

# Genetic Fuzzy Relational Neural Network for Infant Cry Classification

Alejandro Rosales-Pérez, Carlos A. Reyes-García, and Pilar Gómez-Gil

National Institute of Astrophysics, Optics and Electronics (INAOE)  
Computer Science Department  
Luis E. Erro No.1 Tonantzintla,  
Puebla, México

{arosales,kargaxxi,pgomez}@ccc.inaoep.mx

**Abstract.** In this paper we describe a genetic fuzzy relational neural network (FRNN) designed for classification tasks. The genetic part of the proposed system determines the best configuration for the fuzzy relational neural network. Besides optimizing the parameters for the FRNN, the fuzzy membership functions are adjusted to fit the problem. The system is tested in several infant cry database reaching results up to 97.55%. The design and implementation process as well as some experiments along with their results are shown.

**Keywords:** Fuzzy relational neural network, genetic algorithm, infant cry classification.

## 1 Introduction

In this work we present a genetic fuzzy relational neural network. The relational neural network was originally proposed by Pedrycz [6] and it was extended by Reyes, who designed a Fuzzy Relational Neuronal Network (FRNN) for speech recognition [7]. Unlike most of the fuzzy neural network models that only use the elements of fuzzy sets theory for the learning process or in their structure, the FRNN uses fuzzy sets for both input/output data and the structure and functioning of the classifier itself.

The FRNN has been used for pattern recognition. In [8,9,10] the FRNN was implemented for infant cry classification. Also, in [2] a proposal is presented to optimize the parameters for the FRNN using a genetic algorithm. As a complement, in this work we propose to use a genetic algorithm for optimizing the parameters for the FRNN, as well as the parameters of the fuzzy membership functions. The proposed system has been tested on several infant cry databases to classify asphyxia, deaf, normal, hungry and pain cries. All experiments are binary classification. Although the problem of infant cry classification is indeed a multiclass problem, and our team has treated it in that way in several previous works([8,9,10]), but for the present case we present a binary classification because our purpose is to compare our results with a particular similar work

wich precissely had that binary approach. Results are compared with the work of Barajas and Reyes [2], that used the same databases.

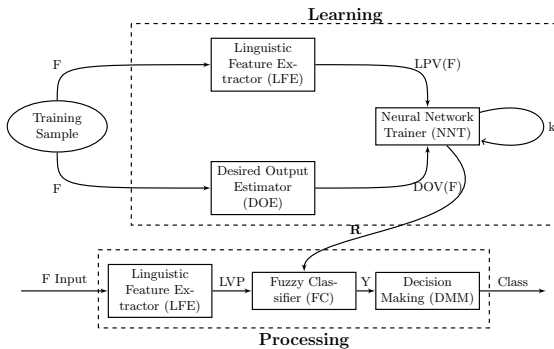
The proposed system can be used for automatic infant cry classification. This classifier may be a powerful tool for health professionals and parents during the detection of pathologies in infants.

The rest of the paper is organized as follows. In Section 2 we present the Fuzzy Relational Neural Network Model. In Section 3 we present the designed genetic algorithm used to optimize the parameters of the membership functions as well as the FRNN. Then, in Section 4 we describe the experiment and the results. Finally, in Section 5 the conclusions and future work are presented.

## 2 Fuzzy Relational Neural Network

The fuzzy relational neural network consists of two layers, the input layer and the output layer. The input layer is formed by a set of  $N \times n$  neurons, with each of them corresponding to one of the  $N$  linguistic properties assigned to each of the  $n$  input features. In the output layer there are  $l$  neurons, where each node corresponds to one of the  $l$  classes. There is a link from every node in the input layer to every node in the output layer. All the connections, instead of regular weights, are described by fuzzy relations  $\mathbf{R} : \mathbf{X} \times \mathbf{Y} \rightarrow [0, 1]$  between the input and output nodes.

The Fuzzy Relational Neural Network (FRNN) operation is divided in two main stages; the first one is for learning and the second one is for processing [7]. Fig. 1 shows the general architecture and stages of this FRNN. Next each phase as well as their modules are described.



**Fig. 1.** General architecture and stages of a FRNN for the Automatic Infant Cry Recognition

### 2.1 Learning Phase

The learning section is divided in three modules: the Linguistic Feature Extractor (LFE), the Desired Output Estimator (DOE) and the Neural Network trainer (NNT).

The first module, LFE, takes the training samples and each input feature (F) is transformed in membership values to each of the linguistic properties assigned. Thus a vector containing  $n$  features can be transformed in a  $3n$ -dimensional (for describing low, medium, high),  $5n$ -dimensional (very low, low, medium, high, very high), or  $7n$ -dimensional (very low, low, more or less low, medium, more or less high, high, very high) vectors. The new resulting vector is called Linguistic Properties Vector (LPV).

To calculate the membership values we use four different membership functions: Gaussian, Trapezoidal, Triangular and Bell.

The FRNN is a supervised learning system; in that sense the second module, the DOE, is in charge to compute the membership values of each sample to each class of the problem, which is latter used to calculate the error of the network after each learning iterations. To obtain the desired membership values it is necessary to calculate the weighted distance of the training pattern  $F_j$  to the  $k^{th}$  class in an  $l$ -class problem domain as in equation:

$$z_{ik} = \sqrt{\sum_{j=1}^n \left[ \frac{F_{ij} - \mu_{kj}}{\sigma_{kj}} \right]^2} \quad | \quad \text{for } k = 1 \dots l \tag{1}$$

where  $F_{ij}$  is the  $j^{th}$  feature of the  $i^{th}$  pattern vector,  $\mu_{kj}$  and  $\sigma_{kj}$  denotes, respectively, the mean and the standard deviation of the  $j^{th}$  feature for the  $k^{th}$  class. The membership value of the  $i^{th}$  pattern to class  $k$  is defined as follows:

$$\mu_l(\mathbf{F}_i) = \frac{1}{1 + \left( \frac{z_{ik}}{fd} \right)^{fe}} \tag{2}$$

where  $fe$  is the exponential fuzzy generator, and  $fd$  is the denominational fuzzy generator controlling the amount of fuzziness 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.

Finally, the third module, NNT, takes both the LPV and DOV vectors as the basis for training the network. The LPV is clamped to the input layer and the DOV is clamped to the output layer during training. The outputs of the network are computed to obtain the error at the output layer. The error is represented by the distance between the actual output and the target output. To minimize this error is the objective of the training process. 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 is a relational matrix<sup>1</sup> containing the knowledge needed to further map the unknown input vectors to their corresponding class during the classification process. The relational neural network, the learning process in a FRNN and parameters updating are explained in detail in [7].

---

<sup>1</sup> Matrix that contains the fuzzy relations corresponding to each feature.

## 2.2 Processing Phase

Once the learning phase is completed, the information collected is used to classify unknown patterns; the processing phase is in charge of that task. The modules that form the processing phase are the Linguistic Feature Extractor (LFE), the Fuzzy Classifier (FC), and the Decision Making Module (DMM) (see Fig. 1).

The LFE is similar to the one described in the learning phase. The only difference is that, in the classification phase, the LFE does not calculate new parameters to be applied by the membership function. The LFE in this phase takes the last values of the membership functions, corresponding to each feature in the sample patterns, collected by the LFE in the learning phase, and uses them to calculate the linguistic properties vector (LPV) corresponding to the testing patterns.

The second module, the FC, performs the classification. The fuzzy classification is done by using fuzzy relational products [1]; in our work it can be done in five different ways: by mean of the max-min composition, square, subtriangle and supertriangle relational products, as well as max-geometrical mean. The relational products used are defined, respectively, in equations 3-7.

$$\mathbf{Y}(y_j) = \max_j (\max (\min (\mathbf{X}(x_{ij}), \mathbf{R}(x_{ij}, y_j))), b) \quad (3)$$

$$\mathbf{Y}(y_j) = \max_j (\min (\mathbf{X}(x_{ij}) \leftrightarrow \mathbf{R}(x_{ij}, y_j)), b) \quad (4)$$

$$\mathbf{Y}(y_j) = \max_j (\min (\mathbf{X}(x_{ij}) \rightarrow \mathbf{R}(x_{ij}, y_j)), b) \quad (5)$$

$$\mathbf{Y}(y_j) = \max_j (\min (\mathbf{X}(x_{ij}) \leftarrow \mathbf{R}(x_{ij}, y_j)), b) \quad (6)$$

$$\mathbf{Y}(y_j) = \max_j (\max (\mathbf{X}(x_{ij}), \mathbf{R}(x_{ij}, y_j))^{1/2}, b) \quad (7)$$

where  $x_{ij}$  is the  $j^{\text{th}}$  component of the  $i^{\text{th}}$  input pattern,  $\mathbf{R}(x_{ij}, y_j)$  is the  $(i, j)^{\text{th}}$  entry in the relational matrix  $\mathbf{R}$  and  $b$  is a threshold.

The FC module uses the LPV and the relational matrix –obtained during the learning phase– for classifying new patterns to their corresponding class. The last module, DMM, takes the highest membership value from the class vector and the input sample is assigned to that class.

## 3 Genetic Algorithms for Optimizing Parameters

Genetic algorithms are part of the fastly emerging evolutionary computation area. Evolutionary computation is inspired in the evolutionary theory and it tries to solve problems using computational models of evolutionary processes, such as selection, reproduction, mutation, survival of the fittest, etc. Generally, a genetic algorithm has five basic components: an encoding that represents the potential solutions for the problem in the form of chromosomes or individuals, a form to create potential initial solutions, a fitness function to measure how close a chromosome is to the desired solution, operators for selection and operators for reproduction [4]. Next we explain how each of these components was designed in this system, in order to optimize the fuzzy membership functions and the parameter of the FRNN.

### 3.1 Representation of the Chromosome

In this work we use a binary representation of the chromosome because, it fits well to the needs of our problem and because according to Holland [5], this representation allows to have more schemes than decimal representation.

The problem is to optimize the parameter for the FRNN. These parameters are: number of linguistic properties (NPL), type of membership function (TMF), number of training epochs (NTE), learning rate (LR), initial weights for the relational matrix (WRM), the initial threshold (IT), the type relational product to be applied for getting the output classification (TRP), as well as the parameters for the selected membership functions, composing altogether a large search space. The chromosome has  $18 + 16 * numLP * n$  bits, where  $numLP$  represents the number of linguistic properties and  $n$  the number of features in the data set. In that sense, the chromosome size is defined at runtime.

The coding scheme adopted is described in detail next. The type of membership function is represented by two bits: a value of  $[0, 0]$  means a triangular function;  $[0, 1]$  represents a trapezoidal function, a value of  $[1, 0]$  means a gaussian function and  $[1, 1]$  represents a general bell function. The number of linguistic properties is represented by two bits:  $[0, 0]$  represents three linguistic properties,  $[0, 1]$  five linguistic properties and  $[1, 0]$  seven linguistic properties. The number of training epochs, learning rate, initial threshold, initial relational weights and the type of output are represented by three bits and their codification is shown in Table 1. The different values for each of these parameters were experimentally established.

**Table 1.** Bits codification for the genetic algorithm

Parameter	Binary codification							
	[0, 0, 0]	[0, 0, 1]	[0, 1, 0]	[0, 1, 1]	[1, 0, 0]	[1, 0, 1]	[1, 1, 0]	[1, 1, 1]
Epochs train	2	5	10	15	20	25	30	50
Learning rate	0.10	0.15	0.20	0.25	0.30	0.33	0.35	0.40
Bias	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.50
Weight	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80
Output	max-min	square	sub-triangle	super-triangle	max-mean	-	-	-

Finally, each membership function can be represented by four parameters  $(a, b, c, d)$ ; every parameter is represented by four bits and its codification is shown in Table 2. For that reason, the updating parameters for the membership functions are represented by  $16 * numPL * n$  bits, it means 16 bits for each membership function belonging to each feature. This part of the algorithm is the main extension to the work of Barajas and Reyes[2]. In Table 2 each value of the gene is associated with a value between 0.50 and 1.50, which represents a scaling value (SV) for to increase or decrease appropriately the parameters of fuzzy membership functions.

**Table 2.** Binary codification for optimizing parameters of the membership function

Gene	SV	Gene	SV	Gene	SV	Gene	SV
[0, 0, 0, 0]	0.50	[0, 1, 0, 0]	0.75	[1, 0, 0, 0]	1.20	[1, 1, 0, 0]	1.40
[0, 0, 0, 1]	0.55	[0, 1, 0, 1]	0.80	[1, 0, 0, 1]	1.25	[1, 1, 0, 1]	1.45
[0, 0, 1, 0]	0.60	[0, 1, 1, 0]	0.90	[1, 0, 1, 0]	1.30	[1, 1, 1, 0]	1.50
[0, 0, 1, 1]	0.70	[0, 1, 1, 1]	1.10	[1, 0, 1, 1]	1.35	[1, 1, 1, 1]	1.00

Initially, the membership functions are uniformly distributed and their parameters take a value into the feature's domain; the dot product with the value obtained by the bits is applied to that value, in that way the parameters for the membership functions are normalized for every specific problem.

### 3.2 Genetic Operations

The selection operation is done by tournament; more information on this operator is detailed in [4]. We use one point crossover for creating the new population. A one-point crossover operator randomly selects a crossover point in the two previously selected parents, and the bitstrings after that point are swapped between the two of them. A crossover probability of 0.8 was used in all experiments.

In addition, we use a mutation operator with the aim to introduce new genetic material into an existing individual. Similarly to crossover, mutation requires probability to be applied; usually this probability is low and in this case the mutation probability is 0.02.

### 3.3 Fitness Function

Fitness function is an important part for a genetic algorithm because it is in charge of evaluating the potential solutions. As a fitness function we use the balanced error rate; in that way we avoid to select parameters that have a good performance classifying only one class. The expression for the fitness function is:

$$BER = \frac{e_{(+)} + e_{(-)}}{2} \quad (8)$$

where  $BER$  is the balanced error rate,  $e_{(+)}$  and  $e_{(-)}$  are the missclassifications rates for the positive and negative classes, respectively.

## 4 Experiments and Results

In this work we experimented with the classification of infant cry databases built by the computer science department of the INAOE. The infant cries were collected by recordings done directly by medical doctors and then, each signal wave was divided in segments of 1 second; each segment represents a sample. For the experiments showed here, we used the acoustic obtained by [2], which

**Table 3.** Results of classifications using the proposed method

Data base	Accuracy	TPR	TNR	ROC
Asphyxia vs Normal	88.67%	90.00%	87.78%	92.85%
Deaf vs Normal	97.55%	98.75%	95.47%	99.75%
Hungry vs Pain	96.03%	95.59%	96.67%	98.35%

**Table 4.** Chromosome interpretation

Data base	Chromosome	NPL	TMF	NTE	LR	IT	WRM	TRP
Asphyxia vs Normal	[1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1]	3	Gaussian	30	0.20	0.50	0.30	Max-mean
Deaf vs Normal	[1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1]	5	Gaussian	50	0.20	0.50	0.50	Supertriangle
Hungry vs Pain	[1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1]	5	Gaussian	5	0.35	0.25	0.70	Max-mean

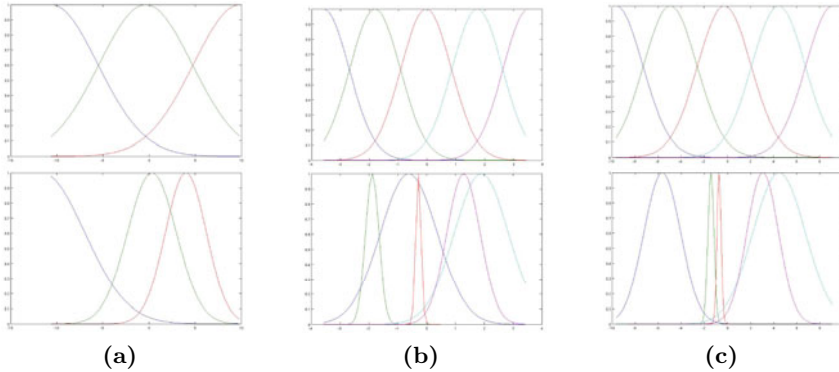
include Mel Frequency Cepstral Coefficients (MFCC). For the feature extraction process every sample of 1 second is divided in frames of 50 milliseconds and from each of which 16 coefficients were extracted, getting vectors with 304 coefficients by sample. The acoustic feature extraction process was done using Praat [3]. Then, we applied Principal Component Analysis to reduce the dimensionality of the vectors, for whose purpose we selected 16 principal components. The corpus has 157 samples of normal infant cry, 340 of asphyxia infant cry, 879 of deaf, 192 samples of pain cry and 350 of hunger cry. Pain and hunger samples were taken from normal babies.

The complete classification system is implemented in Matlab. For our experiments the initial population is formed by 25 individuals and the number of generations is set to 10.

We performed three different sets of experiments. In our first experiment, we classify between two classes: asphyxia and normal cries. The second one, the classification is between deaf and normal cries. Finally, in the third one, we classify between hunger and pain cries. The evaluation was done using 10 fold cross validation and the criteria for evaluation are: percent of correct classification (Accuracy), true positive rate (TPR), true negative rate (TNR) and area under ROC curve. The results of these experiments are showed in Table 3 and Table 4 shows the best chromosome in every case and their corresponding interpretation. Due to the variable dimensionality, the part of the chromosome corresponding to the parameters for fuzzy membership functions is not included in Table 4, but Figure 2 shows some examples of resulting fuzzy membership functions obtained by the algorithm for the best classification results in each case.

### 4.1 Comparisons with Other Work

Several research results have been published using the same database used by us, or a subset of it. For example, in the work of Suaste-Rivas et al. [10] normal, deaf and asphyxia cries are classified with a FRNN obtaining results up to 88.00%. A later implementation of the FRNN on FPGA was done by Suaste-Rivas et al. [8], also classifying normal, deaf and asphyxia cries and obtaining a generalization



**Fig. 2.** Fuzzy membership functions (FMF). The images on top represent the initial FMF uniformly distributed, the lower are the FMF resulting after the evolutionary processing. (a) represents the FMF for one feature for asphyxia vs normal database, (b) for hungry vs pain database and (c) for deaf vs normal database.

rate of 94.61%. It is important to point out that these works did not use  $n$ -fold cross validation; just randomly selected training and testing sets were used.

In Barajas and Reyes' work [2] the classification with the same databases was performed. Their reported results are showed in Table 5, which we use to compare our results. The best result reported for each case is in bold.

**Table 5.** Accuracy of comparison of our method and Barajas and Reyes [2]

Data base	Proposed method	[2]
Asphyxia vs Normal	<b>88.67%</b>	84.00%
Deaf vs Normal	97.55%	<b>98.00%</b>
Hungry vs Pain	<b>96.03%</b>	95.24%

## 5 Conclusions and Future Work

The proposed method can automatically determine all the optimal parameters for the Fuzzy Relation Neural Network, as well as the parameters for the membership functions. As can be observed in Table 5, the preliminary performance is comparable with similar systems, with the advantage that the designer does not have to deal with establishing neither the FRNN parameters nor the fuzzy membership functions parameters. In addition, by automatically finding the optimal membership functions as well as their parameters for an specific problem, the results can be improved.

For future work we will use another fuzzy relational products for the learning phase, given that currently training is done using max-min composition. We will also test the proposed system with larger databases including more classes to evaluate its performance.



## References

1. Bandler, W., Kohout, L.: Fuzzy relational products as a tool for analysis and synthesis of the behaviour of complex natural and artificial systems. In: Fuzzy sets: theory and applications to policy analysis and information systems, pp. 341–367 (1980)
2. Barajas, S.E., Reyes, C.A.: Your Fuzzy Relational Neural Network Parameters Optimization with a Genetic Algorithm. In: The 14th IEEE International Conference on Fuzzy Systems FUZZ 2005, pp. 684–689. IEEE, Los Alamitos (2005)
3. Boersma, P., Weenink, D.: Praat, a system for doing phonetics by computer. Institute of Phonetic Sciences of the University of Amsterdam, Report 132, 182 (1996)
4. Engelbrecht, A.: Computational intelligence: an introduction, 2nd edn. Wiley, Chichester (2007)
5. Holland, J.: Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor (1975)
6. Pedrycz, W.: Neurocomputations in relational systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13(3), 289–297 (1991)
7. Reyes, C.: On the design of a fuzzy relational neural network for automatic speech recognition. Ph.D. thesis, Doctoral Dissertation, The Florida State University, Tallahassee, FL (1994)
8. Suaste-Rivas, I., Díaz-Méndez, A., Reyes-García, C., Reyes-Galaviz, O.: Hybrid Neural Network Design and Implementation on FPGA for Infant Cry Recognition. In: Text, Speech and Dialogue, pp. 703–709. Springer, Heidelberg (2006)
9. Suaste-Rivas, I., Reyes-Galaviz, O.F., Diaz-Mendez, A., Reyes-Garcia, C.: A Fuzzy Relational Neural Network for Pattern Classification. In: Progress in Pattern Recognition, Image Analysis and Applications, pp. 275–299 (2004)
10. Suaste-Rivas, I., Reyes-Galviz, O.F., Diaz-Mendez, A., Reyes-Garcia, C.: Implementation of a linguistic fuzzy relational neural network for detecting pathologies by infant cry recognition. In: Advances in Artificial Intelligence—IBERAMIA 2004, pp. 953–962 (2004)