Proceedings of the IASTED International Conference

Artificial Intelligence and Soft Computing

May 27-30, 1998
Cancún, Mexico

Editor: M.H. Hamza

A Publication of
The International Association of Science and Technology for Development - IASTED

ISBN: 0-88986-256-7
ISSN: 1482-7913

IASTED/ACTA Press
Anaheim ♦ Calgary ♦ Zürich

Proceedings of the IASTED International Conference

Artificial Intelligence and Soft Computing

May 27-30, 1998
Cancún, Mexico

Editor: M.H. Hamza

A Publication of
The International Association of Science and Technology
for Development - IASTED

ISBN: 0-88986-256-7
ISSN: 1482-7913

IASTED/ACTA Press
Anaheim ♦ Calgary ♦ Zürich

Table of Contents

Algorithms, Logic and Soft Computing 143

Linguistics 177

Object-Oriented Systems 223

Neural Networks 269

Proceedings of the IASTED International Conference
Artificial Intelligence and Soft Computing
May 27-30, 1998 - Cancún, Mexico

IMPROVING THE PERFORMANCE OF DIFFERENTIAL COMPETITIVE LEARNING MODEL IN CLUSTERING TASKS

Guilherme de A. Barreto   Aluízio F. R. Araújo
University of São Paulo (USP), Department of Electrical Engineering
Av. Dr. Carlos Botelho, 1465
13560-970 São Carlos - SP, Brazil.
{gbarreto, aluizioa}@sel.eesc.sc.usp.br

ABSTRACT

In this paper we propose two neural algorithms that can be considered a simplification and a generalization of the Differential Competitive Learning (DCL) neural network, respectively. Firstly, we suggest some simplifications for the original DCL model to eliminate redundant aspects of the competition mechanism. We get rid of the lateral connections arguing that it is possible because the winning neuron is chosen based solely on metrical similarity measures and the lateral feedback weights play no effective role. The activation rule is made simpler requiring less computational effort. In the second model, we show how to combine lateral connections with metrical relations on the activation and the learning rules of DCL to effectively estimate cluster centroids. This model is also less sensitive to weight initialization. A number of simulations are carried out to compare the presented models in unsupervised clustering tasks.

Keywords: Neural networks, unsupervised learning, differential competitive learning, inhibition, clustering.

1. INTRODUCTION

Unsupervised artificial neural networks (UANN) models have offered new approaches to the solution of many pattern recognition tasks. Statistical pattern classification, cluster detection, vector quantization and probability density function estimation are some areas in which UANN have successfully been used [1]. In this paper, we are particularly concerned with cluster detection via centroid estimation.

Clustering a set of p patterns comprises finding m disjoint partitions so that the members of each partition are more similar to each other than to the remaining patterns. Unsupervised learning through competitive neural networks can be used for clustering by discovering the salient statistical features in the input [2], [3].

In competitive learning systems, a set of neurons compete among themselves for the right to respond to an input pattern. The winners of the competition are allowed to modify their weight vectors to become more similar to the input patterns. In effect, competitive UANN models modify their weight vectors in an adaptive fashion in order to estimate the center of gravity of the clusters embedded in the input distribution [4]. Centroid estimation is useful in a number of applications, such as speech and image compression [5].

The goal is to minimize the mean square error E in finding m cluster centroids ($\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_m$) for p patterns ($\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_p$) [5], [6]:

$$E = \sum_{j=1}^{m} \left\{ \left(\frac{1}{p} \right) \sum_{l=1}^{p} M_{jl} \left\| \mathbf{x}_l - \mathbf{w}_j \right\|^2 \right\} \qquad (1)$$

in which $M_{jl} = 1$ if $\mathbf{x}_l$ belongs to cluster m and $M_{jl} = 0$ otherwise.

In this paper, we introduce two competitive models for clustering based on Differential Competitive Learning (DCL) [7], [8]. The first model derives directly from DCL requiring less computational efforts. The second is less sensitive to weight initialization than the others two. In addition, both models can detect subtle aspects of the input pattern distribution dividing the input patterns in clusters with low clustering error.

The paper is organized as follows. In Section 2 we present the original DCL model and discuss some features of the model. In Section 3 a simplified version of the DCL is presented based on the discussion in Section 2. The previous two models have some limitations and in Section 4 we present a novel neural network model with the aim of solving some of the problems. Then, in Section 5 we carry out a number of simulations to illustrate the performance of the models in simple clustering tasks. Finally, in Section 6, we discuss the results of the simulations and the issues for further work.

2. DIFFERENTIAL COMPETITIVE LEARNING

Kong and Kosko [7] and Kosko [8], [9] have proposed the DCL model as a new unsupervised learning paradigm for adaptive vector quantization (AVQ). It has been used in a number of applications such as density estimation and phoneme recognition [7] and printed circuit board inspection [10].

The DCL model comprises two layers of neurons (Figure 1). The input layer has n neurons and the output layer has m neurons. The input neurons propagate the observed inputs, through feedforward weight connections, to the output layer. In both layers, the neurons may have linear or non-linear (sigmoid) activations. Each neuron within the competition layer excites itself and inhibits the others through feedback inhibitory connections.

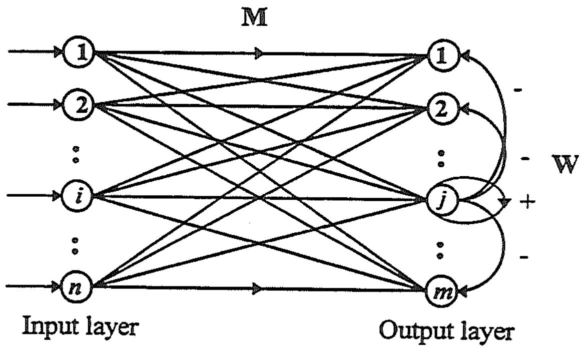

FIGURE 1. TOPOLOGY OF THE LATERALLY INHIBITORY DIFFERENTIAL COMPETITIVE LEARNING MODEL.

The DCL algorithm is summarized below:

(1) Initialize the excitatory weight connections with m samples drawn from the input distribution and initialize the activations of the output neurons:

$$\mathbf{m}_j(0) = \mathbf{x}_j \text{ and } y_j(0) = 0, \quad j = 1, \ldots, m$$

(2) Present a randomly chosen sample $\mathbf{x}(t)$ to the net.

(3) Find the closest ("winner") weight vector $\mathbf{m}_k(t)$ according to the Euclidean norm:

$$\|\mathbf{m}_k(t) - \mathbf{x}(t)\| = \min_j \|\mathbf{m}_j(t) - \mathbf{x}(t)\| \qquad (2)$$

(4) Update the activations of the output neurons through:

$$y_j(t+1) = y_j(t) + \sum_{i=1}^{n} S_i(x_i) m_{ij}(t) + \sum_{r=1}^{m} S_r(y_r) w_{rj}(t) \qquad (3)$$

in which $S_i(x_i)$ and $S_r(x_r)$ are linear or sigmoid-type transfer functions. The inhibitory weight, w_{rj}, connects the output neuron r to the output neuron j.

(5) Update the winning weight vector $\mathbf{m}_k(t)$ through:

$$\mathbf{m}_k(t+1) = \mathbf{m}_k(t) + \eta(t) \operatorname{sgn}[\Delta y_k(t)][\mathbf{x}(t) - \mathbf{m}_k(t)] \quad (4)$$

in which $0 < \eta(t) \ll 1$ is the learning rate and $\operatorname{sgn}[\Delta y_k(t)] = \operatorname{sgn}[y_k(t+1) - y_k(t)]$.

(6) Repeat steps 2-5 continuously for the number of times t_{max} specified by the user.

The learning process takes place only if the neuron activations change. If $\Delta y_k(t) > 0$, the winning weight vector $\mathbf{m}_k(t)$ moves towards the direction of the input and if $\Delta y_k(t) < 0$, the winning weight vector is moved in the opposite direction, away from the input. This signal velocity information provides unsupervised local *reinforcement* during learning [7].

It is worthwhile to note that the winning neuron is chosen based on Euclidean distance measures. Another way to find the winner is via lateral feedback weights [?] by choosing the neuron with highest response for a given input pattern. The use of both mechanism is redundant and brings no special advantage to the model. In the next section, we suggest some simplifications to the DCL model in order to eliminate the redundance in the competition mechanism.

3. THE SIMPLIFIED MODEL

The competition mechanism in DCL determines the winning neuron based on metrical similarity, hence we argue that DCL does not effectively uses its inhibitory connections. Hence, we get rid of the lateral connections introducing the Simplified Differential Competitive Learning (SDCL) neural algorithm. As a result, the activation rule (3) is made simpler and requires less computational efforts than DCL:

$$y_j(t+1) = y_j(t) + \sum_{i=1}^{n} S_i(x_i) m_{ij}(t) \qquad (5)$$

The computational savings in using (5) instead of (3) are of particular interest when the $S_r(y_r)$ signals are sigmoid functions. Sigmoids are computationally more heavier than the distance computations [11]. Moreover, simplicity and locality are the major constraints in designing a neural system. These are the driving requirements if the neural system has to be hard-wired in analog or hybrid VLSI architectures. The absence of lateral connections simplifies VLSI implementation of artificial neural models substantially. The activation rule and the neural network operation require less computational effort and use only the information available locally at the neuron level.

The original DCL and its simplified version update their feedforward excitatory weights according to the competitive learning rule [2]. Winner-take-all (WTA) behavior has the advantage of simplicity, but also has some drawbacks including underutilized output neurons (dead units), which reduces the representational ability of competitive neural networks [1], [3], [4]. In the next section, we propose a neural algorithm in order to show how Euclidean distance can be used together with lateral inhibition to correctly estimate cluster centroids.

4. THE GENERALIZED MODEL

This new model will be called Generalized DCL (GDCL). GDCL has the same topology as DCL. The inputs are non-linearly transformed according to:

$$a_j(t) = \exp\left(\left\|\mathbf{x}(t) - \mathbf{m}_j(t)\right\| / 2\rho^2\right) \tag{5}$$

in which $\mathbf{x}(t)$ is the input vector, $\mathbf{m}_j(t)$ is the weight vector associated with the jth output and ρ is the selectivity parameter. This model has no "winner", that is, all output neurons are updated at each presentation of an input pattern. It is worth noting that simultaneous updating of all the neurons should accelerate the convergence of the network and helps to avoid dead units [1], [6].

Because of the new strategy for weight modification, the simple competitive learning rule can be no longer used. Hence, some kind of correlation-based learning [3], [4] rule must be used. The weights are adjusted through:

$$\mathbf{m}_j(t+1) = \mathbf{m}_j(t) + \eta^{*}(t)y(t)\left[\mathbf{x}(t) - \mathbf{m}_j(t)\right] \tag{6}$$

in which $\mathbf{m}_j(t)$ is the feedforward weight connection between input i and output neuron j, $\mathbf{x}(t)$ is the input vector, $y_j(t)$ is the output transfer function defined as:

$$y_j(t) = g\left(\sum_{r=1}^{m} w_{rj} a_r(t)\right) = g\left(\alpha a_j(t) - \beta \sum_{r \neq j} w_{rj} a_r(t)\right) \tag{7}$$

in which w_{kj} is connection from the output unit r to unit j (w_{jj}=α and w_{rj}=-β). The transfer function $g(v) = 1$ if v>1, $g(v) = 0$ if v<0 and $g(v) = v$ otherwise. The signals $y_j(t)$ are normalized so that $\Sigma_j\, y_j(t) = 1$. The extended learning rate $\eta^{*}(t)$ is defined as $\eta^{*}(t) = \eta(t)\operatorname{sgn}\left[y(t+1) - y(t)\right]$.

In order to avoid different neurons to respond to the same input pattern, we suggest the use of fixed lateral inhibitory connections. It is worth emphasizing that here the lateral connections play an effective role. This is not the case for DCL. Computer simulations illustrate that the combination of these mechanism is able to improve the performance of DCL models in clustering tasks.

5. SIMULATION AND RESULTS

We simulated the models, written in ANSI C, in a SUN workstation ULTRA-1. For all tests n = 2, m = 4. In the first experiment, the pattern set consists of 2000 two-dimensional Gaussian-distributed pattern vectors with standard deviation σ = 0.09, and with centroids or modes at (1/3, 1/3), (1/3, 2/3), (2/3, 1/3) and (2/3, 2/3).

Figure 1a shows the evolution of training for the SDCL model where $\eta(t)$ = 0.01 and the initial weight vectors were {(0.48, 0.48), (0.52, 0.48), (0.52, 0.52), (0.48, 0.52)}. The SDCL weights estimated the centroids of the distributions correctly. The final weights after 5000 iterations were {(0.329, 0.332), (0.673, 0.328), (0.662, 0.662), (0.322, 0.659)}.

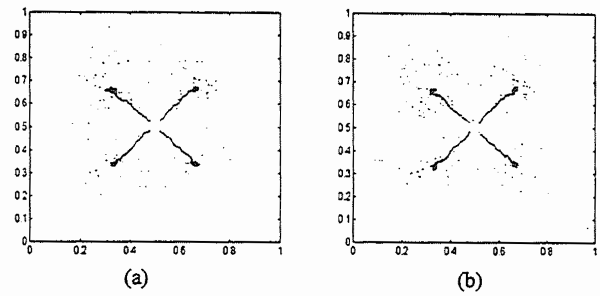

FIGURE 1. CENTROID CONVERGENCE OF SYNAPTIC VECTORS FOR: (a) SDCL AND (b) GDCL. THE INPUTS ARE LINEARLY TRANSDUCED.

We repeat the previous test to evaluate the ability of the GDCL model in estimating centroids. The same training pattern set and initial weights of the previous experiment are used. For this model, $\eta(t)$ = 0.01 and ρ = 0.025, β = -2 and α = 1. Figure 1b shows the results of the simulation. After 5000 iterations the final weights were {(0.323,0.330), (0.660,0.337), (0.324,0.660), (0.672,0.666)}.

Figure 2 shows the trajectories of one of the SDCL and GDCL synaptic vectors reaching the centroid at (2/3, 1/3).

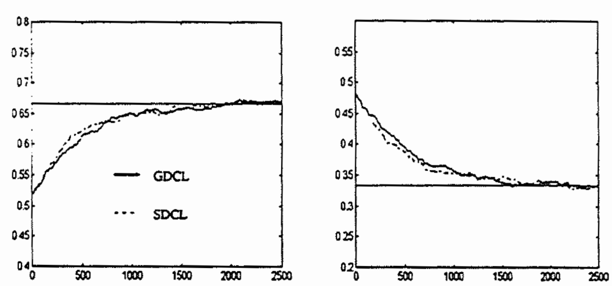

FIGURE 2. TRAJECTORIES OF ONE OF THE SDCL AND GDCL WEIGHT VECTORS REACHING THE CENTROID (2/3, 1/3).

Both models have fast convergence, reaching the centroids after 1600 training iterations. The original DCL was unable to converge to the centroids for this pattern set.

In the third experiment, we evaluate the performance of the models on a simple clustering task [6]. For this test, the pattern set comprises 24 two-dimensional vectors:

$\mathbf{v}_1$ = (0.18, 0.28)	$\mathbf{v}_7$ = (0.23, 0.28)	$\mathbf{v}_{13}$ = (0.27, 0.20)	$\mathbf{v}_{19}$ = (0.45, 0.24)
$\mathbf{v}_2$ = (0.19, 0.26)	$\mathbf{v}_8$ = (0.24, 0.22)	$\mathbf{v}_{14}$ = (0.27, 0.18)	$\mathbf{v}_{20}$ = (0.47, 0.26)
$\mathbf{v}_3$ = (0.20, 0.28)	$\mathbf{v}_9$ = (0.23, 0.20)	$\mathbf{v}_{15}$ = (0.27, 0.16)	$\mathbf{v}_{21}$ = (0.47, 0.24)
$\mathbf{v}_4$ = (0.18, 0.30)	$\mathbf{v}_{10}$ = (0.23, 0.18)	$\mathbf{v}_{16}$ = (0.34, 0.22)	$\mathbf{v}_{22}$ = (0.47, 0.22)
$\mathbf{v}_5$ = (0.22, 0.26)	$\mathbf{v}_{11}$ = (0.25, 0.20)	$\mathbf{v}_{17}$ = (0.43, 0.26)	$\mathbf{v}_{23}$ = (0.47, 0.20)
$\mathbf{v}_6$ = (0.22, 0.24)	$\mathbf{v}_{12}$ = (0.25, 0.20)	$\mathbf{v}_{18}$ = (0.44, 0.28)	$\mathbf{v}_{24}$ = (0.49, 0.22)

In the trained networks, the patterns are grouped in classes according to the response of the output neurons.

The weight vector of each output neuron represents a cluster centroid. Each input pattern belongs to the cluster identified by the output neuron with the highest response.

For this experiment the learning rates are decresead according to: $\eta(t) = \eta_0(1 - t/t_{max})$ in which $\eta_0 = 1$ and $t_{max} = 2500$. The initial weights were chosen at random from the pattern set and $\beta = -0.9$ and $\alpha = 1.3$. We compare SDCL and GDCL with the WTA network [2]. The clusters found by SDCL and GDCL are $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$, $\{v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}\}$, $\{v_{16}\}$ and $\{v_{17}, v_{18}, v_{19}, v_{20}, v_{21}, v_{22}, v_{23}, v_{24}\}$ which can be visualized by the Voronoi tesselation [3] shown in Figure 3. GDCL has achieved this result in 40 out of 55 attempts, SDCL needs 29 out of 55 and WTA in only 13 out 55 trials. In most cases, WTA has tried to shift the weight vectors to oversampled regions (Figure 4) finding the following clusters: $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$, $\{v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}\}$, $\{v_{16}, v_{17}, v_{18}\}$ and $\{v_{19}, v_{20}, v_{21}, v_{22}, v_{23}, v_{24}\}$.

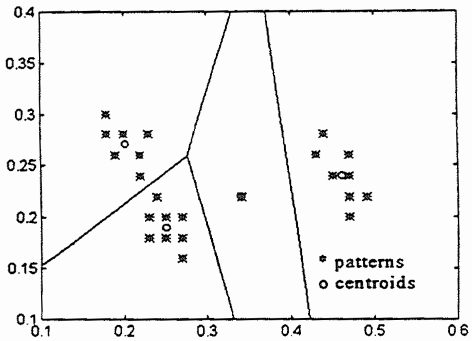

FIGURE 3. VORONOI DIAGRAM REPRESENTING CLUSTERING RESULTS FOR SDCL AND GDCL.

The result shown in Figure 3 suggests that SDCL and GDCL have paid a close attention to pattern v_{16} which is an outlier and assigned a weight vector to represent that.

It is important to note that the original DCL model was unable to cluster the input patterns for the training trials. In fact, this simulation suggests that the negative feedback through lateral connections in DCL generates some kind of instability.

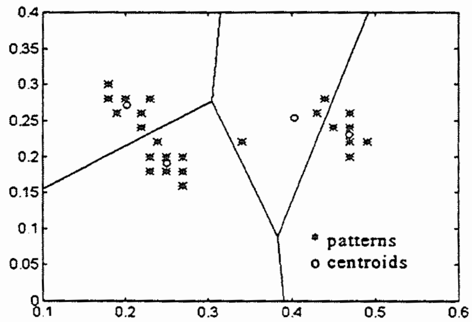

FIGURE 4. VORONOI DIAGRAM REPRESENTING CLUSTERING RESULTS FOR WTA.

This experiment shows that GDCL is less sensitive to weight initialization than SDCL. Both, GDCL and SDCL converge faster than WTA. For the clusters shown in Figure 3 the final clustering error calculated through [illegible] is 0.00070 and for those in Figure 4 the error is 0.000[illegible]

6 CONCLUSION AND FURTHER WORK

We have suggested a set of modifications to the original DCL model introducing the SDCL and GDCL models in order to enhance the performance in clustering tasks. In SDCL, we assessed the role played by inhibitory lateral weights in the competition mechanism. We get rid of the inhibition arguing that the use of both Euclidean distance relations and lateral connection is redundant for determining the winning neuron. In GDCL, we showed how to use Euclidean distances and inhibition effectively updating all the output neurons. The simulations illustrate that SDCL and GDCL are more stable than the original DCL always converging to cluster centroids. Moreover, the GDCL model is less sensitive to weight initialization than SDCL and WTA. SDCL and GDCL also achieve lower clustering error than WTA for certain input patterns.

Further work must be developed in order to simulate the SDCL and GDCL models on real world applications such as speech and image processing.

7 REFERENCES

[1] S.J. Kia and G.G. Coghill, Unsupervised clustering and centroid estimation using dynamic competitive learning, *Biological Cybernetics*, 67, 1992, 433-443.

[2] D.E. Rumelhart and D. Zipser, Feature discovery by competitive learning, *Cognitive Sci.*, 9, 1985, 75-112.

[3] J. Hertz, A. Krogh and R.G. Palmer. *Introduction to the Theory of Neural Computation*. (Redwood City, CA: Addison-Wesley Publishing Company, 1991).

[4] M.H. Hassoun, *Fundamentals of Artificial Neural Networks* (A Bradford Book: MIT Press, 1995).

[5] S.C. Ahalt, A.K. Krishnamurthy, P. Chen and D.E. Melton, Competitive learning algorithms for vector quantization, *Neural Networks*, 3, 1990, 277-290.

[6] S.J. Kia and G.G. Coghill, High performance clustering using dynamic competitive learning, *Electronics Letters*, 28(18), 1992, 1753-1755.

[7] S.G. Kong and B. Kosko, Differential competitive learning for centroid estimation and phoneme recognition, IEEE *Trans. Neural Networks*, 2(1), 1991, 118-124.

[8] B. Kosko, *Neural Networks and Fuzzy Systems: a Dynamical Systems Approach to Machine Intelligence* (New Jersey: Prentice Hall, 1992).

[9] B. Kosko, Unsupervised learning in noise, IEEE *Transactions on Neural Networks*, 1(1), 1990, 44-5[illegible]

[10]Madhav Moganti, PCB inspection using differential competitive learning and Fuzzy associative memories, *Master thesis*, University of Missouri-Rolla, May 94, [illegible]

[11]T. Kohonen, G. Barna and R. Chrisley, Statistical pattern recognition with neural networks: Benchmarking studies, *Proceedings of the IEEE International Conference on Neural Networks*, San Diego, 1988.