Report

问题1

▼ 代码

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is the script for Problem 1.
% Your task is to fill in a few lines in this file
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load c2p3.mat
stim=double(stim);  % Convert stim to double
window=12;          % 12 steps; each step corresponds to 15

%----------calculate the spike-triggered average --------
n=length(counts);    % Number of time bins
C=zeros(16,16,window);

% Loop through all time bins to calculate the spike-trigger
for t=window:n
    if counts(t) > 0  % Only consider bins where there are
        for tau=1:window
            C(:,:,tau) = C(:,:,tau) + counts(t) * stim(:,:,
        end
    end
end

% Normalize the results by the total number of spikes
total_spikes = sum(counts(window:end));  % Sum of spikes af
C = C / total_spikes;

% Create the directory to save figures if it doesn't exist
if ~exist('../Figure', 'dir')
   mkdir('../Figure');
end
```

```
%---------- plot the results ---------------
figure(1)
for i=1:window
    subplot(3,4,i);
    imagesc(C(:,:,i));
    colormap(gray)
    str = sprintf('tau=%d*15.6 ms',window-i); % Plot in rev
    title(str)
    axis off
end
% Save the figure for spike-triggered averages
saveas(gcf, '../Figure/spike_triggered_average.png')

%----- summing up the images across one spatial dimension -
% Sum the images along one dimension (e.g., along x-axis)
summed_C = squeeze(sum(C, 1));

% Plot the summed results (3D mesh plot)
figure(2)
mesh(summed_C); % 3D plot
xlabel('y dimension');
ylabel('Time step');
zlabel('Summed value');
title('Summed response across x dimension');
colorbar;  % Add colorbar for reference

% Save the 3D mesh plot
saveas(gcf, '../Figure/summed_response_3D.png')

%----- plot the contour map -----
figure(3)
contour(summed_C, 20);  % Generate a contour plot with 20 c
xlabel('y dimension');
ylabel('Time step');
title('Contour of summed response across x dimension');
colorbar;

% Save the contour plot
```

```
saveas(gcf, '../Figure/summed_response_contour.png')

return
```

问题描述

在该问题中，我们使用 MATLAB 处理了一个包含猫的外侧膝状体（LGN）细胞对二维视觉图像响应的数据集。目标是计算脉冲触发平均图像，并可视化不同时间步长下的图像。

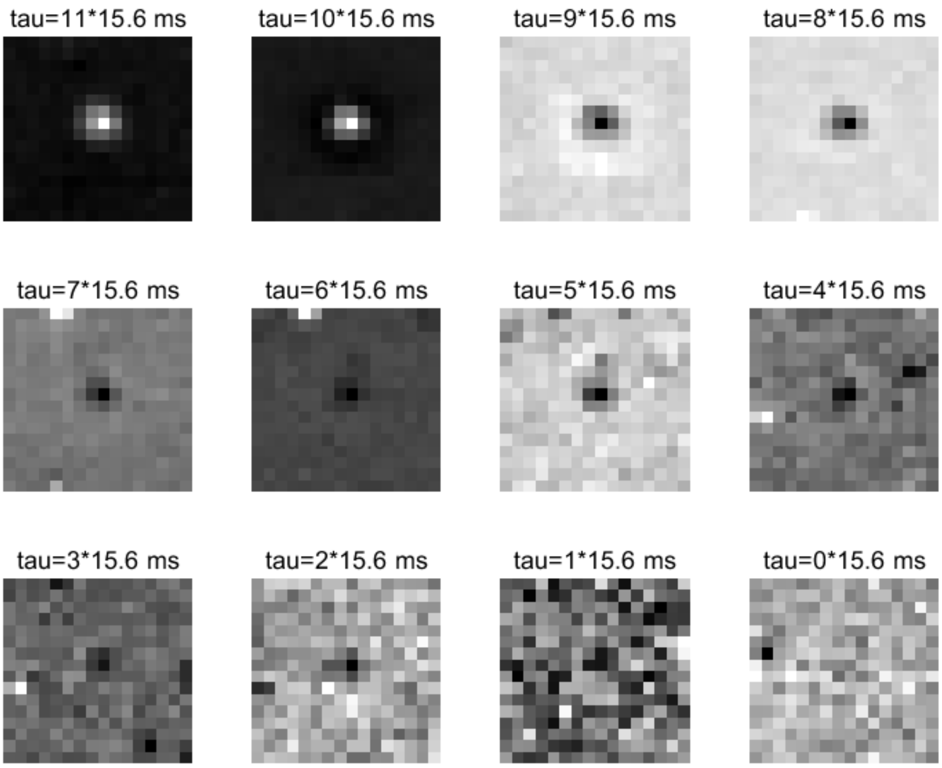
方法

1. **数据加载**：加载数据文件 `c2p3.mat`，并将刺激图像数组转换为 `double` 类型。
2. **脉冲触发平均的计算**：
    - 对于每个脉冲，计算其前 12 个时间步的平均图像。
    - 使用脉冲的数量对刺激图像进行加权。
3. **结果可视化**：
    - 使用 `imagesc` 命令绘制脉冲触发平均图像，并显示各个时间步的变化。
    - 对图像在一个空间维度上求和，绘制三维图和等高线图。

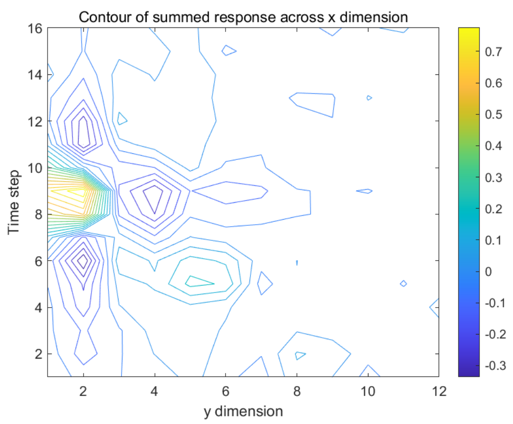
结果

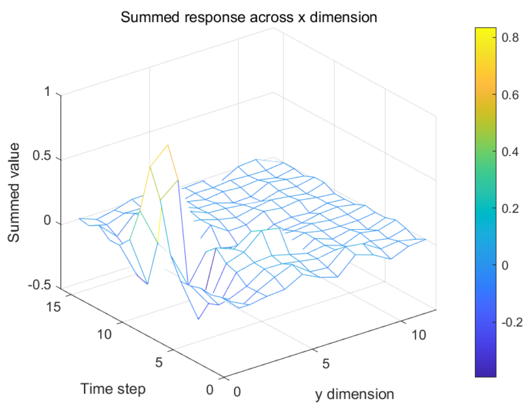
我们得到了 12 张脉冲触发平均图像，展示了随时间变化的视觉响应。此外，绘制了三维图和等高线图，进一步分析了刺激的空间响应。

1. 脉冲触发平均图像



2. 沿着一个空间维度对图像求和





问题2

证明如果从与方向无关的概率分布中均匀地选择无限多个单位向量 $\vec{c}_a$，则有：

$$\sum_{a=1}^{N}(\vec{v}\cdot\vec{c}_a)\vec{c}_a \propto \vec{v}$$

证明步骤：

1. **向量分解**：

   设 $\vec{v}$ 是任意向量， $\vec{c}_a$ 是单位向量：

   $$\vec{c}_a = \frac{\vec{v}}{|\vec{v}|} + \vec{c}_a^{\perp}$$

   其中 $\vec{c}_a^{\perp}$ 是垂直于 $\vec{v}$ 的分量。

2. **计算和式**：

   $$\sum_{a=1}^{N}(\vec{v}\cdot\vec{c}_a)\vec{c}_a = \sum a = 1^N |\vec{v}|\cos(\theta_a)\vec{c}_a$$

   其中 θ_a 是 $\vec{v}$ 与 $\vec{c}_a$ 之间的夹角。

3. **简化和式**：

   $$= |\vec{v}|\sum_{a=1}^{N}\cos(\theta_a)\vec{c}_a$$

4. **利用均匀分布的性质**：

   由于 \(\vec{c}_a \) 均匀分布，垂直于 \(\vec{v} \) 的分量会抵消：

   $$\sum_{a=1}^{N}\cos(\theta_a)\vec{c}_a \to \text{平行于 } \vec{v}$$

5. **得出结论**：

   $$\sum_{a=1}^{N}(\vec{v}\cdot\vec{c}_a)\vec{c}_a \propto \vec{v}$$

总结：

$$\sum_{a=1}^{N}(\vec{v}\cdot\vec{c}_a)\vec{c}_a \propto \vec{v}$$

问题3

问题 3：最大后验估计 (MAP)

假设一组神经元的调谐曲线是高斯函数：

$$f_a(s) = r_{\max}\exp\left(-\frac{1}{2}\left(\frac{s-s_a}{\sigma_a}\right)^2\right)$$

并且这些曲线在 \$s\$ 值的范围内均匀且密集地分布。我们已经获得了刺激的最大似然 (ML) 估计：

$$s_{\mathrm{ML}} = \frac{\sum_a r_a s_a/\sigma_a^2}{\sum_a r_a/\sigma_a^2}$$

如果刺激的先验分布是均值为 s_{prior} 和方差为 σ_{prior}^2 的高斯函数，求刺激的最大后验概率 (MAP) 估计。

解：

1. **定义后验分布**：

根据贝叶斯定理，后验分布 $P(s|\text{data})$ 与似然函数和先验分布的乘积成正比：

$$P(s|\text{data}) \propto P(\text{data}|s)P(s)$$

这里，$P(\text{data}|s)$ 是似然函数，\$P(s)\$ 是先验分布。

1. **计算似然函数**：

根据神经元的调谐曲线，给定 s，每个神经元的脉冲率为 \$r_a\$，因此似然函数可以表示为：

$$P(\text{data}|s) = \prod_a f_a(s) = \prod_a r_{\max}\exp\left(-\frac{1}{2}\left(\frac{s-s_a}{\sigma_a}\right)^2\right)$$

为了简化计算，我们取对数：

$$\log P(\text{data}|s) = \sum_a \log f_a(s) = \sum_a \left(\log r_{\max} - \frac{1}{2} \left(\frac{s - s_a}{\sigma_a} \right)^2 \right)$$

1. **计算先验分布**：

先验分布是高斯函数，其形式为：

$$P(s) = \frac{1}{\sqrt{2\pi\sigma_{\text{prior}}^2}} \exp\left(-\frac{1}{2} \left(\frac{s - s_{\text{prior}}}{\sigma_{\text{prior}}} \right)^2 \right)$$

同样，取对数：

$$\log P(s) = -\frac{1}{2} \log(2\pi\sigma_{\text{prior}}^2) - \frac{1}{2} \left(\frac{s - s_{\text{prior}}}{\sigma_{\text{prior}}} \right)^2$$

1. **计算后验分布的对数**：

将似然和先验结合起来，得到后验分布的对数：

$$\log P(s|\text{data}) \propto \sum_a \left(\log r_{\max} - \frac{1}{2} \left(\frac{s - s_a}{\sigma_a} \right)^2 \right) - \frac{1}{2} \left(\frac{s - s_{\text{prior}}}{\sigma_{\text{prior}}} \right)^2$$

1. **最大化后验分布**：

为了找到最大后验估计，我们可以对 $\log P(s|\text{data})$ 关于 s 进行求导并设其为零：

$$\frac{d}{ds} \left(\sum_a \left(-\frac{1}{2} \left(\frac{s - s_a}{\sigma_a} \right)^2 \right) - \frac{1}{2} \left(\frac{s - s_{\text{prior}}}{\sigma_{\text{prior}}} \right)^2 \right) = 0$$

1. **计算导数并解方程**：

求导后，得到：

$$\sum_a \frac{s - s_a}{\sigma_a^2} + \frac{s - s_{\text{prior}}}{\sigma_{\text{prior}}^2} = 0$$

整理后得：

$$s_{\text{MAP}} = \frac{\sum_a \frac{s_a}{\sigma_a^2} + \frac{s_{\text{prior}}}{\sigma_{\text{prior}}^2}}{\sum_a \frac{1}{\sigma_a^2} + \frac{1}{\sigma_{\text{prior}}^2}}$$

结论：

刺激的最大后验概率 (MAP) 估计为：

$$s_{\mathrm{MAP}} = \frac{\sum_a \frac{s_a}{\sigma_a^2} + \frac{s_{\mathrm{prior}}}{\sigma_{\mathrm{prior}}^2}}{\sum_a \frac{1}{\sigma_a^2} + \frac{1}{\sigma_{\mathrm{prior}}^2}}$$

这个公式结合了来自神经元的测量信息与先验知识。

问题4

▼ 代码

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is the script for Problem 4.
% Your task is to fill in a few lines in this file
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;

% cricket's coordinates
phi = pi/4:pi/2:7*pi/4; % preferred angles of the 4 interne

N_theta = 500;  % number of samples
N_trial = 1000; % number of trials
error = zeros(1, N_theta);  % results
thetas = -pi/2:pi/(N_theta-1):pi/2; % ground truth

for i = 1:N_theta
    theta = thetas(i);
    % Generate average firing rates for the 4 interneurons
    r_mean = 50 * cos(theta - phi);  % Mean firing rates
    r_mean(r_mean < 0) = 0;  % Set negative rates to zero

    % Initialize actual firing rates with noise
    r_actual = r_mean + randn(1, 4) * 5;  % Add Gaussian no
    r_actual(r_actual < 0) = 0;  % Ensure non-negative firi

    % Calculate the x and y components of the population ve
    x = sum(r_actual .* cos(phi));
```

```
    y = sum(r_actual .* sin(phi));

    % Estimate the wind direction
    theta_est = atan2(y, x);  % Estimate the direction in r

    % Convert theta and theta_est to degrees for error calc
    error(i) = error(i) + (theta/pi*180 - theta_est/pi*180)
end

% Average the error over trials
error = sqrt(error / N_trial);

% Plot the results
figure(1);
plot(thetas * 180 / pi, error, 'LineWidth', 2);
ylim([0 1.5]);
xlabel("\theta (degree)");
ylabel("error (degree)");
title("Error in Wind Direction Estimation");
grid on;

% Create the directory to save figures if it doesn't exist
if ~exist('../Figure', 'dir')
    mkdir('../Figure');
end

% Save the figure
saveas(gcf, '../Figure/wind_direction_error.png');
```

问题描述

在该问题中，我们模拟了蟋蟀神经元对风向的响应，并使用向量解码方法估计真实的风向。目标是通过模拟的脉冲频率，检验风向的估计精度。

方法

1. **参数设置**：定义蟋蟀的偏好角度和试次数量。

2. **风向估计**：
    - 对于每个真实风向，生成四个神经元的平均脉冲频率，并添加高斯噪声。
    - 计算群体向量的 x 和 y 分量，并使用 `atan2` 函数估计风向。
3. **误差计算**：
    - 计算估计风向与真实风向之间的误差，并求取均方根误差（RMSE）

结果

绘制了估计风向误差与真实风向之间的关系图。结果显示，总体误差控制在一个很不错的范围之内。