Psych 186B XINTAO ZHANG March 23rd,2018

Final Project Write-up

Arrhythmia Pattern Detection

Introduction

As of right now, cardiologist sometimes have to go through hours of patient Electrocardiogram (EKG) data and manually detect abnormal patterns in the data. Therefore, our group decided to apply and expand the neural network knowledge we have learnt in this class to make a computer program that take over part of the job.

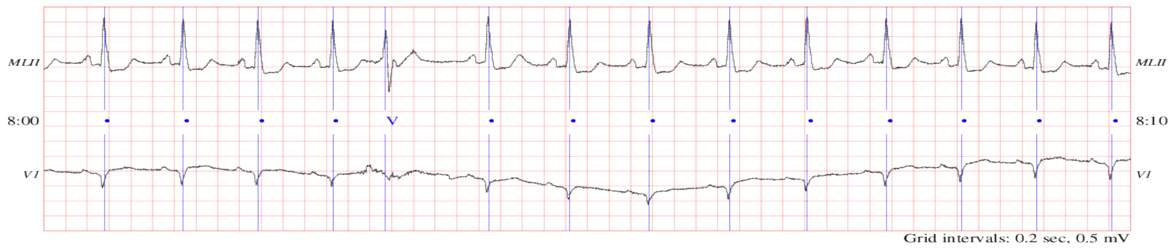
We aim to build a learning algorithm that is able to detect certain arrhythmia patterns in an Electrocardiogram(EKG). The algorithm is expected to take in as an input a piece of EKG and signal if certain arrhythmia pattern is present in the diagram. If successful, the algorithm can assist cardiologist tremendously by going through lengthy EKGs and identify abnormal heart patterns in a very short time.

Method

1. Data source

We used the MIT-BIH Arrhythmia Database to train and test the program. The database contains 48 half-hour excerpts of two-channel ambulatory EKG recordings. And the database allows us to download the raw numerical data of the electrical pulse from a patient's heart (i.e. the y-axis of the EKG). Therefore, we can store pieces of EKG in matrix form. and we used most of them as the training sets and the rest as the testing sets.

Each set of the recordings have "beat-by-beat annotations" by cardiologists. There are a total of 19 different types of annotations, each standing for a different heartbeat conditions. Below is a 10-sec sample of an annotated EKG of a patient from the MIT-BIH Database. Each dot represents a normal heartbeat identified by cardiologists, and the "V" represents a premature ventricular contraction.



2. Data Preprocessing

read_beats.m

Input:

Sample data_file format: '231m.mat'. Contains raw EKG data.
Sample annot_file format: 'clean_231.txt'. Contains annotations
Symptom: numbers from 0 to 8. The kind of symptom the user is looking for
Start: Input a value >=2. The index for the desired first heartbeat
Finish: The index for the desired last heartbeat
Desired_sample_size: The optimal sampling size e.g. 262

Output:

Return a m-by-n matrix of EKG data corresponding to a specified heart condition, with m representing the number of heartbeats, and n representing the number of data points per heartbeat

data_processing.m

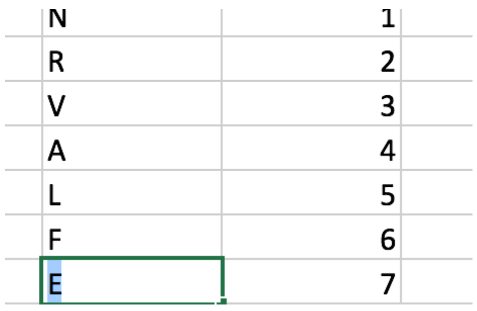
Input:

Results from read_beats.m

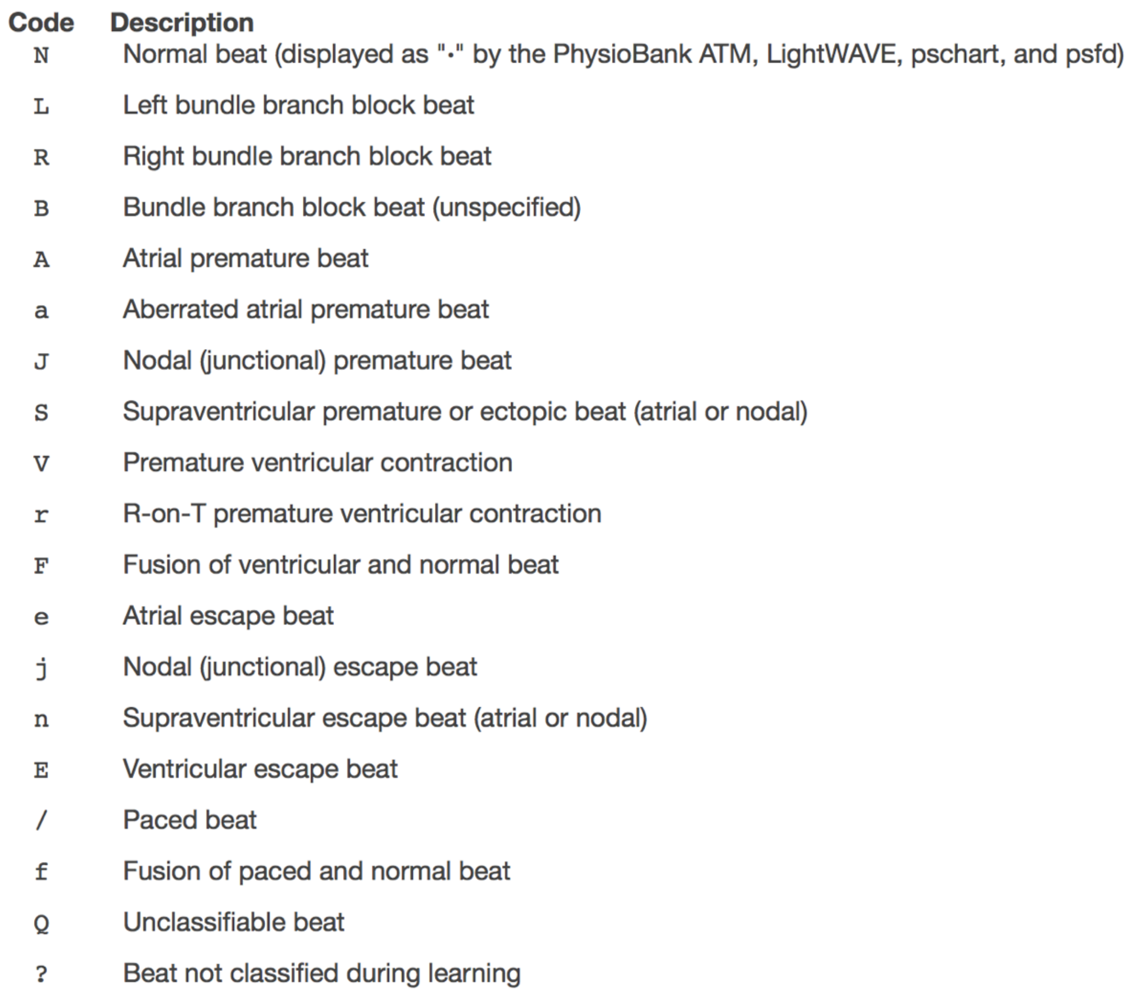
Output:

heartbeat.csv

A csv file containing >40,000 heartbeats numerical data, each of length 262 (i.e. 262 data points per heartbeat). The data has a normalized baseline. The file contains 7 different type of symptoms, labeled from 1 to 7. The correspondence of labels to symptoms is as follows:



N	1
R	2
V	3
A	4
L	5
F	6
E	7



Code	Description
N	Normal beat (displayed as "·" by the PhysioBank ATM, LightWAVE, pschart, and psfd)
L	Left bundle branch block beat
R	Right bundle branch block beat
B	Bundle branch block beat (unspecified)
A	Atrial premature beat
a	Aberrated atrial premature beat
J	Nodal (junctional) premature beat
S	Supraventricular premature or ectopic beat (atrial or nodal)
V	Premature ventricular contraction
r	R-on-T premature ventricular contraction
F	Fusion of ventricular and normal beat
e	Atrial escape beat
j	Nodal (junctional) escape beat
n	Supraventricular escape beat (atrial or nodal)
E	Ventricular escape beat
/	Paced beat
f	Fusion of paced and normal beat
Q	Unclassifiable beat
?	Beat not classified during learning

3. Learning algorithm building

   Final_project.m
   Activation_fn.m
   Activation_de.m

   Our group at first spent more than 15 hours attempting to use a three-layer network with a single hidden layer to train the model, using MATLAB. However, we had to give up in the end, since the network was struggling to achieve above 90% accuracy against testing sets

   Heartbeat.ipynb

   Our group decided to use Python to carry out the training of our network. Taking advantage of the immense free libraries that Python offers like Pandas, Keras and SKLearn.

   This program first read in the *heartbeat.csv* created earlier, and randomized the rows of the dataset. Using Keras, we were then able to created neural networks with 2,3,4 hidden layers with desired units with minimal amount of codes.

Results

To test the accuracy of our model at diagnosing arrhythmia patterns, we divided all the heartbeats data into 10 equal sets. We pick one of the set as the testing set, and the rest nine sets as the training sets. We repeat this process ten times, and take the average of the ten accuracies obtained.

As a result, we were able to obtain the best model when we built a model with 3 hidden layers, each containing 8,16,32 units. In 50 Epoch, this program was able to achieve 97.03% accuracy. In other words, given data about a single heartbeat, there is less than 3% chance that the model is going to make the wrong diagnosis out of the seven total choices.