

Contents

- [let's start fresh!](#)
- [Q1: MATLAB sum, element-wise multiplication](#)
- [Q1b](#)
- [Q1c dot product](#)
- [load the data](#)
- [inspect some elements of the neural responses](#)
- [plot the images](#)
- [plot the target image \(Q2\)](#)
- [compute STA \(Q3\)](#)
- [Write in the comments your thoughts below \(Q3\)](#)
- [load and plot the test image \(for Q4\)](#)
- [plot the convolved results for each STA](#)
- [Write in the comments your thoughts below \(Q4\)](#)
- [load and plot the test image \(for BONUS\)](#)
- [plot the convolved results for each STA](#) %% This code is returning an error for the Bonus
- [Write in the comments your thoughts below \(BONUS - cannot run the section above\)](#)

```
% This is a template for HW2 for BIO 347 / NEU 547 (2020 Fall)
% You'll need to fill in missing parts of this script. (marked as % TODO)
% You are not required to change anything but the lines marked with TODO
% Make sure to save this script in the same folder as the other files:
% * dataset_rf.mat
% * testing2_stop.jpg
% * plotConvSTA.m
```

let's start fresh!

```
clear % clear workspace variables
close all % close all figures
```

Q1: MATLAB sum, element-wise multiplication

```
v = [0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1];
x = sum(v)
% Q1a: I think sum(v) computes the sum of all the values in the row vector,
% which in this case is 5. Since the vector is only 0's and 1's, it also
% indicates the number of 1's in the vector.
```

```
x =
```

```
5
```

Q1b

```

A = [3, 5, 0; -1, -4, 2]; % a 2-by-3 matrix
B = [0, -1, 0; -1, 2, 2];
D = zeros(2, 3);
for k1 = 1:2 % for each row
    for k2 = 1:3 % for each column
        D(k1, k2) = A(k1, k2) + B(k1, k2); % (Q1b)
    end
end
disp(D)
disp('should be equal to this:')
disp(A + B)

```

```

3     4     0
-2    -2     4

```

should be equal to this:

```

3     4     0
-2    -2     4

```

Q1c dot product

```

R = [-3, 0, 0; 0, 0, 5; 0, 0, 0]; % receptive field I
S = [ 4, -1, 1; 1, -1, -2; -1, 0, 2]; % stimulus A
disp(R)
disp(S)
dot_prod = sum((R .* S), 'all'); % (Q1c)
disp(dot_prod)

```

```

-3     0     0
 0     0     5
 0     0     0

```

```

4    -1     1
1    -1    -2
-1     0     2

```

-22

load the data

```

load('dataset_rf');
whos

```

Name	Size	Bytes	Class	Attributes
A	2x3	48	double	
B	2x3	48	double	

D	2x3	48	double
R	3x3	72	double
S	3x3	72	double
dot_prod	1x1	8	double
k1	1x1	8	double
k2	1x1	8	double
neuralResponses	8x10000	640000	double
stimMovie	25x25x10000	50000000	double
v	1x14	112	double
x	1x1	8	double

inspect some elements of the neural responses

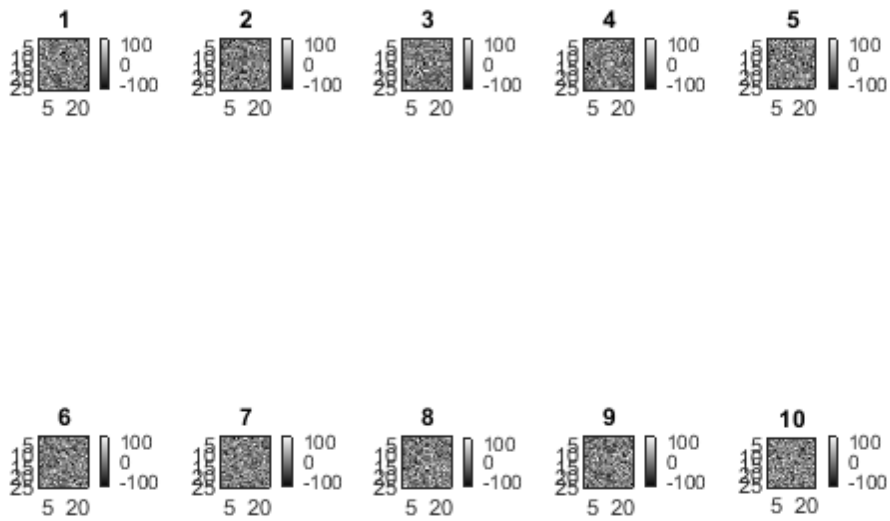
```
y = neuralResponses(2, 1:12); % (Q2)
disp(y)
```

```
0    0    0    0    0    0    0    0    0    1    0    1
```

plot the images

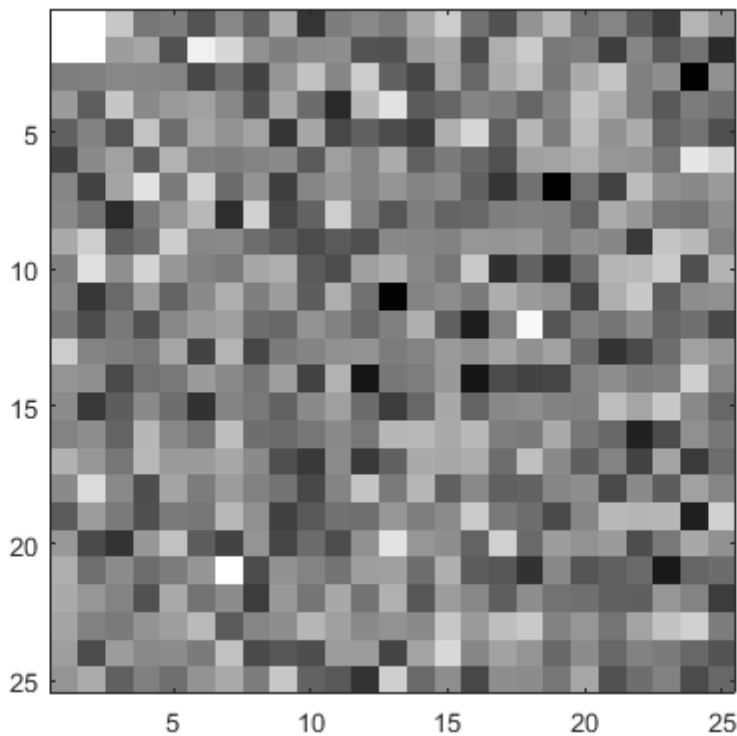
```
figure(99); clf; colormap('gray'); % figure number 99, let's use gray

for t = 1:10 % run the next block for the first 10 time steps
    subplot(2, 5, t); % let's put several plots in one figure
    frame = stimMovie(:, :, t); % take one 25 by 25 frame at time t
    imagesc(frame); % plot the pixels stored in the matrix 'frame'
    caxis([-128, 128]); % each pixel intensity
    title(t);
    colorbar;
    axis square;
end
```



plot the target image (Q2)

```
figure(9292); clf; colormap('gray');  
frame = stimMovie(:, :, 12); % (Q2)  
imagesc(frame);  
axis square;
```



compute STA (Q3)

```

nNeuron = 8; % no of neurons; same as size(neuralResponses, 1)
nT = 10000; % no of presented images; same as size(neuralResponses, 2)
nx = 25;    % same as size(stimMovie, 1)
ny = 25;    % same as size(stimMovie, 2)

figure(347); clf; colormap('gray');
staSaved = cell(nNeuron, 1); % we'll save the results to use later

for kNeuron = 1:nNeuron
    sta = zeros(nx, ny); % initialize with zeros
    y = neuralResponses(kNeuron, :); % call the kNeuron's response y

    for t = 1:nT % for each frame of the movie
        frame = stimMovie(:, :, t);
        sta = sta + frame * y(t); % TODO (Q3) % summation
    end

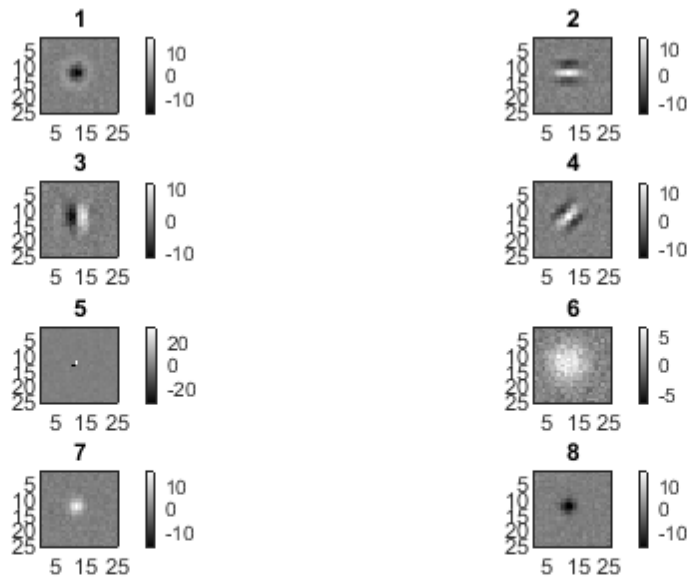
    sta = sta / sum(y, 'all'); % TODO (Q3) % normalization

    % plotting code
    subplot(5,2,kNeuron);
    imagesc(sta);
    axis square
    caxis([-1, 1] * max(abs(sta(:))));
    colorbar;
    title(kNeuron)

    % save your sta to use later (for Q4)

```

```
staSaved{kNeuron} = sta;
end
```



Write in the comments your thoughts below (Q3)

STA1: This neuron responds to dark points somewhat surrounded by lightness.

STA2: This neuron responds to light horizontal lines surrounded by darkness.

STA3: This neuron responds to sharp vertical edges between light and dark.

STA4: This neuron responds to light diagonal lines surrounded by darkness.

STA5: This neuron responds to single light or dark pixels.

STA6: This neuron responds to large areas of light in the image.

STA7: This neuron responds to more precise areas of light in the image.

STA8: This neuron responds to the darkest areas of the image.

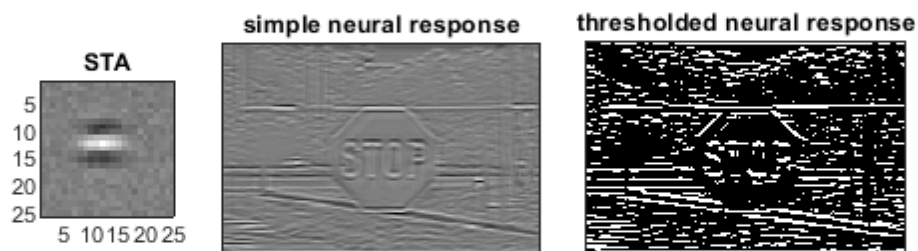
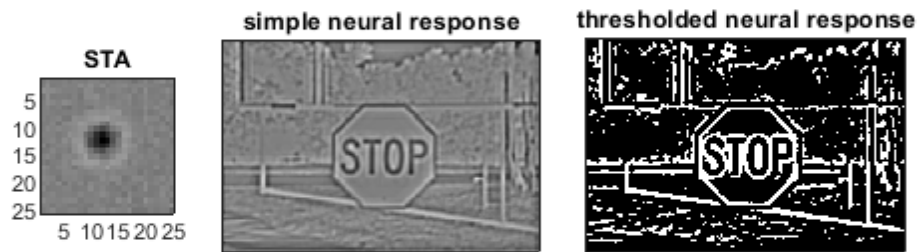
load and plot the test image (for Q4)

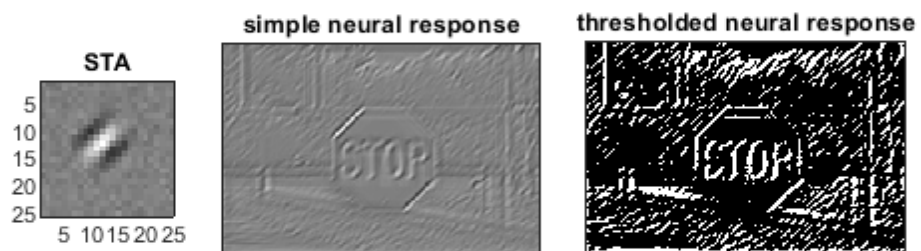
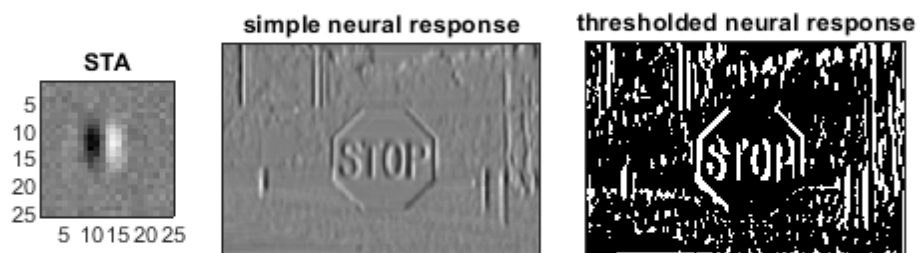
```
img = imread('testing2_stop.jpg');
figure(331); clf; colormap('gray');
image(img);
axis equal; axis tight;
set(gca, 'XTick', [], 'YTick', []);
title('Original image');
```

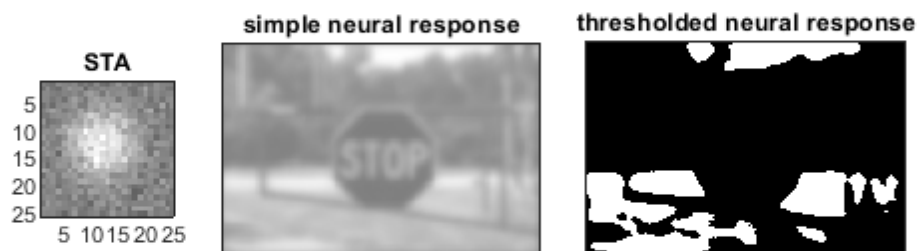
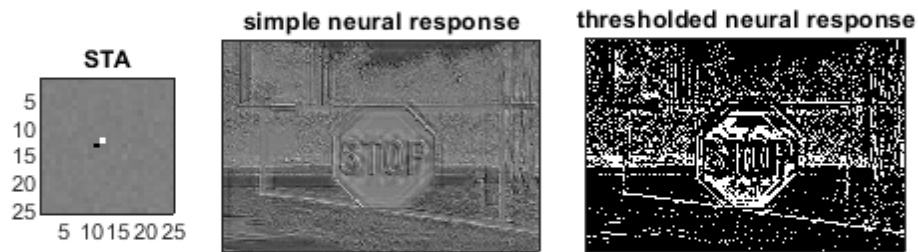
Original image

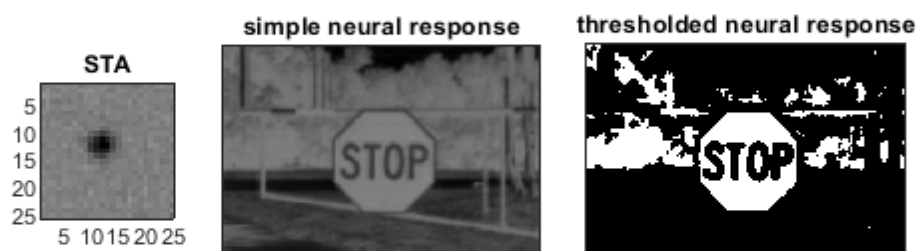
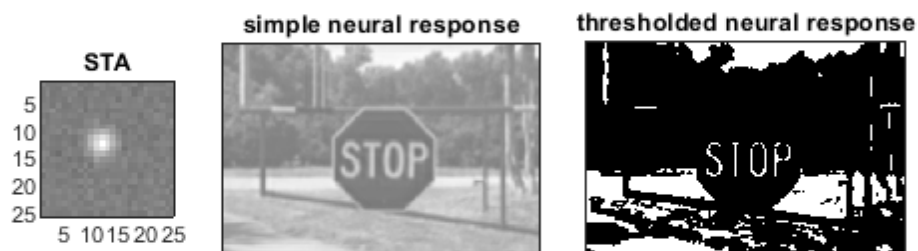
**plot the convolved results for each STA**

```
for kNeuron = 1:nNeuron
    sta = staSaved{kNeuron}; % we saved this above, let's use them now
    figure(906+kNeuron); clf;
    plotConvSTA(sta, img); % TODO (Q4)
end
```









Write in the comments your thoughts below (Q4)

STA1: This RF picked up dark lines and points surrounded by light in the photo, such as dark railings and borders.

STA2: This RF picked up the horizontal lines in the photo, specifically light railings and borders surrounded by dark.

STA3: This RF picked up the vertical edges between dark and light areas of the photo, specifically steep changes between dark and light.

STA4: This RF picked up the diagonal lines in the photo, specifically light railings and borders surrounded by dark. The closer the angle of the line in the image to that shown in the RF, the stronger the signal. (This was true for all RFs that highlighted lines, but especially obvious in this one).

STA5: This RF picked up singular pixels of extreme darkness and lightness, especially in the pattern shown where a light pixel is diagonally above and to the right of a dark pixel.

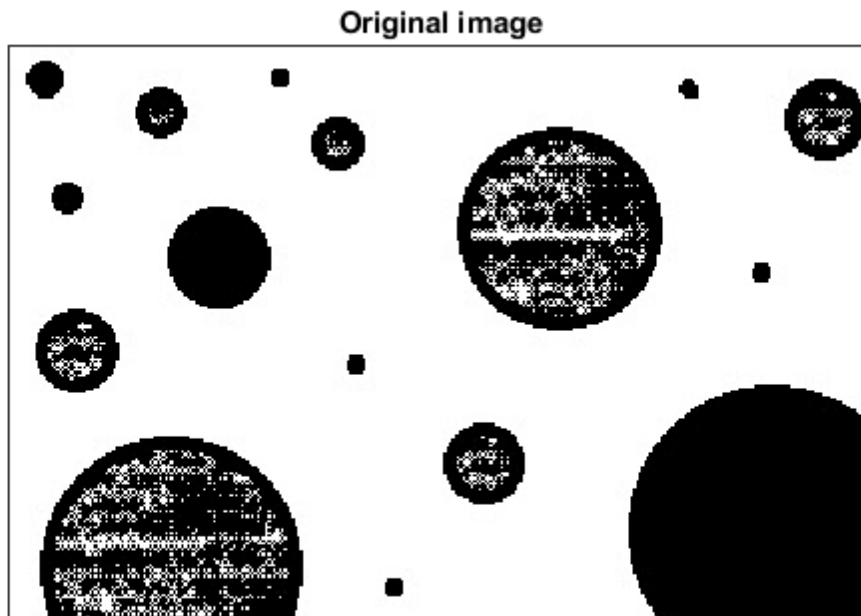
STA6: This RF picked up large areas of light pixels in the photo, such as the sidewalks and sky. The RF is simialar to a gaussian blur, thus it made the simple neural response output image appear blurry.

STA7: This RF picked up only the lightest areas of the photo, and appears to have generally lightened the simple neural response output image.

STA8: This RF picked up only the darkest areas of the photo, and seemingly inverted the colors to produce the simple neural response output image.

load and plot the test image (for BONUS)

```
img = imread('space2.jpg');
figure(331); clf; colormap('gray');
image(img);
axis equal; axis tight;
set(gca, 'XTick', [], 'YTick', []);
title('Original image');
```



plot the convolved results for each STA %% This code is returning an error for the Bonus

```
for kNeuron = 1:nNeuron
    sta = staSaved{kNeuron}; % we saved this above, let's use them now
    figure(906+kNeuron); clf;
    plotConvSTA(sta, img); % TODO (BONUS)
end
```

Error using conv2
N-D arrays are not supported.

Error in plotConvSTA (line 24)
 cimg = conv2(img, sta, 'same');

Error in hw2template (line 164)
 plotConvSTA(sta, img); % TODO (BONUS)

Write in the comments your thoughts below (BONUS - cannot run the section above)

STA1:

STA2:

STA3:

STA4:

STA5:

STA6:

STA7:

STA8:

Published with MATLAB® R2022a