Classification of 12-lead ECG Signals Using Intra-Beat Fast Fourier Transform and Inter-Beat Attention

Ibrahim Hammoud[1], IV Ramakrishnan[1], Petar M. Djurić [1]

[1]Stony Brook University, Stony Brook, US

Abstract

In this work, we built a model to classify 12-lead ECGs based on attention for the PhysioNet/Computing in Cardiology Challenge 2020. We use discrete-time fast Fourier transform to extract essential characteristics of the given high-frequency signals. Since information about different classification outcomes might be present only in specific segments, we tune our feature representation to show the frequency distribution shift as we move through time. This is done by first representing the original signal as a spectrogram, which shows the signal's frequency spectrum during different time windows where a beat represents each window. This spectrogram is then inputted to a bidirectional LSTM network where each heartbeat vector represents a time step. The outputs of the bidirectional LSTM network at each stage are then used as attention vectors. The attention vectors are then multiplied with the original signal window embeddings before they are summed up and multiplied by a final linear architecture that provides an output. Our current approach achieves a score of 0.416 on the leaderboard of the challenge's official phase (team name: SBU_AI).

1. Introduction

Cardiovascular disease is estimated to be one of the leading causes of death worldwide [1]. There are many types of cardiovascular diseases resulting from different underlying causes. To detect such causes and understand the kind of intervention needed, the electrocardiogram (ECG) is an important tool used by healthcare professionals to assess and diagnose abnormalities in the cardiac electrical activity in patients [2, 3]. The PhysioNet/Computing in Cardiology Challenge 2020 is organized to encourage automated, data-driven, and open-source approaches for classifying different types of abnormalities from 12-lead ECGs [4, 5]. In this paper, we describe in more detail the model we submitted to this year's challenge. Our model calculates a discrete-time fast Fourier transform characteristics for every beat and feeds that information to an attention mechanism before outputting probability scores, which are later used for classification.

2. Methods

2.1. Data processing

We separate our data-set into 70% training and 30% testing. For each data-point, we only process the first 10 seconds of each signal, and we also extract up to a max of 10 heartbeats only due to computational and time limitations. After that, we extract features from each data-point. Those features can be separated into fixed-length and variable-length features. Fixed length features are extracted from the header file, in addition to other characteristics that are detected from the overall signal, such as the heart rate. Variable-length features are the ones that are extracted for every heartbeat.

Fixed length features
- Heart rate
- Heart beat intervals mean
- Heart beat intervals standard deviation
- Heart beat difference intervals mean
- Heart beat difference intervals standard deviation
- Age
- Sex
- Signal gain

Variable length features
- Per heart-beat fast Fourier transform spectrogram.

2.2. Peak Detection

Since some heart rate signals can be noisy, we first rank the 12-lead signals by a quality metric that allows us to detect the peak locations accurately for the rest of the signals. To do that, we select our best lead-signal as follows:

After selecting the best lead-signal, we run our peak detection algorithm on that signal. In the first step, we auto

Algorithm 1 : Select best lead

Input: Normalized signals $e_1, e_2 ..., e_{12}$
Output: Index of signal used for peak detection.

procedure GETBESTLEADINDEX
  for $i = 1 \rightarrow 12$ **do**
    $de_i \leftarrow$ DIFFERENCEOPERATOR(e_i)
    $qm \leftarrow \frac{\text{minimum of top 5 gradients}}{\text{average gradient of the bottom 50\% positive gradients}}$
    measures.append(qm)
  end for
  Return $\arg\min_i$ measures
end procedure

correlate the gradient of the signal with itself. Since we search for heart rates within the range $(30, 240)$, we only consider the maximum value within that range in the autocorrelation graph. Moreover, since the autocorrelation plot can have multiple peaks, some of which are higher than the first peak, we add a tolerance parameter to accept the earliest peak if it is greater than or equal to 40% of the maximum peak in the specified range.

After this step, we are able to identify an approximate heart rate for the given signal. We use this information to detect the peaks. We use Scipy's peak detector. We use a min distance between peaks of up to 40% of the distance given the approximate heart rate. We also use a prominence of 60% of the 5^{th} highest gradient. These two parameters allow for an acceptable range of variation along the amplitude and the period in the derived heart rate signal. The parameters were selected and tuned over a sample of 50 random heart rate signals. We stopped the tuning after the overall peak detection pipeline was able to detect peaks correctly with more than 99% accuracy. Figure 1 shows the distribution of the heart rates across the whole challenge dataset after applying our peak detection pipeline.

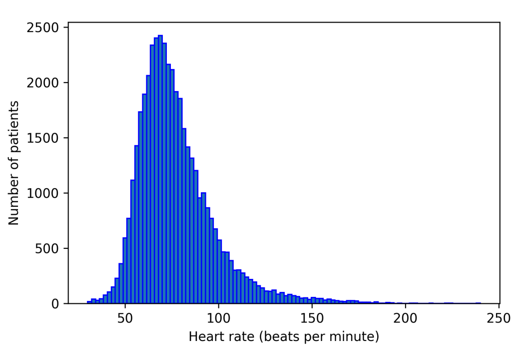

Figure 1. Histogram showing the distribution of the heart rates after applying our peak detection pipeline.

2.3. Model

We append the patient context features to each heartbeat feature vector and feed the concatenated vector to a fully connected layer. After that, we feed the outputs from all the beats in our datapoint into a bidirectional LSTM. We apply attention to the outputs of the bidirectional LSTM as in [6] and then feed the results into a final linear layer. We apply a softmax to the final layer's output and use the binary cross-entropy loss on each of the 24 classes.

2.3.1. Model hyper-parameters and tuning

We fix our model architecture parameters such as the input embedding dimension (LSTM input dimension) to 512, the LSTM hidden dimension, and all subsequent hidden layer dimensions to 256. We fix a batch size of 64. We also set the number of training steps to 10 epochs.

As for other learning parameters, we tune the learning rate, gradient clip norm, λ_1, λ_2, and dropout rate. We use Bayesian optimization with 40 total trials after separating the training dataset into training and validation to tune these parameters. The parameters and their ranges are shown in Table 1. We then select the trial giving the best performance on the validation set.

Parameter	Range	Range type
Learning rate	$[-10, -5]$	Log uniform
Gradient clip norm	[0, 10]	Uniform
L1 lambda	[-10, 2]	Log uniform
L2 lambda	[-10, 2]	Log uniform
Dropout	[0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5]	Discrete

Table 1. Search space for tuned hyper-parameters

2.3.2. Model threshold tuning

After training and selecting a model that minimizes the loss and generalizes well to the validation set, we need to select thresholds for each class in a way that optimizes for the challenge metric. Usually, this is easily done in binary classification tasks using a linear time algorithm that sweeps all possible thresholds and selects the optimal one. In the given challenge metric, however, this task is more challenging for two main reasons:

1. The threshold of each class affects the scoring of all other classes in different ways even if the thresholds of all those other classes are held constant.
2. The number of classes is large (24), which makes grid search based optimization of the thresholds intractable.

Given this, we optimize for the thresholds using Bayesian-optimization. We use 200 trials, and we select the thresholds that maximize the challenge scoring function on the training set. We also apply an additional optimization to accelerate the threshold search. We limit the threshold search to be within the 95^{th} quantile of the positive class distribution across the training set. The reasoning behind this range is that higher values mean omitting that class entirely (all negative predictions). A value below that range might correspond to outliers whose scores are very close to 0. Limiting our search space to this narrower space increased convergence speed compared to searching the whole range (0 to 1).

3. Results

We used one of the 10 submissions in the official phase of the challenge. The results are shown in Table 2

Team Name	SBU_AI	Number of submissions
Official phase score	0.416	1 out of 10
Unofficial phase score	0.596	3 out of 5

Table 2. Official phase results

We also evaluated our algorithm on the training and validation sets, the results are shown in Table 3

Data split	Challenge score
Training	0.548
Validation	0.523

Table 3. Training and validation data scores

4. Discussion and Conclusions

Our model was able to give a relatively good performance, given many limitations. Mainly, we cropped all signals to 10-second windows. This resulted in a reduced dataset. We also did minimal feature engineering after getting the peaks, or on the separated heart-beat signals. We also followed an outcome agnostic approach. In a sense, we didn't include special features to account for specific outcomes. This work aimed to demonstrate this approach's applicability in general, and we leave adding improvements to the model to future work. Moreover, since we are using an attention mechanism, our model can be modified to give more insight by probing the attention weights or each beat's contribution to the final probability score similar to the approach followed in [7].

Acknowledgments

This work was supported by NSF Awards: 1805076, 1936027 and NIH Awards: R01EY030085, R01HD097188.

References

[1] Benjamin EJ, Muntner P, Alonso A, Bittencourt MS, Callaway CW, Carson AP, Chamberlain AM, Chang AR, Cheng S, Das SR, et al. Heart Disease and Stroke Statistics – 2019 Update: a report From the American Heart Association. Circulation 2019;.

[2] Kligfield P, Gettes LS, Bailey JJ, Childers R, Deal BJ, Hancock EW, Van Herpen G, Kors JA, Macfarlane P, Mirvis DM, et al. Recommendations for the standardization and interpretation of the electrocardiogram: part i: the electrocardiogram and its technology a scientific statement from the American Heart Association electrocardiography and arrhythmias committee, council on clinical cardiology; the American college of cardiology foundation; and the Heart Rhythm Society endorsed by the International Society for Computerized Electrocardiology. Journal of the American College of Cardiology 2007;49(10):1109–1127.

[3] Kligfield P. The centennial of the Einthoven electrocardiogram. Journal of Electrocardiology 2002;35(4):123–129.

[4] Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, 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 2000;101(23):e215–e220.

[5] Perez Alday EA, Gu A, Shah A, Robichaux C, Wong AKI, Liu C, Liu F, Rad BA, Elola A, Seyedi S, Li Q, Sharma A, Clifford GD, Reyna MA. Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020. Under Review 2020;.

[6] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv14090473 2014;.

[7] Choi E, Bahadori MT, Sun J, Kulas J, Schuetz A, Stewart W. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R (eds.), Advances in Neural Information Processing Systems 29. Curran Associates, Inc., 2016; 3504–3512.

Address for correspondence:

Ibrahim Hammoud
350 Circle Rd, Stony Brook, NY, United States
ihammoud@cs.stonybrook.edu