Simultaneous estimation of signal and noise in constructive neural networks

Shahram HOSSEINI & Christian JUTTEN *
INPG-LIS, 46, av. Félix Viallet, 38031 Grenoble Cedex, France.

e-mail: hosseini or chris@tirf.inpg.fr

Abstract

When multilayer perceptrons are used for non linear regression, it is generally supposed that data are corrupted by additive stationary white noise and the ordinary least square method usually used as training algorithm, gives a consistent estimator. However, in many practical situations, the noise may be non stationary or colored. In this case, the estimation of noise parameters can be used to improve the approximation.

Estimation of noise structure is also useful to find a stopping criterion in incremental learning of neural networks. To avoid overfitting, a network construction procedure must be stopped when residue reduces to noise. The knowledge on the noise may be used for "whitening" the residue 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 a few special cases where the noise parameters can be determined simultaneously during incremental neural network training. These parameters will be then used to stop the construction procedure.

1 Introduction

A typical non linear regression problem using MLP can be presented as following. Given a data set:

$$(\mathbf{x}_i, y_i) = (\mathbf{x}_i, g(\mathbf{x}_i) + \epsilon_i),\ i = 1, \ldots, N. \quad (1)$$

where $\mathbf{x}_i$ are samples of a known input variable $\mathbf{x} \in \mathcal{R}^r$, y_i are samples of an output variable $y \in \mathcal{R}$ and ϵ_i model the output noise, we want to find a good approximation of the underlying relationship $g(.)$. Considering the universal approximation property of MLP [6], we can find a suitable size network such that its output is:

$$f(\mathbf{x}, \mathbf{W}^*) = E[y|\mathbf{x}] = g(\mathbf{x}) \quad (2)$$

where $\mathbf{W}^*$ is the optimal weight matrix. Suppose $f(\mathbf{x}, \mathbf{W})$ the class of such networks, we would like to find the best approximation of $\mathbf{W}^*$. When ϵ_i are zero mean, independent and identically distributed (i.i.d.) samples, ordinary least square estimation is usually used, and gives a consistent estimator. However, in many practical situations, ϵ_i may be correlated and/or non stationary. In this case, the knowledge on the noise can improve the approximation using more complicated estimators like Generalized Least Square (GLS) or Maximum Likelihood (ML). Although this case is little studied in the neural network literature, statisticians and automatic control engineers considered it more [5], [7] and neural network implementation of such approaches can improve the quality of estimation.

Another motivation to address this problem is to find a stopping criterion for incremental learning of neural networks. In general, the optimal size of an MLP for a given problem is unknown. A too small network can not learn the data with a sufficient accuracy and a too large one leads to overfitting, and thus to poor generalization. Constructive approaches have been used to solve this problem [2], [1]. In these methods, one starts with a simple network and adds progressively new units until the network can fit the target function. The problem is to define a relevant stopping criterion which avoids uncontrolled network increase. One of the solution proposed recently is to use the residual signal properties [3], [4]. It is clear that an optimal neural network approximation $f(\mathbf{x}, \mathbf{W})$ is achieved if $f(\mathbf{x}, \mathbf{W}) = g(\mathbf{x})$ *i.e.* if the error is $y_i - f(\mathbf{x}_i, \mathbf{W}) = \epsilon_i$ that is when residue reduces to noise. If noise samples ϵ_i are zero-

*Christian Jutten is professor in the Institut des Sciences et Techniques de Grenoble (ISTG) of the Université Joseph Fourier (UJF).

mean and i.i.d. , the stopping criterion may consist in comparing the residue with such noise. In the other words, denoting residue with ξ, the criterion is nothing but simple hypothesis tests: $E[\xi_i] = 0$ versus $E[\xi_i] \neq 0$ and $\forall k \neq 0$, $E[\xi_i \xi_{i+k}] = 0$ versus $E[\xi_i \xi_{i+k}] \neq 0$. 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 estimation of its characteristics can be used for "whitening" the residue. Thus, the mentioned hypothesis tests would be still usable. In this paper, we consider the case of Gaussian noise.

The paper is organized as following: in section 2, we study the simplest case where the covariance matrix of noise is known and show the problem can be solved using GLS method. The case of non-stationary noise is studied in section 3. Section 4 describes the neural network realization of ML for an autoregressive noise model. Section 5 presents the simulation results. Finally, we conclude in section 6.

2 GLS method

Consider the noisy data model (1). Denoting $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_N)^T$, $\mathbf{y} = (y_1, y_2, \ldots, y_N)^T$ and $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^T$ and supposing $\mathbf{f}(\mathbf{X}, \mathbf{W}^*)$ a suitable size MLP for modeling $g(.)$, we have:

$$\mathbf{y} = \mathbf{f}(\mathbf{X}, \mathbf{W}^*) + \boldsymbol{\epsilon} \tag{3}$$

Suppose now the covariance matrix of $\boldsymbol{\epsilon}$ is $\mathcal{D}_{\boldsymbol{\epsilon}} = E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] = \sigma^2\mathbf{V}$, where $\mathbf{V}$ is a positive definite matrix. Let $\mathbf{V} = \mathbf{U}^T\mathbf{U}$ be the Cholesky decomposition of $\mathbf{V}$, where $\mathbf{U}$ is an upper triangular matrix and $\mathbf{U}^T$ is its transpose. Multiplying (3) by $\mathbf{R} = (\mathbf{U}^T)^{-1}$, we obtain:

$$\mathbf{z} = \mathbf{h}(\mathbf{X}, \mathbf{W}^*) + \boldsymbol{\delta} \tag{4}$$

where $\mathbf{z} = \mathbf{R}\mathbf{y}$, $\mathbf{h}(\mathbf{X}, \mathbf{W}^*) = \mathbf{R}\mathbf{f}(\mathbf{X}, \mathbf{W}^*)$ and $\boldsymbol{\delta} = \mathbf{R}\boldsymbol{\epsilon}$. Then $E[\boldsymbol{\delta}] = \mathbf{0}$ and $\mathcal{D}_{\boldsymbol{\delta}} = \sigma^2\mathbf{R}\mathbf{V}\mathbf{R}^T = \sigma^2\mathbf{I}$. Hence, this transformation leads to another model with i.i.d. error. As a result, when $\mathbf{V}$ is known, the stopping criterion in an incremental learning scheme may consist in multiplying the residue by $\mathbf{R}$ and in testing if $\boldsymbol{\delta} = \mathbf{R}\boldsymbol{\epsilon}$ is a white noise. The Ordinary Least Square (OLS) estimation for the model (4), which consists in minimizing:

$$\mathcal{R}(\mathbf{W}) = [\mathbf{z} - \mathbf{h}(\mathbf{X}, \mathbf{W})]^T[\mathbf{z} - \mathbf{h}(\mathbf{X}, \mathbf{W})] \tag{5}$$

is equivalent to the Generalized Least Square (GLS) method for the model (3):

$$\begin{aligned}\mathcal{R}(\mathbf{W}) &= [\mathbf{y} - \mathbf{f}(\mathbf{X}, \mathbf{W})]^T\mathbf{R}^T\mathbf{R}[\mathbf{y} - \mathbf{f}(\mathbf{X}, \mathbf{W})] \\ &= [\mathbf{y} - \mathbf{f}(\mathbf{X}, \mathbf{W})]^T\mathbf{V}^{-1}[\mathbf{y} - \mathbf{f}(\mathbf{X}, \mathbf{W})] \end{aligned} \tag{6}$$

The gradient algorithm for updating the weights is written:

$$\begin{aligned}\Delta\mathbf{W} = -2\mu\{&\textstyle\sum_{i=1}^{N} V_{ii}^{-1}[y_i - f(\mathbf{x}_i, \mathbf{W})]\frac{\partial f(\mathbf{x}_i, \mathbf{W})}{\partial \mathbf{W}} \\ &+ \textstyle\sum_{i=1}^{N}\sum_{j=i+1}^{N} V_{ij}^{-1}([y_i - f(\mathbf{x}_i, \mathbf{W})]\frac{\partial f(\mathbf{x}_j, \mathbf{W})}{\partial \mathbf{W}} \\ &+ [y_j - f(\mathbf{x}_j, \mathbf{W})]\frac{\partial f(\mathbf{x}_i, \mathbf{W})}{\partial \mathbf{W}})\}\end{aligned} \tag{7}$$

In the case of colored noise, this method leads to a better approximation in comparison with minimizing $\|\mathbf{y} - \mathbf{f}(\mathbf{X}, \mathbf{W})\|$ usually used as cost function in MLP. If we have a parametric model of noise, $\mathbf{V}^{-1}$ is also parametric and the algorithm becomes more simple [8].

3 ML estimation for independent non-stationary noise

Suppose now output data are corrupted by a zero mean independent non-stationary Gaussian noise: $\epsilon_i \sim N(0, \sigma_i^2)$. As an example, in many measurement instruments the noise power increases with signal amplitude.

For such noise, the hypothesis tests of section (1) are no longer efficient. In fact, these tests can be performed if significance thresholds can be defined, basically using assumption on the noise probability density function. With Gaussian noise, each threshold depends on the noise variance. If the variance is not constant, a stopping criterion, estimated on all the data, is not significant. As an example, consider figure 1. Here, the network has not given yet a good approximation of signal but, by accident, in the region of poor estimation (where the residue contains the signal and consequently there is a high correlation) the noise is weak. If we could do the hypothesis tests using a local estimation of variance, the stopping criterion would fail and the construction procedure would continue. However, as noise is powerful in the other regions, a global approximation of variance gives a small correlation to variance ratio so that the stopping criterion will be satisfied.

If the variance of different noise samples, σ_i^2, is known, covariance matrix $\mathbf{V}$ mentioned in section 2 is diagonal . Thus $\mathbf{R}$ is also diagonal with the diagonal entries equal to inverse of square roots of diagonal entries of $\mathbf{V}$. Therefore, stopping criterion may consist in dividing each residual sample by the standard deviation of noise for this sample and then in

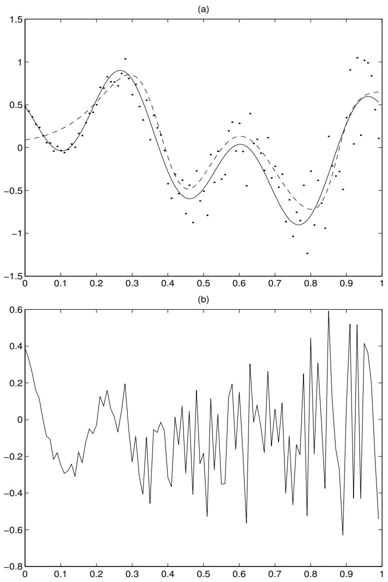

Figure 1: Problem of variable variance. a) Target function (solid line), its noisy samples (dots) and the final network approximation (dashed line). b) Final residue.

comparing this normalized residue with white noise.

When the variances σ_i^2 are unknown, we have to estimate them simultaneously with the network weights. Let's write the likelihood function:

$$P_{\mathbf{Y}}(\mathbf{y}|\sigma_i^2, \mathbf{W}) = \prod_{i=1}^{N} (2\pi\sigma_i^2)^{-1/2} \exp\left(\frac{-[y_i - f(\mathbf{x}_i, \mathbf{W})]^2}{2\sigma_i^2}\right) \tag{8}$$

At first sight, one may imagine that the solution is an ML estimation of $\boldsymbol{\theta} = (\mathbf{W}, \sigma_1, \ldots, \sigma_i, \ldots, \sigma_N)$. However, this approach is not realizable because having only N data samples, the vector $\boldsymbol{\theta}$, whose size is $dim\mathbf{W} + N$, is not identifiable. Another approach, very efficient, can be based on a parametric model of σ_i. For example, in many practical situations, the noise of measurement instruments is a function of signal amplitude. A proper model for σ_i will then be:

$$\sigma_i^2 = a^2|f(\mathbf{x}_i, \mathbf{W}^*)|^b + c^2 \tag{9}$$

In this case the parameter vector which must be estimated is: $\boldsymbol{\theta} = (\mathbf{W}, a, b, c)$. Using (8) the log-likelihood function is written:

$$L_Y(y) = -\frac{1}{2}\sum_{i=1}^{N} \log \sigma_i^2 - \frac{1}{2}\sum_{i=1}^{N} \frac{[y_i - f(\mathbf{x}_i, \mathbf{W})]^2}{\sigma_i^2} \tag{10}$$

and ML estimation of parameters vector $\boldsymbol{\theta}$ is:

$$\boldsymbol{\theta}_{ML} = \arg\min_{\boldsymbol{\theta}} \{\sum_{i=1}^{N} \log \sigma_i^2 + \sum_{i=1}^{N} \frac{[y_i - f(\mathbf{x}_i, \mathbf{W})]^2}{\sigma_i^2}\} \tag{11}$$

Using a gradient descent algorithm to achieve the minimum, the parameters must be updated using the following formulas:

$$\Delta a = -2\mu a \sum_{i=1}^{N} \zeta_i |f(\mathbf{x}_i, \mathbf{W})|^b \tag{12}$$

$$\Delta b = -\mu a^2 \sum_{i=1}^{N} \zeta_i |f(\mathbf{x}_i, \mathbf{W})|^b \log|f(\mathbf{x}_i, \mathbf{W})| \tag{13}$$

$$\Delta c = -2\mu c \sum_{i=1}^{N} \zeta_i \tag{14}$$

$$\Delta\mathbf{W} = -\mu \sum_{i=1}^{N} \frac{\partial f(\mathbf{x}_i, \mathbf{W})}{\partial \mathbf{W}} [\zeta_i a^2 b |f(\mathbf{x}_i, \mathbf{W})|^{b-1} sgn(f(\mathbf{x}_i, \mathbf{W})) - \eta_i] \tag{15}$$

where

$$\zeta_i = \frac{[a^2|f(\mathbf{x}_i, \mathbf{W})|^b + c^2] - [y_i - f(\mathbf{x}_i, \mathbf{W})]^2}{[a^2|f(\mathbf{x}_i, \mathbf{W})|^b + c^2]^2} \tag{16}$$

and

$$\eta_i = \frac{2[y_i - f(\mathbf{x}_i, \mathbf{W})]}{a^2|f(\mathbf{x}_i, \mathbf{W})|^b + c^2} \tag{17}$$

To avoid the ambiguous computation effects like $\log(0)$ and division by zero, we have to replace $|f(\mathbf{x}_i, \mathbf{W})|$ by $|f(\mathbf{x}_i, \mathbf{W}) + \alpha|$ where α is a small constant.

As we discussed in section (1), in practice, the optimal size of MLP is unknown and we use a constructive approach to determine it. In this case, at each step of network construction, we estimate simultaneously the network weights $\mathbf{W}$and the noise variance parameters a,b and c. Then, we divide the residue of estimation by $\sqrt{a^2|f(\mathbf{x}_i, \mathbf{W})|^b + c^2}$ and do the hypothesis tests mentioned in section (1). The results of these tests determine if the construction procedure must be continued or not.

4 ML estimation for stationary autoregressive noise

In this section, we assume that $\mathbf{x}_i$ in (1) are collected at regular time intervals; $\mathbf{x}_i = \mathbf{x}(i\tau)$, and that $\boldsymbol{\epsilon}$ in (3) is a stationary colored noise, resulted from filtering an i.i.d. Gaussian noise $\mathbf{n} \sim N(0, \sigma_n^2)$. The covariance matrix of ϵ is: $\mathcal{D}_{\epsilon} = E[\epsilon\epsilon^T] = \sigma_n^2 \mathbf{V}$ which is supposed unknown. The likelihood function is:

$$P_{\epsilon}(\epsilon|\sigma_n^2, \mathbf{V}^{-1}, \mathbf{W}) = \frac{|\mathbf{V}^{-1}|^{1/2}}{(2\pi\sigma_n^2)^{N/2}} exp(\frac{-\epsilon^T \mathbf{V}^{-1}\epsilon}{2\sigma_n^2}) \tag{18}$$

and log-likelihood:

$$L_{\epsilon}(\epsilon) = -\frac{N log(\sigma_n^2)}{2} + \frac{log|\mathbf{V}^{-1}|}{2} - \frac{\epsilon^T \mathbf{V}^{-1}\epsilon}{2\sigma_n^2} \tag{19}$$

As $\mathbf{V}^{-1}$ is an $N \times N$ matrix, with $\frac{N^2+N}{2}$ independent parameters, it can not be identified using N data samples. Therefore, we have to choose a pertinent parametric noise model. In the following we suppose that colored noise,ϵ, can be modeled using a k_{th} order autoregressive (AR) model.[1]

$$\epsilon_i = \rho_1 \epsilon_{i-1} + \rho_2 \epsilon_{i-2} + \ldots + \rho_k \epsilon_{i-k} + n_i \tag{20}$$

where $n_i \sim N(0, \sigma_n^2)$. In practice, we have data only for a limited interval $\{i = 1, \ldots, N\}$. As a result, the relation (20) is not usable for the k first samples. Denoting $\epsilon^{\mathbf{k}} = (\epsilon_1, \ldots, \epsilon_k)$ et $\mathbf{n}^{\mathbf{k}} = (n_1, \ldots, n_k)$, we have:

$$\mathbf{n}^{\mathbf{k}} = \mathbf{R}^{\mathbf{k}} \epsilon^{\mathbf{k}} \tag{21}$$

where $\mathbf{R}^{\mathbf{k}}$ is a $K \times K$ lower triangular matrix. Hence, denoting $\epsilon = (\epsilon_1, \ldots, \epsilon_N)$ et $\mathbf{n} = (n_1, \ldots, n_N)$, we can write:

$\mathbf{n} = \mathbf{R}\epsilon$,

$$\mathbf{R} = \begin{pmatrix} & & \mathbf{R}^{\mathbf{k}} & & & \mathbf{0} & & \\ & & & & & & & \\ -\rho_k & \ldots & -\rho_2 & -\rho_1 & 1 & 0 & \ldots & 0 \\ 0 & -\rho_k & \ldots & -\rho_2 & -\rho_1 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \ldots & \ldots & \ldots & \ldots & 1 \end{pmatrix} \tag{22}$$

The main advantage of autoregressive model is that there is a simple relation between $\mathbf{V}^{-1}$ and $\mathbf{R}$. Using (22) we have: $\sigma_n^2 \mathbf{V} = E[\epsilon\epsilon^T] = \mathbf{R}^{-1} E[\mathbf{n}\mathbf{n}^T] \mathbf{R}^{-T} = \sigma_n^2 \mathbf{R}^{-1} \mathbf{R}^{-T}$ so that:

$$\mathbf{V}^{-1} = \mathbf{R}^T \mathbf{R} \tag{23}$$

Thus in equation (19) we have:

$$|\mathbf{V}^{-1}| = |\mathbf{R}|^2 = |\mathbf{R}^{\mathbf{k}^T} \mathbf{R}^{\mathbf{k}}| \tag{24}$$

and

$$\epsilon^T \mathbf{V}^{-1} \epsilon = \epsilon^T \mathbf{R}^T \mathbf{R} \epsilon = \epsilon^{\mathbf{k}T} \mathbf{R}^{\mathbf{k}^T} \mathbf{R}^{\mathbf{k}} \epsilon^{\mathbf{k}} + \sum_{i=k+1}^{N} (\epsilon_i - \rho_1 \epsilon_{i-1} - \ldots - \rho_k \epsilon_{i-k})^2 \tag{25}$$

[1]This choice is realistic in many situations and simplifies the computations.

$\mathbf{R}^{\mathbf{k}^T} \mathbf{R}^{\mathbf{k}}$ can be determined using (21): $E[\epsilon^{\mathbf{k}} \epsilon^{\mathbf{k}T}] = \sigma_n^2 \mathbf{R}^{\mathbf{k}^{-1}} \mathbf{R}^{\mathbf{k}^{-T}}$. Hence:

$$\mathbf{R}^{\mathbf{k}^T} \mathbf{R}^{\mathbf{k}} = [\frac{E[\epsilon^{\mathbf{k}} \epsilon^{\mathbf{k}T}]}{\sigma_n^2}]^{-1} \tag{26}$$

For small values of k, (26) can be easily computed using stationary properties of ϵ. In the following subsections, we study two special cases $k = 1$ and $k = 2$. For the other values of k, the procedure is similar.

4.1 ML for first order AR noise

In this case, we have:

$$\epsilon_i = \rho_1 \epsilon_{i-1} + n_i \tag{27}$$

and using (26):

$$\mathbf{R}^{\mathbf{1}^T} \mathbf{R}^{\mathbf{1}} = [\frac{E[\epsilon^1 \epsilon^{1T}]}{\sigma_n^2}]^{-1} = \frac{\sigma_n^2}{\sigma_\epsilon^2} \tag{28}$$

Computing $E[\epsilon_i^2]$ using (27) and noting that ϵ_{i-1} is independent of n_i, we obtain: $\frac{\sigma_n^2}{\sigma_\epsilon^2} = 1 - \rho_1^2$. Thus, denoting the parameters vector by $\boldsymbol{\theta} = (\mathbf{W}, \rho_1, \sigma_n^2)$, the samples of the estimation residue by $\xi_i = y_i - f(\mathbf{x}_i, \mathbf{W})$ and using (19), (24), (25) and (28), the ML estimator of $\boldsymbol{\theta}$ can be obtained:

$$\boldsymbol{\theta}_{ML} = \arg\min_{\boldsymbol{\theta}} \{N log(\sigma_n^2) - log(1 - \rho_1^2) + \frac{1}{\sigma_n^2} [(1 - \rho_1^2)\xi_1^2 + \sum_{i=2}^{N} (\xi_i - \rho_1 \xi_{i-1})^2]\} \tag{29}$$

A gradient descent algorithm updates the parameters using the following formulas:

$$\Delta\sigma_n^2 = -\mu\{\frac{N}{\sigma_n^2} - \frac{1}{\sigma_n^4}[(1-\rho_1)^2 \xi_1^2 + \sum_{i=2}^{N} (\xi_i - \rho_1 \xi_{i-1})^2]\} \tag{30}$$

$$\Delta\rho_1 = -2\mu\{\frac{\rho_1}{1 - \rho_1^2} - \frac{1}{\sigma_n^2}[\rho_1 \xi_1^2 + \sum_{i=2}^{N} \xi_{i-1}(\xi_i - \rho_1 \xi_{i-1})]\} \tag{31}$$

$$\Delta\mathbf{W} = \frac{2\mu}{\sigma_n^2}\{\xi_1(1 - \rho_1^2)\frac{\partial f(\mathbf{x}_1, \mathbf{W})}{\partial \mathbf{W}} + \sum_{i=2}^{N} (\xi_i - \rho_1 \xi_{i-1}) (\frac{\partial f(\mathbf{x}_i, \mathbf{W})}{\partial \mathbf{W}} - \rho_1 \frac{\partial f(\mathbf{x}_{i-1}, \mathbf{W})}{\partial \mathbf{W}})\} \tag{32}$$

Note that the complexity of algorithm is very close to classical backpropagation algorithm.

4.2 ML for second order AR noise

Suppose now noise samples ϵ_i satisfy a second order noise model:

$$\epsilon_i = \rho_1\epsilon_{i-1} + \rho_2\epsilon_{i-2} + n_i \tag{33}$$

Denoting $\beta = \frac{E[\epsilon_i\epsilon_{i-1}]}{\sigma_\epsilon^2}$ and using (26), we have:

$$\mathbf{R^2}^T\mathbf{R^2} = \frac{\sigma_n^2}{\sigma_\epsilon^2(1-\beta^2)}\begin{pmatrix} 1 & -\beta \\ -\beta & 1 \end{pmatrix} \tag{34}$$

Multiplying (33) by ϵ_i, ϵ_{i-1} and ϵ_{i-2} and solving resulted equation system (usually known as Yule-Walker equations), we can represent σ_ϵ^2 and β using σ_n^2, ρ_1 and $\rho2$:

$$\sigma_\epsilon^2 = \frac{1-\rho_2}{(1+\rho_2)[(1-\rho_2)^2-\rho_1^2]}\sigma_n^2 \ , \ \beta = \frac{\rho_1}{1-\rho_2} \tag{35}$$

Denoting $\boldsymbol{\theta} = (\mathbf{W}, \rho_1, \rho_2, \sigma_n^2)$, $\xi_i = y_i - f(\mathbf{x}_i, \mathbf{W})$ and using (19), (24), (25) and (34), the ML estimator of $\boldsymbol{\theta}$ is:

$$\begin{aligned}\boldsymbol{\theta}_{ML} = \arg\min_{\boldsymbol{\theta}} \{&(N-2)log(\sigma_n^2) + 2log(\sigma_\epsilon^2) + \\ & log(1-\beta^2) + \frac{\xi_1^2 - 2\beta\xi_1\xi_2 + \xi_2^2}{\sigma_\epsilon^2(1-\beta^2)} + \\ & \frac{1}{\sigma_n^2}\sum_{i=3}^{N}(\xi_i - \rho_1\xi_{i-1} - \rho_2\xi_{i-2})^2\}\end{aligned} \tag{36}$$

Replacing (35) in (36):

$$\begin{aligned}\boldsymbol{\theta}_{ML} \quad = \quad & \arg\min_{\boldsymbol{\theta}} \{N log(\sigma_n^2) - log[(1-\rho_2)^2 - \rho_1^2] \\ -2log(1+\rho_2) + & \frac{1+\rho_2}{\sigma_n^2}[(\xi_1^2+\xi_2^2)(1-\rho_2) - 2\xi_1\xi_2\rho_1] \\ & + \frac{1}{\sigma_n^2}\sum_{i=3}^{N}(\xi_i - \rho_1\xi_{i-1} - \rho_2\xi_{i-2})^2\}\end{aligned} \tag{37}$$

The parameters will be updated using:

$$\begin{aligned}\Delta\sigma_n^2 = -\mu\{&\frac{N}{\sigma_n^2} - \frac{(1+\rho_2)}{\sigma_n^4}[(\xi_1^2+\xi_2^2)(1-\rho_2) - \\ & 2\xi_1\xi_2\rho_1] - \frac{1}{\sigma_n^4}\sum_{i=3}^{N}(\xi_i - \rho_1\xi_{i-1} - \rho_2\xi_{i-2})^2\}\end{aligned} \tag{38}$$

$$\begin{aligned}\Delta\rho_1 = -2\mu\{&\frac{\rho_1}{(1-\rho_2)^2-\rho_1^2} - \frac{\xi_1\xi_2(1+\rho_2)}{\sigma_n^2} - \\ & \frac{1}{\sigma_n^2}\sum_{i=3}^{N}\xi_{i-1}(\xi_i - \rho_1\xi_{i-1} - \rho_2\xi_{i-2})\}\end{aligned} \tag{39}$$

$$\begin{aligned}\Delta\rho_2 \quad = \quad & 2\mu\{\frac{1}{1+\rho_2} - \frac{1-\rho_2}{(1-\rho_2)^2-\rho_1^2} \\ + \quad & \frac{\rho_2(\xi_1^2+\xi_2^2)+\rho_1\xi_1\xi_2}{\sigma_n^2} \\ + \quad & \frac{1}{\sigma_n^2}\sum_{i=3}^{N}\xi_{i-2}(\xi_i - \rho_1\xi_{i-1} - \rho_2\xi_{i-2})\}\end{aligned} \tag{40}$$

$$\begin{aligned}\Delta\mathbf{W} \quad = \quad & \frac{2\mu}{\sigma_n^2}\{(1+\rho^2)[(1-\rho_2)(\xi_1\frac{\partial f(\mathbf{x}_1,\mathbf{W})}{\partial\mathbf{W}} \\ + \quad & \xi_2\frac{\partial f(\mathbf{x}_2,\mathbf{W})}{\partial\mathbf{W}}) - \rho_1(\xi_1\frac{\partial f(\mathbf{x}_2,\mathbf{W})}{\partial\mathbf{W}} \\ + \quad & \xi_2\frac{\partial f(\mathbf{x}_1,\mathbf{W})}{\partial\mathbf{W}})] + \sum_{i=3}^{N}(\xi_i - \rho_1\xi_{i-1} \\ - \quad & \rho_2\xi_{i-2})(\frac{\partial f(\mathbf{x}_i,\mathbf{W})}{\partial\mathbf{W}} - \rho_1\frac{\partial f(\mathbf{x}_{i-1},\mathbf{W})}{\partial\mathbf{W}} \\ - \quad & \rho_2\frac{\partial f(\mathbf{x}_{i-2},\mathbf{W})}{\partial\mathbf{W}})\}\end{aligned} \tag{41}$$

At each step of network construction, using the estimated noise parameters, the matrix $\mathbf{R}$ can be obtained from (22) and (28) for first order noise or (22), (34) and (35) for second order noise. Referring to the discussions of section 2, the stopping criterion consists then in multiplying the residue of approximation by $\mathbf{R}$ and in comparing the result with white noise.

5 Simulation results

In the first experiment, we try to estimate the function $y = sin(7x - 3.5)cos(14x - 7)$, $x \in [0, 1]$, from 100 samples corrupted by an independent non stationary Gaussian noise $\sim N(0, \sigma_i^2)$, where $\sigma_i^2 = 0.09|y_i|^2 + 0.01$. The noise is illustrated in Fig.2-a. The network was constructed using the approach mentioned in [3]. In this approach, the network is developed in a single hidden layer. At each step of network construction, the residue of approximation is computed and a single neuron tries 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.

We used the ML method mentioned in section 3 to estimate simultaneously the variance parameters and the network weights. At each step, the residue is divided by estimated variance and the hypothesis tests mentioned in section 1 are used to determine if the result is an i.i.d. sequence or not. If this is the case, the construction procedure (as detailed above) will be stopped. In the experiment, this stopping criterion was satisfied with a 6 hidden unit network. The final normalized residue, practically stationary and i.i.d., is shown in Fig.2-b, and Fig.2-c illustrates the approximation result. The estimated variance parameters, a^2, b, c^2 are respectively 0.102, 1.905 and 0.0072, which are very close to actual values (0.09, 2 and 0.01).

The second 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 $\rho_1 = 0.8$ and $\sigma_n^2 = 0.007$. The noise and its correlation function are shown in Fig.3-a. The ML approach of subsection 4.1 was used for whitening the residue, simultaneously with construction of the network. The construction strategy is the same as in previous experiment. After adding 5 units in the hidden layer, the stopping criterion was satisfied. The final whitened residue and its correlation function are shown in Fig.3-b, and Fig.3-c illustrates the approximation result. The estimated values for ρ_1 and σ_n^2 were respectively 0.8803 and 0.0061.

The above experiments were repeated with different initial values and the results were practically similar.

6 Conclusion

In this paper, we addressed the problem of function approximation by neural networks with more realistic hypothesis about noise. In particular, we studied two special cases: a non stationary independent noise whose variance is a function of the signal amplitude and a stationary autoregressive noise. We used the ML estimation which gives a consistent estimator even with colored noise and allows to estimate the noise parameters during neural network training.

Although the approaches are useful even with the fixed size MLP, they are more interesting when applied in an incremental learning scheme. In this case, estimated noise parameters can be used for whitening the residue and for establishing a stopping criterion to control the network development.

Our hypothesis about the noise properties seem restrictive but they are rather realistic. In many measurement instruments, the noise power is a function of signal amplitude. Moreover, note that all the linear procedures can be modeled using an autoregressive structure. If the noise is non Gaussian but the shape of its probability density function is known, the ML approach is still applicable.

If we choose a very high order AR noise model, it risks to fit a part of the signal (and not only the noise) so that the neural network estimation will be wrong.

Finally, if the noise distribution is unknown, one could estimate it (a few approaches are available in statistics) and use the result to drive a quasi ML estimate. We are currently exploring this approach.

References

[1] T.Y. Kwok and D.Y. Yeung, Constructive Algorithms for Structure Learning in Feedforward Neural Networks for Regression Problems. *IEEE Trans. on Neural Networks*, vol. 8, no. 3, May 1997.

[2] S.E. Fahlman and C. Lebiere, The Cascade-Correlation Learning Architecture, in*Advances in Neural Information Processing Systems 2*, D.S. Touretzky Ed., pp. 524-532, Morgan Kaufmann, Los Altos CA, 1990.

[3] Ch. Jutten and R. Chentouf. A New Scheme for Incremental Learning. *Neural Processing Letters*, 2(1):1-4, 1995.

[4] R. Chentouf and C. Jutten. DWINA: Depth and Width Incremental Neural Algorithm. In *IEEE International Conference on Neural Networks Proceedings*, pages 153-158, 1996.

[5] G.A.F. Seber and C.J. Wild. *Nonlinear Regression.* John Wiley & sons, 1989.

[6] K. Hornik, M. Stinchcombe and H. White. Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*, 2:359-366, 1989.

[7] L. Ljoung. *System Identification, Theory for the user.* Prentice Hall, Englewood Cliffs, 1987.

[8] S. Hosseini and Ch. Jutten. Improving Neural Network Estimation in Presence of non i.i.d. Noise. In *6th European Symposium on Artificial Neural Networks Proceedings.* Bruges, Belgium. pages 327-332, April 1998.

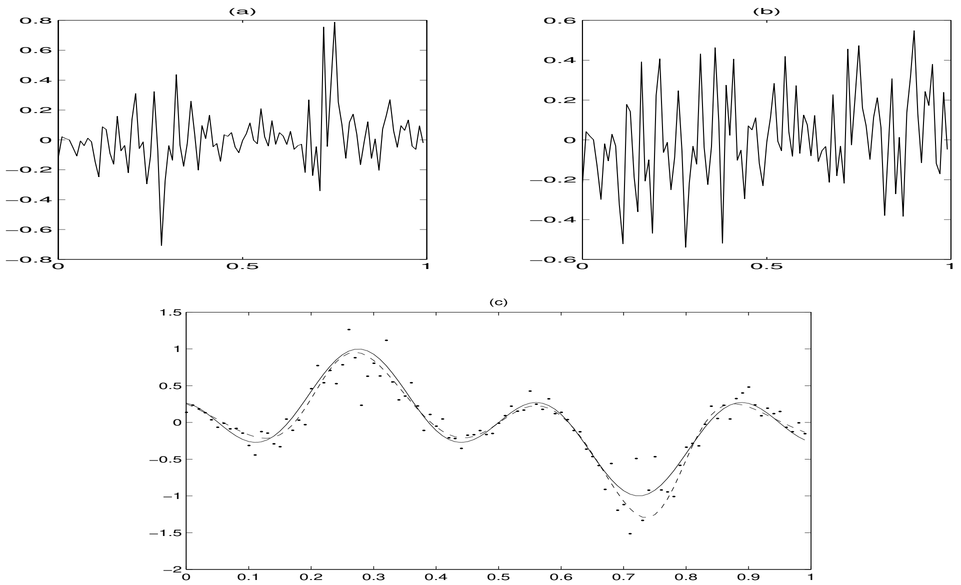

Figure 2: ML estimation in presence of non stationary noise. a) Non stationary simulated noise. b) Final normalized residue. c) Target function (solid line), its noisy samples (dots) and the network approximation (dashed line)

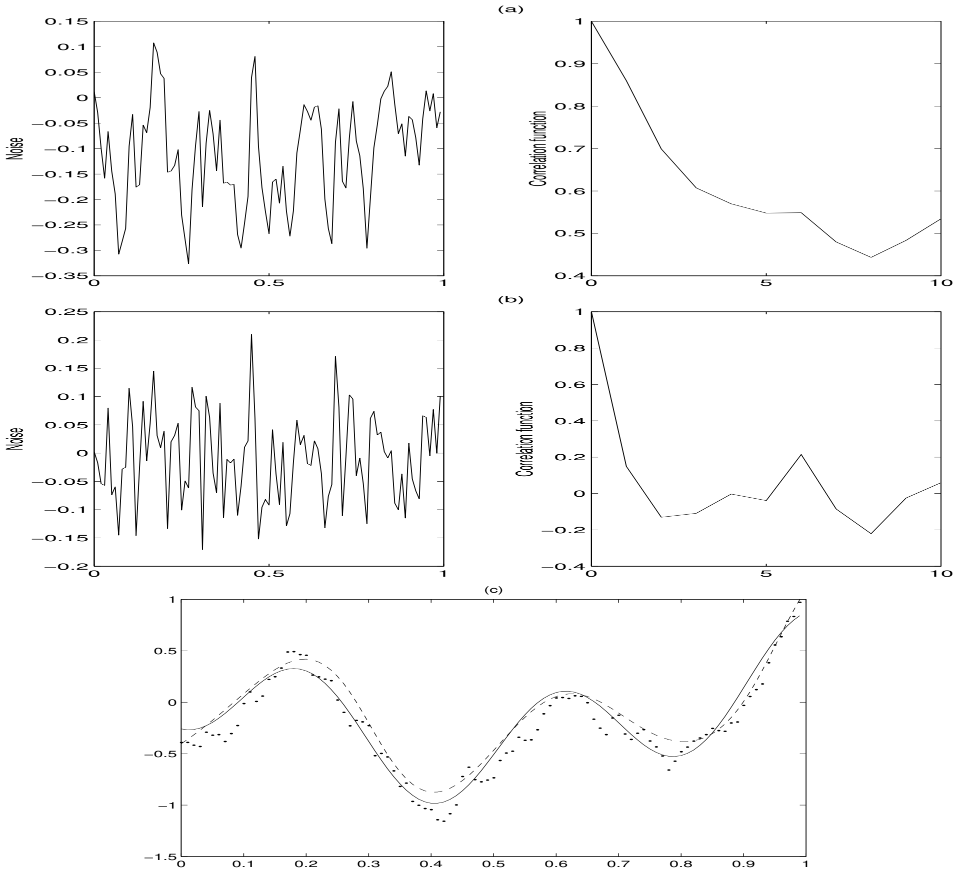

Figure 3: ML estimation in presence of correlated noise. a) Simulated autoregressive noise and its correlation function. b) Final whitened residue and its correlation function. c) Target function (solid line), its noisy samples (dots) and the network approximation (dashed line).