Neural Networks 79 (2016) 12–19

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

When are two multi-layer cellular neural networks the same?

Jung-Chao Ban[a], Chih-Hung Chang[b,*]

[a] Department of Applied Mathematics, National Dong Hwa University, Hualien 970003, Taiwan, ROC

[b] Department of Applied Mathematics, National University of Kaohsiung, Kaohsiung 81148, Taiwan, ROC

ARTICLE INFO

Article history:
Received 5 July 2015
Received in revised form 13 March 2016
Accepted 16 March 2016
Available online 28 March 2016

MSC:
primary 34A33
37B10
secondary 11K55
47A35

Keywords:
Multi-layer cellular neural networks
Hidden layers
Sofic shifts
Topological conjugacy
Total amalgamation
Williams classification theorem

ABSTRACT

This paper aims to characterize whether a multi-layer cellular neural network is of deep architecture; namely, when can an n-layer cellular neural network be replaced by an m-layer cellular neural network for $m < n$ yet still preserve the same output phenomena? From a mathematical point of view, such characterization involves investigating whether the topological structure of two (or multiple) layers is conjugate. A decision procedure that addresses the necessary and sufficient condition for the topological conjugacy between two layers in a network is revealed.

1. Introduction

This paper focuses on the following dynamical systems.

$$\begin{cases} \dfrac{d}{dt}x_i^{(n)}(t) &= -x_i^{(n)}(t) + z^{(n)} + \displaystyle\sum_{k\in\mathcal{N}}(a_k^{(n)} f(x_{i+k}^{(n)}(t)) \\ & \quad + b_k^{(n)} f(x_{i+k}^{(n-1)}(t))), \\ & \vdots \\ \dfrac{d}{dt}x_i^{(2)}(t) &= -x_i^{(2)}(t) + z^{(2)} + \displaystyle\sum_{k\in\mathcal{N}}(a_k^{(2)} f(x_{i+k}^{(2)}(t)) \\ & \quad + b_k^{(2)} f(x_{i+k}^{(1)}(t))), \\ \dfrac{d}{dt}x_i^{(1)}(t) &= -x_i^{(1)}(t) + z^{(1)} + \displaystyle\sum_{k\in\mathcal{N}} a_k^{(1)} f(x_{i+k}^{(1)}(t)), \end{cases} \qquad (1)$$

for some integer $n \geq 2$, $i \in \mathbb{N}$, and $t \geq 0$. Herein $x_i^{(\ell)}(t) = 0$ for $1 \leq \ell \leq n$ and $t \geq 0$ provided $i \leq 0$. The so-called *neighborhood* $\mathcal{N}$ is a finite subset of integers $\mathbb{Z}$; the output function

$$f(x) = \frac{1}{2}(|x+1| - |x-1|) \qquad (2)$$

is a piecewise linear map. $\mathbb{A} = [A^{(1)}, \ldots, A^{(n)}]$ and $\mathbb{B} = [B^{(2)}, \ldots, B^{(n)}]$ are the feedback and controlling templates, respectively, where $A^{(j)} = [a_k^{(j)}]_{k\in\mathcal{N}}$, $B^{(l)} = [b_k^{(l)}]_{k\in\mathcal{N}}$ for $1 \leq j \leq n$, $2 \leq l \leq n$; $\mathbb{z} = [z^{(1)}, \ldots, z^{(n)}]$ is the threshold. The template $\mathbb{T}$ of (1) consists of the feedback and controlling templates and the threshold, namely, $\mathbb{T} = [\mathbb{A}, \mathbb{B}, \mathbb{z}]$. Note that (1) are standard cellular neural networks (CNNs) if we let $n = 1$; in this case we call them *single layer* CNNs. The main reason one chooses (2) to be the output function for (1) is the application of the pattern recognition and image processing (Chua & Yang, 1988a, 1988b).

(1) are called *multi-layer cellular neural networks* (MCNNs, Chua & Shi, 1990) for $n \geq 2$. For the last few decades, MCNNs have received considerable attention due to the fact that they have been successfully applied to many areas such as signal propagation between neurons and image processing (Chua & Yang, 1988a; Crounse & Chua, 1995; Murugesh, 2010; Yang, Nishio, & Ushida, 2001, 2002), pattern recognition (Chua & Roska, 2002; Crounse, Roska, & Chua, 1993; Peng, Zhang, & Liao, 2009), CMOS realization (Carmona, Jimenez-Garrido, Dominguez-Castro, Espejo, & Rodriguez-Vazquez, 2002; Xavier-de Souza, Yalcin, Suykens, & Vandewalle, 2004), VLSI implementation (Chua & Shi, 1991), and self-organization phenomena (Arena, Baglio, Fortuna, & Manganaro, 1998). The sufficient conditions for the complete stability of (1) for $n \geq 1$ can be found in Li (2009), Paolo-Civalleri and Gilli

[*] Corresponding author.
E-mail address: chchang@nuk.edu.tw (C.-H. Chang).

http://dx.doi.org/10.1016/j.neunet.2016.03.005
0893-6080/© 2016 Elsevier Ltd. All rights reserved.

(1999), Savaci and Vandewalle (1992), Török and Roska (2004), Wu and Chua (1997), Xu, Pi, Cao, and Zhong (2007) and Zou and Nossek (1991).

Some kind of stationary solution for (1) is essential, namely, the *mosaic solution*, due to the wide range of complete stability in the parameter space and the application to image processing. For single layer CNNs, a *mosaic solution* $\bar{x}$ is a stationary solution of (1) which satisfies $|\bar{x}_i| \geq 1$ while its corresponding pattern $\bar{y} = (\bar{y}_i) = (f(\bar{x}_i))$ is called a *mosaic output pattern*. Since the output function (2) is a piecewise linear function with $f(x) = 1$ (resp. -1) if $x \geq 1$ (resp. $x \leq -1$), the output of a mosaic solution $\bar{x} = (\bar{x}_i)_{i\in\mathbb{N}}$ is an element in $\Sigma = \{-1, +1\}^{\mathbb{N}}$, and that is why we call it a *pattern*.

Given an n-layer CNN with $n \geq 2$, a stationary solution $\mathbf{x} = (x_i^{(1)}, \ldots, x_i^{(n)})_{i\in\mathbb{Z}} \in \mathbb{R}^{\infty\times n}$ of (1) is called *mosaic* if $|x_i^{(k)}| > 1$ for $1 \leq k \leq N$, $i \in \mathbb{Z}$. The output $\mathbf{y} = (y_i^{(1)} \cdots y_i^{(n)})_{i\in\mathbb{Z}} \in \{-1, 1\}^{\infty\times n}$ of a mosaic solution is called a pattern, where $y_i^{(k)} = f(x_i^{(k)})$. The *solution space* $\mathbf{Y}$ of (1) stores the mosaic patterns $\mathbf{y}$, and the *output space* $\mathbf{Y}^{(n)}$ of (1) is the collection of the output patterns in $\mathbf{Y}$, or, more precisely,

$$\mathbf{Y}^{(n)} = \{(y_i^{(n)})_{i\in\mathbb{Z}} : (y_i^{(1)} \cdots y_i^{(n)})_{i\in\mathbb{Z}} \in \mathbf{Y}\}.$$

There are two important problems for a given MCNN: (i) How can we characterize whether an MCNN has deep architecture?[1] and (ii) How can we train a deep architecture MCNN? Those two problems are closely related to AI design, since deep architecture may be required for any such design. As a general reference, readers are referred to Bengio (2009) for more details.

This work is intended as an attempt to answer problem (i) and study the learning algorithm of (ii). First we try to formulate (i) and (ii) mathematically. Let Σ be a shift space; that is, Σ is a subset of $\mathcal{A}^{\mathbb{N}}$ for some finite set $\mathcal{A}$. $C(\Sigma, \Sigma)$ denotes the collection of maps from Σ to Σ. A map $\tau \in C(\Sigma, \Sigma)$ is called a *factor* (resp. an *embedding*) if it is onto (resp. one-to-one). τ is called a *conjugacy* if it is both a factor and an embedding. Then the above problem is formulated as follows.

Problem 1. Given an n-layer CNN (1) with $n \geq 2$.

(1) Corresponding to a given $\mathbf{Y}^{(1)}$, what kind of $\mathbf{Y}^{(n)}$ can be shown? To be precise, what is the symbolic space $\mathbf{Y}^{(n)}$ according to $\mathbf{Y}^{(1)}$?
(2) Given $\mathbf{Y}^{(i)}$ and $\mathbf{Y}^{(j)}$ for $1 \leq i \neq j \leq n$, does there exist a conjugacy τ between them?

Problem 1-(1) is closely related to the learning algorithm of MCNNs since one can figure out which kind of output solutions of (1) can be produced from a given input. It is also worth pointing out that if τ in Problem 1-(2) exists and $i < j$, then one can merge the ith layer to the jth layer to form one layer since conjugacy ensures that the phenomena exhibited by these two layers are **dynamically the same**. By continuing this process one would obtain a new MCNN such that each layer completely performs a different function between the other layers (thus each layer cannot be removed). Thus, one can characterize the depth of such an MCNN. In Ban, Chang, and Lin (2012), Ban and Chang provided a necessary and sufficient condition for determining whether $\mathbf{Y}^{(i)}$ and $\mathbf{Y}^{(j)}$ are conjugated for some $1 \leq i < j \leq n$; in this case, an n-layer CNN can be replaced by an $(n - j + i)$-layer CNN. Their criterion only works for the case where the symbolic transition matrices of $\mathbf{Y}^{(i)}$ and $\mathbf{Y}^{(j)}$ are both right-resolving (defined later). Later on, Chang (2015) obtained a necessary and sufficient condition for determining whether a multi-layer neural network can be reduced to one with fewer layers without the assumption in Ban et al. (2012); instead of right-resolving of the symbolic transition matrix, the criterion proposed in Chang (2015) hinges on the existence of the so-called factor-like matrix. This work proposes an algorithm, which provides a necessary and sufficient condition for determining the depth of an MCNN, for answering Problem 1-(2); the main contribution of the proposed algorithm is to demonstrate a workable criterion which can be realized by programming, such that the result can be derived in seconds.

Meanwhile, recall that the well-known Hopfield neural networks (Hopfield, 1982, 1984) can be formulated as

$$\begin{cases} C_i\dot{x}_i &= -\dfrac{x_i}{R_i} + \displaystyle\sum_{j=1}^{N} \omega_{ij}y_j + \theta_i, \\ y_i &= g_i(\lambda_i x_i), \end{cases} \quad \text{for } i = 1, \ldots, N, \tag{3}$$

where x_i stands for the state of neuron i with each activation function g_i being sigmoid. It should be highlighted that if $n = 1$, then the main difference between (1) and (3) is their output functions and the weights between neurons. Hence the investigation of the MCNNs in this paper can be extended to Hopfield neural networks (HNNs) with some modification. Roughly speaking, MCNNs are hybrids between conventional neural networks, such as HNNs, and continuous automata; the behavior of the overall systems of both MCNNs and HNNs is driven by the weights of the processing unit's linear interconnection. The major discriminator is that the connections between MCNN processors are local, while all the HNN processors are fully interconnected. Beyond that, our methodology can also be applied to determine whether two stable multi-layer neural networks are topologically conjugate, in other words, whether or not two different neural networks, such as CNN and HNN, recognize the same images up to the change of color. The related work is in preparation.

As stated above, one of the applications of our algorithm is to see whether two completely stable networks are likely to be conjugated, that is, to determine if two networks exhibit the same dynamical behavior eventually. More precisely, the output of a network converges to patterns whenever it is completely stable; in this case, the output patterns are realized symbolically and can be analyzed via our algorithm and theorem. To the best of our knowledge, there is no such elucidation investigating multi-layer neural networks from this perspective. Furthermore, in Rakkiyappan, Chandrasekar, Lakshmanan, and Park (2014), Rakkiyappan, Zhu, and Chandrasekar (2014), the authors demonstrated the asymptotic stability of some types of stochastic neural networks with time-dependent delays and Markovian jump parameters. A natural question is to ask under what conditions a stochastic neural network with delays possesses similar behavior to a multi-layer cellular neural network, up to conjugacy. It is also interesting to elaborate the cost to simulate a stochastic neural network with delays by a deterministic multi-layer network; more precisely, to answer the question of how many layers we need, for instance, for a multi-layer cellular neural network to exhibit the dynamical behavior of a stochastic Cohen–Grossberg neural network with delays. The related work remains in preparation.

The rest of this paper is organized as follows. In Section 2, we consider the simplest case, i.e., $n = 2$ and elucidate how to produce the symbolic space of $\mathbf{Y}^{(2)}$ according to a given $\mathbf{Y}^{(1)}$. This method can be easily extended to the general case where $n > 2$. The so-called *symbolic transition matrices* $\mathbf{S}^{(i)}$ of $\mathbf{Y}^{(i)}$, for $i = 1, 2$, are defined therein, which is helpful for the study of Problem-(2). We prove that $\mathbf{S}^{(i)}$ is the complete invariant for the existence of conjugacy between $\mathbf{Y}^{(1)}$ and $\mathbf{Y}^{(2)}$ (Theorem 2.1). This gives the affirmative answer for Problem-(2) of $n = 2$. Finally we extend this result to the general case for arbitrary $n \geq 2$ (Theorem 3.1) in Section 3, and further discussion and our conclusion are addressed in Section 4.

[1] Deep architectures are composed of multiple levels of nonlinear operations, such as in neural networks with many hidden layers or in complicated propositional formulae re-using many sub-formulae.

2. Topological conjugacy for two-layer cellular neural networks

This section elucidates the conjugacy between the output and hidden spaces of two-layer cellular networks to clarify the idea of the methodology for the general MCNNs, which is explained in the next section.

To make the discussion easier to understand, we consider the *simplified two-layer neural networks* (STCNNs) proposed as

$$\begin{cases} \dfrac{dx_i^{(2)}}{dt} = -x_i^{(2)} + a^{(2)}y_i^{(2)} + a_r^{(2)}y_{i+1}^{(2)} + b^{(2)}u_i^{(2)} \\ \qquad + b_r^{(2)}u_{i+1}^{(2)} + z^{(2)}, \\ \dfrac{dx_i^{(1)}}{dt} = -x_i^{(1)} + a^{(1)}y_i^{(1)} + a_r^{(1)}y_{i+1}^{(1)} + z^{(1)}, \end{cases} \tag{4}$$

where $u_i^{(2)} = y_i^{(1)}$ for all $i \in \mathbb{N}$. Suppose $\mathbf{y} = \begin{pmatrix} \cdots y_{-1}^{(2)}y_0^{(2)}y_1^{(2)}\cdots \\ \cdots y_{-1}^{(1)}y_0^{(1)}y_1^{(1)}\cdots \end{pmatrix}$ is a mosaic pattern. For $i \in \mathbb{N}$, $y_i^{(1)} = 1$ if and only if

$$a^{(1)} + z^{(1)} - 1 > -a_r^{(1)}y_{i+1}^{(1)}. \tag{5}$$

Similarly, $y_i^{(1)} = -1$ if and only if

$$a^{(1)} - z^{(1)} - 1 > a_r^{(1)}y_{i+1}^{(1)}. \tag{6}$$

The same argument asserts

$$a^{(2)} + z^{(2)} - 1 > -a_r^{(2)}y_{i+1}^{(2)} - (b^{(2)}u_i^{(2)} + b_r^{(2)}u_{i+1}^{(2)}), \tag{7}$$

and

$$a^{(2)} - z^{(2)} - 1 > a_r^{(2)}y_{i+1}^{(2)} + (b^{(2)}u_i^{(2)} + b_r^{(2)}u_{i+1}^{(2)}) \tag{8}$$

are the necessary and sufficient conditions for $y_i^{(2)} = -1$ and $y_i^{(2)} = 1$, respectively. Note that the quantity $u_i^{(2)}$ in (7) and (8) satisfies $|u_i^{(2)}| = 1$ for each i. Define $\xi_1 : \{-1, 1\} \to \mathbb{R}$ and $\xi_2 : \{-1, 1\}^3 \to \mathbb{R}$ by

$$\xi_1(w) = a_r^{(1)}w, \qquad \xi_2(w_1, w_2, w_3) = a_r^{(2)}w_1 + b^{(2)}w_2 + b_r^{(2)}w_3.$$

Set

$$\mathcal{B}^{(1)} = \left\{ \boxed{y^{(1)}y_r^{(1)}} : y^{(1)}, y_r^{(1)} \in \{-1, 1\}\text{satisfy (5), (6)} \right\},$$

$$\mathcal{B}^{(2)} = \left\{ \boxed{\begin{matrix} y^{(2)}y_r^{(2)} \\ u^{(2)}u_r^{(2)} \end{matrix}} : y^{(2)}, y_r^{(2)}, u^{(2)}, u_r^{(2)} \in \{-1, 1\} \right.$$
$$\left. \text{satisfy (7), (8)} \right\}.$$

That is,

$$\boxed{y^{(1)}y_r^{(1)}} \in \mathcal{B}^{(1)} \Leftrightarrow \begin{cases} a^{(1)} + z^{(1)} - 1 > -\xi_1(y_r^{(1)}), \\ \quad \text{if } y^{(1)} = 1; \\ a^{(1)} - z^{(1)} - 1 > \xi_1(y_r^{(1)}), \\ \quad \text{if } y^{(1)} = -1. \end{cases}$$

$$\boxed{\begin{matrix} y^{(2)}y_r^{(2)} \\ u^{(2)}u_r^{(2)} \end{matrix}} \in \mathcal{B}^{(2)} \Leftrightarrow \begin{cases} a^{(2)} + z^{(2)} - 1 > -\xi_2(y_r^{(2)}, u^{(2)}, u_r^{(2)}), \\ \quad \text{if } y^{(2)} = 1; \\ a^{(2)} - z^{(2)} - 1 > \xi_2(y_r^{(2)}, u^{(2)}, u_r^{(2)}), \\ \quad \text{if } y^{(2)} = -1. \end{cases}$$

The basic set of admissible local patterns $\mathcal{B}$ of (4) is then

$$\mathcal{B} = \left\{ \boxed{\begin{matrix} yy_r \\ uu_r \end{matrix}} : \boxed{\begin{matrix} yy_r \\ uu_r \end{matrix}} \in \mathcal{B}^{(2)} \text{ and } \boxed{uu_r} \in \mathcal{B}^{(1)} \right\}.$$

The basic set of admissible local patterns plays an essential role for investigating the structure of the solution space $\mathbf{Y}$ of STCNNs. Substitute mosaic patterns -1 and 1 as symbols $-$ and $+$, respectively. Define the ordering matrix of $\{-, +\}^{\mathbb{Z}_{2\times 2}}$ by

$$\mathbb{X}_2 = \begin{array}{c|cccc} & \boxed{\begin{matrix}-\\-\end{matrix}} & \boxed{\begin{matrix}-\\+\end{matrix}} & \boxed{\begin{matrix}+\\-\end{matrix}} & \boxed{\begin{matrix}+\\+\end{matrix}} \\ \hline \boxed{\begin{matrix}-\\-\end{matrix}} & \boxed{\begin{matrix}--\\--\end{matrix}} & \boxed{\begin{matrix}--\\-+\end{matrix}} & \boxed{\begin{matrix}-+\\--\end{matrix}} & \boxed{\begin{matrix}-+\\-+\end{matrix}} \\ \boxed{\begin{matrix}-\\+\end{matrix}} & \boxed{\begin{matrix}--\\+-\end{matrix}} & \boxed{\begin{matrix}--\\++\end{matrix}} & \boxed{\begin{matrix}-+\\+-\end{matrix}} & \boxed{\begin{matrix}-+\\++\end{matrix}} \\ \boxed{\begin{matrix}+\\-\end{matrix}} & \boxed{\begin{matrix}+-\\--\end{matrix}} & \boxed{\begin{matrix}+-\\-+\end{matrix}} & \boxed{\begin{matrix}++\\--\end{matrix}} & \boxed{\begin{matrix}++\\-+\end{matrix}} \\ \boxed{\begin{matrix}+\\+\end{matrix}} & \boxed{\begin{matrix}+-\\+-\end{matrix}} & \boxed{\begin{matrix}+-\\++\end{matrix}} & \boxed{\begin{matrix}++\\+-\end{matrix}} & \boxed{\begin{matrix}++\\++\end{matrix}} \end{array} := (x_{pq})_{1\le p,q\le 4}.$$

Notably, each entry in $\mathbb{X}$ is a 2×2 pattern since $\mathcal{B}$ consists of 2×2 local patterns. Suppose that $\mathcal{B}$ is given. The *transition matrix* $\mathbf{T}_{\mathcal{B}} \equiv \mathbf{T} = (\mathbf{T}(p, q)) \in \mathcal{M}_4(\{0, 1\})$ is defined by

$$\mathbf{T}(p, q) = \begin{cases} 1, & \text{if } x_{pq} \in \mathcal{B}; \\ 0, & \text{otherwise} \end{cases}$$

where $1 \le p, q \le 4$. Let $\mathcal{L} = \{\alpha_1, \alpha_2.\alpha_3, \alpha_4\}$, where

$$\alpha_1 := --, \qquad \alpha_2 = -+, \qquad \alpha_3 = +-, \qquad \alpha_4 = ++\,.$$

For $i = 1, 2$, define the *symbolic transition matrix* $\mathbf{S}_{\mathcal{B}}^{(i)} \equiv \mathbf{S}^{(i)} = (\mathbf{S}^{(i)}(p, q))$ for the ith layer as

$$\mathbf{S}^{(i)}(p, q) = \begin{cases} \alpha_{k_i}, & \text{if } \mathbf{T}(p, q) = 1 \text{ and } x_{pq} = \alpha_{k_2} \diamond \alpha_{k_1}; \\ \varnothing, & \text{otherwise.} \end{cases}$$

Herein $\diamond$ infers piling one pattern above another one; more specifically,

$$t_1t_2 \ldots t_k \diamond t'_1t'_2 \ldots t'_k := \boxed{\begin{matrix} t_1t_2 \ldots t_k \\ t'_1t'_2 \ldots t'_k \end{matrix}}.$$

The structure of the *output space*

$$\mathbf{Y}^{(2)} = \{(y_i)_{i\in\mathbb{N}} : \text{ there exists } (u_i)_{i\in\mathbb{N}} \text{ such that } (y_i \diamond u_i)_{i\in\mathbb{N}} \in \mathbf{Y}\}$$

is determined by $\mathbf{S}^{(2)}$. Similarly, we can investigate the topological property of the *hidden space*

$$\mathbf{Y}^{(1)} = \{(u_i)_{i\in\mathbb{N}} : \text{ there exists } (y_i)_{i\in\mathbb{N}} \text{ such that } (y_i \diamond u_i)_{i\in\mathbb{N}} \in \mathbf{Y}\}$$

via the symbolic transition matrix $\mathbf{S}^{(1)}$.

The solution space of a STCNN is a subspace of $\{-, +\}^{\infty\times 2}$, and the output/hidden space, which is a subspace of $\{-, +\}^{\infty}$, is a projection of the solution space. Since each STCNN is a locally coupled system, it is seen that every element in the output space (resp. hidden space) can be revealed via the symbolic transition matrix $\mathbf{S}^{(2)}$ (resp. $\mathbf{S}^{(1)}$). More precisely, for each $y = (y_i)_{i\in\mathbb{N}} \in \mathbf{Y}^{(2)}$ and $n \ge 2$, the truncated pattern $(y_i)_{i=1}^n$ is stored in $(\mathbf{S}^{(2)})^{n-1}$, wherein the product of symbolic transition matrices indicates combining two patterns whenever the terminal symbol of the first pattern and the initial pattern of the other are coincident. The symbolic transition matrix, briefly speaking, is used for elucidating the properties of the solution, output, and hidden spaces of an MCNN. Example 2.1 provides an intuitive observation of how a symbolic transition matrix assists in the investigation of the output space of an MCNN; readers are referred to Ban, Chang, Lin, and Lin (2009) for more details.

Example 2.1. Suppose the templates of a STCNN are given by

$$[a^{(1)}, a_r^{(1)}, z^{(1)}] = [2.9, 1.7, 0.1]$$
$$[a^{(2)}, a_r^{(2)}, b^{(2)}, b_r^{(2)}, z^{(2)}] = [-0.3, -1.2, 0.7, 2.3, 0.9].$$

Evidently, the solution space $\mathbf{Y}$ is generated by the set of local patterns

$$\mathcal{B} = \left\{ \boxed{\begin{matrix} -+ \\ -- \end{matrix}}, \boxed{\begin{matrix} -+ \\ +- \end{matrix}}, \boxed{\begin{matrix} +- \\ -+ \end{matrix}}, \boxed{\begin{matrix} +- \\ ++ \end{matrix}}, \boxed{\begin{matrix} ++ \\ -+ \end{matrix}}, \boxed{\begin{matrix} ++ \\ ++ \end{matrix}} \right\}$$

and the transition matrix $\mathbf{T}$ is

$$\mathbf{T} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

Let us focus on the hidden space $\mathbf{Y}^{(1)}$. The symbolic transition matrix of $\mathbf{Y}^{(1)}$ is

$$\mathbf{S}^{(2)} = \begin{pmatrix} \varnothing & \varnothing & \alpha_1 & \varnothing \\ \varnothing & \varnothing & \alpha_3 & \varnothing \\ \varnothing & \alpha_2 & \varnothing & \alpha_2 \\ \varnothing & \alpha_4 & \varnothing & \alpha_4 \end{pmatrix}.$$

To see that $(\mathbf{S}^{(1)})^{n-1}$ stores those truncated output patterns of length n, we consider the case where $n = 3$. It can be seen from $\mathcal{B}$ that the truncated hidden patterns of length 3 are

$$--+, \qquad +-+, \qquad -+-, \qquad ++-, \qquad -++, \qquad +++.$$

Meanwhile,

$$(\mathbf{S}^{(1)})^2 = \begin{pmatrix} \varnothing & \alpha_1\alpha_2 & \varnothing & \alpha_1\alpha_2 \\ \varnothing & \alpha_2\alpha_3 & \varnothing & \alpha_3\alpha_2 \\ \varnothing & \alpha_2\alpha_4 & \alpha_2\alpha_3 & \alpha_2\alpha_4 \\ \varnothing & \alpha_4\alpha_4 & \alpha_4\alpha_3 & \alpha_4\alpha_4 \end{pmatrix}$$
$$= \begin{pmatrix} \varnothing & --+ & \varnothing & --+ \\ \varnothing & -+- & \varnothing & +-+ \\ \varnothing & -++ & -+- & -++ \\ \varnothing & +++ & ++- & +++ \end{pmatrix}$$

stores all the truncated hidden patterns above.

For a given symbolic transition matrix $\mathbf{S}$, we construct the *symbolic total amalgamation matrix* $\mathbf{S}_{\text{tot}}$ as follows:

Algorithm 2.1 (*Total Amalgamation Algorithm*)**.** Given a symbolic transition matrix $\mathbf{S}$ that has no empty rows or columns.

Step 1. If there are no two identical rows or columns in $\mathbf{S}$, then $\mathbf{S}_{\text{tot}} = \mathbf{S}$.
Step 2. Delete the jth row (respective column) if it is identical to the ith row (respective column) for some $i < j$.
Step 3. Add the jth column (respective row) to the ith column (respective row), then delete the jth column (respective row).
Step 4. Repeat Steps 1–3.

The total amalgamation algorithm is meant to reduce the complexity of a symbolic transition matrix and, in the meantime, remains capable of presenting the structure of its corresponding space. Example 2.2 shows how the algorithm reduces the dimension of a symbolic transition matrix, and Proposition 2.1 demonstrates that the reduced symbolic transition matrix still presents the same space as the original matrix does.

Example 2.2. Suppose $\mathbf{Y}$ is a sofic shift with a labeled graph presentation $\mathcal{G}$. The symbolic transition matrix $\mathbf{S}$ of $\mathbf{Y}$ is expressed as

$$\mathbf{S} = \begin{pmatrix} a & b & c \\ \varnothing & d & e \\ a & b & c \end{pmatrix}$$

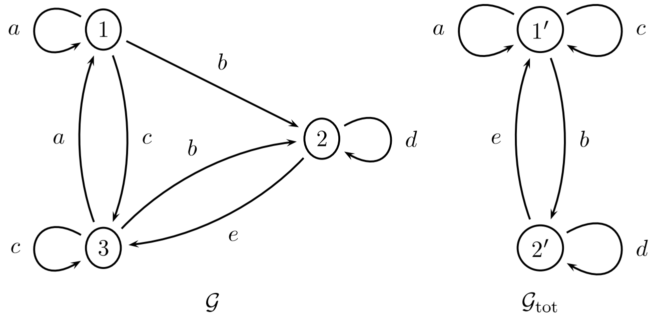

Fig. 1. A labeled graph $\mathcal{G}$ whose symbolic transition matrix having two identical rows gives rise to the inference that the outgoing edges of two vertices are the same. The amalgamation labeled graph $\mathcal{G}_{\text{tot}}$ of $\mathcal{G}$ is a labeled graph that merges such two vertices.

and has two identical rows. Applying the total amalgamation algorithm to $\mathbf{S}$ we have

$$\mathbf{S} = \begin{pmatrix} a & b & c \\ \varnothing & d & e \\ a & b & c \end{pmatrix} \rightsquigarrow \begin{pmatrix} a & b & c \\ \varnothing & d & e \end{pmatrix} \rightsquigarrow \begin{pmatrix} a+c & b & c \\ e & d & e \end{pmatrix}$$
$$\rightsquigarrow \begin{pmatrix} a+c & b \\ e & d \end{pmatrix} = \mathbf{S}_{\text{tot}}.$$

The labeled graph presentation $\mathcal{G}_{\text{tot}}$ of $\mathbf{S}_{\text{tot}}$ is shown in Fig. 1.

Proposition 2.1. *Suppose* $\mathbf{Y}$ *is a sofic shift with the symbolic transition matrix* $\mathbf{S}$ *having no empty rows or columns. Let* $\mathbf{S}_{\text{tot}}$ *be the symbolic transition matrix obtained from applying the total amalgamation algorithm to* $\mathbf{S}$*, and let* $\mathbf{Y}_{\text{tot}}$ *be the sofic shift derived from* $\mathbf{S}_{\text{tot}}$*. Then* $\mathbf{Y}$ *is coincident with* $\mathbf{Y}_{\text{tot}}$*.*

Proof. Without loss of generality, we assume that $\mathbf{S}$ has exactly two identical columns and $\mathbf{S}_{\text{tot}}$ is obtained by removing one of these two columns and replacing two corresponding rows by their summation, which has no identical columns. Let $\mathcal{G} = (G, \mathcal{L})$ be a labeled graph presentation of $\mathbf{S}$ with an underlying directed graph $G = (\mathcal{V}, \mathcal{E})$ consisting of vertices set $\mathcal{V}$ and edges set $\mathcal{E}$. More precisely, $\mathcal{V} = \{v_i\}_{i=1}^r$ with r being the dimension of $\mathbf{S}$, $e = (v_i, v_j) \in \mathcal{E}$ if and only if $\mathbf{S}(i,j) \neq \varnothing$, and $\mathcal{L} : \mathcal{E} \to \mathcal{A}$ is the *labeling* defined as $\mathcal{L}(e) = \mathbf{a}$ if and only if $e = (v_i, v_j) \in \mathcal{E}$ and $\mathbf{S}(i,j) = \mathbf{a}$. It can be seen that the kth and the ℓth columns of $\mathbf{S}$ are coincident which infers $\mathcal{V}^k = \mathcal{V}^\ell$ and $\mathcal{L}(v, v_k) = \mathcal{L}(v, v_\ell)$ for all $v \in \mathcal{V}^k$, where

$$\mathcal{V}^k = \{v \in \mathcal{V} : (v, v_k) \in \mathcal{E}\}$$

consists of those vertices that are the initial states of edges ending at v_k.

Since $\mathcal{V}^k = \mathcal{V}^\ell$, deleting the ℓth column of $\mathbf{S}$ means that the vertex v_ℓ is eliminated and those edges with terminal state v_ℓ are merged with appropriately corresponding edges with terminal state v_k in the sense $\mathcal{L}(v, v_k) = \mathcal{L}(v, v_\ell)$ for all $v \in \mathcal{V}^k$. To complete the mergence of v_k and v_ℓ, we replace

$$\mathcal{E}_k = \{e \in \mathcal{E} : e = (v_k, v) \text{ for some } v \in \mathcal{V}\}$$

by $\mathcal{E}_k \cup \mathcal{E}_\ell$. Namely, v_k is substituted as the initial state of those edges starting from v_ℓ. This operation is the realization of Step 3 in the total amalgamation algorithm on the labeled graph presentation $\mathcal{G}$ of $\mathbf{S}$, with a new labeled graph $\mathcal{G}_{\text{tot}}$ thereby constructed.

For each $n \in \mathbb{N}$, let $P_n(\mathcal{G})$ and $P_n(\mathcal{G}_{\text{tot}})$ denote the set of paths of length n in $\mathcal{G}$ and $\mathcal{G}_{\text{tot}}$, respectively. To be specific,

$$P_n(\mathcal{G}) = \{w_1 w_2 \cdots w_n : w_i = \mathcal{L}(v_{k_i}, v_{k_{i+1}}),$$
$$(v_{k_i}, v_{k_{i+1}}) \in \mathcal{E} \text{ for all } i\}.$$

Obviously, every path w of length n in $\mathcal{G}$ is a path of length n in $\mathcal{G}_{\text{tot}}$. Conversely, suppose $\overline{w} \in P_n(\mathcal{G}_{\text{tot}})$. If $\overline{w}$ does not pass through $\overline{v}_k$ (herein we use the notation $\overline{v}_k$ to refer to the new vertex that is the mergence of v_k and v_ℓ in the original graph), then $\overline{w} \in P_n(\mathcal{G})$. Otherwise, assume that the initial state of $\overline{w}$ is $\overline{v}_k$ and $\overline{v}_k$ is not treated as an intermediate state of $\overline{w}$. Suppose $\overline{w}$ is not a length-n path in $\mathcal{G}$ that starts from v_k. Let m be the smallest index such that $w_m \neq \mathcal{L}(v_k, v)$ for all $v \in \mathcal{V}$. The definition of $\mathcal{G}_{\text{tot}}$ demonstrates that $w_m = \mathcal{L}(v_\ell, v)$ for some $v \in \mathcal{V}$, and hence $\overline{w}$ is a path with initial state v_ℓ. It follows that $P_n(\mathcal{G}_{\text{tot}}) \subseteq P_n(\mathcal{G})$.

Similarly, we can demonstrate that every path of length n in $\mathcal{G}_{\text{tot}}$ is also a path of the same length in $\mathcal{G}$. Since $\mathbf{Y}$ and $\mathbf{Y}_{\text{tot}}$ consist of paths of infinite length in $\mathbf{S}$ and $\mathbf{S}_{\text{tot}}$, respectively, it follows $\mathbf{Y} = \mathbf{Y}_{\text{tot}}$. This completes the proof. □

Given an $n \times n$ symbolic transition matrix $\mathbf{S}$, we say that $\mathbf{S}$ is *right-resolving* if, for $1 \leq p \leq n$, $\mathbf{S}(p, q_1) \neq \mathbf{S}(p, q_2)$ for $1 \leq q_1, q_2 \leq n$, except for empty entries. The graphical meaning of a right-resolving symbolic transition matrix is that, in its related labeled graph presentation, no two edges start from one vertex carrying the same symbol. For two symbolic transition matrices $\mathbf{S}_1$ and $\mathbf{S}_2$, we say that $\mathbf{S}_1$ is equivalent to $\mathbf{S}_2$ by declaring $\mathbf{S}_1 \approx \mathbf{S}_2$ if there exists a one-to-one correspondence between the symbols of $\mathbf{S}_1$ and $\mathbf{S}_2$ such that $\mathbf{S}_1$ equals $\mathbf{S}_2$ modulo such bijection of their underlying symbolic monomials. For example,

$$\begin{pmatrix} \varnothing & b \\ b+c & 2a \end{pmatrix} \approx \begin{pmatrix} \varnothing & a \\ a+d & 2e \end{pmatrix} \approx \begin{pmatrix} \varnothing & \alpha \\ \alpha+tt & 2\beta \end{pmatrix}.$$

It follows immediately that two spaces represented by equivalent symbolic transition matrices are topologically conjugate. Suppose $\mathbf{Y}^{(1)}$ and $\mathbf{Y}^{(2)}$ are the hidden and output spaces of a STCNN with symbolic transition matrices $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$, respectively. Theorem 2.1 proposes a decision procedure for determining whether $\mathbf{Y}^{(1)}$ is conjugate to $\mathbf{Y}^{(2)}$.

Theorem 2.1. *Suppose $\mathbf{S}_{\text{tot}}^{(1)}$ and $\mathbf{S}_{\text{tot}}^{(2)}$ are both right-resolving without empty rows/columns. Then $\mathbf{Y}^{(1)} \cong \mathbf{Y}^{(2)}$ if and only if $\mathbf{S}_{\text{tot}}^{(1)} \approx P^{-1}\mathbf{S}_{\text{tot}}^{(2)}P$ for some permutation matrix P. In general, there exist symbolic transition matrices $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(2)}$ such that $\mathbf{Y}^{(1)} \cong \mathbf{Y}^{(2)}$ if and only if $\mathbf{H}_{\text{tot}}^{(1)} \approx P^{-1}\mathbf{H}_{\text{tot}}^{(2)}P$ for some permutation matrix P.*

Proof. First we consider the case where $\mathbf{S}_{\text{tot}}^{(1)}$ and $\mathbf{S}_{\text{tot}}^{(2)}$ are both right-resolving. Let $\mathbf{T}_{\text{tot}}^{(1)}$ and $\mathbf{T}_{\text{tot}}^{(2)}$ be the incidence matrices of $\mathbf{S}_{\text{tot}}^{(1)}$ and $\mathbf{S}_{\text{tot}}^{(2)}$, respectively. Herein the incidence matrix of a symbolic transition matrix is obtained by replacing every nonempty symbol with 1 and replacing empty symbols with 0. For instance, the incidence matrix of $\left(\begin{smallmatrix} a & a+c \\ b & \varnothing \end{smallmatrix}\right)$ is $\left(\begin{smallmatrix} 1 & 2 \\ 1 & 0 \end{smallmatrix}\right)$.

Let $G^{(1)}$ and $G^{(2)}$ be the graph presentation of $\mathbf{T}_{\text{tot}}^{(1)}$ and $\mathbf{T}_{\text{tot}}^{(2)}$, respectively, and let $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ be the space consisting of those infinite paths in $G^{(1)}$ and $G^{(2)}$, respectively. Then $\mathbf{X}^{(i)}$ is the canonical right-resolving cover of $\mathbf{Y}^{(i)}$ for $i = 1, 2$ (cf. Lind & Marcus, 1995). Suppose $\phi^{(1)} : \mathbf{X}^{(1)} \to \mathbf{Y}^{(1)}$ and $\phi^{(2)} : \mathbf{X}^{(2)} \to \mathbf{Y}^{(2)}$ are such two covers. Fujiwara demonstrated that $\mathbf{Y}^{(1)}$ and $\mathbf{Y}^{(2)}$ are topologically conjugate if and only if there exists some topological conjugacy $\psi : \mathbf{X}^{(1)} \to \mathbf{X}^{(2)}$ which takes the quotient relation of $\phi^{(1)}$ to the quotient relation of $\phi^{(2)}$ (Fujiwara, 1987). Namely, $\mathbf{Y}^{(1)} \cong \mathbf{Y}^{(2)}$ if and only if $\mathbf{X}^{(1)} \cong \mathbf{X}^{(2)}$. Furthermore, Williams showed that $\mathbf{X}^{(1)} \cong \mathbf{X}^{(2)}$ if and only if $\mathbf{T}_{\text{tot}}^{(1)} = P^{-1}\mathbf{T}_{\text{tot}}^{(2)}P$ for some permutation matrix P (Williams, 1973). Since $\mathbf{T}_{\text{tot}}^{(i)}$ is the incidence matrix of $\mathbf{S}_{\text{tot}}^{(i)}$ for $i = 1, 2$, $\mathbf{T}_{\text{tot}}^{(1)} = P^{-1}\mathbf{T}_{\text{tot}}^{(2)}P$ indicates that $\mathbf{S}_{\text{tot}}^{(1)} \approx P^{-1}\mathbf{S}_{\text{tot}}^{(2)}P$. Proposition 2.1 demonstrates that $\mathbf{S}_{\text{tot}}^{(i)}$ produces $\mathbf{Y}^{(i)}$. Therefore, $\mathbf{Y}^{(1)} \cong \mathbf{Y}^{(2)}$ if and only if $\mathbf{S}_{\text{tot}}^{(1)} \approx P^{-1}\mathbf{S}_{\text{tot}}^{(2)}P$ for some permutation matrix P.

For the case where $\mathbf{S}_{\text{tot}}^{(i)}$ is not right-resolving for some i, there exists a right-resolving symbolic transition matrix $\mathbf{H}^{(i)}$ which still exactly represents the original $\mathbf{Y}^{(i)}$ (Lind & Marcus, 1995). (Such a matrix is obtained by applying the so-called *subset construction method* in the automaton theory.) Similar to the discussion above, we conclude that $\mathbf{Y}^{(1)} \cong \mathbf{Y}^{(2)}$ if and only if $\mathbf{H}_{\text{tot}}^{(1)} \approx P^{-1}\mathbf{H}_{\text{tot}}^{(2)}P$ for some permutation matrix P. This completes the proof. □

Example 2.3. Suppose the templates of an STCNN are given by the following:

$$[a^{(1)}, a_r^{(1)}, z^{(1)}] = [2.9, 1.7, 0.1]$$

$$[a^{(2)}, a_r^{(2)}, b^{(2)}, b_r^{(2)}, z^{(2)}] = [-0.3, -1.2, 0.7, 2.3, 0.9].$$

Then the basic set of admissible local patterns is

$$\mathcal{B} = \left\{ \boxed{\begin{matrix} -+ \\ -- \end{matrix}}, \boxed{\begin{matrix} -+ \\ +- \end{matrix}}, \boxed{\begin{matrix} +- \\ -+ \end{matrix}}, \boxed{\begin{matrix} +- \\ ++ \end{matrix}}, \boxed{\begin{matrix} ++ \\ -+ \end{matrix}}, \boxed{\begin{matrix} ++ \\ ++ \end{matrix}} \right\}.$$

The transition matrix $\mathbf{T}$ of the solution space $\mathbf{Y}$ is

$$\mathbf{T} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix},$$

and the symbolic transition matrices of the hidden and output spaces are

$$\mathbf{S}^{(1)} = \begin{pmatrix} \varnothing & \varnothing & \alpha_1 & \varnothing \\ \varnothing & \varnothing & \alpha_3 & \varnothing \\ \varnothing & \alpha_2 & \varnothing & \alpha_2 \\ \varnothing & \alpha_4 & \varnothing & \alpha_4 \end{pmatrix} \quad \text{and} \quad \mathbf{S}^{(2)} = \begin{pmatrix} \varnothing & \varnothing & \alpha_2 & \varnothing \\ \varnothing & \varnothing & \alpha_2 & \varnothing \\ \varnothing & \alpha_3 & \varnothing & \alpha_4 \\ \varnothing & \alpha_3 & \varnothing & \alpha_4 \end{pmatrix}$$

respectively. Deleting the first column and the first row of $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$ and using the original notations to indicate the new matrices, we have

$$\mathbf{S}^{(1)} = \begin{pmatrix} \varnothing & \alpha_3 & \varnothing \\ \alpha_2 & \varnothing & \alpha_2 \\ \alpha_4 & \varnothing & \alpha_4 \end{pmatrix} \quad \text{and} \quad \mathbf{S}^{(2)} = \begin{pmatrix} \varnothing & \alpha_2 & \varnothing \\ \alpha_3 & \varnothing & \alpha_4 \\ \alpha_3 & \varnothing & \alpha_4 \end{pmatrix}$$

respectively. Furthermore, following the total amalgamation algorithm we derive that

$$\mathbf{S}_{\text{tot}}^{(1)} = \begin{pmatrix} \alpha_4 & \alpha_3 \\ \alpha_2 & \varnothing \end{pmatrix} \quad \text{and} \quad \mathbf{S}_{\text{tot}}^{(2)} = \begin{pmatrix} \varnothing & \alpha_2 \\ \alpha_3 & \alpha_4 \end{pmatrix},$$

and it is seen that $\mathbf{S}_{\text{tot}}^{(2)} \approx P^{-1}\mathbf{S}_{\text{tot}}^{(1)}P$ with $P = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$. Applying Theorem 2.1, we can conclude that $\mathbf{Y}^{(1)} \cong \mathbf{Y}^{(2)}$. In fact, for this example, $\mathbf{Y}^{(1)} = \mathbf{Y}^{(2)}$ since $\mathbf{S}_{\text{tot}}^{(2)} = P^{-1}\mathbf{S}_{\text{tot}}^{(1)}P$.

Remark 2.1. We remark that the bijective maps in the determination of the equivalence of $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$ in Theorem 2.1 are either (i) the identity map, or (ii) the change of symbols that interchanges "+" with "−".

3. Topological conjugacy for multi-layer cellular neural networks

The idea addressed in the previous section can be applied to provide a general perspective when it comes to examining the equivalence of the topological structures of the output space and a specific hidden space (or two hidden spaces) in multi-layer cellular neural networks. This section aims to extrapolate the discussion in the previous section to more general cases.

A one-dimensional MCNN is realized as

$$\frac{dx_i^{(l)}}{dt} = -x_i^{(l)} + \sum_{|k| \leq d} a_k^{(l)} y_{i+k}^{(l)} + \sum_{|\ell| \leq d} b_\ell^{(l)} u_{i+\ell}^{(l)} + z^{(l)}, \tag{9}$$

for some $d \in \mathbb{N}$, $1 \le l \le n \in \mathbb{N}$, $i \in \mathbb{N}$, where

$$u_i^{(l)} = y_i^{(l-1)} \quad \text{for } 2 \le l \le n \quad \text{and}$$
$$y_i^{(l)} = f(x_i^{(l)}) = \frac{1}{2}(|x_i^{(l)} + 1| - |x_i^{(l)} - 1|).$$

Once we focus on the mosaic solution of (9), namely, $|x_i^{(l)}| > 1$ for all $i \in \mathbb{N}$, $1 \le l \le n$, the following procedure divides the parameter space into finitely equivalent regions such that any two sets of parameters possess the same output patterns if and only if they belong in the same region. For $1 \le l \le n$, the output $y_i^{(l)} = 1$ if and only if

$$a_0^{(l)} - 1 + z^{(l)} > -\left(\sum_{0<|k|\le d} a_k^{(l)} y_{i+k}^{(l)} + \sum_{|\ell|\le d} b_\ell^{(l)} u_{i+\ell}^{(l)}\right);$$

the output $y_i^{(l)} = -1$ if and only if

$$a_0^{(l)} - 1 - z^{(l)} > \sum_{0<|k|\le d} a_k^{(l)} y_{i+k}^{(l)} + \sum_{|\ell|\le d} b_\ell^{(l)} u_{i+\ell}^{(l)}.$$

Let

$$V^N = \{v \in \mathbb{R}^N : v = (v_1, v_2, \ldots, v_N), \text{ and } |v_i| = 1, 1 \le i \le N\},$$

where $N = 4d + 1$. We denote $\alpha^{(l)} = (a_{-d}^{(l)}, \ldots, a_{-1}^{(l)}, a_1^{(l)}, \ldots, a_d^{(l)})$, $\beta^{(l)} = (b_{-d}^{(l)}, \ldots, b_d^{(l)})$. It follows immediately that the collection of admissible local patterns centered by "+1" and "−1" are

$$\mathcal{B}_+^{(l)} = \{v \diamond w : a^{(l)} - 1 + z^{(l)} > -(\alpha^{(l)} \cdot v + \beta^{(l)} \cdot w)\}$$

and

$$\mathcal{B}_-^{(l)} = \{v \diamond w : a^{(l)} - 1 - z^{(l)} > \alpha^{(l)} \cdot v + \beta^{(l)} \cdot w\},$$

respectively. Herein we reuse the notation $\diamond$ to pile two patterns (which need not be of the same size) together. For instance,

$$t_1t_2t_3 \diamond t_1't_2't_3't_4' = \boxed{\begin{matrix} t_1t_2t_3 \\ t_1't_2't_3't_4' \end{matrix}}.$$

In this case, $\mathcal{B}_+^{(l)}$ and $\mathcal{B}_-^{(l)}$ are consisting of patterns of height 2, whose top pattern is of length $2d$ and bottom pattern is of length $2d + 1$. The basic set of admissible local patterns for the lth layer of (9) is then recorded as $\mathcal{B}^{(l)} = (\widetilde{\mathcal{B}}_+^{(l)}, \widetilde{\mathcal{B}}_-^{(l)})$, where

$$\widetilde{\mathcal{B}}_+^{(l)} = \{v_+ \diamond w : v \diamond w \in \mathcal{B}_+^{(l)}\},$$
$$\widetilde{\mathcal{B}}_-^{(l)} = \{v_- \diamond w : v \diamond w \in \mathcal{B}_-^{(l)}\},$$

and $v_+ \in V^{2d+1}$ (resp. $v_- \in V^{2d+1}$) is obtained by inserting 1 (resp. −1) at the center coordinate of $v \in V^{2d}$. We denote the parameters of (9) by $(\mathbb{A}, \mathbb{B}, \mathbb{z})$, herein

$$\mathbb{A} = (A^{(1)}, \ldots, A^{(n)}), \qquad \mathbb{B} = (B^{(1)}, \ldots, B^{(n)}), \quad \text{and}$$
$$\mathbb{z} = (z^{(1)}, \ldots, z^{(n)}).$$

Then, the admissible local patterns induced by $(\mathbb{A}, \mathbb{B}, \mathbb{z})$ can be denoted by

$$\mathcal{B}(\mathbb{A}, \mathbb{B}, \mathbb{z}) = (\mathcal{B}^{(1)}, \mathcal{B}^{(2)}, \ldots, \mathcal{B}^{(n)}).$$

Following the determination of the basic set of admissible local patterns we introduce the ordering matrix in similar fashion to our investigation in the previous section. We substitute the output patterns "−1" and "+1" by − and +, respectively. First, we assign each pattern in $\{-, +\}^{d' \times n}$ an order so that the ordering matrix $\mathbb{X}_{d' \times n}$ is well-defined, where $d' = 2d + 1$ is the width of admissible local patterns. We define $\chi : \{-, +\} \to \{0, 1\}$ and $\eta : \{-, +\}^{k \times l} \to \mathbb{N}$ as

$$\chi(-) = 0, \qquad \chi(+) = 1,$$

and

$$\eta\left(x = \begin{matrix} x_{1,l} & x_{2,l} & \cdots & x_{k,l} \\ x_{1,l-1} & x_{2,l-1} & \cdots & x_{k,l-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1,1} & x_{2,1} & \cdots & x_{k,1} \end{matrix}\right)$$
$$= 1 + \sum_{i,j} \chi(x_{i,j}) \cdot 2^{l(k-i)+(l-j)},$$

respectively. For example, those patterns in the collection $\{-, +\}^{2\times 2}$ are ordered as

$$\eta\begin{pmatrix} -- \\ -- \end{pmatrix} = 1, \eta\begin{pmatrix} -+ \\ -- \end{pmatrix} = 2, \ldots, \eta\begin{pmatrix} +- \\ ++ \end{pmatrix} = 15, \eta\begin{pmatrix} ++ \\ ++ \end{pmatrix} = 16.$$

The ordering matrix $\mathbb{X}_{d' \times n}$ of (9) is then well-defined according to the η-ordered set $\{-, +\}^{(d'-1)\times n}$. Once the basic set of admissible local patterns $\mathcal{B}$ is assigned, the transition matrix $\mathbf{T} \equiv \mathbf{T}(\mathcal{B}) \in \mathcal{M}_{(d'-1)n}(\{0, 1\})$ is defined as

$$\mathbf{T}(p, q) = 1 \quad \text{if and only if } \mathbb{X}(p, q) \in \mathcal{B},$$

where $1 \le p, q \le 2^{(d'-1)n}$. Let $\mathcal{A}_{d'} = \{-, +\}^{d' \times 1} = \{\alpha_i\}_{i=1}^{2^{d'}}$ be an alphabet consisting of binary patterns of dimension $d' \times 1$, where α_i represents the symbol $x = x_1 \ldots x_{d'} \in \{-, +\}^{d' \times 1}$ satisfying $i = 1 + \Sigma_{j=1}^{d'} \chi(x_j) 2^{d'-j}$. Express $\mathbb{X}(p, q)$, $1 \le p, q \le 2^{(d'-1)N}$, as $\alpha_{k_n} \diamond \alpha_{k_{n-1}} \diamond \cdots \diamond \alpha_{k_1}$. For $1 \le \ell \le n$, let $\mathbf{S}^{(\ell)} \equiv \mathbf{S}^{(\ell)}(\mathcal{B})$ be the ℓth symbolic transition matrix over $\mathcal{A}_{d'} \cup \{\varnothing\}$ and be defined as

$$\mathbf{S}^{(\ell)}(p, q) = \begin{cases} \alpha_{k_\ell}, & \text{if } \mathbf{T}(p, q) = 1, \\ & \mathbb{X}(p, q) = \alpha_{k_n} \diamond \cdots \diamond \alpha_{k_1}; \\ \varnothing, & \text{otherwise.} \end{cases}$$

Let

$$\mathbf{Y}^{(\ell)} = \{(y_i^{(\ell)})_{i\in\mathbb{N}} : (y_i^{(n)} \diamond y_i^{(n-1)} \diamond \cdots \diamond y_i^{(1)})_{i\in\mathbb{N}} \in \mathbf{Y}\}$$

for $1 \le \ell \le n$. It is demonstrated that $\mathbf{S}^{(\ell)}$ well-describes the topological structure of the ℓth hidden space $\mathbf{Y}^{(\ell)}$, while $\mathbf{S}^{(n)}$ describes the topological structure of the output space $\mathbf{Y}^{(n)}$ completely (Ban et al., 2009).

To investigate whether $\mathbf{Y}^{(\ell_1)}$ is topologically conjugate to $\mathbf{Y}^{(\ell_2)}$ for $1 \le \ell_1, \ell_2 \le n$, we start with the construction of the symbolic total amalgamation matrices of $\mathbf{S}^{(\ell_1)}$ and $\mathbf{S}^{(\ell_2)}$. Suppose $\mathbf{S}^{(\ell_1)}$ and $\mathbf{S}^{(\ell_2)}$ consist of no empty rows and columns (we delete the empty rows/columns and their corresponding indexed columns/rows if necessary). Here an empty row/column refers to a row/column consisting of an empty entry $\varnothing$. Let $\mathbf{S}_{\text{tot}}^{(\ell_1)}$ and $\mathbf{S}_{\text{tot}}^{(\ell_2)}$ be obtained by applying the total amalgamation algorithm to $\mathbf{S}^{(\ell_1)}$ and $\mathbf{S}^{(\ell_2)}$, respectively. We then have the following theorem:

Theorem 3.1. *Suppose $\mathbf{S}_{\text{tot}}^{(\ell_1)}$ and $\mathbf{S}_{\text{tot}}^{(\ell_2)}$ are both right-resolving without empty rows/columns. Then $\mathbf{Y}^{(\ell_1)} \cong \mathbf{Y}^{(\ell_2)}$ if and only if $\mathbf{S}_{\text{tot}}^{(\ell_1)} \approx P^{-1}\mathbf{S}_{\text{tot}}^{(\ell_2)}P$ for some permutation matrix P. For general cases, there exist right-resolving matrices $\mathbf{H}^{(\ell_1)}$ and $\mathbf{H}^{(\ell_2)}$ such that $\mathbf{Y}^{(\ell_1)} \cong \mathbf{Y}^{(\ell_2)}$ if and only if $\mathbf{H}_{\text{tot}}^{(\ell_1)} \approx P^{-1}\mathbf{H}_{\text{tot}}^{(\ell_2)}P$ for some permutation matrix P.*

Proof. The proof is analogous to the discussion in the proof of Theorem 2.1, and thus is omitted. □

Example 3.1. Consider a three-layer cellular neural network given as

$$\begin{cases} \dfrac{dx_i^{(3)}}{dt} = -x_i^{(3)} + a^{(3)}y_i^{(3)} + a_r^{(3)}y_{i+1}^{(3)} + b^{(3)}y_i^{(2)} \\ \qquad + b_r^{(3)}y_{i+1}^{(2)} + z^{(3)}, \\ \dfrac{dx_i^{(2)}}{dt} = -x_i^{(2)} + a^{(2)}y_i^{(2)} + a_r^{(2)}y_{i+1}^{(2)} + b^{(2)}y_i^{(1)} \\ \qquad + b_r^{(2)}y_{i+1}^{(1)} + z^{(2)}, \\ \dfrac{dx_i^{(1)}}{dt} = -x_i^{(1)} + a^{(1)}y_i^{(1)} + a_r^{(1)}y_{i+1}^{(1)} + z^{(1)}, \end{cases} \tag{10}$$

where

$$[a^{(1)}, a_r^{(1)}, z^{(1)}] = [2.9, 1.7, 0.1],$$

$$[a^{(i)}, a_r^{(i)}, b^{(i)}, b_r^{(i)}, z^{(i)}] = [-0.3, -1.2, 0.7, 2.3, 0.9] \quad \text{for } i = 2, 3.$$

Since the parameters for the second and the third layer networks are exactly the same, it is intuitive that such a three-layer network can be reduced to a two-layer system. We show that this is true by applying Theorem 3.1.

Evidently, the solution space of (10) is generated by the set

$$\mathcal{B} = \left\{ \boxed{\begin{array}{c}+-\\-+\\--\end{array}}, \boxed{\begin{array}{c}++\\-+\\--\end{array}}, \boxed{\begin{array}{c}+-\\-+\\+-\end{array}}, \boxed{\begin{array}{c}++\\-+\\+-\end{array}}, \boxed{\begin{array}{c}-+\\+-\\-+\end{array}}, \boxed{\begin{array}{c}-+\\+-\\++\end{array}}, \boxed{\begin{array}{c}+-\\++\\-+\end{array}}, \boxed{\begin{array}{c}++\\++\\-+\end{array}}, \boxed{\begin{array}{c}+-\\++\\++\end{array}}, \boxed{\begin{array}{c}++\\++\\++\end{array}} \right\}.$$

Defining the ordering matrix $\mathbb{X}_3$ indexed by

$$\boxed{\begin{array}{c}-\\-\\-\end{array}}, \boxed{\begin{array}{c}+\\-\\-\end{array}}, \boxed{\begin{array}{c}-\\+\\-\end{array}}, \boxed{\begin{array}{c}+\\+\\-\end{array}}, \boxed{\begin{array}{c}-\\-\\+\end{array}}, \boxed{\begin{array}{c}+\\-\\+\end{array}}, \boxed{\begin{array}{c}-\\+\\+\end{array}}, \boxed{\begin{array}{c}+\\+\\+\end{array}},$$

it is seen that the transition matrix is

$$\mathbf{T} = \begin{pmatrix} 0&0&0&0&0&0&0&0\\ 0&0&1&1&0&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&0\\ 0&0&1&1&0&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&1 \end{pmatrix}.$$

Since the first, second, and fifth columns of $\mathbf{T}$ are zero vectors, we can reduce $\mathbf{T}$ to a 5×5 matrix

$$\begin{pmatrix} 0&0&1&0&0\\ 0&0&0&1&1\\ 1&1&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&1 \end{pmatrix}$$

without loss of generality. In this case, the three symbolic transition matrices for (10) are

$$\mathbf{S}^{(1)} = \begin{pmatrix} \varnothing&\varnothing&\alpha_2&\varnothing&\varnothing\\ \varnothing&\varnothing&\varnothing&\alpha_2&\alpha_2\\ \alpha_3&\alpha_3&\varnothing&\varnothing&\varnothing\\ \varnothing&\varnothing&\alpha_2&\varnothing&\varnothing\\ \varnothing&\varnothing&\varnothing&\alpha_2&\alpha_2 \end{pmatrix},$$

$$\mathbf{S}^{(2)} = \begin{pmatrix} \varnothing&\varnothing&\alpha_3&\varnothing&\varnothing\\ \varnothing&\varnothing&\varnothing&\alpha_4&\alpha_4\\ \alpha_2&\alpha_2&\varnothing&\varnothing&\varnothing\\ \varnothing&\varnothing&\alpha_3&\varnothing&\varnothing\\ \varnothing&\varnothing&\varnothing&\alpha_4&\alpha_4 \end{pmatrix},$$

and

$$\mathbf{S}^{(3)} = \begin{pmatrix} \varnothing&\varnothing&\alpha_2&\varnothing&\varnothing\\ \varnothing&\varnothing&\varnothing&\alpha_3&\alpha_4\\ \alpha_3&\alpha_4&\varnothing&\varnothing&\varnothing\\ \varnothing&\varnothing&\alpha_2&\varnothing&\varnothing\\ \varnothing&\varnothing&\varnothing&\alpha_3&\alpha_4 \end{pmatrix},$$

respectively; applying the total amalgamation algorithm to $\mathbf{S}^{(i)}$, $i = 1, 2, 3$, we derive that

$$\mathbf{S}_{\text{tot}}^{(1)} = \begin{pmatrix} \varnothing&\alpha_2\\ \alpha_3&\alpha_2 \end{pmatrix}, \qquad \mathbf{S}_{\text{tot}}^{(2)} = \begin{pmatrix} \alpha_4&\alpha_3\\ \alpha_2&\varnothing \end{pmatrix}, \quad \text{and}$$

$$\mathbf{S}_{\text{tot}}^{(3)} = \begin{pmatrix} \varnothing&\alpha_2\\ \alpha_3&\alpha_4 \end{pmatrix},$$

respectively. It can be verified that $\mathbf{S}_{\text{tot}}^{(2)} = P^{-1}\mathbf{S}_{\text{tot}}^{(3)}P$ with $P = \begin{pmatrix} 0&1\\ 1&0 \end{pmatrix}$ and there exists no permutation matrix M such that $\mathbf{S}_{\text{tot}}^{(1)} = M^{-1}\mathbf{S}_{\text{tot}}^{(i)}M$ for $i = 2, 3$; hence $\mathbf{Y}^{(2)}$ is topologically conjugate to $\mathbf{Y}^{(3)}$ and this three-layer network can be replaced by a two-layer network without compromising its functionality.

4. Conclusion and discussion

This paper investigates whether or not an n-layer cellular neural network can be reduced to an m-layer cellular neural network, where $m < n$. More specifically, this work identifies the "*actual depth*" of a given multi-layer neural network such that the optimized network still exhibits the same phenomena as the original one up to the change of symbols. From a mathematical point of view, this is equivalent to studying whether the output spaces of the ith layer are topologically conjugate to the output space of the jth layer for some i, j satisfying $1 \le i < j \le n$.

We propose an algorithm to achieve our target by introducing a symbolic transition matrix and compare the two "operated" symbolic transition matrices to see if they are the same, up to a permutation and the change of symbols. The contribution of the present work focuses on those multi-layer cellular neural networks where the number of neighbors for each cell in each layer is the same, and the template (also known as weight) is invariant for each layer.

It is remarkable that our methodology can extend to the following cases:

(a) Different numbers of neighbors for distinct layers, i.e., the number of neurons connected to $x_i^{(j)}$ depends on i, j.
(b) Different templates for distinct neurons, i.e., the template of $x_i^{(j)}$ depends on i, j.
(c) Application of different output functions to the same network.
(d) Two individual (multi-layer) neural networks.

Notably, the last case is an extension of Ban and Chang (2015). The related work is in preparation and will be elaborated in a future paper.

We conclude this paper with a further discussion of Example 2.3. It is elucidated that the two-layer cellular neural network given in Example 2.3 can be replaced by a single layer network since its output space is conjugate to the hidden space. Consider the following cellular neural network

$$\frac{x_i}{dt} = -x_i + 2y_i u y_{i+1} - 0.3, \qquad i \in \mathbb{N}, \tag{11}$$

wherein the output space of (11) is generated by the patterns

$$--, \qquad -+, \qquad +-,$$

and the symbolic transition matrix is

$$\mathbf{S} = \begin{pmatrix} \alpha_1&\alpha_2\\ \alpha_3&\varnothing \end{pmatrix}.$$

Notably, the one-to-one correspondence between $\mathbf{S}$ and $\mathbf{S}_{\text{tot}}^{(2)}$ is

$$\alpha_1 \leftrightarrow \alpha_4, \qquad \alpha_2 \leftrightarrow \alpha_3;$$

hence these two networks exhibit the same dynamics.

Acknowledgments

The authors are grateful to the anonymous referees' suggestions and comments, which significantly improved the quality and readability of this paper. Furthermore, some future work inspired by the referees is in preparation.

This work is partially supported by the Ministry of Science and Technology, ROC (Contract No MOST 102-2628-M-259-001-MY3 and 104-2115-M-390-004-).

References

Arena, P., Baglio, S., Fortuna, L., & Manganaro, G. (1998). Self-organization in a two-layer CNN. *IEEE Transactions on Circuits and Systems I Fundamental Theory and Applications*, *45*, 157–162.

Ban, J.-C., & Chang, C.-H. (2015). Realization problem of multi-layer cellular neural networks. *Neural Networks*, *70*, 9–17.

Ban, J.-C., Chang, C.-H., & Lin, S.-S. (2012). The structure of multi-layer cellular neural networks. *Journal of Differential Equations*, *252*, 4563–4597.

Ban, J.-C., Chang, C.-H., Lin, S.-S., & Lin, Y.-H. (2009). Spatial complexity in multi-layer cellular neural networks. *Journal of Differential Equations*, *246*, 552–580.

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, *2*, 1–127.

Carmona, R., Jimenez-Garrido, F., Dominguez-Castro, R., Espejo, S., & Rodriguez-Vazquez, A. (2002). CMOS realization of a 2-layer cnn universal machine chip In *Proceedings of the 2002 7th IEEE international workshop on cellular neural networks and their applications, 2002, CNNA 2002* (pp. 444–451).

Chang, C.-H. (2015). Deep and shallow architecture of multi-layer neural networks. *IEEE Transactions Neural Networks Learning Systems*, *26*, 2477–2486.

Chua, L. O., & Roska, T. (2002). *Cellular neural networks and visual computing*. Cambridge University Press.

Chua, L. O., & Shi, B. E. (1990). *Multiple layer cellular neural networks: A tutorial*. Electronics Research Laboratory, College of Engineering, University of California.

Chua, L. O., & Shi, B. E. (1991). Multiple layer cellular neural networks: A tutorial. In E. F. Deprettere, & A.-J. van der Veen (Eds.), *Algorithms and parallel VLSI architectures* (pp. 137–168). Amsterdam, The Netherlands: Elsevier.

Chua, L. O., & Yang, L. (1988a). Cellular neural networks: Applications. *IEEE Transactions on Circuits and Systems*, *35*, 1273–1290.

Chua, L. O., & Yang, L. (1988b). Cellular neural networks: Theory. *IEEE Transactions on Circuits and Systems*, *35*, 1257–1272.

Crounse, K. R., & Chua, L. O. (1995). Methods for image processing and pattern formation in cellular neural networks: A tutorial. *IEEE Transactions on Circuits and Systems*, *42*, 583–601.

Crounse, K. R., Roska, T., & Chua, L. O. (1993). Image halftoning with cellular neural networks. *IEEE Transactions on Circuits and Systems*, *40*, 267–283.

Fujiwara, M. (1987). Conjugacy for one-sided sofic systems. In *World Sci. Adv. Ser. Dynam. Systems*: *Vol. 2. Dynamical systems and singular phenomena* (pp. 189–202). World Scientific Publishers.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings National Academy Sciences United States of America*, *79*(8), 2554–2558.

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings National Academy Science United States of America*, *81*(10), 3088–3092.

Li, X. (2009). Analysis of complete stability for discrete-time cellular neural networks with piecewise linear output functions. *Neural Computation*, *21*, 1434–1458.

Lind, D., & Marcus, B. (1995). *An introduction to symbolic dynamics and coding*. Cambridge: Cambridge University Press.

Murugesh, V. (2010). Image processing applications via time-multiplexing cellular neural network simulator with numerical integration algorithms. *International Journal of Computer Mathematics*, *87*, 840–848.

Paolo-Civalleri, P., & Gilli, M. (1999). On stability of cellular neural networks. *Journal of VLSI signal processing systems for signal, image and video technology*, *23*(2–3), 429–435.

Peng, J., Zhang, D., & Liao, X. (2009). A digital image encryption algorithm based on hyper-chaotic cellular neural network. *Fundamenta Informaticae*, *90*, 269–282.

Rakkiyappan, R., Chandrasekar, A., Lakshmanan, S., & Park, J. H. (2014). Exponential stability of Markovian jumping stochastic Cohen–Grossberg neural networks with mode-dependent probabilistic time-varying delays and impulses. *Neurocomputing*, *131*, 265–277.

Rakkiyappan, R., Zhu, Q., & Chandrasekar, A. (2014). Stability of stochastic neural networks of neutral type with Markovian jumping parameters: A delay-fractioning approach. *Journal of the Franklin Institute*, *351*, 1553–1570.

Savaci, F., & Vandewalle, J. (1992). On the stability analysis of cellular neural networks. In *Second international workshop on cellular neural networks and their applications, 1992. CNNA-92 Proceedings* (pp. 240–245). IEEE.

Török, L., & Roska, T. (2004). Stability of multi-layer cellular neural/nonlinear networks. *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, *14*, 3567–3586.

Williams, R. F. (1973). Classification of subshifts of finite type. *Annals of Mathematics*, *98*, 120–153; (1974). *Annals of Mathematics*, *99*, 380–381 (errata).

Wu, C. W., & Chua, L. O. (1997). A more rigorous proof of complete stability of cellular neural networks. *IEEE Transaction Circuits Systems. I. Fundamental Theory Applications*, *44*(4), 370–371.

Xavier-de Souza, S., Yalcin, M., Suykens, J., & Vandewalle, J. (2004). Toward CNN chip-specific robustness. *IEEE Transactions on Circuits and Systems. I. Regular Papers*, *51*, 892–902.

Xu, J., Pi, D., Cao, Y.-Y., & Zhong, S. (2007). On stability of neural networks by a lyapunov functional-based approach. *IEEE Transaction Circuits Systems. I. Regular Papers*, *54*(4), 912–924.

Yang, Z., Nishio, Y., & Ushida, A. (2001). A two layer CNN in image processing applications. In *Proc. of the 2001 international symposium on nonlinear theory and its applications* (pp. 67–70).

Yang, Z., Nishio, Y., & Ushida, A. (2002). Image processing of two-layer CNNs – applications and their stability. *IEICE Transactions on Fundamentals*, *E85-A*, 2052–2060.

Zou, F., & Nossek, J. A. (1991). Stability of cellular neural networks with opposite-sign templates. *IEEE Transactions on Circuits and Systems*, *38*(6), 675–677.