

Noise Reduction in Chaotic Attractors

Nejib Smaoui

Department of Mathematics & Computer Science, Kuwait University

P.O. Box 5969 Safat 13060, Kuwait

Phone: 965-481-1188, ext. 5344, Fax: 965-481-7201

email: smaoui@sun470.sci.kuniv.edu.kw

ABSTRACT: A method, based on using Artificial Neural Network (ANN), is presented to reduce noise in chaotic attractors without knowing the underlying maps. The fractal dimension of the attractor is used as an indicator to determine the number of input layer nodes. For moderate levels of additive noise, the ANN method was able to capture a less noisy attractor; however, for high levels of additive noise, a combination of a refinement procedure with ANN is used. In this case, only one refinement is needed for the successful use of ANN. The obtained ANN model is then used for long-term predictions of the future behavior of a Hénon attractor, using information based only on past values.

KEYWORDS: Chaotic attractors; noise reduction; artificial neural network

1. INTRODUCTION

During the last decade, the problem of noise reduction has been a subject of interest in dynamical systems [1], signal processing [2] and experimental measurements [3]. A variety of methods of noise reduction has been treated by different investigators. Kostelich and Yorke [4] implemented a least-square noise reduction procedure that works by taking many nearby points in phase space to find a local approximation of the dynamics. Farmer and Sidorovich [5] described an idea that exploits local expanding and contracting directions when the map is known. Hammel [6] used a noise reduction by shadowing based on the proof of the shadow Lemma [7,8] and where the map is also known. Other related work in this direction is that of Smaoui [9,13] and Casdagli [10]. In this paper, we describe a noise-reduction procedure using Artificial Neural Network (ANN) on the time series data without knowing the underlying map of the attractor.

Chaos is a dynamical system that exhibits aperiodic behavior that depends on initial conditions. This sensitivity to initial conditions makes one wonder whether the orbit of a chaotic process is a true orbit or an orbit that is contaminated with noise. A true orbit $\{x_n\}_{n=0}^N$, satisfying $x_{n+1} = f(x_n)$, $0 \leq n \leq N - 1$ is an orbit free of noise. There are several ways by which a dynamical system, represented by f , can be contaminated by noise. One way is when errors are made upon each iteration of the process, thus generating a noisy orbit. This type of noise generation is called dynamic noise and it is associated with the shadowing problem [7,8]. Chaotic processes have the property that very small numerical errors grow exponentially fast, therefore, it is not clear that a numerical orbit may mimic a true orbit. Hammel [11,12] has demonstrated that some numerically generated chaotic orbits can be shadowed by true orbits for long time periods and Smaoui [13] has shown that a computer-generated orbit of the chaotic attractor outside the periodic window of the quadratic map can be shadowed for all time. Another way by which a dynamical system can be contaminated by noise is when the true physical process is obscured by errors in measurements. This type of noise is called an additive noise. A noisy orbit $\{p_n\}_{n=0}^N$ is generated by $p_n = x_n + \epsilon_n$ for small additive noise $|\epsilon_n| < \delta$. In this work, we apply an ANN noise reduction procedure on an additive noisy orbit of the Hénon attractor where the underlying map is not known.

The paper is organized as follows: In section 2 we describe the ANN method. Section 3 defines the Hénon attractor. Section 4 shows the application of the ANN noise reduction method on the Hénon attractor with and without noise. In that section, a noise reduction approach which is a combination of ANN and a refinement procedure when the noise level is high is also presented and we conclude in section 5.

2. NEURAL NETWORK APPROACH

Artificial Neural Networks (ANN) has been widely used in many different disciplines to perform complex functions in various fields of applications including pattern recognition, identification, classification, speech, vision and control systems [14-18]. Recently ANN has been used to obtain a nonlinear map which, given the present

measurement, and possibly some recent history, will predict the state of the system at the next time step [19]. It is the ability to control nonlinearity in the neural net that allows prediction of chaotic time series with an accuracy far exceeding conventional methods. While chaos can occur in very simple deterministic systems, chaotic time series are sufficiently complicated that they appear to be “random” time series. Thus, random-looking data may contain simple deterministic relationships that involve only a few irreducible degrees of freedom. The Henon map is used as an example to show how ANN can capture the underlying map from the time series. Before analyzing this example, we explain how ANN works.

ANN are composed of many simple elements or nodes, operating in parallel, communicate with each other through connecting synapses. The architecture of these models depends on the node characteristics, network topology and learning algorithm which are described below.

2.1 PROCESSING NODES

Nodes are considered the key components of a neural net. They can be described as simple functions that take n variables as argument and output only one which can serve as input to other nodes. The number of variables connected to a particular node depends on its position in the network. Because the network is inherently parallel, all the nodes perform their computations simultaneously.

2.2 ACTIVATION STATE

The activation state u_j consists of the product of all inputs from nodes of the previous layer multiplied by their corresponding synaptic weights minus the bias of the j th node.

$$u_j = \sum w_{ij}x_i - \beta_j \quad (1)$$

where x_i is the measurement or input pattern of the i th layer, w_{ij} is the weight and β_j is the bias of the j th node. This activation can be positive, zero, or negative, because the synaptic weightings and the inputs can be either positive or negative. A positive weighted input is considered a positive contribution to the activation and therefore represents a stimulus; however, a negative contribution represents an inhibition.

2.3 OUTPUT FUNCTION

Once an activation is determined, then a signal transfer function is applied to that activation to determine an output. Several functional forms might be used as a transfer function. The transfer function used in this work is:

$$S(u) = \frac{2}{(1 + e^{-2u})} - 1. \quad (2)$$

This function belongs to the class of sigmoidal functions and has advantageous characteristics such as being continuous, differentiable at all points, and monotonically increasing. It also accepts inputs varying from $-\infty$ to ∞ and produces outputs over a finite range from -1 to 1.

2.4 NETWORK TOPOLOGY

Several ANN topologies, that differ in the number and character of the processing nodes, the connections and the training procedures, have been proposed [20-22]. In this work, a multilayer feedforward neural network which consists of a six-node input layer, two ten-node hidden layers, with nonlinear sigmoid function S , and a two-node output layer with linear transfer function is used (see Figure 1).

With this network topology, the aim is to obtain an explicit nonlinear map

$$x_{n+1} = f(x_n, x_{n-1}, \dots, x_{n-m}), \quad (3)$$

where m is the number of chosen delays. For future prediction, this map can be iterated to give

$$x_{n+1} = f(f(x_n, x_{n-1}, \dots, x_{n-m}), x_n, x_{n-1}, \dots, x_{n-m+1}) \quad (4)$$

or, using an $m+1$ dimensional vector $x_n = (x_n, x_{n-1}, \dots, x_{n-m})^T$, we can define the map

$$x_{n+1} = F(x_n). \quad (5)$$

2.5 LEARNING

Training a network is an essential factor for the success of the neural network. Levenberg-Marquardt method [23], which makes use of a compromise between the Gauss-Newton and the steepest descent approaches, has been used. The method has been shown to be efficient and reliable for training feedforward networks. The overall error is usually defined as

$$sse = \frac{1}{2} \sum_p \sum_k (z_k - y_k)_p^2 \quad (6)$$

where z_k and y_k are the desired and actual output vector of the k th output node. The p subscript refers to the specific input vector pattern used. The weights leading into an output node k are adjusted in proportion to the difference between the actual node output and its target output. This can be accomplished by considering the derivative of the error function with respect to the change in synaptic weight leading into an output node. The weight adjustment on a hidden node h , with input signal x_h , leading to output node k is given by

$$\Delta w_{hk} = (J^T J + \mu I)^{-1} J^T e, \quad (7)$$

where J is the Jacobian matrix of derivatives of each error to each weight, e is an error vector and μ is a scalar. As limiting cases, we obtain the Gauss-Newton method if $\mu \rightarrow 0$ and the steepest descent method if $\mu \rightarrow \infty$. Large values for μ give a small step-length, i.e. minor corrections of the weights.

3. THE HÉNON ATTRACTOR

The Hénon map was first introduced by Hénon [24] as a simple two-dimensional map possessing special properties of more complicated systems. This map with $\underline{x}_n = (x_{1,n}, x_{2,n})^T$ is given by

$$\underline{x}_{n+1} = f(\underline{x}_n) = \begin{pmatrix} x_{1,n+1} \\ x_{2,n+1} \end{pmatrix} = \begin{pmatrix} 1 + x_{2,n} - ax_{1,n}^2 \\ bx_{1,n} \end{pmatrix}. \quad (8)$$

Hénon [24] studied this map for various values of a and b and observed that for the values of $a = 1.42$ and $b = 0.3$, the map possess chaotic attractor. This chaotic attractor, shown in Figure 2, depicts a scatterplot of the first 3000 iterations of the map for the initial conditions $x_1 = 0$ and $x_2 = 0$.

Theoretically, a good neural network model can be obtained if many time delays or old observations are included, but this approach will not be practical since it leads to a large network with a considerable number of weights to be estimated [25]. To determine the optimal number of input nodes in the ANN, we use the same strategy as Packards [26] who demonstrated that an attractor may be reconstructed from a time series if a correct number of time delayed samples of the series is used. Let τ be a chosen delay, and m be an integer, then any point on the attractor can be written

$$x_{n+P} = f(x_n, x_{n-\tau}, x_{n-2\tau}, \dots, x_{n-(m-1)\tau}) \quad (9)$$

where P is a prediction time into the future chosen to be 1 for our work, and $f()$ is a map. With this formulation, the embedding dimension of the attractor is $d_E = m$. Takens [27] has proved that this embedding dimension is defined as

$$d_F \leq d_E \leq 2d_F + 1 \quad (10)$$

where d_F is the fractal dimension. The fractal dimension was calculated using a box counting algorithm and was found to be $d_F \approx 1.25$. Based on this fractal dimension, an embedding dimension, $d_E = 3$, for the Hénon attractor is estimated using Eq.(10). Therefore, there are six inputs to the nonlinear net consisting of $x_{1,n-2}, x_{1,n-1}, x_{1,n}, x_{2,n-2}, x_{2,n-1}$, and $x_{2,n}$ and two linear output elements representing the values $x_{1,n+1}$ and $x_{2,n+1}$.

A set of 500 points was used to train the network (i.e. $\{x_{1,n}\}_{n=1}^{500}$ and $\{x_{2,n}\}_{n=1}^{500}$). Once the training stage is successful (i.e., $sse=5 \times 10^{-4}$), the network is used to predict $x_{1,n+1}$ and $x_{2,n+1}$ for 500 points not included in the training data set (i.e. $\{x_{1,n}\}_{n=501}^{1001}$ and $\{x_{2,n}\}_{n=501}^{1001}$). Figure 3 shows the attractor in phase space for the predicted and the desired data points. The ANN was successful in capturing the complex structure of the noiseless Hénon attractor. The average relative error of the testing data for the values of x_1 and x_2 was found to be 0.3% and 0.4%, respectively.

4. A NOISE REDUCTION APPROACH

So far, we have shown that, in the absence of noise, the Hénon attractor was completely captured (i.e. the trajectory is completely deterministic). However, consider the case where the original trajectory $\{x_n\}_{n=0}^N$ is distorted by an additive noise ϵ_n to obtain a noisy trajectory $\{p_n\}_{n=0}^N$ defined as

$$p_n = x_n + \epsilon_n. \quad (11)$$

Figure 4 presents a noisy Hénon attractor with an average additive noise $\sigma = 1.15 \times 10^{-2}$, where $\sigma = \frac{1}{N} \sum_{n=0}^N |\epsilon_n|$. The same architecture was used to train a network with a set of 500 noisy data points (i.e. $\{p_{1,n}\}_{n=501}^{1001}$ and $\{p_{2,n}\}_{n=501}^{1001}$). The network compares its actual response with the target noisy response and adjusts its weights in such a way to minimize the *sse*. Since the target response is noisy, we pick an *sse* = 0.05. After convergence, the network captures the hidden attractor obtained from iteration of Eq.(3) rather than reproduce the noisy attractor. Comparing the noisy Hénon attractor shown in Figure 4 with the one obtained from the ANN model shown in Figure 5, one can clearly observe that the ANN model have reduced the noise to a minimum.

The successful noise reduction method using ANN is superior over previous methods since the map is not known in advance as in [5,6]. This noise reduction procedure was successful for all values of $\sigma < 5 \times 10^{-2}$. Of course, there are some limitations to this procedure. One of which is that when the noise level is greater than 5×10^{-2} , the ANN captures a distorted attractor. One solution for such a case is to combine Hammel's refinement procedure [6,11] and ANN. We have successfully applied the refinement procedure on the noisy Hénon attractor with noise level $\sigma = 6.14 \times 10^{-2}$. First we have examined fifty applications of the refinement procedures without using ANN. Because of machine precision, the refinement procedure does not improve after the fifteenth refinement. Next, ANN has been successfully used on the less noisy attractor obtained after one refinement to completely eliminate the noise. The obtained ANN model has the advantage of predicting the true behavior of the system knowing the present state and some recent history.

5. CONCLUSIONS

In this paper we have demonstrated the successful use of ANN to reduce noise in chaotic attractors. We have applied this technique on the Hénon attractor with and without noise. In both cases, only the trajectories or sequences of points are assumed to be known as compared to other noise reduction techniques where the map is assumed to be known. The fractal dimension of the attractor was used as an indicator to determine the number of input layer nodes. In the case where the attractor is noiseless, the ANN was able to predict the chaotic behavior within 1% of accuracy. When additive noise was added to each point of the trajectories, the ANN method was able to capture a less noisy attractor for all noise levels up to $\sigma = 5 \times 10^{-2}$. In case where the noise level σ is high, a combination of a refinement procedure and ANN method is used to successfully reduce the noise. Only one refinement procedure was needed for the successful use of ANN. It must be noted that the technique presented here is used not only to reduce noise, but also to generate the future behavior of chaotic attractors.

6. REFERENCES

1. Moss, F., Lugiato, L., and Schleich, W., 1990, "Noise and Chaos in Nonlinear Dynamical Systems", Cambridge University Press.
2. Vaseghi, S.V., 1996, "Advanced Signal Processing and Digital Noise Reduction", John Wiley.
3. Brillinger, D.R., 1980, "Data Analysis and Theory", Holden-Day, San Francisco.
4. Kostelich, E. and Yorke, J.A., 1988, "Noise reduction in dynamical systems", Phys. Rev. A, Vol. 38, no. 3, 1649-1652.
5. Farmer, J.D., and Sidorovich, J.J., 1988, "in: Evolution, learning and cognition", ed. Y.C. Lee, World Scientific, Singapore.
6. Hammel, S.M., 1990, "A noise reduction method for chaotic systems", Vol. 148, no. 8,9. Phys. Lett., 421-428.

7. Anosov, D.V., 1967, "Geodesic flows and closed Riemannian manifolds with negative curvature", Proc. Steklov Inst., Math. 90.
8. Bowen, R., 1975, " w -limit sets for Axiom A diffeomorphisms", J. Differential Equations, Vol. 18.
9. Smaoui, N., and Al-Adwani, S., 1997, "Using artificial neural networks to model chaotic time series of ethernet traffic", Journal of Mathematical Modelling and Scientific Computing. Vol 8 (1997).
10. Casdagli, M., 1989, "Nonlinear prediction of chaotic time series", Physica D 35, 335-356.
11. Hammel, S.M., Yorke, J.A., and Grebogi, C., 1988, "Numerical orbits of chaotic processes represent true orbits", Vol. 19, no. 2, 465-469.
12. Hammel, S.M., Yorke, J.A., and Grebogi, C., 1987, "Do numerical orbits of chaotic dynamical processes represent true orbits?". Journal of Complexity, Vol. 3, 136-145.
13. Smaoui, N., and Kostelich, E., 1998, "Using chaos to shadow the quadratic map for all time", Intern. J. Computer Math., Vol. 70, pp. 117-129.
14. Boulard, H., and Wellekens, C.J., 1989, "Speech dynamics and recurrent networks", Proc. ICASSP, pp.33-36.
15. Ephraim, Y., and Rabiner, L., 1990, "On the relations between modelling approaches for speech recognition", IEEE Trans. Acoust. Speech. Signal Processing, 36: 372-379
16. Antsaklis, P., 1990, "Neural networks in control systems", IEEE Contr. Syst. Mag., 10 (3): 3-5.
17. Pao, J.H., 1989, "Adaptive Pattern Recognition and Neural Networks", Addison-Wesley.
18. Hopfield, J.J., and Tank, D.W., 1985, "Neural computation of decisions optimizations problems", Bio. Cybern. Vol. 52 ,141 pp.
19. Lapedes, A., and Farber, R., 1987, "Nonlinear signal processing using neural networks: Prediction and signal modelling", preprint, Los Alamos.
20. Hopfield, J.J., 1982, "Neural networks and physical systems with emergent collective computational abilities", Proc. Nat. Academy Sci. 79: 2554-2558.
21. Feldman, J., and Ballard, D.H., 1982, "Connectionist models and their properties", Cognitive Sci. 6: 205-254.
22. Rumelhart, D.E., and McClelland, J., 1986, "Parallel Distributed Processing: Explorations in the Microstructure of Cognition", MIT Press, Cambridge, Mass.
23. Marquardt, D.W., 1963, "An algorithm for least-squares estimation of nonlinear parameters", J. SIAM 11, 431-441.
24. Hénon, M., 1976, "A two dimensional mapping with a strange attractor", Comm. Math. Phys., Vol. 50, 69-77.
25. Billings, S.A., Jamaluddin, H.B., and Chen, S., 1992, "Properties of neural networks with applications to modelling nonlinear dynamical system", Int. J. Control, Vol. 55, 193-224.
26. Packards, N. et al., 1980, Phys. Rev. Lett. 45 (9):712
27. Takens, F., 1981, "Detecting Strange Attractor in Turbulence", Lecture Notes in Mathematics, Rand, D., Young,L., (editors). Springer Berlin