Robust Machine Learning-Enabled Spike Inference in Neuronal Voltage Imaging Data

Vishnu Mangipudi, Ethan Song, Charles Zhang

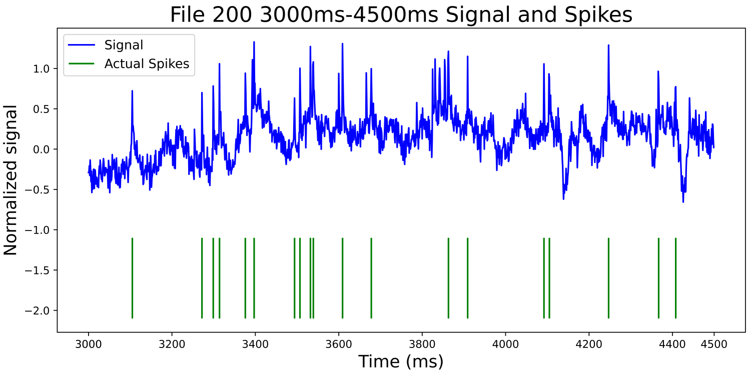
MIT PRIMES-USA

(Mentored by Dr. Lu Lu)

MIT PRIMES October Conference
18 October 2025

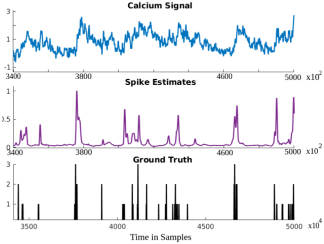
Motivation

- Interpreting neuronal signal data is critical to understanding physiological responses to stimuli of interest.
- Normalized signal data displays low signal-to-noise ratio and signal mean drift, hindering predictive ability.
- Spike sparsity further complicates model trainability.
- Spike detection is difficult but very important.



Existing Methods

- Thresholding:
  - Denoises normalized signal data via Gaussian smoothing;
  - Uses an empirically determined threshold to identify spikes.
- CaImAn (Giovannucci et al., 2019):
  - Applies CNMF and OASIS deconvolution to isolate spike events real-time;
  - Uses calcium fluorescence data instead of direct voltage measurements.
- VolPy (Cai et al., 2021):
  - Extends CaImAn's methods to voltage signal measurements;
  - Develops a standard pipeline for spike detection from voltage signaling movies.
- S2S (Sebastian et al., 2021):
  - Applies source separation methodologies to directly map raw fluorescence or voltage signals to spike trains.



Feed-forward Neural Networks

- A **feed-forward neural network** is a function $\mathcal{N}$ on an *input vector* x with a set of parameters θ called *weights* and *biases*.
- Formally,

Definition

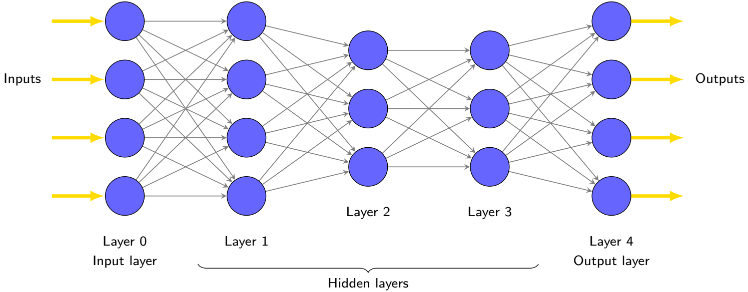
Consider L affine transformations of the form $T^\ell(x) = \mathbf{W}^\ell x + \mathbf{b}^\ell$ for $1 \le \ell \le L$ and some nonlinear *activation function* σ. A **feed-forward neural network** is defined as the function

$$\mathcal{N}(x;\theta) = T^L \circ (\sigma \circ T^{L-1}) \circ (\sigma \circ T^{L-2}) \circ \cdots \circ (\sigma \circ T^1)(x),$$

where $\theta = \{\mathbf{W}^\ell, \mathbf{b}^\ell\}_{\ell=1}^L$ are the network parameters. We call $\mathbf{W}^\ell$ and $\mathbf{b}^\ell$ the *weight matrix* and *bias vector* of the ℓ-th layer, respectively.

Feed-forward Neural Networks

- The network parameters θ are learnable through backpropagation.



Convolutional Neural Networks

- CNNs are similar to FFNNs, except that they allow for immediate contextual information to also be considered.
- Formally,

Definition

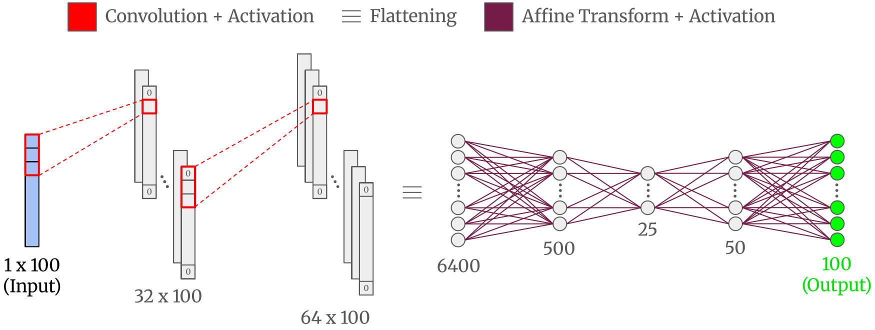
Consider L convolutional transformations of the form $C^\ell(x) = \mathbf{W}^\ell * x + \mathbf{b}^\ell$ for $1 \leq \ell \leq L$ where $*$ denotes discrete convolution, $\mathbf{W}^\ell$ is a learnable *kernel tensor*, and $\mathbf{b}^\ell$ is a learnable *bias vector*. Given a nonlinear activation function σ, a **convolutional neural network (CNN)** is defined as

$$\mathcal{C}(x;\theta) = C^L \circ \sigma \circ C^{L-1} \circ \sigma \circ \cdots \circ \sigma \circ C^1(x),$$

where $\theta = \{\mathbf{W}^\ell, \mathbf{b}^\ell\}_{\ell=1}^{L}$ are the parameters.

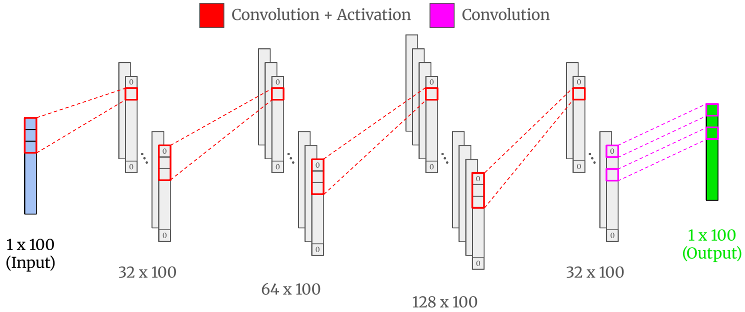
Convolutional Neural Networks

- We typically connect the output of the convolutional layers to a feed forward neural network to generate a result.



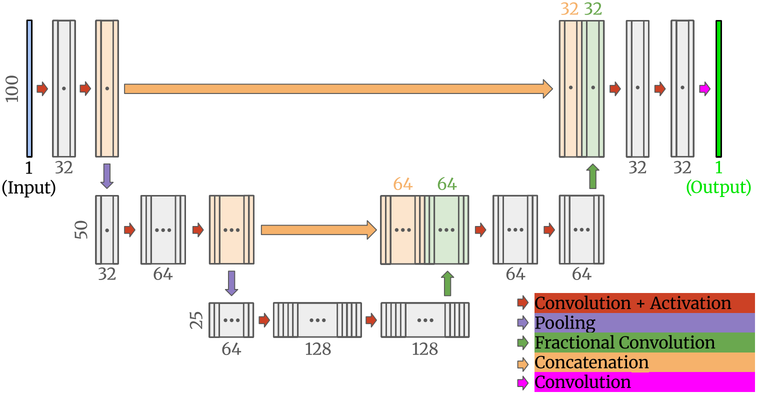
Fully Convolutional Networks

- We exclude the final feed-forward layer of our CNN, creating a **Fully Convolutional Network (FCN)**.
  - Preserves temporal resolution in constructing a 1-1 mapping;
  - Less dense layers → faster, less memory-intensive training/validation.



U-Nets

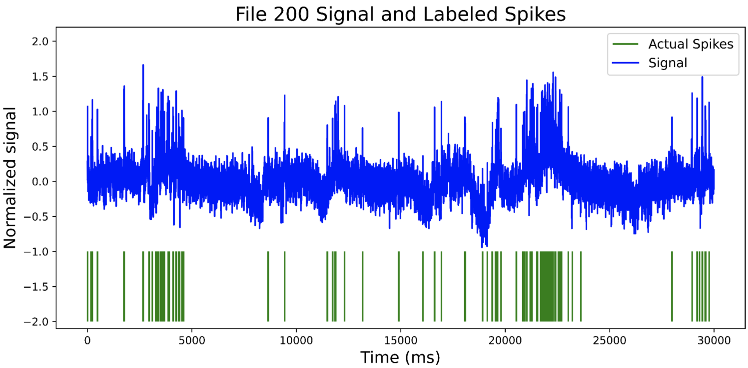
- To further enhance contextualization and localization of spikes, we include sequential downsampling and upsampling operations in the form of a **U-Net**.



Our Data

461 three-channel files containing:

- raw normalized voltages (0–29990 ms, 1 ms resolution);
- semi-manually identified spike locations (0/1).
- total number of semi-manually identified spikes

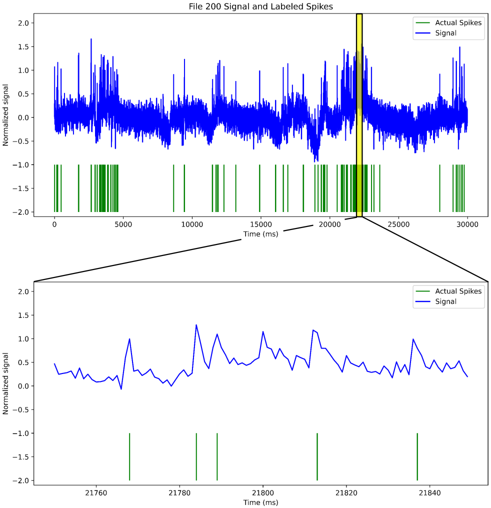


Data Windowing

A naive bijection mapping each data file to a respective array of predicted spike times is not ideal.

- Voltage signal measurements are prone to noise and signal drift;
- Spikes are local events;
- Large mappings are resource-intensive, features become hard to learn.

We instead focus on 100 ms windows of the entire 29990 ms measurement period.

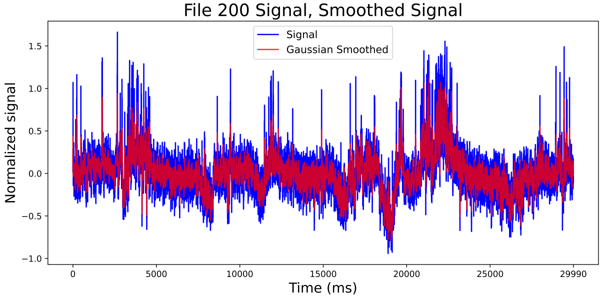


Data → Data Loading Data Windowing → Preprocessing Data Channels → Model → Postprocessing Model Results

Preprocessing Data Channels

We calculate several other parallel feature channels to provide more information to the model.

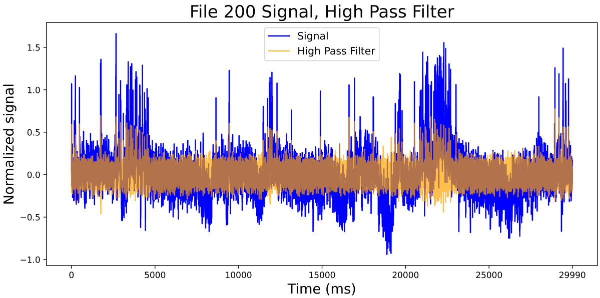
- Smoothed Signal



Preprocessing Data Channels

We calculate several other parallel feature channels to provide more information to the model.

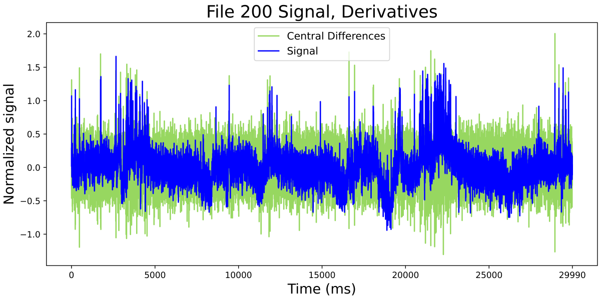
- Smoothed Signal
- High Pass Filter



Preprocessing Data Channels

We calculate several other parallel feature channels to provide more information to the model.

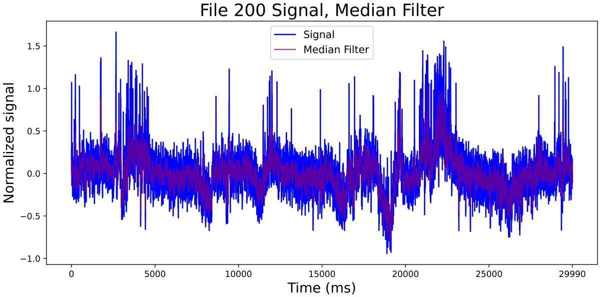
- Smoothed Signal
- High Pass Filter
- Derivatives



Preprocessing Data Channels

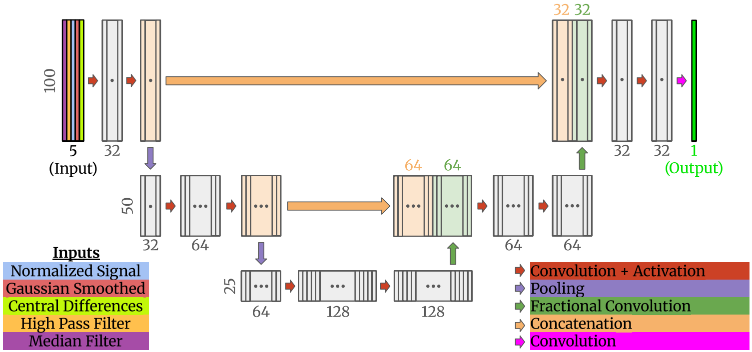
We calculate several other parallel feature channels to provide more information to the model.

- Smoothed Signal
- High Pass Filter
- Derivatives
- Median Filter



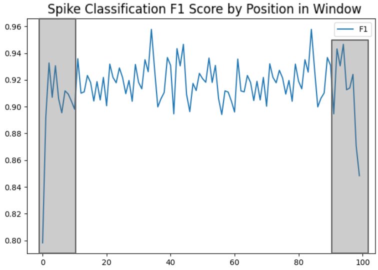
Model

- Input: 100 ms windows $\times$ 5 data channels.
- Output: 100 ms window of predicted probabilities.



Postprocessing

- The predicted probabilities of all windows are averaged over and then rounded to 0 (for a non-spike) or 1 (for a spike) to gather entire-file predictions.
- Further post-processing steps:
  - Confidence-based spike shadowing;
  - Pruning the ends of windows.

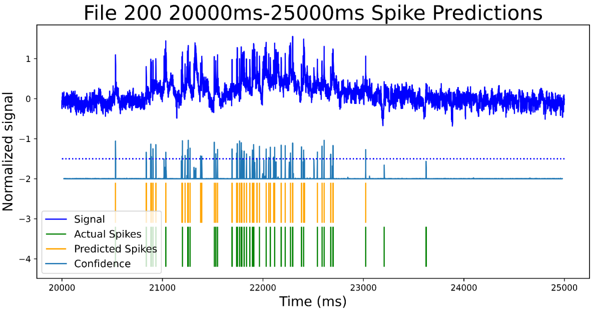


Metrics

		Predicted Label	
		Spike	No Spike
True Label	Spike	6100	337
	No Spike	448	1820065

- Accuracy: 0.9996;
- Precision: 0.9316;
- Recall: 0.9476;
- F1 Score: 0.9395.

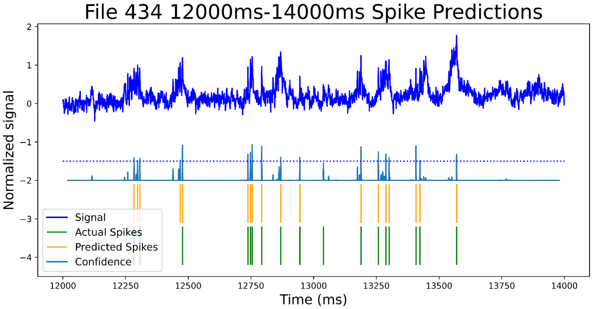
Prediction Results



Predicted Label

True Label	Spike	No Spike
Spike	104	4
No Spike	23	29819

Prediction Results



		Predicted Label	
		Spike	No Spike
True Label	Spike	48	19
	No Spike	14	29869

Future Work

- Further improving the model;
- Detecting harder “complex spikes”;
- Sorting spikes after identification;
- Public spike prediction package release.

Acknowledgments

We'd like to express gratitude towards:

- our mentor Dr. Lu Lu for providing guidance and feedback throughout the past months;
- Mitchell Daneker for helping keep us on track and answering our questions;
- Dr. Linlin Fan for providing the data and helping us explain the biology behind our project;
- the MIT PRIMES-USA program for allowing this research opportunity in the first place;
- and our families for letting us be here today.

References

- Changjia Cai, Johannes Friedrich, Amrita Singh, M. Hossein Eybposh, Eftychios A. Pnevmatikakis, Kaspar Podgorski, and Andrea Giovannucci, *Volpy: Automated and scalable analysis pipelines for voltage imaging datasets*, PLOS Computational Biology **17** (2021), no. 4, 1–27.
- Andrea Giovannucci, Johannes Friedrich, Pat Gunn, Jérémie Kalfon, Brandon L Brown, Sue Ann Koay, Jiannis Taxidis, Farzaneh Najafi, Jeffrey L Gauthier, Pengcheng Zhou, Baljit S Khakh, David W Tank, Dmitri B Chklovskii, and Eftychios A Pnevmatikakis, *Caiman an open source tool for scalable calcium imaging data analysis*, eLife **8** (2019), e38173.
- Jonathan Long, Evan Shelhamer, and Trevor Darrell, *Fully convolutional networks for semantic segmentation*, (2015).
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox, *U-net: Convolutional networks for biomedical image segmentation*, (2015).
- Jilt Sebastian, Mriganka Sur, Hema A Murthy, and Mathew Magimai-Doss, *Signal-to-signal neural networks for improved spike estimation from calcium imaging data.*, PLoS Comput Biol **17** (2021), no. 3, e1007921.

End of Presentation

THANK YOU!