Model Transitions in Descending FLVQ

Andrea Baraldi, Palma Blonda, *Member, IEEE,* Flavio Parmiggiani, Giuseppe Pasquariello, and Guido Satalino

***Abstract*— Fuzzy learning vector quantization (FLVQ), also known as the fuzzy Kohonen clustering network, was developed to improve performance and usability of on-line hard-competitive Kohonen's vector quantization and soft-competitive self organizing map (SOM) algorithms. The FLVQ effectiveness seems to depend on the range of change of the weighting exponent $m(t)$. In the first part of this work, extreme $m(t)$ values (1 and ∞, respectively) are employed to investigate FLVQ asymptotic behaviors. This analysis shows that when $m(t)$ tends to either one of its extremes, FLVQ is affected by trivial vector quantization, which causes centroids collapse in the grand mean of the input data set. No analytical criterion has been found to improve the heuristic choice of the range of $m(t)$ change. In the second part of this paper, two FLVQ and SOM classification experiments of remote sensed data are presented. In these experiments the two nets are connected in cascade to a supervised second stage, based on the delta rule. Experimental results confirm that FLVQ performance can be greatly affected by the user's definition of the range of change of the weighting exponent. Moreover, FLVQ shows instability when its traditional termination criterion is applied. Empirical recommendations are proposed for the enhancement of FLVQ robustness. Both the analytical and the experimental data reported seem to indicate that the choice of the range of $m(t)$ change is still open to discussion and that alternative clustering neural-network approaches should be developed to pursue during training: 1) monotone reduction of the neurons' learning rate and 2) monotone reduction of the overlap among neuron receptive fields.**

***Index Terms*— Fuzzy clustering, growing neural-network, hybrid classification system, probabilistic and possibilistic membership, self-organizing map, soft and hard competitive learning, topology preserving mapping.**

I. Introduction

THE soft competitive self-organizing map (SOM) algorithm has been extensively studied and applied successfully for different applications [1], [2]. The algorithm is based on heuristics derived from neurophysiological studies and implemented as empirical user-defined functions of time [1], [2]. The first Kohonen heuristic rule requires that learning rates decrease monotonically with time according to a cooling schedule, i.e., as the number of processing epochs increases, all learning rates (winner as well as nonwinners) decrease toward zero, so that SOM, rather than convergence to a local minimum of a cost function, reaches termination. This Kohonen constraint was originally developed for the hard competitive Kohonen's vector quantization (VQ) algorithm [1]. It has been proven that, to avoid possible oscillations of the template estimate, hard clustering VQ should satisfy the following recommendations [3], [4]: 1) as $t \to \infty$, then $\alpha(t) \to 0$ and 2) $\sum_{t=0}^{\infty} \alpha(t) = \infty$, where t is the iteration counter and $\alpha(t)$ is the learning rate. A possible expression of $\alpha(t)$ that satisfies these two conditions is the harmonic sequence $\alpha(t) = 1/t$. In this case VQ becomes equivalent to McQueen's c-means [5]. Another possible expression of $\alpha(t)$ is an exponentially decaying learning rate, e.g., $\alpha(t) = \alpha_i(\alpha_f/\alpha_i)^{t/T}$, where $\alpha_i > \alpha_f$ are the initial and final learning rates and T is the iteration limit. Experiments comparing McQueen's c-means and VQ exploiting an exponentially decaying learning rate indicate that the latter method is less susceptible to poor initialization and for many data distributions gives lower mean square error [5].

The second Kohonen heuristic rule requires that the size of the update (resonance) neighborhood centered on the winner neuron must decrease monotonically with time, such that a soft competitive bubble strategy converges to a hard competitive winner-takes-all (WTA) learning paradigm. This is tantamount to stating that the overlap among neuron receptive fields must decrease monotonically with time, until receptive fields become Voronoi polyhedra [6]. This soft competitive approach is well known in the field of data compression, where the vector quantization problem requires the minimization of a cost function, the distortion error, which, in general, has many local minima [7]. Traditional c-means clustering algorithms, either off-line (Lloyd or Forgy's [5], [8]) or on-line (McQueen's [5], [8], [9]), employ a WTA learning strategy to minimize the distortion error. Since purely local adaptations may not be able to get the system out of a poor local minimum when this lies in the proximity of the status where the system was started [5], c-means clustering algorithms are sensitive to initialization of templates, i.e., different initializations may lead to very different minimization results. To avoid confinement to local minima during the adaptation procedure, a common approach is to introduce a *soft competitive learning* scheme that not only adjusts the winning cluster but also affects all cluster centers according to their input pattern proximity [7]. In general, soft competitive learning not only decreases dependency on initialization but also reduces the presence of dead units [5]. One interesting approach of this kind is the "maximum entropy clustering" whose approximate solution is provided by deterministic annealing [10]. Deterministic annealing can be interpreted as a Gaussian mixture estimation by the expectation-maximization (EM) approach [11] where Gaussian variances are fixed [9]. Deterministic annealing interprets Gaussian variances as the temperature and shows that by starting the adaptation process at a high temperature which must be slowly lowered to zero, then local minima in the distortion error are avoided [7].

Manuscript received February 28, 1997; revised November 17, 1997.

A. Baraldi and F. Parmiggiani are with IMGA-CNR, Bologna 40129, Italy.

P. Blonda, G. Pasquariello, and G. Satalino are with IESI-CNR, 70126 Bari, Italy.

Publisher Item Identifier S 1045-9227(98)06255-9.

Despite its many successes in practical applications, SOM suffers from some major deficiencies, many of which are highlighted in [2] and listed below.

- Termination is not based on optimizing any model of the process or its data [12]. Indeed it has been proven 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 [13]. SOM rather features a set of potential functions, one for each neuron, to be independently minimized following a stochastic gradient descent [13].
- The final weight vectors are affected by the order of the input sequence [12].
- The final weight vectors are affected by the initial conditions [12].
- The learning rate, the size of the resonance neighborhood and the strategy to alter these two parameters during learning must be varied from one data set to another to achieve "useful" results [12].
- Topology preserving mapping is not guaranteed [6].
- Probability density function (pdf) estimation is not achieved [14]. Attempts have been made to interpret the density of codebook vectors as a model of the input data distribution but with limited success [5], [14]–[16].
- Prototype parameter estimate may be severely affected by noise points and outliers. This is due to the fact that the learning rate is computed by SOM as a function of the number of processing epochs and of the neuron position in the grid, while it is independent of the actual distance separating the input pattern from the cluster template.

To ameliorate problems suffered by both VQ and SOM, several clustering algorithms have been proposed in recent years. Among these, the fuzzy learning vector quantization (FLVQ) algorithm [20], which was first called fuzzy Kohonen clustering network (FKCN) [12], has quickly gained popularity as a fairly successful batch clustering algorithm.

In this paper the effectiveness of the FLVQ model in overcoming the SOM limitations is investigated by analytical and experimental approaches. The analytical section explores the FLVQ asymptotic behaviors when the weighting exponent is moved to its extremes. This analysis stresses some minor discrepancies between our findings and those proposed in the literature. Unfortunately, no analytical tool but only heuristic rules are provided to optimize the range of change of weighting exponent $m(t)$. The experimental section involves the classification of low- and high-dimensional remote sensed data, obtained by using FLVQ as an unsupervised module in a two-stage classification system [17]. Specifically, the investigation is focused on the FLVQ sensitivity to changes in parameters and termination criterion. Our experimental findings: 1) confirm the theoretical analysis about FLVQ sensitivity to changes in the range of change of the weighting exponent; 2) reveal that FLVQ features instability when its traditional termination criterion is employed; and 3) provide empirical recommendations which can enhance FLVQ robustness. It is therefore suggested that new neural-network approaches enforcing cooperative/competitive learning strategies should be investigated to overcome FLVQ deficiencies in satisfying Kohonen's constraints.

II. Review of FLVQ

FLVQ combines the on-line Kohonen weight adaptation rule with the fuzzy set membership function proposed by the batch fuzzy c-means (FCM) algorithm [12], [20]. This allows FLVQ to employ less user defined parameters than SOM by computing the learning rate and the size of the update neighborhood directly from the data as a function of a weighting exponent $m(t)$, where time t is defined as the number of processing epochs. Unlike SOM, FLVQ is a batch clustering method and employs metrical neighbors in the input space rather than topological neighbors belonging to an output lattice. It is worth noting that we refer implicitly to FLVQ as the descending FLVQ algorithm [20]. Descending FLVQ features a monotonically decreasing weighting exponent $m(t)$, as recommended in [20].

Limiting behaviors of FLVQ and FCM have been extensively studied in [12] and [20], as well as in recent papers [21]–[23]. In this section, we intend to highlight FLVQ trivial learning behaviors as limiting properties. Discrepancies between our analysis and that proposed in the existing literature are stressed wherever necessary.

A. Input Parameters

FLVQ requires the user to define: 1) number c of natural groups to be detected; 2) the initial and final weighting exponent m_0 and m_f, controlling the "amount of fuzziness" of the algorithm, i.e., the degree of overlap among neuron receptive fields; in [20], the heuristic constraint $7 > m_0 > m_f > 1.1$ is recommended (in line with the choice of $m \in [1.1, 5]$ suggested for FCM applications [20]); 3) parameter $t_{\max}$, defined as the maximum number of epochs; and 4) a termination error ϵ, which is employed in the following termination strategy: let us define quantity $E_t = \sum_{i=1}^{c} ||V_{i,t} - V_{i,t-1}||^2$, where $V_{i,t}$ is the ith cluster prototype at time t; if $(E_t \leq \epsilon)$ then the algorithm is terminated.

B. Equations

The objective function of the FCM algorithm to be minimized is the classical within-groups sum of squared errors function generalized to the infinite family written as [12]

$$J_m(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{i,k})^m ||x_k - V_i||^2 \tag{1}$$

where the weighting exponent m, termed "amount of fuzziness" [12], is a user-defined resolution parameter belonging to range $(1, \infty)$; $U = [u_{i,k}]$, $i = 1, \cdots, c$, $k = 1, \cdots, n$, is a fuzzy c-partition of X (whose definition is provided below); c is the total number of categories; n is the total number of input patterns; $X = [x_k]$, $k = 1, \cdots, n$, is the input data set made of input patterns x_k; $V = (V_1, V_2, \cdots, V_c)$ is a vector of unknown cluster centers. In FCM, (1) is differentiated with respect to $u_{i,k}$ (for fixed V_i and m) and to V_i (for fixed $u_{i,k}$ and m), and, by applying the condition $(\sum_{i=1}^{c} u_{i,k} = 1)$, we obtain as the coupled first-order necessary conditions for solutions (U, V) that are local minimizers of the objective function (1) [22], [24], [25]

$$u_{i,k} = \frac{\left(\frac{1}{\|x_k - V_i\|^2}\right)^{1/(m-1)}}{\sum_{j=1}^{c}\left(\frac{1}{\|x_k - V_j\|^2}\right)^{1/(m-1)}} \qquad i = 1, 2, \cdots, c;\ k = 1, 2, \cdots, n \tag{2}$$

$$V_i = \eta_i \cdot \sum_{k=1}^{n} w_{i,k} \cdot x_k = \sum_{k=1}^{n} \alpha_{i,k} \cdot x_k, \qquad i = 1, 2, \cdots, c \tag{3}$$

where

$$w_{i,k} = (u_{i,k})^m \tag{4}$$

is termed the competition function [21]–[23],

$$\eta_i = \frac{1}{\sum_{s=1}^{n} w_{i,s}} \tag{5}$$

is a constant coefficient, and

$$\alpha_{i,k} = \eta_i \cdot w_{i,k} \tag{6}$$

is the learning rate.

Equation (2) provides the degree of compatibility of input pattern x_k with the vague concept represented by cluster center V_i. This equation is a membership function because it satisfies the three conditions required to state that c fuzzy subsets are a fuzzy c-partition of X. These conditions are [12], [25]: 1) $u_{i,k}$ belongs to [0, 1], $\forall i, \forall k$; 2) $\sum_{i=1}^{c} u_{i,k} = 1, \forall k$; and 3) $0 < \sum_{k=1}^{n} u_{i,k} < n, \forall i$. As was the case with $u_{i,k}$, $w_{i,k}$, and $\alpha_{i,k}$ also belong to [0, 1], $\forall i, \forall k$. Constraint 2) above is an inherently probabilistic constraint [26], relating (2) to the class of *soft-max output activation functions* providing posterior probability estimates [28]. The output value of (2) is also termed probabilistic or relative or constrained fuzzy membership (typicality), in contrast with the possibilistic or absolute membership [26], provided by, for example, each single term of the sum at the denominator of (2). It is worth noting that the relationship between probabilistic membership (2) and Bayes' relation is rather loose since the absolute membership provided by each single term of the sum at the denominator of (2) belongs to range $(0, \infty)$, i.e., it cannot be considered as a class conditional likelihood estimate.

It can be observed that, if $\|x_k - V_i\| \to 0, \forall i \in \{1, c\}$, then $u_{i,k} \to 1/c$. This causes "the relative membership problem of FCM" [26]. It means that, since (2) provides membership values that are relative numbers, then noise points and outliers may have significantly high membership values and they can severely affect the prototype parameter estimate (3). Equation (3) shows the first minor discrepancy with the existing literature: $\alpha_{i,k}$, rather than $w_{i,k}$, as suggested in [12, p. 760], is the learning rate coefficient employed by FCM (and FLVQ as well, see below), allowing category prototype V_i to be attracted by input pattern x_k. It can be observed that (3) is equivalent to the weight adaptation rule employed in the batch form of the SOM algorithm [2], [33], i.e., an alternative expression for (3) is the off-line version of the on-line Kohonen weight adaptation rule

$$\begin{aligned} V_{i,t} &= \sum_{k=1}^{n} \alpha_{i,k} \cdot x_k = V_{i,t-1} + \sum_{k=1}^{n} \alpha_{i,k}(x_k - V_{i,t-1}) \\ &= V_{i,t-1} + \eta_i \cdot \sum_{k=1}^{n} w_{i,k}(x_k - V_{i,t-1}) \\ &\quad i = 1, 2, \cdots, c. \end{aligned}$$

FLVQ adopts FCM (2)–(6), where $m = m(t)$ is not fixed by the user but it is rather defined as a monotone decreasing function of time [23]. This means that: 1) when $m(t)$ is fixed, then FLVQ is equivalent to FCM [20] and 2) FLVQ is not expected to minimize (1): since FLVQ uses (2) and (3) at each iteration with $m = m(t)$, all we can say is that at every processing epoch FLVQ uses a pair $(U(t), V(t))$ that is necessary to minimize $J_m(U(t), V(t))$ [20].

The decreasing expression of $m(t)$ is [20]

$$m = m(t) = m_0 - (t \cdot \Delta m) \quad \text{where } \Delta m = (m_0 - m_f)/t_{\max},\ m_0 > m_f. \tag{7}$$

1) Asymptotic Case A—Trivial Vector Quantization: It can be demonstrated (e.g., see [20, p. 736] and [12, p. 758]) that

$$\lim_{m \to \infty} \{u_{i,k}\} = 1/c, \qquad \forall i, \forall k. \tag{8}$$

Equation (8) shows that when $m \to \infty$, every neuron features the same membership value whatever input k and category i may be. In other words, the membership function is "maximally fuzzy" [12], i.e., the size of the receptive field of every processing element tends to its maximum extension (infinity), and the degree of overlap among neuron receptive fields is equal to 100%. Therefore

$$\lim_{m \to \infty} \{w_{i,k}\} = (1/c)^\infty = 0, \qquad \forall i, \forall k, c > 1. \tag{9}$$

As the second minor discrepancy with the existing literature it can be observed that this result is inconsistent with that proposed in [12, p. 760], where, with regard to FKCN, an equation equivalent to the following can be found: $\lim_{m \to \infty} \{w_{i,k}\} = 1/c$. Finally, with regard to learning rates, we obtain

$$\lim_{m \to \infty} \{\alpha_{i,k}\} = \frac{(1/c)^\infty}{n \cdot (1/c)^\infty} = 1/n, \qquad \forall i, \forall k, c > 1 \tag{10}$$

$$\lim_{m \to \infty} \{V_i\} = \frac{\sum_{k=1}^{n} x_k}{n}, \qquad \forall i, c > 1. \tag{11}$$

Equation (11) shows that all input patterns are weighted the same whatever category i may be. This condition is a peculiar soft competitive learning case where the size of the update neighborhood is extended to the entire net, and the intensity of learning rates is constant through the update neighborhood. It leads to a trivial vector quantizer: all node vectors migrate to the same point in the measurement space, i.e., all input patterns are mapped into the same prototype which is equivalent to the grand mean of X [20]. It is worth noting that (e.g., see [21, p. 6])

$$\lim_{m\to\infty} J_m(U, V) = 0.$$

2) Asymptotic Case B—Hard Competitive Learning Plus Trivial Dead Unit Relocation: We observe that

$$\lim_{m\to 1^+} \{u_{i,k}\} = \lim_{m\to 1^+} \{w_{i,k}\} = \begin{cases} 1, & \text{if neuron } i \text{ is the winning neuron, } \forall k \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Then, if $m \to 1^+$, the membership function is hard such that receptive fields become equivalent to Voronoi polyhedra [6], i.e., the degree of overlap among neuron receptive fields becomes equal to zero. This does not mean, however, that the same hard condition holds true for the learning rate function. In fact

$$\lim_{m\to 1^+} \{\alpha_{i,k}\} = \begin{cases} 1/n_i, & \text{if neuron } i \text{ is the winner unit for } x_k \\ & \text{where } 1 \le n_i = \|X_i\| \le n \text{ is the cardinality} \\ & \text{of subset } X_i \subseteq X \text{consisting of patterns} \\ & \text{featuring neuron } i \text{ as their best-matching} \\ & \text{unit} \\ 0, & \text{if neuron } i \text{ is not the winner unit for } x_k \\ & \text{and if } n_i \ge 1, \text{ i.e., unit } i \text{ is the winner for} \\ & \text{at least one pattern } x_h \ne x_k,\ h \in \{1, n\} \\ 1/n, & \text{if } n_i = 0, \text{ i.e., if neuron } i \text{ is not the winner} \\ & \text{unit either for } x_k \text{ or for any other input} \\ & \text{pattern, i.e., if neuron } i \text{ is a dead unit} \end{cases} \tag{13}$$

$$\lim_{m\to 1^+} \{V_i\} = \begin{cases} \dfrac{\sum\limits_{x_k \in X_i} x_k}{n_i}, & \text{if } n_i \ge 1 \\ \dfrac{\sum\limits_{k=1}^{n} x_k}{n}, & \text{if } n_i = 0. \end{cases} \tag{14}$$

Equations (12)–(14) provide the third discrepancy with the existing literature. In [12], [20], it is stated that if $m \to 1^+$, then both FCM and FLVQ are hard c-means, while (12)–(14) show that both FCM and FLVQ become similar but not equivalent to a Forgy's c-means algorithm [8]. Forgy's c-means features a singularity condition when its hard competitive prototype updating, $\sum_{x_k \in X_i} x_k/n_i, \forall n_i$, deals with a dead unit ($n_i = 0$). In this case, the prototype update cannot be calculated. On the contrary, when (14) encounters dead units, these cluster centers are moved to the center of gravity of the input data set. Equation (21b) found in [20, p. 736] oversimplifies (14) by stating that $\lim_{m\to 1^+} \{V_i\} = \sum_{x_k \in X_i} x_k/n_i, \forall i$. Equations (11) and (14) highlight the fact that the most problematic choice in FCM/FLVQ is the weighting exponent [12]. Unfortunately, (2)–(6) applied iteratively are unable to move away from each other identical centroids, i.e., whenever centroids converge to the center of gravity (grand mean) of X then these resources cannot be any longer exploited as separate processing units. In case of a descending FLVQ implementation, since the nonrecoverable collapse of dead units described by (14) eventually occurs at the end of the iteration process (when $m_f \to 1^+$), then this loss of resources cannot be considered as an asymptotic deficiency of the algorithm.

We summarize our asymptotic analysis as follows. To avoid inefficient exploitation of processing resources: 1) m must be kept away from one and infinity in FCM and 2) $m(t)$ must be kept away from infinity in descending FLVQ. Although several studies have attempted to find a good way to choose m, this choice is still largely heuristic [12]. Most users of FCM find a value in the range [1.1, 5] [20]. In line with this empirical range, the heuristic constraint $7 > m_0 > m_f > 1.1$ is recommended for FLVQ [20]. This constraint is consistent with the results of our asymptotic analysis to guarantee efficient exploitation of processing resources. Unfortunately, (8)–(14) do not provide any analytical criterion that optimizes the choice of the range of change of variable $m(t)$ in reducing the degree of overlap among neuron receptive fields. Therefore, this choice can be considered as problematic as that of the monotonically decreasing bubble size in SOM.

C. Comments

Let us compare the main functional features of FCM and FLVQ in comparison with those of SOM (see Section I). FCM, featuring a fixed resolution parameter m, is such that:

1) It is affected by the relative membership problem (high sensitivity to noise, i.e., low robustness).
2) It does not satisfy the first Kohonen constraint, i.e., no cooling schedule is pursued since the learning rate decreases with the distance from the winner node, but it is independent of time.
3) It does not satisfy the second Kohonen constraint, i.e., no model transition from soft competitive to hard competitive learning is driven through time.
4) When parameter m is fixed according to the heuristic rule $m \in [1.1, 5]$, then the possibility of generating coincident clusters is reduced.

Descending FLVQ is such that:

1) It is affected by the relative membership problem (high sensitivity to noise, i.e., low robustness).
2) The learning rate and the size of the update neighborhood are not user defined as in SOM, but are computed directly from the data.
3) It does not satisfy the first Kohonen constraint, i.e., no cooling schedule is pursued because the learning rate decreases with the distance from the winner node, but it does not necessarily decrease monotonically with time, as acknowledged in [20]: (10) and (13) show that, for a given pattern x_k, the learning rate $\alpha_{i,k}$ of the winner neuron i increases to $1/n_i$ while the other $(c-1)$ rates remain close to $1/n$ or tend toward zero.
4) It satisfies the second Kohonen constraint when a heuristic rule such as $7 > m_0 > m_f > 1.1$ is enforced. This

TABLE I
FUNCTIONAL COMPARISONS BETWEEN LEARNING VECTOR QUANTIZATION ALGORITHMS

		Batch updates	Sequential updates
Crisp learning		-	VQ
Soft learning	Membership function where m is constant	FCM, EFLVQ-F[1]	GLVQ-F[2]
	Membership function where $m = m(t)$ is a function of time	FLVQ	-
	Other membership functions with no weighting exponent	-	GLVQ, FALVQ
	Width of the learning rate distrib. is constant	FCM, EFLVQ-F[1]	GLVQ[3], GLVQ-F[2] FALVQ
	Width of the learning rate distrib. decreases with time	FLVQ	-
Cooling schedule (learning rate decreases with time)		EFLVQ-F[4]	VQ, FALVQ, GLVQ, GLVQ-F[5]
No cooling schedule		FCM, FLVQ	-

[1] see [21], p. 252 ($m = 2$).
[2] see [4], p. 1068 ($m = 2$).
[3] extended to the entire net.
[4] see [21], p. 251.
[5] see [23], p. 33.

rule simultaneously reduces the possibility of generating coincident clusters.

5) Despite recent advances in the field, the objective function minimized by FLVQ is still unknown. It has been recently proved that FLVQ updating can be seen as a special case of the extended FLVQ family (EFLVQ-F) learning schemes [21]–[23]. EFLVQ-F tries to minimize the FCM cost function (1) by adopting (2), these equations assuming that m is fixed (see Section II-B). Since this hypothesis does not hold true for FLVQ, we conclude that FLVQ is not derived by minimizing the FCM/EFLVQ-F objective function.
6) Due to batch learning, the final weight vectors are not affected by the order of the input sequence, this feature being shared with the batch version of SOM [2]. It is well known that batch learning, where parameters are estimated on the basis of the full data set, is the preferred choice for small data sets [9], while on-line learning is considered an extremely desirable property of any clustering neural model [34], although it typically results in systems that become order-dependent during training.
7) Unlike SOM, FLVQ employs metrical neighbors in the input space rather than topological neighbors in the output lattice. Thus, FLVQ cannot manage topological information, while SOM guarantees neighborhood preservation of the inverse mapping from the network to the input manifold, but it does not necessarily guarantee neighborhood preservation of the mapping from the input manifold to the network [6].

D. State of the Art

Several neural-network models found in the literature try to solve some of the problems related to the SOM procedure.

We believe that no optimization criterion can be applied to the choice of the range of change of parameter m when membership (2) is adopted to satisfy Kohonen's constraints while reducing the number of parameters of SOM (see Section I). This observation may justify the recent development of several classes of algorithms which are related to FLVQ. EFLVQ-F is a broad family of batch FLVQ algorithms minimizing the FCM cost function (1) [21]–[23]. FLVQ is also related to several on-line fuzzy clustering algorithms such as the sequential generalized LVQ (GLVQ) [3] and GLVQ family algorithms (GLVQ-F) [4], and the class of on-line fuzzy algorithms for learning vector quantization (FALVQ) [23], [30]. Table I summarizes functional comparisons between these LVQ algorithms.

From the analysis of Table I we observe that, to avoid empirical decisions in the exploitation of (2), more recently published fuzzy vector quantizers, either on-line (FALVQ) or off-line (EFLVQ-F), adopt the following strategies. EFLVQ-F exploits the update rule (3) where: 1) coefficient η_i is no longer computed as (5), but it is rather defined as a monotonically decreasing function of the iteration number, according to a Kohonen cooling schedule and 2) $m = 2$ [21], p. 251. This means that, although EFLVQ-F adopts a cooling schedule, it does not allow the degree of overlap among neuron receptive fields to decrease monotonically with time in line with Kohonen's constraints (see Section I). FALVQ no longer employs (3) as its update rule. On the other hand, FALVQ adopts a cooling schedule such that the learning rate decreases monotonically with time. In different FALVQ implementations, one different parameter controls the size of the update neighborhood. When this parameter tends to either one or both of its extreme values, then FALVQ algorithms employ no soft-max adaptation rule, but a crisp WTA strategy which reduces the proposed algorithms to Kohonen's VQ. Although experimental results reveal that there is a wide range of parameter values that leads to satisfactory clustering results, the choice of this parameter can be considered as problematic as that of the weighting exponent m in FCM. To summarize, although FALVQ adopts a cooling schedule, it does not allow the degree of overlap among neuron receptive fields to decrease monotonically with time in line with Kohonen's constraints.

With regard to the issue of developing robust clustering procedures, several strategies have been proposed in the literature aiming to improve outlier detection capability of both SOM and FLVQ. For example, the growing neural gas (GNG) algorithm inserts a new processing unit in the proximity of the neuron featuring the highest accumulated requantization error [15], [35]. This insertion strategy is such that a single poorly mapped pattern does not suffice to initiate the creation of a new unit. Thus, this method avoids overfitting, i.e., avoids that the clustering system fits the noise, not just the signal [28]. Another solution, adopted in the possibilistic c-means (PCM) algorithm [26], is to relax constraint $\sum_{i=1}^{c} u_{i,k} = 1$, $\forall k$, in order not to force the sum of possibilistic memberships of a noise point in all the good clusters to be equal to one (for a review about possibilistic typicality, see [27]). However, this approach features limitations, such as a large sensitivity to the choice of resolution parameters and the generation of coincident clusters depending on the choice of the initial templates, as acknowledged in [29]. A different solution, proposed in [31] and [32], is to estimate, for a given input pattern, the best probabilistic membership (posterior probability) while submitting the possibilistic membership (class conditional probability) to an adaptive resonance theory (ART)-based vigilance test. If the vigilance test is not passed, then the best probabilistic membership estimate is rejected and a new (noise) category is dynamically allocated to categorize that input pattern (since a single poorly mapped pattern suffices to initiate the creation of a new unit, then, to avoid overfitting, noise categories are removed dynamically in [31]).

Solutions to some of the other problems related to the SOM procedure, apart from outlier detection, are found in the literature. For example, the neural gas (NG) algorithm minimizes an objective function and satisfies the two Kohonen learning constraints while using metrical neighbors in the input space rather than topological neighbors belonging to an output lattice [7]. The GNG algorithm (see above), largely based on heuristics, pursues dynamic insertion of neurons, driven by the neuron-based accumulated error, and the dynamic insertion/deletion of synaptic links, based on the competitive Hebbian adaptation rule [6], to achieve robustness against noise as well as perfect topology preserving mapping [6], [15], [35]. The generative topographic map (GTM) employs an output grid of neurons, featuring fixed size, to perform probability density function estimation while the two Kohonen constraints are satisfied and an objective function is minimized [14], [33].

III. Experimental Analysis

The effectiveness of FLVQ was compared with that of SOM in two classification experiments where low-dimensional and high-dimensional remote sensed data were processed, respectively. To perform our experiments, a two-stage (hybrid) classification system, combining both unsupervised and supervised learning, was developed. In the first stage, unsupervised data clustering was carried out by FLVQ and SOM employed in sequence. In the second stage, a feedforward single layer perceptron (SLP), based on the Delta Rule, provided supervised labeling of the extracted clusters. It has been proven that this hybrid classification system features a learning time lower than that required by a purely supervised classifier based on a multilayer perceptron [17].

Specifically in this work, three multitemporal Landsat thematic mapper (TM) subimages, consisting of 1024 × 512 pixels, were used. The TM sensor can measure surface radiance in seven spectral band whose spatial resolution is 30 × 30 m. for all bands except thermal infrared band six. The latter was not considered in this study. The data set analyzed was acquired in April, July, and October 1986 on an area located in Southern Italy, which includes a portion of the Ofanto river basin. In Fig. 1 the original July image is shown. The data used for training and testing the modular system were extracted from 25 fields, covering 7662 pixels of known ground truth determined by means of *in situ* inspections and visual interpretation of aerial photographs [18]. The supervised data set includes seven land cover classes: 1) bare soil; 2) urban areas; 3) pasture; 4) coniferous reforestation; 5) olive groves; 6) vineyards; and 7) cropland.

The multispectral features of each pixel were scaled by dividing each input value parameter by 255, the maximum value that each band can assume.

In the first experimental setting, a low-dimensional data set featuring only two bands of the July image were selected to: 1) monitor the distribution of weights in a bidimensional space and 2) deal with classes featuring low separability, their Jeffries–Matusits (JM) average interclass distance [36] measuring 1.181.

In the second experimental setting, the whole 18-band multitemporal data set was employed. In this case, the classes of interest were characterized by high separability, their average JM value measuring 1.413.

In both experimental setting, the number of input units of the clustering module was fixed as the number of input spectral bands. Different numbers of nodes were used in the output layer to optimize clustering performance. In the second module of the classification system, the size of the SLP input layer was taken as the size of the output clustering module and the number of SLP output neurons was set as the number of land cover classes.

A. FLVQ Numerical Examples

FLVQ stability with respect to changes in m_0 and Δm is investigated both in the low- and high-dimensional data sets, whereas in the literature this investigation regarded only the Anderson Iris data [12], [20]. It should be pointed out that our experiments were conducted using the $m(t)$ formulation provided in [12]. The formulation for given parameters m_0

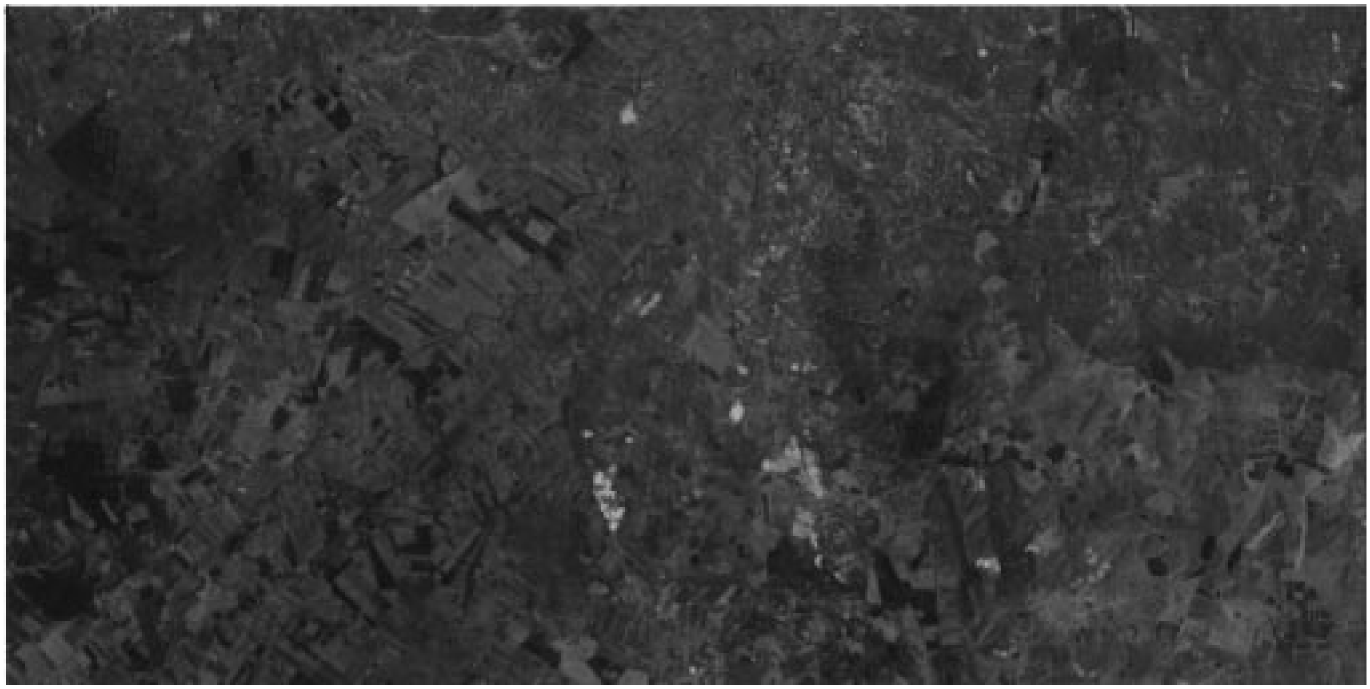


Fig. 1. A color composite of the Landsat TM image with band-5 in red, band-4 in green, and band-3 in blue. The image was acquired over the Ofanto river basin, in Southern Italy, in July 1986.

and $t_{\max}$ (or Δm) becomes

$$m(t) = m_0 - (t \cdot \Delta m), \qquad \text{where } \Delta m = (m_0 - 1)/t_{\max}. \tag{15}$$

Compared to (7), (15) implies that parameter m_f equals one. Since $m(t)$ must be greater than one according to (1) and (2), in our experiments (15) must be interpreted as follows:

$$m(t) = m_0 - (t \cdot \Delta m), \qquad \forall t \in \{1, (t_{\max} - 1)\} \\ \text{where } \Delta m = (m_0 - 1)/t_{\max}. \tag{16}$$

In (16), m_0 and $t_{\max}$ (or Δm) are user-defined parameters. In the sections below, symbol t_{out} identifies the epoch at which FLVQ actually reached termination, either because $t = t_{\max} - 1$ or because the termination criterion is satisfied (see Section II-A). As a consequence t_{out} belongs to the discrete range $t_{\text{out}} \in \{1, (t_{\max} - 1)\}$, whatever parameter $t_{\max}$ (or Δm) selected by the user may be. When termination is reached at epoch t_{out}, the value of parameter $m(t)$ is identified as m_{out}, such that $m_{\text{out}} \geq (1 + \Delta m) > 1$, in agreement with the constraint $m > 1$ adopted in (1) and (2).

1) Classification of Low-Dimensional Data: The FLVQ architecture consists of two nodes in the input layer and 30 nodes in the output layer. The number of output nodes in FLVQ was fixed as follows: starting from seven output nodes, the number of output nodes was increased until performances of the two-stage classification architecture became comparable to those provided by a multilayer perceptron (about 84.0% on training and 83.0% on test). We randomly selected 50% of the labeled pixels for training the network, while the remaining 50% were used for testing.

TABLE II
EXPERIMENT 1: THE PERCENTAGE OF THE OVERALL CLASSIFICATION ACCURACY OBTAINED THROUGH THE MODULAR NEURAL SYSTEM ON LOW-DIMENSIONAL DATA PREPROCESSED BY FLVQ, WITH $m_0 = 2$, TERMINATION PARAMETER ϵ FIXED AT TWO DIFFERENT VALUES AND VARYING PARAMETER Δm

	$\epsilon = 10^{-4}$, $m_0 = 2$				$\epsilon = 10^{-6}$, $m_0 = 2$			
Δm	m_{out}	t_{out}	Train	Test	m_{out}	t_{out}	Train	Test
0.05	1.35	13	82.31%	81.69	1.05	19	82.28%	82.01%
0.02	1.70	15	81.48%	81.72	1.02	49	84.42%	83.08%
0.001	1.98	11	81.11%	80.96%	1.94	56	82.23%	82.16%
0.00001	1.99	11	81.17%	80.91%	1.99	63	81.74%	81.43%

Experiment 1: The stability of FLVQ to changes in Δm was investigated. m_0 was fixed to two and FLVQ was run until the termination test was satisfied. Two different ϵ values were employed in the termination test: $\epsilon = 10^{-4}$ and $\epsilon = 10^{-6}$. t_{out} and m_{out} are the number of epochs and the value of parameter $m(t)$ at termination time. Percentages of overall accuracy obtained by the modular system for the training and testing data sets are shown in Table II. From this table, it can be observed that classification results improve when termination is reached at m_{out} values close to one, regardless of Δm and t_{out}.

Experiment 2: The stability of FLVQ to changes in m_0 was investigated. Δm was fixed to 0.01 and FLVQ was run until the termination test was satisfied. The ϵ values used were the same as the ones in the previous experiment: $\epsilon = 10^{-4}$ and $\epsilon = 10^{-6}$. Percentages of the overall accuracy obtained by the modular system for the training and testing data sets are given in Table III. In agreement with Table II, Table III shows that FLVQ performance tends to improve when termination is reached at decreasing m_{out} values. A comparison between the results reported in Tables II and III reveals that FLVQ

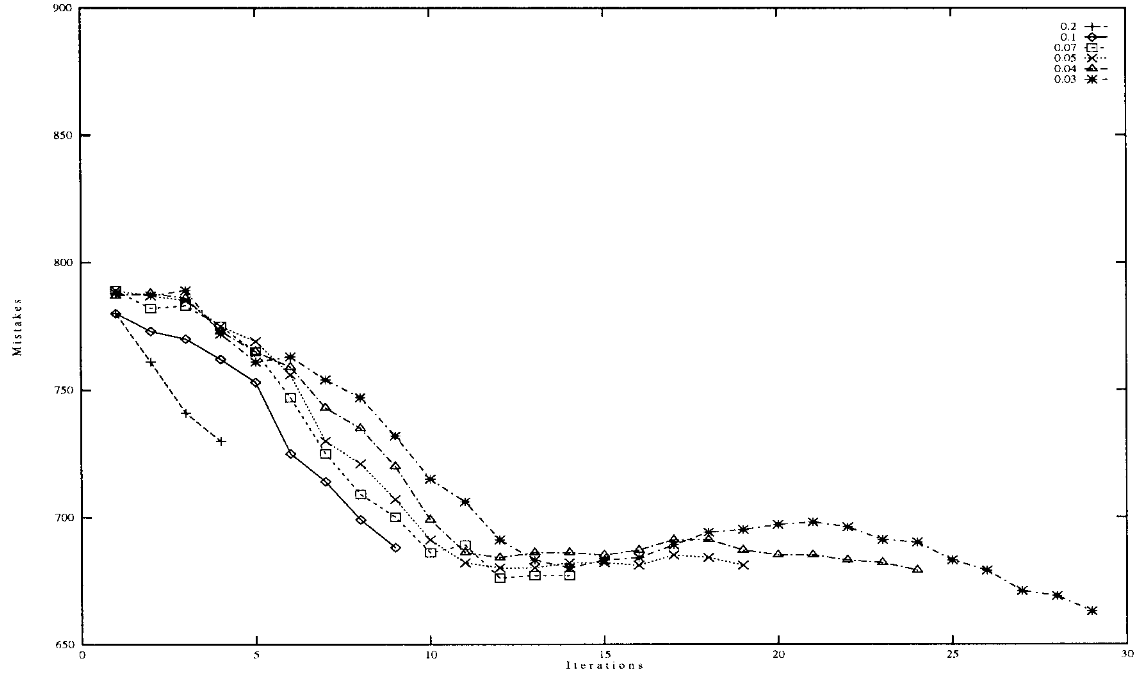


Fig. 2. Experiment 3: Low-dimensional data set. The FLVQ number of mistakes for $m_0 = 2$ and $t_{\max} = 5, 10, 15, 20, 25, 30$ (as a consequence, Δm becomes equal to 0.2, 0.1, 0.07, 0.05, 0.04, 0.03, and m_{out} values are equal to 1.20, 1.10, 1.07, 1.05, 1.04, and 1.03). The termination criterion is removed.

TABLE III
EXPERIMENT 2: THE PERCENTAGE OF THE OVERALL CLASSIFICATION ACCURACY OBTAINED THROUGH THE MODULAR NEURAL SYSTEM ON LOW-DIMENSIONAL DATA PREPROCESSED BY FLVQ, WITH $\Delta m = 0.01$, TERMINATION PARAMETER ϵ FIXED AT TWO DIFFERENT VALUES, AND VARYING PARAMETER m_0

	$\epsilon = 10^{-4}$, $\Delta m = 0.01$				$\epsilon = 10^{-6}$, $\Delta m = 0.01$			
m_0	m_{out}	t_{out}	Train	Test	m_{out}	t_{out}	Train	Test
2	1.89	11	81.01%	80.72%	1.35	65	83.82%	83.18%
3	2.92	8	79.03%	79.05%	1.12	188	83.27%	82.92%
4	3.93	7	78.20%	77.63%	3.58	42	79.24%	78.68%
5	4.92	8	77.97%	77.08%	4.70	30	77.81%	76.77%
6	5.90	10	78.23%	77.34%	5.74	26	77.91%	77.58%

TABLE IV
EXPERIMENT 3: THE PERCENTAGE OF THE OVERALL CLASSIFICATION ACCURACY OBTAINED THROUGH THE MODULAR NEURAL SYSTEM ON LOW-DIMENSIONAL DATA PREPROCESSED BY FLVQ, WITH $m_0 = 2$ AND $t_{\max} = 5$, 10, 15, 20, 25, 30 (AS A CONSEQUENCE, Δm BECOMES EQUAL TO 0.2, 0.1, 0.07, 0.05, 0.04, 0.03 AND m_{out} BECOMES EQUAL TO 1.20, 1.10, 1.07, 1.05, 1.04, AND 1.03). THE TERMINATION CRITERION IS REMOVED. THE PERCENTAGE VALUES AND m_{out} REFER TO EPOCH $t_{\text{out}} = t_{\max} - 1$

	$m_0 = 2$		
t_{max}	m_{out}	Train	Test
5	1.20	81.01%	80.62%
10	1.10	82.10%	81.53%
15	1.07	82.39%	81.77%
20	1.05	82.28%	82.01%
25	1.04	82.34%	81.82%
30	1.03	82.75%	82.29%

is more sensitive to changes in m_0 than to changes in Δm when processing low-dimensional data set. In fact, when m_0 was changed from two to six, the percentage of the overall classification accuracy at $\epsilon = 10^{-6}$ was found to vary from 83.18 to 77.58%, for test data (Table III). The same result suggests that the robustness of the FLVQ algorithm with respect to continuous change in input parameters is affected by the termination criterion. This finding seems to contradict the results obtained in [12] on the Iris data set. It also suggests that the FLVQ termination criterion should be removed.

Experiment 3: In consideration of the findings discussed above, the termination criterion was removed so that FLVQ could iterate $t_{\text{out}} = (t_{\max} - 1)$ times. m_0 was fixed to two, and $t_{\max}$ was set as 5, 10, 15, 20, 25, and 30, respectively. As a result, Δm was assumed equal to 0.2, 0.1, 0.07, 0.05, 0.04, 0.03, and m_{out} values became 1.20, 1.10, 1.07, 1.05, 1.04, and 1.03, respectively. The percentages of the overall accuracy obtained by the modular system for the training and testing data sets are given in Table IV. The FLVQ number of mistakes is shown in Fig. 2. Analysis of Table IV and Fig. 2 shows that after about 15 iterations the performance of FLVQ does not improve significantly. In line with the result reported in Tables II and III, Table IV confirms that FLVQ performance tends to improve as m_{out} tends to one.

Experiment 4: In this step the termination criterion continued to be excluded. Δm was fixed to 0.05, while m_0 was varied according to the following values: two, three, four, five, and six. As a result, $t_{\max}$ became 20, 40, 60, 80, and 100, respectively, while m_{out} became 1.05. Percentages of overall accuracy obtained by the modular system for the training and testing data sets are shown in Table V. The FLVQ number of mistakes is shown in Fig. 3. The combined analysis of Table V and Fig. 3 seems to confirm that when the FLVQ traditional termination criterion is removed, the module becomes robust with respect to both different m_0

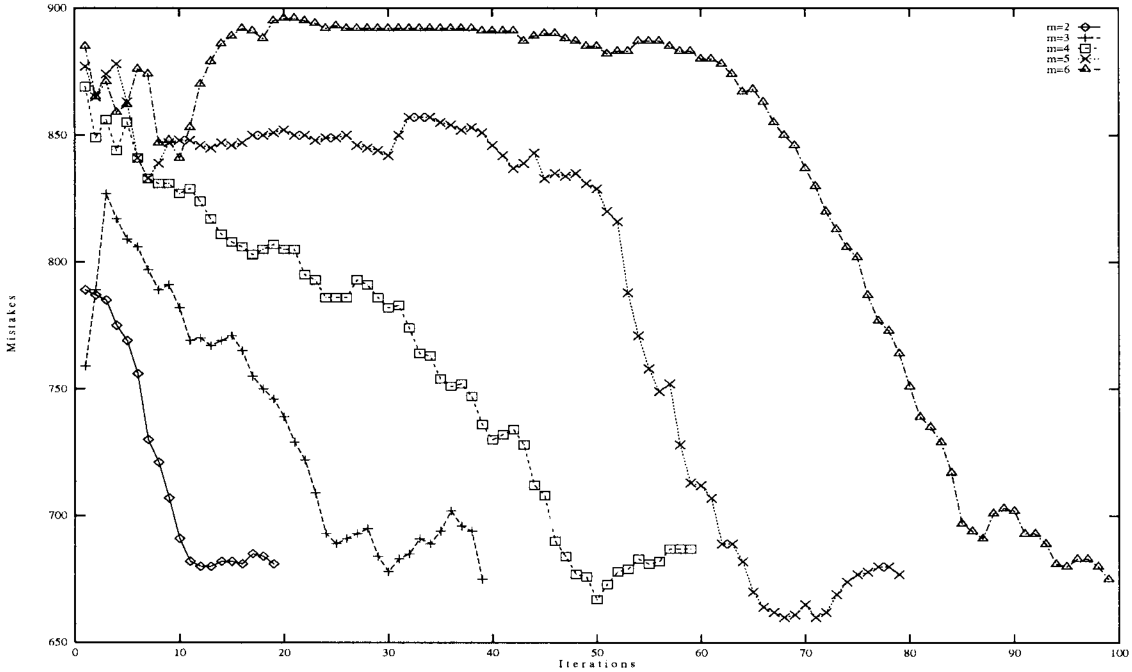


Fig. 3. Experiment 4: Low-dimensional data set. The FLVQ number of mistakes for $\Delta m = 0.05$ and $m_0 = 2$, 3,4, 5 (as a consequence, $t_{\max}$ changes to 20, 40, 60, 80, and 100, while m_{out} becomes equal to 1.05). The termination criterion is removed.

TABLE V
EXPERIMENT 4: THE PERCENTAGE OF THE OVERALL CLASSIFICATION ACCURACY OBTAINED THROUGH THE MODULAR NEURAL SYSTEM ON LOW-DIMENSIONAL DATA PREPROCESSED BY FLVQ, WITH $\Delta m = 0.05$ AND $m_0 = 2, 3, 4, 5, 6$ (AS A CONSEQUENCE, $t_{\max}$ CHANGES TO 20, 40, 60, 80, AND 100, WHILE m_{out} BECOMES EQUAL TO 1.05). THE TERMINATION CRITERION IS REMOVED. PERCENTAGE VALUES REFER TO EPOCH $t_{\text{out}} = t_{\max} - 1$

	$\Delta m = 0.05$, $m_{out} = 1.05$		
m_0	t_{max}	Train	Test
2	20	82.28%	82.01%
3	40	82.44%	82.06%
4	60	82.13%	81.67%
5	80	82.39%	82.35%
6	100	82.44%	82.63%

TABLE VI
EXPERIMENT 5: THE PERCENTAGE OF THE OVERALL CLASSIFICATION ACCURACY OBTAINED THROUGH THE MODULAR NEURAL SYSTEM ON LOW-DIMENSIONAL DATA PREPROCESSED BY FLVQ, WITH $t_{\max}$ EQUAL TO 15 AND 40, AND $m_0 = 2, 3, 4, 5, 6$ (AS A CONSEQUENCE, Δm AND m_{out} ARE FORCED TO CHANGE). THE TERMINATION CRITERION IS REMOVED

	$t_{max} = 15$			$t_{max} = 40$		
m_0	m_{out}	Train	Test	m_{out}	Train	Test
2	1.07	83.39%	81.77%	1.03	83.14%	82.87%
3	1.33	83.04%	82.66%	1.05	82.44%	82.06%
4	1.20	83.39%	82.11%	1.08	82.49%	82.71%
5	1.27	79.73%	80.30%	1.10	83.01%	83.03%
6	1.33	78.69%	77.89%	1.13	81.74%	82.27%

starting values and number of epochs $t_{\max}$ iff the same value $m_{\text{out}} \approx 1^+$ is achieved. Vice versa, if FLVQ is forced to stop before reaching $m_{\text{out}} \approx 1^+$ values, a different performance can be obtained, as shown in Table III.

Experiment 5: In this experimental setting, the different m_0 values were decreased with different Δm values. This involved the choice of the same number of $t_{\max}$ iterations for each m_0 value. In particular, the percentage of the overall accuracy obtained by the modular system for the training and testing data sets in the present experiment are given in Table VI for $t_{\max} = 15$ and $t_{\max} = 40$. In agreement with Tables II–V, Table VI indicates that best performance can be achieved when m_{out} becomes very close to one whatever $m_0 < 7$ and $t_{\max}$ (or Δm) may be.

As an example of the occurrence of each class in the training and test data sets, a pair of confusion matrices is shown in Table VII. These matrices were generated by the hybrid classification system whose clustering stage FLVQ employs no termination criterion and parameters values $m_0 = 2$, $t_{\max} = 20$, $\Delta m = 0.05$, $m_{\text{out}} = 1.05$ (see Experiment 3 above). The overall classification accuracy is computed as in [19].

2) Classification of High-Dimensional Data: The classification of high-dimensional data was performed by choosing an FLVQ output layer consisting of thirty nodes as in the low-dimensional case. In this investigation 50% of the available supervised pixels were employed for training, 50% were used for testing.

Experiment 6: FLVQ stability with changes in Δm was investigated. m_0 was fixed to two and FLVQ was run until the termination test was satisfied. Two different ϵ values were employed in the termination test: $\epsilon = 10^{-3}$ and $\epsilon = 10^{-5}$. The percentages of overall accuracy obtained by the modular system for the training and testing data sets are given in Table VIII. This shows that better classification results can be obtained when termination is reached at m_{out} values close to 1, regardless of Δm and t_{out}.

TABLE VII

THE CONFUSION MATRICES OBTAINED THROUGH THE MODULAR NEURAL SYSTEM ON LOW-DIMENSIONAL DATA PREPROCESSED BY FLVQ WITH $m_0 = 2$ AND $t_{\max} = 20$, $\Delta m = 0.05$, AND $m_{\text{out}} = 1.05$. THE TERMINATION CRITERION IS REMOVED. 1) TRAINING DATA. 2) TEST DATA. LEGEND: I = BARE SOIL; II = URBAN AREAS; III = PASTURE; IV = CONIFEROUS REAFFORESTATION; V = OLIVE GROVES; VI = VINEYARDS; VII = CROPLAND

FLVQ		Ground truth								
Training data		I	II	III	IV	V	VI	VII	TOT	Purity(%)
Classification results	I	100	0	0	0	0	0	0	100	100.00
	II	2	78	2	0	17	0	18	117	66.67
	III	0	0	688	0	46	8	8	750	91.73
	IV	0	0	0	68	8	0	17	93	73.12
	V	5	15	37	9	1082	135	127	1410	76.74
	VI	0	0	6	4	23	196	3	232	84.48
	VII	30	54	84	0	23	0	951	1142	83.27
	TOT	137	147	817	81	1199	339	1124	3844	
Efficiency(%)		72.99	53.06	84.21	83.95	90.24	57.82	84.61		
Overall accuracy 82.28 (%)										

FLVQ		Ground truth								
Test data		I	II	III	IV	V	VI	VII	TOT	Purity (%)
Classification results	I	87	0	0	0	0	0	1	88	98.86
	II	4	82	3	0	18	1	15	123	66.67
	III	1	1	696	0	39	5	12	754	92.31
	IV	0	0	1	69	8	1	25	104	66.35
	V	3	9	50	15	1054	128	91	1350	78.07
	VI	0	0	6	4	17	199	4	230	86.52
	VII	33	72	92	0	28	0	944	1169	80.75
	TOT	128	164	848	88	1164	334	1092	3818	
Efficiency(%)		67.97	50.00	82.08	78.41	90.55	59.58	86.45		
Overall accuracy 82.01 (%)										

TABLE VIII

EXPERIMENT 6: THE PERCENTAGE OF THE OVERALL CLASSIFICATION ACCURACY OBTAINED THROUGH THE MODULAR NEURAL SYSTEM ON HIGH-DIMENSIONAL DATA PREPROCESSED BY FLVQ, WITH $m_0 = 2$, TERMINATION PARAMETER ϵ FIXED AT TWO DIFFERENT VALUES AND BY VARYING PARAMETER Δm

	$\epsilon = 10^{-3}$, $m_0 = 2$				$\epsilon = 10^{-5}$, $m_0 = 2$			
Δm	m_{out}	t_{out}	Train	Test	m_{out}	t_{out}	Train	Test
0.05	1.05	19	96.43%	95.15%	1.05	19	96.43%	95.08%
0.02	1.42	29	94.55%	93.21%	1.02	49	97.41%	95.48%
0.001	1.97	29	88.66%	87.80%	1.93	65	88.48%	87.79%
0.00001	2.0	29	88.39%	86.70%	1.99	57	88.66%	87.74%

TABLE IX

EXPERIMENT 7: THE PERCENTAGE OF THE OVERALL CLASSIFICATION ACCURACY OBTAINED THROUGH THE MODULAR NEURAL SYSTEM ON HIGH DIMENSIONAL DATA PREPROCESSED BY FLVQ, WITH $\Delta m = 0.01$, TERMINATION PARAMETER ϵ FIXED AT TWO DIFFERENT VALUES, AND BY VARYING PARAMETER m_0

	$\epsilon = 10^{-3}$, $\Delta m = 0.01$				$\epsilon = 10^{-5}$, $\Delta m = 0.01$			
m_0	m_{out}	t_{out}	Train	Test	m_{out}	t_{out}	Train	Test
2	1.75	25	90.80%	88.63%	1.01	99	97.05%	95.43%
3	2.98	2	68.13%	67.36%	2.64	36	76.61%	74.58%
4	3.98	2	68.04%	66.98%	3.54	46	75.18%	73.45%
5	4.98	2	67.86%	66.83%	4.56	44	74.20%	71.94%
6	5.98	2	67.77%	66.72%	5.96	5	64.64%	63.05%

Experiment 7: FLVQ stability with changes in m_0 was investigated. Δm was fixed to 0.01 and FLVQ was run until the termination test was satisfied. The ϵ values employed were the same as the ones used in Experiment 6. Table IX shows that in the present setting FLVQ performance tends to improve when termination is reached at decreasing m_{out} values, regardless of t_{out}. Comparison of Tables VIII and IX reveals that FLVQ is sensitive to changes in both m_0 and Δm in the processing of the high-dimensional data set. Moreover, the data reported in Tables IX indicate that the robustness of the FLVQ algorithm can be strongly affected by ϵ values adopted for the termination criterion. In particular, at $m_0 = 3$ with $\epsilon = 10^{-3}$ the percentage of the overall accuracy is 67.36% for the test data, whereas at $m_0 = 3$ with $\epsilon = 10^{-5}$ the overall accuracy reaches 74.58%. Both percentages are lower than the one obtained at $m_0 = 2$ regardless of the ϵ value (Table IX). This contradicts results obtained in [12] on the Iris data.

Experiment 8: In light of the results of experiments 6 and 7, in all the present and following experiments, the termination criterion was removed so that FLVQ was forced to iterate $t_{\text{out}} = (t_{\max} - 1)$ times. m_0 was fixed to two, and $t_{\max}$ was set as 5, 10, 15, 20, 25, and 30, respectively. As a consequence, Δm became equal to 0.2, 0.1, 0.07, 0.05, 0.04, 0.03, and m_{out} values became equal to 1.20, 1.10, 1.07, 1.05, 1.04 and 1.03, respectively. The percentages of overall accuracy obtained by the modular system in this conditions for the training and testing data sets are shown in Table X. The FLVQ number of mistakes is shown in Fig. 4. The combined analysis of Table X and Fig. 4 shows that after about 20 iterations the performance of FLVQ seems not improve significantly for increasing $t_{\max}$ values. In line with the data of Tables VIII and IX, Table X confirms that FLVQ performance tends to improve as m_{out} tends to one.

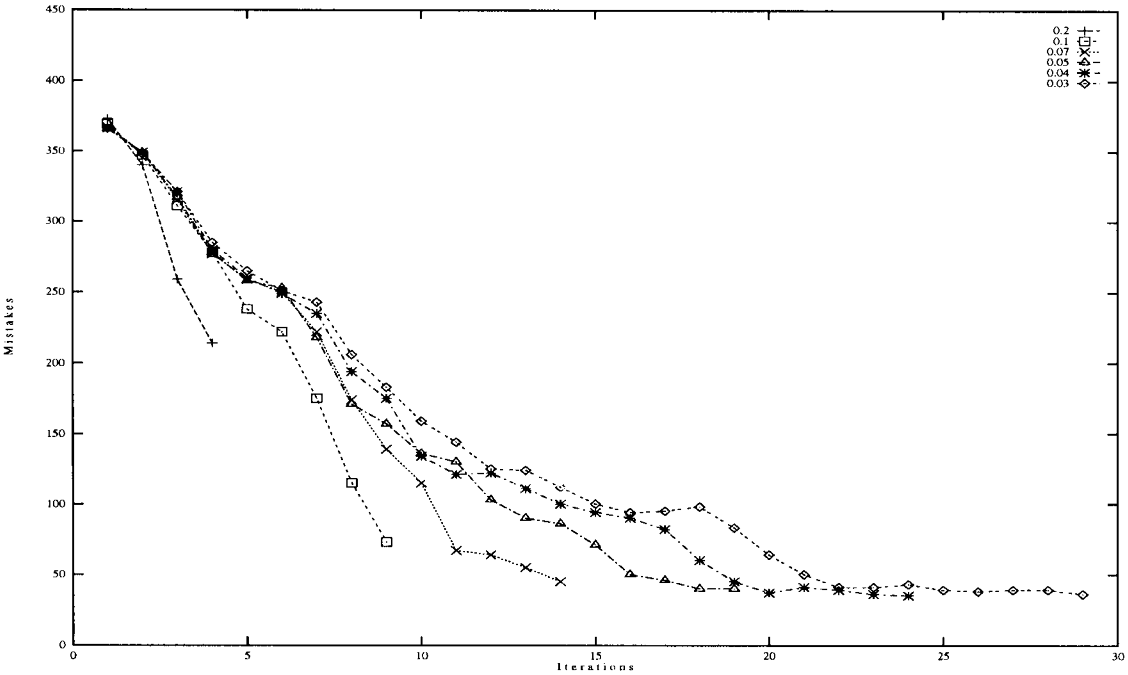


Fig. 4. Experiment 7: High-dimensional data set. The FLVQ number of mistakes for $m_0 = 2$ and $t_{\max} = 5, 10, 15, 20, 25, 30$ (as a consequence, Δm becomes equal to 0.2, 0.1, 0.07, 0.05, 0.04, 0.03, and m_{out} values are equal to 1.20, 1.10, 1.07, 1.05, 1.04, and 1.03). The termination criterion is removed.

TABLE X

Experiment 8: The Percentage of the Overall Classification Accuracy Obtained Through the Modular Neural System on High-Dimensional Data Preprocessed by FLVQ, with $m_0 = 2$ and $t_{\max} = 5, 10, 15, 20, 25, 20$ (as a Consequence, Δm Becomes Equal to 0.2, 0.1, 0.07, 0.05, 0.04, 0.03, and m_{out} Values Are Equal to 1.20, 1.10, 1.07, 1.05, 1.04, and 1.03). The Termination Criterion is Removed. Percentage Values and m_{out} Refer to Epoch $t_{\text{out}} = t_{\max} - 1$

	$m_0 = 2$		
t_{max}	m_{out}	Train	Test
5	1.20	80.89%	80.16%
10	1.10	93.48%	93.11%
15	1.07	95.98%	95.15%
20	1.05	96.43%	95.15%
25	1.04	96.88%	95.38%
30	1.03	96.79%	95.17%

TABLE XI

Experiment 9: The Percentage of the Overall Classification Accuracy Obtained Through the Modular Neural System on High-Dimensional Data Preprocessed by FLVQ, with $\Delta m = 0.05$ and $m_0 = 2, 3, 4, 5, 6$ (as a Consequence, $t_{\max}$ Becomes Equal to 20, 40, 60, 80, and 100, Respectively, While m_{out} Becomes Equal to 1.05). The Termination Criterion Is Removed. Percentage Values Refer to Epoch $t_{\text{out}} = t_{\max} - 1$

	$\Delta m = 0.05$, $m_{out} = 1.05$		
m_0	t_{max}	Train	Test
2	20	96.43%	95.15%
3	40	96.79%	94.97%
4	60	95.45%	94.68%
5	80	96.43%	94.67%
6	100	96.25%	94.15%

Experiment 9: In this experiment, Δm was fixed to 0.05, and m_0 varied as two, three, four, five, and six. As a consequence, $t_{\max}$ became equal to 20, 40, 60, 80, and 100, respectively, while m_{out} became equal to 1.05. The results of this procedure are reported in Table XI. The FLVQ number of mistakes is shown in Fig. 5. The output classification map of the study area obtained at $m_0 = 2$, $t_{\max} = 20$ is displayed in Fig. 6. Analysis of Table XI and Fig. 5 confirms that when its traditional termination criterion is removed, FLVQ is robust with respect to different m_0 starting values and number of epochs $t_{\max}$ iff the same value $m_{\text{out}} \approx 1^+$ is achieved. Vice versa, if FLVQ is forced to stop before this condition, different performance can be obtained (Table IX). This finding is the same as the one reported for the low-dimensional data investigation.

Experiment 10: In the present investigation, the different m_0 values were decreased with different Δm values, by keeping $t_{\max}$ constant. The percentages of the overall accuracy obtained by the modular system for the training and testing data sets are given in Table XII for $t_{\max} = 20$ and $t_{\max} = 40$. In line with the results in Tables VIII–XI, Table XII confirms that the best performance is achieved when m_{out} becomes very close to one.

3) FLVQ Performance Assessment: The findings of the experiments reported can be briefly summerized as follows.

1) Tables II, III, VIII, and IX show that better classification results can be obtained for low- and high-dimensional data sets when termination is reached at m_{out} values close to one, regardless of Δm and t_{out}. Tables IV–VI and X–XII show that in the case of both data sets, FLVQ is robust with respect to different m_0 starting values and

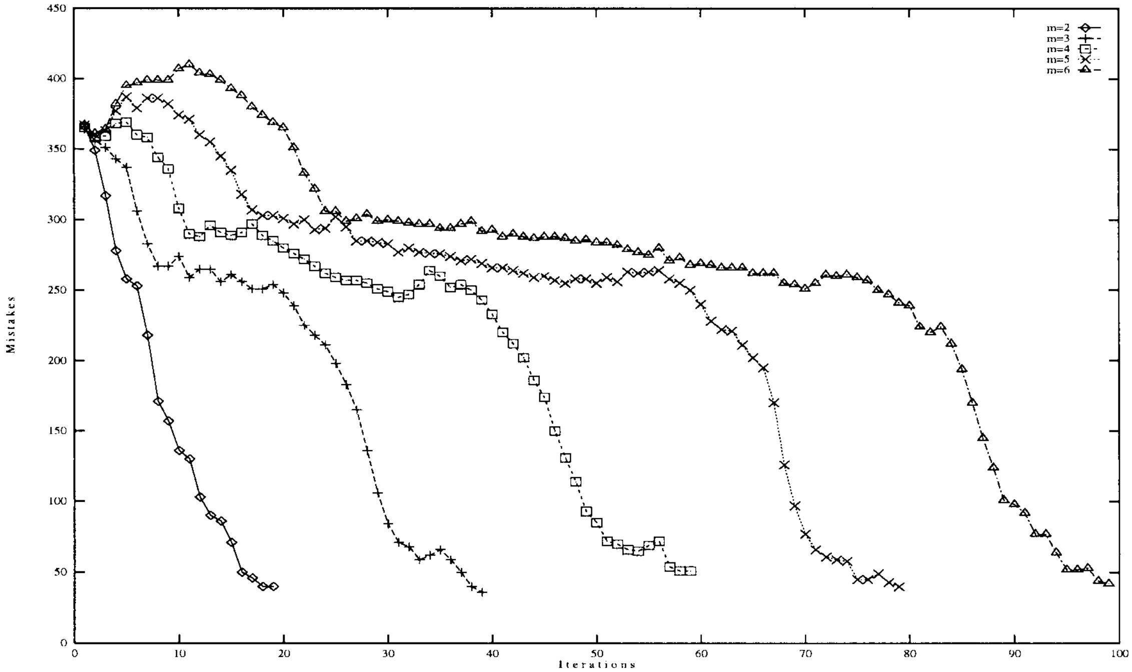


Fig. 5. Experiment 8: High-dimensional data set. The FLVQ number of mistakes for $\Delta m = 0.05$ and $m_0 = 2, 3, 4, 5, 6$ (as a consequence, $t_{\max}$ changes to 20, 40, 60, 80, and 100, while m_{out} becomes equal to 1.05). The termination criterion is removed.

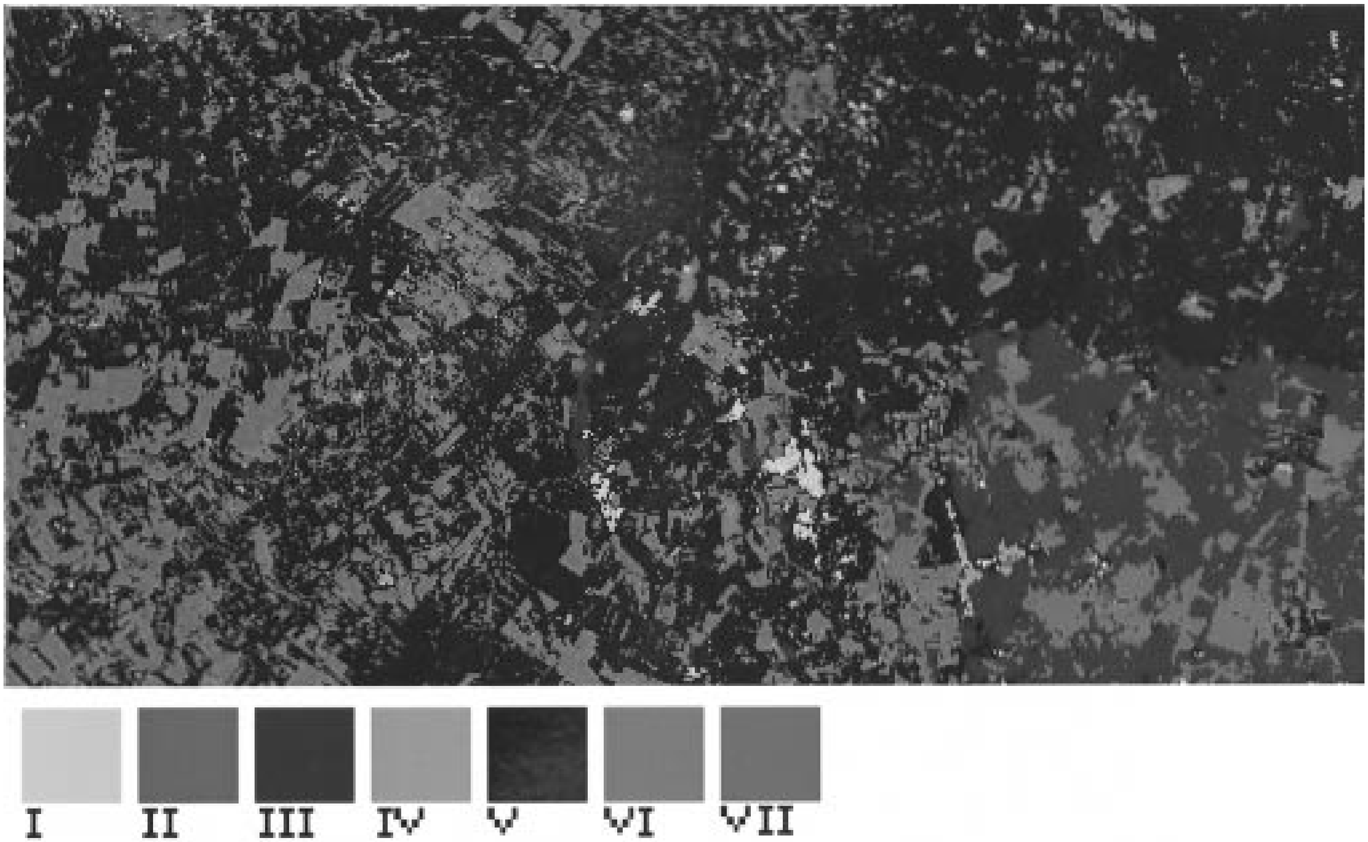


Fig. 6. Experiment 9: High-dimensional data set. Classification map of the study area obtained through the modular system. Legend: I = bare soil; II = urban areas; III = pasture; IV = coniferous reforestation; V = olive groves; VI = vineyards; and VII = cropland. FLVQ employed as the unsupervised module of the two-stage classification system was trained with $m_0 = 2$, $t_{\max} = 20$, and $\epsilon = 0$ (i.e., the termination criterion is removed).

TABLE XII
Experiment 10: The Percentage of the Overall Classification Accuracy Obtained Through the Modular Neural System on High-Dimensional Data Preprocessed by FLVQ, with $t_{\max}$ Equal to 20 and 40, and $m_0 = 2, 3, 4, 5, 6$ (as a Consequence, Δm and m_{out} are Forced to Change). The Termination Criterion Is Removed

	$t_{max} = 20$			$t_{max} = 40$		
m_0	m_{out}	Train	Test	m_{out}	Train	Test
2	1.05	96.43%	95.15%	1.03	97.32%	95.41%
3	1.10	94.91%	93.04%	1.05	96.79%	94.97%
4	1.15	92.05%	90.20%	1.08	95.45%	94.28%
6	1.25	79.38%	77.18%	1.13	93.13%	92.07%

number of epochs $t_{\max}$ (or steps Δm) iff the same value $m_{\text{out}} \approx 1^+$ is achieved and the termination criterion is removed. Therefore these findings seem to advise futher investigation on the opportunity of removing the FLVQ termination criterion.

2) Experiments 1, 2, 6, and 7 show that, for both low- and high-dimensional data set, FLVQ is sensitive to changes in both m_0 and Δm when the FLVQ termination criterion is employed. This finding is not in agreement with the one reported in [12] on the well-structured Anderson Iris data.
3) Experiments 1–10 recommend the removal of the FLVQ termination criterion.

To summarize, Experiments 1–10 support the validity of the following heuristic rules, to be employed in parallel: 1) $7 > m_0$; 2) $m_f = 1.05$, while Δm is fixed anywhere in the range [0.01, 0.05]; and 3) $\epsilon = 0$. These heuristics are such that the traditional FLVQ termination criterion is disactivated, i.e., termination is reached iff $m(t) = m_f = 1.05$. These empirical rules are slightly more severe than the traditional constraint $7 > m_0 > m_f > 1.1$ proposed in [20]. This is probably due to the fact that more complex structured data sets were employed in all our experiments, in place of the well-structured Iris data set adopted in [12] and [20].

B. SOM Numerical Examples

In line with the output number of nodes chosen for FLVQ, the SOM topology consisted of 30 nodes in the output layer. The learning rate chosen was 0.2, since other learning rate tried, i.e., 0.7, 0.9, had provided comparable performance.

The overall classification accuracy obtained by the SOM and SLP module connected in cascade is shown in Tables XIII and XIV for the low- and high-dimensional data set respectively. The epoch numbers are shown in the first column of Tables XIII and XIV, while the first line of the two tables shows the three different radii, expressed in lattice units, being employed to explore the update neighborhood centered on the winner unit.

C. Performance Comparison

The best performance of SOM for both low-dimensional and high-dimensional data (see Tables XIII and XIV) is similar to that provided by FLVQ when the latter method is employed while its traditional termination criterion is suppressed (see Tables IV, V, and X, and XI, respectively). Our conclusion is that relevant differences between the performance of the two systems, if any, seem to be linked to the choice of the starting parameters and heuristic rules.

TABLE XIII
The Percentage of the Overall Classification Accuracy Obtained Through the Modular Neural System on Low-Dimensional Data Preprocessed by SOM with $w = 0.2$, $t_{\max} = 5$, 10, 15, 20, 30, 40, and the Radius of the Resonance Domain Equal to 5, 10, and 15 Pixels

Neigh.	5		10		15	
t_{max}	Train	Test	Train	Test	Train	Test
5	78.46%	77.74%	76.40%	75.54%	74.30%	73.42%
10	82.36%	81.38%	78.64%	77.89%	79.37%	78.92%
15	83.12%	83.05%	83.38%	82.82%	78.49%	78.18%
20	81.35%	81.80%	81.56%	81.77%	82.44%	81.88%
25	84.05%	83.42%	82.96%	82.87%	81.56%	82.06%
30	83.53%	83.08%	82.93%	82.82%	81.01%	81.06%
40	83.17%	82.74%	84.18%	83.73%	81.56%	80.62%

TABLE XIV
The Percentage of the Overall Classification Accuracy Obtained Through the Modular Neural System on High-Dimensional Data Preprocessed by SOM with $w = 0.2$, $t_{\max} = 5$, 10, 15, 20, 30, 40, and the Radius of the Resonance Domain Equal to 5, 10, and 15 Pixels

Neigh.	5		10		15	
t_{max}	Train	Test	Train	Test	Train	Test
5	94.20%	92.40%	91.61%	89.97%	90.00%	88.46%
10	95.27%	94.25%	92.77%	91.42%	93.48%	92.89%
15	95.71%	94.11%	93.75%	91.88%	94.64%	93.23%
20	96.16%	94.51%	93.48%	92.25%	93.66%	91.93%
25	95.45%	94.22%	93.21%	92.79%	93.66%	92.65%
30	96.16%	94.94%	93.30%	92.54%	93.93%	93.09%
40	97.14%	95.46%	92.95%	92.91%	95.98%	94.34%

IV. Conclusions

The features of FLVQ reported as means of improvement regards to SOM usability do not seem to reduce the heuristic nature of the former method with respect to the latter algorithm. On the one hand, in SOM no optimization criterion exists to choose two monotonically decreasing functions of time calculating learning rates and the update neighborhood size [13]. On the other hand, in FLVQ no optimization criterion has been found to choose the range of change of a monotonically decreasing weighting exponent employed to calculate both learning rates and the degree of overlap of neurons' receptive field.

The features of FLVQ reported as means of improvement regards to SOM performance are also questionable. Our experiments show that FLVQ performance is comparable with those of SOM employed as a clustering algorithm iff FLVQ adopts three heuristic parameter constraints (see Section III-A-3). This does not mean, however, that FLVQ and SOM can be considered as two alternative clustering algorithms. Due to their different functional properties, they should rather be assessed as complementary clustering architectures. Besides the difference between their traditional off-line and on-line implementations (a batch version of SOM exists indeed [2]), FLVQ does not process topological information whereas SOM performs fairly successfully in pattern clustering as well as in topological structure detection [6].

Several studies found in the literature focus on the need to build neural-network models more robust and easy to use for vector quantization, density function estimation, and structure detection [27]. The answer to this may come through the integration of features derived from different existing models (see Section II-D). This combination should open new perspectives in neural-network applications.

REFERENCES

[1] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[2] ——, *Self-Organizing Maps*. Berlin, Germany: Spriger-Verlag, 1995.

[3] 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.

[4] N. B. Karayiannis, J. C. Bezdek, N. R. Pal, R. J. Hathaway, and P. Pai, "Repair to GLVQ: A new family of competitive learning schemes," *IEEE Trans. Neural Networks*, vol. 7, pp. 1062–1071, 1996.

[5] B. Fritzke, "Some competitive learning methods," draft document, 1997. Available http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/DemoGNG

[6] T. M. Martinetz and K. J. Schulten, "Topology representing networks," *Neural Networks*, vol. 7, no. 3, pp. 507–522, 1994.

[7] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten, "Neural-gas network for vector quantization and its application to time-series prediction," *IEEE Trans. Neural Networks*, vol. 4, no. 4, pp. 558–569, 1993.

[8] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[9] J. Buhmann, "Learning and data clustering," in *Handbook of Brain Theory and Neural Networks*, M. Arbib, Ed. Cambridge, MA: MIT Press, 1995.

[10] K. Rose, F. Guerewitz, and G. Fox, "A deterministic approach to clustering," *Pattern Recognition Lett.*, vol. 11, no. 11, pp. 589–594, 1990.

[11] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. Series B*, vol. 39, pp. 1–38, 1977.

[12] E. C. Tsao, J. C. Bezdek, and N. R. Pal, "Fuzzy Kohonen clustering network," *Pattern Recognition*, vol. 27, no. 5, pp. 757–764, 1994.

[13] E. Erwin, K. Obermayer, and K. Schulten, "Self-organizing maps: Ordering, convergence properties, and energy functions," *Biol. Cybern.*, vol. 67, pp. 47–55, 1992.

[14] B. Bishop, M. Svensen, and C. Williams, "GTM: A principled alternative to the self-organizing map," in *Proc. Int. Conf. Artificial Neural Networks, ICANN'96*, Springer-Verlag, 1996, pp. 164–170.

[15] C. Fritzke, "A growing neural gas network learns topologies," in *Advances in Neural Information Processing Syst. 7*, G. Tesauro, D. S. Touretzky, and T. K. Leen, Eds. Cambridge, MA: MIT Press, 1995, pp. 625–632.

[16] Y. Zheng and J. F. Greenleaf, "The effect of concave and convex weight adjustments on self-organizing maps," *IEEE Trans. Neural Networks*, vol. 7, pp. 87–96, 1996.

[17] P. Blonda, V. la Forgia, G. Pasquariello, and G. Satalino, "Feature extraction and pattern classification of remote sensing data by a modular neural system," *Opt. Eng.*, vol. 35, no. 2, pp. 536–542, 1996.

[18] P. Blonda, G. Pasquariello, S. Losito, A. Mori, F. Posa, and D. Ragno, "An experiment for the interpretation of multitemporal remotely sensed images based on a fuzzy logic approach," *Int. J. Remote Sensing*, vol. 12, no. 3, pp. 463–476, 1991.

[19] R. G. Congalton, "A review of assessing the accuracy of classifications of remotely sensed data," *Remote Sensing of Environment*, vol. 37, pp. 35–46, 1991.

[20] J. C. Bezdek and N. R. Pal, "Two soft relative of learning vector quantization," *Neural Networks*, vol. 8, no. 5, pp. 729–743, 1995.

[21] N. B. Karayannis and M. Ravuri, "An integrated approach to fuzzy learning vector quantization and fuzzy c-means clustering," in *Intelligent Engineering Systems Through Artificial Neural Networks*, C. H. Dagli, M. Akay, C. L. P. Chen, B. R. Fernandez, and J. Ghosh, Eds., vol. 5. New York: Amer. Soc. Mech. Eng., 1995, pp. 247–252.

[22] N. B. Karayiannis and J. C. Bezdek, "An integrated approach to fuzzy learning vector quantization and fuzzy c-means clustering," *IEEE Trans. Fuzzy Syst*, vol. 5, pp. 622–628, 1997.

[23] N. B. Karayiannis, "Learning vector quantization: A review," *J. Smart Eng. Syst. Design*, vol. 1, pp. 33–58, 1997.

[24] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.

[25] Y. Pao, *Adaptive Pattern Recognition And Neural Networks*. Reading, MA: Addison-Wesley, 1989.

[26] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 98–110, 1993.

[27] R. N. Davè and R. Krishnapuram, "Robust clustering method: A unified view," *IEEE Trans. Fuzzy Syst.*, vol. 5, pp. 270–293, 1997.

[28] Monthly posting to the Usenet newsgroup comp.ai.neural-nets. Available ftp://ftp.sas.com/pub/neural/FAQ.

[29] M. Barni, V. Cappellini, and A. Mecocci, "Comments on 'A possibilistic approach to clustering,'" *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 393–396, 1996.

[30] N. B. Karayannis and P. Pai, "Fuzzy algorithms for learning vector quantization," *IEEE Trans. Neural Networks*, vol. 7, pp. 1196–1211, 1996.

[31] A. Baraldi and F. Parmiggiani, "Novel neural-network model combining radial basis function, competitive Hebbian learning rule, and fuzzy simplified adaptive resonance theory," in *SPIE's Opt. Sci., Eng. Instrumentation'97: Applicat. Fuzzy Logic Technol. IV*, San Diego, CA, July 1997, vol. 3165, pp. 98–112.

[32] 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.

[33] B. Bishop, M. Svensen, and C. Williams, "Magnification factors for the SOM and GTM algorithms," in *Proc. 1997 Wkshp. Self-Organizing Maps*, Helsinki, Finland.

[34] P. K. Simpson, "Fuzzy min-max neural network—Part 2: Clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 32–45, 1993.

[35] C. Fritzke, "Growing cell structures—A self-organizing network for unsupervised and supervised learning," *Neural Networks*, vol. 7, no. 9, pp. 1441–1460, 1994.

[36] P. Swain and S. Davis, *Remote Sensing: the Quantitative Approach*. New York: McGraw-Hill, 1978.

Andrea Baraldi received the doctoral degree in electronic engineering from the University of Bologna, Italy, in 1989. His doctoral thesis was on the development of satellite image classification algorithms.

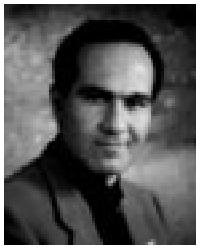
He then worked as a Research Associate at CIOC-CNR, an Institute of the National Research Council in Bologna, and as a consultant at ESA-ESRIN in Frascati, Italy, dealing with object-oriented applications for GIS. He served in the army at the Istituto Geografico Militare in Florence, working on satellite image classifiers and GIS. Since his doctoral thesis he has continued his collaboration with IMGA-CNR in Bologna, where he currently works as a Research Associate. His main interests include low-level vision processing, with special regard to texture analysis and neural-network applications.

Palma Blonda (M'93) received the laurea degree in physics from the University of Bari, Italy, in 1980.

Since 1980, her research activity has been in the area of image processing, with application to remote-sensed data. In 1984, she joined the Insitute For Signal and Image Processing (I.E.S.I.) at the Italian National Research Council (C.N.R.), Bari, Italy. She is involved in research studies on both SAR and TM data analysis for land cover mapping. She is also involved in a research project on the segmentation of magnetic resonance images with the neuroradiologists of Bari University. Her research interests include supervised and unsupervised image processing, fuzzy logic, and neural networks.

Flavio Parmiggiani was born in Campagnola E., Italy, in 1945. He received the doctoral degree in physics from the University of Milan, Italy, in 1970.

From 1970 to 1982, he worked in the field of biological cybernetics at the Italian National Research Council, Milan. From 1978 to 1980, he was with the Laboratory of Neurophysiology, University of Alberta, Edmonton, AB, Canada. In 1978, he joined a new CNR Institute, IMGA-CNR, where he has been working in the field of remote sensing and satellite image processing. He is responsible for the AVHRR receiving station installed for real-time operations at the Italian Base in Antartica. In 1992–1993, he worked at the Scott Polar Research Institute, University of Cambridge, U.K., on the problem of sea-ice dynamics using both AVHRR and ERS-1 SAR images.



Giuseppe Pasquariello received the laurea degree in computer science from the University of Bari, Italy, in 1991, with a thesis on neural networks.

In 1991, he was a “summer student” at the European Organization for Nuclear Research (CERN) in Geneva, Switzerland, for applications of neural networks to high energy physics. Since 1993, he has been with the Institute for Signal and Image Processing (IESI) of the Research National Council (CNR) in Bari, where he is working for some research projects dealing with neural networks applied to radar and medical images, remotely sensed data, and physics experimental data. His research interests include neural networks for digital image processing and data classification.

Guido Satalino received the laurea degree in physics from the University of Bari, Italy, in 1975.

In 1976, he joined the Centro Studi Applicazioni in Tecnologie Avanzate (CSATA), where he worked in the field of statistical data analysis. From 1977 to 1978, he was a scientific fellow with the commission of European Communities at the Central Bureau for Nuclear Measurements, (CBNM) Geel, Belgium, and from 1978 to 1980, he was with the National Laboratory of Frascati, Rome, of the Italian Institute of Nuclear Physics. Since 1980, he has worked on various problems in pattern recognition (from 1980 to 1985 at CSATA of Bari, and since 1985 at the Institute for Signal and Image Processing of the Italian National Research Council). His research interests include application of artificial intelligence tools and neural networks to digital image processing in the field of image understanding, remote sensing, and medical imaging.