# CCN Matlab Lab 3: the 'ring model'

Peggy Seriès, pseries@inf.ed.ac.uk

February 17, 2008

### Introduction

The aim of this lab is to re-implement the classical 'ring model', as described by [1]. By doing so, the aim is to learn about firing rate models of neural activity, proposed architectures for cortical columns and attractor dynamics.

## 1 Setting the model up

We consider a model with N=50 neurons. Each neuron is parametrized by a 'preferred' angle $\theta_i$, ranging from $-\pi/2$ to $\pi/2$. When a bar of orientation $\theta_0$ is presented, each neuron receives a visual input from the LGN, $h_i^{ext}(\theta_0)$, which depends on the similarity between $\theta_0$ and $\theta_i$:

$$h_i^{ext}(\theta_0) = c(1 - \epsilon + \epsilon \cos(2(\theta_i - \theta_0)) \tag{1}$$

When $c$ represents the contrast of the stimulus, and $\epsilon$ controls the selectivity of the input.

- Set $\theta_0 = 0$ , $c = 3$, and $\epsilon = 0.1$. Set `h_ext` to be a vector representing the input received by all the neurons in the population. Plot `h_ext.`

The relationship between the input and output of each neuron is described by an semilinear gain (or 'activation') function $g(h)$. The function is defined by:

$$\begin{cases} g(h) = 0 & \text{if} \quad h \leq T \\ g(h) = \beta(h - T) & \text{if} \quad T \leq h \leq T + 1/\beta \\ g(h) = 1 & \text{if} \quad h \geq T + 1/\beta \end{cases} \tag{2}$$

- Write g as a matlab function $y = g(h, \beta, T)$ where $h$ can be a scalar or a vector and $\beta$ and $T$ are parameters.

- Set $T = 1$ and $\beta = 0.1$. Plot g as a function of the input $h$ when $h$ is a scalar varying between 0 and 15.

The model neurons interact through recurrent connections. The weight $J_{ij}$ between neurons $i$ and $j$ can be positive or negative depending on the difference between their preferred orientations $\theta_i$ and $\theta_j$:

$$J_{ij} = -J_0 + J_2 \cos(2(\theta_i - \theta_j)) \tag{3}$$

- Choose $J_0 = 86$ and $J_2 = 112$. Construct a 2D matrix J containing the connections weight $J_{ij}$ between all pairs of neurons in the network. Plot J and check that it has the desired form.

# 2 Running the model

The dynamics of each neuron are governed by the following equations:

$$\tau_0 \frac{dm_i}{dt} = -m_i + g[h_i] \tag{4}$$

where $m_i$ describes the activity of neuron $i$, $\tau_0$ is a time-constant and $h_i$ is the input to neuron $i$.

## 2.1 Hubel and Wiesel Model

The simplest scenario for orientation tuning is based on the Hubel-Wiesel mechanism. In the context of our model, this scenario coresponds to the case where the only synaptic input is coming from the LGN (all $J_{ij} = 0$) and this input is strongly anisotropic ($\epsilon$ is large). So we set:

$$h_i(\theta_0) = h_i^{ext}(\theta_0) \tag{5}$$

and $\epsilon = 0.9$.

- Set $\tau_0 = 5$ msec and $c = 1.5$. Implement these equations and run the model for 30 iterations, for a stimulus oriented at 0 deg.
- run the model again for $c = 1.2$ and $c = 4$ and plot the activity $m$ at t=30. What do you observe?

### 2.2 Recurrent model, marginal phase.

Now we consider the full recurrent model, with $J_{ij}$ specified as in equation 3. The input to each neuron is now defined by:

$$h_i(\theta_0) = \sum_{j=1}^{j=N} J_{ij} m_j + h_i^{ext}(\theta_0) \tag{6}$$

and we assume that the input from LGN is poorly tuned to orientation. We set $\epsilon = 0.1$.

- Implement these equations and run the model for 30 iterations, for a stimulus oriented at 0 deg with $c = 1.2$.
- Plot $m$ at t=30 and show how the responses of the network evolve in time. To do this, save the activity of the network at each time step in a 2D matrix `A` containing the activity of all neurons at each time step. Plot `A`.
- run the model again for $c = 1.5$ and $c = 4$. Comment.
- run the model again for $\epsilon = 0.01$. Comment.

# 3 Experiment 1: changing the stimulus

- Set $c = 8$ and $\epsilon = 0.8$. Run the network for 1000 iterations, while changing the stimulus orientation at the 30th iteration from $\theta_0 = 0$ deg to $\theta_0 = 60$ deg. Comment.

# 4 Experiment 2: removing the stimulus

- Set $c = 1.5$ and $\epsilon = 0.1$. Run the model for 60 iterations, while removing the input $h^{ext}$ after 30 iterations. Comment.

# 5 Experiment 3: adding noise in the input

Now we consider that the input to the model is noisy. We have

$$h_i^{ext}(\theta_0) = c(1 - \epsilon + \epsilon \cos(2(\theta_i - \theta_0)) + \eta_i \tag{7}$$

where $\eta$ is a gaussian random variable, independent for each neuron, with mean 0 and variance equal to the mean $c(1 - \epsilon + \epsilon \cos(2(\theta_i - \theta_0))$.

- Plot $h_{ext}$.
- Run the model for 30 iterations. Comment.

## References

[1] R. Ben-Yishai, R.L. Bar-Or, and H. Sompolinsky. Theory of orientation tuning in visual cortex. 92:3844–3848, 1995.