Deep Multi-Label Multi-Instance Classification on 12-Lead ECG

Yingjing Feng[1,2], Edward Vigmond[1,2]

[1] IHU Liryc, Electrophysiology and Heart Modeling Institute, fondation Bordeaux Université, Pessac-Bordeaux, France
[2] Univ Bordeaux, IMB, UMR 5251, Talence, France

Abstract

As part of the PhysioNet/Computing in Cardiology Challenge 2020, we developed an end-to-end deep neural network model based on 1D ResNet and an attention-based multi-instance classification (MIC) mechanism, named as MIC-ResNet, requiring minimal signal preprocessing, for identifying 27 cardiac abnormalities from 12-lead ECG data. Our team, ECGLearner, achieved a challenge validation score of 0.486 *and a full test score of* 0.001*, placing us 33 out of 41 in the official ranking of this year's challenge.*

1. Introduction

Cardiovascular diseases are the primary cause of death, and they greatly impact daily life across all demographics. The ECG signal is a common and important screening and diagnostic tool for heart conditions. Given past examples of ECGs and annotations, deep neural networks can perform a supervised learning to learn features of different conditions directly from ECGs for diagnosis.

Following the PhysioNet/Computing in Cardiology Challenge 2020 which promoted automated and open-source approaches for classifying multiple cardiac abnormalities from 12-lead ECG [1, 2], we developed a novel open-source deep learning model called MIC-ResNet, which combines ResNet [3] for time series and multi-instance classification (MIC) to classify multi-center patient ECG for 27 different conditions. The code is available at `https://github.com/SeffyVon/ECG_MICResNet`.

2. Methods

As shown in Figure 1, our MIC-ResNet comprises three major components: an encoder module based on 1D ResNet; a MIC module; and a decoder module to produce an output of 27 classes going through a sigmoid function. The definitions for the acronyms of the abnormalities are listed in [2].

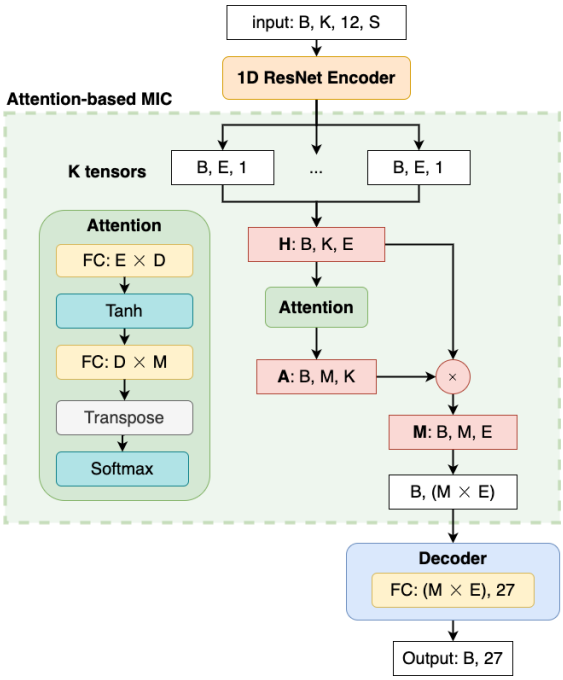

Figure 1: Architecture of MIC-ResNet.

The only preprocessing step that we performed was to filter the ECG by applying a fourth-order Butterworth filter with a passband of 0.5 to 50Hz for each lead of each patient signal. We did not normalize the signal, as we believed that preserving amplitude of the raw ECG signal was important for some conditions such as low QRS voltages.

2.1. 1D ResNet Encoder

ResNet [3] is the state-of-the-art deep network for multiple types of data, from images to time series [4, 5], and has been successfully applied to cardiac abnormality detection using ECGs, such as [6]. It benefits from a *shortcut* module, which enables the network to go deep, whilst remaining relatively low in complexity, thereby making the learning easier.

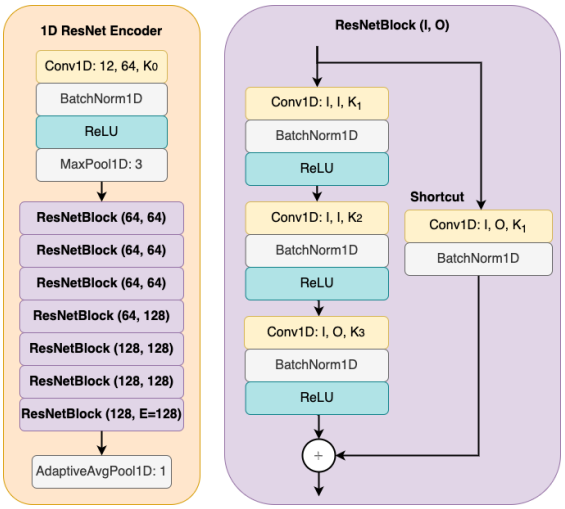

Figure 2: Encoder module in the MIC-ResNet.

We used a customized 1D ResNet as an encoder (in Figure 2) to transform a 12-channel ECG segment of S samples, to a lower-dimensional (E-dimensional) embedding, adapted from the original 2D ResNet [3]. The encoder was composed of a 1D convolutional layer (*Conv1D*) which took I, O as the input and output channel sizes, and K_0 as the kernel size, a batch normalization function (*BatchNorm1D*) [7], a non-linear activation function (*ReLU*) [8], a max pooling function (*MaxPool1D*) with a kernel size of P_0, and then followed by three building blocks (*ResNetBlocks*), where each block took an input signal with I input channels and produced O output channels. The kernel sizes $K_1, K_2,$ and K_3 for the three *Conv1D* layers in the *ResNetBlocks* came from a strong baseline 1D ResNet model for time-series classification [5]. An adaptive average pooling (*AdaptiveAvgPool1D*) with output length of 1 was placed at the end, to automatically select the stride and the kernel size in order to produce E outputs of length-one channels.

2.2. Attention-based MIC

MIC refers to a type of classification problems in which the data samples are instances in bags, and a label is only available for each bag rather than for each instance. Consider a patient ECG recording as segments of a smaller fixed length, for conditions that do not occur in each heartbeat, such as PAC, only some segments are positive for PAC. MIC "pools" the instance probabilities (or labels) to compose the probability (or label) of the bag. By using a bag of K segments of S samples to represent an ECG recording of various lengths, we have spanned our search range of the ECG from K samples to $K \times S$ samples with the same encoder, without assuming that each fixed-length segment was positive.

We adopted an attention-based MIC framework proposed in [9]. For a bag of K instances going through the encoder, we obtained K embeddings as $H = \{h_1, h_2, \ldots, h_K\}$. The MIC pooling was then

$$z = \sum_{k=1}^{K} a_k h_k \tag{1}$$

where

$$a_k = \frac{\exp\{\mathbf{w}^T \tanh \mathbf{V} h_k^T\}}{\sum_{j=1}^{K} \exp\{\mathbf{w}^T \tanh \mathbf{V} h_j^T\}} \tag{2}$$

The attention module was made of two fully connected (*FC*) layers with a $\tanh(\cdot)$ layer in the middle, where the first *FC* layer was to learn the weight $\mathbf{V} \in \mathcal{R}^{D \times E}$, and the second to learn the weight $\mathbf{w} \in \mathcal{R}^{D \times M}$, together with the transpose operation and the *Softmax* layer, implemented Eq (2), and K was another hyperparameter to be optimized. A tensor multiplier $M = A \times H$ implemented Eq (1), and the resulting M went through a *FC* decoder to produce an output of C dimensions.

2.3. Implementation

As all patient ECGs were sampled at 500Hz, each containing a varying number of at least 2500 samples, we picked $S = 3000$ samples representing 6 second intervals as the training input to the ResNet 1D, so the bag input had dimensions of $(B, K, 12, S)$. Zeros were padded on the end of recordings with less than S samples. To augment the training set, we randomly sampled K instances of S samples for a training input across the whole ECG recording, whereas a validation input was composed of evenly sampled K instances of S samples.

We represented the label of each sample as $\mathbf{y} = [y_1, y_2, \ldots, y_C]$, where $C = 27$ is the total number of scored classes, and $y_i = 1$ if class i is positive and 0 otherwise. For those classes with an equivalent class, we relabelled them as positive in a data entry if their equivalent class was positive in the same entry. A multi-label stratified 5-fold cross-validation [10] (using iterative-stratification Python package version 0.1.6) was applied on each of the six training datasets in [2] to constitute the full training-validation set so that the training and the validation sets in each fold have similar class distributions. The class distribution is also similar across different folds, keeping performance stable between different folds.

A binary cross entropy loss (*BCELoss*) was used as the optimization target for the multi-label classification. The total *BCELoss* was defined as the average of sample *BCELoss*, and for each sample of the network output

Hyperparameters	Value
Segment length (S)	3000
Number of segments in a bag (B)	5
Positive class weight (p)	2
Encoder first Conv1D and MaxPool1D (K_0)	7
Encoder first MaxPool1D kernel (P_0)	3
ResNetBlock kernels (K_1, K_2, K_3)	7, 5, 3
ResNetBlock input output channels	see Figure 2
Parameter for attention (D, M)	64, 32
Smoothing term (γ)	1

Table 1: List of hyperparameters.

$\mathbf{x} = [x_1, x_2, \ldots, x_C]$, the *BCELoss* of each sample was:

$$l = -\sum_{i=1}^{C} w_i \left[p \cdot y_i \cdot \log \sigma(x_i) + (1 - y_i) \cdot \log(1 - \sigma(x_i))\right]$$

where $\sigma(\cdot)$ is the sigmoid function. To consider for class imbalance, the class weight for each class i was defined as in [11]:

$$w_i = \log \frac{N - n_i + \gamma}{n_i + \gamma}, N = \sum_{j=1}^{C} n_j,$$

where i is the number of positive instances of class i, and γ is a smoothing term, a larger class weight was given to classes with small samples, and got optimized at a higher priority. A positive weight $p = 2$ was added to the *BCELoss* for all classes to give a higher weight for recall than precision. We applied the sigmoid function to the network outputs to obtain the predicted probability for each class, and used 0.5 as a threshold for the binary label. All hyperparameters in our method are summarized in Table 1.

We used the Adam optimizer [12] with a learning rate of 0.01, and rescaled with a factor of 0.1 when the validation loss reached a plateau for 10 epoch. We used mini-batch gradient descent with a batch size of 64. The training stopped when there was no reduction in the validation loss for over 20 epochs. The network was trained in PyTorch version 1.4, CUDA version 10.2 on a Quadro RTX 8000 Graphic Processing Unit. Each fold stopped at around 55 epochs and three hours. We averaged the validation losses across five folds, and computed the optimal epoch producing the lowest averaged validation loss, and then trained the network on the whole dataset for this optimal epoch.

3. Results

We compared our results with using instance-wise 1D ResNet composed of only the encoder and decoder in Figure 1. During training, only one segment of S samples was drawn randomly from ECG for each training entry and one central segment of S samples for each validation entry during training. During validation, we used the same K instances as in MIC and made predictions for the ECG on two modes: the *First* mode used the output of the first instance, and the *Max* mode used the maximal amongst the K instances. The average competition metrics on the training-validation set across five folds are shown in Table. 2, and were broken down into different classes in Figure 3. Figure 4 showed a multi-label confusion matrix on the training-validation set across five folds, where its diagonal holds the true-positive (TP) rate for each class j, defined as $N_{TP}(j)/N(j)$, and the rest shows the false-negative (FN) rate of a class j, defined as $N_{FN}(j \to k)/N(j)$, where $N_{TP}(j)$ and $N(j)$ are the number of TP entries and the total number of entries of class j, respectively, and $N_{FN}(j \to k)$ is the number of entries where class j was a FN and was classified as class k, but the ground truth of class k are negative in that entry.

On the hidden challenge datasets, our classifier, ECGLearner, received a validation score of 0.486 and a test score of 0.001, with scores of 0.669, 0.452 and -0.347 in the three test databases, respectively.

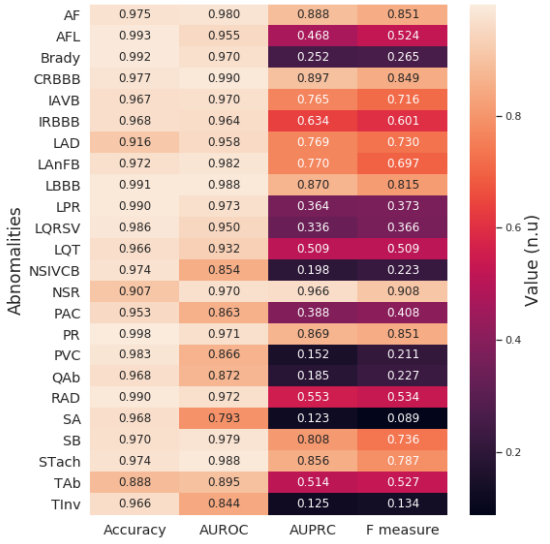

Figure 3: Per-class challenge metrics over five folds of the training-validation set.

4. Discussion and Conclusions

We developed a multi-label classifier for 12-lead ECGs with an attention-based MIC. In Table 2, MIC showed achieved the best scores overall on the training-validation set. Although the *Max* mode received the best challenge score, it biased towards recall. By aggressively selecting the maximal probability, *Max* resulted in a low precision

	Accuracy	AUROC	AUPRC	F measure	Challenge Metric
MIC	$\mathbf{0.936 \pm 0.002}$	$\mathbf{0.551 \pm 0.003}$	0.523 ± 0.010	$\mathbf{0.538 \pm 0.004}$	0.539 ± 0.014
First	0.932 ± 0.002	0.544 ± 0.007	$\mathbf{0.528 \pm 0.006}$	0.519 ± 0.001	0.517 ± 0.004
Max	0.935 ± 0.002	0.550 ± 0.006	0.494 ± 0.012	0.529 ± 0.005	$\mathbf{0.556 \pm 0.014}$

Table 2: Means and standard deviations of the challenge metrics over five folds of the training-validation set.

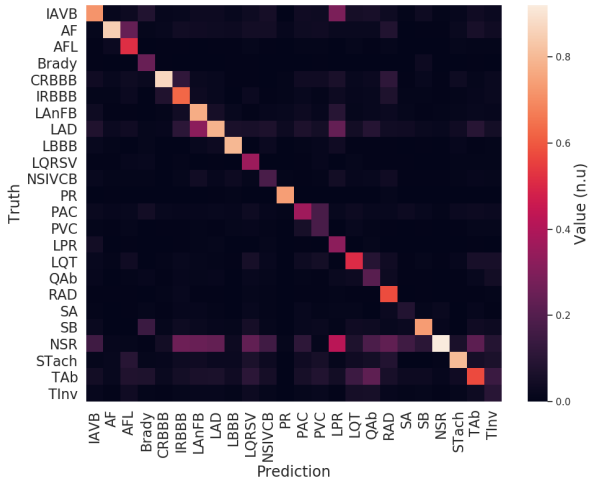

Figure 4: Multi-label confusion matrix on the training-validation set, where the diagonal shows the TP rate and the rest shows the FN rate of classes.

for auprc and F_1, whereas *First* received the best auprc but the worst challenge score. On the other hand, MIC took a balance between precision and recall, and we believe this is important. In Figure 3, the classifier achieved an accuracy close to 1 and auroc ≥ 0.8 in practically all cases. The F-measure score shows that our model worked the best for AF, CRBBB, NSR, and PR, which are all conditions exhibiting abnormality in each beat, whereas the worst were mainly conditions that did not occur in each beat (e.g. PVC), or had abnormal amplitudes (e.g. Tinv), duration (e.g. Brady), and abnormalities with multiple underlying causes (eg. NSIVCB and QAb). In Figure 4, the inter-class misclassification occurred the most from NSR, followed by from TAb to QAb, between PAC and PVC, and a few abnormalities were mistaken as LPR and TInv.

In conclusions, we developed an open-source deep neural network combining 1D ResNet with attention-based MIC to predict for multiple cardiac abnormalities from 12-lead ECG, receiving a validation challenge score of 0.486 and a test score of 0.001 for this year's challenge.

Acknowledgments

Funding has been received from the European Union Horizon 2020 research and Innovation programme "Personalised In-silico Cardiology (PIC)" under the Marie Sklodowska-Curie grant agreement No 764738, and the French National Research Agency (ANR-10-IAHU-04).

References

[1] Goldberger, et al. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation 2000;101(23):e215–e220.

[2] Perez Alday EA, et al. Classification of 12-lead ECGs: the Physionet/Computing in Cardiology Challenge 2020. Physiological Measurement 2020;(In Press).

[3] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016; 770–778.

[4] Wang Z, Yan W, Oates T. Time series classification from scratch with deep neural networks: A strong baseline. In 2017 International Joint Conference on Neural Networks, IJCNN. 2017; 1578–1585.

[5] Ismail Fawaz H, Forestier G, Weber J, Idoumghar L, Muller PA. Deep learning for time series classification: A review. Data Mining and Knowledge Discovery July 2019; 33(4):917–963.

[6] Xiong Z, Nash MP, Cheng E, Fedorov VV, Stiles MK, Zhao J. ECG signal classification for the detection of cardiac arrhythmias using a convolutional recurrent neural network. Physiol Meas Sept. 2018;39(9):094006.

[7] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In 32nd International Conference on Machine Learning, ICML 2015, volume 1. 2015; 448–456.

[8] Agarap AF. Deep Learning using Rectified Linear Units (ReLU). arXiv preprint arXiv180308375 Mar. 2018;.

[9] Ilse M, Tomczak J, Welling M. Attention-based deep multiple instance learning. In Proceedings of Machine Learning Research, volume 80. PMLR, July 2018; 2127–2136.

[10] Sechidis K, Tsoumakas G, Vlahavas I. On the Stratification of Multi-Label Data. In Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part III, ECML PKDD'11. Springer-Verlag, 2011; 145–158.

[11] Li X, Sun X, Meng Y, Liang J, Wu F, Li J. Dice Loss for Data-imbalanced NLP Tasks. arXiv preprint arXiv191102855 Nov. 2019;.

[12] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. In 3rd International Conference on Learning Representations, ICLR 2015. 2015; .

Address for correspondence:

Yingjing Feng
IHU Liryc, F-33600 Pessac-Bordeaux, France
yingjing.feng@ihu-liryc.fr