Sains Malaysiana 39(1)(2010): 115–118

Upgrading Logic Programming in Hopfield Network
(Mempertingkatkan Logik Program dalam Rangkaian Hopfield)

SARATHA SATHASIVAM

ABSTRACT

The convergence property for doing logic programming in Hopfield network can be accelerated by using new relaxation method. This paper shows that the performance of the Hopfield network can be improved by using a relaxation rate to control the energy relaxation process. The capacity and performance of these networks is tested by using computer simulations. It was proven by computer simulations that the new approach provides good solutions.

Keywords: Energy relaxation; Little-Hopfield neural networks; program clauses

ABSTRAK

Kriteria penumpuan untuk melakukan program logik dalam rangkaian Hopfield dapat dipertingkatkan dengan menggunakan kaedah berehat yang baru. Dalam artikel ini ditunjukkan bahawa operasi rangkaian Hopfield dapat dipertingkatkan dengan menggunakan kadar rehat bagi mengawal proses relaksi tenaga. Saiz dan kadar operasi rangkaian ini di uji dengan menggunakan simulasi komputer. Dibuktikan melalui simulasi komputer kaedah baru memberikan penyelesaian yang baik.

Kata kunci: Klausa program; rangkaian Neural Little-Hopfield; santaian tenaga

INTRODUCTION

A Little-Hopfield neural network (Little 1974) minimizes a Lyapunov function, also known as the energy function due to obvious similarities with a physical spin network. Thus, it is useful as a content addressable memory or an analog computer for solving combinatorial-type optimization problems because it always evolves in the direction that leads to lower network energy. This implies that if a combinatorial optimization problem can be formulated as minimizing the network energy, then the network can be used to find optimal (or suboptimal) solution by letting the network evolve freely.

Wan Abdullah (1992) proposed a method of doing logic program on a Hopfield network. Optimization of logical inconsistency is carried out by the network after the connection strengths are defined from the logic program; the network relaxes to neural states which are models (i.e. viable logical interpretations) for the corresponding logic program. Using this method as basis, the energy landscape of a Little-Hopfield neural network programmed with program clauses is proven to be rather flat (Saratha & Wan Abdullah 2008a, b). This is supported by the very good agreement with computer simulation results for corresponding network relaxation.

In the Hopfield network, a solution of an optimization problem is obtained after the network is relaxed to an equilibrium state (Haykin 1999). This paper shows that the performance of the Hopfield network can be improved by using a relaxation rate to control the energy relaxation process.

METHOD

THE LITTLE-HOPFIELD MODEL

The Hopfield model (Hopfield 1982, 1985) is a standard model for associative memory. The Little dynamics is asynchronous, with each neuron updating their state deterministically. The system consists of *N* formal neurons, each of which is described by Ising variables $S_i(t), (i=1,2,\dots N)$. Neurons then are bipolar, $S_i \in \{-1, 1\}$, obeying the dynamics $S_i \rightarrow \mathrm{sgn}(h_i)$, where the field, $h_i = \sum_j J_{ij}^{(2)} V_J + J_i^{(1)}$, *i* and *j* running over all neurons *N*, is the synaptic strength from neuron *j* to neuron *i,* and $-J_i$ is the threshold of neuron *i*.

Restricting the connections to be symmetric and zero-diagonal, $J_{ij}^{(2)} = J_{ji}^{(2)}$, $J_{ii}^{(2)} = 0$, allows one to write a Lyapunov or energy function,

$$E = -\frac{1}{2}\sum_i \sum_j J_{ij}^{(2)} S_i S_j - \sum_i J_i^{(1)} S_i \qquad (1)$$

which monotone decreases with the dynamics.

The two-connection model can be generalized to include higher order connections. This modifies the "field" to be

$$h_i = \dots + \sum_j \sum_k J_{ijk}^{(3)} S_j S_k + \sum_j J_{ij}^{(2)} S_j + J_i^{(1)} \qquad (2)$$

where "….." denotes still higher orders, and an energy function can be written as follows:

$$E = \ldots . - \frac{1}{3}\sum_{i}\sum_{j}\sum_{k} J^{(3)}_{ijk} S_i S_j S_k -$$

$$\frac{1}{2}\sum_{i}\sum_{j} J^{(2)}_{ij} S_i S_j - \sum_{i} J^{(1)}_{i} S_i \qquad (3)$$

provided that for *i, j, k* distinct, with [...] denoting permutations in cyclic order, and for any *i, j, k* equal, and that similar symmetry requirements are satisfied for higher order connections. The updating rule maintains

$$S_i(t + 1) = \text{sgn}[h_i(t)]. \qquad (4)$$

LOGIC PROGRAMMING

In the simple propositional case, logic clauses take the form $A_1, A_2, \ldots, A_n \leftarrow B_1, \ldots, B_m$., which says that ($A_1$ or A_2 or .... or A_n) if (B_1 and B_2 and ... and B_n); they are program clauses if $n = 1$ and $m \geq 0$: we can have rules e.g. $A \leftarrow B,C$. saying $A \vee -(B \wedge C) \equiv A \vee \bar{B} \vee \bar{C}$, and assertions e.g. D←. saying that D is true.

A logic program consists of a set of program clauses and is activated by an initial goal statement. In Conjunctive Normal Form (CNF), the clauses contain one positive literal.

Basically, logic programming in Hopfield model (Altenberg 1997) can be treated as a problem in combinatorial optimization. Therefore it can be carried out in a neural network to obtain the desired solution. Our objective is to find a set of interpretation (i.e., truth values for the atoms in the clauses which satisfy the clauses (which yields all the clauses true). In other words, the task is to find 'models' corresponding to the given logic program.
The following algorithm shows how a logic program can be done in a Hopfield network based on Wan Abdullah's (1992) method.

1. Given a logic program, translate all the clauses in the logic program into basic Boolean algebraic form.
2. Identify a neuron to each ground neuron.
3. Initialize all connections strengths to zero.
4. Derive a cost function that is associated with the negation of all the clauses, such that represents $\frac{1}{2}(1 + S_x)$ the logical value of a neuron X, where S_x is the neuron corresponding to X. The value of is define in such a way that it carries the values of 1 if X is true and -1 if X is false. Negation (neuron X does not occur) is represented by $\frac{1}{2}(1 + S_x)$; a conjunction logical connective is represented by multiplication whereas a disjunction connective is represented by addition.
5. Obtain the values of connection strengths by comparing the cost function with the energy, H.
6. Let the neural networks evolve until minimum energy is reached. Check whether the solution obtained is a global solution.

The applied methodology may be summarized in the following way. Given an optimization problem, find the cost function that describes it, design a Hopfield network whose energy function must reach (one of) its minima at the same point in configuration space as the cost function, so that the stable configurations of the network correspond to solutions of the problem.

RELAXATION RATE

The quality of a solution obtained by the Hopfield network can be affected by certain factors such as parameter setting in the energy function. According to Zeng & Martinez (1999), one of the important factors which influence the quality of a solution is the difference in the frequency that a neuron receives information from other neurons

If the network relaxed too fast, there will be fewer opportunities for exchange of the information between neurons, and therefore a solution formed under this condition has poor quality. However, with a fast relaxation there will exist an inefficient use of the network due to an unnecessarily large number of iterations required to form a solution.

In order to overcome the problems described above the usage of the relaxation rate in the network dynamics is considered. The function of the relaxation rate is to adjust the speed of the relaxation so that solutions with better quality can be obtained. Precisely, the input of the neuron is updated according to the following dynamic equation:

$$\frac{dh_i^{(new)}}{dt} = R\frac{dh_i}{dt} \qquad (5)$$

where R is the relaxation rate and h_i is the local field defined in equation (2). The relaxation rate R reflects how fast the network relaxed. The value of R is an adjustable parameter and can be determined empirically.

There are two types of relaxation rates that are studied and evaluated by computer simulation. There are constant relaxation rate which is invariant through the whole relaxation process and dynamic relaxation which depends on the iteration and is chosen randomly. The main objective for analyzing dynamic relaxation rate is to explore the possibility that there exist different optimal relaxation rates at different levels during the energy relaxation process.

IMPLEMENTATION OF THEORY

Firstly, random program clauses are generated. Then, initializing initial states for the neurons in the clauses is been carried out. Next, let the network evolve until minimum energy is reached. During the energy relaxation phase, local field, h_i, is modified according to equation (5). After the network relaxed to an equilibrium sate, test the final state obtained for the relaxed neuron whether it is a stable state. If the states remain unchanged for five steps, then consider it as stable state. Following this, calculate the corresponding final energy for the stable state. If the difference between the final energy and the global minimum energy is within tolerance value, then consider the solution as global solution. Calculate the global minima ratio (number of global minima solutions/ number of iterations).

We run the relaxation for 1000 trials and 100 combinations of neurons so as to reduce statistical error. The selected tolerance value is 0.001 and the number of neurons set to 0-100 (Saratha 2006). The value of *dt* in equation (5) is set to be 10^{-5} (Zeng & Martinez 1999). All these values were obtained by trial and error, where several values were tried as tolerance values, and the selected value gives better performance than other values. For constant relaxation rate, we initialize R=2, R=4.5 and R= 0.5 (Zeng & Martinez 1999).

Results and Discussion

CONSTANT RELAXATION RATE

Simulation results using a constant relaxation rate is presented. Figure 1, 2 and 3 show global minima ratio for the network with and without relaxation rate. The network without using a relaxation rate based on equation (2) and network using a relaxation rate is based on equation (5).

DYNAMIC RELAXATION RATE

A dynamic relaxation rate with the following form is included in the network:

$$R(M) = R_0 + \frac{(M - M_0)(R_1 - R_0)}{M_1 - M_0}(M_0 \le M \le M_1) \tag{6}$$

$$R(M) = R_1 \ (M \ge M_1) \tag{7}$$

where *R(M)* is the dynamic relaxation rate which is a function of iteration *M*. *R(M)* is equal to R_0 initially (when $M = M_0 = 0$), and then increases linearly with *M* until reaching R_1 at $M = M_1$.

From Figure 4, it can be observed that when the relaxation rate has a lower value when the number of neurons are smaller, the neurons have enough chances to exchange information and to relax to global minima values. The value of R_1 is set to be 3.0 because in the previous section, we had shown that the network shows best performance between $2 \le R \le 4$. When the number of neurons increased, neurons have more time to relax to the global values by avoiding local trapped or oscillations. Due to that, it can be observed from Figure 4 that, the ratio increased as the number of neurons increased. Figure 5 shows the comparison between global minima ratio for the dynamic relaxation rate and constant relaxation rate using several different M_1 alues. The constant relaxation rate for comparison is chosen to be 4.0 because it achieves the best performance as shown in the previous section. Meanwhile, number of neurons is chosen as 100. We can see that the performance of a dynamic relaxation rate depends on the iteration. When the iteration increased, the global minima ratio also increased relatively.

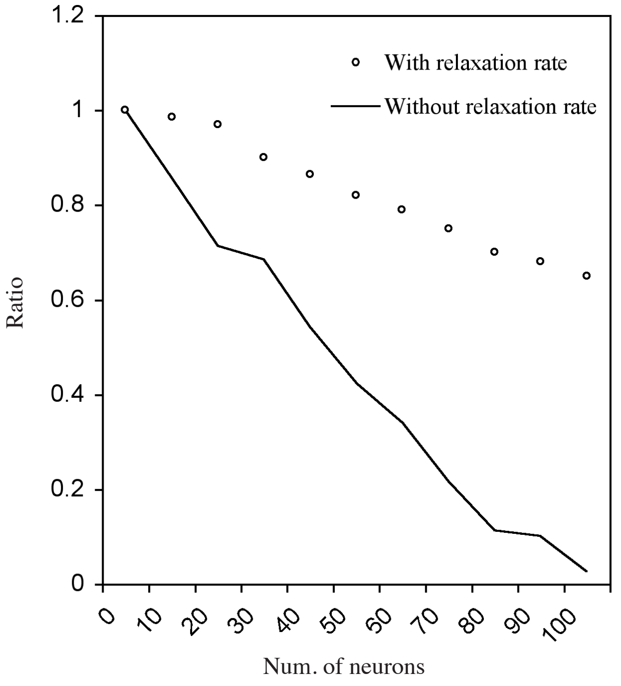

FIGURE 1. Ratio of Global Minima for R=2

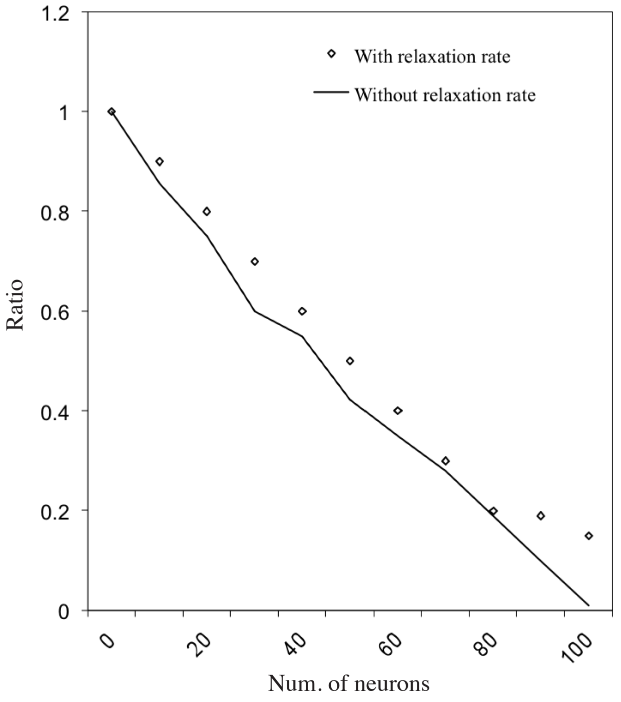

FIGURE 2. Ratio of Global Minima for R=4.5

Conclusion

In this paper, the energy relaxation process of Hopfield network in doing logic programming been analyzed. It was proven that the relaxation has an important impact on the performance of the network. A relaxation rate has been introduced into the network dynamics to control the pace of network relaxation. The network using a relaxation rate

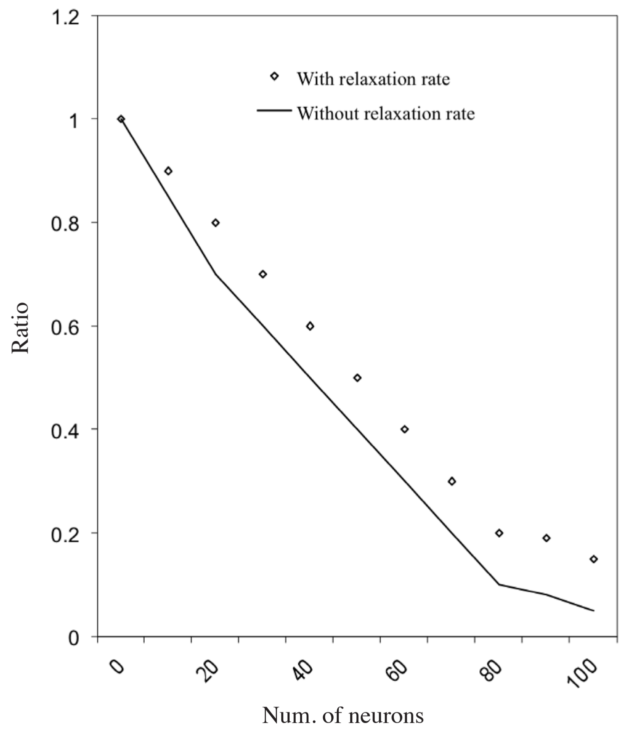

FIGURE 3. Ratio of Global Minima for R=0.5

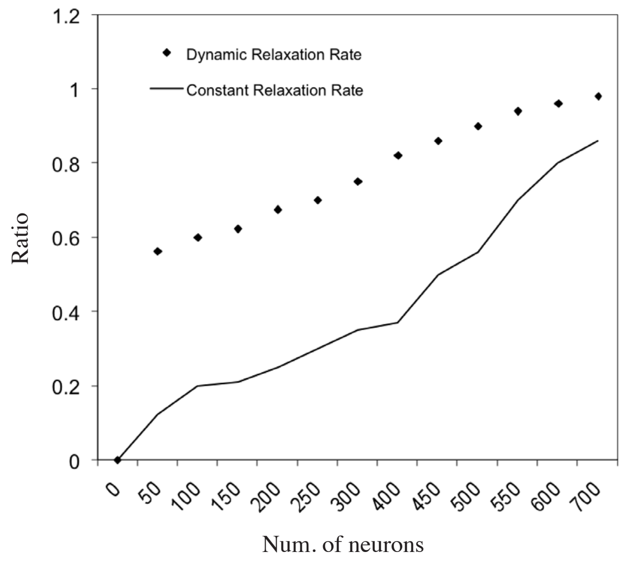

FIGURE 4. Dynamic relaxation rate with the parameters: $M_0 = 0, R_0 = 2.0, R_1 = 3.0$ and for different values of M_1

has been shown to have a better performance than without using a relaxation rate. We also showed that dynamic relaxation rate performs better than constant relaxation rate when the network gets more complex (number of iterations increased).

ACKNOWLEDGEMENT

This research was supported by FRGS grant offered by Ministry of Higher Education and RU grant by Universiti Sains Malaysia.

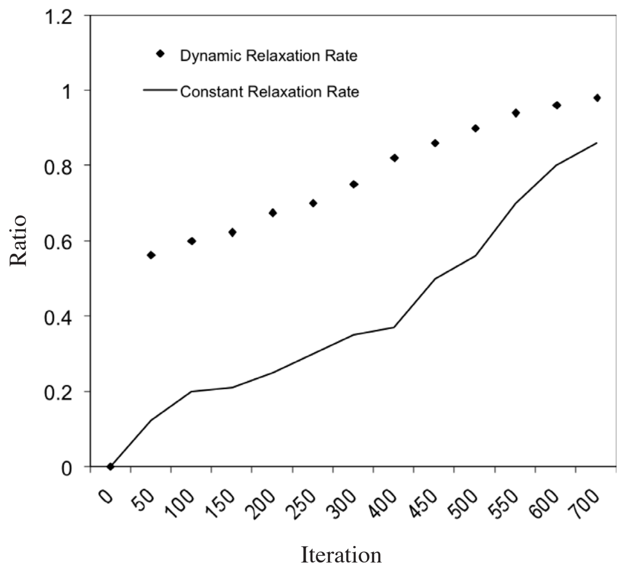

FIGURE 5. Global minima ratio comparison between dynamic relaxation rate and constant relaxation rate

REFERENCES

Altenberg, Lee. 1997. *Handbook of Evolutionary Computation*: Oxford University Press.

Haykin, S. 1999. *Neural Network: A Comprehensive Foundation*. New York: Macmillan.

Hopfield, J.J. 1985. Neural computation of decisions in optimization problems. *Biol. Cybern*. 52: 141-152.

Hopfield, J.J. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci* (USA) 79: 2554-2558.

Little, W.A. 1974. The Existence of persistent states in the brain. *Math. Biosci* 19: 101-120.

Saratha Sathasivam & Wan Abdullah, W.A.T. 2008a. Flatness of the energy landscapes of horn clauses. *MATEMATIKA* 23(2): 147-156.

Saratha Sathasivam & Wan Abdullah, W.A.T. 2008b. Logic Learning in the hopfield Networks. *Modern Applied Science* 2(3): 57-62.

Saratha Sathasivam. 2006. *Logic Mining in Neural Networks*. PhD Thesis: University of Malaya, Malaysia.

Wan Abdullah, W.A.T. 1992. Logic programming on a neural network. *Int. J. Intelligent Sys*. 7: 513-519.

Zeng, S & Martinez, T. 1999. Improving the performance of the Hopfield network by using a relaxation rate, *Proc. Int. Conference on Neural Networks and Genetic Algorithms*: 73-77.

School of Mathematical Sciences
Universiti Sains Malaysia
11800 USM, Penang
Malaysia

*Corresponding author; email: saratha@cs.usm.my

Received: 28 April 2009
Accepted: 17 June 2009