12-1-1992

An Embedding Selection Algorithm for Chaotic Dynamical Systems

W. Hsu
Purdue University School of Electrical Engineering

L. S. Hsu
Purdue University School of Electrical Engineering

M. F. Tenorio
Purdue University School of Electrical Engineering

AN EMBEDDING SELECTION ALGORITHM FOR CHAOTIC DYNAMICAL SYSTEMS

W. Hsu
L. S. Hsu
M. F. TENORIO

TR-EE 92-51
DECEMBER 1992

SCHOOL OF ELECTRICAL ENGINEERING
PURDUE UNIVERSITY
WEST LAFAYETTE, INDIANA 47907-1285

An Embedding Selection Algorithm for Chaotic Dynamical Systems

W. Hsu L. S. Hsu M. F. Tenorio

Parallel Distributed Structures Laboratory
Department of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

Abstract

The selection of an embedding scheme is an important step in the modeling and prediction of chaotic dynamical systems. Theoretical work in embedding selection abounds in the literature. However in neural network research, mostly compute intensive methods for embedding selection exist. In this paper, we propose a novel embedding selection scheme based on cluster analysis. A neural network implementing this method is described and demonstrated on the Mackey-Glass chaotic time series. The result of the method agrees with the embedding schemes used by researchers in neural networks. In addition, other new embedding schemes have been Found and they also enable this chaotic time series to be predicted accurately.

Keywords: Embedding Selection, Chaotic Time Series Prediction.

1 Introduction

The choice of an embedding scheme is an important step in the modeling and prediction of any chaotic dynamical systems. The modeling and prediction of chaotic systems has attracted much recent attention due to the discovery of the presence of chaos in many interesting phenomenal previously thought to be random. Examples are these systems include the economic systems[1, 2], weather[3] and a number of physiological processes[4].

The two step to chaotic time series prediction are the feature extraction and the pattern learning steps. When the chaotic time series assumption can be made, the feature extraction step is equivalent to specifying an embedding scheme.

Specifying an embedding scheme is equivalent to identifying the set of features necessary to characterize the system. In physics, this process is sometimes referred to as state-space reconstruction. A large body of theoretical work has been done in this area. A comprehensive summary can be found in [5]. The most cited work among neural network researchers dealing with chaotic dynamical system is perhaps the work by Takens[6]. Takens showed that a chaotic time series $x(t)$ can be predicted T step in the future by using only m number of equally spaced past samples of the chaotic time series itself as follows :

$$x(t+T) = \mathcal{F}\{x(t), x(t-\Delta), x(t-2\Delta), \ldots, x(t-(m-1)\Delta)\} \quad (1)$$

where $\mathcal{F}$ is nonlinear but continuous under the suitable assumptions[6]. Taken's theorem does not, however, provide a way of constructing $\mathcal{F}$. An embedding scheme for a chaotic time series is given by the 3-tuple

$$\Pi = [m, \Delta, T]. \quad (2)$$

Equation 1 says that a chaotic time series $x(t)$ can be predicted T time step in advance: using only m past samples of $x(t)$ spaced Δ distance apart.

For a specific embedding scheme, the chaotic time series prediction problem becomes that of associating the following pairs of **X** and Y as follows:

$$\mathbf{X}_1 \rightarrow \mathbf{Y}_1; \mathbf{X}_2 \rightarrow \mathbf{Y}_2; \ldots; \mathbf{X}_n \rightarrow \mathbf{Y}_n.$$

X's are called the state vectors and **Y's** the desired predictions. The process of associating the above is referred to as the pattern learning step.

Most of the neural network research on the prediction of chaotic time series are focused on developing pattern learning algorithms. However, the embedding scheme chosen is crucial to the accuracy of the prediction task. A handful of researchers have suggested methods for finding an embedding scheme empirically. Unfortunately, many of these approaches are computational intensive. For example, Casdagli[7] and Mead[8] computed the actual prediction error of their neural networks resulting from an enumeration of different embedding schemes and the enumeration which resulted in the smallest actual prediction error is selected as the embedding scheme for the chaotic time series. Other researchers have proposed different computational approaches to this problem[9]. In this paper, we propose a novel embedding selection procedure that can be use to obtain an initial embedding scheme for a chaotic time series. We demonstrate the proposed algorithm on the Mackey-Glass chaotic time series.

The rest of the paper is organized as follows: The set of a number of past samples of the chaotic time series is called the delayed vectors a. The embedding scheme for a chaotic time series selects a subset of $\mathbf{z}$ as the state vector X described earlier. Section 2 describes how the delayed vectors $\mathbf{z}$ is computed from a chaotic time series. The **SupNet** architecture as well as its learning algorithms are also presented in this section. Section 2 defines the notion of data inconsistencies as a criterion and how it can be computed with **SupNet.** Section 3 explains the search procedure for finding an embedding scheme corresponding to local minimal in the criterion surface. Section 4 applies the procedure to the Mackey-Glass chaotic time series and shows that the embedding schemes used in the literature do coincide with the points of minimum data inconsistency. Section 5 explores new embedding schemes correspond to other regions of minimal data inconsistency and shows that these newly found embedding schemes do result in accurate prediction of the chaotic time series.

2 The Supervised Clustering Network (SupNet)

Given a specific embedding scheme Π_i, the chaotic time series prediction problem can be reformulated as a learning problem associating the following pairs of state vectors X and its corresponding prediction Y. Each pair **(X,**Y) is called a training pattern.

The Supervised Clustering Network or **SupNet** performs clustering in a hierarchical fashion. During the first stage, the training patterns are clustered with respect to **Y's**. These clusters are then subsequently further subdivided by clustering each of the training patterns within each of these clusters with respect to their X's.

The item to be discussed next is the computation of the delayed vectors $\mathbf{z}$ from which the embedding scheme selects the appropriate features.

2.1 The Input Vector

Given a chaotic time series

$$x_1, x_2, \ldots, x_i, \ldots \quad (3)$$

The following steps are taken to construct a set of delayed vectors $\mathbf{z}$.

1. Determine the region of interest by assigning the maximum values to T_{max}, Δ_{max} and m_{max} where these are the range of values for each of the three parameters m, Δ and T respectively. Then the length L of the delayed vectors $\mathbf{z}$ is

$$L = m_{max}\Delta_{max} + T_{max}. \quad (4)$$

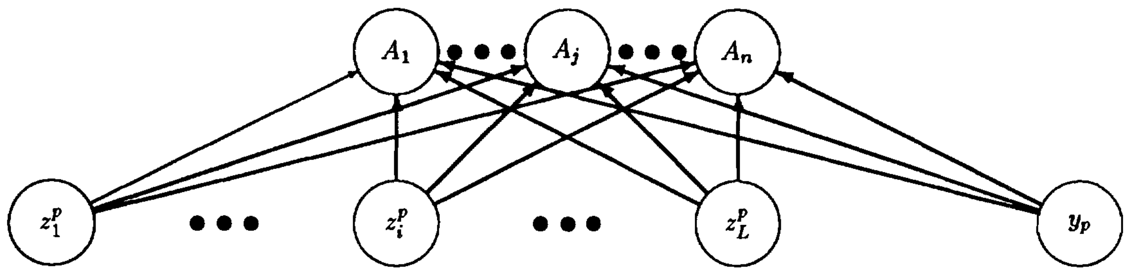

Figure 1: The SupNet Architecture.

2. Let the number of training patterns be n.
3. Compute $\mathbf{z}$ from the chaotic time series $x(t)$ as follows :

$$\begin{array}{cccccccccccc} (z_1^1 & z_2^1 & \ldots & z_L^1 & y_1) & = & (x_1 & x_2 & \ldots & x_L & x_{L+1}) \\ (z_1^2 & z_2^2 & \ldots & z_L^2 & y_2) & = & (x_2 & x_3 & \ldots & x_{L+1} & x_{L+2}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots \\ (z_1^p & z_2^p & \ldots & z_L^p & y_p) & = & (x_p & x_{p+1} & \ldots & x_{L+p-1} & x_{L+p}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots \\ (z_1^n & z_2^n & \ldots & z_L^n & y_n) & = & (x_n & x_{n+1} & \ldots & x_{L+n-1} & x_{L+n}) \end{array}$$

4. The delayed vectors $\mathbf{z}^p$ augmented with their corresponding values y_p **are** then used as inputs to **SupNet** described next.

2.2 Network Architecture

The **SupNet** architecture is shown in Figure 1.

The network consists of 2 layers. The first layer is the input layer. It consists of L + 1 nodes. The first L nodes represent the components of the delayed vector **z.** The last node represents the value of y.

The second layer is the cluster layer. Its size is determined dynamically by the learning algorithm described in the next subsection. The number of nodes corresponds to the number of clusters needed to classify the values of y to within a given accuracy ϵ_y.

The weights connecting a given cluster node c to the input nodes form the components of the weight vector $\mathbf{W}^c$. The values of these weight vectors are determined by the learning algorithm which will be described in the next subsection.

When input vector $[\mathbf{z}^p, y_p]$ is presented, the activation at node c is defined as

$$A_c = (y_p - W_{L+1}^c)^2. \quad (5)$$

2.3 The Learning Algorithm

Learning proceeds in the following two stages. During the first stage, the state vector $\mathbf{z}^p$ is taken to be the zero vector and only the value of y_p is presented. We follow the algorithm used in *ClusNet*

[10] to determine the (L + 1)-th component of the weight vectors for all the clusters. The first L components of W remain at zero.

During the second stage, the n input vectors are presented one at a time. Assuming that when input vector $[z^p, y_p]$ is presented, the c-th cluster node has the lowest activation among other cluster nodes. We say that the c-th node is the winning node and the first L components of its weight vector is updated to:

$$W_i^c = \frac{1}{n_c}\{(n_c - 1)(W_i^c + z_i^p)\}, \quad 1 \leq i \leq L \tag{6}$$

where n_c is the number of vectors belonging to cluster c, after the new vector z^p has been added.

When all the n input vectors have been presented, the weight vectors W are all known.

3 Definition of Data Inconsistency

Using SupNet, the training patterns

$$[z^p, y_p]$$

are clustered with respect the values of y_p. These clusters are called supervised clusters.

We expect better prediction results using SupNet if the training patterns are clustered around their respective supervised cluster centers. This condition can be approximately enforced if

Criterion A The average root-mean-square distance between delayed vectors and their respective cluster centers, Distv, is at a minimum and

Criterion B The average root-mean-square distance between centers of clusters, Distc $\geq 2Distv$.

When both the above criteria are satisfied, a vector belonging to a cluster c is unlikely to be mis-clustered into a different cluster d. In this case, it is less likely for input patterns to be predicted to be in a "wrong" class and thus the resulting prediction is more accurate. When this occurs, we say that the values of z are consistent with the values of y.

3.1 Criterion A

We define a quantity Ω'_j for the j-th component of the delayed vector:

$$\Omega'_j = \sum_{c=1}^{N}\sum_{p=1}^{n_c}(W_j^c - z_j^p)^2 \tag{7}$$

For the complete vector, we can define an average value:

$$\Omega' = \frac{1}{L}\sum_{j=1}^{L}\Omega'_j \tag{8}$$

If we choose a training set of size n, it is clear that

$$\text{Distv} = \frac{\sqrt{\Omega'}}{n} \tag{9}$$

where n is the size of the training set. Enforcing Criterion A is equivalent to minimizing the quantity O'.

Ω' can be reduced by an appropriate choice of embedding scheme, Π. The latter allows us to discard offending components and keeping components which are consistent with the values of y.

3.2 Criterion B

For the j-th component of the delayed vector, we define a quantity Ψ_j such that:

$$\Psi_j = \sum_{c=1}^{N} \sum_{d=c+1}^{N} (W_j^c - W_j^d)^2 \tag{10}$$

$$\Psi = \frac{1}{L} \sum_{j=1}^{L} \Psi_j. \tag{11}$$

The quantity is related to ***Distc*** as follows :

$$Distc = \frac{\sqrt{\Psi}}{\frac{N(N-1)}{2}} \tag{12}$$

where $\frac{N(N-1)}{2}$ is the number of inter-cluster-centers distances computed in Equation ***10.***

We interpret Criterion B as saying that if the following condition is true for a component j,

$$Distv_j > \frac{1}{2} Distc_j \tag{13}$$

then component j should be excluded from the embedding scheme by assigning a large number $MAXFLOAT$ to Ω'_j. This action signals to the subsequent search algorithm that component j is undesirable and should not be selected.

3.3 Computation of the State Vector

The state vectors which are used for prediction are subsets of the delay vectors z using the embedding Π found by the procedure described above. The components of the state vectors are chosen to minimize the quantity R. The state vector X is extracted from z as follows :

$$X_i = z_j \tag{14}$$

where $j = L+(i-m)A-T$ and $1 \leq i \leq m$. With this definition, we can define the ***data inconsistency*** in X with respect to these k clusters as:

$$\Omega = \frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{N} \sum_{p=1}^{n_k} (W_i^k - X_i^p)^2 \tag{15}$$

Similarly, we define a consistency measure for each component of the state vector:

$$\Omega_i = \sum_{k=1}^{N} \sum_{p=1}^{n_k} (W_i^k - X_i^p)^2 \tag{16}$$

In terms of these, we can write:

$$\Omega = \frac{1}{m} \sum_{i=1}^{m} \Omega_i \tag{17}$$

$$= \frac{1}{m} \sum_{i=1}^{m} \Omega'_{L+(i-m)\Delta-T} \tag{18}$$

3.4 The Embedding Selection Procedure

The procedure for choosing an embedding scheme is as follows:

1. Clusters the training patterns with respect to their **Y's** to form supervised clusters.
2. Compute Ω'_j and Ψ_j for all components of the delayed vector. Note that these are independent of any embedding scheme.
3. For each value of j for which Equation 13 is true, assign a large number $MAXFLOAT$ to Ω'_j.
4. For each value of $\Pi = [m, \Delta, T]$, compute Ω according to Equation 18
5. Look for Π with a corresponding minimum value of R.

4 Study of Existing Embedding Schemes

The above is applied to the Mackey-Glass chaotic time series[4] with parameter $\tau = 17$. We choose the region to be explored by setting $m_{max} = 6$, $T_{max} = 100$ and $\Delta_{x} = 10$. The size of the training set is chosen to be $n = 500$.

Following the *ClusNet* algorithm as described in [10], the predictions are computed and recorded. **ClusNet** algorithm proceeds by clustering the state vectors and prediction for input X is given by the cluster node which is most similar to X. Using **ClusNet**, State Vectors that are made up of components with the least inconsistencies as defined in the previous section are less likely to be predicted incorrectly. The prediction results are reported in normalized root mean square values which is defined as follows. If the true Y values of the prediction set is

$$\mathbf{Y}^j \quad j = t : n \tag{19}$$

and let

$$\mathbf{L} = [\mathbf{Y}^j \quad j = t : n] \tag{20}$$

and the predicted Y is

$$\hat{\mathbf{Y}}^j \quad j = t : n \tag{21}$$

and let

$$\mathbf{L}' = [\hat{\mathbf{Y}}^j, j = t : n] \tag{22}$$

then the **nrmse** of the predicted Y with respect to the true Y is

$$\mathbf{nrmse}(\mathbf{L}, \mathbf{L}') = \frac{\sqrt{mean((\mathbf{L} - \mathbf{L}')^2)}}{\sigma(\mathbf{L})} \tag{23}$$

where $\sigma(\mathbf{L})$ denotes the standard deviation of the vector L. The mean operation in Equation 23 makes the measure independent of the length of vector L. The normalization of the quantity in Equation 23 removes the dependence on the dynamic range of the data. From Equation 23, if the mean of L is used as the prediction for L, i.e., $\mathbf{L}' = \mathbf{mean}(\mathbf{L})$, then,

$$\mathbf{nrmse}(\mathbf{L}', \mathbf{L}) = 1.0. \tag{24}$$

The next two subsections examine two commonly used embedding schemes.

4.1 Embedding characterized by $\Pi = [4, 6, 85]$

This embedding scheme have been used in [11] among others. We calculate the value of Ω in the vicinity of $\Pi = [4, 6, 85]$. The result is shown in Table 1.

A minimum value of Ω does occur in the vicinity of $\Pi = [4, 6, 85]$.

Table 1: The value of Ω in the vicinity of $\Pi = [4, 6, 85]$. This embedding is a popular choice for recent neural network researchers.

A		4	5	6	7	8
m	T					
3	83	21.1596	20.2701	18.7856	17.0718	15.5826
	84	21.6966	20.3681	18.6171	16.8678	15.5747
	85	21.8409	20.1556	18.2574	16.6065	15.6092
	86	21.5660	19.6304	17.7195	16.2932	15.6593
	87	20.8799	18.8201	17.0359	15.9399	15.6991
4	83	19.7044	17.3807	15.6253	15.1748	15.6834
	84	19.5063	17.0828	15.6248	15.5339	16.1956
	85	19.0480	16.7026	**15.6432**	15.8947	16.6714
	86	18.3527	16.2590	15.6606	16.2163	17.0690
	87	17.4666	15.7771	15.6600	16.4694	17.3530
5	83	17.2089	15.4647	15.6974	16.6028	16.8898
	84	16.8737	15.5630	16.1115	17.0005	17.0619
	85	16.4674	15.6685	16.4862	17.3036	17.1528
	86	16.0117	15.7591	16.7880	17.4815	17.1552
	87	15.5334	15.8188	16.9909	17.5187	17.0698

4.2 Embedding characterized by $\Pi = [4, 6, 6]$

The second most commonly used embedding is perhaps $\Pi = [4, 6, 6]$. This embedding scheme has been used in [12] among others. The value of Ω in this vicinity is shown in Table *2*.

It can be seen that the point is not at a minimum. Instead a nearby minimum occurs at $\Pi = [4, 8, 1]$. Using this value of Π, predictions were made using *ClusNet*. The results are shown in Table *4*. The prediction obtained by *ClusNet* at the traditional $T = 6$ is not as good as those obtained at the nearby minimal located by our method. (See Table 3).

5 New Embedding Schemes

The same method is used to explore other regions and the following embedding were found. In Table *4*, a new minimal is located at $\Pi = [4, 6, T = 63]$. Good prediction were obtained with *ClusNet* with this embedding as shown in Table 5.

In Table *6*, we located a new minimal at $\Pi = [3, 8, 63]$. This particular embedding has not been used in the literature. We show that with this particular embedding, good prediction can be obtained with *ClusNet* in Table *7*.

6 Conclusions

In this paper, we propose a novel embedding scheme selection procedure for chaotic time series based on the criterion of data inconsistencies computed from the supervised clusters. This systematic procedure can be used in practice to provide an initial embedding scheme because of its simplicity. The proposed procedure was demonstrated on the Mackey-Glass chaotic time series. Experiments show that the embedding schemes used by neural network researchers are identified by the proposed algorithm. Furthermore, two new embedding schemes for the Mackey-Glass chaotic time series are found using this procedure. These embedding schemes also allow accurate prediction of the Mackey-Glass chaotic time series. We are currently exploring the use of this technique on several other chaotic time series.

Table 2: The value of Ω in the vicinity of $\Pi = [4,6,6]$. This embedding scheme has been widely used by neural network researchers working on the prediction problem.

Δ		4	5	6	7	8	9
m	T						
3	1	10.7241	12.3068	13.1095	13.2283	12.8139	12.1065
	2	12.6054	13.7345	14.0986	13.8299	13.1343	12.3010
	3	14.4674	15.1181	15.0490	14.4362	13.5386	12.6709
	4	16.1844	16.3606	15.8936	15.0077	14.0013	13.1827
	5	17.6466	17.3791	16.5761	15.5054	14.4845	13.7778
	6	18.7696	18.1124	17.0534	15.8938	14.9419	14.3850
	7	19.5001	18.5240	17.3001	16.1459	15.3272	14.9371
	8	19.8180	18.6057	17.3128	16.2489	15.6045	15.3866
4	1	13.5018	13.8512	13.2420	12.4536	**12.0856**	12.2120
	2	14.7115	14.5273	13.6189	12.7893	12.4820	12.6728
	3	15.8226	15.1519	14.0322	13.2272	13.0090	13.2885
	4	16.7594	15.6803	14.4526	13.7310	13.6331	14.0209
	5	17.4614	16.0793	14.8489	14.2603	14.3105	14.8156
	6	17.8905	16.3298	15.1901	14.7754	14.9917	15.6103
	7	18.0349	16.4254	15.4503	15.2415	15.6292	16.3443
	8	17.9084	16.3711	15.6159	15.6338	16.1857	16.9690
5	1	14.1826	13.2773	12.5738	12.7206	13.5257	14.4462
	2	14.8198	13.6477	13.0001	13.2597	14.0988	14.8947
	3	15.3860	14.0550	13.5098	13.8888	14.7505	15.3977
	4	15.8435	14.4642	14.0677	14.5706	15.4417	15.9214
	5	16.1655	14.8436	14.6368	15.2655	16.1250	16.4249
	6	16.3383	15.1688	15.1803	15.9335	16.7498	16.8669
	7	16.3614	15.4243	15.6673	16.5365	17.2702	17.2130
	8	16.2467	15.6026	16.0785	17.0423	17.6532	17.4418

Table 3: Prediction Performance of *ClusNet* in the vicinity of $\Pi = [4,6,6]$. The Prediction Accuracy is reported in nrmse. Num Cluster refers to the number of clusters that are allocated by **ClusNet**.

m	T	A	Prediction	Num Cluster
4	6	6	0.05600	54
4	1	8	0.04900	54

Table **4:** The value of Ω in the vicinity of $\Pi = [4,6,64]$. This embedding scheme has not been suggested in the literature.

Δ		4	5	6	7	8
m	T					
3	62	21.0946	19.7294	18.3066	17.1004	16.3822
	63	20.5122	19.0346	17.6432	16.6170	16.2077
	64	19.7012	18.1897	16.9146	16.1504	16.1006
	65	18.6960	17.2357	16.1634	15.7348	16.0662
	66	17.5462	16.2251	15.4416	15.4030	16.1002
4	62	18.7258	16.9818	16.2883	16.8698	17.9399
	63	17.9596	16.4438	16.1917	17.0997	18.1453
	64	17.1034	15.9335	**16.1540**	17.3028	18.2528
	65	16.2067	15.4851	16.1671	17.4544	18.2596
	66	15.3275	15.1281	16.2182	17.5417	18.1671
5	62	16.7149	16.3599	17.5533	18.3695	17.8953
	63	16.2006	16.3906	17.7449	18.3483	17.5840
	64	15.7295	16.4563	17.8651	18.2088	17.1893
	65	15.3328	16.5401	17.9016	17.9481	16.7416
	66	15.0364	16.6251	17.8482	17.5768	16.2762

Table **5:** Prediction Performance of *ClusNet* in the vicinity of $\Pi = [4,6,64]$. The Prediction Accuracy is reported in **nrmse**. Num Cluster refers to the number of clusters that are allocated by **ClusNet.**

m	T	A	Prediction	Num Cluster
4	85	6	0.2000	65
4	64	6	0.1200	60

Table **6:** The value of Ω in the vicinity of $\Pi = [3,8,65]$. This embedding has not been suggested in the literature.

Δ		6	7	8	9	10
m	*T*					
2	63	21.3140	20.5560	19.7292	18.8734	18.0343
	64	20.7168	19.8900	19.0342	18.1951	17.4214
	65	19.8757	19.0200	18.1808	17.4072	16.7482
	66	18.8268	17.9877	17.2141	16.5551	16.0593
	67	17.6226	16.8490	16.1900	15.6942	15.4085
3	63	17.6432	16.6170	16.2077	16.5279	17.4155
	64	16.9146	16.1504	16.1006	16.7542	17.7969
	65	16.1634	15.7348	**16.0662**	16.9975	18.0854
	66	15.4416	15.4030	16.1002	17.2194	18.2439
	67	14.8033	15.1783	16.1860	17.3827	18.2584
4	63	16.1917	17.0997	18.1453	18.4880	18.0424
	64	16.1540	17.3028	18.2528	18.4013	17.7769
	65	16.1671	17.4544	18.2596	18.2063	17.3988
	66	16.2182	17.5417	18.1671	17.8953	16.9167
	67	16.2924	17.5610	17.9752	17.4663	16.3611

Table 7: Prediction Performance of ***ClusNet*** in the vicinity of $\Pi = [3, 8, 65]$. The Prediction Accuracy is reported in **nrmse.** Num Cluster refers to the number of clusters that are allocated by ClusNet.

m	T	A	Prediction	Num Cluster
3	65	8	0.1068	43

References

[1] W. A. Brock and C. L. Sayers. Is the business cycle characterized by deterministic chaos? Journal of Monetary Economics, **22:71–90**, 1988.

[2] E. E. Peters. Chaos and Order in the capital markets. Wiley, 1991.

[3] J. Gleick. Chaos–making a new science. Viking, 1987.

[4] M. C. Mackey and L. Glass. Oscillation and chaos in physiological control **systems.** Science, **197:287,** 1977.

[5] M. Casdagli, D. D. Jardins, S. Eubank, J. D. Farmer, J. Gibson, N. Hunter, and J. Theiler. Nonlinear modeling of chaotic time series : theory and applications. Technical Report **LA-UR-**91-1637, **Los** Alomos National Lab, 1991.

[6] F. Takens. Detecting strange attractors in turbulence. In D. A. Rand and L. S. Young, editors, Dynamical systems and turbulence, page 366. Spinger, 1981.

[7] M. Casdagli. Chaos and deterministic versus stochastic nonlinear modeling. Technical Report 91-07-029, Santa Fe Institute, 1991.

[8] W. C. Mead. Prediction of chaotic time series using cnls-net. Technical Report LA-UR-91-720, **Los** Alomos National Lab, 1991.

[9] W. Liebert, K. Pawelzik, and H. **G**. Schuster. Optimal embeddings of chaotic attractors from topological considerations. Europhysics Letters, pages 521–526, 1991.

[10] W. Hsu, L. S. Hsu, and M. F. Tenorio. A neural network architecture for prediction. Technical Report TR-EE92-38, Purdue University, 1992.

[11] John Moody and C. Darken. Fast learning in networks of locally-tunned processing units. ***Neural*** Computations, **1(2):281–294,** 1990.

[12] M. F. Tenorio and W. T. Lee. Self organizing network for optimum supervised learning. IEEE ***Transactions*** on Neural ***Networks***, **1(1):100–110,** March 1990.