Low-power fixed-point compressed sensing decoder with support oracle

This is the author's version of the article that has been presented at IEEE ISCAS2020
The editorial version of the paper is available at http://dx.doi.org/10.1109/ISCAS45731.2020.9180502

Low-power Fixed-Point Compressed Sensing Decoder with Support Oracle

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

***Abstract*—Approaches for reconstructing signals encoded with Compressed Sensing (CS) techniques, and based on Deep Neural Networks (DNNs) are receiving increasing interest in the literature. In a recent work, a new DNN-based method named Trained CS with Support Oracle (TCSSO) is introduced, relying the signal reconstruction on the two separate tasks of support identification and measurements decoding. The aim of this paper is to improve the TCSSO framework by considering actual implementations using a finite-precision hardware. Solutions with low memory footprint and low computation requirements by employing fixed-point notation and by reducing the number of bits employed are considered. Results using synthetic electrocardiogram (ECG) signals as a case study show that this approach, even when used in a constrained-resources scenario, still outperform current state-of-art CS approaches.**

I. Introduction

Nowadays an increasing number of applications require tight constraints in terms of energy consumption and computational cost. An important example is given by medical systems designed to continuously monitor a patient condition without running out of energy [1]. Due to this, it is essential to find a way to acquire and transmit important information in a way capable of reducing as much as possible the required energy.

The use of the Compressed Sensing (CS) approach [2]–[4] allows for such a desired energy reduction. Under the assumption of a *sparse* input signal (that is commonly verified in many physical phenomena), CS can simultaneously acquire and compress an input signal with a simple linear projection on a set of sensing waveforms. In this way, values called measurements are obtained, and even if their number can be much smaller than the number of samples required according to the Nyquist-Shannon theorem, they are enough for a correct signal reconstruction. It is intuitive that being able to reduce the number of measurement may ensure a large energy saving at acquisition time.

The drawback of this approach is the effort necessary to reconstruct the signal given the compressed measurements. Reconstructing is indeed a quite complex task, that theoretically requires linear programming approaches [5]. To simplify reconstruction, many alternative methods have been proposed such as Spectral Projected Gradient for ℓ_1 Minimization (SPGL1) [6] and the Generalized Approximate Message Passing (GAMP) [7], or iterative methods such as the Orthogonal Matching Pursuit (OMP) [8] and the Compressive Sampling Matching Pursuit (CoSaMP) [9].

In the effort of reducing signal reconstruction complexity, also methods based on Deep Neural Networks (DNNs) have been proposed [10]–[17]. For example in [14] authors proposed a stacked denoising autoencoder (SDA) inspired by iterative solvers. It is implemented using a 3-layer neural network to retrieve encoded sparse images. In [16] authors have proposed a similar approach which uses a DNN inspired by the Iterative Shrinkage-Thresholding Algorithm (ISTA) [18] named ISTA-Net, which optimizes the solution of BP to decode compressed images. Again, in [15] a fully-connected DNN has been applied to measurements of videos for fast recovery and improved reconstruction quality. Finally a deep learning framework applied to EEG signals is presented in [17] where three different DNNs have been optimized together to perform binary measurement matrix multiplication, non-uniform quantization and signal recovery.

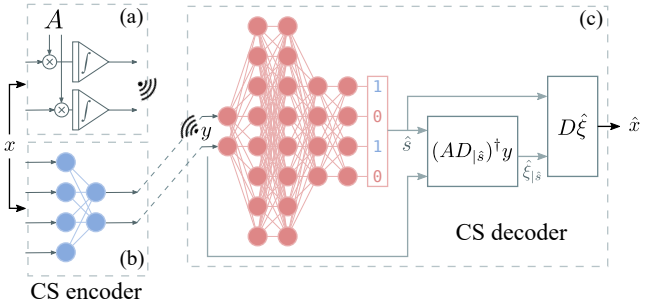

Fig. 1. General scheme of the TCSSO framework. The signal x is converted into a set of measurements y by an analog CS encoder (a), performing a signal manipulation that can be also modeled as a layer of a DNN (b). Finally the decoder (c) performs reconstruction by first support estimation $\hat{s}$ and then signal magnitude evaluation $\hat{x}$.

In this paper we assume that the CS encoder is built upon an analog circuit with a limited precision (as for [19]–[21]), and we focus on the decoding approaches based on DNN. Differences and novelty of our work with respect to other solutions can be summarized as follows. First, as schematized in Figure 1, reconstruction is achieved as a two-step process, where we initially estimate, by means of a DNN, the support of x (roughly speaking, defined as the set of the most important signal components), that is then used to recover the signal with standard linear algebraic operations. By modeling the linear projections of the acquisition process as an additional layer of the DNN performing the support estimation, we are also able to optimize the acquisition stage. This approach has been introduced in [22] and known as Trained CS with Support Oracle (TCSSO).

The main contribution of this work is to face the problem of

This is the author's version of the article that has been presented at IEEE ISCAS2020
The editorial version of the paper is available at http://dx.doi.org/10.1109/ISCAS45731.2020.9180502

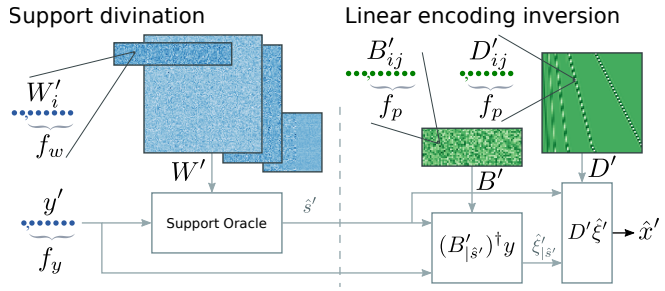

Fig. 2. Testing is performed by varying f_y, f_w and f_p in the TCSSO reconstruction framework which represent the number of fractional bits employed in the fixed-point notation of input compressed data, DNN parameters and sparsity matrix, respectively.

implementing the DNN in a resource constrained environment such as an edge computing node. Even if it has been shown that recovering via DNN may require a lower computational effort with respect to a traditional approach, the complexity of these solutions makes efficient implementation challenging. We aim to move a step into the direction of reducing the CS reconstruction complexity by investigating how reconstruction quality is affected when the DNN proposed in this paper is inferred on a device where only finite resolution arithmetic units are available. In particular, we consider the case where a limited resolution is used for representing input data, DNN parameters and the CS sparsity matrix (see Section II).

Results are proposed considering synthetically generated electrocardiogram (ECG) signals, and by comparing performance of the proposed approach to that achieved by state of the art CS approach.

The paper is organized as follows: Section II reviews CS theory, defines the DNN-based CS decoder and how the involved operations could be implemented in fixed-point resolution. The performance is shown in Section III and finally the conclusion is drawn.

II. Compressed Sensing and TCSSO

CS is a technique introduced to lower the energy required for the acquisition of a signal with respect to traditional approaches based on the Nyquist-Shannon theorem.

Mathematically, let us consider a stream of signal samples, and chop it into many subsequent windows, each of them represented as a n dimensional vector x. CS can be applied if every possible input instance x is sparse. Given an orthonormal sparsity matrix $D \in \mathbb{R}^{n\times n}$, and being $\xi \in \mathbb{R}^n$ the representation of x in D, i.e., the vector such that $x = D\xi$, x is sparse if ξ has at most $\kappa \ll n$ non-zero elements, i.e., only a few coefficients of ξ are significant. The significant elements of ξ are indicated with the *support* of the signal.

For sparse signals, it is possible to define a CS encoder stage that represents the information content of x with only $m < n$ values which are the projections on the rows of a predefined *sensing matrix* $A \in \mathbb{R}^{m\times n}$ so that $y = Ax$ and where $\mathrm{CR} = n/m$ expresses the achieved compression ratio.

Such a simple encoding approach is balanced by a quite complex decoding, i.e., the task of recovering ξ (or $x = D\xi$) from y. This is an ill-posed problem, since an infinite number of vectors ξ may be mapped into the same measurement vector y. In other words, since $m < n$, bringing back the signal from an m dimensional space to an n dimensional space has not a unique solution. To overcome this impasse, CS decoders exploit the fact that only κ values in ξ are non-zero. One of the most classical approach is *Basis Pursuit* (BP), that performs the inversion by selecting the sparsest vector ξ over all possible vectors for which the measurement equation $y = Ax = AD\xi$ holds. Mathematically

$$\hat{\xi} = \arg\min_{\xi\in\mathbb{R}^n} \|\xi\|_1 \quad \text{s.t.} \quad \|y - AD\xi\|_2 < \epsilon.$$

where the ℓ_1-norm $\|\cdot\|_1$ promotes sparsity and $\hat{x} = D\hat{\xi}$ is the reconstructed signal.

Standard CS theory at first proposed matrices whose entries are instances of independent and identically distributed (i.i.d.) Gaussian random variables [4], [23], [24] to be used as sensing matrix A to be paired with the CS decoder. Later, A matrices composed by instances of i.i.d. antipodal random variables i.e., with a Bernoulli distribution, were also employed without degradation in reconstruction performance [25]. In the following we always refer to the latter case.

A. Trained CS with Support Oracle

In this work we employ the TCSSO method recently introduced in [22]. In detail, TCSSO first uses a DNN to perform divination of the signal support $\hat{s}$ of x, i.e., to predict which are the non-zero elements of vector ξ from measurements y. Then, this information is used to evaluate the non-null values of ξ through Moore-Penrose pseudoinverse and to finally recover x.

The DNN is built as follows:

- the first layer applies the projection $y = Ax$ and is used only during training to optimize the sensing matrix; it has n inputs and m outputs, no bias, linear activation function. The sensing matrix is chosen to be antipodal because this would greatly ease the encoder physical implementation [26];
- from the second layer the oracle is built with m inputs and three hidden layers with $2n$, $2n$ and n neurons each and ReLU activation function;
- finally the output layer uses a sigmoid activation function $\alpha(a) = 1/(1+e^{-a})$ to evaluate vector $o \in \mathbb{R}^n$ which represents the probabilities of ξ coefficients to be non-zero.

From vector o we select a vector $\hat{s} \in \{0,1\}^n$ such that its generic element $\hat{s}_j = 1$ if $o_j > o_{\mathrm{th}}$ and $\hat{s}_j = 0$ otherwise. o_{th} is a threshold value obtained in the validation phase. Before that in the training phase we adopt a loss function which gives us the total component-wise clipped cross-entropy between original support $s \in \{0,1\}^n$ and o

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

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

This is the author's version of the article that has been presented at IEEE ISCAS2020
The editorial version of the paper is available at http://dx.doi.org/10.1109/ISCAS45731.2020.9180502

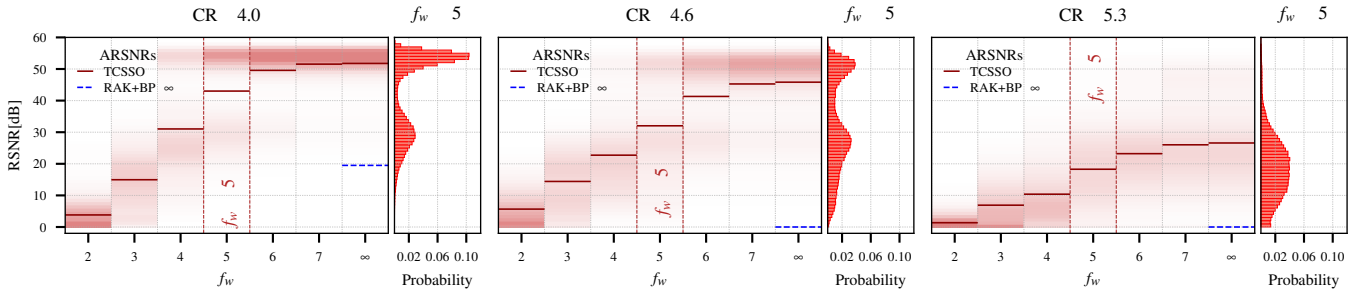

Fig. 3. Performance for TCSSO with synthetic ECG dataset performed with fixed-point oracle and infinite precision pseudoinverse with the variation of f_w. The colored area in the background shows the probability of obtaining a given RSNR value while the values for $f_w = 5$ are shown on the histograms. The ARSNRs of TCSSO and of rakeness-based BP method are highlighted.

With the estimated support, non-null coefficients of $\hat{\xi}$ are computed as follows

$$\hat{\xi}_{|\hat{s}} = \left(AD_{|\hat{s}}\right)^{\dagger} y = \left(B_{|\hat{s}}\right)^{\dagger} y \tag{1}$$

where $(\cdot)^{\dagger}$ is the Moore-Penrose pseudoinverse operation, $\xi_{|\hat{s}} \in \mathbb{R}^{\hat{\kappa}}$ is a vector that collects the entries of $\hat{\xi}$ corresponding to ones in $\hat{s}$, $D_{|\hat{s}} \in \mathbb{R}^{n \times \hat{\kappa}}$ is a matrix whose columns correspond to the columns of D selected by the ones in $\hat{s}$ and $\hat{\kappa}$ counts ones in $\hat{s}$ [1]. In (1), $B_{|\hat{s}}$ is also implicitly defined. We retrieve $\hat{\xi}$ from $\hat{\xi}_{|\hat{s}}$ by rearranging the coefficients as described by $\hat{s}$ and finally $\hat{x} = D\hat{\xi}$, where $\hat{x}$ is the reconstructed signal.

B. Fixed-point implementation

We want to study how this method behaves when we work on low-precision fixed-point values. Training of the DNN and a first evaluation of the performance are done with infinite precision (i.e., using a high-precision floating point representation).

As in our framework we are assuming that measurement evaluation is done in analog domain, allowing a theoretical infinite precision, as for [21], in practice the measurements have a finite resolution dependent on the implementation. Given a normalized measurement vector $y \in [-1, 1)$ with infinite resolution, we define the fixed-point finite-resolution equivalent as

$$y'_i = \lfloor y \cdot 2^{f_y} \rceil \cdot 2^{-f_y} \quad \text{for } i = 1, \ldots, m$$

where $\lfloor \cdot \rceil$ is the round to nearest operation, y'_i is the finite resolution version of the i-th entry of y and f_y is the number of fractional bits we want to employ so that the numeric precision we obtain is $\epsilon = 2^{-f_y}$. The representation of y' includes also 1 sign bit. We also define W as the set the DNN parameters used for support divination with infinite precision. We want to test how the Support Oracle would perform with a reduced resolution of the parameters by employing a fixed-point set

$$W'_i = \lfloor W_i \cdot 2^{f_w} \rceil \cdot 2^{-f_w} \quad \text{with } W_i \in W$$

where f_w is the number of fractional bits employed for the representation along with 1 sign bit and 1 integer bit.

[1] $\hat{\kappa}$ is such that $\kappa \leqslant \hat{\kappa} \leqslant \operatorname{rank}(B_{|\hat{s}})$ in case of correct support estimation

Matrix D and matrix B are used in the pseudoinverse phase of the reconstruction and can be expressed in fixed-point form as

$$B'_{ij} = \lfloor B_{ij} \cdot 2^{f_p} \rceil \cdot 2^{-f_p} \quad \text{for} \begin{cases} i = 1, \ldots, m \\ j = 1, \ldots, n \end{cases}$$
$$D'_{ij} = \lfloor D_{ij} \cdot 2^{f_p} \rceil \cdot 2^{-f_p} \quad \text{for } i, j = 1, \ldots, n$$

where f_p is the number of fractional bits used to encode the values of the two matrices. The values of the two matrices need also 1 sign bit while only B' requires 2 additional integer bits. Fixed-point matrix inversion and more specifically the Moore-Penrose pseudoinverse operation, which requires the singular value decomposition (SVD), are non-trivial tasks both in terms of computational complexity optimization and minimization of errors and their implementation is not obvious [27]–[30]; as working on this is not the specific target of this analysis, for now we have performed the second task with infinite precision while reducing only the precision of matrices B and D in order to save memory. The employment of efficient methods to perform the whole operation completely in fixed-point notation will be assessed in future works. A summary of fixed-point resolution reduction is shown in Figure 2.

III. Reconstruction Performance

We measure performance on a synthetic ECG dataset generated as in [31] using the same parameters described in [32]. The dataset is composed of 8×10^5 instances each of them of size $n = 128$. A portion of the dataset which is the 80% of the total number of windows is used for both training (generation of A and the set of DNN parameters W) and evaluation (define the threshold value o_{th}). The remaining 20% is employed as test set for performance assessment. As sparse matrix D we refer to the $n \times n$ matrix representing the Symlet-6 wavelet family transformation. Training of the DNN is performed with stochastic gradient descent through 500 epochs with a mini-batch size of 30 instances using Keras API with Tensorflow backend [33], [34] while inference is performed with a specially designed C++ framework which works with fixed-point notation.

As a figure of merit for assessing the performance of the TCSSO we use the Reconstruction Signal to Noise Ratio which is defined as follows

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

This is the author's version of the article that has been presented at IEEE ISCAS2020
The editorial version of the paper is available at http://dx.doi.org/10.1109/ISCAS45731.2020.9180502

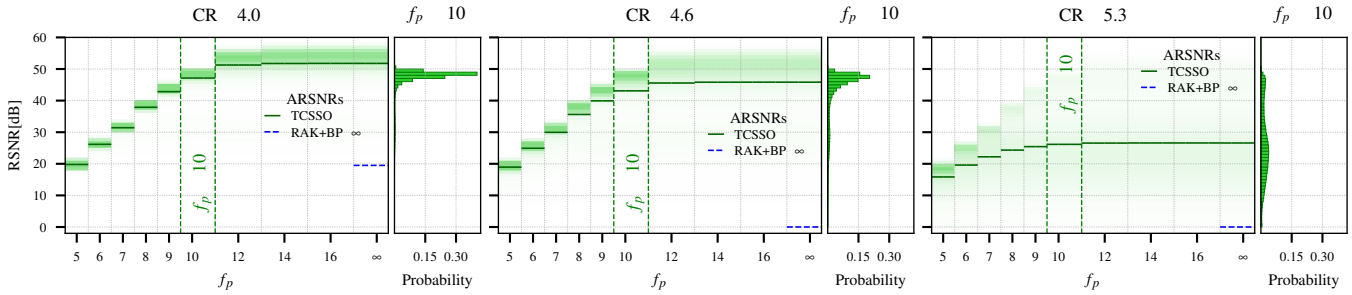

Fig. 4. Performance for TCSSO with synthetic ECG dataset performed with infinite precision oracle and fixed-point B and D matrices with the variation of f_p. The colored area in the background shows the probability of obtaining a given RSNR value while the values for $f_p = 10$ are shown on the histograms. The ARSNRs of TCSSO and of rakeness-based BP method are highlighted.

We aggregate RSNR values for the test set by computing the average value (ARSNR) and by evaluating the probability of obtaining a given RSNR value. Each signal in the test set has been corrupted by additive white Gaussian noise to emulate possible sources of nonidealities. Added noise is such that intrinsic SNR is 60 dB.

Our goal is to measure how the reduction of data precision for measurements y, for the DNN parameters W and for the matrices D and B impacts on the reconstruction of the compressed ECG signals. We take as reference the performance of a state-of-art method named rakeness-based CS [32], [35], which consists in the design of a sensing matrix adapted to the class of ECG signals, coupled with the standard BP decoder. Performance of this reference is evaluated with infinite precision.

A. Support Oracle with fixed-point precision

There are three degree of freedom we can work on when trying to reduce the complexity of the TCSSO: f_y, f_w and f_p. In this first phase Support Oracle performance alone is assessed by divining the support of ξ with low values of f_y and f_w while virtually keeping f_p to maximum, i.e. we perform pseudoinverse operations with infinite precision.

In order to reduce the degrees of freedom we choose to fix f_y to 7 bits that is the value after which, if $f_w = f_y$, performance starts to deteriorate. We reduce further only f_w to minimize also the memory footprint of the application. Results can be seen in Figure 3 for CR values that go from 4 to 5.3. Even with low precision DNN parameters this method outperforms the aforementioned state-of-art CS approach (RAK+BP), whose performance is null for CR > 4. Furthermore, with $f_w = 7$ performance is about the same we have with infinite resolution.

B. Pseudoinverse operation with fixed-point precision

In the second phase we evaluate how using low values of f_p for matrices D and B affects the pseudoinverse operations while support divination is performed with infinite resolution (f_y and f_p are virtually kept to the maximum).

In Figure 4 we can see how performance varies with values of f_p which go from 16 bits to 5 bits. Performance starts to decrease at $f_p = 10$. With a higher compression ratio the RSNR values are less sensitive to the variation to f_p.

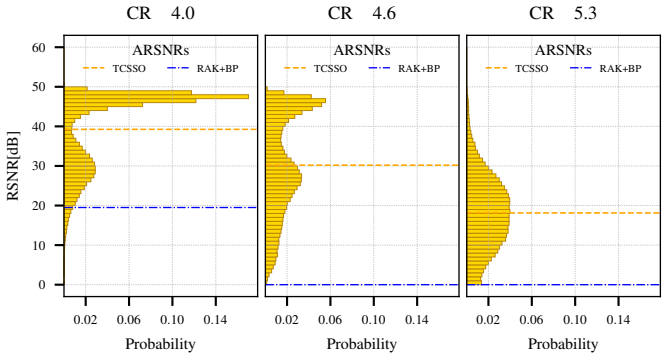

Fig. 5. RSNR histogram for TCSSO with synthetic ECG dataset performed with fixed-point oracle and fixed-point B and D matrices: $f_y = 7$, $f_w = 5$ and $f_p = 10$. ARSNRs of TCSSO, rakeness-based BP and SDA are highlighted.

C. TCSSO with fixed-Point precision

Finally, based on the discussed results, we set $f_y = 7$, $f_w = 5$ and $f_p = 10$. It should be noted that, by reducing the precision of y during the computation of $\hat{\xi}_{|\hat{s}}$, we are introducing a quantization error we didn't estimate before. Nevertheless we observe in Figure 5 that, with this setting, the overall performance is mainly influenced by the resolution reduction of the DNN parameters.

IV. Conclusion

The recently introduced CS paradigm named TCSSO is considered in this paper. TCSSO relies on a DNNs that is trained with the aim of joint optimizing both encoder and decoder stages, and relies on the separation of the compressed signal reconstruction into two tasks: support estimation and inversion of the encoded measurements. In this paper, the framework is improved by considering a fixed-point implementation with a reduced number of bits for data, network parameters and sparsity matrix. The scenario given by synthetic ECG signals is investigated as a case study. Achieved results show that the considered approach performs better with respect to other state-of-art approach even with a low resolution implementation.

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] R. G. Baraniuk, E. Candes, R. Nowak, and M. Vetterli (Eds.), "Special issue on compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, March 2008.

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

[4] 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.

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

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] D. Gangopadhyay, E. G. Allstot, A. M. R. Dixon, K. Natarajan, S. Gupta, and D. J. Allstot, "Compressed sensing analog front-end for bio-sensor applications," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 2, pp. 426–438, Feb 2014.

[20] M. Shoaran, M. H. Kamal, C. Pollo, P. Vandergheynst, and A. Schmid, "Compact low-power cortical recording architecture for compressive multichannel data acquisition," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 6, pp. 857–870, Dec 2014.

[21] 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.

[22] M. Mangia, L. Prono, A. Marchioni, F. Pareschi, R. Rovatti, and G. Setti, "Deep neural oracles for short-window optimized compressed sensing of biosignals," *preprint on IEEE TechRxiv*, 2019. [Online]. Available: https://www.techrxiv.org/articles/NN-CSdecoder2019_pdf/10049843

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

[24] 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.

[25] 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.

[26] F. Pareschi *et al.*, "Energy analysis of decoders for rakeness-based compressed sensing of ecg signals," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, no. 6, pp. 1278–1289, 2017.

[27] P. Salmela, A. Happonen, A. Burian, and J. Takala, "Several approaches to fixed-point implementation of matrix inversion," in *International Symposium on Signals, Circuits and Systems, 2005. ISSCS 2005.*, vol. 2, Jul. 2005, pp. 497–500 Vol. 2.

[28] R. Mohanty, A. Gonnabhaktula, T. Pradhan, B. Kabi, and A. Routray, "Design and performance analysis of fixed-point jacobi svd algorithm on reconfigurable system," *IERI Procedia*, vol. 7, p. 21–27, 12 2014.

[29] B. Kabi, "Analysis of fixed-point singular value decomposition algorithms," Master's thesis, Advanced Technology Development Centerindian Institute of Technology, 07 2015.

[30] T. Pradhan, B. Kabi, and A. Routray, "Fixed-Point Hestenes Algorithm for Singular Value Decomposition of Symmetric Matrices," in *Proceedings of 2013 International Conference on Electronics, Signal Processing and Communication Systems*, 2013.

[31] 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.

[32] 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.

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

[34] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[35] 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.