Learning Properties of Recurrent Neural Network with Parametric Biases

Kazushi Ikeda
Department of Systems Science
Kyoto University
Kyoto 606-8501 Japan

Kentaro Narita
Department of Systems Science
Kyoto University
Kyoto 606-8501 Japan

Abstract

Some properties of the recurrent neural network with parametric biases (RNNPB) are discussed here. The RNNPB is applied to a humanoid robot and is a candidate model of the mirror system in neuroscience. A recent experimental study reported that the RNNPB encodes given periodical patterns into the PB vector in a smart manner. This study is the first step to reveal how the RNNPB works. More concretely, some conditions are given for the transition matrix of the RNNPB to produce the stored dynamical patterns and how the learning algorithm of the RNNPB works is shown.

1 Introduction

One of the applications of neural networks is the central pattern generator of a robot or others since a recurrent neural network can store periodical patterns as attractors (limit-cycles) by learning them [1]. The recurrent neural network with parametric biases (RNNPB) is one of such robot controllers [2] and has recently attracted attentions as a humanoid controller [3] as well as a model of mirror neurons in neuroscience [4–6].

The RNNPB consists of a multi-layer perceptron, feedback connections and the PB neurons, where the multi-layer perceptron receives two types of input. One is the vector of the current internal states via the feedback connections some components of which are observed as the output of the RNNPB, and the other is the vector of parameter biases (PBs) which externally controls the output.

The RNNPB has the following three phases: The learning phase where the RNNPB learns the given dynamical patterns using the back-propagation through time (BPTT) algorithm [1] as well as the PBs are updated in a self-organizing manner, the generation phase where the RNNPB outputs a dynamical pattern according to the PB vector given externally, and the interaction phase where the PB vector is updated so that the output of the RNNPB coincides with the desired one.

In general, the number of PBs is much less than that of internal states. This means that high-dimensional dynamical patterns are encoded into the PB space in the learning phase. In the generation phase, on the other hand, the RNNPB decodes a PB vector and retrieves the corresponding dynamical pattern. Hence, the relationship between the dynamical patterns and the PB vectors is essential in analyzing what the RNNPB does in the learning and generation phases. However, it has little been clarified so far due to its complicated structure and update equations.

Recently, Yamada and Suyari showed by exhaustive experiments that the three parameters of a sinusoidal, the frequency, the amplitude and the offset, are smoothly mapped into the PB space so that the topology of the parameter space is kept [7]. This result suggests that the RNNPB would have an elegant theory in encoding dynamical patterns. The purpose of this study is to give a theoretical background to the useful neural model, taking the experimental results into account.

2 Problem Formulation

In order to understand the RNNPB, we try to give a theoretical background to the results by Yamada and Suyari. Since sinusoids are a basis-set of the space of periodical patterns, we substitute a linear system for a multi-layer perceptron and see how the RNNPB learns periodical patterns as limit-cycles and codes their properties into the PB vector.

The problem we treat in this study is formulated as follows. The one-dimensional output, the $(N-1)$-dimensional unobservable internal state vector and the M-dimensional PB vector at time t are denoted by s_t, $\boldsymbol{c}_t$ and $\boldsymbol{b}_t$, respectively, where $M < N$ as we consider how the dynamical patterns are encoded into the PB vector. In this study, we assume that the PB vector is constant, that is, the self-organization of the PB vector is finished. Then, the N-dimensional vector $(s_t, \boldsymbol{c}_t^{\mathrm{T}})^{\mathrm{T}}$ is the state vector and is updated as

$$\begin{pmatrix} s_{t+1} \\ \boldsymbol{c}_{t+1} \end{pmatrix} = A \begin{pmatrix} s_t \\ \boldsymbol{c}_t \end{pmatrix} + \boldsymbol{b}, \tag{1}$$

where A is the state transition matrix.

It is obvious that an arbitrary time-series with period N is reproduced if A equals to the shift matrix

$$S \equiv \begin{pmatrix} \boldsymbol{0}_{N-1} & I_{N-1} \\ 1 & \boldsymbol{0}_{N-1}^{\mathrm{T}} \end{pmatrix} \tag{2}$$

and $\boldsymbol{b}$ is null, where I_n and $\boldsymbol{0}_n$ are the n-dimensional identity matrix and the n-dimensional null vector, respectively, and $^{\mathrm{T}}$ denotes the transposition of a matrix or a vector.

3 Transition Matrix for Periodical Patterns

We first introduce the conditions on the transition matrix A under which the RNNPB can produces the stored dynamical patterns with period N, assuming $\boldsymbol{b} = \boldsymbol{0}_M$. This problem is not easy since it includes an essential ambiguity in A and the internal state $\boldsymbol{c}_t$. In fact, the RNNPB produces the same output when the internal state is

$$\boldsymbol{c}'_t \equiv \begin{pmatrix} 1 & \boldsymbol{0}_{N-1}^{\mathrm{T}} \\ \boldsymbol{0}_{N-1} & Q \end{pmatrix} \begin{pmatrix} s_t \\ \boldsymbol{c}_t \end{pmatrix} \tag{3}$$

instead of $(s_t, \boldsymbol{c}_t^{\mathrm{T}})^{\mathrm{T}}$ and

$$A' \equiv \begin{pmatrix} 1 & \boldsymbol{0}_{N-1}^{\mathrm{T}} \\ \boldsymbol{0}_{N-1} & Q \end{pmatrix} A \begin{pmatrix} 1 & \boldsymbol{0}_{N-1}^{\mathrm{T}} \\ \boldsymbol{0}_{N-1} & Q^{-1} \end{pmatrix} \tag{4}$$

instead of A, where Q is an arbitrary regular matrix.

To remove this ambiguity, we rewrite (1) to a form without $\boldsymbol{c}_t$, that is, an update equation of

$$\boldsymbol{s}_t^{\mathrm{T}} \equiv \begin{pmatrix} s_t & s_{t+1} & \cdots & s_{t+N-1} \end{pmatrix}. \tag{5}$$

In fact,

$$\boldsymbol{s}_t = C \begin{pmatrix} s_t \\ \boldsymbol{c}_t \end{pmatrix} \tag{6}$$

holds true from (1) where

$$C \equiv \begin{pmatrix} (A^0)_1 \\ (A^1)_1 \\ \vdots \\ (A^{N-1})_1 \end{pmatrix} = \begin{pmatrix} 1 & \boldsymbol{0}_{N-1}^{\mathrm{T}} \\ \boldsymbol{a}' & A'' \end{pmatrix}, \tag{7}$$

$(A^k)_1$ is the first row of A^k, and $\boldsymbol{a}'$ and A'' are a certain vector and a certain matrix. Using the ambiguity of $\boldsymbol{c}_t$ and A, we set $Q = A''^{-1}$ in (3) and get

$$C = \begin{pmatrix} 1 & \boldsymbol{0}_{N-1}^{\mathrm{T}} \\ \boldsymbol{a} & I_{N-1} \end{pmatrix} \tag{8}$$

$$C^{-1} = \begin{pmatrix} 1 & \boldsymbol{0}_{N-1}^{\mathrm{T}} \\ -\boldsymbol{a} & I_{N-1} \end{pmatrix}. \tag{9}$$

From (6), (1) is rewritten as

$$\boldsymbol{s}_{t+1} = CAC^{-1}\boldsymbol{s}_t \tag{10}$$

and then

$$S^{t+1}\boldsymbol{f}_{t+1} = CAC^{-1}S^t\boldsymbol{f}_t \tag{11}$$

where

$$\boldsymbol{f}_t \equiv S^{-t}\boldsymbol{s}_t. \tag{12}$$

Since $\boldsymbol{s}_t$ being periodical is equivalent to $\boldsymbol{f}_t$ being constant, its condition is described as $CAC^{-1} = S$, that is, $A = C^{-1}SC$ or

$$A = \begin{pmatrix} a_1 & 1 & 0 & 0 & \cdots & 0 \\ -a_1^2 + a_2 & -a_1 & 1 & 0 & \cdots & 0 \\ -a_1 a_2 + a_3 & -a_2 & 0 & 1 & \ddots & 0 \\ -a_1 a_3 + a_4 & -a_3 & 0 & 0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ -a_1 a_{N-1} + 1 & -a_{N-1} & 0 & 0 & \cdots & 0 \end{pmatrix} \tag{13}$$

from (8), where a_k is the kth element of $\boldsymbol{a}$.

Substituting this for (1), we get

$$s_{t+1} = a_1 s_t + c_t \tag{14}$$

$$c_{t+1} = (-a_1^2 + a_2)s_t - a_1 c_t + c_{t+1} \tag{15}$$

$$c_{t+2} = (-a_1 a_2 + a_3)s_t - a_2 c_t + c_{t+2} \tag{16}$$

$$\vdots \tag{17}$$

$$c_{t+N-1} = (-a_1 a_{N-1} + a_3)s_t - a_{N-1}c_t + c_{t+N-1} \tag{18}$$

and can show $a_k = 0$ for $k = 1, \ldots, N$ from the periodicity of $\boldsymbol{s}_t$. This means $C = I$ and hence $A = S$. From the ambiguity in (3),

$$A = \begin{pmatrix} 1 & \boldsymbol{0}_{N-1}^{\mathrm{T}} \\ \boldsymbol{0}_{N-1} & Q^{-1} \end{pmatrix} S \begin{pmatrix} 1 & \boldsymbol{0}_{N-1}^{\mathrm{T}} \\ \boldsymbol{0}_{N-1} & Q \end{pmatrix} \tag{19}$$

is the condition under which the RNNPB produces arbitrary periodical patterns, where Q is an arbitrary regular matrix.

Although (19) is derived under the assumption that the RNNPB produces arbitrary patterns with period N, we can easily show that the result above still stands if $\boldsymbol{s}_t, \boldsymbol{s}_{t+1}, \ldots, \boldsymbol{s}_{t+N-1}$ are linearly independent, that is,

$$\sum_{\tau=t}^{t+N-1} \boldsymbol{s}_\tau \boldsymbol{s}_\tau^{\mathrm{T}} \tag{20}$$

has full rank, which we assume in the following.

4 PB Vector and Corresponding Pattern

In this section, we see how the PB vector $\boldsymbol{b}$ affects the dynamical pattern the RNNPB produces. When $\boldsymbol{b}$ is constant, the state vector is updated as

$$\begin{pmatrix} s_{t+1} \\ \boldsymbol{c}_{t+1} \end{pmatrix} = A \begin{pmatrix} s_t \\ \boldsymbol{c}_t \end{pmatrix} + \boldsymbol{b} \tag{21}$$

$$\begin{pmatrix} s_{t+2} \\ \boldsymbol{c}_{t+2} \end{pmatrix} = A^2 \begin{pmatrix} s_t \\ \boldsymbol{c}_t \end{pmatrix} + A\boldsymbol{b} + \boldsymbol{b} \tag{22}$$

$$\vdots \tag{23}$$

$$\begin{pmatrix} s_{t+N} \\ \boldsymbol{c}_{t+N} \end{pmatrix} = A^N \begin{pmatrix} s_t \\ \boldsymbol{c}_t \end{pmatrix} + \sum_{j=0}^{N-1} A^j \boldsymbol{b} = \begin{pmatrix} s_t \\ \boldsymbol{c}_t \end{pmatrix}. \tag{24}$$

Letting the first element of

$$\sum_{j=0}^{k-1} A^j \boldsymbol{b} \tag{25}$$

be the $(k-1)$st element of $\boldsymbol{b}'$, that is, $\boldsymbol{b}' = LC\boldsymbol{b}$ where

$$L \equiv \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \ddots & 0 & 0 \\ 1 & 1 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 & 0 \end{pmatrix} \tag{26}$$

and C is defined in (7), then

$$\boldsymbol{s}_t = C \begin{pmatrix} s_t \\ \boldsymbol{c}_t \end{pmatrix} + LC\boldsymbol{b} \tag{27}$$

holds true, which leads to

$$\boldsymbol{s}_t - LC\boldsymbol{b} = C \begin{pmatrix} s_t \\ \boldsymbol{c}_t \end{pmatrix}. \tag{28}$$

Hence, (21) is rewritten using (12) as

$$\boldsymbol{f}_{t+1} = S^{-t-1} CAC^{-1} S^t \boldsymbol{f}_t + (L - CAC^{-1}L + I)C\boldsymbol{b}. \tag{29}$$

This means

$$\boldsymbol{f}_t = \frac{1}{1-c}(L - CAC^{-1}L + I)C\boldsymbol{b} \tag{30}$$

is constant when $CAC^{-1} = cS$ for $|c| < 1$.

When $\boldsymbol{f}_t$ is a constant vector $\boldsymbol{f}$, conversely,

$$\left(I - S^{-t-1}CAC^{-1}S^t\right) \boldsymbol{f} = (L - CAC^{-1}L + I)C\boldsymbol{b} \tag{31}$$

holds and

$$B \equiv I - S^{-t-1}CAC^{-1}S^t \tag{32}$$

does not depend on t, where all the eigenvalues of B must be in $(0, 2)$. Substituting $t = 0$, we get $S(I - B) = CAC^{-1}$ and hence

$$I - B = S^{-t-1}CAC^{-1}S^t \tag{33}$$

$$= S^{-t}(I - B)S^t \tag{34}$$

is a constant matrix. This means that $I - B$ (and also B) is a cyclic Toeplitz matrix.

5 Learning Properties of BPTT

The RNNPB employs the BPTT algorithm to learn given dynamical patterns, which approximates a recurrent network to a layered one of finite length and applies the error back-propagation algorithm [1]. In this section, we discuss how the linearized RNNPB learns the patterns with the BPTT algorithm.

Let the transition matrix A be divided to four components as

$$A = \begin{pmatrix} a_{11} & \boldsymbol{a}_{12}^{\mathrm{T}} \\ \boldsymbol{a}_{21} & A_{22} \end{pmatrix}, \tag{35}$$

where we assume that A satisfies (19), that is, this is the goal. The transition matrix in the learning phase at time t is denoted by $\hat{A}$ and their components by

$$\hat{A} = \begin{pmatrix} \hat{a}_{11} & \hat{\boldsymbol{a}}_{12}^{\mathrm{T}} \\ \hat{\boldsymbol{a}}_{21} & \hat{A}_{22} \end{pmatrix}. \tag{36}$$

Note that we omit the subscript t for simplicity.

The BPTT algorithm in the RNNPB approximate the recurrent network to the three-layered network, that is,

$$s_{t+1} = a_{11}s_t + \boldsymbol{a}_{12}^{\mathrm{T}}\boldsymbol{c}_t \tag{37}$$

$$= a_{11}s_t + \boldsymbol{a}_{12}^{\mathrm{T}}\boldsymbol{a}_{21}s_{t-1} + \boldsymbol{a}_{12}^{\mathrm{T}}A_{22}\boldsymbol{c}_{t-1}, \tag{38}$$

$$\hat{s}_{t+1} = \hat{a}_{11}s_t + \hat{\boldsymbol{a}}_{12}^{\mathrm{T}}\boldsymbol{c}_t \tag{39}$$

$$= \hat{a}_{11}s_t + \hat{\boldsymbol{a}}_{12}\hat{\boldsymbol{a}}_{21}^{\mathrm{T}}s_{t-1} + \hat{\boldsymbol{a}}_{12}^{\mathrm{T}}\hat{A}_{22}\boldsymbol{c}_{t-1}. \tag{40}$$

and applies the steepest descent method to the squared error $\varepsilon(\hat{A}) = (\hat{s}_{t+1} - s_{t+1})^2/2$.

Since (38) has terms of the second order with respect to $\hat{A}$, a direct analysis is difficult. Therefore, we divide it two parts: One is a_{11} and $\boldsymbol{a}_{12}$ and the other is $\boldsymbol{a}_{21}$, A_{22}, since the latter appears via $\boldsymbol{c}_t$.

From (37) and (39), the BPTT algorithm for a_{11} and $\boldsymbol{a}_{12}$ is expressed as

$$\Delta \begin{pmatrix} \hat{a}_{11} \\ \hat{\boldsymbol{a}}_{12} \end{pmatrix} = -\eta \begin{pmatrix} s_t \\ \boldsymbol{c}_t \end{pmatrix} \begin{pmatrix} s_t & \boldsymbol{c}_t^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \hat{a}_{11} - a_{11} \\ \hat{\boldsymbol{a}}_{12} - \boldsymbol{a}_{12} \end{pmatrix} \tag{41}$$

and can be shown to make $(\hat{a}_{11}, \hat{\boldsymbol{a}}_{12}^{\mathrm{T}})$ converge to $(a_{11}, \boldsymbol{a}_{12}^{\mathrm{T}})$ when the assumption (20) is satisfied and η is appropriately small.

In the analysis of $(\hat{\boldsymbol{a}}_{21}, \hat{A}_{22})$ based on (38), we assume that $(\hat{a}_{11}, \hat{\boldsymbol{a}}_{12}^{T})$ already converges to $(a_{11}, \boldsymbol{a}_{12}^{T})$, that is,

$$\hat{s}_{t+1} - s_{t+1} = \boldsymbol{a}_{12}^{\mathrm{T}}(\hat{\boldsymbol{a}}_{21} - \boldsymbol{a}_{21})s_{t-1} + \boldsymbol{a}_{12}^{\mathrm{T}}(\hat{A}_{22} - A_{22})\boldsymbol{c}_{t-1}. \tag{42}$$

Then, the BPTT algorithm for $\hat{\boldsymbol{a}}_{21}$ and $\hat{A}_{22}$ is expressed as

$$\Delta \begin{pmatrix} \tilde{\boldsymbol{a}}_{21} & \tilde{A}_{22} \end{pmatrix} = -\eta \boldsymbol{a}_{12}\boldsymbol{a}_{12}^{T} \cdot \begin{pmatrix} \hat{\boldsymbol{a}}_{21} - \boldsymbol{a}_{21} & \hat{A}_{22} - A_{22} \end{pmatrix} \begin{pmatrix} s_{t-1} \\ \boldsymbol{c}_{t-1} \end{pmatrix} \begin{pmatrix} s_{t-1} & \boldsymbol{c}_{t-1}^{\mathrm{T}} \end{pmatrix}. \tag{43}$$

Therefore, under the assumption (20) $(\hat{\boldsymbol{a}}_{21}\hat{A}_{22})$ stops when it satisfies

$$\boldsymbol{a}_{12}^{\mathrm{T}} \begin{pmatrix} \hat{\boldsymbol{a}}_{21} & \hat{A}_{22} \end{pmatrix} = \boldsymbol{a}_{12}^{\mathrm{T}} \begin{pmatrix} \boldsymbol{a}_{21} & A_{22} \end{pmatrix}, \tag{44}$$

which means that $\begin{pmatrix} \hat{\boldsymbol{a}}_{21} & \hat{A}_{22} \end{pmatrix}$ does not converge to $\begin{pmatrix} \boldsymbol{a}_{21} & A_{22} \end{pmatrix}$.

6 Conclusions and Discussions

The analysis so far reveals that the essence of the RNNPB may be the shift operator, which explains the result by Yamada and Suyari very well. However, the analysis based on the linear approximation is limited, as shown in the previous section, since the RNNPB cannot learn the dynamics by the BPTT algorithm. More extensive analysis is our future work.

Acknowledgments

This study is supported in part by a Grant-in-Aid for Scientific Research (15700130, 18300078) from the Japanese Government.

References

[1] P. Werbos, "Backpropagation through time: What does it mean and how to do it," *Proc. IEEE*, vol. 78, pp. 1550–1560, 1990.

[2] 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.

[3] M. Ito, K. Noda, Y. Hoshino, and J. Tani, "Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model," *Neural Networks*, vol. 19, no. 3, pp. 323–337, 2006.

[4] J. Tani, M. Ito, and Y. Sugita, "Self-organization of distributedly represented multiple behavior schemata in a mirror systems: Reviews of robot experiments using RNNPB," *Neural Networks*, vol. 17, no. 8–9, pp. 1273–1289, 2004.

[5] M. Ito and J. Tani, "On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system," *Adaptive Behavior*, vol. 12, no. 2, pp. 93–115, 2004.

[6] E. Oztop, M. Kawato, and M. Arbib, "Mirror neurons and imitation: A computationally guided review," *Neural Networks*, vol. 19, no. 3, pp. 254–271, 2006.

[7] K. Yamada and H. Suyari, "The properties of the PB space self-organized by RNNPB learning," IEICE, Tech. Rep. NC2004-183, 2005.