Neural Networks 70 (2015) 9–17

Contents lists available at ScienceDirect

Neural Networks

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

Realization problem of multi-layer cellular neural networks

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

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

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

ARTICLE INFO

Article history:
Received 25 November 2014
Received in revised form 13 April 2015
Accepted 12 June 2015
Available online 22 June 2015

Keywords:
Multi-layer cellular neural networks
Sofic shifts
Learning problem
Covering space
Separation property
Topological entropy

ABSTRACT

This paper investigates whether the output space of a multi-layer cellular neural network can be realized via a single layer cellular neural network in the sense of the existence of finite-to-one map from one output space to the other. Whenever such realization exists, the phenomena exhibited in the output space of the revealed single layer cellular neural network is at most a constant multiple of the phenomena exhibited in the output space of the original multi-layer cellular neural network. Meanwhile, the computation complexity of a single layer system is much less than the complexity of a multi-layer system. Namely, one can trade the precision of the results for the execution time. We remark that a routine extension of the proposed methodology in this paper can be applied to the substitution of hidden spaces although the detailed illustration is omitted.

1. Introduction

Cellular neural networks (CNNs), introduced by Chua and Yang (1988a, 1988b), have been one of the most investigated paradigms in neural information processing (Chua, 1998). CNNs must be completely stable in a wide range of applications (e.g., pattern recognition), i.e., each trajectory should converge toward some stationary state. The study of stationary solutions is thus important. Moreover, the investigation of *mosaic solutions* is essential due to the importance of learning algorithms and the training process. Roughly speaking, a learning algorithm is more efficient if there are more abundant output patterns for a given CNN.

Coupled systems based on CNNs, namely *multi-layer cellular neural networks* (MCNNs), have received considerable attention and have been successfully applied to many areas such as signal propagation between neurons, image processing, pattern recognition, information technology, CMOS realization and VLSI implementation (Arena, Baglio, Fortuna, & Manganaro, 1998; Ban & Chang, 2013; Carmona, Jimenez-Garrido, Dominguez-Castro, Espejo, & Rodriguez-Vazquez, 2002; Chua & Roska, 2002; Chua & Shi, 1991; Chua & Yang, 1988a; Crounse & Chua, 1995; Crounse, Roska, & Chua, 1993; Li, 2009; Murugesh, 2010; Peng, Zhang, & Liao, 2009; Xavier-de Souza, Yalcin, Suykens, & Vandewalle, 2004; Yang, Nishio, & Ushida, 2001, 2002). The development of CNNs has been inspired by the visual systems of mammals (Fukushima, 2013a, 2013b). The sufficient conditions for the complete stability of MCNNs can be found in Török and Roska (2004). Just as with CNNs, the study of mosaic solutions is also important and interesting. Recently, Ban and Chang (2009) showed that for MCNNs, more layers infer that models are capable of more phenomena.

A multi-layer cellular neural network is represented as

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

for some integer $N \geq 2$, $i \in \mathbb{Z}$, and $t \geq 0$. The inputs for the neurons in the kth layer are the outputs of the $(k-1)$th layer in the proposed model (1) for $2 \leq k \leq N$. Fig. 1 shows the connections of a three-layer CNN with the nearest neighborhood. The so-called *neighborhood* $\mathcal{N}$ is a finite subset of integers $\mathbb{Z}$; the output function

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

* Corresponding author.
E-mail addresses: jcban@mail.ndhu.edu.tw (J.-C. Ban), chchang@nuk.edu.tw (C.-H. Chang).

http://dx.doi.org/10.1016/j.neunet.2015.06.003
0893-6080/

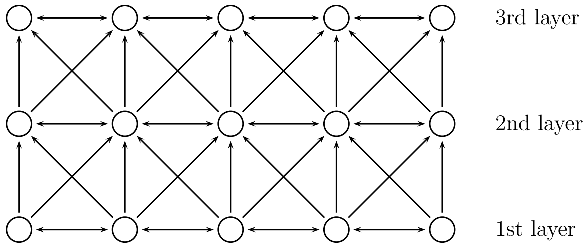

Fig. 1. Three-layer cellular neural networks with nearest neighborhood.

is a piecewise linear map. $\mathbb{A} = [A_1, \ldots, A_N]$ and $\mathbb{B} = [B_2, \ldots, B_N]$ are the feedback and controlling templates, respectively, where $A_j = [a_k^{(j)}]_{k\in\mathcal{N}}, B_l = [b_k^{(l)}]_{k\in\mathcal{N}}$ for $1 \le j \le N, 2 \le l \le N$; $\mathbb{z} = [z^{(1)}, \ldots, z^{(N)}]$ is the threshold. The template $\mathbb{T}$ of (1) consists of the feedback and controlling templates and the threshold, namely $\mathbb{T} = [\mathbb{A}, \mathbb{B}, \mathbb{z}]$. Note that (1) is a standard CNN if $N = 1$. This type of case is referred to as a *single layer CNN*.

A *mosaic solution* $\bar{x}$ is a solution of (1) which satisfies $|\bar{x}_i| > 1$ and its corresponding pattern $\bar{y} = (\bar{y}_i) = (f(\bar{x}_i))$ is called a *mosaic output pattern*. Since the output function (2) is piecewise linear with $f(x) = 1$ (resp. -1) if $x \ge 1$ (resp. $x \le -1$), the output of a mosaic solution $\bar{x} = (\bar{x}_i)_{i\in\mathbb{Z}}$ must be an element in $\Sigma = \{-1, +1\}^{\mathbb{Z}}$, which is why we call them *patterns*.

Given an N-layer MCNN with $N \ge 2$, we denote $\mathbf{Y}^{(N)}$ as the output solution space corresponding to a given input $(u_i)_{i\in\mathbb{Z}}$; namely,

$$\mathbf{Y}^{(N)} = \{(y_i)_{i\in\mathbb{Z}} : (y_i)_{i\in\mathbb{Z}} \text{ is a output solution for some input } (u_i)_{i\in\mathbb{Z}}\}.$$

A natural question arises: *when $Y^{(N)}$ behaves like a single layer CNN, can we find some map which links both systems*? Such a problem is called a *realization problem* since once this phenomenon has occurred, the given MCNN is realized by a single layer CNN system. From a mathematical point of view, the advantage of realizing a MCNN by a CNN is that one can *classify* MCNNs in terms of CNNs. If it indeed behaves like a single layer CNN, then it is easier to control. From an engineering perspective meanwhile, a realizable MCNN helps us to reduce the computation of such machines. In other words, this type of CNN is indeed a *shallow architecture* neural network model (cf. Bengio, 2009, Bengio & LeCun, 2007, Chang, 2015, Fukushima, 2013b, Hinton, Osindero, & Teh, 2006, Utgoff & Stracuzzi, 2002). Let $C(\Sigma, \Sigma)$ denote the set of continuous maps from Σ to Σ. A map $\tau \in C(\Sigma, \Sigma)$ is called *a factor* (resp. *an embedding*) if it is onto (resp. one-to-one). τ is called a conjugacy if it is both a factor and embedding. A realization problem can be achieved by raising the following problem.

Problem 1. Let $\mathbf{Y}^{(N)}$ be the output solution space of a MCNN (1) with $N \ge 2$.

(1) Does a pair $(\mathbf{Y}, \pi) \in \Sigma \times C(\Sigma, \Sigma)$ exist, where $\mathbf{Y}$ is the mosaic solution space of a single layer CNN and $\pi : \mathbf{Y} \to \mathbf{Y}^{(N)}$ is a factor from $\mathbf{Y}$ to $\mathbf{Y}^{(N)}$?
(2) Does π preserve the topological entropy, i.e., $h_{top}(\mathbf{Y}) = h_{top}(\pi(\mathbf{Y}))$?
(3) When does the factor π become an embedding, i.e., π is one-to-one?

Note that $\mathbf{Y}$ and $\mathbf{Y}^{(N)}$ are conjugate once (1) and (3) are satisfied.

It is worth pointing out that if π exists in the Problem 1–(1), then it links $\mathbf{Y}^{(N)}$ with some single layer CNN. Therefore, the output space $\mathbf{Y}^{(N)}$ is *controlled* by the factors π and $\mathbf{Y}$. If Problem 1-(2) holds, then the complexity of both $\mathbf{Y}^{(N)}$ and $\mathbf{Y}$ are the same, and it is important for the application of the learning algorithm. Finally, $\mathbf{Y}^{(N)}$ and $\mathbf{Y}$ are topologically the same if one ensures the factor π is also an embedding (Problem 1-(3)), and one simply replaces $\mathbf{Y}^{(N)}$ with $\mathbf{Y}$ in this case.

The aim of this paper is to study the above problem. Theorem 4.2 provides a natural and intrinsic characterization of Problem 1-(1) and 1-(2) by using the hidden Markov technique of symbolic dynamics. However, Problem 1-(3) is still unknown and is beyond the scope of the current study.

We also emphasize that one may raise the same problems on $\mathbf{Y}^{(i)}$ and $\mathbf{Y}^{(j)}$ for $1 \le i, j \le N$. More precisely, does a factor π between $\mathbf{Y}^{(i)}$ and $\mathbf{Y}^{(j)}$ exist for some $1 \le i, j \le n$ which preserves topological entropy? When does the factor π become an embedding? Some partial results are provided by Chang (2015). We emphasize that such a problem has to do with one given MCNN, and discuss the relationship between the output spaces of certain layers in such MCNNs. On the contrary, this study focuses on the relationship between a MCNN with other single layer CNNs. These two problems are different due to the fact that if $\mathbf{Y}^{(i)}$ and $\mathbf{Y}^{(j)}$ are extracted from the same MCNN, they inherit the same system information, making the discussion easier.

The rest of this paper is organized as follows. Section 2 considers the learning problem of two-layer cellular neural networks in pattern formation. Section 3 focuses on the realization problem of two-layer cellular neural networks, and the necessary and sufficient conditions for the existence of an entropy-preserving map between the output spaces of one and two-layer cellular neural networks. Following the discussion in Sections 2 and 3, Section 4 extends the results to general multi-layer cellular neural networks. Some discussion and suggestions for possible future research are given in Section 5 as a conclusion to the present work.

2. Learning problem of two-layer cellular neural networks

Learning problems (also called inverse problems) are some of the most investigated topics in a variety of disciplines. From a mathematical point of view, determining whether a given collection of output patterns can be seen through a CNN/MCNN is essential for the study of learning problems. This section reveals the necessary and sufficient conditions for the capability of exhibiting the output patterns of single layer cellular neural networks. The discussion can be applied to the elucidation of general cases, has addressed in Section 4.

A two-layer cellular neural network is seen as

$$\begin{cases} \dfrac{dx_i^{(2)}}{dt} = -x_i^{(2)} + \displaystyle\sum_{|k|\le d} a_k^{(2)} y_{i+k}^{(2)} + \sum_{|\ell|\le d} b_\ell^{(2)} u_{i+\ell}^{(2)} + z^{(2)}, \\ \dfrac{dx_i^{(1)}}{dt} = -x_i^{(1)} + \displaystyle\sum_{|k|\le d} a_k^{(1)} y_{i+k}^{(1)} + \sum_{|\ell|\le d} b_\ell^{(1)} u_{i+\ell}^{(1)} + z^{(1)}, \end{cases} \tag{3}$$

for some $d \in \mathbb{N}$, and $u_i^{(2)} = y_i^{(1)}$ for $i \in \mathbb{Z}$; $\mathbb{N}$ represents the set of positive integers and $\mathbb{Z}$ denotes the set of integers. The prototype of (3) is

$$\frac{dx_i}{dt} = -x_i + \sum_{|k|\le d} a_k y_{i+k} + \sum_{|\ell|\le d} b_\ell u_{i+\ell} + z. \tag{4}$$

Here $A = [-a_d, \ldots, a_d]$, $B = [-b_d, \ldots, b_d]$ are called *feedback* and *controlling templates*, respectively; z is known as the *threshold*, and $y_i = f(x_i) = \frac{1}{2}(|x_i + 1| - |x_i - 1|)$ is the output of x_i. The quantity x_i represents the state of the cell at i for $i \in \mathbb{Z}$. The output of a stationary solution $\bar{x} = (\bar{x}_i)_{i\in\mathbb{Z}}$ is called an output pattern. A *mosaic solution* $\bar{x}$ satisfies $|\bar{x}_i| > 1$ and its corresponding pattern $\bar{y}$ is called a *mosaic output pattern*. Considering the mosaic solution $\bar{x}$,