CIRCULAR NODES IN NEURAL NETWORKS

MICHAEL J. KIRBY AND RICK MIRANDA

Department of Mathematics
Colorado State University
Fort Collins, CO 80523

ABSTRACT. In the usual construction of a neural network, the individual nodes store and transmit real numbers which lie in an interval on the real line; the values are often envisioned as amplitudes. In this article we present a design for a *circular* node, which is capable of storing and transmitting angular information. We develop the forward and backward propagation formulas for a network containing circular nodes. We show how the use of circular nodes may facilitate the characterization and parametrization of periodic phenomena in general. We describe applications to constructing circular self-maps, periodic compression, and one-dimensional manifold decomposition. We show that a circular node may be used to construct a homeomorphism between a trefoil knot in $\mathbb{R}^3$ and a unit circle. We give an application with a network which encodes the dynamical system on the limit cycle of the Kuramoto-Sivashinsky equation. This is achieved by incorporating a circular node in the bottleneck layer of a three-hidden-layer bottleneck network architecture. Exploiting circular nodes systematically offers a neural network alternative to Fourier series decomposition in approximating periodic or almost periodic functions.

1. INTRODUCTION

A sigmoidal node of a standard feed-forward neural network stores information as a real number in a bounded interval. Thinking geometrically, such a node is capable of encoding a point in the 1-dimensional manifold I which is the open interval. Up to homeomorphism, there are only two 1-dimensional manifolds: the open interval I and the circle S^1. The open interval is capable of and appropriate for encoding amplitude information; the circle is capable of and appropriate for encoding angular information.

In this article we propose a "circular" node which is capable of encoding angular information and give one implementation of it which we have found expedient. We present both the forward propagation formulas and the back-propagation algorithm for a network containing circular nodes.

Key words and phrases. circular nodes, nonlinear dimensionality reduction, parametrization of periodic phenomena, periodic compression, bottleneck architecture, Kuramoto-Sivashinsky equation.

Research supported in part by the NSF under grant no. ECS-9312092.

January 27, 1995. Set in type by AMSLaTeX Version 1.1

We describe several applications in pattern analysis and the analysis of high-dimensional dynamical systems at the end of this article, culminating in a neural network parametrization and compression of the limit cycle of the Kuramoto-Sivashinsky partial differential equation. This is an extension of the neural network description of the limit cycle for the Van der Pol oscillator, described in (Kirby and Miranda (1994a).

Another approach for encoding of angular information in neural networks has been presented in (Zemel, Williams, and Mozer 1993). The reader may also be interested in the approach of (Gislén, Peterson, and Söderberg 1992) which uses rotor neurons to process data lying on an n-sphere.

2. Implementation of Circular Nodes

In our implementation, a circular node is actually a pair of coupled nodes, whose values are constrained to lie on the unit circle.

To be specific, we suppose that the neural network has L layers, numbered from 0 to $L-1$. In each layer there are nodes, which can be either of two types: circular or sigmoidal. The j^{th} node in layer i will be denoted by $\mathcal{N}_j^{(i)}$. The number of nodes in layer i is $N^{(i)}$, and for fixed i the nodes $\mathcal{N}_j^{(i)}$ are indexed from $j = 0$ to $j = N^{(i)} - 1$. At each node $\mathcal{N}_j^{(i)}$ the state value of the network is denoted by $S_j^{(i)}$.

The circular nodes occur in coupled pairs, i.e., if node $\mathcal{N}_j^{(i)}$ (node j in layer i) is part of a circular node, then there will be a coupled node $\mathcal{N}_{\tau(j)}^{(i)}$ (node $\tau(j)$, also in layer i). Thus there are an even number of such nodes, in coupled pairs; we assume that $\tau(\tau(j)) = j$.

By construction the state values for coupled circular nodes will always satisfy the circular constraint

$$(S_j^{(i)})^2 + (S_{\tau(j)}^{(i)})^2 = 1. \tag{2.1}$$

We think of each pair of such coupled nodes as a single "abstract circular" node, whose joint state value represents true angular information.

This implementation of a circular node fits naturally into other types of networks also (e.g., ones with linear and sigmoidal units, or ones with feedback); we examine the circular node in this simplified yet fundamental situation in order to isolate its unique features.

3. The Forward Propagation of the Network

In the architecture described above, each state value in the network $S_j^{(i)}$ is determined from the state values $S_k^{(i-1)}$ occurring in the previous layer. The inputs are the state values $S_j^{(0)}$ at the nodes of the input layer 0.

Each node $\mathcal{N}_j^{(i)}$ has associated to it weight constants $w_{jk}^{(i)}$ and bias constants $b_j^{(i)}$. For each node $\mathcal{N}_j^{(i)}$ with $i \geq 1$ (that is, not at the input layer), we define a prestate value $P_j^{(i)}$ by the formula

$$P_j^{(i)} = b_j^{(i)} + \sum_{k=0}^{N^{(i-1)}-1} w_{kj}^{(i-1)} \cdot S_k^{(i-1)}. \tag{3.1}$$

This first step is the same for all nodes, whether they are circular or sigmoidal. Note that the bias constants $b_j^{(i)}$ are defined for all $i = 1, \dots, L-1$, while weight constant $w_{jk}^{(i)}$ are defined for all $i = 0, \dots, L-2$. The final state values are then determined by the prestate values using different formulas depending on the type of node:

$$S_j^{(i)} = \begin{cases} P_j^{(i)}/\sqrt{(P_j^{(i)})^2 + (P_{\tau(j)}^{(i)})^2} & \text{if } \mathcal{N}_j^{(i)} \text{ is circular} \\ \sigma_j^{(i)}(P_j^{(i)}) & \text{if } \mathcal{N}_j^{(i)} \text{ is sigmoidal} \end{cases} \tag{3.2}$$

where

$$\sigma_j^{(i)}(x) = r_j^{(i)}(-1 + \frac{2}{1+e^{-\beta_j^{(i)}x}}).$$

Note that in the circular case, the state values automatically satisfy the circular constraint (2.1).

It is useful to define the *radial value*. Let

$$R_j^{(i)} = \sqrt{(P_j^{(i)})^2 + (P_{\tau(j)}^{(i)})^2}$$

if $\mathcal{N}_j^{(i)}$ is circular. Note that $R_j^{(i)} = R_{\tau(j)}^{(i)}$, and the state formula can then be rewritten as

$$S_j^{(i)} = P_j^{(i)}/R_j^{(i)}$$

in the circular case.

This implementaton of circular nodes requires only a relatively minor alteration of existing network architectures.

4. Error and Gradient Computations via Back-Propagation

Given a set of input state values $S_j^{(0)}$, (which then determine the full set of state values using the forward propagation described above), one has a desired set of *goal* values G_j, as j runs from 0 to $N^{(L-1)} - 1$, i.e., over the output nodes.

The *total squared error* E in the network for this state S (relative to this goal G) is the total squared error between the actual state values $S_j^{(L-1)}$ at the output nodes and the goal values G_j there:

$$E = \frac{1}{2} \sum_{j=0}^{N_{L-1}-1} (S_j^{(L-1)} - G_j)^2. \tag{4.1}$$

If more than one state is used to form an average error, the gradient of the error simply sums over all these states; hence to avoid using an index to indicate the state, we will develop the gradient computations for a single state only; the reader may then take the appropriate sum for an average error over many states.

To develop the formulas for the back-propagation algorithm, which computes the gradient of the error with respect to the weights and biases. Hence we compute the values

$$\frac{\partial E}{\partial b_j^{(i)}} \text{ and } \frac{\partial E}{\partial w_{jk}^{(i)}}$$

for all i,j, and k.

Standard application of the chain rule for derivatives leads to the formulas:

$$\frac{\partial E}{\partial b_j^{(i)}} = \begin{cases} \frac{\partial E}{\partial S_j^{(i)}} \cdot \frac{(P_{\tau(j)}^{(i)})^2}{(R_j^{(i)})^3} + \frac{\partial E}{\partial S_{\tau(j)}^{(i)}} \cdot \frac{-P_j^{(i)} P_{\tau(j)}^{(i)}}{(R_j^{(i)})^3} & \text{if } \mathcal{N}_j^{(i)} \text{ is circular} \\ \frac{\partial E}{\partial S_j^{(i)}} \cdot \sigma_j^{(i)}{}'(P_j^{(i)}) & \text{if } \mathcal{N}_j^{(i)} \text{ is sigmoidal} \end{cases} \tag{4.2}$$

and

$$\frac{\partial E}{\partial w_{jk}^{(i)}} = \begin{cases} \left(\frac{\partial E}{\partial S_k^{(i+1)}} \cdot \frac{(P_{\tau(k)}^{(i+1)})^2}{(R_k^{(i+1)})^3} + \frac{\partial E}{\partial S_{\tau(k)}^{(i+1)}} \cdot \frac{-P_k^{(i+1)} P_{\tau(k)}^{(i+1)}}{(R_k^{(i+1)})^3}\right) \cdot S_j^{(i)} & \text{if } \mathcal{N}_k^{(i+1)} \text{ is circular} \\ \frac{\partial E}{\partial S_k^{(i+1)}} \cdot \sigma_k^{(i+1)}{}'(P_k^{(i+1)}) \cdot S_j^{(i)} & \text{if } \mathcal{N}_k^{(i+1)} \text{ is sigmoidal.} \end{cases} \tag{4.3}$$

It remains then to compute the derivatives $\partial E/\partial S_j^{(i)}$, which is done recursively; at the start of the recursion, we have

$$\frac{\partial E}{\partial S_j^{(L-1)}} = S_j^{(L-1)} - G_j. \tag{4.4}$$

For other "lower" layer values, we have

$$\frac{\partial E}{\partial S_k^{(i-1)}} = \sum_{j=0}^{N^{(i)}-1} \frac{\partial E}{\partial S_j^{(i)}} \cdot \frac{\partial S_j^{(i)}}{\partial S_k^{(i-1)}}, \tag{4.5}$$

where

$$\frac{\partial S_j^{(i)}}{\partial S_k^{(i-1)}} = \begin{cases} \frac{(P_{\tau(j)}^{(i)})^2}{(R_j^{(i)})^3} \cdot w_{kj}^{(i-1)} + \frac{-P_j^{(i)} P_{\tau(j)}^{(i)}}{(R_j^{(i)})^3} \cdot w_{k\tau(j)}^{(i-1)} & \text{if } \mathcal{N}_j^{(i)} \text{ is circular} \\ \sigma_j^{(i)}{}'(P_j^{(i)}) \cdot w_{kj}^{(i-1)} & \text{if } \mathcal{N}_j^{(i)} \text{ is sigmoidal.} \end{cases} \tag{4.6}$$

With these modifications the back-propagation algorithm proceeds as usual.

5. Applications

In this section we will briefly describe several applications of circular nodes in the architecture of a neural network. We begin with certain prototypical constructions, which the reader may easily modify for more elaborate applications.

5.1. Prototypical uses for circular nodes.

- **Parametrization of Periodic Phenomena**
  A locus Γ in $\mathbb{R}^n$ is said to be *periodic* if it is the image of a circle. A *parametrization* of this periodic locus is a function from the circle S^1 to the locus Γ. A neural network designed to produce such a function would have a single circular node as the input layer, one or more hidden layers, and an output layer consisting of n sigmoidal nodes.
- **Circular Self-Maps**
  Mappings from the circle to itself are approximated by neural networks having a single circular node as the input layer, one or more hidden layers and a single circular node as the output layer. A linear combination of networks which reproduce multiplication-angle formulas would explicitly approximate any term of a finite Fourier polynomial, and would then be a direct generalization of non-harmonic decomposition; this can be compared with (Lapedes and Farber 1987).
- **Periodic Compression**
  The converse to parametrization of a periodic locus is the idea of *periodic compression*, that is, taking a periodic locus Γ and mapping it to a circle (see for example (Kirby and Miranda 1994b)). Indeed, the locus Γ need not be a periodic locus; in this case a mapping from Γ to S^1 would simply be a "circular feature extraction"; one might view this as a lossy compression of (almost) periodic data.
- **Circular remodeling of boundary value problems**
  Suppose that $\dot{u} = F(u)$ defines a boundary value problem, which is defined on an irregular open locus in $\mathbb{R}^2$ with an irregular boundary Γ which is homeomorphic to the circle. If the interior of Γ is star-shaped, the existence of a compression mapping $G : \Gamma \to S^1$ may be extended to a mapping from the interior of Γ to the interior of the unit disc, and therefore used to transport the original differential equation to the interior of the unit disc, where solutions may be more easily computed via standard numerical techniques.
- **One-dimensional manifold decomposition**
  The natural generalization of single feature extraction, whether sigmoidal or circular, is multiple feature extraction, where several features of a data set are to be captured simultaneously. If some of the features are amplitudes of various outputs, and others

are periodic, our viewpoint is that this is mathematically best expressed as a mapping from the pattern set Γ to $I^n \times (S^1)^k$; the amplitude features are encoded in the n interval coordinates of the I^n factors, and the angular features are encoded in the k circular coordinates of the $(S^1)^k$ factors. Such a mapping can be considered as a 1-*dimensional manifold decomposition* of Γ.

5.2. Applications to Pattern Analysis. Let $\Gamma \subset \mathbb{R}^N$ be a data set which one wants to optimally compress. Linear methods (Principal Component Analysis, or the Karhunen-Loève decomposition, see (Devijver and Kittler 1982)) discover an optimal ordered coordinate system in $\mathbb{R}^N$ such that Γ lies in the subspace defined by the first few coordinates; this works very well and is efficient if Γ fills up an open subset of a linear subspace.

However, if Γ is a nonlinear subset of $\mathbb{R}^N$, then one must have nonlinear coordinates in order to capture Γ completely and efficiently. Mathematically speaking, we are seeking a nonlinear function $G : \Gamma \to V$ where V is an open subset of $\mathbb{R}^n$, and the mapping function G is continuous and invertible (albeit nonlinear). Thus there will be an inverse mapping $H : V \to \Gamma$ and a nonlinear representation of Γ may be thought of as giving inverse mappings G and H as above. The compressed manifold V may be more efficiently represented as lying in $\mathbb{R}^m \times (S^1)^k$; this affords the opportunity for efficient angular feature extraction.

Bottleneck Architectures

Circular nodes fit naturally into the *bottleneck* architecture (see e.g. (Rumelhart and McClelland 1986, Oja 1991, Kramer 1991, Demers and Cottrell 1992, Krischer et. al. 1993, and Doya and Selverston 1994) which permits a non-linear principal component analysis. The use of circular nodes permits this type of architecture to discover both types of amplitude and angular features.

The model for this type of network has 3 (or more) hidden layers consisting of nodes which may be of any of the types described here, i.e., linear, circular, or sigmoidal. This network can be used to train the identity mapping using back-propagation; one of the interior layers acts as the bottleneck layer, and provides the low-dimensional representation of the data.

If one trains this network to reproduce the identity function on the data set Γ, the mapping G obtained from the input layer (of high dimension N) to the bottleneck layer (of low dimension m) will be an *invertible* and *non-linear* modeling of the data set. Its inverse H is given by the mapping from the bottleneck layer to the output layer.

The Failure of the Sigmoidal

For truly periodic phenomena, one strictly needs a circle as a target for the compression map G; an interval (which is the natural output of a sigmoidal node) is not sufficient. This is because there is no invertible mapping from the circle to any interval I. Hence if Γ is periodic with a bijective parametrization mapping $H : S^1 \to \Gamma$, and one tries to construct a bijective compression map $G : \Gamma \to I$, the composition $G \circ H : S^1 \to I$ would be a bijective mapping and therefore could not be continuous. The corresponding network will not be able to reliably generalize.

An example of this can be seen in Kramer's paper (Kramer 1991). There he discussed a pattern set consisting of 100 points on the unit circle, and proceeded to construct a mapping to the interval using the bottleneck architecture. Kramer showed that this architecture trained quite accurately on the 100 data points on the circle. However for the reasons mentioned above, it *cannot* perform well for the general point on the circle.

To demonstrate this we first trained the 2-4-1-4-2 bottleneck network (using all sigmoidal nodes) as above to high precision (average error $\approx .001$) on 20 points on the circle as shown in the upper graph of Figure 1. The removal of the discontinuity point on the circle leaves a pattern set which is topologically an interval, and therefore one can train such a network to arbitrarily high accuracy (using more hidden nodes if necessary) at the expense of missing the point of discontinuity. One clearly sees a discontinuity in the mapping functions when applied to the general point on the circle; this is illustrated in the lower graph of Figure 1.

We remark that these results are a property of the topology of the problem and are not a consequence of training strategies or architecture parameters; the discontinuity in the problem is the underlying source of the generalization failure.

The introduction of a single *circular* node at the bottleneck layer gives a continuous mapping function and removes the discontinuity, allowing the network to fully generalize to the entire circle.

In general when one uses a bottleneck layer consisting of a single circular node one expects to find the best closed curve approximation to the data, i.e., the reconstructed data is necessarily a closed curve which is an image of the circle on the bottleneck. For an example of the best closed curve approximation to the Lorenz attractor, see (Kirby and Miranda 1994b).

The Trefoil Knot

A beautiful example of the technique, which also illustrates the capability of unraveling topological complexity, is the case of pattern data lying on a trefoil knot K in $\mathbb{R}^3$. This knot is also referred to as a $(2,3)$-torus knot; it is obtained by winding around a torus twice in one dimension while going around 3 times in the other. The parametric equations are given below:

$$x(t) = \cos(2t)(2+\cos(3t))$$
$$y(t) = \sin(2t)(2+\cos(3t))$$
$$z(t) = \sin(3t)$$

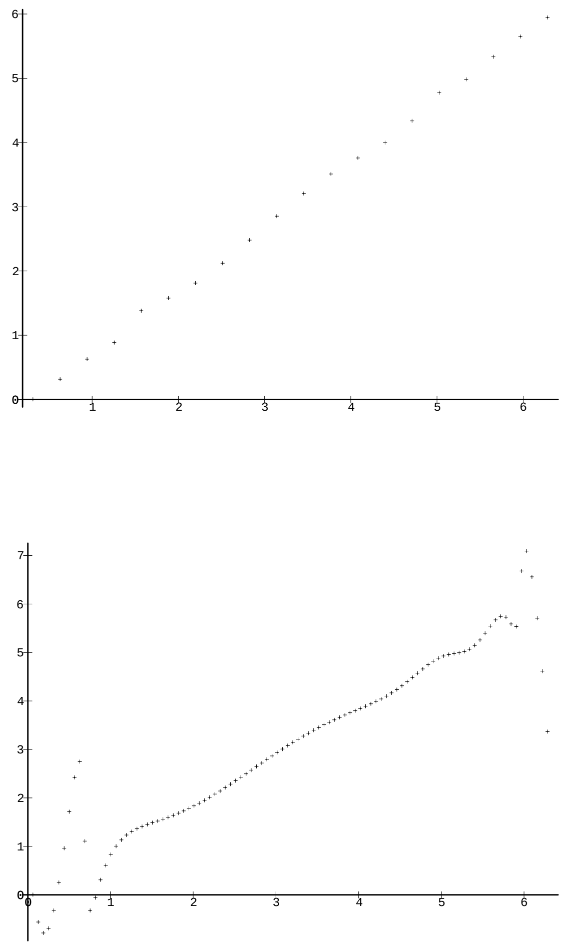

FIGURE 1. Above: The output of a single sigmoidal bottleneck node as a function of angle for 20 trained points. Below: The network trained on 20 points applied to 100 points on the circle.

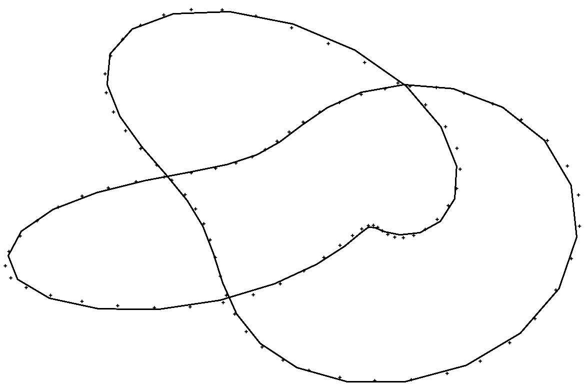

FIGURE 2. The trefoil knot in $\mathbb{R}^3$ is represented by the solid line. The output of the network modeling the trefoil knot is shown as dots.

The knot itself is intrinsically homeomorphic to a circle; the embedding into $\mathbb{R}^3$ is what makes it topologically interesting. A parametrization mapping $H : S^1 \to K$ gives such an embedding. The compression mapping $G : K \to S^1$ may be considered as the "un-knotting" of the trefoil knot.

The architecture used in our experiment was a 3-15-1-15-3 bottleneck network, all of whose nodes were sigmoidal except for the single circular node in the middle (bottleneck) layer. We trained this network to approximate the identity mapping for 1000 points on the trefoil knot K. In Figure 2 we show the trefoil knot and the result of applying the parametrization mapping H to 400 points on the unit circle, to obtain points in $\mathbb{R}^3$ which approximate the trefoil knot.

5.3. Uncovering a High-Dimensional Limit Cycle. The Kuramoto-Sivashinsky equation (K-S)

$$u_t + 4u_{xxxx} + \alpha(u_{xx} + \frac{1}{2}(u_x)^2) = 0 \tag{5.1}$$

has become a benchmark for many theories on dissipative dynamical systems and global attractors. It exhibits *low-dimensional* complex dynamics including chaos and it has been shown to possess an *inertial manifold*, i.e., there exists a smooth (C^1) low-dimensional description of its dynamics. Thus in the spirit of the Lorenz equations one expects that a small system of ordinary differential equations will model the PDE and in fact low-dimensional approximations of the inertial manifold have been constructed for Neuman boundary conditions in (Jolly et. al. 1990) and for periodic boundary conditions in (Armbruster et. al. 1989). The geometry of the solutions has also been studied and Karhunen-Loève-based low-dimensional approximations presented in (Kirby and Armbruster 1992). In this section we investigate how a bottleneck neural network with a circular node can be used to study the dynamics of the K-S equation.

In order to obtain data for the K-S equation (with periodic boundary conditions) we perform a numerical integration by means of a Galerkin approximation. This approach generates a system of ordinary differential equations by decomposing the velocity field via the expansion

$$u(x,t) = \sum_{-\infty}^{\infty} a_n(t) e^{inx} \approx \sum_{-N}^{N} a_n(t) e^{inx}. \tag{5.2}$$

Substituting (5.2) into the K-S equation (5.1) and exploiting the orthogonality of the complex exponentials one is led to a system of ordinary differential equations in the Fourier coefficients $a_n(t)$. Making use of the reality condition $a_l = \overline{a}_{-l}$ and the fact that the a_0 term decouples from the system gives

$$\dot{a}_l = l^2(\alpha - 4l^2)a_l + \frac{\alpha}{2}\sum_{n=1}^{l-1}(l-n)na_{l-n}a_n - \alpha\sum_{n=1}^{N-l}(l+n)na_{l+n}\overline{a}_n \tag{5.3}$$

where $2 \le l \le N-1$ and for $l=1$ and $l=N$ and $\dot{a}_1 = (\alpha-4)a_1 - \alpha\sum_{n=2}^{N}(1-n)n\overline{a}_{n-1}a_n$, $\dot{a}_N = N^2(\alpha - 4N^2)a_N + \frac{\alpha}{2}\sum_{n=1}^{N}(N-n)na_{N-n}a_n$. The system of equations for the Fourier coefficients is then integrated numerically. Thus the output of the simulation is the N complex Fourier coefficients $(a_1(t), \ldots a_N(t))$ and this can be returned to the original coordinate system using (5.2).

It is well established that the K-S equation undergoes a Hopf bifurcation at $\alpha = 83.75$ resulting in a localized oscillatory pattern. We simulate equation (5.1) with $\alpha = 84.25$ and $N = 10$, and the results are shown in Figure 3. Observe that the oscillations appear to be strictly localized in space traveling up the tip of the higher hump. This is considered to be a forerunner of spatio-temporal complexity encountered in spatially concentrated zones of turbulence. The origin of spatio-temporal complexity in moderate turbulence is one of the big open questions in turbulence research.

Using local analysis techniques on the system (5.3) we can show that the solution is temporally periodic near the bifurcation point. Hence, while it lies in a 20-dimensional real vector space, it is a limit cycle Γ homeomorphic to S^1. For this reason it is appropriate to attempt to map the data a circle and in this example we apply the periodic compression architecture to uncover the periodic solution which lies in $\mathbb{R}^{20}$.

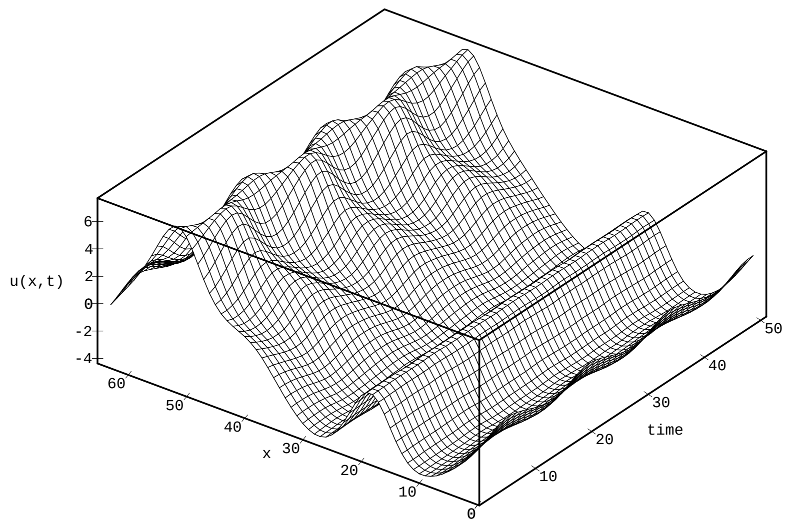

FIGURE 3. The localized oscillatory pattern of the K-S equation for $\alpha = 84.25$.

In our experiment we took $N = 10$ which was sufficient for numerical reasons and the resulting 10 complex Fourier coefficients $\{a_n(t), n = 1..10\}$ can be written as the 20-tuple $\Gamma = (x_1(t), y_1(t), \ldots, x_{10}(t), y_{10}(t))$ with $a_n = x_n + iy_n$. This time series can be viewed as a periodic data set in $\mathbb{R}^{20}$. To construct a mapping from this data to the circle we utilize a 20–15–1–15–20 bottleneck neural network with a circular node in the middle bottleneck layer. After normalizing the data by the variance we trained this network to approximate the identity mapping for 1500 points on the high dimensional cycle Γ. The network trained quickly to an average error of less than 10^{-5} indicating the excellent degree of fit between the data set and the unit circle.

Note that with this construction, the data originates in $\mathbb{R}^{20}$, is funneled through the bottleneck layer into a circle S^1, and is reproduced by the second half of the network. In Figure 4 we present the output of the second half of the neural network, for the first four Fourier coefficients a_1, a_2, a_3, a_4; this is the parametrization $H : S^1 \to \Gamma$, for these same four coefficients.

We also remark that the technique of periodic compression can be used to study a sequence of Hopf, or cycle-producing bifurcations. For example, at $\alpha = 87$ the solution is the oscillating structure of the previous regime which is now a traveling beating wave. This solution could be mapped to $S^1 \times S^1$ and would be an example of manifold decomposition outlined in Section 5.1. In addition, we observe that our analysis is not restricted to a local region about the bifurcation value.

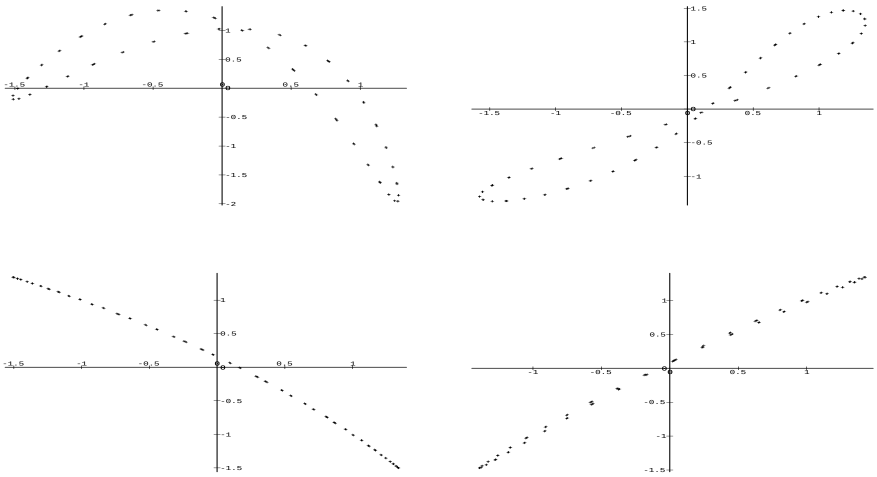

FIGURE 4. The output of the neural network corresponding to the first four complex Fourier coefficients.

The parametrization techniques presented for the analysis of data of dynamical systems such as the K-S equation can also be used as the basis for approximate inertial manifold constructions. In particular, the inverse mappings G and H constructed by the neural network can also be used to transport the dynamical system or flow from the attractor embedded in the high-dimensional space to the model space built as the image of the system in the low-dimensional bottleneck layer. Therefore we have not only a much simpler model for the attractor, but also a potentially simpler model for the differential equation. This work has begun in (Kirby and Miranda 1994a) and will be discussed in terms of the circular node in a forthcoming article (Kirby and Miranda 1995).

6. Summary

In this paper we present an implementation of a circular node for a neural network capable of storing and transmitting angular information. It can be used in conjunction with standard sigmoidal nodes to create networks utilizing both types of node. We present the equations for forward and backward propagation, which have the same general form as the standard implementation. Thus the incorporation of circular nodes and the training of networks utilizing them is not radically different than existing algorithms, and can be easily implemented.

Finally, we present several examples of the use of circular nodes in pattern analysis. We close with a non-trivial reconstruction of the limit cycle of the Kuramoto-Sivashinsky equation whose original geometry lies in a space of 20 dimensions.

Our work suggests that the construction and implementation of other useful types of nodes is both feasible and potentially useful. For example, mappings to 2-dimensional manifolds may be better served by the construction of nodes carrying spherical or multi-toroidal information (Hundley et. al. 1995).

References

1. Armbruster, D., Guckenheimer, J. & Holmes, P. (1989). Kuramoto-Sivashinsky Dynamics on the Center-Unstable Manifold. *SIAM J. Appl. Math.*, **49**, 676.
2. Demers, D. & Cottrell, G. (1992). in Hanson, S.J., J.D. Cowan, J.D, & Giles, C.L (eds). *Neural Information Processing Systems* 5, Morgan Kauffman, p. 582 (1992).
3. Doya, K. and Selverston, A.I. (1994) Dimension Reduction of Biological Neuron Models by Artificial Neural Networks, *Neural Computation*, **6**, 696-717.
4. Devijver, P.A. & Kittler, J. (1982). *Pattern Recognition: A Statistical Approach.* Prentice Hall.
5. Gislén, L., Peterson, C., and Söderberg, B. (1992). Neural Computation, Vol. 4, 737-745.
6. Hertz, J., Krogh, A. & Palmer, R.G. (1991). *Introduction to the Theory of Neural Computation.* Addison Wesley.
7. Hundley, D., Kirby, M. and Miranda, R. (1995) Spherical Nodes in Neural Networks and Applications (in preparation).
8. Jolly, M.S., Kevrekidis, I.G., and Titi, E.S. (1990). *Approximate inertial manifolds for the Kuramoto-Sivashinsky equation: Analysis and computations*, Physica 44 D, 38.
9. Kirby, M. & Armbruster, D. (1992). Reconstructing phase-space from PDE simulations. *Z. angew Math Phys* **43**, 999-1022.
10. Kirby, M. & Miranda, R. (1994a) The Nonlinear Reduction of High-Dimensional Dynamical Systems via Neural Networks. *Physical Review Letters.* **72**, No. 12, 1822-1825.
11. Kirby, M. & Miranda, R. (1994b) The Remodeling of Chaotic Dynamical Systems. In: Intelligent Engineering Systems Through Artificial Neural Networks, Vol. 4, C. H. Dagli, B. R. Fernandez, J. Ghosh, and R. T. S. Kumara, Eds. Proceedings of the ANNIE '94 conference, St. Louis, MO, ASME Press, New York, 831-836.
12. Kirby, M. & Miranda, R. (1995) The Reduction of Dynamical Systems. (in preparation).
13. Kramer, M.A. (1991). Nonlinear Principal Component Analysis Using Autoassociative Neural Networks. *AIChE Journal* **37**, No. 2, 233-243.
14. Krischer, K., Rico-Martinez, R., Kevrikidis, I.G., Rotermund, H.H., Ertl, G. and Hudson, J.L. (1993). Model Identification of a Spatiotemporally Varying Catalytic Reaction. *AIChE Journal*, **39**, No. 1, 89-98.
15. Lapedes, A.S. & Farber, R. (1987). Nonlinear signal processing using neural networks: Prediction and system modeling. Technical Report LA-UR-87-2662, Los Alamos National Laboratory, Los Alamos, New Mexico.
16. Oja, E. (1991). Data Compression, Feature Extraction, and Autoassociation in Feedforward Neural Networks. *Artificial Neural Networks*, 737-745.
17. Rumelhart, D. E. and McClelland, J. L. (1986). *Parallel Distributed Processing.* MIT Press, Cambridge, MA.
18. Zemel, R.S., Williams, C.K.I. & Mozer, M.C. (1993). Lending Direction to Neural Networks. *to appear, Neural Networks.*

Dept. of Mathematics, Colorado State University, Ft. Collins, CO 80523
E-mail address: kirby@ritz.math.colostate.edu

Dept. of Mathematics, Colorado State University, Ft. Collins, CO 80523
E-mail address: miranda@riemann.math.colostate.edu