Automatic 12-lead ECG Classification Using a Convolutional Network Ensemble

Antônio H Ribeiro[1], Daniel Gedon[2], Daniel Martins Teixeira[1], Manoel Horta Ribeiro[3], Antonio L Pinho Ribeiro[1], Thomas B Schön[2], Wagner Meira Jr[1]

[1]Universidade Federal de Minas Gerais, Brazil,
[2]Uppsala University, Sweden,
[3]École Polytechnique Fédérale de Lausanne, Switzerland

Abstract

The 12-lead electrocardiogram (ECG) is a major diagnostic test for cardiovascular diseases and enhanced automated analysis tools might lead to more reliable diagnosis and improved clinical practice. Deep neural networks are models composed of stacked transformations that learn tasks by examples. Inspired by the success of these models in computer vision, we propose an end-to-end approach for the task at hand. We trained deep convolutional neural network models in the heterogeneous dataset provided in the Physionet 2020 Challenge and used an ensemble of seven of these convolutional models for the classification of abnormalities present in the ECG records. Ensembles use the output of multiple models to generate a combined prediction and are known to improve performance and generalization when compared to the individual models. In our submission, we use an ensemble of neural networks with the architecture similar to the one described in Nat Commun 11, 1760 (2020) for 12-lead ECGs classification. Our approach achieved a challenge validation score of 0.657, and full test score of 0.132, placing us, the "Code Team", in 28 out of 41 in the official ranking.

1. Introduction

Cardiovascular diseases are the leading cause of death worldwide [1] and the electrocardiogram (ECG) is a major tool in their diagnoses. Deep neural networks (DNNs) have recently achieved striking success in tasks such as image classification [2] and speech recognition [3]. Recent developments have demonstrated the ability of this technology to produce accurate ECG classifiers both in the single-lead [4] and in the 12-lead setup [5].

The 2020 Physionet Challenge [6] for the classification of 12-lead ECG involves several of the difficulties that are present in deploying a new ECG classifier that has not been fully considered by existing DNN-based solutions. The challenge presents a classification task that includes significantly more classes compared to previous challenges [7] and is based upon heterogeneous data for training and testing, which comes from four different countries and is collected and annotated under different conditions.

We use the challenge as an opportunity to benchmark and to improve the use of convolutional network architectures for 12-lead ECG classification. We improve on our previous work [5] in two ways. Firstly, we describe a simple approach to make it possible to work with signals with variable length. Secondly, we show the strengths of using an ensemble of those models.

2. Challenge description

The 2020 Physionet challenge [6] requires the participants to produce models capable of classifying 12-lead ECGs according to 27 classes. These classes cover different possible rhythms, morphologies and diagnosis. Six databases, from four different countries (United States, Russia, Germany and China), were made available for training the model, summing up to a total of more than 43 thousand ECG records.

The models submitted to the challenge are scored using a separate test set that is not available to the participants. The set contain unseen records both from the same sources as the training data and from an additional source that was not available to participants.

The challenge score metric is a weighted accuracy metric. Let $\mathbf{C}$ denote a multi-class multi-label confusion matrix, for which the diagnosis of each ECG record is accounted for. In case of multiple predictions or multiple simultaneous labels for one record, each prediction is assigned partial credit: it is divided by (the maximum of) the number of prediction or the number of classes. The score metric is given by the weighted sum: $\sum_i \sum_j w_{i,j} C_{i,j}$, where the weights $w_{i,j}$ are set according to the relative clinical relevance of an (in-)correct prediction.

 ISSN: 2325-887X DOI: 10.22489/CinC.2020.130

3. Model

3.1. Architecture and training procedure

We used a convolutional neural network architecture similar to the one proposed in [5] for 12-lead ECG classification. The architecture is an adaptation to unidimensional signals of the image-classification residual network proposed in [8]. This architecture allows deep neural networks to be efficiently trained by including skip connections. We have adopted the modification in the residual block proposed in [9], which place the skip connection in the position displayed in Figure 1.

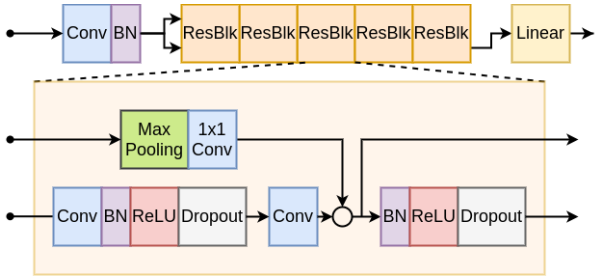

Figure 1. **Residual neural network.** The uni-dimensional neural network architecture. Adapted from [5].

We re-sample all ECG records at 400 Hz and, for this sample rate, the training dataset contains ECGs with lengths ranging from 2,000 to 720,000 samples. The model processes batches with a fixed length of 4,096 samples. An exam that exceeds this amount of samples is split over multiple batches. All splits share the same label. If necessary, the last batch of an exam is filled with zeros to complete 4,096 samples.

The network consists of a convolutional layer (`Conv`) followed by 5 residual blocks with two convolutional layers per block. The output of each convolutional layer is rescaled using batch normalization (`BN`) [10], and fed into a rectified linear activation unit (`ReLU`). `Dropout` [11] is applied after the nonlinearity with rate 0.5.

The convolutional layers have filter length 17, starting with 4,096 samples and 64 filters for the first layer in the first residual block and increasing the number of filters by 64 and subsampling by a factor of 4 every residual block (except for the first one). `Max Pooling` [12] and convolutional layers with filter length 1 (`1x1 Conv`) are included in the skip connections to make the dimensions match those from the signals in the main branch.

The DNN weights are adjusted during training by minimizing the average cross-entropy using the Adam optimizer [13] with default parameters and learning rate $\text{lr} = 10^{-3}$. The training runs for 200 epochs, with the learning rate being reduced by a factor of 10 at epoch 75, 125, and 175. The neural network weights are initialized as in [14] and the biases are initialized with zeros.

3.2. Ensemble model

The neural network architecture described above is trained multiple times starting from different randomly initialized weights. The trained models are combined to generate an ensemble of models for the task by averaging the output, i.e. the logits of all trained models [15]. The motivation behind using ensembles for neural networks is well established: The non-convex loss of deep neural networks is known to have multiple local minima with similar loss values. Starting the optimization process from different random initializations will let the neural network converge to different local optima. However, the learned functions vary and will therefore yield diverse predictions which do not necessarily overlap in which input they misclassify [16]. This yields ensembles of DNNs successful in obtaining higher performance and better generalization capabilities over the individual models. In this work, we use a total of seven models.

3.3. Test-time procedure

The test-time procedure is depicted in Figure 2. As in training time, all records are re-sample to 400 Hz and records that exceed 4,096 samples are split into multiple batches (`split in batches`). For an exam that has been split into multiple batches, the residual neural network, described in Section 3.1 (`ResNet`), is used to compute an output vector (logits) and the average of the values of each split is computed (`avg logits`). This is done for each one of the $n = 7$ ensemble models and the results are averaged over the ensemble values (`avg ens`). This averaged logit is then fed into a sigmoid function (σ) which computes a value between 0 and 1 for each of the classes. The result can be understood as the probability of the occurrence of the given class and is multiplied by a correction factor accounting for class imbalance. If the obtained value is above the threshold of 0.5, the corresponding class is considered to be present. The correction factor we used is one over the relative number of occurrences of a class in the training set (and can be understood as a prior).

4. Results

During the official phase, our team has submitted 5 models to be scored using the challenge hidden test set. Table 1 summarizes the entries. The first entry submitted to the challenge consists of a single convolutional neural network model, described in Section 3.1.

This first submission uses an output layer based on a sigmoid activation function to classify the records into 27

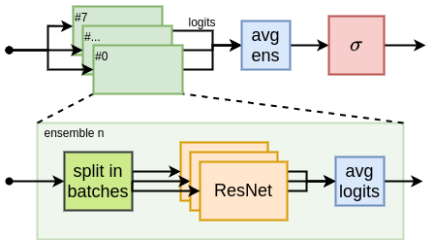

Figure 2. **Test-time prediction.** Ensemble of models which can handle variable length ECGs by splitting them into batches.

entry	score	short description
#1	0.622	DNN model described in Section 3.1
#2	0.626	Jointly predict some of the classes
#3	0.637	Larger weight for the top-k predictions
#4, #5	0.657	Ensemble of 7 models

Table 1. **Challenge submissions.** Challenge metric on the partial test dataset from the official challenge phase for the 5 models submitted by the "Code Team".

classes. The challenge score metric, however, does not penalize the model when it mixes up between three of the pairs of the 27 classes. Hence, in our second submission, we collapsed these pairs of classes (resulting in a sigmoid output layer with output of size 24) to account for that.

The sigmoid output layer predicts the occurrence or non-occurrence of a given class individually. However, we would not like to have too many simultaneous predictions, since these might get penalized by the challenge metric. Hence, heuristically, we limit the number of simultaneous diagnoses by the model to $k = 6$. A modification in our third submission weights the sigmoid output layer from the largest to the smallest value with a decreasing weight vector $\mathbf{w}$ with length k. Entries in $\mathbf{w}$ with $i > k$ are zero to avoid more than k predictions for the same exam.

Finally, in the fourth and fifth submissions (which are identical) we used an ensemble of seven models. Each one of the models in the ensemble is identical (except the random initialization seed) to our third submission.

Our best result in the validation set (submission #5) achieved the score of 0.132 in the full test set placing us in the 28-th position out of 41 teams in the official ranking. The test set comes from three different sources, with our implementation achieving 0.830 on the 1st set, 0.181 on the 2nd and 0.023 on the 3rd one. Our implementation is available at: `github.com/antonior92/physionet-12ecg-classification`.

5. Model design and analysis

For the official submissions described above, we used the entire dataset (that was made available to us) for training. However, in order to set the hyper-parameters and compare between different DNN architectures, we divided the training data into 70%-30%, using the 30% split to evaluate the model. We conducted a random search with different combinations of kernel size in $\{9, 15, 17 \text{ and } 35\}$, learning rate in [0.001, 0.01] range and a dropout rate in (0, 1). We tried around 20 different configurations and the one used in the submission was among the best candidate configurations.

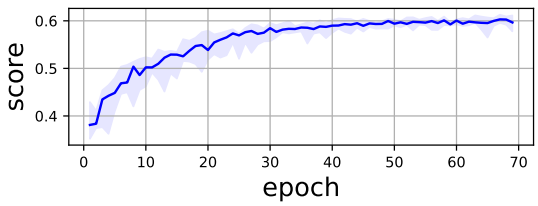

Figure 3. **Performance during training.** Challenge score evaluated on the hold-out validation data as a function of the number of epochs (for the initial 70 epochs). The full line indicates the median for 7 models and the shaded area illustrates the maximum and minimum.

The 70%-30% setup is used to generate the results in Figures 3, 4 and 5. We trained 7 ensemble models in the scenario described above (initialized with different seeds), the training history of these models is displayed in Figure 3. The performance of the ensembles of the models is evaluated in Figure 4, which gives the challenge metric as a function of the ensemble size for all combinations of ensembles with size varying from 1 to 7 obtained from combinations of these 7 models. The figure underlines our design choice of using ensembles to boost the final performance. The confusion matrix for the final ensemble of 7 models is displayed in Figure 5.

Acknowledgements

This research was partially supported by Brazilian Agencies CNPq, CAPES, and FAPEMIG, by the projects MASWeb, EUBra-BIGSEA, INCT-Cyber, ATMOSPHERE, and by the *Wallenberg AI, Autonomous Systems and Software Program (WASP)* funded by Knut and Alice Wallenberg Foundation.

References

[1] Roth GA, Abate D, Abate KH et al. Global, regional, and national age-sex-specific mortality for 282 causes of death in 195 countries and territories, 1980–2017: A systematic analysis for the Global Burden of Disease Study 2017. The Lancet November 2018;392(10159):1736–1788.

[2] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Ad-

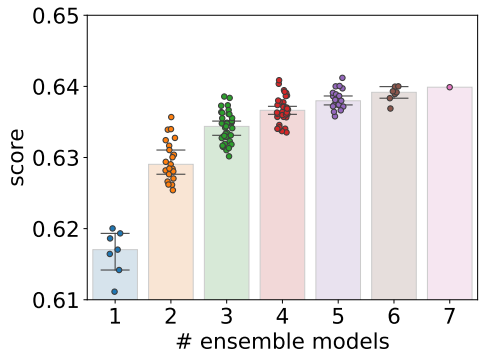

Figure 4. **Ensemble performance.** The plot displays the challenge score metric evaluated on the 30% hold-out validation data. The performance is given as a function of the number of models in the ensemble. For each number of ensemble models n we considered all different ensemble combination $\binom{n}{7}$ to generate the data points and statistics. The median is indicated by the bars and the bootstrapped confidence intervals are represented by the error bars.

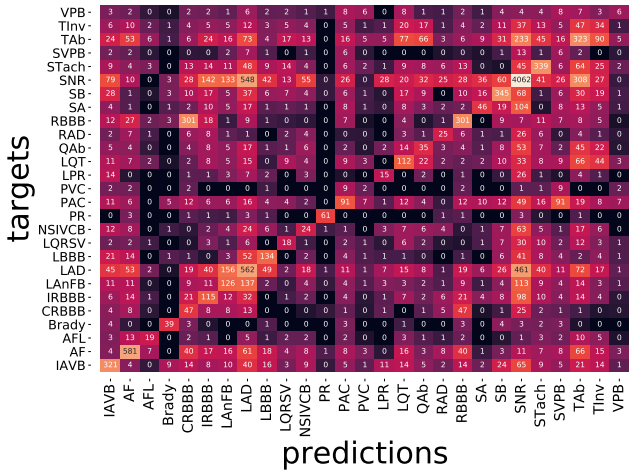

Figure 5. **Confusion matrix.** The confusion matrix $\mathbf{C}$ (see Section 2) for all classes scored by the challenge on the held out validation data. Colors are in log scale with low values of co-occurrence in dark and high values in bright. For a full description of the classes see [6].

vances in Neural Information Processing Systems. 2012; 1097–1105.

[3] Hinton G, Deng L, Yu D, Dahl GE, Mohamed A, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath TN, Kingsbury B. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Processing Magazine, 2012;29(6):82–97.

[4] Hannun AY, Rajpurkar P, Haghpanahi M, Tison GH, Bourn C, Turakhia MP, Ng AY. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. Nature Medicine January 2019;25(1):65–69.

[5] Ribeiro AH, Ribeiro MH, Paixão GMM, Oliveira DM, Gomes PR, Canazart JA, Ferreira MPS, Andersson CR, Macfarlane PW, Meira Jr. W, Schön TB, Ribeiro ALP. Automatic diagnosis of the 12-lead ECG using a deep neural network. Nature Communications 2020;11(1):1760.

[6] Alday EAP, Gu A, Shah A, Robichaux C, Wong AKI, Liu C, Liu F, Rad AB, 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 (In press).

[7] Liu F, Liu C, Zhao L, Zhang X, Wu X, Xu X, Liu Y, Ma C, Wei S, He Z, Li J, Yin Kwee EN. An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection. Journal of Medical Imaging and Health Informatics September 2018; 8(7):1368–1373.

[8] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016; 770–778.

[9] He K, Zhang X, Ren S, Sun J. Identity mappings in deep residual networks. Computer Vision ECCV 2016. Springer International Publishing. 630–645.

[10] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning. PMLR, June 2015; 448–456.

[11] Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 2014;15(1):1929–1958.

[12] Hutchison D, Kanade T, Kittler J, Kleinberg JM et al. Evaluation of pooling operations in convolutional architectures for object recognition. In: Artificial Neural Networks – ICANN 2010, volume 6354. Springer Berlin Heidelberg, 2010; 92–101.

[13] Kingma DP, Ba J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference for Learning Representations (ICLR), 2014.

[14] He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: Proceedings of the IEEE International Conference on Computer Vision, 2015;1026–1034.

[15] Lakshminarayanan B, Pritzel A, Blundell C. Simple and scalable predictive uncertainty estimation using deep ensembles. In Advances in Neural Information Processing Systems 30, 2017; 6402–6413.

[16] Fort S, Hu H, Lakshminarayanan B. Deep ensembles: a loss landscape perspective. arXiv:1912.02757, 2019.

Address for correspondence:

Antônio H. Ribeiro
Computer Science Department, Universidade Federal de Minas Gerais, Av. Pres. Antônio Carlos, 6627, Belo Horizonte (MG), Brazil, 31270-901
antoniohorta@dcc.ufmg.br