Design and Analysis of Positively Self-Feedbacked Hopfield Neural Network for Crossbar Switching

Yalan Zhou , **Jiahai Wang** , **Jian Yin**
zhouylan@163.com wjiahai@hotmail.com issjyin@mail.sysu.edu.cn
Department of Computer Science, Sun Yat-sen University,
No.135, Xingang West Road, Guangzhou 510275, P.R.China.

Summary
After the original work of Hopfield and Tank, a lot of modified Hopfield neural network models have been proposed for combinatorial optimization problems. Recently, a positively self-feedbacked Hopfield neural network architecture was proposed by Li et al. and successfully applied to crossbar switching problem. In this paper, we analysis the dynamics of the positively self-feedbacked Hopfield neural network, then show the role of the self-feedback and point out where the good performance comes from. Based on the theoretical analysis, we get better simulation results for crossbar switching problem by selecting suitably positive self-feedback value of the network.

Key words:
positively self-feedbacked Hopfield neural network, crossbar switching problem, combinatorial optimization problems.

1. Introduction

After Hopfield and Tank's works in [1] [2], the Hopfield neural network has been extensively applied in combinatorial optimization problems. It has been shown that the Hopfield neural network can compete effectively with traditional heuristics to real-world combinatorial optimization problems [3] [4]. The advantages of neural network approach in optimization are that it exploits the massive parallelism and convenient hardware implementation of the neural network architecture. Moreover, it is a common approach for solving various combinatorial optimization problems [3].

However, it has been shown that the Hopfield neural network easily causes infeasible solutions and has local minima problem [5]. Many researchers modified the Hopfield neural network to guarantee the feasibility of the solutions and help network escape from local minima for better solutions [6]. In 2005, a positively self-feedbacked Hopfield neural network was proposed by Li et al. [7]. They tried to theoretically prove the convergence of the positively self-feedbacked Hopfield neural network and applied the neural network algorithm for crossbar switching problem. Their simulation results showed that the positively self-feedbacked Hopfield neural network was much better than the previous works.

In this paper, we analysis the dynamics of the positively self-feedbacked Hopfield neural network. We find that the good performance of the positively self-feedbacked Hopfield neural network does not come from its collective computational property by introducing the positive self-feedback, rather, it comes from a slight oscillatory behavior produced by the intrinsic negative self-feedback of problem's energy function that can be counteracted by the positive self-feedback. Based on the results mentioned above, we set suitably the value of the positive self-feedback and get better solutions for crossbar switch problem.

2. Analysis of Positively Self-Feedbacked Hopfield Neural Network

2.1 Hopfield Neural Network

Hopfield et al. introduced an appropriate energy function with symmetric interconnection weight and a zero diagonal elements of the interconnection matrix in their original paper [1]:

$$E = -\frac{1}{2}\sum_{i}\sum_{j\neq i} w_{ij} v_i v_j - \sum_i b_i v_i \tag{1}$$

$$v_i = 1 \text{ if } u_i > 0 \text{ ; } 0 \text{ otherwise.} \tag{2}$$

$$u_i = \sum_{j\neq i} w_{ij} v_j + b_i \tag{3}$$

where u_i , v_i and b_i is the input, output state and threshold of neuron i , respectively, for $i = 1,\ldots,n$; w_{ij} is the interconnection weight between neurons i and j , where symmetric weights and a zero diagonal elements of the interconnection are considered, for $i, j = 1,\ldots,n$, that is, $w_{ij} = w_{ji}$ and $w_{ii} = 0$. The energy change ΔE in energy function E due to change the state of neuron i by Δv_i is

$$\Delta E = -\Delta v_i \left[\sum_{j\neq i} w_{ij} v_j(t) + b_i \right] = -\Delta v_i u_i(t+1) \tag{4}$$

According to the updating rule (2)-(3), Δv_i is positive only when the bracket is positive, and similar to the negative case. Thus any change in E under the updating rule (2)-(3) is negative.

2.2 Positively Self-Feedbacked Hopfield Neural Network

In Li et al.'s algorithm [7], additional positive self-feedbacks are added to the Hopfield neural network and total inputs of neurons are modified as following:

$$u_i = \sum_{j \neq i} w_{ij} v_j + F_i v_i + b_i \tag{5}$$

$$\Delta E = -\Delta v_i \left[u_i(t+1) - F_i v_i(t) \right] . \tag{6}$$

When the self-feedback F_i is positive or zero, changing state by the updating always leads to a reduction in the energy for the network. This result holds for the original Hopfield neural network with $F_i = 0$, which is a specific case of Eq.(6).

This proof is the same as that in the original Hopfield paper [1]. They all assumed that $w_{ij} = w_{ji}$ and $w_{ii} = 0$ in the energy function.

2.3 Updating Rule for the General Form of Energy Function

A large number of energy functions used in optimization problems [3] [4] include v_i^2 terms. These v_i^2 terms correspond to negative neuronal self-feedbacks, that is, $w_{ii} \neq 0$ holds for a large number of optimization problems.

Therefore, the general form of the energy function for optimization problems is given by

$$E = -\frac{1}{2} \sum_i \sum_j w_{ij} v_i v_j - \sum_i b_i v_i \tag{7}$$

where $w_{ij} = w_{ji}$. Note that we allow arbitrary values of the self-feedback w_{ii} for each neuron, which is different from Eq.(1).

The change ΔE in E due to chang the state of neuron k by Δv_k is

$$\Delta E = -\frac{1}{2} \sum_i \sum_j w_{ij} (v_i + \Delta v_i)(v_j + \Delta v_j)$$

$$- \sum_i b_i (v_i + \Delta v_i) + \frac{1}{2} \sum_i \sum_j w_{ij} v_i v_j + \sum_i b_i v_i$$

$$= -\frac{1}{2} \sum_i \Delta v_i \left[\sum_j w_{ji} v_j + \sum_j w_{ij} v_j + \sum_j w_{ij} \Delta v_j + 2 b_i \right]$$

$$= -\sum_i \Delta v_i \left[\sum_j w_{ji} v_j + \sum_j w_{ij} \Delta v_j / 2 + b_i \right]$$

$$(w_{ij} = w_{ji})$$

$$= -\Delta v_k \left[\sum_j w_{kj} v_j + b_k + w_{kk} \Delta v_k / 2 \right]$$

$$(\Delta v_k = \pm 1, \Delta v_j = 0 (j \neq k)) \tag{8}$$

$$\Delta E < 0 \Leftrightarrow sign\left(\sum_j w_{kj} v_j + b_k + w_{kk} \Delta v_k / 2 \right) \tag{9}$$

$$= sign(\Delta v_k) \neq 0$$

The states of binary Hopfield neural network with arbitrary values of the self-feedbacks are updated asynchronously as following:

$$\Delta v_{ij} = 1 \qquad \text{if } \sum_j w_{ij} v_j + b_i > U \text{ and } v_i = 0 ;$$

$$\Delta v_{ij} = -1 \qquad \text{if } \sum_j w_{ij} v_j + b_i < -L \text{ and } v_i = 1 ;$$

$$\Delta v_{ij} = 0 \qquad \text{otherwise;} \tag{10}$$

where $U = L = -w_{ii} / 2$.

3. Application to Crossbar Switching Problems

In packet-switched telecommunication networks, switches are located at nodes, routing randomly arriving packets so that they may be transmitted from the source to the destination. The basic switch is the crossbar switch. The crossbar switch problem is depicted by Fig.1, which shows how requests to switch packets through a $N \times N$ crossbar switch can be represented by a $N \times N$ binary request matrix r [7] [8]. Rows and columns of the matrix r are associated with inputs and outputs, respectively, of the crossbar switch. A matrix element $r_{ij} = 1$ indicates that there is a request for switching at least one packet from input line i to output line j of the switch, $r_{ij} = 0$ otherwise. If we consider the crossbar switch for point-to-point connections, then at most one crosspoint may be closed on any row or column of the switch during packed transmission. The state of the switch can be represented by a $N \times N$ binary configuration matrix c, where $c_{ij} = 1$ indicates that input line i is connected to output line j by the "closed" crosspoint (ij). $c_{ij} = 0$ indicates that crosspoint (ij) is "open". For proper operation of the switch, there should be at most one closed crosspoint in each row and each column. The throughput of the switch

is the optimal when the matrix c, which is a subset of the matrix r (i.e., $c_{ij} \le r_{ij}$ for every (i, j)), contains at most a "1" in each row/column, and has maximum overlap with r. Examples of optimal matrices are shown in Fig. 1 for a 4×4 crossbar switch [7] [8].

Each switch inlet has a queue manager. When an inlet queue manager receives a packet, it examines the packet's destination address and determines its switch outlet. It then updates the row request vector for that inlet by setting to "1", the bit corresponding to the switch outlet, and places the packet on the inlet queue. The crossbar switch is controlled by a neural network where each neuron is in correspondence to each switch crosspoint. Row request vectors from all the inlets are supplied to the neural network, which use them to compute an optimal configuration matrix for the switch. The resulting row configuration vectors are then returned to the corresponding queue manager, while the crossbar switch crosspoints selected by the computed configuration matrix are closed. Each queue manager presents to its inlet a single packet destined to the outlet selected by the row vector returned by the neural network, which thus gets routed through the closed crosspoint to its proper outlet. The queue manager also updates its row request vector by clearing the selected column bit, provided that no packets remain queued for that output. This process is iterated: new packets are received while queued packets are being transmitted. If all packets were of a constant length, then it would be possible to receive new packets, transmit selected packets, and compute the next configuration in parallel. The computation of an optimal configuration matrix should be completed in a few microseconds, which is less than it takes to transmit a packet in a high-speed fiber optic based communication system [7] [8].

The crossbar switch problem can be mapped onto Hopfield neural network with $N \times N$ neurons. The objective energy function of the crossbar switch problem is given by [7] [8] [9]

$$E = \frac{A}{2}\sum_{i=1}^{N}\left(\sum_{k=1}^{N} v_{ik} - 1\right)^2 + \frac{B}{2}\sum_{j=1}^{N}\left(\sum_{k=1}^{N} v_{kj} - 1\right)^2 \qquad (11)$$

where A and B are coefficients. v_{ik} is the output value of neurons ik and v_{kj} is the output value of neurons kj. The first term will be zero if each row contains no more than one ''1'', with all the other values being zero. Similarly, the second term is zero if each column contains no more than one ''1''. $v_{ik} = 1$ indicates that there is a request for switching at least one packet from input line i to output line k of the switch; $v_{ik} = 0$ indicates no such request. The constraints of crossbar switch problem are considered that no two packets should share the same row and the

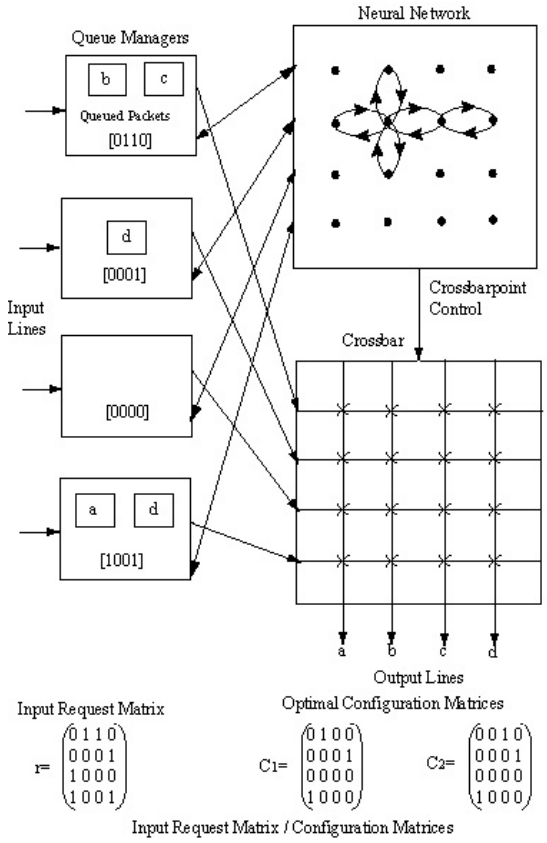

Fig.1. Architecture of crossbar control and an example of input request matrix and its optimal configuration matrices.

column of the $N \times N$ traffic matrix [v_{ij}]. Therefore the crossbar switch problem can be solved by minimizing the energy function (11) to zero.

The total input u_{ij} of the neuron is given by

$$u_{ij} = -A\left(\sum_{k=1}^{N} v_{ik} - 1\right) - B\left(\sum_{k=1}^{N} v_{kj} - 1\right) \qquad (12)$$

where the values of the self-feedback $w_{ij,ij}$ for each neuron is $w_{ij,ij} = -A - B$.

According to the updating rule (10), the state transition of the binary neuron becomes:

$$\Delta v_{ij} = 1 \quad \text{if } -A\left(\sum_{k=1}^{N} v_{ik} - 1\right) - B\left(\sum_{k=1}^{N} v_{kj} - 1\right) > U$$
$$\text{and } v_{ij} = 0;$$

$$\Delta v_{ij} = -1 \quad \text{if } -A\left(\sum_{k=1}^{N} v_{ik} - 1\right) - B\left(\sum_{k=1}^{N} v_{kj} - 1\right) < -L$$
$$\text{and } v_{ij} = 1;$$

$\Delta v_{ij} = 0$ otherwise; (13)

where $U = L = -\frac{w_{ij,ij}}{2} = \frac{1}{2}(A+B)$.

In Li et al.'s algorithm [7] for crossbar switch problem, the state transition of the binary neuron was:

$$v_{ij} = 1 \quad \text{if} \quad -A\left(\sum_{k=1}^{N} v_{ik} - 1\right) - B\left(\sum_{k=1}^{N} v_{kj} - 1\right) + F_{ij} v_{ij} > 0;$$

$v_{ij} = 0$ otherwise. (14)

that is,

$$\Delta v_{ij} = 1 \quad \text{if} \quad -A\left(\sum_{k=1}^{N} v_{ik} - 1\right) - B\left(\sum_{k=1}^{N} v_{kj} - 1\right) > U \quad \text{and } v_{ij} = 0;$$

$$\Delta v_{ij} = -1 \quad \text{if} \quad -A\left(\sum_{k=1}^{N} v_{ik} - 1\right) - B\left(\sum_{k=1}^{N} v_{kj} - 1\right) < -L \quad \text{and } v_{ij} = 1;$$

$\Delta v_{ij} = 0$ otherwise; (15)

where $U = 0$, $L = F_{ij}$.

The proof of Li et al. [7] assumed that the self-feedback value of each neuron $w_{ii} = 0$ in the energy function. But in the crossbar switch problem, the self-feedback value of each neuron is $w_{ij,ij} = -A - B$ and obviously, $w_{ij,ij} < 0$. Since the self-feedback value of each neuron is not equal to 0 in the energy function of crossbar switching problem, the proof with the assumption that the self-feedback value of each neuron is equal to 0 is not enough to explain the simulation results of crossbar switch problem.

In the update rule (15) of Li et al., $U = 0$, $L = F_{ij} = 1$, and $-w_{ij}/2 = (A+B)/2 = (1+0.5)/2 = 0.75$, therefore $U < -w_{ij}/2$, and $L > -w_{ij}/2$. The update rule (15) of Li et al. does not guarantee the energy function (11) always decreases. The network of Li et al. may oscillate and make some energy minimum states unstable, which helps the network to escape from local minimum [11].

4. Simulation Results

In order to verify the theoretical analysis, the simulation for a large number of real crossbar switch problems up to 2000×2000 switches was implemented in C on a PC. The parameters A and B were set to $A = 1$, $B = 1$. According to the updating rule for general energy function, $U = L = -w_{ij,ij}/2 = (A+B)/2 = 1$. In simulations, 100 simulations run with different randomly generated initial states were performed on each of 16 instances whose sizes ranged from 4×4 to 2000×2000. Using Eq.(13), all neurons were computed exactly once in one iteration step. The maximum iteration step was set to 1000. When the iteration steps exceeded the maximum iteration step, the network was terminated. At the same time, we also set the network with parameters $U = -w_{ij,ij}/2 = 1$, $L = 0.5$ for comparison. Note that, $L < -w_{ij,ij}/2 = 1$.

We compared our results with those found by original Hopfield neural network [1], Troudet's neural network [8], maximum neural network [10] and Li et al. [7]. Table 1 shows the results produced by original Hopfield neural network [1], Troudet's neural network [8] and maximum neural network [10]. Table 2 shows the results produced by Li et al. [7] and the proposed algorithm. The convergence rates to optimal solution ("Opt") and the average number of convergence iteration steps ("Step") are also summarized in Table 1 and Table 2.

From Table 1 and Table 2, we can see that the proposed algorithm is very effective, and is better than original Hopfield neural network [1], Troudet's neural network [8], maximum neural network [10] and Li et al. [7] in terms of the computation time and the solution quality for crossbar switching. Further, the number of iteration steps of the proposed algorithm is almost independent on the problem size, while original Hopfield neural network, Troudet's neural network, maximum neural network are somehow problem size-dependent, and even can hardly reach optimum solution to the large size crossbar switching.

In the proposed algorithm, the updating rule for general energy function can guarantee the network reach global minimum within 2 steps. The network with parameters $U = 1 = -w_{ij,ij}/2$, $L = 0.5 < -w_{ij,ij}/2$ yield more desirable oscillation, which make the network escape from local energy minima and reach the global minimum. Furthermore, we fixed parameter $U = -w_{ij,ij}/2 = 1$, and examined different values of the parameter L ($L = 0.0, 0.1, 0.2, \cdots, 0.8$), we can get the same results as those produced by the network with parameters $U = 1$, $L = 0.9$. Therefore, we can see that we can get better results by selecting suitably positive self-feedback value of the network based on our theoretical analysis.

5. Conclusions

We give the updating rule for the general form of energy function, then analyse the dynamics of the positively self-feedbacked Hopfield neural network. We show how self-feedback has effects on the network dynamics and the solution quality. Based on the theoretical analysis, we get better simulation results for crossbar switching problem by

Table 1. Simulated Results.

Crossbar Switches	Original Hopfield neural network [1]		Troudet [8]		Maximum neural network [10]	
	Opt	Step	Opt	Step	Opt	Step
4×4	51	7	100	14	100	3
6×6	43	10	98	23	100	3
8×8	31	13	95	56	100	4
10×10	28	19	96	81	100	5
20×20	28	38	40	217	100	7
30×30	31	39	19	279	100	8
50×50	37	66	0	-	100	7
80×80	37	84	-	-	100	8
100×100	34	99	-	-	100	9
200×200	30	100	-	-	100	10
300×300	-	-	-	-	97	11
500×500	-	-	-	-	97	13
800×800	-	-	-	-	94	13
1000×1000	-	-	-	-	92	13
1500×1500	-	-	-	-	91	12
2000×2000	-	-	-	-	90	14

Table 2. Simulated Results.

Crossbar Switches	Li et al. [7]		Proposed algorithm					
			$U = L = 1$		$U = 1, L = 0.5$		$U = 1, L = 0.9$	
	Opt	Step	Opt	Step	Opt	Step	Opt	Step
4×4	100	2	55	1	100	2	100	2
6×6	100	3	43	1	100	2	100	2
8×8	100	4	33	2	100	2	100	2
10×10	100	6	30	2	100	2	100	2
20×20	100	5	29	2	100	2	100	2
30×30	100	5	31	2	100	2	100	2
50×50	100	5	37	2	100	2	100	2
80×80	100	5	37	2	100	2	100	2
100×100	100	5	34	2	100	2	100	2
200×200	100	5	31	2	100	2	100	2
300×300	100	5	30	2	100	2	100	2
500×500	100	5	31	2	100	2	100	2
800×800	100	5	25	2	100	2	100	2
1000×1000	100	5	23	2	100	2	100	2
1500×1500	100	5	30	2	100	2	100	2
2000×2000	100	5	28	2	100	2	100	2

selecting suitably positive self-feedback value of the network.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (60573097) , Natural Science Foundation of Guangdong Province (05200302, 06104916), Research Foundation of Science and Technology Plan Project in Guangdong Province (2005B10101032), Specialized Research Fund for the Doctoral Program of Higher Education(20050558017), Program for New Century Excellent Talents in University of China(NCET-06-0727), and Scientific Research Foundation for Outstanding Young Teachers, Sun Yat-sen University.

References

[1] J.J. Hopfield, “Neuron with graded response have collective computational properties like those of two-state neurons,” Proc. Natl. Acad. Sci, USA, vol.81, pp.3088-3092, May 1984.

[2] J.J. Hopfield and D.W. Tank, “‘neural’ computation of decisions in optimization problems,” Biological Cybernetics,

vol.52, pp.142-152, 1985.
[3] K. Smith, M. Palaniswami, and M. Krishnamoorthy, "Neural techniques for combinatorial optimization with applications," IEEE Trans. Neural Networks, vol.9, pp.1301-1308, Nov. 1998.
[4] K. Smith, D. Abramson, and D. Duke, "Hopfield neural networks for timetabling: formulations, methods and comparative results," Computers & Industrial Engineering, vol.44, pp.283-305, 2003.
[5] G. V. Wilson and G.S. Pawley, "On the stability of the TSP algorithm of Hopfield and Tank," Biological Cybernetics, vol.58, pp.63-70, 1988.
[6] K. Smith, "Neural networks for combinatorial optimization: A review of more than a decade of research," INFORMS Journal on Computing, vol.11, no.1, 1999.
[7] Y. Li, Z. Tang, G. Xia, and R. Wang, "A positively self-feedback Hopfield neural network architecture for crossbar switching," IEEE Trans. on Circuits and systems-I: Regular Papers, vol.52, no.1, January 2005.
[8] T.P. Troudet and S.M. Walters, "Neural network architecture for crossbar switch control," IEEE Trans. Circuits syst., vol.38, pp.42-56, Jan. 1991.
[9] Y. Takefuji, Neural Network Parallel Computing. Norwell, MA: Kluwer, 1992.
[10] Y. Takenaka, K.C. Lee, and H. Aiso, "An artificial maximum neural network: A winner-take-all neuron model forcing the state of the system in a solution domain," Biological Cybernetics, vol.67, pp.243-251, 1992.
[11] S. Matsuda, "Theoretical considerations on the dynamics of hysteresis binary Hopfield networks for combinatorial optimization," International Joint Conference on Neural Networks (IJCNN2000), pp.480-485, 2000.

Yalan Zhou received the B.S. degrees in Computer Science from Guangdong University of Technology in 2003, and received the M.S. degrees in Software Engineering from South China University of Technology in 2005. She studies as a Ph.D. candidate in Department of Computer Science, Sun Yat-sen University, since 2005. Her research interests include artificial intelligence and data mining.

Jiahai Wang received B.S. degree from Gannan Teachers College, Jiangxi, China, in 1999, M.S. degree from Shandong University, Shandong, China, in 2001, and Ph.D. degree from Toyama University, Toyama, Japan, in 2005. In 2005, he joined Sun Yat-sen University, Guangdong, China, where he is currently an Instructor in the Department of Computer Science. His main research interests include optimization theory, algorithms and applications based on computational intelligence, including genetic algorithm, particle swarm optimization, neural networks.

Jian Yin received the B.S. and Ph.D. degrees in Computer Science from Wuhan University in 1989 and 1994, respectively. In 1994, he joined the Department of Computer Science, Sun Yat-sen University, where now he is a Professor. His main research interests include database and data mining, artificial intelligence and intelligent information processing.