Detection of Atrial Fibrillation Using Artificial Neural Networks

SG Artis, RG Mark, GB Moody

Harvard-MIT
Division of Health Sciences and Technology, Cambridge, MA

Abstract

Artificial neural networks (ANNs) were used as pattern detectors to detect atrial fibrillation (AF) in the MIT-BIH Arrhythmia Database. ECG data was represented using generalized interval transition matrices, as in Markov model AF detectors[1]. A training file was developed, using these transition matrices, for a backpropagation ANN. This file consisted of approximately 15 minutes each of AF and non-AF data. The ANN was succesfully trained using this data. Three standard databases were used to test network performance. Postprocessing of the ANN output yielded an AF sensitivity of 92.86% *and an AF positive predictive accuracy of* 92.34%.

1 Introduction

Cardiac arrhythmias may be classified using both morphology analysis, which classifies beats by shape, and timing analysis, which classifies beats by their arrival rates. Timing analysis is used to classify a subset of rhythms that includes premature beats, rapid heart rate, slow heart rate, and more generally, beats with irregular arrival times. Atrial fibrillation (AF) is a heart rhythm which is usually characterized by beats with normal morphology and with irregular arrival times. AF detection is most often based upon timing analysis.

Atrial fibrillation detection is important because it is a common arrhythmia which often indicates underlying heart disease. AF can also complicate automated detection of other arrhythmias. This happens because it becomes impossible to define the prematurity of a beat in relation to its surrounding beats in AF. Because of this, atrial fibrillation detectors are usually included in automated arrhythmia analyzers. AF detection is difficult, however, because beat intervals in AF form no recognizable pattern, unlike other cardiac arrhythmias. Attempts have been made to detect AF based on R-R interval sequences using a variety of statistical methods [1] but there is room for improvement in these techniques.

Pattern classifiers exist in many forms, and artificial neural networks (ANNs) represent an important subset of these classifiers. ANNs are attractive for solving pattern recognition problems because few assumptions about the underlying data need to be made. The task of the operator of an ANN is to separate the data into subsets. The network will be able classify these subsets according to type as long as they are distinct. Neural network training requires appropriate training data, pre-processing and post-processing algorithms, an appropriate network topology, and a training algorithm, as well as evaluation databases. This document will present the design and evaluation of a technique which detects AF in the presence of other cardiac arrhythmias using a backpropagation artificial neural network.

2 Databases

Three databases were used throughout this study. The first consisted of a subset of the MIT-BIH ECG database, summarized in table 1, which was used as a development database. A subset of this database was used for training of the ANN. The second database, used as an evaluation database and summarized in table 2, has been collected from Holter recordings specifically to test R-R interval-basedAF detectors. This database, called the MIT-BIH Atrial Fibrillation/Flutter Database [2] contains 25 ten-hour records, each from a unique subject, and including over 300 episodes of AF. The database consists of two annotation files for each recording - one containing QRS complex arrival times (for R-R interval measurements) and the other containing accurate rhythm change annotations. In this database, beat labels indicate the time, but not the type, of each beat so the quantities of APBs, PVCs, Normal beats, and other beats are unknown. The third database was the AHA Database for

0-8186-2485-X/92 $3.00 © 1992 IEEE

Record	AF Beats	Total Beats
201	855	2,039
202	935	2,145
203	2,306	3,106
207	0	2,384
209	0	3,050
210	2,585	2,684
213	0	3,293
219	1,658	2,311
220	0	2,068
221	2,346	2,461
222	817	2,605
223	0	2,642
Totals	11,502	30,788

Table 1: Neural network development database

Rhythm	Hours	Episodes	Beats
AF	94.99	319	532,276
AFL	1.98	40	16,563
Other	163.03	293	733,550
Total	260.00	652	1,282,389

Table 2: Summary of the contents of the MIT-BIH Atrial Fibrillation/Flutter Database

the Evaluation of Ventricular Arrhythmias [3], which was used to calculate false positive rates for the detector. A visual inspection of the AHA database was made to remove 13 records[1] that contained either AF or pacemaker beats.

3 Methods

3.1 Pre-processing ECG data

Detection of AF using a backpropagation ANN requires significant preprocessing of the input data stream. This is because AF produces QRS complexes with nearly random arrival times. This is not a pattern that is likely to be recognized by a neural network, for it is, in fact, no pattern. Some form of preprocessing must be used to translate this inherent aperiodicity into a form that has a recognizable and unique pattern. In this experiment, a generalized interval transition matrix, similar to that discussed by Moody and Mark in [1], was applied as input to the neural network. This process classifies each interval into the set of $\{Short, Normal, Long\}$ using an estimator of 'correct' interval length, and then assigns pairs of these intervals to cells of a transition matrix. A short interval is 85% of the estimator and a long interval is

[1] 1007, 1009, 2002, 2005, 3003, 5005, 5009, 6004, 8003, 8004, 8005, and 8010.

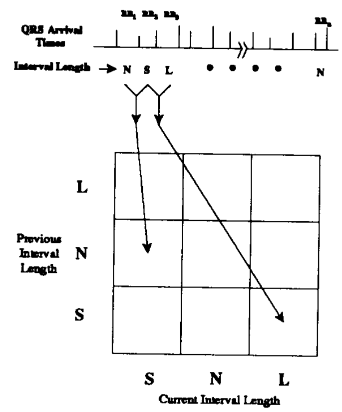

Figure 1: Dividing an interval stream into transition matrices

115% of the estimator. The transition matrix cells are applied as input to the ANN. Figure 1 demonstrates this process. In addition to the above pre-processing technique, we found that network performance could be improved significantly by using a sliding 'window' of intervals, rather than by breaking the R-R interval stream into separate matrices. This means that the intervals $[RR_0 \rightarrow RR_{n-1}]$ form the first matrix, the intervals $[RR_1 \rightarrow RR_n]$ form the second matrix, and so on until the end of the input data. This technique results in a one-to-one correspondence between matrices and beats in the interval stream, and in this way, an attempt was made to label each beat individually, rather than in groups of n.

3.2 Neural Network Processing

The matrices described in the previous section were applied to the inputs of a neural network. A neural network was created using a public-domain software package called StarNet [4], using a VaxStation 2000 running BSD Unix 4.3. This package can be used to create backpropagation networks of arbitrary size and implements a form of the generalized delta training rule (with momentum term) [5]. A network was created with 9 input units, corresponding to the 9 cells of a 3-by-3 interval transition matrix. The output layer consisted of one unit which was trained to represent AF as 1.0 and all else as 0.0. The hidden layer size was empirically set to 12 (determined by training networks with varying hidden layer size from 4 to 16 units).

Training these networks required a training file that

Record	Rhythm Type	Duration (min.)
106	NSR & Bigeminy	5
209	NSR & SVTA	5
209	NSR & APBs	5
122	NSR	5
202	AF	2
201	AF	5.5

Table 3: Neural network training file data

consisted of several common arrhythmia patterns likely to be found in ECG data. The arrhythmias which were found to provide high differentiation in network output were normal sinus rhythm (NSR), NSR with ventricular bigeminy, NSR with atrial tachycardia, NSR with frequent atrial premature beats, NSR with ventricular premature beats, and AF. A training file was created that contained approximately five minutes each of all non-AF rhythms encoded in interval transition matrices. Originally an amount of AF equal to the total of the other rhythm types was used in the training file, but it was found that a somewhat smaller amount reduced the number of false positives, and thus increased positive predictive accuracy. A relatively small subset of all possible rhythms are in the training set. This has two reasons: a large training set can dramatically increase training time, and larger training sets did not significantly improve performance. The training data was extracted from records in the MIT-BIH Arrhythmia Database and is summarized in table 4. To train the network, the training file from table 4 was applied as input to the neural network. The individual matrices were presented in random order. Training was performed until the total RMS error dropped to $\sim 0.1\%$. The resulting weights were used to implement the final detector.

3.3 Post-processing Neural Network Data

Post-processing utilized a 30-point moving average and two thresholds. The post-processing technique is displayed in Figure 2 for a 6-point moving average by way of example. This technique was designed to generate a label for each individual beat in a data record. As shown, 6 outputs from the neural network are created from input matrices that include a given interval. Post-processing averages each of these 6 values to arrive at a single number. This number is compared to two threshold values, T_1 and T_2. If the number exceeds the high threshold, then the beat corresponding to the given interval is labelled AF. If the number is lower than the low threshold then the beat label is left unchanged from its original value (or assigned a Normal label if it is unlabelled). If the number falls between the two thresholds, the assigned label is the same as the previous assigned label. This is designed to account for the tendency of the heart to maintain its current rhythm, rather than change it frequently.

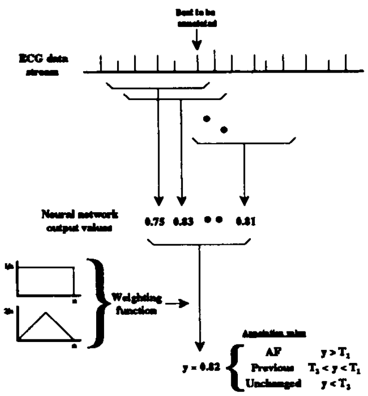

Figure 2: Post-processing ANN output data using a 6-point moving average

4 Results

4.1 Post-processing Neural Network Data

The post-processing techniques from the previous section were applied, and a histogram of some of the results is shown in figure 5. In theory, the two post-processing thresholds could be used to generate receiver-operating curves, and to manipulate the sensitivity and positive predictivity of a neural network. In practice this was generally not the case because of the very wide separation of network output values. The ANNs did not usually generate an even distribution of values between 0.0 and 1.0 which could be affected by thresholds, but instead generated two widely separated clusters around 0.0 and 1.0.

4.2 Evaluation of ANN performance upon Development Database

Table 5 compares the ANN detector developed here to the detectors described in [1] applied to the same development database.

4.3 ANN Performance Upon Evaluation Database

The final process of developing an AF detector was to apply the detector to databases that were entirely

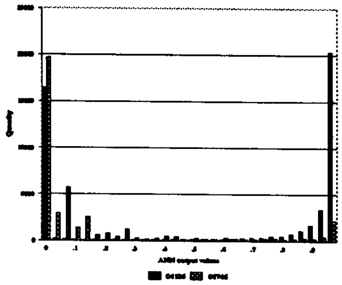

Figure 3: Histograms of ANN output values showing clusters around the extremes

Detector	Se	+ P
Markov Model	90.03%	80.14%
– filt. & interp.	96.09%	86.79%
Predictor array	95.44%	80.52%
ANN	84.87%	75.38%

Table 4: Summary of detector performances upon Development database

disjoint from the training dataset. In this case, the databases evaluated were the MIT-BIH Atrial Fibrillation/Atrial Flutter Database, and the AHA database. Table 5 summarizes detector performances upon the two databases.

5 Alternate Performance Measures

AF detectors must correctly determine the prevailing rhythm. For this reason, the ANN detector's *episode* sensitivity and positive predictive accuracy were determined. For purposes of these measurements, a section of ECG data was considered an episode if its annotations remained consistent for more than 60 beats. To correctly detect an episode, the detector was required to generate an episode of its own that encompassed more than 50% of the duration of the true episode. Episode performance measures for other detectors are unavailable, but for the ANN detector an episode sensitivity of 93.5% and episode positive predictive accuracy of 96.7% were determined for the MIT-BIH AF Database.

Detector	AF DB Se.	AF DB +P	AF DB FPR	AHA DB FPR
Markov model	99.59%	65.97%	21.32%	
– filter & interp.	93.58%	85.92%	6.37%	
Predictor array	75.79%	91.93%	3.76%	
ANN	92.86%	92.34%	3.04%	2.83%

Table 5: Performance summaries for AF detectors upon the two evaluation databases

6 Conclusions

BP ANNs are demonstrated to provide excellent rhythm classifying characteristics. Timing information alone is sufficient for network training. Interestingly, the ANN performed significantly *better* upon the evaluation databases than upon the development database. This is at least partly because the ANN was trained using only a small subset of the test database, and the test database contains rhythms specifically designed to confuse detectors.

The ANNs described herein will fail in certain situations. These circumstances occur when AF occurs with regular ventricular rates, and so this technique is not always suited for detection of long-term, clinically treated AF. A patient whose ECG changes rapidly between rhythms can have detectable AF, but also generates high false positive rates[6]. These situations are problematic for all detectors which utilize interval analysis methods only.

ANNs may be applied to this problem in many other ways. For example, heart rate may be added as an input, interpolation between matrix values may be performed, or other network topologies may be used. Some of these techniques are explored in detail in [6].

References

[1] G. B. Moody and R. G. Mark. A new method for detecting atrial fibrillation using R-R intervals. *Computers in Cardiology*, pages 227–230, 1983.

[2] Harvard University-M.I.T. Division of Health Sciences and Technology. *MIT-BIH Arrythmia Database Directory*, August 1988.

[3] K. L. Ripley and G. C. Oliver. Development of an ECG database for arrythmia detector evaluation. *Computers in Cardiology*, pages 203–209, 1977.

[4] Yoshiro Miyata. *A User's Guide to PlaNet Version 5.6.* Computer Science Department, University of Colorado, Boulder, January 1991.

[5] R. Hecht-Nielsen. *Neurocomputing.* Addison-Wesley Publishing Co., 1990.

[6] S. G. Artis. Detection of atrial fibrillation using artificial neural networks. Master's thesis, Massachusetts Institute of Technology, 1991.