Competitive Layer Model of Discrete-Time Recurrent Neural Networks with LT Neurons

Wei Zhou
zhouwei@uestc.edu.cn
School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 610054, P.R.C.

Jacek M. Zurada
jmzura02@louisville.edu
Computational Intelligence Laboratory, Electrical and Computer Engineering Department, University of Louisville, Louisville, KY, 40292 U.S.A.

This letter discusses the competitive layer model (CLM) for a class of discrete-time recurrent neural networks with linear threshold (LT) neurons. It first addresses the boundedness, global attractivity, and complete stability of the networks. Two theorems are then presented for the networks to have CLM property. We also present the analysis for network dynamics, which performs a column winner-take-all behavior and grouping selection among different layers. Furthermore, we propose a novel synchronous CLM iteration method, which has similar performance and storage allocation but faster convergence compared with the previous asynchronous CLM iteration method (Wersing, Steil, & Ritter, 2001). Examples and simulation results are used to illustrate the developed theory, the comparison between two CLM iteration methods, and the application in image segmentation.

1 Introduction

Perceptual grouping, which can be defined as the ability to detect structural layout of visual objects by human being, was first studied in the 1920s by the Gestalt school of psychology (Koffka, 1962). One of its important theories is the Gestalt law, and some Gestalt laws, like proximity, symmetry, and continuity, were used to explain how humans detect groups in a set of objects. In computer vision, this grouping mechanism can be considered a procedure for feature binding, which aims at binding some related features into common groups, so as to separate those groups originating from different features (von der Malsburg, 1981, 1995).

The competitive layer model (CLM) was first advocated as a model for spatial feature binding by Ritter (1990; Ontrup & Ritter, 1998). This model is based on a combination of competitive and cooperative processes in a recurrent neural network architecture, which can partition a set of input features into salient groups. Due to competitive interactions among layers, each feature is unambiguously assigned to one layer, and feature binding is achieved by a collection of competitive layers. Wersing, Steil, and Ritter (2001) designed a continuous-time CLM neural network with linear threshold (LT) neurons for feature binding and sensory segmentation. Weng, Wersing, Steil, and Ritter (2006) proposed a hybrid learning method based on CLM.

In recent years, many researchers have studied the recurrent neural networks with LT neurons (Hahnloser, Sarpeshkar, Mahowald, Douglas, & Seung, 2000; Tang, Tan, & Zhang, 2005). One important property of LT networks is their multistability (Hahnloser, 1998; Yi, Tan, & Lee, 2003; Yi & Tan, 2004; Zhang, Yi, & Yu, 2008; Zhang, Yi, Zhang, & Heng, 2009). The traditional winner-take-all (WTA) neural networks are almost monostable, and only one neuron among all neurons can be the final "winner". The multistability property provides an interesting way to mediate the WTA competition between groups of neurons, and the final winner will be a group of neurons. Hahnloser, Sebastian, and Slotine (2003) denoted the group winner as a permitted set (more details about permitted set can been found in Xie, Hahnloser, & Seung, 2002).

Compared with continuous-time neural networks, discrete-time neural networks have some advantages for direct computer simulations and implementation in digital hardware. Unfortunately, the analysis of continuous-time recurrent neural networks is not always applicable to the discrete version. Thus, the detailed analysis of discrete version is necessary and important. To the best of our knowledge, almost all studies of the discrete-time neural network models have focused on the behavior of monostable networks (Jin, Nikiforuk, & Gupta, 1994; Si & Michel, 1995; Hu & Wang, 2002; Wang & Xu, 2006), and only a small amount of work has been done on multistable networks (Yi & Tan, 2004; Yi, Heng, & Fung, 2000; Zhang et al., 2009). Furthermore, little attention has been paid to WTA competition between groups of neurons, so our work can be seen as an attempt to cope with it.

In this letter, we propose a class of discrete-time recurrent neural networks with LT neurons based on competitive layer model (CLM-DT-LT-RNNs). In addition to deriving stability criteria, we prove that all stable attractors are potential group winners. We also present some dynamic properties analysis. Based on the convergence theory noted in Feng (1997), an asynchronous CLM (ACLM) iteration method was used in Wersing et al. (2001) and Weng et al. (2006). Because the network updates only one neuron each time, this makes the iterations time-consuming, especially for a large-scale network. In this letter, we propose a novel synchronous CLM (SCLM) iteration method. Compared with the ACLM iteration method, our method has similar performance and storage allocation, but its simulations are faster.

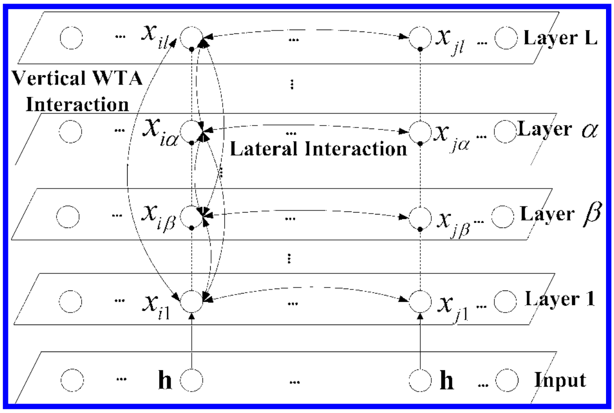


Figure 1: The competitive layer model architecture.

The rest of this letter is organized as follows. The architecture of the proposed discrete-time CLM networks is described in section 2. Preliminaries are given in section 3. In section 4, a theoretical analysis of such networks is given, which explores: boundedness, attractivity, complete stability, equilibrium properties, dynamic properties analysis, and comparison between the SCLM method and the ACLM method. Simulations and illustrative examples are presented in section 5. Conclusions are given in section 6.

2 Competitive Layer Model of Discrete-Time Recurrent Neural Networks with LT Neurons

In this letter, we study a class of discrete-time recurrent neural networks with LT neurons based on the competitive layer model (CLM-DT-LT-RNNs), which is described as

$$x_{i\alpha}(k+1) = \frac{1}{C}\left[hJ - J\sum_{\beta=1}^{l}\sigma(x_{i\beta}(k)) + \sum_{j=1}^{n} f_{ij}\sigma(x_{j\alpha}(k))\right] + \sigma(x_{i\alpha}(k)) \tag{2.1}$$

for $k \geq 0$, $i = 1, \ldots, n$, and $\alpha = 1, \ldots, l$. The CLM, equation 2.1, consists of a set of l layers of feature-selective neurons, and each layer contains n neurons (see Figure 1). J and f_{ij} are the vertical WTA interaction and lateral interaction, respectively. Here, $f_{ij} = f_{ji}$, for $i, j = 1, \ldots, n$. $x_{i\alpha}$ is the activity of a neuron at position i in layer α, and a column i denotes the set of neuron activities $x_{i\alpha}, \alpha = 1, \ldots, l$, that share a common position i in each layer. All neurons in a column i are equally driven by an external input h, and h is fed to the activities $x_{i\alpha}$ with a connection weight equal to J for simplicity. The LT function σ is defined as $\sigma(x) = \max(0, x)$, $x \in \Re$, which is continuous, unbounded, and nondifferentiable. For vector $x \in \Re^n$, we denote $\sigma(x) = (\sigma(x_1), \sigma(x_2), \ldots, \sigma(x_n))^T \in \Re^n$.

Now the equivalent vector form of network 2.1 can be defined as

$$x(k+1) = W\sigma(x(k)) + H \tag{2.2}$$

for $k \geq 0$, $x(k) = [x_{11}(k), x_{21}(k), \ldots, x_{n1}(k), \ldots, x_{1l}(k), x_{2l}(k), \ldots, x_{nl}(k)]^T$, and $W = (w_{ij})_{nl \times nl}$ is a real symmetric matrix. Each element w_{ij} denotes the synaptic weights and represents the strength of the synaptic connection from neuron i to neuron j. $H \in \Re^{nl}$ denotes external input, and $H_{i\alpha} = \frac{hJ}{C}$ for $i = 1, \ldots, n$ and $\alpha = 1, \ldots, l$.

Furthermore, we define

$$W = G + Id_{nl}, \tag{2.3}$$

$$G = \frac{1}{C}\left[f \otimes Id_l - J \cdot \Pi_l \otimes Id_n\right] = \frac{1}{C}F - \frac{J}{C}\mathrm{P}, \tag{2.4}$$

where $\otimes$ is a Kronecker product (Horn & Johnson, 1985), Id_m is the $m \times m$ identity matrix (here $m = nl$, n, or l), Π_l is an $l \times l$ matrix of 1's, $f = (f_{ij})_{nl \times nl}$, $F = f \otimes Id_l$, and $P = \Pi_l \otimes Id_n$.

Note that the CLM has two types of connections: the vertical interaction J and the lateral interaction f within layers. In real applications, f may be the proximity interaction used for clustering or the continuity interaction used for finding continuous curves (Wersing et al., 2001). In our image segmentation examples, we store the pixel-related information based on gray and position relationships in f. The purpose of the CLM architecture is to enforce a dynamical assignment of the input features to the layers by using the contextual information stored in f. This assignment can be considered feature binding. More discussion about these properties can be found in section 4.

3 Preliminaries

In this section, we provide preliminaries used to establish our theory.

Definition 1. *The network, equation 2.1, is said to be bounded if each trajectory is bounded.*

Definition 2. *Let S be a compact subset of equation 2.1. We denote the ϵ-neighborhood of S by S_ϵ. The compact set S is said to globally attract network 2.1 if for any $\epsilon \geq 0$, all trajectories of the equation ultimately enter and remain in S_ϵ. The set S is called an attractive set.*

Obviously, if every trajectory $x(k)$ of equation 2.1 satisfies

$$\begin{cases} \lim\limits_{k\to+\infty} \sup x(k) \in S \\ \lim\limits_{k\to+\infty} \inf x(k) \in S \end{cases},$$

then S globally attracts the network 2.1.

Definition 3. *A vector x^* is called an equilibrium point (fixed point) of equation 2.1 if it satisfies $x^* = W\sigma(x^*) + H$. Denote by Ω to the set of equilibrium points of equation 2.1.*

Definition 4. *An equilibrium point x^* is said to be stable (in the sense of Lyapunov) if the following statement is true: for every $\epsilon > 0$, there exists $\delta > 0$ such that every solution $x(k)$ with $\|x(0) - x^*\| < \delta$ exists for all $k \geq 0$ and satisfies the inequality $\|x(k) - x^*\| < \epsilon$ for $k \geq 0$. The norm $\|\cdot\|$ is an arbitrary norm in $\Re^n$ or C^n. And an equilibrium point x^* is called unstable if it is not stable.*

Definition 5. *Network 2.1 is said to be completely convergent (completely stable), if each trajectory $x(k)$ satisfies $dist(x(k), \Omega) \overset{\Delta}{=} min_{x^* \in \Omega} \|x(k) - x^*\| \to 0$ as $k \to +\infty$.*

Definition 6. *A neuron is said to be activated if its activity is positive. And a column of CLM is said to be activated if there exists at least one activated neuron in this column.*

Definition 7. *Let $A \in \Re^{m\times n}$. For index sets $\alpha \subseteq \{1, \ldots, m\}$ and $\beta \subseteq \{1, \ldots, n\}$, we denote the (sub)matrix that lies in the rows of A indexed by α and the columns indexed by β as $A(\alpha, \beta)$. If $m = n$ and $\beta = \alpha$, the submatrix $A(\alpha, \alpha)$ is called a principal submatrix of A and is abbreviated $A(\alpha)$.*

Lemma 1 (Yi & Tan, 2004). *Suppose network 2.1 is bounded. If there exists a diagonal positive-definite matrix D such that $D(I + W)$ is a symmetric positive-definite matrix, then network 2.1 is completely stable.*

Lemma 2 (Horn & Johnson, 1985). *Let $A \in \Re^{n\times n}$ be Hermitian and A' be a principal submatrix $A(n-1)$ of A, and let the eigenvalues $\{\lambda_i(A)\}$ of A, $\{\lambda_i(A(n-1))\}$ of $A(n-1)$ be arranged in an increasing order. Then the eigenvalues of A' interlace the eigenvalues of A: $\lambda_1(A) \leq \lambda_1(A(n-1)) \leq \lambda_2(A) \leq \cdots \leq \lambda_{n-1}(A(n-1)) \leq \lambda_n(A)$.*

Lemma 3 (Horn & Johnson, 1985). *Let A, $B \in \Re^{n\times n}$ be Hermitian, and let the eigenvalues $\{\lambda_i(A)\}$ of A, $\{\lambda_i(B)\}$ of B, and $\{\lambda_i(A + B)\}$ of A+B be arranged in increasing order. For each $k = 1, \ldots, n$, we have $\lambda_k(A) + \lambda_1(B) \leq \lambda_k(A + B) \leq \lambda_k(A) + \lambda_n(B)$.*

Throughout this letter, for all constant $c \in \Re$, denote $c^+ = \max(0, c), c^- = \min(0, c)$. For any matrix $A = (a_{ij})_{n\times m}$, we denote $A^+ = (a_{ij}^+)$, where $a_{ij}^+ = \max(0, a_{ij})$. We also note that f has eigenvalues $\lambda_i(f)$ or λ_i, $1 \le i \le n$, and $\lambda_{\min}(f)$, $\lambda_{\max}(f)$ are the minimal eigenvalue and the maximal eigenvalue of f, respectively.

Lemma 4. *Let G_Σ be a principal submatrix of G. Suppose f is a positive-definite matrix and $J > max_{1\le i\le n}\{\lambda_i\}$. Then G_Σ must have no zero eigenvalues.*

Since each LT neuron is either active or inactive, the whole ensemble of the LT neurons in the equilibrium state can be divided into two partitions with state $x_{i\alpha}^* > 0$ for $i\alpha \in P$ and $x_{i\alpha}^* \le 0$ for $i\alpha \in N$, where P and N are both index sets and $P \cup N = \{11, 12, ..., 1l, ..., n1, n2..., nl\}$.

4 Theoretical Analysis of CLM-DT-LT-RNNs

4.1 Properties of CLM-DT-LT-RNNs. In this section, conditions guaranteeing boundedness and complete stability of network 2.1 are presented in theorems 1 and 2, respectively. We also present a necessary condition in theorem 3 to let our networks have CLM phenomena. Furthermore, a sufficient and necessary condition for CLM phenomena is given in theorem 4.

Theorem 1. *If there exist constants J and C such that $\begin{cases} J > max_{1\le i\le n}\{\sum_{j=1}^n f_{ij}^+\} \\ C > J \end{cases}$, then network 2.1 is bounded, and the compact set $S = \{x \mid a \le x_{i\alpha} \le b, 1 \le i \le n, 1 \le \alpha \le l\}$ globally attracts network 2.1 where*

$$\begin{cases} b = \dfrac{1}{1-\gamma} \cdot \dfrac{hJ}{C} \\ a = \min\limits_{1\le i\le n}\left\{\dfrac{1}{C}\sum\limits_{j=1}^n f_{ij}^- \cdot b\right\} \\ \gamma = \max\limits_{1\le i\le n}\left\{\dfrac{1}{C}\left(C - J + \sum\limits_{j=1}^n f_{ij}^+\right)\right\}. \end{cases}.$$

Proof. By using the method in Yi and Tan (2004), the boundedness conditions and the compact set S can be easily obtained. The proof is completed.

Theorem 2. *Suppose network 2.1 is bounded and matrix f is symmetric. If there exist constants J and C such that $\begin{cases} J > max\{max_{1\le i\le n}\{\sum_{j=1}^n f_{ij}^+\}, \frac{1}{l} max_{1\le i\le n}\{|\lambda_i|\}\} \\ C > Jl \end{cases}$, then network 2.1 is completely convergent.*

Proof. From equations 2.3 and 2.4, an orthonormal eigenvector basis $\{V_{i\beta}, \Lambda_{i\beta}\}$ for W can be obtained from the orthonormal eigenvector bases $\{b_i, \lambda_i\}$ and $\{q^\beta, \mu_\beta\}$ for f and Π_l respectively (Wersing et al., 2001):

$$\begin{cases} V_{i1} = \dfrac{1}{\sqrt{l}}\left(b_i^T, \ldots, b_i^T\right)^T, \quad \Lambda_{i1} = \dfrac{1}{C}\left(\lambda_i - J\,l\right) + 1 \\ V_{i\beta\neq 1} = \left(\sqrt{\displaystyle\sum_{\alpha=1}^{l}\left(q_\alpha^{\beta\neq 1}\right)^2}\right)^{-\frac{1}{2}} \cdot \left(q_1^{\beta\neq 1} b_i^T, \ldots, q_l^{\beta\neq 1} b_i^T\right)^T, \\ \qquad\qquad\qquad\qquad \Lambda_{i\beta\neq 1} = \frac{1}{C}\lambda_i + 1 \end{cases}, \tag{4.1}$$

where $i = 1, \ldots, n$ and $\beta = 1, \ldots, l$.

Since $J > \frac{1}{l}\max_{1\leq i\leq n}\{|\lambda_i|\}$ and $C > J\,l$, by equation 4.1, we have

$$\begin{cases} -1 < \Lambda_{i1} < 1 \\ \Lambda_{i\beta\neq 1} > 0 \end{cases}. \tag{4.2}$$

Clearly it holds that $\Lambda_{i\beta} + 1 > 0$ for all i, β. By lemma 1, the network is convergent.

Theorem 3. *Suppose network 2.1 satisfies theorem 2. If the lateral interaction is self-excitatory, $f_{ii} > 0$ for all i, then a stable equilibrium of CLM in network 2.1 has in each column i at most one activated neuron $x_{i\alpha}^*$ with*

$$\sum_{j\alpha\in P} f_{ij} x_{j\alpha}^* \geq \sum_{j\beta\in P} f_{ij} x_{j\beta\neq\alpha}^*.$$

Proof. Now assume the contrary. Let x^* be a stable equilibrium with at least two neurons in a column i at two layers: $x_{i\alpha}^* > 0$ and $x_{i\beta}^* > 0$. Then at x^*, we have

$$\begin{cases} x_{i\alpha}^* = \dfrac{1}{C}\left[hJ - J\left(x_{i\alpha}^* + x_{i\beta}^*\right) + f_{ii}x_{i\alpha}^* - J\displaystyle\sum_{\gamma\neq\alpha,\beta}^{l}\sigma(x_{i\gamma}^*) + \sum_{j\neq i}^{n} f_{ij}\sigma(x_{j\alpha}^*)\right] + x_{i\alpha}^* \\ x_{i\beta}^* = \dfrac{1}{C}\left[hJ - J\left(x_{i\alpha}^* + x_{i\beta}^*\right) + f_{ii}x_{i\beta}^* - J\displaystyle\sum_{\gamma\neq\alpha,\beta}^{l}\sigma(x_{i\gamma}^*) + \sum_{j\neq i}^{n} f_{ij}\sigma(x_{j\beta}^*)\right] + x_{i\beta}^* \end{cases}.$$

The coefficient matrix of $x^*_{i\alpha}, x^*_{i\beta}$ is

$$H_{\alpha\beta} = \begin{bmatrix} -\frac{J-f_{ii}}{C} & -\frac{J}{C} \\ -\frac{J}{C} & -\frac{J-f_{ii}}{C} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = W_{\alpha\beta} + Id_2.$$

Because $|W_{\alpha\beta}| = \frac{f_{ii}}{C}(\frac{f_{ii}}{C} - \frac{2J}{C}) < 0$, it shows that one eigenvalue of $H_{\alpha\beta}$ is larger than 1. So x^* cannot be a stable equilibrium for the system.

Furthermore, if there exists only an activated neuron $x^*_{i\alpha}$ in column i, it holds that

$$\sum_{j\alpha \in P} f_{ij} x^*_{j\alpha} \geq \sum_{j\beta \in P} f_{ij} x^*_{j\beta}.$$

Theorem 4. *Suppose network 2.1 satisfies theorem 3. If there exists constant $h > 0$, then the equilibrium has at least one activated column.*

Proof. By contradiction, assume no activated neuron. Then $x^* \leq 0$ and $\sigma(x^*) = 0$, and it follows from equation 2.2 that $x^* = W\sigma(x^*) + H = H > 0$.

4.2 Dynamic Properties of CLM-DT-LT-RNNs. Below we discuss the dynamic properties of CLM-DT-LT-RNNs, which also helps for understanding the ACLM method set out in section 4.3.

Note that for G being invertible, network 2.1 is dynamically equivalent to

$$z_{i\alpha}(k+1) = \sigma\left(\frac{1}{C}\left[hJ - J\sum_{\beta=1}^{l} z_{i\beta}(k) + \sum_{j=1}^{n} f_{ij} z_{j\alpha}(k)\right] + z_{i\alpha}(k)\right) \quad (4.3)$$

by the transformation $z_{i\alpha}(k) = \sigma(x_{i\alpha}(k))$. For simplicity, we use network 4.3 to study the dynamic properties of network 2.1 and assume that network 4.3 satisfies theorem 4, W has no eigenvalue 1, and $Id_{nl} - W$ is nonsingular. Apart from the constraint of the LT function, network 4.3 is similar to a linear dynamics, where $z_{i\alpha}(k) \geq 0$ for all i, α:

$$z(k+1) = Wz(k) + H. \quad (4.4)$$

Therefore, we use the linear system eigensubspace analysis of network 4.4 as an indirect measure to study the dynamic of network 4.3. By inspecting the change of $z(k)$ on the zero boundary step by step in network 4.4, we try to find the possible or necessary conditions for the stable equilibria of network 4.3.

By equation 4.1, the eigenmodes of linear system 4.4 can be divided into two classes: DC-subspace, which is spanned by the eigenmodes

$$V_{i1} = \frac{1}{\sqrt{l}}\left(b_i^T, \ldots, b_i^T\right)^T, \quad \Lambda_{i1} = \frac{1}{C}(\lambda_i - Jl) + 1,$$

and AC-subspace, which is spanned by the remaining eigenmodes,

$$V_{i\beta\neq 1} = \left(\sqrt{\sum_{\alpha=1}^{l}\left(q_{\alpha}^{\beta\neq 1}\right)^2}\right)^{-\frac{1}{2}} \cdot \left(q_1^{\beta\neq 1} b_i^T, \ldots, q_l^{\beta\neq 1} b_i^T\right)^T,$$

$$\Lambda_{i\beta\neq 1} = \frac{1}{C}\lambda_i + 1,$$

where $i = 1, \ldots, n$ and $\beta = 1, \ldots, l$.

We denote that z_0^* is the fixed point of linear system 4.4. Because there exists at least one $\Lambda_{i\beta\neq 1} > 1$, z_0^* is unstable for both networks 4.3 and 4.4. Therefore, the possible stable equilibria of network 4.3 can exist only on the zero boundary.

In DC-subspace, we have

$$\begin{bmatrix} \frac{1}{l}\sum\limits_{\beta=1}^{l} z_{1\beta}(k+1) \\ \vdots \\ \frac{1}{l}\sum\limits_{\beta=1}^{l} z_{n\beta}(k+1) \end{bmatrix} = W_{DC} \begin{bmatrix} \frac{1}{l}\sum\limits_{\beta=1}^{l} z_{1\beta}(k) \\ \vdots \\ \frac{1}{l}\sum\limits_{\beta=1}^{l} z_{n\beta}(k) \end{bmatrix} - \frac{Jh}{C}\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \tag{4.5}$$

where $W_{DC} = \frac{1}{C} f + (1 - \frac{Jl}{C}) Id_n$. By equation 4.2, $|\Lambda_{i1}| < 1$. Then we can get

$$\begin{aligned} &\left(z_{DC}(k+1) - z_{DC}^*\right)^T \cdot \left(z_{DC}(k+1) - z_{DC}^*\right) \\ &\quad \le \left(z_{DC}(k) - z_{DC}^*\right)^T \cdot \left(z_{DC}(k) - z_{DC}^*\right) \end{aligned}, \tag{4.6}$$

where $z_{DC}(k+1) = [\frac{1}{l}\sum_{\beta=1}^{l} z_{1\beta}(k+1), \ldots, \frac{1}{l}\sum_{\beta=1}^{l} z_{n\beta}(k+1)]^T$ for $k \ge 0$, and z_{DC}^* is the fixed point of equation 4.5. We denote that $(z_{DC}^*)_i$ is the ith component of z_{DC}^*, and the relationship between z_{DC}^* and z_0^* is described as $z_0^* = (z_{DC}^*, z_{DC}^*, \ldots, z_{DC}^*)^T$.

On the other hand, from equation 4.4, we can find $z(k)$ in AC-subspace that

$$\begin{aligned} &\sum_{i=1}^{n}\sum_{\alpha=1}^{l}\left(z_{i\alpha}(k+1) - \frac{1}{l}\sum_{\beta=1}^{l} z_{i\beta}(k+1)\right)^2 \\ &\quad \ge \sum_{i=1}^{n}\sum_{\alpha=1}^{l}\left(z_{i\alpha}(k) - \frac{1}{l}\sum_{\beta=1}^{l} z_{i\beta}(k)\right)^2, \end{aligned} \tag{4.7}$$

where "=" holds only for the unstable equilibrium z_0^*.

By equations 4.6 and 4.7, four states can exist for neuron activities at the kth step:

State I:

$$\begin{cases} \frac{1}{l}\sum_{\beta=1}^{l} z_{i\beta}(k+1) > \frac{1}{l}\sum_{\beta=1}^{l} z_{i\beta}(k) \\ \frac{1}{l}\sum_{\beta=1}^{l} z_{i\beta}(k) < (z^*_{DC})_i \end{cases}$$

State II:

$$\begin{cases} \frac{1}{l}\sum_{\beta=1}^{l} z_{i\beta}(k+1) < \frac{1}{l}\sum_{\beta=1}^{l} z_{i\beta}(k) \\ \frac{1}{l}\sum_{\beta=1}^{l} z_{i\beta}(k) > (z^*_{DC})_i \end{cases}$$

State III:

$$\frac{1}{l}\sum_{\beta=1}^{l} z_{i\beta}(k) = (z^*_{DC})_i$$

and State IV:

$$\sum_{\beta=1}^{l} z_{i\beta}(k+1) = \sum_{\beta=1}^{l} z_{i\beta}(k) = 0.$$

For network 4.3, at a stable equilibrium, the trajectory at step k must hold that

$$\frac{1}{l}\sum_{\beta=1}^{l} \sigma(z_{i\beta}(k+1)) = \frac{1}{l}\sum_{\beta=1}^{l} z_{i\beta}(k). \tag{4.8}$$

Because state I cannot satisfy equation 4.8, the trajectory cannot get to a steady state at the kth step under state I. For state III, by theorem 4, the trajectory may move to and stay at the zero boundary. And if the trajectory crosses the zero boundary, the current state may convert to other states. Therefore, at the kth step, a trajectory will arrive at a steady equilibrium in two ways: (1) all neurons are under state II; (2) some neurons are under the state II, and others are under those states except for the state I.

A sketch of the initial CLM dynamic referring to our eigensubspace analysis and three states is depicted in Figure 2. Here, $z'_i(k+1)$ is the iteration solution at the kth step for network 4.4, while $z_i(k+1)$ is the

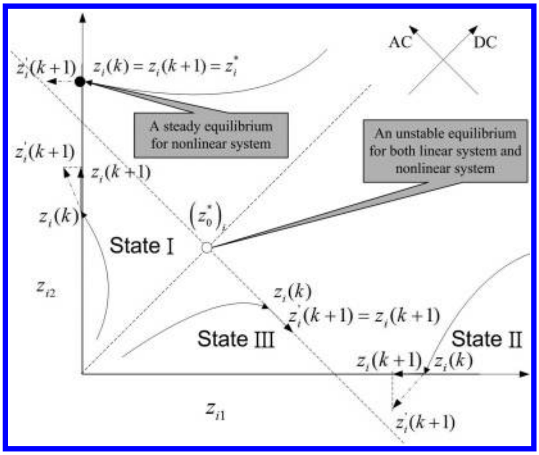


Figure 2: Discrete-time CLM dynamics for two layers. Shown are activity trajectories states for two activities z_{i1}, z_{i2} of a single column i. Here denote $z_i(k) = (z_{i1}(k), z_{i2}(k))^T$, $(z_0^*)_i = ((z_{DC}^*)_{i1}, (z_{DC}^*)_{i2}))^T$, and $z_i^* = (z_{i1}^*, z_{i2}^*)^T$. $z'(k+1)$ and $z(k+1)$ are the components of the solution for linear system 4.4 and nonlinear system 4.3 at step $k+1$, respectively.

corresponding solution for network 4.3 affected by nonlinear function σ. When $z_{i\alpha}(k), z_{i\alpha}(k+1) \geq 0$ for all i, α, the trajectories of two networks are the same. If some neurons' activities reach or cross the zero boundary to a negative area, they no longer contribute to the nonlinear system 4.3, and their value in the corresponding network 4.4 is set as zero. Therefore, we can study the trajectory of network 4.3 in R_+^{nl} step by step with the similarity of two networks. Furthermore, if we choose $0 < z_{i\alpha}(0) \ll h$, $J\,L \gg \lambda_i(f)$, then the trajectory will first go to the affine subspace passed though (z_0^*) and approach the boundary along the affine subspace, which is similar to the description in Wersing et al. (2001).

4.3 Comparison Between the SCLM Method and the ACLM Method. In Wersing et al. (2001), an asynchronous CLM (ACLM) method comes from a continuous-time CLM neural network, which can be described as

$$\dot{y_{i\alpha}}(t) = -y_{i\alpha}(t) + \sigma\left(J\,h - J\sum_{\beta=1}^{l} y_{i\beta}(t) + \sum_{j=1}^{n} f_{ij}\,y_{j\alpha}(t) + y_{i\alpha}(t)\right). \quad (4.9)$$

For networks 4.9 and 4.3, with the same h, J, their equilibria are in the same solution space. Furthermore, if f is a positive-definite matrix, then they have the same equilibria.

Here, we give a novel synchronous CLM (SCLM) method based on network 3.4 to implement the CLM dynamic, which can be found in appendix B. Because the weight W always needs an enormous amount of memory, it is impossible to directly simulate a large network 4.3 on a PC. For example, if the feature number is 10,000, the layer number is 10, and the computer needs 2 bytes to store one data point, then the storage for W is $(10{,}000 \times 10 \times 2)^2/(1024)^3 \simeq 37.25$ GB. The SCLM method can greatly decrease the storage requirements. For the previous example, we do not need to store W but we do need to store f and Π_l (here $l = 10$), which occupy $(10{,}000 * 2)^2/(1024)^2 \simeq 381.47$ MB and $(10 * 2)^2/1024 \simeq 0.39$ KB, respectively (here 1 GB = 1024 MB = 1024^2 KB). On the other hand, the ACLM method mainly needs to store f. Therefore, both the SCLM and the ACLM methods have similar memory requirements. But the synchronous update mode can significantly improve CLM efficiency. (See the comparison table of two CLM methods in example 4.) It turns out that the SCLM method is always about 90 times faster than the ACLM method with 10,000 features and 10 layers.

Theorem 5. *Suppose network 3.4 satisfies theorem 2. If f is a positive-definite matrix and constant $h > 0$, then the equilibrium points of network 3.4 have some properties such as:*

(i) For any $x^* \in \Omega$, there must exist at least one activated neuron.
(ii) If more than one activated neuron exist in a columns, then their outputs are equal.

Proof. Here we need only to prove property ii. Let $x^* = [x^*_\Sigma, x^*_z]^T$, where x^*_z is the inactive neurons set, and $x^*_\Sigma = [x^*_{\Sigma_{11}}, x^*_{\Sigma_{12}}, \ldots, x^*_{\Sigma_{1d_1}}, \ldots, x^*_{\Sigma_{n1}}, x^*_{\Sigma_{n2}}, \ldots, x^*_{\Sigma_{nd_n}}]^T$, where the index $\Sigma_{i\alpha} \in P$, $\bigcup_{\alpha=1}^{d_i} \{x^*_{\Sigma_{i\alpha}}\} = \{x^*_{i\alpha} | x^*_{i\alpha} > 0\}$ and d_i is the total element number of the set $\{x^*_{i\alpha} | x^*_{i\alpha} > 0\}$ for $i = 1, \ldots, n$, and $\alpha = 1, \ldots, l$.

Now we can rewrite equation 2.2 in the equilibrium state by

$$x^* = \begin{bmatrix} x^*_\Sigma \\ x^*_z \end{bmatrix} = \begin{bmatrix} W_\Sigma & W_1 \\ W_2 & W_3 \end{bmatrix} \sigma\left(\begin{bmatrix} x^*_\Sigma \\ x^*_z \end{bmatrix}\right) + \begin{bmatrix} H_\Sigma \\ H_z \end{bmatrix}, \tag{4.10}$$

where W_Σ, W_1, W_2, W_3 are the submatrices of W and H_Σ and H_z are the part of H according to the transform. Clearly W_Σ is a principal submatrix of W.

From equation 4.11 in an equilibrium point state, it holds that

$$\begin{cases} x^*_\Sigma = W_\Sigma x^*_\Sigma + H_\Sigma > 0 \\ x^*_z = W_2 x^*_\Sigma + H_z \leq 0 \end{cases}. \tag{4.11}$$

From equations 2.3, 2.4, A.1, and 4.11, we have

$$G_\Sigma x^*_\Sigma = -H_\Sigma. \tag{4.12}$$

By lemma 4, G_Σ has no zero eigenvalues, so equation 4.12 has only one solution. On the other hand, by observing W_Σ, we can find that if more neurons coactivate in a column at the equilibrium state, the solution that satisfies property (ii) can be one of possible solutions. Therefore, the unique solution must satisfy this property. Furthermore, if this solution cannot satisfy equation 4.11, there is no equilibrium under such condition.

By theorem 5, if networks 4.3 and 4.9 satisfy theorem 5 and have the same h, J, their equilibria all satisfy equations 4.11 and 4.12. Because there is only one solution for equation 2.16, the solution set is unique. Therefore, both networks have the same equilibria.

5 Simulations

In this section, we provide simulation results to illustrate and verify the theory developed. All programs were coded in Matlab 2008a and were run on a PC with 1 Intel i7 920@2.67 GHz CPU, 6 GB RAM, and Windows Vista Ultimate Service Pack 1 64-bit operating system. Theorems 1 and 2 can be looked at as special cases in Yi and Tan (2004), so we skip the simulations for these two theorems.

Example 1. Consider a CLM neural network with three layers with four neurons in each layer,

$$x(k+1) = W\sigma(x(k)) + H, \tag{5.1}$$

where $W = \frac{1}{C}[f \otimes Id_3 - J \cdot \Pi_3 \otimes Id_4] + Id_{12}$, $H = \frac{J}{C}[1, 1, \ldots, 1]^T$, $J = 5$, $C = 16$, and

$$f = \begin{bmatrix} 0.21 & 0.01 & 0.02 & -0.03 \\ 0.01 & 0.21 & 0.03 & -0.04 \\ 0.02 & 0.03 & 0.16 & -0.05 \\ -0.03 & -0.04 & -0.05 & 0.31 \end{bmatrix}.$$

Network 5.1 satisfies theorem 4. With initial condition $x(0) = [0.0762\ 0.0474\ 0.0484\ 0.1706\ 0.0757\ 0.1354\ 0.0377\ 0.1745\ 0.0946\ 0.2150\ 0.2134\ 0.1484]^T$, Figure 3 shows the convergence result of the trajectory. This network displays a column WTA behavior, and only one neuron in a column is activated in the stable state $x^* = [-0.0177\ -0.0191\ -0.0171\ 1.0661\ -0.0157\ -0.0164\ -0.0137\ -0.0207\ 1.05041.0526\ 1.0439\ -0.0285]^T$. Note that two group winners can be found: $\{x_{41}\}$ in layer 1 and $\{x_{13}, x_{23}, x_{33}\}$ in layer 3.

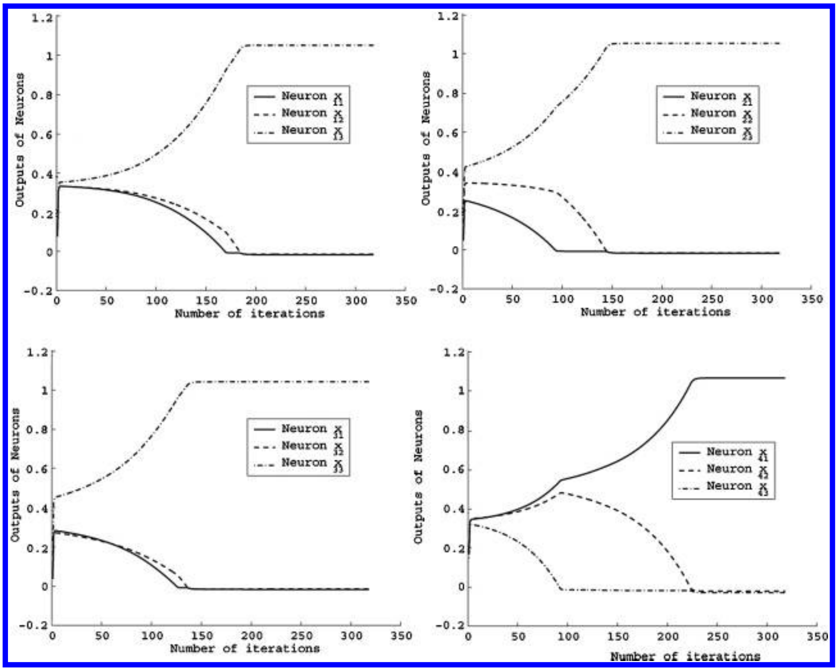


Figure 3: Trajectory of network 5.1 in four columns. A column WTA behavior can be observed.

Example 2. Consider network 5.1 in example 1, and set $H = -\frac{J}{C}[10, 10, \ldots, 10]^T$. Clearly, network 5.1 satisfies theorem 3, and there is no activated column in this network.

Example 3. Consider a CLM neural network with two layers with two neurons in each layer,

$$z(k+1) = \sigma(Wz(k) + H), \tag{5.2}$$

where $W = \frac{1}{C}[f \otimes Id_2 - J \cdot \Pi_2 \otimes Id_2] + Id_4$, $H = \frac{J}{C}[1, 1, \ldots, 1]^T$, $J = 2.25$, $C = 5$, and $f = [\begin{smallmatrix} 1 & -0.5 \\ -0.5 & 1 \end{smallmatrix}]$. We use a continuous-time CLM neural network for comparison:

$$\dot{y_{i\alpha}}(t) = -y_{i\alpha}(t) + \sigma\left(Jh - J\sum_{\beta=1}^{l} y_{i\beta}(t) + \sum_{j=1}^{n} f_{ij}\, y_{j\alpha}(t) + y_{i\alpha}(t)\right), \tag{5.3}$$

where $1 \leq i, \alpha \leq 2$.

Both networks satisfy theorem 5. By simple calculations, we can know five equilibria for them: $x_1^* = [1.8\ \ 0\ \ 0\ \ 1.8]^T$, $x_2^* = [0\ \ 1.8\ \ 1.8\ \ 0]^T$, $x_3^* = [0\ \ 0\ \ 1.2857\ \ 1.2857]^T$, $x_4^* = [1.2857\ \ 1.2857\ \ 0\ \ 0]^T$, and $x_5^* = [0.5625\ \ 0.5625\ \ 0.5625\ \ 0.5625]^T$. Figure 4 shows the convergence of both networks for 100 trajectories originating from randomly selected initial points. We also add the ACLM method for comparison. Clearly, x_1^*, x_2^*, x_3^*, x_4^*, are all stable, and x_5^* is unstable.

Example 4. In this example, we compare our SCLM method with the ACLM method. f is defined in appendix C. Both methods are applied to a 100 × 100 picture for image segmentation. Here, the feature number 10,000 and the layer number $L = \{5\ 10\}$. For each selected L, the test time is 10. Therefore, there are $2 \times 2 \times 10 = 40$ tests.

Here, we compare two methods in two aspects: iteration speed and performance. For simplicity, we set the annealing temperature $T = 0$. Although each method has a different energy function, we use a lateral contribution energy function as an indirect measure to estimate the performance. The lateral contribution energy function is described as $E_{CLM} = -\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\alpha=1}^{l} f_{ij}\zeta_{i\alpha}^*\zeta_{j\alpha}^*$, where $\zeta_{i\alpha}^*$ is defined for both methods by

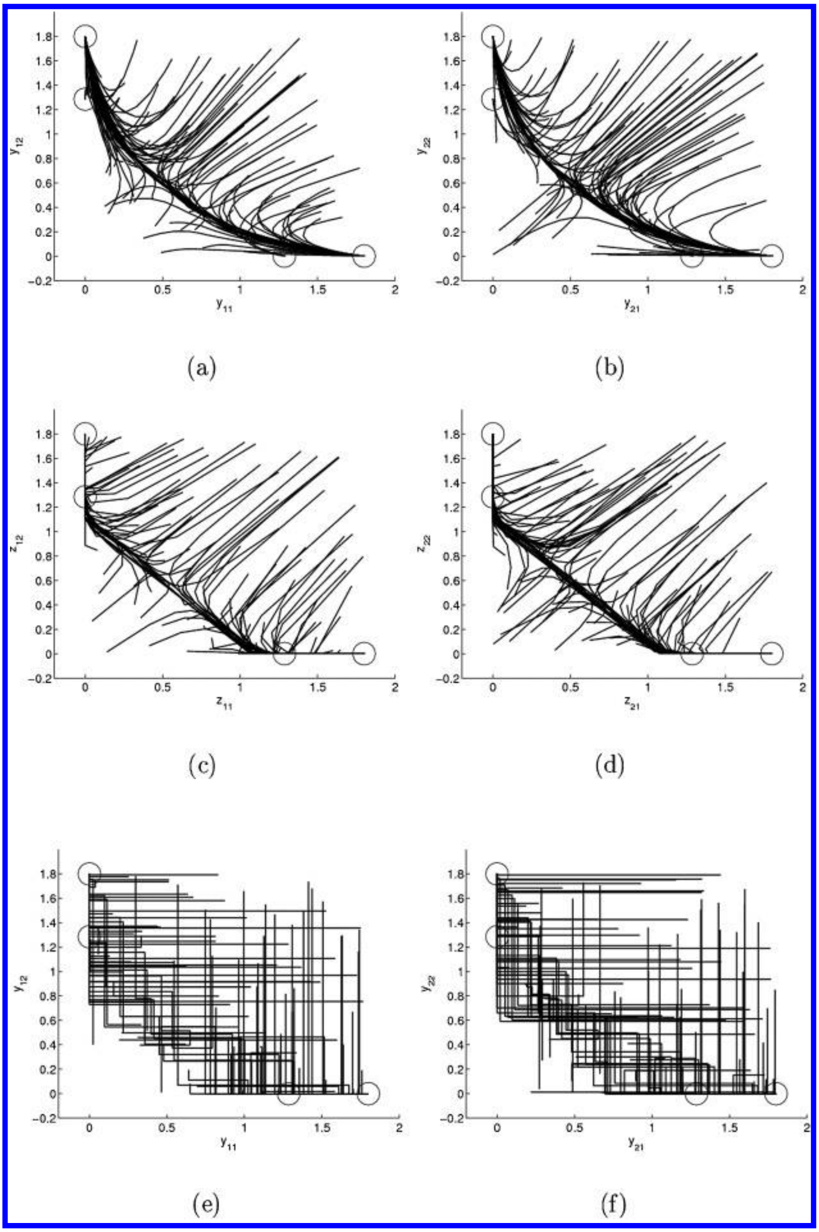
$$\zeta_{i\alpha}^* = \begin{cases} 1, & if\ y_{i\alpha}^* > 0\, or\ z_{i\alpha}^* > 0 \\ 0, & if\ y_{i\alpha}^* \le 0\, or\ z_{i\alpha}^* \le 0 \end{cases},$$

where $y_{i\alpha}^*$, $z_{i\alpha}^*$ are the stable fixed points of networks 4.9 and 3.4, respectively. In brief, for fixed n and l, the lower the energy value, the better the performance.

Table 1 shows the comparison. From the table, it is hard to tell which method performs better. This conclusion can also be derived from the similarity of dynamics between continuous-time and discrete-time networks. On the other hand, performance always depends on the initial value, f, and some parameters, so the choice between the two methods in real applications is not unique. It can be observed that with more features and layers, the ACLM method is more time-consuming. With the same conditions, our method runs faster than the ACLM method. We also notice that for fixed N, the ACLM method is more sensitive to layer number L than our method. Figure 5 compares the image segmentation results for both methods.

Example 5. Here, the SCLM method is applied to 128 × 128 Lena pictures for image segmentation. There are 16,384 total features and $L = \{3\ \ 6\ \ 9\}$ layers.

Figure 6 shows the image segmentation results. Here, the time and the energy for three layers are 1.9682×10^3 secs, and 3.8590×10^6, respectively. For



(a) (b)

(c) (d)

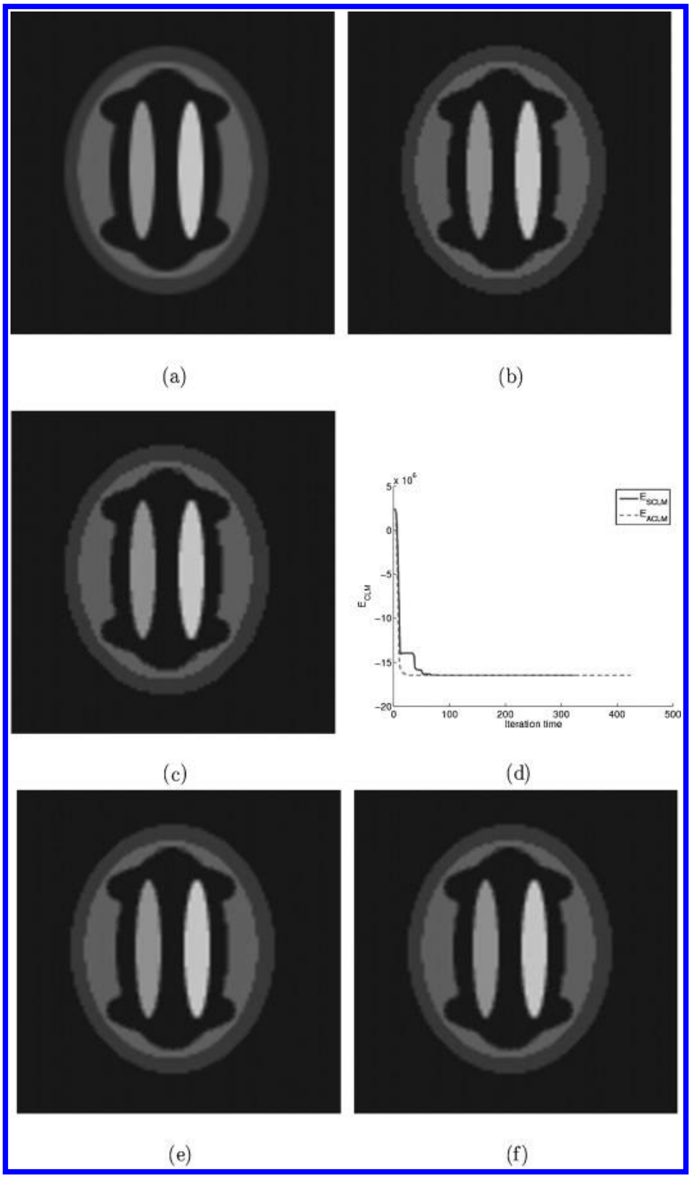
(e) (f)

Figure 4: Convergence results for different networks, with each stable fixed point marked with an ellipse. (a, b) Convergence results for network 5.3. (c, d) Convergence results for network 5.2. (e, f) Convergence results for network 5.2 using the ACLM method.

Table 1: Comparisons of ACLM Methods and SCLM Method ($N = 10{,}000$, $L = 5, 10$).

		Time (sec.)			E_{CLM} ($\times 10^7$)		
CLM Method	Number of Layers	Minimum	Maximum	Average	Minimum	Maximum	Average
Asynchronous	5	2.3558×10^4	9.0035×10^4	5.1264×10^4	-1.6521	$-$ **1.6481**	-1.6485
CLM	10	6.5646×10^4	1.4473×10^5	9.9519×10^4	$-$**1.6538**	$-$**1.6538**	$-$**1.6538**
Synchronous	5	**850.2630**	**1.5700** $\times 10^3$	**1.1429** $\times 10^3$	$-$**1.6523**	-1.6477	$-$**1.6492**
CLM	10	**752.2480**	**1.4640** $\times 10^3$	**1.0427** $\times 10^3$	-1.6535	-1.6522	-1.6530

Note: The best results, which include running time and performance (E_{CLM}) according to the different layers, are shown in boldface.



(a) (b)

(c) (d)

(e) (f)

Figure 5: Image segmentation for 100 × 100 image. (a) Original image. (b) Image segmented with 5 layers using the SCLM method. (c) Image segmented with 5 layers using the ACLM method. (d) Lateral contribution energies of two methods. Here, we calculate E_{SCLM} once per l iterative times, and calculate E_{ACLM} once per $n \times l$ iterative times. (e) Image segmented with 10 layers using SCLM method. (f) Image segmented with 10 layers using the ACLM method.

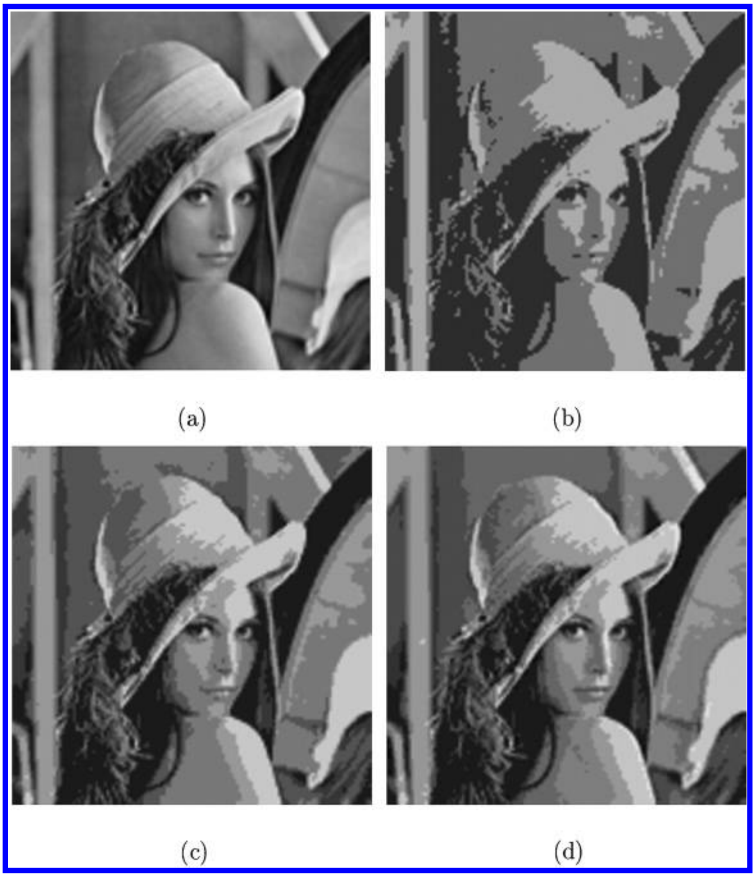


Figure 6: Image segmentation worked with a 128 × 128 Lena image using the SCLM method. (a) Original image. (b) Image segmented with three layers. (c) Image segmented with six layers. (d) Image segmented with nine layers.

six layers, they are 3.5139×10^3 secs, and -5.9667×10^6, respectively. For nine layers, they are 4.8886×10^3 secs, and -6.3551×10^6, respectively. Clearly, the image segmentation quality can be improved with more layers, which decreases the lateral contribution energy as well.

6 Conclusion

In this letter, we investigate the competitive layer model for a class of discrete-time recurrent neural networks with LT neurons. We first discuss three basic dynamical problems: boundedness, global attractivity, and complete stability. Then we present two theorems to let the networks have CLM phenomena. In addition, we outline some dynamic properties analysis and give a novel SCLM method, which has similar performance and storage allocation but simulations are faster compared with the ACLM method. Simulations have been carried out to validate the performance of our theoretical findings.

Considering current technical trends, multicore processor technology significantly increases parallel processing capability compared with single-core processing. Therefore, with the development of software and hardware of parallel computation, the efficiency and speed of our SCLM method can be improved in the future.

The method we have described may well be extended to other applications dealing with complex optimization problems. In essence, the CLM can be looked at as an optimization method to search for a better feature-grouping solution among those possible solutions. Therefore, the method described here may be extended to other applications dealing with complex optimization problems.

Appendix A: Proof of Lemma 4

According to equation 2.4, we define

$$G_\Sigma = -\frac{J}{C}P_\Sigma + \frac{1}{C}F_\Sigma, \tag{A.1}$$

where P_Σ, F_Σ are the $s \times s$ principal submatrix of P, F respectively. Let $rank(P_\Sigma)$, $rank(P)$ be the rank of P_Σ and P respectively, and $rank(P_\Sigma) = m$. Clearly, $1 \leq m \leq rank(P) = n$. Since P_Σ is a principal submatrix of P, by lemma 2, the eigenvalues $\{\lambda_i(P_\Sigma)\}$ of P_Σ satisfy (in increasing order) $\lambda_i(P_\Sigma) \leq \lambda_{\max}(P) = l$, where $i = 1, \ldots, s$ and $\lambda_{\max}(P)$ is the max eigenvalue of P.

It can be proved that the eigenvalues $\{\lambda_i(P_\Sigma)\}$ satisfy

$$\begin{cases} \lambda_1(P_\Sigma) = \lambda_2(P_\Sigma) = \cdots = \lambda_{s-m}(P_\Sigma) = 0 \\ 1 \leq \lambda_{s-m+1}(P_\Sigma) \leq \cdots \leq \lambda_{s-1}(P_\Sigma) \leq \lambda_s(P_\Sigma) \leq l \end{cases}. \tag{A.2}$$

Define $A = -\frac{J}{C}P_\Sigma$ and $B = \frac{1}{C}F_\Sigma$. Obviously, by equation A.2, the eigenvalues $\{\lambda_i(A)\}$ of A satisfy

$$\begin{cases} -\frac{Jl}{C} \leq \lambda_1(A) \leq \lambda_2(A) \leq \cdots \leq \lambda_m(A) \leq -\frac{J}{C} \\ \lambda_{m+1}(A) = \lambda_{m+2}(A) = \cdots = \lambda_s(A) = 0 \end{cases}. \tag{A.3}$$

By lemma 2, the eigenvalues $\{\lambda_i(B)\}$ of B also satisfy

$$\frac{1}{C}\lambda_{\min}(f) \leq \lambda_1(B) \leq \lambda_2(B) \leq \cdots \leq \lambda_s(B) \leq \frac{1}{C}\lambda_{\max}(f). \tag{A.4}$$

Since f is a positive-definite matrix and $J > \max_{1\le i\le n}\{\lambda_i\}$, by lemma 3 and equations A.3 and A.4, it holds that the eigenvalues $\{\lambda_i(G_\Sigma)\}$ of G_Σ satisfy

$$\begin{cases} \lambda_1(G_\Sigma) \le \lambda_1(A) + \lambda_s(B) \le -\frac{J}{C} + \frac{1}{C}\lambda_{\max}(f) < 0 \\ \vdots \\ \lambda_m(G_\Sigma) \le \lambda_m(A) + \lambda_s(B) \le -\frac{J}{C} + \frac{1}{C}\lambda_{\max}(f) < 0 \\ \lambda_{m+1}(G_\Sigma) \ge \lambda_{m+1}(A) + \lambda_1(B) \ge \frac{1}{C}\lambda_{\min}(f) > 0 \\ \vdots \\ \lambda_s(G_\Sigma) \ge \lambda_s(A) + \lambda_1(B) \ge \frac{1}{C}\lambda_{\min}(f) > 0 \end{cases}.$$

Appendix B: SCLM Method

1. Initialize all $z_{i\alpha}(0)$ with small random values around $z_{i\alpha}(0) \in (0, 0+\epsilon]$. Calculate $\lambda_{\max}$ and $\lambda_{\min}$. Set $J_0 = \max\{\frac{1}{l}\max\{|\lambda_{\max}|, |\lambda_{\min}|\}, \max_{1\le i\le n}\{\sum_{j=1}^n f_{ij}^+\}\}$, $J = 1.01 * J_0$, $C = 1.01 * J\,l$, $\tau = 1.01$, $\tau_{\max} = 1.35$, $\rho = 0.001$, $\omega = 1$, $\zeta = 0.1$, $N^1_{X^1(K+1)} = 0$, $N_{queue} = 0$, $N^1_{queue} = 100$, $T = \lambda_{\max}$, $\eta_T = 0.99$, $k = 0$, $h = 1$,

$$E_C = \frac{J}{C}\Pi_l,\ Z(0) = \begin{bmatrix} z_{11}(0) & \cdots & z_{1l}(0) \\ \vdots & \ddots & \vdots \\ z_{n1}(0) & \cdots & z_{nl}(0) \end{bmatrix}_{n\times l},$$

$$H_0 = \begin{bmatrix} h & \cdots & h \\ \vdots & \ddots & \vdots \\ h & \cdots & h \end{bmatrix}_{n\times l}, \quad \text{and}$$

$$H = \frac{J}{C}H_0;$$

2. Calculate $\xi = H + \frac{1}{C} * f * Z(k) - \frac{1}{C} * T * Z(k) - Z(k) * E_C + Z(k)$. If $mod(k+1, l) == 0$, then $T = T * \eta_T$ (without self-inhibitory annealing, set $T = 0$);
3. Set $Z(k+1) = \max(\xi, 0)$. If $\omega == 1$, then find $N_{X(K+1)}$, the number of $z_{i\alpha}(k+1) == -1$ in $Z(k+1)$. If $N_{Z(K+1)} < n$, then $Z(k) = Z(k+1)$, $k = k+1$, go to Step 2; otherwise $\omega = 0$, go to Step 4;
4. Calculate the error between $Z(k)$ and $Z(k+1)$. If $|z_{i\alpha}(k+1) - z_{i\alpha}(k)| < \rho$ for all i, α, then go to Step 5; otherwise $Z(k) = Z(k+1)$, $k = k+1$, go to Step 2;

5. Find $Z_0(K+1)$, here $Z_0(K+1) = \{z_{i\alpha}(k+1) \mid z_{i\alpha}(k+1) > \zeta\}$ for all i, α. Set $Z^1(K+1) = Z_0(K+1) - Mean(Z_0(K+1))$, where $Mean(Z_0(K+1))$ is the mean number of $Z_0(K+1)$, then find $N_{Z^1(K+1)}$, the number of $z^1_{i\alpha}(k+1) > 0.1/n$ in $Z^1(k+1)$. If $N_{Z^1(K+1)} == 0$, then go to the end; otherwise go to Step 6;
6. If $N_{queue} == 0$, then $N_{queue} = N_{queue} + 1$, $N^1_{Z^1(K+1)} = N_{Z^1(K+1)}$, go to Step 9; otherwise go to Step 7;
7. if $N^1_{Z^1(K+1)} \neq N_{Z^1(K+1)}$, then $N_{queue} = 0$, $N^1_{Z^1(K+1)} = N_{Z^1(K+1)}$, go to Step 9; otherwise $N_{queue} = N_{queue} + 1$, Go to Step 8;
8. If $N_{queue} == N^1_{queue}$, then go to the end; otherwise go to Step 9;
9. Set $J = J * \tau$, $\tau = \tau + 0.01$, $C = 1.01 * Jl$, $H = \frac{l}{C} H_0$, $E_C = \frac{l}{C} \Pi_l$, $Z(k) = Z(k+1)$, $k = k+1$. If $\tau < \tau_{\max}$, then go to Step 2; otherwise go to the end.

Appendix C: Lateral Interaction for Image Segmentation

In this letter, the lateral interaction f is identical in all l layers and is symmetric:

$$f_{ij} = \Gamma(\phi_{i,j}) = \begin{cases} 0.5, & i == j \\ \dfrac{\phi_{i,j}}{\max\limits_{1\le i',j'\le n} \{\phi_{i',j'}\}}, & i \neq j, \phi_{i,j} \geq 0 \\ \dfrac{\phi_{i,j}}{\max\limits_{1\le i',j'\le n} \{-\phi_{i',j'}\}}, & i \neq j, \phi_{i,j} < 0 \end{cases}.$$

The compatibility $\phi_{i,j}$ between pixel $P(i_x, i_y)$ and $P(j_x, j_y)$ with gray-level $g(i_x, i_y)$ and $g(j_x, j_y)$ is given by $\phi_{i,j} = m_1 e^{-v/k_1}(m_2 e^{-d/k_2} + 1) - \theta$, where $v = |g(i_x, i_y) - g(j_x, j_y)|$ is the difference between the two gray-levels, $d = \sqrt{(i_x - j_x)^2 + (i_y - j_y)^2}$ is the Euclidean distance of the two pixels, $m_1(> 0)$ and $m_2(> 0)$ balance between the influence of v and d to ϕ, $k_1(>0)$ controls the sharpness of v, $k_2(>0)$ controls the spatial range of d, and θ is the threshold. The parameters used here are $m_1 = 2$, $m_2 = 3$, $k_1 = 100$, $k_2 = 0.5$, $\theta = 1.7$.

Acknowledgments

The research reported here has been supported by the China Scholarship Council and National Science Foundation of China under Grant 60973070. We thank Zhang Yi, Yonglin Zeng, Ping Li, Artur Abdullin, Dongqing Chen, Lijun Zhang, Minqing Zhang, Zehui Qu, and Ziming Zheng for their useful discussions and comments. We also thank the anonymous reviewers for their suggestions.

References

Feng, J. (1997). Lyapunov functions for neural nets with nondifferentiable input-output characteristics. *Neural Computation*, *9*(1), 43–49.

Hahnloser, R. H. R. (1998). On the piecewise analysis of networks of linear threshold neurons. *Neural Networks*, *11*, 691–697.

Hahnloser, R. H. R., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., & Seung, H. S. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, *405*, 947–951.

Hahnloser, R. H., Sebastian, H., & Slotine, J.-J. (2003). Permitted and forbidden sets in symmetric threshold-linear networks. *Neural Computation*, *15*(3), 621–638.

Horn, R. A., & Johnson, C. R. (1985). *Matrix analysis*. Cambridge: Cambridge University Press.

Hu, S., & Wang, J. (2002). Global stability of a class of discrete-time recurrent neural networks. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, *49*(8), 1104–1117.

Jin, L., Nikiforuk, P. N., & Gupta, M. M. (1994). Absolute stability conditions for discrete-time recurrent neural networks. *IEEE Transactions on Neural Networks*, *5*(6), 954–964.

Koffka, K. (1962). *Principles of gestalt psychology*. London: Routledge & Paul.

Ontrup, J., & Ritter, H. (1998). *Perceptual grouping in a neural model: Reproducing human texture perception* (Tech. Rep. SFB360-TR-98/6). Bielefeld: University of Bielefeld.

Ritter, H. (1990). A spatial approach to feature linking. In *Proceedings of the International Neural Network Conference Paris* (Vol. 2, pp. 898–901). Norwell, MA: Kluwer.

Si, J., & Michel, A. N. (1995). Analysis and synthesis of a class of discrete-time neural networks with multilevel threshold neurons. *IEEE Transactions on Neural Networks*, *6*(1), 105–116.

Tang, H. J., Tan, K. C., & Zhang, W. (2005). Analysis of cyclic dynamics for networks of linear threshold neurons. *Neural Computation*, *17*, 97–114.

von der Malsburg, C. (1981). *The correlation theory of brain function* (Tech. Rep. 81-2). Göttingen: MPI Göttingen.

von der Malsburg, C. (1995). Binding in models of perception and brain function. *Current Opinion in Neurobiology*, *5*, 520–526.

Wang, L., & Xu, Z. (2006). Sufficient and necessary conditions for global exponential stability of discrete-time recurrent neural networks. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *53*(6), 1373–1380.

Weng, S., Wersing, H., Steil, J. J., & Ritter, H. (2006). Learning lateral interactions for feature binding and sensory segmentation from prototypic basis interactions. *IEEE Transactions on Neural Networks*, *17*(2), 843–862.

Wersing, H., Steil, J. J., & Ritter, H. (2001). A competitive layer model for feature binding and sensory segmentation. *Neural Computation*, *13*, 357–387.

Xie, X., Hahnloser, R. H. R., & Seung, H. S. (2002). Selectively grouping neurons in recurrent networks of lateral inhibition. *Neural Computation*, *14*, 2627–2646.

Yi, Z., Heng, P. A., & Fung, P. F. (2000). Winner-take-all discrete recurrent neural networks. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, *47*(12), 1584–1589.

Yi, Z., & Tan, K. K. (2004). Multistability of discrete-time recurrent neural networks with unsaturating piecewise linear activation functions. *IEEE Transactions on Neural Networks*, *15*(2), 329–336.

Yi, Z., Tan, K. K., & Lee, T. L. (2003). Multistability analysis for recurrent neural networks with unsaturating piecewise linear transfer functions. *Neural Computation*, *15*, 639–662.

Zhang, L., Yi, Z., & Yu, J. (2008). Multiperiodicity and attractivity of delayed recurrent neural networks with unsaturating piecewise linear transfer functions. *IEEE Transactions on Neual Networks*, *19*(1), 158–167.

Zhang, L., Yi, Z., Zhang, S. L., & Heng, P. A. (2009). Activity invariant sets and exponentially stable attractors of linear threshold discrete-time recurrent neural networks. *IEEE Transactions on Automatic Control*, *54*(6), 1341–1347.

Received November 11, 2008; accepted February 2, 2010.

This article has been cited by:

2. Jan Steffen, Michael Pardowitz, Jochen J. Steil, Helge Ritter. 2011. Integrating feature maps and competitive layer architectures for motion segmentation. *Neurocomputing* . [CrossRef]