CCN Exercises session 4, to be discussed on Thu 18th Feb

Aapo Hyvärinen

9th February 2016

Note that no bonus points are given for these exercises (or any during the second half of the course).

1 Written exercises

1. Show that convolution is a symmetric operation, i.e. $f * g = g * f$.
2. Show Equation (2.3) in the NIS book.
3. Prove Equation (2.17). Hint: Find two different values for x so that you get the two equations

$$A \cos \psi = C \tag{1}$$

$$-A \sin \psi = S \tag{2}$$

   Now, solve for A and ψ as follows. First, take the squares of both sides of both equations (1) and (2) and sum the two resulting equations. Recall the sum of squares of a sine and a cosine function. Second, divide both sides of the equations (1) and (2) with each other.

4. Summarize the basic forms of nonlinear behaviour in simple cells, each with a single word if possible, and a short explanation.
5. Assume a visual neuron is linear. Stimulus A does not elicit any (non-zero) response. Stimulus B does not elicit any response either. Is it possible that the superimposed stimulus A+B elicits a response?
6. Assume a visual neuron is follows the nonlinear model of section 3.4.1, with Equations (3.3) and (3.6). Assume again that Stimulus A does not elicit any response, and Stimulus B does not elicit any response either. Is it possible that the superimposed stimulus A+B elicits a response?
7. Assume Stimulus A elicits no reponse, but Stimulus B does elicit a response. What can we say about Stimulus A+B, for the two models considered above?

2 Computer assignments

Let's assume we have done `import matplotlib.pyplot as plt` and `import numpy as np` in the following.

1. The orthogonality of the Fourier basis is based on the following kinds of properties

$$\int_{-\pi}^{\pi} \sin x \cos x \, dx = 0 \tag{3}$$

$$\int_{-\pi}^{\pi} \sin x \sin 2x \, dx = 0 \tag{4}$$

   where the integral is interpreted as a continuous-valued version of a dot-product. Plot these two integrands and figure out why the integrals are zero. To do the plotting, consider functions `np.arange` and `plt.plot`. At the end you need `plt.show()`.

2. Define a receptive field W so that it corresponds to a 2-D Fourier basis vector, like in Fig. 2.7b) in NIS book. You should produce W as a 100×100 two-dimensional array of the `np.array` class. You may find `np.meshgrid` useful. You can plot W by:

```
img=plt.imshow(W)
img.set_cmap('grayscale')
plt.show()
```

3. Same as above, but produce a Gabor receptive field, a bit like in Figure 3.11a)—play with the parameters so that your receptive field looks as close to this figure as possible. Plot this receptive field as above. From now on, we will use this as W. Produce also an image stimulus I which is a Gabor function but with parameters different from I.

4. Define a function (or, if you don't want to get into python details, some short piece of code) which computes the dot-product between W and I as in Equation (3.1). Now the complication here is that W is two-dimensional, but usually we compute dot-products between one-dimensional vectors.

5. Produce image stimuli A,B which are also of the form of Gabor functions, and which have the behaviours of A, B in the exercises 1.5-1.7, given receptive field W (let's only consider the linear case). Reproduce the results in those exercices numerically.