CLUSTERING OF PATTERNS USING RADIAL BASE FUNCTION NETWORKS

BOGDAN M. WILAMOWSKI and KARINA VIEIRA
Dept. of Electrical Engineering, University of Wyoming, Laramie, WY, USA

ABSTRACT:
A supervised pruning algorithm, an approach to train the neural networks to prune hidden layers, was devised to reduce the total number of neurons. It first sets the number of clusters equal to the number of input patterns, then, checks the possible overlap of the output and by taking the centroid of the input patterns a new layer is generated. This approach is performed iteratively until the minimum number of clusters is obtained. Compared to the ART and to the subtractive clustering training technique, this process is much more efficient in the clustering of patterns. Algorithm can operate in both unsupervised and supervised modes.

INTRODUCTION

Several attempts were made for automatic pattern clustering. The most known are ART developed by Grossberg and Carpenter and WTA [1][2] developed by Kohonen [3][4]. Both algorithms work in an unsupervised mode and they are able to group similar patterns into clusters. In the Kohonen self-organizing feature maps, the number of neurons is fixed and due to neighbor interaction after training, certain neighborhoods are responding stronger than others for certain group of patterns. In the ART approach each cluster is acquiring a designated neuron and the number of separate clusters (neurons) depends on the vigilance parameter. The ART network is actually composed of the main network and the short term memory network. The interaction of these networks in a rather complicated way leads to the unsupervised pattern clustering. Many modifications of the ART algorithm were developed, all of them follow the basic scheme. A similar result as in the ART network can be obtained by using the recently developed subtractive clustering approach [5][6], which is much simpler and more computationally efficient. The subtractive clustering method is an extension of the mountain clustering method and it assumes that each data point is a potential cluster center and calculates each data point based on the density of the surrounding data points.

All mentioned above algorithms use the unsupervised training algorithm and therefore the information about pattern category is not used during the clustering process. Such unsupervised training cannot distinguish patterns of different categories and may group patterns form different categories into one cluster. To illustrate this undesired effect let us consider a simple example with 20 patterns of two categories shown in Fig. 1(a). When the information about categories is lost

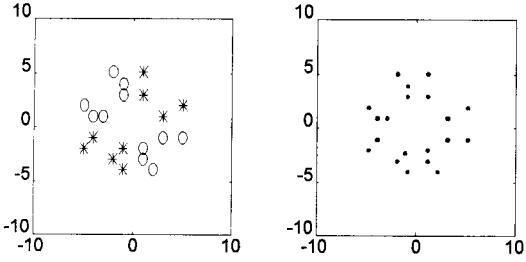

Figure 1. Twenty patterns example (a)with category information, (b)without category information.

(unsupervised training) as shown in Fig. 1(b), then most likely any of discussed above algorithm will lead to a wrong clustering.

The purpose of this study was to develop an algorithm that will automatically cluster patterns only of the same category. The algorithm starts with the number of RBF neurons equal to the number of patterns and then the layer is pruned.

THE ALGORITHM

Let us consider a two layer network as shown in Fig. 2. The hidden layer consist of RBF neurons with the following activation function:

$$y_i = \exp\left[\frac{(\mathbf{x}-\mathbf{s}_i)^T * (\mathbf{x}-\mathbf{s}_i)}{r_i^2}\right] \tag{1}$$

where $\mathbf{x}$ is the input vector, $\mathbf{s}_i$ is the stored pattern in *i-th* neuron, and r_i is the radius of attraction. The output layer consist of standard sigmoidal neurons performing the "OR" operation for each category. During the initialization, the number of hidden neurons is chosen to be equal to the number of patterns and $\mathbf{s}_i = \mathbf{x}_i$ for $i=1,2..p$. Where p is the number of patterns.

The central part of this algorithm is aimed at autonomously minimizing the total number of neurons in the hidden layer. The pruning algorithm of this study computes the matching score reflecting the degree of closeness of each pattern without misclassification of the input vectors. The task of this technique, of pattern clustering is to automatically group input vectors by checking possible overlap of the output so that a new layer is generated until the minimum number of neurons in the hidden layer is obtained.

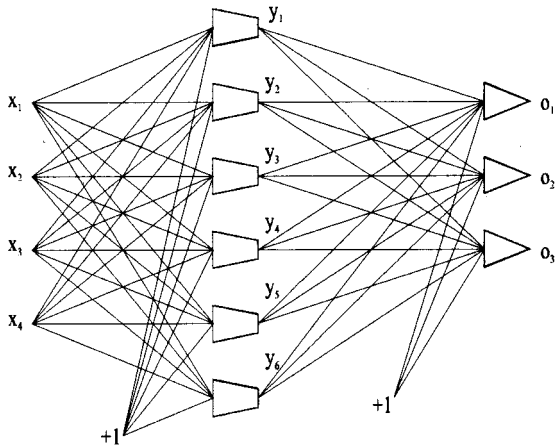

Figure 2. Two layer neural network. First layer with RBF neurons, and second layer with sigmoidal neurons performing "OR" operation.

The algorithm follows the steps:

Step 1: Initializations. The number of neurons is set equal to the number of input patterns, and the N-dimensional stored vector $\mathbf{s}_i$ is initialized equal to the input vector $\mathbf{x}_i$

$$\mathbf{s}_i = \mathbf{x}_i \quad \text{and} \quad r_i = r \qquad \text{for} \quad i \in (1,n) \tag{2}$$

Each neuron i associated with one input pattern and also each neuron is assigned to a category.

Step 2: Pattern application. A pattern $\mathbf{x}_j$ applied to the network input and outputs of the hidden layer are computed using equation

$$y_i = \exp\left[\frac{\left(\mathbf{x}_j - \mathbf{s}_i\right)^T * \left(\mathbf{x}_j - \mathbf{s}_i\right)}{r^2}\right] \qquad \textit{for} \quad i = 1,2\ldots p \tag{3}$$

Step 3: Find two winning neurons. Find two neurons with the maximum output. If both neurons belong to the same category then go to Step 4 , otherwise return to Step 2 where a new pattern is applied.

Step 4: Neuron elimination. The winning neuron is eliminated and for the second neuron the new stored pattern is computed as an average of stored patterns of two winners $\mathbf{s}_1$ and $\mathbf{s}_2$:

$$\mathbf{s}_2 = 0.5\left(\mathbf{s}_1 + \mathbf{s}_2\right) \tag{4}$$

After the elimination of the winner and the modification of a pattern stored in the second neuron the algorithm returns to Step 2.

Once all patterns were applied to the network the algorithm can be terminated. This process can be also repeated if further hidden layer pruning is required. Many modifications of this algorithm were investigated, but this one is the simplest and the most efficient.

This algorithm was developed for supervised clustering, but it can be also used in an unsupervised mode of operations. In this case the condition in the Step 3 has to be changed. Instead of checking if both winning neurons are in the same category the algorithm checks if the difference in outputs y_1 and y_2 of the two winners are smaller than certain parameter *delta* from range of (0,1).

$$|y_1 - y_2| < delta \tag{5}$$

The small value of *delta* leads to limited pruning, because only very similar patterns are combined. The small value of *delta* corresponds to the large value of the vigilance parameter in the ART algorithm.

ILLUSTRATIVE EXAMPLES

A couple a simple two-dimensional examples are described below. Two-dimensional examples were chosen so the operation of the algorithm can be easily visualized. At first, the supervised and the unsupervised approaches were applied to the previously mentioned example shown in Fig. 1. The results are shown in Fig. 3. A correct clustering were obtained in the supervised mode as shown in Fig. 3(c). And as it was expected the unsupervised training resulted in a wrong clustering as shown in Fig. 3(d) with *delta*=0.6. When the ART and subtractive clustering methods are used the results were equally wrong. This is because the unsupervised training have their limitations.

Next example with 15 patterns belonging to 3 categories are presented. The initial patterns are shown in Fig. 4. For 9 applied patterns out of 15 the algorithm eliminated a neuron. This 9 reduction steps are also shown in Fig. 5. Note that on Fig. 5 the stored patterns of existing hidden neurons are shown. As the result of pruning, only 6 neurons are left in the network.

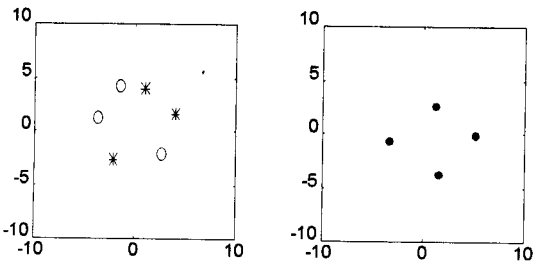

Figure 3. Clustering results of the patterns shown in Fig. 1: (a) in supervised mode, (b) in unsupervised mode (*delta*=0.6)

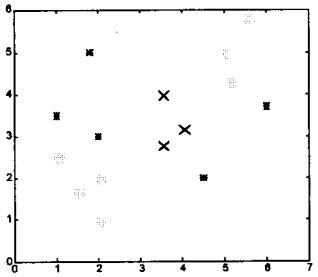

Figure 4. Example with 15 patterns belonging to three categories.

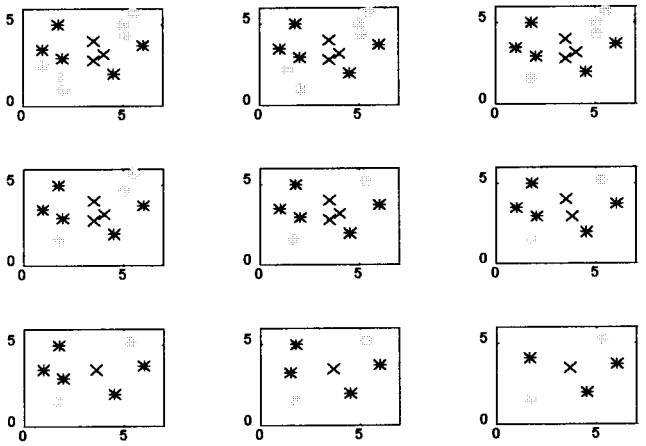

Figure 5. Nine of fifteen clustering steps of the patterns shown in Fig. 4.

Another example consists of 60 patterns of 5 categories as shown in Fig. 6 (a). After pruning, the network was able to reduce the network to 14 neurons as it is shown in Fig. 6 (b). Two additional training processes on the same set lead to solutions shown in Fig 6 (c) and Fig. 6 (d). Further training did not result in a change in the network structure. Note that excessive repetition of training leads to simple structure but some clusters are actually lost as one can see it in Fig. 6 (d). The same set of 60 patterns can be trained in the unsupervised mode. Results are shown in Fig. 7. Various clustering were obtained depending on the value of the *delta* parameter. This is illustrated in Figs 7 (b) to 7(d). Note, that the obtained results do not reassemble the actual clusters as shown in Fig. 6 (a).

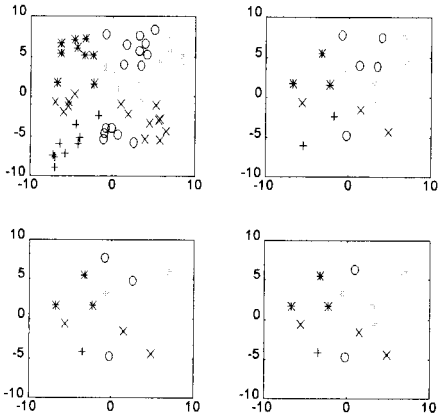

Figure 6. Example with 60 patterns of 5 categories trained in the supervised mode; (a) original patterns, (b) clusters after one training cycle, (c) clusters after two training cycles, and (d) clusters after three training cycles

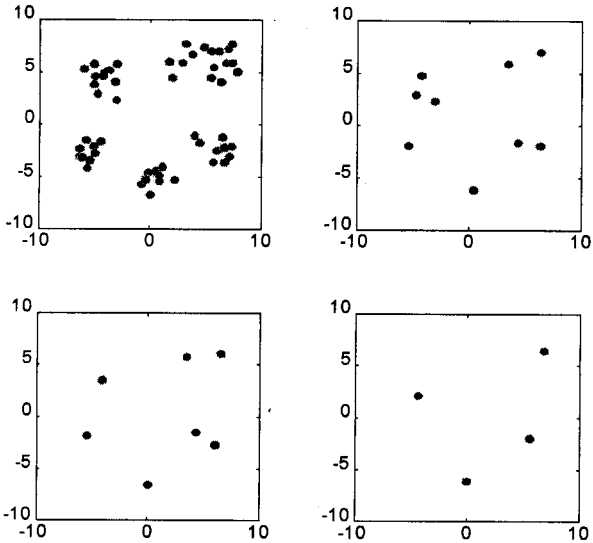

Figure 7. Example with 60 patterns of 5 categories trained in the unsupervised mode; (a) original patterns, (b) clusters obtained using one iteration with delta=0.5 (c) clusters obtained using one iteration with delta=0.6, and (d) clusters obtained using one iteration with delta=0.8

CONCLUSIONS

It can be noted, that the algorithm presented in this paper can be used in both supervised and unsupervised mode of training. The supervised training leads to much better results. The algorithm is very fast and efficient The significant reduction of neurons in the hidden layer is achieved without misclassification of the input patterns. When applied in the unsupervised training mode and can efficiently substitute the ART or subtractive clustering method.

REFERENCES

Carpenter, G. A. and S. Grossberg. "A massively parallel architecture for a self-organizing neural pattern recognition machine", *Computer Vision, Graphics, and Image Processing*, vol. 37, pp. 54-115, 1987.

Carpenter, G. A. and S. Grossberg. *Pattern Recognition by Self Organizing Neural Networks*, MA: MIT Press 1991

Chiu, S. "Fuzzy Model Identification Based on Cluster Estimation," *Journal of Intelligent and Fuzzy System,* Vol. 2, No. 3, pp. 267-278, Sep. 1994.

Kohonen T. A Simple Paradigm for the Self-Organized Formation of Structured Feature Maps," in *Competition and Cooperation in Neural Nets*, Lecture Notes in Biomathemetics, ed. S. Amari, M. Arbib, Berlin: Springer-Verlag 1982.

Kohonen T. The Self-organizing Map." *Proc. IEEE* 78(9): 1464-1480.

Yager, R. and Filev D. "Generation of Fuzzy Rules by Mountain Clustering," *Journal of Intelligent and Fuzzy System,* Vol. 2, No. 3, pp. 209-219, Sep. 1994.

INTELLIGENT ENGINEERING SYSTEMS THROUGH ARTIFICIAL NEURAL NETWORKS

VOLUME 5

FUZZY LOGIC AND EVOLUTIONARY PROGRAMMING

Editors:

Cihan H. Dagli
Metin Akay
C. L. Philip Chen
Benito R. Fernández
Joydeep Ghosh