Movement Artifacts Reduction from PPG Signals Using Learned Convolutional Sparse Coding

Giulio Basso[1,3], Xi Long[1,3], Reinder Haakma[2,3], Rik Vullings[1,3]

[1] Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands
[2] Philips, Eindhoven, The Netherlands
[3] Eindhoven MedTech Innovation Center (e/MTIC), Eindhoven, The Netherlands

Abstract

Deep learning models can effectively reduce the impact of movement artifacts in photoplethysmography (PPG) signals from wearable devices. However, they require large datasets, since they are often trained in a purely data-driven way. We previously proposed an unfolded neural network that integrates prior knowledge of the PPG signal's sparse structure by encoding it into a sparse representation with a learned dictionary of kernels. In this study, we extend this framework by analysing how hyperparameters affect the extracted representations and the model's performance. We train the model using a synthetic dataset based on the PulseDB dataset and test with daily life signals from PPG-DaLiA. Our analysis suggests that the sparsity parameter and the kernel size influence the recruitment of kernels and the sparsity of their temporal activations. Kernel size and dictionary size also affect the denoising performance. On synthetic data, the method improved the signal-to-noise ratio from –7.06 dB to 13.63 dB and reduced the heart rate mean absolute error (MAE) by 74%. On the PPG-DaLiA, it lowered the MAE by 34%. The proposed method outperformed the reference methods. Our findings show that the proposed method effectively improves the quality of PPG signals from wearable devices.

1. Introduction

Photoplethysmography (PPG) has emerged as a valid alternative to electrocardiography (ECG) for cardiovascular monitoring during daily life. While ECG requires electrodes, which are not convenient for continuous monitoring, PPG uses optical sensors that can be embedded in wrist or finger-worn devices. Nevertheless, obtaining reliable cardiovascular parameters is challenging due to the presence of movement artifacts (MAs).

Recently, deep learning methods, such as denoising autoencoders and generative adversarial networks (GANs) [1] [2], have been shown to be effective for reducing the impact of MAs. These methods extract features related to the pulsatile PPG component and exploit them to eliminate the MAs. However, they require large datasets because they are trained in a purely data-driven way. Algorithm unfolding [3] offers an alternative by embedding prior knowledge into the network architecture rather than learning it from intensive data. Hence, in our recent work [4], we proposed a denoising method based on algorithm unfolding. A convolutional sparse coding framework (CSC) encodes the PPG into a sparse representation, leveraging the sparsity properties of the signal. Dictionary learning allows the extraction of recurrent patterns in the signal, exploiting the similarities between consecutive cardiac pulses.

When designing such a model, the hyperparameters not only influence the performance, but also the features captured in the sparse representations and dictionary. For instance, [5] showed that the parameter λ controlling the sparsity of the representation also affects the learned dictionary, since for smaller λ the kernels resemble short-time patterns, while for larger λ they look like fuller templates of the input signal. Therefore, in this study, we assess the independent effect of the hyperparameters on the sparsity of the representation and denoising performance. Finally, the performance of the proposed method is compared with that of the reference methods.

2. Methods

2.1. Dataset and preprocessing

We first created a synthetic dataset using the PulseDB dataset [6], which comprises fingertip PPG from intensive care units (ICUs) sampled at 125 Hz. For each subject, 400 10-s segments were selected, and subjects were randomly split into training (70%, 2024 subjects), validation (15%, 434 subjects) and test (15%, 436 subjects) sets. Assuming minimal MAs in ICU recordings, PulseDB signals were

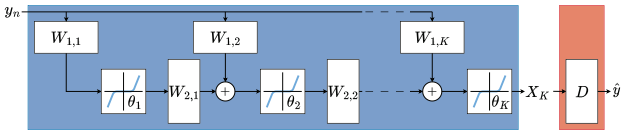

Figure 1: Architecture of the proposed model with y_n the noisy PPG, X_K the sparse code and $\hat{y}$ the reconstructed PPG. The encoder (blue) has K folds, each with convolutional layers $W_{1,k}$ and $W_{2,k}$, and learned thresholds θ_k. The decoder (red) stores the learned dictionary D.

used as clean references, and synthetic MAs were added using the model from [7] with one MA per PPG segment with random onset and duration. To test the method with real-life MAs, the PPG-DaLiA dataset [8] was used, which provides wrist PPG and ground-truth heart rate (HR) from the ECG of 15 subjects during daily activities.

The following preprocessing steps were applied. The recordings from PPG-DaLia, provided with a sampling rate of 64 Hz, were upsampled to 125 Hz to match the sampling rate of the synthetic dataset. The signals from both datasets were filtered with a 4th-order Chebyshev type II band-pass filter between [0.5, 18] Hz and normalized in amplitude between 0 and 1.

2.2. Learned convolutional sparse coding

A typical PPG signal consists of a series of events, such as the systolic peak, the dicrotic notch and the diastolic peak. It can thus be viewed as a superimposition of components, each of which is zero everywhere except when the event occurs. This suggests that the signal has an underlying sparse structure.

In our previous work [4], we proposed to denoise PPG signals using a learned convolutional sparse coding method. This approach incorporates knowledge about the signal structure by modelling the PPG signal according to a CSC model. The core idea of the CSC model is that the signal is represented as a sum of convolutions between a dictionary of kernels D and their temporal activations encoded in the sparse code X. Promoting sparsity over X ensures that only a few kernels from D are selected, and that their temporal activations are localized in time. The CSC model can be formulated as follows:

$$\hat{y} = D \star X \,, \tag{1a}$$

$$\text{with} \quad D \star X = \sum_{i=1}^{M} d_i * x_i \,, \quad \|X\|_0 \ll N \cdot M \,, \tag{1b}$$

where $\hat{y} \in \mathbb{R}^{\mathbb{N}}$ is the reconstructed PPG signal, the dictionary $D \in \mathbb{R}^{\mathbb{L}\times\mathbb{M}}$ constitutes M kernels with kernel size L, and the sparse code $X \in \mathbb{R}^{\mathbb{N}\times\mathbb{M}}$ constitutes the kernels' temporal activations $x_i \in \mathbb{R}^{\mathbb{N}}$.

Table 1: Configurations of hyperparameters used in the experiments with L kernel size, λ sparsity parameter and M size of the dictionary.

$L \in \{0.2, 0.4, 0.8, 1.6\}$ s	$\lambda = 0.05$	$M = 32$
$\lambda \in \{0.005, 0.01, 0.05, 0.1\}$	$M = 32$	$L = 0.4$ s
$M \in \{4, 8, 16, 32\}$	$L = 0.4$ s	$\lambda = 0.05$

The architecture of the learned convolutional sparse coding method is reported in Figure 1. The encoder was implemented using the deep unfolded iterative shrinkage algorithm by Van Sloun et al. [9], which adapts the LISTA architecture [3] to the convolutional case:

$$X_{k+1} = \tau_{\theta_k} \left(W_{1,k} * y_n + W_{2,k} * X_k \right) \,, \tag{2}$$

with y_n the noisy PPG signal, $W_{1,k} \in \mathbb{R}^{\mathbb{L}\times 1\times\mathbb{M}}$, $W_{2,k} \in \mathbb{R}^{\mathbb{L}\times\mathbb{M}\times\mathbb{M}}$ convolutional layers and τ_{θ_k} the smooth soft-thresholding function [10] with learned thresholds $\theta_k \in \mathbb{R}^{\mathbb{M}}$. The sparse code X_K obtained after K folds was then fed to a single-convolutional layer decoder $D \in \mathbb{R}^{\mathbb{L}\times\mathbb{M}\times 1}$ that implements equation 1a, as proposed in [11].

2.3. Experiments and evaluation metrics

The model was trained using the mean squared error loss, with kernels constrained to have unit norms to avoid the vanishing of X_K. Sparsity was promoted by applying l_1-norm regularization on X_K with sparsity parameter λ. l_2-norm regularization with parameter 10^{-3} was applied to $W_{1,k}$ and $W_{2,k}$ to avoid overfitting. Adam optimizer was used with learning rate 10^{-4}, batch size 256, and early stopping based on the validation loss. To prevent exploding gradients caused by the large network's depth, the norms of the gradients were clipped to 10.

We conducted a series of experiments to evaluate the effect of the kernel size L, the sparsity parameter λ, and the number of kernels M. Multiple models were trained, where each parameter was varied individually while keeping the others fixed, as reported in Table 1.

Three sparsity-related metrics were computed to assess the distribution of the nonzero elements across the representation. The first metric n_{tot} was the number of nonzero elements in the sparse code, i.e. its l_0 norm, which measures the overall sparsity of the representation. The second metric n_{ker} measured the sparsity in terms of the number of selected kernels by counting the number of x_i that have at least one nonzero element. This metric was used to evaluate how the hyperparameters affect the recruitment of the kernels. The last sparsity-related metric n_{act} was the l_0 norm of the x_i of the selected kernels, expressed as a percentage relative to N. In that way, we focused only on the sparsity of the temporal activations, quantifying the support over which the kernels were active.

Two additional metrics were computed to evaluate the denoising performance. First, the quality of the reconstruction was assessed using the signal-to-noise ratio:

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\sum_{\mathrm{i}=1}^{\mathrm{N}} y^2}{\sum_{\mathrm{i}=1}^{\mathrm{N}} (\hat{y}-y)^2}\right), \tag{3}$$

where y is the clean PPG and $\hat{y}$ is the reconstructed PPG. We further assessed the performance by measuring the mean absolute error (MAE) between the HR measured from the i-th reconstructed PPG segment $\mathrm{HR}_{\widehat{y_i}}$ and the one measured from the clean PPG segment HR_{y_i}:

$$\mathrm{MAE}_{\mathrm{HR}} = \frac{1}{\mathrm{W}}\sum_{\mathrm{i}=1}^{\mathrm{W}} |\mathrm{HR}_{\widehat{y_i}} - \mathrm{HR}_{y_i}|, \tag{4}$$

where W is the number of segments per subject. The peak detection, which was used to calculate the HR, was done using the Neurokit2 toolbox [12].

3. Results and discussion

3.1. Effect of hyperparameters

The effect of the hyperparameters was examined using the synthetic validation set. Figures 2a to 2c report the sparsity-related metrics. Focusing on the kernel size L, n_{tot} decreased with increasing L, meaning that the overall sparsity of the sparse code increased. The increase in sparsity was not due to a smaller number of selected kernels, but rather to sparser temporal activations, as n_{ker} remained almost stable, while n_{act} generally decreased with L. This might be because large kernels require being activated less frequently, since each kernel spans a large support of the signal. Considering the sparsity parameter λ, as expected, a larger λ increased the overall sparsity of the representation, indicated by the decreasing n_{tot}. Contrary to the effect of L, this time the increase in sparsity was caused by the fact that the algorithm recruited fewer kernels, as shown by the decreasing n_{ker}. Instead, the increasing n_{act} indicates that the temporal activations became less sparse, because fewer kernels were used, but they were activated more frequently to span the temporal support of the signal. The size M of the dictionary did not seem to have a significant effect on the sparsity of the representation. n_{tot} remained indeed stable with M, while n_{ker} and n_{act} did not show any clear trend with M.

Figures 2d and 2e report the denoising metrics. The denoising performance improved with increasing L and M, as shown by the increasing SNR and decreasing $\mathrm{MAE}_{\mathrm{HR}}$. This is because larger L and M translate to a higher number of trainable parameters, which increases the flexibility of the model to capture complex dependencies in the data, but also the risk of overfitting. In our experiments, varying

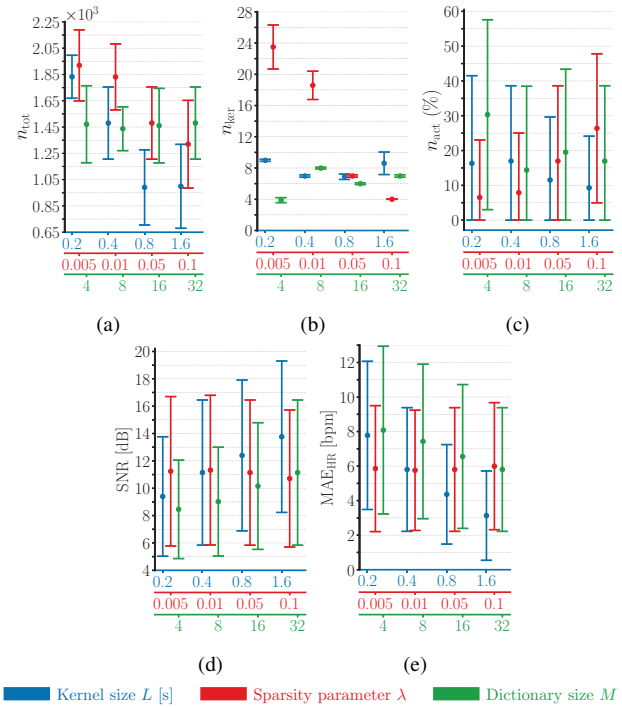

Figure 2: Effect of hyperparameters on (a) l_0 norm of the sparse code n_{tot}, (b) number of selected kernels n_{ker}, (c) l_0 norm of the x_i of the selected kernels n_{act}, (d) SNR, and (e) $\mathrm{MAE}_{\mathrm{HR}}$. The means $\pm$ standard deviation of the metrics are reported. Each point denotes a trained model, with colours indicating the varied hyperparameters.

λ within the range 0.005 to 0.1 did not lead to noticeable differences in the performance. One possible explanation is that, with λ within this range, the reconstruction term remains the dominant component of the loss, hence the network is still able to produce good quality reconstructions.

3.2. Comparison reference methods

The model's performance was compared to the convolutional denoising autoencoder (CDA) [1] and the fully connected GAN (FC-GAN) [2] (see [4] for implementation details). The methods were evaluated on the synthetic test set using both the SNR and $\mathrm{MAE}_{\mathrm{HR}}$, while only the $\mathrm{MAE}_{\mathrm{HR}}$ was assessed on the PPG-DaLia dataset. We used the proposed model with hyperparameters $L = 1.6$, $\lambda = 0.05$, and $M = 32$, as larger kernel sizes improve the performance (see Sec. 3.1). The matched-pairs Wilcoxon signed-rank test was used to evaluate statistically significant differences between each reference method (including without denoising) and the proposed method. We tested the alternative hypothesis that each reference method had significantly smaller SNR or larger $\mathrm{MAE}_{\mathrm{HR}}$ than the proposed method.

Table 2: Mean ± standard deviation of the SNR [dB] and the $\mathrm{MAE_{HR}}$ [bpm] without denoising (W. d.) and after applying each method on the synthetic and PPG-DaLiA datasets. Stars indicate that the reference method yields a statistically significantly smaller SNR or larger $\mathrm{MAE_{HR}}$ than the proposed method (* $p < 0.01$ and ** $p < 0.001$).

	Synthetic dataset		PPG-DaLiA
	SNR	$\mathrm{MAE_{HR}}$	$\mathrm{MAE_{HR}}$
W. d.	$-7.06 \pm 8.44^{**}$	$12.48 \pm 5.44^{**}$	$11.29 \pm 4.39^{**}$
CDA	$7.39 \pm 4.38^{**}$	$9.67 \pm 6.45^{**}$	$12.03 \pm 4.29^{**}$
FC-GAN	$10.00 \pm 4.39^{**}$	$5.72 \pm 5.12^{**}$	$9.48 \pm 5.09^{*}$
Proposed	$\mathbf{13.63 \pm 5.54}$	$\mathbf{3.21 \pm 2.79}$	$\mathbf{7.40 \pm 3.57}$

The denoising metrics are reported in Table 2. The proposed method improved the signal quality and the accuracy of the HR, as shown by the significantly higher SNR than the signals without denoising in the synthetic dataset, and the smaller $\mathrm{MAE_{HR}}$ in both datasets. It also obtained a significantly larger SNR and smaller $\mathrm{MAE_{HR}}$ than the CDA and FC-GAN methods across both datasets, achieving the best denoising performance.

Considering the limitations of the study, the synthetic MA model may not fully capture the complexity of real MAs, as denoising was more challenging on the PPG-DaLiA dataset, where all methods yielded lower SNR and higher $\mathrm{MAE_{HR}}$. Despite the limitations of the synthetic data used during training, the proposed method still achieved a considerable improvement in the $\mathrm{MAE_{HR}}$ of the PPG-DaLiA dataset.

4. Conclusion

In this work, we denoised PPG signals using an unfolded neural network that incorporates prior knowledge about the signals' sparse structure. We found that the kernel size affects the sparsity of the kernels' temporal activations, while the sparsity parameter influences the number of selected kernels. The denoising performance increases with both the kernel size and the dictionary size. The proposed method significantly improved the signal-to-noise ratio and mean absolute error of the heart rate measurement, and outperformed the reference CDA and FC-GAN methods. These results demonstrate that the method effectively reduces the impact of the MAs on PPG signals from wearable devices.

Acknowledgments

This work was labelled by ITEA and funded by local authorities under grant agreement "ITEA-2021-21022-RM4Health". It was performed within the framework of the Eindhoven MedTech Innovation Center (e/MTIC, incorporating Eindhoven University of Technology, Royal Philips, Catharina Hospital, Maxima Medical Center and Kempenhaeghe Epilepsy and Sleep Center).

References

[1] Mohagheghian F, Han D, Ghetia O, Peitzsch A, Nishita N, Nejad MPS, Ding EY, Noorishirazi K, Hamel A, Otabil EM, et al. Noise Reduction in Photoplethysmography Signals Using a Convolutional Denoising Autoencoder With Unconventional Training Scheme. IEEE Transactions on Biomedical Engineering 2023;71(2):456–466.

[2] Castro IAA, Oliveira HP, Correia R, Hayes-Gill B, Morgan SP, Korposh S, Gomez D, Pereira T. Generative Adversarial Networks With Fully Connected Layers to Denoise PPG Signals. Physiological Measurement 2025;13(2):025008.

[3] Gregor K, LeCun Y. Learning Fast Approximations of Sparse Coding. In Proceedings of the 27th International Conference on International Conference on Machine Learning. 2010; 399–406.

[4] Basso G, Long X, Haakma R, Vullings R. Reduction of Motion Artifacts From Photoplethysmography Signals Using Learned Convolutional Sparse Coding. arXiv preprint arXiv250810805 2025;.

[5] Evtimova K, LeCun Y. Sparse Coding With Multi-Layer Decoders Using Variance Regularization. arXiv preprint arXiv211209214 2021;.

[6] Wang W, Mohseni P, Kilgore KL, Najafizadeh L. PulseDB: A Large, Cleaned Dataset Based on MIMIC-III and VitalDB for Benchmarking Cuff-Less Blood Pressure Estimation Methods. Frontiers in Digital Health 2023;4:1090854.

[7] Paliakaitė B, Petrėnas A, Sološenko A, Marozas V. Modeling of Artifacts in the Wrist Photoplethysmogram: Application to the Detection of Life-Threatening Arrhythmias. Biomedical Signal Processing and Control 2021; 66:102421.

[8] Reiss A, Indlekofer I, Schmidt P, Van Laerhoven K. Deep PPG: Large-Scale Heart Rate Estimation With Convolutional Neural Networks. Sensors 2019;19(14):3079.

[9] Van Sloun RJ, Cohen R, Eldar YC. Deep Learning in Ultrasound Imaging. Proceedings of the IEEE 2019;108(1):11–29.

[10] Zhang XP. Thresholding Neural Network for Adaptive Noise Reduction. IEEE Transactions on Neural Networks 2001;12(3):567–584.

[11] Sreter H, Giryes R. Learned Convolutional Sparse Coding. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018; 2191–2195.

[12] Makowski D, Pham T, Lau ZJ, Brammer JC, Lespinasse F, Pham H, Schölzel C, Chen SA. NeuroKit2: A Python Toolbox for Neurophysiological Signal Processing. Behavior Research Methods 2021;1–8.

Address for correspondence:

Giulio Basso

Department of Electrical Engineering, Flux building, PO Box 513, 5600 MB Eindhoven, The Netherlands

g.b.basso@tue.nl