A Fuzzy Relational Neural Network for Pattern Classification

Israel Suaste-Rivas[1], Orion F. Reyes-Galaviz[2], Alejandro Diaz-Mendez[1], and Carlos A. Reyes-Garcia[1]

[1] Instituto Nacional de Astrofisica Optica y Electronica (INAOE), Mexico
isuaste@susu.inaoep.mx, ajdiaz@inaoep.mx, kargaxxi@inaoep.mx
[2] Instituto Tecnologico de Apizaco, Mexico
orionfrg@yahoo.com

Abstract. In this paper we describe the implementation of a fuzzy relational neural network model. In the model, the input features are represented by fuzzy membership, the weights are described in terms of fuzzy relations. The output values are obtained with the max-min composition, and are given in terms of fuzzy class membership values. The learning algorithm is a modified version of back-propagation. The system is tested on an infant cry classification problem, in which the objective is to identify pathologies in recently born babies.

1 Introduction

In this paper we describe the implementation of a general pattern classifier based on a neural network architecture which uses fuzzy sets for both input/output data and the structure of the classifier itself. The general system architecture was inspired by the work presented by Pal in [1, 2]. The idea of using a relational neural network was taken from the general classifier presented by Pedrycz in [3]. The implemented system has been tested on an infant cry recognition problem. The pathological diseases in infants are commonly detected several months, and often times, years, after the infant is born. If any of these diseases would have been detected earlier, they could have been attended and maybe avoided by the opportune application of treatments and therapies. It has been found that the infant's cry has much information on its sound wave. Based on the information contained inside the cry's wave, it can be determined the infant's physical state; and even detect physical pathologies in very early phases [4].

2 The Infant Cry Recognition Process

The Automatic Infant Cry Recognition (AICR) process is similar to Automatic Speech Recognition (ASR). In AICR the goal is to take the crying wave as the input pattern, and at the end to obtain the class of cry the vector corresponds to. Generally, the whole process can be divided into two phases. The first phase is known as *signal processing* or *acoustic feature extraction* and the second is known as *pattern classification*.

A. Sanfeliu et al. (Eds.): CIARP 2004, LNCS 3287, pp. 358–365, 2004.

2.1 Signal Processing

The analysis of the raw cry waveform provides the information needed for its recognition. Acoustic feature extraction is a transformation of measured data into pattern data. The features may be spectral coefficients, linear prediction coefficients (LPC), or Mel frequency cepstral coefficients (MFCC) among others [5]. The set of values for n features may be represented by a vector in an n-dimensional space. Each vector represents a pattern.

2.2 Pattern Classification

During the second phase of the infant cry recognition process, the goal is usually to determine the class or category of each pattern. In this work we classify the patterns by means of a hybrid connectionist model.

3 The Fuzzy Neural Network Model

The system proposed in this work is based upon fuzzy set operations in both the structure of the neural network and in the learning process. Following Pal's idea of a general recognizer [2], the model is divided in two main parts, one for learning and another for processing, as shown in figure 1.

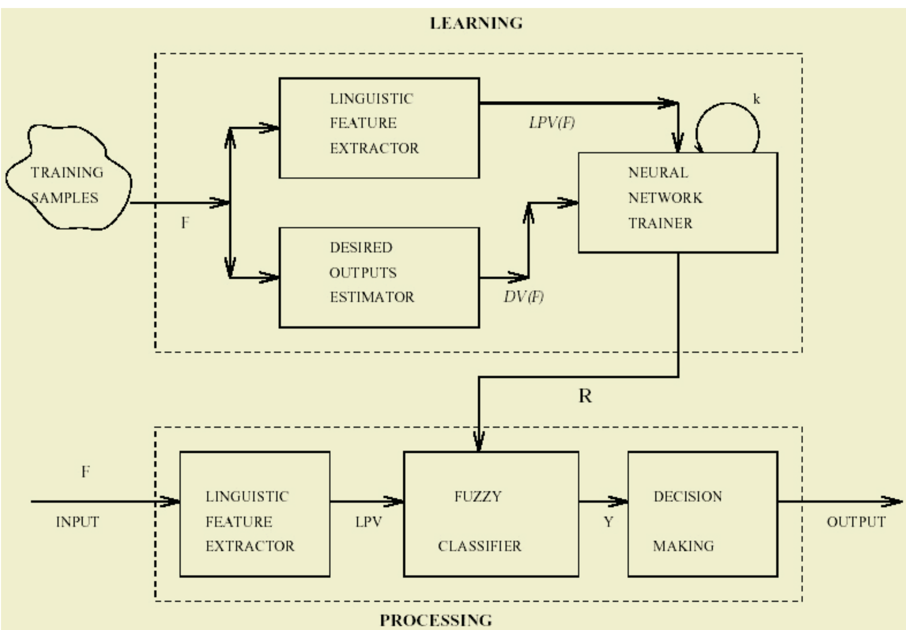

Fig. 1. General Automatic Infant Cry Recognition System

3.1 Fuzzy Learning

The fuzzy learning section is composed by three modules, namely the Linguistic Feature Extractor (LFE), the Desired Output Estimator (DOE), and the Neural Network Trainer (NNT). The Linguistic Feature Extractor takes training samples in the form of n-dimensional vectors containing n features, and converts them to a Nn-dimensional form vectors, where N is the number of linguistic properties. In case the linguistic properties are *low, medium,* and *high*, the resulting $3n$-dimensional vector is called Linguistic Properties Vector (LPV). In this way an input pattern $\mathbf{F}_i = [F_{i1}, F_{i2}, ..., F_{in}]$ containing n features, may be represented as [2]:

$$\begin{aligned}\mathbf{F}_i = [&\mu_{low(F_{i1})}(\mathbf{F}_i), \mu_{med(F_{i1})}(\mathbf{F}_i),\\ &\mu_{high(Fi1)}(\mathbf{F}_i), \ldots, \mu_{high(F_{in})}(\mathbf{F}_i)]\end{aligned} \tag{1}$$

The DOE takes each vector from the training samples and calculates its membership to class k, in an l-class problem domain. The vector containing the class membership values is called the Desired Vector (DV). Both LPV and DV vectors are used by the neural Network Trainer (NNT), which takes them as the bases for training the network.

The neural network consists of only two layers. The input layer is formed by a set of Nn neurons, with each of them corresponding to one of the linguistic properties assigned to the n input features. In the output layer there are l neurons, where each node corresponds to one of the l classes; in this implementation, each class represents one type of crying. There is a link from every node in the input layer to every node in the output layer. All the connections are described by means of fuzzy relations $R : X \times Y \longrightarrow [0, 1]$ between the input and output nodes. The error is represented by the distance between the actual ouput and the target or desired output. During each learning step, once the error has been computed, the trainer adjusts the relationship values or weights of the corresponding connections, either until a minimum error is obtained or a given number of iterations is completed. The output of the NNT, after the learning process, is a fuzzy relational matrix (R in Figure 1) containing the knowledge needed to further map the unknown input vectors to their corresponding class during the classification process.

3.2 Fuzzy Processing

The fuzzy processing section is formed by three different modules, namely the Linguistic Feature Extractor (LFE), the Fuzzy Classifier (FC), and the Decision Making Module (DMM). The LFE works in exactly the same way as the one in the learning phase. It is used to calculate the corresponding membership value of each input feature in the classifying vector to each of the linguistic properties. The output of this module is an LPV vector. The LPV vector, along with the fuzzy relational matrix R, are used by the Fuzzy Classifier, which obtains the actual outputs from the neural network. The classifier applies the max-min composition to calculate the output. The ouput of this module is an output vector containing the membership values of the input vector to each of the classes. Finally, the Decision Making module takes the values coming from the classifier, and after applying some decision criteria assigns the corresponding class to the testing vector. The assigned class, in this implementation, represents one kind of infant cry.

3.3 Membership Functions

A fuzzy set is defined by a function that maps objects in a domain to their membership value in the set. Such a function is called the *membership function.* In many cases it is recomended to use standard functions whose parameters may be adjusted to fit a specified membership function in an approximate fashion. In the reported experiment the triangular membership function was used. There is some evidence that shows that the use of more linguistic properties to describe a pattern point makes a model more accurate [6]. One possibility is the use of seven linguistic properties: *very low, low, more or less low, medium, more or less high, high, very high.*

3.4 Desired Membership Values

Before defining the output membership function, we define the equation to calculate the weighted distance of the training pattern $\mathbf{F}_j$ to the kth class in an l-class problem domain as in [1]

$$z_{ik} = \sqrt{\sum_{j=1}^{n} \left[\frac{F_{ij} - o_{kj}}{v_{kj}}\right]^2}, : for\ k = 1, \ldots, l \tag{2}$$

where F_{ij} is the jth feature of the ith pattern vector, o_{kj} denotes the mean, and v_{kj} denotes the standard deviation of the jth feature for the kth class. The membership value of the ith pattern to class k is defined as follows

$$\mu_k(\mathbf{F}_i) = \frac{1}{1 + (\frac{z_{ik}}{f_d})^{f_e}}, : \mu_k(\mathbf{F}_i) \in [0, 1] \tag{3}$$

where f_e is the exponential fuzzy generator, and f_d is the denominational fuzzy generator controlling the amount of fuzzines in this class-membership set. In this case, the higher the distance of the pattern from a class, the lower its membership to that class. Since the training data have fuzzy class boundaries, a pattern point may belong to one or more classes in the input feature space.

3.5 The Neural Network Trainer

The neural network model discussed here is based on the fuzzy neural structure proposed by Pedrycz in [3].

The Relational Neural Network. Let $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ be a finite set of input nodes and let $\mathbf{Y} = \{y_1, y_2, \ldots, y_l\}$ represent the set of output nodes in an l-class problem domain. When the max-min composition operator denoted $X \circ R$ is applied to a fuzzy set X and a fuzzy relation R, the output is a new fuzzy set Y, we have

$$Y = X \circ R \tag{4}$$

$$Y(y_j) = max_{x_i}(min(X(x_i), R(x_i, y_j)))$$

where X is a fuzzy set, Y is the resulting fuzzy set and R is a fuzzy relation $R : X \times Y \longrightarrow [0, 1]$ describing all relationships between input and output nodes.

We will take the whole neural network represented by expression (4) as a collection of l separate n-input single-output cells.

Learning in a Fuzzy Neural Network. If the actual response from the network does not matches the target pattern, the network is corrected by modifying the link weights to reduce the difference between the observed and target patterns. For the relational neural network Pedrycz [3] defines a performance index called equality index, which is

$$T(y) \equiv Y(y) = \begin{cases} 1 + T(y) - Y(y), \text{if } Y(y) > T(y) \\ 1 + Y(y) - T(y), \text{if } Y(y) < T(y) \\ 1, \qquad\qquad\qquad\quad \text{if } Y(y) = T(y) \end{cases} \tag{5}$$

where $T(y)$ is the target output at node y, and $Y(y)$ is the actual output at the same node. $\overline{T}$ is the complement of T defined by $\overline{T}(y) = 1 - T(y)$. In a problem with n input patterns, there are n input-output pairs (x_{ij}, t_i) where t_i is the target value when the input is $\mathbf{X}_{ij}$.

Parameters Updating. In [3] Pedricz discusses the learning scheme for the structure of a neural network with n-inputs and single output, and proposes to complete the process of learning separately for each output node. The updating procedure is made independent of the size of the training set. The learning algorithm is a version of the gradient-descent-based backpropagation algorithm.

Lets consider an n-input-L-output neural network having the following form

$$y_i = f(\mathbf{x}_i; \mathbf{a}, \upsilon) = (\bigvee_{j=1}^{n} (a_j \wedge x_{ij}))$$

where $\mathbf{a} = [a_1,a_2, \ldots, a_L]$ is a vector containing all the weights or relations, $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{in}]$ is the vector with values observed in the input nodes. The parameters a and υ are updated iteratively by taking increments $\triangle a_m$ and $\triangle \upsilon_m$ resulting from deviations between all pairs y_i and t_i as follows

$$\begin{aligned} a(k+1) = {} & a(k) + \\ & \Psi_1(k) \left[\tfrac{\triangle a(k+1)}{Nn} + \eta \tfrac{\triangle a(k)}{Nn} \right] \end{aligned} \tag{6}$$

where k is the iteration or learning step. Ψ_1 and Ψ_2 are non-increasing functions of k controlling the decreasing influence of increments $\triangle a_m$ and $\triangle \upsilon_m$. Ψ is the learning momentum specifying the level of modification of the learning parameters with regard to their values in the previous learning step k. A way of determining the increments $\triangle a_m$ and $\triangle \upsilon_m$ is with regard to the mth coordinates of a, $m = 1,2,...,L$. The computation of the overall performance index, and the derivatives to calculate the increments for each cordinate of a, and υ are explained in detail in [7].

Once the training has been terminated, the output of the trainer is the updated relational matrix, which will contain the knowledge needed to map unknown patterns to their corresponding classes.

4 Implementation and Results

In the first place, the infant cries are collected by recordings obtained directly by medical doctors. Later, each signal wave is divided in segments of 1 second; each segment constitutes a sample. For the present experiments we have a corpus of 157 samples of normal infant cry, 879 of hypo acoustics, and 340 with asphyxia. At the following step the samples are processed extracting their MFCCs, by the use of the freeware program Praat 4.0 [8]. The acoustic characteristics are extracted as follows: every one second sample is windowed by 50-millisecond frames from which we extract 16 coefficients, generating vectors with 304 coefficients by sample. This vectors are further reduced, to a desired size, by the applicacion of PCA. The neural network and the training algorithm are implemented with Matlab. In order to make the training and recognition test, we select 157 samples randomly on each class. The number of normal cry samples available determines this number. From them, 107 samples of each class are randomly selected for training. The training is made up to 20 epochs. After the network is trained, we test it with the 50 samples of each class set apart from the original 157 samples. The recognition accuracy percentage, from each experiment, is presented in a confusion matrix. The best results at present have been obtained with the following parameters $\eta = 0.2$ and $k = 40$. The initial values for the relational matrix were set as 0.8. And the number of input features per vector, after the application of PCA, equal to 3. The feature space was divided in 7 linguistic terms, which makes the dimensionality of the input vector equal 21.

4.1 Preliminary Results

The results of the model when using the above mentioned set of values, triangular membership functions, and considering only the highest membership value of each input pattern in the output vector, are as given in the confusion matrix in Table 1

4.2 Performance Comparison with Other Models

Taco Ekkel [9] tried to classify sound of newborn cry in categories called normal and abnormal (hypoxia), and reports a result of correct classification of around

Table 1. Confusion Matrix.

Class	Samples	Normal	Deaf	Asphyxia	Accuracy
Normal	50	47	2	1	
Deaf	50	5	45		
Asphyxia	50	10		40	
Total	150				88 %

85% based on a neural network of radial base. In [10] Reyes and Orozco classify cry samples only from deaf and normal infants, obtaining recognition results that go from 79.05% up to 97.43%.

5 Further Work

One of the major proposed change is the automatic optimization of the required parameters, by the use of genetic algorithms. Another important addition is the use of the square relational product as in [11] besides the circlet product during the recognition phase to generate another criterion to help in the recognition decision.

6 Conclusions

Given the simplicity of the model implementation, the results obtained to date are very encouraging. Based on the observed results, we are convinced that by implementing the proposed changes the improved fuzzy relational neural network model can provide an effective, reliable and compact infant cry recognizer.

Acknowledgments

This work is part of a project that is being financed by CONACYT-Mexico (37914-A). We want to thank Dr. Edgar M. Garcia-Tamayo and Dr. Emilio Arch-Tirado for their invaluable collaboration when helping us to collect the crying samples, as well as for their wise advice.

References

1. Pal, S.K., “Multilayer Perceptron, Fuzzy Sets, and Classification,” in *IEEE Trans. on Neural Networks*, vol 3, No 5, Sep 1992, pp 683-697.
2. Pal, S.K. and Mandal, D.P., “Linguistic Recognition Systems Based on Approximated Reasoning,” in *Information Science*, vol 61, No 2, 1992, pp 135-161.
3. Pedrycz, W., “Neurocomputations in Relational Systems,” in *IEEE Trans. on Pattern Analysis and Mach. Intelligence*, vol 13, No 3, Mar 91, pp 289-296.
4. O. Wasz-Hockert, J. Lind, V. Vuorenkoski, T. Partanen y E. Valanne, El Llanto en el Lactante y su Significación Diagnóstica, Cientifico-Medica, Barcelona, 1970.

5. Schafer, R.W., and Rabiner, L.R., “Digital Representations of Speech Signals,” in *Readings in Speech Recognition*, Morgan Kauffmann Publishers Inc., San Mateo, Calif, 1990, pp 49–64.
6. Park, D., Cae, Z., and Kandel, A., “Investigations on the Applicability of Fuzzy Inference,” in *Fuzzy Sets and Systems*, vol 49, 1992, pp 151-169.
7. Reyes, C.A., “On the Design of a Fuzzy Relational Neural Network for Automatic Speech Recognition”, Doctoral Dissertation, The Florida State University , Tallahassee, Fl., Apr 94.
8. Boersma, P., Weenink, D. Praat v. 4.0.8. A system for doing phonetics by computer. In-stitute of Phonetic Sciences of the University of Amsterdam. February, 2002
9. Ekkel, T, *Neural Network-Based Classification of Cries from Infants Suffering from Hypoxia-Related CNS Damage*, Master Thesis. University of Twente, The Netherlands, 2002.
10. Orozco Garcia, J., Reyes Garcia, C.A. (2003), “Mel-Frequency Cepstrum Coeficients Extraction from Infant Cry for Classification of Normal and Pathological Cry with Feed-forward Neural Networks”, in *proceedings of ESANN 2003*, Bruges, Belgium.
11. Bandler, W. and Kohout, L.J., “Fuzzy Relational Products as a tool for analysis and artificial systems,” in P.P. Wang, and S.K. Chang, editors, *Fuzzy Sets: Theory and Applications to Policy Analysis and Information Systems*, Plenum Press, New York and London, 1980, pp 341-367.