数理解析研究所講究録
第760巻 1991年 71-87

リカレントニューラルネットによる非線形力学系の学習

Learning Nonlinear Dynamics by Recurrent Neural Networks

ATR視聴覚機構研究所
ATR Auditory and Visual Perception Research Laboratories

佐藤雅昭　　村上由彦
Masa-aki Sato and Yoshihiko Murakami

任意のフィードバック結合を持つニューラルネット（リカレントネット）は、複雑な非線形ダイナミスを持つシステムであり、リミットサイクルやカオスなどの様々な時間的振る舞いを示す。我々はこれらの現象を情報処理に利用する目的で、リカレントネットを研究している。本稿では、非線形ダイナミクスの学習に関する研究を紹介する。

Learning Nonlinear Dynamics by Recurrent Neural Networks

Masa-aki Sato and Yoshihiko Murakami

ATR Auditory and Visual Perception Research Laboratories

Sanpeidan Inuidani Seika-cho Soraku-gun Kyoto, 619-02, Japan

ABSTRACT

A recurrent network, which can approximate a universal class of nonlinear dynamic systems, and its learning algorithm are presented. The possibility of learning chaotic dynamics by the recurrent network was investigated. The Lorentz attractor was used as an example of chaotic dynamics. When the trajectory of the Lorentz attractor was used as the teacher signal, the network was able to acquire the time evolution rule of the Lorentz dynamics and generated a chaotic attractor similar to the Lorentz attractor. The possibility of learning the hidden chaotic dynamics was also investigated.

1. INTRODUCTION

There are three types of neural networks. The first type is a multilayerd feed-forward network. It has been shown that a three-layer network can approximate any nonlinear function. The second type is a relaxation network, such as the Hopfield network. Although its output changes in time, only the stable output is used for information processing. Therefore, these two types of networks can be considered static information systems. The third type is a recurrent neural network with arbitrary feedback connections. Since the recurrent networks are complex nonlinear dynamic systems, they exhibit a variety of complex temporal behavior, such as limit cycle and chaos. Our main aim is to use the nonlinear behavior of a recurrent network for information processing [1,2]. This may open new areas for active and dynamic information processing.

In fact, chaos and other nonlinear phenomema have been found in many biological systems including squid giant axons, rat hippocampus, rabbit olfactory bulb and brain EEG[3,4,5]. These nonlinear dynamic phenomena seem to play an important role for information processing in biological systems [3]. We would like to control the chaotic dynamics by using recurrent networks. As a first step, we trained the recurrent network to learn the chaotic dynamics [1]. Although it is impossible to learn the long term behavior of chaotic dynamics because of the initial value sensitivity [6], it is possible to learn the time evolution rule of the chaotic dynamics.

Recently, Lapedes and Farber [7] trained feedforward backpropagation networks [8] to learn discrete chaotic maps, and studied the accuracy of the network's short time prediction. In our approach, on the other hand , the recurrent network can acquire the time evolution rule of the chaotic dynamics described by nonlinear differential equations.

In our previous paper [1], we proposed a new recurrent neural network architecture for general purposes. It is composed of two types of units. One is a dynamic unit whose output is determined by a differential equation. The other is a sigmoid unit which transforms an input to an output through a sigmoid function. These are connected each other by feedback connections. It is shown that this recurrent network can approximate a universal class of nonlinear dynamic systems if a sufficient number of hidden units is introduced. A supervised learning rule for this recurrent network was also derived. In this article, we summarized the previous results of the recurrent network architecture and the learning algorithm, and presented simulation results in detail.

In the computer simulation, the Lorentz attractor was used as an example of chaotic dynamics. The trained recurrent networks were composed of three dynamic units, which correspond to the three dynamic variables in the Lorentz dynamics, and thirty hidden sigmoid units. In one simulation, the trajectory of the Lorentz attractor was used as the teacher signal. After 30,000 weight updates, the recurrent network generated a chaotic attractor whose structure was very similar to that of the Lorentz attractor. The value of the largest Liapunov exponent calculated by the trained network was 0.85 (desired value: 0.90), which means that the recurrent network was able to learn the instability of the chaotic dynamics.

Next, we investigated the possibility of learning the hidden dynamic variables of the chaotic dynamics. When one variable was hidden, the trained network generated a chaotic attractor after 50,000 weight updates. The trayectories for visible variables were very chose to those of the Lorentz attractor, while the hidden variable trajectory was deviated from that of the Lorentz attractor. The implication of this result is also discussed.

2. UNIVERSAL APPROXIMATION FOR NONLINEAR DYNAMIC SYSTEMS

Most of nonlinear dynamic systems can be described by the following equations of motions if sufficient number of auxiliary variables are introduced:

$$dX(t)/dt = F(X(t), U(t)) \tag{2.1}$$

where X, U and F represent a N-dimensional vector dynamic variable, a K-dimensional vector external force and a N-dimensional vector nonlinear function which is called a vector field, respectively. For example, any Hamilton system can be written in this form.

Recently, it was shown that any nonlinear function can be approximated by a finite sum of sigmoid functions [9,10]. Let $G(x)$ be a sigmoid function. Let Ω be a compact region of a space spanned by X and U. The vector field $F(X,U)$ is assumed to be continuous in Ω. Then, for an arbitrary $\varepsilon > 0$, there exists an integer M and real constant's $WA_{im}, WB_{mi}, WC_{mk}, WD_m$(i=1,..,N; m=1,.., M;k=1,.., K)such that the following relation is hold:

$$\max_{(X,U)\in\Omega} |F_i(X,U) - H_i(X,U)| < \varepsilon \tag{2.2}$$

,where $H(X,U)$ is defined by

$$H_i(X,U) = \sum_{m=1}^{M} WA_{im} \cdot G(\sum_{j=1}^{N} WB_{mj} \cdot X_j + \sum_{k=1}^{K} WC_{mk} \cdot U_k + WD_m). \qquad (2.3)$$

If the nonlinear dynamic system defined by (2.1) is structurally stable [6], the vector field $F(X,U)$ can be approximated by the vector function $H(X,U)$ in the compact region Ω. Therefore, universal class of nonlinear dynamic systems described by the equation (2.1) can be approximated by recurrent neural networks defined by the following equations of motions:

$$dX(t)/dt = WA \bullet Z(t) \qquad (2.4a)$$

$$Z(t) = G(WB \bullet X(t) + WC \bullet U(t) + WD) \qquad (2.4b)$$

where the N-dimensional vector $X(t)$ and the M-dimensional vector $Z(t)$ represent outputs of dynamic units and sigmoid units, respectively. The dynamic units receive signals from the sigmoid units through the N X M connection weight matrix WA. The sigmoid units receive the M-dimensional vector bias WD, signals from the dynamic units through the M X N connection weight matrix WB and external inputs through the M X K connection weight matrix WC. They transform these inputs to outputs through a sigmoid function, G. In the learning process, some dynamic units receive desired temporal behavior as teacher signals. They are called visible units and denoted by VD. The other dynamic units have no teacher signal and are called hidden dynamic units. They are denoted by HD. The sigmoid units are all hidden since there is no teacher signal for them. The structure of the network is shown in fig.1.

3. LEARNING ALGORITHM

In this section, a supervised learning algorithm for the recurrent network defined by (2.4) is derived [1,13]. Although we can derive a learning rule for any error function, here we will use the teacher forcing error function [11,12]. In the teacher forcing method, the visible units are clamped to the teacher signal, $Q(t)$, by receiving additional external forces,

$$J_i(t) = dQ_i(t)/dt - (WA \bullet Z(t))_i \text{ for } i \in VD.$$

The magnitude of the external forces can be considered as the deviation from the desired network. Therefore, an error function is define by

$$E = \int_{t1}^{t2} dt \sum_{i \in VD} J_i^2(t). \tag{3.1}$$

By introducing the Lagrange multipliers, PX and PZ[13], the error function can be written as:

$$E = \int_{t1}^{t2} dt \Big[\sum_{i \in VD} J_i^2 - \sum_{i \in HD} PX_i (dX/dt - WA \cdot Z)_i - \sum_m PZ_m (Z_m - G((WB \cdot X + WC \cdot U + WD)_m)) \Big]. \tag{3.2}$$

Let us calculate the variation of the error function in order to get the expression for the gradient of the error function. The calculation is straightforward. The equations of motions for Lagrange multiplier can be derived from the requirement that the coefficient of the variations δX and δZ should be vanish:

$$d(PX_i)/dt = -\sum_m PZ_m \cdot G'((WB \cdot X + WC \cdot U + WD)_m) \cdot (WB)_{mi} \tag{3.3a}$$

and

$$PX_i(t2) = 0 \text{ for } i \in HD \tag{3.3b}$$

where $G'(x)$ represents the gradient of the sigmoid function, and

$$PZ_m = \sum_i P_i \cdot (WA)_{im} \qquad for \quad m = 1,...,M, \tag{3.3c}$$

where

$$P_i = -J_i \qquad for \quad i \in HD$$

and

$$P_i = PX_i \qquad for \quad i \in HD.$$

Then the variation of the error function can be written as

$$\delta E = \int_{t1}^{t2} dt \big[P^T \cdot \delta WA \cdot Z + \big(PZ \cdot G'(WB \cdot X + WC \cdot U + WD) \big)^T$$
$$\cdot (\delta WB \cdot X + \delta WC \cdot U + \delta WD) \ \big] + \sum_{i \in HD} PX_i(t1) \cdot \delta X_i(t1). \tag{3.4}$$

where matrix notations are used and the superscript T denotes the transpose of a vector. The derivatives of error function with respect to adjustable parameters WA, WB, WC, WD and $X(t1)$ are given by the coefficients of δWA, δWB, δWC, δWD and $\delta X(t1)$ in (3.4), respectively. The adjustable parameters can be modified by using the steepest descent method or other method like conjugate-gradient algorithm so that the error value will decrease.

The learning schedule is as follows [2]. First, the network is run forward in time from T to $(T + TB)$. The outputs of the hidden units are calculated by clamping the visible units to the teacher signals. Second, the error response variables, PX and PZ, are calculated backward in time from $(T + TB)$ to T, following equation(3.3). Then, the weight values are modified to decrease the error function. The initial value for hidden dynamic units are also updated. Finally, the recurrent network with new parameter values is run forward in time from T to $(T + TF)$, and the current time, T, is updated to $(T + TF)$. The above steps are repeated until the error value becomes sufficiently small. There are some comments on the initial condition in the above learning scheme. Although initial condition for the visible units are known, the initial condition for the hidden units are not known. An improper choice of the initial condition for the hidden units causes errors of the visible units even for the desired weight values. Therefore, the initial values for the hidden units are considered as learning parameters in our learning scheme. When the desired trajectory is chaotic motion, it is impossible to impose a initial condition at a fixed time because of sensitive dependence on the initial condition [6]. Therefore, the initial condition should be reset for each learning trial and the learning interval *TB* should not be large compared with the time scale corresponding to the largest Lyapunov exponent [6]. This means that the recurrent network learns different trajectories in the chaotic attractor for each learning trial. Since these trajectories are derived from the same time evolution rule, one can expect that the recurrent network is able to aquire the time evolution rule of the chaotic dynamics.

4. LEARNING CHAOTIC DYNAMICS

4.1 Lorentz Attractor

In this section, the possibility of learning chaotic dynamics by the recurrent network is investigated. The Lorentz attractor (fig.2) is used as an example of the chaotic dynamics. It is defined by the following differential equations [6].

$$dx/dt = F_1(x,y,z) = 10\cdot(x-y) \qquad (4.1a)$$
$$dy/dt = F_2(x,y,z) = -y + (28-z)\cdot x \qquad (4.1b)$$
$$dz/dt = F_3(x,y,z) = -(8/3)\cdot z + x\cdot y \qquad (4.1c)$$

This is an autonomous system and there is no external input.

The trained network was composed of three dynamic units and thirty sigmoid units. In the numerical simulation, these differential equations were approximated by the second order Runge-Kutta method. The time step was set to 0.01. The initial weights of the network were chosen randomly. In the learning phase, the learning internal *TB* and the free running time *TF* are set to 1.0 and 0.5, respectively.

4.2 All Visible Case

In one simulation, all the dynamic units received the teacher signals $x(t)$, $y(t)$ and $z(t)$ calculated by the equation (4.1). As learning proceeded, the network exhibited numerous bifurcations and the error increased at these points because of instability near the bifurcation points. Accordingly, we observed considerable qualitatively different behavior such as fixed points, limit cycles, etc (fig.3). After 30,000 weight updates, the recurrent network generated the chaotic attractor shown in fig.4. The structure of the attractor is very close to the Lorentz attractor. The accuracy of the approximation for the dynamic evolution rule (4.1) can be evaluated by the difference between the vector field F_i for the Lorentz dynamics (4.1) and the effective vector field $(WA{\cdot}Z)_i$ for the recurrent network (2.4). The error for the vector field $F_i(x, y, z)$ in a 2-D section of the phase space is shown in fig.7, where the average with respect to the remaining axis is taken. One can see that the error on the attractor is very small. The error inside the attractor is also small, while the error outside the attractor becomes large. One should note that the network has never been supplied the teacher signal in these regions. The error average over the attractor was 0.0002%. The above results show that the Lorentz dynamics (3.1) are well approximated by the recurrent network in the neighborhood of the attractor. We also calculated the largest Liapunov exponent [6] which characterize the degree of the instability of the chaotic dynamics. The value for the trained network was 0.85 (the value for the Lorentz attractor was 0.90). This indicated that the recurrent network was able to learn the instability of the chaotic trajectories in the Lorentz attractor.

4.3 Learning Hidden Dynamics

Next, we investigated the possibility of learning the hidden dynamic variables of the chaotic dynamics. Chaotic behavior does not appear for continuous dynamic systems with less than three degrees of freedom [6]. When only two dynamic variables, y and z, were used as the teacher signals, the recurrent network should estimate hidden dynamics in order to produce the chaotic attractor. However, there is an ambiguity corresponding to the coordinate transformation of the dynamic variables, since there is no teacher signal for x. Under the coordinate transformation,

$$x = h(x',y',z') \tag{4.2a}$$
$$y=y' \tag{4.2b}$$
$$z=z' \tag{4.2c}$$

,the trajectory of y and z do not change. Then, the hidden unit of the recurrent network could correspond to the transformed variable x'. The equations of motion for the transformed variables are given by

$$dx'/dt = \left[F_1(h(x',y',z'),y',z') - F_2(h(x',y',z'),y',z')\cdot \partial h/\partial y' - F_3(h(x',y',z'),y',z')\cdot \partial h/\partial z'\ \right]/(\partial h/\partial x') \tag{4.3a}$$
$$dy'/dt = F_2(h(x',y',z'),y',z') \tag{4.3b}$$
$$dz'/dt = F_3(h(x',y',z'),y',z') \tag{4.3c}$$

,and the trained recurrent network may aquire this time evolution rule. In this case, the vector field of the recurrent network is different from that of the Lorentz equation (4.1), although the dynamics of both systems are equivalent.

In the simulation, the recurrent network generated the chaotic attractor shown in fig.5 after the 50,000 weight updates. The trajectories for visible variables y and z are very close to that of the Lorentz attractor, while the hidden variable trajectories are deviated from those of the Lorentz attractor. The vector field errors corresponding to the visible variables are very small while that corresponding to the hidden variable is large (fig.8). The largest Liapunov exponent calculated by the trained network was 0.75.

The above results seem to indicate that the hidden unit of the trained network corresponds to the transformed variable x' in

(4.2). An attractor transformed from the Lorentz attractor by the coordinate transformation

$$x = x' - 2y' \quad (4.4a)$$
$$y = y' \quad (4.4b)$$
$$z = z' \quad (4.4c)$$

, is shown in fig.6. The attractor generated by the trained network (fig.5) is more similar to the transformed attractor (fig.6) than the Lorentz attractor (fig.2). However, we have not yet find the precise form of the transformation by which the trained recurrent network is mapped into the Lorentz attractor. There is another possibility that there exists different dynamics which generates the same trajectories for some of the variables of the Lorentz attractor. We are still investigating this problem.

5. Conclusion

A recurrent network, which can approximate a universal class of nonlinear dynamic systems, and its learning algorithm were presented. The possibility of learning chaotic dynamics was investigated. The Lorentz attractor was used as an example of the chaotic dynamics. When the trajectories of all the dynamic variables were used as the teacher signal, the recurrent network was able to acquire the time evolution rule of the Lorentz dynamics and generated a chaotic attractor which was very similar to the Lorentz attractor. The possibility of learning the hidden chaotic dynamics is still an open problem and we will study it further in our future publication. We hope recurrent networks and chaos may open a new area of active and dynamic information processing.

Reference

[1] M.Sato, Y.Murakami and K.Joe,"Learning chaotic dynamics by recurrent neural networks" Proc.Inter.Conf.on Fuzzy Logic &

Neural Networks, 601 (1990)
[2] M.Sato, K.Joe and T.Hirahara,"APOLONN brings us to the real world" IJCNN Vol.1, 581-587 (1990)
[3] C.A.Skarda and W.J.Freeman, "How brains make chaos in order to make sense of the world", Behavior and Brain Science, 10, 161-195 (1987)
[4] Proceeding of International Conference on Fuzzy Logic and Neural Networks, at IIZUKA, JAPAN(1990)
[5] H.G.Schuster, "Deterministic chaos", VCH (1988)
[6] J.Guckenheimer and P.Holmes, "Nonlinear oscillations, dynamical systems, and bifucations of vector field", Springer-Verlag, New York (1983)
[7] Lapedes A and Farber R, "Nonlinear signal processing using neural network" LA-UR-87-2662, Los Alamos National Lab. (1987)
[8] Hinton GE, Rumelhart DE and Williams RJ "Learning internal representation by error propagation", in Parallel Distributed Processing I, Rumelhart DE and McClelland JL, M.I.T. Press, Cambridge, MA, 318-362, (1986)
[9] Funahashi K, "On the approximate realization of continuous mapping by neural networks" Neural Networks, 2, 183-192 (1989)
[10] B. Irie and S. Miyake, "Capabilities of three-layered perceptrons" Proc.of IJCNN88, 1, 641-648 (1988)
[11] Pearlmutter BA, "Learning state space trajectories in recurrent network", Neural Computation, 1, 263-269 (1989)
[12] Williams RJ and Zipser D, "A learning algorithm for continually running fully recurrent neural network", Neural Computation, 1, 270-280 (1989)
[13] M. Sato, "A Learning Algorithm to Teach Spatiotemporal Patterns to Recurrent Neural Networks" Biol. Cybernetics, 62, 259-263 (1990)

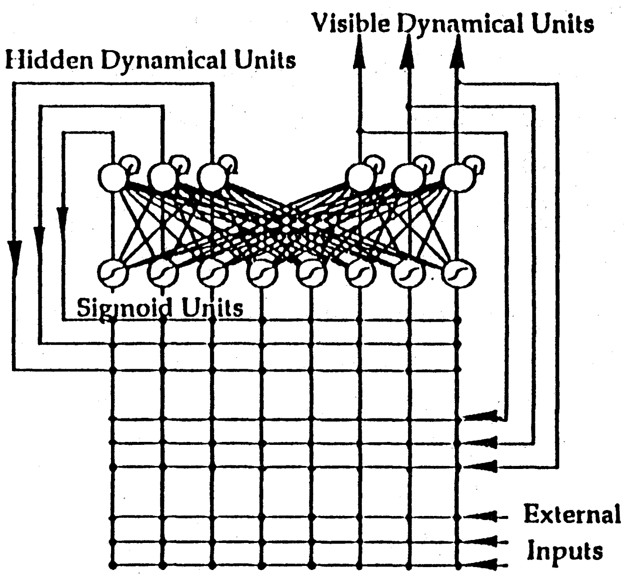


Fig.1 The structure of the recurrent network

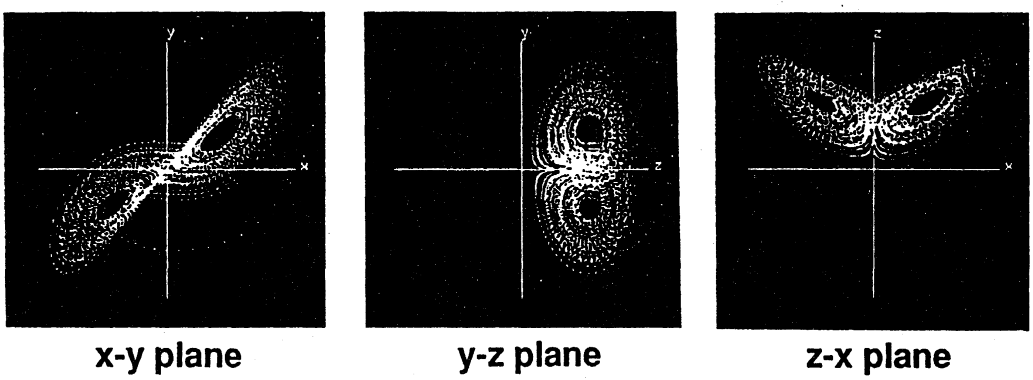


Fig.2 Lorentz attractor

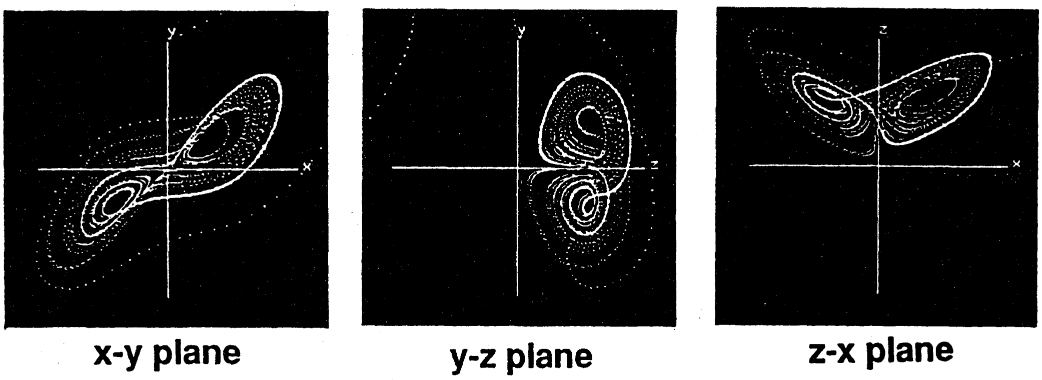
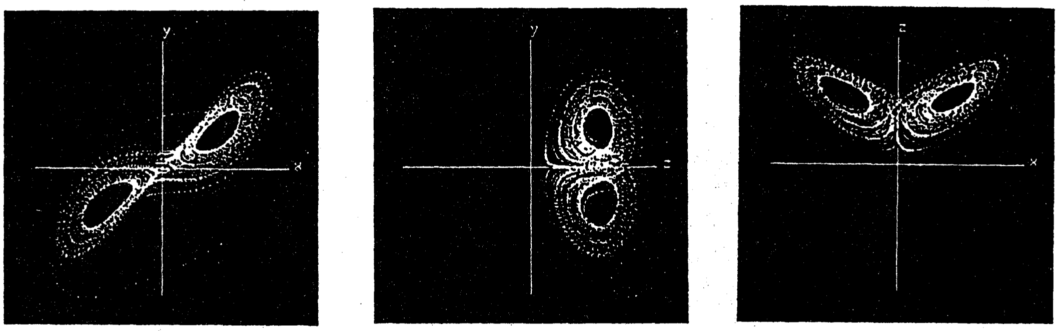
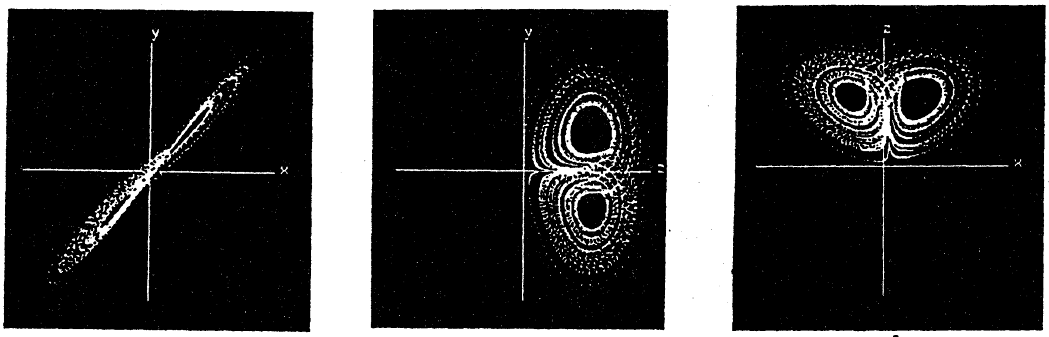


Fig.3 A limit cycle generated by the recurrent net under training



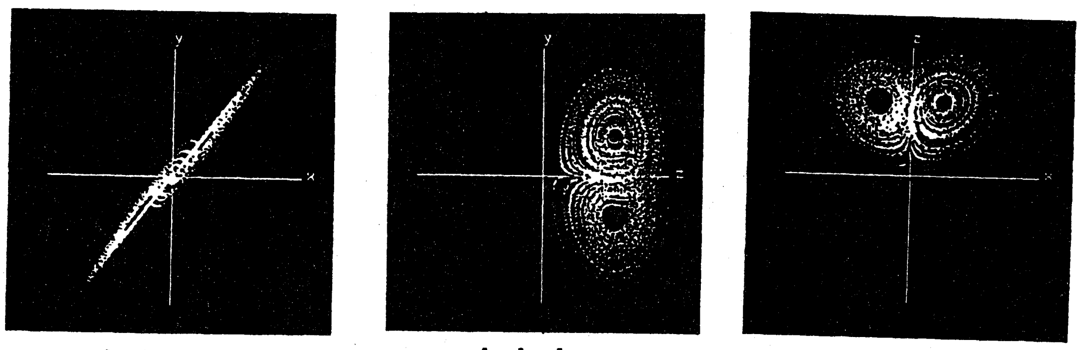
x-y plane y-z plane z-x plane

Fig.4 The attractor generated by the all visible recurrent net



x-y plane y-z plane z-x plane

Fig.5 The attractor generated by the one hidden recurrent net



x'-y' plane y'-z' plane z'-x' plane

Fig.6 Transformed Lorentz attractor

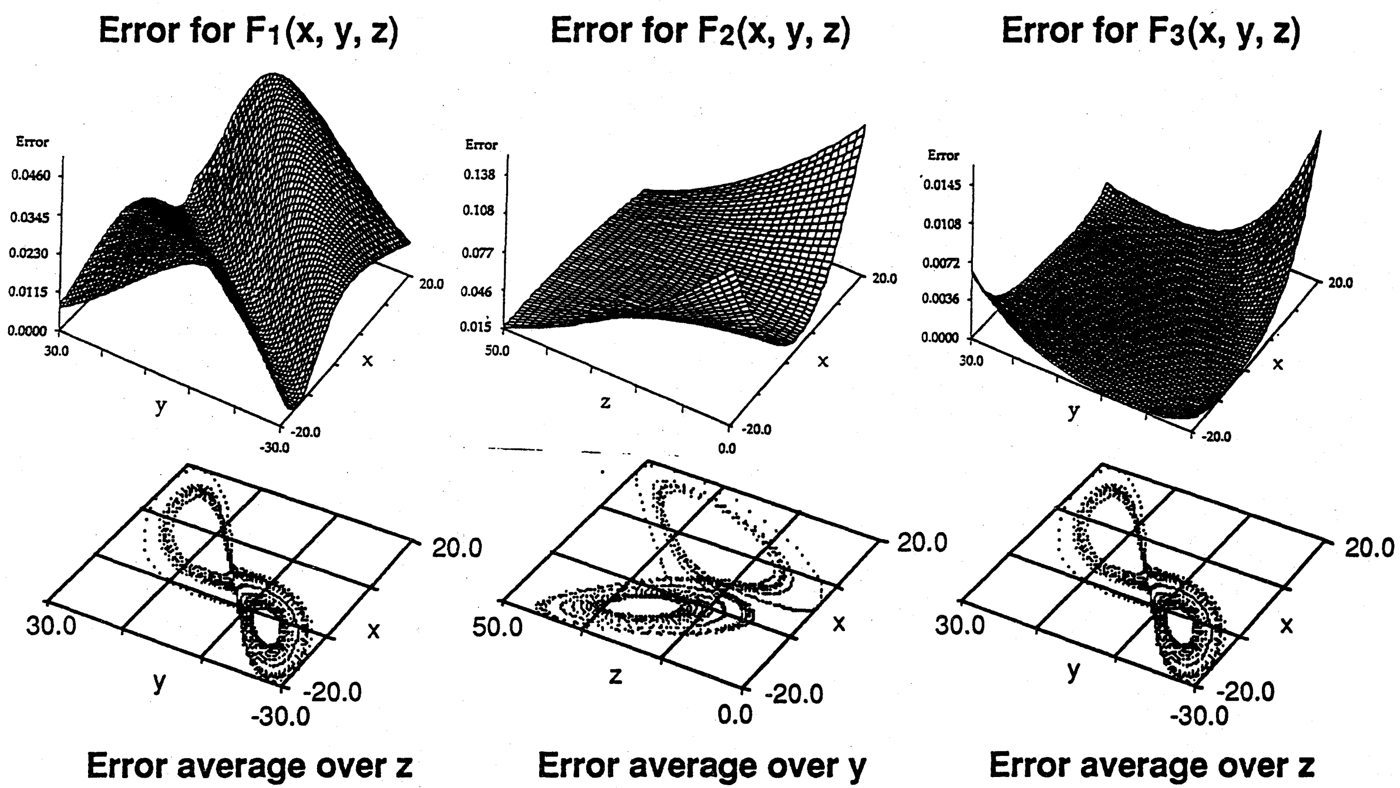


Fig.7 Error for the vector field F_1, F_2 and F_3 with all visible dynamic units

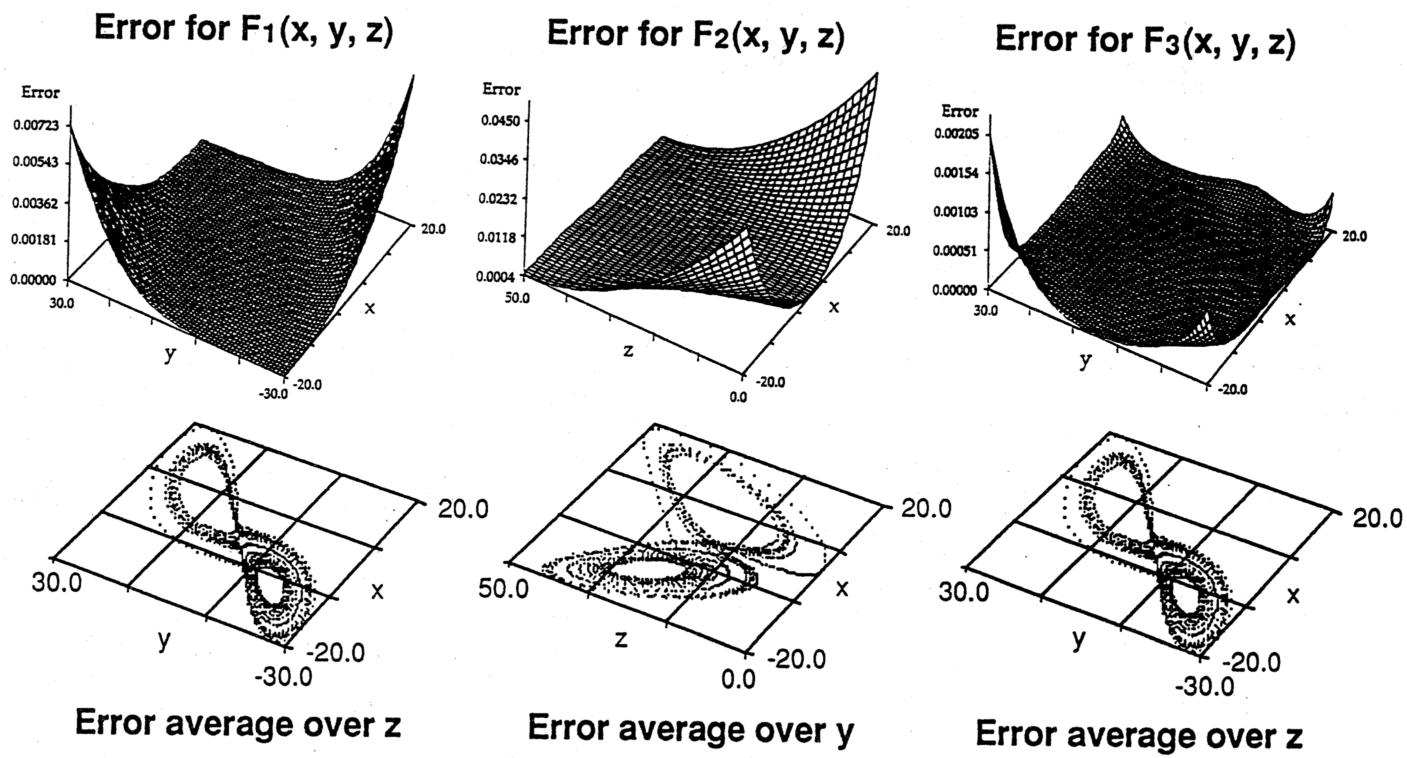


Fig.8 Error for the vector field F_1, F_2 and F_3 with one hidden dynamic unit