Matlab, part 2: 2+D matrices, visualizations

Matrix indexing

Assume we have a 10x500 matrix of spike patterns for 10 neurons spikeMat

- spikeMat(1,:) contains spikes for neuron 1
- spikeMat(4,:) contains spikes for neuron 4

In general:
- name(:,col) accesses all elements in column

Matrices: rows and columns

B=[2.2 1.4; -5 3.5; -7.8 4.3];

- Spaces/commas separate columns:
- Semi-colons (;) separate rows
- name(row,col) accesses single element:
  B(2,1) returns -5

Matrices in n dimensions

x=[1 2 3; 4 5 6]   y(:,:,1)=[1 2; 3 4]
y(:,:,2)=[5 6; 7 8] y(:,:,3)=[9 10; 11 12]
size(y) ->[2 2 3]

Typical brain data: location of neurons (x,y,z) or location of neurons + time
Heat-maps

imagesc(Data) – view 2D matrix of scaled data as image
  • Red/yellow is highest value, blue is lowest value

Visualize a 2D slice of brain data (size(brainData) -> 128x128x88)
slice=squeeze(brainData(:,:,20))
  -> slice 20 of brain
imagesc(slice)

squeeze out 1-entry dimensions

currSlice=squeeze(brainData(:,60,:));
  % currSlice has size 128x1x88
  % won’t be plotted by imagesc
  % - expects 2D matrix

currSlice=squeeze(currSlice);
  % now currSlice has size 128x88

Multiple plots

figure -> opens new plotting window

subplot(r,c,i) -> creates grid of plots with
  • r rows
  • c columns
  • fill in position i

subplot(2,1,1); plot(vec1);
subplot(2,1,2); plot(vec2);

squeeze out 1-entry dimensions

currSlice=squeeze(brainData(:,60,:));
  % currSlice has size 128x1x88
  % won’t be plotted by imagesc
  % - expects 2D matrix

currSlice=squeeze(currSlice);
  % now currSlice has size 128x88

figure -> opens new plotting window

subplot(r,c,i) -> creates grid of plots with
  • r rows
  • c columns
  • fill in position i

currSlice=squeeze(brainData(:,60,:));
  % currSlice has size 128x1x88
  % won’t be plotted by imagesc
  % - expects 2D matrix

currSlice=squeeze(currSlice);
  % now currSlice has size 128x88

Multiple plots

figure -> opens new plotting window

subplot(r,c,i) -> creates grid of plots with
  • r rows
  • c columns
  • fill in position i

subplot(2,1,1); plot(vec1);
subplot(2,1,2); plot(vec2);
Saving graphics results

• print -dpng filename.png
• print -djpg filename.jpg

Multiple plots

With looping

for i=1:32,
currSlice=squeeze(brainData(:,i*4,:));
subplot(4,8,i), imagesc(currSlice);
end;

Scaling vs. not-scaling

imagesc(Data) – view 2D matrix of scaled data as image
• Yellow (or red) is highest value, blue is lowest value
image(Data) – view 2D matrix of data as image
• Yellow (or red) is >=256 (or 64), blue is <=0

slice=squeeze(brainData(:,:,10));
figure; imagesc(slice);
vs
figure; image(slice)

Finding desired values

find(vector<number) find(c<2)
Return indices in vector that are less than number
Example: vector=[5, -1, 0, 12];
smallLocations=find(vector<2);

Comparisons
• d<2, d>2 strict inequality
• d<=2, d>=2 semi-inequality
• d==2 equality
Computing mask matrix

% grab slice
slice=squeeze(S1brain(:,44,:));
% find body inds
hiValues = find(slice>=50);

% create new mask, all 0 by default
maskMat = zeros(128,88);
% add in 1s
maskMat(hiValues)=1;
% will have inner circle of 0s

% isolate column 40 in slice
oneColumn = maskMat(:,40);
% find zeros in column 40
zeroInds = find(oneColumn==0);
% find the front 0 of brain
frontHalf = find(zeroInds<=64);
frontInd = zeroInds(end);
% find the back 0 of brain
backHalf = find(zeroInds>64);
backInd = zeroInds(1);
% set non-brain data to 0
slice(1:frontInd,40) = 0;
slice(backInd:end,40) = 0;

How can we loop for all cols?

for col=1:88,
    oneColumn = maskMat(:,col);
    zeroInds = find(oneColumn==0);
    frontHalf = find(zeroInds<=64);
    frontInd = zeroInds(end);
    backHalf = find(zeroInds>64);
    backInd = zeroInds(1);
    slice(1:frontInd,col)=0;
slice(backInd:end,col)=0;
end;