Hindawi Publishing Corporation
Advances in Artificial Neural Systems
Volume 2009, Article ID 846040, 11 pages
doi:10.1155/2009/846040

*Research Article*

# Building Recurrent Neural Networks to Implement Multiple Attractor Dynamics Using the Gradient Descent Method

**Jun Namikawa and Jun Tani**

*Brain Science Institute, RIKEN, 2-1 Hirosawa, Wako City, Saitama 351-0198, Japan*

Correspondence should be addressed to Jun Namikawa, jnamika@bdc.brain.riken.jp

Received 31 March 2008; Accepted 22 August 2008

Recommended by Akira Imada

The present paper proposes a recurrent neural network model and learning algorithm that can acquire the ability to generate desired multiple sequences. The network model is a dynamical system in which the transition function is a contraction mapping, and the learning algorithm is based on the gradient descent method. We show a numerical simulation in which a recurrent neural network obtains a multiple periodic attractor consisting of five Lissajous curves, or a Van der Pol oscillator with twelve different parameters. The present analysis clarifies that the model contains many stable regions as attractors, and multiple time series can be embedded into these regions by using the present learning method.



## 1. Introduction

Recurrent neural networks (RNNs) have been successfully applied to the modeling of various types of dynamical systems. Since the universal approximation ability of multilayer neural networks has been proved, RNNs can model arbitrary dynamical systems and turing machines [1–3]. However, applying RNNs to a desired model may be very difficult even if such RNNs exist [4]. For example, building RNNs to implement required multiple attractor dynamics is a difficult problem for standard training, such as the gradient descent method. Doya and Yoshizawa [5] demonstrated that RNNs can acquire two limit cycles in the gradient descent method using initialization with small connection weights, whereas learning for more than three limit cycles is difficult [6]. This is due to the fact that the learning of several time series causes a conflict with respect to the changing of the connection weights. How to form RNN models that can learn several temporal sequence patterns has proved to be a challenging problem.

There have been some approaches to this problem. In order to avoid conflicts in the change of parameters, the mixture-of-experts-type architecture has been investigated [7, 8]. The mixture-of-experts model consists of RNNs as experts and a hierarchical gating mechanism. At the end of successful learning, each expert implements attractor dynamics as locally represented knowledge, and a gating mechanism chooses only one expert at any time. The system can acquire many attractor patterns although there is a disadvantage in that the system does not have the generalization ability on the attractor patterns. As the other approach to implement multiple patterns, the parametric bias (PB) method has been developed to improve the learning capability of RNNs [9, 10]. In an RNN that employs the PB method (RNNPB), PB values provide the information needed in order to individualize each sequence. It has been reported that the number of time series that RNNPBs can learn is greater than that which RNNs without PB can learn. However, the PB method cannot avoid the conflict caused by each attractor learning. Therefore, learning multiple time series by an RNNPB tends to fail when the number of time series increases.

In the present study, we will focus on the training method for RNNs to learn multiple attractor dynamics. Furthermore, we will show that the present research is related to research into RNNs with contraction transition functions. In recent years, RNNs with contraction transition mapping have been investigated with respect to the performance of time series learning [11–13], generalization

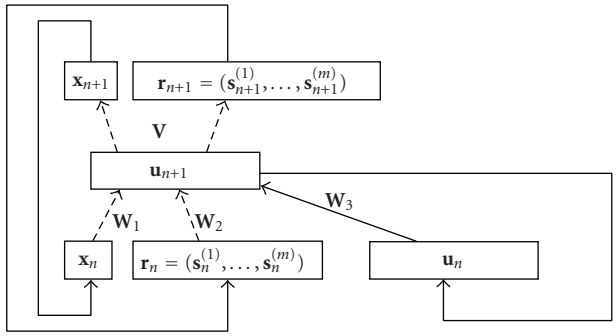



Figure 1: Architecture of the recurrent neural network. Solid arrows, dotted arrows, and boxes represent fixed connections, adjustable connections, and network states, respectively.

ability [14], and memory capacity [15]. Jaeger [11, 12] demonstrated that an "echo state network," which is an RNN with contraction mapping, successfully learns the Mackey-Glass chaotic time series, a well-known benchmark system for time series prediction. In order to formally express the generalization ability, Hammer and Tiňo proved that RNNs with contraction are distribution-independent learnable in the probably approximately correct (PAC) sense [14]. From the above results, RNNs with contraction might be regarded as powerful tools for modeling dynamical systems. However, RNNs with contraction have difficulty in representing multiple attractor dynamics because dynamic states governed by the contraction transition function are globally attracted to one point. In this paper, the representation capability of RNNs with contraction mapping will be improved such that the RNNs can obtain multiple attractor dynamics.

We start by defining the concepts of the RNN and the training method for multiple attractor dynamics. The RNN has the Elman net-type architecture, and the training method for RNNs is basically based on the backpropagation through-time (BPTT) algorithm [16]. We then show in numerical simulation that the RNNs can acquire multiple periodic attractors constituted by five Lissajous curves, or a Van der Pol oscillator with twelve different parameters. Moreover, we consider why the RNNs successfully learn multiple attractors and how the performance of learnability depends on parameters of the RNNs. Finally, we link the results obtained herein to other learning strategies, and consider other advanced research topics.

## 2. Model

*2.1. Recurrent Neural Network.* We first consider a neural network model with recurrent connection, such as the Elman net [17] (see Figure 1). The RNN contains I/O units, orthogonal units, and internal units. We denote the dynamic states of I/O units, orthogonal units, and internal units at time step $n$ by $\mathbf{x}_n \in \mathbb{R}^{N_1}$, $\mathbf{r}_n \in \mathbb{R}^{N_2}$, and $\mathbf{u}_n \in \mathbb{R}^{N_3}$, respectively. The RNN is defined by functions $f_\theta$ and $g_\theta$ with a parameter $\boldsymbol{\theta} \equiv (\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{V}, \mathbf{b}, \mathbf{d})$, where $f_\theta : \mathbb{R}^{N_1} \times \mathbb{R}^{N_2} \times \mathbb{R}^{N_3} \to \mathbb{R}^{N_3}$ and $g_\theta : \mathbb{R}^{N_3} \to \mathbb{R}^{N_1} \times \mathbb{R}^{N_2}$ are of the forms

$$f_\theta(\mathbf{x}, \mathbf{r}, \mathbf{u}) = (1-\epsilon)\mathbf{u} + \epsilon(\mathbf{W}_1 \cdot \mathbf{x} + \mathbf{W}_2 \cdot \mathbf{r} + \mathbf{W}_3 \cdot F(\mathbf{u}) + \mathbf{b}), \tag{1}$$

$$g_\theta(\mathbf{u}) = F(\mathbf{V} \cdot F(\mathbf{u}) + \mathbf{d}), \tag{2}$$

where $\mathbf{W}_1 \in \mathbb{R}^{N_1 \times N_3}$, $\mathbf{W}_2 \in \mathbb{R}^{N_2 \times N_3}$, $\mathbf{W}_3 \in \mathbb{R}^{N_3 \times N_3}$, and $\mathbf{V} \in \mathbb{R}^{N_3 \times N_1 N_2}$ are matrices, $b \in \mathbb{R}^{N_3}$ and $d \in \mathbb{R}^{N_1 N_2}$ are vectors, $\epsilon \in \mathbb{R}$ is a time constant that satisfies $0 \le \epsilon \le 1$, and $F$ denotes a componentwise application such as $F_i = \tanh$.

Dynamic states of the RNN at time step $n$ are updated according to

$$\begin{aligned} (\mathbf{x}_n, \mathbf{r}_n) &= g_\theta(\mathbf{u}_n), \\ \mathbf{u}_{n+1} &= f_\theta(\mathbf{x}_n, \mathbf{r}_n, \mathbf{u}_n) = f_\theta(g_\theta(\mathbf{u}_n), \mathbf{u}_n). \end{aligned} \tag{3}$$

From these equations, the RNN can be represented by an $N_3$-dimensional dynamical system.

We now define bistability for the RNN.

*Definition 1.* Assume $f_\theta : \mathbb{R}^{N_1} \times \mathbb{R}^{N_2} \times \mathbb{R}^{N_3} \to \mathbb{R}^{N_3}$ is as above. The function $f_\theta$ is *bistable* with respect to the third variable $\mathbf{u}$ if a real value $\omega > 1$ and an integer $N_s$ exist such that

$$\begin{aligned} w_{ij} &= \omega \quad \text{if } i \le N_s,\ i = j, \\ w_{ij} &= 0 \quad \text{if } i \le N_s,\ i \ne j, \\ |w_{ij}| &< \frac{1}{N_3} \quad \text{otherwise,} \end{aligned} \tag{4}$$

for every element $w_{ij}$ of the matrix $\mathbf{W}_3$.

The bistability of a function $f_\theta$ is a key concept of our learning method. We will show in Section 4.1 that the bistable function $f_\theta$ plays an important role in the learning of multiple attractor dynamics.

*2.2. Learning Method.* We present a formulation of the training procedure for the RNN with a multiple teacher I/O time series. For every $1 \le k \le m$ and $L_k \in \mathbb{N}$, we assume that $(\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_{L_k}^{(k)})$ is a sequence of teacher I/O of length $L_k$.

*Initialization of Parameters.* We initialize every element of matrices $\mathbf{W}_1(0)$, $\mathbf{W}_2(0)$, and $\mathbf{V}(0)$ and vectors $\mathbf{b}(0)$ and $\mathbf{d}(0)$ randomly from the uniform distribution in the interval $(-1/N_3, 1/N_3)$. A matrix $\mathbf{W}_3(0)$ is randomly assigned such that $f_\theta$ is bistable. For all $1 \le k \le m$, $\mathbf{u}_1^{(k)}(0)$ is randomly initialized in the interval $[-1, 1]$.

Assume that $\mathbf{r}_n^{(k)}(0)$ is an $m$-tuple of vectors $(\mathbf{s}_n^{(k,1)}(0), \ldots, \mathbf{s}_n^{(k,m)}(0))$ for $1 \le k \le m$ and $1 \le n \le L_k$, and that the dimension of $\mathbf{s}_n^{(k,l)}(0)$ is equivalent to that of $\mathbf{s}_{n'}^{(k',l')}(0)$ if $l = l'$. We initialize $\mathbf{s}_n^{(k,l)}(0)$ such that

$$\mathbf{s}_n^{(k,l)}(0) = \begin{cases} \mathbf{0} & \text{if } k = l, \\ -\mathbf{1} & \text{otherwise.} \end{cases} \tag{5}$$

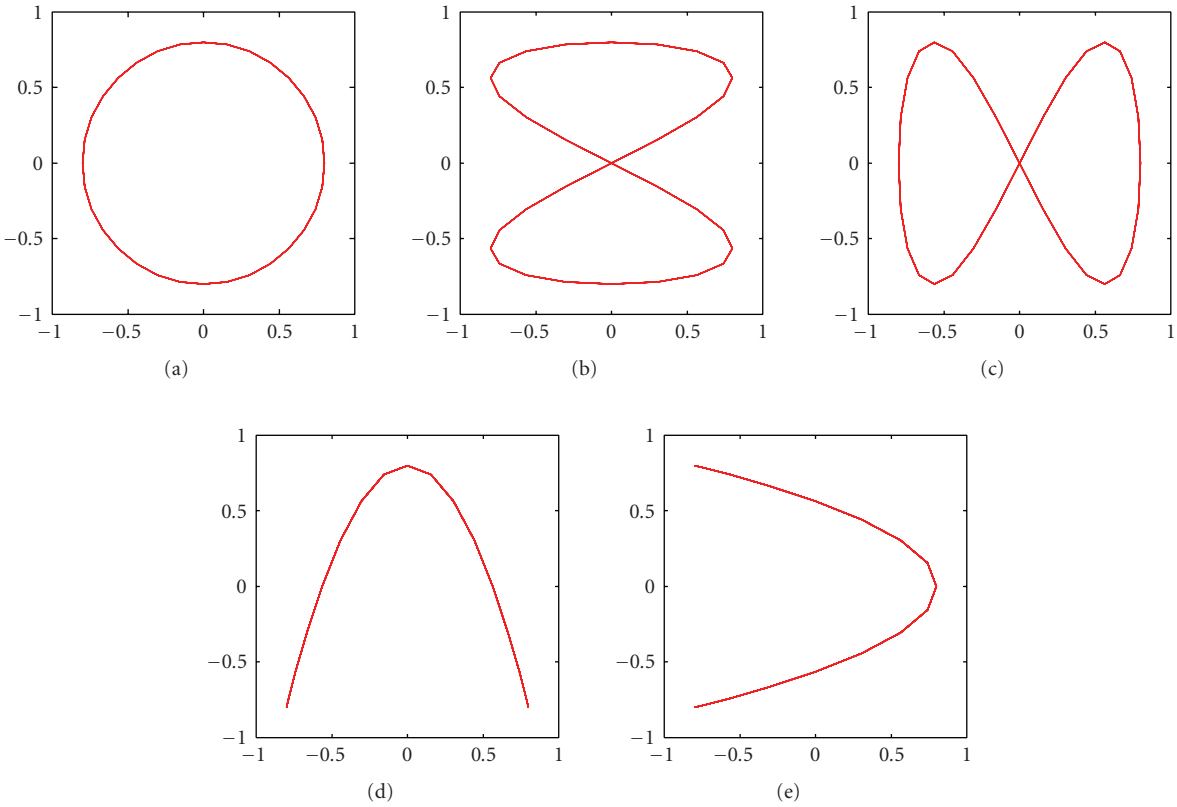



Figure 2: Trajectories of the teacher I/O time series in experiment 1.

*Run Network with Teacher I/O and Compute Error Function.* For every $1 \le k \le m$, the sequence $(\hat{\mathbf{x}}_1^{(k)}(t), \ldots, \hat{\mathbf{x}}_{L_k}^{(k)}(t))$ of I/O units of the RNN at learning step $t$ is defined by

$$\begin{aligned}(\hat{\mathbf{x}}_n^{(k)}(t), \hat{\mathbf{r}}_n^{(k)}(t)) &= g_{\boldsymbol{\theta}(t)}(\mathbf{u}_n^{(k)}(t)),\\ \mathbf{u}_{n+1}^{(k)}(t) &= f_{\boldsymbol{\theta}(t)}(\mathbf{x}_n^{(k)}, \mathbf{r}_n^{(k)}(t), \mathbf{u}_n^{(k)}(t)).\end{aligned} \tag{6}$$

The error function $E^{(k)}(t)$ of the RNN at learning step $t$ with the $k$th teacher I/O time series is defined by

$$E^{(k)}(t) = \sum_{n=1}^{L_k} e(\hat{\mathbf{x}}_n^{(k)}(t), \mathbf{x}_n^{(k)}) + e(\hat{\mathbf{r}}_n^{(k)}(t), \mathbf{r}_n^{(k)}(t)), \tag{7}$$

where $e$ denotes the mean square error function $e(\mathbf{x}, \mathbf{x}') = (1/2)(\mathbf{x} - \mathbf{x}')^{\mathrm{T}} \cdot (\mathbf{x} - \mathbf{x}')$. Finally, the error function $E(t)$ at learning step $t$ is defined by

$$E(t) = \sum_{k=1}^{m} E^{(k)}(t). \tag{8}$$

*Update Parameters.* Let $\boldsymbol{\rho}(t) \in \{\mathbf{W}_1(t), \mathbf{W}_2(t), \mathbf{V}(t), \mathbf{b}(t), \mathbf{d}(t)\}$ be a parameter of the RNN at learning step $t$. We determine the parameter $\boldsymbol{\rho}(t+1)$ by

$$\begin{aligned}\boldsymbol{\rho}(t+1) &= \boldsymbol{\rho}(t) + \alpha \Delta\boldsymbol{\rho}(t),\\ \Delta\boldsymbol{\rho}(t) &= \beta \Delta\boldsymbol{\rho}(t-1) - \frac{\partial E(t)}{\partial \boldsymbol{\rho}(t)},\end{aligned} \tag{9}$$

where $\Delta\boldsymbol{\rho}(0) = 0$; $\alpha$ and $\beta$ are the constants of the learning rate and momentum, respectively. On the other hand, a connection matrix $\mathbf{W}_3(t)$ is not changed as $\mathbf{W}_3(t+1) = \mathbf{W}_3(t)$ in order to hold the bistability condition. We compute the initial state $\mathbf{u}_1^{(k)}(t+1)$ of the internal units at learning step $t+1$ such that

$$\begin{aligned}\mathbf{u}_1^{(k)}(t+1) &= \mathbf{u}_1^{(k)}(t) + \alpha' \Delta\mathbf{u}_1^{(k)}(t),\\ \Delta\mathbf{u}_1^{(k)}(t) &= \beta \Delta\mathbf{u}_1^{(k)}(t-1) - \frac{\partial E(t)}{\partial \mathbf{u}_1^{(k)}(t)},\end{aligned} \tag{10}$$

where $\Delta\mathbf{u}_1^{(k)}(0) = 0$, and $\alpha'$ is the constant of the learning rate of the initial state. Assume that $\mathbf{s}_n^{(k,l)}(t)$ is a vector as a component of the orthogonal units $\mathbf{r}_n^{(k)}(t)$, such as $\mathbf{r}_n^{(k)}(t) = (\mathbf{s}_n^{(k,1)}(t), \ldots, \mathbf{s}_n^{(k,m)}(t))$. The vector $\mathbf{s}_n^{(k,l)}(t+1)$ is defined by

$$\mathbf{s}_n^{(k,l)}(t+1) = \begin{cases}\mathbf{s}_n^{(k,l)}(t) + \alpha'' \Delta\mathbf{s}_n^{(k,l)}(t) & \text{if } k = l,\\ -\mathbf{1} & \text{otherwise,}\end{cases} \tag{11}$$

$$\Delta\mathbf{s}_n^{(k,l)}(t) = \beta \Delta\mathbf{s}_n^{(k,l)}(t-1) - \frac{\partial E(t)}{\partial \mathbf{s}_n^{(k,l)}(t)}, \tag{12}$$

where $\Delta\mathbf{s}_n^{(k,l)}(0) = 0$, and $\alpha''$ is the constant of the learning rate of the orthogonal units.

Note that the maximum value of the error function $E(t)$ depends on the number of units and the length of the teacher

I/O time series. Thus, we should scale the learning rates $\alpha$, $\alpha'$, and $\alpha''$ with the number of units and length of sequences. In the present paper, we consider parameters $\gamma$, $\gamma'$, and $\gamma''$ such that $\alpha = \gamma/(N_1 + N_2)\sum_k L_k$, $\alpha' = \gamma'/(N_1 + N_2)$, and $\alpha'' = \gamma''/(N_1 + N_2)$.

## 3. Numerical Experiments

In this section, we conduct two types of experiments as examples of using the training method for RNNs proposed in Section 2. The first experiment shows the learning of five Lissajous curves. The second experiment shows the training of multiple attractors of a Van der Pol oscillator with 12 different parameters.

### 3.1. Experiment 1: Lissajous Curves

*3.1.1. Teacher I/O Time Series.* Our first task is to learn the five Lissajous curves defined by

$$\begin{aligned}
x_{n,1}^{(1)} &= \frac{4}{5}\sin\left(\frac{2\pi n}{M}\right), \qquad x_{n,2}^{(1)} = \frac{4}{5}\cos\left(\frac{2\pi n}{M}\right),\\
x_{n,1}^{(2)} &= \frac{4}{5}\sin\left(\frac{4\pi n}{M}\right), \qquad x_{n,2}^{(2)} = \frac{4}{5}\cos\left(\frac{2\pi n}{M}\right),\\
x_{n,1}^{(3)} &= x_{n,2}^{(2)}, \qquad\qquad\qquad x_{n,2}^{(3)} = x_{n,1}^{(2)},\\
x_{n,1}^{(4)} &= \frac{4}{5}\sin\left(\frac{2\pi n}{M}\right), \qquad x_{n,2}^{(4)} = \frac{4}{5}\cos\left(\frac{4\pi n}{M}\right),\\
x_{n,1}^{(5)} &= x_{n,2}^{(4)}, \qquad x_{n,2}^{(5)} = x_{n,1}^{(4)},
\end{aligned} \tag{13}$$

and we consider constants $M = 32$ and $L_k = 200$ for all $1 \le k \le 5$ (see Figure 2).

*3.1.2. Learning and Testing.* We now describe the specific conditions applied to RNN training. The time constant $\epsilon$ is set to 0.1. The number $N_2$ of orthogonal units is 10, and the dimension of a vector $\mathbf{s}_n^{(k,l)}$ is 2 for all $1 \le l \le 5$. Suppose that $f_{\boldsymbol{\theta}}$ is bistable with $N_3 = 30$, $N_s = 15$, and $\omega = 2.5$. The learning rates and momentum are given by $\gamma = 0.1$, $\gamma' = \gamma'' = 0.01$, and $\beta = 0.9$, respectively.

Figure 3 shows the error function $E^{(k)}(t)$ for 20 000 learning steps. We also show the Kullback-Leibler divergence between the teacher I/O time series and a sequence of I/O units in the RNN computed by (3) which do not use external perturbation by the teaching sequences. We use the Kullback-Leibler divergence as a measure of the discrepancy between two sequences. Formally, the Kullback-Leibler divergence between two probability distributions $p$ and $q$ is defined as

$$d_{\mathrm{KL}}(p,q) = \int (\log p(x) - \log q(x))\, p(x)dx. \tag{14}$$

By definition, in order to compute the Kullback-Leibler divergence, it is necessary to obtain probability distributions of the teacher I/O time series and a sequence of I/O units. However, obtaining the probability distribution of a sequence of I/O units is very difficult. Therefore, we quantize a time series of real-valued vectors into a symbolic sequence such that if the real value is less than 0, then the symbol 0 is appropriated, and otherwise the symbol 1 is appropriated. In addition, we use the probability distribution whereby sub-blocks with a block length of $l$ appear in the symbolic sequence given by the above quantization.

Figure 4 describes attractors of the trained RNN computed by (3) of which the initial state of internal units is $\mathbf{u}_1^{(k)}(t)$ for each $1 \le k \le 5$. By comparing the attractors with the teacher I/O time series displayed in Figure 2, we can see that the RNN can generate sequences similar to training data.

In Figure 5, examples of attractors for the RNN with random initial states are displayed. This shows that, in addition to the attractors corresponding to teacher I/O time series, there exist many attractors of the RNN.

### 3.2. Experiment 2: Van der Pol Attractors

*3.2.1. Teacher I/O Time Series.* Our second task is to learn multiple attractors given by the Van der Pol oscillator with different parameters. The Van der Pol oscillator defined by

$$\frac{d^2y}{dt^2} - \mu(1 - y^2)\frac{dy}{dt} + y = 0 \tag{15}$$

is a model of an electronic circuit that appeared in very early radios. It is well known that there exists a limit cycle for the Van der Pol oscillator. In this experiment, we consider twelve teacher I/O time series, where the $k$th teacher I/O time series $\mathbf{x}_n^{(k)}$ is given by

$$x_{n,1}^{(k)} = ay(t) + b_k, \quad x_{n,2}^{(k)} = a\frac{dy(t)}{dt} + c_k, \quad t = \frac{n}{\tau_k}, \tag{16}$$

for $\mu = 0.25$ and $a = 0.15$, where $b_k$ and $c_k$ are constant parameters representing the center position of the limit cycle, and $\tau_k$ is a time constant of the oscillator. We assume that the parameters $b_k$, $c_k$, and $\tau_k$ are given by combining the values of $b_k = \pm 0.4$, $c_k = \pm 0.4$, and $\tau_k = 2, 4, 6$. Figure 6 shows the teacher I/O time series given by (16). The length of training data is $L_k = 200$ for $1 \le k \le 12$.

*3.2.2. Learning and Testing.* The parameters for learning are set as follows. Let $f_{\boldsymbol{\theta}}$ be bistable with $N_3 = 40$ and $N_s = 20$. The dimension of the vector $\mathbf{s}_n^{(k,l)}$ is 1 for every $1 \le l \le 12$ so that $N_2 = 12$. Other parameters are the same as in experiment 1.

The error function and the Kullback-Leibler divergence for 200 000 learning steps are displayed in Figure 7. Figure 8 shows attractors of the trained RNN, and the initial state of the internal units of which is set to $\mathbf{u}_1^{(k)}(t)$ for every $1 \le k \le 12$.

This result allows us to consider that the RNN acquires multiple periodic attractors constituted by the teacher I/O time series.

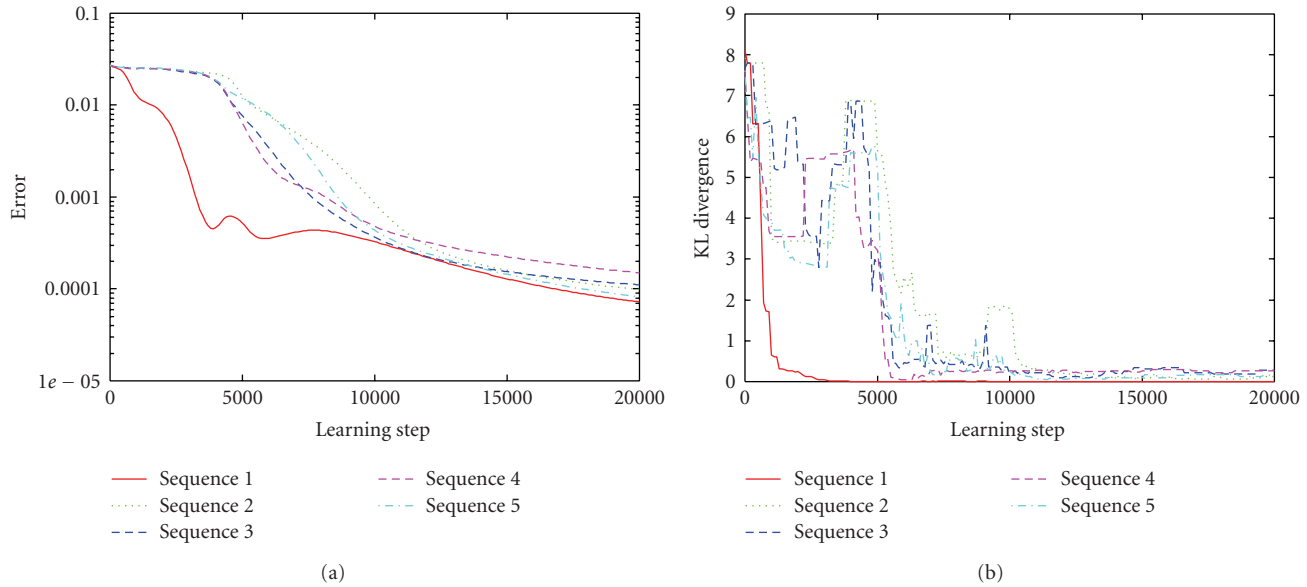



FIGURE 3: Error and Kullback-Leibler divergence between the teaching sequences and output generated by the RNN for 20 000 learning steps in experiment 1.

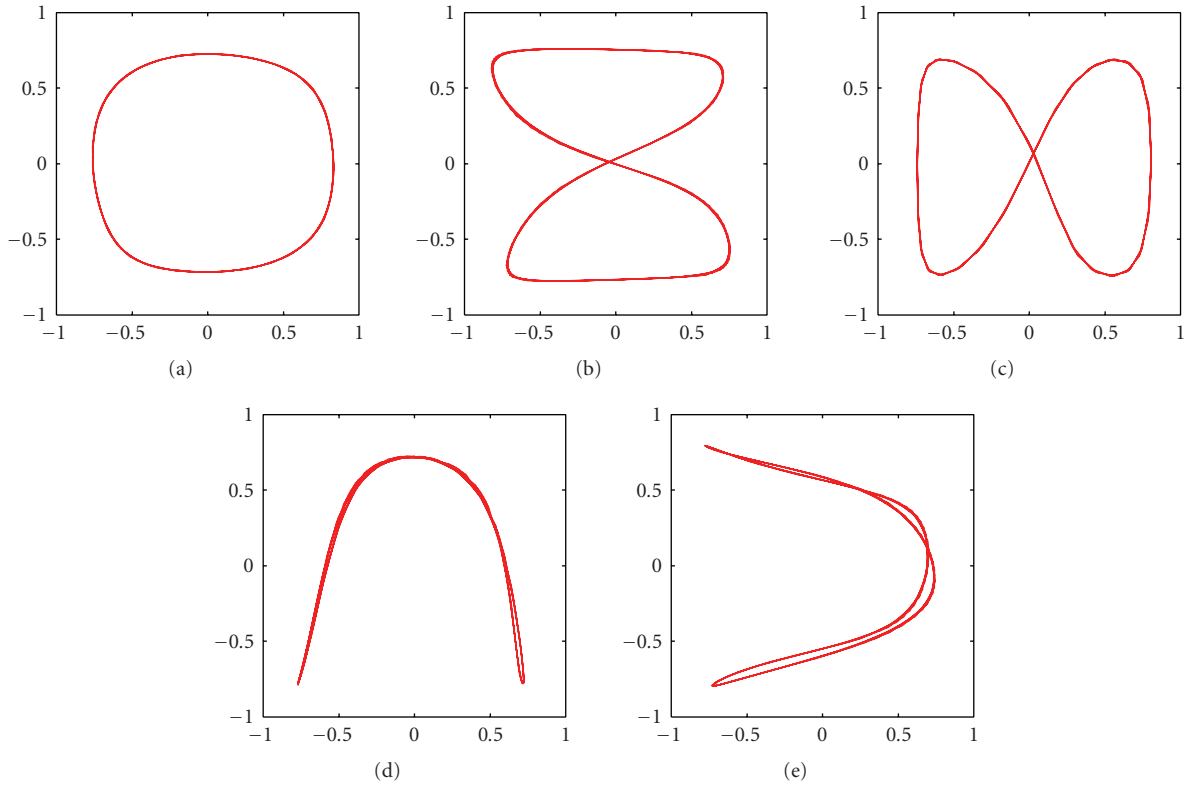



FIGURE 4: Time series $\mathbf{x}_n$ generated by the trained RNN in experiment 1. For each time series, only the initial state $\mathbf{u}_0$ is different.

## 4. Numerical Analysis

*4.1. Contraction and Bistability.* Assume that $X$ and $U$ are sets and that $U$ is equipped with a metric structure. A function $f : X \times U \to U$ is a contraction with respect to $U$ if a real value $C \in [0, 1)$ exists such that the inequality

$$\forall u_1, u_2 \in U \quad \forall x \in X \, | f(x, u_1) - f(x, u_2) | \leq C | u_1 - u_2 | \tag{17}$$

holds for all $x \in X$ and $u_1, u_2 \in U$.

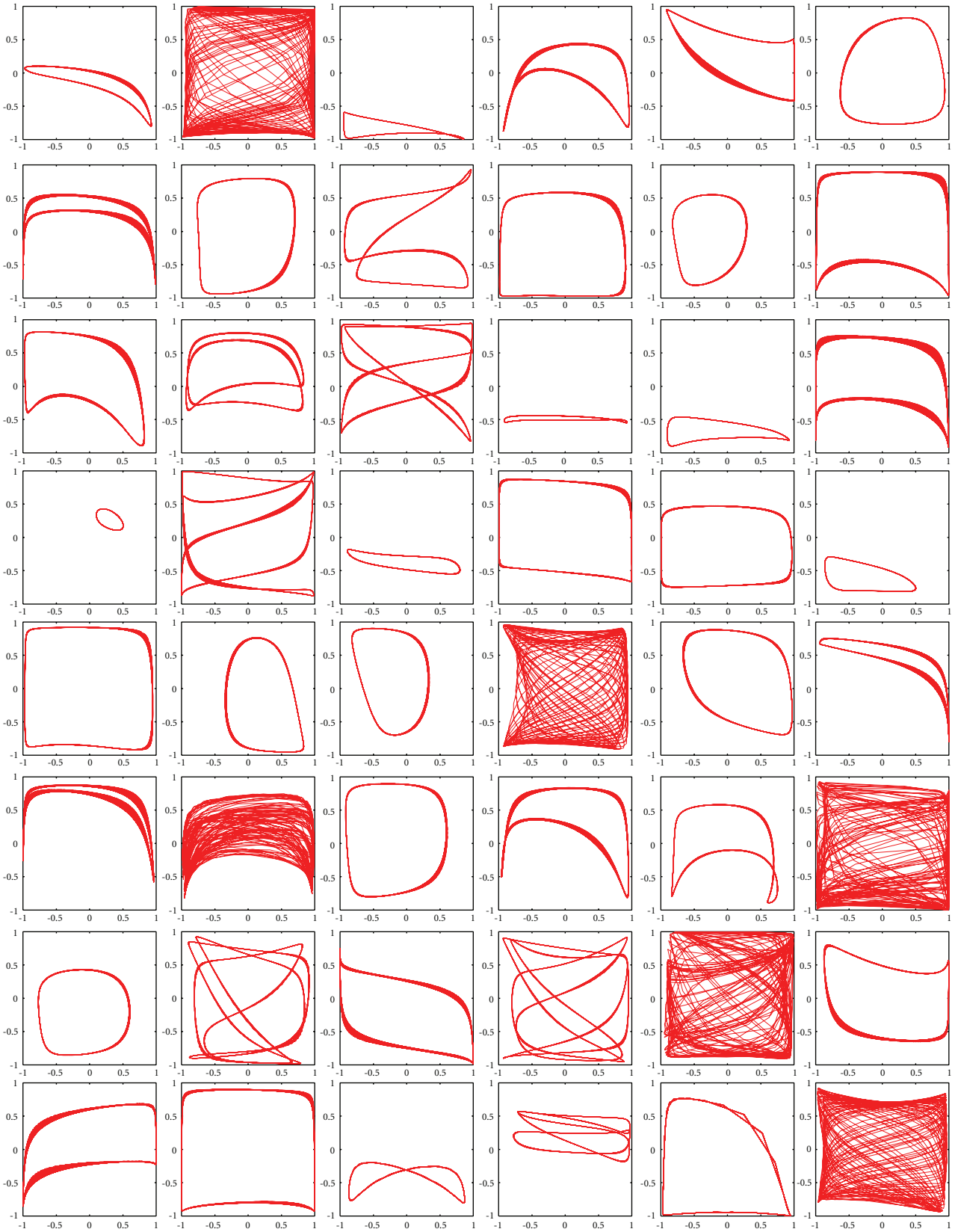


Figure 5: Time series $\mathbf{x}_n$ generated by the trained RNN with random initial state $\mathbf{u}_0$ in experiment 1.

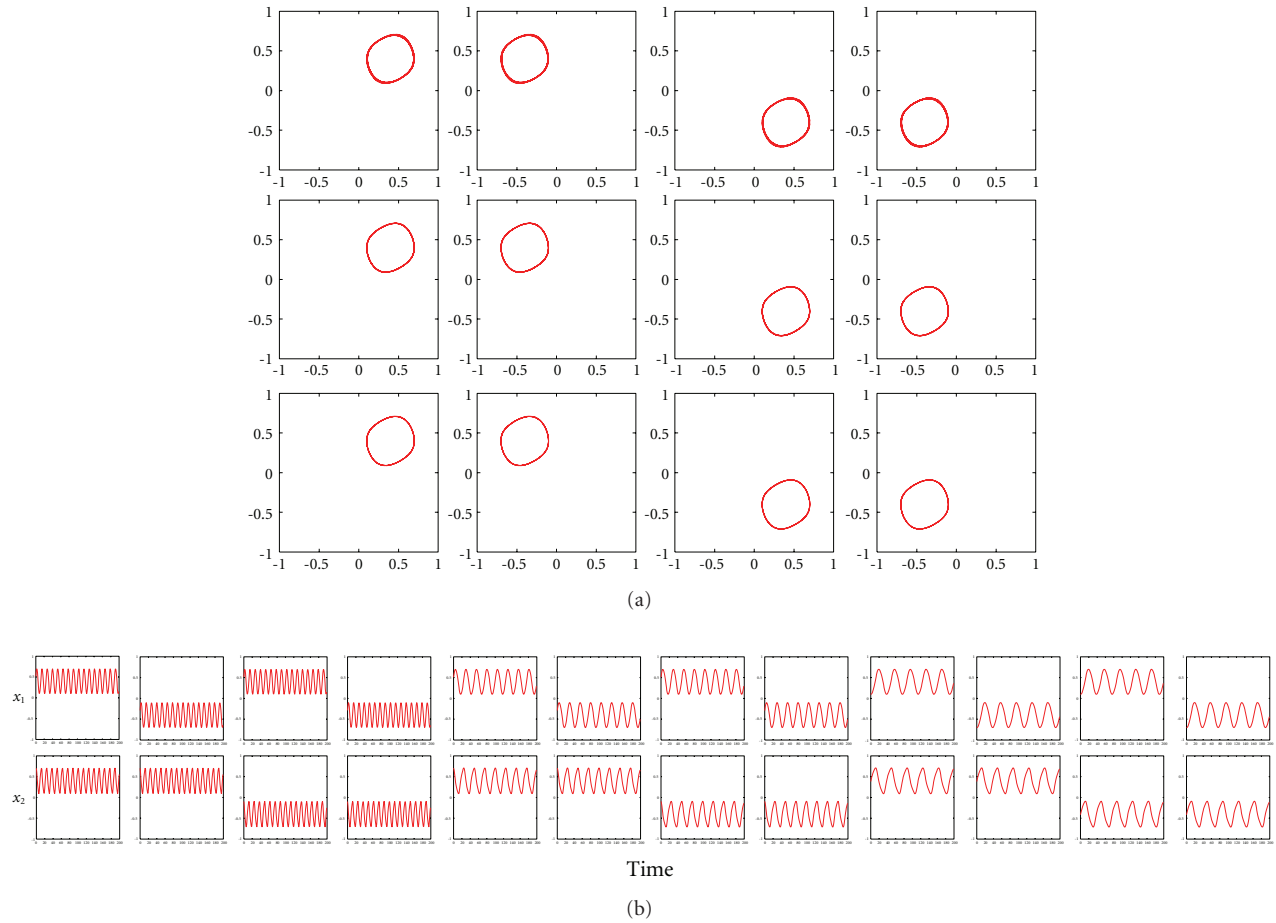



Figure 6: Teaching sequences of experiment 2. (a) Trajectories on $\mathbb{R}^2$. (b) Temporal trajectories of teaching sequences.

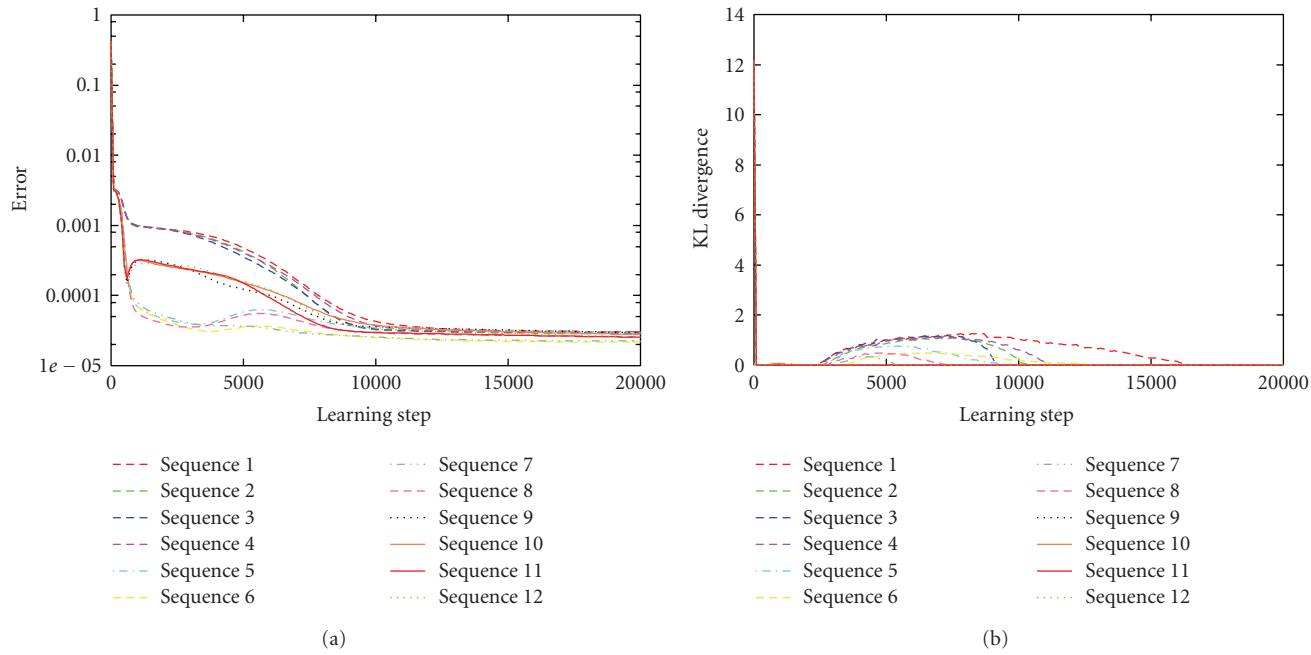
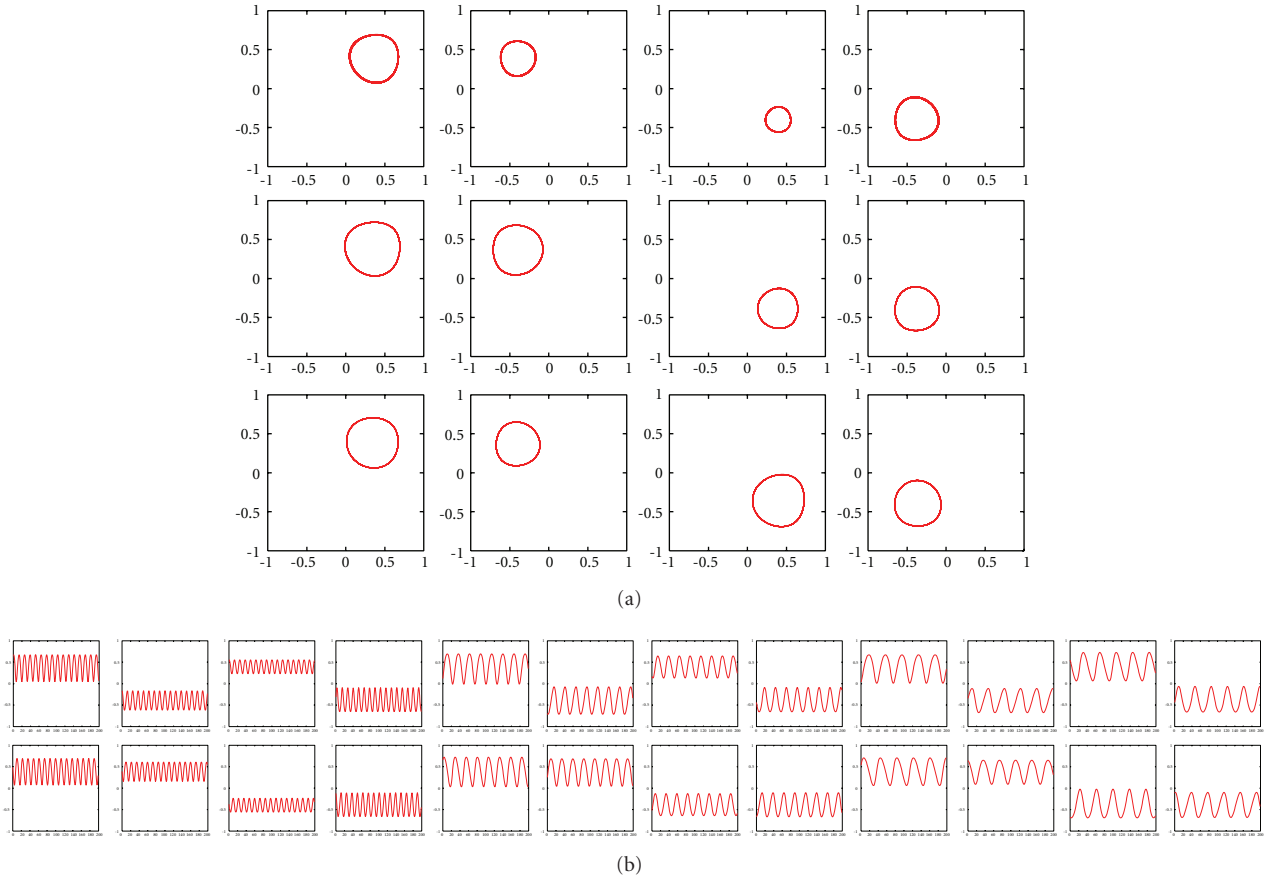



Figure 7: Error and Kullback-Leibler divergence between the teaching sequences and output generated by the RNN for 200 000 learning steps in experiment 1.



(a)

(b)

Figure 8: Time series $\mathbf{x}_n$ generated by the trained RNN in experiment 2. For each time series, the initial state $\mathbf{u}_0$ is the same as the training phase. (a) Trajectories on $\mathbb{R}^2$. (b) Temporal trajectories.

**Lemma 2.** *Let one consider a dynamical system on $\mathbb{R}^{N_3}$ defined by the transition function $\mathbf{u}_{n+1} = f_{\boldsymbol{\theta}}(g_{\boldsymbol{\theta}}(\mathbf{u}_n), \mathbf{u}_n)$, where $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}}$ are defined in (1) and (2), respectively. Assume that each element $w_{ij}$ of the matrix $\mathbf{W}_3$ satisfies (4), and $\nu \in \mathbb{R}$ is the maximum absolute value of elements in $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{b}$. If there exist three solutions of*

$$x - \omega \tanh(x) + \nu(N_1 + N_2 + 1) = 0, \quad x \in \mathbb{R}, \tag{18}$$

*then*

(1) *there are $2^{N_s}$ invariant sets of a dynamical system $\mathbf{u}_{n+1} = f_{\boldsymbol{\theta}}(g_{\boldsymbol{\theta}}(\mathbf{u}_n), \mathbf{u}_n)$;*

(2) *suppose that $U \subset \mathbb{R}^{N_3}$ is an invariant set of the dynamical system; then, the restriction of $f_{\boldsymbol{\theta}}$ to $\mathbb{R}^{N_1} \times \mathbb{R}^{N_2} \times U$ is a contraction with respect to $U$ and the maximum norm $|\cdot|_\infty$, where $|\mathbf{u} - \mathbf{u}'|_\infty \equiv \max_{1 \le i \le N_3} |u_i - u_i'|$.*

*Proof.* We suppose that (18) has three solutions, such as $x_1 > x_2 > x_3$ (see Figure 9). In general, $x_1, x_2 > 0$ and $x_3 < 0$.

(1) Assume $1 \le i \le N_s$ and $x_1 \ge u_n^{(i)} \ge x_2$. Then, the expression

$$\begin{aligned} & u_n^{(i)} - \omega \tanh\left(u_n^{(i)}\right) + \nu(N_1 + N_2 + 1) \le 0 \\ & \Longrightarrow u_n^{(i)} - \omega \tanh\left(u_n^{(i)}\right) - O_n^{(i)} \le 0 \\ & \Longleftrightarrow -\epsilon\left(u_n^{(i)} - \omega \tanh\left(u_n^{(i)}\right) - O_n^{(i)}\right) \ge 0 \\ & \Longleftrightarrow (1-\epsilon)u_n^{(i)} + \epsilon\left(\omega \tanh\left(u_n^{(i)}\right) + O_n^{(i)}\right) \ge u_n^{(i)} \\ & \Longleftrightarrow u_{n+1}^{(i)} \ge u_n^{(i)} \end{aligned} \tag{19}$$

is satisfied for all $\mathbf{x}_n \in \mathbb{R}^{N_1}$, $\mathbf{r}_n \in \mathbb{R}^{N_2}$, and $\mathbf{u}_n, \mathbf{u}_n' \in \mathbb{R}^{N_3}$, where $O_n^{(i)}$ is the $i$th element of the vector $\mathbf{O}_n = \mathbf{W}_1 \cdot \mathbf{x}_n + \mathbf{W}_2 \cdot \mathbf{r}_n + \mathbf{b}$. Hence, $u_{n+1}^{(i)} \ge u_n^{(i)}$ if $x_1 \ge u_n^{(i)} \ge x_2$. Furthermore, if $u_n^{(i)} \ge x_1$, then $u_{n+1}^{(i)} \ge x_1$ because

$$\begin{aligned} & u_n^{(i)} \ge x_1 \\ & \Longrightarrow \tanh\left(u_n^{(i)}\right) \\ & \ge \tanh\left(x_1\right) \text{ (because } x_1 > 0) \\ & \Longrightarrow (1-\epsilon)u_n^{(i)} + \epsilon\left(\omega \tanh\left(u_n^{(i)}\right) + O_n^{(i)}\right) \\ & \ge (1-\epsilon)x_1 + \epsilon\left(\omega \tanh\left(x_1\right) + O_n^{(i)}\right) \\ & \Longrightarrow u_{n+1}^{(i)} \ge x_1 \text{ (because } x_1 \ge u_n^{(i)} \ge x_2 \Longrightarrow u_{n+1}^{(i)} \ge u_n^{(i)}). \end{aligned} \tag{20}$$

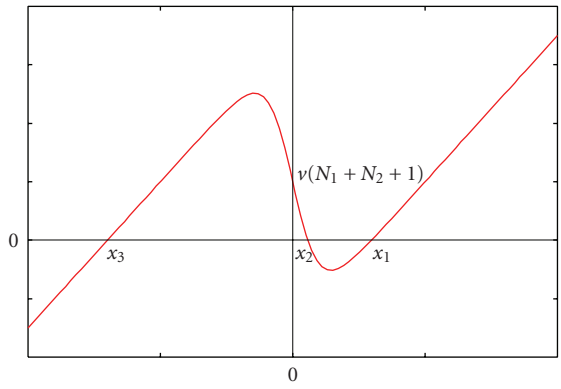



FIGURE 9: Schematic diagram of (18).

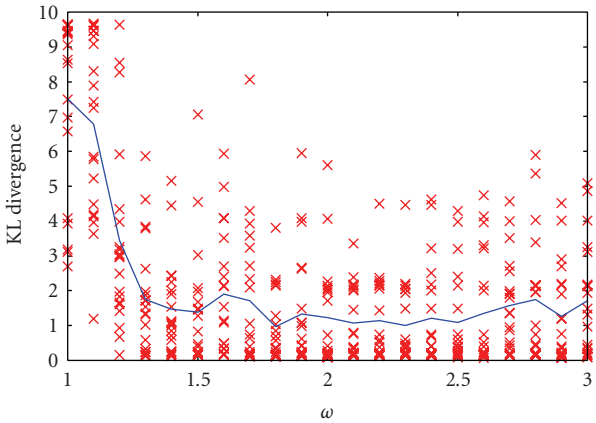



FIGURE 10: Kullback-Leibler divergence between the teaching sequences and output generated by the trained RNN with $\epsilon = 0.1$, $N_2 = 10$ ($\dim \mathbf{s}_n^{(k,l)} = 2$), $N_3 = 30$, and $N_s = 15$.

Therefore, the region $[x_1, \infty)$ is a stable set of the $i$th element of vector $\mathbf{u}_n$ satisfying the fact that if $u_n^{(i)} \in [x_1, \infty)$ then $u_{n+k}^{(i)} \in [x_1, \infty)$ for any $k \in \mathbb{N}$.

Similarly, we can easily show that if $u_n^{(i)} \leq -x_1$, then $u_{n+1}^{(i)} \leq -x_1$. Thus, there are two stable regions of the $i$th element of vector $\mathbf{u}_n$ for each $1 \leq i \leq N_s$. Then, there are $2^{N_s}$ invariant sets.

(2) Let $U \subset \mathbb{R}^{N_3}$ be the invariant set presented above. Assume that $\mathbf{u}_n, \mathbf{u}'_n \in U$.

For any $1 \leq i \leq N_s$, the inequality

$$\begin{aligned} &|u_{n+1}^{(i)} - u_{n+1}'^{(i)}| \\ &= |(1-\epsilon)(u_n^{(i)} - u_n'^{(i)}) \\ &\quad + \epsilon\omega(\tanh(u_n^{(i)}) - \tanh(u_n'^{(i)}))| \\ &< |(1-\epsilon)(u_n^{(i)} - u_n'^{(i)}) + \epsilon(u_n^{(i)} - u_n'^{(i)})| \\ &= |u_n^{(i)} - u_n'^{(i)}| \end{aligned} \tag{21}$$

holds because if $|x| \geq x_1$, then $\omega |d \tanh(x)/dx| < 1$.

On the other hand, for every $N_s \leq i \leq N_3$,

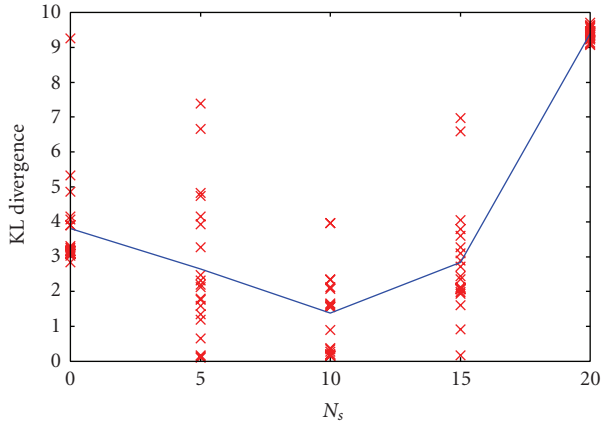
$$\begin{aligned} &|u_{n+1}^{(i)} - u_{n+1}'^{(i)}| \\ &= \Big|(1-\epsilon)(u_n^{(i)} - u_n'^{(i)}) \\ &\quad + \epsilon \sum_j w_{ij}(\tanh(u_n^{(j)}) - \tanh(u_n'^{(i)}))\Big| \\ &\leq (1-\epsilon)|u_n^{(i)} - u_n'^{(i)}| \\ &\quad + \epsilon \Big|\sum_j w_{ij}(\tanh(u_n^{(j)}) - \tanh(u_n'^{(j)}))\Big| \\ &\leq (1-\epsilon)|u_n^{(i)} - u_n'^{(i)}| + \epsilon \Big|\sum_j w_{ij}(u_n^{(j)} - u_n'^{(j)})\Big| \\ &\leq (1-\epsilon)|u_n^{(i)} - u_n'^{(i)}| + \epsilon N_3 \max_j |w_{ij}(u_n^{(j)} - u_n'^{(j)})| \\ &\leq (1-\epsilon)|u_n^{(i)} - u_n'^{(i)}| + \epsilon N_3 \max_j |w_{ij}| \max_j |u_n^{(j)} - u_n'^{(j)}| \\ &\leq ((1-\epsilon) + \epsilon N_3 \max_j |w_{ij}|) \max_j |u_n^{(j)} - u_n'^{(j)}| \\ &< \max_j |u_n^{(j)} - u_n'^{(j)}|. \end{aligned} \tag{22}$$

Then, $|\mathbf{u}_{n+1} - \mathbf{u}'_{n+1}|_\infty < |\mathbf{u}_n - \mathbf{u}'_n|_\infty$ is obtained for any $\mathbf{u}_n, \mathbf{u}'_n \in U$. Accordingly, the restriction of $f_\theta$ to $\mathbb{R}^{N_1} \times \mathbb{R}^{N_2} \times U$ is a contraction with respect to $U$ and the maximum norm $|\cdot|_\infty$. □

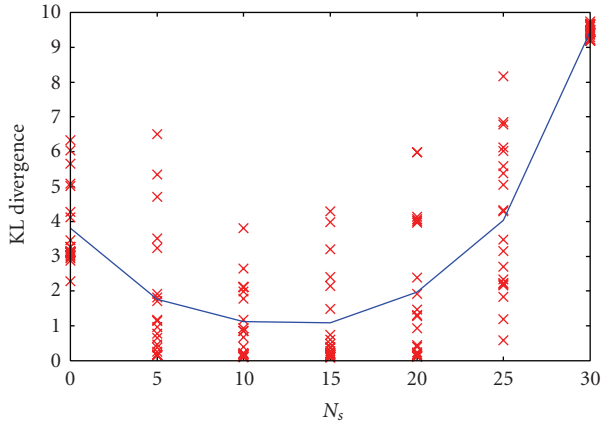
For any $N_1, N_2 \in \mathbb{N}$ and $\nu \geq 0$, there is a real number $q$ such that if $\omega \geq q$, then (18) has three solutions. Thus, if $\omega$ is large enough and matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ represent small connection weights, then $f_\theta$ contains $2^{N_s}$ invariant sets, and each restriction of $f_\theta$ to an invariant set is a contraction with respect to a third input. Moreover, the integer $N_3 - N_s$ is the effective degree of freedom for each contraction mapping restricted to an invariant set. If $N_3 - N_s$ is a large value, then RNN can acquire a more complex time sequence. In Figures 10 and 11, we plot the Kullback-Leibler divergence of the trained RNN for parameters $\omega$ and $N_s$, in which the training data are the same as those for experiment 1. These results imply that it is necessary that $\omega$, $2^{N_s}$, and $N_3 - N_s$ be large values in order to learn multiple attractor dynamics.

*4.2. Orthogonality.* In the last paragraph of the previous section, we have shown that RNNs have many stable regions, and the existence of the stable regions plays an important role in the learning of multiple sequences. However, the existence of multiple stable regions is not sufficient for success in the multiple attractor learning because if the change of parameters corresponding to each time series influences other changes, each time series cannot necessarily be embedded into each region. Similarly, this problem appears in the method of RNNPB.

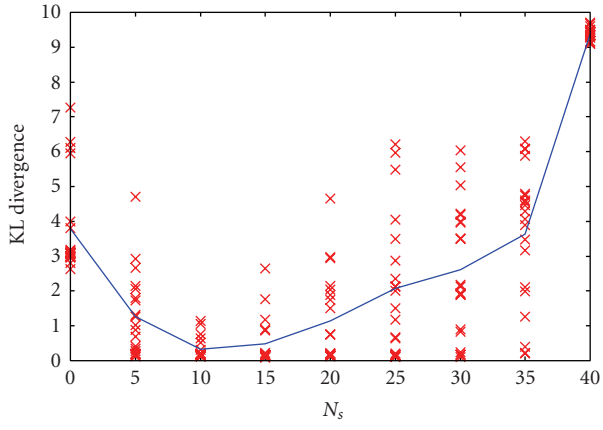
In the training algorithm defined in Section 2, each state of orthogonal units $\mathbf{r} = (\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(m)})$ is trained by (5) and (11). Thus, firing of $\mathbf{s}^{(k)}$ only occurs in the generation




(a)




(b)




(c)

FIGURE 11: Kullback-Leibler divergence between the teaching sequences and output generated by the trained RNN with $\epsilon = 0.1$, $\omega = 2.5$, and $N_2 = 10$ ($\dim \mathbf{s}_n^{(k,l)} = 2$). (a) $N_3 = 20$, (b) $N_3 = 30$, and (c) $N_3 = 40$.

of the $k$th teaching sequence. This implies that orthogonal units allow the conflict of parameter changes caused by multiple time series learning to be avoided because orbits

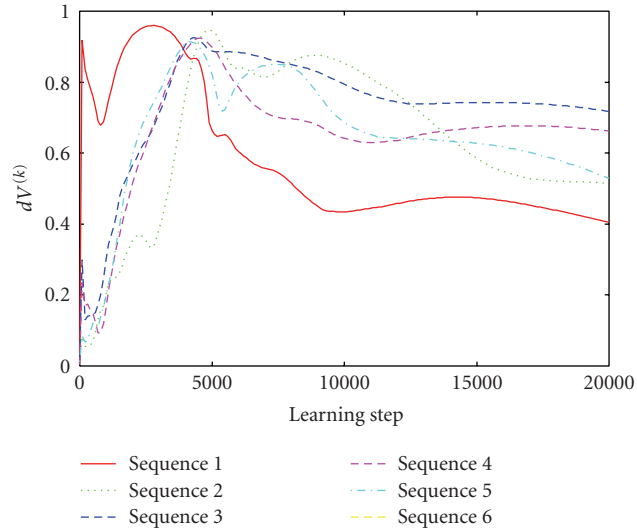



FIGURE 12: Average $dV^{(k)}(t)$ of the $k$th learning ratio for the connections between internal units and orthogonal units $\mathbf{s}^{(k)}$ for 200 000 learning steps in experiment 1.

corresponding to each teaching I/O time series run around the orthogonal state space of the trained RNN.

In order to show the effect of the orthogonal units on the conflict among teaching sequences, we consider the $k$th learning ratio $dv_{ij}^{(k)}(t)$ defined by

$$dv_{ij}^{(k)}(t) = \frac{\partial E^{(k)}(t)}{\partial v_{ij}} \left( \frac{\partial E(t)}{\partial v_{ij}} \right)^{-1}, \tag{23}$$

where $v_{ij}$ is an element of the matrix $V$. If $dv_{ij}^{(k)}(t)$ is nearly equal to 1, then the change in $v_{ij}$ is approximately independent of teaching sequences rather than the $k$th sequence. In Figure 12, we plot the value $dV^{(k)}(t)$ determined by

$$dV^{(k)}(t) = \frac{1}{(N_3 - N_s)|R^{(k)}|} \sum_{i \in R^{(k)}} \sum_{j=N_s}^{N_3} dv_{ij}^{(k)}(t), \tag{24}$$

where $R^{(k)}$ is a set of indices corresponding to the elements of the vector $\mathbf{s}^{(k)}$. The value $dV^{(k)}(t)$ represents the average of the $k$th learning ratio for connections between internal units and orthogonal units $\mathbf{s}^{(k)}$. In this numerical experiment, for each learning step, $dV^{(k)}(t)$ is clearly larger than $1/m = 0.2$, where $m$ is the number of teaching sequences. Then, the sum of the $k$th learning ratios of connection weights between internal units and orthogonal units $\mathbf{s}^{(k)}$ is dominant. Therefore, in changing matrix $V$, there is no conflict generated by multiple teaching sequences. However, we could not find a strong bias of the learning ratio for the matrices $W_1$ and $W_2$ and every element $v_{ij}$ of $V$ with $i < N_s$. Thus, we consider that connection weights between internal units and orthogonal units encode information on an individual time series, and other connection weights encode whole information.

## 5. Discussion

In this report, we have investigated a method of embedding multiple time series into a single RNN. In order to clarify the characteristics of the proposed approach, we compare the proposed approach with other approaches with respect to information representation of multiple sequences in the models. The mixture-of-RNN-experts-type model composes local representation in an RNN for each sequence. The local representation provides robustness against changing the parameters in learning, but it lacks the ability to extract common patterns included in the sequences because of the independency of the local representation. In the proposed model, the local representation is constructed into orthogonal units, while the global representation is also constructed into internal units using the connection weights between I/O units and internal units. Since each sequence generated by the proposed model shares the state space and connection weights, the model can extract common patterns of the sequences as well as conventional neural networks.

Another characteristic, which clarifies the difference between our model and other models, is whether the classification of each time series is self-organized into the state space. For example, in the mixture-of-RNN-experts-type model, the allocation of time series to each RNN is determined automatically. As another example, in the RNNPB model, PB values are self-organized such that the PB can individualize each time series. On the other hand, the proposed model needs the information of orthogonalization for each time series. Since the sparse firing patterns which appear in orthogonal units, corresponding to time series, are given as teaching information externally, the classification of sequences is not self-organized. The characteristic whereby the time series cannot be automatically classified is a disadvantage of the proposed model. However, the time series can be classified using other clustering techniques before applying the proposed method. Thus, by combining the proposed method and other clustering techniques, an algorithm that automatically classifies and generates multiple time series can be constructed.

## 6. Conclusion

In this paper, we have presented an RNN model and a learning algorithm that can acquire the ability to generate multiple sequences. The RNN model consists of two distinct properties called bistability and orthogonality. Bistability guarantees the existence of multiple attractor structures in RNNs, and provides the RNNs with contraction transition mapping. Orthogonality, which is given as a function of the orthogonal vectors of RNNs, helps prevent conflicts with respect to parameter changes caused by multiple training sequences. In the numerical experiments, RNNs which have bistability and orthogonality can learn multiple periodic attractors constituted by five Lissajous curves or 12 Van der Pol oscillators. Based on these results, the proposed model can be applied to the modeling of various types of dynamical systems that include multiple attractors.

## References


[1] K. Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks," *Neural Networks*, vol. 6, no. 6, pp. 801–806, 1993.

[2] H. T. Siegelmann and E. D. Sontag, "Analog computation via neural networks," *Theoretical Computer Science*, vol. 131, no. 2, pp. 331–360, 1994.

[3] H. T. Siegelmann and E. D. Sontag, "On the computational power of neural nets," *Journal of Computer and System Sciences*, vol. 50, no. 1, pp. 132–150, 1995.

[4] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[5] K. Doya and S. Yoshizawa, "Memorizing oscillatory patterns in the analog neuron network," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN '89)*, vol. 1, pp. 27–32, Washington, DC, USA, June 1989.

[6] F.-S. Tsung, *Modeling dynamical systems with recurrent neural networks*, Ph.D. thesis, Department of Computer Science, University of California, San Diego, Calif, USA, 1994.

[7] D. M. Wolpert and M. Kawato, "Multiple paired forward and inverse models for motor control," *Neural Networks*, vol. 11, no. 7-8, pp. 1317–1329, 1998.

[8] J. Tani and S. Nolfi, "Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems," *Neural Networks*, vol. 12, no. 7-8, pp. 1131–1141, 1999.

[9] J. Tani, "Learning to generate articulated behavior through the bottom-up and the top-down interaction processes," *Neural Networks*, vol. 16, no. 1, pp. 11–23, 2003.

[10] J. Tani and M. Ito, "Self-organization of behavioral primitives as multiple attractor dynamics: a robot experiment," *IEEE Transactions on Systems, Man and Cybernetics Part A*, vol. 33, no. 4, pp. 481–488, 2003.

[11] H. Jaeger, "Short term memory in echo state networks," Tech. Rep. 152, National Research Center for Information Technology, Bremen, German, 2001.

[12] H. Jaeger and H. Haas, "Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004.

[13] W. Maass, T. Natschläger, and H. Markram, "A fresh look at real-time computation in generic recurrent neural circuits," Tech. Rep., Institute for Theoretical Computer Science, TU Graz, Graz, Austria, 2002.

[14] B. Hammer and P. Tiňo, "Recurrent neural networks with small weights implement definite memory machines," *Neural Computation*, vol. 15, no. 8, pp. 1897–1929, 2003.

[15] O. L. White, D. D. Lee, and H. Sompolinsky, "Short-term memory in orthogonal neural networks," *Physical Review Letters*, vol. 92, no. 14, Article ID 148102, 4 pages, 2004.

[16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, D. E. Rumelhart and J. L. McLelland, Eds., pp. 318–362, MIT Press, Cambridge, Mass, USA, 1986.

[17] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.