Theoretical Neuroscience II
Exercise 1: Linear recurrent network

Jochen Braun & Ehsan Kakaei

Due date: 20 April 2018

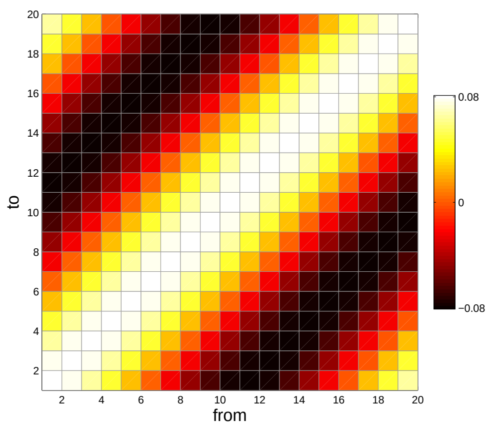
A linear recurrent network

Investigate the response of a linear recurrent network with $N = 20$ populations, indexed by $k \in \{1, \ldots, N\}$.

$$\frac{d\boldsymbol{v}}{dt} = -\boldsymbol{v} + \boldsymbol{h} + \boldsymbol{M} \cdot \boldsymbol{v}$$

to different input vectors $\boldsymbol{h}_i$. Generate input vectors by setting two adjacent elements to h and all other elements to zero (*e.g.*, $\boldsymbol{h}_1 = (h, h, \ldots, 0)$, $\boldsymbol{h}_2 = (0, 0, h, h, \ldots, 0)$, $\boldsymbol{h}_3 = (0, 0, 0, 0, h, h, \ldots, 0)$, etc. Choose the value of h randomly from the interval $h \in [1, 3]$.

Obtain the $N \times N$ connectivity matrix $\boldsymbol{M}$ from the Matlab function **ConnectionMatrix.m** provided. Note that this function also supplies two principal eigenvectors and eigenvalues.



Task A

Compare two alternative ways of visualising population activity!

First, visualise the two principal eigenvectors $\boldsymbol{e}_{1,2}$ together with some input vectors $\boldsymbol{h}_i$ by plotting their respective elements as a function of population index k.

Second, plot the *projection* of these vectors into the plane of principal eigenvectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$. For each vector $\boldsymbol{h}_i$, compute the length of the projections as dot products $x_1 = \boldsymbol{e}_1 \cdot \boldsymbol{h}_i$ and $x_2 = \boldsymbol{e}_2 \cdot \boldsymbol{h}_i$ and then represent $\boldsymbol{h}_i$ as a point (x_1, x_2) in the plane of $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$. Additionally, project the two eigenvectors into this plane! (Yes, you do have to project each eigenvector onto the other and onto itself!)

Task B

In the lecture, we derived the initial activity and steady-state activity as

$$\boldsymbol{v}_0 = (\boldsymbol{h} \cdot \boldsymbol{e}_1)\ \boldsymbol{e}_1 + (\boldsymbol{h} \cdot \boldsymbol{e}_2)\ \boldsymbol{e}_2, \qquad \boldsymbol{v}_{ss} = \frac{\boldsymbol{h} \cdot \boldsymbol{e}_1}{1-\lambda_1}\,\boldsymbol{e}_1 + \frac{\boldsymbol{h} \cdot \boldsymbol{e}_2}{1-\lambda_2}\,\boldsymbol{e}_2$$

Compute and plot initial activity $\boldsymbol{v}_0$ and steady-state activity $\boldsymbol{v}_{ss}$ for several input vectors $\boldsymbol{h}_i$ in the $\boldsymbol{e}_1$ - $\boldsymbol{e}_2$ -plane!

Based on the formula above, explain how the *direction* of steady-state solutions are determined by eigenvectors, eigenvalues, and input!

Further, explain how the *amplitude* of steady-state solutions are determined by eigenvectors, eigenvalues, and input!

Task C

Simulate the time-evolution of the system for several input vectors $\boldsymbol{h}_i$ (one at a time)!

Set the initial activity vector to the input, $\boldsymbol{v}_0 = \boldsymbol{h}$, and iteratively compute subsequent activity vectors $\boldsymbol{v}_i$. To this end, proceed from time 0 to time 20 in steps of $\Delta t = 0.05$. For each step i, compute the current equilibrium vector (given the current activity, input, and connectivity)

$$\boldsymbol{v}_i^{ss} = \boldsymbol{M} \cdot \boldsymbol{v}_i + \boldsymbol{h}$$

and let the system to relax exponentially towards this equilibrium:

$$\boldsymbol{v}_{i+1} = \boldsymbol{v}_i^{ss} + (\boldsymbol{v}_i - \boldsymbol{v}_i^{ss})\ \exp(-\Delta t)$$

Plot the initial states $\boldsymbol{v}_0 = \boldsymbol{h}$, the steady-states $\boldsymbol{v}_{ss}$ from Task B, and the simulated time-evolutions in the $\boldsymbol{e}_1$ - $\boldsymbol{e}_2$ -plane!

Task D (optional)

For one input vector, plot the time-evolution of all N populations with a suitable 3D plotting command, such as **surface**, **pcolor**, or **contour**.