Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT

ELSEVIER

Signal Processing 83 (2003) 649 – 654

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

Short communication

Centroid neural network adaptive resonance theory for vector quantization☆

Tzu-Chao Lin, Pao-Ta Yu*

Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, 62107 Taiwan, ROC

Received 12 November 2001; received in revised form 11 April 2002

Abstract

In this paper, a novel unsupervised competitive learning algorithm, called the centroid neural network adaptive resonance theory (CNN-ART) algorithm, is proposed to relieve the dependence on the initial codewords of the codebook in contrast to the conventional algorithms with vector quantization in lossy image compression. The design of the CNN-ART algorithm is mainly based on the adaptive resonance theory structure, and then a gradient-descent-based learning rule is derived so that the CNN-ART algorithm does not require a predetermined schedule for learning rate. Furthermore, the appropriate initial weights obtained by the CNN-ART algorithm can be applied as an initial codebook for the Linde–Buzo–Gray (LBG) algorithm such that the compression performance can be greatly improved. In this paper, the extensive simulations demonstrate that the CNN-ART algorithm does outperform other algorithms like LBG, self-organizing feature map and differential competitive learning.

Keywords: Adaptive resonance theory; Codebook; Codeword; Vector quantization

1. Introduction

The design of a codebook is the typical application in lossy image compression based on vector quantization. The well-known k-means clustering algorithm is one of the most popular competitive learning vector quantization schemes [1]. Although the k-means algorithm is simple and appealing, it has some inevitable problems [4]. The first one is that the number k of clusters must be pre-known and fixed. Intuitively, the mean square error (MSE) seems to monotonically decrease with an increasing k. However, the MSE may sometimes increase even when the value of k increases. Another problem is that the initial codebook strongly affects the performance of the k-means algorithm. Still another problem is that the algorithm may not converge towards an optimal solution. A variation of the k-means algorithm, known as the LBG algorithm [5], still suffers from these problems.

The self-organizing feature map (SOFM) is one of the most popular competitive neural network algorithms [2]. During the training procedure, the SOFM algorithm finds a winner neuron and updates the weights of both the winner and its neighbors. In order to obtain the best results from SOFM, the updated neighborhood, the learning rate and the total number of iterations for a given set of data should be chosen

☆ This work is supported by National Science Council of the Republic of China under Grant NSC 90-2213-E-194-043.

* Corresponding author. Tel.: +886-5-272-0411; fax: +886-5-272-0859.

E-mail address: csipty@cs.ccu.edu.tw (P.-T. Yu).

0165-1684/03/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0165-1684(02)00453-X

carefully. However, it is difficult to determine an appropriate set of initial weights (initial codebook) such that the final learning performance can be acceptable.

In the learning process, the differential competitive learning (DCL) algorithm provides rewards and punishments [3]. However, it is difficult to determine an appropriate set of initial weights, and a schedule for the optimal learning rate as mentioned in the previous paragraph is still nowhere to be found.

A new unsupervised competitive learning algorithm, called the centroid neural network adaptive resonance theory (CNN-ART) algorithm, is proposed in this paper. Since the neurons of CNN-ART grow dynamically until the size of the codebook, the set of synaptic weights can be treated as a codebook. Due to this significant property of the CNN-ART algorithm, the appropriate initial codebook can be easily obtained in contrast to the conventional algorithms mentioned earlier. The CNN-ART algorithm can derive a gradient-descent-based learning rule, so one of the very advantageous features is that CNN-ART does not require a schedule for the learning rate. As with DCL, CNN-ART provides reward and punishment for the winner and the loser, respectively, in accordance with the learning rate. After iterations, the weights can converge fast towards the centroids of clusters for the expected codebook size.

In Section 2, the design of CNN-ART is described in detail. In Section 3, the simulation results are exhibited. Finally, the conclusion is given in Section 4.

2. A new competitive learning algorithm—CNN-ART

2.1. The CNN-ART architecture

The adaptive resonance theory (ART) algorithm proposed by Grossberg is a special type of neural network that can realize classification without supervision [6]. Based on the ART algorithm, we have developed the new CNN-ART algorithm in order to design a codebook. The graphical representation of the CNN-ART architecture is shown in Fig. 1. The CNN-ART network consists of an input layer and an output layer called MINNET, which is a simple net for determining the nearest cluster exemplar. Each node in the input layer is connected directly to the neurons in the output layer. A synaptic weight w_p, $p=0,1,\dots k$, that has the same dimension as the input data x_i, is associated with each neuron in the output layer. As a result, the set of weights w_p can be treated as a codebook. In the MINNET subnet, the number of neurons starts with one and grows by one each time until the desired codebook size is reached; that is, the number of neurons is proportional to the size of the codebook. Due to this property, the CNN-ART algorithm does not require the selection of any appropriate initial codeword in advance. Each neuron of the MINNET is weighted and interconnected so as to maintain its own value $(+1)$ and attempt to be "lateral inhibition" $(-\varepsilon)$ for the others, and the output value $O_j^{(t)}$ of the neuron j at time t is given by the relationship

$$O_j^{(t)} = f_t\left(O_j^{(t-1)} - \varepsilon \sum_{i\neq j} O_i^{(t-1)}\right),$$
$$j,i=0,1,\dots,n,\ t\geqslant 1, \tag{1}$$

where

$$O_j^{(0)} = \|x_i - w_j\|^2, \qquad f_t(\beta) = \begin{cases} \beta & if\ \beta<0, \\ 0 & otherwise, \end{cases} \tag{2}$$

and n is the current number of output layer neurons, $\varepsilon > 1/n$. The MINNET is repeatedly run until only one output $O_j^{(t)}$ remains to be negative with all the others being 0, and that is when the winner neuron j is found. In other words, the competitive output layer is a "winner-take-all" type of net.

2.2. Learning rules in CNN-ART

In the CNN-ART algorithm, the centroid of every cluster is computed by the following equation:

$$w_j = \frac{1}{M_j}\sum_{i=1}^{M_j} x_i(j). \tag{3}$$

Note that the cluster j has M_j members and that $x_i(j)$ denotes the input data vector x_i in the cluster j. When an input data vector x_i is applied to the network at time t, the CNN-ART algorithm rewards the winner neuron j. The adaptive equation $w_j^{(t)}$ can be written

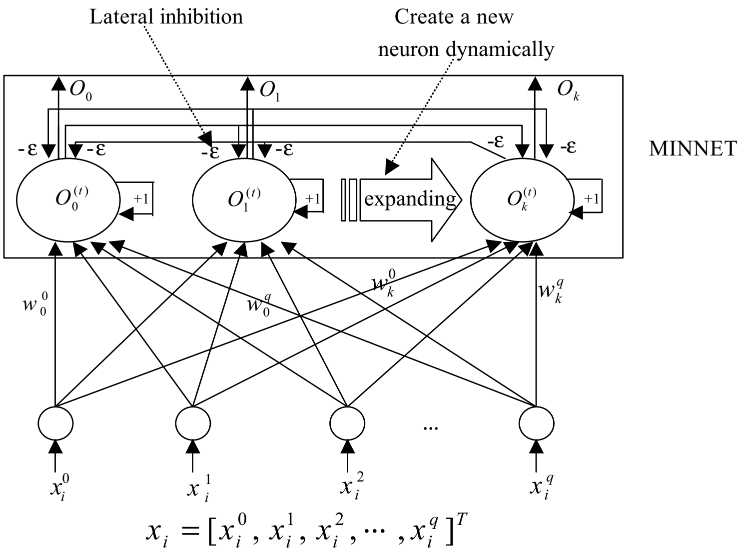

Fig. 1. The architecture of CNN-ART.

as follows:

$$w_j^{(t)} = \frac{M_j w_j^{(t-1)}}{M_j+1} + \frac{x_i}{M_j+1}$$
$$= w_j^{(t-1)} + \frac{1}{M_j+1}[x_i - w_j^{(t-1)}]. \quad (4)$$

At the same time, the CNN-ART algorithm will punish the loser neuron y. The adaptive equation $w_y^{(t)}$ can be written as follows:

$$w_y^{(t)} = \frac{M_y w_y^{(t-1)}}{M_y-1} - \frac{x_i}{M_y-1}$$
$$= w_y^{(t-1)} - \frac{1}{M_y-1}[x_i - w_y^{(t-1)}]. \quad (5)$$

From Eqs. (4) and (5), a gradient-descent-based learning rule can be derived. The learning rate can be summarized as

$$\alpha(t) = \begin{cases} (+1)\dfrac{1}{M_j+1} & if\ j\ is\ a\ winner, \\ (-1)\dfrac{1}{M_y-1} & if\ y\ is\ a\ loser, \\ 0 & otherwise. \end{cases} \quad (6)$$

Through Eq. (6), CNN-ART provides rewards and punishments for competitive learning by using positive and negative learning gains, respectively. Furthermore, the learning rate is adaptively according to the number of members in the winner or loser cluster, so CNN-ART does not require a predetermined schedule.

2.3. The CNN-ART algorithm

In the CNN-ART competitive learning algorithm, at first, an input training vector x_0 is selected as the centroid of the first cluster, and then the next input training vector is compared to the first cluster centroid. It is classified as part of the first cluster if its Euclidean distance is smaller than the vigilance parameter. Otherwise, it forms the centroid of a new cluster. This process is repeated for all the training input vectors. Table 1 shows a pseudo-code of the CNN-ART algorithm. If we set the vigilance to be low, then it will gather a large number of clusters. On the other hand, if the vigilance is high, then long distances can be tolerated, and a smaller set of clusters will be formed. As the input training vectors are reiterated in the CNN-ART algorithm, two cases may occur:

(1) Input training vector x_i changes the cluster from the old cluster to a new one. In this case, the weights of the new neuron are updated in

Table 1
The CNN-ART algorithm

Step 1:	[Initialize]
	Set CodebookSize k, Total_input_vectors N;
	$n = 0,\ w_0 = x_0$; // n : the number of neurons.
Step 2:	[Set vigilance] Select *vigilance*;
Step 3:	[Do loop for designing a codebook]
	Repeat
	For ($i = 0;\ i \leqslant N;\ i++$)
	(a) [Calculate Euclidean distances]
	Apply an input vector x_i to the network, and calculate Euclidean distances between x_i and exiting weights.
	(b) [Decide the winner neuron j]
	Find the smallest Euclidean distance, say d_j.
	(c) [Test tolerance]
	If ($d_j >$ *vigilance*) AND ($n < k$)
	goto (d);
	Else goto (e);
	(d) [Form a new neuron]
	$n = n + 1$; $w_n = x_i$;
	goto (a);
	(e) [Update the weights]
	UpdatewinnerWeight w_j; //update the weights of a winner neuron
	UpdateloserWeight w_y; //update the weights of a loser neuron
	(f) [Check non-uniformed clusters]
	If a cluster has not many input vectors to itself, in a snapshot
	Spare this cluster for later use;
	End for
	Until convergence criterion satisfied

accordance with Eq. (4), and the weights of the old neuron are updated in accordance with Eq. (5).

(2) Input training vector x_i is classified to belong to the same cluster as before. In this case, no learning action is performed.

The CNN-ART algorithm is reiterated until a stable cluster formation occurs, so it is not necessary to decide the total number of iterations to process in advance. The final set of synaptic weights can be treated as a codebook.

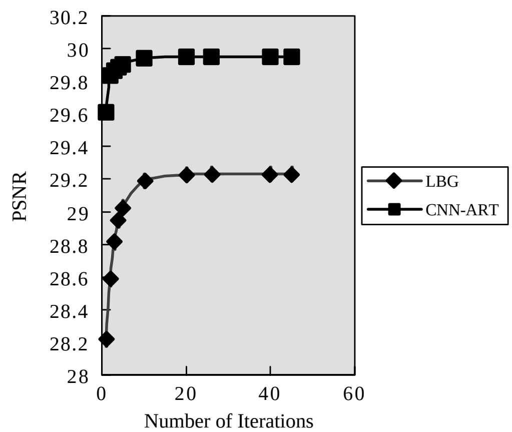

Fig. 2. Comparison in PSNR (dB) between CNN-ART and LBG with different numbers of iterations.

3. The simulation of image compression for codebook design

Experiments have been conducted on LBG, SOFM, DCL, and the proposed CNN-ART algorithm to see how they perform on the task of designing a codebook for image compression. The "Lena" and "Peppers" images, 256 × 256 pixels of size each, are used as the training images in the experiments. From the comparison of the performances between LBG and CNN-ART shown in Fig. 2, we see that

Table 2
PSNR (dB) comparison among algorithms with different compression rates for "Lena" and "Peppers" images

Algorithms	Lena			Peppers		
	0.5	0.5625	0.625	0.5	0.5625	0.625
LBG	29.47	30.85	32.93	29.79	31.46	32.79
SOFM	29.30	29.94	30.43	29.89	31.22	32.64
CNN-ART-LBG	29.97	31.97	36.63	30.18	32.33	36.21
CNN-ART	29.95	31.92	36.21	30.16	32.25	35.50

Fig. 3. (a) The original "Lena" image and (b) the reconstructed image (0.5 bpp) by CNN-ART, PSNR = 29.951 dB.

our CNN-ART algorithm always does better than the LBG algorithm, especially when it comes to one iteration only. Note that the first k input training vectors are used as the initial codebook of LBG. Due to this significant property, the one iterative set of appropriate weights can be applied as the initial codebook of LBG, now called the CNN-ART-LBG algorithm, such that the compression performance can be greatly improved. Table 2 shows the comparison of the resulting PSNR among the algorithms, where CNN-ART-LBG has the best result of all. Fig. 3(a) shows the original "Lena" image, and Fig. 3(b) shows its reconstructed image by using the CNN-ART algorithm. Here, still another experiment is conducted to demonstrate the robustness of the CNN-ART algorithm. The "Peppers" image is used in this experiment for training the codebook, and the resulting codebook is tested with the "Lena" image. According to Table 3, CNN-ART has exhibited a satisfactory result. Note that the first k input training vectors are used as the initial codebook of DCL and SOFM, and the splitting initialization method is used for designing the initial codebook of LBG in Tables 2 and 3.

Table 3
PSNR (dB) comparison on the compression results of the "Lena" image at different compression rates with codebooks designed by the "Peppers" image

Algorithms	Bits per pixel (bpp)		
	0.5625	0.625	0.6875
LBG	27.76	28.14	28.32
SOFM	29.31	29.92	30.31
DCL	29.98	30.39	30.74
CNN-ART	30.71	31.25	31.37

When we simulate the clustering using the CNN-ART algorithm, two problems may occur to affect the final performance. The first problem is the density of the non-uniformed clusters, and the second is the decision of the size of vigilance. To overcome the first problem, we set some "inspection points" to check the size of each current cluster. If a cluster has

Table 4
The effect of the vigilance

Vigilance	Clusters (k)	PSNR(dB)
0.15	204	29.06
0.65	491	31.89
0.045	676	33.34
0.025	942	35.62
0.015	1208	37.34
0.0095	1495	39.07
0.0065	1843	40.85
0.005	2048	41.61
0.0035	2048	39.08
0.0025	2048	37.43
0.001	2048	35.94

few input vectors to itself, especially only one, in a snapshot, we eliminate the cluster during the process. These empty clusters can be used again. As for the second problem, that is, how to determine the number of clusters in the classical MSE analysis, Table 4 shows that the PSNR value can sometimes be lower even if the number of clusters is larger when the size of vigilance is poorly decided. In addition, Table 4 shows that as the number of clusters is fixed, say at 2048, the resulting PSNR for the "Peppers" image depends on the size of vigilance, capable of varying from 35.94 to 41.61 as the vigilance size goes from 0.001 to 0.005. Finally, the size of vigilance is set to be equal to 0.005 to get the best result from extensive experiments. Note that the number of clusters can be seen as the size of the codebook, usually chosen to be a power of 2 for convenience when designing the codebook.

4. Conclusion

The design of the CNN-ART algorithm is not necessary to select the appropriate initial codewords of a codebook and a predetermined schedule for learning rate. Furthermore, CNN-ART adaptively learns through the training vectors and converges much faster than conventional learning algorithms, and it does not require the total number of iterations in advance. Due to the significant vigilance property of the CNN-ART algorithm, the appropriate initial weights obtained from the CNN-ART algorithm can be applied as an initial codebook to the LBG algorithm such that the compression performance can be greatly improved. Moreover, our experimental results have also demonstrated that CNN-ART for image compression based on the fact that vector quantization (VQ) outperforms other competitive learning algorithms.

References

[1] A. Gersho, R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic, Boston, MA, 1992.
[2] T. Kohonen, Self-organization and Associative Memory, 3rd Edition, Springer, Berlin, Germany, 1989.
[3] S. Kong, B. Kosko, Differential competitive learning for centroid estimation and phoneme recognition, IEEE Trans. Neural Networks 2 (January 1991) 118–124.
[4] Lei Xu, Bayesian Ying-Yang machine, clustering and number of clusters, Pattern Recognition Lett. 18 (1997) 1167–1178.
[5] Y. Linde, A. Buzo, R.M. Gray, An algorithm for vector quantizatizer design, IEEE Trans. Commun. 28 (1980) 84–95.
[6] Y.H. Pao, Adaptive Pattern Recognition and Neural Networks, Addison Wesley, Reading, MA, 1989.