Atrial Fibrillation Classification Using QRS Complex Features and LSTM

Vykintas Maknickas[1], Algirdas Maknickas[2]

[1] Tesonet LLC, Vilnius, Lithuania
[2] Institute of Mechanical Science, Faculty of Mechanics,
& Department of Information Technologies, Faculty of Fundamental Sciences,
Vilnius Gediminas Technical University, Vilnius, Lithuania

Abstract

Classification of Atrial Fibrillation from diverse electrocardiographic (ECG) signals is the challenging objective of the 2017 Physionet Challenge. We suggest a Long Short Term Memory (LSTM) network, which learns patterns directly from pre-computed QRS complex features that classifies ECG signals. Although our architecture is considered deep, it only consists of 1791 parameters. The result is an accurate, lightweight solution that classifies ECG records as Normal, Atrial fibrillation, Other or Too noisy with final challenge score of 0.78.

1. Introduction

Atrial fibrillation is associated with a wide range of cardiovascular events, including cardiovascular mortality, major cardiovascular events, heart failure, ischaemic heart disease, chronic kidney disease, and sudden cardiac death, in addition to a stroke and all-cause mortality [1]. Electrocardiogram (ECG) signal data is available to hospital staff and individuals at home using a variety of sensors that collect data with a wide range of accuracies. Accurate classification of signals could help non-clinically trained individuals to benefit from commercial ECG collection devices by preventing cardiovascular events using an algorithm that detects Atrial Fibrillation from ECG data. The objective of the 2017 Physionet Challenge is the classification of ECG signals as Normal, Atrial Fibrillation, Other and Too noisy. Because ECG data is a reliable indicator of various heart arrhythmias, automated algorithms that analyse ECG data is a popular research topic. Long Short Term Memory (LSTM) networks are designed to classify, process and predict data points, which are listed in temporal order. LSTM network are known to be powerful, clinical, medical time series data classifiers [2]. Inspired by the recent success of deep learning, we trained LSTM networks on pre-computed QRS complex features.

Rajpurkar et al. [3] developed an algorithm that exceeded the performance of board-certied cardiologists when detecting a wide range of heart arrhythmias from electrocardiograms recorded with a single-lead wearable monitor. To accomplish this impressive feat, they trained on a dataset of 500 patients with a 34-layer convolutional neural network that maps a sequence of ECG samples to a sequence of rhythm classes. Although our architecture is considered deep, it only consists of 1791 parameters. This result is an accurate lightweight solution to classify ECG data as Normal, Atrial fibrillation, Other or Too noisy that can easily be developed for use in mobile applications (apps).

2. Data preparation and future extraction

Features extracted in the former stage are highly dependent on successful R peak detection since inaccuracies can lead to inaccurate derived features. To reduce noise in the signal, Finite impulse response (FIR) bandpass filters [4], between 3 Hz and 45 Hz, were used. To detect R peaks in raw ECG signals, we used a Hamiltonian-based modification of the Pan-Tompkins R peak detection algorithm [5] [6]. Features, derived from detected R peaks, along with their extraction technique, are listed below.

RR Interval - Hamiltonian transform-based modification of Pan-Tompkins R peak detection algorithm [5] [6].

QQ Interval - Q Peaks were detected by finding a minimal value in range of 6 values (or sample rate multiplied by 0.02 threshold) before R peak.

SS Interval - S Peaks were detected by finding a minimal value in range of 15 values (or sample rate multiplied by 0.05 threshold) after R peak.

PP Interval - P peaks were detected using a moving average crosses algorithm for P and T wave detection by Xinqi Louis Wang and J. Mikael Eklund [7].

TT Interval - T peaks were detected using a moving

ISSN: 2325-887X DOI:10.22489/CinC.2017.350-114

average crosses algorithm for P and T wave detection by Xinqi Louis Wang and J. Mikael Eklund [7].

SQ Interval length - SQ time Interval length was calculated by subtracting the corresponding S and Q peak indices.

PR Interval length - PR time Interval length was calculated by subtracting the corresponding R and R peak indices.

QT Interval length - QT time Interval length was calculated by subtracting the corresponding Q and T peak indices.

ST Interval length - ST time Interval length was calculated by subtracting the corresponding S and T peak indices.

Standard deviation in range $Q_i - S_{i+1}$ - Standard deviation of the time interval between Q peak and following S peak.

Number of peaks in between Q_i and S_{i+1} - Number of peaks detected, in the interval between Q peak and S peak, using moving average crossover algorithms for P and T wave detection by Xinqi Louis Wang and J. Mikael Eklund [7].

$RR_i - RR_{i+1}$ - Difference between two consecutive RR time intervals.

$R_{Amplitude_i} - R_{Amplitude_{i+1}}$ - Difference between two consecutive R amplitudes.

R Amplitude - R peak value (mV).

S Amplitude - S peak value (mV).

RS Amplitude - Sum of R and S amplitudes.

RQ Amplitude - Sum of R and Q amplitudes.

P Amplitude - P peak value (mV).

T Amplitude - T peak value (mV).

R Amplitude difference - Difference between two consecutive R amplitude values.

PQ Interval length - Number of signal samples in-range between P and Q peaks.

P width - Signal length between the moving average crossovers of the corresponding P peak.

T Width - Signal length between moving average crossovers of the corresponding T peak.

Heart rate turbulence offset - Heart rate turbulence offset is the difference between the heart rate immediately following Premature Ventricular Contraction (PVC) and the heart rate immediately preceding PVC, but we calculated offset for each RR interval using the formula $TO_i = ((RR_{i+1} + RR_i) - (RR_i + RR_{i-1}))/(RR_i + RR_{i-1})$.

The sequence of features were divided into chunks of 46 time steps. Sequences that were shorter were padded with zeros. All features were standardized by removing the mean and by scaling to unit variance. Examples of the single recording of the feature set is shown in Figures 1, 2 and 3.

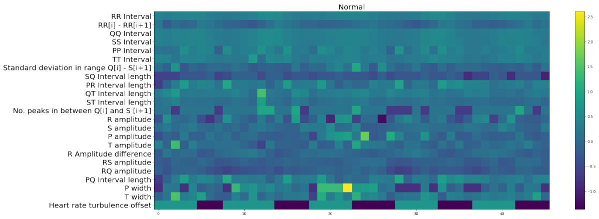

Figure 1. Signal labelled as Normal.

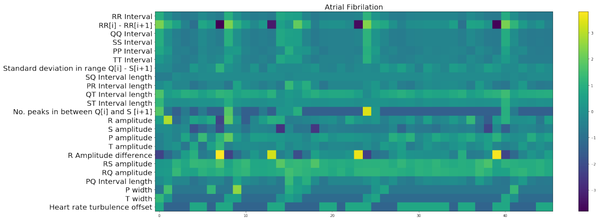

Figure 2. Signal labelled as Atrial Fibrillation.

3. Neural network architecture

In this section, we describe the architecture of the neural networks in more detail. We optimize the architecture in multiple steps. The final structure of the deep neural network is displayed in Table 1. LSTM networks were introduced by Sepp Hochreiter and Jrgen Schmidhuber [8]. They are designed to learn when incoming information is important, when it should be integrated into the internal state of a cell, and when it should propagate its internal state with input, forget, and output gates. Since extracted features are distributed in the time domain, such architectures could accurately classify ECG signal into the given categories.

Table 1. Structure of the deep neural network.

	Layer Type, Size	Parameters
1	LSTM (8x46)	1024
2	LSTM (8x46)	544
3	LSTM (4)	208
4	MLP (3)	15

4. Submission for test

LSTM is trained with error backpropagation using Adaptive Momentum. The learning rate is set to 0.0009. The optimizer and categorical cross-entropy is set as the loss function. 80% of samples were used to train algorithm and the other 20% were kept as a validation sub-set. The training samples were divided into batches; there were 128 samples per batch. The algorithm was trained for 15000 epochs. After each epoch, a validation accuracy was determined. The weights that had the best validation accuracy

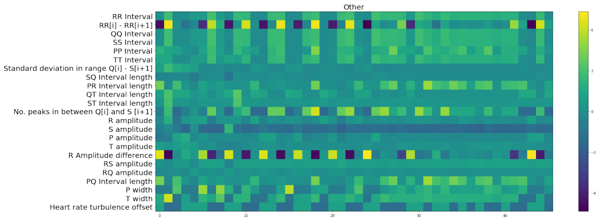

Figure 3. Signal labelled as Other.

were selected and set aside after epoch; those weights were then used in a submission to test the algorithm in the Physionet Challenge environment. The calculated scores are listed in Table 2.

Table 2. The classification accuracy of the algorithm.

Type		F1 score
Normal		0.90
Atrial fibrillation		0.75
Other		0.69
	Average	0.78

5. Conclusion and future work

An algorithm capable of classifying ECG signals as Normal, Atrial fibrillation, Other and Too noisy, with an average F1 score of 0.78, has been developed. A total of 23 features were extracted from signals for each detected QRS complex. These features are then standardized across the entire training data. Classification is done using a three-layer LSTM neural network and trained with error backpropagation. Future work may include using LSTM attention mechanisms to determine a time-step importance measure to better understand each features relevance.

References

[1] Ayodele Odutayo, Christopher X Wong, Allan J Hsiao, Sally Hopewell, Douglas G Altman, Connor A Emdin, Atrial fibrillation and risks of cardiovascular disease, renal disease, and death: systematic review and meta-analysis, BMJ 2016; 354 https://doi.org/10.1136/bmj.i4482

[2] ZacharyC., DavidC., CharlesElkan, RandallWetzel Laura P. and Leland K. Whittier. LEARNING TO DIAGNOSE WITH LSTM RECURRENT NEURAL NETWORKS, https://arxiv.org/pdf/1511.03677.pdf

[3] PranavRajpurkar, AwniY.Hannun, Masoumeh Haghpanahi, CodieBourn, Andrew Ng Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks https://arxiv.org/pdf/1707.01836.pdf

[4] Technical report, report by: John Treichler. Notes on the Design of Optimal FIR Filters http://cnx.org/contents/c2c5f5d0-6692-49d9-9ceb-8126f9ac0a34%403.2

[5] Patrick S. Hamilton, E.P. Limited. Open Source ECG Analysis Software Documentation http://www.eplimited.com/osea13.pdf

[6] JIAPU PAN AND WILLIS J. TOMPKINS, A Real-Time QRS Detection Algorithm, IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. BME-32, NO. 3, MARCH 1985

[7] Xinqi Louis Wang, J. Mikael Eklund. A real-time ECG feature detection algorithm, 7th Canadian Student Conference on Biomedical Computing and Engineering and the IEEE EMBS International Student Conference (CSCBCE/ISC), Oshawa, ON, Canada from June 25 26, 2014.

[8] Sepp Hochreiter; Jrgen Schmidhuber (1997). ”Long short-term memory”. Neural Computation. 9 (8): 17351780. PMID 9377276. doi:10.1162/neco.1997.9.8.1735.

Address for correspondence:

Algirdas Maknickas
Institute of Mechanics & Department of Information Technologies,
Vilnius Gediminas Technical University,
Basanaviciaus st. 28, 01223 Vilnius, Lithuania
algirdas.maknickas@vgtu.lt