LOCAL DYNAMIC MODELING WITH SELF-ORGANIZING FEATURE MAP

TABLE OF CONTENTS

Abstract of Dissertation Presented to the Graduate School of the University of Florida in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

LOCAL DYNAMIC MODELING
WITH SELF-ORGANIZING FEATURE MAP

By

Ludong Wang

December 1996

Chairman: Dr. Jose C. Principe
Major Department: Electrical and Computer Engineering

Chaotic signals are associated with autonomous response of certain nonlinear dynamical systems. While they are deterministic with few degrees of freedom, chaotic signals are not predictable in the long term. They clearly have considerably more structure than can be inferred from and exploited by traditional stochastic modeling techniques. Consequently it is important to develop new signal processing techniques that are matched to the special characteristics of this class of signals.

In this research, a finite set of linear local models is established as a feasible and effective implementation of dynamic modeling. As a self-organizing feature map (SOFM) is used as the modeling infrastructure, the method is called SOFM-based local linear modeling network.

The previous work on local models only focuses on interpolating local data for short-term prediction. An example is state-dependent functional mapping which does not consider deducing the equations of motion. The aim of the proposed modeling method is to derive a finite set of local linear models to approximate the global dynamics. These models are constructed with spatial constraints while matching the dynamics of a signal in

the temporal sense. This spatial-temporal architecture requires fewer assumptions about the underlying dynamics.

The SOFM constructed with Kohonen's learning law is a localized neural representation of signals. We appended a new linear layer to the SOFM which is trained to approximate the tangent space to the dynamics represented by the local neural field. The use of a neighborhood function imposes a strong statistical constraint over the converged neural field, such that the irregular spacing of local data can be smoothed out. This is significant to the local modeling method as a given signal is usually of finite length. Based on this structure, the constructed local linear models are not totally independent of each other, which helps to reduce the discontinuity between them. On the other hand, the statistical density matching property makes SOFM a good vector quantization procedure. This is not an optimal structure for the overall purpose of dynamic modeling. Therefore a modified Kohonen learning rule is proposed as a new training procedure of the employed SOFM. As the prediction error is involved to reflect the local dynamic fluctuation, this procedure is thus called dynamic learning. With this procedure, the converged neural field becomes a dynamic-oriented representation of the signal.

The proposed SOFM-based modeling scheme has been applied to both synthetic and real-world signals. Experiments demonstrate that this method is capable of faithfully capturing the underlying dynamics of chaotic signals.

The overall structure of the proposed method is a significant extension of the local linear modeling techniques. It also represents a new research direction to explore the potential of the vigorous self-organizing feature map.

CHAPTER 1
INTRODUCTION

1.1 Motivations

Temporal pattern recognition, system control, signal prediction, and chaotic data analysis share a common problem: deducing equations of motion from observations of time-dependent data. Each of them seeks to model the physical world with a certain goal in mind. These models encapsulate the data complexity in a compact, algorithmic specification producing a vast data reduction. Therefore modeling can be defined as a process to extract the underlying dynamic model from a given signal.

In conventional signal processing a wide range of approaches have been developed for modeling signals that are deterministic and predictable. The signals are generally taken as the realization of some stochastic process, and the underlying system is modeled as linear [92]. In such a modeling scheme, the random input produces the uncertainty in the system output.

Chaotic signals are associated with the autonomous response of certain nonlinear dynamical systems. Without any random input, the output of such systems exhibits complex behavior and possesses noise-like spectra. These systems, named chaotic dynamical systems [27], are ubiquitous, e.g., laser beams, biological systems, astronomical motions, weather patterns, fluid flows, and electric circuits, etc. [20] [32] [67] [93]. As the uncertainty in its output originates from the system dynamics instead of an external driving force, they clearly possess considerably more structure than can be inferred from and exploited by traditional stochastic modeling techniques.

To model the chaotic time series, many techniques have been developed. They can generally be classified as global and local models. Global models are constructed once and

fit all the state space, while local models represent local regions. Structure in chaotic attractors tends to be very intricate and nonuniform. Depending on the functional representation, global models have difficulty mimicking such systems adequately. Localized representations can chart out the nuances of chaotic morphology and provide an atlas for the entire state space. Among the local methods, linear models are proved to be computationally cost-effective.

The self-organizing feature map (SOFM) constructed with Kohonen's learning law [52] is normally utilized for feature extraction. It is characterized by the formation of a topographic map of the input patterns, in which the spatial location of the neurons in the output neural field correspond to intrinsic features of the input patterns. As a computational map, it constitutes a basic building block for information processing. The feature map so derived provides a better representation of the local information.

Thus, exploring the application of local linear models in global dynamic modeling, and utilizing SOFM as the modeling infrastructure for this purpose, compose the motivation of this research.

1.2 Research Outline and Objectives

So far the research on the method and application of local linear models is restricted to the forecasting of chaotic signals. In this scenario, a linear predictive model is estimated as the local dynamic map based on the nearest neighbors of the current state, and the prediction is thus obtained by simple function evaluation. Due to the fact that chaotic time series is not predictable in the long-term sense, small short-term prediction error is the goal of this scenario. Under the assumption that the state dynamics is smooth and sufficient data samples are available, this method can generally provide a good predicting performance.

Capturing the dynamics, i.e., the long term behavior of the system, is definitely a significant extension of the local linear forecasting scenario. The goal is to fit a set of local

linear models to the given signal reconstructed in the state space such that autonomous output of the constructed structure exhibits similar dynamic behavior. However, such attempts have been hindered by some concerns [23]. Depending on the partitioning and scales, piecewise linear equations of motion may exhibit periodic behavior when the original dynamics is chaotic and vice versa. Piecewise linear dynamics are considered as violating the physically motivated hypothesis of smooth dynamical systems. Most physical processes do not exhibit arbitrarily fast changes in their first derivatives. Moreover the established set of local linear models generally may not be a discontinuous functional map, which can lead to undesired behavior it is iterated as an autonomous system.

In this research, the above concerns are reconsidered in the following aspects. As long as the local linear models can exhibit chaotic behavior when the original dynamics is periodic as showed in the investigation by Crutchfield, et al. [23], it is reasonable to expect that the structure of local linear models is capable of possessing chaotic dynamics. In the context of chaotic dynamics, the lack of smooth continuation may degrade the short-term prediction performance. However, it is not sufficient to conclude that a faithful approximation of global dynamics is impossible in terms of a finite set of local linear models. Finally the discontinuity problem may be reduced to a sufficient extent, if not eliminated, such that the established local linear models pieced together as a whole may not be hindered to approximate the original dynamics.

To realize the scenario of local linear modeling, it is proposed here to utilize the SOFM as the modeling infrastructure. While SOFM is a localized neural representation of the input, it is constructed in the global sense via the competitive learning process. The local linear models are directly fitted from the constructed SOFM map. Two major issues will be investigated: a) Can the neurons of each local segment in the SOFM neural field collectively bear more reliable information about the local dynamics? b) Can all local dynamics extracted from the individual local segments collectively represent the global dynamics of the given signal?

As with other modeling techniques, we need to make an assumption about the signal. The output of a high-dimensional chaotic system with large Lyapunov exponents will behave very much like random noise, which is extremely difficult to model. Therefore, the proposed modeling approach is mainly for signals whose underlying systems have low dimensionality of less than 4 and small positive Lyapunov exponents of less than 2.5 bits/sec. This is not a very restrictive constraint since lots of chaotic signals of practical interest actually belong to this category. Therefore it has been actually accepted as the practical constraints by most efforts of dynamic modeling [116].

The framework of this research is as shown in Fig. 1.1. The SOFM is explored as an optimal neural representation of signal for the purpose of dynamic modeling, which leads to a modified SOFM learning rule. Based on the obtained neural modeling infrastructure, a procedure of weighted estimation of local models is proposed and utilized. With the overall objective of dynamic modeling, the model validation will be based on the dynamical invariants, as proposed by Principe, et al. [98].

In general, the proposed dynamic modeling scenario differs from the previous work in the sense that the idea of local linear models is employed for the purpose of global dynamics approximation, instead of merely local data interpolating and forecasting.

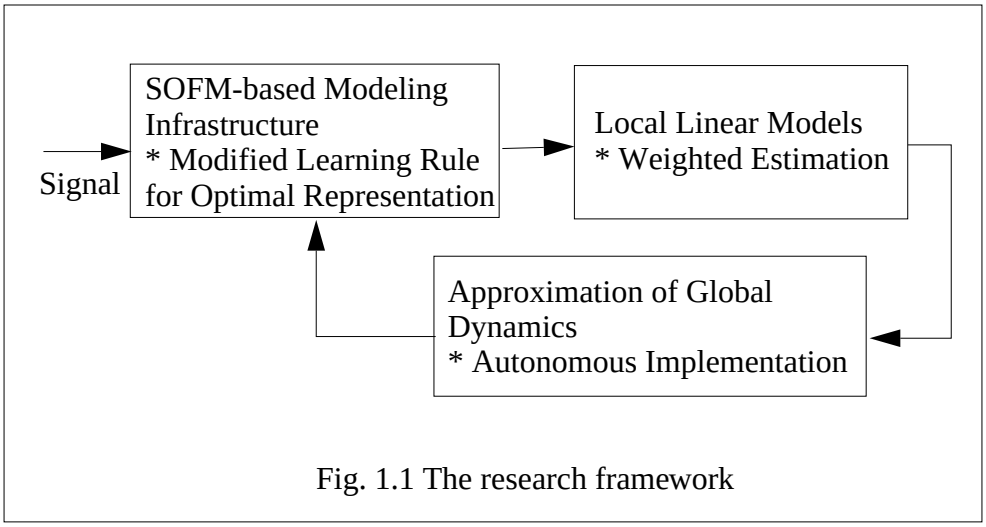


Fig. 1.1 The research framework

1.3 Overview of the Dissertation

This dissertation is composed of seven chapters. Chapter 2 gives a brief description of chaotic dynamics with respect to signals and systems. In the context of signal processing, the major signal modeling techniques are surveyed. In addition, the criterion to quantify the dynamic structure of chaotic systems are also described. In Chapter 3, the research on self-organizing feature map is reviewed, and a demonstrative simulation is conducted for the Lorenz time series. At the end, the application potential of SOFM is also discussed. In Chapter 4, the state-dependent local linear models are first discussed in the context of signal processing. Then the scenario of SOFM-based dynamic modeling network is presented. Based on this modeling architecture, two special techniques are proposed for better modeling performance in Chapter 5. In Chapter 6, the SOFM-based modeling networks together with the proposed techniques are experimented with both equations-generated and real-world signals. Based on the preceding analysis and experimental results, some conclusion are drawn in Chapter 7, where future research is also discussed.

CHAPTER 2
NONLINEAR DYNAMIC MODELING

In the case of a linear time-invariant system, the trajectories will relax on a single hyper-plane through the origin in state space [117], and the state evolution can be described by a linear closed-form solution. The uncertainty in the output is accounted for by the random input. A nonlinear autonomous system, however, can exhibit such a wide variety of dynamical behaviors that the traditional signal processing techniques are not able to capture them effectively. In contrast to the linear time-invariant system, a nonlinear system can not be solved analytically. Instead the solutions are usually obtained by numerically integrating the system equations from different initial conditions. These numerical solutions constitute the state trajectories of the system in the state space, which as a whole represents the underlying dynamics.

When refining a model of a physical process, the attention is usually focused on the agreement of theoretically predicted and experimentally observed behavior. If these agree in some accepted sense, then the model is “correct” within that context. However this procedure is fundamentally limited when the underlying physical process is chaotic [67]. There is an irreducible long-term error in the prediction of a system’s state that is on the order of the chaotic attractor’s size in state space, and one must resort to some dynamical invariants as validation criteria of the reconstructed model.

This chapter is devoted to the discussion of the above observations. In the first section, the dynamical systems and signals are briefly introduced. The characteristic aspects of nonlinear dynamical systems are discussed in comparison with the linear counterpart in Section 2.2. Section 2.3 gives a review of two major classes of nonlinear dynamic models. In the last section, the importance of the dynamical invariants as criteria

of model validation is discussed. In addition, the issues relating to the dynamics reconstruction are also addressed.

2.1 Dynamical Systems and Signals

In the analysis of signals generated by physical systems, it can be assumed from the outset that a dynamical system, expressed as a differential equation or a discrete-time evolution rule, is responsible for the observations, i.e.,

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t)) \tag{1}$$

where $\mathbf{x}(t)$ are the system states, and f is usually referred as the vector field. The vector field f maps a manifold, or a state space $S \subset R^f$ to a tangent space $T \subset R^f$. If f is a linear function of the system states, the underlying system is linear; otherwise, the system is non-linear. Suppose a closed-form solution for equation 1 exists, $\phi_t : S \to S$. For a given initial condition $\mathbf{x}_0$, $\phi_t(\mathbf{x}_0)$ represents a state-space trajectory of the system, or system flow.

A dynamical system can also be expressed in discrete time, which is the realistic situation when observations are sampled every τ_s. The evolution is given by a map from vectors in R^f to other vectors in R^f, identified by a discrete time index n:

$$\mathbf{x}(n) = \mathbf{x}(t_0 + n\tau_s) \tag{2}$$

$$\mathbf{x}(n+1) = F(\mathbf{x}(n)) \tag{3}$$

One assumption we make is that the vector field f is a continuously differentiable function. This is a reasonable condition for most real-world system. With this assumption, the existence of the system flow ϕ_t and its inverse ϕ_t^{-1} for any finite time are ensured [46]. As a consequence, the trajectories of an autonomous system will never cross one another.

For a stable system, the system trajectories will fall into the neighborhoods of some

finite-dimensional limit set after transients die out, and limit set is referred to as the system attractors. In this research, we explore and model the steady-state behavior of a system in the long-term sense.

2.2 Signal Analysis: Linear and Non-linear Models

2.2.1 Linear Models and Limitations

ARMA models have dominated all areas of traditional signal processing for more than half a century. As a linear model, ARMA has two particularly desirable features: it can be understood in great detail and it is straightforward to implement.

For a signal generated by a linear system, the observation $x(t)$ can be linearly related to previous observations and the current and previous driving forces. This leads to a model of the form

$$x(t) = \sum_{i=1}^{p} a_i x(t - iT) + \sum_{i=0}^{q} b_i u(t - iT) \tag{4}$$

where the coefficients $\{a_i\}$ and $\{b_i\}$ are to be determined by fitting to the signal process, and $u(t)$ is a deterministic or stochastic forcing term. This is the well-known Autoregressive-Moving Average (ARMA) model[49]. Taking the Z-transform leads to

$$X(z) = \frac{\sum_{i=0}^{q} b_i z^{-i}}{1 - \sum_{i=1}^{p} a_i z^{-i}} U(z) = \frac{B(z)}{1 - A(z)} U(z) \tag{5}$$

with $U(z)$ the z-transform of the driving force. $A(z)$ and $B(z)$ represent the AR and MA parts of ARMA model respectively.

As shown by Weigend and Gershenfeld [127], the ARMA coefficients contain the same information as the power spectra and autocorrelation function is driven by uncorrelated white noise. The characteristics of the system is independent of the input

while the output uncertainty comes from the external input. Thus, if and only if the power spectrum is a useful characterization of the relevant features of a signal, an ARMA model will be a good choice for the purpose of modeling. However, this appealing simplicity can fail entirely for even simple nonlinearities if they lead to complicated power spectra. Two signal processes can have very similar broadband spectra but can be generated by systems with very different properties, such as a linear system that is driven stochastically by external noise, and a noise-free deterministic non-linear system with a small number of degrees of freedom. The broadband power spectra in the context of an ARMA model comes from external noise input, but in the non-linear counterpart, it arises from the internal non-linearities of the system.

Historically, an important step beyond linear models was taken by Tong and Lim [122] [124]. They suggested the use of globally nonlinear threshold autoregressive model (TAR) which consists of switching between two local linear autoregressive models based on the system's state. Local approximation of the dynamics over small regions in the state space was discussed in the time series literature by Priestley [96] under the name of locally-linear autoregressive moving average (ARMA) models. In another step suggested by Farmer and Sidorowich [29], the dynamics is approximated by simple linear regression or interpolation between sample points, which was used for predicting chaotic time series. The simple idea behind this approach is to recognize that any manifold is locally linear, or locally a hyperplane. A natural extension to ARMA model is to include quadratic and higher order powers of the state, which is called a Volterra series [126].

TAR models, Volterra models, and their extensions significantly expand the scope of possible functional relationships for signal modeling, but these come at the expense of the simplicity with which linear models can be understood and fit to a signal.

2.2.2 Analysis of Linear and Non-linear Systems

The task of analyzing signals observed produced by linear or non-linear systems is basically the same. However the methodologies taken for that purpose are substantially different [2]. This can be reflected in three aspects, namely, 1) analysis space, 2) classification, 3) Modeling.

Analysis space. For a signal produced by a linear process which is invariant with time, sinusoidal functions are effective basis functions for expansion leading to the Fourier space. In the case of signals generated from a process governed by non-linear dynamics, sinusoidal decomposition usually does not simplify the analysis since the process that give rise to chaotic behavior is fundamentally multivariate. The phase space of the underlying system needs to be reconstructed. This reconstruction process will result in vectors in a multi-dimensional space. Methods have been developed to determine the dimension and components of the expected multi-dimensional vectors as explained in the last section of this chapter.

Classification. The characteristic frequencies, or resonant frequencies, of the linear system are invariants of the dynamics. Thus the linear system can be identified and classified by their characteristic frequencies. However non-linear systems generally do not display strong resonant frequencies when they are operating in a chaotic regime. Other invariants have to be defined. Among these invariants are the fractal dimensions and Lyapunov exponents as introduced in section 2.4. They are invariant under any smooth transformation of coordinates of the embedding space.

Modeling. For a linear system, ARMA is sufficient for the purpose of modeling. Non-linear modeling of chaotic processes relies on the idea of a compact geometric attractor on which the observations evolve. Since the orbit is continually folded back on itself by the dissipative forces and the non-linear part of the dynamics, it can be expected to find in the neighborhood of any orbit point $\mathbf{x}(n)$ other orbit points $\mathbf{x}_r(n); r = 1, 2, \ldots, N_B$, which arrive in the neighborhood of $\mathbf{x}(n)$ at different times.

Various forms of interpolation functions can be built to account for whole neighborhoods of state-space and how they evolve from near $\mathbf{x}(n)$ to the whole set of points near $\mathbf{x}(n+1)$. The use of state-space information in the modeling of the temporal evolution of the underlying dynamics is the key innovation in modeling chaotic processes.

2.3 Nonlinear Dynamic Modeling

Many techniques have been developed for the purpose of non-linear dynamical modeling. Following the traditional classification, they can be categorized into local and global models.

Assuming for the moment that the data $x(n)$ have been successfully embedded in the state space in R^F, the task of nonlinear dynamic modeling reduces to selecting a functional form and determining a set of parameters ($\mathbf{a}$) in the map $F(\mathbf{x}, \mathbf{a})$ which evolve as $\mathbf{x}(k) \rightarrow \mathbf{x}(k+1)$. Unlike the linear case, however, there is no algorithmic way to determine the functional form [100]. For a selected functional form of the map, the estimation of parameters is performed with respect to a chosen cost function or metric. Let $\varepsilon_D(k)$ denote the local deterministic error,

$$\varepsilon_D(k) = \mathbf{x}(k+1) - F(\mathbf{x}(k), \mathbf{a})$$

If the map $F(\mathbf{x}(k), \mathbf{a})$ to be constructed is local, then for each neighbor of $\mathbf{x}(k)$, $\mathbf{x}^{(r)}(k), r = 1, 2, \ldots, N_B$,

$$\varepsilon_D^{(r)}(k) = \mathbf{x}^{(r)}(k+1) - F(\mathbf{x}^{(r)}(k), \mathbf{a}) \tag{6}$$

where $\mathbf{x}^{(r)}(k+1)$ the point in state-space to which the neighbor $\mathbf{x}^{(r)}(k)$ evolves. Actually it is not necessarily the rth nearest neighbor of $\mathbf{x}(k+1)$. For a least-squares metric, the local neighborhood-to-neighborhood cost function can be defined as

$$W(\varepsilon, k) = \frac{\sum_{r=1}^{N_B} \left| \varepsilon_D^{(r)}(k) \right|^2}{\sum_{r=1}^{N_B} \left(\mathbf{x}(k) - \langle \mathbf{x}(k) \rangle \right)^2} \tag{7}$$

and minimizing $W(\varepsilon, k)$ results in the time-dependent parameters $\mathbf{a}(k)$. The normalization is arbitrary, but the one shown in (7) uses the size of the attractor to scale the deterministic error. If the fit is to be a global least-squares determination of $\mathbf{a}$, then

$$W(\varepsilon) = \frac{\sum_{k=1}^{N} \left| \varepsilon_D(k) \right|^2}{\sum_{r=1}^{N} \left(\mathbf{x}(k) - \langle \mathbf{x}(k) \rangle \right)^2} \tag{8}$$

which results in a set of time-independent parameters $\mathbf{a}$ in global sense. An interesting option for modeling is to require the parameter $\mathbf{a}$ to be determined by maximizing the average mutual information between $\mathbf{x}^{(r)}(k+1)$ over the neighborhood and the $F(\mathbf{x}^{(r)}(k), \mathbf{a})$, i.e., maximize

$$I(\mathbf{x}^{(r)}(k+1), F(\mathbf{x}^{(r)}(k), \mathbf{a})) = \sum_{r=1}^{N_B} P(\mathbf{x}^{(r)}(k+1), F(\mathbf{x}^{(r)}(k), \mathbf{a})) \times \log\left(\frac{P(\mathbf{x}^{(r)}(k+1), F(\mathbf{x}^{(r)}(k), \mathbf{a}))}{P(\mathbf{x}^{(r)}(k+1)) P(F(\mathbf{x}^{(r)}(k), \mathbf{a}))} \right) \tag{9}$$

with respect to the $\mathbf{a}(k)$ at each k.

All procedures, which construct functional maps and then predict the evolution of the dynamical system in the state space, are based on the scenario described above. In the remaining of this section, major approaches of nonlinear dynamic modeling are reviewed

in two general classes: global models and local models. In addition, the limitations of these methods are also discussed.

2.3.1 Global Models

A number of global models have been developed a present a closed functional representation of the dynamics in the whole state space or at least the whole attractor. Each method uses some expansion of the dynamical vector field $\mathbf{F}(\mathbf{x})$ in a set of basis functions in R^d. One natural global method uses polynomials. The polynomial representation of a global map can be estimated using the measure-based functional reconstruction [33] [14] which uses orthogonal polynomials whose weights are determined by the invariant density on the attractor. Finding the coefficients of the polynomials and the coefficients of the function $\mathbf{F}(\mathbf{x})$ requires the computation of moments of data points in state space. Obviously global modeling is straightforward which leads to equations of motion we are familiar to. However it is unlikely that a standard functional basis set would be capable of representing the intricate geometry of a strange attractor throughout the entire state space.

Different ideas are exploited in neural networks when they are used as nonlinear models for prediction [22]. Neural networks belong to a kind of model utilizing interconnected processing elements with a specific architecture. After a training procedure to establish the interconnections among the units, a nonlinear model is obtained. Actually this is just a nonlinear functional model composed of sums of sigmoid function instead of sums of polynomials. There are many different modifications of neural networks. A feed-forward artificial neural network is usually trained as a one-step predictor of a chaotic time series [61]. The resulting neural network, by iteration, can produce multi-step predictions which are much more accurate than the outputs of a polynomial or a Widrow-Hoff adaptive FIR model. Based on that approach, a modified cost function is designed to reduce those insignificant parameters [128]. Unfortunately the learning process of the neural network appears to be a kind of nonlinear least-squares problem, which is much slower than

the linear fitting process.

In predicting chaotic time series, Casdagli proposed to approximate the partial mapping of f, denoted by πf, as

$$\pi f(\mathbf{x}) \approx \sum_{n=1}^{N} \lambda_n \rho(\|\mathbf{x} - \mathbf{x}_n\|) + \sum_{n=1}^{d} \mu_n p_n(\mathbf{x}) \tag{10}$$

where $\mathbf{x}$ is the state vector, ρ is a radial basis function, $\mathbf{x}_n$ is the center of a radial function, $\{p_n : n = 1, 2, \ldots, d\}$ is a set of polynomial bases of degree at most d, λ_n and μ_n are the parameters which will be determined by using least squares. The number of the radial basis functions, N, is equal to the number of the reconstructed state vectors, each of which is the center of one of the basis functions. Although this is a global approximation, the model can still preserve very well the local properties due to the use of the radial basis functions. However, the disadvantage is that the computation of the parameters will become impractical when the number of data points is large.

The radial basis function approximation can also be implemented in a neural network manner [81]. The output of the network is computed by

$$\phi(\mathbf{x}) = \sum_{j=1}^{k} w_j \rho_j(\mathbf{x}) \tag{11}$$

$$\rho_j(\mathbf{x}) = exp(-\beta_j \|\mathbf{x} - \mathbf{x}_j\|^2) \tag{12}$$

where $\mathbf{x}$ is the state vector, $\mathbf{x}_j$ and β_j are the center and the width of a Gaussian-shape radial function, and w_j are the coefficients of the network. Mead et al. modified this network by including an extra term [79]. The output of the network becomes

$$\phi(\mathbf{x}) = \sum_{j=1}^{k} [w_j + (\mathbf{x} - \mathbf{x}_j)\,\mathbf{d}_j] \frac{\rho_j(\mathbf{x})}{\sum_i \rho_i(\mathbf{x})} \tag{13}$$

where w_j and $\mathbf{d}_j$ are adjustable parameters associated with the jth basis function. The

incorporation of the linear term $(\mathbf{x} - \mathbf{x}_j)\,\mathbf{d}_j$ relaxes the constraint which is imposed on the adaptation in interpolating the point $\mathbf{x}_j$. Consequently, the number of the basis functions can be reduced without degrading the performance. The main drawback is that the training of this network may become unstable occasionally.

2.3.2 Local Models

Unlike the global models which are based on the data from the entire attractor, the local models are established using data from local regions. This method is closely allied to differential topology and is more general than the global approach in the sense that fewer statistical and geometric assumptions about the data are required [23]. Consequently it can be successfully applied to a wider class of behavior.

With the method of local models the state space is partitioned into small regions and a local model is established to describe the dynamics in each region. Thus local models vary from region to region. Based on the consideration of effectiveness and simplicity, local linear models are considered.

The approach of locally linear ARMA models has been discussed in the time series literature by Priestley [96] as a local approximation of the dynamics over small regions. It is a state-dependent sequential type of recursive algorithm for identifying dynamic models using simple linear regression or interpolation. Similar approach was also proposed by Farmer and Sidorowich [29] for predicting chaotic time series. In their work, this approach is also extended to locally polynomial approximations of higher order. They concluded that the first order, or linear, approximation is an effective local approximation, and the approximation with higher-order polynomials are not significantly better than those obtained with first order.

A local linear model considered in the local approach is,

$$\mathbf{x}(n+1) = \mathbf{J}\mathbf{x}(n) + \mathbf{b} \tag{14}$$

where $\mathbf{J}$ is the Jacobian of F at $\mathbf{x}(n)$, and $\mathbf{b}$ is a constant vector. The implicit assumption of this method is that the state dynamics are locally smooth. The parameters $(\mathbf{J}, \mathbf{b})$ can be estimated for the local linear map in the least square sense from the local data samples. as shown in Fig. 2.1.

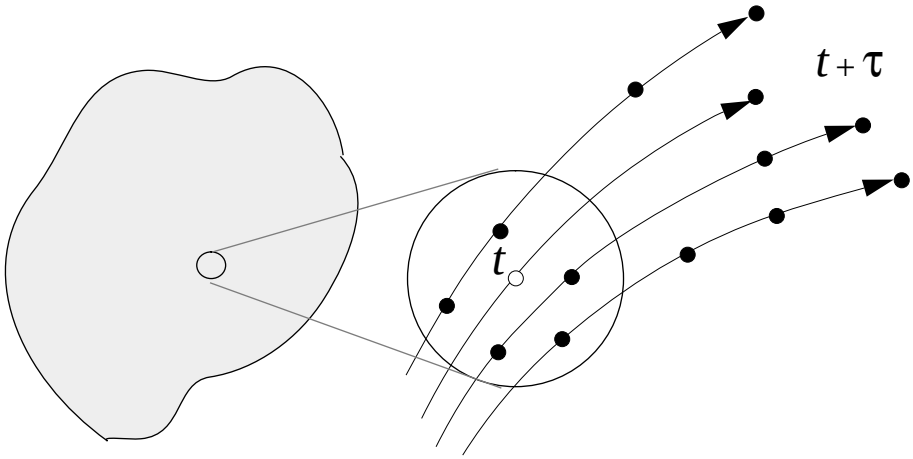


Fig. 2.1 Symbolic illustration of state prediction

Using $N_B > d_L$ nearest neighbors to fit the local linear map, the cost function is

$$W(\varepsilon, k) = \sum_{r=1}^{N_B} \left\| \mathbf{x}^{(r)}(k+1) - \mathbf{J}^{(r)} \mathbf{x}^{(r)}(k) - \mathbf{b}^{(r)} \right\|^2 \tag{15}$$

where d_L is the dimension of the local dynamics. By piecing together the resulted local models, a global approximation to F is established, and we obtain a dynamical system for the observed signal process. The applications of such an approach include the prediction of the future evolution of the system [29][19][1][94][63], removing nose from chaotic data [57][40], obtaining Lyapunov exponents from experimental data [26][113][131][15], and controlling chaotic dynamical systems by application of small controls [90][115][25].

On the other hand, however, the method of local linear models has not been fully explored regarding to the global dynamics modeling. So far it is mainly considered in the

application of interpolating the local data to produce the short-term prediction. No successful work has been done about deducing equations of motion, or simulating them. In the investigation by Crutchfield, et al., it is reported that the piecewise linear models are unreliable indicators of the underlying deterministic behavior [23]. Their trial of one-dimension case showed that piecewise linear equations of motion can exhibit periodic behavior when the original dynamics is chaotic and vice versa. Another concern is that the piecewise linear dynamics violates the physically motivated hypothesis of smooth dynamical systems. Most physical processes do not exhibit arbitrarily fast changes in their first derivatives. In addition, the established function $\tilde{F}$ is in general a discontinuous functional map. Some undesired behavior can be expected when $\tilde{F}$ is iterated as an autonomous system.

These preliminary observations need to be reconsidered in the following aspects. As long as the local linear models can exhibit chaotic behavior when the original dynamics is periodic as showed in the investigation by Crutchfield, et al., it is reasonable to expect that the structure of local linear models is capable of possessing chaotic dynamics. The remaining problem is how to establish its chaotic motions consistent with the desired dynamics. In the context of chaotic dynamics, the lack of smooth continuation may degrade the short-term prediction performance. However, it is unknown if a faithful approximation of nonlinear dynamics is obtainable with a finite set of local linear models. Finally what is not clear is if the discontinuity problem can be reduced sufficiently, if not eliminated, such that it will not hinder the established local linear models pieced together as a whole to approximate the original dynamics. The investigation of these aspects represents a significant exploration of the local linear modeling scenario.

2.4 Dynamical Invariants in Modeling

Packard et al. [91] introduced a procedure to reconstruct a state space trajectory from the output of a dynamical system. Takens proved that such reconstruction preserves

two important quantities: the dimension and the Lyapunov exponents of the system attractor [120]. The dimension of the system attractor is the irreducible number of degrees of freedom of the system, and the Lyapunov exponents are the measurement of the growth or destruction rate of information within the system flow. Since they will not change with any diffeomorphic transformation applied to the system dynamics, they are called dynamical invariants, and employed to quantify nonlinear dynamics.

2.4.1 Dynamics Reconstruction and Correlation Dimension

Since the processes that give rise to chaotic dynamics are fundamentally multivariate, we need to reconstruct the state space of the system from the scalar measurements $x(n)$. This procedure is called dynamics reconstruction. The reconstruction not only provides a theoretical justification for the dynamic modeling but also enables us to measure the underlying dynamics that produced the signal.

As suggested by Packard et al., an observable of a dynamical system, $x(n)$, and either its delayed samples $x(t+\tau), x(t+2\tau), \ldots$ or its derivatives $x'(t), x''(t), \ldots$ can be used as the coordinates of the points on a trajectory in an Euclidean space [91]. In this research, the delay coordinate method is used for the reconstruction

$$\mathbf{x}(t) = \left[x(t)\ \ x(t+\tau)\ \ \ldots\ x(t+(d-1)\tau)\right] \tag{16}$$

where τ is a constant delay, d is the embedding dimension of the state space. According to Takens' embedding theory, the reconstruction dimension must satisfy $d \geq 2d_A$, where d_A is the dimension of the attractor. This condition guarantees that there is no crossing among trajectories in the reconstruction space.

The embedding dimension d can be determined in the estimation procedure of correlation dimension. As proposed by Grassberger and Procaccia, d is determined using the properties of the correlation integral

$$D(\varepsilon, d_\varepsilon) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} H(\varepsilon - \|\mathbf{x}_i - \mathbf{x}_j\|) \tag{17}$$

where d_ε is the dimension of the reconstruction space, and $H(x)$ is the Heaviside function $H(x>0) = 1$ and $H(x<0) = 0$. The value of the correlation integral is proportional to the average number of pairs of the points within a spheroid with radius ε. Thus the correlation integral from a reconstructed trajectory can be computed by counting the number of points located inside the ε-neighborhood of each state vector, and averaging the count. The correlation dimension is the slope of the curve of $\ln C(\varepsilon)$ vs. $\ln(\varepsilon)$, which is called Correlation Integral Map (CIM). When the trajectory is constructed in a space of dimension higher than a certain value, the slope of the CIM curves will saturate and become a constant in a scaling region of ε. This slope is the estimate of the correlation dimension.

On the other hand, when the saturation occurs, the reconstruction is topologically equivalent to the original system dynamics since the attractors are of the same dimension. The low bound of the embedding dimension d can be identified by the smallest reconstruction dimension d_ε. The use of this procedure will be illustrated in Chapter 6.

2.4.2 The Lyapunov Exponents

Lyapunov exponents are a generalization of the eigenvalues in linear systems. They give a means of characterizing the stretching and contracting characteristics of attractors. For an nth order dynamical system, Lyapunov exponents can be defined by the averaged long-term evolution of an infinitesimal n-sphere with initial radius $\varepsilon(0)$. Due to the locally deforming nature of the flow, the sphere will become an n-ellipse. In terms of the length of the ellipsoidal principal axis $\varepsilon_i(t)$, the ith one-dimensional Lyapunov exponent is then defined as

$$\lambda_i = \lim_{t \to \infty} \frac{1}{t} \log_2 \frac{\varepsilon_i(t)}{\varepsilon(0)} \tag{18}$$

Here, a based two logarithm is used to define the Lyapunov exponents in bits/sec. If the natural logarithm is used, the unit becomes nats/sec. The sign of a Lyapunov exponent represents the divergence (plus) or the convergence (minus) of the system flow in the considered direction. When a system contains positive Lyapunov exponents, say $\lambda_1 > 0$, the output of the system is a chaotic type with a broad-band spectrum. In this case typical, infinitesimally displaced, initial conditions separate from each other exponentially in time, with the infinitesimal distance between them on average growing as $exp(\lambda_1)$. Hence, the condition $\lambda_1 > 0$ is usually referred to as implying exponential sensitivity to initial conditions for the attractor. The largest Lyapunov exponent can be estimated using the robust algorithm proposed by Wolf, et al [131].

A chaotic system has at least one positive Lyapunov exponent. This endows the system dynamics with so high sensitivity to the initial condition that the dynamical behavior of a chaotic system is unpredictable in any long term sense.

It has been conjectured by Kaplan and Yorke [48] that there is a relationship giving the fractal dimension of a typical chaotic attractor in terms of Lyapunov exponents [28]. Let K be the largest integer satisfying

$$\sum_{j=1}^{K} \lambda_j \geq 0 \tag{19}$$

with the index in descending order, i.e., $\lambda_j \geq \lambda_{j+1}$. Define the quantity D_L called the Lyapunov dimension [89],

$$D_L = K + \frac{1}{|\lambda_{K+1}|} \sum_{j=1}^{K} \lambda_j \tag{20}$$

The conjecture is that the Lyapunov dimension is the same as the information dimension of the attractor [38]. Although no rigorous proof of the Kaplan-Yorke conjecture exists, some rigorous results related to it have been obtained by Young [132] and by Ladrappier [59].

2.4.3 Model Validation

For non-chaotic signals, the accuracy of their dynamic models can be assessed in the time or in the frequency domains. In the time domain, the output of the model can be compared with the original signal in a sample-by-sample way. In the frequency domain, the locations and the amplitudes of the fundamental frequencies for the original signal and the model output should be identical.

As discussed in section 2.2.2, this is not feasible for chaotic dynamic model. A difficulty arises due to the positive Lyapunov exponents possessed by the underlying chaotic systems. Given an initial condition, the iterates of the model will eventually diverge from the original signal. Therefore the quality of the constructed model can not be assessed by the sample-by-sample criterion.

On the other hand, the divergence of two nearby trajectories is mainly determined by the largest Lyapunov exponent. Based on this token, we may validate a constructed model by comparing the divergences of the output of the model and the original signal.

The divergence of the original signal is obtained using Casdagli's conjecture curve, which is calculated by

$$\varepsilon_k = \varepsilon_0 e^{\lambda_{max}(k\Delta t)} \tag{21}$$

where ε_0 is the initial separation, λ_{max} is the largest Lyapunov exponent, and Δt is the sampling period of the signal. In the time domain, the divergence of the constructed model can be reflected by the curve of the average multi-step prediction error which is calculated

computed by

$$\zeta_{\tilde{f}}^{2}(k) = \frac{1}{\sigma^{2}(N-k)} \sum_{i=1}^{N-k} (x(i+k) - \tilde{x}(i+k))^{2} \tag{22}$$

where k is the prediction step, σ^2 is the variance of the original signal, N is the total number of data signal samples. and $\tilde{x}(i+k)$ is a k-step iterative prediction of the model $\tilde{F}$. For a good model, its divergence $\zeta_{\tilde{f}}(k)$ is expected to be close to the divergence ε_k of the original signal.

Though the average multi-step prediction error criterion has its practical importance, it still belongs to the category of the sample-by-sample comparison, and thus only local approximation of the model can be reflected. In this research, the measurements of the dynamical invariants will also be considered for the model validation, as proposed by Principe, et al. [98]. For a successful model which has really captured the underlying dynamics of the chaotic time series, a trajectory reconstructed from the iterates of the model is expected to be consistent with that of the original system in terms of dynamical invariants as defined in the two previous sections.

CHAPTER 3
SELF-ORGANIZING FEATURE MAP

3.1 Introduction

Based on different training mechanism, the neural network architectures for signal processing can roughly be divided into three classes [55]. Feed-forward networks [111] transform sets of input signals into sets of output signals. The desired input-output transformation is usually determined by external, supervised adjustment of the system parameters. In Feedback networks [47], the input information defines the initial activity state of a feedback system, and after state transitions the asymptotic final state is identified as the outcome of the computation. In the last category, neighboring neurons in the network compete with each other by means of mutual lateral interactions, and develop adaptively into specific detectors of different signal patterns. Self-organizing feature map (SOFM) belongs to this third category.

SOFM was first developed as a topographically ordered computational map to represent a continuous input by a place-coded populational response, whose peak location reflects the best match [50]. The cortical neurons in the map, each with a slightly different range of stimulus selectivity established during training, operate as parallel filters on the input almost simultaneously. The stimulus parameter, coded as the location of the most active neuron, can be accessed by higher processing layers via relatively simple neural connection. The spatial ordering of neural responses was first simulated by Malsburg [72]. His model explained the formation of the orientation columns in the visual cortex. The self-organization of the nervous system and the formation of cortical maps have been studied quite extensively by Kohonen [52], Swindale [119], Linsker [64], and Miller et al [80]. Kohonen succeeded in defining a process which takes a computational shortcut to achieve

the effect accomplished by the lateral interactions. This process leads to a topological approximation of the input space.

SOFM networks are fundamentally based on competitive learning. In the course of competitive learning, the output neurons of the network compete among themselves to be activated or fired, with the result that only one output neuron, the *winner-takes-all neuron*, is on at any one time. To induce a winner-takes-all competition, the idea of using negative feedback paths between the output neurons was originally proposed by Rosenblatt [110].

In a SOFM, the neurons are placed at the nodes of a lattice that is usually two-dimensional. The output neurons become specifically tuned to various input signal patterns through an unsupervised learning process. The characteristic aspect of SOFM is that the locations of the neurons so tuned tend to become ordered with respect to each other in such a way that a meaningful coordinate system for different input features is created over the lattice [55].

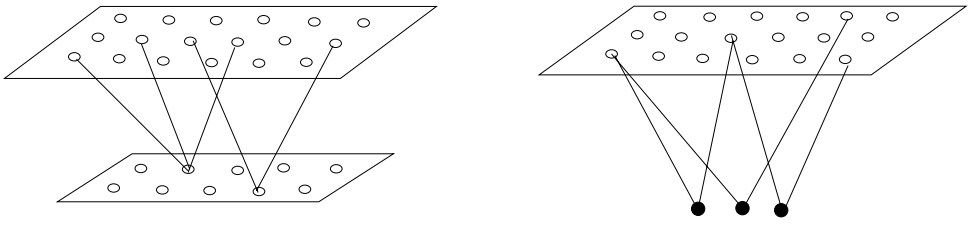
Different ways have been proposed to derive the Kohonen model. It can be derived using basic ideas of self-organization, motivated by neurobiological considerations as in [52]. Alternatively, a vector quantization approach involving an encoder and a decoder has been suggested, which is motivated by communication-theoretic considerations [69][70]. This method bring us more insight into the time dependent parameters of the Kohonen model.

3.2 Formation of SOFM

3.2.1 Structures

The manner in which the input patterns are specified determines the nature of the feature-mapping model. There are actually two commonly considered models as illustrated in Fig. 3.1. In both architectures, a two-dimensional lattice of output neurons are fully connected to the inputs, though only a few connections are shown. Both models were inspired by the pioneering self-organizing studies of von der Malsburg, who noted that a

model of the visual cortex could not be entirely genetically predetermined. Instead, a self-organizing process involving synaptic learning may be responsible for the local ordering of feature-sensitive cortical cells. However, a global topographic ordering can not be achieved, if only fixed neighborhoods were considered.



(a) (b)

Fig. 3.1 Two types of feature mapping architecture.

The model of Fig. 3.1 (a) was originally proposed by Willshaw and vonder Malsburg [130] based on the biological analysis to explain the process of retinotopic mapping from the retina to the visual cortex in higher vertebrates. In this model two separate two-dimensional lattices of neurons are connected together, with one projecting onto the other. One lattice represents presynaptic or input neurons while the other lattice represents postsynaptic or output neurons. The basic idea of this model is for the geometric proximity of presynaptic neurons to be coded in the form of correlations in their electrical activity, and to use these correlations in the postsynaptic lattice so as to connect neighboring presynaptic neurons to neighboring postsynaptic neurons. A topologically ordered mapping is thus generated by self-organization.

With the purpose of capturing the essential feature of computational maps in the brain and yet remain computational tractable, the model of Fig. 3.1 (b) was introduced by Kohonen [52]. Unlike the above Willshaw-Malsburg model where the input dimension is the same as the output dimension, the two lattices have different dimensionality. The

Kohonen model has received much more attention in the literature than the Willshaw-Malsburg model, since the Latter is more tractable, albeit more specific. This research only focus on the Kohonen model.

3.2.2 The Algorithm of Kohonen Model

There are several ways of unsupervised learning which can be considered to self-organize a network into a feature map [45]. Two of them are as follow:

1. Lateral interconnections within the output lattice are employed in the ordinary competitive learning process. Such lateral interconnections usually take the Mexican hat form, which is excitary between nearby output neurons and inhibitory at longer range with strength falling off with distance as shown in Fig. 3.2.

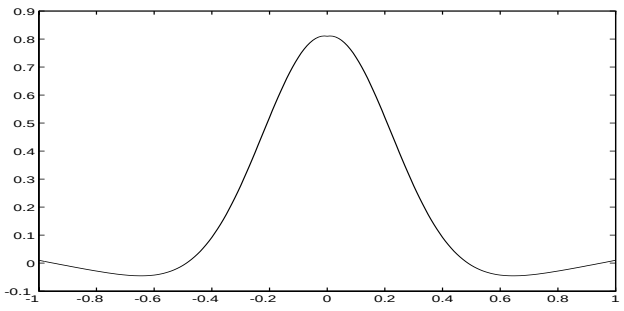


Fig. 3.2 Mexican hat form of lateral connection

2. Based on the ordinary competitive learning, the learning procedure is extended to the neighbors of the winning neurons with fall-off scale. This is the Kohonen's algorithm considered in this study.

Kohonen's algorithm takes a computational shortcut to achieve the effect accomplished by the Mexican hat lateral interactions. There are no lateral connections, but the neurons are updated in blocks, concentrated around the inners. In this way, due to frequent overlap of such blocks in learning and similar corrections imposed within each black, the values of the neurons updated tend to become smoothed. Simultaneously they also tend to

become ordered.

Let $\mathbf{x}$ and $\mathbf{w}_j$ denote the input signals and the synaptic weight vector of neuron j respectively

$$\mathbf{x} = [x_1, x_2, \ldots, x_p]^T \in \mathbf{X} \tag{23}$$

$$\mathbf{W} = \{\mathbf{w}_j | \mathbf{w}_j = [w_{j1}, w_{j2}, \ldots, w_{j2}]^T, j = 1, 2, \ldots, N\} \tag{24}$$

where N is the total number of neurons on the output lattice.

The layout of the output lattice and the structure of each neuron in Fig. 3.3. With

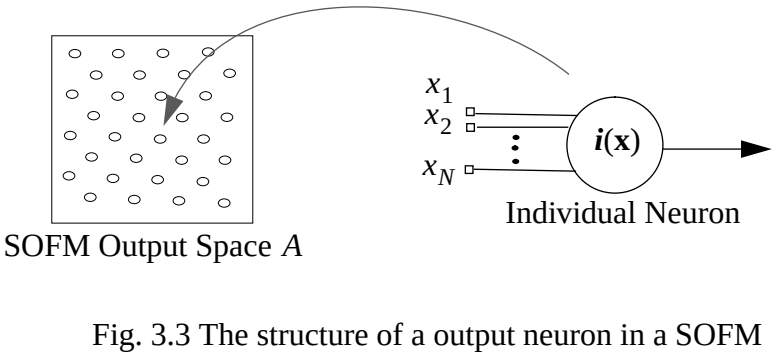


Fig. 3.3 The structure of a output neuron in a SOFM

the weight vectors randomly initialized, the adaptive process of Kohonen's algorithm is composed of repeated cycles: Sampling, Similarity Matching and updating.

Let $i^*_n(\mathbf{x})$ denote the index of the neuron that best matches the input vector $\mathbf{x}$ at time n. After the sample $\mathbf{x}$ is drawn from the input space with a certain probability. $i_n^*(\mathbf{x})$ is determined by applying the condition

$$i_n^*(\mathbf{x}) = arg(\min_j \|\mathbf{x} - \mathbf{w}_j\|), j = 1, 2, \ldots, N \tag{25}$$

where $\|\circ\|$ denotes the Euclidean norm of the argument vector. Then the nth adaptation of

the weight space follows in the manner,

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \eta(n)\,\Lambda(\mathbf{r}_{\boldsymbol{i}_n(\mathbf{x})}, \mathbf{r}_j)\,(\mathbf{x} - \mathbf{w}_j(n)) \tag{26}$$

where $\eta(n)$ is the learning rate as applied in other neural networks. The novel aspect of the adaptive process with SOFM is the employment of a neighborhood function $\Lambda(\mathbf{r}_{\boldsymbol{i}_n(\mathbf{x})}, \mathbf{r}_j)$, which makes the SOFM essentially different from other competitive networks. The neighborhood function $\Lambda(\mathbf{r}_{\boldsymbol{i}_n(\mathbf{x})}, \mathbf{r}_j)$ is both a function of time index n and

$$\left\| \mathbf{r}_{\boldsymbol{i}_n(\mathbf{x})} - \mathbf{r}_j \right\| \tag{27}$$

as well, where $\mathbf{r}_{i_n(\mathbf{x})}$ and $\mathbf{r}_j$ designate respectively the coordinate position of the winner-takes-all neuron $i_n(\mathbf{x})$ and the neuron j on the output lattice space. Numerous simulations have shown that the best results in self-organization are obtained if the neighborhood function $\Lambda(\mathbf{r}_{\boldsymbol{i}_n(\mathbf{x})}, \mathbf{r}_j)$ is selected fairly wide at the beginning and then gradually shrinks with time n [55]. This behavior is equivalent to initially using a strong positive lateral feedback, and then enhancing the negative lateral feedback. The use of a neighborhood function around the winning neuron $i_n(\mathbf{x})$ provides a computational shortcut for emulating the formation of a localized response by lateral feedback. In the next section, it will be shown that the shrinking trend of the neighborhood function corresponds to the decreasing variance of a deviation noise.

A typical choice for $\Lambda(\mathbf{r}_{i_n(\mathbf{x})}, \mathbf{r}_j)$ is

$$\Lambda(\mathbf{r}_{i(\mathbf{x})}, \mathbf{r}_j) = exp\left(-\frac{\left\| \mathbf{r}_{i(\mathbf{x})} - \mathbf{r}_j \right\|^2}{2\sigma(n)^2}\right) \tag{28}$$

where $\sigma(n)$ is a time dependence parameter which controls the effective adaptation coverage of $\Lambda(\mathbf{r}_{i(\mathbf{x})}, \mathbf{r}_j)$. A neighborhood function with two different value of $\sigma(n)$ are shown in Fig. 3.4.

In general, the essence of Kohonen's SOFM algorithm is that the more detailed properties of the Hebb-like rule and lateral interactions are achieved with a simple geometric computation. The weight update equation is fundamental to the formation of a self-organizing feature map, which represents a combination of three distinct ideas:

* The selection of the winner-takes-all neuron $\boldsymbol{i}_n(\mathbf{x})$, which is achieved by applying the similarity-matching procedure of (25).

* The formulation of a neighborhood function $\Lambda(i^*, i)$ around the winner-takes-all neuron $\boldsymbol{i}_n(\mathbf{x})$, which is originally introduced by Kohonen [52].

* A competitive learning rule first introduced into the neural network literature by Grossberg [37][38].

It is the combination of these three ideas that endows the SOFM algorithm with its selective tuning capability.

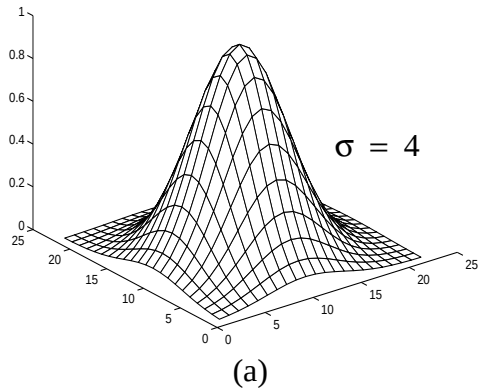


Fig. 3.4 Neighborhood functions with two different size of coverage

3.2.3 Convergence

In the analysis of Kohonen's SOFM by Ritter et al [106], a cost function based on the extension of the simple competitive learning process is defined as

$$C(\mathbf{w}) = \frac{1}{2}\sum_{\mu}\sum_{i}\Lambda(i, i^*)\,\|\mathbf{x}_\mu - \mathbf{w}_i\|^2 \tag{29}$$

Gradient descent on this cost function yields

$$\langle\Delta\mathbf{w}_j\rangle = \eta\sum_{\mu}\Lambda(i, i^*)\,(\mathbf{x}_\mu - \mathbf{w}_j) \tag{30}$$

which is just the sum of the Kohonen adaptation rule over all the patterns μ. Thus on average with small enough η the Kohonen rule decreases the cost function (29) until a minimum is reached, although it might be a local minimum.

In general, there is no well-defined criterion that can be used to assure the stability of the map formed by the algorithm. The convergence process in a general context can be analyzed by viewing the learning process involved in the formation of the map as a stochastic process. At time n the state of the map is described by the set of synaptic weight vectors

$$\mathbf{W}(n) = \{\mathbf{w}_i(n), 1 \le i \le N\} \tag{31}$$

where N is the total number of neurons in the model. Indeed, the SOFM algorithm used to update the state $\mathbf{W}(n)$ is a Markov process, in terms of a set of possible states and a set of transition probabilities between states [42]. Ritter and Schulten [105] have derived a Fokker-Planck equation for the SOFM algorithm in the vicinity of equilibrium condition and for small values of the learning rate, which provides a means to describe the time evolution of the underlying weight-space probability distribution of the model. They have used this approach to investigate the conditions for convergence and the stability properties of the algorithm. Good agreement between theory and statistical measurements based on computer simulations is reported.

Basically the success of map formation is critically dependent on the scheduling parametric quantities, namely, the learning-rate parameter η and the neighborhood function Λ_i, as involved in (26). The adaptive schedule of these two quantities should be in accordance with the training depth. The experimental observations by Kohonen [53] provides a useful guide.

a) The exact form of variation of $\eta(n)$ is not critical. During the initial phase of training, it should begin with a value close to unity, and thereafter gradually decreases. During this phase, the weight vector $\mathbf{w}_j(n)$ experience larger adaptation such that they can to oriented toward the right topological order. The remaining iterations of the algorithm is characterized by small learning value of learning rate which leads the training process into the convergence phase. This phase normally takes longer time for the fine tuning of the computational map.

b) The neighborhood function Λ_i plays the role of orientation for the topological ordering of weight vectors $\mathbf{w}_j(n)$. During the initial phase of training, it should be sufficiently wide to cover more neurons. As the topological order takes place in the neural field, Λ_i is permitted to shrink with time n.

Based on the experimental simulation, and in accordance with the above reported observations, I found that the following schedule form is a good choice

$$\eta(n) = \frac{a}{1+bt} \tag{32}$$

$$\sigma(t) = \frac{c}{1+dt} \tag{33}$$

where a, b, c and d are constants, which can be easily found by experimental simulations in accordance with the each application.

3.3 Localized Neural Representation of Signals

3.3.1 Properties of SOFM

The real world signals can be represented in two different neural network models [7]. One is the global model in which each signal is represented by a specific stable excitation pattern in the network. This global and distributed representation is used in associative memory models [51] [82] [6] [8] [47] and in PDP models [111]. Another model is the cortical cognitive map of localized representation. Kohonen's SOFM model belongs to the localized representation model. Such local representation may be used as a pattern analyzer, whose outputs are then supplied to neural encoders where various features are combined to form a distributed representation.

With the neighborhood relations involved in the learning process, the converged feature map of Kohonen model possesses important statistical and metric characteristics of the input. Let $X \subset \Re^N$ denote a continuous input space with its topology represented by the metric relationship of vectors $\mathbf{x} \in X$. A feature map is represented by a non-linear transformation relation

$$\Phi: X \to A \tag{34}$$

where A represent the set of computational neurons located at the grids of a lattice as shown in Fig. 3.1. The neural representation of the input can be taken as a map of duality. For a given input $\mathbf{x} \in X$, the constructed SOFM identified the winner-takes-all neuron $i(\mathbf{x})$ in the output space A as the best-matching in accordance with the feature map Φ. On the other hand, the synaptic weight vector $\mathbf{w}_j$ of neuron $i(\mathbf{x})$ also serves as a pointer for that neuron into the input space X. This relationship is as depicted in Fig. 3.5. The relationship among the SOFM neurons can be further viewed in terms of an excitation pattern in the neural field. Let $E(\mathbf{x})$ is defined as the response region of input $\mathbf{x}$

$$E(\mathbf{x}) = \{\mathbf{w}_j \mid \|\mathbf{x} - \mathbf{w}_j\| \le r_E, j = 1, 2, \ldots, N\} \tag{35}$$

which consists of the neurons excited in response to $\mathbf{x}$. That is, the input $\mathbf{x}$ is collectively represented by the region $E(\mathbf{x})$ of the neural field A. On the other hand, we can also define $R(\mathbf{w}_i)$ as the receptive field of neuron $\mathbf{w}_i$

$$R(\mathbf{w}_i) = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{w}_j\| \le r_R, j = 1, 2, \ldots, N\} \tag{36}$$

which in turn is the set of input signal that excites the neurons at $\mathbf{w}_i$.

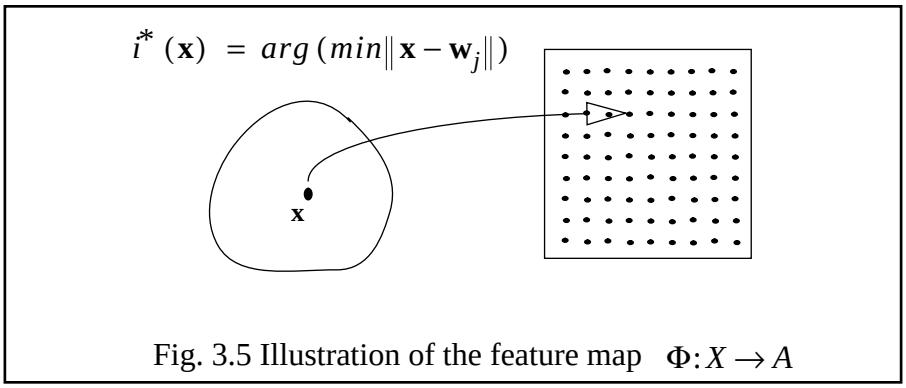


Fig. 3.5 Illustration of the feature map $\Phi: X \to A$

The self-organizing feature map possesses some important properties which can be summarized as: approximation of input space, density matching and topological preserving.

Approximation of the input space. The SOFM obtained with Kohonen's learning law in the form of synaptic weight vectors provides a good approximation to the input space. The theoretical basis of this observation is rooted in the vector quantization theory with the motivation of data reduction or compression [36]. Actually the LBG (Linde-Buzo-Gray) algorithm is closely related to the SOFM algorithm. The relationship was delineated through the derivation of Kohonen's learning algorithm by Luttrell [69]. In Luttrell's scheme as shown in Fig. 3.6, a signal-independent noise process $\boldsymbol{\nu}$ is introduced following the encoder $\mathbf{c}(\mathbf{x})$ to account for the possible distortion of the output code $\mathbf{c}(\mathbf{x})$

between the encoder and the decoder. For this noisy encoder-decoder model, the following average distortion function is considered

$$D = \frac{1}{2} \int_{-\infty}^{\infty} d\mathbf{x} f(\mathbf{x}) \int_{-\infty}^{\infty} d\boldsymbol{v} \pi(\boldsymbol{v}) \| \mathbf{x} - \mathbf{x}'(\mathbf{c}(\mathbf{x}) + \boldsymbol{v}) \|^2 \tag{37}$$

where $\pi(\boldsymbol{v})$ is the probability density function of the noise $\boldsymbol{v}$ over which the second integration is performed. Actually the noiseless encoder-decoder model, which the LBG algorithm [62] is based on, is a simplified form of the above noisy model.

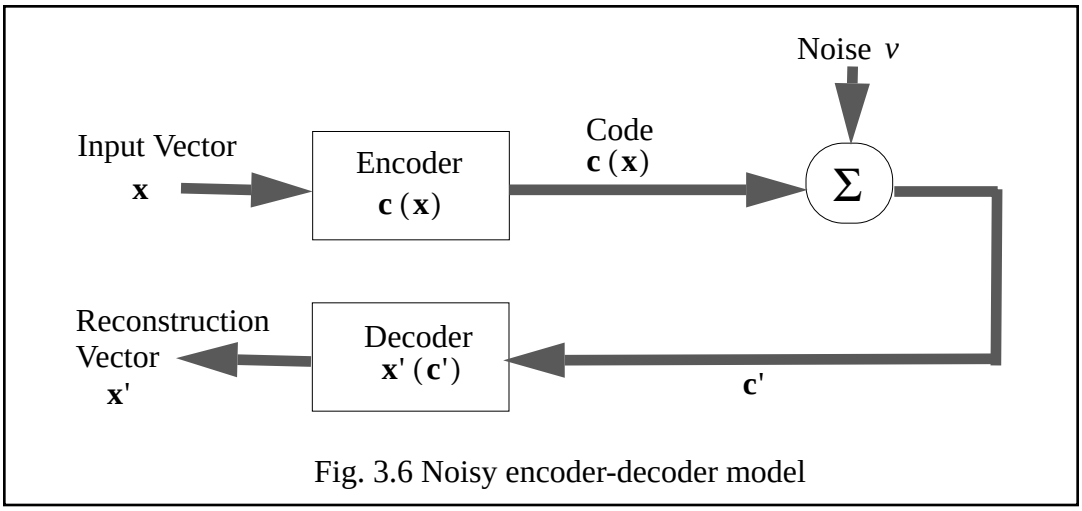


Fig. 3.6 Noisy encoder-decoder model

The optimum encoding-decoding scheme is determined by varying the functions $\mathbf{c}(\mathbf{x})$ and $\mathbf{x}'(\mathbf{c}(\mathbf{x}) + \boldsymbol{v})$ to minimize an average distortion function D. Differentiating D with respect to $\mathbf{x}'(\mathbf{c})$, the derivative takes the form

$$\frac{\partial D}{\partial \mathbf{x}'(\mathbf{c})} = -\int_{-\infty}^{\infty} d\mathbf{x} f(\mathbf{x}) \pi(\mathbf{c}' - \mathbf{c}(\mathbf{x})) \, (\mathbf{x} - \mathbf{x}'(\mathbf{c})) \tag{38}$$

Two necessary conditions are embodied in the minimization of the average distortion D.

Condition 1. Given the input vector $\mathbf{x}$, the optimal code is $\mathbf{c} = \mathbf{c}(\mathbf{x})$ to obtain the minimum distortion measure

$$\int_{-\infty}^{\infty} d\boldsymbol{v}\pi(\boldsymbol{v}) \| \mathbf{x} - \mathbf{x}'(\mathbf{c}(\mathbf{x}) + \boldsymbol{v}) \|^2 \tag{39}$$

Condition 2. Given the code $\mathbf{c}'$, the reconstruction vector $\mathbf{x}'(\mathbf{c})$ should satisfy

$$\mathbf{x}'(\mathbf{c}) = \frac{\int_{-\infty}^{\infty} d\mathbf{x} f(\mathbf{x}) \pi(\mathbf{c}' - \mathbf{c}(\mathbf{x})) \mathbf{x}}{\int_{-\infty}^{\infty} d\mathbf{x} f(\mathbf{x}) \pi(\mathbf{c}' - \mathbf{c}(\mathbf{x}))} \tag{40}$$

which is obtained simply be setting the partial derivative $\partial D / \partial \mathbf{x}'(\mathbf{c})$ equal to zero and then solving for $\mathbf{x}'(\mathbf{c})$ in closed form.

When the stochastic gradient descent learning is used to realize the condition 2, and the input vectors $\mathbf{x}$ is selected at random from the input space X in accordance with $\int d\mathbf{x} f(\mathbf{x})$, an adaptation procedure for the reconstruction vector $\mathbf{x}'(\mathbf{c}')$ takes the form

$$\mathbf{x}'(\mathbf{c}') \leftarrow \mathbf{x}'(\mathbf{c}') + \eta\pi(\mathbf{c}' - \mathbf{c}(\mathbf{x}))(\mathbf{x} - \mathbf{x}'(\mathbf{c}')) \tag{41}$$

where η is the learning-rate parameter and $\mathbf{c}(\mathbf{x})$ is the nearest-neighbor encoding approximation to condition1. This procedure is applied to all $\mathbf{c}'$, for which we have

$$\pi(\mathbf{c}' - \mathbf{c}(\mathbf{x})) > 0 \tag{42}$$

The significant aspect of Luttrell's derivation is that the identical relationship between the noisy encoder-decoder model and Kohonen's SOFM algorithm becomes manifest as shown in Table 3.1.

The LBG algorithm can be derived from a special case of the noisy encoder-decoder model without incorporating the explicit noise $\boldsymbol{v}$ process, or by setting the proba-

bility density function $\pi(\boldsymbol{v})$ equal to a Dirac delta function $\delta(\boldsymbol{v})$. Accordingly, the LBG algorithm for vector quantization is the batch training version of the SOFM algorithm with zero neighborhood size: $\pi(0) = 1$.

Luttrell's derivation shows that the SOFM learning law results in an approximation of the input space similar to vector quantization (VQ). Therefore SOFM has been widely used as a VQ.

Table 3.1 SOFM algorithm vs. Luttrell's derivation

Encoding-Decoding Model	SOFM Algorithm
Encoder **c(x)**	Best-matching neuron $i^*(\mathbf{x})$
Reconstruction vector x'(c')	Synaptic weight vector $\mathbf{w}_j$
Function $\pi(\mathbf{c}'\text{-}\mathbf{c}(\mathbf{x}))$	Neighborhood Function $\Lambda_{i(\mathbf{x})}$

Density Matching. The feature map Φ reflects variations in the statistics of the input distribution: regions in the input space $\boldsymbol{X}$ from which sample vectors $\mathbf{x}$ are drawn with a high probability of occurrence are mapped onto larger domains of the output space A, and therefore with better resolution than regions in $\boldsymbol{X}$ from which sample vectors $\mathbf{x}$ are drawn with a low probability of occurrence.

Let $Pr(\mathbf{x})$ designate the multi-dimensional probability density function (pdf) of the input vector $\mathbf{x}$. The pdf, integrated over the entire input space $\boldsymbol{X}$ complies with the constraint

$$\int_{\boldsymbol{X}} Pr(\mathbf{x})\, d\mathbf{x} = 1 \tag{43}$$

Let $m(\mathbf{x})$ denote the map magnification factor as the number of neurons in a small volume $d\mathbf{x}$ of the input space $\boldsymbol{X}$. Integrated over $\boldsymbol{X}$, this magnification factor $m(\mathbf{x})$

must contain the total number N of neurons in the networks

$$\int_{X} m(\mathbf{x})\, d\mathbf{x} = N \tag{44}$$

For the SOFM algorithm to match the input density exactly, it is required that

$$m(\mathbf{x}) \propto Pr(\mathbf{x}) \tag{45}$$

This implies that the pdf is reflected in the resolution and neural response regions. Let $Pr(\mathbf{x})$ denote the multidimensional probability density function of the input vector $\mathbf{x}$. Then the Kohonen model adapts itself in a volume number density sense to conform approximately to $Pr(\mathbf{x})$. In terms of response regions, this can be expressed as

$$N(E(X_p)) > N(E(X_q)), \quad \text{if } Pr(X_p) > Pr(X_q) \tag{46}$$

where X_p and X_q represent two equi-volume subregions in the input space X, and $N(\circ)$ denote the number of elements contained by its argument. On the other hand, however, Kohonen learning law does not, in general, produce a set of equiprobable weight vectors. Depending on the encoding metric implemented, different results have been reported in the literature. For the nearest-neighbor encoding as in the standard form of the SOFM algorithm, the encoding method results in [107]

$$m(\mathbf{x}) \propto Pr(\mathbf{x})^{2/3} \tag{47}$$

As a general rule confirmed by computer simulations, the feature map by SOFM model tends to overrepresent regions of low input density and underrepresent regions of high input density [42]. To improve the density-matching property of the SOFM model, some heuristics can be added to the algorithm. One such method is the addition of conscience to the competitive learning procedure, which controls with bias the competition process such

that each neuron, regardless of its location in the output lattice, has the chance to win the competition with a probability close to the ideal rate $1/N$, where N is the total number of neurons.

In the proposed application, the deviation of SOFM from the equiprobable representation is not a serious problem. To preserve the information of the input in our signal processing task, it is more important and advantageous to construct a SOFM map which represents the complexity of the trajectory as will be explained later. The nice aspect of SOFM is the flexibility of the learning law. Using the same mechanism as the addition of conscience but with different criterion, it is possible to have the SOFM emphasize region of interest in the trajectory.

Topology Preserving Property. The learning law of Kohonen model not only drives the synaptic weight vector of the winner-takes-all neuron $\boldsymbol{i}(\mathbf{x})$ toward the current input $\mathbf{x}$, but also has the effect of pulling the synaptic weight vectors of the neighboring neurons along with $\boldsymbol{i}(\mathbf{x})$. As a direct consequence, a spatially ordered map emerges with a topology-preserving property (topographic map). As a topographic map, the spatial location of a neuron in the output lattice corresponds to a particular domain or feature of the input pattern. That is, nearby outputs correspond to nearby input patterns. Therefore neurons within a response region $E(\mathbf{x})$ of the input $\mathbf{x}$ bear certain similarities, and $E(\mathbf{x})$ naturally has a clustering tendency in the output lattice as shown in next section. Due to the smaller dimension of the output space, the clustered receptive fields usually appear with irregular shapes.

Generally the dimension of the input space X is greater than the dimension of the output space A. In this case, there exists no topological correspondence between them in the rigorous mathematical sense. For each input, however, the corresponding response region usually clusters in the output lattice. In terms of mapping between the input vector $\mathbf{x}$ and the synaptic weight vector $\mathbf{w}_i$, a dimension reduction is obviously involved converging and what is interesting is that the resulting feature map Φ is still able to form a

topographic representation of the input distribution in spite of the dimension mismatch. One example is the nonlinear projection method based on the Kohonen's SOFM dimension reduction as described by Kraaijveld, et al [58]. In their work, a large two-dimension Kohonen SOFM is trained over an input space with dimension as high as 10. When the input class labels are assigned to the output neurons after training, the nonlinear projection clearly shows that the data is clustered in terms of pattern similarity.

The topology-preserving property of SOFM is thus evident in the overall aim of the learning algorithm and can be summarized as follows [42]:

> *Approximate the input space X by pointers or prototypes in the form of synaptic weight vectors $\mathbf{w}_j$, in such a way that the feature map Φ provides a faithful representation of the important features that characterize the input vectors $\mathbf{x} \in X$.*

3.3.2 Simulations of SOFM with Temporal Signals

The topology-preserving by SOFM occurs in the case that the input is drawn from a spatial domain. When the input is drawn from a state space reconstructed using the method as discussed in Chapter 2, a transformation from temporal space into spatial space is involved. In this section, we applied SOFM to a chaotic signal process resulting from Lorenz system. The Lorenz signal will be used as testing signal in Chapter 6. Here the simulation is only performed to demonstrate how the inherent structure of this signal attractor is revealed by the SOFM topographic map.

The Lorenz system is composed of equations established in weather forecasting [67], which are given as

$$\begin{aligned} \dot{x} &= \sigma (y - x) \\ \dot{y} &= x (r - z) - y \\ \dot{z} &= xy - bz \end{aligned} \tag{48}$$

where σ, r, and b are constants. With $\sigma = 10$, $r = 28$, and $b = 8/3$, the system exhibits a chaotic dynamic. By numerical integration, the signal of x coordinates of the solution is obtained as shown in Fig. 3.7.

Using the delay coordinates method, a d-dimensional state vectors are obtained as

$$\mathbf{x}(t) = [x(t), x(t-\tau), \ldots, x(t-(d-1)\tau)] \tag{49}$$

where τ is a delay time. A 2-dimensional projections of the trajectories reconstructed from the Lorenz x-coordinate signal is as shown in Fig. 3.8.

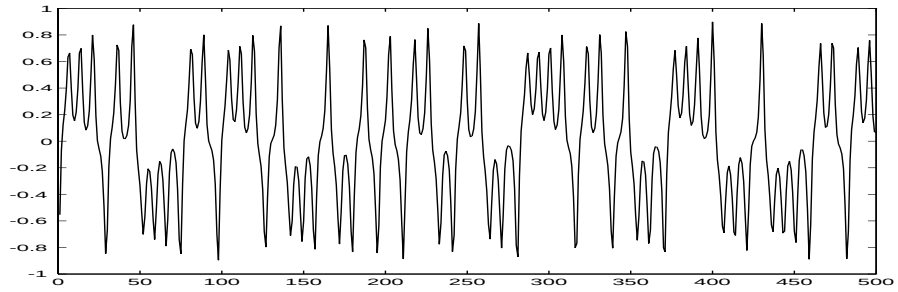


Fig. 3.7 x-coordinate signal of Lorenz equations

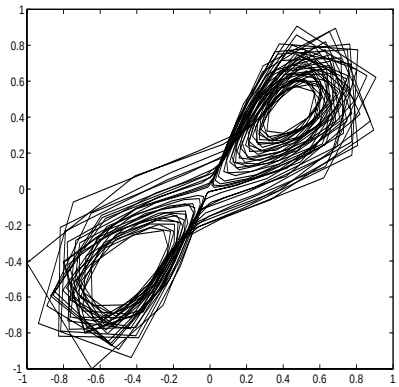


Fig. 3.8 A 2-dimensional projection of Lorenz x-coordinate

The simulation of the nonlinear projection is performed with the Kohonen SOFM network of the structure as shown in Fig. 3.1 (a). The output lattice is composed of 22x22 neurons. The network is first initialized with random weights. Training is performed over the state vectors $\mathbf{x}(t)$ reconstructed from a segment of 3000 data samples with the embedding dimension $d = 4$.

Let the learning curve is defined as the averaged deviation between the input $\mathbf{x}(t)$ and the winner-takes-all neural

$$\xi(n)^2 = \frac{1}{M}\sum_{j=1}^{M} \left\| \mathbf{x}_j - \mathbf{w}_{\boldsymbol{i}(\mathbf{x}_j)}(n) \right\|^2 \tag{50}$$

where n represents the index of epoch iterated. With $M = 3000$, the learning curve for 50 epoch is as shown in Fig. 3.9. After the initial cycles of large errors, the network begin to converge quickly. At the end of training, the averaged deviation $\xi(n)^2$ becomes approximately constant.

The averaged deviation as defined in (50) is a time-dependent quantity which reflect the overall closeness of winner-takes-all neurons to the input samples during the training process. Although the spatial order of the neural relation is not considered in this quantity, the decreasing trend significantly indicates the convergence of the whole structure

.

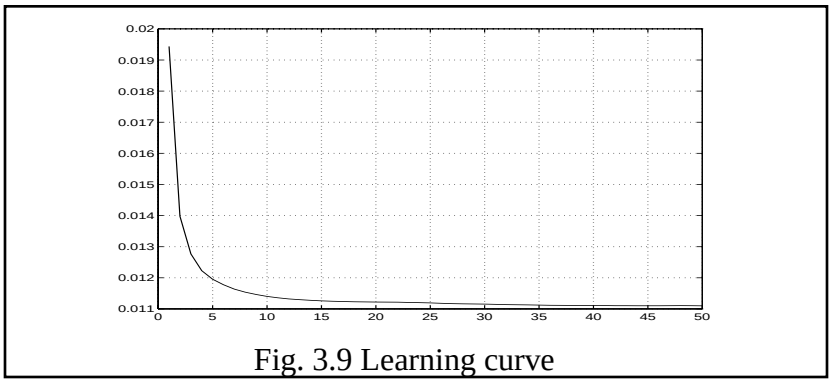


Fig. 3.9 Learning curve

A procedure for the evaluation of the neighborhood relations constructed by SOFM has been developed by Bauer, et al [13]. With that procedure a index quantity of performance named topographic product gives some significant indications. A vanishing value of the topographic product indicates a perfect neighborhood preservation while negative (positive) values indicate a too small (too large) output space dimensionality. This is an innovative procedure since it provides a quantitative measure of the topographic relationship in SOFM. Due to its complexity and subjective property, however, that procedure is not used in this simulation. Instead the neighborhood relationship can be visualized by the shape of the output trajectory.

After training, a 500 sample input are presented to the SOFM network. In Fig. 3.10, the small empty circles indicate the winner-takes-all neurons. By connecting them in accordance to the consecutive order, the output trajectory is thus formed. Comparing Fig. 3.8 and Fig. 3.10, it can be seen that the essential structure of the Lorenz attractor has been reflected by the output trajectory. This exemplifies the interesting property of topology-preserving.

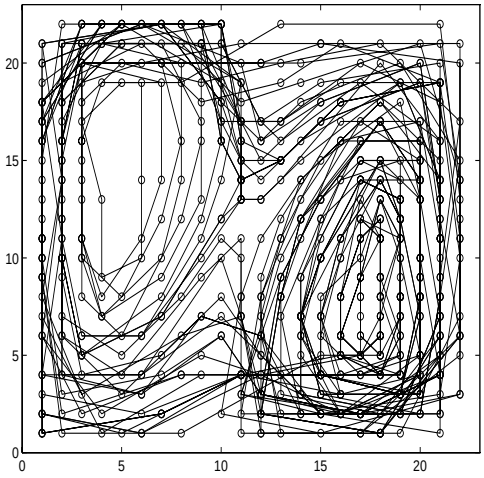


Fig. 3.10 Output trajectory

Corresponding to the input $\mathbf{x}$, the response region tends to a clustered structure. This can also be visualized from the structure of the output neural field. Let's define the Euclidean distance between two output neurons i and j as

$$\Upsilon_{ij}^2 = \|\mathbf{w}_i - \mathbf{w}_j\|^2, 1 \le i, j \le N \tag{51}$$

where $\mathbf{w}_i$ and $\mathbf{w}_j$ are the synaptic weight vectors respectively.

Consider the case that the neuron j is taken as the reference, a set of Υ_{ij} corresponding to each neuron is thus obtained. A contour plot is drawn by dividing all distance Υ_{ij} into multiple levels. Then the gradient lines display the contour levels of the similarity of all neurons toward the reference neuron j. The clustered structure of the response region centered at neuron $(8,12)$ is shown in Fig. 3.11, where 9 contour levels range from 0.1 to 1.54 with increment size of 0.18.

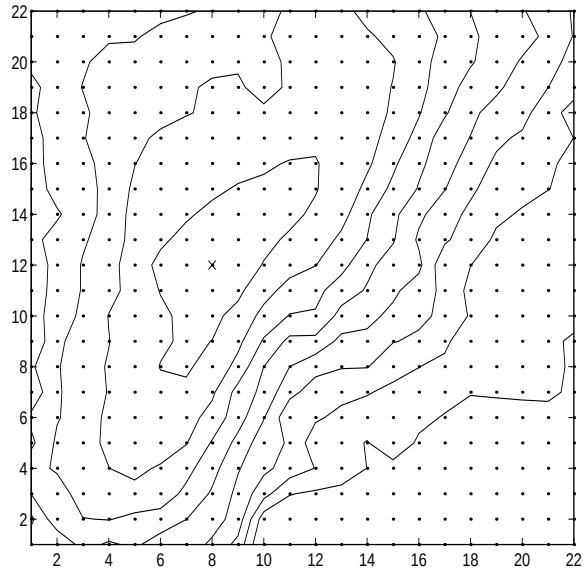


Fig. 3.11 Contour plot

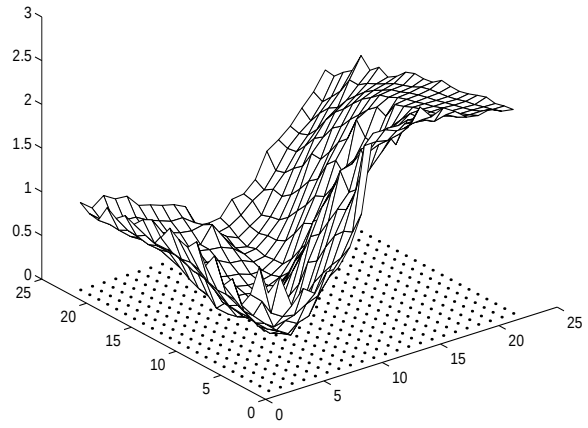


Fig. 3.12 Mesh plot

A corresponding mesh plot in Fig. 3.12, where the portion of the valley corresponds to the area enclosed by the inner contour lines. From the gradient descent structure of Υ_i over the output lattice, it can be seen that the mapping with the dimensional mismatch or dimension reduction is a nonlinear projection under the constraint that shape and position of the receptive fields of the neurons vary smoothly over the output lattice. The clustered structure of the response regions is another reflection of the topology-preserving feature of the SOFM.

3.4 Application Potential

The computational maps possess four prominent advantages [50]:

* *Efficient Information Processing.* Computational maps provide a method for the rapid sorting and processing of complex stimuli, and representing the results obtained in a simple and systematic form.

* *Simplicity of Access to Processed Information.* The use of computational maps simplifies the schemes of connectivity required to utilize the information by higher-order processors.

* *Common Form of Representation.* A common, mapped representation of the results of different kinds of computations permits the system to employ a single strategy for making sense of information.

* *Facilitation of additional interactions.* By representing a feature of interest in topographic form, maps enable us to sharpen tuning of the processor in ways that would not be possible otherwise. For example, regional interactions such as excitatory facilitation and lateral inhibitions can work only on sensory information that is mapped.

Kohonen's self-organizing feature map is a mathematical model of the computational map. Due to its parallel structure, competitive similarity matching, and the characteristic topology-preserving, the advantages of the computational maps are reflected by the use of Kohonen's Self-organizing feature map.

The characteristic properties of SOFM's make them an interesting tool for two distinct classes of applications:

1) Simulators used for the purpose of understanding and modeling of computational maps in the human brain;

2) Subsystems for practical applications such as robot movement, speech recognition, vector quantization, and adaptive equalization.

The first category includes the work by Cottrell and Fort [21], Ritter et al. [103]. One example is the simulation of the spatial structure of cortical feature maps by Obermayer [84]. In that simulation, the model is based on the Kohonen's self-organizing feature map algorithm, and it is shown that the observed structure of the topographic map can arise from a principle of continuous mapping. The patterns of orientation preference and selectivity generated by the model are similar to the patterns seen in the visual cortex of macaque monkey and cat. This correspond to a neural projection that maps a more than two-dimensional feature space onto a two-dimensional cortical surface under the constraint that shape and position of the receptive fields of the neurons vary smoothly over the cortical surface.

A large variety of applications of self-organizing feature maps have been reported in the literature for the second category which include control of robot arms [77], phonetic typewriter [54], vector quantization [68], adaptive equalization [56] and blind equalization [41], texture segmentation [85], radar classification of sea ice [86], clone detection in large telecommunication software systems [18], visualization of high-dimensional data [58].

Among these applications, SOFM are usually employed as a subsystems or infra-structure in one way or another. The prototypes derived by SOFM are accessed by higher-order processors. These higher-order processors are also constructed directly from the input space with the guidance of the organized SOFM. Based on the observation of the topology-preserving feature, dimension reduction, and statistical density approximation, as discussed previously, it is evident that the prototypes of each clustered response region represent the local structure of the input. In the application by Kraaijveld, et al. [58], for example, the underlying structure of the input data space such as the class patterns are graphically displayed by the simple interpoint distances in the SOFM feature space. Therefore it is reasonable to expect that some higher-order information can be extracted from these clustered receptive field. In another word, the higher-order processors can be constructed directly from the converged neural field. When the input space is recon-structed from a temporal signal process instead of spatial patterns, each of these higher-order processors naturally is a local representation of the temporal dynamics. This is the fundamental idea behind the scenario presented in the next Chapter.

CHAPTER 4
NON-LINEAR TIME SERIES MODELING WITH SOFM

4.1 Introduction

In Chapter 2, it has been assumed from the outset that signals in our analysis result from dynamical systems which can be expressed as differential equations or discrete-time evolution rules as depicted in equations (1) and (3). In most real-life problems, however, equations of motion are unknown. What we usually have is just a sampled output signal of a dynamical system. Therefore the task of dynamic modeling is to find $\tilde{F}$ as an approximation of the functional model of F to preserve the same dynamical properties implicit in the signal $\mathbf{x}(n)$.

As discussed previously, the approaches for nonlinear dynamic modeling are generally categorized as global and local models. The method of local models is closely related to differential topology [39] and is more general than the global approach in the sense that fewer statistical and geometric assumptions about the data are required. Consequently it can be applied to a wider class of dynamic system.

Local linear modeling is the simplest implementation of the local approach. However, using this method to deduce a set of local linear equations to approximate a chaotic dynamics has not been fully explored. In this chapter, a SOFM-based local dynamic modeling scheme is proposed. In the following sections, the idea of state-dependent AR models is first introduced, which simplifies the local linear approach of (14) into a AR model. Then the issues of dynamic modeling using a set of linear equations is discussed. Finally the SOFM-based modeling architecture is presented.

4.2 State-Dependent AR Models

As a member of the ARMA family as shown in equations (4) and (5), AR signal model can be expressed in terms of nth order difference equation

$$x(n+1) = \sum_{i=1}^{N-1} a_i x(n-i) + u(n) \tag{52}$$

where $x(n)$ is the output signal and the input, $u(n)$, is stationary white noise. Not only have these models well-suited for many signals of interest, but the linearity of the model also allows for simple analysis, especially when the mean-square error criterion is used.

In the scenario of localized linear prediction of chaotic time series [29], the state dynamics F of (3) is approximated over small regions of the attractor as locally linear functions with the assumption that F is locally smooth. Singer, et al., summarized the idea of local linear modeling as a state-dependent AR scheme. When the state vectors are constructed from the time series of a single variable by a delay embedding, the locally linear functions are actually reduced to the state-dependent AR models, and thus a code-book prediction scheme reminiscent of vector quantization is formed. This is composed of two parts, nonlinear autoregressive models and dynamic modeling with localized AR.

4.2.1 Nonlinear Autoregressive Models

A broad class of systems, including AR models, can be represented in a common state space form

$$\mathbf{x}(n+1) = F(\mathbf{x}(n), \mathbf{u}(n), n) \tag{53}$$

$$\mathbf{y}(n) = G(\mathbf{x}(n), \mathbf{u}(n), n) \tag{54}$$

where $\mathbf{x}(n) \in R^d$, $\mathbf{y}(n) \in R^p$ and $\mathbf{u}(n) \in R^q$ are the vectors of state, output and input respectively. When the system is linear time-invariant, the state equations (53) and (54) can

be reduced to

$$\mathbf{x}(n+1) = \boldsymbol{A}\mathbf{x}(n) + \boldsymbol{B}\mathbf{u}(n) \tag{55}$$

$$\mathbf{y}(n) = \boldsymbol{C}\mathbf{x}(n+1) + \boldsymbol{D}\mathbf{u}(n) \tag{56}$$

In the AR case, the matrix $\boldsymbol{A}$ in (55) takes the companion form

$$\begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ : & : & : & "" & : & : \\ 0 & 0 & 0 & \dots & 0 & 1 \\ a_{n-1} & a_{n-2} & a_{n-3} & \dots & a_1 & a_0 \end{bmatrix} \tag{57}$$

while matrices $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ become $\begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix}^T$, $\begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix}$, and 0 respectively.

Extending the above model to account for the nonlinear system, while retaining the companion state variable structure, lead to systems of dth order nonlinear difference equations

$$y(n+1) = F(y(n), y(n-1), \dots, y(n-d+1)) + u(n) \tag{58}$$

where F is the map $R^d \to R^1$. The state space representation can be viewed as a simple extension of (53) and (54), that is,

$$\mathbf{x}(n+1) = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ : & : & : & "" & : & : \\ 0 & 0 & 0 & \dots & 0 & 1 \\ a_{n-1} & a_{n-2} & a_{n-3} & \dots & a_1 & a_0 \end{bmatrix} \mathbf{x}(n) + \begin{bmatrix} 0 \\ : \\ 0 \\ F(\mathbf{x}(n)) \end{bmatrix} + \begin{bmatrix} 0 \\ : \\ 0 \\ 1 \end{bmatrix} u(n) \tag{59}$$

$$y(n) = \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix} \mathbf{x}(n) \tag{60}$$

equations (59) and (60) can be taken as a genearalized nonlinear autoregressive processes in terms of system state.

4.2.2 State-Dependent Prediction of Nonlinear Processes

From the Markov structure of non-linear AR processes, the statistics for $y(k+1)$ given its entire history depend only on the most recent d values:

$$Pr(y(n+1)\,|\,y(i), 0 \le i \le k) = Pr(y(n+1)\,|\,y(i), n-d+1 \le i \le n) \quad (61)$$

As shown in equations (59) and (60), the state vector, $\mathbf{x}(n)$, can be reconstructed from the observations of the scalar output,

$$\mathbf{x}(n) = [y(n-d+1), \ldots, y(n-1), y(n)] \quad (62)$$

The conditional statistics become

$$Pr(y(n+1)\,|\,y(i), 0 \le i \le n) = Pr(y(n+1)\,|\,\mathbf{x}(n)) \quad (63)$$

Therefore the minimum mean square error (MMSE) of $y(n+1)$ based on its entire history is

$$\tilde{y}(n+1) = E[y(n+1)\,|\,\mathbf{x}(n)] = E[F(\mathbf{x}(n)) + u(n)\,|\,\mathbf{x}(n)] = F(\mathbf{x}(n)) \quad (64)$$

Although $F(\mathbf{x}(n))$ is not available, the state dynamics of the system can be observed through

$$y(n+1) = F(\mathbf{x}(n)) + u(n) \quad (65)$$

Thus, given $y(k)$ and the state vector $\mathbf{x}(k)$ recovered in (62), the signal history represents

a set of noisy samples of $F(\mathbf{x})$, nonuniform distributed in state space. And consequently, the estimation for $y(k+1)$ becomes a problem of interpolating $F(\mathbf{x})$ from the measured noisy samples.

For the purpose of time series prediction, local linear models can yield precise short-term performance. Assuming only that the state dynamics are locally smooth, and given a long enough signal history, the samples near the present state of the system describe a local model of the state dynamics from which $F(\mathbf{x}(k))$ can be inferred. The strategy of predicting $x(k+1)$ given $x(i)$, $0 \le i \le k$ can be summarized as follows,

* Form a codebook of samples $[\mathbf{x}^T(i), y(i+1)]^T$ from the given signal process.
* Select samples $[\mathbf{x}^T(i), y(i+1)]^T$ from the constructed codebook such that

$$\| \mathbf{x}(n) - \mathbf{x}(i) \| \le r \tag{66}$$

where r is a pre-specified distance criterion parameter.

* Fit a local model $y(i+1) \approx \tilde{F}(\mathbf{x}(i))$ to the selected samples.
* Apply the local model to obtain

$$y(n+1) = \tilde{F}(\mathbf{x}(n)) \tag{67}$$

In general, the simplified nonlinear prediction is useful in modeling a variety of chaotic processes. In fact, Farmer and Sidorowich have used essentially such an approach for the prediction of chaotic time series [29]. This is actually a local interpolating method for state-dependent prediction without deducing the equations of motion to approximate the original dynamics.

4.3 Localized Linear Approximation of Global Dynamics and Limitation

If the underlying dynamics are sufficiently smooth, that we can closely represent $F(\mathbf{x})$ in the vicinity of $\mathbf{x}(k)$ by the first few terms of its multidimensional Taylor series

expansion, i.e.,

$$F(\mathbf{x}) = F(\mathbf{x}(k)) + \nabla F^T(\mathbf{x}(k))\,(\mathbf{x}+\mathbf{x}(k)) + O(F(\mathbf{x}(k))) \cong b + \mathbf{a}^T\mathbf{x} \qquad (68)$$

then $F(\mathbf{x})$ may be approximated as a linear function near $\mathbf{x}(n)$. Constructing a locally linear approximation $F(\mathbf{x}) \approx F(\mathbf{x}(k)) = b + \mathbf{a}^T\mathbf{x}$, amounts to fitting the parameters b and $\mathbf{a}$ to the selected pairs, $(\mathbf{x}(i), y(i+1))$, in the region of state space near $\mathbf{x}(k)$. The model will generally provide an overdetermined set of linear equations in the model parameters,

$$y(n+1) = \mathbf{a}^T\mathbf{x}(n) + b \qquad (69)$$

where $\mathbf{a} = [a_{d-1}, a_{d-2}, \ldots, a_0]^T$.

The above locally linear model is thus a generalization of the AR model of (57), with the matrix $\boldsymbol{A}$ varying as a function of the state. In addition, the local mean of the signal is accounted for by the constant b.

Though the above generalized AR model has been successfully utilized for the nonlinear time series prediction, The method of dynamic approximation using a finite set of local linear models has not been fully studied. For convenience, we call this method local linear dynamic modeling, which is based on the generalized local AR models. Instead of directly state dependent, however, the localized AR models are constructed for each individual partitioned neighborhood in the state space. The underlying dynamics $F(\mathbf{x})$ is then approximated as

$$F \approx \hat{F} = \bigcup_{k=1}^{L} \hat{F}_k \qquad (70)$$

and for an input $\mathbf{x}$,

$$F(\mathbf{x}) = \hat{F}_{i^*}(\mathbf{x}) = \mathbf{a}_{i^*}^T \mathbf{x} + b_{i^*} \tag{71}$$

where L is the total number of the partitioned subregions, and i^* is determined by competitive pattern selection of (25). That is, the global state dynamics F is represented collectively by a set of generalized AR models pieced together with each accounting for a subregion. The idea behind this method can be perceived by an example of second order system in Fig. 4.1, where the state space, X, is a plane with axes x_1 and x_2. The dynamics

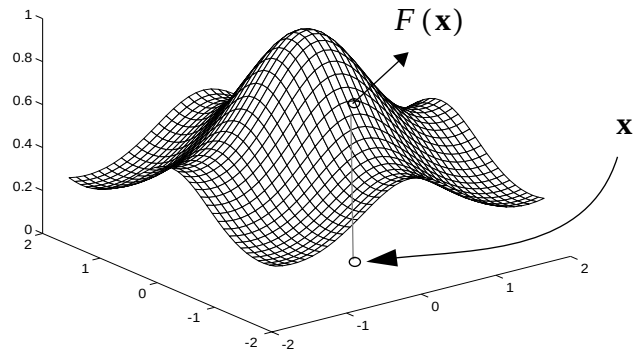


Fig. 4.1 (a) A nonlinear dynamics

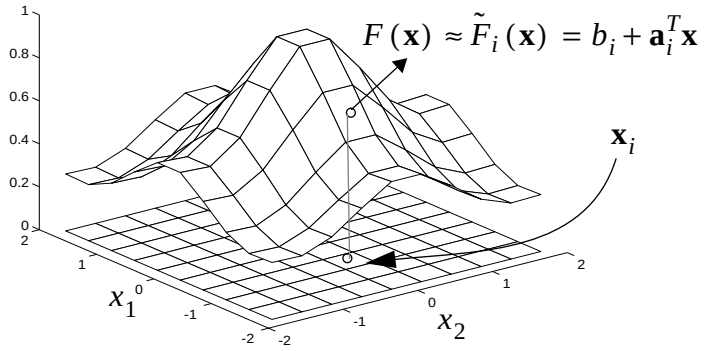


Fig. 4.1 (b) Locally linear state space model

$F(\mathbf{x})$ is a surface over the plane, X, as shown in Fig. 4.1 (a), which is partitioned into a set of small regions for the purpose of local modeling. For the local region i anchored at $\mathbf{x}_i$ as

shown in Fig. 4.1 (b), the local dynamics is approximated by a plane $\tilde{F}_i(\mathbf{x}) = b_i + \mathbf{a}_i^T\mathbf{x}$ tangent to $F(\mathbf{x})$ at $\mathbf{x} = \mathbf{x}_i$. In contrast, traditional AR modeling of the system would approximate the function $F(\mathbf{x})$ with a single plane through the origin. Obviously the locally linear models will reduce to the AR model when the local region extends to include all of the state space.

This method naturally involves two procedures to be performed: partitioning the state space and constructing the local models. The function of partitioning is to divide the state space into a set of subregions in a way that the local linear models can be constructed reliably for each of them. This may be accomplished by a vector quantization algorithm provided that the size of the subregions is approximately proportional to the statistical distribution of the signal after embedding in the state space. Based on the partitioned state space, the local linear models can be estimated over the samples composed of state vectors and the values to which they subsequently evolve to. Martinetz, et al., have used this scenario for the chaotic time series prediction, in which the signal space partitioning process is implemented with neural networks, and simultaneously the local model estimation is performed in a way of adaptive learning [78]. They have reported good performance of multiple-step prediction of chaotic time series. However, since each local linear model is primarily determined independently, discontinuity at the boundaries between the subregions can be expected. Therefore the discontinuity between the local models poses a fundamental limitation for the attempt of reliable description of global dynamics in the long-term sense.

The distribution of the data samples in the reconstructed state space is determined by the system dynamics. Since any practical signal sequence is always of finite length, the data samples may not regularly cover in the local area. The local linear models constructed from these irregularly scattered samples will be locally disturbed. This composes a major reason of discontinuity and error. Such irregular distributions can not be improved in the original state space as long as the signal is of finite length. However, the problem of irreg-

ular spacing of the data samples can be alleviated by a vector quantization procedure. The statistical distribution is usually preserved in the global sense by a vector quantization procedure, but the irregular spacing of the data samples can be "averaged out" through the data reduction.

Based the above observation, a SOFM-based local linear modeling network is proposed in the next section.

4.4 SOFM-Based Local Linear Modeling Networks

As discussed in Chapter 2, the self-organizing feature map is a localized representation of a signal constructed through competitive learning. Due to the use of the neighborhood function, the converged neural field bears a stronger global resemblance to the input space than other competitive learning. The positioning of each neuron is more strictly constrained by the overall statistical distribution of the signal, which helps to smooth out the irregular spacing tendency of the local data samples in the state space. Therefore the SOFM is employed as a modeling infrastructure to construct the local linear model.

4.4.1 Methodology

Based on these observations of the SOFM and the goal of local time series prediction, we develop a localized modeling scheme with the construction of a neural field embedding of the input space. The objective is to construct a neural architecture capable of capturing the underlying dynamics as measured by the two dynamical invariants (correlation dimension and Lyapunov exponent). In other words, the developed scheme is supposed to possess rich nonlinear dynamical behavior which is consistent with the long-term behavior of the given time series. This is fundamentally different from short-term prediction.

The basic idea is to embed the given input space into a compact neural field through Kohonen SOM algorithm. Then a simple local model estimation process is per-

formed to construct the linearized local models for each response region. The global description of the dynamics is composed of all these local models pieced together. The whole process is composed of two separate procedures: the embedding process of the input space into the neural field followed by the local model estimation. Correspondingly two requirements are imposed by this scenario such that the following information-theoretic rule of thumb is satisfied. For the embedding process, the information pertaining to the local model construction should be preserved optimally while such information should be used efficiently with a simple connection scheme during the local model estimation process.

4.4.2 Architecture

The architecture to be constructed is composed of three layers: input layer $\mathbf{x}$, neural field layer A, and the layer of local linear models $\tilde{F}(\mathbf{x})$ as shown in Fig. 4.2. The time series is embedded in a state space to create a state vector $\mathbf{x}$. The function $\boldsymbol{i}^*(\mathbf{x})$ constitutes a SOFM map Φ. That is, the input is fully connected to the nodes on the second layer

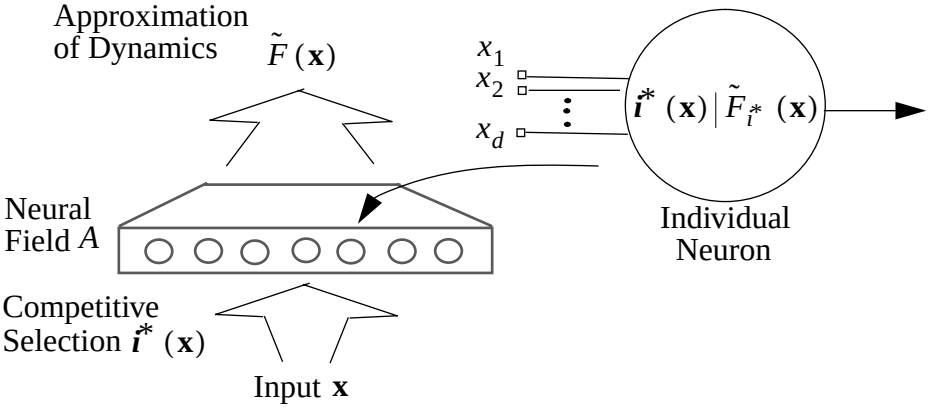


Fig. 4.2 The SOFM-based modeling architecture

through a set of weight vectors $\mathbf{w}_i$, and the winner-takes-neuron is identified by the competition. On the other hand, each neuron on the neural field layer corresponds to a specific processor $\tilde{F}_i$: $[\mathbf{a}_i, b_i]$ which represents the linear approximation of the local dynamics. Basically this structure can also be viewed as a gated-network with the gating mechanism defined by the competitive operation $i^*(\mathbf{x})$.

4.4.3 Network Construction

The employed SOFM performs two major functions: positioning the local models in the state space, and identifying the matched local model for the current input state $\mathbf{x}$. The former is completed during the training phase of SOFM, and the latter is performed during the modeling phase. The implementation issues of dynamical modeling will be discussed in Chapter 6. Here the network construction is considered.

The construction of the whole architecture is composed of three consecutive steps, as shown in Fig. 4.3: Reconstruction of the state space, mapping the state space in the SOFM neural field, and estimation of the local linear predictors.

Reconstruction of the state space from the training signal. Following the approach by Takens, a sequence of $d+1$ dimensional state vectors $[\mathbf{x}(n)^{\mathrm{T}}, x(n+\tau)]^{\mathrm{T}}$ is created from the given training time series, where $\mathbf{x}(n) = [x(n-(d-1)\tau), x(n-(d-2)\tau), \ldots, x(n)]^T$ and τ is the appropriate time delay where $d \geq d_A$ and d_A the dimension of the underlying dynamical process.

Mapping the state space in the neural field. This step is accomplished via the Kohonen learning process. With each vector-scalar pair $[\mathbf{x}(n), x(n+1)]$ presented as the input to the network, the Kohonen feature map learning algorithm adaptively discretizes the continuous input space $\boldsymbol{X} \subset \mathbf{R}^{d+1}$ into a set of disjoint cells $\mathbf{A}$ to construct the mapping Φ: $\boldsymbol{X} \rightarrow \boldsymbol{A}$. This process continues until the learning rate decreases close to zero and the neighborhood function covers one output unit. After learning, a neural field representa-

tion $\boldsymbol{A}$ of the input space $\mathbf{X}$ via the constructed mapping relationship Φ is formed in terms of a set of disjoint units topologically organized in the output space.

Estimation of the locally linear predictors. For each neuron $u_i \in \boldsymbol{A}$, its local linear predictor in terms of $[\mathbf{a}_i^T, b_i]$ is estimated based on $\alpha_i \subset \boldsymbol{A}$, which is a set of L_i neurons in the neighborhood of u_i including u_i itself. Each of them has a corresponding weight vector $[\mathbf{w}_{i_j}^T, w_{i_j}(d+1)]^T \in \mathbf{R}^{d+1}$, where $\mathbf{w}_{i_j}^T = [w_{i_j}(1), w_{i_j}(2), \ldots, w_{i_j}(d)]$. The local prediction model $[\mathbf{a}_i^T, b_i]$ is fitted in the least-square sense to the set of weights in α_i, i.e.

$$w_{i_j}(d+1) = b + \mathbf{a}^T \mathbf{w}_{i_j} \tag{72}$$

After the above construction procedure, a modeling network is obtained with a global functional map composed of a set of local linear equations

$$x(n+1) = \tilde{F}_i(\mathbf{x}(n)) = \mathbf{a}_i^T \mathbf{x}(n) + b_i \tag{73}$$

where i is the winner-takes-all neuron identified by competition of (25).

In general, the above procedure differs from others in the following aspects: a) instead of recursive prediction with state-dependent local linear models, we construct concurrently a finite set of local linear equations to approximate the global dynamics; b) Instead of modeling the input only in the temporal sense, the proposed method is constructed and performs a temporal-spatial description of the dynamics; c) Instead of using directly the signal history, the local linear equations are simply estimated from the embedding neural field with each neuron as a pointer to the response region of the input; d) As a consequence, the neural field is an explicit quantification of the dynamics and thus becomes an infrastructure for local model construction.

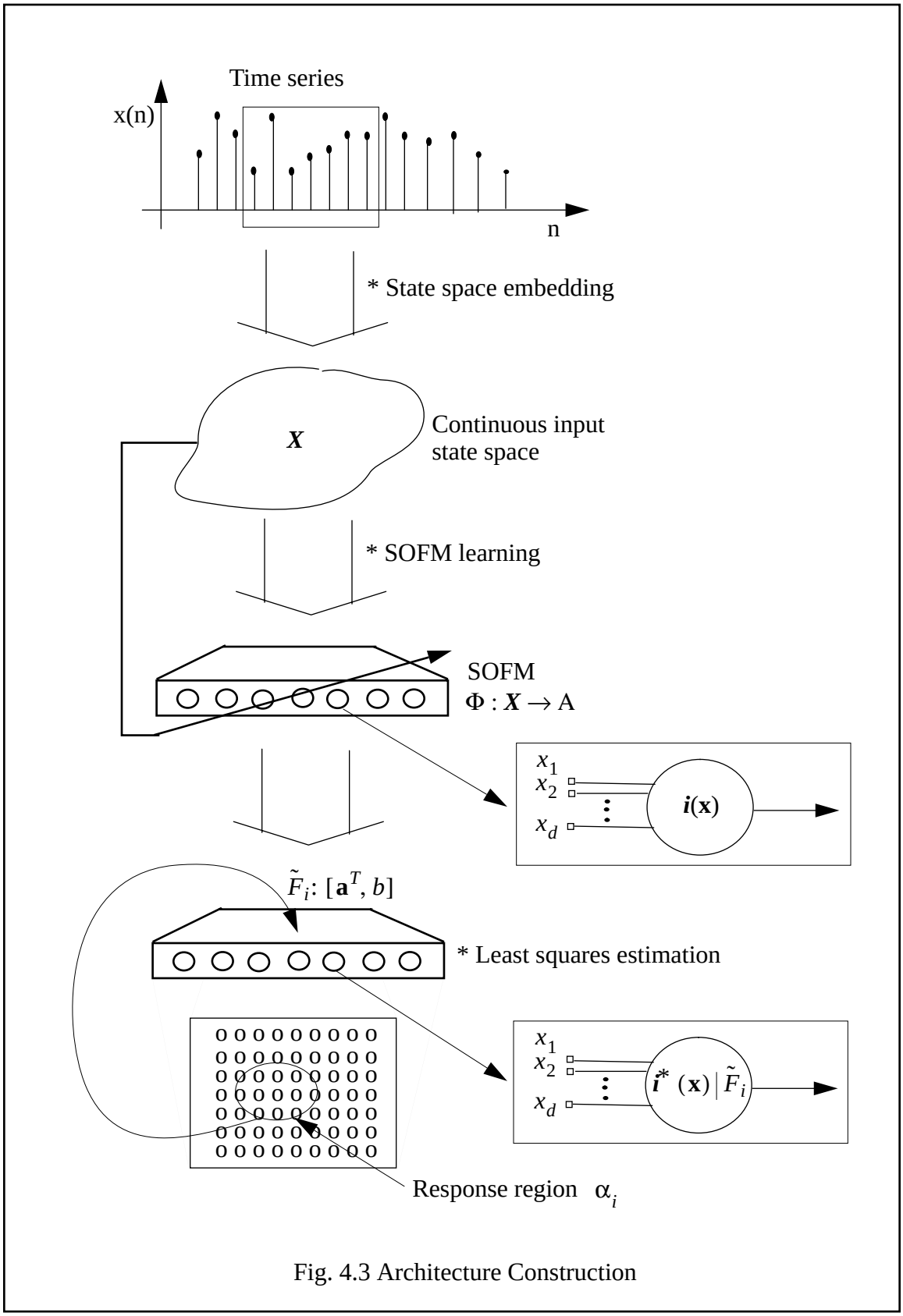


Fig. 4.3 Architecture Construction

CHAPTER 5
MODIFICATION OF LEARNING EQUATIONS

The task of chaotic time series modeling is to deduce effective equations of motion from observations of time-dependent behavior. These equations of motion represent a vast reduction of a chaotic data set's observed complexity to a compact, algorithmic specification. The idea behind the dynamic modeling procedure proposed in the last chapter is that the global dynamics is approximated by a finite set of local linear models, which has not been fully studied. In this chapter, the properties of the proposed modeling scenario are first reviewed. Then the methodology is expressed in form of matrix to exemplify the advantage of using SOFM. Finally the potential of network is explored which leads to two improvements.

5.1 Properties of the SOFM-Based Local Modeling

The SOFM-based dynamic modeling network proposed in Chapter 4 represents a new exploration of the local linear predictive modeling. The significance is that the equations of motion is obtained for the modeling, instead of just using local data for state-dependent interpolating. Therefore it is a kind of local linear atlas, and is characterized by the following aspects.

Approximation**.** Although the input domain and the range of the map are assumed noncountable and continuous subsets of Euclidean space respectively, the feature map as the infrastructure is countable and discrete. The current dynamic pattern is identified by the SOFM and the corresponding local linear model matched. Therefore the global dynamics are approximated by a finite set of local linear models. This is in contrast with the other local linear modeling scenarios.

Reduced discontinuity. Due to the neighborhood function, the feature map neurons are more deeply constrained toward the overall input data distribution. The local linear models, constructed directly from the converged neural field, will not be perturbed by the irregular spacing of the input data, which reduces the discontinuity between the local models. Although it is not known how close we can approach toward the seamless concatenation of local linear models, the proposed method represents one step ahead towards this direction.

The processing units**.** In addition to the SOFM, an additional layer of processing units is appended. These units have a one-to-one connection with the corresponding neural field neurons, and compose the localized linear models. The discrete map is complemented by local linear interpolation to compose an approximation of continuous functional maps. When a diverse set of linear processing units are pieced together, it is reasonable to believe that the underlying dynamics is not linear.

The network construction**.** The construction is composed of two subsequent processes. The feature map is obtained with the Kohonen learning law, and the corresponding local linear models are constructed from the corresponding response regions using the least square algorithm. This procedure reflects two different objectives: the feature map optimally preserves the structure of the input space while the least square algorithm extracts the model information efficiently.

Data Reduction. From Luttrell's derivation, it can be seen that SOFM is a vector quantization procedure. The construction of the feature map Φ is actually a data-reduction process. The estimation of the local models becomes simpler since smaller set of data is involved in the computation. In addition, the problem of the irregular distribution of the local data samples is alleviated.

5.2 Improved Estimation of Local Linear Models

The estimation of local models from the constructed SOFM neural field A, as presented in section 4.4.3, can be expressed in the matrix form, from which the advantage of using SOFM as the modeling infrastructure can be easily perceived.

The estimation of the local linear model F_i, as depicted in equation (73), can be generalized as

$$\mathbf{y}_i = \mathbf{H}_i \Theta_i \tag{74}$$

$$\mathbf{y}_i = \begin{bmatrix} w_{i_1}(d+1) \\ \circ \\ \circ \\ \circ \\ w_{i_{L_i}}(d+1) \end{bmatrix}, \mathbf{H}_i = \begin{bmatrix} [\mathbf{w}_{i_1}^T, 1] \\ \circ \\ \circ \\ \circ \\ [\mathbf{w}_{i_{L_i}}^T, 1] \end{bmatrix}, \Theta_i = \begin{bmatrix} \mathbf{a}_i \\ b_i \end{bmatrix} \tag{75}$$

The least squares solution of $[\mathbf{a}_i^T, b_i]$ is thus obtained as

$$\Theta_i = [\mathbf{a}_i^T, b_i]^T = \mathbf{H}_i^* \mathbf{y}_i = (\mathbf{H}_i^T \mathbf{H}_i)^{-1} \mathbf{H}_i^T \mathbf{y}_i \tag{76}$$

where $\mathbf{H}_i^*$ is the generalized inverse of $\mathbf{H}_i$. Equation (76) takes the form of Wiener-Hopf equations. $\mathbf{H}_i^T \mathbf{H}_i$ and $\mathbf{H}_i^T \mathbf{y}_i$ are equivalent to approximations of autocorrelation matrix and cross-correlation matrix respectively.

These two matrices can be estimated from the data samples in the local neighborhood of the input space X. As discussed in Chapter 4, a practical signal sequence is of finite length, and thus the data samples may not regularly cover the local area. The matrices $\mathbf{H}_i^T \mathbf{H}_i$ and $\mathbf{H}_i^T \mathbf{Y}_i$ can be perturbed by such irregular distribution, which will lead the least square solution Θ_i deviate from the optimal estimation. This composes a major reason of discontinuity between local linear models. Such discontinuity can give rise to

undesired behavior when the constructed local linear models are pieced together as a whole to approximate the global dynamics.

Using SOFM as a modeling infrastructure in the way as proposed in Chapter 4, the input space is represented by a compact set of neural weight vectors. During the training of SOFM, all the neural weight vectors are adapted toward the input space in the global sense. Especially the neighborhood function employed in the Kohonen learning algorithm constrains the weight vectors to better conform to the true distribution in term of neural resolution. Using the procedure in form of equations (74) to (76), more stable $\mathbf{H}_i^T\mathbf{H}_i$ and $\mathbf{H}_i^T\mathbf{Y}_i$ can thus be obtained, which in turn leads Θ_i more closer to the optimal estimation.

In the global sense, this is significant since the discontinuity between local linear models can be alleviated, and therefore a reliable approximation of the underlying dynamics can be achieved. This advantage is actually the direct result of global learning of SOFM, which smooth-out the irregular distribution trend of the original input. Another advantage of using SOFM is the flexibility of its learning rule, which is presented in the next section.

5.3 Dynamic-Oriented Representations

For different applications, the preserved information should be conceived with different emphasis. For the purpose of dynamic modeling, the emphasis should be given to the dynamic representation, which is not taken into consideration in the original SOFM since it is a neighborhood preserving vector quantizer [12]. Using the Kohonen learning rule, the feature map will approach a statistically optimal codebook in a smooth way, better than other suboptimal competitive quantization. The obtained prototypes collectively represent a compact statistical distribution image of the specified application domain.

When the same learning rule is employed to construct a modeling infrastructure, as considered in this research, the SOFM algorithm may not end up with the desired features. This is due to the fact that the adaptation of the feature map is independent of the local

model construction, and only statistical features are involved in the learning process. Based on this observation, it can be seen that it is necessary to explore the potential of SOFM in the way that the learning process of the feature map is tied up with the local modeling.

5.3.1 The Constraints of the SOFM Learning Process

The basic aim of the SOFM algorithm is to store a large set of input vectors by finding a smaller set of prototype vectors that represent the input in the statistical sense. The theoretical basis of this observation is rooted in the vector quantization theory with the motivation of data reduction or compression [36]. Actually the LBG algorithm is closely related to the SOFM algorithm as explained before.

The statistical variations of the SOFM neural field is mainly reflected by the magnification factor $m(\mathbf{x})$, as defined in equation (44). To optimally exploit the potential of the SOFM, approaches have been proposed to modify the SOFM learning algorithm to obtain a feature map with desired property of $m(\mathbf{x})$.

It has been hypothesized that the feature map transfers the maximum amount of information about the input data ensemble when the magnification factor is proportional to the input pdf $Pr(\mathbf{x})$, as illustrated in equation (45). According to the simulation results reported in the literature, however, the relationship between $m(\mathbf{x})$ and $Pr(\mathbf{x})$ has to be generalized as

$$m(\mathbf{x}) \propto Pr(\mathbf{x})^{\mu} \tag{77}$$

where μ takes the form of the magnification exponent which deviates from the information theoretically optimal value $\mu = 1$ as well as from the values that optimize the mean square distortion error ($\mu = 1/3$ for one-dimensional maps). Thus an optimization of neural maps with regard to various distortion error measures would require a control of the

map magnification exponents. Adding conscience to the competitive learning mechanism is an example improve the density-matching property. Recently a more advanced method has been proposed to control the magnification factor of the SOFM [12]. The idea behind this method is to involve input data density in the learning process, which is called node-dependent adaptability. By changing a single parameter, maps with optimal information transfer, with various minimal reconstruction errors, or with an inverted magnification can be generated [12].

Therefore the magnification factor is an important feature of the SOFM. However all the methods only concentrate on statistical considerations in such a way that the magnification factor is tied up to the input data density. To construct a dynamic modeling infrastructure, the SOFM learning rule should be modified to allow the magnification factor reflect the dynamic fluctuations of the input.

5.3.2 Dynamic Learning Rule

As discussed in Chapter 3, the magnification factor $m(\mathbf{x})$ denotes the number of neurons in a small volume $d\mathbf{x}$ of the input space $\boldsymbol{X}$. Alternatively, this property can be interpreted in terms of an excitation pattern in the neural field. For the SOFM to conform with the input stimulus density, equation (46) must hold. To associate the magnification factor with the overall objective of dynamic modeling, a dynamic complexity is introduced as the new condition in equation (46). Let us see what should be the form of the new term. The idea is very simple: more complicated parts of the trajectory should be represented by spatially larger neighborhoods, i.e., the magnification factor $m(\mathbf{x})$ should be larger. More precisely, for inputs $\mathbf{x}_i$ and $\mathbf{x}_j$, the following relation holds if the underlying dynamics at $\mathbf{x}_i$ is more complicated than that at $\mathbf{x}_j$,

$$N(E(\mathbf{x}_i)) > N(E(\mathbf{x}_j)) \tag{78}$$

As defined in equation (35), an identical distance metric r_E has been assumed for $E(\mathbf{x}_i)$ and $E(\mathbf{x}_j)$. If different distance metric r_{E_i} and r_{E_j} are assumed for $E(\mathbf{x}_i)$ and $E(\mathbf{x}_j)$ such that,

$$N(E(\mathbf{x}_i)) = N(E(\mathbf{x}_j)) \tag{79}$$

it is also true that

$$r_{E_i} < r_{E_j} \tag{80}$$

When the local models are constructed from $E(\mathbf{x}_i)$ and $E(\mathbf{x}_j)$ respectively, the more complicated dynamics at $\mathbf{x}_i$ is accounted for by a linear equation over smaller local regions than that for $\mathbf{x}_j$ in the input space $\boldsymbol{X}$. This is the scenario of the dynamic-oriented neural representation, which obviously can not be realized unless the SOFM learning rule is modified.

As shown in equation (26), only two time dependent parameters can be considered for the purpose of such modification. They are the neighborhood function $\Lambda(\mathbf{r}_{\boldsymbol{i}_n(\mathbf{x})}, \mathbf{r}_j)$ and the learning rate $\eta(n)$.

The role of the a neighborhood function is essentially to correlate the directions of the weight updates to achieve the positive lateral feedback in the output lattice. According to the analysis of Luttrell, gradually decreasing the neighborhood function is critical for this purpose [69]. As shown in equation (41) and Fig. 3.5, the neighborhood function is equivalent to the signal-independent noise, which is introduced to account for the distortion in the SOFM. Thus changing the adaptive schedule of the neighborhood function will interfere with the convergence of the SOFM, which is not appropriate. Therefore the only choice for the purpose of modification is the learning rate $\eta(n)$.

From both (26) and (41), it can be seen that larger learning rate exerts larger adaptation for the area covered by the neighborhood function. If a larger learning rate is selec-

tively applied to a certain input pattern, a stronger response region corresponding to that input will be formed exactly as derived. The remaining issue is to find the “driving force” to guide the learning rate.

Since each local linear model is obtained by taking the first two terms of multidimensional Taylor series expansion of equation (68), the sum of the remaining higher-order terms is a good indication of the fitting performance. This quantity is actually the prediction error

$$\varepsilon(\mathbf{x}(n)) = F(\mathbf{x}(n)) - \tilde{F}_i(\mathbf{x}(n)) = x(n+1) - (\mathbf{a}_i^{\mathrm{T}}\mathbf{x}(n) + b_i) \tag{81}$$

Large value of $\|\varepsilon(\mathbf{x}(n))\|$ indicates a large curvature at $\mathbf{x}(n)$, which needs a large response region such that the constructed local linear model accounts for the underlying dynamics over a smaller local area in the input space $\boldsymbol{X}$.

For unit-magnitude learning rate, a straightforward way to involve $\|\varepsilon(\mathbf{x}(n))\|$ in the SOFM learning process is

$$\eta_\varepsilon = \frac{1 - exp(-\mu(\eta + \bar{\varepsilon}))}{1 + exp(-\mu(\eta + \bar{\varepsilon}))} \tag{82}$$

where μ is a constant that controls the slope of the dynamic range, and η is the conventional SOFM learning rate. One of the time-dependent form for η is given in equation (32). Moreover a normalized value of $\|\varepsilon(\mathbf{x}(n))\|$ is used to offset the signal amplitude variation,

$$\bar{\varepsilon} = \frac{\|\varepsilon(\mathbf{x}(n))\|}{\|x(n+1)\|} \tag{83}$$

The functional relationship between η_ε and $\eta + \bar{\varepsilon}$ with $\mu = 2$ is shown in Fig. 5.1, where the dotted line represents the linear reference. The advantage of (82) is apparent: the dynamic learning rate η_ε satisfies the unit magnitude constraint, and simulta-

neously it linearly reflects the variation of the prediction error within the fine-tuning range of [0, 0.5].

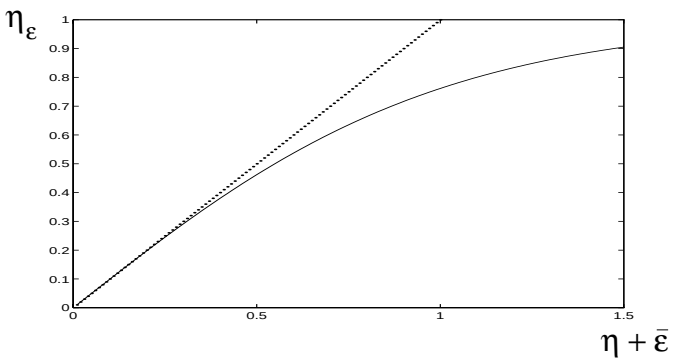


Fig. 5.1 Dynamic learning rate function

When $\bar{\varepsilon}$ is small, η_ε adhere to the conventional SOFM learning step size. When $\bar{\varepsilon}$ is large, however, a reinforced adaptation is performed on the neighborhood that corresponds to large-curvature dynamics. This learning process will lead to a feature map with a selective magnification factor consistent with the local complexity of the underlying dynamics. This modified Kohonen learning procedure is thus called dynamic learning (DL). DL is guided by the information regarding the local dynamics other than the local input data density, therefore this procedure is fundamentally different from either the addition of conscience or the node-dependent adaptability.

With dynamic learning, the adaptation process of the feature map is still composed of two stages: the ordering phase and convergence phase. During the ordering phase, η is coupled with a possible large, but erratic $\bar{\varepsilon}$, and the topological ordering of the neural field takes place. Simultaneously, large values of $\bar{\varepsilon}$ enforce the adaptation over the matched neuron and its neighbors. The property of the dynamic-oriented magnification factor is mainly formed during this stage. As the learning proceeds, $\bar{\varepsilon}$ decreases on average. During the convergence phase, iterations of the algorithm performs fine tuning of the feature map. Due to small $\| \mathbf{x} - \mathbf{w}_j(n) \|$ and the small neighborhood size, the convergence will not be

drastically disturbed by the sporadic surge of the prediction error $\bar{\varepsilon}$, although the training process may not be smooth.

The significance of the dynamic learning can also be perceived from the system architecture. With the dynamic learning, the construction of the feature map is not independent of the overall modeling objective. Due to the direct involvement of the prediction error in the learning process, the training process becomes a close-loop procedure as indicated by the dark block in Fig. 5.2, which effectively results in a dynamic-oriented neural field.

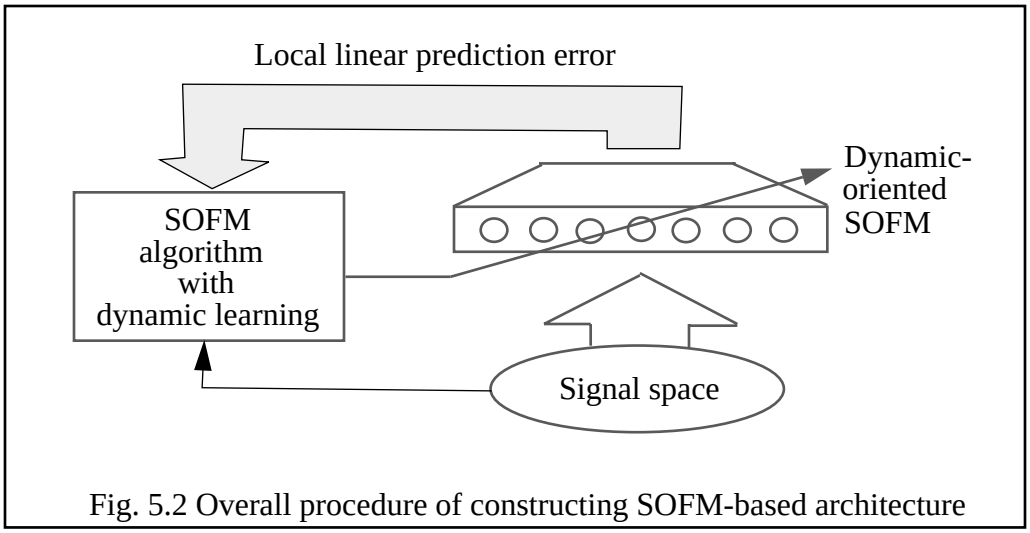


Fig. 5.2 Overall procedure of constructing SOFM-based architecture

Moreover the idea behind the dynamic-oriented feature map represents a new exploration of SOFM beyond the solely optimal representation in the statistical sense. Based on this token, more potential may be exploited by involving right information in the learning procedure.

5.4 The Weighted Least-Squares Solution

In the SOFM-based dynamic modeling, the neural field excitation pattern is considered for the locally linear model estimation. This makes the static SOFM mapping a plausible dynamical modeling infrastructure. The assumption behind this scheme is that

each response region is sufficiently small such that the higher order component of the given dynamics can be ignored, and the local dynamics can be accounted for by a local tangent plane. The determination of local linear models depends on the prototypes in the respective response regions, assuring that the collective response of each region is uniform (i.e., all neurons have an equal response). This may not be an optimal choice due to their deviations from the region center.

Instead, allowing a diversified response according to the distribution among the matched response region is expected to provide better approximation performance. One plausible way is to make the contribution of each neuron to the collective response inversely proportional to its deviation to the metric center. A scaling procedure is needed for that purpose, which leads to the weighted least square solution. Based on (74), the weighted least square solution is mathematically equivalent to insertion of a weighting matrix in the least squares optimization,

$$min\,(\mathbf{y}-\mathbf{H}\Theta)^{T}\mathbf{S}\,(\mathbf{y}-\mathbf{H}\Theta) \tag{84}$$

where $\mathbf{S}$ is a nonsingular symmetric matrix. The weighted least squares solution is required to satisfy

$$\mathbf{H}^{T}\mathbf{S}\,(\mathbf{y}-\mathbf{H}\Theta) \;=\; 0 \tag{85}$$

Therefore the solution becomes

$$\tilde{\Theta}_{WLS} \;=\; (\mathbf{H}^{T}\mathbf{S}\mathbf{H})^{-1}\,(\mathbf{H}^{T}\mathbf{S}\mathbf{y}) \tag{86}$$

and it is unique if and only if $\mathbf{H}^{T}\mathbf{S}\mathbf{H}$ is invertible. Using the Euclidean metric, the distance between neuron j and the winner-takes-all i^{*} is computed

$$d_j = \|\mathbf{w}_{i^*} - \mathbf{w}_j\| \tag{87}$$

A straightforward way is to select $\mathbf{S}$ as a diagonal matrix,

$$\mathbf{S} = \{s_{pq}\}\big|_{1 \le p,\, q \le N_r} \tag{88}$$

where N_r is the number of neurons considered for the estimation of the local linear model, and $s_{pq} = 0$ for $p \neq q$, and

$$s_{pp} = 1 - \frac{d_p^m}{\sum_{k=1}^{N_r} d_k^m} \tag{89}$$

That is, the weighting is solely determined by the diagonal elements of $\mathbf{S}$. For the neuron p, the weight coefficient is s_{pp}. Smaller d_p corresponds to larger s_{pp}, which means that a neighboring neuron that is closer makes a larger contribution to the determination of the local model, and vice versa. This functional relationship is illustrated in Fig. 5.3, where $N_r = 14$, and the value d_p ranges from 0.01 to 1.4. It can be seen that the integer m controls the scaling. Larger m will sharpen the contribution scaling.

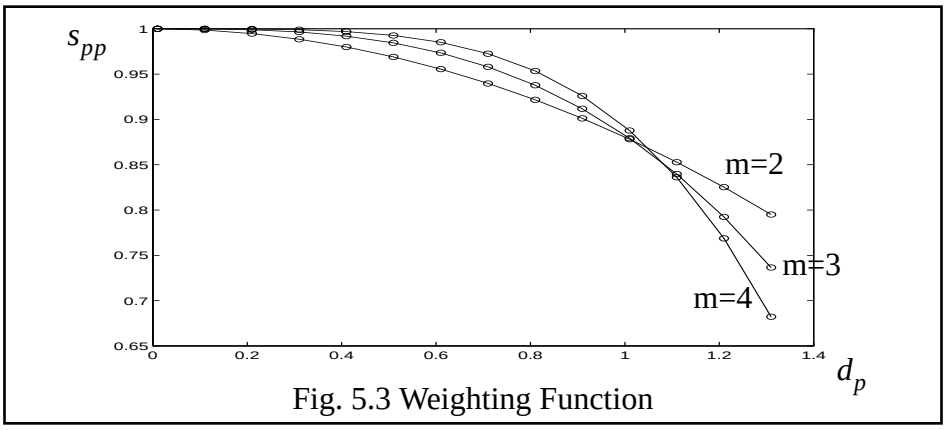


Fig. 5.3 Weighting Function

When $\mathbf{S} = \mathbf{I}$, the regular least square solution is obtained. Therefore (86) is a generalized least squares solution.

On the other hand, the least square solution does not deviate from the statistical sense. Actually it can be derived from statistical consideration [114]. For example, the least squares solution in the linear signal-plus-noise model can be obtained in the case that the error is assumed a realization of a normal random vector. Based on the maximum likelihood theory, the optimal estimator is obtained in form of a generalized least squares solution involving weighted component as the covariance matrix.

Here the least squares method is applied to the construction of the local models from the corresponding response regions. Basically we lack the information about the local statistical distribution of the underlying dynamics. The determination of (89) is mainly based on experimental results. The improvement will be addressed in the next chapter.

CHAPTER 6
EXPERIMENTAL RESULTS

In Chapter 4 and 5, the SOFM is explored as a modeling infrastructure, and a corresponding set of local linear models are directly estimated from the SOFM neural field. When an input **x** is applied to this architecture, one of these local models is identified by the feature map as the local dynamic map. With all these local models pieced together, a global dynamic representation is thus obtained, which is here called SOFM-based dynamic modeling.

In this chapter, the SOFM-based modeling is validated with both equation-based time series and real-world signals. Instead of just local mapping capability, the major interest of this research is the approximation of the global dynamics carried by the chaotic time series. It will be shown that the SOFM-based modeling is capable of preserving the original dynamics. The result of the experiment also demonstrates that the SOFM-based modeling system is not hindered by the discontinuity between local linear models, although piece-wise linear modeling was deemed impossible in the past.

Based on the derivation in previous chapters, the SOFM-based modeling is established following the procedure shown in Fig. 6.1. By involving the proposed dynamic learning rule in the feature map construction, the procedure constitutes a close-loop implementation, which results in a dynamic-oriented SOFM neural field as discussed in Chapter 5. Using the weighted least square estimation, the local models are estimated directly from the converged SOFM neural field.

In all the experimental examples, a square output lattice is used with the configuration of $N \times N$ neurons, or N rows by N columns of neurons arranged at the grid nodes. To demonstrate the performance improvement achieved by the dynamic learning,

SOFM will be trained using both regular learning (RL) and dynamic learning (DL) procedures respectively, and the resulting networks, which are referred to as RL network and DL network for convenience, are compared with respect to their modeling performances. In addition, the overall performance will also be compared with the local linear models constructed using direct least squares and weighted least squares solutions respectively.

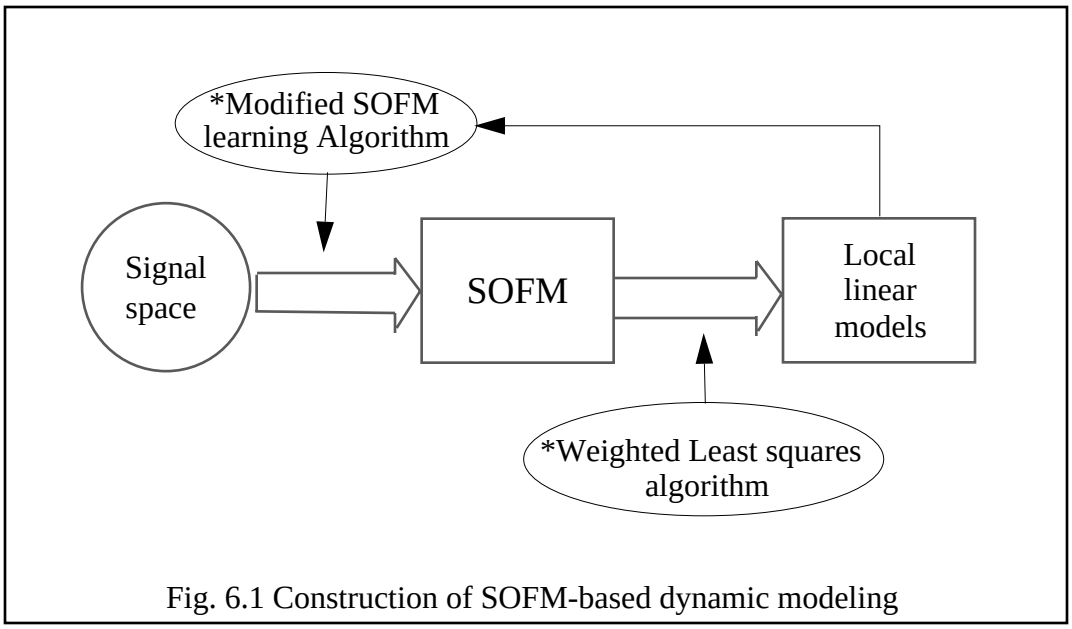


Fig. 6.1 Construction of SOFM-based dynamic modeling

In section 6.1, the modeling network is tested with chaotic signals numerically integrated from two mathematical models. These two signals exhibit chaotic behaviors of different degrees: the Mackey-Glass time series is characterized by a relatively small Lyapunov exponent while the Lorenz time series possesses large Lyapunov exponent. Several aspects relating to the proposed networks are also discussed and demonstrated with the testing results. Since the final state of SOFM network is dependent upon the initial state of weight vectors, the consistency of the proposed modeling scenario is tested with the result demonstrated in section 6.2. In that section, the temporal consistency is also tested to prove that the SOFM-based dynamic modeling is a stable system which maintains the same

dynamic behavior over time. Finally the experiment is extended to the real-world signals in section 6.3, where performance improvement by the dynamic learning rule are manifest.

6.1 Modeling of Numerically Generated Time Series

Due to the existence of positive Lyapunov exponents, the prediction of the chaotic signal will inevitably diverge from the original signal. Only one-step and multi-step prediction can be evaluated. A criterion to compare one-step prediction has been established in the literature [19], while multi-step prediction is also widely utilized to compare the short-term modeling performance [61]. According to the analysis of [98], one-step prediction error is not a good indication of how well the predictors learned the trajectory or dynamics in state space. Since the multi-step prediction is usually produced by the predictive models iterated with the predicted values, a generalized mean square error criterion is suggested by Principe et al. [98] as

$$\zeta_f^2(k) = \frac{1}{(M-k)\,\sigma^2} \sum_{n=N}^{N+M-k} \left(x[(n+k)\tau] - \left(-\tilde{F}_N\{\hat{x}[(n+k-1)\tau],\ldots,\hat{x}[(n+k-d)\tau]\}\right)^2\right. \tag{90}$$

$$\hat{x}[(n-i)\tau] = \begin{cases} x[(n-i)\tau],\ i=0,1,\ldots,d-1 \\ \tilde{F}_N\{\hat{x}[(n+k-1)\tau],\ldots,\hat{x}[(n+k-d)\tau]\},\ i=-1,-2,\ldots \end{cases} \tag{91}$$

where the σ^2 is the estimated variance of the time series $x(n)$, $\tilde{F}$ is the estimated map of the underlying dynamics, M is the number of testing samples, k is the prediction steps, d is the model order, and τ is a delay parameter. As discussed in section 2.4.3, however, the average multi-step prediction error criterion belongs to the category of the sample-by-sample comparison, and thus only local approximation of the model can be gauged.

Considering that the major interest of this research is dynamic modeling of chaotic time series, the evaluation of the modeling performance should not depend on these two

short-term prediction error criterion. Instead the two dynamical invariants discussed in section 2.4 are utilized for this purpose based on the test model as shown in Fig. 6.2, where the modeling network is iterated as an autonomous dynamical system.

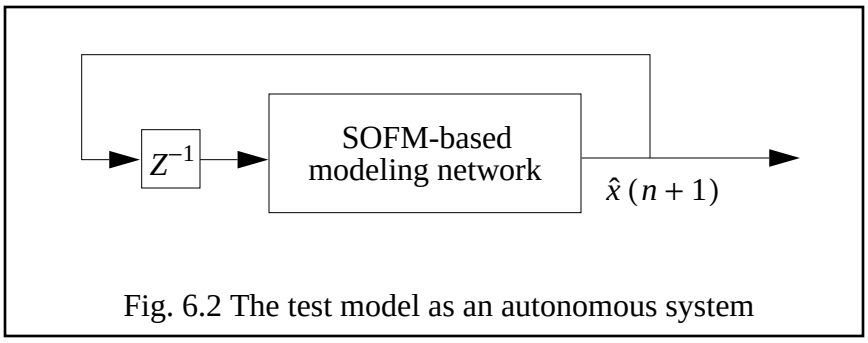


Fig. 6.2 The test model as an autonomous system

The largest Lyapunov exponent and correlation dimension as described in Chapter 3 are estimated from the generated time series. When these two invariants of the iterative output $\hat{x}(n)$ is consistent with the original time series $x(n)$, one can say that the underlying dynamics have been captured by the SOFM-based modeling map $\tilde{F}$.

To construct an infrastructure with a self-organized neural field suitable for the purpose of local modeling, a modified Kohonen learning rule is proposed in Chapter 5. In stead of a strictly decreasing learning rate, a dynamic learning rate is used as shown in (45). That is, the learning rate is formed with a decreasing trend coupled with the prediction error which is estimated over each matched response region. Since such prediction error reflects the metric topology with respect to the local dynamics, the training of SOFM is thus systematically linked to the overall objective of dynamic modeling as depicted by the connection between the block of local linear model and the modified SOFM learning algorithm. This gradually leads the SOFM converging to a final state consistent with the overall modeling objective. In this section, the regular SOFM learning rule is also implemented as an counterpart to demonstrate the advantages of the dynamic learning rule.

6.1.1 Mackey-Glass Time Series

Low dimensional deterministic dynamics can produce complex behavior as chaotic time series is characterized by positive Lyapunov exponents. This feature indicates that the underlying system is so sensitive to the initial condition that it is impossible to reproduce the same time series even with an exact model. In the long term sense, any negligible deviation in specifying the initial condition can lead the output of the model to diverge from the signal.

In this experiment, the SOFM-based architecture is tested using the Mackey-Glass time series which is characteristic of a small Lyapunov exponent. The Mackey-Glass system is described by the differential equation

$$\frac{dx}{dt} = -0.1x(t) + \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} \tag{92}$$

where the delay τ controls depth of underlying chaotic motion. In this research, the dynamics with $\tau = 30$ is considered. The signal is obtained by integrating the equation (92) with the 4th order Runge-Kutta method at a step size of 1. The integrated signal is then down-sampled by 6 and normalized to the range of $[-1, 1]$. The resulted signal and its spectrum is as shown in Fig. 6.3.

To model the Mackey-Glass time series with the SOFM-based network, the state space embedding dimension d is determined from the obtained time series as discussed in Chapter 2. Using the properties of the correlation function $D(r)$, the embedding dimension is identified by the value of the trial d where the structure in $D(r)$ becomes independent of d. The correlation integral map (CIM) curves and the corresponding slopes are computed as shown in Fig. 6.4. The reconstruction dimension d varies from 2 to 10 by an increment of 2. It can be seen that the slope of the CIM curves saturates when the reconstruction dimension goes over 6. Operationally this value is identified as the embedding dimension for the model order. In addition, Fig. 6.4 also shows that the

estimated correlational dimension for the numerically obtained Mackey-Glass time series is 2.70±0.04.

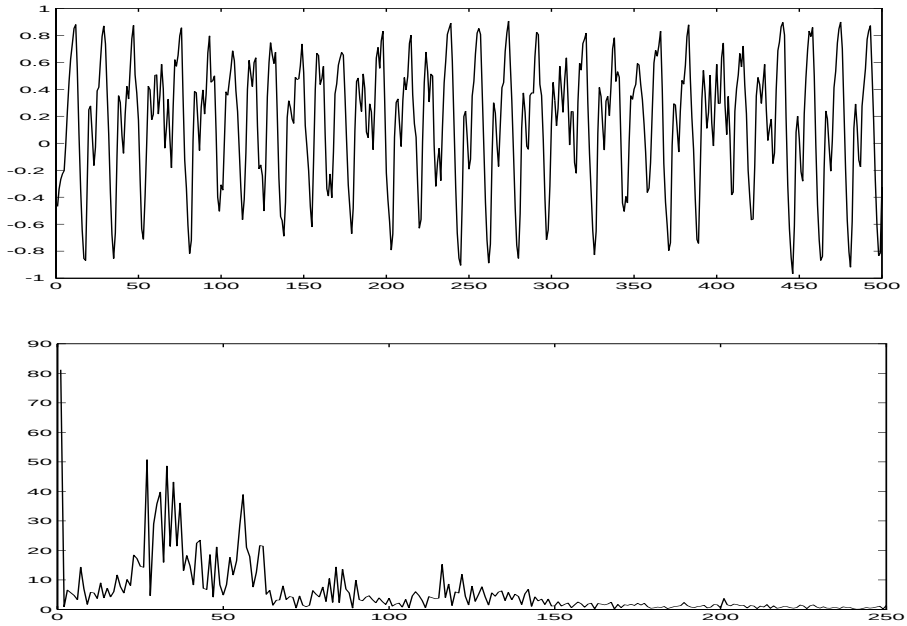


Fig. 6.3 Mack-Glass signal: waveform and spectrum

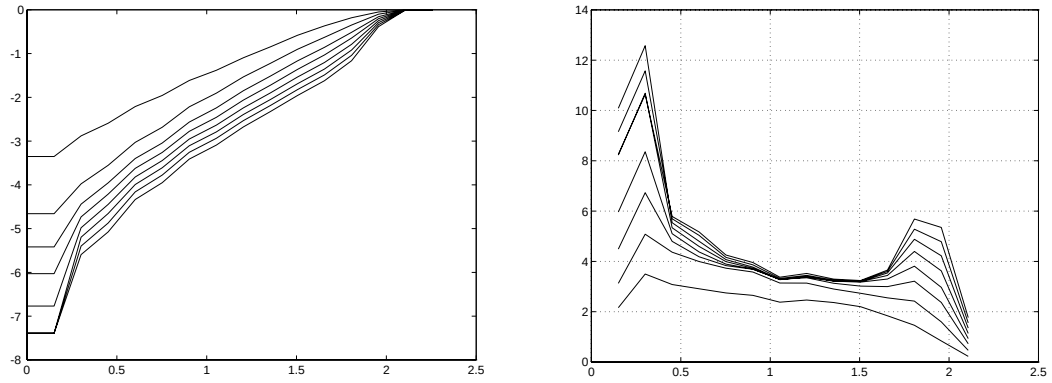


Fig. 6.4 CIM curves and the slope estimation for Mackey-Glass signal

Two SOFM networks of dimension 22x22 are trained over 3000 samples with regular learning and dynamic learning respectively. The time-dependent parameters for the learning rate and evolution of the neighborhood function in (33) and (34) are chosen as

$a_\eta = 1$, $b_\eta = 1/500$, $c_\sigma = 1/8$ and $d_\sigma = 1/4000$. In addition, $\mu = 2$ is chosen for the slope in the dynamic learning process. The training process is stopped after 110 epochs. The learning curve, as defined in (50) in terms of the averaged deviation between the input and the winner-takes-all neuron, is shown in Fig. 6.5. The regular learning takes the same time-dependent parameters without the prediction error involved. The final averaged deviation for the dynamic learning is 0.0185. The learning curve is still decreasing at 110 epochs, the regular learning process is continued for another 50 epochs.

In Fig. 6.5, only the training process of epoch 11 to 100 are shown to illustrate the fine detail. It can be seen that the dynamic learning process converges faster than the regular learning process. This is due to the fact that the instantaneous prediction error can be thought of as an additive noise source which provides momentum, speeding up the convergence.

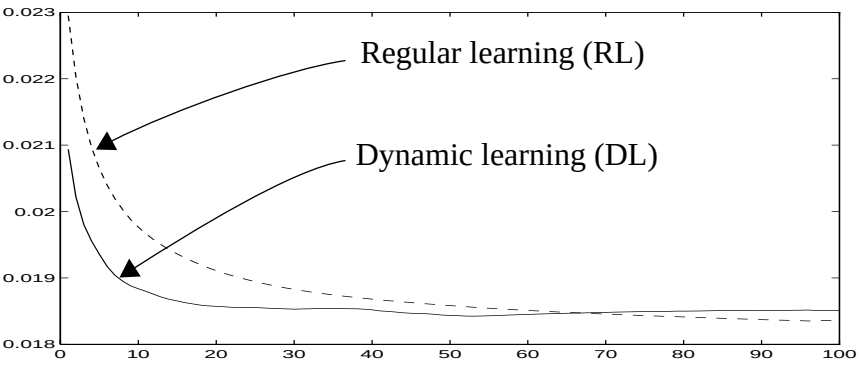


Fig. 6.5 The learning curve

Since the learning process involves a shrinking neighborhood function, faster convergence may not be easily obtained by simply using larger learning rate so the erratic momentum is an asset. On the other hand, it can also be noted that the curve with dynamic learning is not strictly decreasing. After reaching the minimum, it begins an increasing trend at slower rate and finally approaches a value larger than that obtained by the regular learning. This is not an unexpected phenomenon. Recall that the objective of the dynamic learning is to obtain

a neural field with the resolution which reflects not only the distribution of the input space, but also the local complexity of the input dynamics. Therefore the averaged deviation between the input and the winner-takes-all neurons may be larger than that of the regular learning, and the converged map is thus not optimum in the statistical sense.

From the above observation, it is certain that the minimum point of the learning curve is not a reliable indication of stopping the training. The experience shows that the dynamic learning SOFM network with the training process stopped at the minimum valley of the learning curve provides a similar modeling performance to that of regular learning SOFM.

To ensure the convergence of feature map with the dynamic SOFM learning, the adaptive schedule of the neighborhood function remains the same as with regular SOFM learning. The experiment shows that the neighborhood function $\Lambda(\mathbf{r}_{i_n(\mathbf{x})}, \mathbf{r}_j)$, as defined in (28), should cover at least a region including at least the closest neighbors of the winner-takes-all neuron as long as the value of the overall learning rate is larger than 0.1.

Based on the converged SOFM networks, the set of local linear models are constructed from the converged neural fields. In the weighted least square solution of (88) and (89), $m = 4$ is used for the weighting scale. For convenience, the networks with regular learning and dynamic learning are called RL network and DL network respectively.

Table 6.1 Comparison of One-step Prediction Errors

Technique	$\log_{10}(\zeta_1)$
Adaptive LMS	-1.68
Radial Basis	-1.60
TDNN	-1.54
SOFM Local linear	-1.29
Local linear	-1.24

Using (22) with $k = 1$, the average one-step prediction error was computed. Table 6.1 shows the comparison of the one-step prediction errors for different techniques obtained over a test set of 500 samples of the Mackey-Glass system. From Table 6.1, it may be concluded that the constructed SOFM-based network is not a good model of the underlying dynamics because it provides the largest one-step error. This is actually not true, as shown later. Small one-step prediction error is just a a necessary condition for better long-term prediction.

To evaluate the performance of long-term prediction, the multi-step mean squared prediction errors are estimated using (90), where $\zeta_f(k)$ is computed over $M = 1000$ prediction samples. The error curves for the DL and RL modeling are shown in Fig. 6.6.

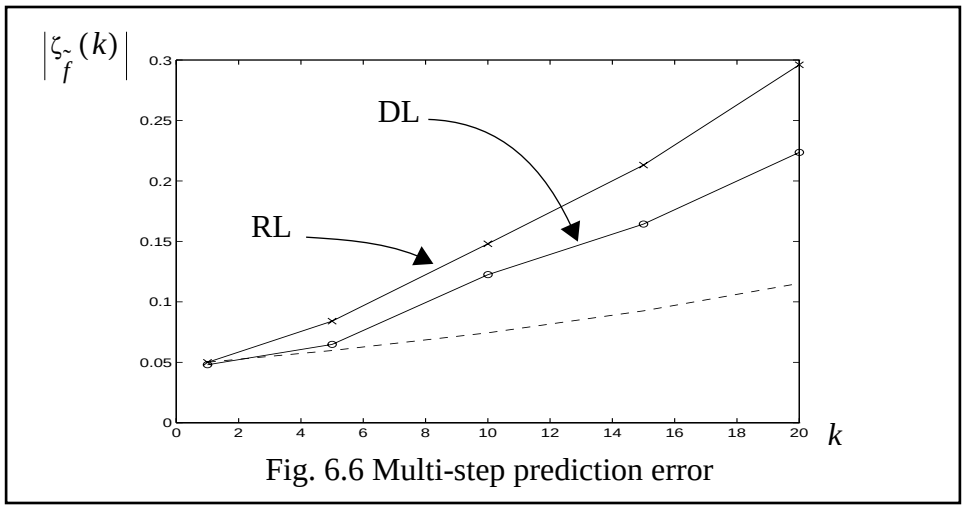


Fig. 6.6 Multi-step prediction error

The broken line in Fig. 6.6 is the Casdagli's conjecture curve computed by

$$\varepsilon(i) = \varepsilon_0 e^{\lambda'(i\Delta t)} \tag{93}$$

where $\varepsilon_0 = 0.0480$, $\Delta t = 6$ and the largest Lyapunov exponent $\lambda' = 0.0073$. Here it can be seen that the prediction errors of the RL modeling diverges faster than that of the DL modeling though their 1-step prediction error are almost the same. Moreover, the multi-step prediction error curve of the DL network is close to the Casdagli's conjecture curve up to

5-step prediction. However, large deviation between them begin to appear when the prediction goes over 5 steps. This is an example that the Casdagli's conjecture curve is only an indicator of performance for short-term prediction. On the other hand, what is illustrated in Fig. 6.6 is actually not the best the best multi-prediction performance. As shown later, the short-term prediction can be improved with larger dimension of SOFM. However, the DL modeling with the SOFM configuration of $22 \mathrm{x} 22$ output lattice is capable of successfully capturing the underlying dynamics in the global sense, as illustrated next.

To test the modeling performance of the constructed structures in the global sense, the DL and RL modeling networks are iterated as autonomous systems in the way as shown in Fig. 6.2. That is, each of them is seeded with a state vector from the input space and iterated by feeding back the output to the input for 5000 samples. Fig. 6.7 and Fig. 6.8 are the waveform and the spectrum of the first 500 data samples of iterative sequences by DL and RL modeling respectively. Compared with the original signal as shown in Fig. 6.3, it can be seen that the output of the DL network is quite close to the original Mackey-Glass signal. In Fig. 6.8, however, certain regularity and periodic tendency can be noted in the output of the RL modeling network, which is characterized by spurious peaks in the corresponding spectrum. This demonstrates that the dynamics captured by the RL network is simpler than that preserved by the DL network. The same observation can also be made from Fig. 6.9, where the trajectories of both the original signal and the iterative outputs of both DL and RL networks are reconstructed in the 2-dimensional space using the delay coordinate method.

The topology-preserving feature of SOFM can be perceived by the mapping output trajectory. Fig. 6.10 illustrates a trajectory of the winner-takes-all neurons for the 500 sample autonomous iteration of the DL network. The similar spatial pattern, although in a 2-D space, shows that the essential structure of the original signal as shown in Fig. 6.8 (a) has been revealed by trajectory of the winner-takes-all neurons.

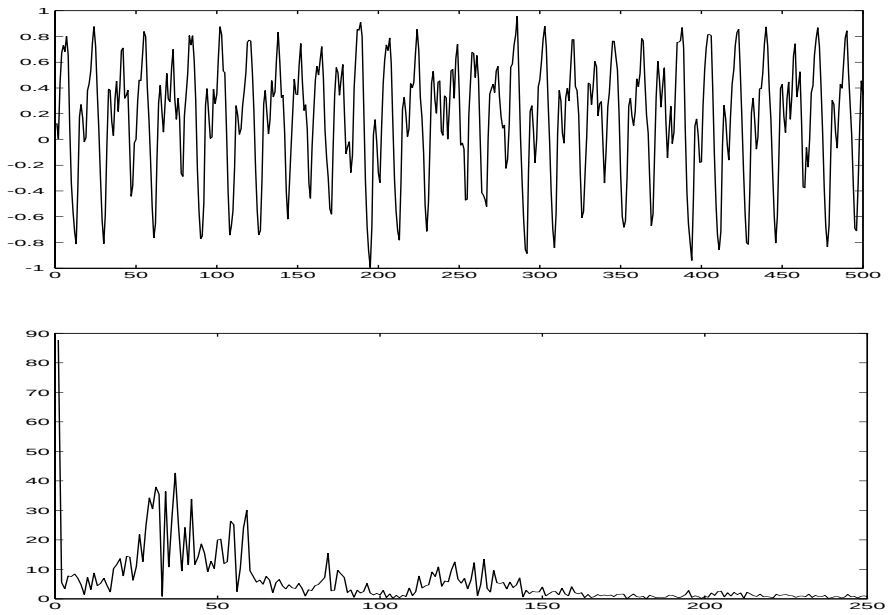


Fig. 6.7 Autonomous output of the DL network: waveform and spectrum

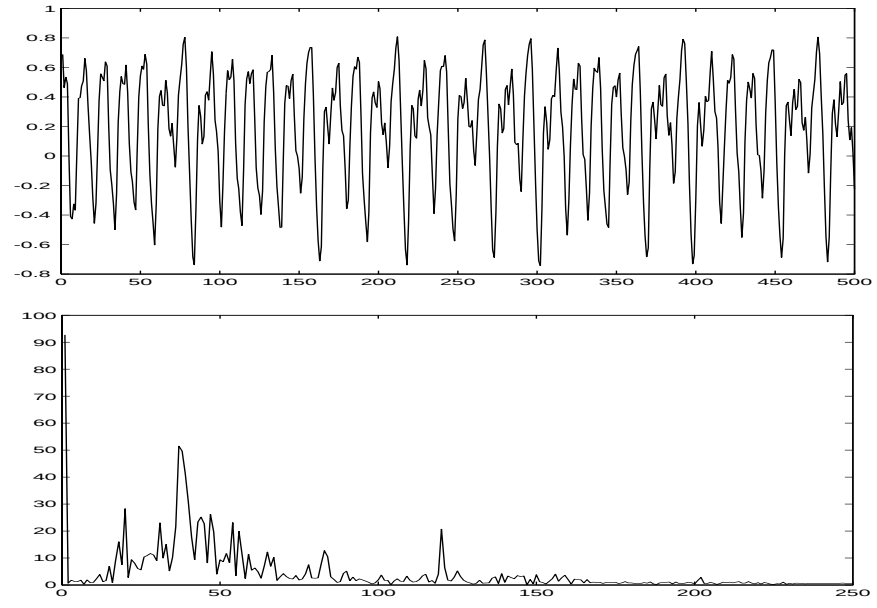
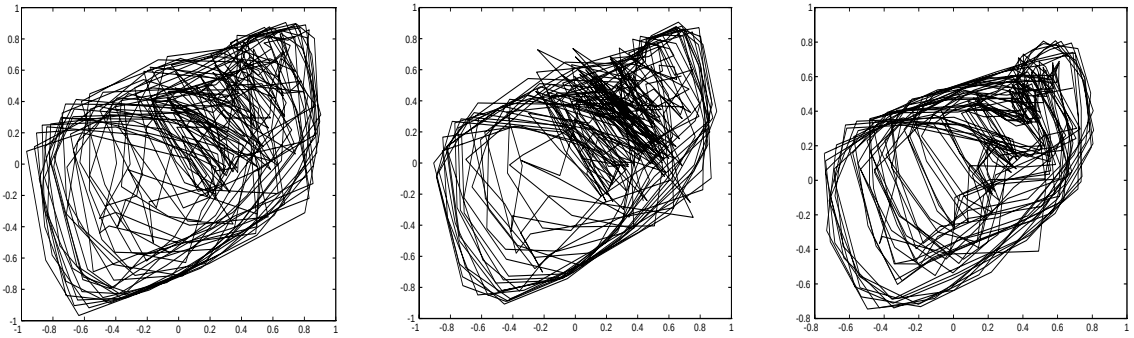


Fig. 6.8 Autonomous output of the RL network: waveform and spectrum



(a) Original signal (b) Output of DL network (b) Output of RL network

Fig. 6.9 Reconstructed trajectories

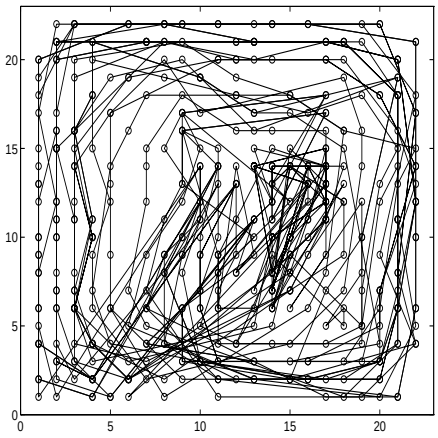


Fig. 6.10 The trajectory of winning neurons

As discussed in Chapter 2, the multi-step prediction error can only examine the local approximation of the model. To validate the constructed model, the dynamical invariants are estimated and compared for the iterative outputs of DL and RL modeling. Using Wolf's procedure, the largest Lyapunov exponents of the original Mackey-Glass time series and the obtained two iterative outputs are computed as shown in Fig. 6.11. In Fig. 6.12, the correlation integral maps and the corresponding slopes are computed and displayed.

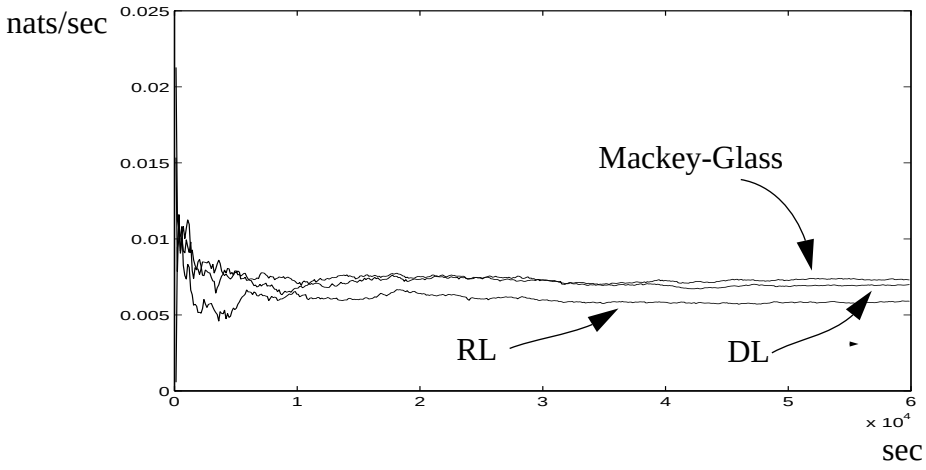
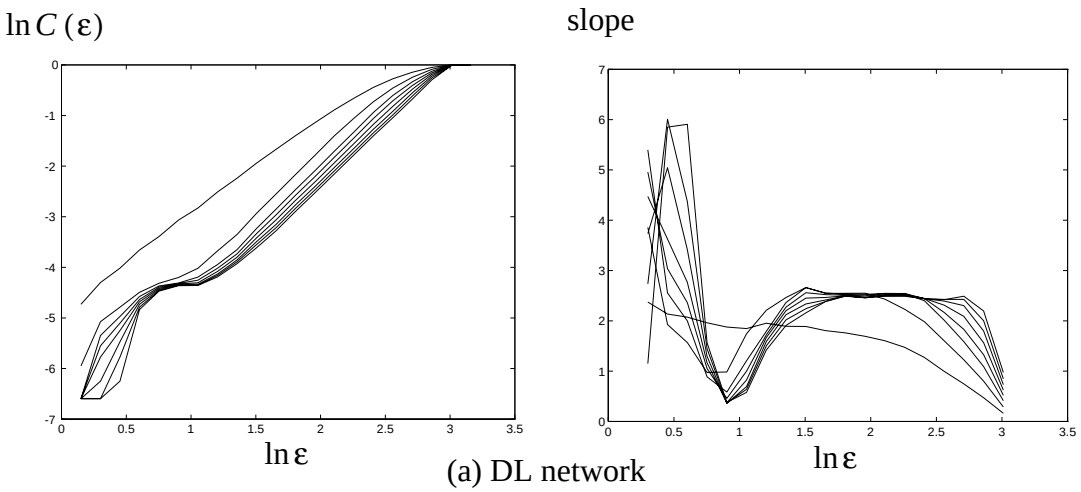
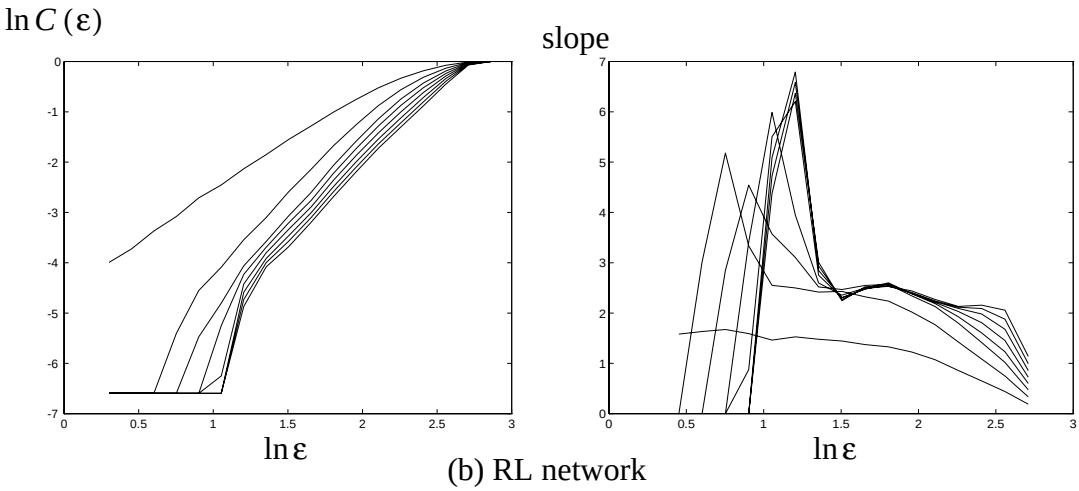


Fig. 6.11 Lyapunov exponent estimation



(a) DL network



(b) RL network

Fig. 6.12 CIM curves and the slope estimation

The estimated values of the dynamical invariants are listed in Table 6.2. It can be noted that the dynamics preserved by DL network possesses a divergence rate and correlation dimension closer to that of the original Mackey-Glass signal, which demonstrates that DL network is a better modeling structure than RL network.

Table 6.2 Estimated dynamical invariants

Time Series	Largest Lyapunov Expo.	Correlation Dim.
Mackey-Glass	0.0071 ± 0.0002	2.69 ± 0.04
DL Network	0.0073 ± 0.0002	2.68 ± 0.03
RL Network	0.0059 ± 0.0004	2.60 ± 0.05

From the discussion in Chapter 5, it is known that the performance of local linear models depends on the resolution of the neural field, which is directly related to the pre-specified dimension of the output lattice. Based on this observation, six SOFM networks with different lattice dimension $N \mathrm{x} N$ are trained over 3000 sample of Mackey-Glass time series with the proposed dynamic learning rule, and 10-step prediction error ζ_{10} is computed for each of the constructed modeling networks. As a function of the dimension index N the estimated errors are shown in Fig. 6.13.

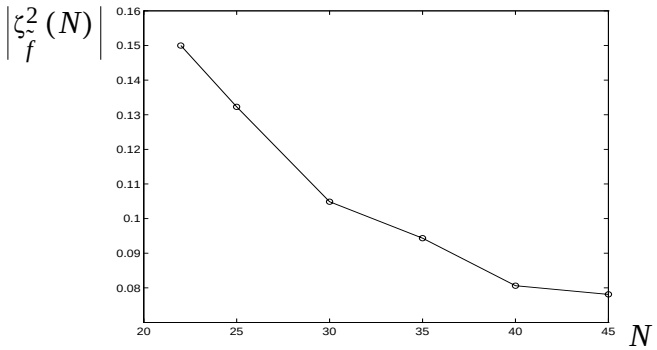


Fig. 6.13 10-step prediction error vs. network dimension

As expected, the 10-step prediction error decreases with the larger network dimension. Therefore large network dimension provides a better modeling infrastructure. To achieve a good modeling map for a more complicated dynamics, a larger network dimension is a natural choice. For the example of Mackey-Glass time series considered here, it has been shown that a SOFM with a 22×22 output lattice is sufficient for the DL modeling to successfully preserving the global dynamics.

As discussed in Chapter 5, least square solution is not an optimal method to estimate the local models directly from the finite neural field. Weighted least square solution is proposed as an improvement. To see the performance improvement by the weighted least solution, the local linear models are constructed from DL and RL neural fields using conventional least squares and weighted least squares solutions respectively. The largest Lyapunov exponents are then estimated over the iterative outputs using these techniques. The results of the estimation are listed in Table 6.3, where "*" indicates the local linear models constructed using the conventional least square solution.

Table 6.3 Estimated largest Lyapunov exponents

Time Series	Largest Lyapunov Expo.
Mackey-Glass	0.0071 ± 0.0002
DL Network	0.0073 ± 0.0002
RL Network	0.0059 ± 0.0004
* DL Network	0.0068 ± 0.0003
* RL Network	0.0055 ± 0.0004

Obviously the weighted least square solution helps both DL and RL networks to preserve the original dynamics. Based on the above results and comparisons, it can be concluded that SOFM-based modeling networks with the dynamic learning rule and weighted least square estimation is capable of reliably capturing the dynamics of the given

Mackey-Glass signal for the span of 5000 samples. Although the regular learning (RL) network has similar one-step prediction performance to that of the dynamic learning (DL) network, the measurements of the dynamical invariants demonstrate that the dynamics carried by the given signal is not reliably preserved by the RL network. Later in this chapter, this tendency is also verified by the experiments with real-world signal.

6.1.2 Lorenz Time Series

The previous experiment demonstrates that the SOFM-based modeling network is capable of reliably preserving the signal dynamics in the small positive Lyapunov exponent case. When the signal dynamics possess a large positive Lyapunov exponent, the underlying dynamical system is more sensitive to the initial conditions which poses a more harsh challenge to the modeling task. In this section, the SOFM-based modeling scenario is tested using a chaotic signal with large positive Lyapunov exponents.

The Lorenz signal with the same system configuration ($\sigma = 10$, $r = 28$, $b = 8/3$) as considered in Chapter 3 is taken as the testing signal. The time series is obtained by sampling the signal at 10 Hz. The signal and its spectrum are as shown in Fig. 6.14. The estimated correlation integral map (CIM) curves and their slope are as shown in Fig. 6.15, where the construction dimension is incremented from 4 to 16 with step of 2.

Using $\tau = 1$, the estimate of the correlation dimension is 2.07 ± 0.02. Since the CIM curve saturates when the reconstruction dimension d is over 4, $d = 4$ is selected as the model order. A SOFM network of dimension 22x22 is trained over 3000 samples with both the regular learning and dynamic learning rules. The time-dependent parameters for the learning rate and evolution of the neighborhood function are chosen as $a_\eta = 1$, $b_\eta = 1/500$, $c_\sigma = 1/8$ and $d_\sigma = 1/4000$. In addition, $\mu = 2$ is chosen for the slope of dynamic range in the dynamic learning process. The training proceeds for 150 epochs. The learning curve in terms of the averaged deviation is as shown in Fig. 6.16, where only the portion of 11 to 150 epochs are displayed to reveal the fine detail of the subtle varying trend

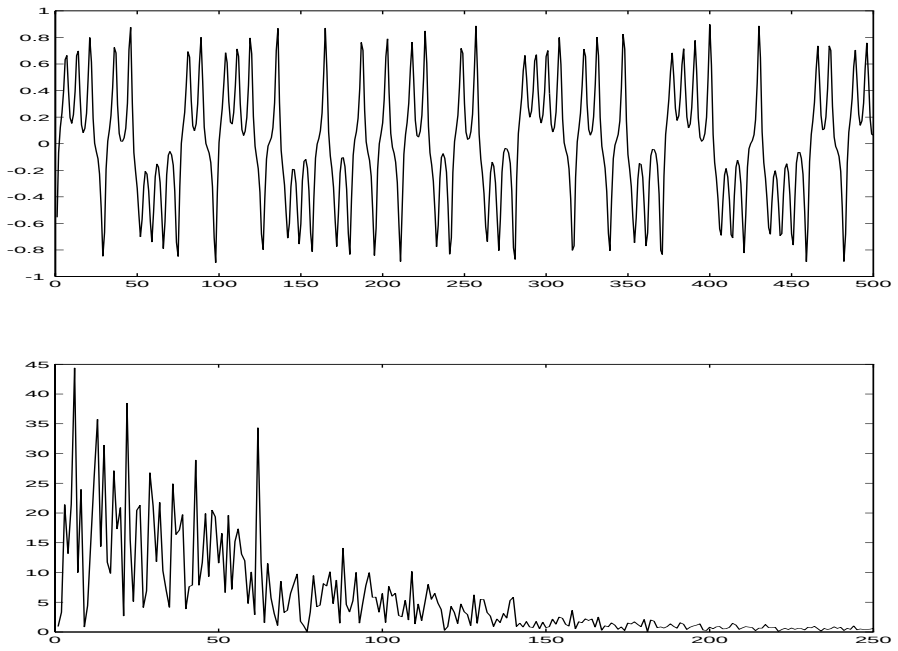


Fig. 6.14 The Lorenz Signal: Waveform and Spectrum

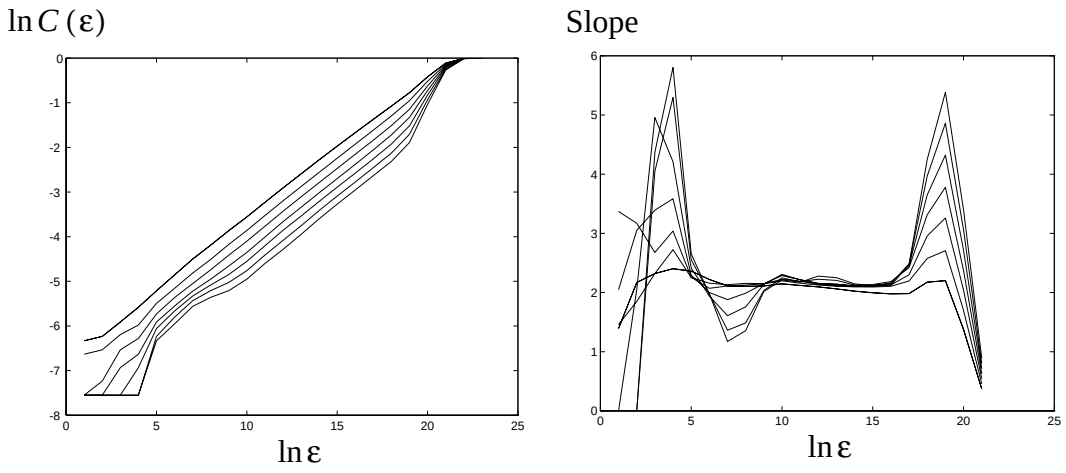


Fig. 6.15 CIM curves and the slope estimation for Lorenz signal

of the dynamic learning. As in the case of Mackey-Glass signal, the learning curve is not monotonically decreasing. The final averaged deviations are $1.123e^{-3}$ and $1.11e^{-3}$ for the regular learning and dynamic learning respectively. From Fig. 6.16, it can be noted that the regular learning process has not yet reached the minimum value. However, the experiment shows that the modeling performance is not improved at all when the training process is continued for the regular learning curve to reach the same value as that of the dynamic learning.

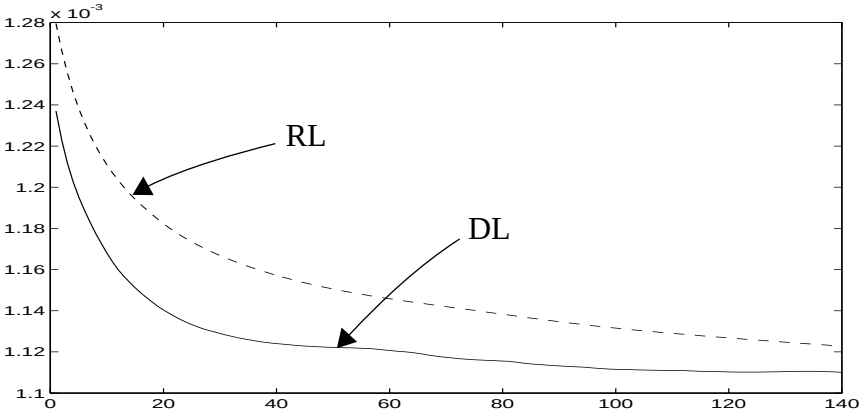


Fig. 6.16 The learning curve

After the training, the local linear models are constructed from the two converged neural fields using the weighted least square solution with $m = 4$. The obtained modeling networks are then iterated as autonomous systems, each of which produces 5000 sample sequence are thus reproduced by the DL and RL networks respectively. The waveforms and the corresponding spectrum are as shown in Fig. 6.17 and Fig. 6.18.

Comparing with Fig. 6.14, certain regularity can be visualized in the waveform of the RL network output. The deviation of its spectrum from the original signal can also be noted. The DL network output, however, is quite close to the original signal in terms of both waveform and spectrum as shown in Fig. 6.17.

The superiority of the DL network over the RL network can also be noted with respect to the dynamical invariants. Using the algorithm by Wolf et al., the Largest Lyapunov exponents of the SOFM-based network outputs and the original Lorenz signal are estimated as shown in Fig. 6.19. The computed correlational integral maps and the corresponding slopes are shown in Fig. 6.20. To compare the performances, the computed results of theabove dynamical invariants are listed in Table 6.4.

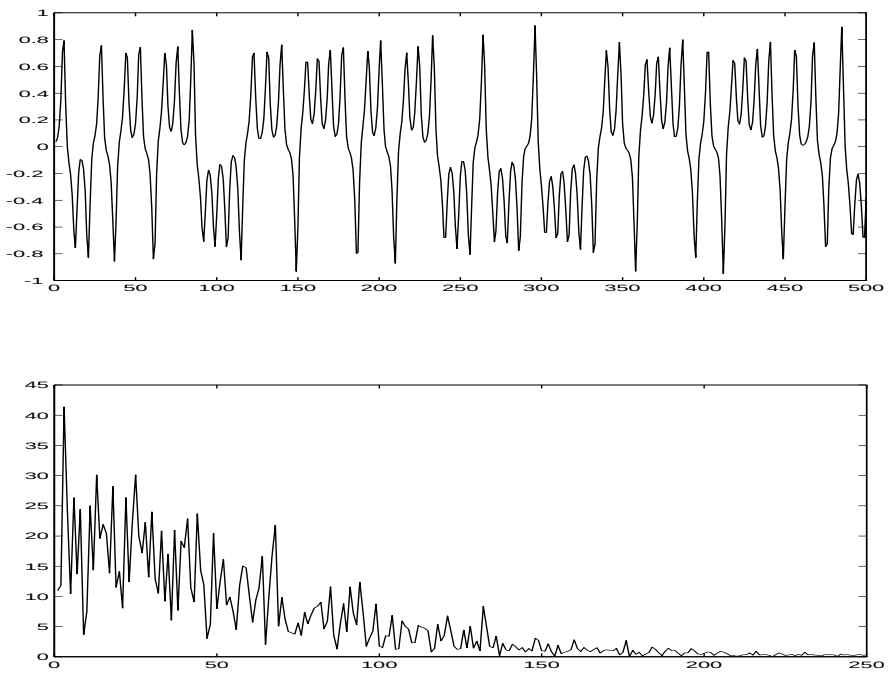


Fig. 6.17 Autonomous output of the DL network: Waveform and Spectrum

Table 6.4 Estimated dynamical invariants

Time Series	Largest Lyapunov Expo.	Correlation Dim.
Lorenz System	2.17 ± 0.03	2.07 ± 0.02
DL Network	2.09 ± 0.02	2.08 ± 0.02
RL Network	1.83 ± 0.03	2.01 ± 0.03

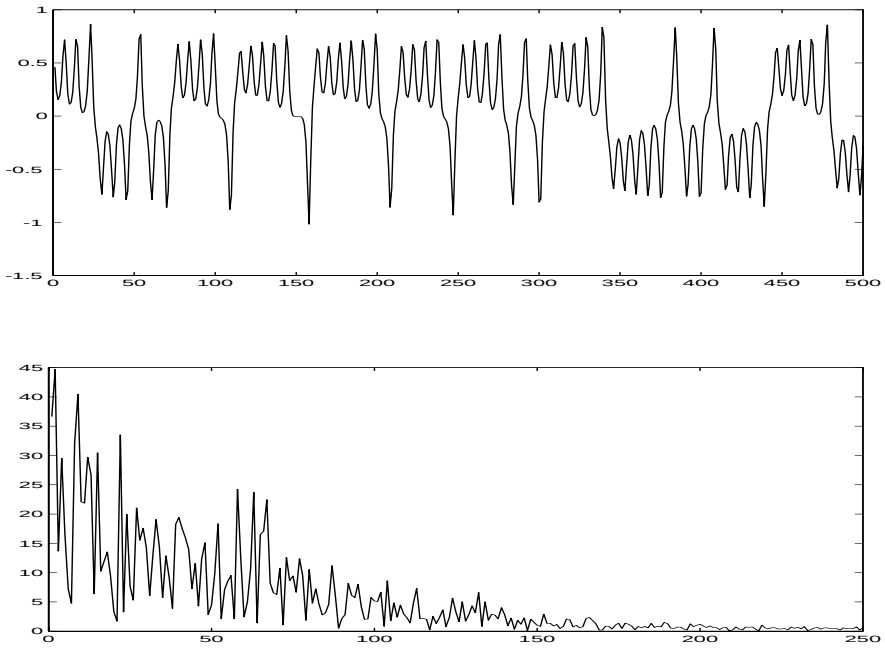


Fig. 6.18 Autonomous output of the RL network: Waveform and Spectrum

From the estimated values of dynamical invariants, it can be seen that DL network has preserved the dynamics of the Lorenz signal to a certain extent. In comparison, the motion captured by the RL network demonstrates certain regularity. Therefore it can be concluded that DL network is a more powerful modeling structure than the RL network.

The advantage of DL network can actually be attributed to the modified SOFM learning algorithm. The impact of dynamic learning rule over the neural field can be visualized in the following example. The waveform in Fig. 6.21 is a one-step prediction output of the RL SOFM-based network constructed from the 3000 sample Lorenz time series as discussed above. The segment marked within the broken lines is used to predict the sample marked with the cross where larger prediction error occurs. Fig. 6.21 (b) and (c) are the contour plots depicting the similarity of all neurons to the reference neuron marked

by crosses. The inner contour line encircles a set of neurons with an activation pattern that differs less than 0.1. The response patterns of the remaining neurons are sliced into 8 activation levels with the descending step size of 0.15. Comparing the two response regions marked by the inner lines, it can be seen that the dynamic learning effectively attracted more neurons toward the current state pattern, and thus enhancing the resolution for the high curvature portion of the underlying dynamics. On the other hand, it can also be noted that the general response pattern has not been fundamentally changed by the dynamic learning. In another word, the dynamic learning rule does not disturb the original topology-preserving tendency but the local resolution structure. In the statistical sense, the resulted feature map not be an optimum neural representation of the input space though it provides a better modeling platform.

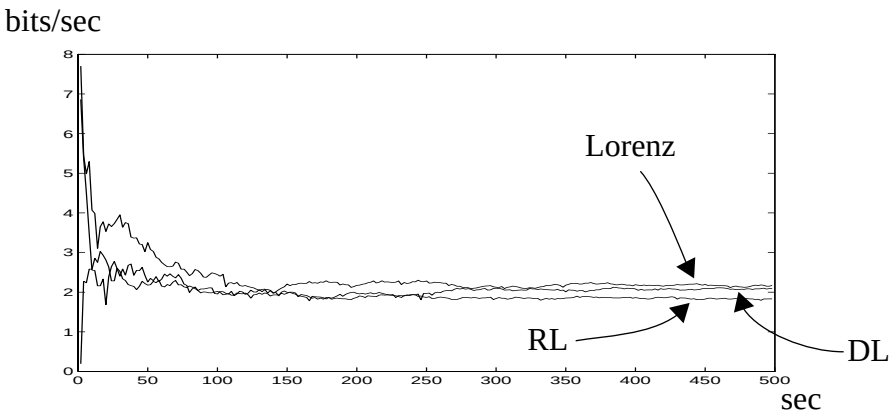
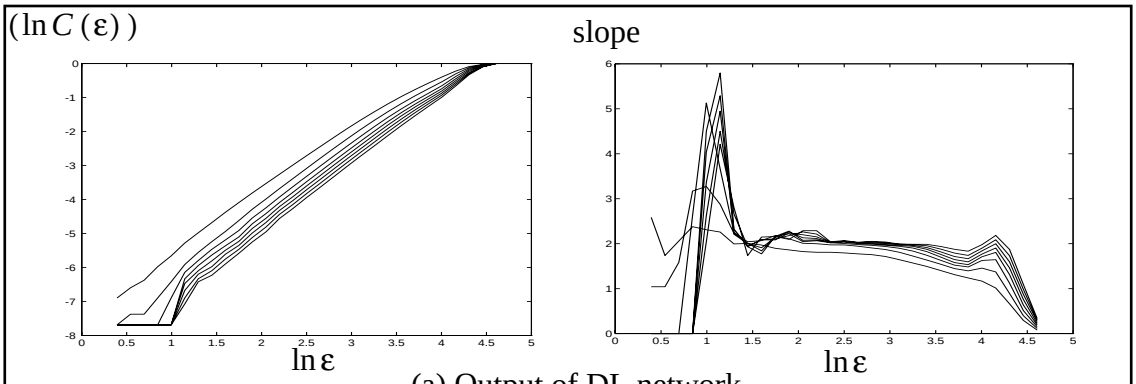
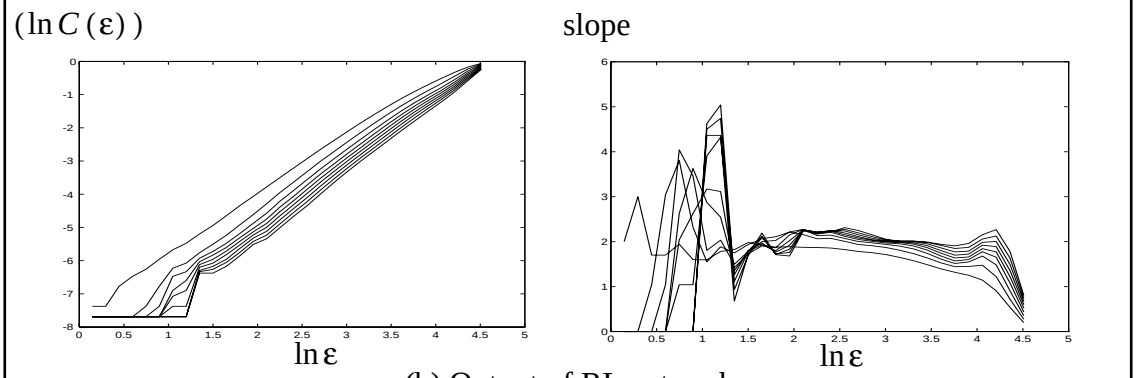


Fig. 6.19 Estimate of Largest Lyapunov Exponent

From the experiment and discussion in this subsection, it can be concluded that the SOFM-based dynamic modeling with dynamic learning is capable of capturing the signal dynamics which has large positive Lyapunov exponents. In addition to the synthetic time series, it will be shown in Section 6.3 that the dynamic learning rule is also applicable to the real-world signals. The advantage of DL network can be perceived in the experimental test with laser time series.



(a) Output of DL network



(b) Output of RL network

Fig. 6.20 CIM curves and the slope estimation for network outputs

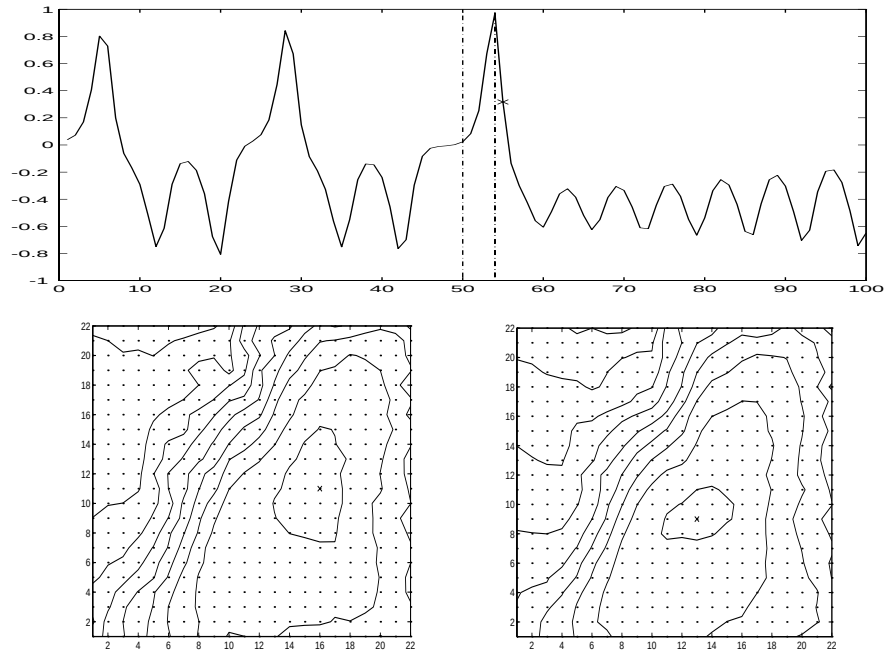


Fig. 6.21 1-step prediction and similarity contour plots

6.1.3 Approximation of Seamless Patching of Local Models

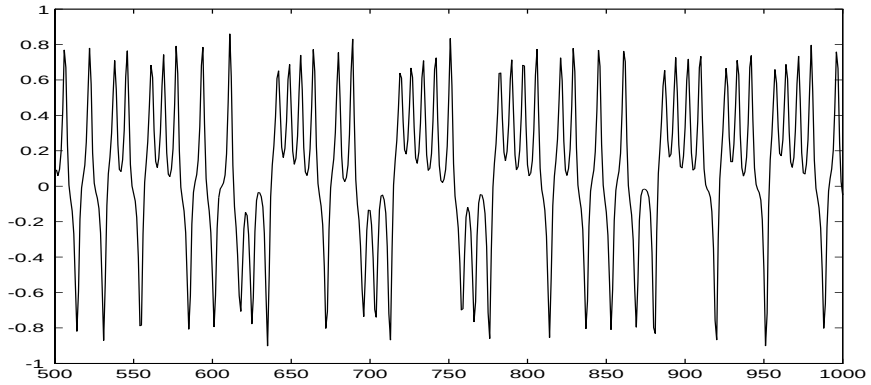
Using a finite set of local linear models to approximate a global functional map of a complicated dynamics have been investigated before. Crutchfield and McNamara [23] suggested that the piecewise linear models are unreliable indicators of the underlying deterministic behavior. Based on their investigation, it is claimed that piecewise linear equations of motion can exhibit periodic behavior when the original dynamics are chaotic and vice versa. However, from the experimental results in the last two sections, it is seen that these observation are not applicable to the SOFM-based modeling system.

Another big concern is the lack of smooth continuation between two local models. Such discontinuity can be reflected by a discontinuous approximation $\tilde{F}$ of the continuous motion of the underlying dynamics. Although they may provide a nice short-term predictive performance, such discontinuity can give rise to some undesirable behavior when the constructed $\tilde{F}$ in form of a finite set of local linear models are iterated as an autonomous system. Such discontinuity may lead the autonomous motion to subsections where $\tilde{F}$ is not defined during the fitting process. Therefore it is hard to predict such undesirable behavior. We can experimentally find such discontinuity, if they exist, by iterating the defective model. This is illustrated in the next example.

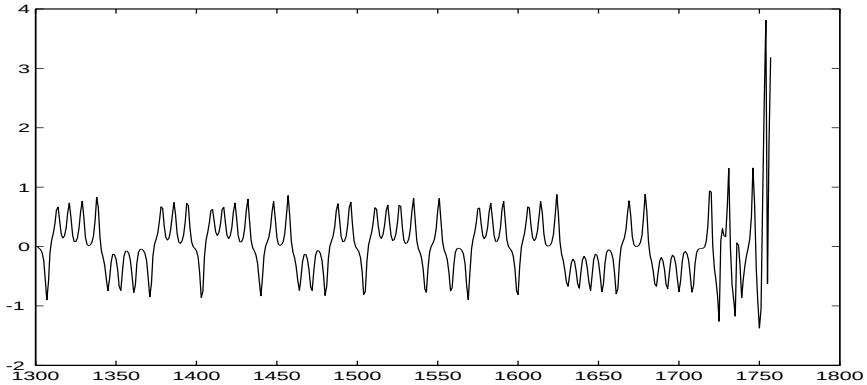
The x-component of Lorenz system as used in the last section is a well-known chaotic time series. The experimental signal of 4000 samples is generated by sampling this x-component signal at 10 Hz. By normalization the signal is further scaled to the unit amplitude. The portion of the state space is partitioned with a vector quantization procedure using a simple competitive network. The model dimension is 4 and 484 competitive neurons are used. The competitive network is then trained with the Kohonen learning rule as in last subsection, but with 0 radius neighborhood function. After the training, the input portion of the state space is thus partitioned into Voronoi polygons centered at each neurons. Each local linear model is then constructed using 14 nearest neighbors from the

input history. The solution is obtained by fitting the local linear equation these nearest neighbors in the least square sense. The approximation to the unknown global model is thus formed by piecing together all these local models.

The constructed modeling network is then started with an input as an autonomous system, and Fig. 6.22 shows two segments of iterative outputs ranging from iteration 501 to 1000 and 1301 to around 1750. The dynamic motion of the first segment can be noted to be close to the original Lorenz's system. However, the unpredictable collapse in the second segment reveals the instability of the collection of the local linear models. Moreover the experiment also shows that it has a 1-step prediction performance comparable with the SOFM-based modeling. Thus large discontinuity at the boundary between local models is a natural reason to account for such phenomenon.



(a) Iteration 501 to 1000



(b) Iteration 1501 to 2000

Fig. 6.22 The iterative output of localized models

The signal in a practical application is always of finite length. Though the overall statistical distribution holds in the global sense, in each local area the data samples may be spaced irregularly. Such local irregular distribution can lead to biased local linear models, and thus large discontinuity can be expected. This phenomenon is even more probable in the case of recursive local linear predictive modeling. In the following example, a 600 sample Lorenz time series is used for the recursive prediction using the local linear approximation method. The local linear model is fitted to the 18 nearest neighbors with the dimension of 5. A 500 sample segment is as shown in Fig. 6.23. The unpredictable amplitude-jumping is clearly visible. Similar phenomenon persists even when different number of nearest neighbors are used for the estimation of the local models.

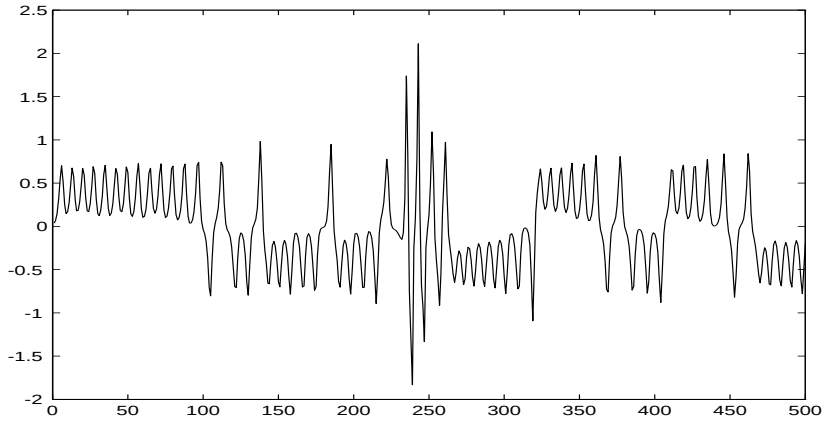


Fig. 6.23 State-dependent Autonomous Prediction

Comparing with the above methods, SOFM-based modeling is not hindered by such discontinuity. In the SOFM-based scenario, the local models are estimated from the neural field. Due to the use of neighborhood function in SOFM, a strong statistical constraint is gradually imposed over the converging neural field during the training process. Therefore the irregular spacing of the data samples can be smoothed out, which alleviates the discontinuity problem to a satisfactory extent, if not eliminated.

Although it is not known how close we can approach toward seamless patching of local linear models, the SOFM-based modeling represents a feasible realization of faithful local linear dynamic models.

6.2 Consistency of the Constructed Models

In section 6.1, the SOFM-based dynamic modeling has been simulated with two chaotic time series. Based on the measurement of Lyapunov exponents and correlation dimension it has been shown that the underlying dynamics have been faithfully preserved by the SOFM-based local linear models. This demonstrates that SOFM with a modified learning rule is a reliable infrastructure for dynamic modeling. On the other hand, SOFM is known for the tendency that the final state depends on the initial states. For all the application of SOFM, the training process is always started with the network randomly initialized. Thus it is natural to ask if the constructed local models are consistent with respect to different initialization. Moreover, the method of local linear modeling has been suggested to be an unreliable indicator of the underlying deterministic behavior. One concern is the long-term consistency of the generated time series. In this section, the study proceeds to investigate the property of this approach with respect to the consistency of SOFM, reliability and temporal stability.

6.2.1 Consistency vs. Different Initial SOFM States

For a fair training process, the weight vectors are usually started with random values. On the other hand, it is known that the final state of converged SOFM has the tendency of depending on the initial condition of the weight vectors. Numerous examples have been reported regarding the converged neural field which is not unique if the training is started from different initial states. In this section, the SOFM with dynamic learning is used to evaluate the impact of such uncertain final state upon the consistency of the constructed local models.

In this experiment, the modeling process in subsection 6.2.1 is applied to four SOFM's with the same configuration using the Lorenz signal. However the four SOFM's are initialize with different set of random weight vectors. Therefore the training process of the four SOFM will end up with the different neural field structure.

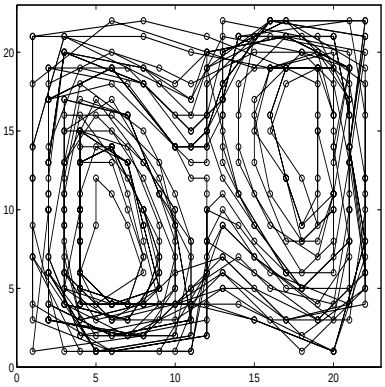
After the training process, the local linear models are constructed from the four converged SOFM's, and four SOFM-based dynamic modeling networks are thus formed. The test is then performed by iterating these networks as autonomous systems started with the same input, and four 5000 sample sequences are obtained as the autonomous prediction outputs. The trajectories of the winner-takes-all are shown in Fig. 6.24. The feature of topology-preserving is manifest by the trajectory structure. In addition, the similar structures of these trajectories demonstrate that the SOFM's have converged to similar neural fields. Obviously different initial SOFM states have led to different orientations in the converged neural fields.

Table 6.5 Dynamical invariants

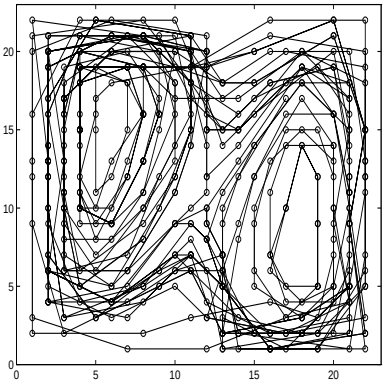
Network	Cor. Dim.	Lyap. Expo.
1	2.08 ± 0.02	2.09 ± 0.02
2	2.04 ± 0.02	2.11 ± 0.02
3	2.06 ± 0.02	2.09 ± 0.02
4	2.05 ± 0.02	2.12 ± 0.02

Since these four iterative output sequences are not distinguishable from the one obtained in Subsection 6.1.2, the further evaluation has to be based on the dynamical invariants. The largest Lyapunov exponents and correlational dimensions of these four sequences computed with the results are illustrated in Table 6.5. According to Table 6.5, it can be concluded that the four modeling networks have captured the underlying dynamics

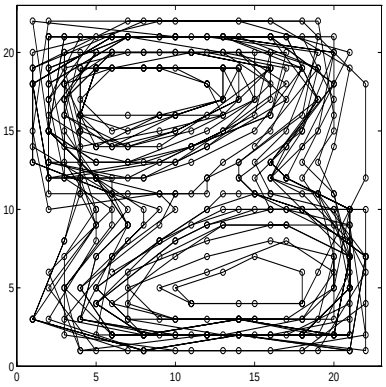
to approximately the same extent. In other words, the local linear models constructed from the four converged neural fields are consistent with each other in the long-term sense, and independent of the initial states of the SOFM weight vectors.



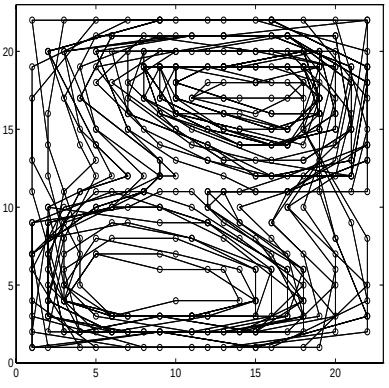
(a) Network 1



(b) Network 2



(c) Network 3

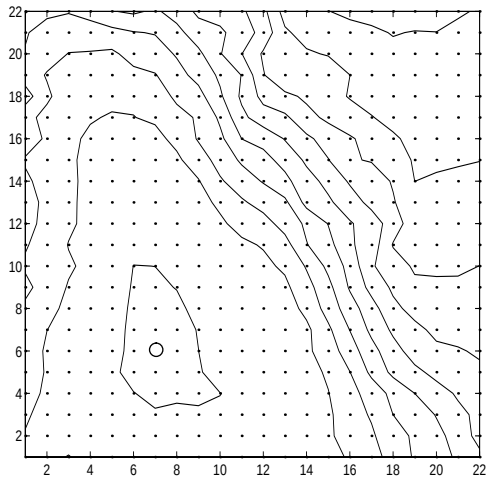


(d) Network 4

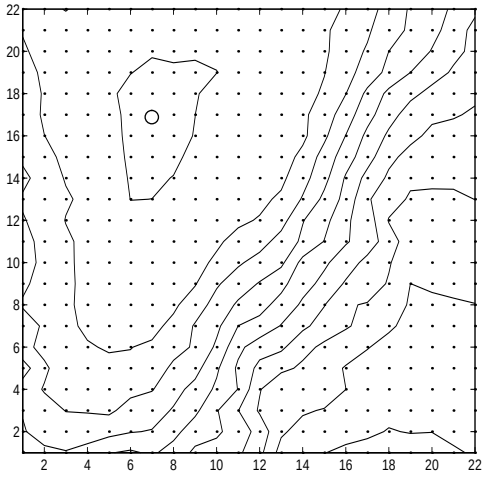
Fig. 6.24 Trajectories of winner-takes-all neurons

Actually the consistency of the local models is a direct result of consistency of the neural field. This can be partially perceived by the clustered dynamic activity distributions of the neural fields. In Fig. 6.25, contour plots are drawn based on the four converged neural fields. They depict the similarity of all neurons to the reference neuron in each neural field. The reference neuron is identified by the same input. The neurons encircled by the inner contour lines represent a pattern of activation with deviation less than 0.1 from the reference neuron. By comparison, it can be seen that the four networks have approximately

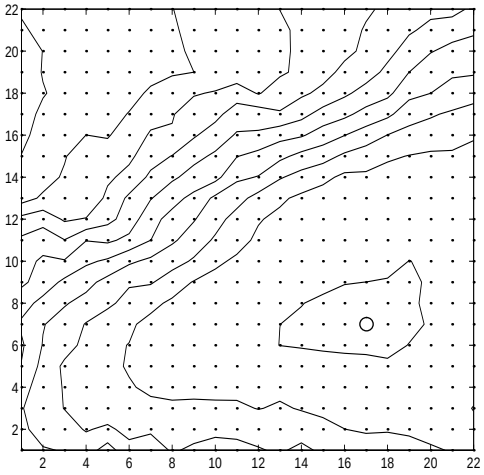
identical neural activity patterns. The same observation can be made with the reference neurons located anywhere in the output lattice. This demonstrates that the clustered areas have the same activation patterns and bear the same resolution. In terms of metric topology structure, the final state of the SOFM is unique, which is determined by the topology of the input space.



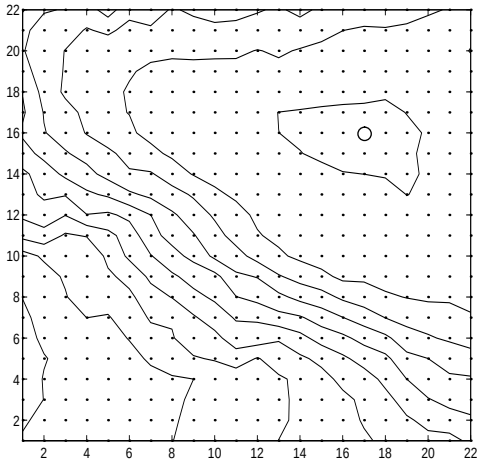
(a) Network 1



(b) Network 2



(c) Network 3



(d) Network 4

Fig. 6.25 Similarity contour plots

Based on the above observation, it can be concluded that different initial weight states may just lead the SOFM toward final states with different orientations. The converged neural fields have the same neural activity patterns which are independent of the initial weight vectors. And the constructed local models as a whole for the global representation are consistent. Of course, all these occur under the condition that appropriate system configuration is specified for the SOFM, including the adaptive schedule of the training process.

6.2.2 Temporal Consistency

In section 6.1, it has been shown that the SOFM-based local modeling approach is capable of capturing the complicated dynamic structure of chaotic signals. In this section, the experiment is extended to the assessment of the long-term dynamic stability. If the SOFM-based local modeling method is hindered by the discontinuity problem, it must be reflected by the long-term performance. The test is still based on evaluation of the dynamical invariants as in section 6.1. The assessment is based on the comparison of two widely separated segments taken from a single long autonomous iteration.

For the local linear modeling systems fitted to Mackey-Glass and Lorenz systems in section 6.1, the autonomous iterations are continued for 15000 samples. The last 500 sample segments of these iterative output sequences by two modeling networks are as shown in Fig. 6.26 and Fig. 6.27.

In comparison with the corresponding waveforms and spectra in Fig. 6.3 and 6.13, it can be seen that the iterative outputs looks very close to the original signal both in time domain and frequency domain after a long autonomous iteration.

The dynamical invariants are also computed for these segments of iterative outputs, and the results, in comparison with the quantities obtained in Section 6.1, are listed in Table 6.6.

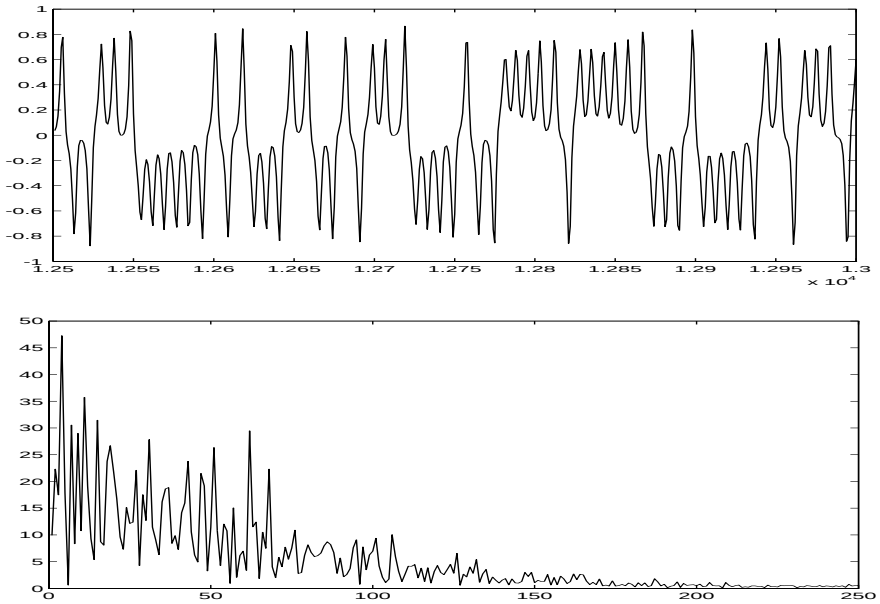


Fig. 6.26 Output of the DL Network: Waveform and Spectrum

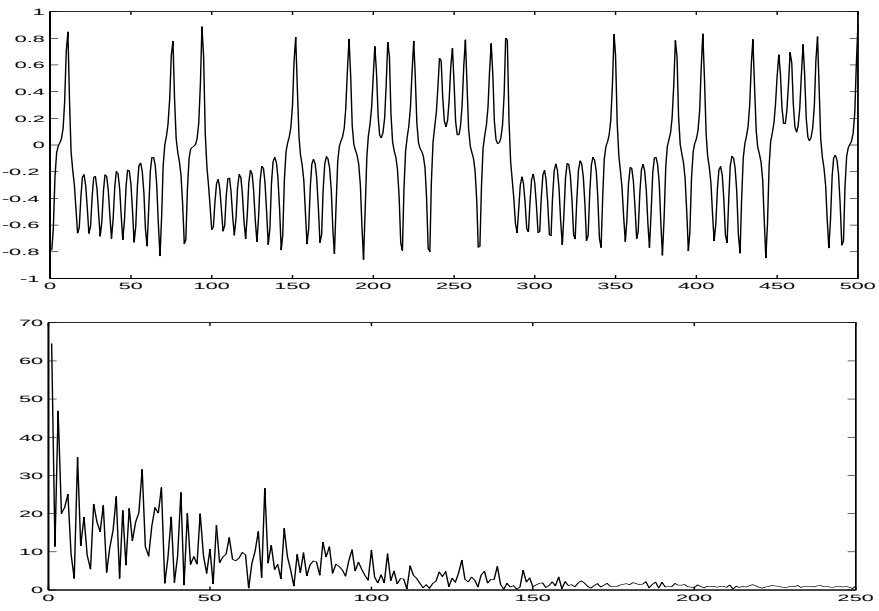


Fig. 6.27 Output of the RL Network: Waveform and Spectrum

From these dynamical invariants obtained from widely separated segments, it can be noted that all these dynamical invariants are both consistent with the original signal, and consistent over time as well, though some subtle deviation exist for the case of RL networks. It can be concluded that the constructed modeling networks are stable as autonomous systems with the underlying dynamics preserved.

Table 6.6. Dynamical invariants

modeling of Lorenz System	Correlation Dimension		Largest Lyapunov Exponent	
	First segment	Last Segment	First Segment	Last Segment
DL Network	2.08 ± 0.02	2.05 ± 0.02	2.09 ± 0.03	2.11 ± 0.03
RL network	2.01 ± 0.02	2.02 ± 0.02	1.83 ± 0.04	1.84 ± 0.04

On the other hand, these results also further demonstrate that the constructed modeling networks are not hindered by the boundary discontinuity, and they can be taken as continuous approximation of the underlying dynamics in the global sense.

6.3 Modeling Real-world Signals

In the previous two sections, the proposed modeling network is considered in the context of numerically integrated signals from the well-known chaotic dynamical systems. In this section, the experiment is extended to two real-world signal processes: Laser time series and electroencephalogram and sun spot time series. It will be shown that the proposed modeling network is a robust modeling scenario which can be easily applied to signals with different nonlinear characteristics.

6.3.1 Laser Time Series

The laser data set consists of 5000 samples. A segment of 500 samples is shown in Fig. 6.28. It is composed of high-frequency oscillations “cycles”, the sudden “collapses”, i.e., large decreases in the amplitude of the cycles, and the packet of growing oscillations between two collapses which is called “an event”. Thus two complete events are present and 3 collapses can observed in Fig. 6.28. One challenge for the modeling task is the repeated collapses with no periodicity.

A 28x28 SOFM is used with the training schedule as $a_\eta = 1$, $b_\eta = 1/500$, $c_\sigma = 1/8$ and $d_\sigma = 1/4000$. The embedding dimension of the signal is 5. In addition $\mu = 2$ is chosen for the slope of dynamic range. The training is performed over the first 4000 samples, and proceeds for 120 epochs with the final averaged deviation is 0.003. Using the weighted least square approach, the local linear models are constructed directly from the converged neural field.

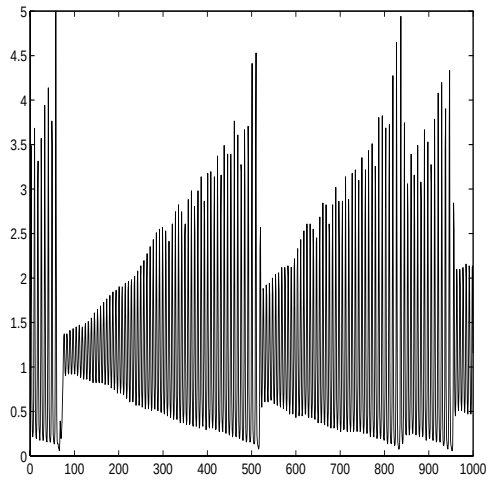


Fig. 6.28 Laser time series

The resulted modeling networks is then iterated as an autonomous system with an input from the remaining segment of 1000 samples. A 2000 sample segment of the iterative

output is shown in Fig. 6.29. Obviously the characteristic collapses and bursting behavior have been successfully captured and reproduced by the constructed modeling network. Similar observation can also be made from their spectra in Fig. 6.30. Without dynamic learning, the “collapse” feature can not be modeled successful. This is demonstrated in the following example.

An RL modeling network is also constructed with the same training schedule but without dynamic learning involved. The first segment of 2000 sample segment of the iterative output is shown in Fig. 6.31. Though the “collapses” has been preserved to certain extent, they do not occur with original depth. In addition, some peaks prior to the “collapse” is too large, which make the sequence to explode after about 3710 iterations. Thus the weak modeling capability of the occurrence of the “collapses” demonstrates that the training of SOFM should be enhanced.

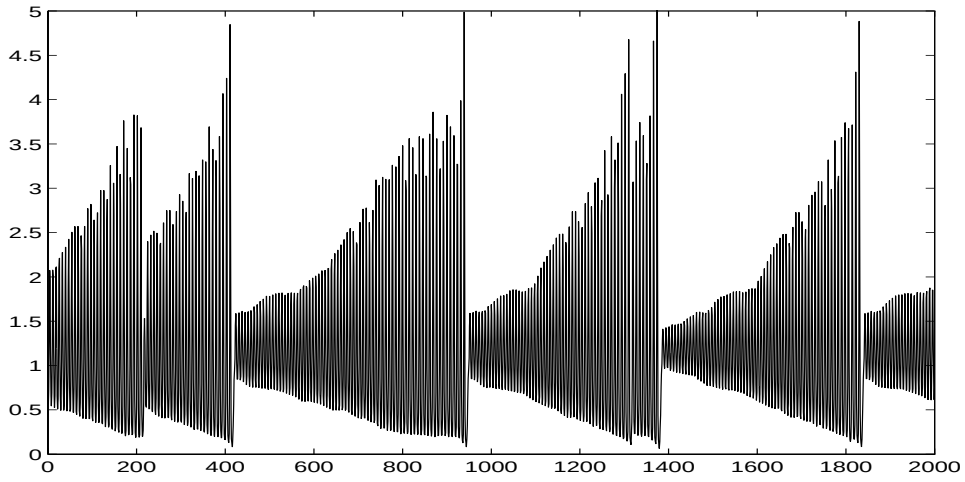
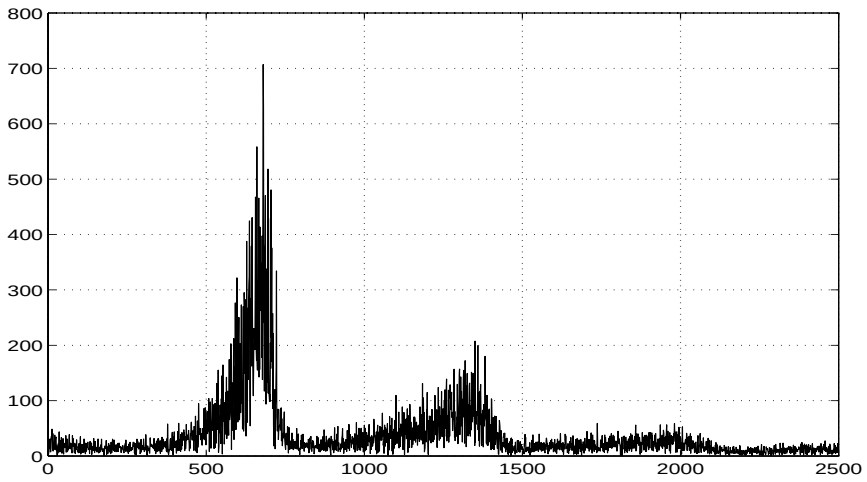
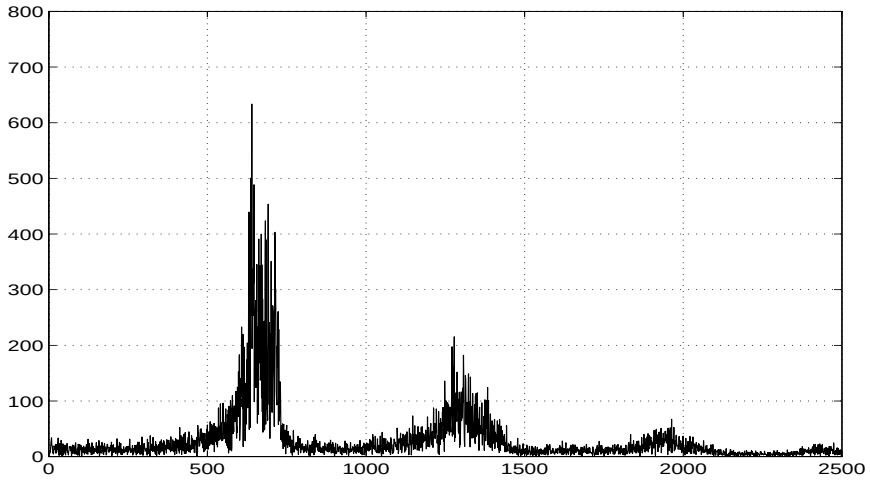


Fig. 6.29 Autonomous prediction of laser time series with DL



(a) Laser



(b) Autonomous

Fig. 6.30 Spectra

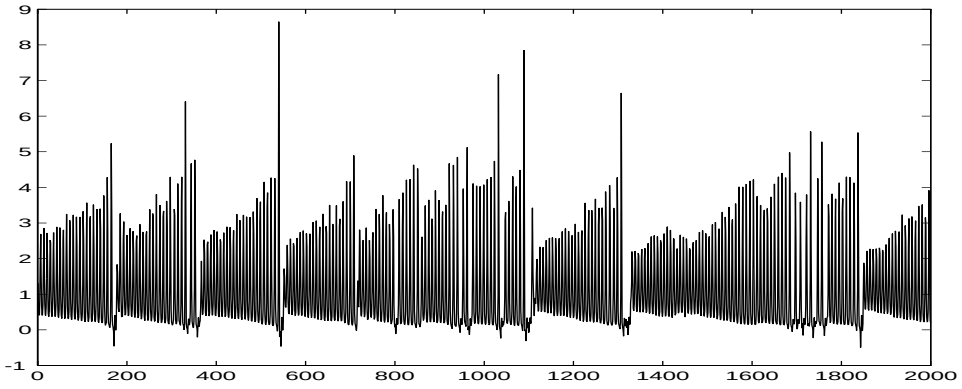


Fig. 6.31 Autonomous prediction of laser time series with RL

6.3.2 EEG Signal

The EEG signal is composed of 5200 samples. A segment of 300 samples is shown in Fig. 6.32. It is basically characterized by irregularly spaced peaks. To model this signal, a SOFM with output lattice 25x25 is used. The signal is embedded in a 7-dimensional state space. The training schedule is set as $a_\eta = 1$, $b_\eta = 1/500$, $c_\sigma = 1/8$ and $d_\sigma = 1/3000$, which is performed over the 5000 samples for 120 epochs. The final averaged error is 0.014. The local linear models are constructed using the weighted least square procedure. The obtained network is then iterated as an autonomous system. A 300 sample segment of the reconstructed output is shown in Fig. 6.33.

Comparing Fig. 6.32 and 6.33, it can be seen that the characteristic structure of irregularly spaced peaks in original EEG signal has been reflected in the reproduced series. That is, the obtained local linear models exhibits dynamics much more complicated than periodic motion. The SOFM-based modeling is a faithful indicator of nonlinear dynamics in this EEG signal case. On the other hand, the reproduced series contains higher frequency components, which can be noted by comparing their spectra as shown in Fig. 6.34. This may be due to the larger discontinuity caused by the noise present in the EEG signal.

6.3.3 Sunspot Time Series

The sunspot series is composed of 12X232 data samples, which are daily relative sunspot numbers based upon counts of spots and group entities of spots on the sun's surface each day. Here the original series is normalized to unit magnitude. One column segment is as shown in Fig. 6.35 (a). Its major component is quasi-periodic coupled with the upper-end envelope oscillating irregularly. The network is chosen to have a 24x24 node output lattice, and the model dimension is 5. The training schedule is set as $a_\eta = 1$, $b_\eta = 1/500$, $c_\sigma = 1/8$ and $d_\sigma = 1/3500$, which is performed over the first 200 samples of each row for 90 epochs. The final averaged deviation is 0.002.

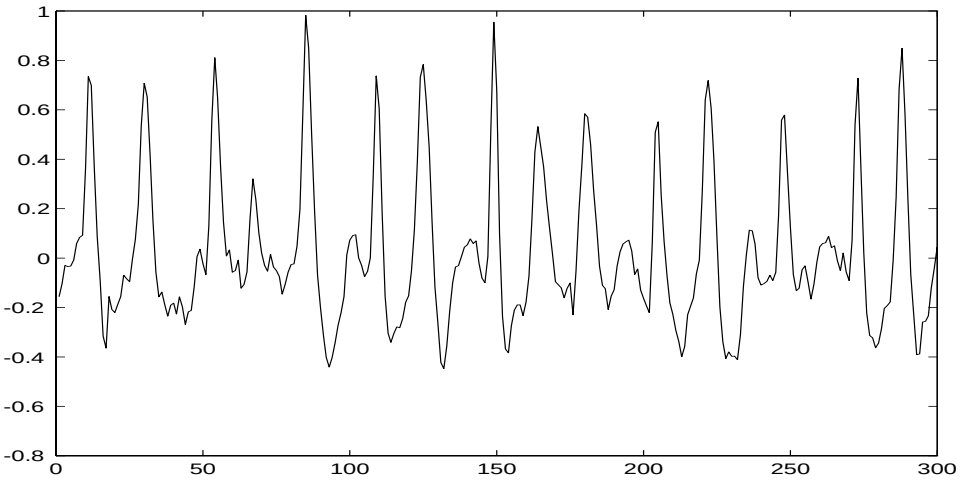


Fig. 6.32 EEG Signal

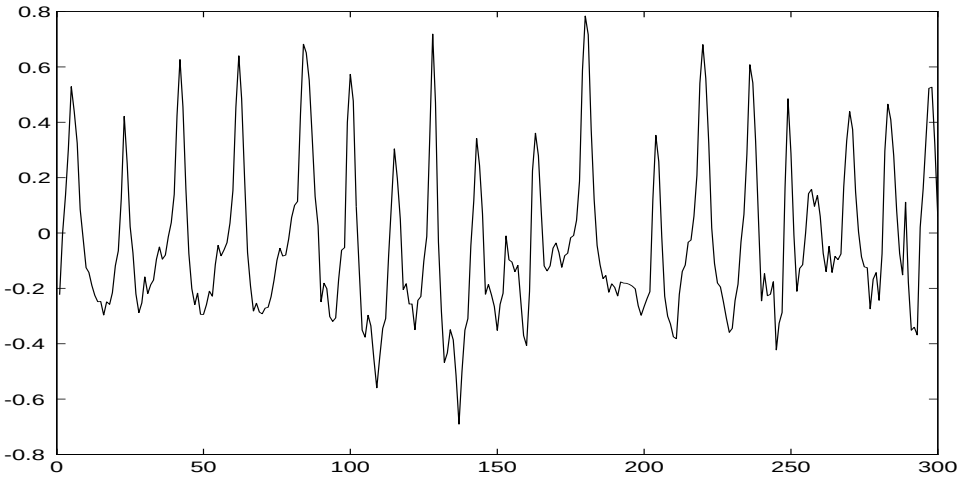
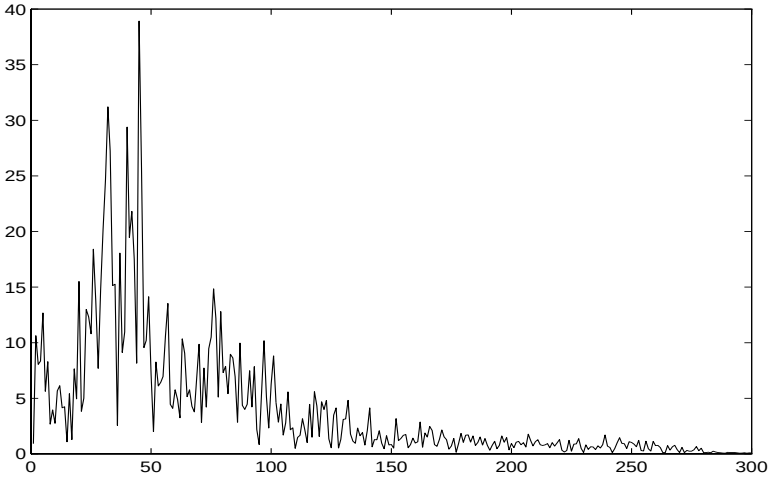
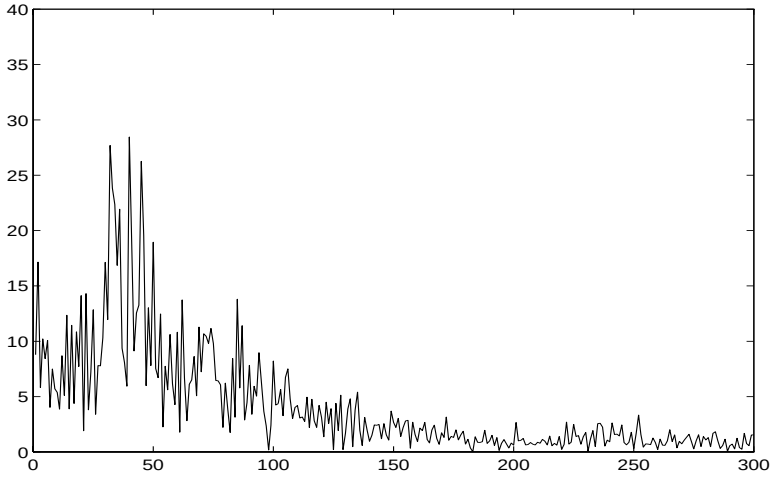


Fig. 6.33 Reproduction

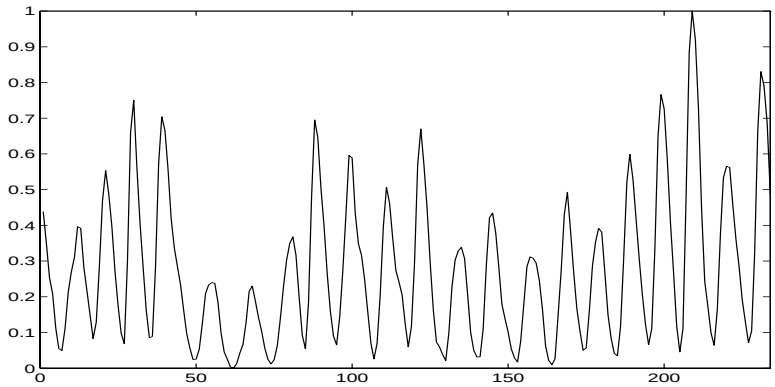


(a) EEG

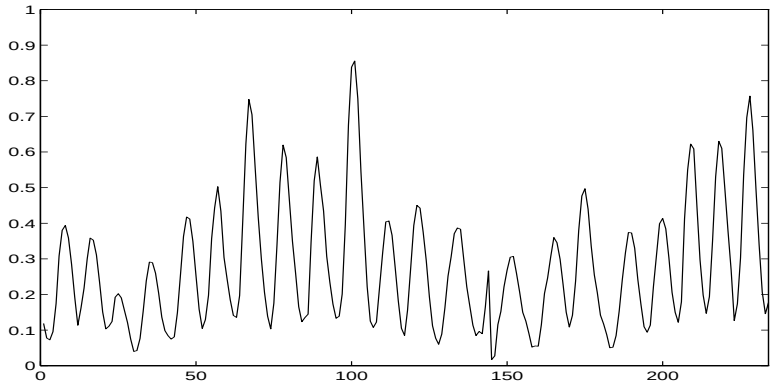


(b) Autonomous

Fig. 6.34 Spectra of EEG and autonomous output



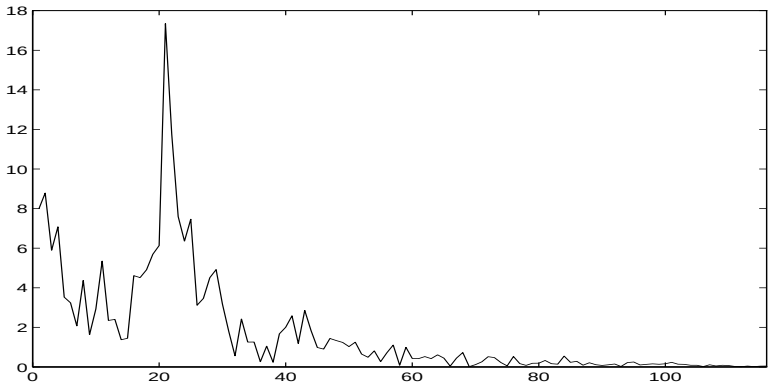
(a) Sunspot time series



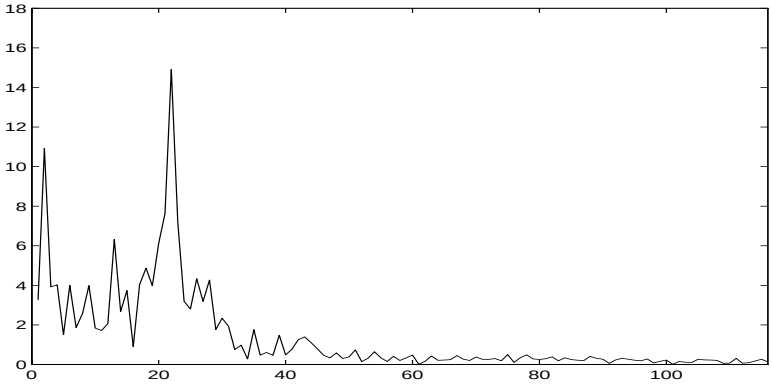
(b) Iterative Output

Fig. 6.35 Waveforms

After the training, the local linear models are constructed from the converged neural field using the weighted least square procedure. The modeling network is then started as an autonomous system. A segment of the iterative output is as shown in Fig. 6.34 (b), where it can be noted that, in addition to the quasi-periodicity, the envelope also exhibits similar feature to the original signal. by the obtained local linear models. These aspects are also reflected in the spectra of the original series and iterated output as shown in Fig. 6.36, where DC components have been removed to reveal the fine spectral structure.



(a) Sunspot



(b) Iterative Output

Fig. 6.36 Spectra

On the other hand, the generated time series seems simpler than the original, in particular at the low frequency end. This is due to the insufficient training samples. Although the original data is composed of 12x232 samples, there are strong correlations between different rows. In this situation, it is very hard for the local linear models to provide fine description of the underlying dynamics. What is significant is that the iterated is still close to the original quasi-periodic signal avoiding the periodic behavior.

Considering the two previous examples presented in this section, it can be seen that the SOFM-based modeling scheme is not hindered by the discontinuity problem in the

context of modeling noisy real-world signals. In the case of laser time series, the underlying dynamic has been preserved to a satisfactory extent. In EEG modeling, the constructed local linear models exhibit a dynamic motion which carries the characteristic of the original signal, although noise present brings up high-frequency components in the iterative output. With sufficient training data, however, the improvement can be made. Therefore it can be concluded that the SOFM-based modeling is an effective modeling scheme applicable to the real-world signals.

CHAPTER 7
CONCLUSIONS AND FUTURE RESEARCH

7.1 Conclusions

Signals generated by chaotic systems represent a potentially rich class of signals both for analyzing and characterizing physical phenomena. Since classical techniques for signal analysis do not exploit the particular structure of chaotic signals, intensive study of chaotic dynamics have focused on new methods.

The approaches developed for dynamic modeling are generally categorized as global and local models. For global models, it is unlikely to find a standard functional basis set capable of representing the intricate geometry of the underlying system. Local models are attractive since fewer statistical and geometric assumptions are required. Local linear modeling is the simplest implementation of the local modeling method. However, most of the work on local linear approaches only addresses the recursive state-dependent prediction. Dynamic approximation using a finite set of local linear models has not been fully studied.

In this research, a dynamic modeling scheme is established as a feasible and effective implementation of dynamical approximation. The whole architecture is composed of a finite set of local linear models. The dynamic pattern is identified based on the current input. Although the constructed systems was not proved to be a smooth functional map as a whole, the experimental result demonstrates that they are actually reliable indicators of the underlying deterministic behavior, which is in contrast with observations of previous investigations. As expected, this modeling scheme is capable of preserving the global dynamics of chaotic systems. The significance of the proposed method is that the scenario of local linear modeling is extended to the application of chaotic dynamics instead of just

temporal interpolation, which marks the fundamental difference between the work of this research and others.

Self-organizing feature map (SOFM) as an unsupervised neural networks has been employed in many practical applications. In most cases, it is used as vector quantization. No previous attempt has been made to explore the dynamical aspect of the static neural field. In this research, SOFM is considered as a modeling infrastructure for approximation of chaotic dynamics. This scheme has the advantages of simple connectivity and efficient implementation. In addition, it also helps reduce the discontinuity between local linear models.

A signal in any application is always of a finite duration. The data samples are usually irregularly spaced in the local area, though they comply with the overall statistical distribution. This poses a problem for the method of local linear modeling, since each local model is constructed from the corresponding response region excited by the local dynamic activities. It can cause the local linear model to be perturbed from the optimum direction which results in discontinuity between the local models. Using the least square fitting method, the finite set of local linear models are constructed directly from the converged neural field. By utilizing a novel neighborhood function, the SOFM learning rule imposes a strong global statistical constraint among all the neurons, such that the irregular spacing of the local data samples are effectively smoothed out. Therefore the use of SOFM reduces the discontinuity problem caused by the irregular local distribution.

The successful use of SOFM in dynamic modeling also reflects an interesting application area for SOFM in addition to the vector quantization. Unlike most other applications of SOFM where the statistical matching is one of the required features, SOFM is required to have dynamic-oriented structure in the dynamic modeling application. That is, the curvature information of the input should also be coded in the feature map. To achieve this objective, a modified SOFM learning rule is proposed with the prediction error involved in the learning process. With this procedure, the local linear models can be con-

structed more close to the optimal architecture in the global sense. From the system consideration, this procedure is a close-loop learning process. The information preserved by the converged feature map neural field is oriented toward the goal of global dynamical approximation. Therefore this also opens an area to explore more applications of the SOFM.

Although the converged feature map is not unique, it is found that the initial state of the weight space only affects the overall orientation of the neural field, but not the representation structure. Thus the obtained modeling network is a consistent functional map.

The determination of local linear models depends on the prototypes of the response regions, and the collective response is composed of dynamic activities of all its neurons equally. Using a least square solution, the constructed local linear models reflects the local dynamics evenly averaged over neurons. This may not be an optimal choice due to the natural deviation from the metric center. To better represent the local dynamics, a weighted least square method is utilized for the local model estimation. Although the weighted least square solution can be derived from the statistical consideration, the weighting matrix is selected based on both heuristics and experimental results. This is due to the fact that the local distribution of the given signal is not available, and its estimation leads to another research subject. The proposed procedure is simple and effective, and it is justified by the experimental results.

So far the SOFM-based modeling network has been tested with both synthetic chaotic signals and real-world signals. In short-term modeling, the multi-step prediction error is consistent with the Casdagli's conjecture up to about 5 steps. This does not mean our network is not capable of dynamic modeling in the long-term sense. Actually the multi-step prediction error criterion still belongs to the category of the sample-by-sample comparison even though it is computed as an average. Thus in the testing experiment, the model validation is based on measurements of the dynamical invariants. The correlation dimension and largest Lyapunov exponents are considered for this purpose, and the result

shows that the proposed method is capable of capturing the chaotic dynamics. This also demonstrates that the smooth dynamics can be approximated by a finite set of local linear functional map.

7.2 Future Research

In this research, the method of local linear modeling has been extended to a new architecture which approximate a smooth dynamics with a finite set of local linear models. In conjunction with the proposed techniques of dynamic learning and weighted least square solution, this architecture is not only a reliable indicator, but also capable of capturing the underlying chaotic dynamics. Its application potential has been justified by experiments with both equation-generated signals and real-world signals as well.

On the other hand, there are some open issues related to the proposed modeling architecture which deserve closer scrutiny and further research.

1. SOFM has been employed as the modeling infrastructure. One of its strong points is tighter statistical constraint over the neural field during the training process. A related aspect of SOFM is the pre-specified size of its output lattice, which plays a critical role in its application. It is reasonable to believe that there is a lower bound for its size. On the other hand, it is also natural to expect that an optimal size exists. Developing a theory to support this is of significance for the applications.

2. Dynamic learning is derived from system considerations, which involves the prediction error in the learning process. Its contribution to the improvement of modeling performance has been demonstrated by the experimental results. However, it is still not clear if the optimal amount of prediction has been utilized to modulate the naturally decreasing learning rate. A systematic analysis is definitely necessary.

3. Weighted least squares solution for the construction of local linear models is proposed based on a heuristic consideration. Actually it bears a statistical ground. The best structure of this procedure heavily depends on the local distribution of the signal. Finding

their relationship is meaningful for both dynamic modeling and application of SOFM.

4. The SOFM learning process is stochastic in nature. To some extent, the successful formation of a feature map depends on the system configuration and the training schedule. The related theory have not been fully developed. At this time, broader systematic experiments are still needed to obtain more useful and reliable guidance for the selection of better system configuration.

In general, clarifying all these open issues will not only help to extend the application of the SOFM-based modeling network to broader class of real-world signals, but also bring deeper understanding of the self-organizing feature map.

REFERENCES

[1] Abarbanel, H.D.I., R. Brown, and J.B. Kadtke, "Prediction in chaotic nonlinear systems: Methods for time series with broadband Fourier spectra," Phys. Rev. A, Vol. 41, pp. 1782-1789, 1990.

[2] Abarbanel, H.D.I., R. Brown, J.J. Sidorowich, and L.S. Tsimring, "The analysis of Observed chaotic data in physical systems," Rev. Mod. Phy., Vol. 65, No. 4, pp. 1331-1335, 1993.

[3] Ahalt, S.C., A.K. Krishnamurthy, P. Chen, and D.E. Melton, "Competitive learning algorithms for vector quantization," Neural networks, Vol. 3, pp. 277-290, 1990.

[4] Amari, S.-L., "Geometrical theory on manifolds of linear systems," Dept. Mathematical Engineering and Instrumentation Physics Technical Reports, University of Tokyo, METR 86-1, March 1966.

[5] Amari, S.-L., "A theory of adaptive pattern classifiers," IEEE Trans. Electronic Computers, EC, Vol. 16, No. 3, pp. 299-307, 1967.

[6] Amari, S.-I., "Learning patterns and pattern sequences by self-organizing nets of threshold elements," IEEE Trans., Vol. 21, pp. 1197-1206, 1972.

[7] Amari, S.-I, "Formation of cortical cognitive map by self-organization," Computational Neuroscience, ed. by Eric L. Schwartz, The MIT Press, pp. 21-28, 1990.

[8] Anderson, J.A., "Simple neural network generating interactive memory," Math. Biosci., Vol. 14, pp. 194-220, 1972.

[9] Baras, J.S., and A. LaVigna, "Convergence of Kohonen's learning vector quantization," International Joint Conference on Neural Networks, Vol. 3, pp. 17-20, San Diego, CA, 1990.

[10] Barlow, H.B., "Unsupervised learning," Neural Computation, Vol. 1, pp. 295-311, 1989.

[11] Barlow, H., and P. Foldiak, "Adaptation and decorrelation in the Cortex," In The Computing Neuron, ed. by R. Durbin, C. Miall, and G. Mitchison, pp. 54-72, MA: Addison-Wesley, 1989.

[12] Bauer, H.-U., R. Der, M. Herrmann, "Controlling the magnification factor of self-organizing feature maps," Neural Computation, Vol. 8, pp. 757-771, 1996.

[13] Bauer, H.-U., and K.R. Pawelzik, “Quantifying the neighborhood preservation of self-organizing feature maps,” IEEE Transactions on Neural Networks, Vol. 3, No. 4, , pp. 131-139, 1992.

[14] Brown, R., “Orthonormal polynomials as prediction functions in arbitrary phase space dimensions,” University of California, San Diego, INLS, 1992.

[15] Bryant, P., R. Brown, and H.D.I. Abarbanel, “Lyapunov exponents from observed time series,” Phys. Rev. Lett., Vol. 65, pp. 1523-1526, 1990.

[16] Carpenter, G.A., and S. Grossberg, “A massively parallel architecture for a self-organizing neural pattern recognition machine,” Computer Vision, Graphics and Image Processing, Vol. 37, pp. 54-115, 1987.

[17] Carpenter, G.A., and S. Grossberg, “ART 2: Self-organization of stable category recognition codes for analog input patterns,” Allied Optics, Vol. 26, No. 23, pp. 4919-4930, December 1987.

[18] Carter, S., R.J. Frank, and D.S.W. Tansley, “Clone detection in telecommunications software systems: A neural net approach,” In Applications of Neural Networks to Telecommunications, ed. by J.Alspector, R. Goodman, and T.X. Brown, pp. 273-280, Hillsdale, NJ: Lawrence Erlbaum, 1993.

[19] Casdagli, M., “Nonlinear prediction of chaotic time series,” Phys. D, Vol. 35, pp. 335-339, 1989.

[20] Chua, L.O., and T. Lin, “Chaos in digital filters,” IEEE Transactions on Circuits and Systems, CAS, Vol. 34, pp. 648-658, 1988.

[21] Cottrell, M., and J.C. Fort, “A stochastic model of retinotopy: A self organizing process,” Biological Cybernetics, Vol. 53, pp. 405-411, 1986.

[22] Cowan, J.D., and D.H. Sharp, Q. Rev. Biophysics, Vol. 2, pp. 365, 1988.

[23] Crutchfield, J.P., B.S. McNamara, “Equations of motion from a data series,” Complex Systems, Vol. 1, pp. 417-421, 1987.

[24] DeSieno, D., “Adding a conscience to competitive learning,” IEEE International Conference on Neural Networks, Vol. 1, pp. 117-124, San Diego, CA, 1988.

[25] Ditto, W.L., S.N. Rauseo, and M.L. Spano, “Experimental control of chaos,” Phys. Rev. Lett., Vol. 65, pp. 3211-3216, 1990.

[26] Eckmann, J.P., S. Oliffson Kamphorst, D. Ruelle, and S. Ciliberto, “Lyapunov exponents from time series,” Physical Review A, Vol. 34, No. 6, pp. 4971-4975, 1986.

[27] Eckmann, J.P., and D. Ruelle, “Ergodic theory of chaos and strange attractors,” Reviews of Modern Physics, Vol. 57, No. 3, Part 1, pp. 617-656, 1985.

[28] Farmer, J.D., E. Ott, and J.A. Yorke, “The dimension of chaotic attractors,” Physica D, Vol. 7, pp. 153-158, 1983.

[29] Farmer, J.D., and J.J. Sidorowich, “Predicting chaotic time series,” Phy.Rev. Let., Vol. 59, No. 8, pp. 845-849, 1987.

[30] Ferran, E.A., and P. Ferrara, “Topological maps of protein sequence,” Biological Cybernetics, Vol. 65, pp. 451-458, 1991.

[31] Fukushima, K., “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” Biol. Cyber., Vol. 36, No. 4, pp. 193-202, 1980.

[32] Gallez, D., and A. Babloyantz, “Predictability of human EEG: a dynamical approach,” Biological Cybernetics, Vol. 64, pp. 381-391, 1991.

[33] Giona, M., F. Lentini, and V. Cimagalli, “Functional reconstruction and local prediction of chaotic time series,” Phys. Rev. A, Vol. 44, pp. 3496-3502, 1991.

[34] Grassberger, P., “Generalized dimensions of strange attractors,” Physics Letters A, Vol. 97, No. 6, pp. 227-230, 1983.

[35] Grassberger, P. and I. Procaccia, “Characterization of strange attractors,” Physical Review Letters, Vol. 50, No. 5, pp. 346-349, 1983.

[36] Gray, R.M., “Vector Quantization,” IEEE ASSP Magazine, Vol. 1, pp. 4-29, 1984.

[37] Grossberg, S., “A prediction theory for some nonlinear functional-difference equations,” Journal of Mathematical Analysis and Applications, Vol. 22, pp. 490-522, 1969.

[38] Grossberg, S., “On learning and energy-entropy dependence in recurrent and nonrecurrent signed networks,” Journal of Statistical Physics, Vol. 1, pp. 319-350, 1969.

[39] Guilleman, V., and A. Pollack, Differential Topology, Prentice-Hall, Englewood Cliffs, New Jersey, 1974.

[40] Hammel, S.M., “A noise reduction method for chaotic systems,” Physics Letters A, Vol. 148, pp. 421-426, 1990.

[41] Haykin, S., “Blind equalization formulated as a self-organized learning process,” Proceedings of the Twenty-Sixth Asilomar Conference on Signals, Systems, and Computers, pp. 346-350, Pacific Grove, CA, 1992.

[42] Haykin, S., Neural Networks–A Comprehensive Foundation, Macmillan College Publishing Company, Inc.

[43] Haykin, S., “Neural networks expand SP’ Horizons,” IEEE Signal Processing Maga-

zine, Vol. 13, No. 2, pp. 24-49, 1996.

[44] Hecht-Nielsen, Robert, Neurocomputing, Addison-Wesley Publishing, 1989.

[45] Hertz, J., A. Krogh, and R.G. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley Publishing Company, 1991.

[46] Hirsch, M.W., and S. Smale, Differential Equations, Dynamical Systems, and Linear Algebra, Academic Press, New York, 1974.

[47] Hopfield, J.J., "Neural Networks and physical systems with emergent collective computational abilities," Proceeding of National Academy of Science, U.S.A., Vol. 79, pp. 2445-2558, 1982.

[48] Kaplan, J.L., and J.A. Yorke, "Chaotic behavior of multidimensional difference equations," in Functional Differential Equations and Approximation of Fixed Points, edited by H.-O Peitgen and H.-O Walther, Lecture Notes in Mathematics, Vol. 730, pp. 204-410, Springer-Verlag, Berlin, 1979.

[49] Kay, S.M., and S.L. Marple, "Spectrum analysis-A Modern perspective," Proceedings of the IEEE, Vol. 69, pp. 1380-1385, 1981.

[50] Knudsen, E.I., S. du Lac, and S.D. Esterly, "Computational maps in the brain," Annual Review of Neuroscience, Vol. 10, pp. 41-65, 1987.

[51] Kohonen, T., Associative Memory, Springer, New York, 1977.

[52] Kohonen, T., "Self-organized formation of topologically correct feature maps," Biological Cybernetics, Vol. 43, pp. 59-69, 1982.

[53] Kohonen, T., Self-Organization and Associative Memory, 3rd ed. New York, Springer-Verlag, 1988.

[54] Kohonen, T., "The 'neural' phonetic typewriter," Comput., Vol. 21, pp. 11-22, Mar. 1988.

[55] Kohonen, T., "The self-organizing map," Proceedings of the IEEE, Vol. 78, No. 9, pp. 1-13, 1990.

[56] Kohonen, T., J. Kangas, J. Laaksonen, and K. Torkkola, "LVQ-PAK: The learning vector quantization Program Package," Helsinki University of Technology, Finland, 1992.

[57] Kostelich, E.J., and J.A. Yorke, "Noise reduction: finding the simplest dynamical system consistent with the data," Physica D, Vol. 41, pp. 183-191, 1990.

[58] Kraaijveld, M.A., Jianchang Mao, Anil K. Jain, "A nonlinear projection method based on Kohonen's topology preserving maps," IEEE Transactions on Neural Networks,

Vol. 6, No. 3, pp. 548-559, 1995.

[59] Ladrapppier, F., “Some relations between dimension and Lyapunov exponent,” Comm. Math. Phys. Vol. 81, pp. 229-234, 1981.

[60] Landa, P.S., and M.G. Rosenblum, “Time series analysis for system identification and diagnostics,” Physica D, Vol. 48, pp. 232-254, 1991.

[61] Lapedes, R., and R. Farber, “Nonlinear signal processing using neural networks: prediction and system modeling,” Technical Report LA-UR87-2662, Los Alamos National Laboratory, Los Alamos, New Mexico, 1987.

[62] Linde, Y., A. Buzo, R.M. Gray, “An algorithm for vector quantizer design,” IEEE Transactions on Communications COM, Vol. 28, pp. 84-95, 1980.

[63] Linsay, P.S., “An efficient method of forecasting chaotic time series using linear interpolation,” Physics Letters A, Vol. 153, pp. 353-358, 1991.

[64] Linsker, R., “From basic network principles to neural architecture,” Proceedings of National Academy of Science, U.S.A., Vol. 83, pp. 7508-7512, 8390-8394, 8779-8783, 1986.

[65] Linsker, R., “Self-organization in a perceptual network,” Computer, pp. 105-117, March 1988.

[66] Lippmann, R.P., “An introduction to computing with neural nets,” IEEE ASSP Magazine, pp. 4-22, April, 1987.

[67] Lorenz, E. N., “Deterministic nonperiodic flow,” Journal of Atmospheric Science, Vol. 20, pp. 130-141, 1963.

[68] Luttrell, S.P., “Hierarchical vector quantization,” IEE Proceedings (London), Vol. 136, pp. 405-413, 1989.

[69] Luttrell, S.P., “Self-organization: A derivation from first principle of a class of learning algorithms,” IEEE Conference on Neural Networks, pp. 495-498, Washington, DC, 1989.

[70] Luttrell, S.P., “Code vector density in topographic mappings: Scalar case,” IEEE Transactions on Neural Networks, Vol. 2, pp. 427-436, 1991.

[71] Mackey, M.C., and L. Glass, “Oscillation and chaos in physiological control systems,” Science, Vol. 197, pp. 287-289, 1977.

[72] Malsburg, von der, “Self-organization of rientation sensitive cells in the striate cortex,” Kybernetik, Vol. 14, pp. 85-100, 1973.

[73] Malsburg, von der, and J.D. Cowan, Outline of a Theory for the Ontogenesis of Iso-

Orientation Domains in Visual Cortex, Biological Cybernetics, Vol. 45, pp. 49-56, 1982.

[74] Mandelbrot, B., The Fractal Geometry of Nature, Freeman, San Francisco, 1982.

[75] Mane, R., in Dynamical Systems and Turbulence, Vol. 898 of Lecture Notes in Mathematics, edited by D. Rand and L.S. Young, Springer, Berlin, pp. 230-242, 1981.

[76] Marple, S.L. Jr., Digital Spectral Analysis with Applications, Prentice-Hall, NJ, 1987.

[77] Martinetz, T.M., H.J. Ritter, and K.J. Schulten, "Three-dimensional neural net for learning visuomotor coordination of a robot arm," IEEE Transactions on Neural Networks, Vol. 1, pp. 131-136, 1990.

[78] Martinetz, Thomas M., et al., " "Neural-Gas" Network for vector quantization and its application to time-series prediction," IEEE Transactions on neural networks, Vol. 4, No. 4, pp. 558-562, 1993.

[79] Mead, W.C., R.D. Jones, Y.C. Lee, C.W. Barnes, G.W. Flake, L.A. Lee, and M.K. O'rourke, "Prediction of chaotic time series using CNLS-NET -- example: the Makcey-Glass equation," Technical Report: LA-UR-91-720, Los Alamos National Laboratory, Los Alamos, New Mexico, 1991.

[80] Miller, K.D., J.B. Keller, and M.P. Stryker, "Ocular dominance column development: Analysis and stimulation," Science, Vol. 245, pp. 605-615, 1989.

[81] Moody, J., and C.J. Darken, "Fast learning in networks of locally-tuned processing units," Neural Computation, Vol. 1, pp. 281-294, 1989.

[82] Nakano, K., "Associatron—a model of associative memory," IEEE Trans. SMC, Vol. 2, pp. 381-388, 1972.

[83] Narendra, K.S., and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," IEEE Transactions on Neural Networks, Vol. 1, No. 1, pp. 4-27, 1990.

[84] Obermayer, K., H. Ritter, and K. Schulten, "A principle for the formation of the spatial structure of cortical feature maps," Proceeding of National. Academy of Science USA, Vol. 87, pp. 8345-8349, Nov. 1990.

[85] Oja, E., "Self-organizing maps and computer vision," In Neural Networks for Perception (H. Wechsler, ed.), Vol. 1, pp. 368-385. San Diego, CA: Academic Press. 1992.

[86] Oppenheim, A.V., and R.W. Schafer, Discrete-time Signal Processing, Englewood Cliffs, NJ: Prentice Hall, 1989.

[87] Oppenheim, A.V., et al., "Signal processing in the context of chaotic signals," IEEE International Conference ASSP, Vol. 4, pp. 117-119, 1992.

[88] Orlando, J., R. Mann, and S. Haykin, “Classification of sea-ice using a dual-polarized radar,” IEEE Journal of Oceanic Engineering, Vol. 15, pp. 228-237, 1990.

[89] Ott, E., Chaos in Dynamical Systems, Cambridge University Press, 1993.

[90] Ott, E., C. Grebogi, and J.A. Yorke, “Controlling chaos,” Physical Review Letters, Vol. 64, No. 11, pp. 1196-1199, 1990.

[91] Packard, N.H., J.P. Crutchfield, J.D. Farmer, and R.S. Shaw, “Geometry from a time series,” Phys. Rev. Lett., Vol. 45, pp. 712-717, 1980.

[92] Papoulis, A., Probability, Random Variables, and Stochastic Processes, 3rd ed., McGraw-Hill, New York, 1991.

[93] Parker, T.S., and L.O. Chua, “Chaos: a tutorial for engineers,” Proceedings of the IEEE, Vol. 75, No. 8, pp. 982-1008, 1987.

[94] Poggio, T., and F. Girosi, “Regularization algorithms for learning that are equivalent to multilayer networks,” Science, Vol. 247, pp. 978, 1990.

[95] Powell, M.J.D., Approximation Theory and Methods, Cambridge University Press, Cambridge, 1981.

[96] Priestley, M.B., “State-Dependent Models: A general approach to nonlinear time series analysis,” Journal of time series analysis, Vol. 1, pp. 47-64, 1980.

[97] Principe, J.C., and J.M. Kuo, “Dynamics modeling of chaotic time series with neural networks”, Proceeding of Neural Information Processing Systems, Vol. 7, in press, 1995.

[98] Principe, J.C., A. Rathie, and J.M. Kuo, “Prediction of chaotic time series with neural networks and the issue of dynamic modeling,” International Journal of Bifurcation and Chaos, Vol. 2, No. 4, pp. 989-996, 1992.

[99] Principe, J.C., and L. Wang, “Nonlinear time series modeling with self-organizing feature maps,” Proceeding of IEEE Workshop on Neural Networks for Signal Processing, pp. 11-20, 1995.

[100] Rissanen, J., Stochastic Complexity in Statistical Inquiry, World Scientific, 1989.

[101] Ritter, H., “Asymptotic level density for a class of vector quantization processes,” IEEE Transactions on Neural Networks, Vol. 2, pp. 173-175, 1991.

[102] Ritter, H., “Learning with the self-organizing map,” In Artificial Neural Networks (T. Kohonen, K Makisara, O. Simula, and J. Kangas, eds.), Vol. 1, pp. 379-384, Amsterdam: North Holland, 1991.

[103] Ritter, H., and T. Kohonen, “Self-organizing semantic maps,” Biological Cybern.,

Vol. 61, pp. 241-254, 1989.

[104]Ritter, H.J., T. Martinetz, and K.J. Schulten, “Topology conserving maps for learning visuo-motor-coordination,” Neural Networks, Vol. 2, pp. 159-168, 1989.

[105]Ritter, H.J., T. Martinetz, and K. Schulten, Neural Computation and Self-Organizing Maps: An Introduction. Reading, MA: Addison-Wesley, 1992.

[106]Ritter, H., and K. Schulten, “On the stationary state of Kohonen’s self-organizing sensory mapping,” Biological Cybern., Vol. 54, pp. 99-106, 1986.

[107]Ritter, H., and K. Schulten, “Convergence properties of Kohonen’s topology conserving maps: Fluctuations, stability and dimension selection,” Biological Cybern., Vol. 69, pp. 59-71, 1988.

[108]Ritter, H., and K. Schulten, “Kohonen’s self-organizing Maps: Exploring their computational capabilities,” IEEE International Conference on Neural Networks (San Diego), Vol. I, pp. 109-116, New York: IEEE, 1988.

[109]Robert, H.-N., Neurocomputing, Addison-Wesley Publishing Company, 1989.

[110]Rosenblatt, F., “The Perceptron: A probabilistic model for information storage and organization in the brain,” Psychological Review, Vol. 65, pp. 386-408, 1958.

[111]Rumelhart, D.E., J.L. McClelland, and the PDP Research Group, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations, MIT Press, Cambridge, 1986.

[112]Rumelhart, D.E. and D. Zipser, Feature Discovery by Competitive Learning, Cognitive Science, Vol. 9, pp. 75-112, 1985.

[113]Sano, M., and Y. Sawada, “Measurement of the Lyapunov spectrum from a chaotic time series,” Physical Review Letters, Vol. 55, No. 10, pp. 1082-1091, 1985.

[114]Scharf, L.L., Statistical Signal Processing--Detection, Estimation, and Time Series Analysis, Addison-Wesley, 1989.

[115]Shinbrot, T., E. Ott, C. Grebogi, and J.A. Yorke, “Using chaos to direct trajectories to targets,” Phys. Rev. Lett., Vol. 65, pp. 3250-3261, 1990.

[116]Sidorowich, John J., “Modeling of chaotic time series for prediction, interpolation, and smoothing,” IEEE Int. Conf. ASSP, IV, pp. 121-127, 1992.

[117]Singer, A.C., G.W. Wornell, and A.V. Oppenheim, “Codebook Prediction: a nonlinear signal modeling paradigm,” IEEE Int. Conf. ASSP, V, pp. 325-329, 1992.

[118]Strum, R.D., and D.E. Kirk, First Principles of Discret Systems and Digital Signal Processing, Addison-Wesley Publishing, 1988.