Pergamon

Computers chem. Engng Vol. 20, Suppl., pp. S1089–S1094, 1996

S0098-1354(96)00189-5
0098-1354/96 $15.00+0.00

SELF-CONSISTENCY IN NEURAL NETWORK-BASED NLPC ANALYSIS WITH APPLICATIONS TO TIME-SERIES PROCESSING

R. RICO-MARTINEZ[1], J. S. ANDERSON[2] and I. G. KEVREKIDIS[2]

[1] Instituto Tecnológico de Celaya, Depto. de Ingeniería Química, Celaya, Gto. 38010 México
[2] Department of Chemical Engineering, Princeton University, Princeton, N. J. 08544 U.S.A.

Abstract.- The success of nonlinear system identification and characterization from experimental time-series often depends on the appropriate pre-processing of the data. This pre-processing can, in many instances, be achieved through linear and/or "nonlinear" principal component analysis. In recent years, neural network based techniques as a means to perform the nonlinear principal component (NLPC) analysis have gained increased attention. In this contribution, we address an inherent shortcoming of these techniques: the *self-consistency* problem. We present a modification to the usual feedforward neural network architecture used to extract the NLPCs that results in attenuation of this problem. The proposed modification may significantly reduce representation errors in many other applications for which NLPC analysis is a powerful tool, such as feature extraction and image processing.

INTRODUCTION

The processing of experimental time-series using principal component analysis (linear or nonlinear) has become a powerful tool towards the understanding of the dynamics of nonlinear systems and the ensuing construction of empirical or semi-empirical models for prediction and control purposes (Broomhead and King (1986), Gibson *et al.* (1992), Weigend and Gershenfeld (1993), Kung and Diamantaras (1991), Oja (1992)). Autoassociative neural networks have been proposed as a means to perform the "nonlinear principal component analysis" (NLPCA) (Boulard and Kamp (1988), Demers (1992), Kramer (1991), Rico-Martínez *et al.* (1992), Saund (1989), Usui *et al.* (1990)). Through the use of NLPC neural networks, significant compression and filtering of the data can be achieved. Furthermore, the inherent ability of such networks to capture nonlinear correlations makes the NLPC analysis superior, in principle, to its linear counterpart. Consider, as an illustration, a one-parameter curve on a plane embedded in $\mathcal{R}^3$ (see Fig. 1). By projecting this curve on its two *linear* principal components, spanning the plane, the dimensionality of the data can be reduced to two (the coordinates (a_1, a_2) identify every point on the curve). By using NLPCs (e.g. by using an arc-length-like measure, λ) the trajectory may be parametrized by a single quantity as illustrated in Fig. 1.

LPC analysis as a filtering or compression procedure (reducing noise -and dimensionality- by neglecting low-energy, "high-mode" components of the data) has a built-in self-consistency property which is lacking in the NLPC case: Iterating the process (*i.e.* projecting the filtered data back on the LPC vectors) obviously gives the same result.

NLPCs are also used for filtering or compression (dimensionality reduction). The usual procedure for the NLPC case involves feeding the available data points as inputs to a feedforward neural network with a linear "bottleneck" layer in the middle (Fig 4(a)). The network is trained to learn the identity mapping, i.e to approximate its inputs (autoassociation). In the often-used "simultaneous" NLPC training, where the number of bottleneck neurons (number of "significant" NLPC modes) is postulated *a priori*, the output of the network constitutes (upon convergence) the filtered data. However, since the identity mapping is only approximated, when the filtering process is repeated (i.e., when the output is processed through the same NLPC network) the data will not in general map to themselves. In that sense, NLPCs are *non-self-consistent.*

We are concerned, in this paper, with "rectifying" this conceptual shortcoming of the NLPC procedure by incorporating a measure of self-consistency as part of the objective function during training. This modification consists of incorporating recurrent connections to the usual feedforward NLPC network, which now becomes an Elman-type network (Elman (1990)) (see Fig. 5). We discuss issues arising in the training of such networks, and demonstrate that our procedure results in attenuation of the problem using time-series data from a nonlinear dynamical system. The resulting reduction of the representation error could prove a significant enhancement for time-series processing, as well as for other common applications of principal component analysis such as feature extraction and image processing. We also briefly discuss another type of self-consistency problem arising in phase-space reconstruction from filtered time-series data.

SELF-CONSISTENCY IN PC ANALYSIS

In order to illustrate the self-consistency inherent in the LPC analysis and its absence in the standard neural network-based NLPC analysis we will make use of time-series from the well-known Lorenz system:

$$\dot{x} = -\sigma x + \sigma y; \quad \dot{y} = -xz + rx - y; \quad \dot{z} = xy - bz.$$

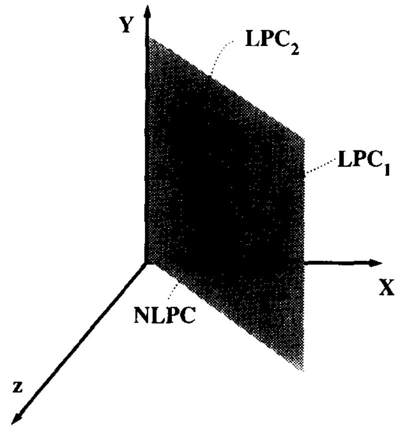

Figure 1: Principal component analysis used to describe a curve on a plane embedded in $\mathcal{R}^3$. The curve can be parametrized in $\mathcal{R}^2$ by projecting on its linear principal components. Nonlinear principal components (e.g. an arc-length-like measure on the curve), further reduce the dimensionality of the system to one.

Our illustrations will use trajectories along a periodic attractor (double-loop limit cycle) obtained at $\sigma = 10$, $b = 8/3$ and $r = 220$.

Time-series of all the state variables were recorded by numerically integrating these equations using a robust integrator with error control (Leis and Kramer (1988)) and a sampling interval of 0.005 units of time (roughly 1/200 of the period of oscillation). For principal component analysis, the data were divided in time-series segments of the variable z spanning 0.01 time units (i.e. each segment consists of 20 consecutive sampling instances of the variable z). For a discussion of the effect of the time-series vector length on the results of the principal component analysis see Broomhead and King (1986). A total of 2000 time-series segments were used for the illustrations presented below. The segments were selected at random from the converged trajectory on the attractor (Fig 2(a)).

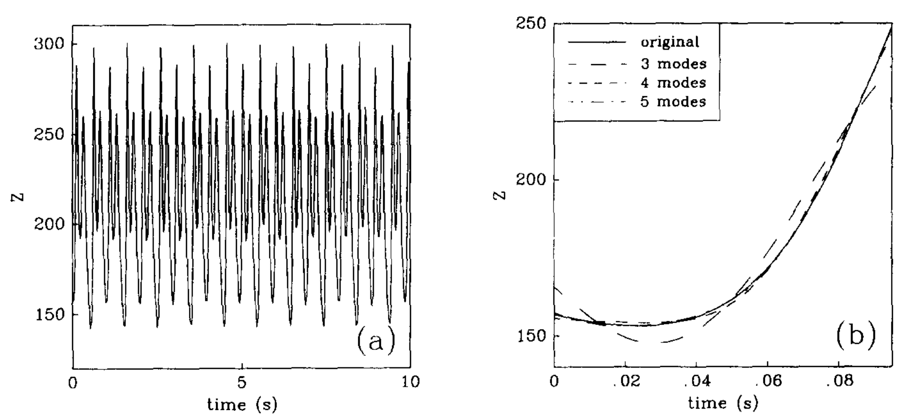

Figure 2: (a) The periodic trajectory of the Lorenz system used for the illustrations. Parameter values are given in the text. (b) Comparison of a time-series segment of the periodic trajectory of the Lorenz system with its reconstructed version using 3, 4 and 5 linear principal components ("modes"). At least 4 LPCs are needed for a reasonable reproduction of the trajectory.

Figure 2(b) shows a comparison of one of the original time-series segments of the Lorenz system and the corresponding reconstruction using 3, 4 and 5 linear principal components. The LPCs are found by performing a singular value decomposition (SVD) on the "trajectory" matrix (**T**) whose rows are the time-series segments in vector form. The SVD allows us to write **T** as the product of three matrices: $\mathbf{T} = \mathbf{U}\Lambda\mathbf{V}^T$. **U** and **V** contain as rows the left and right singular vectors of **T**, and Λ is a diagonal matrix containing the singular values of **T**. The LPC reconstruction of **T** is then obtained by projecting it on the space spanned by its right singular vectors. By neglecting those directions associated to small singular values (the singular values give a measure of the "energy" contained by each mode) the LPCA is attained. From Fig. 2(b) is obvious that at least four linear principal components should be used to reproduce the original time-series

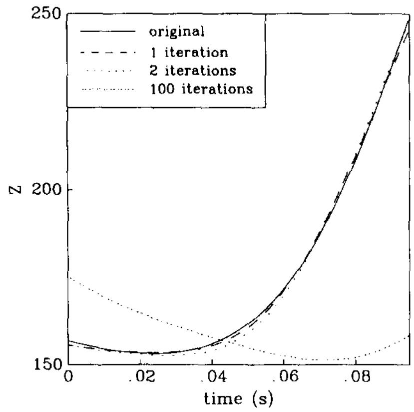

Figure 3: Comparison of a time-series segment of the periodic trajectory of the Lorenz system with its reconstructed version using 3 NLPCs. The non-self-consistent nature of the procedure is illustrated by iterating the network. If the reconstructed time-series segment is processed once more (2 iterations), or until convergence (100 iterations) through the NLPC network, one obtains different representations.

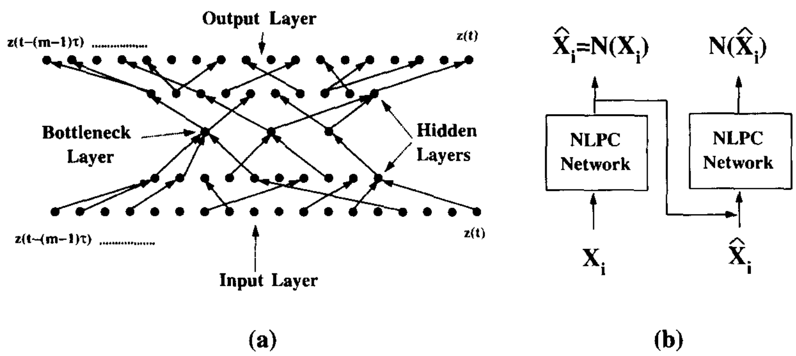

Figure 4: (a) Schematic of the standard NLPC neural network architecture. All layers are fully interconnected although only a few connections are depicted. (b)A doubly evaluated NLPC network for the problem of self-consistency.

even though the minimal dimension of the phase-space in which such trajectory may live is three (which is also the dimension of the original system).

The standard neural network-based NLPCA, although successful in the processing of time-series segments with a minimal number of "modes" (see below), lacks the inherent self-consistency of the LPCA. Figure 3 shows the results of performing the NLPCA on the same set of time-series segments. The NLPC network used for this illustration consisted of 10 nonlinear neurons in each hidden layer, 3 neurons in the bottleneck layer (i.e. three nonlinear principal components are sought) and 20 neuron input and output layers (according to the length of the time-series segments being processed). The network was trained using a conjugate gradient (CG) algorithm and convergence was achieved after 15 complete CG cycles. The NLPC analysis allows an accurate reconstruction of the time-series segment with only 3 modes; however, by iterating the NLPC mapping, we can attest to the lack of self-consistency of the procedure (Fig. 3).

PROPOSED ELMAN-NLPC ARCHITECTURE

We now present the underlying arguments for the proposed modification of the NLPC neural network architecture aimed to improve the self-consistent properties of the NLPC procedure. The training of the NLPC neural network can be expressed as the following optimization problem:

$$min \sum_i ||\hat{x}_i - x_i||^2 \quad such \quad that \qquad \hat{x}_i = N_{m,h,b,h,m}(x_i), \quad \forall i \tag{1}$$

where now $N_{m,h,b,h,m}$ represents the five-layer NLPC network, m is the length of the input (and output) vector, h is the number of neurons in the (nonlinear) hidden layers and b is the number of bottleneck neurons. The variable $\hat{x}$ is the reconstructed time-series segment after the NLPC preprocessing and x is the original time-series segment.

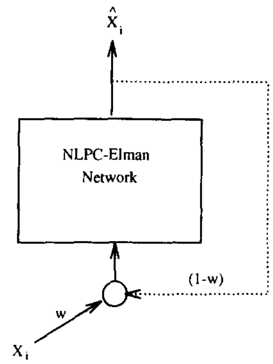

Figure 5: Elman-type network for self-consistent extraction of NLPC components. W is a weighting factor.

If the NLPC procedure is self-consistent, it follows that:

$$\hat{x}_i = N_{m,h,b,h,m}(\hat{x}_i), \quad \forall i \tag{2}$$

One might attempt to include this set of equations as constraints for the optimization problem that constitutes the NLPC network training; however, the resulting problem (a nonlinear optimization with equality constraints) is very difficult and may not have feasible solutions. Furthermore, note that this equation *cannot* (for a general nonlinear case) be satisfied exactly because it implies the existence of a fixed point of the NLPC mapping for each time-series segment processed ($\hat{x}$), and thus $N_{m,h,b,h,m}(.)$ will need to have an *infinite* number of fixed points if a chaotic time-series is being processed. As an alternative to this formulation, consider the following modified objective function:

$$min(\sum_i P||\hat{x}_i - x_i||^2 + \sum_i ||\hat{x}_i - N_{m,h,b,h,m}(\hat{x}_i)||^2) \tag{3}$$

such that $\hat{x}_i = N_{m,h,b,h,m}(x_i)$, for all i, as before, and P is a weighting factor.

By using this modified objective function the NLPC network will be trained not only by considering how well it fits the data, but also by taking into account the self-consistent properties of the post-processed time-series segments. An analogous modification to improve self-consistency was proposed, in the context of local linear approximations, by Kostelich and Yorke (1990).

Using Eq. 3 as the basis to develop the NLPC network training requires the evaluation of the network with two different sets of inputs (the second being a function of the first) and two evaluations of the network per iteration per point in the training set (with a two-step calculation of derivatives). The required "iterated" function evaluation is analogous to the previously proposed procedure for the construction of continuous neural network approximations based on explicit numerical integrators (Rico-Martínez *et al.* (1992)). Figure 4(b) schematically depicts the needed configuration with repeated evaluations of the NLPC network.

This "double evaluation" can be incorporated into a neural network configuration by using an Elman-type architecture with recurrent connections (Fig. 5). Thus solving Eq. 3 will be analogous to training an Elman-type network and the usual objective function:

$$min \sum_i ||\hat{x}_i - x_i||^2 \tag{4}$$

such that $\hat{x}_i = N^{Elman}_{m,h,b,h,m}(x_i, \hat{x}_i)$, for all i. Note that, for this architecture, the weighting factor W is bounded in the interval [0,1]. $W = 1$ will give back the original NLPC network architecture while $W = 0$ will represent a totally self-consistent (but useless) network.

RESULTS AND CONCLUSIONS

The proposed architecture, built around the same configuration as the previously used standard NLPC network, is tested with the periodic data from the Lorenz system. Training was performed using the standard recurrent backpropagation algorithm (Pineda (1987)) coupled with a conjugate gradient subroutine with frequent restarts to speed-up the process. Convergence was achieved after approximately 5000 network parameter updates. Figure 6 shows a comparison of the reproduction of one of the time series segments of the periodic trajectory used for our illustration. These results were obtained setting the value of the weighting factor W equal to 0.5. The network retains the accurate reproduction of the segments with only 3 NLPCs. Furthermore, the self-consistency of the procedure appears to be improved: repeated iteration of

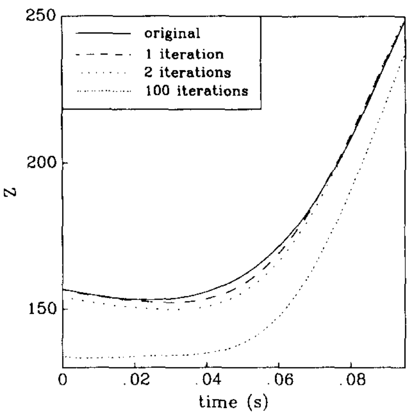

Figure 6: Comparison of a time-series segment of the periodic trajectory of the Lorenz system against its reconstructed version using the proposed Elman-type architecture with 3 bottleneck neurons. The non-self-consistent nature of the procedure is illustrated by iterating the network. The procedure still lacks total self-consistency (second iterate) but its self-inconsistent nature is considerably diminished: note that the hundredth iterate still resembles the initial shape of the original segment.

the time-series segment through the Elman-NLPC network shows that the procedure still lacks total self-consistency, however the hundredth iterate of the segment still resembles the original. Statistics over a large number of time-series segments shows marginal improvement in the self-consistency nature of the proposed procedure (the difference between an input in the vicinity of the periodic trajectory and its iterate for a single pass grows slower for the proposed architecture).

The training of this Elman-type network exhibits several difficulties when performed with the usual recurrent backpropagation algorithm. The results presented in Fig. 6 were obtained for a run with a final error 2.5 times larger than for its standard NLPC counterpart, thus the marginality of the improvement on the self-consistency of the results may be in part attributed to our lack of success in reducing this error measure even further. Currently we are investigating alternative methods to perform the training of these types of architectures in an attempt to minimize the effect of the sensitivity of the evolution rules introduced by the standard recurrent backpropagation algorithm (Anderson *et al.* (1995)). Our preliminary results seem to indicate that more "rigorous" methods (involving exact solution of the output of the network at each stage of training and exact calculation of derivatives), although more computationally expensive per iteration, may be a valid alternative for the overall training of these types of networks. We are also investigating the effect of the weighting factor W on the quality of the results and ease of training of the proposed neural network architecture.

In this contribution we have demonstrated the capabilities of Elman-type NLPC network architectures in an attempt to improve the self-consistent properties of the NLPC procedure based on neural networks. The self-consistency of the processed time-series is slightly improved. In addition to the self-consistency problem described here, the processing of time-series using principal component analysis exhibits another related problem when used for attractor reconstruction. In that application, consecutive time-series segments (windows) with overlapping intervals, or with common edges, constitute the inputs for the analysis. For these applications additional constraints that account for the "continuity" of the time trace (i.e. points on the overlapping interval of two consecutive segments must be the same, and in the absence of an overlapping interval the two consecutive segments must be "connected smoothly" at their common boundary) must be satisfied. These constraints (that may be seen as another self-consistency problem) cannot be enforced by any of the principal components techniques described here. Current efforts addressing this problem involve the use of time-lag neural network algorithms (Werbos (1988)) as well as modifications similar in spirit to the so-called τ-method used in spectral discretizations of differential equations: the components of the reconstructed segments in their "higher" NLPC modes are computed so as to enforce smoothness and continuity at the segment boundaries.

In addition to their non-self-consistent character, NLPC procedures based on neural networks present other drawbacks when compared against the LPC analysis for time series data. While for the LPC analysis a well-developed and established theory is available, NLPC procedures are based mostly on heuristic rules and very general properties (universal approximation) of neural networks. This is important particularly in applications related to phase-space reconstruction of attractors since the relationship between PC analysis and other reconstruction techniques (using time-delays or higher order derivatives) is well understood for

LPC analysis (Gibson *et al.* (1992)) but very poorly understood for NLPC analysis.

A fundamental problem in the analysis of experimental data concerns the correspondence between the dynamics of the system and the discretely measured time-series that comprises the information available to characterize and identify such systems. Often the success in the characterization of nonlinear systems relies upon the appropriate pre-processing of the data. Identification methodologies based on principal component techniques rely on a two-tier approach: principal component decomposition followed by characterization of the dynamical evolution of the system to be identified. A self-consistent pre-processing of the data would reduce representation errors resulting from the first stage of this procedure, thus improving the chances of successful characterization of the dynamical behavior of the system during the second stage of the identification effort. Performing the NLPC analysis with the proposed Elman-type neural network configuration, as a preliminary step towards the construction of empirical models, may prove a significant enhancement on the identification of nonlinear systems.

ACKNOWLEDGEMENTS

This work has been partially funded by the NSF, DuPont, ARPA/ONR, and CONACyT. The hospitality of the Center for Nonlinear Studies at Los Alamos National Laboratory is gratefully acknowledged.

REFERENCES

Anderson, J. S., I. G. Kevrekidis and R. Rico-Martínez, 1995, A comparison of recurrent network training algorithms for time series analysis and system identification, *Computers chem. Engng*, These proceedings.

Boulard, H., and Y. Kamp, 1988, Auto-association by multilayer perceptrons and singular value decomposition, *Biol. Cybern.*, **59**, 291–294.

Broomhead, D. S., and G. P. King, 1986, Extracting qualitative dynamics from experimental data, *Physica D*, **20**, 217–236.

DeMers, D., 1992, Dimensionality reduction for non-linear time series, *SPIE Vol. 1766, Neural and Stochastic Methods in Image and Signal Processing*, 200–211.

Elman, J. L., 1990, Finding structure in time, *Cognitive Science*, **14**, 179–211.

Gibson, J. F., J. D. Farmer, M. Casdagli and S. Eubank, 1992, An analytic approach to practical state space reconstruction, *Physica D*, **57**, 1–30.

Kostelich, E. J., and J. A. Yorke, 1990, Noise reduction: Finding the simplest dynamical system consistent with the data, *Physica D*, 41, 183–196.

Kramer, M. A., 1991, Nonlinear principal component analysis using autoassociative neural networks, *AIChE J.*, **37**, 233–243.

Kung, S. Y., and K. I. Diamantaras, 1991, Neural networks for extracting unsymmetric principal components, *Proc. 1991 IEEE Workshop in Neural Networks for Signal Processing*, Princeton, NJ, 50–59.

Leis, J. R., and M. A. Kramer, 1988, ODESSA - an ordinary differential equation solver with explicit simultaneous sensitivity analysis, *ACM Trans. Math. Software*, 14, 61–67.

Oja, E., 1992, Principal components, minor components and linear neural networks, *Neural Networks*, **5**, 927–935.

Pineda, F. J., 1987, Generalization of back-propagation to recurrent neural networks, *Phys. Rev. Letters*, **59**, 2229–2232.

Rico-Martínez, R., K. Krischer, I. G. Kevrekidis, M. C. Kube and J. L. Hudson, 1992, Discrete- vs. continuous-time nonlinear signal processing of Cu electrodissolution data, *Chem. Eng. Comm.*, **118**, 25–48.

Saund, E., 1989, Dimensionality-reduction using connectionist networks, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **11**, 305–314.

Usui, S., S. Nakauchi and M. Nakano, 1990, Reconstruction of munsell color space by a five-layered neural network, *Proc. 1990 IEEE Int. J. Conf. in Neural Networks*, **2**, 515–520.

Weigend, A. S., and N. A. Gershenfeld, 1993, *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley.

Werbos, P. J., 1988, Generalization of backpropagation with application to a recurrent gas market model, *Neural Networks*, **1**, 339–356.