

Problem Set 3 (due Thursday, Feb. 27) Visual receptive fields and least squares

Feb. 20, 2003

1. Difference of Gaussians in one dimension. In class I told you about some of the properties of Gaussians, and how they are used to model the spatial receptive fields with center-surround structure.

(a) In a future assignment, you will show that the integral of a Gaussian satisfies

$$1 = \int_{-\infty}^{\infty} \frac{dx}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$$

For now, take this result for granted and use it to prove that

$$1 = \int_{-\infty}^{\infty} \frac{dx}{\sqrt{2\pi}\sigma_c} \exp\left(-\frac{x^2}{2\sigma_c^2}\right)$$

(b) Consider the one-dimension difference of Gaussians

$$D(x) = \frac{1}{\sqrt{2\pi}\sigma_c} \exp\left(-\frac{x^2}{2\sigma_c^2}\right) - \frac{B}{\sqrt{2\pi}\sigma_s} \exp\left(-\frac{x^2}{2\sigma_s^2}\right)$$

where $B > 0$. For what value of B is it true that $\int_{-\infty}^{\infty} dx D(x) = 0$?

(c) Using MATLAB, plot $D(x)$ for $\sigma_c = 1$, $\sigma_s = 2$, and $B = 0.7$. Make sure to choose your axes to give the best visualization of the curve. On your own, you can play around with the parameters to see what happens.

(d) For what value of x is the maximum of $D(x)$ achieved? Derive a formula for the values of x that minimize $D(x)$. You can check your formula by comparing with your MATLAB plots.

2. Gabor function in one dimension. The function

$$D(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \cos(kx - \phi)$$

is named after Dennis Gabor, the inventor of holography. x

(a) Using MATLAB, plot $D(x)$ with $k = 2$ and $\sigma = 1$. Try the values $\phi = 0$, $\phi = \pi/4$, and $\phi = \pi/2$.

(b) Show that the local maxima or minima satisfy the equation

$$\frac{x}{k\sigma^2} = -\tan(kx - \phi)$$

(c) How many local maxima/minima exist? x

(d) Does increasing k make them more or less visible? How about increasing σ ? Explain why, in terms of the roles played by the exponential and cosine functions in $D(x)$.

3. Is $\cos(kx - \omega t)$ space-time separable? How about $\cos(kx - \omega t) + \cos(kx + \omega t)$? What about $\exp(kx - \omega t)$? Explain your answers.

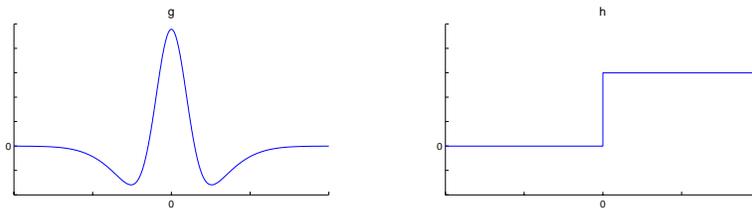


Figure 1: Problem 4(b).

4. More fun with convolution.

- (a) Prove that $\frac{\partial}{\partial t}((g * h)(t)) = ((\frac{\partial g}{\partial t}) * h)(t) = (g * (\frac{\partial h}{\partial t}))(t)$.
 (b) Sketch $g * h$, where g and h are as shown in Fig. 1.

5. Retinal ganglion cell. Download the data file: `ganglion.mat` (available in the assignments section.)

These data were collected in the course of experiments described in P. Kara, P. Reinagel, and R. C. Reid, *Neuron* 27, 635–46 (2000), and generously provided by Prof. Clay Reid of Harvard Medical School.

There are two arrays in the data file:

- `stim` contains a time series of random ± 1 images, which were displayed at a rate of roughly 64 Hz. It has dimensions $16 \times 16 \times 32767$, and contains 32767 images, each of size 16×16 . I'll refer to such a three-dimensional array as an image deck. It's stored as an `int8` array to save space, so you will have to use the `double` operation to convert it to floating point before performing any mathematical operations. If you have limited memory, you won't want to convert it all at once, because that takes up 67 Mbytes. Instead, you'll want to operate on it piece by piece.
- `counts` is a vector containing the spike counts in each time bin. Since 64 Hz is not very fast, often there is more than one spike in a time bin, so this vector is not binary.

Compute the spike-triggered average of the stimulus. The spike-triggered average is defined as follows:

- It is an image deck `c`.
- The first image in `c` is the weighted average of all the images, where each image is weighted by the number of spikes in the same time bin. Note that images in time bins with zero spikes do not contribute to the average at all.
- The second image in `c` is the weighted average of all the images, where each image is weighted by the number of spikes that are *one* time bin in the future.
- The third image in `c` is the weighted average of all the images, where each image is weighted by the number of spikes that are *two* time bins in the future.
- And so on for the rest of the images in `c`. Make your answer a $16 \times 16 \times 12$ image deck. There's no point in using larger time lags, for which the weighted average results in pure noise.

There are many ways to accomplish the above in a MATLAB program. Feel free to use whatever method you want, as long as it gets the job done (loops are fine). Your only constraints will be execution time and memory.

Submit your code, along with a figure or figures of the images in `c` (this can be either hard copy or a computer file). In grading, we will look for both a center and surround, as well as a reversal in their signs.

To display a single image, use the `imagesc` command. Experiment with statements like `colormap gray` or `colormap hot` to change the color map. Note that `imagesc` automatically normalizes the image to use the

full colormap. To display many images in one figure, you can use `imagesc` in combination with `subplot`. Or you can use the `montage` command, but to get good results you will have to normalize the images yourself so that the minimum value is zero and the maximum is one.

You will have to be creative in finding a good way of visualizing c . The chief problem is that the surround is weak, and may not show up if you don't normalize correctly. One trick might be to display the positive and negative pixels of c separately, but you may be able to think of better solutions.

6. The first few lectures were devoted towards explaining how to estimate the parameters of linear models. We started with the equations for linear regression and moved on to the more complex Wiener-Hopf equations. In this problem, you'll see that both are instances of the *normal equations*, which are associated with least squares problems.

- (a) Generate a synthetic data set in MATLAB via the commands

```
x=rand(100,1);
y=x+0.1*randn(100,1);
```

These two vectors contain a set of ordered pairs $\{(x_1, y_1), \dots, (x_m, y_m)\}$, where $m = 100$. By construction, these ordered pairs are scattered about a straight line. Use the command `polyfit` to fit a straight line to the data. Do the slope and intercept of the line correspond to what you would expect, given the way that data was generated? Plot the data points and the line on a single graph.

- (b) The fit can also be performed by using the general matrix operations of MATLAB, rather than the specialized function `polyfit`. Convince yourself that a least squares fit to the straight line $y = c_0 + c_1x$ can be expressed as the optimization $\operatorname{argmin}_c |Ac - y|^2$, where the matrix A is defined as

$$A = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{pmatrix},$$

$c = [c_0 \ c_1]^T$, $y = [y_1 y_2 \dots y_m]^T$, and the function $\operatorname{argmin}_x f(x)$ returns the value of x that minimizes $f(x)$. Returning to the MATLAB example, create the matrix A , solve the least squares optimization with the command `A\y`, and compare with the previous result of `polyfit`.

- (c) In general, the least squares optimization is equivalent to solving the *normal equations* $A^T A c = A^T y$. Show that these general equations take the specific form

$$\begin{pmatrix} m & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = \begin{pmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{pmatrix} \quad (1)$$

for the linear regression problem, by calculating $A^T A$ and $A^T y$. These are the linear regression equations that you find in textbooks. Now you see that they are just a special case of the normal equations.

- (d) Suppose that you would like to use a quadratic model $y = c_0 + c_1x + c_2x^2$ instead of a linear model. Find the matrix A such that the least squares optimization can be expressed as $\min_c |Ac - y|^2$. Write down the specific form of the normal equations in this case, which is a 3×3 system of linear equations analogous to (1).
- (e) The Wiener-Hopf equations are also a special case of the normal equations. Look back to your derivation of these equations, and figure out what A , y , $A^T A$, and $A^T y$ correspond to.