Sparse Coding of Movement-Related Neural Activity

Marcello M. DiStasio
SUNY Downstate Medical Center
and Polytechnic Institute of NYU
Brooklyn, NY
Email: marcello.distasio@downstate.edu

Pratik Y. Chhatbar
MUSC Neurosciences
Charleston, SC
Email: chhatbar@musc.edu

Joseph T. Francis
SUNY Downstate Medical Center
Brooklyn, NY
Email: joseph.francis@downstate.edu

***Abstract*—Modern systems neuroscience benefits from the ability to record from and digitize a large amount of functional data from hundreds or even thousands of neurons. Understanding, transmitting, storing, and parsing information of such volume and complexity calls for methods of dimensionality reduction. One observation about neuronal activity in mammalian brains is that populations are sparsely active; that is, only a small subset of the whole ensemble is coactive at any moment. This property may be exploited to summarize information content succinctly. This paper tests the hypothesis that information contained in ensemble activity recorded from the primate motor cortex about limb movements is preserved when the activity is projected onto a sparse basis. Spiking rate data from neurons in the motor cortex of an awake behaving macaque monkey was compressed using a sparse autoencoder network, and classifications of movement directions were made in the compressed space. Classifier performance is shown to be similar when using either compressed (sparsened) or uncompressed neural activity, demonstrating the potential use of the sparse autoencoder as an unsupervised compression algorithm for low power/low bandwidth wireless transmission of neural ensemble data.**

I. Introduction

Sparse coding has been proposed as a method by which a relatively small number of concurrently active processing elements in the brain can efficiently encode large amounts of information. Theoretical studies of associative memory structures indicate that sparsity in activations minimizes interference between encoded patterns [1] [2] and allows for increased specificity in representations of information that has structure known *a priori* [3] [4]. Calculations of energy constraints imposed by the relatively high cost of neural firing have led to the estimate that only 3% of cortical neurons are actively spiking at any given time [5]. In recordings of neural ensembles, these considerations all motivate the use of decoding algorithms that employ similar criteria, favoring schemes that minimize assumed concurrent activity among disparate neuronal inputs. Here we present results from application of an unsupervised sparse autoencoder (SA) algorithm that summarizes ensemble activity in just a few bases that are constrained to be minimally coactive. By training the autoencoder, a summary of neural activity is generated that identifies groups of neurons that tend to become active at different times. The question we aim to answer in this paper is whether such a separation (and dimensionality reduction) preserves information about limb movement known to be contained in the neuronal firing when each neuron is considered independently. If so, the sparse autoencoder may find use as a dimensionality reducing compression mechanism for encoding activity patterns recorded from large populations of neurons. Examination of activity in such bases constrained to be sparsely active is also illuminating of brain function since downstream brain regions may decode activity in a sparse manner.

II. Methods

A. Behavioral Task and Neural Recordings

The data for this study was collected from an awake behaving bonnet macaque monkey during performance of a manual center-out reaching task. The monkey controlled a cursor on a computer monitor by planar movements of its right arm inside a robotic exoskeleton (Kinarm, BKIN Technologies, Kingston ON). The monkey was required to hold the cursor within a fixation target until appearance of a reach target located at one of eight radially arranged positions, at which time it was required to move to the reach target in order to receive a juice reward. The movements were required to be completed within 4s or the trial was aborted. Primary motor cortex activity during this type of movement is known to be predictive of movement direction [6] [7]. Movement trajectories were recorded by the exoskeleton.

Recordings of neural activity were made using a surgically implanted microelectrode array (Blackrock Microsystems, Salt Lake City UT) in the primary motor cortex (M1) ipsilateral to the arms used to perform the task [8]. The array was a 10x10 platinum electrode grid with $450\mu\mathrm{m}$ interelectrode distance at tip and 1.5 mm shank length. During recording sessions, amplification and preprocessing were performed with a multi-acquisition processing system (Plexon, Inc., Dallas TX). Signals from all array channels were amplified, band pass filtered (170 Hz to 8 KHz), sampled at 40 KHz, thresholded, and single units were sorted based on their waveforms using principal-component clustering. Spike times thus identified were saved for subsequent analysis.

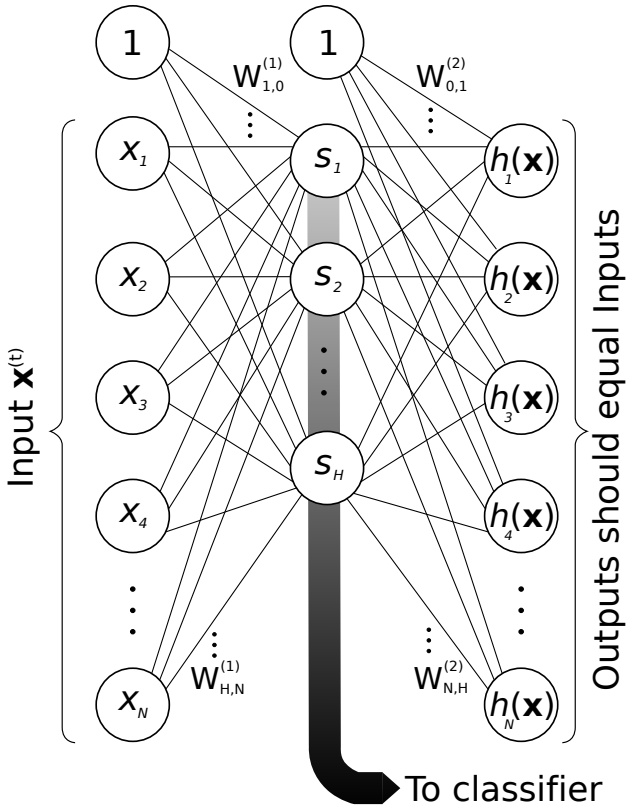

Fig. 1. **Sparse autoencoder network architecture.** The input for a time bin t is a vector $\mathbf{x}^{(t)} \in \mathbf{R}^{N+1}$, where N is the number of neurons, and $x_0 = 1$. The first set of weights $W^{(1)}_{j,i}$ specifies the connection strength from inputs $i \in 0...N$ to hidden units $j \in 1...H$, where H is the prescribed number of hidden units in the network. The sigmoid non-linearity results in hidden unit activations $\mathbf{S} \in \mathbf{R}^{H+1}$. These, which include another bias term $s_0 = 1$, constitute the inputs to the output layer, scaled by a second set of weights $W^{(2)}_{j,i}$ from hidden units $i \in 0...H$ to output units $j \in 1...N$. The output layer activations are computed by application of the sigmoid again, and are thus in the range $(0, 1)$. During training, the error signal is a function of the difference between input $\mathbf{x}^{(t)}$ and output $h(\mathbf{x}^{(t)})$, a rule which requires no supervised teaching signal. During subsequent classification, the activations of the hidden layer units are the independent variables of interest.

B. Sparse Autoencoder

For N dimensional input, with desired compression to H dimensions, the sparse autoencoder network [9] [10] is formulated as a feedforward, fully connected, single-hidden-layer perceptron network with an input layer of $N + 1$ units, a hidden layer of $H + 1$ units (both layers contain a bias term set to 1), and an output layer of N units (Figure 1). Sigmoid $(1 + e^{-x})^{-1}$ nonlinearities are used as squashing functions. The input vectors $\mathbf{x}^{(t)} \in \mathbf{R}^{N+1}$ are the binned spike counts from all neurons at a single time step (along with a bias term x_0 set to 1). During training, for all timesteps t, the network output $h_W(\mathbf{x}^{(t)}) \in \mathbf{R}^N$ for the current weight matrix $W = \{W^{(1)}_{j,i}, W^{(2)}_{j,i}\}$ is compared against the input, with error taken as the Euclidean distance between them. Thus the goal of the network is to reproduce the input vector at the output after passing it through a restricted hidden layer. To make the network learn such an identity function, a cost function $J(W, \mathbf{x}) = \frac{1}{2} \parallel h_W(x) - x \parallel^2$ is applied to each training example. For all training data (u time steps) taken at once, the overall cost function is expressed as

$$J(W) = \left[\frac{1}{u}\sum_{t=1}^{u}\left(\frac{1}{2} \parallel h_W(x^{(t)}) - x^{(t)} \parallel^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{2}\sum_{i=0}^{n_l}\sum_{j=1}^{n_{l+1}}\left(W^{(l)}_{j,i}\right)^2 \quad (1)$$

where the second term is a regularization penalty, weighted by parameter λ, introduced to prevent overfitting. n_l is the number of units in layer l. In order to enforce the sparsity criterion that the input layer units should be inactive most of the time, a sparsity penalty on the hidden units is added to the cost function. This penalty was computed as the Kullback-Leibler divergence between a Bernoulli random variable with mean ρ (the desired average activation of the hidden units over training inputs) and another with mean $\hat{\rho}_j$ (the observed average activation of hidden unit j over training inputs). This results in a total cost function

$$J(W)_{sparse} = J(W) + \beta\sum_{j=1}^{s_2}\mathrm{KL}(\rho \parallel \hat{\rho}_j). \quad (2)$$

$\mathrm{KL}(\rho \parallel \hat{\rho}_j) = \rho\log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j}$ and β is a parameter controlling the contribution of the sparsity penalty to the total cost (set to 3 in all subsequent analyses). The network is trained using backpropagation slightly modified to include the sparsity penalty in equation (2). Thus for each training example a feedforward pass is made to compute the activation of hidden units ρ_j which are used in the KL divergence term in the cost function, and thus contribute to the gradient on the input weights $W^{(1)}_{j,i}$ for the hidden layer.

The cost function for each dataset was minimized using the L-BFGS algorithm [11], which is a hill-climbing method for finding a stationary point of a function (where the gradient is zero). The resulting values for the weight matrix for the hidden layer $W^{(1)}_{j,i}$ provide the information necessary to project the neural data onto the sparse basis.

Input Data: Spike times from all 274 analyzed neurons were binned at $25\mathrm{ms}$ resolution and smoothed with causal $200\mathrm{ms}$ boxcar smoothing windows and for the entire recording period. The spike rates for all neurons in bin t, normalized to the range $(0, 1)$ for the whole recording, form an input vector $\mathbf{x}^{(t)} \in \mathbf{R}^N$, where N is the number of neurons. All timesteps from the recording session were used as training data for the sparse autoencoder network, forming a training set $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(t_{max})}\}$, where t_{max} is the last time bin in the recording.

C. Classification

The class labels for movement analysis were $y \in \{1...8\}$, corresponding to each of the eight radial directions (equally spaced) at which targets appeared. Neural data from windows of length $L_{window} = 900\mathrm{ms}$ surrounding successful reaching

movements (-200ms to 700ms from movement onset; average total reach time observed was 790ms) was binned with windows of length $L_{bin} = 25$ms as above. This resulted in a collection of $L_{window}/L_{bin} = 36$ vectors $\{\mathbf{x}^{(t)}\} \in \mathbf{R}^N$, each representing the neural activity for one time step. These were projected onto the sparse basis by passing them through the first layer of the autoencoder network:

$$\mathbf{s}_t = (1 + e^a)^{-1} \in \mathbf{R}^H, \quad a = W^{(1)}_{j,i}\mathbf{x}^{(t)} + b \tag{3}$$

The collection of the 36 sparsened vectors for all time bins in one movement window $\mathbf{S}^{(m)} = \{\mathbf{s}_t\}, \quad t \in \{1...36\}$, along with the associated class label $y^{(m)}$ form a single training example for a multinomial logistic regression classifier. Thus the classifier uses examples of the form $\left(\mathbf{S}^{(m)}, y^{(m)}\right), m \in \{1....M\}$, where M is the number of successful movements made during a recording session.

The classifier used was softmax multinomial logistic regression using nominal response variables with L^2-regularization [12]; the regularization parameter was set to 0.001 (determined by an independent set of cross-validation tests).

D. Validation

We used data from 137 successful reaching movements in a cross validation scheme to ensure that the classifier was not overfitting to the training data. During each round of cross validation, 20% of the reaching movements were set aside as "test" data, and the remaining 80% were used to train the logistic regression model. This separation does not apply to the sparse autoencoder algorithm, which was trained on all neural data recorded throughout the experiment. Classification performance measures reported in this paper refer to misclassification rates for test data only, to which the trained classifier was naive (for both the sparsened and unsparsened case).

III. Results

A. Sparse Autoencoder Performance

The quality of the projection of the neural data onto the reduced-dimension sparse basis is evaluated based on the ability of the network to reproduce the neural data at the output. Fidelity of reconstruction at the output ensures that all ensemble information has been preserved in the compression. We computed the mean across all time steps of the L^2-norm difference between the sparse autoencoder input and output. The mean output error for sparse autoencoder networks with varying numbers of hidden layer units is shown in Fig. 2.

The low error in reconstruction ($<$10% for all numbers of hidden units) suggests that the neural activity being encoded is well summarized by activations of only a few bases that are constrained to be rarely coactive. These results are for networks trained with sparsity parameter $\rho = 0.1$; changing this parameter (i.e. tightening or relaxing the sparsity constraint) affected the reconstruction error slightly ($\pm$0.02 mean error over all values from 0.001 to 0.5) but had no effect on the

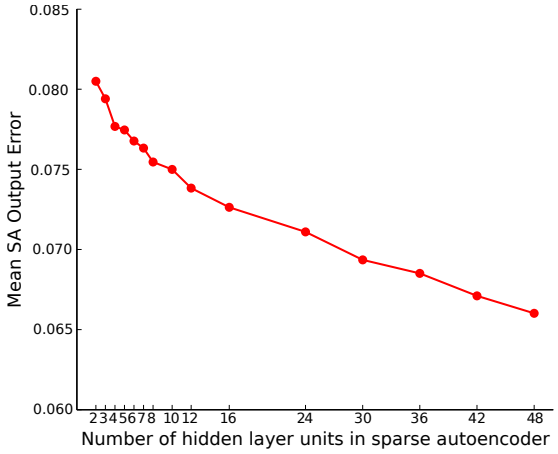

Fig. 2. **Sparse autoencoder reconstruction performance** During training of the sparse autoencoder network, a record of the L^2-norm difference between network output and training signal ($|h_W(\mathbf{x}^{(t)}) - \mathbf{x}^{(t)}|$) was kept. The mean value of this error statistic over all over all time samples in the recording $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(u)}\}$ is plotted for various numbers of hidden layer units.

movement direction classification error (see section III-B; data not shown) and so was not pursued further.

Inspection of the average activation of the hidden units (bases) in peri-movement windows ($\mathbf{S}^{(m)}$) shows that activity for a given unit differs across movement directions (Fig. 3). It is also evident that different bases capture features at different phases of the movement. Activity in basis 8, for example, is high for all movement directions during the pre-movement and late-movement phases, but varies conspicuously during the period immediately following movement onset. Such differences provide good features for classification.

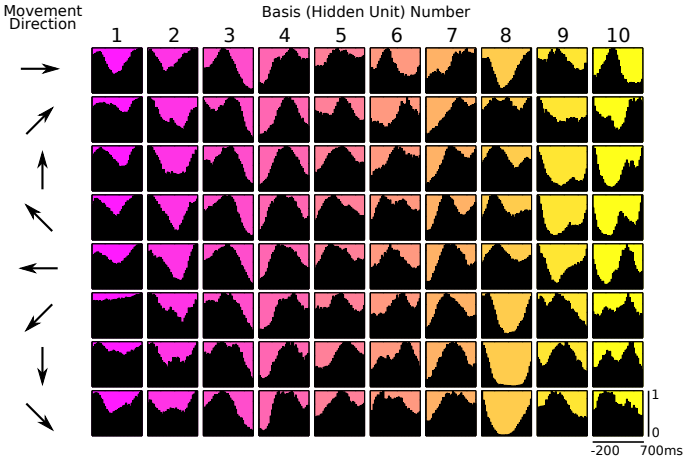

Fig. 3. **Averaged activities in sparse bases for eight movement directions**. For each movement direction, the average output of each of the 10 hidden units (shown in columns) across all examples of that movement were computed for 36 time bins surrounding the movement, from 200ms before to 700ms after onset. The direction for each row is indicated by the arrow in the left column. Bar heights are scaled to the maximum value for all bases for any direction; range (0,1).

B. Classifier Performance

Does the activity in such a reduced basis still contain information about movement direction, or has it been destroyed by the dimensionality reduction and sparsening criteria? To address this, the neural activity projected onto various set numbers of sparse bases were used as input for classification of movement directions. Performance on this problem was quantified in terms of fraction of reach directions in the test data set misclassified on each round of cross validation. The results are shown in Fig. 4. As a baseline, the multinomial logistic regression classification was performed on identically preprocessed but unsparsened neural activity. The mean error rate for the classification of sparsened data was found to be similar to that for unsparsened data when the the sparse autoencoder was equipped with a sufficient number of hidden units (i.e. dimensions). For the data set used, the minimum number of bases needed for commensurate performance proved to be 10. It is notable, however, that even with as few a two bases, the performance of the classifier on test data remained well above chance level, which was established by classification on a dataset where the labels $y^{(m)}$ for all movements were shuffled randomly. This suggests that even aggressive dimensionality reduction by the sparse autoencoder preserved much of the information needed to infer movement direction from neural activity.

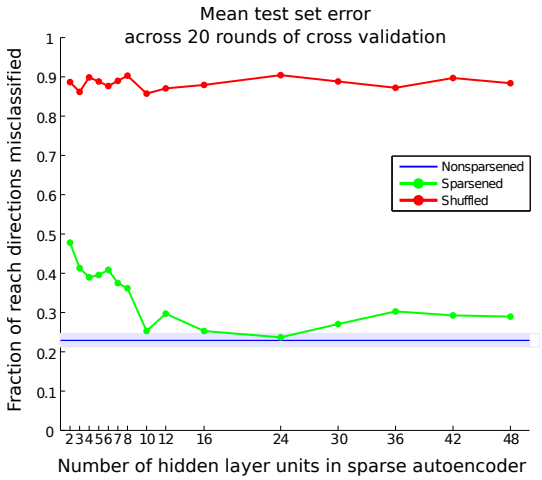

Fig. 4. **Classifier performance on test data** Multinomial logistic regression was applied to the neural activity projected onto the sparse basis after SA training (activity in H bases x 36 time steps for each example) for SA networks with different numbers of hidden units (X axis). The mean fraction of examples in the test sets of 20 rounds of cross validation that were misclassified is indicated by the green dots. For comparison, training and classification were performed on the same sparsened dataset but with the labels shuffled. The test set misclassification rate is indicated by the red dots. The blue line and band shows the mean±SD misclassification rate for the multinomial logistic regression applied to unsparsened neural data.

Finally, it was noted that the mean percentage of recorded neurons coactive within a bin throughout the recorded file was $12.6 \pm 4.5\%$. Such a small proportion is consistent with the idea that networks of coactive cells are sparsely connected.

IV. Conclusion

We emphasize that the results presented here do not conclusively establish that a sparse code is employed in the motor cortex (though they are consistent with this hypothesis), but rather that a downstream decoder which is constrained to be sparsely active is able to capture the same amount of information about movement direction as the raw neural activity. The fact that the sparse autoencoder preserves information about movement direction does suggest that neurons downstream from the motor cortex engaged in maintenance of internal forward models of movements (in striatum or cerebellum, for example) could successfully capture movement information with a sparse code. The hidden unit activations may be interpreted as summaries of activity of subassemblies of neurons within the whole recorded population. Since relatively few of these summaries are required to reconstruct the neural activity, we can conclude that there is statistical regularity in the identities of the subassemblies, which can be exploited by downstream decoders.

Note that the sparse autoencoder is applied here to quasi-static binned data. The dynamics of the neuronal firing at fine time scales are not taken into account. An important extension of this work would be to apply similar methods to a dynamical system model in order to identify temporal patterns in ensemble spiking that provide useful bases for efficient coding of time-varying motor control signals.

There is a practical use for the dimensionality reduction presented here as well. Despite the rich information about neural coding that neuroscientific preparations uncover, translation into practical use in the clinical or home setting has been slow partly due to the obtrusiveness of implants. Wireless transmission of brain-derived information would further the use of low-profile devices that could be implanted more safely and permanently. This requires an economy of power and bandwidth, both of which are facilitated by the sparse autoencoder, an unsupervised algorithm that could be implemented on board a fully implantable microprocessor.

Acknowledgments

This work was supported in part by the Joint Graduate Program in Biomedical Engineering at SUNY Downstate/NYU Polytechnic and DARPA REPAIR project N66001-10-C-2008.

References

[1] D. Willshaw, O. Buneman, and H. Longuet-Higgins, "Non-holographic associative memory," *Nature*, vol. 222, pp. 960–2, 1969.
[2] D. Field, "What is the goal of sensory coding?" *Neural Computation*, vol. 6, no. 4, pp. 559–601, 1994.
[3] H. Barlow, "Single units and sensation: a neuron doctrine for perceptual psychology?" *Perception*, vol. 1, pp. 371–94, 1972.
[4] E. Simoncelli and B. Olshausen, "Natural image statistics and neural representation," *Annual Rev Neurosci*, vol. 24, pp. 1193–1215, 2001.
[5] P. Lennie, "The cost of cortical computation," *Current Biology*, vol. 13, pp. 493–7, 2003.
[6] A. P. Georgopoulos, A. B. Schwartz, and R. E. Kettner, "Neuronal population coding of movement direction," *Science*, vol. 233, pp. 1461–9, 1986.

[7] K. Ganguly, L. Secundo, G. Ranade, A. Orsborn, E. F. Chang, D. F. Dimitrov, J. D. Wallis, N. M. Barbaro, R. T. Knight, and J. M. Carmena, "Cortical representation of ipsilateral arm movements in monkey and man," *J Neuroscience*, vol. 29, no. 41, pp. 12 948–56, 2009.

[8] P. Chhatbar, L. von Kraus, M. Semework, and J. Francis, "A bio-friendly and economical technique for chronic implantation of multiple microelectrode arrays," *J Neuroscie Methods*, vol. 188, no. 2, pp. 187–94, 2010.

[9] A. Ng. (2010, Aug) Cs294a lecture notes: Sparse autoencoder. [Online]. Available: www.stanford.edu/class/archive/cs/cs294a/cs294a.1104/sparseAutoencoder.pdf

[10] A. Coates, H. Lee, and A. Ng, "An analysis of single-layer networks in unsupervised feature learning," *JMLR Workshop and Conference Proceedings 14th International Conference on AISTATS*, vol. 15, pp. 215–223, 2011.

[11] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. New York: Wiley and Sons, 1987.

[12] A. J. Dobson and A. G. Barnett, *An Introduction to Generalized Linear Models*, 3rd ed. Boca Raton, FL: Chapman and Hall/CRC Press, 2008.