Machine Learning Methods for Neural Data Analysis

Lecture 3: Basic Neurobiology and Simple Spike Sorting

Scott Linderman

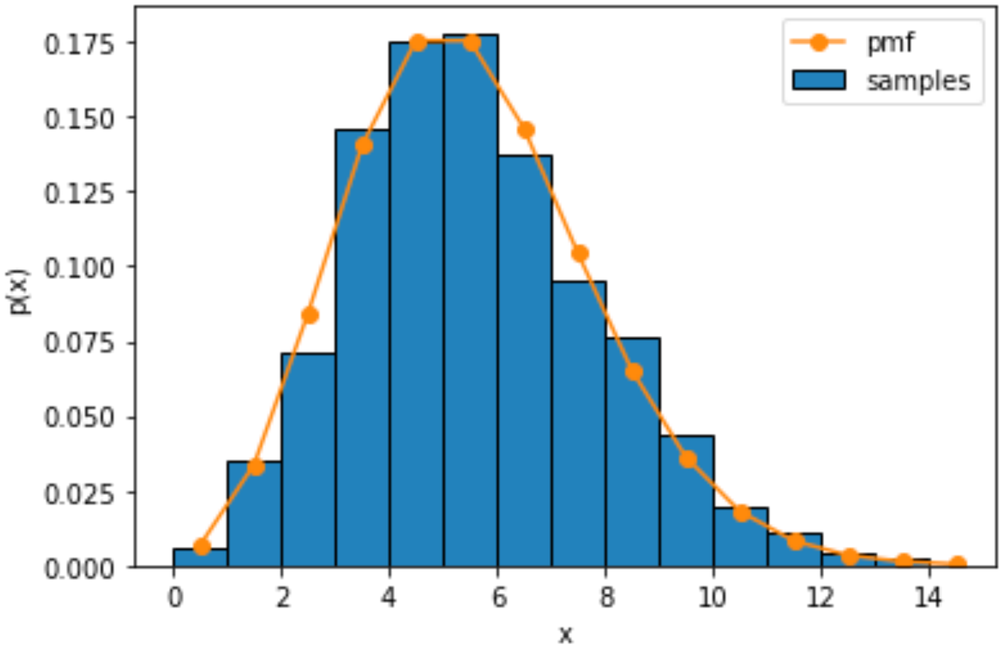
STATS 220/320 (NBIO220, CS339N). Winter 2023.

Announcements

- Office hours:
  - Sifan: **Tues 6-7:30pm** in **Sequoia 207**
  - Scott: **Weds 1:15-2:30** in **Wu Tsai Neurosciences Inst. Room M252G**
  - Ying: **Thurs 11:00am-12:30pm**
- Lab 0: PyTorch primer (not graded) will be released this evening.

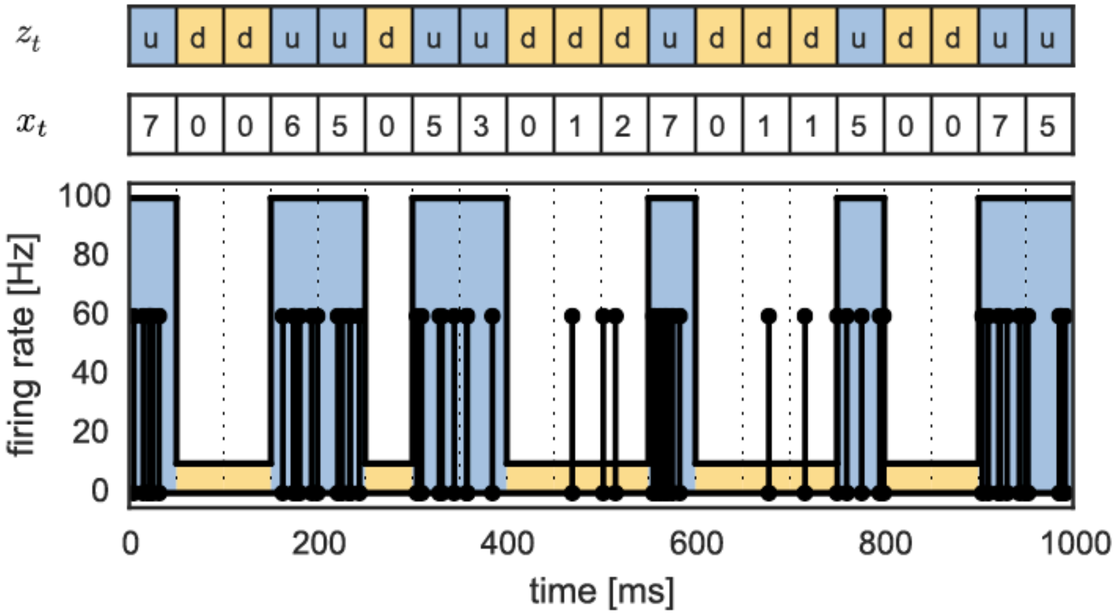
Last Time

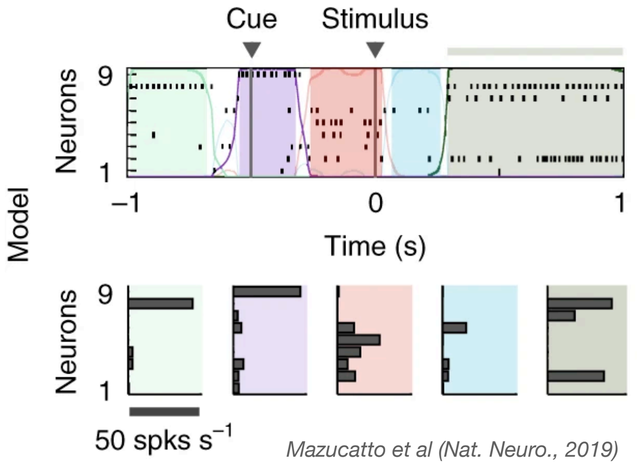
- **Poisson** distribution and its **conjugate prior**, the gamma distribution.
- We learned how to construct **joint distributions** using the **product rule**, how to compute **marginal distributions** with the **sum rule**, and how to find the **posterior distribution** with **Bayes' rule**.
- We learned about **maximum likelihood estimation (MLE)** and **maximum *a posteriori* (MAP)** estimation.
- We encountered **conjugate priors** where the posterior distribution is in the same family, making calculations particularly simple.



Mixture models and latent variables

- Real data is rarely so simple!
- One way to build richer models is via **latent variables**.
- Let $z_t \in \{0,1\}$ be the *latent state:*
  - E.g. high firing (“up”) and low firing (“down”) states.
  - Sequences of “coding states” in gustatory cortex.
- Each state has its own firing rate.
- Our goal is to infer these states given only the spike trains.





Mazucatto et al (Nat. Neuro., 2019)

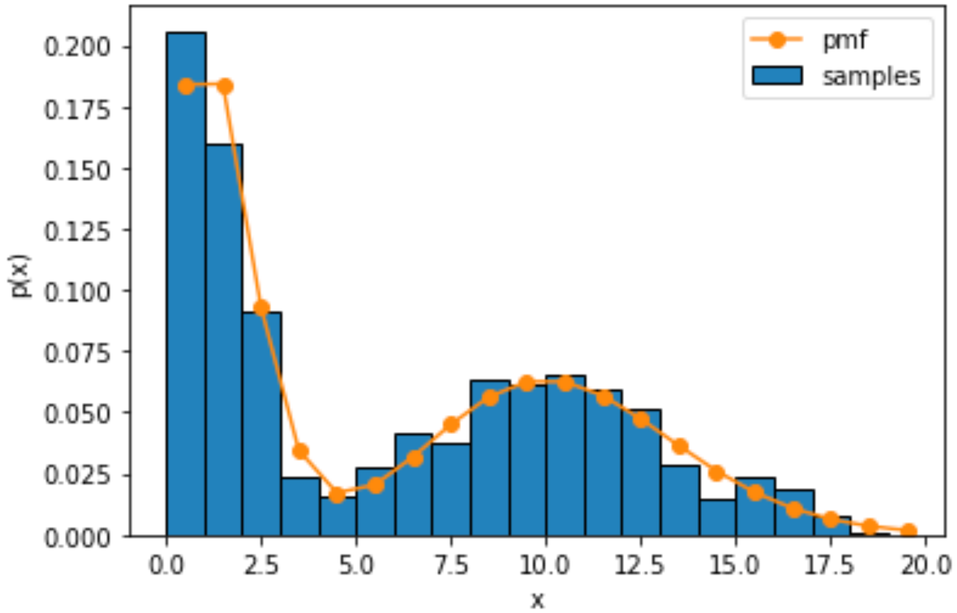
A Poisson mixture model

Finally, assume that the latent variables are equally probable and independent across time. Formally, we can write that as a **categorical distribution** with equal probabilities for both states,

$$z_t \sim \text{Cat}([\tfrac{1}{2}, \tfrac{1}{2}]).$$

The resulting model is called a **mixture model** because the marginal distribution, $p(x_t \mid \boldsymbol{\lambda})$ where $\boldsymbol{\lambda} = (\lambda_0, \lambda_1)$, is a mixture of two Poisson distributions,

$$\begin{aligned} p(x_t \mid \boldsymbol{\lambda}) &= \sum_{z_t \in \{0,1\}} p(x_t, z_t \mid \boldsymbol{\lambda}) \\ &= \sum_{z_t \in \{0,1\}} p(x_t \mid z_t, \boldsymbol{\lambda})\, p(z_t) \\ &= \frac{1}{2}\text{Pois}(x_t \mid \lambda_0) + \frac{1}{2}\text{Pois}(x_t \mid \lambda_1) \end{aligned}$$



Fitting a mixture model by coordinate ascent

Conceptually, fitting the mixture model is no different than fitting the the simple Poisson model above.

We will perform MAP estimation to find,

$$\mathbf{z}_{\text{MAP}}, \boldsymbol{\lambda}_{\text{MAP}} = \arg\max p(\mathbf{z}, \boldsymbol{\lambda} \mid \mathbf{x})$$

where $\mathbf{z} = (z_1, \ldots, z_T)$. Again, this is equivalent to maximizing the joint probability.

Expanding the joint distribution over spike counts, latent variables, and rates,

$$\begin{aligned} p(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}) &= \left[\prod_{t=1}^{T} p(x_t \mid z_t, \boldsymbol{\lambda})\, p(z_t)\right] p(\boldsymbol{\lambda}) \\ &= \left[\prod_{t=1}^{T} \text{Pois}(x_t \mid \lambda_{z_t}) \times \frac{1}{2}\right] \text{Ga}(\lambda_0; \alpha, \beta)\, \text{Ga}(\lambda_1; \alpha, \beta) \end{aligned}$$

Fitting a mixture model by coordinate ascent

Fixing the rates, the most likely state at time t is,

$$z_t = \begin{cases} 1 & \text{if } \text{Pois}(x_t \mid \lambda_1) \geq \text{Pois}(x_t \mid \lambda_0) \\ 0 & \text{otherwise} \end{cases}$$

Fixing the states, the most likely rates are

$$\lambda_k = \frac{\alpha'_k - 1}{\beta'_k}$$

$$\alpha'_k = \alpha + \sum_{t=1}^{T} x_t \, \mathbb{I}[z_t = k]$$

$$\beta'_k = \beta + \sum_{t=1}^{T} \mathbb{I}[z_t = k]$$

Question: How does this relate to K-Means?

Further Reading

There are many great references on probabilistic modeling. I like:

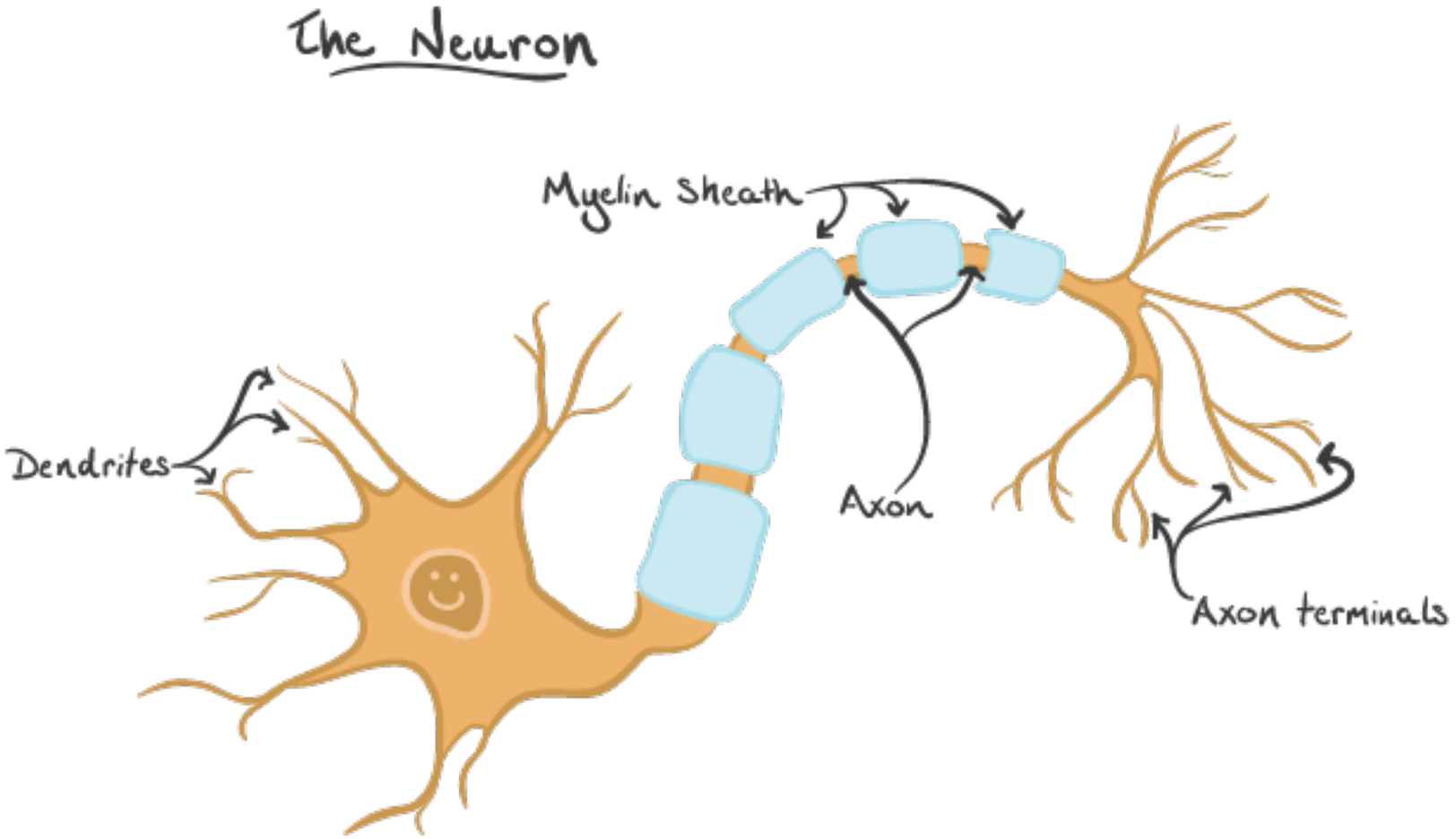
- Ch 2.1 and 2.2 of [Murphy, 2023]
- Ch 1.2 of [Bishop, 2006]
- See references on the course website.
- **Next time: Basic Neurobio and Simple Spike Sorting!**

Agenda

1. Basic neurobiology
2. Spike sorting as matrix factorization
3. Maximum a posteriori inference

Neurobiology 101

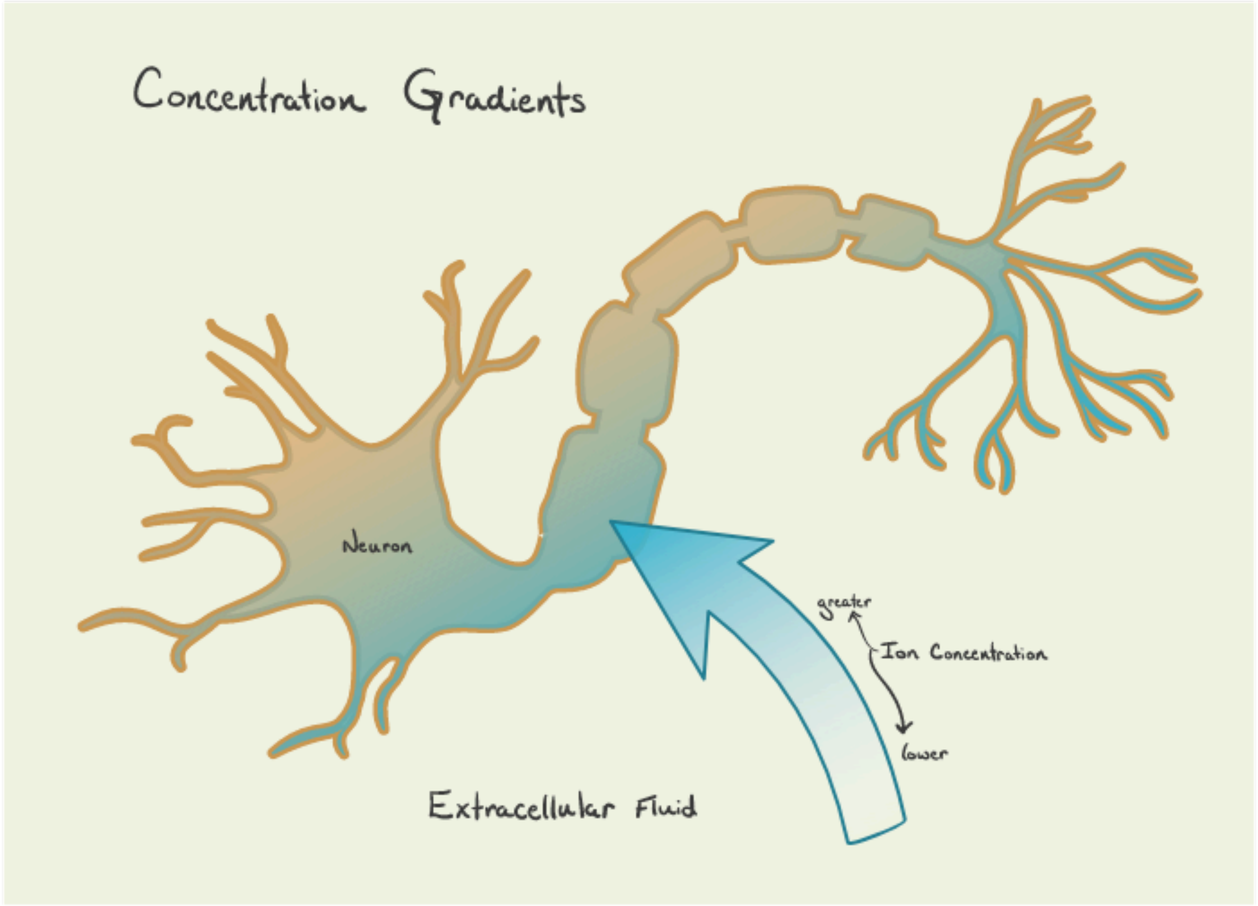
Anatomy of a neuron



https://www.khanacademy.org/test-prep/mcat/organ-systems/neuron-membrane-potentials/a/neuron-action-potentials-the-creation-of-a-brain-signal

Neurobiology 101

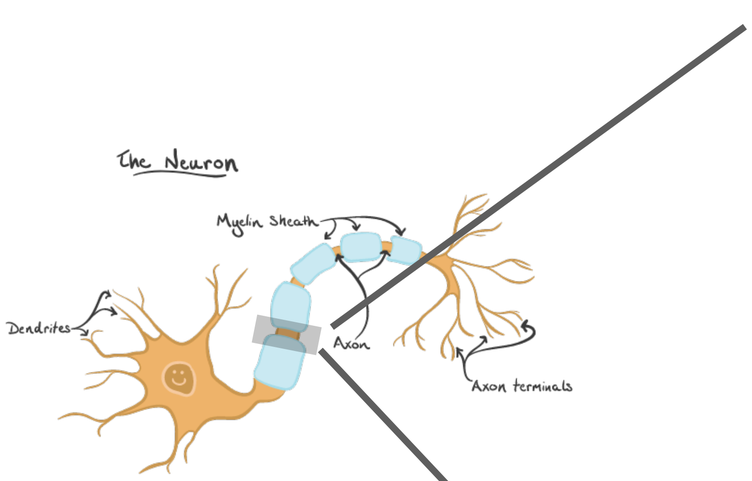
Anatomy of a neuron

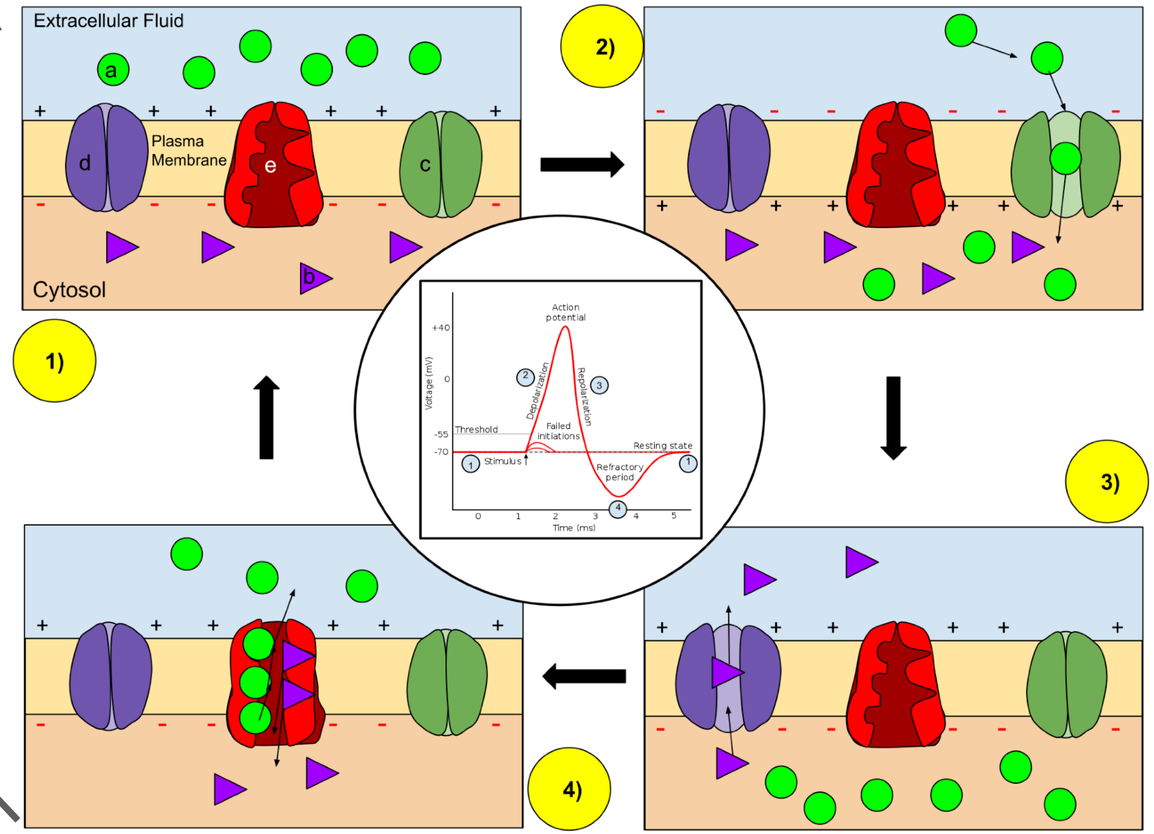


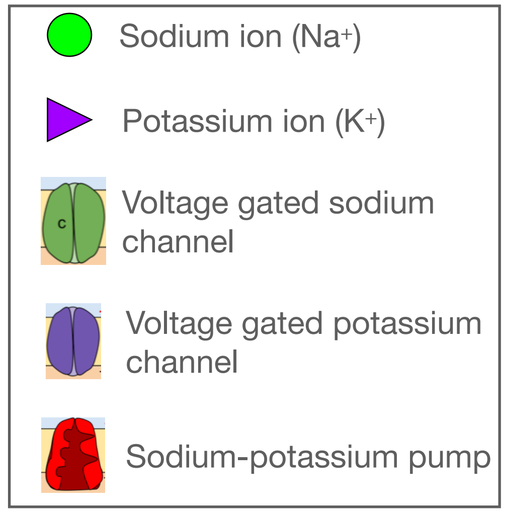
https://www.khanacademy.org/test-prep/mcat/organ-systems/neuron-membrane-potentials/a/neuron-action-potentials-the-creation-of-a-brain-signal

Neurobiology 101

Voltage-gated ion channels



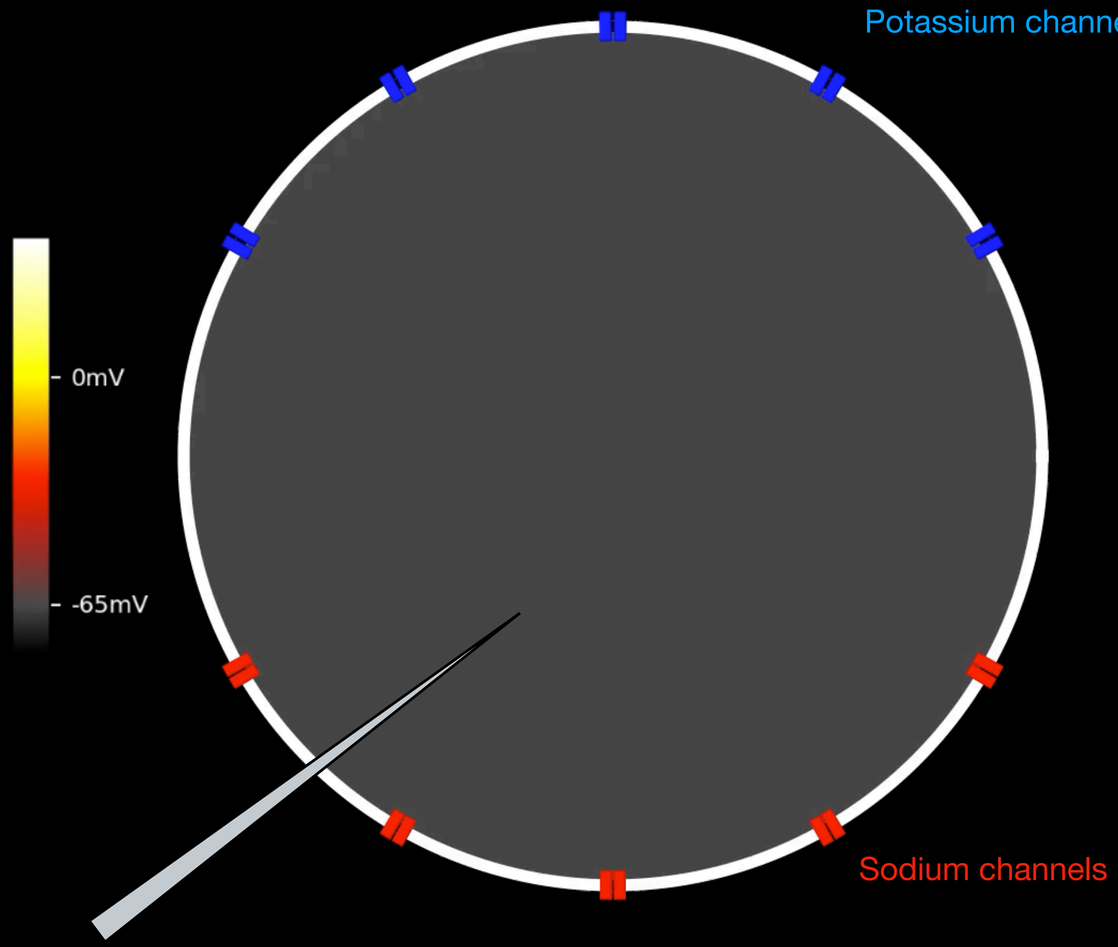


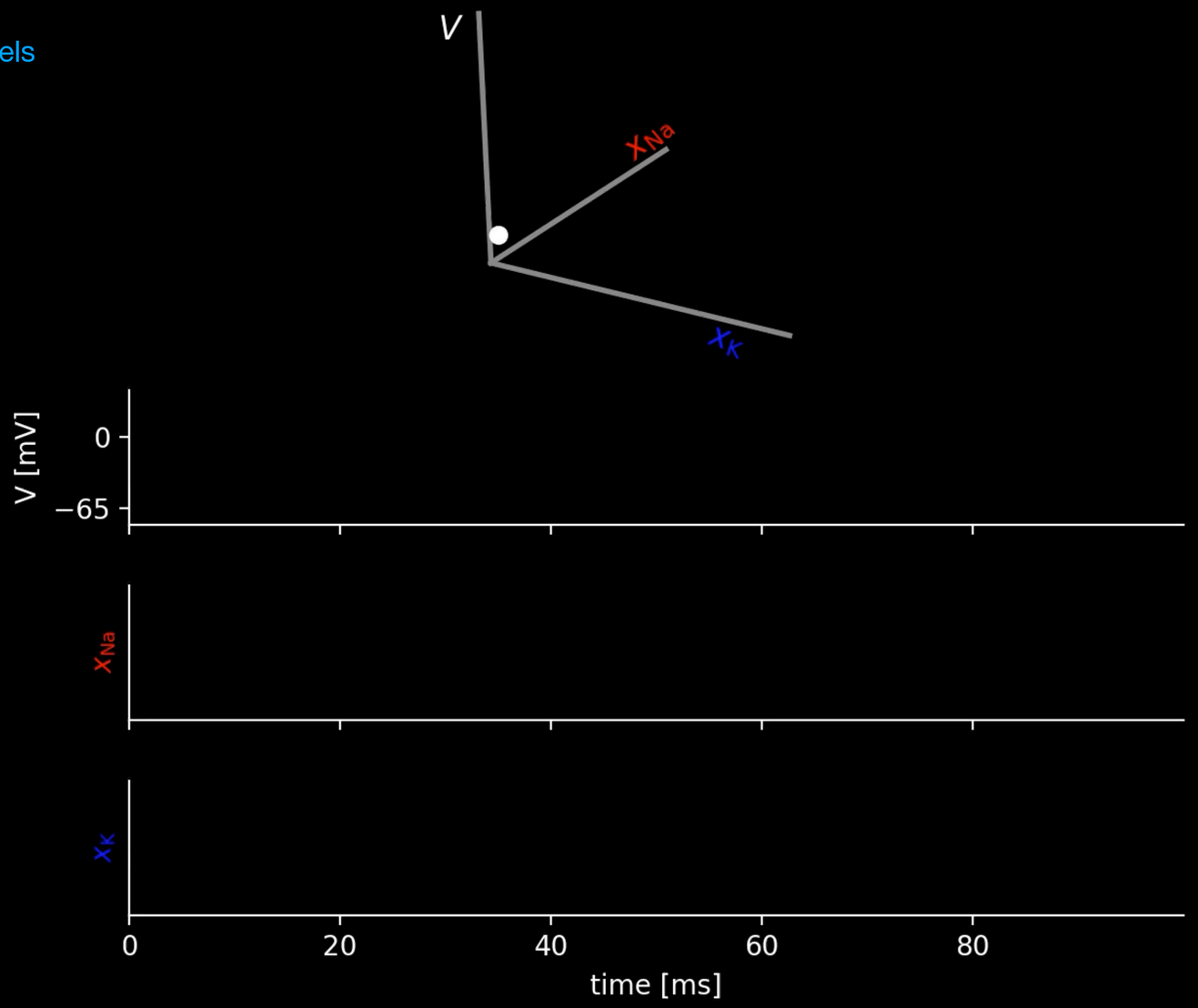


https://en.wikipedia.org/wiki/Action_potential

Neurobiology 101

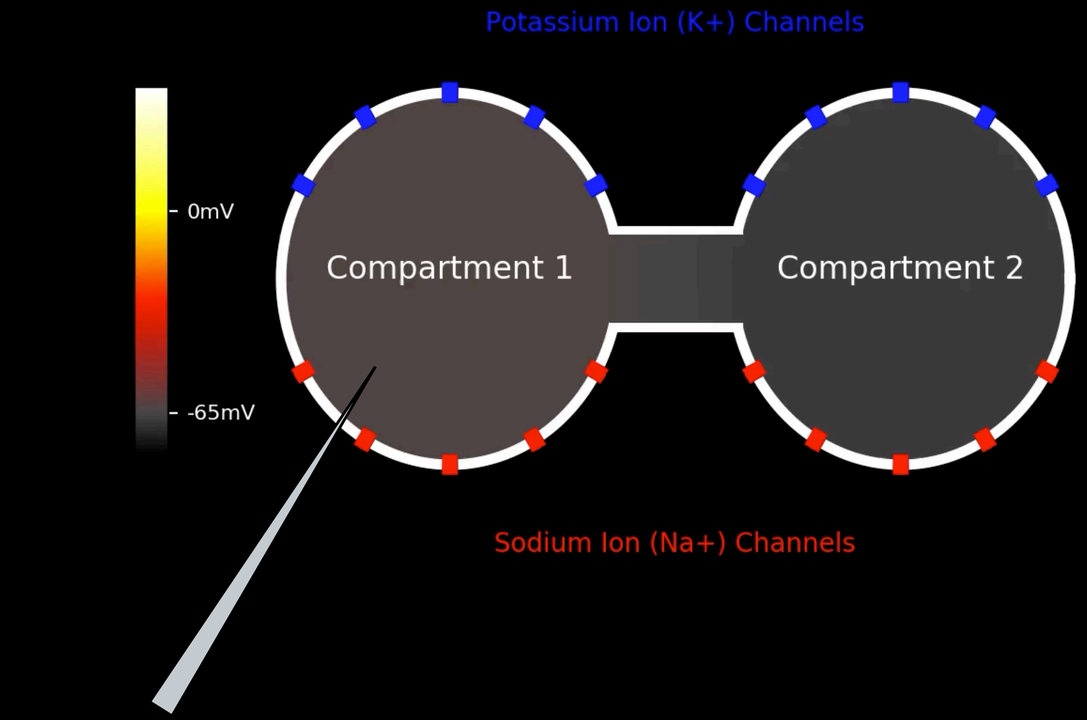
Action potentials

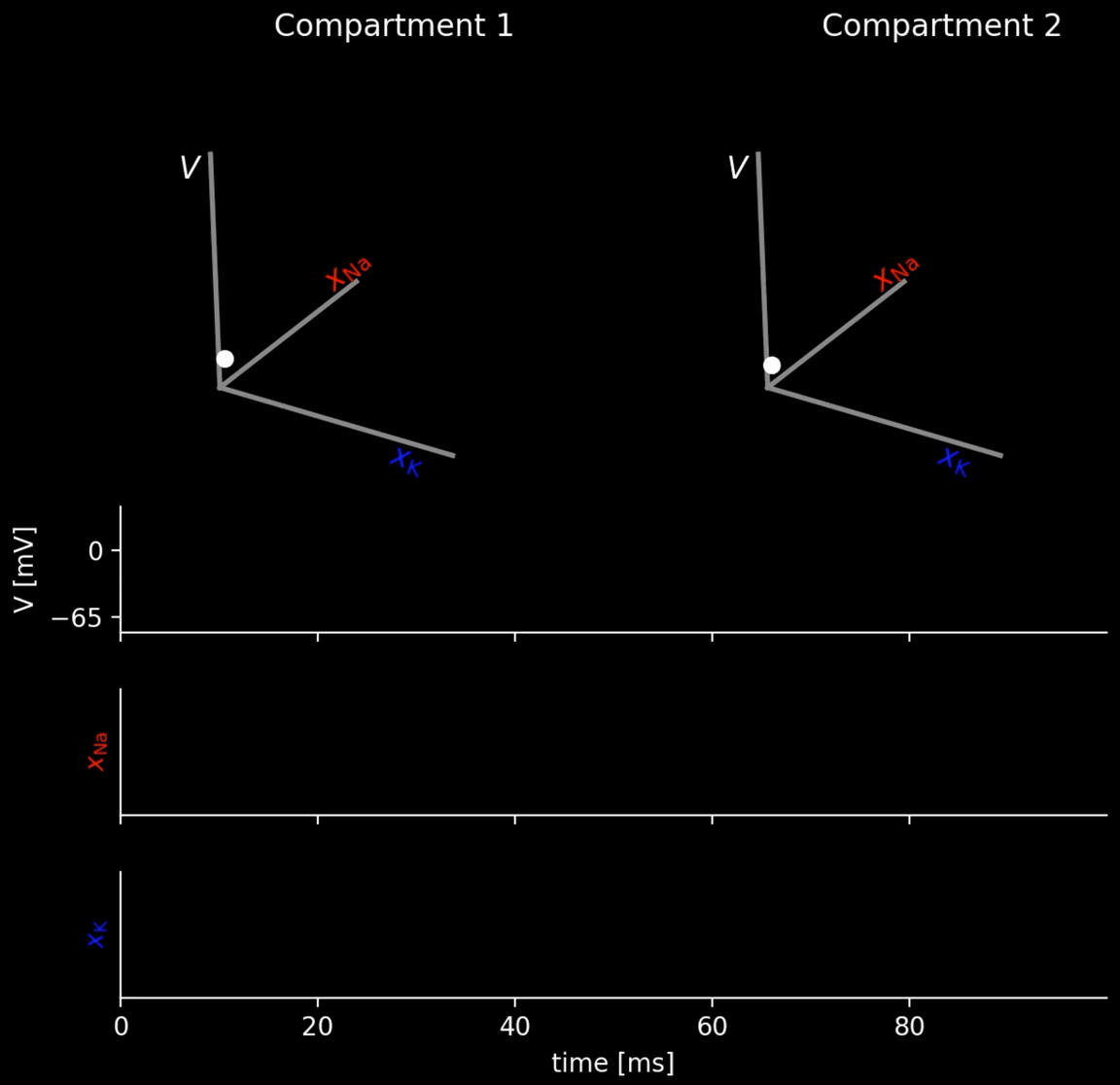




Neurobiology 101

Action potential propagation

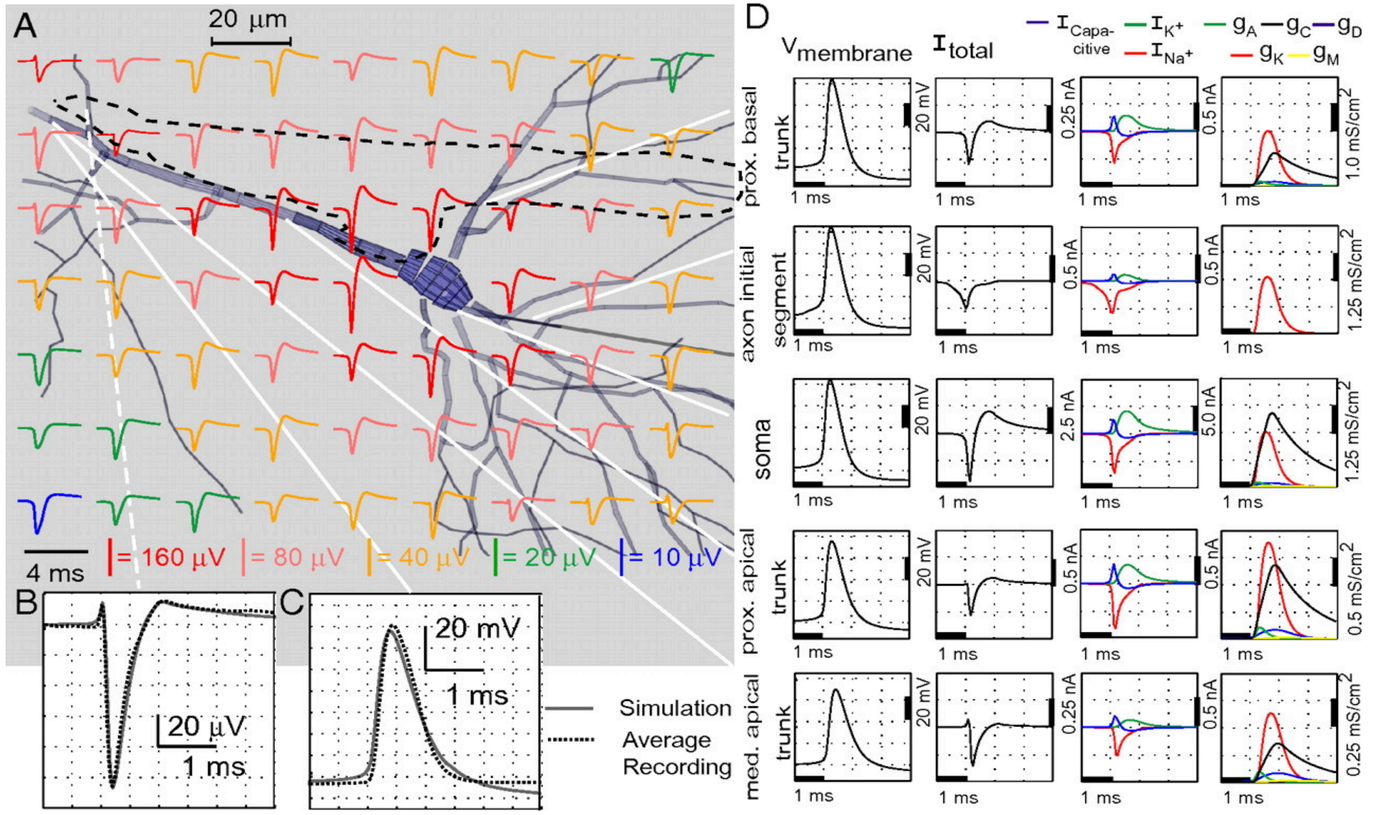




Neurobiology 101

Extracellular voltage recordings

- The *membrane potential* spikes by 50-100mV during an action potential.
- The *extracellular action potential* (EAP) is roughly proportional to the total current (I_{total}) in nearby compartments of the cell.
- The EAP shows a triphasic response with a sharp negative deflection of 50-100μV.
- Amplitudes fall off with distance from the cell.

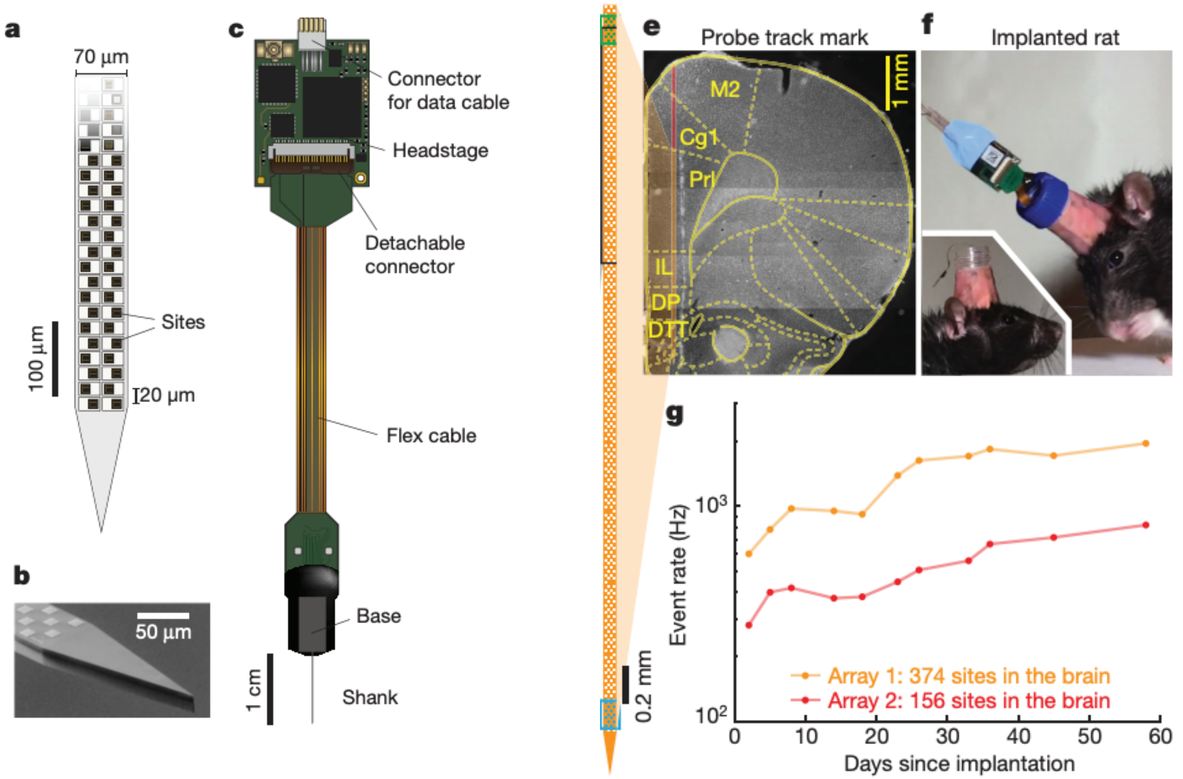


Gold et al (J Neurophysiol., 2006)

Neuropixels

High-density silicon probes

- Modern recording probes like **Neuropixels** measure the electrical activity of **hundreds of cells** across **multiple brain regions** simultaneously.
- First gen. Neuropixels had 960 recording sites spaced 20μm apart, of which 384 could be used simultaneously.
- Finely spaced sites means that single neurons can activate 5-50 sites.
- Compare spacing to scale bar on previous slide.

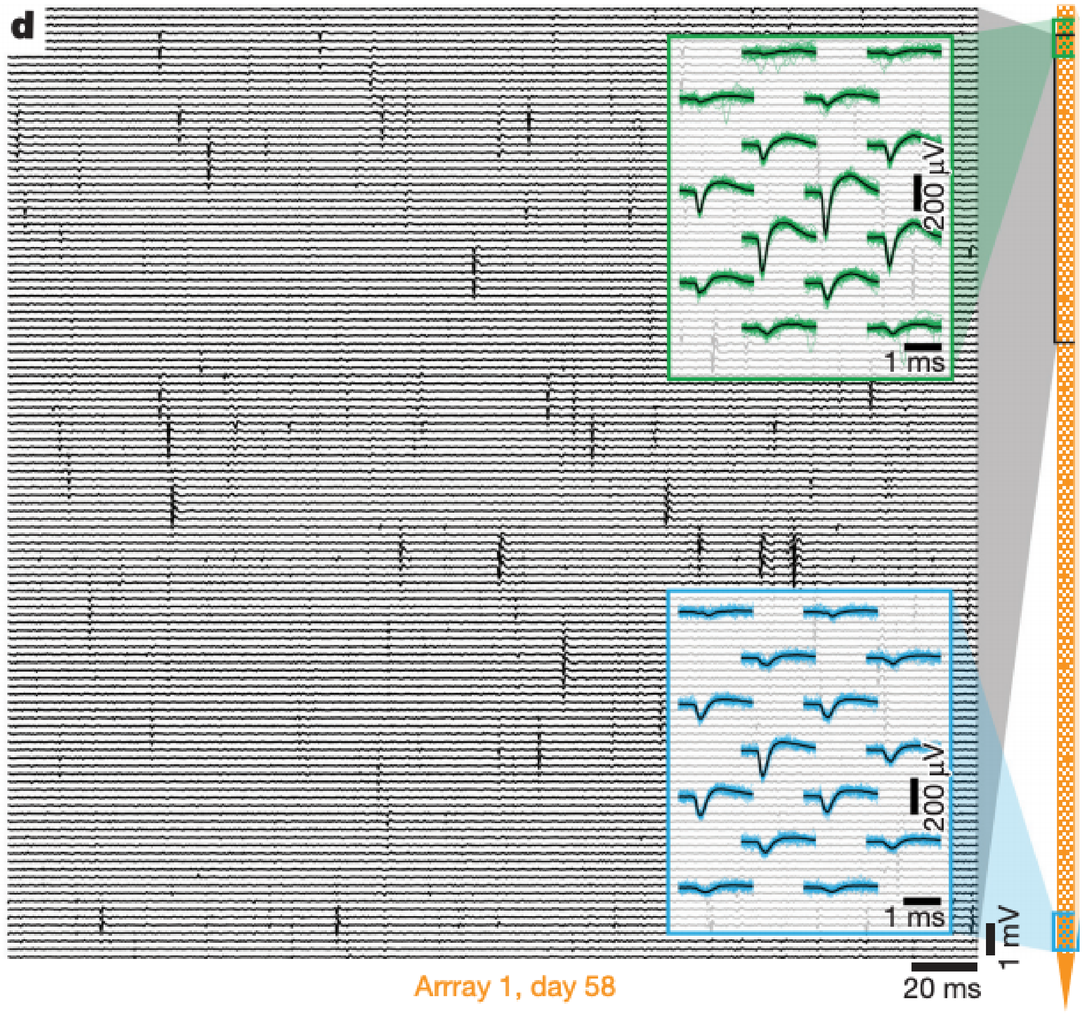


Jun et al, 2017.

Neuropixels

High-density silicon probes

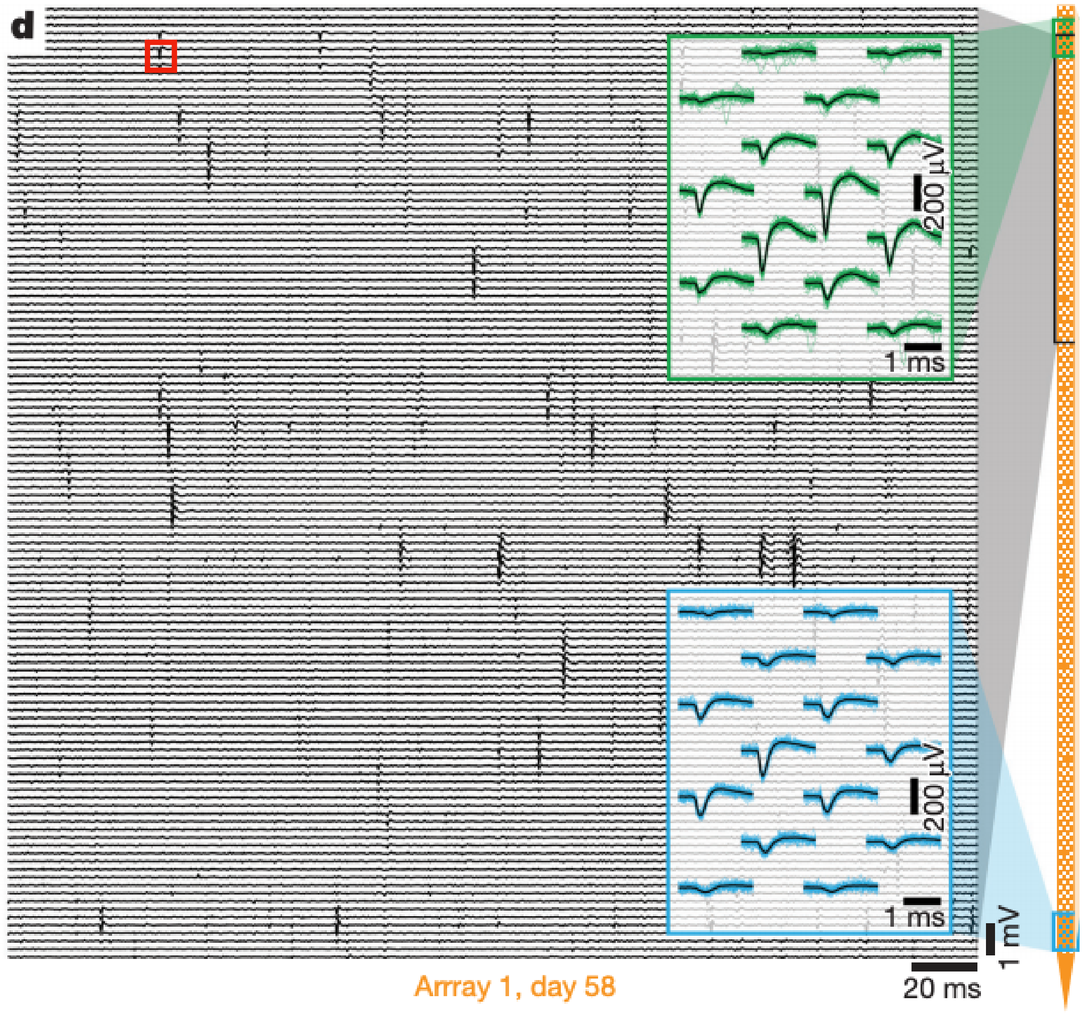
- The raw data is a **multidimensional time series** of **voltage measurements**, one for each recording site on the probe.
- When neurons near the probe fire an **action potential**, it registers a **spike in the voltage** on nearby channels.
- Our goal is to **find the spikes** in this time series and **assign neuron labels** based on their waveforms.



Simple Spike Sorting

A simple probabilistic model

- Start with a zoomed-out view of average voltage in relatively large time bins (e.g. 2ms).
- Let C be the number of channels.
- Let T be the number of 2ms time bins.
- Let $x_{c,t}$ be the average voltage on channel c in time bin t.
- At this resolution, spikes can be contained to a single bin.



A simple probabilistic model

Assumptions

- There are K neurons. When neuron k spikes it produces a **waveform** $\mathbf{w}_k = (w_{k,1}, \ldots, w_{k,C}) \in \mathbb{R}^C$
- Let $\mathbf{a}_k = (a_{k,1}, \ldots, a_{k,T}) \in \mathbb{R}_+^T$ denote the time series of spike **amplitudes** for neuron k.
  - Since neurons spike only a few times a second, amplitudes are mostly zero.
  - Amplitudes are non-negative.
- If two neurons spike at the same, waveforms add.
- Voltage recordings have additive noise.

A simple probabilistic model

Matrix factorization perspective

A simple probabilistic model

Accounting for scale invariance

- Notice that the model is **invariant to rescaling**.
  - Multiple $\mathbf{a}_k$ by constant $c > 0$ and scale $\mathbf{w}_k$ by c^{-1}.
- We can remove this degree of freedom by forcing $\|\mathbf{w}_k\|_2 = 1$; e.g., with a **uniform prior** on the unit hypersphere,

$$\mathbf{w}_k \sim \text{Unif}(\mathbb{S}_{C-1})$$

- where $\mathbb{S}_{C-1} = \left\{\mathbf{u} : \mathbf{u} \in \mathbb{R}^C \text{ and } \|\mathbf{u}\|_2 = 1\right\}$

A simple probabilistic model

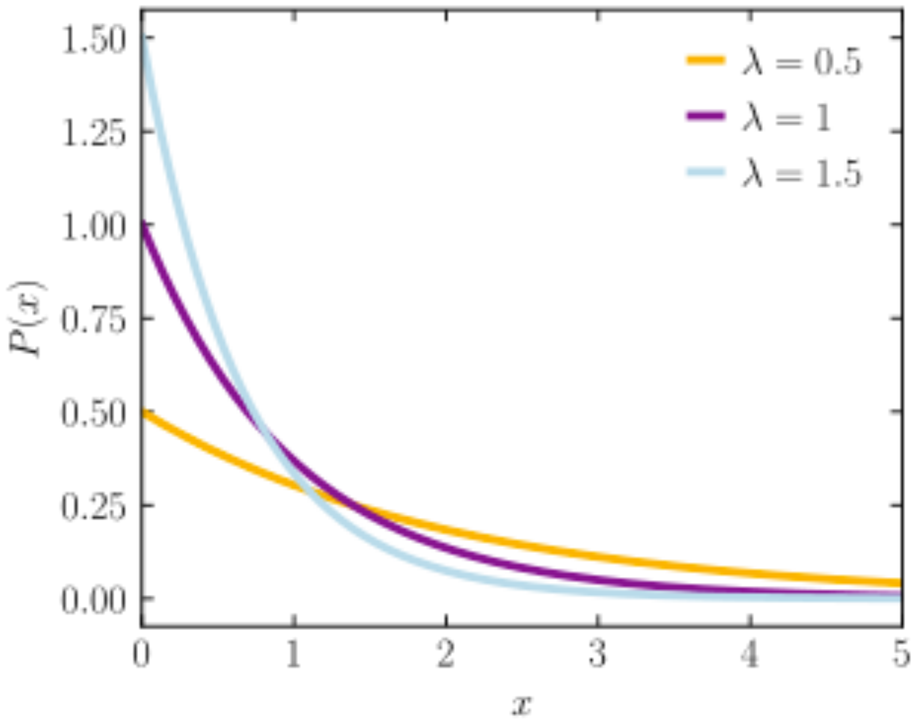
Prior on amplitudes

- To complete the model, we place an **exponential** prior on amplitudes,

$$a_{k,t} \sim \mathrm{Exp}(\lambda)$$

where λ is the inverse-scale (aka rate) parameter.

- It's pdf is $\mathrm{Exp}(x;\lambda) = \lambda e^{-\lambda x}$.
- As we will see, this prior will lead to **sparse** estimates.



https://en.wikipedia.org/wiki/Exponential_distribution

A simple probabilistic model

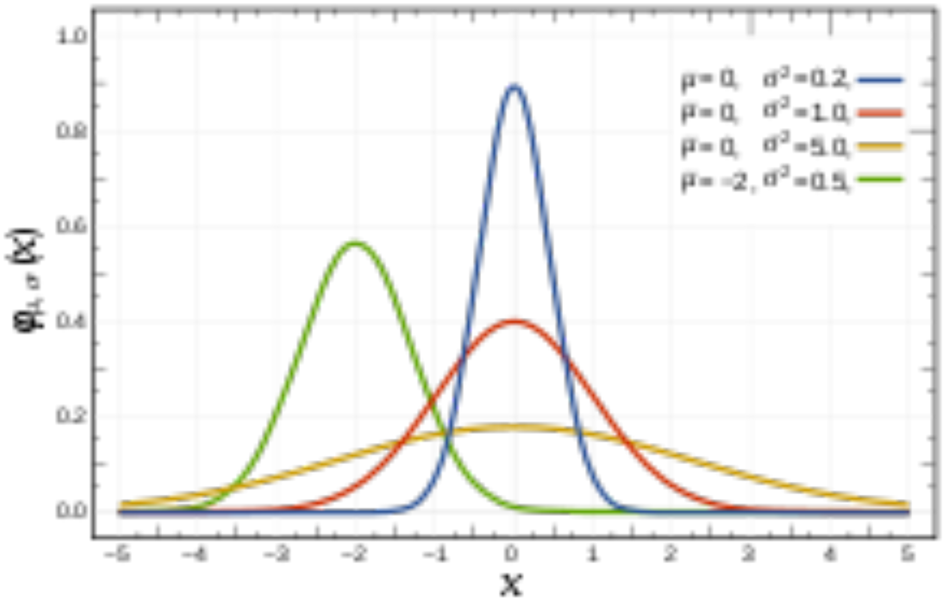
Noise model

- So far, $\mathbf{X} = \mathbf{W}\mathbf{A}^\top + \mathbf{E}$ where $\mathbf{E} = [[\epsilon_{c,t}]]$ is a matrix of "noise." How to model the noise?
- Simple assumption: $\epsilon_{c,t} \sim \mathcal{N}(0, \sigma^2)$ where

$$\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2\sigma^2}(x-\mu)^2 \right\}$$

  is the **Gaussian** or **normal distribution**.
- Linear transformations of Gaussians are still Gaussian!

$$x \sim \mathcal{N}(\mu, \sigma^2) \Rightarrow ax + b \sim \mathcal{N}(a\mu + b, a^2\sigma^2)\,.$$



https://en.wikipedia.org/wiki/Normal_distribution

A simple probabilistic model

The joint distribution

$$p(\mathbf{X}, \mathbf{W}, \mathbf{A}) = p(\mathbf{X} \mid \mathbf{W}, \mathbf{A})\, p(\mathbf{W})\, p(\mathbf{A})$$
$$=$$

This is called **semi-nonnegative matrix factorization (semi-NMF).**

Fitting the model

MAP estimation by coordinate ascent

- repeat until convergence:
  - for $k = 1, \ldots, K$:
    - Set $\mathbf{w}_k = \arg\max p(\mathbf{X}, \mathbf{W}, \mathbf{A})$ holding all else fixed
    - Set $\mathbf{a}_k = \arg\max p(\mathbf{X}, \mathbf{W}, \mathbf{A})$ holding all else fixed

Fitting the model

Optimizing the waveforms

Maximizing the joint probability wrt $\mathbf{w}_k$ is equivalent to maximizing the log joint probability,

$$\begin{aligned}
\log p(\mathbf{X}, \mathbf{W}, \mathbf{A}) &= \sum_{c=1}^{C} \sum_{t=1}^{T} \log \mathcal{N}\left(x_{c,t} \,\middle|\, \sum_{j=1}^{K} w_{j,c} a_{j,t}, \sigma^2 \right) \\
&= -\frac{1}{2\sigma^2} \sum_{c=1}^{C} \sum_{t=1}^{T} \left(x_{c,t} - \sum_{j=1}^{K} w_{j,c} a_{j,t} \right)^2 + c' \\
&= -\frac{1}{2\sigma^2} \sum_{c=1}^{C} \sum_{t=1}^{T} \left(r_{c,t} - w_{k,c} a_{k,t} \right)^2 + c'
\end{aligned}$$

where $r_{c,t} = x_{c,t} - \sum_{j \neq k} w_{j,c} a_{j,c}$ is the **residual**.

Fitting the model

Optimizing the waveforms

It's easier to solve in vector form. Let $\mathbf{r}_t = (r_{1,t}, \ldots, r_{C,t})$. Then,

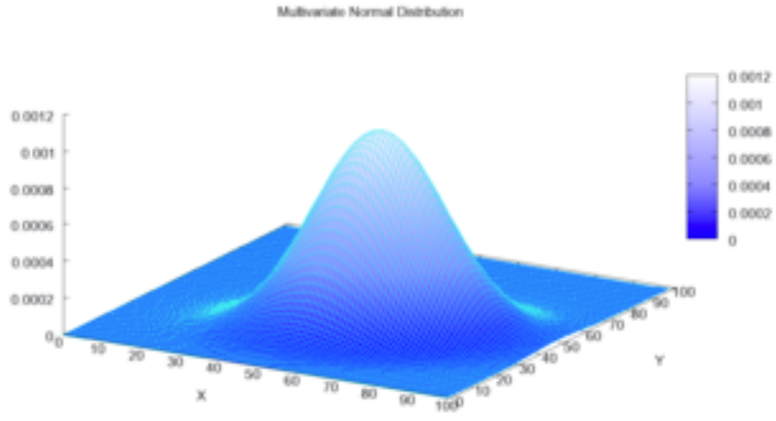
$$\log p(\mathbf{X}, \mathbf{W}, \mathbf{A}) = -\frac{1}{2\sigma^2} \sum_{t=1}^{T} (\mathbf{r}_t - \mathbf{w}_k a_{k,t})^\top (\mathbf{r}_t - \mathbf{w}_k a_{k,t}) + c'$$

$$= \sum_{t=1}^{T} \mathcal{N}(\mathbf{r}_t; \mathbf{w}_k a_{k,t}, \sigma^2 \mathbf{I}) + c'$$

where $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the **multivariate normal distribution**.

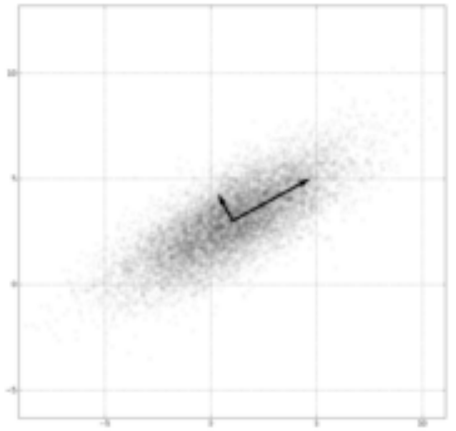
Fitting the model

The multivariate normal distribution



The multivariate normal density for $\mathbf{x} \in \mathbb{R}^D$ is,

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{D}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \right\}$$



where $\boldsymbol{\mu} \in \mathbb{R}^D$ is the **mean** and $\boldsymbol{\Sigma} \in \mathbb{R}^{D \times D}_{\succeq 0}$ is the (positive definite) **covariance matrix**.

When $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}$, we call it a **spherical Gaussian** distribution.

https://en.wikipedia.org/wiki/Multivariate_normal_distribution

Fitting the model

Optimizing the waveforms

Returning to the optimization

$$\begin{aligned}\log p(\mathbf{X}, \mathbf{W}, \mathbf{A}) &= \sum_{t=1}^{T} \mathcal{N}(\mathbf{r}_t; \mathbf{w}_k a_{k,t}, \sigma^2 \mathbf{I}) + c' \\ &= -\frac{1}{2\sigma^2} \sum_{t=1}^{T} (\mathbf{r}_t - \mathbf{w}_k a_{k,t})^\top (\mathbf{r}_t - \mathbf{w}_k a_{k,t}) + c' \\ &= \frac{1}{\sigma^2} \sum_{t=1}^{T} \left(\mathbf{r}_t^\top \mathbf{w}_k a_{k,t} - \frac{a_{k,t}^2}{2} \mathbf{w}_k^\top \mathbf{w}_k \right) + c''\end{aligned}$$

Note: $\mathbf{w}_k^\top \mathbf{w}_k = 1$ by the constraint $\mathbf{w}_k \in \mathbb{S}_{C-1}$.

Fitting the model

Optimizing the waveforms

$$
\begin{aligned}
\mathbf{w}_k^\star &= \arg \max_{\mathbf{w}_k \in \mathbb{S}_{C-1}} \left(\sum_{t=1}^{T} a_{k,t} \mathbf{r}_t \right)^\top \mathbf{w}_k \\
&= \arg \max_{\mathbf{w}_k \in \mathbb{S}_{C-1}} \left\langle \sum_{t=1}^{T} a_{k,t} \mathbf{r}_t, \mathbf{w}_k \right\rangle \\
&= \arg \max_{\mathbf{w}_k \in \mathbb{S}_{C-1}} \left\langle \mathbf{R}\mathbf{a}_k, \mathbf{w}_k \right\rangle \\
&\propto \mathbf{R}\mathbf{a}_k \,.
\end{aligned}
$$

where $\mathbf{R} \in \mathbb{R}^{C \times T}$ is the matrix of residuals with columns $[\mathbf{r}_1, \ldots, \mathbf{r}_T]$.

Fitting the model

Optimizing the amplitudes

As a function of $a_{k,t}$, the log joint probability is,

$$\log p(\mathbf{X}, \mathbf{W}, \mathbf{A}) = \frac{\mathbf{r}_t^\top \mathbf{w}_k a_{k,t}}{\sigma^2} - \frac{a_{k,t}^2}{2\sigma^2} - \lambda a_{k,t} + c'$$

This is a **quadratic optimization subject to a non-negativity constraint.**

Fitting the model

Generic solution

Assume $\alpha > 0$. Solve

$$\arg \max_{x \geq 0} \quad f(x) = -\frac{\alpha}{2}x^2 + \beta x + \gamma,$$

Fitting the model

Optimizing the amplitudes

By pattern matching to our problem, we have

$$a_{k,t}^{\star} = \max\left\{0, \sigma^2\left(\frac{\mathbf{r}_t^\top \mathbf{w}_k}{\sigma^2} - \lambda\right)\right\} = \max\left\{0, \mathbf{r}_t^\top \mathbf{w}_k - \lambda\sigma^2\right\}$$

$\mathbf{r}_t^\top \mathbf{w}_k$, is the **projection** of the residual onto the waveform for neuron k.

$\lambda\sigma^2$ the **threshold** that projection must exceed to designate a spike in amplitude.

The final algorithm

MAP estimation by coordinate ascent

- repeat until convergence:
  - for $k = 1, \ldots, K$:
    - Compute the residual $\mathbf{R} = \mathbf{X} - \sum_{j \neq k} \mathbf{w}_j \mathbf{a}_j^\top$
    - Set $\mathbf{w}_k \propto \mathbf{R}\mathbf{a}_k$
    - Set $\mathbf{a}_k = \max\{0, \mathbf{R}^\top \mathbf{w}_k - \lambda\sigma^2\}$

Note: You don't have to recompute the residual from scratch each iteration.

What did we learn?

- Some basic neurobiology. How action potentials (spikes in membrane potential) appear in extracellular voltage recordings (e.g. from neuropixels).
- We started with a simplifying assumption: downsample the data to ~500Hz, then model it with **semi-NMF**.
- More distributions! Uniform on a hypersphere, exponential, normal, multivariate normal.
- More practice with MAP estimation by coordinate ascent. Solving (scalar) quadratic optimization problems with inequality constraints.

Further reading

- For a (far) deeper introduction to neurobiology, check out the first few chapters of Luo 2020, “Principles of Neurobiology.” Computationally minded folks might also like the first chapters of Dayan and Abbott’s “Theoretical Neuroscience.”

- I don’t have a reference for this specific semi-NMF model, but check out the course notes online for references to other (probabilistic) matrix factorization models.

Next time

- We'll relax our downsampling assumption and build a similar model called **convolutional matrix factorization**, which is essentially what state-of-the-art spike sorting algorithms use.