PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Determination of adaptively adjusted coefficients for Hopfield neural networks utilizing the energy function

Chiyeon Park
Donald W. Fausett

Determination of adaptively adjusted coefficients
for Hopfield neural networks
utilizing the energy function

Chiyeon Park and Donald W. Fausett*

Florida Institute of Technology
Computer Science Program
*Applied Mathematics Program
150 West University Boulevard
Melbourne, Florida 32901-6988

ABSTRACT

With its potential for parallel computation and general applicability, the Hopfield neural network has been investigated and improved by many researchers in order to extend its usefulness to various combinatorial problems. In spite of its success in several applications with different energy function formulations, determination of the energy coefficients has been based primarily on trial and error methods since no practical and systematic way of finding good values has been available previously, although some theoretical analyses have been presented. In this paper, we present a methodical procedure which adaptively determines the energy coefficients leading to a valid solution as the network evolves. This method directly utilizes the value of each competing term in the energy function to balance the coefficients at each stage of the computation of the network. The advantage of this method is that the system itself controls the amount of energy which each term contributes to the total energy. To demonstrate the effectiveness of this approach, the N-Queens problem (a well known example of a constraint satisfaction problem) is studied and verified. Also, an inexpensive method for computation of the new energy level of each term at each stage of iteration is described based on incremental updating.

Keywords: adaptive coefficients, energy function, Hopfield network, N-Queens problem

1. INTRODUCTION

In 1985, Hopfield and Tank first showed that some combinatorial optimization problems, specifically the Traveling Salesman Problem in the original article[4], can be solved by their neural network model, which intrinsically has the potential of parallel computation. In this network model, the problems to be solved are represented by the neurons in a net which is fully interconnected, in the sense that each unit is connected to every other unit, with predefined strengths of synapses between the neurons.[2] These symmetric weights are determined when the problem is represented by the corresponding form of the energy function, which is to be minimized as the network moves along the energy surface. The energy functions for optimization problems are often of a form such as:

$$Energy = Term_{CON1} + Term_{CON2} + \ldots + Term_{OBJ} , \tag{1}$$

0-8194-2141-3/96/$6.00

where $Term_{CON1}$, $Term_{CON2}$, and $Terms_{OBJ}$ are the first constraint, second constraint, and objective terms, respectively in the total energy of a combinatorial optimization problem. After these terms are represented by weights on the synapses of the network, the energy function can be expressed via the network model:

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij}V_iV_j - \sum_{i=1}^{N} V_iI_i \quad , \tag{2}$$

where T_{ij} is the weight between neuron i and j, I_i is the input bias, V_i is the output of neuron i, and N is the number of neurons in the network for the problem. Then, the network changes its state until it reaches a stable state, where the total energy is minimized, by updating its neurons according to the equation of motion:

$$\frac{dU_i}{dt} = -\frac{\partial E}{\partial V_i} \quad , \tag{3}$$

where V_i is an output value of a non-decreasing monotonic function g, usually a hyperbolic tangent, applied to the activation value, U_i, of neuron i: $V_i = g(U_i)$.

Since this network was shown to be effective for solving optimization problems with proper energy forms, the Hopfield networks have received quite a lot of attention, resulting in many other applications, although some difficulties[8] have been found. One of these difficulties is the determination of numerical values for the multipliers appearing in all terms in the energy function, which may directly influence the final state of the network in terms of the validity and the quality of solutions being sought. Some alterations[5] of the energy formulation have been suggested to overcome this difficulty, although without great success because they still require trial-and-error determination of multiplier values. While a successful theoretical investigation[1] has been conducted by analyzing the eigenvalues of the weight matrix, it did not offer a practical method for computing effective values of the coefficients, covering many possible applications with full generality.

In our previous work[6], one method of finding good coefficients with adaptation has been devised and applied to the Traveling Salesman Problem, which showed the efficacy of the approach suggested. The major difficulty encountered with the method was caused by the coefficient of the objective term, representing the tour distance when the network reached a valid solution. Its value could not be adaptively determined since the expected final energy of this term could not be predicted, unlike the constraint terms whose final energy should become zero upon reaching a valid solution. In this paper, we will extend this method of adaptively adjusting the coefficients of a Hopfield network to the constraint satisfaction problem in a straightforward way, using the non-attacking N-Queens problem as an example, which can be found in some reports[7].

2. ADAPTIVE HOPFIELD NETWORK APPLIED TO N-QUEENS PROBLEM

The N-Queens problem is a well known example of a constraint satisfaction problem, a class of problems that involves assigning values to variables subject to a set of binary constraints. When this problem is represented in terms of a Hopfield network, the computational energy function has several terms which must be minimized in order for the network to reach a feasible solution. Because of the high sensitivity of the network to the coefficients[3] of these constraint terms, the parameters should be carefully tuned to allow the network to proceed to a valid solution. In other words, over-emphasis of one term by giving its coefficient a larger value may guide the network to a state which satisfies the corresponding constraint too much in advance of the minimization of the other terms, which results in the network being more likely to become stuck in local minima. In order to avoid this situation our method evaluates the current energy of each term at every stage and determines the amount of emphasis to be put on each term by adaptively adjusting the coefficients.

In the proposed method, to find good coefficients for the energy function, the energy of each component is traced through each 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. In this way, according to the distance to the expected minimum value of each term, the corresponding coefficients are either relatively emphasized by increasing the value, or de-emphasized by decreasing the value, thus maintaining a balance between all terms until they reach a state of equilibrium. To achieve this, the new coefficient of each term is computed as follows:

$$Coefficient_i(t+1) = \frac{Energy_i(t)}{\sum_j Energy_j(t)}, \tag{4}$$

where $Coefficient_i(t+1)$ and $Energy_i(t)$ are the new value of the coefficient of i-th term at time $t+1$ and the energy of i-th term at time t. In this way, the energy of each term is proportionately reduced, keeping it in balance with the other terms as the total energy is minimized, whereby all constraints are satisfied.

2.1. Problem definition and representation

The goal of the N-Queens problem is to place N queens on an N X N chessboard such that no queen attacks any of the others. This is equivalent to placing the N queens so that:

a) one and only one queen is placed in each row;
b) one and only one queen is place in each column;
c) at most one queen is placed on each diagonal.

The positions that are under attack by placing a queen on a square is illustrated in Fig. 1. The position occupied by the queen is the dark square, the positions under attack are the gray squares, and the positions not under attack are the white squares.

With the Hopfield network, this problem is represented by an N by N matrix of neurons in which a solution is achieved when the state of the network satisfies all constraints. These constraints are represented by the terms in the corresponding energy function for this problem. One possible energy formulation we can use is:

$$E = \frac{A}{2}\sum_x \sum_i \sum_{j \neq i} V_{xi}V_{xj} + \frac{B}{2}\sum_i \sum_x \sum_{y \neq x} V_{xi}V_{yi} + \frac{C}{2}\sum_x \left(\sum_j V_{xi} - 1 \right)^2 + \frac{D}{2}\sum_i \left(\sum_y V_{yi} - 1 \right)^2$$

$$+ \frac{F}{2}\sum_x \sum_i \sum_{\substack{1 \le x+k,\, i+k \le N \\ k \neq 0}} V_{xi}V_{x+k,\, i+k} \; + \; \frac{G}{2}\sum_x \sum_i \sum_{\substack{1 \le x+k,\, i-k \le N \\ k \neq 0}} V_{xi}V_{x+k,\, i-k} \tag{5}$$

When the energy is minimized, the network is expected to have reached a representation of a valid solution of the N-Queens problem. The implication of the minimization of each term is as follows. The first and third terms together enforce constraint (a). The second and fourth terms together enforce constraint (b). The fifth and sixth terms enforce constraint (c), one term for each direction of the diagonals. All of these constraints can be satisfied simultaneously when the network reaches a feasible solution.

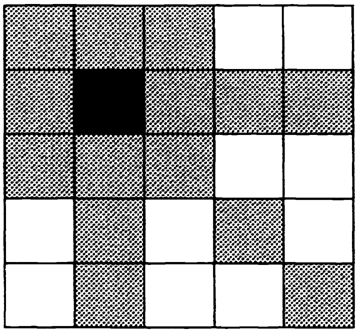

Figure 1. Squares under attack by a queen

2.2. Network Evolution with Adaptively Adjusted Coefficients

In order to represent the N-Queens problem in the network model, the connection weights on synapses between neurons are set with the values transformed from the energy function in Eq. (5). Then the network evolves until it reaches a stable point. The overall procedure is as follows:

a) set the initial activation of the neurons U_{xi} so that $\sum V_{xi} \approx N$ (with some small noise added) ;

b) compute initial $Energy_i$, the energy of each term ;

c) repeat (c-1)-(c-3) until the network is stabilized:
i.e. $TotalEnergy(t+1) - TotalEnergy(t) < \varepsilon$;

c-1) update the activation of each neuron using Euler's first order difference equation

$$U_{xi}(t + \Delta t) = U_{xi}(t) + \frac{dU_{xi}}{dt}\Delta t \tag{6}$$

and the motion equation Eq. (3)

$$\frac{dU_i}{dt} = -\frac{\partial E}{\partial V_i}$$

$$= -A\sum_{j\neq i} V_{xj} - B\sum_{y\neq x} V_{yi} - C\left(\sum_j V_{xj} - 1\right) - D\left(\sum_y V_{yi} - 1\right)$$

$$-F \sum_{\substack{1\leq x+k,\, i+k\leq N \\ k\neq 0}} V_{x+k,\, i+k} \;-\; G \sum_{\substack{1\leq x+k,\, i-k\leq N \\ k\neq 0}} V_{x+k,\, i-k} \tag{7}$$

c-2) compute new energy of each term, $Energy_i$. Note that update of the energy of each term can be obtained in an incremental method as follows.

$$E(t+1) - E(t) = \Delta V_{xi}\left(A\sum_{j\neq i} V_{xj}\right) + \Delta V_{xi}\left(B\sum_{y\neq x} V_{yi}\right)$$

$$+ \Delta V_{xi}\left[C\left(\sum_j V_{xj} - 1\right) + \frac{C}{2}\Delta V_{xi}\right] + \Delta V_{xi}\left[D\left(\sum_y V_{yi} - 1\right) + \frac{D}{2}\Delta V_{xi}\right]$$

$$+ \Delta V_{xi}\left(F \sum_{\substack{1\leq x+k,\, i+k\leq N \\ k\neq 0}} V_{x+k,\, i+k}\right) + \Delta V_{xi}\left(G \sum_{\substack{1\leq x+k,\, i-k\leq N \\ k\neq 0}} V_{x+k,\, i-k}\right) \tag{8}$$

where $\Delta V_{xi} = V_{xi}(t+1) - V_{xi}(t)$

c-3) update the new coefficient of each term using Eq. (4)

Computation of the new energy of each term by the incremental method avoids additional computation burden since most of terms in Eq. (8) are already computed when updating the neuron activation $U_{xi}(t+1)$, such as in Eq. (6) and Eq. (7). The only parts requiring additional computation are the ones outside the summations in the third and fourth terms of Eq. (8).

3. SIMULATION AND RESULT

The network with the proposed adaptive coefficients has been applied to the 8-Queens problem. After each iteration, the coefficients are updated with normalization as in Eq. (4) so that A+B+C+D+F+G=1. The hyperbolic tangent function, $0.5\,[1 + \tanh\,(U_{xi}/U_0)]$, is employed for the output function, where U_0 is set to 0.02. The value of U_0 determines the steepness of the slope of the output function. The size of the time step, Δt, is set to 0.01, which is small enough to keep the network stable. Too large of a time step can reduce the convergence time, but increase the possibility of missing a solution state.

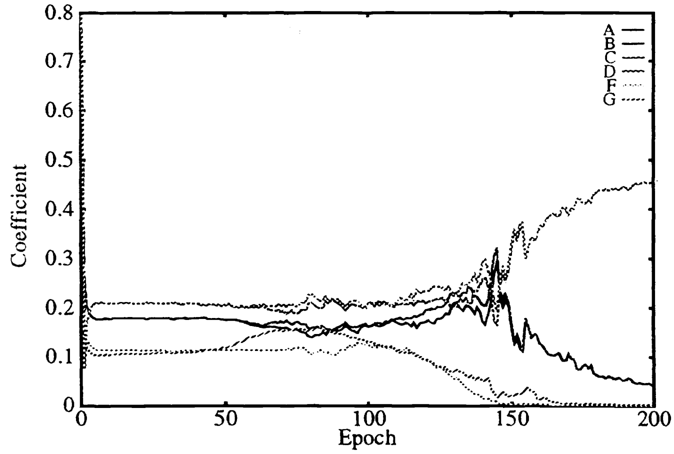

Figure 2. Adapted Coefficients

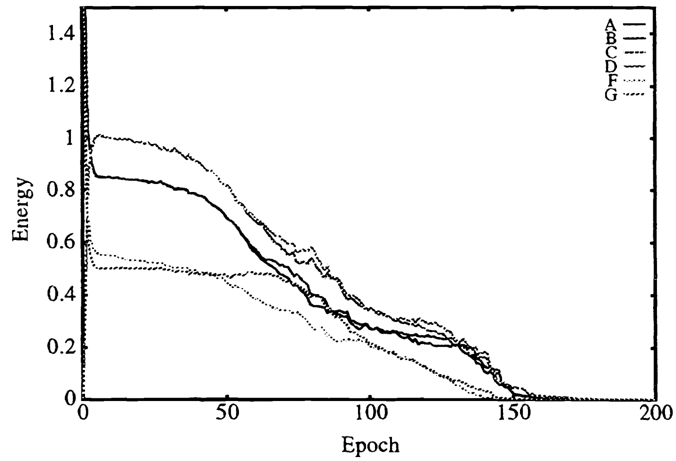

Figure 3. Energy of Each Term

Fig. 2 shows the values of the adjusted coefficients during the evolution of the network along with the corresponding energy level of each term in Fig. 3. From the energy curve, it is seen that the energy levels of all terms are being lowered while keeping the balance between them by pushing the term of higher energy with more force until the network reaches a state of solution. The convergence of the total energy is depicted in Fig. 4.

In our experiments, the network produced valid solutions in more than 30% of the cases with different random initializations of neurons of the network. Fig. 5 shows an example of a solution produced by the network for the 8-Queens problem.

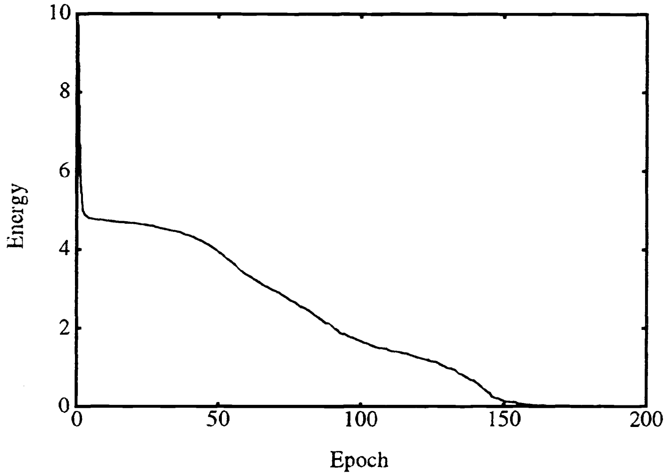

Figure 4. Convergence of Total Energy

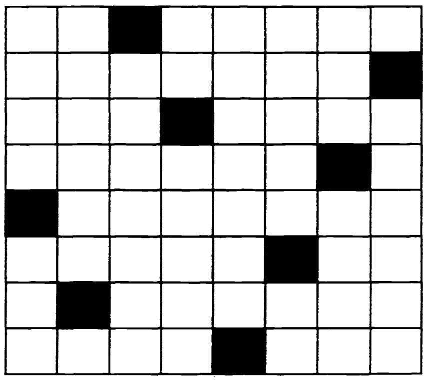

Figure 5. A Solution of 8-Queens Problem

4. SUMMARY

We have shown a practical way of using adaptive coefficients in a Hopfield neural network for solving constraint satisfaction problems. In this approach, the network can be guided in the direction of a solution state by controlling the level of each energy term by itself. In order to see the efficacy of this proposed method, the N-Queens problem has been used as an example, and verified. We also have obtained some promising results by applying a similar approach to the Map-Coloring Problem. Those results will be reported in a future publication.

REFERENCES

1. Aiyer, S. V. B., Niranjan, M., and Fallside, F. (1990). "A Theoretical Investigation into the Performance of the Hopfield Model", IEEE Trans. Neural Networks, 1 (2), 204-215
2. Fausett, L. (1994). *Fundamentals of Neural Networks*, Prentice Hall.
3. Hegde, S. U., Sweet, J. L., and Levy, W. B. (1988). "Determination of Parameters in a Hopfield/Tank Computational Network", ICNN, Vol.2, 291-298.
4. Hopfield, J. J., and Tank, D. W. (1985). "Neural Computation of Decisions in Optimization Problems", Biological Cybernetics, (52), 141-152.
5. Lin, W., Delgado-Frias, J. G., Pechanek, G. G., and Vassiliadis, S. (1994). "Impact of Energy Function on a Neural Network Model for Optimization Problems", ICNN, Vol.7, 4518-4523.
6. Park, C., and Fausett, D. W. (1995). "Energy Function Analysis for Improved Performance of Hopfield-Type Neural Networks", ANNIE, Vol.5, 995-1000.
7. Takefuji, Y. (1992). *Neural Network Parallel Computing*, Kluwer Academic Publishers.
8. Wilson, G. V., and Pawley, G. S. (1988). "On the Stability of the Traveling Salesman Problem Algorithm of Hopfield and Tank", Biological Cybernetics, (58), 63-70.