SYDE 556/750
Simulating Neurobiological Systems
Assignment 3

Due date: November 6, 2023

Important Notes – Please Read Carefully

- This assignment is worth 15 marks (15% of the final grade). The number of marks for each question is indicated in brackets to the left of each question.
- Please use Python 3 along with the `numpy` and `matplotlib` libraries to solve these assignments. In particular, fill out this Jupyter Notebook template:

  `https://github.com/celiasmith/syde556-f23/blob/master/assignments/assignment_03/syde556_assignment_03_template.ipynb`

- Clearly label any plot you produce, including the axes. Provide a legend if there are multiple graphs in the same plot.
- You won't be judged on the quality of your code, but writing reusable functions will greatly simplify this and future assignments.
- Make sure to execute the Jupyter command "Restart Kernel and Run All Cells" before submitting your solutions. You will lose marks if your code fails to run or produces results that differ significantly from what you've submitted.
- Rename the completed notebook to `syde556_assignment_03_<STUDENT ID>.ipynb` and submit it to the TA (Ben Masters <bpmasters@uwaterloo.ca>). The deadline is at `23:59 EST` on November 6, 2023.
- There is a late penalty of one mark per day late. Please contact celiasmith@uwaterloo.ca if there are extenuating circumstances.
- **Do not use or refer to any code from Nengo!**

1 Decoding from a population

As you did in previous assignments, make a population of 20 LIF neurons representing a 1-dimensional value with radius $r = 2$, and compute a decoder for them. For parameters, $\tau_{\text{ref}} = 2\,\text{ms}$, $\tau_{\text{RC}} = 20\,\text{ms}$, the maximum firing rates are chosen randomly from a uniform distribution between 100 and 200 Hz (maximum rate at $\langle \mathbf{x}, \mathbf{e} \rangle = r = 2$), and the x-intercepts are chosen randomly from a uniform distribution between -2 and 2. Remember that the α and J^{bias} terms are computed based on these x-intercepts and maximum firing rates.

It is generally easiest to compute decoders using the original method from Assignment 1, where we use the rate-mode approximation for the neurons to generate the A matrix, then find $\mathbf{D}^T = (\mathbf{A}\mathbf{A}^T + N\sigma^2\mathbf{I})^{-1}\mathbf{A}\mathbf{X}^T$. These decoders should work even when you simulate the neurons in terms of spikes (from question 2 on). However, you have to scale the output depending on the time step `dt`.

Use this method for computing decoders for the whole assignment.

[0.5] a) *Tuning curves.* Plot the tuning curves (firing rate of each neuron for different x values between -2 and 2).

[1.0] b) *Decoder and error computation.* Compute the decoders and plot $(x - \hat{x})$. When computing decoders, take into account noise ($\sigma = 0.1 \cdot 200\,\text{Hz}$). When computing $\hat{x}$, add random Gaussian noise with $\sigma = 0.1 \cdot 200\,\text{Hz}$ to the activity. Report the Root Mean-Squared Error (RMSE).

2 Decoding from two spiking neurons

Choose a neuron from part 1 that has a firing rate of somewhere between 20 Hz to 50 Hz for $x = 0$. Using that neuron's α and J^{bias} value, construct two neurons: both with the same α and J^{bias}, but one with $e = 1$ and the other with $e = -1$. With the function from the last assignment, generate a random input $x(t)$ that is 1 s long, with `rms` $= 1$, `dt` $= 1\,\text{ms}$, and an upper limit of 5 Hz. Feed that signal into the two neurons and generate spikes. Decode the spikes back into $\hat{x}(t)$ using a post-synaptic current filter $h(t)$ with a time constant of $\tau = 5\,\text{ms}$.

[0.5] a) *Synaptic filter.* Plot the post-synaptic current

$$h(t) = \begin{cases} 0 & \text{if } t < 0\,, \\ \frac{1}{\tau} e^{-t/\tau} & \text{otherwise}\,. \end{cases}$$

[1.5] b) *Decoding using a synaptic filter.* Plot the original signal $x(t)$, the spikes, and the decoded $\hat{x}(t)$ all on the same graph.

[0.5] c) *Error analysis.* Compute the RMSE of the decoding.

3 Decoding from many neurons

Repeat question 2, but with more neurons. Instead of picking particular neurons, randomly generate them with x-intercepts uniformly distributed between -2 and 2 and with maximum firing rates between 100 and 200 Hz. Randomly choose encoder values to be either -1 or $+1$.

[1.5] a) *Exploring the error for an increasing neuron count.* Plot the Root Mean-Squared Error as the number of neurons increases, on a log-log plot. Try 8 neurons, 16 neurons, 32, 64, 128, up to 256. For the RMSE for a particular number of neurons, average over at least 5 randomly generated groups of neurons. For each group of neurons, randomly generate the signal $x(t)$. Use the same parameters as in question 2.

- 📌 The RMSE should go down as the number of neurons increases.
- 📖 This should be similar to Figure 5.3 in the book.

[0.5] b) *Discussion.* Discuss your results. What is the systematic relationship between the neuron count and the error?

4 Connecting two groups of neurons

For this question, use two groups of neurons with radius $r = 1$ and intercepts in $[-1, 1]$ to compute $y = 2x+1$. The first group of neurons will represent x and the second group will represent y.

Start by computing decoders. You will need two decoders: one to decode $f(x) = 2x + 1$ from the first population, and one to decode $f(y) = y$ (the standard representational decoder) from the second population.

Use the same neuron parameters as for previous questions (other than the radius and the intercepts), and use 200 randomly generated neurons in each population.

[2.0] a) *Computing a function.* Show the behaviour of the system with an input of $x(t) = t - 1$ for 1 s (a linear ramp from -1 to 0). Plot the ideal $x(t)$ and $y(t)$ values, along with $\hat{y}(t)$.

- 📌 Note that you should use the decoders that work for any input over the range of intercepts: do not re-compute decoders for any particular input (i.e., set of x values). Input $x(t)$ into the first group of neurons and produce spikes. Decode from those spikes using the decoder for $f(x) = 2x + 1$. Input that decoded result into the second group of neurons to produce spikes. Use the second decoder ($f(y) = y$) to decode $\hat{y}(t)$.
- 📌 Make sure the maximum firing rates are now at -1 or 1 (i.e., the radius is 1).

[0.5] b) *Step input.* Repeat part (a) with an input that is ten randomly chosen values between -1 and 0, each one held for 0.1 seconds (a randomly varying step input)

[0.5] c) *Sinusoidal input.* Repeat part (a) with an input that is $x(t) = 0.2 \sin(6\pi t)$.

[1.0] d) *Discussion.* Briefly discuss the results for this question. Does the output match the ideal output? What kind of deviations do you see and why do those exist?

5 Connecting three groups of neurons

For this question, use three groups of neurons with intercepts from $[-1, 1]$ to compute $z = 2y + 0.5x$. Follow the same steps as question 4, but take the decoded outputs from the first two groups of neurons ($f(y) = 2y$ and $f(x) = 0.5x$), add them together, and feed that into the third group of neurons.

[1.5] a) *Sinusoidal input.* Plot $x(t)$, $y(t)$, the ideal $z(t)$, and the decoded $\hat{z}(t)$ for an input of $x(t) = \cos(3\pi t)$ and $y(t) = 0.5\sin(2\pi t)$ (over 1 s).

[0.5] b) *Random input.* Plot $x(t)$, $y(t)$, the ideal $z(t)$, and the decoded $\hat{z}(t)$ for a random input over 1 s. For $x(t)$ use a random signal with a limit of 8 Hz and `rms` $= 1$. For $y(t)$ use a random signal with a limit of 5 Hz and `rms` $= 0.5$.

6 Computing with vectors

Do the same thing as questions 4 and 5, but with 2-dimensional vectors instead of scalars. Everything else is the same. For your encoders e, randomly generate them over the unit circle.

The function to compute is $w = x - 3y + 2z - 2q$. This requires five groups of neurons: x, y, z, q, and w. Each of them represents a 2-dimensional value. The outputs from x, y, z, and q all feed into w.

[2] a) *Constant inputs.* Plot the decoded output $\hat{w}(t)$ and the ideal w for

$$x = (0.5, 1), \qquad y = (0.1, 0.3), \qquad z = (0.2, 0.1), \qquad q = (0.4, -0.2)\,.$$

📌 Note that these are all constants, so they don't change over time. However, still plot the results for 1 s on one or more 2D graphs.

[0.5] b) *Sinusoidal input.* Produce the same plot for

$$x = (0.5, 1), \qquad y = (\sin(4\pi t), 0.3), \qquad z = (0.2, 0.1), \qquad q = (\sin(4\pi t), -0.2)\,.$$

[0.5] c) *Discussion.* Describe your results and discuss why and how they stray from the expected answer.