# Improved Production of Competitive Learning Rules with an Additional Term for Vector Quantization

Enrique Mérida-Casermeiro[1], Domingo López-Rodríguez[1],
Gloria Galán-Marín[2], and Juan M. Ortiz-de-Lazcano-Lobato[3]

[1] Department of Applied Mathematics,
University of Málaga, Málaga, Spain
{merida,dlopez}@ctima.uma.es

[2] Department of Electronics and Electromechanical Engineering,
University of Extremadura, Badajoz, Spain
gloriagm@unex.es

[3] Department of Computer Science and Artificial Intelligence,
University of Málaga, Málaga, Spain
jmortiz@lcc.uma.es

**Abstract.** In this work, a general framework for developing learning rules with an added term (perturbation term) is presented. Many learning rules commonly cited in the specialized literature can be derived from this general framework. This framework allows us to introduce some knowledge about vector quantization (as an optimization problem) in the distortion function in order to derive a new learning rule that uses that information to avoid certain local minima of the distortion function, leading to better performance than classical models. Computational experiments in image compression show that our proposed rule, derived from this general framework, can achieve better results than simple competitive learning and other models, with codebooks of less distortion.

## 1 Introduction

Vector quantization (VQ) is a coding method designed to represent a multidimensional space by means of a finite number of vectors, called representatives or prototypes. A vector quantizer maps each input vector in the $p$-dimensional Euclidean space $\mathbb{R}^p$ into one of the $K$ prototypes. The construction of a vector quantizer can be modelled as an optimization problem in which a distortion function is minimized. If the set of input vectors is finite, $X = \{\boldsymbol{x_1}, \ldots, \boldsymbol{x_N}\}$, and the set of $K$ prototypes (the codebook) is given by $\{\boldsymbol{w_1}, \ldots, \boldsymbol{w_K}\}$, an usual measure of the distortion introduced in the coding process is given by:

$$F(\boldsymbol{W}) = \frac{1}{N} \sum_{i=1}^{N} \min_{j=1,\ldots,K} ||\boldsymbol{x_i} - \boldsymbol{w_j}||^2 \qquad (1)$$

where the matrix $\boldsymbol{W} = (\boldsymbol{w_1}, \ldots, \boldsymbol{w_K})$ can also be considered as a vector with $Kp$ components.

B. Beliczynski et al. (Eds.): ICANNGA 2007, Part I, LNCS 4431, pp. 461–469, 2007.

Among the most popular applications of VQ one can find image and speech signals compression. VQ can also be considered as an approach to data clustering by means of combinatorial optimization techniques which divide the data into clusters according to a suitable cost (or distortion) function, like the one given in (1).

According to Shannon's rate distortion theory, VQ can always achieve better compression performance than any conventional coding technique based on the encoding of scalar quantities [1].

In its beginnings, the high amount of computation required by existing encoding techniques did not allow the use of VQ techniques. Linde, Buzo and Gray [2] proposed the well-known LBG algorithm for VQ which made no use of differentiation, and it is the standard approach to compute the codebook. While the LBG algorithm converges to a local minimum, it is not guaranteed to reach the global minimum.

Competitive neural networks are designed to cluster the input data. Thus, by using VQ techniques in this type of networks, tasks such as data coding and compression can be performed. This fact explains that the competitive learning is an appropriate algorithm for VQ of unlabelled data. A multitude of VQ techniques were developed in conjunction with competitive networks: Ahalt, Krishnamurthy and Chen [3] developed a training algorithm for designing VQ codebooks with near-optimal results, that can be used to develop adaptive vector quantizers. Yair, Zeger and Gersho [4] proved certain convergence properties of the Kohonen algorithm for VQ design, and also introduced the so-called soft competition scheme, which updates all the codevectors simultaneously with a step size that is proportional to its probability of winning. Pal, Bezdek and Tsao [5] proposed a generalization of learning VQ for clustering which avoids the necessity of defining an update neighbourhood scheme and the final centroids do not seem sensitive to initialization. The rival penalized competitive learning was introduced by Xu, Krzyzak and Oja [6]. In this new algorithm for each input not only the winner unit is modified to adapt itself to the input, but also its rival unlearns with a smaller learning rate. Ueda and Nakano [7] presented a new competitive learning algorithm with a selection mechanism based on the equidistortion principle for designing optimal vector quantizers. The selection mechanism enables the system to escape from local minima. Uchiyama and Arbib [8] showed the relationship between clustering and VQ and presented a competitive learning algorithm which generates units where the density of input vectors is high and showed its efficiency in color image segmentation based on the least sum of squares criterion. Mao and Jain [9] have proposed a self-organizing network for hyperellipsoidal clustering that is applied to texture segmentation problems. More recently, Gómez-Ruiz and Muñoz-Pérez [10,11] presented two new learning rules based on the principle of maximizing the distance between codevectors, introducing the concept of expansive and competitive learning achieving very good results.

We propose a new heuristic strategy to develop learning rules for competitive networks whose main contribution is the inclusion of an additional term in the distortion function allowing to escape from local minima when suitably defined. New learning rules can be derived from this generalized distortion function and used as weight update schemes for the network, as proved in the following sections.

## 2 Construction of Learning Rules with Additional Terms

In this section we will suppose that the set of possible solutions $\boldsymbol{W} = (\boldsymbol{w_1}, \ldots, \boldsymbol{w_K})$ to VQ is bounded ($||\boldsymbol{W}|| \leq M < \infty$).

We will also consider a sequence of distortion functions $\{F_n\}$ obtained as small perturbations of the original $F$. The perturbation term decreases as $n$ tends to $\infty$ and helps the learning process (optimization of the distortion function) to avoid certain local minima, that is, non-global solutions.

The analytic expression for the family of functions $F_n$ considered in this work is:

$$F_n(\boldsymbol{W}) = \frac{1}{N}\sum_{i=1}^{N}\left(\min_{j=1,\ldots,K} ||\boldsymbol{x_i} - \boldsymbol{w_j}||^2 + \alpha_n \cdot g(\boldsymbol{x_i}, X, \boldsymbol{W})\right) \tag{2}$$

where $g(\boldsymbol{x_i}, X, \cdot)$ is a differentiable and bounded function (for every $i$), that is, there exists a number $M'$ such that $||g(\boldsymbol{x_i}, X, \cdot)||_\infty \leq M' < \infty$ and $\{\alpha_n\}$ is a sequence of real numbers converging to 0. This perturbation function $g$ brings all information necessary to avoid local minima, and corresponds to the additional term in the learning rule, as we will see next.

This sequence satisfies one convergence condition (condition of uniform convergence): $\lim\limits_{n\to\infty} F_n(\boldsymbol{W}) = F(\boldsymbol{W})$ for all $\boldsymbol{W} \in V$, where $V$ is a compact (closed and bounded) subset of $\mathbb{R}^{Kp}$ defined as follows:

$$V = \{\boldsymbol{W} = (\boldsymbol{w_1}, \ldots, \boldsymbol{w_K}) = (w_{11}, w_{12}, \ldots, w_{1p}, \ldots, w_{Kp}) \in \mathbb{R}^{Kp} : ||\boldsymbol{W}|| \leq M\}$$

This convergence result ensures that the sequence $\{F_n\}$ of “perturbed” distortion functions converges to the original $F$. This fact implies that the learning rule associated to $F$ can be approximated by the ones associated to the successive $F_n$, as long as $\lim_{n\to\infty} \alpha_n = 0$.

To obtain learning rules from the definition of $F_n$, the stochastic gradient method will be used. This method can be described as follows:

- Consider a random $i \in \{1, \ldots, N\}$ and define:

$$T_i(\boldsymbol{W}) = \min_{j=1,\ldots,K} ||\boldsymbol{x_i} - \boldsymbol{w_j}||^2 + \alpha_n \cdot g(\boldsymbol{x_i}, X, \boldsymbol{W})$$

- The weight update rule is $\Delta \boldsymbol{w_j} = -\lambda \frac{\partial T_i}{\partial \boldsymbol{w_j}} =$

$$= \begin{cases} \lambda(\boldsymbol{x_i} - \boldsymbol{w_j}) - \lambda\alpha_n \frac{\partial g(\boldsymbol{x_i}, X, \boldsymbol{W})}{\partial \boldsymbol{w_j}} & \text{if } \boldsymbol{w_j} = \arg\min_{j=1,\ldots,K} ||\boldsymbol{x_i} - \boldsymbol{w_j}||^2 \\ -\lambda\alpha_n \frac{\partial g(\boldsymbol{x_i}, X, \boldsymbol{W})}{\partial \boldsymbol{w_j}} & \text{if } \boldsymbol{w_j} \neq \arg\min_{j=1,\ldots,K} ||\boldsymbol{x_i} - \boldsymbol{w_j}||^2 \end{cases}$$

that is, the winning neuron updates its weight $\boldsymbol{w} = \boldsymbol{w_j}$ accordingly to the formula $\lambda(\boldsymbol{x_i} - \boldsymbol{w}) - \lambda\alpha_n \frac{\partial g(\boldsymbol{x_i}, X, \boldsymbol{W})}{\partial \boldsymbol{w}}$, that includes the original competitive learning scheme $(\boldsymbol{x_i} - \boldsymbol{w})$ plus a perturbation term. For a non-winning neuron, the update is only caused by the perturbation in the distortion function.

The addition of this perturbation term provides a way to include more information in the learning process, as well as a generalization of the usual updating schemes.

By giving different values of the perturbation function $g$, we can obtain learning rules already known:

1. By defining $g \equiv 0$ (no perturbation), we obtain the classical learning rule:

$$\Delta\boldsymbol{w_j} = \begin{cases} \lambda(\boldsymbol{x_i} - \boldsymbol{w_j}) & \text{if } \boldsymbol{w_j} = \boldsymbol{w} \\ 0 & \text{if } \boldsymbol{w_j} \neq \boldsymbol{w} \end{cases} \tag{3}$$

2. If we define $g(\boldsymbol{x_i}, X, \boldsymbol{W}) = -||\boldsymbol{w} - \bar{\boldsymbol{x}}||^2$, where $\boldsymbol{w}$ represents the winning prototype when the input to the net is $\boldsymbol{x_i}$, that is,

$$||\boldsymbol{x_i} - \boldsymbol{w}||^2 = \min_{j=1,\ldots,K} ||\boldsymbol{x_i} - \boldsymbol{w_j}||^2 \ ,$$

we arrive at

$$F_n(\boldsymbol{W}) = \frac{1}{N}\sum_{i=1}^{N} \left(||\boldsymbol{x_i} - \boldsymbol{w}||^2 - \alpha_n \cdot ||\boldsymbol{w} - \bar{\boldsymbol{x}}||^2\right) \ .$$

With this definition, we are trying to minimize the usual distortion $||\boldsymbol{x_i} - \boldsymbol{w}||^2$ and, at the same time, maximize (note the change of sign) the distance from this prototype to the data centroid, $||\boldsymbol{w} - \bar{\boldsymbol{x}}||^2$.

As explained before, we can derive a learning rule given by the expression

$$\Delta w_j = \begin{cases} \lambda \cdot (\boldsymbol{x_i} - \boldsymbol{w}) - \lambda \cdot \alpha_n(\bar{\boldsymbol{x}} - \boldsymbol{w}) & \text{if } \boldsymbol{w} = \boldsymbol{w_j} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

By naming $\beta_n = \lambda \cdot \alpha_n$, the learning rule described in [10] is obtained.

3. The updating scheme from [11] can be obtained by defining the perturbation term $g(\boldsymbol{x_i}, X, \boldsymbol{W}) =< \bar{\boldsymbol{x}}, \boldsymbol{w} >$ where, as usual, $\boldsymbol{w}$ is the winning prototype and $< \cdot, \cdot >$ is the Euclidean inner product. The associated learning rule is derived:

$$\Delta\boldsymbol{w_j} = \begin{cases} \lambda \cdot (\boldsymbol{x_i} - \boldsymbol{w}) - \lambda \cdot \alpha_n \bar{\boldsymbol{x}} & \text{if } \boldsymbol{w} = \boldsymbol{w_j} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

To get the same analytic expression as in [11], it suffices to define $\beta_n$ such that $(1 - \lambda)\beta_n = \lambda\alpha_n$ and substitute in the last expression.

The aim of this rule is to minimize the inner product $< \bar{\boldsymbol{x}}, \boldsymbol{w} >$, which is achieved when vectors $\bar{\boldsymbol{x}}$ and $\boldsymbol{w}$ are in opposite directions, that is $\boldsymbol{w} \propto -\bar{\boldsymbol{x}}$.

By defining in an adequate way the perturbation term $g$, we can obtain multiple learning rules, including most of the mentioned in specialized literature, for example [6] and others. These rules are derived from an optimization problem, where the perturbation term helps to avoid local minima of the original distortion function.

## 3 A New Learning Rule

As mentioned in [10,11], a pair of necessary and sufficient conditions to ensure that the optimum of $F$ is global are:

- Prototypes (that is, $\boldsymbol{w_1}, \ldots, \boldsymbol{w_K}$) must be as far away as possible from the centroid of data, that is, the quantity

$$\sum_{j=1}^{K} n_j ||\boldsymbol{w_j} - \bar{\boldsymbol{x}}||^2 \tag{6}$$

  where $n_j$ is the number of data whose associated prototype is $\boldsymbol{w_j}$, must be maximized.
- At the same time, prototypes must be the centroids of the set of input patterns represented by them, that is, if

$$S_j = \{\boldsymbol{x} \in X : ||\boldsymbol{x} - \boldsymbol{w_j}||^2 = \min_{l=1,\ldots,K} ||\boldsymbol{x} - \boldsymbol{w_l}||^2\}$$

  then it must be satisfied $\boldsymbol{w_j} = \dfrac{1}{|S_j|} \sum_{\boldsymbol{x} \in S_j} \boldsymbol{x}$.

Our approach is based on these two conditions. The learning rule developed in this paper will try to maximize the value of (6), but in an indirect way.

The two references mentioned earlier [10,11] presented learning rules based on maximize that quantity directly.

We consider an alternative way of maximizing the distance from the prototypes to the data centroid that consists in maximizing the distance from the prototypes to the prototypes centroid and in minimizing the distance between both centroids:

$$\text{maximize } ||\boldsymbol{w} - \bar{\boldsymbol{w}}||^2 \tag{7}$$

$$\text{minimize } ||\bar{\boldsymbol{x}} - \bar{\boldsymbol{w}}||^2 \tag{8}$$

where $\boldsymbol{w}$ is the winning prototype, and $\bar{\boldsymbol{w}} = \frac{1}{K}\sum_{j=1}^{K} \boldsymbol{w_j}$ is the prototypes centroid.

This new learning rule has an important feature: by (8), the centroid of the prototypes approach the data centroid, so data are better represented by the prototypes. Moreover, since $\bar{\boldsymbol{w}} \approx \bar{\boldsymbol{x}}$ in the limit, (7) can be approximately rewritten as maximize $||\boldsymbol{w} - \bar{\boldsymbol{x}}||^2$. And this implies that the value of (6) is maximized. But, in addition, we have obtained another property of the solution: prototypes centroid is close to data centroid.

Then, the definition of the perturbation term, $g$, is as follows:

$$g(\boldsymbol{x_i}, X, \boldsymbol{W}) = ||\bar{\boldsymbol{w}} - \bar{\boldsymbol{x}}||^2 - ||\boldsymbol{w} - \bar{\boldsymbol{w}}||^2 \tag{9}$$

where $\boldsymbol{w}$ is the prototype verifying $||\boldsymbol{x_i} - \boldsymbol{w}||^2 = \min_{j=1,\ldots,K} ||\boldsymbol{x_i} - \boldsymbol{w_j}||^2$.

With this definition, the expression for the $n$-th distortion function $F_n$ is:

$$F_n(\boldsymbol{W}) = \frac{1}{N}\sum_{i=1}^{N}\left(\min_{j=1,\dots,K} ||\boldsymbol{x_i} - \boldsymbol{w_j}||^2 + \alpha_n \cdot (||\bar{\boldsymbol{w}} - \bar{\boldsymbol{x}}||^2 - ||\boldsymbol{w} - \bar{\boldsymbol{w}}||^2)\right)$$
$$= F(\boldsymbol{W}) + \alpha_n||\bar{\boldsymbol{w}} - \bar{\boldsymbol{x}}||^2 - \frac{\alpha_n}{N}\sum_{j=1}^{K} n_j||\boldsymbol{w_j} - \bar{\boldsymbol{w}}||^2 \qquad (10)$$

This expression shows that by minimizing $F_n$ we are also minimizing $F$ and the expression $||\bar{\boldsymbol{w}} - \bar{\boldsymbol{x}}||^2$ (that is, $\bar{\boldsymbol{w}} \approx \bar{\boldsymbol{x}}$), as well as maximizing the total dispersion of the prototypes $\sum_{j=1}^{K} n_j||\boldsymbol{w_j} - \bar{\boldsymbol{w}}||^2$, very related to the maximization of (6), as mentioned before.

The learning rule associated to this $F_n$ is given by:

$$\Delta \boldsymbol{w_j} = \begin{cases} \lambda(\boldsymbol{x_i} - \boldsymbol{w}) + \frac{\lambda\alpha_n}{K}(\bar{\boldsymbol{x}} - \bar{\boldsymbol{w}}) - \frac{(K-1)\lambda\alpha_n}{K}(\bar{\boldsymbol{w}} - \boldsymbol{w}) & \text{if} \quad \boldsymbol{w_j} = \boldsymbol{w} \\ \lambda\frac{\alpha_n}{K}(\bar{\boldsymbol{x}} - \boldsymbol{w}) & \text{if} \quad \boldsymbol{w_j} \neq \boldsymbol{w} \end{cases} \qquad (11)$$

It must be noted that, in this case, non-winning prototypes are also updated, that is, in each step, network weights are completely changed. This fact does not imply an increment of the computational time, since all updates are made in parallel, but it helps to avoid non-optimal solutions.

## 4 Experimental Results

In this section we illustrate the effectiveness of proposed approach in image compression.

In order to perform image compression with unsupervised learning, the set of input patterns is built by subdividing the gray level image into square subimages named windows. Hence, if the image size is $m \times n$ pixels and the window size is $k \times k$ pixels, we will obtain approximately $\frac{m \times n}{k^2}$ windows. These windows are our input patterns with $p = k \times k$ components (these patterns are obtained by arranging the pixel values row by row from top to bottom). The compression process consists in selecting a reduced set of $K$ representative windows (corresponding to the solution prototypes) and replacing each window of the original image with the closest representative window among the prototypes. In this experiment we have considered a window size of $4 \times 4$ pixels and $K = 32$ representative windows. Thus, the neural network has 32 output neurons.

As test images we have used the ones represented in Fig. 1. Each of these images has $256 \times 256$ pixels, so the number of input patterns is $N = \frac{256^2}{4^2} = 4096$.

The compression was made by using all of these sequential methods (no batch training is used in this work):

- Simple Competitive Learning (SCL), given by (3).
- Expansive and Competitive Learning from [10] (ECL1), given by (4).

(a) (b)

**Fig. 1.** Test images used in this work: (a) lenna, (b) kids

- Expansive and Competitive Learning from [11] (ECL2), given by (5).
- The proposed algorithm, whose learning rule is described by (11).

After 10 executions of each algorithm, the average distortion over the minimum is showed on Table 1. That is, if for one image the minimum distortion obtained among all the algorithms is $m_0$ and $\mu_i$ represents the average distortion of the $i$-th algorithm in those 10 executions, the measure of the goodness present in Table 1 is given by:

$$M_i = \frac{\mu_i - m_0}{m_0}$$

**Table 1.** Average results of the 4 algorithms compared in this work after 10 independent executions

| Image | SCL | ECL1 | ECL2 | Proposed |
|---|---|---|---|---|
| lenna | 2.33 | 22.02 | 3.60 | 0.32 |
| kids | 1.48 | 3.99 | 2.39 | 1.42 |

It can be observed that the proposed algorithm achieves better results on average than the other learning rules compared in this work.

In Fig. 2 we can compare the compression results of the four algorithms on the test images.

If $K = 32$ representatives are used, and window size is $4 \times 4$, then 128 bits are needed to represent each window, but only 5 to represent the codewords, so we may obtain a compression rate of 128 to 5, that is, 25 to 1 approximately.

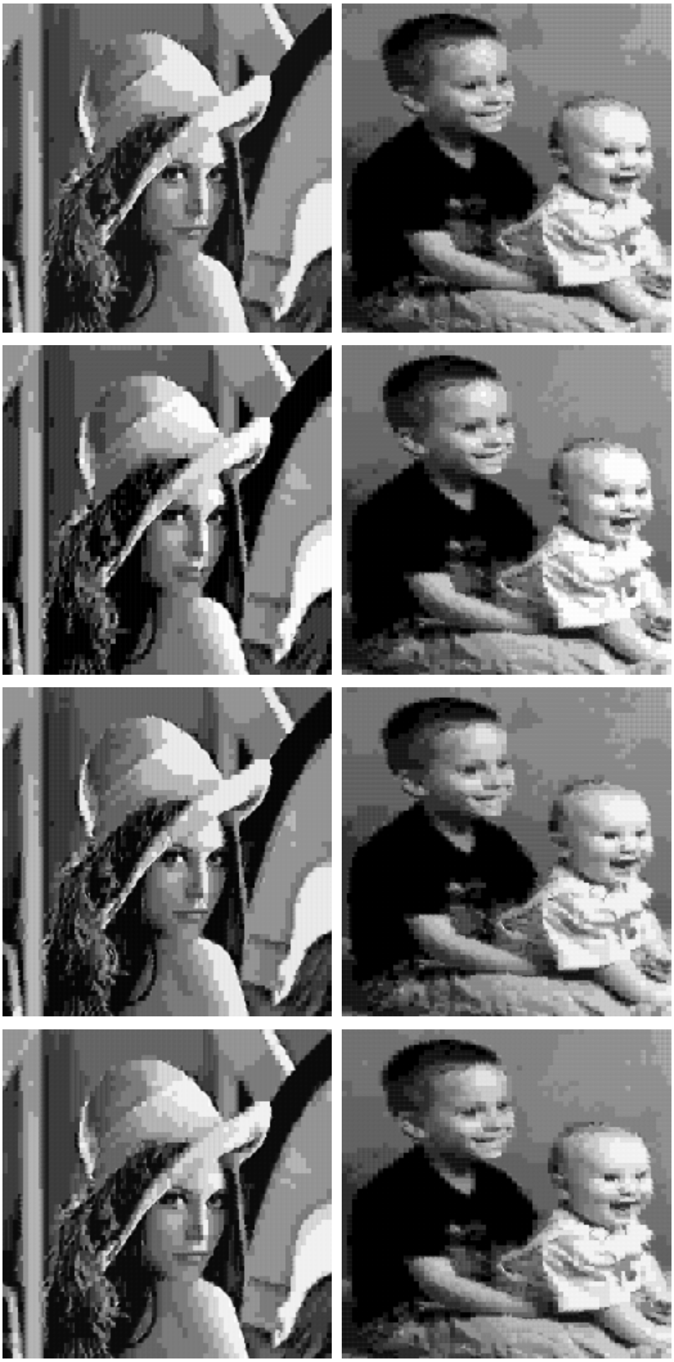

**Fig. 2.** Compressed images using (from top to bottom): SCL, ECL1, ECL2 and the proposed learning rule

## 5 Conclusions

In this work we have proposed a general framework for developing learning rules with an added term that plays the role of a perturbation leading to better compression results by including some kind of knowledge about the problem of vector quantization.

This general framework englobes many of the learning rules most commonly cited in the literature, just by defining in a proper way the perturbation term $g$.

With the help of some previous work [10,11], we have derived a new learning rule that achieves better results by avoiding some local minima of the distortion function, which measures the quality of the compression.

As a future research line, we intend to study the use of frameworks of this kind, generalizing the usual competitive learning, and its convergence to (global or local) minima of the distortion function. It will be interesting to study the convergence in sequential training as well as in batch training.

## References

1. Gray, R.M.: Vector quantization. IEEE ASSP Magazine **1** (1980) 4–29
2. Linde, Y., Buzo, A., Gray, R.M.: An algorithm for vector quantizer design. IEEE Trans. on Communications **28**(1) (1980) 84–95
3. Ahalt, S.C., Krishnamurthy, A.K., Chen, P., Melton, D.E.: Competitive learning algorithms for vector quantization. Neural Networks **3** (1990) 277–290
4. Yair, E.K., Zeger, K., Gersho, A.: Competitive learning and soft competition for vector quantizer design. IEEE Trans. Signal Processing **40**(2) (1992) 294–308
5. Pal, N.R., Bezdek, J.C., Tsao, E.C.: Generalized clustering networks and kohonens self-organizing scheme. IEEE Trans. Neural Networks **4**(4) (1993) 549–557
6. Xu, L., Krzyzak, A., Oja, E.: Rival penalized competitive learning for clustering analysis, rbf net, and curve detection. IEEE Trans. Neural Networks **4**(4) (1993) 636–649
7. Ueda, N., Nakano, R.: A new competitive learning approach based on an equidistortion principle for designing optimal vector quantizers. Neural Networks **7**(8) (1994) 1211–1227
8. Uchiyama, T., Arbib, M.: Color image segmentation using competitive learning. IEEE Trans. on Pattern Analysis and Machine Intelligence **16**(2) (1994) 1197–1206
9. Mao, J., Jain, A.K.: A self-organizing network for hyperellipsoidal clustering (hec). IEEE Trans. Neural Networks **7** (1996) 16–29
10. Gómez-Ruiz, J.A., Muñoz Pérez, J., López-Rubio, E., García-Bernal, M.A.: Expansive and competitive neural networks. Lecture Notes in Computer Science **2084** (2001) 355 – 362
11. Muñoz Pérez, J., Gómez-Ruiz, J.A., López-Rubio, E., García-Bernal, M.A.: Expansive and competitive learning for vector quantization. Neural Processing Letters **15** (2002) 261–273