On the Inverse of Hopfield-Type Dynamical Neural Networks

Angela Hodge, Wei Zhen, and Robert W. Newcomb
Microsystems Laboratory, Electrical Engineering Department
University of Maryland, College Park, MD 20742 USA
http://www.ee.umd.edu/newcomb/mslab.html

Abstract:

A technique is given for finding the system inverse to an Hopfield class of continuous time dynamical artificial neural networks, that is, for finding the system which yields the equivalence class of inputs which lead to a given output. This is accomplished by applying the theory of inverse semistate linear systems to the linear part and directly inverting the activation functions. An example is given for a two-input two-output degree two (two neuron) system. The results could be of use in finding the set of patterns which fall into different classes of a neural network dynamic pattern classifier.

I. Introduction

Since its introduction the continuous time Hopfield type artificial neural network (ANN) [1] has been of interest to those attempting to make electronic architectures which classify using principles similar to those of neuro-biological systems. Basically its structure is general enough that it contains most other common ones as special cases; consequently, it is a good starting point for any theoretical development. Here we use it to begin a development of inverse ANNs. For that we assume as given an Hopfield continuous time ANN with n neurons, n^2 fixed weights, and n known (nonlinear) activation functions. We also assume that there are n inputs and that all of the outputs of the n neurons are available for observation. Correspondingly, the problem is to determine the inputs from the observations on the n neuron outputs.

Recently there has been an effort to create inverse systems [2][3][4], especially for chaotic ones where the input can be coded into a chaotic signal and then decoded via an inverse system [4] Our results are similar in that our system has the ability to retrieve an original input to an Hopfield network by observation of the output in real time.

II. The Basic Inverse Design

The Hopfield continuous time semistate (time-domain) system equations take the form

$$C dx/dt = Wv + I_b + u, \qquad x(0)=x_o \quad (1a)$$
$$0 = v - f(x) \quad (1b)$$
$$y = v \quad (1c)$$

where $[x^T,v^T]^T$ is the semistate (column) 2n-vector [here superscript T = transpose], I_b is the n-vector dc bias, u is the external n-vector input, $f(\cdot)$ is the n-vector of monotonic (strictly increasing) nonlinear activation functions, and y is the n-vector of neuron outputs. In the semistate itself, v is the n-vector of neuron outputs that are fed back into the neurons via the nxn weight matrix W and x is the n-vector of inputs to the activation portion of the neurons, with its value, x_o, given at t=0. The nxn matrix C is included to handle capacitors in a physical implementation and is taken to be nonsingular.

Given C, W, I_b, $f(\cdot)$, x_o, and $y(\cdot)$ over the interval $[0, t_f]$, t_f=final time, we desire to determine $u(\cdot)$ over the same time interval $[0, t_f]$.

Toward solving this problem we design a system inverse to the Hopfield ANN of Equations (1), which naturally we call an Inverse Hopfield ANN; we will also refer to the original Hopfield ANN as the forward system. Thus, we are interested in solving for the forward system input u as (approximated by) the output of the inverse system, y_{inv}, for which we put the forward system output y as the inverse system input, u_{inv}=y. Since each component of $f(\cdot)$ is assumed to be strictly monotone increasing, each component has a functional inverse which will be needed in the sequal, so we write $f^{(-1)}(\cdot)$ for the n-vector of inverses of activations. Next we rearrange Equations (1) into the form

1058-6393/96 $5.00 © 1996 IEEE
Proceedings of ASILOMAR-29

$$u = C(dx/dt) - Wv - I_b \quad (1'a)$$
$$x = f^{(-1)}(v) \quad (1'b)$$
$$v = y \quad (1'c)$$

Using

$$u \approx y_{inv},\ y = u_{inv} \quad (2a,b)$$

we rewrite these equations as

$$v_{inv} = u_{inv} \quad (3a)$$
$$x_{inv} = f^{(-1)}(v_{inv}) \quad (3b)$$
$$y_{inv} = C(dx_{inv}/dt) - Wv_{inv} - I_b \quad (3c)$$
$$x_{inv}(0) = x_o \quad (3d)$$

Figure 1 shows in one figure the forward and the inverse Hopfield ANNs in a Simulink block diagram. In Figure 1 the top half gives an implementation, after solving for x by integrating Eq. (1a) and inverting C, of the forward ANN and the bottom half gives an implementation of the inverse ANN (the numbers are those for the example of the next section).

III. Example

By way of illustration of the basic theory we include a simple example which illustrates most of the points. Here we take a two neuron Hopfield ANN as per the upper half of Figure 1 in which we choose $C=I_2=2x2$ identity, $W=[w_{ij}]$ with $w_{11}=w_{22}=0$, $w_{12}=w_{21}=-0.8$, $I_b=[0.2,-0.3]^T$, $x_o=[0.1,-0.1]^T$ along with $f(x)=[(2/\pi)\arctan(x)]$ for all of the activation nonlinearities (the Mux of Simulink combines the scalar functions into a vector). To show the capabilities we choose the two inputs to be sines of different amplitudes and frequencies, the first component being of amplitude 1 and frequency 1 and the second component being of amplitude 2 and frequency 10, as shown in Figure 2a). In Figure 2b) we show the output, y, of the forward ANN and we take that output to be the input, u_{inv}, of the inverse ANN. In Figure 2b) the upper curve is for the first component and it is seen how both components eventually saturate. The inverse ANN is shown in the bottom half of Figure 1 where the inverses of the activation functions are $\tan((\pi/2)u)$. An important observation is to be made on the saturation of the two components of y as time increases. Even though these saturate, Figure 2c) shows the output, y_{inv}, of the inverse ANN; when compared with the input to the forward ANN it is seen that to a very good approximation $y_{inv}=u$. In fact the resemblance of y_{inv} to u is much more striking than we first imagined, and surprisingly it is retained over a large range of inputs and bias vectors. From our experimentation this system is quite robust. However, the accuracy naturally depends upon the step size used in the simulations; with too large of a step size the output yinv(t) is not such a good track of the input u(t). We used a maximum of 0.1 and a minimum of 0.00001 with a tolerance of 1e-5 and Runge-Kutta 5 of Simulink.

IV. Discussion

Given the Hopfield continuous time dynamic ANN we have shown how it is possible to formulate a solution to the problem of determining the input given the output, this resulting by passing the output of the forward network through the inverse network described here. Analysis on a two neuron Hopfield network and its inverse is included and the resulting output from the inverse system is seen to be closely equivalent to the pattern of the original network's input. We have tried the system with many other inputs, for example for sines with 100/1 frequency differences and 10/1 amplitude differences with equally encouraging results. After running the systems for long periods the y(t) approach constants but even then the $y_{inv}(t)$ is virtually indistinguishable from u(t). As the normal purpose of an Hopfield network is to lead to constants of saturation, it is fascinating that one can use these saturating outputs to reproduce highly variable inputs. The results are so encouraging that further studies are in order to determine what are the limitations on the retrieval properties of the forward - inverse ANN pairs. Certainly this research shows that the inverse system can be used for determining the nature of inputs to the forward system; thus, it may be possible to use the overall system for coding.

V. Acknowledgments

The authors wish to express their appreciation for the Sloan Foundation grant for undergraduate Women in Engineering Research Fellowships which supported the first two authors during the academic year and to the University of Maryland for a Senior Summer Scholarship to support the second author during the summer of 1995.

VI. References

[1]. J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," Proceedings of the National Academy of Sciences of the United States of America, Vol. 81, No. 10, May 1984, pp. 3088 - 3092.

[2]. U. Feldmann, M. Hasler, and W. Schwarz, "Communication by Chaotic Signals: the Inverse System Approach," Proceedings of the IEEE International Symposium on Circuits and Systems, April 30 - May 3, 1995, Seattle pp. 680 - 683.

[3]. F. L. Lewis, "A Survey of Linear Singular Systems," Circuits, Systems, and Signal Processing, Vol. 5, No. 1, 1986, pp. 3 - 36.

[4]. R. W. Newcomb and B. Dziurla, "Some Circuits and Systems Applications of Semistate Theory," Circuits, Systems, and Signal Processing, Vol. 8, No. 3, 1989, pp. 235 - 260.

[5]. T. L. Carroll and L. M. Pecora, "Synchronizing Nonautonomous Chaotic Circuits," IEEE Transactions on circuits and Systems - II, Vol. 40, No. 10, October 1993, pp. 646 - 650.

Figure 1: Forward (top) and Inverse (bottom) Hopfield ANNs

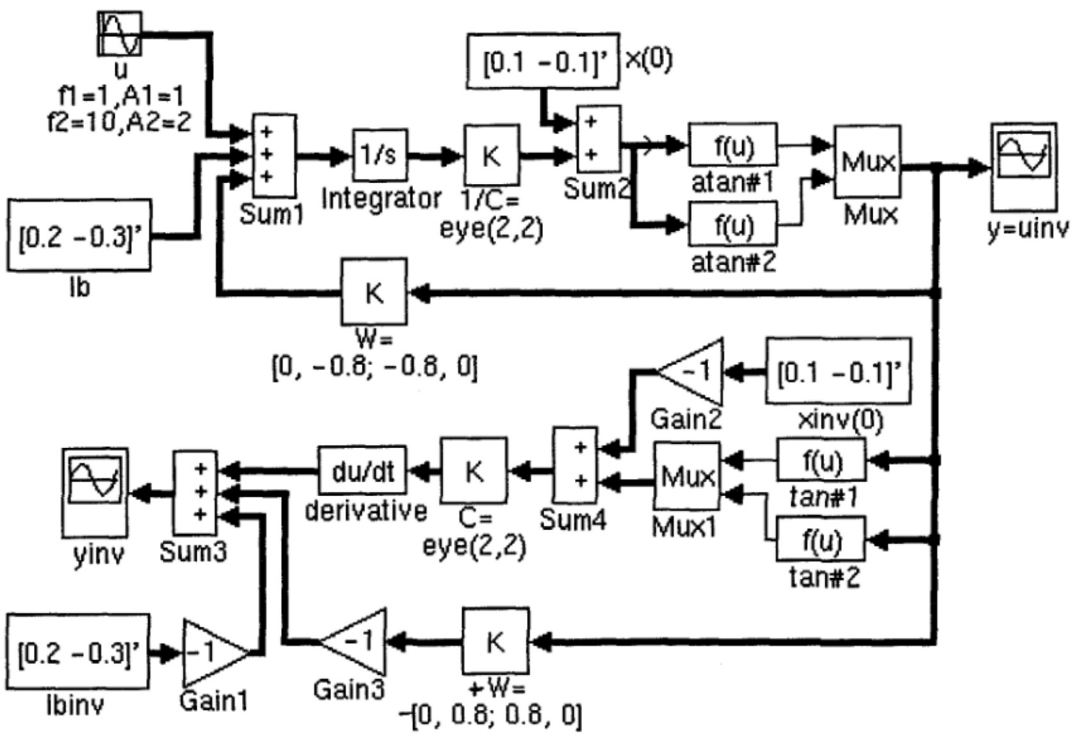

Figure 2a): Forward input, u(t)

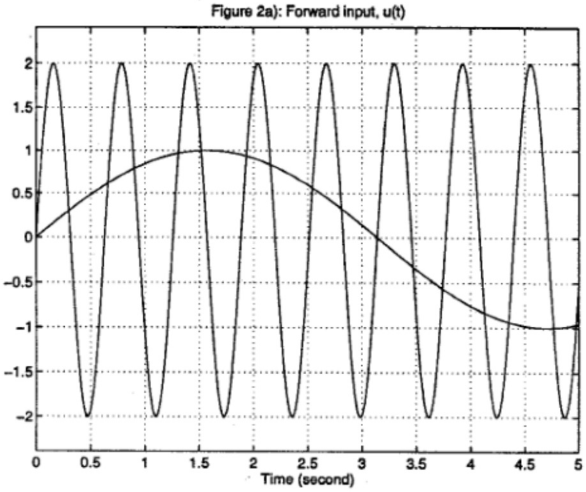

Figure 2b): Forward output, y(t)

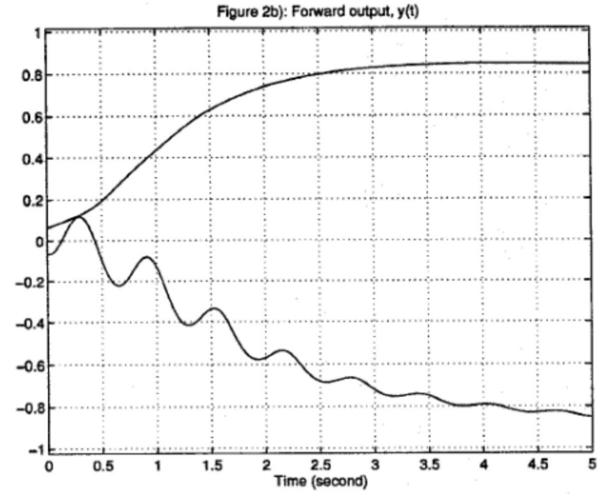

Figure 2c): Inverse output, yinv(t)