Assignment 1: Neuroscience of Decision Making PSY 307 (Monsoon 2023)

Name: Tushar Chandra
Roll Number: 2021211

Instructions: Please write your own responses and do not copy or lift text/code from any source (including the paper). If you are referring to credible external sources other than the attached paper for your answers, please cite those sources (within the body of text and the provide a reference list at the end) in the APA citation format (https://www.mendeley.com/guides/apa-citation-guide). Word limits given are indicative and less than the indicated numbers may also be used.

Please download this MS word question-cum-response template to TYPE your answers and feel free to add sheets as required. Convert this document to a PDF and rename the file: name_RollNo. before submitting. Please note that answers in this template only will be evaluated and hand-written or scanned answer sheets will not be evaluated.

[Strict deadline for submission: 18 September, Monday, 9:00 AM]

Q1) Fill out the google form: https://forms.gle/k93XPaUspgxg4NN6A
Done

Q2) An experimenter recorded a SINGLE cortical neuron's activity from the part of the brain processing visual information as the organism viewed visual stimuli on the screen. The collected dataset is attached herewith: 'Data1_NDM.mat'. It is a Three-dimensional MATLAB array. Note the following.

- **Dimension 1 = Stimulus orientations (45 degree to 202.5 degree with increments in step size of 22.5 degree visual angle).**
- **Dimension 2 = Time (sampling frequency of the neuronal activity = 1000 Hz); Total neuronal recording duration = 3.5 seconds; Time = 0 (stimulus not moving);Time = 500 (stimulus moving) ; Time = 2500 (stimulus off);**
- **Dimension 3 = Trials or stimulus repetitions**
- **Value 1 = action potential (spike) fired by the neuron; Value 0 = no action potential fired by the neuron**

Now solve the following. Insert a figure (wherever required) and paste the MATLAB/Python/R code for the same. Any figure must provide all information necessary to interpret it including axes labels, captions/legends (see Fig.1 of the attached paper for a sample; simple figure titles as captions are not enough).

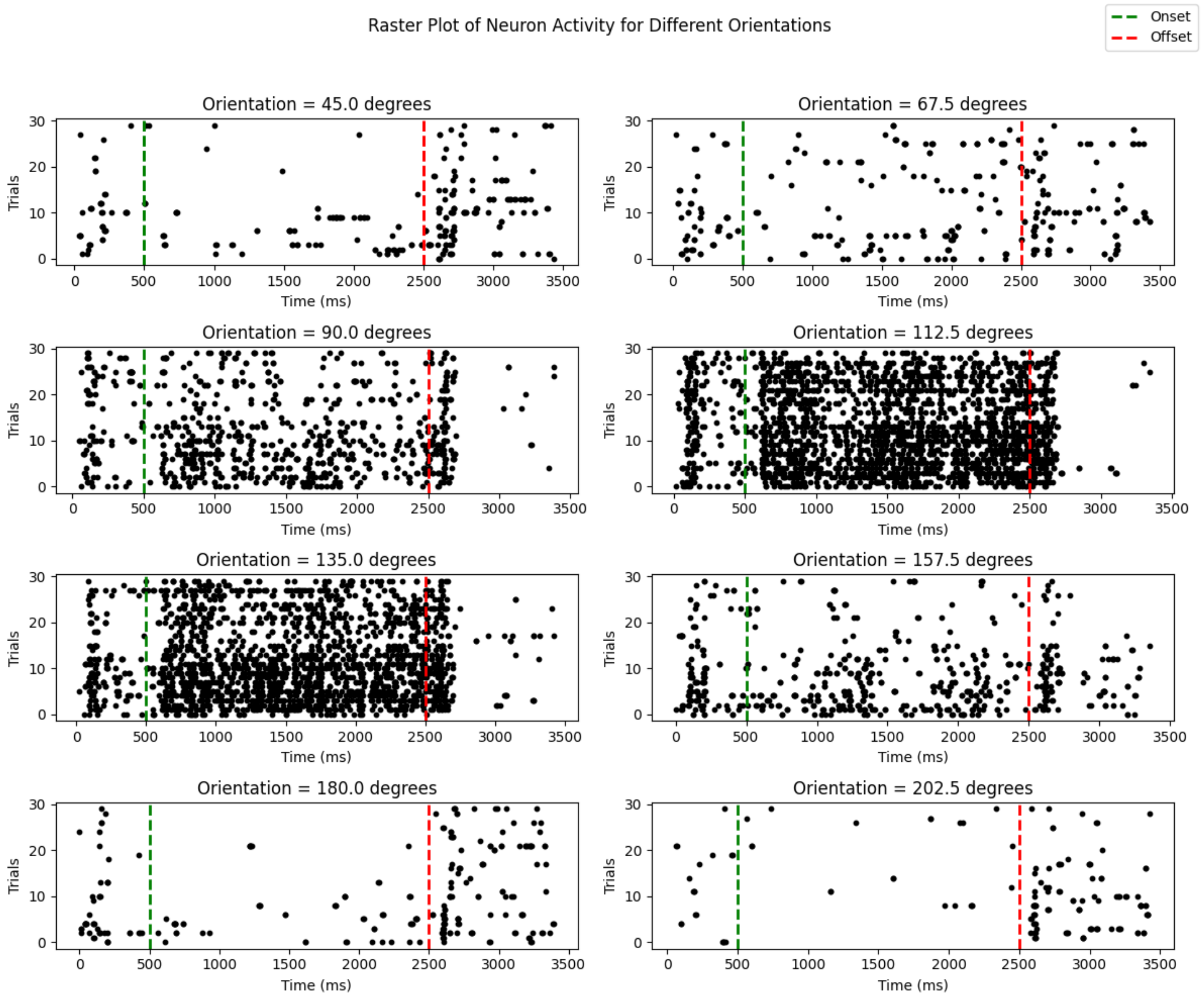
[links about the 3-dimensional array in MATLAB and importing MATLAB data arrays into Python and R

- **https://in.mathworks.com/help/matlab/math/multidimensional-arrays.html**
- **https://in.mathworks.com/help/matlab/matlab_external/matlab-arrays-as-python-variables.html**
- **https://stackoverflow.com/questions/11671883/importing-an-array-from-matlab-into-r]**

A) Create a Raster plot of the neuron for ALL EIGHT orientations of the stimulus and mark the onset of stimulus movement and offset of the stimulus by a vertical green and red line respectively on the same individual subplots. Mark the spikes (action potentials) with solid black circles. [5 marks]

Hint: Create a larger figure with eight subplots (positioned as 4 rows x 2 columns); Indicate the stimulus orientation on top of each subplot as subplot title

Ans.



Code:

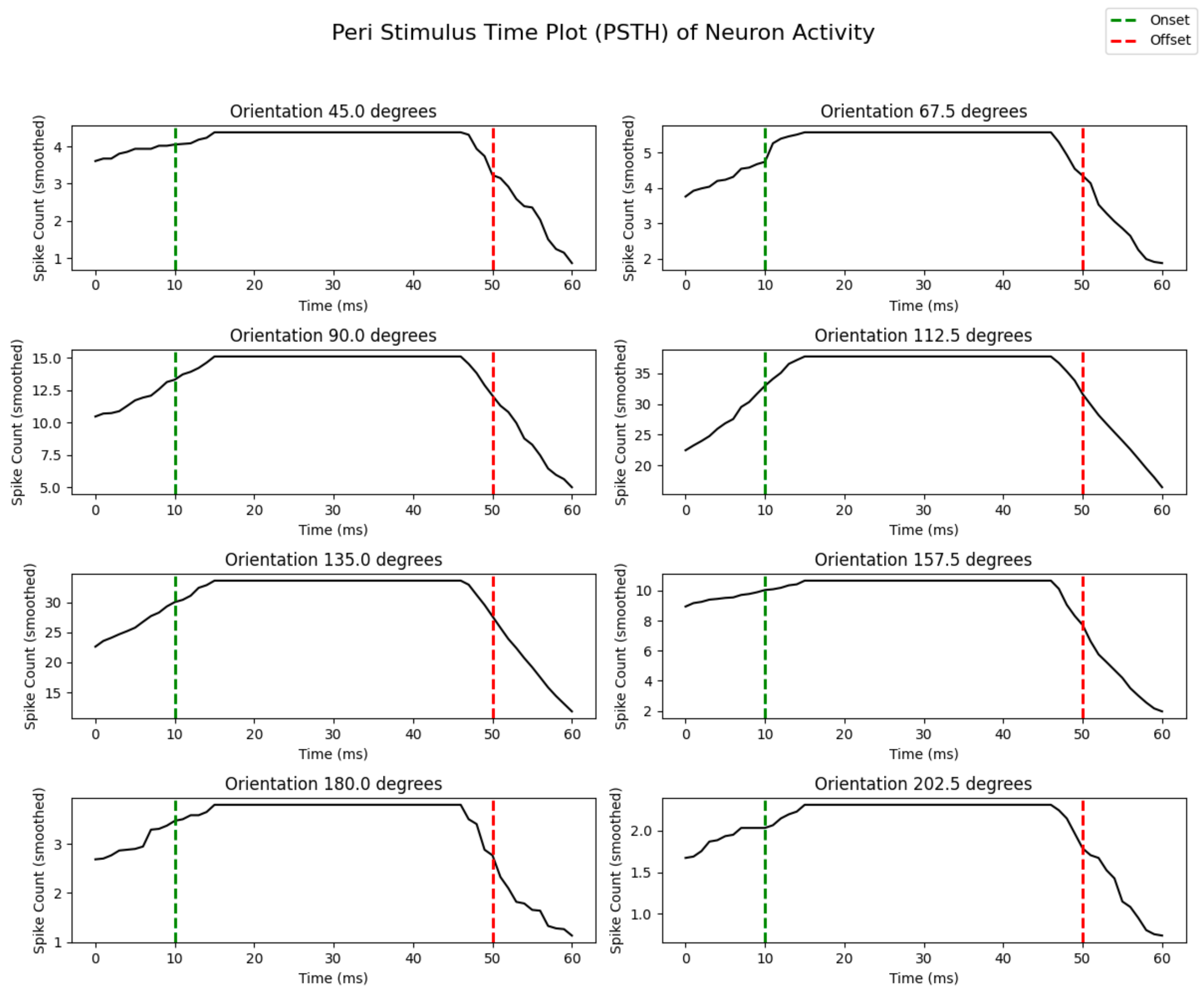
```
# Importing necessary libraries
import scipy.io
import numpy as np
```

```python
import matplotlib.pyplot as plt

# Reading the Data
data = scipy.io.loadmat('Data1_NDM.mat')['Data1_NDM']
onsetTime = 500  # Stimulus start time
offsetTime = 2500  # Stimulus end time

labels = []
flag = 1

# Subplots (4 rows x 2 columns)
fig, axs = plt.subplots(4, 2, figsize=(12, 10))

for orientation in range(8):
    ax = axs[orientation // 2, orientation % 2]

    # Slicing Data for visualization
    currData = data[orientation, :, :]

    # Spikes
    spikes, trialIndexes = np.where(currData)

    # Plotting spikes
    ax.scatter(spikes, trialIndexes, marker='o', color='black', s=10)

    # Marking
    ax.axvline(x=onsetTime, color='green', linestyle='--', linewidth=2)
    ax.axvline(x=offsetTime, color='red', linestyle='--', linewidth=2)
    ax.set_title(f'Orientation = {45 + orientation * 22.5} degrees')

    # Labels
    ax.set_xlabel('Time (ms)')
    ax.set_ylabel('Trials')

    if flag:
      labels.append(ax.axvline(x=onsetTime, color='green', linestyle='--', linewidth=2,
label='Onset'))
      labels.append(ax.axvline(x=offsetTime, color='red', linestyle='--', linewidth=2,
label='Offset'))
      flag = 0

# Title
fig.suptitle('Raster Plot of Neuron Activity for Different Orientations')

fig.legend(handles=labels, bbox_to_anchor=(1, 1))

# Adjusting Spacing
plt.tight_layout(rect=[0, 0, 1, 0.95])
plt.show()
```

B) Create a Peri Stimulus Time Plot of the neuron for ALL EIGHT orientations of the stimulus and mark the onset of stimulus movement and offset of the stimulus by a vertical green and red line respectively on the same individual subplots. Before computing the histogram, smooth the data for each subplot over a time window of 61 ms. [5 marks]

Hint: Create a larger figure with eight subplots (positioned as 4 rows x 2 columns); Smooth the data by moving average method; A line plot would suffice and depict the trend; Indicate the stimulus orientation on top of each subplot as subplot title

Ans.

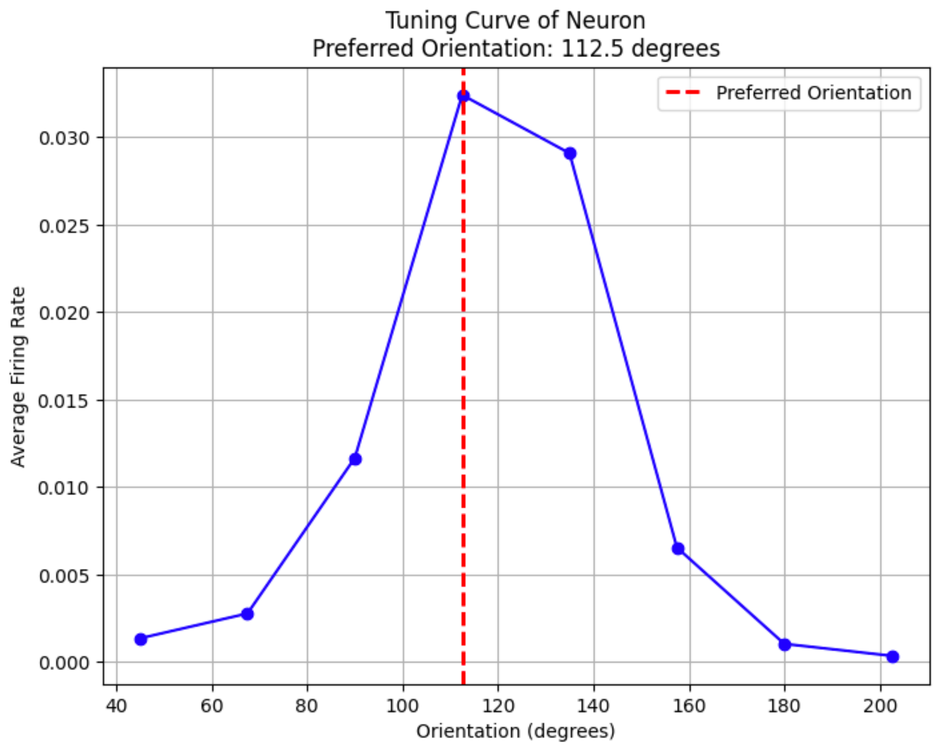


```
onsetTime = 10  # Stimulus start time
offsetTime = 50  # Stimulus end time

labels = []

# Defining the time for smoothing
size = 61  # in ms
```

```
sampling = int(size / 1000 * 1000)  # 1000Hz Sampling Rate

fig, axs = plt.subplots(4, 2, figsize=(12, 10))

for orientation in range(8):
    ax = axs[orientation // 2, orientation % 2]

    # Slicing the data for visualization
    currData = data[orientation, :, :]

    # Smoothing the data
    fineData = np.convolve(currData.sum(axis=0), np.ones(sampling)/sampling, mode='same')

    time = np.arange(len(fineData))

    # Plot the PSTH as a line plot
    ax.plot(time, fineData, color='black')

    ax.axvline(x=onsetTime, color='green', linestyle='--', linewidth=2)
    ax.axvline(x=offsetTime, color='red', linestyle='--', linewidth=2)

    # Set axis labels and title
    ax.set_xlabel('Time (ms)')
    ax.set_ylabel('Spike Count (smoothed)')
    ax.set_title(f'Orientation {45 + orientation * 22.5} degrees')

    # labels = [
    #     plt.Line2D([0], [0], color='green', linestyle='--', linewidth=2, label='Onset',
loc='upper right'),
    #     plt.Line2D([0], [0], color='red', linestyle='--', linewidth=2, label='Offset', loc='upper
right')
    # ]

    # ax.legend(handles=labels)

    # x-axis limits
    # ax.set_xlim(0, len(fineData))

# Final Labels
# plt.axvline(x=onsetTime, color='green', linestyle='--', linewidth=2, label='Onset')
# plt.axvline(x=offsetTime, color='red', linestyle='--', linewidth=2, label='Offset')

# labels for both onset and offset

# dummy1 = plt.Line2D([0], [0], color='green', linestyle='--', linewidth=2, label='Onset')
# dummy2 = plt.Line2D([0], [0], color='red', linestyle='--', linewidth=2, label='Offset')

fig.suptitle('Peri Stimulus Time Plot (PSTH) of Neuron Activity', fontsize=16)

labels = [
    plt.Line2D([0], [0], color='green', linestyle='--', linewidth=2, label='Onset'),
    plt.Line2D([0], [0], color='red', linestyle='--', linewidth=2, label='Offset')
]

fig.legend(handles=labels, loc='upper right')
plt.tight_layout(rect=[0, 0, 1, 0.95])
# plt.legend(handles=[dummy1,dummy2])
plt.show()
```

C) Create a figure representing the Tuning Curve of the neuron from the average firing rate of the neuron (between 600 – 2500 ms and all trials) for each orientation. Computationally calculate the 'preferred orientation' of the neuron. Report the same on the title of the plot and mark it on the plot. [5+5 marks]

Ans.



```
onsetTime = 600  # Stimulus start time
offsetTime = 2500   # Stimulus end time

#(1000Hz Sampling Rate)
startIndex = int(onsetTime * 1000 / 1000)
endIndex = int(offsetTime * 1000 / 1000)

# Extracting Data
currData = data[:, startIndex:endIndex, :]
```

```python
avgRate = np.mean(currData, axis=(1, 2))

# Maximum firing rate
orientation = (np.argmax(avgRate) * 22.5) + 45

# Creating tuning curve plot
plt.figure(figsize=(8, 6))
orientations = np.arange(45, 225, 22.5)
plt.plot(orientations, avgRate, marker='o', linestyle='-', color='blue')
plt.title(f'Tuning Curve of Neuron\nPreferred Orientation: {orientation} degrees')
plt.xlabel('Orientation (degrees)')
plt.ylabel('Average Firing Rate')
plt.grid(True)

# Adding Markers
plt.axvline(x=orientation, color='red', linestyle='--', linewidth=2, label='Preferred Orientation')
plt.legend()
plt.show()
# print(f'Preferred Orientation: {orientation} degrees')
```