Improving Neural Network Estimation in Presence of non i.i.d. Noise

Shahram HOSSEINI & Christian JUTTEN *

INPG-LIS, 46, av. Félix Viallet, 38031 Grenoble Cedex, France.
e-mail: hosseini or chris@tirf.inpg.fr

Abstract. Basically, MLP are trained by ordinary least square methods which insure consistent estimations only if data are corrupted with additive white noise. Unfortunately, this assumption is not very realistic in many practical situations. In this paper, we show how the generalized least square method, well known in statistics and in automatic control, can be used in MLP with modified backpropagation algorithm, and can improve the estimation when data are corrupted by colored noise.

1. Introduction

Feedforward neural networks trained by error backpropagation are used widely for nonlinear regression. Given a data set $(\mathbf{x}_i, y_i = g(\mathbf{x}_i) + n_i)$ where $\mathbf{x}_i$ are samples of a variable $\mathbf{x} \in \mathcal{R}^r$, y_i are samples of a variable $y \in \mathcal{R}$ and n_i are zero-mean noise samples (usually supposed statistically independent of $\mathbf{x}_i$), we want to find a good approximation of the underlying relationship $g(.)$.

When n_i are i.i.d. variables, it has been proved that least square regressor is consistent [1]. It means that for a large number of data, the estimation residue can be considered as a white noise. This may be used as a stopping criterion for incremental learning of neural networks [2], [3]; to avoid the overfitting, the network growing procedure is stopped when the residue reduces to white noise. Evidently, if the noise in the model is not i.i.d., this criterion fails and the algorithm may progress uncontrollably toward overfitting.

Although in the neural network literature, the case of non-i.i.d. noise is little studied, the statisticians and the automatic control engineers considered it more [4], [5]; because in many realistic situations, additive noise is colored, for example due to transfer function of the measurement devices. The Generalized Least Square (GLS) algorithm is one of the methods which has been proposed for improving the estimation when the additive noise samples are correlated and/or non stationary.

In this paper, we study the neural network realization of the GLS method and in particular, its variants in the case of autoregressive noise models for

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

two purposes: (i) improving neural network approximation, (ii) Improving the stopping criterion of the incremental algorithms. In section 2, we present the GLS method. Section 3 describes the algorithm for the autoregressive noise models. Simulation results are presented in section 4, before the conclusion.

2. Generalized least square method

Assume data corrupted by zero-mean non-i.i.d. noise samples ϵ_i:

$$(\mathbf{x}_i, y_i) = (\mathbf{x}_i, g(\mathbf{x}_i) + \epsilon_i), \; i = 1, \ldots, n. \tag{1}$$

Considering the universal approximation property of MLP [6], we can find a suitable size MLP such that $f(\mathbf{x}, \mathbf{W}^*) = E[y|\mathbf{x}] = g(\mathbf{x})$, $\mathbf{W}^*$ 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 $\mathbf{W}^*$. Denoting $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)^T$, $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$ and $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)^T$, we have:

$$\mathbf{y} = \mathbf{f}(\mathbf{X}, \mathbf{W}^*) + \boldsymbol{\epsilon} \tag{2}$$

Suppose 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. This hypothesis implies (i) $Var\{\boldsymbol{\epsilon}\}$ may be proportional to some function of $\mathbf{x}$, (ii) $\boldsymbol{\epsilon}$ may be correlated.

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 (2) by $\mathbf{R} = (\mathbf{U}^T)^{-1}$, we obtain:

$$\mathbf{z} = \mathbf{h}(\mathbf{X}, \mathbf{W}) + \boldsymbol{\delta} \tag{3}$$

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 $E[\boldsymbol{\delta}] = 0$ and $\mathcal{D}_{\boldsymbol{\delta}} = \sigma^2\mathbf{R}\mathbf{V}\mathbf{R}' = \sigma^2\mathbf{I}$. Hence, this transformation leads to another model with i.i.d. error. The Ordinary Least Square (OLS) estimation for the model (3) consists in minimizing: $\mathcal{R}(\mathbf{W}) = [\mathbf{z} - \mathbf{h}(\mathbf{X}, \mathbf{W})]^T[\mathbf{z} - \mathbf{h}(\mathbf{X}, \mathbf{W})] = [\mathbf{y} - \mathbf{f}(\mathbf{X}, \mathbf{W})]^T\mathbf{R}^T\mathbf{R}[\mathbf{y} - \mathbf{f}(\mathbf{X}, \mathbf{W})]$ and we have:

$$\mathcal{R}(\mathbf{W}) = [\mathbf{y} - \mathbf{f}(\mathbf{X}, \mathbf{W})]^T\mathbf{V}^{-1}[\mathbf{y} - \mathbf{f}(\mathbf{X}, \mathbf{W})] \tag{4}$$

The minimization of this last function is usually called Generalized Least Square (GLS) estimation of (2). Therefore, the GLS method for a colored noise model is equivalent to the OLS method for a white noise model deduced from the colored model by a simple linear mapping. Moreover, according to (3), if we have a good approximation of $f(\mathbf{x}, \mathbf{W})$ (so $h(\mathbf{x}, \mathbf{W})$), $\boldsymbol{\delta}$ is a white noise. Hence, the stopping criterion in an incremental learning scheme may consist in comparing $\delta = \mathbf{R}\epsilon$ with white noise.

As $\mathbf{V}$ is basically unknown, we have to estimate it. A possible algorithm can be the following: (i) OLS estimating of $\mathbf{W}$, computing the residue, (ii) Is the residue white? If yes, STOP, (iii) Estimating $\mathbf{V}$ using residue, (iv) Computing $\mathbf{V}^{-1}$, (v) GLS estimating of $\mathbf{W}$. Another strategy, more time-consuming, consists in repeating the steps (iii)-(v) until convergence.

There are a few difficulties with these strategies. At first, unless we have multiple measurements y_i for same input $\mathbf{x}_i$, estimation of $\mathbf{V}$ will be inexact and even impossible. Moreover, the approximation error of $\hat{\mathbf{V}}$ implies a larger error in the computation of its inverse. On the other hand, for large training data bases, the inverse computation is expensive. Finally, if all the entries of $\hat{\mathbf{V}}^{-1}$ are non-zero, the backpropagation algorithm will be very time-consuming. To avoid these problems, we can suppose an a priori model of noise. In the following section, we choose autoregressive models for their ability to model sparsely a large class of colored noise.

3. GLS for autoregressive models

Suppose $\boldsymbol{\epsilon}$ can be modeled by the autoregressive (AR) model:

$$\boldsymbol{\epsilon} = \mathbf{A}\boldsymbol{\epsilon} + \mathbf{n}\ , \quad \mathbf{n}\ \ i.i.d. \tag{5}$$

$\mathbf{A}$ characterizes the influence of ϵ_i on ϵ_j. If $(\mathbf{I}-\mathbf{A})$ is invertible, (5) becomes: $\boldsymbol{\epsilon} = (\mathbf{I}-\mathbf{A})^{-1}\mathbf{n}$ and the covariance of $\boldsymbol{\epsilon}$ is: $E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] = \sigma^2\mathbf{V} = (\mathbf{I}-\mathbf{A})^{-1}E[\mathbf{n}\mathbf{n}^T](\mathbf{I}-\mathbf{A})^{-1^T}$. Finally, as $E[\mathbf{n}\mathbf{n}^T] = \sigma^2\mathbf{I}$:

$$\mathbf{V}^{-1} = (\mathbf{I}-\mathbf{A}^T)(\mathbf{I}-\mathbf{A}) \tag{6}$$

The main advantage of this model is the possibility of approximating $\mathbf{V}^{-1}$ directly from $\mathbf{A}$. Usually $\mathbf{A}$ is defined by a few parameters so that: (i) Its estimation is very simple, (ii) the most of the entries of $\mathbf{V}^{-1}$ being zero, the time of backpropagation algorithm stays reasonable. Now, we present the backpropagation algorithm for two special cases.

3.1. First order AR noise for dynamic systems and time series

In this section, we assume that x_i in (1) are collected at regular time intervals[1]; $x_i = x(i\tau)$. Suppose that the ϵ_i form a stationary series satisfying a first order autoregressive model:

$$\epsilon_i = \rho\epsilon_{i-1} + n_i\ , \quad n_i\ \ i.i.d.\ \ , \ |\rho| < 1 \tag{7}$$

As n_i is independent of ϵ_{i-1}: $\sigma_\epsilon^2 = \frac{\sigma_n^2}{1-\rho^2}$. In practice, we have data only for a limited interval $\{i = 1, \ldots, N\}$. For ϵ_1, if we select $\epsilon_1 = n_1$, the variance of ϵ_i will not be constant. Hence, in our model we admit $\epsilon_1 = \alpha n_1$ and we determine α so that $\sigma_{\epsilon_1}^2 = \sigma_{\epsilon_i}^2 = \frac{\sigma_n^2}{1-\rho^2}$. So we obtain: $\epsilon_1 = \frac{n_1}{\sqrt{1-\rho^2}}$. Then $\mathbf{A}$ and $\mathbf{V}^{-1}$ in (5) and (6) will be:

$$\mathbf{A} = \begin{pmatrix} 1-\sqrt{1-\rho^2} & 0 & 0 & \ldots & 0 & 0 \\ \rho & 0 & 0 & \ldots & 0 & 0 \\ 0 & \rho & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \rho & 0 \end{pmatrix} \quad \mathbf{V}^{-1} = \begin{pmatrix} 1 & -\rho & 0 & \ldots & 0 \\ -\rho & 1+\rho^2 & -\rho & \ldots & 0 \\ 0 & -\rho & 1+\rho^2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1+\rho^2 \end{pmatrix}$$

[1] Otherwise, we can choose: $\epsilon_i = \rho^{|x_i - x_{i-1}|}\epsilon_{i-1} + n_i$.

The only parameter which must be estimated is ρ and we estimate it by:

$$\hat{\rho} = \frac{\frac{1}{N-1}\sum_{i=2}^{N}\hat{\epsilon}_i\hat{\epsilon}_{i-1}}{\frac{1}{N}\sum_{i=1}^{N}\hat{\epsilon_i}^2} \tag{8}$$

For a Gaussian noise, we accept $\rho = 0$ with the significance threshold α if [7]:

$$-t_{\alpha/2,N-2} < \frac{\rho\sqrt{N-2}}{\sqrt{1-\rho^2}} < t_{\alpha/2,N-2} \tag{9}$$

where $t_{\alpha,p}$ is such that $Prob(T_p > t_{\alpha,p}) = \alpha$, T_p being the Student variable with p degrees of freedom. If ρ can not be considered as null, we use the GLS version of backpropagation algorithm which minimizes the modified cost function (4):

$$\Delta\mathbf{W} = -2\mu(1-\hat{\rho}^2)\hat{\epsilon}_1\frac{\partial f(x_1)}{\partial\mathbf{W}} - 2\mu\sum_{i=2}^{N}(\hat{\epsilon}_i - \hat{\rho}\hat{\epsilon}_{i-1})(\frac{\partial f(x_i)}{\partial\mathbf{W}} - \hat{\rho}\frac{\partial f(x_{i-1})}{\partial\mathbf{W}}) \tag{10}$$

3.2. k-order AR noise for dynamic systems and time series

In this case, we have:

$$\epsilon_i = \rho_1\epsilon_{i-1} + \rho_2\epsilon_{i-2} + \ldots + \rho_k\epsilon_{i-k} + n_i, \qquad n_i\ i.i.d. \tag{11}$$

Denoting $\hat{\rho}_0 = -1$ and using a variance homogenizing scheme similar to previous section, it can be shown that [8] the backpropagation algorithm becomes:

$$\Delta\mathbf{W} = -\mu\sum_{i=1}^{k}\sum_{j=1}^{k}\mathrm{b_{ij}}[\hat{\epsilon}_\mathrm{i}\frac{\partial\mathrm{f(x_j)}}{\partial\mathbf{W}} + \hat{\epsilon}_\mathrm{j}\frac{\partial\mathrm{f(x_i)}}{\partial\mathbf{W}}] - 2\mu\sum_{\mathrm{i=k+1}}^{\mathrm{N}}(\sum_{\mathrm{l=0}}^{\mathrm{k}} -\hat{\rho}_\mathrm{l}\hat{\epsilon}_\mathrm{i-l})(\sum_{\mathrm{l=0}}^{\mathrm{k}} -\hat{\rho}_\mathrm{l}\frac{\partial\mathrm{f(x_{i-l})}}{\partial\mathbf{W}}) \tag{12}$$

where: $b_{i,i+l} = \sum_{j=0}^{i-1}\hat{\rho}_j\hat{\rho}_{j+l} - \sum_{j=k+1-i-l}^{k-l}\hat{\rho}_j\hat{\rho}_{j+l}$, $\quad (i = 1,\ldots,k \quad l = 0,1,\ldots,k-i)$ and it can be verified that: $b_{ij} = b_{ji}$ and $b_{ij} = b_{k+1-j,k+1-i}$. To estimate $\hat{\boldsymbol{\rho}} = (\hat{\rho_1},\hat{\rho_2},\ldots,\hat{\rho_k})^T$, denoting $\hat{\boldsymbol{\gamma}} = (\hat{\gamma_1},\hat{\gamma_2},\ldots,\hat{\gamma_k})^T$ and $\hat{\gamma_k} = \frac{1}{N-k}\sum_{i=1}^{N-k}\hat{\epsilon}_i\hat{\epsilon}_{i+k}$, we can use the Yule-Walker equations derived from (11):

$$\hat{\boldsymbol{\rho}} = \hat{\boldsymbol{\Gamma}}^{-1}\hat{\boldsymbol{\gamma}} \quad , \quad \hat{\boldsymbol{\Gamma}} = \begin{pmatrix} \hat{\gamma_0} & \hat{\gamma_1} & \cdots & \hat{\gamma_{k-1}} \\ \hat{\gamma_1} & \hat{\gamma_0} & \cdots & \hat{\gamma_{k-2}} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\gamma_{k-1}} & \hat{\gamma_{k-2}} & \cdots & \hat{\gamma_0} \end{pmatrix} \tag{13}$$

4. Simulation results

In the first experiment, we try to estimate the function $y = 0.5sin(3x)$, $x \in [-0.5, 0.5]$, from 100 samples corrupted by a first order AR noise, using an MLP with a single neuron in the hidden layer. Fig.1.a illustrates the results of OLS and GLS approximation for $\rho = 0.9$. The test error for two estimators, averaged

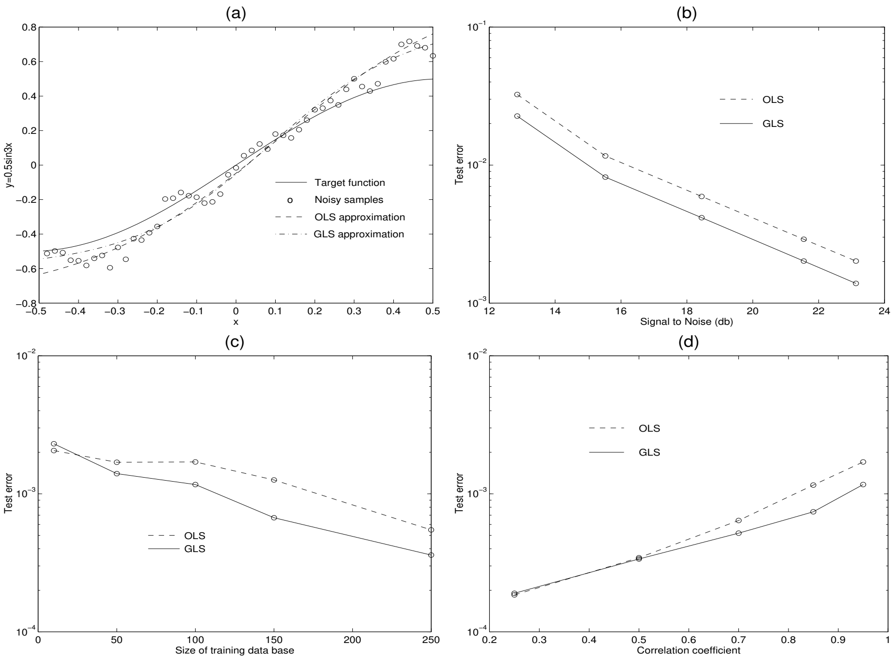

Figure 1: (a) OLS and GLS approximations for a first order AR noise. Test error as a function of (b) S/N, (c) size of training base, (d) correlation coefficient.

over 5 experiments, is given as a function of signal to noise (Fig.1.b), the size of training base (Fig.1.c) and ρ (Fig.1.d). As can be seen, the generalizing capacity of GLS estimator is nearly always better especially for the great values of ρ and the large training size. The bad quality of GLS for the small training size is partly due to poor approximation of $\hat{\rho}$. In this case, the first OLS estimator *memorizes* the noisy data so that the residue does not represent the noise. In fact, in all of our experiments, the training error of OLS estimator is less than the GLS one because the former has a tendency to learn the noise structure. In Fig.2, OLS and GLS approximation from 100 samples of the function $0.7sin(\pi x)cos(2\pi x)$, $x \in [0, 2]$ corrupted by a second order AR noise with $\rho_1 = 0.55$ and $\rho_2 = 0.4$ is given where we used a one hidden layer MLP with 10 neurons in the hidden layer. For the sake of clarity, only 25 noisy samples are shown in the figure.

5. Conclusion

In this paper, we derive generalized least square (GLS) versions of the back-propagation algorithm which provide consistent estimations in the case of data corrupted by colored noise based on autoregressive models. The comparison

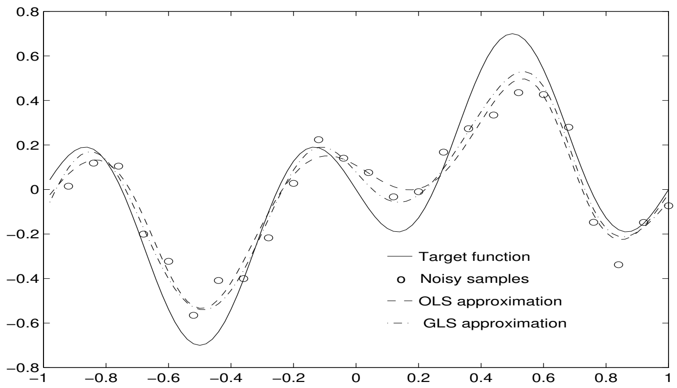

Figure 2: OLS and GLS approximation for a 2^{nd} order AR noise.

with ordinary least square (OLS) algorithms points out interesting improvements, even with small data samples. In practice, we first use OLS algorithm, which provides a good initialization of GLS algorithm, which then converges very fast. Consequently, the complete algorithm complexity is very close to OLS complexity.

References

[1] A. Antoniadis, J. Berruyer and R. Carmona. *Régression Non Linéaire at Applications*. Economica, 1992.

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

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

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

[5] L. Ljung. *System Identification, Theory for the User*. Prentice Hall, Englewood Cliffs, 1987.

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

[7] G. Baillargeon. *Méthodes Statistiques de l'Ingénieur*. volume 1. Les éditions SMG, 1994.

[8] S. Hosseini. Etude des Algorithmes Constructifs en Présence des Bruits non i.i.d. Technical Report, Laboratory TIRF, INPG, Grenoble, France, 1997.