Learning Chaotic Dynamics in Recurrent RBF Network

T. Miyoshi, H. Ichihashi, S. Okamoto and T. Hayakawa

Department of Industrial Engineering
College of Engineering, University of Osaka Prefecture
1-1 Gakuen-cho, Sakai, Osaka 593, JAPAN
miyoshi@center.osakafu-u.ac.jp

ABSTRACT

Recurrent neural network with feedback and self-connection seems suited for temporal dynamics which expresses the input-output relation depending on time. Two learning procedures of the recurrent networks for computing gradients of the error function have been proposed in the literatures. One is to use sensitivity equations, the other is to use adjoint equations. We propose a recurrent RBF network and describe a procedure for finding the error gradients. We take advantage of the excellent function approximation capability of the RBF (Radial Basis Function) network.

1. Intoroduction

In recent years, there have been increasing research interests of artificial neural networks. Error backpropagation learning algorithm in feedforward neural networks models has its roots in nonlinear estimation and optimization. The radial basis function (RBF) network is a technique for interpolating in a high dimensional space and the training of RBF networks are potentially faster than the sigmoidal basis function networks with backpropergation for comparable error rate. "Gaussian Potential Functions"[7], "Localized Receptive Fields"[8],"Regularization networks"[11] and "Locally Tuned Processing Units"[9] are closely related paradigms which are collectively referred to as a three layered neural network or a simplified fuzzy reasoning[3, 4]. However, these feedforward neural networks can only deal with the static input-output relation. The recurrent neural network with feedback and self-connection seems suited for temporal dynamics which expresses the input-output relation depending on time.

The continuous time model is described by a differential equation and the discrete time model is described by a difference equation. Two learning procedures of recurrent networks for computing gradients of the error function have been proposed. One is to use sensitivity equations, the other is to use adjoint equations. We propose a recurrent RBF network and describe a procedure for finding the error gradient. We take advantage of the excellent function approximation capability of the RBF (Radial Basis Function) network.

Various approaches to learning in recurrent neural networks have been proposed. D.E.Rumelhart, G.E.Hinton and R.J.Williams[12] unfolded the recurrent network into multilayer feedforward network, which is called "backpropagation through time". R.J.Williams and D.Zisper[15] suggested to apply the parameter tuning technique in the general dynamical systems identification literature, which is called "real time recurrent network". K.Doya and S.Yoshizawa[2], B.A.Pearlmutter[10] and M.Satho[14] have shown how to train the temporally continuous recurrent network.

In this paper we apply these techniques of parameter optimization to the recurrent RBF networks for identification and prediction of chaotic dynamical systems. The organization of this paper is as follows: In Section 2, we review the learning algorithm for continuous time system and extension for RBF networks. Section 3 presents a simulation study on nonlinear dynamical system represented by the ordinary differential equation by O. E. Rössler[13]. In Section 4 we derive learning algorithm for discrete time chaotic system. Section 5 presents simulation study on the chaotic neural network by K.Aihara[1].

0-7803-2768-3/95/$4.00 © 1995 IEEE

2. Learning of Recurrent RBF Networks for Continuous Time System

In this section, we show how to train a recurrent RBF network of a temporally continuous system. We follow Kuroe's derivation[6] of the learning rule for the sigmoidal recurrent neural networks. Let $\boldsymbol{g} = \{g_1, g_2, \cdots, g_N\}$ be a vector of Gaussian RBFs

$$\begin{aligned} & g_r(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{a}, \boldsymbol{b}) \\ &= \sum_{k=1}^{K} \mu_k^r w_k^r \\ &= \sum_{k=1}^{K} \left\{ \prod_{j=1}^{N} \exp\left(-\frac{(x_j - a_{jk}^r)^2}{b_{jk}^r} \right) \right\} w_k^r \end{aligned} \quad (1)$$

where $\boldsymbol{x}$ denotes the state vector of the network, w_k^r, a_{jk}^r, b_{jk}^r $(j, r = 1, 2, \cdots, N, k = 1, 2, \cdots, K)$ are the unknown parameters. Let the network be governed by the system of ordinary differential equations as

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{a}, \boldsymbol{b}), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \quad (2)$$

Let the output of the network $\boldsymbol{y} \in R^M$ $(M < N)$ be a part of the elements of $\boldsymbol{x}$. The output $\boldsymbol{y}$ is written as

$$\boldsymbol{y} = I \cdot \boldsymbol{x} \quad (3)$$

$$I = \left(\begin{array}{cccc|c} 1 & & & O & \\ & 1 & & & O \\ & & \ddots & & \\ O & & & 1 & \end{array} \right), \quad (I \in R^{M \times N}) \quad (4)$$

We call this network represented by Eqs.(2) $\sim$ (4) "the recurrent RBF network". We assume that the time series of $\boldsymbol{y}$ is observed, and the objective is to minimize the deviation of $\boldsymbol{y}$ from $\boldsymbol{y}^*$ which is the target value of the time series. The cost function is defined as

$$\begin{aligned} E &= \int_{t_0}^{t_f} L(\boldsymbol{y})\, dt \\ &= \int_{t_0}^{t_f} \frac{1}{2} (\boldsymbol{y} - \boldsymbol{y}^*)^T \cdot (\boldsymbol{y} - \boldsymbol{y}^*) dt \end{aligned} \quad (5)$$

where T denotes transpose. The learning rule is written as

$$c_i^{NEW} = c_i^{OLD} - \eta \frac{\partial E}{\partial c_i} \quad (6)$$

where η is a positive learning rate. We denote the unknown parameters of the network $\boldsymbol{w}$, $\boldsymbol{a}$,$\boldsymbol{b}$ as $\boldsymbol{c}$ collectively, i.e., $c_i \in \{w_k^r, a_{jk}^r, b_{jk}^r \,|j, r = 1, 2, \cdots, N, k = 1, 2, \cdots, K\}$. Eq.(2) can be transformed into the following equation by differentiating both side of Eq.(2) with respect to parameter c_i.

$$\frac{\partial}{\partial c_i} \frac{d\boldsymbol{x}}{dt} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial c_i} + \frac{\partial \boldsymbol{g}}{\partial c_i}, \quad \frac{\partial \boldsymbol{x}(t_0)}{\partial c_i} = \boldsymbol{O} \quad (7)$$

where $\boldsymbol{O}$ denotes zero vector. Let G denote Jacobian matrix of $\boldsymbol{g}$ with respect to $\boldsymbol{x}$, then Eq.(7) is rewritten as the following sensitivity equation

$$\frac{d\boldsymbol{z}_i}{dt} = G \cdot \boldsymbol{z}_i + \boldsymbol{g}_{ci}, \quad \boldsymbol{z}_i(t_0) = \boldsymbol{O} \quad (8)$$

where $\boldsymbol{z}_i$ and $\boldsymbol{g}_{c_i}$ are defined as $\boldsymbol{z}_i = \partial \boldsymbol{x} / \partial c_i$, $\boldsymbol{g}_{c_i} = \partial \boldsymbol{g} / \partial c_i$ respectively, and we can obtain Eq.(9) by differentiating Eq.(5) with respect to c_i,

$$\frac{\partial E}{\partial c_i} = \int_{t_0}^{t_f} \left(\frac{\partial \boldsymbol{y}}{\partial c_i} \right)^T \cdot \frac{\partial L}{\partial \boldsymbol{y}} dt \quad (9)$$

Then we can calculate Eq.(9) by solving the sensitivity equations (8). It should be noted that $\partial \boldsymbol{y} / \partial c_i$ is a part of the vector $\boldsymbol{z}_i$.

Let $\boldsymbol{v}$ be the adjoint variables of $\boldsymbol{z}_i$, and calculate inner product with each term of Eq.(8), then we obtain the following equation.

$$\frac{d\boldsymbol{z}_i}{dt}^T \cdot \boldsymbol{v} = (G \cdot \boldsymbol{z}_i)^T \cdot \boldsymbol{v} + \boldsymbol{g}_{c_i}{}^T \cdot \boldsymbol{v} \quad (10)$$

Hence,

$$\begin{aligned} & \boldsymbol{z}_i^T \cdot \frac{d\boldsymbol{v}}{dt} + \boldsymbol{z}_i{}^T \cdot (G^T \cdot \boldsymbol{v}) \\ = \quad & -\boldsymbol{g}_{c_i}{}^T \cdot \boldsymbol{v} + \frac{d}{dt} (\boldsymbol{z}_i{}^T \cdot \boldsymbol{v}) \end{aligned} \quad (11)$$

By adding $(\partial \boldsymbol{y} / \partial c_i)^T \cdot (\partial L / \partial \boldsymbol{y})$ to both sides of Eq.(11), we obtain Eq.(12).

$$\begin{aligned} & \boldsymbol{z}_i{}^T \cdot \frac{d\boldsymbol{v}}{dt} + \boldsymbol{z}_i{}^T \cdot (G^T \cdot \boldsymbol{v}) + \frac{\partial \boldsymbol{y}}{\partial c_i}^T \cdot \frac{\partial L}{\partial \boldsymbol{y}} \\ &= \frac{\partial \boldsymbol{y}}{\partial c_i}^T \cdot \frac{\partial L}{\partial \boldsymbol{y}} - \boldsymbol{g}_{c_i}{}^T \cdot \boldsymbol{v} + \frac{d}{dt} (\boldsymbol{z}_i{}^T \cdot \boldsymbol{v}) \end{aligned} \quad (12)$$

Since $\partial \boldsymbol{y} / \partial c_i = I \cdot \boldsymbol{z}_i$, Eq.(12) is rewritten as

$$\begin{aligned} & \boldsymbol{z}_i{}^T \cdot \left(\frac{d\boldsymbol{v}}{dt} + G^T \cdot \boldsymbol{v} + I^T \cdot \frac{\partial L}{\partial \boldsymbol{y}} \right) \\ &= \frac{\partial \boldsymbol{y}}{\partial c_i}^T \cdot \frac{\partial L}{\partial \boldsymbol{y}} - \boldsymbol{g}_{c_i}{}^T \cdot \boldsymbol{v} + \frac{d}{dt} (\boldsymbol{z}_i{}^T \cdot \boldsymbol{v}) \end{aligned} \quad (13)$$

Let us assume that the following equation holds.

$$\frac{d\boldsymbol{v}}{dt} = -G^T \cdot \boldsymbol{v} - I^T \cdot \frac{\partial L}{\partial \boldsymbol{y}} \quad (14)$$

Then we obtain Eq.(15) from Eqs.(13) and (14)

$$\frac{\partial \boldsymbol{y}}{\partial c_i}^T \cdot \frac{\partial L}{\partial \boldsymbol{y}} = \boldsymbol{g}_{c_i}{}^T \cdot \boldsymbol{v} - \frac{d}{dt} (\boldsymbol{z}_i{}^T \cdot \boldsymbol{v}) \quad (15)$$

Letting $\boldsymbol{z}_i(t_0) = 0$, and $\boldsymbol{v}(t_f) = 0$, we obtain Eq.(16) from Eqs.(9) and (15)

$$\frac{\partial E}{\partial c_i} = \int_{t_0}^{t_f} \boldsymbol{g}_{c_i}{}^T \cdot \boldsymbol{v}\, dt \quad (16)$$

Let $\tau = t_0 + t_f - t$ then $d\tau / dt = -1$. We obtain the following adjoint equation

$$\frac{d\boldsymbol{v}}{d\tau} = G^T \cdot \boldsymbol{v} + I^T \cdot \frac{\partial L}{\partial \boldsymbol{y}}, \quad \boldsymbol{v}(t_0) = \boldsymbol{O} \quad (17)$$

We can obtain the gradient vector of Eq.(16) by solving Eq.(17) with the boundary condition of $\tau = t_0 (t = t_f)$, $\tau = t_f (t = t_0)$ and then change the value of unknown parameters which consist of the RBF networks by applying the learning rule of Eq.(6).

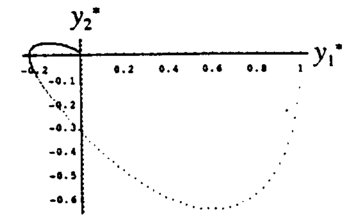
3. Numerical Experiment of Learning for Continuous Time Systems

We show the numerical experiments of learning for temporally continuous systems by using (I)sensitivity equations and (II)adjoint equations by the personal computer NEC PC-9801 BA (the Intel 486 DX2 (66MHz)). Eq.(18) is a system of a nonlinear ordinary differential equation of second order which has the property that the phase space changes with the initial values and is called separatrix.

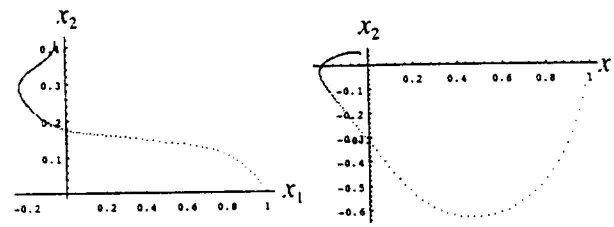
$$\begin{aligned} \frac{dx_1}{dt} &= x_2 \\ \frac{dx_2}{dt} &= -x_1 - 0.5x_2 - x_1^2 \end{aligned} \tag{18}$$

where the initial values of $x_1(0)$ and $x_2(0)$ are set to 1.0 and 0.0 respectively. Assuming that the time series x_1 and x_2 are observed at every 0.05 interval in the interval [0, 10], 200 data are obtained. And, let x_1 and x_2 equal the targets y_1^* and y_2^* respectively. We use the recurrent RBF network which consists of two RBF networks ($N = 2$) and each RBF network has 9 Gaussian basis functions ($K = 9$). The initial values of w_k^r are chosen randomly from the interval $[-0.5, 0.5]$, a_{jk}^k are uniformly spaced within the maximum value and the minimum value of the targets and b_{jk}^k are set to 0.05. Fourth order Runge-Kutta-Gill method is used in order to integrate numerically. Table 1 shows learning rates, the number of iterations, learning time and errors after learning (i)by the single target y_1^* and (ii)by the targets y_1^* and y_2^*, (I)by the sensitivity equations and (II)by the adjoint equations respectively.

Fig.1 shows (a)the trajectory of the targets (y_1^* and y_2^*) and (b)the trajectories of the output of the recurrent RBF networks after learning the time series from Eq.(18) using the sensitivity equations. The time series of targets is approximated by the series of x from the recurrent RBF network. Fig.2 shows the target time series y_1^* and y_2^* from Eq.(18) (training data) and Fig.3 shows the time series of the output of the recurrent RBF network after learning the time series from Eq.(18). Fig.4 shows generalizing capability when the initial values of (x_1, x_2) are changed to $(-1.0, 1.0)$. These results prove the generalizing capability of the recurrent RBF network. Since Eq.(18) is a stable system, the learning of the recurrent RBF network is very efficient.



(a)Trajectory of the targets from Eq.(18) (y_1^*, y_2^*) (training data).



(i)By the single target y_1^* (ii)By the targets y_1^* and y_2^*.
(b)Trajectories of the output of the recurrent RBF network after learning the time series from Eq.(18).

Fig. 1. Trajectories of the continuous time systems.

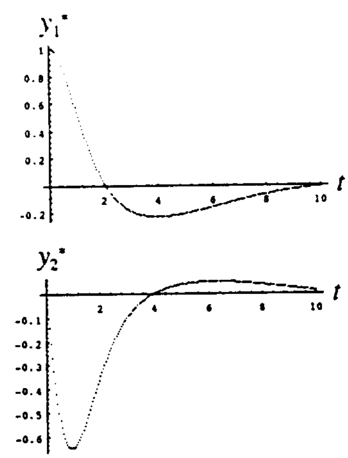


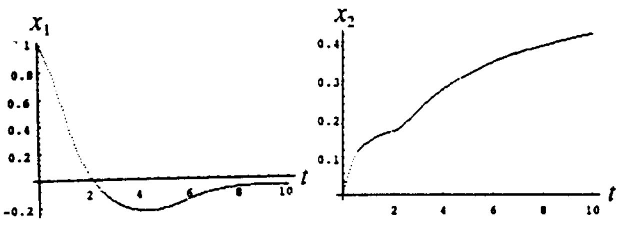
Fig. 2. Target time series y_1^* and y_2^* from Eq.(18) (training data).

$$\begin{aligned} \frac{dx_1}{dt} &= -x_2 - x_3 \\ \frac{dx_2}{dt} &= x_1 + 0.344\,x_2 \\ \frac{dx_3}{dt} &= 0.4\,x_1 - (4.5 - x_1)\,x_3 \end{aligned} \tag{19}$$

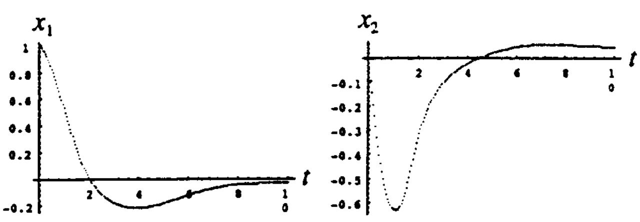
Eq.(19) is the ordinary differential equations by O. E. Rössler[13]. The solution of Eq.(19) is shown by a chaotic attractor in Fig.5(a) where the initial values of (x_1, x_2, x_3) are set to $(0.6, 1.0, 0.09)$. The time series x_1, x_2 and x_3 are observed by every 0.05 interval in the scope of [0, 25], and 500 data are obtained. Let x_1, x_2 and x_3 be the targets y_1^*,y_2^* and y_3^* respectively and normalized in the

Table 1. The results of learning by using the sensitivity equations and the adjoint equations. The time series of y^* is obtained from Eq.(18).

	(I)Sensitivity Equation		(II)Adjoint Equation	
	(i)y_1^*	(ii)y_1^*, y_2^*	(i)y_1^*	(ii)y_1^*, y_2^*
Learning rate λ	0.001	0.01	0.001	0.01
Cost function E	0.00213	0.00579	0.00221	0.00456
Ieration	1000	2000	1000	2000
Learning time(hour)	1.0	2.0	0.25	0.5

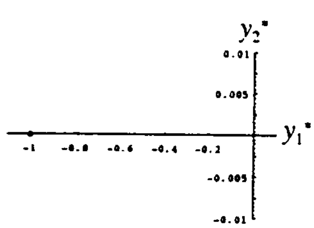


(i)By the single target y_1^*.

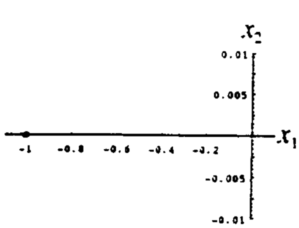


(ii)By the targets y_1^* and y_2^*.

Fig. 3. Time series of the output of the recurrent RBF network after learning the time series from Eq.(18).



(a)Target trajectory y_1^* and y_2^* from Eq.(18) (training data).



(b)Trajectory of the output of the recurrent RBF network after learning.

Fig. 4. Generalizing capability when the initial values are changed to $(x_1, x_2) = (-1.0, 1.0)$.

Table 2. The results of learning by using the adjoint equations. The time series of y^* is obtained from Eq.(19). (i)y_1^* (ii)y_1^*, y_2^* (iii)y_1^*, y_2^*, y_3^*.

	(i)	(ii)	(iii)
Learning rate λ	0.0005	0.0001	0.0005
Cost function E	0.01402	0.12361	0.04852
Ieration	30000	20000	20000
Learning time (hour)	90.0	60.0	60.0

interval $[-1.0, 1.0]$. We use the recurrent RBF network which consists of three RBF networks ($N = 3$) and each RBF network has 27 Gaussian basis functions ($K = 27$). The initial values of w_k^r are chosen randomly from an interval $[-0.5, 0.5]$, a_{jk}^r are uniformly spaced between the maximum and minimum values of the target and b_{jk}^r are set to 0.1.

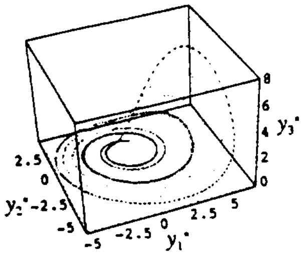
Table 2 shows learning rates, the number of iterations, learning time and errors after learning. We show the simulation results using adjoint equation in Fig. 5. Fig.5 shows (a)the trajectory of the targets (y_1^*,y_2^* and y_3^*), which forms the chaotic attractor by the Rössler equation and (b)the trajectories of the output of the recurrent RBF networks after learning by the time series from Eq.(19) (i)by the single target y_1^* (ii)by the targets y_1^* and y_2^* and (iii)by the targets y_1^*,y_2^* and y_3^*.

4. Learning of Recurrent RBF Network for Discrete Time Systems

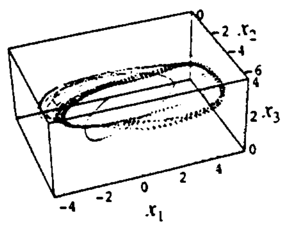
In this section, we consider the recurrent RBF network for discrete time system. Let g be a vector of Gaussian RBFs

$$\boldsymbol{x}[t+1] = \boldsymbol{g}(\boldsymbol{x}[t], \boldsymbol{w}, \boldsymbol{a}, \boldsymbol{b}), \qquad \boldsymbol{x}[t_0] = \boldsymbol{x}_0 \quad (20)$$

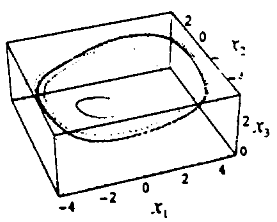
where $\boldsymbol{x}$ is the state vector of the network ($\boldsymbol{x} \in R^N, \boldsymbol{g} \in R^N$), and $\boldsymbol{w}$, $\boldsymbol{a}$, $\boldsymbol{b}$ are the unknown parameters of the model. Let the output of the network $\boldsymbol{y} \in R^M (M < N)$ be a part of the elements of $\boldsymbol{x}$ and we assume that the time series of the target $\boldsymbol{y}^*$ is observed. The output $\boldsymbol{y}$ of the RBF network is written as Eqs.(3) and (4). The cost function is



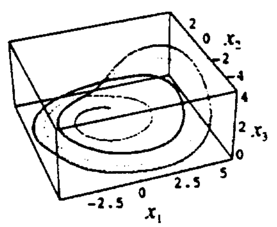
(a)Trajectory of the targets y_1^*, y_2^* and y_3^*, which show the chaotic attractor by the Rössler equation.



(i)By the single target y_1^*.



(ii)By the targets y_1^*, y_2^*.



(iii)By the targets y_1^*, y_2^* and y_3^*.
(b)Trajectories of the output of the recurrent RBF network after learning the time series from Eq.(19).

Fig. 5. Chaotic attractors of the continuous time systems

defined as

$$E = \sum_{t=t_0}^{t_f} L(\boldsymbol{y}) = \sum_{t=t_0}^{t_f} \frac{1}{2}(\boldsymbol{y}[t] - \boldsymbol{y}^*[t])^T \cdot (\boldsymbol{y}[t] - \boldsymbol{y}^*[t]) \quad (21)$$

where $\boldsymbol{y}^*$ is the target time series, and

$$\frac{\partial E}{\partial c_i} = \sum_{t=t_0}^{t_f} \frac{\partial \boldsymbol{y}}{\partial c_i}^T \cdot \frac{\partial L}{\partial \boldsymbol{y}} \quad (22)$$

The sensitivity equation for discrete time system is written as

$$\boldsymbol{z}[t+1] = G \cdot \boldsymbol{z}_i[t] + \boldsymbol{g}_{c_i}, \qquad \boldsymbol{z}[t_0] = \boldsymbol{O} \quad (23)$$

where $\boldsymbol{z}_i = \partial \boldsymbol{x} / \partial c_i$.

Let $\boldsymbol{v}$ be the adjoint variable of $\boldsymbol{z}_i$, and calculate inner product with each side of Eq.(23), then we obtain

$$\boldsymbol{z}_i[t+1]^T \cdot \boldsymbol{v}[\tau] = (G \cdot \boldsymbol{z}_i[t])^T \cdot \boldsymbol{v}[\tau] + \boldsymbol{g}_{c_i}^T \cdot \boldsymbol{v}[\tau] \quad (24)$$

By adding $\boldsymbol{z}_i[t]^T \cdot \boldsymbol{v}[\tau+1]$ to both sides of Eq.(24), we have

$$\begin{aligned} &\boldsymbol{z}_i[t]^T \cdot (\boldsymbol{v}[\tau+1] - G^T \cdot \boldsymbol{v}[\tau] - I^T \cdot \frac{\partial L}{\partial \boldsymbol{y}}) \\ &= -\frac{\partial \boldsymbol{y}}{\partial c_i}^T \cdot \frac{\partial L}{\partial \boldsymbol{y}} + \boldsymbol{g}_{c_i}^T \cdot \boldsymbol{v}[\tau] \\ &\quad -\boldsymbol{z}_i[t+1]^T \cdot \boldsymbol{v}[\tau] + \boldsymbol{z}_i[t]^T \cdot \boldsymbol{v}[\tau+1] \quad (25) \end{aligned}$$

Let

$$\boldsymbol{v}[\tau+1] = G^T \cdot \boldsymbol{v}[\tau] + I^T \cdot \frac{\partial L}{\partial \boldsymbol{y}} \quad (26)$$

where $\tau = t_0 + t_f - t$ and $\boldsymbol{v}[t_0] = \boldsymbol{O}$. Since $\boldsymbol{x}[t_0]$ is independent of the values of parameters $\boldsymbol{w}$, $\boldsymbol{a}$ and $\boldsymbol{b}$, we set $\boldsymbol{z}_i[t_0] = 0$. Then we have

$$\sum_{t=t_0}^{t_f} (-\boldsymbol{z}_i[t+1]^T \cdot \boldsymbol{v}[\tau] + \boldsymbol{z}_i[t]^T \cdot \boldsymbol{v}[\tau+1]) = \boldsymbol{O} \quad (27)$$

and

$$\sum_{t=t_0}^{t_f} \frac{\partial \boldsymbol{y}}{\partial c_i}^T \cdot \frac{\partial L}{\partial \boldsymbol{y}} = \sum_{t=t_0}^{t_f} \boldsymbol{g}_{c_i}^T \cdot \boldsymbol{v}[\tau] \quad (28)$$

We can obtain the gradient vector of Eq.(22) by determing $\boldsymbol{v}(\tau)$ ($\tau = t_0 + t_f - t$) from Eq.(26). Eq.(26) is called the error backpropagation rule.

5. Numerical Experiment of Learning for Discrete Time Systems

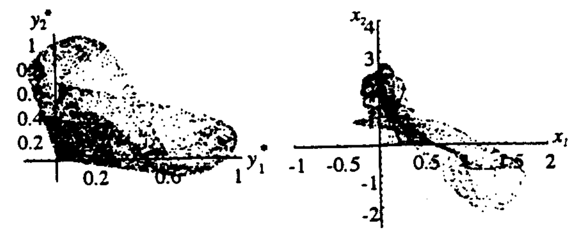
Let us now consider the chaotic neural network[1] by K.Aihara

$$x_i[t] = \left(1.0 + \exp\left(-\frac{y_i[t]}{\varepsilon}\right)\right)^{-1} \quad (29)$$

$$y_i[t+1] = k y_i[t] + \sum_j d_{ij} x_j[t] - \alpha x_i[t] + a_i \quad (30)$$

where $i, j \in \{1, 2\}$, $t = 0, 1, \cdots, 99$, $k = 0.8$, $d_{12} = d_{21} = 0.2$, $d_{11} = d_{22} = 0.0$, $a_1 = a_2 = 0.68$, $\varepsilon = 0.04$. The initial values of $y_1(0)$ and $y_2(0)$ are set to 0.5 and 0.0 respectively. Let y_1^* be the target and normalized in the interval $[-1.0, 1.0]$.

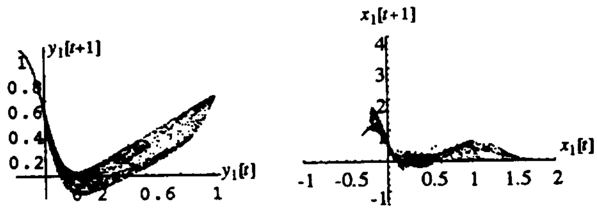
We use the recurrent RBF network which consists of two RBFs ($N = 2$) with 25 Gaussian basis functions ($K = 25$). The initial values of w_k^1 are randomly chosen from the interval [-0.25,0.25] and w_k^2 are set to 0.0, a_{jk}^r are uniformly spaced between the maximum and minimum target values, and b_{jk}^r are set to 0.3. In the numerical experiment, we substituted y_1^* for y_1 in order to speed up the learning, and the number of iterations is 20000.



(a) The Chaos neural network.

(b) The recurrent RBF network after learning.

Fig. 6. Strange attractors of the discrete time systems.



(a) The Chaos neural network.

(b) The recurrent RBF network after learning.

Fig. 7. Return maps of the discrete time systems.

Fig.6(a) shows the strange attractor of the CNN (Chaotic Neural Network) obtained from Eqs.(29) and (30), where $(y_1^*[t], y_1^*[t+1])$are plotted in the scope of $[0, 5000]$. Fig.6(b) shows the strange attractor of the output of the recurrent RBF network after learning, where $(x_1[t], x_2[t])$ are plotted in the same scope. Fig.7 shows the return map of CNN and the output of the recurrent RBF network after learning. Since the Lyapunov spectrum[5] is $(\lambda_1, \lambda_2) = (0.0027, 0.00083)$, the recurrent RBF network shows the orbital instability.

6. Conclusion

In this paper, we have proposed the recurrent RBF network for temporally continuous systems and discrete time systems. We have applied the learning rule using adjoint equations. By the excellent function approximation capability of the RBF networks, the chaotic behavior has been recovered.

7. References

[1] K.Aihara, T.Takabe and M.Toyoda, "Chaotic Neural Networks", Physics Letters A, Vol.144, No.6, pp.367-373 (1991)

[2] K.Doya and S.Yoshizawa, "Adaptive Neural Oscillator Using Continuous-Time Back-Propagation Learning", Neural Networks, Vol.2, No.5, pp.375-385 (1989)

[3] H.Ichihashi and I.B.Turksen, "A Neuro-Fuzzy Approach to Data Analysis of Pairwise Comparisons", Int. J. of Approximate Reasoning, Vol.9, No.3, pp.227-248 (1993)

[4] H.Ichihasi, "Hierarchical and Recurrent Networks of Fuzzy Models", Proc. of the 2nd Int. Conf. on Fuzzy Logic and Neural Networks, Iizuka, pp.19-22 (1992).

[5] T.Ikeguchi and K.Aihara, "An Analysis of Electroencephalographic Potentials by Deterministic Nonlinear Prediction", Proc. of the 2nd Int. Conf. on Fuzzy Logic and Neural Networks, Iizuka, pp.967-970 (1992).

[6] Y.Kuroe, "Learning Method of Recurrent Neural Networks", System, Control and Information, Vol.36, No.10, pp.634-643 (1992).

[7] S.Lee and R.M.Kil, "A Gaussian Potential Function Network with Hierarchically Self-organizing Learning", Neural Networks, Vol.4, No.2, pp.204-224 (1991).

[8] J.Moody and C.J.Darken, "Learning with Localised Receptive Fields", In:Eds: D.Touretzky, Hinton and Sejnowski, Proceedings of the 1988 Connectionist Models Summer School, Morgen Kaufmann Publishing, San Mateo, CA., pp.133-143 (1989)

[9] J.Moody and C.J.Darken, "Fast Learning in Networks of Locally-tuned Processing Uints", Neural Computation, Vol.1, No.2, pp.281-294 (1989).

[10] B.A.Pearlmutter, "Learning State Space Trajectories in Recurrent Neural Networks", Neural Computation, Vol.1, No.2, pp.263-269 (1989).

[11] T.Poggio and F.Girosi, "Regularization Algorithms for Learning that Are Equivalent to Multilayer Networks", Sciences, Vol.247, pp.987-982 (1990).

[12] D.E.Rumelhart, G.E.Hinton and R.J.Williams, "Learning Representations by Back-Propagation Errors", Nature, Vol.323, No.9, pp.533-536 (1986).

[13] O.E.Rössler, "Continuous Chaos", Annals New York Academic Scdi., Vol.31, pp.376-392 (1979).

[14] M.Sato, "A Learning Algorithm to Teach Spatiotemporal Patterns to Recurrent Neural Networks", Biological Cybernetics, Vol.62, No.32, pp.259-263 (1990).

[15] R.J.Williams and D.Zipser, "Experimental Analysis of the Real-Time Recurrent Learning Algorithms", Connection Science, Vol.1, pp.87-111 (1989)