Convolutional Recurrent Neural Networks for Electrocardiogram Classification

Martin Zihlmann, Dmytro Perekrestenko, Michael Tschannen

Dept. IT & EE, ETH Zurich, Switzerland

Abstract

We propose two deep neural network architectures for classification of arbitrary-length electrocardiogram (ECG) recordings and evaluate them on the atrial fibrillation (AF) classification data set provided by the PhysioNet/CinC Challenge 2017. The first architecture is a deep convolutional neural network (CNN) with averaging-based feature aggregation across time. The second architecture combines convolutional layers for feature extraction with long-short term memory (LSTM) layers for temporal aggregation of features. As a key ingredient of our training procedure we introduce a simple data augmentation scheme for ECG data and demonstrate its effectiveness in the AF classification task at hand. The second architecture was found to outperform the first one, obtaining an F_1 score of 82.1% on the hidden challenge testing set.

1. Introduction

We consider the task of atrial fibrillation (AF) classification from single lead electrocardiogram (ECG) recordings, as proposed by the PhysioNet/CinC Challenge 2017 [1]. AF occurs in 1-2% of the population, with incidence increasing with age, and is associated with significant mortality and morbidity [2]. Unfortunately, existing AF classification methods fail to unlock the potential of automated AF classification as they suffer from poor generalization capabilities incurred by training and/or evaluation on small and/or carefully selected data sets.

In this paper, we propose two deep neural network architectures for classification of arbitrary-length ECG recordings and evaluate them on the AF classification data set provided by the PhysioNet/CinC Challenge 2017. The first architecture is a 24-layer convolutional neural network (CNN) with averaging-based feature aggregation across time. The second architecture is a convolutional recurrent neural network (CRNN) that combines a 24-layer CNN with a 3-layer long-short term memory (LSTM) network for temporal aggregation of features. CNNs have the ability to extract features invariant to local spectral and spatial/temporal variations, and have led to many breakthrough results, most prominently in computer vision [3, Chap. 9]. LSTM networks, on the other hand, were shown

to effectively capture long term temporal dependencies in time series [3, Chap. 10]. As a key ingredient of our training procedure we introduce a simple yet effective data augmentation scheme for the ECG data at hand.

Related work: Our network architectures are loosely inspired by [4–6]. More specifically, a CRNN for polyphonic sound detection was proposed in [6]. Here, unlike in AF classification where one has to infer a single label per ECG, the input audio sequence is mapped to sequences labels, inferring the sound events as a function of time. Work [5] employs a CRNN for mental state classification from electroencephalogram (EEG) data. In [4], LSTM networks are used for multilabel classification of diagnoses in electronic health recordings. Shortly before finalizing this work, we became aware of the preprint [7], which proposes a deep CNN architecture for arrhythmia detection in ECGs, but unlike in the classification problem considered here, maps the ECG signal to a sequence of rhythm classes. Finally, we refer to [8] for an overview over existing methods for AF classification that are not based on deep neural networks.

2. Methods[1]

In this section we give a detailed description of our network architectures as well as the training and evaluation procedures used.

2.1. Network architectures

We propose two neural network architectures for ECG classification, a CNN and a CRNN, illustrated in Fig. 2. Both architectures consist of four parts: 1) data preprocessing computing a logarithmic spectrogram of the input; 2) a stack of convolutional layers for feature extraction; 3) aggregation of features across time by averaging and an LSTM block in case of the CNN and the CRNN, respectively; 4) a linear classifier. In the following we describe each of the aforementioned parts in detail.

1) Logarithmic spectrogram: To preprocess the data we compute the one-sided spectrogram of the time-domain input ECG signal and apply a logarithmic transform. Pre-

---

[1] Source code is available at:
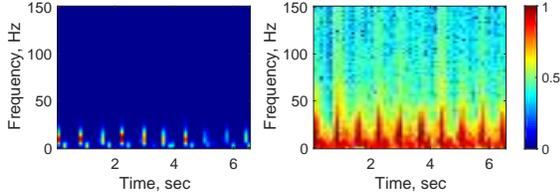https://github.com/yruffiner/ecg-classification

Figure 1: Normalized spectrogram (left) and normalized logarithmic spectrogram (right) of an example ECG signal.

liminary experiments showed that the logarithmic transform considerably increases the classification accuracy; Fig. 1 illustrates the effect of the logarithmic transform. The spectrogram is computed using a Tukey window of length 64 (corresponding to 213ms at the 300Hz sampling rate of the challenge data and resulting in 33 effective frequency bins) with shape parameter 0.25 and 50% overlap.

2) Convolutional layers: All convolutional layers first apply a set of 5×5 convolutional filters, followed by Batch-normalization and ReLU activation. The convolutional layers are grouped in blocks of 4 and 6 layers for the CNN and CRNN architecture, respectively, referred to as ConvBlock4 and ConvBlock6. The number of channels (feature maps) as well as the size of the feature maps remains constant in all but the last layer of each ConvBlock. The last layer applies max-pooling over 2×2 windows and increases the number of channels. Specifically, the number of channels at the output of the first ConvBlock is 64 and is increased by 32 by each subsequent ConvBlock, resulting in 224 1-dimensional and 160 3-dimensional feature maps (per output time step) for the CNN and CRNN, respectively, at the output of the last ConvBlock fed to the feature aggregation part 3) (see Fig. 2).

3) Feature aggregation across time: As the ConvBlocks process the variable-length input ECG signals in full length, they produce variable length outputs, which have to be aggregated across time before they can be fed to a standard classifier (which typically requires the dimension of the input to be fixed). In our CNN architecture, temporal aggregation is achieved simply by averaging, whereas in the CRNN architecture the 3-dimensional feature maps are first flattened and then feed to a 3-layer bidirectional LSTM network with 200 neurons in each layer. The (temporally) last output of the LSTM network then serves as the aggregated feature vector.

Averaging realizes temporal smoothing of features and may therefore not be suited to classify episodic phenomena occurring only during a short time span relative to the signal length, as in certain types AF. The LSTM network, on the other hand, aggregates the features in a highly non-linear manner across time and potentially preserves episodic phenomena better.

4) Linear classifier: We employ a standard linear layer with SoftMax to compute the class probabilities.
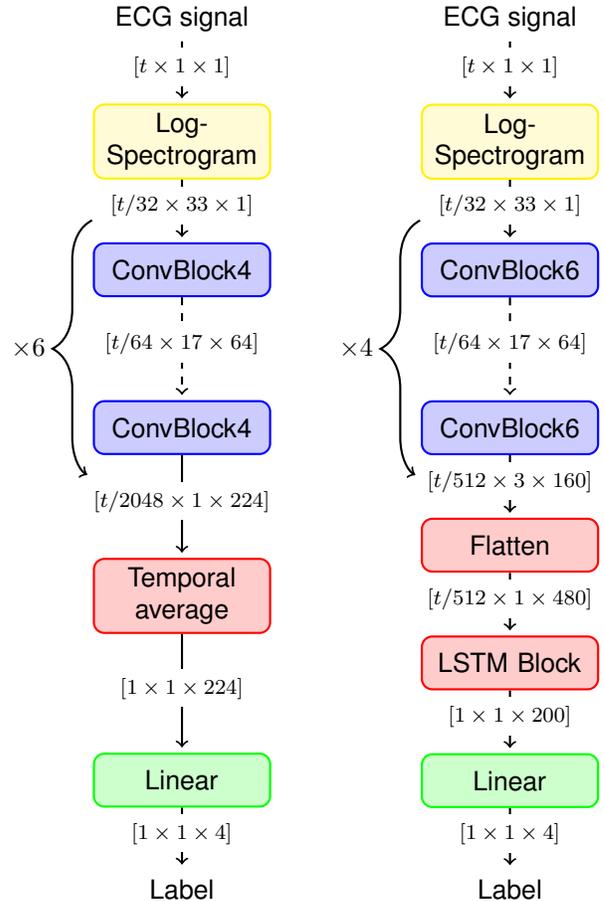

Figure 2: The proposed CNN (left) and CRNN (right) architecture. The tensor dimensions are given in the format [time t × nbr. of features × nbr. of channels].

2.2. Training

For both network architectures we used the cross-entropy loss (reweighted as to account for the class frequencies) as training objective, and employed the Adam optimizer with the default parameters recommended in [9]. The batch size was set to 20. Furthermore, we used dropout with probability 0.15 in all layers and early stopping based on the F_1 measure described in Sec. 2.3.

Training protocols: We trained the CNN end-to-end from scratch without encountering any issues. Training the convolutional and recurrent layers in the CRNN jointly from scratch, on the other hand, did not lead to convergence. We therefore adopted the following 3-phase protocol to train the CRNN. In phase 1, the LSTM block was replaced by feature averaging across time and the convolutional layers were trained together with a linear classifier for 500 epochs. In phase 2, the feature averaging operator was swapped with the LSTM block and the recurrent

layers were trained for 100 epochs, while keeping the convolution layers fixed. In phase 3, the convolutional and recurrent layers were trained jointly, reducing the learning rate by a factor of 10 every 200 epochs.

Data augmentation: We observed severe overfitting in preliminary experiments. This can be attributed to the fact that number of parameters in the proposed architectures is large compared to the size of data set used for evaluation (see Sec. 2.3). It was demonstrated in [10] that data augmentation can act as a regularizer to prevent overfitting in neural networks, and also improves classification performance in problems with imbalanced class frequencies [11]. We therefore developed a simple data augmentation scheme tailored to the ECG data at hand. Specifically, we employ two data augmentation techniques, namely *dropout bursts* and *random resampling*.

Dropout bursts are created by selecting time instants uniformly at random and setting the ECG signal values in a 50ms vicinity of those time instants to 0. Dropout burst hence model short periods of weak signal due to, e.g., bad contact of ECG leads.

Assuming a heart rate of 80bpm for all training ECG signals, random resampling emulates a broader range of heart rates by uniformly resampling the ECG signals such that the heart rate of the resampled signal is uniformly distributed on the interval $[60, 120]$bpm. These emulated heart rates may be unrealistically high or low due to the assumption of an 80bpm heart rate independently of the signal.

Ensembling: To exploit the entire data set at hand (recall that we employ early stopping which uses part of the data set for validation) we used ensembles of 5 networks of the same type (i.e., either CNN or CRNN) to build production models, combining the individual predictions by majority voting. Specifically, we partitioned—in a stratified manner—the data set into 5 equally sized subsets, and, for every network in the ensemble, used 4 of the subset for training and the remaining subset for validation, choosing a different subset for validation for every network.

2.3. Evaluation

We evaluated the proposed CNN and CRNN architecture on the publicly available PhysioNet/CinC Challenge 2017 data set containing 8,528 single lead ECG recordings of length ranging from 9 to 61sec, sampled at 300Hz. Each recording is labeled with one of the classes "normal rhythm", "AF rhythm", "other rhythm", and "noisy recording" (we will henceforth use the abbreviations "N", "A", "O", and "~", respectively). The classification performance was measured using the average over the class F_1 scores of the classes N, A, and O, i.e., $F_{1,\text{avg}} = \frac{1}{3} \sum_{c \in \{\text{N, A, O}\}} F_{1,c}$, where $F_{1,c} = 2\#\text{TP}_c / (2\#\text{TP}_c + \#\text{FN}_c + \#\text{FP}_c)$ (using TP_c, FP_c, and FN_c to denote the true positives, false positives,

Arch.	metric	N	A	O	~	overall
CNN	acc.	88.1	83.6	66.9	77.1	81.2
	F_1	87.8	79.0	70.1	65.3	79.0
CRNN	acc.	89.9	77.8	69.4	71.5	82.3
	F_1	88.8	76.4	72.6	64.5	79.2

Table 1: Accuracies (acc.) and F_1 scores (in %) for the proposed network architectures (estimated using 5-fold cross validation).

Arch.	metric	N	A	O	~	overall
CNN	acc.	90.5	64.2	68.0	54.9	80.5
	F_1	88.3	69.9	69.1	59.6	75.8
CRNN	acc.	90.2	69.1	63.0	51.1	79.2
	F_1	87.4	69.9	66.5	54.9	74.6

Table 2: Accuracies (acc.) and F_1 scores (in %) for the proposed network architectures with *data augmentation deactivated* (estimated using 5-fold cross validation).

and false negatives, respectively, for class c). We refer the reader to [1] for a detailed description of the data set. We evaluated the proposed network architectures via stratified 5-fold cross-validation. To realize early stopping, for every fold, we split the training data into two partitions, one for training and one for validation containing $5/6$ and $1/6$, respectively, of the training data. Thus, for every fold, the effective training set size amounted to $4/5 \cdot 5/6 = 2/3$ or 66.6% of all data available. We hence expect that an ensemble of 5 networks yields a higher $F_{1,\text{avg}}$ as it exploits all data available.

To demonstrate the effectiveness of the proposed data augmentation scheme, we trained the CNN and CRNN exactly as described in Sec. 2.2, but without data augmentation.

3. Results

Tables 1 and 2 show the class F_1 scores and $F_{1,\text{avg}}$ (overall) along with the corresponding classification accuracies for the proposed architectures with and without data augmentation, respectively. The CRNN yielded a higher overall accuracy and slightly higher $F_{1,\text{avg}}$ than the CNN when data augmentation was employed. The opposite can be observed in the case when data augmentation was deactivated. In both cases, none of the architectures has consistently higher class accuracies or class F_1 scores than the other. Data augmentation is seen to considerably increase $F_{1,\text{avg}}$ for both CNN and CRNN, with a slightly better improvement for the CRNN.

Based on these results we chose to submit an ensemble of CRNNs to the PhysioNet/CinC Challenge 2017. This ensemble obtained an $F_{1,\text{avg}}$ of 0.82 on the private chal-

lenge testing set, which corresponds to the second best score (after rounding to two decimal places as per [1]) obtained in the challenge. In terms of running time, the ensemble on average consumed 58.1% of the computation quota available on the challenge evaluation server.

4.   Discussion

The results presented in Sec. 3 indicate that aggregation of features across time using an LSTM network is more effective than averaging in the ECG classification task under consideration, when data augmentation is employed. However, this has to be taken with a grain of salt as the CRNN has more parameters, and thereby potentially a higher model capacity, than the CNN. In addition, we observed that phase 3 of the CRNN training protocol did not consistently lead to an increase in $F_{1,\text{avg}}$, and further improvements might be achieved by refining the training protocol. Furthermore, the results in Sec. 3 also show the effectiveness of the proposed data augmentation scheme, indicating that it captures certain real world phenomena— at least to some extent.

We briefly comment on directions we explored in preliminary experiments, but which did not lead to improvements and were therefore not included in our final training protocols. As an alternative to data augmentation we tried to pretrain the CNN and the convolutional layers of the CRNN on the PTB Diagnostic ECG Database [12], which contains 549 14-lead ECG recordings of 290 subjects with a variety of different cardiac conditions. This pretraining procedure did not lead to improvements compared to initialization with random weights. We further explored [3, Alg. 7.3] to incorporate the knowledge in the validation set into a single production model, which is more effective than ensembling from a computational and storage point of view. In a nutshell, [3, Alg. 7.3] continues training on the union of the training and the validation set after activation of early stopping until the average loss on the validation set attains the average loss on the training set obtained at the time of activation of early stopping. However, continuing training according [3, Alg. 7.3] led to a decrease in $F_{1,\text{avg}}$ in our challenge submissions.

5.   Conclusion

We developed and evaluated two deep neural network architectures for ECG classification. In addition, we proposed a simple data augmentation scheme for ECG data and demonstrated its effectiveness. Applying our architectures to multi lead ECG data, possibly with different pathology, as well as refining and extending the data augmentation scheme, e.g., by taking the actual heart rate into account for random resampling (instead of assuming 80bpm), are interesting directions to be explored in the future.

References

[1] Clifford G, Liu C, Moody B, Lehman L, Silva I, Li Q, Johnson A, Mark R. AF classification from a short single lead ECG recording: The Physionet/Computing in Cardiology Challenge 2017. In Computing in Cardiology. 2017; .

[2] Lip GYH, Fauchier L, Freedman SB, Van Gelder I, Natale A, Gianni C, Nattel S, Potpara T, Rienstra M, Tse HF, Lane DA. Atrial fibrillation. Nature Reviews Disease Primers 2016;2:16016.

[3] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press, 2016.

[4] Lipton ZC, Kale DC, Elkan C, Wetzell R. Learning to diagnose with LSTM recurrent neural networks. In Proc. Int. Conf. on Learn. Representations (ICLR). 2016; .

[5] Bashivan P, Rish I, Yeasin M, Codella N. Learning representations from EEG with deep recurrent-convolutional neural networks. In Proc. Int. Conf. on Learn. Representations (ICLR). 2016; .

[6] Cakır E, Parascandolo G, Heittola T, Huttunen H, Virtanen T. Convolutional recurrent neural networks for polyphonic sound event detection. IEEE ACM Trans on Audio Speech and Language Processing 2017;25(6):1291–1303.

[7] Rajpurkar P, Hannun AY, Haghpanahi M, Bourn C, Ng AY. Cardiologist-level arrhythmia detection with convolutional neural networks. arXiv preprint arXiv170701836 2017;.

[8] Larburu N, Lopetegi T, Romero I. Comparative study of algorithms for atrial fibrillation detection. In Computing in Cardiology. 2011; 265–268.

[9] Kingma DP, Ba J. Adam: A method for stochastic optimization. In Proc. Int. Conf. on Learn. Representations (ICLR). 2015; .

[10] Simard P, Steinkraus D, Platt J. Best practices for convolutional neural networks applied to visual document analysis. In Proc. Int. Conf. on Document Analysis and Recognition. 2013; .

[11] Chawla N, Bowyer K, Hall L, Kegelmeyer W. Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 2002;.

[12] Bousseljot R, Kreiseler D, Schnabel A. Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet. Biomedizinische Technik Biomedical Engineering 1995; 40(s1):317–318.

Address for correspondence:

Dmytro Perekrestenko, Michael Tschannen
ETH Zürich, Communication Technology Laboratory
Sternwartstrasse 7
CH-8092 Zürich
Switzerland
{pdmytro, michaelt}@nari.ee.ethz.ch