Colored Noise in Dynamic Systems : Unbiased Modeling by Constructive Neural Networks

Shahram Hosseini and Christian Jutten
LIS, INPG, 46 Av. Félix Viallet, 38031 Grenoble, France
Email : hosseini or chris@lis-viallet.inpg.fr

Abstract

In this paper, we propose an algorithm to model dynamic systems using constructive feedforward neural networks when the system output is corrupted with additive colored noise. The algorithm tries to whiten the approximation residue during the network construction. The whitened residue is also used to stop the algorithm and to avoid overfitting.

1. Introduction

In the recent years, the dynamic system identification using MultiLayer Percepterons (MLP) has been considered by many researchers [1], [2]. The universal approximation property of MLPs [3] is a great reason of this interest, which guarantees that under mild conditions, all the nonlinear systems can be modeled by a single hidden layer MLP. Unfortunately, the existent theorems can not determine *a priori* the optimal number of neurons necessary in the hidden layer for a given problem. Many approaches have been proposed to determine dynamically the network size during training and constructive algorithms are among the most promising [4]. These algorithms grow dynamically the size of the hidden layer until a satisfying performance is achieved *i.e.* a stopping criterion is satisfied. One of the methods for stopping the algorithms is to use the residual signal properties [5]. If we suppose that the system noise is i.i.d., a correlation test on the residue can determine if it is reduced to noise or not. If it is not the case, we can imagine that the residue includes still a part of the signal and the construction procedure should be continued.

When the system noise is not i.i.d., the problem is more complicated and we have to « whiten » the residue. The whitening procedure is also interesting because it helps to improve the estimation [6]. In [7], using a maximum likelihood framework, we have proposed a method to improve the estimation and to find a stopping criterion for the constructive MLPs used for the function approximation. In this paper, we try to adapt that idea for the dynamic system identification. In fact, in dynamic systems, the colored noise is more dangerous because it may lead to a highly biased estimation. Although the algorithm presented in this paper can be used for many constructive methods, we used it for a novel constructive approach which we developed recently [8].

2. Colored noise in dynamic systems

The presence of colored noise in dynamic systems has been considered for a long time and for linear systems, many methods have been proposed to obtain an unbiased estimation. However, this problem is rarely studied in the neural networks literature. In the following we recall briefly the problem. At first, we consider the case of an additive white output noise in a multi-input, single-output system :

$$y(t) = f(\mathbf{u}^{t-1}, y^{t-1}, \boldsymbol{\theta}^*) + e(t) \tag{1}$$

where $\mathbf{u}^{t-1} = [\mathbf{u}(t-1), \mathbf{u}(t-2), \dots]$ and $y^{t-1} = [y(t-1), y(t-2), \dots]$ are respectively past observations of inputs and output, e(t) is an i.i.d. zero mean noise and $\boldsymbol{\theta}^*$ is the model parameter vector. Using universal approximation property of MLPs, it can be shown that the Ordinary Least Square (OLS) method gives an unbiased consistent estimation of parameters. When the output noise is not i.i.d., the OLS method leads to a biased estimation. Suppose data are generated following this model :

$$y(t) = f(\mathbf{u}^{t-1}, y^{t-1}, \boldsymbol{\theta}^*) + g(e(t), e^{t-1}) \tag{2}$$

where $e^{t-1} = [e(t-1), e(t-2), \dots]$ represents the noise past values. The OLS estimator tries to minimize this cost function versus $\boldsymbol{\theta}$:

$$J = E\left\{(\hat{y}(t) - y(t))^2\right\} = E\left\{(f(\mathbf{u}^{t-1}, y^{t-1}, \boldsymbol{\theta}) - y(t))^2\right\} \quad (3)$$

Evaluating the derivative of J versus $\boldsymbol{\theta}$ at $\boldsymbol{\theta}^*$, we obtain :

$$\left.\frac{\partial J}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} = E\left\{-2g(e(t), e^{t-1}) \left.\frac{\partial f(\mathbf{u}^{t-1}, y^{t-1}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}\right\} \quad (4)$$

Since y^{t-1} is a function of e^{t-1}, one denotes $\left.\frac{\partial f(\mathbf{u}^{t-1}, y^{t-1}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} = h(e^{t-1})$, and we can write :

$$\left.\frac{\partial J}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} = -2E\left\{g(e(t), e^{t-1})h(e^{t-1})\right\} \quad (5)$$

As both g(.) and h(.) contain the noise past values, the mathematical expected value (and so the derivative) is in general non-null at the optimum value of $\boldsymbol{\theta}$ (*i.e.* $\boldsymbol{\theta}^*$) and we have a biased estimator. On the other hand, it is clear that if $\hat{y}$ is determined so that $y(t) - \hat{y}(t) = g_1(e(t))$, we obtain :

$$\left.\frac{\partial J}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} = -2E\left\{g_1(e(t))h(e^{t-1})\right\} = 0 \quad (6)$$

and the estimator will be unbiased. Thus, the solution consists in separating $g(e(t), e^{t-1})$ in two parts :

$$g(e(t), e^{t-1}) = g_1(e(t)) + g_2(e^{t-1}, \boldsymbol{\rho}) \quad (7)$$

and to estimate the noise parameters $\boldsymbol{\rho}$ together with the system parameters $\boldsymbol{\theta}$. In this case, it is necessary to use the noise past values as the accessory regressors together with the past observations of input and output signals. In practice, the noise is unknown and in an adaptive approach, it is replaced by the residue of estimation.

The choice of the regressors and their combinations is very important in the nonlinear models [9]. A rather general example can be the NARMAX model [10] :

$$y(t) = f(\mathbf{u}^{t-1}, y^{t-1}, e^{t-1}, \boldsymbol{\theta}) + e(t) \quad (8)$$

In the concept of neural networks, it means that the neural network inputs include not only the known training data but also the output approximation error and the network has a recurrent structure. As [9] mentions, it is in general harder to work with recurrent structures. Among other things, it becomes difficult to check under what conditions the obtained model is stable, and it takes an extra effort to calculate gradients for model parameters estimation.

A more interesting method which we propose is to use a nonlinear model for system (MLP) and a linear model for noise. In this case, we have :

$$y(t) = f(\mathbf{u}^{t-1}, y^{t-1}, \boldsymbol{\theta}) + \varepsilon(t) \quad \text{where } \varepsilon(t) = H(q)e(t) \quad (9)$$

Therefore, we have not the difficulties arising the recurrent networks and using the parametric model for noise, we will be able to whiten the estimation residue using its past values. As we showed in [7], the choice of an autoregressive structure for noise is realistic and simplifies considerably the computations. We will continue our discussion with this hypothesis.

3. Maximum likelihood estimation of system and noise parameters

We suppose henceforth that the dynamic system may be modeled by :

$$y(t) = f(\mathbf{u}^{t-1}, y^{t-1}, \boldsymbol{\theta}) + \varepsilon(t) \quad (10)$$

$$\text{where } \varepsilon(t) = e(t) + \sum_{j=1}^{k} \rho_j \varepsilon(t-j) \quad (11)$$

and e(t) is an i.i.d. Gaussian noise: e(t) ~ $N(0, \sigma_n^2)$. Supposing there are N training samples and using the vectorial notations $\mathbf{e} = [e_1, e_2, \ldots, e_N]$ and $\boldsymbol{\varepsilon} = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N]$, The covariance matrix of $\boldsymbol{\varepsilon}$ is : $\mathbf{D}_\varepsilon = E\left\{\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T\right\} = \sigma_n^2 \mathbf{V}$, and the log-likelihood function :

$$L_\varepsilon(\boldsymbol{\varepsilon}) = \frac{-N\log(\sigma_n^2)}{2} + \frac{\log\left|\mathbf{V}^{-1}\right|}{2} - \frac{\boldsymbol{\varepsilon}^T \mathbf{V}^{-1} \boldsymbol{\varepsilon}}{2\sigma_n^2} \quad (12)$$

For an autoregressive stationary Gaussian noise, it can be shown [7] that $L_\varepsilon(\boldsymbol{\varepsilon})$ is equal to :

$$L_\varepsilon(\boldsymbol{\varepsilon}) = \frac{-N\log(\sigma_n^2)}{2} + \frac{\log\left|\mathbf{R}^{\mathbf{k}T}\mathbf{R}^{\mathbf{k}}\right|}{2} - \frac{\boldsymbol{\varepsilon}^{\mathbf{k}T}\mathbf{R}^{\mathbf{k}T}\mathbf{R}^{\mathbf{k}}\boldsymbol{\varepsilon}^{\mathbf{k}} + \sum_{i=k+1}^{N}(\varepsilon_i - \sum_{j=1}^{k}\rho_j\varepsilon_{i-j})^2}{2\sigma_n^2} \quad (13)$$

where $\boldsymbol{\varepsilon}^{\mathbf{K}}$ contains the K first samples of $\boldsymbol{\varepsilon}$, and $\mathbf{R}^{\mathbf{k}}$ is a matrix relating $\boldsymbol{\varepsilon}^{\mathbf{K}}$ and the K first samples of $\mathbf{e}$, $\mathbf{e}^{\mathbf{k}}$, by : $\mathbf{e}^{\mathbf{k}} = \mathbf{R}^{\mathbf{k}}\boldsymbol{\varepsilon}^{\mathbf{K}}$ and N is the size of the training set. Using the stationary properties of $\mathbf{e}$ and $\boldsymbol{\varepsilon}$, $\mathbf{R}^{\mathbf{k}}$ can be computed as a function of σ_n^2 and ρ_i [7]. Then, the noise parameters (σ_n^2 et ρ_i) and the network weights ($\mathbf{W}$) can be estimated by maximizing $L_\xi(\xi)$ where $\xi_i = \xi(t) = y(t) - f(\mathbf{u}^{t-1}, y^{t-1}, \mathbf{W})$ is the network estimation residue.

Note that if we neglect the K first samples of data, the log-likelihood function for the N-K last samples of training set will be :

$$L_{2_\xi}(\xi) = \frac{-N\log(\sigma_n^2)}{2} - \frac{\sum_{i=k+1}^{N}(\xi_i - \sum_{j=1}^{k}\rho_j\xi_{i-j})^2}{2\sigma_n^2} \quad (14)$$

Moreover, as σ_n^2 can be estimated *a posteriori*, we can minimize the following simple cost function :

$$L_{3_\xi}(\xi) = \sum_{i=k+1}^{N}(\xi_i - \sum_{j=1}^{k}\rho_j\xi_{i-j})^2 \quad (15)$$

The following remarks should be considered :

1) The stability condition imposes that all the roots of the characteristic equation of noise filter (11) are inside the unit circle. This imposes constraints on the acceptable values of ρ_i. This problem is more important when we use the more general formula (13), because in this case, the out-of-range values of ρ_i lead to logarithm of a negative value and the algorithm fails. For example, for a first order autoregressive model, ρ_1 must be between -1 and 1. So, we have to use a kind of optimization with constraint or make a variable change like $\sin(\alpha) = \rho_1$, α being the new noise parameter.

2) In general, the gradient values for the signal parameters and the noise parameters are not in the same order of magnitude. Hence, choosing the same learning rate for all the parameters may slow down the algorithm or lead to divergence. The best remedy is to use an adaptive learning rate algorithm.

4. Constructive algorithm

Although the mentioned algorithm can be used with the fixed size MLPs, it is more interesting when we use a constructive approach. In this case, the whitened residue may be tested to determine if the construction procedure must be stopped or not. The main idea is that the network growing should be continued until residue reduces to noise. As in our model the noise parameters are estimated simultaneously with the network weights, we can use them to construct the inverse filter and to whiten the residue. Then, a correlation hypothesis test on the residue determines if it can be considered as white noise or not.

Although the idea may be used for nearly all the constructive algorithms, we made the experiments with a version of a special algorithm which we developed recently [8]. The algorithm, called « Accessory networks based freezing algorithm », constructs the main network by adding small and variable size accessory networks. After constructing and training each accessory network, its input weights are frozen and its output is connected to the main network output. Then in order that the network output remains orthogonal to the subspace spanned by the different accessory networks, the weights of the output layer will be updated and the final residue is used to verify the stopping criterion. The algorithm can be summarized as following :

```
BEGIN
main-network-size = 0 ;
target = output-data ;
DO {
  new-accessory-network-size = 0 ;
  DO {
    new-accessory-network-size ++ ;
    train accessory network to estimate target using(13) ;
  } WHILE ((residue-power/target-power)>TH1)
  connect the accessory network to the main network ;
  main-network-size += accessory-network-size ;
  Adapt the output layer weights using (13);
  target = new residue ;
} WHILE (processed residue is not white) ;
END
```

5. Simulation results

The simulation example which we used was a generalization of the example introduced by [11] with colored noise :

$$y(t) = y(t-1)/(1+y^2(t-1)) + u^3(t-1) + \varepsilon(t)$$
$$\varepsilon(t) = 0.9\varepsilon(t-1) + e(t) \quad (16)$$

where e(t) was a zero mean Gaussian white sequence with variance $\sigma_e^2 = 0.04$ and the system input u(t) was an independent sequence of uniform distribution with mean zero and variance $\sigma_u^2 = 0.3$. The MLP inputs were u(t-1) and y(t-1) and its output was y(t). Using the constructive algorithm mentioned above, an MLP was constructed and trained. The experiment consisted of 10 runs, each run corresponding to a different noise seed value. To evaluate the results, for each run, another network whose size is the final size obtained by our algorithm is trained using the Ordinary Least Square (OLS) algorithm. The results are illustrated in the Table 1. The test data are generated with a noiseless system and using a different input.

	ML		OLS	
	mean	STD	mean	STD
Training error	99.8	23.8	49.6	8.7
Test error	3.5	4.0	20.3	5.9
Network size	3.8	0.8	3.8	0.8
Noise parameter (ρ)	0.89	0.03		

Table 1 : The simulation results for our algorithm (ML) and Ordinary Least Square (OLS) algorithm.

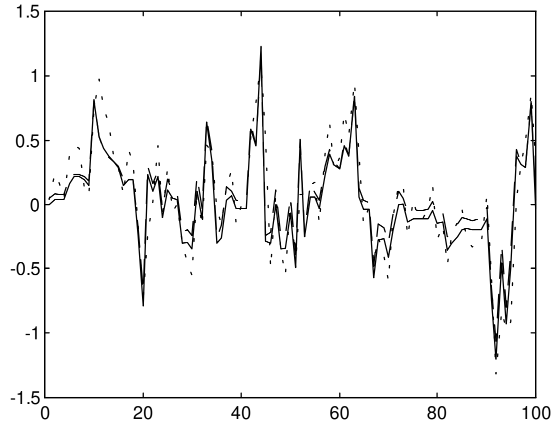

Figure 1: Target test signal (solid line), ML estimation (dashed line) and OLS estimation (dotted line).

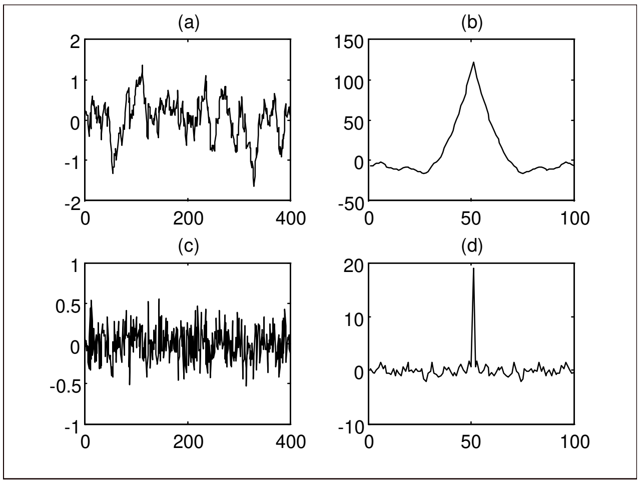

Figure 2: a) System colored noise b) its correlation function c) whitened residue d) its correlation function.

As can be seen, the networks trained by our algorithm have a much better performance in generalization, thanks to take into account the noise properties. However, the training error of OLS-based networks is less than that of our networks. It is not surprising because the OLS method has a tendency to memorize the data, which leads to a good performance on the training data but a weak generalization capacity. Note also the estimated noise parameter value, ρ, is very close to its actual value, 0.9.

Figure 1 illustrates the test output signal with its estimations using OLS and our algorithm. The training output noise, ε(t), and the final whitened residue of our algorithm are shown in Figure 2 with their autocorrelation functions.

6. Conclusion

An algorithm is presented to model the dynamic systems perturbed by colored noise using feedforward neural networks. With an autoregressive Gaussian noise hypothesis, a maximum likelihood approach is used to estimate simultaneously the noise parameters and the neural network weights. A new freezing algorithm is used to construct dynamically the network. The estimated noise parameters are used to whiten the residue which can be used as a stopping criterion for construction procedure. The simulations show that the generalization performance of the network constructed by this algorithm is much better than that of the network trained by ordinary least square method.

References

[1] S. Chen & S.A. Billings, *Neural networks for nonlinear dynamic system modeling and identification* , Int. J. Control, vol. 56, no. 2, pp. 319-346, 1992.

[2] J. Sjöberg, H. Hjalmarsson & L. Ljung, *Neural networks in system identification* , 10th IFAC symp. On system identification, Copenhagen, vol. 2, pp. 49-72, 1994.

[3] K. Hornik, M. Stinchcombe & H. White, *Multilayer feedforward networks are universal approximators*, Neural Networks, vol. 2, pp. 359-366, 1989.

[4] T.Y. Kwok & 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.

[5] Ch. Jutten & R. Chentouf, *A new scheme for incremental learning*, Neural Processing Letters, vol. 2, no. 1, pp. 1-4, 1995.

[6] Sh. Hosseini & Ch. Jutten, *Improving neural networks estimation in presence of non i.i.d. noise*, in proceedings of ESANN'98, Bruges, pp. 327-332, 1998.

[7] Sh. Hosseini & Ch. Jutten, *Simultaneous estimation of signal and noise in constructive neural networks*, in proceedings of NC'98, Vienna, pp. 412-417, 1998.

[8] Sh. Hosseini & Ch. Jutten, *Weight freezing in constructive neural networks : a novel approach*, to be presented in IWANN'99, Alicante, 1999.

[9] J. Sjöberg et *al.*, *Nonlinear Black-box modeling in system identification : a unified overview*, Automatica, vol. 31, no. 12, pp. 1691-1724. 1995.

[10] I.J. Leontaritis & S.A. Billings, *Input-output parametric models for non-linear systems*, Int. J. Control, vol. 41, pp. 303-344, 1985.

[11] K.S. Narendra & K. Parthasarathy, *Identification and control of dynamical systems using neural networks*, IEEE Trans. On Neural Networks, vol. 1, pp. 4-27, 1990.