Nonlinear Reduction of High-Dimensional Dynamical Systems via Neural Networks

Michael Kirby* and Rick Miranda

Department of Mathematics, Colorado State University, Ft. Collins, Colorado 80523

(Received 16 August 1993)

A technique for empirically determining optimal coordinates for modeling a dynamical system is presented. The methodology may be viewed as a nonlinear extension of the Karhunen-Loève procedure and is implemented via an autoassociative neural network. Given a high-dimensional system of differential equations which model a dynamical system asymptotically residing on a stable attractor, the task of the network is to compute a reembedding of the attractor and the dynamics into an ambient space which reflects the intrinsic dimensionality of the problem. The method is demonstrated on the unforced Van der Pol oscillator, the forced Van der Pol, and the Kuramoto-Sivashinsky equation.

PACS numbers: 42.79.Ta, 02.60.Lj, 89.70.+c

Many physical systems are modeled by high-dimensional systems of nonlinear differential equations. It is generally the case that these equations cannot be solved analytically and consequently some type of numerical approximation of the solutions is sought. Inherent in this process is the introduction of a coordinate system. For instance, a Galerkin approximation of a partial differential equation requires a projection onto some set of complete functions, e.g., sinusoids if the problem has periodic boundary conditions. Such an approach often produces a system of equations which is larger than the true number of degrees of freedom of the phenomena being modeled. This Letter proposes a technique for constructing a system of equations which more closely reflects the intrinsic dimensionality of the dynamics under investigation.

The Karhunen-Loève (KL) decomposition has been widely used as a tool for the dimensionality reduction of dynamical systems, see [1]. The eigenfunctions produced by this procedure provide an optimal *linear* coordinate transformation which permits the original dynamical model to be reformulated. Ideally, a subset of the transformed equations actually models the dynamics and the remaining equations may be ignored.

Unfortunately, however, this method fails to produce a minimal coordinate system in simple cases. For instance, a periodic solution should be parametrizable by a single equation given that a closed curve, such as a circle, may be viewed topologically as a one-dimensional manifold. It is apparent that the KL expansion requires two eigenvectors and hence two equations to reconstruct the dynamics in this case. In general, the Euclidean space spanned by the KL basis may be viewed as *encapsulating* the nonlinear manifold and the intrinsic dimensionality of the dynamics may be of lower dimension.

The method we propose is a nonlinear approach to dimensionality reduction of dynamical systems. The nonlinearity is achieved by the use of a neural network which compresses the dynamics via a nonlinear parametrization. We note that our goal is fundamentally different from the studies which attempt to reconstruct the attractor without knowledge of a high-dimensional system of governing equations. Neural networks have been applied extensively with very positive results for attractor reconstruction and more generally time-series analysis where no high-dimensional equations are available; see, e.g., [2-5]. The problem of reducing the dimensionality of a given dynamical description is extremely important and challenging and how to best arrive at a system of low-dimensional equations given a much larger system is an area of active research [1,6,7].

Examples of the method we propose are presented for the unforced Van der Pol oscillator, the forced Van der Pol oscillator, and the Kuramoto-Sivashinsky equation. This last equation offers an example of a twenty-dimensional system [obtained from a partial differential equation (PDE)] whose stable attractor is diffeomorphic to a circle; our neural network employs a single *circular* node capable of modeling this topology.

We assume that we are given a system of ordinary differential equations

$$\dot{u}(t)=F(u) \tag{1}$$

the trajectories of which eventually lie on an attractor U which is a subset of $\mathbb{R}^n$, i.e., $u\in U\subset\mathbb{R}^n$. We postulate that, as a topological space, the attractor U can be differentiably embedded as a manifold $V\subset\mathbb{R}^m$ in a lower-dimensional space. The dynamical system or vector field on U may then be transported to one on V via the (typically nonlinear) embedding.

Let G be a nonlinear embedding which takes an element of the attractor in U to an element in V. Specifically

$$G\colon U\rightarrow V$$

is given by

$$v_i(t)=G_i(u(t))\,.$$

The assumption that G is a differentiable embedding implies that the induced vector field

$$\dot{v}_j(t)=\sum_{i=1}^{n}\frac{\partial G_j}{\partial u_i}F_i(u(t)) \tag{2}$$

0031-9007/94/72(12)/1822(4)$06.00

on V is topologically equivalent to the original dynamical system on U (G is a homeomorphism of U onto V). If the original equations F are known and the embedding function G is known then Eq. (2) gives explicit equations for the dynamical system on the low-dimensional manifold V. Hence it is the construction of such an embedding G which offers a direct route to the dimensionality reduction for the original system.

Notice in Eq. (2) that the value of $u(t)$ is required in the expression for the low-dimensional vector field equations. Hence an inverse to G (giving $u \in U$ as a function of coordinates on V) is needed for (2) to be useful.

Suppose that we have a mapping

$$H:\ V \rightarrow W \subset \mathbb{R}^n$$

defined by

$$w_i(t) = H_i(v(t)),$$

which is an *approximate* inverse of G. The space W may be viewed as a reconstruction of the flow in the original parametrization, namely, that $U \approx W$ or in other words $u_i(t) \approx w_i(t)$.

The nonlinearly transformed dynamical system on the manifold V is then given by

$$\left.\frac{dv_j}{dt}\right|_{v(t)} = \sum_{i=1}^{n} \frac{\partial G_j}{\partial u_i} F_i(H(v(t)))\,, \qquad (3)$$

where $j = 1, \ldots, m$. Hence this formulation has reduced the original system of n equations to a smaller system of $m < n$ equations on the manifold V.

Below the empirical construction of these mapping functions will be presented.

The presentation given above assumes that the spaces U and V are connected by a pair of mappings G and H which enables us to derive a reduced set of dynamical equations. Cybenko [8] has shown that finite sums of sigmoidals, i.e., functions $\sigma(x)$ with the property that $\sigma(x) \rightarrow 0$ as $x \rightarrow -\infty$ and $\sigma(x) \rightarrow 1$ as $\sigma(x) \rightarrow \infty$ are dense with respect to continuous functions on the unit cube. Thus we can approximate any mapping G from U to another space V of dimension m by the superposition of sigmoidals

$$v_i \approx \sum_{j=1}^{N_h} w_{ji}\sigma\left[\sum_{k=1}^{n} \bar{w}_{kj}u_k - \theta_j\right] - \theta_i\,, \qquad (4)$$

where $i = 1, \ldots, m$ and $\sigma(x) = [1 + \exp(-x)]^{-1}$. Cybenko's result suggests that a neural network with one hidden layer can construct a mapping G to any desired accuracy. The error in the approximation is reduced as we increase the number N_h of nodes in the hidden layer.

One possibility for constructing G and H is based on a neural network architecture originally proposed by Kramer [9] as a nonlinear principal component analyzer for data analysis problems. The network has the general form

$$\left\{\begin{array}{c}\text{input}\\ \text{layer}\\ n\ \text{nodes}\end{array}\right\} \rightarrow \left\{\begin{array}{c}\text{mapping}\\ \text{layer}\\ N_h\ \text{nodes}\end{array}\right\} \rightarrow \left\{\begin{array}{c}\text{bottleneck}\\ \text{layer}\\ m \ll n\ \text{nodes}\end{array}\right\} \rightarrow \left\{\begin{array}{c}\text{demapping}\\ \text{layer}\\ M_h\ \text{nodes}\end{array}\right\} \rightarrow \left\{\begin{array}{c}\text{output}\\ \text{layer}\\ n\ \text{nodes}\end{array}\right\}.$$

We identify the input data as being elements of the flow of Eq. (1) and the space V as the output of the bottleneck layer. The reconstructed data are generated at the output layer. If one trains this network to reproduce the identity function on the data set, the mapping obtained from the input layer (of dimension n) to the bottleneck layer (of lower dimension m) will be an *invertible* and *nonlinear* modeling of the data set; this is the desired mapping G. The mapping obtained from the second half of the network, which maps the bottleneck layer to the output layer, will be the approximate inverse H. It is on the *bottleneck manifold* V that we construct the reduced set of differential equations.

The network is presented with an ensemble of training patterns which lie in U and is adapted until the output approximates the input, i.e., $w \approx u$. Thus the network is trained by adjusting the parameters in Eq. (4) so that the quantity $E = \langle \|u - w\|^2 \rangle$ is minimized, where the angle brackets denote averaging over the data ensemble. This error is minimized using backpropagation [10] and a standard conjugate gradient procedure [11]. We note that other measures of error may lead to improved accuracy for G and H and/or reduced network training costs.

The forced Van der Pol oscillator is modeled by the system

$$\dot{u}_1 = u_2\,,$$

$$\dot{u}_2 = -u_1 + \alpha(1 - u_1^2)u_2 + \beta\cos(\gamma u_3)\,,$$

$$\dot{u}_3 = 1\,.$$

When β is zero this reduces to the unforced Van der Pol oscillator in two dimensions; in this case the existence of a stable periodic orbit has been shown for any $\alpha \in \mathbb{R}$; see, e.g., [12]. A phase portrait obtained via numerical integration of this system for $\alpha = 1$, $\beta = 0$ produces the limit cycle shown in Fig. 1. Since this two-dimensional system $U \subset \mathbb{R}^2$ describes an attractor in the form of a limit cycle, we should be able to compute the flow on a one-dimensional manifold V; specifically, $V = S^1$.

The unforced Van der Pol oscillator possesses symmetry which enables us to further reduce the domain on which it is necessary to define the functions G and H. Specifically, if $u(t) = (u_1(t), u_2(t)) \in U$ then $\Gamma u \in U$ where $\Gamma u = (-u_1, -u_2)$. Thus we construct the function $G(u_1, u_2)$ on its fundamental domain $u_2 \geq 0$ and extend

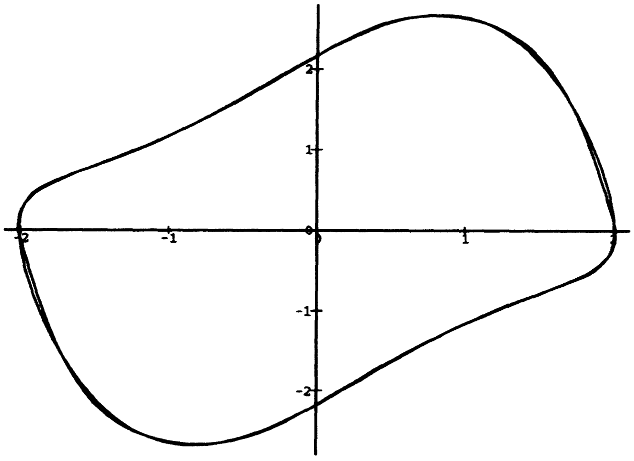

FIG. 1. The limit cycle solution of the Van der Pol equation and its overlapping reconstruction.

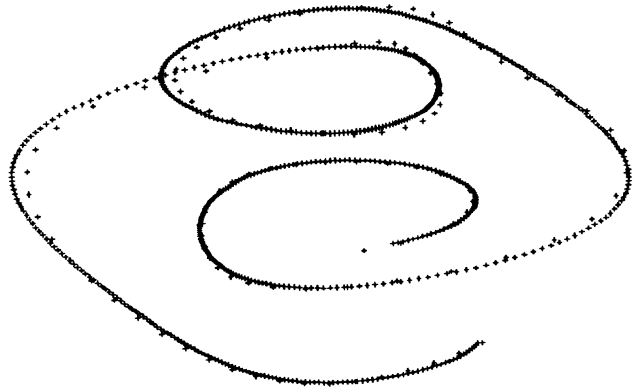

FIG. 3. Trajectory of the forced Van der Pol and its reconstruction.

this via symmetry to the entire plane. Thus on the fundamental domain we integrate the single equation

$$\frac{dv}{dt} = \frac{\partial G}{\partial u_1}\dot{u}_1(t) + \frac{\partial G}{\partial u_2}\dot{u}_2(t)\Big|_{u(t)\approx H(v(t))} \qquad (5)$$

while for $u_2 < 0$ we replace $v(t)$ above with $v(t) = G(\Gamma u(t))$. For this example exploiting the symmetry in this manner reduces the number of computations required to construct the mapping function G by a factor of 2. The role of symmetry here is merely to reduce the computational expense of the calculations for the mappings G and H; it is an independent effect from the dimensionality reduction which the bottleneck neural network technique offers.

In Fig. 2 we graph the value of the bottleneck variable $v(t)$ evolved by integrating Eq. (5). The accuracy of the reformulation may be determined by mapping v back onto the reconstructed attractor W. The reconstruction is also displayed in Fig. 1 and it is seen to be nearly identical to the original curve.

Integrating the forced Van der Pol equation with $\alpha = 1$, $\beta = 10$, and $\gamma = 3$ produces a trajectory shown in Fig. 3. The construction of the mapping functions G and H in this case used a 3-20-1-20-3 bottleneck neural network architecture. The reconstruction of the dynamics is also displayed in Fig. 3 and is seen to approximate the original trajectory quite well.

A more serious example is offered by the Kuramoto-Sivashinsky partial differential equation

$$u_t + 4u_{xxxx} + \alpha[u_{xx} + \tfrac{1}{2}(u_x)^2] = 0\,. \qquad (6)$$

A standard approach for numerically integrating this PDE is based on using a Galerkin approximation to produce a large system of ordinary differential equations [13]. We integrated ten complex equations which produce a limit cycle for (6) in $\mathbb{R}^{20}$. This is displayed as a

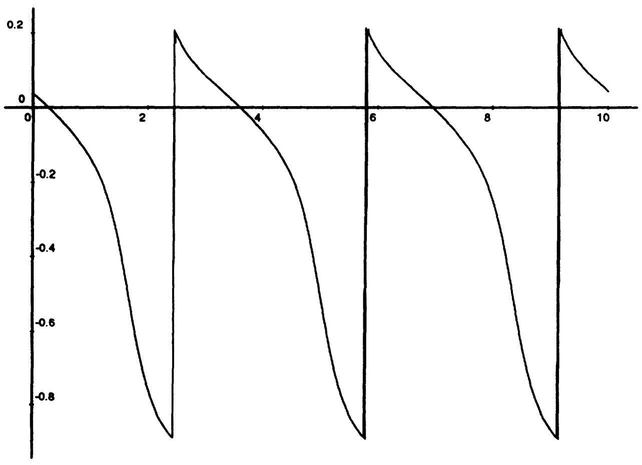

FIG. 2. Evolution of $v(t)$ on the bottleneck manifold V.

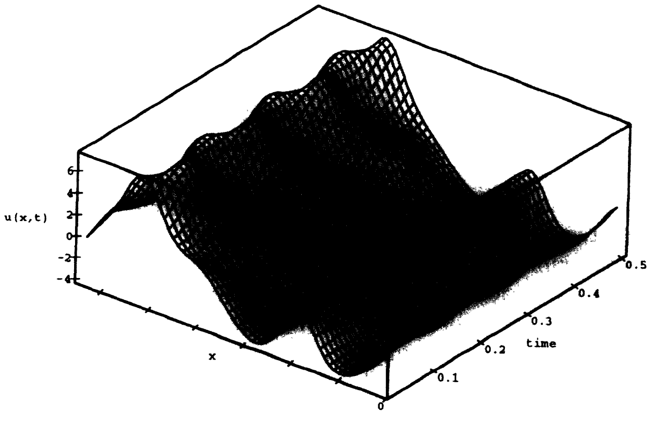

FIG. 4. Numerical simulation of the KS equation for $\alpha = 84.0$.

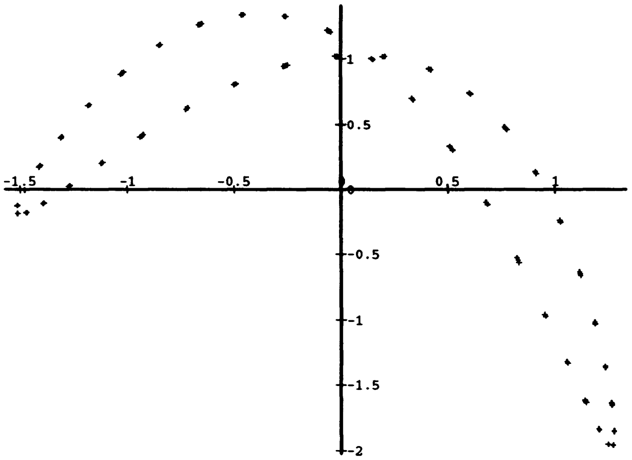

FIG. 5. First complex mode of the reconstructed data for the Kuramoto-Sivashinsky equation.

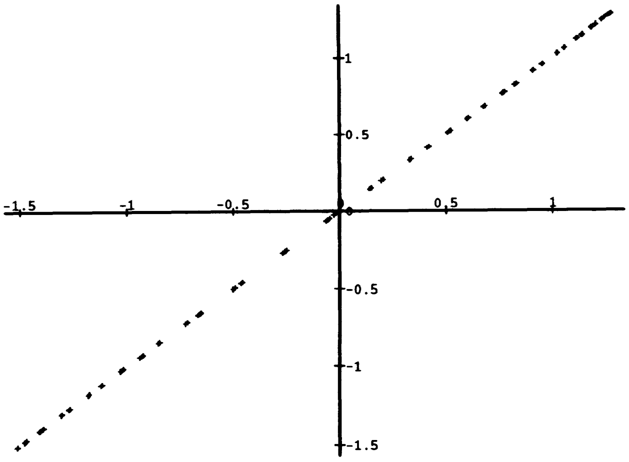

FIG. 6. Graph of the input versus output for the first real mode of the Kuramoto-Sivashinsky equation.

complicated structure with a broad Fourier spectrum in physical space in Fig. 4.

A special type of neural architecture incorporating circular nodes which are capable of encoding angular information [14] was used in a bottleneck neural network to produce mapping functions G and H exhibiting inverse diffeomorphisms from the limit cycle $U \subset \mathbb{R}^{20}$ to $V = S^1$. The reconstruction of first Fourier mode of the simulated data is displayed in Fig. 5. In Fig. 6 we display the first real input node values versus the first real output node values; the fact that they lie on the straight line $y = x$ shows that the identity mapping is being well approximated. All twenty nodes exhibit the same accuracy.

In summary, we have presented an approach for reducing a given set of dynamical equations by means of a nonlinear transformation constructed empirically by a feedforward neural network with three hidden layers. The resulting dynamical system is of lower dimensionality but the solutions are qualitatively equivalent to the original system. The approach is capable of producing low-dimensional equations modeling the dynamics of the original high-dimensional system. The approach was demonstrated on the forced and unforced Van der Pol oscillators and on the Kuramoto-Sivashinsky equation; in each of these cases a single scalar equation on the bottleneck manifold captured the dynamics.

The research is supported in part by the NSF under Grant No. ECS-9312092.

*Electronic address: kirby@ritz.math.colostate.edu

[1] L. Sirovich, Q. Appl. Math. XLV **3**, 561 (1987).

[2] A. S. Lapedes and R. Farber, Technical Report No. LA-UR-87-2662, Los Alamos National Laboratory, Los Alamos, NM (unpublished).

[3] M. Casdagli, Physica (Amsterdam) **35D**, 335 (1989).

[4] D. Demers and G. Cottrell, in *Neural Information Processing Systems,* edited by S. J. Hanson, J. D. Cowan, and C. L. Giles (Morgan Kauffman, 1992), p. 582.

[5] *Time Series Prediction,* edited by A. S. Weigend and N. A. Gershenfeld (Addison-Wesley, Reading, MA, 1993).

[6] C. Foias, O. P. Manley, and R. Temam, J. Math. Anal. Appl. **178**, 567 (1993).

[7] R. D. Russell, D. M. Sloan, and M. R. Trummer, SIAM. J. Sci. Comput. **14**, No. 1, 19 (1993).

[8] G. Cybenko, Math. Control, Signal Sys. **2**, 303 (1989).

[9] M. A. Kramer, AIChE J. **37**, 233 (1991).

[10] *Parallel Distributed Processing, Vol. 1,* edited by D. E. Rummelhart, J. E. McClelland, and the PDP research group (MIT Press, Cambridge, MA, 1986).

[11] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing* (Cambridge Univ. Press, Cambridge, 1988).

[12] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields* (Springer-Verlag, New York, 1986).

[13] M. Kirby and D. Armbruster, Z. Angew. Math. Phys. **43**, 999 (1992).

[14] M. Kirby and R. Miranda (to be published).