에너지 함수 분석에 의한 홉필드 신경망의 계수 결정 및 성능 개선

박지연[†]

요약

홉필드 신경망은 병렬 연산 및 일반적 응용성을 가진 장점으로 인해 여러 분야의 조합형 최적화 문제에 도입하기 위해 많은 연구가 이루어져 왔다. 그러나, 다양한 응용분야에서의 성공적 적용 사례에도 불구하고, 에너지 함수의 계수를 결정하는 실용적인 방법의 부재로 인해 대부분의 경우 실험에 의해 얻어진 값에 의존해야 하는 단점이 있다. 본 논문에서는 문제에 대한 해의 유효성을 만족하면서 모델이 최적에 근접할 수 있도록 에너지 수위를 직접 이용하여 계수를 결정하는 정형적인 기법을 제안한다. 제안한 방식은 기존의 계수 결정법과 함께 시뮬레이션을 수행하였으며, 그 결과는 제안한 접근방식이 유효함을 보여준다.

Performance Improvement with Determination of Coefficients by Energy Function Analysis in Hopfield Neural Networks

Chi-Yeon Park[†]

ABSTRACT

With its potential for parallel computation and general applicability, Hopfield network has been investigated and improved by many researchers in order to extend its usefulness to various combinatorial problems. Despite its success in many applications, determination of the energy coefficients has been based primarily on trial and error methods since no practical systematic way of finding good values has been available previously. In this paper, a methodical procedure for determining the coefficients is proposed, which utilizes direct energy level to guide the network along a path to a valid and high quality solution. Simulations on TSP comparing with other methods have shown the efficacy of the proposed approach.

키워드 : 홉필드 신경망(Hopfield neural networks), 최적화(optimization), 에너지 함수(energy function), 계수(coefficients)

1. Introduction

Since Hopfield neural network [8] was proposed as a very fast way of solving combinatorial problems with its potential for high speed, parallel computation and its apparent applicability to a wide variety of problems, many researchers [9, 17] have worked on this model to ascertain its effectiveness. In 1985, Hopfield has shown that neural network method can be applied to the TSP(Traveling Salesman Problem) which is one of the most popular NP hard problems [6]. In this network method, when the corresponding energy function is formulated for a specific optimization problem, it usually is the sum of several terms, in which each term represents either a particular constraint requirement or the objective function. The coefficient of each term influences the path on the energy surface along which the network evolves. Therefore these coefficients should be carefully chosen so that the network can reach the areas in which good solutions reside. Previously the selection of these coefficients has been based primarily on trial-and-error methods, which means that for each problem the coefficients must be re-determined by many experiments before successful application.

For the TSP with 10 cities, Hopfield and Tank determined good values for the coefficients in the energy function using an expensive trial-and-error approach. They reported impressive results [8], although others [14, 19] have gotten poor results even for the same problem with the same parameters.

[†] 정 회 원 : 관동대학교 정보기술공학부 교수
논문접수 : 2000년 7월 11일, 심사완료 : 2001년 4월 18일

On the other hand, many other researchers have proposed remedies [1, 3, 4, 13], mostly by altering the energy function, to correct the poor performance reported, since the global energy function may have a variety of different forms. But these modified forms of the energy function did not remove the difficulty of finding proper values for the parameters. One theoretical investigation of these parameters has been reported by a Cambridge research group. Aiyer et al [2]. have introduced a theoretical method of utilizing the eigenvalues of the weight matrix, to determine values for the coefficients which yield valid solutions. However, this method requires an extensive amount of preprocessing of analysis for problems.

In fact, experimental work discussed in Hedge et al [7]. indicates that good values for the coefficients may exist in very narrow, difficult-to-find regions of parameter space. As the pure Hopfield method seems to be difficult to implement successfully, some [15, 18] have employed other heuristic methods as a postprocessing stage to improve the solutions, resulting in a degradation of the benefits of a purely neural network approach.

In this paper, the effect of the weight parameters on the behavior of the network is described. By analysis of the role of these parameters, we introduce a systematic and practical way of evolving the network by adaptively adjusting the parameters utilizing the energy information of the network directly. Also, to avoid an additional computational burden in the evaluation of the energy level at each stage, an efficient incremental update of the energy has been described.

2. Effect and Analysis of Weights in Energy Functions

When an optimization problem is mapped to the network model, the total computational energy of the state of the network is defined as either

$$E = -\frac{1}{2}\sum_i\sum_j T_{ij}V_iV_j - \sum_i I_iV_i \tag{1}$$

where V_i is the output response value of neuron i, T_{ij} is the synaptic weight between neuron i and j, and I_i is the input bias to the neuron i, or in terms of multiplication of a weight matrix $\mathbf{T}$ and a response vector $\mathbf{V}$.

$$E(V) = -\frac{1}{2}V^TTV - V^TI \tag{2}$$

With this general form of the energy function, the network iterations will converge if the internal activity value of neuron i, u_i, is updated at each time step of iteration according to the equation

$$\frac{du_i}{dt} = -\frac{\partial E}{\partial V_i} \tag{3}$$

where V_i for neuron i has the range $0 \le V_i \le 1$ by applying a monotone-increasing function to the activation of input u_i. The simulation can be achieved in digital computer simply by the first order Euler method such as

$$u_i(t+\triangle t) = u_i(t) + [\sum_j T_{ij}V_j(t) + I_i]\triangle t \tag{4}$$

where the time step, $\triangle$t, is to be set small enough to keep the numerical stability.

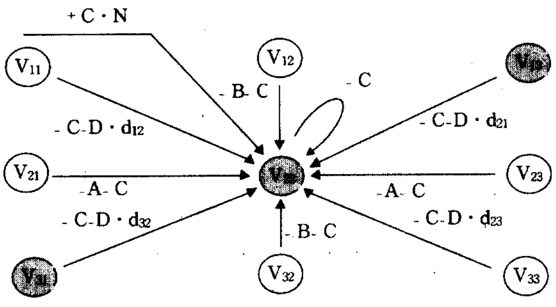
Many combinatorial optimization problems can be mapped onto a Hopfield neural network by constructing a suitable energy function and transforming the minimization of the energy function into an associated differential equation or system of differential equations. For example of TSP, the energy function was formulated as

$$\begin{aligned} E = & \frac{A}{2}\sum_x\sum_i\sum_{j\neq i}V_{xi}V_{xj} + \frac{B}{2}\sum_i\sum_x\sum_{y\neq x}V_{xi}V_{yi} \\ & + \frac{C}{2}(\sum_x\sum_i V_{xi} - N)^2 \\ & + \frac{D}{2}\sum\sum\sum d_{xy}V_{xi}(V_{y,i+1} + V_{y,i-1}) \end{aligned} \tag{5}$$

Indices x and y refer to cities, and i and j refer to positions of the cities in the tour, and the sums run from 1 to N where N is the number of cities and d_{xy} represents the 'distance' between city x and city y. The first three terms correspond to the constraints of the problem. The last term represents the cost of the solution when the network reaches a valid solution, i.e. one satisfying all constraints for the problem.

Notice that each term of Eq. (5) has its coefficient multiplied by the current energy of that term, for constraint terms and the objective term. These parameters exist for balancing the energy of each term so that the network can be guided to the region where it can attain high quality solutions. But this causes a serious difficulty when the energy function is to be formulated. Since the energy surface created by this form of energy function may have many local minima, improperly chosen parameters may cause the network to become stuck in a local minimum in which some constraints are violated or the solution obtained is of low quality.

In order to examine the behavior of each neuron, let us consider the energy function described in Eq. (5) as an example for simple 3 city TSP. After this energy function is mapped onto the neural network model, the weights to a particular neuron, say u_{22}, from the other neurons, and its bias are shown in (Fig. 1).



(Fig. 1) Weight connections to a neuron

The change of the activation of the neuron, $\triangle u_{22}$, can be easily computed according to the updating rule given in Eq. (3). Thus the amount of change of internal activity for the neuron, $\triangle u_{22}$, is decided by the all neurons connected with the given unit and the weights between them. By comparison with its total energy function, these weights are directly assigned from the coefficients in the energy function. The choice of these coefficients directly affects not only the sign of $\triangle u_{22}$, either positive or negative, but also the magnitude of the value. Each unit transmits its output value across the corresponding weight on the connection between the neurons for which the terms need to be satisfied. Usually these terms conflict with each other in the sense that the change of u_i in favor of reducing one term may increase the energy of the other terms so that the network is moving away from the region where those constraints can be satisfied. Therefore, putting too much emphasis on a particular term may result in the network moving to a region far removed from optimality for the other terms. This behavior is the cause of difficulty in determining good parameter values, since we do not know at the beginning of iterations of the network how much emphasis to place on each term in order to maintain balance during evolution.

Some alternative forms of the energy function have been devised by various researchers to obtain an improved Hopfield network method for solving the Traveling Salesman Problem, without a clear explanation of the cause of the poor performance of the original energy function given in the article of Hopfield et al. [8]. Before a different energy function is to be sought, some facets of the original energy formulation should be pointed out. With the energy function proposed in the original article, there exists an ambiguity which is not clearly explained regarding the use of such parameters, regardless of the coefficient values chosen. Suppose we have reached the state of a valid solution after some iterations, perhaps even an optimal solution in which V_{31}, V_{22}, and V_{13} are on(1) and all others are off(0). This must be a state which is one of the local minima which stabilizes the network, so that none of the neurons needs to change its internal activation. But, the neurons in the figure still have negative change of activation. For instance, the activation of a neuron, $\triangle u_{22}$, is

$$\triangle u_{22} = -D \cdot d_{21} - D \cdot d_{32} \tag{6}$$

since all V_{ij} are 0, except V_{13},V_{22}, and V_{31} which have output values of 1.

Due to this inevitable excessive inhibitory effect caused by the distance term, the network always tends to have less than N neurons ON at its final state. This situation can be verified by simple experimentation. In our experiments, usually the network has only 7, 8 or 9 neurons ON for 10 city TSP when it reaches a stable state. The simple way of overcoming this problem may be to increase the coefficient, C, in the energy function. But, increasing this value not only affects the excitatory value, i.e. the external positive bias $+CN$, but also the inhibitory value, $-C$, between all neurons at the same time. One way to increase only the excitatory value is to increase the value N by a factor of α in the energy function so that external input $\alpha \cdot CN$ can compensate for the excessive inhibition caused by the cost term. Indeed, for a 10 city TSP, the value N is set to 15 instead of 10 in the original article [8]. Another way [16] to accomplish this effect is to simply remove the self inhibition weight, T_{ii}, in which case the energy function also needs to be modified.

Regardless of the quality of solutions obtained with the network, a fundamental difficulty in this approach may arise. By modifying the value of N, the implication of the total energy function may not be the same as originally was intended, and therefore no conceptual justification can be made for the implication of the terms in the energy formulation. In addition, it results in adding one more parameter, N, to be tuned along with the energy function coefficients. The actual values chosen for A, B, C, D, N in the original paper [8] were 500, 500, 200, 500, 15, from which we see that the task is not trivial. A typical invalid final state of the network corresponds to either visiting the same city twice, or visiting two cities at the same time, which can avoid the energy cost based on the distance term.

3. Proposed Weight Adaptation and Energy Evaluation

3.1 Adaptively Adjusted Weights

In our method, to find good coefficients for the energy function, the energy of each component is traced through each epoch of iteration. At the same time, it is utilized to control the effect of each term on the evolution of the network toward the point we hope to reach. Since each constraint term has its own minimal energy level which results in satisfaction of one of the requirements for a solution of the problem, the current energy level of each term can be treated as the distance from its goal state. By comparison of the energy levels of the different terms, the network can determine how well these terms are balanced for moving toward their common goal by either competing or cooperating with others.

The basic idea of this method is that the weights are adaptively adjusted to prevent the network from following a path that is biased in favor of some terms at the expense of others. After each epoch, the new connection weights are computed depending on the current energy of each term. This results in the following formula for the new coefficient of each term.

$$C_i^* = \frac{\partial E_{total}(V)}{\partial C_i} = \frac{\partial(\sum_j C_j E_j(V))}{\partial C_i} = E_i(V) \qquad (7)$$

In early stages of the evolution of the network, these energy levels will be large, which causes the network to be somewhat unstable due to the large values obtained by direct use of Eq. (7). For this reason, the new coefficients are normalized so that the relative proportions of these values can be kept without excessive jumping around on the energy surface. The normalization is done simply as

$$C_i(normalized) = \frac{C_i^*}{\sum_j C_j^*} \qquad (8)$$

This is a steepest ascent procedure. It represents maximizing the energy function with respect to its coefficients. While this may seem counter-intuitive at first, it has the desired effect of increasing the coefficients of those terms that are contributing the most to the value of the energy function. It is those terms that most need to be reduced during network iteration, so more weight(emphasis) is placed on moving in a direction which reduces the larger terms most rapidly.

Note that the coefficient of the objective term has a difficulty in being adapted in this way. The minimal energy of the objective term can not be predicted in advance, because it is the value that the network will find as the cost of a solution of the problem. Since the lowest energy level of the objective term is expected to have some positive value for typical problems, attempting to reduce the energy of the objective term can cause the network to violate the constraint requirements. This is due to the fact that the network tries to reduce the objective energy value below the minimal value necessary for a solution. In order to overcome this situation, we must allow the network to maintain a reasonable level of energy in the objective term. The coefficient of the objective term may be specified before the network begins adaptive determination of the other coefficients, so that it always maintains an allowed weight for this energy term.

3.2 Incremental Energy Evaluation

In order to adjust the weights depending on the distance of each term from its ground level energy, the corresponding energy terms should be checked at every stage of evolution. Since the computational work required to evaluate directly the current energy level of each term of the energy function is somewhat expensive, an incremental method has been devised. It is performed during each update of the neurons. In this way, the new energy value can be simply computed without requiring a complete re-evaluation at each change of activations. The change of energy when the i-th neuron's output at time t, $V_i(t)$, is changed at time t+1 to $V_i(t+1)$ is derived in Eq. (9), which can be obtained by applying Eq. (2).

$$\begin{aligned}\triangle E &= E(t+1) - E(t)\\ &= [-\tfrac{1}{2}V^T(t+1)TV(t+1) - V^T(t+1)I]\\ &\quad -[-\tfrac{1}{2}V^T(t)TV(t) - V^T(t)I']\\ &= -\tfrac{1}{2}[V^T(t)+\triangle V e_i{}^T]T[V(t)+\triangle V e_i]\\ &\quad -[V^T(t)+\triangle V e_i{}^T]I + \tfrac{1}{2}V^T(t)TV(t) + V^T(t)I\\ &= -\tfrac{1}{2}\triangle V V^T(t)T e_i{}^T - \tfrac{1}{2}\triangle V e_i{}^T TV(t)\\ &\quad -\tfrac{1}{2}(\triangle V)^2 e_i{}^T T e_i - \triangle V e_i{}^T I\end{aligned} \qquad (9)$$

where $\mathbf{e}_i$ is a column vector having 1 as its i-th element and 0 in others. Since the weight matrix $\mathbf{T}$ is symmetric, Eq. (9) can be simplified by rearrangement to

$$\begin{aligned}\triangle E &= -\triangle V e_i{}^T TV(t) - \tfrac{1}{2}(\triangle V)^2 e_i{}^T T e_i - \triangle V e_i{}^T I\\ &= -\triangle V T_{i\cdot}\, V(t) - \tfrac{1}{2}(\triangle V)^2 T_{ii} - \triangle V I_i\end{aligned} \qquad (10)$$

where $\mathbf{T}_{i\cdot}$ is i-th row of the weight matrix $\mathbf{T}$. Again this equation can be rewritten in the summation form with indices as

$$\triangle E = -\triangle V[\sum_{j} T_{ij} V_j(t) + I_i] - \frac{1}{2}(\triangle V)^2 T_{ii} \qquad (11)$$

Note that most of the computational work in Eq. (11) has already been done when updating the neuron activation u_i, as shown in Eq. (4). The only parts requiring additional computation are the ones outside the summations. In this way, once the total energy has been computed at the initialization time of the main procedure, the new energy value can be efficiently obtained after each update of neurons.

4. Simulation and Result

Ten, twenty and thirty city problems were generated on a Euclidean space in which the distance between every pair of cities is determined by the coordinates of the cities. The data for the 10 city problem is taken from [8], which also have been used in many reports for the investigation of Hopfield network simulations. For the other two cases, we have generated cities within the unit square with a random distribution to investigate the scaling properties of neural network solutions of the TSP as the size of the problem increases.

The energy function [10, 16] that was used for our experimentation is somewhat similar to the original energy function, Eq. (5), except that the global term, *C*, has been replaced by two terms, C_3 and C_4, such that the constraints on the rows and columns of the permutation matrix are enforced separately, rather than together. The effect of this modification is to distribute the strong inhibitory effect of the cost term to all the neurons so that the network does not strongly favor an invalid state in order to reduce the objective term, representing the objective function. This energy function has the formulation

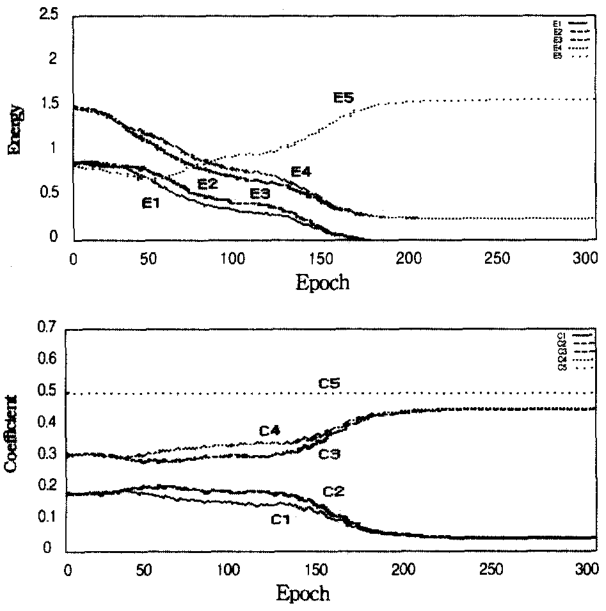
$$\begin{aligned} E = & \frac{C_1}{2}\sum_{x}\sum_{i}\sum_{j\neq i} V_{xi}V_{xj} + \frac{C_2}{2}\sum_{i}\sum_{x}\sum_{y\neq x} V_{xi}V_{yi} \\ & + \frac{C_3}{2}\sum_{x}(\sum_{j} V_{xj}-1)^2 + \frac{C_4}{2}\sum_{i}(\sum_{y} V_{yi}-1)^2 \qquad (12) \\ & + \frac{C_5}{2}\sum_{x}\sum_{y\neq x}\sum_{i} d_{xy}V_{xi}(V_{y,i+1}+V_{y,i-1}) \end{aligned}$$

The elements of the connection matrix **T** and the excitation vector **I**, implying the connection weights between neurons in the network representation, are derived such as

$$\begin{aligned} T_{xi,yj} = & -C_1\delta_{xy}(1-\delta_{ij}) - C_2\delta_{ij}(1-\delta_{xy}) - C_3\delta_{xy} - C4\delta_{ij} \\ & - C_5 d_{xy}(\delta_{j,i+1}+\delta_{j,i-1}) \qquad (13) \\ I_{xi} = & C_3 + C_4 \end{aligned}$$

where δ_{ij} is 1 if $i=j$, 0 otherwise.

In our experiment, the coefficient of the objective term is specified before the network begins adaptive determination of the other coefficients, such that it always maintains this energy term. Then the network was allowed to adapt all other coefficients as it evolved. The evolution of the other coefficients with several different settings of C_5 has been investigated. As the coefficient of the objective term is set larger, the quality of solutions found gets better but the probability of reaching a valid state is lowered, as expected. A proper value for this coefficient, in the sense of the percentage of valid solutions produced, shows a slight difference according to the size of the problem. Empirical study shows that for larger problems, this value had to be set lower than for smaller problems. To be able to produce a similar rate of valid solutions, say 50%, the 30 city problem required a value of 0.4, while the 10 city problem required a value of 0.7. With C_5 set to 0.5 for the 10 city TSP, a typical curve for the evolution of the other coefficients with normalization is depicted in (Fig. 2). It is apparent that C_1 and C_2 tend toward a common value, as do C_3 and C_4. This is due to the similar role of the terms in the energy function, i.e. row and column inhibitory terms, and row and column excitatory terms. Using different fixed values of C_5, sets of values for the other coefficients have been obtained, as given in <Table 1>.



(Fig. 2) Evolution of energy and coefficient of each term

As described in Fig. 1., the output signal of each neuron can not increase to 1 (ON) exactly without adding some extra excitatory stimulus, because of the strongly inhibitory effect caused by a positive objective term energy. Instead, the neurons compensate for this by relaxing the energy of each constraint term. As a result, it is seen that the coefficients, C_1 and C_2, with only inhibitory terms get relatively smaller values, while the coefficients, C_3 and C_4, the only excitatory terms, get relatively larger values. Thus, quite small inhibitory weights are enough, since inhibitory weights are intrinsically stronger. A sample of some sets of coefficients obtained is shown in <Table 1>.

〈Table 1〉 Coefficients obtained for different size of problems

Problem Size	Objective	Inhibitory		Excitatory	
	C_5	C_1	C_2	C_3	C_4
10	0.7	0.065	0.079	0.421	0.435
20	0.5	0.097	0.119	0.381	0.403
30	0.4	0.107	0.132	0.368	0.393

Since the final adapted values of the coefficients are those which balance the network, the network is rerun using these final values as fixed coefficients. During simulation, different random initializations of the neurons result in different solutions, which is to be expected, since the network is trying to reach the bottom of the basin of attraction from which it starts. For each size of problem studied, the 10, 20, and 30 city TSP, 3 different methods were applied, one with adaptively obtained coefficients except for the objective coefficient, one with unit coefficients ($C_1, C_2, C_3, C_4 = 1$) except for the objective coefficient, and one with Hopfield and Tank's energy function and their coefficient values. For each method, 100 runs of different neuron-initializations were made with several different objective coefficients, except for Hopfield's coefficients. To check the validity of solutions obtained, a binary decision criterion is used, i.e. if a neuron's output is greater than a specified threshold then it is regarded as ON, otherwise it is regarded as OFF. In our observations, since the network is following a stable path to a minimum, either local or global, once the state of the network has entered a valid solution region under the binary criterion, the network does not then visit a different valid solution state.

The data corresponding to solutions obtained by the 3 different methods are listed in <Table 2>. For the method of adaptive coefficients, all other coefficient values are found by the network itself except for different fixed values of the objective coefficient. For the method of unit coefficients, each initialization is repeatedly tried with increasingly larger (but fixed for each trial) values of the objective coefficient, until valid solutions are no longer possible because of the dominance of the objective term over the constraint terms in the energy function. For the method of Hopfield's energy function, 15% of trials have produced valid solutions for the 10 city TSP and no valid solutions were obtained for the 20 and 30 city TSP, because of the high sensitivity of the network to the parameter values, as reported by other investigators [11, 20]. As expected, the network tended to produce valid solutions with lower probability but of higher quality as the objective coefficient was increased. The comprehensive results obtained for other sets of 10, 20, and 30 city TSP have shown similar results.

〈Table 2〉 Solutions obtained by each method

Method	Problem size	Objective Coefficient	Valid Rate	Shortest Length	Longest Length	Average
Adaptive	10	0.7	56%	2.69	3.04	2.76
	20	0.5	58%	4.04	5.57	4.70
	30	0.4	43%	4.93	8.10	6.47
Unit	10	2.0	41%	2.78	3.56	3.13
	20	1.5	38%	4.93	6.77	5.86
	30	1.5	41%	6.94	9.55	8.12
Hopfield	10	500	15%	2.78	3.78	3.40
	20	500	0	N/A	N/A	N/A
	30	500	0	N/A	N/A	N/A

For ease of comparison, the data presented for each method are based on the case for which valid solutions were obtained approximately 50% of the time. It is seen that the solutions with the adaptively obtained coefficients produced much higher quality solutions than the other two methods in all cases.

5. Summary and Discussion

The basic problem of the formulation of the energy function for combinatorial optimization problems is analyzed with the example of the Traveling Salesman Problem. An adaptive computation scheme for determining the coefficients has been devised that makes use of this improved energy function to obtain good coefficients which can maintain balance among the energy terms during evolution of the network. In this way, we obtained good coefficient values which helped the network to find not only valid but also high quality solutions. The Hopfield type of neural network approach to solving combinatorial problems can be made more practical by avoiding the tedious preprocessing task of finding the proper coefficients of the energy function. To reduce the additional computational expense needed to trace the energy level of the network at every stage in order to update the coefficients in our method, an efficient incremental update has been devised, which utilizes the computations done previously for the update of neuron activations.

In spite of the impressive improvements over previous reports on the Hopfield network, there are some other issues to point out. To be more practical, the performance in the sense of solution quality needs to be improved to be more competitive with currently available problem-tailored methods [11] for TSP. A possible future research direction for combinatorial optimization problems would be an extension of our approach to make all network coefficients fully adaptive, including the objective coefficient, by compensating for excessively inhibitory weight connections. This may be achieved only when the energy function is devised in such a way that its global minimum corresponds exactly to the state of an optimal solution.

References

[1] Abe, S., "Theories on the Hopfield Neural Networks," International Joint Conference on Neural Networks, Vol.1, pp.557-564, 1989.

[2] Aiyer, S. V. B., Niranjan, M., and Fallside, F. "A Theoretical Investigation into the Performance of the Hopfield Model," IEEE Trans. Neural Networks, Vol.1, No.2, pp.204-215, 1990.

[3] Bizzarri, A. R. "Convergence Properties of a Modified Hopfield-Tank Model," Biological Cybernetics, Vol.64, pp.293-300, 1991.

[4] Brandt, R. D., Wang, Y., Laub, A. J., and Mitra, S. K. "Alternative Networks for Solving the Traveling Salesman Problem and List-Matching Problem," IEEE International Conference on Neural Networks, Vol.2, pp.333-340, 1988.

[5] Cichochki, A., and Unbehauen, R. "Neural Networks for Optimization and Signal Processing," John Wiley & Sons, 1993.

[6] Garey, M. R., and Johnson, D. S., "Computers and Intractability : A Guide to the Theory of NP-Completeness," Freeman, San Fransisco, 1979.

[7] Hedge, S. U., Sweet, J. L., and Levy, W. B., "Determination of Parameters in a Hopfield Tank Computational Network," International Conference on Neural Networks, Vol.2, pp.291, 1988.

[8] Hopfield, J. J., and Tank, D. W. "Neural Computation of Decisions in Optimization Problems," Biological Cybernetics, Vol.52, pp.141-152, 1985.

[9] Jagota, A. "Approximating Maximum Clique with a Hopfield Network," IEEE Transactions on Neural Networks, Vol.6, No.3, pp.724-735, 1995.

[10] Kamgar-Parsi, B., and Kamgar-Parsi, B. "On Problem Solving with Hopfield Neural Networks," Biological Cybernetics, Vol.62, pp.415-423, 1990.

[11] Lin, S., and Kernighan, B. W. "An Effective Heuristic Algorithm for the Traveling Salesman Problem," Operations Research, Vol.21, pp.498-516, 1973.

[12] Lin, W., Delgado-Frias, J. G., Pechanek, G. G., and Vassiliadis, S. "Impact of Energy Function on a Neural Network Model for Optimization Problems," IEEE International Conference on Neural Networks, Vol.7, pp.4518-4523, 1994.

[13] Matsuda, S. "Theoretical Characterizations of Possibilities and Impossibilities of Hopfield Neural Networks in Solving Combinatorial Optimization Problems," IEEE International Conference on Neural Networks, Vol.7, pp.4563-4566, 1994.

[14] Paielli, R.A. "Simulation tests of the optimization method of Hopfield and Tank using neural networks," NASA Technical Memorandum 101047, 1988.

[15] Shirai, H., Ishigame, A., Kawamoto, S., and Taniguchi, T., "A Solution of Combinatorial Optimization Problem by Uniting Genetic Algorithms with Hopfield's Model," IEEE International Conference on Neural Networks, Vol.7, pp.4704-4709, 1994.

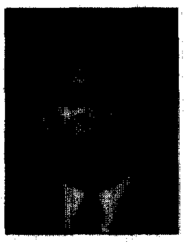
[16] Szu, H. H. "Fast TSP Algorithm Based on Binary Neuron Output and Analog Neuron Input Using the Zero-Diagonal Interconnect Matrix and Necessary and Sufficient Constraints of the Permutation Matrix," IEEE International Conference on Neural Networks, Vol.2, pp.259-265, 1988.

[17] Takefuji, Y. "Neural Network Parallel Computing," Kluwer Academic Publishers, 1992

[18] Xu, X., and Tsai, W. T. "Effective Neural Algorithms for the Traveling Salesman Problem," Neural Networks, Vol.4, pp.193-205, 1991.

[19] Wilson, G. V., and Pawley, G. S., "On the Stability of the Traveling Salesman Problem Algorithm of Hopfield and Tank," Biological Cybernetics, Vol.58, pp.63-70, 1988.

[20] Zak, M. "Terminal Attractors for Addressable Memory in Neural Networks," Physics Letter A, Vol.133, pp.18-12, 1988.



박지연

e-mail : cypark@mail.kwandong.ac.kr

1985년 경희대학교 기계공학과 졸업(학사)

1987년 플로리다공과대학교 전자계산학과 (석사)

1996년 플로리다공과대학교 전자계산학과 (박사)

1989년~1992년 LG-EDS 연구원

1996년~1997년 삼성SDS 책임연구원

1997년~현재 관동대학교 정보기술공학부 부교수

관심분야 : 인공지능, 신경망, 알고리즘, 영상처리