Int. J. Mach. Learn. & Cyber. (2015) 6:567–579
DOI 10.1007/s13042-014-0244-2

ORIGINAL ARTICLE

Neural network equations and symbolic dynamics

Jung-Chao Ban

Received: 31 July 2013 / Accepted: 21 February 2014 / Published online: 19 March 2014

Abstract In this paper we provide an up-to-date survey on the study of the complexity of the mosaic solutions on neural network equations. Three types of equations, namely, cellular neural networks (CNNs), multi-layer CNN (MCNNs) and inhomogeneous CNNs (ICNNs) are discuss herein. Such topic strong related to the learning algorithm and training process on neural network equations. Each neural network produces different mosaic solution space, and each mosaic solution space induces an different symbolic dynamics. To understand the complexity (spatial entropy) of the mosaic solution space for a given neural network equation, we need to identify which the underlying symbolic space is, then using the established knowledge of symbolic dynamical systems to compute its spatial entropy. Recently there has been substantial progress in this field. This paper is a comprehensive survey of this field. It provides a summary of the interesting results in this field. It is our hope that the paper will provide a good overview of major results and techniques, and a friendly entry point for anyone who is interested in studying problems in this field.

Keywords Cellular neural networks · Multi-layer CNN · Inhomogeneous CNN · Separation property · Topological entropy

1 Introduction

The present survey paper is devoted to the recent progress made in the mathematical foundations of the complexity of mosaic solutions of CNNs. We focus on the recent results on CNNs, MCNNs and ICNNs (or MPCNNs).

In the past few decades, CNNs, introduced by Chua and Yang [25, 26] have been one of the most investigated paradigms for neural information and image processing [22]. The one-dimensional CNN is realized in the following form.

$$\frac{dx_i}{dt} = -x_i + \sum_{|k|\le d} a_k y_{i+k} + z, \quad i \in \mathbb{Z}, \tag{1}$$

where

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

is the *output function*; $\mathbf{A} = (\mathbf{a_{-d}}, \ldots, \mathbf{a_0}, \ldots, \mathbf{a_d})$ is called the *feedback template*, and z is the *threshold*. $\mathbb{T} = [\mathbf{A}, \mathbf{z}]$ is the *template* of (1). In a wide range of applications, the CNNs are required to be completely stable [27, 55, 61, 62, 65, 72], i.e., each trajectory should converge toward stationary states.

Learning algorithms (also called *Machine learning*), a branch of artificial intelligence (AI), is about the construction and study of systems that can learn from the given data or the parameter space. For example, a machine learning system could be trained on visual systems to learn to distinguish between dogs and cats. The learning algorithm in CNNs is most essential due to its powerful and substantial applications in image processing and pattern recognitions. Formally speaking, the purpose of the learning algorithm in neural networks is to study the relationship between the output solution and parameter spaces. More abundant output patterns make the learning algorithm more efficient. In *homogeneous* (template is space-invariant)1-d CNNs, the general method to identify all solutions on $\mathbb{Z}$ is to identify all *local patterns* [17, 47, 48, 50, 56] on local lattices. By using the space-invariant property one can build up the *global patterns* according to the local patterns.

Ban is partially supported by the National Science Council, ROC (Contract No. NSC 100-2115-M-259-009-MY2).

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

However, the well-known *separation property* [10, 47] reveals that " not all" sets of local patterns will appear as local patterns for some parameters. Thus, the global output solution of 1-d CNNs has some constraints. One important method to overcome this problem is the *multi-layer cellular neural networks* and the other one is *inhomogeneous cellular neural networks*.

A one-dimensional MCNN [24] is realized in the following form.

$$\frac{dx_i^{(n)}}{dt} = -x_i^{(n)} + \sum_{|k|\le d} a_k^{(n)} y_{i+k}^{(n)} + \sum_{|k|\le d} b_k^{(n)} u_{i+k}^{(n)} + z^{(n)}, \qquad (3)$$

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

$$u_i^{(n)} = y_i^{(n-1)} \text{for} 2 \le \mathrm{n} \le \mathrm{N}, \quad \mathrm{u_i^{(1)}} = \mathrm{u_i}, \quad \mathrm{x_i}(0) = \mathrm{x_i^0}, \qquad (4)$$

and the output function is defined in (2). Note that the linear coupling[1] $u_i^{(n)} = y_i^{(n-1)}$ for $2 \le n \le N$ in (4) makes such systems a coupled map lattice. MCNNs have received considerable attention and were successfully applied to many areas such as signal propagation between neurons, image processing [23, 25, 28, 29, 59, 73, 74], pattern recognition [6, 8], information technology [24, 60], CMOS realization [18, 35, 37] and VLSI implementation [38, 60, 70, 71]. MCNN is also the base of the deep neural networks, which is the kernel technology of the Microsoft's speech recognition system [30, 46, 58, 63, 64, 77, 78]. Another important reason for coupling CNNs is the simulation of the visual systems of mammals [40, 41], with each layer symbolizing a single cortex in the visual system.) In [68], the authors demonstrated a sufficient condition for the complete stability of MCNNs. Recently, Ban and Chang [10] showed that, for MCNNs, the more layers infers the more phenomena that the models are capable of. Thus, MCNNs indeed produce more abundant output spaces.

Another method to release the constraints of the separation property is the inhomogeneous CNNs [5, 7, 9]. A one-dimensional ICNN on $\mathbb{Z}$ is defined.

$$\frac{d}{dt}x_i(t) = -x_i(t) + z_i + \sum_{k\in\mathcal{N}_i} a_{k;i} f(x_k(t)), \qquad (5)$$

where $i \in \mathbb{Z}$, and $\mathcal{N}_i$, which is a finite subset of $\mathbb{Z}$, indicates the *neighborhood* for neuron x_i. As usual, the output function $f(x)$ is defined in (2). An ICNN is also called a *multiple cellular neural network* (MPCNN) since one may see later (Sect. 4) that the mosaic solution space of such structure of interaction will conjugate to some multiple shift space. Traditionally, the template for CNN is homogeneous, i.e., the template is space-invariant. However, there are more and more neural networks that use inhomogeneous templates to describe some of the problems that arise from the biological and ecological contexts [33, 34, 45, 54, 75, 76]. Some new and interesting phenomena of pattern formation and spatial chaos were also found in ICNNs.

Spatial complexity for a given system is a concept that unveils how complex the system is. Such an idea was first defined by Chow et al. on the mosaic solutions of an array of scalar non-linear dynamical systems [19–21]. The *spatial entropy* (see [57] for the formal definition in symbolic dynamical systems, and also see [69] for the general definition in ergodic theory) is used to measure the spatial complexity of such systems. Therefore, spatial entropy is commonly used to measure the complexity of the output patterns for neural network equations. To be precise, let $X \subset \{0,1\}^{\mathbb{N}}$ be a shift space and σ be its shift map, i.e., $\sigma(x)_i = x_{i+1}$ for $x \in X$ and $i \in \mathbb{Z}$. Denote by $\Gamma_k(X)$ the cardinality of the collection of words of length k. The *spatial entropy* is then defined by the growth rate of $\Gamma_k(X)$ with respect to k. That is,

$$h(X) = \lim_{k\to\infty} \frac{\log \Gamma_k(X)}{k}, \qquad (6)$$

whenever the limit (6) exists. Since the output function of CNNs are piecewise linear [defined later in (2)] and the mosaic solutions (see Definition 2.1) usually conjugate to some shift space. Equation (6) is often used to measure the complexity of mosaic solutions. Roughly speaking, positive entropy of a neural network indicates that the number of mosaic solutions grows exponentially with respect to the length of the lattice. In this case, we call such a system *spatially chaotic*, i.e., $h_{top}(X) > 0$. Otherwise, we call such systems *pattern formation*. There is plenty of literature that has studied these pattern formation systems [31, 32, 43, 44, 66, 67]. However, systematic studies of spatially chaotic systems are few, especially in MCNNs and ICNNs. The main difficulty appears in identifying what the underlying symbolic space of the output solution is. Once the underlying symbolic space is identified, well-established knowledge of such symbolic space will be used to compute the topological entropy or other statistic invariants.

Juang and Lin [50] proved that, for 1-d homogeneous CNN, the underlying symbolic space of the mosaic solutions forms a 1-*d* subshift of finite type (SFT for short, [57]). Ban et al. [13, 14] proved that, for MCNNs, its underlying symbolic space forms a *sofic space*, which is a *factor* of SFT. In a sense, one sees that the MCNNs indeed produce more complicated dynamics than single layer CNNs.

More recently, Ban and Chang [12] defined the so-called *initial value problem (IVP) of a MCNN* as follows. Given $\mathbf{a} \in \mathbb{R}^{\mathbf{n}}$ and $\ell \in \mathbb{N}$, the IVP of a MCNN is investigating those solutions that satisfy 3 with initial condition $\mathbf{a}$ at coordinate ℓ. More precisely, the IVP of a MCNN focuses on investigating the space

$$\mathbf{X_a} = \{\mathbf{x} = (\mathbf{x_i^{(n)}})_{\mathbf{i}\in\mathbb{N},\mathbf{1}\le\mathbf{n}\le\mathbf{N}} \in \mathbb{R}^{\infty\times\mathbf{N}} \mathbf{:}\ \mathbf{x}\ \text{satisfies}(3), \times (x_\ell^{(n)})_{n=1}^N = \mathbf{a}\}. \qquad (7)$$

[1] Such coupling is suggested by Chua and Roska. The purpose is to design MCNNs for solving some image processing and pattern recognition problems [25].

The recent study has shown that the MCNN with initial values produces a new underlying symbolic space, called the *Path set* [1], with such space being used to solve some problems in number theory, i.e., *p -adic path set fractal* [2].

As expected, ICNNs produce a new symbolic space. Ban and Chang [11] proved that such systems produce new underlying symbolic spaces called *multiple shift spaces*. Such symbolic spaces are defined by Furstenberg [42], Fan and Liao [39] in their studies of arithmetic progression and Szemerédi's theorem [36].

On the one hand, one may see that the MCNN and ICNN produce abundant symbolic spaces for mathematical study, with the mathematical foundation of such symbolic space helping to compute the spatial complexity of such neural systems. On the other hand, various neural network equations present themselves as natural examples for new symbolic dynamics. Thus, this study will set up a link between symbolic dynamics and neural network equations.

We organize the materials as follows. In Sect. 2 we review the results of classical CNNs, with the methods to compute the topological entropy presented therein. Section 3 review the results for MCNNs, and MCNNs with initial values. The solutions structure and the method to compute their topological entropy are also presented. Finally, the results for ICNN are provided in Sect. 4 and the conclusion is given in Sect. 5.

2 Classical cellular neural networks

This chapter introduces the classical one-dimensional CNN without input. The methodology for computing entropy will also be introduced. First we introduce the concept of *mosaic patterns*, which is crucial in our study.

Definition 2.1 (*Mosaic patterns* [50]) A *mosaic solution* $\overline{x} = (\overline{x}_i)_{i\in\mathbb{Z}}$ of (1) is a stationary solution satisfying $|\overline{x}_i| \geq 1$ for all $i \in \mathbb{Z}$. The output of mosaic solutions are called *mosaic patterns*.

Since the output function in (2) is piecewise linear with $f(x) = 1$ (resp. -1) if $|x| \geq 1$ (resp. $|x| \leq 1$), if $\overline{x} = (\overline{x}_i)_{i\in\mathbb{Z}}$ is a mosaic solution, then its output must be an element in $\{-1, +1\}^{\mathbb{Z}}$, which is why we call it "*patterns*". Suppose y is a mosaic pattern, the necessary and sufficient conditions for $y_i = 1$, is

$$a - 1 + z > -\left(\sum_{0<|k|\leq d} a_k y_{i+k}\right), \tag{8}$$

where $a = a_0$. Similarly, the necessary and sufficient conditions for $y_i = -1$, is

$$a - 1 - z > \sum_{0<|k|\leq d} a_k \bar{y}_{i+k}. \tag{9}$$

Substitute -1 and 1 by $-$ and $+$, respectively.

Let $\alpha = (a_{-d}, \cdots, a_{-1}, a_1, \cdots, a_d)$. The *basic set* of admissible local patterns with " + " state in the center is defined as

$$\mathcal{B}(+) = \{v \in \{-1, 1\}^{2d} : a - 1 + z > -(\alpha \cdot v)\},$$

where $\cdot$ is the inner product in Euclidean space. Similarly, the basic set of admissible local patterns with " − " state in the center is defined as

$$\mathcal{B}(-) = \{v \in \{-1, 1\}^{2d} : a - 1 - z > \alpha \cdot v\}.$$

Furthermore, the admissible local patterns induced by $\mathbb{T}$ can be denoted by

$$\mathcal{B} = (\mathcal{B}(+), \mathcal{B}(-)).$$

For clarity, we consider the CNNs with *nearest neighborhood*, i.e., $d = 1$. In this case, the *ordering matrix* of (1) can be defined as follows.

$$\mathbb{X} = \begin{array}{c} \\ \boxed{--} \\ \boxed{-+} \\ \boxed{+-} \\ \boxed{++} \end{array} \begin{array}{c} \begin{array}{cccc} \boxed{--} & \boxed{-+} & \boxed{+-} & \boxed{++} \end{array} \\ \begin{pmatrix} \boxed{---} & \boxed{--+} & \varnothing & \varnothing \\ \varnothing & \varnothing & \boxed{-+-} & \boxed{-++} \\ \boxed{+--} & \boxed{+-+} & \varnothing & \varnothing \\ \varnothing & \varnothing & \boxed{++-} & \boxed{+++} \end{pmatrix} \end{array} = (x_{ij})_{1\leq i,j\leq 4}. \tag{10}$$

The idea of the ordering matrix is to " store" the basic set in $\mathbb{X}$, then using $\mathbb{X}$ to generate all possible global patterns (since template $\mathbb{T}$ is space-invariant). The transition matrix according to the ordering matrix in (1) is defined by

$$T(i,j) = \begin{cases} 1 & \text{if } x_{ij} \in \mathcal{B}; \\ 0 & \text{otherwise.} \end{cases}$$

Denote by $\mathbb{Z}_n$ the lattice in $\mathbb{Z}$ with length $2n+1$ and $\mathbf{Y}_n$ (resp.$\mathbf{Y}$) the mosaic solution of (1) in $\mathbb{Z}_n$ (resp. $\mathbb{Z}$). Let $\mathbf{M_{m\times m}}(\mathbb{R})$ be the collection of $m \times m$ matrices with real entries, suppose $A \in \mathbf{M_{m\times m}}(\mathbb{R})$ we define $\Sigma(A)$ and $\Sigma_n(A)$ for $n \in \mathbb{N}$ as follows.

$$\begin{aligned} \Sigma(A) &= \left\{(i_0,\ldots) \in \{0,\ldots,m\}^{\mathbb{N}} : A(i_k, i_{k+1}) = 1 \text{ for k} \in \mathbb{N}\right\}, \\ \Sigma_n(A) &= \{(i_0,\ldots) \in \{0,\ldots,m\}^{n} : A(i_k, i_{k+1}) = 1 \text{ for } 0 \le \text{k} \le \text{n}\}. \end{aligned} \tag{11}$$

Sometimes we call $\Sigma_n(A)$ the *n -cylinder set* of $\Sigma(A)$. Joung and Lin [50] characterized the underlying space of $\mathbf{Y}$ and presents its entropy formula.

Theorem 2.2 ([50]). Given $\mathbb{T} = [\mathbf{A}, \mathbf{z}]$. Suppose the basic set $\mathcal{B}$ is constructed according to $\mathbb{T}$ and the transition matrix T is also constructed with respect to $\mathcal{B}$ and ordering matrix (10). Then the following hold.

(1) The mosaic solution space $\mathbf{Y} = \mathbf{Y}(\mathbb{T})$ is conjugate to the subshift of finite type (also called Markov system) induced by T. That is $\mathbf{Y} \simeq \Sigma(\mathbf{T})$. The solution $\mathbf{Y_n} = \mathbf{Y_n}(\mathbb{T})$ on $\mathbb{Z}_n$ is equivalent to the n -cylinder set of $\Sigma(T)$. That is $\mathbf{Y_n} = \Sigma_\mathbf{n}(\mathbf{T})$.

(2) The topological entropy of $\mathbf{Y}$ is the logarithm of the maximal eigenvalue of A, i.e.,

$$h_{top}(\mathbf{Y}) = \log \rho(\mathbf{T}), \tag{12}$$

where $\rho(B)$ stands for the maximal eigenvalue of B for B.

We give the idea for the proof of Theorem 2.2, since Eq. (1) is space-invariant, the constraint from (8) and (9) make that every solution of (1) in $\mathbb{Z}$ satisfies the local rule of $\mathcal{B}(+)$ and $\mathcal{B}(-)$. Thus, the solution space is equivalent to the shift space (11). Once the underlying symbolic space is identified as (11), $\mathbb{X}$ and T are systematic tools to generate " all" possible global patterns and the formula (12) is a well-known result on computing (6) (Theorem 4.3.1 of [57]). First, we emphasis here that once the local pattern $\mathcal{B}$ according to $\mathbb{T}$ is constructed, (1) of Theorem 2.2 is derived by using the spatially invariant (also called *isotropic*) property of the template $\mathbb{T}$ for (1). However, if $\mathbb{T}$ is spatially variant, then (1) becomes inhomogeneous, i.e., ICNN, and Theorem 2.2 is no longer true. In this case, $\mathbf{Y}$ produces an inhomogeneous symbolic dynamical system, which makes the entropy formula difficult to compute. Second, for multidimensional CNN, i.e., (1) is defined in $\mathbb{Z}^d$ with $d \ge 2$, the computation of entropy is extremely difficult, only few result is known. Ban et al. [15, 16] establish the higher dimensional transition matrices to overcome this difficulty, some lower and upper bounds for entropy can be derived by using the skill therein. In the following section we will introduce the MCNNs, as such a system will produce more general underlying symbolic dynamical systems. Namely, the sofic shift.

3 Multi-layer cellular neural networks

In this section the one-dimensional MCNN is study. We first introduce how to produce a set of local patterns for *cover space* $\mathbf{Y}$, and factor space $\mathbf{Y^{(n)}}$, then we compute its topological entropy by proving that $\mathbf{Y^{(n)}}$ is conjugate to a sofic space. The collection of parameters is as follows. As usual, $\mathbf{A} = [\mathbf{A^{(n)}}]_{\mathbf{1}\le\mathbf{n}\le\mathbf{N}}$ with $\mathbf{A^{(n)}} = (\mathbf{a}_{-\mathbf{d}}^{(\mathbf{n})}, \ldots, \mathbf{a}_{\mathbf{d}}^{(\mathbf{n})})$ is the *feedback template*, $\mathbf{B} = [\mathbf{B^{(n)}}]_{\mathbf{1}\le\mathbf{n}\le\mathbf{N}}$ with $\mathbf{B^{(n)}} = (\mathbf{b}_{-\mathbf{d}}^{(\mathbf{n})}, \ldots, \mathbf{b}_{\mathbf{d}}^{(\mathbf{n})})$ is the *controlling template*, and $\mathbf{z} = [\mathbf{z^{(n)}}]_{\mathbf{1}\le\mathbf{n}\le\mathbf{N}}$ is the *threshold*. $\mathbb{T} = [\mathbf{A}, \mathbf{B}, \mathbf{z}]$ is the *template* of (3). As in Definition 2.1, we define the mosaic patterns for MCNN as follows.

Definition 3.1 (*Mosaic patterns for MCNN* [14]). A *mosaic solution* $\overline{x} = (\overline{x}_i^{(n)})_{i\in\mathbb{Z}}$ of (3) is a stationary solution satisfying $\left|\overline{x}_i^{(n)}\right| \ge 1$ for all $i \in \mathbb{Z}$ and $1 \le n \le N$. The output of mosaic solutions is called *mosaic patterns*.

For clarity, we consider the fundamental part of a MCNN: the one-layer cellular neural networks with inputs.

$$\frac{dx_i}{dt} = -x_i + \sum_{|k|\le d} a_k y_{i+k} + \sum_{|\ell|\le d} b_\ell u_{i+\ell} + z,$$

where $A = [-a_d, \cdots, a, \cdots, a_d]$, $B = [-b_d, \cdots, b, \cdots, b_d]$ are the feedback and controlling templates, respectively, and z is the threshold.

Consider the mosaic solution $\mathbf{x}$, the necessary and sufficient conditions for state " $+$" at cell C_i , i.e., $y_i = 1$, is

$$a - 1 + z > -\left(\sum_{0<|k|\le d} a_k y_{i+k} + \sum_{|\ell|\le d} b_\ell u_{i+\ell}\right), \tag{13}$$

where $a = a_0$. Similarly, the necessary and sufficient conditions for state " $-$" at cell C_i, i.e., $y_i = -1$, is

$$a - 1 - z > \sum_{0<|k|\le d} a_k \bar{y}_{i+k} + \sum_{|\ell|\le d} b_\ell u_{i+\ell}. \tag{14}$$

Rewrite the output patterns $y_{-d}\cdots y\cdots y_d$ coupled with input $u_{-d}\cdots u\cdots u_d$ as

$$\boxed{\begin{matrix} y_{-d}\cdots y_{-1}y_0y_1\cdots y_d \\ u_{-d}\cdots u_{-1}u_0u_1\cdots u_d \end{matrix}} \equiv y_{-d}\cdots y_d \diamond u_{-d}\cdots u_d \in \{-,+\}^{\mathbb{Z}_{(2d+1)\times 2}}$$

Let

$$V^n = \{v \in \mathbb{R}^n : v = (v_1, v_2, \cdots, v_n), |\mathrm{v_i}| = 1 \text{ for } 1 \le \mathrm{i} \le \mathrm{n}\},$$

where $n = 4d + 1$, (13) and (14) can be rewritten in a compact form by introducing the following notation.

Denote $\alpha = (a_{-d}, \cdots, a_{-1}, a_1, \cdots, a_d)$, $\beta = (b_{-d}, \cdots, b, \cdots, b_d)$. Then, α can be used to represent A', the surrounding template of A without center, and β can be used to represent the template B. The basic set of admissible local patterns with " $+$" state in the center is defined as

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

where $\cdot$ is the inner product in Euclidean space. Similarly, the basic set of admissible local patterns with " $-$ " state in the center is defined as

$$\mathcal{B}(-, A, B, z) = \{v \diamond w \in V^n : a - 1 - z > \alpha \cdot v + \beta \cdot w\}.$$

Furthermore, the admissible local patterns induced by (A, B, z) can be denoted by

$$\mathcal{B}(A, B, z) = (\mathcal{B}(+, A, B, z), \mathcal{B}(-, A, B, z)).$$

For $1 \le n \le N$, let $\mathcal{B}^{(n)} = (\mathcal{B}^{(n)}(+), \mathcal{B}^{(n)}(-))$ be the basic set of admissible local patterns of the nth layer.

Definition 3.2 (*Mosaic solution space* $\mathbf{Y}$ *and output solution space* $\mathbf{Y^{(N)}}$ [14]). The *mosaic solution space* $\mathbf{Y}$ of (3), as defined by

$$\mathbf{Y} = \left\{ \begin{matrix} \cdots \mathbf{y_{-1}^{(N)}} \mathbf{y_0^{(N)}} \mathbf{y_1^{(N)}} \cdots \\ \vdots \\ \cdots \mathbf{y_{-1}^{(2)}} \mathbf{y_0^{(2)}} \mathbf{y_1^{(2)}} \cdots \\ \cdots \mathbf{y_{-1}^{(1)}} \mathbf{y_0^{(1)}} \mathbf{y_1^{(1)}} \cdots \end{matrix} \right\} \subseteq \{-\mathbf{1}, \mathbf{1}\}^{\mathbb{Z}_{\infty\times \mathbf{N}}},$$

is generated by the basic set of admissible local patterns

$$\begin{aligned} \mathcal{B} &= (\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(N)}) \\ &= \left\{ \begin{matrix} y_{-d}^{(N)} \cdots y_{-1}^{(N)} y_0^{(N)} y_1^{(N)} \cdots y_d^{(N)} \\ \vdots \\ y_{-d}^{(2)} \cdots y_{-1}^{(2)} y_0^{(2)} y_1^{(2)} \cdots y_d^{(2)} \\ y_{-d}^{(1)} \cdots y_{-1}^{(1)} y_0^{(1)} y_1^{(1)} \cdots y_d^{(1)} \end{matrix} \right\} \subseteq \{-1, 1\}^{\mathbb{Z}_{(2d+1)\times N}}. \end{aligned} \tag{15}$$

We call $\mathbf{Y^{(N)}}$ the *output space* of (3). That is,

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

For $n \ne N$, $Y^{(n)}$ is called the *hidden space*.

The reason is named $\mathbf{Y^{(N)}}$ as the output space is due to the fact that only $Y^{(N)}$ will " show up" in chip devices. However, one easily sees that $Y^{(N)}$ is a part of $\mathbf{Y}$ and is influenced by $Y^{(n)}$ for $n \ne N$, thus the mosaic solution at another array is called *hidden*.

For simplicity we consider MCNNs with the nearest neighborhood, that is, $d = 1$ in the next subsection.

3.1 Two-layer cellular neural networks

A two-layer MCNN with nearest neighborhood is realized as follows.

$$\begin{cases} \dfrac{dx_i^{(1)}}{dt} = -x_i^{(1)} + \displaystyle\sum_{|k|\le 1} a_k^{(1)} y_{i+k}^{(1)} + z^{(1)}, \\ \dfrac{dx_i^{(2)}}{dt} = -x_i^{(2)} + \displaystyle\sum_{|k|\le 1} a_k^{(2)} y_{i+k}^{(2)} + \displaystyle\sum_{|k|\le 1} b_k^{(2)} u_{i+k}^{(2)} + z^{(2)}, \end{cases} \tag{16}$$

where $u_i^{(2)} = y_i^{(1)}$ for $i \in \mathbb{Z}$. The ordering matrix $\mathbb{X}_2$ of (16) is a 16×16 symbolic matrix defined in Fig. 1.

Let

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

Define

$$a_{i_1 i_2} a_{i_2' i_3} = \varnothing \quad \Leftrightarrow \quad i_2 \ne i_2'. \tag{17}$$

If $a_{i_1 i_2} a_{i_2' i_3} \ne \varnothing$, then it is a pattern with size 3×1 and denoted by $a_{i_1 i_2 i_3}$. Write

$$\mathbb{X}_2 = \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} \\ X_{21} & X_{22} & X_{23} & X_{24} \\ X_{31} & X_{32} & X_{33} & X_{34} \\ X_{41} & X_{42} & X_{43} & X_{44} \end{pmatrix} \text{ and}$$

$$X_{ij} = \begin{pmatrix} x_{ij;11} & x_{ij;12} & x_{ij;13} & x_{ij;14} \\ x_{ij;21} & x_{ij;22} & x_{ij;23} & x_{ij;24} \\ x_{ij;31} & x_{ij;32} & x_{ij;33} & x_{ij;34} \\ x_{ij;41} & x_{ij;42} & x_{ij;43} & x_{ij;44} \end{pmatrix},$$

for $1 \le i, j \le 4$ as Fig. 1. $x_{ij;kl}$ means the pattern $\begin{matrix} a_{r_1 r_2} a_{r_2' r_3} \\ a_{s_1 s_2} a_{s_2' s_3} \end{matrix}$, where

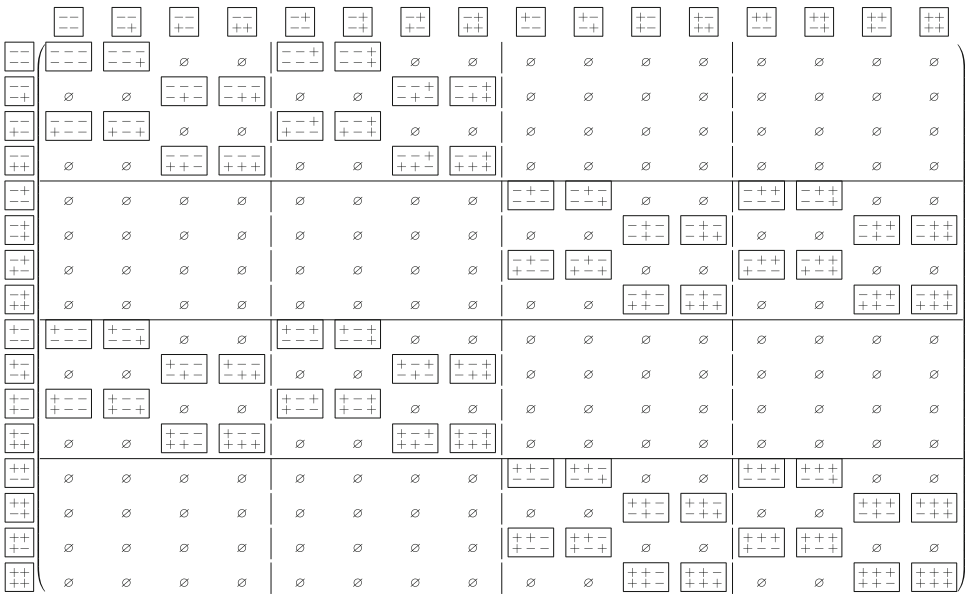

Fig. 1 The enlarged ordering matrix of two-layer neural networks

$$
\begin{aligned}
r_1 &= \left\lfloor \frac{i-1}{2} \right\rfloor, \quad r_2 = i-1-2r_1, \quad r_2' = \left\lfloor \frac{j-1}{2} \right\rfloor, \\
r_3 &= j-1-2r_2', \\
s_1 &= \left\lfloor \frac{k-1}{2} \right\rfloor, \quad s_2 = k-1-2s_1, \quad s_2' = \left\lfloor \frac{l-1}{2} \right\rfloor, \\
s_3 &= l-1-2s_2'.
\end{aligned} \tag{18}
$$

If $a_{r_1 r_2} a_{r_2' r_3} = \varnothing$ or $a_{s_1 s_2} a_{s_2' s_3} = \varnothing$, then $x_{ij;kl} = \varnothing$. Furthermore, if $x_{ij;kl} \neq \varnothing$, then it is denoted by the pattern $\begin{matrix} a_{r_1} a_{r_2} a_{r_3} \\ a_{s_1} a_{s_2} a_{s_3} \end{matrix}$ in $\{+,-\}^{\mathbb{Z}_{3\times 2}}$.

Roughly speaking, the above discussion unveils that in each small block of the ordering matrix $\mathbb{X}_2$, the " top" patterns are the same. For example, in X_{43} the top 3×1, patterns are the same as $++-$, and for X_{12} is $--+$.

Suppose that $\mathcal{B}$ is given. The transition matrix $T \equiv T(\mathcal{B}) \in \mathcal{M}_{16}(\mathbb{R})$ is a 16×16 matrix defined by

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

Herein we express the ordering matrix $\mathbb{X}_2 = (x_{pq})_{1 \le p,q \le 16}$ to ease the notation. We have

$$
T = T_2 \circ (E_4 \otimes T_1), \tag{19}
$$

where E_k is a $k \times k$ matrix with all entries being 1's, and T_1 and T_2 are the transition matrices of the first and second layer, respectively.

Let $\mathbf{Y} \subseteq \{-,+\}^{\mathbb{Z}_{\infty\times 2}}$ be the solution space of (16). That is,

$$
\mathbf{Y} = \left\{ \begin{pmatrix} y_i \\ u_i \end{pmatrix}_{i\in\mathbb{Z}} : \boxed{\begin{matrix} y_i y_{i+1} y_{i+2} \\ u_i u_{i+1} u_{i+2} \end{matrix}} \in \mathcal{B} \text{ for } i \in \mathbb{Z} \right\}
$$

where $\mathcal{B}$ is the set of admissible local patterns. It comes immediately from Theorem 2.2 that $\mathbf{Y}$ is a shift of finite type.

To ease the notation, denote $\boxed{\begin{matrix} y_1 y_2 y_3 \\ u_1 u_2 u_3 \end{matrix}}$ by $y_1 y_2 y_3 \diamond u_1 u_2 u_3$ and

$$
\mathbf{y} \diamond \mathbf{u} \equiv \boxed{\begin{matrix} \mathbf{y} \\ \mathbf{u} \end{matrix}} = \begin{matrix} \cdots y_{-2} y_{-1} y_0 y_1 y_2 \cdots \\ \cdots u_{-2} u_{-1} u_0 u_1 u_2 \cdots \end{matrix}, \tag{20}
$$

where $\mathbf{y} = (\mathbf{y_i})_{\mathbf{i}\in\mathbb{Z}}, \mathbf{u} = (\mathbf{u_i})_{\mathbf{i}\in\mathbb{Z}}$. Then, we can write $\mathbf{Y}$ as

$$\mathbf{Y} = \{\mathbf{y} \diamond \mathbf{u} : \mathbf{y_i y_{i+1} y_{i+2}} \diamond \mathbf{u_i u_{i+1} u_{i+2}} \in \mathcal{B} \textbf{ for } \mathbf{i} \in \mathbb{Z}\}$$

One can see that the ordering matrix in Fig. 1 is a systematic way to store the the patterns of size 3×2, and note that T^n (the multiplication of T) generates global patterns of size $(n+2) \times 2$. However, we are only interested in the top layer pattern, Theorem 2.2 is not applicable in this occasion. Thus we need some new idea to overcome this problem. Namely, the labeled graph approach. Define $\phi^{(1)}, \phi^{(2)} : \mathbf{Y} \to \{-,+\}^{\mathbb{Z}}$ by

$$\phi^{(1)}(\mathbf{y} \diamond \mathbf{u}) = \mathbf{u} \qquad \phi^{(\mathbf{2})}(\mathbf{y} \diamond \mathbf{u}) = \mathbf{y}$$

Set $Y^{(\ell)} = \phi^{(\ell)}(\mathbf{Y})$ for $\ell = 1, 2$. If $\mathbf{y} \diamond \mathbf{u} \in \mathbf{Y}$ is a solution of (16), then only the top pattern $\mathbf{y}$ can be observed. The bottom pattern $\mathbf{u}$ is hidden in this system. Recall that the dynamical behavior of the output space $Y^{(2)}$ is influenced by the hidden space $Y^{(1)}$. Unlike $\mathbf{Y}$, the symbolic dynamics of $Y^{(i)}$ $i = 1, 2$ is not a subshift of finite type. Actually, they are sofic instead.

Definition 3.3 (*Labeled graph and sofic shift*). A *labeled graph* $\mathcal{G} = (G, \mathcal{L})$ consists of an underlying graph G with edge set $\mathcal{E}$, and the labeling $\mathcal{L} : \mathcal{E} \to \mathcal{A}$ assigned to each edge a label from the finite alphabet $\mathcal{A}$. A *sofic shift* is defined by $\mathbf{X} = \mathbf{X}_{\mathcal{G}}$ for labeled graph $\mathcal{G}$.

Definition 3.4 (*Right-resolving*). A labeled graph $\mathcal{G} = (G, \mathcal{L})$ is *right-resolving* if, for some vertex I of G, the edges starting from I carrying different labels.

Let $\mathcal{A} = \{\alpha_0, \ldots, \alpha_8\}$, where

$$\alpha_0 = ---, \ldots, \alpha_8 = +++.$$

That is, we arrange the patterns of length 3 in a lexicographic order. For $i = 1, 2$, define $\mathcal{L}^{(i)} : \{+,-\}^{\mathbb{Z}_{3\times 2}} \to \{+,-\}^{\mathbb{Z}_{3\times 1}}$ by

$$\mathcal{L}^{(1)}(y_0y_1y_2 \diamond u_0u_1u_2) = u_0u_1u_2, \mathcal{L}^{(2)}(\mathrm{y_0y_1y_2} \diamond \mathrm{u_0u_1u_2}) = y_0y_1y_2.$$

The following theorem demonstrates that the solution space $Y^{(i)}$ for $i = 1, 2$ is conjugate to a sofic shift. The knowledge of computing entropy of sofic shifts can be used to compute their topological entropy. Given a transition matrix $T \in \mathbb{R}^{n\times n}$, denoted by G_T, the *induced graph* of T which consists of the vertex set $\mathcal{V} = \{1, \ldots, n\}$ and edge set $\mathcal{E} = \{(i,j) : 1 \le i,j \le n\}$ with $(i,j) \in \mathcal{E}$ if and only if $T(i,j) = 1$ for some $1 \le i,j \le n$.

Theorem 3.5 (Ban et al. [14]). Let $\mathbb{T} = [\mathbf{A}, \mathbf{B}, \mathbf{z}]$ be given for (16), the basic set $\mathcal{B}$ is constructed in (15) according to $\mathbb{T}$ and T is the transition matrix according to ordering matrix $\mathbb{X}_2$. Then

(1) $\mathbf{Y}$ is conjugate to a subshift of finite type. Namely, $\mathbf{Y} \simeq \Sigma(\mathbf{T})$ (defined in (11)).

(2) For $i = 1$ or 2, $Y^{(i)}$ is conjugate to the labeled graph $\mathcal{G}^{(i)} = (G_T, \mathcal{L}^{(i)})$. That is

$$Y^{(i)} \simeq \mathcal{G}^{(i)} = (G_T, \mathcal{L}^{(i)}).$$

(3) If $\mathcal{G}^{(i)} = (\Sigma(T), \mathcal{L}^{(i)})$ is right resolving for $i = 1$ or 2, then

$$h_{top}(Y^{(i)}) = \log \rho(T).$$

(4) If $\mathcal{G}^{(i)} = (\Sigma(T), \mathcal{L}^{(i)})$ is not right resolving for some $i = 1$ or 2. Then there exists a scheme (called *subset construction*) transforming $\mathcal{G}^{(i)}$ to be a right resolving labeled graph. That is, there exists a new $\widehat{T}$ and $\widehat{\mathcal{L}}^{(i)}$ such that $\widehat{\mathcal{G}}^{(i)} = (\Sigma(\widehat{T}), \widehat{\mathcal{L}}^{(i)})$ is a right resolving labeled graph and

$$h_{top}(Y^{(i)}) = \log \rho(\widehat{T}).$$

Note here that Theorem 3.5 provides a complete result for computing the topological entropy in a 2-layer MCNN. (1) of Theorem 3.5 represents that the *covering space* is still a subshift of finite type, i.e., the Markov system. (3) of Theorem 3.5 represents that the *projection space* $Y^{(i)}$ $i = 1, 2$ (hidden and output solution spaces) produce new underlying symbolic space, namely, the sofic space. It is well-known that the subshift of finite type is a sofic space; however, there are many strict sofic spaces, i.e., it is sofic but not a subshift of finite type. From a mathematical point of view, Theorem 3.5 reveals that the MCNNs are indeed a generalization of the one-layer CNN.

3.2 Recursive formula for MCNNs

From the previous section, one can divide the scheme of computing topological entropy into three parts: (I) Identify the local patterns (basic sets according to $\mathbb{T}$); (II) Construct the *appropriate* ordering matrix to store the local patterns, and (III) Identify the underlying symbolic dynamical system according to the ordering matrix and then applying the knowledge from the SD to compute the entropy.

For CNNs and MCNNs, step (I) is robust; once step (II) is constructed, step (III) is also robust. Thus step (II) is crucial. However, a systematic scheme for constructing the ordering matrices is found in general layers MCNNs.

The ordering matrix $\mathbb{X}_N$ of all possible local patterns in $\{+,-\}^{\mathbb{Z}_{3\times N}}$ is defined recursively as

$$\mathbb{X}_N = \begin{pmatrix} \mathbf{X_{11}} & \mathbf{X_{12}} & \varnothing & \varnothing \\ \varnothing & \varnothing & \mathbf{X_{23}} & \mathbf{X_{24}} \\ \mathbf{X_{31}} & \mathbf{X_{32}} & \varnothing & \varnothing \\ \varnothing & \varnothing & \mathbf{X_{43}} & \mathbf{X_{44}} \end{pmatrix},$$

where

$$\mathbf{X_{i_1j_1}} = \begin{pmatrix} \mathbf{X_{i_1j_1;11}} & \mathbf{X_{i_1j_1;12}} & \varnothing & \varnothing \\ \varnothing & \varnothing & \mathbf{X_{i_1j_1;23}} & \mathbf{X_{i_1j_1;24}} \\ \mathbf{X_{i_1j_1;31}} & \mathbf{X_{i_1j_1;32}} & \varnothing & \varnothing \\ \varnothing & \varnothing & \mathbf{X_{i_1j_1;43}} & \mathbf{X_{i_1j_1;44}} \end{pmatrix},$$

$$X_{i_1j_1;i_2j_2;\cdots;i_kj_k} = \begin{pmatrix} X_{i_1j_1;i_2j_2;\cdots;i_kj_k;11} & X_{i_1j_1;i_2j_2;\cdots;i_kj_k;12} & \varnothing & \varnothing \\ \varnothing & \varnothing & X_{i_1j_1;i_2j_2;\cdots;i_kj_k;23} & X_{i_1j_1;i_2j_2;\cdots;i_kj_k;24} \\ X_{i_1j_1;i_2j_2;\cdots;i_kj_k;31} & X_{i_1j_1;i_2j_2;\cdots;i_kj_k;32} & \varnothing & \varnothing \\ \varnothing & \varnothing & X_{i_1j_1;i_2j_2;\cdots;i_kj_k;43} & X_{i_1j_1;i_2j_2;\cdots;i_kj_k;44} \end{pmatrix}, \tag{21}$$

for $1 \le k \le N-2$, and

$$X_{i_1j_1;i_2j_2;\cdots;i_{N-1}j_{N-1}} = \begin{pmatrix} x_{i_1j_1;\cdots;i_{N-1}j_{N-1};11} & x_{i_1j_1;\cdots;i_{N-1}j_{N-1};12} & \varnothing & \varnothing \\ \varnothing & \varnothing & x_{i_1j_1;\cdots;i_{N-1}j_{N-1};23} & x_{i_1j_1;\cdots;i_{N-1}j_{N-1};24} \\ x_{i_1j_1;\cdots;i_{N-1}j_{N-1};31} & x_{i_1j_1;\cdots;i_{N-1}j_{N-1};32} & \varnothing & \varnothing \\ \varnothing & \varnothing & x_{i_1j_1;\cdots;i_{N-1}j_{N-1};43} & x_{i_1j_1;\cdots;i_{N-1}j_{N-1};44} \end{pmatrix}, \tag{22}$$

where $1 \le i_k, j_k \le 4$, and $1 \le k \le N$. The construction contains a self-similarity property in $\mathbb{X}_N$. As discussed in the previous section, $x_{i_1j_1;i_2j_2;\cdots;i_{N-1}j_{N-1};i_Nj_N}$ means the pattern

$$(a_{r_{11}r_{12}}a_{r'_{12}r_{13}}) \diamond (a_{r_{21}r_{22}}a_{r'_{22}r_{23}}) \diamond \cdots \diamond (a_{r_{N1}r_{N2}}a_{r'_{N2}r_{N3}})$$

in $\{+,-\}^{\mathbb{Z}_{3\times N}}$, where $a_{r_{k1}r_{k2}}a_{r'_{k2}r_{k3}}$ is defined in (17), and

$$r_{k1} = \left\lfloor \frac{i_k - 1}{2} \right\rfloor, \quad r_{k2} = i_k - 1 - 2r_{k1}, \quad r'_{k2} = \left\lfloor \frac{j_k - 1}{2} \right\rfloor, \quad r_{k3} = j_k - 1 - 2r'_{k2}.$$

The pattern is $\varnothing$ if $a_{r_{k1}r_{k2}}a_{r'_{k2}r_{k3}} = \varnothing$ for some $1 \le k \le N$. Otherwise, it is denoted by the pattern

$$(a_{r_{11}}a_{r_{12}}a_{r_{13}}) \diamond (a_{r_{21}}a_{r_{22}}a_{r_{23}}) \diamond \cdots \diamond (a_{r_{N1}}a_{r_{N2}}a_{r_{N3}})$$

in $\{+,-\}^{\mathbb{Z}_{3\times N}}$. The following Theorem asserts that enlarging the local patterns to be rectangles helps for the determination of the transition matrix $\mathbf{T}$ of the solution space.

Theorem 3.6 Suppose $\mathbf{T}$ is the transition matrix of the solution space of (3) with nearest neighborhood, and T_k is the transition matrix of the kth layer. Then

$$\mathbf{T} = (\mathbf{T_N} \otimes \mathbf{E_{4^{N-1}}}) \circ (\mathbf{E_4} \otimes \mathbf{T_{N-1}}) \in \mathcal{M}_{\mathbf{4^{n+1}\times 4^{n+1}}}(\mathbb{R}), \tag{23}$$

where

$$\mathbf{T_k} = (\mathbf{T_k} \otimes \mathbf{E_{4^{k-1}}}) \circ (\mathbf{E_4} \otimes \mathbf{T_{k-1}}) \in \mathcal{M}_{\mathbf{4^{k+1}\times 4^{k+1}}}(\mathbb{R}), \quad \text{for } 3 \le \mathrm{k} \le \mathrm{N}-1, \tag{24}$$

and

$$\mathbf{T_2} = \mathbf{T_2} \circ (\mathbf{E_4} \otimes \mathbf{T_1}) \in \mathcal{M}_{\mathbf{16\times 16}}(\mathbb{R}). \tag{25}$$

3.3 MCNNs with initial values

In this section we review the studies on the MCNN with initial conditions. Recently, Ban and Chang [12] proved that the solution space of MCNN with initial condition forms a so-called *path set* in symbolic dynamics. To ease the notation, let $f(\mathbf{x}) = (\mathbf{f}(\mathbf{x}_\mathbf{i}^{(\mathbf{n})}))_{\mathbf{i}\in\mathbb{N},\mathbf{1}\le\mathbf{n}\le\mathbf{N}}$, where $\mathbf{x} = (\mathbf{x}_\mathbf{i}^{(\mathbf{n})})_{\mathbf{i}\in\mathbb{N},\mathbf{1}\le\mathbf{n}\le\mathbf{N}} \in \mathbb{R}^{\infty\times\mathbf{N}}$. Set

$$\mathbf{Y_a} = \{\mathbf{y} = (\mathbf{y}_\mathbf{i}^{(\mathbf{n})})_{\mathbf{i}\in\mathbb{N},\mathbf{1}\le\mathbf{n}\le\mathbf{N}} \in \mathbb{R}^{\infty\times\mathbf{N}} \mathbf{:}\ \mathbf{y} = \mathbf{f}(\mathbf{x}), \mathbf{x} \in \mathbf{X_a}\}. \tag{26}$$

For $1 \le n \le N$, define $\Phi^{(n)} : \mathbb{R}^{\infty\times N} \to \mathbb{R}^{\infty\times 1}$ by $\Phi^{(n)}(\mathbf{x}) = (\mathbf{x}_\mathbf{i}^{(\mathbf{n})})_{\mathbf{i}\in\mathbb{N}}$, where $\mathbf{x} = (\mathbf{x}_\mathbf{i}^{(\mathbf{n})})_{\mathbf{i}\in\mathbb{N},\mathbf{1}\le\mathbf{n}\le\mathbf{N}}$. Set

$$\mathbf{Y}_{\mathbf{a}}^{(\mathbf{n})} = \{\mathbf{y} = (\mathbf{y_i})_{\mathbf{i}\in\mathbb{N}} : \mathbf{y} = \mathbf{\Phi}^{(\mathbf{n})}(\mathbf{y}), \mathbf{y} \in \mathbf{Y}\}.$$

We call $\mathbf{Y_a}$ the solution space, $\mathbf{Y}_{\mathbf{a}}^{(\mathbf{N})}$ the output space, and $\mathbf{Y}_{\mathbf{a}}^{(\mathbf{n})}$ with $n \neq N$ the n-th hidden space of (3) with initial condition.

$$(x_\ell^{(n)})_{n=1}^N(0) = \mathbf{a}, \ell \in \mathbb{N}, \mathbf{a} \in \mathbb{R}^{\mathbf{N}}. \tag{27}$$

We leave it as an open problem to characterize the complete stability for mosaic solutions of IVP of MCNN.

Problem 3.7 Find the conditions for the complete stability for mosaic solutions of IVP of MCNN.

To understand the underlying symbolic space of $\mathbf{Y}_{\mathbf{a}}^{(\mathbf{N})}$ and $\mathbf{Y}_{\mathbf{a}}^{(\mathbf{n})}$ for $n \neq N$, we introduce the notion of path set which has been defined by Abram and Lagarias [1] recently.

Definition 3.8 (*Path set* [1]). Suppose $\mathcal{G} = (G, \mathcal{L})$ is a labeled graph with underlying directed graph $G = (\mathcal{V}, \mathcal{E})$ and labeling $\mathcal{L} : \mathcal{E} \to \mathcal{A}$. The *path set* (or *pointed follower set*) $\mathcal{P} = X_{\mathcal{G}}(v)$ is the subset of $\mathcal{A}^{\mathbb{N}}$ made up of the symbol sequences of successive edge labels of all possible one-sided infinite walks in $\mathcal{G}$ issuing from the distinguished vertex v. Many different $(\mathcal{G}, v)$ may give the same path set $\mathcal{P} \subset \mathcal{A}^{\mathbb{N}}$, and we call any such $(\mathcal{G}, v)$ a presentation of $\mathcal{P}$.

The following theorem unveils that the solution space $\mathbf{Y}_{\mathbf{a}}^{(\mathbf{n})}$of MCNN with initial condition forms a Path set for $n = 1, \ldots, N$.

Theorem 3.9 (Theorem 1.2. [12]). Given $\mathbf{a} \in \mathbb{R}^{\mathbf{n}}$. Suppose $\mathbf{Y}_{\mathbf{a}}^{(\mathbf{N})}$, and $\mathbf{Y}_{\mathbf{a}}^{(\mathbf{n})}$ are the solution, output, and hidden spaces of 3 with initial condition 27, $1 \leq n \leq N-1$. Then $\mathbf{Y}_{\mathbf{a}}^{(\mathbf{n})}$ are topologically conjugated to path sets for $1 \leq n \leq N$.

The identification of the solution space of MCNN with initial condition with Path set helps us to compute their spatial entropy. There is another definition for path set, called the path topological entropy.

Definition 3.10 (*Theorem 1.7.* [1]) Suppose $\mathcal{P}$ is a path set. Let $N_n^I(\mathcal{P})$ denote the number of distinct initial blocks of length n in $\mathcal{P}$. The *path topological entropy* of $\mathcal{P}$ is defined by

$$h_p(\mathcal{P}) = \limsup_{n\to\infty} \frac{1}{n} \log N_n^I(\mathcal{P}).$$

The difference between the spatial entropy (6) and the path topological entropy is that the spatial entropy considers the growth rate of " all" distinct blocks in the given space [3]. In [1], the authors showed that the spatial entropy of a path set coincides with its path topological entropy.

Theorem 3.11 (Theorem 1.8. [1]). For a path set $\mathcal{P}$, we have

$$h_P(\mathcal{P}) = h(\mathcal{P}).$$

It is known that the spatial entropy of a sofic shift relates to the maximal eigenvalue of its corresponding transition matrix. Theorems 3.9 and 3.11 indicate that the path topological entropy of either one of the solution, hidden, and output spaces also relates to the maximal eigenvalue of their corresponding transition matrices if the transition matrix comes from a *reachable presentation*. Herein a presentation $(\mathcal{G}, v)$ of a path set $\mathcal{P}$ is called *reachable* if each vertex of $\mathcal{G}$ can be reached by a directed path from v.

Theorem 3.12 (Theorem 1.3. [12]). Suppose $\mathbf{X} \in \{\mathbf{Y_a}, \mathbf{Y}_{\mathbf{a}}^{(\mathbf{1})}, \ldots, \mathbf{Y}_{\mathbf{a}}^{(\mathbf{n})}\}$ and $(\mathcal{G}, v)$ is a reachable presentation of $\mathbf{X}$. If the labeled graph $\mathcal{G}$ is right-resolving, then

$$h(\mathbf{X}) = \log \rho(\mathbf{T}).$$

where T is the transition matrix of $\mathcal{G}$.

Remark 3.13 For any path set $\mathcal{P} = (\mathcal{G}, v)$, Theorem 3.2. of [1] ensures that there always exists such presentation in Theorem 3.12. That is, $\mathcal{G}$ is right-resolving and $(\mathcal{G}, v)$ is reachable.

4 Multiple cellular neural networks

This section considers the ICNNs, i.e., inhomogeneous CNN. As we mentioned in introduction, we also call them MPCNNs. As usual $\mathbb{T} = [\mathbf{A}, \mathbf{z}]$ is the template according to (5). Two different types, namely, constant- and arithmetic-type MPCNNs are discussed herein. One may see that such two different structures of interaction will produce different multiple symbolic spaces and these two types include most class of MPCNNs.

4.1 Constant-type MPCNN

A MPCNN is called a *constant-type* (CCNN) if the neighborhood $\mathbf{N} = \{\mathcal{N}_\rangle \neg \rangle \in \mathbb{Z}\}$ is a finite set and the threshold $\mathbf{z}$ consists of finite components. It is seen that the CCNNs generalize the concept of the CNNs that were introduced in [25, 26]. More precisely, a CNN is a CCNN with $\mathcal{N}_i = \mathcal{N}_j$ and $\mathbf{A_i} = \mathbf{A_j}$ for all $i, j \in \mathbb{Z}$. An essential description of a CCNN is that there exists a positive integer $\ell \geq 2$ such that $\mathbf{z} = [\mathbf{z_1}, \ldots, \mathbf{z}_\ell]$ and $\mathbf{N} = \{\mathcal{N}_\infty, \ldots, \mathcal{N}_\ell\}$ satisfy $\mathcal{N}_{p+q\ell} = \mathcal{N}_p, z_{p+q\ell} = z_p$ for $1 \leq p \leq \ell, q \in \mathbb{Z}$. Restated, a one-dimensional CCNN is of the form

$$\frac{d}{dt}x_i(t) = -x_i(t) + z_{\bar{i}} + \sum_{k\in\mathcal{N}_{\bar{i}}} a_{k;\bar{i}} f(x_{i+\ell k}(t)), \tag{28}$$

where $1 \le \bar{i} \le \ell$ and $\bar{i} = i \pmod{\ell}$. Without loss of generality, we assume that every neighborhood $\mathcal{N}_i = \{-1, 0, 1\}$, i.e., the nearest neighborhood. In this case, the feedback template of (28) is $\mathbf{A} = [\mathbf{A_j}]_{\mathbf{1}\le\mathbf{j}\le\ell}$, where $\mathbf{A_j} = [a_{-1;j}, a_{0;j}, a_{1;j}]$.

Let $\Lambda = \{1, \ldots, \ell\}$. The one-dimensional lattice $\mathbb{Z}$ can be decomposed into ℓ non-overlapping subspaces

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

Set $j_i = j + \ell i$. Equation (28) can then be restated as follows.

$$\frac{d}{dt}x_{j_i} = -x_{j_i} + z_j + \sum_{|k|\le 1} a_{k;j} f(x_{j_{i+k}}), \quad j \in \Lambda, i \in \mathbb{Z}. \tag{29}$$

Suppose y is a mosaic pattern, for each $j \in \Lambda$ and $i \in \mathbb{Z}$, the construction of $\mathcal{B}_j(+)$ and $\mathcal{B}_j(-)$ is the same as Sect. 2 for $j \in \Lambda$. The set of admissible local patterns $\mathcal{B}$ of a constant CNN is then

$$\mathcal{B}(\mathbb{T}, \mathbf{z}) = (\mathcal{B}_{\mathbf{1}}(+), \ldots, \mathcal{B}_\ell(+), \mathcal{B}_{\mathbf{1}}(-), \ldots, \mathcal{B}_\ell(-)).$$

Similar to the discussion in [50], the output space $\mathbf{Y}$ can be represented as

$$\mathbf{Y} = \{y = (y_{j_i}) : y_{j_{i-d}} \cdots y_{j_i} \cdots y_{j_{i+d}} \in (\mathcal{B}_j(+), \mathcal{B}_j(-)) \text{ for } \mathrm{j} \in \Lambda, \mathrm{i} \in \mathbb{Z}\}.$$

Ban and Chang [11] showed that $\mathbf{Y}$ is topologically conjugated to the direct product of the output spaces $\mathbf{Y_j}$ of the classical CNNs, where $\mathbf{Y_j}$ is determined by $(B)_j = (\mathcal{B}_j(+), \mathcal{B}_j(-))$ for $j \in \Lambda$. That is, $\mathbf{Y} \cong \prod_{\mathbf{j}\in\Lambda} \mathbf{Y_j}$.

Theorem 4.1 (Theorem 4.9 [11]). *The spatial entropy of the output solution* $\mathbf{Y}$ *for the constant-type multiple cellular neural networks* (28) *is*

$$h(\mathbf{Y}) = \frac{\mathbf{1}}{\ell} \sum_{\mathbf{j}\in\Lambda} \mathbf{h}(\mathbf{Y_j}).$$

4.2 Arithmetic-Type MPCNN

A MPCNN is called an *arithmetic-type* (ACNN) if there exists a positive integer $\ell \ge 2$ such that

$$\mathcal{N}_i = \mathcal{N}_j, \mathrm{z_i} = \mathrm{z_j} \quad \text{whenever} \quad \frac{\mathrm{j}}{\mathrm{i}} = 0\,(\mathrm{mod}\ell).$$

The essential description of an ACNN is that $\mathbf{z} = [\mathbf{z_j}]_{\ell\nmid\mathbf{j}}$ and $\mathbf{N} = \{\mathcal{N}_|\}_{\ell\nmid\mathbf{i}}$. More precisely, an ACNN with nearest neighborhood is realized as the form

$$\frac{d}{dt}x_i(t) = -x_i(t) + z_{\bar{i}} + \sum_{k\in\mathcal{N}_{\bar{i}}} a_{k;\bar{i}} f(x_{i\cdot\ell^{i+k}}(t)), \quad i \in \mathbb{N}, \tag{30}$$

where $\ell\nmid\bar{i}$, $i \ge \bar{i}$, $\frac{i}{\bar{i}} = 0 \pmod{\ell}$, and $\mathcal{N}_{\bar{i}} = \{0, 1\}$.

Let $\Lambda = \{j : \ell\nmid j\}$ be an infinite index set. The set of positive integers $\mathbb{N}$ is then decomposed into the disjoint union of infinitely many subsets by

$$\mathbb{N} = \bigcup_{j\in\Lambda} \mathbb{N}_j = \bigcup_{j\in\Lambda} \{j\ell^i : i \ge 0\} := \bigcup_{j\in\Lambda} \{j_i\}_{i\ge 0}.$$

(30) can then be represented as

$$\frac{d}{dt}x_{j\ell^i}(t) = -x_{j\ell^i}(t) + z_j + \sum_{0\le k\le 1} a_{k;j} f(x_{j\ell^{i+k}}(t)), \quad j \in \Lambda, i \ge 0. \tag{31}$$

In this case, the feedback template $\mathbf{A} = [\mathbf{A_j}]_{\mathbf{j}\in\Lambda}$ consists of **infinitely many** smaller templates $\mathbf{A_j} = [\mathbf{a_{0;j}}, \mathbf{a_{1;j}}]$, and the threshold is $\mathbf{z} = [\mathbf{z_j}]_{\mathbf{j}\in\Lambda}$.

Suppose y is a mosaic pattern, for each $j \in \Lambda$ and $i \ge 0$, the necessary and sufficient condition for $y_{j_i} = 1$ is

$$a_{0;j} - 1 + z_j > a_{1;j} y_{j_{i+1}}, \tag{32}$$

and the necessary and sufficient condition for $y_{j_i} = -1$ is

$$a_{0;j} - 1 - z_j > a_{k;1} y_{j_{i+1}}. \tag{33}$$

Set

$$\mathcal{B}_j(+) = \left\{ \boxed{y_0 y_1} : y_1 \in \{-1, 1\} \text{ satisfy } (32), y_0 = 1 \right\},$$

$$\mathcal{B}_j(-) = \left\{ \boxed{y_0 y_1} : y_1 \in \{-1, 1\} \text{ satisfy } (33), y_0 = -1 \right\}.$$

The set of admissible local patterns $\mathcal{B}$ of a constant CNN is then

$$\mathcal{B}(\mathbb{T}, \mathbf{z}) = (\mathcal{B}_{\mathbf{j}}(+), \mathcal{B}_{\mathbf{j}}(-))_{\mathbf{j}\in\Lambda}.$$

The output space $\mathbf{Y}$ is then represented as

$$\mathbf{Y} = \{\mathbf{y} = (\mathbf{y_{j_i}}) : \mathbf{y_{j_i}} \cdots \mathbf{y_{j_{i+1}}} \in (\mathcal{B}_{\mathbf{j}}(+), \mathcal{B}_{\mathbf{j}}(-)) \mathrm{for} \mathrm{j} \in \Lambda, \mathrm{i} \ge 0\}.$$

Similar to that the output space $\mathbf{Y}$ of a constant CNN can be decomposed into finitely many subspaces $\mathbf{Y_j}$ such that $\mathbf{Y_j}$ is a SFT for each j. The output space of an arithmetic CNN is decomposed into countable subspaces; more precisely, $\mathbf{Y} \cong \prod_{\mathbf{j}\in\Lambda} \mathbf{Y_j}$, where $\mathbf{Y_j}$ is determined bythe basic set of admissible local pattern $\mathcal{B}_j = (\mathcal{B}_j(+), \mathcal{B}_j(-))$.

A set function $\chi : 2^{\mathbb{R}} \to \{0, 1\}^{\mathbb{R}}$ is defined by $\chi(E)(x) := \chi_E(x) = 1$ if and only if $x \in E$ for E being a non-empty subset of $\mathbb{R}$. For $n \in \mathbb{N}$ and $j \in \Lambda$ such that $j < n$, define

$$k_j(n) = \left[\log_\ell \frac{n}{j}\right],$$
$$K_j(n) = (k_j(n) - d_j + 1)\chi_{\mathbb{N}}(k_j(n) - d_j + 1).$$

It is seen that both $k_j(n)$ and $K_j(n)$ are non-negative integers. To compute the spatial entropy of ACNN, we introduce some notations first. Set

$$m_j(n) = \begin{cases} \left\| T_j^{k_j(n)} \right\| & \text{if } \mathrm{K_j(n)} > 0; \\ 2^{k_j(n)+1} & \text{otherwise.} \end{cases} \tag{34}$$

Recall that T_j is the transition matrix of $\mathbf{Y_j}$. The spatial entropy of ACNN can be computed as follows.

Theorem 4.2 (Theorem 3.1. [11]). Suppose that there exists $d \in \mathbb{N}$ such that $d_j \leq d$ for $j \in \Lambda$. Then the spatial entropy of the output solution $\mathbf{Y}$ for the arithmetic-type multiple cellular neural networks (31) is

$$h(\mathbf{Y}) = \lim_{\mathbf{n}\to\infty} \frac{\mathbf{1}}{\mathbf{n}} \sum_{\mathbf{j} \leq \mathbf{n}, \mathbf{j}\in\Lambda} \log \mathbf{m_j}(\mathbf{n}), \tag{35}$$

where $m_j(n)$ is defined in (34).

5 Conclusion

In this survey we review the recent results for computing the topological entropy for various types of CNN

(1) Classical cellular neural networks (Theorem 2.2):
   (a) The solutions structure forms a subshift of finite type;
   (b) The topological entropy is the logarithm of the maximal eigenvalue of its transition T.

(2) Multi-layer cellular neural networks (Theorem 2.2):
   (a) The solutions structure forms a sofic shift;
   (b) If the labeled graph is right-resolving, then the topological entropy is the logarithm of the maximal eigenvalue of its transition T;
   (c) If the labeled graph is not right-resolving, one can use subset construction to form a right-resolving one.

(3) Multi-layer cellular neural networks with initial values:
   (a) The solutions structure forms a path set (Theorem 3.9);
   (b) If the presentation $(\mathcal{G}, v)$ is right-resolving and reachable, then the topological entropy is the logarithm of the maximal eigenvalue of its transition T (Theorem 3.12);
   (c) There always exists a presentation $(\mathcal{G}, v)$ of a path set which is right-resolving and reachable.

(4) Multiple cellular neural networks:
   (a) The solutions structures forms a multiple shift (Sect. 4);
   (b) The entropy formula is presented in Theorems 4.1 and 4.2.

5.1 Problems

We list some open problems for further study.

Problem 5.1 General method on the computation of spatial entropy for MPCNN? For Arithmetic-Type MPCNN, Theorem 4.2 provides formula on computing the entropy. However, the formula rely heavily on its architecture and rules on each $\mathbf{Y_j}$. Therefore, a general method on the computation of entropy for a given MPCNN is still lacking.

Problem 5.2 As Problem 5.1, what is the rigorous number of (35) in Theorem 4.2?

Problem 5.3 General method for the computation of the number of n-cylinders of $\mathbf{Y}$ with some boundary conditions for MPCNN. For example, *periodic*, *Dirichlet* and *Neumann*.

Problem 5.4 (*Realization problem*). What kind of values of spatial entropy can be realized by a CNN, MCNN, MCNN with initial values or ICNN?

5.2 Outlooks

[2-d CNN] How to compute the spatial entropy for 2-dimensional CNN, MCNN, MCNN with initial values or ICNN? What are their underlying spaces? These are very difficult problems, even in the symbolic dynamical systems, only few result is known.

[Multiple-valued output functions] Note that f in (2) is 2-valued output function. That is, it has only 2 saturate states $\{+1, -1\}$. Recently, there are more and more research on designing CNN by using multiple-valued output function [49, 53] and apply to associative memories [51, 52, 79, 81] and disease diagnosis [4, 80]. Nature problems arise: How to compute the entropy for CNN with multiple-valued output function? What are their dynamics, e.g., stability, synchronization or machine learning related problems?

Acknowledgments The author thanks for the anonymous referees' valuable opinions. The suggestions improve this paper and motivate some further works.

References

1. Abram W, Lagarias JC (2012) Path sets and their symbolic dynamics. arXiv:1207.5004 [math.DS]
2. Abram W, Lagarias JC (2013) p-adic path set fractals and arithemetics. arXiv:1210.2478 [math.MG]
3. Adler RL, Marcus B (1979) Topological entropy and equivalence of dynamical systems. Memoris of American Mathematical Society, vol 219, Amarican Mathematical Society, Providence
4. Akiduki T, Zhong Z, Imamura T, Miyake T (2009) Associative memories with multi-valued cellular neural networks and their application to disease diagnosis. In: IEEE international conference on systems, man and cybernetics, SMC 2009, pp 3824–3829
5. Alecsandrescu I, Ungureanu P, Goras L (2008) Nonhomogeneous cnns and their use for texture classification. In: 9th symposium on IEEE neural network applications in electrical engineering, NEUREL 2008, pp 153–156
6. Alsultanny YA, Aqul MM (2003) Pattern recognition using multilayer neural-genetic algorithm. Neurocomputing 51:237–247
7. Andreyev Y, Belsky Y, Dmitriev A, Kuminov D (1992) Inhomogeneous cellular neural networks: possibility of functional device design. In: IEEE second international workshop on CNNA-92 proceedings cellular neural networks and their applications, pp 135–140
8. Arena P, Baglio S, Fortuna L, Manganaro G (1998) Self-organization in a two-layer CNN. IEEE Trans Circuits Syst I Fundam Theory Appl 45:157–162
9. Ayoub A, Tokes S (2005) A new cnn template for poac peak enhancement. In: Proceedings of the 2005 European conference on circuit theory and design, vol 1, pp 1–157
10. Ban J-C, Chang C-H (2009) On the monotonicity of entropy for multi-layer cellular neural networks. Int J Bifurc Chaos Appl Sci Eng 19:3657–3670
11. Ban J-C, Chang C-H (2013) Inhomogeneous lattice dynamical systems and the boundary effect. Bound Value Probl 2013(1):249
12. Ban J-C, Chang C-H (2013) Solution structure of multi-layer neural networks with initial conditions, (submitted)
13. Ban J-C, Chang C-H, Lin S-S (2012) The structure of multi-layer cellular neural networks. J Differ Equ 252:4563–4597
14. 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
15. Ban J-C, Hu W-G, Lin S-S, Lin Y-H (2013) Zeta functions for two-dimensional shifts of finite type. Memoirs of the American Mathematical Society 221, no.1037
16. Ban J-C, Lin S-S (2005) Patterns generation and transition matrices in multi-dimensional lattice models. Discret Contin Dyn Syst 13(3):637
17. Ban J-C, Lin S-S, Shih C-W (2001) Exact number of mosaic patterns in cellular neural networks. Int J Bifurc Chaos Appl Sci Eng 11:1645–1653
18. 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, CNNA 2002, pp 444–451.
19. Chow S-N, Mallet-Paret J (1995) Pattern formation and spatial chaos in lattice dynamical systems: I and II. IEEE Trans Circuits Syst I Fundam Theory Appl 42:746–756
20. Chow S-N, Mallet-Paret J, Van Vleck ES (1996) Dynamics of lattice differential equations. Int J Bifurc Chaos Appl Sci Eng 6:1605–1621
21. Chow S-N, Mallet-Paret J, Van Vleck ES (1996) Pattern formation and spatial chaos in spatially discrete evolution equations. Random Comput Dyn 4:109–178
22. Chua LO (1998) Cnn: a paradigm for complexity. World scientific series on nonlinear science, series A, 31. World Scietific, Singapore
23. Chua LO, Roska T (2002) Cellular neural networks and visual computing. Cambridge University Press, Cambridge
24. Chua, LO Shi B-E (1991) Multiple layer cellular neural networks: a tutorial. In: Deprettere EF, van der Veen A-F (eds) Algorithms and parallel VLSI architectures, pp 137–168. Elsevier, Amsterdam
25. Chua LO, Yang L (1988) Cellular neural networks: applications. IEEE Trans Circuits Syst 35:1273–1290
26. Chua LO, Yang L (1988) Cellular neural networks: theory. IEEE Trans Circuits Syst 35:1257–1272
27. Civalleri PP, Gilli M, Pandolfi L (1993) On stability of cellular neural networks with delay. In: IEEE transactions on circuits and systems I: fundamental theory and applications 40(3):157–165
28. Crounse KR, Chua LO (1995) Methods for image processing and pattern formation in cellular neural networks: a tutorial. IEEE Trans Circuits Syst 42:583–601
29. Crounse KR, Roska T, Chua LO (1993) Image halftoning with cellular neural networks. IEEE Trans Circuits Syst 40:267–283
30. Dahl GE, Yu D, Deng L, Acero A (2012) Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. In: IEEE transactions on audio, speech, and language processing, 20(1):30–42
31. Dawes JHP (2008) Localised pattern formation with a large-scale mode: slanted snaking. SIAM J Appl Dyn Syst 7:186–206
32. Dawes JHP, Lilley S (2010) Localized states in a model of pattern formation in a vertically vibrated layer. SIAM J Appl Dyn Syst 9:238–260
33. Doebeli M, Hauert C, Killingback T (2004) The evolutionary origin of cooperators and defectors. Science 306:859–862
34. Doebeli M, Killingback T (2003) Metapopulation dynamics with quasi-local competition. Theor Popul Biol 64:397–416
35. Dominguez-Castro R, Espejo S, Rodriguez-Vazquez A, Carmona R (1994) A cnn universal chip in cmos technology. In: IEEE proceedings of the third ieee international workshop on cellular neural networks and their applications, CNNA-94, pp 91–96
36. Einsiedler ML, Ward T (2011) Ergodic theory: with a view towards number theory, vol 259. Springer, Berlin
37. Espejo S, Carmona R, Dominguez-Castro R, Rodrigúez-Vázquez A (1996) A cnn universal chip in cmos technology. Int J Circuit Theory Appl 24(1):93–109
38. Espejo S, Carmona R, Domínguez-Castro R, Rodriguez-Vazquez A (1996) A vlsi-oriented continuous-time cnn model. Int J Circuit Theory Appl 24(3):341–356
39. Fan A-H, Liao L, Ma J-H (2012) Level sets of multiple ergodic averages. Monatsh Math 168:17–26
40. Fukushima K (2013) Artificial vision by multi-layered neural networks: neocognitron and its advances. Neural Netw 37:103–119
41. Fukushima K (2013) Training multi-layered neural network neocognitron. Neural Netw 40:18–31
42. Furstenberg H (1977) Ergodic behavior of diagonal measures and a theorem of szemeredi on arithmetic progressions. J Anal Math 31:204–256
43. Golubitsky M, Stewart I (2002) The symmetry perspective: from equilibrium to chaos in phase space and physical space. Progress in Mathematics, vol 200. Birkhauser, Basel
44. Golubitsky M, Stewart I, Török A (2005) Patterns of symmetry in coupled cell networks with multiple arrows. SIAM J Appl Dyn Syst 4:78–100
45. Hauert C, Doebeli M (2004) Spatial structure often inhibits the evolution of cooperation in the snowdrift game. Nature 428:643–646

46. Hinton G, Deng L, Yu D, Dahl GE, Mohamed A-R, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath TN et al (2012) Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE signal processing magazine 29(6):82–97
47. Hsu C-H, Juang J, Lin S-S, Lin W-W (2000) Cellular neural networks: local patterns for general template. Int J Bifurc Chaos Appl Sci Eng 10:1645–1659
48. Hsu C-H, Yang T-H (2002) Abundance of mosaic patterns for cnn with spatial variant templates. Int J Bifurc Chaos Appl Sci Eng 12:1321–1332
49. Itoh M, Julián P, Chua LO (2001) Rtd-based cellular neural networks with multiple steady states. Int J Bifurc Chaos 11(12):2913–2959
50. Juang J, Lin S-S (2000) Cellular neural networks: mosaic pattern and spatial chaos. SIAM J Appl Math 60:891–915
51. Kamio T, Fujisaka H, Morisue M (2002) Associative memories using interaction between multilayer perceptrons and sparsely interconnected neural networks. IEICE Trans Fundam Electron Commun Comput Sci 85(6):1220–1228
52. Kamio T, Morisue M (2003) A synthesis procedure for associative memories using cellular neural networks with space-invariant cloning template library. In: IEEE proceedings of the international joint conference on neural networks, vol 2, pp 885–890
53. Kanagawa A, Kawabata H, Takahashi H (1996) Cellular neural networks with multiple-valued output and its application. IEICE Trans Fundam Electron Commun Comput Sci 79(10):1658–1663
54. Killingback T, Loftus G, Sundaram B (2012) Competitively coupled maps and spatial pattern formation. arXiv:1204.2463
55. Lin S-S, Shih C-W (1999) Complete stability for standard cellular neural networks. Int J Bifurc Chaos Appl Sci Eng 9:909–918
56. Lin S-S, Yang T-S (2000) Spatial entropy of one-dimensional cellular neural network. Int J Bifurc Chaos 10(09):2129–2140
57. Lind D, Marcus B (1995) An introduction to symbolic dynamics and coding. Cambridge University Press, Cambridge
58. Mohamed A-R, Yu D, Deng L (2010) Investigation of full-sequence training of deep belief networks for speech recognition. In: INTERSPEECH, pp 2846–2849
59. Murugesh V (2010) Image processing applications via time-multiplexing cellular neural network simulator with numerical integration algorithms. Int J Comput Math 87:840–848
60. Roska T, Chua LO (1993) The cnn universal machine: an analogic array computer. In: IEEE transactions on circuits and systems II: analog and digital signal processing, 40(3):163–173
61. Roska T, Wu C-W, Balsi M, Chua LO (1992) Stability and dynamics of delay-type general and cellular neural networks. In: IEEE transactions on circuits and systems I: fundamental theory and applications, 39(6):487–490
62. Roska T, Wu C-W, Chua LO (1993) Stability of cellular neural networks with dominant nonlinear and delay-type templates. In: IEEE Transactions on circuits and systems I: fundamental theory and applications, 40(4):270–272
63. Seide F, Li G, Chen X, Yu D (2011) Feature engineering in context-dependent deep neural networks for conversational speech transcription. In: IEEE Workshop on automatic speech recognition and understanding (ASRU), pp 24–29
64. Seide F, Li G, Yu D (2011) Conversational speech transcription using context-dependent deep neural networks. In: INTERSPEECH, pp 437–440.
65. Shih C-W (2001) Complete stability for a class of cellular neural networks. Int J Bifurc Chaos Appl Sci Eng 11:169–178
66. Stewart I (2004) Networking opportunity. Nature 427:601–604
67. Stewart I, Golubitsky M, Pivato M (2003) Symmetry groupoids and patterns of synchrony in coupled cell networks. SIAM J Appl Dyn Syst 2:609–646
68. Török L, Roska T (2004) Stability of multi-layer cellular neural/nonlinear networks. Int J Bifurc Chaos Appl Sci Eng 14:3567–3586
69. Walters P (1982) An introduction to ergodic theory. Springer, New York
70. Xavier de Souza S, Yalcin ME, SuykensJAK, Vandewalle J (2004) Toward CNN chip-specific robustness. In: IEEE transactions on circuits and systems I: regular papers 51:892–902
71. Yang L, Chua LO, Krieg KR (1990) VLSI implementation of cellular neural networks. In: IEEE international symposium on circuits and systems, pp 2425–2427
72. Yang T, Yang L-B (1996) The global stability of fuzzy cellular neural network. In: IEEE transactions on circuits and systems I: fundamental theory and applications, 43(10):880–883
73. Yang Z, Nishio Y, Ushida A (2001) A two layer cnn in image processing applications. In: Proceedings of the 2001 international symposium on nonlinear theory and its applications, pp 67–70
74. Yang Z, Nishio Y, Ushida A (2002) Image processing of two-layer CNNs: applications and their stability. IEICE Trans Fundam E85–A:2052–2060
75. Yokozama M, Kubota Y, Hara T (1998) Effects of competition mode on spatial pattern dynamics in plant communities. Ecol Model 106:1–16
76. 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
77. Yu D, Seide F, Li G, Deng L (2012) Exploiting sparseness in deep neural networks for large vocabulary speech recognition. In: IEEE International conference on acoustics, speech and signal processing (ICASSP), pp 4409–4412
78. Yu D, Seltzer ML (2011) Improved bottleneck features using pretrained deep neural networks. In: INTERSPEECH, pp 237–240
79. Zhang Z, Akiduki T, Miyake T, Imamura T (2006) Design of multi-valued cellular neural networks for associative memory. In: IEEE international joint conference on SICE-ICASE, pp 4057–4061
80. Zhang Z, Liu Z-Q, Kawabata H (1999) Tri-output cellular neural network and its application to diagnosing liver diseases. In: IEEE SMC'99 conference proceedings. 1999 IEEE international conference on systems, man, and cybernetics, vol 3, pp 372–377
81. Zhang Z, Taniai R, Akiduki T, Imamura T, Miyake T (2009) A new design method of multi-valued cellular neural networks for associative memory. In: IEEE 2009 fourth international conference on innovative computing, information and control (ICICIC), pp 1562–1565