Neural Network Modeling for Small Datasets

Salvatore INGRASSIA
Dipartimento di Economia e Statistica
Università della Calabria
87036 Arcavacata di Rende (CS), Italy
(*s.ingrassia@unical.it*)

Isabella MORLINI
Dipartimento di Scienze Sociali
Cognitive e Quantitative
Università di Modena e Reggio Emilia
42100 Reggio Emilia, Italy
(*morlini.isabella@unimore.it*)

Neural network modeling for small datasets can be justified from a theoretical point of view according to some of Bartlett's results showing that the generalization performance of a multilayer perceptron (MLP) depends more on the L_1 norm $\|\mathbf{c}\|_1$ of the weights between the hidden layer and the output layer rather than on the total number of weights. In this article we investigate some geometrical properties of MLPs and drawing on linear projection theory, we propose an equivalent number of degrees of freedom to be used in neural model selection criteria like the Akaike information criterion and the Bayes information criterion and in the unbiased estimation of the error variance. This measure proves to be much smaller than the total number of parameters of the network usually adopted, and it does not depend on the number of input variables. Moreover, this concept is compatible with Bartlett's results and with similar ideas long associated with projection-based models and kernel models. Some numerical studies involving both real and simulated datasets are presented and discussed.

KEY WORDS: Degrees of freedom; Indirect measure; Neural model; Small dataset.

1. INTRODUCTION

An important issue in statistical modeling is related to so called indirect measures or *virtual sensors*. This involves the prediction of variables that are quite expensive to measure (e.g., the viscosity or the concentration of certain chemical species, some mechanical features) using other variables that are more easily measured, like, for example, the temperature or the pressure (see De Veaux and Ungar 1996). Such problems often involve some difficulties: (1) The available datasets are small (e.g., typical chemical datasets have only 30–100 observations); (2) the input–output relation to be estimated is nonlinear; and (3) there are many predictor variables, but, because linearity cannot be assumed, it is quite difficult to reduce the dimensionality of the problem by choosing a good subset of predictors or suitable underlying features.

Assume that we are provided with a dataset of N pairs $(\mathbf{x}_n, y_n)$ of m-dimensional input vectors $\mathbf{x}_n$ and scalar target values y_n. In many applications the scientific law underlying the relationship between the response variable and the predictors is (at least partially) known, and the experimental data can be used to test the model assumptions and to estimate the adaptive parameters; such models are called *first-principle models*. Here we focus on the opposite case, that is, on applications in which the underlying first principle is unknown or the system is too complex to be mathematically described, so that the data are used to extract knowledge and afterward to derive some practical and useful models (see Cherkassky and Mulier 1998). In this context, neural networks may be attractive.

It may happen that the size N of the available sample is small compared with the number of weights of the neural network [e.g., a single hidden-layer multilayer perception (MLP) with p nodes in the hidden layer has $W = p(m + 2) + 1$ weights], so that the resulting neural model is considered overparameterized. In the literature, this problem has been approached in different ways. One strategy is based on the regularization technique called *jittering*, which consists of adding artificial noise to the input during training; training with jitter is a form of smoothing related to kernel regression and other regularization methods, such as ridge regression (see, e.g., An 1996; Azencott, Doutriaux, and Younes 1993). Another possibility is to reduce the dimension m of the input space using suitable preprocessing techniques; this strategy is very useful when the dimension m of the input space is quite large (see, e.g., Yuan and Fine 1998).

Here we consider a third approach, based on overparameterized networks, which has been pursued in the literature by some authors. For instance, De Veaux, Schumi, Schweinsberg, and Ungar (1998) worked on a one-hidden layer perceptron with more than 200 weights trained with 61 data units concerning measurements taken from a polymer pilot plant; Lawrence, Giles, and Tsoi (1996, 1997) provided many simulations with overparameterized MLPs, concluding that oversized networks can result in low training and generalization errors. These neural models can be justified from a theoretical point of view according to some results published by Bartlett (1998) showing that the generalization performance of an MLP depends more on the L_1 norm $\|\mathbf{c}\|_1$ of the weights between the hidden layer and the output layer rather than on the total number of weights. These results suggest caution when transferring some basic statistical paradigms into the neural framework. Indeed, they seem to contradict the usual assumption in statistical modeling according to which the number of parameters should be (much) smaller than the number of observations used for their estimation. This contradiction is particularly apparent when the neural model is selected according to some goodness-of-fit statistic, like, for example, the Akaike information criterion (AIC), the Bayesian information criterion (BIC/SBC), or the final prediction error (FPE) (see Sec. 2 for details). Indeed, even if the underlying theories do not hold for neural models, they are quite simple to compute, and they are often considered crude estimates of the generalization error. Actually, in these model selection criteria the number K of degrees of freedom is set equal

TECHNOMETRICS, AUGUST 2005, VOL. 47, NO. 3
DOI 10.1198/004017005000000058

to the number W of weights, so they are useless when $W > N$; in this case, for example, both the FPE and the unbiased estimate of variance (UEV) assume negative values, and this does not make sense.

In contrast, in the smoothing literature, notions of degrees of freedom different from the number of adaptive parameters in the final model, have a long history, and are well established. For example, for a linear smoothing operator $\mathbf{S}$, where $\widehat{\mathbf{y}} = \mathbf{S}\mathbf{y}$ is the smooth of $\mathbf{y}$, Hastie and Tibshirani (1990, sec. 3.5) gave three definitions of degrees of freedom useful for different purposes. In the generalized cross validation (GCV) criterion the model complexity is measured by the trace of the smoothing matrix $\mathbf{S}$; Friedman (1991) and many of the discussants of his article proposed various definitions of degrees of freedom for multivariate adaptive regression splines. They motivated their definitions in a parametric framework. Recently, Ye (1998) and Hodges and Sargent (2001) extended the necessity of finding new notions of degrees of freedom, different from the number of adaptive parameters, to all richly parameterized models and to complex hierarchical procedures. The generalized degrees of freedom proposed by Ye (1998) is completely general in that it is applicable to all complex modeling procedures. However, it is costly to compute, because it requires Monte Carlo simulations. Hodges and Sargent's degrees of freedom are defined for models that can be reexpressed as linear models, in the formal sense that the left side is known and the right side consists of known linear combinations of unknown parameters. Both of these notions arise from a parametric framework.

In this article we focus on nonlinear projection methods of the form

$$f_p(\mathbf{x}) = \sum_{k=1}^{p} c_k \tau(\mathbf{a}_k'\mathbf{x} + b_k) + c_0, \tag{1}$$

where $\mathbf{a}_1, \dots, \mathbf{a}_p \in \mathbb{R}^m$, $b_1, \dots, b_p, c_0, c_1, \dots, c_p \in \mathbb{R}$, and τ is a sigmoidal function. However, the obtained results can be extended to other kinds of neural models, like *radial basis function* and *projection pursuit regression* models (see Hastie, Tibshirani, and Friedman 2001), where the function τ has a different nonlinear form. These are additive models, but in the derived features $\tau(\mathbf{a}_k'\mathbf{x} + b_k)$ rather than in the inputs themselves.

We move from a nonparametric point of view, exclusively drawing on geometrical considerations. We extend some geometrical properties of the MLP reported by Ingrassia (1999) and generalize preliminary results given by Ingrassia and Morlini (2002, 2004), showing that the input-to-hidden weights and the hidden-to-output weights in an MLP play different roles. Drawing on the projection theory of linear models, here we introduce the *equivalent number of degrees of freedom* (edf), K, which is equal to the trace of a suitable projection matrix. For model selection purposes, according to indices like AIC and BIC, this quantity can be set equal to the number of neurons $p + 1$ in the hidden layer [in eq. (1) the constant c_0 can be considered a weight between the output and a hidden neuron with value set to 1]—that is, the dimension of the projection space intrinsically found by the mapping function—whereas to estimate the error variance, sometimes better approximations of K (which also depend on the adopted regularization technique) should be considered.

Finally, using both real and simulated datasets, we present some empirical studies linking the number of neurons in the hidden layer to the L_1 norm $\|\mathbf{c}\|_1$. However, as shown in our examples, it is unstable across different simulations for large networks as well. Besides, as outlined by Bartlett (1998), for the size of the weights between the hidden layer and the output layer to be an appropriate measure of the mapping function complexity, the gradient descent algorithm must reach a suitable minimum of the error function. This is not always verified in practice, especially with small training datasets. The simulation examples show that the estimates of the error variance using such an edf K behaves quite well, whereas the error variances estimates based on the total number of weights W are far from the true value or may assume even negative values.

The article is organized as follows. In the next section we review the generalization performance of a neural model in the context of Bartlett's results and state the purpose for which we want to define the degrees of freedom of an MLP and similar projection methods. In Section 3 we prove some geometric properties of the sigmoidal functions that motivate our notion of degrees of freedom. In Section 4 we introduce our definition of equivalent degrees of freedom for multilayer perceptrons and other projection methods; we also provide relationships with other approaches. In Section 5 we present simulations relating the constant $\|\mathbf{c}\|_1$ to the number of hidden units and compare the error variance estimates computed with different measures of degrees of freedom. Finally, in Section 6 we give some concluding remarks.

2. EVALUATION OF THE PERFORMANCE OF THE NETWORK WITH SMALL DATASETS

2.1 The Problem

Let $(\mathbf{X}, Y)$ be a pair of a random vector $\mathbf{X}$ and a random variable Y with joint probability distribution $p(\mathbf{x}, y)$, where $\mathbf{X}$ is the m-dimensional input vector with values in some space $\mathcal{X} \subseteq \mathbb{R}^m$ and Y is a response variable with values in $\mathcal{Y} \subseteq \mathbb{R}$. Throughout this article we assume that the input–output relation can be written as $Y = \phi(\mathbf{x}) + \varepsilon$, where ε is a random variable with mean 0 and finite variance. The unknown functional dependency $\phi(\mathbf{x}) = \mathbb{E}[Y|\mathbf{x}]$ is estimated by means of the function $f_p(\mathbf{x})$ realized by an MLP with m inputs, p neurons in the hidden layer, and one neuron in the output,

$$f_p(\mathbf{x}) = \sum_{k=1}^{p} c_k \tau(\mathbf{a}_k'\mathbf{x} + b_k) + c_0, \tag{2}$$

where $\tau(\cdot)$ is a sigmoidal function-like $\tau(z) = \tanh(z)$ or $\tau(z) = (1 + \mathrm{e}^{-z})^{-1}$—and $\mathbf{a}_1, \dots, \mathbf{a}_p \in \mathbb{R}^m$, $b_1, \dots, b_p, c_0, c_1, \dots, c_p \in \mathbb{R}$. We denote by $\mathbf{A}$ the $p \times m$ matrix with rows $\mathbf{a}_1', \dots, \mathbf{a}_p'$, and we set $\mathbf{b} = (b_1, \dots, b_p)$ and $\mathbf{c} = (c_1, \dots, c_p)$. Because such quantities are called *weights*, we denote them by $\mathbf{w}$, so that $\mathbf{w} \in \mathbb{R}^{p(m+2)+1}$, and sometimes we write $f(\mathbf{x}, \mathbf{w})$.

Let $\mathcal{F}_p$ be the set of all functions of type (2) for a fixed p,

$$\mathcal{F}_p = \left\{ f_p(\mathbf{x}) = \sum_{k=1}^{p} c_k \tau(\mathbf{a}_k'\mathbf{x} + b_k) + c_0 : \right.$$
$$\left. \mathbf{a}_1, \dots, \mathbf{a}_p \in \mathbb{R}^m, b_1, \dots, b_p, c_0, c_1, \dots, c_p \in \mathbb{R} \right\},$$

for $1 \le p \le N$. (We discuss the upper bound N on p in Sec. 3.) In what follows we suppress the index p in f_p and $\mathcal{F}_p$ for simplicity of notation. The problem is to find the function $f^{(0)} = f(\mathbf{w}^{(0)})$ in the set $\mathcal{F}$ such that the *generalization error* (also called the *expected risk*),

$$\mathcal{E}(f) = \int [y - f(\mathbf{x})]^2 p(\mathbf{x}, y)\, d\mathbf{x}\, dy, \tag{3}$$

where the integral is over $\mathcal{X} \times \mathcal{Y}$, attains its minimum, that is,

$$f^{(0)} = \arg\min_{f \in \mathcal{F}} \mathcal{E}(f), \tag{4}$$

and $\mathbf{w}^{(0)}$ denotes the weights of $f^{(0)}$. In practice, the distribution $p(\mathbf{x}, y)$ is unknown, but we have a sample, $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, of N iid realizations of $(\mathbf{X}, Y)$, so that we can compute the *empirical error* or the *empirical risk*,

$$\widehat{\mathcal{E}}(f, \mathcal{D}) = \sum_{(\mathbf{x}_n, y_n) \in \mathcal{D}} (y_n - f(\mathbf{x}_n))^2. \tag{5}$$

Let $f_{\mathcal{D}}^{(0)} = f(\widehat{\mathbf{w}}^{(0)})$ denote the function of $\mathcal{F}$ that minimizes $\widehat{\mathcal{E}}(\cdot)$,

$$f_{\mathcal{D}}^{(0)} = \arg\min_{f \in \mathcal{F}} \widehat{\mathcal{E}}(f), \tag{6}$$

where the $\widehat{\mathbf{w}}^{(0)}$ are the weights of $f_{\mathcal{D}}^{(0)}$.

The problem is to explore the linkage between the empirical error (5) and the true statistical performance of the network, namely the generalization error (3); in other words, to investigate the conditions that may ensure that $f_{\mathcal{D}}^{(0)}$ is a reasonable estimate of $f^{(0)}$. Usually the sample $\mathcal{D}$ is partitioned in three independent datasets: a learning set (or training set) $\mathcal{L}$, a validation set $\mathcal{V}$, and a test set $\mathcal{T}$. The error (5) is referred to as the *learning error* (or *training error*) $\widehat{\mathcal{E}}(f, \mathcal{L})$, the *validation error* $\widehat{\mathcal{E}}(f, \mathcal{V})$, or the *test error* $\widehat{\mathcal{E}}(f, \mathcal{T})$, the first computed using the learning set $\mathcal{L}$, the second computed with the validation set $\mathcal{V}$, and the last computed using the test set $\mathcal{T}$. The learning error $\mathcal{L}$ is a set of examples used to fit the parameters of the model. The validation set and the test set play different roles; the set $\mathcal{V}$ is a set of examples used to tune the parameters of a model (e.g., to choose the number of hidden units in a neural network), and the set $\mathcal{T}$ is a set of examples used only to assess the performance of a fully specified model (see, e.g., Ripley 1996; Hastie et al. 2001). However, very often the sample $\mathcal{D}$ is split into only two different groups, the learning set $\mathcal{L}$ and the test set $\mathcal{T}$.

Both the learning set and the test set are used to estimate the parameters $\widehat{\mathbf{w}}^{(0)}$, because $\widehat{\mathcal{E}}(f, \mathcal{L})$ is the function to be minimized, whereas $\widehat{\mathcal{E}}(f, \mathcal{T})$ is the function used to control overfitting. As the learning (or training) process proceeds, at the beginning both the learning and the test errors generally decrease, but there comes a time when the test error starts to increase even though the learning error is still decreasing. It is then judged that overfitting is occurring. The training is then halted, and the current estimate of the weights is chosen to be $\widehat{\mathbf{w}}^{(0)}$. This technique is called *early stopping*. If the test error never decreases during the learning process, then the network is judged to be underparameterized, and a larger one is recommended. Otherwise, a local minimum is reached by the optimization algorithm, and initialization parameters should be changed.

In virtual sensors, the total amount of data is often quite limited, and one would like to use most of the data for training purposes to achieve a higher likelihood of selecting a good network. For example, De Veaux et al. (1998) worked on a set $\mathcal{D}$ of 61 observations, and they first trained the MLP with a learning set $\mathcal{L}$ of 50 samples to select the best architecture (the one minimizing the test error $\mathcal{T}$ of the remaining 11 observations). They subsequently trained the selected model on the basis of the whole set $\mathcal{D}$ of available observations. When the data cannot be split into three sets and the validation error cannot be computed for estimating the generalization error, model selection criteria are used to compare different networks. These criteria are based on the test error but also take into account a model complexity measure. (See Kadane and Lazar 2004 for a complete review of these indices and their theoretical basis both in the Bayesian and frequentist framework.) Traditionally, for general modeling problems, practitioners tend to measure model complexity with the number of parameters in the final model, because of the coincidence of this quantity and the degrees of freedom in linear models. Still, for linear models, the number of parameters can also be interpreted as the cost of the estimation process and thus can be used for obtaining an unbiased estimate of the error variance. The seminal work on model selection is based on the parametric statistics literature and is quite vast, but it must be noted that although model selection techniques for parametric models have been widely used in the past 30 years, surprisingly little work has been done on the application of these techniques in a semiparametric or nonparametric context.

Let f_K be a statistical model based on K degrees of freedom; in the rest of the article, N denotes the size of the learning set. In general, these model selection criteria, denoted here by Π, are an extension of the maximum likelihood and have the form

$$\Pi = \widehat{\mathcal{E}}(f_K) + \mathcal{C}_K, \tag{7}$$

where the term $\widehat{\mathcal{E}}(f_K) = \widehat{\mathcal{E}}(f_K, \mathcal{L})$ is the empirical error of the model f_K based on the training set and $\mathcal{C}_K$ is a complexity term representing a penalty that grows as the number K of degrees of freedom in the model increases. If the model f_K is too simple, it will give a large value for the criterion Π because the empirical error is large; whereas a model f_K that is too complex will have a large value for the criterion Π because the complexity term is large. Typical indices include the AIC (Akaike 1974),

$$\mathrm{AIC} = \log(\widehat{\mathcal{E}}(f_K)) + \frac{2K}{N},$$

where N is the size of the learning set. Another criterion used in linear regression models with least squares estimates of the parameters is the Schwarz (1978) BIC (or SBC). The BIC/SBC results in the selection of the model for which the following expression is the minimum:

$$\mathrm{BIC} = \log(\widehat{\mathcal{E}}(f_K)) + \frac{K \log(N)}{N}.$$

Besides AIC and BIC, other selection methods for which the algebraic expression requires normality and least squares estimates of the parameters are the FPE (Akaike 1970),

$$\mathrm{FPE} = 6\widehat{\mathcal{E}}(f_K)\left(\frac{1 + K/N}{1 - K/N}\right), \tag{8}$$

the GCV error,

$$\mathrm{GCV} = \widehat{\mathcal{E}}(f_K)\left(1 - \frac{K}{N}\right)^{-2}, \tag{9}$$

and the well-established UEV,

$$\hat{\sigma}^2 = \frac{\widehat{\mathcal{E}}(f_K)}{N - K}. \tag{10}$$

For these indices in neural network modeling, some properties have been investigated under key assumptions, but statistical optimality has not been made clear. These statistics are often used as crude estimates of the generalization error in nonlinear models, because correcting the statistics for nonlinearity requires much computation and the fulfillment of regularity conditions that are often violated by these models (Moody 1992). As outlined earlier, for smoothing operators, the notion of degrees of freedom has a long history, and in the GCV criterion model complexity is measured by the trace of the smoothing matrix. In contrast, little work has been done on neural networks and models of the form (1), and practitioners still measure the dimensionality of the model complexity by $K = W$, that is, the total number W of the weights in the model (1). Both theoretical and empirical studies (see, e.g., Bartlett 1998; De Veaux et al. 1998; Lawrence et al. 1996, 1997) support neural network modeling in which the number of sample data points N is (even much) smaller than the number of the weights W in the selected model. In this case, as $W > N$, both the FPE and the UEV assume negative values, and the unbiased estimates of the error variance also may assume negative values, and this does not make sense. Then the problem is to define K when model selection criteria are applied in neural network modeling.

2.2 The Size of the Weights Is More Important Than the Size of the Network

Some recent results for large networks (see Bartlett 1998) prove that the generalization performance of an MLP depends more on the size of the weights than on the size of the network (namely, on the number of adaptive parameters). Here an important role is played by the quantity $\|\mathbf{c}\|_1 = \sum_{k=0}^{p} |c_p|$, that is, by the sum of the values of the absolute weights between the hidden layer and the output.

Let us introduce the concept of *misclassification error with margin* γ. Let $\mathcal{X}_1$ and $\mathcal{X}_2$ be two populations in $\mathbb{R}^m$ and set $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$; for each $\mathbf{x} \in \mathcal{X}$ with $y \in \{-1, +1\}$, let $y = +1$ if $\mathbf{x}$ comes from $\mathcal{X}_1$ and $y = -1$ if $\mathbf{x}$ comes from $\mathcal{X}_2$. Finally, let P denote a probability distribution on $\mathcal{X} \times \{-1, +1\}$. Let $f : \mathcal{X} \to \mathbb{R}$ be a discriminant function of type (1) such that $\mathbf{x}$ is assigned to $\mathcal{X}_1$ if $f(\mathbf{x}) > 0$ and to $\mathcal{X}_2$ if $f(\mathbf{x}) < 0$. The misclassification error probability is given by

$$P\big\{\mathrm{sgn}[f(\mathbf{x})] \neq y\big\},$$

where $\mathrm{sgn}(u) = 1$ for $u > 0$ and $\mathrm{sgn}(u) = -1$ for $u < 0$ [if $u = 0$, then we assume that $\mathrm{sgn}(u) = 0$]. Given $(\mathbf{x}, y)$, the function f correctly classifies the point $\mathbf{x}$ if and only if $y \cdot f(\mathbf{x}) > 0$; more generally, the function f correctly classifies the point $\mathbf{x}$ with margin $\gamma > 0$ if and only if $y \cdot f(\mathbf{x}) \geq \gamma$. Given $\mathcal{L} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where $y_n = 1$ if $\mathbf{x}_n$ comes from $\mathcal{X}_1$ and $y_n = -1$ if $\mathbf{x}_n$ comes from $\mathcal{X}_2$, with $n = 1, \ldots, N$, we introduce the *misclassification error with margin* γ,

$$\widehat{\mathcal{E}}_\gamma(f, \mathcal{L}) = \frac{1}{N}\#\{n : y_n f(\mathbf{x}_n) < \gamma\}, \tag{11}$$

where $\#\{\cdot\}$ denotes the number of elements in the set $\{\cdot\}$, which is the proportion of the number of cases that are not correctly classified with margin γ by f. If $\mathcal{F}_p$ is the set of functions like (1) and for given constant $C \geq 1$ we consider only those $\mathbf{c}$ for which

$$\|\mathbf{c}\|_1 = \sum_{k=1}^{p} |c_k| \leq C,$$

then we have the following results.

Theorem 1 (Bartlett 1998). Let P be a probability distribution on $\mathcal{X} \times \{-1, +1\}$, $0 < \gamma \leq 1$ and $0 < \eta \leq 1/2$. Let $\mathcal{F}_p$ be the set of functions $f(\mathbf{x})$ like (1) such that $\sum_k |c_k| \leq C$ with $C \geq 1$. If the training set $\mathcal{L}$ is a sample of size N and has $\{-1, +1\}$-valued targets (i.e., the true values of the response), then with probability at least $1 - \eta$, for each $f \in \mathcal{F}_p$,

$$\mathcal{E}(f) \leq \widehat{\mathcal{E}}_\gamma(f, \mathcal{L}) + \epsilon(\gamma, N, \eta), \tag{12}$$

where for some α, a universal constant,

$$\epsilon(\gamma, N, \eta) = \sqrt{\frac{\alpha}{N}\left(\frac{C^2 m}{\gamma^2} \ln\left(\frac{C}{\gamma}\right) \ln^2 N - \ln \eta\right)}. \tag{13}$$

The quantity $\epsilon(\gamma, N, \eta)$ is called the *confidence interval*. Equation (12) shows that the error is bounded by the sum of the empirical error with margin γ and by a quantity depending on $\|\mathbf{c}\|_1$ through C but not on the number of weights. The proof of this theorem is based on a quantity called the *fat-shattering dimension*, which was introduced by Kearns and Schapire (1990) (see also Fine 1999).

An important consequence is that the quantity $\|\mathbf{c}\|_1$ characterizes the property of generalization of a neural network. If $f_{p_1}(\mathbf{x})$ and $f_{p_2}(\mathbf{x})$ are two different models, with $p_1 < p_2$, then they will have the same confidence interval $\epsilon(\gamma, N, \eta)$ in (12) if $\|\mathbf{c}^{(1)}\|_1 = \|\mathbf{c}^{(2)}\|_1$, where $\mathbf{c}^{(1)}$ and $\mathbf{c}^{(2)}$ denote the weights between the hidden layer and the output layer of f_{p_1} and f_{p_2}. Thus a network f_1 with a norm $\|\mathbf{c}\|_1$ and a large number of nodes (with small absolute values of the output layer weights) can be well approximated by a network f_2 with the same norm $\|\mathbf{c}\|_1$ but fewer nodes when the difference between $\widehat{\mathcal{E}}_\gamma(f_1, \mathcal{L})$ and $\widehat{\mathcal{E}}_\gamma(f_2, \mathcal{L})$ is negligible. This approximation has negligible consequences on classification by a positive margin γ. From a practical point of view, the foregoing results justify the implementation of ad hoc strategies in the training algorithm, called *regularization methods*, which try to shrink the size of the weights. Indeed, large weights cause the sigmoids to saturate, and this leads to quite irregular error surfaces. An important regularization technique is the *weight decay*, which adds a penalty term on the error function (5), so that the quantity

$$\widehat{\mathcal{E}}^*(f; \mathcal{L}) = \widehat{\mathcal{E}}(f; \mathcal{L}) + \lambda \sum w_i^2 \tag{14}$$

is minimized during the learning process, where the sum on the right side is over all weights of the network (see, e.g., Bishop 1995). Even if Bartlett's results reported in this section suggest a penalty that depends on the absolute values of the weights (a kind of penalty that is at the heart of Tibshirani's LASSO; see Tibshirani 1996), the weight decay regularization technique seems the most used practically for both computational reasons (it is implemented in many specialized packages) and theoretical justifications (from a Bayesian perspective, the minimization of a cost function with a penalty that depends on the squares of the weights correspond to the minimization of a likelihood with a multinormal prior distribution of the weights, whereas a cost function depending on the absolute value of the weight corresponds to the maximization of a likelihood with a Laplacian prior distribution of the weights). The parameter λ is the *decay constant* (or *smoothing* or *shrinkage* parameter), and suitable values are usually obtained by trial and error, by cross-validation, or by using GCV.

Another simple type of regularization is *early stopping*, discussed previously. If the starting values are small in magnitude and the weights increase as the learning process is carried out, then stopping the training before convergence constrains the weights to remain small.

2.3 Discussion

One of the main consequences of Theorem 1 is that we should not apply to the neural framework the paradigm peculiar to linear models according to which the number of parameters should be less (or even substantially less) than the number of sample values. This consequence implies a deeper look at the role of the weights in the network. We remark that the confidence interval (13) is based only on quantities involving the weights between the hidden layer and the output layer. (In contrast, the weights between the input layer and the hidden layer do not appear.) This suggests that the input-to-hidden set of weights and the hidden-to output weights play different roles. We explore this issue further in Section 3. Besides, because the constant $\|\mathbf{c}\|_1$ is easily computable, it is interesting to analyze its relationship to the learning error $\widehat{\mathcal{E}}(f, \mathcal{L})$ and the test error $\widehat{\mathcal{E}}(f, \mathcal{T})$. We explore this issue further in Section 5.

3. GEOMETRIC PROPERTIES OF THE SIGMOIDAL FUNCTIONS

In this section we investigate some geometrical properties of the MLP. First, we recall some ideas given by Ingrassia (1999) for discrimination problems; then we prove analogous results for regression. For simplicity, throughout this section and in the next one, without loss of generality, we assume that $b_1 = \cdots = b_p = 0$ and $c_0 = 0$, so that the function $f(\mathbf{x})$ is given by

$$f(\mathbf{x}) = \sum_{k=1}^{p} c_k \tau(\mathbf{a}_k' \mathbf{x}). \tag{15}$$

In addition, we assume that the sigmoidal function $\tau(\cdot)$ is analytic; that is, it can be represented by a power series on some interval $(-r, r)$, where r may be $+\infty$. The hyperbolic tangent $\tau(z) = \tanh(z)$ or the logistic function $\tau(z) = (1 + \mathrm{e}^{-z})^{-1}$ are examples of analytic sigmoidal functions. We point out that the function f is a combination of certain transformations of the input data: (a) a nonlinear projection from $\mathbb{R}^m$ to $\mathbb{R}^p$ given by the sigmoidal function τ, that is, $\mathbf{x} \to \tau(\mathbf{a}_1'\mathbf{x}), \ldots, \tau(\mathbf{a}_p'\mathbf{x})$, and (b) a linear transformation from $\mathbb{R}^p$ to $\mathbb{R}$ according to $c_1, \ldots, c_p$. The results in this section are based on the following theorem (see, e.g., Rudin 1966).

Theorem 2. Let g be analytic and not identically 0 in the interval $(-r, r)$, with $r > 0$. Then the set of the 0's of g in $(-r, r)$ is at most countable.

Let $\mathbf{x}_1 = (x_{11}, \ldots, x_{1m}), \ldots, \mathbf{x}_p = (x_{p1}, \ldots, x_{pm})$ be p points of $\mathbb{R}^m$, with $p > m$; evidently, these points are linearly dependent as $p > m$. Let $\mathbf{A} = (a_{ij})$ be a $p \times m$ matrix with values in some hypercube $[-u, u]^{mp}$ for some $u > 0$, where we use the notation $\mathbf{A} \in [-u, u]^{mp}$ to mean that all of the entries of $\mathbf{A}$ are in $[-u, u]$; thus the points $\mathbf{A}\mathbf{x}_1, \ldots, \mathbf{A}\mathbf{x}_p$ are linearly dependent, because they are obtained by a linear transformation acting on $\mathbf{x}_1, \ldots, \mathbf{x}_p$. However, for $u = 1/m$ the points $\tau(\mathbf{A}\mathbf{x}_1), \ldots, \tau(\mathbf{A}\mathbf{x}_p)$, where

$$\begin{aligned}\tau(\mathbf{A}\mathbf{x}_i) &= \left(\tau\left(\sum_{j=1}^{m} a_{1j} x_{ij}\right), \ldots, \tau\left(\sum_{j=1}^{m} a_{pj} x_{ij}\right)\right) \\ &= \left(\tau(\mathbf{a}_1'\mathbf{x}_i), \ldots, \tau(\mathbf{a}_p'\mathbf{x}_i)\right), \qquad i = 1, \ldots, p,\end{aligned}$$

are *linearly independent* for almost all matrices $\mathbf{A} \in [-u, u]^{mp}$, according to the following theorem.

Theorem 3 (Ingrassia 1999). Let $\mathbf{x}_1, \ldots, \mathbf{x}_p$ be p distinct points in $(-r, r)^m$ with $\mathbf{x}_h \neq \mathbf{0}$ $(h = 1, \ldots, p)$, and let $\mathbf{A} = (a_{ij}) \in [-u, u]^{mp}$ be a $p \times m$ matrix, with $u = 1/m$. Let τ be a sigmoidal analytic function on $(-r, r)$, with $r > 0$. Then the points $\tau(\mathbf{A}\mathbf{x}_1), \ldots, \tau(\mathbf{A}\mathbf{x}_p) \in \mathbb{R}^p$ are linearly independent for almost all matrices $\mathbf{A} = (a_{ij}) \in [-u, u]^{mp}$.

This result proves that, given $N > m$ points $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^m$, the transformed points $\tau(\mathbf{A}\mathbf{x}_1), \ldots, \tau(\mathbf{A}\mathbf{x}_N)$ generate an *overspace* of dimension $p > m$ if the matrix $\mathbf{A}$ satisfies suitable conditions. In particular, the largest overspace is attained when $p = N$, that is, when the hidden layer has as many units as the number of points in the learning set. Moreover, it shows why neural networks have been shown to work well in presence of multicollinearity. On this topic De Veaux and Ungar (1994) presented a case study in which the temperature of a flow is measured by six different devices at various places in a production process. Even though the inputs are highly correlated, a better prediction of the response is gained using a weighted combination of all six predictors rather than choosing the single best measurement having the highest correlation with the response.

Theorem 3 introduces the concept of $\mathcal{F}_p$-linearization. We say that a learning set $\mathcal{L}$ is *$\mathcal{F}_p$-linearizable* or *linearized by $\mathcal{F}_p$* if there exists $\mathbf{A} \in [-1/m, 1/m]^{mp}$ such that the empirical error $\widehat{\mathcal{E}}(f, \mathcal{L})$ is 0, in other words, if there exists a hyperplane in $\mathbb{R}^p$ that correctly separates the points of the learning set (discrimination) or that perfectly interpolates them (regression). It is obvious that if $\mathcal{L}$ is $\mathcal{F}_p$-linearizable, then it is also $\mathcal{F}_{p+1}$-linearizable; the counterpart is not true in general, as a simple analysis of the system (16) shows. Thus the family of learning sets that can be linearized by $\mathcal{F}_p$ is a strict subset of the learning sets, which can be linearized by $\mathcal{F}_{p+1}$. The smallest value p_c such that $\mathcal{L}$ is $\mathcal{F}_p$-linearizable is here called the *critical dimension of linearization*. The next result, which generalizes theorem 8 of Ingrassia (1999), gives an upper bound on p_c.

Theorem 4. Let $\mathcal{L}$ be a given learning set and $f = \sum_{k=1}^{p} c_k \times \tau(\mathbf{a}_k'\mathbf{x})$. If $p = N$, then the error $\widehat{\mathcal{E}}(f, \mathcal{L})$ is 0 for almost all matrices $\mathbf{A} \in [-1/m, 1/m]^{mp}$.

Proof. Theorem 3 implies that the points $\tau(\mathbf{Ax}_1), \ldots, \tau(\mathbf{Ax}_N)$ are linearly independent for almost all matrices $\mathbf{A} \in [-1/m, 1/m]$ for $p \geq N$. In particular, if $p = N$, then these points generate $\mathbb{R}^N$, and thus the system

$$\begin{array}{llll} c_1\tau(\mathbf{a}_1'\mathbf{x}_1) & + \cdots + c_N\tau(\mathbf{a}_N'\mathbf{x}_1) & = y_1 & \\ \vdots & + \ \vdots \ + \vdots & = \vdots & \\ c_1\tau(\mathbf{a}_1'\mathbf{x}_n) & + \cdots + c_N\tau(\mathbf{a}_N'\mathbf{x}_n) & = y_n & \qquad (16) \\ \vdots & + \ \vdots \ + \vdots & = \vdots & \\ c_1\tau(\mathbf{a}_1'\mathbf{x}_N) & + \cdots + c_N\tau(\mathbf{a}_N'\mathbf{x}_N) & = y_N & \end{array}$$

has a unique solution.

The upper bound on p_c given earlier looks too large, but it refers to the worst-case situation. In neural modeling, given a learning set $\mathcal{L}$ of N sample data, the correct question seems to not be "what is the largest network we can train by $\mathcal{L}$ (if any)," but rather "what is the suitable size—namely, the dimension p of the space $\mathbb{R}^p$—necessary for fitting the input–output unknown dependence $\phi = \mathbb{E}[Y|\mathbf{X}]$." This dimension p essentially depends more on the geometry of the data, and this explains why neural models may be successfully applied as virtual sensors when the predictors exhibit a high degree of multicollinearity. As a matter of fact, the hidden units break the multicollinearity and exploit the contribution of each single predictor. This is the reason why in modeling virtual sensors, the optimal size p of the hidden layer often may be greater than the number m of predictors.

Finally, we remark that Theorem 1 and Theorems 3 and 4 provide insight into the penalization term $\lambda \sum w_i^2$ in (14). Indeed, we can split the quantity $\sum_i w_i^2$ into two components linking the weights of the input-to-hidden layer and of the hidden-to-output layer,

$$\sum w_i^2 = \sum_{j=1}^{m} \sum_{i=1}^{p} a_{ij}^2 + \sum_{i=1}^{p} c_i^2 = \mathrm{tr}(\mathbf{AA}') + \mathbf{c}'\mathbf{c}. \qquad (17)$$

Small values of the a_{ij}'s constrain the quantities $\mathbf{a}_n'\mathbf{x}_n$ ($n = 1, \ldots, N$) in the nonlinear range of the sigmoidal function $\tau(\cdot)$, whereas small values of the c_i's help prevent overfitting. This also provides insight into two different choices of the decay constant λ for each term in (17) proposed by some authors on the basis of empirical studies (see, e.g., Bishop 1995).

4. EQUIVALENT NUMBER OF DEGREES OF FREEDOM IN MLP

Many authors have remarked that the number of degrees of freedom is often much smaller than the number of parameters involved in the model. For example, in presence of regularization, what Moody (1992) called the *effective number of parameters* and MacKay (1992) called the *number of good parameters measurements* does not equal the number of weights in the model. It is less than W and depends on the size of the regularization term. Recent studies regarding richly parameterized models have proposed various definitions of degrees of freedom that depend not on the number of parameters in the models but rather on, for example, the sum of the sensitivity of each fitted value to perturbation in the corresponding observed value (Ye 1998) or on the properties of the space in which the fitted values lie (Hodges and Sargent 2001). In the present study we focus on neural networks and models of the form (1) and propose some easy corrections to the model selection criteria that are automatically computed by most common data-mining software.

We recall that in the usual linear model,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \qquad (18)$$

where $\mathbf{y}$ is $N \times 1$, $\mathbf{X}$ is $N \times m$, $\boldsymbol{\beta}$ is $m \times 1$, and $\boldsymbol{\varepsilon}$ is $N \times 1$, with $\mathrm{cov}(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}_N$, $\mathbf{I}_N$ being the N-dimensional identity matrix. The number of degrees of freedom of the model is the dimension of the space in which the fitted values $\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ lie and is given by

$$K = \mathrm{tr}\{\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\} = \mathrm{tr}\{(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}\} = \mathrm{tr}(\mathbf{I}_m) = m. \qquad (19)$$

Here we assume that $m \leq N$ and that $\mathbf{X}$ is of full rank; however, even if $\mathbf{X}$ is not of full rank, then the definition of K is fine provided that one uses the generalized inverses. When the constant term is included in the model, $\boldsymbol{\beta}$ is $(m+1) \times 1$, and the matrix $\mathbf{X}$ has $m + 1$ columns, the first being an N-dimensional unit vector; in this case $K = m + 1$. As noted by Hodges and Sargent (2001), defining degrees of freedom in this way avoids quandaries created by counting parameters. We can also consider the principal component (PC) regression, where now for simplicity $\mathbf{y}$ and $\mathbf{X}$ are measured about their mean. Let $\mathbf{A}$ be the $m \times m$ matrix whose kth column is the kth eigenvector of $\bar{\mathbf{X}}'\bar{\mathbf{X}}$, where $\bar{\mathbf{X}}$ is the centered version of $\mathbf{X}$, so that the values of the PCs for each observation are given by $\mathbf{Z} = \mathbf{XA}$. Because $\mathbf{A}$ is orthogonal, $\mathbf{X}\boldsymbol{\beta}$ can be rewritten as $\mathbf{XAA}'\boldsymbol{\beta} = \mathbf{Z}\boldsymbol{\gamma}$, where $\mathbf{Z} = \mathbf{XA}$ and $\boldsymbol{\gamma} = \mathbf{A}'\boldsymbol{\beta}$, so that (18) can therefore be written as

$$\mathbf{y} = \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}, \qquad (20)$$

which simply replaces the predictor variables by their PCs in the regression model. PC regression can be defined as the use of the model (20) or the reduced model,

$$\mathbf{y} = \mathbf{Z}_p\boldsymbol{\gamma}_p + \boldsymbol{\varepsilon}_p, \qquad (21)$$

where $\boldsymbol{\gamma}_p$ is a vector of p elements that are a subset of $\boldsymbol{\gamma}$, $\mathbf{Z}_p$ is an $N \times p$ matrix whose columns are the corresponding subset of columns of $\mathbf{Z}$, and $\boldsymbol{\varepsilon}_p$ is the appropriate error term (see, Jolliffe 1986, sec. 8.1). The degrees of freedom in model (20) that results from defining the projection of the centered matrix $\bar{\mathbf{X}}$ in the space spanned by the m PCs and then $K = p$.

Let us switch to nonlinear models of the form (15). For a given $p \times m$ matrix $\mathbf{A}$, let $\mathbf{T}$ be the $N \times p$ matrix with rows $\tau(\mathbf{Ax}_1)', \ldots, \tau(\mathbf{Ax}_N)'$, with $p \leq N$. According to Theorems 3 and 4, the matrix $\mathbf{T}$ has rank p (and then it is nonsingular) for almost all matrices $\mathbf{A} \in [-1/m, 1/m]^{mp}$. The empirical error $\widehat{\mathcal{E}}(f, \mathcal{L})$ can be written as

$$\begin{aligned} \widehat{\mathcal{E}}(f, \mathcal{L}) = \sum_{(\mathbf{x}_n, y_n) \in \mathcal{L}} (y_n - f(\mathbf{x}_n))^2 &= \sum_{(\mathbf{x}_n, y_n) \in \mathcal{L}} \big(y_n - \mathbf{c}'\tau(\mathbf{Ax}_n)\big) \\ &= (\mathbf{y} - \mathbf{Tc})'(\mathbf{y} - \mathbf{Tc}) \\ &= \mathbf{y}'\mathbf{y} - 2\mathbf{c}'\mathbf{T}'\mathbf{y} + \mathbf{c}'\mathbf{T}'\mathbf{Tc}, \end{aligned}$$

and for any fixed matrix $\mathbf{A}$, the error $\widehat{\mathcal{E}}(f,\mathcal{L})$ attains its minimum when

$$\frac{\partial\widehat{\mathcal{E}}(f,\mathcal{L})}{\partial\mathbf{c}} = -2\mathbf{T}'\mathbf{y} + 2\mathbf{T}'\mathbf{T}\mathbf{c} = \mathbf{0},$$

which implies that $\mathbf{c} = (\mathbf{T}'\mathbf{T})^{-1}\mathbf{T}'\mathbf{y}$.

Thus the matrix

$$\mathbf{H} = \mathbf{T}(\mathbf{T}'\mathbf{T})^{-1}\mathbf{T}' \tag{22}$$

is a projection matrix because $\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}$ and $\mathbf{H}$ is symmetric, and positive semidefinite, idempotent, and it results in

$$\begin{aligned}\operatorname{rank}(\mathbf{H}) &= \operatorname{tr}(\mathbf{H}) = \operatorname{tr}\{\mathbf{T}(\mathbf{T}'\mathbf{T})^{-1}\mathbf{T}'\} \\ &= \operatorname{tr}\{(\mathbf{T}'\mathbf{T})^{-1}\mathbf{T}'\mathbf{T}\} = p,\end{aligned} \tag{23}$$

so that $\hat{\mathbf{y}}$ lies in the space $\mathbb{R}^p$ and thus to the model $f(\mathbf{x}) = \sum_{k=1}^{p} c_k\tau(\mathbf{a}_k,\mathbf{x})$ should be given p *equivalent number of degrees of freedom* (edf), according to Hastie and Tibshirani (1990, sec. 2.8).

This notion of edf applies to other kinds of networks, including the *radial basis function network* (RBFN) (see, e.g., Bishop 1995, chap. 5) and *projection pursuit regression* (PPR) (see, e.g., Hastie et al. 2001, chap. 11).

We remark that the stopped training minimization procedure imposes some constraints on the estimates of the weights, but they are not explicit, so that we can set $K = \operatorname{tr}(\mathbf{H}) = p$; in contrast, the other common procedure is based on the weight decay strategy, and it imposes explicit constraints on the minimization of the error function. In this case the error function (14) is

$$\begin{aligned}\widehat{\mathcal{E}}^*(f;\mathcal{L}) &= \widehat{\mathcal{E}}(f;\mathcal{L}) + \lambda\sum w_i^2 \\ &= (\mathbf{y}-\mathbf{T}\mathbf{c})'(\mathbf{y}-\mathbf{T}\mathbf{c}) + \lambda\{\operatorname{tr}(\mathbf{A}\mathbf{A}') + \mathbf{c}'\mathbf{c}\} \\ &= \mathbf{y}'\mathbf{y} - 2\mathbf{c}'\mathbf{T}'\mathbf{y} + \mathbf{c}'\mathbf{T}'\mathbf{T}\mathbf{c} + \lambda\{\operatorname{tr}(\mathbf{A}\mathbf{A}') + \mathbf{c}'\mathbf{c}\} \\ &= \mathbf{y}'\mathbf{y} - 2\mathbf{c}'\mathbf{T}'\mathbf{y} + \mathbf{c}'\mathbf{T}'\mathbf{T}\mathbf{c} + \lambda\operatorname{tr}(\mathbf{A}\mathbf{A}') + \lambda\mathbf{c}'\mathbf{c},\end{aligned}$$

and it attains its minimum when

$$\begin{aligned}\frac{\partial\widehat{\mathcal{E}}^*(f;\mathcal{L})}{\partial\mathbf{c}} &= -2\mathbf{T}'\mathbf{y} + 2\mathbf{T}'\mathbf{T}\mathbf{c} + 2\lambda\mathbf{c} \\ &= -2\mathbf{T}'\mathbf{y} + 2(\mathbf{T}'\mathbf{T} + \lambda\mathbf{I}_p)\mathbf{c} = \mathbf{0},\end{aligned}$$

that is, for

$$\mathbf{c} = (\mathbf{T}'\mathbf{T} + \lambda\mathbf{I}_p)^{-1}\mathbf{T}'\mathbf{y}.$$

Thus we get

$$\hat{\mathbf{y}} = \mathbf{T}(\mathbf{T}'\mathbf{T} + \lambda\mathbf{I}_p)^{-1}\mathbf{T}'\mathbf{y} = \mathbf{H}_\lambda\mathbf{y},$$

where $\mathbf{H}_\lambda = \mathbf{T}(\mathbf{T}'\mathbf{T} + \lambda\mathbf{I}_p)^{-1}\mathbf{T}'$.

Thus the equivalent degrees of freedom in this case are

$$\begin{aligned}\operatorname{tr}(\mathbf{H}_\lambda) &= \operatorname{tr}\{\mathbf{T}(\mathbf{T}'\mathbf{T} + \lambda\mathbf{I}_p)^{-1}\mathbf{T}'\} = \operatorname{tr}\{(\mathbf{T}'\mathbf{T} + \lambda\mathbf{I}_p)^{-1}\mathbf{T}'\mathbf{T}\} \\ &= \operatorname{tr}\{(\mathbf{T}'\mathbf{T} + \lambda\mathbf{I}_p)^{-1}(\mathbf{T}'\mathbf{T} + \lambda\mathbf{I}_p - \lambda\mathbf{I}_p)\} \\ &= \operatorname{tr}\{(\mathbf{T}'\mathbf{T} + \lambda\mathbf{I}_p)^{-1}(\mathbf{T}'\mathbf{T} + \lambda\mathbf{I}_p) - (\mathbf{T}'\mathbf{T} + \lambda\mathbf{I}_p)^{-1}\lambda\mathbf{I}_p)\} \\ &= \operatorname{tr}(\mathbf{I}_p) - \operatorname{tr}\{(\mathbf{T}'\mathbf{T} + \lambda\mathbf{I}_p)^{-1}\lambda\mathbf{I}_p\} \\ &= p - \lambda\operatorname{tr}\{(\mathbf{T}'\mathbf{T} + \lambda\mathbf{I}_p)^{-1}\},\end{aligned} \tag{24}$$

which shows that p is decreased by the quantity $\lambda\operatorname{tr}\{(\mathbf{T}'\mathbf{T} + \lambda\mathbf{I}_p)^{-1}\}$. Because $\mathbf{T}'\mathbf{T}$ is positive semidefinite, the p eigenvalues of $\mathbf{T}'\mathbf{T}$, say $l_1,\dots,l_p$, are nonnegative. Thus $(\mathbf{T}'\mathbf{T} + \lambda\mathbf{I}_p)$ has eigenvalues $(l_1+\lambda),\dots,(l_p+\lambda)$, and then the eigenvalues of $(\mathbf{T}'\mathbf{T} + \lambda\mathbf{I}_p)^{-1}$ are $(l_1+\lambda)^{-1},\dots,(l_p+\lambda)^{-1}$. Hence, we have

$$\operatorname{tr}\{(\mathbf{T}'\mathbf{T} + \lambda\mathbf{I}_p)^{-1}\} = \sum_{i=1}^{p}\frac{1}{l_i+\lambda}, \tag{25}$$

and finally, from (24) and (25), we get

$$K = \operatorname{tr}(\mathbf{H}_\lambda) = p - \sum_{i=1}^{p}\frac{\lambda}{l_i+\lambda}. \tag{26}$$

The matrix $\mathbf{H}_\lambda$ is no longer a projection matrix, and thus the edf should be given by $\operatorname{tr}(\mathbf{H}_\lambda)$ rather than p. However, in practice λ always assumes small values, so that p is slightly larger than $\operatorname{tr}(\mathbf{H}_\lambda)$, and for practioners the simple quantity p can be used as a measure of edf in model selection criteria even when the decay strategy is implemented. In contrast, more accurate values could be required in the error variance estimation (10). We present some numerical studies in Section 5.

4.1 Relationship With Other Approaches

Recently, for richly parameterized models, Hodges and Sargent (2001) have introduced a new measure of degrees of freedom, say ρ, based on a property of the vector space into which the outcomes y_i are projected to give the fitted values $\hat{y}_i$. For the cases where both our definition of equivalent degrees of freedom and ρ are defined, they are equal if the model parameters are estimated by the minimization of the sum of squares error function. For example, this is true for RBFNs,

$$f(\mathbf{x}) = \sum_{k=1}^{p} c_k\phi(\mathbf{x},\mathbf{a}_k) + c_0, \tag{27}$$

where the second-layer parameters are estimated by least squares and the basis functions parameters are chosen by forward selection of the input vector or k-means cluster analysis. In (27) the nonlinear transfer function $\phi(\cdot)$ is a Gaussian radial basis. Without loss of generality, if we suppress the constant term in the linear combination, then (27) can be expressed in matrix form as

$$\boldsymbol{\Phi}\mathbf{c} = \hat{\mathbf{y}} \tag{28}$$

where $\boldsymbol{\Phi}$ is an $N\times p$ matrix with generic element $\phi_{nk} = \phi(\mathbf{x}_n,\mathbf{a}_k)$, $n = 1,\dots,N$, $k = 1,\dots,p$, and it satisfies the assumptions of the Hodges and Sargent approach. The parameters $\mathbf{c}$ are given by $\mathbf{c} = (\boldsymbol{\Phi}'\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}'\mathbf{y}$. The matrix

$$\mathbf{H} = \boldsymbol{\Phi}(\boldsymbol{\Phi}'\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}'$$

is a real projection matrix, because $\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}$ and $\mathbf{H}$ is symmetric, positive semidefinite, and idempotent and has rank p (or $p+1$ if we include the constant term in the model). Because $\operatorname{tr}(\mathbf{H}) = p$ (or $p+1$ if we include the constant term), our notion of degrees of freedom is equal to ρ. If the Y variable is normally distributed with known variance σ^2, our notion of edf is equal to the generalized degrees of freedom (GDF) introduced by Ye (1998). Indeed, Hodges and Sargent (2001) proved that the two quantities ρ and GDF are equal when both measures are defined. When the variance σ^2 is not known, as usually happens in practice, Ye's GDF must be computed by simulations, whereas our definition of edf is still explicit.

With a weight-decay cost function (and a penalty term λ chosen by trial and error), we have

$$\rho = p - \lambda\,\mathrm{tr}(\boldsymbol{\Phi}'\boldsymbol{\Phi})^{-1} = p - \lambda \sum_{k=1}^{p} \frac{1}{\lambda + l_k},$$

where the l_k's are the eigenvalues of the matrix $\boldsymbol{\Phi}'\boldsymbol{\Phi}$, which is a result formally equivalent to (26). Then ρ is smaller than p, and its value depends on λ. If the parameter λ is estimated by cross-validation, then the model cannot be reexpressed in a linear form, and Hodges and Sargent's degrees of freedom are not defined, whereas our measure is a good explicit approximation.

In PPR, the nonlinear function is a smoother, and the vectors $\mathbf{a}'_k$ are constrained to be unit vectors. The model $f(\mathbf{x}) = \sum_{k=1}^{p} c_k \tau_k(\mathbf{a}'_k, \mathbf{x}) + c_0$ results in a linear combination of ridge functions. It is built in a forward-stagewise manner, adding a pair $(\tau_k, \mathbf{a}'_k)$ at each stage and then estimating the coefficient c_k. Given the $(N \times p)$ basis function matrix $\mathbf{T}$ with generic (n, k) element $\tau_{nk} = \tau_k(\mathbf{x}_n, \mathbf{a}_k)$, the matrix $\mathbf{P} = \mathbf{T}(\mathbf{T}'\mathbf{T})^{-1}\mathbf{T}'$ is a real projection matrix, and thus our definition of degrees of freedom is equal to $\mathrm{tr}(\mathbf{P})$.

As far as smoothers are concerned, for models of the form $\mathbf{S}\mathbf{y} = \hat{\mathbf{y}}$, where $\mathbf{S}$ is an $n \times n$ smoothing matrix (with rank n and with $\mathbf{S}'\mathbf{S} < \mathbf{S}$), our definition does not apply, because it is defined for projection models. For smoothers that can be expressed as $\mathbf{P}\mathbf{y} = \hat{\mathbf{y}}$, where $\mathbf{P}$ is a linear operator, with rank p (or $p + 1$ if the constant term is included in the model) and with $\mathbf{P}'\mathbf{P} = \mathbf{P}$, our definition of degrees of freedom is equal to $\mathrm{tr}(\mathbf{P})$. So it is equal to Hastie and Tibshirani's first definition (they give three definitions of degrees of freedom, depending on the purpose for which they are used) and to Hodges and Sargent's definition.

5. NUMERICAL RESULTS

In this section we present some numerical results to investigate the behavior of Bartlett's constant $\|\mathbf{c}\|_1$ and its relation to the learning error $\widehat{\mathcal{E}}(f_p, \mathcal{L})$, the test error $\widehat{\mathcal{E}}(f_p, \mathcal{T})$, and number of edf $K = p + 1$ in model selection criteria. Moreover, we analyze the unbiased estimate of the variance (10) considering both $K = p + 1$ and $K = \mathrm{tr}(\mathbf{H}_\lambda)$ as the edf. For this purpose, we consider three cases involving both real and simulated data.

The first dataset is the *polymer dataset* modeled by De Veaux et al. (1998) by means of an MLP with 18 neurons in the hidden layer. This dataset contains 61 units with 10 predictor variables concerning measurements of controlled variables in a polymer process plant and 4 responses concerning the outputs of the plant. We choose the 11th variable (i.e., the first of the four response variables) as the response variable y. These data, which lie in the interval [.1, .9], are particularly good for testing the robustness of nonlinear methods for irregularly spaced data. De Veaux, Psichogios, and Ungar (1993) showed that in this case neural networks are superior to multivariate adaptive splines regression and both are superior to linear regression. The data exhibit quite a large degree of multicollinearity, as can be seen by an analysis of the variance inflation factor $\mathrm{VIF}_j = (1 - R_j^2)^{-1}$, where R_j^2 is the coefficient of determination when the jth variable X_j is regressed on the remaining $p - 1$ variables. If X_j is nearly orthogonal to the remaining predictor variables, then R_j^2 is small and then VIF_j is close to unity, whereas if X_j is nearly linearly dependent on some subset of the remaining variables, then R_j^2 is near unity and VIF_j is large. Table 1 lists the VIF values of the 10 predictors. In general, variance inflation factors larger than 10 imply serious problems with multicollinearity; here this is true for variables X_3, X_4, X_5, X_6, and X_7. Following De Veaux et al. (1998), we fix 50 units for the learning set and 11 units for the test set. We then consider 100 different samples with different units for the training set and the test set (these sets all have 50 and 11 units). We consider neural networks with increasing numbers of hidden units from $p = 2$ to $p = 25$. For each p, we train the network 1,000 times, varying the samples and the initial weights; we adopt either the weight decay or the early-stopping regularization technique. The distribution of the learning and test errors and the distribution of $\|\mathbf{c}\|_1$ versus the number p of neurons in the hidden layer are plotted in Figure 1 using boxplots, for both values obtained with stopped training and values obtained with weight decay.

Table 1. VIF Values for Polymer Data

	Variables									
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
VIF	2.82	.83	25.03	100.25	49.73	95.90	57.99	1.65	3.75	.55

Figures 1(a)–1(d) show that for the errors to be rather stable across simulations, the network must be large enough; in particular, Figures 1(a) and 1(c) refer to training and test error with weight decay, and Figures 1(b) and 1(d) refer to training and test errors with stopped training. For small values of p, results are very unstable, and the algorithms are influenced mostly by local minima. The fit, as measured by the median value of the learning errors, increases as the complexity rises, and stabilizes (i.e., the spread of the boxplots becomes small) after a certain threshold (given by about 10 or 11 hidden units). The different behavior of boxplots in Figures 1(c) and 1(d) is due to the fact that test observations are used for controlling overfitting in the stopped training criterion, whereas they are independent of training with the weight decay and can be used for measuring the generalization performance of the network only in this second case.

Figures 1(e) and 1(f) show that the median values of $\|\mathbf{c}\|_1$ are in practice linearly related to our measure p; however, the quantity $\|\mathbf{c}\|_1$ is too unstable across simulations to be a good complexity measure. We remark that such median values of the training errors and of $\|\mathbf{c}\|_1$ versus p have a similar trend for both weight decay and stopped training. Table 2 gives for each p the mean values of the training error $\widehat{\mathcal{E}}(f_p; \mathcal{L})$, the test error $\widehat{\mathcal{E}}(f_p; \mathcal{T})$, the L_1 norm $\|\mathbf{c}\|_1$, and the model selection criteria AIC, BIC/SBC, GCV, and FPE, computed with $K = p + 1$ because here we implemented the stopped training. We remark that analogous results have been obtained using the weight decay strategy, where in model selection criteria $K = \mathrm{tr}(\mathbf{H}_\lambda)$ can be approximated by $K = p + 1$ because here we are interested

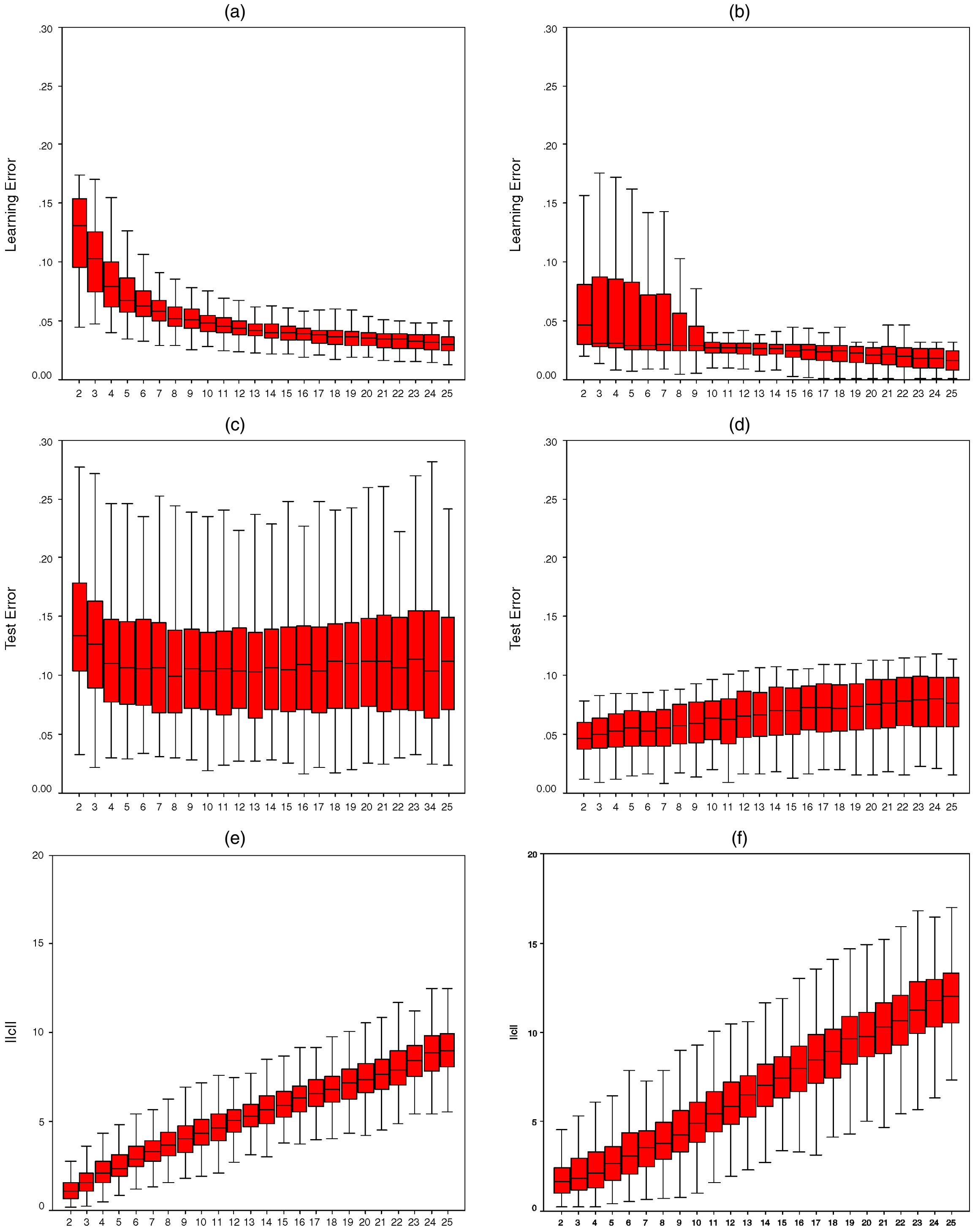

Figure 1. Polymer Dataset: Distribution of (a) Training Errors With Weight Decay, (b) Training Errors With Stopped Training, (c) Test Errors With Weight Decay, (d) Test Errors With Stopped Training, and (e) $\|\mathbf{c}\|_1$ With Weight Decay, (f) $\|\mathbf{c}\|_1$ With Stopped Training versus p.

in the rank rather than in the size of the indices AIC, BIC, and so on.

The numerical results show that there is no unique p that minimizes the different model selection criteria, even if the model selected by the BIC agrees with the smallest value of the test error while the other ones suggest larger models; however, we note that many numerical experiments have shown that the BIC often works well for neural networks, whereas AIC and FPE tend to overfit with neural networks (see Sarle 1999; Kadane and Lazar 2004). For the sake of completeness, Table 3 gives the values for model selection criteria when the number of degrees of freedom is selected equal to W. In this case some indices are useless because they assume negative values, and some others have a high variability and anomalous peaks. FPE is useless because it assumes negative values for more than four hidden units; GCV suggests large models and shows an anomalous peak for the model with four hidden units (which holds a number W near N); this peak and the high variability are not

Table 2. Polymer Dataset: Values of Some Model Selection Criteria for $K = p + 1$

p	$\widehat{\mathcal{E}}(f_p;\mathcal{L})$	$\widehat{\mathcal{E}}(f_p;\mathcal{T})$	$\|\mathbf{c}\|_1$	K	BIC	AIC	FPE	GCV
2	.0540	.0275	1.513	3	1.2280	1.1133	3.0447	3.0557
3	.0381	.0273	1.982	4	**.9570**	.8040	2.2352	2.2496
4	.0384	.0286	2.535	5	1.0426	.8514	2.3444	2.3681
5	.0338	.0282	2.966	6	.9933	.7638	2.1489	2.1803
6	.0363	.0294	3.467	7	1.1441	.8764	2.4067	2.4548
7	.0307	.0296	3.776	8	1.0539	.7480	**2.1187**	**2.1743**
8	.0314	.0313	4.517	9	1.1551	.8109	2.2590	2.3346
9	.0284	.0306	4.913	10	1.1326	.7502	2.1289	2.2176
10	.0330	.0323	5.569	11	1.3603	.9397	2.5779	2.7091
11	.0260	.0312	6.033	12	1.2026	**.7437**	2.1240	2.2538
12	.0283	.0351	6.420	13	1.3648	.8676	2.4105	2.5853
13	.0295	.0353	7.271	14	1.4832	.9478	2.6201	2.8430
14	.0281	.0362	7.353	15	1.5143	.9407	2.6111	2.8693
15	.0264	.0352	8.030	16	1.5297	.9178	2.5629	2.8553
16	.0242	.0367	8.605	17	1.5216	.8715	2.4588	2.7802
17	.0247	.0378	9.003	18	1.6188	.9305	2.6228	3.0133
18	.0221	.0375	9.537	19	1.5844	.8579	2.4547	2.8689
19	.0224	.0394	9.946	20	1.6783	.9135	2.6138	3.1117
20	.0212	.0379	10.671	21	1.6999	.8969	2.5916	3.1466
21	.0221	.0393	11.235	22	1.8205	.9792	2.8396	3.5213
22	.0188	.0382	11.614	23	1.7383	.8588	2.5431	3.2256
23	.0206	.0398	12.060	24	1.9091	.9913	2.9367	3.8159
24	.0175	.0413	12.594	25	1.8237	.8677	2.6281	3.5042
25	.0175	.0425	12.755	26	1.9010	.9068	2.7717	3.7989

NOTE: The minimum values for the different criteria are in bold.

supported by values summarized in the boxplots of Figure 1. Furthermore, the BIC and AIC give strictly increasing values as p grows, so we should select the model with $p = 2$.

Table 3. Polymer Dataset: Values of Some Model Selection Criteria for $K = W$

p	W	BIC	AIC	FPE	GCV
2	25	2.949	1.993	8.100	10.800
3	37	3.539	2.124	12.743	28.167
4	49	4.485	2.611	189.900	4,795.446
5	61	5.296	2.964	−17.038	34.886
6	73	6.308	3.516	−9.709	8.580
7	85	7.078	3.828	−5.918	3.131
8	97	8.040	4.331	−4.910	1.777
9	109	8.878	4.710	−3.825	1.019
10	121	9.967	5.340	−3.970	.817
11	133	10.670	5.584	−2.870	.472
12	145	11.692	6.148	−2.906	.392
13	157	12.672	6.668	−2.851	.322
14	169	13.563	7.101	−2.587	.248
15	181	14.439	7.518	−2.328	.192
16	193	15.292	7.911	−2.058	.148
17	205	16.250	8.410	−2.031	.128
18	217	17.076	8.778	−1.763	.099
19	229	18.031	9.274	−1.746	.087
20	241	18.913	9.697	−1.613	.073
21	253	19.894	10.219	−1.648	.067
22	265	20.672	10.539	−1.378	.051
23	277	21.704	11.111	−1.486	.050
24	289	22.479	11.428	−1.243	.038
25	301	23.417	11.907	−1.224	.035

NOTE: BIC and AIC are strictly increasing with p; moreover, FPE is negative for $p \geq 5$, FPE and GCV present an anomalous peak when W approximates the size of the learning set.

The second dataset concerns a vibration severity chart (see Cavarra, Crupi, Guglielmino, and Ingrassia 2002; Ingrassia and Morlini 2002) for centrifugal pumps in an ethylene system. There are 51 recordings of the overall level of vibration in the horizontal, vertical, and axial directions. Besides the overall vibration, the ratios $\frac{v_{t+1}-v_t}{t+1-t}$, where v_{t+1} is the vibration level at time $t+1$ and v_t is the vibration level at time t for each direction, are also considered input variables. Thus input patterns are vectors in $\mathbb{R}^6$. The response variable is a measure of the status of the centrifugal pump after a period of time, which takes value 0 for the 32 pumps working perfectly, value .5 for the 12 pumps with something wrong, and value 1 for the 11 pumps that are about to be broken down. In other words, the value 0 is associated with the target class low risk, the value .5 with the target class medium risk, and the value 1 with the class high risk. A total of 1,000 different partitions in a training set with 39 recordings and a test set with 12 recordings are randomly generated so that each test set contains 6 observations belonging to class 1 ($y = 0$), 3 observations belonging to class 2 ($y = .5$), and 3 observations belonging to class 3 ($y = 1$). As far as the misclassification error rate is concerned, if $\hat{y} \leq .3$, then the observation is assigned to class 1; if $.3 < \hat{y} \leq .7$, then the observation is assigned to class 2; and if $\hat{y} > .7$, then the observation is assigned to class 3. Figure 2 reports boxplots of the training and the test misclassification errors and the values of $\|\mathbf{c}\|_1$ obtained with stopped training and weight decay according to the same outline of Figure 1. As in Figure 1, values of $\|\mathbf{c}\|_1$ are very unstable across simulations. In this example, values of $\|\mathbf{c}\|_1$ for each number p of hidden units obtained with stopped training have a higher variability around their median than do values obtained with weight decay. However, median values obtained with stopped training and weight decay for each number p are similar. Table 4 gives for each p the mean values of model selection criteria obtained with $K = p + 1$ and the mean value of the validation errors obtained with weight decay. According to the validation error obtained with weight decay, all model selection criteria, with the exception of the BIC, suggest models with a number of hidden units ranging from 8 to 12 (with more than 12 hidden units, values obtained for these criteria start increasing), whereas the BIC selects models with a number p ranging from 5 to 8. However, we remarked that for neural networks, the BIC is preferable over the AIC and the FPE. The

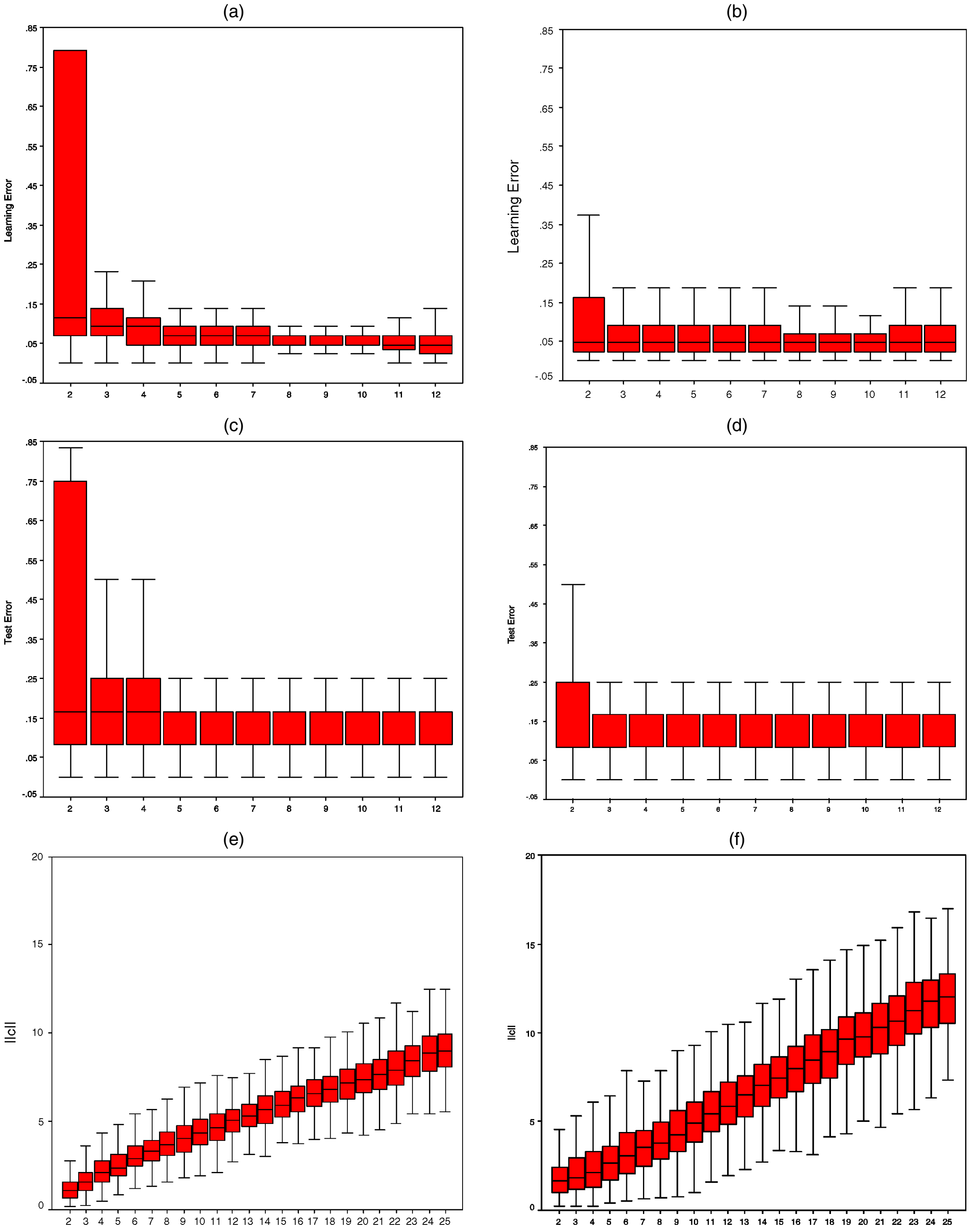

Figure 2. Vibration Dataset: Distributions of (a) Misclassification Training Errors With Weight Decay, (b) Misclassification Training Errors With Stopped Training, (c) Misclassification Test Errors With Weight Decay, (d) Misclassification Test Errors With Stopped Training, and (e) $||\mathbf{c}||_1$ With Weight Decay, (f) $||\mathbf{c}||_1$ With Stopped Training.

presence of models with different complexity but similar generalization errors is supported by values obtained in the different simulations and plotted in boxplots of Figures 2(a), 2(b), 2(c), and 2(d).

Also in this case, for the sake of completeness, Table 5 lists the values for model selection criteria when the number of degrees of freedom is selected equal to W. As in the previous example, some indices are useless because they assume negative values, and some others have a high variability and anomalous peaks. FPE and UEV are useless, because these indices assume negative values for more than five hidden units. GCV suggests large models; however, values obtained for this criterion for a number p of hidden units ranging from 2 to 12 have an extremely high variability and an anomalous peak for the model with 5 hidden units (which holds a number W not far from N). This peak and the high variability are not supported by val-

Table 4. Vibration Dataset: Values of Some Model Selection Criteria for $K = p + 1$

p	$\widehat{\mathcal{E}}(f_p;\mathcal{L})$	BIC	AIC	FPE	GCV	UEV
2	.143	1.863	1.735	5.670	5.703	.135
3	.130	1.757	1.586	4.888	4.940	.114
4	.117	1.642	1.428	4.178	4.247	.095
5	.113	1.605	1.349	3.864	3.957	.086
6	.114	**1.600**	1.301	3.687	3.810	.080
7	.112	1.603	1.262	3.554	3.710	.076
8	.109	1.610	1.226	3.438	**3.632**	.072
9	.110	1.656	1.230	3.460	3.703	.071
10	.108	1.677	**1.208**	**3.399**	3.693	.068
11	.108	1.724	1.212	3.430	3.788	.067
12	.110	1.762	**1.208**	3.436	3.866	**.066**

NOTE: The minimum values for the different criteria are in bold.

ues summarized in the boxplots of Figure 2. Furthermore, the BIC and AIC give strictly increasing values as p grows, so we should select the model with $p = 2$. To summarize, we note that with W degrees of freedom, model selection criteria give contrasting results, whereas with $K = p + 1$, results are much more satisfactory.

The third analysis concerns the UEV introduced in (10), that is,

$$\hat{\sigma}^2 = \frac{\widehat{\mathcal{E}}(f_K)}{N - K}, \tag{29}$$

for the two most popular regularization techniques (early stopping and weight decay) and for different choices of p and K. First, we generated three datasets according to the following models:

$$M_1: \quad \mathbf{x} = \mathbf{0} \in \mathbb{R}^5, \qquad y = \epsilon;$$

$$M_2: \quad \mathbf{x} = \mathbf{0} \in \mathbb{R}^{10}, \qquad y = \epsilon;$$

and

$$M_3: \quad x \in (0, 130) \subset \mathbb{R},$$

$$y = \theta_1 \left[\left(\frac{\theta_3}{\theta_3 - \theta_4} \right) e^{-\theta_4 x} + \left(\frac{\theta_1}{\theta_2} - \frac{\theta_3}{\theta_3 - \theta_4} \right) e^{-\theta_3 x} \right]^{-1} + \epsilon,$$

where ϵ follows a normal distribution $N(0, \sigma^2)$, and $\theta_1 = 105$, $\theta_2 = 2.6$, $\theta_3 = .5$, and $\theta_4 = -.02$. Data coming from the models M_1 and M_2 have been randomly generated with $\sigma^2 = .5$; two datasets coming from the model M_3 have been randomly generated choosing $\sigma^2 = 1$ and $\sigma^2 = 4$.

Table 5. Vibration Dataset: Values of Some Model Selection Criteria for $K = W$

p	W	BIC	AIC	FPE	GCV	UEV
2	17	3.178	2.453	12.370	15.272	.221
3	25	3.636	2.612	16.711	26.897	.265
4	33	4.084	2.762	28.246	76.717	.404
5	41	4.611	2.990	218.17	4,309.6	2.833
6	49	5.169	3.250	−35.91	108.37	−.43
7	57	5.737	3.519	−16.40	21.097	−.180
8	65	6.307	3.791	−10.53	8.171	−.107
9	73	6.917	4.101	−7.963	4.272	−.076
10	81	7.501	4.387	−6.271	2.505	−.056
11	89	8.111	4.699	−5.270	1.643	−.044
12	97	8.714	5.003	−4.510	1.134	−.036

NOTE: BIC and AIC are strictly increasing with p; moreover, FPE and UEV assume negative values for $p \geq 6$, FPE and GCV present an anomalous peak when W approximates the size of the learning set.

Data from M_1 and M_2 have been fitted using an MLP with $p = 5$ and $p = 10$ neurons in the input layer, whereas in the other case we considered an MLP with one neuron in the input layer. All simulations were performed in MATLAB. For each model, we generated 100 samples constituting 50 units, $N = 39$ units in the training set and 11 units in the test set for all cases; the input values in model M_3 have been sampled uniformly in the interval (0, 130). The values of $\hat{\sigma}^2$ obtained will be compared with the sample variance $s^2 = \sum_i e_i/(N-1)$, where $e_1, \ldots, e_N$ are the errors (i.e., the realizations of the sample $\epsilon_1, \ldots, \epsilon_N$). Data coming from M_1 and M_2 exhibit a sample variance equal to $s^2 = .2381$; as far as the two datasets coming from the model M_3 are concerned, the sample variances were equal to $s^2 = .7551$ and $s^2 = 2.4490$.

First, we considered the early-stopping technique. Table 6 summarizes the obtained results; such results are the average values over 100 samples with different partitions of observations in the training and in the test sets. In both models M_1 and M_2, the response does not depend on inputs, and thus the error variance estimate for these models must be independent on the number of inputs. This is verified when $K = p + 1$ is selected so that the values of $\hat{\sigma}^2$ are slightly greater than s^2. In contrast, in many cases the error variance estimates with $K = W$ degrees of freedom cannot be computed for MLPs, because the values obtained are negative; moreover, positive estimates are considerably larger than s^2.

We performed another set of simulations implementing the weight decay regularization technique to compare $K = p + 1$ and $K = \text{tr}(\mathbf{H}_\lambda)$ given in (26) for different values of the smoothing parameter λ; here we denote by $\hat{\sigma}^2_{p+1}$ and $\hat{\sigma}^2_\lambda$ the UEV (29) based on $K = p + 1$ and $K = \text{tr}(\mathbf{H}_\lambda)$. In this case the eigenvalues l_i $(i = 1, \ldots, p)$ of $\mathbf{T}$ depend on $\mathbf{x}$, so that we considered their means over the learning set. Tables 7 and 8 list the values that we obtained. The estimates agree quite well with s^2 (even if it is obvious that $\hat{\sigma}^2_{p+1}$ is a little larger than $\hat{\sigma}^2_\lambda$).

As far as the model M_3 is concerned, the results that we obtained are listed in Tables 9 and 10 for $s^2 = .7551$ and $s^2 = 2.4490$. Also in this case the obtained estimates agree quite well with s^2.

Finally, we remark that in many statistical software packages for neural networks, the smoothing parameter λ can be selected automatically according to cross-validation procedures, but this value is not available, so that it is impossible to compute the unbiased estimate of the variance using $\hat{\sigma}^2_\lambda$. Table 11 lists the estimates $\hat{\sigma}^2$ for data coming from the model M_3 for both $s^2 = .7551$ and $s^2 = 2.4490$ adopting both early stopping and weight decay (with automatic λ selection).

Table 6. Simulated Data: Error Variance Estimates Using $K = p + 1$ and $K = W$ (early-stopping regularization technique)

	$K = p + 1$		$K = W$	
p	M_1	M_2	M_1	M_2
2	.2677	.2557	.4015	.6576
3	.2748	.2681	.5658	4.6916
4	.2852	.2807	.9697	−.9543
5	.2977	.2922	3.2742	−.4383

NOTE: The values must be compared with $s^2 = .2381$.

Table 7. Simulated Data From Model M_1: Error Variance Estimates Using $K = p + 1$ and $K = tr(\mathbf{H}_\lambda)$ (weight decay regularization technique)

	$p=2$			$p=3$			$p=4$		
λ	$\hat\sigma^2_{p+1}$	$tr(\mathbf{H}_\lambda)$	$\hat\sigma^2_\lambda$	$\hat\sigma^2_{p+1}$	$tr(\mathbf{H}_\lambda)$	$\hat\sigma^2_\lambda$	$\hat\sigma^2_{p+1}$	$tr(\mathbf{H}_\lambda)$	$\hat\sigma^2_\lambda$
.02	.2410	1.9041	.2355	.1678	3.3583	.1655	.2512	1.9223	.2351
.03	.2416	1.8278	.2358	.2462	1.8526	.2352	.2513	1.8669	.2350
.04	.2424	1.7416	.2361	.2467	1.7997	.2354	.2516	1.8188	.2350
.05	.2434	1.6387	.2366	.2471	1.7271	.2354	.2519	1.7604	.2350
.06	.2446	1.5137	.2371	.2478	1.6470	.2358	.2523	1.7034	.2351
.08	.2458	1.3602	.2375	.2484	1.5759	.2360	.2527	1.6447	.2352
.09	.2472	1.1692	.2379	.2492	1.4886	.2363	.2532	1.5825	.2354
.10	.2483	.9995	.2381	.2501	1.3922	.2367	.2538	1.5171	.2356
.11	.2483	.9988	.2381	.2511	1.2857	.2371	.2544	1.4485	.2358
.12	.2483	.9985	.2381	.2522	1.1684	.2375	.2551	1.3766	.2361
.14	.2483	.9983	.2381	.2533	1.0399	.2380	.2559	1.3012	.2365
.15	.2483	.9981	.2381	.2537	.9986	.2381	.2568	1.2222	.2369
.16	.2483	.9979	.2381	.2537	.9983	.2381	.2577	1.1396	.2373
.18	.2483	.9977	.2381	.2537	.9981	.2381	.2587	1.0529	.2378
.19	.2483	.9975	.2381	.2537	.9979	.2381	.2593	.9983	.2381
.20	.2483	.9973	.2381	.2537	.9977	.2381	.2593	.9981	.2381
.22	.2483	.9971	.2381	.2537	.9975	.2381	.2593	.9979	.2381
.23	.2483	.9969	.2381	.2537	.9974	.2381	.2593	.9977	.2381
.25	.2483	.9967	.2381	.2537	.9972	.2381	.2593	.9975	.2381

NOTE: The values must be compared with $s^2 = .2381$.

Table 8. Simulated Data From Model M_2: Error Variance Estimates Using $K = p + 1$ and $K = tr(\mathbf{H}_\lambda)$ (weight decay regularization technique)

	$p=2$			$p=3$			$p=4$		
λ	$\hat\sigma^2_{p+1}$	$tr(\mathbf{H}_\lambda)$	$\hat\sigma^2_\lambda$	$\hat\sigma^2_{p+1}$	$tr(\mathbf{H}_\lambda)$	$\hat\sigma^2_\lambda$	$\hat\sigma^2_{p+1}$	$tr(\mathbf{H}_\lambda)$	$\hat\sigma^2_\lambda$
.05	.2180	1.8250	.2127	.1448	2.8709	.1414	.1149	3.7284	.1117
.06	.2192	1.7344	.2135	.1534	2.6732	.1491	.1458	2.9299	.1394
.08	.2204	1.6847	.2144	.2223	1.7173	.2118	.1523	2.7635	.1451
.09	.2216	1.6404	.2153	.2237	1.6538	.2128	.1585	2.5937	.1505
.10	.2229	1.5982	.2164	.2241	1.6527	.2132	.2270	1.6807	.2114
.11	.2242	1.5574	.2175	.2244	1.6141	.2134	.2275	1.6457	.2118
.12	.2256	1.5173	.2187	.2263	1.5467	.2149	.2280	1.6185	.2120
.14	.2270	1.4778	.2199	.2274	1.5127	.2157	.2285	1.5892	.2124
.15	.2284	1.4387	.2211	.2277	1.5053	.2160	.2293	1.5542	.2130
.16	.2299	1.3997	.2224	.2279	1.4903	.2161	.2299	1.5264	.2134
.18	.2334	1.3147	.2253	.2289	1.4611	.2169	.2306	1.4997	.2140
.19	.2331	1.3221	.2251	.2299	1.4324	.2177	.2313	1.4738	.2145
.20	.2347	1.2834	.2265	.2309	1.4042	.2186	.2321	1.4487	.2151
.22	.2364	1.2450	.2279	.2331	1.3538	.2204	.2328	1.4244	.2157
.23	.2381	1.2069	.2294	.2343	1.3257	.2214	.2337	1.4006	.2164
.25	.2398	1.1692	.2308	.2344	1.3216	.2215	.2345	1.3775	.2170

NOTE: The values must be compared with $s^2 = .2381$.

Table 9. Simulated Data From Model M_3: Error Variance Estimates Using $K = p + 1$ and $K = tr(\mathbf{H}_\lambda)$ (weight decay regularization technique)

	$p=2$		$p=3$		$p=4$		$p=5$	
λ	$\hat\sigma^2_{p+1}$	$\hat\sigma^2_\lambda$	$\hat\sigma^2_{p+1}$	$\hat\sigma^2_\lambda$	$\hat\sigma^2_{p+1}$	$\hat\sigma^2_\lambda$	$\hat\sigma^2_{p+1}$	$\hat\sigma^2_\lambda$
.005	.5757	.5568	.5610	.5280	.5660	.5155	.5770	.5091
.05	.6050	.5738	.6180	.5701	.6343	.5688	.6523	.5681
.1	.6314	.5983	.6338	.5839	.6456	.5778	.6614	.5747
.2	.6883	.6526	.6780	.6247	.6811	.6095	.6911	.6002
.3	.7288	.6911	.7182	.6618	.7199	.6443	.7282	.6324
.4	.7646	.7249	.7498	.6910	.7525	.6734	.7620	.6618

NOTE: The values must be compared with $s^2 = .7551$.

Table 10. Simulated Data From Model M_3: Error Variance Estimates Using $K = p + 1$ and $K = tr(\mathbf{H}_\lambda)$ (weight decay regularization technique)

	$p=2$		$p=3$		$p=4$		$p=5$	
λ	$\hat\sigma^2_{p+1}$	$\hat\sigma^2_\lambda$	$\hat\sigma^2_{p+1}$	$\hat\sigma^2_\lambda$	$\hat\sigma^2_{p+1}$	$\hat\sigma^2_\lambda$	$\hat\sigma^2_{p+1}$	$\hat\sigma^2_\lambda$
.005	2.1347	2.0363	2.1414	2.0160	2.1919	2.0045	2.2496	1.9925
.05	2.3873	2.2657	2.4505	2.2619	2.5188	2.2602	2.5916	2.2602
.1	2.4062	2.2818	2.4632	2.2707	2.5302	2.2659	2.6028	2.2627
.2	2.4674	2.3404	2.5087	2.3121	2.5665	2.2971	2.6342	2.2881
.3	2.5092	2.3800	2.5490	2.3493	2.6045	2.3312	2.6701	2.3192
.4	2.5459	2.4141	2.5813	2.3789	2.6375	2.3606	2.7039	2.3485

NOTE: The values must be compared with $s^2 = 2.4490$.

Table 11. Simulated Data From Model M_3: Error Variance Estimates Using $K = p + 1$ for Both Early Stopping (ES) and Weight Decay (WD) (λ selected automatically) Regularization Strategies

	$s^2 = .7551$		$s^2 = 2.4490$	
p	ES	WD	ES	WD
2	.6885	.6323	2.4931	2.4625
3	.6387	.5869	2.5780	2.5086
4	.6828	.5915	2.6395	2.4408
5	.6638	.5892	2.5626	2.3848

NOTE: The values must be compared with $s^2 = .7551$ and $s^2 = 2.4490$.

We note that in all simulations, the unbiased estimates of the variance using both $K = p + 1$ and $K = \text{tr}(\mathbf{H}_\lambda)$ agree with the true values.

6. CONCLUDING REMARKS

Bartlett's (1998) theorem 1 gives the theoretical basis for using neural models with a total number of weights larger than the number of sample data points used to estimate the weights. Based on this result, we have investigated the role of the weights of a neural network with a mapping function of the form $f(\mathbf{x}) = \sum_{k=1}^{p} c_k \tau(\mathbf{a}_k'\mathbf{x} + b_k) + c_0$. We have shown that the two levels of weights play quite different roles; the input-to-hidden weights concern just a (nonlinear) projection from $\mathbb{R}^m$ to $\mathbb{R}^p$, whereas the hidden-to-output weights fit the projected data and perform the regression or the classification, according to the problem at hand. Both the projection and the fit are optimized according to the target values. From this point of view, the complexity of the network depends more on the number of hidden units than on the whole set of weights, and we have shown that the greatest complexity is reached by a network with a number of hidden neurons equal to the number of sample data points.

According to results for richly parameterized models, the concept of equivalent number of degrees of freedom K has been introduced for one-hidden-layer networks like MLP and RBF as the trace of the projection matrix, and it proves to be $K \le p + 1$, where $p + 1$ is the number of the hidden to output weights. The value of K also depends on the adopted regularization technique. If early stopping is implemented, then we set $K = \text{tr}(\mathbf{H}) = p + 1$, whereas when weight decay is implemented, it is $K = \text{tr}(\mathbf{H}_\lambda) < p + 1$, and it also depends on the value of the smoothing parameter λ. However, in both cases, for model selection according to the BIC and AIC we can adopt $K = p + 1$, because we are interested mainly in the number of hidden units p that minimizes such indices. In contrast, estimation of the error variance (29) requires more attention; if early stopping is adopted, then we set $K = p + 1$, whereas in the other case (i.e., the weight decay), using $K = p + 1$ leads to slightly larger estimates than those obtained using the trace $K = \text{tr}(\mathbf{H}_\lambda)$. However, our numerical simulations showed that $K = p + 1$ also could be adopted in general for this aim.

In contrast to Ye's GDF and according to the definition of ρ and to Hastie and Tibshirani's definitions, our notion of degrees of freedom does not depend on the underlying true function. For a certain MLP, it is equal for all three models, and fitting an MLP to a pure noise dataset does not require more degrees of freedom than fitting an MLP to a dataset with a structure between the inputs and the target.

We have also presented some numerical studies of the behavior of the constant $\|\mathbf{c}\|$, given by the sum of the absolute values of the hidden to output weights, and of the training and test error of an MLP across the simulation. These studies show that the values of $\|\mathbf{c}\|$ have a similar increasing trend, with respect to the number p of hidden units, for both stopped training and weight decay. They also show that this quantity is particularly unstable across simulations, whereas our measure of degrees of freedom does not depend on local minima or on the convergence of the learning algorithm. The simulation studies show that besides model selection, the proposed measure also may be used to achieve a good estimate of the error variance, especially for small networks, whereas the total number of weights is useless for this task.

Finally, we remark that for deeper networks with more than one hidden layer, analogous considerations apply. The equivalent number of degrees of freedom relates to the units in the highest hidden layer (i.e., on the hidden layer immediately before of the output layer), with the other layer performing only geometric transformations of data; this is also congruent with theorem 28 of Bartlett (1998).

ACKNOWLEDGMENTS

The authors thank S. Zani and G. D. Costanzo for helpful suggestions. They also thank the referee and the editor for their comments, which improved the earlier version of the article.

[Received June 2002. Revised September 2004.]

REFERENCES

Akaike, H. (1970), "Statistical Predictor Identification," *The Annals of Statistics*, 22, 203–217.

——— (1974), "A New Look at the Statistical Model Identification," *IEEE Transactions on Automatic Control*, 19, 716–723.

An, G. (1996), "The Effects of Adding Noise During Backpropagation Raining on a Generalization Performance," *Neural Computation*, 8, 643–674.

Azencott, R., Doutriaux, A., and Younes, L. (1993), "Synchronous Boltzmann Machines and Curve Identification Tasks," *Network*, 4, 461–480.

Bartlett, P. L. (1998), "The Sample Complexity of Pattern Classification With Neural Networks: The Size of the Weights Is More Important Than the Size of the Network," *IEEE Transaction on Information Theory*, 44, 525–536.

Bishop, C. M. (1995), *Neural Networks for Pattern Recognition*, Oxford, U.K.: Clarendon Press.

Cavarra, S., Crupi, V., Guglielmino, E., and Ingrassia, S. (2002), "Reti Neurali per l'Analisi Dei Dati Vibrometrici in Campo Petrolchimico: Un Caso Studio," *Statistica Applicata*, 13, 5–16 [in Italian].

Cherkassky, V., and Mulier, F. (1998), *Learning From Data*, New York: Wiley.

De Veaux, R. D., Psichogios, D. C., and Ungar, L. H. (1993), "A Comparison of Two Nonparametric Estimation Schemes: MARS and Neural Networks," *Computers and Chemical Engineering*, 17, 819–837.

De Veaux, R. D., Schumi, J., Schweinsberg, J., and Ungar, L. H. (1998), "Prediction Intervals for Neural Networks via Nonlinear Regression," *Technometrics*, 40, 273–282.

De Veaux, R. D., and Ungar, L. H. (1994), "Multicollinearity: A Tale of Two Nonparametric Regressions," in *Selecting Models From Data: AI and Statistics IV*, eds. P. Cheeseman and R. W. Oldford, New York: Springer-Verlag, pp. 293–302.

——— (1996), Discussion of "Neural Networks in Applied Statistics," by H. S. Stern, *Technometrics*, 38, 215–218.

Fine, T. L. (1999), *Feedforward Neural Network Methodology*, New York: Springer-Verlag.

Friedman, J. H. (1991), "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, 19, 1–141.

Hastie, T., and Tibshirani, R. (1990), *Generalized Additive Models*, New York: Chapman & Hall.

Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning, Data Mining, Inference and Prediction*, New York: Springer-Verlag.

Hodges, J. S., and Sargent, D. J. (2001), "Counting Degrees of Freedom in Hierarchical and Other Richly Parameterized Models," *Biometrika*, 88, 367–379.

Ingrassia, S. (1999), "Geometrical Aspects of Discrimination by Multilayer Perceptrons," *Journal of Multivariate Analysis*, 68, 226–234.

Ingrassia, S., and Morlini, I. (2002), "Modelli Neuronali per Piccoli Insiemi di Dati," in *Analisi Multivariata per la Qualità Totale: Metodologia Aspetti Computazionali ed Applicazioni*, eds. N. C. Lauro and G. Scepi, Milan, Italy: Franco Angeli Editore, pp. 29–40 [in Italian].

——— (2004), "On the Degrees of Freedom in Richly Parameterised Models," in *Proceedings of COMPSTAT 2004 16th Symposium*, ed. J. Antoch, Heidelberg: Physica-Verlag, pp. 1237–1244.

Jolliffe, I. T. (1986), *Principal Component Analysis*, New York: Springer-Verlag.

Kadane, H. B., and Lazar, N. A. (2004), "Methods and Criteria for Model Selection," *Journal of the American Statistical Association*, 99, 279–290.

Kearns, M. J., and Schapire, R. E. (1990), "Efficient Distribution-Free Learning of Probabilistic Concepts," in *Proceedings of th 31th Symposium on Foundation of Computer Science*, Los Alamos, CA: IEEE Computer Society Press, pp. 382–391.

Lawrence, S., Giles, C. L., and Tsoi, A. C. (1996), "What Size Neural Network Gives Optimal Generalization? Convergence Properties of Backpropagation," Technical Reports UMIACS-TR-96-22 and CS-TR-3617, Institute for Advanced Computer Studies, University of Maryland.

——— (1997), "Lessons in Neural Network Training: Overfitting May Be Harder Than Expected," in *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, AAAI-97, pp. 540–545.

MacKay, D. J. C. (1992), "Bayesian Interpolation," *Neural Computation*, 4, 415–447.

Moody, J. (1992), "The *Effective* Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning System," in *Neural Information Processing Systems 4*, eds. J. Moody, S. Hanson, and R. Lippmann, San Mateo, CA: Morgan Kaufmann, pp. 847–854.

Ripley, B. D. (1996), *Pattern Recognition and Neural Networks*, Cambridge, U.K.: Cambridge University Press.

Rudin, W. (1966), *Real and Complex Analysis*, New York: McGraw-Hill.

Sarle, W. (1999), "Donoho–Johnstone Benchmarks: Neural Net Results," available at *ftp://ftp.sas.com/pub/neural/dojo/dojo.html*.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *The Annals of Statistics*, 6, 461–464.

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society*, Ser. B, 36, 111–147.

Ye, J. (1998), "On Measuring and Correcting the Effects of Data Mining and Model Selection," *Journal of the American Statistical Association*, 93, 120–131.

Yuan, J. L., and Fine, T. L. (1998), "Neural Network Design for Small Training Sets of High Dimension," *IEEE Transactions on Neural Networks*, 9, 266–280.