2^{ND} International Conference on Computer and Information Science(ICIS 2002)

Dynamic Competitive Learning Neural Network

Seongwon Cho*, Jaemin Kim*, Cheol Su Lim and Robert Marks*****

*** School of Electronic and Electrical Engineering, Hongik University, 72-1 Sangsu-dong, Mapo-gu, Seoul 121-791, Korea**
**** Department of Computer Engineering, Seokyeong University, 16-1 Jungneung-dong, Sungbuk-gu, Seoul, 136-704, Korea**
***** Department of Electrical Engineering, University of Washington Seattle, Washington 98195, USA**

swcho@hongik.ac.kr jaemin@hongik.ac.kr cslim@skuniv.ac.kr

ABSTRACT

In this paper, a new competitive learning algorithm called Dynamic Competitive Learning (DCL) is presented. DCL is a supervised learning method that dynamically generates output neurons and initializes automatically the weight vectors from training patterns. It introduces a new parameter called LOG (Limit of Grade) to decide whether an output neuron is created or not. If the class of at least one among LOG number of nearest output neurons is the same as the class of the present training pattern, then DCL adjusts the weight vector associated with the output neuron to learn the pattern. If the classes of all the nearest output neurons are different from the class of the training pattern, a new output neuron is created and the given training pattern is used to initialize the weight vector of the created neuron. The proposed method is significantly different from the previous competitive learning algorithms in the point that the selected neuron for learning is not limited only to the winner and the output neurons are dynamically generated during the learning process. In addition, the proposed algorithm has a small number of parameters, which are easy to be determined and applied to real-world problems. Experimental results for indicate the superiority of DCL in comparison to the conventional competitive learning methods.

Keywords : Dynamic Competitive Learning, Neural Network

1. Introduction

Recently, neural network approach is highlighted because of its merits of parallel processing, robustness, generalization capability compared to other approaches. Competitive learning has been used for rapid training of neural networks since it is of low computational complexity. Typical competitive learning models such as Simple Competitive Learning (SCL) [1], Frequency Sensitive Competitive Learning (FSCL) [2][3], and Self-Organizing Feature Map (SOFM) [4][5], Learning Vector Quantization (LVQ) [6][7] have the static structure of neural networks. Their learning performance extremely depends on the. distribution of initial weight vectors and the number of output neurons, which has to be determined before learning.

Contrariwise, as a typical model using a changeable structure of neural network, Adaptive Resonance Theory (ART) [8] holds the plasticity by consistently integrating the newly acquired knowledges to the whole knowledge base while keeping the previous knowledges. ART can simulate human learning behavior. However, its structure and learning algorithm is so complex and the adjustment of numerous learning parameters is so difficult, and thus it has some difficulties to apply to practical problems.

Hence, in this paper, we propose a new supervised competitive learning neural network called dynamic competitive learning (DCL). In the proposed neural network, the output neurons are dynamically generated during the learning process and the initial weight vectors are automatically selected from the training patterns. The major differences between DCL and the conventional competitive learning models are the point that DCL introduces a new learning parameter, Limit of Grade (LOG), to decide the generation of output neurons and the selected neuron for learning is not limited only to the winner. In DCL, the output layer is changeable and there is a small number of parameters which have to be determined by users.

The paper consists of 4 sections. In Section II, DCL is described. Experimental results are presented in Section III. Section IV is conclusions.

2. DCL Neural Network

1. Basic Concept

DCL neural network consists of 2 layers, input layer and output layer, as with the conventional competitive learning neural network. DCL is similar to SCL and FSCL in the point that only one weight vector is updated for the given training vector. However, in DCL the neurons of output layer are dynamically generated during learning and the next nearest output neurons are allowed to be updated if the class of the winning neuron with the minimum Euclidean distance is different from the class of the training vector incorrectly.

The proposed method decides to generate output neurons during learning process by introducing a new learning parameter, LOG. DCL first checks whether the class of the winner is the same as the training vector or not. If the class are identical, the winner learns. Otherwise the second nearest output neuron becomes a candidate for learning and its class is checked. The same procedure is repeated for the next nearest output neurons. LOG represents the number of the candidates for learning If all the candidates fail to learn, a new output neuron is generated and the given training vector is used as the initial weight vector for the output neuron. As learning proceeds, LOG of each class increases monotonically for preventing the number of generated output neurons to diverge. The terminologies used in this paper are as follows:

t : the present learning time
n : total number of the generated output neurons
J : total number of classes
j : index for class, $1 \leq j \leq J$
I : index for the output neuron, $1 \leq i \leq n$
$\mathbf{x}_j$: input vector belonging to class j
LOG_j : LOG of class j
ε : gain for LOG parameter
n_j : number of output neurons assigned to class j
$\mathbf{w}_i$: weight vector for output neuron i
u_i : labelled class of output neuron i
$\mathbf{u}$: vector consisting of the classes of all the output neurons,

i.e., $\mathbf{u} = [u_1, u_2, \Lambda, u_n]^T$

c_r : the rth nearest output neuron (the winner is the first nearest output neuron)

$\alpha(t)$: learning rate at time t

2. Learning Algorithm

DCL algorithm is composed of parameter initialization, selection of candidates for learning, weight vector update, generation of a new output neuron, and parameter adjustment. The final weight vector and the label vector **u** can be obtained after all the learning and relabelling process. More detailed algorithm can be described as follows:

Step 0. Initialize parameters:

$$t = 0, n = 0$$
$$\alpha(0) \in (0,1)$$
$$n_j = 0 \ for \ j = 1,2,\Lambda, J$$
$$LOG_j = 1 \ for \ j = 1,2,\Lambda, J$$

Step 1. While stopping condition is false, do Steps 2-8.

Step 2. For each training input vector $\mathbf{x}_i$, do Steps 3-7.

Step 3. For r = 1,…,LOG_j

a. Find Cr such that

$$\left\| x_{j-w_{c_r}} \right\| = \min_{i \neq c_{1,\Lambda, c_{r-1}}} \{\left\| x_j - w_i \right\|\}$$

b. If $u_{c_r} = j$, update the weight vector and go to Step 5:

$$w_{c_r}(new) = w_c(old) + \alpha(t) \cdot [x_j - w_{c_r}(old)]$$

c. Next r.

Step 4. If $u_{c_r} \neq j$ for all r, then create a new output neuron and initialize the weight vector:

$$n = n + 1, \ n_j = n_j + 1.$$
$$w_n(new) = x_j, u_n = j$$

Step 5. Adjust limit of grade LOG_j

Step 6. Reduce learning rate alpha (t)

Step 7. Next t ;

Step 8. Test stopping condition :

The condition may specify a fixed number of iterations.

Step 9. Relabelling.

In DCL the weight vector for an output neuron can be updated using several training vectors with different classes after initial labeling. Thus, after learning ends, relabeling is necessary for assigning the optimal class of the output neuron and removing dead neurons. DCL uses LOG as a parameter for output neuron generation. LOG takes the monotonically increasing form such as (1) or (2) to prevent the divergence of generated neurons.

$$LOG_j = 1 + \varepsilon \cdot n_j \qquad (1)$$

$$LOG_j = 1 + \varepsilon \cdot n \qquad (2)$$

LOG parameter are associated with the number of output neurons and the gain ε is between 0 and 1 $(0 < \varepsilon \quad 1)$. The equation of LOG parameter and its gain are predetermined before learning, and the number of generated neurons can be adjusted accordingly. As ε approximates to 1, the small number of output neurons is generated. In the contrary, as ε approximates to 0, relatively large number of output neurons are generated. Especially, in case of ε= 1 in (2) the number of generated output neurons is the same with the number of classes. In case of ε= 0, only the winner learns and DCL is similar to LVQ except that in DCL the weight vector is not updated when the training vector is classified incorrectly.

DCL dynamically generates output neurons during learning using the changeable structure of neural network such as in ART instead of static structure such as in SCL, FSCL, SOFM and LVQ. In this sense, it is not required to predetermine the number of output neurons and to initialize weight vectors. ART model is similar to DCL in the sense that it also has changeable structure of neural network, but ART model, which has both bottom-up and top-down vector, performs two-way learning and clusters input by using unsupervised competitive learning. Contrariwise, DCL performs only forward learning and its implementation is simple because it has only one parameter, LOG.

Since DCL allows the weight vector for one of nearest output neurons to be updated or a new output neuron to be generated, it makes the generation probability of dead neurons low, and prevents the number of output neurons from diverging by introducing a monotonically increasing form of LOG. DCL constructs more sophisticated classifier by assigning many output neurons near the decision surfaces that misclassification occurs frequently.

3. Experimental Results

1. Recognition of remote sensing data

In this section, to check applicability and to demonstrate the superior performance of the DCL in the real problems, experiments with a remote sensing data were conducted in order to compare the proposed DCL with the other neural networks.

A. Data Set

Remote sensing data are used to get an information concerning terrestrial, environmental, vegetation, resource and so on from spatial and spectral information of wide area, which are periodically gathered by artificial satellites or airplanes equipped with sensor.

The data set for the experiment is a multi-spectral earth observational remote sensing data called Flight Line C1 (FCL1) [9] [10]. The geographical location of the FCL1 is the southern part of Tippecanoe Country, Indiana, U.S.A. This multi-spectral image was collected with an airborne scanner in June 1966 at noon time.

The FLC1 consists of 12 band signals. Each point of one spectral image represents one of 256 gray levels. In this experiment, only 8 spectral bands out of 12 bands are used. Each data vector is generated by concatenating eight corresponding band signal values together.

8 dominant farm products (alfalfa, corn, oats, red clover, soybean, wheat, bare soil, rye) are chosen to represent 8 particular classes. Training and testing with the networks were done with 200 signal samples per class and 375 other signal samples per class. Therefore, the total numbers of the training and the testing samples are 1600 and 3000, respectively.

B. Simulation Results

DCL generated 89 output neurons. In order to have a fair comparison, the number of output neurons used was 89 for FSCL and 10 x10 for SOFM. In the case of DCL, the learning rate coefficient was chosen as

$$\alpha(t) = 0.3 \cdot (1 - \frac{t}{Total\ number\ of\ iterations})$$

and the gain for the LOG parameter was chosen as ε= 0.2.

In the case of FSCL and SOFM, the learning rate coefficients were chosen as

$$\alpha(t) = 0.9 \cdot (1 - \frac{t}{Total\ number\ of\ iterations})$$

and the first 89 or 100 training patterns were used to initialize the weight vectors.

Table 1 and Table 2 show the experimental results with the FCL1 data set. The training and testing accuracies of the DCL were 97.667% and 93.856%, respectively. The training and testing accuracies of FSCL were 94.125% and 92.217%, respectively. The training and testing accuracies of SOFM were 92.677% and 91.172%, respectively.

Table 1 Recognition Rates of DCL

Epochs	No. of neurons	training(%)	testing(%)
50	82	97.938	93.9
60	84	97.875	93.7
70	84	97.688	93.9
80	82	97.438	93.933
90	89	97.688	93.867
100	84	97.438	93.833
Average	84.1675	97.667	93.856

Table 2 Recognition Rates of FSCL and SOFM

Epoch	FSCL		SOFM	
	training(%)	testing(%)	training(%)	testing(%)
50	94	91.733	92.688	92.267
60	94.75	92.167	92.75	90.267
70	93.125	91.333	92.438	90.2
80	95.312	92.733	92.625	92
90	93.25	92.767	92.5	91.1
100	94.312	93.567	93.062	91.2
Average	94.125	92.217	92.677	91.172

2. Handwritten Numeral Recognition

Pattern recognition system is generally divided into 2 parts. First, feature extraction which computes the parameters that can provide a maximum information for the given pattern. Second, classifier which recognizes the object using extracted features. In the recognition of handwritten letters, we might want to recognize patterns without being influenced by distortions or transformations. For handwritten numeral recognition independent of the position or orientation of the pattern or its size, the Fourier descriptors were used as the features.

A. Data Set

Experimental data are 2000 handwritten Arabic numbers (0, 1, 2, 3, 4, 5, 6, ,7, 8, 9) with various positions, sizes and orientations. They were made by 100 persons and acquired using a CCD camera and an image grabber. Each pattern has 128 x 128 pixels and 128 gray levels. 14 Fourier descriptors were computed for each pattern. Training and testing with the networks were done with 100 patterns per class and 100 other patterns per class.

B. Simulation Results

The number of output neurons, learning rate and LOG parameter are set to the same as experiments with remote sensing data. The initial learning rates of DCL, LVQ and SOFM were chosen as 0.3, 0.4 and 0.9, respectively. In the case of LVQ, initialization of the weight vectors were carried out by choosing the training patterns belonging to each class. In the case of SOFM, the first 100 training patterns were used for the initial weight vectors. The recognition rates of DCL, LVQ and SOFM are shown in Tables 3 and 4.

Table 3. Recognition Rate of DCL

Epochs	No. of neurons	training(%)	testing(%)
50	110	97.4	84.4
60	110	97.6	84.6
70	110	97.8	84.6
80	110	97.8	84.4
90	111	97.8	84.6
100	111	97.8	84.6
Average	110.333	97.7	84.533

Table 4. Recognition Rate of LVQ and SOFM

Epoch	LVQ		SOFM	
	training(%)	testing(%)	training(%)	testing(%)
50	92.2	83.4	85.6	76.8
60	92	82.2	86.4	76
70	91.8	82.6	86.2	75.2
80	91.6	82.6	85.8	79.2
90	91.8	82.6	86.6	78.2
100	92	82.8	86.8	79.2
Average	91.9	82.7	86.233	77.433

4. Conclusions

We presented a new competitive learning neural network called DCL. It dynamically generates output neurons instead of determining before learning, and the initial weight vectors are selected from training patterns. DCL has the following features: 1) Learning algorithm itself includes the initialization process of weight vectors; 2) it has simple structure, learning algorithm and small number of parameters compared to the conventional neural networks with changeable structure; 3) the number of dead neurons can be reduced.

For the best recognition, appropriate LOG gain needs to be chosen according to the distribution of given patterns. For the performance evaluation of the proposed method, we conducted the experiments using practical remote sensing data and handwritten numeral data, and demonstrated its superiority in comparison to FSCL, LVQ and SOFM.

References

[1] D.E. Rummelhart, and D. Zipser, "Feature Discovery by Competitive Learning", In Rummelhart, D.E. and McClelland, J.L. eds Paralled Distributed Processing, Cambridge, MA: MIT Press, 1988.

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

[3] D. DeSieno, "Adding a Conscience to Competitive Learning", Proc. IEEE International Conference on Neural Networks (ICNN-88, San Diego, California), pp. I117-I124, 1988.

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

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

[6] T. Kohonen, "Learning Vector Quantization", Neural Networks, vol. 1, suppl. 1, pp. 303, 1988.

[7] L. Fausett, Fundamentals of Neural Networks, Prentice Hall, 1994.

[8] G.A. Carpenter, and S. Grossberg, "The ART of Adaptive Pattern Recognition by Self-Organization Neural Network", Computer, pp.77-88, March 1988.

[9] Seongwon Cho, Okan K. Ersoy, and Mark Lehto, "Parallel, Self-Organizing, Hierarchical Neural Networks with Competitive Learning and Safe Rejection Schemes", IEEE Transactions on Circuits and Systems, Vol 40, No 9, pp. 556-566, September 1993.

[10] Seongwon Cho, "Parallel, Self-Organizing, Hierarchical Neural Networks with Fuzzy Input Signal Representation, Competitive Learning and Safe Rejection Schemes", Ph.D. Thesis, Purdue University, July 1992.