Neural Process Lett (2016) 43:31–47
DOI 10.1007/s11063-014-9400-7

The Spatial Complexity of Inhomogeneous Multi-layer Neural Networks

Jung-Chao Ban · Chih-Hung Chang

Published online: 26 November 2014

Abstract Inhomogeneous multi-layer neural networks (IHMNNs) have been applied in various fields, for example, biological and ecological contexts. This work studies the learning problem of IHMNNs with an activation function $f(x) = \frac{1}{2}(|x+1| - |x-1|)$ that derives from cellular neural networks, which can be adapted to the study of the vision systems of mammals. Applying the well-developed theory of symbolic dynamics, the explicit formulae of the topological entropy of the output and hidden spaces are given. We also demonstrate that, for any $\lambda \in [0, \log 2]$ and $\epsilon > 0$, parameters such that the topological entropy h of the hidden/output space of IHMNN that satisfies $|h - \lambda| < \epsilon$ exists. This means that the collection of topological entropies is dense in the closed interval $[0, \log 2]$, which leads to the fact that IHMNNs are universal machines in some sense and hence are more efficient in learning algorithms. This paper aims to provide a mathematical foundation for the illustration of the capability of machine learning, while the method we have adopted can be extended to the investigation of multi-layer neural networks with other activation functions.

Keywords Multi-layer neural networks · Topological entropy · Sofic shift · Cellular neural networks

1 Introduction

In the past few decades, multi-layer neural networks (MNNs) have received considerable attention and have been successfully applied to many areas such as combinatorial optimization, signal processing and pattern recognition [1–6].

J.-C. Ban
Department of Applied Mathematics, National Dong Hwa University, Hualien 970003, Taiwan, ROC
e-mail: jcban@mail.ndhu.edu.tw

C.-H. Chang (✉)
Department of Applied Mathematics, National University of Kaohsiung, Kaohsiung 81148, Taiwan, ROC
e-mail: chihhung@mail.fcu.edu.tw; chchang@nuk.edu.tw

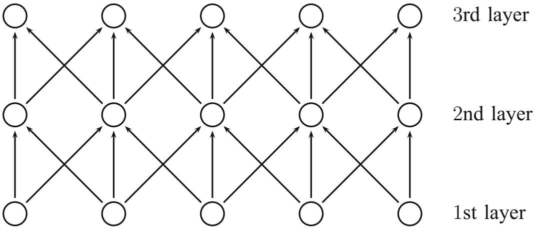

Fig. 1 Three-layer neural networks with the nearest neighborhood

One of the most important reasons for coupling NNs is the simulation of the visual systems of mammals ([7,8], each layer symbolizing a single cortex in the visual system). T. Serre et al. [9] proved that the mammal brain is organized into deep architectures, i.e., the number of layers in MNNs is larger than was first thought. Scientists are interested in learning and training in deep architectures [10–14] due to the architectural depth of the mammalian brain. A great and successful series of results on deep architectures was obtained by Hinton et al. in 2006 using Deep Belief Networks (DBNs) [15] and Restricted Boltzmann Machine (RBM) [16] methods, and have been applied to many fields, e.g., classification tasks [17,18], regression [19], dimensionality reduction [20,21], modeling textures [22], modeling motion [23,24], and natural language processing [25,26]. The best general reference is referred to as [27], wherein the reader can find the complete bibliography. The model studied in this paper (15) is an extension of a sigmoid belief network, which is one kind of DBN. Differently from investigating RBMs and DBNs as probabilistic networks, this work studies the abundance of the pattern formation of MNNs. A MNN presented as a directed graph model is shown in Fig. 1. Traditionally, the template (or weight, more specifically, the set of parameters) for MNN is homogeneous (also known as isotropic), i.e., the template is space-invariant. However, there are more and more MNNs that use inhomogeneous templates to describe some of the problems that arise from biological and ecological contexts [28–33]. Some new and interesting phenomena of pattern formation and spatial chaos were also found in inhomogeneous MNNs (IHMNNs).

In this paper, we considered an activation function which arises from cellular neural networks (CNNs, [34]). Namely,

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

Since CNNs have been applied to many areas such as image processing [35–40], the choice of the activation function $f(x)$ seems to be the best adapted to our goal on the study of the vision systems of mammals. Related topics, such as pattern formation and spatial chaos for mosaic solutions have been discussed in CNNs [41] and multi-layer CNN (MCNNs, [42]) models, herein an equilibrium solution of a neural network is called mosaic if $|x_i| > 1$ for all i. Aein and Talebi demonstrate that CNNs are able to model multidimensional systems whose behaviors are governed by unknown/uncertain partial differential equations [43]. They use a modified backpropagation algorithm to train a CNN so that it is capable of modeling the dynamics of a two-dimensional mechanical vibration problem with unknown system equations. However, it seems that there are few studies on MNN models with respect to the activation function $f(x)$.

The aim of this paper is to establish the mathematical foundation of mosaic solutions on IHMNNs with activation function $f(x)$. We focus on the complexity (entropy) of the mosaic solution space. From the last paragraph, this topic seems to be more important and interesting regarding the study of vision systems and is approaching the subject of artificial intelligence (AI) [27]. It is remarkable that classical CNNs are a special case of IHMNNs when we use an activation function $f(x)$ derives from CNNs. We also emphasize that the method we have provided herein is more general, an easy extension that leads us to consider the classical McCulloch-Pitts model and signum activation function.

MNNs are relatively crude networks of neurons based on the neural structure of the brain. They process records one at a time, and learn by comparing their classification of the records with the known actual classification of the records. In the training phase, the correct class for each record is known (this is termed "*supervised training*"), and the output nodes can therefore be assigned "correct" values 1 for the node corresponding to the correct class, and 0 for the others. Recently, Fukushima [8] proposed a new learning rule named "*add-if-silent*" which produces a higher recognition rate for handwritten digits recognition with a smaller scale of the network than the neocognitron of previous versions. Assigning 0–1 values during the training process can be treated as the mosaic output ± 1 of $f(x)$.

The investigation of mosaic solutions is most essential in MNN models due to the learning algorithm and training processing since the complete stability of the systems and the activation function we chose (cf. [37,44–46]). More abundant output patterns make the learning algorithm more efficient. From the mathematical point of view, the larger the set of topological entropies of MNNs is, the more phenomena MNNs are capable of. Denseness of the entropy set indicates that, in some sense, MNNs are capable of exhibiting any phenomena one requests, and hence leads MNNs to the universal machine. This makes the learning more efficient. Many types of activation function, e.g., linear, McCulloch-Pitts, signum, sigmoid and Ramp functions, are chosen for many specific purposes. The activation function represents which conditions of the synapses will be activated. Different activation function produces different dynamical systems on output solutions. Related investigation will be illustrated in future works.

This paper is organized as follows. In Sect. 2, we set up the notation and terminology of homogeneous MNNs, with the necessary materials from symbolic dynamical systems provided to compute the complexity (entropy) for the mosaic solutions. Section 3 is devoted to the study of complexity of the constant IHMNNs (CIHMNNs). The rigorous value for the entropy can be computed therein and this leads us to solve well-known ϵ-dense property for entropy which comes from the universality of neural networks. A brief conclusion is presented in Sect. 4.

2 Preliminary

This section introduces notations and terminologies of homogeneous MNNs. Some results in our previous work [47], such as the determination of the basic set of admissible local patterns and the transition matrix, are presented to make the present investigation self-contained. Ban and Chang [47] relate the elucidation of the output space to symbolic dynamics and apply the well-developed theory of symbolic dynamical systems to find the explicit formula of the topological entropy of the output space. It is remarkable that topological entropy is related to the spectral radius of some matrix presentations of the system, which reduce the reliance on familiarity with symbolic dynamics. Readers who are interested in more detailed information about symbolic dynamics are referred to the book by Lind and Marcus [48] or our previous work [42].

A one-dimensional MNN is realized as

$$\begin{cases} \dfrac{d}{dt}x_i^{(k)}(t) = -x_i^{(k)}(t) + z^{(k)} + a^{(k)} f(x_i^{(k)}(t)) + \displaystyle\sum_{\ell \in \mathcal{N}} b_\ell^{(k)} f(x_{i+\ell}^{(k-1)}(t)), \\ \dfrac{d}{dt}x_i^{(1)}(t) = -x_i^{(1)}(t) + z^{(1)} + a^{(1)} f(x_i^{(1)}(t)) + \displaystyle\sum_{\ell \in \mathcal{N}} a_\ell^{(1)} f(x_{i+\ell}^{(1)}(t)), \end{cases} \tag{1}$$

for some $N \in \mathbb{N}$, $k = 2, \ldots, N$ and $i \in \mathbb{Z}$. We call the finite subset $\mathcal{N} \subset \mathbb{Z}$ the *neighborhood*, and the piecewise linear map $f(x) = \frac{1}{2}(|x+1| - |x-1|)$ is called the *output function*. The *template* $\mathbb{T} = [\mathbf{A}, \mathbf{B}, \mathbf{z}]$ is composed of a *feedback template* $\mathbf{A} = (A_1, A_2)$ with $A_1 = (a^{(1)}, \ldots, a^{(N)})$, $A_2 = (a_\ell^{(1)})_{\ell \in \mathcal{N}}$, a *controlling template* $\mathbf{B} = (B_2, \ldots, B_N)$, and the *threshold* $\mathbf{z} = (z^{(1)}, \ldots, z^{(N)})$, where $B_k = (b_\ell^{(k)})_{\ell \in \mathcal{N}}$ for $k \geq 2$. An equilibrium solution $\mathbf{x} = (x_i^{(1)}, \cdots, x_i^{(N)})_{i \in \mathbb{Z}} \in \mathbb{R}^{\mathbb{Z}_{\infty \times N}}$ of (1) is called *mosaic* if $|x_i^{(k)}| > 1$ for $1 \leq k \leq N, i \in \mathbb{Z}$. The output $\mathbf{y} = (y_i^{(1)} \cdots y_i^{(N)})_{i \in \mathbb{Z}} \in \{-1, 1\}^{\mathbb{Z}_{\infty \times N}}$ of a mosaic solution is called a *mosaic pattern*, where $y_i^{(k)} = f(x_i^{(k)})$. The *solution space* $\mathbf{Y}$ of (1) stores the patterns $\mathbf{y}$, and the *output space* $\mathbf{Y}^{(N)}$ of (1) is the collection of the output patterns, more precisely,

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

For $1 \leq \ell \leq N-1$, the space

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

is called the (ℓth) hidden space of (1). It turns out that the complexity of the solution and output spaces are closely related to how many neighbors a neuron is connected to, i.e., the size of its neighborhood. To investigate the structure of $\mathbf{Y}$ and $\mathbf{Y}^{(N)}$, it is essential to consider MNNs with the nearest neighborhood. A neighborhood $\mathcal{N}$ is called the nearest neighborhood if $\mathcal{N} = \{-1, 1\}$. In [47], the authors showed that there is a positive integer K and unique set of non-overlapping open sub-regions $\{P_k\}_{k=1}^K$ of the parameter space $\mathcal{P}$ of (1) such that $\mathcal{P}$ is the union of the closure of $\{P_k\}_{k=1}^K$, and

$$\mathbf{Y}^{(N)}(\mathbb{T}) = \mathbf{Y}^{(N)}(\mathbb{T}') \quad \text{if and only if} \quad \mathbb{T}, \mathbb{T}' \in P_k \quad \text{for some } k,$$

where $\mathbf{Y}^{(N)}(\mathbb{T})$ indicates the output space of (1) with respect to the template $\mathbb{T}$.

To clarify the discussion, we concentrate on MNNs with the number of layers $N = 2$. For a two-layer neural network with its nearest neighborhood, since $\mathcal{N}$ is finite and $\mathbb{T}$ is invariant for each i, the output space is determined by the so-called *basic set of admissible local patterns*. Replace -1 and 1 by $-$ and $+$, respectively; the basic set of admissible local patterns of the first layer is a subset of

$$\{---, --+, -+-, -++, +--, +-+, ++-, +++\},$$

while the basic set of admissible local patterns of the second layer is a subset of the ordered set $\{p_1, \ldots, p_8\}$, where $p_1, \ldots, p_8$ are

$$-\overset{-}{}-,\ -\overset{-}{}+,\ +\overset{-}{}-,\ +\overset{-}{}+,\ -\overset{+}{}-,\ -\overset{+}{}+,\ +\overset{+}{}-,\ +\overset{+}{}+, \tag{2}$$

respectively. Denote the local pattern $\alpha_1 \overset{\alpha}{} \alpha_2$ by $\alpha \diamond \alpha_1\alpha_2$. Suppose $\mathbf{y}$ is a mosaic pattern; for each $i \in \mathbb{Z}$, the necessary and sufficient condition for $y_i^{(2)} = 1$ is

$$a^{(2)} - 1 + z_2 > -(b_{-1}^{(2)} y_{i-1}^{(1)} + b_1^{(2)} y_{i+1}^{(1)}), \tag{3}$$

and the necessary and sufficient condition for $y_i^{(2)} = -1$ is

$$a^{(2)} - 1 - z_2 > b_{-1}^{(2)} y_{i-1}^{(1)} + b_1^{(2)} y_{i+1}^{(1)}. \tag{4}$$

Set

$$\begin{aligned}
\mathcal{B}^{(2)}(+) &= \left\{ y^{(2)} \diamond y_1^{(1)} y_2^{(1)} : y_1^{(1)}, y_2^{(1)} \text{ satisfy (3)}, y^{(2)} = 1 \right\}, \\
\mathcal{B}^{(2)}(-) &= \left\{ y^{(2)} \diamond y_1^{(1)} y_2^{(1)} : y_1^{(1)}, y_2^{(1)} \text{ satisfy (4)}, y^{(2)} = -1 \right\}.
\end{aligned}$$

The basic set of admissible local patterns of the second layer is denoted by $\mathcal{B}^{(2)} = (\mathcal{B}^{(2)}(+), \mathcal{B}^{(2)}(-))$. Similarly, for each $i \in \mathbb{Z}$, the necessary and sufficient conditions for $y_i^{(1)} = 1$ and $y_i^{(1)} = -1$ are

$$a^{(1)} - 1 + z_1 > -(a_{-1}^{(1)} y_{i-1}^{(1)} + a_1^{(1)} y_{i+1}^{(1)}), \tag{5}$$

and

$$a^{(1)} - 1 - z_1 > a_{-1}^{(1)} y_{i-1}^{(1)} + a_1^{(1)} y_{i+1}^{(1)}, \tag{6}$$

respectively. Set

$$\begin{aligned}
\mathcal{B}^{(1)}(+) &= \left\{ y_1^{(1)} y^{(1)} y_2^{(1)} : y_1^{(1)}, y_2^{(1)} \text{ satisfy (5)}, y^{(1)} = 1 \right\}, \\
\mathcal{B}^{(1)}(-) &= \left\{ y_1^{(1)} y^{(1)} y_2^{(1)} : y_1^{(1)}, y_2^{(1)} \text{ satisfy (6)}, y^{(1)} = -1 \right\}.
\end{aligned}$$

The basic set of admissible local patterns of the first layer is denoted by $\mathcal{B}^{(1)} = (\mathcal{B}^{(1)}(+), \mathcal{B}^{(1)}(-))$. The solution space $\mathbf{Y}$ of (1) is then described as

$$\mathbf{Y} = \left\{ \mathbf{y} = \begin{pmatrix} y_i^{(2)} \\ y_i^{(1)} \end{pmatrix}_{i \in \mathbb{Z}} : \begin{array}{c} y_i^{(2)} \diamond y_{i-1}^{(1)} y_{i+1}^{(1)} \in \mathcal{B}^{(2)}, \\ y_{i-1}^{(1)} y_i^{(1)} y_{i+1}^{(1)} \in \mathcal{B}^{(1)} \end{array} \right\}. \tag{7}$$

Since the solution space $\mathbf{Y}$ is determined by the basic set of admissible local patterns, these local patterns play an essential role for investigating MNNs. To reveal the complexity of the global patterns, we assign each local pattern its order and introduce the *ordering matrix*. Define

$$\mathbb{X}_2 = \begin{array}{c|cccccccc}
 & \boxed{\substack{- \\ -\ \ -}} & \boxed{\substack{- \\ -\ \ +}} & \boxed{\substack{- \\ +\ \ -}} & \boxed{\substack{- \\ +\ \ +}} & \boxed{\substack{+ \\ -\ \ -}} & \boxed{\substack{+ \\ -\ \ +}} & \boxed{\substack{+ \\ +\ \ -}} & \boxed{\substack{+ \\ +\ \ +}} \\
\hline
\boxed{\substack{- \\ -\ \ -}} & \boxed{\substack{-\,- \\ -\,-\,-\,-}} & \boxed{\substack{-\,- \\ -\,-\,-\,+}} & \boxed{\substack{-\,- \\ -\,+\,-\,-}} & \boxed{\substack{-\,- \\ -\,+\,-\,+}} & \boxed{\substack{-\,+ \\ -\,-\,-\,-}} & \boxed{\substack{-\,+ \\ -\,-\,-\,+}} & \boxed{\substack{-\,+ \\ -\,+\,-\,-}} & \boxed{\substack{-\,+ \\ -\,+\,-\,+}} \\
\boxed{\substack{- \\ -\ \ +}} & \boxed{\substack{-\,- \\ -\,-\,+\,-}} & \boxed{\substack{-\,- \\ -\,-\,+\,+}} & \boxed{\substack{-\,- \\ -\,+\,+\,-}} & \boxed{\substack{-\,- \\ -\,+\,+\,+}} & \boxed{\substack{-\,+ \\ -\,-\,+\,-}} & \boxed{\substack{-\,+ \\ -\,-\,+\,+}} & \boxed{\substack{-\,+ \\ -\,+\,+\,-}} & \boxed{\substack{-\,+ \\ -\,+\,+\,+}} \\
\boxed{\substack{- \\ +\ \ -}} & \boxed{\substack{-\,- \\ +\,-\,-\,-}} & \boxed{\substack{-\,- \\ +\,-\,-\,+}} & \boxed{\substack{-\,- \\ +\,+\,-\,-}} & \boxed{\substack{-\,- \\ +\,+\,-\,+}} & \boxed{\substack{-\,+ \\ +\,-\,-\,-}} & \boxed{\substack{-\,+ \\ +\,-\,-\,+}} & \boxed{\substack{-\,+ \\ +\,+\,-\,-}} & \boxed{\substack{-\,+ \\ +\,+\,-\,+}} \\
\boxed{\substack{- \\ +\ \ +}} & \boxed{\substack{-\,- \\ +\,-\,+\,-}} & \boxed{\substack{-\,- \\ +\,-\,+\,+}} & \boxed{\substack{-\,- \\ +\,+\,+\,-}} & \boxed{\substack{-\,- \\ +\,+\,+\,+}} & \boxed{\substack{-\,+ \\ +\,-\,+\,-}} & \boxed{\substack{-\,+ \\ +\,-\,+\,+}} & \boxed{\substack{-\,+ \\ +\,+\,+\,-}} & \boxed{\substack{-\,+ \\ +\,+\,+\,+}} \\
\boxed{\substack{+ \\ -\ \ -}} & \boxed{\substack{+\,- \\ -\,-\,-\,-}} & \boxed{\substack{+\,- \\ -\,-\,-\,+}} & \boxed{\substack{+\,- \\ -\,+\,-\,-}} & \boxed{\substack{+\,- \\ -\,+\,-\,+}} & \boxed{\substack{+\,+ \\ -\,-\,-\,-}} & \boxed{\substack{+\,+ \\ -\,-\,-\,+}} & \boxed{\substack{+\,+ \\ -\,+\,-\,-}} & \boxed{\substack{+\,+ \\ -\,+\,-\,+}} \\
\boxed{\substack{+ \\ -\ \ +}} & \boxed{\substack{+\,- \\ -\,-\,+\,-}} & \boxed{\substack{+\,- \\ -\,-\,+\,+}} & \boxed{\substack{+\,- \\ -\,+\,+\,-}} & \boxed{\substack{+\,- \\ -\,+\,+\,+}} & \boxed{\substack{+\,+ \\ -\,-\,+\,-}} & \boxed{\substack{+\,+ \\ -\,-\,+\,+}} & \boxed{\substack{+\,+ \\ -\,+\,+\,-}} & \boxed{\substack{+\,+ \\ -\,+\,+\,+}} \\
\boxed{\substack{+ \\ +\ \ -}} & \boxed{\substack{+\,- \\ +\,-\,-\,-}} & \boxed{\substack{+\,- \\ +\,-\,-\,+}} & \boxed{\substack{+\,- \\ +\,+\,-\,-}} & \boxed{\substack{+\,- \\ +\,+\,-\,+}} & \boxed{\substack{+\,+ \\ +\,-\,-\,-}} & \boxed{\substack{+\,+ \\ +\,-\,-\,+}} & \boxed{\substack{+\,+ \\ +\,+\,-\,-}} & \boxed{\substack{+\,+ \\ +\,+\,-\,+}} \\
\boxed{\substack{+ \\ +\ \ +}} & \boxed{\substack{+\,- \\ +\,-\,+\,-}} & \boxed{\substack{+\,- \\ +\,-\,+\,+}} & \boxed{\substack{+\,- \\ +\,+\,+\,-}} & \boxed{\substack{+\,- \\ +\,+\,+\,+}} & \boxed{\substack{+\,+ \\ +\,-\,+\,-}} & \boxed{\substack{+\,+ \\ +\,-\,+\,+}} & \boxed{\substack{+\,+ \\ +\,+\,+\,-}} & \boxed{\substack{+\,+ \\ +\,+\,+\,+}}
\end{array}$$

and

$$\mathbb{X}_1 = \begin{array}{cc} & \begin{array}{cccc} \boxed{--} & \boxed{-+} & \boxed{+-} & \boxed{++} \end{array} \\ \begin{array}{c} \boxed{--} \\ \boxed{-+} \\ \boxed{+-} \\ \boxed{++} \end{array} & \left(\begin{array}{cccc} \boxed{---} & \boxed{--+} & \varnothing & \varnothing \\ \varnothing & \varnothing & \boxed{-+-} & \boxed{-++} \\ \boxed{+--} & \boxed{+-+} & \varnothing & \varnothing \\ \varnothing & \varnothing & \boxed{++-} & \boxed{+++} \end{array} \right) \end{array}.$$

Herein $\varnothing$ means no pattern is generated. Suppose $\mathcal{B}(\mathbb{T}) = (\mathcal{B}^{(1)}, \mathcal{B}^{(2)})$ is the basic set of admissible local patterns of (1) with respect to the template $\mathbb{T}$. Set $\mathbb{X}_i(p, q) = \varnothing$ if $\mathbb{X}_i(p, q) \notin \mathcal{B}^{(i)}$ for $i = 1, 2$. The symbolic product of the ordering matrices then assert all possibilities of global patterns that are generated by the admissible local patterns. See [47] for more details. Aside from elucidating the set of global patterns, we are also interested in the number of global patterns. For this purpose, we consider the *transition matrix*.

The transition matrix $\mathbf{T}$ is defined by

$$\mathbf{T}(i, j) = \begin{cases} 1, & p_i, p_j \in \mathcal{B}^{(2)} \text{ and} \\ & \alpha_{i-1}\alpha_{j-1}\alpha_{i+1}, \alpha_{j-1}\alpha_{i+1}\alpha_{j+1} \in \mathcal{B}^{(1)}; \\ 0, & \text{otherwise}; \end{cases} \tag{8}$$

herein p_k defined in (2), and presented as $\alpha_k \diamond \alpha_{k-1}\alpha_{k+1}$ for $k = 1, \ldots, 8$. Furthermore, the transition matrix of the second layer $T_2 \in \mathcal{M}_{8\times 8}(\{0, 1\})$ is defined by

$$T_2(i, j) = 1 \quad \text{if and only if} \quad p_i, p_j \in \mathcal{B}^{(2)}, \tag{9}$$

while the transition matrix of the first layer $T_1 \in \mathcal{M}_{4\times 4}(\{0, 1\})$ is defined by

$$T_1(i, j) = 1 \quad \text{if and only if} \quad \mathbb{X}_1(i, j) \in \mathcal{B}^{(1)}. \tag{10}$$

Denote $T_1^2 = (T_{i,j})_{i,j=1}^2$ as four smaller 2×2 matrices and define $\overline{T}_1$ by

$$\overline{T}_1(p, q) = T_{i,j}(k, l), \quad \text{where} \quad p = 2i + j - 2, q = 2k + l - 2. \tag{11}$$

Ban and Chang [47] showed that

$$\mathbf{T} = T_2 \circ (E_2 \otimes \overline{T}_1), \tag{12}$$

where E_k is a $k \times k$ matrix with all entries being 1's, $\circ$ and $\otimes$ are the Hadamard and Kronecker products, respectively. Furthermore, the output space $\mathbf{Y}^{(2)}$ is topologically conjugated to a so-called *sofic shift* in symbolic dynamical systems (cf. [47, Theorem 2.4]).

To make this manuscript self-contained, the following is a brief introduction of sofic shifts in symbolic dynamical systems. A detailed instruction for symbolic dynamical systems is referred to in [48].

A *labeled graph* $\mathcal{G} = (G, \mathcal{L})$ consists of an underlying graph $G = (\mathcal{V}, \mathcal{E})$ and the *labeling* $\mathcal{L} : \mathcal{E} \to \mathcal{A}$ which assigns to each edge a label from the finite alphabet $\mathcal{A}$, where $\mathcal{V}$ and $\mathcal{E}$ refer to the sets of vertices and edges, respectively. A sofic shift $\mathbf{X}$ is defined by

$$\mathbf{X} = \{(\omega_i)_{i\in\mathbb{Z}} : \omega_i = \mathcal{L}(e_i), e_i \in \mathcal{E}, \text{ter}(e_i) = \text{init}(e_{i+1})\}$$

for some labeled graph $\mathcal{G}$, where ter(e) and init(e) mean the terminal and initial vertices of the edge $e \in \mathcal{E}$, respectively. Without loss of generality, we may assume that there is at most one edge connecting two vertices. The transition matrix $\mathbf{T}_{\mathcal{G}}$ of the labeled graph $\mathcal{G}$ is indexed by the vertices $\mathcal{V}$ and $\mathbf{T}_{\mathcal{G}}(p, q) = 1$ if and only if there is an edge from p to q. Set the alphabet $\mathcal{A} = \{a_{00}, a_{01}, a_{10}, a_{11}\}$, where

$$a_{00} = --, a_{01} = -+, a_{10} = +-, a_{11} = ++\,.$$

Define the *symbolic transition matrix* as

$$\mathbf{S} = \left(\begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} \otimes E_4 \right) \circ \mathbf{T}, \quad \mathbf{S}(i, j) = \varnothing \text{ if } \mathbf{T}(i, j) = 0. \tag{13}$$

Let $\mathcal{V} = \{p_1, \ldots, p_8\}$, and $e_{ij} \in \mathcal{E}$ if $\operatorname{init}(e_{ij}) = p_i$, $\operatorname{ter}(e_{ij}) = p_j$ and $\mathbf{T}(i, j) = 1$. Define $\mathcal{L} : \mathcal{E} \to \mathcal{A}$ by

$$\mathcal{L}(e_{ij}) = a_{\overline{i}\,\overline{j}}, \quad \text{where } \overline{k} = \left\lfloor \frac{k-1}{4} \right\rfloor,$$

where $\lfloor \cdot \rfloor$ is the Gauss function. Let $\mathcal{G} = (G, \mathcal{L})$ be the labeled graph with underlying graph $G = (\mathcal{V}, \mathcal{E})$ and labeling $\mathcal{L}$. It can be seen that the output space $\mathbf{Y}^{(2)} = \mathbf{X}$ is the sofic shift defined by $\mathcal{G}$.

One of the most frequently used quantums for the measure of spatial complexity is the *topological entropy*. Let X be a symbolic space and let $\Gamma_n(X)$ denote the number of the patterns in X of length n. The topological entropy of X is defined by

$$h(X) = \lim_{n\to\infty} \frac{\log \Gamma_n(X)}{n}, \quad \text{provided the limit exists.}$$

The space X is called *pattern formation* if $h(X) = 0$, and *spatial chaos* otherwise. Similarly to [42], it can be verified that the topological entropy of the output space of (1) is $h(\mathbf{Y}^{(2)}) = \log \rho_{\mathbf{T}}$ if the labeled graph constructed from $\mathbf{T}$ is *right-resolving*, where $\rho_{\mathbf{T}}$ is the spectral radius of $\mathbf{T}$. Here a labeled graph $\mathcal{G} = (G, \mathcal{L})$ is called right-resolving if the restriction of $\mathcal{L}$ to $\mathcal{E}_I$ is one-to-one, where $\mathcal{E}_I$ consists of those edges starting from I.

If $\mathcal{G}$ is not right-solving, there exists a labeled graph $\mathcal{H}$, derived by applying the *subset construction method* (SCM) to $\mathcal{G}$, such that the sofic shift defined by $\mathcal{H}$ is identical to the original space. The new labeled graph $\mathcal{H} = (H, \mathcal{L}')$ is constructed as follows:

The vertices I of H are the nonempty subsets of the vertex set $\mathcal{V}$ of G. If $I \in \mathcal{V}'$ and $a \in \mathcal{A}$, let J denote the set of terminal vertices of edges in G starting at some vertices in I and labeled a, i.e., J is the set of vertices reachable from I using the edges labeled a. There are two cases.

1) If $J = \varnothing$, do nothing.
2) If $J \neq \varnothing$, $J \in \mathcal{V}'$ and draw an edge in H from I to J labeled a.

Carry this out for each $I \in \mathcal{V}'$ with each $a \in \mathcal{A}$ producing the labeled graph $\mathcal{H}$. Then, each vertex I in H has at most one edge with a given label starting at I. This implies that $\mathcal{H}$ is right-resolving. Theorem 2.1 indicates that the topological entropy of the output space $\mathbf{Y}^{(2)}$ is related to the spectral radius of the transition matrix of (1).

Theorem 2.1 (See [47, Theorem 2.6]) *Let $\mathcal{G}$ be the labeled graph obtained from the transition matrix $\mathbf{T}$ of (1). The topological entropy of the output space $\mathbf{Y}^{(2)}$ is*

$$h(\mathbf{Y}^{(2)}) = \begin{cases} \log \rho_{\mathbf{T}}, & \textit{if } \mathcal{G} \textit{ is right-resolving;} \\ \log \rho_{\mathbf{H}}, & \textit{otherwise;} \end{cases} \tag{14}$$

where $\mathbf{H}$ is the transition matrix of the labeled graph $\mathcal{H}$, which is obtained by applying SCM to $\mathcal{G}$.

Example 2.2 (See [47, Example 2.7]) Suppose $\mathbb{T} = (\mathbf{A}, \mathbf{B}, \mathbf{z})$ with $A_1 = (2.2, 1.7)$, $A_2 = (-4, -2)$, $\mathbf{B} = (-2.6, -1.4)$, and $\mathbf{z} = (-1.2, 0.3)$. The transition matrices for the first and second layer are

$$T_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

and

$$T_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

respectively. Furthermore, it is seen that

$$\overline{T}_1 = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

and

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

Applying symbols on the transition matrix derives the symbolic transition matrix of the MNN as

$$\mathbf{S} = \begin{pmatrix} \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & a_{01} & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & \varnothing & a_{01} & a_{01} & \varnothing & \varnothing \\ \varnothing & \varnothing & a_{10} & \varnothing & \varnothing & a_{11} & \varnothing & \varnothing \\ \varnothing & \varnothing & a_{10} & \varnothing & \varnothing & a_{11} & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \end{pmatrix}.$$

Since the labeled graph $\mathcal{G}$, which is obtained from $\mathbf{T}$, is not right-resolving, applying the subset construction method to $\mathcal{G}$ derives a right-resolving labeled graph $\mathcal{H}$ (cf. Figure 2). The transition matrix of $\mathcal{H}$, indexed by $p_3, p_4, p_5, p_6, \{p_5, p_6\}$, is

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

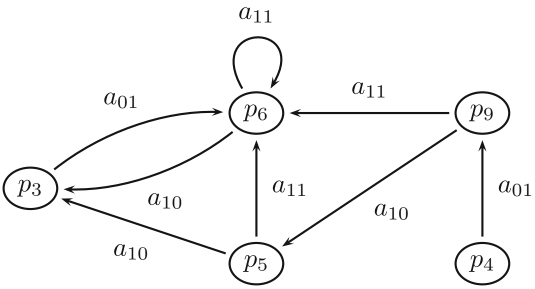

Fig. 2 The labeled graph $\mathcal{H}$ obtained by applying SCM to $\mathcal{G}$. An extra vertex $p_9 = \{p_5, p_6\}$ is created so that $\mathcal{H}$ is right-resolving

Theorem 2.1 indicates that the topological entropy of the output space $\mathbf{Y}^{(2)}$ is $h(\mathbf{Y}^{(2)}) = \log \rho_{\mathbf{H}} = \log g$, where $g = \frac{1+\sqrt{5}}{2}$ is the golden mean.

3 Inhomogeneous Multi-layer Neural Networks

Based on the discussion of homogeneous MNNs, we now consider IHMNNs. Traditionally, the template for MNN is homogeneous (also known as isotropic), i.e., the template is space-invariant. However, there are more and more MNNs that use inhomogeneous templates to describe some of the problems that arise from biological and ecological contexts. This motivates the investigation of the topological complexity of the output spaces of IHMNNs.

A one-dimensional IHMNN is realized as

$$\begin{cases} \frac{d}{dt}x_i^{(k)}(t) = -x_i^{(k)}(t) + z_i^{(k)} + a_i^{(k)} f(x_i^{(k)}(t)) + \sum\limits_{\ell \in \mathcal{N}_{i,k}} b_\ell^{(k)} f(x_{i+\ell}^{(k-1)}(t)), \\ \frac{d}{dt}x_i^{(1)}(t) = -x_i^{(1)}(t) + z_i^{(1)} + a_i^{(1)} f(x_i^{(1)}(t)) + \sum\limits_{\ell \in \mathcal{N}_{i,1}} a_\ell^{(1)} f(x_{i+\ell}^{(1)}(t)), \end{cases} \tag{15}$$

for some $N \in \mathbb{N}$, $k = 2, \ldots, N$ and $i \in \mathbb{Z}$. The template $\mathbb{T} = [\mathbb{T}_i]_{i \in \mathbb{Z}}$ consists of infinite sub-templates $\mathbb{T}_i = [\mathbf{A}_i, \mathbf{B}_i, \mathbf{z}_i]$, where $\mathbf{A}_i = (A_{i;1}, A_{i;2})$ with $A_{i;1} = (a_i^{(1)}, \ldots, a_i^{(N)})$, $A_{i;2} = (a_\ell^{(1)})_{\ell \in \mathcal{N}_{1,i}}$, $\mathbf{B} = (B_{i;2}, \ldots, B_{i;N})$ with $B_{i;k} = (b_\ell^{(k)})_{\ell \in \mathcal{N}_{i,k}}$, and $\mathbf{z}_i = (z_i^{(1)}, \ldots, z_i^{(N)})$. The neighborhood $\mathbf{N} = [\mathcal{N}_{i,1}, \ldots, \mathcal{N}_{i,N}]_{i \in \mathbb{Z}}$ also consists of infinitely many components. An IHMNN is called a *constant* IHMNN (CIHMNN) if the neighborhood $\mathbf{N}$ and the template $\mathbb{T}$ are periodic up to shifts. More precisely, there exists a positive integer $L \in \mathbb{N}$ such that $\mathbf{N}' = [\mathcal{N}'_{i,1}, \ldots, \mathcal{N}'_{i,N}]_{i \in \mathbb{Z}}$ and $\mathbb{T}' = [\mathbb{T}'_i]_{i \in \mathbb{Z}}$ satisfy $\mathcal{N}_{i+jL,k} = \mathcal{N}'_{i,k}$ and $\mathbb{T}'_{i+jL} = \mathbb{T}'_i$ for $i, j \in \mathbb{Z}$ and $1 \le k \le N$, where

$$\mathbf{K}'_i = \mathbf{K}_i - i = \{j - i : j \in \mathbf{K}_i\}, \quad \mathbf{K} = \mathcal{N}, \mathbb{T}.$$

It can be seen that CIHMNNs generalize the concept of the classical MNNs. More precisely, a classical MNN is a CIHMNN with $L = 1$. The essential description of a one-dimensional CIHMNN is presented in the following form:

$$\begin{cases} \frac{d}{dt}x_i^{(k)}(t) = -x_i^{(k)}(t) + z_{\bar{i}}^{(k)} + a_{\bar{i}}^{(k)} f(x_i^{(k)}(t)) + \sum\limits_{\ell \in \mathcal{N}_{\bar{i},k}} b_\ell^{(k)} f(x_{i+\ell L}^{(k-1)}(t)), \\ \frac{d}{dt}x_i^{(1)}(t) = -x_i^{(1)}(t) + z_{\bar{i}}^{(1)} + a_{\bar{i}}^{(1)} f(x_i^{(1)}(t)) + \sum\limits_{\ell \in \mathcal{N}_{\bar{i},1}} a_\ell^{(1)} f(x_{i+\ell L}^{(1)}(t)), \end{cases} \tag{16}$$

where $k = 2, \ldots, N$, $1 \leq \bar{i} \leq L$, and $\bar{i} = i \pmod{L}$. To clarify the discussion, we assume that $N = 2$ and $\mathcal{N}_{\bar{i}} = \{-1, 1\}$ is the nearest neighborhood for $1 \leq \bar{i} \leq L$.

Let $\Lambda = \{1, \ldots, L\}$ be a finite index set. The one-dimensional lattice $\mathbb{Z}$ can be decomposed into L non-overlapping subspaces

$$\mathbb{Z} = \bigcup_{j \in \Lambda} \mathbb{Z}_j = \bigcup_{j \in \Lambda} \{j_i, i \in \mathbb{Z}\}, \{m : m = CL + j, C \in \mathbb{Z}\}$$

where $j_i = j + iL$. Thus, (16) immediately can then be restated as

$$\begin{cases} \dfrac{d}{dt} x_{j_i}^{(2)}(t) = -x_{j_i}^{(2)}(t) + z_j^{(2)} + a_j^{(2)} f(x_{j_i}^{(2)}(t)) + b_{j;-1}^{(2)} f(x_{j_{i-1}}^{(1)}(t)) + b_{j;1}^{(2)} f(x_{j_{i+1}}^{(1)}(t)), \\ \dfrac{d}{dt} x_{j_i}^{(1)}(t) = -x_{j_i}^{(1)}(t) + z_j^{(1)} + a_j^{(1)} f(x_{j_i}^{(1)}(t)) + a_{j;-1}^{(1)} f(x_{j_{i-1}}^{(1)}(t)) + a_{j;1}^{(1)} f(x_{j_{i+1}}^{(1)}(t)), \end{cases} \tag{17}$$

for $j \in \Lambda, i \in \mathbb{Z}$. Similarly to the discussion in the previous section, for a fixed $j \in \Lambda$, there exist basic sets of admissible local patterns of the first and second layers, say, $\mathcal{B}_j^{(1)} = (\mathcal{B}_j^{(1)}(+), \mathcal{B}_j^{(1)}(-))$ and $\mathcal{B}_j^{(2)} = (\mathcal{B}_j^{(2)}(+), \mathcal{B}_j^{(2)}(-))$, respectively. The basic set of admissible local patterns $\mathcal{B}_j = (\mathcal{B}_j^{(1)}, \mathcal{B}_j^{(2)})$ determines the solution space $\mathbf{Y}_j$ as

$$\mathbf{Y}_j = \left\{ \mathbf{y} = \begin{pmatrix} y_{j_i}^{(2)} \\ y_{j_i}^{(1)} \end{pmatrix}_{i \in \mathbb{Z}} : \begin{array}{l} y_{j_i}^{(2)} \diamond y_{j_{i-1}}^{(1)} y_{j_{i+1}}^{(1)} \in \mathcal{B}_j^{(2)}, \\ y_{j_{i-1}}^{(1)} y_{j_i}^{(1)} y_{j_{i+1}}^{(1)} \in \mathcal{B}_j^{(1)} \end{array} \right\}. \tag{18}$$

It follows that, for $1 \leq j \leq L$, $\mathbf{Y}_j$ is the solution space $\mathbf{Y}$ defined in (7) after relabeling. Hence $\mathbf{Y}_j$ is topologically conjugated to the solution space of a traditional two-layer neural network. Notably, (17) infers that a CIHMNN is decomposed into L independent MNNs. More precisely, let $\phi_j : \mathbf{Y} \to \mathbf{Y}_j$ be the canonical projection for $1 \leq j \leq L$. A straightforward examination demonstrates that

$$\phi : \mathbf{Y} \to \mathbf{Y}_1 \times \mathbf{Y}_2 \times \cdots \times \mathbf{Y}_L$$

defined by $\phi = (\phi_1, \phi_2, \ldots, \phi_L)$ is an isomorphism, where × is the Cartesian product. If one replaces $\mathbf{Y}$ and $\mathbf{Y}_j$ by $\mathbf{Y}^{(2)}$ and $\mathbf{Y}_l^{(2)}$, respectively, for $1 \leq j \leq L$, analogous discussion leads to topological conjugacy between $\mathbf{Y}^{(2)}$ and $\mathbf{Y}_1^{(2)} \times \cdots \times \mathbf{Y}_L^{(2)}$. Theorem 3.1 then follows.

Theorem 3.1 *Suppose* $\mathbf{Y}$ *and* $\mathbf{Y}^{(2)}$ *(respectively* $\mathbf{Y}_j$ *and* $\mathbf{Y}_j^{(2)}$*) are the solution and output spaces of* (16) *(respectively* (18) *), respectively. Then*

$$\mathbf{Y} \cong \mathbf{Y}_1 \times \cdots \times \mathbf{Y}_L \quad \textit{and} \quad \mathbf{Y}^{(2)} \cong \mathbf{Y}_1^{(2)} \times \cdots \times \mathbf{Y}_L^{(2)}, \tag{19}$$

where × *is the Cartesian product.*

The ordering matrix of $\mathbf{Y}_j$ is defined by

$$\mathbb{X}_2' = \begin{array}{c|cccccccc}
 & \boxed{\begin{smallmatrix} & - & \\ - & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & - & \\ - & & + \end{smallmatrix}} & \boxed{\begin{smallmatrix} & - & \\ + & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & - & \\ + & & + \end{smallmatrix}} & \boxed{\begin{smallmatrix} & + & \\ - & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & + & \\ - & & + \end{smallmatrix}} & \boxed{\begin{smallmatrix} & + & \\ + & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & + & \\ + & & + \end{smallmatrix}} \\
\hline
\boxed{\begin{smallmatrix} & - & \\ - & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & - & & - & \\ - & & - & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & - & & - & \\ - & & - & & + \end{smallmatrix}} & \varnothing & \varnothing & \boxed{\begin{smallmatrix} & - & & + & \\ - & & - & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & - & & + & \\ - & & - & & + \end{smallmatrix}} & \varnothing & \varnothing \\
\boxed{\begin{smallmatrix} & - & \\ - & & + \end{smallmatrix}} & \varnothing & \varnothing & \boxed{\begin{smallmatrix} & - & & - & \\ - & & + & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & - & & - & \\ - & & + & & + \end{smallmatrix}} & \varnothing & \varnothing & \boxed{\begin{smallmatrix} & - & & + & \\ - & & + & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & - & & + & \\ - & & + & & + \end{smallmatrix}} \\
\boxed{\begin{smallmatrix} & - & \\ + & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & - & & - & \\ + & & - & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & - & & - & \\ + & & - & & + \end{smallmatrix}} & \varnothing & \varnothing & \boxed{\begin{smallmatrix} & - & & + & \\ + & & - & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & - & & + & \\ + & & - & & + \end{smallmatrix}} & \varnothing & \varnothing \\
\boxed{\begin{smallmatrix} & - & \\ + & & + \end{smallmatrix}} & \varnothing & \varnothing & \boxed{\begin{smallmatrix} & - & & - & \\ + & & + & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & - & & - & \\ + & & + & & + \end{smallmatrix}} & \varnothing & \varnothing & \boxed{\begin{smallmatrix} & - & & + & \\ + & & + & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & - & & + & \\ + & & + & & + \end{smallmatrix}} \\
\boxed{\begin{smallmatrix} & + & \\ - & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & + & & - & \\ - & & - & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & + & & - & \\ - & & - & & + \end{smallmatrix}} & \varnothing & \varnothing & \boxed{\begin{smallmatrix} & + & & + & \\ - & & - & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & + & & + & \\ - & & - & & + \end{smallmatrix}} & \varnothing & \varnothing \\
\boxed{\begin{smallmatrix} & + & \\ - & & + \end{smallmatrix}} & \varnothing & \varnothing & \boxed{\begin{smallmatrix} & + & & - & \\ - & & + & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & + & & - & \\ - & & + & & + \end{smallmatrix}} & \varnothing & \varnothing & \boxed{\begin{smallmatrix} & + & & + & \\ - & & + & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & + & & + & \\ - & & + & & + \end{smallmatrix}} \\
\boxed{\begin{smallmatrix} & + & \\ + & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & + & & - & \\ + & & - & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & + & & - & \\ + & & - & & + \end{smallmatrix}} & \varnothing & \varnothing & \boxed{\begin{smallmatrix} & + & & + & \\ + & & - & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & + & & + & \\ + & & - & & + \end{smallmatrix}} & \varnothing & \varnothing \\
\boxed{\begin{smallmatrix} & + & \\ + & & + \end{smallmatrix}} & \varnothing & \varnothing & \boxed{\begin{smallmatrix} & + & & - & \\ + & & + & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & + & & - & \\ + & & + & & + \end{smallmatrix}} & \varnothing & \varnothing & \boxed{\begin{smallmatrix} & + & & + & \\ + & & + & & - \end{smallmatrix}} & \boxed{\begin{smallmatrix} & + & & + & \\ + & & + & & + \end{smallmatrix}}
\end{array},$$

while $\mathbb{X}_1$ is the same as that defined in the previous section. Similar to the discussion above, the transition matrix of $\mathbf{Y}_j$ is defined by

$$\mathbf{T}_j(m,n) = \begin{cases} 1, \ p_m, p_n \in \mathcal{B}_j^{(2)}, \alpha_{m-1}\alpha_{n-1}\alpha_{n+1} \in \mathcal{B}^{(1)}, \\ \quad \text{and } \alpha_{n-1} = \alpha_{m+1}; \\ 0, \ \text{otherwise}. \end{cases} \tag{20}$$

Suppose $T_{j,1}$ and $T_{j,2}$ are the transition matrices of the first and second layers of $\mathbf{Y}_j$, respectively. It is an immediate result from the definition of $\mathbb{X}_1$ and $\mathbb{X}_2'$ that

$$\mathbf{T}_j = T_{j,2} \circ (E_2 \otimes T_{j,1}) \quad \text{for} \quad j \in \Lambda. \tag{21}$$

Theorem 3.2, which comes immediately from Theorem 3.1, reveals the spatial complexity of a CIHMNN.

Theorem 3.2 *Suppose* $\mathbf{Y}^{(2)}$ *is the output space of* (16) *with subspaces* $\mathbf{Y}_j^{(2)}$ *defined in* (18), $1 \leq j \leq L$. *Then the topological entropy of* $\mathbf{Y}^{(2)}$ *is*

$$h(\mathbf{Y}^{(2)}) = \frac{1}{L}(h(\mathbf{Y}_1^{(2)}) + \cdots + h(\mathbf{Y}_L^{(2)})). \tag{22}$$

Similarly, the topological entropy of the solution space $\mathbf{Y}$ *is*

$$h(\mathbf{Y}) = \frac{1}{L}(h(\mathbf{Y}_1) + \cdots + h(\mathbf{Y}_L)). \tag{23}$$

Moreover, let ρ_j *be the spectral radius of* $\mathbf{T}_j$ *and let* ϱ_j *be the spectral radius of* $\mathbf{H}_j$ *for* $1 \leq j \leq L$, *where* $\mathbf{H}_j$ *is the transition matrix of the labeled graph obtained by applying SCM to the labeled graph of* $\mathbf{Y}_j^{(2)}$. *Then*

$$h(\mathbf{Y}) = \frac{1}{L}\sum_{j=1}^{L} \rho_j, \quad h(\mathbf{Y}^{(2)}) = \frac{1}{L}\sum_{j=1}^{L} \varrho_j.$$

Since the output space of a CIHMNN consists of binary patterns, the topological entropy is either zero or is less than or equal to log 2. Thus, it is natural to ask whether or not the *entropy set*, i.e., the collection of the topological entropies of all output spaces, of CIHMNNs is dense in the closed interval [0, log 2]. Denseness of the entropy set indicates that CIHMNNs are capable of exhibiting "arbitrary" phenomena. In other words, CIHMNNs are universal in some sense. The upcoming corollary asserts an affirmative result.

Table 1 The topological entropy $h(\mathbf{Y}^{(2)})$ of CIHMNNs with 2 -components and the templates being given by $(a_1^{(1)}, z_1^{(1)}, a_{1;-1}^{(1)}, a_{1;1}^{(1)}) = (a_2^{(1)}, z_2^{(1)}, a_{2;-1}^{(1)}, a_{2;1}^{(1)}) = (1.5, -2, 2, -4)$ and $(b_{1;-1}^{(2)}, b_{1;1}^{(2)}) = (b_{2;-1}^{(2)}, b_{2;1}^{(2)}) = (1, 3)$

	0	$\log\sqrt{2}$	$\log 2$
0	0	$\log\sqrt[4]{2}$	$\log\sqrt{2}$
$\log\sqrt{2}$	$\log\sqrt[4]{2}$	$\log\sqrt{2}$	$\log\sqrt[4]{8}$
$\log 2$	$\log\sqrt{2}$	$\log\sqrt[4]{8}$	$\log 2$

The topological entropies $h(\mathbf{Y}_1^{(2)})$ and $h(\mathbf{Y}_2^{(2)})$ as the parameters $a_1^{(2)}, a_2^{(2)}, z_1^{(2)}$, and $z_2^{(2)}$ vary, are listed in the column and row, respectively. A richer choice of topological entropies is observed

Corollary 3.3 *The set of topological entropies of CIHMNNs is dense in the closed interval* $[0, \log 2]$. *More precisely, given* $\epsilon > 0$ *and* $\lambda \in [0, \log 2]$, *there exists a CIHMNN such that* $|h(\mathbf{Y}^{(2)}) - \lambda| < \epsilon$.

Proof It suffices to show that, for $1 \leq \ell \leq 2^k$ and $k \in \mathbb{N}$, there exists a CIHMNN satisfying

$$h(\mathbf{Y}^{(2)}) = \frac{\ell}{2^k} \log 2.$$

To achieve this, let $L = 2^k$, $\mathbf{Y}_j^{(2)}$ be the full 2-shift for $1 \leq j \leq \ell$, and let $\mathbf{Y}_j^{(2)}$ consist of either $(-)^\infty$ or $(+)^\infty$ for $\ell + 1 \leq j \leq 2^k$. Theorem 3.2 demonstrates that

$$\begin{aligned} h(\mathbf{Y}^{(2)}) &= \frac{1}{2^k}(h(\mathbf{Y}_1^{(2)}) + \cdots + h(\mathbf{Y}_{2^k}^{(2)})) \\ &= \frac{1}{2^k}\sum_{j=1}^{\ell} \log 2 = \frac{\ell}{2^k} \log 2. \end{aligned}$$

This completes the proof. □

Example 3.4 Suppose the templates of CIHMNNs are given by

$$\begin{aligned} (a_1^{(1)}, z_1^{(1)}, a_{1;-1}^{(1)}, a_{1;1}^{(1)}) &= (a_2^{(1)}, z_2^{(1)}, a_{2;-1}^{(1)}, a_{2;1}^{(1)}) \\ &= (1.5, -2, 2, -4) \end{aligned}$$

and

$$(b_{1;-1}^{(2)}, b_{1;1}^{(2)}) = (b_{2;-1}^{(2)}, b_{2;1}^{(2)}) = (1, 3).$$

It is seen that, for all possible choices of $(a_1^{(2)}, z_1^{(2)})$ and $(a_2^{(2)}, z_2^{(2)})$, the topological entropies of $\mathbf{Y}_1^{(2)}$ and $\mathbf{Y}_2^{(2)}$ are a subset of $\{0, \log\sqrt{2}, \log 2\}$. The topological entropy of the output space $\mathbf{Y}^{(2)}$ such that $\mathbf{Y}^{(2)} \cong \mathbf{Y}_1^{(2)} \times \mathbf{Y}_2^{(2)}$ is presented in Table 1.

Remark 3.5 An immediate extension of Theorems 3.1, 3.2, and Corollary 3.3 is that, given an inhomogeneous N-layer neural network (16), for $0 \leq \ell \leq N$,

a. $\mathbf{Y}^{(\ell)} \cong \mathbf{Y}_1^{(\ell)} \times \cdots \times \mathbf{Y}_L^{(\ell)}$;

b. $h(\mathbf{Y}^{(\ell)}) = \frac{1}{L}\sum_{j=1}^{L} h(\mathbf{Y}_j^{(\ell)}) = \frac{1}{L}\sum_{j=1}^{L} \log \rho_j^{(\ell)}$;

herein $\mathbf{Y}^{(0)}$ refers to $\mathbf{Y}$, and $\rho_j^{(\ell)}$ is the spectral radius of the transition matrix of $\mathbf{Y}_j^{(\ell)}$ after applying SCM to its corresponding labeled graph. Furthermore,

c. the entropy set $\{h(\mathbf{Y}^{(\ell)})\}$ is dense in the closed interval $[0, \log 2]$.

To see the affirmation of statements in Remark 3.5, one can tell that a CIHMNN (16) with the nearest neighborhood $\mathcal{N}_{\bar{i}} = \{-1, 1\}$ for $1 \le \bar{i} \le L$ and $\bar{i} = i \pmod{L}$ can be restated as

$$\begin{cases} \dfrac{d}{dt} x_{j_i}^{(k)}(t) = -x_{j_i}^{(k)}(t) + z_j^{(k)} + a_j^{(k)} f(x_{j_i}^{(k)}(t)) + b_{j;-1}^{(k)} f(x_{j_{i-1}}^{(1)}(t)) + b_{j;1}^{(k)} f(x_{j_{i+1}}^{(1)}(t)), \\ \dfrac{d}{dt} x_{j_i}^{(1)}(t) = -x_{j_i}^{(1)}(t) + z_j^{(1)} + a_j^{(1)} f(x_{j_i}^{(1)}(t)) + a_{j;-1}^{(1)} f(x_{j_{i-1}}^{(1)}(t)) + a_{j;1}^{(1)} f(x_{j_{i+1}}^{(1)}(t)), \end{cases}$$

where $2 \le k \le N$, $j \in \Lambda = \{1, \ldots, L\}$, and $i \in \mathbb{Z}$. Similar discussion on the elucidation of Theorem 3.1 approves Remark 3.5(a), i.e.,

$$\mathbf{Y}^{(\ell)} \cong \mathbf{Y}_1^{(\ell)} \times \cdots \times \mathbf{Y}_L^{(\ell)}$$

for $0 \le \ell \le N$. Following Remark 3.5(a), Remark 3.5(b) and (c) reveal the explicit formula of the topological entropy of the hidden/output/solution space and the capability of universal machines for CIHMNNs.

To see that Remark 3.5 holds for an arbitrary neighborhood, an inhomogeneous two-layer neural network with two-nearest neighborhood $\mathcal{N}_{\bar{i}} = \{-2, -1, 1, 2\} \equiv \mathcal{N}$ is addressed to clarify the methodology. The generalized investigation follows by combining the upcoming illustration and the recursive formulae in [47]. In this case, (16) is expressed as

$$\begin{cases} \dfrac{d}{dt} x_{j_i}^{(2)}(t) = -x_{j_i}^{(2)}(t) + z_j^{(2)} + a_j^{(2)} f(x_{j_i}^{(2)}(t)) + \displaystyle\sum_{\ell \in \mathcal{N}} b_{j;\ell}^{(2)} f(x_{j_{i+\ell}}^{(1)}(t)), \\ \dfrac{d}{dt} x_{j_i}^{(1)}(t) = -x_{j_i}^{(1)}(t) + z_j^{(1)} + a_j^{(1)} f(x_{j_i}^{(1)}(t)) + \displaystyle\sum_{\ell \in \mathcal{N}} a_{j;\ell}^{(1)} f(x_{j_{i+\ell}}^{(1)}(t)), \end{cases}$$

for $j \in \Lambda = \{1, \ldots, L\}$, $i \in \mathbb{Z}$, and $j_i = j + Li$. Analogous to the study of Theorem 3.1, the solution space $\mathbf{Y}_j$ is realized as

$$\mathbf{Y}_j = \left\{ y = \begin{pmatrix} y_{j_i}^{(2)} \\ y_{j_i}^{(1)} \end{pmatrix} : \begin{array}{r} y_{j_i}^{(2)} \diamond y_{j_{i-2}}^{(1)} y_{j_{i-1}}^{(1)} y_{j_{i+1}}^{(1)} y_{j_{i+2}}^{(1)} \in \mathcal{B}_j^{(2)}, \\ y_{j_{i-2}}^{(1)} y_{j_{i-1}}^{(1)} y_{j_i}^{(1)} y_{j_{i+1}}^{(1)} y_{j_{i+2}}^{(1)} \in \mathcal{B}_j^{(1)} \end{array} \right\},$$

where $\mathcal{B}_j^{(1)}$ and $\mathcal{B}_j^{(2)}$ is the basic set of admissible local patterns of the first and second layer, respectively. The ordering matrix $\mathbb{X}_2$, indexed by

$$\{\alpha \diamond \beta_{-2}\beta_{-1}\beta_1\beta_2 : \alpha, \beta_i \in \{-, +\}, i = -2, -1, 1, 2\},$$

is defined as

$$\mathbb{X}_2(\alpha \diamond \beta_{-2}\beta_{-1}\beta_1\beta_2, \alpha' \diamond \beta'_{-2}\beta'_{-1}\beta'_1\beta'_2) = \alpha\alpha' \diamond \beta_{-2}\beta_{-1}\beta_1\beta_2\beta'_1\beta'_2$$

if $\beta_{-1}\beta_2 = \beta'_{-2}\beta'_1$, and

$$\mathbb{X}_2(\alpha \diamond \beta_{-2}\beta_{-1}\beta_1\beta_2, \alpha' \diamond \beta'_{-2}\beta'_{-1}\beta'_1\beta'_2) = \varnothing$$

otherwise. The ordering matrix $\mathbb{X}_1$, indexed by

$$\{\gamma_1\gamma_2\gamma_3\gamma_4 : \gamma_i \in \{-, +\}, 1 \le i \le 4\},$$

is defined as

$$\mathbb{X}_1(\gamma_1\gamma_2\gamma_3\gamma_4, \gamma_1'\gamma_2'\gamma_3'\gamma_4') = \gamma_1\gamma_2\gamma_3\gamma_4\gamma_4'$$

if $\gamma_2\gamma_3\gamma_4 = \gamma_1'\gamma_2'\gamma_3'$, and

$$\mathbb{X}_1(\gamma_1\gamma_2\gamma_3\gamma_4, \gamma_1'\gamma_2'\gamma_3'\gamma_4') = \varnothing$$

otherwise. Furthermore, the transition matrix $\mathbf{T}_j$ of the solution space, indexed by

$$\{\alpha \diamond \beta_{-2}\beta_{-1}\beta_1\beta_2 : \alpha, \beta_i \in \{-, +\}, i = -2, -1, 1, 2\},$$

is a 0–1 matrix defined as

$$\mathbf{T}_j(\alpha \diamond \beta_{-2}\beta_{-1}\beta_1\beta_2, \alpha' \diamond \beta_{-2}'\beta_{-1}'\beta_1'\beta_2') = 1$$

if and only if

1. $\beta_{-1}\beta_2 = \beta_{-2}'\beta_1'$;
2. $\alpha\alpha' \diamond \beta_{-2}\beta_{-1}\beta_1\beta_2\beta_1'\beta_2' \in \mathcal{B}_j^{(2)}$;
3. $\beta_{-2}\beta_{-1}\beta_1\beta_2\beta_1', \beta_{-1}\beta_1\beta_2\beta_1'\beta_2' \in \mathcal{B}_j^{(1)}$.

Meanwhile, the transition matrix $T_{j;2} \in \mathcal{M}_{32\times 32}(\{0, 1\})$ of the second layer is given by

$$T_{j;2}(\alpha \diamond \beta_{-2}\beta_{-1}\beta_1\beta_2, \alpha' \diamond \beta_{-2}'\beta_{-1}'\beta_1'\beta_2') = 1$$

if and only if

1. $\beta_{-1}\beta_2 = \beta_{-2}'\beta_1'$;
2. $\alpha\alpha' \diamond \beta_{-2}\beta_{-1}\beta_1\beta_2\beta_1'\beta_2' \in \mathcal{B}_j^{(2)}$;

the transition matrix $T_{j;1} \in \mathcal{M}_{16\times 16}(\{0, 1\})$ of the first layer is given by

$$T_{j;1}(\gamma_1\gamma_2\gamma_3\gamma_4, \gamma_1'\gamma_2'\gamma_3'\gamma_4') = 1$$

if and only if $\mathbb{X}_1(\gamma_1\gamma_2\gamma_3\gamma_4, \gamma_1'\gamma_2'\gamma_3'\gamma_4') \in \mathcal{B}_j^{(1)}$. Denote $T_{j;1}^2 = (T_{p,q})_{p,q=1}^8$ as 64 smaller 2×2 pieces; let $\overline{T}_{j;1}$ be the rearrangement of $T_{j;1}^2$ so that the patterns present in $\overline{T}_{j;1}$ are the same as the bottom patterns presenting in the left-top 16×16 block of $T_{j;2}$. Then it can be seen that $\mathbf{T}_j = T_{j;2} \circ (E_2 \otimes \overline{T}_{j;1})$. Applying appropriate labeling on $\mathbf{T}_j$ derives the labeled graph presentation of $\mathbf{Y}_j$, $\mathbf{Y}_j^{(1)}$, and $\mathbf{Y}_j^{(2)}$, respectively. More precisely, the labeling $\mathcal{L}^{(1)}$, $\mathcal{L}^{(2)}$ for $\mathbf{Y}_j^{(1)}$, $\mathbf{Y}_j^{(2)}$ are given by

$$\mathcal{L}^{(1)}(\mathbf{T}_j(\alpha \diamond \beta_{-2}\beta_{-1}\beta_1\beta_2, \alpha' \diamond \beta_{-2}'\beta_{-1}'\beta_1'\beta_2')) = \beta_{-2}\beta_{-1}\beta_1\beta_2\beta_1'\beta_2'$$

and

$$\mathcal{L}^{(2)}(\mathbf{T}_j(\alpha \diamond \beta_{-2}\beta_{-1}\beta_1\beta_2, \alpha' \diamond \beta_{-2}'\beta_{-1}'\beta_1'\beta_2')) = \alpha\alpha'$$

respectively, provided

$$\mathbf{T}_j(\alpha \diamond \beta_{-2}\beta_{-1}\beta_1\beta_2, \alpha' \diamond \beta_{-2}'\beta_{-1}'\beta_1'\beta_2') = 1.$$

Let ρ_j be the spectral radius of $\mathbf{T}_j$, and let $\rho_j^{(1)}$, $\rho_j^{(2)}$ be the spectral radius of matrices obtained by applying SCM to the labeled graph presentation of $\mathbf{Y}_j^{(1)}$, $\mathbf{Y}_j^{(2)}$, respectively. Theorem 2.1 infers that

$$h(\mathbf{Y}_j^{(k)}) = \log \rho_j^{(k)}, \quad k = 0, 1, 2,$$

where $\mathbf{Y}_j^{(0)} = \mathbf{Y}_j$ and $\rho_j^{(0)} = \rho_j$. This completes the elucidation of Remark 3.5.

4 Conclusion

This paper investigates the spatial complexity of the mosaic solution space and mosaic output space of constant IHMNNs (16). The reason we concentrate on mosaic solutions is that the investigation of mosaic solutions is most essential in MNN models due to the learning algorithm and training processing. More abundant output patterns make the learning algorithm more efficient. A quantity that is frequently used for the number of output patterns is topological entropy. From the mathematical viewpoint, the denseness of the entropy set asserts that CIHMNNs are almost capable of exhibiting any phenomena requested. This makes CIHMNNs the universal machine in some sense and approves of CIHMNNs as efficient in learning algorithms. The activation function considered in this paper derives from CNNs. Since cellular neural networks have been applied to many areas such as image processing, the choice of the activation function seems to be convincingly adapted to our goal on the study of the vision systems of mammals. By applying the well-developed theory of symbolic dynamics, we demonstrate the explicit formula of the topological entropy of the solution, and the hidden and output spaces of IHMNNs.

We also emphasize that the method we have provided herein is more general, an easy extension leading us to consider the classical McCulloch-Pitts model and signum activation function. The related works are in preparation.

Acknowledgments The authors would like to express their gratitude to the anonymous referees. Their valuable comments have not only significantly improved the quality and readability of the paper, but also inspired some interesting further works. Ban is partially supported by the Ministry of Science and Technology, ROC (Contract No MOST 102-2628-M-259-001-MY3). Chang is grateful for the partial support of the Ministry of Science and Technology, ROC (Contract No MOST 103-2115-M-390-004-).

References

1. Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Netw 2:359–366
2. Widrow B, Lehr M (1990) 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. Proc IEEE 78:1415–1442
3. Peterson C, Söderberg B (1989) A new method for mapping optimization problems onto neural network. Int J Neural Syst 1:3–22
4. Hopfield JJ, Tank DW (1985) Neural computation of decisions in optimization problems. Biol Cybern 52:141–152
5. Alsultanny YA, Aqul MM (2003) Pattern recognition using multilayer neural-genetic algorithm. Neurocomputing 51:237–247
6. Widrow B (1962) Layered neural nets for pattern recognition. IEEE Trans Acoust Speech Signal Process 36:1109–1118
7. Fukushima K (2013) Artificial vision by multi-layered neural networks: neocognitron and its advances. Neural Netw 37:103–119
8. Fukushima K (2013) Training multi-layered neural network neocognitron. Neural Netw 40:18–31
9. Serre T, Kreiman G, Kouh M, Cadieu C, Knoblich U, Poggio T (2007) A quantitative theory of immediate visual recognition. Prog Brain Res Comput Neurosci 165:33–56
10. Bengio Y, LeCun Y (2007) Scaling learning algorithms towards AI. In: Bottou L, Chapelle O, DeCoste D, Weston J (eds) Large scale kernel machines. MIT Press, Cambridge
11. Fukushima K, Miyake S (1982) Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position. Pattern Recognit 15:455–469
12. Fukushima K (1988) Neocognitron: a hierarchical neural network capable of visual pattern recognition. Neural Netw 1:119–130
13. Fukushima K (2003) Neocognitron for handwritten digit recognition. Neurocomputing 51:161–180
14. Utgoff PE, Stracuzzi DJ (2002) Many-layered learning. Neural Comput 14:2497–2539

15. Hinton GE, Osindero S, Teh Y (2006) A fast learning algorithm for deep belief nets. Neural Comput 18:1527–1554
16. Freund Y, Haussler D (1994) Unsupervised learning of distributions on binary vectors using two layer networks. University of California, Santa Cruz, Technical Report UCSC-CRL-94-25
17. Ahmed A, Yu K, Xu W, Gong Y, Xing EP (2008) Training hierarchical feed-forward visual recognition models using transfer learning from pseudo tasks. In: Proceedings of the 10th European Conference on Computer Vision, pp 69–82
18. Bengio Y, Lamblin P, Popovici D, Larochelle H (2007) Greedy layer-wise training of deep networks. Adv Neural Inf Process Syst 19:153–160
19. Salakhutdinov R, Hinton GE (2008) Using deep belief nets to learn covariance kernels for Gaussian processes. Adv Neural Inf Process Syst 20:1249–1256
20. Salakhutdinov R, Hinton GE (2007) Learning a nonlinear embedding by preserving class neighbourhood structure. In: Proceedings of the Eleventth International Conference on Artificial Intelligence and Statistics. Omnipress, Madison
21. Hinton GE, Salakhutdinov R (2006) Reducing the dimensionality of data with neural networks. Science 313:504–507
22. Osindero S, Hinton GE (2008) Modeling image patches with a directed hierarchy of Markov random field. Adv Neural Inf Process Syst 20:1121–1128
23. Taylor G, Hinton G (2009) Factored conditional restricted Boltzmann machines for modeling motion style. In: Proceedings of the 26th International Conference on Machine Learning. Omnipress, Madison
24. Taylor G, Hinton GE, Roweis S (2007) Modeling human motion using binary latent variables. Adv Neural Inf Process Syst 19:1345–1352
25. Collobert R, Weston J (2008) A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the Twenty-fifth International Conference on Machine Learning. ACM, New York, pp 160–167
26. Mnih A, Hinton GE (2009) A scalable hierarchical distributed language model. Adv Neural Inf Process Syst 21:1081–1088
27. Bengio Y (2009) Learning deep architectures for AI. Found Trends Mach Learn 2:1–127
28. Killingback T, Loftus G, Sundaram B (2012) Competitively coupled maps and spatial pattern formation. arXiv:1204.2463
29. Yokozawa M, Kubota Y, Hara T (1999) Effects of competition mode on the spatial pattern dynamics of wave regeneration in subalpine tree stands. Ecol Model 118:73–86
30. Yokozama M, Kubota Y, Hara T (1998) Effects of competition mode on spatial pattern dynamics in plant communities. Ecol Model 106:1–16
31. Doebeli M, Killingback T (2003) Metapopulation dynamics with quasi-local competition. Theor Popul Biol 64:397–416
32. Doebeli M, Hauert C, Killingback T (2004) The evolutionary origin of cooperators and defectors. Science 306:859–862
33. Hauert C, Doebeli M (2004) Spatial structure often inhibits the evolution of cooperation in the snowdrift game. Nature 428:643–646
34. Chua LO, Yang L (1988) Cellular neural networks: theory. IEEE Trans Circuits Syst 35:1257–1272
35. Chua LO (1998) CNN: a paradigm for complexity. Singapore: world scientific series on nonlinear science, Series A,31. World Scietific
36. Chua LO, Roska T (2002) Cellular neural networks and visual computing. Cambridge University Press, Cambridge
37. Yang Z, Nishio Y, Ushida A (2002) Image processing of two-layer CNNs—applications and their stability. IEICE Trans Fundam E85–A:2052–2060
38. Yang T (2003) Multi-layer cellular neural networks: Theory and applications to modeling nitric oxide diffusion in nervous systems. Int J Comput Cogn 1:1–23
39. Peng J, Zhang D, Liao X (2009) A digital image encryption algorithm based on hyper-chaotic cellular neural network. Fundam Inform 90:269–282
40. Murugesh V (2010) Image processing applications via time-multiplexing cellular neural network simulator with numerical integration algorithms. Int J Comput Math 87:840–848
41. Juang J, Lin S-S (2000) Cellular neural networks: mosaic pattern and spatial chaos. SIAM J Appl Math 60:891–915
42. Ban J-C, Chang C-H, Lin S-S, Lin Y-H (2009) Spatial complexity in multi-layer cellular neural networks. J Differ Equ 246:552–580
43. Aein MJ, Talebi H (2009) Introducing a training methodology for cellular neural networks with application to mechanical vibration problem. In: Control Applications, (CCA) Intelligent Control, (ISIC), 2009 IEEE, July 2009, pp 1661–1666

44. Takahashi N, Chua LO (1998) On the complete stability of nonsymmetric cellular neural networks. IEEE Trans Circuits I 45:754–758
45. Li XM, Huang LH (2003) On the complete stability of cellular neural networks with external inputs and bias. Acta Math Appl Sin 26:475–486
46. Li X (2009) Analysis of complete stability for discrete-time cellular neural networks with piecewise linear output functions. Neural Comput 21:1434–1458
47. Ban J-C, Chang C-H (2013) The learning problem of multi-layer neural networks. Neural Netw 46: 116–123
48. Lind D, Marcus B (1995) An introduction to symbolic dynamics and coding. Cambridge University Press, Cambridge