FPE-Based Criteria to Dimension Feedforward Neural Topologies

Cesare Alippi, *Member, IEEE*

***Abstract*—This paper deals with the problem of dimensioning a feedforward neural network to learn an unknown function from input/output pairs. The ultimate goal is to tune the complexity of the neural model with the information present in the training set and to estimate its performance without needing new data for cross-validation. For generality, it is not assumed that the unknown function belongs to the family of neural models. A generalization of the final prediction error to biased models is provided, which can be applied to learn unknown functions both in noise free and noise affected applications. This is based on a new definition of the effective number of parameters used by the neural model to fit the data. New criteria for model selection are introduced and compared with the generalized prediction error and the network information criteria.**

***Index Terms*— FPE, learning from samples, model selection, neural networks.**

NOMENCLATURE

θ	Neural network parameters vector.
$\hat{\theta}$	Trained parameters vector.
θ^o	Optimal parameters vector.
V_N	Training error function.
$\bar{V}$	Error function.
N	Number of training data.
Z^N	Training data set.
(x, y)	Training pair.
$y(\theta)$	Neural network characterized by θ.
ϵ	Difference between the real value and the neural output $y - y(\hat{\theta})$.
P_θ	covariance matrix of θ.
A'	Gradient of A.
A''	Hessian of A.
A^+	Moore-Penrose pseudoinverse of A.
$\Psi = \frac{d\epsilon}{d\theta}$	gradient w.r.t θ.
P	Orthogonal projector onto the column space of $\bar{V}''$.
p_{eff}	Moody's effective number of parameters.
$\bar{p}$	Alippi's effective number of parameters.
$\bar{E}[A] = \frac{1}{N}\sum_{i=1}^{N} A_i$	empirical average of A.

I. INTRODUCTION

LEARNING an input–output relationship from a set of value pairs is a fundamental problem in many fields. Examples include reconstructing unknown functions [1], time series forecasting [2], and modeling very complex processes [3].

The determination of a model which approximates a function, given a set of input/output pairs, comprises three distinct phases: model selection (to choose the correct complexity of the model), model parameterization or learning (to determine the parameters of the model), and model validation (to evaluate the generalization ability of the model). The model with the optimal generalization ability is then chosen to solve the function-approximation task.

The function-approximation problem has been widely addressed in the literature, usually with respect to linear models under the assumption that the function to be learned is linear or quasilinear. If this is not the case, the family of approximation models must be extended to include nonlinear models such as neural networks [4]. A number of powerful neural techniques have been developed, such as radial-basis functions [5], [6], mixture of Gaussians [7], feedforward [8], and recurrent [9] topologies.

Several criteria have been suggested to select an appropriate neural topology by reducing/optimizing the number of neurons/weights in the network (e.g., optimization based on spectral decomposition [10], covariance matrix [11], optimal brain damage (OBD) [12], surgeon (OBS) [13], and growing algorithms [14]). For these methods, model selection is carried out by evaluating the performance of different topologies on a new set of examples (crossvalidation). The best model is the one minimizing the generalization error on the crossvalidation set. Unfortunately, crossvalidation presents a serious disadvantage, especially when a limited data set is available. Saving examples to crossvalidate a model reduces the data available for configuring the parameters (thus impairing the efficiency of learning). In such a case, all data should be used for training, thereby making it necessary for the model selection and validation process to use criteria which estimate the generalization ability of the neural model from the training data itself.

Of particular relevance, among criteria following this principle, are the generalized prediction error (GPE) [8] and the network information criterion (NIC) [15]. In this paper, we introduce the final prediction error biased (FPEB) criterion which extends the final prediction error (FPE) [16] to the case of biased models.

GPE provides a trivial model selection in noise-free applications by selecting the model with the minimal training error. This procedure is not correct if the number of training pairs is small. This limitation is solved by FPEB, which introduces a correction term which is a function of the number of training pairs.

Manuscript received November 15, 1995; revised September 24, 1998. This paper was recommended by Associate Editor A. Kuh.

The author is with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, 20133 Milano, Italy.

Publisher Item Identifier S 1057-7122(99)06358-8.

FPEB differs from NIC in that FPEB distinguishes between noise-free and noise-affected cases to take advantage of *a priori* information. It is always computationally feasible, even when NIC is ill-conditioned, and considers an early stopping strategy to limit overtraining effects (overfitting caused by the learning phase) in overdimensioned networks.

The problem of learning from examples can be formalized as follows. Let $\bar{y} = f(x) \mid x \in \Re^n, \bar{y} \in \Re$ be the unknown function to be learned and Z^N the set containing the N pairs

$$(x_1, y_1), \ldots, (x_N, y_N) \tag{1.1}$$

drawn from a stationary density function Λ and generated according to the classical signal-plus-noise model

$$y_i = \bar{y}_i + \zeta = f(x_i) + \zeta. \tag{1.2}$$

In other words, y_i is the generic actual measurable output, corrupted by an independent and identically distributed (i.i.d.) noise ζ with zero mean and a variance λ_o which is generally unknown.

Our goal is to find the function y^o which best approximates $\bar{y}$ given (1.1) and a loss criterion V_N [e.g., a mean square error (MSE)]. The search for the best approximating function is carried out within a hierarchical model structure M. The model structure $M = \{M_k\}, k \in \mathcal{N}$ considered in this paper contains two-layered feedforward neural networks with n inputs, k hidden units (characterized by a nonlinear differentiable activation function, e.g., a sigmoidal-like function), and a single linear output. The interest for such models derives from the fact that, under weak hypotheses, they are universal function approximators [17]. Each element $M_k = \hat{y}(\theta, x)$ is completely defined by a column vector of parameters θ, which contains all free parameters of the network (weights and biases in our case). We will assume that the p-dimensional θ vector belongs to a $\mathcal{C}^1$ differentiable manifold of parameters Θ (if the neural network is fully connected between layers then $p = k(n+2)+1$). The model corresponding to a particular $\theta \in \Theta$ will be denoted $M_k(\theta)$. We say that M_k is biased if there does not exist a θ such that $M_k(\theta) = \bar{y}$. As a consequence, even in the best case when the learning process provides the optimal approximating function y^o, we have that $\|\bar{y} - y^o\|_{V_N} \neq 0$ (see also [18] for a detailed analysis of the bias/variance dilemma). We consider as a simple example of model bias the problem of learning the noise-free function $\bar{y} = x$ defined in the $[-1, 1]$ interval. We choose V_N to be the MSE N tending to infinity, with x_i subject to a uniform distribution and the model family $M_a = \{\hat{y} = a, a \in \Re\}$. The best approximating function $y^o = 0$ is such that $\|\bar{y} - y^o\| = 1/3 \neq 0$.

In this paper we adopt (1.3) as the general error-based criterion for configuring the neural parameters

$$V_N(\theta, Z^N) = \frac{1}{N} \sum_{i=1}^{N} l(\theta, \epsilon_i) \tag{1.3}$$

where $l(\cdot, \cdot)$ is a discrepancy or error function (e.g. $l(\theta, \epsilon_i) = \epsilon_i^2$) and $\epsilon_i = y_i - \hat{y}_i$ with $\hat{y}_i = \hat{y}(\theta, x_i)$. Minimization of (1.3) with a learning procedure will provide a minimum $\hat{\theta}$, dependent on the given Z^N. As a consequence, it seems reasonable that as the number of pairs N tends to infinity, $\hat{\theta}$ should converge to an optimal parameter configuration θ^o (for which $y^o = \hat{y}(\theta^o, x)$) yet to be defined.

The structure of this paper is as follows. Section II investigates the asymptotic behaviors of V_N and $\hat{\theta}$ by describing the elements to which the sequences converge.

The general criterion is derived by considering asymptotic results and is tailored to neural networks. The section ends with a brief description of Moody's GPE. In Section III, results are specialized to the case where $l(\theta, \epsilon) = \epsilon^2$ and we obtain the FPEB. On the basis of the effective number of parameters, the criterion is then refined to take advantage of *a priori* information, namely, whether the application is noise free or not. Relationships and differences between FPEB, GPE, and NIC are derived. Finally, in Section IV, the effectiveness of the method is demonstrated on examples of learning nonlinear functions.

II. The General Criterion for Selection and Validation

A. Extending Asymptotic Results to Neural Networks

Let us define $D \subset \Theta$ to be the subset of point(s) minimizing the function

$$\bar{V}(\theta) = E[V_N(\theta, Z^N)] = E[l(\theta, \epsilon)] \tag{2.1}$$

evaluated with respect to the probability density function Λ under the hypothesis of i.i.d. inputs. We might interpret $y^o = \hat{y}(\theta^o, x), \theta^o \in D$ as the best average approximation of $\bar{y}$ given Z^N and V_N.

The relationship between $\bar{V}$ and V_N is such that [19]

R1: when N tends to infinity, $V_N(\theta, Z^N) - \bar{V}(\theta)$ converges uniformly to zero with probability 1 in D.

This convergence result implies that the set of accumulation points of local/global minima of V_N are, respectively, the points of local/global minima of $\bar{V}$.

Several important results in system identification are based on the asymptotic relationships between points minimizing (1.3) and those minimizing (2.1). Results on convergence and rapidity of convergence are well known under the strong hypothesis that the true system belongs to the model family [16] and, more specifically, to linear models. Results (valid for modeling dynamical systems) have been extended in [20] and [21] to cover the general case where the system does not belong to the model. Now, by assuming that there exists a unique global minimum θ^o and denoting with $\bar{V}''$ the Hessian matrix (obtained by differentiating $\bar{V}$ twice with respect to θ), it has been proved [19] that:

R2: if $\bar{V}'' > \delta I$ (where I is the identity matrix and $\delta > 0$), then $\sqrt{N}(\hat{\theta} - \theta^o) \to 0$ as N tends to infinity and, for a sufficiently large N , $\sqrt{N}(\hat{\theta} - \theta^o)$ is asymptotically normal (AsN) with zero mean and P_θ covariance matrix

$$\sqrt{N}(\hat{\theta} - \theta^o) = \mathrm{AsN}(0, P_\theta) \tag{2.2}$$

where

$$P_\theta = [\bar{V}''(\theta^o)]^{-1} Q [\bar{V}''(\theta^o)]^{-1} \tag{2.3}$$

$$Q = NE[(V_N'(\theta^o, Z^N))(V_N'(\theta^o, Z^N))^T]. \tag{2.4}$$

It can easily be proved that *R1* and *R2* still hold when considering feedforward neural networks, but the use of *R2* requires some additional care. The assumption of a unique

point for $\bar{V}$ in *R2* is intended to confine the analysis to the neighborhood of θ^o to which $\hat{\theta}$ converges. In any case, it should be noted that being in different global minima will not modify the behavior of the V entities present in (1.3) and (2.1).

A second strong hypothesis of *R2* requires $\bar{V}''$ to be positive definite in the neighborhood of θ^o. If there are isolated minima (i.e. for each of which there exists a safe neighborhood satisfying the positive definite condition), then *R2* still holds. On the other hand, when $\bar{V}''$ is singular, we cannot obtain the inverse needed in (2.3). The problem can be overcome by considering the Moore–Penrose pseudoinverse $\bar{V}''^{+}(\theta)$ [30], [31] and we can extend (2.3) as

$$PP_\theta = \bar{V}''^{+}(\theta^o)Q\bar{V}''^{+}(\theta^o) \tag{2.5}$$

where $P = \bar{V}''^{+}(\theta^o)\bar{V}''(\theta^o) = \bar{V}''(\theta^o)\bar{V}''^{+}(\theta^o)$ is an idempotent matrix. The pseudoinverse is the same as the inverse when $\bar{V}''(\theta^o)$ is nonsingular. In such a case, P becomes the identity matrix and (2.5) coincides with (2.3). The proof is given in Appendix A. Such an extension is relevant, since it allows the learning of functions from real data where $\bar{V}''$ (or its estimate) is often singular (see Section IV).

A second aspect to be considered is the effect of training time in estimating the parameter vector $\hat{\theta}$ in overdimensioned networks (we do not know *a priori* whether the chosen network topology is overdimensioned to the application).

This problem does not arise in linear systems where no training procedures are necessary and the best estimate $\hat{\theta}^o$ is generally simply computed offline in a single step according to the linear regression theory [16]. This is not the case in neural networks where the parameter configuration evolves during training (being updated by the learning algorithm) and, in a long training run, we have that $\lim_{t_{\mathrm{tr}}\to\infty}\hat{\theta}(t_{\mathrm{tr}}) = \hat{\theta}^o$, which might be far from being a good estimate of any $\theta^o \in D$. This problem has also been observed in [22].

Such a behavior is common with overdimensioned networks where overtraining effects are evident (see Section IV). The best estimate of θ^o is reached in correspondence with a finite training time $t_{\mathrm{tr}} = \bar{t}$. To keep the effect of overtraining under control, the stopping point for the training phase should be carefully determined (e.g., by evaluating the network's performance on the test set [22]). If, however, no test sets are available because of the shortage of data, then we should also solve this problem. This will be done in Section III where a strategy is implemented to determine $t_{\mathrm{tr}} = \bar{t}$ (and therefore the correct $\hat{\theta}$ to be considered).

A further problem to be analyzed is the local minima issue which can be experimentally overcome by using suitable learning algorithms and stochastic minimization procedures such as simulated annealing or genetic algorithms which guarantee to reach a global minimum with probability one (even if these methods are often computationally impractical).

B. The Criterion

The classical derivation [16] may now be followed by introducing a figure of merit which takes into account the complexity of a model. A natural criterion to validate a given model (which in the following for ease of notation will be indicated as M_k) is to consider how the estimate obtained performs on the average

$$J(M_k) = E[\bar{V}(\hat{\theta})]. \tag{2.6}$$

We can prove that the following relationships hold:

$$E[\bar{V}(\hat{\theta})] \approx \bar{V}(\theta^o) + \frac{1}{2N}\,\mathrm{tr}([\bar{V}''(\theta^o)PP_\theta]) \tag{2.7}$$

$$E[V_N(\hat{\theta}, Z^N)] \approx \bar{V}(\theta^o) - \frac{1}{2N}\,\mathrm{tr}([\bar{V}''(\theta^o)PP_\theta]) \tag{2.8}$$

where tr is the matrix trace (see Appendix B for the proof). By substituting $\bar{V}(\theta^o)$, obtained from (2.8) in (2.7), expression (2.6) can be approximated as

$$J(M_k) = E[\bar{V}(\hat{\theta})] \approx E[V_N(\hat{\theta}, Z^N)] + \frac{1}{N}\,\mathrm{tr}\big(\big[\bar{V}''(\theta^o)PP_\theta\big]\big). \tag{2.9}$$

Expression (2.9) is of fundamental importance and needs to be interpreted both under the validation and the selection aspects.

Validation Aspect: Expression (2.9) states that the averaged expected performance of the model is approximately the sum of the expected loss criterion and a second term, depending on the characteristics of the noise and the sensitivity of the estimate with respect to the parameters. Expression (2.9), once given a trained model $M_k(\hat{\theta})$, validates it by providing a measure of its generalization ability.

Selection Aspect: Expression (2.9) underlines the compromise between the model complexity and training error performances. Obviously, the balance depends on the current trained model. It is well known that by increasing the complexity of the model, the training error will decrease. However, the second term of (2.9), being the trace of a matrix of order $(p_k \times p_k)$ (and therefore dependent on the model complexity), increases. The optimal model (selection aspect) is then the element $M_{\bar{k}}$ of M for which $J(M_{\bar{k}}) \le J(M_k) \mid M_k \in M$, namely, the model minimizing the generalization error.

If a single run is taken into account then, as with the classic analysis, we can replace the expectation of $V_N(\hat{\theta}, Z^N)$ with the only observation we have

$$J(M_k) \approx V_N(\hat{\theta}, Z^N) + \frac{1}{N}\,\mathrm{tr}([\bar{V}''(\theta^o)PP_\theta]). \tag{2.10}$$

Due to the simplification, all criteria coming from (2.10) provide only an approximation of the generalization ability of the model and, as a consequence, limit the model validation in the following analysis. Fortunately, the introduced approximation does not impair the effectiveness of model selection, at least for the considered M structure, as proved in [15]. There, the authors, by extending results given in [23], provide the NIC criterion formally similar to (2.10). In its simple form, NIC does not immediately allow the introduction of the concept of an effective number of parameters (namely the effective number of degrees of freedom used by the model to solve the approximating task). However, in the case of additive noise and a MSE discrepancy function, it can be shown to be related to GPE, which does allow for such a concept.

Whenever the network degenerates to a submodel [15], the effective number of parameters is defined by the limit

case when θ tends to a critical value (which, although it is theoretically correct, we experimentally determined to be impractical from the application point of view). Furthermore, NIC may be ill conditioned when the Hessian is singular, thus impairing the effectiveness of the criterion. Finally, NIC is evaluated in correspondence with $\hat{\theta}$, which is determined during the long training run. We already indicated that $\hat{\theta}$ is not necessarily a good estimate of θ^o. The criterion proposed in this paper attempts to resolve such limitations.

C. A Different Approach: The Generalized Prediction Error

An interesting criterion for model selection and validation different from (2.6) has been suggested in [8] where the author introduces the prediction risk for training sets of size N, with input density equal to the empirical density defined by the available training set

$$\Omega' = \frac{1}{N}\sum_{i=1}^{N}\delta(x - x_i). \tag{2.11}$$

For such training sets, the N inputs are held fixed but the y_i's may vary according to the conditional probability density $P(y \mid x_i)$. The criterion is then defined as the expected validation set error for validation sets of size N, in which the unknown input density Ω is replaced with that of the training set Ω'. The criterion becomes

$$E[V_N]_{\text{Val}} \approx E[V_N]_{\text{train}} + \sigma^2\frac{p_{\text{eff}}(\gamma)}{N} \tag{2.12}$$

where γ is the weight decay [25] and p_{eff} the Moody's effective number of parameters. (2.12) generalizes the well-known relationship valid for linear systems [24] to nonlinear and biased models. In particular, for the case of the signal-plus-noise model of expression (1.1), we have that $\sigma^2 = \lambda_o$.

Finally, the GPE becomes

$$\text{GPE}(\gamma) = V_N(\hat{\theta}) + \hat{\sigma}^2\frac{\hat{p}_{\text{eff}}(\gamma)}{N} \tag{2.13}$$

where

$$\hat{p}_{\text{eff}} = \text{tr}(TU^{-1}T^T) \tag{2.14}$$

U is the Hessian of the objective function and T is the $N \times p$ matrix of the derivatives of the training error.

III. FPEB: Extending FPE to Biased Models

In the following, we consider the case in which a sufficiently large but finite number of data N is given, thus making effective results given in previous analyses. To take into account the model bias, we must refine the signal-plus-noise model of (1.2) by considering $\overline{y}(x)$ as the sum of two terms: the approximating function $\hat{y}(x)$ and the distortion function $\rho(x)$

$$y_i = \overline{y}_i + \zeta = \hat{y}(x_i, \hat{\theta}) + \rho(x_i, \hat{\theta}) + \zeta. \tag{3.1}$$

By invoking *R1*, when N tends to infinity, (3.1) becomes

$$y_i = y^o(x_i, \theta^o) + \rho(x_i, \theta^o) + \zeta. \tag{3.2}$$

We define $\rho(x_i, \theta^o)$ to be the punctual bias in x_i.

In all subsequent analyses we will focus the attention on the MSE loss criterion

$$V_N(\theta, Z^N) = \frac{\text{MSE}}{2} = \frac{1}{2N}\sum_{i=1}^{N}\epsilon_i^2. \tag{3.3}$$

The structure of the section is as follows. In Section III-A, the final prediction error for biased models FPEB is introduced. Different criteria can then be derived from the general one by exploiting *a priori* knowledge (e.g., by knowing that data are noise free). First, the criterion is specialized to the case of pure punctual bias, as happens when data are not affected by noise. The goal is to approximate a deterministic function. Then, to cope with the fact that, in general, the punctual bias is unknown (data are in this case affected by noise), we consider an approximation which treats the punctual bias as a random variable. This last approximation, even if particularly appealing since it provides a criterion formally similar to FPE, is *a priori* not correct because the bias is deterministic. By relaxing this last assumption, the correct criterion may be finally obtained by directly deriving it from the general one. In Section III-B, it will be demonstrated that GPE may be related to FPEB. Finally, in Section III-C, we will analyze the impact of training time on the proposed criteria.

A. Evaluating the FPEB

Having chosen a loss function V_N, we have to adapt the general criterion to it. This requires computation of the trace term present in (2.10). To this end, remembering that $\epsilon = \rho(x, \theta) + \zeta$ and by indicating

$$\Psi(x, \theta^*) = \left.\frac{d\epsilon}{d\theta}\right|_{\theta=\theta^*} \tag{3.4}$$

as the column vector of partial derivatives $\frac{\partial\epsilon}{\partial\theta_i}$ of the error with respect to the generic ith parameter component, the Q matrix of (2.4) can be rewritten as

$$Q = NE\left[V_N' V_N'^T\right] \tag{3.5}$$

with

$$V_N' = \left(\frac{1}{N}\sum_{i=1}^{N}\Psi_i\rho_i + \frac{1}{N}\sum_{i=1}^{N}\Psi_i\zeta_i\right)$$
$$\Psi_i = \Psi(x_i, \theta^o); \quad \Psi_{\rho,i} = \Psi_i\rho(x_i, \theta^o) = \Psi_i\rho_i.$$

Now, by remembering that ζ and x are i.i.d. random variables, we obtain that

$$Q = (\lambda_o\bar{E}[\Psi\Psi^T] + \bar{E}[B]) \tag{3.6}$$

$$\bar{E}[\Psi\Psi^T] = \frac{1}{N}\sum_{i=1}^{N}\Psi_i\Psi_i^T \quad \bar{E}[B] = \frac{1}{N}\sum_{i=1}^{N}\Psi_{\rho,i}\Psi_{\rho,i}^T. \tag{3.7}$$

We next compute $\bar{V}''(\theta^o)$. By differentiating (2.1) twice

$$\bar{V}''(\theta^o) = E[\Psi\Psi^T] + E\left[\epsilon\frac{\partial^2\epsilon}{\partial\theta^2}\right]$$
$$= E[\Psi\Psi^T] - E\left[(y - y^o)\frac{\partial^2 y^o}{\partial\theta^2}\right] \tag{3.8}$$

all terms being evaluated at θ^o. If this is extended to the general case where $\bar{V}''(\theta^o)$ may be singular, the trace term present in (2.10) can be rewritten as

$$\begin{aligned}&\operatorname{tr}(\bar{V}''(\theta^o)PP_\theta)\\&\quad=\lambda_o\operatorname{tr}(P\bar{E}[\Psi\Psi^T]\bar{V}''^{+})+\operatorname{tr}(P\bar{E}[B]\bar{V}''^{+}) \quad (3.9)\\&\quad=\lambda_o\bar{p}+p_b=\bar{p}\left(\lambda_o+\frac{p_b}{\bar{p}}\right)=\bar{p}(\lambda_o+\lambda_b). \quad (3.10)\end{aligned}$$

We define $\bar{p}$ to be the effective number of parameters used by the model to fit the data (the rank is full whenever $\bar{p}=p_k$) and λ_b to be a virtual variance associated with the bias. If $\bar{p}=0$ (as may happen during training) we should consider p_b instead of λ_b.

Finally, we can rewrite (2.10) as

$$\begin{aligned}J(M_k)&=V_N(\hat{\theta},Z^N)+\frac{\lambda_o\bar{p}+p_b}{N}\\&=V_N(\hat{\theta},Z^N)+\frac{\bar{p}}{N}(\lambda_o+\lambda_b) \quad (3.11)\end{aligned}$$

For large N we can substitute in (3.8) the expectation $E[\Psi\Psi^T]$ with its empirical value $\bar{E}[\Psi\Psi^T]$ (now evaluated at $\hat{\theta}$) and an estimate of the effective number of parameters can be computed as

$$\hat{\bar{p}}=\operatorname{tr}(\hat{P}\bar{E}[\Psi\Psi^T](\bar{E}[\Psi\Psi^T]-\bar{E}_\epsilon)^{+}) \quad (3.12)$$

$$\hat{P}=(\bar{E}[\Psi\Psi^T]-\bar{E}_\epsilon)(\bar{E}[\Psi\Psi^T]-\bar{E}_\epsilon)^{+}$$

$$\bar{E}_\epsilon=\frac{1}{N}\sum_{i=1}^{N}\epsilon_i\frac{\partial^2\hat{y}(\theta,x_i)}{\partial\theta^2} \quad (3.13)$$

evaluated at $\hat{\theta}$. We can further approximate (3.12) to reduce the computational complexity in the evaluation of $\hat{\bar{p}}$ by considering the quasi-Newton Hessian [i.e., neglecting the second term in (3.8) and, therefore, in (3.13)]: $\hat{P}$ becomes the approximated $\hat{P}_{\mathrm{QN}}$ and the approximated effective number of parameters is

$$\hat{\bar{p}}=\operatorname{tr}(\bar{E}[\Psi\Psi^T]\bar{E}[\Psi\Psi^T]^{+})=\operatorname{tr}(\hat{P}_{\mathrm{QN}})=\operatorname{rank}(\bar{E}[\Psi\Psi^T]). \quad (3.14)$$

The effective number of parameters as defined in (3.14) has also been suggested in [26] under the hypothesis of a non-singular quasi-Newton Hessian. Unfortunately, this is not the case in real applications, where the matrix is generally singular [see also observations following (2.10)]. To overcome such problems, different authors (see [27] for instance) evaluate the effective number of parameters by counting the number of non-null weights and biases. This procedure is definitely not correct and either (3.12) or (3.14) should be used. Moreover, since (3.14) does not always provide a good approximation of the effective number of parameters, (3.12) should be used in preference. The two entities coincide when the term given in (3.13) is null. This happens when dealing with linear models or when the error surface has a relatively constant curvature in the $\hat{\theta}$ neighborhood and/or the network fits the data well. If the punctual bias is null when N tends to infinity, then $\bar{E}_\epsilon=0$ and we say that the real function to be approximated almost belongs to M_k. In a noise-free environment this always happens by allowing the number of hidden units to increase freely, since the chosen neural models are universal function approximators [18]. The second term in (3.13) may also become null with reduced N in overdimensioned models if the network overfits the training data in the long training run (see results of Section IV).

1) The Criterion in the Noise-Free Case: Let us now compare (2.12) with (3.11). Whenever the process generating the data is not affected by noise ($\lambda_o=0$), the model selection, as suggested by Moody [(2.12)], is unrealistic since it is based only on the training error. Equation (3.11) provides a more robust model selection by considering a corrective term, which depends on the complexity of the model and the number of data samples. The criterion in the pure bias case directly derives from (3.11)

$$\mathrm{FPEB}=V_N(\hat{\theta},Z^N)+\frac{\hat{\bar{p}}}{N}\hat{\lambda}_b=V_N(\hat{\theta},Z^N)+\frac{\hat{p}_b}{N} \quad (3.15)$$

where the effective number of parameters is given either in (3.12) or (3.14) and $\hat{p}_b$ comes from (3.10) by substituting expectations with the empirical quantities of (3.7) and (3.13). It should be noted that, in a noise-free case, we simply have $\epsilon_i=\rho_i$: a known quantity.

As a simple example let us consider V_N, as defined in (3.3), with N examples uniformly extracted from the $[-1,1]$ interval, $\bar{y}=x$, $\lambda_o=0$ and the model family $M_a=\{\hat{y}=a, a\in\Re\}$. We have seen that the best approximation is $y^o=0$. From (3.4) we have that $\Psi(x,\theta^o)=1$ and therefore, from (3.7) $\bar{E}[B]=\hat{\rho}_o=N^{-1}\sum_{i=1}^{N}\rho(x_i,\theta^o)^2$. From (3.12) $\hat{\bar{p}}=1$ and $\lambda_b=p_b=\hat{\rho}_o$. Equation (3.15), therefore, finally provides the criterion

$$\mathrm{FPEB}=V_N+\frac{\hat{\rho}_o}{N}.$$

2) The Criterion in the Noise-Plus-Bias Case: In the case of noise plus distortion, if the variance λ_o of the noise is known, the analysis is straightforward and we have to add $\frac{\hat{\bar{p}}\lambda_o}{N}$ to criterion (3.15). Conversely, if the variance of the noise is unknown, it must be estimated. By definition, we have that

$$2\bar{V}(\theta^o)=E[\epsilon^2(x,\theta^o)]=\lambda_o+\rho_o \quad (3.16)$$

with $\rho_o=E[\rho(x,\theta^o)^2]$. By considering (2.8) with terms coming from (3.10) and (3.16), we have that

$$V_N(\hat{\theta})=\frac{\lambda_o+\hat{\rho}_o}{2}-\frac{\bar{p}(\lambda_o+\lambda_b)}{2N} \quad (3.17)$$

and, therefore, the estimate $\hat{\lambda}_o$ becomes

$$\hat{\lambda}_o=\frac{2V_N(\hat{\theta},Z^N)+\lambda_b\bar{p}/N-\rho_o}{1-\bar{p}/N}. \quad (3.18)$$

By substituting (3.18) in (3.11), we obtain the expression for the criterion in the noise-plus-bias case

$$\begin{aligned}J(M_k)&=V_N(\hat{\theta},Z^N)\frac{N+\bar{p}}{N-\bar{p}}+\frac{p_b-\rho_o\bar{p}}{N-\bar{p}}\\&=\mathrm{FPE}_b+[\lambda_b-\rho_o]\frac{\bar{p}}{N-\bar{p}} \quad (3.19)\end{aligned}$$

where FPE_b is the final prediction error term structurally similar to the well-known FPE for unbiased models. Even if the expression is similar to FPE, it should be noted that now

V_N contains also the bias contribution (and thus $\text{FPE}_b \neq \text{FPE}$). When N grows indefinitely, the training error becomes a good estimate of the generalization error.

The criterion suggested in (3.19) relies on the hypothesis that ρ_o and λ_b are available or estimable. If this is not the case, we can address the problem by assuming that also the bias ρ is an i.i.d. variable with zero mean and an unknown λ_ρ variance. The hypothesis relies on the fact that it is reasonable to assume both positive and negative punctual biases with a null expectation and that the effect caused by a punctual bias added to the noise is equivalent to a realization of a different noise with an increased variance [this is due the approximation introduced in (2.9) when deriving (2.10)]. The analysis is now straightforward by simply considering the additive contribution in the two random variables ρ and ϵ in (3.1) and (3.2), whose effect is equivalent to a random variable η with zero mean and $\lambda_\eta = \lambda_o + \lambda_\rho$ variance. Without presenting all details, we can repeat the procedure accomplished in Section III-A. Briefly, (3.11) becomes

$$J(M_k) = V_N(\hat{\theta}, Z^N) + \frac{\bar{p}\lambda_\eta}{N} \tag{3.20}$$

with

$$\hat{\lambda}_\eta = \frac{2V_N(\hat{\theta}, Z^N)}{1 - \bar{p}/N}. \tag{3.21}$$

As a consequence, the final criterion becomes formally similar to FPE

$$\text{FPEB} = \text{FPE}_b = V_N \frac{N + \hat{\bar{p}}}{N - \hat{\bar{p}}} \tag{3.22}$$

where V_N accounts both for the noise on data and the bias. The effective number of parameters is, again, either that of expression (3.12) or (3.14). Whenever the model is unbiased, we have that $\lambda_b = 0$ and $\rho_o = 0$, V_N does not contain any bias contribution and the criterion reduces to FPE.

The hypothesis of assuming the bias as a random variable, even if appealing, is not correct from a theoretical point of view and we should consider (3.19). Without specializing the effects of noise and bias, we can still derive from (2.10) a criterion formally similar to NIC.

By substituting (2.5) in (2.10) and expectations with empirical quantities, the criterion becomes

$$\text{FPEB} = V_N(\hat{\theta}) + \frac{1}{N}\,\text{tr}(\hat{P}\bar{E}[Q(\hat{\theta})](\bar{E}[V''(\hat{\theta})])^+) \tag{3.23}$$

where

$$\bar{E}[Q(\hat{\theta})] = \frac{1}{N}\sum_{i=1}^{N} \epsilon_i^2 \Psi_i(\hat{\theta})\Psi_i(\hat{\theta})^T \tag{3.24}$$

and

$$\bar{E}[V''(\hat{\theta})] = \bar{E}[\Psi\Psi^T] - \bar{E}_\epsilon. \tag{3.25}$$

Once again, the number of parameters used by the model is that of (3.12). The validity of the presented criteria relies on the validity of substituting expectations with empirical values. In other words, this is equivalent to assuming that P_θ (or PP_θ) can be reasonably estimated from the available data. Of course, if N is sufficiently large, then the approximation holds because of the law of large numbers. The validity of estimating the covariance matrix from the available data $\hat{P}_\theta = P(Z^N)_{\hat{\theta}}$ for finite N was studied in [16], with respect to ARX models. By using Monte Carlo simulations, the authors obtained good approximations of P_θ with only 50 data instances. Similar results were also obtained in [28]. Monte Carlo experiments should also be performed for the case of nonlinear systems, to determine experimentally the impact of finite N on P_θ. Experiments presented in Section IV prove the validity of the framework outlined in this paper. It seems, therefore, reasonable to assume that we would obtain a good approximation of P_θ with only a few tens of data.

B. Relationships Between Moody's GPE and FPEB

Under Moody's assumptions, we can estimate the generalization error of (2.10) with the one provided in (2.12)

$$E[\bar{V}(\hat{\theta})] \approx E[V_N]_{\text{Val}} \tag{3.26}$$

and therefore, from (3.10) and (3.16), we obtain a relationship among $\lambda_o, \rho_o, \lambda_b, \bar{p}$ and p_{eff}

$$\frac{\lambda_o + \rho_o}{2} + \frac{\bar{p}(\lambda_o + \lambda_b)}{2N} = V_N(\hat{\theta}) + \lambda_o \frac{p_{\text{eff}}}{2N}. \tag{3.27}$$

Equations (3.17) and (3.27) constitute a linear system whose solution gives the estimates

$$\hat{\lambda}_o = \frac{2V_N(\hat{\theta}, Z^N) - \rho_o}{1 - p_{\text{eff}}/2N}; \quad \hat{\lambda}_b = \hat{\lambda}_o\left(\frac{p_{\text{eff}}}{2\bar{p}} - 1\right) \tag{3.28}$$

and, therefore, (3.11) becomes

$$J(M_k) = \text{GPE} = V_N(\hat{\theta}, Z^N) + \hat{\lambda}_o \frac{\hat{p}_{\text{eff}}}{2N} \tag{3.29}$$

which is equivalent to the GPE given in (2.13), with (3.28) as the estimate of variance.

Finally, we could use the estimates of (3.28) to reformulate (in this case) the FPEB

$$\text{FPEB} = \text{FPE}_b - \rho_o \frac{\hat{p}_{\text{eff}}}{2N - \hat{p}_{\text{eff}}} \tag{3.30}$$

where $\text{FPE}_b = V_N(2N + \hat{p}_{\text{eff}})/(2N - \hat{p}_{\text{eff}})$. Equation (3.30) states that, under Moody's assumptions, the final prediction error in the biased case is structurally equivalent to FPE_b, corrected by a factor depending on the effective number of parameters and on the bias degree ρ_o.

C. The Impact of Training Time in the Use of FPEB

In all these criteria, whenever the network is overdimensioned with sufficient degrees of freedom with respect to the given application, the term V_N decreases asymptotically with training epochs. On this basis, we should identify the optimal network as the one obtained after infinite training time.

With respect to a simple gradient based procedure, and under the hypothesis that the learning coefficient tends to zero and that the number of training epochs tends to infinity, Amari, in [15], proved that for a sufficiently large training time t_{tr}, $\hat{\theta}(t_{\text{tr}})$ has a Gaussian distribution whose expectation

is a point minimizing V_N with variance proportional to the learning coefficient. The proof is only valid for gradient descent algorithms and does not imply that the best estimate of θ^o is obtained after infinite training but simply that, with a resampling training procedure, we will converge to a point minimizing V_N. In fact, it is not true that after infinite training epochs we necessarily end in a good minimum (i.e., $\hat{\theta}$ may be a bad estimate of θ^o) because of overtraining. We already discussed the issue in observations following (2.5). The problem can be solved by implementing an early stopping strategy based on the effective number of parameters.

Experimentally, we have seen that training should be stopped whenever $\hat{\bar{p}}$ converges and/or $J(M_k(\hat{\theta}))$ is constant or increases. The rationale behind this is that $\hat{\bar{p}}$ represents the effective number of degrees of freedom used by the model to infer the unknown function from the input/output pairs. $\hat{\bar{p}}$ evolves during the early stages of training as if it were driven by an internal growing/optimizing algorithm, which provides the appropriate degrees of freedom. Heuristically, when $\hat{\bar{p}}$ converges and $J(M_k(\hat{\theta}))$ is constant or increases, the training procedure needs to be stopped and, at that time, the associated entities (e.g., $\hat{\theta}, \hat{\bar{p}}, \hat{\rho}$, etc.) should be used in the criteria. A different theory, dealing with the determination of the optimal stopping point, has been developed in [32] where it has been proven that the optimal point $\hat{\theta}(t_{\rm tr})$ belongs to a $1/\sqrt{(}N)$ neighborhood of θ^o. The two criteria are equivalent since, in such a neighborhood, (2.7), (2.8) and the following relationships are valid.

IV. Learning Input/Output Relationships

In this section we apply the criteria proposed above to determine the optimal neural topology in three applications.

The first application deals with a smooth function in a noise-free set up. The second application also refers to a noise-free case but, in this example, the function to be learned is quite irregular and the data do not contain sufficient information to properly configure the neural model. In the third application, the function to be learned is affected by noise.

Selection of the optimal topology according to FPEB requires the training of several hierarchical models (the hierarchy can be obtained by increasing the number of hidden units). Obviously, the training procedure is time consuming and training a large subset of the model structure may be computationally infeasible. Heuristics can therefore be used to guide the search toward the determination of a reduced subset of models. To this end, two different approaches can be found in [2] and [29]. Here, we implemented the second one. Briefly, instead of training and evaluating the criterion for each different neural topology, we apply an OBD-like technique [25] to connections ending in the output neuron of a strongly overdimensioned topology. The hidden layer complexity can be reduced (thus exploring the hierarchy) by removing the connectio,n after which the increase in MSE is minimized. A new topology is given and the criterion can be applied without requiring a new training phase. This OBD-like process iterates until the number of hidden units is equal to one. The error in estimating the criterion without any effective training increases as the number of hidden neurons decreases, but this strategy provides useful guidelines in reducing the number of models to be considered. Generally, in a few iterations we can identify a small set of candidate topologies containing the optimal one.

Training was implemented with an optimized Levenberg–Marquardt learning algorithm. In the rest of the section, we will compare several criteria which are derived from FPE or FPEB, by considering the novel definition of the effective number of parameters. More specifically, the following are the criteria.

1) *FPE*: the criterion is the well-known FPE for which p is the number of nonnull degrees of freedom of the network;
2) *FPE1*: the criterion is the FPE for which the number p of parameters has been evaluated according to (3.14) (we consider the approximated version of $\bar{p}$).
3) *FPE2*: the criterion is the FPE for which the number p of parameters has been evaluated according to (3.12) (we consider the correct $\bar{p}$).
4) *FPEB*: the criterion is the one given in (3.15) for the pure bias case and the one in (3.23) for the most general case with the approximated effective number of parameters from (3.14).
5) *FPEB2*: the criterion is the one given in (3.15) for the pure bias case and the one in (3.23) for the general case with the correct effective number of parameters from (3.12). This criterion is the most accurate one.

A. Example 1: A Reduced-Bias Case

Our goal is to approximate the nonlinear function

$$\bar{y}(x) = -x\sin(x^2) + \frac{e^{-0.23x}}{1+x^4}. \tag{4.1}$$

For this example, the training set was composed of $N = 80$ pairs uniformly extracted from the $[-2, 2]$ interval. The function is smooth and the training set rich enough to guarantee that (4.1) almost belongs to the neural model family. No noise was added to the data: this is a pure bias case.

The heuristic outlined above was used to consider topologies with hidden units ranging from one to seven. For each neural model we have to apply the training procedure and then compute the criteria. According to the early stopping strategy, we monitored the evolution of the correct and the approximated effective number of parameters during training time. The learning phase lasted for 1000 epochs (each epoch implements two minimizations of the training error). In the following plots, nh indicates the number of hidden units and p the asymptotic value (with respect to the training epochs) assumed by $\bar{p}$.

The evolution of the correct effective number of parameters for each topology of the subset is given in Fig. 1. We can immediately see that with $nh \leq 3$, the network utilizes all the degrees of freedom available, whereas for networks with higher complexity, the effective number of parameters is less than the maximum (for nh hidden neurons we have $2nh$ weights and $nh + 1$ biases). The behavior of the models with $nh > 3$ is such that $\bar{p}$ first evolves during the initial

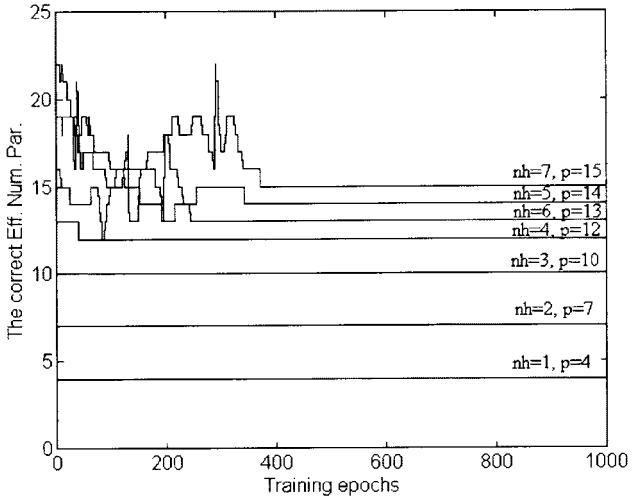

Fig. 1. The correct effective number of parameters.

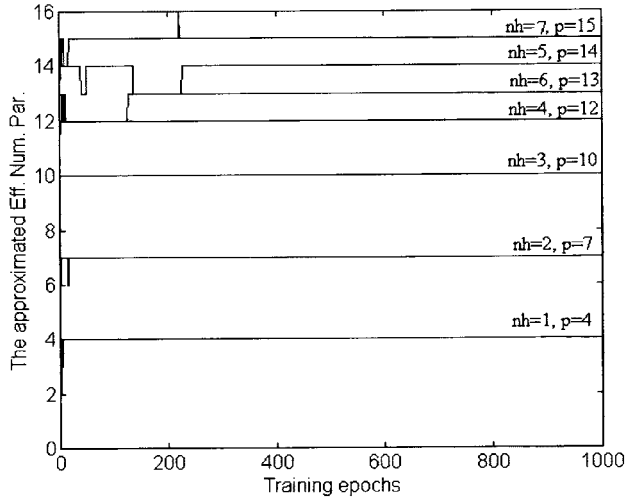

Fig. 2. The approximated effective number of parameters.

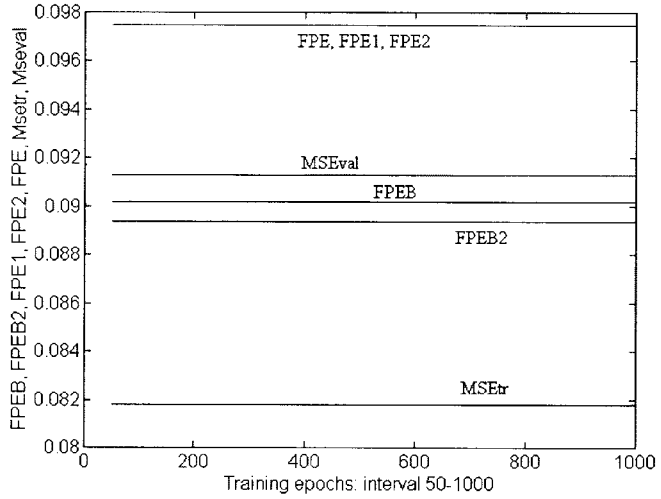

Fig. 3. Comparing different criteria for the model with $nh = 2$.

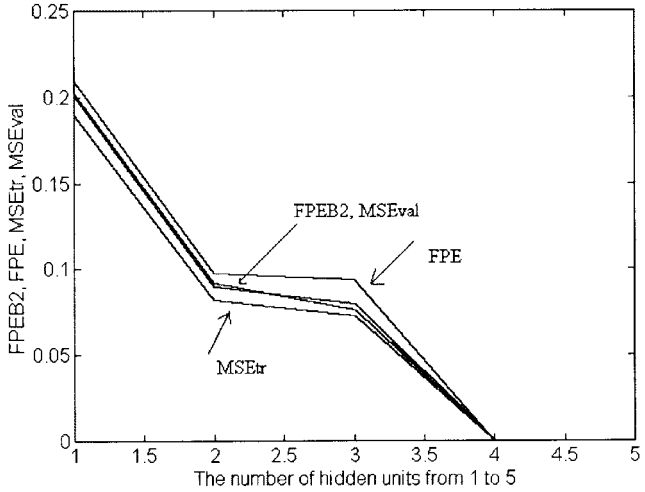

Fig. 4. Different criteria for the models from one to five hidden units.

stages of learning, before reaching a steady state (note that for the $nh = 6$ experiment the model degenerated to a model with lower complexity). The evolution of the approximated effective number of parameters is given in Fig. 2.

The approximation does not consider the E_ϵ term of expression (3.13). Since the function almost belongs to the neural model family, after a small number of training epochs we have that $E_\epsilon = 0$ and the correct and the approximated $\bar{p}$ coincide (as happens) in the long training run.

Comparisons among different criteria are given in Fig. 3 for the case $nh = 2$ where we indicated the MSE validation with MSEval and MSE training with MSEtr. We can see that, despite the approximation leading to (2.10), FPEB and FBEB2 are reasonable estimates of the MSE validation (evaluated over the whole definition interval) while all other criteria provide a worse estimate. The effective number of parameters and the criteria have been determined with the early stopping strategy suggested in Section III-C. The most accurate criteria are then compared in Fig. 4 for neural models with hidden units varying from one to five. For all models, FPEB2 provided a good estimate of the generalization ability, while all criteria selected the model with $nh = 4$, in full agreement with the validation plot. Actually, there is a very small improvement in performance when increasing the number of hidden units, but we are interested in the smallest model keeping the same performance. The training data (circled), the function to be learned, and the best neural model selected by the criteria are plotted in Fig. 5.

B. Example 2: A High-Bias Case

In this experiment, we drastically reduced the number of training data to 32 and enlarged the definition interval to $[-5, 15]$. Data was regularly sampled from the function

$$\bar{y}(x) = \sqrt{0.1\sin(2x)^2 + \frac{2\,\mathrm{atan}(x-3)^2}{(x+7)(\cos(x)+2)}} \qquad (4.2)$$

and no noise was added. This function is definitely more irregular than the previous one (see Fig. 11). The OBD-like procedure identified the interval from one to ten hidden units. To monitor overtraining effects, we have to track the evolution of the correct (Fig. 6) and the approximated (Fig. 7) effective number of parameters. In the figures, nh and p increase from the bottom to the top of the plots. It can be seen that now there is a difference in the estimated effective number of parameters

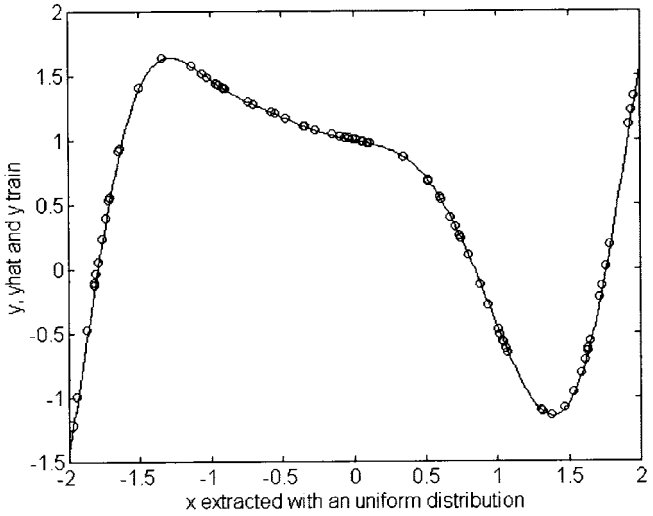

Fig. 5. The training data, the real function and the best neural model.

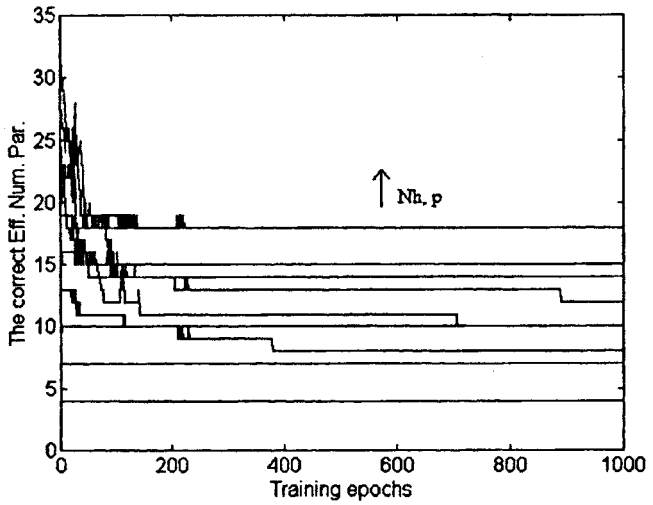

Fig. 6. The correct effective number of parameters.

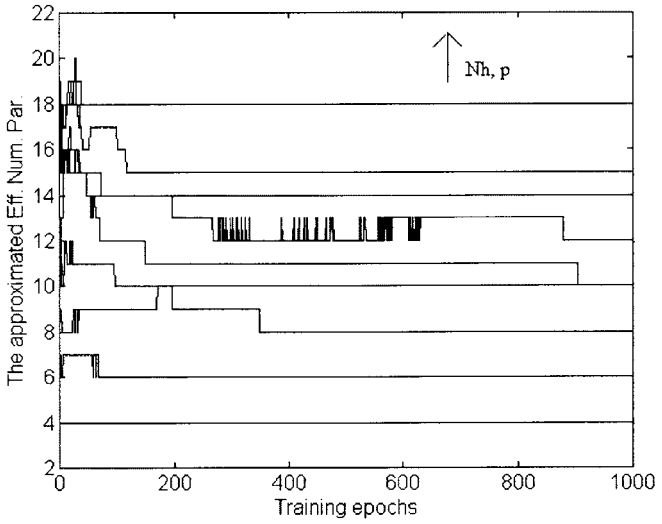

Fig. 7. The approximated effective number of parameters.

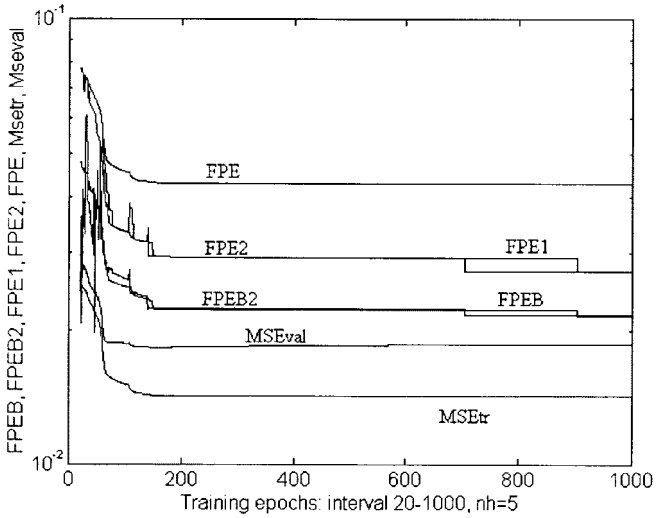

Fig. 8. Evolution of the criteria in the long training run for the model with $nh = 5$.

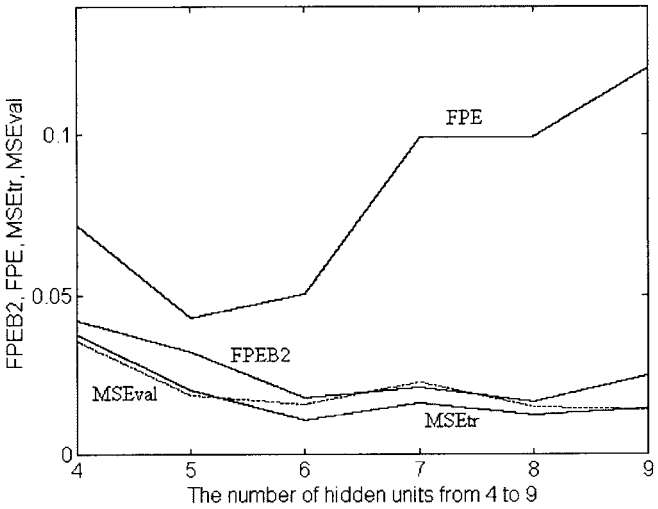

Fig. 9. Different criteria for the models from four to nine hidden units.

for the case $nh = 2$ (we have $p = 7$ for the correct and $p = 6$ for the approximated one in the long training run). Since for high values of nh the network is overdimensioned, the stopping points suggested by the two estimates are different. For overdimensioned models, the best stopping point is at the early stages of learning where the correct and the approximated effective number of parameters differ. we should, therefore, always consider the correct $\bar{p}$.

In Fig. 8, the evolution of different criteria over the training time for the case $nh = 5$ is plotted on a semilog scale. We realize, once more, that the FPEB's provide better estimates of the generalization ability of the model (again there is a discrepancy between FPEB and FPEB2 because of the difference in $\bar{p}$). FPE, itself, is always the worst criterion. We determined the best criterion FPEB2 and FPE (the worst one) on different network topologies with hidden units varying from four to nine. Results are given in Fig. 9. FPE selected the model with five hidden units, while FPEB2 selected the one with six. The model with eight hidden units almost provides the same performance (according to the criterion) as the one with six, but it requires a more complex model. On the other

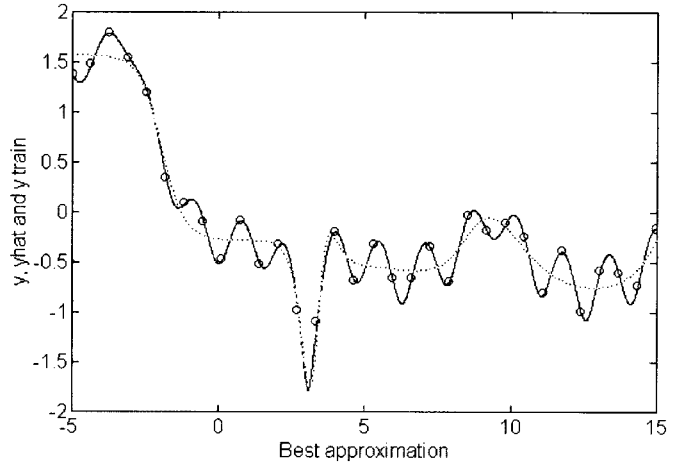

Fig. 10. The training data, the real function, and the best neural model.

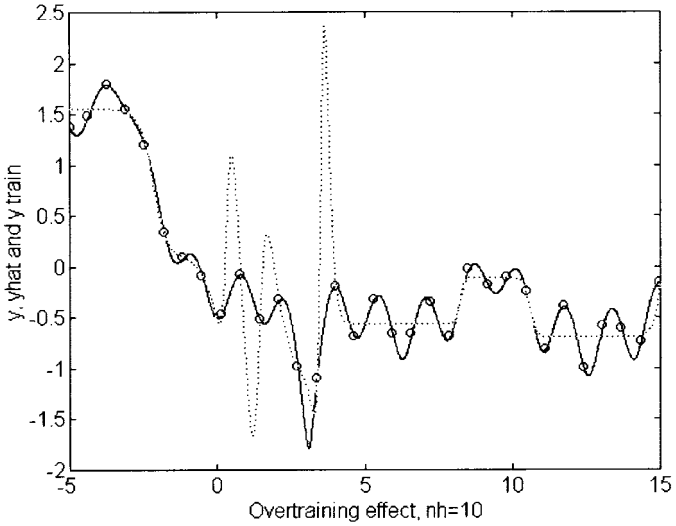

Fig. 11. Overtraining effects for the model with ten hidden units.

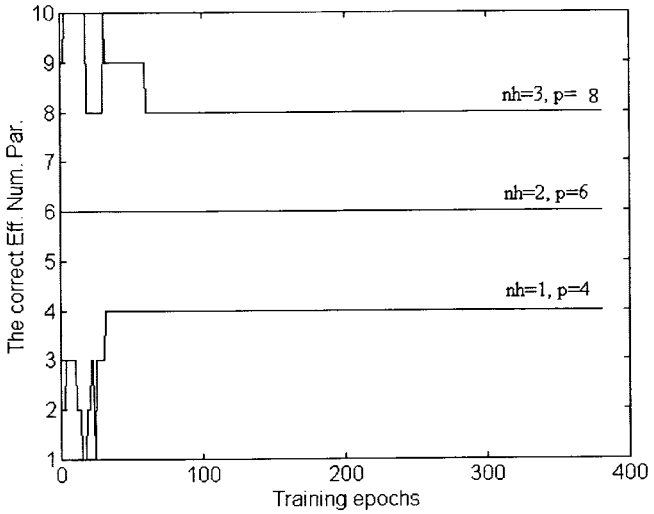

Fig. 12. The correct effective number of parameters.

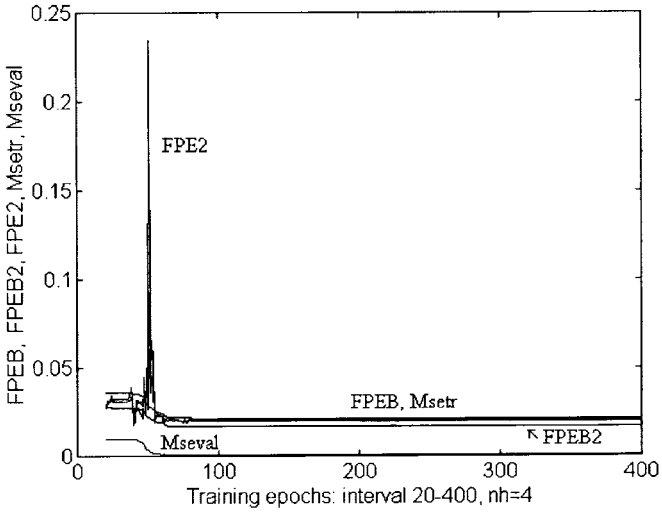

Fig. 13. Evolution of the criteria in the long training run for the model with $nh = 4$.

hand, the criterion will penalize such a model if the model complexity is overdimensioned with respect to the number of data elements. The selected model is then the one which best solves the performance/model complexity tradeoff according to (3.23), even if this may imply the selection of a model with a high bias (see Fig. 10). Validation results support the selections made and prove the efficacy of estimating the validation error with FPEB2. The best approximation, as suggested by the criterion, is given in Fig. 10 where the training data are circled, the real function of (4.2) is plotted in a continuous line, and the best estimate with a dotted line.

If the learning termination point is not correctly selected (e.g., according to the early stopping strategy based on $\bar{p}$ and the criteria), we could easily end up with overtrained networks. An example is the plot of Fig. 11, obtained after only 400 training epochs for the $nh = 10$ topology. We can see that after too much training $\hat{\theta}$ is a bad estimate of θ^o.

C. Example 3: Noise and Bias

As a third example, we consider the function

$$\bar{y}(x) = 4.26(e^{-x} - 4e^{-2x} + 3e^{-3x}) \tag{4.3}$$

as suggested in [17]. A set of $N = 100$ x's were uniformly extracted from the $[0, 2]$ interval and the associated y's were corrupted with a Gaussian noise with zero mean and $\lambda_o = 0.04$ variance. We assume that the process generating the data is unknown. Since we do not have *a priori* knowledge, we have to consider the most general criterion given in (3.23). The same heuristic suggests examination of models with from one to five hidden neurons. As with previous cases, we monitored the evolution of the correct effective number of parameters. Fig. 12 shows the 1–400 training epoch interval for models with from to three hidden units. We can see that $\bar{p}$ for the case $nh = 1$ reaches one after some training epochs and then it converges by using all the degrees of freedom provided by the model (i.e., four). The model with two hidden units uses only six parameters out of seven, while in the case of $nh = 3$, $\bar{p}$ decreases and converges to the steady state with eight parameters out of ten. It is easy to determine the learning termination points for such topologies. The behavior of the most relevant criteria over the training run is given in Fig. 13 for the case $nh = 4$. The behavior of the criteria with respect to different neural topologies (from one to five hidden units) are plotted in Fig. 14. We

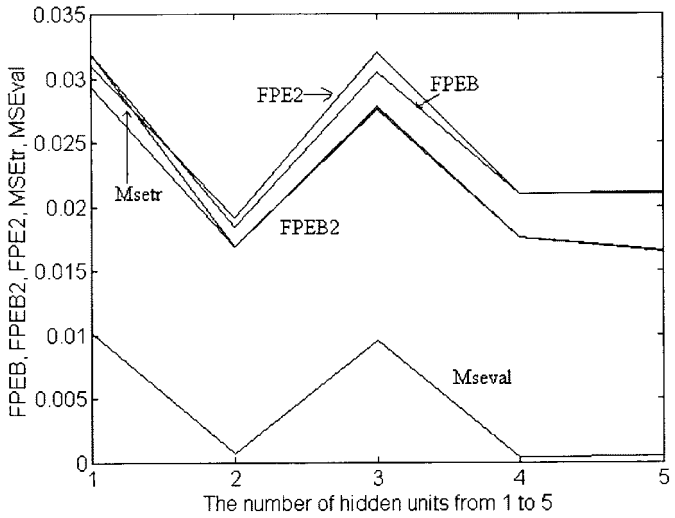

Fig. 14. Different criteria for the models from one to five hidden units.

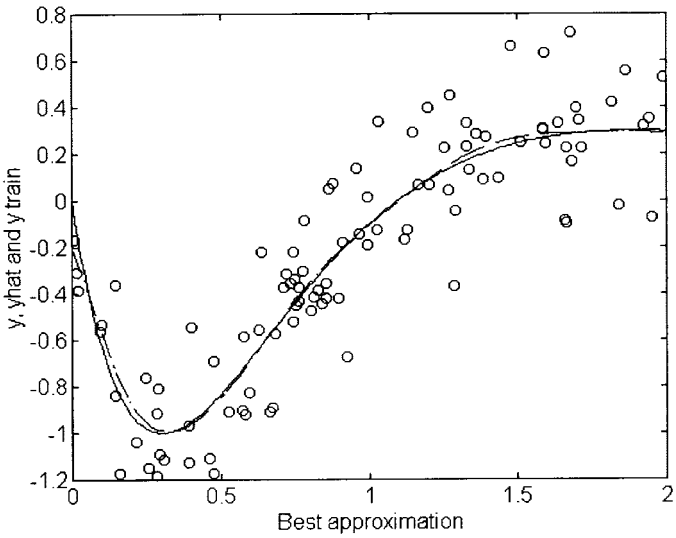

Fig. 15. The training data, the real function, and the best neural model.

can see that the networks with two and four hidden units provide the same performance. Since we choose the simplest model when determining the complexity/performance tradeoff, we consider only the network with two hidden units. The best estimate is given in Fig. 15 (training data: circled, real function: continuous line, best approximation: dashed line).

V. Conclusion

In this paper we presented a theoretical framework which provides effective criteria to select and validate neural topologies for learning an unknown function. A generalization of the FPE criterion to biased models has been introduced, which is shown to be related to the one suggested by Moody. The criteria solve problems posed by NIC. This has been achieved by suitably estimating the covariance matrix of the parameters with the Moore–Penrose pseudoinverse by introducing a novel definition of the effective number of parameters and by implementing an early learning termination strategy which helps prevent overtraining.

Appendix A

To calculate the parameter covariance matrix $P_\theta = E[(\theta - \theta^o)(\theta - \theta^o)^T]$ of (2.3) and (2.5) we need to expand $V'_N(\hat{\theta}, Z^N)$ around θ^o (to which $\hat{\theta}$ converges) with Taylor and evaluate the expansion at $\hat{\theta}$. Since $V'_N(\hat{\theta}, Z^N) = 0$ (the learning procedure ends in a minimum) the expansion provides

$$0 = V'_N(\hat{\theta}, Z^N) = V'_N(\theta^o, Z^N) + V''_N(\bar{\chi}, Z^N)(\hat{\theta} - \theta^o) \quad \text{(A.1)}$$

where each component of the $\bar{\chi}$ vector is within a sphere of radius $\|\hat{\theta} - \theta^o\|$ centred on θ^o.

In the limit, when N tends to infinity $\bar{\chi}$ tends to θ^o and V_N converges uniformly to $\bar{V}$ with probability 1 in D from *R1*. Under the regularity hypothesis, this convergence also holds for the Hessians and therefore V''_N converges to $\bar{V}''$ (see [19] and [20] for the proof). (A.1) thus becomes

$$V'_N(\theta^o, Z^N) + \bar{V}''(\theta^o)(\hat{\theta} - \theta^o) = 0. \quad \text{(A.2)}$$

(A.2) constitute a linear system whose Moore–Penrose solution in the mean square sense is [30], [31]

$$(\hat{\theta} - \theta^o) = -\bar{V}''^{+}(\theta^o)V'_N(\theta^o, Z^N) + (I - P)v \quad \text{(A.3)}$$

where I is the identity matrix, $P = \bar{V}''^{+}(\theta^o)\bar{V}''(\theta^o)$ is an idempotent matrix ($P^2 = P$) orthogonal projector onto the column/row space of $\bar{V}''$, and v is an arbitrary p-dimensional column vector.

From simple manipulations and remembering that $P(I - P) = 0$, we can write that

$$PP_\theta = E[P(\theta - \theta^o)(\theta - \theta^o)^T] = \bar{V}''^{+}(\theta^o)Q\bar{V}''^{+}(\theta^o) - \bar{V}''^{+}(\theta^o)E[V'_N(\theta^o, Z^N)]v^T(I - P). \quad \text{(A.4)}$$

The second term of (A.4) can be neglected since $E[V'_N(\theta^o, Z^N)]$ is asymptotic to $\bar{V}'(\theta^o)$, which is null (see also [15]) and (2.5) is proved. When $\bar{V}''^{+}(\theta^o)$ is nonsingular we have simply that $P = I$ and we obtain (2.3).

Appendix B

With respect to (2.6) we note that $J(M_k)$ and $\bar{V}(\theta)$ are not known, since we do not know the true data. Such quantities will therefore be approximated by using Taylor expansions in the Lagrange notation around the minimum θ^o (to which $\hat{\theta}$ will converge) of $\bar{V}(\theta)$. Recalling that $\bar{V}'(\theta^o) = 0$ and evaluating the expansion for $\theta = \hat{\theta}$, we obtain

$$\bar{V}(\hat{\theta}) = \bar{V}(\theta^o) + \frac{1}{2}(\hat{\theta} - \theta^o)^T\bar{V}''(\chi)(\hat{\theta} - \theta^o) \quad \text{(B.1)}$$

where χ is a point whose components lie within a sphere of radius $\|\hat{\theta} - \theta^o\|$ centered on θ^o. We recall that (A.1) holds

$$V'_N(\hat{\theta}, Z^N) = V'_N(\theta^o, Z^N) + V''_N(\bar{\chi}, Z^N)(\hat{\theta} - \theta^o) = 0. \quad \text{(B.2)}$$

We expand $V_N(\theta, Z^N)$ around θ^o, consider $V'_N(\theta^o, Z^N)$ as given by (B.2), and evaluate the expansion for $\theta = \hat{\theta}$

$$V_N(\hat{\theta}) = V_N(\theta^o) - \frac{1}{2}(\hat{\theta} - \theta^o)^T V''_N(\bar{\bar{\chi}})(\hat{\theta} - \theta^o) \quad \text{(B.3)}$$

where $\bar{\bar{\chi}}$ is a convenient point similar to χ and $\bar{\chi}$. We now take expectations of (B.1) and (B.3) by considering the asymptotic relationships given by *R1* and expression (2.2), thus obtaining

$$\begin{aligned} &E[(\hat{\theta}-\theta^o)^T\bar{V}''(\chi)(\hat{\theta}-\theta^o)] \\ &\quad = E[\mathrm{tr}(\bar{V}''(\chi)(\hat{\theta}-\theta^o)(\hat{\theta}-\theta^o)^T)] \\ &\quad \approx \mathrm{tr}([\bar{V}''(\theta^o)P_N]) \end{aligned} \tag{B.4}$$

$$\begin{aligned} &E[(\hat{\theta}-\theta^o)^T V_N''(\bar{\chi})(\hat{\theta}-\theta^o)] \\ &\quad = E[\mathrm{tr}(V_N''(\bar{\chi})(\hat{\theta}-\theta^o)(\hat{\theta}-\theta^o)^T)] \\ &\quad \approx \mathrm{tr}([\bar{V}''(\theta^o)P_N]) \end{aligned} \tag{B.5}$$

being $P_N = P_\theta/N$ and tr the matrix trace. From (2.1) we recall that

$$E[V_N(\theta^o, Z^N)] = \bar{V}(\theta^o). \tag{B.6}$$

From (B.1) to (B.6), we obtain finally

$$E[\bar{V}(\hat{\theta})] \approx \bar{V}(\theta^o) + \frac{1}{2}\,\mathrm{tr}([\bar{V}''(\theta^o)P_N]) \tag{B.7}$$

$$E[V_N(\hat{\theta}, Z^N)] \approx \bar{V}(\theta^o) - \frac{1}{2}\,\mathrm{tr}([\bar{V}''(\theta^o)P_N]) \tag{B.8}$$

from which (2.7) and (2.8) follow by noting that $\bar{V}''(\theta^o) = \bar{V}''(\theta^o)\bar{V}''^{+}(\theta^o)\bar{V}''(\theta^o) = \bar{V}''(\theta^o)P$.

ACKNOWLEDGMENT

The author wishes to thank the reviewers for their insights and fruitful suggestions in early versions of the paper and J. Taylor and Dr. K. Sammut for help in improving its readability.

REFERENCES

[1] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*. Reading, MA: Addison-Wesley, 1991.
[2] J. Moody and J. Utans, "Principal architecture selection for neural networks: Application to corporate bond rating prediction," in *Proc. NIPS4*, San Mateo, CA, 1992, pp. 683–690.
[3] A. G. Parlos, K. T. Chong, and A. F. Atiya, "Application of the recurrent multilayer perceptron in modeling complex process dynamics," in *IEEE Trans. Neural Networks*, vol. 5, pp. 255–266, Mar. 1994.
[4] M. Stinchombe and H. White, "Approximating and learning unknown mappings using multilayer feed-forward networks with bounded weights," in *Proc. IJCNN'90*, 1990, vol. 3, pp. 7–16.
[5] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," *Neural Computat. 1994*, vol. 7, 1995.
[6] F. Girosi, M. Jones, and T. Poggio, "Priors, stabilizers and basis functions: From regularization to radial, tensor and additive splines," Dept. Brain and Cognitive Sciences, MIT, Cambridge, MA, A.I. Memo 1430, 1993.
[7] J. L. Marroquin, "Measure fields for function approximation," , Artificial Intelligence Lab., MIT, Cambridge, MA, A.I. Memo 1433, 1993.
[8] J. Moody, "The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems," in *Proc. NIPS4*, San Mateo, CA, 1992, pp. 847–854.
[9] *IEEE Trans. Neural Networks (Special Issue on Recurrent Networks)*, vol. 5, pp. 153–337, Mar. 1994.
[10] C. Alippi and V. Piuri, "Topological minimization of multi-layered feed-forward neural networks by spectral decomposition," in *Proc. IEEE-IJCNN'92*, Beijing, China, Nov. 3–6, 1992.
[11] A. S. Weigend and D. E. Rumelhart, "The effective dimension of the space of hidden units," in *Proc. IEEE-IJCNN*, Singapore, 1991.
[12] Y. Le Cun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Proc. NIPS2*, San Mateo, CA, 1990.
[13] B. Hassibi and D. G. Stork, "Second order derivative for network pruning: Optimal brain surgeon," in *Proc. NIPS5*, San Mateo, CA, 1993.
[14] A. Sankar and R. J. Mammone, "Growing and pruning neural trees network," in *IEEE Trans. Comput.*, vol. 52, pp. 291–299, Mar. 1993.
[15] N. Murata, S. Yoshizawa, and S. Amari, "Network information criterion—Determining the number of hidden units for an artificial neural network model," in *IEEE Trans. Neural Networks*, vol. 5, pp. 865–872, Nov. 1994.
[16] L. Ljung, *System Identification, Theory for the User*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
[17] K. Hornik, M. Stinchombe, and H. White, "Multilayer feedforward networks are universal approximators," in *Neural Networks*, vol. 2, 1989.
[18] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," in *Neural Comput.*, vol. 4, pp. 1–58, 1992.
[19] L. Ljunga and P. Caines, "Asymptotic normality of prediction error estimators for approximate system models," in *Stochastic*, vol. 3, pp. 29–46, 1979.
[20] L. Ljung, "Convergence analysis of parametric identification methods," *IEEE Trans. Automat. Contr.*, vol. AC-23, no. 5, pp. 770–783, 1978.
[21] P. Caines, "Stationary linear and nonlinear system identification and prediction set completeness," *IEEE Trans. Automat. Contr.*, vol. AC-23, no. 4, pp. 583–594, 1978.
[22] S. Amari, "Statistical and information-geometrical aspects of neural learning," in *Computational Intelligence: A Dynamic Perspective*. New York: IEEE, 1995, pp. 71–82.
[23] H. Akaike, "Statistical predictor identification," in *Ann. Inst. Stat. Math.*, vol. 22, 1970.
[24] A. Barron, "Predicted squared error: A criterion for automatic model selection," in *Self-Organizing Methods for Modeling*, S. Farlow Ed. New York: Marcel Dekker, 1984.
[25] I. Guyon, V. Vapnik, B. Boser, L. Bottou, and S. A. Solla, "Structural of risk minimization for character recognition," in *Proc. NIPS4*, 1992.
[26] J. Larsen and L. K. Hansen, "Generalization performance of regularised neural network models," in *Proc. Workshop Neural Networks Signal Processing, 4*, Piscataway, NJ, 1994, pp. 42–51.
[27] M. Norgaard, "Neural network based system identification toolbox for Matlab," Inst. Automation, Tech. Univ. Denmark, Tech. Rep. 95-E-773, 1995.
[28] T. Soderstrom and P. Stoica, "Instrumental variable methods for systems identification," in *Lecture Notes in Control and Information Sciences*. New York: Springer-Verlag, 1983.
[29] C. Alippi, "Spectral decomposition, Rissanen's and Akaike's criteria to select and validate neural structures," in *Proc. World Cong. Neural Networks*, Washington, DC, 1995.
[30] G. Strang, *Linear Algebra and Its Applications*. Orlando, FL: Harcourt Brace Jovanovich, 1986.
[31] C. R. Rao, *Linear Statistical Inference and Its Applications*. New York: Wiley, 1973.
[32] S. Amari, N. Murata, K. R. Muller, M. Finke, and H. Yang, "Asymptotical statistical theory of overtraining and cross-validation," in *Proc. NIPS8*, 1996.

Cesare Alippi (S'92–M'97) received the Dr.Ing. degree in electronic engineering (*summa cum laude*) and the Ph.D. degree in computer engineering from the Politecnico di Milano, Milano, Italy, in 1990 and 1995, respectively.

He is currently an Associate Professor in Computer Science at the Politecnico di Milano and a Research Fellow in the Italian National Research Council. His research interests include neural networks (learning theories, implementation issues and applications), genetic algorithms, nonlinear signal and image processing, and VLIW architectures. His research results have been published in more that 70 technical papers in international journals and conference proceedings.