Bag of Tricks for Electrocardiogram Classification With Deep Neural Networks

Seonwoo Min[1], Hyun-Soo Choi[2], Hyeongrok Han[1], Minji Seo[1], Jin-Kook Kim[3], Junsang Park[3], Sunghoon Jung[3], Il-Young Oh[4], Byunghan Lee[5], Sungroh Yoon[1, 6]

[1]Department of Electrical and Computer engineering, Seoul National University, Seoul, South Korea
[2]T3K, SK Telecom, Seoul, South Korea
[3]HUINNO Co., Ltd., Seoul, South Korea
[4]Division of Cardiology, Department of Internal Medicine, Seoul National University Bundang Hospital, Seongnam, South Korea
[5]Department of Electronic and IT Media Engineering, Seoul National University of Science and Technology, Seoul, South Korea
[6]Department of Biological Sciences, Interdisciplinary Program in Bioinformatics, Interdisciplinary Program in Artificial Intelligence, ASRI, INMC, and Institute of Engineering Research, Seoul National University, Seoul, South Korea

Abstract

Recent algorithmic advances in electrocardiogram (ECG) classification are largely contributed to deep learning. However, these methods are still based on a relatively straightforward application of deep neural networks (DNNs), which leaves incredible room for improvement. In this paper, as part of the PhysioNet / Computing in Cardiology Challenge 2020, we developed an 18-layer residual convolutional neural network to classify clinical cardiac abnormalities from 12-lead ECG recordings. We focused on examining a collection of data pre-processing, model architecture, training, and post-training procedure refinements for DNN-based ECG classification. We showed that by combining these refinements, we can improve the classification performance significantly. Our team, DSAIL_SNU, obtained a 0.695 challenge score using 10-fold cross-validation, and a 0.420 challenge score on the full test data, placing us 6th in the official ranking.

1. Introduction

Electrocardiogram (ECG) is a commonly used non-invasive diagnostic tool that records the electrical activity of the heart. The standard 12-lead ECG is pivotal for detecting a wide spectrum of cardiac abnormalities such as atrial fibrillations [1]. Computer-aided classification of ECG has become more significant for automated ECG interpretation. However, substantial misdiagnosis rates of classical algorithms are often a serious issue in the everyday practice of clinical medicine [2].

Building upon the success of deep learning, recent algorithmic advances in ECG classification are largely contributed to deep neural networks (DNNs). Previous works have used DNNs for single-lead and 12-lead ECGs and demonstrated high diagnostic performance similar to that of cardiologists [3,4]. Nevertheless, these methods are still based on a relatively straightforward application of DNNs. Training procedure refinements, such as the changes in loss functions, have not been studied thoroughly, which leaves incredible room for expansion and innovation [5].

In this paper, as part of the PhysioNet / Computing in Cardiology Challenge 2020 [6], we developed an 18-layer residual convolutional neural network to classify clinical cardiac abnormalities from 12-lead ECGs. We focused on examining a collection of data pre-processing, model architecture, training, and post-training procedure refinements for DNN-based ECG classification. They introduce small modifications that barely change computational complexity. However, our empirical evaluations showed that combining these refinements can lead to significant and consistent performance improvement.

2. Methods

We first define our baseline experiment setup, and then present a collection of refinements. We used a wide residual network (WRN) similar to the model used in the previous work [4]. Each model was implemented using PyTorch [7] and trained for 100 epochs on an NVIDIA V100 GPU. We used binary cross-entropy (BCE) loss, the Adam optimizer, a batch size of 128, a learning rate of 0.001, L2 weight decay of 0.0005, and a dropout probability of 0.3.

Dataset	Number of Recordings	w/ Scored Labels	Length (Seconds)
PTB-XL	21,837	21,604	10
Georgia	10,344	9,458	9
CPSC	6,877	5,279	15
CPSC-Extra	3,453	1,278	15
PTB	516	97	110
StPetersburg	74	33	1,800

Table 1. Data statistics.

2.1. Data Filtering and Split

PhysioNet / Computing in Cardiology Challenge 2020 provides 43,101 12-lead ECGs with 27 scored SNOMED-CT labels [8] from 6 datasets (Table 1). To the best of our knowledge, this is the first public competition to focus on a realistic clinical using multiple heterogeneous sources.

For the ease of training DNNs, we used the following procedures for the data filtering and split. First, we excluded the two datasets (*i.e.*, PTB and StPetersburg) with long average lengths. Second, we removed the recordings without any positive scored labels. Finally, the remained 37,619 recordings were split into 10-folds for cross-validation. We used iterative stratification to maintain distributions of positive examples of each label [9].

2.2. Data Pre-processing

The ECG recordings should naturally be diverse due to the differences in individuals and data acquisition environments. To improve the quality of the data, we used two pre-processing refinements. First, we used 50Hz and 60Hz notch filters to remove the external electrical noises. Then, we used a scaler to standardize each recording by removing the mean and scaling to unit variance. If necessary, recordings were resampled to a 500 Hz sampling rate.

2.3. Model Architecture

We used WRN-*l*-*k* denoting a residual network with *l* convolution/dense layers and a widening factor *k* [10]. It consists of an input stem, four stages of *N* residual blocks, and an output stem (Figure 1). The widening factor *k* scales the number of convolutional filters in the model. We used WRN-10-1 for ablation studies and WRN-18-2 for the final model.

The residual blocks consist of two 11x1 pre-activation convolutions with batch normalization (BN) and a rectified linear unit (ReLU). Note that due to the BN-ReLU in the input stem, the pre-activation is skipped for the first layer of stage 1. We also applied dropout between the convolution layers after the pre-activation. The first layers of each stage perform down-sampling with a factor of 2.

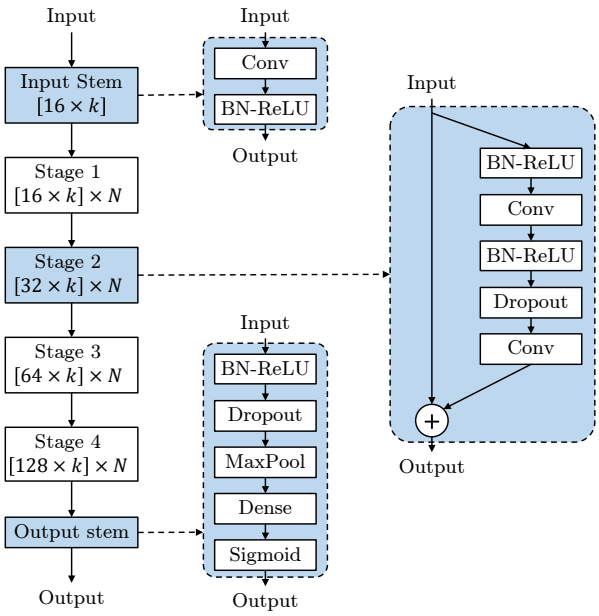

Figure 1. Model overview.

To efficiently handle variable-length ECGs, we used the following steps during both training and validation:

1. Divide each recording using a sliding window of 4,096 samples ($\approx$ 4 seconds) with overlaps of 512 samples ($\approx$ 1 second).
2. Feed-forward each window through the model up until its max-pooling layer in the output stem.
3. Concatenate the intermediate outputs from the windows and use the max-pooling layer to obtain a fixed-length vector from them.
4. Feed-forward the fixed-length vector into the last dense layer and the sigmoid activation function.

2.4. Training

Learning rate scheduler

Learning rate adjustment is critical for the training of DNNs. The widely used strategy is to start from the initial learning rate and exponentially decrease it when the model has stopped improving. In contrast, we used a learning rate scheduler with warmup and cosine annealing strategy [5]. It linearly increases the learning rate from 0 to the initial learning rate η, then decreases it to 0 by following the cosine function (Figure 2). The learning rate η_t at training epoch t is defined as:

$$\eta_t = \begin{cases} \frac{t}{T_w}\eta & \text{if } t <= T_w \\ \frac{1}{2}(1+\cos(\frac{(t-T_w)\pi}{T}))\eta & \text{otherwise} \end{cases}$$

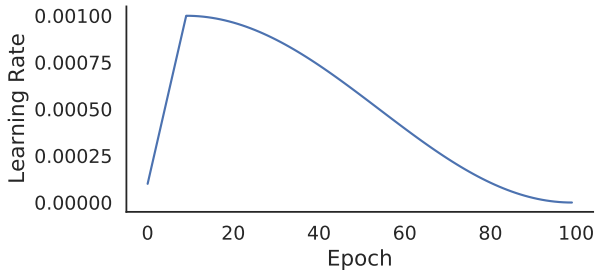

Figure 2. Learning rate scheduler.

where T and T_w are the number of total training epochs and warmup training epochs, respectively. In this work, we used $T = 100$, $T_w = 10$, and $\eta = 0.001$.

Confusion weighted binary cross-entropy loss

Using a training objective that more closely resembles an evaluation metric can lead to better performance. In this work, the target evaluation metric is a generalized accuracy called challenge score [6]. It is defined as:

$$s_{\text{unnormalized}} = \sum_{i=1}^{m} \sum_{j=1}^{m} w_{ij} a_{ij}$$

where $A = [a_{ij}]$ and $W = [w_{ij}]$ are a multi-class confusion matrix and a reward matrix, respectively. a_{ij} is the normalized number of recordings for output class c_i with a positive label c_j. w_{ij} is the reward for each entry defined by cardiologists based on the similarity of their risks or treatments. The final score is obtained by normalizing the score so that an inactive classifier receives a score of 0 and a ground-truth classifier receives a score of 1.

The conventional BCE loss does not take into account that some misdiagnoses are more harmful than others. Instead, we proposed a novel training objective to differentiate the misdiagnoses based on the reward matrix W. Formally, we defined confusion weighted binary cross-entropy (CoW-BCE) loss as:

$$\mathcal{L} = \sum_j (y_j \log(p_j) + cow_j (1 - y_j) \log(1 - p_j)),$$

$$cow_j = \frac{\sum_i y_i (1 - w_{ij})}{\sum_i y_i},$$

where $y_j \in \{\pm 1\}$ and $p_j \in [0, 1]$ specifies the label and the model's output for class c_j, respectively. Compared to the conventional BCE loss, it has an additional confusion weight constant cow_j. When an example is misclassified and w_{ij} is 0, the loss is unaffected and identical to the BCE loss. As $w_{ij} \to 1$, the constant goes to 0 and the loss for misdiagnoses with similar risks is down-weighted.

Manifold Mixup

The decision boundary of DNNs is often sharp and thus hurts its generalization performance. To alleviate the problem, we adopted Manifold Mixup [11]. The original Mixup constructs virtual training examples by weighted linear interpolations of both inputs and labels. In this work, we used Manifold Mixup to deal with variable-length recordings. Instead of the raw ECG inputs, we used weighted linear interpolations of hidden representations.

Formally, consider training a WRN $f(x) = o(g(x))$, where $h = g(x)$ denotes the front part of the model up to the max-pooling layer, and $o(h)$ denotes the rear part mapping the intermediate representations to the model outputs. Given two random examples (x_i, y_i) and (x_j, y_j), the mixed examples are generated as:

$$h_i = g(x_i), \quad h_j = g(x_j),$$
$$\hat{h} = \lambda h_i + (1 - \lambda) h_j,$$
$$\hat{y} = \lambda y_i + (1 - \lambda) y_j,$$

where the mixing coefficient $\lambda \sim Beta(0.2, 0.2)$. Then, the model continues the forward pass with the mixed intermediate representations. The model outputs $o(\hat{h})$ and mixed labels $\hat{y}$ are used to compute the training loss and train the entire model.

2.5. Post-training

Class-specific thresholds

After training, the model produces scalar prediction scores $p \in [0, 1]$ for each class. Since our target evaluation metric is computed on the predicted classes, a certain threshold must be applied to binarize the scores. The obvious choice would be applying the threshold of 0.5 for all the classes but it could be suboptimal.

To address the problem, we selected class-specific thresholds that perform best in the validation set. For each class, we independently computed the challenge scores for the validation set using 100 thresholds evenly spaced within 0 and 1. We selected the one with the highest challenge score.

Model ensemble

Ensemble learning methods combine multiple models to produce final predictions. Both theoretically and empirically, it has been shown that it usually yields higher predictive performance than the individual models [12].

We used an ensemble of 10 models from the 10-fold cross-validation. Since each model was trained with different composition of the training data, the diversity could lead to a more powerful ensemble model. We averaged both their scalar prediction scores and class-specific thresholds for the binarization of the scores.

Model	F2	G2	CS
WRN-10-1 (standard)	0.522	0.345	0.557
+ Filter & Scaler	0.534	0.356	0.568
+ LR scheduler	0.615	0.412	0.623
+ CoW-BCE loss	0.646	0.420	0.651
+ Manifold Mixup	0.654	0.423	0.658
+ Class-specific thresholds	0.679	0.426	0.676
WRN-18-2	**0.696**	**0.448**	**0.695**

Table 2. 10-fold cross-validation results.

Model	DB 1	DB 2	DB 3	Full Test Set
WRN-18-2	**0.688**	**0.654**	**0.228**	**0.420**

Table 3. Challenge scores on the test data.

3. Results

Table 2 presents the 10-fold cross-validation results for applying the refinements one-by-one. We report F2 and G2 scores as well as the challenge score (CS). Note that the evaluation of the model ensemble requires an additional holdout test dataset and could not be done in the cross-validation. By stacking each refinement, we have steadily improved the 12-lead ECG classification performance with DNNs. Our final model, WRN-18-2, trained with the collection of refinements obtained a 0.695 challenge score.

Table 3 presents the test results. Our final model, the ensemble of 10 WRN-18-2 models, obtained a 0.420 challenge score on the full test data, placing our team (DSAIL_SNU) 6th in the official ranking. For the three test databases (DB) from the full test data, we obtained 0.872, 0.654, and 0.228 challenge scores, respectively.

4. Concluding Remarks

In this work, we explored a collection of refinements to train DNNs for ECG classification. These refinements introduce small modifications to data pre-processing, model architecture, training, and post-training procedures. Stacking all of them together enabled significant performance improvement for identifying cardiac abnormalities.

As part of the PhysioNet / Computing in Cardiology Challenge 2020, the proposed method was evaluated in a realistic clinical setting using heterogeneous datasets and a new scoring metric. Thus, the improved results showed its potential taking it closer to the everyday practice of clinical medicine. We believe one important future work would be to compare its performance and confusion matrix with those of cardiologists. Through the analyses, we would be able to evaluate the model more objectively and better understand its strengths and weaknesses.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant (Ministry of Science and ICT, 2018R1A2B3001628), the HPC Support Project (NIPA), and the Brain Korea 21 Plus Project in 2020.

References

[1] Schlant RC, Adolph RJ, DiMarco J, et al. Guidelines for electrocardiography. Circulation 1992;85(3):1221–1228.

[2] Schläpfer J, Wellens HJ. Computer-interpreted electrocardiograms: benefits and limitations. Journal of the American College of Cardiology 2017;70(9):1183–1192.

[3] Pourbabaee B, Roshtkhari MJ, Khorasani K. Deep convolutional neural networks and learning ECG features for screening paroxysmal atrial fibrillation patients. IEEE Transactions on Systems Man and Cybernetics Systems 2018;48(12):2095–2104.

[4] Ribeiro AH, Ribeiro MH, Paixão GM, et al. Automatic diagnosis of the 12-lead ECG using a deep neural network. Nature communications 2020;11(1):1–9.

[5] He T, Zhang Z, Zhang H, et al. Bag of tricks for image classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019; 558–567.

[6] Perez Alday EA, Gu A, Shah A, et al. Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020. Physiological Measurement 2020;.

[7] Paszke A, Gross S, Chintala S, et al. Automatic differentiation in PyTorch. NIPS Autodiff Workshop 2017;.

[8] De Silva TS, MacDonald D, Paterson G, et al. Systematized nomenclature of medicine clinical terms to represent computed tomography procedures. Computer Methods and Programs in Biomedicine 2011;101(3):324–329.

[9] Sechidis K, Tsoumakas G, Vlahavas I. On the stratification of multi-label data. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 2011; 145–158.

[10] Zagoruyko S, Komodakis N. Wide residual networks. arXiv preprint arXiv160507146 2016;.

[11] Verma V, Lamb A, Beckham C, et al. Manifold mixup: Better representations by interpolating hidden states. In International Conference on Machine Learning. PMLR, 2019; 6438–6447.

[12] Kwon S, Bae H, Jo J, Yoon S. Comprehensive ensemble in QSAR prediction for drug discovery. BMC bioinformatics 2019;20(1):521.

Addresses for correspondence:

Byunghan Lee (bhlee@seoultech.ac.kr)
Rm. 203, Changhak Hall, 232 Gongneung-ro, Nowon-gu, Seoul 01811, South Korea

Sungroh Yoon (sryoon@snu.ac.kr)
Rm. 908, Bldg. 301, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, South Korea