Assignment 1: Neuroscience of Decision Making PSY 307 (Monsoon 2024)

Name: Kunal Sharma

Roll Number: 2021331

2.

A) Create a Raster plot of the two neurons recorded from two distinct brain regions for ALL SIX of the stimuli (2 X face; 2 X text; 2 X speech) and mark the onset of stimulus onset and offset of the stimulus by a vertical green and red line respectively on the same individual subplots. Mark the spikes (action potentials) with blue colour. [5 marks]

Hint: Create a larger figure with six subplots (positioned as 3 rows x 2 columns). Plot dataset_H in the first column and dataset_A in the second column of the larger figure. Indicate the stimulus type - face, text, speech on top of each subplot as subplot title

Answer:

```
import matplotlib.pyplot as plt

import scipy.io


#Loading data from the .mat file

dataset_A = scipy.io.loadmat('dataset_A.mat')

dataset_H = scipy.io.loadmat('dataset_H.mat')


# Accessing data where: Face data is stored in index 0, Text data is stored in index 1, Speech data is stored in index 2

face_H, text_H, speech_H = dataset_H['dataset_H'][0]

data_H = [face_H, text_H, speech_H]

face_A, text_A, speech_A = dataset_A['dataset_A'][0]

data_A = [face_A, text_A, speech_A]


# To create a figure with 6 subplots arranged in 3 rows and 2 columns

fig, axs = plt.subplots(nrows=3, ncols=2, figsize=(12, 8), sharex=True, sharey=True)

green_l=[]

red_l=[]
```

```python
# Used list comprehension to define stimuli and regions
stimuli, regions = [stim for stim in ['Face', 'Text', 'Speech']], [region for region in ['H', 'A']]
i = 0
# Iterate over the stimuli types (Face, Text, Speech) using a while loop
while i < 3:
    trial = 0
    # Left column: dataset_H (Region H) for 30 trials
    while trial < 30:
        arr = []
        x = 0
        # Collecting data points from data_H[i][trial][0] using a while loop
        while x < len(data_H[i][trial][0]):
            temp = data_H[i][trial][0][x][0]
            arr.append(temp)
            x += 1
        axs[i, 0].plot(arr, [trial] * len(arr), 'bo', markersize=2)
        trial += 1
    # Define stimulus onset and offset properties
    line_info = [
        (0, 'green', 'Stimulus - Onset'),
        (1000, 'red', 'Stimulus - Offset')
    ]
    # Looping on the line information to add vertical lines
    for position, color, label in line_info:
        if color=='green':
            green_l=axs[i, 0].axvline(position, color=color, linestyle='--', label=label)
        else:
            red_l=axs[i, 0].axvline(position, color=color, linestyle='--', label=label)
```

```
    # Set title and ylabel
    axs[i, 0].set_title(f'{stimuli[i]} - Region H')
    axs[i, 0].set_ylabel('Trials')


    trial = 0
    # Right column: dataset_A (Region A) for 30 trials
    while trial < 30:
        arr = []
        x = 0
        # Collecting data points from data_A[i][trial][0] using a while loop
        while x < len(data_A[i][trial][0]):
            temp = data_A[i][trial][0][x][0]
            arr.append(temp)
            x += 1
        axs[i, 1].plot(arr, [trial] * len(arr), 'bo', markersize=2)
        trial += 1
    # Define positions and colors for vertical lines
    positions = [0, 1000]
    colors = ['green', 'red']
    # Looping on positions and colors to add vertical lines
    for pos, color in zip(positions, colors):
        axs[i, 1].axvline(pos, color=color, linestyle='--')
    # Set title for the subplot
    axs[i, 1].set_title(f'{stimuli[i]} - Region A')
    i += 1
# Used to Set common labels by using a single invisible axis
fig.add_subplot(111, frameon=False)  # Adding a subplot to act as the frame
plt.tick_params(labelcolor='none', top=False, bottom=False, left=False, right=False)   # Used to hide tick labels
```

```
lables = [green_l, red_l]

fig.legend(handles=lables, loc='upper right')

plt.xlabel('Time (ms)', fontsize=12)

plt.ylabel('Trial Number', fontsize=12, labelpad=20)

plt.show()
```

Output Graph:

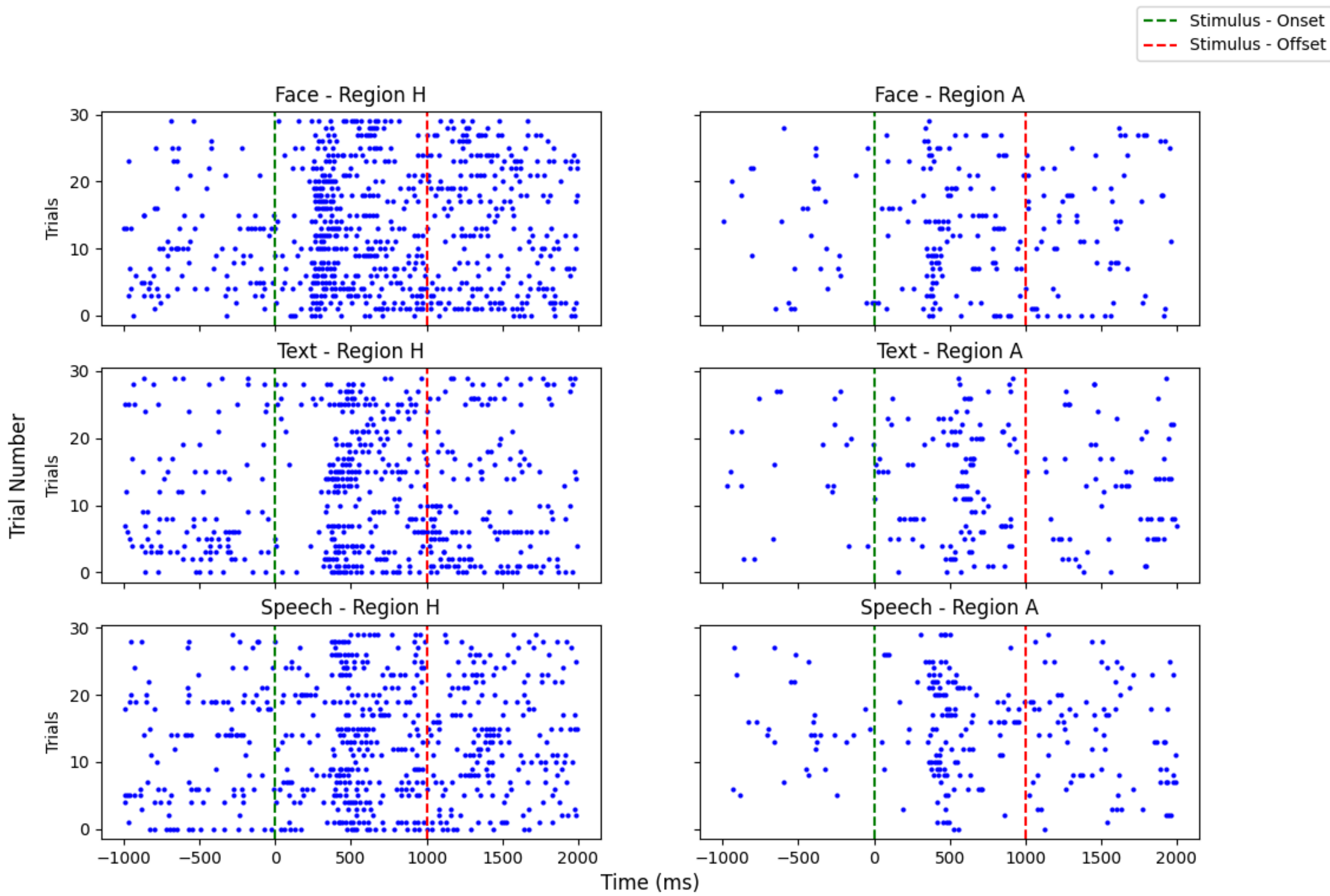

Figure 1.1 Raster plots showing the brain responses to three distinct stimulus types—face, text, and speech—in Regions H and A. Neural activity over 30 trials is represented trial by trial in each subplot. The green dashed line indicates the start of the stimulus, while the red dashed line indicates the offset of the stimulus. The x-axis displays time in milliseconds. The trial number is displayed on the y-axis. Our entire visualization explains to us the variety of neural responses elicited by several stimulus types in different brain areas.

B) Create a Peri Stimulus Time Plot of two neurons recorded from two distinct brain regions for ALL SIX of the stimuli and mark the onset of stimulus 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 50 ms. [5 marks]

Hint: Create a larger figure with six subplots (positioned as 3 rows x 2 columns); Smooth the data by moving average method; a line plot would suffice and depict the trend. Plot dataset_H in the first column and dataset_A in the second column of the larger figure. Indicate the stimulus type - person, text, speech on top of each subplot as subplot title

Answer:

```
import scipy.io

import matplotlib.pyplot as plt

import numpy as np


#Loading data from the .mat file

dataset_A = scipy.io.loadmat('dataset_A.mat')

dataset_H = scipy.io.loadmat('dataset_H.mat')


# Accessing data where: Face data is stored in index 0, Text data is stored in index 1, Speech data
is stored in index 2

face_H, text_H, speech_H = dataset_H['dataset_H'][0]

data_H = [face_H, text_H, speech_H]

face_A, text_A, speech_A = dataset_A['dataset_A'][0]

data_A = [face_A, text_A, speech_A]


green_l=[]

red_l=[]


# Define function for PSTH computation

def psth(data, duration=3000):
```

```
    num_bins = duration // 10
    psth = np.zeros(num_bins)
    for trial in data:

        # Extract timestamps and convert to appropriate format
        timestamps = [spike[0] for spike in trial[0]]

        # Bin the spike timestamps
        binned_spikes, _ = np.histogram(timestamps, bins=np.arange(-1000, duration - 1000 + 10, 10))
        psth += binned_spikes

    # Average over the number of trials
    psth /= len(data)
    return psth

# Function for moving average smoothing
def smooth_psth(psth, window_size=50):
    window = window_size // 10  # Convert window size to number of bins

    return np.convolve(psth, np.ones(window)/window, mode='same')

window_size = 50  # ms
duration = 3000  # ms (total duration of recording)
# making figure with 6 subplots: 3 rows x 2 columns
fig, axs = plt.subplots(3, 2, figsize=(12, 8), sharex=True, sharey=True)
stimuli = ['Face', 'Text', 'Speech']

# Looping over the stimuli types (Face, Text, Speech)
for i in range(3):
```

```
    # Region H (left column)
    psth_H = psth(data_H[i] ,duration=duration)
    time_bins = np.arange(-1000, duration - 1000, 10)

    axs[i, 0].plot(time_bins, smooth_psth(psth_H, window_size=window_size), color='blue')

      green_l = axs[i, 0].axvline(0, color='green', linestyle='--', label='Stimulus Onset')    #
Stimulus onset
      red_l = axs[i, 0].axvline(1000, color='red', linestyle='--', label='Stimulus Offset')    #
Stimulus offset

    axs[i, 0].set_title(f'{stimuli[i]} - Region H')
    axs[i, 0].set_ylabel('Firing Rate (Hz or spikes/bin)')
    axs[i, 0].grid(True)

    # Region A (right column)
    psth_A = psth(data_A[i], duration=duration)
    axs[i, 1].plot(time_bins, smooth_psth(psth_A, window_size=window_size), color='blue')
    axs[i, 1].axvline(0, color='green', linestyle='--', label='Stimulus Onset')  # Stimulus onset
    axs[i, 1].axvline(1000, color='red', linestyle='--', label='Stimulus Offset')  # Stimulus offset
    axs[i, 1].set_title(f'{stimuli[i]} - Region A')
    axs[i, 1].grid(True)

# Set common labels and layout
lables = [green_l, red_l]
fig.text(0.5, 0.04, 'Time (ms)', ha='center', fontsize=12)
fig.text(0.04, 0.5, 'Smoothed Firing Rate', va='center', rotation='vertical', fontsize=12)
fig.legend(handles=lables, loc='upper right')
plt.show()
```

Output Graph:

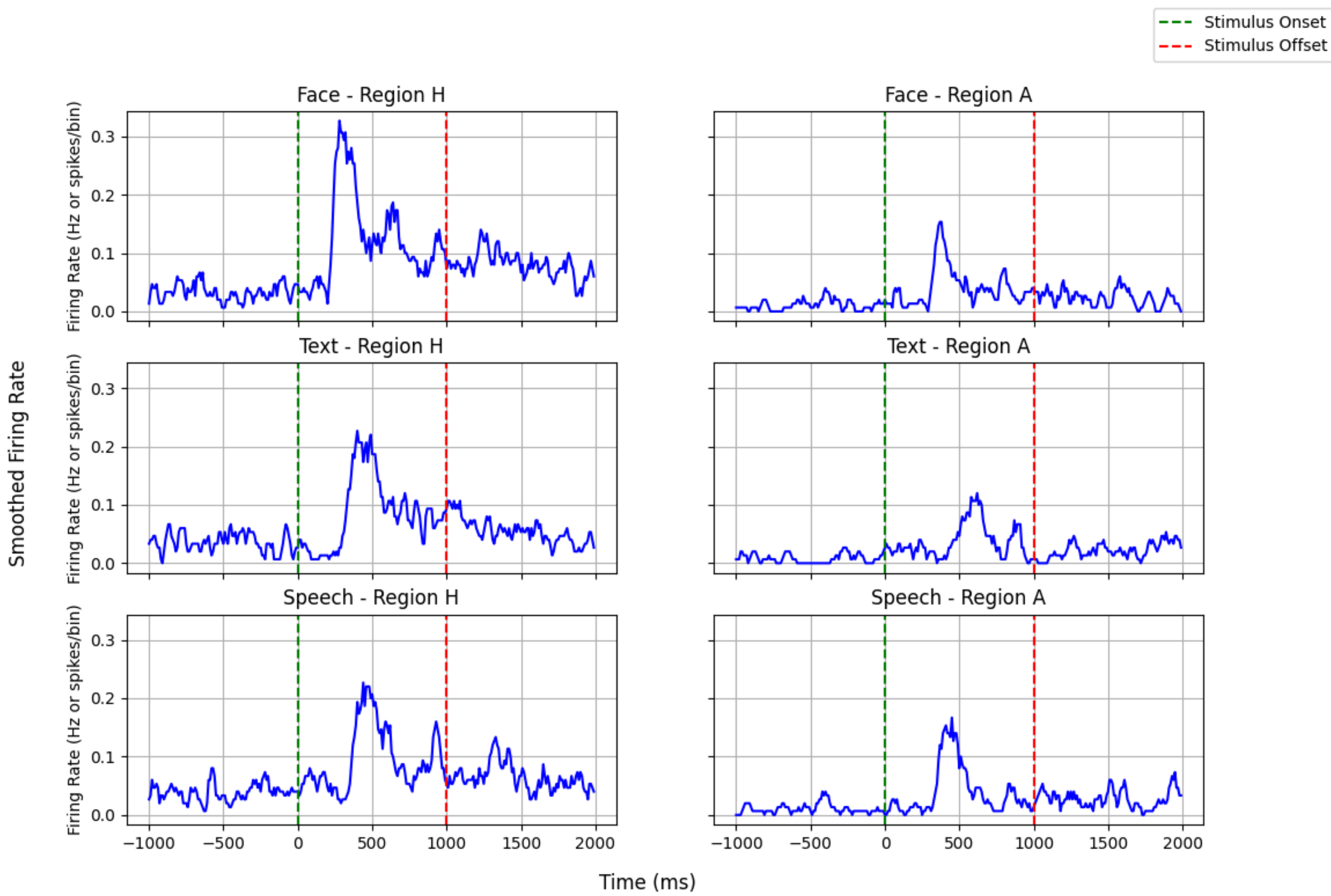

Figure 1.2 Neural activity from two different brain regions (Region H and Region A) in response to three different stimuli (Face, Text, and Speech) is shown in peri-stimulus time histograms (PSTH). For every type of stimulation, the firing rate (spikes/bin) is displayed against time in milliseconds. Every subplot's data has been smoothed over a 50 ms time interval. The stimulus onset is indicated by the green vertical line, and the offset is indicated by the red vertical line. Our entire visualisation provides us insight into the time-dependent nature of neural responses by both the brain areas to various stimuli.

C) Create figures representing the preferred stimulus type of two neurons (dataset_H & dataset_A) recorded from two distinct brain regions from the average firing rate of the neuron (between 0 – 1000 ms and all trials) for each stimulus type using a binwidth of 200 ms. Plot the curves of each stimulus type recorded from the same neuron in one graph. There will be two separate graphs for the two neurons/datasets. Mark the curves for stimulus type -person in red colour, text in green colour, and speech in blue colour. Computationally calculate the

‘most preferred stimulus type’ for each neuron and report it on the title of the plot. Also, report the order of stimulus preference for each neuron. [5+2+1 marks]

What kind of neuronal information coding is represented by these neurons and why?[2 marks]
Hint: Create a larger figure with two subplots (positioned as 1 row x 2 columns).

Answer:

```
import scipy.io
import matplotlib.pyplot as plt
import numpy as np

# #Loading data from the .mat file
dataset_H = scipy.io.loadmat('dataset_H.mat')
dataset_A = scipy.io.loadmat('dataset_A.mat')

# Accessing data where: Face data is stored in index 0, Text data is stored in index 1, Speech data
is stored in index 2
face_H, text_H, speech_H = dataset_H['dataset_H'][0]
data_H = [face_H, text_H, speech_H]

face_A, text_A, speech_A = dataset_A['dataset_A'][0]
data_A = [face_A, text_A, speech_A]

# Extracting the columns from each dataset
H_face = []
H_text = []
H_speech = []

i = 0
while i < 3:

   # List to store the extracted data for the current dataset
   arr1 = []
   trial = 0

   while trial < 30:

       # List to store the values from the current trial
       arr = []
       x = 0

       while x < len(data_H[i][trial][0]):
           temp = data_H[i][trial][0][x][0]
```

```
            arr.append(temp)
            x += 1

        arr1.append(arr)
        trial += 1

    # Assigning the extracted data to the appropriate dataset based on the index
    if i == 0:
        H_face = arr1

    if i == 1:
        H_text = arr1
    else:
        H_speech = arr1

    i += 1


A_face = []
A_text = []
A_speech = []

i = 0
while i < 3:

    # List to store the extracted data for the current dataset
    arr1 = []
    trial = 0
    while trial < 30:

        # List to store the values from the current trial
        arr = []
        x = 0

        while x < len(data_A[i][trial][0]):
            temp = data_A[i][trial][0][x][0]
            arr.append(temp)
            x += 1

        arr1.append(arr)
        trial += 1

    # Assigning the extracted data to the appropriate dataset based on the index
    if i==0:
```

```
        A_face=arr1

    if i==1:
        A_text=arr1
    else:
        A_speech=arr1

    i += 1

# Function to calculate average firing rate
def average_firing_rate(trials):

    firing_rates = np.zeros(1000 // 200)

    for trial in trials:

        trial = np.array(trial)  # Convert list to numpy array

         trial_spikes = trial[(trial >= 0) & (trial <= 1000)]  # Only consider spikes between 0-1000
ms
        hist, _ = np.histogram(trial_spikes, bins=np.arange(0, 1000 + 200, 200))
        firing_rates += hist

     firing_rates = firing_rates / len(trials) / (200 / 1000)   # Average over trials and convert to
Hz
    return firing_rates

# Calculate average firing rates for each condition and region
fr_H_face = average_firing_rate(H_face)
fr_H_text = average_firing_rate(H_text)
fr_H_speech = average_firing_rate(H_speech)

fr_A_face = average_firing_rate(A_face)
fr_A_text = average_firing_rate(A_text)
fr_A_speech = average_firing_rate(A_speech)

# Function to determine preferred stimulus
def preferred_stimulus(fr_face, fr_text, fr_speech):

    # Created a dictionary to store the average firing rates
    avg_fr = dict(zip(['Face', 'Text', 'Speech'], map(np.mean, [fr_face, fr_text, fr_speech])))

    preferred_stim = max(avg_fr.items(), key=lambda x: x[1])[0]
```

```python
    # Sorting stimuli by average firing rates in descending order
    sorted_stimuli = [stim for stim, _ in sorted(avg_fr.items(), key=lambda x: x[1], reverse=True)]

    return preferred_stim, sorted_stimuli, avg_fr


# Determine preferred stimulus for each neuron
pref_stim_H, order_H, avg_fr_H = preferred_stimulus(fr_H_face, fr_H_text, fr_H_speech)
pref_stim_A, order_A, avg_fr_A = preferred_stimulus(fr_A_face, fr_A_text, fr_A_speech)

# Plotting
fig, axs = plt.subplots(1, 2, figsize=(15, 5), sharex=True, sharey=True)
time_bins = np.arange(0, 1000, 200)

# Plot for Neuron H
axs[0].plot(time_bins, fr_H_face, 'r', label='Face')
axs[0].plot(time_bins, fr_H_text, 'g', label='Text')
axs[0].plot(time_bins, fr_H_speech, 'b', label='Speech')
axs[0].set(title=f'Region H - Preferred: {pref_stim_H}\nOrder: {order_H}', xlabel='Time (ms)',
ylabel='Average Firing Rate (Hz)')
axs[0].legend(loc='best')
axs[0].grid(True)

# Plot for Neuron A
axs[1].plot(time_bins, fr_A_face, 'r', label='Face')
axs[1].plot(time_bins, fr_A_text, 'g', label='Text')
axs[1].plot(time_bins, fr_A_speech, 'b', label='Speech')
axs[1].set(title=f'Region A - Preferred: {pref_stim_A}\nOrder: {order_A}',
          xlabel='Time (ms)', ylabel='Average Firing Rate (Hz)')
axs[1].legend(loc='best')
axs[1].grid(True)

plt.tight_layout()
plt.show()

# Printing average firing rates for Region H
print("Region H - Average Firing Rates:")
print(f"Face: {avg_fr_H['Face']:.2f} Hz | Text: {avg_fr_H['Text']:.2f} Hz | Speech:
{avg_fr_H['Speech']:.2f} Hz")

# Printing average firing rates for Region A
print("Region A - Average Firing Rates:")
print(f"Face: {avg_fr_A['Face']:.2f} Hz | Text: {avg_fr_A['Text']:.2f} Hz | Speech:
{avg_fr_A['Speech']:.2f} Hz")
```

Graph Output:

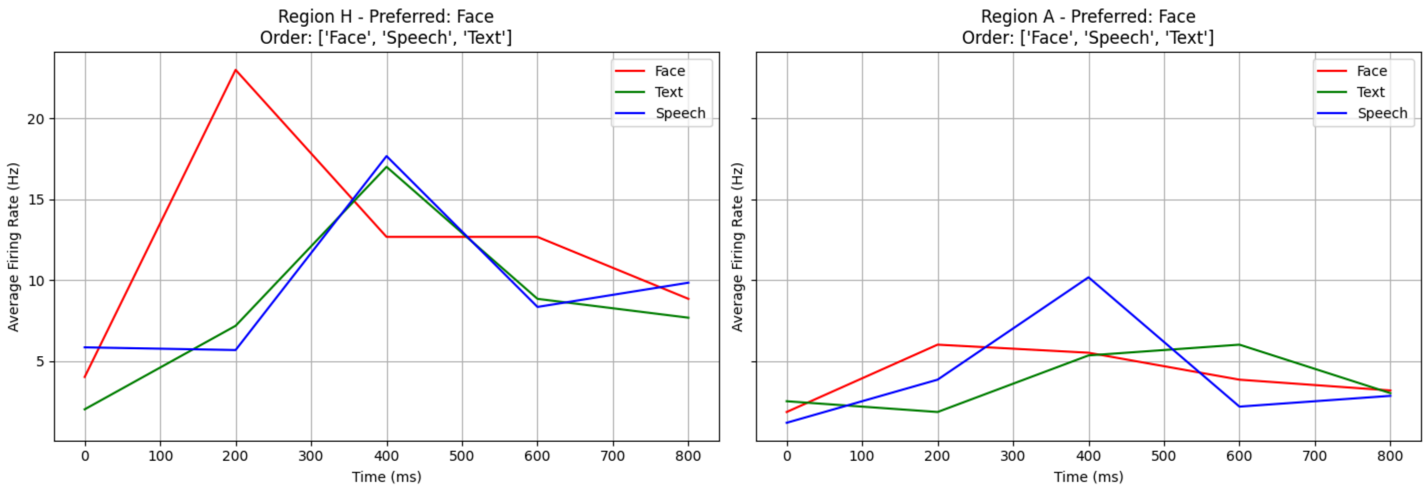

Figure 1.3 Two neurons' average firing rates in response to three different stimulus types face, text, and speech—recorded from two different brain regions (Region H and Region A). With a bin width of 200 ms, the firing rates are calculated over a window of 0-1000 ms. The titles of the plots, which were created by averaging the corresponding curves, show the preferred stimulus type and the stimulus preference order for each neuron in descending order (the most favored stimulus being the first one). The response to a face stimulus is shown by a red line, to a text stimulus by a green line, and to a speech stimulus by a blue line. The most favored stimulus type and the relative preference order for every neuron are shown in this visualization.

Output on the Console:

Region H - Average Firing Rates:
Face: 12.23 Hz | Text: 8.53 Hz | Speech: 9.47 Hz
Region A - Average Firing Rates:
Face: 4.07 Hz | Text: 3.73 Hz | Speech: 4.03 Hz

Answer to Question:

These neurons primarily reflect Rate Coding as a type of neural information coding. The characteristic of rate coding is that firing rates fluctuate over time and are elicited by a variety of stimuli. Given that the 'Face' stimulus causes a higher firing rate than the 'Text' and 'Speech' stimuli, the neuron in both areas in this instance exhibits a preference for the Face stimulus. The differences in firing rates are quite large in Neuron H and the disparities in firing rates between the Face and Speech stimuli are rather small in Neuron A, but the Text stimulus still lags far behind in Neuron A.