parallel computation. The neural network is greatly suitable for implementation on a digital machine.

REFERENCES

[1] N. Ansari, E. S. H. Hou, and Y. Yu, "A new method to optimize the satellite broadcasting schedules using the mean field annealing of a Hopfield neural network," *IEEE Trans. Neural Networks*, vol. 6, pp. 470–483, Mar. 1995.

[2] P. Bourret, S. Goodall, and M. Samuelides, "Optimal scheduling competitive activation: application to the satellite antennas scheduling problem," in *Proc. IJCNN'89*, 1989, pp. I565–572.

[3] P. Bourret, F. Rem, and S. Goodall, "A special purpose neural network for scheduling satellite broadcasting times," in *Proc. IJCNN'90*, 1990, pp. II535–538.

[4] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, pp. 141–152, 1985.

[5] W. S. McCulloch and W. H. Pitts, "A logical calculus of ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, p. 115, 1943.

[6] N. Funabiki and Y. Takefuji, "A parallel algorithm for channel routing problems," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 464–474, Apr. 1992.

[7] ——, "A neural-network parallel algorithm for channel assignment problems in cellular radio networks," *IEEE Trans. Veh. Technol.*, vol. 41, pp. 430–437, Nov. 1992.

[8] ——, "A neural network approach to topological via minimization problems," *IEEE Trans. Computer-Aided Design*, vol. 12, pp. 770–779, June 1993.

[9] N. Funabiki and S. Nishikawa, "An improved neural network for channel assignment problems in cellular mobile communication systems," *IEICE Trans. Commun.*, vol. E78-B, no. 8, pp. 1187–1196, Aug. 1995.

[10] T. Kurokawa and H. Yamashita, "Bus connected neural-network hardware system," *Electron. Lett.*, vol. 30, no. 12, pp. 979–980, June 1994.

Discrete-Time Convergence Theory and Updating Rules for Neural Networks with Energy Functions

Lipo Wang

***Abstract*—We present convergence theorems for neural networks with *arbitrary energy functions* and *discrete-time dynamics* for both *discrete* and *continuous neuronal input–output functions*. We discuss systematically how the neuronal updating rule should be extracted once an energy function is constructed for a given application, in order to guarantee the descent and minimization of the energy function as the network updates. We explain why the existing theory may lead to inaccurate results and oscillatory behaviors in the convergence process. We also point out the reason for and the side effects of using hysteresis neurons to suppress these oscillatory behaviors.**

I. INTRODUCTION

Since Hopfield and Tank [1] first applied their neural network to the traveling salesman problem, neural networks have been shown to provide powerful approaches for a wide variety of combinatorial optimization problems. In particular, Takefuji and coworkers have obtained remarkable solutions for many practical optimization problems using neural networks (e.g., [2] and [3]) and they found *discrete* neurons computationally more efficient in comparison with *continuous* neurons. Furthermore, *discrete-time dynamics* is usually much easier to implement compared to *continuous-time dynamics* which is described by differential equations. The basic idea behind these applications is the following. An optimization problem is first mapped onto a neural network in such a way that the network configurations correspond to possible solutions to the problem. A function of neuronal states, called the energy function, is constructed and this energy function is proportional to the cost function of the problem. The dynamics of the network is determined so that the energy function (therefore the cost function) is minimized as the neurons update.

Despite the large body of work in this area, the following *fundamental* question has not been adequately answered: for *discrete-time dynamics*, how the neuronal updating rule should be extracted once an energy function is constructed for a given application, in order to guarantee the descent and minimization of the energy function as the network updates? Adequately answering this question is clearly important from both theoretical and practical points of view, since incorrect updating algorithms will lead to inaccurate results in theoretical investigations and practical applications. As we will point out below, the existing results are error-prone.

Let us first review the existing results. Consider a network of N neurons. Suppose the output (or the state) V_i of neuron i at time $t + \Delta t$ is determined by the input U_i to neuron i at time t, through a nonincreasing input–output response function, $V_i(t+\Delta t) = f(U_i(t))$. Takefuji and Lee [2] showed that if the following updating rule is used:

$$dU_i/dt = -\partial E/\partial V_i \tag{1}$$

then the energy function E is a *nonincreasing* function of time

$$dE/dt \le 0. \tag{2}$$

It has been pointed out later [3], [4] that (2) holds for the updating rule given by (1) only if the input–output function f is also *continuous*, like the sigmoid function, $V_i = (1+\tanh(\lambda U_i))/2 \equiv g(\lambda U_i)$, where λ is the gain of the input–output function. An example was given in [4] to show that a *discrete input–output function* f may not guarantee (2) if (1) is used as the updating rule.

Lee and Takefuji [3] considered two discrete input–output functions: 1) the McCulloch–Pitts neuron

$$V_i(t+\Delta t) = \begin{cases} 1 & \text{if } U_i(t) > 0 \\ 0 & \text{otherwise} \end{cases}$$

and 2) the hysteresis McCulloch–Pitts neuron

$$V_i(t+\Delta t) = \begin{cases} 1 & \text{if } U_i(t) > \text{ UTP (upper trip point)} \\ V_i(t) & \text{if LTP} \le U_i(t) \le \text{UTP} \\ 0 & \text{if } U_i(t) < \text{ LTP (lower trip point).} \end{cases}$$

We regard the neuronal hysteresis *positive (negative)* if LTP $\le 0 \le$ UTP $(\text{if UTP} \le 0 \le \text{LTP})$, excluding UTP $=$ LTP $= 0$. Lee and Takefuji [3] proposed that if neurons are updated according to

$$\Delta U_i/\Delta t = -\Delta E_i/\Delta V_i \tag{3}$$

where

$$\Delta V_i(t) \equiv V_i(t+\Delta t) - V_i(t), \quad \text{and}$$
$$\Delta E_i(t) \equiv E[V_1(t), \cdots, (V_i(t) + \Delta V_i(t)), \cdots, V_N(t)] - E[V_1(t), \cdots, V_i(t), \cdots, V_N(t)]$$

Manuscript received August 31, 1995; revised September 17, 1996.

The author is with the School of Computing and Mathematics, Deakin University, Clayton, Victoria 3168, Australia.

Publisher Item Identifier S 1045-9227(97)01760-8.

then the energy function *does not increase*

$$\Delta E_i / \Delta t \leq 0. \tag{4}$$

Nevertheless Takefuji and coworkers have used (1) instead of (3) in their applications of networks of discrete neurons (see, e.g., [2] and [3]).

The drawbacks of the convergence theory and updating rules given by (1)–(4) are summarized below and will be illustrated subsequently. First, Lee and Takefuji [3] found undesirable oscillatory behaviors in the convergence process, which can be suppressed by introducing neuronal hysteresis [3], [5]; however, a convincing explanation for these phenomena has not been provided and will be provided in this paper. Second, the condition whether *time* is continuous or discrete was not explicitly spelled out. This formulation seems to imply that time is continuous when the input–output function is continuous [(1), (2)] and that time is discrete when the input–output function is discrete [(3), (4)]. However, whether the input–output function is discrete should be *independent* of whether the time is discrete. Third, the updating rule for discrete neurons given by (3) provides ΔU_i instead of U_i itself and differs from Hopfield's original updating rule [6] in the case of binary Hopfield network. Equation (3) is computationally inefficient, since a neuron may not be able to update its state until it has accumulated enough input during several evaluations of (3). Fourth, [4] suggests that (1) should be used when the input–output function is continuous but the time is discrete. We will show that this is incorrect for a large class of energy functions, including the ones used in [1]–[4]. In the next section we present updating rules and convergence theorems without these problems, thereby firmly establishing the starting point for solving optimization problems using discrete-time neural networks. Two examples and discussions will be presented in Section III.

II. DISCRETE-TIME CONVERGENCE THEOREMS

Theorem 1 (discrete Input–Output Neuronal Response Functions): For a network of neurons with regular McCulloch–Pitts or hysteresis McCulloch–Pitts input–output functions, for any nonzero change of state in any neuron i, i.e., $\Delta V_i \neq 0$, the energy is guaranteed to *decrease*

$$\Delta E_i(t) < 0 \tag{5}$$

if LTP $\leq$ 0, UTP $\geq$ 0, and the network is updated according to the following rules.

1) The network is updated *asynchronously*, that is, only one neuron i is selected for updating at time t.
2) The input to neuron i is calculated as follows. Suppose a "virtual" energy change $\Delta E_i'(t)$ is the result of a "virtual" change of state $\Delta V_i'(t)$ in neuron i at time t. The input to neuron i at time t is

$$U_i(t) = -\omega \Delta E_i'(t) / \Delta V_i'(t) \tag{6}$$

where $\omega > 0$ is the updating rate.

Proof: A change of state occurs if and only if one of the following occurs:

1) $V_i(t) = 1$ and $U_i(t) <$ LTP, thus $V_i(t + \Delta t) = 0$ and $\Delta V_i(t) = -1$ or
2) $V_i(t) = 0$ and $U_i(t) >$ UTP, thus $V_i(t + \Delta t) = 1$ and $\Delta V_i(t) = 1$.

In both cases

$$U_i(t) \Delta V_i(t) > 0. \tag{7}$$

Thus (5) holds, according to (6) and (7). Q.E.D.

The energy and state changes in (6) are "virtual" for two reasons. First, the input to *every* neuron can be calculated using these "virtual" changes *at any time*, but only the change associated with the particular neuron selected for updating may be realized. Second, only the state changes that *decrease* the energy may be realized during updating, as shown in the above proof.

Corollary 1: The necessary and sufficient condition under which (6) is equivalent to

$$U_i(t) = -\omega \partial E / \partial V_i \tag{8}$$

is that one of the following holds.

1) The energy function $E(V_1, V_2, \cdots, V_N)$ is in the form $\Sigma_s \; T^{(s)}_{\{i_1 i_2 \cdots i_N\}} V_1^{k_1^{(s)}} V_2^{k_2^{(s)}} \cdots V_N^{k_N^{(s)}}$, where $k_i^{(s)} = 0$ or 1, for all neurons (i) and all interacting neuronal sets (s) or
2) There are no self-interactions in any neuron.

Proof: 1) means that the energy is a linear function of $V_i, i = 1, 2, \cdots, N$, though E may contain terms such as $T_{ijk} V_i V_j V_k$ for higher order interactions. In this case (6) and (8) are equivalent. Self-interactions in neuron i correspond to energy terms that are nonlinear in V_i. Hence 2) is equivalent to 1). Q.E.D.

Theorem 2 (continuous Input–Output Neuronal Response Functions): For a network of neurons with continuous input–output neuronal response functions, for any change of state in any neuron i, the energy is guaranteed to *decrease* (5), if f is a monotonously increasing function and the network is updated according to the following rules.

1) The network is updated *asynchronously* and
2) If the following equations have nonzero solutions for ΔV_i and ΔU_i

$$V_i + \Delta V_i = f(U_i + \Delta U_i), \tag{9a}$$

$$\Delta U_i(t) = -\omega \Delta E_i(t) / \Delta V_i(t) \tag{9b}$$

the state of neuron i is updated; it remains unchanged otherwise.

Proof: Because f is a continuous and monotonously increasing function, for any $\Delta V_i \neq 0$, we have, instead of (7)

$$\Delta U_i(t) \Delta V_i(t) > 0. \tag{10}$$

Thus (5) holds according to (9b) and (10). Q.E.D.

Despite the resemblance between our (9b) and Takefuji and Lee's (3) [3], they are quite different: (9b) is for *continuous input–output functions only*, whereas (3) is for *binary input–output functions only*.

Corollary 2: The necessary and sufficient condition under which (9a) and (9b) are equivalent to

$$\Delta U_i(t) = -\omega \partial E / \partial V_i \tag{11}$$

is that one of the two conditions in Corollary 1 holds.

Proof: (Similar to the proof of Corollary 1—omitted) In this case, there is no need to solve (9a) and (9b), since (9b) no longer depends on ΔV_i.

III. EXAMPLES AND DISCUSSIONS

First, we consider the well-known energy function [6], which satisfies the conditions in Corollaries 1 and 2

$$E_h(t) \equiv -\sum_{i,j} T_{ij} V_i(t) V_j(t) / 2 \tag{12}$$

where $T_{ij} = T_{ji}$ and $T_{ii} = 0$.

In the case of *discrete* neurons, the updating rule should be, according to Theorem 1 (6) or Corollary 1 (8)

$$U_i(t) = \omega \sum_{j \neq i} T_{ij} V_j(t) \tag{13}$$

which is the original Hopfield formula [6] if we choose $\omega = 1$. In contrast, the formulation of Lee and Takefuji (3) gives an inefficient result: $\Delta U_i(t) = \Sigma_{j \neq i} \; T_{ij} V_j(t)$.

In the case of *continuous* sigmoid neurons with an energy function given by (12), will $E_h(t)$ monotonously decrease as the network updates asynchronously according to (13) if *time is discrete*? The answer is negative. For instance, consider a network of this type with two neurons connected by symmetric weights $T_{12} = T_{21} = T > 0$ and $T_{11} = T_{22} = 0$. At $t = 0$, let $V_1 = V_2 = v$, where v satisfies $g(\lambda T v) < v < 1$ (If $g(\lambda T v_o) = v_o$ has a positive solution, let $v_o < v < 1$. Otherwise let $0 < v < 1$.) Suppose neuron 1 is selected to update at the next time step $t = 1, U_1 = Tv$ according to (13), $V_1(1) = g(\lambda T v) < v$, and $V_2(1) = v$ (not updated). Thus $E_h(1) = -Tvg(\lambda T v) > E_h(0) = -Tv^2$. In fact, if (13) is used in the case of a continuous input–output function, a different function is minimized [7]–[9] and this function may deviate signifcantly from (12) for a low gain (small λ) [7].

How should the neurons be updated if an optimization application requires that the energy function given by (12) must be minimized in the case of continuous neuronal response function and discrete-time dynamics? According to our Theorem 2 (9b) or Corollary 2 (11), we should adopt $\Delta U_i(t) = \omega \Sigma_{j \neq i} \; T_{ij} V_j(t)$, instead of (13). For the two-neuron network considered above, $\Delta U_1 = \omega T v$ and $U_1(1) = U_1(0) + \Delta U_1 > U_1(0)$. Hence $V_1(1) = g(\lambda U_1(1)) > g(\lambda U_1(0)) = V_1(0)$ and $E_h(1) < E_h(0)$, for any $\omega > 0$. If $\omega \gg 1, V_1(1) = V_2(2) = 1, E_h(2) = -T$, i.e., the system rapidly reaches the energy minimum. There is no need to solve (9a) and (9b) in this case.

Now let us consider the energy function used in [4], which includes neuronal self-interaction terms and therefore does not satisfy the conditions in Corollaries 1 and 2

$$E(t) \equiv 2[V_1(t) - 1]^2 + 2[V_2(t) - 1]^2 + V_1(t)V_2(t).$$

In the case of binary neurons, Tateishi and Tamura [4] showed that Takefuji and Lee's (1) does not guarantee the descent of this energy function. However, Tateishi and Tamura [4] did not provide any updating algorithm that always decreases the energy and they expected that (1) would guarantee the decrease if continuous sigmoid neurons are used instead of binary neurons. This is *false*, as we will now show. In the case of continuous neurons, the sole minimum of this energy function $E_{\min} = 0.8$ is at $V_1 = V_2 = 0.8$. Suppose at $t = 0$, the network is at $V_1 = 0.8, V_2 = 0.7$, and $E = 0.82$. It is straightforward to verify the following. For a large gain λ, the energy increases to $E = 0.88$ if (1) is used to update neuron 2 to $V_2 = 1$. In contrast, the solutions of our (9a) and (9b) do give reduced energy. In the case of binary neurons, our (6) must be used as the updating rule to guarantee energy minimization for this type of energy function.

Energy functions that include V_i^2 terms account for a large number of energy functions used in optimization problems (e.g., [1]–[3]) and these terms correspond to *negative* neuronal self-interactions. We attribute the undesirable oscillatory behaviors in the convergence process found by Takefuji and coworkers (e.g., [2] and [3]) to the fact that they have used (1) as their updating rules, which can lead to increases in energy when the conditions in Corollaries 1 and 2 are not satisfied, as shown in the second example. Neuronal hysteresis is equivalent to a *positive* neuronal self-interaction [5], can thus suppress these oscillatory behaviors. However, introduction of hysteresis effectively changes the energy function and may lead to different outcomes. Hence, to efficiently obtain accurate results in optimization problems, one should use the *correct updating rules* presented in this paper. Practical applications of the present results are the subject of future work.

ACKNOWLEDGMENT

The authors thank the reviewers for many helpful comments and suggestions.

REFERENCES

[1] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, pp. 141–152, 1985.

[2] Y. Takefuji and K. C. Lee, "Artificial neural networks for four-coloring map problems and K-colorability problems," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 326–333, Mar. 1991.

[3] K. C. Lee and Y. Takefuji, "Finding knight's tours on an $M \times N$ chessboard with $O(MN)$ hysteresis McCulloch–Pitts neurons," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, pp. 300–306, Feb. 1994.

[4] M. Tateishi and S. Tamura, "Comments on 'Artificial neural networks for four-coloring map problems and K-colorability problems,'" *IEEE Trans. Circuits Syst. I: Fundamental Theory Applicat.*, vol. 41, pp. 248–249, Mar. 1994.

[5] L. Wang and J. Ross, "Synchronous neural networks of nonlinear threshold elements with hysteresis," in *Proc. Nat. Academy Sci. USA*, vol. 87, Feb. 1990, pp. 988–992.

[6] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," in *Proc. Nat. Academy Sci. USA*, vol. 79, 1982, pp. 2554–2558.

[7] ——, "Neurons with graded response have collective computational properties like those of two-state neurons," in *Proc. Nat. Academy Sci. USA*, vol. 81, May 1984, pp. 3088–3092.

[8] F. Fogelman Soulie, C. Mejia, Eric Goles, and S. Martinez, "Energy function in neural networks with continuous local functions," *Complex Syst.*, vol. 3, pp. 269–293, 1989.

[9] P. Koiran, "Dynamics of discrete-time, continuous state Hopfield networks," *Neural Computa.*, vol. 6, pp. 459–468, 1994.