Constructive Feedforward ART Clustering Networks—Part II

Andrea Baraldi and Ethem Alpaydın

***Abstract*—Part I of this paper defines the class of constructive unsupervised on-line learning simplified adaptive resonance theory (SART) clustering networks. Proposed instances of class SART are the symmetric Fuzzy ART (S-Fuzzy ART) and the Gaussian ART (GART) network. In Part II of our work, a third network belonging to class SART, termed fully self-organizing SART (FOSART), is presented and discussed. FOSART is a constructive, soft-to-hard competitive, topology-preserving, minimum-distance-to-means clustering algorithm capable of: 1) generating processing units and lateral connections on an example-driven basis and 2) removing processing units and lateral connections on a minibatch basis. FOSART is compared with Fuzzy ART, S-Fuzzy ART, GART and other well-known clustering techniques (e.g., neural gas and self-organizing map) in several unsupervised learning tasks, such as vector quantization, perceptual grouping and 3-D surface reconstruction. These experiments prove that when compared with other unsupervised learning networks, FOSART provides an interesting balance between easy user interaction, performance accuracy, efficiency, robustness, and flexibility.**

***Index Terms*—Absolute and relative membership function, adaptive resonance theory, clustering, Delaunay triangulation, soft-to-hard competitive learning, topology preserving mapping, Voroni partition.**

I. Introduction

In PART I of this paper, the symmetric Fuzzy ART (S-Fuzzy ART) and Gaussian ART (GART) networks are proposed as two instances of the simplified adaptive resonance theory (SART) group of ART clustering algorithms (see Part I, Section VII). In Part II of this paper, we present and discuss a novel constructive unsupervised on-line learning SART algorithm, termed fully self-organizing SART (FOSART), designed to address all recommendations proposed in Part I, Section IV-C, to overcome potential weaknesses of Fuzzy ART.

With respect to existing clustering algorithms, FOSART aims at combining useful properties derived from well-known clustering networks such as neural gas (NG) [1], self-organizing map (SOM) [2], [3], and growing neural gas (GNG) [4], [5] (for a review, refer to [6] and [7]).

NG is successful because it minimizes a known cost function which converges on the hard c-means quantization error via a soft-to-hard competitive model transition. NG employs no lateral connection and a fixed number of processing units. Limitations of NG are that:

- topology-preserving mapping as defined in [8] is not pursued;
- prototype parameter estimates may be affected by noise points and outliers because learning rates are computed independently of the actual distance separating the input pattern from the cluster template.

Unlike NG, SOM employs internode distances in a fixed output lattice rather than interpattern distances in input space to compute learning rates. Noticeably, SOM deals with topological relationships (e.g., adjacency) among output nodes without employing any explicit model of internode (lateral) connectivity. Despite its many successes in practical applications, SOM has some limitations, most of which are acknowledged in [2].

- Termination is not based on optimizing any model of the process or its data [9]. Indeed, it has been shown that an objective function cannot exist for the SOM algorithm, i.e., there exists no cost function yielding Kohonen's adaptation rule as its gradient [10], [11]. SOM instead features a set of potential functions, one for each node, to be independently minimized following a stochastic (on-line) gradient descent [10]. In [12], a cost function that leads to an update strategy that is similar to, but not precisely the same as, that of SOM is discussed. This cost function, originally introduced in a nonneural context to design an optimal vector quantizer codebook for encoding data for transmission along a noisy channel [13], has recently been generalized in [14].
- The size of the output lattice, the learning rate and the size of the resonance neighborhood must be varied empirically from one data set to another to achieve useful results [9].
- Topology preserving mapping as defined in [8] is not guaranteed.
- Prototype parameter estimates may be severely affected by noise points and outliers.

GNG, which is capable of generating and removing neurons and lateral connections dynamically, features an expressive power potentially superior to that of NG and SOM. In GNG, lateral connections are generated according to the competitive Hebbian learning rule (CHR) [8]. To remove links, generate neurons and remove neurons, GNG adopts heuristics based on mini-batch statistics, i.e., statistics collected over subsets of the input sequence to average information over the noise on the data. Limitations of GNG are that:

- it minimizes no known cost function;
- its heuristics employ up to seven user-defined parameters.

Unlike Fuzzy ART, which belongs to the ART 1-based group of networks (see Part I, Section III) and to the class of hy-

Manuscript received May 3, 1999; revised February 8, 2001. This work was supported in part by the Italian Space Agency (ASI) under Contract 98-135.

A. Baraldi is with ICSI, Berkeley, CA, and ISAO-CNR, Bologna, Italy.

E. Alpaydın is with ICSI, Berkeley, CA, and Boğgaziçi University, Istanbul, Turkey.

Publisher Item Identifier S 1045-9227(02)04450-8.

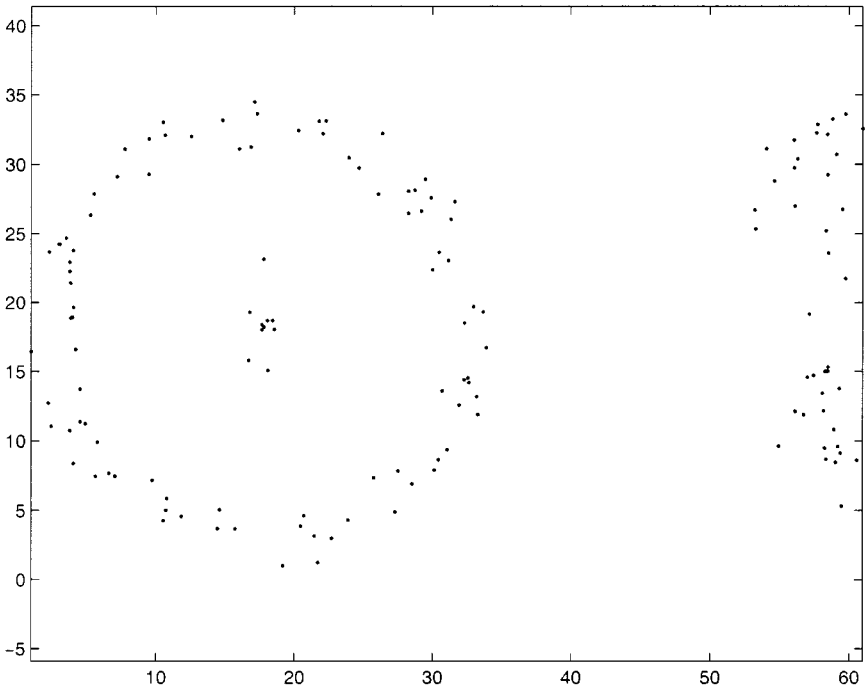

Fig. 1. Nonconvex data set consisting of 140 data points belonging to a circular ring plus three Gaussian clusters.

perbox clustering algorithms (see Part I, Section IV), and unlike GART, which belongs to the SART group of networks (see Part I, Section VII) and to the class of maximum likelihood (ML) probability density function estimators for Gaussian mixtures (see Part I, Appendix IV), FOSART is designed as a minimum-distance-to-means clustering network [15], that tries to minimize a sum-of-squares (also termed quantization or distorsion) error. This means that FOSART generates a partition of data space where clusters' receptive fields (regions of support) are parameterized in terms of their center of mass, i.e., FOSART belongs to the class of clustering-by-replacement algorithms. It has been proved that clustering-by-replacement algorithms, in which cluster prototypes are extrapolated, rather than selected, from the data set, can be more efficient than clustering-by-selection techniques, where cluster selection finds a proper subset of the input data set [16]. Our idea is that the combination of clustering-by-replacement with lateral-connection adaptation mechanisms may allow FOSART to deal with nonconvex data structures (particularly relevant in perceptual grouping problems [17], [18], e.g., curved string-like or concentric structures such as those shown in Fig. 1) that may be difficult to detect with the hyperbox clustering approach of Fuzzy ART.

Part II of this paper is organized as follows: in Section II, FOSART is presented. An experimental comparison between FOSART, GART and other well-known clustering algorithms is proposed in Section III; conclusions are reported in Section IV.

II. FOSART

In synthesis, FOSART:

- is a clustering-by-replacement and minimum-distance-to-means clustering network (i.e., it tries to minimize a quantization error);
- belongs to the SART clustering framework (see Part I, Section VII-B) and can be implemented according to version 2 of the efficient ART (EART) implementation framework (see Part I, Section II-B2);
- employs a soft-to-hard competitive model transition, which is adapted from NG, to minimize a quantization error [1], [4];
- generates processing elements (PEs) dynamically, on an example-driven basis, according to the vigilance test of class SART [in Part I, see (3) and Section VII-B], i.e., an individual input example suffices to initiate the creation of a new unit;
- removes PEs dynamically, based on a mini-batch learning framework, i.e., based on statistics collected over subsets of the input sequence [4];
- generates lateral connections between processing unit pairs dynamically, based on an example-driven mechanism derived from the CHR [4], [5], [8], [19], [20];
- removes lateral connections between unit pairs dynamically, based on a mini-batch learning framework.

Note that while example-driven parameter adaptation is very sensitive to the presence of noise, mini-batch learning gains in robustness by collecting statistics which are averaged over the noise on the data.

A. Distorsion Error Minimization in NG and FOSART

Given a presentation sequence of m unlabeled analog patterns $\mathbf{X_i} \in \mathcal{R}^d$, $i = 1, \ldots, m$, where d is the dimensionality of input space, unsupervised learning systems detect a set of parameters capable of modeling hidden data structures (e.g., linear substructures), statistical data regularities or probability density functions [4]. Among unsupervised learning tasks, the problem of clustering is that of separating the unlabeled data

set into groups (i.e., hidden data structures), called clusters, for which samples within a cluster are more similar than samples from different clusters. Usually, vector prototypes, also called reference or template vectors $\mathbf{W_j} \in \mathcal{R}^d$, $j = 1, \ldots, c$, with $c \leq m$, are generated to characterize the members of a cluster as a group. Since the goal of clustering is to group the data at hand rather than provide an accurate characterization of unobserved (future) samples generated from the same probability distribution, the task of clustering may fall outside the framework of predictive (inductive) learning. In spite of this, clustering analysis often employs unsupervised learning techniques originally developed for vector quantization, which is a predictive learning problem [21].

In this framework, a frequent goal of clustering systems is the minimization of the *distorsion (quantization, reconstruction) error*, identified as the mean square error (MSE), defined as

$$E_{dis} = MSE = \frac{1}{m}\sum_{i=1}^{m} \left\|\mathbf{X_i} - \mathbf{W_{w1(i)}}\right\|^2 \qquad w1(i) \in \{1, c\} \tag{27}$$

where m is the size of a finite data set and $w1(i)$ is the index of best-matching template $\mathbf{W_{w1(i)}}$ detected as

$$\left\|\mathbf{X_i} - \mathbf{W_{w1(i)}}\right\| \leq \|\mathbf{X_i} - \mathbf{W_j}\| \qquad w1(i) \in \{1, c\},\ j = 1, \ldots, c \tag{28}$$

where symbol $\|\cdot\|^2 = (\mathbf{X_i} - \mathbf{W_{w1(i)}})^T(\mathbf{X_i} - \mathbf{W_{w1(i)}})$ identifies the square Euclidean distance. Equation (27) describes a region of support of an output unit as a Voronoi polyhedron centered on its reference vector, the whole set of reference vectors providing a partition of the input space known as Voronoi tesselation [4], [8]. Voronoi tessellation is the dual of Delaunay triangulation, which is a peculiar form of triangulation in various geometrical and functional respects [4], [22], [23]. Equation (27) can be considered the hard competitive version of the more general, soft competitive *distance-weighted sum-of-squares clustering cost function* [1], [9], [24]

$$E_{dwdis} = \sum_{i=1}^{m}\sum_{j=1}^{c} \|\mathbf{X_i} - \mathbf{W_j}\|^2 k_j\left(d\left(\mathbf{X_i}, \hat{\mathbf{W}}\right)\right) \tag{29}$$

where symbol $d(\mathbf{X_i}, \hat{\mathbf{W}}) = \{d(\mathbf{X_i}, \mathbf{W_1}), \ldots, d(\mathbf{X_i}, \mathbf{W_c})\}$ identifies the set of interpattern distances between data point $\mathbf{X_i}$ and each vector prototype in codebook $\hat{\mathbf{W}} = \{\mathbf{W_1}, \ldots, \mathbf{W_c}\}$, where prototype $\mathbf{W_j}$, $j = 1, \ldots, c$, is the center of mass of the region of support (receptive field) of the jth processing unit, while term $k_j(d(\mathbf{X_i}, \hat{\mathbf{W}})) \geq 0$ is a *distance-weighting function* [25], also termed *kernel function* [21], subjected to a set of constraints specified in [21, p. 222], such that function $k_j(d(\mathbf{X_i}, \hat{\mathbf{W}}))$ is monotonically nonincreasing with distance $d(\mathbf{X_i}, \mathbf{W_j})$ and monotonically nondecreasing with distances $d(\mathbf{X_i}, \mathbf{W_h})$, $h = 1, \ldots, c$, $h \neq j$. For example [9]

$$k_j\left(d\left(\mathbf{X_i}, \hat{\mathbf{W}}\right)\right) = \frac{\frac{1}{d(\mathbf{X_i}, \mathbf{W_j})^2}}{\sum_{h=1}^{c} \frac{1}{d(\mathbf{X_i}, \mathbf{W_h})^2}}. \tag{30}$$

Equation (30) guarantees that

$$\sum_{j=1}^{c} k_j\left(d\left(\mathbf{X_i}, \hat{\mathbf{W}}\right)\right) = 1 \tag{31}$$

which is not always the case, e.g., see (39). This implies that weighting function $k_j(d(\mathbf{X_i}, \hat{\mathbf{W}}))$ is a mathematical tool providing a model for "network-wide internode communication by subsuming that processing elements are coupled through feedsideways (lateral) connections" [26].

The necessary condition that guarantees approximate minimization of (29) is [52]

$$\frac{\partial E_{dwdis}}{\partial \mathbf{W_j}} = -2\sum_{i=1}^{m}(\mathbf{X_i} - \mathbf{W_j})k_j\left(d\left(\mathbf{X_i}, \hat{\mathbf{W}}\right)\right) + R_j = 0 \qquad j = 1, \ldots, c \tag{32}$$

where

$$R_j = \sum_{i=1}^{m}\sum_{h=1}^{c}(\mathbf{X_i} - W_h)^2 \frac{\partial k_h\left(d\left(\mathbf{X_i}, \hat{\mathbf{W}}\right)\right)}{\partial \mathbf{W_j}} \qquad j = 1, \ldots, c. \tag{33}$$

If we assume that

$$R_j = 0, \qquad j = 1, \ldots, c \tag{34}$$

then the iterative batch solution of (33) becomes

$$\mathbf{W_j^{(e+1)}} = \frac{\sum_{i=1}^{m} \mathbf{X_i} k_j\left(d\left(\mathbf{X_i}, \hat{\mathbf{W}}^{(e)}\right)\right)}{\sum_{g=1}^{m} k_j\left(d\left(\mathbf{X_g}, \hat{\mathbf{W}}^{(e)}\right)\right)} \qquad j = 1, \ldots, c \tag{35}$$

where variable e identifies the number of processing epochs, i.e., the number of times the finite data set is repeatedly presented to the network. It is easy to prove that if $k_j(d(\mathbf{X_i}, \hat{\mathbf{W}}^{(e)})) = 1$ when $j = w1(i)$ based on (28), and zero otherwise, then (34) is satisfied, (29) converges at (27) and (35) becomes the classical hard c-means (HCM) update expression. Notice that (35) computes any template vector as a convex combination of input patterns: since the convex combination of a nonconvex data set may lie well outside the data manifold, it is obvious that (27) and (29) cannot perform well for nonconvex types of data [27].

If the assumption about vanishing term R_j holds, i.e., if (34) holds, the recursive batch gradient descent solution of (29) is defined as

$$\mathbf{W_j^{(e+1)}} \doteq \mathbf{W_j^{(e)}} - \epsilon(e)\frac{\partial E_{dwdis}}{\partial \mathbf{W_j^{(e)}}} = \mathbf{W_j^{(e)}} + \epsilon(e)\sum_{i=1}^{m}\left(\mathbf{X_i} - \mathbf{W_j^{(e)}}\right)k_j\left(d\left(\mathbf{X_i}, \hat{\mathbf{W}}^{(e)}\right)\right) \qquad j = 1, \ldots, c \tag{36}$$

where learning rate $\epsilon(e)$ has to satisfy the three conditions applied to the coefficients of the Robbins–Monro algorithm for

finding the roots of a function iteratively (in our case, the function whose roots are investigated is $\partial E_{dwdis}/\partial \mathbf{W_j}$). These conditions are [24, pp. 47 and 96], [28]

$$1) \lim_{t\to\infty} \epsilon(t) = 0; \quad 2) \sum_{t=1}^{\infty} \epsilon(t) = \infty; \quad 3) \sum_{t=1}^{\infty} \epsilon^2(t) < \infty.$$

For example, when $\epsilon(t) = 1/t$ (harmonic series) [4], [9], [24, p. 96], then $\epsilon(t)$ decreases monotonically with t under Robbins–Monro conditions. Condition 3) states that learning rate $\epsilon(t)$ must decrease fairly quickly, while condition 2) limits the rate of decrease of the learning rate: indeed, if this rate of decrease is too quick, then it could stop the progression of the algorithm toward the minimum [29]. According to condition 2), the infinite sum of the learning rates diverges. This is tantamount to saying that even after a large number of input signals and correspondingly low values of the learning rate $\epsilon(t)$, arbitrarily large modifications of reference vectors may occur in principle, although they are most unlikely to occur [4]. When the data distribution is stationary, conditions 1) to 3) are necessary but not sufficient to guarantee that true (batch) and stochastic (on-line) gradient descent algorithms converge to a point in the parameter space [29].

In the NG algorithm [1], a sequential (stochastic, on-line) update process, whose goal is to avoid the storage of all data points by assuming that they are arriving one at a time, is derived from (36) by dropping the sum over input patterns [24]. A different approach is to separate out from (36) the contribution of the $(m+1)$th data point [24], [30], which gives [52]

$$\mathbf{W_j}^{(t+1)} = \mathbf{W_j}^{(t)} + \beta_j^{(t)} \cdot \left(\mathbf{X}^{(t)} - \mathbf{W_j}^{(t)}\right), \qquad j = 1, \ldots, c \tag{37}$$

where

$$\beta_j^{(t)} = \frac{k_j\left(d\left(\mathbf{X}^{(t)}, \hat{\mathbf{W}}^{(t)}\right)\right)}{\sum_{T=1}^{t} k_j\left(d\left(\mathbf{X}^{(T)}, \hat{\mathbf{W}}^{(T)}\right)\right)}. \tag{38}$$

In the NG algorithm, the distance-weighting function is defined as [1]

$$k_j\left(d\left(\mathbf{X}^{(t)}, \hat{\mathbf{W}}^{(t)}\right)\right) = k_\lambda\left(r_j\left(\mathbf{X}^{(t)}, \hat{\mathbf{W}}^{(t)}\right)\right) = k_\lambda\left(r_j^{(t)}\right) = e^{-r_j(\mathbf{X}^{(t)}, \hat{\mathbf{W}}^{(t)})/\lambda(t)} \tag{39}$$

where $\lambda(t)$ is a scale parameter, monotonically decreasing with time, which controls the degree of overlap (degree of fuzziness) between receptive fields and $r_j^{(t)} = r_j(\mathbf{X}^{(t)}, \hat{\mathbf{W}}^{(t)})$ is the neighborhood-ranking of vector $\mathbf{W_j}^{(t)}$ such that $r_j^{(t)} = 0$ if $\mathbf{W_j}^{(t)}$ is the best-matching template, i.e., $\mathbf{W_j}^{(t)} = \mathbf{W_{w1(t)}}^{(t)}$, otherwise $r_j^{(t)} = 1$ if $\mathbf{W_j}^{(t)}$ is the second best-matching template, etc. A possible expression for $\lambda(t)$ is

$$\lambda(t) = \lambda_{ini}(\lambda_{fin}/\lambda_{ini})^{t/t_{\max}} \tag{40}$$

where $t_{\max}$ is the maximum number of input presentations, while $\lambda_{ini} \geq \lambda_{fin}$. Widely employed settings for these parameters are $\lambda_{ini} = 5$, and $\lambda_{fin} = 0.01$ [1], [26], [31].

It can be proved that (39) satisfies condition (34), i.e., (37) and (38) hold [1]. In this case, when $\lambda(t) \to \lambda_{fin} \approx 0$ in line with (40), then (29) becomes equivalent to (27). This learning strategy is the soft-to-hard competitive model transition implemented by NG to minimize (27) while aiming at preventing the set of reference vectors from being trapped in suboptimal states.

Note that implementation of (39) is time-consuming: for each input pattern, computational complexity of neighborhood-ranking is $c \log c$. However, it has been shown, both theoretically and empirically, that NG performs in almost the same way when only few PEs are considered coupled at a time, i.e., NGs performance is not affected when soft competitive learning involves only a few (five to ten) "top" positions in the list of sorted distances [26].

Our aim is to adapt (37)–(40) to FOSART. First, FOSART generates and removes output units dynamically at different presentation times ts. Thus, (40) is replaced by

$$\lambda_j^{(t)} = \lambda_{ini}(\lambda_{fin}/\lambda_{ini})^{e_j^{(t)}/e_{\min}}, \qquad j = 1, \ldots, c(t) \tag{41}$$

where $e_{\min}$, λ_{ini} and λ_{fin} are user-defined parameters (see Section II-C1), while $e_j^{(t)}$ is a PE-based (local) variable counting the "age" of the jth processing unit as the number of times the finite input data set has been iteratively presented to the system while that processing unit exists (see further Section II-C1). Second, in FOSART, (potentially) coupled PEs are those topologically connected through lateral connections, i.e., (potentially) coupled PEs belong to the same output map. Thus, in FOSART, at a given presentation time t

1) best-matching unit $E_{w1(t)}^{(t)}$ is detected;
2) soft-to-hard competitive update equations, (37)–(39) and (41), are applied to neural units belonging to the same map of best-matching unit $E_{w1(t)}^{(t)}$, where $r_j^{(t)} = 1$ if processing unit $E_j^{(t)}$ is directly linked to $E_{w1(t)}^{(t)}$, $r_j^{(t)} = 2$ if processing unit $E_j^{(t)}$ is indirectly linked to $E_{w1(t)}^{(t)}$, but directly linked to any processing unit featuring neighborhood-ranking equal to one, etc.

The advantage of this strategy is that FOSART detects neighbors of unit $E_{w1(t)}^{(t)}$ efficiently (by means of lateral connections), i.e., no time-consuming neighborhood-ranking is required. The disadvantage is that the FOSART minimization of cost function (27) has a less rigorous mathematical foundation than that of NG.

B. Generation of Lateral Connections

Delaunay triangulation is the only form of triangulation in which the circumcircle of each triangle contains no other point from the original point set than the vertices of this triangle [4]. The dual of Delaunay triangulation is a Voronoi diagram, defined as the graph that connects, in the data space, each vector pair that has adjacent Voronoi polyhedra (receptive fields). It has been proved that Delaunay triangulation and its dual, the Voronoi diagram, solve or, at least, yield a starting point for efficiently solving proximity problems in a metric space (such as the k-nearest neighbor search, the Euclidean minimum spanning tree, the traveling salesman problem, etc.) [8], [22]. Moreover, Delaunay triangulation has proved to be optimal for piecewise linear function regression over a triangulation of the input samples [22], [23].

By projecting input patterns onto a network of processing units such that similar patterns are projected onto units adjacent in the network and, vice versa, such that adjacent units in the network code similar patterns, topology preserving mapping (TPM) in the sense proposed in [8] plays an important role in a variety of natural as well as artificial distributed processing systems. Examples of TPMs in the nervous system are the retinotopic map in the visual cortex and the mapping from the body surface onto the somatosensory cortex [32].

The CHR is an example-driven connection rule such that if, at presentation time t, input pattern $\mathbf{X}^{(t)}$, extracted from input manifold $\mathcal{X} \subseteq \mathcal{R}^d$, features output units $E^{(t)}_{w1(t)}$ and $E^{(t)}_{w2(t)}$ as the best and second-best matching, respectively, then a lateral connection between this unit pair is established [8]. Under the hypothesis that the distribution of reference vectors (codebook) $\hat{\mathbf{W}}^{(t)} = \{\mathbf{W}^{(t)}_{\mathbf{1}}, \ldots, \mathbf{W}^{(t)}_{\mathbf{c}}\}$, is dense on $\mathcal{X}$, i.e., for each input $\mathbf{X}^{(t)}$, triangle $\triangle(\mathbf{X}^{(t)}, \mathbf{W}^{(t)}_{\mathbf{w1(t)}}, \mathbf{W}^{(t)}_{\mathbf{w2(t)}})$ lies completely on $\mathcal{X}$, it is proved that CHR forms an output graph (lattice) which is the *induced Delaunay triangulation* of codebook $\hat{\mathbf{W}}^{(t)}$ and forms a perfectly TPM of $\mathcal{X}$ in the sense proposed in [8].

GNG, which employs CHR [4], ignores the above requirement about codebook $\hat{\mathbf{W}}^{(t)}$ being dense (e.g., see simulations at: http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VIDM/research/gsn/DemoGNG/GNG.html). To account for this constraint we propose a constrained CHR (CCHR) version, based on the following heuristic criterion: given input pattern $\mathbf{X}^{(t)}$ and the best and second-best matching units $E^{(t)}_{w1(t)}$ and $E^{(t)}_{w2(t)}$, a lateral connection between this unit pair is established iff

$$\left\|\mathbf{X}^{(t)} - \mathbf{W}^{(t)}_{\mathbf{w2(t)}}\right\| \Big/ \left\|\mathbf{X}^{(t)} - \mathbf{W}^{(t)}_{\mathbf{w1(t)}}\right\| \leq er \qquad (42)$$

where $er \geq 1$ is a user-defined edge ratio threshold, such that tetrahedra in the induced Delaunay triangulation have circumradius-to-shortest edge ratio below threshold er. In a Delaunay triangulation, the circumradius-to-shortest edge ratio is a measure inversely proportional to the quality of a simplex, i.e., one would like this ratio to be as close to one as possible [23]. Threshold er, which may be scale-dependent in perceptual grouping, has a default value equal to 1.62, based on empirical evidence. Note that this default value is the so-called aurea measure and is also considered a quality bound in [23].

C. The FOSART Algorithm

To overcome potential weaknesses of Fuzzy ART, FOSART is consistent with the SART clustering framework proposed in Part I, Section VII-B, and tries to address all recommendations proposed in Section IV-C of Part I.

For simplicity's sake, we assume to deal with an analog data set which is finite, i.e., the input data set consists of $0 < m < \infty$ analog data vectors in $\mathcal{R}^d$. This finite data set of size m is repeatedly presented to the clustering network until a termination criterion is satisfied. Each presentation sequence is termed a training epoch. Adaptive parameters $\mathbf{W}^{(t)}_{\mathbf{j}}$, $j = 1, \ldots, c(t)$, also belong to input space $\mathcal{R}^d$.

1) Input Parameters: FOSART requires the user to define:

- An ART-based vigilance threshold as a relative number $\rho \in (0, 1]$, such that coarser grouping of input patterns is obtained when the vigilance parameter is lowered. In other words, ρ is a model of top-down external requirements (expectations, or prior knowledge) provided by the external environment (supervisor) (see Part I, Section II-A).
- An edge ratio threshold $er \geq 1$ (see Section II-B). By default, $er = 1.62$ (aurea section).
- To reach termination, FOSART requires a lower limit for the number of training epochs during which each node has to survive, $e_{\min} \geq 1$, this parameter affecting the overall number of training epochs required by the algorithm to reach termination (consider that, in FOSART, units are generated and removed dynamically as the number of input pattern presentations, t, increases).
- Scale variables [see $\lambda^{(t)}_j$, $j = 1, \ldots, c(t)$, in (41)] are bounded by parameters $\lambda_{ini} \geq \lambda_{fin}$ which control the soft-to-hard competitive learning transition. By default, we employ $\lambda_{ini} = 0.5$ and $\lambda_{fin} = 0.01$. Note that $\lambda_{fin} \approx 0$ is set by the application developer, i.e., it does not need to be user-defined.[1]

2) Implementation Scheme: FOSART is implemented according to version 2 of the EART processing framework proposed in Part I, Section II-B2.

Step 0. Initialization: Pattern counter t is set to zero. One input pattern $\mathbf{X}^{(t)}$ is chosen (either sequentially or randomly) from the input data set. Next, processing element (PE) counter $c(t+1)$ is set to one and processing unit $E^{(t+1)}_{c(t+1)}$ is generated such that $\mathbf{W}^{(t+1)}_{\mathbf{c(t+1)}} = \mathbf{X}^{(t)}$. Mini-batch PE-based (local) epoch counter $e^{(t+1)}_{c(t+1)}$ is initialized to zero. FOSART employs PE-based epoch counters to compute PE-based learning rates. In FOSART, the "age" (local time) of a processing unit is an integer value equal to the number of times the finite input data set has been iteratively presented to the system while that processing unit exists. Mini-batch PE-based (local) best-matching counter, $bm^{(t+1)}_{c(t+1)}$, is initialized to one. Scale parameter σ in Gaussian activation functions is set equal to

$$\sigma = 1/\rho. \qquad (43)$$

[1]Let us examine the meaning of default values $\lambda_{ini} = 0.5$ and $\lambda_{fin} = 0.01$. In (39), this default option implies that, while for the best-matching template $\mathbf{W}^{(t)}_{\mathbf{w1(t)}}$, whose rank is zero, learning rate $k_{w1(t)}(d(\mathbf{X}^{(t)}, \hat{\mathbf{W}}^{(t)})) = 1$ regardless of $e^{(t)}_{w1(t)}$, for the second best-matching template, $\mathbf{W}^{(t)}_{\mathbf{w2(t)}}$, whose rank is one, if local epoch counter $e^{(t)}_{w2(t)} = 0$ (or, respectively, $\geq e_{\min}$), such that $\lambda^{(t)}_{w2(t)} = \lambda_{ini}$ (or, respectively, $\leq \lambda_{fin}$), then learning rate $k_{w2(t)}(d(\mathbf{X}^{(t)}, \hat{\mathbf{W}}^{(t)})) = \exp(-1/0.5) = 0.135$ [or, respectively, $\leq \exp(-1/0.01) = 3.27 \cdot 10^{(-44)} \approx 0$]. In words, the learning rate of a second best-matching unit $E^{(t)}_{w2(t)}$ decreases from 0.135 to zero as the (local) epoch counter $e^{(t)}_{w2(t)}$ increases from zero to $e_{\min}$. To summarize, in FOSART, scale parameter λ_{ini} controls the initial degree of overlap (i.e., the degree of fuzziness) between receptive fields of processing units (when this scale parameter reduces to zero, receptive fields do not overlap and become equivalent to Voronoi polyhedra). This is evident if we employ three values of $\lambda_{ini} = 5, 0.5$ and 0.01, respectively. In (39), for the second best-matching template whose rank is one, if local counter $e^{(t)}_{w2(t)} = 0$, such that $\lambda^{(t)}_{w2(t)} = \lambda_{ini}$ according to (41), then learning rate $k_{w2(t)}(d(\mathbf{X}^{(t)}, \hat{\mathbf{W}}^{(t)})) = \exp(-1/\lambda_{ini}) = 0.818$, 0.135 and $3.27 \cdot 10^{(-44)} \approx 0$ when $\lambda_{ini} = 5$, 0.5 and 0.01, respectively, see (39). Note that when $\lambda_{ini} = \lambda_{fin} = 0.01$, FOSART is purely hard-competitive because only the learning rate of the winner category, $k_{w1(t)}(d(\mathbf{X}^{(t)}, \hat{\mathbf{W}}^{(t)}))$, is larger than zero (and equal to one), while $k_j(d(\mathbf{X}^{(t)}, \hat{\mathbf{W}}^{(t)})) \approx 0, \forall e^{(t)}_j \geq 0, j = 1, \ldots, c(t), j \neq w1(t)$.

Intuitively, (43) means that if vigilance threshold ρ decreases then it causes coarser grouping by making Gaussian receptive fields larger. When hard competitive learning is enforced among Gaussian units featuring the same spread parameter, then Voronoi tessellation of the input space is accomplished [4].

Step 1. Input Pattern Presentation: The pattern counter is increased by one as $t = t + 1$, and a new pattern $\mathbf{X}^{(t)}$ is chosen (either sequentially or randomly) from the input set and presented to the network.

Step 2. Detection of Processing Units Eligible for Resonance—Activation Value Computation and Best-Matching Unit Selection [See (1) in Part I]: Determine the best and second best-matching units, if any, such that

$$w1(t) = \arg \max_{j=1,\ldots,c(t)} \left\{ AF_{FOSART}\left(\mathbf{X}^{(t)}, \mathbf{W}_{\mathbf{j}}^{(t)}\right)\right\} = \arg \min_{j=1,\ldots,c(t)} \left\{ \left\| \mathbf{X}^{(t)} - \mathbf{W}_{\mathbf{j}}^{(t)} \right\| \right\} \quad (44)$$

$$w2(t) = \arg \max_{j=1,\ldots,c(t);\, j\neq w1(t)} \left\{ AF_{FOSART}\left(\mathbf{X}^{(t)}, \mathbf{W}_{\mathbf{j}}^{(t)}\right)\right\} = \arg \min_{j=1,\ldots,c(t);\, j\neq w1(t)} \left\{ \left\| \mathbf{X}^{(t)} - \mathbf{W}_{\mathbf{j}}^{(t)} \right\| \right\} \quad (45)$$

where the FOSART activation and match functions are the same function defined as

$$AF_{FOSART}\left(\mathbf{X}^{(t)}, \mathbf{W}_{\mathbf{j}}^{(t)}\right) = MF_{FOSART}\left(\mathbf{X}^{(t)}, \mathbf{W}_{\mathbf{j}}^{(t)}\right) = \exp\left(-\frac{\left\| \mathbf{X}^{(t)} - \mathbf{W}_{\mathbf{j}}^{(t)} \right\|^2}{\sigma^2}\right) \in (0, 1] \quad (46)$$

where (46) is a normal absolute membership (NAM) function that satisfies constraints applied by the SART clustering framework to activation and match functions, see Part I, Section VII-B. Since FOSART activation and match functions are identical, it is obviously true that the former function monotonically increases with the latter, and vice versa. Thus, it is proved that FOSART can be implemented efficiently according to version 2 of the EART processing scheme (see, in Part I, Section II-B2 and Table I).

Step 3. Resonance Domain Detection—Vigilance Testing [See (3) in Part I]: If vigilance test

$$MF_{FOSART}\left(\mathbf{X}^{(t)}, \mathbf{W}_{\mathbf{w1(t)}}^{(t)}\right) \geq \rho, \qquad \rho \in (0, 1] \quad (47)$$

is satisfied, then "resonance" occurs [goto Step 4(a)]. Otherwise, goto Step 4(b).

Step 4(a). Resonance Condition—Reinforcement Learning: The following sequence of operations is performed:

1) The best-matching counter of unit $E_{w1(t)}^{(t)}$ is increased by one, i.e., $bm_{w1(t)}^{(t+1)} = bm_{w1(t)}^{(t)} + 1$.
2) Apply CCHR via (42). If (42) holds true, then:
   a) If between output units $E_{w1(t)}^{(t)}$ and $E_{w2(t)}^{(t)}$ a lateral connection $L_{w1(t),\, w2(t)}^{(t)}$ already exists, then increase its local best-matching counter $bm_{w1(t),\, w2(t)}^{(t+1)} = bm_{w1(t),\, w2(t)}^{(t)} + 1$.
   b) Otherwise (i.e., there is no lateral connection), link $L_{w1(t),\, w2(t)}^{(t)}$ is generated and its best-matching local counter is set to one, i.e., $bm_{w1(t),\, w2(t)}^{(t+1)} = 1$.
3) Apply soft-to-hard competitive update equations, (37)–(39) and (41), to output units belonging to the same map of best-matching unit $E_{w1(t)}^{(t)}$, i.e., to output units that are topologically connected to the best-matching unit. Best ranking $r_{w1(t)}^{(t)} = 0$ is assigned to the best-matching unit $E_{w1(t)}^{(t)}$. Next, $r_j^{(t)} = 1$ if processing unit $E_j^{(t)}$ is directly linked to $E_{w1(t)}^{(t)}$, $r_j^{(t)} = 2$ if processing unit $E_j^{(t)}$ is indirectly linked to $E_{w1(t)}^{(t)}$, but directly linked to any processing unit featuring neighborhood-ranking equal to one, etc. (see Section II-A).

Step 4(b). Nonresonance Condition—New Processing Element Allocation: If resonance condition (47) is not satisfied, one new processing unit is dynamically allocated to match external expectations. Thus, the PE counter is increased as $c(t + 1) = c(t) + 1$ and a new node $E_{c(t+1)}^{(t+1)}$ is allocated and initialized such that $\mathbf{W}_{c(t+1)}^{(t+1)} = \mathbf{X}^{(t)}$. As a consequence, FOSART requires no randomization of initial templates since initial values are data-driven. Finally, the PE-based epoch and best-matching counters, $e_{c(t+1)}^{(t+1)}$ and $bm_{c(t+1)}^{(t+1)}$, are initialized to zero and one, respectively.

Step 5. Controls at Epoch Termination: When the entire input data set is presented to the system, i.e., if $[(t\%m) == 0]$, where operator $\%$ computes the remainder of t divided by m, then the following operations occur:

- superfluous cells are removed, such that if output unit $E_j^{(t)}$ features local counter $bm_j^{(t)} == 0$, $j = 1, \ldots, c(t)$, i.e., $E_j^{(t)}$ has not been the best-matching unit in any pattern assignment during the last processing epoch, then it is removed and PE counter $c(t)$ is decreased as $c(t + 1) = c(t) - 1$.
- Superfluous lateral connections are removed, such that $\forall j \in \{1, c(t)\}$, $\forall h \in \{1, c(t)\}$, $h \neq j$, if connection $L_{j,h}^{(t)}$ exists and features $bm_{j,h}^{(t)} == 0$, i.e., connection $L_{j,h}^{(t)}$ has not been selected by any pattern assignment during the last processing epoch, then it is removed.
- PE-based epoch counters are incremented by one as $e_j^{(t+1)} = e_j^{(t)} + 1$, $j = 1, \ldots, c(t)$.
- PE-based best-matching counters $bm_j^{(t)}$, $j = 1, \ldots, c(t)$, are reset to zero.
- All connection-based best-matching counters $bm_{j,h}^{(t)}$, $\forall j \in \{1, c(t)\}$, $\forall h \in \{1, c(t)\}$, $h \neq j$, are reset to zero.

Step 6. Check for Convergence: If PE-based epoch counter $e_j^{(t)} \geq e_{\min}$, $j = 1, \ldots, c(t)$, then stop. Otherwise, goto Step 1.

D. Potential Weaknesses of FOSART

Potential limitations of FOSART are listed below [6].

- Since FOSART employs some heuristic criteria in neighborhood-ranking [see Step 4(a) in Section II-C2], then FOSART does not minimize any known objective function, i.e., its termination is not based on optimizing any model of the process or its data [9].
- FOSART cannot shift codewords through noncontiguous Voronoi regions. This increases the chances of FOSART being trapped in local minima of the distorsion error.
- FOSART is order-dependent due to on-line learning and example-driven generation of reference vectors and lateral connections.
- It combines mini-batch learning techniques, for neuron and synapse removal, with example-driven generation of neurons and synapses, the latter strategy being more sensitive to the presence of noise than mini-batch learning.
- Besides vigilance threshold ρ and minimum epoch number $e_{\min}$, FOSART employs two parameters, er and λ_{ini}, which are user-defined rather than data-driven (note that λ_{fin} is always set equal to a very small positive value, e.g., 0.01, see Section II-C1). These parameters can be considered as significant prior structures, i.e., an important property of the model that must be "hard-wired or built-in, perhaps to be tuned later by experience, but not learned in any statistically meaningful way" [33].

E. Potential Advantages of FOSART

Potential advantages of FOSART are listed below [6].

- Owing to its soft-to-hard competitive implementation, FOSART is expected to be less prone to being trapped in local minima and less likely to generate dead units than hard competitive alternatives [1], [4].
- Owing to its neuron removal strategy, it is robust against noise, i.e., it avoids overfitting.
- Feedback interaction between attentional and orienting subsystems allows FOSART to self-adjust its network size depending on the complexity of the clustering task.
- Owing to its ability to distribute initial reference vectors in the input manifold uniformly, FOSART reduces the risk of dead unit formation and may reduce computation time with respect to traditional random or splitting by two initialization techniques [4], [34].
- FOSART is computationally efficient because its computation time increases linearly as $(c(t)+$ no. of links$(t))$.
- The expressive power of networks that incorporate competition among lateral connections in a constructive framework, like FOSART and GNG, is superior to that of traditional constructive (e.g., see [31], [35] and [36]) or nonconstructive clustering systems (e.g., NG and SOM) which employ no lateral connection explicitly [4]. As a consequence, FOSART, like GNG, features an application domain extended to: 1) vector quantization; 2) entropy maximization (where each reference vector has the same chance of being the winner); and 3) structure detection in input data to be mapped in a topologically correct way onto submaps of an output lattice pursuing dimensionality reduction [4].

TABLE I
HARD-COMPETITIVE FOSART. EPOCHS = 10. THIRTY PRESENTATIONS OF THE IRIS DATA SET

c	$\overline{c}$	$\sigma(c)$	no. ρs	$\overline{\rho}$	$\sigma(\rho)$	ρ_m	ρ_M
3	3.00	0.00	1	0.240	0.0000	0.240	0.240
5	5.00	0.71	2	0.440	0.0102	0.430	0.450
8	8.41	0.96	3	0.518	0.0087	0.510	0.530
12	11.83	0.80	3	0.615	0.0073	0.605	0.623

III. EXPERIMENTAL RESULTS

In this section, FOSART is applied to the Iris and Simpson data sets for comparison with Fuzzy ART and S-Fuzzy ART (see Part I, Section VI). Unlike Fuzzy ART and S-Fuzzy ART, which are hyperbox clustering algorithms not applicable to vector quantization tasks, FOSART is a minimum-distance-to-means clustering network that can be employed in vector quantization, entropy maximization and perceptual grouping (see Section II-E).

FOSART is also compared with another SART algorithm, GART (see Appendix IV in Part I). Unlike FOSART, which pursues soft-to-hard competitive learning to minimize a distorsion error, GART is purely hard competitive to maximize the joint probability of a Gaussian mixture. Moreover, to detect the best-matching unit, GART employs prior probability terms which are ignored in FOSART (in other words, FOSART assumes that cluster types are equiprobable on an *a priori* basis, i.e., before processing the data).

Finally, in vector quantization and perceptual grouping tasks, FOSART is compared with other well-known clustering algorithms found in the literature (e.g., NG, SOM, etc.).

A. FOSART

Iris Data Set: In line with Part I, Section VI, FOSART is input with 30 different sequences of the Iris data set while a majority vote mechanism provides a multiple-to-one class prediction function (multiple-category classification [16]).

Vigilance threshold ρ is adjusted with a trial-and-error procedure until the number of detected clusters in every input sequence equals the desired number of clusters $c = 3, 5, 8,$ and 12, respectively.

Default values of scale parameters λ_{ini} and λ_{fin} are 0.5 and 0.01, respectively, see Section II-C1. To test how λ_{ini} affects FOSART, we employ three values of $\lambda_{ini} = 5, 0.5,$ and 0.01, respectively, corresponding to decreasing intensities of soft competition among processing units, see Section II-C1.

Average output results collected when the number of detected categories equaled the desired number of clusters are presented in Tables I–VI, where, in addition to symbols already employed in Tables II and III of Part I, acronym MSE stands for mean quantization square error.

Tables I and II describe the situation in which FOSART is hard-competitive by setting $\lambda_{ini} = \lambda_{fin} = 0.01$ in (41). In this experiment, hard-competitive FOSART performs better than

TABLE II
HARD-COMPETITIVE FOSART. EPOCHS = 10. THIRTY PRESENTATIONS OF THE IRIS DATA SET

c	$\overline{E}$	$\sigma(E)$	E_m	E_M	$\overline{MSE}$	$\sigma(MSE)$	MSE_m	MSE_M
3	15.833	1.493	12	19	0.536	0.0157	0.526	0.569
5	17.667	3.319	15	24	0.326	0.0163	0.312	0.347
8	13.333	2.988	7	16	0.228	0.0103	0.219	0.249
12	3.348	0.487	3	4	0.169	0.0050	0.163	0.178

TABLE III
FOSART. THIRTY PRESENTATIONS OF THE IRIS DATA SET. $\lambda_{ini} = 0.5$, $\lambda_{fin} = 0.01$. EPOCHS = 10

c	$\overline{c}$	$\sigma(c)$	**no. ρs**	$\overline{\rho}$	$\sigma(\rho)$	ρ_m	ρ_M
3	3.00	0.00	1	0.240	0.0000	0.240	0.240
5	5.00	0.00	1	0.445	0.0000	0.445	0.445
8	7.91	1.18	3	0.500	0.0093	0.490	0.510
12	11.60	1.12	3	0.567	0.0196	0.560	0.624

TABLE IV
FOSART. THIRTY PRESENTATIONS OF THE IRIS DATA SET. $\lambda_{ini} = 0.5$, $\lambda_{fin} = 0.01$. EPOCHS = 10

c	$\overline{E}$	$\sigma(E)$	E_m	E_M	$\overline{MSE}$	$\sigma(MSE)$	MSE_m	MSE_M
3	15.833	1.551	11	17	0.533	0.0090	0.526	0.555
5	14.833	1.239	14	17	0.312	0.0029	0.310	0.317
8	13.625	2.242	8	16	0.229	0.0095	0.217	0.249
12	3.733	0.457	3	4	0.160	0.0019	0.157	0.163

S-Fuzzy ART in terms of efficiency (smaller deviations in performance), while in terms of accuracy it performs significantly better in cases $c = 3$ and $c = 12$ and worse in cases $c = 5$ and $c = 8$. When $c = 3$, the FOSART misclassification error is in line with those of other clustering algorithms found in the literature (see Part I, Section VI).

Tables III and IV describe the situation in which FOSART employs $\lambda_{ini} = 0.5$ (default value) in (41). This parameter setting allows FOSART to improve its performance when $c = 5$, while no improvement is recorded when $c = 8$. Overall, the system seems to gain in stability.

Tables V and VI describe the situation in which FOSART employs $\lambda_{ini} = 5$ in (41). Accuracy improves in cases $c = 5$ and $c = 8$, but greatly worsens when $c = 3$. In all clustering situations, lower accuracy, and stability (larger deviations in performance) are recorded.

Table VII shows the numerical values of the centers of the three Iris classes, while Table VIII provides the mean values of

TABLE V
FOSART. THIRTY PRESENTATIONS OF THE IRIS DATA SET. $\lambda_{ini} = 5$, $\lambda_{fin} = 0.01$. EPOCHS = 10

c	$\overline{c}$	$\sigma(c)$	**no. ρs**	$\overline{\rho}$	$\sigma(\rho)$	ρ_m	ρ_M
3	3.30	0.55	3	0.201	0.0131	0.180	0.220
5	5.02	0.81	3	0.330	0.0309	0.300	0.400
8	7.36	1.25	3	0.459	0.0107	0.440	0.475
12	11.68	1.15	3	0.521	0.0134	0.500	0.540

TABLE VI
FOSART. THIRTY PRESENTATIONS OF THE IRIS DATA SET. $\lambda_{ini} = 5$, $\lambda_{fin} = 0.01$. EPOCHS = 10

c	$\overline{E}$	$\sigma(E)$	E_m	E_M	$\overline{MSE}$	$\sigma(MSE)$	MSE_m	MSE_M
3	25.916	10.512	14	45	0.747	0.1277	0.590	0.950
5	10.388	5.042	6	25	0.346	0.0125	0.336	0.377
8	9.947	2.437	7	14	0.235	0.005	0.230	0.242
12	3.722	0.751	3	5	0.175	0.0081	0.166	0.189

reference vectors detected by hard-competitive FOSART when the number of output clusters is three.

To summarize, our experiments show that in terms of classification rate, an optimal value of λ_{ini} (degree of fuzziness) exists for each data set. If λ_{ini} is too large, i.e., if the degree of fuzziness is too large, then all cluster templates tend to converge (collapse) on the center of gravity (grand mean) of the data set. The comparison of Table III in Part I with Table IV shows that FOSART may be preferable to S-Fuzzy ART in several clustering situations both in terms of stability and accuracy. The advantage of FOSART with respect to S-Fuzzy ART may become relevant by considering the larger domain of applications of FOSART, which includes vector quantization (e.g., for surface reconstruction), entropy maximization and perceptual grouping (see further on this section).

Simpson Data Set: In line with Part I, Section VI, FOSART is input with six different presentations of the Simpson data set. When user parameters are $\rho = 0.021$, $\lambda_{ini} = 0.5$ (default value) and $\lambda_{fin} = 0.01$ (default value), FOSART detects three clusters. The corresponding confusion matrix is shown in Table IX, where $\overline{MSE} = 12.70$, $\sigma(MSE) = 1.07$. In this case FOSART is insensitive to the order of the input sequence, i.e., its robustness to changes in the order of presentation of the input sequence is superior to that featured by Fuzzy ART and S-Fuzzy ART (see Tables IV and V in Part I).

When the number of detected clusters is five ($\rho = 0.1$, $\lambda_{ini} = 0.5$, $\lambda_{fin} = 0.01$), the average confusion matrix reporting point allocations and, in parentheses, standard deviation per cell is equivalent to Table VI in Part I, where the number of misclassification points is equal to two. In this case, FOSART accuracy is inferior to that featured by S-Fuzzy ART (see Table VII in Part I).

TABLE VII
CENTERS OF THE THREE IRIS CLASSES

	Band 1	Band 2	Band 3	Band 4
Sup. Label 1	5.006	3.428	1.462	0.246
Sup. Label 2	5.936	2.770	4.260	1.326
Sup. Label 3	6.588	2.974	5.552	2.026

TABLE VIII
HARD-COMPETITIVE FOSART. EPOCHS = 10. THIRTY PRESENTATIONS OF THE IRIS DATA SET. NO. OF CLUSTERS = 3. MEAN VALUES OF THE REFERENCE VECTORS

	Band 1	Band 2	Band 3	Band 4
Sup. Label 1	5.0061	3.4231	1.4701	0.2500
Sup. Label 2	5.9078	2.7510	4.4018	1.4319
Sup. Label 3	6.8388	3.0686	5.7230	2.0667

TABLE IX
FOSART. $\rho = 0.021$, $\lambda_{ini} = 0.5$ (DEFAULT VALUE) AND $\lambda_{fin} = 0.01$ (DEFAULT VALUE). EPOCHS = 10. SIX PRESENTATIONS OF THE SIMPSON DATA SET. NO. OF CLUSTERS = 3. $\overline{MSE} = 12.70$, $\sigma(MSE) = 1.07$. AVERAGE CONFUSION MATRIX REPORTING POINT ALLOCATIONS AND, IN PARENTHESES, STANDARD DEVIATION PER CELL

	Cluster 1	Cluster 2	Cluster 3
Sup. Label 1	7 (0)	0 (0)	0 (0)
Sup. Label 2	0 (0)	8 (0)	0 (0)
Sup. Label 3	0 (0)	1 (0)	0 (0)
Sup. Label 4	0 (0)	0 (0)	2 (0)
Sup. Label 5	0 (0)	0 (0)	6 (0)

Perceptual Grouping: Let us apply FOSART to a perceptual grouping problem in vision. In other words, FOSART is employed to extract the global impression of an image, i.e., to detect the "right" partition of an image into subsets [17]. Fig. 2 shows projections onto input space of lateral connections detected by FOSART when the nonconvex and concentric data set shown in Fig. 1 is processed, where receptive field centers are depicted as circles. Note that the inner concentric structure is clustered by a single and isolated reference vector. This behavior is made possible by CCHR, while it would be impossible with traditional CHR. Due to the subjective nature of clustering problems, it is obviously true that results shown in Fig. 2 depend on parameter tweaking, i.e., these results have no absolute relevance or validity. Nonetheless, we consider these results useful to illustrate the potential (subjective), but peculiar ability of FOSART in clustering nonconvex data sets.

Another interesting perceptual grouping application regards the two-spirals data set shown in Fig. 3 [19]. This noiseless data set consists of 194 patterns belonging to two concentric spirals such that in the outer parts of the spirals data points from one spiral are farther apart from each other than from points of the inner spiral [19]. The hypothetical task is to construct a two-stage classifier, employing FOSART as its first stage, which is able to distinguish between the two spirals. This task appears to be rather difficult for typical multilayer perceptrons trained with backpropagation (BP) (see Table X, adapted from [19]). Fig. 4 (to be compared with [19, Fig. 21]) shows the projection onto input space of the output graph generated by FOSART when the two spirals data set is processed with input parameters: $\rho = 0.88$ and $e_{\min} = 1$; output parameters are: number of nodes = 148, $MSE = 0.019$, number of maps = 16. In Fig. 4, receptive field centers are depicted as white squares, lines represent projections of lateral connections and black circles indicate input patterns. The overall result is consistent with human perception of structures shown in Fig. 3, which makes this result interesting and peculiar in the panorama of existing clustering techniques, despite its (obvious) dependence on parameter tweaking. In this example, FOSART guarantees an important saving in computation time with respect to the growing cell structure (GCS) algorithm [19], see Table X. It is interesting to note that GCS and its evolution, GNG [4], both employ a mini-batch output unit generation criterion which: 1) averages over the noise on the data and 2) locates initial templates within the convex hull of the input data. On the one hand, GCS and GNG may require more training epochs than FOSART to reach termination when a nonconvex input data set is processed, since initial templates of GCS and GNG may lie outside a nonconvex input manifold (see simulations at http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VIDM/research/gsn/DemoGNG/GNG.html). On the other hand, FOSART locates initial templates on an example-driven basis, i.e., FOSART is more sensitive than GCS and GNG to the presence of noise (see Section II-D) [4], [37].

Surface Reconstruction: The fifth data set employed to test FOSART properties is 3-D and consists of 9371 vectors representing a digitized human face [38]. This data set is requantized by FOSART in comparison with the NG algorithm employing Hyperboxes for efficient initialization of reference vectors [39]. Fig. 5 shows the 3-D digitized human face employed as input sequence, while Fig. 6 depicts the digitized face resampled by FOSART which does not require any preprocessing. Input parameters are: $\rho = 0.39$, $e_{\min} = 14$. Output information is: no. of nodes = 1745, no. of maps = 19, $MSE = 2.98$, number of epochs = 15. Tables XI and XII provide output statistics of FOSART and NG in this application setting. These tables show that for any fixed number of epochs FOSART performs better than NG with Hyperbox in terms of MSE minimization, i.e., FOSART trains faster than NG with Hyperbox in this clustering case. Owing to its ability to localize initial reference vectors which lie on the input manifold, FOSART requires no input data preprocessing and fewer training epochs than NG with Hyperbox to converge. Moreover, FOSART is not affected by the presence of dead units. Finally, projections onto input space of

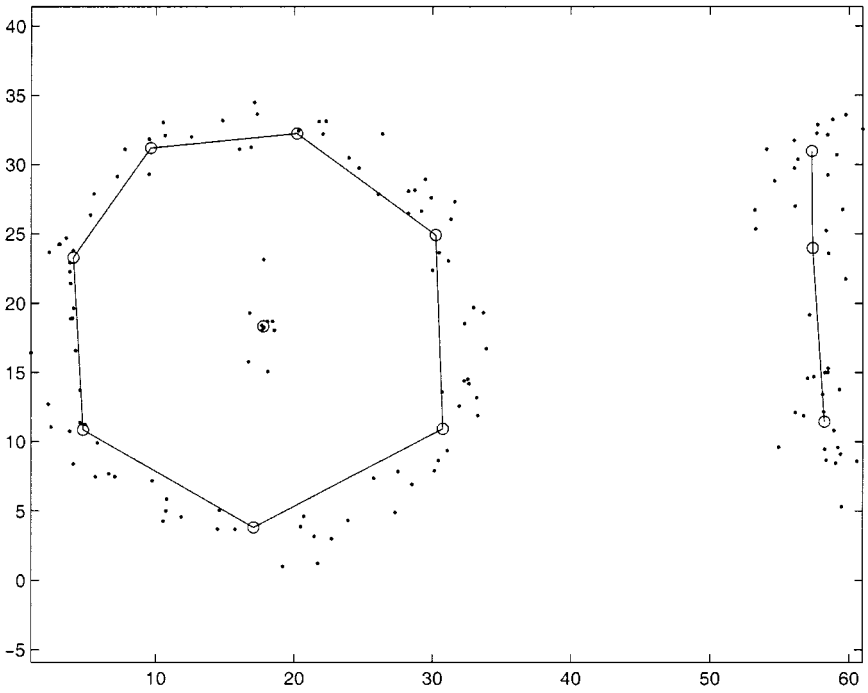

Fig. 2. FOSART processing of the nonconvex data set. Output information is: 11 templates, three maps, training epochs = 2.

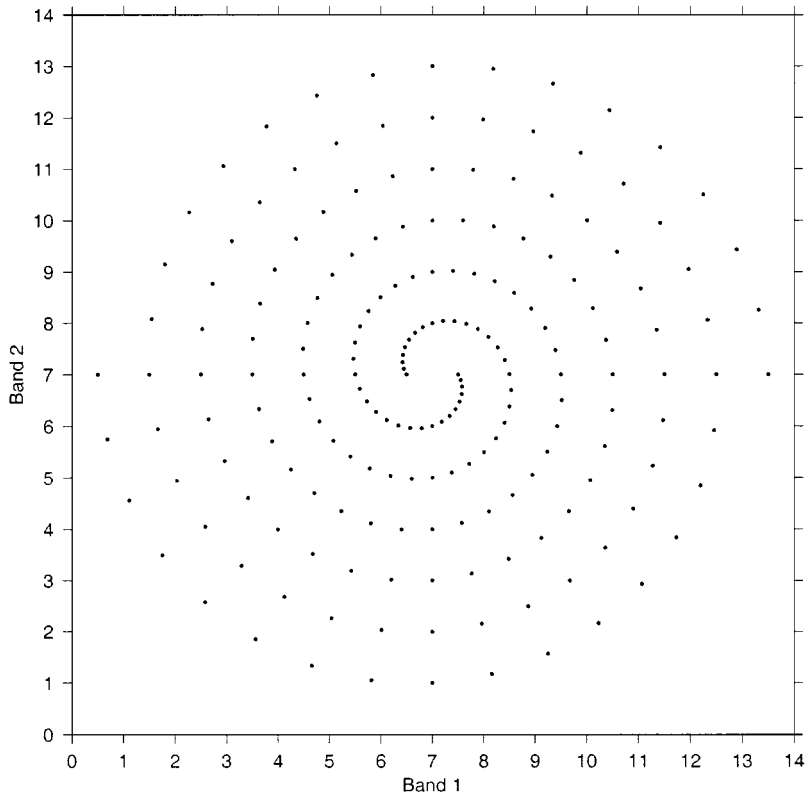

Fig. 3. Two-spiral data set consisting of 194 data points.

lateral connections detected by FOSART can be directly employed for surface reconstruction, as shown in Fig. 7.

Vector Quantization: In vector quantization, FOSART may be employed as the pre-processing module of the enhanced Linde–Buzo–Gray (ELBG) clustering algorithm [40], [41]. This network combination may be interesting owing to the complementary functional features of FOSART and ELBG. On the one hand, FOSART is on-line learning, constructive and cannot shift codewords through noncontiguous Voronoi regions. On the other hand, ELBG is nonconstructive, batch learning and capable of moving codewords through contiguous as well as noncontiguous Voronoi regions to reduce quanti-

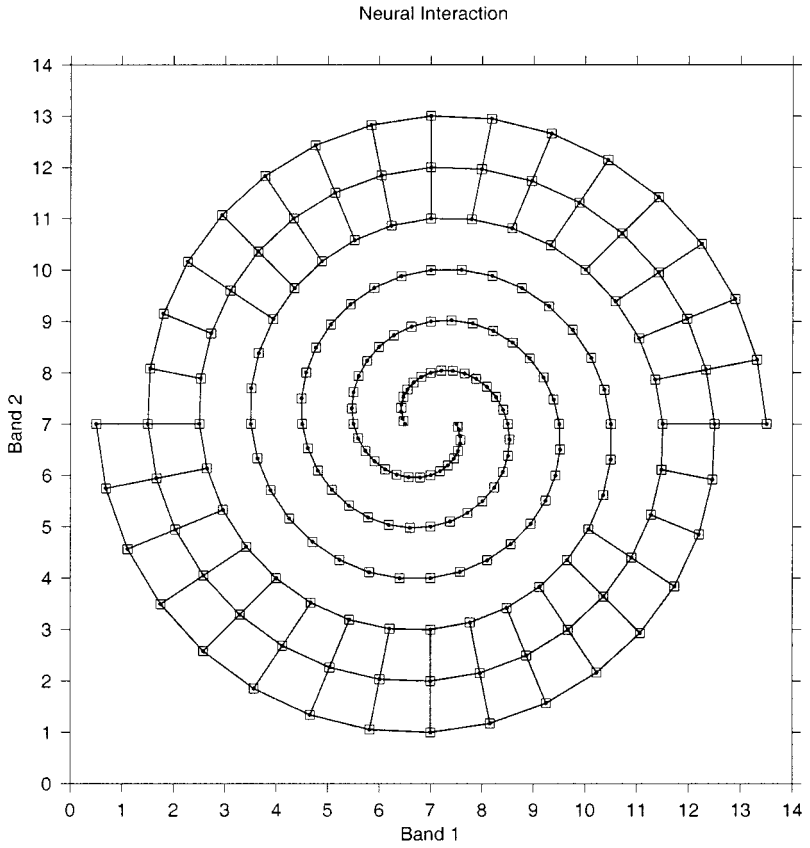

Fig. 4. FOSART processing of the two-spiral data set. Output information is: 148 templates, 16 maps, training epochs = 2 (to be compared with [19, Fig. 21]).

TABLE X
TRAINING EPOCHS REQUIRED TO SOLVE THE TWO-SPIRAL PROBLEM (ADAPTED FROM [19])

Network model	Epochs
BP	20000
Cross entropy BP	10000
Cascade-correlation	1700
Growing Cell Structure (GCS)	180
FOSART	2

zation error (27) [40], [41]. To the best of our knowledge, this latter feature makes ELBG, together with the LBG-utility (LBG-U) algorithm proposed in [42], quite unique in the panorama of clustering algorithms found in the literature.

In ELBG, templates eligible for shifting and splitting are those whose "local" contribution to the MSE value is, respectively, below and above the mean distorsion. Templates eligible for shifting are selected sequentially and those eligible for splitting are selected stochastically (in a way similar to the roulette wheel selection in genetic algorithms). Each selected pair of templates is adjusted locally based on the traditional LBG (i.e., c-means) batch clustering algorithm [43]. A backtracking mechanism allows ELBG to recover from inconvenient shift attempts. Thus, one main difference between ELBG and LBG-U is that the former algorithm allows several shifts per iteration, while LBG-U allows only one. The capability of moving codewords through contiguous as well as noncontiguous Voronoi regions allows ELBG to perform better than LBG and the modified LBG (M-LBG) version proposed in [44]. Owing to its efficient implementation of local LBG adjustments [41], the ELBG increase in computational complexity with respect to LBG remains negligible (below 8%) [40]. In [40], ELBG is initialized either randomly or with the splitting-by-two technique proposed in [43]. It has also been shown that the ELBG distorsion value does not depend on initial conditions, but ELBG convergence time does [40].

In line with [40], an image compression task is considered, where the 8-bit Lena image, consisting of 512×512 pixels, is divided into blocks of 4×4 pixels to generate 16 384 vectors in a 16-dimensional data space. The M-LBG algorithm [44], ELBG with initialization by splitting-by-two and ELBG with initialization by FOSART are compared. Output results are shown in Table XIII, where the peak signal to noise ratio (PSNR) is computed as

$$PSNR = 10 \log_{10} \frac{d \cdot (255)^2}{MSE}$$

where d is the dimensionality of input space (in this Lena example, $d = 16$).

In this experiment, Table XIII shows that: 1) ELBG is robust with respect to changes in initial conditions (compare PSNR values); 2) ELBG is always more accurate than M-LBG; 3)

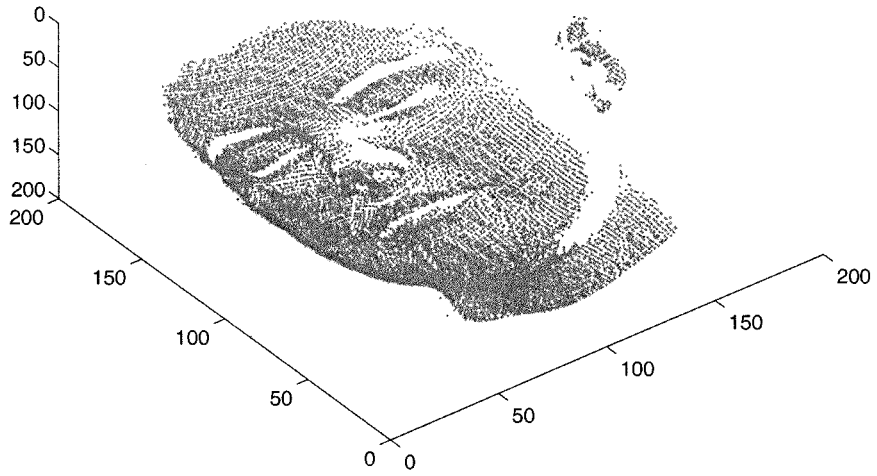

Fig. 5. 3-D digitized human face consisting of 9371 data points.

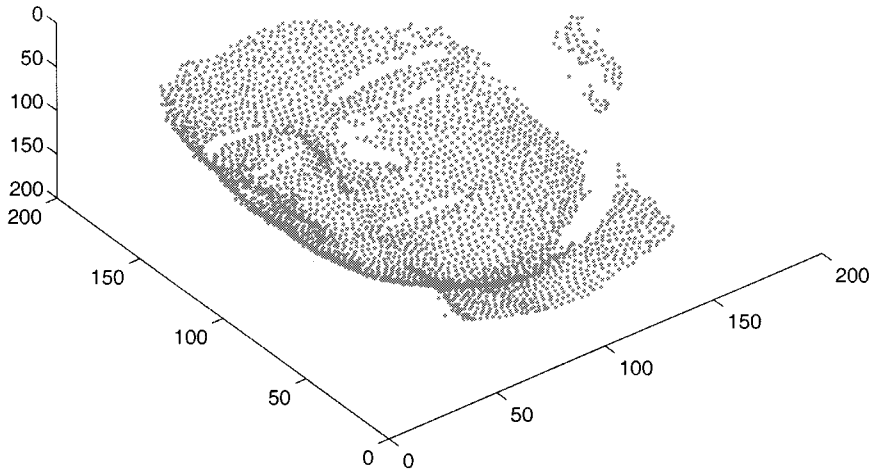

Fig. 6. Reference vectors detected by FOSART when the digitized human face data set is processed. Output information is: 1745 templates, 19 maps, mean square error (MSE) $= 2.98$, training epochs $= 5$.

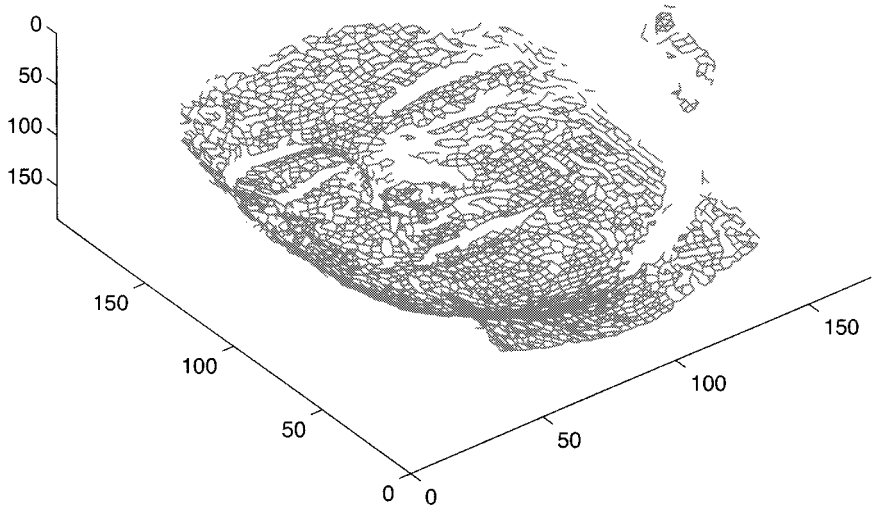

Fig. 7. Projection onto input space of lateral connections detected by FOSART in the output map of the digitized human face data set.

ELBG benefits from being initialized with FOSART, both in terms of accuracy and computation time; and 4) ELBG initialized with FOSART is faster to train than M-LBG.

Convergence in Training: Several works in the literature show that with its ability to distribute initial reference vectors in the input manifold uniformly, FOSART may be faster to

TABLE XI
OUTPUT STATISTICS OF FOSART WHEN APPLIED TO THE 3-D DIGITIZED HUMAN FACE DATA SET

Epoch	PEs	Connections	Maps	MSE
1	1743	3102	20	3.40
5	1744	3558	19	3.08
10	1745	3521	19	2.98
15	1745	3519	19	2.98

TABLE XII
OUTPUT STATISTICS OF NG IMPROVED WITH HYPERBOX PREPROCESSING WHEN APPLIED TO THE 3-D DIGITIZED HUMAN FACE DATA SET

Epoch	PEs	Dead units	MSE
1	1743	35	6.54
5	1743	11	3.45
10	1743	7	3.11
15	1743	3	2.99

train than other vector quantizers which employ random initialization of templates, such as NG (see above in this section), SOM and the fuzzy learning vector quantization (FLVQ) algorithm [45]. For example, MSE values of the training phase of FOSART, FLVQ and SOM in an ERS-1 SAR image clustering task are shown in Fig. 8 [46]. More examples of this kind can be found in [47], [48].

B. GART

To the best of our knowledge, Gaussian ART (GART), which is sketchily described in [49] and further discussed in Appendix IV of Part I, has never been employed and investigated as a standalone module in the existing literature. Rather, it is employed in the ARTMAP classification framework where the Gaussian ARTMAP (GAM) supervised network is proposed in both hard- and soft-competitive versions (the latter being more successful) [49], [50]. In GART [49], the number of categories created during training is a function of a two-dimensional parameter space consisting of parameters ρ and γ, see Appendix IV in Part I, whereas FOSART employs parameter ρ exclusively. In other words, for a desired number of output clusters and a given data set, the global minimum of the GART error function is found with an optimal pair of γ and ρ values to be detected by the user with a trial-and-error procedure. This implies that GART is more difficult to use than FOSART in practical applications. For example, there are values of ρ (e.g., 0.01) for which no γ value is found to allow GART to detect five output categories in the Simpson data set. GART is applied to the Iris and Simpson data sets to obtain a comparison with Fuzzy ART, S-Fuzzy ART and FOSART.

Iris Data Set: GART is evaluated under two different regimes of parameters γ and ρ. In particular, ρ is set to 0.01 and 0.1 while γ is increased until the desired number of output clusters (3 or 5) is detected. Best, worst, and average classification performances of GART are shown in Tables XIV and XV, where statistics relating to the negative log-likelihood (NLL) cost value are included. When compared with Tables I–IV, Tables XIV and XV reveal that GART is less stable and effective than FOSART in minimizing MSE, in line with theoretical expectations, and that the misclassification error E of the classifier consisting of GART combined with a majority vote mechanism is worse than that obtained with FOSART. Besides its worse accuracy, the main deficiency of GART is to feature two internal parameters that affect the number of categories created during training, i.e., GART is twice as difficult to use as FOSART.

Simpson Data Set: When compared with Table VI in Part I and Table IX in Part II, which both hold in the case of FOSART, Tables XVI and XVII reveal that GART is less stable and effective than FOSART in clustering the Simpson data set consistently with perceptual grouping mechanisms. In line with the assessment of GART in the Iris data clustering case, these results confirm that minimization of the negative log-likelihood (NLL) seems less useful than minimization of MSE in supervised and unsupervised learning problems, such as data classification, perceptual grouping and vector quantization, other than probability density function estimation. Moreover, GART is more difficult to use than FOSART.

IV. CONCLUSION

In Part I of this paper, two algorithms, S-Fuzzy ART and GART, the latter taken from the literature, are presented as two instances of the SART group of ART clustering framework.

In Part II of this paper, another instance of class SART, termed FOSART, is proposed to take advantage of the combination of the SART optimization framework with useful properties driven by successful clustering algorithms such as NG, SOM, and GNG. FOSART is a constructive, on-line learning, topology-preserving, soft-to-hard competitive, minimum-distance-to-means SART clustering network whose aim is to minimize a quantization error. FOSART features several peculiar properties when compared to existing clustering algorithms:

1) unlike GNG and SOM, FOSART tries to minimize a quantization (sum-of-squares) error via a soft-to-hard competitive model transition.
2) Unlike Fuzzy ART, the system requires no complement coding of the input data.
3) Unlike SOM and NG, FOSART requires no randomization of the initial template vectors.
4) Unlike SOM and NG, the system requires no *a priori* knowledge of the size of the network.
5) Unlike SOM, the system requires no *a priori* knowledge of the topology of the network.
6) Unlike SOM, NG, and Fuzzy ART, FOSART explicitly deals with lateral connections.
7) Unlike GNG, FOSART attempts to address all constraints required to make the CHR guarantee perfect

TABLE XIII
COMPARISON OF M-LBG AND ELBG IN THE CLUSTERING OF THE 16-DIMENSIONAL LENA DATA SET, CONSISTING OF 16 384 VECTORS. $*$: RESULTS TAKEN FROM THE LITERATURE

c	M-LBG			ELBG with splitting-by-two			ELBG with FOSART		
	PSNR* (db)	MSE	Iter.*	PSNR (db)	MSE	Iter. (split.+ELBG)	PSNR (db)	MSE	Iter. (FOSART+ELBG)
256	31.92	668.6	20	31.97	660.9	46 + 8	31.98	659.4	3 + 10
512	33.09	510.7	17	33.17	499.2	54 + 8	33.22	494.0	3 + 9
1024	34.42	376.0	19	34.72	349.3	64 + 9	34.78	344.3	3 + 9

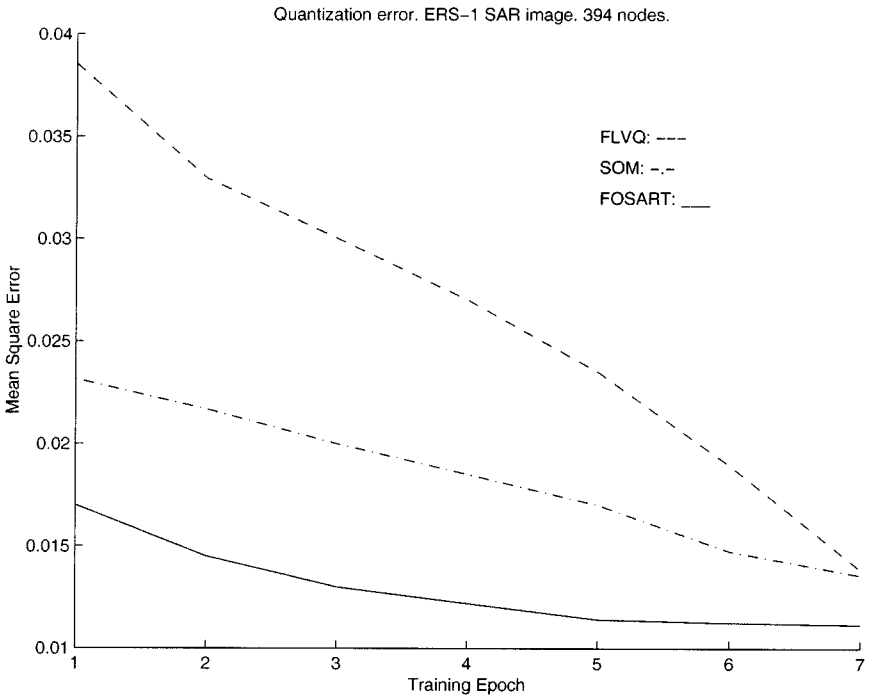

Fig. 8. MSE as a function of the number of training epochs for FOSART, FLVQ, and SOM in the clustering of an ERS-1 SAR image, 512 × 512 pixel size, 394 clusters (taken from the literature).

TABLE XIV
GART. THIRTY PRESENTATIONS OF THE IRIS DATA SET. EPOCHS = 10

c	$\bar{c}$	$\sigma(c)$	(ρ, γ)	$\overline{NLL}$	$\sigma(NLL)$	NLL_m	NLL_M
3	3.23	0.60	(0.01, 6), (0.1, 13)	232.96	56.02	186.70	311.36
5	4.74	0.60	(0.01, 3), (0.1, 9)	159.90	14.28	149.35	187.07
8	8.47	1.27	(0.01, 0.85), (0.1, 3.5)	141.34	3.25	136.67	146.75
12	12.23	0.83	(0.01, 0.4), (0.1, 2)	139.83	1.75	138.24	145.88

topology-preserving mapping in the sense proposed in [8].

8) Unlike parameters of SOM and NG, FOSART parameters are not affected by outliers which are instead mapped onto noise categories.
9) Unlike Fuzzy ART, the system is capable of removing noise categories to avoid overfitting.
10) Unlike Fuzzy ART, FOSART is competitive with other clustering models found in the literature when the Iris data set is clustered with three reference vectors [51].

TABLE XV
GART. THIRTY PRESENTATIONS OF THE IRIS DATA SET. EPOCHS = 10

c	$\overline{E}$	$\sigma(E)$	E_m	E_M	$\overline{MSE}$	$\sigma(MSE)$	MSE_m	MSE_M
3	17.952	5.978	10	40	0.959	0.2156	0.740	1.350
5	21.503	4.032	16	28	0.447	0.0693	0.390	0.600
8	13.571	5.543	8	22	0.277	0.0240	0.250	0.330
12	6.611	2.200	4	12	0.196	0.0141	0.180	0.220

TABLE XVI
GART. SIX PRESENTATIONS OF THE SIMPSON DATA SET. $(\rho, \gamma) = (0.01, 3.5)$ AND $(\rho, \gamma) = (0.1, 12)$. EPOCHS = 10. NO. OF CLUSTERS = 3. $\overline{MSE} = 17.45$, $\sigma(MSE) = 5.09$, $\overline{NLL} = 58.96$, $\sigma(NLL) = 1.89$. AVERAGE CONFUSION MATRIX REPORTING POINT ALLOCATIONS AND, IN PARENTHESES, STANDARD DEVIATION PER CELL

	Cluster 1	**Cluster 2**	**Cluster 3**
Sup. Label 1	7 (0)	0 (0)	0 (0)
Sup. Label 2	4.8 (3.9)	3.1 (3.9)	0 (0)
Sup. Label 3	0 (0)	1 (0)	0 (0)
Sup. Label 4	0 (0)	0 (0)	2 (0)
Sup. Label 5	0 (0)	0 (0)	6 (0)

TABLE XVII
GART. SIX PRESENTATIONS OF THE SIMPSON DATA SET. $(\rho, \gamma) = (0.1, 0.27)$ AND $(\rho, \gamma) = (0.3, 15)$. EPOCHS = 10. NO. OF CLUSTERS = 5. $\overline{MSE} = 7.72$, $\sigma(MSE) = 1.21$, $\overline{NLL} = 54.66$, $\sigma(NLL) = 2.51$. AVERAGE CONFUSION MATRIX REPORTING POINT ALLOCATIONS AND, IN PARENTHESES, STANDARD DEVIATION PER CELL

	Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**
Sup. Label 1	6 (0)	0 (0)	0 (0)	0 (0)	1 (0)
Sup. Label 2	0 (0)	6.6 (0.5)	0 (0)	0 (0)	1.3 (0.5)
Sup. Label 3	0 (0)	0 (0)	1 (0)	0 (0)	0 (0)
Sup. Label 4	0 (0)	0 (0)	0.6 (1.0)	1.3 (1.0)	0 (0)
Sup. Label 5	0 (0)	0 (0)	0 (0)	6 (0)	0 (0)

FOSART performances are assessed in a wide range of supervised and unsupervised learning tasks, such as data classification, vector quantization, and perceptual grouping.

To compare FOSART with Fuzzy ART, S-Fuzzy ART, and GART, the Iris and Simpson data sets are chosen from the literature. In these comparisons, FOSART tends to be more accurate and stable to changes in the order of presentation of the input sequence than the other algorithms. Moreover, FOSART features an applicability domain broader than that of Fuzzy ART, S-Fuzzy ART, and GART, i.e., FOSART expressive power is superior to that of these algorithms.

Results with the Iris data set also reveal that FOSART is competitive with other clustering algorithms found in the literature when the number of detected clusters is three.

In comparison with some well-known clustering networks like NG, SOM, and FLVQ, FOSART trains faster while it remains competitive in terms of quantization error minimization.

In the field of neural networks for vector quantization, an interesting development of FOSART is the combination of FOSART with the ELBG batch clustering algorithm, which is capable of moving codewords through noncontiguous Voronoi regions [40], [41].

As a future development in the field of neural networks for classification we plan to combine FOSART with the "match tracking" mechanism employed in the Gaussian ARTMAP (GAM) classifier [50]. Match tracking involves raising vigilance threshold ρ, whose initial (baseline) value is low (e.g., 10^{-7}), when an incorrect prediction is made. Since match tracking adapts vigilance threshold ρ automatically, where ρ is the only FOSART internal parameter which affects the number of categories created during training, this constructive classification scheme would require no user-defined parameter in adapting the network size to problem complexity until no incorrect prediction is made during training (thus, this classifier would belong to the class of consistent classifiers [16], [37]).

ACKNOWLEDGMENT

The authors are grateful to R. Savarè, G. Ferrigno, N. A. Borghese, and S. Ferrari for providing the 3-D digitized human face data set and N. A. Borghese for preparing Table XII. A. Baraldi thanks F. Parmiggiani for his support. Both authors wish to thank Prof. J. Feldman for the stimulating environment at the International Computer Science Institute (ICSI), Berkeley, CA, where this work was started and where E. Alpaydın was a Fulbright Scholar. They are also grateful to the anonymous referees for their thoughtful comments about this paper.

REFERENCES

[1] T. Martinetz, G. Berkovich, and K. Schulten, "Neural-Gas network for quantization and its application to time-series predictions," *IEEE Trans. Neural Networks*, vol. 4, pp. 558–569, 1993.
[2] T. Kohonen, *Self-Organizing Maps*. Berlin, Germany: Springer-Verlag, 1995.
[3] ——, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.
[4] B. Fritzke. (1997) Some competitive learning methods. [Online]. Available: Draft document, http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/ research/gsn/DemoGNG.
[5] ——, "A growing neural gas network learns topologies," in *Advances in Neural Information Processing Systems 7*, G. Tesauro, D. S. Touretzky, and T. K. Leen, Eds. Cambridge, MA: MIT Press, 1995, pp. 625–632.
[6] A. Baraldi and P. Blonda, "A survey on fuzzy neural networks for pattern recognition: Part I," *IEEE Trans. Syst., Man, Cybern. B*, vol. 29, pp. 778–785, Dec. 1999.
[7] ——, "A survey on fuzzy neural networks for pattern recognition: Part II," *IEEE Trans. Syst., Man, Cybern. B*, vol. 29, pp. 786–801, Dec. 1999.
[8] T. Martinetz, G. Berkovich, and K. Schulten, "Topology representing networks," *Neural Networks*, vol. 7, no. 3, pp. 507–522, 1994.
[9] E. C. Tsao, J. C. Bezdek, and N. R. Pal, "Fuzzy Kohonen clustering network," *Pattern Recognition*, vol. 27, no. 5, pp. 757–764, 1994.
[10] E. Erwin, K. Obermayer, and K. Schulten, "Self-organizing maps: Ordering, convergence properties and energy functions," *Biol. Cybern.*, vol. 67, pp. 47–55, 1992.
[11] C. Bishop, M. Svensen, and C. Williams, "GTM: A principled alternative to the self-organizing map," in *Proc. Int. Conf. Artificial Neural Networks, ICANN'96*. New York: Springer-Verlag, 1996, pp. 164–170.
[12] S. P. Luttrell, "A Bayesian analysis of self-organizing maps," *Neural Comput.*, vol. 6, pp. 767–794, 1994.

[13] H. Kumazava, M. Kasahara, and T. Namekawa, "A construction of vector quantisers for noisy channels," *Elect. Eng. Jpn.*, vol. 67B, pp. 39–47, 1984.
[14] T. Hofmann and J. M. Buhmann, "Competitive learning algorithms for robust vector quantization," *IEEE Trans. Signal Processing*, vol. 46, pp. 1665–1675, 1998.
[15] T. M. Lillesand and R. W. Kiefer, *Remote Sensing and Image Interpretation*. New York: Wiley, 1979.
[16] J. C. Bezdek, T. Reichherzer, G. S. Lim, and Y. Attikiouzel, "Multiple-prototype classifier design," *IEEE Trans. Syst., Man, Cybern. C*, vol. 28, pp. 67–79, 1998.
[17] J. Shi and J. Malik, "Normalized cuts and image segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, Puerto Rico, June 1997.
[18] P. Perona and W. Freeman, "A factorization approach to grouping," in *Proc. ECCV '98*, vol. 1, 1998, pp. 655–670.
[19] B. Fritzke, "Growing cell structures—A self-organizing network for unsupervised and supervised learning," *Neural Networks*, vol. 7, no. 9, pp. 1441–1460, 1994.
[20] ——, "Incremental neuro-fuzzy systems," in *Proc. SPIE Opt. Sci., Eng. Instrumentation '97: Applicat. Fuzzy Logic Technol. IV*, San Diego, CA, 1997.
[21] V. Cherkassky and F. Mulier, *Learning From Data: Concepts, Theory, and Methods*. New York: Wiley, 1998.
[22] S. M. Omohundro, "The Delaunay triangulation and function learning," Int. Comput. Sci. Inst., Berkeley, CA, Tech. Rep. TR-90-001, 1990.
[23] J. R. Shewchuck, "Delaunay refinement mesh generation," Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-97-137, 1997.
[24] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.
[25] C. G. Atkeson, S. A. Schall, and A. W. Moore, "Locally weighted learning," *AI Rev.*, vol. 11, pp. 11–73, 1997.
[26] F. Ancona, S. Ridella, S. Rovetta, and R. Zunino, "On the importance of sorting in 'Neural Gas' training of vector quantizers," in *Proc. Int. Conf. Neural Networks '97*, vol. 3, Houston, TX, 1997, pp. 1804–1808.
[27] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[28] J. C. Bezdek and N. R. Pal, "Generalized clustering networks and Kohonen's self-organizing scheme," *IEEE Trans. Neural Networks*, vol. 4, pp. 549–557, 1993.
[29] L. Bottou, "Online learning and stochastic approximations," in *Online Learning in Neural Networks*, D. Saad, Ed. Cambridge, U.K.: Cambridge Univ. Press, 1998.
[30] J. Buhmann, "Learning and data clustering," in *Handbook of Brain Theory and Neural Networks*, M. Arbib, Ed. Cambridge, MA: Bradford Books/MIT Press, 1995.
[31] S. Ridella, S. Rovetta, and R. Zunino, "Plastic algorithm for adaptive vector quantization," *Neural Comput. Applicat.*, vol. 7, pp. 37–51, 1998.
[32] E. R. Kandel, *Principles of Neural Science*, E. Kandel and J. Schwartz, Eds. Norwalk, CT: Appleton and Lange, 1991.
[33] S. Geman, E. Bienenstock, and R. Dourstat, "Neural networks and the bias/variance dilemma," *Neural Comput.*, vol. 4, no. 1, pp. 1–58, 1992.
[34] J. Moody and C. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Comput.*, vol. 1, pp. 281–294, 1989.
[35] H. Frigui and R. Krishnapuram, "A robust competitive clustering algorithm with applications in computer vision," *IEEE Trans. Neural Networks*, vol. 10, pp. 450–465, Mar. 1999.
[36] N. Karayiannis, "Growing radial basis neural networks: Merging supervised and unsupervised learning with network growth techniques," *IEEE Trans. Neural Networks*, vol. 8, pp. 1492–1506, Nov. 1997.
[37] T. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.
[38] N. A. Borghese, G. Ferrigno, G. Baroni, R. Savarè, S. Ferrari, and A. Pedotti, "AUTOSCAN: A flexible and portable scanner of 3D surfaces," *IEEE Comput. Graphics I/O Devices*, pp. 1–5, 1998.
[39] M. Fontana, N. A. Borghese, and S. Ferrari, "Image reconstruction using improved neural-gas," in *Proc. Workshop Italiano Reti Neurali '95*, M. Marinaro and R. Tagliaferri, Eds, Singapore: World Scientific, 1995, pp. 260–265.
[40] G. Patanè and M. Russo, "The enhanced-LBG algorithm," *Neural Networks*, vol. 14, no. 9, pp. 1219–1237, 2001.
[41] ——, "ELBG implementation," *Int. J. Knowledge-Based Intell. Eng. Syst.*, vol. 4, pp. 94–109, Apr. 2000.
[42] B. Fritzke, "The LBG-U method for vector quantization—An improvement over LBG inspired from neural networks," *Neural Processing Lett.*, vol. 5, no. 1, 1997.
[43] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, pp. 84–94, Jan. 1980.
[44] D. Lee, S. Baek, and K. Sung, "Modified k-means algorithm for vector quantizer design," *IEEE Signal Processing Lett.*, vol. 4, pp. 2–4, Jan. 1997.
[45] J. C. Bezdek and N. R. Pal, "Two soft relatives of learning vector quantization," *Neural Networks*, vol. 8, no. 5, pp. 729–743, 1995.
[46] P. Blonda, G. Satalino, J. Wasowskj, M. Parise, and A. Baraldi, "Neural techniques for SAR intensity and coherence data classification," in *Proc. EOS-SPIE: Image and Signal Processing for Remote Sensing V*, vol. 3871, Florence, Italy, Sept. 1999, pp. 374–381.
[47] P. Blonda, G. Satalino, A. Baraldi, and R. De Blasi, "Segmentation of multiple sclerosis lesions in MRI by fuzzy neural networks: FLVQ and FOSART," in *Proc. NAFIPS '98*, Pensacola Beach, FL, August 1998, pp. 39–43.
[48] P. Blonda, A. Baraldi, G. Bafunno, G. Satalino, and G. Ria, "Experimental comparison of FOSART and FLVQ in a remotely sensed image classification task," in *Proc. SPIE Opt. Sci., Eng. Instrumentation '97: Applications of Fuzzy Logic Technology IV*, vol. 3165, San Diego, CA, July 1997, (invited), pp. 113–122.
[49] J. R. Williamson, "Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps," *Neural Networks*, vol. 9, no. 5, pp. 881–897, 1996.
[50] ——, "A constructive, incremental-learning network for mixture modeling and classification," *Neural Comput.*, vol. 9, pp. 1517–1543, 1997.
[51] A. Baraldi and E. Alpaydın, "Simplified ART: A new class of ART algorithms," Int. Comput. Sci. Inst., Berkeley, CA, TR-98-004.
[52] A. Baraldi and L. Shenato, "Soft-to-hard model transition in clustering: A review," Int. Comput. Sci. Inst., Berkeley, CA, TR-99–010, 1999.

Andrea Baraldi was born in Modena, Italy, in 1963. He graduated in electronic engineering from the University of Bologna, Bologna, Italy, in 1989. His Master's thesis focused on the development of unsupervised clustering algorithms for optical satellite imagery.

From 1989 to 1990, he was a Research Associate at CIOC-CNR, an Institute of the National Research Council (CNR) in Bologna, and served in the army at the Istituto Geografico Militare in Florence, working on satellite image classifiers and GIS. As a consultant at ESA-ESRIN, Frascati, Italy, he worked on object-oriented applications for GIS from 1991 to 1993. From December 1997 to June 1999, he joined the International Computer Science Institute, Berkeley, CA, with a postdoctoral fellowship in Artificial Intelligence. Since his master thesis, he has continued his collaboration with ISAO-CNR, Bologna. As a postdoctoral researcher, he currently works at the European Commission Joint Research Centre, Ispra, Italy, in the development and validation of classification algorithms applied to wide area radar mosaics of forest ecosystems. His main interests center on image segmentation and classification, with special emphasis on texture analysis and neural-network applications in computer vision.

Ethem Alpaydın was born on June 23, 1966. He received the B.Sc. degree in 1987 from Boğaziçi University, Istanbul, Turkey, and the Ph.D. degree in 1990 from Ecole Polytechnique Fédérale, Lausanne, Switzerland.

In 1991, he was a Postdoctoral Researcher with the International Computer Science Institute (ICSI), Berkeley, CA. Since October 1991, he has been teaching with the Department of Computer Engineering, Boğaziçi University, where he is now Associate Professor. He held visiting research positions at Massachusetts Institute of Technology, Cambridge, in 1994, ICSI (as a Fulbright scholar) in 1997, and IDIAP in 1998. His research interests are artificial neural networks and machine learning.