# Cardiac Condition Detection using Artificial Neural Networks

By

Sean Franklin & Joseph Wallcave

Senior Project

ELECTRICAL ENGINEERING DEPARTMENT

California Polytechnic State University

San Luis Obispo

June 2013

## Table of Contents

## List of Figures

## List of Tables

## Acknowledgements

We would like to thank Dr. Xiao-Hua (Helen) Yu for her guidance throughout the process of this project, as well as Dr. Dennis Derickson for his generosity and support.

We are thankful for our families and their strong support of this project.

## Abstract

Electrocardiography (also called ECG or EKG) is a non-invasive process using electrodes to interpret the electrical activity of the heart and thus measure the rate and regularity of heartbeats. This is useful because it can be used to determine the size and position of the chambers, detect any damage to the tissue, and detect any cardiac pathologies that might be present. Our goal is to detect important characteristic points of ECG signals to determine if the patient's heart beat is normal or irregular, accentuating one of several already pre-determined heart diseases. This will be accomplished by acquiring various ECG signals from an online database, and feeding the signal's characteristic points through an artificial neural network which will train, test, and validate the ECG signal appropriately.

## Introduction/Background

Electrocardiography (ECG) is an interpretation of the heart's electrical activity (amplitude) over time. By accurately measuring the heart beat of the patient, ECG has been clinically proven as an effective method for diagnosing heart diseases and other heart irregularities. An ECG test is performed with electrodes hooked up to a patient's chest to pick up electrical activity of the heart. These electric signals are then sent to an external recording device so that visual data can be observed and compared to a normal heart rate in order to determine the health of the patient's heart and heart chambers [4]. Due to its non-intrusive approach, ECG is widely accepted by both patients and doctors as a quick and effective procedure with real-time results. Figure 1 depicts the general hookup of electrodes to the human body to pick up neuron activity in the heart chambers. Accurately acquiring the electrical activity is of course critical to the doctor's synopsis of the patient, so blocking out unwanted signal traces such as white noise, other respiratory and muscle noise, and 60 Hz power line noise is imperative. Basic linear filters cannot be used when trying to rid of these unwanted signals since accuracy is of the utmost importance. A more systematic approach must be taken to deliver precise results.

Electrocardiography has developed significantly since its first contributions by Willem Einthoven's beginning in 1901. Then, Einthoven's string galvanometer invention paved the way for low current detection and recording of the human heart's electrical activity. However, Einthoven's ECG prototype itself was a laboratory and instead of electrodes the patients would immerse themselves into containers of salt solutions from which the recordings were made [1]. Just 20 years after Einthoven's breakthrough, the string galvanometer decreased in size from 600 lbs to 30 lbs. Today, with the development of computers and microelectronics, electrocardiography treatment has made great strides in terms of accuracy and reliability.

Currently, there are ECG records in the MIT-BIH (Massachusetts Institute of Technology-Beth Israel Hospital) Arrhythmia Database. Several signals will be chosen to simulate in MATLAB based on their determined heart problems at specific points of interest. By consulting with various references with information to accurately determine the specific heart irregularity/disease, we will be able to pick out the points of interest ourselves and acquire the QRS complexes in terms of amplitude (millivolts) and length (time in seconds) [8].

The project will utilize artificial neural networks (ANN) to “train” data results (QRS complexes). The ANN will be able to adapt to the behavioral model of non-stationary electric activity of the heart and provide the most accurate results.

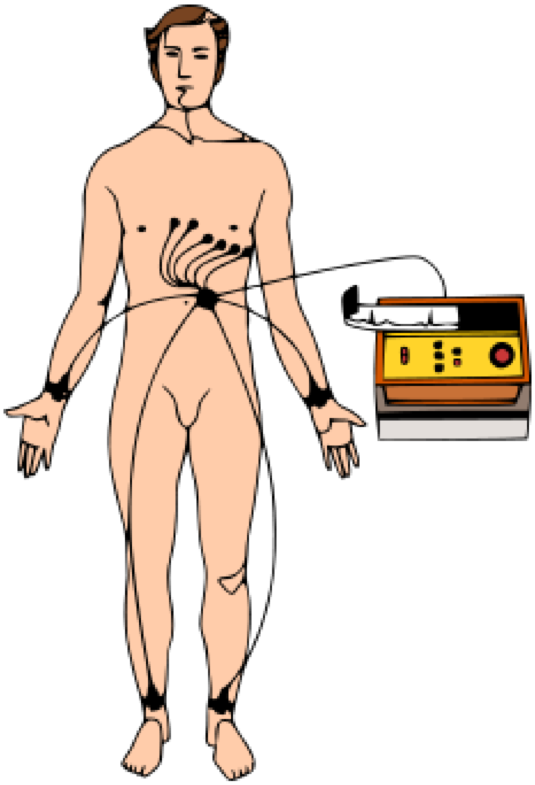


**Figure 1: Electrocardiography Setup: The following is an example set-up for an electrocardiography diagnostic test. The standard 12-lead system consists of 10 electrodes: 6 on the chest/heart, 1 on each inner forearm and 1 on the inside of each leg just above the ankle. Although only 10 leads are used, 2 of the leads are referenced to two separate points (two pathways) to give a 12-dimensional view of the heart’s electrical activity, hence the 12-lead system [15].**

## Background on Neural Networks

Analysis of ECG is a complex issue that usually requires the expertise of a doctor to determine if the results are normal or classify any abnormalities that occur. This can be expensive and time consuming for someone who simply needs preliminary results. It can also save doctors time if they only need to confirm results. However, the non-linear, pattern recognition tasks required of ECG analysis can take hours using conventional computing methods [3].

Conventional computing is not necessarily the best solution for pattern recognition, as is required in applications such as analysis of an ECG signal. These types of applications can take a long time via conventional computing methods. However, the human brain is amazingly adept (and fast) with pattern recognition. To solve this problem, we can use the concept of an Artificial Neural Network (ANN). The idea of the ANN is derived from the massively parallel connection of neurons in the human brain (nervous system). In an artificial neural network, a computer is made to mimic these connections [3] [4].

To understand the idea of the artificial neural network, we must first understand the concepts it is based on. A silicon IC has a response time in the range of nanoseconds ($10^{-9}$ s), but a neuron has a response time in the millisecond ($10^{-3}$ s) range. It would seem as if the brain should be slower that a computer, however this is not the case. This can be attributed to the fact that the brain is massively parallel. It consists of billions of neurons connected to each other by trillions of connections (called synapses) [2].

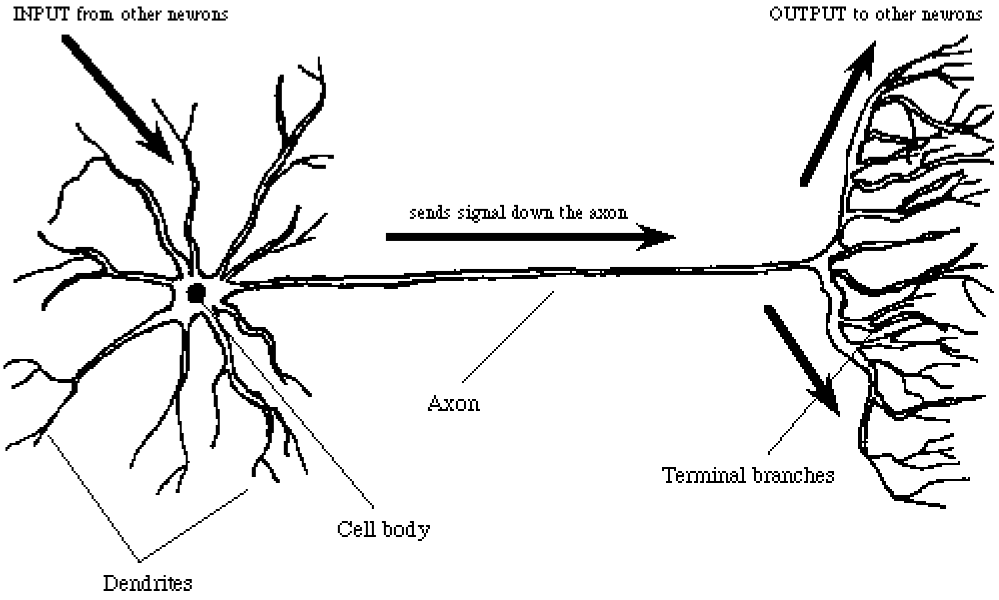


**Figure 2: Biological model of a neuron. Dendrites receive several differently weighted inputs and the output is sent down the axon to several other neurons [16].**

As seen in Figure 2, a neuron has several inputs as well as several outputs that contribute to the parallel structure of the brain. All the inputs (consisting of different strengths) are added together and then output to many more neurons.

While we can't have billions of processing units as the brain does, we can use the properties of the biological neural network to model small parts of the brain to perform a specific task [4]. There are several advantages to using this approach: [2]

1. **Non-linear**
   Many real world problems, such as pattern recognition, are not linear.

2. **Input-Output mapping**
   This allows for a learning mechanism where one can feed a given input and specify the corresponding output. If the output does not match the specified output, the system can adjust its parameters.

3. **Adaptability**
   The ability to adjust parameters allows neural networks to adapt to changes in the environment.

4. **Evidential Response**
   Decisions can be made with a certain measure of confidence.

5. **Fault Tolerance**
   Failures result in a graceful degradation.

6. **VLSI Implementation**
   Transistors can be exploited to build very efficient neurons.

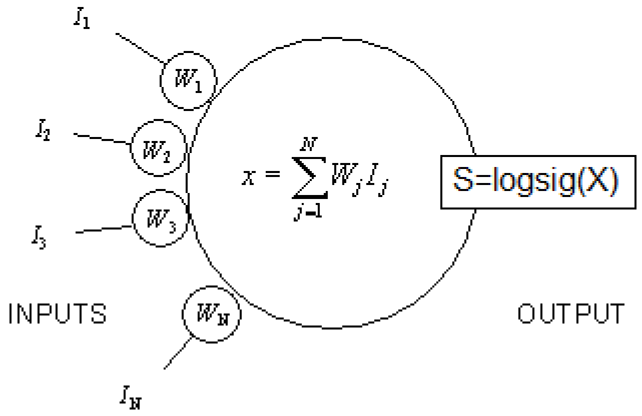


**Figure 3: Equivalent electronic model of a neuron. Several weighted inputs are summed to make up the net signal. This is then output to other neurons [17].**

As seen in figure 3, we can create an equivalent electronic model of a neuron with several inputs with different weights. The net signal of the neuron is the sum of the weighted inputs [3]. The network can "learn" by changing the weight of any input to any neuron. We will discuss our inputs and outputs in a later section.

By consulting with various references with information to accurately determine the specific heart irregularity/disease, we will be able to pick out the points of interest ourselves and acquire th QRS complexes in terms of amplitude (millivolts) and length (time in seconds).

### Neural Network

Our neural network consisted of 10 input nodes, a hidden layer with 100 neurons, and 6 output nodes as seen in figure 4. We chose 100 hidden neurons for increased accuracy and due to the fact that it did not require a large computational sacrifice but increased accuracy. We used a scaled conjugate gradient back-propagation training function with MATLAB over 18 iterations because scaled conjugate gradient algorithms don't require much computational power and are thus very fast. We set sigma, which determines change in weight for second derivative approximation, to be 5.0e-5. We found this to be optimal for accuracy. We set lambda as 5.0e-7 because it was a good balance between performance and speed.

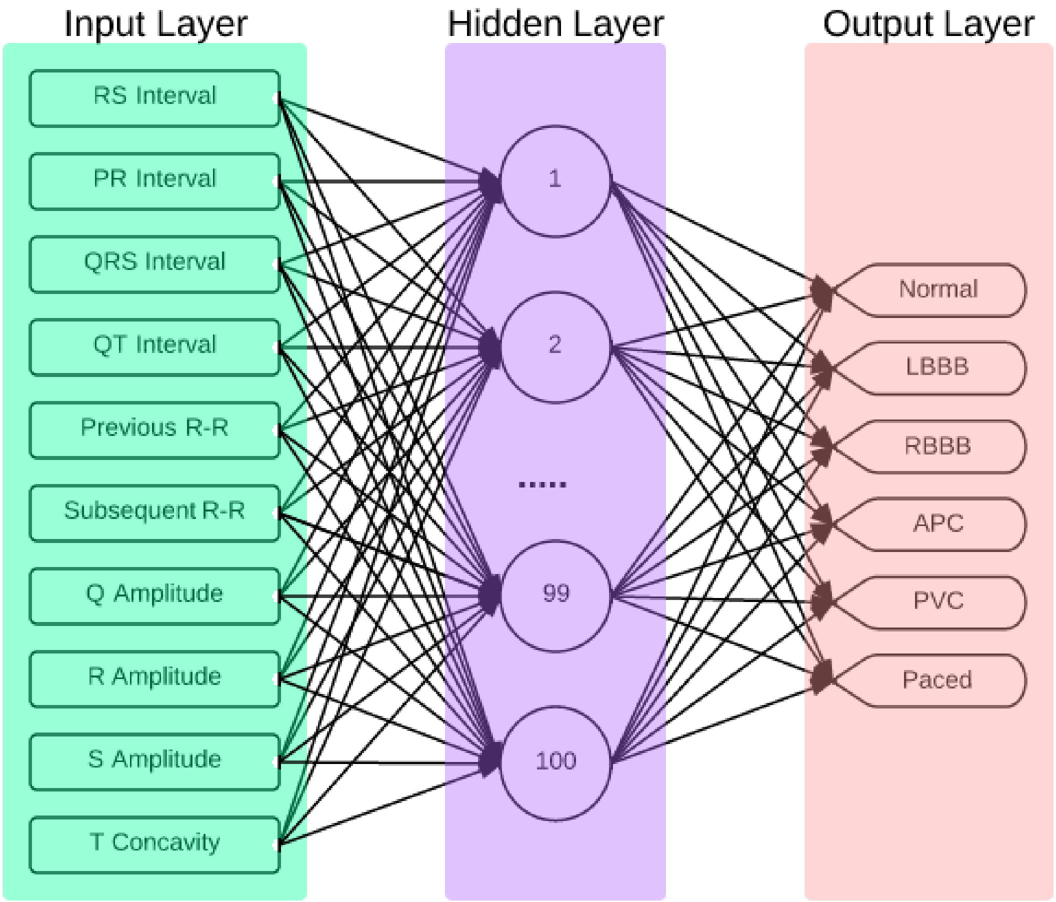


Figure 4: Architecture of the neural network used.

The input data is fed to the neural network which then computes the output as either normal, LBBB, RBBB, APC, PVC, or paced, based on the expected values in the target data [9].

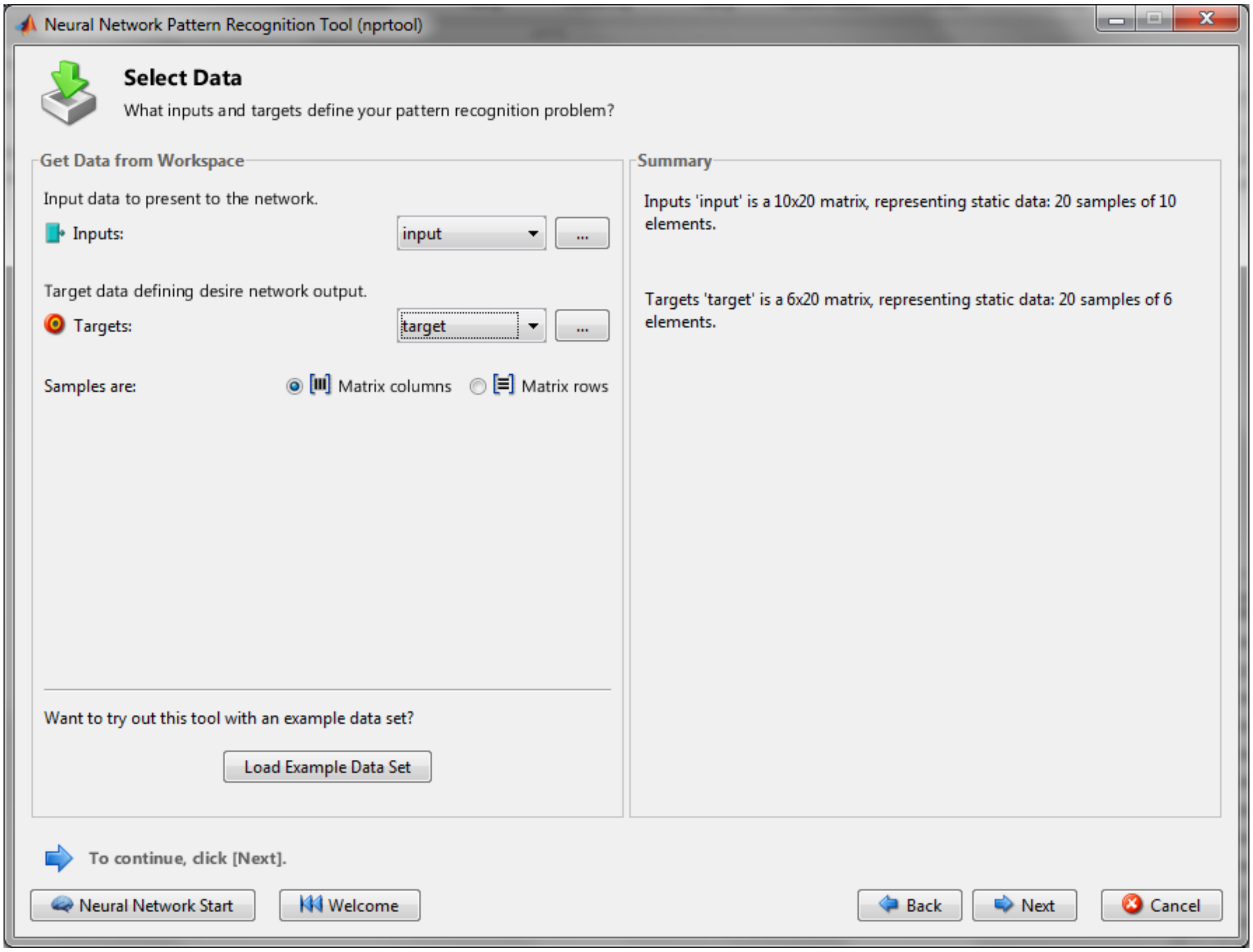
The activation function of the output layer is a log-sigmoid transfer function. We used one hot encoding for interpreting the max output in each column as 1 and the rest as 0, assuming that there is only one output that can be 1. This does not consider a situation where more than one abnormality is present. For example, in a situation where an output is 0.48, if the 0.48 is the maximum value in the output column for the wave, then 0.48 would be interpreted as a 1. Here is an example of the output of one waveform:

**Table 1: Output table of the NN for the normal wave form.**

| | |
|---|---|
| Normal sinus rhythm | 0.992880 |
| LBBB | 0.019367 |
| RBBB | 0.000039 |
| PVC | 0.000231 |
| APC | 0.001282 |
| Paced Beat | 0.000027 |

The output of the NN should be a list of 1's and 0's as it computes which disease is present based on the inputs and target data (which is also 1's and 0's). Here, the numbers are slightly off the theoretical 1's and 0's because of some error in the training, validating, and testing samples used. In this example, the normal sinus rhythm would be identified as the 1 and is within 0.7% of 1. LBBB, which had the next closest value to 1 is not even within 98.0% of 1. The Data set was 20 samples with training being 50%, validation being 35%, and test being 15%.

MATLAB has a neural network toolbox that was helpful in quickly making changes to variables such as network size and testing additional sets of data. The toolbox also generated the script in Appendix C, which is editable for changing parameters.



Neural Network Pattern Recognition Tool (nprtool)

**Select Data**

What inputs and targets define your pattern recognition problem?

Get Data from Workspace

Input data to present to the network.

Inputs: input

Target data defining desire network output.

Targets: target

Samples are: ● Matrix columns ○ Matrix rows

Want to try out this tool with an example data set?

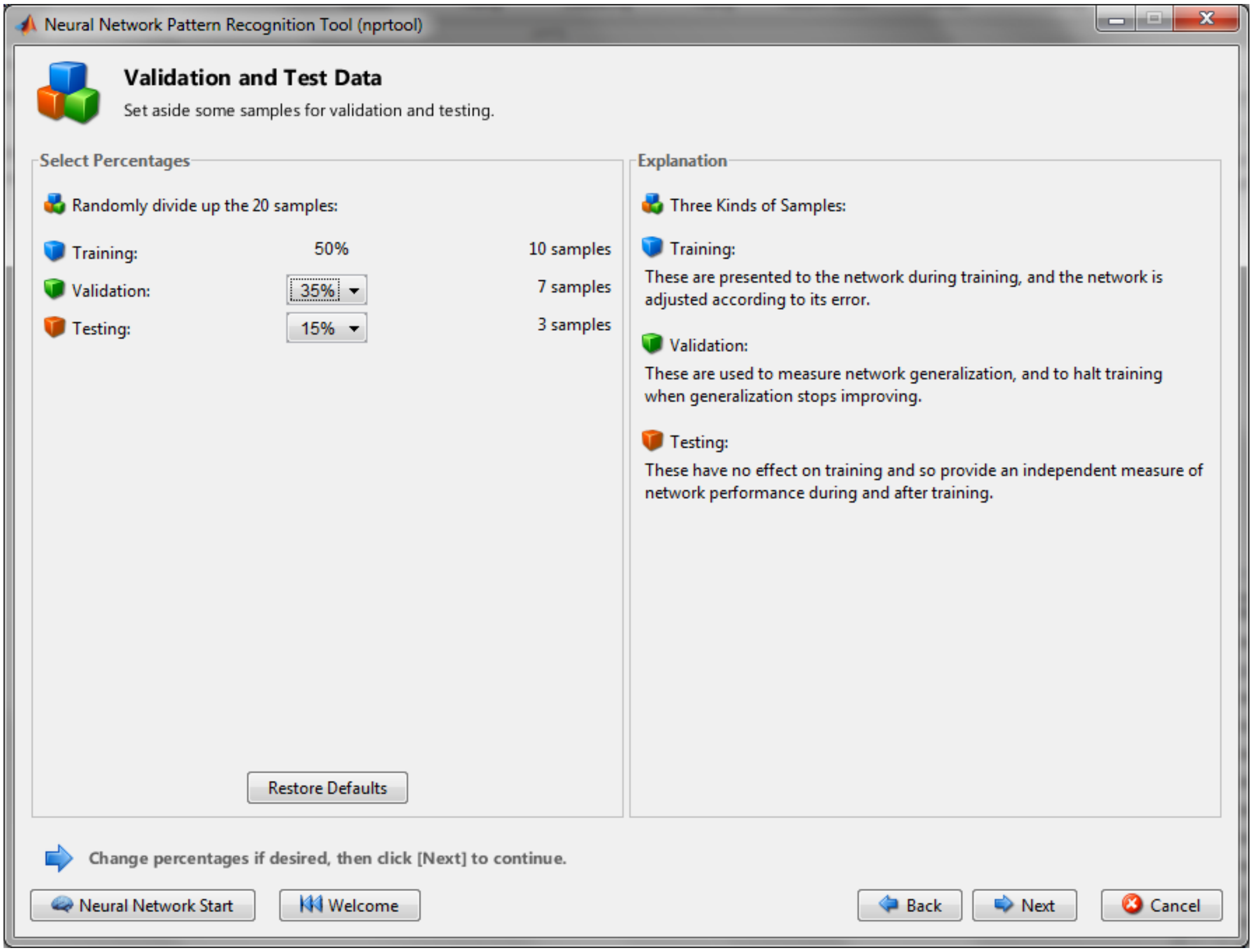
Summary

Inputs 'input' is a 10x20 matrix, representing static data: 20 samples of 10 elements.

Targets 'target' is a 6x20 matrix, representing static data: 20 samples of 6 elements.

To continue, click [Next].

Figure 5: First page of the NN toolbox allows the user to select input and target data.



Neural Network Pattern Recognition Tool (nprtool)

Validation and Test Data

Set aside some samples for validation and testing.

Select Percentages

Randomly divide up the 20 samples:

| | | |
|---|---|---|
| Training: | 50% | 10 samples |
| Validation: | 35% | 7 samples |
| Testing: | 15% | 3 samples |

Explanation

Three Kinds of Samples:

Training:

These are presented to the network during training, and the network is adjusted according to its error.

Validation:

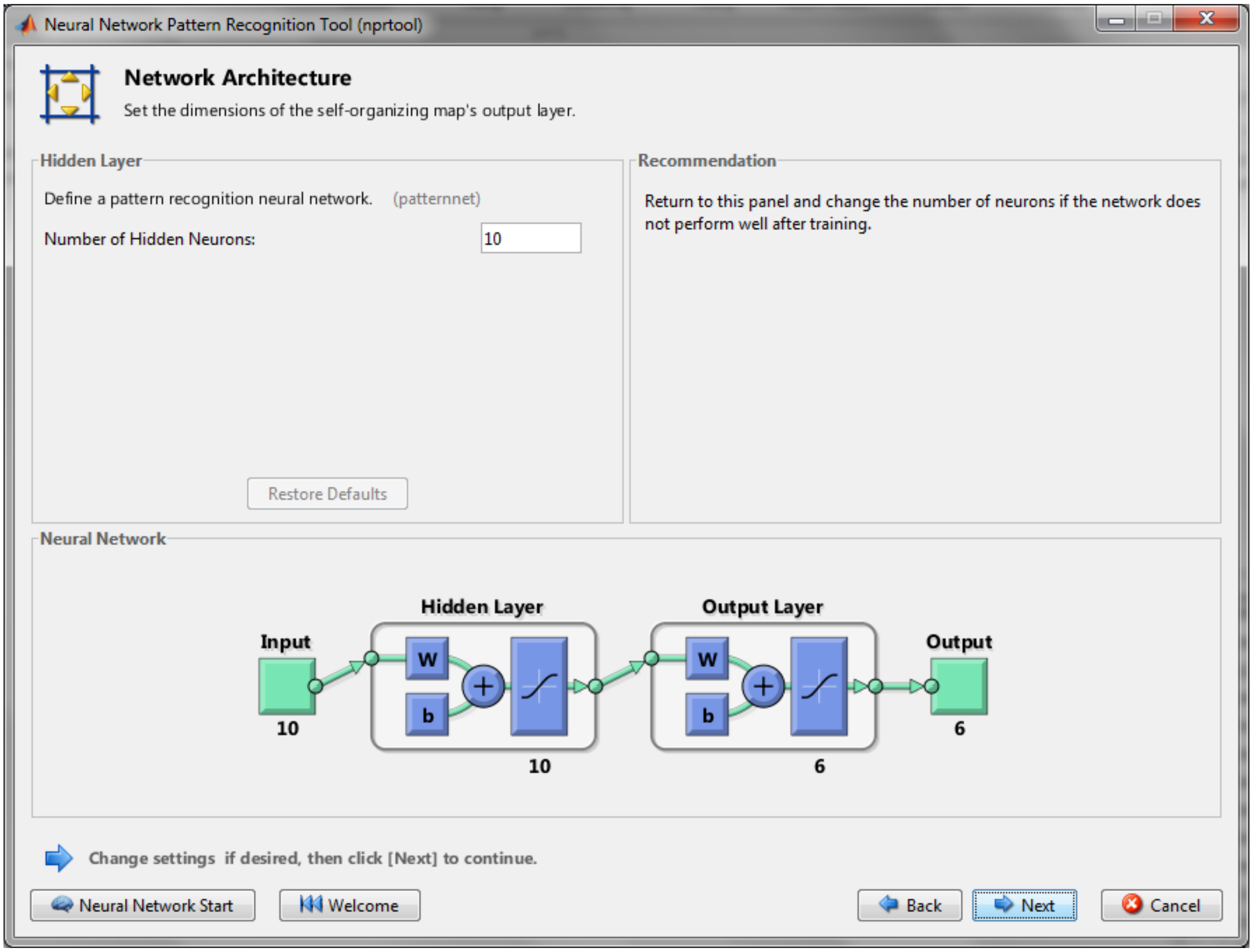
These are used to measure network generalization, and to halt training when generalization stops improving.

Testing:

These have no effect on training and so provide an independent measure of network performance during and after training.

Change percentages if desired, then click [Next] to continue.

**Figure 6: Second page of the NN toolbox allows the user to adjust percentage of samples to be used for training, validation, and testing.**



Neural Network Pattern Recognition Tool (nprtool)

**Network Architecture**

Set the dimensions of the self-organizing map's output layer.

**Hidden Layer**

Define a pattern recognition neural network. (patternnet)

Number of Hidden Neurons: 10

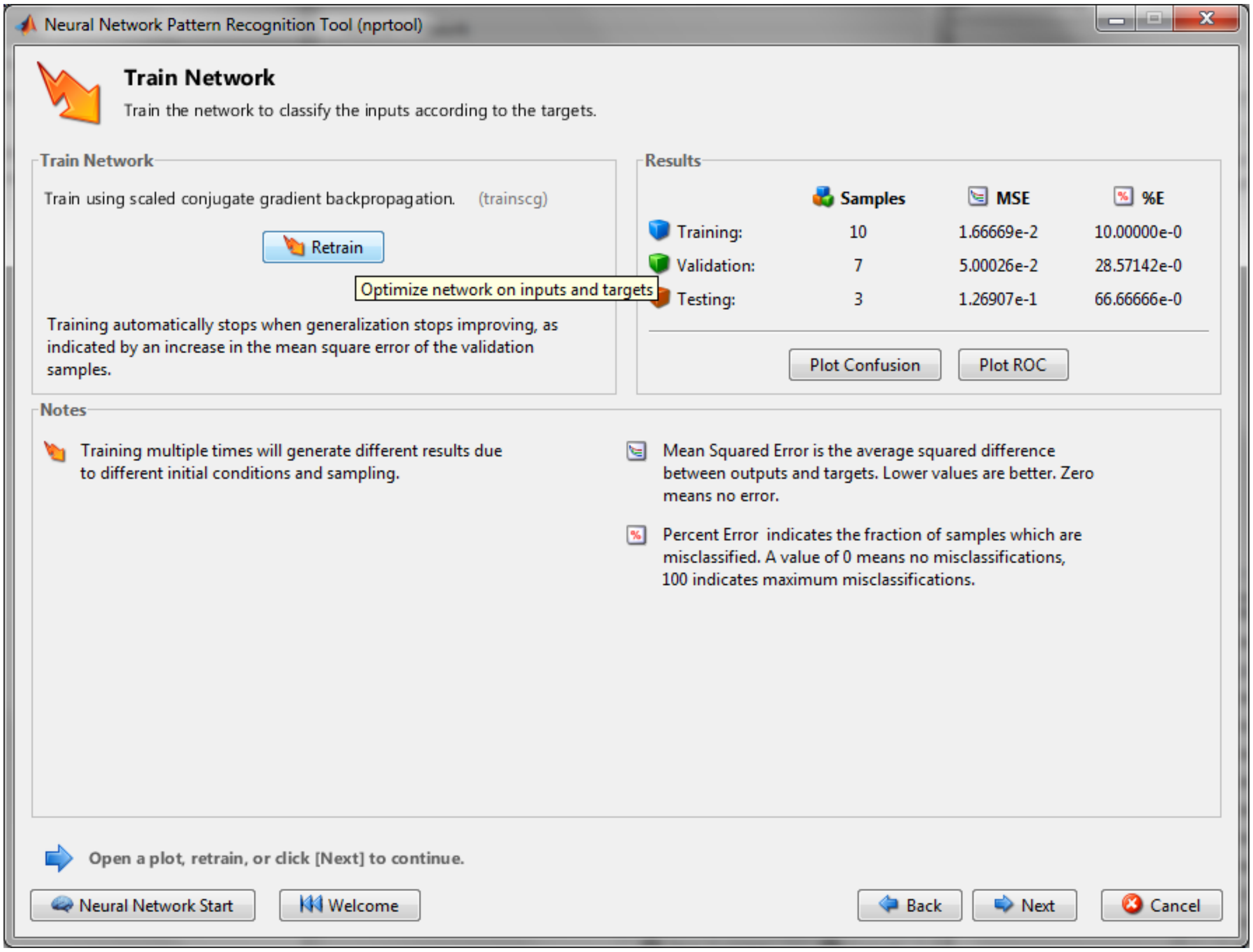
**Recommendation**

Return to this panel and change the number of neurons if the network does not perform well after training.

**Neural Network**

Change settings if desired, then click [Next] to continue.

**Figure 7: Third page of the NN toolbox allows the user to adjust NN size.**


Neural Network Pattern Recognition Tool (nprtool)
Retrain
Optimize network on inputs and targets
Plot Confusion
Plot ROC
Neural Network Start
Welcome
Back
Next
Cancel

**Train Network**

Train the network to classify the inputs according to the targets.

Train Network

Train using scaled conjugate gradient backpropagation. (trainscg)

Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

Results

| | Samples | MSE | %E |
|---|---|---|---|
| Training: | 10 | 1.66669e-2 | 10.00000e-0 |
| Validation: | 7 | 5.00026e-2 | 28.57142e-0 |
| Testing: | 3 | 1.26907e-1 | 66.66666e-0 |

Notes

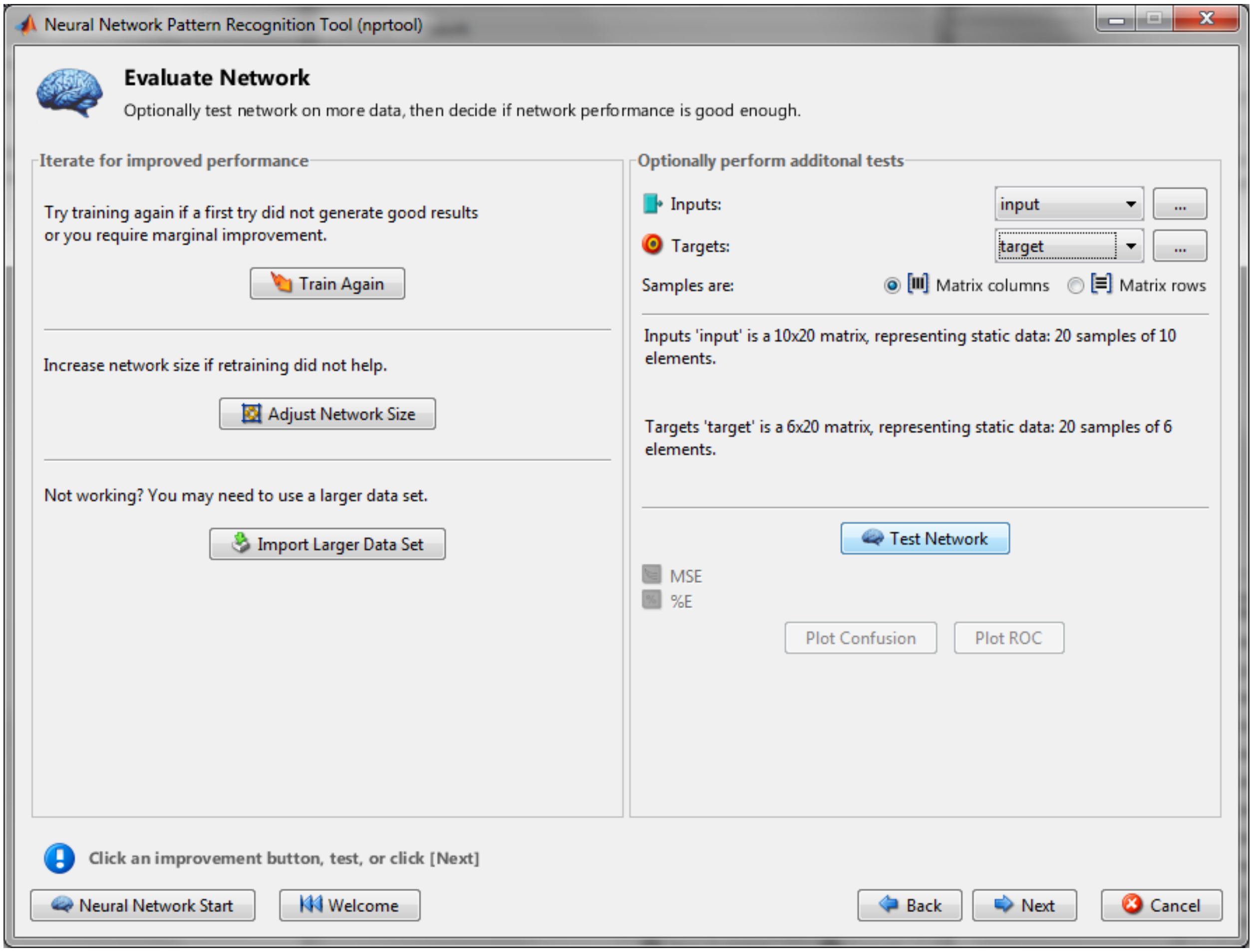
Training multiple times will generate different results due to different initial conditions and sampling.

Mean Squared Error is the average squared difference between outputs and targets. Lower values are better. Zero means no error.

Percent Error indicates the fraction of samples which are misclassified. A value of 0 means no misclassifications, 100 indicates maximum misclassifications.

Open a plot, retrain, or click [Next] to continue.

**Figure 8: Fourth page of the NN toolbox allows the user to train data, see results, and retain the network if desired.**



Neural Network Pattern Recognition Tool (nprtool)

**Evaluate Network**

Optionally test network on more data, then decide if network performance is good enough.

Iterate for improved performance

Try training again if a first try did not generate good results or you require marginal improvement.

Increase network size if retraining did not help.

Not working? You may need to use a larger data set.

Optionally perform additonal tests

Inputs: input

Targets: target

Samples are: ● Matrix columns ○ Matrix rows

Inputs 'input' is a 10x20 matrix, representing static data: 20 samples of 10 elements.

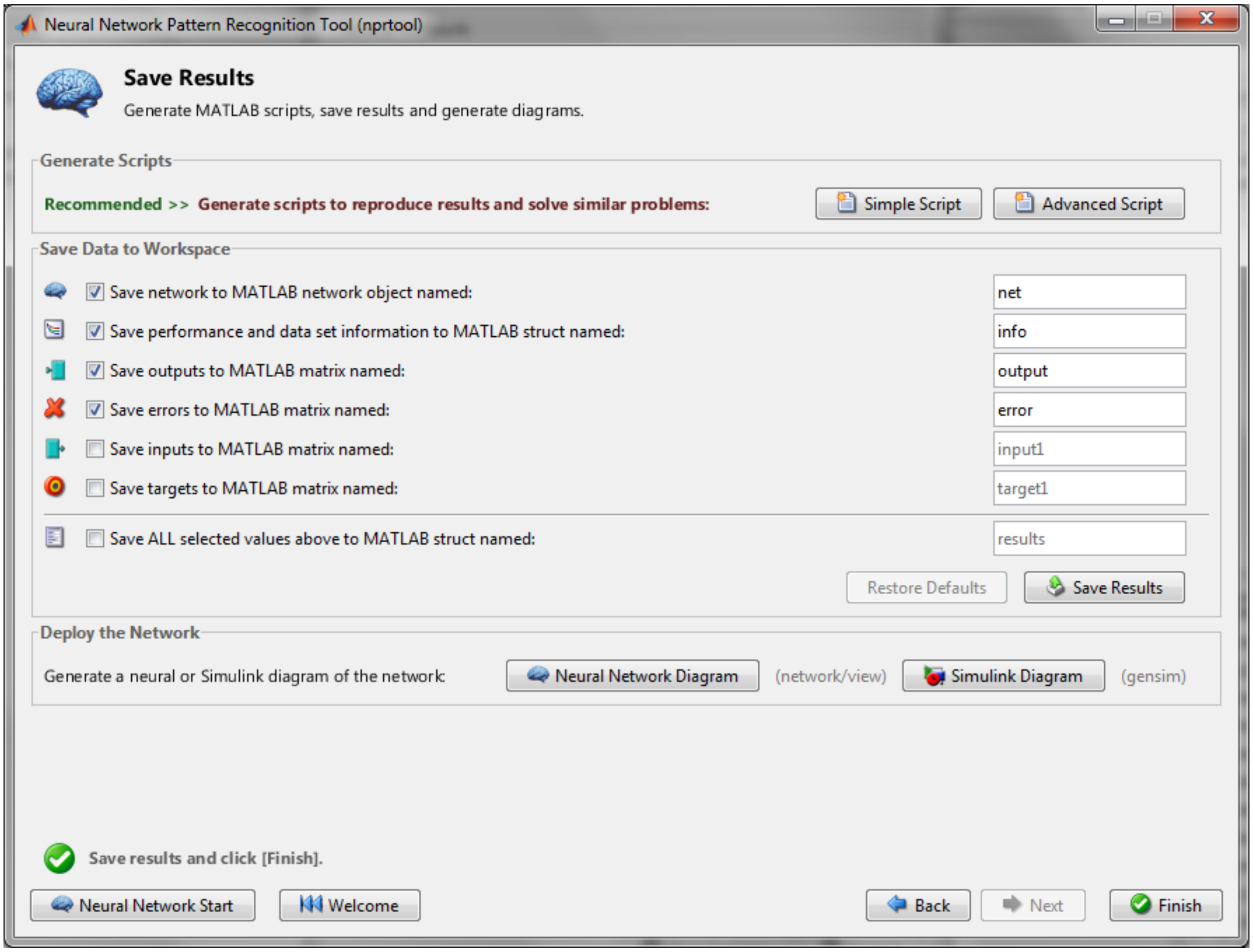
Targets 'target' is a 6x20 matrix, representing static data: 20 samples of 6 elements.

MSE

%E

Click an improvement button, test, or click [Next]

Figure 9: Fifth page of the NN toolbox allows the user to train the network again, adjust parameters, import new data sets, and perform additional testing.



Neural Network Pattern Recognition Tool (nprtool)

**Save Results**

Generate MATLAB scripts, save results and generate diagrams.

Generate Scripts

Recommended >> Generate scripts to reproduce results and solve similar problems:

Save Data to Workspace

| | |
|---|---|
| ☑ Save network to MATLAB network object named: | net |
| ☑ Save performance and data set information to MATLAB struct named: | info |
| ☑ Save outputs to MATLAB matrix named: | output |
| ☑ Save errors to MATLAB matrix named: | error |
| □ Save inputs to MATLAB matrix named: | input1 |
| □ Save targets to MATLAB matrix named: | target1 |
| □ Save ALL selected values above to MATLAB struct named: | results |

Deploy the Network

Generate a neural or Simulink diagram of the network:

Save results and click [Finish].

Figure 10: Sixth page of the NN toolbox allows the user to save results and generate scripts.

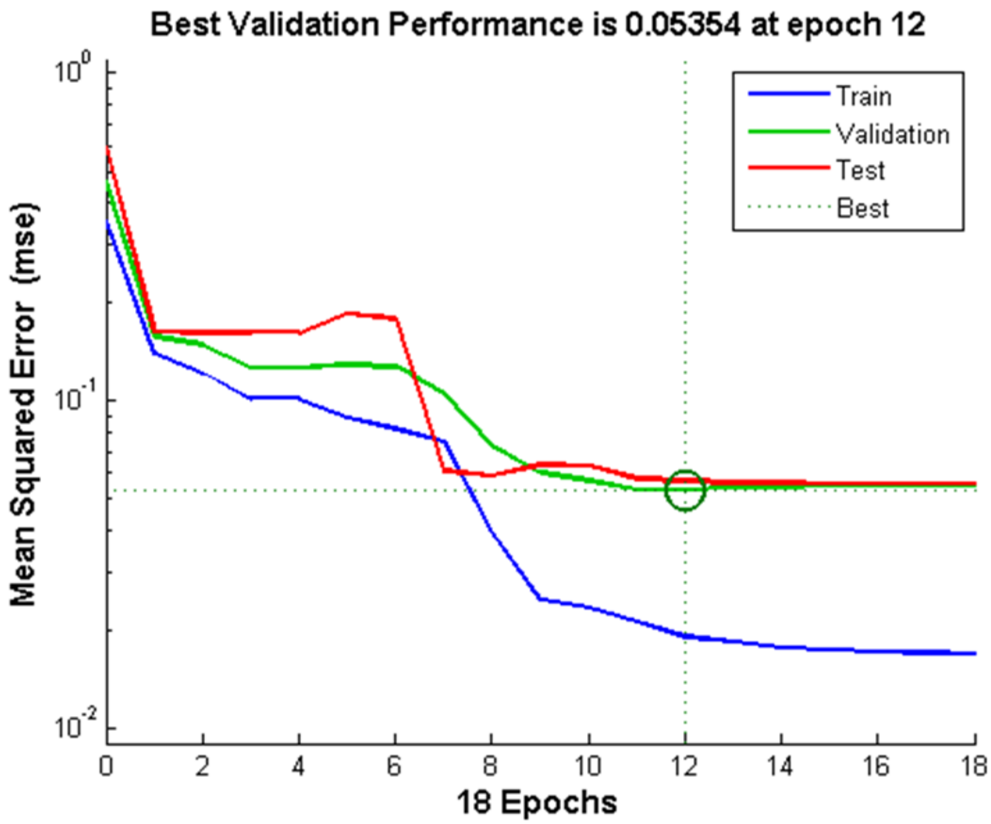


**Figure 11: This plot shows the difference between our target or expected data and the actual simulation of the neural network. The best performances occurred at epoch 12, or iteration 12.**

### QRS Complex Detection

The QRS complex is a name for the deflections of a heart's electrical activity. In ECG monitoring, each heartbeat is acquired via electrodes hooked up to the patient's body that pick up traces of the heart signals. These signals are graphically displayed with various spikes and bumps in terms of amplitude (millivolts) that yield important information about a patient's health [7]. The heart has various parts that all work together to pump blood in and out of the muscle as seen in Figure 13. Each heartbeat begins with an electrical signal derived in the sinoatrial node, or SA node, near the entrance of the superior vena cava [14]. Blood enters your right and left atriums, and when these compartments are filled, the electrical signal potential across the atriums causes them to contract, producing the P wave. The atriums contract the blood into the right and left ventricles which then also fill up with blood. The left and right ventricles are the largest parts of the heart and contract heavily with each heartbeat so that the potential difference between them is large and easily visible as the QRS complex. This is due to the depolarization of left and right ventricles. Once contracted, the right ventricle pumps blood into the pulmonary artery to your lungs while the left ventricle pumps blood through the aortic valve to the rest of the body. As the electrical signal passes along, the heart's ventricles relax resulting in the T wave [14].

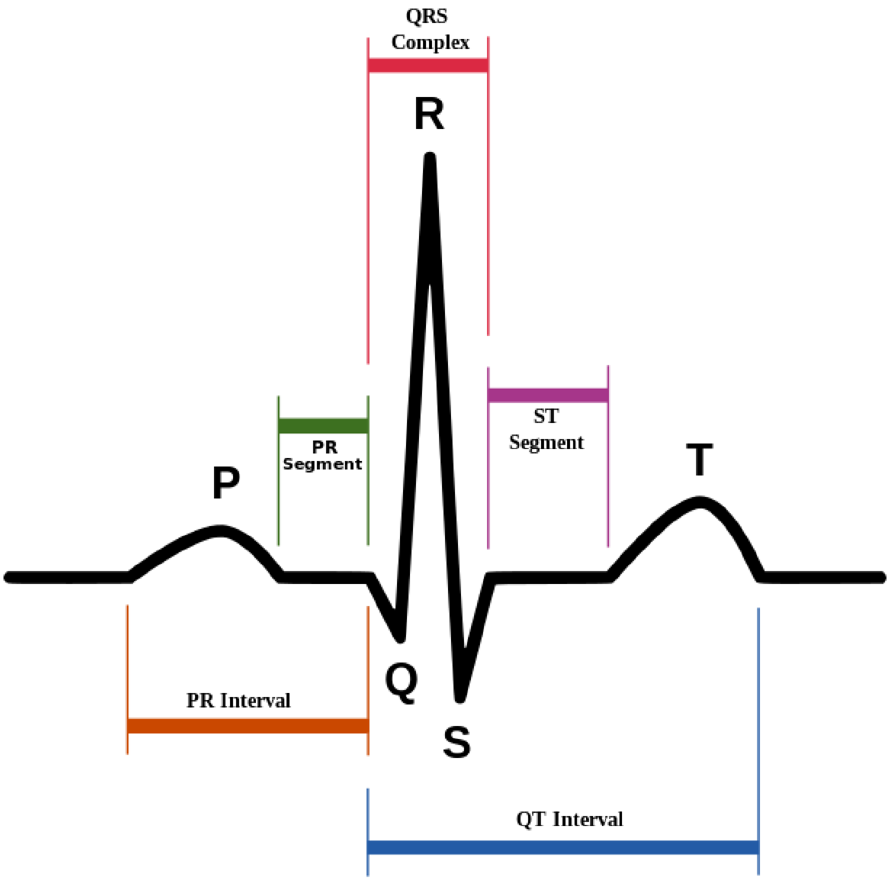


**Figure 12: Example ECG waveform with intervals labeled [18].**

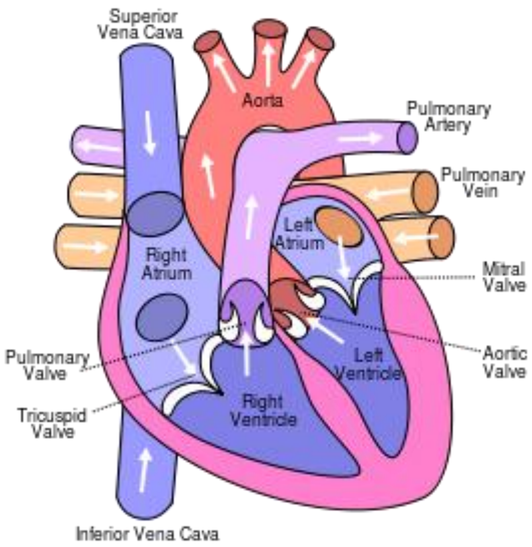


Figure 13: Diagram of blood flow in the heart [19].

There are nominal ranges for the PR and QT intervals as well as RR intervals from preceding or proceeding heartbeats to determine whether a person's heart is normal or irregular. For a normal heartbeat, the QRS duration is in the range of 0.6-0.10 sec, the PR interval is usually 0.12-0.20 sec, and the QT interval is under 0.4 sec [7]. All these numbers are based on a person's heartbeat, and since a normal heartbeat is in the range of 60-100 beats per minute, each person's nominal values will be different from another person. Differences in age, gender, stress, and illness can also affect a person's heart rate. We will simulate several ECG signals from the MIT-BIH Arrhythmia Database to acquire each person's QRS points as well as relevant PR, QT, and RR intervals and validate whether a person has a heart problem by running their ECG characteristics through an artificial neural network.

## QRS Detection

Our QRS detection is mainly done through use of the derivative. We look for the maximum and minimum of the derivative of a wave. The highest peak (R wave) should be the zero crossing between the maximum and minimum of the derivative. Likewise, the Q point should be at the zero crossing before the maximum and the S point should be at the zero crossing after the minimum. The P and T waves are done similarly by looking for local maximums in the original waveform and then using the derivative to identify peak and end points.

However, not all waves were ideal, and we had to take into account different conditions where the ECG signal might not be correctly identified by the previous algorithm. To do this, we compared values such as the median and minimum and if the minimum was very low such as in the waves for LBBB or PVC; we looked for if the minimum was before or after the maximum (R wave) and ran a similar algorithm to the one above that takes into account the unique structure of heartbeats with different conditions.

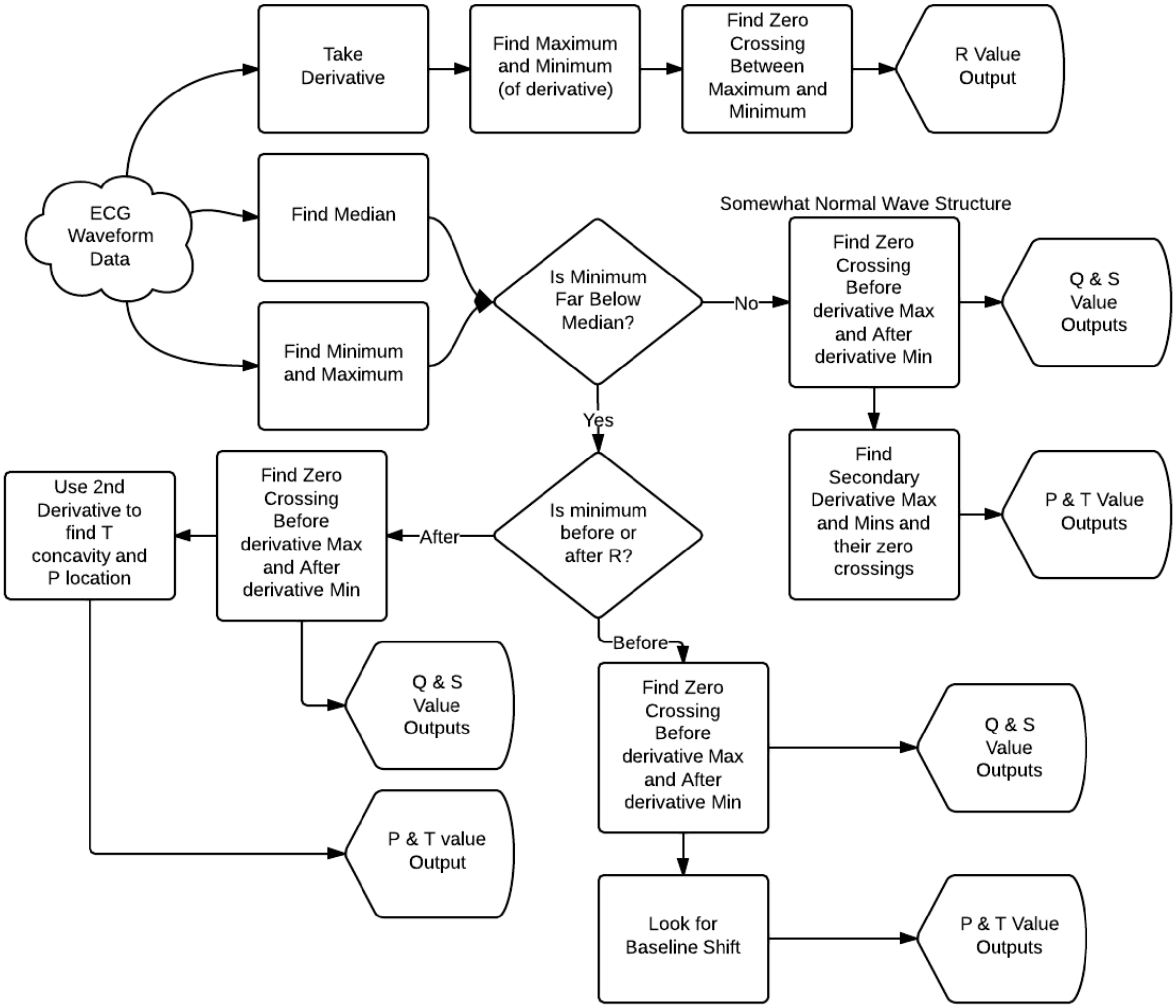

Take Derivative
Find Maximum and Minimum (of derivative)
Find Zero Crossing Between Maximum and Minimum
R Value Output
ECG Waveform Data
Find Median
Find Minimum and Maximum
Somewhat Normal Wave Structure
Is Minimum Far Below Median?
No
Find Zero Crossing Before derivative Max and After derivative Min
Q & S Value Outputs
Yes
Find Secondary Derivative Max and Mins and their zero crossings
P & T Value Outputs
Is minimum before or after R?
After
Find Zero Crossing Before derivative Max and After derivative Min
Use 2nd Derivative to find T concavity and P location
Q & S Value Outputs
P & T value Output
Before
Find Zero Crossing Before derivative Max and After derivative Min
Q & S Value Outputs
Look for Baseline Shift
P & T Value Outputs

Figure 14: QRS detection program flowchart.

To be able to classify each cardiac condition, we needed to identify the following ten characteristics of each signal: [10] this is done outside of the neural network using the QRS detection method described above and shown in figure 14. Each of the following ten characteristics is an input to the neural network given as a raw number.

1) RS Interval
2) PR Interval
3) QRS Interval
4) QT Interval
5) R-R with previous signal
6) R-R with subsequent signal
7) Q wave amplitude
8) R wave amplitude
9) S wave amplitude
10) T wave concavity

We first filtered the signal to remove artifacts such as power line interference using a zero phase Butterworth filter with a cutoff frequency of 30 Hz. The transfer function of the filter was:

$$H(s) = \frac{B(s)}{A(s)} =$$

$$\frac{3.49*10^{-7}s^{10}+3.49*10^{-6}s^{9}+1.57*10^{-5}s^{8}+4.19*10^{-5}s^{7}+7.33*10^{-5}s^{6}+8.79*10^{-5}s^{5}+7.33*10^{-5}s^{4}+4.19*10^{-5}s^{3}+1.57*10^{-5}s^{2}+3.49*10^{-6}s+3.49*10^{-7}}{s^{10}-6.655s^{9}+20.336s^{8}-37.467s^{7}+45.996s^{6}-39.251s^{5}+23.549s^{4}-9.798s^{3}+2.703s^{2}-0.446s+0.033}$$

Next, we developed a script in MATLAB to identify each relevant data point (P, Q, R, S, and T) using derivatives and thresholds in both the waveform and its derivative [10].

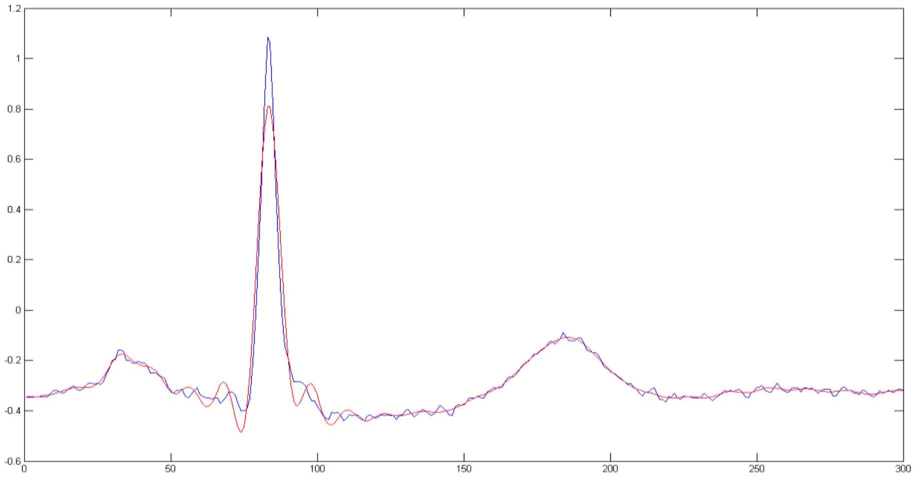
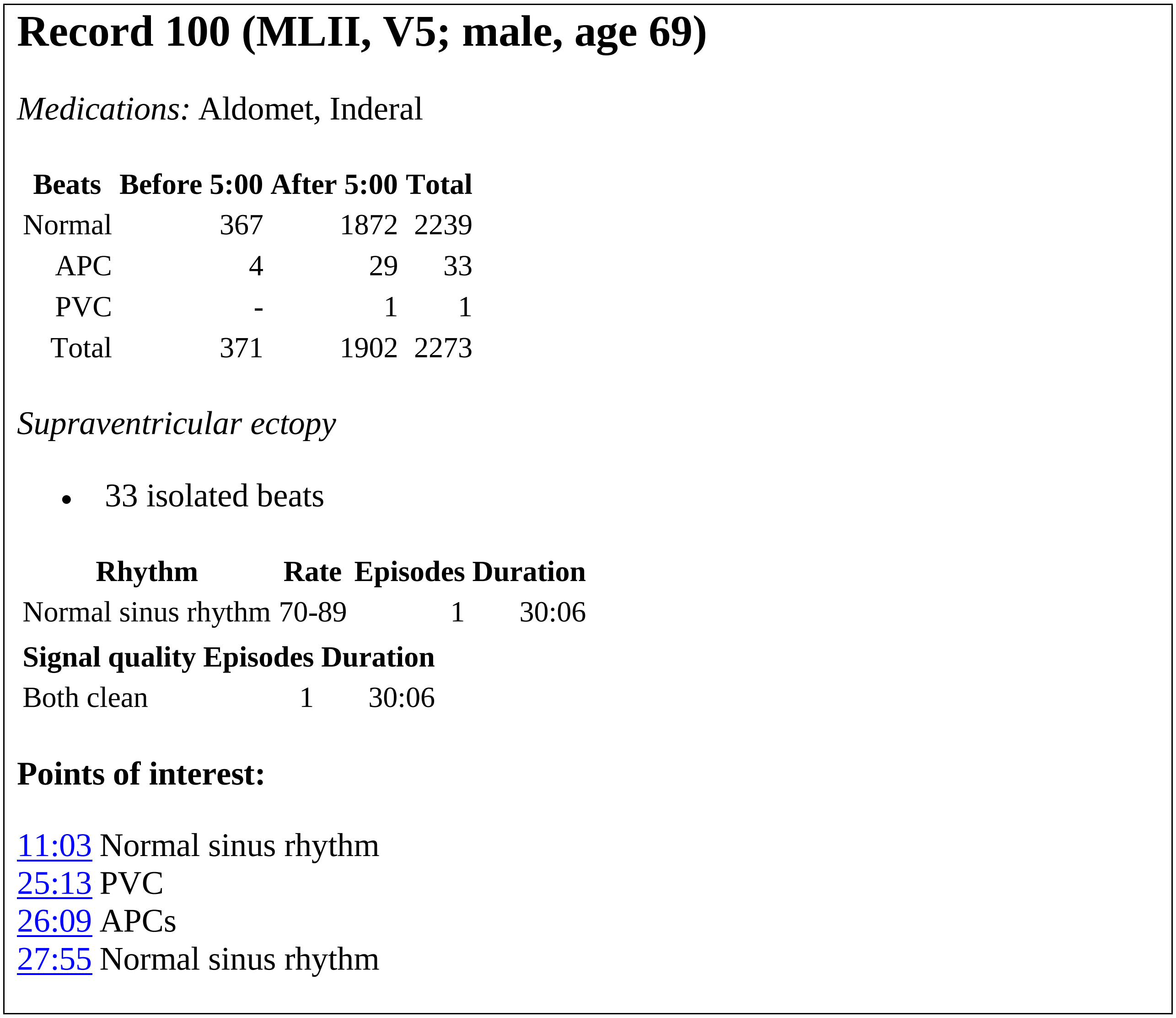


**Figure 15: The normal ECG signal is in blue and the red wave is produced after applying the Butterworth zero-phase filter.**

The MIT-BIH database gives points of interest for each ECG signal with the time and a description of the feature present (e.g. RBBB). We simply looked up the waveforms at the given times and fed the waveform into our QRS point detection program. To the untrained eye, it is difficult to determine exactly where an irregular heart beat is and which class of diseases it belongs to so although this information was very helpful we had to use several references to confirm the class of the signal. An example of the patient record with points of interest is as follows:



# Record 100 (MLII, V5; male, age 69)

*Medications:* Aldomet, Inderal

| Beats | Before 5:00 | After 5:00 | Total |
|---|---|---|---|
| Normal | 367 | 1872 | 2239 |
| APC | 4 | 29 | 33 |
| PVC | - | 1 | 1 |
| Total | 371 | 1902 | 2273 |

*Supraventricular ectopy*

- 33 isolated beats

| Rhythm | Rate | Episodes | Duration |
|---|---|---|---|
| Normal sinus rhythm | 70-89 | 1 | 30:06 |

| Signal quality | Episodes | Duration |
|---|---|---|
| Both clean | 1 | 30:06 |

**Points of interest:**

11:03 Normal sinus rhythm
25:13 PVC
26:09 APCs
27:55 Normal sinus rhythm

Figure 16: Record for patient 100. We used this patient to extract normal sinus rhythm, PVC, and APC heartbeats.

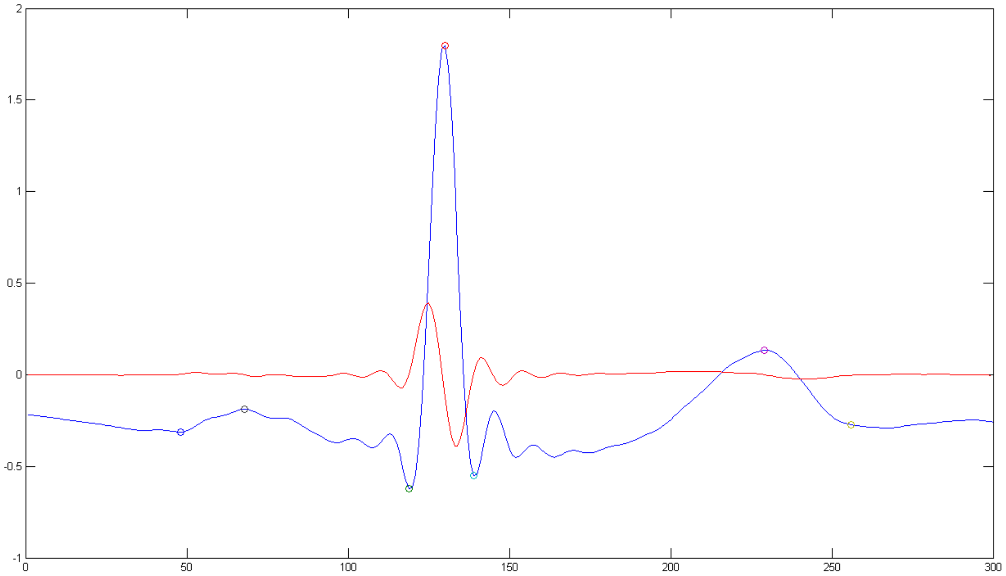


Figure 17: Detection of characteristic points with ECG in blue and its derivative in red.

## Cardiac Conditions

We wanted be able to detect 6 different classes of cardiac conditions: [10]

1) Normal beat
2) Left Bundle Branch Block beat (LBBB)
3) Right Bundle Branch Block beat (RBBB)
4) Atrial Premature Contraction beat (APC)
5) Premature Ventricular Contraction beat (PVC)
6) Paced beat

On the following pages are descriptions, symptoms, and pictures for each class.

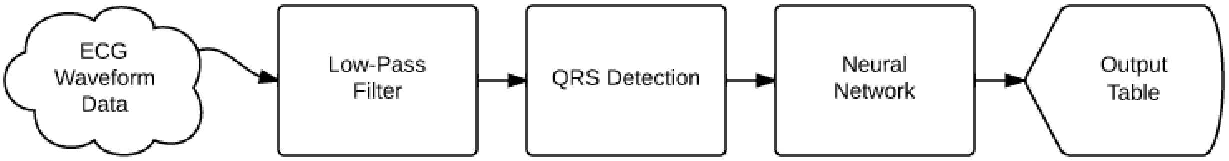


Figure 18: Flow chart showing the process of determining the class of an ECG signal.

## Normal Heartbeat:

Normal heartbeats are characterized by the following: [12]

1) QRS duration between 60 ms and 100 ms
2) R-R interval between 600 ms and 1200 ms
3) PR interval between 120 ms and 200 ms
4) QT interval up to 420 ms
5) Q amplitude less than ¼ of R amplitude

An example can be seen in figure 19 below:

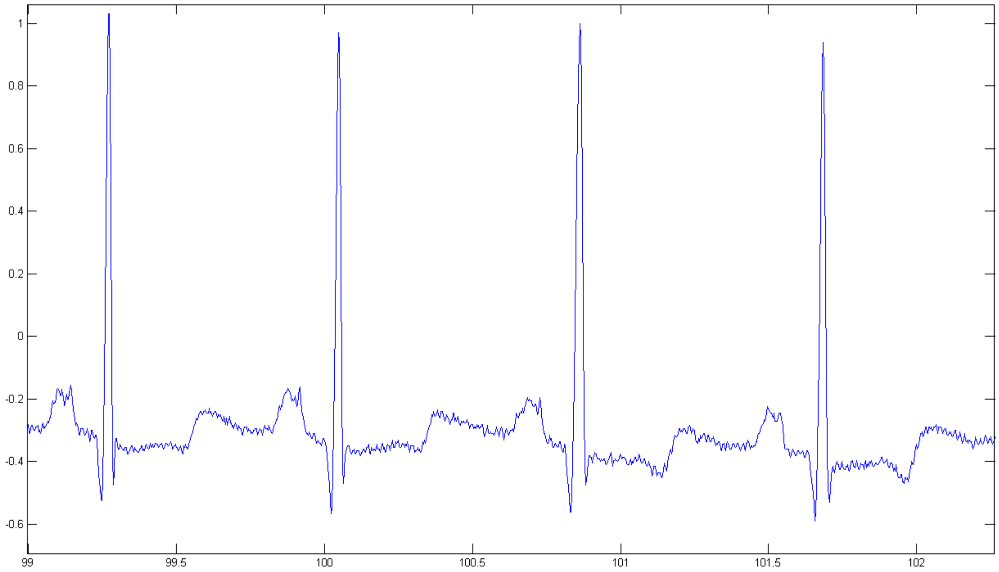


**Figure 19: Example of normal sinus rhythm from patient 100 of MTI-BIH Arrhythmia database.**

## Left Bundle Branch Block (LBBB):

LBBB is a condition where the left ventricle contracts later than the right ventricle due to delayed activation of the left ventricle. It is primarily diagnosed by the following characteristics: [12]

1) Widened QRS with duration greater than 120 ms
2) ST wave is deflected opposite of the QRS complex.

An example can be seen in figure 20 below:

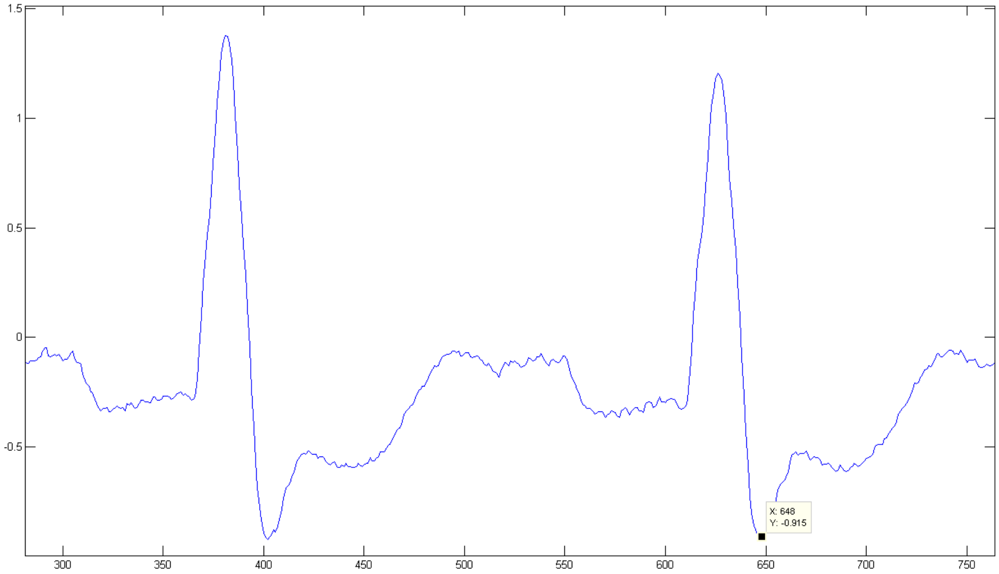


**Figure 20: Example of LBBB from patient 109 of MTI-BIH Arrhythmia database.**

### Right Bundle Branch Block (RBBB):

RBBB is a condition where impulses travelling through the right bundle branch do not directly activate the right ventricle. The left bundle branch, however, still activates the left ventricle normally. It is primarily diagnosed by the following characteristics: [12] [11]

1) Widened QRS with duration greater than 120 ms
2) Slurred S waves
3) The QRS complex sometimes shows an extra deflection

An example can be seen in figure 21 below:

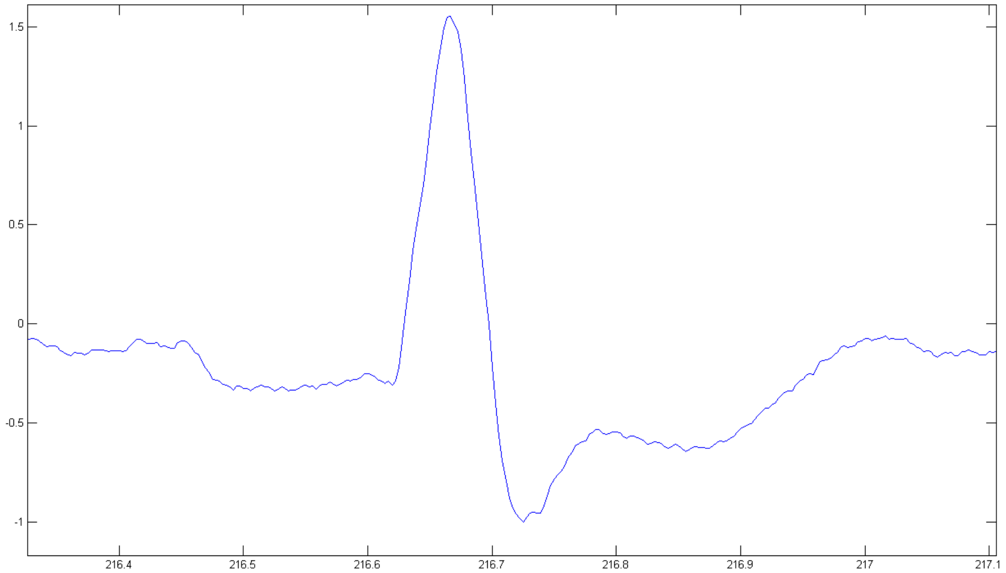


Figure 21: Example of RBBB from patient 118 of MTI-BIH Arrhythmia database.

**Premature Ventricular Contractions (PVC):**

PVC is a common condition where the ventricles contract before the atria optimally fill the ventricles with blood because the Purkinje fibres in the ventricles initiate the heartbeat rather than the sinoatrial node, the normal heartbeat initiator. Single beat PVC arrhythmias are usually nonthreatening. PVCs are primarily diagnosed by the following characteristics: [12]

1) Widened QRS with duration greater than 120 ms
2) Different than the normal QRS morphology
3) Premature, with a compensatory pulse

An example can be seen in figure 22 below:

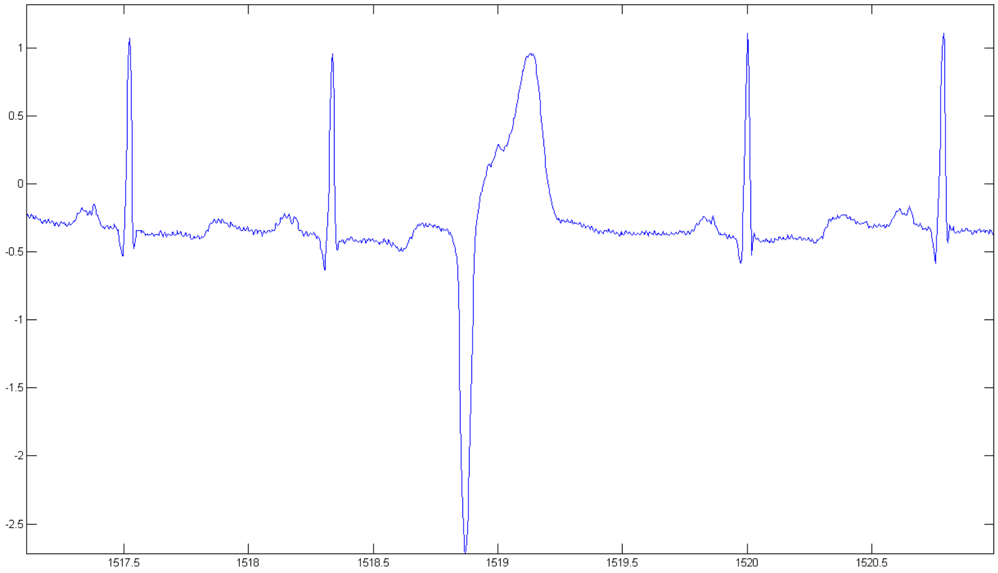


**Figure 22: Example of PVC from patient 102 of MTI-BIH Arrhythmia database.**

### Atrial Premature Complex (APC):

APC is a condition in which premature heartbeats originating in the atria occur due to another region of the atria depolarizing before the sinoatrial node, the normal heartbeat initiator. APCs are primarily diagnosed by the following characteristics: [12]

1) Premature, with a shortened R-R with the previous beat
2) Compensatory pause, with a lengthened R-R with the subsequent beat
3) Narrow QRS duration

An example can be seen in figure 23 below:

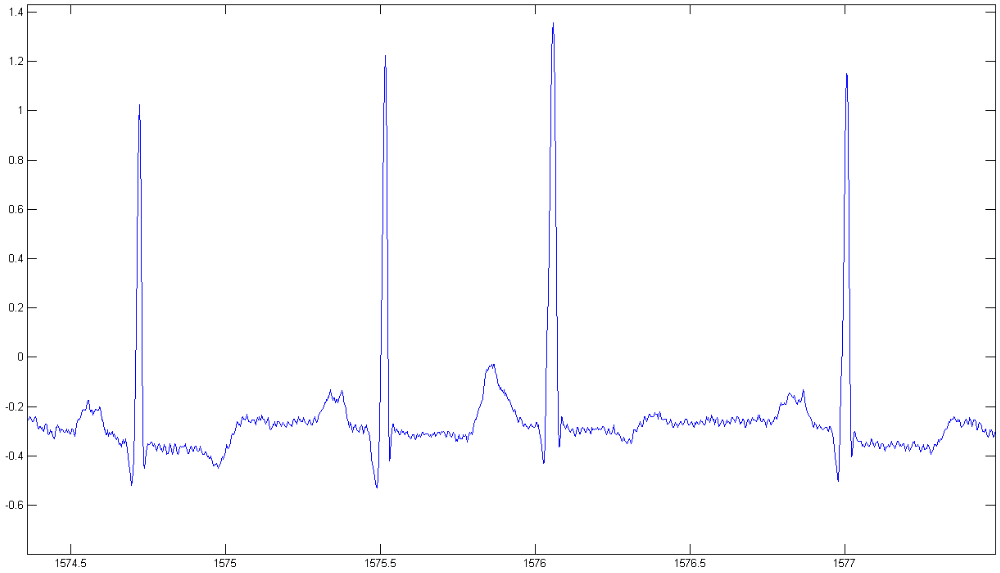


Figure 23: Example of APC from patient 100 of MTI-BIH Arrhythmia database.

**Paced Beat:**

A paced beat is simply the result of a patient with an artificial pacemaker, which is a device that contracts the heart muscles using electrical pulses to help regulate the beating of the heart. Paced beats are characterized by: [13]

1) Spikes representing electrical pulse of pacemaker (before either the P or Q wave)
2) QRS complex that is wide, bizarre, and resembles a ventricular beat

An example can be seen in figure 24 below:

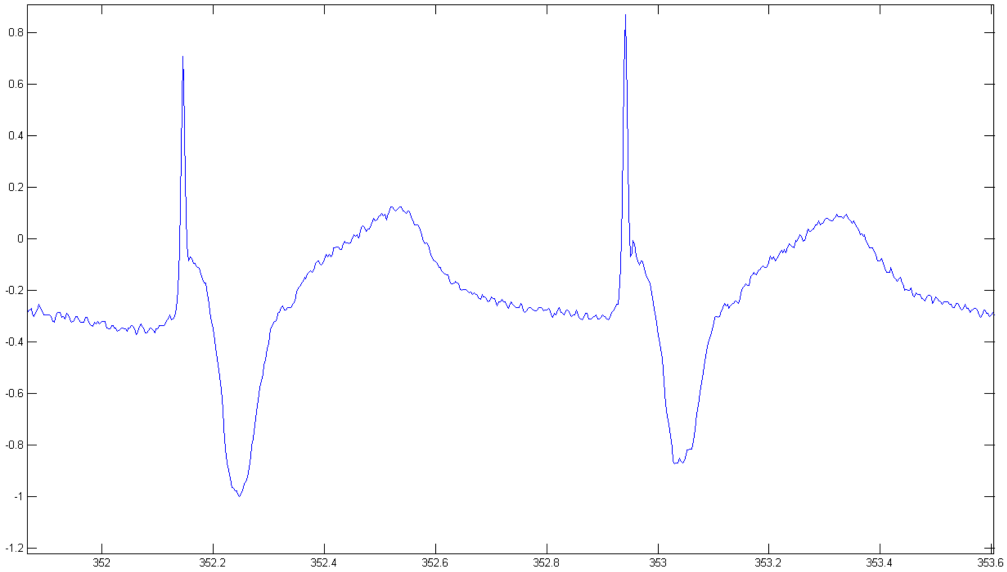
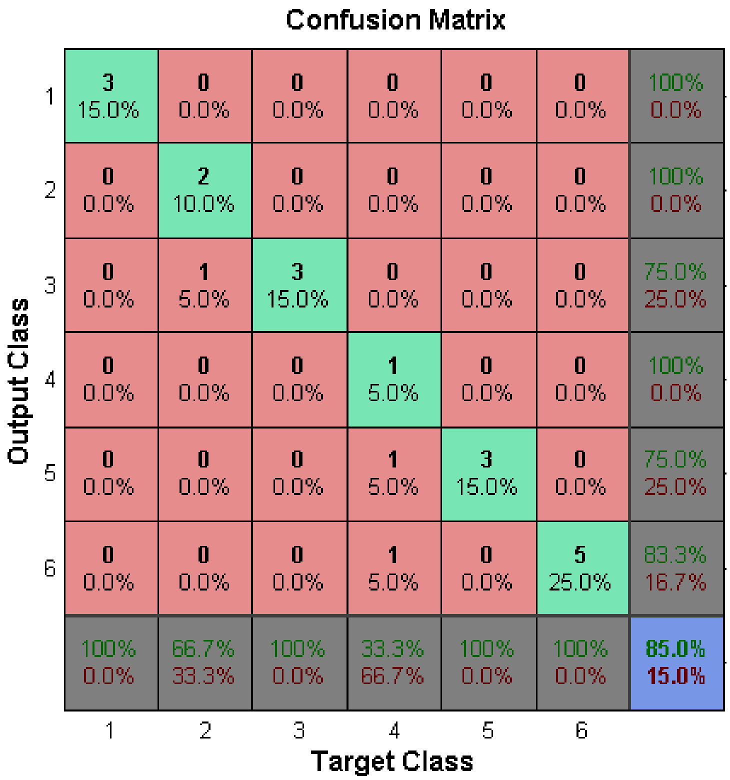


**Figure 24: Example of a paced beat from patient 104 of MTI-BIH Arrhythmia database.**

## Results

As seen in figure 25 below, our neural network was able to correctly identify 85% of the waveforms it was given. It correctly identified 100% of all PVC, RBBB, Paced, and Normal beats. It correctly identified 66.7% of all APC beats and 33.3% of all LBBB beats. Although the APC beat identification rate was far below what we hoped to achieve, our overall rate of identification was still very good.



| | 1 | 2 | 3 | 4 | 5 | 6 | |
|---|---|---|---|---|---|---|---|
| 1 | 3<br>15.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 100%<br>0.0% |
| 2 | 0<br>0.0% | 2<br>10.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 100%<br>0.0% |
| 3 | 0<br>0.0% | 1<br>5.0% | 3<br>15.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 75.0%<br>25.0% |
| 4 | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 1<br>5.0% | 0<br>0.0% | 0<br>0.0% | 100%<br>0.0% |
| 5 | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 1<br>5.0% | 3<br>15.0% | 0<br>0.0% | 75.0%<br>25.0% |
| 6 | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 1<br>5.0% | 0<br>0.0% | 5<br>25.0% | 83.3%<br>16.7% |
| | 100%<br>0.0% | 66.7%<br>33.3% | 100%<br>0.0% | 33.3%<br>66.7% | 100%<br>0.0% | 100%<br>0.0% | 85.0%<br>15.0% |

**Figure 25: Confusion matrix, showing the number of each class of inputs correctly and incorrectly identified. There are 20 samples and for this iteration in the neural network, there were 3 samples that were determined incorrectly, as seen in the red squares. The total number of samples determined correctly is the 85%, or 17 samples, and the rest is the 15%, or 3 samples.**

The error, as seen in figure 26, follows a narrow bell curve with very little significant error. This agrees with the results in the confusion matrix. Because our target data is binary (a wave can only either belong or not belong to a class), only significant errors where the output is closer to 1 when it should be 0 or closer to 0 when it should be 1 make a difference.

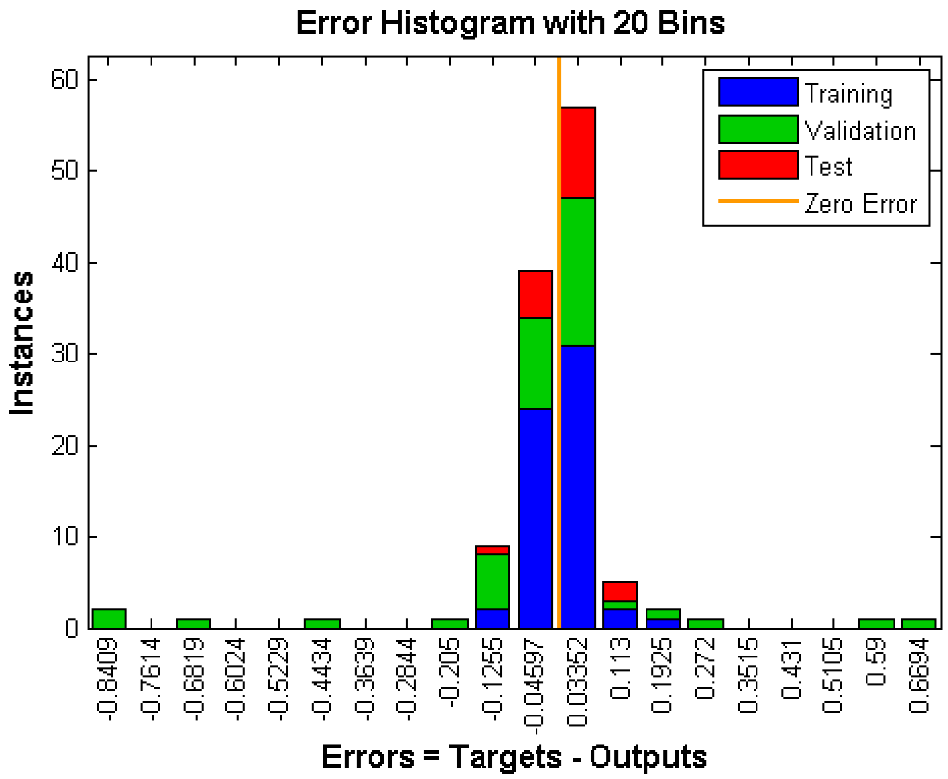


**Figure 26: Error histogram showing how outputs deviated from target.**

**Table 2: Statistics table showing the training and testing performance of the Neural Network with 17 sets of training data and 20 sets of testing data**

| | Max | Min | Mean | Standard Dev. |
|---|---|---|---|---|
| MSE (Training) | 0.1218 | 0.0037 | 0.044375 | 0.033427 |
| MSE (Test) | 0.193082102 | 1.54246E-07 | 0.029769 | 0.067138 |

Our results have both advantages and disadvantages when compared to the results of the Intelligent Heart Disease Recognition using Neural Networks [10]. We had an average MSE of 0.02977 compared to 0.00484. Although our MSE was larger, in some categories, we had higher recognition rates. This can be seen in table 3 below.

Table 3: Comparison of our NN's recognition rate to that of another report on ECG detection.

| | Normal | APC | PVC | RBBB | LBBB | Paced |
|---|---|---|---|---|---|---|
| Their Recognition Rate (%) | 88 | 96 | 91 | 98 | 84 | 87 |
| Our Recognition Rate (%) | 100 | 67 | 100 | 100 | 33 | 100 |

## Conclusion

We were able to use an artificial neural network to classify different heartbeat waveforms into 1 of 6 different categories. Our neural network was able to achieve an 85% correct identification rate. With such a high rate of identification, it could easily be used as a preliminary analysis before a cardiologist is consulted to save time and money. It could also be used to analyze ECG signals in remote places where people do not have easy access to a professional doctor.

Although our results proved efficient, the ANN could be improved even more with better accuracy if more inputs and associated target values were used. Unfortunately, the database has a limited number of ECG signals with diseases that are not separate from others in the signal. For example, although several PVC beats may be graphically displayed, they are also associated with another disease such as ventricular couplets which makes it very difficult to distinguish where the exact PVC wave is to the naked eye unless you are a licensed doctor. In the future, this project could be improved by acquiring more similar input signals so that the ANN can better classify the heart signals as diseased or normal. Once many more inputs are used and the ANN provides steady, high accuracy (98% or better) results, an interface can be developed to test on real patients and deliver real time results to them about their heart condition and whether they need further diagnosis. In addition, our simulations assumed that there was only one heart condition present at any given time. In the future, work could be done to account for heartbeats where symptoms of more than one condition are present.

## References

[1] Burch, George and DePasquale, Nicholas. “A History of Electrocardiography.” 8 October 2012. <http://rmccrory.tripod.com/historyekg.html>. October 2012.

[2] Galkin, Ivan. "Crash Introduction to Artificial Neural Networks." Crash Introduction to Artificial Neural Networks. U. MASS Lowell, n.d. Web. November 2012. <http://ulcar.uml.edu/~iag/CS/Intro-to-ANN.html>.

[3] Hassoun, Mohamad H. *Fundamentals of Artificial Neural Networks*. Cambridge, MA: MIT, 1995. Print. November 2012.

[4] Haykin, Simon S. Neural networks: a comprehensive foundation. 2nd ed. Upper Saddle River, N.J.: Prentice Hall, 1999. November 2012.

[5] “Introduction to 12 Lead ECGs.” Mississippi State Department of Health. n.d. 10 Nov. 2012. <http://msdh.ms.gov/msdhsite/_static/resources/2889.pdf>.

[6] Treleaven, P, M Pacheco, and M Vellasco. "VLSI Architectures for Neural Networks." IEEE Micro, 9.6 (1989): 8-27. 20 Nov. 2012.

[7] “Wikipedia contributors. "QRS complex." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 13 Apr. 2013. Web. February 2013.

[8] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng CK, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* 101(23):e215-e220 [Circulation Electronic Pages; http://circ.ahajournals.org/cgi/content/full/101/23/e215]; 2000 (June 13). Web. March 2013.

[9] “Recognizing Patterns”. MathWorks. Documentation Center. n.d. Web. 15 March, 2013. <http://www.mathworks.com/help/nnet/gs/recognizing-patterns.html>.

[10] Chowdhry, Dr. Bhawani Shanker, Baqai, Attiya, Bhagwandas. “Intelligent Heart Disease Recognition using Neural Networks.” 24th Annual Multi-topic International Symposium IEEEP Karachi 2009. March 15, 2013.

[11] Bouthillet, Tom. “Terminal-r Wave”. EMS 12-Lead. 19, June 2009. <http://ems12lead.com/tag/terminal-r-wave/>. Web. 20, March 2013.

[12] Lome, Steve. "ECG Review and Criteria." Learn the Heart. 30, March 2013. <http://www.learntheheart.com/PACs.html>.

[13] Fogorous, Richard N. "Pacemakers – What You Should Know." 10, November 2011. About.com, Hearth Health Center. Web. 2, April 2013. <http://heartdisease.about.com/cs/arrhythmias/a/pacemakers.htm>.

[14] "Your Heart's Electrical System." National Heart, Lung, and Blood Institute. How the Heart Works. 17, November 2011. Web. 20, April 2013. <http://www.nhlbi.nih.gov/health/health-topics/topics/hhw/electrical.html>.

[15] <http://upload.wikimedia.org/wikipedia/commons/c/c1/ECGcolor.svg>.

[16] <http://biomed.brown.edu/Courses/BI108/BI108_2001_Groups/Nerve_Regeneration/Introduction/neurne.gif>.

[17] <http://ulcar.uml.edu/~iag/CS/A-Neuron.gif>.

[18] <http://en.wikipedia.org/wiki/File:SinusRhythmLabels.svg>.

[19] <https://en.wikipedia.org/wiki/File:Diagram_of_the_humanheart(cropped).svg.>

## APPENDIX A – Butterworth Zero-Phase Filter

```
fNorm = 30 / (360/2); %30Hz cutoff frequency, 360Hz sample rate
[b, a] = butter(10, fNorm, 'low'); %generates some vectors, b &
a for the filter
Y = filtfilt(b, a, waven2); %filter the data vector data and
return Y
```

## APPENDIX B – Code for PQRST Detection

```
signal = diff(Y)

%find Threshold
Top = max(signal)
Bottom = min(signal)

%find max & min point
NumSamples = length(signal)
for(i=1:NumSamples)
    if(signal(i) == Top)
        TopLocation = i
    elseif(signal(i) == Bottom)
        BottomLocation = i
    end
end

%find Q
j=TopLocation
    while(signal(j)>.05*Top)
        j=j-1
    end
Q=j;

%find R
k=TopLocation
    while(signal(k)>.05*Top)
        k=k+1
    end
R=k;

%find S
l=BottomLocation
    while(signal(l)<.05*Bottom)
        l=l+1
    end
S=l;

%find T peak
Tp=S
peakT=Y(S)
for m=S:NumSamples,
    if Y(m) > peakT
        Tp = m
        peakT = Y(m)
    end
end

%Find T end
p=Tp+7
while(signal(p)<-.0035)
        p=p+1
    end
Te=p;

%find P peak
Pp=Q
peakP=Y(Q)
for n=Q:-1:40,
    if Y(n) > peakP
        Pp = n
        peakP = Y(n)
    end
end

%find P start
Ps=Pp
startP=Y(Pp)
for o=Pp:-1:2,
    if Y(o) < startP
        Ps = o
        startP = Y(o)
    end
end

plot(Y)
hold all
plot(Q,Y(Q),'o',R,Y(R),'o',S,Y(S),'o',Tp, Y(Tp),'o',Te,
Y(Te),'o',Pp, Y(Pp),'o', Ps, Y(Ps),'o' )
plot(signal,'r')
hold off
```

## APPENDIX C – Code for Neural Network

```
% Solve a Pattern Recognition Problem with a Neural Network
% Script generated by NPRTOOL
% Created Thu May 23 12:21:57 PDT 2013
%
% This script assumes these variables are defined:
%
%   input - input data.
%   target - target data.

inputs = input;
targets = target;

% Create a Pattern Recognition Network
hiddenLayerSize = 100;
net = patternnet(hiddenLayerSize);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
net.outputs{2}.processFcns = {'removeconstantrows','mapminmax'};


% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 50/100;
net.divideParam.valRatio = 35/100;
net.divideParam.testRatio = 15/100;

% For help on training function 'trainscg' type: help trainscg
% For a list of all training functions type: help nntrain
net.trainFcn = 'trainscg';  % Scaled conjugate gradient

% Choose a Performance Function
% For a list of all performance functions type: help
nnperformance
net.performFcn = 'mse';  % Mean squared error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist',
...
  'plotregression', 'plotfit'};
```

```
% Train the Network
[net,tr] = train(net,inputs,targets);

% Test the Network
outputs = net(inputs);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs)

% Recalculate Training, Validation and Test Performance
trainTargets = targets .* tr.trainMask{1};
valTargets = targets  .* tr.valMask{1};
testTargets = targets  .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,outputs)
valPerformance = perform(net,valTargets,outputs)
testPerformance = perform(net,testTargets,outputs)

% View the Network
%view(net)

% Plots
% Uncomment these lines to enable various plots.
figure, plotperform(tr)
%figure, plottrainstate(tr)
figure, plotconfusion(targets,outputs)
%figure, plotroc(targets,outputs)
%figure, ploterrhist(errors)
%mse(net,target,outputs)
```

## APPENDIX D – Input, Target, and Output Tables for 20 Different Simulations

Input Table

| | RS | PR | QRS | QT | R-R i | R-R s | Q amp | R amp | S amp | T |
|---|---|---|---|---|---|---|---|---|---|---|
| Simulation 1 | 0.117 | 0.000 | 0.378 | 0.000 | 0.772 | 0.861 | -2.715 | 0.960 | -0.280 | 0 |
| Simulation 2 | 0.133 | 0.000 | 0.333 | 0.000 | 0.728 | 0.667 | -1.400 | 0.190 | -0.190 | 0 |
| Simulation 3 | 0.125 | 0.000 | 0.369 | 0.000 | 0.747 | 1.344 | -2.210 | 1.125 | 0.165 | 0 |
| Simulation 4 | 0.019 | 0.261 | 0.050 | 0.000 | 0.542 | 0.947 | -0.430 | 1.355 | -0.365 | 0 |
| Simulation 5 | 0.047 | 0.100 | 0.069 | 0.328 | 0.658 | 1.275 | -0.530 | 1.530 | -0.305 | 1 |
| Simulation 6 | 0.044 | 0.047 | 0.078 | 0.453 | 0.708 | 1.319 | -0.980 | 1.285 | -1.010 | 1 |
| Simulation 7 | 0.108 | 0.197 | 0.169 | 0.294 | 0.817 | 0.825 | -1.040 | 0.540 | -2.280 | -1 |
| Simulation 8 | 0.069 | 0.194 | 0.111 | 0.414 | 0.681 | 0.756 | -0.210 | 1.245 | -0.750 | -1 |
| Simulation 9 | 0.053 | 0.264 | 0.108 | 0.264 | 0.833 | 0.808 | -0.895 | 0.400 | -2.005 | -1 |
| Simulation 10 | 0.122 | 0.303 | 0.172 | 0.419 | 0.697 | 0.681 | -0.330 | 1.205 | -0.915 | 1 |
| Simulation 11 | 0.322 | 0.308 | 0.397 | 0.406 | 0.906 | 0.883 | 0.650 | -0.280 | -0.565 | 1 |
| Simulation 12 | 0.253 | 0.228 | 0.308 | 0.564 | 0.775 | 0.881 | -0.295 | 1.885 | -0.785 | 1 |
| Simulation 13 | 0.086 | 0.000 | 0.144 | 0.567 | 0.886 | 0.856 | -0.250 | 1.470 | -1.860 | 1 |
| Simulation 14 | 0.119 | 0.000 | 0.175 | 0.578 | 0.875 | 0.883 | -0.265 | 1.805 | -2.385 | 1 |
| Simulation 15 | 0.117 | 0.000 | 0.172 | 0.550 | 0.858 | 0.875 | -0.500 | 1.660 | -2.765 | 1 |
| Simulation 16 | 0.058 | 0.189 | 0.086 | 0.386 | 0.844 | 0.836 | -0.445 | 1.055 | -0.470 | 1 |
| Simulation 17 | 0.025 | 0.219 | 0.056 | 0.369 | 0.858 | 0.850 | -0.580 | 1.910 | -0.535 | 1 |
| Simulation 18 | 0.047 | 0.186 | 0.083 | 0.381 | 0.697 | 0.678 | -0.280 | 1.300 | -0.390 | 1 |
| Simulation 19 | 0.022 | 0.156 | 0.047 | 0.386 | 0.792 | 0.800 | -0.320 | 1.900 | -0.650 | 1 |
| Simulation 20 | 0.017 | 0.186 | 0.042 | 0.450 | 0.892 | 1.053 | -0.460 | 2.035 | -0.700 | 1 |

Target Table

| | PVC | APC | RBBB | LBBB | Paced | Normal |
|---|---|---|---|---|---|---|
| Simulation 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| Simulation 2 | 1 | 0 | 0 | 0 | 0 | 0 |
| Simulation 3 | 1 | 0 | 0 | 0 | 0 | 0 |
| Simulation 4 | 0 | 1 | 0 | 0 | 0 | 0 |
| Simulation 5 | 0 | 1 | 0 | 0 | 0 | 0 |
| Simulation 6 | 0 | 1 | 0 | 0 | 0 | 0 |
| Simulation 7 | 0 | 0 | 1 | 0 | 0 | 0 |
| Simulation 8 | 0 | 0 | 1 | 0 | 0 | 0 |
| Simulation 9 | 0 | 0 | 1 | 0 | 0 | 0 |
| Simulation 10 | 0 | 0 | 0 | 1 | 0 | 0 |
| Simulation 11 | 0 | 0 | 0 | 1 | 0 | 0 |
| Simulation 12 | 0 | 0 | 0 | 1 | 0 | 0 |
| Simulation 13 | 0 | 0 | 0 | 0 | 1 | 0 |
| Simulation 14 | 0 | 0 | 0 | 0 | 1 | 0 |
| Simulation 15 | 0 | 0 | 0 | 0 | 1 | 0 |
| Simulation 16 | 0 | 0 | 0 | 0 | 0 | 1 |
| Simulation 17 | 0 | 0 | 0 | 0 | 0 | 1 |
| Simulation 18 | 0 | 0 | 0 | 0 | 0 | 1 |
| Simulation 19 | 0 | 0 | 0 | 0 | 0 | 1 |
| Simulation 20 | 0 | 0 | 0 | 0 | 0 | 1 |

Output Table

| | PVC | APC | RBBB | LBBB | Paced | Normal |
|---|---|---|---|---|---|---|
| Simulation 1 | 0.99942 | 0.00000 | 0.00005 | 0.00019 | 0.00046 | 0.00059 |
| Simulation 2 | 0.97716 | 0.00001 | 0.00003 | 0.00011 | 0.00396 | 0.13611 |
| Simulation 3 | 0.99902 | 0.00105 | 0.00013 | 0.00005 | 0.00007 | 0.00001 |
| Simulation 4 | 0.00065 | 0.99926 | 0.00061 | 0.00048 | 0.00003 | 0.00080 |
| Simulation 5 | 0.00047 | 0.99913 | 0.00019 | 0.00001 | 0.00004 | 0.00036 |
| Simulation 6 | 0.00067 | 0.99892 | 0.00010 | 0.00000 | 0.00093 | 0.00060 |
| Simulation 7 | 0.02417 | 0.00445 | 0.00017 | 0.36459 | 0.00527 | 0.00069 |
| Simulation 8 | 0.00204 | 0.14257 | 0.00004 | 0.00575 | 0.00005 | 0.00225 |
| Simulation 9 | 0.00487 | 0.05733 | 0.00060 | 0.39405 | 0.00537 | 0.03293 |
| Simulation 10 | 0.00027 | 0.00004 | 0.00024 | 0.99814 | 0.00016 | 0.00112 |
| Simulation 11 | 0.00327 | 0.00000 | 0.00127 | 0.99996 | 0.00153 | 0.00862 |
| Simulation 12 | 0.00058 | 0.00001 | 0.00040 | 0.99997 | 0.00085 | 0.00000 |
| Simulation 13 | 0.00026 | 0.00065 | 0.00027 | 0.00005 | 0.97755 | 0.08700 |
| Simulation 14 | 0.00026 | 0.00047 | 0.00034 | 0.00074 | 0.99649 | 0.00093 |
| Simulation 15 | 0.00027 | 0.00069 | 0.00049 | 0.00057 | 0.99868 | 0.00070 |
| Simulation 16 | 0.00051 | 0.00041 | 0.00006 | 0.00057 | 0.00016 | 0.99821 |
| Simulation 17 | 0.00060 | 0.00451 | 0.00027 | 0.00174 | 0.00015 | 0.96335 |
| Simulation 18 | 0.00018 | 0.00106 | 0.00017 | 0.00104 | 0.00066 | 0.99767 |
| Simulation 19 | 0.00014 | 0.01279 | 0.00035 | 0.00007 | 0.00195 | 0.98992 |
| Simulation 20 | 0.00015 | 0.21206 | 0.00055 | 0.00034 | 0.00036 | 0.56822 |

## APPENDIX E – Project Analysis

**Project Title**: Electrocardiography Detection
**Student's Name**: Sean Franklin, Joseph Wallcave

**Summary of Functional Requirements**
Our project detect abnormalities in ECG signals using an artificial neural network

**Primary Constraints**
Noise in signals makes analysis difficult, computational power is limited

**Economic**
Human Capital – Minimal human capital; this project aims to reduce the need for humans in ECG analysis
Financial Capital – As no physical materials were used, the only financial capital would be in man-hours to deploy our system
Manufactured or Real Capital – Our program should be able to run on any general purpose computer and shouldn't need a dedicated computer for just this task
Natural Capital – We don't anticipate new computers being built specifically for our program, therefore natural capital should be limited to the cost of computers already made and in use

The only inputs to our system are ECG data taken by a doctor

The only cost accrued during this project was purchasing MATLAB with the neural network toolbox
The student edition cost $130

Although this project earns no money, it can help reduce cost for patient care and increase accessibility to care

No additional maintenance or operation costs occur apart from regular computer maintenance

Estimated Development Time: 6 months
Actual Development Time: 6 months

After the project ends, future work can be done to improve results or add more features

- **If manufactured on a commercial basis:**

We do not anticipate our project being manufactured on a commercial basis. Higher accuracy would be necessary to be competitive in the medical industry.

- **Environmental**

The only environmental resources needed are electricity to run a computer. Thus, this project only uses natural resources indirectly to generate power (e.g. coal, oil, natural gas). This project doesn't affect species other than humans.

- **Manufacturability**

As there is no physical component to our project (only software), manufacturability is limited to software distribution. This can be done over the internet for little cost.

- **Sustainability**

**Manufacturing and upgrading/updating our system are limited to software distribution. The only end of life concern is indirect (computer disposal).**

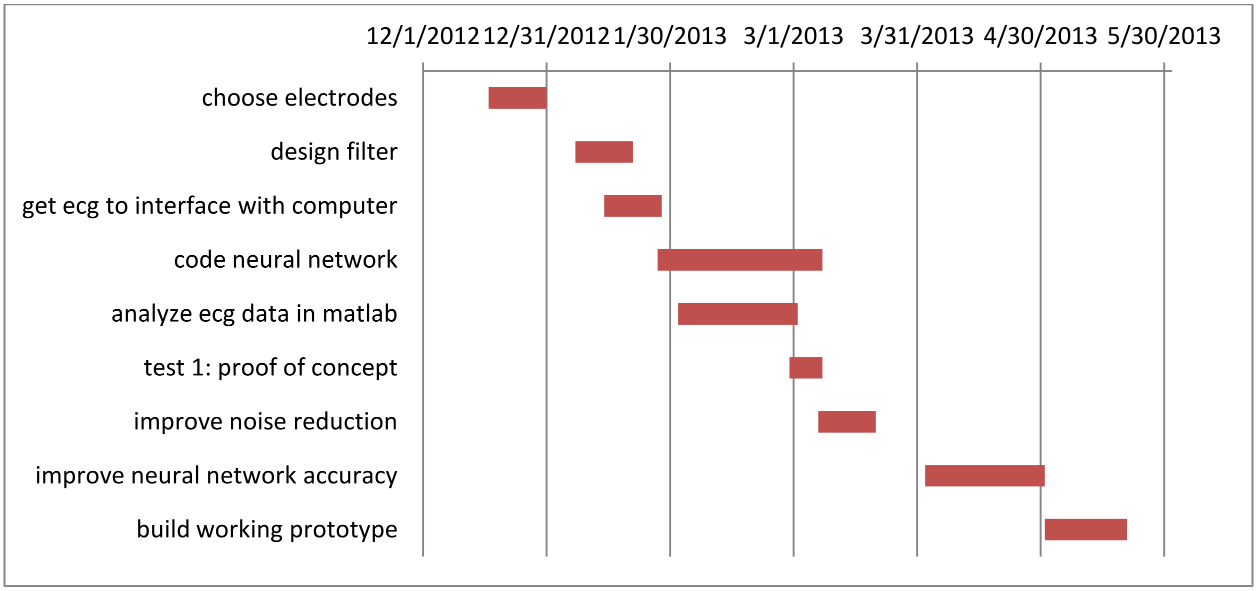
## APPENDIX F – Parts List and Cost

| Part name | Cost |
|---|---|
| MATLAB R2013a Student Edition | $100.00 |
| Neural Network Toolbox Student Edition | $30.00 |

## APPENDIX G – Specifications and Requirements

- Use Neural Network to detect the following 6 Conditions:
  1) Normal beat
  2) Left Bundle Branch Block beat (LBBB)
  3) Right Bundle Branch Block beat (RBBB)
  4) Atrial Premature Contraction beat (APC)
  5) Premature Ventricular Contraction beat (PVC)
  6) Paced beat
- Achieve At least 70% accuracy in recognition Rate

## APPENDIX H – Time Estimates



| Task Description/Deliverables | Task Description/Deliverables |
|---|---|
| 1. Choose Electrodes Select and purchase electrodes based on performance and cost | 1. Choose Electrodes Select and purchase electrodes based on performance and cost |
| 2. Design Filter | Design and construct active filters that remove noise such as 60Hz power line noise. Verify the performance. |
| 3. Interface ECG & Computer | Be able to read filtered ECG signal into computer. This data should be in a format usable by programs such as Matlab. |
| 4. Code Neural Network | Write code to implement artificial neural network. This will most likely be done using Matlab. |
| 5. Analyze ECG Data | Analyze ECG test data to train neural network |
| 6. Test 1: Proof of Concept | First use of the neural network to predict heart condition based on ECG signals. Accuracy will probably not be very high at this point, but this stage should demonstrate our idea is feasible |
| 7. Improve Noise Reduction | Based on observations from our system in use, we should be able to better understand what types of noise will be present. This will help us redesign our filtering as necessary. |
| 8. Improve Network Accuracy | Based on the results from our proof of concept, we should be able to adjust parameters such as the number of hidden layers and learning rate to achieve better performance. |
| 9. Build Working Prototype | This will be the final product shown at the May 30th Project Expo |