Yet Another Algorithm which can Generate Topography Map

John Sum, Chi-sing Leung, Lai-wan Chan, and Lei Xu

***Abstract*— This paper presents an algorithm to form a topographic map resembling to the self-organizing map. The idea stems on defining an energy function which reveals the local correlation between neighboring neurons. The larger the value of the energy function, the higher the correlation of the neighborhood neurons. On this account, the proposed algorithm is defined as the gradient ascent of this energy function. Simulations on two-dimensional maps are illustrated.**

***Index Terms*— Kohonen net, neural network, self-organizing map.**

I. Introduction

In recent decades, many researchers have attempted to mimic the mammalian sensory ordered map, in particular the topography map. Goodhill [4], Kohonen [5], and Willshaw and van der Malsburg [12] have proposed self-organizing algorithms based on the idea of competition. Durbin and Willshaw [2] on the other hand considered the map formation as a process similar to that of elastic string. Linsker proposed a multilayered probabilistic network structure and defined the learning algorithm based on the idea of maximum mutual information. In [11] van Velzen applied the idea of Ising spin to define a Hamiltonian. By minimizing this Hamiltonian, he derived an algorithm and network model, Ising spin network, to generate a topographic map. Recently, the idea of minimum description length (MDL) has also been applied to define such an algorithm [13]. Some of these algorithms can be described as gradient algorithms because they are embedded with well-defined objective functions which not only ease the formulation of the convergence proof but also suggest possible physical interpretations for the algorithms [3].

This paper will be organized into six sections. In the next section, the algorithm of self-organizing map (SOM) will be briefly reviewed and its mechanism is reinterpreted. In accordance with this new interpretation, we suggest the idea of maximum neighborhood coupling. Using this idea, the proposed algorithm is elucidated in Section III. In Section IV, a simple simulated example is presented to demonstrate the topographical property. Section V presents the connection between the proposed algorithm and two other algorithms. Finally, the conclusion is presented in Section VI.

II. Preliminary Review of Self-Organizing Map

SOM was proposed by Kohonen in the early 1980's [5] and was based on the idea of competition and neighborhood update concepts. As its mechanism is simple, it has been widely applied to solve many engineering problems [1], [6], [7], [14].

Manuscript received January 31, 1996; revised April 30, 1996 and October 25, 1996.

The authors are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N. T., Hong Kong.

Publisher Item Identifier S 1045-9227(97)05245-4.

For clarity, we assume that the SOM consists of N neurons. The input to the SOM is denoted by $x \in R^n$ and the weight vectors are denoted by $\mu_1, \mu_2, \cdots, \mu_N \in R^n$. The neurons are arranged in lattice form. The neighborhood relationship amongst these neurons is defined by the connection matrix $G = (g_{ij})$, where $g_{ij} = 1$ in G means that the ith and jth neurons are neighbor. In Kohonen's definition [6], the set consisting of all the neighborhood neurons of the ith neuron is denoted by N_i and is called the ith neighborhood set.

The mechanism of SOM is defined as follows:

$$\mu_i(t+1) = \begin{cases} \mu_i(t) + \alpha(t)[x - \mu_i(t)] & \text{if } i \in N_c, \\ \mu_i(t) & \text{otherwise} \end{cases} \tag{1}$$

where c is the index of the winning neuron. The vector x is the input pattern presented at step t and $\alpha(t)$ is the step size satisfying the conditions for stochastic approximation (see [8, Theorem 4.3.1]). The winning neuron c is defined as the one whose weight vector is the closest to the input x, i.e.,

$$c = \arg\min_k \{\|x - \mu_k\|\}.$$

Two points should be noted from the step (1). First, the winner and its neighbors are now closer to the input pattern than before. Second, they move closer to each other.

Although the map is able to search for the location of the set of data and self-organize to an ordered map by repeating (1), it has been proven that there is no energy function reflecting the mechanism of SOM [16]. Thus SOM does not have a physical meaning such as maximum entropy or maximum likelihood, etc., except in the special case that G is an identity matrix. In such case, SOM learning reduces to K-means clustering which objective function is an approximated maximum likelihood function ([15, ch. 6]).

III. From New Interpretation to the New Algorithm

Considering the points noted in the above section, it is possible to interpret the topological map formation based on an idea similar to correlation. Imagine that each of the weight vectors is the center of a set of data. Assume that the set is of fixed size. Once an ordered map is formed, each set is overlapped to its neighbor sets and the total area of the overlapping between neighbor sets should be large.

Suppose that the ith neuron will fire if the input pattern belongs to its corresponding set, then the maximum overlapping between two neighboring neurons is equivalent to the maximum correlation between their outputs. Therefore the algorithm should also be interpreted as one which tries to maximize the correlation between the outputs of the neighboring neurons.

With this new interpretation, an alternative algorithm can be defined. It is accomplished by a probabilistic neural network. This network consists of n input neurons and N output

neurons, Fig. 1. Each of the output neuron is fully connected to the neurons in the input layer via a weight vector, denoted by μ_i. The response of each of the output neuron is binary, {0, 1}, governed by the Gaussian probability

$$p\{s_i = 1|\hat{x}\} = \frac{1}{\sqrt{(2\pi\tau)}} \exp\left(-\frac{\|\hat{x} - \mu_i\|^2}{\tau}\right)$$
$$p\{s_i = 0|\hat{x}\} = 1 - \frac{1}{\sqrt{(2\pi\tau)}} \exp\left(-\frac{\|\hat{x} - \mu_i\|^2}{\tau}\right) \quad (2)$$

where $\tau > 0$. We denote this probability by $p_i(x)$. Note that it is a conditional probability dependent on input x. Assume that once neighboring neurons output one, a coupling energy, ξ, will be generated. This coupling energy reflects the correlation between the output of the ith and jth neurons, the larger the average coupling energy, the larger the average correlation between the neighboring neurons. As the ith and jth output neurons response independently, the ensemble average of the coupling energy between the ith and jth neuron is given as follows:

$$\begin{aligned}\langle \xi_{ij} \rangle_x &= \langle s_i s_j \xi \rangle_x \\ &= \xi \langle s_i \rangle_x \langle s_j \rangle_x \\ &= \xi p_i(x) p_j(x) \end{aligned} \quad (3)$$

where $\langle z \rangle_x$ means the expectation of z given an input x. For simplicity, let g_{ij} be a value which indicates the neighborhood relation between output neurons

$$g_{ij} = \begin{cases} 1 & \text{if } i\text{th and } j\text{th neuron are neighbor} \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Then the coupling energy (3) can be rewritten as follows:

$$\langle \xi_{ij} \rangle_x = \xi g_{ij} p_i(x) p_j(x) \quad (5)$$

where $i, j = 1, 2, \cdots, N$. Summing $\langle \xi_{ij} \rangle_x$ over i, j, we can obtain the average network coupling energy

$$E_x = \xi \sum_{i=1}^{N} \sum_{j=1}^{N} g_{ij} p_i(x) p_j(x). \quad (6)$$

Inspired by the algorithm of maximum likelihood competitive learning (MLCL) [10], we have

$$\begin{aligned} C &= \sum_x \log E_x \\ &= M \log \xi + \sum_{k=1}^{M} \log \left[\sum_{i=1}^{N} \sum_{j=1}^{N} g_{ij} p_i(x_k) p_j(x_k) \right]. \end{aligned} \quad (7)$$

As our objective is to maximize the correlation between neighboring neurons, the network parameters, i.e., μ_is, should maximize the following cost function:

$$E = \sum_{k=1}^{M} \log \left[\sum_{i=1}^{N} \sum_{j=1}^{N} g_{ij} p_i(x_k) p_j(x_k) \right] \quad (8)$$

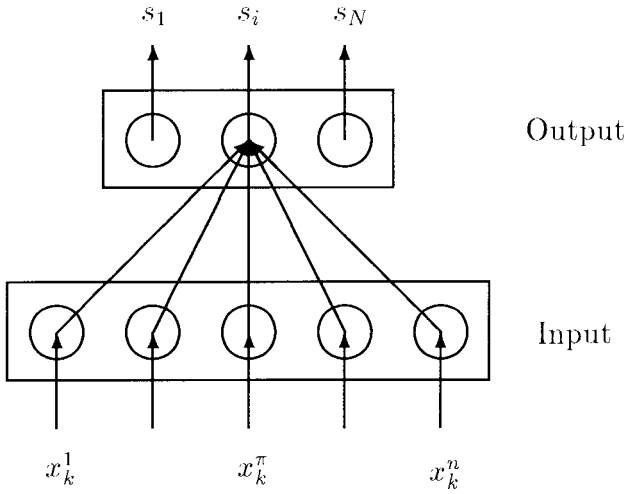

Fig. 1. The network structure of the proposed network model.

where $\{x_1, x_2, \cdots, x_M\}$ is the input data set.

Since p_i is a Gaussian function with mean μ_i and variance τ, taking the gradient of the cost function (8), the algorithm can be defined as follows:

$$\mu_i(t+1) = \mu_i(t) + \alpha(t) \sum_{j=1}^{N} w_{ij}[x - \mu_i(t)] \quad (9)$$

$$w_{ij} = \frac{g_{ij} \exp\left(-\frac{\|x - \mu_i\|^2}{\tau}\right) \exp\left(-\frac{\|x - \mu_j\|^2}{\tau}\right)}{\sum_{l,m} g_{lm} \exp\left(-\frac{\|x - \mu_l\|^2}{\tau}\right) \exp\left(-\frac{\|x - \mu_m\|^2}{\tau}\right)}. \quad (10)$$

Notice that this algorithm cannot be further reduced to SOM. While $\tau \to 0$, (9) and (10) reduce to simple competitive learning (CL).

IV. Simulation

Fig. 2 shows a simulated example using (9) and (10). The map is of size 15 × 15, each of the weight vectors is two dimensional (2-D). Initially, all the weight vectors are randomly distributed inside the unit square. Then during each step of update, an input data x is selected randomly from the unit square. In this simulation, τ is decreasing with time

$$T(t) = \beta_1 - \beta_2 \left\lfloor \frac{t}{10^6} \right\rfloor$$

for all $0 \le t < 8 \times 10^6$. The parameters β_1 and β_2 are set to be 0.16 and 0.02, respectively. $\lfloor a \rfloor$ corresponds to the largest integer below a. It is found that the map stretches to larger size as τ decreases, see Fig. 2(d)–(i) and note that the size of the map increases from $[0.3, 0.7] \times [0.3, 0.7]$ to $[0.1, 0.9] \times [0.1, 0.9]$. Simulations for the maps with size 10 × 10 and 20 × 20 are carried out under the conditions. All of them can form ordered maps as well.

Remark 1: It should be noted that while the value of τ is too small, no ordered map can be formed as the algorithm will reduce to simple competitive learning while $\tau \to 0$.

Remark 2: In case the value of τ is too large, say $\tau \gg 1$, $\|x - \mu_i\|^2 \to 0$. Then all the exponential terms in (10) will

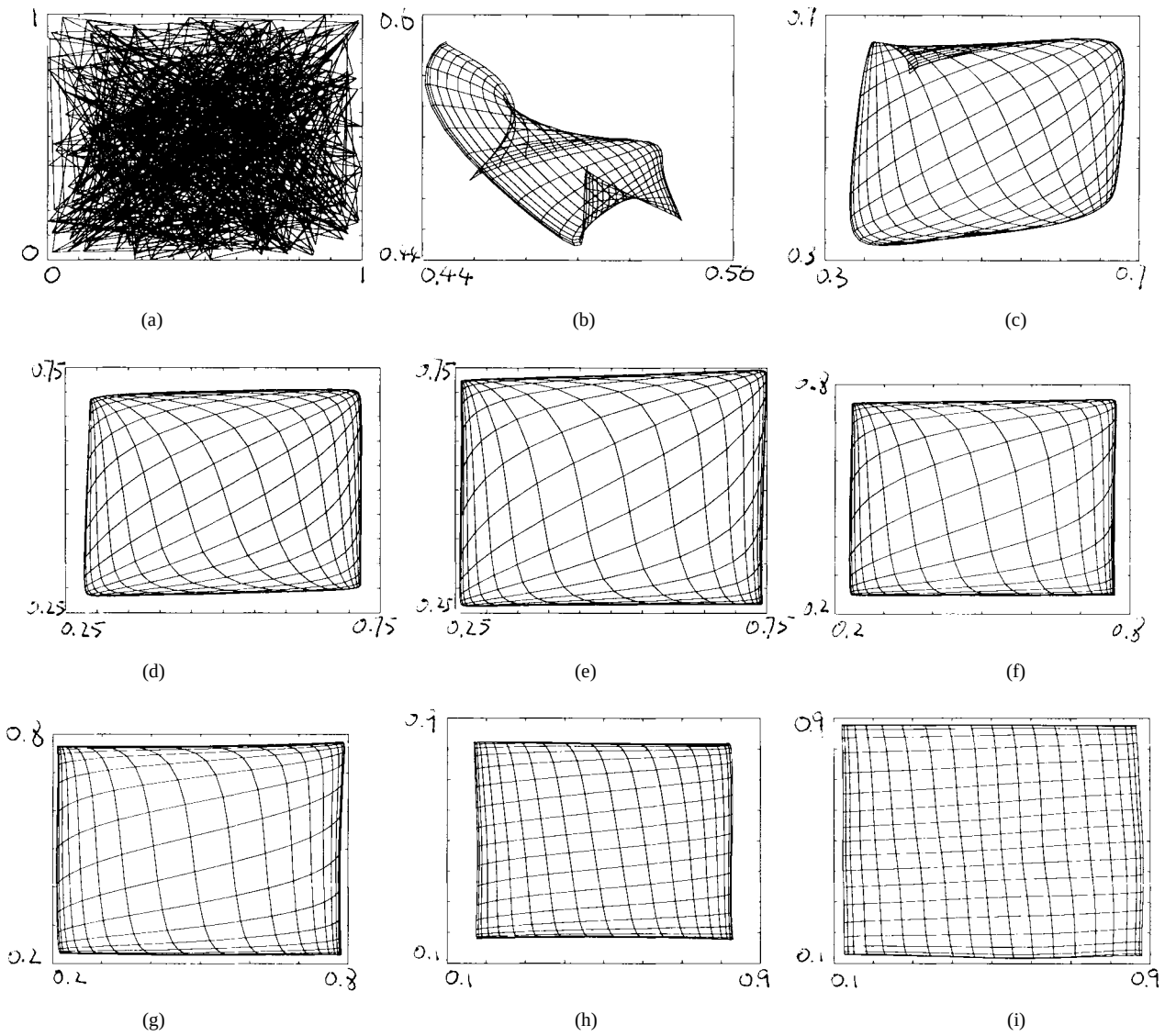

Fig. 2 The topographic map formation in the simulation of 15 × 15 map at the time when (a) $t = 0$, (b) $t = 10^6$, (c) $t = 2 \times 10^6$, (d) $t = 3 \times 10^6$, (e) $t = 4 \times 10^6$, (f) $t = 5 \times 10^6$, (g) $t = 6 \times 10^6$, (h) $t = 7 \times 10^6$, and (i) $t = 8 \times 10^6$.

be close to one and all the values of w_{ij} will be equal. As a results, all the weight vectors μ_i will converge to the sample mean, i.e., the center of the unit square.

Remark 3: Similarly, once the weight vectors are initialized very closed to the center of the unit square, no map can be formed in general unless the value τ is turned to very small. In accordance with our simulation results on one-dimensional (1-D) and 2-D maps, the recommended range of the τ value is [0.15, 0.25].

V. Connection with Other Algorithms

Suppose that the weight vectors and the input patterns are confined to have unit length, i.e., $\|\mu_i\| = 1$ and $\|x_k\| = 1$ for all $i = 1, \cdots, N$ and $k = 1, \cdots, M$, the probability function can be written as follows:

$$p\{s_i = 1 \text{ given input } \hat{x}\} = \frac{1}{\sqrt{(2\pi\tau)}} \exp\left(-\frac{2}{\tau} + \frac{2\hat{x}'\mu_i}{\tau}\right) \tag{11}$$

where $\hat{x}\prime$ is the transpose of $\hat{x}$. Hence, the cost function (8) and (10) can be expressed as follows:

$$E = -\frac{4M}{\tau} + \sum_{k=1}^{M} \log \cdot \left[\sum_{i=1}^{N}\sum_{j=1}^{N} g_{ij} \exp\left(2\frac{\hat{x}_k'\mu_i}{\tau}\right) \exp\left(2\frac{\hat{x}_k'\mu_j}{\tau}\right)\right] \tag{12}$$

$$w_{ij} = \frac{g_{ij} \exp\left(2\frac{\hat{x}'\mu_i}{\tau}\right) \exp\left(2\frac{\hat{x}'\mu_j}{\tau}\right)}{\sum_{l,m} g_{lm} \exp\left(2\frac{\hat{x}'\mu_l}{\tau}\right) \exp\left(2\frac{\hat{x}'\mu_m}{\tau}\right)}. \tag{13}$$

This is just the van Velzen algorithm [11]. Therefore, our proposed algorithm can be treated as a generalization of his model.

TABLE I
SUMMARY OF THE SIMILARITIES AND DIFFERENCES BETWEEN SOM AND THE PROPOSED ALGORITHM

Properties	SOM	Our algorithm
Existence of energy function	×	√
Formation of order map	√	√
Reduced to CL	√	√
Reduced to van Velzen algorithm	×	√
Reduced to MLCL	×	√

Furthermore, suppose that the matrix G is defined as an identity matrix. The algorithm (9) and (10) reduce to

$$\mu_i(t+1) = \mu_i(t) + \alpha(t) w_{ii}[x - \mu_i(t)] \tag{14}$$

and

$$w_{ii} = \frac{\exp\left(-\frac{\|x-\mu_i\|^2}{\tau}\right)\exp\left(-\frac{\|x-\mu_i\|^2}{\tau}\right)}{\sum_{l,l}\exp\left(-\frac{\|x-\mu_l\|^2}{\tau}\right)\exp\left(-\frac{\|x-\mu_l\|^2}{\tau}\right)} \tag{15}$$

$$= \frac{\exp\left(-\frac{2\|x-\mu_i\|^2}{\tau}\right)}{\sum_{l}\exp\left(-\frac{2\|x-\mu_l\|^2}{\tau}\right)}. \tag{16}$$

The cost function (8) reduces to

$$E = \sum_{k=1}^{M} \log\left[\sum_{i=1}^{N}\exp\left(-\frac{2\|x_k-\mu_i\|^2}{\tau}\right)\right] \tag{17}$$

where $\{x_1, x_2, \cdots, x_M\}$ is the input data set. This is exactly the algorithm of MLCL.

VI. CONCLUSION

In summary, this paper has presented a new interpretation of the mechanism of SOM. Consider that SOM is an algorithm maximizing the correlation of the outputs of the neighboring neurons, a cost function expressing this objective is defined and hence an alternative algorithm is deduced. Furthermore, we have shown that the proposed algorithm can be reduced to van Velzen algorithm [11] and maximum likelihood competitive learning [10]. For clarity, the similarities and differences between SOM and the proposed algorithm are summarized in Table I. As the algorithm proposed by van Velzen is based on the idea of Ising spinning, it is suspected that our algorithm can also be interpreted in the same manner. Recently, many researchers are looking for a meaningful interpretation for the mechanism of SOM, this paper may shed light on this direction.

REFERENCES

[1] B. Angeniol *et al.*, "Self-organizing feature maps and the traveling salesman problem," *Neural Networks*, vol. 1, pp. 289–293, 1988.
[2] R. Durbin and D. Willshaw, "An analogue approach to the travelling salesman problem using an elastic net method," *Nature*, vol. 326, 644–647, 1987.
[3] R. Durbin and G. Mitchison, "A dimension reduction framework for understanding cortical maps," *Nature*, vol. 343, pp. 644–647, Feb. 1990.
[4] G. J. Goodhill, "Topography and ocular dominance: A model exploring positive correlations," *Biol. Cybern.*, vol. 69, pp. 109–118, 1993.
[5] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.*, vol. 43, pp. 59–69, 1982.
[6] ———, *Self Organizing Map.* Berlin: Springer-Verlag, 1995.
[7] M. A. Kraaijveld *et al.*, "A nonlinear projection method based on Kohonen's topology preserving maps," *IEEE Trans. Neural Networks*, vol. 6, pp. 548–559, 1995.
[8] H. K. Kushner and D. S. Clark, *Stochastic Approximation for Constrained and Unconstrained Systems.* Berlin: Springer-Verlag, 1978.
[9] C. van der Malsburg, "Development of ocularity domains and growth behavior of axon terminals," *Biol. Cybern.*, vol. 32, pp. 49–62, 1979.
[10] S. Nowlan, "Maximum likelihood competitive learning," *Advance in Neural Information Processing Systems*, pp. 574–582, 1990.
[11] G. A. van Velzen, "Topological maps on Ising spin networks," *Physica A*, vol. 185, pp. 439–443, 1992.
[12] D. Willshaw and C. von der Malsburg, "A marker induction for the establishment of order neural mapping: Its application to the retinotectal problem," *Philosophical Trans. Roy. Soc. London*, vol. 287, no. B1021, pp. 203–243, 1979.
[13] R. S. Zemel and G. Hinton, "Learning population codes by minimizing description length," *Neural Computa.*, vol. 7, pp. 549–564, 1995.
[14] C. S. Leung and L. W. Chan, "Kohonen map: Transmission of vector quantization data under noisy channel," *IEEE Trans. Neural Networks*, accepted for publication.
[15] R. Duda and P. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.
[16] E. Erwin *et al.*, "Self-organizing maps: Ordering, convergence properties, and energy functions," *Biol. Cybern.*, vol. 67, pp. 47–55, 1992.