Denoising autoencoder for ECG signals

Vincent Eurasto
Aalto University
vincent.eurasto@aalto.fi

December 13, 2022

Abstract

Electrocardiogram (ECG) signals are known to be prone to noise due to their small amplitude and noisy recording environments. As the ECG signal is used for diagnosis purposes and analysis of heart diseases, a sound-quality recording is necessary. One of the most common noise sources comes from the mains, which is in the range of 50 Hz in Europe. There are numerous different denoising techniques to enhance the quality of ECG signals. These methods range from traditional filtering methods to artificial intelligence techniques. In this study, a machine-learning-based denoising autoencoder is built and trained to denoise ECG recordings. In addition, the results of the developed method are presented in detail and compared to a more traditional filtering technique.

1 Introduction

An electrocardiogram (ECG) is a test that can be used to monitor a patient's heart rhythm and electrical activity. The electrical signals produced by the heart are measured using electrodes attached to the patient's skin. The heart activity is recorded with a computer and analyzed by a doctor. An ECG is usually performed to investigate symptoms of possible heart problems. [1]

Despite the usefulness of ECG, the recorded signals are prone to be interfered with by various noise sources. The primary noise sources include power line interference, baseline drift, and muscle contractions. As the ECG signal is used for diagnosis purposes and analysis of heart diseases, a sound-quality recording is necessary. [2] Thus, various denoising techniques have been developed to enhance the quality of ECG signals. These methods range from simple Butterworth filtering to machine learning (ML) based filters [3].

This study aims to implement and train a neural network-based filtering technique using a denoising autoencoder. An autoencoder is a special neural network trained to copy its input to its output, and they are often used for data compression [4]. However, the model can also be trained to perform denoising by training the autoencoder using a corrupted signal as input, and the original signal as the target [5]. Thus, by feeding a noisy ECG signal to the autoencoder, the model should be able to reconstruct the original signal.

In addition, the results of the developed machine-learning model are presented in detail and compared to a more traditional filtering technique.

2 Dataset

Since we are using a supervised ML method in the study, the training step requires us to have a rich and high-quality data set. The data used in this analysis were downloaded from PhysioNet, which offers a rich resource of a variety of different biological signals [6].

The downloaded data set is an ECG data set that was offered during The PhysioNet/Computing in Cardiology Challenge 2017. The data set contains over 8500 single lead ECG recordings lasting from 9 seconds to just over 60 seconds. The ECG recordings were sampled at 300 Hz and they were band pass filtered. In total, the data set contains four classes: normal rhythm, atrial fibrillation (AF) rhythm, other rhythms, and noisy recordings. [7]

Before using the data, the ECG recordings were preprocessed. First, the noisy signals and all the recordings that were less than 10 seconds were removed from the data set. Secondly, all the records that were longer than 10 seconds were cut to be exactly 10 seconds. In the end, the data set contained 8229 recordings in total. The total number of recordings in each class is summarized in Table 1. Examples of each class are shown in Figure 1.

Type	Number of recordings
Normal	5040
AF	737
Other	2452

Table 1: Data profile for the data set.

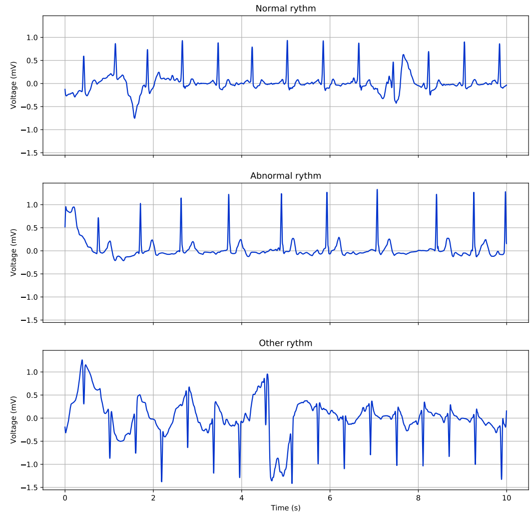

Figure 1: Examples of ECG waveforms from each class after the preprocessing step.

3 Implementation

ECG signals are prone to be interfered with by various noise sources which mainly include power line interference, baseline drift, and muscle artifacts. [2] Each of these noise sources lay in different frequency bands and is different in terms of amplitude. A summary of the primary noise sources can be seen in Table 2.

Type	Frequency range	Amplitude
Power-line interference	50 ± 0.2 Hz	Up to 50 % FSD
Baseline wander	0.15 – 0.3 Hz	Up to 15 % FSD
Muscle artefacts	0 – 1000 Hz	Up to 10 % FSD

Table 2: Predominant noises in ECG signals. The full-scale deflection (FSD) is the peak-to-peak ECG amplitude. [3]

The useful bandwidth of an ECG signal can range from 0.5 Hz to 50 Hz [8], which overlaps with the noise bandwidths. Thus, removing noise from ECG signals can be difficult without losing valuable information.

The final training data set was created by corrupting each recording by adding each type of interference with random amplitudes, frequencies (but still within the given limits given by Table 2), and phase shifts. In addition, the training data was normalized using Min-Max normalization. Finally, the data set was split into training and test sets using a 90-10 split with random shuffling. In other words, the model was trained with 90 % of the total data and the rest was used for testing.

Since we are training an autoencoder to remove the noise, the augmented noisy recordings are fed to the network and are compared to the corresponding non-noisy recording. The autoencoder then tries to adjust the weights so that the noise can be removed from various types of ECG signals.

The autoencoder was built using TensorFlow Python Framework [9]. The structure of the designed model is depicted in Figure 2. The model uses mean-squared error (MSE) as the loss function and it was trained for a total of 20 epochs. The training and validation error as a function of epochs is presented in Figure 3.

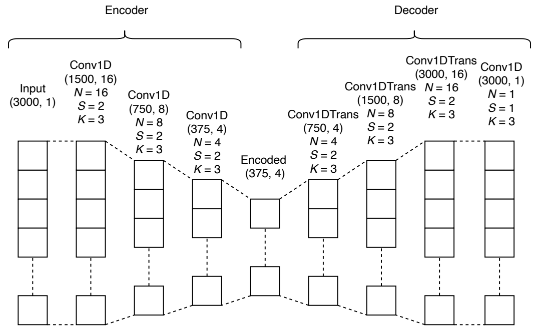

Figure 2: Structure of the autoencoder used in this study. The model takes a 3000×1 vector as input (the noisy 10-second ECG signal). The ECG signals are encoded to contain only 375 data points with the help of convolutional layers (Conv1D) using filters N for each convolution. Finally, the original ECG signal is reconstructed from the encoded signal using deconvolutional layers (Conv1DTrans). The kernel sizes K and strides S can be seen in the figure.

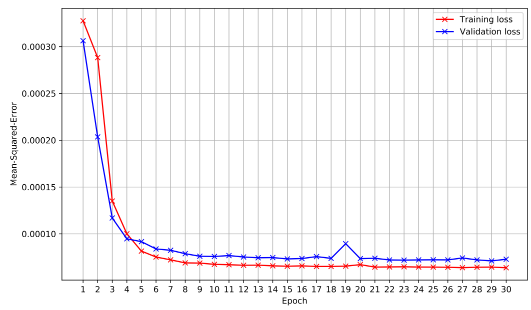

Figure 3: The training and validation error as a function of epochs.

4 Results

In this chapter, the results of using the designed and trained autoencoder on the noisy test set are presented. In addition, the proposed method is compared to a more traditional filtering method.

The root-mean-squared error (RMSE) of the reconstructed ECG signals is the main assessment

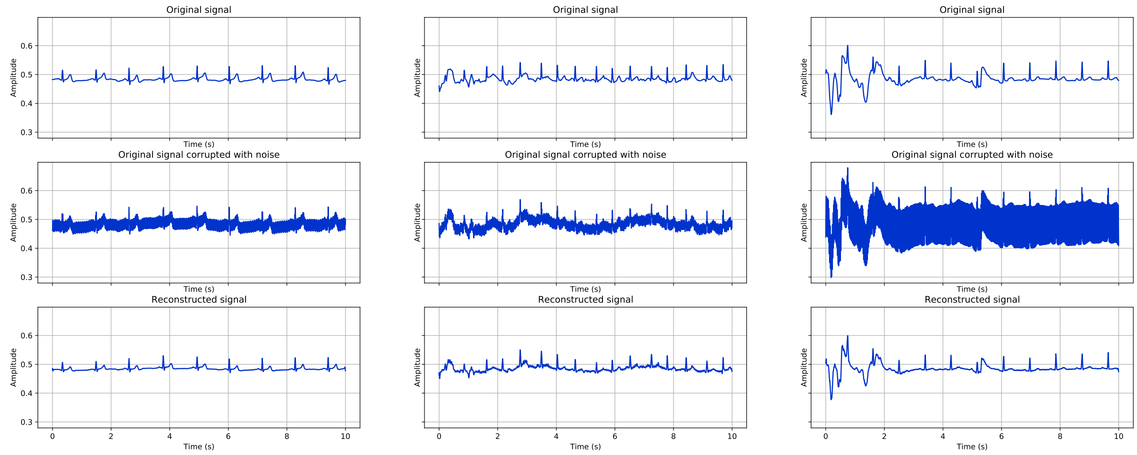

Figure 4: Results of using the designed denoising autoencoder on the noisy ECG recordings. The three different classes, normal, AF, and other, (from left to right) are presented in this figure. The autoencoder is clearly capable of enhancing the quality of the signal by removing noise.

method in this analysis. However, some of the reconstructed signals are presented in Figure 4 for visualization purposes. The filtering performance of the autoencoder was compared to a 6th-order Butterworth low-pass filter with a cutoff frequency of 45 Hz combined with a 6th-order Butterworth band-stop filter at 50 Hz. The RMSEs of both methods are presented in Table 3.

Method	**RMSE**
Denoising autoencoder	0.010726
Low-pass + band-stop filter	0.012757

Table 3: Average root-mean-squared errors using two different methods, the designed denoising autoencoder and a 6th order Butterworth low-pass filter with a cutoff frequency at 45 Hz combined with a 6th order Butterworth band-stop filter at 50 Hz.

From Table 3, we observe that the traditional filtering technique is similar to the autoencoder method in terms of performance for this data set. This can also be seen in Figure 5.

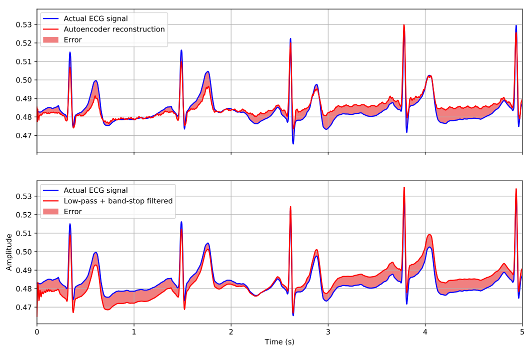

Figure 5: An example of using the designed denoising autoencoder and the traditional filter on a noisy ECG recording.

5 Summary and discussion

In this study, a denoising autoencoder was designed and trained to remove noise from ECG recordings. The ECG data were downloaded from PhysioNet and it was preprocessed before training the model. The autoencoder was developed using TensorFlow Python Framework [9] and the model was trained until the training and validation error converged.

After the training, the performance of the model was analyzed. In addition, the results were compared to a traditional filtering method. Indeed, both methods can be used to denoise the recordings and enhance the quality of the signals. However, one should consider the extra complexity and timing requirements needed to train the autoencoder neural network. In contrast, it's relatively simple to implement conventional digital Butterworth filters.

Nevertheless, after computing the average RMSE of the whole denoised dataset, it can be seen that the designed autoencoder performs slightly better than the traditional filtering technique. However, it's still challenging to assess the quality of the denoising process by only analyzing the RMSE error. One method to assess the quality of the filtering is to classify the ECG signals before and after the filtering to see if the filtering process changes the classifier's performance (i.e., the denoising process might change or remove some important features).

References

[1] NHS, "Electrocardiogram (ecg)." Online [Visited 15/8/2022], 2021. https://www.nhs.uk/conditions/electrocardiogram/.

[2] S. L. Joshi, R. A. Vatti, and R. V. Tornekar, "A survey on ecg signal denoising techniques,"

2013 International Conference on Communication Systems and Network Technologies, pp. 60–64, 2013.

[3] S. Chatterjee, R. S. Thakur, R. N. Yadav, L. Gupta, and D. K. Raghuvanshi, "Review of noise removal techniques in ecg signals," *IET Signal Processing*, vol. 14, no. 9, pp. 569–590, 2020.

[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[5] TensorFlow, "Intro to autoencoders." Online [Visited 15/8/2022], 2022. https://www.tensorflow.org/tutorials/generative/autoencoder.

[6] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. Ivanov, R. Mark, J. Mietus, G. Moody, C.-K. Peng, and H. Stanley, "Physiobank, physiotoolkit, and physionet : Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, pp. E215–20, 07 2000.

[7] G. D. Clifford, C. Liu, B. Moody, L.-w. H. Lehman, I. Silva, Q. Li, A. E. Johnson, and R. G. Mark, "Af classification from a short single lead ecg recording: The physionet/computing in cardiology challenge 2017," *2017 Computing in Cardiology (CinC)*, pp. 1–4, 2017.

[8] E. Company-Bosch and E. Hartmann, "Ecg front-end design is simplified with microconverter." Online [Visited 15/8/2022], 2003. https://www.analog.com/en/analog-dialogue/articles/ecg-front-end-design-simplified.html.

[9] TensorFlow, "Tensorflow." Online [Visited 15/8/2022], 2022. https://www.tensorflow.org/.