

# Quantized Compressed Sensing via Deep Neural Networks

Markus Leinonen

Centre for Wireless Communications – Radio Technologies  
FI-90014, University of Oulu, Finland  
e-mail: markus.leinonen@oulu.fi

Marian Codreanu

Department of Science and Technology  
Linköping University, Sweden  
e-mail: marian.codreanu@liu.se

**Abstract**—Compressed sensing (CS) is an efficient technique to acquire sparse signals in many wireless applications to, e.g., reduce the amount of data and save low-power sensors' batteries. This paper addresses efficient acquisition of sparse sources through quantized noisy compressive measurements where the encoder and decoder are realized by deep neural networks (DNNs). We devise a DNN based quantized compressed sensing (QCS) method aiming at minimizing the mean-square error of the signal reconstruction. Once trained offline, the proposed method enjoys extremely fast and low complexity decoding in the online communication phase. Simulation results demonstrate the superior rate-distortion performance of the proposed method compared to a polynomial-complexity QCS reconstruction scheme.

## I. INTRODUCTION

In a myriad of wireless applications, the signals of interest are *sparse*, i.e., they have many zero-valued elements, either naturally or after a transformation [1]. Sparse signals are encountered in, e.g., environmental monitoring [2], source localization [3], and spectrum sensing [4]. The sparsity can be leveraged by the joint sampling and compression paradigm, *compressed sensing* (CS) [5], [6], which enables a sparse length- $N$  signal to be accurately reconstructed from its  $M < N$  (random) linear measurements. CS benefits from simple and universal encoding by shifting the most computational work load to the decoder [7], making it especially attractive for low-power sensor applications.

Early works on CS considered real-valued signals by treating the CS merely as a dimensionality reduction technique. Recently, the interest in applying CS for transmission/storage in digital systems, where the compressive measurements must be converted into bits using a quantizer, initiated the *quantized compressed sensing* (QCS) framework [8]. To achieve satisfactory rate-distortion performance in a QCS setup, the conventional CS reconstruction methods must be appropriately redesigned. The main idea behind QCS algorithms is to accommodate the non-linear impact of CS and quantization in the encoder/decoder to ameliorate the signal reconstruction. First works on QCS considered scalar quantizers (SQs) and optimized either the encoder or quantization-aware decoder [9], [10]. At the cost of increased complexity, QCS methods relying on vector quantization have been devised in, e.g., [11], [12], [13], outperforming the SQ based methods. Empirical performance of various QCS methods is reported in [14].

Although the encoding in SQ based QCS methods is simple, the decoder executes a (quantization-aware) convex/greedy algorithm. Thus, the decoding can become computationally infeasible for high-dimensional signals or cause intolerable delays for real-time applications. On the other hand, VQ based QCS methods yield superior rate-distortion performance, but their encoding complexity may become a bottleneck.

The aforementioned problem could be counteracted by *deep learning*, i.e., realizing the CS decoder by a *deep neural network* (DNN), combined with a simple quantizer. Namely, if such a scheme meets the pre-defined performance criterion after intensively trained *offline*, the *online* communication phase would be extremely fast with low complexity decoding. The authors in [15] were the first to apply deep learning in a non-quantized CS framework. The first work that applies deep learning into a QCS framework is [16], where the authors devised a DNN based method that optimizes a binary measurement matrix, a compander based non-uniform quantizer, and the CS decoder to reconstruct neural spikes.

We address efficient acquisition of sparse sources through quantized noisy compressive measurements where the encoder and decoder are realized by feedforward DNNs. We propose a non-iterative DNN based QCS method which, once trained offline, enjoys extremely fast and low complexity decoding in the online communication phase, conducive to real-time applications with high-dimensional signals. Stochastic gradient descent (SGD) and backpropagation are used to train the DNNs with the aim of minimizing the mean-square error of the signal reconstruction. As a key design factor, we use the *soft-to-hard quantization* proposed in [17] to approximate the effect of the SQ in the training phase, which 1) facilitates backpropagating the gradient information, and 2) allows learning an efficient non-uniform SQ in the end-to-end SGD optimization loop. Simulation results illustrate the superior rate-distortion performance of the proposed method compared to a polynomial-complexity baseline QCS method.

The most related work is [16] which assumes that an encoder has direct access to the source; we consider a remote sensing setup where the encoder observes the source only *indirectly* through CS principles, and in the presence of measurement noise. The work [16] backpropagates the gradient through the quantizer by the straight-through estimator, whereas we use the soft-to-hard quantization. As a novelty, we

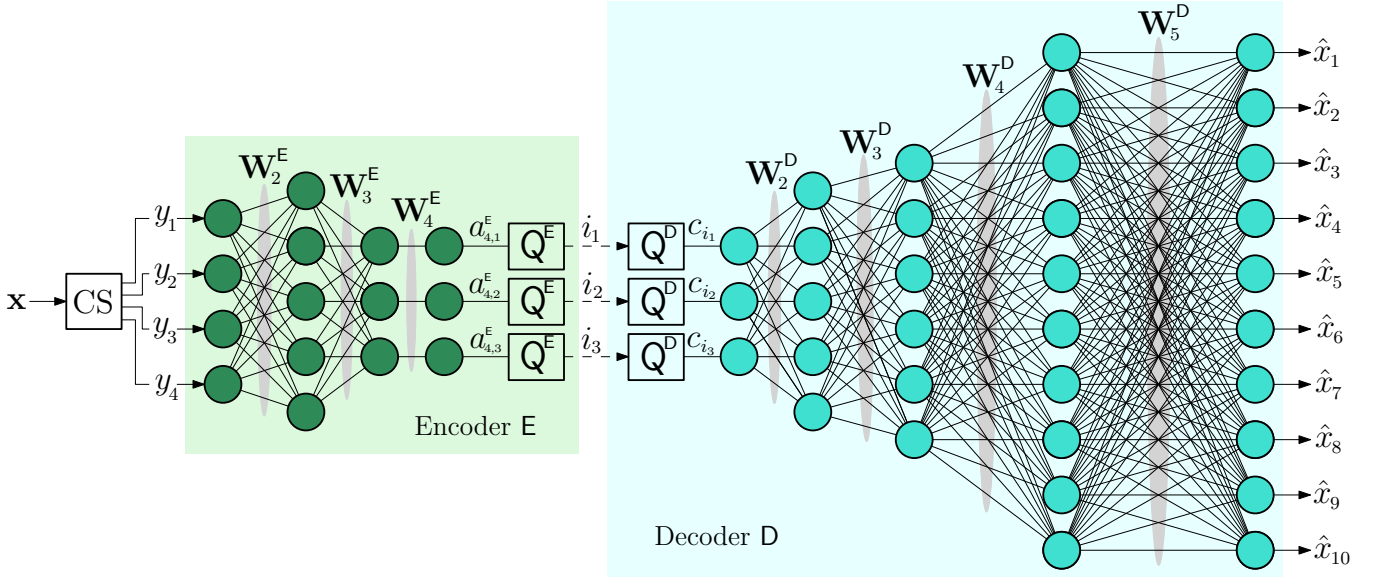


Fig. 1. Illustration of the proposed DNN based QCS method for the signal length  $N = 10$  and the number of measurements  $M = 4$ , where 1) EncNet has  $L = 4$  layers with  $\gamma_1 = M = 4$ ,  $\gamma_2 = 5$ ,  $\gamma_3 = P = 3$ , and  $\gamma_4 = P = 3$  neurons, and 2) DecNet has  $J = 5$  layers with  $\gamma_1 = P = 3$ ,  $\gamma_2 = 5$ ,  $\gamma_3 = 6$ ,  $\gamma_4 = 10$ , and  $\gamma_5 = N = 10$  neurons.

incorporate an encoder DNN to pre-process the compressive measurements prior to quantization, conducive to improve the rate-distortion performance. Since our encoder DNN performs a non-linear transform, it effectively replaces the compander used in [16] to perform non-uniform quantization.

## II. SYSTEM MODEL

Consider a remote signal acquisition setup depicted in Fig. 1. Encoder E observes a remote information source *indirectly* in the form of noisy compressive measurements (i.e., it performs a dimensionality reduction), pre-processes the measurements, and quantizes them into a finite-rate bit sequence. The bit sequence is communicated in an error-free fashion to decoder D for signal reconstruction.

### A. Source Signal and Compressive Measurements

Let  $\mathbf{x} \in \mathbb{R}^N$  denote the length- $N$  source vector, representing, e.g., a sequence of values of a random process at consecutive discrete time instants. The realizations of random vector  $\mathbf{x}$  are assumed to be independent and identically distributed across time. We assume that the vector  $\mathbf{x}$  is  $K$ -sparse, i.e., it has at most  $K$  non-zero entries,  $\|\mathbf{x}\|_0 = K \leq N$ . The *a priori* probabilities of the sparsity patterns are assumed to be unknown.

Encoder E observes the remote source  $\mathbf{x}$  indirectly in the form of noisy compressive measurements as [5], [6]

$$\mathbf{y} = \Phi \mathbf{x} + \mathbf{n} \quad (1)$$

where  $\mathbf{y} \in \mathbb{R}^M$  is a measurement vector,  $\Phi \in \mathbb{R}^{M \times N}$  is a fixed and known measurement matrix, and  $\mathbf{n} \in \mathbb{R}^M$  is a noise vector,  $M \leq N$ . It is worth emphasizing that encoder E does not have access to the information source  $\mathbf{x}$ ; the encoder device samples and acquires  $\mathbf{x}$  merely via the CS principles.

### B. Encoder

1) *Encoder DNN*: As the first stage at encoder E, the measurements in (1) are fed into a *feedforward neural network*, dubbed EncNet. EncNet consists of  $L$  layers; the first layer  $l = 1$  is called the *input* layer and the last layer  $l = L$  is called the *output* layer. For  $L \geq 3$ , EncNet contains at least one *hidden* layer; in this case, EncNet is referred to as a *deep neural network* (DNN). Since our main interest is in  $L \geq 3$ , we refer to EncNet as a DNN from here onward.

Formally, let  $\mathbf{z}_l^E \in \mathbb{R}^{\gamma_l}$  denote the *weighted input* of layer  $l$  of EncNet, defined as

$$\mathbf{z}_l^E = \mathbf{W}_l^E \mathbf{a}_{l-1}^E + \mathbf{b}_l^E, \quad l = 2, \dots, L, \quad (2)$$

where  $\mathbf{W}_l^E \in \mathbb{R}^{\gamma_l \times \gamma_{l-1}}$  is the weight matrix at the  $l$ th layer,  $\mathbf{b}_l^E \in \mathbb{R}^{\gamma_l}$  is the bias vector at the  $l$ th layer,  $\gamma_l$  denotes the number of neurons at layer  $l$ , and  $\mathbf{a}_l^E \in \mathbb{R}^{\gamma_l}$  is the *output* of layer  $l$ , defined as

$$\mathbf{a}_l^E = \sigma_l^E(\mathbf{z}_l^E) = \sigma_l^E(\mathbf{W}_l^E \mathbf{a}_{l-1}^E + \mathbf{b}_l^E), \quad l = 2, \dots, L, \quad (3)$$

where  $\sigma_l^E$  is an (element-wise) activation function of the  $l$ th layer. Let  $P \triangleq \gamma_L$  denote the number of neurons at the output layer of EncNet. Recalling that  $\gamma_1 = M$ , EncNet takes a measurement vector  $\mathbf{y} \in \mathbb{R}^M$  as an input, and produces an output vector  $\mathbf{a}_L^E \in \mathbb{R}^P$ . Thus, EncNet is defined as a mapping

$$\Gamma^E : \mathbb{R}^M \rightarrow \mathbb{R}^P, \quad \Gamma^E(\mathbf{y}) = \mathbf{a}_L^E. \quad (4)$$

2) *Quantizer Encoder*: The second stage at encoder E is to quantize the output of EncNet, a vector  $\mathbf{a}_L^E \in [a_{L,1}^E \dots a_{L,P}^E]^T$ , into a finite-rate bit sequence. We consider that the quantization of each  $a_{L,p}^E$ ,  $p = 1, \dots, P$ , is performed by a fixed-rate (non-uniform) *scalar quantizer* (SQ). More precisely, we assume that each continuous-amplitude element  $a_{L,p}^E$  is converted into a discrete representation using a single quantization

rule. Thus, the quantization operation of vector  $\mathbf{a}_L^E$  can be modeled by  $P$  parallel identical SQs [18]. A formal definition of an SQ is given next.

**Definition 1.** (SQ) Let  $\mathbf{Q} = \{\mathbf{Q}^E, \mathbf{Q}^D\}$  represent an  $I$ -level SQ, where  $\mathbf{Q}^E$  is the quantizer encoder and  $\mathbf{Q}^D$  is the quantizer decoder. Let  $\mathcal{R}_1, \dots, \mathcal{R}_I$  denote the quantization regions, where  $\mathcal{R}_i \cap \mathcal{R}_{i'} = \emptyset, \forall i \neq i' \in \mathcal{I}, \bigcup_{i \in \mathcal{I}} \mathcal{R}_i = \mathbb{R}$ , and  $\mathcal{I} = \{1, \dots, I\}$  is an index set. Let  $\mathcal{C} = \{c_1, \dots, c_I\}$  denote the quantizer code book, where  $c_i \in \mathbb{R}$  is the discrete code word associated with region  $\mathcal{R}_i, i \in \mathcal{I}$ . The quantizer encoder is a mapping from a real-valued scalar input to the set of indices  $\mathbf{Q}^E: \mathbb{R} \rightarrow \mathcal{I}$ ; thus, for an input  $a_{L,p}^E$  (i.e., the  $p$ th output of EncNet),  $\mathbf{Q}^E$  operates as

$$\mathbf{Q}^E(a_{L,p}^E) = i_p \in \mathcal{I}, \text{ if } a_{L,p}^E \in \mathcal{R}_{i_p}, p = 1, \dots, P. \quad (5)$$

The quantizer decoder is a mapping from the set of indices to the set of discrete code words  $\mathbf{Q}^D: \mathcal{I} \rightarrow \mathcal{C}$ ; thus, for a given index  $i_p \in \mathcal{I}$ ,  $\mathbf{Q}^D$  operates as

$$\mathbf{Q}^D(i_p) = c_{i_p} \in \mathcal{C}, p = 1, \dots, P. \quad (6)$$

We assume that the quantization indices  $i_p \in \mathcal{I}$  are communicated from encoder E to decoder D using binary code words of the same length, i.e., entropy coding is not applied. Thus, the rate of quantizer Q is  $R_Q = \lceil \log_2 I \rceil$  bits.

Combining the operations of EncNet and quantizer encoder  $\mathbf{Q}^E$ , encoder E can be expressed as a composite function

$$\mathbf{E} = \mathbf{Q}^E \circ \Gamma^E: \mathbb{R}^M \rightarrow \underbrace{\mathcal{I} \times \dots \times \mathcal{I}}_P. \quad (7)$$

The full operation of encoder E is summarized in (8).

*Remark 1.* Note that quantization of measurements  $\mathbf{y}$  in (1) is a practical necessity in any digital communication system. Moreover, for devices like low-power sensors having stringent resource limitations, the quantization resolution can be as low as one bit per sample [19]. In such scenarios, the quantization error becomes a pronounced factor in deteriorating the end-to-end signal reconstruction accuracy, and thus, the effect of quantization must be appropriately taken into account in the design. Albeit advanced techniques like vector quantization and entropy coding can provide significant compression gains [12], [13], [14], we use SQ due to its simple implementation and prevalent use in practical devices.

### C. Decoder

1) *Quantizer Decoder:* As the first stage of decoder D, each received index  $i_p, p = 1, \dots, P$ , is fed into quantizer decoder  $\mathbf{Q}^D$  which produces a discrete code word  $c_{i_p}$  according to (6).

2) *Decoder DNN:* As the second stage of decoder D, the quantizer decoders' outputs are fed into a  $J$ -layer feedforward neural network, dubbed DecNet. More precisely, a code word  $c_{i_p}$  is fed into the  $p$ th neuron of the input layer of DecNet,  $p = 1, \dots, P$ . Similarly as for EncNet, we consider  $J \geq 3$  and refer to DecNet as a DNN.

Formally, let  $\mathbf{z}_j^D \in \mathbb{R}^{\gamma_j}$  denote the weighted input of layer  $j$  of DecNet, defined as

$$\mathbf{z}_j^D = \mathbf{W}_j^D \mathbf{a}_{j-1}^D + \mathbf{b}_j^D, j = 2, \dots, J, \quad (9)$$

where  $\mathbf{W}_j^D \in \mathbb{R}^{\gamma_j \times \gamma_{j-1}}$  is the weight matrix at the  $j$ th layer,  $\mathbf{b}_j^D \in \mathbb{R}^{\gamma_j}$  is the bias vector at the  $j$ th layer, and  $\mathbf{a}_j^D \in \mathbb{R}^{\gamma_j}$  is the output of layer  $j$ , defined as

$$\mathbf{a}_j^D = \sigma_j^D(\mathbf{z}_j^D) = \sigma_j^D(\mathbf{W}_j^D \mathbf{a}_{j-1}^D + \mathbf{b}_j^D), j = 2, \dots, J, \quad (10)$$

where  $\sigma_j^D$  is an (element-wise) activation function of the  $j$ th layer. Since the output of DecNet represents the estimate of source signal  $\mathbf{x}$ , we have  $\gamma_J = N$ . Accordingly, DecNet takes  $P$  discrete code words  $c_{i_p}, p = 1, \dots, P$ , as an input, and produces an output vector  $\mathbf{a}_J^D \in \mathbb{R}^N$ . Defining  $\mathbf{c} = [c_{i_1} \dots c_{i_P}]^T$ , DecNet is given as a mapping

$$\Gamma^D: \mathbb{R}^P \rightarrow \mathbb{R}^N, \quad \Gamma^D(\mathbf{c}) = \mathbf{a}_J^D. \quad (11)$$

Combining the operations of quantizer decoder  $\mathbf{Q}^D$  and DecNet, decoder D can be expressed as a composite function

$$\mathbf{D} = \Gamma^D \circ \mathbf{Q}^D: \underbrace{\mathcal{I} \times \dots \times \mathcal{I}}_P \rightarrow \mathbb{R}^N. \quad (12)$$

The full operation of decoder D is summarized in (13).

## III. DEEP QUANTIZED COMPRESSED SENSING METHOD

### A. Problem Formulation

Our aim is to optimize encoder  $\mathbf{E} = \mathbf{Q}^E \circ \Gamma^E$  and decoder  $\mathbf{D} = \Gamma^D \circ \mathbf{Q}^D$  for a given total quantization bit-rate  $PR_Q$  bits to minimize a cost function  $C(\cdot)$  that is defined as the mean-square error (MSE) distortion between the source  $\mathbf{x}$  and the output of decoder D, i.e.,

$$C(\Gamma^E, \mathbf{Q}^E, \mathbf{Q}^D, \Gamma^D) = \mathbb{E} \left[ \|\mathbf{D}(\mathbf{E}(\mathbf{y})) - \mathbf{x}\|_2^2 \right], \quad (14)$$

where the expectation is with respect to the randomness of the source vector  $\mathbf{x}$  and noise vector  $\mathbf{n}$ .

### B. Optimization of Encoder and Decoder

First, jointly optimizing quantizer  $\mathbf{Q} = \{\mathbf{Q}^E, \mathbf{Q}^D\}$  and the DNN mappings  $\Gamma^E$  and  $\Gamma^D$  to minimize (14) is intractable. Thus, we remove quantizer Q in the (offline) optimization/training phase and minimize an alternative cost function defined as

$$\tilde{C}(\Gamma^E, \Gamma^D) = \mathbb{E} \left[ \|\Gamma^D(\Gamma^E(\mathbf{y})) - \mathbf{x}\|_2^2 \right]. \quad (15)$$

As a countermeasure to the removal of Q in the training phase, we consider the following approach. We use 1) particular sizes for layers  $L-1$  and  $L$  of EncNet, and 2) a particular weight matrix  $\mathbf{W}_L^E$ , bias vector  $\mathbf{b}_L^E$ , and activation function  $\sigma_L^E$  at the  $L$ th layer of EncNet to accurately approximate the effect of an SQ between the output layer of EncNet and the input of DecNet. By this procedure, the training provides regions and code words of a "soft quantizer" as a by-product. Once an actual SQ having similar regions and code words is implemented in the device in the (online) communication

$$\{i_1, \dots, i_P\} = \mathbf{Q}^E \left( \sigma_L^E \left( \mathbf{W}_L^E (\dots \sigma_3^E (\mathbf{W}_3^E (\sigma_2^E (\mathbf{W}_2^E \mathbf{y} + \mathbf{b}_2^E)) + \mathbf{b}_3^E) \dots) + \mathbf{b}_L^E \right) \right) \quad (8)$$

$$\mathbf{a}_J^D = \sigma_J^D \left( \mathbf{W}_J^D (\dots \sigma_3^D (\mathbf{W}_3^D (\sigma_2^D (\mathbf{W}_2^D (\mathbf{Q}^D(i_1, \dots, i_P)) + \mathbf{b}_2^D)) + \mathbf{b}_3^D) \dots) + \mathbf{b}_J^D \right) \quad (13)$$

phase, the distortion in (14) is expected to be similar to that achieved in the training phase in terms of (15).

Let us focus on the layers  $L-1$  and  $L$  of EncNet. We set the number of neurons as  $\gamma_{L-1} = \gamma_L = P$ , the weight matrix as  $\mathbf{W}_L^E = \mathbf{I}_P$ , and the bias vector as  $\mathbf{b}_L^E = \mathbf{0}_P$  (see Fig. 1). Thus, we have  $\mathbf{a}_L^E = \sigma_L^E(\mathbf{z}_L^E) = \sigma_L^E(\mathbf{a}_{L-1}^E)$ . The activation function  $\sigma_L^E$  is modeled as the *soft-to-hard quantization* [17] so that  $p$ th output of EncNet is given as

$$\begin{aligned} a_{L,p}^E &= \sigma_L^E(z_{L,p}^E), \quad p = 1, \dots, P \\ &= \sum_{i=1}^{I-1} d_i \tanh(g_i z_{L,p}^E - s_i), \end{aligned} \quad (16)$$

where  $d_i$ ,  $g_i$ , and  $s_i$ ,  $i = 1, \dots, I-1$ , are tunable real-valued parameters and  $\tanh(\cdot)$  is the hyperbolic tangent  $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ . Thus, (16) approximates a non-uniform SQ as a weighted sum of shifted hyperbolic tangents, where 1) the shift coefficients  $\mathbf{s} = [s_1 \dots s_{I-1}]^T$  adjust the decision regions, 2) the level coefficients  $\mathbf{d} = [d_1 \dots d_{I-1}]^T$  adjust the values of the discrete code words, and 3) the step coefficients  $\mathbf{g} = [g_1 \dots g_{I-1}]^T$  control the accuracy of continuous-to-discrete approximation.

1) *DNN Training*: We propose a supervised learning method based on stochastic gradient descent (SGD) and back-propagation [20] to optimize the DNN mappings  $\Gamma^E$  and  $\Gamma^D$  to minimize (15). An advantage of the soft-to-hard quantization (16) is that, unlike a quantizer, it is *differentiable* with respect to its parameters. Therefore, the gradients of the cost function with respect to its parameters can be passed (precisely) in the backpropagation so that the shifts  $\mathbf{s}$  and levels  $\mathbf{d}$  can be jointly optimized with DNN mappings  $\Gamma^E$  and  $\Gamma^D$  in the end-to-end SGD optimization. The hyperparameters  $\mathbf{g}$  are tuned outside the loop, similarly as in [17].

In summary, we apply SGD for the cost function (15) to obtain the parameters of EncNet  $\{\mathbf{W}_l^E, \mathbf{b}_l^E\}_{l=2, \dots, L-1}$ ,  $\mathbf{d}$ , and  $\mathbf{s}$  (with  $\mathbf{W}_L^E = \mathbf{I}_P$  and  $\mathbf{b}_L^E = \mathbf{0}_P$ ) and the parameters of DecNet  $\{\mathbf{W}_j^D, \mathbf{b}_j^D\}_{j=2, \dots, J}$ . Then, the obtained  $\mathbf{d}$  and  $\mathbf{s}$  are used to determine the quantizer regions and discrete code words of the SQ implemented in the system. We call the proposed method as QCS-DNN.

#### IV. SIMULATION RESULTS

Simulation results are presented to assess the rate-distortion performance of the proposed QCS-DNN method.

##### A. Simulation Setup

For the signal model, we consider that 1) each non-zero element of signal  $\mathbf{x}$  follows the zero-mean Gaussian distribution  $\mathcal{N}(0, 1)$ , 2) the sparsity patterns are drawn from a uniform distribution, 3) each element of the measurement noise vector is Gaussian as  $\mathcal{N}(0, 0.0001)$ , and 4) the measurement matrix  $\Phi$  is generated by taking the first  $M$  rows of an  $N \times N$  discrete cosine transform matrix and normalizing the columns

as  $\|\cdot\|_2^2 = 1$ . The QCS-DNN method is implemented by a handcrafted Matlab script. It is trained using a mini-batch SGD of batch size 100 and the Adam optimizer [21] to adaptively tune the learning parameters. The activation functions  $\{\sigma_l^E\}_{l=2, \dots, L-1}$  and  $\{\sigma_j^D\}_{j=2, \dots, J-1}$  are set as the hyperbolic tangent, and  $\sigma_J^D$  is set as the bent identity. For EncNet, we use  $L = 4$  layers with  $\gamma_2 = 5P$ . For DecNet, we use  $J = 5$  layers with  $\gamma_2 = \gamma_3 = \gamma_4 = 4N$ .

For a baseline, we use the compress-and-estimate method QCS-CE [14], that 1) applies uniform SQ of measurements  $\mathbf{y}$  in (1), and 2) reconstructs the signal using the basis pursuit denoising method [22]. Reconstruction accuracy is measured as the normalized MSE as  $D = 10 \log_{10}(\mathbb{E}[\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2] / \mathbb{E}[\|\mathbf{x}\|_2^2])$  (dB), and the rate is measured as  $R = R_{\text{tot}}/N$  (bits), where  $R_{\text{tot}}$  is the total number of quantization bits for a method.

##### B. Rate-Distortion Performance of the Proposed Method

Fig. 2 illustrates the rate-distortion performance of the proposed QCS-DNN method compared to the baseline method for the setups of different signal dimensions. The QCS-DNN method scales well for the increasing signal dimensions, outperforming the QCS-CE method for all signal setups.

Fig. 3 demonstrates the effect of the number of neurons at the output layer of EncNet (i.e.,  $P$ ) on the rate-distortion performance of the QCS-DNN method. It can be observed that  $P = 10$  results in poor performance for high rates since it applies too intensive "compression" at the encoder. For this signal setting,  $P = M = 15$  results in the best performance.

It is worth noting that while QCS-CE requires running a polynomial time CS reconstruction algorithm, the proposed non-iterative QCS-DNN method applies only fast matrix multiplications and activation functions, beneficial in practical real-time applications with high-dimensional signals. Moreover, all results for the QCS-DNN method were obtained without fine tuning the DNN structures and learning parameters specifically for each quantization rate. In summary, once the DNNs are trained even without extensive parameter adjustment, the QCS-DNN method achieves higher reconstruction accuracy with significantly faster decoding and lower computational complexity than the baseline method.

#### V. CONCLUSIONS

This paper addressed deep learning based efficient acquisition of sparse sources through quantized noisy compressive measurements. We proposed a non-iterative DNN based QCS method that aims at minimizing the MSE signal reconstruction distortion. The method benefits from extremely fast and low complexity decoding in the online communication phase. Simulation results showed the superior rate-distortion performance of the proposed method compared to a polynomial-complexity QCS reconstruction method.

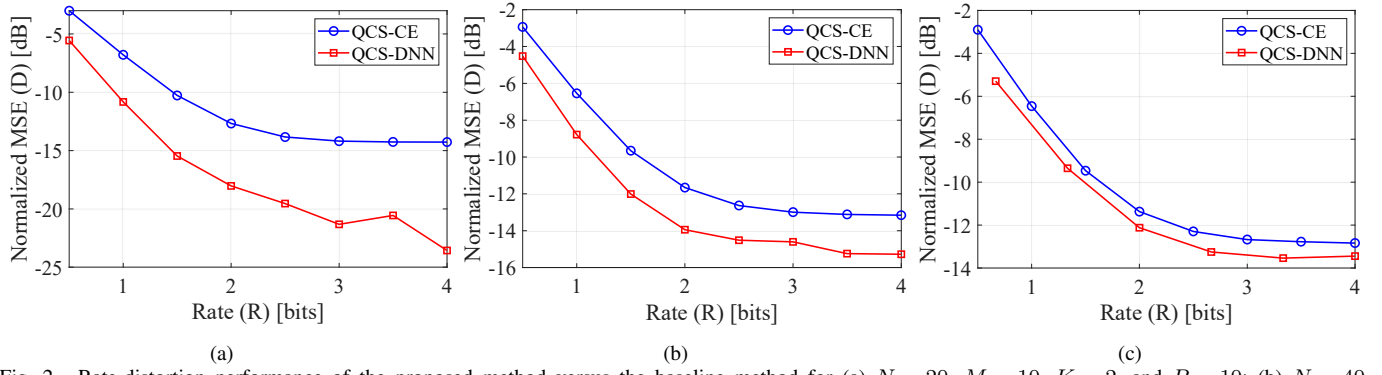


Fig. 2. Rate-distortion performance of the proposed method versus the baseline method for (a)  $N = 20$ ,  $M = 10$ ,  $K = 2$ , and  $P = 10$ ; (b)  $N = 20$ ,  $M = 20$ ,  $K = 4$ , and  $P = 20$ ; (c)  $N = 60$ ,  $M = 30$ ,  $K = 6$ , and  $P = 40$ .

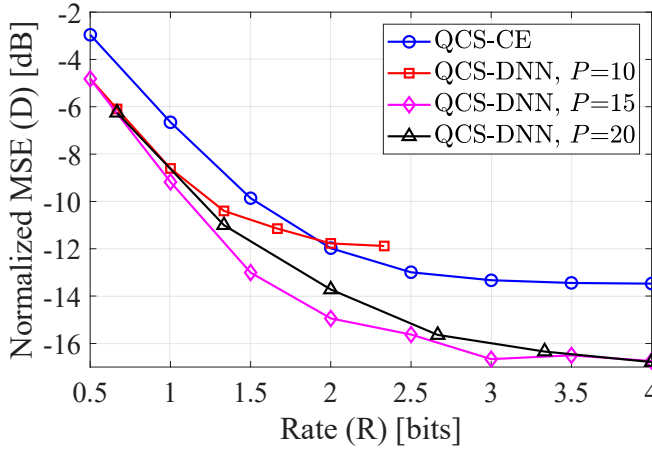


Fig. 3. Rate-distortion performance of the proposed method for  $N = 30$ ,  $M = 15$ ,  $K = 3$ , and the different number of neurons at the output layer of EncNet.

#### ACKNOWLEDGEMENTS

The work has been financially supported in part by Infotech Oulu, the Academy of Finland (grant 323698), and Academy of Finland 6Genesis Flagship (grant 318927). The work of M. Leinonen has also been financially supported in part by the Academy of Finland (grant 319485). M. Codreanu would like to acknowledge the support of the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 793402 (COMPRESS NETS).

#### REFERENCES

- [1] M. Leinonen, M. Codreanu, and G. B. Giannakis, "Compressed sensing with applications in wireless networks," *Found. Trends Signal Process.*, vol. 13, no. 1-2, pp. 1–282, Nov. 2019. [Online]. Available: <http://dx.doi.org/10.1561/2000000107>
- [2] G. Quer, R. Masiero, G. Pillonetto, M. Rossi, and M. Zorzi, "Sensing, compression, and recovery for WSNs: Sparse signal modeling and monitoring framework," *IEEE Trans. Wireless Commun.*, vol. 11, no. 10, pp. 3447–3461, Oct. 2012.
- [3] D. Malioutov, M. Çetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3010–3022, Aug. 2005.
- [4] J. A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1847–1862, Mar. 2010.
- [5] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.
- [6] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
- [7] M. Duarte, M. Wakin, D. Baron, and R. Baraniuk, "Universal distributed sensing via random projections," in *Proc. IEEE Int. Symp. on Inform. Proc. in Sensor Netw.*, New York, NY, USA, 2006, pp. 177–185.
- [8] V. Goyal, A. Fletcher, and S. Rangan, "Compressive sampling and lossy compression," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 48–56, 2008.
- [9] J. Sun and V. Goyal, "Optimal quantization of random measurements in compressed sensing," in *Proc. IEEE Int. Symp. Inform. Theory*, Seoul, Korea, Jun. 28 – Jul. 3 2009, pp. 6–10.
- [10] A. Zymnis, S. Boyd, and E. J. Candès, "Compressed sensing with quantized measurements," *IEEE Signal Process. Lett.*, vol. 17, no. 2, pp. 149–152, 2010.
- [11] A. Shirazinia, S. Chatterjee, and M. Skoglund, "Joint source-channel vector quantization for compressed sensing," *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3667–3681, Jul. 2014.
- [12] M. Leinonen, M. Codreanu, and M. Juntti, "Distributed distortion-rate optimized compressed sensing in wireless sensor networks," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1609–1623, Apr. 2018.
- [13] M. Leinonen, M. Codreanu, M. Juntti, and G. Kramer, "Rate-distortion performance of lossy compressed sensing of sparse sources," *IEEE Trans. Commun.*, vol. 66, no. 10, pp. 4498–4512, Oct. 2018.
- [14] M. Leinonen, M. Codreanu, and M. Juntti, "Practical compression methods for quantized compressed sensing," in *Proc. IEEE Int. Conf. on Comp. Commun.*, Paris, France, Apr. 29–May 2 2019, pp. 756–761.
- [15] A. Mousavi, A. B. Patel, and R. G. Baraniuk, "A deep learning approach to structured signal recovery," in *Proc. Allerton Conf. Commun., Contr., Comput.*, Illinois, USA, Sep. 29–Oct. 2 2015, pp. 1336–1343.
- [16] B. Sun, H. Feng, K. Chen, and X. Zhu, "A deep learning framework of quantized compressed sensing for wireless neural recording," *IEEE Acc.*, vol. 4, pp. 5169–5178, Sep. 2016.
- [17] N. Shlezinger and Y. C. Eldar, "Deep task-based quantization," Aug. 2019, available at <https://arxiv.org/abs/1908.06845>.
- [18] N. Shlezinger, Y. C. Eldar, and M. R. D. Rodrigues, "Hardware-limited task-based quantization," *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5223–5238, Oct. 2019.
- [19] P. T. Boufounos and R. G. Baraniuk, "1-bit compressive sensing," in *Proc. Conf. Inform. Sciences Syst.*, Princeton, NJ, USA, Mar. 19–21 2008, pp. 16–21.
- [20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.
- [21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, available at <https://arxiv.org/abs/1412.6980>.
- [22] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Scient. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.