Pergamon

Neural Networks, Vol. 9, No. 4, pp. 671–684, 1996

0893-6080/96 $15.00+.00

0893-6080(95)00111-5

CONTRIBUTED ARTICLE

Self-Organization Using Potts Models

CHENG-YUAN LIOU AND JIANN-MING WU

National Taiwan University

(*Received* 21 *June* 1994; *revised and accepted* 5 *September* 1995)

Abstract—*In this work, we use Potts neurons for the competitive mechanism in a self-organization model. We obtain new algorithms on the basis of a Potts neural network for coherent mapping, and we remodel the Durbin algorithm and the Kohonen algorithm with mean field annealing. The resulting dimension-reducing mappings possess a highly reliable topology preservation such that the nearby elements in the parameter space are ordered as similarly as possible on the cortex-like map, and the objective function costs between neighboring cortical points are as smooth as possible. The proposed Potts neural network contains two sets of interactive dynamics for two kinds of mappings, one from the parameter space to the cortical space and the other in the reverse way. We present a theoretical approach to developing self-organizing algorithms with a novel decision principle for competitive learning. We find that one Potts neuron is able to implement the Kohonen algorithm. Both implementation and simulation results are encouraging. Copyright ©1996 Elsevier Science Ltd*

Keywords—Self-organization map, Neural network, Potts model, Elastic ring, Mean field annealing, Hairy model.

1. INTRODUCTION

The idea of self-organization is an important principle in many powerful neural systems in solving complex tasks, such as vector quantization (Kohonen, 1982), speech recognition (Kohonen, 1988), combinatorial optimization (Angeniol et al., 1988; Durbin & Willshaw, 1987), and formation of ocular dominance stripes (Durbin & Mitchison, 1990; Goodhill & Willshaw, 1990). Two typical examples of such self-organizing algorithms are the Kohonen algorithm (Kohonen, 1988) and the Durbin algorithm (Durbin & Mitchison, 1990). These methods produce cortex-like maps from the many-dimensional parameter space to the surface of the cortex with the property of topology preservation. The surface of the cortex is realized by a lattice structure with a cortical point or receptive field on each node site. By the Conwey argument (Conwey, 1979), the principle of self-organization can be stated as that the neighboring elements in parameter space should map closely together on the cortex. The two self-organizing algorithms we develop operate explicitly from the cortex to parameter space. They assign parameter values to cortical points, and attempt to put nearby cortical points close in parameter space. In general, the algorithms contain a competitive sharing-out of the domain of inputs among units of the cortex and a continuity constraint for a coherent mapping. To obtain the competitive interactions between units, the Durbin algorithm uses the normalized Gaussian activation function, for which the response property is characterized by a control parameter. When the control parameter is set to one extreme, all units equally respond and when to the other extreme, they act as the winner-take-all, which is the exact mechanism used in the Kohonen algorithm. For the continuity constraint the Kohonen algorithm uses a dynamical neighboring structure, which is initially set to be large and then monotonically decreased, whereas the Durbin algorithm uses a static neighboring structure as defined by the lattice.

The central idea of this work is to use the Potts neurons as a competitive mechanism for self-organization. Following this idea, we incorporate essentials of self-organization into energy functions

Acknowledgements: This work is supported by the National Science Council under project NSC 84-2213-E-002-012.

Requests for reprints should be sent to Cheng-Yuan Liou, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 10764, Republic of China; e-mail: cyliou@csie.ntu.edu.tw.

and derive dynamics for each component by mean field annealing. Along the annealing process, the states of Potts neurons will converge toward the global or near global minimum of the energy function and internally represent the desired coherent mapping. The self-organization will be naturally derived from the competitive mechanism, and the effective objective and continuity constraints.

Yuille et al. (1991) devised a generalized deformable model to encode criteria of the development of ocularity. In their work, based on an energy function containing both binary and continuous variables, one can use analytical statistical techniques (Yuille, 1990) to derive two different algorithms for ocular formation, the elastic net algorithms of Durbin and Willshaw (1987), Durbin and Mitchison (1990), Goodhill and Willshaw (1990), and the model of Miller et al. (1989) and Tanaka (1990). In current work, the energy functions used for encoding the criteria of self-organization also contain two sets of variables, one binary and the other continuous. The difference from all previous works (Miller et al., 1989; Durbin & Mitchison 1990; Tanaka, 1990; Yuille, 1990; Yuille et al., 1991) is that we discretize the continuous variables into the feature space and use a joint free energy to characterize two sets of variables and further theoretically derive mean field equations for both sets of variables. This leads to Potts models with coupling dynamics. This idea has been applied to treat the traveling salesman problem in the thesis (Wu, 1994) and successfully unifies three typical neural approaches to the traveling salesman problem. They are the elastic ring method (Durbin & Willshaw, 1987), the Hopfield neural network (Hopfield & Tank, 1985) and the Peterson and Söderberg's Potts encoding (1989).

In accordance with the different operation mode of the two algorithms, we classify the self-organizing algorithms into two types, the batch type and the on-line type. The former has prior knowledge about the full description of the parameter space so that the algorithm processes inputs in batch to make use of explicit information among all parameter elements. The Durbin algorithm is the batch-type algorithm. The on-line algorithms lack such prior knowledge so that they process inputs one by one. No immediate information between parameter elements can be used. The Kohonen algorithm belongs to the on-line type of algorithm.

In the next section, we will formulate models for both types of algorithms. For the batch algorithm, we simultaneously maintain two essential mappings, one from the parameter space to the cortex and the other in the opposite way. The two mappings work together so that the nearby elements in the parameter space are ordered as closely as possible on the cortex and the wiring difference between neighboring cortical points are maximally reduced to meet the continuity constraints. We formulate the two mappings as an optimization model and then devise an energy function to incorporate them. By mean field annealing, we derive two sets of mean field equations for the two mappings. The Durbin algorithm is then shown to be a special form of the new batch algorithm. For the on-line algorithm, our model greatly reduces the complexity of the algorithm so that one Potts neuron is sufficient for the required computation. The new on-line algorithm is experimentally shown to be consistent with the Kohonen algorithm in competitive decision making. In Section 3, we generally state our algorithms in the forms of hairy models proposed by Szu (1989) to guarantee the convergence properties of our algorithms. We list our simulation results and conclude our work in the final section.

2. DESIGNING SELF-ORGANIZING ALGORITHM BY MEAN FIELD ANNEALING

In this section, we first derive the new batch-type self-organizing algorithm in Subsection 2.1, and then use our model to examine the Durbin algorithm in Subsection 2.2. The on-line self-organizing algorithm is developed in Subsection 2.3.

2.1. The New Batch-Type Self-Organizing Algorithm

In this subsection, a topology preserving mapping from K-dimensional parameter space, $\Omega = \{\mathbf{x}_i, 1 \leqslant i \leqslant N\}$ and $\mathbf{x}_i = [x_{i1}, \ldots, x_{iK}]^t$, to a two-dimensional lattice structure is first formulated using an energy function and then incorporated into Potts neural networks. The lattice has the size $M \times M$ and each site (a, b), $1 \leqslant a, b \leqslant M$, has four neighbors, $(a-1, b)$, $(a+1, b)$, $(a, b-1)$, $(a, b+1)$, collectively denoted by $\mathscr{B}_{ab}$; "0" and "$M+1$" in site index are automatically equivalent to "M" and "1" respectively for the circular representation of the lattice to smooth the edge effect of the lattice plane. Each site (a, b) defines a cortical point $\mathbf{y}_{ab}$ which has the same dimensions as each parameter element. In the general case, the total number of the cortical points is much less than the size of the parameter space. That is $M^2 \ll N$ in the usual case but in our formulation it is also valid for M^2 to be larger than N in recognition of the fact that the number of cortex neurons is much larger than necessary. We will describe the modified method where $M^2 \gg N$ briefly.

In the general case, to obtain a valid mapping, we enforce the constraint that (i) each element in the parameter space should be mapped to only one site, (ii) each cortical point should be assigned to one parameter element, and (iii) no two cortical points are assigned to the same parameter element.

The second and third constraints lead to the competitive sharing-out of the cortical points among parameter space. In computation, we need a batch process of parameter elements for such competitive interaction. When all three constraints are satisfied, we have selected M^2 different representatives, each for a cortical point, among N parameter elements and mapped each element to one representative, which is closest to the element. An underlying neighboring structure for the M^2 representatives is also formed as the lattice structure. Each representative should have four neighboring representatives which are correspondingly attached to the four neighboring sites on the lattice. Two explicit quantities that have been equipped for self-organization (Durbin & Mitchison, 1990) are used to qualify the valid mappings. The first quantity, L_1, is the sum of the distance between each element and its corresponding representative and the second quantity, L_2, is the sum of the distance between any two neighboring representatives. Simultaneous minimization of these two quantities aims at both putting all nearby parameter elements as close as possible on the lattice sites and assigning neighboring cortical points close in the parameter space. The latter goal obtains a smooth distribution of the cortical points on the lattice. With the constraints and the minimizing objectives, we can formulate the self-organizing algorithm in terms of energy functions as in the Hopfield model. In the unusual case of $M^2 > N$, the third constraint is then altered to say that two parameter elements should not be mapped to the same cortical point which implies a competitive sharing-out of the parameter elements among the cortical nodes. Formulating the energy function for each case uses the same technology. The unusual case can be easily translated to the usual case by duplicating the parameter elements and by adding very small random noise to each parameter element evenly. Without losing generality, we only need discuss the formulation for the general case as follows.

Let $\boldsymbol{\sigma}_i$ denote a unitary vector with M^2 components, $\boldsymbol{\sigma}_i = [\sigma_{i1}, \ldots, \sigma_{iM^2}]^t$, $1 \leqslant i \leqslant N$, $\sigma_{i\alpha} \in \{0, 1\}$. The only active bit, for example the αth bit, among M^2 components is used to indicate that the ith parameter element is mapped to the site (a, b) or (α) with $\alpha = (a-1) \times M + b$. In notation, the site index can appear in one or two dimensions. When appearing in subscript, in the following context, α and ab denote the same thing, as $\alpha = (a-1) \times M + b$ is satisfied, for instance, $\mathbf{y}_\alpha = \mathbf{y}_{ab}$, $\mathscr{B}_\alpha = \mathscr{B}_{ab}$. Let $\mathbf{q}_\alpha = [q_{\alpha 1}, \ldots, q_{\alpha N}]^t$ also be a unitary vector. The only active component $q_{\alpha i}$ indicates that the ith parameter element is occupied by the cortical point $\mathbf{y}_\alpha$. The sets $\{\boldsymbol{\sigma}_i\}$ and $\{\mathbf{q}_\alpha\}$ constitute two elementary mappings, one from the parameter space to the lattice, the other in the opposite way. Then the self-organizing algorithm is modeled by minimizing

$$\begin{aligned} L &= L_1 + L_2 \\ &= C \sum_{1 \leqslant i \leqslant N} \sum_{1 \leqslant \alpha \leqslant M^2} \sigma_{i\alpha} |\mathbf{x}_i - \mathbf{y}_\alpha| \\ &\quad + \sum_{1 \leqslant \alpha \leqslant M^2} \sum_{\gamma \in \mathscr{B}_\alpha} |\mathbf{y}_\alpha - \mathbf{y}_\gamma|, \end{aligned} \tag{1}$$

subject to

$$(a) \sum_{1 \leqslant \alpha \leqslant M^2} \sigma_{i\alpha} = 1,\ 1 \leqslant i \leqslant N,$$

$$\text{all } \sigma_{i\alpha} \in \{0, 1\} \text{ [as stated in (i)]}$$

$$(b) \sum_{1 \leqslant i \leqslant N} q_{\alpha i} = 1,\ 1 \leqslant \alpha \leqslant M^2,$$

$$\text{all } q_{\alpha i} \in \{0, 1\} \text{ [as stated in (ii)]}$$

$$(c) \sum_{1 \leqslant \alpha \leqslant M^2} q_{\alpha i} = 0 \text{ or } 1,\ 1 \leqslant i \leqslant N \text{ [as stated in (iii)]}$$

$$(d)\ \mathbf{y}_\alpha = \sum_{1 \leqslant i \leqslant N} q_{\alpha i} \mathbf{x}_i \text{ [as stated in (ii)]}. \tag{2}$$

The self-organizing algorithm now turns to find the sets $\{\boldsymbol{\sigma}_i\}$ and $\{\mathbf{q}_\alpha\}$ satisfying all constraints and minimizing the objective L. In the constraint (d), $\mathbf{y}_\alpha$ is expressed as a linear combination of all parameter elements. This representation discretizes the domain of the cortical points. In conventional self-organizing algorithms (Durbin & Willshaw, 1987; Kohonen, 1988), the cortical points are dynamically continuously traced within the space spanned by all parameter elements. The final position of a cortical point may not be at the position of any parameter element. In the current algorithm, we will apply the mean field annealing to find the mappings $\{\boldsymbol{\sigma}_i\}$ and $\{\mathbf{q}_\alpha\}$. At each intermediate temperature, the two mappings are represented in terms of the mean configurations $\{\langle\boldsymbol{\sigma}_i\rangle\}$ and $\{\langle\mathbf{q}_\alpha\rangle\}$. During the annealing process, the intermediate cortical points $\langle\mathbf{y}_\alpha\rangle = \Sigma_{1 \leqslant i \leqslant N} \langle q_{\alpha i}\rangle \mathbf{x}_i$ are also dynamically traced among the space spanned by all parameter elements. However we expect each cortical point $\mathbf{y}_\alpha$ will finally stand at the position of some parameter element at the end of the annealing process, since the constraint (b) has encoded the winner-take-all principle in the model when the temperature parameter is reduced sufficiently low. Constraint (c) says that no two cortical points should occupy the same parameter elements. This leads to a competitive sharing-out of the cortical points among the parameter space, which is a critical feature in designing a batch-type self-organizing algorithm.

The terms in model (2) are combined to obtain an energy function. The constraints (*a*) and (*b*) are considered as the unitary conditions of the Potts neurons for vectors $\boldsymbol{\sigma}_i$ and $\mathbf{q}_\alpha$ correspondingly. They will be implicitly embedded within the activation function of the Potts neurons. Using Lagrange multipliers, the following energy function sums up the objective L, constraints (*a*), (*b*) and (*c*), and reserves the notation $\mathbf{y}_\alpha$.

$$\begin{aligned}
H(\sigma, q) = {} & C \sum_i \sum_\alpha \sigma_{i\alpha} |\mathbf{x}_i - \mathbf{y}_\alpha| \\
& + \sum_\alpha \sum_{\gamma \in \mathcal{B}_\alpha} |\mathbf{y}_\alpha - \mathbf{y}_\gamma| + \frac{B}{2} \sum_i \left(\sum_\alpha q_{\alpha i} \right)^2 \\
& + \frac{A}{2} \sum_\alpha \sum_i \sum_{j \neq i} q_{\alpha i} q_{\alpha j} \\
& + \frac{A}{2} \sum_i \sum_\alpha \sum_{\beta \neq \alpha} \sigma_{i\alpha} \sigma_{i\beta} \\
= {} & C \sum_i \sum_\alpha \sigma_{i\alpha} |\mathbf{x}_i - \mathbf{y}_\alpha| \\
& + \sum_\alpha \sum_{\gamma \in \mathcal{B}_\alpha} |\mathbf{y}_\alpha - \mathbf{y}_\gamma| \\
& - \frac{A}{2} \sum_{\alpha i} q_{\alpha i}^2 - \frac{A}{2} \sum_{i\alpha} \sigma_{i\alpha}^2 + \frac{B}{2} \sum_i \left(\sum_\alpha q_{\alpha i} \right)^2 \\
& + \frac{A}{2} (N + M^2), \qquad (3)
\end{aligned}$$

where σ and q denote the collections of all $\boldsymbol{\sigma}_i$ and $\mathbf{q}_\alpha$ respectively, the index i runs from 1 to N, α and γ range between 1 and M^2, and A, B, and C are Lagrange multipliers. The constant term is negligible in (3). The two A terms in the second line of (3) are used to represent the constraint terms (*a*) and (*b*) and are further reduced to these simple forms based on the unitary conditions of $\mathbf{q}_\alpha$ and $\boldsymbol{\sigma}_i$. The B term represents the constraint (*c*) to prevent more than one cortical point from being assigned to the same parameter element. In (3), the vectors $\mathbf{y}_\alpha$ are not resolved by the constraint term (*d*) in model (2) until the derivation of the mean field equations for the energy function in the latter context, where we need to calculate the means of all terms containing vectors $\mathbf{y}_\alpha$.

We apply the mean field annealing (MFA) to minimize the energy function (3). The minimization includes two steps, each for one kind of variable. We briefly review the mean field annealing for optimizations. The mean field annealing has been shown powerful for optimizations (Peterson & Söderberg, 1989). For a system, of which the energy function is well defined as a function of the system configuration, the MFA attempts to find the mean configuration under thermal equilibrium at each temperature. The temperature parameter is initially set high and then slowly scheduled down. When applying the MFA to minimize an energy function, such as $H(s)$, where s denotes the system configuration with variables $\{\mathbf{s}_i\}$, we first characterize the MFA by the common formal free energy function $\psi(u, v, \beta)$ (Peterson & Söderberg, 1989):

$$\begin{aligned}
\psi(u, v, \beta) &= \langle H(s) \rangle + \sum_i \mathbf{v}_i^t \mathbf{u}_i - \frac{1}{\beta} \sum_i \ln z(\mathbf{u}_i, \beta) \\
z(\mathbf{u}_i, \beta) &= \sum_\alpha \exp(\beta u_{i\alpha}),
\end{aligned} \qquad (4)$$

where $\langle \cdot \rangle$ denote the mean operator. We then determine the mean field configuration $\langle s_i \rangle$ at each control temperature $T = 1/\beta$ as the stationary point of the free energy function:

$$\frac{\partial \psi}{\partial v_{i\alpha}} = 0 \Rightarrow u_{i\alpha} = \frac{\partial \langle H(s) \rangle}{\partial v_{i\alpha}} \qquad (5)$$

$$\frac{\partial \psi}{\partial u_{i\alpha}} = 0 \Rightarrow v_{i\alpha} \equiv \langle s_{i\alpha} \rangle = \frac{\exp(\beta u_{i\alpha})}{\sum_\beta \exp(\beta u_{i\beta})}. \qquad (6)$$

We use the following procedures to operate the MFA for our model (3). The formulation of our procedure is similar to the hairy model developed by Szu (1989). We will develop two sets of mean field equations for these two kinds of variables, $\{\boldsymbol{\sigma}_i\}$ and $\{\mathbf{q}_\alpha\}$, within the energy function $H(\sigma, q)$ in (3). When we fix one kind of variable at its mean configuration under thermal equilibrium, we can obtain a set of the mean field equations similar to eqns (5) and (6) for the mean configuration of the other kind of variable. The two sets of the mean field equations are then executed in turn to determine the means of the two mappings. Let $H_\sigma(\sigma, q)$ denote the energy function when each variable $\mathbf{q}_\alpha$ is fixed at its mean $\langle \mathbf{q}_\alpha \rangle$ and the quantities involving $\mathbf{y}_\alpha$ are also fixed at their means. We have:

$$\begin{aligned}
H_\sigma(\sigma, q) = {} & C \sum_i \sum_\alpha \sigma_{i\alpha} \langle |\mathbf{x}_i - \mathbf{y}_\alpha| \rangle \\
& + \sum_\alpha \sum_{\gamma \in \mathcal{B}_\alpha} \langle |\mathbf{y}_\alpha - \mathbf{y}_\gamma| \rangle - \frac{A}{2} \sum_{\alpha i} \langle q_{\alpha i} \rangle^2 \\
& + \frac{B}{2} \sum_i \left(\sum_\alpha \langle q_{\alpha i} \rangle \right)^2 - \frac{A}{2} \sum_{i\alpha} \sigma_{i\alpha}^2. \qquad (7)
\end{aligned}$$

The mean of the energy function $H_\sigma(\sigma, q)$ is derived as follows.

$$\begin{aligned}
H_\sigma(\sigma, q) = {} & C \sum_i \sum_\alpha \langle \sigma_{i\alpha} \rangle \langle |\mathbf{x}_i - \mathbf{y}_\alpha| \rangle \\
& + \sum_\alpha \sum_{\gamma \in \mathcal{B}_\alpha} \langle |\mathbf{y}_\alpha - \mathbf{y}_\gamma| \rangle - \frac{A}{2} \sum_{\alpha i} \langle q_{\alpha i} \rangle^2 \\
& + \frac{B}{2} \sum_i \left(\sum_\alpha \langle q_{\alpha i} \rangle \right)^2 - \frac{A}{2} \sum_{i\alpha} \langle \sigma_{i\alpha} \rangle. \qquad (8)
\end{aligned}$$

In the above derivation, the last term $\langle \sigma_{i\alpha}^2 \rangle$ is substituted by $\langle \sigma_{i\alpha} \rangle$ since in the Bernoulli distribution:

$$\begin{aligned}\langle \sigma_{i\alpha}^2 \rangle &= 0^2 P(\sigma_{i\alpha} = 0) + 1^2 P(\sigma_{i\alpha} = 1) \\ &= P(\sigma_{i\alpha} = 1) = \langle \sigma_{i\alpha} \rangle.\end{aligned} \tag{9}$$

By substituting the mean energy function $\langle H_\sigma(\sigma, q) \rangle$ into the $\langle H(s) \rangle$ term in the free energy (4) and setting the partial derivatives of the free energy with respect to $v_{i\alpha}$ and $u_{i\alpha}$ to zero, we can obtain the following mean field equations for each variable $\sigma_{i\alpha}$.

$$u_{i\alpha} = -C\langle |\mathbf{x}_i - \mathbf{y}_\alpha| \rangle + \frac{A}{2}, \tag{10}$$

$$v_{i\alpha} \equiv \langle \sigma_{i\alpha} \rangle = \frac{\exp(\beta_1 u_{i\alpha})}{\sum_\gamma \exp(\beta_1 u_{i\gamma})}, \tag{11}$$

where the form of $\langle |\mathbf{x}_i - \mathbf{y}_\alpha| \rangle$ will be determined later in developing the mean field equations for the other set of variables $\{\mathbf{q}_\alpha\}$, and the control parameter is denoted by β_1.

We then consider the following energy function, which is obtained by fixing the variables $\{\boldsymbol{\sigma}_i\}$ at their mean values:

$$\begin{aligned}H_q(\sigma, q) = {} & C \sum_i \sum_\alpha \langle \sigma_{i\alpha} \rangle |\mathbf{x}_i - \mathbf{y}_\alpha| \\ & + \sum_\alpha \sum_{\gamma \in \mathscr{B}_\alpha} |\mathbf{y}_\alpha - \mathbf{y}_\gamma| - \frac{A}{2} \sum_{\alpha i} q_{\alpha i}^2 \\ & + \frac{B}{2} \sum_i \left(\sum_\alpha q_{\alpha i} \right)^2 - \frac{A}{2} \sum_{\alpha i} \langle \sigma_{i\alpha} \rangle^2.\end{aligned} \tag{12}$$

The mean energy function $\langle H_q(\sigma, q) \rangle$ is approximated as

$$\begin{aligned}\langle H_q(\sigma, q) \rangle = {} & C \sum_i \sum_\alpha \langle \sigma_{i\alpha} \rangle \sum_k \langle q_{\alpha k} \rangle |\mathbf{x}_i - \mathbf{x}_k| \\ & + \sum_\alpha \sum_{\gamma \in \mathscr{B}_\alpha} \sum_{jk} \langle q_{\alpha j} \rangle \langle q_{\gamma k} \rangle |\mathbf{x}_j - \mathbf{x}_k| \\ & - \left(\frac{A}{2} - \frac{B}{2} \right) \sum_{\alpha i} \langle q_{\alpha i} \rangle^2 \\ & + B \sum_i \sum_\alpha \sum_{\beta \neq \alpha} \langle q_{\alpha i} \rangle \langle q_{\beta i} \rangle \\ & - \frac{A}{2} \sum_{\alpha i} \langle \sigma_{i\alpha} \rangle^2.\end{aligned} \tag{13}$$

The middle three terms on the right side in eqn (13) are based on the property in eqn (9) and the strong independent assumption that the joint probability distribution of $q_{\alpha j}$ and $q_{\gamma k}$, $P(q_{\alpha j}, q_{\gamma k})$, is separable, $P(q_{\alpha j}, q_{\gamma k}) = P(q_{\alpha j}) P(q_{\gamma k})$, or that the two variables $q_{\alpha j}$ and $q_{\gamma j}$ are independent. That is,

$$\begin{aligned}\langle q_{\alpha j} q_{\gamma k} \rangle &= \int \int q_{\alpha j} q_{\gamma k} P(q_{\alpha j}, q_{\gamma k}) dq_{\alpha j} dq_{\gamma k} \\ &= \int \int q_{\alpha j} q_{\gamma k} P(q_{\alpha j}) P(q_{\gamma k}) dq_{\alpha j} dq_{\gamma k} \\ &= \int q_{\alpha j} P(q_{\alpha j}) dq_{\alpha j} \int q_{\gamma k} P(q_{\gamma k}) dq_{\gamma k} \\ &= \langle q_{\alpha j} \rangle \langle q_{\gamma k} \rangle.\end{aligned} \tag{14}$$

Then we insert the constraint (*d*) in deriving (13) by substituting the quantity $\langle |\mathbf{x}_i - \mathbf{y}_\alpha| \rangle$ with $\Sigma_k \langle q_{\alpha k} \rangle |\mathbf{x}_i - \mathbf{x}_k|$. In the same way as for the variables $\{\boldsymbol{\sigma}_i\}$, the free energy and the mean field equations for the variables $\{\mathbf{q}_\alpha\}$ can be obtained. The mean field equations are:

$$\begin{aligned}w_{\alpha i} = {} & \frac{A}{2} - \frac{B}{2} - B \sum_{\gamma \neq \alpha} p_{\gamma i} \\ & - \sum_l \left(C \langle \sigma_{l\alpha} \rangle + \sum_{\gamma \in \mathscr{B}_\alpha} p_{\gamma l} \right) |\mathbf{x}_l - \mathbf{x}_i|,\end{aligned} \tag{15}$$

$$p_{\alpha i} \equiv \langle q_{\alpha i} \rangle = \frac{\exp(\beta_2 w_{\alpha i})}{\sum_\gamma \exp(\beta_2 w_{\gamma i})}. \tag{16}$$

The two sets of mean field equations (10), (11) and (15), (16) are used to compute the means of two elementary mappings, $\{\boldsymbol{\sigma}_i\}$ and $\{\mathbf{q}_\alpha\}$. The annealing process is controlled by two temperature parameters,

$$T_1 = \frac{1}{\beta_1}$$

and

$$T_2 = \frac{1}{\beta_2}.$$

At each intermediate temperature, the two elementary mappings are described by the two sets of probabilities, $\{\mathbf{v}_i\}$ and $\{\mathbf{p}_\alpha\}$ as depicted in mean field annealing (Peterson & Söderberg, 1989). When the temperature parameters are reduced sufficiently low, the probabilities will approach binary values. We then determine each cortical point at the position of the parameter element, to which the cortical point is mapped. The procedure for the new self-organizing algorithm thus invokes the two sets of mean field equations (10), (11) and (15), (16) in turn during iteration as listed in Appendix A. The method for setting initial cortical points is also suggested in Appendix A.

The above batch-type self-organizing algorithm makes most use of prior knowledge. In eqns (10) and

(15), the distances between parameter elements are coupled with the evolution of the mean configuration $\{\mathbf{v}_i\}$ and $\{\mathbf{p}_\alpha\}$. These two sets of mean field equations can be interpreted as the balance of two kinds of collective forces, which are the attracting forces and the restoration forces. The collective attracting forces are the forces between parameter elements and the cortical points. The collective restoration forces are the forces among the neighboring cortical points with the essential constraints, L_2, (*b*) and (*c*), as for an elastic ring (Durbin & Willshaw, 1987). The activation functions in eqns (10) and (15) encode these essential constraints and serve as a competitive mechanism. When the parameter elements have cluster type distributions, these collective forces are enhanced and will direct the cortical points toward the cluster center. These collective forces eliminate the individual element differences and constitute a kind of global field surrounding the cluster centers. This property leads to the success of our method in good behavior of convergence.

Examining the two sets of mean field equations developed for the two elementary mappings, we see that the computational load of the mean mapping $\{\mathbf{p}_\alpha\}$ is heavier than that of the mean mapping $\{\mathbf{v}_i\}$. By replacing $\langle|\mathbf{x}_i - \mathbf{y}_\alpha|\rangle$ in eqn (10) with $\Sigma_k\langle q_{\alpha k}\rangle |\mathbf{x}_i - \mathbf{x}_k|$, we have

$$u_{i\alpha} = \frac{A}{2} - C\sum_k \langle q_{\alpha k}\rangle |\mathbf{x}_i - \mathbf{x}_k| = \frac{A}{2} - C\sum_k p_{\alpha k}|\mathbf{x}_i - \mathbf{x}_k|. \tag{17}$$

In the above equation, the effective mean field of each variable $v_{i\alpha}$ is not affected by any v variables except itself. However in eqn (15), there are $4N + M^2 + 1p$ variables contributing the effective mean field of each variable $p_{i\alpha}$. To balance the computation of these two mean mappings, we revise the energy function (3) as

$$\begin{aligned} H^*(\sigma, q) = {} & C\sum_i \sum_\alpha \sigma_{i\alpha}|\mathbf{x}_i - \mathbf{y}_\alpha| \\ & + \sum_\alpha \sum_{\gamma\in\mathscr{R}_\alpha} \sum_k \sigma_{k\alpha}|\mathbf{x}_k - \mathbf{y}_\gamma| \\ & - \frac{A}{2}\sum_{i\alpha}\sigma_{i\alpha}^2 - \frac{A}{2}\sum_{\alpha i} q_{\alpha i}^2 \\ & + \frac{B}{2}\sum_i\left(\sum_\alpha q_{\alpha i}\right)^2, \end{aligned} \tag{18}$$

where each distance term $|\mathbf{y}_\alpha - \mathbf{y}_\gamma|$ in the original energy function which represents the restoration force is replaced by the term

$$\sum_k \sigma_{k\alpha}|\mathbf{x}_k - \mathbf{y}_\alpha|$$

and the constant term is neglected. The original term $|y_\alpha - y_\gamma|$ measures the distance between two neighboring cortical points, α and γ. The new term sums up the distance between each parameter element that is represented by the cortical point α and the cortical point γ. This replacement makes the representation of restoration forces consistent with that of attracting forces as in the first term of eqn (18). This consistency becomes obvious when the second term of eqn (18) is rewritten as

$$\sum_k \sum_\alpha \sum_{\gamma\in\mathscr{R}_\alpha} \sigma_{k\alpha}|\mathbf{x}_k - \mathbf{y}_\gamma|.$$

For each pair of cortical point α and parameter element i, the first two terms of the quantity $H^*(\sigma, q)$ is a weighting sum of the distance between them and that between the parameter element i and each neighboring cortical point of the cortical point α, and can be combined as

$$\sum_i \sum_\alpha \sigma_{i\alpha}\left(C|\mathbf{x}_i - \mathbf{y}_\alpha| + \sum_{\gamma\in\mathscr{R}_\alpha}|\mathbf{x}_i - \mathbf{y}_\gamma|\right).$$

The replaced term will smooth the effective individual cortical point and act in a way similar to the restoration forces which smooth the local cortical point with its neighbors. In the same way as for the two sets of mean field eqns (10), (11) and (15), (16), we can obtain another set of mean field equations for the energy function (18):

$$u'_{i\alpha} = -\sum_k \left(Cp_{\alpha k} + \sum_{\gamma\in\mathscr{R}_\alpha} p_{\gamma k}\right)|\mathbf{x}_i - \mathbf{x}_k| + \frac{A}{2}, \tag{19}$$

$$v_{i\alpha} \equiv \langle\sigma_{i\alpha}\rangle = \frac{\exp(\beta_1 u'_{i\alpha})}{\sum_\gamma \exp(\beta_1 u'_{i\gamma})}, \tag{20}$$

$$\begin{aligned} w'_{\alpha i} = {} & \frac{A}{2} - \frac{B}{2} - B\sum_{\gamma\neq\alpha} p_{\gamma i} \\ & - \sum_l \left(Cv_{l\alpha} + \sum_{\gamma\in\mathscr{R}_\alpha} v_{l\gamma}\right)|\mathbf{x}_l - \mathbf{x}_i|, \end{aligned} \tag{21}$$

$$p_{\alpha i} \equiv \langle q_{\alpha i}\rangle = \frac{\exp(\beta_2 w'_{\alpha i})}{\sum_\gamma \exp(\beta_2 w'_{\gamma i})}. \tag{22}$$

Here we briefly introduce the fundamental model used in obtaining the procedure of relaxing two sets of variables. Szu (1989) has developed the hairy neural network model which shows how the dynamics of the two sets of neural variables in the neural network can be characterized by a joint global energy function and their convergence can be guaranteed. To prove the convergence properties, we develop the hairy models for our new algorithms. For the energy function (3), we can obtain a joint global free energy function to characterize the two sets of the mean field equations (10), (11) and (15), (16):

$$\begin{aligned}\phi(u, v, w, p, \beta_1, \beta_2) = \langle H(\sigma, q)\rangle \\ + \sum_i \mathbf{v}_i \mathbf{u}_i + \sum_\alpha \mathbf{p}_\alpha \mathbf{w}_\alpha \\ - \frac{1}{\beta_1} \sum_i \ln z(\mathbf{u}_i, \beta_1) \\ - \frac{1}{\beta_2} \sum_\alpha \ln z(\mathbf{w}_\alpha, \beta_2), \end{aligned} \quad (23)$$

where the form of $\langle H(\sigma, q)\rangle$ is the same as in eqn (13). By taking the partial derivatives of the joint free energy with respect to $v_{i\alpha}$, $u_{i\alpha}$, $p_{\alpha i}$, and $w_{\alpha i}$, we can obtain the mean field equations (10), (11), (15), and (16) correspondingly for the stationary point of the ϕ function. To derive the continuous time mode of evolution of the mean field equations, we set the time ratios of $u_{i\alpha}$ and $w_{\alpha i}$ proportional to the negative of the partial derivative of the free energy (23) with respect to $v_{i\alpha}$ and $p_{\alpha i}$ correspondingly:

$$\frac{du_{i\alpha}}{dt} = -\frac{\partial \phi(u, v, w, p, \beta_1, \beta_2)}{\partial v_{i\alpha}} \quad (24)$$

$$\frac{dw_{i\alpha}}{dt} = -\frac{\partial \phi(u, v, w, p, \beta_1, \beta_2)}{\partial p_{\alpha i}}. \quad (25)$$

The whole set of continuous mean field equations then consist of the eqns (11), (16), (24), and (25). The convergence of the set of continuous mean field equations can be shown for the joint free energy (23). The proof is given in Appendix B. For the on-line self-organizing algorithm developed in the latter context, we can also develop the corresponding continuous mode of evolution for the obtained mean field equations and show the gradient descent property of the corresponding free energy.

The only approximation used in our derivation is the strong independent assumption as in eqn (14), which is used to obtain the form of the mean of an energy function, such as $\langle H(S)\rangle$ in free energy (4). In two places the strong independent assumptions are employed: in eqns (8) and (13). The strong independent assumption is also the basic assumption of the mean field annealing. The derivation of a hairy neural network uses no assumption. The derivation of eqns (24) and (25) from the joint free energy is exactly based on the gradient descent method. However, to obtain a stationary point of a hairy neural network, eqns (24), (25), (11), and (16), we use a simulation algorithm as in Appendix A. In the simulation algorithm, two Potts models are relaxed separately in turn. Such an execution saves computational effort, but actually makes an independent assumption between two Potts models. This is a problem involving implementation not theory. This is reasonable since our computational tool is a digital computer.

2.2. Relation to the Durbin Algorithm

To obtain Durbin's self-organizing algorithm, the distance terms in the energy function (3) are first changed to the terms of distance squared. The changed energy function is denoted by H^c. The absolute distance $|\cdot|$ will provide a median type of smoothing effect among the neighbors, and the distance squared will smooth neighbors in terms of mean. In the same way as for the derivation of eqns (10) and (11), we can then derive a set of mean field equations for the variables $\{\boldsymbol{\sigma}_i\}$ in the energy function H^c. Let H^c_y denote the energy function when each $\sigma_{\alpha i}$ is fixed at its mean value similar to the energy function H_q (12). When we simply relax the energy function H^c_y by using the gradient descent method instead of using mean field annealing, we obtain the dynamic equation for the cortical points proposed by Durbin. That is, we set the change of each cortical point $\mathbf{y}_\alpha$ as follows.

$$\begin{aligned}\Delta \mathbf{y}_\alpha &\equiv -\frac{\partial H^c_y}{\partial \mathbf{y}_\alpha} \\ &= 2C \sum_i \langle \sigma_{\alpha i}\rangle(\mathbf{x}_i - \mathbf{y}_\alpha) + 2 \sum_{\gamma \in \mathcal{B}_\alpha} (\mathbf{y}_\gamma - \mathbf{y}_\alpha)\end{aligned} \quad (26)$$

where $\langle\sigma\rangle$ can be obtained from eqn (11) with a minor modification, replacing $\langle|\mathbf{x}_i - \mathbf{y}_\alpha|\rangle$ with $|\mathbf{x}_i - \mathbf{y}_\alpha|^2$. The version in eqn (26) is exactly the same as the gradient version of the elastic ring proposed by Durbin and Willshaw (1987). The derivation of eqns (10) and (11) illustrates that Durbin and Willshaw's elastic net algorithm makes the same mean field approximations as in a Potts model. From the dynamics of the Durbin algorithm, we categorize the Durbin algorithm as a batch-type self-organizing algorithm. However the Durbin algorithm uses only weak constraints and makes no use of the distribution information among parameter elements. In the above derivation for the Durbin algorithm, the partial derivative of H^c_y with respect to $\mathbf{y}_\alpha$ eliminates all of the q variables, which carry with them the critical constraints. Thus the competitive sharing-out of the cortical points among the parameter space is not included in the Durbin algorithm. The key points of the new algorithm are to use the mean field annealing for determining the motion of the cortical points gradually and to add essential constraints for coherent mapping from the cortex to the parameter space. The mean field annealing is able to avoid becoming trapped in any of the tremendous number of bad local minima within the energy function. Hence the new algorithm can much improve the Durbin algorithm in quality.

2.3. A New On-Line Self-Organizing Algorithm

This type of algorithm can process one input at a time since no explicit information between any two parameter elements is used in the algorithm. This property restricts the development of the new on-line algorithm toward a Potts model with two sets of coupling dynamics. Unlike the batch-type algorithm, an on-line algorithm can not resolve the mean position of a cortical pont into a linear combination of parameter elements, since at each time instance the system knows only one parameter element and memorizes nothing. It is impossible to replace the mean position of a cortical point with a Potts neural variable. Following this argument, we reserve the dynamics of the cortical positions in the last subsection for constructing a new on-line self-organizing algorithm. This construction may result in incompletion of Potts models with two sets of dynamics as in Subsection 2.1, but it stresses that the function of the Kohonen algorithms can be accomplished by using Potts models, which possess solid theoretical foundation.

The Kohonen algorithm is an on-line self-organizing algorithm. It uses a dynamical neighborhood structure to maintain the smooth distribution of the cortical points and a winner-take-all principle to implement the competition among cortical sites for each parameter element. In this section, we will provide a new on-line self-organizing algorithm on the basis of using only one Potts neuron. We start with Ritter's version (Ritter & Schulten, 1988) of the Kohonen algorithm. For each parameter element $\mathbf{x}_i$, the change in the cortical point $\mathbf{y}_\alpha$ is:

$$\Delta y_\alpha = \eta \Lambda(\alpha, \alpha^*)(\mathbf{x}_i - y_\alpha), \tag{27}$$

where α^* denotes the winner site, of which the cortical point is closest to the input element:

$$|\mathbf{y}_{\alpha^*} - \mathbf{x}_i| \leqslant |\mathbf{y}_\alpha - \mathbf{x}_i|. \tag{28}$$

The neighborhood function $\Lambda(\alpha, \alpha^*)$ is 1 for $\alpha = \alpha^*$ and falls off with distance $r_{\alpha\alpha^*}$, which is the distance between sites α and α^* on the cortex surface lattice. A typical choice for $\Lambda(\alpha, \alpha^*)$ is:

$$\Lambda(\alpha, \alpha^*) = \exp\left(-\frac{r^2_{\alpha\alpha^*}}{d^2}\right), \tag{29}$$

where d is a tunable width parameter that is gradually decreased during the training.

In the above Kohenon algorithm, the movement of the cortical point $\mathbf{y}_\alpha$ depends on the neighborhood function $\Lambda(\alpha, \alpha^*)$, of which the time varying size and distribution are heuristically predetermined and the center is chosen as the nearest cortical point to the input element. In the following formulation, we propose an altenative design for the neighborhood function. We aim at obtaining an optimal neighborhood function for each parameter element under the critical objective and the constraints.

Given a parameter element $\mathbf{x}_i$, we need to determine the membership function of the input to all cortical points. We use a set of binary neural variables ξ_α, $1 \leqslant \alpha \leqslant M^2$, to serve as the membership function. Each ξ_α plays the same role as each $\Lambda(\alpha, \alpha^*)$ in eqn (27). The objectives that need to be optimized include the weighted distance between the input and the cortical map and the smoothness of the membership function. The only constraint is the unitary condition for all neural variables. All terms can be quantified. Our formulation now is:

$$\text{minimizing } L_1 = \sum_\alpha \xi_\alpha |\mathbf{x}_i - \mathbf{y}_\alpha|^2,$$

and

$$L_2 = \sum_\alpha \sum_{\gamma \in \mathcal{B}_\alpha} (\xi_\alpha - \xi_\gamma)^2$$

subject to

$$\sum_\alpha \xi_\alpha = 1,$$
$$\xi_\alpha \in \{0, 1\}. \tag{30}$$

The first objective L_1 in the above formulation favors the activation of the neural variable for which the corresponding cortical point is closest to the input. We interconnect M^2 neural variables as a lattice structure. In the second objective, $\mathcal{B}_\alpha$ denotes the set of the four neighbors of the αth neural variable; L_2 is used to measure the smoothness of the analog membership function at each intermediate state toward the binary solution. The optimal binary solution of the membership function is easy to find for the formulation (30). By the unitary condition, we know only one neural variable is on and the others are off. Then the L_2 term is a constant with value 4. The L_1 term can be simply minimized by activating the neural variable, for which the corresponding cortical point is closest to the input $\mathbf{x}_i$. The optimal binary solution of the formulation (30) indeed follows the winner-take-all principle. To naturally emulate the dynamics of the neighborhood function used in eqn (27), we need to calculate the analog membership function at each intermediate state toward binary solution. For this purpose, we will first translate the formulation (30) into a Hopfield type energy function and then use mean field annealing to find the mean configuration.

To include the Kohonen algorithm into a Potts neural network, we consider the following energy function:

$$\begin{aligned} H^K &= \frac{C_K}{2} \sum_\alpha \xi_\alpha |\mathbf{x}_i - \mathbf{y}_\alpha|^2 \\ &\quad + \sum_\alpha \sum_{\gamma \in \mathscr{B}_\alpha} (\xi_\alpha - \xi_\gamma)^2 + A_K \sum_\alpha \sum_{\gamma \neq \alpha} \xi_\alpha \xi_\gamma \\ &= \frac{C_K}{2} \sum_\alpha \xi_\alpha |\mathbf{x}_i - \mathbf{y}_\alpha|^2 \\ &\quad - 2 \sum_\alpha \sum_{\gamma \in \mathscr{B}_\alpha} \xi_\alpha \xi_\gamma \\ &\quad - (A_K - 8) \sum_\alpha \xi_\alpha^2 + A_K, \end{aligned} \tag{31}$$

where C_K and A_K are the Lagrange multipliers. The constant is negligible. The term with coefficient A_K in the first line of eqn (31) is equivalent to the unitary condition. All terms are rearranged and further simplified by applying the unitary condition. By mean field annealing, we can derive the free energy and the mean field equations for the energy function H^K:

$$\psi^K = \langle H^K \rangle + \sum_\alpha m_\alpha \langle \xi_\alpha \rangle - \frac{1}{\beta_K} \sum_\alpha \ln \sum_\alpha \exp(\beta_K m_\alpha), \tag{32}$$

and

$$m_\alpha = -\frac{C_K}{2} |\mathbf{y}_\alpha - \mathbf{x}_i|^2 - 2 \sum_{\gamma \in \mathscr{B}_\alpha} \langle \xi_\gamma \rangle - (A_K - 8), \tag{33}$$

$$\langle \xi_\alpha \rangle = \frac{\exp(\beta_K m_\alpha)}{\sum_\gamma \exp(\beta_K m_\gamma)}, \tag{34}$$

where m_α is an additional auxiliary variable and β_K is a control temperature parameter. The mean of the energy (31) can be approximated by replacing each neural variable with its mean.

The mean field equations (33) and (34) characterize the behavior of the Potts neuron. This Potts neuron performs an idealized dynamical winner-take-all function as mentioned by Kohonen (1993). The critical temperature of this Potts neuron can be analytically obtained by the method in the work of Peterson and Söderberg (1989). Above the critical temperature, each neural variable responds equally.

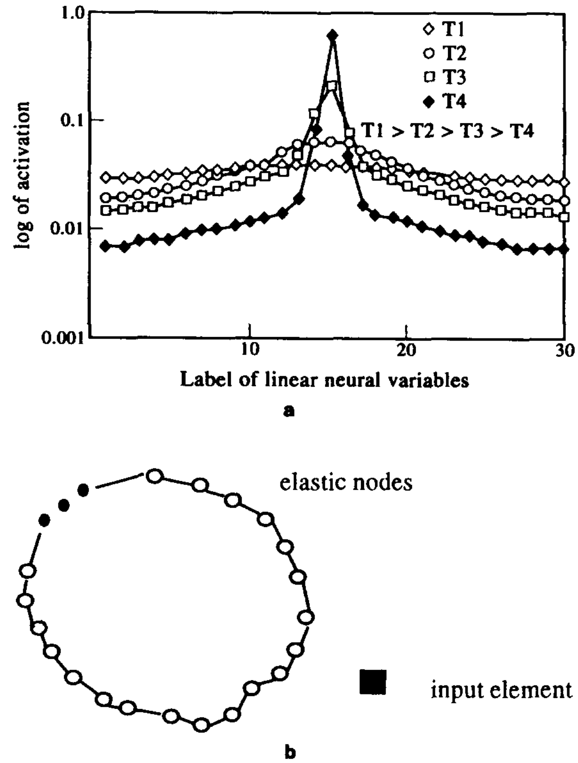


FIGURE 1. (a) Clustering phenomenon of a Potts neuron in one dimension. (b) Positions of nodes and input point used in Figure a. Thirty nodes are arranged as a small circle.

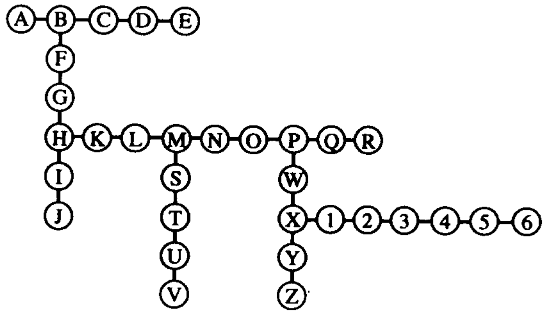
At sufficiently low temperature, among the M^2 neural variables, only one variable is one and the others are zero. At each intermediate temperature, the responses of neural variables show a clustering phenomenon as a result of the excitatory interaction and the normalized activation function, which represents the inhibitary interactions among neural variables. Figure 1a shows the function of this Potts neuron in the one-dimensional case. In this case, the positions of nodes and input element in a plane are arranged as in Figure 1b. The Potts neuron can be naturally used to emulate the neighboring function (29) in the parallel and distributed process. The interconnections within the Potts neuron is indeed simple and regular such that the hardware implementation is easily achievable. The function of the Potts neuron describes the responses of all cortical points to the input. The responses are then used to tune the cortical points toward the input $\mathbf{x}_i$. By setting the change of each cortical point negatively proportional to the partial derivative of the free energy ψ_i^K with respect to y_α, we have:

$$\begin{aligned} \Delta y_\alpha &\equiv -\frac{\partial \psi_i^K}{\partial y_\alpha} \\ &= \eta \langle \xi_\alpha \rangle (\mathbf{x}_i - \mathbf{y}_\alpha). \end{aligned} \tag{35}$$

The updating rule is similar to the Kohonen algorithm, except the neighborhood function is replaced by $\langle \xi_\alpha \rangle$.

3. SIMULATIONS AND CONCLUSIONS

We test the new algorithm described by (19)–(22) using two examples. The first one is the taxonomy example, for which the data are the same as in Kohonen (1988). We duplicate each element by adding small random noise to form the parameter space, which then contains 64 parameter elements. In the following simulations, we use the mean field equations (19)–(22) with constants $A = 0.2$ and $B = 0.4$, and an 8×8 lattice. The four edges of the neuron lattice are connected as in Subsection 2.1. The annealing process is controlled by two temperatures T_1 and T_2; T_1 is initially set larger than T_2 to synchronize the convergence of the two mappings. The taxonomy example demonstrates the topology preserving property of the new algorithm, which has potential for hierarchical classification. The results are shown in Figure 2. Figure 2a shows the hierarchical structure of the original data. We see that the embedded hierarchical structure within the data set is captured by the algorithm in Figure 2b as squeezed into the lattice space. Figure 2b is obtained by labeling each data symbol on its mapped site. The



a

*	*	A	T	*	3	4	5
*	B	*	*	U	*	2	*
E	D	C	*	V	Z	1	*
*	*	F	*	*	Y	X	*
*	H	G	*	*	*	W	*
I	*	K	*	O	P	Q	*
*	J	L	M	N	*	R	*
*	*	*	S	*	*	*	6

b

FIGURE 2. (a) Hierarchical presentation of the data in the taxonomy example. (b) Self-organization map of the data in (a), which is obtained by labeling each data on its mapped site on the lattice.

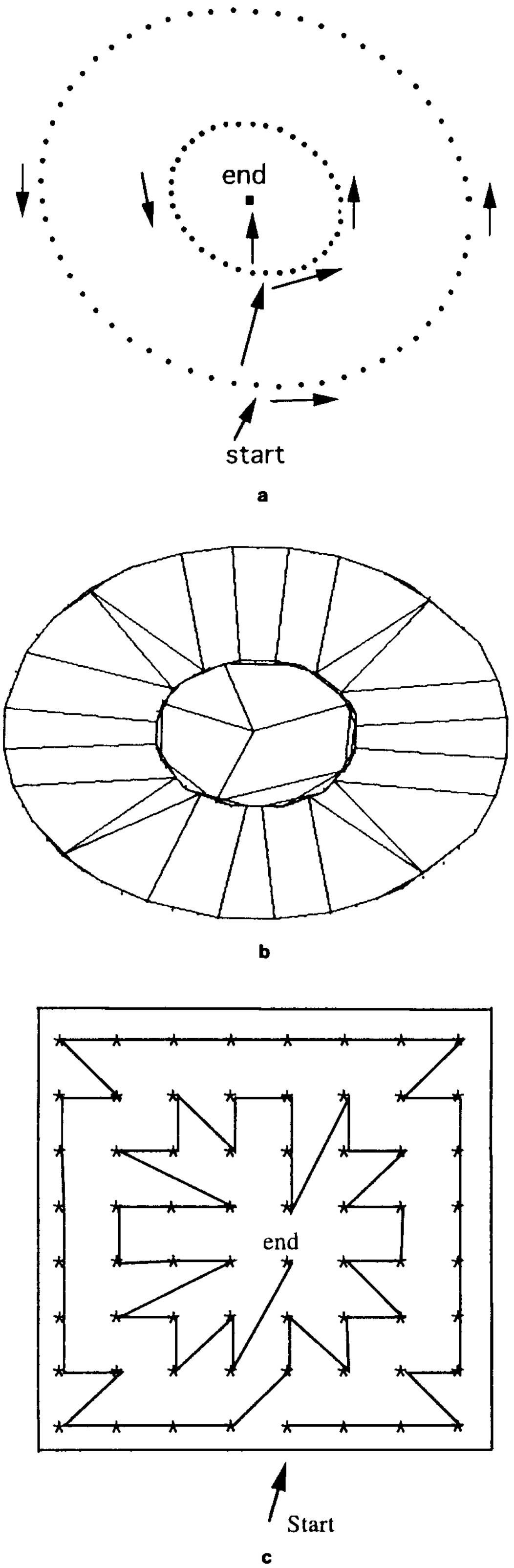


FIGURE 3. (a) The distribution of the data in the second example. The arrows indicate a tracing path starting from the outer circle to the inner point. (b) The trained cortical points on the lattice are drawn on the parameter space. The two neighboring cortical points are connected by a line in the figure. (c) A corresponding path in the self-organization map captures the geometrical feature of the tracing path in Figure 2a.

TABLE 1
The Resulting Measurement Quantities, L_1 and L_2, of the three Self-Organizing Algorithms for two Examples

		HAPER Alg.	Durbin Alg.	Kohonen Alg.
Example 1	L_1	0	1.587	0.546
	L_2	25.986	29.342	27.386
	$L_1 + L_2$	25.986	30.929	27.932
Example 2	L_1	1.745	2.995	3.653
	L_2	17.756	23.488	24.115
	$L_1 + L_2$	19.501	26.483	27.768

second example is a two-dimensional case. The four edges of the lattice are not connected in this example. The 100 parameter elements are arranged as in Figure 3a. As a result, the cortical points with lines to its neighbors are shown in parameter space as in Figure 3b. If we trace all elements by following the arrows as in Figure 3a, we see a corresponding path in the cortex as shown in Figure 3c. The tracing path in the cortex explicitly displays the geometrical feature of the path in the parameter space. These two examples experimentally show the success of self-organization by the new self-organization algorithm. When compared with the Durbin algorithm and the Kohonen algorithm, the new self-organizing algorithm produces a well-qualified self-organization map. The resulting measurement quantities, L_1 and L_2, from the new self-organizing algorithm are shorter than those from the other two algorithms as in Table 1. The measurement quantities of the new algorithm and the Durbin algorithm for the second example along simulation epochs are shown in Figures 4 and 5 respectively.

Now we examine application to the formation of ocular dominance stripes (Goodhill & Willshaw, 1990). This application mainly employs the self-organizing algorithm as the mechanism for projecting the cells from the two retinae onto the cortex and explain the development of ocular dominance stripes in the vertebrate visual system. We follow the simulation model of Goodhill and Willshaw (1990) to test the new self-organizing algorithm. Two cases are studied: a simplified case of two one-dimensional retinae mapping to a one-dimensional region of the cortex, and the general case of a two-dimensional retinae mapping to a two-dimensional region of the cortex. For the one-dimensional case, the simulation model can be described as a traveling salesmen problem, of which the cities are regularly arranged as two parallel rows within a unit square, the rows running in the horizontal direction and separated by a certain specified distance. The position along the horizontal axis then represents the position within one retina and the vertical separation of two rows indicates ocularity. Then a one-dimensional elastic ring representing the cortex is used to classify the city set. The elastic ring we use has a break in it like that used by Goodhill and Willshaw. Each node on the elastic ring has two neighbors except for the two end points, which have only one neighbor. The receptive field of a node contains two real components for representing the node position. By appropriately adjusting the neighboring structure, the new algorithm described by eqns (19)–(22) can be used. In our simulations, 40 nodes and 40 cities are used. When the elastic ring is initially set as a small circle, the city set is well classified into two disjoint clusters and no stripe occurs. When the initial positions of the elastic nodes are set as in Figure 6, the resulting patterns form stripes. Figure 7 shows the patterns generated for different vertical separations. All these patterns are obtained by drawing each node at the position specified by its receptive field and connecting any two neighboring nodes. When the distance d between two cells is not less than the vertical separation l, the

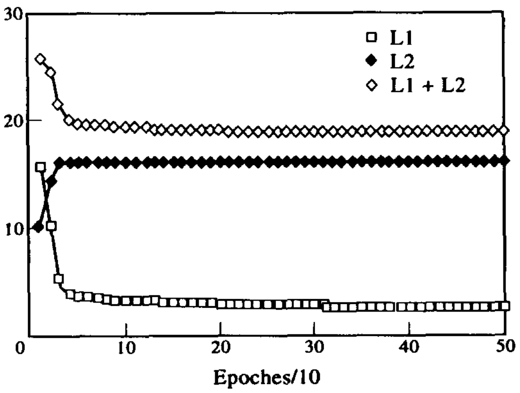


FIGURE 4. Measurement quantities of the new self-organizing algorithm for the second example. Lower curve: L_1. Middle curve: L_2. Upper curve $L_1 + L_2$.

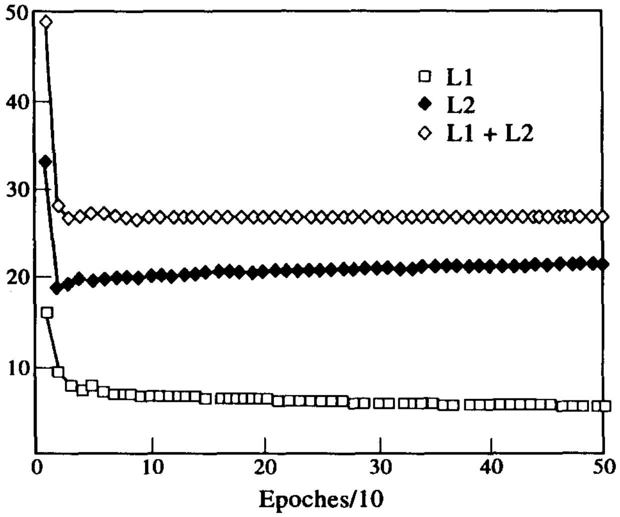


FIGURE 5. Measurement quantities of the Durbin algorithm for the second example. Lower curve: L_1. Middle curve: L_2. Upper curve: $L_1 + L_2$.

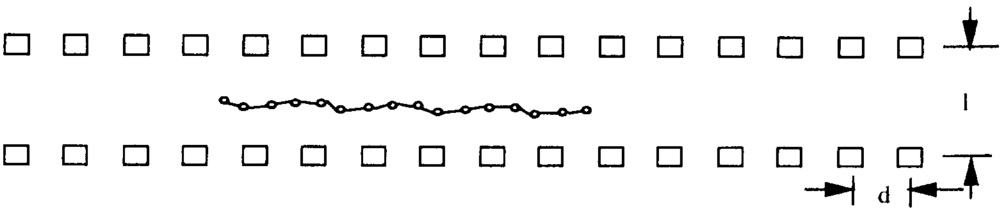


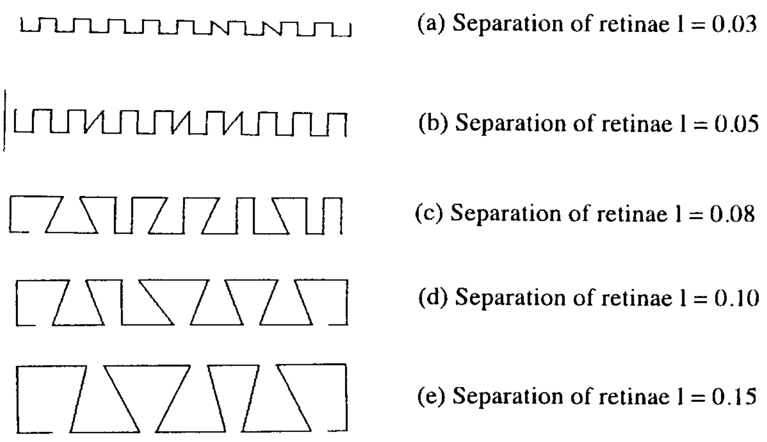
FIGURE 6. One-dimensional simulation model for ocular dominance stripes. The elastic ring is linearly initialized along the middle of two layers. *d* denotes the distance between two cells. *l* denotes the ocularity.

resulting stripes have minimal size, such as the patterns in Figures 7a and 7b. As the ratio d/l gets smaller, the size of the predicted stripes gets larger as shown in Figures 7c–7e.

Obviously, the formation of stripes is caused by deeply local minima of the energy function. From the viewpoint of optimization, although the mean field annealing has improved the gradient descent method in performance, yet it is unable to obtain global minima in complicated situations. This is for two reasons, one the mean field approximation made during derivation and the other the numerical simulations in the digital computer. For the former reason, further theoretical effort can be made. The independent assumption between two Potts neurons in deriving a Potts model can be released by using a new version of the free energy the same as that for deriving the TAP equations (Thouless et al., 1977) in spin glasses. Potts models using TAP equations (Wu, 1994) have been shown to essentially compensate for the illegality at high temperature and overcome the initial problem in the mean field annealing neural networks.

The 2D case is a straightforward generalization of the 1D case. The two retinae are represented as two planes of cells, lying on top of one another and separated by a small gap. The cortex is represented by an elastic net arranged as a 25 × 25 lattice. The resulting patterns are shown in Figure 8.

In this work, the self-organizing algorithms are derived thoroughly from the viewpoint of optimization. With proper optimization of the objective and constraints, the task of dimensional reduction with the property of topology preservation is incorporated into energy functions as in the Hopfield models and the Potts neural networks. The optimizing quantities we use are viewed as significant clues for understanding principles behind the formation of cortical maps in physiology (Durbin & Mitchison, 1990). Our formulation further exploits these quantities for self-organization. We are able to add the direct objective to the model. In minimizing the developed energy functions, we use the technology of mean field annealing and then obtain Potts neural networks as the computational model. The Potts neural networks are the general modes of the Hopfield neural networks and are for the first time extended to the task of self-organization. The Potts neural networks have a high degree of parallelism. This is helpful for applying the new self-organizing algorithm to many real time applications. We also show that the Potts neural networks obtained belong to the category of hairy models proposed by Szu (1989) which we use to show their convergence properties. The convergence of the new batch-type self-organizing algorithm can be qualified by measuring the optimizing quantities of the final mapping table. This makes the neural networks realizable.



(a) Separation of retinae l = 0.03

(b) Separation of retinae l = 0.05

(c) Separation of retinae l = 0.08

(d) Separation of retinae l = 0.10

(e) Separation of retinae l = 0.15

FIGURE 7. The generated patterns for different vertical separations in one-dimensional case. *d* = 0.05 for all patterns.

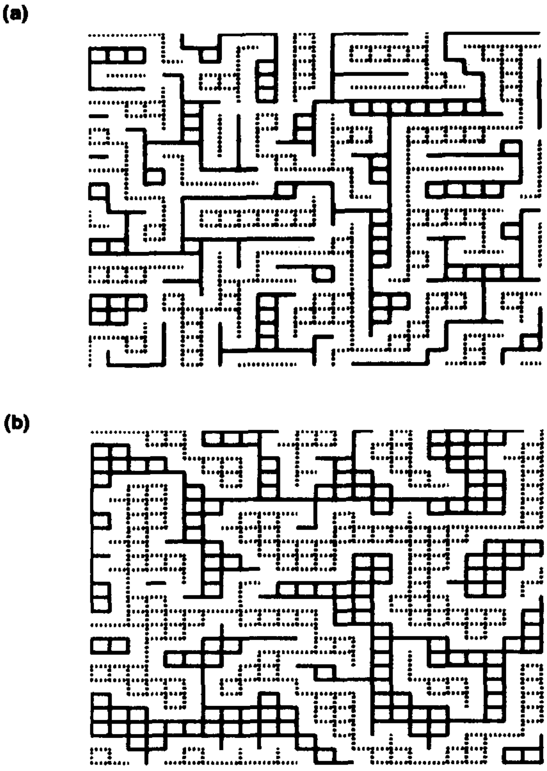


FIGURE 8. (a) Separation of retinae 1 = 0.1. (b) Separation of retinae 1 = 0.4.

The new algorithm provides high reliability and efficiency for self-organization. For the batch-type self-organizing algorithm, we have enhanced the design of the Durbin algorithm by applying mean field annealing to increase reliability, by including complete constraints to make the most use of the prior knowledge of the parameter space, and by introducing collective attractive and collective restoration forces to smooth the resulting cortex surface to facilitate convergence. These collective forces can form a rough field surrounding the cluster centers. These collective forces near a cluster can eliminate the individual force differences of each element in the cluster and enhance the force toward the cluster center. This collective behavior results in well-behaved convergence in our method when the distributions of the parameter element are clustered in several places. For the on-line self-organizing algorithm, we have provided a new set of differential equations to model the idea of the Kohonen algorithm. This novel method can serve as the foundation for exploring the physiological implications of this popular self-organizing algorithm. The hardware implementation of the serial-type self-organizing algorithm can be realized by a regularly interconnected Potts neuron. Using this new technique, temporal relational data can be processed in real time.

REFERENCES

Aiyer, S. V. B., Niranjan, M., & Fallside F. (1990). A theoretical investigation into the performance of the Hopfield model. *IEEE Trans. Neural Netorks*, **1(2)**, 204–215.

Angeniol, B., de la Crox Vaubois, G., & le Texier, J. Y. (1988). Self-organizing feature maps and the traveling salesman problem. *Neural Networks*, **1**, 189–293.

Conwey, A. Q. (1979). *J. Exp. Psychol.*, **31**, 1–17.

Dayan, P. (1993) Arbitrary elastic topologies and ocular dominance. *Neural Computation*, **5**, 392–401.

Durbin, R., & Willshaw, G. (1987). An analogue approach to the traveling salesman problem using an elastic net method. *Nature*, **326**(16), April.

Durbin, R., & Mitchison, G. (1990). A dimension reduction framework for understanding cortical maps. *Nature*, **343**(15), Feb.

Goodhill, G. J., & Willshaw, D. J. (1990). Application of the elastic net algorithm to the formation of ocular dominance stripes. *Network*, **1**, 41–59.

Hopfield, J. J., & Tank, D. W. (1985). Neural computation of decisions in optimization problems. *Biol. Cybernet.*, **52**, 141.

Kohonen, T. (1988). *Self-organization and associative memory*. Berlin: Springer-Verlag.

Kohonen, T. (1993). Physiological interpretation of the self-organizing map algorithm. *Neural Networks*, **6**, 895–905.

Kohonen, T., Makisara, K., & Saramaki, T. (1982). Phonotopic maps—insightful representation of phonological features for speech recognition. *Proc. Seventh Int. Conf. Pattern Recognition*, Montreal Canada (pp. 182–185).

Liou, C. Y., & Wu, J. M. (1992). Designing energy surface for truck backer-upper problem. IJCNN Beijing, China (pp. II102–II109).

Miller, K. D., Keller, J. B., & Stryker, M. P. (1989). Ocular dominance column development: analysis and simulation. *Science*, **245**, 605–615.

Peterson, C., & Söderberg, B. (1989). A new method for mapping optimization problems onto neural network. *Int. J. Neural Syst.* **1**, 3.

Ritter, H., & Schulten, K. (1988). Convergence properties of Kohonen's topology conserving maps: fluctuation, stability, and dimension selection. *Biol. Cybernet.*, **60**, 59–71.

Szu, H. H. (1989). Reconfigurable neural nets by energy convergence learning principle based on extended McCulloch–Pitts neurons and synapses. IJCNN-89-Washington, **1**, 485–496.

Tanaka, S. (1990). Theory of self organization of cortical maps: mathematical framework. *Neural Networks*, **3**, 625–640.

Thouless, D. J., Anderson, P. W., & Palmer, R. G. (1977). Solution of solvable model of a spin glass. *Phil. Mag.*, **35**(3), 593–601.

Wu, J. M. (1994). Hairy Potts neural networks for elastic rings. Ph.D. thesis, National Taiwan University Press.

Yuille, A. L. (1990). Generalized deformable model, statistical physics, and matching problems. *Neural Computation*, **2**, 1–24.

Yuille, A. L., Kolodny, J., & Lee, C. W. (1991). Dimension reduction, generalized deformable models, and the development of occularity and orientation. Harvard Robotics Laboratory Technical Report 91-3.

APPENDIX A

The procedure for the new batch-type self-organizing algorithm is:

1. Set initial T_1 and T_2, and initial cortical points.
2. Relax eqns (10) and (11).
3. Relax eqns (15) and (16).
4. Reduce T_1 and T_2 by a scalar 0.98 and if terminating criterion is satisfied, end the algorithm, otherwise go to step 3.

One exception in the above algorithm is notable. When the mean field equations (10) and (11) are invoked for the first time, the term $\langle |\mathbf{x}_i - \mathbf{y}_{\nu(\alpha)}| \rangle$ in eqn (10) is substituted by the term $|\mathbf{x}_i - \langle \mathbf{y}_{\nu(\alpha)} \rangle|$, where $\langle \mathbf{y}_\alpha \rangle$ is the initial cortical point. In the latter relaxation, this mean term is substituted by $\Sigma_k P_{\alpha k} |\mathbf{x}_i - \mathbf{x}_k|$.

Two methods are suggested for setting the initial cortical points. One is by regular assignment. We first find the mass center of the set of the parameter elements. Around this mass center, we have a K-dimensional hyper-sphere. We shift the origin of the parameter space to the center of mass. Assume the radius of the hyper-sphere is a small value r. The hyper-sphere contains an inner hyper-cube, which has 2^K vertices. The positions of the 2^K vertices are

$$\{(a_1, \ldots, a_K | a_i) = \pm \frac{1}{\sqrt{K}}, 1 \leqslant i \leqslant K\}.$$

We recursively partition the lattice R into 2^K regions, denoted by $R = \{R_{b_1 \cdots b_K} | b_i = 0 \text{ or } 1, 1 \leqslant i \leqslant K\}$ and assign the coordinate of each vertex to the cortical point of the center of the corresponding region. At each partition step, a subregion R_b is equally partitioned into left-up, right-up, left-down, right-down subregions, denoted by R_{b00}, R_{b01}, R_{b10}, R_{b11} respectively. Then the cortical point of the center of each region $R_{b_1 \cdots b_K}$ is assigned to the coordinate

$$\left(\frac{c_1 r}{\sqrt{K}}, \ldots, \frac{c_K r}{\sqrt{K}}\right)$$

with each $c_i = 2b_i - 1$, $1 \leqslant i \leqslant K$. If K is even and 2^{K-1} divides into the side length M of the lattice integer times, all primitive regions have equal size. If not, a slight modification can partition the lattice as equally as possible. We assign the undetermined cortical points to zeros and then determine them on the basis of the determined cortical points. We synchronously update cortical points by the equation

$$\mathbf{y}_\alpha^{t+1} = \frac{r(\mathbf{y}_\alpha^t + \sum_{\gamma \in \mathcal{B}_\alpha} \mathbf{y}_\gamma^t)}{\left|\mathbf{y}_\alpha + \sum_{\gamma \in \mathcal{B}_\alpha} \mathbf{y}_\gamma^t\right|} \tag{36}$$

until all points are stable. The above updating rule smooths the initial distribution of cortical points. This procedure has also been used to calculate the artificial flow fields for autonomous navigation within complex environments (Liou & Wu, 1992).

The other method is to first randomly set initial cortical points and then use the serial-type self-organization algorithm to organize the distribution of the cortical points for several steps

APPENDIX B

$d\psi/dt \leqslant 0$ is proved as follows:

$$\begin{aligned} \frac{d\psi}{dt} &= \sum_i \left(\frac{\partial \psi}{\partial \mathbf{v}_i}\right)^t \frac{d\mathbf{v}_i}{dt} + \sum_\alpha \left(\frac{\partial \psi}{\partial \mathbf{p}_\alpha}\right)^t \frac{d\mathbf{p}_\alpha}{dt} \\ &= -\frac{1}{\beta_1} \sum_i \left(\frac{d\mathbf{u}_i}{dt}\right)^t \left(M_1 \frac{d\mathbf{u}_i}{dt}\right) \\ &\quad - \frac{1}{\beta_2} \sum_\alpha \left(\frac{d\mathbf{w}_\alpha}{dt}\right)^t \left(M_2 \frac{d\mathbf{w}_\alpha}{dt}\right) \leqslant 0, \end{aligned} \tag{37}$$

where M_1 is the Hessian of $\ln z(\mathbf{u}_i, \beta_1)$, and M_2 is that of $\ln z(\mathbf{w}_\alpha, \beta_2)$.

$$M_1 = \frac{\sum_{[\boldsymbol{\sigma}_i]} \exp(\beta_1 \mathbf{v}_i^t \boldsymbol{\sigma}_i)[\boldsymbol{\sigma}_i - \mathbf{v}_i][\boldsymbol{\sigma}_i - \mathbf{v}_i]^t}{\sum_{[\boldsymbol{\sigma}_i]} \exp(\beta_1 \mathbf{v}_i^t \boldsymbol{\sigma}_i)}$$

$$M_2 = \frac{\sum_{[\mathbf{q}_\alpha]} \exp(\beta_2 \mathbf{p}_\alpha^t \mathbf{q}_\alpha)[\mathbf{q}_\alpha - \mathbf{p}_\alpha][\mathbf{q}_\alpha - \mathbf{p}_\alpha]^t}{\sum_{[\mathbf{q}_\alpha]} \exp(\beta_2 \mathbf{p}_\alpha^t \mathbf{q}_\alpha)},$$

$[\boldsymbol{\sigma}_i]$ and $[\mathbf{q}_\alpha]$ run over $\{\mathbf{e}_1, \ldots, \mathbf{e}_N\}$. Since both M_1 and M_2 are positive definite,

$$\sum_i \left(\frac{d\mathbf{u}_i}{dt}\right)^t M_1 \frac{d\mathbf{u}_i}{dt} > 0,$$

and

$$\sum_\alpha \left(\frac{d\mathbf{w}_\alpha}{dt}\right)^t M_2 \frac{d\mathbf{w}_\alpha}{dt} > 0.$$

$$\frac{d\psi}{dt} \leqslant 0$$

is shown.