

Entry no. 62



Chaos in Back Propagation Neural Networks and its Control

By

Md. Fazle Elahi Khan

A project submitted to the department of Electrical & Electronic Engineering in partial fulfillment of the requirements for the degree of Master of Science in Engineering




Khulna University of Engineering and Technology, Khulna 9203, Bangladesh

November 2010

Declaration

This is to certify that the project work entitled “**Chaos in Back Propagation Neural Networks and its Control**” has been carried out by Md. Fazle Elahi Khan in the Department of Electrical and Electronic Engineering (EEE), Khulna University of Engineering & Technology (KUET), Khulna, Bangladesh. The above research work or any part of the work has not been submitted anywhere for the award of any degree or diploma.


03.11.10

Signature of the Supervisor
Dr. Md Shahjahan
Associate Professor
Department of EEE, KUET, Khulna


3.11.10


Signature of the Candidate
Md. Fazle Elahi Khan
Roll No : 0503503


Approval

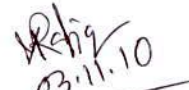



This is to certify that the project work submitted by Md. Fazle Elahi Khan entitled "**Chaos in Back propagation Neural Network and its Control**" has been approved by the Board of Examiners for the partial fulfillment of the requirements for the degree of **Master of Science in Engineering** in the Department of Electrical and Electronic Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh in November 03, 2010.

BOARD OF EXAMINERS

1. 
03.11.10

Dr. Md. Shahjahan
Associate Professor, Department of Electrical & Electronic Engineering
Khulna University of Engineering & Technology
Chairman
(Supervisor)
2. 
03.11.10

Head of the Department
Department of Electrical & Electrical Engineering
Khulna University of Engineering & Technology
Ex-Officio
3. 
03.11.10

Dr. Md. Abdur Rafiq
Professor, Department of Electrical & Electronic Engineering
Khulna University of Engineering & Technology, Khulna
Member
4. 
03.11.10

Dr. Md. Mahbubur Rahman
Professor, Computer Science Engineering Discipline
Khulna University, Khulna
Member
(External)

Acknowledgement

All praise to Allah tala for his taufiq for giving me ability to complete the thesis project. I would like to thank Dr. Md. Shahjahan for showing me how research is properly done. His blunt assessment of both my written work and ideas was at times hard to accept, but was a necessary component towards becoming a more mature writer and researcher. I will state for the record that any of my future achievements would not have been possible without the early guidance of Dr. Md. Shahjahan.

I would like to thank my external examiner Prof. Dr. Md. Mahbubur Rahman of Khulna University for his insightful comments, suggestions and constructive criticisms which really improved the quality of the thesis. Also I would like to extend my thanks to all members of the board.

I would like to thank Md. Assaduzzaman, H. M. Imran Hassan, and Sultan Uddin Ahmed to maintain a cooperative research group together. During my work, I went to several faculty members of this department and many anonymous persons. I would also like to thank all of them.

I also want to give thanks to my wife, Dr. Negar Fouzia who always inspired me for going ahead with study and knowledge. When I started my M.Sc course, I had only one kid of 5 years who had looked to me with an intuitive eye that what his father is doing sitting in a table and chair! Now I have a second kid of 4 years and he is doing the same! Thanks to my little family.



Abstract

It is interesting to determine the states of the neural network (NN) when it falls into chaos. This is because chaos has been found in biological brain. This paper investigates the several chaotic behaviors of supervised neural networks using Lyapunov exponent (LE), Hurst Exponent (HE), fractal dimension (FD) and bifurcation diagram. The update rule for NN trained with back propagation (BP) algorithm contains the function of the form $x(1-x)$ which is responsible for exhibiting chaos in the output of the network at increased learning rate. The HE is computed from the time series taken from the output of a NN. One can comment on the classification of the network from the values of HEs. We have examined the chaotic dynamics of NNs for two-bit parity, cancer, and diabetes classification problems. It is found that the distribution network output is absorbed at the increase of size of the network. As a result chaosness is marginally reduced.

Contents

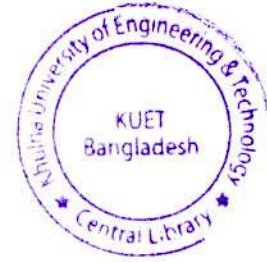
| | |
|--|----------|
| Title page | i |
| Declaration | ii |
| Approval | iii |
| Acknowledgement | iv |
| Abstract | v |
| Contents | vi |
| List of Figures | viii |
| List of Tables | ix |
| Chapter 1. Introduction | 1 |
| 1.1 Background..... | 1 |
| 1.2 Motivation | 2 |
| 1.3 Thesis organization | 3 |
| Chapter 2. Chaos and its Computation | 4 |
| 2.1 Chaos invariants..... | 4 |
| 2.2 Correlation Dimension and its Computation..... | 4 |
| 2.3 Computation of Correlation dimension | 5 |
| 2.4 Lyapunov exponent and its Computation..... | 6 |
| 2.5 Computation of Lyapunov Exponent..... | 7 |
| 2.6 Detection of unstable periodic orbits | 12 |
| 2.7 Hurst Exponent and its computation | 12 |

| | |
|---|----|
| Chapter 3. Chaos and Neural Networks | 16 |
| 3.1 Neural networks | 16 |
| 3.2 The Multilayered Feed forward Neural Network | 16 |
| 3.3 The Artificial Neuron..... | 17 |
| 3.4 Back Propagation | 18 |
| 3.5 Verhulst Equation | 19 |
| 3.6 Chaos in Back Propagation Learning | 19 |
| 3.7 Relation of Back Propagation and the Verhulst Equation | 20 |
| 3.7 .1 Explanation of Classification of Neural Network in Chaos with Hurst Exponent | 20 |
| 3.7.2 Network training | 21 |
| 3.7.3 Basics of BP in chaos | 22 |
| 3.8 Description of the Experiment and simulation result..... | 22 |
| 3.8.1 Using Hurst exponent | 23 |
| 3.8.2 Validation using bifurcation diagram | 26 |
| 3.8.3 Validation with Lyapunov Exponents: | 26 |
| 3.9 Other benchmark problems | 27 |
| Chapter 4. Conclusion | 31 |
| References | 32 |

List of Figures

| Figure No | Caption of the Figure | Page number |
|------------------|--|--------------------|
| 3.1 | Model of Artificial Neural Network | 16 |
| 3.2 | An Artificial Neuron Model | 18 |
| 3.3 | Bifurcation diagram of output of 2-bit parity network. | 26 |
| 3.4 | Bifurcation diagram of output of cancer problem. a) First pattern from Class 1, and b) First pattern from Class 2. | 28 |
| 3.5 | Return map for cancer problem | 29 |
| 3.6 | Bifurcation diagram of output of diabetes problem. a) Class 1, first pattern, and b) Class 2, first pattern | 30 |

List of Tables



| Table No | Caption of the Table | Page Number |
|-----------------|--|--------------------|
| 3.1 | Hurst Exponent and Fractal dimension of 2-bit parity Network for first output unit | 24 |
| 3.2 | Lyapunov exponents for the time series obtained from two-bit (XOR) network | 27 |
| 3.3 | Characteristics of Disease Classification Datasets | 27 |
| 3.4 | Hurst exponent and fractal dimension of Cancer problem for first output unit | 28 |
| 3.5 | Hurst exponent and fractal dimension of diabetes problem for first output unit | 30 |

Chapter 1

Introduction

1.1 Background

Chaos is a behavior that only appears in dynamic systems. A dynamic system consists of a phase space which represents all the possible states of a system. For a dynamical system to be classified as chaotic following properties should hold:

- Non-linearity
- Recursiveness
- Sensitivity to initial conditions
- Topologically mixing
- Dense periodic orbits

A dynamic system often exhibits nonlinear characteristics. A recursive system is the system that repeats after a particular time period. Say for example, rainfall repeats every year and it is chaotic. Chaos is characterized by its extreme sensitivity to initial conditions. That is, two nearly-indistinguishable sets of initial conditions for the same system could result in two final outcomes which differ vastly from each other.

Some dynamical systems are chaotic everywhere, but in many cases chaotic behavior is found only in a subset of phase space. A phase diagram of a periodic motion is called an orbit. The cases of most interest arise when the chaotic behavior takes place on an attractor, since then a large set of initial conditions will lead to orbits that converge to this chaotic region. Attractors that display chaotic features are called "strange attractors" and are very often fractal objects, some cross-section of them reveals similar structure on all scales. A fractal is a geometric object that can be divided into parts, each of which is similar to the original object. Fractals are generally self-similar and independent of scale. The conceptual roots of fractals can be traced to attempts to measure the size of objects for which traditional definitions based on Euclidean geometry or calculus fail. Fractal dimension is a static (or geometric) descriptor of the attractor, whereas the dynamics which formed the attractor is not described.

Complexity has always been part of our environment and many scientific fields have dealt with complex systems, which display variation without being purely random. Complex systems tend to be high dimensional and non-linear but may exhibit low-dimensional behavior. The different parts of complex systems are linked and affect one another. A complex system may exhibit deterministic and random characteristics with the level of complexity depending on the system's dynamics and its interactions with the environment. One of the objectives in quantifying complex systems is to explain emergent structures, self-organization. Phase transition is a property of self-organizing systems that move from static or chaotic states to a semi-stable balance between. Self-organized criticality is characterized by power-law distribution of events around the phase boundary.

Complexity might be related to chaos, a periodic long-term behavior that exhibits sensitive dependence on initial conditions and has limited predictability of the dynamics. Certain nonlinear dynamical systems under certain conditions exhibit chaos and detection of its emergence in the system would allow active control at a low cost not only to attain highly positive outcome but also to prevent costly crisis situations. This can be accomplished through the sensitivity to initial conditions, meaning that two points in a chaotic system may move in vastly different trajectories in their phase space, even if the difference in their initial configurations is very small.

1.2 Motivation

Recently, a chaotic neural network constructed with chaotic neurons has received much attention because of its rich dynamic behaviors and potential application of the associative dynamics in optimization and information processing, etc. [1, 2]. There are speculations that chaos plays important roles in neural networks. However, a definitive study on the role of chaos in neural networks is still missing. The chaotic neural network has shown a nonperiodic associative memory, but its associative memory is realized in the chaos dynamics of the network. The outputs of the network are non-periodic state which changes continuously and can not be stabilized in one of its stored patterns. One therefore meets difficulties in the application of the associative memory in information processing. The tasks of the brain include information processing and control. Some information processing or computing functions are now modeled by artificial neural networks (ANN), and almost none of the widely used ANNs require dynamical chaos as an essential element in their performance.

Multilayer feed-forward neural networks are widely used and are based on minimization of an error function. The basic learning method with chaos for the feed-forward networks is the backpropagation (BP) algorithm. BP learning uses the gradient descent procedure to modify the connection weights such that the network can approximate an objective function. BP works well for many problems, such as classification and function approximation.

Some methods have been proposed which embed chaotic dynamics into the neural network. Nozawa [3] showed the existence of chaos in Euler approximation of the Hopfield network by adding a negative self-feedback connection. Chaotic simulated annealing (CSA) is proposed by Chen and Aihara [4] and uses a sufficiently large negative self-feedback to a Hopfield neural network and gradually reduces the self-feedback. The detail analysis of chaos in the standard BP learning is still not studied. In this work, we study the chaos and its characteristics in the BP classification network. The major issues we discuss here are chaos formation, the time of chaos formation, effect of chaos on the network classification performance, explanation of classification using bifurcation diagram, analysis of network outputs using Lyapunov exponent, Hurst exponent and correlation dimension.

1.3 Thesis organization

In chapter 2, we have discussed about correlation dimension, Lyapunov exponent and Hurst exponent and their computation. These will be required for the computation of chaos from time series. In chapter -3, we have discussed about Neural Network, back propagation learning, chaos formation, Verhaast equation, chaos in classification problems such as two-bit parity, cancer, diabetes, bifurcation and the interpretation of result.



Chapter 2

Chaos and its Computation

This chapter describes chaos and how we can compute chaos from the time series.

2.1 Chaos invariants

Detecting the existence of deterministic chaos and its characteristics is one of the important studies from the viewpoint of time series analysis on chaos. For quantitative characterization of deterministic chaos, we have several quantities such as:

- 1) Correlation dimension (Grassberger-Procaccia algorithm)
- 2) Lyapunov exponents
- 3) Hurst exponent

In this chapter we consider the correlation dimension, Liapunov exponents, and the Hurst exponent. As for estimating the fractal dimensions, the Grassberger-Procaccia algorithm [5] has been widely applied to real time series data. The Liapunov exponents and its spectrum are also important statistics to quantify deterministic chaos. Several methods of estimating Liapunov spectra have been proposed [6, 7, and 8]. Even if the observable is only a single-variable time series in case of observing with enough number of data points, the Liapunov exponent and its spectrum of the original dynamical systems can be estimated with high accuracy. The Hurst exponent plays a central role in characterizing Brownian motion, pink noise and black noise. The Hurst exponent can also be calculated for chaotic time series.

2.2 Correlation Dimension and its Computation

The correlation dimension (CD) provides information on the minimum number of dynamic variables needed to model a system. It places a lower bound on the number of possible degrees of freedom. The correlation dimension indicates how likely it is to find another point within the distance a from a given point. The correlation dimension introduced by Grassberger and Procaccia (1983) allows an analyst to distinguish between determinism and stochasticity in a time-series. When the increase of the embedding dimension leads to convergence in the correlation dimension, the data are consistent with deterministic behavior.

As the embedding dimension goes to infinity the correlation dimension should not only saturate, but should also below, for the system to be chaotic. Short-term prediction will be possible within the parameters given by the Lyapunov exponents, but long-term prediction will not be possible. The correlation dimension may increase, when we increase the embedding dimension, but stays below the embedding dimension. Some long-term predictions are possible under these circumstances. If the correlation dimension keeps increasing in line with the embedding dimension, and stays close to the embedding dimension, no prediction is possible and the process is martingale. The Grassberger-Procaccia algorithm, although most popular, exhibits sensitivity to variations in its parameters that diminish its practical application [9].

2.3 Computation of Correlation dimension:

An often computed dimension in nonlinear time series analysis is the correlation dimension D_2 , which is a good approximation of the box-counting dimension D_0 : $D_2 \leq D_0$. Grassberger and Procaccia show in their seminal contribution [10] that D_2 can be evaluated by using the correlation integral $C(\varepsilon)$, which is the probability that a pair of points, chosen randomly in the reconstructed phase space, is separated by a distance less than ε .

The correlation integral can be approximated by the following sum,

$$C_N(\varepsilon) = \frac{2}{N(N-1)} \sum_{j=1}^N \sum_{i=j+1}^N \Theta(\varepsilon - |x_i - x_j|), \quad (2.1)$$

where $\Theta(\cdot)$ is the Heaviside function: $\Theta(x) = 1$ for $x \geq 0$ and 0 otherwise, and $|x_i - x_j|$ stands for the distance between points x_i and x_j . Grassberger and Procaccia argue that the correlation dimension is given by

$$D_2 = \lim_{\varepsilon \rightarrow 0} \lim_{N \rightarrow \infty} \frac{\log C_N(\varepsilon)}{\log \varepsilon}. \quad (2.2)$$

In practice, for a time series of finite length, the sum in Eq. (2.1) also depends on the embedding dimension m . Due to such dependencies, the correlation dimension D_2 is usually estimated by examining the slope of the linear portion of the plot of $\log C_N(\varepsilon)$ versus $\log \varepsilon$ for a series of increasing values of m . For $m < D_2$, the dimension of the reconstructed phase space is not high enough to resolve the structure of the dynamical state and, hence, the slope approximates the embedding dimension. As m increases, the resolution of the dynamical state in the reconstructed phase space is improved. Typically, the slope in the plot of $\log C_N(\varepsilon)$ versus $\log \varepsilon$ increases with m until it reaches a plateau; its value at the plateau is then taken as the estimate of D_2 . For an infinite and noiseless time series, the value of m at which this

plateau begins to satisfy is $m = \text{Ceil}(D_2)$, where $\text{Ceil}(D_2)$ is the smallest integer greater than or equal to D_2 [11]. In a realistic situation, short data sets and observational noise can cause the plateau onset to occur at a value of m which can be larger than $\text{Ceil}(D_2)$. Even so, the embedding dimension at which the plateau is reached still provides a reasonably sharp upper bound for the true correlation dimension D_2 .

After the delay time τ is chosen, the next step is to compute the correlation integral $C_N(\varepsilon)$ for a set of systematically increasing values of the embedding dimension m . A dimension estimate does not necessarily require such a one-to-one correspondence. For instance, consider a two-dimensional surface in a three-dimensional space. The projection of this surface onto a two-dimensional plane is still a two dimensional region. Thus, its dimension can be estimated even in a two-dimensional subspace. The D_2 is equivalent to fractal dimension (FD).

2.4 Lyapunov exponent and its Computation

Financial economics reduces a complex system of exchange to one-dimensional function, the prices of assets. The financial prices are the observable result of the interaction of traders with different motivation, ability to process information, rationality etc. The complex dynamics of the system occurs in multivariate state or phase space that can be reconstructed from scalar observations. According to Takens' delay embedding theorem an existing attractor should unfold its dynamics in a phase space with dimensions larger than two times the Hausdorff dimensions we discussed in the previous part. The embedding theorem though does not specify the choice of time delay to use for the reconstruction of the multidimensional vectors. This delay is determined by the Average Mutual Information. To determine the minimal sufficient embedding dimension we use the false nearest neighbor method [12]. False nearest neighbor method was introduced because of certain pitfalls of the correlation dimension estimation procedure, e.g. serial correlations and small sample fluctuations can easily be mistaken for nonlinear determinism. The false nearest neighbor method allows specifying a minimal temporal separation of valid neighbors. To quantify the stability of orbits around an attractor, it is necessary to calculate the Lyapunov exponents, i. e. the rates at which those orbits converge or diverge. Dechert and Gençay [13] propose an algorithm for estimation of the Lyapunov exponents when the equations generating the chaos are unknown. A positive largest Lyapunov exponent indicates chaos. Bask and Gençay [14] propose a test statistic for the presence of chaotic dynamics using the Lyapunov exponents.

To calculate the Lyapunov exponents from time series, it is first necessary to reconstruct the state space from the experimental data record. The original time series data and its time-delayed copies determine the topological structure of a dynamical system according to the Embedding Theorem [15]:

$$Y(n)=[X(t),X(t+1),\dots,X(t+(d-1)T)],$$

Where $Y(n)$ is the reconstructed d -dimensional state vector, $X(t)$ is the observed variable, T is a time lag, and d is the embedding dimension.

→ T is calculated from the first minimum of the Average Mutual Information (AMI) function [16].

→ d is computed with the Global False Nearest Neighbors (GFNN).

The dynamical properties of the system are the same in both the original and reconstructed state spaces, providing multidimensional dynamic information from a one-dimensional time series.

2.5 Computation of Lyapunov Exponent

For a dynamical system, sensitivity to initial conditions is quantified by the Lyapunov exponents. For example, consider two trajectories with nearby initial conditions on an attracting manifold. When the attractor is chaotic, the trajectories diverge, on average, at an exponential rate characterized by the largest one-dimensional Lyapunov exponent [17]. This concept is also generalized for the spectrum of one-dimensional Liapunov exponents, λ_j ($j = 1, 2, \dots, n$), by considering a small n -dimensional sphere of initial conditions, where n is the number of first order ordinary differential equations used to describe the dynamical system. As time t evolves, the sphere evolves into an ellipsoid whose principal axes expands (or contract) at rates given by the one-dimensional Lyapunov exponents. The presence of a positive exponent is sufficient for diagnosing chaos and represents local instability in a particular direction. Note that for the existence of an attractor, the overall dynamics must be dissipative, i.e. globally stable, and the total rate of contraction must outweigh the total rate of expansion. Thus, even when there are several positive one-dimensional Lyapunov exponents, the sum across the entire spectrum is negative. In most case a differential equation or difference equation is not given only a data set from an experiment, i.e., a time series.

A large number of authors have discussed the calculation of the spectrum of the one-dimensional Lyapunov exponents from time series [6,7,8]. There are two types of methods to find Lyapunov exponents. One is the Jacobian matrix estimation algorithm [6,7]. The Jacobian matrix estimation algorithm can find the whole spectrum of the one-dimensional Lyapunov exponents. It involves the least-square-error algorithm and the Gram-Schmidt procedure. Since this algorithm does not have built-in checks against noise, except the fact that the Lyapunov spectrum must not depend on the number of near neighbors and the dimension of the reconstructed state space, it would be better to use other methods which have a built-in-check. The method is called the direct method for finding the largest Lyapunov exponent. As for estimating largest Lyapunov exponents, several algorithms have been already proposed. These algorithms can be called a direct method, since they calculate the divergence rates of nearby trajectories and can evaluate whether the orbital instabilities are exponential on t or a power of t .

The direct method of Wolf is as follows. Let $x_0, x_1, x_2, \dots, x_{T-1}$ be a scalar time series. In the fixed evolution time program the time step $\Delta = t_{k+1} - t_k$ between replacements is held constant and normalized to 1. A d_E -dimensional phase portrait (rig embedding dimension) is reconstructed with delay coordinates, i.e., a point on the attractor is given by

$$x_t = (x_t, x_{t+1}, \dots, x_{t+d_E-1}).$$

Let $\|\cdot\|$ denote a norm, for example the Euclidian norm, the max norm or sum norm. Using the selected norm we find the nearest neighbor vector to the initial point vector

$$x_t = (x_t, x_{t+1}, \dots, x_{t+d_E-1}).$$

We denote the distance between these two points by $d(0)$. At a later time 1, the initial length $d(0)$ will have evolved to length $d'(1)$. The length element is propagated through the attractor for a time short enough so that only small scale attractor structure is likely to be examined. If the evolution time is too large we may see d' shrink as the two trajectories which define it pass through a folding region of the attractor. This would lead to an underestimation of the largest one-dimensional Lyapunov exponent λ . We now look for a new data point that satisfies the following two criteria:

- (i) Its separation, $d(1)$, from the evolved reference point is small,
- (ii) And the angular separation between the evolved and replacement elements is small.

If an adequate replacement point cannot be found, we retain the points that were being used. This procedure is repeated until the reference trajectory has traversed the entire data file, at which point we estimate the largest one-dimensional Lyapunov exponent as

$$\lambda = \frac{1}{M} \sum_{k=1}^M \log \frac{d'(k)}{d(k-1)}$$

where M is the total number of replacement steps. In the limit of an infinite amount of noise-free data our procedure always provides replacement vectors of infinitesimal magnitude with no orientation error, and λ is obtained by definition.

The algorithm proposed by Kantz [18] evaluates the following quantity

$$S(\tau) = \frac{1}{T} \sum_{t=0}^{T-1} \ln \left(\sum_{k_i}^M d(x_t, x_{k_i}; \tau) \right)$$

where T is the number of data points from the scalar time series, x_t (is a reference point, x_{k_i} is a ε -near neighbor of x_t , M is the number of nearest neighbors and τ is the relative time and $d(x_t, x_{k_i}; \tau)$ is the distance between $x_{t+\tau}$ and $x_{k_i+\tau}$. If the analyzed time series is produced from nonlinear dynamical systems with a positive largest one-dimensional Lyapunov exponent, there is a positive constant slope of the function $S(\tau)$ which corresponds to the largest one-dimensional Lyapunov exponent.

a. Embedding Methods

Let $u_i(t)$ ($i = 1, \dots, l$) be a set of l measurements. In principle, the measured time series come from an underlying dynamical system that evolves the state variable in time according to a set of deterministic rules, which are generally represented by a set of differential equations, with or without the influence of noise. Mathematically, any such set of differential equations can be easily converted to a set of first-order, autonomous equations. The dynamical variables from all the first-order equations constitute the phase space, and the number of such variables is the dimension of the phase space, which we denote by M . The phase-space dimension can in general be quite large. For instance, in a fluid experiment, the governing equation is the Navier Stokes equation which is a nonlinear partial differential equation. In order to represent the system by first-order ordinary differential equations via, say, the procedure of spatial discretization, the number of required equations is infinite. The phase-space-dimension in this case is thus infinite.

However, it often occurs that the asymptotic evolution of the system lives on a dynamical invariant set of only finite dimension. The assumption here is that the details of the system equations in the phase space and of the asymptotic invariant set that determines what can be observed through experimental probes are unknown. The task is to estimate, based solely on one or few time series, practically useful statistical quantities characterizing the invariant set, such as its dimension, its dynamical skeleton, and its degree of sensitivity on initial conditions. The delay-coordinate embedding technique established by Takens provides a practical solution to this task. In particular, Takens' embedding theorem guarantees that a topological equivalence of the phase space of the intrinsic unknown dynamical system can be reconstructed from time series, based on which characteristics of the dynamical invariant set can be estimated. Takens' delay-coordinate embedding method goes, as follows. From each measured time series $u_i(t) (i=1, \dots, l)$, the following vector quantity of q components is constructed,

$$u_i(t) = \{u_i(t), u_i(t+\tau), \dots, u_i[t+(q-1)\tau]\},$$

where τ is the delay time. Since there are l time series, a vector with $m \equiv ql$ components can be constructed, as follows:

$$\begin{aligned} x(t) &= \{u_1(t), u_2(t), \dots, u_l(t)\} \\ &= \{u_1(t), u_1(t+\tau), \dots, u_1[t+(q-1)\tau]\}, \\ &\quad \{u_2(t), u_2(t+\tau), \dots, u_2[t+(q-1)\tau]\}, \dots, \\ &\quad \{u_l(t), u_l(t+\tau), \dots, u_l[t+(q-1)\tau]\}, \end{aligned}$$

where m is the embedding dimension. Clearly, the delay time τ and the embedding dimension m are the two fundamental parameters in the delay coordinate embedding method.

b. Delay time τ :

In order for the time-delayed components $u_i(t+j\tau) (j=1, \dots, q-1)$ to serve as independent variables, the delay time τ needs to be chosen carefully. Roughly, if τ is too small, then adjacent components $u_i(t)$ and $u_i(t+\tau)$ will be too correlated for them to serve as independent coordinates. If, on the other hand, τ is too large, then neighboring components are too uncorrelated for the purpose. Empirically, one can examine the autocorrelation function of $u_i(t)$ and decide a proper delay time [19]. In particular, one computes

$$c(T) \equiv \frac{\langle u_i(t)u_i(t+T) \rangle}{\langle u_i^2(t) \rangle},$$

where $\langle \cdot \rangle$ stands for time average. The delay time τ can be chosen to be the value of T such that $\frac{c(T)}{c(0)} \approx e^{-1}$.

c. Embedding dimension m

In order to have a faithful representation of the true dynamical system, the embedding dimension m should be sufficiently large. Takens' theorem provides a lower bound for m . In particular, suppose the dynamical invariant set lies in a d -dimensional manifold (or subspace) in the phase space. Then, if $m > 2d$, the m -dimensional reconstructed vectors $\mathbf{x}(t)$ have a one-to-one correspondence to the vectors of the true dynamical system. This result can be understood by the following simple mathematical argument. Consider two smooth surfaces of dimensions d_1 and d_2 in an M -dimensional space and examine the set of their intersections. The question is whether they intersect generically in the sense that the intersections cannot be removed by small perturbations to either surface. The natural approach is then to look at the dimension d_I of the intersecting set, which is

$$d_I = d_1 + d_2 - M.$$

If $d_I \geq 0$, the intersection is generic. For example, consider the intersection of two one-dimensional curves in a two-dimensional plane: $d_1 = d_2 = 1$ and $M = 2$. We obtain: $d_I = 0$, which means that the intersecting set consists of points, and the intersections are generic because small perturbations cannot remove them. If, however, $M = 3$, then $d_I < 0$, which means that two one-dimensional curves do not intersect generically in a three-dimensional space. For the case of embedding, we can ask whether the dynamical invariant set would intersect itself in the reconstructed phase space. In order to obtain a one-to-one correspondence between points on the invariant sets in the actual and reconstructed phase spaces, self-intersection must not occur. Thus, taking $d_1 = d_2 = d$ and $M = m$, no self-intersection requires $d_I < 0$, which means that $m > 2d$.

While Takens' theorem assumes that the relevant dimension d of the set is that of the manifold in which the set lies, this dimension can be quite different from the dimension of the set itself, which is physically more relevant.

2.6 Detection of unstable periodic orbits

A chaotic set has embedded within itself an infinite number of unstable periodic orbits. A fundamental feature that differs a deterministic chaotic system from a stochastic one is the existence of an infinite number of unstable periodic orbits which constitute the skeleton of the chaotic invariant set.

At a fundamental level, unstable periodic orbits embedded in a chaotic invariant set are related to its natural measure, which is the base for defining physically important quantities such as the fractal dimensions and Lyapunov exponents. At a practical level, successful detection of unstable periodic orbits indicates the deterministic origin of the underlying dynamical process.

One of the most important problems in dealing with a chaotic system is to compute long term statistics such as averages of physical quantities, Lyapunov exponents, dimensions, and other invariants. The interest in the statistics roots in the fact that trajectories of deterministic chaotic systems are apparently random and ergodic. These statistical quantities, however, are physically meaningful only when the measure being considered is the one generated by a typical trajectory in the phase space. This measure is called the natural measure and it is invariant under the evolution of the dynamics [20].

2.7 Hurst Exponent and its computation

Einstein discovered that the distance covered by a random particle undergoing random collisions from all sides is directly related to the square root of time. Thus

$$R = kT^{1/2}$$

Where R is the distance covered, k is some constant and T is the time index. Hurst proposed a generalization of Brownian motion that could apply to a broader class of time-series. His generalized equation is

$$R/S = kT^H$$

Where R/S = rescaled range (range/standard deviation), T = index for number/time of observations, K = some constant for the time-series, H = Hurst exponent. Thus, Hurst

generalized the $T^{1/2}$ law to a T^H law. Analogously, Brownian motion can be generalized to fractal Brownian motion. Fractal Brownian motion exists whenever the Hurst exponent is well-defined.

The R/S value is called the rescaled range and is a dimensionless ratio formed by dividing the range by the standard deviation of the observations. It scales as one increase the time increment by a power law value equal to H . This is the key point in Hurst's analysis: by rescaling, Hurst could compare diverse data points, including periods of time that may be separated by many years. In addition, this type of analysis can be used to describe time series that possess no characteristic scale. This equation has a characteristic of fractal geometry: it scales according to a power law. In the lung, for instance, the size of each branch decreases in scale according to an inverse-power law. Likewise, the R/S function increases as a power of H . If the data of the system being measured were independently distributed, or followed a random walk, the equation would fit with Einstein's "T to the one-half" rule, and the value of the Hurst exponent would be $1/2$.

There are three possibilities for values of H . If $H = 0.5$ the system follows a random walk. We recover the original scenario of Brownian motion. If not, the observations are not independent; each carries a memory of events which precede it.

- $H = 0.5$

Independent series. (Brown noise or Brownian motion) The series is a random walk.

- $0 \leq H < 0.5$

Antipersistent series. (Pink noise) The system is covering less distance than a random walk. Thus, it has a tendency to reverse itself often. If increasing, it is more likely to be decreasing the next period; if decreasing, it is more likely to be increasing.

- $0.5 < H \leq 1$

Persistent series. (Black noise) This series covers more distance than a random walk. Thus, if the system increases in one period, it is more likely to keep increasing in the immediately following period.

Thus the Hurst exponent is a useful measure for fractal distributions. There is no characteristic time scale in such a distribution. Hence an exponential, or relative, relation dominates over a polynomial, or absolute, characterization.

The following statements are believed equivalent for a time-series:

1. The Hurst exponent is well-defined for the time-series.

2. The time-series exhibits fractional Brownian motion.
3. The probability distribution is stable (Paretian or Levy).
4. The slope of the log-log R/S graph is constant.

The value $1/H$ is the fractal dimension of the probability space. The random walk has a fractal dimension (capacity) of $1/0.5 = 2$. Thus it completely fills the phase space. The value $2 - H$ is the fractal dimension of the time-series. The value $2H + 1$ is the rate of decay of the Fourier series. This means the Fourier coefficients decrease in proportion to $1/f^{(2H+1)}$.

Estimations of H can be found by taking the slope of the log/log graph of R/S versus T , where

$$\log(R/S) = \log(kT^H) = \log(k) + H \log(T).$$

If there is no long term memory present, scrambling the data should have no effect on this estimate of H . If, however, we destroy the structure by randomizing the data points, the estimate of H should be much lower. Therefore, the Hurst exponent is a meaningful measure of the memory of a system.

Algorithm for the Hurst Exponent:

In the following we use the notation as in our C++ program. Let

$$u_0, u_1, \dots, u_{T-1}$$

be a given time series of length T . We divide this time period into a contiguous sub periods of length n , such that

$$an = T.$$

We label each sub period I_j , with $j = 0, 1, 2, a - 1$. Each element in I_j is labeled $N[j][k]$ such that $k = 0, 1, 2, \dots, n - 1$. Thus N is a $a \times n$ matrix. For each I_j of length n , the average value is defined as

$$E_j := \frac{1}{n} \sum_{k=0}^{n-1} N[j][k]$$

where E_j is the average value of the u_i contained in sub period I_j of length n . The time series of accumulated departures $X[j][k]$ from the mean value for each sub period I_j is defined as

$$X[j][k] := \sum_{i=0}^{k-1} (N[j][i] - E_j), \quad j = 0, 1, 2, \dots, a-1, \quad k = 0, 1, 2, \dots, n-1$$

Thus X is also an $a \times n$ matrix. The range is defined as the maximum minus the minimum value of $X[j][k]$ within each sub period I_j

$$R_j := \max(X[j][k]) - \min(X[j][k])$$

where $k = 0, 1, 2, \dots, n-1$. The sample standard deviation calculated for each sub period I_j is

$$S_{I_j} := \left(\frac{1}{n} \sum_{k=0}^{n-1} (N[j][k] - E_j)^2 \right)^{1/2}$$

Each range, R_{I_j} , is now normalized by dividing by the S_{I_j} corresponding to it. Therefore, the rescaled range for each I_j sub period is equal to R_{I_j} / S_{I_j} . We had *contiguous sub periods* of length n . Therefore, the average R/S value for a fixed length n is defined as

$$(R/S)_n := \frac{1}{a} \sum_{j=0}^{a-1} \frac{R_{I_j}}{S_{I_j}}$$

The length n is increased to the next higher value, and $(T - 1) / n$ is an integer value. We use values of n that include the beginning and ending points of the time series, and steps described above are repeated until $n = (T - 1)/2$.

We can now apply

$$(R/S)_n = cn^H$$

or

$$\log((R/S)_n) = \log(c) + H \log(n)$$

by performing an ordinary least squares regression on $\log(n)$ as the independent variable and $\log(R/S)_n$ as the dependent variable. The intercept is the estimate for $\log(c)$, the constant. The slope of the equation is the estimate of the Hurst exponent, H . In general, one runs the regression over values of $n \geq 10$. Small values of n produce unstable estimates when sample sizes are small.

Chapter 3

Chaos and Neural Networks

3.1 Neural networks

Neural networks are mathematical models based on observations that have been made on biological neuron cells. One has to keep in mind that artificial neural networks are a mere simple and small model of the brain neuron. The brain is composed of approximately 10^{10} neurons, each one of which can have up to 10000 connections to other neurons; there are different types of neurons and around 50 different chemicals that are used for transmission of signals. In general, a neural network is an assembly of interconnected artificial neurons. Each neuron has inputs and produces a single output which then forms the input for other neurons in the network [21].

3.2 The Multilayered Feedforward Neural Network

There exist different types of neural networks depending on the topology of the connections. A multilayered feedforward neural network is composed of several layers. Each layer consists of several neurons parallel to each other and is fully connected to the neuron of the layers above and below. Each neuron receives for input the output values of all the neurons of the layer below. The output produced serves as an input for all the neurons of the layer above as shown in Fig.3.1.

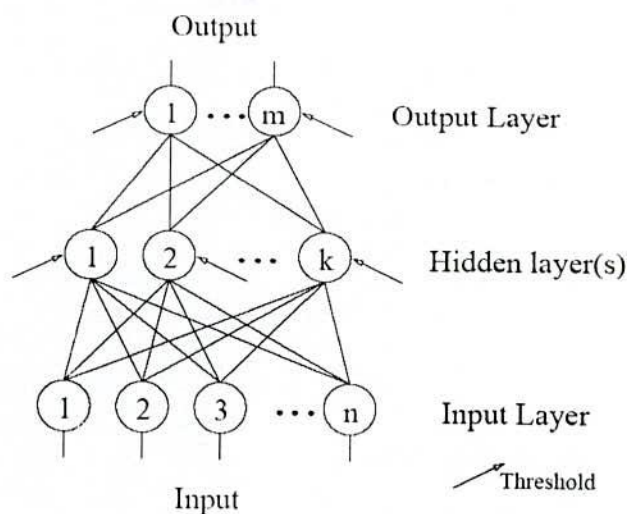


Fig. 3.1: Model of Artificial Neural Network

The input signal propagates through the network in a forward direction, on a layer-by-layer basis. The layers in-between, called the hidden layers, sequentially compute their outputs and pass them on as inputs to the next layer until the top most layers; the output layer is reached which produces the actual output of the network.

3.3 The Artificial Neuron

Figure 3.2 shows the mathematical model of the biological neuron. The biological neuron, simply described, is a cell that receives electrical signals, by chemical means, from extensions that are called dendrites. These signals are weighted by the strength of the connection to the cell. An artificial neuron has several input lines and one output line. To each input line a weight is assigned. The input of the line is multiplied with that weight value. Then the products of each incoming line are summed to produce the activation value. In general, there are three types of activation functions. First, there is the Threshold Function which takes on a value of 0 if the summed input is less than a certain threshold value (v), and the value 1 if the summed input is greater than or equal to the threshold value. Secondly, there is the Piecewise-Linear function. This function again can take on the values of 0 or 1, but can also take on values between that depending on the amplification factor in a certain region of linear operation. Thirdly, there is the sigmoid function. This function can range between 0 and 1, but it is also sometimes useful to use the -1 to 1 range. An example of the sigmoid function is the hyperbolic tangent function. Then the threshold value of the neuron is subtracted from the activation value and the result is used in a non-linear function which produces the output of the neuron [22]. We will use the sigmoid function $f_{sig}(x) = \frac{1}{1 + e^{-x}}$ (fig 3.2). The parameter corresponds to the steepness of the transition from 0 to 1. Thus, to summarize, this is how the output of the neuron is computed:

- Activation value: $A = \sum_i w_i I_i + \theta$
- Output value: $O = f_i(A) = \frac{1}{1 + e^{-s/I}}$

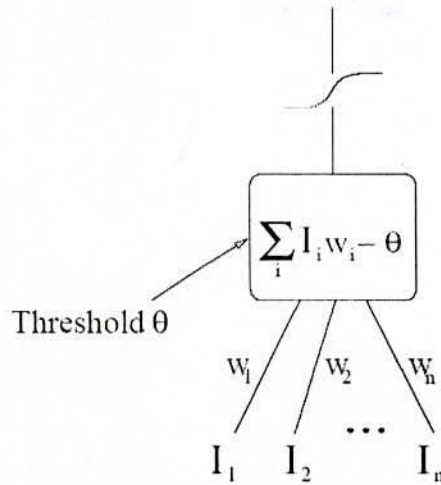


Figure 3.2: An Artificial Neuron Model

Where I_i and w_i are the input value and the weight that correspond with the i th input line of the neuron. T is the temperature and corresponds to the steepness of the sigmoid function.

3.4 Back Propagation

Back propagation was first proposed by McClelland and Rumelhart [23] and is the widest used algorithm for training neural networks today. This learning is called supervised learning. This means that a supervisor supplies the neural network with target value(s) that correspond to the input pattern that is presented. These are used to compute the error of the actual output. This error information is then used by the learning algorithm to determine how the weights are to be adapted in order to obtain an output closer to the desired one. Hopefully, the algorithm will converge to correct weight combination. We rely on the generalizing ability of the neural network to produce the right output on similar inputs. The generalizing ability is related to the topology of the network, the weight combination reached and also the representativeness of the learning set that was used to train the network.

In this version of the back propagation just described the weights are adapted after each one of the patterns of the set is fed to the network. We call this non-epoch learning. In epoch learning the algorithm is slightly different. The computations are the same, with the only difference that the adaptation of the weights is done after the whole set is used. This means that at each step, the errors and the adaptations are computed but they take place after the whole set is used. This is of course only possible when the whole set is known.

3.5 Verhulst Equation

$$x_{t+1} = \alpha x_t (1 - x_t)$$

This family of equations with α as a parameter is called logistic maps. The parameter gives the steepness of the function and is a very important for the behavior of the function as we shall see.

We divided the description in different regions depending on the value of the parameter α :

- $0 < \alpha < 1$: In this region the iteration converges to the trivial solution $x_0 = 0$.
- $1 < \alpha < 3$: The iteration converges to the non-trivial solution $x_1 = 1 - 1/\alpha$ for any start value in $(0, 1)$. It does so the fastest when $\alpha = 2$ and the first derivative of $v(x)$, $\frac{dv(x)}{dx}$ is equal to zero.
- $3 < \alpha < 3.569999$: We pass value 3 something interesting happens. The iteration no longer converges to a single value but a so called bifurcation occurs and the iteration converges to 2 values periodically. As we keep increasing α more and more bifurcations keep appearing faster and faster. The bifurcations continue until reaches the value 3.569999 where we enter the chaotic regime.
- $3.569999 < \alpha < 4$ The iterative process does not settle to predictable periodic values but becomes unpredictable. For example we have a periodicity of 3 for $\alpha = 3.839$ and one of period 5 for $\alpha = 3.74$. These regions also undergo bifurcations when is increased.

3.6 Chaos in Back Propagation Learning

As mentioned in the previous chapter the back propagation algorithm appears to have similar behavior with the verhulst equation. In [24] the authors showed that by varying the learning rate, the output of the network exhibits a period doubling route to a chaotic regime while learning in a similar manner the logistic map does. Others have also showed a relation of neural networks and chaotic behavior [25, 26]. Wang showed that chaos can occur even in a simple neural network consisting of two neurons, one inhibitory and one excitatory. This was proved analytically.

3.7 Relation of Back Propagation and the Verhulst Equation

Before we continue with the experiments, we will first present the relation of the back propagation algorithm and the verhulst equation. If we look at the adaption rule for weights and thresholds in the back propagation algorithm we see a similarity with the verhulst equation. The variable corresponding to the variable in the verhulst equation is in this case the learning rate.

$$x_{t+1} = \alpha x_t (1 - x_t)$$

Adaptation rule for output nodes:

$$\Delta_p W_{ji} = \beta (t_{pj} - O_{pj}) O_{pj} (1 - O_{pj}) O_{pi}$$

and for hidden nodes:

$$\Delta_p W_{ji} = \beta \left(\sum_{k=1}^n \delta_{pk} W_{kj} \right) O_{pj} (1 - O_{pj}) O_{pi}$$

The interesting part in the adaption equation is the $O(1-O)$ which is the same as the term in the verhulst equation. In the second equation the weight W is a part of the equation for O_j , namely $O_j = 1/(1 + e^{-\sum W_j O_j})$ which makes the whole a recursive system. The changes in W_i affect the output O_j and O_j in turn affects the adaption of W_i . Because of this we expect a similar behavior when we look at the values that the network produces while the back propagation is used for teaching the network.

3.7.1 Explanation of Classification of Neural Network in Chaos with Hurst Exponent

It is very difficult to compute classification of the neural network (NN) when it falls into time out condition during chaos. The update rule for NN trained with back propagation (BP) algorithm involves the function of the form $x(1-x)$ which is responsible for exhibiting chaos in the output of the network at increased learning rate. The HE is computed from the time series taken from the output of a NN. The distinct values of HE for different input patterns suggest that the misclassification & classification probabilities for xor network can be determined. The result is validated with the help of bifurcation diagram of the output of the NN. It is found that the values of HE are repositioned marginally depending on the size of NN. In effect, the NN escapes from the randomness at larger size of NN.



3.7.2 Network training

Back-propagation (BP) is a famous training method used in multilayer feed-forward neural networks (NNs). NNs, that are networks of artificial neurons, are capable of variety of task such as classification, pattern recognition, forecasting, function approximation etc. The artificial neuron is a simple mathematical model motivated from the biological neuron of the brain. The interesting property of these networks is their ability to learn. They do so by adapting the weights of connections between the neurons. The widest used type of neural network today is multilayer feed-forward neural network using the backpropagation algorithm as the learning method.

Chaos in neural network has attracted much interest in recent years [1,2]. There have been many reports that claim chaos plays important role in neural networks. The phenomena that appears random but is regulated by under a deterministic rule is called deterministic chaos. Chaos can be found in living organs such as brain activity [3]. In the brain, spontaneous neural activity exhibits some properties of deterministic chaos [4]. Several papers have discussed the emergence of chaos in NNs[5,6]. Nozawa [7] showed the existence of chaos in Euler approximation of the Hopfield network by adding a negative self-feedback connection. Chaotic simulated annealing (CSA) is proposed by Chen and Aihara [8] and uses a sufficiently large negative self-feedback to a Hopfield neural network and gradually reduces the self-feedback. The object of above paper was to get the benefit from NN in chaos.

The occurrence of chaos in backpropagation algorithm was shown in [9]. The BP network exhibits chaos due to the presence of a term of the form $x(1-x)$ in the update rule of it, although other parameters are involved in the update rule. Weight space instead of output of the network was taken against learning rate in [9] to describe the chaos formation. The role of chaos for classification was not addressed in any of the above reports. Therefore, a definite study on this issue is still missing. This paper explains empirically the classification probability of BP network at chaotic regime with the help of Hurst exponent (HE). HE is a new statistical tool to analyse the time series. This method represents the entire time series into several subintervals. This has a great advantage for the system where time series are collected in a regular interval fashion.

3.7.3 Basics of BP in chaos

Fully connected multilayer neural networks, consist of one input layer, one output layer, and one or more hidden layers is trained by BP. Weights are updated minimizing the MSE function:

$$E = \frac{1}{n} \sum_{p=1}^n \sum_{i=1}^m (y_{pi} - t_{pi})^2$$

Sigmoid logistic function is used as activation function in BP:

$$f(x) = \frac{1}{1 + e^{-x}}$$

One needs to compute the derivative of the sigmoid function to update the weight in BP algorithm. The derivative of it is $f(x)(1-f(x))$ which is the main driving force to converge the chaos in BP network. The update rules for output and hidden unit weights are as follows.

$$\Delta_p W_{ji} = \beta (t_{pj} - O_{pj}) O_{pj} (1 - O_{pj}) O_{pi} \quad (1)$$

$$\Delta_p W_{ji} = \beta \left(\sum_{k=1}^n \delta_{pk} W_{kj} \right) O_{pj} (1 - O_{pj}) O_{pi} \dots \dots \dots (2)$$

This equation has complete chaotic dynamics for appropriate choice of parameter alpha. It has been known that the chaos may appear in weight space and output space of the network when the learning rate is increased. In this paper the output of the NN is taken as main parameter with respect to learning rate. Since the output is the appropriate candidate for classifying the input patterns.

3.8 Description of the Experiment and simulation result

The first derivative of sigmoid logistic function, which is used in adapting weight in a NN, contains a term similar with that of verhulst equation.

In chaotic regime, network is unable to classify all the patterns. We investigate this classification probability (CP) using Hurst exponent and fractal dimension and validated by bifurcation diagram.

In order to investigate the classification of NN, we take 2-bit parity problem. Back-propagation learning is used with online mode. The NN is trained with BP algorithm with a learning rate starting from 0.1 with a step of 0.1. The outputs of last 50 iterations are taken among 20,000 iterations in each run. The network is trained 600 times. Hence the time series contains 30,000 data points. These data points are taken to construct the bifurcation diagram. In order to see the classification in chaos, the experimental data are taken when NN runs in

chaos. Additional 30,000 data points are collected with different initial weights and with a learning rate of 30.00.

Three different sizes of the NN are taken for the experiment. They are 2-2-2, 2-3-2 and 2-4-2. A size of 2-2-2 means two inputs - two hidden units - two output units NN. An additional hidden unit gives four additional connection weights in a NN. For each size of NN, a total of 30,000 data points are collected from the training of NN. HE is computed from these time series data. Since the NN has winner-takes-all strategy to make decision at the output of the NN, there are two outputs.

3.8.1 Using Hurst exponent

The numerical values of HE and fractal dimension are listed in Tables 3.1 & 3.2. There are some interesting observations from the values of HE described below. These are the main focuses of the work.

- I. **Randomness to persistence:** We know that if $HE = 0.5$ the data are determined to be random. From the Table 3.1, one can see that the data for pattern '11' has the HE approximately equal to 0.5 for a NN size of 2-2-2. Since the number of outputs of NN is two. We can easily understand that the pattern '11' is misclassified since much randomness is included in the data. It is observed that the randomness is reduced and persistence is increased gradually from pattern '11' to '10' '01' and '00'. This is reasonable since online mode learning is usually noisy than batch mode training to achieve the target concept.
- II. **Hurst exponent to classification probability:** Can we correlate HE with network classification? Our suggestion is yes. There are a number of reasons behind this. From the Tables 3.1& 3.2, it is seen that the HE is approximately unity for pattern '00'. We want to claim that this pattern is classified. Since there is no uncertainty for other chances. However, the values of HEs are in between 0.6 and 0.8 for '01' and '10' patterns. This means there is a mixture of probabilities whether or not patterns '01' and '10' are classified. We can say that pattern '01' is classified since the value of HE corresponding to this pattern is approximately 0.80 or more. Therefore, this pattern is certainly classified. The classification uncertainty increases for the pattern '10'. Since the value of HE for pattern '10' is approximately 0.6 or more. This means that the chance of classification is slightly larger than the pattern '11'. If we consider that the pattern '10' is classified then the total classification probability of the network is $\frac{3}{4} =$

75% in chaos and misclassification probability is 25%. However, this is not always the case because a considerable number of runs fall into time out conditions.

III. **Neural network size to classification probability:** There is a nice dependence of classification probability on the size of the network in chaos. It is known that the minimum network for 2-bit parity is 2-2-2. This is the necessary and sufficient size for this problem. In chaos, the possibility of weight oscillation is limited. That is why sometimes the network goes into time out condition. We found the value of HE for '11' pattern in 2-2-2 NN is about 0.50. This is very near to the random series. The probability of classification is almost nil. However, when we increase the size of the NN to 2-3-2, the probability of misclassification decreases by a certain amount as shown in Table 3.1 & 3.2 (see column 4 and 6). The randomness in the time series is* reduced by $(0.5746-0.5010) = 0.0736$ for 2-3-2 network and $(0.5819-0.5010) = 0.0809$ for 2-4-2 network respectively as shown in Table 3.1. The values of other HEs for other patterns are approximately similar from 2-2-2 to 2-3-2 and 2-4-2 sized NN.

IV. **Hurst exponent to neural network size:** One interesting thing is that HE is related in some extend with the size of the network. If the size of the network is increased the HE is also increased meaning that the network tends to reduce classification uncertainty and moves towards classification states, although it is not clear how many times the network classifies and how many times it does not classify. It is clear that the network escapes from the randomness at increased size of the network.

V. **Hurst exponent to fractal dimension:** It is known that fractal dimension (FD) is related with Hurst exponent with an equation of $FD = 2 - H$. One can comment on their dynamics with fractal dimension. Large FD indicates large amount of noise is included in the time series. Therefore higher FD means that the underlying dynamics behind the series involves much randomness in one or the other.

Table 3.1 Hurst Exponent and Fractal dimension of 2-bit parity network for first output unit.

| Pattern | Network 2-2-2 | | Network 2-3-2 | | Network 2-4-2 | |
|---------|----------------|-------------------|----------------|-------------------|----------------|-------------------|
| | Hurst Exponent | Fractal dimension | Hurst Exponent | Fractal dimension | Hurst Exponent | Fractal dimension |
| 00 | 0.9898 | 1.0102 | 0.9889 | 1.0111 | 0.9877 | 1.0123 |
| 01 | 0.7978 | 1.2022 | 0.8055 | 1.1945 | 0.8164 | 1.1836 |
| 10 | 0.6865 | 1.3135 | 0.6500 | 1.3500 | 0.6786 | 1.3214 |
| 11 | 0.5010 | 1.499 | 0.5746 | 1.4254 | 0.5819 | 1.4181 |

3.8.2 Validation using bifurcation diagram

The truth table for two-bit parity problem is shown below. In each run we kept a record of the last 50 output values generated by the network. These values are then plotted against the corresponding learning rate value resulting in a bifurcation diagram. Using the bifurcation diagram, classification can be explained as shown in Fig. 3.3. The winner-take-all network has two output units for two classes as arranged in Tables 3.1 and 3.2. The target outputs for the input patterns '00', '01', '10' and '11' are assigned as '01' '10' '10' and '01' respectively. The actual outputs of the network are shown in Fig. 3.3 for all input patterns from learning rate 0 to 60 with a step of 0.1. From the Fig. 3.3, one can understand very easily that first three patterns 00, 01, 10 are classified but the fourth pattern 11 was not classified since it is similar with pattern 10. The output should be identical for input patterns of '10' and '01' and the same are true for input patterns '00' and '11'. The oscillation of bifurcation of the output for '11' pattern is similar with opposite class when the network enters into chaotic regime. Therefore, this pattern '11' is misclassified.

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

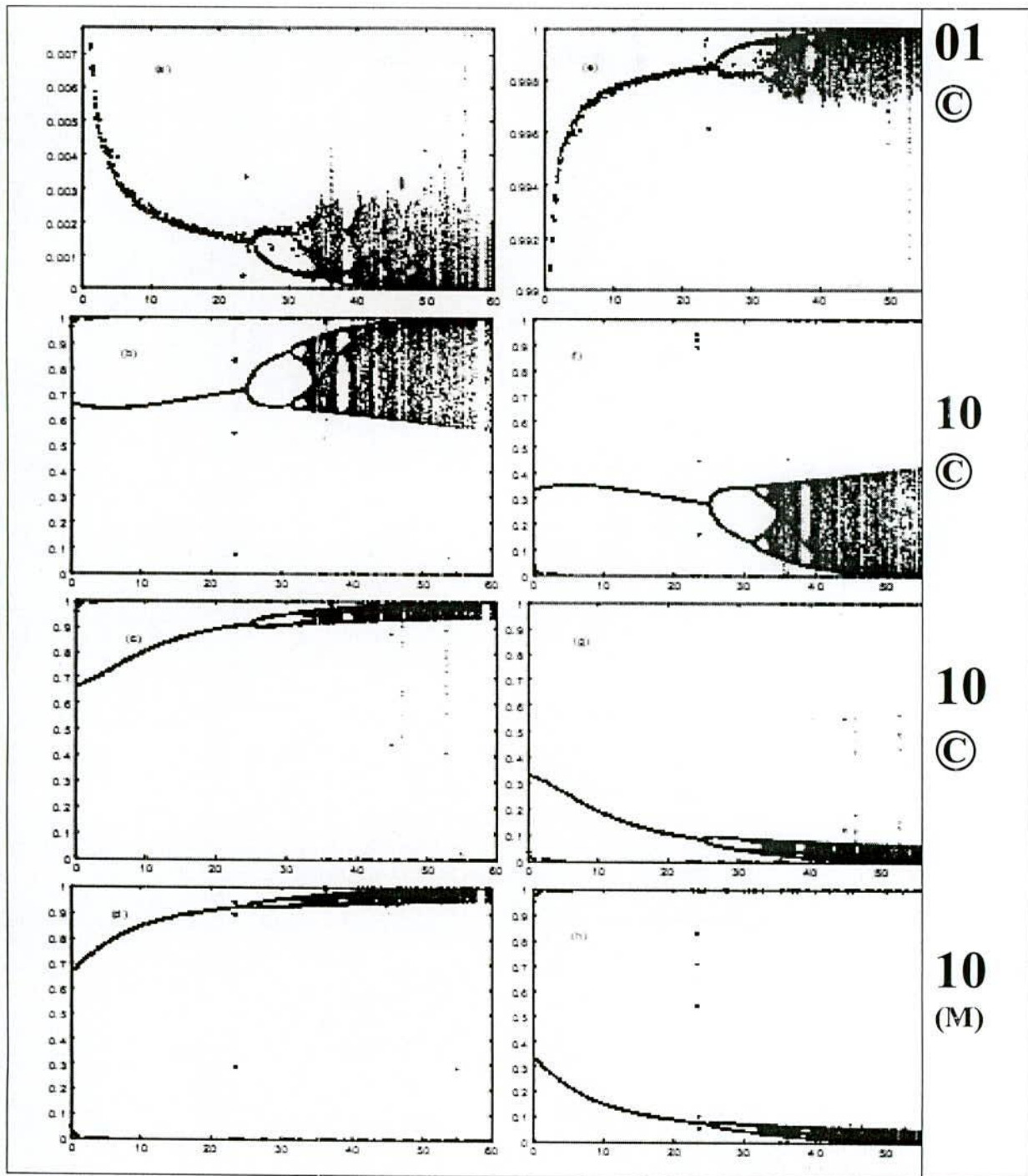


Fig. 3.3 Bifurcation diagram of output of 2-bit parity network. Figures of left column for output unit 1 and that of right column for output unit 2.

3.8.3 Validation with Lyapunov Exponents:

The maximum Lyapunov exponent must be positive for a time series to be chaotic. We have tested the time series for Lyapunov exponents and found that they are all positive for all the networks and outputs.

Table 3.2: Lyapunov exponents for the time series obtained from two-bit (XOR) network

| Network size | 2-2-2 | 2-3-2 | 2-4-2 |
|--------------|--------|--------|--------|
| First Output | 0.2118 | 0.0835 | 0.1630 |
| Unit | 0.7588 | 0.5973 | 0.7261 |
| | 0.6208 | 0.4817 | 0.5915 |
| | 1.1347 | 0.5922 | 0.9026 |

3.9 Other benchmark problems

Table 3.3 describes the characteristics of benchmark data sets. The cancer data set has 350 input patterns, 9 attributes and two classes. The diabetes data set contains 384 input patterns, 8 attributes and 2 classes.

Table 3.3
Characteristics of Classification Datasets.

| Problem | Training Examples | Attributes | Class |
|----------|-------------------|------------|-------|
| Cancer | 350 | 9 | 2 |
| Diabetes | 384 | 8 | 2 |

a) Cancer Problem:

In this case, the learning rate is fixed at 30.00. The outputs of the last 50 iterations are taken from 200 iterations in each run. The network is trialed 200 times. Hence the time series contains 10,000 data points. Since there is large number of training patterns (350), we have just investigated the two patterns – one from each class in order to study the characteristics of chaos for the problem. Each time a new network is taken and trained and the outputs of first unit at the output layer are stored to make the time series to be investigated. The HEs and FDs are computed and listed in Table 3.4 for both classes. It is seen that the NN structure has a significant effect on HEs. HE decreases with the increase of hidden nodes in the network as shown in Table 3.4 There may be one reason behind this.

We know that there is a relation between network size and over fitting. If the network is oversized it begins to overfit. It means that the network mixes up some randomness in their activations although it stays outside the classification zone.

Table 3.4
Hurst exponent and fractal dimension of cancer problem for first output unit.

| Class | Network 9-2-2 | | | Network 9-3-2 | | | Network 9-4-2 | | |
|---------|---------------|------|------|---------------|------|------|---------------|------|------|
| | Λ | H | FD | Λ | H | FD | Λ | H | FD |
| Class 1 | 1.3012 | 0.99 | 1.01 | 1.2418 | 0.88 | 1.12 | 0.6863 | 0.73 | 1.27 |
| Class 2 | 0.5316 | 1.00 | 1.00 | 0.6901 | 0.88 | 1.12 | 0.6745 | 0.73 | 1.27 |

Fig. 3.4 shows the bifurcation diagram for the first output unit activation. It is drawn for two examples- one from first class and one from other. Figure shows that classification falls in the chaotic regime.

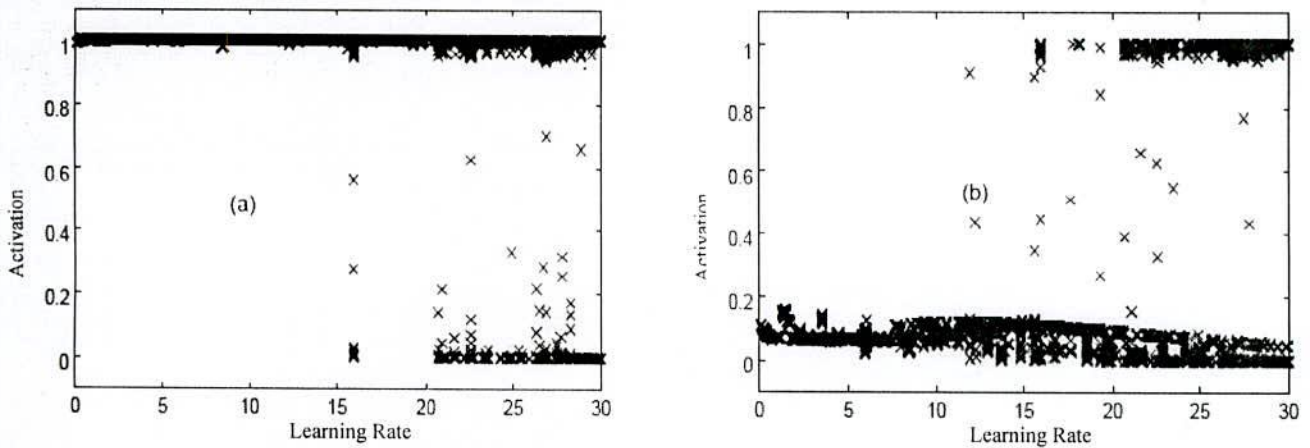


Fig: 3.4 Bifurcation diagram of output of cancer problem. a) First pattern from Class 1, and b) First pattern from Class 2.

We have shown return map for cancer problem in Fig. 3.5 Fig. 3.5(a), 3.5(c) and 3.5(e) indicate the time series of the outputs of the 9-2-2, 9-3-2 and 9-4-2 network at LR 30.00 for the class 1. Similarly Fig. 3.5(b), 3.5(d), and 3.5(f) indicate the time series of the 9-2-2, 9-3-2 and 9-4-2 network for class 1 at 30.00 LR. It is clear that the distribution of the output activations gradually decrease at the increase of network structure. This means the larger network has the ability to absorb the deviation of the points. The relation between the network structure and the complexity in chaotic regime will be discussed in Section

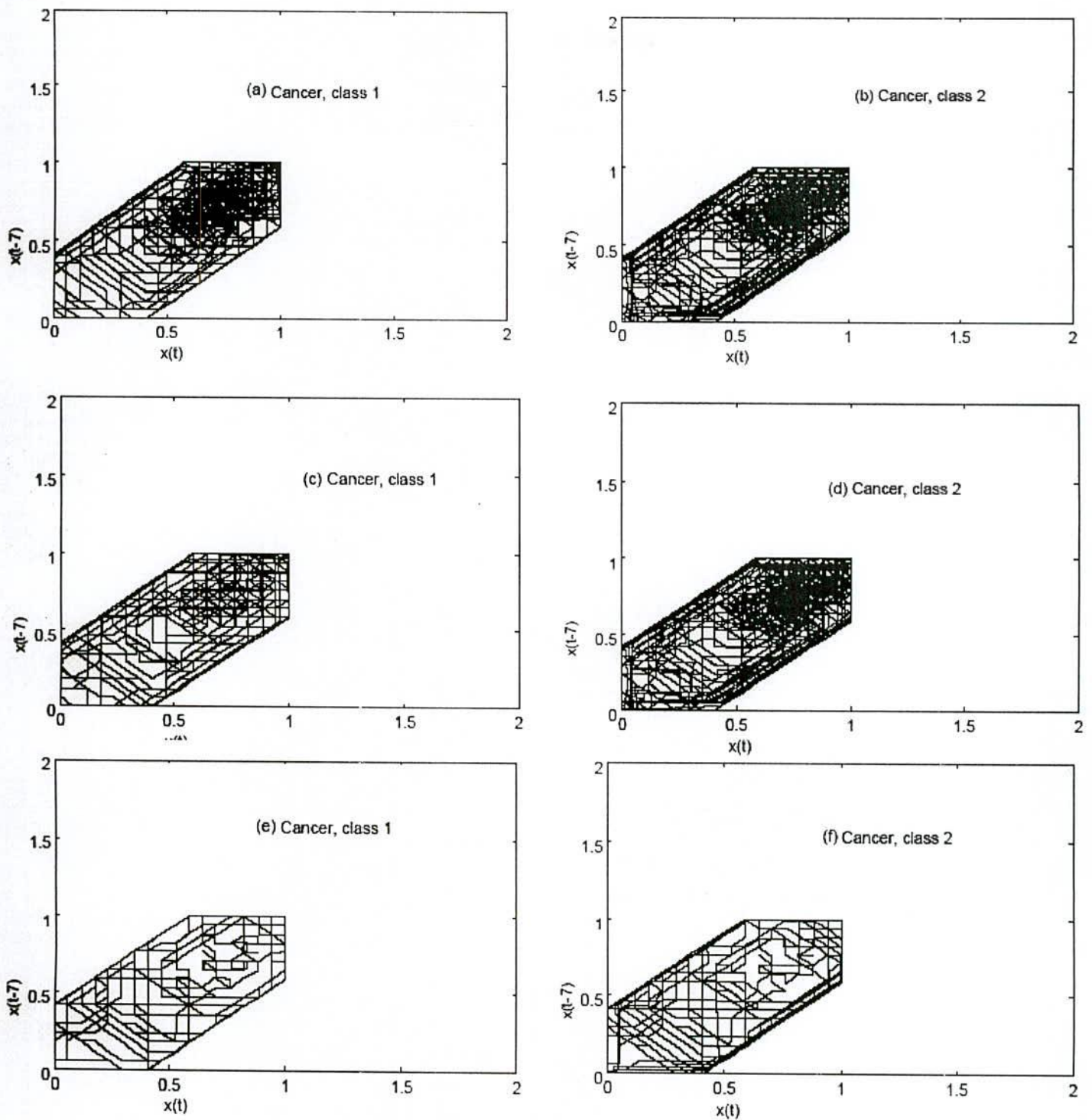


Fig. 3.5 Return map for cancer problem.

In chaos theory, control of chaos is based on the fact that any chaotic attractor contains an infinite number of unstable periodic orbits. Chaotic dynamics then consists of a motion where the system state moves in the neighborhood of one of these orbits for a while, then falls close

to a different unstable periodic orbit where it remains for a limited time, and so forth. This results in a complicated and unpredictable wandering over longer periods of time.

Control of chaos is the stabilization, by means of small system perturbations, of one of these unstable periodic orbits. The result is to render an otherwise chaotic motion more stable and predictable, which is often an advantage. The perturbation must be tiny, to avoid significant modification of the system's natural dynamics. In this paper we choose structure of the neural network to stabilize chaos some extend.

b) Diabetes Problem:

We have taken three NN structures (8-2-2, 8-3-2, and 8-4-2) to collect series of output activations. The data for the last 50 iterations are taken from 200 iterations in each run. The network is trialed 200 times. Hence the time series contains 10,000 data points. There are 384 training examples for diabetes problem. We generate the time series considering the first example from class 1 and first example from class 2. The HEs and FDs are listed in Table 3.5.

Table 3.5
Hurst exponent and fractal dimension of diabetes problem for first output unit.

| Class | Network 8-2-2 | | | Network 8-3-2 | | | Network 8-4-2 | | |
|---------|---------------|------|------|---------------|------|------|---------------|------|------|
| | Λ | H | FD | Λ | H | FD | λ | H | FD |
| Class 1 | 1.5479 | 1.00 | 1.00 | 0.9254 | 0.74 | 1.26 | 1.3369 | 0.65 | 1.35 |
| Class 2 | 1.5229 | 1.00 | 1.00 | 0.7994 | 0.74 | 1.26 | 1.0665 | 0.65 | 1.35 |

Bifurcation diagram is shown in Fig. 3.6. In this case, the actual output should lie around zero because the target output was zero.

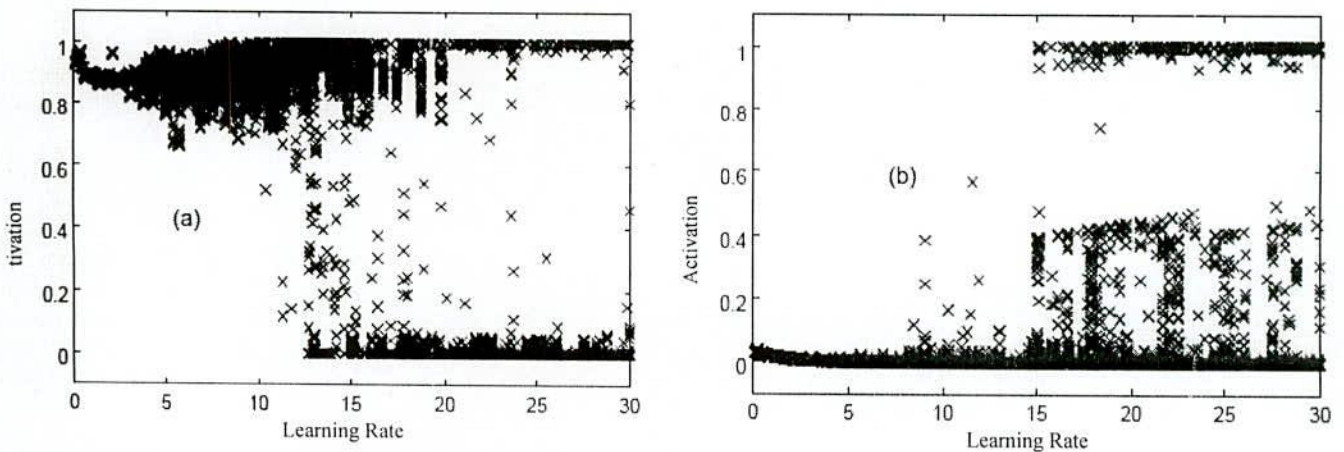


Fig: 3.6 Bifurcation diagram of output of diabetes problem. a) Class 1, first pattern, and b) Class 2, first pattern.

Chapter 4

Conclusion

This paper investigates the several chaotic behaviors of supervised neural networks. Lyapunov exponent (LE), Hurst Exponent (HE), fractal dimension (FD) and bifurcation diagram has been used to conjecture the findings. The chaotic dynamics of NNs for two-bit parity, cancer, and diabetes classification problems is investigated. It is found that HE can explain classification probability of NNs under chaos. The misclassification probability increases for pattern 00 to 11 for parity problem. The most difficult pattern is found to be 11, since the HE is most uncertain state at 0.5. It is found that the distribution of the network output is absorbed at the increase of size of the network. As a result chaosness is some extend marginally reduced. It is however interesting to see how much chaos is removed.

References

- [1] Skarda, C., & Freeman, W. J., "How brains make chaos in order to make sense of the world." *Behavioral and Brain Sciences*, 10, 161—195, 1987.
- [2] James Glanz, "Mastering the Nonlinear Brain," *Science*, Vol. 277. no. 5333, pp. 1758 – 1760.
- [3] H. Nozawa, "A neural-network model as a globally coupled map and applications based on chaos," *Chaos* vol. 2, no.3, pp. 377-386, 1992.
- [4] L. N. Chen and K. Aihara, "Chaotic simulated annealing by a neural network model with transient chaos," *Neural Netw.*, vol. 8, no. 6, pp. 915-930, 1995.
- [5] Peter Grassberger, Itamar Procaccia, "Estimation of the Kolmogorov entropy from a chaotic signal," *Phys. Rev. A*, 28, 2591 – 2593, 1983.
- [6] M. Sano and Y. Sawada, "Measurement of the Lyapunov Spectrum from a Chaotic Time Series," *Phys. Rev. Lett.* 55, 1082 – 1085, 1985.
- [7] J. -P. Eckmann, S. Oliffson Kamphorst, D. Ruelle, and S. Ciliberto, "Lyapunov exponents from time series," *Phys. Rev. A*, 34, 4971 – 4979, 1986.
- [8] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining Lyapunov exponents from a time series," *Physica D* 16, 285-317 (1985).
- [9] Rosenstein MT, Collins JJ and De Luca CJ., "A practical method for calculating largest Lyapunov exponents from small data sets," *Physica D*, 65: 117-134, 1993.
- [10] Peter Grassberger and Itamar Procaccia, "Characterization of Strange Attractors," *Phys. Rev. Lett.* 50, 346 - 349, 1983.
- [11] Mingzhou Ding, Celso Grebogi, Edward Ott, Tim Sauer, and J. A. Yorke, "Plateau onset for correlation dimension: When does it occur?," *Phys. Rev. Lett.* 70, 3872 – 3875, 1993.
- [12] Matthew B. Kennel, Reggie Brown, and Henry D. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Phys. Rev. A*, 45, 3403 – 3411, 1992.
- [13] Dechert and Gençay(1992), "An algorithm for the n Lyapunov exponents of an n-dimensional unknown dynamical system," *Physica D*, 59 (1992) 142-157.
- [14] M. Bask and R. Gençay, "Testing Chaotic Dynamics via Lyapunov Exponents," *Physica D*, 114 (1998) 1–2.
- [15] Floris Takens, "Detecting strange attractors in turbulence," , *Springer-Verlag*, Vol.898, Pages 366–381, 1981.
- [16] Fraser and Swinney, "Independent coordinates for strange attractors from mutual information," *Phys. Rev. A*, 33, 1134 - 1140 (1986).

- [17] J.-P. Eckmann, and D. Ruelle, "Addendum: Ergodic theory of chaos and strange attractors," *Rev. Mod. Phys.*, 57, 1115 - 1115 (1985)
- [18] H. Kantz and T. Schreiber. *Nonlinear Time Series Analysis*. Cambridge University Press, 1997.
- [19] James Theiler, "Spurious dimension from correlation algorithms applied to limited time-series data," *Phys. Rev. A*, 34, 2427 - 2432 (1986)
- [20] Celso Grebogi, Edward Ott, and James A. Yorke, "Unstable periodic orbits and the dimensions of multifractal chaotic attractors," *Phys. Rev. A*, 37, 1711 - 1724 (1988).
- [21] Simon Haykin, *Neural Networks, A Comprehensive Foundation*, PearsonPrentice Hall, Second Edition, 2001.
- [22] <http://www.learnartificialneuralnetworks.com>
- [23] McClelland, J. L., & Rumelhart, *Parallel distributed processing, Exploration in the microstructure of cognition*, Vol. 1&2, MIT Press.
- [24] K. Bertels, L. Neuberg, and S. Vassiliadis, "Xor and backpropagation: In and out of chaos?," *European Symposium on Artificial Neural networks*, 1995.
- [25] B. Derida and R. Meir, "Chaotic behavior of a layered neural network," *Physical Review A*, 38, 3116-3119, (1988).
- [26] K. Aihara, T. Takabe, and M. Toyoda, "Chaotic neural Network," *Physics Letters A*, 144, 333-340, 1990.