Final Report - Project B

Compression For Continuous Long-Term Electrocardiography Recordings

Students: Noam Ben-Moshe and Noa Cohen
Supervised by: Sheina Gendelman and Prof. Joachim Behar

Semester: Winter, 2021
Date: 17/10/2021

In collaboration with: Artificial Intelligence in Medicine Laboratory, Faculty of Biomedical Engineering

Contents

List of Figures

List of Tables

1 Abstract

This project researches compression algorithms to create a sparse representation of a continuous long-term electrocardiography (ECG) recording while preserving its diagnostic information. This will enable storing and transmitting continuous long-term ECG recordings, crucial for a portable collection system that will collect mobile ECG data continuously recorded for hours to days long.

2 Introduction

An electrocardiogram (ECG) records the electrical signal from the heart. A long-term ECG provides insight into the behavior of the heart in the everyday life of the patient, for long periods of time. Such recordings have very large memory requirements and require compression for storing and transmitting. When lossy compression is applied to biomedical signals such as ECG, avoiding loss of important diagnostic data elements is critical. In this project the goal is to develop an ECG compression method that will have an efficient compression ratio and will preserve the diagnostic equivalence.

We implemented two baselines, one is a lossy method that is transformed based presented in [1], [2] and the other is parameter extraction based which is a deep convolutional autoencoder presented in [3]. We then tried to improve the deep convolutional autoencoder method and the results are presented in this report.

3 Literature Review

ECG compression methods have been researched in various groups and contexts. Compression methods are divided to lossless methods and lossy methods, differing by whether a reconstruction error exists.

In lossless methods for continuous long-term ECG recordings, in which all the original data remains after reconstruction, the compression ratio is much lower which may be inefficient. These methods can be used to acquire a lower bound on the compression ratio. In "A low complexity lossless compression scheme for wearable ECG sensors"[4] Deepu and Lian proposed a technique that achieves an average compression ratio of 2.38 on long-term ECG signals, by using linear predictors in the time domain.

In lossy methods compression ratio is highly efficient, but the reconstruction error must be medically acceptable. These methods can be applied directly on the time-domain signal, transform based, or use parameter extraction. The wavelet transform is commonly used in transformed based lossy compression methods for long-term ECG recording, as [1] and [2], due to its ability to analyze the signal both in time as well as frequency domains simultaneous.

Deep learning methods enabled a leap in many domains, and are also used for compression tasks in various domains, e.g. with autoencoder neural networks. Such architectures encode the signal to a latent space, and later decode the latent space representation into a time signal. An autoencoder for compression of long-term ECG signals is presented in [3].

Since the data dealt with is medical and lossy compression methods are applied, medical evaluation is important in addition to the mathematical quality measures to assess whether diagnostic data elements are preserved. In ECG reconstruction a distortion method for comparing the distortion between original ECG signal and reconstructed ECG signal that is commonly used was introduced in [5]. The weighted diagnostic distortion (WDD) is based on diagnostic features and thus very useful.

4 Database

The database used in this project is The University of Virginia Database (UVAF) consisting of RR interval time-series and rhythm annotations. This database gathers the electrocardiogram (ECG) recordings of patients over 39 years for whom University of Virginia health system physicians ordered Holter monitoring from December 2004 to October 2010.

This database contains annotated files from 2,891 individual patients. Each file contains relative timestamps for each heartbeat, their type and rhythm type. These annotations were automatically generated by the medical monitor Philips Holter software, based on the automated analysis of the ECG trace. Part of these files (52.3%) were reviewed by medical school students. Each record lasts approximately 24 hours. The database contains normal sinus rhythms (NSR) and various abnormal rhythms. The three most common abnormal rhythms categories in the database are atrial fibrillation (AF), sinus bradycardia (SBR), and supraventricular tachyarrhythmia (SVTA).

To improve running time while training the HDF5 libary was used to store the data. The HDF5 is a data software library and file format that manages and stores data. The main purpose of HDF5 is fast I/O processing and storage.

5 Pre-Processing

The database contains 2,891 patients with 24 hours recording each thus totalling over 69,000 hours of continuous data. This database was also used in [6] and consists of ECG signals sampled at a frequency of 200 Hz, with four abnormal rhythms and NSR. To make the data cleaner and remove noise the data is passed through a filter. The filter is composed of a band pass filter and a notch filter. The filter is a Butterworth filter of type Band Pass Filter from 0.33Hz to 50Hz of order 75 as suggested in [7] and a Hamming notch filter at 60Hz. An example for the power spectrum density of an ECG signal before and after applying the filter is shown in 1. An example of how the filter effected the ECG signal in the time domain is depicted in 2. In order to compare the results to those presented in [3] we resample all ECG signals to 360 Hz. We then rescaled the data to the range of [0,1].

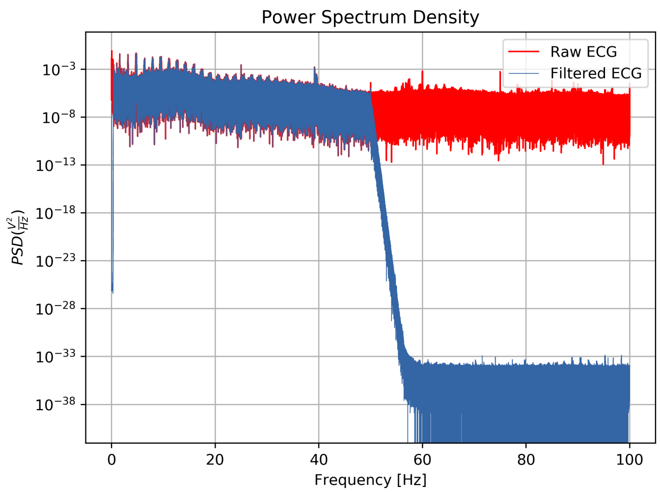


Figure 1: Power spectrum density of an ECG signal before and after applying the filter

During the training each data-point is a window of 2000 samples. Since different windows of the same patient are related, in order to maintain independent train and test sets we randomly split the patients into train and test.

6 Evaluation Criteria

To evaluate the quality of the reconstructed signals we use five performance measures. In all equations S_o indicates the original signal, S_r indicates the reconstructed signal, S_m is the mean of the original signal, namely $S_m = mean(S_o)$, and D is the length of the signal.

6.1 Root Mean Square

Root mean square (RMS) measures the error between the reconstructed signal and the original signal.

$$RMS = \sqrt{\frac{\sum_{i=0}^{D-1} (S_o(i) - S_r(i))^2}{D}} \tag{1}$$

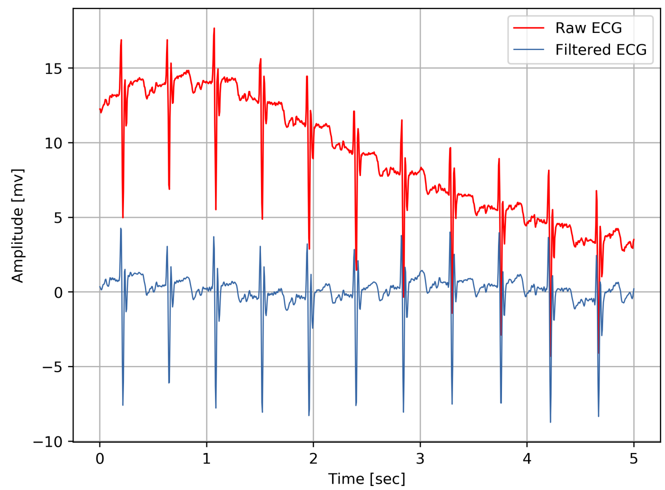


Figure 2: Raw ECG segment and the ECG segment after passing through the filter

6.2 Percentage Root Mean Square Difference

Percentage root mean square difference (PRD) also measures the error between the reconstructed signal and the original signal, dividing not by the length of the signal but by the summation of the samples in the signal to the power of two.

$$PRD(\%) = 100 * \sqrt{\frac{\sum_{i=0}^{D-1} (S_o(i) - S_r(i))^2}{\sum_{i=0}^{D-1} (S_o(i))^2}} \tag{2}$$

6.3 Signal To Noise Ratio

Signal to noise ratio (SNR) measures the degree of noise energy introduced by the compression, in dB scale.

$$SNR = 10 * log\left(\frac{\sum_{i=0}^{D-1} (S_o(i) - S_m)^2}{\sum_{i=0}^{D-1} (S_o(i) - S_r(i))^2}\right) \tag{3}$$
$$where\ S_m = mean(S_o)$$

6.4 Quality Score

Quality score (QS) quantifies the performance of the compression by taking into acount both the compression ratio and the PRD value as a measure of the reconstruction error.

$$QS = \frac{CR}{PRD} \tag{4}$$

7 Baselines

7.1 Wavelet Baseline

Since ECG is a periodical signal, multiple papers reviewed the results of the wavelet transform as a compression method, and we have chosen it as our non-deep lossy baseline method. We rely on the following paper [1] and on the GitHub repository [8].

The flow of the Wavelet Transform based compression method we examined is as follows:

1. Wavelet decomposition.
2. Coefficient thresholding.
3. Coefficients scaling and quantization.
4. Compressing wavelet coefficients and binary map.
5. Decompressing and Reconstructing signal.

For this workflow, six hyper parameters had to be adapted. The first two define the wavelet to be used in the transform – wavelet family and type, we choose the wavelet Bior 4.4 as suggested in [1]. The third hyper parameter crucial for the transformation is the number of decomposition levels, choosing 5 levels as used in [1] yielded fine results on our data as well. Next, we divided the coefficients into 3 groups, approximation coefficients of highest level, detailed coefficients of highest level and detailed coefficients in all other levels, and for each group we needed a coefficient threshold to be used in the second stage of the flow. The thresholds that were chosen were the ones suggested in [1] after performing a grid search to ensure they are optimal for our data as well. For approximation coefficients of highest level the threshold was 99.9% , detailed coefficients of highest level the threshold chosen was 97% and for the detailed coefficients in all other levels the threshold was 85%.

7.2 Deep Baseline

The deep baseline is a deep convolutional autoencoder based on the paper [3]. The architecture used is depicted in Figure 3. In this baseline every ECG window of 2000 samples is encoded into 62 features. As a result the compression ration (CR) is fixed.

$$CR = \frac{|S_o|}{|S_r|} = \frac{2000}{62} = 32.25 \tag{5}$$

7.2.1 Training

The training was done using 100 ECG windows from each individual. Each window consisting of 2000 samples taken at a frequency of 360Hz. The patients were divided into two groups, training set and test set, each saved in an individual hdf5 file to ensure there is no data leakage. During the training 80% of the patients in the training set were used for the training and the remaining 20% were used for validation. The test set, which is independent of the training set,

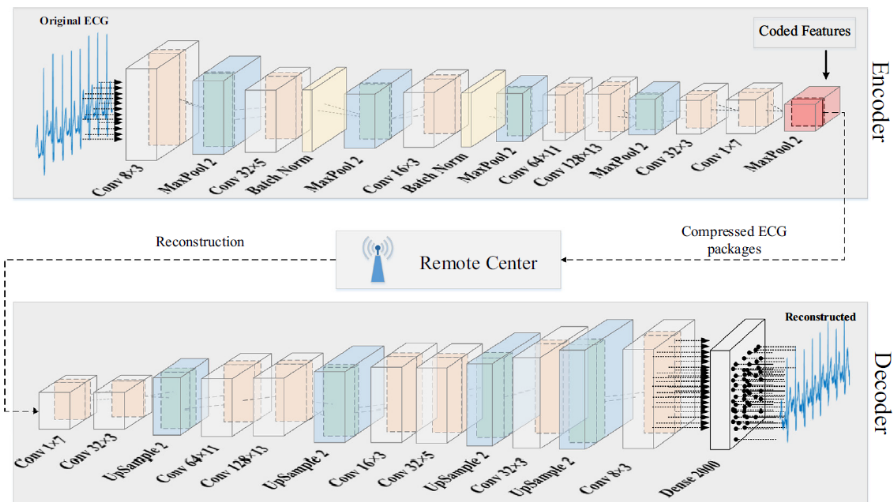


Figure 3: Autoencoder architecture as presented in [3]

included 20 NSR (Normal Sinus Rhythm) patients. Implementation was created using pyTorch. The setting follows [3] where an Adam Optimizer is employed with initial learning rate of 0.001, weight decay of 1e-5 and batch size of 32, for 100 epochs.

7.3 Baselines Comparison

We compare the results of our implementation with the results of the original paper[3], and the results of our wavelet baseline based on [1]. The results from our implementation of [3] were obtained with the same number of patients and training parameters as mentioned in the paper. The wavelet parameters were taken from the recommendation in [8] that is based on [1]. This yielded an average compression ratio of 8.5, while the deep convolutional autoencoder has a fixed compression ratio of 32.25. In order to compare methods with a different compression ratio the quality score (QS) may be used, but to be able to compare all evaluation criteria another parameter search was performed, to acquire best possible results with an compression ratio similar to the deep method. The chosen thresholds are 96% for approximation coefficients of highest level, 20% for detailed coefficients of highest level, and 10% for the detailed coefficients in all other levels. This yielded an average compression ratio of 32.34, and other evaluation criteria are depicted in 1 under Wavelet Baseline - Adapted Parameters.

As shown in 1, in every evaluation criteria the results obtained by our implementation were not as good as the results presented in [3]. One reason for that is that a different data set was used for training and evaluating. For

Criteria	Autoencoder (reported results)[3]	Deep Baseline	Wavelet Baseline[2]	Wavelet Baseline (Adapted Parameters)
RMS	0.013	0.114	0.024	0.078
PRD	2.73%	22.098%	19.383%	61.791%
SNR	23.96 dB	1.384dB	14.676 dB	4.404 dB
QS	13.38	1.664	0.475	0.558
CR	32.25	32.25	8.532	32.342

Table 1: Comparison between baselines

example, noisy windows in the dataset that do not resemble a regular ECG signal may interfere with both training and evaluating. Another reason for the difference might be the fact that in the paper [3] different windows from the same patient were in the training set and the test set, meaning there was a data leakage [1].

The Deep Baseline and the Wavelet Baselines were both tested on the same data. The Wavelet Baseline yielded better results in RMS, PRD, and SNR, though in the QS and the CR the Deep Baseline yielded better results. When comparing the Deep Baseline and the Wavelet Baseline with the adapted parameters which both have similar CR the Deep Baseline yielded better results in PRD and QS, though RMS and SNR remain better in the Wavelet Baseline with the adapted parameters. Reconstruction of a typical ECG segment from the data by the baselines methods is shown in 4.

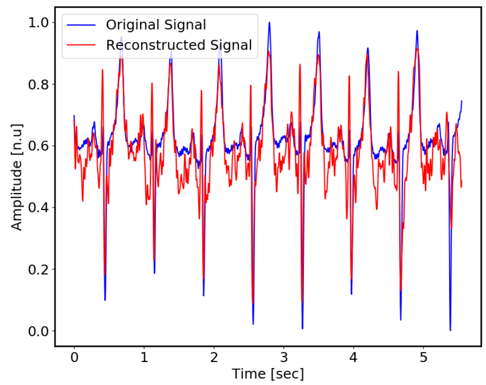
8 Deep Method Improvements

In the Deep Baseline model we aimed to obtain best reconstruction results without changing the architecture used in [3]. For this purpose we implemented the architecture of the encoder and the decoder with pyTorch and tried to train it with the hyper-parameters that were used in [3]. These parameters are Adam Optimizer with initial learning rate of 0.001, weight decay of 1e-5, batch size of 32 and 48 as number of patients. In order to try to improve the results obtained by the deep method a hyper parameter search was done and a regularization term was added.

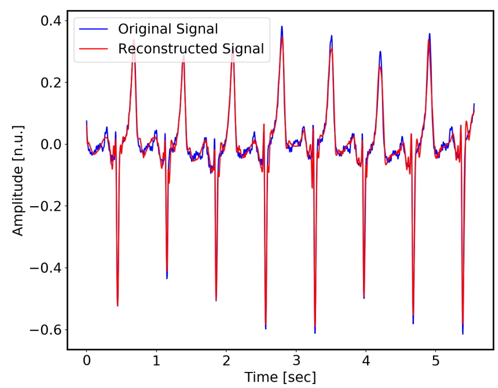
8.1 Hyper Parameters search

A Bayesian search was done in order to find the hyper parameters that will yield the best results in the sense of the evaluation criteria described is section 6.

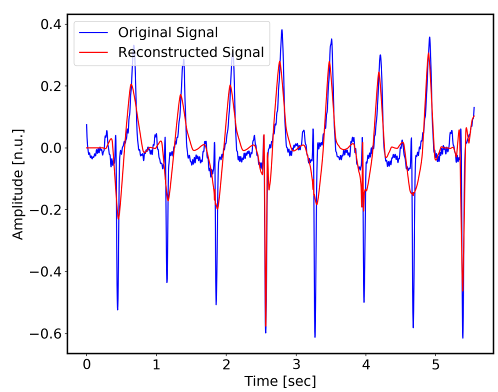
[1]In the experiment at [3] they take 100 windows from 48 patients = 4800 windows and split them to 10% (480) test, 10% (480) validation, and 80% (3840) train. All three numbers do not divide by 100 and so it is assumed that some of the patients have their windows in more than one group.



(a) Reconstructed segment by our implementation of the deep baseline CR-32.25



(b) Reconstructed segment by the wavelet baseline CR-8.532



(c) Reconstructed segment by the wavelet baseline with adapted parameters CR-32.342

Figure 4: Graphic comparison of original and reconstructed ECG signal

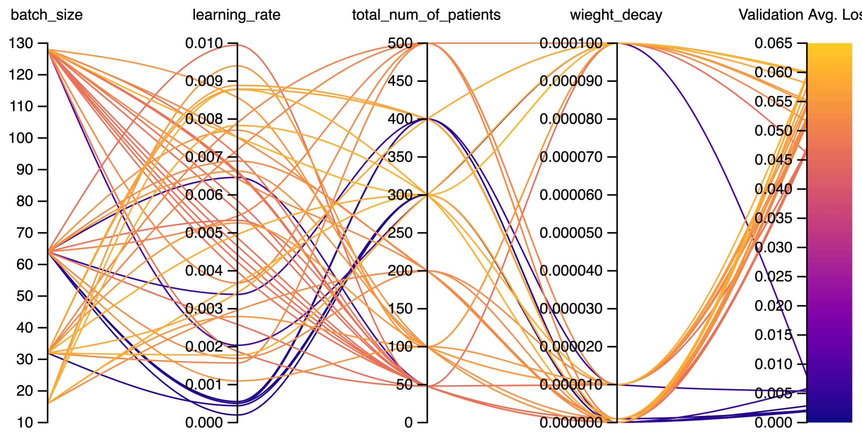


Figure 5: Bayesian search for hyper-parameters using [9]

A Bayesian search is a search that uses a Bayesian Optimization. In this approach that Bayes Theorem is used to direct the search in order to find the minimum or maximum of an objective function.

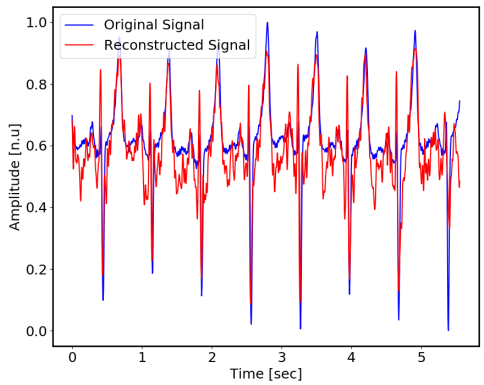
The hyper parameters that were searched were batch size (range [16 - 128]), learning rate (range [1e-5 - 0.01]), weight decay (range [1.0e-04 - 1.0e-07]). Another hyper parameter that was searched was the number of patients used for training. The range searched was 48 patients to 500 patients.

Figure 5 depicts the different combinations of hyper-parameters that were tested and on the right is the value obtained in the loss function which is mean squared error loss. The hyper-parameters that achieved the best values are: batch size - 64, learning rate - 0.0005, weight decay - 1e-07, and 400 patients.

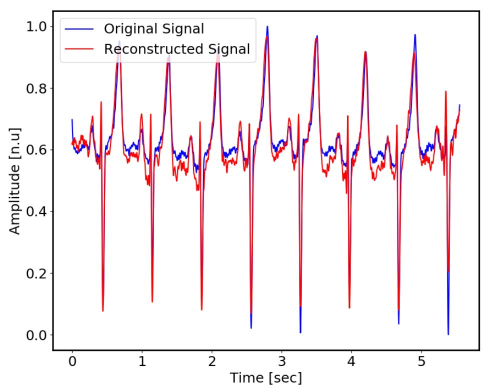
After the Bayesian search the model was trained with the best parameters as found in the Bayesian search, and was tested on the test set. The results are presented in 2. The model that was trained with the best hyper-parameters reached better results than the Deep Baseline though the results were not as good as the original paper [3]. In figure 6 there is an example of the same ECG segment compressed and reconstructed by the Deep Baseline model and the best hyper-parameters model. You can see how the signal reconstructed by the best hyper-parameters model is less noisy.

8.2 Regularisation

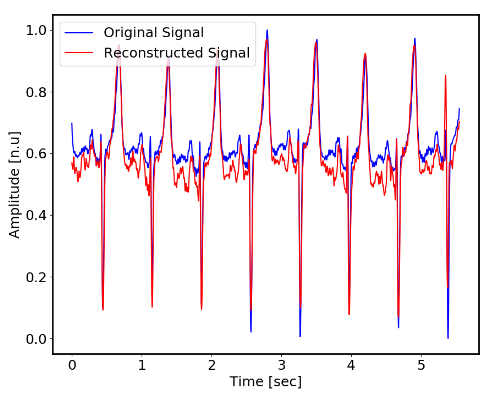
The ECG signal is characterized by its diagnostic peaks, with the R point being the most prominent one as can be seen in 7. In addition, the ECG measured from patients is usually noisy. Consequently, a regularization that would preserve the diagnostic characteristics and would penalize noisy solutions was desired, the Total Variation (TV) regularisation does that. An example of how the TV decreases noise but keeps the shape of the signal is in figure 8.



(a) Reconstructed segment by our implementation of the deep baseline



(b) Reconstructed segment by our implementation of the deep baseline after tuning hyper parameters



(c) Reconstructed segment by our implementation of the deep baseline after training with TV regularization

Figure 6: Graphic comparison of original and reconstructed ECG signal

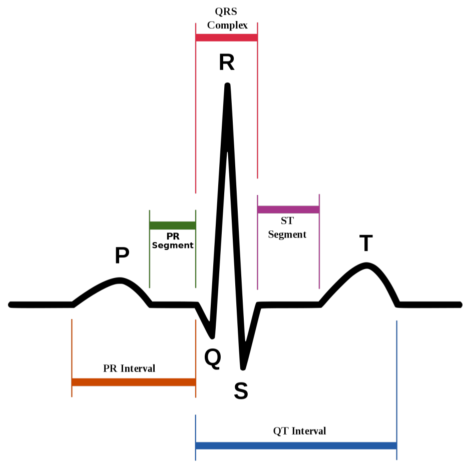


Figure 7: Schematic diagram of normal sinus rhythm for a human heart as seen on ECG

The TV regularization can be formalized as follows:

$$TV(S_r) = \sum_{i=0}^{D-1} |S_r(i+1) - S_r(i)| \tag{6}$$

The MSE part of the loss is in charge of keeping the reconstruction similar to the original signal and the TV regularisation is in charge of penalising the noise, and preferring smooth signals.

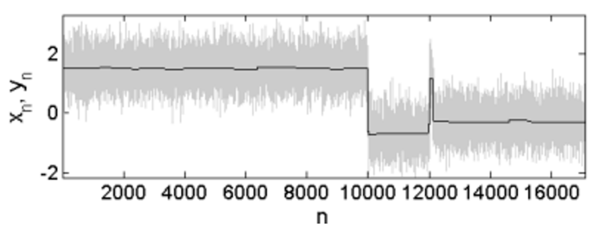


Figure 8: Application of 1D TV denoising to a signal. Gray is the original signal, black is the denoised signal. Image from slides by Prof. Guy Gilboa

The ECG signal is characterized by its diagnostic peaks and has major changes, which result in a larger TV value, that are important and should not be flattened. Applying the TV regularization should be held with care in order to smooth the noise without loosing the diagnostic data. Two option were applied. The first is weighting the terms with a factor denoted as λ resulting in a loss of the form:

$$MSE(S_o, S_r) + \lambda \cdot TV(S_r) \tag{7}$$

The value of the lambda weight would be chosen using a Bayesian search. The second option is applying a threshold denoted as α on the values summed in the TV, and omitting drastic changes. This should penalize on minor changes, but not on major changes as R peaks, resulting in smoother signals. The way the threshold was applied is described in equation 8. Here too a Bayesian search was used to determine the value of the threshold.

$$\begin{aligned} diff &= \hat{X}[n+1] - \hat{X}[n]] \\ threshold &= \alpha \cdot max_n X[n+1] - X[n] \\ TV_{th}(\hat{X}) &= \sum_{n=0}^{D-1} |diff[n]| \cdot \mathbb{I}_{(diff[n]<threshold)} \end{aligned} \tag{8}$$

In the Bayesian search we checked λ values in the range of [1e-12,1] and α values from the values [0.25,0.5,0.75,1]. Best results were reached with λ=1e-09 and α=1. The results on the test set are listed in table 2. The results show that the only criteria that was better in the model with the regularization was the PRD. An example of an ECG window reconstructed by the model trained with the TV regularization is depicted in figure 6.

Criteria	Original Paper[3]	Deep Baseline	After Bayesian Search	With Regularization
RMS	0.013	0.114	0.09	0.093
PRD	2.73%	22.014%	19.436%	18.792%
SNR	23.96 dB	1.384dB	3.743dB	3.353dB
QS	13.38	1.664	2.413	2.198
CR	32.25	32.25	32.25.	32.25

Table 2: Comparison between Deep Methods

9 Future Work

In this project we focused on compressing NSR ECG and evaluating the signal reconstruction with quality assessments such as RMS. When lossy compression is applied to biomedical signals such as ECG, avoiding loss of important diagnostic data elements is critical. Therefore, in addition to regular signal reconstruction quality assessments it will be interesting to evaluate the reconstruction using evaluation criteria that are specific for ECG as presented in [5]. It will be interesting to try to build a loss function that will be designated to ECG signals and will make the deep network learn to preserve the diagnostic critical elements of the ECG. As long ECG recordings are designated to be collected trough protable devices it is expected that some of the ECG collected will be too noisy and will not be usefull for diagnositc evaluation. Consequently, it will be interesting to use a quality assessment of an ECG window to decide if to use it or not, if a window has diagnostic data compress and save it, if a window is too noisy do not save it. Another interesting option to research is training an auto-encoder with ECG recordings from sick patients and see if it improves the ability to compress and reconstruct ECG while preserving the diagnostic evaluation. Also exploring different deep architectures that are not CNN based for encoding and decoding might be able to reach better results.

10 Summary

Since a long-term ECG provides insight into the behavior of the heart for long periods of time but also has very large memory requirements it requires compression for storing and transmitting. The compression method must be both relevant in terms of having an efficient compression ratio, and reliable in terms of preserving the diagnostic equivalence.

In this project we compared five models, two baselines and adapted baseline and 2 improvements to those baselines. One baseline is a lossy method based on the Wavelet Transform, and the second is a deep convolutional autoencoder. A comparison between the baselines is listed in 1. Using the Wavelet Transform achieved better equivalency but with a compression ratio that is inefficient. The deep convolutional autoencoder on the other hand has a much better compression ratio but the values of reconstruction quality criteria did not reach those presented in [3]. The three other models we implemented were Wavelet Baseline with adapted parameters, Auto-Encoder after Bayesian Search and Auto-Encoder with regularization. A comparison between the deep methods is listed in 2. From the comparisons it is seen that the model that reached the best QS is the Deep Auto-encoder after the Bayesian Search. Future work on diagnostic evaluation would shed light on the quality of the reconstruction for medical usage.

References

[1] B. A. Rajoub, "An efficient coding algorithm for the compression of ECG signals using the wavelet transform," *IEEE Transactions on Biomedical Engineering*, vol. 49, no. 4, pp. 355–362, 2002, ISSN: 00189294. DOI: 10.1109/10.991163.

[2] M. Elgendi, A. Mohamed, and R. Ward, "Efficient ECG Compression and QRS Detection for E-Health Applications," *Scientific Reports*, vol. 7, no. 1, pp. 1–16, Dec. 2017, ISSN: 20452322. DOI: 10.1038/s41598-017-00540-x. [Online]. Available: www.nature.com/scientificreports.

[3] O. Yildirim, R. S. Tan, and U. R. Acharya, "An efficient compression of ECG signals using deep convolutional autoencoders," *Cognitive Systems Research*, vol. 52, pp. 198–211, Dec. 2018, ISSN: 13890417. DOI: 10.1016/j.cogsys.2018.07.004.

[4] C. Deepu and Y. Lian, "A low complexity lossless compression scheme for wearable ecg sensors," in *2015 IEEE International Conference on Digital Signal Processing (DSP)*, 2015, pp. 449–453. DOI: 10.1109/ICDSP.2015.7251912.

[5] Y. Zigel, A. Cohen, and A. Katz, "The weighted diagnostic distortion (WDD) measure for ECG signal compression," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 11, pp. 1422–1430, Nov. 2000, ISSN: 00189294. DOI: 10.1109/TBME.2000.880093.

[6] A. Chocron, J. Oster, S. Biton, F. Mendel, M. Elbaz, Y. Zeevi, and J. Behar, "Remote atrial fibrillation burden estimation using deep recurrent neural network," *IEEE Transactions on Biomedical Engineering*, Aug. 2020, ISSN: 15582531. DOI: 10.1109/TBME.2020.3042646. arXiv: 2008.02228. [Online]. Available: http://arxiv.org/abs/2008.02228.

[7] P. Kligfield, L. S. Gettes, J. J. Bailey, R. Childers, B. J. Deal, E. W. Hancock, G. van Herpen, J. A. Kors, P. Macfarlane, D. M. Mirvis, O. Pahlm, P. Rautaharju, and G. S. Wagner, "Recommendations for the standardization and interpretation of the electrocardiogram," *Circulation*, vol. 115, no. 10, pp. 1306–1324, 2007. DOI: 10.1161/CIRCULATIONAHA.106.180200.

[8] N. Bobra. (). "Wavelet based ecg compression," [Online]. Available: https://github.com/nerajbobra/wavelet-based-ecg-compression.

[9] L. Biewald, *Experiment tracking with weights and biases*, Software available from wandb.com, 2020. [Online]. Available: https://www.wandb.com/.