APPLICATION OF THE FUZZY MIN-MAX NEURAL NETWORK CLASSIFIER TO PROBLEMS WITH CONTINUOUS AND DISCRETE ATTRIBUTES

A. Likas, K. Blekas and A. Stafylopatis
National Technical University of Athens
Department of Electrical and Computer Engineering
Computer Science Division
157 73 Zographou, Athens, Greece

Abstract. The fuzzy min-max classification network constitutes a promisimg pattern recognition approach that is based on hyberbox fuzzy sets and can be incrementally trained requiring only one pass through the training set. The definition and operation of the model considers only attributes assuming continuous values. Therefore, the application of the fuzzy min-max network to a problem with continous and discrete attributes, requires the modification of its definition and operation in order to deal with the discrete dimensions. Experimental results using the modified model on a difficult pattern recognition problem establishes the strengths and weaknesses of the proposed approach.

INTRODUCTION

Fuzzy min-max neural networks [2, 3] consitute one of the many models of computational intelligence that have been recently developed from research efforts aiming at synthesizing neural networks and fuzzy logic [1].

The fuzzy min-max classification neural network [2] is an on-line supervised learning classifier that is based on *hyperbox* fuzzy sets. A hyperbox constitutes a region in the pattern space that can be completely defined once the minimum and the maximum points along each dimension are given. Each hyperbox is associated with exactly one from the pattern classes and all patterns that are contained within a given hyperbox are considered to have full class membership. In the case where a pattern is not completely contained in any of the hyperboxes, a properly computed fuzzy membership function (taking values in $[0, 1]$) indicates the degree to which the pattern falls outside of each of the hyperboxes. During operation, the hyperbox with the maximum membership value is selected and the class associated with the winning hyperbox is considered as the desicion of the network. Learning in the fuzzy min-max classification network is an *expansion-contraction* process that consists of creating and adjusting hyperboxes (the minimum and maximum points along each dimension) and also associating a class label to each of them.

In this work, we study the performance of the fuzzy min-max classification neural network on a pattern recognition problem that involves both discrete and continuous attributes. In order to handle the discrete attributes, the definition of a hyperbox must be modified to incorporate crisp (not fuzzy) sets in the discrete dimensions. Moreover, a modification is needed of the way the membership values are computed, along with changes in the criterion under which the hyperboxes are expanded. Besides extending the definition and operation of the fuzzy min-max network, the purpose of this work is also to gain insight into the factors that affect operation and training and test its classification capabilities on a difficult problem.

In the following section a brief description of the operation and training of the fuzzy min-max classification network is provided, while in Section 3 the modified approach is presented. Section 4 provides experimental results from the application of the approach to a difficult classification problem. It also presents results from the comparison of the method with the backpropagation algorithm and summarizes the major advantages and drawbacks of the fuzzy min-max neural network when used as a pattern classifier.

LEARNING IN THE FUZZY MIN-MAX CLASSIFICATION NETWORK

Consider a classification problem with n continuous attributes that have been rescaled in the interval $[0, 1]$, hence the pattern space is I^n ($[0, 1]^n$). Moreover, consider that there exist p classes and K hyperboxes with corresponding minimum and maximum values v_{ji} and w_{ji} respectively ($j = 1, \ldots, K$, $i = 1, \ldots, n$). Let also c_k denote the class label associated with hyperbox B_k.

When the h^{th} input pattern $A_h = (a_{h1}, \ldots, a_{hn})$ is presented to the network, the corresponding membership function for hyperbox B_j is ([3])

$$b_j(A_h) = \frac{1}{n}\sum_{i=1}^{n}[1 - f(a_{hi} - w_{ji}, \gamma) - f(v_{ji} - a_{hi}, \gamma)] \tag{1}$$

where $f(x, \gamma) = x\gamma$, if $0 \le x\gamma \le 1$, $f(x, \gamma) = 1$ if $x\gamma > 1$ and $f(x, \gamma) = 0$ if $x\gamma < 0$. If the input pattern A_h falls inside the hyperbox B_j then $b_j(A_h) = 1$, otherwise the membership decreases and the parameter $\gamma \ge 1$ regulates the decrease rate. As already noted, the class of the hyperbox with the maximum membership is considered as the output of the network.

In a neural network formulation, each hyperbox B_j can be considered as a hidden unit of a feedforward neural network that receives the input pattern and computes the corresponding membership value. The values v_{ji} and w_{ji} can be considered as the weights from the input to the hidden layer. The output layer contains as many output nodes as the number of classes. The weights u_{jk} ($j = 1, \ldots, K$, $k = 1, \ldots, p$) from the hidden to the output layer express the class corresponding to each hyperbox: $u_{jk} = 1$ if B_j is a hyperbox for class c_k, otherwise it is zero.

During learning, each training pattern A_h is presented once to the network and the following process takes place: First we find the hyperbox B_j with the maximum membership value among those that correspond to the same class as pattern A_h and meet the expansion criterion:

$$n\theta \ge \sum_{i=1}^{n}(\max(w_{ji}, a_{hi}) - \min(v_{ji}, a_{hi})) \tag{2}$$

The parameter θ ($0 \le \theta \le 1$) is a user-defined value that imposes a bound on the size of a hyperbox and its value significantly affects the effectiveness of the training algorithm. In the case where an expandable hyperbox (of the same class) cannot be found, then a new hyperbox B_k is spawned and we set $w_{ki} = v_{ki} = a_{hi}$ for each i. Otherwise, the hyperbox B_j with the maximum membership value is *expanded* in order to incorporate the new pattern A_h, i.e., for each $i = 1, \ldots, n$:

$$v_{ji}^{new} = \min(v_{ji}^{old}, a_{hi}) \tag{3}$$

$$w_{ji}^{new} = \max(w_{ji}^{old}, a_{hi}) \tag{4}$$

Following the expansion of a hyperbox, an *overlap test* takes place to determine if any overlap exists between hyperboxes from different classes. In case such an overlap exists, it is eliminated by a *contraction process* during which the size of each of the overlapping hyperboxes is minimally adjusted. Details concering the overlap test and the contraction process can be found in [2].

From the above description it is clear that the effectiveness of the training algorithm mainly depends on two factors: the value of the parameter θ and the order with which the training patterns are presented to the network.

TREATING DISCRETE ATTRIBUTES

A basic assumption concerning the application of the fuzzy min-max classification network to a pattern recognition problem is that all attributes take continuous values. Hence, it is possible to define the pattern space (union of hyperboxes) corresponding to each class by providing the minimum and maximum attribute values along each dimension. In the case of pattern recognition problems that are based on both analog and discrete attributes, it is nessecary for the discrete features to be treated in a different way. This is mainly due to the fact that it is not possible to define a meaningful ordering of the values of discrete attributes. Thus, it is not possible to apply the minimum and maximum operations on which the original fuzzy min-max neural network is based.

Consider a pattern recognition problem with n attributes (both continous and discrete). Let $\mathcal{D}$ denote the set of the indices of the discrete attributes and $\mathcal{C}$ denote the set of indices of the continuous attributes. Let also $n_C = |\mathcal{C}|$ and $n_D = |\mathcal{D}|$ denote the number of continuous and discrete attributes respectively and D^i denote the domain of each discrete feature $i \in \mathcal{D}$. A pattern $A_h = (a_{h1}, \ldots, a_{hn})$ of this problem has the characteristic that $a_{hi} \in [0, 1]$ for $i \in \mathcal{C}$ and $a_{hi} \in D^i$ for $i \in \mathcal{D}$. In order to deal with problems characterized by such mixture of attributes, we consider that each hyperbox B_j is described by providing the minimum v_{ji} and maximum w_{ji} attribute values for the case of continuous features ($i \in \mathcal{C}$) and by explicitly providing a set of attribute values $D_{ji} \subseteq D^i$ for the case of discrete features $i \in \mathcal{D}$. Since it is not possible to define any distance measure between the possible values of discrete attributes, we cannot assign any fuzzy membership values to the elements of sets D_{ji}. Therefore, the sets D_{ji} are crisp sets, i.e., an element either belongs to a set or not. Taking this argument into account, equation (1) providing the membership degree of a pattern A_h to a hyperbox B_j, takes the following form:

$$b_j(A_h) = \frac{1}{n}\{\sum_{i\in\mathcal{C}}[1 - f(a_{hi} - w_{ji}, \gamma) - f(v_{ji} - a_{hi}, \gamma)] + \sum_{i\in\mathcal{D}} m_{D_{ji}}(a_{hi})\} \quad (5)$$

where $m_S(x)$ denotes the membership function corresponding to the crisp set S, which is equal to 1 if $x \in S$, otherwise it is equal to 0.

In a neural network implementation, the continuous input units are connected to the hidden units via the two kinds of weights v_{ji} and w_{ji} as mentioned in the previous section. In what concerns the discrete attributes, we can assign one input unit to each atribute value, that is set to 1 in case this value exists in the input pattern, while the other units corresponding to the same attribute are set equal to 0. If a specific value $d_{ik} \in D^i$ belongs to the set D_{ji}, then the weight between the corresponding input unit and the hidden unit j is set equal to 1, otherwise it is 0.

During training, when a pattern A_h is presented to the network the expansion criterion has to be modified in order to take into account both the discrete and the continuous dimensions. More specifically, we have considered two distinct expansion criteria: The first one concerns the continuous dimensions and remains the same as in the original network given by equation (2) with n being replaced by n_C which denotes the number of continuous attributes. The second expansion criterion concerns the discrete features and has the following form:

$$\lambda \leq \sum_{i\in\mathcal{D}} m_{D_{ji}}(a_{hi}) \quad (6)$$

where the parameter λ $(0 \leq \lambda \leq n_D)$ expresses the minimum number of discrete attributes in which the hyperbox B_j and the pattern A_h must agree in order for the hyperbox to be expanded to incorporate the pattern.

During the test for expansion process, we test whether there exist expandable hyperboxes (according to the two criteria) from the same class as A_h and we expand the hyperbox with the maximum membership. If no expandable hyperbox is found a new one B_k is spawned and we set $v_{ki} = w_{ki} = a_{hi}$ for $i \in \mathcal{C}$ and $D_{ki} = \{a_{hi}\}$ for $i \in \mathcal{D}$.

When a hyperbox is expanded, its parameters are adjusted as follows: If $i \in \mathcal{C}$

$$v_{ji}^{new} = \min(v_{ji}^{old}, a_{hi}) \quad (7)$$

$$w_{ji}^{new} = \max(w_{ji}^{old}, a_{hi}) \quad (8)$$

If $i \in \mathcal{D}$

$$D_{ji}^{new} = D_{ji}^{old} \cup \{a_{hi}\} \tag{9}$$

During overlap test and contraction the discrete dimensions are not considered and ovelap is eliminated by adjusting ony the continuous dimensions of the hyperboxes following the minimum disturbance principle as in the original network. Although it is possible to separate two hyperboxes B_j and B_k by removing common elements from some of the sets D_{ji} and D_{ki}, we have not followed this approach. The main reason is that the disturbance in the already allocated patterns would be more significant, since these sets do not contain many elements in general.

EXPERIMENTS AND CONCLUSIONS

We have studied the modified fuzzy min-max neural network classifier on a difficult classification problem concerning the assignment of credit to consumer applications. The data set (obtained from the UCI repository [5]) contains 690 examples and was originally studied by Quinlan [4] using decision trees. Each example in the data set concerns an application for credit card facilities described by 9 discrete and 6 continuous attributes, with two decision classes (either accept of reject the application). Some of the discrete attributes have large collections of possible values (one of them has 14) and there exist examples in which some attribute values are missing. As noted in [4] these data are both scanty and noisy making accurate prediction on unseen cases a difficult task.

Two series of experiments were performed. In the first series, the data set was divided into a training set of 460 examples (containing equal number of positive and negative cases) that were used to adjust the network hyperboxes, while the remaining 230 examples were used as a test set to estimate the performance of the resulting classifier. Each experiment in a series consisted of training the network (in a single pass) for certain values of θ and λ and then computing the percentage of correct classifications over the test set. Moreover, the order of presentation of the training patterns to the network was held fixed in all experiments. Best results were found for $\theta = 0.237$ and $\lambda = 8$. For these parameter values the resulting network contained 136 hyperboxes and the success rate was 87%. It must be noted that the success rate was very sensitive both on the choice of the parameter θ and on the order with which the training examples are presented. This of course constitutes a weakness of the fuzzy min-max classifier, but on the other hand, each training experiment is very fast and the process of adjusting θ can be performed in reasonable time. We have also tested the classification performance in case the training data are presented to the network more than once and we have found that only marginal performance improvement is obtained.

We have also used the same data set to train a multilayer perceptron using the backpropagation algorithm (the on-line version). A network with one hidden layer was considered. Several experiments were conducted for different values of the number of hidden units. The best classification rate we were able to obtain was 83% for a network of 10 hidden units and with learning rate 0.1. It must be noted that the required training time was excessively long compared to the one-shot training of the fuzzy min-max network.

During experiments, we have observed that some of the examples were 'bad', in the sense that they were very difficult predict, and, in addition, when used as part of the training set, the resulting network exhibited poorer classification performance, than in the case in which these examples were not used for training. For this reason, a second series of experiments were conducted on a subset of the data set (400 examples) that resulted from the removal of the bad examples. We considered a training set and a test set of size 200, each of them containing 100 positive and 100 negative examples. Best performance was obtained for $\theta = 0.115$ and $\lambda = 8$ (112 hyperboxes) with classification rate 97.5%. Moreover, the performance was very robust with respect to the value of θ with the classification rate being more than 90% for all tested values. The best classification rate we have obtained for this data set using the backpropagation algorithm was 89.5%.

As the experiments indicate, the fuzzy min-max classification neural network constitutes a promising method for pattern recognition problems that has the advantage of fast one-shot training with its only drawback coming from its sensitivity in the parameter values used in the test for expansion criteria. Therefore, further research should be focused on developing algorithms for automatically adjusting these parameters during training.

REFERENCES

[1] IEEE Trans. on Neural Networks, Special Issue on Fuzzy Logic and Neural networks, vol. 3, No. 5, September 1992.

[2] P. K. Simpson, "Fuzzy Min-Max Neural Networks-Part 1: Classification," IEEE Trans. on Neural Networks, vol. 3, No. 5, pp. 776-786, September 1992.

[3] P. K. Simpson, "Fuzzy Min-Max Neural Networks-Part 2: Clustering," IEEE Trans. on Fuzzy Systems, vol. 1, No. 1, pp. 32-45, February 1993.

[4] J. R. Quinlan, "Simplifying Decision Trees," Int. J. Man-Machine Studies, vol. 27, pp. 221-234, 1987.

[5] P. M. Murphy and D. W. Aha, "UCI Repository of Machine Learning Databases," Irvine, CA: University of California, Department of Computer Science, 1992.