Neuron Detection and Decoding in Fluorescence Microscopy

Tony Hyun Kim
kimth@stanford.edu

Lacey Kitch
ljkitch@stanford.edu

1 Introduction

A central goal of neuroscience is to explain the observable actions of a behaving animal by the activity of neurons in the brain. To accomplish this goal, scientists must: (1) record the activity of individual neurons during behavior; and, (2) correlate the measured neural activity to animal behavior. The current project explores machine learning methods to facilitate both tasks.

One method for recording neuronal activity is fluorescence microscopy, in which neurons in the brain express a fluorescent marker whose emission is modulated by the "firing" of the neuron. Our work is based on the integrated fluorescence microscope of the Schnitzer group [GBC$^+$11] which enables the acquisition of activity from large numbers of neurons in freely-behaving mice and rats. As shown in Fig. 1(a), the fluorescence microscope is surgically attached to the skull of a laboratory mouse. Fig. 1(b) shows a typical microscope image with individually outlined neurons. In a prototypical experiment, the animal is permitted to wander about its environment, while the microscope images the hippocampus – a region of the brain associated with spatial representation.

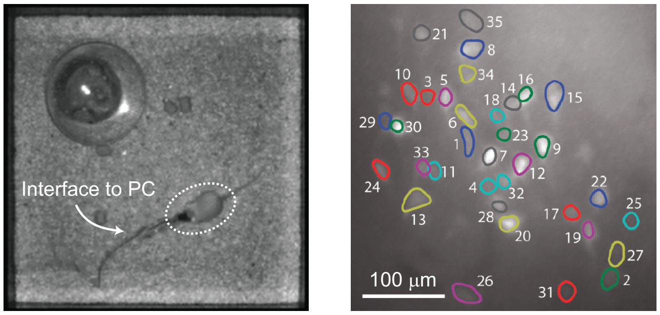

Figure 1: Fluorescence microscopy of a freely-moving animal. (Left) A fluorescence microscope is surgically attached to the skull of a lab mouse, which is free to move about in its environment. The microscope's interface to the PC is clearly visible. (Right) A typical view through the microscope [GBC$^+$11]. Individual neurons are outlined and labeled.

Our project is organized into two parts:

- In Section 2, we present a supervised learning algorithm for the automatic classification of neurons from fluorescence microscopy data.
- In Section 3, we develop a supervised algorithm to predict (*i.e.* "decode," in the parlance of neuroscience) the position of the mouse in its environment, based on hippocampal neural activity.

2 Neuron classification

The first part of our project is to automate the identification of neurons in the calcium imaging data. Essentially, the data from the integrated microscope is a 3–5 minute-long movie that shows a 500 μm × 500 μm section of the brain. The desired machine learning algorithm will identify which pixels of the microscope's field of view correspond to neurons, in a further automation of the data-processing pipeline published in Ref. [MNS09].

The microscope movie is preprocessed as follows. First, the dimensionality of the data is compressed and noise is reduced by throwing out low-variance components resulting from PCA. Second, ICA is performed to identify independently varying pixels within the movie's frame. Each resulting Independent Component (IC) is a matrix which gives a weight to each pixel in the field of view, *i.e.* a spatial filter. The filter is then applied frame-by-frame to the original movie to extract a temporal fluorescence/activity trace. The final preprocessed output consists of (spatial filter, activity trace) pairs as in Fig. 2 that must

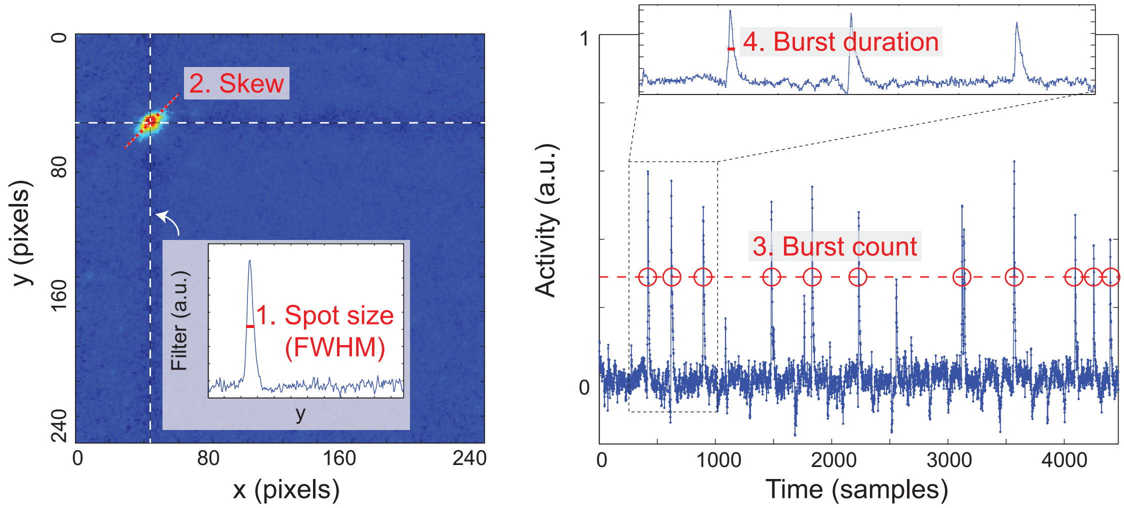

Figure 2: Extraction of the feature vector from a single IC (spatial filter, temporal trace) pair that was classified by the human operator to be a valid neuron. (Left) From the spatial filter, we compute the spot size and the skew of the region of interest. (Right) From the temporal activity trace, we count the number of fluorescence bursts and the average duration of each burst.

be classified as valid (or invalid) neurons. The classification step has hitherto been performed manually by a human operator. Our machine learning algorithm seeks to automate this step; we do so by leveraging the existing database of thousands of human-classified examples.

2.1 Feature vector definition

While it is in principle possible to feed the full spatial filter and the temporal trace into the learning algorithm, we have opted instead to extract a small number of features from each IC pair. Our feature definitions are inspired by the intuition of the human classifier, and are as follows:

1. *Spot size*. From the spatial filter, we estimate the size (in pixels) of the region of interest. As shown in the inset of Fig. 2(a), the region of interest may be determined by using a criterion such as the full-width half-maximum (FWHM).
2. *Skew*. We also consider the nonconcentricity of the spatial filter, as shown in Fig. 2(a). The skew may be measured, for instance, by interpreting the spatial filter as a probability distribution and calculating the xy-correlation.
3. *Burst count*. Neuronal activity consists of short-duration electrical impulses known as action potentials, and large numbers of action potentials generate "bursts" of fluorescence in the experimental set-up. Hence, the presence of bursts in the temporal trace is a natural indicator that the IC represents a neuron. As in Fig. 2(b), we count the number of bursts in the temporal signal by establishing a threshold (*e.g.* the standard deviation of the activity trace), and subsequently counting the number of positive crossings over the threshold.
4. *Duration of bursts*. We also consider the duration of each burst (*i.e.* the number of samples beyond the threshold divided by the burst count). This feature roughly corresponds to the "quality" of the recorded bursts. An abnormally short burst duration, for instance, may be caused by random optical scatterers in the microscope path.

2.2 Details of experimental data

Our training data consists of 17 datasets (each containing a few hundred labeled examples, of which 70% are positive on average), where a "dataset" represents one experiment on a single day. These datasets span 17 animals, one year of experimentation, and several experimental set-ups, thus allowing for a potentially-large variation in the calculated features. We are interested in an algorithm that performs robustly across datasets, *i.e.* achieving high accuracy on a test dataset that is a distinct from the training dataset. If our algorithm achieves such performance, it may perform neuron identification reliably across animals and across experimental trials.

2.3 Logistic regression

Our original classifier is based on logistic regression, using the standard formulation [Ng2011] and optimized with Newton's method. First, we trained the algorithm on *one* of our 17 datasets and tested "pairwise" on each of the others. The results demonstrated excellent performance ($> 95\%$) on some test/train pairs, good performance on most ($\sim 90\%$), but very poor performance (as low as 50%) on a few cases. The histogram is shown in Fig. 3(a).

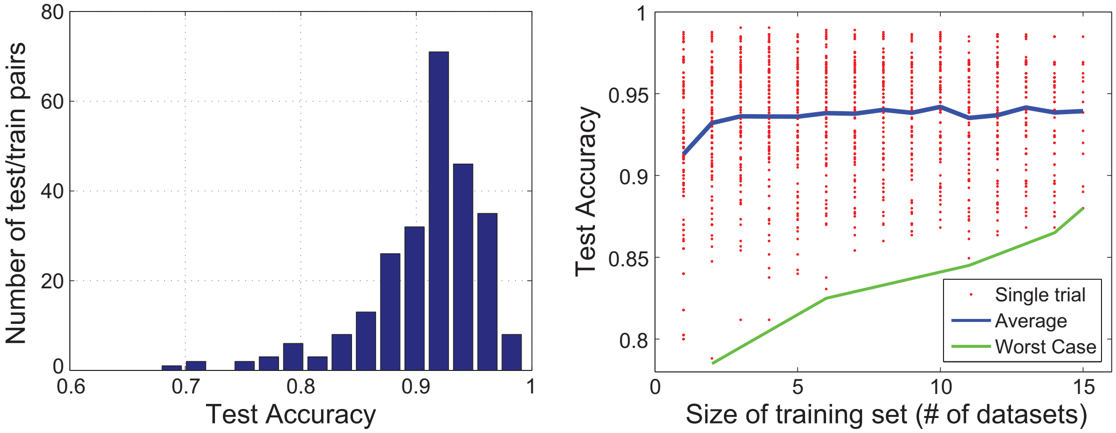

Figure 3: Summary of logistic regression performance. (Left) Histogram of pairwise test errors. (Right) k-fold cross-validation as the number of datasets in the training set is increased. Red dots indicate individual test errors; the blue line shows the mean k-fold accuracy; and the horizontal axis indicates the number of datasets used to train the algorithm (corresponding roughly to the number of training examples).

Next, we trained the logistic regression algorithm on multiple datasets, and then tested its predictions against the remaining datasets. Despite the potential variations between experimental trials, we found that the mean accuracy generally rose as a function of the size of training set and quickly saturated around 94% (after about 1000 total examples) as shown in Fig. 3(b). The worst-case performance, however, continued to rise as the size of the training set was further increased. This continued scaling is fortunate, since we would like to guarantee a lower-bound performance on any new dataset to be classified.

2.4 Support vector machine (SVM)

Motivated by our observation that correct IC classification has a slightly nonlinear relationship to the features, we performed classification using an SVM with the Gaussian kernel. We compared the k-fold cross-validation performance ($k = 10$) of SVM (using the LibSVM package) against logistic regression, both using all 17 datasets. We performed a grid search over the regularization parameter C and the Gaussian kernel prefactor γ to minimize the cross-validation error, and found a maximum SVM accuracy of $\sim 95\%$. This performance is comparable to the $\sim 94\%$ accuracy of logistic regression. Thus, due to its simplicity of implementation, logistic regression remains our preferred algorithm.

2.5 Discussion of results

The accuracy of our machine learning algorithm, when compared directly to the labels of the human classifier, is 94%. Beyond this basic performance, we have performed manual inspection of the incorrectly-labeled examples in order to identify possible biases in the machine's (or the human's) classification. Particularly interesting is the case of false negatives, *i.e.* examples where the machine algorithm predicted "no neuron" while the human classifer specified "neuron." We were motivated to investigate this case after we were informed by the human classifier that she tended to be "generous" in granting the neuron label, and that she tended to de-emphasize information from the activity trace in her classification decision.

Indeed, upon closer inspection, we found that the false negatives generally possess significantly fewer fluorescence bursts in their activity trace. Over 75% of false negatives have fewer than half of the mean burst count of valid neurons; and, 30% have an insignificant number or no bursts. Hence, we believe that many of our false negatives originate from the bias of the human classifier, and the performance of logistic regression for correctly labeling neurons is greater than the reported 94%.

Finally, we have applied our algorithm for automatic neuron classification to other animals (*e.g.* rats) used in the Schnitzer laboratory and found comparable performance to our mouse results without having to re-code or even retrain the algorithm. Due to this demonstrated robustness and high performance, our algorithm will be used in the future by scientists in the Schnitzer lab for automated neuron identification.

3 Movement prediction

The second part of our project aims to utilize machine learning techniques to predict animal behavior based solely on neural data, in essence reading the animal's mind. In a prototypical experiment, a mouse runs on a linear track of approximately 80 cm in length (trajectory $x(j)$ of the mouse shown in top panel of Fig. 4), while the fluorescence microscope simultaneously records the activity in the mouse's hippocampus, a region of the brain associated with spatial representation. As shown in the

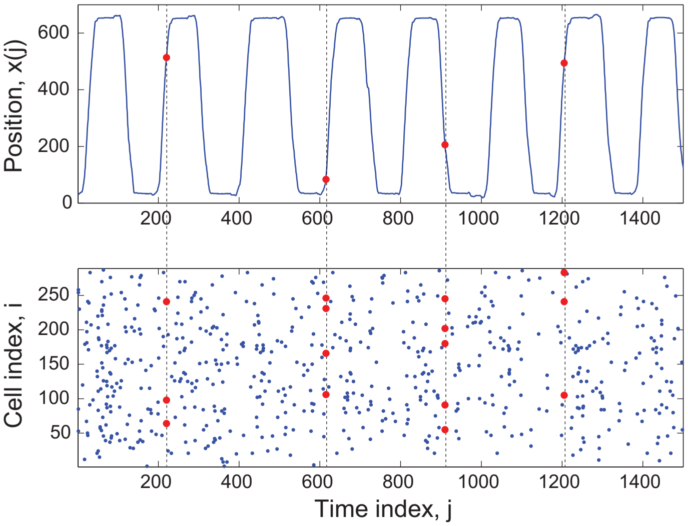

Figure 4: The temporal pairing of mouse position with neural activity. Dashed columns indicate the temporal pairing between the mouse's instantaneous position and the neural "burst vector." (Top) A mouse runs back and forth on a linear track of approximately 80 cm in length. (Bottom) Fluorescence activity is segmented into a binary matrix $B(i, j) = \{0, 1\}$ where $B(i, j) = 1$ (denoted by a blue dot) indicates the presence of a fluorescence burst in cell i at time index j.

bottom panel of Fig. 4, fluorescence activity is segmented into a binary matrix $B(i, j) = \{0, 1\}$ where $B(i, j) = 1$ indicates the presence of a fluorescence burst in cell i at time index j.

As shown by dashed columns of Fig. 4, the position of the mouse is temporally paired with a "burst vector" that indicates the small subset of concurrently active neurons. The movement prediction algorithm will be trained on a set of such (position, burst vector) pairs. The trained algorithm will then predict the position of the mouse based on its neural activity alone.

3.1 Feature vector definition

Naturally, the minimal set of features used for predicting position $x(j)$ is B_j, the burst vector at time j. In our exploration of movement prediction, however, we have found it important to consider:

1. *Inclusion of past/future burst vectors*. Empirically, we find that B_j alone is insufficient to achieve acceptable error in position prediction even for the training set. Hence, in addition to B_j, we allow the algorithm to utilize activity information from burst vectors in the vicinity of time j as in $\{B_{j-N}, \cdots, B_{j-1}, B_j, B_{j+1}, \cdots, B_{j+N}\}$. We parameterize the inclusion of such past and future burst vectors by N, the half-width of non-present vectors used in the feature definition. Of course, increasing the dimension of feature vector this way could overfit the training data; thus, we determine the optimal value of N with respect to test error.
2. *Burst vector smoothing*. As shown in Fig. 4, the nonzero entries of the burst matrix $B(i, j)$ are distributed rather sparsely. The sparseness is partly a consequence of the segmentation process, in which $B(i, j) = 1$ is assigned only at the time index corresponding to the peak of a fluorescence burst. However, as discussed in our previous work on neuron classification, the duration of a fluorescence burst typically extends many time samples. Hence, we convolve the burst matrix with a box-shaped "smoothing" filter of length $(2M+1)$, in order to account for the extended duration of each burst.

3.2 Details of experimental data

Each experiment is performed on a single mouse on a single day, and is separated into $5 - 6$ trials where each trial consists of a ~ 3 minute sequence. We compute test error as the k-fold cross validation error among the trials. Test error is cited in units of centimeters, which is the square root of the mean square error ($\sqrt{\text{MSE}}$) between the predicted and actual positions in real space. The full length of the track is 80 cm. Due to differences in neural coding between animals, one cannot not train on one animal's data and then test on another's. Thus, our algorithm is evaluated on several (five) independent sets of train/test data.

3.3 Naive Bayes prediction of mouse position

Our movement prediction algorithm is based on Naive Bayes. The left panel of Fig. 5 shows test error as a function of the number of past/future vectors (N) and the size of the smoothing filter (M). It is readily observed that past/future burst vectors

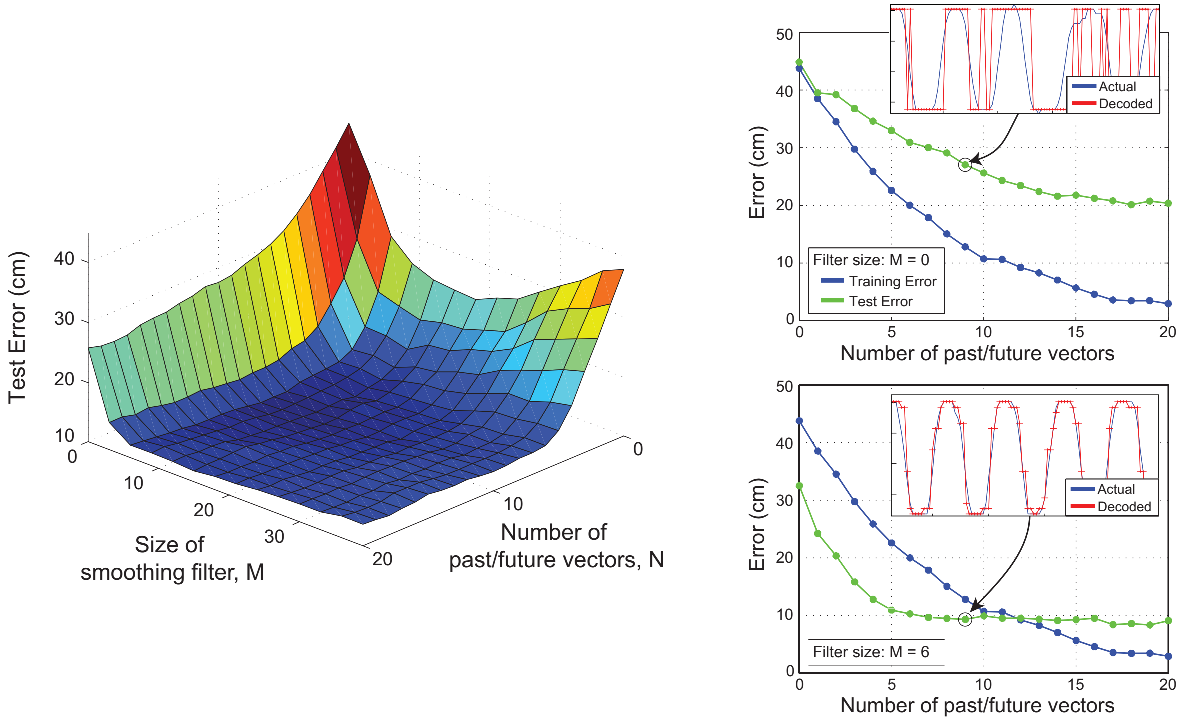

Figure 5: Results of the Naive Bayes movement prediction algorithm. (Left) One animal's 5-fold cross-validation test error as a function of number of past/future vectors (N) and the size of the smoothing filter (M). Optimal performance is obtained when $(N, M) = (11, 6)$ (Right) Test and training error as a function of N for two different values of M. Insets show the accuracy of the decoded trajectory.

or burst vector smoothing independently does not yield optimal performance. Careful inspection of the test error surface shows that optimal performance of $\sqrt{\text{MSE}} \approx 10$ cm is obtained when $(N, M)^* = (11, 6)$. Notably, the optimal size of the smoothing filter corresponds to the typical duration of fluorescence bursts.

The right panels of Fig. 5 show the scaling of both training and test error as a function of N for two different values of M. As previously noted, increasing N will eventually eliminate the training error (independently of M) as it permits overfitting of the training set. On the other hand, the limiting value of the test error is highly dependent on M. The corresponding insets show qualitatively the accuracy of the decoded trajectory.

3.4 Discussion of results

With Naive Bayes, we achieved a baseline position decoding performance that is comparable to state-of-the-art results using electrophysiological methods. This is exciting because optical data, as compared to electrical recordings, has a much lower temporal resolution and is, in general, a less direct indicator of neural spiking activity. For this reason, fluorescence-based decoding had not yet been demonstrated. Thus, we have set a new benchmark, and shown that the information content in our hippocampal imaging data is indeed sufficient for reconstructing spatial position very precisely.

Acknowledgments

T.H.K. and L.K. thank Laurie Burns for providing experimental data, for pre-processing the data, and for having manually classified thousands of neuron candidates, and Yaniv Ziv for performing the mouse experiments.

References

[GBC$^+$11] Kunal K. Ghosh, Laurie D. Burns, Eric D. Cocker, Axel Nimmerjahn, Yaniv Ziv, Abbas El Gamal, Mark J. Schnitzer. Miniaturized integration of a fluorescence microscope. *Nature Methods*, 8:871, 2011.

[MNS09] Eran A. Mukamel, Axel Nimmerjahn, Mark J. Schnitzer. Automated Analysis of Cellular Signals from Large-Scale Calcium Imaging Data. *Neuron*, 63:747-760, 2009.

[Ng2011] Andrew Y. Ng. CS 229 Lecture Notes 1.