ROBUST AND ADAPTIVE TECHNIQUES IN SELF-ORGANIZING NEURAL NETWORKS

PITAS I.[†], KOTROPOULOS C.[†], NIKOLAIDIS N.[‡], and BORŞ A.G.[†]

[†] Department of Informatics, Aristotle University of Thessaloniki, Box 451, GR-540 06 Thessaloniki, Greece
[‡] Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, GR-540 06 Thessaloniki, Greece
e-mail: {costas,nikolaid,adrian,pitas}@zeus.csd.auth.gr

Key words and phrases: Self-organizing neural networks, Kohonen's Self-Organizing feature map, Learning Vector Quantizer, Order Statistics Learning Vector Quantizers, L_2 Learning Vector Quantizer, Split-merge Learning Vector Quantizers, Median Radial Basis Function neural network.

1. INTRODUCTION

Neural networks (NN) is a rapidly expanding research field which attracted the attention of scientists and engineers in the last decade. A large variety of artificial neural networks has been developed based on a multitude of learning techniques and having different topologies [1]. One prominent class of neural networks is the Learning Vector Quantizer (LVQ) or Kohonen's Self-Organizing Feature Map [2]. Another class that is closely related to self-organizing neural networks are the Radial Basis Functions (RBF) networks.

In this paper, we shall describe robust and adaptive training algorithms that have been developed the past three years and aim at enhancing the capabilities of the self-organizing and the RBF neural networks [3]–[12].

To begin with, let us briefly describe our motivation and our research objectives. The observation that the standard LVQ does not update the winner neuron towards the optimal estimator of location (i.e., the maximum likelihood estimator of location) for the multivariate distribution of input observations motivated us to propose two variants of LVQ, namely,

1. the so called *order statistics LVQ* which relies on multivariate data ordering principles [3]–[5],
2. the L_2 *LVQ* which is based on the L_2 mean that has been proved to be the maximum likelihood estimator of the noiseless observations both in the case of pure multiplicative noise as well as in the case of signal-dependent Gaussian noise in ultrasonic images [6, 7].

The class of order statistics LVQ encompasses the following two LVQ variants: the marginal median LVQ and the vector median LVQ. The properties of these variants as well as potential applications are described in Section 2.

The L_2 LVQ is another example of self-organizing neural network design whose weight vectors correspond to the maximum likelihood estimator of the input observations instead of the arithmetic mean of these observations. The application of L_2 LVQ neural network in the segmentation of ultrasonic images and an overview of its properties is presented in Section 3.

The need for an outlier rejection mechanism in self-organizing feature maps in addition to a fusion algorithm when more than one LVQ's are trained on subsets of the training set has led us to incorporate two tests in the learning procedure of the standard LVQ [8, 9], namely a statistical test that determines if cluster splitting is statistically significant, and, additional statistical tests

that decide if cluster merging is acceptable. The proposed class of *split-merge* LVQ's is outlined in Section 4. It is shown that it possesses the following capabilities: (a) It yields an optimal number of output neurons. (b) It rejects the outliers in the formation of minimum distortion partition. (c) It enables the implementation of training parallelism.

In the area of RBF neural networks, a novel on-line learning algorithm based on robust estimators has been proposed. The so-called *Median Radial Basis Functions* neural network (MRBF) uses the marginal median LVQ in the estimation of the RBF centers [11, 12]. The Median Absolute Deviation (MAD) has been used in the estimation of the covariance matrix. A fast implementation based on data sample histogram analysis is derived for the MRBF. The properties of the MRBF neural networks and an application of the MRBF neural network in the motion field segmentation is presented in Section 5.

Finally, conclusions are drawn and future research objectives are highlighted in Section 6.

2. ORDER STATISTICS LEARNING VECTOR QUANTIZERS

Unsupervised Learning Vector Quantizer (or Kohonen's self-organizing feature map) is an autoassociative nearest-neighbor classifier which classifies arbitrary patterns into classes using an error correction encoding procedure related to competitive learning [2]. In order to make a distinction between this algorithm and the proposed LVQ variants that are based on multivariate order statistics, the LVQ algorithm will be called linear LVQ algorithm hereafter. The updating equations for the weight vectors of LVQ are given by:

$$\begin{aligned} \mathbf{w}_i(n+1) &= \mathbf{w}_i(n) + \alpha(n)[\mathbf{x}(n) - \mathbf{w}_i(n)] \;\; \forall i \in \mathcal{N}_c(n) \\ \mathbf{w}_i(n+1) &= \mathbf{w}_i(n) \;\; \forall i \notin \mathcal{N}_c(n) \end{aligned} \tag{1}$$

where $\alpha(n)$ is the adaptation step and $\mathcal{N}_c(n)$ denotes a neighborhood around the winner $\mathbf{w}_c$, i.e., the vector for which the following property holds :

$$\| \mathbf{x} - \mathbf{w}_c \| = \min_i \{ \| \mathbf{x} - \mathbf{w}_i \| \}. \tag{2}$$

It can easily be seen that the reference vector for each class $i = 1, \ldots, K$ at time $n+1$ is a linear combination of the input vectors $\mathbf{x}(j)$ $j = 0, \ldots, n$ that have been assigned to class i. Moreover, it can be shown that in the special case of only one class and the adaptation step sequence $\alpha(n) = 1/(n+1)$, the winner vector is the arithmetic mean of the observations that have been assigned to the class (i.e., the maximum likelihood estimator of location). Neither in the case of multiple classes that are normally distributed nor in the case of non-Gaussian multivariate data distributions the linear LVQ is the optimal estimator of the cluster means. In general, linear LVQ and its variations suffer from the following drawbacks: (i) They do not use optimal estimators for obtaining the reference vectors $\mathbf{w}_i$, $i = 1, \ldots, K$ that match the probability density function (pdf) of each class. (ii) They do not have robustness against erroneous choices for the winner vector, since it is well known that linear estimators have poor robustness properties [13]. (iii) They do not have robustness against outliers that may exist in the vector observations.

In order to overcome these problems, we propose Learning Vector Quantizer variants that are based on multivariate order statistics [14]. It is well known that there is no unambiguous, universally agreeable total ordering of N p-variate samples $\mathbf{x}_1, \ldots, \mathbf{x}_N$ where $\mathbf{x}_i = (x_{1i}, x_{2i}, \ldots, x_{pi})^T$, $i = 1, \ldots, N$. The following so-called sub-ordering principles are discussed in [14]: *marginal ordering, reduced (aggregate) ordering, partial ordering*, and *conditional (sequential) ordering*. In our experiments we have used the marginal and the reduced subordering principles. In marginal ordering, the

multivariate samples are ordered along each one of the p-dimensions independently, i.e.,:

$$x_{i(1)} \leq x_{i(2)} \leq \ldots \leq x_{i(N)} \quad i = 1, \ldots, p. \tag{3}$$

The marginal median has the following definition:

$$\mathbf{x}_{med} = \begin{cases} \left(x_{1(\nu+1)}, \ldots, x_{p(\nu+1)}\right)^T & \text{for } N = 2\nu + 1 \\ \left(\frac{x_{1(\nu)}+x_{1(\nu+1)}}{2}, \ldots, \frac{x_{p(\nu)}+x_{p(\nu+1)}}{2}\right)^T & \text{for } N = 2\nu. \end{cases} \tag{4}$$

It can be used in the following way in order to define the *marginal median LVQ*. Let us denote by $\mathbf{x}(n)$ the current observation and by $\mathbf{X}_i(n)$ the set of the vector observations that have been assigned to each class i, $i = 1, \ldots, K$ until time $n-1$. We find at time n the winner vector $\mathbf{w}_c(n)$ that minimizes $\| \mathbf{x}(n) - \mathbf{w}_i(n) \|$, $i = 1, \ldots, K$. The marginal median LVQ (MMLVQ) updates the winner reference vector as follows:

$$\mathbf{w}_c(n+1) = \text{median} \left\{ \mathbf{x}(n) \cup \mathbf{X}_c(n) \right\}. \tag{5}$$

The median operation is given by (4). Thus, all past class assignment sets $\mathbf{X}_i(n)$, $i = 1, \ldots, K$ are needed for MMLVQ. MMLVQ needs the calculation of the median of data sets of ever increasing size, as can be seen from (5). This may pose severe computational problems for relatively large n. However, for integer-valued data, a modification of the *running median algorithm* proposed by Huang et *al.* [15] can be devised to facilitate median calculations by exploiting the fact that the marginal median of the already assigned samples $\mathbf{X}_i(n)$ is known.

Another definition of the multichannel median (based on R-ordering principles) is the so-called *vector median*. The vector median is the observation that has the minimum sum of distances from all the remaining observations, i.e.:

$$\sum_{i=1}^{N} |\mathbf{x}_i - \mathbf{x}_{med}| \leq \sum_{i=1}^{N} |\mathbf{x}_i - \mathbf{x}_j| \quad j = 1, \ldots, N. \tag{6}$$

The *vector median* LVQ (VMLVQ) uses the following formula to update the winner vector $\mathbf{w}_c(n)$ at step n:

$$\mathbf{w}_c(n+1) = \text{vector median} \left\{ \mathbf{x}(n) \cup \mathbf{X}_c(n) \right\} \tag{7}$$

where $\mathbf{X}_i(n)$ is again the set of vector-valued observations that have been assigned to class i. The vector median operator in the previous expression is the one defined in (6). Vector median LVQ keeps tract of all its history and therefore all data samples have equal contribution to the reference vector update procedure. In the case of non-stationary data, we can evaluate the vector median using a moving window to discard the older samples as new observations become available.

The expected stationary state of the MMLVQ and the VMLVQ has been derived and compared to the expected stationary state of the linear LVQ [4, 5]. Both MMLVQ and VMLVQ have been proved robust against outliers. Moreover, they perform well in cases where overlapping clusters exist. Potential applications of the order-statistics LVQ's in noisy color image quantization have been reported in [3].

A situation frequently encountered in industrial computer vision applications is color-based recognition of objects having a simple shape. The following experiment provides strong evidence of the superiority of MMLVQ in such applications. Let us suppose that each of the five rectangles in the synthetic image presented in Figure 1a (of dimensions 256×256) corresponds to an object having a distinct and a priori known color. The RGB triplets of the five objects can be seen in Table 1. It is worth noting that the five colors are very close to each other in the RGB color space. Let us

Original			MMLVQ			LVQ		
R	G	B	R	G	B	R	G	B
136	124	160	137	123	169	134	121	162
144	128	160	146	130	156	134	121	162
136	100	160	136	99	159	134	121	162
144	120	128	143	119	126	143	120	132
120	124	160	123	124	157	134	121	162

Table 1: RGB coordinates of the five objects along with the corresponding output reference vectors for the MMLVQ and LVQ networks.

also suppose that the five objects present in the image are parts of a larger set of n objects (9 in our case). This image has been corrupted, independently on each channel, by adding mixed zero-mean white Gaussian noise having $\sigma = 9$ and impulsive noise having probability of impulses (both positive and negative ones) 4%. The noisy image can be seen in Figure 1b. The MMLVQ and the LVQ have been applied on the noisy image. Ten output classes have been used (9 for the objects and one for the background) and the reference vectors for the 10 classes have been initialized using the a priori known colors of the objects and the background. The training set consists of 16384 randomly selected pixels. Various adaptation steps for the LVQ algorithm have been tried out in order to find the one that gives the best results. The recall images for the two algorithms are presented in Figures 1c, 1d. It can be seen that the MMLVQ algorithm succeeds to distinguish between the five objects, assigning one output class to each object. The output reference vectors for the classes that correspond to the five objects are very close to the real colors of the object, as can be seen in Table 1. On the other hand, it is clearly seen that LVQ is susceptible to noise, because it assigns four out of five objects to the same output class while the rest of the classes are dominated by noise.

3. L_2 LEARNING VECTOR QUANTIZER

The derivation of the LVQ variant to be described in this section has been driven by the need for accurately segmenting ultrasonic images into several tissue classes. Ultrasonic images suffer from a special kind of noise called *speckle*. Speckle is an interference effect caused by ultrasound (US) beam scattering from microscopic tissue inhomogeneities. Ultrasonic speckle can be modeled as multiplicative Rayleigh distributed noise or as signal-dependent Gaussian noise. The first model refers to envelope-detected US B-mode data. The second model describes more accurately ultrasonic images where the displayed image data have undergone excessive manipulation (e.g. logarithmic compression, low and high-pass filtering, postprocessing, etc.). In the case of pure multiplicative Rayleigh speckle, it has been proved that the maximum likelihood (ML) estimator of the original (noiseless) signal is the L_2 mean [16]. Furthermore, for signal-dependent Gaussian speckle, it has been shown that the ML estimator closely resembles the L_2 mean [6]. These observations motivated us to modify the standard LVQ algorithm so that the reference vectors correspond to the L_2 mean instead of the sample arithmetic mean. Such a modification will provide more accurate reference vectors for each Voronoi neighborhood and will result in a better segmentation of both ultrasonic B-mode data as well as displayed US image data. Accordingly, in this application we are interested in the vectors of *squared weights* and *squared observations*. The winner vector is determined by comparing the Euclidean distances between the vector of squared observations and the vectors of squared weights similarly to (2). Let us denote by $\mathbf{w}_i'$ the $(p \times 1)$ vector having as elements the weights comprising the reference vector $\mathbf{w}_i$ squared, i.e., $\mathbf{w}_i' = (w_{i1}^2, w_{i2}^2, \ldots, w_{ip}^2)^T$ and by $\mathbf{x}'$ the

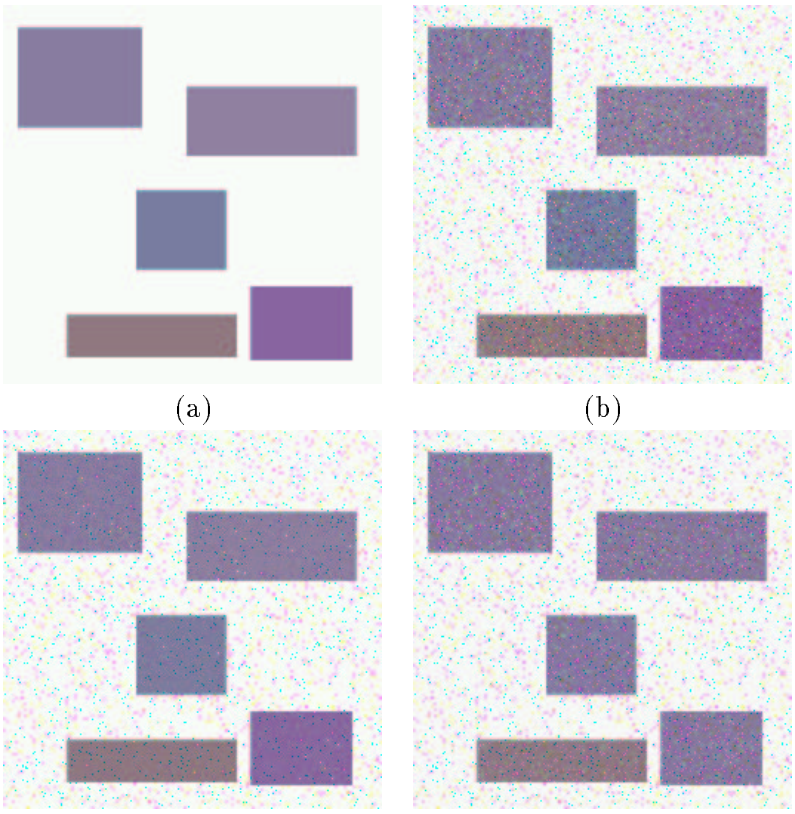

Fig. 1: Application of linear LVQ and marginal median LVQ in color-based object recognition in the presence of mixed additive Gaussian and impulsive noise. (a) Original image. (b) Noisy image used in the learning phase. (c) MMLVQ recall image. (d) Linear LVQ recall image.

vector $\mathbf{x}' = (x_1^2, x_2^2, \ldots, x_p^2)^T$. The updating equations of the L_2 LVQ are given by:

$$\begin{aligned} \mathbf{w}_i'(n+1) &= \mathbf{w}_i'(n) + \alpha(n)[\mathbf{x}'(n) - \mathbf{w}_i'(n)] \;\; \forall i \in \mathcal{N}_c(n) \\ \mathbf{w}_i'(n+1) &= \mathbf{w}_i'(n) \;\; \forall i \notin \mathcal{N}_c(n) \end{aligned} \tag{8}$$

A deep insight into the performance of this LVQ variant is obtained by the study of its convergence properties. Two types of convergence, namely, the convergence in the mean and in the mean square are examined for continuous time t. L_2 LVQ network *converges in the mean*, if the average vector of squared weights converges to the expected stationary state of the network as t approaches infinity. L_2 LVQ network *converges in the mean square*, if the trace of the correlation matrix of the squared weight error-vectors tends to zero or remains bounded as t approaches infinity. The case of a constant adaptation step $\alpha(t) = \alpha$ is considered for mathematical simplicity. Generalization for the optimal adaptation step sequence $\alpha(t) = 1/t$ is treated in [6]. We shall confine ourselves to the analysis of a single-winner L_2 LVQ network, i.e., $\mathcal{N}_c(t) = \{c\}$. Bounds on the overall time constant for any squared weight and on the trace of the correlation matrix of the squared weight error-vectors are derived. For example, the overall time constant τ_a for any average squared weight can be bounded as follows [6]:

$$\frac{1}{\alpha\lambda_{\max}} \leq \tau_a \leq \frac{1}{\alpha\lambda_{\min}} \tag{9}$$

Method	$P_F(\%)$	$P_D(\%)$	Threshold	$\hat{P}_D(\%)$
Image thresholding	13.04	29.34	24	31.99
	15.19	32.18	23	
median 7×7	14.85	37.85	20	38.13
	18.88	43.33	19	
arithmetic mean	13.78	38.95	20	41.28
7×7	17.59	45.90	19	
L_2 mean	13.79	40.05	19	42.39
7×7	17.85	47.55	18	
L_2 LVQ NN	15.06	59.07	-	59.07

Table 2: Figures of Merit for Lesion Detection on a Simulated US B-Mode Image.

where $\lambda_{\min}$ and $\lambda_{\max}$ denote the smallest and largest eigenvalue of $(Kp \times Kp)$ matrix $\mathbf{B}$ given by:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} & \cdots & \mathbf{B}_{1K} \\ \vdots & & \ddots & \vdots \\ \mathbf{B}_{K1} & \mathbf{B}_{K2} & \cdots & \mathbf{B}_{KK} \end{bmatrix}. \tag{10}$$

Each $\mathbf{B}_{kl}$ $k, l = 1, \ldots, K$ is a $(N \times N)$ square submatrix with mn-element given by:

$$[\mathbf{B}_{kl}(\overline{\mathbf{W}'})]_{mn} = \left[w_{km}^2 \frac{\partial}{\partial w_{ln}^2} \hat{F}_k(\mathbf{W}') + \hat{F}_k(\mathbf{W}')\delta(k-l, m-n) - \frac{\partial}{\partial w_{ln}^2} \int_{\mathcal{V}_k(\mathbf{W}')} x_m^2 f(\mathbf{x}) d\mathbf{x} \right]_{\mathbf{W}'=\overline{\mathbf{W}'}} \tag{11}$$

where $\hat{F}_k(\mathbf{W}') = \int_{\mathcal{V}_k(\mathbf{W}')} f(\mathbf{x})d\mathbf{x}$, $\delta(k-l, m-n)$ is the 2-D Kronecker-delta function and $\overline{\mathbf{W}'}$ denotes the stationary state of the network. The convergence in the mean square can be studied in a similar way. Due to lack of space the interested reader is referred to [6]. The ability of the L_2 LVQ to segment ultrasonic images in classes representing various tissue and lesion characteristics is combined with signal-adaptive filtering in order to allow preservation of image edges and details as well as maximum speckle reduction in homogeneous regions [6]. The design of filtering processes combining segmentation and optimum L-filtering, and their use for speckle noise suppression in ultrasonic images is pursued in [7]. The proposed neural network has been applied both to simulated US B-mode data as well as to displayed US image data for image segmentation. Due to lack of space only the simulations that have been performed on a simulated image showing an homogeneous tissue of size 4 cm $\times$ 4 cm with a lesion in the middle of diameter 2 cm is discussed. We have compared the performance of the several strategies tabulated in Table 2 using the probability of detection (P_D) and the probability of false alarm (P_F) as figures of merit. The comparison is based on the probability of detection $\hat{P}_D$ which has been calculated by linearly interpolating between the experimental values of probabilities of detection that correspond to the two probabilities of false alarm that are closest to the one of L_2 LVQ. It is seen that an almost 16.7% higher probability of detection is obtained by using the L_2 LVQ NN instead the L_2 mean filter of dimensions 7×7.

4. A CLASS OF SPLIT-MERGE LEARNING VECTOR QUANTIZERS

In this section, a split-merge Learning Vector Quantizer is described. As its name suggests, the proposed algorithm employs split-merge tests. In general, when the whole training set is presented in the input of LVQ for the first time, many wrong decisions are expected, because the winner

vectors fail to approximate adequately the true cluster means. Therefore, a need for testing further the correctness of the classification of the input training patterns to the cluster represented by the winner is recognized. Similarly, when a training pattern moves from one cluster to another, an additional test is needed to approve the correctness of such a decision. In the later case, the cluster where the input training pattern was formerly classified to, may be considered as an unstable one, because it has been affected by outliers. Consequently, during the session when a pattern removal has occurred, we have decided to check further if the classification of input training patterns to this unstable cluster on the basis of the Euclidean distance metric (2) is still correct. The outline of the split-merge LVQ learning algorithm is as follows.

1. For each pattern presentation $\mathbf{x}(n)$:

   a. Find the winner $\mathbf{w}_c(n)$.

   b. Test if $\mathbf{x}(n)$ is outlier to the patterns that are represented by $\mathbf{w}_c(n)$ for : (i) pattern presentations during the first session, (ii) patterns that are moved from one cluster to another, and, (iii) patterns of a cluster where a removal of a pattern has occurred during the session this modification took place.

   c. If $\mathbf{x}(n)$ is not an outlier, proceed as in standard LVQ.

   d. If $\mathbf{x}(n)$ is an outlier, examine if the cluster represented by the winner can be split in two subclusters, and test possible inclusion of $\mathbf{x}(n)$ in one of the resulting subclusters. Otherwise, create a new cluster having seed $\mathbf{x}(n)$.

2. When the training set is exhausted, test the integrity of the cluster associated with each output neuron.

3. Repeat steps 1-2, until convergence is attained.

The criteria used in steps 1.b. and 1.d have been described in detail in [8, 9].

Furthermore, a novel two-layer LVQ architecture which incorporates second-order statistics in its training phase and allows training parallelism by splitting patterns into groups has been introduced. It is comprised of L LVQ networks that are trained independently in the first layer and a single LVQ network in the second layer. The training patterns of the first layer LVQ's are input patterns. The training patterns of the second layer LVQ are the weight vectors of the first layer LVQ's after their convergence. The second layer classifies the weight vectors provided by the L networks of the first layer. Let us suppose that the L LVQ's of the first layer classify p-dimensional data into N-many classes, then the second layer LVQ has p input nodes and $L \times N$ output nodes at most. Some of the weight vectors of the first layer LVQ's have been trained by patterns extracted from the same population, therefore they must be merged. Some others are reference vectors associated with different populations, therefore they must be preserved. The incorporation of homogeneity and proximity statistical tests based on second-order statistics in the second layer LVQ learning algorithm is proposed so that the second layer LVQ can group the partial results provided by the first-layer LVQ's in order to provide the final weight vectors. The final weight vectors are the reference vectors that represent the whole training set. Furthermore, the proposed two-layer LVQ architecture is easily parallelized.

The LVQ network of the second layer is used to find the weight vectors provided by the first layer LVQ's which are candidates for merging. As has already been discussed, the criterion of minimum Euclidean distance metric used in the LVQ is not sufficient for the above-described task, because it does not take into account the presence of outliers. Consequently, additional tests must be implemented in order to test the similarity between the weight vector provided by the first layer LVQ's and the winner vector determined by the second layer LVQ. The homogeneity of the winner

Neural Network	Number of neurons	Learning MSE	Recall MSE	PSNR (dB)	Iterations
Standard LVQ random initialization	256	146.369	91.35	28.523	560
Standard LVQ initialization of LBG	256	51.866	44.863	31.611	572
Single-layer Split-Merge LVQ	256	48.795	44.137	31.682	562
Two-layer Split-Merge LVQ	241	73.987	52.183	30.955	529 (1st FL LVQ) 546 (2nd FL LVQ) 18 (SL LVQ)

Table 3: Figures of merit for color image quantization (FL:First Layer, SL:Second Layer).

vectors evaluated by the LVQ in the second layer and the input weight vectors provided by the LVQ's of the first layer can be tested by employing statistical tests on the mean vectors and on the covariance matrices as well. The interested reader is referred to [10].

The proposed algorithms have been applied to color image quantization. The performance of the proposed split-merge LVQ's and the standard LVQ algorithm in color image quantization is summarized in Table 3. The number of output neurons for the standard LVQ NN is set to 256. The performance of the standard LVQ algorithm depends strongly on the initialization procedure that is employed. Two different initialization procedures have been used: (a) random initialization and (b) the initialization of LBG algorithm. By inspecting Table 3, it is seen that the proposed single-layer split-merge LVQ achieves a slightly better performance than a standard LVQ algorithm that uses the same initialization with LBG, but with one fundamental difference: The split-merge LVQ algorithm *has found* the number of clusters present in the input training set, while the standard LVQ has been initialized in an optimal way for 256 output neurons. The two-layer split-merge LVQ architecture is expected to give the best result when there is strong correlation (e.g. overlap) between the training subsets. It seems that this is not the case in our experiment. If the two single-layer split-merge LVQ's in the first layer were trained in parallel, then the two-layer split-merge LVQ architecture would provide almost identical results with a (single-layer) split-merge LVQ, but in half computation time.

5. MEDIAN RADIAL BASIS FUNCTION NEURAL NETWORK

The RBF network has a feed-forward topology which models a mapping between a set of vector entries and a set of outputs, by using an intermediate level of representation implemented by the radial basis functions. Each network input is assigned to a vector entry (feature in a pattern recognition application) and the outputs correspond either to a set of functions to be modeled by the network or to various associated classes.

In supervised learning, the network is provided with a training set of patterns consisting of vectors and their corresponding classes. Each pattern is assigned to one class C_k only, according to an unknown mapping. After an efficient learning stage, the network implements the mapping rule and generalizes it for patterns that do not belong to the training set. According to Bayes theorem we can express the relation among the a posteriori probabilities $P(C_k|\mathbf{x})$ of different classes by using

their a priori probabilities $P(C_k)$:

$$P(C_k|\mathbf{x}) = \max_{j=1}^{K} P(C_j|\mathbf{x}) \tag{12}$$

$$p_k(\mathbf{x}) = P(C_k)\, p(\mathbf{x}|C_k) = \max_{j=1}^{K} [P(C_j)\, p(\mathbf{x}|C_j)] \tag{13}$$

where K is the number of classes. Due to their approximation capabilities, RBF networks can be used to describe the underlying probability as a sum of components with respect to a base (denoted by the function family ϕ) :

$$p_k(\mathbf{x}) = \sum_{j=1}^{L} \lambda_{k,j} \phi_j(\mathbf{x}) \tag{14}$$

where L is the number of kernel functions and $\lambda_{k,j}$ are the hidden unit to output weights.

Each hidden unit implements a kernel function. We have chosen the Gaussian function as the kernel function

$$\phi_j(\mathbf{x}) = \exp\left[-(\mathbf{w}_j - \mathbf{x})^T\, \mathbf{\Sigma}_j^{-1} (\mu_j - \mathbf{x})\right] \tag{15}$$

for $j = 1, \ldots, L$, where $\mathbf{w}_j$ is the mean vector and $\mathbf{\Sigma}_j$ is the covariance matrix. Geometrically, $\mathbf{w}_j$ represents the center and $\mathbf{\Sigma}_j$ the shape of the j-th basis function. A hidden unit function can be represented as a hyper-ellipsoid in the N-dimensional space. As can be seen in (15), an activation region is defined around the mean vector. If a pattern falls inside the activation region of a hidden unit, that neuron will fire.

The properties of Radial Basis Functions (RBF's) make them suitable for modelling probability density functions (*pdf*'s) in nonparametric classification tasks. The Gaussian centers correspond to the local estimates for the first order statistics and covariance matrices for the second order statistics.

A combined unsupervised-supervised learning technique is employed in order to estimate the RBF weights. The classical approach consists of an on-line technique which employs the LVQ algorithm [2] in order to find the hidden unit centers, in the unsupervised part as provided in (1). In order to chose the hidden-unit center to be updated we use either the Euclidean distance as in (2) or the Mahalanobis distance which takes into account the respective covariance matrix. In the latter case (2) translates into :

$$(\mathbf{w}_c - \mathbf{x}(n))^T\, \mathbf{\Sigma}_c^{-1} (\mathbf{w}_c - \mathbf{x}(n)) = \min_{k=1}^{L} [(\mathbf{w}_k - \mathbf{x}(n))^T\, \mathbf{\Sigma}_k^{-1} (\mathbf{w}_k - \mathbf{x}(n))] \tag{16}$$

For the covariance matrix the classical sample deviation estimate has been used :

$$\begin{aligned} \mathbf{\Sigma}_j(n+1) &= \mathbf{\Sigma}_j(n) + \alpha(n)[\mathbf{x}(n) - \mathbf{w}(n)]^T [\mathbf{x}(n) - \mathbf{w}(n)] \;\; \forall i \in \mathcal{N}_c(n) \\ \mathbf{\Sigma}_j(n+1) &= \mathbf{\Sigma}_j(n) \;\; \forall i \notin \mathcal{N}_c(n) \end{aligned} \tag{17}$$

We propose a robust statistics based training algorithm. The network resulting from training based on robust statistics is called Median Radial Basis Function (MRBF) neural network [11]. In the first stage we employ the MMLVQ algorithm for evaluating the RBF's centers (5) and the Median of the Absolute Deviations (MAD) [13] for estimating the covariance matrix associated with each Gaussian function. The MAD-based estimation of the dispersion parameter is provided by :

$$\sigma_j = \frac{\text{med}\, \{|\mathbf{x}(0) - \mathbf{w}_j|, \ldots, |\mathbf{x}(n-1) - \mathbf{w}_j|\}}{0.6745} \tag{18}$$

where 0.6745 is a scaling parameter in order to make the estimator Fisher consistent for the normal distribution [13]. The off-diagonal components of the covariance matrix can be calculated using

robust statistics. We consider two arrays containing the difference and sum of each two different components for a data sample from the moving window

$$Z^{+}_{i,hl} = x_h(i) + x_l(i), \quad Z^{-}_{i,hl} = x_h(i) - x_l(i). \tag{19}$$

First, the median of these new data populations is calculated according to (5). The squares of the correspondent MAD estimates (18) for the arrays $\mathbf{Z}^{+}_{hl}$ and $\mathbf{Z}^{-}_{hl}$ represent their variances and they are denoted as $V^{+}_{j,hl}$ and $V^{-}_{j,hl}$. The off-diagonal components of the covariance matrix are derived as :

$$\sigma^2_{j,hl} = \sigma^2_{j,lh} = \frac{1}{4}\,(V^{+}_{j,hl} \; - \; V^{-}_{j,hl}\,). \tag{20}$$

The second layer is used in order to group the clusters found in the unsupervised stage in classes. The output weights are updated as follows :

$$\lambda_{k,j}(t+1) = \lambda_{k,j}(t) + \eta_\lambda \, (F_k(\mathbf{x}) - Y_k(\mathbf{x}))Y_k(\mathbf{x})(1 - Y_k(\mathbf{x}))\phi_j(\mathbf{x}) \tag{21}$$

for $k = 1, \ldots, K$ and $j = 1, \ldots, L$, where the learning rate is $\eta_\lambda \in (0, 1]$. $F_k(\mathbf{x})$ is the desired output for the pattern vector $\mathbf{x}$. The formula (21) corresponds to the backpropagation for the output weights of an RBF network with respect to the mean squared error cost function.

A fast implementation algorithm based on histogram modelling as well as a theoretical comparative study of the MRBF and RBF networks is provided in [11]. The bias obtained after robust statistics-based learning algorithms like MMLVQ and MAD in estimating the parameters of a mixture of Gaussian functions is smaller compared to the classical statistics-based estimation.

In [11] the MRBF network was applied for optical flow smoothing in the "Hamburg taxi" image sequence. The MRBF network had only two inputs and provided better motion vectors estimates when compared with the RBF network. This scheme was extended in [12] in order to approximate the optical flow and moving object probabilities. The image is partitioned in blocks situated on a rectangular lattice. Each block site is associated with a five-dimensional feature vector describing the position, the gray level and the local motion information. The classification is done according to a decision criterion derived from the Bayesian theory and representing a metric in the parameter space [12]. The moving scene is split accordingly in moving regions. We consider an MRBF network for modeling the optical flow and moving object segmentation in the image sequence. This structure is embedded in a two-layer feed-forward neural network, where each output is assigned to a moving object. The mixture of basis functions approximate the probabilities associated to the optical flow estimation and segmentation of the moving objects. The number of moving objects does not need to be specified *a priori*. It is found according to a compactness measure.

Numerical comparisons between the MRBF network and the Iterated Conditional Modes (ICM) [17], a widely used optical flow smoothing algorithm, applied in "Hamburg taxi" sequence are provided in Table 4. The comparison criteria for the training frame are the optical flow mean absolute error (MAE), mean square error (MSE), the misclassification error, the number of parameters required by each algorithm in order to estimate the optical flow and segment the motion, and the necessary number of iterations in order to achieve the convergence. The MRBF network trained using data samples drawn from the first frame was applied on frames 2-9 of the "Hamburg taxi" sequence. The average classification error and time per frame (including the training time for the MRBF) evaluated on a Silicon Graphics Indy Workstation are provided in Table 4.

In Figure 2a a frame from the "Hamburg taxi" sequence is shown. The MRBF learning algorithm is applied for the given data. The segmented moving objects by means of the MRBF network trained on a 4×4 pixel block partition are shown in Figure 2b. Results obtained by applying the above

Algorithm	Training Frame					Imag e Sequence	
	Class. error (%)	MAE	MSE	No of Param.	No of Iter.	Class. error (%)	Process. time/frame (s)
MRBF	3.02	0.17	0.85	210	13	3 .55	8.35
ICM	4.07	0.33	1.15	9216	23	8.70	6.61

Table 4: Comparison between MRBF network and ICM when applied in the "Hamburg taxi" image sequence.

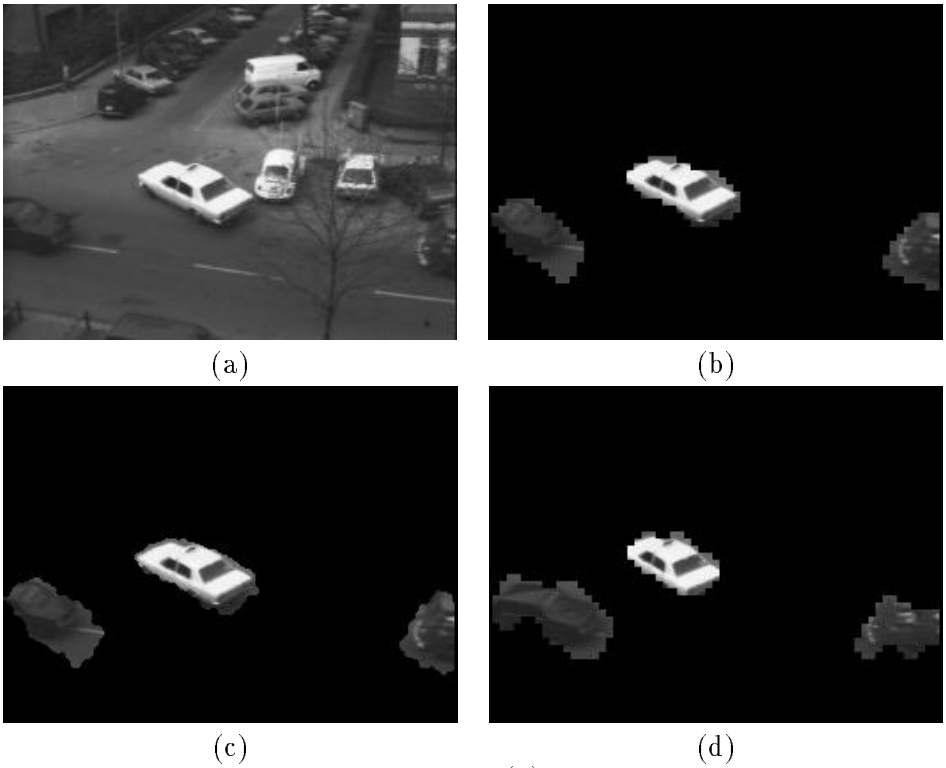

Fig. 2: (a) Frame from the "Hamburg taxi" sequence, (b) The segmentation of the moving objects based on the MRBF network, (c) The MRBF network based segmentation when applied at pixel level partition in the "Hamburg taxi" sequence, (d) Moving object segmentation in the 8th frame obtained when applying the network trained with data samples drawn from the first and third frames.

network on a 1×1 pixel partition of the same image sequence (after scaling the location features) , in a hierarchical approach can be seen in Figure 2c. It is obvious that the "taxi" and "van" moving objects are better segmented in this case. The segmentation results obtained when the MRBF network trainned on the first and third frames is applied on the eighth frame of the same image sequence can be seen in Figure 2d. This result shows the network capability to embed in its weights the parameters associated with the moving objects. The above results as well as other results provided in [11, 12] ilustrate the capababilities of the MRBF network.

6. CONCLUSIONS-FUTURE RESEARCH

A number of new training techniques that can robustify and enhance RBF and self-organizing neural networks have been presented in this paper. These techniques include the Order Statistics LVQ, L_2 LVQ, Split-Merge LVQ and Median Radial Basis Function Neural Networks. Simulations that prove the superiority of the proposed variants in various applications (noisy color image quantization, color-based object recognition, segmentation of ultrasonic images, motion field smoothing and

moving object segmentation) have been also presented. However, the list of potential applications is not limited to the above topics. Future research topics include MMLVQ-based artifact rejection on multichannel biomedical signals and RBF-based modelling of 3-D objects.

REFERENCES

1. HAYKIN S., *Neural Networks: A Comprehensive Foundation*. Macmillan Publ. Co., Englewood Cliffs, N.J. (1994).
2. KOHONEN T.K., *Self-Organization and Associative Memory*, 3rd ed. Springer-Verlag, Berlin, Heidelberg (1989).
3. PITAS I., KOTROPOULOS C., NIKOLAIDIS N., YANG R. and GABBOUJ M., Order Statistics Learning Vector Quantizer, *Special Issue on Nonlinear Image Processing, IEEE Trans. on Image Processing* **6**, 1048–1056 (1996).
4. PITAS I., KOTROPOULOS C., NIKOLAIDIS N., YANG R. and GABBOUJ M., A class of order statistics Learning Vector Quantizers, *Proc. of the IEEE Int. Symposium on Circuits and Systems (ISCAS 94)*, 387–390 (1994).
5. KOTROPOULOS C., PITAS I. and. GABBOUJ M., Marginal Median Learning Vector Quantizer, *Proc. of the European Signal Processing Conference (EUSIPCO '94)*, 1496–1499 (1994).
6. KOTROPOULOS C., MAGNISALIS X., PITAS I. and STRINTZIS M.G., Nonlinear ultrasonic image processing based on signal-adaptive filters and self-organizing neural networks, *IEEE Trans. on Image Processing* **3**, 65–77 (1994).
7. KOFIDIS E., THEODORIDIS S., KOTROPOULOS C. and PITAS I., Segmentation-based L-filtering of speckle-noise in ultrasonic images, *Proc. SPIE Symposium on Electronic Imaging Science and Technology: Nonlinear Image Processing V* (Chairs: DOUGHERTY E.R., ASTOLA J. and LONGBOTHAM C.H.) **2180**, 280–288 (1994).
8. KOTROPOULOS C. and PITAS I., A variant of Learning Vector Quantizer based on split-merge statistical tests, *Lecture Notes in Computer Science: Computer Analysis of Images and Patterns* (Edited by CHETVERIKOV D. and KROPATSCH W.G.), 822–829, Springer Verlang (1993).
9. KOTROPOULOS C. and PITAS I., Split-merge Learning Vector Quantizer algorithm, *Proc. of the European Conference on Circuit Theory and Design (ECCTD '93)*, 465–468 (1993).
10. KOTROPOULOS C., AUGE E. and PITAS I., Two layer Learning Vector Quantizer for color image quantization ,*Signal Processing VI: Theories and Applications*, 1177–1180, Elsevier (1992).
11. BORŞ A. G., and PITAS I., Median radial basis functions neural network, *IEEE Trans. on Neural Networks*, vol 7, no. 6, Nov. 1996.
12. BORŞ A. G., and PITAS I., Optical flow estimation and moving object segmentation based on median radial basis function network, accepted at *IEEE Trans. on Image Processing*.
13. PITAS I. and VENETSANOPOULOS A.N., *Nonlinear Digital Filters: Principles and Applications*. Kluwer Academic, Publishers Hingham, MA (1990).
14. BARNETT V., The ordering of multivariate data *J. R. Statist. Soc. A* **139:3**, 318–354 (1976).
15. HUANG T.S., YANG G.J. and TANG G.Y., "A fast two-dimensional median filtering algorithm," *IEEE Trans. on Acoustics, Speech and Signal Processing* **27**, 13–18 (1979).
16. KOTROPOULOS C. and PITAS I., Optimum nonlinear signal detection and estimation in the presence of ultrasonic speckle, *Ultrasonic Imaging* **14**, 249–275, (1992).
17. F. HEITZ and P. BOUTHEMY, Multimodal estimation of discontinuous optical flow using Markov Random Fields, *IEEE Trans. on Pattern Anal. and Machine Intel.,* vol. PAMI-15, No. 12, pp. 1217-1232, Dec. 1993.