A Recurrent Multivalued Neural Network for codebook generation in Vector Quantization

J. Muñoz-Pérez[1], R. Benítez-Rochel[1], and E. Mérida-Casermeiro[2]

[1] Departamento de Lenguajes y Ciencias de la Computación,
[2] Departamento de Matemática Aplicada,
Universidad de Málaga
Complejo Tecnológico, Campus Universitario de Teatinos s/n, 29079 Málaga
munozp@lcc.uma.es, benitez@lcc.uma.es, merida@ctima.uma.es

Abstract. In this paper we propose a multivaluated recurrent neural network for vector quantization where the synaptic potential is given by a weighted sum of values of a function that evaluates the consensus between the states of the process units. Each process unit presents the state with the largest activation potential, that is, it depends on the state of the nearest process units (more strongly connected according to the synaptic weights). Like Hopfield network, it uses a computational energy function that always decreases (or remains constant) as the system evolves according to its dynamical rule based on an energy function that is equivalent to the distortion function of the vector quantization problem. It does not use tuning parameters and so it attains computational efficiency.

1 Introduction

Vector Quantization (VQ) is a well known technique that has been studied in a variety of contexts but most prominently for signal coding. In particular for speech coding and for image and video coding. In VQ the input space is divided into a number of distinct regions, and for each region a *reproduction* (reconstruction or prototype) *vector* is defined [5]. When the Euclidean distance is used as a similarity measure to decide on the region to which that input belongs, the quantizer is called *Voronoi quantizer*. However, practical use of VQ techniques has been limited because of the prohibitive amount of computation associated with existing algorithms. Artificial Neural Networks (ANN) with unsupervised learning have been succesfully applied to pattern recognition and signal detection problems. One objective of using unsupervised learning is to define the classes or categories of the input data. These categories have to be discovered by the network on the basis of correlations or similarity measures. A number of competitive learning (CL) algorithms have been proposed for constructing VQ, [1], [3], [7], [9], [12] . The major objective of the CL algorithms is to effectively utilize the neural units as much as possible so that the average distortion for quantizing the input data can be minimized.

On the other hand, clustering is defined as the partitioning of data into classes with similar characteristics. This is done by allowing data samples with common attributes to be grouped into the same class. In [11] the relationship between clustering and vector quantization is shown; the problem of selecting a clustering based on the least sum of squares becomes the problem of selecting the reproduction codebook.

When the Euclidean distance is used then competitive neural networks can be used for clustering and the synaptic vectors give us the prototypes (centroids).

In this paper, we propose a recurrent neural networks that could be used with any similarity measure. Moreover, it evolves according to a dynamical rule so that the distortion function (computational energy function) always decreases (or remains constant).

This paper is organized as follows; in section 2 we briefly describe the basic theory of Vector Quantization. In section 3 we present our multivalued recurrent neural network in order to be applied to Vector Quantization. The effectiveness of the proposed model for a synthetic data set, real-world data (Anderson's IRIS data) and for uniformly generated data is shown in section 4. Finally, the conclusions are given in section 5.

2 Vector Quantization

Next a basic definition of VQ and the structural properties are presented. They are independent of any statistical considerations or distortion measures.

Definition 1

A vector quantizer Q of dimension k and size N is a mapping from a vector in a k-dimensional Euclidean space, $\Re^k$, into a finite set $C = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N)$ containing N outputs or reproduction points, called *code vectors* $\mathbf{y}_i \in \Re^k$ for each $i \in \{1, 2, \ldots, N\}$. The set C is called the *codebook*.

Given a sample $\mathbf{x}_j \in \Re^k$, $j \in \{1, 2, \ldots, n\}$, a quantizer with size N is optimal if it minimizes the distortion function between those input vectors $\mathbf{x}_j$ and the reproduction vectors $\mathbf{y}_i$, $i \in \{1, 2, \ldots, N\}$.

The problem of finding an optimal quantizer can be expressed as

$$\text{Minimize} \qquad E(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N) = \sum_{i=1}^{N} \sum_{j \in R_i} \|\mathbf{x}_j - \mathbf{y}_i\|^2 \tag{1}$$

Each N points vector quantizer has associated a partition of $\Re^k$ into N disjoint and exhaustive regions or *cells*, R_i for $i \in \{1, 2, \ldots, N\}$. The ith cell is defined by

$$R_i = \{\mathbf{x} \in \Re^k : Q(\mathbf{x}) = \mathbf{y}_i\}, \tag{2}$$

For a given partition $\{R_i; i = 1, 2, \ldots N\}$, two necessary conditions to optimum code vector are (see [5]):

- C1) *Centroid condition*:

$$\mathbf{y}_i = \frac{\sum_{j \in R_i} \mathbf{x}_j}{|R_i|}$$

- C2) *Voronoi partition (competitiveness)*:
  For a fixed representative vectors, $(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N)$, the optimal partition should be constructed in such a manner that

$$Q(\mathbf{x}) = \mathbf{y}_i \Leftrightarrow \|\mathbf{x} - \mathbf{y}_i\| \leq \|\mathbf{x} - \mathbf{y}_r\|, \ \forall r \neq i$$

The problem (1) can be formulated in an alternative way and that is established in the following proposition:

Proposition 1

The problem (1) is equivalent to find the partition $\{C_1, C_2, \ldots, C_N\}$ of the input vector set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ that minimizes the expression

$$E = \sum_{i=1}^{N} \frac{1}{|C_i|} \sum_{\substack{r,s \in C_i \\ r<s}} \|\mathbf{x}_r - \mathbf{x}_s\|^2 \tag{3}$$

Proof:

It is easy to see that

$$\frac{1}{|C_i|} \sum_{\substack{r,s \in C_i \\ r<s}} \|\mathbf{x}_r - \mathbf{x}_s\|^2 = \sum_{j \in C_i} \|\mathbf{x}_j - \mathbf{y}_i\|^2.$$

In the following section a recurrent neural network is proposed in order to solve the above problem.

3 Multivalued Neural Network

3.1 Topology

The principal characteristics of our multivalued neural model $\mathcal{H}$ are

- The state of the neuron i is characterized by its output. So, the global state of the network with n neurons is determined by its state vector $(x_1, x_2, \ldots, x_n)$.
- The neurons outputs belong to a given set $\mathcal{M}$ where $\mathcal{M}$ can be $\Re$, $\Re^L$, $\{1, 2, \ldots, N\}$ or even a symbolic set.
- The network is fully connected and a weight $w_{ij} \in \Re$ is associated to each connection. The matrix of weights $\mathbf{W} = (w_{ij})$ is symmetric.
- Each state of the network has an associated energy given by

$$E = -\frac{1}{2} \sum_{i=1}^{N} \frac{1}{\sum_{j=1}^{N} S(x_i, x_j)} \sum_{j=1}^{N} w_{ij} S(x_i, x_j) \tag{4}$$

  where w_{ij} measures the influence of neuron i into neuron j and the application $S : \mathcal{M} \times \mathcal{M} \to \Re^+$ measures the matching between the outputs of neurons i and j.

The state of a neuron i indicates the group where the vector $\mathbf{x}_i$ is assigned. So a configuration of the network $(x_1, x_2, \ldots, x_i, \ldots, x_n)$ reflects the group where each vector $\mathbf{x}_i \in \Re^k$ in the sample is assigned.

The outputs of the neurons belongs to the set $\mathcal{M} = \{1, 2, \ldots, N\}$ where N is the size of the quantizer. It means that when $x_i = j$, $j \in \mathcal{M}$, then the network assigns the sample vector $\mathbf{x}_i$ to the group j.

Each state of the network has an associated energy given by (4) where the function S is given by (5), it establishes the consensus between the state of the neurons.

$$S(x_i, x_j) = \begin{cases} 1 & \text{if } x_i = x_j \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

In next section a computational dynamics is proposed where the computational energy function decreases. The network will evolve using that computational dynamics in order to get a maximum decrease in the energy associated to the configuration of the network at each step.

If each configuration of the network is associated with a possible solution for the Vector Quantization problem then the network will look for a configuration that minimizes the expression (3). So, the synaptic weights are identified as $w_{ij} = -\|\mathbf{x}_i - \mathbf{x}_j\|$.

3.2 Computational Dynamics

Let $x_r(t)$ be the state of neuron r at time t, $x_r(t) \in \{1, 2, \ldots, N\}$, and let $\mathbf{x}(t) = (x_1(t), x_2(t), \ldots, x_n(t))$ be the global state of the recurrent network at time t. In order to define a dynamics two concepts are used: the synaptic potential h_r and the activation potential h_r^* of neuron r.

Definition 2

The *synaptic potential* h_r of neuron r is defined by the expression

$$h_r(x_1(t), x_2(t), \ldots, x_N(t)) = \sum_{j=1}^{N} w_{rj} S(x_r(t), x_j(t))$$

where a pair of neurons r and j in the network are connected by a synaptic weight, w_{rj}, which specifies the contribution of the output signal $x_r(t)$ of neuron r to the synaptic potential acting on neuron j, and the function S is a function that measures the consensus between the states of the neurons.

Definition 3

The *activation potential* Δh_r of neuron r when $x_r(t) = b$, is defined by

$$\Delta h_r^b(t) = \frac{1}{1 + |C_b|}[h_r(t) - \theta_b]$$

where

$$\theta_b = \frac{1}{|C_b|} \sum_{i \in C_b} \frac{1}{2} h_i(t) \quad \text{and} \quad C_b = \{i \in \{1, 2, \ldots, N\} : x_i(t) = b, i \neq r\}$$

Note that θ_b is a half of the mean synaptic potential of neurons in state b.

If the neuron r is selected at time t, its state will be modified according to the deterministic rule.

$$x_r(t+1) = a \quad \text{si} \quad h_r^a(x_1(t), \ldots, x_{r-1}(t), a, x_{r+1}(t), \ldots, x_N(t)) = \tag{6}$$

$$\max_{b \in \mathcal{M}} h_i^b(x_1(t), \ldots, x_{i-1}(t), b, x_{r+1}(t), \ldots, x_N(t))$$

this rule that updates the unit is called *computational dynamics*.

The particular state of neuron r that satisfies the condition (6) is called the best matching for the input vector $\mathbf{x}_r$.

The selection of a neuron to perform updating is done randomly. The asynchronous (serial) updating procedure is continued until there no further changes to report. The state $\mathbf{x}(t) = (x_1(t), x_2(t), \ldots, x_N(t))$ that satisfies the condition $x_i(t+k) = x_i(t), \forall k > 1$, is called a *stable state* or *fixed point* of the phase space of the system.

3.3 Convergence

If the network is updated according to (6), the new configuration adopted will be a new state with less energy than the previous one. The network will evolve until a minimum of energy function is reached. It means that when the network is stabilized, then any change in one neuron will augment the value of the associated energy or will not cause any change in the previous energy.

Theorem 1

If the synaptic weight matrix is symmetric with null self-connections and S is given by (5) then the computational energy function decreases in each iteration when the networks evolves according to the computational dynamics (6).

Proof:

If the neuron r is updated at $t+1$, then the energy associated to the new state of the network is

$$E(t+1) = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \frac{S(x_i(t+1), x_j(t+1))}{\sum_{k=1}^{N} S(x_i(t+1), x_k(t+1))}$$

Clearly

$$-2E(t+1) = \sum_{\substack{i=1 \\ i \neq r}}^{N} \sum_{\substack{j=1 \\ j \neq r}}^{N} w_{ij} \frac{S(x_i(t+1), x_j(t+1))}{\sum_{k=1}^{N} S(x_i(t+1), x_k(t+1))} +$$

$$\sum_{\substack{i=1 \\ i \neq r}}^{N} w_{ir} \frac{S(x_i(t+1), x_r(t+1))}{\sum_{k=1}^{N} S(x_i(t+1), x_k(t+1))} + \sum_{j=1}^{N} w_{rj} \frac{S(x_r(t+1), x_j(t+1))}{\sum_{k=1}^{N} S(x_r(t+1), x_k(t+1))}$$

Since $w_{ii} = 0 \ \forall i$ and $w_{ij} = w_{ji} \ \forall i, j$, the second and third terms at the right are equals. It can be noticed that when $S(x_i(t+1), x_r(t+1)) = 1$ then

$\sum_{k=1}^N S(x_i(t+1), x_k(t+1)) = \sum_{k=1}^N S(x_r(t+1), x_k(t+1))$ (that is, denominators have same values in the above expression). So we have

$$-2E(t+1) = \sum_{\substack{i=1\\i\neq r}}^N \sum_{\substack{j=1\\j\neq r}}^N w_{ij} \frac{S(x_i(t+1), x_j(t+1))}{\sum_{k=1}^N S(x_i(t+1), x_k(t+1))}$$
$$+2\sum_{j=1}^N w_{rj} \frac{S(x_r(t+1), x_j(t+1))}{\sum_{k=1}^N S(x_r(t+1), x_k(t+1))}$$

Moreover, as $x_i(t+1) = x_i(t), \forall i \neq r$, then the first term at the right can be written as

$$\sum_{\substack{i=1\\i\neq r}}^N \sum_{\substack{j=1\\j\neq r}}^N w_{ij} \frac{S(x_i(t+1), x_j(t+1))}{\sum_{k=1}^N S(x_i(t+1), x_k(t+1))} = \sum_{\substack{i=1\\i\neq r}}^N \sum_{\substack{j=1\\j\neq r}}^N w_{ij} \frac{S(x_i(t), x_j(t))}{\sum_{k=1}^N S(x_i(t+1), x_k(t+1))}$$

Let C_a be the set defined by $C_a = \{i \neq r : x_i(t) = a\}$ and $C_b = \{i \neq r : x_i(t) = b\}$. Suppose that $x_r(t) = a$ and $x_r(t+1) = b$. Then we have that the only terms in the above expression updated from t to $t+1$ are

$$\sum_{i\in C_a} \sum_{\substack{j=1\\j\neq r}}^N w_{ij} \frac{S(x_i(t), x_j(t))}{\sum_{k=1}^N S(x_i(t), x_k(t))} + \sum_{i\in C_b} \sum_{\substack{j=1\\j\neq r}}^N w_{ij} \frac{S(x_i(t), x_j(t))}{\sum_{k=1}^N S(x_i(t), x_k(t))} =$$

$$\sum_{i\in C_a} \sum_{j\in C_a}^N w_{ij} \frac{S(x_i(t), x_j(t))}{|C_a|+1} + \sum_{i\in C_b} \sum_{j\in C_b}^N w_{ij} \frac{S(x_i(t), x_j(t))}{|C_b|}$$

Thus,

$$-2[E(t+1) - E(t)] = \sum_{i\in C_a} \sum_{j\in C_a}^N w_{ij} \frac{S(x_i(t), x_j(t))}{|C_a|} + \sum_{i\in C_b} \sum_{j\in C_b}^N w_{ij} \frac{S(x_i(t), x_j(t))}{|C_b|+1}$$
$$+2\sum_{j\in C_b} w_{rj} \frac{S(x_r(t+1), x_j(t+1))}{|C_b|+1}$$
$$-[\sum_{i\in C_a} \sum_{j\in C_a}^N w_{ij} \frac{S(x_i(t), x_j(t))}{|C_a|+1} + \sum_{i\in C_b} \sum_{j\in C_b}^N w_{ij} \frac{S(x_i(t), x_j(t))}{|C_b|}$$
$$+2\sum_{j\in C_a} w_{rj} \frac{S(x_r(t), x_j(t))}{|C_a|+1}]$$
$$= \frac{1}{|C_a|+1} \sum_{i\in C_a} \sum_{j\in C_a}^N w_{ij} \frac{S(x_i(t), x_j(t))}{|C_a|} - \frac{1}{|C_b|+1} \sum_{i\in C_b} \sum_{j\in C_b}^N w_{ij} \frac{S(x_i(t), x_j(t))}{|C_b|} +$$

$$\frac{1}{|C_b|+1}\sum_{j\in C_b} w_{rj}S(x_r(t+1),x_j(t+1)) - \frac{1}{|C_a|+1}\sum_{j\in C_a} w_{rj}S(x_r(t),x_j(t+1))$$

Hence $\Delta E = E(t+1) - E(t) =$

$$= \frac{1}{|C_a|+1}\Big[h_r(t) - \frac{1}{2|C_a|}\sum_{i\in C_a} h_i(t)\Big] - \frac{1}{|C_b|+1}\Big[h_r(t+1) - \frac{1}{2|C_b|}\sum_{i\in C_b} h_i(t+1)\Big]$$

$$= \frac{1}{2}[\Delta h_r^a(t) - \Delta h_r^b(t+1)] \leq 0$$

■

Theorem 2

The recurrent network is stable and the stable states of the network are the local minima of the computational energy function.

Proof:

This recurrent network could oscillate between adjacent states with equal synaptic potential. However, if $x_i(t) = b$, $x_i(t+1) = a$, $b \neq a$ only when $\Delta h_i(t+1) > \Delta h_i(t)$, then the recurrent network is stable since the computational energy function can take only a finite number of values, N^n, at most, and decreases in each iteration. Moreover, if $(x_1(t), x_2(t),, x_j(t),, x_N(t))$ is a stable state but it is not a local minima then there exists an adjacent state $(x_1(t), x_2(t),, x_j^\star(t),, x_N(t))$ with energy $E^\star(t)$, $E^\star(t) < E(t)$. From theorem 1, we have $\Delta h_j^\star(t) > \Delta h_j(t)$ and so the state $(x_1(t), x_2(t),, x_j(t),, x_N(t))$ is not stable, a contradiction.

■

4 Experimental results.

4.1 Iris data

We use Anderson's IRIS data [2] as an experimental data set. Properties of the data are well known [4] and has been used in many papers to illustrate various clustering (unsupervised). In figure 1(b) we present the clustering obtained for IRIS data with a network constituted by 150 process units and 3 states. Typical error rates for unsupervised designs are around 15 'mistakes'. The network finds a clustering with 13 classification errors. Note that our network does not use learning parameters or prototypes (centroids). The algorithms with unsupervised learning such as k-means, Lloyd [8] or LBG [7] find clustering between 13 and 17 misclassifications. All errors occur in the overlapping region between Versicolor and Virginica (see [10] and [13]) . In figure 1(a) we can see the decreasing energy function associated. It is interesting notice that the minimum energy is obtained after a few iterations that is because the considered dynamics provides the maximum diminution of the energy at each step.

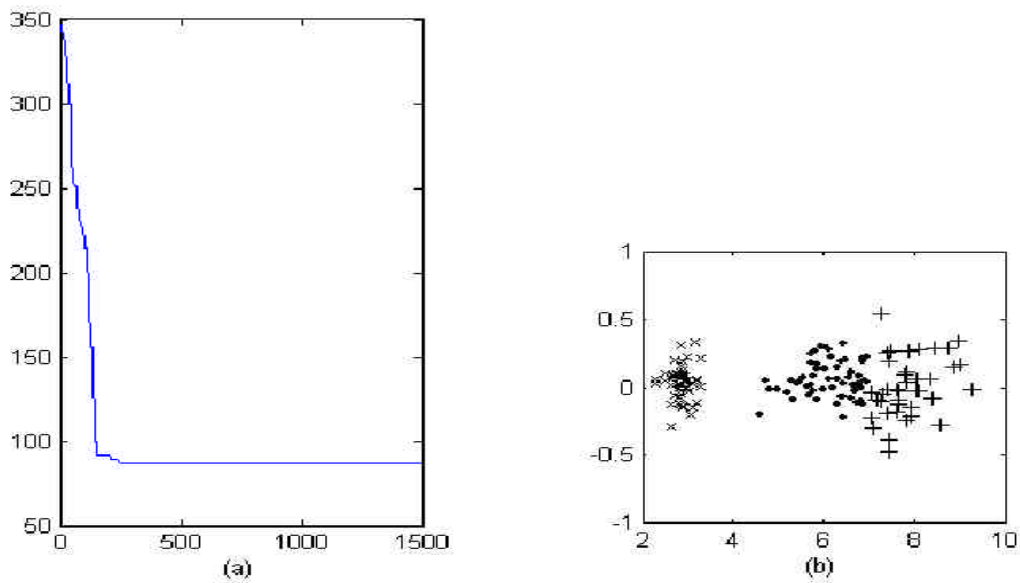

Fig. 1. (a)Evolution of the energy function. (b)IRIS data clustering with the recurrent network.

4.2 Uniform data.

In this experiment a set containing 200 points uniformly distributed in $[0,1] \times [0,1]$ is used. In figure 2 we present the clustering obtained by our proposed neural model. The network achieves 4 groups with similar sizes as it could be expected. It is interesting to note that the network suitably builds a Voronoi partition. It has been included a representation of the energy function and it can be observed that the minimum value is reached very fast, at 5 epochs.

5 Conclusions.

We have proposed a multivaluated recurrent neural network for vector quantization where the synaptic potential is given by a weighted sum of synaptic weights of process units with the same value. This synaptic potential is used to define the activation potential associated to the neural units. The synaptic weights are the opposite values of the distance between the sample patterns. Each process unit presents the state with maximum activation potential and the system evolves according to this dynamical rule so that the computational energy function (distortion function) always decreases (or remains constant) and it does not use tuning parameters. The network has n process units where n is the sample size, and eventually reaches a stable state at a local minimum of the energy function, that is a local minimum of the distortion function in Vector

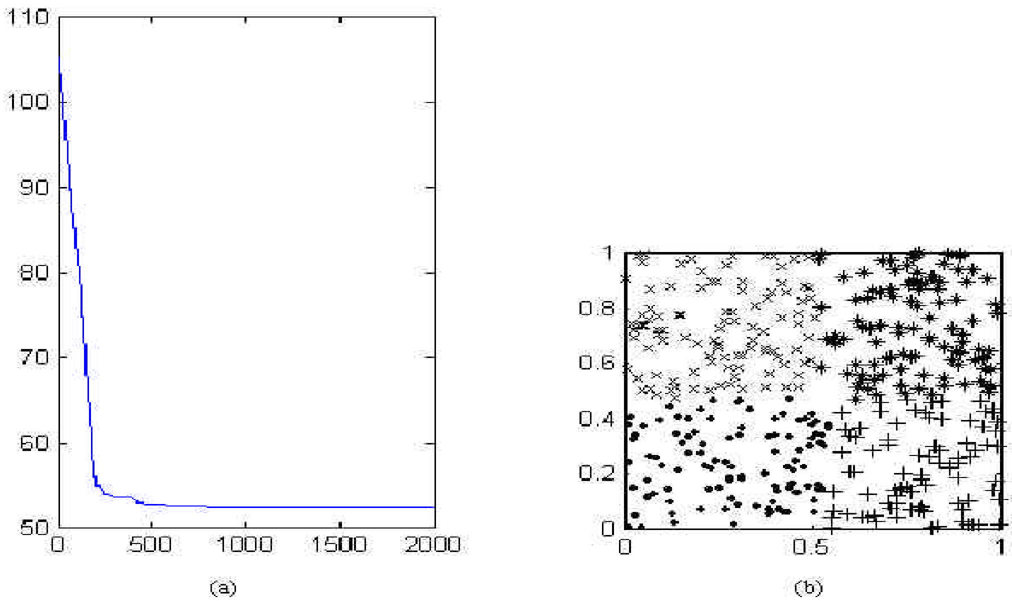

Fig. 2. (a)Evolution of the energy function. (b)Uniform data clustering with recurrent network.

Quantization problem. Moreover, the solution attained here forms the clusters or *cells* using the distances between the sample vectors, it does not use centroids.

References

1. Ahalt S.C., Krishnamurphy A.C., Chen P., and Melton D.E. Competitive learning algorithms for vector quantization. Neural Networks, **3** (1990) 277–290.
2. Anderson, E., The IRISes of the Gaspe Peninsula. Bulletin of the American IRIS Society, **59** (1939) 2–5
3. Dony R.D., and Haykin S. Neural Networks approaches to image compression. Proceeding of the IEEE, **83**(2) (1995) 288–303.
4. R. Duda and P. Hart, Pattern Classification and Scene Analysis. New York: Wiley, 1973.
5. Gersko A. and Gray R.M. Vector Quantization. Kluwer Academic Publishers (1992.
6. Gray R.M. Vector Quantization and signal compression IEEE ASSSP Magazine, **1** (1980) 4–29.
7. Linde Y., Buzo A., and Gray R.M. An algorithm for vector quantizer design. IEEE Trans. on Communication, **28**(1) (1980) 84–95.
8. Lloyd, S.P. Least squares quantization in PCM's. Bell Telephone Laboratories Paper. Murray Hill, NJ, (1957).
9. Mao J. and Jain A.K. A self-organizing network for hyperellipsoidal clustering (HEC). IEEE Trans. Neural Networks, **7** (1996) 16–29.

10. Pal, N.R., Bezdek, J.C. Tsao, E. C-K. Generalized Clustering Networks and Kohonen's Self-Organizing Scheme. IEEE Transactions on Neural Networks, **4**(2) (1993) 549–557.
11. Uchiyama T. and Arbib M.A. Color image segmentation using competitive learning. IEEE Trans. on Pattern Analysis and Machine Intelligence, **16**(12) (1994) 1197–1206.
12. Ueda N., and Nakano R. A new competitive learning approach based on a equidistorsion principle for designing optimal vector quantizers. Neural Networks, **7**(8) (1980) 81211–1227.
13. Zahid, N., Abouelala, O., Limouri, M. Essaid, A. Unsupervised fuzzy clustering. Pattern Recognition Letters. , **20** (1999) 123–129.