Selective and Compressive Sensing for Energy-Efficient Implantable Neural Decoding

Aosen Wang[1], Chen Song[1], Xiaowei Xu[1], Feng Lin[1], Zhanpeng Jin[2] and Wenyao Xu[1]
[1]CSE Dept., SUNY at Buffalo, NY, USA
[2] ECE Dept., SUNY at Binghamton, NY, USA
{aosenwan, csong5, xiaoweix, flin28, wenyaoxu}@buffalo.edu, zjin@binghamton.edu

***Abstract*—The spike classification is a critical step in implantable neural decoding. The energy efficiency issue in the sensor node is a big challenge in the entire system. Compressive sensing (CS) provides a potential way to tackle this problem. However, the overhead of signal reconstruction constrains the compression in sensor node and analysis in remote server. In this paper, we design a new selective CS architecture for wireless implantable neural decoding. We implement all the signal analysis on the compressed domain. To achieve better energy efficiency, we propose a two-stage classification procedure, including a coarse-grained screening module with softmax regression and a fine-grained analysis module based on deep learning. The screening module completes the low-effort classification task in the front-end and transmits the compressed data of high-effort task to remote server for fine-grained analysis. Experimental results indicate that our selective CS architecture can gain more than 50% energy savings, yet keeping the high accuracy as state-of-the-art CS architectures.**

I. Introduction

Neural decoding is one of the most significant procedure to investigate the neural activity mechanisms [1]. The spike classification is an effective method to help understanding the neural encoding and decoding. The typical systems for spike classification comprise Electroencephalography (EEG) signal acquisition sensor and remote processing center. Huge amount of data exchange in wireless communication makes the sensor design face with severe energy efficiency issues. Compressive sensing (CS) is a promising way to relieve the energy issue in the sensor node. It samples the signal information directly, breaking the conventional Shannon-Nyquist rule on the data sampling. In practice, the CS-based sampling rate is much lower and can reduce a significant portion of energy consumption on processing and wireless communication.

There are some prior work on spike classification and CS. Zhang *et al.* proposed a compact microsystem to apply CS into implantable neural recording [2]. Suo *et al.* optimized the neural recording system by a two-stage sensing and sparsifying dictionary [3]. Vogelstein *et al.* demonstrated the good performance of applying supporting vector machine to spike classification [4]. These work split the signal recording and classification into two independent procedures, and only concentrate on the single-facet optimization. *It will be more effective to consider compressed domain data analysis, i.e., combining the biosignal recording and classification together.*

In this paper, we propose a selective CS architecture to combine the signal acquisition and analysis into the front-end sensing architecture to further reduce the wireless transmission burden. We design a coarse-grained screening module to evaluate the classification effort in the front-end. It can process the low-effort task and transmit the high-effort task back to the server. We also deploy the deep learning algorithm to execute the fine-grained analysis. By seamlessly cooperating the two proposed modules, the selective CS architecture can provide more than 50% energy savings for wireless implantable neural decoding.

The remainder of this paper is organized as follows: Section II introduces the background of the Quantized CS theory. Our proposed selective CS architecture is discussed in Section III, and Section IV presents the related experiments and evaluations. The paper is concluded in Section V.

II. Preliminary of Compressive Sensing

The compressive sensing theory is a new emerging analog-to-information sampling scheme. We assume that the x is an N-dimension vector and sampled using M-measurement vector y:

$$y = \Phi x, \tag{1}$$

where $\Phi \in R^{M \times N}$ is the sensing array and M is defined as the sampling rate. The elements in Φ are random variables. Because of $M << N$, the formulation in Eq. (1) is undetermined. However, under certain sparsity-inducing basis $\Psi \in R^{N \times N}$, the signal x can be represented by a set of sparse coefficients $u \in R^N$:

$$x = \Psi u. \tag{2}$$

Therefore, based on Eq. (1) and (2), the sparse vector, u, can be represented as follows:

$$y = \Phi \Psi u = \Theta_{M \times N} u, \tag{3}$$

where $\Theta_{M \times N} = \Phi \Psi$ is an $M \times N$ matrix, called the measuring matrix. In practical applications, original signals need to be quantized for transmitting. Then the compressed signal, y, is processed by a quantization model formulated as follows:

$$\widehat{y} = Q_b(y), \tag{4}$$

where $Q_b(.)$ is the quantization function [5], [6] , and $\widehat{y}$ is the quantized representation of y with b bits. Due to the prior knowledge that the unknown vector u is sparse, u can be estimated by ℓ_1 minimization to approximate the optimal ℓ_0 minimization formulation as follows:

$$\widehat{u} = min \ \|u\|_1 \quad s.t. \quad \|\widehat{y} - \Theta u\| < \epsilon, \tag{5}$$

where ϵ is the reconstruction error margin. The ℓ_1 minimization is convex and can be solved within the polynomial time.

978-1-4799-7234-0/15/$31.00 ©2015 IEEE

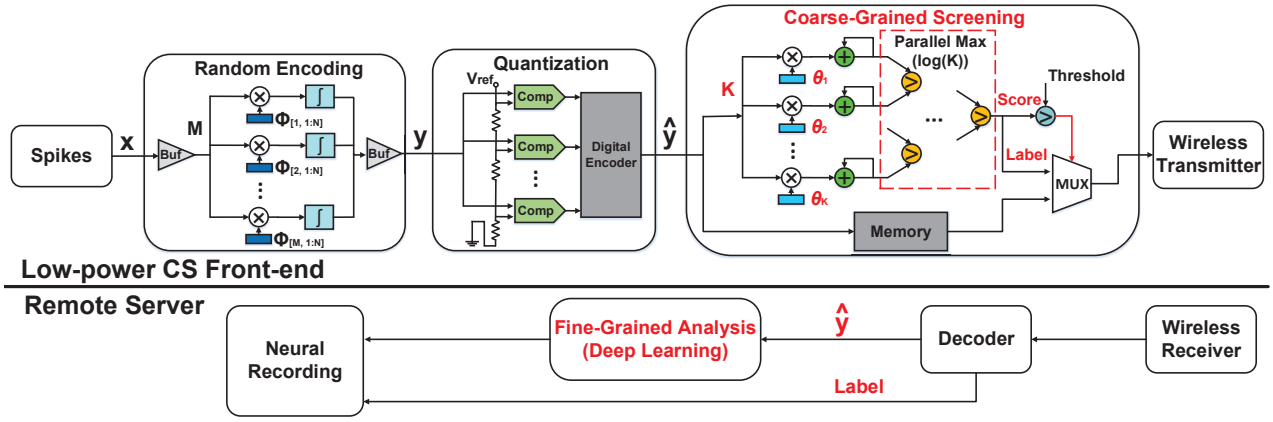

Fig. 1. The block diagram of the selective CS architecture for wireless implantable neural decoding.

Therefore, the reconstructed signal, $\widehat{x}$, is retrieved by:

$$\widehat{x} = \Psi\widehat{u}. \tag{6}$$

III. Selective CS Architecture

In this section, we present our proposed selective CS architecture for neural decoding. We design a coarse-grained screening module to judge the effort of signal identification in the front-end. Correspondingly, we use a robust deep learning algorithm as the fine-grained analysis procedure to process the high-effort task in the server-end.

A. Architecture Overview

The selective CS architecture for wireless implantable neural decoding is illustrated in Fig. 1. The entire architecture includes a low-power CS front-end and a remote server. Our ultimate goal is to optimize the energy efficiency in the front-end.

The front-end design comprises *three* key components, a random encoding module, a quantization module and a coarse-grained screening module. As shown in Fig. 1, analog N-dimension raw sensor signal x is compressed into M-dimension measurements y in the random encoding module. The random encoding module consists of M branches with each completing a randomized combination for one measurement. Every branch includes a multiplier, a column vector in sensing array Φ and an integrator to accumulate the intermediate results. In the quantization module, there are b comparators and a digital encoder. Each comparator outputs a binary decision of comparing the input analog signal and a reference voltage level. The digital encoder organizes the final quantization result $\widehat{y}$ based on these comparison decisions. The newly proposed coarse-grained screening module analyzes the compressed measurements $\widehat{y}$. It outputs the category prediction and a confidence score. If the score is larger than the predefined threshold, the wireless transmitter sends the final prediction to the remote server. Otherwise, it streams the compressed measurements $\widehat{y}$ to the server for fine-grained analysis by deep learning algorithm.

Note that the coarse-grained screening module and fine-grained analysis module are two most significant components to improve the energy efficiency of the CS front-end, yet keeping the classification accuracy. In the following section, we continue to discuss the design of these two modules in detail.

B. Coarse-Grained Screening Module

It is important to have a reliable hint to determine the effort of signal category prediction in the front-end. Some low-effort task can be completed in the sensor node, avoiding the energy overhead of transmitting compressed data. To this end, we consider the softmax regression, a probabilistic model, to construct the coarse-grained screening module.

The softmax regression [7] is the extensive form of logistic regression to deal with multi-class classification problem. As the logistic regression, the hypothesis function h_θ for the softmax regression outputs a probability vector:

$$h_\theta(z) = \begin{bmatrix} P(l=1|z,\theta) \\ P(l=2|z,\theta) \\ \vdots \\ P(l=K|z,\theta) \end{bmatrix} = \frac{1}{\sum_{k=1}^{K} e^{\theta_k z}} \begin{bmatrix} e^{\theta_1 z} \\ e^{\theta_2 z} \\ \vdots \\ e^{\theta_K z} \end{bmatrix}, \tag{7}$$

where θ is the weight matrix, denoted as $\theta = [\theta_1, \theta_2, \ldots, \theta_K]$. Each weight component θ_j is a weight vector as in the logistic regression. It is reasonable to identify the input signal as the category with the largest conditional probability. Let z_i be the input feature vector in the training set, and l_i be its corresponding multi-class label, which ranges from 1 to K. The cost function to evaluate the hypothesis function h_θ is:

$$J(\theta) = -\sum_{i}\sum_{j=1}^{K}(l_i == j) log(\frac{e^{\theta_j z}}{\sum_{k=1}^{K} e^{\theta_k z}}), \tag{8}$$

where $l_i == j$ is to judge the equality. If l_i is not equal to the label j, the judge is false, denoted as "0". Otherwise, the judge is true, as "1". Similarly, we can use the gradient descent algorithm to minimize this cost function to search for the optimal weight matrix θ. The derivative of the cost function with respect to the specific weight vector θ_k is as follows:

$$\nabla_{\theta_j} J(\theta) = -\sum_{i}[z_i((l_i == j) - P(l_i = j|z_i, \theta))]. \tag{9}$$

This training phase is computation-intensive, and will be accomplished offline. Therefore, we implement the prediction

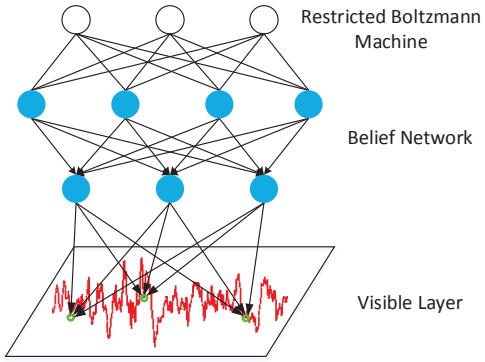

Fig. 2. The framework of deep belief networks.

phase into the front-end to analyze the compressed measurements with the pre-trained parameters. The formal formula of the prediction phase is as the following:

$$P(l=j|z,\theta)=\frac{e^{\theta_j z}}{\sum_{k=1}^{K} e^{\theta_k z}}. \tag{10}$$

When z is given, the l with the maximal probability is the prediction category. We also take this maximal probability as the confidence score S. It is reasonable that larger score has more probability of correct classification.

The prediction phase of softmax has low hardware complexity yet robust performance, as the coarse-grained screening module shown in Fig. 1. When compressed measurements come, the screening module has two parallel procedures, one is to store the input vector and the other is to start the prediction of softmax method. In the prediction procedure, We first calculate the score $e^{\theta_j z}$ by parallel chains, where each chain is equipped with a multiplier, an accumulator and an exponential calculator. Then the parallel max structures compute the maximal probability and record the related category. The confidence score compares with the pre-defined threshold Sth. If the confidence score is larger than the threshold, the selector, MUX module, chooses the prediction result for wireless transmission, trusting the softmax classification. Otherwise, the MUX outputs the compressed measurements buffed in the memory.

C. Fine-Grained Analysis Module

When the coarse-grained screening module bypasses the compressed measurements to remote server, a fine-grained classifier is required for accurate classification without energy concerns. We adopt deep belief networks (DBN) [8] to construct the fine-grained analysis module. DBN is a popular algorithm in deep learning, with superior performance to the shallow learning. It unsupervised learns the effective signal representation from the training set. A typical framework of DBN is illustrated in Fig. 2. It can be divided into three key parts, visible layer, belief network and restricted Boltzmann machine (RBM). The visible layer uses the compressed data $\widehat{y}$. The belief network extracts the feature from the visible layer. There can be multiple layers of belief networks. The top layer is the RBM, which uses undirected neuron connections. After training this network, including the pre-training and finetuning, the DBN can obtain signal's predominant representation automatically for the classification. If we add a classifier on the top of RBM, the entire framework can accurately complete the classification task.

TABLE I. ACCURACY OF ALL THE CLASSIFIERS

CR	5%	10%	20%	30%
Comp. + Softmax (%)	72.80	80.67	97.27	98.4
Comp. + DBN (%)	**85.67**	**92.33**	**97.33**	**98.6**
Recon. + SVM (%)	84.24	91.40	96.13	98.19

IV. EXPERIMENTS

In this section, we evaluate the performance of our proposed selective CS architecture. We first introduce the experimental setup and compare the accuracy of different classifiers. After demonstrating the accuracy of the confidence score of softmax, we investigate the trade-off improvement compared with traditional CS architectures.

A. Experimental Setup

We choose 3000 spikes from the *difficult-noise-01* case in spike sorting dataset [9]. Each spike has the length of N=64. The first 1500 spikes are taken as the training set and the other spikes as the testing set. We choose four compression ratios ($CR = M/N$) in the experiment, 5%, 10%, 20% and 30%. We use the inverse discrete wavelet transform (IDWT) as the sparsity-inducing basis Ψ. The Bernoulli random variable is taken as the sensing array and uniform quantization strategy is applied. The energy model is defined as $E = C \times M \times b$, where the average energy consumption C = 3 nJ/bit, based on an efficient 350 μW MSK/FSK transmitter [10]. *We take traditional CS reconstruction plus SVM as the benchmark. The DBN runs on compressed-domain data*. We use two hidden layers with 48 neurons for each. The top classifier is the softmax regression. The activation function is set as "sigmoid", the batch size is 10 and the learning rate is 0.1.

B. Accuracy of Classifiers

We examine the accuracy of three classifiers in this section, the `compressed-domain+softmax` in the coarse-grained screening, the `compressed-domain+DBN` in the fine-grained analysis, and the `reconstruction+SVM` for the state-of-the-art method. All the three cases have been finetuned. The related results are shown in TABLE I.

We can find that the DBN in compressed domain holds the best accuracy. The softmax in compressed domain is worse than the traditional `reconstruction+SVM` when CR is small (less than 10%), but has much better accuracy than SVM when CR is large (more than 20%). In our proposed architecture, the sofamax has already gained a high accuracy for the energy-oriented coarse-grained screening. Its uncertainty will be processed by the DBN for a better accuracy. We discuss the detail in the next subsection.

C. Confidence Score of Softmax Regression

We investigate the distribution of confidence score S in softmax regression to discuss the threshold Sth in the coarse-grained screening module. We show all the results in Fig. 3 under four CRs. The blue points are the confidence scores for each testing sample and the red rectangles mark the misclassifications.

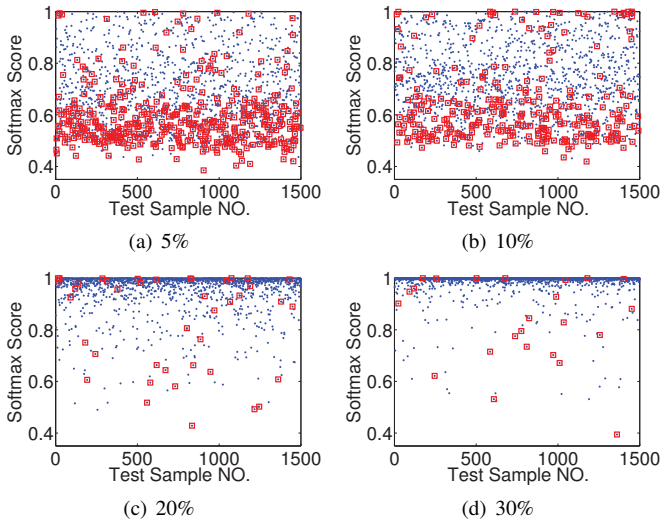

Fig. 3. The confidence scores of softmax regression under different CRs.

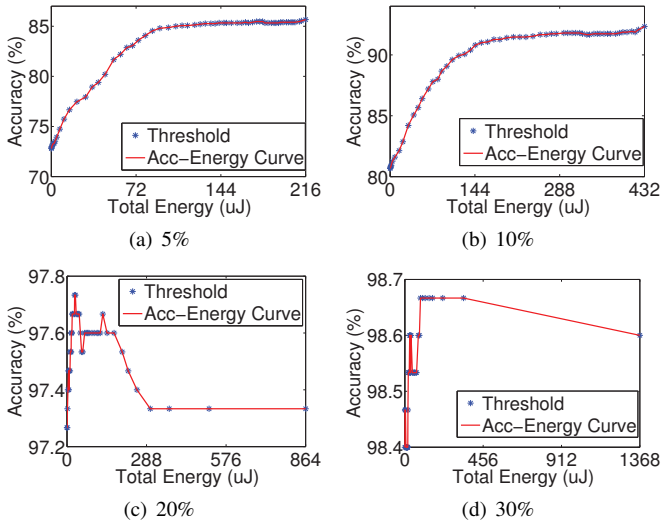

Fig. 4. The accuracy-energy curve under different thresholds.

We can observe that the majority misclassifications happen when the softmax score S is small, between 0.4 and 0.7, in the cases of CR = 5% and 10%. While in the other two cases with higher CR, there are much fewer misclassifications, resulting in a random-like distribution. We can conclude that the misclassifications are more probable on the small softmax score. This trend strongly demonstrates the feasibility to adopt softmax regression to design our screening module.

D. Trade-off on Accuracy and Energy

Based on the observations of the softmax score, we depict the accuracy-energy curve by taking Sth as a variable. In this experiment, we range Sth from 0.01 to 1.00 with the step of 0.01. The related curves are shown in Fig. 4.

We can see that for the low CR cases, 5% and 10%, the accuracy first increases rapidly as the total energy of processing all the testing samples grows, and then saturates. This is because the DBN holds much better performance than softmax, as shown in TABLE I, resulting from its unsupervised feature extraction. However, in the large CR cases, 20% and 30%, the accuracy has small fluctuations as the total energy increases. The performance of softmax approximates to the DBN. Another interesting observation is the combination of the two methods may have better accuracy than the maximal between them. It is possible that some misclassifications of DBN are accidentally recognized by softmax.

Furthermore, if we take the performance of the `reconstruction+SVM` in TABLE I as the accuracy bound, we can find that the four thresholds of softmax under different CRs are 0.62, 0.70, 0 and 0, respectively. The traditional SVM case needs to transmit all the compressed measurements back to the remote sever, but our proposed architecture only requires much less data transmission burden. The energy savings are 60.27%, 54.13%, 100% and 100%, under the four CRs. This demonstrates the accuracy-energy trade-off improvement of our proposed selective CS architecture.

V. Conclusion

In this paper, we presented a selective CS architecture for wireless implantable neural decoding. We designed two modules, coarse-grained screening module with softmax regression and fine-grained analysis module with DBN, for the different effort-level classification tasks. Experimental results validated the data analysis in compressed domain. By the collaboration of the two proposed modules, our selective CS architecture gained more than 50% energy savings,while keeping the high accuracy as the traditional CS architecture.

Acknowledgment

This work is in part supported by NSF CNS-1423061/1422417, ECCS-1462498/146247 and CNS-1547167.

References

[1] M. Aghagolzadeh and K. Oweiss, "Compressed and distributed sensing of neuronal activity for real time spike train decoding," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 17, no. 2, pp. 116–127, 2009.

[2] Zhang, *et al.*, "An efficient and compact compressed sensing microsystem for implantable neural recordings," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 8, no. 4, pp. 485–496, 2014.

[3] Suo, *et al.*, "Energy-efficient multi-mode compressed sensing system for implantable neural recordings," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 8, no. 5, pp. 648–659, 2014.

[4] Vogelstein, *et al.*, "Spike sorting with support vector machines," in *Engineering in Medicine and Biology Society, 2004. 26th Annual International Conference of the IEEE*, vol. 1, 2004, pp. 546–549.

[5] Wang, *et al.*, "Quantization effects in an analog-to-information front end in eeg telemonitoring," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 62, no. 2, pp. 104–108, 2015.

[6] A. Wang, Z. Jin, C. Song, and W. Xu, "Adaptive compressed sensing architecture in wireless brain-computer interface," in *Proceedings of the 52nd Annual Design Automation Conference*. ACM, 2015, p. 173.

[7] Huang, *et al.*, "Active learning: learning a motor skill without a coach," *Journal of neurophysiology*, vol. 100, no. 2, pp. 879–887, 2008.

[8] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 609–616.

[9] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural computation*, vol. 16, no. 8, pp. 1661–1687, 2004.

[10] J. L. Bohorquez, A. P. Chandrakasan, and J. L. Dawson, "A 350 w cmos msk transmitter and 400 w ook super-regenerative receiver for medical implant communications," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 4, pp. 1248–1259, 2009.