An encoder-decoder approach for model-based spike inference

Timothy A. Machado[1,*,+] **and Liam Paninski**[1]

[1]Department of Statistics and Grossman Center for the Statistics of Mind, New York, NY, USA
[+]Current affiliation: Department of Bioengineering, Stanford University, Stanford, CA, USA
[*]tim.machado@gmail.com

1 Overview

In our submission to the Spikefinder competition, we had two main goals. First, we wanted to develop an approach that could take advantage of our strong prior knowledge of Ca^{2+} dynamics in neurons. Second, we wanted to construct a simple means of testing the relative importance of different features of fluorescence data in accurately decoding spikes. While model-based approaches have been used extensively before[1–3], we wanted an approach that could also easily incorporate nonlinear models, and enable tractable inference in models that depend on the long-term time history of a neuron's activity.

Our inference framework consists of two parts: a linear *encoding model* that takes in spikes and outputs simulated fluorescence traces (trained on paired spike train and fluorescence data), and a simple convolutional neural network to serve as a *decoding model* that outputs estimates of spikes given fluorescence observations and encoding model parameter estimates and which is trained on large datasets simulated from the encoder model. The advantage of this approach (which can be considered a form of data augmentation) is that we can train the decoder model to "saturation," by providing it as much training data as necessary to achieve good performance. Importantly, on linear-Gaussian simulated data, our neural network decoder performed comparably to OASIS[3], the state-of-the-art inference method for efficiently solving the spike inference problem under linear-Gaussian assumptions (though OASIS remains significantly faster than the neural network decoder at test time). Moving forward, both the encoder and decoder components presented here can be easily interchanged for more detailed models that might yield further improvements in performance in some experimental settings.

2 A Linear Encoder to Generate Fluorescence Data

Instead of directly using the Spikefinder datasets to train a decoding algorithm, we generated simulated training datasets consisting of 5,000 traces, each of length 3,000 timesteps. Each fluorescence trace was generated with the following second-order autoregressive model ($p = 2$):

$$c_t = \sum_{k=1}^{p} \gamma c_{t-k} + a s_t \tag{1}$$

$$f_t = c_t + b + \varepsilon_t \tag{2}$$

The parameterization of each trace generated using Equation 2 was random. The noise was modeled as $\varepsilon_t \sim N(0, \sigma_t^2)$. The jump size a of each spike was randomly sampled from a uniform distribution between 0.5 and 1.5. Each simulated spike train, s, was sampled from a homogeneous Poisson process with a mean firing rate between 0 and 2 Hz. The baseline drift component, b, was modeled by low-pass filtering white noise. Instead of a first-order autoregressive model for baseline drift[4], we found smoother low-frequency noise generated better performing decoders–and also qualitatively more closely match the type of slow drift seen in real fluorescence data. Of these parameters, we found that the baseline drift component was most important as incorporating it significantly improved inference quality. In contrast, the statistics of the simulated spike trains, as well as the randomized jump sizes following each spike, had a much smaller impact on decoder performance. This makes sense, as these parameters governing firing rate would be more significant in a nonlinear Ca^{2+} model.

The time constants of the autoregressive model, γ, and the scale of the noise, σ, were also sampled from random uniform distributions (such that a single decoder trained on these data could work well across multiple indicators and frame rates), but the precise parameterization was varied between model training sessions. However, in all cases, we found that a wide range of γ values could be learned by a single decoder model. For instance, we successfully trained decoder models across data with γ values spanning approximate decay times for fast OGB data recorded at 25 Hz, to slow GCaMP6S data recorded at 100 Hz. This is significant as it demonstrates that a single decoding model trained on simulated data can be used to analyze data produced by many different indicators, and many different acquisition rates. Similarly, wide ranges of σ values could be learned by single decoders. However, there was a slight performance enhancement seen by training single encoders on mostly low SNR or high SNR data to improve performance on low SNR and high SNR real data, respectively.

Finally, we found that in almost all cases, the second-order models (i.e. $p = 2$ in Equation 1) outperformed first-order models ($p = 1$). An exception to this could be seen with OGB datasets, as that sensor displays very rapid rise time kinetics following action potentials and has been previously shown to be especially well-described by first-order models.[1]

3 A Simple Neural Network Architecture for Decoding Spiking Activity

To estimate the most likely spike train underlying a given fluorescence trace, we built a convolutional neural network. During training, the network was presented with f as well as parameter estimates for σ and γ given by methods published in earlier work.[2] Because the Spikefinder data was upsampled from its native resolution and this introduced artifacts in the power spectrum of each fluorescence trace, we decimated each trace by a factor of 7-10 (depending on the approximate native time resolution of each dataset) before performing subsequent parameter fitting and analysis. The target of the network during training was the set of simulated spike trains, s, used to generate f using Equation 1.

We found a fairly simple architecture inspired by research into the construction of generative models for audio data to be effective.[5] In brief, the network consists of four 1D dilated convolutional layers containing 100 units, a filter size of 32, and rectified-linear (relu) nonlinearities. The first layer was dilated by a factor of 1, the second layer by 2, the third by 4 and the fourth by a factor of 8. Dropout (rate = 0.5) at each layer was also used. Finally, a fifth 1D convolutional layer with one unit, a filter size of one, and a relu nonlinearity was used to read-out a nonnegative estimate of s from each f vector provided.

This architecture contained about 950,000 parameters and could be trained on a simulated dataset of 5,000 traces in about 20 minutes (over 20 epochs) using the Google ML Engine. We found that a single model trained on simulated data that spanned a wide range of σ and γ values performed well, but that an ensemble of four models, each trained on a slightly different simulated dataset, worked better–as some decoders tended to work better or worse on each Spikefinder dataset. For our submission, we chose the decoding model that worked best for each dataset to use as our submission. For dataset 5, which had high firing rates, we found that convolving the results with a small Gaussian kernel resulted in a modest improvement to our inference quality.

4 Discussion

Our experience with the competition was productive in that it yielded a new approach for spike inference, as well as a few lessons and ideas that will be useful moving forward.

We think the contest was a valuable exercise overall, but we have a few ideas for how it could be improved. First, the data supplied often contained contamination from adjacent neurons or other signals that resembled action potentials that were not present in the ground truth. In addition, the process of resampling each dataset to a fixed frame rate simplified the contest, but also introduced artifacts into the data that are not typically seen in a real experimental setting. To remedy these issues, perhaps the next competition can be conducted on entirely unprocessed image patches instead of upsampled univariate traces.

In terms of the results of the contest, we find it interesting that linear-Gaussian model-based methods still remain fairly competitive versus nonlinear approaches–even on GCaMP6S data exhibiting significant nonlinearities. We experimented with training nonlinear encoding models but obtained inconsistent improvements; this would be a worthy direction for further research (and perhaps other entries in the competition obtained better results with nonlinear models than we did).

More generally, this raises the question of what the actual performance bounds of an ideal decoder are. Certainly the results of the contest have revealed some insight here: there is considerable dataset-dependent variance in the performance of each algorithm, and many different approaches were tried. But how can we evaluate performance in the case where, as is usually the case, ground truth information is not available? One potential approach, enabled by our framework, is to construct a neuron- or dataset-specific encoder model. Then, a decoder model can be trained to saturation on synthetic data matched to the neuron or dataset of interest. Then we can quantify the performance of this highly-trained decoder on simulated data drawn from the encoder. This process may be valuable for computing a neuron- or dataset-dependent approximate “bootstrap ceiling” on achievable spike inference performance (assuming the estimated encoding model provides an accurate summary of the fluorescent data generation process).

References

1. Vogelstein, J. T. *et al.* Fast nonnegative deconvolution for spike train inference from population calcium imaging. *J. Neurophysiol.* **104**, 3691–3704 (2010).

2. Pnevmatikakis, E. A. *et al.* Simultaneous denoising, deconvolution, and demixing of calcium imaging data. *Neuron* **89**, 285–299 (2016).

3. Friedrich, J., Zhou, P. & Paninski, L. Fast online deconvolution of calcium imaging data. *PLOS Comput. Biol.* **13**, e1005423 (2017).

4. Deneux, T. *et al.* Accurate spike estimation from noisy calcium signals for ultrafast three-dimensional imaging of large neuronal populations in vivo. *Nat. Commun.* **7** (2016).

5. van den Oord, A. *et al.* Wavenet: A generative model for raw audio. *CoRR abs/1609.03499* (2016).