Theoretical Neuroscience I

Exercise 6: Tuning curves and receptive fields

Due: Wednesday, 11 December 2019

1 Introduction

This exercise simulates neurophysiological experiments in the visual cortex of cat. You will simulate visually (with images) and collect neuronal responses. By comparing the responses to different kinds of stimuli, you will characterise the 'responsiveness' or 'stimulus selectivity' of each neurone. Your specific goals are to map the 'receptive fields' and to establish some 'tuning curves' of your neurones.

2 Matlab programmes

You have received two Matlab functions - **MysteriousNeuron1.m**, **MysteriousNeuron2.m** - which implement unknown neurone. Each function accepts as input a two-dimensional stimulus array S and produces as output a scalar response r, i.e. the firing rate in Hz:

```
>> r = MysteriousNeuron( S );
```

You have also received Matlab functions that produce a variety of stimulus arrays S, including ON-contrast spots, OFF-contrast spots, ON-contrast bars, OFF-contrast bars, and white noise. The format of these functions is as follows:

`>> S = OnSpot( n );`	n = integer array size
`>> S = OffSpot( n );`	
`>> S = OnBar( n, theta );`	theta = bar orientation θ in radians
`>> S = OffBar( n, theta );`	
`>> S = WhiteNoise( n );`	

All stimulus functions produce a stochastic result, in other words, a different image each time the function is called : spots and bars change position, and white noise changes all pixel values. Examples of the output of stimulus functions can be generated and viewed by executing the function **ViewStimuli.m**.

3 Assignment

Characterize the responsiveness of each neurone by stimulating with multiple images of size $n = 50$ and by collecting the resulting responses. Note that you have to adjust the number of stimulus repetitions such as to obtain a valid result. White noise stimulation will have to be repeated more often than dot or bar stimulation.

- Stimulating repeatedly (n times) with *bright bars* (**OnBar**) with one particular orientation θ, establish the average response $\langle r(\theta) \rangle$, average square response $\langle r^2(\theta) \rangle$, and standard deviation of the response $\sigma_r(\theta)$ to bars of this orientation! Repeat for approximately 12 different orientations to establish a 'tuning curve':

$$\langle r(\theta) \rangle = \frac{1}{n} \sum_{i=1}^{n} r_i, \qquad \langle r^2(\theta) \rangle = \frac{1}{n} \sum_{i=1}^{n} r_i^2, \qquad \sigma_r(\theta) = \sqrt{\langle r^2 \rangle - \langle r \rangle^2}$$

  Plot the average response $\langle r(\theta) \rangle$, average plus standard deviation $\langle r(\theta) \rangle + c_v(\theta)$, and average minus standard deviation $\langle r(\theta) \rangle + c_v(\theta)$, as a function of θ. What do you conclude about the response properties of each neurone?

- Repeat all steps with *dark bars* (**OffBar**)! (You can use the same programme, sampling replacing OnBar with OffBar.)

- Stimulating repeatedly (k times) with *bright spots* (**OnSpot**) at varying positions and obtain the response-weighted average of the **stimulus images**! In other words, combine stimulus images S_i and associated responsse r_i according to the formula

$$\bar{S} = \frac{\sum_{i=1}^{k} S_i \, r_i}{\sum_{i=1}^{k} r_i}$$

  Plot the resulting average image using Matlab-function **pcolor** and discuss your conclusions for both neurones!

- Repeat with *dark spots* (**OffSpot**) and with *white noise* (**WhiteNoise**)! Plot the resulting average images and discuss your conclusions, not in in particular any differences between stimulus types and between the two neurones.

- Discuss the advantages and disadvantages of using different stimulus types! Consider the number of trials needed and the amount of detailed information gained!

4 Optional assignment (advanced)

Consider the possibility that one or both of your neuronal responses reflect more than one linear receptive field. In this case, the computation of a response-weighted average will fail (i.e., show no structure). However, the computation of a response-weighted **covariance** may succeed, as the principal eigenvectors of this matrix may reveal the contributing receptive fields:

$$S_n \qquad \textit{stimulus pixel vector} \qquad [n \cdot n, 1]$$

$$C_{nm} = S_n \cdot S_m \qquad \textit{stimulus covariance matrix} \qquad [n \cdot n, n \cdot n]$$

$$\bar{S} = \frac{\sum_i r_i \cdot S_i}{\sum_i r_i} \qquad \textit{response weighted average stimulus} \qquad [n, n]$$

$$\bar{C}_{nm} = \frac{\sum_i r_i \cdot C_{nm_i}}{\sum_i r_i} \qquad \textit{response weighted average covariance} \qquad [n \cdot n, n \cdot n]$$

where i is an index over trials and n, m are indices over stimulus pixels.

Obtain eigenvectors and eigenvalues of the weighted covariance matrix with the Matlab command (see Matlab help for details):

[V,D] = eig(Cnm)

View the principal eigenvectors as receptive fields, using the Matlab function **reshape()** to sort the pixelvalues of the eigenvectors back into the stimulus area. What does the response-weighted covariance tell you about mysterious neuron 2?