

# Multi-Thread Implementation of a Fuzzy Neural Network for Automatic ECG Arrhythmia Detection

CA Ramírez-Rodríguez, MA Hernández-Silveira

Universidad Nacional Experimental del Táchira, San Cristóbal, Venezuela

## Abstract

*A fuzzy neural network was implemented using a multi-threading approach for detection of atrial fibrillation, bigeminy, and normal sinus rhythm in the MIT-BIH Arrhythmia Database. The feedforward multilayer perceptron neural network produces fuzzy outputs due to a modification of the learning algorithm that changes the crisp target labels for fuzzy target labels. The input data to the neural network consisted of nine inputs: Seven contiguous RR intervals, their average and their standard deviation.*

*The trained fuzzy neural network was implemented using concurrent thread synchronization with critical sections for mutual exclusion and process synchronization with semaphores. Concurrent process synchronisation is slower but allows data sharing among different process. Sensitivity and positive predictivity rates above 90 % for atrial fibrillation episode and duration detection were reached in the database.*

## 1. Introduction

The electrocardiogram (ECG) records the heart bioelectric potential variation as it beats. Although new techniques offer more specific diagnostic evidence in some instances, the ECG still has an important role in cardiology since it is an effective, simple, noninvasive and low-cost procedure.

### 1.1. Arrhythmia

Analysis of the ECG signal is vital as many arrhythmia are potentially dangerous and life threatening. An arrhythmia episode can be produced by an alteration of electrical impulses formation or conduction. Also, a combination of both is possible [1].

In particular, the diagnosis of atrial fibrillation (AF) is made when the electrical activity of the atria becomes completely disorganised and its component fibres are discharged irregularly and asynchronously, resulting in a general failure of the contraction mechanism of these

cardiac chambers. During this process, the atrial mass may react to more than 500 impulses per minute, completing each depolarisation-repolarisation. On direct visualisation, the atrial chambers are found to have a continuous shimmering action rather than intermittent forceful contractions. Atrial fibrillation in the ECG is represented by continuous, irregular waves on the baseline. In AF, the ventricles appears to respond randomly to the extremely rapid assault of f waves. In fact, irregularity of the ventricular rate is a characteristic of AF. The QRS retain their usual configuration and the most common ventricular rate response is about 150 to 180 per minute, but this is variable[1].

Another common alteration of the heart rhythm are premature ventricular contractions (PVC). PVCs can appear in groups in which case, the term bigeminy (B) or trigeminy is applied depending on the number of PVCs on each group. Long episodes of PVCs can be a warning of incoming severe arrhythmia.

## 1.2. Neural networks and fuzzy logic

The advancement of computer technology in recent years has intensified research on automated methods for arrhythmia analysis and interpretation. Many systems have been implemented in order to perform such tasks. Artificial neural networks are one of the most recent techniques applied in this field for designing classifiers [2-4]. In particular, the feedforward multilayer perceptron (FMP) or "backpropagation" neural network has been the most widely applied as it is capable of nonlinear discrimination among classes.

A FMP consists of a set of source nodes that constitute the input layer, one or more hidden layer of neurons, and an output layer of neurons. A neuron receives an input pattern which is then weighted by connections to finally produce an output as a non-linear function of this weighted input. FMPs learn to solve a classification problem through supervised training which results from the many presentations of a prescribed set of training examples to the FMP. The training examples consist of the input pattern and the desired target output, known also as target labels [5].

On the other hand, the utility of fuzzy sets is inherent in their ability to model the uncertain or ambiguity so often encountered in real data. Since the original work of Zadeh in 1965 [6], fuzzy logic has been successfully applied to model the uncertainty involved in many engineering problems by allowing elements to have degrees of membership in a fuzzy set; these degrees range from 1, when the element is in the set, to 0 when it is out of the set. The intention is to quantify precisely the intrinsically imprecise.

In conventional supervised classifier training, the data is supplied with target labels which unequivocally assign the patterns to one of the many classes. During the training process, similar patterns can be assigned to different classes which creates ambiguity in the information given to the FMP and could eventually produce oscillations in the learning algorithm. Nevertheless, after training, the FMP comes out with a decision surface which attempts to separate in the most optimal way the classes (although in reality they overlap).

Instead of limiting the problem to separate the classes, a more plausible approach is to identify the area of overlapping and train the FMP with this information. In this way, the FMP can identify previously unseen patterns as belonging to a "crisp" area or to an overlapping or fuzzy area (as defined by the training set). In the former case, the system is capable of signaling uncertain QRS patterns for further analysis in a second stage based on a different approach or the decision deferred until more discriminative data is available. This improvement in the learning stage can be achieved by substituting crisp target labels with fuzzy target labels.

### 1.3. Threads

In WIN32, a process is a program that is currently loaded into memory. Processes do not perform any action. The entity that does something inside a program is called a thread. Each program has at least one thread. All Windows 3.1 programs, by definition, had only one thread. The idea that each program can have its own thread or that a program can have multiple threads is specific to operating systems such as UNIX, OS/2, Microsoft Windows 9x, Windows NT<sup>o</sup> and recently Windows Me and Windows 2000.

At any time, these operating systems can have a number of threads running. Multitasking is the term used to describe the operating system feature that allows it to run multiple threads concurrently. In the case of Windows<sup>o</sup> 9x, it assigns a time slice known as quantum to each of these threads [7].

Computer based biomedical signal acquisition can be benefited from multitasking programming. The main advantage over sequential programming techniques is steady signal acquisition rates when the computer is also performing graphical and signal processing tasks.

The classic problem encountered when using threads involves a global variable that can be accessed by more than one thread. Variations of this scheme involve a series of threads, all of which have to access the same file, the same DLL, the same communications resource, or any of a number of different objects. This problem is solved by using concurrent thread synchronization with critical sections for mutual exclusion or process synchronization with semaphores.

In this article, a fuzzy neural network for automatic arrhythmia detection using multi-threads programming techniques is proposed. The model is evaluated on selected records of the MIT-BIH Arrhythmia Database. Also, a study is carried out to understand the different synchronization schemes for handling concurrency.

## 2. Multi-thread arrhythmia detection

### 2.1. ECG data

The data used to train and test the arrhythmia detector is obtained from the MIT-BIH Arrhythmia Database. The database contains 23 records (numbered from 100 to 124 inclusive with some numbers missing) that serves as a representative sample of the variety of waveforms and artifact that an arrhythmia detector might encounter in routine clinical use. It also has 25 additional records that present rare but clinically important phenomena that would not be well-represented by a small random sample of Holter recordings. They were selected because of the rhythm, QRS morphology variation, or signal quality may be expected to present significant difficulty to arrhythmia detectors [8]. Table 1 shows the subset of records used throughout this study. Records 106, 201 and 202 were used for training and evaluation of the neural network. All other records were only used for evaluation purposes.

Table 1. ECG database records.

Record	Total episode duration			
	AF	B	NSR <sup>a</sup>	P <sup>b</sup>
106	0:0	7:15	22:36	0:0
201	10:06	0:0	12:57	0:0
202	9:46	0:0	19:31	0:0
203	24:15	0:0	0:0	0:0
210	29:30	0:23	0:0	0:0
217	4:12	0:42	0:0	25:10
219	23:47	0:08	6:01	0:0
221	29:17	0:03	0:00	0:0
222	8:33	0:0	15:57	0:0

<sup>a</sup> Normal sinus rhythm

<sup>b</sup> Paced rhythm

## 2.2. FMP neural network training

A total of 300 input patterns, 100 for each class, were extracted from records 106, 201 and 202 representing AF, B and NSR rhythms. The annotation file "atruth" included in the database was used to select the rhythm within each record. The input patterns consisted of nine real values. The first seven values corresponded to contiguous RR intervals measured in seconds. The eighth value was the average of the seven RR intervals. Finally, the ninth value resulted from calculating the standard deviation of the seven RR intervals of the previously obtained average.

The original binary target labels were fuzzified using a modification of the algorithm proposed in [9] that allows modeling of non-convex classes. In this modified algorithm, the input patterns of each class are clustered. Then, fuzzy set membership functions are constructed for each cluster and a final class membership function is obtained as the union of the cluster membership functions. Thus, for every input pattern  $x_k$  and every class  $C_i$ , the target label is calculated as follows:

$$T_{ik} = \begin{cases} 1 & \text{if } x_k \in C_i, \\ \mu_i(x_k) & \text{if otherwise} \end{cases} \quad (1)$$

where  $\mu_i(x_k)$  is the membership value of input pattern to class  $C_i$  and target value of 1 is assigned to all patterns which were initially labeled as belonging to a specific class by the hard partition (i.e. binary target labels). In this way, the FMP neural network is provided with labels representing the membership of a training pattern to all classes while preserving the maximum degree of membership to the class it was originally assigned to. Training patterns in class overlapping areas will have less influence in determining the neural network weight vector, but the information contained in the hard partition is preserved.

A FMP with a 9 – 8 – 3 network structure was trained using the input patterns and the fuzzy target labels. Training was carried out for 10000 iterations with a learning rate of 0.3 and a momentum term of 0.2 on a Sun Microsystems SUN Ultra 60 workstation. The programs necessary to train the network were coded in ANSI C++.

## 2.3. Multi-thread implementation

The multi-thread implementation of the trained fuzzy FMP was carried out using the programming environment Borland Builder C++ 5.0 running under the operating system Microsoft Windows 2000<sup>®</sup>. The hardware platform consisted of a HP Vectra with a Pentium<sup>®</sup> III 866 Mhz processor and 128 Mega Bytes in RAM.

Two approaches were implemented. The first one was based on concurrent thread synchronization with critical sections. The program was divided into three different

threads. One for reading the ECG signal from the hard disk, another for QRS and arrhythmia detection and the last one for printing. In this case, signal acquisition is carried out from the hard disk in order to evaluate the arrhythmia detector with the database. However, the acquisition thread could be reading the information from the computer ports or any other source connected to an ECG acquisition system. Critical sections were introduced for all global variables that could be accessed by the different threads in a synchronize way.

The second approach based on process synchronization with semaphores also involved three threads. The difference here was the way how the ECG signal was handle within the threads. A memory file-mapped object was declared to store the ECG in a memory area that could be accessed by different processes at the same time, in other words, an special handle address (0xFFFFFFFF) was implemented to instruct the operating system to map a view of shared memory backed by the system paging file, commonly known as swap file. Next, a new application was created were the shared ECG data was used to calculate and display the heart rate.

## 2.4. Evaluation

The proposed fuzzy FMP neural network was evaluated on selected records shown in Table 1. These records were chosen because AF and B episodes were present. The annotation of the episodes was as follows. A QRS detector developed in early work [9] was applied to the ECG signal and RR intervals calculated. These intervals together with the average and standard deviation values were input to the FMP after 15 seconds of processing time. This 15 seconds period was necessary for stabilization of the QRS detector. As soon as the FMP produced an output, a set of counters were initialized. An AF or B episode was annotated when five or more consecutive patters were assigned to one of this classes by the FMP. The episode was considered to has finished as soon as four different consecutive responses from the FMP were obtained. If a particular pattern produced a value above 0.3 in all output neurons, it was considered as fuzzy and the previous pattern classification was assumed.

## 3. Experimental results

Table 2 shows the statistics obtained by using the *epic* program included in the MIT-BIH Arrhythmia Database for AF detection. The *epic* program is in conformity with recommended practice of the Association of Medical Instrumentation (AAMI). The AAMI has developed a standard protocol (designated EC38) for evaluating automated ECG analysis algorithms.

Table 2. Performance statistics of the classifier

Record	% Se		% +P	
	Episode	Duration	Episode	Duration
201	100	99	75	93
202	100	98	100	97
203	100	96	100	97
210	100	98	100	99
217	95	86	50	57
219	89	82	100	91
221	100	92	100	97
222	100	95	96	64
Gross	98.14	93.30	91.51	90.66

These experimental results show a fairly good performance on most records. However, a low episode and duration positive predictivity can be observed for record 217. This record has 14 episodes of AF for the test period. The total duration of these episodes is approximately 5 minutes and 33 seconds. Most of this episodes are very short (less than 5 beats in many cases) making difficult for the neural network to detect. Evaluation of the fuzzy neural network for bigeminy classification was performed using an in house developed episode evaluator. Detection statistics showed an episode sensitivity of 91.4 % and a positive predictivity of 92.3%.

The multi-thread applications following the two proposed approaches were successfully implemented. Compared to a single thread application, the multi-thread applications did not modified the signal morphology, because the processor performed different tasks in several time slices. Also, it was corroborated that critical sections are less computationally expensive than synchronization with semaphores. However, the later allowed the possibility of synchronizing different processes for data sharing while critical section only allowed synchronization among threads. Figure 1 shows a print screen of two different applications that use common data to calculate the heart rate.

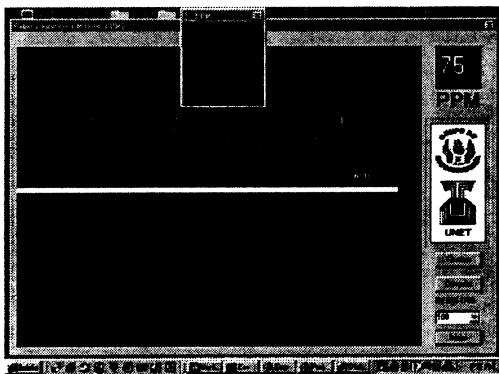


Figure 1. Print screen of the multi-thread application using semaphores.

## 4. Conclusions

A fuzzy neural network arrhythmia detector has been implemented following a multi-thread approach. The model has proved to be reliable and computationally efficient for AF and B detection. Global benchmark parameters measured in a subset of the MIT-BIH Arrhythmia Database produced gross episode detection and duration statistics above 90%. The multi-thread implementation using semaphores has proven to be more computationally expensive than the one based on critical sections. However it has the advantage of sharing data with other process using file-mapping.

## Acknowledgements

This work was supported by CONICIT Project S1-97003816 and the Decanato de Investigación of the Universidad Nacional Experimental del Táchira, Venezuela.

## References

- [1] Watanabe Y, Dreifus L. Cardiac Arrhythmias. Electrophysiologic basis for clinical interpretation. USA: Grune & Stratton, Inc.. 1997.
- [2] Silipo R, Marchesi C. Artificial neural networks for automatic ECG analysis. IEEE Trans. on Signal Processing. 1998; 46-5:1417 - 25.
- [3] Afonso VX, Tompkins WJ, Nguyen T, Luo S. ECG beat detection using filter banks. IEEE Trans. Biomed. Eng. 1999; 46-2:192-202.
- [4] Artis SG, Mark RG, Moody GB. Detection of atrial fibrillation using artificial neural networks. Computers in Cardiology 1991;173-176.
- [5] Haykin S, Neural networks: A comprehensive foundation. USA: Prentice Hall. 1994.
- [6] Zadeh LA. Fuzzy Sets. Information and Control. 1965; 8:338-52.
- [7] Charte F. Programación con C++ Builder 5. España: Anaya Multimedia. 2000.
- [8] Moody G, Mark R. The impact of the MIT-BIH Arrhythmia Database. IEEE Eng. Med. Biol. Mag. 2001; May/June: 45-50.
- [9] Ramirez C. QRS detection in ECG using fuzzy neural networks. In: Dagli C *et al.* Intelligent engineering systems through artificial neural networks. USA: ASME Press, 2000: 375-80.

Address for correspondence.

Carlos A. Ramirez R.  
UNET

Grupo de Bioingeniería, Decanato de Investigación, Av Universidad, Sector Paramillo, San Cristóbal, Estado Táchira.  
cram@unet.edu.ve.