Discrete Synchronous Neural Algorithm for Minimization

Hyuk Lee
Department of Electrical Engineering, Polytechnic Institute of New York, Brooklyn, NY 11201 USA

A general discrete minimization algorithm that can be implemented by highly parallel neural networks is developed. It can be applied to the energy functions that can be expressed as arbitrary types of polynomial functions of the state variables. The algorithm can be operated in a synchronous way.

1 Introduction

Highly parallel neural computing algorithms have been investigated extensively. The Hopfield model has been successfully applied for solving combinatorial minimization problems (Hopfield 1982). However, the energy function in the Hopfield model is restricted to a symmetric quadratic form having all the diagonal elements zero. A higher order Hopfield model (Maxwell et al. 1986; Psaltis and Park 1986; Paek et al. 1988) has also been considered. In this case, the energy function is a polynomial of the state variables and it is assumed to have special symmetry properties.

Furthermore, the updating rules of such algorithms are based on the serial operation. At each step, a state variable is selected randomly and minimization is carried out by updating only the selected state variable and leaving all the other state variables unchanged. Therefore, the algorithm can be operated in an asynchronous way, but it cannot be operated in a synchronous way. Synchronous algorithms for continuous state variables have been investigated (Marcus and Westervelt 1989). However, synchronous discrete neural algorithms have not been studied. In this paper, a partially synchronous discrete neural algorithm that can be applied to the arbitrary types of polynomial energy functions is developed. In Section 2 a general algorithm is developed. In Section 3 the Hopfield and quadratic neural models are considered as specific examples of the general algorithm.

Neural Computation, 1, 522–531 (1989) © 1989 Massachusetts Institute of Technology
2 Partially Synchronous Algorithm

The energy function is assumed to be an arbitrary type of polynomial function of the state variables. Real binary variables having values -1 and 1 are considered as state variables. The energy function can be described as

\[ E = F(\{B_1, B_2, \ldots, B_N\}) \quad (2.1) \]

where \( B \)'s are state binary variables and the total number of state variables is \( N \).

Partially synchronous minimization is considered for the most general case. Totally synchronous or totally asynchronous minimizations are specific examples of the general case. Assume that, at each step, \( M \) state variables are selected randomly, and minimization is carried out by updating the \( M \) state variables simultaneously and leaving all the other state variables unchanged. \( M \) can be any integer from 1 to \( N \), and the minimization algorithm becomes totally asynchronous or totally synchronous if \( M \) is equal to 1 or \( N \). At each step of updating, the number \( M \) can be changed and chosen arbitrarily. Even if the same number \( M \) is chosen, a different set of \( M \) neurons out of \( N \) neurons can be used to minimize the neural network. A set \( P \) is defined to consist of the indices for the selected state variables. Another set \( P' = \{1, 2, \ldots, N\} - P \) is defined to represent the indices of the state variables that are unchanged. As an example, consider a neural system consisting of five neurons numbered 1, 2, 3, 4, 5. If the system is updated by changing the states of three neurons 1, 2, and 5, then the set \( P \) is \( \{1, 2, 5\} \) and the set \( P' \) is \( \{3, 4\} \). In the case of totally synchronous updating algorithm, the set \( P \) is given by \( \{1, 2, 3, 4, 5\} \) and \( P' \) is an empty set. On the other hand, the set \( P \) for the totally asynchronous updating rule by changing the state of the neuron 3 becomes \( \{3\} \), and the set \( P' \) for this case is given by \( \{1, 2, 4, 5\} \). The updated state variables \( B' \) and the change of the state variables \( \Delta B \) satisfy the relation

\[ B'_i = B_i + \Delta B_i \quad (2.2) \]

where \( i \in P \). The updated state variables are also binary variables having values -1 and 1. Therefore, the possible values of \( \Delta B \) are

\[ \Delta B_i = -2, 0, 2 \quad (2.3) \]

The incremental change in energy \( \Delta E \) due to the updated state variables given by equation 2.2 is considered to develop an algorithm that minimizes the energy function described by equation 2.1. \( \Delta E \) is defined as

\[ \Delta E = \{[B'_i, B_j] - E'[B_i, B_j]\} \quad (2.4) \]
where $i \in P$ and $j \in P'$, and utilizing equation 2.2, it becomes
\[
\Delta E = E[\{B_i + \Delta B_i, B_j\}] - E[\{B_i, B_j\}] \tag{2.5}
\]
The first term in the right-hand side of equation 2.5 can be expanded as a Taylor’s series in several variables because $E$ is a polynomial of state variables. Therefore, the incremental energy change $\Delta E$ can be written as (Toralballa 1967)
\[
\Delta E = \sum_{m=1}^{\infty} \sum_{i_1 \in P} \cdots \sum_{i_m \in P} \frac{1}{m!} \left[ \Delta B_{i_1} \cdots \Delta B_{i_m} \right] \frac{\partial^m E}{\partial B_{i_1} \cdots \partial B_{i_m}} \tag{2.6}
\]
The total number of terms in the summation of equation 2.6 is finite because $E$ is a polynomial.

To reduce the products of changes in equation 2.6 to linear forms, the following relations are derived. For an arbitrary state variable $B$ and a positive integer $n$, $(\Delta B)^n$ satisfies
\[
(\Delta B)^n = (-2B)^{n-1} \Delta B \tag{2.7}
\]
which can be proved as follows. If $\Delta B$ is zero, equation 2.7 is satisfied automatically. If $\Delta B$ is 2, $B$ and $B'$ should be -1 and 1 according to equation 2.2. In this case, the right-hand side of equation 2.7 becomes the same as the left-hand side, that is,
\[
[(-2)(-1)]^{n-1}[2] = 2^n = (\Delta B)^n \tag{2.8}
\]
On the other hand, if $\Delta B$ is -2, $B$ and $B'$ should be 1 and -1 according to equation 2.2. Therefore, the right-hand side of equation 2.7 becomes
\[
[(-2)(1)]^{n-1}[-2] = [-2]^n \tag{2.9}
\]
which is the same as the left-hand side of equation 2.7 in the case of $\Delta B = -2$. Consider next a product of changes given by $A(\Delta B_{i_1} \cdots \Delta B_{i_m})$ with an arbitrary coefficient $A$. Assume that $m > 1$. The product can be written as
\[
A(\Delta B_{i_1} \cdots \Delta B_{i_m}) = |A|(1/2)^2 \left[ -S(A)\Delta B_{i_1} - \Delta B_{i_2} \cdots \Delta B_{i_m} \right]^2 + (\Delta B_{i_1})^2 + (\Delta B_{i_2} \cdots \Delta B_{i_m})^2 \tag{2.10}
\]
where $S(A)$ is the sign of $A$. The first term in equation 2.10 is always negative or zero and equation 2.11 follows. The second term in equation 2.11 is a product of smaller number of changes than the left term in equation 2.10. Therefore, this technique can be applied iteratively to reduce the product of changes in equation 2.10 to a sum of individual changes, and it is given by
\[
A(\Delta B_{i_1} \cdots \Delta B_{i_m}) \leq |A|^{(m-1)} \left( \sum_{j=1}^{m-1} 2^{-j}(\Delta B_{i_j})^{2^j} \right) + 2^{-(m-1)}(\Delta B_{i_m})^{2^{(m-1)}} \tag{2.12}
\]
Equation 2.12 can be symmetrized by considering \( m \) cyclic orderings of indices \( \{i_1, \ldots, i_m\} \). Equation 2.12 now is given by

\[
A \Delta B_{i_1} \cdots \Delta B_{i_m} \leq (1/m)|A| \sum_{k=1}^{m-1} \left\{ \sum_{j=1}^{m-1} 2^{-j}(\Delta B_{i_j})^{2^j} \right\} \\
+ 2^{-(m-1)}(\Delta B_{i_k})^{2^{m-1}} \quad (2.13)
\]

If equation 2.7 is utilized, equation 2.13 can be written as

\[
A \Delta B_{i_1} \cdots \Delta B_{i_m} \leq I(m)|A| \sum_{k=1}^{m} B_{i_k} \Delta B_{i_k} \quad (2.14)
\]

where

\[
I(m) = -(1/m) \left\{ \sum_{j=1}^{m-1} 2^{2j-j-1} \right\} + 2^{(2m-1)-m} \quad (2.15)
\]

The incremental energy change introduced in equation 2.6 can be written as a sum of the linear term in \( \Delta B_i \) and the higher order terms as follows:

\[
\Delta E = \sum_{i \in P} \Delta B_i \frac{\partial E}{\partial B_i} \\
+ \sum_{i \in P} \left\{ \sum_{m=1} \sum_{t_1 \in P} \cdots \sum_{i_m \in P} \frac{1}{(m+1)!} [\Delta B_{i_1} \Delta B_{i_1} \cdots \Delta B_{i_m}] \\
\frac{\partial^{m+1} E}{\partial B_{i_1} \partial B_{i_1} \cdots \partial B_{i_m}} \right\} \quad (2.16)
\]

where the index \( m \) is rearranged in the second term. From equation 2.14, we have the following relation:

\[
A \Delta B_{i_1} \Delta B_{i_1} \cdots \Delta B_{i_m} \leq |A|I(m+1) \{ \sum_{k=1}^{m} B_{i_k} \Delta B_{i_k} + B_i \Delta B_i \} \quad (2.17)
\]

and utilizing the above equation, the second term in equation 2.16 satisfies

\[
\sum_{i \in P} \left\{ \sum_{m=1} \sum_{t_1 \in P} \cdots \sum_{i_m \in P} \frac{1}{(m+1)!} [\Delta B_{i_1} \Delta B_{i_1} \cdots \Delta B_{i_m}] \frac{\partial^{m+1} E}{\partial B_{i_1} \partial B_{i_1} \cdots \partial B_{i_m}} \right\} \leq \sum_{i \in P} \left\{ \sum_{m=1} \sum_{t_1 \in P} \cdots \sum_{i_m \in P} \frac{1}{(m+1)!} I(m+1) \left| \frac{\partial^{m+1} E}{\partial B_{i_1} \partial B_{i_1} \cdots \partial B_{i_m}} \right| \right\} \\
\{ \sum_{k=1}^{m} B_{i_k} \Delta B_{i_k} + B_i \Delta B_i \} \quad (2.18)
\]
Equation 2.16 now satisfies the following relation if equation 2.18 is used:

\[
\Delta E \leq \sum_{i \in P} \frac{\partial E}{\partial B_i} \Delta B_i \leq \sum_{i \in P} \sum_{k=1}^{m} \left( \sum_{m=1}^{I(m+1)} \frac{I(m+1)!}{m!} \sum_{m \in P} \sum_{i \in P} \frac{\partial^{m+1} E}{\partial B_i \partial B_{i_k} \cdots \partial B_{i_m}} \right)
\]

\[
= \sum_{i \in P} \frac{\partial E}{\partial B_i} \Delta B_i + \sum_{i \in P} \Delta B_i \left( \sum_{m=1}^{I(m+1)} \frac{I(m+1)!}{m!} \sum_{i \in P} \frac{\partial^{m+1} E}{\partial B_i \partial B_{i_k} \cdots \partial B_{i_m}} \right)
\]

\[
\sum_{m \in P} \left\{ \sum_{i \in P} \Delta B_{i_k} + B_i \Delta B_i \right\}
\]

(2.19)

where the dummy indices \( i \) and \( i_k, j = 1, \ldots, m \), have been interchanged and

\[
\frac{\partial^{m+1} E}{\partial B_i \partial B_{i_k} \cdots \partial B_{i_m}} = \frac{\partial^{m+1} E}{\partial B_{i_k} \partial B_i \cdots \partial B_{i_m}}
\]

(2.20)

has been used to obtain the equality in the above equation. The incremental energy change is given by

\[
\Delta E \leq -\sum_{i \in P} G_i \Delta B_i
\]

(2.21)

where

\[
G_i = -\left[ B_i \left\{ \sum_{m=1}^{I(m+1)} \sum_{i \in P} \left( \frac{1}{m!} \right) I(m+1) \right\} \frac{\partial^{m+1} E}{\partial B_i \partial B_{i_k} \cdots \partial B_{i_m}} \right] + \frac{\partial E}{\partial B_i}
\]

(2.22)

From equation 2.21 it is clear that if

\[
G_i \Delta B_i \geq 0 \text{ for all } i \in P
\]

(2.23)

the incremental energy change is always negative or zero. Therefore, equation 2.23 updates the state variables in such a way that it minimizes the energy function in the limit of iteration. For positive \( G_i \), equation 2.23 is satisfied if \( \Delta B_i \) is positive or zero. If \( B_i \) is equal to 1, \( B'_i \) should be equal to 1 because \( \Delta B_i = 0 \) is the only solution. However, if \( B_i \) is equal to -1, \( B'_i \) should be 1 because this makes the incremental energy change more negative than using the condition \( \Delta B_i = 0 \). Therefore, if \( G_i \) is positive, the updated value becomes 1, which is the same as the sign of the value \( G_i \). If \( G_i \) is negative, the above argument can be applied to show that the updated value for \( B'_i \) becomes -1. This is the same as the negative of the sign of \( G_i \). If \( G_i \) is zero, \( B'_i \) can have any values. In this case, there are three possibilities. The first one is to choose 1 for \( B'_i \). Using the value -1 for \( B'_i \) is the second possibility. The third one is to use the relation \( B'_i = -B'_i \). As an example, the first possibility, that is, using the value 1
for \( B'_i \) when \( G_i \) is zero, is used in the following. Summarizing the above result, the updating rule can be written as

\[
B'_i = T(G_i) \text{ for all } i \in P
\]  

(2.24)

where \( T \) is a unit step function defined by \( T(x) = 1 \) if \( x \geq 0 \) and \( T(x) = -1 \) if \( x < 0 \). Equation 2.24 is the general discrete neural algorithm that minimizes the energy functions consisting of arbitrary types of polynomials of state variables in a partially synchronous way.

To illustrate the general algorithms obtained in the above, a simple example consisting of three neurons is considered and the results of the simulation are described. The energy function for this example is chosen as

\[
E = B_1B_2 + 2B_2B_3 + B_1
\]  

(2.25)

A totally synchronous algorithm is considered in the following. Therefore, the set \( P \) introduced in the beginning of this section is given by \{1, 2, 3\} and the set \( P' \) is empty. The intermediate variables \( G_i \) are obtained by using equation 2.22 and they are given by

\[
G_1 = B_1 - B_2 - 1
\]  

(2.26)

\[
G_2 = -B_1 + 3B_2 - 2B_3
\]  

(2.27)

\[
G_3 = -2B_2 + 2B_3
\]  

(2.28)

where \( I(2) = -1 \) was used. As an example, select \( B_1 = 1, B_2 = 1, B_3 = 1 \) as an initial state. Then the values \( G_i \) become \(-1, 0, 0\), respectively, using the same set of values for \( B_i \), that is, in a totally synchronous way. The updated states are given by \( B'_1 = -1, B'_2 = 1, B'_3 = 1 \) and the energy

\[
\begin{array}{c|c|c|c|c|c}
\text{Initial states} & \text{E} & \text{G1,G2,G3} & \text{B1,B2,B3} & \Delta E \\
\hline
(1,1,1) & 4 & (-1,0,0) & (-1,1,1) & -4 \\
(1,1,-1) & 0 & (-1,4,-4) & (-1,1,-1) & -4 \\
(1,-1,1) & -2 & (1,-2,4) & (1,-1,1) & 0 \\
(1,-1,-1) & 2 & (1,-2,0) & (1,-1,1) & -4 \\
(-1,1,1) & 0 & (-3,2,0) & (-1,1,1) & 0 \\
(-1,1,-1) & -4 & (-3,6,-4) & (-1,1,-1) & 0 \\
(-1,-1,1) & -2 & (-1,-4,4) & (-1,-1,1) & 0 \\
(-1,-1,-1) & 2 & (-1,0,0) & (-1,1,1) & -2 \\
\end{array}
\]

Table 1: Values of \( G_i \).
change $\Delta E$ becomes $-4$. In Table 1, the values $G_i$ corresponding to the eight initial states are shown. The updated values $B'_i$, $i = 1, 2, 3$, and the energy change $\Delta E$ for each case are also shown. It is clearly seen that the energy always decreases or remains the same, which demonstrates the validity of the algorithm.

3 Partially Synchronous Hopfield and Quadratic Neural Models

As an example, an energy function consisting of cubic, quadratic, and linear terms is considered. In general, a partially synchronous algorithm is considered. The most general form of such energy function is given by

$$E = -\sum_i \sum_j \sum_k Q_{ijk} B_i B_j B_k - \sum_i \sum_j T_{ij} B_i B_j - \sum_i I_i B_i$$  \hspace{1cm} (3.1)

where $Q_{ijk}$, $T_{ij}$, and $I_i$ are constant coefficients. The intermediate variable $G_k$ in equation 2.22 is obtained by calculating the derivatives

$$\frac{\partial E}{\partial B_k} = -\sum_i \sum_j (Q_{kij} + Q_{ikh} + Q_{ijk}) B_i B_j - \sum_i (T_{ki} + T_{ik}) B_i - I_k$$  \hspace{1cm} (3.2)

$$\frac{\partial^2 E}{\partial B_k \partial B_{i_1}} = -\{\sum_j (Q_{kij} + Q_{jss} + Q_{jss} + Q_{jss}) B_j + Q_{i_1kj} B_j\} - (T_{ki_1} + T_{ik})$$  \hspace{1cm} (3.3)

$$\frac{\partial^2 E}{\partial B_k \partial B_{i_1} \partial B_{i_2}} = -(Q_{k_{i_1}i_2} + Q_{k_{i_2}i_1} + Q_{i_2} + Q_{i_2} + Q_{i_1} + Q_{i_1} + Q_{i_2} + Q_{i_2}) + Q_{i_1kj}$$  \hspace{1cm} (3.4)

If equations 3.2–3.4 are substituted in equation 2.22, $G_k$ is given by

$$G_k = \sum_i \sum_j (Q_{kij} + Q_{ikj} + Q_{ijk}) B_i B_j + \sum_i (T_{ki} + T_{ik}) B_i + I_k + \{\sum_{i_1} \sum_j (Q_{k_{i_1}j} + Q_{j_{i_1}s} + Q_{j_{i_1}s} + Q_{j_{i_1}s}) B_j + (T_{ki_1} + T_{ik} B_k + (5/6)B_k [\sum_{i_1} \sum_{i_2} (Q_{k_{i_1}i_2} + Q_{k_{i_2}i_1} + Q_{i_2} + Q_{i_2} + Q_{i_1} + Q_{i_1} + Q_{i_2} + Q_{i_2}) + Q_{i_1kj}]\}$$  \hspace{1cm} (3.5)

where $I(2) = -1$ and $I(3) = -(5/3)$ are used, and the indices $k$, $i_1$, and $i_2$ represent the neurons used to update the system, that is, they belong to the set $P$ defined in Section 2.
A generalized form of the Hopfield model in the case of partially synchronous algorithm is obtained if \( Q_{ijk} \) and \( I_i \) are all zero. The memory matrix \( M_{ij}^{H} \) in the Hopfield model is defined as

\[
G_k = \sum_j M_{kj}^{H} B_j
\]

and from equation 3.5 the memory matrix is given by

\[
M_{kj}^{H} = (T_{kj} + T_{jk}) + (1/2)\delta_{kj} \sum_i \{|T_{ki} + T_{ik}| + |T_{ji} + T_{ij}|\}
\]

where \( \delta_{ij} \) is the Kronecker symbol. The memory vectors \( B_k^{(m)} \) are contained in the \( T \) matrix as

\[
T_{kj} = \sum_m B_k^{(m)} B_j^{(m)}
\]

where \( m \) represents the number of memory vectors, and \( T_{ii} \) are all zero. It is clear that the memory matrix is symmetric. However, the diagonal components of the memory matrix \( M^{H} \) are not zero in general.

As an example, we consider a neural network consisting of 32 neurons. Three memory vectors \( a, b, c \) shown in Table 2 were stored using equation 3.8. Four partially synchronous as well as totally synchronous and totally asynchronous algorithms were used. In Table 2, the degree

<table>
<thead>
<tr>
<th>Memory</th>
<th>Memory Vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>(-1) (-1) (1) (1) (-1) (1) (-1) (1) (-1) (1) (-1) (1) (-1) (1) (-1) (1)</td>
</tr>
<tr>
<td>( b )</td>
<td>(1) (-1) (-1) (1) (-1) (-1) (-1) (-1) (-1) (-1) (-1) (-1) (-1) (-1) (-1) (-1)</td>
</tr>
<tr>
<td>( c )</td>
<td>(1) (1) (1) (1) (-1) (-1) (-1) (-1) (-1) (-1) (-1) (-1) (-1) (-1) (-1) (-1)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Degree of synchronism</th>
<th>Attraction radii</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>( b )</td>
</tr>
<tr>
<td>( 1 )</td>
<td>( 11 )</td>
</tr>
<tr>
<td>( 2 )</td>
<td>( 16 )</td>
</tr>
<tr>
<td>( 4 )</td>
<td>( 10 )</td>
</tr>
<tr>
<td>( 8 )</td>
<td>( 9 )</td>
</tr>
<tr>
<td>( 16 )</td>
<td>( 3 )</td>
</tr>
<tr>
<td>( 32 )</td>
<td>( 0 )</td>
</tr>
</tbody>
</table>

Table 2: Memory Vectors.
of synchronism was defined as the number of neurons that was used to update the neural network simultaneously. For example, the number 4 for the degree of synchronism implies that the neural network was updated by selecting four neurons and changing the states of the four neurons simultaneously. As special cases, degree of synchronism 1 and 32 represents the totally asynchronous (conventional Hopfield model) and totally synchronous algorithm, respectively. Attraction radii for the three memory vectors were obtained by computer simulation using different degrees of synchronism. The results are shown in Table 2. It shows that as the degree of synchronism increases, the attraction radii, that is, the effectiveness of the algorithm, decreases. Specially, the attraction radii for the totally synchronous algorithm are zero. This phenomenon can be explained if equation 2.22 is investigated in detail. In equation 2.22, the effect of an increased degree of synchronism is to have a larger set \( P \), and the coefficient of the term \( B_i \) in the first part of the right-hand side of equation 2.22 is always negative. This implies that as the set becomes larger, the first part of the right-hand side of equation 2.22 becomes the dominant one. Therefore, the sign of \( G_i \) becomes the same as that of \( B_i \), and the updated states are the same as the initial states.

A higher order Hopfield model is obtained by assuming \( T_{ij} = 0 \) and \( I_i = 0 \). The memory vectors are contained in the \( Q \) matrix and it is given by

\[
Q_{kij} = \sum_m B_{k}^{(m)} B_{i}^{(m)} B_{j}^{(m)}
\]  

(3.9)

The updating rule for the synchronous algorithm can be read from equation 3.5

\[
G_k = \sum_i \sum_j (Q_{kij} + Q_{ikj} + Q_{ijk}) B_i B_j
\]

\[+ \{ \sum_{\tilde{t}_1} \sum_{\tilde{t}_2} (Q_{k\tilde{t}_1\tilde{t}_2} + Q_{\tilde{t}_1k\tilde{t}_2} + Q_{\tilde{t}_2k\tilde{t}_1} + Q_{\tilde{t}_1\tilde{t}_2k}) B_{\tilde{t}_1} B_{\tilde{t}_2} + (5/6) B_k \{ \sum_{\tilde{t}_1} \sum_{\tilde{t}_2} |Q_{k\tilde{t}_1\tilde{t}_2} + Q_{\tilde{t}_1k\tilde{t}_2} + Q_{\tilde{t}_2k\tilde{t}_1} + Q_{\tilde{t}_1\tilde{t}_2k} + Q_{\tilde{t}_1k\tilde{t}_2} \} \}
\]

(3.10)

Considering the second term of equation 3.10, it is clear that the state variables are inside the absolute value. This makes it impossible to define a memory matrix \( M^Q \) in the form of

\[
G_k = \sum_i \sum_j M_{kij}^Q B_i B_j
\]  

(3.11)

which is the same form as that used in the asynchronous quadratic model.

4 Conclusion

In conclusion, a general discrete neural algorithm that minimizes arbitrary types of polynomial energy functions has been developed for the
first time within my knowledge. The updating rules have been obtained for a general partially synchronous algorithm. The algorithms obtained in this paper do not need any special conditions, such as no starvation condition, bounded delays, etc., which were assumed in the past partially synchronous algorithms. As shown in the simulation results on the Hopfield type associative memory in Section 3, there is a trade-off between the degree of synchronism, that is, degree of parallelism, and the effectiveness of minimization process. This is understood by the fact that the algorithms for different degrees of synchronism obtained in this paper differ only in the higher order derivative terms as shown in equation 2.22. Therefore, as the total number of neurons in a neural network and the degree of synchronism increase, the linear derivative term in equation 2.22 becomes less important and the minimization process becomes less effective. Detailed analysis on this trade-off will be the subject of future publications.

Acknowledgments

This work was supported by the National Science Foundation Grant No. EET-8810288 and the Center for Advanced Technology in Telecommunications. The author appreciates the reviewer’s insightful comments, which made the paper clearer.

References


Received 15 December 1988; accepted 15 October 1989.