Compression of Spike Data Using the Self-Organizing Map

António R. C. Paiva, José C. Príncipe and Justin C. Sanchez
Computational NeuroEngineering Laboratory
Electrical and Computer Engineering Department
University of Florida, Gainesville, FL 32611 USA
{arpaiva,principe,justin}@cnel.ufl.edu

***Abstract*— Motivated by current attempts to use wireless in Brain-Machine Interfaces (BMIs), this paper presents a method for the compression of spike data. Supported by Vector Quantization (VQ) theory, we use a 1-dimensional Self-Organizing Map (SOM) to quantize vectors of input samples. The indices are entropy coded to further reduce the necessary bandwidth, taking advantage of the non-uniform frequency of firing of the SOM processing elements (PEs). The complexity of the use of the SOM is also considered and addressed. After training several SOMs, the method was simulated with real data achieving compression ratios as high as 185.7:1, i.e. a bitrate of 862 bits-per-second-per-channel, assuming sampling at 20 kHz with 8 bits-per-sample (bps).**

I. Introduction

Brain-Machine Interfaces (BMI) have been receiving increasing attention in the last years. In this context, a large number of channels are collected from brain areas through multi-electrode arrays. Then, these signals are transferred to a processing unit which decodes the spike firings and predict the intended action. However, due to the considerable complexity of the algorithms used, the processing is typically done "remotely" [1]. The most common approach is to wire connect the subject to the processing unit. This is uncomfortable to the subject and can greatly restrict the experimental setup. A natural alternative is to replace the connectors with a wireless link. In this situation, a unit connected to the electrodes would collect, possibly performing some pre-processing, and transmit through a wireless communication channel the spike data to the processing unit.

Although very attractive, the usage of a wireless link gives rise to other challenges. Due to the large number of channels to be transmitted and the high bitrate associated with each one, the bandwidth required for transmitting all the signals is very large. For instance, even if only 32 channels are selected, and considering sampling at 20 kHz with 16 bps, a total bandwidth of more than 9.76 Mbps is required. Associated with this is the fact that high bandwidth imply high power consumption. Since current neural decoding algorithms use only the firing rate of neurons [1], [2], an efficient compression method would be to perform the spike detection onsite and then transmit only when a spike occurs, or even the result of binning (i.e., counting of the number of spikes in a channel during a given time interval). In fact, this is the method of choice for current attempts of using a collecting unit communicating through a wireless link [3]. The great advantage of this approach is that only a very small bandwidth is required to transmission. On the other hand, it does not provide the processing unit with any other information than the firing rate, which means neither spike sorting nor correction of spike detection can be made. Therefore, a large gap exists in the amount of information available to the processing unit between these two approaches. A not so drastic alternative, analyzed from the communication point of view by Bossetti *et al.* [4], is to do spike detection and transmit all the samples during the spike. This allows us to perform spike sorting if such is intended. Nevertheless, there is still no possibility to recover spikes not detected.

Following a quite different path, our approach tries to transmit as much information as possible, under the constraints on low bandwidth and low complexity, leaving the responsibility of spike detection and spike sorting to the processing unit. This allows for more sophisticated algorithms to be used without the computational restrictions of a miniaturized unit, while maintaining all the advantages of a remote collecting unit communication through wireless. To do so, we vector quantize the input, using a Self-Organizing Map (SOM), and transmit and entropy coded version of the index of an approximate reconstruction vector. Since we group the samples in non-overlapping vectors, we are able to achieve great compression ratios while preserving most of the structure of the signal. The theoretical concept behind this approach is that of vector quantization (VQ), from which the SOM can be regarded as a particular case. In the work of Shannon [5], it is proven that due to correlation between subsequent samples, grouping samples and coding then jointly results in compression closer to optimal. Based on this concept, several VQ methods were developed since then [6].

The remainder of this paper is organized as follows. In Section II we expose the architecture of our strategy. Then, in Section III, some considerations are made concerning the implementation of this method, and results are presented in Section IV. Finally, some conclusions are drawn in Section V.

II. System overview

Figure 1 presents a block diagram of the main elements of the communication process considered in this paper, and specifically showing the base elements of the encoder and decoder. As is explicitly depicted in Fig. 1, compression is

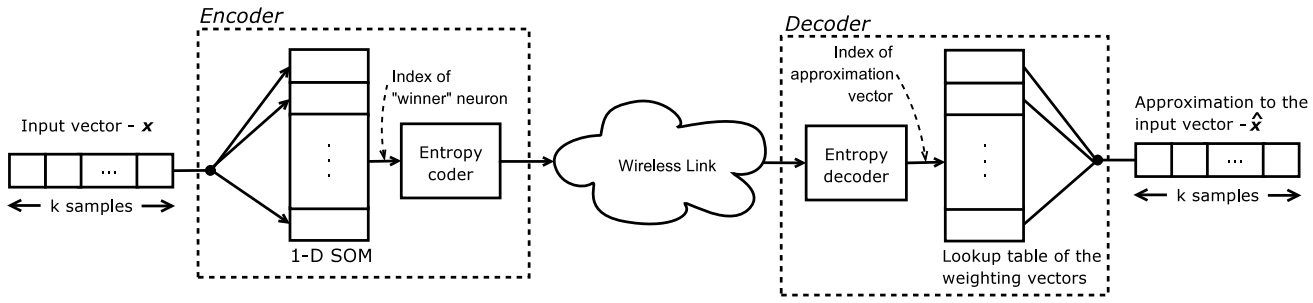

Fig. 1. Block diagram of the communication process.

achieved through a two step process. First, through quantization of the input vector due to the application of a SOM and, secondly, through entropy coding of the index resulting from the application of the SOM. Although this process is lossy (i.e. there is loss of information), this happens only in the quantization step. The algorithm begins by grouping k samples[1] from the input such as to form a vector, in which each sample is a component of this vector. This can be thought as a non-overlapping sliding window over the input. This vector is then used as input to the SOM, resulting in the index of the firing neuron for this vector. Then, the index is entropy coded and transmitted through the communication channel. Conversely, at the receiver, the index is first entropy decoded and used as the reference to a lookup table with the weighting vectors of processing elements (PEs), or neurons, of the SOM.

It is important to note that the SOM, acting as a vector quantization method, insures through its competitive strategy that the firing neuron is the one which the corresponding weighting vector is nearest (according to L_2 metric, in our case) in state space, i.e., the index of the firing neuron is the one for which $\arg\min_i ||x - w_i||$, where x is the input vector. In other words, if we use this weighting vector as an approximation of the input vector, it will yield the smallest reconstruction error. Since we selected the weighting vector with the smallest reconstruction error, due to the training of the SOM, at the receiver, this vector can be used as a good approximation of the input vector.

The entropy coding method to be used can be any. For example, arithmetic coding or Huffman coding. However, due to the specific characteristics of each signal it is not possible to estimate in advance the PDF of the indices, thus an adaptive method must be used. This, nevertheless, poses no restrictions since most entropy coding methods have an adaptive version.

III. Implementation remarks

In our method, as should be obvious, the application of the SOM is the most computationally demanding task. While the entropy coders are generally simple algorithms, the naive application of the SOM would require kN multiplies and $(k-1)N$ adds for each vector. Of course, such an implementation would require a powerful DSP, which would consume considerable power to operate. But, as was stated right at the beginning of this paper, the application serving as our motivation poses tight constraints in the power consumption. However, since the competive process of the SOM is basically a search for the nearest neighbor in high dimension space, fast search algorithms can be used instead.

Several fast search algorithms where developed in the context of computational geometry (see e.g. [7]). Although, generally these algorithms are very fast ($\mathcal{O}(\log_2 N)$ time), most where developed for approximate search, which would increase the error and render our approximation through the SOM non-optimum. In the work by Arya *et al.* [8] several algorithms are presented, most of then based on KD-trees. KD-trees break the space in hyperrectangles which very unlikely match the Voronoi division of the space. However, it is also shown how using KD-trees plus a small distance search in the neighborhood of the KD-tree hyperrectangle we can ensure that the found vector is truly the nearest. Comparing with the original algorithm for the application of the SOM this represents a great improvement. For instance, for a vector length of 20 and a SOM with 64 PEs, instead of performing 2496 operations (1280 multiplies and 1216 adds) for each vector, we would need only a search with 6 steps, plus only a few distance computations; e.g. for 3 distance computations (60 multiplies and 57 adds) is enough to have a high probability of being selecting the true nearest vector.

IV. Results

A. Data

Multielectrode array recordings were collected from male Sprague-Dauley rats performing a go-no go lever pressing task. Array configurations of $2 \times 850\mu$m tungsten electrodes were chronically implanted in the forelimb region of M1 (+1.0mm anterior, 2.5mm lateral of bregma). Neuronal activity was collected with a Tucker-Davis recording rig with sampling frequency of 24414.1Hz and digitized to 16 bits of resolution. Before being stored to disk the neuronal potentials were band-pass filtered between 0.5 and 12000Hz. From these recordings we considered only channels 6 and 7. These signals where spike detected under human supervision.

B. SOM Training

For training of the SOM we began by creating a signal with content from both channels. Assuming the width of a spike as

[1]Throughout this paper we will use k as the vector length, and N as the number of neurons in the SOM.

TABLE I

COMPRESSION RESULTS ACHIEVED WITH OUR METHOD. THE SOM QUANTIZATION FACTOR IS THE REDUCTION OF THE BITRATE THROUGH APPLICATION OF THE SOM, WHILE THE CHANNELS BITRATE AND COMPRESSION RATIO TAKES INTO ACCOUNT ALSO ENTROPY CODING OF THE INDICES, CONSIDERING SAMPLING AT 20 KHZ WITH 8 BITS-PER-SAMPLE (BPS) AND "IDEAL" ENTROPY CODING. THE RECONSTRUCTION ERROR IS THE AVERAGE OF THE ABSOLUTE DIFFERENCE BETWEEN SAMPLES DURING SPIKES.

Vector length	No. of PEs	SOM quantization factor	Channel 6			Channel 7		
			Bitrate (kbits/sec)	Compression ratio	Normalized reconstruction error ($\times 10^{-2}$)	Bitrate (bits/sec)	Compression ratio	Normalized reconstruction error ($\times 10^{-2}$)
10	32	16	4.14	37.7 : 1	3.04	3.76	41.6 : 1	2.37
	64	13.3	5.44	28.7 : 1	2.61	4.96	31.5 : 1	2.00
	128	11	6.73	23.2 : 1	2.22	6.18	25.3 : 1	1.70
20	64	26	1.71	91.6 : 1	3.37	1.27	123.2 : 1	2.58
	128	22	2.21	70.7 : 1	3.02	1.95	80.0 : 1	2.35
	256	20	2.75	56.9 : 1	2.45	2.34	66.9 : 1	2.04
30	128	34	1.02	152.8 : 1	3.80	0.84	185.7 : 1	3.12
	256	30	1.44	108.7 : 1	3.56	1.22	128.4 : 1	2.90

27 samples, we extracted the same number of spikes from each channel, and concatenated 2% (relative to the samples of from spikes) of noise samples (1% originating from each channel). To speed the training of the SOM, for the first five SOMs considered, we used 1000 spikes and 540 samples of noise from each channel. For the remaining SOMs, we used 4000 spikes and 2160 samples of noise from each channel. These values where chosen to ensure that the number of training vectors is greater than the vector length times the number of PEs.

The training signals were prepared with two concerns in mind. First, to emphasize the importance of the spikes in the training of the SOM. If we had used real data directly, due to the PDF matching property of the SOM, only a very little amount of PEs would be assigned to the approximation of spikes, which is the most important element in the data. Thus, proceeding this way we make the SOM tuned to approximate spikes. Secondly, the presence of a small percentage of background samples (noise) makes the SOM "noise aware" and prevent PEs dedicated to approximate spikes from firing through background areas which would cause misinterpretation, at the decoder, on the content in the input waveform.

The training of each SOM was made in two steps. We used 1000 iterations for the self-organization step and $100 \times N$ for the converge. In the first step, the learning rate exponentially decreases between 0.1 and 0.02, and the neighborhood function width parameter also exponentially decreases between N and 0.4. In the second step, both parameters also decrease exponentially but the former between 0.02 and 0.005, and the later between $N/10$ and 0.25.

C. Tests

To test the whole framework we simulated the process using data from the same channels used for training of the SOM (400,000 samples of each), but we were careful not to use the portions of the dataset used for training. However, we did one simplification. We assumed the knowledge of the histogram of the indices in advance. This is, for testing purposes, we simply applied the SOM and calculated the entropy of the resulting histogram of the indices, as $h = -\sum_{i=1}^{N} p_i \log_2 p_i$, where p_i is the relative frequency of the index. Then, we used the entropy (i.e., the average number of bits per index) to calculate the compression ratios, with $8k/h$, assuming 8 bps. The reconstruction error was calculated as the average absolute value of the difference between the original and reconstructed signal, but considering spikes only, i.e., just for the samples that correspond to a spike, accordingly to the spike detection made.

D. Analysis

Table I presents the compression results. The first two columns specify the characteristics of the SOM, namely, the vector length (and, thus, the dimension of state space) and the number of PEs, respectively. The third column indicates the quantization factor or compression ratio just from the application of the SOM. It is calculated as the number of bits in the input vector over the number of bits of an index of a PE from the SOM, i.e., $8k/(\log_2 N)$. The bitrates, specified in kbits-per-second, and the compression ratios presented take into consideration both the quantization factor of the SOM and the compression due to entropy coding. We achieved results as impressive as 185.7, i.e. 862 bits-per-second. Although these results assume prior knowledge of the histogram of the indices, and hence a effective implementation would not perform as good, they show the extraordinary potential of this approach.

It is visible that much better compression ratios are achieved as the vector length is increased, at the expense of a small decrease in fidelity. But, as Fig. 2 reveals, the increase in distortion inherently associated to the increase in the quantization factor, does not increase proportially to the dimension of state space. This is in fact one of the fundamental consequences of the rate-distortion theory. Due to the intersample correlation, increasing the vector length, and consequently the state space dimension, does not imply a proportional increase of *dimensionality* of the vector subset in space. From Fig. 2, we also notice that for smaller vector length the reconstruction fidelity is greater, both in the precision as the reconstruction signal follows the original as in the amplitude of the spikes. In fact, the precision in reproducing the amplitude of the spikes is

one of greater problems to the SOM since the percentage of samples corresponding to the peaks of the spikes is very low.

V. Conclusions

We have presented a method for the compression of spike data, motivated by the BMI paradigm. Our method uses a SOM as a vector quantization method, followed by entropy encoding of the indices of the firing PEs. The key advantage of using a SOM instead of other VQ methods, and of crucial importance for future work, is that the SOM is topology preserving, i.e., neighbor PEs in the SOM correspond to neighbors in state space. This allows for the division of PEs in adjacent classes and coding of PEs as classes when the distortion is bellow some threshold. Likewise, searches can be made approximate with very small increase in distortion, following the idea of hierarchical vector quantization [9]. Nevertheless, the fact that a SOM is also a particular case of a VQ method allows us to achieve great compression ratios while providing the freedom to optimally and smoothly balance the fidelity of the reconstruction versus the desired/allowable bitrate to the communication link.

Future work will be mostly concentrated on exploring how the properties of the SOM can optimally being adjusted with our knowledge of the data. Furthermore, we have the conscience that ultimately a fundamental component of this research must be to quantify the effects of quantization due to the application of the SOM in spike detection and spike sorting.

Acknowledgments

This work was partially supported by Fundação para a Ciência e a Tecnologia under grant SFRH/BD/18217/2004, and DARPA under grant ONR-450595112.

References

[1] J. W. *et al.*, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, vol. 408, no. 16, pp. 361–365, Dec. 2000.

[2] D. M. Taylor, S. I. H. Tillery, and A. B. Schwartz, "Direct cortical control of 3d neuroprosthetic devices," *Science*, vol. 296, no. 5574, pp. 1829–1832, June 2002.

[3] K. D. Wise, D. J. Anderson, J. F. Hetke, D. R. Kipke, and K. Najafi, "Wireless implantable microsystems: high-density electronic interfaces to the nervous system," *Proceedings of the IEEE*, vol. 92, no. 1, pp. 76–97, Jan. 2004.

[4] C. A. Bossetti, J. M. Carmena, M. A. L. Nicolelis, and P. D. Wolf, "Transmission latencies in a telemetry-linked brain-machine interface," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 919–924, June 2004.

[5] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, July 1948, continued 27(4):623-656, October 1948.

[6] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Press/Springer, 1992.

[7] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, 2nd ed. Springer-Verlag, 2000.

[8] S. Arya and D. M. Mount, "Algorithms for fast vector quantization," in *Proc. of Data Compression Conference, DCC'93*, 1993, pp. 381–390.

[9] S. P. Luttrell, "Hierarchical vector quantisation," *IEE Proceedings–Communications, Speech and Vision*, vol. 136, no. 6, pp. 405–413, Dec. 1989.

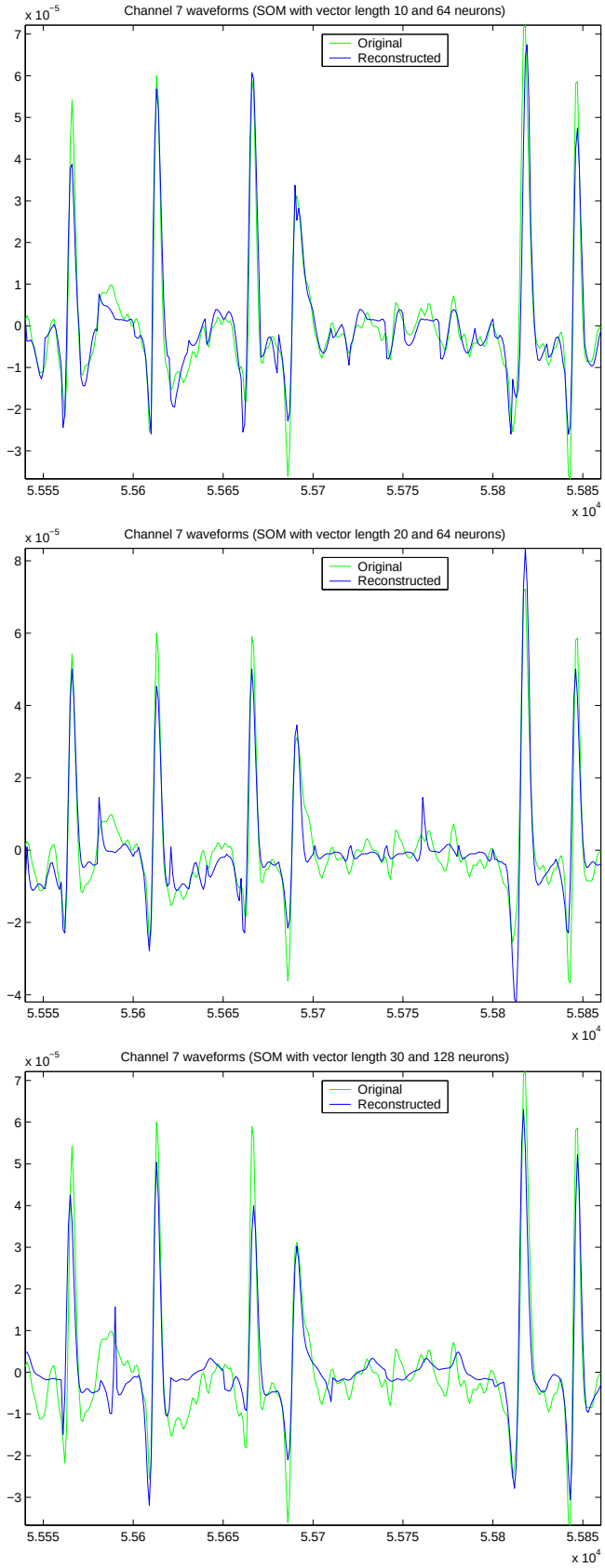

Fig. 2. Original and reconstructed waveforms at the decoder, for channel 7. A SOM with vector length of 10, 20 and 30 was used, with 64 PEs in the first two SOMs and 128 and the last.