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

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

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

Abstract

The objective of this study was to classify 27 cardiac abnormalities based on a data set of 43101 ECG recordings. A hybrid model combining a rule-based algorithm with different deep learning architectures was developed.

We compared two different Convolutional Neural Networks, a Fully Convolutional Neural Network and an Encoder Network, a combination of both, and with the addition of another neural network using age and gender as input. Two of these combinations were finally combined with a rule-based model using derived ECG features. The performance of the models was evaluated on validation data during model development using hold-out validation. Finally, the models were deployed to a Docker image, trained on the provided development data, and tested on the Challenge validation set. The model that performed best on the Challenge validation set was then deployed and tested on the full Challenge test set. The performance was evaluated based on a particular Challenge score.

Our team, TeamUIO, achieved a Challenge validation score of 0.377, and a full test score of 0.206 for our best model. The score on the full test set placed us at 20th out of 41 teams in the official ranking.

1. Introduction

The 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 done by the automatic interpretation software, doctors have to read over the ECGs [3]. This is time-consuming for the doctors and requires a high degree of expertise [4]. There is clearly a need for better ECG interpretation algorithms.

Recent years have 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 Neural Networks (FCN) outperformed eight other CNN architectures compared. We also wanted to test the second-best architecture from their study which was an Encoder network. Finally, we assessed the integration of a rule-based algorithm within these models to test the performance of a CNN and rule-based hybrid classifier.

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

2. Methods

2.1. Data

To train the CNN models a data set containing 43.101 ECG recordings with corresponding information files describing the recording, patient attributes, and the diagnosis was used [6, 7]. The recording lengths varied across the different ECG signals, 83.4% were 5000 samples long. 98.5% of the recordings were sampled at a frequency of 500Hz, 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 [6]. The 27 labels to classify were One-Hot encoded, with each diagnosis rep-

resented 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]. Two models were tested without any modifications to the architecture other than changing the input and output layers to fit our input data and output classes. All output layers of each model used a Sigmoid activation function.

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

Age and gender data were passed into the simple neural network as integers, but in some information files, the age of the patient was not given and was assigned a value of -1. The gender data was transformed into integers, where a 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 parallel models, concatenated on the second last layer. This model was tested with and without a parallel dense layer[1].

2.4. Rule-based model

The rule-based algorithm used the raw ECG signal, without any padding or truncating, as input. R-peak detection [8], and heart rate variability (HRV) analysis was programmed to add relevant derived features to the algorithm. 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.

[1]All models and algorithms are available here: `https://www.kaggle.com/bjoernjostein/physionet-challenge-2020`

2.5. Model development

The models were trained and validated on the development data using hold-out validation with a split of 90% for training and 10% for validation. The first fold in a stratified K-fold was used with a random seed of 42 [9]. The splitting was arranged such that the distribution of diagnoses was the same in both the train and validation data.

During training, the Area Under the Curve (AUC) score on the validation set was used 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 [10], for each epoch that the AUC score did not improve. Early stopping [10] was triggered when the AUC score on the validation data did not improve over two successive epochs.

2.6. Threshold optimization

The prediction thresholds were optimized during model development. This was done by running the classifier on the hold-out validation data and receiving a score between 0 and 1 for each of the classes. The Nelder-Mead downhill simplex method [11, 12] was applied to optimize the threshold individually for the 27 classes. The Nelder-Mead downhill simplex method is used to find the local minimum of a function using the function itself and an initial guess of the variable of the function. The 27-element long array was optimized using the negative of the PhysioNet/CinC Challenge score [6]. To increase the possibility of finding the global maximum of the PhysioNet/CinC Challenge score, all elements in the 27-element long array was given 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. The value that gave the highest PhysioNet/CinC Challenge score was used as the initial guess for the Nelder-Mead downhill simplex method.

2.7. Model deployment

To obtain a valid score in the PhysioNet/CinC Challenge we submitted the models to the PhysioNet/CinC committee for testing on a Challenge validation and test set [6].

A Docker image was used to create a virtual Python environment for the model to be tested. During model deployment, the model was trained on the whole development set. The first three Challenge validation scores were obtained using AUC on the development data to schedule the reduction of the learning rate.

The two last Challenge validation scores were obtained using a learning rate scheduler. The learning rate schedule was programmed to be the same as in model development.

Model ID and name	Rule-based model	AUC	F1	F2	G2	Challenge score
A) FCN	No	0.875	0.381	0.446	0.230	0.348
B) Encoder	No	0.866	0.396	0.429	0.228	0.398
C) FCN + age, gender	No	0.877	0.368	0.438	0.222	0.385
D) Encoder + age, gender	No	0.828	0.334	0.389	0.190	0.333
E) Encoder + FCN	No	0.872	0.399	0.436	0.237	0.409
F) Encoder + FCN	Yes	0.872	0.361	0.413	0.203	0.348
G) Encoder + FCN + age, gender	No	0.866	0.400	0.434	0.233	0.395
H) Encoder + FCN + age, gender	Yes	0.866	0.356	0.405	0.198	0.338

Table 1. Scores were obtained by eight different models during model development. The models were evaluated by five different metrics, AUC, F1, F2, G2, and the PhysioNet/CinC Challenge score, during model development.

2.8. General parameters for both validation and testing procedures

For all models in both development and deployment, we used Adam optimizer, a batch size of 30, and binary cross-entropy as the 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.

Weights based on the number of occurrences of the different classes were calculated to deal with the skewed classes in the development data [9]. The calculated weights were passed to the model during training to give higher priority to rare diagnoses and lower priority to diagnoses that occurred more frequently.

3. Results

3.1. Scoring metrics

During model development, all models were validated on a subset of the development data using the metrics AUC (Eq 1), F_1-score (Eq 2), F_2-score (Eq 3), G_2-score (Eq 4), and the PhysioNet/CinC Challenge score, as seen in Table 1. On the Challenge validation set, we only obtained the PhysioNet/CinC Challenge score as seen in Table 2. After the evaluation of the performance on the full Challenge test set we were provided AUC (Eq 1), F_1-score (Eq 2), PhysioNet/CinC Challenge score, an Area Under the Precision-Recall Curve (AUPRC) score, and an accuracy 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 tested during the development phase, as seen in Table 1, were successfully deployed and obtained a score on the Challenge validation set, presented in Table 2.

Model ID and name	Rule-based model	Challenge score
B) Encoder	No	0.229
C) FCN + age, gender	No	0.302
D) Encoder + age, gender	No	0.272
F) Encoder + FCN	Yes	0.377
H) Encoder + FCN + age, gender	Yes	0.364

Table 2. The scores are obtained on the Challenge validation set and only the PhysioNet/CinC Challenge score was given. The Challenge validation set is a subset of the Challenge test set and not the final score in the challenge. The scores achieved on the Challenge validation set was used to select one model for deployment on the full Challenge test set.

The best score on the Challenge validation set was achieved by model H, an Encoder in parallel with an FCN with the rule-based algorithms added, as seen in Table 2. Model H was finally deployed and scored on the full Challenge test set [6]. The model achieved an AUC-score of 0.728, an F_1-score of 0.233, and a PhysioNet/CinC Challenge score of 0.206. This score brought us, TeamUIO, to 20th place in the PhysioNet/CinC Challenge 2020.

4. Discussion and conclusion

We chose to pad and truncate the signals to 5000 samples which were necessary to be able to feed the signal to

the CNN. The disadvantage was that some important information from segments of the ECG recordings might 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. Thus, the models that combined both CNN and rule-based algorithms used the entire signal when classifying the ECG.

Deployment of the models was done using two different ways of controlling the learning rate. The scores of models B, C, and D, on the Challenge validation set (Table 2), were obtained by using AUC on the development data to schedule the reduction of the learning rate. This might have contributed to overfitting indicated by the difference of the Challenge score of models B, C, and D in Table 1 compared with the same models in Table 2. The Challenge score achieved on the Challenge validation data by model F and G (Table 2), were obtained using a learning rate scheduler [10]. The PhysioNet/CinC Challenge scores achieved on the Challenge validation data by model F and G (Table 2) are more consistent with the PhysioNet/CinC Challenge score obtained on the development data in Table 1 for the same models. In summary, our result indicates that the models, deployed on the Challenge validation set, which kept the same training schedule as in the development model, seem to avoid overfitting and perform better on unseen data.

During the model development, we observed that the Encoder (model B) performed better than the FCN (model A) on the PhysioNet/CinC Challenge score as seen in Table 1. A plain FCN (model A) was not scored on the Challenge validation set and thus it remains unclear which of a plain FCN or a plain Encoder perform best on unseen data like the Challenge validation data.

The Encoder (model B) decreased in performance when a parallel model for age and gender was added (model D) during model development (Table 1). However, the performance increased when the Encoder (model B) was added a parallel model for age and gender (model D) when scoring the models on the Challenge validation set (Table 2). Based on the PhysioNet/CinC Challenge score, during model development (Table 1), the FCN (model A) improved in performance when adding a parallel model for age and gender (model C). However, we did not deploy a plain FCN (model A) to the Challenge validation set and thus it remains unclear if the FCN + age and gender (model C) would outperform the FCN (model A) on the Challenge validation set.

During model development (Table 1), the Encoder + FCN (model E) and the Encoder + FCN + age, gender (model G), decreased in performance when adding the rule-based model (model F and H). However, the PhysioNet/CinC Challenge score, achieved by model F and G on the Challenge validation set (Table 2), was better than the PhysioNet/CinC Challenge score achieved by the same models during model development (Table 1). Our results indicate that the hybridization of CNN with a rule-based model could improve the 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.

References

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

[2] Smulyan H. The Computerized ECG: Friend and Foe. The American Journal of Medicine February 2019;132(2):153–160.

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

[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.

[5] Fawaz HI, et al. Deep Learning for Time Series Classification: A Review. Data Mining and Knowledge Discovery July 2019;33(4):917–963.

[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. Physiol Meas 2020;(Under Review).

[7] Goldberger AL, et al. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation June 2000;101(23).

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

[9] Pedregosa F, et al. Scikit-Learn: Machine Learning in Python. Journal of Machine Learning Research 2011; 12(85):2825–2830.

[10] Martín Abadi, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015.

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

[12] Virtanen P, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods March 2020;17(3):261–272.

Address for correspondence:

Bjørn-Jostein Singstad
Sem Sælands vei 24, 0371 Oslo, Norway
b.j.singstad@fys.uio.no