Winter 2019 PHYS 178/278, Homework 4
Due 11:59 PM on March 17^{th}

Please submit your assignment as one "LastName, FirstName_PID_HW4.pdf" file via email (whsu@physics.ucsd.edu).
Welcome to use any software or programming language you are familiar for doing the simulation.

1 Ring Attractor Network

We are going to construct a simplified ring network that simulates the activity of neural population, $r_i(t)$, tuned to the external angle stimulus $\theta_0(t)$. The assumptions in our model are:
(1) N neurons in the network have evenly distributed preferred angle θ_i,

$$\theta_i = \frac{2\pi}{N} i, \text{ for } i = 1, 2, ..., N.$$

(2) the synaptic connections between neurons i, j is a function of the angular distance between the two,

$$\begin{aligned}\mathbf{W}_{i,j} &= W(\theta_i - \theta_j), \\ &= \frac{J_1}{N}\left[\cos(\theta_i - \theta_j) - 1\right], \text{ with } J_1 \geq 0.\end{aligned}$$

(3) the input to the i^{th} angle-specific cells is defined as,

$$\begin{aligned}\boldsymbol{I}_{\text{ext},i}(t) &= I_{\text{ext}}(\theta_i - \theta_0(t)), \\ &= I_1(t)\cos(\theta_i - \theta_0(t)),\end{aligned}$$

where the input reaches the maximum when the external angle, $\theta_0(t)$, matches the preferred one, i.e., $\theta_0(t) = \theta_i$; $I_1(t)$ is the input intensity; both $\theta_0(t)$ and $I_1(t)$ can be time-dependent or -independent. The dynamics of neural activities follow the differential equation,

$$\tau \frac{dr_i(t)}{dt} + r_i(t) = f\left[\sum_{j=1}^{N} W(\theta_i - \theta_j) r_i(t) + I_{\text{ext}}(\theta_i - \theta_0(t))\right], \text{ for } i = 1, 2, ..., N$$

with the nonlinear gain function $f(\cdots) = \frac{1+\tanh(\cdots)}{2}$, and a homogeneous decay time constant $\tau (= 10)$. The same equation in the matrix notation is,

$$\tau \frac{d\mathbf{r}(t)}{dt} + \mathbf{r}(t) = f\left[\mathbf{W}\mathbf{r}(t) + \boldsymbol{I}_{\text{ext}}\left(\vec{\theta} - \theta_0(t)\right)\right].$$

1.1 Plot $W(\theta_i - \theta_j) = W(\Delta\theta)$ as a function of $\Delta\theta$, $\Delta\theta \in [-\pi, \pi] = [-180°, 180°]$. What does it tell you about the connections to the local (having similar preferred angle) and to the distal neurons? Are the connections excitatory or inhibitory?

1.2 Set $I_1 = 1$, and $J_1 = 5$. Plot the steady-state activity profile at a constant input $\theta_0 = \frac{2\pi}{3}, \pi, \frac{7\pi}{4}$, respectively.
Hint: simulate the dynamics of neural population $N \geq 500$, and plot $r_i(t \to \infty)$ VS θ_i.

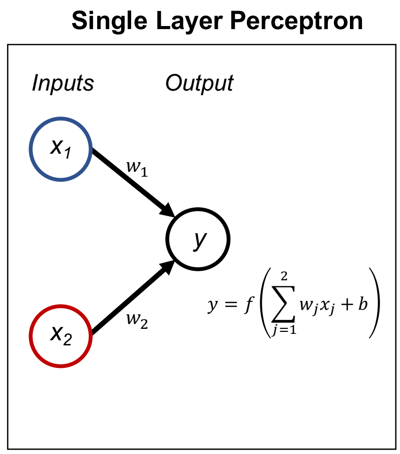
1.3 How does the **width** of steady-state activity profile, $r_i(t \to \infty)$ VS θ_i, **change with the connectivity strength** J_1**?**

1.4 From **1.2** we have verified the ring network can be in any state that matches the angle of constant input. How would the network trace a time-dependent external angle stimulus? Let's design the time-dependent angle stimulus,

$$\theta_0(t) = \begin{cases} \Theta_a, & 0 \leq t < t_a \\ \Theta_b, & t_a \leq t < t_b \\ \Theta_c, & t_b \leq t \end{cases}, \text{ with } |\Theta_a - \Theta_b| \cong |\Theta_b - \Theta_c| \in \left[\frac{\pi}{2}, \frac{2\pi}{3}\right]$$

then feed it to your ring network model. Please try to simulate with different connectivity strengths $J_1\,(\geq 5)$, and show the activity profile $\mathbf{r}(t)$ as a function of time. **How does the population activity track the external input angle?**

2 Single Layer Perceptron



Here you will demonstrate how a single-layer perceptron *network* performs a NAND (NOT-AND) function.

x_1	0	0	1	1
x_2	0	1	0	1
$d(x_1, x_2) = d(x)$	1	1	1	0

Table 1: NAND truth table.

A simple perceptron network consists of two inputs $x_{1,2}$ and an desired output $d(x_1, x_2)$.The output y is defined as:

$$y = f(\mathbf{w}\vec{x} + b), \tag{1}$$

$$f(z) = \begin{cases} 1 & z > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

where $\mathbf{w} = (w_1, w_2)$ are the connection weights between the inputs and the output layer, $\vec{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ are the inputs, b is the bias and $(\mathbf{w}\vec{x} + b) = (w_1x_1 + w_2x_2 + b)$. To learn the function, the weights and the bias have to be updated according to:

$$\delta(t) = d(\vec{x}(t)) - y(t) \tag{3}$$

$$w_i(t+1) = w_i(t) + \eta\,\delta(t)\,x_i(t), \tag{4}$$

$$b(t+1) = b(t) + \eta\,\delta(t), \tag{5}$$

where η is the learning rate $(0 < \eta \leq 1)$ and $y(t) = f(\mathbf{w}(t)\vec{x} + b(t))$ is the calculated output. The value of w_1, w_2 and b that perform the NAND function will be learned when you repeatedly introduce the four truth statements as inputs.

2.1 Run the perceptron simulation by constructing a "test set". What are the resulting w_1, w_2 and b?

2.2 Does your result correctly separate the 2-d input space? *Hint*: plot the line $w_1x_1 + w_2x_2 + b = 0$ on x_1, x_2-plane.

2.3 What η did you use? Vary η and plot the number of iterations as a function of η.