**9.49 / 9.490 Neural Circuits for Cognition**
**Homework 5**
Due: **Friday** Dec 6, by midnight.

In this assignment, we will explore the nonlinear continuous attractors through pattern formation. Because these patterned states are attractors and thus persistent, they can be used for memory. When modified slightly to include a velocity-based pattern shift mechanism, the networks can perform analog integration on velocity inputs. We will explore these properties below.

1) **Network with rotation or shift-invariant connectivity.** In a linear network of $N$ neurons with dynamics specified by

$$\tau \frac{d\mathbf{x}}{dt} = -\mathbf{x} + \mathbf{W}\mathbf{x} + \mathbf{b}$$

consider a weight matrix that takes the form of a *circulant* matrix:

$$\mathbf{W} = \begin{bmatrix} c_0 & c_{N-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{N-1} & \cdots & c_2 \\ \vdots & c_1 & c_0 & \ddots & \\ c_{N-2} & \vdots & \ddots & \ddots & c_{N-1} \\ c_{N-1} & c_{N-2} & \cdots & c_1 & c_0 \end{bmatrix}$$

The entries $W_{mn}$ depend only on the difference $m - n$, and not on the specific values of $m$ or $n$. Thus, the dynamics are invariant to circular permutations of the neural indices (rotations of the set of neurons). Circulant matrices are a special case of Toeplitz matrices, and are very important in the context of signal processing. We will find they are also important for modeling nonlinear continuous attractor networks and integrators.

a. Show that the vector $\mathbf{1}$ of all 1's is an eigenvector of $W$. Recall that we showed the existence of this common mode eigenector for the winner-take-all network, a network with a special form of circulant matrix, with $c_0 = \alpha - \beta$, and all the remaining $c_i = \beta$.

b. Show that the vector whose $m$th component is $e^{ik_\alpha m}$ – where $i$ is the imaginary number $i \equiv \sqrt{-1}$ and $k_\alpha$ is some common spatial frequency (satisfying $k_\alpha N = 2\pi$) for all components – is an eigenvector of $\mathbf{W}$. (Hint: use the Circulant matrix identity $W_{mn} = W(m - n)$.) Recall that you want to show that $\sum_n W_{mn} v_n =$

$v_m\lambda$ for some $m$-independent scalar $\lambda$. You will find it helpful to define $l \equiv m - n$ and replace $n$ with $m - l$ in the eigenvector equation.) What is the associated eigenvalue? The matrix $W$ actually has $N - 1$ of these eigenvectors; the eigenvectors are spatially periodic; the $\alpha$th eigenvector has spatial frequency $k_\alpha = 2\pi(\alpha - 1)/N$. Suppose $c_1 = c_{N-1} \equiv c$ and all the remaining $c_i = 0$. Over what range of $c$ is the network stable? Draw the eigenvalue spectrum (eigenvalues ordered by size, plotted against index $\alpha$).

c. Excitatory coupling: For $c > 0$ (but in the stable range), which eigenvector will dominate the network response? Plot or draw this eigenvector as a function of neuron index.

d. Inhibitory coupling: For $c < 0$ (but in the stable range), which eigenvector will dominate the network response? Plot or draw this eigenvector as a function of neuron index.

e. Pattern formation with localized inhibition: For $c < 0$ but just above the critical value of instability, which eigenvector will dominate the network response? Plot or draw this eigenvector as a function of neuron index. Simulate the network with these weights, using some positive, uniform $\mathbf{b}$; instead of the linear neural transfer function above, use a rectified-linear transfer function. Compare the steady states of your simulated nonlinear network with your predicted dominant eigenvector from the linear network equation with the steady states of your nonlinear networks. This is a 1D model of the pattern formation required for grid-cell like activity (without the velocity-based shifting mechanism).

f. Multistability: The nonlinear network simulated in e. exhibits multi-stability: Like the bistable switch, more than one state is a stable state of the network dynamics. What is the set of stable patterns in e? What determines/how can one control which pattern is expressed by your network?

2) **A ring network integrator.** Here, we will construct a 1D ring network that integrates velocity inputs to update an internal estimate of an angular variable.

a. Single ring network: Simulate a network of $N = 256$ neurons with rectified-linear transfer function and a circulant weight matrix $W$

$$\tau \frac{d\mathbf{x}}{dt} = -\mathbf{x} + [\mathbf{W}\mathbf{x} + b]^+$$

with weights $W_{mn} = f(\theta_m - \theta_n)$, where $\theta_m = 2\pi m/N$ and $f(\theta) = (\cos(\theta) - 1)$. We have chosen all negative weights (inhibition dominated), and replaced local self-excitation with a local lack of inhibition together with a common feedforward excitatory drive $b = 5$ to all neurons. Use $dt = 0.1$ ms, $\tau = 10$, and choose small random initial activities for the entries of $\mathbf{x}$ (e.g. variance 0.1). Plot $W_n$, the weights emanating from neuron $n$ (choose any n, say n = N/2 = 128) to all the rest, and plot the resulting activity vectors $\mathbf{x}(t)$ as snapshots over time: you should see how the activity vectors evolve from low, random activity into stable bumps. What sets the amplitude of the bump?

b. Integrator network: In a new file, code two copies of the ring network from a.; call these copies left and right, with activities $\mathbf{x}_L$ and $\mathbf{x}_R$. We will slightly modify the network weights so that all outgoing projections from the left network, to both the left and right networks, are now shifted slightly to the left; the same for the right network, except that the shifts are to the right. Thus, $W^L_{mn} = f(\theta_m - \theta_n - \Delta)$ and $W^R_{mn} = f(\theta_m - \theta_n + \Delta)$, and

$$\tau \frac{d\mathbf{x}_L}{dt} = -\mathbf{x}_L + [\mathbf{W}^L \mathbf{x}_L + \mathbf{W}^R \mathbf{x}_R + b_L]^+$$

$$\tau \frac{d\mathbf{x}_R}{dt} = -\mathbf{x}_R + [\mathbf{W}^L \mathbf{x}_L + \mathbf{W}^R \mathbf{x}_R + b_R]^+$$

Each ring has a different common excitatory drive, $b_L, b_R$, respectively, instead of $b$. The common drive is modulated by angular velocity: $b_L = b_0(1 - gv), b_R = b_0(1 + gv)$, where $v$ is a velocity in radians per second, and $g$ is a gain factor that modulates how strongly the velocity drives the integrator. With $g = 1$, $v = 1$rad/s, $b_0 = 5$, and $\Delta = 2\pi(2/N)$, i.e., the minimum-inhibition point of the outgoing weights from ring $R, L$ is shifted 2 neurons to the right or left, respectively, plot the activity vectors over time (as your code is running), and watch how the activity bumps move. In your submission, show how they move. Using a max command, find and record the location of the peak and plot this location as a function of time for positive and negative velocities. You have built a nonlinear neural integrator: also try giving a time-varying angular velocity, and show that the network bumps track the integral of this velocity.