Maximum Likelihood Neural Approximation in Presence of Additive Colored Noise

Shahram Hosseini and Christian Jutten, *Associate Member, IEEE*

***Abstract*—When multilayer perceptrons (MLPs) are used for nonlinear regression, it is generally supposed that output data are corrupted by additive independent and identically distributed (i.i.d.) noise and the ordinary least square method is usually used to estimate the network weights. However, in many practical situations, the noise samples may be correlated. In this case, the estimation of noise parameters can be used to improve the approximation. Estimation of the noise structure can also be used to find a stopping criterion in constructive neural networks. To avoid overfitting, a network construction procedure must be stopped when residual can be considered as noise. The knowledge on the noise may be used for "whitening" the residual so that a correlation hypothesis test determines if the network growing must be continued or not. In this paper, supposing a Gaussian noise model, we study the problem of multi-output nonlinear regression using MLP when the noise in each output is a correlated autoregressive time series and is spatially correlated with other output noises. We show that the noise parameters can be determined simultaneously with the network weights and used to construct an estimator with a smaller variance, and so, to improve the network generalization performance. Moreover, if a constructive procedure is used to build the network, the estimated parameters may be used to stop the procedure.**

***Index Terms*—Colored noise, generalization, maximum likelihood estimation, neural networks.**

I. Introduction

In statistics, it is often desirable to find a relation between a group of variables. In general, some of these variables, called *dependent variables* or *output variables* or *responses*, denoted by $y_1, y_2, \ldots, y_M$, are of a particular interest. The other variables $x_1, x_2, \ldots, x_P$, called *independent variables* or *input variables* or *regressors*, are used to predict or to explain the behavior of $y_1, y_2, \ldots, y_M$. The relation is expressed using the functions $f_1, f_2, \ldots, f_M$

$$y_k \approx f_k(x_1, x_2, \ldots, x_P), \quad k = 1, 2, \ldots, M. \tag{1}$$

This relation remains approximative because of the influence of other unknown variables, on the one hand, and the measurement errors, on the other hand. This uncertainty is usually modeled by additive zero-mean random terms[1]

$$\begin{aligned} y_k &= f_k(x_1, x_2, \ldots, x_P) + \xi_k \\ E[\xi_k] &= 0, \quad k = 1, 2, \ldots, M \end{aligned} \tag{2}$$

which implies directly

$$\begin{aligned} E[y_k \,|\, x_1, x_2, \ldots, x_P] &= f_k(x_1, x_2, \ldots, x_P) \\ & k = 1, 2, \ldots, M. \end{aligned} \tag{3}$$

Usually, the variables evolve according to another independent variable t, which can for example correspond to time. In practice, only a limited number of samples of these variables associated to various values of t are available which form the data base.

A typical nonlinear regression problem using MLP can so be presented as following. Given a N-size data set

$$(\mathbf{x}_i, \mathbf{y}_i) = (\mathbf{x}_i, \mathbf{f}(\mathbf{x}_i) + \xi_i), \quad i = 1, ..., N \tag{4}$$

where $\mathbf{x}_i$ are samples of the input variable $\mathbf{x} \in \mathcal{R}^P$, $\mathbf{y}_i$ are samples of the output variable $\mathbf{y} \in \mathcal{R}^M$ and $\boldsymbol{\xi}_i$ represent the output noise, one wants to find a good approximation of the vector function $\mathbf{f}(\cdot)$. The noise is thus defined as the part of $\mathbf{y}$ which does not depend on $\mathbf{x}$. Supposing $\mathbf{f}(\cdot)$ a continuous and bounded function, the universal approximation theorem of Cybenko [2] guarantees that $\mathbf{f}(\cdot)$ can be approximated with an arbitrary precision using a suitable size single hidden layer MLP whose output can be written as $\mathbf{g}(\mathbf{x}, \mathbf{W}^*)$ where $\mathbf{W}^*$ is the optimal weight matrix. In other words, for all $\delta > 0$, there exists a single hidden layer MLP with weight matrix $\mathbf{W}^*$, so that

$$\|\mathbf{g}(\mathbf{x}, \mathbf{W}^*) - \mathbf{f}(\mathbf{x})\| < \delta. \tag{5}$$

Let $\mathbf{g}(\mathbf{x}, \mathbf{W}^*)$ be the output of a single hidden layer MLP containing R sigmoidal neurons and satisfying (5) for a desired, sufficiently small value of δ which allows replacing $\mathbf{f}(\mathbf{x})$ by $\mathbf{g}(\mathbf{x}, \mathbf{W}^*)$ in (4). In this case, (4) may be rewritten as

$$(\mathbf{x}_i, \mathbf{y}_i) \simeq (\mathbf{x}_i, \mathbf{g}(\mathbf{x}_i, \mathbf{W}^*) + \boldsymbol{\xi}_i), \quad i = 1, ..., N. \tag{6}$$

Suppose $\mathbf{g}(\mathbf{x}, \mathbf{W})$ represents the output of the class of single hidden layer MLP containing R sigmoidal neurons; denoting $\epsilon(\mathbf{W}) = \mathbf{y} - \mathbf{g}(\mathbf{x}, \mathbf{W})$, we would like to optimize $\mathbf{W}$ to obtain the best approximator for $\mathbf{W}^*$. When the noise samples, $\boldsymbol{\xi}_i$, are independent and identically distributed (i.i.d.), ordinary least square (OLS) estimation is usually used and gives a consistent estimator. It consists in minimizing $E_{\mathrm{OLS}} = \varepsilon(\mathbf{W})^T \varepsilon(\mathbf{W})$ where ε is the vector containing all the samples of all the

Manuscript received April 6, 2000; revised January 17, 2001 and August 29, 2001.

The authors are with the Laboratoire des Images et des Signaux (UMR CNRS 5083, INPG, UJF), 38031 Grenoble, France (e-mail: hosseini@lis.inpg.fr; chris@inpg.fr).

Publisher Item Identifier S 1045-9227(02)00357-0.

[1]Other error structures like multiplicative errors are sometimes considered [1].

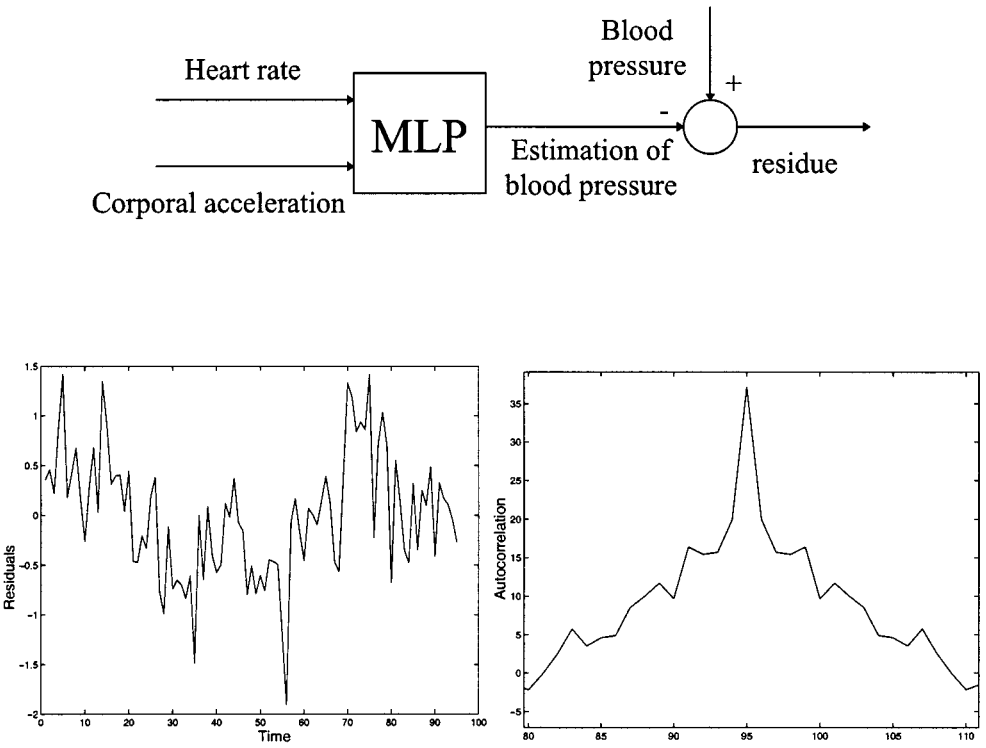


Fig. 1. Blood pressure modeling: Estimation residual and its autocorrelation function.

residual outputs. However, in many practical situations, the noise samples may be correlated and/or non stationary. Such a case is shown in Fig. 1, in which one would like to model the blood pressure of a patient as a function of his corporal acceleration and his heart rate. The input and output samples are collected during 48 h with a sampling period $T_s = 15$ min. The OLS method is used to train the different single hidden layer MLP corresponding to different number of neurons in the hidden layer. This experiment shows that all the models, including the best of them according a performance estimate,[2] provide a temporally correlated residual (Fig. 1). One may conclude that this correlation is due to the unknown inputs influence (feeding, stress,...) and cannot be removed using only two known inputs. According to our definition of noise in (4), we can so conclude the existence of a temporally correlated noise in the model. The question is: "may this correlation be used to improve the neural-network estimation?" Although this problem has not been intensively studied in the neural-network literature, statisticians, and automatic control researchers have considered it more.

It is well known in linear dynamic system identification that the presence of the colored noise leads usually to a biased parameter estimation [3]. Many approaches have been proposed to obtain unbiased estimation which depends essentially on the noise structure. The main strategy consists in identifying the noise parameters simultaneously with the system parameters in order to obtain a whitened residual [3].

In statistics, the presence of colored noise has been largely studied. When the covariance matrix of the noise is known, the well-known generalized least square (GLS) method which consists in minimizing $E_{\mathrm{GLS}} = \varepsilon(\mathbf{W})^T \mathbf{V}^{-1} \varepsilon(\mathbf{W})$, $\mathbf{V}$ being the noise covariance matrix, provides a better parameter estimation [1]. As $\mathbf{V}$ is basically unknown, we have to estimate it. A possible strategy can be the following [4]: a first OLS estimation, computation of the residual, estimation of $\mathbf{V}$ using the residual, computation of $\mathbf{V}^{-1}$ and final GLS estimation of $\mathbf{W}$. Unfortunately, this strategy cannot be used in practice. At first, unless we have multiple measurements $\mathbf{y}_i$ for the same input $\mathbf{x}_i$, the estimation of $\mathbf{V}$ will be impossible because an $(M \times N) \times (M \times N)$ symmetric matrix, with $((M \times N)^2 + (M \times N))/2$ independent parameters, cannot be identified using $M \times N$ data samples. Moreover, the approximation error of $\mathbf{V}$ implies a larger error in the computation of its inverse. Finally, for large training data bases, the inverse computation is expensive and the optimization algorithm will be very time-consuming, unless the majority of the entries of $\mathbf{V}^{-1}$ are zero. For these reasons, in practice, we have to choose a pertinent and simple parametric noise model with few parameters so that the estimation of its parameters is very simple. Moreover, as the optimal network size for a given problem cannot generally be determined *a priori*, the above two-stage estimator is not desirable and it is better to estimate the noise parameters simultaneously with the system parameters, for example using a maximum likelihood approach. As we will see, this method is especially very efficient when a constructive procedure is used to build the network. In the literature of linear time series modeling, the case of temporal correlated noise is largely studied and two-stage and maximum likelihood estimators are derived [5], [6]. In the field of parametric nonlinear regression, there are also a few similar works for single output systems where the output is corrupted by temporally correlated noise [7]–[9]. A review of these works can be found in [1, Ch. 6].

[2]A test database is used to estimate the performance.

Another frequent practical case [10], [11] is the case of spatially correlated noise. This situation can be occurred when a common noise source influences the different outputs of a system.[3] The noises in different outputs can so be considered as the components of a multidimensional noise, defined by its covariance matrix (for a Gaussian noise). The knowledge on this matrix may be used to improve the system modeling.

Motivated by these works, we find interesting to study the more general case of spatio-temporal noise in a multioutput system, modeled by a multilayer perceptron, in order to reduce the variance of the estimator, and so, to improve the network generalization capacity. Another motivation to address this problem is to find a stopping criterion for constructive neural-network algorithms. In general, the optimal size of an MLP for a given problem is unknown. A too small network cannot learn the data with a sufficient accuracy (the corresponding estimator is biased), and a too large one leads to overfitting (the corresponding estimator has a large variance), and thus, to poor generalization. Constructive approaches have been used to solve this problem [12]–[15]. In these methods, one starts with a simple network and adds progressively new units until the network can fit the target function. The main problem is to define a relevant stopping criterion which avoids an uncontrolled network increase. One of the solutions proposed recently is to use the residual signal properties [16]: to avoid overfitting, the network growing procedure is stopped when the residual can be considered as noise. If the system noise is i.i.d., a simple correlation test on the residual can determine if it can be considered as i.i.d. noise or not. Evidently, if the noise is not i.i.d., this criterion fails. As we will see, in this case, prior knowledge on the noise or on the estimation of its characteristics can be used for "whitening" the residual so that the correlation test is still usable.

This paper is organized as follows. In Section II, after the problem statement for a spatio-temporal noise model, the general maximum likelihood (ML) solution is proposed and two special cases (first- and second-order autoregressive noises) are studied with more details. Section III describes the simplified case of single output network. In Section IV, a stopping criterion for constructive neural networks by whitening the estimation residual is proposed. Section V describes the experiments. A summary and discussion are given in Section VI.

II. Spatio-Temporal Correlated Noise in Multi-Output Neural Networks

As mentioned in the previous section, when the noise covariance matrix is unknown, we have to choose a pertinent parametric noise model and try to identify the parameters. In the following, two kinds of noise correlation are investigated: autocorrelation between the successive noise samples of each output, and cross-correlations between the different output noises. In a time-series estimation problem, the first kind may be due to a filtering effect (temporal correlation) and the second one may be the effect of a common noise source on the different outputs (spatial correlation).

[3]For example, an audio noise influencing many microphones.

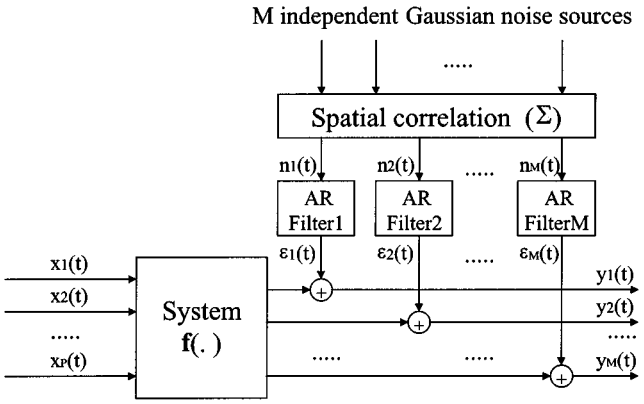


Fig. 2. Signal generating model.

The noise model which we used is a joint Gaussian M-dimensional stationary white noise, filtered by M qth-order autoregressive filters (Fig. 2). The choice of autoregressive model simplifies considerably the computations and is not so restricting because each stationary ARMA system can be modeled using an infinite-order AR model whose high-order terms may be neglected.

A. Statement of the Problem

Suppose $\boldsymbol{\xi}_i$ in (4) are the samples of the mentioned spatio-temporal correlated Gaussian noise and suppose that $\mathbf{g}(\mathbf{x}, \mathbf{W}^*)$ is the output vector of a single hidden layer MLP providing a sufficiently accurate approximation of $\mathbf{f}(\mathbf{x})$ so that the difference $\mathbf{g}(\mathbf{x}, \mathbf{W}^*) - \mathbf{f}(\mathbf{x})$ is negligible in comparison with the noise. In the following, we neglect this theoretical imprecision and suppose that $\mathbf{f}(\mathbf{x})$ can be exactly represented by $\mathbf{g}(\mathbf{x}, \mathbf{W}^*)$.

Considering this remark, suppose we have a P input, M output system which can be represented using the following equations:

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{W}^*) + \boldsymbol{\xi}(t) \tag{7}$$

$$\boldsymbol{\xi}(t) = \sum_{i=1}^{q} \boldsymbol{\Phi}_i \boldsymbol{\xi}(t-i) + \mathbf{n}(t) \tag{8}$$

$$\mathbf{n}(t) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}) \tag{9}$$

where $\mathbf{y}(t) = [y_1(t), \ldots, y_M(t)]^T$ are the system outputs, $\mathbf{x}(t) = [x_1(t), \ldots, x_P(t)]^T$ are the system inputs, $\mathbf{g}(\mathbf{x}(t), \mathbf{W}^*) = [g_1(\mathbf{x}(t), \mathbf{W}^*), \ldots, g_M(\mathbf{x}(t), \mathbf{W}^*)]^T$ are the neural-network outputs (approximations), $\xi(t) = [\xi_1(t), \ldots, \xi_M(t)]^T$ are qth-order autoregressive stationary colored noises and $\mathbf{n}(t) = [n_1(t), \ldots, n_M(t)]^T$ are jointly Gaussian white noises with $E[n_i(t)n_i(t-j)] = \sigma_i^2$ if $j = 0$; and zero otherwise, and $E[n_i(t)n_j(t)] = \sigma_{ij}$. $\boldsymbol{\Sigma}$ is the covariance matrix of $\mathbf{n}(t)$, $\boldsymbol{\Phi}_i$ are the diagonal matrices, modeling the qth-order autoregressive structure of $\boldsymbol{\xi}(t)$ and $\mathbf{W}^*$ is the network optimal weights matrix that must be estimated. Denoting jth diagonal element of matrix $\boldsymbol{\Phi}_i$ by $\phi_{i_{jj}}$, note that if $n_j(t)$ is stationary, $\xi_j(t)$ is stationary and causal if and only if [17]

$$1 - \sum_{i=1}^{q} \phi_{i_{jj}} z^i \neq 0 \quad \forall |z| \leq 1. \tag{10}$$

Assuming that N samples of the system inputs and outputs are collected at regular time intervals, we want to find the maximum likelihood estimator for $\mathbf{W}^*, \mathbf{\Phi} = [\mathbf{\Phi}_1, \ldots, \mathbf{\Phi}_q]$ and $\mathbf{\Sigma}$.

The problem can be treated in two different manners. In the first approach, one can use M separate, independent multi-input–single-output (MISO) networks for estimating M outputs. In this case, since the matrices $\mathbf{\Phi}_i$ are diagonal, each output only depends on one component of the noise $\mathbf{n}(t)$

$$y_k(t) = g_k(\mathbf{x}(t), \mathbf{W}^*) + \sum_{i=1}^{q}[\phi_{i_{kk}}(y_k(t-i) - g_k(\mathbf{x}(t-i), \mathbf{W}^*))] + n_k(t). \quad (11)$$

Thus, the spatial correlation of the noise has no influence on the estimation of the network weights, so that only the diagonal elements of the covariance matrix $\mathbf{\Sigma}$ have to be estimated. If this approach is chosen, the reader may go directly to the Section III of this paper, where the special case of temporally correlated noise in the MISO networks is treated.

A second approach, however, consists in using a sole multi-input–multi-output (MIMO) network for estimating the M outputs. In this case, one has to consider the spatial correlation between the noise components and the estimation of the off-diagonal elements of $\mathbf{\Sigma}$ becomes indispensable. Note that a MIMO network can generally be modeled with less parameters in comparison with M separate MISO networks because some parameters (early layer weights for an MLP) can participate in the estimation of many outputs. The reduction in the number of parameters can be crucial if the outputs can be implemented by sums of the same hidden neurons. On the other hand, the training of a MIMO network is generally more difficult because more parameters must be updated simultaneously. Consequently, the convergence problems are more probable to happen with the MIMO networks. Moreover, the use of a MIMO network makes the analysis of the complexity of the problem difficult: some outputs may require many hidden neurons, whereas others may be quasilinear functions of the inputs, but this will be concealed in the weights of the MIMO network.

The choice between the two approaches depends on the treated problem. Anyway, as a theoretical point of view, the study of the problem in its complete form (the second approach) is important. Thus, in the remaining of this section, the second approach is considered and the off-diagonal elements of the matrix $\mathbf{\Sigma}$ will be estimated. As $\mathbf{\Phi}_i$ are $M \times M$ diagonal matrices and $\mathbf{\Sigma}$ is a $M \times M$ symmetric matrix, there are $\dim(\mathbf{W}^*) + qM + (M^2 + M)/2$ independent parameters to estimate. Let us write the likelihood function, decomposed in two parts

$$\begin{aligned}\mathcal{F}(\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(N); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma}) &= \mathcal{F}(\mathbf{y}(1) \ldots \mathbf{y}(q); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma}) \\ &\cdot \mathcal{F}(\mathbf{y}(q+1) \ldots \mathbf{y}(N) \,|\, \mathbf{y}(1), \ldots, \mathbf{y}(q); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma}). \quad (12)\end{aligned}$$

The second part can be easily determined using

$$\begin{aligned}&\mathcal{F}(\mathbf{y}(q+1) \ldots \mathbf{y}(N) \,|\, \mathbf{y}(1), \ldots, \mathbf{y}(q); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma}) \\ &= \prod_{i=q+1}^{N} \mathcal{F}(\mathbf{y}(i) \,|\, \mathbf{y}(i-1), \ldots, \mathbf{y}(i-q)); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma}). \quad (13)\end{aligned}$$

Conditioning on a random variable means treating it as a deterministic constant. So, using (7)–(9)

$$\begin{aligned}&(\boldsymbol{\xi}(i) \,|\, \boldsymbol{\xi}(i-1), \ldots, \boldsymbol{\xi}(i-q)) \\ &\quad \sim \mathcal{N}\left(\sum_{j=1}^{q} \mathbf{\Phi}_j \xi(i-j), \mathbf{\Sigma}\right) \quad (14) \\ &(\mathbf{y}(i) \,|\, \mathbf{y}(i-1), \ldots, \mathbf{y}(i-q)) \\ &\quad \sim \mathcal{N}\left(\mathbf{g}(\mathbf{x}(i), \mathbf{W}^*) + \sum_{j=1}^{q} \mathbf{\Phi}_j \boldsymbol{\xi}(i-j), \mathbf{\Sigma}\right). \quad (15)\end{aligned}$$

Denoting the estimation residual by

$$\epsilon(i, \mathbf{W}) = \mathbf{y}(i) - \mathbf{g}(\mathbf{x}(i), \mathbf{W}) \quad (16)$$

where $\mathbf{g}(\mathbf{x}(i), \mathbf{W})$ is the network output for the ith observation and $\mathbf{W}$ is the actual network weights matrix, we obtain

$$\begin{aligned}&\mathcal{F}(\mathbf{y}(i) \,|\, \mathbf{y}(i-1), \ldots, \mathbf{y}(i-q); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma}) \\ &= \frac{1}{(2\pi)^{M/2}|\Sigma|^{1/2}} \exp\left(\frac{-1}{2}\left[\epsilon(i, \mathbf{W}) - \sum_{j=1}^{q} \mathbf{\Phi}_j \epsilon(i-j, \mathbf{W})\right]^T \right. \\ &\left. \times \mathbf{\Sigma}^{-1}\left[\epsilon(i, \mathbf{W}) - \sum_{j=1}^{q} \mathbf{\Phi}_j \epsilon(i-j, \mathbf{W})\right]\right). \quad (17)\end{aligned}$$

Thus, (13) and (17) give the second right-side term of (12). The first term is more difficult to obtain. The solution consists in decomposing it in conditional likelihood terms and in determining each term separately

$$\begin{aligned}&\mathcal{F}(\mathbf{y}(1), \mathbf{y}(2), \ldots \mathbf{y}(q); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma}) \\ &\quad = \mathcal{F}(\mathbf{y}(1); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma}) \cdot \mathcal{F}(\mathbf{y}(2) \,|\, \mathbf{y}(1); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma}) \ldots \\ &\qquad \cdot \mathcal{F}(\mathbf{y}(q) \,|\, \mathbf{y}(1), \ldots, \mathbf{y}(q-1); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma}). \quad (18)\end{aligned}$$

In the following, we study the two special cases of first- and second-order AR noise: $q = 1$ and $q = 2$. Higher order structures could be identified similarly.

B. AR1 Model

In this case, (18) reduces to $\mathcal{F}(\mathbf{y}(1); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma})$. Using (8) and noting that $\epsilon(t-1)$ is independent of $\mathbf{n}(t)$, we can write

$$E[\epsilon(t)\epsilon(t)^T] = \mathbf{\Phi}_1 E[\epsilon(t-1)\epsilon(t-1)^T]\mathbf{\Phi}_1^T + E[\mathbf{n}(t)\mathbf{n}(t)^T]. \quad (19)$$

Denoting $E[\epsilon(t)\epsilon(t)^T]$ by $\mathbf{\Lambda}$ and using the stationary properties of $\epsilon(t)$, (19) reduces to discrete-time Lyapunov equation [18][4]

$$\mathbf{\Lambda} = \mathbf{\Phi}_1 \mathbf{\Lambda} \mathbf{\Phi}_1^T + \mathbf{\Sigma}. \quad (20)$$

Using matrix to vector operator $\mathrm{vec}(\cdot)$ and the Kronecker product defined, respectively, by

$$\begin{aligned}&\mathrm{vec}(\mathbf{X}) \\ &= [x_{11}, x_{21}, \ldots, x_{m1}, x_{12}, \ldots, x_{m2}, \ldots, x_{1m}, \ldots, x_{mm}]^T \\ &\text{and} \\ &\mathbf{A} \otimes \mathbf{B}\end{aligned}$$

[4]For an AR1 model, the solution (23) may follow directly from (20). However, the notations being used in the next section, we keep the details of the proof.

$$= \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{n1}\mathbf{B} & \cdots & a_{nn}\mathbf{B} \end{bmatrix}$$

where m and n represent, respectively, the size of $\mathbf{X}$ and $\mathbf{A}$, and using the following property: $\mathrm{vec}(\mathbf{AXB}) = (\mathbf{B}^T \otimes \mathbf{A})\mathrm{vec}(\mathbf{X})$, we can obtain the vector equivalent of (20)

$$\mathrm{vec}(\mathbf{\Lambda}) = (\mathbf{\Phi}_1 \otimes \mathbf{\Phi}_1)\mathrm{vec}(\mathbf{\Lambda}) + \mathrm{vec}(\mathbf{\Sigma}) \tag{21}$$

which can be solved by

$$\mathrm{vec}(\mathbf{\Lambda}) = (\mathbf{I} - \mathbf{\Phi}_1 \otimes \mathbf{\Phi}_1)^{-1}\mathrm{vec}(\mathbf{\Sigma}). \tag{22}$$

But, since $\mathbf{\Phi}_1$ is a diagonal matrix with diagonal entries ϕ_{1ii} and $\mathbf{\Lambda}$ and $\mathbf{\Sigma}$ are the symmetric matrices with entries λ_{ij} and σ_{ij} respectively,[5] it can be easily shown that

$$\lambda_{ij} = \frac{\sigma_{ij}}{1 - \phi_{1ii}\phi_{1jj}}. \tag{23}$$

Then, using

$$\mathbf{y}(1) \sim \mathcal{N}(\mathbf{g}(\mathbf{x}(1), \mathbf{W}), E[\epsilon(1, \mathbf{W})\epsilon(1, \mathbf{W})^T]) \tag{24}$$

$\mathcal{F}(\mathbf{y}(1); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma})$ can be computed considering that $\epsilon(1, \mathbf{W}) = \mathbf{y}(1) - \mathbf{g}(\mathbf{x}(1), \mathbf{W})$

$$\mathcal{F}(\mathbf{y}(1); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{M/2}|\mathbf{\Lambda}|^{1/2}} \times \exp\left(-\frac{1}{2}\epsilon(1, \mathbf{W})^T \mathbf{\Lambda}^{-1} \epsilon(1, \mathbf{W})\right). \tag{25}$$

Taking the logarithm of (12), and using (13), (17), and (25), the log-likelihood function can be computed

$$\begin{aligned} \mathcal{L}_Y(y) = \mathrm{const} &- \frac{1}{2}\log|\mathbf{\Lambda}| \\ &- \frac{1}{2}\epsilon(1, \mathbf{W})^T \mathbf{\Lambda}^{-1} \epsilon(1, \mathbf{W}) \\ &- \frac{N-1}{2}\log|\mathbf{\Sigma}| - \frac{1}{2}\sum_{i=2}^{N}[\epsilon(i, \mathbf{W}) \\ &- \mathbf{\Phi}_1 \epsilon(i-1, \mathbf{W})]^T \mathbf{\Sigma}^{-1}[\epsilon(i, \mathbf{W}) \\ &- \mathbf{\Phi}_1 \epsilon(i-1, \mathbf{W})]. \end{aligned} \tag{26}$$

After eliminating the constants, the maximum likelihood estimation of the unknown matrices $\mathbf{W}, \mathbf{\Phi}$ and $\mathbf{\Sigma}$ may be obtained by minimizing the following cost function:

$$\begin{aligned} \mathcal{J}_1 = &\log|\mathbf{\Lambda}| + \epsilon(1, \mathbf{W})^T \mathbf{\Lambda}^{-1} \\ &\times \epsilon(1, \mathbf{W}) + (N-1)\log|\mathbf{\Sigma}| \\ &+ \sum_{i=2}^{N}[\epsilon(i, \mathbf{W}) - \mathbf{\Phi}_1 \epsilon(i-1, \mathbf{W})]^T \\ &\times \mathbf{\Sigma}^{-1}[\epsilon(i, \mathbf{W}) - \mathbf{\Phi}_1 \epsilon(i-1, \mathbf{W})]. \end{aligned} \tag{27}$$

Notice that the number of independent parameters of (27) is $\dim(\mathbf{W}) + (3M + M^2)/2$.

[5] $\sigma_{ii} = \sigma_i^2$.

C. AR2 Model

Denoting $\mathbf{\Lambda}_0 = E[\epsilon(t)\epsilon(t)^T], \mathbf{\Lambda}_1 = E[\epsilon(t)\epsilon(t-1)^T]$ and $\mathbf{\Lambda}_2 = E[\epsilon(t)\epsilon(t-2)^T]$, we can compute (18). Using the previous section results, one can write

$$\mathcal{F}(\mathbf{y}(1); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{M/2}|\mathbf{\Lambda}_0|^{1/2}} \times \exp\left(-\frac{1}{2}\epsilon(1, \mathbf{W})^T \mathbf{\Lambda}_0^{-1} \epsilon(1, \mathbf{W})\right). \tag{28}$$

The computation of $\mathcal{F}(\mathbf{y}(2) \,|\, \mathbf{y}(1); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma})$ is more tricky. To compute this function, we can compute $\mathcal{F}(\epsilon(2, \mathbf{W}) \,|\, \epsilon(1, \mathbf{W}); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma})$ and replace $\epsilon(1, \mathbf{W})$ and $\epsilon(2, \mathbf{W})$ by $\mathbf{y}(1) - \mathbf{g}(\mathbf{x}(1), \mathbf{W})$ and $\mathbf{y}(2) - \mathbf{g}(\mathbf{x}(2), \mathbf{W})$, respectively. For this purpose, we must determine the density function of the part of $\epsilon(2, \mathbf{W})$ which is not predictable from $\epsilon(1, \mathbf{W})$

$$\begin{aligned} &\mathcal{F}(\epsilon(2, \mathbf{W}) \,|\, \epsilon(1, \mathbf{W}); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma}) \\ &\qquad = \mathcal{F}(\epsilon(2, \mathbf{W}) - \mathbf{\Gamma}\epsilon(1, \mathbf{W}); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma}) \end{aligned} \tag{29}$$

where $E[(\epsilon(2, \mathbf{W}) - \mathbf{\Gamma}\epsilon(1, \mathbf{W}))\epsilon(1, \mathbf{W})^T] = \mathbf{0}$, i.e.

$$\begin{aligned} \mathbf{\Gamma} &= E[\epsilon(2, \mathbf{W})\epsilon(1, \mathbf{W})^T][E[\epsilon(1, \mathbf{W})\epsilon(1, \mathbf{W})^T]]^{-1} \\ &= \mathbf{\Lambda}_1 \mathbf{\Lambda}_0^{-1}. \end{aligned} \tag{30}$$

Clearly, $\eta(\mathbf{W}) = \epsilon(2, \mathbf{W}) - \mathbf{\Gamma}\epsilon(1, \mathbf{W})$ is a Gaussian random vector: $\eta \sim \mathcal{N}(\mathbf{0}, E[\eta(\mathbf{W})\eta(\mathbf{W})^T])$. Its covariance matrix can be computed as follows:

$$\begin{aligned} E[\eta(\mathbf{W})\eta(\mathbf{W})^T] &= E[(\epsilon(2, \mathbf{W}) - \mathbf{\Gamma}\epsilon(1, \mathbf{W})) \\ &\qquad \times (\epsilon(2, \mathbf{W})^T - \epsilon(1, \mathbf{W})^T \mathbf{\Gamma}^T)] \\ &= \mathbf{\Lambda}_0 - \mathbf{\Lambda}_1 \mathbf{\Gamma}^T - \mathbf{\Gamma}\mathbf{\Lambda}_1^T + \mathbf{\Gamma}\mathbf{\Lambda}_0\mathbf{\Gamma}^T. \end{aligned}$$

Using (30) and considering that $\mathbf{\Lambda}_0$ is symmetric, a simplified equation can be obtained

$$E[\eta(\mathbf{W})\eta(\mathbf{W})^T] = \mathbf{\Lambda}_0 - \mathbf{\Lambda}_1 \mathbf{\Lambda}_0^{-1} \mathbf{\Lambda}_1^T. \tag{31}$$

Finally, using (30) and (31)

$$\begin{aligned} &\mathcal{F}(\mathbf{y}(2) \,|\, \mathbf{y}(1); \mathbf{W}, \mathbf{\Phi}, \mathbf{\Sigma}) \\ &\quad = \frac{1}{(2\pi)^{M/2}|\mathbf{\Lambda}_0 - \mathbf{\Lambda}_1 \mathbf{\Lambda}_0^{-1} \mathbf{\Lambda}_1^T|^{1/2}} \\ &\qquad \times \exp\Big(-\frac{1}{2}\left(\epsilon(2, \mathbf{W}) - \mathbf{\Lambda}_1 \mathbf{\Lambda}_0^{-1} \epsilon(1, \mathbf{W})\right)^T \\ &\qquad \times \left(\mathbf{\Lambda}_0 - \mathbf{\Lambda}_1 \mathbf{\Lambda}_0^{-1} \mathbf{\Lambda}_1^T\right)^{-1} \left(\epsilon(2, \mathbf{W}) \right. \\ &\qquad \left. - \mathbf{\Lambda}_1 \mathbf{\Lambda}_0^{-1} \epsilon(1, \mathbf{W})\right)\Big). \end{aligned} \tag{32}$$

Once more, using the matrix to vector conversion and the Kronecker product, $\mathbf{\Lambda}_0$ and $\mathbf{\Lambda}_1$ satisfy (see Appendix)

$$\mathrm{vec}(\mathbf{\Lambda}_0) = \mathbf{D}^{-1}\mathrm{vec}(\mathbf{\Sigma}) \tag{33}$$

$$\mathrm{vec}(\mathbf{\Lambda}_1) = \mathbf{C}\mathrm{vec}(\mathbf{\Lambda}_0) \tag{34}$$

where

$$\mathbf{C} = (\mathbf{I} - \mathbf{\Phi}_2 \otimes \mathbf{\Phi}_2)^{-1}(\mathbf{I} \otimes \mathbf{\Phi}_1 + \mathbf{\Phi}_1 \otimes \mathbf{\Phi}_2) \tag{35}$$

$$\begin{aligned} \mathbf{D} = &\mathbf{I} - \mathbf{\Phi}_1 \otimes \mathbf{\Phi}_1 - \mathbf{\Phi}_1^2 \otimes \mathbf{\Phi}_2 - \mathbf{\Phi}_2 \otimes \mathbf{\Phi}_2 \\ &- (\mathbf{\Phi}_2 \otimes \mathbf{\Phi}_1 + \mathbf{\Phi}_2\mathbf{\Phi}_1 \otimes \mathbf{\Phi}_2)\mathbf{C}. \end{aligned} \tag{36}$$

Using (13), (17), (28), and (32), taking the logarithm of (12) and eliminating the constants, the maximum likelihood estima-

tion of the unknown matrices $\mathbf{W}$, $\mathbf{\Phi}$, and $\mathbf{\Sigma}$ is obtained by minimizing the following objective function:

$$\begin{aligned}\mathcal{J}_2 &= \log|\mathbf{\Lambda}_0| + \epsilon(1,\mathbf{W})^T\mathbf{\Lambda}_0^{-1}\epsilon(1,\mathbf{W}) \\ &\quad + \log\left|\mathbf{\Lambda}_0 - \mathbf{\Lambda}_1\mathbf{\Lambda}_0^{-1}\mathbf{\Lambda}_1^T\right| + \big(\epsilon(2,\mathbf{W}) \\ &\quad - \mathbf{\Lambda}_1\mathbf{\Lambda}_0^{-1}\epsilon(1,\mathbf{W})\big)^T\left(\mathbf{\Lambda}_0 - \mathbf{\Lambda}_1\mathbf{\Lambda}_0^{-1}\mathbf{\Lambda}_1^T\right)^{-1}\big(\epsilon(2,\mathbf{W}) \\ &\quad - \mathbf{\Lambda}_1\mathbf{\Lambda}_0^{-1}\epsilon(1,\mathbf{W})\big) + (N-2)\log|\mathbf{\Sigma}| \\ &\quad + \sum_{i=3}^{N}(\epsilon(i,\mathbf{W}) - \mathbf{\Phi}_1\epsilon(i-1,\mathbf{W}) \\ &\quad - \mathbf{\Phi}_2\epsilon(i-2,\mathbf{W}))^T\mathbf{\Sigma}^{-1}(\epsilon(i,\mathbf{W}) \\ &\quad - \mathbf{\Phi}_1\epsilon(i-1,\mathbf{W}) - \mathbf{\Phi}_2\epsilon(i-2,\mathbf{W})) \end{aligned} \tag{37}$$

in which there are $\dim(\mathbf{W}) + (5M + M^2)/2$ parameters to estimate.

III. A Simplified Case: Single Output Networks

The case of single output systems is interesting because simpler cost functions can be extracted for an arbitrary-order AR noise model. In this case, given a data set:

$$(\mathbf{x}_i, y_i) = (\mathbf{x}_i, f(\mathbf{x}_i) + \xi_i), \quad i = 1,\ldots,N \tag{38}$$

where $\mathbf{x}_i$ are the samples of the input variables $\mathbf{x} \in \mathcal{R}^p$, collected at regular time intervals: $\mathbf{x}_i = \mathbf{x}(i\tau)$, y_i are samples of the output variable $y \in \mathcal{R}$ and ξ_i are samples of a stationary correlated noise, resulted from filtering an i.i.d. Gaussian noise $n \sim \mathcal{N}(0,\sigma_n^2)$ by a qth-order autoregressive filter

$$\xi_i = \sum_{j=1}^{q}\phi_j\xi_{i-j} + n_i \tag{39}$$

we want to find a good approximation of the underlying relationship $f(\cdot)$. Using the results of previous section, (17) reduces to

$$\begin{aligned}&\mathcal{F}(y_i \,|\, y_{i-1},\ldots,y_{i-q};\mathbf{W},\phi_1,\ldots,\phi_q,\sigma_n) \\ &= \frac{1}{\sqrt{2\pi}\sigma_n}\exp\left(\frac{-1}{2\sigma_n^2}\left(\epsilon_i(\mathbf{W}) - \sum_{j=1}^{q}\phi_j\epsilon_{i-j}(\mathbf{W})\right)^2\right)\end{aligned} \tag{40}$$

where $\epsilon_i(\mathbf{W})$ represents the residual of the actual network. Thus, using (12) and (13) the complete likelihood function is written

$$\begin{aligned}&\mathcal{F}(y_1,y_2,\ldots,y_N;\mathbf{W},\phi_1,\ldots,\phi_q,\sigma_n) \\ &\quad = \mathcal{F}(y_1,\ldots,y_q;\mathbf{W},\phi_1,\ldots,\phi_q,\sigma_n)\prod_{i=q+1}^{N}\frac{1}{\sqrt{2\pi}\sigma_n} \\ &\qquad \times\exp\left(\frac{-1}{2\sigma_n^2}\left(\epsilon_i(\mathbf{W}) - \sum_{j=1}^{q}\phi_j\epsilon_{i-j}(\mathbf{W})\right)^2\right).\end{aligned} \tag{41}$$

Denoting $\sigma_n^2\mathbf{V}_q$, the covariance matrix of $\epsilon^q(\mathbf{W}) = [\epsilon_1(\mathbf{W}),\ldots,\epsilon_q(\mathbf{W})]^T$ i.e., the q first samples of $\epsilon(\mathbf{W}) = \mathbf{y} - \mathbf{g}(\mathbf{x},\mathbf{W})$, we can write

$$\begin{aligned}&\mathcal{F}(y_1,\ldots,y_q;\mathbf{W},\phi_1,\ldots,\phi_q,\sigma_n) \\ &\quad = \frac{|\mathbf{V}_q^{-1}|^{1/2}}{(2\pi\sigma_n^2)^{q/2}}\exp\left(\frac{-1}{2\sigma_n^2}\left(\epsilon^q(\mathbf{W})^T\mathbf{V}_q^{-1}\epsilon^q(\mathbf{W})\right)\right).\end{aligned} \tag{42}$$

Taking the logarithm of (41), using (42) and after eliminating the constant terms, maximizing the likelihood function reduces finally to minimizing the following function:

$$\begin{aligned}\mathcal{J} = N\log(\sigma_n^2) - \log\left|\mathbf{V}_q^{-1}\right| + \frac{1}{\sigma_n^2}\Bigg[&\epsilon^q(\mathbf{W})^T\mathbf{V}_q^{-1}\epsilon^q(\mathbf{W}) \\ &+ \sum_{i=q+1}^{N}\left(\epsilon_i(\mathbf{W}) - \sum_{j=1}^{q}\phi_j\epsilon_{i-j}(\mathbf{W})\right)^2\Bigg].\end{aligned} \tag{43}$$

Evaluation of this cost function requires the inversion of the $(q \times q)$ matrix $\mathbf{V}_q$. Denoting (i,j)th entries of $\mathbf{V}_q^{-1} = v_{q_{ij}}^{-1}$, it is shown that [19]

$$v_{q_{ij}}^{-1} = \sum_{k=0}^{i-1}\phi_k\phi_{k+j-i} - \sum_{k=q+1-j}^{q+i-j}\phi_k\phi_{k+j-i} \quad \text{for } 1 \le i \le j \le q \tag{44}$$

where $\phi_0 \equiv -1$. Values of $v_{q_{ij}}^{-1}$ for $i > j$ can be inferred from the fact that $v_{q_{ij}}^{-1}$ is symmetric. For the special case of AR1 process, it can be easily verified that $\mathbf{V}_q^{-1} = 1 - \phi_1^2$ which implies

$$\begin{aligned}\mathcal{J}_1 = N\log\left(\sigma_n^2\right) - \log\left(1-\phi_1^2\right) + \frac{1}{\sigma_n^2}\Bigg[&\left(1-\phi_1^2\right)\epsilon_1(\mathbf{W})^2 \\ &+ \sum_{i=2}^{N}(\epsilon_i(\mathbf{W}) - \phi_1\epsilon_{i-1}(\mathbf{W}))^2\Bigg].\end{aligned} \tag{45}$$

For an AR2 process, the (44) implies

$$\mathbf{V}_q^{-1} = \begin{pmatrix} 1-\phi_2^2 & -(\phi_1+\phi_1\phi_2) \\ -(\phi_1+\phi_1\phi_2) & 1-\phi_2^2 \end{pmatrix} \tag{46}$$

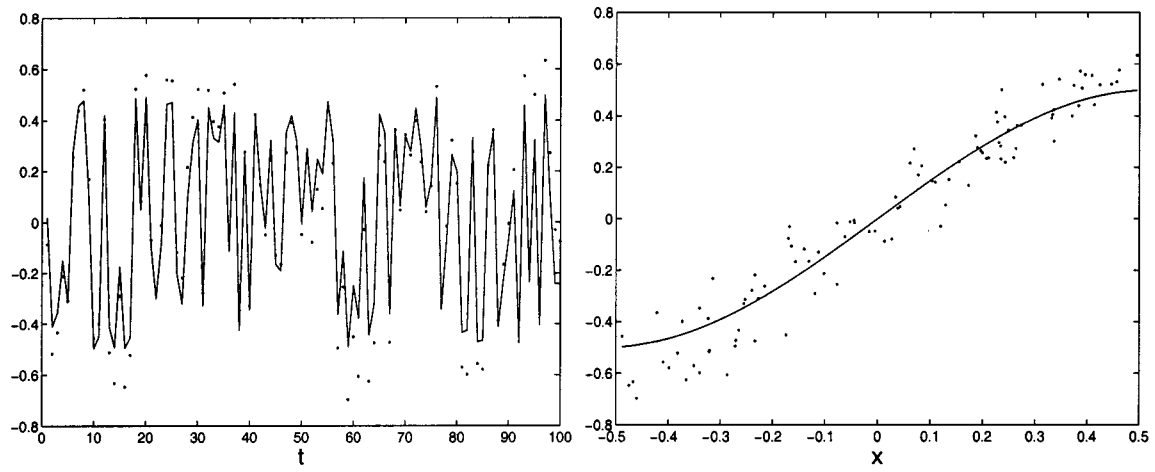
and

$$\begin{aligned}\mathcal{J}_2 &= N\log\left(\sigma_n^2\right) - \log\left[(1-\phi_2)^2 - \phi_1^2\right] \\ &\quad - 2\log(1+\phi_2) + \frac{1+\phi_2}{\sigma_n^2}[(1-\phi_2) \\ &\quad \times(\epsilon_1(\mathbf{W})^2 + \epsilon_2(\mathbf{W})^2) - 2\phi_1\epsilon_1(\mathbf{W})\epsilon_2(\mathbf{W})] \\ &\quad + \frac{1}{\sigma_n^2}\sum_{i=3}^{N}(\epsilon_i(\mathbf{W}) - \phi_1\epsilon_{i-1}(\mathbf{W}) - \phi_2\epsilon_{i-2}(\mathbf{W}))^2.\end{aligned} \tag{47}$$

It can be easily verified that (45) and (47) are the special versions of (27) and (37) for single output systems.

IV. A Stopping Criterion for Constructive Neural Networks

In general, the optimal size of an MLP for a given problem is unknown. A too small network cannot learn the data with a sufficient accuracy and a too large network leads to overfitting and thus to poor generalization. Many approaches have been proposed to determine dynamically the network size during training [20]–[23] and constructive algorithms are among the most promising [12]–[15], [24]. These algorithms grow dynamically the size of the hidden layers until a satisfying performance is achieved. The problem is to define a relevant stopping criterion which avoids uncontrolled network increase. This criterion may be based on a separate validation set [25], cross validation [26]



(a) (b)

Fig. 3. Target function (solid line) and its noisy samples (points) as functions of (a) time and (b) input signal x (a uniformly distributed random signal).

or bootstrapping [27]. It may also be obtained from a number of information criteria like Akaike's information criterion (AIC) [28], Bayesian information criterion (BIC) [29] or generalized prediction error (GPE) [30]. However, these methods either require additional data or are very time consuming.

Another solution proposed recently, inspired by the classical system identification tests, is to use the residual signal properties [16]. From (4) and (5), it is clear that an optimal neural-network approximation is achieved if $\mathbf{g}(\mathbf{x}, \mathbf{W}) = \mathbf{f}(\mathbf{x})$ i.e., if the error samples are $\mathbf{y}_i - \mathbf{g}(\mathbf{x}_i, \mathbf{W}) = \boldsymbol{\xi}_i$, i.e., if the residual can be considered as noise. If noise samples $\boldsymbol{\xi}_i$ are i.i.d., the stopping criterion may consist in comparing the residual with such noise. In other words, denoting $\epsilon_{i,j}$, the jth sample of the ith residual output, the criterion is nothing but a simple hypothesis test: $\forall k, l \,|\, kl \neq 0, E[\epsilon_{i,j}\epsilon_{i+k,j+l}] = 0$ versus $E[\epsilon_{i,j}\epsilon_{i+k,j+l}] \neq 0$. The correlation test determines if the residual can be considered as noise or not. If it is not the case, one concludes that the residual still includes a part of the signal and the construction procedure must be continued. Evidently, if the noise is not i.i.d., this criterion fails and the algorithm may uncontrollably progress toward overfitting. In this case, prior knowledge on the noise or the estimate of its characteristics can be used for "whitening" the residual. Thus, the mentioned hypothesis tests would be still usable.

Thus, the methods mentioned in the previous sections are also useful to stop the construction procedure. At each step of the network construction, using the maximum likelihood approach, the noise parameters are estimated simultaneously with the network weights. Then, these parameters are used to construct a whitening linear filter. The residual of the neural-network estimation is applied to this filter and a correlation hypothesis test is applied to verify if the result can be considered as white noise. If this is the case, the procedure is stopped; otherwise, it continues. The procedure is very similar to linear system identification in presence of colored noise where the noise inverse filter is identified during the system identification [3].

This technique, being very efficient in many cases, has unfortunately its own drawbacks. Suppose we have an undersized network whose residual contains the correlated noise plus a part of the signal. Evidently, the whitening filter, which can be considered as the inverse of the colored noise generating AR filter, is an MA high-pass filter. If the signal part of the residual is low frequency, it may be eliminated by the inverse filter so that the filter output is considered as white noise and a false stopping decision is made.

V. Experimental Results

In this section, we present our experimental results with three simulation examples and a real-world problem.

A. Temporally Correlated Noise in a Single Output System

In the first experiment, we study a simple, single output model where only temporal correlation of noise is considered. We would like to estimate the function

$$f(x) = 0.5\sin(3x), \quad x \in [-0.5, 05]$$

using a single hidden unit MLP, from N samples of $f(x)$, corrupted by a first-order AR noise

$$(x_i, y_i = f(x_i) + \xi_i), \quad i \in [1, N]$$
$$\xi_i = \phi\xi_{i-1} + n_i$$

where $n \sim \mathcal{N}(0, \sigma^2)$ is a Gaussian, i.i.d. noise and the subscript i indicates a sample collected at time $\tau i, \tau$ being the sampling period. The inputs x_i are the samples of a temporal i.i.d. random sequence, uniformly distributed on $[-0.5, 0.5]$. Fig. 3(a) presents the signals $f(x_i)$ and y_i as functions of time for $N = 100, \phi = 0.95$ and S/N[6] $= 18.4$ dB. The correlated structure of noise is quite clear in the figure. Fig. 3(b) illustrates the same signals as functions of input signal x_i, where the noise correlation is not appeared directly. Note that while the OLS method needs only Fig. 3(b) (i.e., the couples (x_i, y_i)) to make its estimation, Fig. 3(a) (the order of couples in time) is also necessary for an ML estimation. In the following, we compare the performance of four different estimators for different values of N, ϕ and S/N.

1) OLS estimator.

[6]Signal-to-noise ratio $= 10\log_{10}(E_S/E_N)$, where E_S represents the signal power and E_N is the power of white noise, n, before AR filtering.

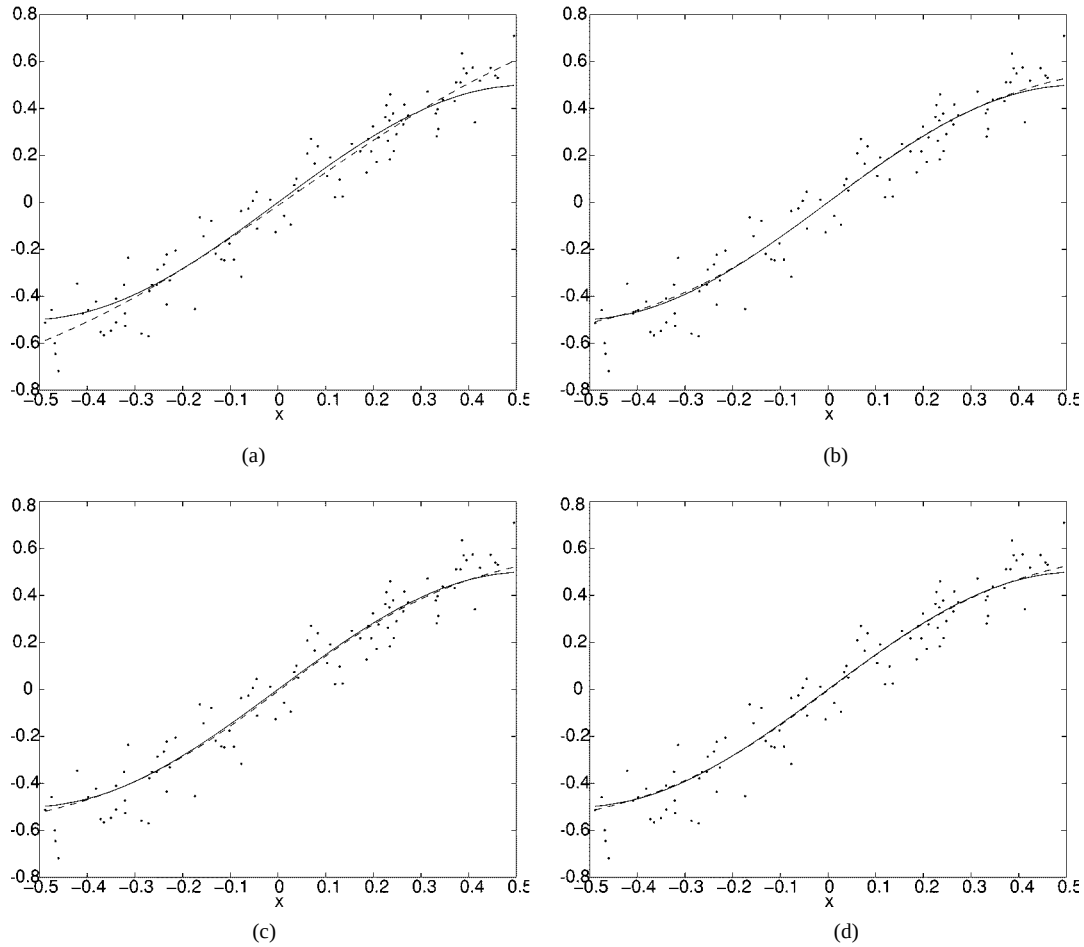


Fig. 4. (a) OLS estimation. (b) GLS Estimation. (c) Two-stages Estimation. (d) ML Estimation. In all the figures, the solid line represents the target function, the points its noisy samples, and the dashed line the estimation.

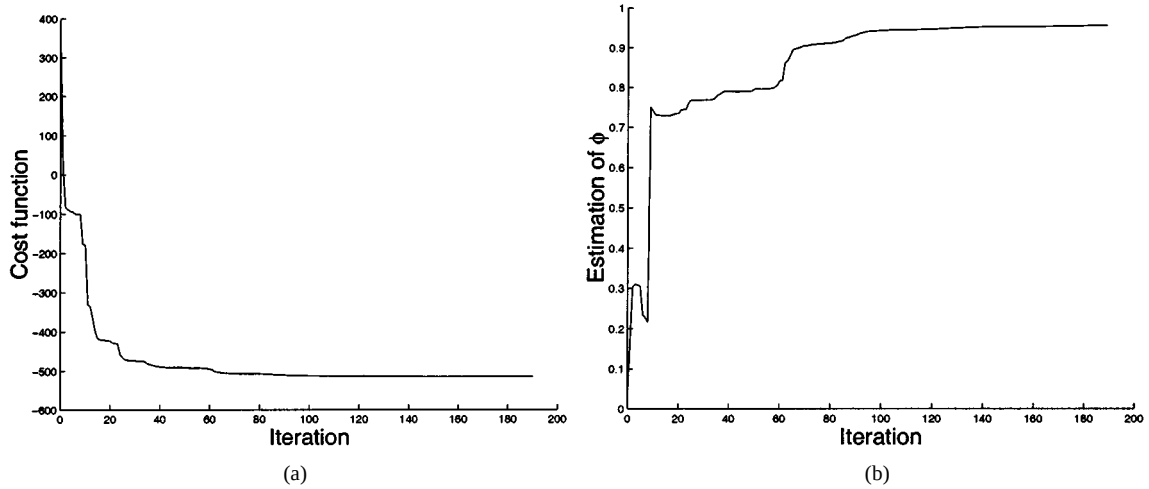


Fig. 5. Evolution of (a) the cost function and (b) the estimation of correlation coefficient of noise during ML training for example of Fig. 4.

2) GLS estimator. To realize this estimator, we use the true value of ϕ. It should then provide the best performance among four estimators.
3) Two-stage estimator. This estimator uses the value $\hat{\phi}$ estimated by OLS estimator for a second GLS estimation.
4) ML estimator. This estimator minimizes the cost function (45).

For each chosen triplet N, ϕ and $S/N, 10$ experiences corresponding to different noise seed values are done. The result for a sample run corresponding to $N = 100, \phi = 0.95$ and $S/N = 18.4$ dB is shown in Fig. 4 for each of four estimators. For the same example, the evolution of the cost function in ML algorithm and the evolution of the estimation of correlation coefficient, ϕ, during training is presented in Fig. 5. Fig. 6 presents

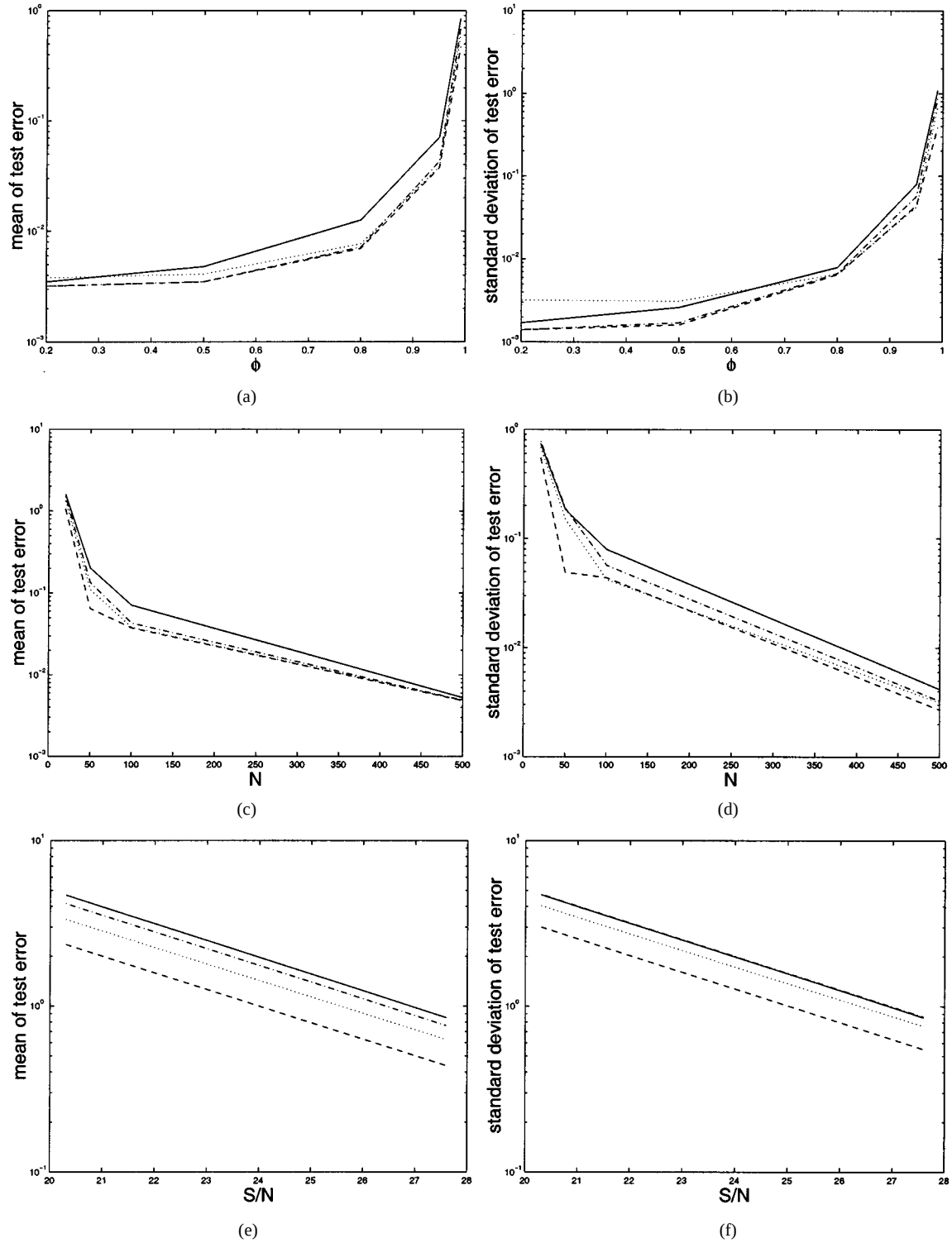


Fig. 6. Mean (at left) and standard deviation (at right) of the test error as functions of ϕ (for $N = 100$ and S/N $= 24.7$ dB), N (for $\phi = 0.95$ and S/N $= 24.7$ dB) and S/N (for $\phi = 0.99$ and $N = 50$) for the algorithms OLS (solid line), GLS (---), two-stage (-.-) and ML (. . .).

the mean and the standard deviation of test error[7] for each estimator as functions of ϕ, N and S/N.

According to the figures, it is clear that the estimators GLS and OLS are, respectively, the best and the worst of the four estimators in generalization. This result is not surprising since the GLS estimator knows exactly the correlation structure of noise and the OLS estimator pays no attention to it. Among the two others, estimator ML is the best for highly correlated noises (great values of ϕ) and the worst for the lowly correlated noises.

Moreover, Fig. 6 shows that when all the estimations are very good or very bad, the difference of their performance is not considerable. Indeed, in absence of correlation (small values of ϕ)

[7]In all the simulation examples of this section, we mean by test error, the sum of squared estimation errors with respect to the regression function, evaluated on a large number of inputs, different from those of the training set, in the domain limited by the training set.

or for the great training data bases, the noise structure has not much influence on the estimatation. Consequently, all the estimators manage to follow the regression function. On the other hand, when the correlation is too strong or the training data base is too small, the distinction between the signal structure and the noise structure is not easy and the estimation of MLP is so poor that the knowledge on the noise structure will not be of much use.

The four estimators are also compared on a more complicated problem, i.e., the estimation of the function:

$$f(x) = 0.7\sin(\pi x)\cos(2\pi x)$$

using an MLP with five hidden units from 100 samples corrupted by a first-order AR noise

$$(x_i, y_i = f(x_i) + \xi_i), \quad i \in [1, N = 100]$$
$$\xi_i = \phi\xi_{i-1} + n_i, \quad \phi = 0.95$$

where $n \sim \mathcal{N}(0, \sigma^2 = 0.0044)$ is a Gaussian i.i.d. noise and the inputs x_i are the samples of an i.i.d. sequence, uniformly distributed on $[0, 2]$. The result, for a sample run, is shown in Fig. 7 and confirms the previous conclusions.

B. Simulation With Spatio-Temporal Noise

We try to estimate two functions: $s_1 = 2\tanh(x+1) - 2\tanh(2x-1.5)$ and $s_2 = 3\tanh(2x-1) + 5\tanh(-3x+1)$ in the range $x \in]-5, 5]$, from 100 samples corrupted by a spatio-temporally correlated Gaussian noise

$$\begin{cases} y_1(i) = s_1(i) + \xi_1(i) \\ y_2(i) = s_2(i) + \xi_2(i) \end{cases} \quad \text{and} \quad x(i) = 0.1i - 5$$
$$\text{for } i \in [1, 100]$$

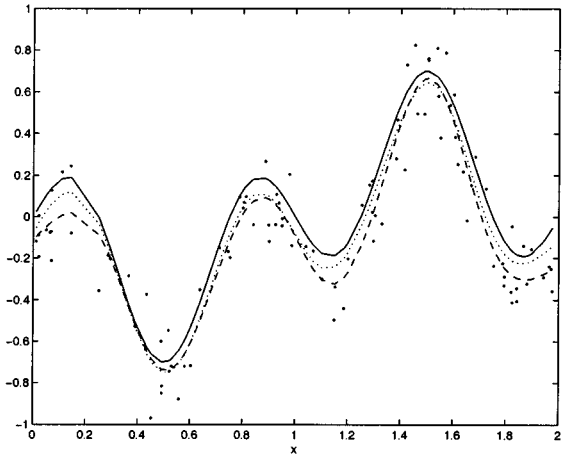
where

$$\begin{cases} \xi_1(i) = \phi_1\xi_1(i-1) + n_1(i) \\ \xi_2(i) = \phi_2\xi_2(i-1) + n_2(i) \end{cases}$$

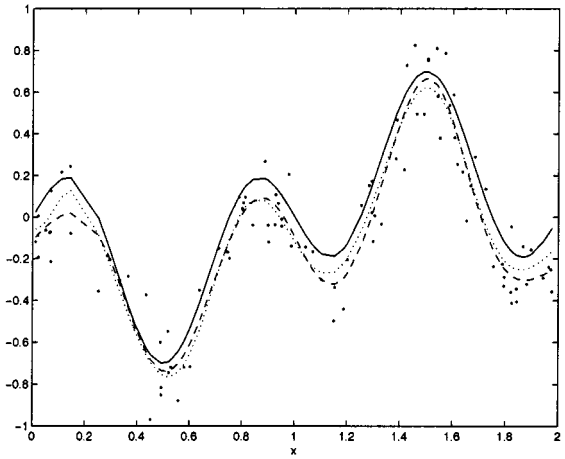
and $(n_1(i), n_2(i)) \sim N(0, \Sigma)$ with $\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix}$ where $\phi_1 = 0.95, \phi_2 = 0.9, \sigma_1 = 0.2, \sigma_2 = 0.2$ and $\sigma_{12} = 0.024$. The functions are chosen intentionally to be sure that a four hidden unit network (with tangent hyperbolic activation) can exactly approximate them. Thus, the noise will be the most important factor influencing the estimation quality and the comparison of our method with the OLS method becomes easier. The jointly Gaussian white noise samples (n_1, n_2) are produced by generating two independent Gaussian sequences with mean zero and variance 1 : $\alpha \sim \mathcal{N}(0,1), \beta \sim \mathcal{N}(0,1)$ and using $n_1 = 0.09(2\alpha + \beta), n_2 = 0.09(2\alpha - \beta)$.

We use a one-hidden layer MLP with tangent hyperbolic activation function, with four units in the hidden layer. The ML estimation is obtained by minimizing (27) using a gradient descent algorithm. Thus, the gradient of this function which contains 23 components (18 network weights and five noise parameters) is computed. Two important problems should be considered here.

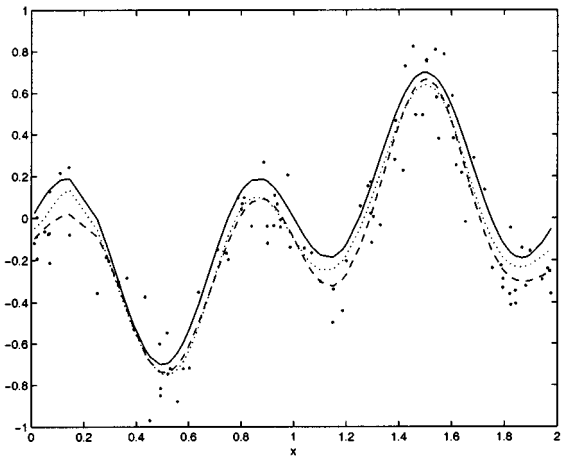
1) The descent trajectory may reach regions of the parameters space where the arguments of the logarithmic terms of (27) are negative. It causes typically an execution error and the search procedure crashes. A good remedy is a variable change strategy which guarantees that the numer-



(a)



(b)



(c)

Fig. 7. Comparison of the estimations obtained by GLS, two-stage and ML methods with OLS estimation. In each figure, the solid line represents the target function, the points its noisy samples, the dashed line the OLS estimation and the dotted line represents (a) GLS estimation, (b) two-stages estimation, and (c) ML estimation.

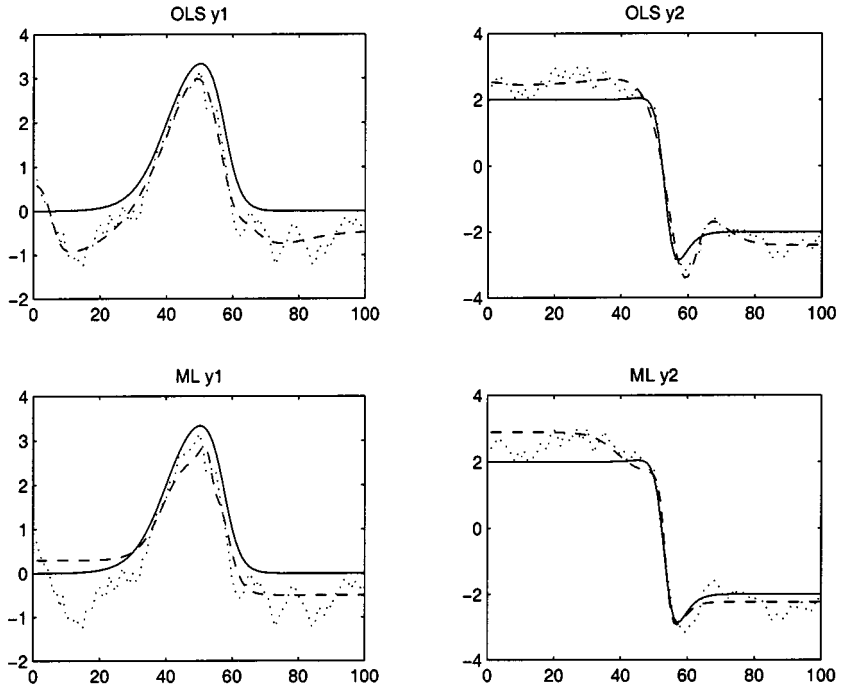


Fig. 8. The result of a sample run for a two-output system. Target functions (solid lines), noisy samples (dots) and the network approximations (dashed lines) by OLS and ML algorithm.

ical search always stays within certain specified boundaries. In our problem, we choose

$$\sigma_{12} = \sigma_1\sigma_2\frac{a}{1+|a|}, \quad \phi_1 = \frac{b}{1+|b|}, \quad \phi_2 = \frac{c}{1+|c|}.$$

It can be easily shown that the above changes lead to: $\sigma_1^2\sigma_2^2 \geq \sigma_{12}^2, (1-\phi_1^2)(1-\phi_2^2) \geq 0$ and $(1-\phi_1\phi_2)^2 \geq (1-\phi_1^2)(1-\phi_2^2)$ so that $|\mathbf{\Lambda}|$ and $|\mathbf{\Sigma}|$ in (26) are positive. Thus, the new parameter vector contains the network weights plus σ_1, σ_2, a, b, and c.

2) Our experiences show that the gradient values for the network weights and the noise parameters have not the same order of magnitude and choosing a fixed learning rate for all the parameters may slow down the algorithm or lead to divergence. To avoid this problem, we used the conjugate gradient method which finds the optimal value of the learning rate using a line minimization algorithm.

The experiment consists of ten runs, each run corresponding to a different noise seed value [31]. To evaluate the results, another network is trained using the OLS algorithm. The results are given in Table I, where performance is evaluated as the distance between the noiseless target signal and the estimations: $\sum_i (\hat{y}_1(i) - s_1(i))^2 + (\hat{y}_2(i) - s_2(i))^2$, which may be used as the test error. As it can be seen, the networks trained by our algorithm have a better performance in generalization, thanks to the noise modeling. As shown in Fig. 8, it is clear that the OLS estimation has a tendency to follow the noisy data, which leads to a good performance on the training data but a weak generalization capacity.

TABLE I
SIMULATION RESULTS FOR OUR ALGORITHM (ML) AND ORDINARY LEAST SQUARE ALGORITHM (OLS)

	ML		OLS	
	mean	STD	mean	STD
test error	63.11	24.90	73.98	35.00
ϕ_1	0.83	0.13	-	-
ϕ_2	0.81	0.07	-	-
σ_1	0.24	0.08	-	-
σ_2	0.24	0.06	-	-
σ_{12}	0.05	0.04	-	-

C. Stopping Criterion in Constructive MLP

In the third experiment, we would like to use the method to whiten the estimation residual in a constructive approach and to use the result as a stopping criterion. The experiment consists in estimating the function $y = \sin(5.5x - 0.5)\cos(10x - 1)$ using 100 samples corrupted by a first-order autoregressive stationary Gaussian noise, with the correlation coefficient $\phi_1 = 0.8$ and the noise variance $\sigma_n^2 = 0.007$. The noise and its correlation function are shown in Fig. 9(a). The network was constructed using the approach mentioned in [16], using a single hidden layer MLP with linear output. At each step of the network construction, the residual of the current network approximation is computed and a single neuron is trained to estimate it. Afterwards, the output of this neuron is added to main network output and the whole network is trained shortly to find the optimal parameters in the new weight space. The network training is done to minimize (45). The method explained in Section IV is used for whitening the residual during the network construction. After adding five units in the hidden layer, the stopping criterion is satisfied. The final whitened residual and its correlation function are shown in Fig. 9(b). Fig. 9(c) illustrates the approximation result. The estimated values for ρ_1 and σ_n^2 are, respectively, 0.8803 and 0.0061. The above experiments have been repeated with different initial values with practically similar results.

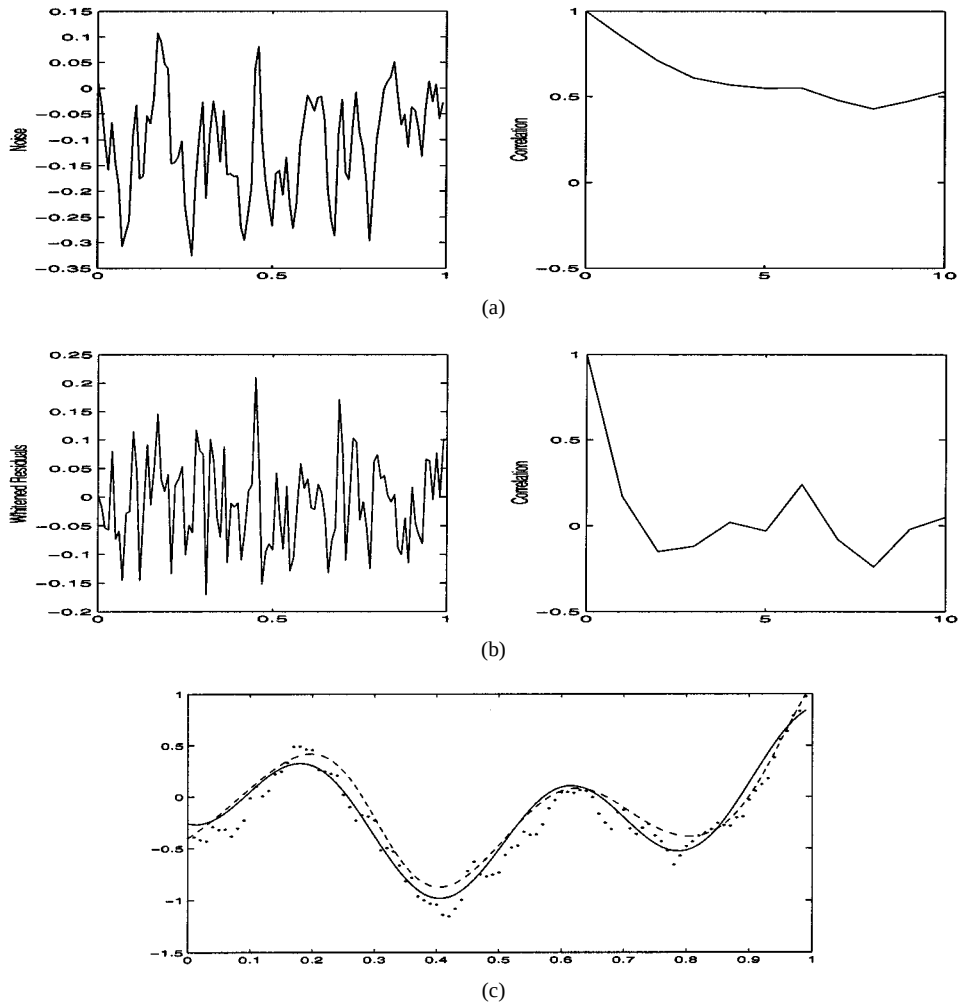


Fig. 9. ML estimation in presence of correlated noise. (a) Simulated autoregressive noise and its correlation function. (b) Final whitened residual and its correlation function. (c) Target function (solid line), its noisy samples (dots) and the network approximation (dashed line).

D. Experience With Real Data

Our last experiment consists in applying our algorithm to a real-world problem. One wishes to model the variation of the blood pressure of a patient as a function of his corporal acceleration and his heart rate. For each patient, one has two series of recordings, each one over 24 h and separated by at least a week. One of the two series can be used as the training base and the other as the test base. See [32] and [33] for additional information on the conditions of recording and the applications.

Unfortunately, for most of the patients, data are not collected at regular time intervals so that our proposed methods cannot be directly applied. We are currently working to generalize the methods for irregular sampled data. However, we present here the results for a patient whose training data are nearly regularly sampled.

To determine the optimal number of neurons in the network hidden layer, we could use a constructive algorithm. However, since the constructive approaches may lead to suboptimal solutions, and our principal objective is to compare the optimal solutions obtained by ML and OLS methods, we preferred to use a trial and error approach, especially as we have a test data set. Thus, the networks containing one to ten hidden neurons are trained by training data[8] and using OLS algorithm. For each network, the test error is evaluated using test data. The network providing minimum test error has four hidden units. The residual of this network estimation (as a function of time) and its autocorrelation function are presented in Fig. 1. The existence of the temporal correlation in the residual can be due to the influence of unknown temporally correlated sources (stress, feeding, activities,...) and can be explored by ML method to improve the estimation. Afterwards, the networks with one to ten hidden units are trained using ML algorithm to minimize the cost function (45), supposing a first-order autoregressive Gaussian noise model. Once more, the network with four neurons provides the best generalization performance.

Training and test errors for the ten networks are given in Table II. Fig. 10 represents the variation of blood pressure and its estimations by OLS and ML methods as functions of logarithm of corporal acceleration and heart rate for the networks with four hidden units. The extremely noisy nature of data can be remarked in this figure. The results confirm the better quality

[8]Data are first centered and divided by their standard deviation.

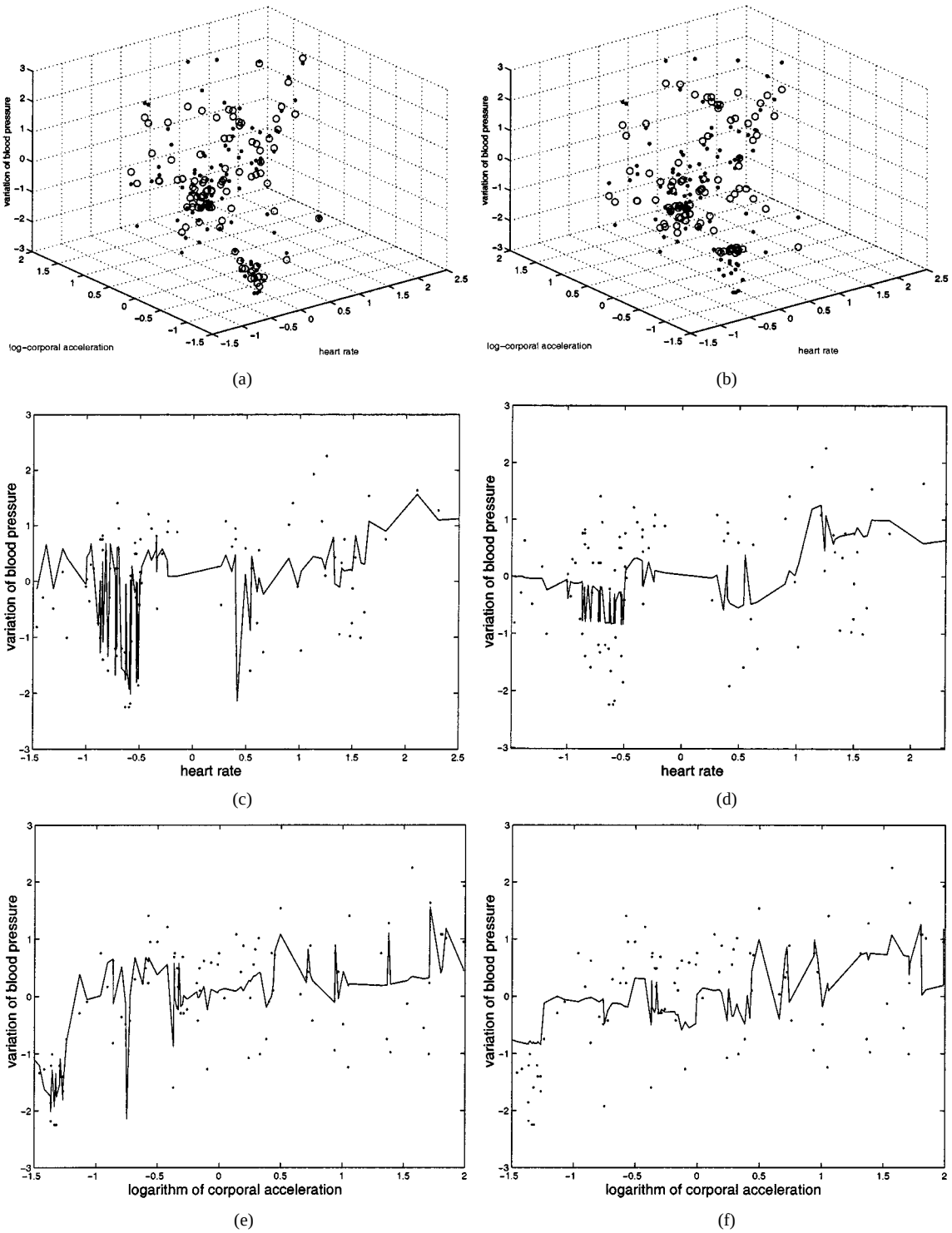


Fig. 10. Variation of blood pressure as a function of heart rate and logarithm of corporal acceleration, and its projections on two axes. (a), (c), and (e) OLS estimation. (b), (d), and (f) ML estimation. The points represent real data. The estimations are represented by the circles in (a) and (b), and by the solid lines in the other figures.

of ML estimator in generalization. Since OLS estimator learns better the noise structure, it is natural that it has a better performance on the training data.

VI. Discussion

In this paper, we addressed the problem of nonlinear regression using MLP with more realistic hypotheses about noise. In particular, we studied the case of spatio-temporally correlated Gaussian noise with autoregressive temporal structure and we showed that using a maximum likelihood approach, the noise parameters may be estimated simultaneously with the MLP weights during the training. It is shown that the generalization performance can be improved using this method.

A multioutput system can be modeled by either a MIMO MLP or several separate MISO MLP. Although the training of

TABLE II
TRAINING AND TEST ERRORS FOR EACH OF OLS AND ML ESTIMATORS AS FUNCTIONS OF THE NUMBER OF HIDDEN NEURONS (N)

	training error		test error	
N	OLS	ML	OLS	ML
1	0.53	0.98	1.44	1.19
2	0.49	1.03	1.97	1.21
3	0.46	1.03	1.43	1.23
4	0.39	0.68	1.37	1.18
5	0.36	1.08	1.95	1.82
6	0.34	0.68	1.69	1.66
7	0.30	0.60	1.71	1.79
8	0.19	0.60	1.72	1.22
9	0.18	0.68	1.36	1.72
10	0.15	0.73	1.58	1.67

several MISO networks seems simpler and does not require to take into account the spatial correlation of the noise, we studied the more general case of MIMO network modeling and showed that the spatial correlation of the noise must be explored in such a case to achieve a maximum likelihood estimation. Anyway, in the absence of *a priori* information about the problem, it seems more reasonable to use several MISO networks.

Although the approach is useful even with fixed size MLP, it is still more interesting when applied together with a network construction scheme. In this case, the estimated noise parameters can be used for whitening the residual and for providing a stopping criterion able to control the network development.

Although in this paper MLP are considered, the method may be used with other universal approximator structures like radial basis functions (RBFs) [34]. Our hypotheses about the noise properties may seem restrictive but they are rather realistic, since any linear system can be modeled using an autoregressive structure. If the noise is non-Gaussian but with a known probability density function, the ML approach is still applicable. In practice, the noise may be also nonstationary. As an example, in many instrumentation devices the noise power increases with the signal amplitude. In [35], we have studied the special case of independent Gaussian noise with a variance function of the output $\sigma^2(y)$, and we have shown that, based on a parametric model of $\sigma^2(y)$ which is itself estimated, the estimation may be improved. However, the more general case of correlated nonstationary noise may be the subject of further studies.

More investigations seem necessary to study the convergence of the solutions. In the general case, it is necessary that the estimation of the spatial covariance matrix $\mathbf{\Sigma}$ be nonnegative definite and the estimations of the temporal correlation matrices $\mathbf{\Phi}_i$ satisfy the conditions of stationarity and causality expressed by (10). But, are these conditions sufficient? The question merits more studies.

Finally, everywhere in this paper, we supposed that data were collected at regular time intervals. Further studies seem necessary for generalizing the method for irregular sampling or missing data values and for applying the method for any actual data. Currently, our investigations are focused on the problems pointed out in the three last paragraphs.

APPENDIX
COMPUTATION OF $\mathbf{\Lambda}_0$ AND $\mathbf{\Lambda}_1$ FOR AR2 MODEL

Multiplying the main autoregressive equation

$$\epsilon(t) = \mathbf{\Phi}_1\epsilon(t-1) + \mathbf{\Phi}_2\epsilon(t-2) + \mathbf{n}(t) \tag{48}$$

by $\epsilon(t)^T, \epsilon(t-1)^T$ and $\epsilon(t-2)^T$, and taking the mathematical expectation of the results, we obtain the following system of equations:

$$\begin{aligned}\mathbf{\Lambda}_0 &= \mathbf{\Phi}_1\mathbf{\Lambda}_1^T + \mathbf{\Phi}_2\mathbf{\Lambda}_2^T + \mathbf{\Sigma}\mathbf{\Lambda}_1\\ &= \mathbf{\Phi}_1\mathbf{\Lambda}_0 + \mathbf{\Phi}_2\mathbf{\Lambda}_1^T\mathbf{\Lambda}_2\\ &= \mathbf{\Phi}_1\mathbf{\Lambda}_1 + \mathbf{\Phi}_2\mathbf{\Lambda}_0.\end{aligned} \tag{49}$$

Using the symmetry of $\mathbf{\Lambda}_0, \mathbf{\Sigma}, \mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$

$$\begin{aligned}\mathbf{\Lambda}_1^T &= \mathbf{\Lambda}_0\mathbf{\Phi}_1 + \mathbf{\Lambda}_1\mathbf{\Phi}_2\\ \mathbf{\Lambda}_2^T &= \mathbf{\Lambda}_1^T\mathbf{\Phi}_1 + \mathbf{\Lambda}_0\mathbf{\Phi}_2.\end{aligned} \tag{50}$$

Replacing in two first equations of (49), $\mathbf{\Lambda}_2$ is eliminated and after simplifying, we obtain

$$\begin{aligned}\mathbf{\Lambda}_0 &= \mathbf{\Phi}_1\mathbf{\Lambda}_0\mathbf{\Phi}_1 + \mathbf{\Phi}_1\mathbf{\Lambda}_1\mathbf{\Phi}_2 + \mathbf{\Phi}_2\mathbf{\Lambda}_0\mathbf{\Phi}_1^2\\ &\quad + \mathbf{\Phi}_2\mathbf{\Lambda}_1\mathbf{\Phi}_2\mathbf{\Phi}_1 + \mathbf{\Phi}_2\mathbf{\Lambda}_0\mathbf{\Phi}_2 + \mathbf{\Sigma}\\ \mathbf{\Lambda}_1 &= \mathbf{\Phi}_1\mathbf{\Lambda}_0 + \mathbf{\Phi}_2\mathbf{\Lambda}_0\mathbf{\Phi}_1 + \mathbf{\Phi}_2\mathbf{\Lambda}_1\mathbf{\Phi}_2.\end{aligned} \tag{51}$$

Using matrix to vector operations and the Kroneker product properties, we can write

$$\begin{aligned}\mathrm{vec}(\mathbf{\Lambda}_0) &= (\mathbf{\Phi}_1 \otimes \mathbf{\Phi}_1)\mathrm{vec}(\mathbf{\Lambda}_0) + (\mathbf{\Phi}_2 \otimes \mathbf{\Phi}_1)\\ &\quad \times \mathrm{vec}(\mathbf{\Lambda}_1) + (\mathbf{\Phi}_1^2 \otimes \mathbf{\Phi}_2)\mathrm{vec}(\mathbf{\Lambda}_0)\\ &\quad + (\mathbf{\Phi}_2\mathbf{\Phi}_1 \otimes \mathbf{\Phi}_2)\mathrm{vec}(\mathbf{\Lambda}_1)\\ &\quad + (\mathbf{\Phi}_2 \otimes \mathbf{\Phi}_2)\mathrm{vec}(\mathbf{\Lambda}_0) + \mathrm{vec}(\mathbf{\Sigma})\\ \mathrm{vec}(\mathbf{\Lambda}_1) &= (\mathbf{I} \otimes \mathbf{\Phi}_1)\mathrm{vec}(\mathbf{\Lambda}_0) + (\mathbf{\Phi}_1 \otimes \mathbf{\Phi}_2)\\ &\quad \times \mathrm{vec}(\mathbf{\Lambda}_0) + (\mathbf{\Phi}_2 \otimes \mathbf{\Phi}_2)\mathrm{vec}(\mathbf{\Lambda}_1)\end{aligned} \tag{52}$$

which can be simplified to

$$\begin{aligned}&(\mathbf{I} - \mathbf{\Phi}_1 \otimes \mathbf{\Phi}_1 - \mathbf{\Phi}_1^2 \otimes \mathbf{\Phi}_2 - \mathbf{\Phi}_2 \otimes \mathbf{\Phi}_2)\mathrm{vec}(\mathbf{\Lambda}_0)\\ &\quad - (\mathbf{\Phi}_2 \otimes \mathbf{\Phi}_1 + \mathbf{\Phi}_2\mathbf{\Phi}_1 \otimes \mathbf{\Phi}_2)\mathrm{vec}(\mathbf{\Lambda}_1) = \mathrm{vec}(\mathbf{\Sigma})\\ &(\mathbf{I} - \mathbf{\Phi}_2 \otimes \mathbf{\Phi}_2)\mathrm{vec}(\mathbf{\Lambda}_1) = (\mathbf{I} \otimes \mathbf{\Phi}_1 + \mathbf{\Phi}_1 \otimes \mathbf{\Phi}_2)\mathrm{vec}(\mathbf{\Lambda}_0)\end{aligned} \tag{53}$$

and leads to

$$\begin{aligned}\mathrm{vec}(\mathbf{\Lambda}_0) &= \mathbf{D}^{-1}\mathrm{vec}(\mathbf{\Sigma})\\ \mathrm{vec}(\mathbf{\Lambda}_1) &= \mathbf{C}\mathrm{vec}(\mathbf{\Lambda}_0)\end{aligned} \tag{54}$$

with $\mathbf{C}$ and $\mathbf{D}$ as defined by (35) and (36).

ACKNOWLEDGMENT

The authors would like to thank Dr. S. Charbonnier and the CHU of Grenoble for the medical data base used in this paper. They are also very grateful to the reviewers for their constructive comments and suggestions which improved the paper.

REFERENCES

[1] G. A. F. Seber and C. J. Wild, *Nonlinear Regression*, ser. probability and mathematical statistics: Wiley, 1989.
[2] G. Cybenko, "Approximation by superposition of sigmoidal functions," *Math. Contr., Signals. Syst.*, vol. 2, pp. 303–314, 1989.
[3] L. Ljung, *System Identification, Theory for the User*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[4] S. Hosseini and C. Jutten, "Improving neural-network estimation in presence of non i.i.d. noise," in *Proc. 6th Europ. Symp. Artificial Neural Networks*, Bruges, Belgium, Apr. 1998, pp. 327–332.
[5] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1994.
[6] J. D. Hamilton, *Time Series Analysis*. Princeton, NJ: Princeton Univ. Press, 1994.
[7] A. R. Gallant and J. J. Goebel, "Nonlinear regression with autocorrelated errors," *J. Amer. Statist. Assoc.*, vol. 71, pp. 961–967, 1976.
[8] C. A. Glasbey, "Correlated residuals in nonlinear regression applied to growth data," *Appl. Statist.*, vol. 28, pp. 251–259, 1979.
[9] J. J. Spitzer, "Small-sample properties of nonlinear least squares and maximum likelihood estimation in the context of autocorrelated errors," *J. Amer. Statist. Assoc.*, vol. 74, pp. 41–47, 1979.
[10] J. P. Le Cadre, "Parametric methods for spatial signal processing in the presence of unknown colored noise fields," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 965–983, July 1989.
[11] M. Viberg, "Sensitivity of parametric direction finding to colored noise fields and undermodeling," in *Signal Processing*. Amsterdam, THe Netherlands: Elsevier, 1993, vol. 34, pp. 207–222.
[12] T. Ash, "Dynamic node creation in backpropagation networks," *Connection Sci.*, vol. 1, no. 4, pp. 365–375, 1989.
[13] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed. Los Altos, CA: Morgan Kaufmann, 1990, pp. 524–532.
[14] J. N. Hwang, S. R. Lat, M. Maechler, D. Martin, and J. Schimert, "Regression modeling in backpropagation and projection pursuit learning," *IEEE Trans. Neural Networks*, vol. 5, pp. 342–353, 1994.
[15] T. Y. Kwok and D. Y. Yeung, "Constructive algorithms for structure learnign in feedforward neural networks for regression problems," *IEEE Trans. Neural Networks*, vol. 8, pp. 630–645, May 1997.
[16] C. Jutten and R. Chentouf, "A new scheme for incremental learning," *Neural Processing Lett.*, vol. 2, no. 1, pp. 1–4, 1995.
[17] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*. New York: Springer-Verlag, 1996.
[18] *The Control Handbook*, CRC, Boca Raton, FL, 1996.
[19] R. F. Galbraith and J. I. Galbraith, "On the inverse of some patterned matrices arising in the theory of stationary time series," *J. Appl. Prob.*, vol. 11, pp. 63–71, 1974.
[20] A. S. Weigend, D. E. Rumelhart, and B. A. Huberman, "Generalization by weight-elimination with application to forecasting," *Neural Inform. Processing Syst. 3*, pp. 875–882, 1991.
[21] P. M. Williams, "Bayesian regularization and pruning using a Laplace prior," *Neural Comput.*, vol. 7, no. 1, pp. 117–143, 1995.
[22] Y. Le Cun, J. Denker, and S. Solla, "Optimal brain damage," in *Advances in Neural Information Processing Systems 2*, Denver, CO, 1990, pp. 598–605.
[23] B. Hassibi and D. Stork, "Second-order derivatives for network pruning: Optimal brain surgeon," *Neural Inform. Processing Syst. 5*, 1993.
[24] T. Y. Kwok and D. Y. Yeung, "Objective functions for training new hidden units in constructive neural networks," *IEEE Trans. Neural Networks*, vol. 8, pp. 1131–1148, Sept. 1997.
[25] R. E. Parker and M. Tummala, "Identification of Volterra systems with a polynomial neural network," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 4, pp. 561–564, Mar. 1992.
[26] J. Moody, *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*. ser. NATO ASI Series F, V. Cherkassky, J. H. Friedman, and H. Wechsler, Eds: Springer, 1994. Prediction risk and architecture selection for neural networks.
[27] B. Efron and R. J. Tibshirani, *Monographs on Statistics and Applied Probability*. New York: Chapman and Hall, 1993, vol. 57. An introduction to the bootstrap.
[28] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automat. Contr.*, vol. AC-19, pp. 716–723, Dec. 1974.
[29] G. Schwartz, "Estimating the dimensino of a model," *Ann. Statist.*, vol. 6, pp. 461–464, 1978.
[30] J. Moody, *Neural Networks for Signal Processing*, B. H. Juang, S. Y. Kung, and C. A. Kamm, Eds. Piscataway, NJ: IEEE Press, 1991.
[31]
[32] S. Charbonnier, J. P. Siché, and J. M. Mallion, "Toward a portable bllos pressure recorder device equipped with accelerometers," *Med. Eng. Phys.*, vol. 21, pp. 343–352, 1999.
[33] S. Charbonnier, S. Galichet, G. Mauris, and J. P. Siché, "Statstical and fuzzy models of ambulatory systolic blood presure for hypertension diagnosis," *Proc. 16th IEEE Instrum. Meas. Technol. Conf.*, pp. 24–26, May 1999.
[34] E. J. Hartman, J. D. Keeler, and J. M. Kowalski, "Layered neural networks with Gaussian hidden units as universal approximations," *Neural Comput.*, vol. 2, no. 2, pp. 210–215, 1990.
[35] S. Hosseini and C. Jutten, "Simulataneous estimation of signal and noise in constructive neural networks," in *Proc. Int. ICSC/IFAC Symp. Neural Comput.*, Vienna, Austria, Sept. 1998, pp. 412–417.

Shahram Hosseini was born in Shiraz, Iran, in 1968. He received the B.Sc. and M.Sc. degrees, both in electrical engineering, from Sharif University of Technology, Tehran, Iran, in 1991 and 1993, respectively, and the Ph.D. degree in signal processing from the Institut National Polytechnique, Grenoble, France, in 2000.

From January 2000 to August 2001, he was Assistant Professor of electrical engineering at INPG, France. He is now a Postdoctoral Fellow in the Laboratoire des Images et des Signaux, Grenoble, France. His research interests include artificial neural networks, blind separation of sources, and adaptive signal processing.

Christian Jutten (A'92) received the Ph.D. degree in 1981 and the Docteur Sciences degree in 1987 from the Institut National Polytechnique, Grenoble, France.

He taught as an Associate Professor in Ecole Nationale Suprieure d'Electronique et de Radioelectricite, Grenoble, from 1982 to 1989. He was Visiting Professor in Swiss Federal Polytechnic Institute, Lausanne, in 1989, before becoming Full Professor in the Sciences and Techniques Institute at the Universite Joseph Fourier, Grenoble. For 15 years, his research interests have been source separation and independent component analysis and learning in neural networks.

Dr. Jutten was Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS from 1994 to 1995, and Co-organizer of the First International Conference on Blind Signal Separation and Independent Component Analysis, Aussois, France, in January 1999. He is currently a member of a technical committee of IEEE Circuits and Systems Society on blind signal processing.