Chapter 1

Introduction

- Problem statement
- Opportunity of voltage imaging, core idea of inverting the causal 'presynaptic spike → postsynaptic PSP' relation
- Overview of thesis

1.1 Connectomics

- Motivation
- Ground-truth connectomics: tracing of electron microscopy and fluoresent injection imaging
- Necessity of connection *inference*
- Mention 'invasive' connection testing (stimulate one cell, record possible neighbours)
- Limitations of 'connectomics', and of inferred vs 'actual' connectomics.
  Terminology, e.g. 'functional connectomics'

1.2 Voltage imaging

- Technologies (from dyes to GEVIs)
- Specs: cell yield, tissue depth, recording duration, SNR, species
- ..and growth of these over time, and comparison with calcium imaging.
  To extrapolate how these might advance in the future
- Comparison with other recording techniques: ephys, calcium imaging, (and briefly mention coarser methods)

1.3 Network inference

- Working with events/spikes only, versus working with continuous signals; or a hybrid as here.
- Overview of the spikes-only methods
- The connectomics competition, and the findings about the best methods
- Mention other application domains, like gene regulatory networks
- 'Spike-triggered voltage regression' of Zhou/Cai

Chapter 2

Simulation details

In this chapter, we decribe our experimental setup: the neuron model we simulate, its inputs, and how we simulate voltage imaging.

2.1 Neuron model

We choose to simulate the 'AdEx' neuron model, or the 'adaptive exponential integrate-and-fire' neuron.[GB09] This is a leaky-integrate-and-fire (LIF) neuron model, with two additions. First, the full upstroke of each spike is simulated, as an exponential runoff. Second, an extra dynamic variable is added: the adaptation current. This allows the simulation of many non-linear effects of real neurons, like spike-rate adaptation and post-inhibitory rebound. (Note however that we do not focus on any of these effects in this thesis).

The AdEx model consists of two differential equations, one to simulate the membrane voltage V, and one for the adaptation current w:

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = -g_L(V - E_L) + g_L\Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right) - I_{\text{syn}} - w \tag{2.1}$$

$$\tau_w\frac{\mathrm{d}w}{\mathrm{d}t} = a(V - E_L) - w \tag{2.2}$$

I_{syn} is the synaptic current, explained in section 2.1. We use the sign convention of inter alia Dayan & Abbott ([DA01], ch. 5.3, p. 162) where membrane currents are defined as positive when positive charges flow *out* of the cell. I.e. a positive I_{syn} decreases the membrane voltage (itself defined as the electric potential inside minus outside the cell).

We solve these equations using first-order (Euler) integration.

In addition to the two differential equations, the AdEx model also consists of an instantaneous reset condition. When the membrane voltage V reaches a certain threshold, a spike is recorded, V is reset, and w is increased:

$$\text{if } V > \theta \text{ then: } V \leftarrow V_r \tag{2.3}$$

$$w \leftarrow w + \Delta w \tag{2.4}$$

[explain parameters]

Alternative neuron models

Why did we choose the AdEx model to simulate neuron voltages? In short, because it strikes a good balance between realism and complexity. (Complexity, as in: still having a compact set of free parameters). We discuss here briefly two alternative neuron models: the simple leaky-integrate-and-fire (LIF) neuron, and the more complex Hodgkin-Huxley (HH) neuron.

A simpler model than AdEx would be the well-known LIF neuron:

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = -g_L(V - E_L) - I_{\text{syn}}$$

$$\text{if } V > \theta \text{ then: } V \leftarrow V_r$$

As is apparent from comparing with equations (2.1) and (2.4), the AdEx model is an extension of the LIF model. The LIF neuron lacks a simulation of the upstroke of spikes (the exponential term in equation (2.1)), and the slower time-scale adaptation current (equation (2.2)), which allows the simulation of many qualitatively different real neuron types. It is especially this first addition, the full upstroke simulation, that seems relevant in generating realistic voltage traces.

Would this thesis have been very different had we used LIF neurons instead? Probably not, though it might depend on the mean voltage level of the simulated neuron: if it is well below the firing threshold, both LIF and AdEx are linear (the exponential term is negligible), and they behave quasi identically. When a spike is generated in the AdEx model, the exponential feedback makes the upstroke very fast, and thus not many timesteps in the simulation are spent on it, versus the linear regime. [Reference the Vdot-V, dynamical system fig]

On the other hand, when the neuron would continuously teeter just below its firing threshold, the LIF and AdEx models do not behave similarly. LIF's $\dot{V}(V)$ curve is still fully linear, while AdEx's is not, and AdEx will behave more like a real neuron. [Reference fig with real neuron Vdot-V curve]

Another well-known alternative neuron model is the class of Hodgin-Huxley (HH)-like neurons. These models simulate the full trajectory of a spike: both its upstroke and its downstroke. Unfortunately they also have many free parameters. They also take a bit longer to simulate, being higher dimensional (having more differential

equations), and containing many more exponential terms, which take the brunt of the time when numerically evaluating a differential expression.

Comparison between the AdEx and Izhikevich neurons

These models are very similar. Their phase spaces are topologically identical: the adaptive current equation is identical (up to a renaming of the variables); and the $\dot{V}(V)$-graph has the same shape, with two fixed points: a stable fixed point at the resting potential, and an unstable one at the firing threshold.

They differ in the exact shape: Izhikevich's $\dot{V}(V)$ is a parabola, while AdEx is the more realistic 'linear subthreshold and then transitioning to an exponential'. [ref bio fig] As a result, Izhikevich neurons have an unrealistically slow spike upstroke.

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = (V - E_L)(V - E_T) - I_{\text{syn}} - w \tag{2.5}$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = a(V - E_L) - w \tag{2.6}$$

[equation with original names]

[comparison table: from nb]

Synapse model

One unexplained term in our neuron model equation (2.1) is the synaptic current, I_{syn}. This is the following sum over all input synapses i of the neuron:

$$I_{\text{syn}} = \sum_i g_i(V - E_i) \tag{2.7}$$

where V is the global membrane voltage of the neuron, E_i is the reversal potential of that synapse, and g_i is the local synaptic conductance, which is modulated by presynaptic spikes.

For an excitatory synapse, $V < E_i$, making $g_i(V - E_i)$ negative, increasing the membrane voltage according to the sign convention for I_{syn} in equation (2.1).

We simulate the synaptic conductances g_i as exponentially decaying signals (with time constant τ), and bump them up instantaneously on arrival of a presynaptic spike:

$$\frac{\mathrm{d}g_i}{\mathrm{d}t} = -g_i/\tau \tag{2.8}$$

On incoming presynaptic spike:

$$g_i \leftarrow g_i + \Delta g_i \tag{2.9}$$

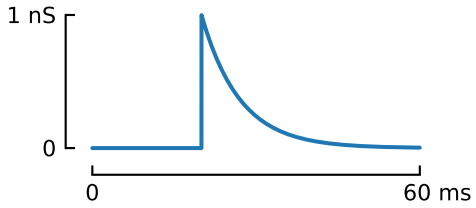

Figure 2.1: Example synaptic conductance trace $g_1(t)$, with a single incoming spike at $t = 20$ ms.

Note that these are not the so called alpha-synapses. Those are two dimensional and also have an exponential rise, instead of just an exponential decay. (For an infinitely fast rise though, these models are of course the same). Simulating a full alpha synapse might increase the realism of our voltage traces, for a small simulation cost. We did not try this however. Foremost because alpha synapses fit to real data often have very fast rise times that are almost indistinguishable from instantaneous jumps.

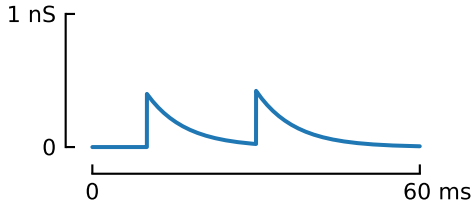

Figure 2.2: Another example trace $g_2(t)$, with spikes at $t = 10$ ms and 30 ms, and a smaller Δg.

For efficiency, we give all our excitatory synapses the same reversal potential, E_{exc}. Idem for the inhibitory synapses, with E_{inh}. This allows us to factor the synaptic current sum (equation (2.7)) as follows:

$$I_{\text{syn}} = (V - E_{\text{exc}}) \sum_{\text{exc } i} g_i \; + \; (V - E_{\text{inh}}) \sum_{\text{inh } i} g_i \tag{2.10}$$

The sums of conductance signals $g_i(t)$ can also be simplified. Say that the values of g_i at $t = 0$ are G_i. The solution to equation (2.8) (at least in the time until a new presynaptic spike arrives) is then

$$g_i(t) = G_i \, e^{-t/\tau} \tag{2.11}$$

With this, and when all synapses have the same time constant τ, the two sums in equation (2.10) can be factored as follows:[1]

$$\sum_i g_i(t) = \sum_i \left(G_i \, e^{-t/\tau} \right) = \left(\sum_i G_i \right) e^{-t/\tau} \tag{2.12}$$

[1] This is only valid in the time before any new spikes arrive. To see that the 'summability' still holds after a new spike arrives, the above reasoning can be repeated, but simply with different values for the G_i (all decayed by an amount $e^{-t_{\text{spike}}/\tau}$, and one increased by a bump Δg_i), and then redefining t_{spike} to be $t = 0$.

This means that we need to only keep track of two conductance signals: g_{exc} and g_{inh}, each the sum of all excitatory or all inhibitory synaptic conductances.

Our synaptic current sum then becomes simply:

$$I_{\text{syn}} = (V - E_{\text{exc}})\, g_{\text{exc}} \; + \; (V - E_{\text{inh}})\, g_{\text{inh}}, \tag{2.13}$$

and we only need to simulate two differential equations, instead of one for every synapse:

$$\frac{\mathrm{d}g_{\text{exc}}}{\mathrm{d}t} = -g_{\text{exc}}/\tau$$
$$\frac{\mathrm{d}g_{\text{inh}}}{\mathrm{d}t} = -g_{\text{inh}}/\tau,$$

where on arrival of a spike at synapse i either g_{exc} or g_{inh} is instantaneously increased by a value Δg_i, depending on whether that synapse is excitatory or inhibitory.

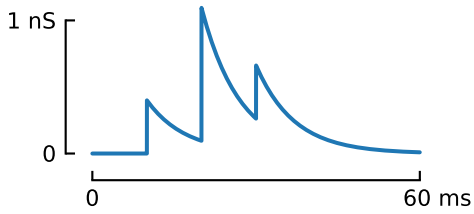

Figure 2.3: A third synaptic conductance trace $g_3(t)$, with three input spikes at the same times and strengths as in figures 2.1 and 2.2. This signal is simulated independently, but turns out to be equal to the sum of the two others: $g_3(t) \equiv g_1(t) + g_2(t)$.

2.2 Input spikes

In our simplest experimental setup, we simulate just one AdEx neuron. Its input is provided by an array of N Poisson neurons, i.e. they each generate spike trains according to a Poisson process. We call this the 'N-to-1' setup.

The inter-event intervals of a Poisson process follow an exponential distribution. We use that fact to generate spike trains: we draw samples from $\mathrm{Exp}(\lambda)$ (with λ the desired firing rate), and cumulatively sum up these intervals to obtain spike times. This is done until we have reached the desired input train duration.

2.3 Voltage imaging

The signals detected by a light microscope in a voltage imaging setup are not the same as the real membrane voltage signals of which they are a reflection.

We model this lossy transformation by simply adding Gaussian noise to our simulated membrane voltage. As in the voltage imaging literature, we quantify the amount of this noise by a 'spike-SNR' measure (spike signal-to-noise ratio). This is defined as the height of an average spike relative to the standard deviation of the noise.

A more realistic model of the voltage-imaging transformation would also incorporate the exponential decay over time of the SNR, and the short-term 'smearing in time' of voltage indicators. The latter could be done by passing the voltage signal through a linear filter with some non-instantaneous impulse response.

Chapter 3

Spike-triggered averaging

- Principle, example STAs
- Influence on STA of E/I balance, output firing rate, reversal potentials
- Use as connection test: shuffled spike trains and height of STA, p-values, and the 'area-over-start' heuristic for E vs I classification.
- Evaluation of a connection test: 'ternary' classification, summary measures, AUROC
- Performance of the simple STA-height test, for different N
- Influence of window length

Chapter 4

New connection inference methods

4.1 STA Template correlation

- Idea for the two-pass test
- Template examples. Both ideal and template found with first-pass (STA height-only)
- Peformance for different N, & comparison with previous method

4.2 Linear regression of the upstroke

- Non-STA method: concatenated individual windows as (X, y)
- Examples of pooled windows, and fits
- Mention the problem of unknown transmission delays
- Perfomance for different N, & comparison with previous methods

4.3 Fitting a full STA model

- Model design
- Iterative model fitting
- Problem of overfitting, and parameter-constraints to solve it
- An advantage: fit parameters (like transmission delay and time constants) are biologically meaningful
- Perfomance for different N, & comparison with previous methods

4.4 Clustering, & Hierarchical model fitting

- (Time-permitting)

4.5 Zhou/Cai's 'Spike-triggered regression'

- (Time-permitting)

4.6 Computational cost

- Timings of each method, extrapolation for larger number of tested connections

4.7 Summary

- Conclusions of the N-to-1 experiment
- Leadup to the network experiments: what we could not yet test (the problem of indirect connections, as e.g. identified in the connectomics challenge)

Chapter 5

Network model

- Network connectivity, E/I balance, raster plots
- Too many possible connections to test them all → Subsampling
- Performance of last chapter's methods
- If time: experiment with a network that is less densely connected than our current fully-random one. Why? To better examine the effect of indirect connections / colliders (For the current connectivity, there are too many of those. But in a more realistic, 'localized' network, there are less, and so it seems easier to isolate and examine their effect).

Chapter 6

Discussion

6.1 Summary & conclusions

..

6.2 Future work

- Test on real data
- Direct comparison with spikes-only methods
- More complexity in the testing setup: different transmission delays and time constants per synapse / neuron, plus:
- Short term synaptic plasticity. Bursting. Oscillations.
- Simulate different brain areas (different cell types and connectivity patterns). Simulate the same area, but in different states (up vs down, e.g.)
- New connection test method to try: something deep learning-based (we have infinite training data, given our simulation)