POLITECNICO DI TORINO
Repository ISTITUZIONALE

Low-power ECG acquisition by Compressed Sensing with Deep Neural Oracles

Original
Low-power ECG acquisition by Compressed Sensing with Deep Neural Oracles / Mangia, M.; Marchioni, A.; Prono, L.; Pareschi, F.; Rovatti, R.; Setti, G.. - STAMPA. - (2020), pp. 158-162. (Intervento presentato al convegno 2020 IEEE International Conference on Artificial Intelligence Circuits and Systems, AICAS 2020 tenutosi a virtuale nel AUGUST 31ST - SEPTEMBER 4TH 2020) [10.1109/AICAS48895.2020.9073945].

Availability:
This version is available at: 11583/2846086 since: 2020-09-18T15:18:34Z

Publisher:
Institute of Electrical and Electronics Engineers Inc.

Published
DOI:10.1109/AICAS48895.2020.9073945

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

12 October 2024

This is the author's version of the article that has been presented at IEEE AICAS2020
The editorial version of the paper is available at http://dx.doi.org/10.1109/10.1109/AICAS48895.2020.9073945

Low-power ECG acquisition by Compressed Sensing with Deep Neural Oracles

Mauro Mangia*, Alex Marchioni*, Luciano Prono‡, Fabio Pareschi‡†, Riccardo Rovatti*† and Gianluca Setti‡†
*DEI, †ARCES, University of Bologna, Italy - Email: {alex.marchioni, mauro.mangia2, riccardo.rovatti}@unibo.it
‡DET, Politecnico di Torino, Italy - Email: {luciano.prono,fabio.pareschi,gianluca.setti}@polito.it

***Abstract*—The recovery of sparse signals from their linear mapping on a lower-dimensional space is traditionally performed by finding the sparsest solution compatible with such solutions. This task can be partitioned in two phases: support estimation and coefficient estimation. We propose to perform the former with a deep neural network jointly trained with the encoder that divines a support that is used in the latter phase to estimate the coefficients by pseudo-inversion. Numerical evidence demonstrates that the proposed encoder-decoder architecture outperforms state-of-the-art Compressed Sensing (CS) approaches in the recovery of synthetic ECG signals for a compression ratio higher than 2.5. Further tests on real ECG prove the applicability in real-world scenarios.**

I. Introduction

Contexts in which sensors are designed with strict requirements in terms of computational resources and energy budget are continuously increasing. An example is the continuous monitoring of the heart activities that is of primary importance in the detection and prevention of cardiovascular diseases and requires the development of sensor networks that are either wearable or wireless [1].

In this setting, the design of the sensing device aims at reducing to the minimum the energy necessary to acquire and transmit the valuable information. This is precisely the aim of Compressed Sensing (CS) that leverages on the sparsity property, which characterizes many real-word signals to compress the waveform directly in the acquisition phase. Compression is achieved by means of linear projection of the signal on random sensing waveforms (called measurements), which are in numbers potentially much lower than the dimensionality of the signal acquired at Nyquist-rate. Obviously the number of measurements is strictly linked to the energy budget as the lower they are the larger the compression and the lower the computational cost.

The advantage provided by CS at the encoder side comes at the cost of additional complexity required for decoding. More precisely, the signal recovery corresponds to the solution of the problem of finding an n-dimensional sparse signal x from the set of m measurements y with $m < n$, i.e., determining the sparsest vector x among the infinite solutions that are linearly mapped to y by the encoder. This problem is known as *Basis Pursuit* (BP) and it is generalized to *BP with Denoising* (BPDN) in the presence of noise.

The literature provides different approaches to solve this combinatorial problem, starting from linear programming [2] that was the first approach able to obtain a solution in polynomial time, thus making the use of CS practical. This first approach has been further improved by exploiting more advanced solvers such as Spectral Projected Gradient for L1 Minimization (SPGL1) [3] and the Generalized Approximate Message Passing (GAMP) [4], or iterative methods such as the Orthogonal Matching Pursuit (OMP) [5] and the Compressive Sampling Matching Pursuit (CoSaMP) [6].

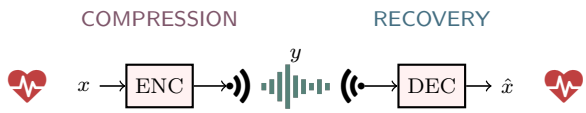

Fig. 1. General scheme of an encoder-decoder pair for ECG signals.

More recently, some works have succeeded in reducing computational complexity or in improving the quality of signal reconstruction by employing Deep Neural Networks (DNNs) at the decoder stage [7]–[14]. In [11] authors, inspired by iterative solvers, proposed a stacked denoising autoencoder (SDA) implemented using a 3-layer neural network to recover sparse images from their measurements. Similarly, in [13], authors have proposed a DNN, called ISTA-Net, inspired by the Iterative Shrinkage-Thresholding Algorithm (ISTA) [15], which optimizes the solution of BP to reconstruct compressed images. In [12], a fully-connected DNN is applied to measurements of videos for fast recovery and improved reconstruction quality. A deep learning framework applied to EEG signals is presented in [14], where three neural networks have been jointly optimized to perform binary measurement matrix multiplication, non-uniform quantization and signal recovery, respectively.

In this work, we propose an innovative approach that uses a DNN to improve the reconstruction of CS-sampled ECG signals, thus allowing the encoder to decrease the number of measurements needed to reconstruct the signal compared to more classical CS. The neural network, that is jointly trained with the encoder, is not used to recover the input signal directly but only to estimate its support, i.e., the positions of the non-null elements of the sparse representation of the input signal. Once the support has been *divined*, the coefficients can be obtained by pseudo-inverting the linear mapping in the encoder. This approach is proved to improve reconstruction quality compared to standard techniques in case of synthetically-generated ECGs, and it is shown to be effective also on real

Copyright (C) 2020 IEEE. Personal use is permitted.
For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org

This is the author's version of the article that has been presented at IEEE AICAS2020
The editorial version of the paper is available at http://dx.doi.org/10.1109/10.1109/AICAS48895.2020.9073945

ECGs.

The paper is organized as follows: Section II briefly reviews the CS theory, introduces the concept of reconstruction based on the DNN oracle, and shows a possible implementation of the overall architecture. System performances are shown in Section III, and finally, conclusions are drawn.

II. Compressed Sensing and Deep Neural Oracle

Let us assume that a sensor readings stream is chopped into subsequent n-samples length time windows, each of which is represented by a vector $x = (x_0, \ldots, x_{n-1})$. Let us also assume that each possible vector x is *κ-sparse* with respect to the basis defined by the columns of an orthonormal matrix S, that is, when we express $x = S\xi$, then only κ elements in the vector $\xi = (\xi_0, \ldots, \xi_{n-1})$ are non-null, with $\kappa \ll n$. With these assumptions and considering the scheme in Figure 1, the CS paradigm declines a proper encoder and decoder pair as follows.

Signal compression is performed by projections of x over the rows of a predefined $m \times n$ sensing matrix A with a resulting measurement vector $y = Ax$, where $m < n$ guarantees that signal is compressed by a factor n/m called *compression ratio* (CR).

The fact that the information content in x depends only on κ scalars hints at the possibility of recovery ξ, and therefore x, from y. This is the idea behind many methods already proposed in literature for the decoder stage [4]–[6]. As mentioned in the introduction, BP [2] is the most classical one. Here, $y = AS\xi$ (an ill-posed problem with infinite vectors ξ mapped into the same y) is done by selecting the sparsest vector $\hat{\xi}$.

$$\hat{\xi} = \arg\min_{\xi \in \mathbb{R}^n} \|\xi\|_1 \quad \text{s.t.} \quad y = AS\xi \tag{1}$$

where the ℓ_1-norm $\|\cdot\|_1$ promotes sparsity, A is such that CS theory holds and $\hat{x} = S\hat{\xi}$ is the resulting reconstructed signal.

As a class of sensing matrices A to be paired with the CS decoder, the standard CS theory initially proposed matrices whose entries are instances of independent zero-mean and unit-variance Gaussian random variables [16]–[18]. Then A matrices composed by instances of independent antipodal random variables (i.e., ± 1) were also adopted without performance degradation [19]. Here we will focus on this class of matrices since they reduce the acquisition complexity by lowering the resources needed to compute Ax. As a further step in resource optimization, methods able to reduce the number of rows in A by exploiting adaptation to the acquired class of signal have been discussed in literature [20]–[23]. Notably, the rakeness-based approach is also compliant with the class of antipodal sensing matrices [23] so that we will use performance associated with this method as a reference.

A. CS decoder with Deep Neural Oracle

Referring to a κ-sparse signal ξ, let us define its support $\operatorname{supp}\xi$ as a binary vector $s \in \{0,1\}^n$ such that $s_j = 1$ if $\xi_j \neq 0$ and $s_j = 0$ otherwise. Such kind of vectors can be also used to address either elements of a vector or columns of a matrix, i.e., the notation $\cdot_{|s}$ defines the subvector/submatrix collecting only the κ elements/columns corresponding to $s_j = 1$. Furthermore, we can say that $x = S\xi$ could be represented by a n-dimensional binary vector $s = \operatorname{supp}\xi$ and by the κ-dimensional vector $\xi_{|s}$.

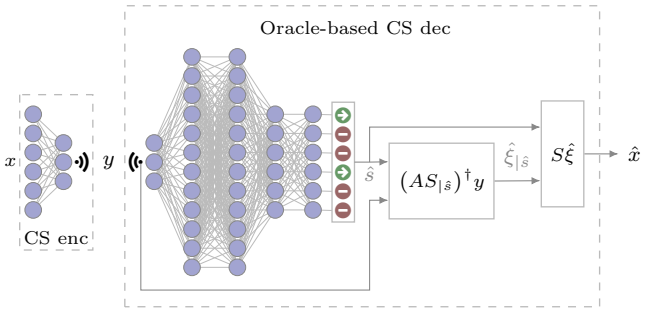

Fig. 2. Trained CS with support oracle block scheme. Here, the first part of the fully connected DNN performs CS encoder while the final estimated support $\hat{s}$ contains ones where the DNN outputs exceed a trained threshold. The estimated support is employed during the signal reconstruction, where the reconstructed signal $\hat{x}$ is estimated.

With this notation, it is now possible to introduce our alternative to the recovery stage in (1), which splits signal reconstruction into two stages, the estimation of s and the computation of $\xi_{|s}$.

For the latter, if an oracle gives us $\hat{s}$, that estimates s by looking at vector y, then one may note that $y = AS\xi$ can be inverted by solving $y = AS_{|\hat{s}}\hat{\xi}_{|\hat{s}}$, so that recovering ξ from

$$\hat{\xi}_{|\hat{s}} = \left(AS_{|\hat{s}}\right)^\dagger y \tag{2}$$

where $\cdot^\dagger$ stands for the Moore-Penrose pseudo-inversion that is needed since $AS_{|\hat{s}} \in \mathbb{R}^{m,\hat{\kappa}}$ is a matrix with more rows than columns and where $\hat{\kappa}$, that count the ones in $\hat{s}$, is in the same order of $\kappa < m$.

The adopted oracle is based on a Deep Neural Network (DNN) trained with a large set of couples of vectors x and their corresponding supports s. Since the encoder stage is similar to a neural network layer, the training phase also involves the design of the antipodal matrix A. This is done by inserting an additional layer devoted to computing y that has n inputs, m outputs, a linear activation function, zero bias, and weights constrained to be 1 or -1. As a result, encoder and decoder are jointly defined in the support identification task.

Finally, $\hat{s}$ estimated by the DNN and the corresponding $\hat{\xi}_{|\hat{s}}$ defines the recovered signal $\hat{x} = S\hat{\xi}$ as shown in Fig. 2. We name this approach Trained CS with Support Oracle, TCSSO.

B. DNN structure and training

As shown in Fig. 2, the DNN we propose takes an n-dimensional vector x as input while the final layer is composed by n neurons that correspond to the estimation of s and generate an output vector o. There are also a first hidden fully connected layer with m neurons and antipodal weights (implementing the sensing matrix A) and 3 additional fully connected layers of cardinality $2n$, $2n$, and n. The adopted

Copyright (C) 2020 IEEE. Personal use is permitted.
For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org

This is the author's version of the article that has been presented at IEEE AICAS2020
The editorial version of the paper is available at http://dx.doi.org/10.1109/10.1109/AICAS48895.2020.9073945

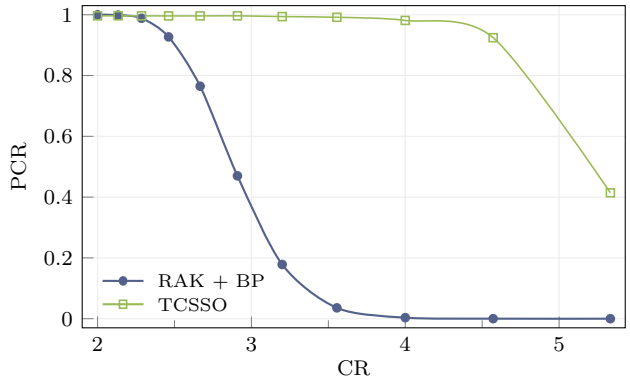

Fig. 3. Performance in terms of PCR as function of the compression ratio for both TCSSO and the BP decoder coupled with rakeness-based encoder adaptation (RAK + BP).

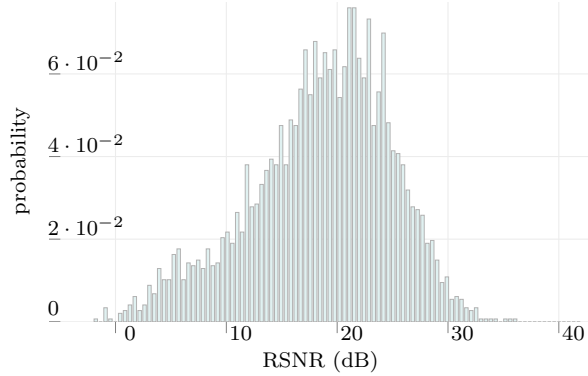

Fig. 4. RSNR histogram for TCSSO with ECG in [24], $m = 40$ (CR $= 3.2$).

activation function is linear for the first hidden layer while ReLu functions are used for the successive three layers. The output layer is also fully connected with a sigmoidal activation function[1]. Finally the support estimation is done by selecting the binary vector $\hat{s} \in \{0,1\}^n$ such that $\hat{s}_j = 1$ if $o_j \geqslant o_{\min}$ and $\hat{s}_j = 0$ otherwise.

During the training phase, weights and bias computation is based on the minimization of a loss function that give us the total component-wise clipped cross-entropy between s and o

$$X = -\sum_{j|s_j=1} \mathrm{logc}_\epsilon\left(o_j\right) - \sum_{j|s_j=0} \mathrm{logc}_\epsilon\left(1-o_j\right)$$

where ϵ is a small value and $\mathrm{logc}_\epsilon(\cdot)$ is a clipped log function defined as $\min\{\log_2(1-\epsilon), \max\{\log_2(\epsilon), \log_2(\cdot)\}\}$.

Note that, for the layer corresponding to the encoder stage, during the forward pass, we refer to antipodal weights, but this kind of constraint conflicts with the application of the backpropagation algorithm. To solve this issue, we use a matrix of unconstrained weights W^{enc} in the backward pass such that the gradient can be computed, while the forward pass employs $A = \mathrm{sign}(W^{\mathrm{enc}})$, where sign is applied component-wise at each training step. As a final remark, the training set used to compute all the DNN parameters is also employed to estimate the threshold $o_{\min}$.

III. Numerical evidence

The first considered dataset is composed by 8×10^5 signal instances of synthetic ECG randomly generated as in [25] with the same parameters of [22] where the sample rate is $256\,\mathrm{sample/s}$ and with $n = 128$ as in [26]. A set of 80% of these instances is for the DNN training, while the remaining 20% are for a first TCCSO performance assessment (this is the test set).

Training and inference of the proposed DNN were done employing the TensorFlow framework [27] with Keras API [28]. More precisely, a stochastic gradient descent algorithm has been considered for the DNN fitting, where each gradient step is computed on a 30 instances length mini-batch. The entire fitting process is iterated for 500 epochs, with an initial learning rate equal to 0.1.

As a figure of merit for performance assessment in the reconstruction of single instances we adopt the Reconstruction Signal-to-Noise Ratio (RSNR) expressed in dB

$$\mathrm{RSNR} = \left(\|x\|_2 / \|x - \hat{x}\|_2\right)_{\mathrm{dB}}$$

For the noiseless case, i.e., decoding of synthetic ECGs, RSNR can also be used to define an ensemble-level performance figure, the Probability of Correct Reconstruction (PCR)

$$\mathrm{PCR} = \Pr\left\{\mathrm{RSNR} \geqslant 100\,\mathrm{dB}\right\}$$

PCR values are computed on the entire test set and represent the probability that signal reconstruction does not fail. It was done for DNN with different m values and with the $o_{\min}$ value that maximizes the average RSNR observed on the training set.

Results are in Fig. 3, where the number of measurements ranges from $m = 24$ to $m = 64$ with CR that goes from 5.3 to 2. As anticipated in Sec. II, we use the decoder in (1) coupled with adapted sensing matrices that follow the rakeness-based CS approach [22], [23] as a reference so that a comparison with a state of the art approach is included.

TCSSO outperforms the already proposed CS framework technique. In particular, $m = 28$ (CR ≈ 4.6) corresponds to a PCR value higher than 0.9, while the traditional approach does not correctly reconstruct any input instance.

To move a step forward to real-applications, the DNN trained with synthetic ECG chunks has been tested on real ECG signals coming from the online repository [24] that is part of the Physionet project [29]. This database is composed of 168 ECG tracks (20.48 seconds each) sampled at 250 Hz such that each record corresponds to 40 windows, i.e., $n = 128$. Samples are obtained by 12 bit ADC. The choice of this database is motivated by the abundance of records representing possible abnormal ECG waveforms ranging from motion artifacts to arrhythmia.

The entire dataset has been randomly split into two parts such that the 20% of the instances are involved in the evaluation of $o_{\min}$ while the remaining part is for performance

[1] we refer to a function that maps any scalar a into $\left(1 + e^{-a}\right)^{-1}$.

Copyright (C) 2020 IEEE. Personal use is permitted.
For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org

This is the author's version of the article that has been presented at IEEE AICAS2020
The editorial version of the paper is available at http://dx.doi.org/10.1109/10.1109/AICAS48895.2020.9073945

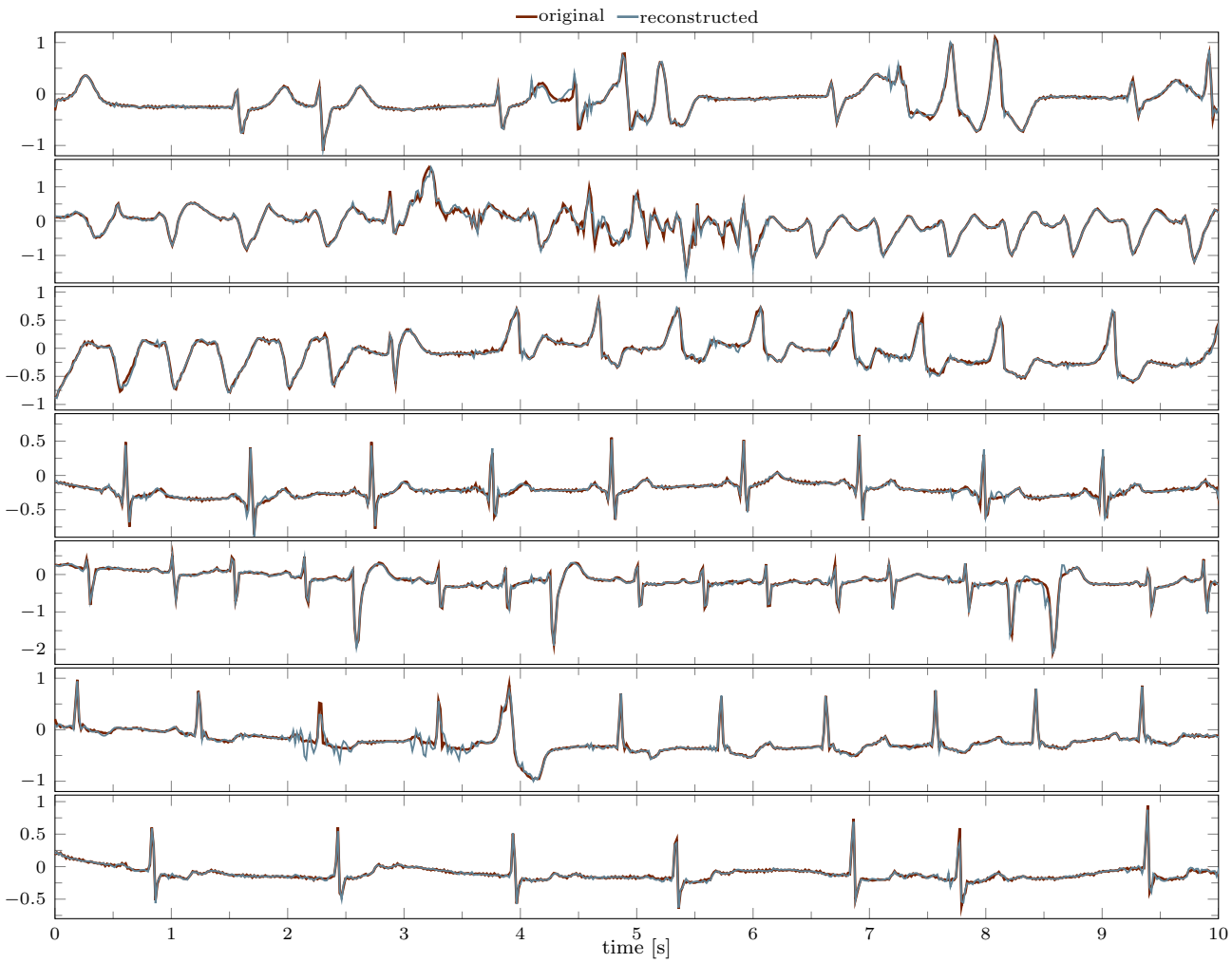

Fig. 5. Comparison between original ECG records and the corresponding reconstruction using TCCSO with $m = 40$. From the top to the bottom the waveforms refer to the first 20 s of the record 1, 4, 5, 13, 29, 37, 106 of the MIT-BIH Compression Test Database [24]. In each plot the y-axis represents the signal amplitude expressed in mV.

assessment. To limit the effect of noise affecting real signals, we also limit the number of ones in the vector s to 32, i.e., if the number of entries in $o \geqslant o_{\min}$ exceeds 32 then the ones in s correspond to the 32 highest values in o.

Fig. 4 shows the RSNR values distribution for the considered test set where $m = 40$ ($CR = 3.2$). This value is a conservative choice motivated by the processing of real waveforms. As a result, we observe a non-negligible variance in the quality of reconstruction, and instances with poor recovery performance exist. Moreover, the comparison with original ECG records and the corresponding reconstructed waveforms in Fig. 5 confirms that the proposed method correctly reconstructs ECG windows except for a few wrong recoveries.

IV. Conclusion

A CS decoder that first estimates the positions of non-zero coefficients and then computes their magnitudes has been proposed, where the support guessing is performed by a DNN-based oracle that also imposes the sensing matrix.

The proposed decoder largely outperforms state of the art approaches allowing for a further reduction in the amount of measurements to be computed and transmitted by the encoder stage. Furthermore, preliminary tests on real ECG records confirm the effectiveness of this method where few wrong instances reconstructions are also observed. To mitigate these unwanted signal reconstructions, the DNN we propose will be trained on an enlarged dataset that will include instances of real ECG instances.

Acknowledgments

This work was supported in part by the Italian Ministry for Education, University and Research (MIUR) under the program "Dipartimenti di Eccellenza (2018-2022).

Copyright (C) 2020 IEEE. Personal use is permitted.
For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org

This is the author's version of the article that has been presented at IEEE AICAS2020
The editorial version of the paper is available at http://dx.doi.org/10.1109/10.1109/AICAS48895.2020.9073945

References

[1] A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 1, pp. 1–12, Jan 2010.

[2] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[3] E. van den Berg and M. P. Friedlander, "SPGL1: A solver for large-scale sparse reconstruction," Jun. 2007, http://www.cs.ubc.ca/labs/scl/spgl1.

[4] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *2011 IEEE International Symposium on Information Theory Proceedings*, July 2011, pp. 2168–2172.

[5] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *Information Theory, IEEE Transactions on*, vol. 53, no. 12, pp. 4655–4666, 2007.

[6] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[7] A. Mirrashid and A. A. Beheshti, "Compressed remote sensing by using deep learning," in *2018 9th International Symposium on Telecommunications (IST)*, Dec 2018, pp. 549–552.

[8] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "Reconnet: Non-iterative reconstruction of images from compressively sensed measurements," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 449–458.

[9] A. Mousavi and R. G. Baraniuk, "Learning to invert: Signal recovery via deep convolutional networks," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 2272–2276.

[10] W. Shi, F. Jiang, S. Zhang, and D. Zhao, "Deep networks for compressed image sensing," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, July 2017, pp. 877–882.

[11] A. Mousavi, A. B. Patel, and R. G. Baraniuk, "A deep learning approach to structured signal recovery," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sep. 2015, pp. 1336–1343.

[12] M. Iliadis, L. Spinoulas, and A. K. Katsaggelos, "Deep fully-connected networks for video compressive sensing," *Digital Signal Processing*, vol. 72, pp. 9 – 18, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1051200417302130

[13] J. Zhang and B. Ghanem, "Ista-net: Interpretable optimization-inspired deep network for image compressive sensing," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 1828–1837.

[14] B. Sun, H. Feng, K. Chen, and X. Zhu, "A deep learning framework of quantized compressed sensing for wireless neural recording," *IEEE Access*, vol. 4, pp. 5169–5178, 2016.

[15] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[16] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[17] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. on Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb 2006.

[18] M. Mangia, F. Pareschi, V. Cambareri, R. Rovatti, and G. Setti, *Adapted Compressed Sensing for Effective Hardware Implementations: A Design Flow for Signal-Level Optimization of Compressed Sensing Stages*. Springer International Publishing, 2018.

[19] J. Haboba, M. Mangia, F. Pareschi, R. Rovatti, and G. Setti, "A pragmatic look at some compressive sensing architectures with saturation and quantization," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 3, pp. 443–459, Sep. 2012.

[20] M. Elad, "Optimized projections for compressed sensing," *Signal Processing, IEEE Transactions on*, vol. 55, no. 12, pp. 5695–5702, 2007.

[21] J. Xu, Y. Pi, and Z. Cao, "Optimized projection matrix for compressive sensing," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, no. 1, p. 560349, 2010.

[22] M. Mangia, R. Rovatti, and G. Setti, "Rakeness in the design of analog-to-information conversion of sparse and localized signals," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 5, pp. 1001–1014, May 2012.

[23] M. Mangia, F. Pareschi, V. Cambareri, R. Rovatti, and G. Setti, "Rakeness-based design of low-complexity compressed sensing," *IEEE Trans. on Circuits and Systems I: Reg. Papers*, vol. 64, no. 5, pp. 1201–1213, 2017.

[24] G. B. Moody and A. L. Goldberger, "Evaluation of the "trim" ecg data compressor," *Computers in Cardiology*, vol. 15, pp. 167–170, 1988.

[25] P. E. McSharry, G. D. Clifford, L. Tarassenko, and L. A. Smith, "A dynamical model for generating synthetic electrocardiogram signals," *IEEE Trans. on Biom. Eng.*, vol. 50, no. 3, pp. 289–294, Mar. 2003.

[26] F. Pareschi, P. Albertini, G. Frattini, M. Mangia, R. Rovatti, and G. Setti, "Hardware-Algorithms Co-Design and Implementation of an Analog-to-Information Converter for Biosignals Based on Compressed Sensing," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 1, pp. 149–162, Feb. 2016.

[27] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[28] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[29] A. L. Goldberger *et al.*, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. 215–220, Jun. 2000.

Copyright (C) 2020 IEEE. Personal use is permitted.
For any other purposes, permission must be obtained from the IEEE by emailing pubs-permissions@ieee.org