2022 International Symposium on Nonlinear Theory and Its Applications, NOLTA2022, Virtual, December 12-15, 2022

Synthesis of Permutation Binary Neural Networks

Kento Saka† and Toshimichi Saito†

†Department of Electronics and Electrical Engineering, Hosei University
3–7–2 Kajino-cho, Koganei, Tokyo 184-8584, Japan, tsaito@hosei.ac.jp

Abstract— This paper studied a simple synthesis method of permutation binary neural networks. The network is characterized by local binary connection, global permutation connection, and the signum activation function. Depending on the connection, the network generates various periodic orbits of binary vectors. The synthesis method is based on the genetic algorithm. Performing basic numerical experiments, efficiency of the synthesis method is confirmed.

1. Introduction

Discrete-time recurrent-type neural networks are characterized by real valued connection parameters and nonlinear activation functions [1]-[3]. In the networks, fixed points have been analyzed sufficiently. However, analysis of periodic orbits is extremely hard. In order to consider periodic orbits, we introduce a permutation binary neural network (PBNN [4]). The PBNN is simple recurrent-type network with a variety of periodic orbits. The PBNN is characterized by local binary connection, global permutation connection, and the signum activation function. The PBNN can be regarded as a simplified version of the dynamic binary neural network (DBNN [5] [6]). Depending on the parameters, the PBNN generates various periodic orbits of binary vectors (BPOs). Engineering applications of the BPOs include control signal of switching power converters [7], control signal of walking robots [8], and reservoir computing for time-series approximation/prediction [9].

This paper presents synthesis of a PBNN that generates longer BPOs. In order to evaluate BPOs, we define two feature quantities. The first feature quantity evaluates period of the BPOs. The second feature quantity evaluates direct stability of the BPOs. As the system size increases, the brute force attack of permutation connection becomes impossible. In order to realize the synthesis effectively, we present a simple algorithm based on the genetic algorithm. In the algorithm, an individual corresponds to a permutation and the objective function is the first feature quantity. Performing basic numerical experiments, efficiency of the synthesis method is confirmed. This is the first paper of a synthesis method of PBNN.

2. Permutation Binary Neural Networks

First, we introduce a simple binary neural network (SBNN) corresponding to the input to hidden layers in the PBNN. The dynamics is described by

$$\begin{gathered} x_i^{t+1} = \mathrm{sgn}\left(w_a x_{i-1}^t + w_b x_i^t + w_c x_{i+1}^t\right) \\ i \in \{1, \cdots, N\} \\ \mathrm{sgn}(X) = \begin{cases} +1 & \text{for } X \geq 0 \\ -1 & \text{for } X < 0 \end{cases} \end{gathered} \tag{1}$$

where $x_i^t \in \boldsymbol{B}$ is the i-th binary state variable at discrete time t. As shown in Fig. 1 (a), $x_{N+1}^t \equiv x_1^t$ and $x_0^t \equiv x_N^t$ for ring-type connection. The SBNN is characterized by local binary connection and the signum activation function. The local binary connection is determined by three binary valued connection parameters $w_a \in \boldsymbol{B}$, $w_b \in \boldsymbol{B}$, and $w_c \in \boldsymbol{B}$. Fig. 1 illustrates an example of SBNN and BPO.

Adding global permutation connection from hidden to output layer, the PBNN is constructed. The dynamics is described by

$$\begin{aligned} x_i^{t+1} &= y_{\sigma(i)}^t \\ y_i^t &= \mathrm{sgn}\left(w_a x_{i-1}^t + w_b x_i^t + w_c x_{i+1}^t\right) \\ \sigma &= \begin{pmatrix} 1 & 2 & \cdots & N \\ \sigma(1) & \sigma(2) & \cdots & \sigma(N) \end{pmatrix} \end{aligned} \tag{2}$$

where $y_i^t \in \boldsymbol{B}$ is the i-th binary hidden state and σ is a permutation. Let $\boldsymbol{x}^t \equiv (x_1^t, \cdots, x_N^t)$ and let $\boldsymbol{y}^t \equiv (y_1^t, \cdots, y_N^t)$. The local binary connection (from input to hidden layers) transforms a binary input vector $\boldsymbol{x}^t$ into the binary hidden vector $\boldsymbol{y}^t$. The global

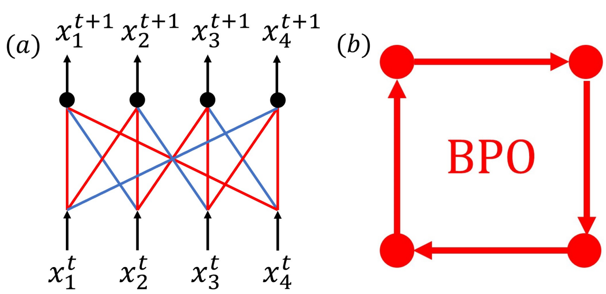

Figure 1: An example of SBNN and BPO. (a) Network configuration. Red and blue branches denote positive and negative connections, respectively. (b) BPO with period 4.

permutation connection (from hidden to output layers) transforms the $\boldsymbol{y}^t$ into the binary output vector $\boldsymbol{x}^{t+1}$. Depending on the parameters and initial conditions, the PBNNs can generate various BPOs. A BPO with period p is defined by

$$\boldsymbol{z}^1, \cdots, \boldsymbol{z}^p, \cdots \quad \begin{cases} \boldsymbol{z}^{t_1} = \boldsymbol{z}^{t_2} & \text{for } |t_2 - t_1| = np \\ \boldsymbol{z}^{t_1} \neq \boldsymbol{z}^{t_2} & \text{for } |t_2 - t_1| \neq np \end{cases} \tag{3}$$

where $\boldsymbol{z}^t = (z_1^t, \cdots, z_N^t)$, $z_i^t \in \boldsymbol{B}$ and n denotes positive integers. An element $\boldsymbol{z}^t$ of the BPO is referred to as a binary periodic point (BPP). An initial point $\boldsymbol{z}^e$ is said to be a direct eventually periodic point (DEPP) if $\boldsymbol{z}^e$ is not a BPP but falls into a BPO directly. Fig. 2 illustrates an example of PBNN given by

$$(w_a, w_b, w_c) = (+1, +1, -1), \ \sigma_A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{pmatrix} \tag{4}$$

As shown in Fig. 2 (b), this PBNN exhibits a BPO with period 6. In Fig. 2 (a), the local binary connection is the same as the connection of SBNN in Fig. 1 (a): applying the global permutation connection to the SBNN of BPO with period 4, we obtain the PBNN that generates BPO with period 6.

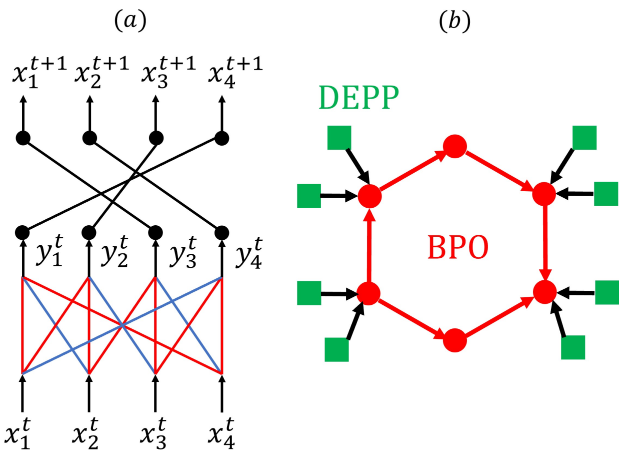

Figure 2: An example of PBNN and BPO. (a) Network configuration. Black branches denote permutation connection. (b) BPO with period 6 and DEPPs.

3. Feature quantities and synthesis algorithm

In order to evaluate BPOs, we define two feature quantities. The PBNN can have multiple BPOs and exhibits one of them depending on the initial points. Since consideration of multiple BPOs is not easy, we consider one BPO with the maximum period (MBPO). The first feature quantity is defined by

$$F_1(\sigma) = \frac{\text{The period of an MBPO}}{2^N} \tag{5}$$

where $0 \leq F_1(\sigma) \leq 1$. Roughly speaking, $F_1(\sigma)$ evaluates complexity of the MBPO. The second feature quantity is defined by

$$F_2(\sigma) = \frac{\text{\# DEPPs falling into the MBPO}}{2^N} \tag{6}$$

where $0 \leq F_2(\sigma) \leq 1$. $F_2(\sigma)$ evaluates direct stability of the MBPO. As $F_2(\sigma)$ increases, direct stability of the MBPO becomes stronger. If multiple MBPOs exist, one MBPO with larger $F_2(\sigma)$ is adopted. The direct stability corresponds to 1-bit error correction in binary code. If Fig. 2 (b) gives the MBPO, we obtain

$$F_1(\sigma_A) = \frac{6}{16}, F_2(\sigma_A) = \frac{8}{16} \tag{7}$$

In order to obtain an MBPO with long period, we introduce a simple evolutionary algorithm. In the algorithm, $F_1(\sigma)$ is used as the objective function.

$$\text{Maximize } F_1(\sigma) \tag{8}$$

Let σ^k denote the k-th individual corresponding to the permutation. A set of M_g individuals $\{\sigma^1, \cdots, \sigma^{M_g}\}$ is referred to as a population. Let g denote the generation of the algorithm evolution. The evolutionary algorithm is defined as the following.

Step 1 (Initialization):
Let $g = 1$. We prepare an initial identity populations.

$$\sigma^1 = \cdots = \sigma^{M_g} = \begin{pmatrix} 1 & 2 & \cdots & N \\ 1 & 2 & \cdots & N \end{pmatrix} \tag{9}$$

Step 2(Genetic operation): For $k = 1, \cdots, M_g$, do.

- Using ranking selection, one individual is selected.
- Applying a mutation to the selected individual, a new individual is generated. The mutation means exchanging two randomly selected elements in σ. For example, if the 2nd element and Nth element are selected, then the following is executed:

$$\begin{gathered} \sigma = \begin{pmatrix} 1 & 2 & \cdots & N \\ \sigma(1) & \sigma(2) & \cdots & \sigma(N) \end{pmatrix} \\ \Downarrow \\ \sigma = \begin{pmatrix} 1 & 2 & \cdots & N \\ \sigma(1) & \sigma(N) & \cdots & \sigma(2) \end{pmatrix} \end{gathered} \tag{10}$$

Step 3 (Evaluation):
Each individual is evaluated by $F_1(\sigma)$. We apply the elite strategy. Using $F_1(\sigma)$, we sort the individuals and save the best individual for the next generation. The best individual is evaluated by $F_2(\sigma)$.
Step 4 (Termination):
Let $g \leftarrow g + 1$, go to Step 2, and repeat until the maximum generation g_{max}.

It should be noted that, in this algorithm, the number of evaluations is at most $M_g \times g_{max}$. This number corresponds to computation cost. The brute force requires the much larger number of evaluations $N!$.

4. Numerical experiments

We have applied the algorithm with parameters

$$\begin{array}{l} \text{Conection parameters: } (w_a, w_b, w_c) = (+1, +1, -1) \\ \text{Dimension of input vector: } N = 10 \\ \text{The number of individuals: } M_g = 50 \\ \text{The maximum generation: } g_{max} = 20. \end{array} \tag{11}$$

The algorithm has an initial individual σ_0 in Fig. 3 (a) at generation $g = 0$.

$$\sigma_0 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix} \tag{12}$$

$$F_1(\sigma_0) = \frac{10}{1024}, F_2(\sigma_0) = \frac{40}{1024}. \tag{13}$$

At $g = 10$, we obtain σ_{10} in Fig. 3 (b).

$$\sigma_{10} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 6 & 5 & 2 & 3 & 10 & 7 & 4 & 8 & 1 & 9 \end{pmatrix} \tag{14}$$

$$F_1(\sigma_{10}) = \frac{62}{1024}, F_2(\sigma_{10}) = \frac{98}{1024}. \tag{15}$$

At $g = g_{max}$, we obtain σ_{20} in Fig. 3 (c).

$$\sigma_{20} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 8 & 4 & 10 & 3 & 5 & 6 & 2 & 9 & 7 \end{pmatrix} \tag{16}$$

$$F_1(\sigma_{20}) = \frac{98}{1024}, F_2(\sigma_{20}) = \frac{132}{1024}. \tag{17}$$

$F_1(\sigma)$ for g is shown in Fig. 3 (d).

5. Conclusions

We have studied a simple synthesis method of PBNN. The objective function is period of BPO. The method is based on the genetic algorithm with a mutation operator corresponding to exchange. Executing elementary numerical experiments, the algorithm performance is investigated.

In our future works, we should consider optimization of parameters, multiobjective optimization problem of PBNN, and application to real world problems.

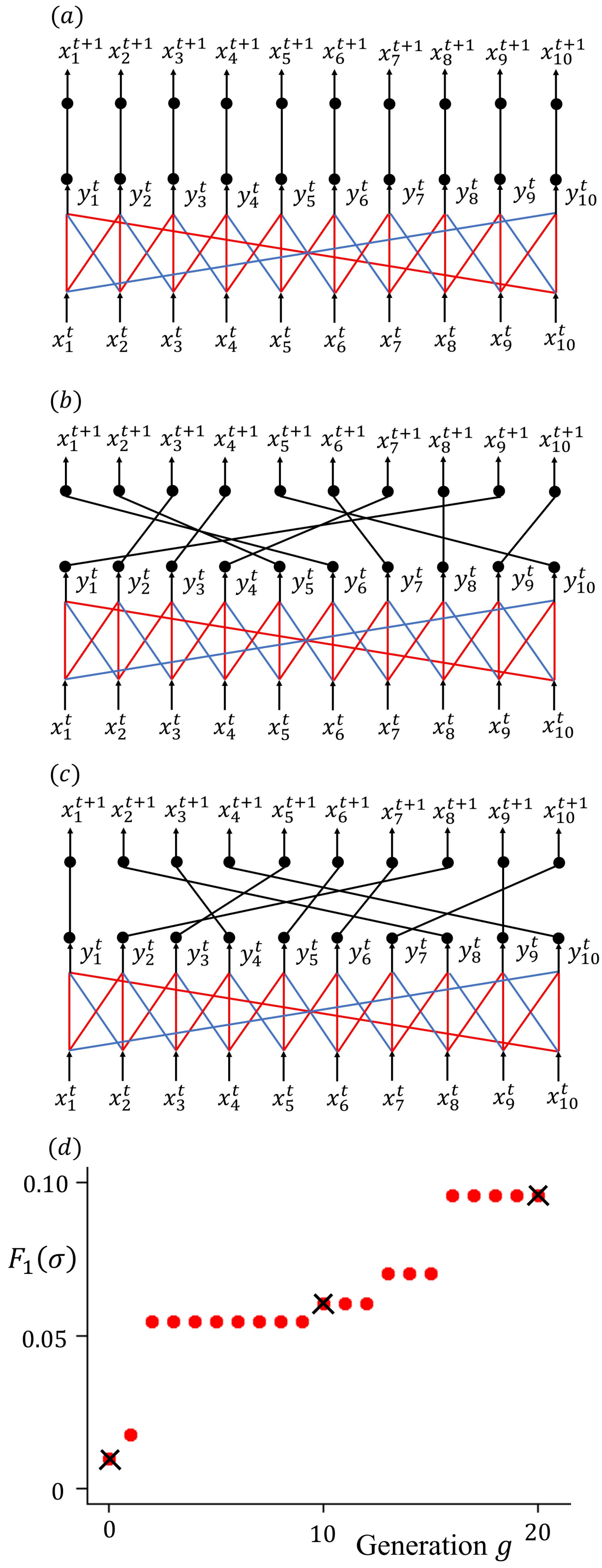

Figure 3: PBNNs and $F_1(\sigma)$ for g. (a) PBNN of σ_0. (b) PBNN of σ_{10}. (c) PBNN of σ_{20}. (d) The maximum of $F_1(\sigma)$ for g.

References

[1] J. J. Hopfield, Neural networks and physical systems with emergent collective computation abilities, Proc. Nat. Acad. Sci., 79, pp. 2554-2558, 1982.

[2] A. N. Michel and J.A. Farrell, Associative memories via artificial neural networks, IEEE Control Systems Magazine, 10, pp. 6-17, 1990.

[3] M. Adachi and K. Aihara, Associative dynamics in a chaotic neural network, Neural Networks, 10, pp. 83-98, 1997.

[4] H. Udagawa, T. Okano and T. Saito, Permutation binary neural networks: analysis of periodic orbits and its applications, Discrete Contin. Dyn. Syst. Ser. B., doi: 10.3934/dcdsb.2022097, 2022.

[5] R. Sato and T. Saito, Stabilization of desired periodic orbits in dynamic binary neural networks, Neurocomputing 248, pp. 19-27, 2017.

[6] S. Anzai, T. Suzuki and T. Saito, Dynamic binary neural networks with time-variant parameters and switching of desired periodic orbits, Neurocomputing 457, pp. 357-364, 2021.

[7] W. Holderbaum, Application of neural network to hybrid systems with binary inputs, IEEE Trans. Neural Netw., 18, pp. 1254-1261, 2007.

[8] M. Lodi, A. L. Shilnikov and M. Storace, Design principles for central pattern generators with preset rhythms, IEEE Trans. Neural Netw. Learn. Syst., 31, pp. 3658-3669, 2020.

[9] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano and A. Hirose, Recent advances in physical reservoir computing: a review, Neural Networks, 115, pp. 100-123, 2019.