Neural Computation (VS 265), Problem Set 2

Due date: September 22, 3:30pm

Fall 2022

General guidelines:

- We are grading problem sets anonymously. **Include your student ID in the submission, but do not include your name.**
- You may work in small groups of 2-3. Note that you are responsible for writing up and submitting your submission individually.
- You are expected to attach any code you used for this assignment but will be evaluated primarily on the writeup.

Part 1: Eigengrau

What determines the lowest light level you can reliably see? There are multiple factors responsible, but the principal one is the spontaneous activation of rhodopsin molecules that occurs in total darkness, which in turn leads to rod hyperpolarization and a change in glutamate release that is indistinguishable from the capture of a photon. The rate of these spontaneous activations is roughly 1 event per 160 seconds per rod. This scintillating background of rod activity is also known as the Eigengrau, or "dark light," as subjects report perceiving it as a field of gray or sometimes shimmering dots when in total darkness. In order to reliably detect real light coming from the outside world, the arrival rate of photons must be sufficiently high to compete with this background level of visual noise. For human observers, this has been measured to be at a rate of 1 photon catch per 5000 seconds per rod - i.e., far lower than the spontaneous activation rate! How can this be?

i. First, to get an intuitive feel for what this means, create a simulated movie of a 1 deg. x 1 deg. patch of retina, or a roughly 300 x 300 array of rods, spontaneously activating in total darkness. To make it somewhat realistic, assume each rod is a leaky integrator with $\tau = 0.5$ sec, so that the response to a photon lasts about half a second.

   Hint: Remember that you can simulate a leaky integrator via $y_{t+1} = (1 - \alpha)y_t + \alpha x_t$, where y_t is the rod response at time t and $x_t = \frac{1}{\Delta t}$ if a rhodopsin molecule is activated at time t and zero otherwise, and $\alpha = \Delta t/\tau$, with Δt being the step size of the simulation. You may want to pick Δt according to the frame rate you will use to play the movie — e.g., for 30 frames/sec use $\Delta t = 0.033$.

ii. Now, create a second movie of a set of rods activating in response to photons only (assuming no spontaneous activity), at 1 photon/5000 sec/rod. Then, create a third movie of this photon catch rate combined with the background spontaneous rate (i.e., a rate of [1/160 + 1/5000] per second per rod. Place all three movies side by side so you can visually compare them as they play. Can you detect a difference between movies 1 and 3?

iii. Finally, simulate an ideal observer (a.k.a. "homunculus") inside your brain, comparing movies 1 and 3 created above — akin to comparing two spatial patches within the retina — to determine which one contains a flash of light. Let us assume the ideal observer estimates the event rate in each patch by counting the total number of events occurring within some spatial area over some interval of time — fixed to be equal for both patches — and that it reports seeing light in whichever patch has the highest total count. If the spatial area and temporal interval are too small, these estimates will be noisy and its answers will be near chance. As the amount of spatial and temporal integration is increased, the estimates will become less noisy and thus its answers will become more accurate. At psychophysical threshold — when then the observer's responses are correct 75% of the time — the rates for "light" vs. "no light" are [1/160 + 1/5000] vs. [1/160], or just a 3% difference. Thus, the system must integrate over a sufficient spatial and temporal extent such that the total count leads to a correct answer 75% of the time (as opposed to guessing, which would correspond to being correct 50% of the time). **Find a plausible combination of spatial area and temporal interval that meets this requirement.**

Hint: Here is some math that may save you some hunting and pecking: 75% correct on such a two-alternative forced-choice task corresponds to d'=1, which is when the difference between means equals the standard deviation of the two distributions being discriminated - see **[DA]** section 3.2. For a Poisson process (which this is), the count distribution will be more or less Gaussian, with variance equal to the mean. Putting these two together, we want to find the count n that yields a 3% difference with $d' = 1$, or where $0.03n = \sqrt{n}$. So the spatial and temporal integration must be large enough to reach this value of n.

Part 2: LIF Encoding/Decoding of Sensory Signal

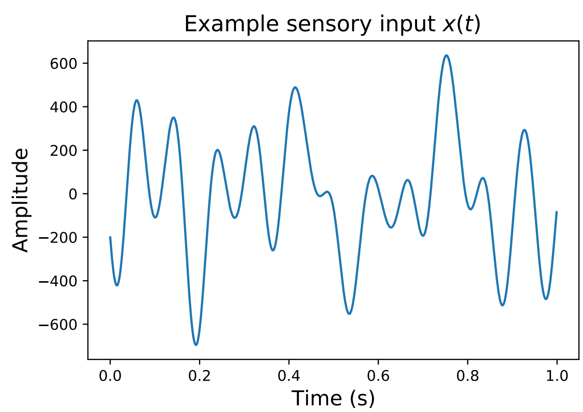

Figure 1: Continuous sensory input.

i. Drawing upon the methods described in Neural Engineering Chapter 4, show how the above time-varying signal can be nonlinearly encoded into spikes from a leaky-integrate-and-fire (LIF) neuron. For the LIF model you may wish to build upon the membrane equation from Problem Set 1, Part 1 with a current source as input, but now adding on a threshold and spike generating and reset mechanism. Use two LIF neurons: one neuron will be an on-type cell, and the other an off-type cell. Simulate these neurons, and plot their spikes with the signal. Also report the total number of spikes emitted from the on-cell, and the off-cell.

Hint: The `plt.eventplot` function from `matplotlib` is useful for plotting spikes.

Here are the parameters you should use for each LIF neuron:

$$\begin{aligned}
V_0 = V_r &= -70 \text{ mV} \quad \text{(initial/resting membrane voltage)} \\
V_{thresh} &= -55 \text{ mV} \quad \text{(spiking threshold voltage)} \\
C_m &= 50 \text{ pF} \quad \text{(membrane capacitance)} \\
G_{leak} &= 5 \text{ nS} \quad \text{(leak conductance)} \\
t_{rest} &= 5 \text{ ms} \quad \text{(post-spike refactory time)}
\end{aligned}$$

ii. Next, show how LIF encoded signal can be decoded by downstream post-synaptic processes. The basic methods for deriving a decoding filter and signal reconstruction are described in equations 4.19 and 4.23 of Neural Engineering Chapter 4, but you may wish to consider other methods as well. Plot the decoded signal on top of the plot from the previous section as a dotted line.

iii. Finally, report the root mean squared error (RMSE) between the sensory signal, and the reconstruction.

For reference, the equation for RMSE between two signals is:

$$\text{RMSE}(x(t), y(t)) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left[x(t) - y(t)\right]^2}$$

where T is the number of time steps for $x(t), y(t)$.