Neural Computation
Practical 7: Ben-Yishai network (ring model)

Mark van Rossum
2018 version by Peggy Series, Matthias Hennig, and Theoklitos Amvrosiadis

November 14, 2018

1 Aims

- Simulate the responses of a recurrently connected network in V1 to input
- Explore how orientation tuning arises as a function of feedforward input and lateral connections

2 Theoretical Background

Orientation tuning first arises in the primary visual cortex (**V1**)[1]. Hubel and Wiesel hypothesized that orientation selectivity arose due to the structure of the feedforward connections from neurons of the lateral geniculate nucleus (**LGN**) of the thalamus to V1 neurons[2]. However, later studies showed the importance of intracortical inhibition and local connections for the emergence of orientation selectivity[3].

Ben-Yishai et al in 1995 developed a computational model of the V1 network to explore the factors contributing to orientation selectivity.The model included neurons whose activity is modeled with a firing rate, receiving feedforward thalamic input and also local excitatory and inhibitory input. The feedforward input could be broadly or finely tuned, and the local input depends on the similarity of orientation preferences between neurons. The model is described by the following equations:

The feedforward thalamic input for each neuron i :

$$h_i^{ext} = c[1 - \varepsilon + \varepsilon cos(2(\theta_i - \theta_0))]$$

where c represents the contrast of the stimulus (you can think of it as amplitude), ε is a measure of the anisotropy of the stimulus (how finely or broadly tuned the input is), θ_i is the preferred orientation of neuron i, and θ_0 the angle of the stimulus (supposed to be a bar of light).

The interactions between neurons i, j are given by:

$$J_{i,j} = -J_0 + J_2 cos(2(\theta_i - \theta_j))$$

J_0 represents a uniform, orientation-independent present in the cortex, while J_2 represents the maximum amplitude of the interaction between two neurons. The maximum is achieved for neurons with the same orientation preference, whereas the second term becomes negative (inhibitory) as θ_i, θ_j become sufficiently different.

The total recurrent input the neuron receives at one time t is just the integral over all the inputs, scaled by the activity of each input neuron (basically, a weighted average):

$$h_i^{rec}(t) = \int_{-\pi/2}^{\pi/2} \frac{d\theta_j}{\pi} J_{i,j} \upsilon_j(t)$$

Here, υ_j is the firing rate of neuron j, with orientation preference θ_j, which ranges from $-\pi/2$ to $\pi/2$.

[1] In primates and cats, although it is present even earlier in mice and rabbits, see *Scholl et al, 2013* for more information.
[2] see *Hubel and Wiesel, 1962, 1968.*
[3] for example see *Tsumoto et al, 1979.*

The total input a neuron receives is just the sum of the feedforward and the recurrent input.

The output of the neuron is given by a rectifying, semilinear function $g(h)$, where:

- $g(h) = 0$, if $h \leq T$
- $g(h) = \beta(h - T)$, if $T \leq h \leq T + 1/\beta$
- $g(h) = 1$, if $h \geq T + 1/\beta$

where h is the total input to the neuron, T is the threshold, and β is a gain parameter. For very low contrast inputs, the output of the neuron is zero, whereas it saturates for very high values. For intermediate values, the response is linear with a slope given by β.

Finally, the firing rate, υ_i, of each neuron varies with time and its dynamics is given by:

$\tau \frac{d\upsilon_i(t)}{dt} = -\upsilon_i(t) + g(h, t)$

where τ is the time constant that shows how quickly the the firing of the neuron changes.

3 Model Setup

Your task is to create and simulate the behaviour network described above.[4]

- First, create the preferred orientations for a number of neurons (e.g 50), ranging from $-\pi/2$ to $\pi/2$.
- Then, create a vector with the feedforward input, h^{ext}, for each neuron. You can take $c = 2$, $\varepsilon = 0.1$, and $\theta_0 = 0$, to begin. Plot the feedforward input that each neuron receives as a function of its orientation preference. Does the graph make sense? Discuss.
- Create a matrix to hold the interactions between all the neurons. Take $J_0 = 1$ and $J_2 = 5$ as a start.How do you expect this to look like? Visualize the interactions with the function *surf.*
- The recurrent input for a neuron can be calculated as a sum across the corresponding row in the above matrix. However, don't forget the normalization. (The term $\frac{d\theta_j}{\pi}$ becomes $1/no.neurons$ in our case)
- Create the response function of the neuron, that takes its *total* input and outputs the appropriate value. You can take $T = 1$ and $\beta = 0.1$

For the above you can work with scalar values or with vectors/matrices. You can choose either, although the latter is more efficient.

4 Running the Simulation

Now, run the simulation for a certain time. Set the timestep size, the total duration of the simulation, and especially the parameter τ appropriately. (Which one should be larger, timestep or τ?) Store the activities of all the neurons at each time point in a matrix.

- Plot the steady state activity of the neurons. Is this as you would expect?
- Plot the time course of one neuron's activity.
- Now, at some point in the simulation, change the orientation of the stimulus. What do you expect to see?

[4]Note: the parameter values given below are examples and might need further tweaking..

5 Exploring the model

- Does the tuning of the responses depend on the the contrast of the input c? For at least 4 different values of c, plot the activity profile of the population (in the same figure). Discuss your observations.
- Look at the simple Hubel and Wiesel model. This would have no recurrent connectivity and much higher anisotropy ε (but still ≤ 1). What happens to the orientation tuning?
- Check the case where the only intracortical input is uniform inhibition. This is done by having $J_0 > 0$ and $J_2 = 0$. Is the orientation tuning maintained? What about contrast dependence?
- Provide a stimulus with a time-varying orientation and plot the activity of the population. (Bonus: make a movie!) Plot the orientation preference of the neuron that has the highest activity at each time point. What do you observe?
- Explore the effects of the other model parameters on the behaviour of the network. Try to make predictions beforehand and then check your intuitions.

6 (Optional) Adding noise

Add a noise term to each neuron's input. The noise should be random and independent for each neuron.

7 (Optional) Cross-correlations

Look at the cross correlations between the time-varying activity of pairs of neurons. *xcorr* is the relevant function. What do you expect to see? Do so for different parameter sets. Do your findings correspond to what Ben-Yishai et al show?