Convolutional Neural Network and Rule-Based Algorithms for classifying 12-lead ECGs

Bjørn-Jostein Singstad[1], Christian Tronstad[1, 2]

[1]University of Oslo, Oslo, Norway
[2] Oslo University Hospital, Oslo, Norway

Abstract

This study is a part of the PhysioNet/Computation in Cardiology (CinC) Challenge 2020. Our objective was to classify 27 cardiac abnormalities based on a provided dataset of 43101 ECG recordings. We developed a hybrid model combining a rule-based algorithm with different Deep Learning architectures.

We compared two different Convolutional Neural Networks (FCN and Encoder), a combination of both, and with the addition of another neural network. Two of these combinations were finally combined with a rule-based model using derived ECG features. We evaluated the performance of the models on validation data during model development using hold- out validation. Finally, we deployed the models to a Docker image, trained the model on the provided development data before the models were tested on a hidden data set, giving a performance based on a particular Physionet Challenge score.

Our team, TeamUIO, found that the FCN in parallel with an Encoder without any rule-based model performed best on the validation data with a score of 0.412. However, the best score on the test set was the Encoder in parallel with a Fully Convolutional Network with the rule-based model added, receiving a score of 0.377.

1. Introduction

The electrocardiogram (ECG) reflects the electrical activity of the heart, and the interpretation of this recording can reveal numerous pathologies of the heart. An ECG is recorded using an electrocardiograph, where modern clinical devices usually contain automatic interpretation software that interprets the ECGs directly after recording. Although automatic ECG interpretation started in the 1950s, there are still some limitations [1, 2]. Because of the errors they make, doctors have to read over the ECGs [3]. This is time consuming for the doctors and requires high degree of expertise [4]. There is clearly a need for better ECG interpretation algorithms.

The recent years has shown a rapid improvement in the field of machine learning. A sub-field of machine learning is called Deep Learning, where more complex architectures of neural networks are better able to scale with the amount of data in terms of performance. This type of machine learning has shown promising performance in many fields including medicine, and in this study, we have explored the usefulness of deep learning in classifying 12-lead ECGs.

As a starting point for our model architecture we chose to use the two best performing Convolutional Neural Networks (CNN) used on ECG data in Fawaz HI et al 2019 [5]. They reported that Fully Convolutional Networks (FCN) outperformed eight other CNN architectures compared. We also wanted to test the second-best architecture which was an Encoder Network. We also assessed the integration of a rule-based algorithm within these models in order to test the performance of a CNN and rule-based hybrid classifier.

This study is a part of the PhysioNet/Computing in Cardiology Challenge 2020, where the aim was to develop an automated interpretation algorithm for identification of clinical diagnoses from 12-lead ECG recordings.

2. Methods

2.1. Data

To train the CNN models we used open data from six different sources [6–9]. The data set contained 43.101 ECG recordings in total, where each ECG recording also included one associated information file. The information file described the recording, patient attributes (age and gender) and the diagnosis (the label we want to predict). All diagnoses where decoded according to Systematised Nomenclature of Medicine Clinical Terms (SNOMED-CT) encoding. A total number of 111 diagnoses where represented in the data set. As some of the different diagnoses could co-exist with each other, there was a total number of 1414 different combinations of diagnoses represented in this data set.

The recording length varied across the different ECG signals, but 83.4% were 5000 samples long. 98.5% of the recordings were sampled at a frequency of 500Hz except for 1.3% signals sampled at 1kHz and 0.2% signals sampled at 257Hz.

2.2. Preprocessing

According to the goal of this challenge we aimed to classify 27 of the 111 diagnoses [10]. The 27 labels to classify were One-Hot encoded, with each diagnosis represented as a bit in a 27-bit long array. All recordings were padded and truncated to a signal length of 5000 samples. Padding and truncation were done by removing any parts longer than 5000 samples and adding a tail of $5000 - n$ zeros to any recording of length $n < 5000$.

2.3. CNN architectures

As a starting point for classifying the ECG-signals we employed FCN and Encoder types of CNN models as described in Fawaz HI et al 2019 [5]. We tested the two models without any modifications to the architecture other than changing the input and output layers to fit our input data and output classes. We ensured that all output layers of each models used the Sigmoid activation function.

To make use of the provided age and gender data we added a simpler neural network model with 2 inputs, one hidden layer of 50 units and 2 outputs. We combined this new model with our FCN and Encoder models by concatenation of the last layer of the CNNs.

Age data were passed into the model as integers, but in some information files the age of the patient was not given, and we assigned them a value of -1. The gender data was transformed into integers, where male was set equal to 0, female equal to 1 and Unknown was set to 2.

The two CNN models (FCN and Encoder) were combined as two parallel models, concatenated on the second last layer. This model was also tested with and without a parallel dense layer.

2.4. Rule-based model

The rule-based algorithm took the raw ECG signal, without any padding or truncating, as input. R-peak detection [11], and heart rate variability (HRV) analysis was programmed in order to add relevant derived features to the model. An HRV- score was obtained by computing the root mean square of successive differences between normal heartbeats (RMSSD) using the detected R-peaks as timing indicators of each heartbeat.

The rule-based algorithm was able to classify eight different diagnoses: atrial fibrillation, bradycardia, low QRS-complex, normal sinus rhythm, pacing rhythm, sinus arrhythmia, sinus bradycardia and sinus tachycardia.

The rule-based algorithm performed classification independent of the deep learning models. If there was disagreement between the rule-based algorithm and the CNN model, the rule-based algorithm overwrote the classification from the CNN model.

2.5. Model development

We trained and validated the model on the development dataset using hold-out validation with a split of 38790 (90%) for training and 4311 for validation (10%). We used the first fold in a stratified K-fold with a random seed of 42 [12]. The splitting was arranged such that all the 1414 unique combinations of diagnoses were present in both the training and validation data.

During training we used the Area Under the Curve (AUC) score on the validation set to determine if the learning rate should drop or stay. The learning rate was initially set to 0.001 for all models and decreased by a factor of 10, using the reduce on plateau method [13], each epoch the AUC score did not improve. Early stopping [13] was triggered when the AUC score on the validation data did not improve over two successive epochs.

2.6. Threshold optimization

After training the model on the validation set, we optimized the prediction thresholds. This was done by running the classifier on all the validation data and receiving a score between 0 and 1 for each of the classes. We then used Nelder-Mead Downhill Simplex Method [14, 15] to optimize the threshold individually for the 27 classes. Downhill Simplex Method is used to find the local minimum of a function using the function itself and a initial guess of the variable of the function. We optimized the 27-element long array using the negative of the PhysioNet scoring function [10]. To increase the possibility of finding the global maximum of the PhysioNet score we gave all elements in the 27-element long array a value of 1 and multiplied it with a variable that was given values from 0 to 1, with a step size of 0.05. We used the value that gave the highest PhysioNet score as the initial guess for the Downhill Simplex Method.

2.7. Model deployment

To obtain a valid score in the PhysioNet/CinC Challenge we submitted a model to the PhysioNet/CinC committee for testing on a hidden test set. We used a Docker image to create a virtual Python environment for the model to be tested. When the model was trained for deployment it was trained on the whole development set. The first test scores were obtained using AUC on the development data

Model	Rule-based model	validation AUC	validation F1	validation F2	validation G2	validation score	test score
FCN	No	0.875	0.381	0.446	0.230	0.348	-
Encoder	No	0.866	0.396	0.429	0.228	0.398	0.229
FCN + Age, Gender	No	0.877	0.368	0.438	0.222	0.385	0.302
Encoder + Age, Gender	No	0.828	0.334	0.389	0.190	0.333	0.272
Encoder + FCN	No	0.872	0.399	0.436	0.237	0.409	-
Encoder + FCN	Yes	0.872	0.361	0.413	0.203	0.348	0.377
Encoder + FCN + Age, Gender	No	0.866	0.400	0.434	0.233	0.395	-
Encoder + FCN + Age, Gender	Yes	0.866	0.356	0.405	0.198	0.338	0.364

Table 1. Scores obtained by eight different models during model development and model deployment. The models were evaluated by 5 different metrics, AUC, F1, F2, G2 and PhysioNet scoring metric, during model development. In the deployment phase the model was only evaluated by the PhysioNet scoring metric. 3 scores are missing in the test score column due to unsuccessful deployment

to schedule the reduction of the learning rate. The second half of the test scores were obtained using a learning rate scheduler. The learning rate schedule was programmed to be the same as in the model development.

2.8. General parameters for both validation and testing procedures

For all models in both validation and deployment we used binary cross entropy as loss function. A batch generator was used to feed the model with data during training, programmed to shuffle the order of data for each epoch.

To deal with the imbalanced data (skewed classes) in our data set we calculated weights based on the number of occurrences of the different classes [12]. The calculated weights were passed to the model during training to give higher priority to rare diagnoses and lower priority to diagnoses that occur more frequently.

Hyperparameter tuning was done on a subset of data to select the best optimizer [Adam, SDG, Adamax, RMSprop, Adadelta, Ftrl, Nadam] and batch size [20, 30, 40, 50]

3. Results

3.1. Scoring metrics

All models were scored on the validation data and compared using the metrics AUC (Eq 1), F1-score (Eq 2), F2-score (Eq 3), G2-score (Eq 4) and the PhysioNet Challenge score[10]. On the test set we only obtained the PhysioNet Challenge Score.

$$AUC_{(t_i - t_{i-1})} = (t_i - t_{i-1}) \times \frac{f(t_i) + f(t_{i-1})}{2} \quad \text{(Eq 1)}$$

$$F_1 = \frac{2 \times TP}{2 * TP + FP + FN} \quad \text{(Eq 2)}$$

$$F_2 = \frac{(1 + 2^2) \times TP}{(1 + 2^2) \times TP + FP + 2^2 \times FN} \quad \text{(Eq 3)}$$

$$G_2 = \frac{TP}{TP + FP + 2 \times FN} \quad \text{(Eq 4)}$$

3.2. Classification performance

Five out of the eight models we have validated in this study were successfully deployed and thus obtained a score on the test set. The best performance during the model development, on the validation set, was achieved by an Encoder in parallel with an FCN as seen in row five in table 1. The best result on the test set was achieved by Encoder in parallel with an FCN with rule-based algorithms as seen in row six in table 1.

4. Discussion and conclusion

In this study we chose to pad and truncate the signals to 5000 samples which was necessary to be able to feed the signal to the CNN. The disadvantage of doing this was that some important information from segments of the ECG recordings could have been omitted in training the models. On the other hand, the derived features used in the rule-based implementation were based on complete recordings.

Deployment of the models were done using two different ways of controlling the learning rate. Row 2, 3 and 4 in Table 1 shows test scores that were obtained by using AUC on the development data to schedule the reduction of the learning rate. This could possibly have contributed to the overfitting indicated by the validation to test score differences for these models. The test scores in row 6 and 8 in Table 1 were obtained using a learning rate scheduler are more consistent with their validation scores . In summary, our result indicates that the deployed models that keep the same training schedule as in the development model seems to avoid overfitting and performs better on unseen data.

The results in Table 1 show that FCN performed better than the Encoder on AUC, F2 and G2-score on the validation set during model development. The encoder on the other hand performed better on the F1 score and the PhysioNet Challenge metric. This can indicate that FCN is slightly better than the Encoder, but since the PhysioNet metric is the main metric in this study this can be used to argue for the opposite.

Our parallel model for gender and age decreased the performance on every metric for the Encoder. However, for the FCN the AUC and the PhysioNet score improved when adding the parallel model for gender and age.

Encoder + FCN and Encoder + FCN + Age, Gender decreased in performance on the validation data during model development when adding the rule-based model. However, the performance of the deployed model was actually better than the development model. Our results indicate that the hybridization of CNN with a rule-based model could improve diagnostic classification of ECG, but further analysis is needed to confirm whether, and to which extent such implementation improves the performance of the proposed CNN models.

5. Code Availability

The code used in model development is available in a Kaggle Notebook environment [1]

References

[1] Schläpfer J, Wellens HJ. Computer-Interpreted Electrocardiograms. Journal of the American College of Cardiology August 2017;70(9):1183–1192. ISSN 07351097.

[2] Smulyan H. The Computerized ECG: Friend and Foe. The American Journal of Medicine February 2019;132(2):153–160. ISSN 0002-9343, 1555-7162. Publisher: Elsevier.

[3] Alpert JS. Can You Trust a Computer to Read Your Electrocardiogram? The American Journal of Medicine June 2012;125(6):525–526. ISSN 0002-9343.

[4] Bickerton M, Pooler A. Misplaced ECG electrodes and the need for continuing training. British Journal of Cardiac Nursing March 2019;14(3):123–132. Publisher: Mark Allen Group.

[5] Fawaz HI, 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. ISSN 1573-756X. Company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 4 Publisher: Springer US.

[6] Goldberger AL, Amaral LAN, 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 June 2000;101(23). ISSN 0009-7322, 1524-4539.

[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. ISSN 2156-7018.

[8] Wagner P, Strodthoff N, Bousseljot RD, Kreiseler D, Lunze FI, Samek W, Schaeffter T. PTB-XL, a large publicly available electrocardiography dataset. Scientific Data May 2020; 7(1):154. ISSN 2052-4463. Number: 1 Publisher: Nature Publishing Group.

[9] Wagner P, Strodthoff N, Bousseljot R, Samek W, Schaeffter T. PTB-XL, a large publicly available electrocardiography dataset (version 1.0.1). PhysioNet. 2020;.

[10] 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, (Under Review). Physiological Measurement ;.

[11] Pan J, Tompkins WJ. A Real-Time QRS Detection Algorithm. IEEE Transactions on Biomedical Engineering March 1985;BME-32(3):230–236. ISSN 0018-9294.

[12] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D. Scikit-learn: Machine Learning in Python. MACHINE LEARNING IN PYTHON ;6.

[13] Chollet F, others. Keras, 2015. Publisher: GitHub.

[14] Nelder JA, Mead R. A Simplex Method for Function Minimization. The Computer Journal January 1965;7(4):308–313. ISSN 0010-4620.

[15] SciPy 1.0 Contributors, Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods March 2020;17(3):261–272. ISSN 1548-7091, 1548-7105.

Address for correspondence:

Bjørn-Jostein Singstad
Rødbergveien 2b, Oslo, Norway
b.j.singstad@fys.uio.no

[1] code can be found here: https://www.kaggle.com/bjoernjostein/fcn-encoder-dense-physionet-challenge-2020