SIGNAL PROCESSING VI: Theories and Applications
J. Vandewalle, R. Boite, M. Moonen, A. Oosterlinck (eds.)
© 1992 Elsevier Science Publishers B.V. All rights reserved

TWO-LAYER LEARNING VECTOR QUANTIZER FOR COLOR IMAGE QUANTIZATION

C. Kotropoulos E. Augé I. Pitas

Department of Electrical Engineering, University of Thessaloniki,
Thessaloniki 540 06, GREECE

A novel two-layer LVQ architecture is proposed which incorporates second order statistics in its training phase and allows training parallelism by splitting patterns into groups. The learning and recall procedures for the LVQ networks in the first and second layer are described. The proposed algorithm is based on statistical tests on the mean vectors and the dispersion matrices. A simplification based on a proximity test is presented. The application of the two-layer LVQ to color image quantization is also discussed.

1. INTRODUCTION

Neural networks is a rapidly expanding research field which attracted the attention of engineers and scientists in the last decade. One of the most prominent neural networks in the literature is the Learning Vector Quantizer (LVQ) [1,2]. It is an autoassociative nearest-neighbor classifier which classifies arbitrary patterns into p-many classes using an error-correction encoding procedure related to the competitive learning. It has already found extensive application for phoneme recognition [3,4] as well as in image processing, control and combinatorial optimization [2]. The dynamic weighting of input signals and a definition of neighborhood in the LVQ by a minimal spanning tree are proposed in [5]. The modification of the LVQ by using a differential competitive learning algorithm is discussed in [6].

The main contribution of this paper is the design of a novel two-layer LVQ architecture which incorporates second order statistics in its training phase and allows training parallelism by splitting patterns into groups. The proposed two-layer LVQ architecture is shown in Figure 1. It is comprised of N LVQ networks working 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 after the convergence of the first layer LVQ's. The second layer classifies the weight vectors provided by the N networks of the first layer. Let us suppose that the N LVQ's of the first layer classify q-dimensional data into p-many classes, then the second layer LVQ has q input nodes and $N \times p$ output nodes at most. Some of them 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 in order to group partial results provided by the first-layer LVQ's and considered in the evaluation of the final winner vectors. Thus, the proposed learning algorithm takes into account the presence of outliers and provides more accurate reference vectors for the clusters presented in the input patterns. Furthermore, the proposed two-layer LVQ architecture is easily parallelized and consequently makes faster computationally intensive tasks such as color image quantization and segmentation.

2. TWO-LAYER LVQ ALGORITHM

In the following, the learning and recall procedures of the first and second layer LVQ's are described. The learning procedure of each first layer LVQ is the typical one of a multiple winner LVQ [2,3]. The recall procedure of each LVQ network in the first layer is applied only to the patterns used for the training of this network. It provides the necessary information about the sample mean vector and the sample dispersion matrix of the classes produced at the output of the network. Let $\mathbf{V}_l = (v_{1l}, \dots, v_{ql})^T$ $l = 1, \dots, p$ be the weight vectors for a first layer LVQ. The recall procedure of a network in the first layer has the following steps:

1. Initialize the sample mean vector $\mathbf{m}_j$, the sample dispersion matrix $\mathbf{S}_j$ and the number of patterns n_j associated with each class.

$$\mathbf{m}_j(0) = \mathbf{0}_{q\times 1} \quad \mathbf{S}_j(0) = \mathbf{0}_{q\times q} \quad n_j(0) = 0 \quad j = 1, \dots, p \tag{1}$$

2. Determine the class C_g represented by the weight vector $\mathbf{V}_g$ to which the training pattern $\mathbf{X}(k)$ is most closely associated with.

$$\mathbf{X}(k) \in C_g \text{ if } \|\mathbf{X}(k) - \mathbf{V}_g\| = \min_{j=1}^{p}\{\|\mathbf{X}(k) - \mathbf{V}_j\|\} \quad (2)$$

3. Increment the number of patterns belonging to C_g by one and update the sample mean vector and the sample dispersion matrix of this class.

$$\begin{aligned}
n_g(k) &= n_g(k-1)+1 \\
\mathbf{d}_g(k) &= \mathbf{X}(k) - \mathbf{m}_g(k-1) \\
\mathbf{m}_g(k) &= \mathbf{m}_g(k-1) + \frac{1}{n_g(k)}\mathbf{d}_g(k) \\
\mathbf{Q}_g(k) &= \mathbf{Q}_g(k-1) + \frac{n_g(k-1)}{n_g(k)}\mathbf{d}_g(k)\ \mathbf{d}_g^T(k) \quad (3) \\
\mathbf{S}_g(k) &= \frac{1}{n_g(k)}\mathbf{Q}_g(k)
\end{aligned}$$

For the remaining classes $j = 1, \dots, p\ j \neq g$, the number of patterns, the sample mean and the sample dispersion matrix are not altered:

$$n_j(k) = n_j(k-1) \quad \mathbf{m}_j(k) = \mathbf{m}_j(k-1) \quad \mathbf{S}_j(k) = \mathbf{S}_j(k-1) \quad (4)$$

The LVQ network of the second layer is used to find the input vectors which are candidates for merging or not. The criterion of minimum Euclidean norm 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 following learning algorithm for the second layer LVQ is proposed:

1. Initialize randomly all the weight vectors $\mathbf{W}_l = (w_{1l}, \dots, w_{ql})^T \ \ l = 1, \dots, p'$ where $p \le p' < Np$.

2. For each weight vector provided by the first layer LVQ's $\mathbf{V}(k) = (v_1(k), \dots, v_q(k))^T$:

   a. Find the closest weight vector of the second layer LVQ, i.e., the final winner vector $\mathbf{W}_g(k)$ by using:

   $$\|\mathbf{V}(k) - \mathbf{W}_g(k)\| = \min_{j=1}^{p'}\{\|\mathbf{V}(k) - \mathbf{W}_j(k)\|\} \quad (5)$$

   b. If there is an output node of the second layer LVQ which has not been activated, i.e., there exists a free class:

   (i) Test the similarity between $\mathbf{W}_g(k)$ and $\mathbf{V}(k)$

   (ii) If $\mathbf{W}_g(k)$ and $\mathbf{V}(k)$ are proved similar, then merge them. The final winner is updated as LVQ suggests:

   $$\mathbf{W}_g(k+1) = \mathbf{W}_g(k) + a(t)[\mathbf{V}(k) - \mathbf{W}_g(k)] \quad (6)$$

   Modify the sample mean vector and the sample dispersion matrix. Let $n_g(k), \mathbf{m}_g(k), \mathbf{S}_g(k)$ denote the number of patterns, the sample mean vector and the sample dispersion matrix associated with the class of $\mathbf{W}_g$ respectively. Let also $n_V, \mathbf{m}_V, \mathbf{S}_V$ be the corresponding quantities associated with the class of $\mathbf{V}(k)$. The above-mentioned modifications are given by:

   $$\begin{aligned}
   \mathbf{m}_g(k) &= \frac{n_g(k-1)\mathbf{m}_g(k-1) + n_V\mathbf{m}_V}{n_g(k-1)+n_V} \quad (7) \\
   \mathbf{S}_g(k) &= \frac{n_g(k-1)\mathbf{S}_g(k-1) + n_V\mathbf{S}_V}{n_g(k-1)+n_V} \quad (8)
   \end{aligned}$$

   If $\mathbf{V}(k)$ were previously merged with another class, say C_e, $\mathbf{m}_e(k), \mathbf{S}_e(k)$ would also be modified by replacing addition with subtraction in both the numerator and the denominator of (7),(8).

   (iii) Otherwise, assign $\mathbf{V}(k)$ to a free class, say C_f. Update $\mathbf{W}_f(k)$ by using (6). Set the sample mean vector $\mathbf{S}_f(k)$ and the sample dispersion matrix $\mathbf{m}_f(k)$ of the free class as follows:

   $$\mathbf{m}_f(k) = \mathbf{m}_V \ \ \mathbf{S}_f(k) = \mathbf{S}_V \quad (9)$$

   c. If there is no free class, merge unconditionally $\mathbf{V}(k)$ and $\mathbf{W}_g(k)$. Update the winner vector and modify the sample mean vector and the sample dispersion matrix associated with the class C_g by using (6)–(8).

3. Repeat step 2 for $t = 1, 2, \dots$ until convergence is attained.

The recall procedure of the second layer LVQ is used for the classification of input patterns which have either been taken from the training set or not. It is used to determine the class C_g represented by $\mathbf{W}_g$ to which the input pattern $\mathbf{X}(k)$ is most closely associated with.

$$\mathbf{X}(k) \in C_g \text{ if } \|\mathbf{X}(k) - \mathbf{W}_g\| = \min_{j=1}^{p'}\{\|\mathbf{X}(k) - \mathbf{W}_j\|\} \quad (10)$$

The homogeneity of the winner 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 (step 2.b.(i)) by employing statistical tests on the mean vectors as well as on the dispersion matrices. Let $\mu_g(k), \Sigma_g(k)$ denote the statistical mean vector and the statistical dispersion matrix associated with the class of $\mathbf{W}_g$ respectively. Let also μ_V, Σ_V be the corresponding quantities associated with the class of $\mathbf{V}(k)$. The homogeneity of the dispersion matrices Σ_g and Σ_V is tested by using the statistic [8] :

$$T_1 = n_g \ln|\mathbf{S}_g^{-1}\mathbf{S}| + n_V \ln|\mathbf{S}_V^{-1}\mathbf{S}| \quad (11)$$

where

$$\mathbf{S} = \frac{1}{n_g + n_V}(n_g \mathbf{S}_g + n_V \mathbf{S}_V) \tag{12}$$

and $|.|$ denotes the determinant of a matrix. The statistic T_1 is distributed as $\chi^2_{\frac{q(q+1)}{2}}$. The following two cases are considered in order to test the homogeneity of the mean vectors:

1. Inhomogeneous dispersion matrices: A test statistic for the hypothesis that μ_g, μ_V are homogeneous is given by [8]:

$$T_2 = (\mathbf{m}_g - \mathbf{m}_V)^T \ (\frac{1}{n_g}\mathbf{S}_g + \frac{1}{n_V}\mathbf{S}_V \ )^{-1} \ (\mathbf{m}_g - \mathbf{m}_V) \leq k_\alpha \tag{13}$$

The threshold k_α in (13) can be approximately evaluated by the procedure described in [8].

2. Homogeneous dispersion matrices: A test statistic for the hypothesis that μ_g, μ_V are homogeneous is the following [7]:

$$T_3 = |\mathbf{I}_{q\times q} + \mathbf{S}_s^{-1} \ \mathbf{B}|^{-1} \tag{14}$$

where

$$\begin{aligned} \mathbf{S}_s &= n_g \mathbf{S}_g + n_V \mathbf{S}_V \\ \mathbf{B} &= \sum_{i=g,V} n_i (\mathbf{m}_i - \mathbf{m})\,(\mathbf{m}_i - \mathbf{m})^T \\ \mathbf{m} &= \frac{n_g \mathbf{m}_g + n_V \mathbf{m}_V}{n_g + n_V} \end{aligned} \tag{15}$$

The statistic T_3 is approximately distributed according to the Wilks distribution $\Lambda(q, n_g + n_V - 2, 1)$.

In many practical cases the above-described rigorous procedure is computationally demanding since it requires matrix inversion (although matrix inversion lemma [9] can be invoked) and the calculation of the determinant of a matrix. In addition, the number of matrices to be handled may be extraordinarily large as is the case in color image quantization to be discussed in the section 3. These problems can be alleviated by testing if there is any intersection between the hyperellipsoids associated with the winner vector of the second layer LVQ and the weight vectors provided by the first layer LVQ's. A simple proximity test of the form:

$$\frac{|w_{ig} - v_i(k)|}{\sqrt{S_{g_{ii}}} + \sqrt{S_{V_{ii}}}} \leq 1 \tag{16}$$

where $S_{g_{ii}}, S_{V_{ii}}$ are the i-th diagonal elements of the sample dispersion matrices of the corresponding classes can be used in step 2.b.(i). Inequality (16) implies that the hyperellipsoids are approximated by hyperparallelipipedes and simply tests if there is overlap along the i-th dimension. If such an overlap exists along any dimension, it is inferred that $\mathbf{V}(k)$ and $\mathbf{W}_g$ are similar.

3. EXPERIMENTAL RESULTS

The two-layer LVQ architecture has been applied successfully to color image quantization both in serial and parallel implementations. Figure 2 shows an RGB color image of dimensions 256 $\times$ 256 with 24 bits per pixel. Color image quantization aims at encoding each color pixel with one byte instead of three, thus reducing the number of RGB triplets to 256. The major drawback of the typical LVQ algorithm is the excessively large duration of the training phase due to the number of input pixels, i.e., color triplets to be considered. Therefore, parallel LVQ implementation is considered for speed-up. A straightforward parallelization (e.g. a parallel computation of the Euclidean distances) results in an implementation which has heavy communication load. If communication operations are slow, the implementation is very slow as well. On the contrary, if training parallelism is applied by splitting pixels into groups and the above-described novel LVQ architecture is used, a faster training procedure is attained. A two-layer LVQ having 16 networks in the first layer has been used. Each LVQ network of the first layer has 3 input nodes and classifies 1000 randomly selected pixels into 256 classes. The second layer LVQ receives 4096 weight vectors determined after four iterations of the 16 first layer networks. Six iterations have been shown adequate for the training of the second layer LVQ. The result of quantization is shown in Figure 3. The performance of the proposed algorithm has been compared to the one of the following algorithms:

(1) standard LVQ (single-layer) [2,3] having 3 input nodes and producing 256 output classes

(2) a modified LBG vector quantization algorithm which produces a codebook of 256 color vectors in 8 iterations [4]

(3) nonuniform sampling of each color histogram by first equalizing each color histogram separately, uniformly sampling each equalized histogram in a number of predetermined samples and inverse transforming. Seven samples have been chosen for red channel, six for green channel and six for blue one. In total, 252 RGB triplets have been used.

In the comparative study outlined above, we have used as figures of merit the mean-squared-error (MSE) and the signal-to-noise ratio (SNR) measured in dB, given by:

$$\begin{aligned} \text{MSE} &= \sum_{i=1}^{M}\sum_{j=1}^{M} ||\mathbf{X}(i,j) - \tilde{\mathbf{X}}(i,j)||^2 \\ \text{SNR}_{\text{dB}} &= 10 \log \frac{\text{MSE}}{\sum_{i=1}^{M}\sum_{j=1}^{M} ||\mathbf{X}(i,j)||^2} \end{aligned} \tag{17}$$

where $\mathbf{X}(i,j) = (X_R(i,j), X_G(i,j), X_B(i,j))^T$ represents the (i,j) pixel in the original color image, $\tilde{\mathbf{X}}(i,j)$ is a (3×1) vector which represents the (i,j) pixel in the quantized image and M is the number of rows/columns.

The results of the comparison are summarized in Table 1.

Table 1: FIGURES OF MERIT FOR COLOR IMAGE QUANTIZATION

Method	MSE	SNR
Two-layer LVQ	156	-25.6 √
Single-Layer LVQ	207	-24.3
Non-uniform sampling of each color histogram	297	-22.8
LBG	3777	-11.7

It is seen that the proposed two-layer LVQ is superior than any other method used.

A two-layer LVQ which incorporates the proximity test (16) has also been implemented using 16 transputers T800 working at 20 MHz under HELIOS operating system in farm topology. A speed-up of 3 has been observed between the parallel two-layer LVQ and a single-layer LVQ running in one transputer using the same number of pixels both in the learning and recall phase. A much larger speed-up would have been obtained, if other languages (e.g. OCCAM) supporting much faster communication had been used.

ACKNOWLEDGMENT

The first author is supported by a scholarship from the State Scholarship Foundation of Greece and the Bodosaki Foundation.

References

[1] T.K. Kohonen, *Self-Organization and Associative Memory*, 3rd ed., Berlin, Heidelberg, Germany: Springer-Verlag, 1989.

[2] P.K. Simpson, *Artificial Neural Systems*, Pergamon Press, 1990.

[3] T.K. Kohonen, "The Self-Organizing Map", *Proceedings of the IEEE*, vol. 78, no. 7, pp.1464–1480, September 1990.

[4] S.C. Ahalt, A.K. Krishnamurthy, P. Chen, D.E. Melton, "Competitive Learning Algorithms for Vector Quantization", *Neural Networks*, vol. 3, pp. 277–290, 1990.

[5] J.A. Kangas, T.K. Kohonen, J.T. Laaksonen, "Variants of Self-Organizing Maps", *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 93–99, March 1990.

[6] S.-G. Kong, B. Kosko, "Differential Competitive Learning for Centroid Estimation and Phoneme Recognition", *IEEE Transactions on Neural Networks*, vol. 2, no. 1, pp. 118–124, January 1991.

[7] K.V. Mardia, J.T. Kent, J.M. Bibby, *Multivariate Analysis*, Academic Press, 1979.

[8] G.A.F. Seber, *Multivariate Observations*, J. Wiley, 1984.

[9] A.S. Householder, *The Theory of Matrices in Numerical Analysis*, Blaisdell Waltham, Mass., 1964.

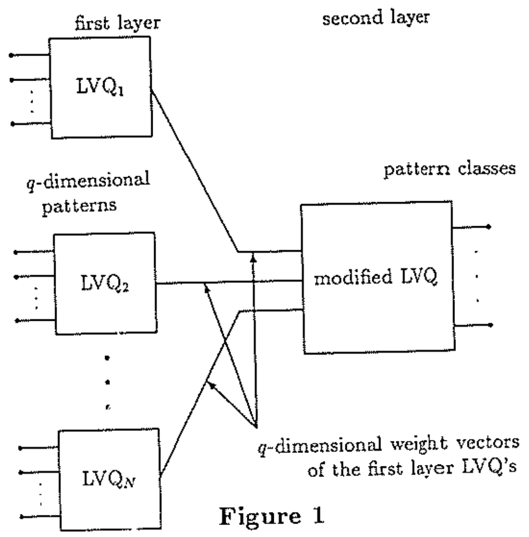

Figure 1

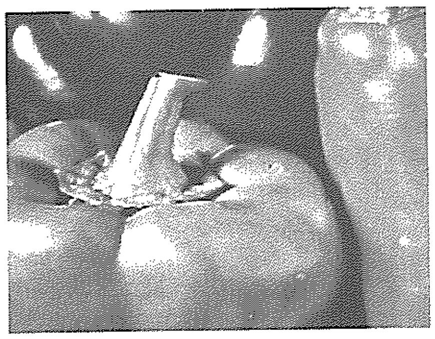

Figure 2

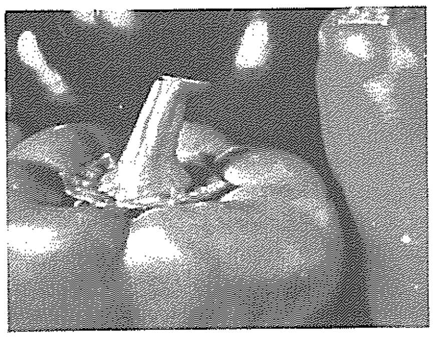

Figure 3