Classification of 12 Lead ECG Signal Using 1D-Convolutional Neural Network With Class Dependent Threshold

Rohit Pardasani[1,*] and Navchetan Awasthi[2,*]

[1] General Electric Healthcare, Bangalore, Karnataka, India
[2] Massachusetts General Hospital, Harvard University, Boston, Massachusetts, USA

Abstract

The goal of the proposed work is to classify the ECG signals into 24 different classes using the data obtained from the 12 Lead ECG signal. As part of the PhysioNet/Computing in Cardiology Challenge 2020, an approach based on 1 Dimensional - Convolutional Neural Network (1D-CNN) with class dependent threshold was developed for identifying cardiac abnormalities from 12-lead electrocardiogram (ECG). The method uses 1D-CNNs stacked in parallel with each CNN tuned to identify one of the classes. Each of these CNNs have same architecture comprising of convolutional layers, batch normalizations, activation layers and a dense layer with added regularizations and dropouts. The class dependent threshold gives the benefit of optimizing the model for each of the class individually without the need of training separate models for each category. This property of the network makes it ideal for real time setup where one inference run of this model is sufficient for multi-label and multi-class classification. The class dependent thresholds were chosen based on the ROC curve for each of the class respectively. Our approach achieved a challenge validation score of 0.342*, and full test score of* 0.077*, placing our team (AI Strollers) 32 out of 41 in the official ranking.*

1. Introduction

According to the World Health Organization, cardiovascular disease is one of the leading causes of mortality in the world [1]. Different cardiac diseases have different causes and symptoms, and are generally diagnosed by measurement of electrocardiogram (ECG) [2,3]. ECG diagnosis is considered as an important tool for screening and classification of various abnormalities. The PhysioNet/Computing in Cardiology Challenge 2020 approached this problem by automating the diagnosis and calling for open-source approaches used for classifying cardiac abnormalities from 12-lead ECGs [4,5]. Our best entry in the Challenge[5] implemented 1D CNN based network with class dependent threshold for this task.

2. Method

The sections below describe in detail the approach used for the task. This involved pre-processing of data (Sec. 2.1), designing & training 1D CNN model (Sec. 2.2) and finally classifying the samples (Sec. 2.3).

2.1. Preprocessing

1. The sampling rate of all signals in the dataset is same (500 Hz), hence time resampling of data was not required.
2. The signals in the dataset have different amplitude resolutions viz. 200/mV, 4880/mV, 1000/mV. Each signal was converted to physical units (mV) by dividing it with respective analog-to-digital gain before feeding it as input to the model.
3. The number of time samples for ECG signals in the dataset vary from 2500 to 462600. It is important to homogenize the size of input data before consuming it in a model. The mean length of signals was computed and was found to be $\sim$7700. Approximately 95$\%$ of the signals have length less than 10000. Hence, the signals that have length greater than 10000 were trimmed and those with size less than 10000 were 'wrap' padded at the end (signal was repeated till its vector length reached 10000).

2.2. Model Architecture

According to the scoring guidelines, there are only 27 diagnosis that are scored while the others are ignored in the challenge metric [5]. It seems prudent to focus only on these 27 diagnosis as this will lead to simpler model, compared to one that recognizes all 111 diagnosis. Out of these 27 diagnosis, 3 pairs of diagnosis are scored equally, hence these were combined pairwise to give a single diagnosis label for the pair. The following diagnosis have same scoring:

- CRBBB and RBBB (majority)

[*] Equal contribution

ISSN: 2325-887X DOI: 10.22489/CinC.2020.277

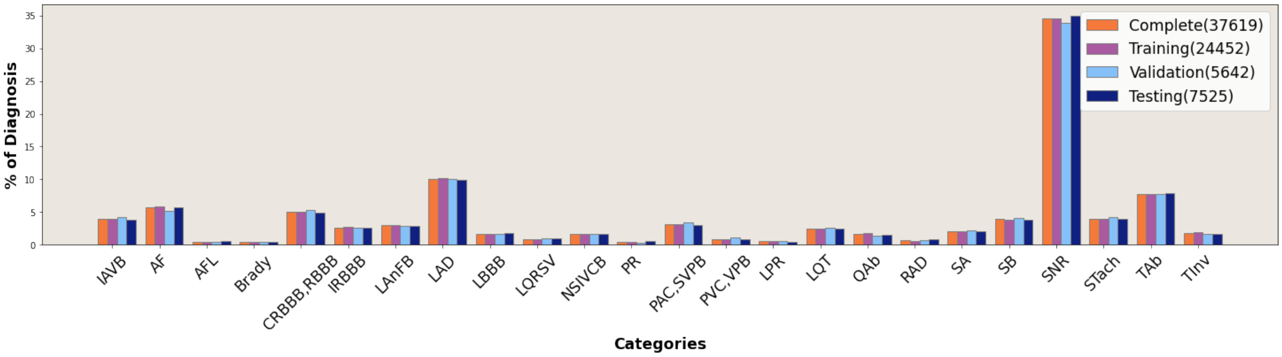

Figure 1. Proportion of diagnosis (in %) for each of the 24 classes (27 mapped to 24) with respect to total diagnosis in each subset created from the annotated data. Total number of samples in each set is mentioned in legend.

- SVPB and PAC (majority)
- PVC and VPB (majority)

For each of the diagnosis pairs that are identically scored, 'majority' tag in above list indicates the class that is relatively in majority with respect to other class. Since both classes in each pair are considered equivalent, sample from any of the class in the pair is tagged with the majority class label. So, CRBBB diagnosis is tagged as RBBB, SVPB as PAC and PVC as VPB. Thus, our modelling is further simplified, now model needs to distinguish only among 24 diagnosis. These 24 categories are depicted on x-axis of the bar plot in Fig. 1.

After the above process, diagnosis labels of each sample were converted into a 24 dimension binary vector (using one hot encoding, allowing vectors with multiples 1s for samples that have multi-class labels). This 24 dimension vector corresponding to each sample becomes our target for training the model.

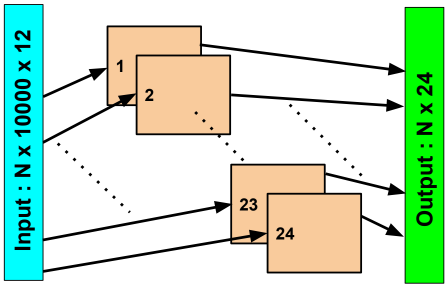

Figure 2. The architecture involves stacking 24 identical networks in parallel, each model receives same input and returns a scalar value from 0 to 1. The final output is obtained by concatenating these scalars into a 24 dimension vector. Here, each of the 24 boxes shown are models having architecture as depicted in Fig.3.

The annotated dataset contains a total of 43101 samples while only 37619 samples have diagnosis that belong to one or more of the 27 scored classes (now mapped to 24 classes). The samples that do not have any diagnosis belonging to these 24 categories were removed for further analysis. The aim is to reduce the complexity of model by skipping learning of samples that do not belong to the scoring category. Thus, 5482(=43101-37619) samples were removed completely from the dataset and these samples were not involved in our training, validation and test set. After removing these 5482 samples, the data was randomly split into training, validation and testing. The number of samples in each of these sets is mentioned in Fig. 1. It is important to mention again that this split of annotated data for training, validating, testing is at our end and it is not related to the test datasets that was used by challenge organizers for leaderboard scores. Since the above sets have been created randomly, a check needs to be done to make sure whether all these sets represent the population distri-

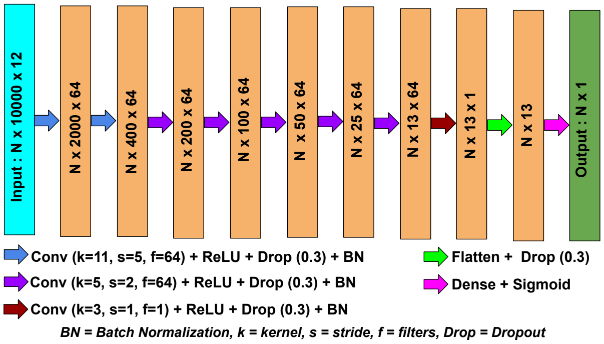
Input : N x 10000 x 12
N x 2000 x 64
N x 400 x 64
N x 200 x 64
N x 100 x 64
N x 50 x 64
N x 25 x 64
N x 13 x 64
N x 13 x 1
N x 13
Output : N x 1
Conv (k=11, s=5, f=64) + ReLU + Drop (0.3) + BN
Conv (k=5, s=2, f=64) + ReLU + Drop (0.3) + BN
Conv (k=3, s=1, f=1) + ReLU + Drop (0.3) + BN
Flatten + Drop (0.3)
Dense + Sigmoid
BN = Batch Normalization, k = kernel, s = stride, f = filters, Drop = Dropout

Figure 3. CNN Architecture for multivariate time series classification model that is replicated 24 times in parallel to obtain the final model. Twelve input channels correspond to the 12 leads of the ECG signal and the length of input signal is 10000.

Table 1. Metrics by class on challenge validation set along with class dependent thresholds applied to model output for deciding diagnosis. AUROC: Area Under Receiver Operating Characteristic, AUPRC: Area Under Precision-Recall Curve.

SNOMED CT Code	Abbreviation	Threshold	AUROC	AUPRC	F-measure
270492004	IAVB	0.6445	0.603	0.107	0.155
164889003	AF	0.4137	0.829	0.418	0.288
164890007	AFL	0.5644	0.519	0.018	0.034
426627000	Brady	0.5307	0.919	0.001	0.000
713427006, 59118001	CRBBB, RBBB	0.7846	0.769	0.197	0.326
713426002	IRBBB	0.4686	0.581	0.037	0.000
445118002	LAnFB	0.3369	0.432	0.051	0.162
39732003	LAD	0.3915	0.852	0.374	0.486
164909002	LBBB	0.3692	0.936	0.623	0.352
251146004	LQRSV	0.6249	0.704	0.059	0.067
698252002	NSIVCB	0.3655	0.591	0.019	0.030
10370003	PR	0.2081	0.455	0.001	0.000
284470004,63593006	PAC, SVPB	0.6115	0.513	0.071	0.129
427172004,17338001	PVC, VPB	0.6526	0.497	0.026	0.055
164947007	LPR	0.2325	-	-	-
111975006	LQT	0.7803	0.562	0.126	0.177
164917005	QAb	0.4344	0.687	0.062	0.072
47665007	RAD	0.2594	0.613	0.009	0.011
427393009	SA	0.3053	0.620	0.048	0.072
426177001	SB	0.7778	0.623	0.165	0.266
426783006	SNR	0.5129	0.799	0.394	0.359
427084000	STach	0.4635	0.827	0.265	0.287
164934002	TAb	0.6148	0.600	0.205	0.325
59931005	TInv	0.6331	0.538	0.072	0.127

bution w.r.t. diagnosis. Also, it should be ensured that there are sufficient samples of minority classes in all the sets. So, proportion analysis of complete, training, validation and testing set was done before utilizing the data. The percentage of diagnosis was calculated for each of the 24 classes in a set w.r.t. the total number of diagnosis present in the set. The result of this analysis is shown in the form of bar plot in Fig. 1. It could be inferred from the graph that distribution of diagnosis in subsets is roughly same as that in population.

After creating subsets for training, validation, testing and deciding on representation of target classes, a CNN model was planned accordingly. We decided to use a multi-path network, consisting of 24 paths, as base architecture of the model. Each path in such a model must correspond to a different class of diagnosis. This structure consisting of 24 models (one for each diagnosis) stacked in parallel, is shown in Fig. 2. Each of these models will receive input from all 12 channels (truncated or padded till 10000 samples) and will output a single scalar value ranging from 0 to 1. This value indicates the probability of a sample belonging to diagnosis mapped by the model. The outputs of all the 24 models are concatenated to obtain a 24 dimension probability vector. The target for this probability vector is 24 dimension one hot encoded vector (created from diagnosis labels as explained earlier).

The architecture of each of these 24 models is exactly same and is shown in Fig. 3. The model was trained with cross entropy loss, learning rate of 1e-4 and Adam [6] as an optimizer. The different parameters used were β_1=0.99, β_1=0.9, ϵ=1e-7, decay=0.0001, amsgrad=False and clipvalue=0.5 (to restrict the exploding of the gradient). Kernel regularizers l_1(=1e-7) & l_2(=1e-7) and bias regularizer l_2(=1e-7) were used to reduce overfitting of the model[7]. The total number of parameters in the model are 3,800,520 which consists of 3,778,968 trainable parameters and 21,552 non-trainable parameters. The network was developed, trained, validated and tested using Keras[8] with Tensorflow[9] as backend. The model was trained for 100 epochs with batch size of 128. The model that gave best validation loss during the training was retained as the final model. All computations were carried out on a Linux workstation with Intel Xeon Silver 4110 CPU with 2.10 GHz clock speed, having 128 GB RAM and a TITAN RTX GPU with 24 GB memory. Each epoch took approximately 210 seconds on this machine.

Table 2. Results of classification for the CNN model on different sets of challenge test data

Dataset	AUROC	AUPRC	Accuracy	F-measure	Challenge metric
Validation	0.625	0.124	0.000	0.140	0.342
Test 1	0.783	0.437	0.001	0.117	0.212
Test 2	0.622	0.135	0.000	0.153	0.359
Test 3	0.629	0.197	0.000	0.144	0.096
Full Test	0.599	0.136	0.000	0.142	0.077

2.3. Classification using the Model

Once the model was trained, Receiver Operating Characteristic (ROC) and Area Under Curve (AUC) were computed on the training set for each of the 24 classes. Using the ROC, an optimal threshold was found for each class, this threshold maximized the difference between True Positive Rate (TPR) and False Positive Rate (FPR) for that class. It is pertinent to mention that these thresholds were decided based on model output and labels of the training set. For inference, the optimal threshold for each class was used (after getting probabilities from model) to return the diagnosis for a sample. The optimal thresholds are given in Table-1 corresponding to each of the 24 ECG diagnosis. Thus, our deep learning based auto-diagnosis model has two parts viz. a trained deep learning model and a table containing optimal threshold for each class.

3. Results

Various metrics on challenge test sets were calculated on the results obtained using the final model. The corresponding results for AUROC, AUPRC, F-measure and challenge metric are given in Table-1 and Table-2. Our model gave 0.342 and 0.077 as the challenge metric value on validation set and full test set. On these scores, our team (AI Strollers) was ranked 32 out of 41 in the official ranking.

4. Discussion and Conclusions

According to score and ranking on leaderboard, there seems plenty of scope for improvement in model. There are some experiments with respect to pre-processing, hyper-parameter tuning, increasing epochs etc. that could have improved the results but were not performed due to time constraint. In conclusion, this approach of having 1D-CNNs in parallel with class dependent threshold, can be termed as a ‘start in the right direction to accomplish the goal of automating the ECG diagnosis’.

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 (Under Review);.

[6] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint 2014;URL https://arxiv.org/abs/1412.6980v9.

[7] Bisong E. Regularization for deep learning. In Building Machine Learning and Deep Learning Models on Google Cloud Platform. Springer, 2019; 415–421.

[8] Chollet F, et al. Keras: The python deep learning library. ascl 2018;ascl–1806.

[9] Abadi M, Agarwal A, Barham P, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint 2016;URL http://arxiv.org/abs/1603.04467.

Address for correspondence:

Rohit Pardasani
GE Healthcare,
2nd Floor-Odyssey Building, JFWTC, Bangalore, India.
rohit.r.pardasani@gmail.com