ADDING DECAYING SELF-FEEDBACK CONTINUOUS HOPFIELD NEURAL NETWORK CONVERGENCE ANALYSIS IN THE HYPER-CUBE SPACE*

Chunguo Fei†, *Baili Su*‡

Abstract: After sigmoid activation function is replaced with piecewise linear activation function, the adding decaying self-feedback continuous Hopfield neural network (ADSCHNN) searching space changes to hyper-cube space, i.e. the simplified ADSCHNN is obtained. Then, convergence analysis is given for the simplified ADSCHNN in hyper-cube space. It is proved through convergence analysis that the ADSCHNN outperforms the continuous Hopfield neural network (CHNN), when they are applied to solve optimization problem. It is also proved that when extra self-feedback is negative, the ADSCHNN is more effective than the extra self-feedback is positive, when the ADSCHNN is applied to solve TSP.

Key words: *Adding decaying self-feedback continuous Hopfield neural network, convergence analysis, piecewise linear function, hyper-cube*

Received: April 24, 2011
Revised and accepted: June 26, 2012

1. Introduction

Since Hopfield proposed Hopfield Neural Network (HNN) [1, 2], many studies about the convergence or stability of HNN or Hopfield-type neural networks have been made [3-7]. These obtained results provide some theoretical foundation of performance analysis of Hopfield-type neural networks. In this paper, one convergence

*This work is supported by the Central College Basic Research Foundation of China (ZXH2010D005, ZXH2012G005), Natural Science Foundation of Shandong Province (ZR2010FM022), National Specialized Research Fund for the Doctoral Programme of Higher Education (20113705120003), Science and Technology Planning Project of Shandong Higher Education Institutions of China (J10LG28) and Tianjin Research Program of Application Foundation and Advanced Technology (11JCYBJC07000).

†Chunguo Fei
College of Aeronautical Automation, Civil Aviation University of China, Tianjin, China, 300300, Tel: +086 13662080986, E-mail: `fchunguo@gmail.com`

‡Baili Su
School of Electrical Information and Automation, Qufu Normal University, Shandong, China, 276826

analysis is investigated about one kind of Hopfield-type neural network which was proposed in [8].

Earlier we proposed the adding decaying self-feedback continuous Hopfield neural network (ADSCHNN) [8] by adding the decaying extra-feedback to continuous Hopfield neural network (CHNN). This neural network was applied to solve traveling salesman problem (TSP). Results of some simulation published in [8] showed that the ADSCHNN is more effective than the CHNN and the ADSCHNN with negative extra self-feedback is better than that with positive extra self-feedback, when they were used to solve TSP. However, authors did not explain the reasons, just gave the conditions under which the ADSCHNN energy increases, decreases or stays constant. This paper will simplify the ADSCHNN by using piecewise linear function as activation function, and then perform convergence analysis to explain the reasons in the hyper-cube space.

2. Adding Decaying Self-Feedback Continuous Hopfield Neural Network

The network differential equation for the ADSCHNN is

$$\begin{cases} C_i \frac{dy_i}{dt} = -\frac{y_i}{R_i} + \sum\limits_{j=1}^{n} w_{ij} x_j + I_i + z_{ii}(t)\, x_i, \\ x_i = \psi(y_i) = \frac{1}{1+e^{-y_i/\varepsilon}}, \\ z_{ii}(t) = z_{ii}(0) e^{-\beta t} \end{cases} \tag{1}$$

where $C_i > 0$, $R_i > 0$, ψ is monotonously increasing and continuous, z_{ii} is the adding decaying extra self-feedback, n is the number of neuron, x_i is the internal state of neuron i, y_i is the output of neuron i, I_i is the threshold value of neuron i, and w_{ij} is the symmetric synaptic weight. Its energy function is

$$E = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} x_i x_j - \sum_{i=1}^{n} x_i I_i + \sum_{i=1}^{n} \frac{1}{R_i} \int_0^{x_i} \psi^{-1}(x) dx, \tag{2}$$

which is the same as the CHNN's. In [8], the following theorem was given to show the conditions under which the ADSCHNN's energy increases, decreases or maintains:

Theorem 1 Under the conditions that ψ is monotonously increasing and continuous activation function, $C_i > 0$, and $w_{ij} = w_{ji}$, and asynchronous cyclic updating of the neural network model is employed. The ADSCHNN has the following properties:

(A) When $z_{ii} x_i < 0$, if $z_{ii} x_i > \frac{\partial E}{\partial x_i}$ or $\frac{\partial E}{\partial x_i} > 0$, then $\frac{dE}{dt} < 0$, i.e. the energy decreases; if $z_{ii} x_i < \frac{\partial E}{\partial x_i} < 0$, then $\frac{dE}{dt} > 0$, i.e. the energy increases.

(B) When $z_{ii} x_i > 0$, if $z_{ii} x_i < \frac{\partial E}{\partial x_i}$ or $\frac{\partial E}{\partial x_i} < 0$, then $\frac{dE}{dt} < 0$, i.e. the energy decreases; if $z_{ii} x_i > \frac{\partial E}{\partial x_i} > 0$, then $\frac{dE}{dt} > 0$, i.e. the energy increases.

(C) If and only if $z_{ii} x_i = \frac{\partial E}{\partial x_i}$ or $\frac{\partial E}{\partial x_i} = 0$, then $\frac{dE}{dt} = 0$, i.e. the energy maintains.

According to **Theorem 1**, the ADSCHNN energy can increase, decrease or maintain when network satisfies some conditions. Therefore, the ADSCHNN may avoid converging to local minima. In order to make the neural network eventually stable, the effect of the extra self-feedbacks is reduced, i.e. let $|z_{ii}(t)|$ approach zero as time increases. During the periodic time that the extra self-feedbacks are in effect, the network may satisfy conditions of **Theorem 1**, so the energy of neural network can increase or decrease to sufficiently search minimum. When z_{ii} approaches zero, the neural network will only converge to a stable equilibrium point.

3. Simplification of the Continuous Hopfield Neural Network

In general, the CHNN chooses the sigmoid as activation function, i.e. $x_i = \psi(y_i) = \frac{1}{1+e^{-y_i/\varepsilon}}$ and ε is very small, which is the slope of $\psi(y_i)$. When ε is small enough, the activation function can be replaced by piecewise linear function [4, 5]. The piecewise linear function is

$$x_i = \varphi(y_i) = \begin{cases} 0 & -0.5/K > y_i \\ Ky_i + 0.5 & -0.5/K \le y_i \le 0.5/K \\ 1 & y_i > 0.5/K \end{cases} \tag{3}$$

The figures of sigmoid function and piecewise linear function are given in Fig. 1, when $\varepsilon = 1/250$ and $K = 62.5$. In Fig. 1, the real line is the figure of the sigmoid function and the broken line is the figure of the piecewise linear function. Next we will simplify the CHNN.

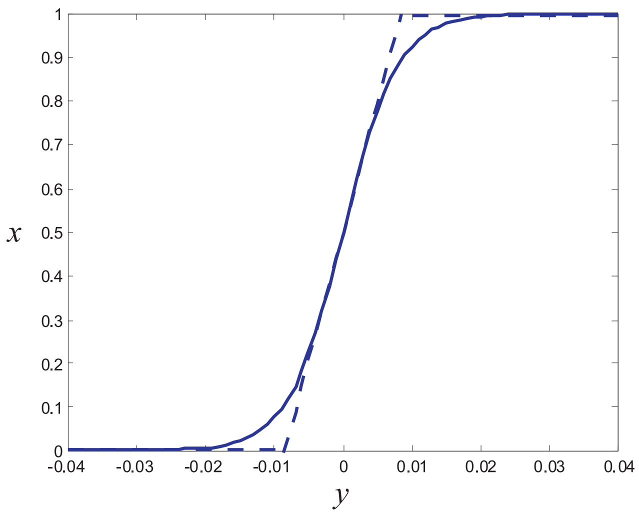

Fig. 1 *Figures of Sigmoid function and Piecewise linear function.*

After replacing the sigmoid function with the piecewise linear, the networks differential equation of the CHNN changes to

$$\begin{cases} C_i \frac{dy_i}{dt} = -\frac{y_i}{R_i} + \sum\limits_{j=1}^{n} w_{ij} x_j + I_i, \\ x_i = \varphi\left(y_i\right), \end{cases} \tag{4}$$

where $w_{ij} = w_{ji}$, $R_i > 0$, $C_i = C' \ > 0\,(i = 1, 2, \cdots, n)$, C' is constant. The CHNN energy function becomes

$$E_{CHNN} = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} x_i x_j - \sum_{i=1}^{n} x_i I_i + \sum_{i=1}^{n} \frac{1}{R_i} \int_0^{x_i} \varphi^{-1}\left(x\right) dx. \tag{5}$$

From (3), we can get

$$\frac{dx_i}{dt} = \begin{cases} 0 & -0.5/K > y_i, \\ K\frac{dy_i}{dt} & -0.5/K \le y_i \le 0.5/K, \\ 0 & y_i > 0.5/K. \end{cases} \tag{6}$$

When $-0.5/K \le y_i \le 0.5/K$, bring (6) into (4). We obtain

$$\frac{dx_i}{dt} = -\frac{1}{C' R_i} x_i + \frac{K}{C'} \sum_{j=1}^{n} w_{ij} x_j + \frac{K}{C'}\left(I_i + \frac{0.5}{R_i K}\right). \tag{7}$$

Setting

$$T_{ij} = \begin{cases} \frac{K}{C'} w_{ij} & i \ne j, \\ \frac{K}{C'} w_{ii} - \frac{1}{C' R_i} & i = j, \end{cases} \tag{8}$$

and

$$b_i = \frac{K}{C'}\left(I_i + \frac{0.5}{R_i K}\right). \tag{9}$$

Because w_{ij} is symmetric synaptic weight, T_{ij} is symmetric too. Then, the CHNN becomes

$$\frac{dx_i}{dt} = \sum_{j=1}^{n} T_{ij} x_j + b_i \quad 0 \le x_i \le 1. \tag{10}$$

Let $\mathbf{X} = \left(x_1, x_2, \cdots, x_n\right)^T, 0 \le x_i \le 1$, $\mathbf{T} = \left[T_{ij}\right]_{n\times n}$, $\mathbf{b} = \left(b_1, b_2, \cdots, b_n\right)^T$. Then, the matrix form of (10) is

$$\frac{d\mathbf{X}}{dt} = \mathbf{TX} + \mathbf{b}. \tag{11}$$

From (3) and (5), the simplified CHNN energy function is obtained as

$$\begin{aligned} E_{CHNN} &= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} x_i x_j - \sum_{i=1}^{n} x_i I_i + \sum_{i=1}^{n}\frac{1}{R_i}\int_0^{x_i}\frac{1}{K}\left(x - 0.5\right) dx \\ &= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} x_i x_j - \sum_{i=1}^{n} x_i I_i + \sum_{i=1}^{n}\frac{1}{R_i}\left(\frac{1}{2K}x_i^2 - \frac{0.5}{K}x_i\right) \\ &= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} x_i x_j - \sum_{i=1}^{n} x_i I_i + \frac{1}{2}\sum_{i=1}^{n}\frac{1}{R_i K}x_i^2 - \sum_{i=1}^{n}\frac{0.5}{R_i K}x_i \\ &= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} x_i x_j + \frac{1}{2}\sum_{i=1}^{n}\frac{1}{R_i K}x_i x_i - \sum_{i=1}^{n}\left(I_i + \frac{0.5}{R_i K}\right) x_i. \end{aligned} \tag{12}$$

Let

$$E'_{CHNN} = \frac{K}{C'}E_{CHNN} = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{K}{C'}w_{ij}x_ix_j + \\ +\frac{1}{2}\sum_{i=1}^{n}\frac{1}{C'R_i}x_ix_i - \sum_{i=1}^{n}\frac{K}{C'}\left(I_i + \frac{0.5}{R_iK}\right)x_i. \quad (13)$$

After bringing (8) and (9) into (13), we can obtain

$$\begin{aligned} E'_{CHNN} &= \frac{K}{C'}E_{CHNN} \\ &= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}T_{ij}x_ix_j - \sum_{i=1}^{n}b_ix_i. \end{aligned} \quad (14)$$

Because of $T_{ij} = T_{ji}$,

$$\frac{\partial E'_{CHNN}}{\partial x_i} = \frac{K}{C'}\frac{\partial E_{CHNN}}{\partial x_i} = -\sum_{j=1}^{n}T_{ij}x_j - b_i = -\frac{dx_i}{dt}. \quad (15)$$

Because of $C' > 0$ and $K > 0$,

$$\begin{aligned} \frac{dE_{CHNN}}{dt} &= \sum_{i=1}^{n}\frac{\partial E_{CHNN}}{\partial x_i}\frac{dx_i}{dt} = \sum_{i=1}^{n}\frac{C'}{K}\frac{\partial E'_{CHNN}}{\partial x_i}\frac{dx_i}{dt} = \\ &= \frac{C'}{K}\sum_{i=1}^{n}\frac{\partial E'_{CHNN}}{\partial x_i}\frac{dx_i}{dt} = -\frac{C'}{K}\sum_{i=1}^{n}\left(\frac{dx_i}{dt}\right)^2 \le 0. \end{aligned} \quad (16)$$

This means the dynamic neural system of simplified CHNN moves from any initial point in the state space in the direction that decreases its energy E and converges to one stable equilibrium point that is a minimum of the energy function. Convergence analysis for simplified CHNN is the same as convergence analysis for CHNN. Therefore, the method to simplify CHNN is correct and effective. The simplifying CHNN does not change the characteristics of CHNN. In next part, convergence analysis for the ADSCHNN will be given after the ADSCHNN is simplified.

4. Simplification of the ADSCHNN and Convergence Analysis

The ADSCHNN is proposed by adding an extra decaying self-feedback to every neuron of CHNN. According to (1) (8) and (10), after simplifying, the differential equation of the ADSCHNN is

$$\begin{cases} \frac{dx_i}{dt} = \sum\limits_{j=1}^{n}T_{ij}x_j + b_i + \frac{K}{C'}z_{ii}(t)\,x_i \quad 0 \le x_i \le 1, \\ z_{ii}(t) = z_{ii}(0)\,e^{-\beta t}, \end{cases} \quad (17)$$

where $\beta > 0$. According to (2), the energy function for the simplified ADSCHNN is the same as (12), i.e. $E_{ADSCHNN} = E_{CHNN}$. From (15), there is

$$\frac{\partial E_{ADSCHNN}}{\partial x_i} = \frac{\partial E_{CHNN}}{\partial x_i} = \frac{C'}{K}\frac{\partial E'_{CHNN}}{\partial x_i} = \frac{C'}{K}\left(-\sum_{j=1}^{n} T_{ij}x_j - b_i\right). \tag{18}$$

(17) becomes

$$\begin{cases} \frac{dx_i}{dt} = -\frac{K}{C'}\frac{\partial E_{ADSCHNN}}{\partial x_i} + \frac{K}{C'}z_{ii}(t)\,x_i \quad 0 \le x_i \le 1, \\ z_{ii}(t) = z_{ii}(0)\,e^{-\beta t}. \end{cases} \tag{19}$$

After discretizing (19), the optimization learning model for the ADSCHNN is got as (20)

$$\begin{aligned} x_i(k+1) &= x_i(k) + \left(-\frac{K}{C'}\frac{\partial E_{ADSCHNN}}{\partial x_i(k)} + \frac{K}{C'}z_{ii}(k)\,x_i(k)\right)\Delta t \\ &= x_i(k) + \left(-\frac{K}{C'}\frac{\partial E_{CHNN}}{\partial x_i(k)} + \frac{K}{C'}z_{ii}(k)\,x_i(k)\right)\Delta t \\ &= x_i(k) + \left(-\frac{K}{C'}\left(\frac{\partial E_{CHNN}}{\partial x_i(k)} - z_{ii}(k)\,x_i(k)\right)\right)\Delta t. \end{aligned} \tag{20}$$

From (20), **Theorem 2** is attained.

Theorem 2 When any point of the ADSCHNN is in the interior of the hyper-cube space ($\{\mathbf{X}|0 < x_i < 1,\ i = 1,2,3\cdots,n\}$) or on the border of the hyper-cube (at least one neuron equal to 0 or 1, $\{\mathbf{X}|\textit{some } x_i = 0 \textit{ or } 1 \textit{ and the other } 0 < x_i < 1\}$), the neural network is unstable.

Proof According to (20), when the point of the ADSCHNN is in the interior of the hyper-cube, because of $K > 0$ and $C' > 0$, if $\frac{\partial E_{CHNN}}{\partial x_i(k)} \neq z_{ii}(k)\,x_i(k)$, $x_i(k+1) \neq x_i(k)$ and if $\frac{\partial E_{CHNN}}{\partial x_i(k)} = z_{ii}(k)\,x_i(k)$, $x_i(k+1) = x_i(k)$. Because z_{ii} is a decaying self-feedback, $z_{ii}(k+1)\,x_i(k+1) \neq z_{ii}(k)\,x_i(k)$. So $x_i(k+2) \neq x_i(k+1)$. Therefore, any point is in the interior of the hyper-cube is unstable. At the same reason, the point on the border of the hyper-cube is also unstable.

In order to solve the TSP problem, map the solution of TSP with n cities to ADSCHNN with $n \times n$ neurons. Assume x_{ij} to be the neuron output. $x_{ij} = 1$ denotes that city i is visited in order j, while $x_{ij} = 0$ denotes that city i is not visited in order j. d_{ij} is the distance between city i and city j. A, B, C, D are the coupling parameters corresponding to the constraints and the cost function of the tour length.

When the ADSCHNN is used to solve TSP in [8], the network energy function for TSP is

$$E = \frac{A}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\substack{l=1\\ l\neq j}}^{n} x_{ij}x_{il} + \frac{B}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}\sum_{\substack{l=1\\ l\neq i}}^{n} x_{ij}x_{lj} + \frac{C}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} - n\right)^2$$
$$+\frac{D}{2}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\sum_{l=1}^{n}\left(x_{l,j+1}+x_{l,j-1}\right)x_{ij}d_{ik} + \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{R_{ij}}\int_0^{x_{ij}}\psi^{-1}(x)dx, \tag{21}$$

$A, B, C, D > 0$. When the first three terms of (21) are equal to zero, the ADSCHNN finds a valid tour. Otherwise it gets an invalid tour.

According to **Theorem 2**, the extra self-feedback makes the points are unstable, which are in the interior or on the border of the hyper-cube. The ADSCHNN only searches the result on the vertex of the hyper-cube. Because the valid tours only exist on the vertex of the hyper-cube space, the searching space for valid tours is smaller than the CHNN. Therefore, the ADSCHNN outperforms the CHNN, when they are used to solve optimization problems.

From (21), there is

$$\frac{\partial E}{\partial x_{ij}} = A\sum_{\substack{l=1\\ l\neq j}}^{n} x_{il} + B\sum_{\substack{l=1\\ l\neq i}}^{n} x_{lj} + C\left(\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} - n\right) +$$
$$+D\sum_{l=1}^{n} d_{il}\left(x_{l,j+1}+x_{l,j-1}\right) + \frac{\psi^{-1}\left(x_{ij}\right)}{R_{ij}}. \tag{22}$$

Because the output of the neurons is $n\times n$ matrix, (2) becomes

$$E = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n}\sum_{k=1}^{n} w_{ij,lk}x_{ij}x_{lk} - \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}I_{ij} +$$
$$+\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{R_i}\int_0^{x_{ij}}\psi^{-1}(x)\,dx. \tag{23}$$

So

$$\frac{\partial E}{\partial x_{ij}} = -\frac{1}{2}\sum_{l=1}^{n}\sum_{k=1}^{n} w_{ij,lk}x_{lk} - I_{ij} + \frac{\psi^{-1}\left(x_{ij}\right)}{R_{ij}}. \tag{24}$$

When piecewise linear function is used as activation function, (21) becomes

$$
\begin{aligned}
E = {} & \tfrac{A}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\substack{l=1\\ l\neq j}}^{n} x_{ij}x_{il} + \tfrac{B}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}\sum_{\substack{l=1\\ l\neq i}}^{n} x_{ij}x_{lj} + \tfrac{C}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} - n\right)^2 \\
& + \tfrac{D}{2}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\sum_{l=1}^{n}\left(x_{l,j+1}+x_{l,j-1}\right)x_{ij}d_{ik} + \\
& + \sum_{i=1}^{n}\sum_{j=1}^{n}\tfrac{1}{R_{ij}}\int_0^{x_{ij}}\tfrac{1}{K}\,(x-0.5)dx \\
= {} & \tfrac{A}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\substack{l=1\\ l\neq j}}^{n} x_{ij}x_{il} + \tfrac{B}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}\sum_{\substack{l=1\\ l\neq i}}^{n} x_{ij}x_{lj} + \tfrac{C}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} - n\right)^2 \\
& + \tfrac{D}{2}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\sum_{l=1}^{n}\left(x_{l,j+1}+x_{l,j-1}\right)x_{ij}d_{ik} + \\
& + \tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\tfrac{1}{R_{ij}K}x_{ij}^2 - \sum_{i=1}^{n}\sum_{j=1}^{n}\tfrac{0.5}{R_{ij}K}x_{ij}.
\end{aligned}
\tag{25}
$$

Let $R = R_{ij}$,

$$
\begin{aligned}
E = {} & \tfrac{A}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\substack{l=1\\ l\neq j}}^{n} x_{ij}x_{il} + \tfrac{B}{2}\sum_{j=1}^{n}\sum_{i=1}^{n}\sum_{\substack{l=1\\ l\neq i}}^{n} x_{ij}x_{lj} + \tfrac{C}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} - n\right)^2 \\
& + \tfrac{D}{2}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\sum_{l=1}^{n}\left(x_{l,j+1}+x_{l,j-1}\right)x_{ij}d_{ik} + \tfrac{1}{2}\tfrac{1}{RK}\sum_{i=1}^{n}\sum_{j=1}^{n}x_{ij}^2 - \tfrac{0.5}{RK}\sum_{i=1}^{n}\sum_{j=1}^{n}x_{ij}.
\end{aligned}
\tag{26}
$$

Therefore,

$$
\begin{aligned}
\frac{\partial E}{\partial x_{ij}} = {} & A\sum_{\substack{l=1\\ l\neq j}}^{n} x_{il} + B\sum_{\substack{l=1\\ l\neq i}}^{n} x_{lj} + C\left(\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} - n\right) + \\
& + D\sum_{l=1}^{n} d_{il}\left(x_{l,j+1}+x_{l,j-1}\right) + \tfrac{1}{RK}\sum_{i=1}^{n}\sum_{j=1}^{n}x_{ij} - \tfrac{0.5}{Rk} \\
= {} & A\sum_{\substack{l=1\\ l\neq j}}^{n} x_{il} + B\sum_{\substack{l=1\\ l\neq i}}^{n} x_{lj} + C\left(\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} - n\right) + \\
& + D\sum_{l=1}^{n} d_{il}\left(x_{l,j+1}+x_{l,j-1}\right) + \tfrac{1}{RK}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}x_{ij} - n\right) + \tfrac{1}{RK}n - \tfrac{0.5}{RK}
\end{aligned}
\tag{27}
$$

$$= A \sum_{\substack{l=1 \\ l \neq j}}^{n} x_{il} + B \sum_{\substack{l=1 \\ l \neq i}}^{n} x_{lj} + \left(C + \tfrac{1}{RK}\right) \left(\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} - n \right) + \\ + D \sum_{l=1}^{n} d_{il} \left(x_{l,j+1} + x_{l,j-1}\right) + \tfrac{1}{RK} \left(n - 0.5\right). \tag{27}$$

Let $C^{''} = C + \frac{1}{RK}$ because C, R, $K \in R$ and C, R, $K > 0$, $C^{''} > 0$. (27) becomes

$$\frac{\partial E}{\partial x_{ij}} = A \sum_{\substack{l=1 \\ l \neq j}}^{n} x_{il} + B \sum_{\substack{l=1 \\ l \neq i}}^{n} x_{lj} + C^{''} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} - n \right) + \\ + D \sum_{l=1}^{n} d_{il} \left(x_{l,j+1} + x_{l,j-1}\right) + \frac{1}{RK} \left(n - 0.5\right). \tag{28}$$

When the ADSCHNN is applied to solve TSP, the optimization learning model is

$$x_{ij}\left(k+1\right) = x_{ij}\left(k\right) + \left(-\frac{K}{C'} \left(\frac{\partial E}{\partial x_{ij}\left(k\right)} - z_{ij,ij}\left(k\right) x_{ij}\left(k\right) \right) \right) \Delta t. \tag{29}$$

x_{ij} is the component in the matrix $\mathbf{X}$. $\mathbf{X}$ is the output of the ADSCHNN, which represents the order of visiting all cities. When the ADSCHNN gets any valid tour, i.e. a stable equilibrium point, all the components of $\mathbf{X}$ for the ADSCHNN are "0" or "1". Matrix $\mathbf{X}$ has only one "1" component in the one column and has only one "1" component in the one row. The sum of the all components of matrix $\mathbf{X}$ is the city number n. For the neuron which is "1", the first two terms in (28) are equal to 0; for the neuron which is "0", the first two terms in (28) are equal to $A + B$. Therefore, when the ADSCHNN gets any valid tour, there are

(1) If the output of a ADSCHNN neuron is "0" at the k^{th} step, i.e. $x_{ij}\left(k\right) = 0$ $k = 1, 2, \cdots, n$, there is

$$\frac{\partial E}{\partial x_{ij}\left(k\right)} = A + B + D \sum_{l=1}^{n} d_{il} \left(x_{l,j+1}\left(k\right) + x_{l,j-1}\left(k\right)\right) + \frac{1}{RK} \left(n - 0.5\right) > 0. \tag{30}$$

From (29), at the $k+1^{th}$ step $x_{ij}\left(k+1\right) = 0 - \frac{K}{C'} \frac{\partial E}{\partial x_{ij}(k)} \Delta t < 0$. Because of $0 \leq x_{ij}\left(k\right) \leq 1$, $x_{ij}\left(k+1\right) = 0$. Therefore, if $x_{ij}\left(k\right) = 0$, no matter whether $z_{ij,ij}\left(k\right)$ is positive or negative, at the $k+1^{th}$ step, the output of this neuron does not change.

(2) If the output of a ADSCHNN neuron is "1" at the k^{th} step, i.e. $x_{ij}\left(k\right) = 1 \quad k = 1, 2, \cdots, n$,

$$\frac{\partial E}{\partial x_{ij}\left(k\right)} = D \sum_{l=1}^{n} d_{il} \left(x_{l,j+1}\left(k\right) + x_{l,j-1}\left(k\right)\right) + \frac{1}{RK} \left(n - 0.5\right) > 0. \tag{31}$$

(I) when $z_{ij,ij}(k) > 0$.

If at the k^{th} step $z_{ij,ij}(k) \geq \frac{\partial E}{\partial x_{ij}(k)}$, $-\frac{K}{C'}\left(\frac{\partial E}{\partial x_{ij}(k)} - z_{ij,ij}(k)\, x_{ij}(k)\right) \geq 0$. From (29), $x_{ij}(k+1) \ = 1 + \left(-\frac{K}{C'}\left(\frac{\partial E}{\partial x_{ij}(k)} - z_{ij,ij}(k)\right)\right)\Delta t \geq 1$. Because $0 \leq x_{ij}(k) \leq 1$, $x_{ij}(k+1) = 1$. Therefore, if $x_{ij}(k) = 1$, at the $k+1^{th}$ step, the output of this neuron does not change.

If at the k^{th} step $z_{ij,ij}(k) < \frac{\partial E}{\partial x_{ij}(k)}$, $-\frac{K}{C'}\left(\frac{\partial E}{\partial x_{ij}(k)} - z_{ij,ij}(k)\, x_{ij}(k)\right) < 0$. From (29), $x_{ij}(k+1) \ = 1 + \left(-\frac{K}{C'}\left(\frac{\partial E}{\partial x_{ij}(k)} - z_{ij,ij}(k)\right)\right)\Delta t < 1$. This neuron will decrease from "1". In case of $x_{ij}(k+1) = x'_{ij}$, $0 < x'_{ij} < 1$.

(A) For $x_{ij}(k+1) = 0$ case,

$$\begin{aligned}\frac{\partial E}{\partial x_{ij}(k+1)} &= Ax'_{ij} + B + C''\left(\left(n-1+x'_{ij}\right)-n\right)\\ &+D\sum_{l=1}^{n} d_{il}\left(x_{l,j+1}(k+1) + x_{l,j-1}(k+1)\right) + \frac{1}{RK}(n-0.5)\end{aligned} \tag{32}$$

or

$$\begin{aligned}\frac{\partial E}{\partial x_{ij}(k+1)} &= A + Bx'_{ij} + C''\left(\left(n-1+x'_{ij}\right)-n\right)\\ &+D\sum_{l=1}^{n} d_{il}\left(x_{l,j+1}(k+1) + x_{l,j-1}(k+1)\right) + \frac{1}{RK}(n-0.5)\,.\end{aligned} \tag{33}$$

Because of $C \leq A$,$C \leq B$ and $R, K >> 0$, $\frac{\partial E}{\partial x_{ij}(k+1)} > 0$. So $-\frac{K}{C'}\left(\frac{\partial E}{\partial x_{ij}(k+1)} - z_{ij,ij}\right.$ $\left.(k+1)x_{ij}(k+1)\right) < 0$ and $x_{ij}(k+2) = 0 - \frac{K}{C_{ij}}\frac{\partial E}{\partial x_{ij}(k+1)}\Delta t < 0$. Because of $0 \leq x_{ij}(k) \leq 1$, $x_{ij}(k+2) = 0$. At the $k+1^{th}$ step, if $x_{ij}(k+1) = 0$, the output of this neuron does not change.

(B) For $x_{ij}(k+1) = x'_{ij}$ or 1 cases,

$$\begin{aligned}\frac{\partial E}{\partial x_{ij}(k+1)} &= C''\left(\left(n-1+x'_{ij}\right)-n\right)\\ &+D\sum_{l=1}^{n} d_{il}\left(x_{l,j+1}(k+1) + x_{l,j-1}(k+1)\right) + \frac{1}{RK}(n-0.5)\,.\end{aligned} \tag{34}$$

Because of $x'_{ij} < 1$, $C''\left(\left(n-1+x'_{ij}\right)-n\right) < 0$. If at this time, $\frac{\partial E}{\partial x_{ij}(k+1)} > 0$ and $\frac{\partial E}{\partial x_{ij}(k+1)} > z_{ij,ij}(k+1)\, x_{ij}(k+1)$, $-\frac{K}{C'}\left(\frac{\partial E}{\partial x_{ij}(k+1)} - z_{ij,ij}(k+1)\, x_{ij}(k+1)\right) < 0$. At the $k+2^{th}$ step, x'_{ij} or "1" will decrease continuously. If $\frac{\partial E}{\partial x_{ij}(k+1)} \leq 0$ or $0 < \frac{\partial E}{\partial x_{ij}(k+1)} < z_{ij,ij}(k+1)\, x_{ij}(k+1)$, $-\frac{K}{C'}\left(\frac{\partial E}{\partial x_{ij}(k+1)} - z_{ij,ij}(k+1)\, x_{ij}(k+1)\right) > 0$. At the $k+2^{th}$ step, x'_{ij} will increase and reach "1". For $x_{ij}(k+1) = 1$, $x_{ij}(k+2) = 1$.

(II) when $z_{ij,ij}(k) < 0$

At the k^{th} step, because of $\frac{\partial E}{\partial x_{ij}(k)} > 0$, $-\frac{K}{C'}\left(\frac{\partial E}{\partial x_{ij}(k)} - z_{ij,ij}(k)\, x_{ij}(k)\right) < 0$. From (35), there is $x_{ij}(k+1) \ = 1 - \frac{K}{C'}\left(\frac{\partial E}{\partial x_{ij}(k)} - z_{ij,ij}(k)\right)\Delta t < 1$ at the $k+1^{th}$

step. The neuron will decrease to $x_{ij}(k+1) = x'_{ij}$, $0 < x'_{ij} < 1$. According to (1), when $x_{ij}(k) = 0$, its state does not change at the $k+1^{th}$ step, i.e. $x_{ij}(k+1) = 0$. Then, $\frac{\partial E}{\partial x_{ij}(k+1)}$ changes to (32) or (33). There is also $\frac{\partial E}{\partial x_{ij}(k+1)} > 0$. So $-\frac{K}{C'}\left(\frac{\partial E}{\partial x_{ij}(k+1)} - z_{ij,ij}(k+1)\, x_{ij}(k+1)\right) < 0$ and $x_{ij}(k+2) = 0 - \frac{K}{C'}\frac{\partial E}{\partial x_{ij}(k+1)}\Delta t < 0$. Because of $0 \le x_{ij}(k) \le 1$, $k = 1, 2, \cdots, n$, $x_{ij}(k+2) = 0$. For $x_{ij}(k+1) = x'_{ij}$ or 1, $\frac{\partial E}{\partial x_{ij}(k+1)}$ change to (34). So $z_{ij,ij}(k+1)\, x_{ij}(k+1) < \frac{\partial E}{\partial x_{ij}(k+1)} \le 0$ and $-\frac{K}{C'}\left(\frac{\partial E}{\partial x_{ij}(k+1)} - z_{ij,ij}(k+1)\, x_{ij}(k+1)\right) < 0$. Therefore, x'_{ij} or "1" will decrease continuously at the $k+2^{th}$ step. If $\frac{\partial E}{\partial x_{ij}(k+1)} < 0$ or $\frac{\partial E}{\partial x_{ij}(k+1)} < z_{ij,ij}(k+1)\, x_{ij}(k+1)$, $-\frac{K}{C'}\left(\frac{\partial E}{\partial x_{ij}(k+1)} - z_{ij,ij}(k+1)\, x_{ij}(k+1)\right) > 0$. x'_{ij} will increase at the $k+2^{th}$ step and reach "1" at the last. The outputs of neurons, which were "1" on $k+1^{th}$ step, do not change at the $k+2^{th}$ step.

From the above analysis, a theorem is obtained as follows:

Theorem 3 At the any k^{th} step, if the ADSCHNN gets a valid tour,

(1) when $z_{ij,ij}(k) > 0$

(A) If $z_{ij,ij}(k) \ge \frac{\partial E}{\partial x_{ij}(k)}$ at the k^{th} step, the valid tour is stable.

(B) If $z_{ij,ij}(k) < \frac{\partial E}{\partial x_{ij}(k)}$ at the k^{th} step and $\frac{\partial E}{\partial x_{ij}(k+1)} > 0$, $\frac{\partial E}{\partial x_{ij}(k+1)} > z_{ij,ij}$ $(k+1)\, x_{ij}(k+1)$ at the $k+1^{th}$ step, the valid tour is unstable.

(C) If $z_{ij,ij}(k) < \frac{\partial E}{\partial x_{ij}}$ at the k^{th} step and $\frac{\partial E}{\partial x_{ij}(k+1)} \le 0$, $0 < \frac{\partial E}{\partial x_{ij}(k+1)} < z_{ij,ij}(k+1)\, x_{ij}(k+1)$, the ADSCHNN will leave from the valid tour. However, the ADSCHNN will go back to the valid tour with neural network iteration.

(2) When $z_{ij,ij}(k) < 0$

(A) If $z_{ij,ij}(k+1)\, x_{ij}(k+1) < \frac{\partial E}{\partial x_{ij}(k+1)} \le 0$ at the $k+1^{th}$ step, the valid tour is unstable.

(B) If $\frac{\partial E}{\partial x_{ij}(k+1)} > 0$ or $\frac{\partial E}{\partial x_{ij}(k+1)} < z_{ij,ij}(k+1)\, x_{ij}(k+1)$ at the $k+1^{th}$ step, the ADSCHNN will leave from the valid tour. However, the ADSCHNN will go back to the valid tour with neural network iteration.

According to **Theorem 3**, when the ADSCHNN gets a valid tour, no matter what the extra self-feedbacks are, the network may move away from the result. However, the ADSCHNN will get back to the result, when some conditions are satisfied. The ADSCHNN with positive extra self-feedback may not go away from a valid tour, which is got by the network under some conditions. If the valid tour is not a global optimization, the network converges to a local optimization. Therefore, the ADSCHNN with a positive extra self-feedback more easily converges to a local optimum than the ADSCHNN with negative extra self-feedback.

When the result of the ADSCHNN is an invalid tour, the sum of the all component of a row and a column of $\mathbf{X}$ is larger than one and the sum of the all component of $\mathbf{X}$ is larger than n. There are

(1) If the output of a ADSCHNN neuron is "0" at the k^{th} step, i.e. $x_{ij}(k) = 0 \quad k = 1, 2, \cdots, n$, there is

$$\begin{aligned}\frac{\partial E}{\partial x_{ij}(k)} &= \alpha A + \beta B + \gamma C'' + D\sum_{l=1}^{n} d_{il}\left(x_{l,j+1}(k) + x_{l,j-1}(k)\right) + \frac{1}{RK}(n - 0.5)\\ &> A + B + D\sum_{l=1}^{n} d_{il}\left(x_{l,j+1}(k) + x_{l,j-1}(k)\right) + \frac{1}{RK}(n - 0.5) > 0,\end{aligned} \tag{35}$$

$\alpha, \beta, \gamma \in \{1, 2, \cdots, n\}$. From (29), $x_{ij}(k+1) = 0 - \frac{K}{C'}\frac{\partial E}{\partial x_{ij}(k+1)}\Delta t < 0$. Because of $0 \le x_{ij}(k) \le 1$, $k = 1, 2, \cdots, n$, $x_{ij}(k+1) = 0$. Therefore, the output of the ADSCHNN is "0", no matter what the $z_{ij,ij}(k)$ is, the state of the output of the ADSCHNN does not change.

(2) If the output of an ADSCHNN neuron is "1" at the k^{th} step, i.e. $x_{ij}(k) = 0$ $k = 1, 2, \cdots, n$, there is

$$\frac{\partial E}{\partial x_{ij}(k)} = \alpha A + \beta B + \gamma C'' + D\sum_{l=1}^{n} d_{il}\left(x_{l,j+1}(k) + x_{l,j-1}(k)\right) + \frac{1}{RK}(n - 0.5) > 0, \tag{36}$$

$$\alpha, \beta, \gamma \in \{1, 2, \cdots, n\}.$$

(I) when $z_{ij,ij}(k) > 0$

If $z_{ij,ij}(k) > \frac{\partial E}{\partial x_{ij}(k)}$, $x_{ij}(k+1) = 1 - \frac{K}{C'}\left(\frac{\partial E}{\partial x_{ij}(k)} - z_{ij,ij}(k)\right)\Delta t > 1$. The ADSCHNN will converge at the invalid tour. If $z_{ij,ij}(k) < \frac{\partial E}{\partial x_{ij}(k)}$, $x_{ij}(k+1) = 1 - \frac{K}{C'}\left(\frac{\partial E}{\partial x_{ij}(k)} - z_{ij,ij}(k)\right)\Delta t < 1$. x_{ij} will decrease at the $k+1^{th}$ step. Then the invalid tour is unstable.

(II) when $z_{ij,ij}(k) < 0$

There is always $\frac{\partial E}{\partial x_{ij}(k)} - z_{ij,ij}(k) > 0$. So $x_{ij}(k+1) = 1 - \frac{K}{C'}\left(\frac{\partial E}{\partial x_{ij}(k)} - z_{ij,ij}(k)\right)\Delta t < 1$. x_{ij} will decrease. The invalid tour will be unstable.

From above analysis, the following theorem is attained.

Theorem 4 If the ADSCHNN gets any invalid tour at the k^{th} step,

1. When $z_{ij,ij}(k) > 0$, if $z_{ij,ij}(k) > \frac{\partial E}{\partial x_{ij}(k)}$, the invalid tour is stable; if $z_{ij,ij}(k) < \frac{\partial E}{\partial x_{ij}(k)}$, the invalid tour is unstable.

2. $z_{ij,ij}(k) < 0$, the invalid tour is unstable.

According to **Theorem 4**, when the ADSCHNN is used to solve TSP, if the extra self-feedback is negative, all the invalid tours are unstable. However, if the extra self-feedback is positive, the ADSCHNN may converge to an invalid tour. It is the reason why the ADSCHNN with $z_{ij,ij}(k) < 0$ is more effective than the ADSCHNN with $z_{ij,ij}(k) > 0$.

5. Conclusion

This paper gives convergence analysis for the ADSCHNN in the hyper-cube space, after the simplified ADSCHNN is obtained by using the piecewise linear activation function as activation function. The analysis shows that any point in the interior or on the border of the hyper-cube space is unstable. The extra self-feedback makes the ADSCHNN only searching the optimization results at the vertex of hyper-cube space. The ADSCHNN outperforms the CHNN, when they are applied to solve an optimization problem, because the searching space is reduced. The analysis shows that when the ADSCHNN is applied to solve TSP, the ADSCHNN with negative extra self-feedback is more effective than that with positive extra self-feedback. The ADSCHNN with positive extra self-feedback more easily converges to local optima or provides an invalid tour comparing to the ADSCHNN with negative extra self-feedback. Because the extra self-feedback is reduced, the ADSCHNN will converge to an equilibrium point at the last. If some conditions are satisfied, the equilibrium point is a valid tour.

References

[1] Hopfield J. J.: Neurons with graded response have collective computational properties like those of two-state neurons, Proceedings of the National Academy of Sciences of the United States of America, **81**, 1984, pp. 3088-3092.

[2] Hopfield J. J., Tank D. W.: Neural computation of decisions in optimization problems, Biological Cybernetics, **52**, 1985, pp. 141-152.

[3] Aiyer S. V. B., Niranjan M., Fallside F.: A theoretical investigation into the performance of the Hopfield model, Neural Networks, IEEE Transactions on, **1**, 1990, pp. 204-215.

[4] Abe S., Gee A. H.: Global convergence of the Hopfield neural network with nonzero diagonal elements, Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II: Express Briefs, IEEE Transactions], **42**, 1995, pp. 39-45.

[5] Abe S.: Neural networks and fuzzy systems theory and applications: Kluwer Academic Publishers, 1997.

[6] Peng J., Xu Z.-B., Qiao H., Zhang B.: A critical analysis on global convergence of Hopfield-type neural networks, IEEE Transactions on Circuits and Systems I: Regular Papers, **52**, 2005, pp. 804-814.

[7] Li L., Yang J., Wu W.: Intuitionistic fuzzy Hopfield neural network and its stability, Neural Network World, **21**, 2011, pp. 461-472.

[8] Fei C., Qi G., Jimoh A.: Adding decaying self-feedback continous Hopfield neural networks and its application to TSP, International Review of Modelling and Simulations, **3**, 2010, pp. 272-282.