Pattern Recognition, Vol. 25, No. 2, pp. 197–209, 1992
Printed in Great Britain

0031-3203/92 $5.00 + .00
Pergamon Press plc

RELAXATION BY THE HOPFIELD NEURAL NETWORK

SHIAW-SHIAN YU* and WEN-HSIANG TSAI†‡

*Advanced Technology Center, Computer and Communication Research Laboratories, Industrial Technology Research Institute, Chutung, Hsinchu, Taiwan 31015, Republic of China; †Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 30050, Republic of China

(*Received* 27 *June* 1990; *in revised form* 20 *June* 1991; *received for publication* 16 *July* 1991)

Abstract—The relaxation process is a useful technique for using contextual information to reduce local ambiguity and achieve global consistency in various applications. It is basically a parallel execution model, adjusting the confidence measures of involved entities based on interrelated hypotheses and confidence measures. On the other hand, the neural network is a computational model with massively parallel execution capability. The output of each neuron depends mainly on the information provided by other neurons. Therefore, there exist certain common properties in the relaxation process and the neural network technique. A mapping method that makes the Hopfield neural network perform the relaxation process is proposed. By this method, the neural network technology can be easily adapted to solve the many problems which have already been solved by the relaxation process. An advantage of this is that the relaxation process can be performed in real time since the Hopfield network can be implemented by conventional analog circuits. Experimental results are given to demonstrate the feasibility of the proposed method by performing the image thresholding operation on the proposed neural network.

Neural networks   Relaxation   Labeling problem   Hopfield model   Image thresholding

1. INTRODUCTION

The relaxation process is a useful technique for using contextual information to reduce local ambiguity and achieve global consistency. It has been applied successfully to a lot of image analysis tasks, such as scene labeling,[1,2] shape matching,[3,4] line and curve enhancement,[5,6] handwritten character recognition,[7,8] and thinning.[9]

There are three types of the relaxation processes, namely discrete relaxation, probabilistic relaxation, and fuzzy relaxation.[10] The one implemented by the neural network technique in this study is probabilistic relaxation.

First of all, we describe the general concept of probabilistic relaxation.[10] Suppose that a set of n objects $A_1, A_2, \ldots, A_n$ are to be classified into m classes $C_1, C_2, \ldots, C_m$. Suppose further that the class assignments are correlated; in other words, for each pair of class assignments $A_i \in C_j$ and $A_h \in C_k$, there is a quantitative measure of the compatibility of the pair, which will be denoted by $c(i, j; h, k)$.

Let $p_{ij}^{(0)}$ be an initial estimate of the probability that $A_i \in C_j$, $1 \le i \le n$, $1 \le j \le m$. Thus for each i we have $0 \le p_{ij}^{(0)} \le 1$ and $\Sigma_{j=1}^{m} p_{ij}^{(0)} = 1$. The goal of the relaxation process is to find a set of n classifications of all the objects which are as compatible as possible using the initial estimate $p_{ij}^{(0)}$ and the compatibility measure $c(i, j; h, k)$.

In each iteration of the relaxation process, the probability value of each assignment of an object must be adjusted according to the current probability value of the object and the other objects' contributions. The net contributions from the other objects, denoted as $q_{ij}^{(r)}$, to the probability value of assigning object i to class j at the rth iteration is calculated by

$$q_{ij}^{(r)} = \frac{1}{n-1} \sum_{h=1, h \ne i}^{n} \left(\sum_{k=1}^{m} c(i, j; h, k) p_{hk}^{(r)} \right).$$

And the new estimate of the probability value of assigning object i to class j at the $(r+1)$th iteration is defined as follows:

$$p_{ij}^{(r+1)} = \frac{p_{ij}^{(r)}(1 + q_{ij}^{(r)})}{\Sigma_{j=1}^{m} p_{ij}^{(r)}(1 + q_{ij}^{(r)})}.$$

The above process is performed iteratively until the process converges or until a certain termination condition is satisfied.

On the other hand, neural networks are systems constructed to make use of some of the organizational principles that are felt to be used in the human brain. Neural networks may be viewed from many different perspectives. Some researchers regard neural networks as physical systems. They tend to think of neural networks as embodying an

‡ Author to whom correspondence should be addressed.

energy surface, whose states of minimal energy represent solutions to many combinatorial optimization problems. Each of such systems solves a problem through a process, in which the energy state of the network slides downhill to an energy minimum. The Hopfield network configuration[11–13] is a typical example. It has been successfully used to give good solutions to certain difficult problems such as the well-known NP-complete traveling salesman problem in which the shortest path through multiple cities is searched.

Some other researchers view neural networks from a biological and psychological perspective. These researchers tend to consider neural networks as means for implementing cognitive mechanisms operating over a wide range of domains, including vision processing,[14–17] associative memories,[18] and models of diverse neurobiological functions.[19,20]

Also included in the neural network study is the information-processing approach known as connectionist computing,[21,22] in which a task is performed by many processors sending information to one another through point-to-point connections. Finally, from a parallel processing viewpoint, a neural network can also be regarded as a computational model with massively parallel execution capability,[23,24] forming an attractive model for efficient massively parallel machines with fine-grained distributed memory.

Physically, a neural network includes many neurons and links between pairs of neurons. Each of the neurons of the network has a state variable known as its activation level; and each of the links has a value known as its connection weight. Neurons communicate by transmitting their activation levels to one another over the links. Neurons and links can be configurated into an arbitrary topology. The computational process envisioned with a neural network is as follows. A neuron receives inputs from a number of other neurons or from the external world. The weighted sum of these inputs is the argument of an activation function. This activation function is assumed to be nonlinear. The resulting value of the activation function is the output of the neuron. The output is distributed along the weighted links to the other neurons.

Genis[25] proposed two ways of establishing a correspondence between the fields of relaxation labeling and neural learning process: one in which the short-term neural network function is interpreted as relaxation and thus neural learning amounts to discovering the constraints between the states of connected neurons, and the other in which neural learning itself is interpreted as an extended form of relaxation that permits incorporating new evidences at each iteration.

In this paper, we propose a new method for constructing a neural network which can perform the relaxation process. Using this network, we can use neural network techniques to solve the many problems which have already been solved by the relaxation process. The advantages of the neural network can thus be injected into the numerous relaxation applications.

This paper is organized as follows. The next section includes the description of the method of solving optimization problems using the Hopfield neural network. The relationship between the relaxation process and the Hopfield neural network model is examined in Section 3, and a method to perform the relaxation operation on the Hopfield neural network is proposed. Experimental results demonstrating the feasibility of the proposed method by performing the thresholding operation on the proposed neural network are given in Section 4. Concluding remarks are included in Section 5.

2. A REVIEW ON THE HOPFIELD NEURAL NETWORK

There are a number of ways of organizing the computing elements in a neural network. Typically the elements are arranged in groups or layers. A single layer of neurons that connects to one another usually is called an autoassociative system. One of the most famous single layer neural networks is the Hopfield network developed by Hopfield and Tank.[11–13]

The Hopfield network is a recurrent network containing feedback paths from the outputs of the neurons back into their inputs so that the response of such a network is dynamic. This means that after applying a new input, the output is calculated and fed back to modify the input. The output is then recalculated, and the process is repeated again and again. Successive iterations produce smaller and smaller output changes, until eventually the outputs become constant and the network stable. Hopfield suggested a similarity of the network to the movement and the kinetic energy values of atoms under different temperatures. Plotting the values of neuron outputs as heights on a 2D state–space plane creates a landscape of hills and valleys. The function of a neural network will develop a number of locally stable points or valleys in the state space. Other points in the state space flow into the stable points where the corresponding energy field is minimized.

The connection weights between the neurons in the network may be considered to form a matrix T. It has been shown that a recurrent network is stable if the matrix is symmetrical with zeros on its diagonal, that is, if $T_{ij} = T_{ji}$ for all i and j and $T_{ii} = 0$ for all i.[26]

To illustrate the Hopfield networks in more detail, consider the special case of a Hopfield network with a symmetric matrix. The input to the ith neuron comes from two sources: external inputs and inputs from the other neurons. The total input u_i to neuron i is then

$$u_i = \sum_{j \neq i} T_{ij} V_j + I_i,$$

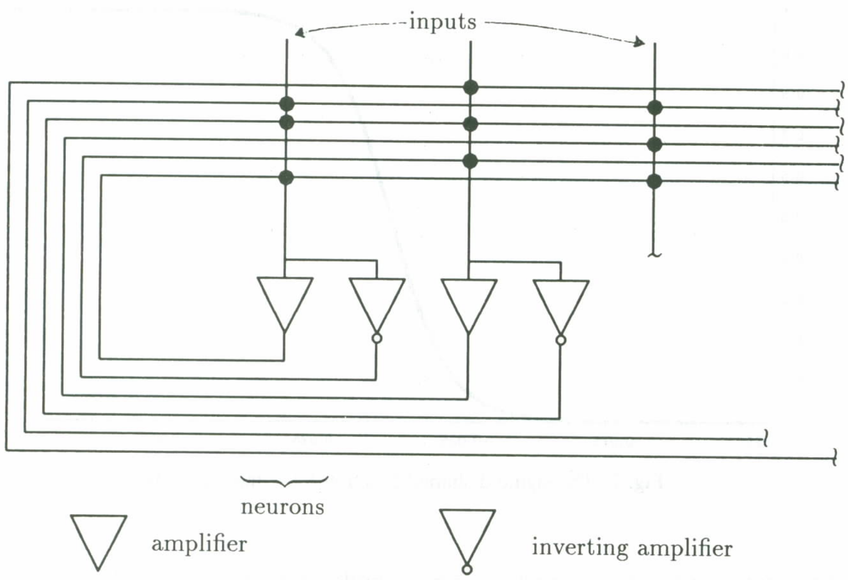


Fig. 1. Hopfield network architecture.[12]

where the V_j value represents the output of the jth neuron; T_{ij} is the weight of the connection between neurons V_i and V_j; and I_i represents an external input bias value which is used to set the general level of excitability of the network through constant biases.

At any time, the V_i values are updated in accordance with the following rule:

$$\begin{aligned} V_i &\to 0 \text{ if } \sum_{j \neq i} T_{ij}V_j + I_i < \theta_i; \text{ or} \\ V_i &\to 1 \text{ if } \sum_{j \neq i} T_{ij}V_j + I_i > \theta_i, \end{aligned} \tag{1}$$

where θ_i is the threshold of the ith neuron which will be taken to be zero for this study. The neurons are interrogated and updated in a stochastic asynchronous manner. A quantity to describe the state of the network, called energy, is defined as

$$E = -\frac{1}{2}\sum_i \sum_{j \neq i} T_{ij}V_iV_j - \sum_i V_iI_i. \tag{2}$$

The change ΔE in E due to the state change ΔV_i of neuron i is given by

$$\Delta E = -\left[\sum_{j \neq i} T_{ij}V_j + J_i\right]\Delta V_i. \tag{3}$$

According to Equation (1), ΔV_i is positive only when $\Sigma_{j \neq i} T_{ij}V_j + I_i$ is positive, and is negative only when $\Sigma_{j \neq i} T_{ij}V_j + I_i$ is negative. Thus, from Equation (3), any change in E under the algorithm is negative. Furthermore, E is bounded, so iterations of the algorithm will lead to stable states that do not change further with time.

For applications which cannot be solved by binary neuron networks described above, networks of neurons with graded responses have also been proposed.[13,27] The total input into a neuron is converted into an output value by a sigmoid monotonic activation function instead of the thresholding operation described by Equation (1). The evolving of a neuron in this case is defined by

$$\frac{du_i}{dt} = -\frac{u_i}{\tau_i} + \sum_{j \neq i} T_{ij}V_j + I_i, \tag{4}$$

where

$$V_j = g_j(u_j) \text{ for all } j, \tag{5}$$

$g_j(u_j)$ is the sigmoid activation function of neuron j, and τ_i is a time constant which can be set to 1 for simplicity. As the system evolves, due to the feedback dynamics, the energy decreases until it reaches a minimum.[12] A major advantage of the Hopfield network with graded responses is that the whole network can be implemented by conventional analog circuits.[28,29] A circuit diagram that illustrates this type of network appears in Fig. 1. It consists of a set of network inputs, a memory storage matrix (T_{ij}), a set of nonlinear outputs, a set of switches, and indicated signal-flows. Hopfield and Tank[13] suggested a sigmoid activation function as follows:

$$V_{ij} = g(u_{ij}) = \tfrac{1}{2}(1 + \tanh(\lambda u_{ij})) \tag{6}$$

which is employed in this study (see Fig. 2). The λ value in the above equation is the gain of the network, and a larger λ value results in a large slope of the g function.

To solve an optimization problem using the Hopfield network, one must decide first the representation scheme which allows the outputs of the neurons to be interpreted as a solution of the

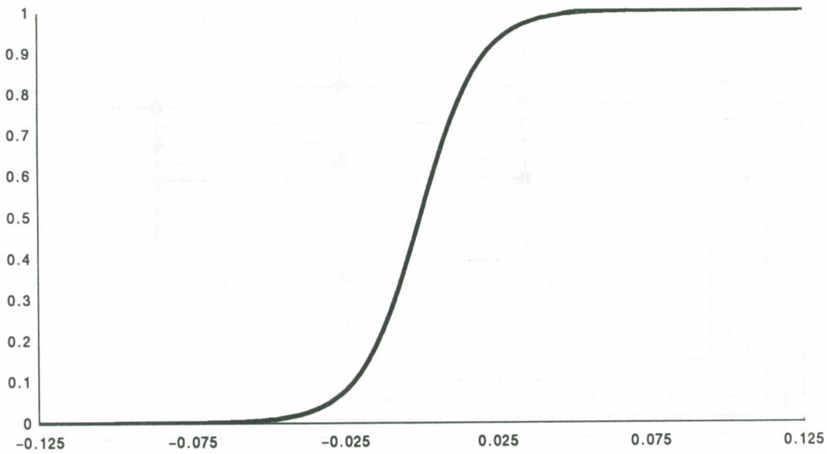


Fig. 2. The sigmoid shaped function $\frac{1}{2}(1 + \tanh(100u_{ij}))$.

problem, define next an energy function whose minimum value corresonds to the best solution of the problem, and derive finally the connection weights and the input bias values from the form of the energy function. After setting up the initial values for the input bias values, one can then start to evolve the system to get the solution.

3. RELAXATION BY THE HOPFIELD NETWORK

In the relaxation process, the intention is to get an assignment of each object to a class in a manner consistent with certain predefined constraints. In such a process, it is often necessary to adjust the current degree of confidence of each possible conclusion based on many interrelated hypotheses and confidence measures. The process is parallel and cooperative in nature because the hypotheses in such a process comprise a parallel ensemble of interdependent issues. On the other hand, the Hopfield network can be treated as a kind of parallel and cooperative computation mechanism. This means that there exist some properties common to the relaxation process and the Hopfield network.

The goal of the relaxation process is to increase the consistency among the constraints so that a better state can be reached in each iteration. Therefore, if a goodness function can be found to represent the consistency between the current state and the constraints, then problem solving by relaxation can be regarded as a process which increases iteratively the value of the goodness function. In this viewpoint, the Hopfield network turns out to be appropriate to perform the relaxation process because a Hopfield network tends to reach a stable state by minimizing the energy function when the network evolves. Hence our objective now is to find a goodness function which will increase when the relaxation process proceeds, and transform the goodness function into the form of an energy function for the Hopfield network. Furthermore, in such a network the neural nodes conceptually can be used to represent the various possible hypotheses and pieces of evidence in the relaxation process and the neural links may embody the relationships among them.

There are many ways to measure the goodness function of the relaxation process. A quantity called average local consistency[30] is found in this study to be useful for this purpose. More specifically, in reference (30) the support for assigning object i to class j by the current state of the relaxation process is defined as

$$\sum_h \sum_k c(i, j; h, k) p_{hk},$$

and this support measure was employed to define a measure of the average local consistency of the whole system as

$$\sum_i \sum_j p_{ij} \left(\sum_h \sum_k c(i, j; h, k) p_{hk} \right). \quad (7)$$

As mentioned above, the neural nodes in a Hopfield network can be used to represent the various possible hypotheses and pieces of evidence in the original relaxation problem. Accordingly, let a neuron with index (i, j) in a Hopfield network with graded responses represent the hypothesis of object A_i being from class C_j. Also regard the probability value p_{ij} as the neuron output value V_{ij}. Then, according to the form of the goodness function of Equation (7), it is proposed in this study to use the following energy function of the Hopfield network:

$$E = -\frac{A}{2} \sum_i \sum_j V_{ij} \left(\sum_h \sum_k c(i, j; h, k) V_{hk} \right) + \frac{B}{2} \sum_i \left(\left(\sum_j V_{ij} \right) - 1 \right)^2, \quad (8)$$

in which the first term is basically identical to Equation (7) except the constant value $-A/2$, and the second term is minimized to be zero when the sum of the outputs of the neurons representing different assignments of a certain object is equal to 1 (which means that each object is assigned only to a unique class). Both A and B are positive numbers. Expanding Equation (8), we get

$$E = -\frac{A}{2}\sum_i\sum_j\sum_h\sum_k c(i,j;h,k)V_{ij}V_{hk} + \frac{B}{2}\sum_i\left(\left(\sum_j V_{ij}\right)^2 - 2\left(\sum_j V_{ij}\right) + 1\right)$$

$$= -\frac{A}{2}\sum_i\sum_j\sum_h\sum_k c(i,j;h,k)V_{ij}V_{hk} + \frac{B}{2}\sum_i\left(\sum_j V_{ij}\right)^2 - B\sum_i\sum_j V_{ij} + \frac{nB}{2}. \quad (9)$$

From Equations (2) and (9) and by negating the constant term, it is easy to derive the values of the connection matrix and the input bias value as follows:

$$T_{(ij)(hk)} = A \times c(i,j;h,k) - B \times \delta_{ih}, \quad (10)$$

$$I_{ij} = B, \quad (11)$$

where $\delta_{ih} = 1$ for $i = h$, 0 for $i \neq h$. Note that the subscripts (ij) and (hk) of T in Equation (10) are considered as two single indices.

To ensure that the mapped Hopfield network will converge, sometimes the compatibility coefficients must be modified. The reason is as follows. It has been shown that under certain conditions, the relaxation process generally leads to consistent labeling results if the compatibility coefficient matrix is symmetric;[30] and as described in Section 2, the convergence of the Hopfield network is ensured if the connection matrix is symmetric and has zero diagonal elements. According to Equation (10), the connection matrix is symmetric if the compatibility coefficient matrix is symmetric. Therefore, it is proposed in this study to modify, whenever necessary, the compatibility coefficients to be symmetric, and zero when the coefficients appears as the diagonal elements in the connection matrix, according to the following rules:

$$c((x,y),j;(u,v),k) \leftarrow \frac{c((x,y),j;(u,v),k) + c((u,v),k;(x,y),j)}{2},$$

and

$$c((x,y),j;(x,y),j) \leftarrow 0.$$

For any relaxation problem, the set of $T_{(ij)(hk)}$ and I_{ij} values embody all the characteristics of the problem. Once the $T_{(ij)(hk)}$ and I_{ij} values are fixed, the final stable state of the neural network will depend on the initial state only. We must set the initial probability value for each hypothesis by using certain a priori knowledge before the start of a relaxation process. For example, when using the relaxation process to detect the edges of the objects in an image, because a larger gradient value at a certain pixel implies a higher possibility that this pixel is an edge pixel, the gradient value of each pixel can be used to define the initial probability value of the event that the corresponding pixel will be classified to be an edge pixel. Accordingly, in the proposed Hopfield network for relaxation, the initial values of the neural input values u_{ij} are set according to the following equation:

$$u_{ij} = g_{ij}^{-1}(p_{ij}^{(0)}),$$

where $p_{ij}^{(0)}$ is the initial probability value of assigning object A_i to class C_j.

After defining the compatibility coefficients and the initial probability values, the probabilistic relaxation process is a deterministic iterative process, i.e. each time the whole system will reach the same conclusion through the same process. The operation of a Hopfield network also is a deterministic process. After assigning the connection weights, input bias values, and the initial input value of each neuron, the energy surface and the local minimum points are completely defined, and the whole network will reach a specific local minimum on the energy surface.

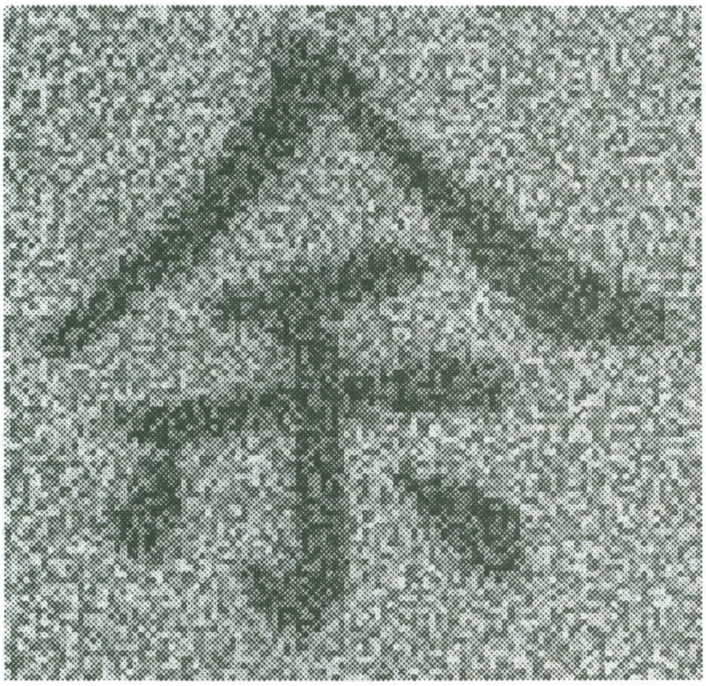
4. EXPERIMENTAL RESULTS

To demonstrate the correctness of the proposed mapping method between the relaxation process and the Hopfield network, a typical image processing problem, namely, thresholding, is performed. We first use the relaxation process to do the thresholding operation on a gray scale image, and then use the proposed Hopfield network to perform the same operation on the same image.

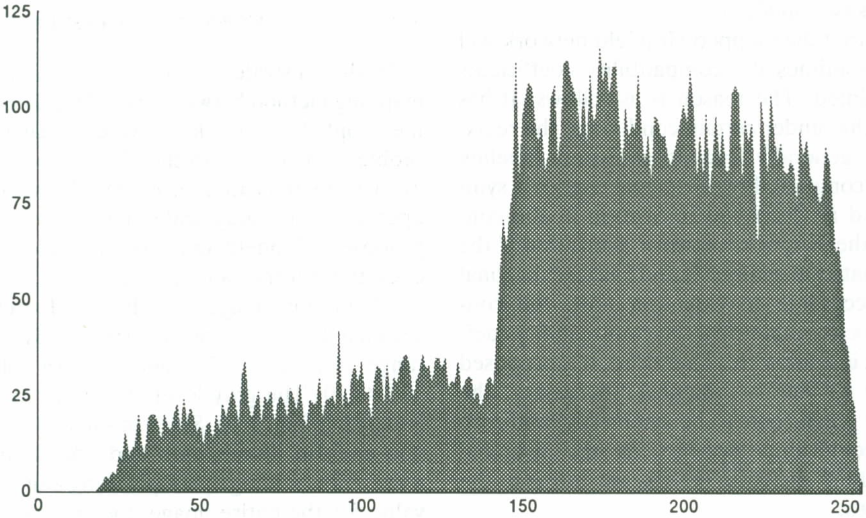
The original image of a Chinese character "Yu" is shown in Fig. 3(a). The intensity values of the images range from 0 to 255, and the size of images is 100×100. The gray level histogram for this image is shown in Fig. 3(b). Since this image is taken under non-uniform illumination and added with random noise, thresholding by simply using a single threshold value for the entire image does not work (see Fig. 3(c) and (d)), so the relaxation method is a good selection for solving this problem.

When applying the probabilistic relaxation method to threshold,[31,32] there are two pixel classes, corresponding to low (object) and high (background) gray levels. For each pixel $A_{x,y}$ there are two probability values, $p_{(x,y)0}$ and $p_{(x,y)1}$, where $p_{(x,y)1} = 1 - p_{(x,y)0}$. Moreover, for each pixel $A_{x,y}$ and any of its eight neighboring pixels, $A_{u,v}$, there are four compatibility coefficients, $c((x,y),0;\ (u,v),0)$, $c((x,y),0;\ (u,v),1)$, $c((x,y),1;\ (u,v),0)$, and $c((x,y),1;\ (u,v),1)$.

To assign initial probability values to each pixel, we need a scheme to determine the likelihood of the pixel being in the object or background class. For this, the initialization scheme based on gray levels



(a)



(b)

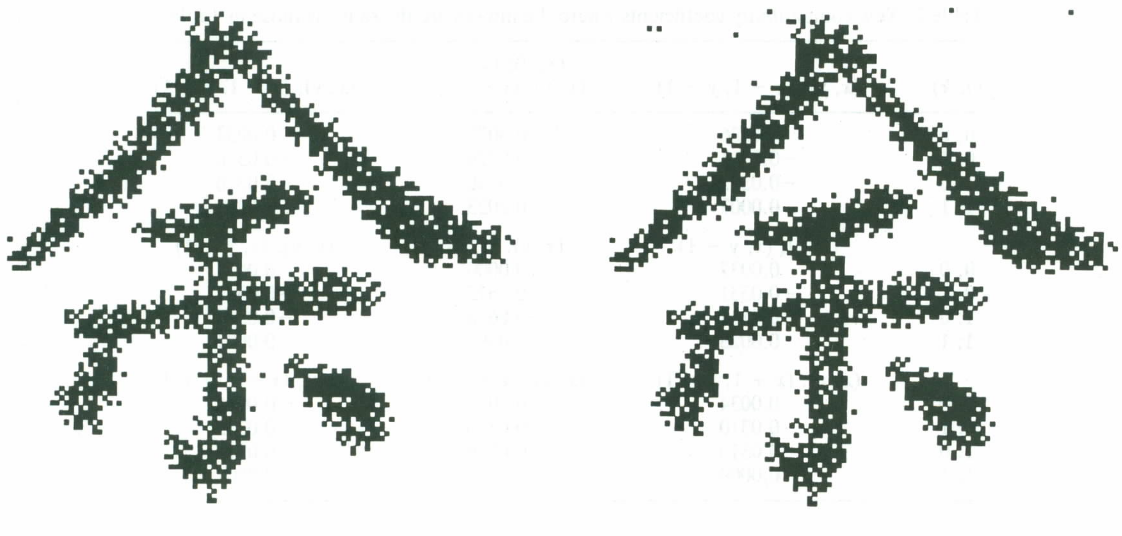
Fig. 3. *Continued on following page.*

developed by Rosenfeld and Smith[32] is adopted. Let $G_{x,y}$ be the gray level of pixel $A_{x,y}$, m be the mean gray level, and O and B be the gray levels corresponding to the object and the background, respectively; then we define

$$p^{(0)}_{(x,y)1} = \frac{1}{2} + \frac{G_{x,y} - m}{2(B - m)} \quad \text{for } G_{x,y} > m$$

$$p^{(0)}_{(x,y)0} = \frac{1}{2} + \frac{m - G_{x,y}}{2(m - O)} \quad \text{for } G_{x,y} \leq m.$$

On the other hand, a general method of defining compatibility coefficients for Fig. 3(a) based on the mutual information of the classes at neighboring pixels was proposed by Peleg and Rosenfeld.[6] The compatibility coefficients computed by this method are shown in Table 1. Theoretically, the compatibility coefficients generated from the mutual information are symmetric (i.e. $c(i, j; h, k) = c(h, k; i, j)$), but the compatibility coefficients shown in Table 1 are not so because they are influenced by



(c) (d)

Fig. 3. Thresholding result of using a single threshold value for the whole image: (a) original image of the Chinese character "Yu"; (b) gray level histogram of the image; (c) result of using 124 as threshold value; (d) result of using 125 as threshold value.

Table 1. Compatibility coefficients computed from the use of mutual information where the element in row (j, k) and column $(x, y); (u, v)$ means the coefficient $c((x, y), j; (u, v), k)$

		$(x, y); (u, v)$	
(j, k)	$(x, y); (x-1, y-1)$	$(x, y); (x-1, y)$	$(x, y); (x-1, y+1)$
0; 0	0.0034	0.0036	0.0033
0; 1	−0.0280	−0.0313	−0.0278
1; 0	−0.0346	−0.0348	−0.0343
1; 1	0.0004	0.0021	0.0009
	$(x, y); (x, y-1)$	$(x, y); (x, y)$	$(x, y); (x, y+1)$
0; 0	0.0037	0.0202	0.0038
0; 1	−0.0315	−0.0612	−0.0315
1; 0	−0.0347	−0.0612	−0.0346
1; 1	0.0021	0.0199	0.0024
	$(x, y); (x+1, y-1)$	$(x, y); (x+1, y)$	$(x, y); (x+1, y+1)$
0; 0	0.0035	0.0038	0.0037
0; 1	−0.0278	−0.0313	−0.0280
1; 0	−0.0342	−0.0346	−0.0343
1; 1	0.0008	0.0024	0.0009

the boundary pixels of the original image. To solve this problem, the compatibility coefficients were modified according to the rules described in Section 3. The new compatibility coefficients are shown in Table 2.

The result of applying the relaxation process with the original compatibility coefficients to Fig. 3(a) is shown in Fig. 4(a); and the result of applying the relaxation process with the new compatibility coefficients to Fig. 3(a) is shown in Fig. 4(b). The two results have no obvious difference, but the experiments show that the relaxation process converges faster when the new compatibility coefficients are used. During the relaxation process, each pixel is affirmed to a class when the corresponding probability value is greater than 0.99 instead of 1. This reduces the number of iterations to terminate the process by one-fourth.

In this study, the proposed Hopfield network for the thresholding problem using the connection matrix defined in Equation (10) and the input bias terms of Equation (11) was simulated on a Sun-4/280 workstation. The equations which describe the evolving of the neurons of the network for this problem are Equations (4) and (6). The graded response characteristics of the neurons in the Hopfield network represent partial knowledge or belief. A value for $V_{(x,y),j}$ between 0 and 1 represents the strength

Table 2. New compatibility coefficients where the indices are the same as those in Table 1

(j, k)	$(x, y); (x-1, y-1)$	$(x, y); (u, v)$ $(x, y); (x-1, y)$	$(x, y); (x-1, y+1)$
0; 0	0.0036	0.0037	0.0034
0; 1	−0.0311	−0.0329	−0.0310
1; 0	−0.0313	−0.0330	−0.0310
1; 1	0.0007	0.0023	0.0009
	$(x, y); (x, y-1)$	$(x, y); (x, y)$	$(x, y); (x, y+1)$
0; 0	0.0037	0.0000	0.0037
0; 1	−0.0331	−0.0612	−0.0331
1; 0	−0.0331	−0.0612	−0.0331
1; 1	0.0023	0.0000	0.0023
	$(x, y); (x+1, y-1)$	$(x, y); (x+1, y)$	$(x, y); (x+1, y+1)$
0; 0	0.0034	0.0037	0.0036
0; 1	−0.0310	−0.0330	−0.0313
1; 0	−0.0310	−0.0329	−0.0311
1; 1	0.0009	0.0023	0.0007

of the hypothesis that pixel (x, y) should be assigned to class j.

In the simulation, the gain value λ was set to 100, both A and B values of the energy function to 1, and the value of the time step dt in Equation (4) to 10^{-4}. The system is regarded as having converged when the energy does not change any more.

The result of applying the relaxation process to Fig. 3(a) using the proposed Hopfield network is shown in Fig. 4(c). There is almost no difference between Fig. 4(b) and Fig. 4(c).

Although the probabilistic relaxation process and the function of the proposed relaxation network are conceptually the same, there are differences between them. The relaxation process took about 2600 iterations and 1 CPU hour before it terminated. On the other hand, the simulated Hopfield network took about 30,000 time steps and 34 CPU hours to converge. There is no direct correspondence between the iterations and the time steps, since the time steps are only used for simulating an analog circuit which can reach a stable state in real time. Furthermore, during the process of the probabilistic relaxation, the total sum of the probability values of each class of any object should remain 1. On the contrary, in the relaxation network, we just *encourage* the sum to be 1 by introducing a term into the energy function (see Equation (8)). While theoretically the sum can range from 0 to 2, in our simulation it keeps very close to 1. Consequently, the second term of Equation (8), the energy function, always keeps a value close to zero. The reason is that the compatibility coefficients $c((x, y), j; (x, y), k)$ are negative when $j \neq k$. This means that neurons $V_{(x,y),0}$ and $V_{(x,y),1}$ are competing with each other.

Figure 5(a) shows the negated values of the average local consistency of the relaxation process, as a function of the iteration number. Figure 5(b) shows the energy values of the proposed neural network, as a function of the time step. Both the negated average local consistency value and the energy value dropped very quickly. This means that most pixels are affirmed to appropriate classes at the early stages. It also means that the proposed network has a convergence behavior very similar to that of the relaxation process.

The gain in the activation function plays an important role during the operation of the network. As mentioned above, larger gain values result in larger slopes of the activation function and faster converging speeds. For example, when the gain value λ was set to 10^4, the simulated Hopfield network took about 700 time steps and 0.7 CPU hours to converge (see Fig. 4(d)). However, a very high gain will result in a fast cooling network, so that the outputs of most neurons will be either 0 or 1, and the potential analog characteristics of the network will not be fully utilized. For the thresholding problem, this means that the network will produce a thresholding result (see Fig. 4(e) for an example) little better than that of using a single threshold value. Figure 6 shows the energy values of the proposed neural network at different gain values, as a function of the time step.

5. CONCLUDING REMARKS AND FUTURE RESEARCH DIRECTIONS

The relaxation process is a useful technique for using contextual information to reduce local ambiguity and achieve global consistency in various applications. It is basically a parallel execution model, adjusting the confidence measures of involved entities based on interrelated hypotheses and confidence measures. On the other hand, the neural network is a computational model with massively parallel execution capability. The output of each neuron depends mainly on the information provided by other neurons. Therefore, there exist certain common properties in the relaxation process and the neural network technique.

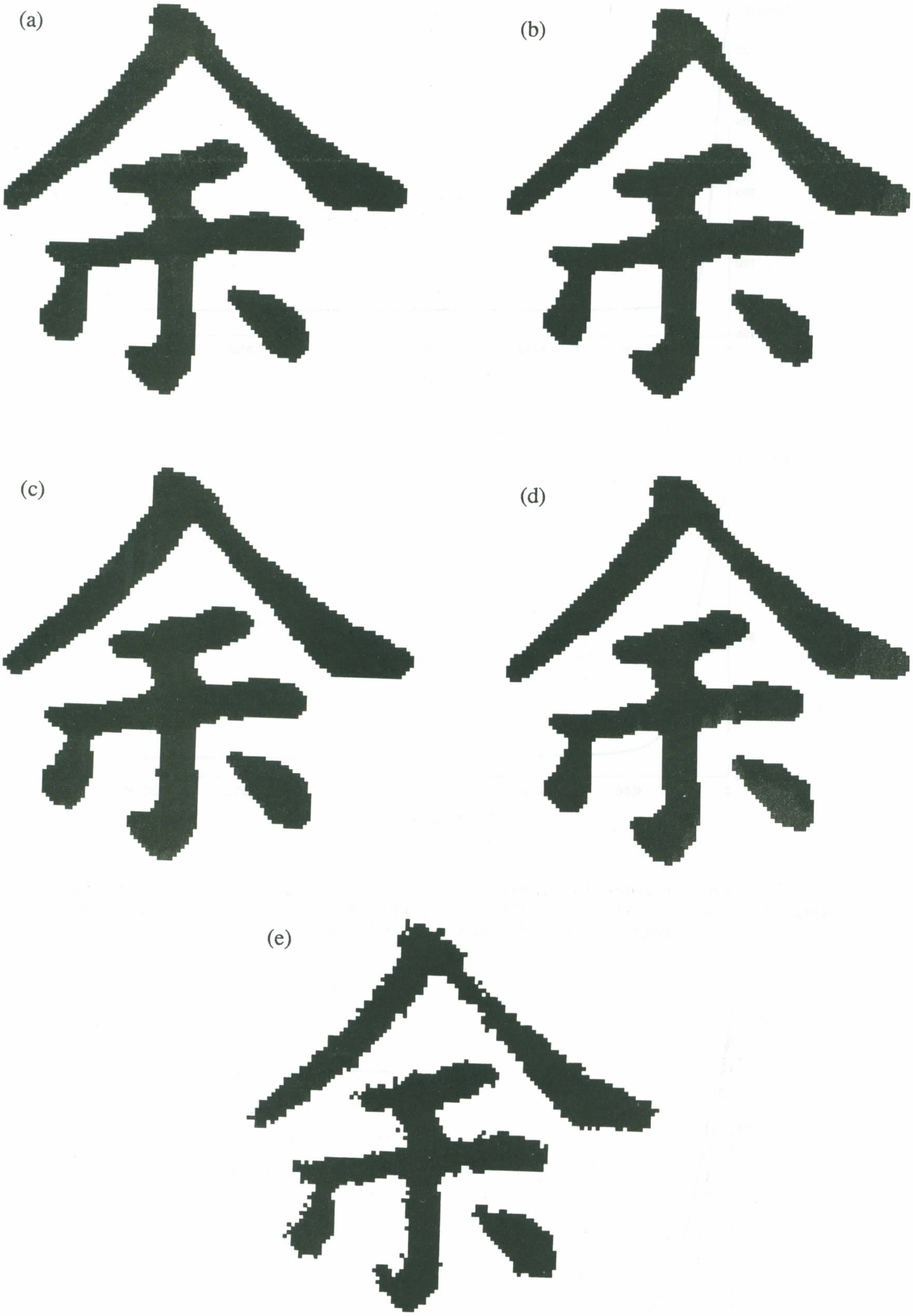
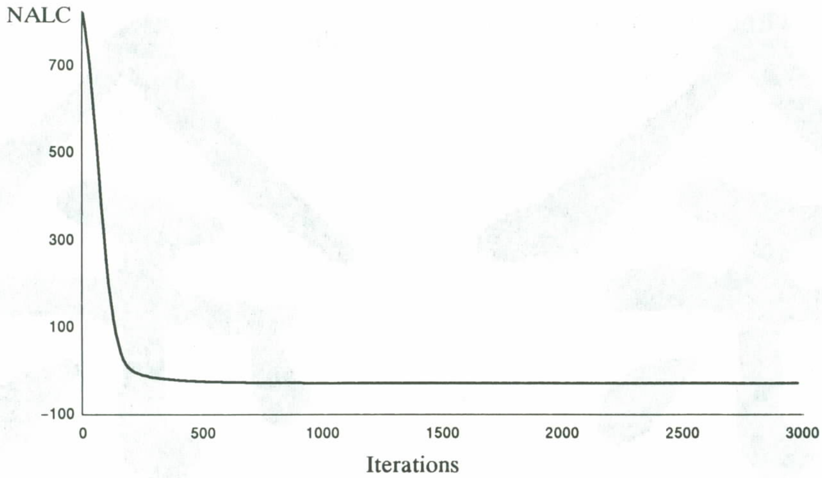
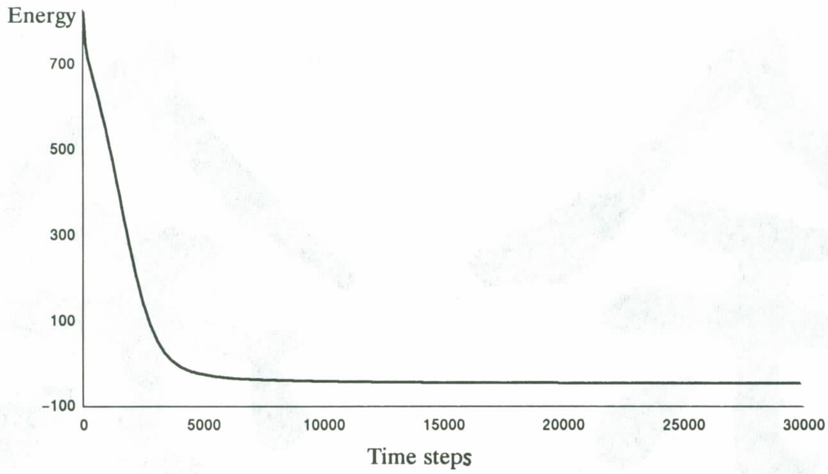


Fig. 4. Thresholding result of the relaxation method and the proposed neural network: (a) result of the relaxation method with the original compatibility coefficients; (b) result of the relaxation method with the new compatibility coefficients; (c) result of the neural network approach with gain = 100; (d) result of the neural network approach with gain = 10^4; (e) result of the neural network approach with gain = 4×10^4.



(a)



(b)

Fig. 5. Convergence behaviors of the relaxation process and the proposed network: (a) the negated average local consistency value, as a function of the iteration number; (b) the energy values of the proposed neural network, as a function of the time step.

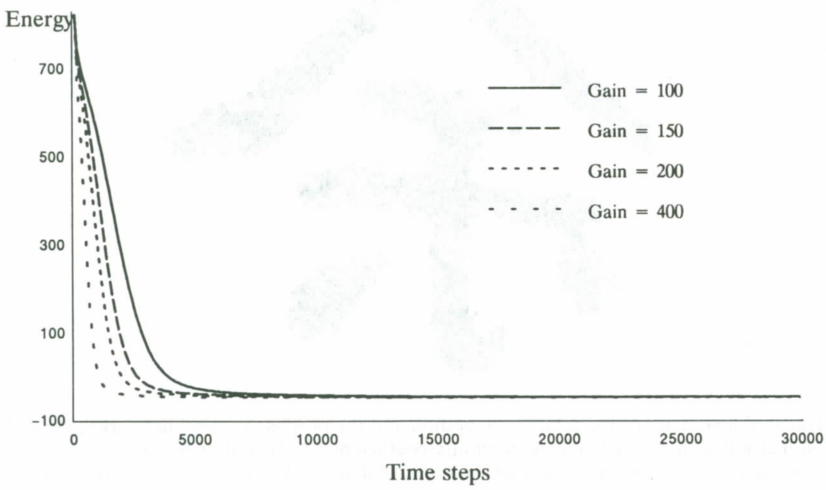


Fig. 6. The energy values at different gain values, as a function of the time step.

A mapping method that makes the Hopfield neural network perform the relaxation process is presented in this paper. In such a network the neural nodes are used to represent the various possible hypotheses and pieces of evidence in the relaxation process, and the neural links embody the relationships among them. By this method, the neural network technology can be easily adapted to solve the many problems which have already been solved by the relaxation process. The proposed mapping method is quite effective in making the Hopfield neural network perform the relaxation process as shown by the experimental results. The advantages of the neural network can thus be injected into the numerous relaxation applications. One resulting advantage is that after being mapped onto the Hopfield neural network, the relaxation process can be performed in real time, since the Hopfield network can be implemented by analog circuits. Note that a few hardware implementations of the relaxation process[33–35] have been proposed; all of them take advantage of parallel architectures (e.g. SIMD) which are digital circuits and essentially not real time.

The relaxation technique is a very important technique for many application domains; and hundreds of research papers have been published in this area. On the other hand, the neural network technology has also attracted much attention recently. Therefore, another resulting advantage of establishing the relationship between the relaxation process and the neural network technique is that research results of one domain can be very instructive to the other. For example, there exist some common properties in the convergence problem of both the relaxation process and the neural network operations.[2,26,36,37] Also, a number of works have been done for systematically generating compatibility coefficients of the relaxation process.[6,38,39] The generation of the connection weights of a neural network by teaching or learning is also an important issue in neural network research.[25,40] It seems possible to apply the results to each other. Moreover, what the relationship is between stochastic relaxation[3,41] and neural networks with stochastic properties (e.g. the Boltzmann machine and the Cauchy machine), and how to build a hierarchical neural network for the hierarchical relaxation process[3] can also be investigated in the future. Finally and maybe more basically, are the relaxation technique and the Hopfield neural network technique mathematically equivalent in certain conditions? Do the two methods converge to the same solution? If not, how do they differ? These are other interesting topics for future research.

6. SUMMARY

The relaxation process is a useful technique for using contextual information to reduce local ambiguity and achieve global consistency in various applications. It is basically a parallel execution model, adjusting the confidence measures of involved entities based on interrelated hypotheses and confidence measures. On the other hand, the neural network is a computational model with massively parallel execution capability. The output of each neuron depends mainly on the information provided by other neurons. Therefore, there exist certain common properties in the relaxation process and the neural network technique.

In each step of the relaxation process, the whole system tries to reach a better state, i.e. to increase the consistency among the constraints; therefore, if there is a goodness function which can represent the consistency between the current state and the constraints, problem solving by relaxation can be regarded as a process which increases iteratively the value of the goodness function. In this viewpoint, the Hopfield network turns out to be appropriate to perform the relaxation process because a Hopfield network tends to reach a stable state by minimizing the energy function when the network evolves.

A mapping method that makes the Hopfield neural network perform the relaxation process is proposed. In such a network the neural nodes are used to represent the various possible hypotheses and pieces of evidence in the relaxation process, and the neural links embody the relationships among them. By this method, the neural network technology can be easily adapted to solve the many problems which have already been solved by the relaxation process. The advantages of the neural network can thus be injected into the numerous relaxation applications. Another resulting advantage is that after being mapped onto the Hopfield neural network, the relaxation process can be performed in real time, since the Hopfield networks can be implemented by conventional electrical circuits. Experimental results demonstrating the feasibility of the proposed method by performing the image thresholding operation on the proposed neural network are given.

Acknowledgements—The authors thank the two anonymous referees for their very detailed comments on this paper and for their very valuable suggestions about the topics for future research.

REFERENCES

1. A. Rosenfeld, R. A. Hummel and S. W. Zucker, Scene labeling by relaxation operations, *IEEE Trans. Syst. Man Cybern.* **6**, 420–433 (1976).
2. S. W. Zucker, E. V. Krishnamurthy and R. L. Haar, Relaxation processes for scene labeling: convergence, speed, and stability, *IEEE Trans. Syst. Man Cybern.* **8**, 41–48 (1978).
3. B. Bhanu and O. D. Faugeras, Shape matching of 2-d object using a hierarchical stochastic labeling technique, *Proc. Conf. Pattern Recognition Image Process.*, Las Vegas, NV, pp. 688–690, June (1982).
4. C. Wang, H. Sun, S. Yada and A. Rosenfeld, Some

experiments in relaxation image matching using corner features, *Pattern Recognition* **16**, 167–182 (1983).
5. S. W. Zucker, R. A. Hummel and A. Rosenfeld, An application of relaxation labeling to line and curve enhancement, *IEEE Trans. Comput.* **26**, 394–403 (1977).
6. S. Peleg and A. Rosenfeld, Determining compatibility coefficients for curve enhancement relaxation processes, *IEEE Trans. Syst. Man Cybern.* **8**, 548–555 (1978).
7. H. Ogawa and K. Taniguchi, On machine recognition of hand-printed Chinese characters by feature relaxation, *Pattern Recognition* **21**, 1–7 (1988).
8. I. Sekita, K. Toraichi, R. Mori, K. Yamamoto and H. Yamada, Feature extraction of hand-written Japanese characters by spline functions for relaxation, *Pattern Recognition* **21**, 9–17 (1988).
9. Shiaw-Shian Yu and Wen-Hsiang Tsai, A new thinning algorithm for gray-scale images by the relaxation technique, *Pattern Recognition* **23**, 1067–1076 (1990).
10. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Vol. 2. Academic Press, New York (1982).
11. J. J. Hopfield, Neural networks and physical systems with emergent collective computation abilities, *Proc. natn. Acad. Sci. USA* **79**, 2554–2558 (April 1982).
12. J. J. Hopfield, Neurons with graded response have collective computational properties like those of two-state neurons, *Proc. natn. Acad. Sci. USA* **81**, 3088–3092 (1984).
13. J. J. Hopfield and D. W. Tank, Neural computation of decisions in optimization problems, *Biol. Cybern.* **52**, 141–152 (1985).
14. Y. Yong, Handprinted Chinese character recognition via neural networks, *Pattern Recognition Lett.* **7**, 19–25 (1988).
15. K. Fukushima, Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biol. Cybern.* **36**, 193–202 (1980).
16. K. Fukushima, Neural network model for selective attention in visual pattern recognition and associative recall, *Appl. Opt.* **26**, 4985–4992 (1987).
17. K. Fukushima, Neocognitron: a hierarchical neural network capable of visual pattern recognition, *Neural Networks* **1**, 119–130 (1988).
18. T. Kohonen, *Self-organization and Associative Memory*. Springer-Verlag, New York (1984).
19. S. G. Grossberg, Adaptive pattern classification and universal recording: I. Parallel development and coding of neural feature detectors, *Biol. Cybern.* **23**, 121–134 (1976).
20. S. G. Grossberg, How does a brain build a cognitive code? *Psychol. Rev.* **87**, 1–51 (1980).
21. S. E. Fahlman and G. E. Hinton, connectionist architectures for artificial intelligence, *Computer*, **20**(1), 100–109 (January 1987).
22. J. A. Feldman and D. H. Ballard, Connectionist models and their properties, *Cognitive Sci.* 205–254 (1982).
23. D. E. Rumelhart, J. L. McClelland and the PDP research group, *Parallel Distributed Processing* (2 volumes). MIT Press, Cambridge, MA (1986).
24. J. L. McClelland and D. E. Rumelhart, *Explorations in Parallel Distributed Processing*. MIT Press, Cambridge, MA (1988).
25. C. T. Genis, Relaxation and neural learning: points of convergence and divergence, *J. Parallel Distributed Comput.* **6**, 217–244 (1989).
26. M. A. Cohen and S. G. Grossberg, Absolute stability of global pattern formation and parallel memory storage by competitive neural networks, *IEEE Trans. Syst. Man Cybern.* **13**, 815–826 (1983).
27. J. J. Hopfield and D. W. Tank, Computing with neural circuits: a model, *Science* **233**, 625–633 (1986).
28. J. Hutchinson, C. Koch, J. Luo and C. Mead, Computing motion using analog and binary resistive networks, *Computer*, **21**(3), 52–63 (March 1988).
29. C. Mead, *Analog VLSI and Neural Systems*. Addison-Wesley, Reading, MA (1989).
30. R. A. Hummel and S. W. Zucker, On the foundations of relaxation labeling, *IEEE Trans. Pattern Anal. Mach. Intell.* **5**, 267–287 (1983).
31. G. Fekete, Relaxation: evaluation and applications, *IEEE Trans. Pattern Anal. Mach. Intell.* **3**, 459–469 (1981).
32. A. Rosenfeld and R. C. Smith, Thresholding using relaxation, *IEEE Trans. Pattern Anal. Mach. Intell.* **3**, 598–606 (1981).
33. M. Kamada, K. Toraichi and R. Mori, A parallel architecture for relaxation operations, *Pattern Recognition* **22**, 175–181 (1989).
34. J. S. Duncan and W. Frei, Relaxation using modular operators, *Proc. SPIE Conf. Appl. Digital Image Process.*, Switzerland, pp. 153–164 (1983).
35. T. J. Willet, C. W. Brooks and G. E. Tisdale, Relaxation, systolic arrays and universal arrays, *Image Understanding Proc.*, pp. 164–170, April (1979).
36. M. W. Hirsch, Convergence in neural nets, *Proc. IEEE Int. Conf. Neural Networks*, San Diego, CA, Vol. 2, pp. 115–126, June (1987).
37. J. Bruck and J. W. Goodman, A generalized convergence theory for neural networks, *IEEE Trans. Inf. Theory* **34**(5), 1089–1092 (1988).
38. J. Kittler, Compatibility and support functions in probabilistic relaxation, *Proc. 8th Int. Conf. Pattern Recognition*, Paris, France, pp. 186–189, October (1986).
39. H. Yamamoto, A method of deriving compatibility coefficients for relaxation operators, *Comput. Graphics Image Process.* **10**, 256–271 (1979).
40. D. E. Rumelhart, G. E. Hinton and R. Williams, Learning internal representations by error propagation, *Parallel Distributed Processing*, D. E. Rumelhart, J. L. McClelland and the PDP research group, eds, Vol. 2, pp. 318–362. MIT Press, Cambridge, MA (1986).
41. S. Geman and D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 721–741 (1984).

About the Author—SHIAW-SHIAN YU was born on 20 June 1959 in Taiwan, Republic of China. He received his B.S. and M.S. degrees in Information Engineering and a Ph.D. degree in Information Engineering and Computer Science, all from National Chiao Tung University, Hsinchu, Taiwan, in 1981, 1983 and 1990, respectively.

From November 1983 to December 1988 he worked with the Electronics Research and Service Organization, Industrial Technology Research Institute, as an Associate Researcher. Now he is a Researcher and the leader of the Intelligent Man–Machine Interface Research Group of the Advanced

Technology Center, Computer and Communication Research Laboratories, Industrial Technology Research Institute. He is also an adjunct associate professor of the Department of Information Engineering, Yuan Ze Institute of Technology. He is a member of the IEEE and the Computer Soceity. His current research interests include pattern recognition, image processing, neural networks, Chinese character recognition, and Chinese speech recognition.

About the Author—WEN-HSIANG TSAI was born on 10 May 1951 in Tainan, Taiwan, Republic of China. He received the B.S. degree from National Taiwan University in 1973, an M.S. degree from Brown University in 1977, and his Ph.D. degree from Purdue University in 1979, all in Electrical Engineering.

From 1973 to 1975, he served in the Chinese Navy as an electronics officer. Since November 1979, he has been on the faculty of the Institute of Computer Science and Information Engineering at National Chiao Tung University, Hsinchu, Taiwan. In the academic year of 1983–1984, he was a visiting scholar at the University of Illinois at Chicago. From 1984 to 1986, he was an Assistant Director and later an Associate Director of Microelectronics and Information Science and Technology Research Center at National Chiao Tung University. He joined the Department of Computer and Information Science at National Chiao Tung University in August 1984, acted as the Head of the Department from 1984 to 1988, and is currently Professor there. He is also a Researcher in the Institute of Information Science at Academia Sinica, Taipei, Taiwan. His current research interests include pattern recognition, computer vision, image procession and Chinese information processing.

Dr Tsai is an associate editor of *Pattern Recognition*, *International Journal of Pattern Recognition and Artificial Intelligence*, *Journal of Chinese Institute of Engineers*, *Journal of Information Science and Engineering*, and *Processings of National Science Council* of the Republic of China. He was elected as an Outstanding Talent of Information Science of the Republic of China in 1986. He was the winner of the 13th Annual Best Paper Award of the Pattern Recognition Society. He obtained the 1987 Outstanding Research Award and the 1988–1991 Distinguished Research Awards, all of the National Science Council of the Republic of China. He also obtained the 1989 Distinguished Teaching Award of the Ministry of Education of the Republic of China. He was the winner of the 1990 Outstanding Paper Award of the Computer Society of the Republic of China. Dr Tsai has published more than 45 papers in well-known international journals.