ECG Analysis using Convolutional Neural Network

Adnan N. Qureshi, Shaighan Khan, Awais M. Lodhi, and Saeed Iqbal

Faculty of Information Technology,
University of Central Punjab,
Lahore, Pakistan

Abstract

Heart attack, occurs when blood flow decreases or stops to a part of the heart, causing irreversible damage to the heart muscle. It is a leading cause of mortality around the world according to the WHO reports and, therefore, it is critical to estimate the location and extent of the damaged tissue. Similarly, localization of MI is also significantly important to correctly manage the patient medically and/or surgically. In this paper we propose and implement a system in which the signals from 3 leads (I, 11, II) of the ECG are used to detect the cases with Heart Attack in the lateral and Inferior walls of the heart. These signals are collected from a wearable hardware device through weight and ECG sensors in real time. Detection is done by processing and passing the data through a pre-trained neural net classifier on cloud via a smartphones connected to the device as these hand held phones are carried by almost every one which is more convenient than developing a separate or integrated hardware.

Introduction

Medical conditions like Sudden Cardiac Heart Attack and Lungs infections are critical and sometimes led to unexpected death or permanent health damage and may need a prolonged recovery time. According to an estimated 17.9 million people died from Cardio Vascular Diseases (CVDs) in 2016, representing 31all global deaths. Of these deaths, 85stroke [1]. Another estimate shows that 235 million people are suffering from asthma [2]. More than 200 million people have chronic obstructive pulmonary disease (COPD) [2]. Pneumonia a type of lungs disorder is the world's leading killer of young children [2].

If a system is able to predict such attacks before they happen then a substantial amount of risk factor can be terminated. We have developed solution which is a combination of hardware and software, incorporating the ideas of Internet of Things and Deep Learning to perform real time analyses on the data stream captured by the device. After a predictive analyses of the data is done in real time a notifications and report would be sent to your phone.

Methodology

In this study we only considered patients with heart problems involving the inferior or lateral wall of the cardiac muscle. Firstly the R-peak detection was carried on the ECG signals of the selected patients. Then the signal was segmented using the detected after omitting the first and last peak in each lead.

Each segment consisted of 651 samples as described in (250 samples before the R-peak and 400 samples after the peak ensuring the complete P-QRS-T wave is present in each segment. Z-Score normalization was applied on each segment with the aim to resolve the amplitude scaling problem. Furthermore no noise removal or base line wandering correction was performed on any of the signals before or after the segmentation. After the preprocessing step we had 33,796 beats for the training set and 8,449 beats were set aside for testing purposes. The training set was now shuffled in order to ensure a randomized data is being fed to the neural network. Overall we had data for 42, 245 individual beats out of which 31,671 were diagnosed as having heart issues in the Interior or Lateral walls while 10, 574 beats were extracted from ECG records of healthy patients for control purposes. This system is started with wearable device having ECG sensors and weight load sensors. The device is connected with an application using a bluetooth device which transfers data to cloud where deep learning is applied to detect the flaws in patient's heart beat, ECG and load.

Hardware Architecture

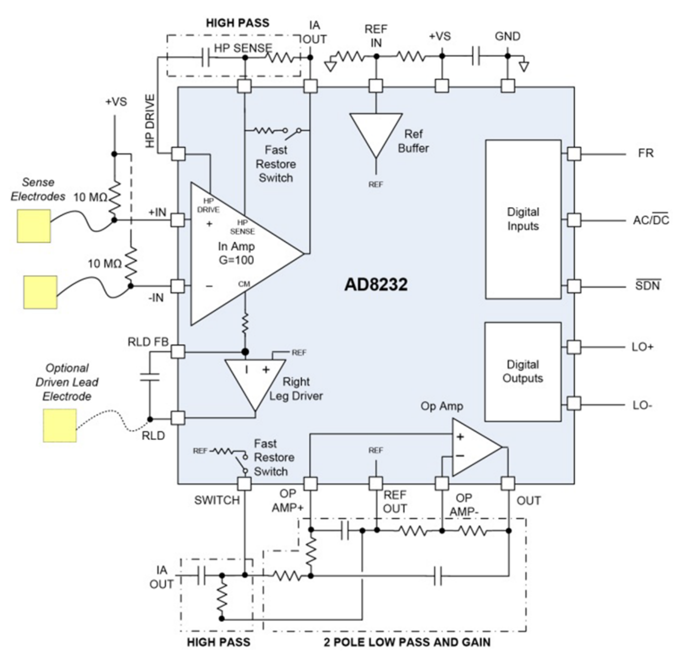


Fig. 1. AD8232 architecture

Hardware Description

The AD8232 is an integrated signal conditioning block for ECG and other biopotential measurement applications. It is designed to extract, amplify, and filter small biopotential signals in the presence of noisy conditions, such as those created by motion or remote electrode placement. The design allows for an ultralow power analog-to-digital converter (ADC) or an embedded microcontroller to acquire the output signal easily.[av]

TRAINING

Dataset

In this paper, we have used the Physikalisch-Technische Bundesanstalt diagnostic ECG database [14] available on PhysioNet. Out of the 12 leads we have taken the second lead data (V2) simultaneous signal data for 127 out of the 148 patients diagnosed with MI in different cardiac regions.

Reprocessing

In this study we considered those patients who had detailed diagnosis information available in the said dataset. Firstly, the R-peak detection was carried on the ECG signals of the selected patients using Pan Thompkins algorithm [15]. The signal was then segmented using the detected R-peaks after omitting the first and last peak in each lead. Each segment consisted of 651 samples as described in [11] (250 samples before the R-peak and 400 samples after the peak ensuring the complete P-QRS-T wave is present in each segment. To resolve the amplitude scaling problem Z-Score normalization [19] was applied on each segment. Furthermore, no noise removal or base line wandering correction was performed on any of the signals before or after the segmentation. The resultant segments were partitioned into 2 sets for training and testing purposes, also we ensured that each partition had a distinct patients ratio of 80/20 respectively. Hence, after the preprocessing step we had 65443 beats for the training set and 16209 beats were set aside for testing purposes. The data from the healthy controls were duplicated twice to get 31722 healthy beats because there was a disparity amongst the number of normal and MI patients which to a non negligible level affected the classification accuracy.. The data were further divided to apply 10-fold cross validation during the training.

CNN

CNN is a representation based learning which consists of an input layer, hidden layer(s), and an output layer [12]. This provides a network of systematic procedures which can be fed the raw data and the system automatically learns the necessary representations for classification. The term deep describes the multiple stages in the learning process of the network structure [12]. The deep learning neural network is trained using the back-propagation algorithm. The CNN is one of the most popular neural network techniques and provides quite desirable results [13].

A- Convolutional Layers

The basic building block for any CNN is the convolutional layers which are responsible for most of the computational processing. These layers automatically extract the features from the input data.

B- Batch Normalization

A batch normalization layer normalizes each input channel across a mini-batch. The layer first normalizes the activations of each channel by subtracting the mini-batch mean and dividing by the mini-batch standard deviation. Then, the layer shifts the input by a learnable offset and scales it by a learnable scale factor γ [22].

C- Rectifier Linear Unit

Activation functions are an important part of the neural networks and are used to evaluate the excitement level of the neurons. Rectifier linear units (RELU) are interesting especially since they introduce nonlinearity in the data [17].

D- Pooling Layers

Pooling function serve to down-sample and condense the features in the network thus reducing the overall computational complexity. Max-Pooling outputs only the maximum value from each kernel while sliding by a preset amount over the whole feature set. The sliding operation is called stride [17].

E- Fully Connected Layer

Neurons in a fully connected layer have connections to all activations in the previous layer, as seen in regular neural networks [17].

CNN ARCHITECTURE

CNN Model

The network as a whole contains 10 layers which include 4 *1D Convolution* layers each followed by a *batch-normalization* and a *max pooling* layer with both *pool size* and *stride* of 2. At the end of this network there are three *fully connected* layers including the output layer.

The first convolution is of 102×3.The second convolution is of 24×10. The third convolution is of 11×10 and the last convolution is of 9×10. We call the network leaky network because we have used *leakyRelu* as the activation in both the *conv* and *fully connected layer*.

During training 10-Fold Cross Validation was applied with each fold running 60 epoch which includes a validation patience of 6 with a minimum change criteria of 1×10^{-7} in the accuracy.Each result of the 10 folds were taken by taking the average of all the epochs in each respective fold.

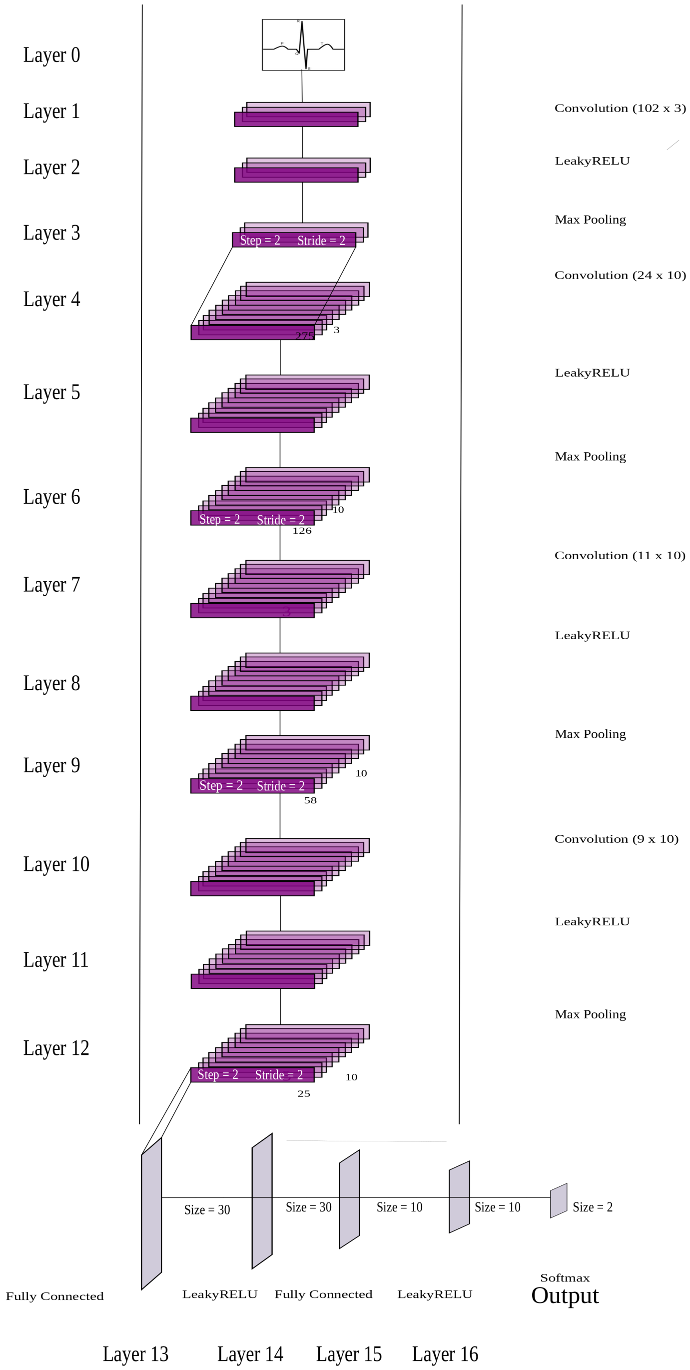
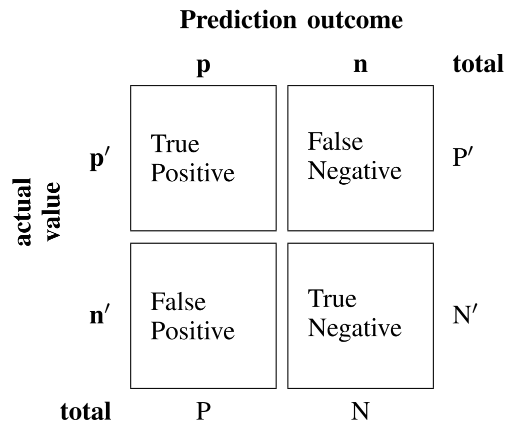


Fig. 2. Network Architecture

RESULTS

10 Fold Cross-Validation				
	Train-Acc	**Train-Loss**	**Test-Acc**	**Test-loss**
Fold-1	98.85%	29.12%	98.78%	11.64%
Fold-2	99.44%	15.84%	98.72%	8.86%
Fold-3	99.34%	22.09%	99.42%	9.74%
Fold-4	99.49%	13.23%	98.91%	7.51%
Fold-5	99.33%	20.72%	99.47%	8.71%
Fold-6	99.23%	24.88%	98.74%	11.23%
Fold-7	98.96%	27.93%	99.29%	11.22%
Fold-8	98.91%	29.41%	99.21%	11.96%
Fold-9	99.34%	18.54%	99.24%	8.81%
Fold-10	99.21%	20.95%	99.12%	9.54%

The fold with the highest test accuracy(Test-Acc) was selected and was used to test on further unknown data. We need our model to be as much robust and accurate as possible due to which we need to test it on new data again to check if it produces favorable results or not.



References

[av]https://www.analog.com/en/products/ad8232.html