CN 530: Neural and Computational Models of Vision

Simulation Assignment 2

Note: No part of this assignment involves numerically solving the dynamics of any system of differential equations. Only equilibrium solutions, algebraicly computed, are required.

Item 1: Distance-dependent network

Consider the distance dependent network:

$$\frac{d\boldsymbol{x}_i}{d\boldsymbol{t}} = -\boldsymbol{A}\boldsymbol{x}_i + (\boldsymbol{B} - \boldsymbol{x}_i)\sum_{\boldsymbol{k}=1}^{\boldsymbol{n}} \boldsymbol{I}_k \boldsymbol{C}_{ki} - (\boldsymbol{x}_i + \boldsymbol{D})\sum_{\boldsymbol{k}=1}^{\boldsymbol{n}} \boldsymbol{I}_k \boldsymbol{E}_{ki}$$

To implement this network, either choose sufficiently large *n* so that “border effects” can be ignored or implement a “wrap-around” whereby node *1* is a nearest neighbor of node *n.*

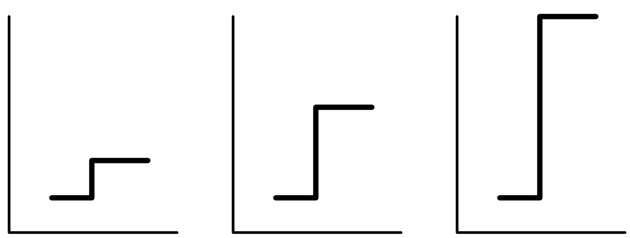
Define coefficients that are Gaussian functions of distance, with on-center, off-surround weights:

$$C_{ki} = C\exp\left[-\mu(k-i)^2\right]$$

$$E_{ki} = E\exp\left[-\nu(k-i)^2\right]$$

You may wish to truncate the kernels (interaction coefficients) to some size much less than that of the total network for computational ease. (Be careful in so doing that you are truncating only “negligible” portions of the curve.) Be sure that your kernels satisfy the noise suppression property numerically -- that is, **after** the truncation. (See Equation 26 of Grossberg, 1983.) Plot your kernels graphically, labeling the excitatory kernel, the inhibitory kernel, and their difference explicitly.

Next, consider a sequence of “step-function” inputs, with non-zero lower parts, such as:



Step functions with same baseline intensity

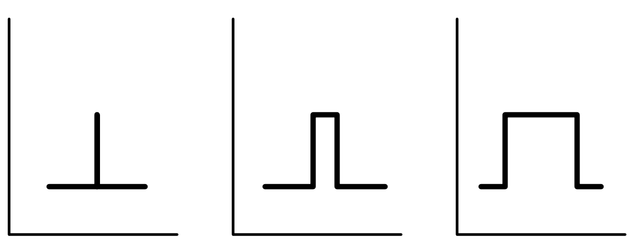
For computations of Item 1, hold the lower portion of the input constant, and generate successively larger inputs on the other part. Calculate the response of your network to the series of input patterns. (As in Item 1, you may wish to vary the numerical values of inputs considerably --that is, by orders of magnitude, rather than just doubling or tripling them.) How does your network's response near the step edge compare with the “reflectances” of the input near the edge? What aspect or aspects of the networks response (e.g. peak amplitude, “trough” amplitude, “thickness” of non-zero response, slope of response at zero-crossing,...) appear to code what aspect or aspects of the input?

Item 2: Reflectance processing in distance-dependent networks

Use the same network as in Item 1. Produce another sequence of plots, hold the proportions (ratios) of the input on the two sides of the step constant, but vary the total input energy over a wide range. First describe candidate “measures” of reflectance processing for your plots. That is, what features of your curves (e.g. height, width, arc-length between inflection points, etc.) is most closely tied to the (Grossbergian) reflectances of the input? Describe quantitatively (numerically) what happens to the *accuracy of reflectance processing* in your simulation as parameters and inputs are varied. You should repeat such sequences of different total energy with same input ratios for *vastly* different decay rate parameters, *A*, relative to the size of your inputs, and describe qualitative changes in network behavior. (This last request is not a “trick question.” If you are not observing qualitative changes in network behavior, you are not varying you parameters, or input, enough.)

Item 3: “Featural noise suppression”

Referring again to the equations of Item 1, fix the interaction coefficients (kernels) so that the “on-center” sums over a small number of inputs. (“Small” here means 3 or more, but much less that half of the network.) Produce an input sequence of “bars” of varying thickness, but equal intensities, again on a non-zero background, like so:



The narrowest bar, whose thickness corresponds to the area covered by *one* network node in some "input registration" layer, should be *insufficient* in amplitude to produce a strong response in the network and the largest (widest) bar should produce *"suppression of homogeneous input"* for at least some network nodes. Describe output for intermediate cases. Repeat the entire sequence of bar thicknesses for bar intensities that are much higher (or much lower) than the original, keeping background intensity constant. *How do the results compare with the original sequence, and why?*

Item 4: Even-symmetric and odd-symmetric kernels:

Return to the inputs described in Item 3 of this assignment. Could "Grossbergian reflectance processing" be better achieved through the use of odd-symmetric, as opposed to even-symmetric kernels? Include some plots for this item and explain in a concise paragraph why this improvement can or cannot be observed.

If you are not familiar with the terms "even-symmetric" and "odd-symmetric" above, consider a function $y = f(x)$ on a Cartesian plane.

If $f(-x) = f(x)$, that function is even-symmetric.

If $f(-x) = -f(x)$, that function is odd-symmetric.

The UMAP article, and your simulations up to this point, deal with even-symmetric functions for Gaussian weightings that are centered over a particular node for purposes of computation. This item asks you to consider replacing a difference of two Gaussians of different widths, but both centered at the same place, by a difference of two Gaussians that are the same width, but whose centers are shifted left or right.

(Note: This item is specifically designed to be more "open-ended" than the previous ones. That does not mean that you need to perform countless hours of simulation or write endless pages. You simply need to explore enough variations of inputs to develop some intuitions about the differences between the even-symmetric and odd-symmetric cases, and then describe in words, supplemented by graphs, what patterns you have observed.)