Classification of 12-lead ECGs Using Intra-Heartbeat Discrete-time Fourier Transform and Inter-Heartbeat Attention

Ibrahim Hammoud, IV Ramakrishnan, Petar M. Djurić

Stony Brook University, Stony Brook, US

Abstract

In this work, we built a model to classify 12-lead ECGs using attention for the PhysioNet/Computing in Cardiology Challenge 2020. 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 (heartbeats). The frequency spectrum at each heartbeat is extracted using discrete-time Fourier transform. The 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, which are used to generate the final output. Our approach achieved a challenge validation score of 0.416 and a test score of 0.024 but were not ranked due to omissions in the submission (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 discuss the design and the results of the model we submitted to this year's challenge.

2. Methods

2.1. Data processing

We merged all the challenge datasets into one larger dataset of 43,101 12-lead ECGs. For each data-point (a single 12-lead ECG recording), we only process a 10 second window due to computational limitations. We extract the window from the middle of the signal to avoid potential noise at the beginning or end of the signal when possible. We also extract up to a maximum of 10 heartbeats per data-point in the specified 10 second window. After that, features were extracted for each data-point from the signal and its accompanying header file. We group the features into two main types: static and time-varying, as shown in Table 1. Static features are characteristics of the patient and the signal as a whole, while time-varying features are characteristics calculated at the heartbeat level.

2.2. Peak detection

Since the dataset contains noisy ECG signals, we leverage the availability of 12-leads per ECG to improve the quality of our peak detection pipeline. As shown in Algorithm 1, we use a heuristic to select the lead ECG with the most prominent peaks. This signal will be later used to extract the actual timestamps of the peaks. The heuristic calculates the ratio of the 5^{th} highest gradient within the 10-second window from each lead signal over the average of the bottom 50% of the positive gradients within that window. Since we assume a minimum heart rate of 30 beats per minute, the 5^{th} highest gradient is used as a more conservative estimate of a peak gradient than taking the average or the median of the top 5 gradients. Also, taking the bottom 50% of the positive gradients as in the denominator of line 5 of Algorithm 1 aims at capturing the gradients from the noise while excluding the gradients from the peaks and other characteristics of the signal.

After selecting the lead signal maximizing our signal quality metric, we extract an approximate heart rate for the signal before applying our peak detection algorithm. This step is needed to adjust the minimum width between the

Feature name	Type	Source
Heart rate	Static	Signal
Standard deviation of inter-beat intervals	Static	Signal
Mean of difference intervals	Static	Signal
Standard deviation of difference intervals	Static	Signal
Age	Static	Header file
Sex	Static	Header file
Signal gain	Static	Header file
Per heartbeat discrete-time Fourier transform	Time-varying	Signal
Inter-beat interval duration	Time-varying	Signal

Table 1: Features used in our model. Time-varying features are calculated for each heartbeat. Static features are extracted or calculated once for the whole signal before they are fed to our model.

Algorithm 1 : Get best lead signal index

Input: Lead 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)
    $qm \leftarrow \frac{\text{minimum of top 5 gradients in } de_i}{\text{average of the bottom 50\% positive gradients in } de_i}$
    quality_measures.append(qm)
  end for
  Return $\arg\min_i$ quality_measures[i]
end procedure

peaks appropriately. To achieve this, we cross-correlate the derivative of the lead signal with itself. Since we assume the heart rates (per minute) fall within the range $(30, 240)$, we only search for peak values in the autocorrelation plot corresponding to a signal period within that range. Moreover, since the autocorrelation plot can have multiple peaks, some of which are higher than the first peak (corresponding to the true signal period or the actual heart rate), we add a tolerance parameter in our algorithm to accept the earliest peak if it is greater than or equal to 40% of the maximum peak in the specified range. We use the earliest peak to calculate the period and thus the approximate heart rate of the signal.

After this step, and in order to identify peaks in the signal, we use a minimum distance between peaks of at least 40% of the distance corresponding to the given approximate heart rate (as long as it doesn't correspond to a heart rate higher than 240). We also use a prominence of 60% of the 5^{th} highest gradient within the 10-second window. Values below this threshold are removed from the pool of potential peaks. These two parameters allow for an acceptable range of variation in the amplitude and the period of the derived heart rate signal. After that, we select our peaks, given these two parameters. 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.

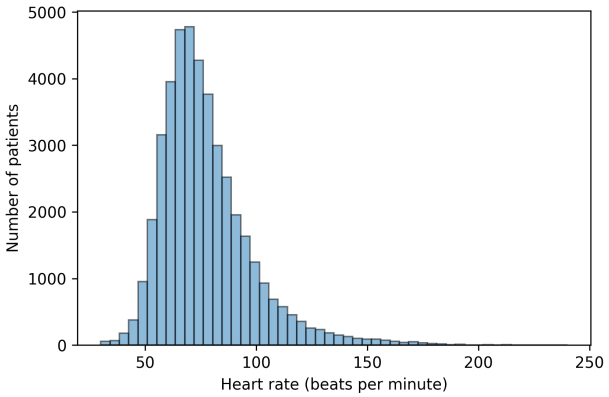

Figure 1: Histogram showing the distribution of the heart rates in the challenge training dataset after applying our peak detection pipeline.

2.3. Model

2.3.1. Model architecture and loss function

After detecting the peaks in each signal, we extract the distribution of the energy of the signal at each heartbeat over frequencies between 0 and 10 Hz using the discrete-time Fourier transform (DTFT). This information is concatenated with the patient context features. We multiply these vectors with an embedding matrix to generate a multivariate time-series represented as embedding vectors. After that, we feed the vector embeddings into a bidirectional Long Short Term Memory (LSTM) network. We apply an

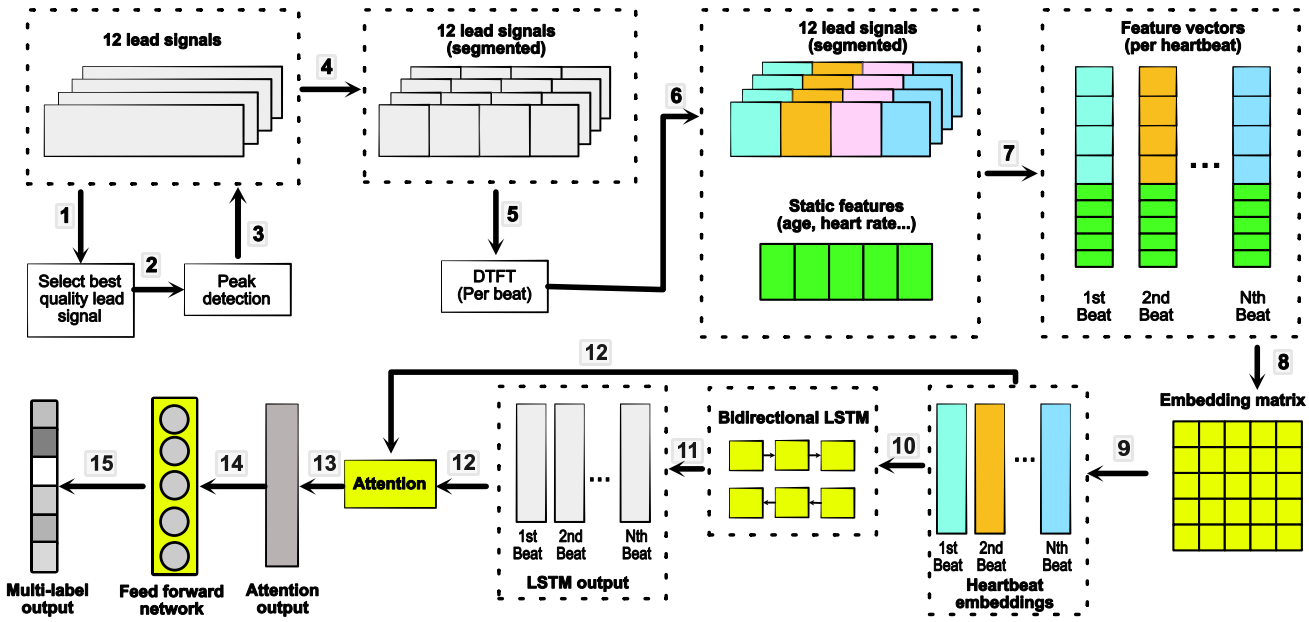

Figure 2: Proposed model architecture. We first detect the peaks in the given signals and then apply a discrete-time Fourier transform at the heartbeat level (1-6). We then generate per heartbeat input feature vectors, which are then concatenated with the static features before they are multiplied by the embedding matrix to generate heartbeat embedding vectors (7-9). The embedding vectors are fed into a bidirectional LSTM (10). The outputs of the bidirectional LSTM and the embedding vectors are fed into an attention mechanism whose fixed-length output is inputted to a final feed-forward network that outputs class probabilities after applying the sigmoid function for each class output (multi-label classification) (11-15).

attention mechanism [6] to the embedding vectors using the outputs of the bidirectional LSTM as in [7]. The resulting output vector from the attention layer is then fed into a final linear layer. Since we have a multi-label prediction task, we apply a sigmoid operator to the final layer's output to generate a probability score for each class output. The model was then optimized using the binary cross-entropy loss averaged over all the 24 class outputs. The overall model architecture is shown in Figure 2.

2.3.2. Model hyper-parameters and tuning

We fix the input embedding dimension (LSTM input dimension) to 512, the LSTM hidden dimension and all subsequent hidden layer dimensions to 256, the batch size to 64, and the number of training epochs to 10.

As for other learning parameters, we tune the learning rate, gradient clip norm, L1 and L2 regularization coefficients (λ_1 and λ_2), and dropout rate. We tune the model hyper-parameters using the Bayesian optimization implementation from the Hyperopt python package [8]. We use 40 total trials after separating the training dataset into 70% training and 30% validation to tune these parameters. The parameters and their ranges are shown in Table 2. We then selected the parameters from the trial giving the best performance on the validation set. After that, the model was retrained on the full dataset and submitted to the challenge for final evaluation after tuning the classification thresholds.

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 2: Search spaces for the model hyper-parameters.

2.3.3. Model thresholds tuning

After training and selecting a model that minimizes the loss and generalizes well to the validation set, we need to choose thresholds for each class in a way that optimizes for the challenge metric. Usually, this is an easier task 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- Grid search based threshold optimization is intractable given the number of classes (24).

Given this, we optimize 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 heuristic to accelerate the threshold search. We limit the threshold search to be within the 95 middle percentiles of each positive class's model probability output distribution across the training set. The reasoning behind this range is that higher values mean omitting that class entirely (all negative predictions). On the other hand, 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

Table 3 shows the final results of our submitted model. We used one of the ten submissions in the official phase of the challenge. The submitted model was not ranked due to omissions in the submission (team name: SBU_AI). Table 4 shows the results on each of the final test set databases.

Dataset	Score
Validation	0.416
Full test set	0.024

Table 3: Results on the challenge validation and full test set.

Test set database	Score
database 1	0.513
database 2	0.016
database 2	-0.028

Table 4: Results on the challenge full test set by database.

4. Discussion and Conclusions

In this work we designed a model to classify 12-lead ECGs using attention and DTFT. Tables 3 and 4 show that the model failed to generalize across different databases. This can be attributed to multiple design choices such as cross-validating after aggregating all data-points instead of cross-validating at a database level, limiting the maximum duration to 10 seconds per signal, and following an outcome agnostic approach. Addressing these limitations will be the subject of our future work. We also intend to analyze our model's attention weights to give more insight about each heartbeat'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. Physiological Measurement 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 Advances in Neural Information Processing Systems 29. Curran Associates, Inc., 2016; 3504–3512.

[8] Bergstra J, Yamins D, Cox D. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In International Conference on Machine Learning. 2013; 115–123.

Address for correspondence:

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