THE CRITERIA FOR SYMMETRIC VECTOR STATE EQUATIONS OF CELLULAR NEURAL NETWORKS

by

Tao Yang and Leon O. Chua

Memorandum No. UCB/ERL M98/42

29 June 1998

THE CRITERIA FOR SYMMETRIC VECTOR STATE EQUATIONS OF CELLULAR NEURAL NETWORKS

by

Tao Yang and Leon O. Chua

Memorandum No. UCB/ERL M98/42

29 June 1998

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

The Criteria for Symmetric Vector State Equations of Cellular Neural Networks

Tao Yang, *Member, IEEE* and Leon O. Chua, *Fellow, IEEE*
Electronics Research Laboratory and
Department of Electrical Engineering and Computer Sciences,
University of California at Berkeley,
Berkeley, CA 94720, USA

June 29, 1998

Abstract

In this paper we present criteria for checking the symmetry of the feedback matrix of the CNN vector state equations under different packing schemes, boundary conditions and weights. Examples of a 3×3 CNN are given for illustrating theoretic results. Since the symmetry of the feedback matrix of a CNN vector state equation is important for stability, applications of our results for checking the reliability of CNN chips are presented. Theoretical results on the complete stability of CNNs whose states are constrained by a state mask are also given as another application of our results.

1 Introduction

There are different ways to write the state equation of a cellular neural network(CNN). The most common way is a 2-D convolution form whose kernels are called *templates*[2, 3]. Although the template form can be easily used to describe the functional property of a CNN, the difficulty of studying the stability of the corresponding nonlinear *matrix* ODE makes it very difficult to study the stability of the CNN.

To study stability, it is much more convenient to write the state equation of a CNN into a *vector* equation because there exist many mathematical tools for studying the stability of nonlinear vector ODEs. Some important stability results on CNN had been presented under the condition that the vector state equation is endowed with some symmetry properties[1]. Since the symmetric property of a CNN vector state equation is critical to its stability, it is desirable to provide some conditions under which a CNN vector state equation is symmetric.

There are many conditions which affects the symmetry of a CNN vector state equation; namely, templates, shapes of CNN arrays, boundary conditions and packing schemes. In this paper, we present some theoretical results which guarantee the symmetry property of CNN vector state equations under all of the above conditions. Applications of our theoretical results to the complete stability of a CNN array with defective cells and state mask are provided.

The organization of this paper is as follows. In section 2, some basic notions, definitions and conditions are presented. In section 3, theoretical results for checking the symmetry property of

CNN vector state equations are given. In section 4, some examples of a 3×3 CNN are used to illustrate the theoretical results. In section 5, applications of the theoretical results to the reliability of CNN and the complete stability of state mask techniques are given. In section 6, some concluding remarks are given.

2 Definitions and conditions

To make this paper self-contained, we present some basic notations, definitions and equations of CNN in this section. Most of them are adopted from [1].

Definition 1. Sphere of influence

Each CNN cell C_{ij} is, by definition, coupled locally only to those neighbor cells which lie inside a prescribed *sphere of influence* $S_{ij}(r)$ of *radius* r, where

$$S_{ij}(r) = \{C_{kl} : \max(|k-i|, |l-j|) \le r, \ 1 \le k \le M, \ 1 \le l \le N\}$$

A standard CNN is described by the following equations originally proposed in [2, 3] for an $M \times N$ CNN array with M rows and N columns:

State equations of standard CNN:

$$\dot{x}_{ij} = -x_{ij} + z_{ij} + \sum_{kl \in S_{ij}(r)} a(i,j;k,l)\, y_{kl} + \sum_{kl \in S_{ij}(r)} b(i,j;k,l)\, u_{kl} \tag{1}$$

Output equation of standard CNN:

$$y_{kl} = f(x_{kl}) \tag{2}$$

$i = 1, 2, ..., M, j = 1, 2, ..., N$, where $f(\cdot)$ is usually defined by

$$f(x_{ij}) \triangleq \frac{1}{2}(|x_{ij}+1| - |x_{ij}-1|) = \begin{cases} 1, & x_{ij} \ge 1 \\ x_{ij}, & |x_{ij}| < 1 \\ -1, & x_{ij} \le -1 \end{cases} \tag{3}$$

For a space-invariant (homogeneous) CNN, the control (feed-forward) coefficients and feedback coefficients can be represented compactly by two templates $\boxed{B}$ and $\boxed{A}$. In this case, the CNN can be represented by the following 2-D convolution form[1]

[1]To be precise, $\star$ is the "correlation" operator. It is the same as the "convolution" operator when $\boxed{A}$ and $\boxed{B}$ are symmetric with respect to the center, respectively.

$$
\begin{aligned}
\dot{x}_{ij} &= -x_{ij} + z_{ij} + \boxed{A} \star y_{ij} + \boxed{B} \star u_{ij} \\
y_{ij} &= f(x_{ij}) = \frac{1}{2}(|x_{ij}+1| - |x_{ij}-1|) \\
& \quad i = 1,2,...,M; j = 1,2,...,N
\end{aligned} \tag{4}
$$

where "$\star$" denotes a 2-D correlation operator.

For a CNN which has a sphere of influence $S_{ij}(r)$, the templates $\boxed{B}$ and $\boxed{A}$ are $(2r+1) \times (2r+1)$ matrices. We label each entry in $\boxed{A}$ as $\boxed{a}(p,q)$ where $-r \leq p, q \leq r$. The center of $\boxed{A}$ is located at $(p,q) = (0,0)$ and the relationship between $\boxed{a}(p,q)$ and $a(i,j;k,l)$ is given by

$$
a(i,j;k,l) = \boxed{a}(k-i, l-j), \quad i,k = 1,2,...,M; j,l = 1,2,...,N \tag{5}
$$

Observe that Eq.(1) is *not* completely defined for cells whose sphere of influence $S_{ij}(r)$ extends outside of the boundary of the array. Consequently, additional *boundary conditions* must be specified in order for Eq. (1) to be well defined. The 3 most commonly chosen boundary conditions are:

1. *Fixed (Dirichlet) boundary condition*

Here, the state x_{kl} of each cell C_{kl} in Eq. (1) which lies outside of the boundary is assigned a fixed constant value.

2. *Zero flux(Neumann) boundary condition*

Here, the states x_{kl} of corresponding neighbor cells perpendicular to the boundaries are constrained to be equal to each other.

3. *Periodic(Toroidal) boundary condition*

Here, the first and last rows(resp., columns) of the array are identified, thereby forming a torus.

In this paper, we call cells which implement boundary conditions as *boundary cells*(also known as "virtual cells").

In order to apply the theory of dynamical systems, which has been developed for *vector* differential equations[5, 4], let us repack the state equation of an $M \times N$ CNN into a $\nu \times 1$ *vector* differential equation:

$$
\dot{\mathbf{X}} = -\mathbf{X} + \mathbf{Z} + \mathbf{AY} + \mathbf{BU} \tag{6}
$$

where $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{U}$, $\mathbf{Z} \in \Re^{\nu}$, $\mathbf{A}, \mathbf{B} \in \Re^{\nu} \times \Re^{\nu}$, $\nu = MN$. We henceforth refer to $\mathbf{A}$ as the *feedback matrix*.

We call the procedure in which we repack a CNN as a *packing scheme* and denote it by $\mathcal{P}$ in this paper. To define $\mathcal{P}$ formally, we need the following definition.

Definition 2: Packing scheme $\mathcal{P}$

A packing scheme $\mathcal{P}$ repacks the states x_{ij}, $1 \leq i \leq M, 1 \leq j \leq N$, of all cells in an $M \times N$ CNN into a 1-D vector $\mathbf{x}$. By defining a *packing map* $p : \mathbf{Z}^2 \mapsto \mathbf{Z}$, $\mathcal{P}$ maps cell C_{ij} to cell $C_{p(i,j)}$. $\mathcal{P}$ also maps x_{ij} and $a(i,j;k,l)$ to $x_{p(i,j)}$ and $a_{p(i,j),p(k,l)}$, respectively.

We then give some examples to illustrate the concept of packing scheme. Let us consider the following 4×4 CNN:

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4)

Example 1. Row-wise packing scheme $\mathcal{P}_R$

There are many possible row-wise packing schemes. The mostly common ones are shown in Fig.1. The packing map of the packing scheme shown in Fig.1(a) for the 4 × 4 CNN is given by

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4)

$\longrightarrow$

(i,j)	$p_R(i,j)$
(1,1)	1
(1,2)	2
(1,3)	3
(1,4)	4
(2,1)	5
(2,2)	6
(2,3)	7
(2,4)	8
(3,1)	9
(3,2)	10
(3,3)	11
(3,4)	12
(4,1)	13
(4,2)	14
(4,3)	15
(4,4)	16

In this paper we choose the above row-wise packing scheme as the standard one. For an $M \times N$ CNN, the above row-wise packing scheme has a very simple packing map given by

$$p_R = (i-1) \times M + j, \; 1 \leq i \leq M; 1 \leq j \leq N \tag{7}$$

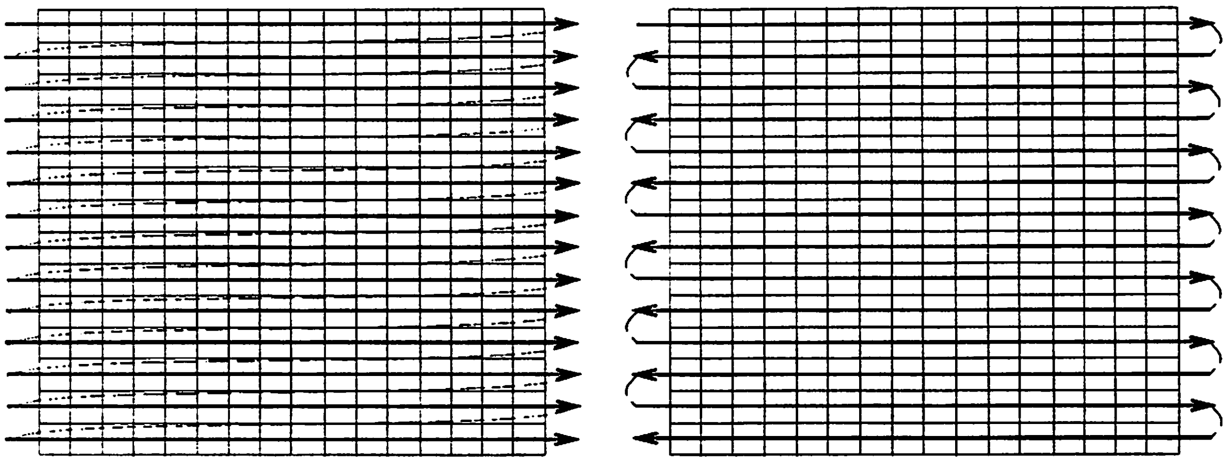
The packing map of the packing scheme shown in Fig.1(b) is given by

$$p_R = \begin{cases} (i-1) \times M + j, & \text{if } i \bmod 2{=}1 \\ (i-1) \times M + N - j + 1, & \text{if } i \bmod 2{=}0 \end{cases} \quad 1 \leq i \leq M; 1 \leq j \leq N \tag{8}$$

Observe that this is a piecewise linear function.

Example 2. Column-wise packing scheme $\mathcal{P}_C$

Similar to the row-wise packing schemes, the standard column-wise packing scheme for the 4 × 4 CNN is given by



(a) (b)

Figure 1: Two different row-wise packing schemes. (a) The standard row-wise packing scheme used in this paper. (b) Another common row-wise packing scheme.

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4)

$\longrightarrow$

(i,j)	$p_C(i,j)$
(1,1)	1
(2,1)	2
(3,1)	3
(4,1)	4
(1,2)	5
(2,2)	6
(3,2)	7
(4,2)	8
(1,3)	9
(2,3)	10
(3,3)	11
(4,3)	12
(1,4)	13
(2,4)	14
(3,4)	15
(4,4)	16

The corresponding packing map can be written explicitly as

$$p_C = (j-1) \times N + i, \;\; 1 \leq i \leq M; 1 \leq j \leq N \tag{9}$$

Example 3. Diagonal packing scheme $\mathcal{P}_D$

Among many choices, the diagonal packing scheme shown in Fig.2 is chosen as the standard one. In this case, the packing map for the 4×4 CNN is given by

(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	(4,2)	(4,3)	(4,4)

$\longrightarrow$

(i,j)	$p_D(i,j)$
(1,1)	1
(1,2)	2
(2,1)	3
(3,1)	4
(2,2)	5
(1,3)	6
(1,4)	7
(2,3)	8
(3,2)	9
(4,1)	10
(4,2)	11
(3,3)	12
(2,4)	13
(3,4)	14
(4,3)	15
(4,4)	16

This is a nonlinear packing scheme.

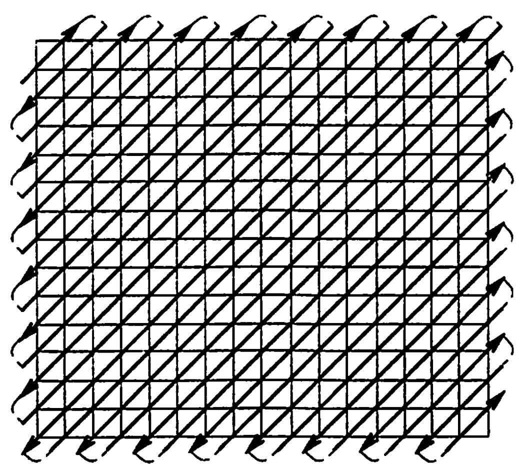


Figure 2: The standard diagonal packing scheme used in this paper.

Observe that the $\nu \times \nu$ matrices **A** and **B** in Eq.(6) are *sparse matrices* whose non-zero elements corresponds to the nonzero entries from the $\boxed{A}$ and $\boxed{B}$ templates, respectively.

3 Theoretical results

In this section, we present theoretical results which depend on the symmetry of the feedback template $\boxed{A}$. In these results we consider different packing schemes, shapes of CNN arrays and boundary conditions.

Theorem 1. *If* $\boxed{A}$ *is symmetric with respect to its center and the packing scheme* $\mathcal{P}$ *is a bijective map, then* $\mathbf{A}$ *is symmetric.*

Proof. Since $\boxed{A}$ is symmetric with respect to its center, we have $\boxed{a}(p,q) = \boxed{a}(-p,-q)$, from which we have $a(i,j;k,l) = \boxed{a}(k-i, l-j) = \boxed{a}(i-k, j-l) = a(k,l;i,j)$, $1 \le i,k \le M, 1 \le j,l \le N$. Since $\mathcal{P}$ is a bijective map, we know that $p(i,j)$ and $p(k,l)$ exist and are distinct. Hence, we have $a_{p(i,j),p(k,l)} = a(i,j;k,l) = a(k,l;i,j) = a_{p(k,l),p(i,j)}$ which leads to $\mathbf{A}^T = \mathbf{A}$. □

Remarks:

1. Theorem 1 is very general because it is satisfied by many distinct packing schemes. Observe from Theorem 1 that even a random but *bijective* packing scheme will result in a symmetric $\mathbf{A}$ matrix.

2. Theorem 1 does not depend on the boundary condition. Theorem 1 is also independent of the shapes of the CNN array. This is a very important result because it guarantees that the stability of a CNN can not be destroyed even if part of its cells had failed during operation.

3. If a packing scheme is not bijective then at least the state of one cell is packed more than once in the vector $\mathbf{x}$, or at least the state of one cell is not packed in the vector $\mathbf{x}$.

4. The examples in Section 4 show that $\boxed{A}^T = \boxed{A}$ does *not* imply the symmetry of $\mathbf{A}$. On the contrary, even though $\boxed{A}^T \neq \boxed{A}$, we get a symmetric $\mathbf{A}$ if $\boxed{A}$ is symmetric with respect to its center.

The most commonly used packing schemes are linear ones, e.g., row-wise or column-wise. The following theorem gives the criterion for linear packing schemes.

Theorem 2: *If a packing scheme* $\mathcal{P}$ *is linear with a packing map*

$$p(i,j) = \alpha i + \beta j + \gamma \tag{10}$$

and $\boxed{A}$ *is symmetric with respect to its center, then* $\mathbf{A}$ *is symmetric.*

Proof: Since $\boxed{A}$ is symmetric with respect to its center and $p(i,j)$ in Eq.(10) is a bijective map, it follows from Theorem 1 that $\mathbf{A}$ is symmetric. □

Corollary 1: *If* $\boxed{A}$ *is symmetric with respect to its center and the packing scheme is either row-wise or column-wise, then* $\mathbf{A}$ *is symmetric.*

Proof:

1. For a row-wise packing scheme $\mathcal{P}_R$ and an $M \times N$ CNN, the packing map is a linear transformation defined by $p_R(i,j) = (i-1) \times M + j$. It follows from Theorem 2 that $\mathbf{A}$ is symmetric.

2. For a column-wise packing scheme $\mathcal{P}_C$ and an $M \times N$ CNN, the packing map is a linear transformation defined by $p_C(i,j) = (j-1) \times N + i$. It follows from Theorem 2 that $\mathbf{A}$ is symmetric. □

Corollary 2: *If* $\boxed{A}$ *is symmetric with respect to its center and a diagonal packing scheme* $\mathcal{P}_D$ *is used, then* $\mathbf{A}$ *is symmetric.*

Proof. Since $\mathcal{P}_D$ is a bijective packing scheme, the above corollary follows immediately from Theorem 1. □

4 Examples

In this section we present some examples based on a 3×3 CNN with a sphere of influence $S_{ij}(1)$ to demonstrate the theoretical results presented in the previous section.

Consider an $\boxed{A}$ template in $S_{ij}(1)$

$$\boxed{A} = \begin{array}{|c|c|c|} \hline a & d & f \\ \hline d & b & e \\ \hline f & e & c \\ \hline \end{array} \tag{11}$$

If $a \neq c$ or $d \neq e$ then $\boxed{A}$ is symmetric with respect to the diagonal a-b-c, but is asymmetric with respect to its center. Consider first a 3×3 CNN with a toroidal boundary condition, and a row-wise packing scheme $\mathcal{P}_R$. The CNN and its boundary condition is depicted as

C_{33}	C_{31}	C_{32}	C_{33}	C_{31}
C_{13}	$\mathbf{C_{11}}$	$\mathbf{C_{12}}$	$\mathbf{C_{13}}$	C_{11}
C_{23}	$\mathbf{C_{21}}$	$\mathbf{C_{22}}$	$\mathbf{C_{23}}$	C_{21}
C_{33}	$\mathbf{C_{31}}$	$\mathbf{C_{32}}$	$\mathbf{C_{33}}$	C_{31}
C_{13}	C_{11}	C_{12}	C_{13}	C_{11}

where bold type characters denote normal cells and normal type characters denote boundary cells.

In this case, $\mathbf{A}$ is a 9×9 matrix given by

$$\mathbf{A} = \begin{pmatrix} b & e & d & e & c & f & d & f & a \\ d & b & e & f & e & c & a & d & f \\ e & d & b & c & f & e & f & a & d \\ d & f & a & b & e & d & e & c & f \\ a & d & f & d & b & e & f & e & c \\ f & a & d & e & d & b & c & f & e \\ e & c & f & d & f & a & b & e & d \\ f & e & c & a & d & f & d & b & e \\ c & f & e & f & a & d & e & d & b \end{pmatrix}$$

Observe that $\mathbf{A}$ is a symmetric matrix if $d = e$ and $a = c$. Observe also that the symmetry of $\boxed{A}$ with respect to the diagonal a-b-c does not imply the symmetry of $\mathbf{A}$.

To show that the symmetry of $\mathbf{A}$ is independent of the boundary conditions, let us rewrite the matrix **A** under a fixed (Dirichlet) boundary condition

$$\boxed{\mathrm{A}} = \begin{array}{|c|c|c|} \hline a & d & f \\ \hline d & b & e \\ \hline f & e & c \\ \hline \end{array} \longrightarrow \quad \mathbf{A} = \begin{pmatrix} b & e & 0 & e & c & 0 & 0 & 0 & 0 \\ d & b & e & f & e & c & 0 & 0 & 0 \\ 0 & d & b & 0 & f & e & 0 & 0 & 0 \\ d & f & 0 & b & e & 0 & e & c & 0 \\ a & d & f & d & b & e & f & e & c \\ 0 & a & d & 0 & d & b & 0 & f & e \\ 0 & 0 & 0 & d & f & 0 & b & e & 0 \\ 0 & 0 & 0 & a & d & f & d & b & e \\ 0 & 0 & 0 & 0 & a & d & 0 & d & b \end{pmatrix}$$

Observe that **A** is a symmetric matrix if $d = e$ and $a = c$.

To show that an asymmetric $\boxed{\mathrm{A}}$ can also generate a symmetric **A** matrix, consider the following $\boxed{\mathrm{A}}$ template

$$\boxed{\mathrm{A}} = \begin{array}{|c|c|c|} \hline a & c & e \\ \hline d & b & d \\ \hline e & c & a \\ \hline \end{array} \tag{12}$$

with a sphere of influence $S_{ij}(1)$. This template is symmetric with respect to the center but not symmetric with respect to the diagonal line. The **A** matrix of the same 3×3 CNN with a toroidal boundary condition and a row-wise packing scheme $\mathcal{P}_R$ is given by the following matrix

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

Observe that **A** is a symmetric matrix.

5 Applications

In this section, we apply the results from Section 3 to demonstrate the reliability of CNN chips with static input images. We also present a stability criterion for state mask techniques. To make this paper self-contained, the following theorem is reproduced from [1].

Theorem 3.[Theorem 2.3.2 of [1]] Complete Stability Criterion

All trajectories of the standard CNN defined by Eqs.(1)-(2) with constant thresholds, constant inputs, and a sphere of influence of arbitrary size, converge to an equilibrium state, which in general depends on the initial states, if the following 3 hypotheses are satisfied:

(i) The matrix **A** *in Eq.(6) is symmetric.*

(ii) The scalar function $f(\cdot)$ in the output equation (2) is differentiable with positive slopes, and is bounded.

(iii) All equilibrium points are isolated.

In this paper we say a cell is damaged, or has failed, when this cell is disconnected from the other cells, i.e., all weights connected between this cell and the neighbor cells are zero. Damage in a small part of the retina does not make a person blind shows that the partial damage does not destroy the stability of the human retina. When a CNN is used with constant input images, we want it to remain stable even if some of its cells had failed. This problem is closely related to the reliability of a CNN chip. Since the damaged cells may change the symmetry of the CNN array and the boundary condition, it is important to investigate 1) the relation between stability and the number of damaged cells, and 2) the relation between stability and the locations of the damaged cells. Our next theorem provides some theoretical results on these relationships.

Theorem 4. Complete stability of CNN after damage

All trajectories of the standard CNN defined by Eqs.(1)-(2) with constant thresholds, constant inputs, and an arbitrary number of damaged cells at random positions, converge to an equilibrium state, if the same hypotheses as Theorem 3 are satisfied; namely,

1. $\boxed{A}$ *is symmetric with respect to its center*

2. The scalar function $f(\cdot)$ in the output equation (2) is differentiable with positive slopes, and is bounded.

3. All equilibrium points are isolated.

Proof. Suppose that all the cells in the CNN is lumped into two groups. Group 1, $\{C_{ij}^{(1)}\}$, consists of all cells which are still functioning. Group 2, $\{C_{ij}^{(2)}\}$, consists of all cells which had failed. Suppose that a cell $C_{ij}^{(2)}$ belongs to Group 2, by the definition of a damaged cell we know that

$$a(i,j;k,l) = a(k,l;i,j) = 0, \quad i,j \neq k,l \tag{13}$$

Since cell $C_{ij}^{(2)}$ had failed, the weight $a(i,j;i,j)$ and $b(i,j;i,j)$ may be two arbitrary (space-varying) values denoted by $a_{00}^{(2)}(i,j)$ and $b_{00}^{(2)}(i,j)$, respectively. Since $\{C_{ij}^{(2)}\}$ is not coupled to the other cells, its state equation is given by

$$\dot{x}_{ij} = -x_{ij} + z_{ij} + a_{00}^{(2)}(i,j)y_{ij} + b_{00}^{(2)}(i,j)u_{ij}, \quad C_{ij} \in \{C_{ij}^{(2)}\} \tag{14}$$

We choose such a bijective packing scheme $\mathcal{P}$ that the state of the entire CNN is given by the vector

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \end{pmatrix} \tag{15}$$

where $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ pack all states of cells in Group 1 and Group 2, respectively. Then the vector state equation in Eq.(6) assumes the form

$$\begin{pmatrix} \dot{\mathbf{X}}^{(1)} \\ \dot{\mathbf{X}}^{(2)} \end{pmatrix} = -\begin{pmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \end{pmatrix} + \begin{pmatrix} \mathbf{Z}^{(1)} \\ \mathbf{Z}^{(2)} \end{pmatrix} + \begin{pmatrix} \mathbf{A}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^{(2)} \end{pmatrix} \begin{pmatrix} \mathbf{Y}^{(1)} \\ \mathbf{Y}^{(2)} \end{pmatrix} + \begin{pmatrix} \mathbf{B}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}^{(2)} \end{pmatrix} \begin{pmatrix} \mathbf{U}^{(1)} \\ \mathbf{U}^{(2)} \end{pmatrix} \tag{16}$$

Let λ be the number of the damaged cells, then $\nu = MN - \lambda$ is the number of the normal cells. $\mathbf{X}^{(1)}$, $\mathbf{Y}^{(1)}$, $\mathbf{U}^{(1)}$, $\mathbf{Z}^{(1)} \in \Re^{\nu}$, are state vector, output vector, input vector and threshold vector of the cells in Group 1, respectively. $\mathbf{A}^{(1)}, \mathbf{B}^{(1)} \in \Re^{\nu} \times \Re^{\nu}$, are feedback and feedforward matrices for all the cells in Group 1, respectively. $\mathbf{X}^{(2)}$, $\mathbf{Y}^{(2)}$, $\mathbf{U}^{(2)}$, $\mathbf{Z}^{(2)} \in \Re^{\lambda}$ are state vector, output vector, input vector and threshold vector of the cells in Group 2, respectively. $\mathbf{A}^{(2)}, \mathbf{B}^{(2)} \in \Re^{\lambda} \times \Re^{\lambda}$ are feedback and feedforward matrices for all cells in Group 2, respectively. Observe that the cells in Group 1 are not coupled to the cells in Group 2.

From Eq.(14) we know that $\mathbf{A}^{(2)}$ is a diagonal matrix. Since $\boxed{A}$ is symmetric with respect to its center, it follows from Theorem 1 that $\mathbf{A}^{(1)}$ is symmetric. It is easy to see that the $\mathbf{A}$ matrix given by

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^{(2)} \end{pmatrix} \tag{17}$$

is symmetric. It follows from Theorem 3 that the damaged CNN is still completely stable. □

Remark: Although Theorem 4 concludes that the damaged CNN is still completely stable, it does not guarantee that the damaged CNN will function as originally designed. Theorem 4 only guarantees that the damaged CNN can function partially if a partial damage occurs.

In some applications such as CNN interpolation[6] a special technique called *state mask* is widely used. A state mask consists of a subset of cells, $\{C_{ij}^{(M)}\}$, called *mask cells*. The states of the mask cells are fixed during the time evolution of the CNN. Although some applications of CNNs with state masks have worked correctly, so far there exist no theoretical results for explaining why this happens, in spite of the fact that the existence of mask cells changes the symmetry of the original CNN array, thus potentially changing the stability of the CNN. It is important therefore to investigate 1) the relation between stability and the number of mask cells, and 2) the relation between stability and the locations of the mask cells. Our next theorem provides some theoretical results on these relationships.

Theorem 5 Complete stability of CNN with state mask

All trajectories of the standard CNN defined by Eqs.(1)-(2) with constant thresholds, constant inputs, and an arbitrary number of mask cells at random positions, converge to an equilibrium state, if the same hypotheses in Theorem 3 are satisfied; namely,

1. $\boxed{A}$ *is symmetric with respect to its center*

2. The scalar function $f(\cdot)$ in the output equation (2) is differentiable with positive slopes, and is bounded.

3. All equilibrium points are isolated.

Proof. Suppose that all cells in the CNN is lumped into two groups. Group 1, $\{C_{ij}^{(1)}\}$, consists of all cells which are not mask cells. Group 2, $\{C_{ij}^{(M)}\}$, consists of all cells which are mask cells. Since the state of a mask cell $C_{ij}^{(M)}$ keeps its initial value $x_{ij}(0)$ all the time, its dynamics is given by

$$\dot{x}_{ij} = 0, \quad x_{ij}(0). \tag{18}$$

We choose such a bijective packing scheme $\mathcal{P}$ that the state of the entire CNN is given by the state vector

$$\mathbf{X} = \begin{pmatrix} \mathbf{X^{(1)}} \\ \mathbf{X^{(M)}} \end{pmatrix} \tag{19}$$

where $\mathbf{X^{(1)}}$ and $\mathbf{X^{(M)}}$ pack the states of all cells in Group 1 and Group 2, respectively. Then the vector state equation in Eq.(6) assumes the form

$$\begin{aligned} \begin{pmatrix} \dot{\mathbf{X}}^{(1)} \\ \dot{\mathbf{X}}^{(M)} \end{pmatrix} &= -\begin{pmatrix} \mathbf{X^{(1)}} \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{Z^{(1)}} \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{A^{(1)}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{Y^{(1)}} \\ \mathbf{0} \end{pmatrix} \\ &+ \begin{pmatrix} \mathbf{B^{(11)}} & \mathbf{B^{(1M)}} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U^{(1)}} \\ \mathbf{U^{(M)}} \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \mathbf{A^{(1M)}} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ \mathbf{Y^{(M)}(0)} \end{pmatrix} \end{aligned} \tag{20}$$

Let λ be the number of the mask cells, then $\nu = MN - \lambda$ is the number of the normal cells. $\mathbf{X^{(1)}}$, $\mathbf{Y^{(1)}}$, $\mathbf{U^{(1)}}$, $\mathbf{Z^{(1)}} \in \Re^{\nu}$ are state vector, output vector, input vector and threshold vector of the cells in Group 1, respectively. $\mathbf{A^{(1)}}, \mathbf{B^{(11)}} \in \Re^{\nu} \times \Re^{\nu}$, are feedback and feedforward matrices for all cells in Group 1, respectively. $\mathbf{X^{(M)}}$ and $\mathbf{U^{(M)}} \in \Re^{\lambda}$ are state vector and input vector of the cells in Group 2, respectively. $\mathbf{B^{(1M)}} \in \Re^{\nu} \times \Re^{\lambda}$ is the feedforward matrix which denotes the feedforward effect of the cells in Group 2 on the cells in Group 1. $\mathbf{Y^{(M)}(0)} \in \Re^{\lambda}$ is the fixed output vector of the cells in Group 2. $\mathbf{A^{(1M)}} \in \Re^{\nu} \times \Re^{\lambda}$ is the feedback matrix which denotes the feedback effect of the cells in Group 2 on the cells in Group 1. Since $\mathbf{Y^{(M)}(0)}$ is unchanged it can be viewed as a virtual input In this case the input of the CNN is given by

$$\mathbf{U} = \begin{pmatrix} \mathbf{B^{(11)}} & \mathbf{B^{(1M)}} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U^{(1)}} \\ \mathbf{U^{(M)}} \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \mathbf{A^{(1M)}} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ \mathbf{Y^{(M)}(0)} \end{pmatrix}. \tag{21}$$

The threshold vector of the CNN is given by

$$\mathbf{Z} = \begin{pmatrix} \mathbf{Z^{(1)}} \\ \mathbf{0} \end{pmatrix}. \tag{22}$$

Observe that $\mathbf{U}$ and $\mathbf{Z}$ are still constant vectors. Since $\dot{\mathbf{X}}^{(M)} = \mathbf{0}$ is always satisfied, the vector state equation in Eq.(20) reduces to

$$\dot{\mathbf{X}}^{(1)} = -\mathbf{X^{(1)}} + \mathbf{Z^{(1)}} + \mathbf{A^{(1)}Y^{(1)}} + [\mathbf{B^{(11)}U^{(1)}} + \mathbf{B^{(1M)}U^{(M)}} + \mathbf{A^{(1M)}Y^{(M)}(0)}] \tag{23}$$

Since $\boxed{A}$ is symmetric with respect to its center, it follows from Theorem 1 that $\mathbf{A^{(1)}}$ is symmetric. It follows from Theorem 3 that the cells in Group 1 are completely stable. □

6 Concluding Remarks

In this paper we give criteria for the symmetry of CNN *vector* state equations. We also show that the symmetry of the feedback matrix in the vector state equation is independent of any bijective packing schemes. Applications of our results give conditions under which a CNN which is subject to uncontrollable and unpredictable damages (such as in the vision system of an autonomous robot), can still function. By applying our theorems we also give theoretical results on the complete stability of CNNs found in many practical applications with state masks.

Acknowledgment

This work is supported in part by a NATO Linkage grant No. OUTR.LG 960578, and by the Office of Naval Research under grant No. N00014-96-1-0753.

References

[1] L.O. Chua. CNN: A vision for complexity. *International Journal of Bifurcation and Chaos*, vol.7, no.10, Oct.1997(in press).

[2] L.O. Chua and L. Yang. Cellular neural networks: Theory. *IEEE Transactions on Circuits and Systems*, 35(10):1257-1272, Oct.1988.

[3] L.O. Chua and L. Yang. Cellular neural networks: Applications. *IEEE Transactions on Circuits and Systems*, 35(10):1273-1290, Oct.1988.

[4] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. Springer-Verlag, New York, 1983.

[5] M. W. Hirsch and S. Smale. *Differential Equations, Dynamical Systems, and Linear Algebra*. Academic Press, New York, 1974.

[6] T. Roska, L. Kek, L. Nemes, and A. Zarandy. CNN software library (templates and algorithms). Technical Report version 7.0 (DNS-1-1997), Analogical and Neural Computing Laboratory, Computer Automation Institute, Hungarian Academy of Sciences, Budapest, Hungary, 1997.