ADEXLIF Neuron Modeling

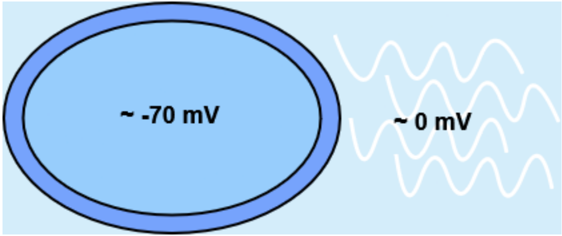
Jules Hunter and Dr. Brian Fischer

December 2024

Table of Contents

Membrane Potential

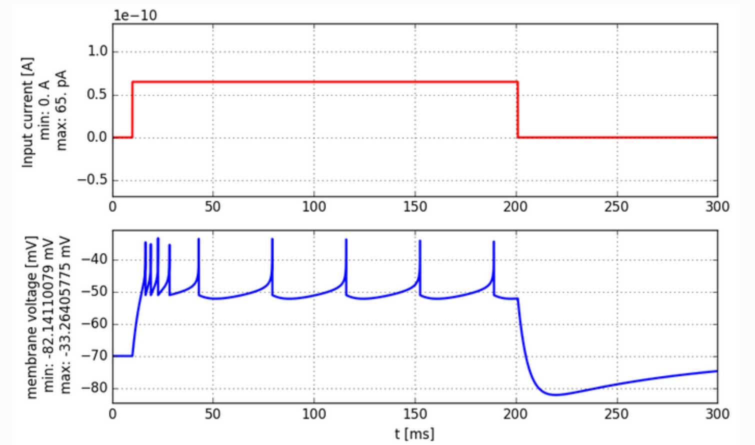
- Neurons are polarized: they have a net negative charge with respect to the surrounding bath
- The cell membrane, a lipid bilayer that is effectively impermeable to most charged molecules, separates the inside of the neuron from the extracellular bath.



- The difference between the external and internal voltage of the cell is called the membrane potential (difference)

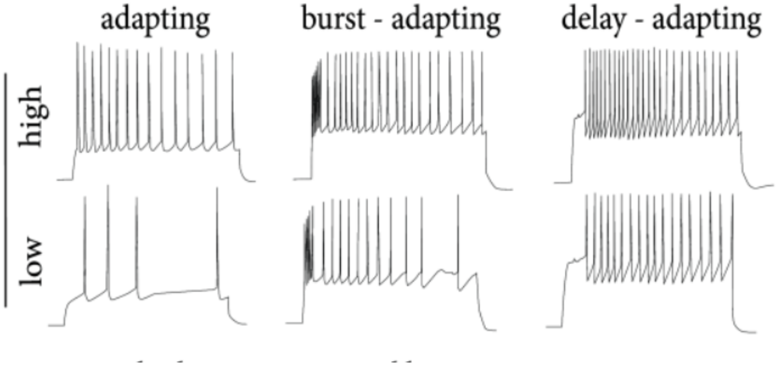
Action Potentials

- When current passes into a neuron, the neuron depolarizes. At a certain threshold, typically around -54mV, a reaction is triggered that causes an exponential increase in the charge of the neuron. This reaction is called an **action potential**, or a **spike**.

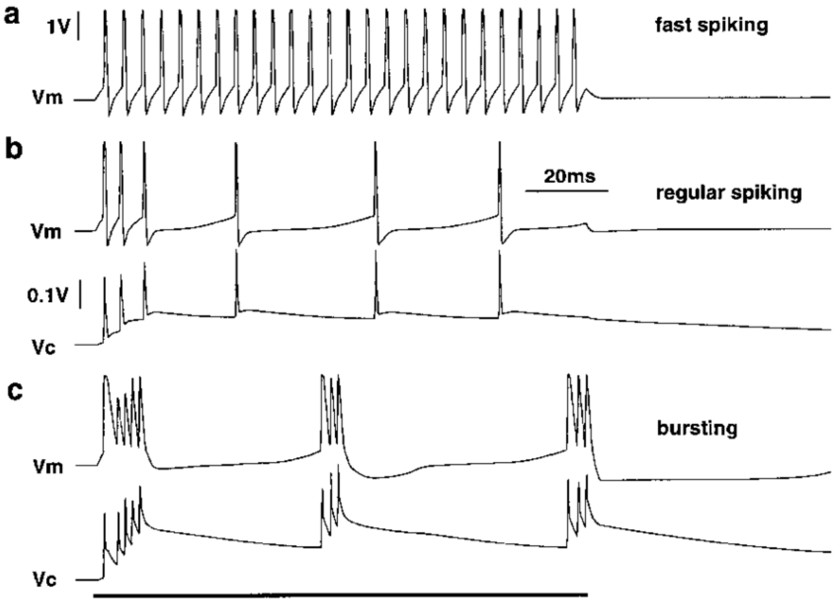


Adaptation current

- Adaptation Current: Adaptation current is a response to spiking where the neuron adapts to the input current, curtailing voltage spiking
- Adaptation current is represented by a separate differential equation
- Physically, the adaptation current is governed by potassium channels



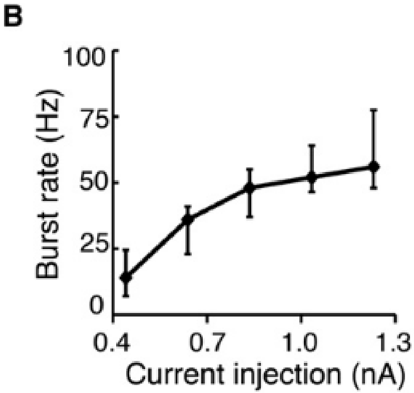
Bursting



a
1V
Vm
fast spiking
b
Vm
20ms
regular spiking
0.1V
Vc
c
Vm
bursting
Vc

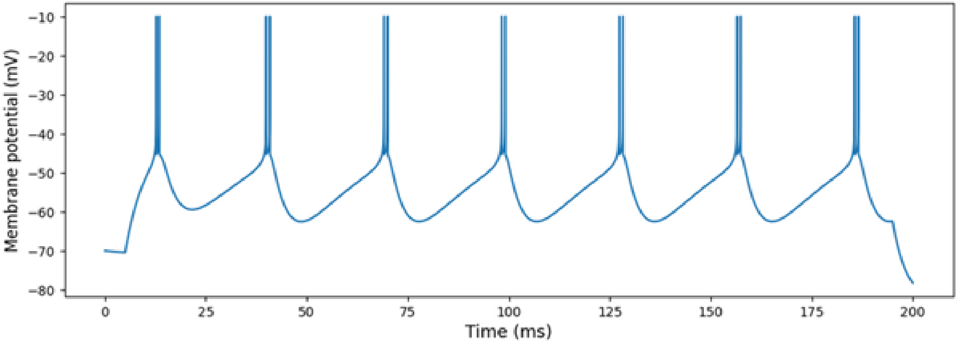
Objectives

For this research, we are interested in the frequency of bursting action potentials. We are attempting to replicate the results of Goddard et. al.:



ADEXLIF Model

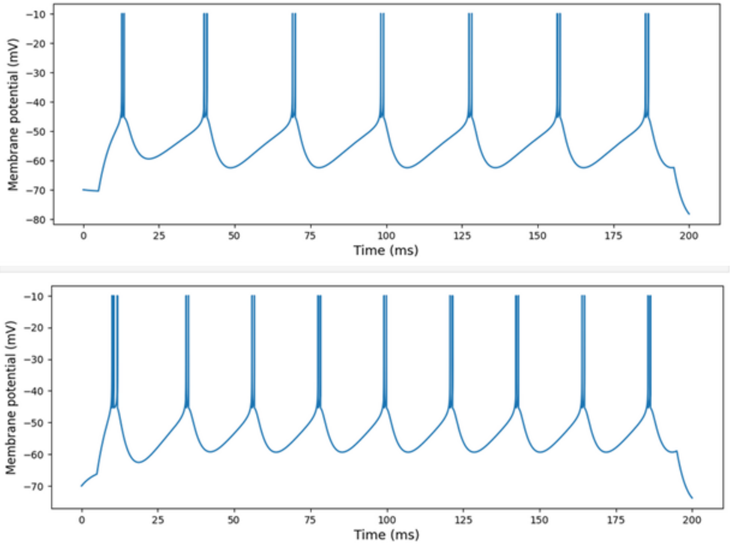
- Parameters (constants) tell us about the biophysical nature of the neuron
- A set of coupled differential equations map injected currents to membrane potential



ADEXLIF Model: Parameters

Parameter	Value
T_{ref} (Refractory Period)	0.05
V_{rest} (Resting Potential)	-70
V_{reset} (Reset Potential)	-45.5
E_L (Leak Reversal Potential)	-70.6
τ_{RC} (Membrane Time Constant)	5
R (Membrane Resistance)	$\frac{1000}{30}$
Δ_T (Spike Slope Factor)	2
V_T (Threshold Voltage)	-50.4
V_{thres} (Spike Threshold)	20
Jitter Range	0
Spike Probability	1
τ_w (Adaptation Time Constant)	13
b (Spike-Triggered Adaptation)	0.5
a (Subthreshold Adaptation)	4×10^{-3}
E_w (Adaptation Reversal Potential)	-48

ADEXLIF Parameters: Effects of changing El



ADEXLIF Model: Equations

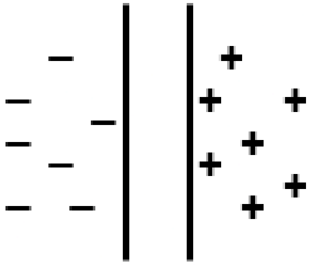
The Adaptive Exponential Leaky Integrate-and-Fire model (AdEx) consists of an exponential nonlinearity in the voltage equation coupled to a single adaptation variable w.

$$\text{Membrane Potential:} \quad \tau_m \frac{dV}{dt} = -(V - E_L) + \Delta_T e^{\frac{u - v_{th}}{\Delta_T}} - Rw + R_m I_e$$

$$\text{Adaptation Current:} \quad \tau_w \frac{dw}{dt} = a(V - E_L) - w + b\tau_w \Sigma_{t^{(f)}} \delta(t - t^{(f)})$$

Membrane Potential Equation: The Capacitor Analogy

- The negative charge inside a neuron is separated from the outside by the cell membrane
- Additionally, ion concentrations differ inside and outside the neuron
- There is typically an excess negative charge on the inside surface of the cell membrane, and a balancing positive charge on its outside surface. In this arrangement, the cell membrane functions like a capacitor.

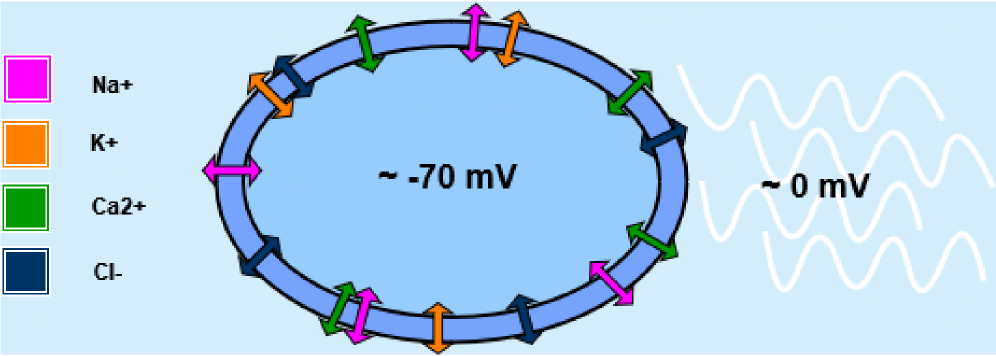


Membrane Potential Equation: Derivation

$$\begin{aligned} \text{Capacitor:} \quad & Q = C_m V \\ \text{Time derivative } (\frac{dQ}{dt} \text{ is current}): \quad & \frac{dQ}{dt} = C_m \frac{dV}{dt} \\ \text{Current per unit area:} \quad & -i_m + \frac{I_e}{A} = c_m \frac{dV}{dt} \end{aligned}$$

Different models use different i_m (membrane currents) to represent different physical phenomena, namely the types of channels present in the neuron's membrane.

Membrane Potential Equation: Ion Channels



Membrane Potential Equation: i_m Components

$$\boxed{c_m \frac{dV}{dt} = -i_m + \frac{I_e}{A}}$$

Linear leak, $\overline{g}_L(V - E_L)$:

$$c_m \frac{dV}{dt} = -\overline{g}_L(V - E_L) + \frac{I_e}{A}$$

$$\tau_m \frac{dV}{dt} = -(V - E_L) + R_m I_e$$

Exponential activation term, $\Delta_T e^{\frac{v - v_{th}}{\Delta_T}}$:

$$\tau_m \frac{dV}{dt} = -(V - E_L) + \Delta_T e^{\frac{v - v_{th}}{\Delta_T}} + R_m I_e$$

Adaptation Current, w :

$$\tau_m \frac{dV}{dt} = -(V - E_L) + \Delta_T e^{\frac{u - v_{th}}{\Delta_T}} - Rw + R_m I_e$$

Adaptation current equation

Membrane Potential: $$\tau_m \frac{dV}{dt} = -(V - E_L) + \Delta_T e^{\frac{u - v_{th}}{\Delta_T}} - Rw + R_m I_e$$

$$\boxed{\text{Adaptation Current:} \quad \tau_w \frac{dw}{dt} = a(V - E_L) - w + b\tau_w \sum_{t^{(f)}} \delta(t - t^{(f)})}$$

- The adaptation current is fed back into the voltage equation with resistance R, influenced by parameters a and τ)
- At each threshold crossing the voltage is reset to $V = E_L$ and the adaptation variable w is increased by an amount b, representing spike-triggered adaptation

Evaluating the ADEXLIF Model: Euler's method

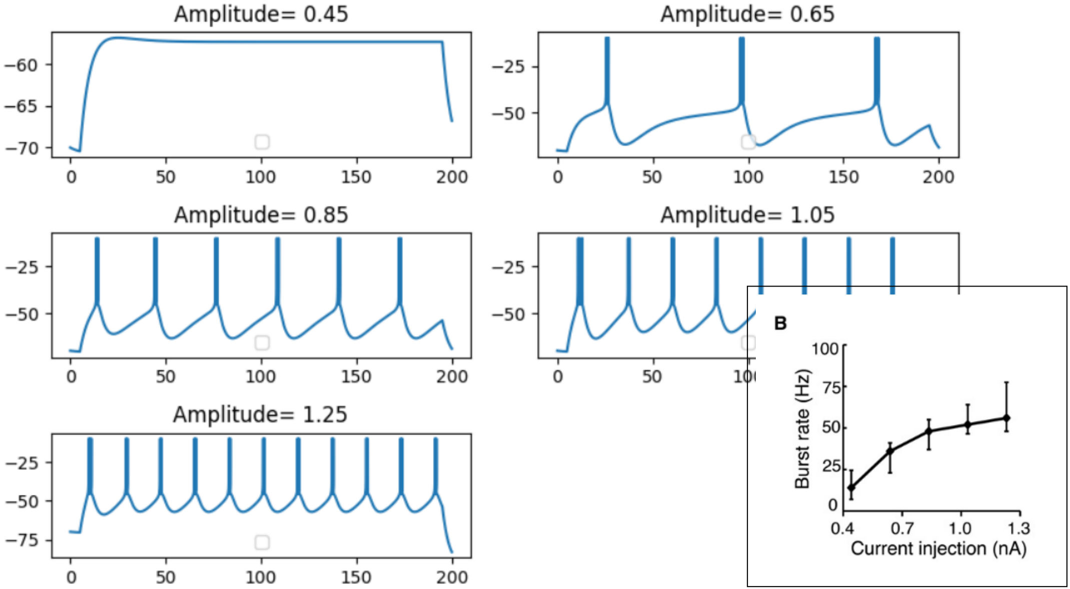
Discretize time: Choose a time step Δt and discretize time into intervals. Let $t_n = t_0 + n\Delta t$, where n is the time step index and t_0 is the initial time.
Initialize values: Set initial conditions for $v(0)$ and $w(0)$, denoted as v_0 and w_0, respectively.
Apply Euler's method: At each time step, update the values of v_n and w_n using the formulas:

$$v_{n+1} = v_n + \Delta t \cdot f_1(v_n, w_n, t_n)$$

$$w_{n+1} = w_n + \Delta t \cdot f_2(v_n, w_n, t_n)$$

Repeat for each time step: For each new time step n, update the values of u_n and w_n using the above equations until the desired final time t_N is reached.

ADEXLIF Model: Outputs for 5 amplitudes



Calculating Statistics: Counting bursts

Python Code

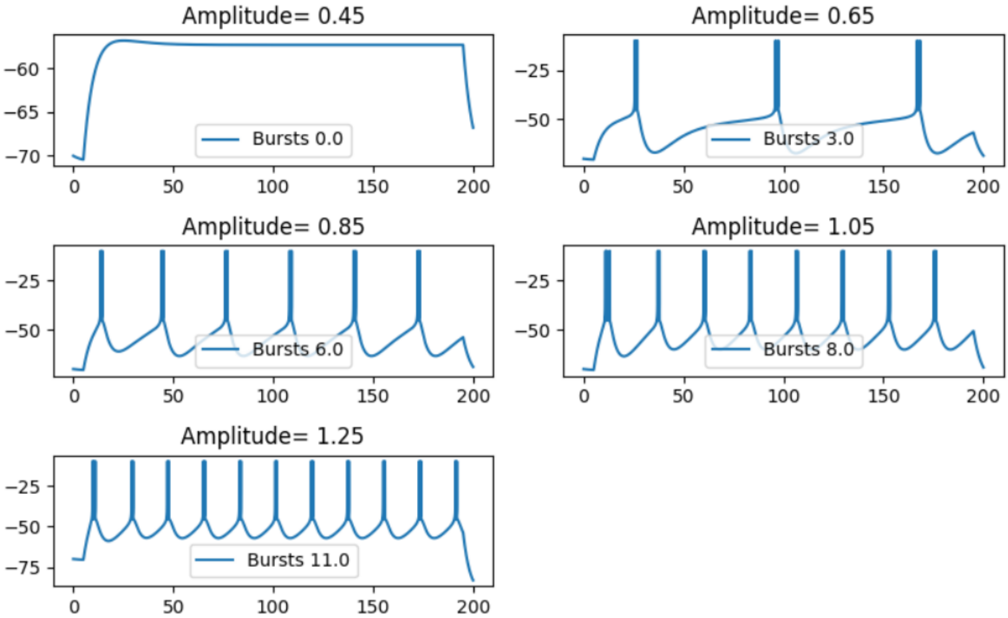
```
kmeans = KMeans(n_clusters=2, random_state=0).fit(spike_diffs_shape)

# Get cluster labels
labels = kmeans.labels_

#separate data based on labels
small_values = spike_diffs_shape[labels == 0]
large_values = spike_diffs_shape[labels == 1]

# Identify which cluster is for "small" and which is for "large"
# Check the cluster centroids
centroids = kmeans.cluster_centers_
if centroids[0] < centroids[1]:
    small_cluster, large_cluster = small_values, large_values
else:
    small_cluster, large_cluster = large_values, small_values
```

Counting Bursts Output



Statistical Inference

- Now that we have a good model, we can calculate the burst frequency at the necessary amplitudes.
- However, we still don't know what the parameters of the neurons in the paper were.

- We need to the relationship between changing parameters (for this we select E_l and b) and burst frequency.
- We rely on **statistical inference**, specifically **simulation-based inference (SBI)**.
- SBI is fundamentally based on **Bayesian inference**.

Bayes' Theorem

Bayes' Theorem:

$$P(\theta \mid D) = \frac{P(D \mid \theta)P(\theta)}{P(D)}$$

where:

- $P(\theta \mid D)$ is the posterior probability of θ given D.
- $P(D \mid \theta)$ is the likelihood, representing how probable the observed data D is under θ.
- $P(\theta)$ is the prior probability of θ.
- $P(D)$ is a normalizing factor.

Bayesian Inference

1. Start with a Prior $P(\theta)$
   - This is our initial guess about the parameters before seeing any data.
2. Collect Data D
   - We observe new evidence, such as burst count/burst frequency.
3. Compute the Likelihood $P(D \mid \theta)$
   - This tells us how likely the data D is under different possible values of θ.
4. Apply Bayes' Theorem to Update the Posterior
   - The posterior distribution is:

$$P(\theta \mid D) = \frac{P(D \mid \theta)P(\theta)}{P(D)}$$

   - This gives a refined estimate of θ based on the observed data.
5. Use the Posterior as the New Prior
   - When there is more data, we repeat the process, treating the new posterior as our updated prior.

Bayesian Inference: Limitations

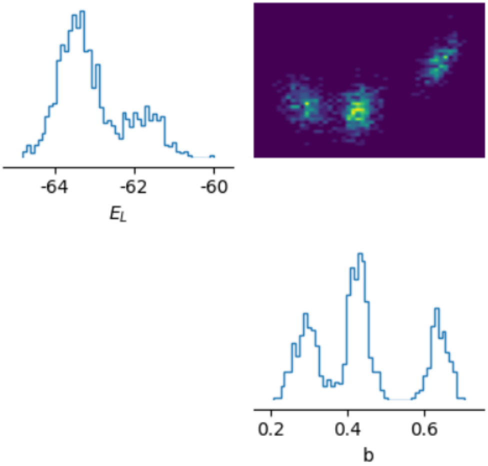
- The key computational challenge in Bayesian inference is computing the **likelihood** $P(D \mid \theta)$, which is often intractable.
- In the case of an ADEXLIF neuron model, the likelihood $P(D \mid \theta)$ isn't calculable.
- We use a machine learning based form of statistical inference called **Simulation-Based Inference (SBI)**

Simulation-Based Inference (SBI)

1. Define a prior distribution over model parameters.
2. Sample parameters from the prior, simulate the model, and extract summary statistics.
3. Train a neural network to learn the mapping between parameters and summary statistics.
4. Construct the posterior distribution using one of two approaches:
   - Estimate the likelihood and apply Bayes' Theorem.
   - Directly estimate the posterior distribution.

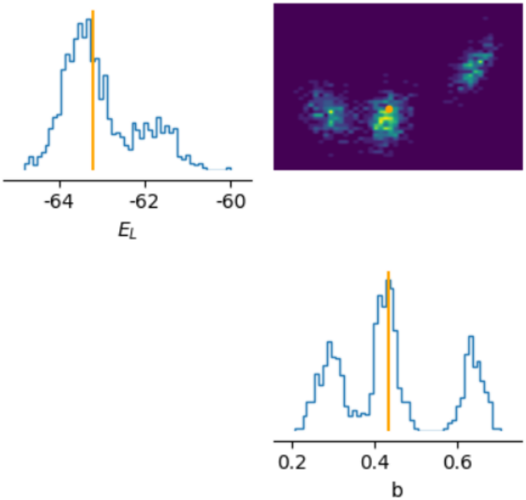
In our research, we use Neural Posterior Estimation (NPE), which directly models the posterior distribution using deep learning.

Posterior Distribution

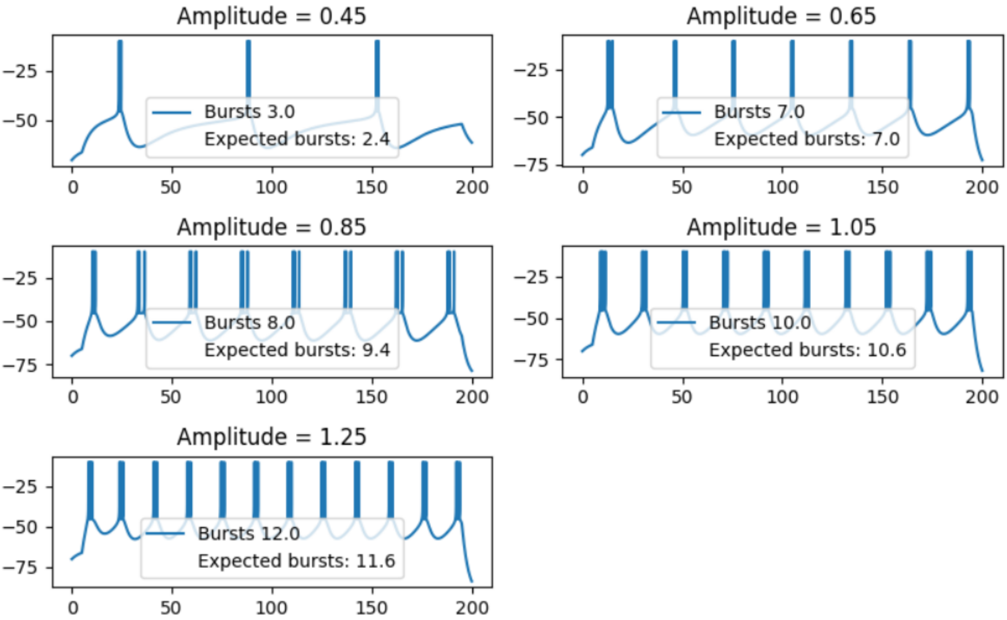


The posterior distribution represents our updated beliefs about the model parameters after observing neuron behavior.

Validation: Choose a single sample from the posterior



Validation


Amplitude = 0.45
Bursts 3.0
Expected bursts: 2.4
Amplitude = 0.65
Bursts 7.0
Expected bursts: 7.0
Amplitude = 0.85
Bursts 8.0
Expected bursts: 9.4
Amplitude = 1.05
Bursts 10.0
Expected bursts: 10.6
Amplitude = 1.25
Bursts 12.0
Expected bursts: 11.6
−25
−50
−75
0
50
100
150
200

Thank you!

Resources

- Dayan, Peter, and L F Abbott. Theoretical Neuroscience : Computational and Mathematical Modeling of Neural Systems. Cambridge, Mass., Mit Press, [Ca, 2009]
- Gerstner, Wulfram , et al. “Neuronal Dynamics.” Neuronaldynamics.epfl.ch, neuronaldynamics.epfl.ch/index.html.
- Rasche, C, and R Douglas. Analog Integrated Circuits and Signal Processing, vol. 23, no. 3, 1 Jan. 2000, pp. 227–236, https://doi.org/10.1023/a:1008357931826. Accessed 5 Dec. 2024.