NORTH SOUTH UNIVERSITY
Center of Excellence in Higher Education
The first private university in Bangladesh

Intern/Co-op/Directed Research

ECG Interval Segmentation Using Deep Learning

Submitted by
Name : Akhtaruzzaman Khan
Id: 1611319042

Supervised by
Dr. Tanzilur Rahman
Assistant Professor
Department of Electrical and Computer Engineering

1. INTRODUCTION

Identifying individual components (segments) in the ECG waveform accurately is important to correctly diagnose various heart conditions. Clinicians look for subtle patterns and repeating features in order to correctly identify each region of the ECG wave. They spend a significant amount of time identifying and annotating individual components of data to aid in diagnosis [1]. This is true when analyzing large volumes of ECG data e.g., Holter analysis, research purposes, mass screening of patients. Thus, having clinicians annotate ECGs on a large scale is infeasible. Automation of the segmentation and annotation of these large volumes of ECG data would enable the clinicians to pick the anomalous portions of the ECG data for further analysis and thereby save a significant amount of time. The automation of ECG analysis would help medical practitioners and researchers in easily accessing and analyzing various complexes of ECGs across multiple patients the main challenge with ECG segmentation is that ECG waves have a lot of variations. They can have different characteristics in terms of shape, frequency, duration and amplitude. These variations stem from the difference in patients, the monitoring equipment, the placement of the ECG leads etc. An automated approach needs to be able to efficiently handle all possible variations in the ECG waves. Traditional automated models first identify the peak of the R wave and then use rule-based approaches or Hidden Markov Models (HMM) to find the position of the remaining waves. These approaches suffer from low to average performance as they are not robust enough to handle the multitude of variations in ECGs and therefore are not generalized [2]. In this research We propose a Feed Forward, LSTM and Bidirectional LSTM neural network model for the segmentation of ECG waves. It labels each data point as either a P-wave, QRS-wave, T-wave or neutral.

2. DATASET

For this research, the dataset used is the QT data-set from PhysioNet. This data-set contains 50 signals where each signal contains 1250 data points two lead ECG recordings, with onset, peak and end markers for P, QRS and T waves. The recorded ECG's were sampled at 250 Hz and then provided as a digital signal. For this study every ECG sample is divided into segments each of length 1250 data points. Both leads of the ECG were used as features rather than splitting them up into separate records. These 1250 data points act as input to the deep learning model along with the additional features.

3. MODEL ANALYSIS

3.1 <u>Feed Forward Neural Network:</u>

Deep Feed-forward networks or also known multi-layer perceptions are the foundation of most deep learning models. Networks like CNN and RNN are just some special cases of Feed-forward networks. These networks are mostly used for supervised machine learning tasks where we already know the target function. the result we want our network to achieve and are extremely important for practicing machine learning and form the basis of many commercial applications, areas such as Computer vision and NLP were greatly affected by the presence of these networks. Feed Forward Neural Network it consists of an organized layer between input, hidden and output. Every unit in a layer is connected with all the units in the previous layer .

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 48)                96
_________________________________________________________________
dense_1 (Dense)              (None, 15)                735
_________________________________________________________________
dense_2 (Dense)              (None, 4)                 64
=================================================================
Total params: 895
Trainable params: 895
Non-trainable params: 0
_________________________________________________________________
```

Fig: 1 Model Architecture of Feed Forward NN

These connections are not all equal: each connection may have a different strength or weight. The weights on these connections encode the knowledge of a network. The first layer of the neural network takes raw data as an input, processes it, extracts some information and passes it to the next layer as an output. Each layer then processes the information given by the previous one and repeats, until data reaches the final layer, which makes a prediction. This prediction is compared with the known result and then, by a method called "back-propagation" the model is able to learn the weights that yield accurate outputs [3]. In this research 3 dense layer has been used where 1st layer contains 48 neurons, 2nd layer contains 15 neurons and finally the output layer is 4 (fig: 1).

3.2 LSTM:

Long Short-Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997), and were refined and popularized by many people in following work. They work tremendously well on a large variety of problems, and are now widely used. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn! All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer. LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 1, 128)            66560
_________________________________________________________________
lstm_1 (LSTM)                (None, 1, 110)            105160
_________________________________________________________________
lstm_2 (LSTM)                (None, 50)                32200
_________________________________________________________________
dense (Dense)                (None, 4)                 204
=================================================================
Total params: 204,124
Trainable params: 204,124
Non-trainable params: 0
_________________________________________________________________
```

Fig: 2 Model Architecture of LSTM

having a single neural network layer, there are four, interacting in a very special way. In the above diagram, each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines merging denote concatenation, while a line forking denotes its content being copied and the copies going to different locations. The key to LSTMs is the cell state, the horizontal line running through the top of the diagram. The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It’s very easy for information to just flow along it unchanged. The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates. Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation. The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means “let nothing through,” while a value of one means “let everything through!” An LSTM has three of these gates, to protect and control the cell state [4].

3.3 Biderectional LSTM

Bidirectional LSTMs are an extension of traditional LSTMs that can improve model performance on sequence classification problems. In problems where all timesteps of the input sequence are available, Bidirectional LSTMs train two instead of one LSTMs on the input sequence. The first on the input sequence as-is and the second on a reversed copy of the input sequence. This can provide additional context to the network and result in faster and even fuller learning on the problem. LSTM and their bidirectional variants are popular because they have tried to learn how and when to forget and when not to using gates in their architecture. In RNN architectures, vanishing gradients was a big problem and caused those nets not to learn so much. Using Bidirectional LSTM, feed the learning algorithm with the original data once from beginning to the end and once from end to beginning [5].

4. *RESULTS*

Model	Accuracy	Cross Validation (k=5)
Feed Forward	58%	64%
LSTM	58%	70%
Bidirectional LSTM	58%	61%

Table: 1 Accuracy table for Feed Forward, LSTM, Bidirectional LSTM

Feed Forward NN, LSTM and Bidirectional LSTM models is trained over 100 epochs with batch normalization to avoid over-fitting. Categorical cross entropy is used as the loss metric throughout the training process. For each of the wave segments in ECG. In Feed Forward NN model 3 dense layer has been used where 1st layer contains 48 neurons, 2nd layer contains 15 neurons and finally the output layer is 4 see the model summery (Fig: 1). after doing random train-test split over 50 ECG signal we got the accuracy 58%. and after Cross validation we get 64% accuracy. In LSTM RNN model 3 LSTM layer has been used where 1st layer contains 128 neurons, 2nd layer contains 110 neurons, 3rd layer contains 50 and finally the output Dense layer is 4 see the model summery (Fig: 2). after doing random train-test split over 50 ECG signal we got the accuracy 58%. and after Cross validation we get 70% accuracy and for Bidirectional LSTM 3 LSTM layer has been used where 1st layer contains 128 neurons, 2nd layer contains 110 neurons, 3rd Bidirectional layer contains 50 and finally the output Dense layer is 4. after doing random train-test split over 50 ECG signal we got the accuracy 58%. and after Cross validation we get 61% accuracy

5. *CONCLUSION*

In this paper we propose Feed-forward, LSTM and Bidirectional LSTM for ECG segmentation. The proposed deep learning model performs Average because we used extremely small dataset (50 samples) than existing signal processing methods. Its performance also surpasses other deep learning segmentation models, especially in segmenting P-waves and QRS-waves. Accurate automated ECG segmentation would help in developing automated cardiac diagnosis methods and simplify large scale screening tasks.

References

[1] Q. A. a. T. L. G. a. K. M. a. A. T. a. A. M. R. a. S. H. Rahman, Identifying hypertrophic cardiomyopathy patients by classifying individual heartbeats from 12-lead ECG signals, IEEE, 2014, pp. 224--229.

[2] A. a. H. S. a. S. G. a. S. K. a. S. N. Malali, "Supervised ECG wave segmentation using convolutional LSTM," *ICT Express,* 2020.

[3] Y. Upadhyay, "Introduction to FeedForward Neural Networks," March 2019. [Online]. Available: https://towardsdatascience.com/feed-forward-neural-networks-c503faa46620.

[4] Colah, "Colah's Blog," 27 August 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[5] J. Brownlee, "Machine Learning Mastery," June 2017. [Online]. Available: https://machinelearningmastery.com/develop-bidirectional-lstm-sequence-classification-python-keras/.