Diffusion Approximation of Frequency Sensitive Competitive Learning

Aristides S. Galanopoulos, *Member, IEEE,* Randolph L. Moses, *Senior Member, IEEE,*
and Stanley C. Ahalt, *Member, IEEE*

***Abstract*— The focus of this paper is a convergence study of the frequency sensitive competitive learning (FSCL) algorithm. We approximate the final phase of FSCL learning by a diffusion process described by a Fokker–Plank equation. Sufficient and necessary conditions are presented for the convergence of the diffusion process to a local equilibrium. The analysis parallels that by Ritter and Schulten for Kohonen's self-organizing map (SOM). We show that the convergence conditions involve only the learning rate and that they are the same as the conditions for weak convergence described previously. Our analysis thus broadens the class of algorithms that have been shown to have these types of convergence characteristics.**

***Index Terms*— Diffusion processes, Fokker–Plank equations, neural networks, learning systems, vector quantization.**

I. Introduction

THE frequency sensitive competitive learning (FSCL) is a conscience type competitive learning algorithm developed by Ahalt *et al.* [1] to overcome problems associated with the simple competitive learning (CL) and Kohonen's self-organizing feature maps (SOM's) in vector quantization applications.

The FSCL algorithm is a modification of simple CL in which units are penalized in proportion to some function of the frequency of winning, so that eventually all units participate in the quantization of the data space as representative vectors of a data cell of nonzero probability. This frequency-sensitive conscience mechanism overcomes the codeword underutilization problem of simple CL [4], [13]. The SOM algorithm [6], [7], also successfully solves the above problem, however, it appears less suitable for some vector quantization applications for the following reasons. First, since SOM was developed to establish feature maps, an essential feature of the SOM algorithm is the definition of a topology on the set of codewords, which is a difficult problem for high-dimensional data spaces. Second, the required update of several codewords at each iteration makes the algorithm more computationally intensive than CL and FSCL.

One of the main issues in vector quantizer designs is the ability of the algorithms to generate a codebook that is as near-optimal as possible in a reasonable amount of time. Simple CL is a stochastic gradient descent algorithm; like the batch LBG algorithm [9] it may become stuck in a local minimum of the total distortion function, although it has some ability to escape local minima because of its stochastic nature. The strongest type of convergence for CL has been shown for the case of sufficiently sparse patterns in [4], otherwise, sufficient and necessary conditions for convergence with probability one to a local minimum are the corresponding conditions of the Robbins–Monro process [12]. More relaxed conditions are required for weak convergence to a local minimum [8].

The convergence of the SOM algorithm has been analyzed by Ritter and Schulten [11] under the assumption that the state of the algorithm is close to some local equilibrium and in the limit of small learning rate. Ritter and Schulten analyzed a Fokker–Plank equation (FPE) describing SOM's and found necessary and sufficient conditions that make the mean and variance of the state deviation from equilibrium vanish. Their conditions correspond to the weak convergence conditions presented in [8]. Cottrell and Fort [2] analyzed a similar process, although restricted to a uniform input data probability density function and data spaces of dimension one and two, formulated as a Robbins–Monro recursion—thus arriving at the conditions in [12].

In this paper, we study the final phase of the FSCL algorithm learning. We assume that both the time index and the learning rate are small and we follow the same analysis as described in [11] deriving the FPE that approximates the evolution of the process. From the analysis point of view, this work should be considered as an extension of Ritter and Schulten's work [11] to a different class of learning algorithms. The results show that in the limit of large time, i.e., for the final learning phase, only conditions on the learning rate must be imposed to guarantee the convergence of the diffusion process to a local equilibrium. This result supports the use of the FSCL algorithm in applications, e.g., video and speech encoding, in which accurate representation, computation, and underutilization all must be managed simultaneously in an on-line coding process.

A global convergence analysis, not restricted to the neighborhood of some local equilibrium, would certainly involve all the algorithm parameters and not just the learning rate. We are unaware of any global convergence analysis for self organizing maps and conscience type algorithms. The only

Manuscript received June 1, 1994; revised September 16, 1996 and April 2, 1997.

A. S. Galanopoulos was with the Department of Electrical Engineering, The Ohio State University, Columbus, OH 43210 USA. He is now with Northern Telecom, Richardson, TX 75082 USA.

R. L. Moses and S. C. Ahalt are with the Department of Electrical Engineering, The Ohio State University, Columbus, OH 43210 USA.

Publisher Item Identifier S 1045-9227(97)05237-5.

1045–9227/97$10.00 © 1997 IEEE

algorithm that has been proved to converge to the global minimum of the distortion function is simulated annealing (SA) [5], [10], and heuristic combinations of SA with previously mentioned algorithms have been used to cope with the very large computational requirements of SA [14].

II. FSCL Algorithm

In FSCL, a network of N units or codewords is trained by a set of input data vectors. The data vectors $\mathbf{v}$ belong to a d-dimensional space and their distribution is described by a probability density function $P(\mathbf{v})$. The units have associated positions $\{\mathbf{w}_i(t)\}_{i=1}^N$ in the d-dimensional space and associated update frequencies $\{f_i(t)\}_{i=1}^N$, where the update frequencies are defined as $f_i = c_i(t)/t$ with the (count) $c_i(t)$ being the number of times that unit i has been updated up to time t. Clearly, $\sum_i f_i(t) = 1$. The time index t takes on integer values.

At time t, an input data vector $\mathbf{v}(t)$ is presented to the network and a winning unit is selected as the one that minimizes the product of a *fairness function*, F, which is an increasing function of the update frequency, times the distance (distortion measure) from the input data vector, as

$$\text{winner } s = \operatorname{ArgMin}_i\{F[f_i(t)]\|\mathbf{w}_i(t) - \mathbf{v}(t)\|\}. \qquad (1)$$

The winning unit's position is updated as

$$\mathbf{w}_s(t+1) = \mathbf{w}_s(t) + \epsilon(t)[\mathbf{v}(t) - \mathbf{w}_s(t)] \qquad (2)$$

where $\epsilon(t)$ is the learning rate, and the winning unit count $c_s(t)$ is incremented by one. All other units keep the same counts and positions.

III. Diffusion Approximation

The FSCL network can be described as a Markov process with state $\sigma(t)$ defined as

$$\sigma(t) = \begin{bmatrix} \mathbf{w}_1(t) & \cdots & \mathbf{w}_N(t) \\ f_1(t) & \cdots & f_N(t) \end{bmatrix}.$$

The analysis of FSCL convergence parallels the analysis of the SOM algorithm by Ritter and Schulten in [11]. We consider an ensemble of networks whose states at time t are distributed according to a density function $\tilde{S}(\sigma, t)$ defined over the set of all possible network states. This density function obeys the Chapman–Kolmogoroff equation

$$\tilde{S}(\sigma, t+1) = \int d^{N(d+1)}\sigma' Q(\sigma, \sigma', t)\tilde{S}(\sigma', t)$$

where $Q(\sigma, \sigma', t)$ is the transition probability from state σ' to σ expressed as

$$\begin{aligned} Q(\sigma, \sigma', t) &= \int d\mathbf{v}\delta[\sigma - \mathbf{T}(\sigma', \mathbf{v}, t)]P(\mathbf{v}) \\ &= \sum_{s=1}^{N} \int_{A_s^1(\sigma)} d\mathbf{v}\delta[\sigma - \mathbf{T}(\sigma', \mathbf{v}, t)]P(\mathbf{v}) \end{aligned}$$

where we denote by $A_s^1(\sigma)$ the set of possible previous inputs given previous winner s and present state σ. The transformation $\mathbf{T}(\sigma', \mathbf{v}, t)$ denotes the state at time $t+1$ given the state σ' and input $\mathbf{v}$ at time t. Thus

$$\begin{aligned} \tilde{S}(\sigma, t+1) = \sum_{s=1}^{N} \int_{A_s^1(\sigma)} d\mathbf{v} \int d^{N(d+1)}\sigma' \\ \cdot \delta[\sigma - \mathbf{T}(\sigma', \mathbf{v}, t)]P(\mathbf{v})\tilde{S}(\sigma', t). \end{aligned}$$

Every term in the above sum is conditional upon the winner at time t. Given the winner s at time t, the inverse transformation $\mathbf{T}_s^{-1}(\sigma, \mathbf{v}, t)$ uniquely defines the state at time t from the state $\sigma(t+1)$ and the input $\mathbf{v}(t)$; every unit r is transformed as

$$[\mathbf{T}_s^{-1}(\sigma, \mathbf{v}, t)]_r = \begin{bmatrix} \mathbf{w}_r(t) + \dfrac{\epsilon(t)}{1 - \epsilon(t)}[\mathbf{w}_r(t) - \mathbf{v}]\,\delta_{rs} \\ \dfrac{t+1}{t} f_r(t) - \dfrac{1}{t}\delta_{rs} \end{bmatrix}.$$

To express the right-hand side in terms of σ, instead of σ', we integrate with respect to $\mathbf{T}$

$$\begin{aligned} \tilde{S}(\sigma, t+1) &= \sum_{s=1}^{N} \int_{A_s^1(\sigma)} d\mathbf{v} \int \frac{d^{N(d+1)}\mathbf{T}}{\left|\dfrac{\partial \mathbf{T}}{\partial \sigma'}\right|} \\ &\quad \cdot \delta[\sigma - \mathbf{T}(\sigma', \mathbf{v}, t)]P(\mathbf{v})\tilde{S}[\mathbf{T}_s^{-1}(\sigma, \mathbf{v}, t), t] \\ &= \sum_{s=1}^{N} \int_{A_s^1(\sigma)} d\mathbf{v} \frac{1}{\left|\dfrac{\partial \mathbf{T}}{\partial \sigma'}\right|} P(\mathbf{v})\tilde{S}[\mathbf{T}_s^{-1}(\sigma, \mathbf{v}, t), t]. \end{aligned}$$

The matrix $\partial\mathbf{T}/\partial\sigma'$ is of dimensions $N(d+1) \times N(d+1)$. Assuming s is the winner, the update rule for the state of unit r is

$$\mathbf{T}_r = \begin{bmatrix} \mathbf{w}_r(t) + \epsilon(t)[\mathbf{v}(t) - \mathbf{w}_r(t)]\,\delta_{sr} \\ f_r(t)\,\dfrac{t}{t+1} + \dfrac{1}{t+1}\delta_{sr} \end{bmatrix}.$$

The Jacobian $|\partial\mathbf{T}/\partial\sigma'|$ is independent of the winner

$$\begin{aligned} J(\epsilon) &\triangleq \left(\left|\frac{\partial \mathbf{T}}{\partial \sigma'}\right|\right)^{-1} \\ &= \left(\frac{t+1}{t}\right)^N [1 - \epsilon(t)]^{-d} \end{aligned}$$

and

$$\tilde{S}(\sigma, t+1) = J(\epsilon) \sum_{s=1}^{N} \int_{A_s^1(\sigma)} d\mathbf{v}P(\mathbf{v})\tilde{S}[\mathbf{T}_s^{-1}(\sigma, \mathbf{v}, t), t].$$

In the above expression only the volume $A_s^1(\sigma)$ depends on the winner selection rule and the fairness function in particular. Our goal is to derive a linear FPE that approximates the evolution of the process. We expand $J(\epsilon)$ and $\tilde{S}(\mathbf{T}_s^{-1}, t)$ keeping only derivatives up to second order (FPE neglects higher order derivatives) and of these only the leading order in ϵ

$$\begin{aligned}
&\tilde{S}(\sigma, t+1) \\
&\quad = (1+\epsilon d) \sum_{s=1}^{N} \int_{A_s^1(\sigma)} d\mathbf{v} P(\mathbf{v}) \tilde{S}(\sigma, t) \\
&\qquad + \sum_{s=1}^{N} \int_{A_s^1(\sigma)} d\mathbf{v} P(\mathbf{v}) \sum_{m=1}^{d} \frac{\epsilon}{1-\epsilon} \\
&\qquad \cdot (\mathbf{w}_{sm} - \mathbf{v}_m) \frac{\partial \tilde{S}(\sigma, t)}{\partial \mathbf{w}_{sm}} \\
&\qquad + (1+\epsilon d) \sum_{s=1}^{N} \int_{A_s^1(\sigma)} d\mathbf{v} P(\mathbf{v}) \sum_{r=1}^{N} \left(-\frac{1}{t}\right) \\
&\qquad \cdot [\delta_{rs} - f_r(t+1)] \frac{\partial \tilde{S}(\sigma, t)}{\partial f_r} \\
&\qquad + \epsilon^2 \sum_{s=1}^{N} \int_{A_s^1(\sigma)} d\mathbf{v} P(\mathbf{v}) \tfrac{1}{2} \sum_{m=1}^{d} \sum_{n=1}^{d} \\
&\qquad \cdot \frac{1}{(1-\epsilon)^2} (\mathbf{w}_{sm} - \mathbf{v}_m)(\mathbf{w}_{sn} - \mathbf{v}_n) \frac{\partial^2 \tilde{S}(\sigma, t)}{\partial \mathbf{w}_{sm} \partial \mathbf{w}_{sn}} \\
&\qquad - \frac{\epsilon}{2t} \sum_{s=1}^{N} \int_{A_s^1(\sigma)} d\mathbf{v} P(\mathbf{v}) \sum_{m=1}^{d} \sum_{r=1}^{N} \\
&\qquad \cdot \frac{1}{1-\epsilon} (\mathbf{w}_{sm} - \mathbf{v}_m)[\delta_{rs} - f_r(t+1)] \frac{\partial^2 \tilde{S}(\sigma, t)}{\partial \mathbf{w}_{sm} \partial f_r} \\
&\qquad + \frac{1+\epsilon d}{2t^2} \sum_{s=1}^{N} \int_{A_s^1(\sigma)} d\mathbf{v} P(\mathbf{v}) \sum_{r=1}^{N} \sum_{r'=1}^{N} \\
&\qquad \cdot [\delta_{rs} - f_r(t+1)][\delta_{r's} - f_{r'}(t+1)] \frac{\partial^2 \tilde{S}(\sigma, t)}{\partial f_r \partial f_{r'}} \qquad (3)
\end{aligned}$$

where $J(\epsilon) \approx (1+\epsilon d)$ for large t.

Let $A_s^0(\sigma)$ be the set of inputs that make s the winning unit when the network is at state σ. The sets $A_s^0(\sigma)$ cover the whole space without overlapping. In contrast, the sets $A_s^1(\sigma)$ may, in general, overlap, i.e., given present state σ, there are inputs $\mathbf{v}$ such that two (or more) possible previous states σ_i and σ_j exist for which $\mathbf{T}(\sigma_i, \mathbf{v}, t) = \sigma$ and $\mathbf{T}(\sigma_j, \mathbf{v}, t) = \sigma$. Furthermore, the sets $A_s^1(\sigma)$ do not necessarily cover the whole space. As a result of the fairness function weighting, the regions $A_s^0(\sigma)$ and $A_s^1(\sigma)$ are not separated by hyperplanes but by higher order surfaces, even if the Euclidean distance is used as the distortion measure. Assuming a fairness function of the form $F(f) = f^\beta$, where β is a positive parameter, we could write the following approximation for the case of one-dimensional data, i.e., $d = 1$

$$\text{volume } [A_s^1(\sigma)] \approx \left[1 + \frac{\alpha_s(t)}{2}\right] \cdot \text{volume } [A_s^0(\sigma)]$$

where

$$\begin{aligned}
1 + \alpha_s(t) &= [1 - \epsilon(t)] \left[1 + \frac{1}{t f_s(t)}\right]^\beta \\
&\approx 1 - \epsilon(t) + \frac{\beta}{t f_s(t)}.
\end{aligned}$$

In d-dimensions we would roughly approximate

$$\begin{aligned}
&\text{volume } [A_s^1(\sigma)] \\
&\quad \approx \left[1 + \frac{\alpha_s(t)}{2}\right]^d \cdot \text{volume } [A_s^0(\sigma)] \\
&\quad \approx \left[1 - \frac{\epsilon(t) d}{2} + \frac{\beta d}{2 t f_s(t)}\right] \cdot \text{volume } [A_s^0(\sigma)].
\end{aligned}$$

The important conclusion is that the difference between the volumes of $A_s^1(\sigma)$ and $A_s^0(\sigma)$ is of the order of the learning rate $\epsilon(t)$ and of $1/t$, and it can be neglected in our expansion where we keep only the leading-order terms. Consequently, the following analysis is valid for *any* fairness function that preserves this close relation between $A_s^1(\sigma)$ and $A_s^0(\sigma)$. The difference between the first moments of $A_s^1(\sigma)$ and $A_s^0(\sigma)$ would be even smaller (if the two volumes have similar shapes) while the difference between the second moments is slightly larger (they would roughly relate through a factor of $[1 + \alpha_s(t)/2]^{d+2}$ instead of $[1 + \alpha_s(t)/2]^d$).

Assume that the system is close to the equilibrium state $\overline{\sigma}$. We express (3) in terms of the deviation y from $\overline{\sigma}$

$$\begin{aligned}
y &= \sigma - \overline{\sigma} \\
&= \begin{bmatrix} \mathbf{u}_1(t) & \cdots & \mathbf{u}_N(t) \\ x_1(t) & \cdots & x_N(t) \end{bmatrix}.
\end{aligned}$$

Let $S(y, t) \triangleq \tilde{S}(\sigma, t)$. Keeping only the leading terms in ϵ and $1/t$, we obtain the FPE

$$\begin{aligned}
\partial_t S(y, t) &= \epsilon \sum_{sm} \overline{f}_s \frac{\partial}{\partial \mathbf{u}_{sm}} [\mathbf{u}_{sm} S(y, t)] \\
&\quad + \frac{1}{t} \sum_{s} \frac{\partial}{\partial x_s} [x_s S(y, t)] \\
&\quad + \frac{\epsilon^2}{2} \sum_{smn} D_{smn}(\overline{\sigma}) \frac{\partial^2 S(y, t)}{\partial \mathbf{u}_{sm} \partial \mathbf{u}_{sn}} \\
&\quad + \frac{1}{t^2} \frac{1}{2} \sum_{rr'} \overline{f}_r (\delta_{rr'} - \overline{f}_{r'}) \frac{\partial^2 S(y, t)}{\partial x_r \partial x_{r'}}
\end{aligned}$$

where

$$\overline{f}_s = \int_{A_s^0(\overline{\sigma})} d\mathbf{v} P(\mathbf{v})$$

$$D_{smn}(\overline{\sigma}) = \int_{A_s^0(\overline{\sigma})} d\mathbf{v} P(\mathbf{v}) (\mathbf{v}_m \mathbf{v}_n - \overline{\mathbf{v}}_{sm} \overline{\mathbf{v}}_{sn})$$

and

$$\overline{\mathbf{v}}_{sm} = \frac{1}{\overline{f}_s} \int_{A_s^0(\overline{\sigma})} d\mathbf{v} P(\mathbf{v}) \mathbf{v}_m.$$

We make the notation more compact by defining

$$\begin{aligned}
\mathbf{u}_{s,\, d+1} &= x_s \\
B_{sm}(t) &= \begin{cases} \epsilon \overline{f}_s & \text{for } m = 1, 2, \cdots, d \\ \dfrac{1}{t} & \text{for } m = d+1, \end{cases} \\
\Delta_{rmr'n}(t) &= \begin{cases} \epsilon^2 D_{smn}(\overline{\sigma}) & \text{for } r = r' = s \text{ and} \\ & \qquad m \le d,\ n \le d \\ \dfrac{1}{t^2} \overline{f}_r (\delta_{rr'} - \overline{f}_{r'}) & \text{for } m = n = d+1 \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}$$

Thus, we obtain the following linear FPE with time-dependent coefficients

$$\partial_t S(y,t) = \sum_{sm} B_{sm}(t)\,\frac{\partial}{\partial \mathbf{u}_{sm}}\,(\mathbf{u}_{sm}S) + \frac{1}{2}\sum_{sms'n} \Delta_{sms'n}(t)\,\frac{\partial^2 S}{\partial \mathbf{u}_{sm}\partial \mathbf{u}_{s'n}}.$$

Assuming a given value of this process at time t_0, the mean $\langle\cdot\rangle_t$ at time t is given as

$$\langle \mathbf{u}_{sm}\rangle_t = \exp\left[-\overline{f}_s \int_{t_0}^{t} \epsilon(\tau)\,d\tau\right]\langle \mathbf{u}_{sm}\rangle_{t_0} \quad \text{for } m = 1, 2, \cdots, d$$

$$\langle \mathbf{u}_{s,\,d+1}\rangle_t = \frac{t_0}{t}\,\langle \mathbf{u}_{s,\,d+1}\rangle_{t_0}.$$

Necessary and sufficient condition for this mean value to vanish as $t \to \infty$ is

$$\int_{t_0}^{\infty} \epsilon(\tau)\,d\tau = \infty.$$

The covariances

$$Z_{sms'n} \triangleq \langle\langle \mathbf{u}_{sm}\mathbf{u}_{s'n}\rangle\rangle = \langle \mathbf{u}_{sm}\mathbf{u}_{s'n}\rangle - \langle \mathbf{u}_{sm}\rangle\,\langle \mathbf{u}_{s'n}\rangle$$

obey the differential equation

$$\partial_t Z_{sms'n} = -B_{sm}(t)Z_{sms'n} - B_{s'n}(t)Z_{sms'n} + \Delta_{sms'n}(t)$$

with solution

$$Z_{sms'n}(t) = Y_{sm}(t)\left\{Z_{sms'n}(t_0) + \int_{t_0}^{t} Y_{sm}^{-1}(\tau) \cdot \Delta_{sms'n}(\tau) Y_{s'n}^{-1}(\tau)\,d\tau\right\} Y_{s'n}(t)$$

where

$$Y_{sm}(t) \triangleq \frac{\langle \mathbf{u}_{sm}\rangle_t}{\langle \mathbf{u}_{sm}\rangle_{t_0}}.$$

The covariance matrix Z vanishes, as $t \to \infty$, if equivalently its diagonal terms (variances) vanish. The variances obey the differential equation

$$\partial_t \langle \mathbf{u}_{sm}^2\rangle = -2\epsilon\overline{f}_s\langle \mathbf{u}_{sm}^2\rangle + \epsilon^2 D_{smm}(\overline{\sigma})$$

with solution

$$\langle \mathbf{u}_{sm}^2\rangle_t = \exp\left[-\int_{t_0}^{t} 2\overline{f}_s\epsilon(\tau)\,d\tau\right] \cdot \left[\int_{t_0}^{t} D_{smm}(\overline{\sigma})\epsilon^2(t')\,dt' + \langle \mathbf{u}_{sm}^2\rangle_{t_0}\right].$$

This tends to zero as $t \to \infty$ if equivalently

$$\int_{t_0}^{\infty} \epsilon(\tau)\,d\tau = \infty.$$

Thus, the necessary and sufficient conditions for the mean and covariance to converge to zero are

$$\epsilon(t) \to 0, \qquad \text{and} \quad \int_{t_0}^{\infty} \epsilon(\tau)\,d\tau = \infty. \tag{4}$$

These conditions are of the same type as those encountered in [8] to guarantee weak convergence of stochastic approximation processes. The similarity is not surprising since weak convergence is similar to convergence in distribution and in our analysis we demand that the mean and variance of deviations from the equilibrium vanish.

We also note that the final phase of FSCL learning imposes conditions only on the learning rate and not on the fairness function. As discussed earlier in the analysis, the same results are obtained for any reasonable choice of the fairness function. The exact form of the fairness function affects the global behavior of the algorithm, its ability to approach the global minimum of the total distortion measure and it also determines the set of possible equilibrium states of the algorithm. The effect of the fairness function on the equilibrium codeword distribution has been studied in [3].

IV. Conclusion

Our conclusion is that, by selecting learning rates that offer sufficient excitation, one expects, regardless of the specific fairness function, to converge to a solution that is locally optimal. The fairness function can then be independently selected, e.g., to yield a desired codeword distribution [3]. Thus, this result supports the use of FSCL clustering for applications such as VQ codebook design, unsupervised clustering for mixture density analysis, and design of radial basis funcsions.

References

[1] S. C. Ahalt, A. K. Krisnamurthy, P. Chen, and D. E. Melton, “Competitive learning algorithms for vector quantization,” *Neural Networks*, vol. 3, pp. 277–290, 1990.

[2] M. Cottrell and J. C. Fort, “A stochastic model of retinotopy: A self organizing process,” *Biol. Cybern.*, vol. 53, pp. 405–411, 1986.

[3] A. S. Galanopoulos and S. C. Ahalt, “Codeword distribution for frequency sensitive competitive learning with one-dimensional input data,” *IEEE Trans. Neural Networks*, vol. 7, pp. 752–756, May 1996.

[4] S. Grossberg, “Competitive learning: From interactive activation to adaptive resonance,” *Cognitive Sci.*, vol. 11, pp. 23–63, 1987.

[5] B. Hajek, “A tutorial survey of theory and applications of simulated annealing,” in *Proc. 24th Conf. Decision Contr.*, Ft. Lauderdale, FL, Dec. 1985.

[6] T. Kohonen, “Self-organized formation of topologically correct feature maps,” *Biol. Cybern.*, vol. 43, pp. 59–69, 1982.

[7] ——, *Self-Organizing Maps.* Berlin: Springer-Verlag, 1995.

[8] H. J. Kushner and D. S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems.* Berlin: Springer-Verlag, 1978.

[9] Y. Linde, A. Buzo, and R. M. Gray, “An algorithm for vector quantizer design,” *IEEE Trans. Commun.*, vol. 28, pp. 84–95, 1980.

[10] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *J. Chem. Physics*, vol. 21, no. 6, pp. 1087–1092, June 1953.

[11] H. Ritter and K. Schulten, “Convergence properties of Kohonen’s topology conserving maps: Fluctuations, stability, and dimension selection,” *Biol. Cybern.*, vol. 60, pp. 59–71, 1988.

[12] H. Robbins and S. Monro, “A stochastic approximation method,” *Ann. Math. Statist.*, vol. 22, pp. 400–407, 1951.

[13] D. E. Rumelhart and D. Zipser, “Feature discovery by competitive learning,” *Cognitive Sci.*, vol. 9, pp. 75–112, 1985.

[14] E. Yair, K. Zeger, and A. Gersho, "Competitive learning and soft competition for vector quantizer design," *IEEE Trans. Signal Processing,* vol. 40, Feb. 1992.

Aristides S. Galanopoulos (S'86–M'87) received the B.S. degree in electrical engineering from the National Technical University of Athens, Greece, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from the Ohio State University, Columbus, in 1989 and 1996, respectively.

Since 1995, he is with Northern Telecom, Broadband Systems Engineering, Richardson, TX. His research interests include communication networks and neural networks.

Randolph L. Moses (S'78–M'83–SM'90) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Virginia Polytechnic Institute and State University, Blacksburg, in 1979, 1980, and 1984, respectively.

During the summer of 1983, he was a SCEEE Summer Faculty Research Fellow at Rome Air Development Center, Rome, NY. From 1984 to 1985, he was with the Eindhoven University of Technology, Eindhoven, The Netherlands, as a NATO Postdoctoral Fellow. Since 1985, he has been with the Department of Electrical Engineering, The Ohio State University, Columbus, and is currently a Professor there. During 1994 to 1995, he was on sabbatical leave as a visiting researcher at the System and Control Group at Uppsala University in Sweden. His research interests are in digital signal processing, and include parametric time series analysis, radar signal processing, and sensor array processing.

Dr. Moses served on the Technical Committee on Statistical Signal and Array Processing of the IEEE Signal Processing Society from 1991 to 1994. He is a member of Eta Kappa Nu, Tau Beta Pi, Phi Kappa Phi, and Sigma Xi.

Stanley C. Ahalt (S'77–M'79) received the B.S. and M.S. degrees in electrical engineering from Virginia Polytechnic Institute and State University, Blacksburg, in 1978 and 1980, respectively. He received the Ph.D. degree in electrical and computer engineering from Clemson University, SC, in 1986.

He was a member of the Technical Staff at Bell Telephone Laboratories from 1980 to 1981. He has been with the Ohio State University Department of Electrical Engineering, Columbus, since 1987, where he is currently an Associate Professor. During 1996, he was on sabbatical leave as a Visiting Professor at the Universidad Polytecnica de Madrid, Spain. His research is in neural networks, with application to image analysis, automatic target recognition, image processing, and special purpose hardware to support real-time processing.