Learning objectives and goals:

In this problem set, we will study simplified linear recurrent neural networks with a symmetric connectivity matrix. This exercise is not only and opportunity to understand their properties as dynamical systems but more importantly to illustrate general principles that go beyond this specific case. The PSET will take you to explore these properties by doing analytical work and performing numerical simulations in MATLAB. By the end of the problem set we will study the conditions under which these networks can act as neuronal integrators, and thus represent simplified models of how short-term memory arises in neuronal circuits.

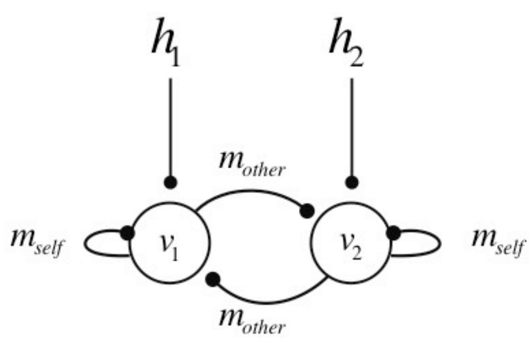
The topics of this PSET were covered in lecture 18 and will be covered in recitation 12.

This problem provides a great way to put all the linear algebra we have learned so far to work and relate those concepts back to ordinary linear differential equations that we studied during the first third of this course. By the end of this PSET, you should be able to:

- Decompose the network into eigen modes.
- Compute gain factors and effective time constants
- Predict the response of the network to different input vectors.
- Plot state-space trajectories for different input vectors.
- Implement these networks in MATLAB and compute solutions numerically, by using the Euler integration scheme.

Introduction:

Many neural networks consist of 2 populations of neurons whose effective connectivity is such that they provide self-excitation to neurons in their own population and inhibit neurons in the other population. A simple case of these circuits occurs when the connections are symmetric as in the following diagram:



For simplicity, we will treat each of these populations as a single neuron. Here, m_{self} gives the connection between each neuron and itself and m_{other} gives the connection between the different neurons. In this problem, we will find the eigenvalues and eigenvectors of this circuit in terms of m_{self} and m_{other}, and then apply our findings to understand specific cases.

Part 1: Decomposing the network into eigen modes

The recurrent connectivity matrix **M** of the above network, is a **special** case of symmetric matrices. The eigenvectors ($\widehat{f_i}$) of this matrix are the columns of a rotation matrix (Φ) at 45 degrees counter clockwise:

$$\Phi(45^{\circ}) = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

Note again the symmetry of the network and recall that the eigenvectors of **M** are patterns of inputs, with the **remarkable** property that the steady state outputs of the network will be a scaled version of these same patterns. With this information:

1. Write down the equations describing this network. Assume that each of the neurons has an intrinsic time constant τ_n. this means that with no recurrent connections, the activity of these neurons decays to zero with time constant τ_n. First, write separate equations for dv_1/dt and dv_2/dt. Then, using matrix notation, write a single equation that describes the entire network in terms of the firing rate vector $\vec{v} = (v_1, v_2)$, an input vector $\vec{h} = (h_1, h_2)$, and a recurrent connectivity matrix **M**.
2. Write expressions for eigenvalues λ_1 and λ_2 of the connectivity matrix, as a function of m_{self} and m_{other}. Also, show that the eigenvalue-eigenvector ($\mathbf{M}\widehat{f_i} = \lambda_i \widehat{f_i}$) equation is satisfied for each eigenvector.
3. Rewrite the input vector ($\vec{h}$) as a linear combination of the eigenvectors (i.e. $\vec{h} = b_1\hat{f}_1 + b_2\hat{f}_2$). Show that the coefficient $b_1 = h_{common}$, where $h_{common} = \frac{\sqrt{2}}{2}(h_1 + h_2)$. Also, show that $b_2 = h_{diff}$, where $h_{diff} = \frac{\sqrt{2}}{2}(h_2 - h_1)$.

Part 2: Simple model of selective amplification of differences between inputs.

Suppose that each neuron excites itself by setting m_{self} to 0.2 and inhibits the other neuron (m_{other} = -0.7). Furthermore, assume that this network receives the following input vector, $\vec{h}(t \geq 0) = (117, 123)\text{Hz}$. With this information:

1. Calculate the eigenvalues of the network.
2. What will happen to inputs that are common to the two cells? (Will they be amplified or attenuated?) Determine this by looking at the eigenvalue for the appropriate eigenvector.
3. What will happen to inputs that are opposite for the two cells (amplified or attenuated)?
4. Calculate the gain factors $1/(1-\lambda)$ for each mode (eigenvector) of the network, using the values in part (c) above.

5. If each cell in the network has an intrinsic time constant τ = 18 ms, what will be the corresponding time constants τ_{eff} for each mode? Does the amplified mode change more, or less rapidly, than the attenuated mode?
6. For each mode write down a differential equation describing the response of mode activity (c_1 and c_2) to an input vector $\vec{h}$.
7. Write the input vector in the form $\vec{h} = b_1\hat{f}_1 + b_2\hat{f}_2$. What are the coefficients b_1 and b_2?
8. Find the steady state activity of the 2 neurons $\vec{v}_\infty = (v_{\infty,1}, v_{\infty,2})$.
9. Sketch the state-space trajectory of the firing rates $\vec{v} = (v_1, v_2)$ as they approach this steady state value. Assume the network starts at zero firing rate, and that the input is turned on at t=0. On the same set of axes plot the eigenvectors $\hat{f}_1, \hat{f}_2$.
10. Simulate the network by numerically solving the equations with the Euler method[1], which is simple to implement in vector notation. Do not use exponential Euler, as this is more difficult to implement. Set the initial condition to $\vec{v}(t=0) = \vec{0}$, and the input vector to $\vec{h}(t \geq 0) = (117,123)\text{Hz}$. Integrate for 1 second using an integration time-step of 10^{-4} seconds.
11. Plot v_1 and v_2 as a function of time to confirm that your simulation works properly by checking that you get the same steady state solution you found analytically.

Part 3: Simple model of neuronal integration.

1. Modify the value of m_{other} so that that one of the **modes** of the network integrates a quantity proportional to the input difference (i.e. $h_2 - h_1$).
2. What is the condition on m_{self} and m_{other} for this to occur?
3. Demonstrate by numerical integration that you observe persistent activity even after h_1 and h_2 are turned off. Do this by considering the following input vector:

$$\vec{h}(t) = \begin{cases} \vec{0}\ \text{Hz} & \text{if } t < 0 \\ (117,123)\ \text{Hz} & \text{if } 0 \leq t < 0.8\text{s} \\ \vec{0}\ \text{Hz} & \text{if } t \geq 0.8\text{s} \end{cases}$$

   Continue to use the Euler method with the previously specified initial condition, simulation length and integration time-step.

   To demonstrate persistent activity, make a figure with 2 panels. On the upper panel plot v_1 and v_2 as a function of time. On the lower panel plot the modes activity (c_1 and c_2) as a function of time. Remember that $\vec{c} = \Phi^T \vec{v}$.
4. Use numerical simulations and relevant plots to show that this behavior is different from the network you simulated in part 2 questions 3 and 4.

[1] Have a look at EulerMethod.pdf for a refresher on numerical integration.