Proceedings of 1993 International Joint Conference on Neural Networks

Quantum Neurons and their Fluctuation

Satoshi Matsuda
Computer and Communication Research Center,
Tokyo Electric Power Company
1-3, Uchisaiwai-cho, 1-Chome, Chiyoda-ku, Tokyo, 100 Japan

ABSTACT: A new model of symmetric neural networks is presented, where each neuron takes one of the quantized values (e.g. integers) rather than just a binary values (i.e. 0 or 1) or continuous values (i.e. real numbers). By applying this model to combinatorial optimization problems which take integers as solutions, the number of neurons and connections between neurons, and computation time decrease greatly as compared with the traditional counting method[5]. Therefore, it is possible to get better solutions in the same total computation time. The simulation of Hitchcock problem is made to show these advantages. It is also illustrated, by the simulation, that some fluctuation coming from this quantization makes it possible to get better or best solution more easily. This fluctuation suggests an effective way to escape from the local minimum.

1. Introduction

The symmetric neural networks, so called Hopfield networks, have been widely applied to the combinatorial optimization problems such as travelling salesman problem[3]. Though neurons in symmetric neural networks take binary values[1] or continuous values[2], the final values neurons converge are supposed to be binary[2] [3]. Hence one makes efforts to design neural networks to represent and solve the target problems by using these binary values/neurons. In the previous paper[4], we proposed a new method suitable to combinatorial optimization problems which take real numbers or very large integers as solutions. On the other hand, in combinatorial optimization problems taking integers as solutions such as Hitchcock problems, these integers are usually represented by the number of (binary) neurons finally fired, which is called as the counting method[5]. However, it generally requires a large number of neurons and connections between neurons, which also causes a large amount of memories and computation time.

In this paper, a new model of the neuron is presented, which is suitable to combinatorial optimization problems taking integers as solutions. These neurons take one of the quantized values (e.g. integers) rather than just a binary values (i.e., 0 or 1) or continuous values (i.e. real numbers). These neurons are called as "quantum neurons", and the method representing integers of the solution by quantum neurons are called as "quantum method". By applying quantum method to combinatorial problems which take integers as solutions, it makes the number of neurons and connections, and computation time decrease greatly as compared with the traditional counting method. Therefore it is possible to get better solutions in the same total computation time. The simulation of Hitchcock problem is made to show these advantages. It is also illustrated, by the simulation, that some fluctuation coming from this quantization makes it possible to get better or best solution more easily. This fluctuation suggests an effective way to escape from the local minima.

2. Symmetric Neural Networks with Quantum Neurons (QSNN)

In the symmetric neural networks, for any two of neurons, say neuron i and j, there can be two connections from i to j and from j to i, of which weights, w_{ji} and w_{ij}, are equal, that is, $w_{ij}=w_{ji}$. The symmetric neural networks with quantum neurons, abbreviated as QSNN, are the symmetric neural networks where each neuron, for example, neuron k, takes one of integer values in $\{m_k, m_k+1, \cdots, M_k-1, M_k\}$. At each time one of the values of neurons is updated as follows:

$$\Delta x_k = \begin{cases} 1 & \text{if } \Sigma_j w_{kj} x_j + h_k + w_{kk}/2 > 0 \\ & \text{and } x_k < M_k \\ -1 & \text{if } \Sigma_j w_{kj} x_j + h_k - w_{kk}/2 < 0 \\ & \text{and } x_k > m_k \\ 0 & \text{otherwise} \end{cases}$$

and all the values of other neurons are not updated, that is, $\Delta x_j = 0$ for all $j \neq k$, where x_k is the value of neuron k, Δx_k is the value to be added to the current value of neuron k, w_{kj} is the weight of connection from neuron j to k, h_k is the threshold of neuron k, and $w_{kk} \neq 0$ in general. Thus QSNN is asynchronous model and each neuron is updated in turn. Note that we can generalize the values neuron takes to be any quantized values, however, in the above definition the values are restricted to be integers, for simplicity.

Same as an ordinary networks which take binary or continuous values[3], the energy of QSNN is defined as:

$$E = -1/2\ \Sigma_{ij} w_{ij} x_i x_j - \Sigma_i h_i x_i$$

Now the next theorem holds as same as ordinary symmetric neural networks.

[THEOREM] QSNN converges to a local minimum of energy function E.
[proof]

$$\begin{aligned} \Delta E(t) &= E(t+\Delta t) - E(t) \\ &= -1/2\ \Sigma_i \Sigma_j w_{ij}(x_i+\Delta x_i)(x_j+\Delta x_j) \\ &\quad - \Sigma_i h_i (x_i+\Delta x_i) \\ &\quad + 1/2\ \Sigma_i \Sigma_j w_{ij} x_i x_j + \Sigma_i h_i x_i \\ &= -1/2\ \Sigma_i \Sigma_j w_{ij}(x_i \Delta x_j + x_j \Delta x_i \\ &\quad + \Delta x_i \Delta x_j) - \Sigma_i h_i \Delta x_i \\ &= -1/2\ \Sigma_i \Delta x_i (\Sigma_j w_{ji} x_j + \Sigma_j w_{ij} x_j \\ &\quad + \Sigma_j w_{ij} \Delta x_j) - \Sigma_i h_i \Delta x_i \\ &= -\Sigma_i \Delta x_i (\Sigma_j w_{ij} x_j + h_i \\ &\quad + 1/2\ \Sigma_j w_{ij} \Delta x_j) \\ &\qquad (\because w_{ij} = w_{ji}) \end{aligned}$$

Here we can assume that only one neuron, say neuron k, updates its value, that is, $\Delta x_k \neq 0$, and $\Delta x_j = 0$ for all $j \neq k$. Hence, in this case, by denoting ΔE as $\Delta_k E$, we have

$$\Delta_k E = -\Delta x_k (\Sigma_j w_{kj} x_j + h_k + w_{kk} \Delta x_k / 2)$$

Since, by the definition of QSNN, it is impossible for $\Sigma_j w_{kj} x_j + h_k + w_{kk} \Delta x_k / 2$ and Δx_k to take the opposite sign, we can say

$$\Delta_k E \leqq 0.$$

Hence the energy function always decreases and finally converges to one of the local minima. [Q.E.D.]

3. Simulations

For combinatorial optimization problems with integers as solutions such as Hitchcock problem, the counting method [5], which represents integers of the solution by the number of neurons finally fired, is used. The quantum method certainly requires less number of neurons and connections than the counting method. Therefore, it is possible to reduce the computation time, and is expected to get good solutions in the same total computation time. Here we examine these by simula tions of Hitchcock problem.

(1) Hitchcock Problem

Hitchcock problem is a combinatorial optimization problem with integers as solution. Formally stated as

$$A\Sigma_i(\Sigma_j x_{ij} - s_i)^2 + B\Sigma_j(\Sigma_i x_{ij} - d_j)^2 + C\Sigma_{ij} c_{ij} x_{ij} \rightarrow \min$$

where s_i is the number of products supplier i $(i=1,\cdots,m)$ has, and d_j is the number of products consumer j $(j=1,\cdots,n)$ demands. c_{ij} is the unit cost of transportation from supplier i to consumer j. Hitchcock problem is to find a flow x_{ij} of products from supplier i to consumer j. A, B and C are weights of these constraints to be satisfied.

Now, our simulations are made for the same Hitchcock problem as Takeda and Goodman[5], where the number of suppliers and consumers are 4 and 5 respectively, i.e., $m=4$ and $n=5$, and the minimum transportation cost is 38. Two kinds of simulations are made, one is by the traditional counting method of binary neurons (BSNN) and another by the quantum method (QSNN). Each flow x_{ij} is represented by one neuron in the quantum method, and by 7 neurons in the counting method since $\max\{s_i, d_i\} = 7$. Except for this, followings are same in two cases. The weights of constraints are set as $A=B=80$ and $C=0.23$. The initial values of neurons are determined by using random number. Five hundred simulations are made by the counting method, and 5000 by quantum method. The simulation results are shown in Table 1.

(2) Simulation Results

We first notice that it takes near 11 hours to make 500 simulations by the counting method, but only about 5.5 hours to make 5000 simulations by the quantum method. Thus, as average time per simulation also shows, the quantum method takes only 1/20 simulation time of the counting method. In either cases, the average transportation cost obtained, and the rate of obtaining solutions, which satisfy first two constraints, is also almost same. Hence we can say that there is no great differences in quality of the solutions obtained by these two methods. However the least transportation cost obtained in these methods is 41 or 44 respectively. The quantum method seems superior to the counting method. This superiority comes from 10 times trials of simulations (note that, as mentioned above, even 20 times trials are possible in the same computation time), and the possibility of 10 or 20 times trials comes from the less computation time of quantum method.

4. Fluctuation

We have shown that the quantum method is superior to the counting method in the sense that it is possible to get better solutions in the same total computation time. However, even the quantum method in the simulations above could not get the best solution (cost=38). In general, one of the faults of symmetric neural network, including QSNN, is to be caught in a trap of the local minimum of energy function and not to get the best solution in many cases. So, there have been many trials to get better or best solutions by giving stochastic characteristics to the state transition of neurons such as Boltzmann machine. However, in general, these models are not practical because they require a large amount of computation time.

In this chapter we show, by simulation, that some fluctuation, which naturally comes from quntization, make it possible to get better solutions or best solution more easily.

(1) Quantum Fluctuation

In the traditional symmetric neural networks, the state transition of neuron depends on the sign of $\Sigma_j w_{kj} x_j + h_k$. On the other hand, in QSNN, it depends on the sign of $\Sigma_j w_{kj} x_j + h_k \pm 0.5 w_{kk}$. Here new state transition is presented as, in the case of $w_{kk} < 0$, for c such that $0 < c < 0.5$,

$$\Delta x_k = \begin{cases} 1 & \text{if } \Sigma_j w_{kj} x_j + h_k + c w_{kk} > 0 \text{ and } x_k < M_k \\ -1 & \text{if } \Sigma_j w_{kj} x_j + h_k - c w_{kk} < 0 \text{ and } x_k > m_k \\ 0 & \text{otherwise} \end{cases}$$

and, in the case of $w_{kk} > 0$, for c such that $0.5 < c < 1$, we can define same as above.

Obviously, any state transition made in QSNN is also made in these networks. On the other hand, there can be state transitions in these networks, which are impossible in QSNN, and these state transitions are called "quantum fluctuation". And these networks are called as FQSNN (Symmetric Neural Networks with Fluctuation to Quantum Neurons).

In general, a light fluctuation ($c \fallingdotseq 0.5$) can not let the network escape from the local minimum and causes less effects. A heavy fluctuation ($0.5 \gg c > 0$) can not let the network reach even the local minimum, and diverge, so causes a harm.

(2) Simulations of FQSNN

Five hundred simulations of the same Hitchcock Probelm above are made by FQSNN with $c=1/3$ fluctuation. For fairness, any of these simulations of FQSNN has the same initial values of neurons in that of QSNN. And, since FQSNN does not converge in general, we terminate the computation as convergence if the same energy value lasts for 5 rounds of updates of all the neurons. If such a condition does not occur until 50th round of updates, we stop the computation as no convergence.

Also in Table 1, the results of these simulations are shown. The performance of FQSNN increases greatly. Though 5000 simulations of QSNN can not get best solution, 500 simulations of FQSNN get best solutions 141 times. The average transportation cost obtained decreases from 52.9 to 40.9 greatly. These phenomena suggests that "there exists best solution near the better solutions." Moreover, 99% of these simulations converge and get approximate solutions, and the average number of iterations (computation time) of the simulation is only less than twice of that of QSNN. This good convergence of FQSNN also suggests that "the valley of energy surface with bad local minimum take the shape of a pan and the valley with good/gloval minimum a pot."

In the ordinary symmetric neural networks, the solutions depend on the initial values of all the neurons (sensitivity of initial values). Hence the choice of initial values is another target for the user or designer. These simulations show that the solutions obtained by FQSNN does not heavily depend on the initial values of the neurons. The quantum fluctuation make also user and designer free from the choice of initial values.

Thus, FQSNN does not always get best solutions, however, the simulations illustrate that it can get better or best solutions easily in a practical time. Simulations of many other problems which are not shown in this paper, also present a similar performance. Moreover, the quantum fluctuation is also effective for binary neurons.

References:

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

(2)Hopfield, J. J.: Neurons with graded response have collective computational properties like those of two-state neurons, Proc. Natl. Acad. Sci. USA, 81, 3088-3092(1984)

(3)Hopfield, J. J. and D. W. Tank: "Neural" computation of decisions of optimization problems, Biol. Cybern. 52, 141-152 (1985)

(4)Matsuda, S. and Y. Akimoto: The representation of large numbers in neural networks and its application to economical load dispatching of electric power, IJCNN89, I-587～592 (1989)

(5)Takeda, M. and J. W. Goodman: Neural network for computation:number representation and programming complexity, Appl. Opt. 25, 3033 (1986)

Table 1: Simulation results of Hitchcock problem

	counting method	quantum method	
	BSNN	QSNN	FQSNN
number of simulations	500	5000	500
number of simulations converges	500	5000	493
number of simulations obtaining solution	110 (22%)	1194 (24%)	493 (99%)
least transportation cost obtained	44	41	38 (best solution)
number of simulations obtaining best solution	0 (0%)	0 (0%)	141 (28.2%)
average transportation cost obtained	52.9	52.9	40.9
average number of iterations / simulation	7.8	6.6	12.6
total time of simulations	10H 44M 58S	5H 35M 49S	59M 15S
average time / simulation	78S	4.0S	7.1S
average time / solution	5M 52S	16.9S	7.2S