - F75005 Paris, FRANCE
E-mail: Isabelle.Rivals@espci.fr, Leon.Personnaz@espci.fr

Abstract

Our aim is to stress the importance of Jacobian matrix conditioning for model validation. We also comment (Monari and Dreyfus, 2002) where, following (Rivals and Personnaz, 2000), it is proposed to discard neural candidates which are likely to overfit, and/or for which quantities of interest such as confidence intervals cannot be computed accurately. In (Rivals and Personnaz, 2000), we argued that such models are to be discarded on the basis of the condition number of their Jacobian matrix. But (Monari and Dreyfus, 2002) suggest to take the decision on the basis of the computed values of the leverages, the diagonal elements of the projection matrix on the range of the Jacobian, or "hat" matrix: they propose to discard a model if computed leverages are outside some theoretical bounds, pretending that it is the symptom of the Jacobian rank deficiency.

We question this proposition because, *theoretically*, the hat matrix is defined whatever the rank of the Jacobian, and because, *in practice*, the computed leverages of very ill-conditioned networks may respect their theoretical bounds while confidence intervals cannot be estimated accurately enough, two facts that have escaped Monari and Dreyfus's attention. We also recall the most accurate way to estimate the leverages and the properties of these estimations.

Finally, we make an additional comment concerning the performance estimation in (Monari and Dreyfus, 2002).

Key-Words

Condition number, confidence interval, hat matrix, Jacobian matrix, leave-one-out cross-validation, leverages computation, model validation, neural network, overfitting, QR decomposition, singular value decomposition.

1. On the Jacobian matrix of a nonlinear model

We deal with the modeling of processes having a certain n-input vector $\boldsymbol{x} = [x_1\ x_2\ \dots\ x_n]^T$ and a measured scalar output y. We assume that, for any fixed value $\boldsymbol{x}^a$ of $\boldsymbol{x}$:

$$y^a = \mu(\boldsymbol{x}^a) + w^a \tag{1}$$

where μ is the regression function, and w^a is a random variable with zero expectation. A training set $\{\boldsymbol{x}^k, y^k\}_{k=1 \text{ to } N}$ is available. The goal is to validate a set of candidate models approximating the regression as accurately as possible, and well conditioned enough for a meaningful estimation of a confidence interval.

1.1. Theoretical results

We consider a family of parameterized functions $\mathcal{F} = \{f(\boldsymbol{x}, \boldsymbol{\theta}), \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{\theta} \in \mathbb{R}^q\}$ implemented by a neural network. A least squares estimate $\boldsymbol{\theta}_{LS}$ of the parameters minimizes the quadratic cost function:

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

The Jacobian matrix Z of the model with parameter $\boldsymbol{\theta}_{LS}$ plays an important role in the statistical properties of least squares estimation. It is defined as the (N,q) matrix with elements:

$$z_{ki} = \left. \frac{\partial f(\boldsymbol{x}^k, \boldsymbol{\theta})}{\partial \theta_i} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{LS}} \tag{3}$$

If $\mathcal{F}$ contains the regression, if Z is full rank, and if the noise w is homoscedastic with variance λ, an estimate of the $(1-\alpha)$% confidence interval for the regression for any input $\boldsymbol{x}^a$ is *asymptotically* given by:

$$f(\boldsymbol{x}^a, \boldsymbol{\theta}_{LS}) \pm g(\alpha)\ s\ \sqrt{(\boldsymbol{z}^a)^T \left(Z^T Z\right)^{-1} \boldsymbol{z}^a} \tag{4}$$

with $\boldsymbol{z}^a = \partial f(\boldsymbol{x}^a, \boldsymbol{\theta}) / \partial \boldsymbol{\theta} \,|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{LS}}$, where the function g denotes the inverse of the gaussian cumulative distribution, and where $s^2 = \sum_{k=1}^{N} (e^k)^2/(N-q)$ with $e^k = y^k - f(\boldsymbol{x}^k, \boldsymbol{\theta}_{LS})$ denoting the k-th residual.

In (Rivals and Personnaz, 2000), we further showed that, whatever $\mathcal{F}$, if Z is full rank, the k-th leave-one-out error $e^k_{(k)}$ can be approximated with:

$$e^k_{(k)} \approx \frac{e^k}{1 - h_{kk}} \quad \text{for } k=1 \text{ to } N \tag{5}$$

where the leverages $\{h_{kk}\}$ are the diagonal elements of the orthogonal projection matrix on the range of Z, the "hat" matrix $H = Z\,Z^I$. The matrix Z being full rank, we have $H = Z\,(Z^T Z)^{-1}\,Z^T$.

Like any orthogonal projection matrix, its diagonal elements verify:

$$0 \le h_{kk} \le 1 \quad \text{for } k=1 \text{ to } N \tag{6}$$

For a linear as well as for a non linear model, a leverage value measures the

influence of the corresponding example on the parameter estimate $\boldsymbol{\theta}_{LS}$. In practice, a leverage value larger than 0.5 is considered to indicate a very influent example. For a model linear in its parameters, relation (5) is an equality. The case $h_{kk} = 1$ implies $e^k = 0$, and $e^k_{(k)}$ is not determined. But conversely, $e^k = 0$ does not imply $h_{kk} = 1$, and a small residual does not necessarily correspond to a leverage close to one. For a non linear model, (5) being only an approximation, $h_{kk} = 1$ does not imply that the corresponding residual is zero, nor among the smallest residual values.

Another property of the diagonal elements of a projection matrix is:

$$trace(H) = \sum_{k=1}^{N} h_{kk} = rank(Z) \tag{7}$$

The results (4) and (5) assume that $rank(Z) = q$, the number of parameters of the neural network. If $rank(Z) < q$, this means that some parameters are useless, i.e. that the model is unnecessarily complex given the training set. Thus, models such that $rank(Z) < q$ should be discarded. However, note that if $rank(Z) < q$, the hat matrix and hence the leverages are still defined (see Appendix 1), and verify (6) and (7). Theoretically, if $rank(Z) < q$, at least one of Z's singular values is zero (see Appendix 1).

1.2. Numerical considerations

In practice, it is difficult to make a statement about the rank deficiency of Z. As a matter of fact, though there exist efficient algorithms for the estimation of the singular values (Golub and Reinsch, 1970) (Anderson et al., 1999), these algorithms do generally not lead to zero singular values, except in very particular cases. Thus, the notion of the rank of Z is not usable in order to take the decision to discard a model.

A decision should rather be taken on the basis of whether or not the inverse of $Z^T Z$ needed for the estimation of a confidence interval can be estimated accurately. The most accurate estimation is performed using Z's singular value decomposition (SVD) $Z = U \Sigma V^T$, rather than using the Cholesky or LU decompositions of $Z^T Z$, or the QR decomposition of Z (Golub and Van Loan, 1983). We denote by $\widehat{U}$, $\widehat{V}$, and $\{\widehat{\sigma_i}\}$ the computed versions of U, V, and of the singular values $\{\sigma_i\}$ obtained with a Golub-Reinsch like SVD algorithm (used by Linpack, Lapack, Matlab, etc.). According to Appendix 1, one obtains the computed version of $(Z^T Z)^{-1}$ using:

$$\widehat{(Z^T Z)^{-1}} = \widehat{V}\,\widehat{(\Sigma^T \Sigma)^{-1}}\,\widehat{V}^T \quad \text{with} \quad \begin{cases} \left[\widehat{(\Sigma^T \Sigma)^{-1}}\right]_{ii} = \dfrac{1}{(\widehat{\sigma_i})^2} & \text{for } i=1 \text{ to } q \\ \left[\widehat{(\Sigma^T \Sigma)^{-1}}\right]_{ij} = 0 & \forall i \neq j \end{cases} \tag{8}$$

The question is how to detect whether (8) is accurate or not.

The absolute error on the i-th singular value is bounded by (Golub and Van

Loan, 1983):

$$| \sigma_i - \widehat{\sigma_i} | \leq \sigma_1 \, \varepsilon \quad \text{for } i=1 \text{ to } q \tag{9}$$

where ε is the computer unit roundoff ($\varepsilon \approx 10^{-16}$ for usual computers). The relative error on the smallest singular value is hence bounded by:

$$\left| \frac{\sigma_q - \widehat{\sigma_q}}{\sigma_q} \right| \leq \frac{\sigma_1}{\sigma_q} \varepsilon = \kappa(Z) \, \varepsilon \tag{10}$$

It is directly related to the condition number $\kappa(Z)$ of Z, the ratio of its largest to its smallest singular value. When $\kappa(Z)$ reaches $1/\varepsilon = 10^{16}$, the relative error on the smallest singular value may be 100%. Since (8) involves the inverse of the squared Jacobian, and hence the inverse of the computed squared singular values, neural candidates with $\widehat{\kappa(Z)} > 10^8$ should be discarded, as recommended in (Rivals and Personnaz, 2000).

If $\widehat{\kappa(Z)} << 10^8$, one can compute $(Z^T Z)^{-1}$ with a good precision using (8). Moreover, according to the error bound (10), for a network with $\kappa(Z) \leq 10^8$, the relative precision on the smallest singular value σ_q is smaller than 10^{-8}, and it is even smaller for the other singular values. The precision on $\kappa(Z)$ itself is hence excellent when $\widehat{\kappa(Z)} \leq 10^8$.

According to (9), it makes sense to estimate $r = \text{rank}(Z)$, if needed, as the number of computed singular values that are larger than the threshold $\widehat{\sigma_1} \, \varepsilon$:

$$\hat{r} = card \, \{\widehat{\sigma_i} > \widehat{\sigma_1} \, \varepsilon, \, i=1 \text{ to } q\} \tag{11}$$

However, the decision to discard a network should not be based on $\hat{r}$, because one may have $\hat{r} = q$ while $\kappa(Z) > 10^8$, and hence while $(Z^T Z)^{-1}$ is inaccurate.

When $r = q$, one could think of computing the leverages using (8), since $H = Z \, (Z^T Z)^{-1} \, Z^T$. However, as mentioned in (Rivals and Personnaz, 2000), it is better to use (see Appendix 1):

$$\widehat{h_{kk(R\&P)}} = \sum_{i=1}^{\hat{r}} (\widehat{u_{ki}})^2 \quad \text{for } k=1 \text{ to } N \tag{12}$$

where $\widehat{r}$ is computed with (11). As opposed to (8), expression (12) does not involve the inverse of the square of possibly inaccurate small singular values. The leverages computed with (12) are hence less sensitive to the ill-conditioning of Z than $(Z^T Z)^{-1}$ computed with (8).

However, we must consider the error on q first columns of U which span the range of Z. Let U_q denote the matrix of the q first columns of U. The angle between the range of U_q and that of its computed version $\widehat{U_q}$ (see Appendix 1) is approximately bounded by (Anderson et al., 1999):

$$angle\left(R(U_q), R(\widehat{U_q})\right) \leq \frac{\sigma_1 \, \varepsilon}{min_{j \neq i} \left| \sigma_i - \sigma_j \right|} \tag{13}$$

Thus, even if a model is very ill-conditioned, as long as the singular values of its Jacobian are not too close to one another, the leverages can be computed accurately with (12).

It can also be shown that $\widehat{U}$ is quasi orthogonal (Golub and Van Loan, 1983):

$$\widehat{U} = W + \Delta W \quad \text{with} \quad W^T W = W W^T = I_N \quad \text{and} \quad \|\Delta W\|_2 \leq \varepsilon \quad (14)$$

where I_N denotes the identity matrix of order N. Thus, for leverage values obtained with (12), even if $\widehat{U}$ is not an accurate version of U, the relations (6) and (7) are satisfied to roughly unit roundoff[1].

Note that if one is only interested in the leverages, they can be computed as accurately using the QR decomposition as with (12) (Dongarra et al., 1979) (see Appendix 1). The advantage is that the QR decomposition demands far less computations than SVD.

2. Comment for the proposition of (Monari and Dreyfus, 2002)

Monari and Dreyfus do not consider the notion of condition number, but focus on the values of the leverages. They choose to discard a model when the relations (6) and (7) are not satisfied for the computed values of its leverages. There are two problems with this proposition:

a) Property (7) is an equality, but Monari and Dreyfus do not specify a numerical threshold that could be used in order to take a decision. There is a similar problem with (6).

b) Instead of using (12), Monari and Dreyfus compute the leverages according to:

$$\widehat{h_{kk(M\&D)}} = \sum_{i=1}^{q} \left(\frac{1}{\widehat{\sigma_i}} \sum_{j=1}^{q} z_{kj} \, \widehat{v_{ji}} \right)^2 \quad \text{for } k=1 \text{ to } N \quad (15)$$

equation (A.4) in (Monari and Dreyfus, 2002). This equation (A.4) is derived from the expression $Z\,V\,(\Sigma^T \Sigma)^{-1}\,V^T Z^T$ of the hat matrix if Z if full rank, expression which is, strangely enough, obtained by using the SVD of Z for $(Z^T Z)^{-1}$ (expression A5 of the present paper), but not for Z itself. Whereas the computed values of the leverages using (12) are accurate provided the singular values are not to close (property (13)) and always satisfy (6) and (7) to unit roundoff (property (14)), there is no such result concerning the computed values obtained with (15).

---

[1] Proof that (6) is satisfied to unit roundoff:

$$\widehat{h_{kk(R\&P)}} = \sum_{i=1}^{\hat{r}} (\widehat{u_{ki}})^2 \leq \sum_{i=1}^{N} (\widehat{u_{ki}})^2 = 1 \quad \text{to unit roundoff}$$

Proof that (7) is satisfied to unit roundoff:

$$\sum_{k=1}^{N} \widehat{h_{kk(R\&P)}} = trace\left(\widehat{U_q} \left(\widehat{U_q}\right)^T\right) = trace\left(\left(\widehat{U_q}\right)^T \widehat{U_q}\right) = \hat{r} \quad \text{to unit roundoff}$$

This has the following consequences:

a) Assuming that a consistent threshold were specified for (6) and (7), because the leverages computed with (15) may be inaccurate, models well enough conditioned for the accurate estimation of confidence intervals may wrongly be discarded.

b) Conversely, in the case of models whose Jacobian Z is ill-conditioned, the leverages computed with (15) (and *a fortiori* with (12)) may satisfy (6) and (7), and hence the corresponding models may not be discarded, while confidence intervals are meaningless.

In the next section, these consequences are illustrated with numerical examples.

3. Numerical examples

3.1. Accuracy of the leverages computed with (12) and (15)

We construct a (N,2) matrix Z such that its condition number can be adjusted by tuning a parameter α, while the range of Z does not depend on the value of α:

$$Z = \left[\mathbf{1} \quad \mathbf{1} + \alpha\, \mathbf{c} \right] = \begin{bmatrix} 1 & 1 + \alpha\, c^1 \\ \cdot & \cdot \\ \cdot & \cdot \\ 1 & 1 + \alpha\, c^N \end{bmatrix} \tag{16}$$

where the $\{c^k\}$ are realizations of gaussian random variables (see the Matlab program given in Appendix 2). For a given $\mathbf{c}$, $R(Z) = span(\mathbf{1}, \mathbf{c})$ and hence the true leverages $\{h_{kk}\}$ have the same values $\forall\ \alpha \neq 0$. As an example, for a single realization of Z, with $N = 4$, and $\alpha = 10^{-12}$, we obtain the results reproduced in Appendix 2. In order to give statistically significant results, we performed 10 000 realizations of the matrix Z. Table 1 displays averaged results.

α	$\left\langle \widehat{\kappa(Z)} \right\rangle$	$\left\langle \left\| \hat{r} - \sum_{k=1}^{N} \widehat{h_{kk(R\&P)}} \right\| \right\rangle$	$\left\langle \left\| q - \sum_{k=1}^{N} \widehat{h_{kk(M\&D)}} \right\| \right\rangle$
10^{-6}	3.2154 10^{6}	3.7881 10^{-16}	1.2029 10^{-10}
10^{-8}	3.2154 10^{8}	1.8516 10^{-16}	1.2316 10^{-8}
10^{-12}	3.2154 10^{12}	1.8443 10^{-16}	1.2213 10^{-4}
10^{-15}	Inf	1.5901 10^{-16}	–

Table 1. Estimation of the leverages with formula (12) ($\{\widehat{h_{kk(R\&P)}}\}$) and formula (15) ($\{\widehat{h_{kk(M\&D)}}\}$).

When using (12), the computed values $\{\widehat{h_{kk(R\&P)}}\}$ satisfy (6) and their sum satisfies (7) to roughly unit roundoff for values of $\widehat{\kappa(Z)}$ up to 10^{16}: as a

matter of fact, this is ensured by property (14) of the estimate $\widehat{U}$. Moreover, according to property (13), since $\widehat{\sigma_1} - \widehat{\sigma_2} \approx 2.8$ $\forall\alpha$, the values of the $\{\widehat{h_{kk(R\&P)}}\}$ themselves are accurate.

When using (15), the sum of the $\{\widehat{h_{kk(M\&D)}}\}$ by far does not satisfy (7) to unit roundoff even when Z is well-conditioned. The $\{\widehat{h_{kk(M\&D)}}\}$ may also not satisfy (6) whereas the $\{\widehat{h_{kk(R\&P)}}\}$ always do (run the program of Appendix 2 with $\alpha = 10^{-15}$).

This example illustrates the lower accuracy of (15) with respect to (12). If the values of leverages are central to a given procedure, they definitely should be computed according to (12). The results are similar for any N and any vector $\boldsymbol{c}$, as well as for Jacobian matrices of real life problems[2].

3.2. Neural modeling

We consider a single input process simulated with:

$$y^k = sinc\left(10\ (x^k + 1)\right) + w^k \quad \text{for } k=1 \text{ to } N \tag{17}$$

where "*sinc*" denotes the cardinal sine function, and $N = 50$. The noise values $\{w^k\}$ are drawn from a gaussian distribution with variance $\lambda = 2\ 10^{-3}$, and the input values $\{x^k\}$ from an uniform distribution in [–1; 1].

Neural models with one layer of n_h "tanh" hidden neurons and a linear output neuron are considered, except the network without hidden neurons which consists of a single linear neuron. They are trained 50 times using a quasi-Newton algorithm starting from different small random initial parameters values, in order to increase the chance to reach an absolute minimum; the parameters corresponding to the lowest value of the cost function (2) are kept. The corresponding Mean Square Training Error is denoted by MSTE. The Approximate Leave One Out Score computed with (5) and (12) is denoted by ALOOS. The ALOOS is to be compared to a better, unbiased estimate of the performance computed on an independent test set of 500 examples drawn from the same distribution as the training set (such a test set is usually not available in real life), the MSPE.

The simulations are programmed in the C language. The SVD decomposition is performed with the Numerical Recipes routine "*svdcmp*" (Press et al. 2002), but in double precision. The leverages $\{\widehat{h_{kk(M\&D)}}\}$ (15) and the $\{\widehat{h_{kk(R\&P)}}\}$ (12) are then computed like in the Matlab program given in Appendix 2. The results obtained are shown in Table 2.

---

[2] The economic QR decomposition of Z can also be used (see Appendix 1): the values computed with (A13) do not differ from those computed with (12) by more than roughly the computer unit roundoff (check with the Matlab program of Appendix 2).

n_h	*MSTE*	*ALOOS*	*MSPE*	$\widehat{\kappa(Z)}$
0	$3.699\ 10^{-2}$	$4.136\ 10^{-2}$	$7.039\ 10^{-2}$	1.8
1	$9.506\ 10^{-3}$	$1.144\ 10^{-2}$	$1.083\ 10^{-2}$	$6.7\ 10^{2}$
2	$3.181\ 10^{-3}$	$4.831\ 10^{-3}$	$6.866\ 10^{-3}$	$8.4\ 10^{2}$
3	$2.153\ 10^{-3}$	$4.039\ 10^{-3}$	$4.783\ 10^{-3}$	$2.1\ 10^{9}$
4	$1.888\ 10^{-3}$	$5.316\ 10^{-7}$	$4.436\ 10^{-3}$	$9.0\ 10^{9}$

Table 2. Training of neural models with an increasing number n_h of hidden neurons. The rows corresponding to $\widehat{\kappa(Z)} > 10^8$ (ill-conditioned networks) are shaded.

The outputs and the residuals of the networks with 2 and 3 hidden neurons are shown on Figures 1 and 2 respectively. Though the model with 3 hidden neurons has a slightly lower MSPE than the network with 2, it is unreliable in the sense that one is unable to estimate correct confidence intervals for the regression with this network: computing $\left(Z^T Z\right)^{-1}$ with (8) and multiplying it by $Z^T Z$ leads to a matrix that differs significantly from the identity matrix (by more than 2 for some of its elements). Fortunately, following (Rivals and Personnaz, 2000), we can discard this network right away on the basis of its too large condition number.

But, suppose that $\widehat{\kappa(Z)}$ is ignored, and that, following (Monari and Dreyfus, 2002), only the computed leverage values are considered. The results obtained with (12) and (15) are given in Table 3.

n_h	$max\left(\widehat{h_{kk(R\&P)}}\right)$	$\hat{r} - \sum_{k=1}^{N} \widehat{h_{kk(R\&P)}}$	$max\left(\widehat{h_{kk(M\&D)}}\right)$	$q - \sum_{k=1}^{N} \widehat{h_{kk(M\&D)}}$
0	$8.1648771\ 10^{-2}$	$2.22\ 10^{-16}$	$8.1648771\ 10^{-2}$	0.0
1	$8.5215146\ 10^{-1}$	0.0	$8.5215146\ 10^{-1}$	$-1.1\ 10^{-14}$
2	$8.6468122\ 10^{-1}$	0.0	$8.6468122\ 10^{-1}$	$9.8\ 10^{-15}$
3	$8.8121097\ 10^{-1}$	$-1.8\ 10^{-15}$	$8.8121097\ 10^{-1}$	$-6.2\ 10^{-11}$
4	$9.9999983\ 10^{-1}$	0.0	$9.9999981\ 10^{-1}$	$-1.6\ 10^{-8}$

Table 3. Observing the computed leverage values.

Both relations (6) and (7) are satisfied for networks with 3 and even 4 hidden neurons, be the leverages computed with (12) ($\{\widehat{h_{kk(R\&P)}}\}$) or (15) ($\{\widehat{h_{kk(M\&D)}}\}$). It proves that only checking (6) and (7) for the leverage values does not lead to discard the unusable models with 3 and 4 hidden neurons.

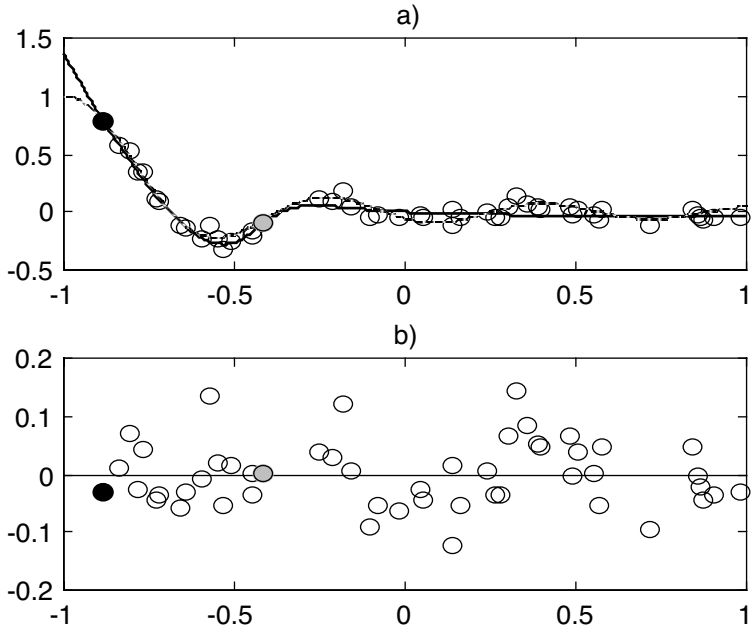

Figure 1. Network with 2 hidden neurons ($\widehat{\kappa(Z)} = 8.4\ 10^{2}$): a) regression (dotted line), model output (continuous line), training set (circles); b) residuals. The training example and the residual corresponding to the largest leverage ($8.65\ 10^{-1}$) are marked with a circle filled in black. A second leverage is larger than 0.5 ($5.89\ 10^{-1}$), and the corresponding training example and residual are marked with a circle filled in grey.

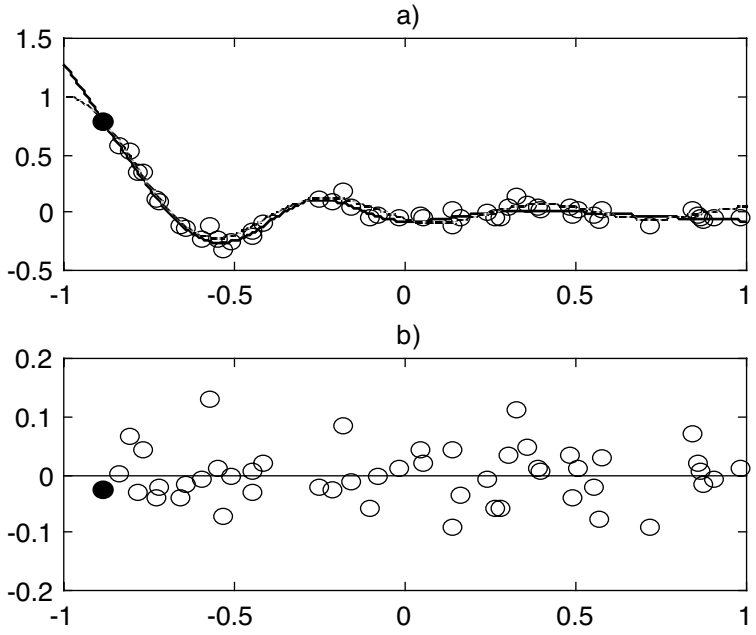

Figure 2. Network with 3 hidden neurons ($\widehat{\kappa(Z)} = 2.1\ 10^{9}$): a) regression (dotted line), model output (continuous line), training set (circles); b) residuals. The training example and the residual corresponding to the largest leverage ($8.81\ 10^{-1}$) are marked with a circle filled in black.

Let us have a closer look at the performance of the models with 2 and 3

hidden neurons, and at the interpretation of the leverage values. Figures 1 and 2 display the training examples and the residuals corresponding to leverage values larger than 0.5. For both networks, the largest leverage value corresponds to an example which lies at the boundary of the input domain explored by the training set. This is a very typical situation of an influent example. For the network with 2 hidden neurons, a second leverage value is larger than 0.5: the fact that the corresponding example is located at an inflexion point of the model output is the sign of its large influence on the parameter estimate.

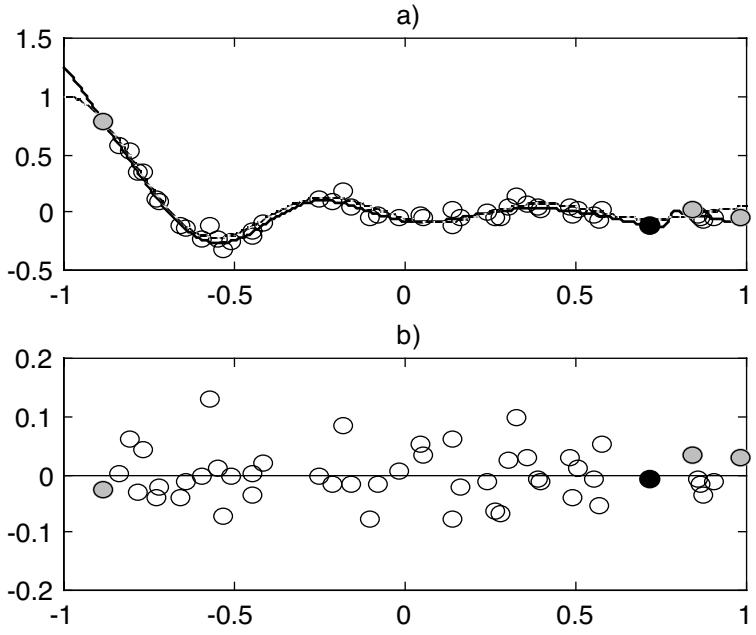

Figure 3. Network with 4 hidden neurons ($\widehat{\kappa(Z)} = 9.0\ 10^{9}$): a) regression (dotted line), model output (continuous line), training set (circles); b) residuals. The training example and the residual corresponding to the largest leverage ($9{,}9999983\ 10^{-1}$) are marked with a circle filled in black. The training examples and residuals corresponding to three other leverages larger than 0.5 are marked with a circle filled in grey.

Let us now examine the network with 4 hidden neurons. Four of its leverage values are larger than 0.5. These values are the following (from the smallest to the largest corresponding abscissa, see Figure 3):

$$\begin{cases} \widehat{h_{38\,38(R\&P)}} = 7.3268195\ 10^{-1} \\ \widehat{h_{39\,39(R\&P)}} = 9.9999983\ 10^{-1} \\ \widehat{h_{33\,33(R\&P)}} = 9.9981558\ 10^{-1} \\ \widehat{h_{1\,1(R\&P)}} = 8.4477304\ 10^{-1} \end{cases}$$

The large leverage values correspond to:

– the two extreme examples (38 and 1),

– two overfitted examples (39 and 33).

As a matter of fact, large leverage values, i.e. values close to one but not necessarily larger than one, are the symptom of local overfitting at the corresponding training examples, or of extreme examples of the training set that are relatively isolated, and hence influent. However, checking only (6) and (7) for the leverage values would not lead to the detection of the misbehavior of this network.

Finally, this example shows that ill-conditioning is not systematically related to leverage values close to one: the largest leverage value of the very ill-conditioned neural network with 3 hidden neurons ($\widehat{\kappa(Z)} = 2.1\ 10^{9}$) equals $8.81\ 10^{-1}$, and is hence not much larger than the largest leverage value of the well-conditioned neural network with 2 hidden neurons ($\widehat{\kappa(Z)} = 8.4\ 10^{2}$), which equals $8.65\ 10^{-1}$. Ill-conditioning is not systematically related to local overfitting, but rather to a global parameter redundancy.

4. Conclusion

Our conclusion is threefold:

– In order to validate only neural candidates whose approximate parameter covariance matrix and confidence intervals can be reliably estimated, the first condition should be that the condition number of their Jacobian does not exceed the square root of the inverse of the computer unit roundoff (usually 10^{8}).

– If a procedure relies heavily on the computed values of the diagonal elements of the hat matrix, the leverages, the latter should be computed according to expression (12), as recommended in (Rivals and Personnaz, 2000), rather than according to the expression (15) given in (Monari and Dreyfus, 2002): only the computation according to (12) ensures that the computed hat matrix is a projection matrix, and that it is accurate.

– For candidates whose condition number is small enough, and for which the leverages have been computed as accurately as possible according to (12), one may check *additionally* if none of the leverage values is close to one, as already proposed in (Rivals and Personnaz, 1998). Leverage values close to, but not necessarily larger than one are indeed the symptom of overfitted examples, or of isolated examples at the border of the input domain delimited by the training set.

5. Other comment for (Monari and Dreyfus, 2002)

This comment concerns the estimation of the performance of the selection method presented in (Monari and Dreyfus, 2002), for its comparison to selection methods proposed by other authors. As in (Anders and Korn,

1999), the process to be modeled is simulated, and its output is corrupted by a gaussian noise with known variance λ. In order to perform statistically significant comparisons between selection methods, 1000 realizations of a training set of size N are generated. A separate test set of 500 examples is used for estimating the "generalization mean square error" (GMSE) of the selected models, and the following performance index is computed:

$$\rho = \frac{GMSE - \lambda}{\lambda} \tag{18}$$

equation (5.3) in (Monari and Dreyfus, 2002). In the case $N = 100$, two values of $\langle\rho\rangle$, the average value of ρ on the 1000 training sets, are given:

a) a value of 126% corresponding to the above definition;

b) a value of 27% corresponding to a GMSE computed on a part of the test set only: strangely enough, 3% of the examples of the test set are considered as "outliers", and discarded from the test set. This value of 27% is compared to the values of $\langle\rho\rangle$ obtained by other selection procedures with the whole test set.

This second value of 27% is meaningless. Putting apart the fact that considering examples of the performance estimation set as outliers is questionable, lets call GMSE* the GMSE obtained in b). In the most favorable case for (Monari and Dreyfus, 2002), i.e. the case where we assume that the examples discarded by Monari and Dreyfus correspond to the largest values of the gaussian noise (and not to model errors), this GMSE* should not be compared to λ, but to the variance λ^* of a noise with a truncated gaussian distribution (without its two 1.5% tails). In the example, $\lambda = 5\ 10^{-3}$, $\lambda^* = 4.4\ 10^{-3}$. Thus, the ratio:

$$\rho^* = \frac{GMSE^* - \lambda^*}{\lambda^*} > \frac{GMSE^* - \lambda}{\lambda} \tag{19}$$

would be more representative of the real model performance.

To conclude, the second value of $\langle\rho\rangle$ obtained in (Monari and Dreyfus, 2002) by discarding some examples of the test set can by no means be compared to those obtained by other selection procedures correctly using the whole test set for the performance estimation.

Appendix 1

This appendix summarizes results used in the paper; for details, see (Golub and Van Loan, 1983).

Theorem for the Singular Value Decomposition (SVD)

Consider a (N,q) matrix Z with $N \geq q$ and $rank(Z) = r \leq q$. There exist a (N,N) orthogonal matrix U and a (q,q) orthogonal matrix V such that:

$$Z = U \Sigma V^T \tag{A1}$$

where Σ is a (N,q) matrix such that $[\Sigma]_{ij} = 0$ for $i \neq j$, and whose elements $\{[\Sigma]_{ii}\}$, denoted by $\{\sigma_i\}$, are termed the singular values of Z, with:

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_q \geq 0 \tag{A2}$$

If $rank(Z) = r < q$, $\sigma_1 \geq \ldots \geq \sigma_r \geq \sigma_{r+1} = \ldots = 0$.
The r first columns of U form an orthonormal basis of the range of Z.

Condition number using SVD

If $rank(Z) = q$, and using the matrix 2-norm[3], the condition number $\kappa(Z)$ of Z is expressed as:

$$\kappa(Z) = \|Z\|_2 \, \|Z^{-1}\|_2 = \frac{\sigma_1}{\sigma_q} \tag{A3}$$

If $\kappa(Z)$ is large, the matrix Z is ill-conditioned. We have the property:

$$\kappa(Z^T Z) = (\kappa(Z))^2 \tag{A4}$$

Inverse of $Z^T Z$ using SVD

If $rank(Z) = q$, the inverse of $Z^T Z$ exists and, using the SVD of Z, can be expressed as:

$$(Z^T Z)^{-1} = V (\Sigma^T \Sigma)^{-1} V^T \tag{A5}$$

where $(\Sigma^T \Sigma)^{-1}$ is a (q,q) diagonal matrix with:

$$[(\Sigma^T \Sigma)^{-1}]_{ii} = \frac{1}{\sigma_i^2} \quad \text{for } i=1 \text{ to } q \tag{A6}$$

Pseudo-inverse of Z using SVD

Any (N,q) matrix Z with rank $r \leq q$ has a pseudo-inverse. It equals:

$$Z^I = V \Sigma^I U^T \tag{A7}$$

where Σ^I is a (q,N) matrix whose only non zero elements are the first r diagonal elements:

$$[\Sigma^I]_{ii} = \frac{1}{\sigma_i} \quad \text{for } i=1 \text{ to } r \tag{A8}$$

Orthogonal projection matrix on the range of Z using SVD

The (N,N) projection matrix H on the range of any (N,q) matrix Z is given by:

$$H = Z Z^I \tag{A9}$$

Using the SVD of Z, we obtain:

$$H = U \Sigma V^T V \Sigma^I U^T = U \Sigma \Sigma^I U^T \tag{A10}$$

where the matrix $\Sigma \Sigma^I$ is hence a (N,N) diagonal matrix whose r first diagonal elements are equal to 1 and all the others to 0, see (Golub and Van Loan,

---

[3] The 2-norm of a matrix A is defined as:

$$\|A\|_2 = \sup_{\boldsymbol{x} \neq \boldsymbol{0}} \left(\frac{\|A \boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} \right)$$

1983). Thus, the diagonal elements of H, the leverages, are given by:

$$h_{kk} = \sum_{i=1}^{r} (u_{ki})^2 \quad \text{for } k=1 \text{ to } N \tag{A11}$$

Theorem for the QR Decomposition

Consider a (N,q) matrix Z with $N \geq q$ and $rank(Z) = q$. There exist a (N,N) orthogonal matrix Q and an upper triangular (q,q) matrix R such that:

$$Z = Q \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{A12}$$

The q first columns of Q form an orthonormal basis of the range of Z.

Leverages using QR

Using the QR decomposition of Z, we obtain:

$$h_{kk} = \sum_{i=1}^{q} (q_{ki})^2 \quad \text{for } k=1 \text{ to } N \tag{A13}$$

Angle between two subspaces

Let S_1 and S_2 denote the ranges of two (N,q) matrices Z_1 and Z_2, and H_1 and H_2 the orthogonal projection matrices on S_1 and S_2. The distance between the two subspaces S_1 and S_2 is defined as:

$$dist(S_1, S_2) = \|H_1 - H_2\|_2 \tag{A14}$$

The angle between S_1 and S_2 is defined as:

$$angle(S_1, S_2) = arcsin\big(dist(S_1, S_2)\big) \tag{A15}$$

Appendix 2

Below follows the text of a Matlab program which constructs an ill-conditioned matrix Z (for a small value of α), and computes the leverage values using SVD, and formula (12) and formula (15), and also using the more economic QR decomposition which is as accurate as (12):

```
clc
clear all
format compact
format short;

% construction of the (N,q) matrix Z
randn('seed',12345);
N=4;
q=2;
alpha = 1e-12
c = randn(N,1);
Z = [ones(N,1) ones(N,1)+alpha*c];
condZ = cond(Z)

% singular value decomposition of Z
[U,S,V] = svd(Z);
s = diag(S);
diff_s = s(1)-s(2)
```

```
% "True" leverages
Z1 = [ones(N,1) c];
[U1,S1,V1] = svd(Z1);
diagH_true = zeros(N,1);
for k=1:N
  for i=1:q
    diagH_true(k) = diagH_true(k) + U1(k,i)^2;
  end
end
diagH_true = diagH_true

% Rivals and Personnaz estimates (12) of the leverages
tol = max(s)*eps;
r = sum(s > tol);
diagH_RP = zeros(N,1);
for k=1:N
  for i=1:r
    diagH_RP(k) = diagH_RP(k) + U(k,i)^2;
  end
end
diagH_RP = diagH_RP
r_sumd_RP = r-sum(diagH_RP)

% Monari and Dreyfus estimates (15) of the leverages
diagH_MD = zeros(N,1);
for k=1:N
  for i=1:q
    toto = 0;
    for j=1:q
      toto = toto + Z(k,j)*V(j,i);
    end
    diagH_MD(k) = diagH_MD(k) + (toto/s(i))^2;
  end
end
diagH_MD = diagH_MD
q_sumd_MD = q-sum(diagH_MD)

% Economic estimates of the leverages using the QR
decomposition
[Q,R] = qr(Z);
diagH_QR = zeros(N,1);
for k=1:N
  for i=1:q
    diagH_QR(k) = diagH_QR(k) + Q(k,i)^2;
  end
end
diagH_QR = diagH_QR
r_sumd_QR = r-sum(diagH_QR)
```

Output of the program:

```
alpha =
   1.0000e-12
condZ =
   2.8525e+12
diff_s =
    2.8284
diagH_true =
    0.2719
    0.2580
    0.7758
    0.6943
diagH_RP =
    0.2719
    0.2580
```

```
    0.7759
    0.6943
r_sumd_RP =
     0
diagH_MD =
    0.2720
    0.2579
    0.7756
    0.6944
q_sumd_MD =
   1.8643e-05
diagH_QR =
    0.2719
    0.2580
    0.7759
    0.6943
r_sumd_QR =
     0
```

References

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

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. (1999). LAPACK User's Guide, Third Edition. Siam, Philadelphia.

Dongarra, J., Moler, C. B., Bunch, J. R., and Stewart, G. W. (1979). LINPACK User's Guide. Siam, Philadelphia.

Golub, G. H., and Reinsch, C. (1970). Singular value decomposition and least-squares solutions. Numerische Mathematik, 14, 403-420.

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

Monari, G., and Dreyfus, G. (2002). Local Overfitting Control via Leverages. Neural Computation, 14, 1481-1506

Press, W. H., Teukolsky, S.A., Vetterling, W. T., and Flannery, B. P. (2002). Numerical recipes in C. Cambridge University Press.

Rivals I., and 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, Leuwen, 8-10 July 1998.

Rivals I., and Personnaz, L. (2000). Construction of confidence intervals for neural networks based on least squares estimation. Neural Networks 13, 463-484.