Plasticity and Pattern Formation in the Visual Cortex with Applications to Stroke Recovery

Drew Kouri, Eric Webb

May 5, 2008

Contents

List of Figures

1 Introduction

The visual cortex transforms the geometry of visual information into a space of concepts. There is not a photographic mapping from the retina to the cortex, but instead a neuronal wiring that leads to the ability to transform raw visual input into geometric patterns. Three key observations, compiled in von der Malsburg's 1973 paper, have been made about how the cortex performs this transformation:

- There are neurons, which are selectively sensitive to the presentation of light bars and edges of a certain orientation
- The neurons seem to be organized in "functional columns", i.e. the neurons lying within one cylinder vertical to the cortical surface are sensitive to the same orientation
- Neighboring columns tend to respond to stimuli of similar orientation

By what mechanisms are the retinal to cortical connections organized in order to produce these patterns? They are unlikely to be caused by genetic pre-determination, as the identification of so many connections would be taxing upon the genetic code. In addition, genetic pre-determination cannot account for the high degree of plasticity seen in experiments. Alternatively, there must be a way for the visual cortex to undergo self-organization as it is presented with various inputs.

The model employed in this paper will describe a method of self-organization. There will be two main planes of cells. The retinal cells take input from the environment and constantly fire when presented with light. The retinal cells will be

connected to a plane of cortical cells. The firing of the retinal cells will excite the excitatory cells (E-cells) of the cortex. The strength of connection from each retinal cell to each cortical cell is initially random. See Figure 1.

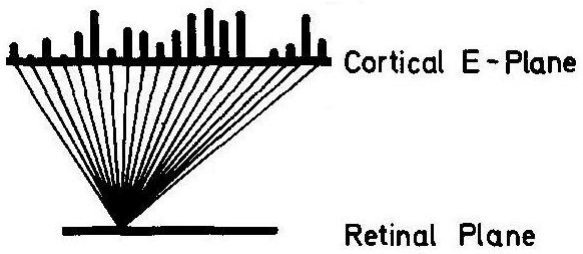

Figure 1: Each retinal cell is connected to each excitatory cortical cell (E-Cell) in this model. The strengths of connection are initially drawn from a flat distribution [12].

In turn, there are also intra-cortical connections. The cortex is composed of both excitatory (E-cells) and inhibitory (I-cells). Upon crossing a threshold, all cortical cells have the ability to fire and affect those cells around them. Active E-cells will excite both the nearby E-cells and the I-cells and move them toward their firing thresholds. Active I-cells will inhibit their next-to-nearest neighbors. The range of inhibitory cells is thus larger than the range of excitatory cells, as this will prevent instabilities due to cycles of excitation and inhibition in the system of cells. See Figure 2 for a visual representation of the intra-cortical wiring.

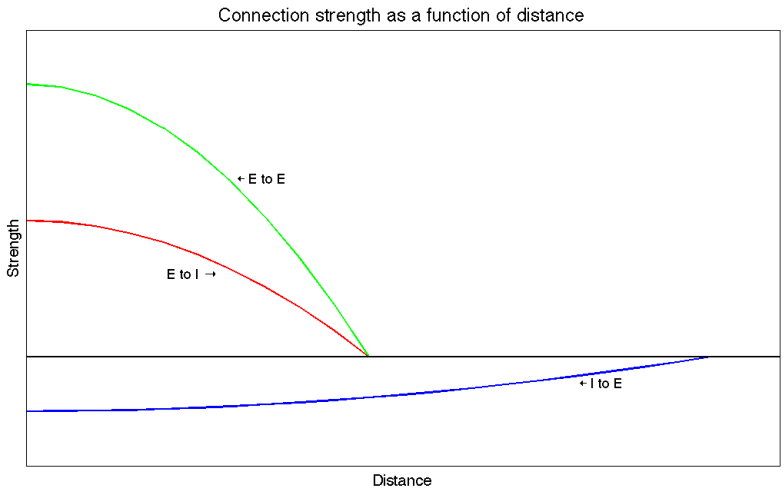

Figure 2: This figure depicts the connection strengths between excitatory cells (E to E), from excitatory cells to inhibitory cells (E to I), and from inhibitory cells to excitatory cells (I to E). The connection from inhibitory cells to inhibitory cells is ignored.

Given the retinal to cortical connections and the intra-cortical connections, there must be a way to update the system so that it can undergo self-organization. This is accomplished through an implementation of Hebbian learning. Cortical E-cells fire due to positive input from other E-cells and from retinal cells. As intra-cortical connection strengths are a function of distance (Figure 2), the learning must happen in the wiring between the retina and the cortex. If a cortical E-cell fires, the strength of connection between it and all active retinal cells is increased. Over time, this will lead to more likely firing given the same retinal input. In order to keep overall connection strength constant, other connections to the E-cell are weakened at the expense of strengthening the connection between active retinal cells and the E-cell.

2 The Model

The model that we implemented was presented by von der Malsburg in [19] and is similar to models investigated in [8, 4, 16, 17]. The model is presented as a hexagonal array of cells, which allows for a nearly circular neighborhood for each cell. The cortical hexagonal array is set up with 15 cells on the principal diagonal and thus is an array of 169 cells. The model is further split into two planes: the excitatory plane (E-plane) and the inhibitory plane (I-plane). Each cell is thus two cells superimposed on each other (excitatory E-cell and inhibitory I-cell). Therefore, the differential system consists of 2×169 equations. The cortical plane of 338 cells is attached to the retinal plane which consists of 19 cells. The connection strength between the cortical and retinal plane are initially drawn from the uniform distribution on the interval $[0, s]$. The connection strengths between E-cells and I-cells, I-cells and E-

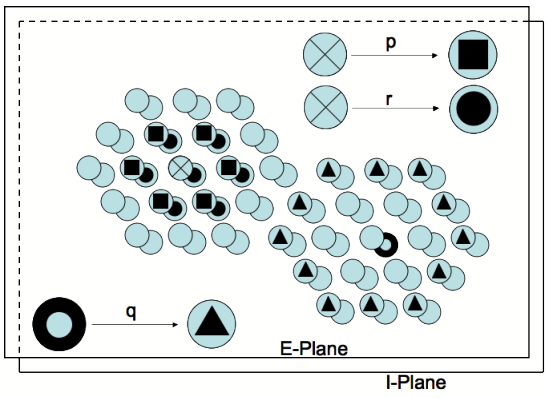

Figure 3: This is a schematic of the planes containing the excitatory and inhibitory cells. The connectivity parameters are set according to the arrows described by connectivity strengths p, q, and r.

cells, and two E-cells are set up according to Figures 2 and 3. One E-cell only affects its immediately neighboring E-cells. Similarly, one E-cell affects its corresponding

I-cell and the immediately neighboring I-cells. Finally, one I-cell affects the E-cells that are two cells away from the I-cell.

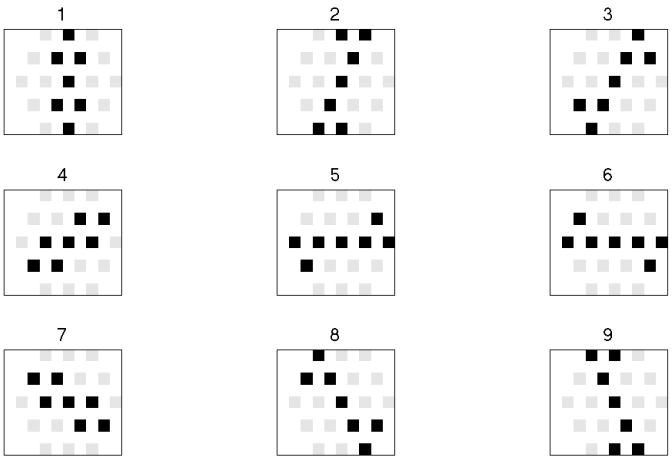

Figure 4: The set of nine retinal stimuli.

There is a set of nine standard stimuli used to excite the cortical cells, see Figure 4. These stimuli represent light bars presented with different orientations. The stimuli consist of 7 excited cells (signal strength of 1) and 12 non-excited cells (signal strength of 0) and can be viewed as rotations of the first stimulus by an angle of $\frac{\pi(k-5.5)}{9}$ for $k = 1, 2, 3, 4, 5, 6, 7, 8, 9$. Moreover, the geometric arrangement of these stimuli is not the important aspect, but rather the relationships created via the mutual overlapping of the light bars.

2.1 Time Evolutionary Model

Let $H_k(t)$ be the excitatory state of cortical cell k at time t. If $H_k(t)$ crosses the threshold of 1, then the cell fires. In this model, the output signal of the cell, $H_k^*(t)$,

increases linearly with the excitatory state. So

$$H_k^*(t) = \begin{cases} H_k(t) - 1, & \text{if } H_k^*(t) > 1, \\ 0, & \text{otherwise.} \end{cases} \tag{2.1.1}$$

Let $A_i^*(t)$ be the signal of afferent fibre i at time t. This value is identically 1 if retinal cell i is active and identically 0 if it is not. The dynamics of $H_k(t)$ can be found using

$$\frac{d}{dt}H_k(t) = -\alpha_k H_k(t) + \sum_{l=1}^{169} p_{lk} H_l^*(t) + \sum_{i=1}^{19} s_{ik} A_i^*(t), k = 1, ..., 169, \tag{2.1.2}$$

where α_k is the decay constant of the excitatory state of cortical cell k, p_{lk} is the strength of connection from cell l to cell k, and s_{ik} is the strength of connection between fibre i and cell k (see Figure 3). These dynamics lead to an initial transient response eventually succumbing to a steady state solution ($\frac{dH_k}{dt} = 0$).

Physiologically, equation (2.1.2) has a very straight forward meaning. In this equation, the first sum, $\sum_{l=1}^{169} p_{lk} H_l^*(t)$ represents the neighboring cells contributions to the excitatory state of cell k; while, the second sum, $\sum_{i=1}^{19} s_{ik} A_i^*(t)$, represents the contributions to the excitatory state of cell k from afferent fibers (the nine standard stimuli). In general, $A_i^*(t)$ could vary with time, but, as will be seen, we are not concerned with these dynamics. We are only concerned with how the cells act during steady state. Thus, we fix $A_i^*(t)$ and compute solutions until steady states are attained.

2.2 Steady State Solutions

In the steady state solution, it is possible to analyze the behavior of the cortical cells in response to various stimuli. In order to accomplish this, a more specialized set of equations is needed. The specialization of Equation 2.1.2 to $\frac{dH_k}{dt} = 0$ and the absorbtion of the α_k term leads to

$$H_k = \sum_{l=1}^{169} p'_{lk} H_l^* + \sum_{i=1}^{19} s'_{ik} A_i^*. \tag{2.2.1}$$

We can split H_k into excitatory and inhibitory signals. Let E_k be the excitatory state of E-cell number k and I_k be the excitatory state of I-cell number k. Similarly, let E_k^* and I_k^* be the signals of the E-cell k and I-cell k if they cross the threshold. Now it is possible to elucidate the dynamics of E_k and I_k:

$$E_k = \sum_{i=1}^{169} p_{lk} E_l^* - \sum_{i=1}^{169} q_{lk} I_l^* + \sum_{i=1}^{19} s'_{ik} A_i^* \tag{2.2.2}$$

$$I_k = \sum_{i=1}^{169} r_{lk} E_l^* \tag{2.2.3}$$

where $p_{lk} > 0$ is the connection strength from E-cell l to E-cell k, $q_{lk} > 0$ is the connection strength from I-cell l to E-cell k, $r_{lk} > 0$ is the connection strength from E-cell l to I-cell k, and $s'_{ik} > 0$ is the connection strength from afferent fibre i to E-cell k. Recall that s'_{ik} is the value being updated by Hebbian learning.

Again, these equations have a very straight forward physiological meaning. The sum, $\sum_{i=1}^{169} p_{lk} E_l^*$, represents the contribution to the k^{th} E-cell's excitatory state by

the neighboring E-cells, the sum, $\sum_{i=1}^{169} q_{lk} I_l^*$, represents the contribution by the neighboring I-cells, and the sum, $\sum_{i=1}^{19} s'_{ik} A_i^*$, represents the contribution from the afferent fibers. Similarly, the sum, $\sum_{i=1}^{169} r_{lk} E_l^*$, represents the contribution to the k^{th} I-cells excitatory state by the neighboring E-cells. By looking at the values of E_k given different inputs, it is possible to see what stimuli cause cell k to fire. In this way, the observations given in the introduction can be found.

3 Analysis and Results

Equation (2.1.2) gives a general equation for the time evolutionary dynamics of our neural network. In practice, we solve an adaptation of this equation. The adaptation takes into account both the inhibitory and excitatory cells and can be formulated as follows:

$$\frac{dE_k}{dt} = \alpha_k(-E_k + \sum_{i=1}^{169} p_{lk} E_l^* - \sum_{i=1}^{169} q_{lk} I_l^* + \sum_{i=1}^{19} s'_{ik} A_i^*) \tag{3.0.4}$$

$$\frac{dI_k}{dt} = \beta_k(-I_k + \sum_{i=1}^{169} r_{lk} E_l^*). \tag{3.0.5}$$

It is worth noting that by changing the α_k and β_k parameters, one compresses or lengthens the time it takes to reach steady state, but differing these values does not change the values of the equations at steady state. Thus, to make solving the differential system more time efficient, one should make these parameters as large as possible. This will make the equations reach steady state quicker. Some care needs to be taken in choosing these parameters because if α_k and β_k are too large, the system will blow up and not attain a steady state.

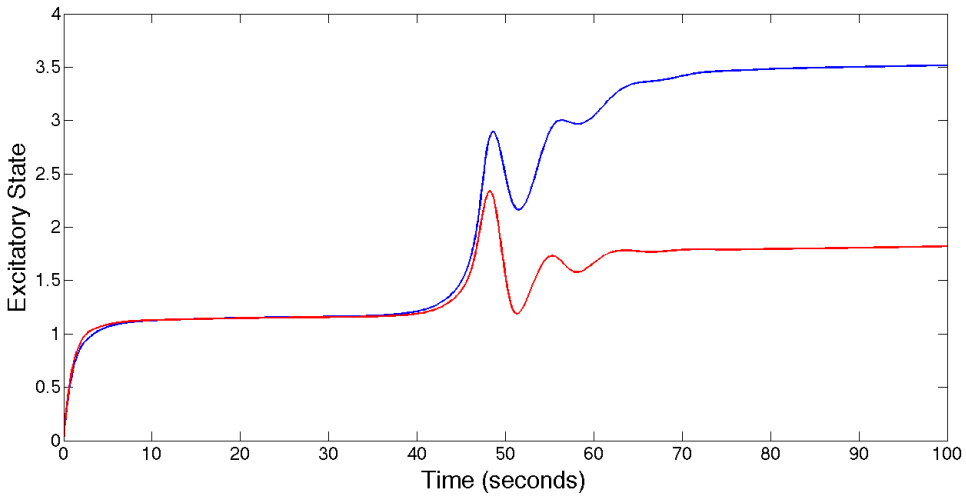

Figure 5: This figure depicts the typical solution of the excitatory state of two cells. This image clear depicts the three phases: logarithmic growth with first plateau, oscillations, and final plateau.

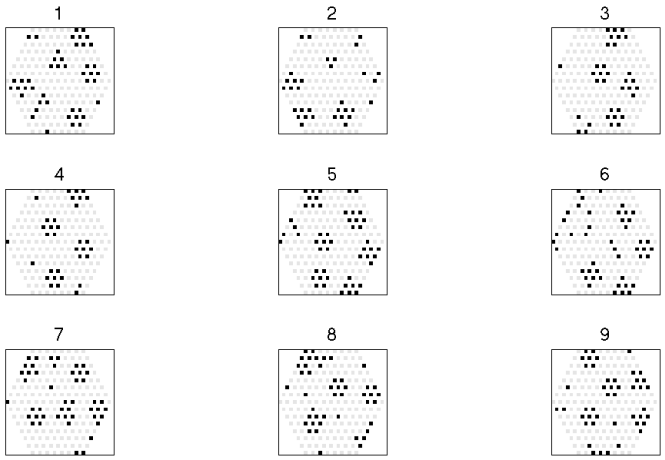

Figure 6: Steady state solutions without learning for each of the nine stimuli in the previous figure.

We solved the above time evolutionary equations (3.0.4) and (3.0.5) in MATLAB using a forward Euler's method (as per von der Malsburg's suggestion). We iterated the 2×169 differential equations until we attained possible steady states. In [19], they describe the above system and state that they iterated the system of differential

equations until the mean change in solutions between two steps was less then 0.5%. Upon investigation of the solutions to this differential system, we notice that the solutions begin to grow logarithmically, then plateau. This plateau lasts for a while (about $t = 30$ milliseconds for some cells), then the solutions begin to oscillate. After oscillation, the solutions finally reach steady state which occurs at about $t = 70$ milliseconds. Once we have reached the post oscillation steady state, we notice that the hexagonal array has already formed crude patterns (see Figure 6) .

With these "steady state" solutions, we then subjected the afferent connectivity strength values to Hebbian learning. This learning phase consisted of the following algorithm:

1. Choose a stimuli from the set of nine standard stimuli in Figure 4;
2. Determine the steady state solution to the network of differential equations;
3. Update the afferent fiber connection strengths via Hebb's rule;
4. Renormalize the afferent fiber connection strengths.

One stage of learning consists of presenting all nine stimuli in the following order: 1,6,2,7,3,8,4,9,5. Von der Malsburg choose this scheme in order to reduce possible special effects due to maximally overlapping stimuli. After each presentation, the synaptic strengths are updated using Hebb's rule with respect to some learning constant h:

$$s'_{ik} = s_{ik} + hA^*_i E^*_k. \tag{3.0.6}$$

This formula updates s'_{ik} by increasing connectivity strength between retinal cell i and excitatory cell k if the stimuli caused the cell to excite. Following the connection

strength update, s'_{ik} is then normalized so that the total connectivity for each cell is held constant at $19s/2$. This normalization is as follows:

$$s_{ik} = \frac{s'_{ik}}{\sum_{i=1}^{19} s'_{ik}} \frac{19s}{2}. \tag{3.0.7}$$

The learning phase as described represents the reorganization of the afferent fibers between the retinal and cortical planes. This reorganization plays a large role in stroke recovery [17, 12]. Strokes can compromise the transport of blood to neurons in the brain, which may lead to the damage and death of cells. This damage could impair the functionality of the stroke victim. Plasticity in the cortical cells surrounding the damaged cells plays a central role in recovering from strokes. This plasticity is described by Hebbian learning and the learning phases in which we subject our model.

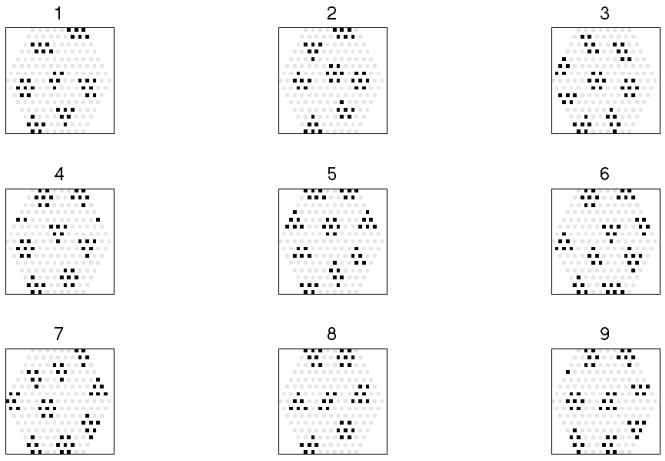

Figure 7: Solutions for the nine stimuli after 20 learning phases. Clearly, the cells are beginning for fire in patterns.

The results of the Hebbian learning are depicted in Figures 7 and 8. Patterns begin to form after the first stage of learning and become increasingly defined as

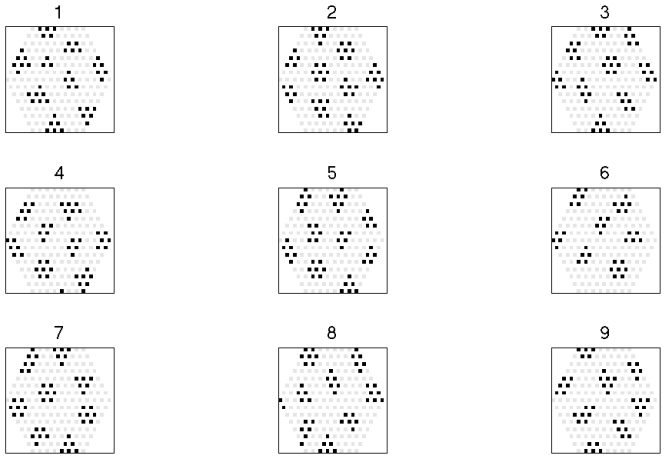

Figure 8: Solutions for the nine stimuli after 100 learning phases. The patterns are now well defined.

we perform more and more learning phases. These patterns are due to the increased afferent connectivity strengths updated via Hebb's rule. One tool that may be utilized to analyze the pattern formation is the tuning curve for each cell. The tuning curve at a given learning step is the excited state of the E-cell with respect to all nine initial stimuli. If the tuning curve at $k(= 1, 2, 3, 4, 5, 6, 7, 8, 9)$ is above the threshold (set identically to 1), then the k^{th} stimuli causes the cell to excite. Since the nine stimuli are just rotations of the of each other by $\frac{\pi(k-5.5)}{9}$ radians, we can determine the most likely orientation of the cell given by the angle of the stimuli that causes the greatest excitatory state. As von der Malsburg has shown, the tuning curves for each cell become unimodal (if the cell is excited) as learning progresses because the cell becomes sensitive to similar light orientations. This activity signifies that the afferent strengths are reorganizing and learning.

Due to vague descriptions in the literature about the implementation of the this model, we were unable to obtain similar tuning curve results. Although we do form

patterns as seen in Figures 7 and 8, the tuning curves do not become unimodal. We hypothesize that this is due, in part, to how we solve our differential equations, the choice in parameters, and the equations actually implemented (be it the time evolutionary equation presented in the Subsection "Time Evolutionary Model" or the equations presented in the "Analysis and Results" section). In fact, we have been in contact with Professor Christoph von der Malsburg (see Appendix), who was unable to recall the details of his implementation.

4 Discussion

Neural network models have many applications in neuroscience as well as in computer science and other computational science. In fact, pattern forming neural networks are currently used to analyze high dimensional data using a method called Self Organizing Maps. In our study, we focused on the application of neural networks to stroke recovery. With an implemented neural network of the cortical receptive field, similar to the one we investigated, one can subject the afferent connections to stroke-like ablations. That is, we can severe some of the afferent connections (set the connectivity strength identically to zero), then investigate how the surrounding cells reorganize to compensate. There have been many investigations of this sort (see [5, 17]). In fact, Einarsdottir et al. showed that one possible influence to cortical plasticity and stroke recovery is the level of γ-aminobutyric (GABA-ergic) inhibition. Also, Sober et al. utilize a similar model to hypothesis about possible stroke recovery phases. Stroke recovery is similar to the development of an immature nervous system in

which the synaptic (afferent) strengths increase or decrease with respect to Hebbian learning. Sober et al. propose that stroke recovery has two phases: a dynamic phase and a plasticity phase. The dynamic phase consists of dynamic changes in neural activity, while the plasticity phase consists of updating synaptic strengths via a Hebbian learning rule.

5 Conclusions

The study of neural networks is ever increasing. With increased computational power, larger and larger networks can be implemented and studied. Also, there has been a lot of research performed in the area of pattern formation. As we have seen, this research has applications in stroke recovery and, in fact, can pinpoint important parameters that could effect the success of recovery. Being able to determine the important parameters in this dynamical system has the potential of massive impact in health care and rehabilitation. For this and many other reasons, further understanding of these neural networks and their dynamics has the possibility of greatly improving stroke recovery.

6 Author Contributions

Drew Kouri implemented the model and graphical displaying of the model. He also wrote the Analysis and Results, Discussion, and Conclusions sections. Eric Webb wrote code to generate possible tuning curve representations, but those were not

included in our paper because we did not see similar patterns as von der Malsburg. He also wrote the Introduction and Model sections. Kouri and Webb worked together in editing the paper and creating the presentation.

7 Appendix

7.1 Email from Christoph von der Malsburg

From View message header detail Christoph von der Malsburg <malsburg@fias.uni-frankfurt.de>

Sent Sunday, May 4, 2008 3:51 pm

To Drew Kouri <drew.kouri@case.edu>

Cc

Bcc

Subject Re: Questions on 1973 Self-Organization paper

Drew Kouri wrote:

> I am studying neuroscience at Case Western Reserve University in
> Cleveland, OH. Currently, I am trying to implement the model you
> present in your 1973 paper "Self-Organization of Orientation Sensitive
> Cells in the Striate Cortex." I have some questions involving your
> implementation. First of all, you say that you stop your differential
> equation solver after 20 iterative (time) steps (page 91), what is the
> size of the time steps that you use? Also, what are the initial
> conditions and the values for the decay constant that you use?
>

Hi,

This was 35 years ago and I don't remember the details, as you can imagine. But here are some remarks that may help you get the thing going. What time steps mean in real time (milliseconds in neurophysiological terms) is your phantasy -- one would have to know the exact time constants of the cells involved, and these are not so well known, especially at the early ontogenetic time where this model is to be applied. But that is probably not your concern anyway. I have used the Euler method, which I'm sure I have described. There is a coefficient that scales the change in neural excitation (and another one that scales synaptic changes). If you make that too big, the system

gets unstable and all variables blow up. On the other hand, in the interest of speed you want to set these coefficients as big as possible. What you should do is repeat a calculation with half the coefficients and see whether you get approximately the same results (after twice the number of steps, of course). If that is approximately the case you are on the safe side. The first problem I had to battle with at the time was that the system went into oscillations, and I first had to learn how to set parameters such that these would at least be dampened.

As for initial values, the synaptic strengths were random numbers (normalized to one, as described in the paper). The initial values of cellular excitation I don't remember, but they may just have been zero.

What bothered me about the model for a long time was that it could work only after eye opening, whereas the orientation columns are present already at eye opening. I finally have come around to publish a model that can work in the dark. See http://www.jneurosci.org/cgi/content/full/28/1/249. That model is not much more complicated than my 1973 one, and you should at least look into it.

Have fun,

Christoph von der Malsburg

7.2 von der Malsburg Differential System Code

```
function [H,i] = eulers(tspan,n,icond,N,A,s,p,q,r)

% Determine Time Step
dt = tspan/n;

% Define Differential System
alpha = 2;
```

```
ys = @(y) (y-1).*(y>1);

F = @(x,y,k) alpha*(-x(k)+p(:,k)'*ys(x)-q(:,k)'*ys(y)+s(:,k)'*A);

G = @(x,y,k) alpha*(-y(k)+r(:,k)'*ys(x));


% Set Initial Conditions

x(:,1) = icond(1:N);

y(:,1) = icond(N+1:2*N);


% Begin Forward Euler's

i = 2;

m = 1;

while((m>0.005)+(i<n)==2)

    for j = 1:N

        x(j,i) = x(j,i-1)+dt*F(x(:,i-1),y(:,i-1),j);

        y(j,i) = y(j,i-1)+dt*G(x(:,i-1),y(:,i-1),j);

    end

    % Compute mean change

    m = mean([abs(x(:,i)-x(:,i-1));abs(y(:,i)-y(:,i-1))]);

    i = i+1;

end

H = [x;y];
```

7.3 Solver and Hexagonal Array Code

```
% ========================================================================

% Stationary von der Malsburg Model of Changes in Receptive Fields

% Drew Kouri

% ========================================================================

% Here is a quick implementation of the stationary model.

% For some reason, I am getting negative populations, then once the

% connection strength reaches equilibrium, there is a periodic solutions.

% This is probably not correct, but I cannot find any error in the code.

% Have a look at it.

% ========================================================================
```

```
for k = 1:9


N = 169; % Number of Cortical Cells

M = 19;  % Number of Retinal Fibres


% Cell Signal Function

star = @(y,th) (y-th).*(y>th);


% Signal of Afferent Fibres

%      01  02  03

%    04  05  06  07

%  08  09  10  11  12

%    13  14  15  16

%      17  18  19

A = zeros(M,1);

state(:,1) = [2;5;6;10;14;15;18];

state(:,2) = [2;3;6;10;14;17;18];

state(:,3) = [3;6;7;10;13;14;17];

state(:,4) = [6;7;9;10;11;13;14];

state(:,5) = [7;8;9;10;11;12;13];

state(:,6) = [4;8;9;10;11;12;16];

state(:,7) = [4;5;9;10;11;15;16];

state(:,8) = [1;4;5;10;15;16;19];

state(:,9) = [1;2;5;10;15;18;19];


% Beginning Stimuli

% k = 1;

A(state(:,k)) = ones(7,1);


% Learning Presentation Sequence

list = [1;6;2;7;3;8;4;9;5];


% Strength of Connections

load paramStat.mat
```

```
% Learning Parameter
h = 0.05;

% Find steady state solution
% Define Initial Conditions
icond = zeros(2*N,1);

% Solve ODS (Explicit Runge-Kutta (4,5) formula)
Y = eulers(50,200,icond,N,A,s,p,q,r);

% Initialize Excitatory Populations
E = zeros(N,101);
E(:,1) = Y(1:N,length(Y(1,:)));

% Begin Learning
for j = 2:101
    b = 0;
    if(j==61), h=0.1; end % Increasing Learning Parameter after 60
    tic
    for i = 1:9
        % Stimuli Presentation
        A = zeros(M,1);
        A(state(:,list(i))) = ones(7,1);

        % Solve To Steady State and Update Excitatory Cells
        [Y,z] = eulers(50,200,Y(:,length(Y(1,:))),N,A,s,p,q,r);
        b = b+z;
        E(:,j) = Y(1:N,length(Y(1,:)));

        % Hebbian Learning and Normalization
        s = s+h*A*star(E(:,j),1)';
        s = (M*0.25/2)*(s./repmat(sum(s),M,1));
    end
    disp(strcat('Learning Phase #',num2str(j-1)))
```

```
    disp(strcat('# Euler Steps =',num2str(b-9)))
    toc
end

% Define Postions in plot window
pos = [0,8:15,14:-1:8];
cpos = cumsum(pos);
R = [6;6;4;4;2;2;0;0;0;2;2;4;4;6;6];
row = @(i) (mod(i,2)==1)*(2*(1:pos(i))+R(i-1)-1)+(mod(i,2)==0)*(2*(1:pos(i))+R(i-1));

% Create Image Array
im = zeros(2*15-1,2*15,101);
Ys = (E>1)+1;
for j = 1:101
    for i =2:16
        im(2*(i-1)-1,row(i),j) = Ys(cpos(i-1)+1:cpos(i),j);
    end
end

name = strcat('run',num2str(k));
save(name,'E','im');
clear all
end

% % Record .avi of Receptive Field Time Series
% fig=figure;
% imagesc(im(:,:,1)-1)
% set(fig,'DoubleBuffer','on');
% set(gca,'xlim',[0 30],'ylim',[0 30],...
%     'NextPlot','replacechildren')
% colormap([1 1 1;.9 .9 .9])
% % mov = avifile('plasticity2.avi');
% for i = 1:101
%     if((sum(sum((im(:,:,i)-1)>0))==0)==1)
```

```
%         colormap([1 1 1;0.9 0.9 0.9])
%     else
%         colormap([1 1 1;0.9 0.9 0.9;0 0 0])
%     end
%     imagesc(im(:,:,i)-1)
%     title(strcat('Learning Phase #',num2str(i-1)))
% %     F = getframe(gca);
% %     mov = addframe(mov,F);
%     pause(.5)
% end
% % mov = close(mov);
```

References

[1] Amari S. *Dynamics of Pattern Formation in Lateral-Inhibition Type Neural Fields*. Biological Cybernetics. Vol 27 (1977).

[2] Buonomano DV, Merzenich MM. *Cortical Plasticity: From Synapses to Maps* Annual Reviews of Neuroscience. Vol 21 (1998).

[3] Carpenter GA, Grossberg S. *The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Networks*. Computer. Vol 21, No 3 (Mar 1988).

[4] Cho S, Reggia JA. *A Computational Model of Proprioceptive Maps*. Proceedings of 1993 International Joint Conference on Neural Networks.

[5] Einarsdottir H, Montani F, Schultz SR. *A Mathematical Model of Receptive Field Reorganization Following Stroke*. Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference. (July 2007).

[6] Grabska-Barwinska A, von der Malsburg C. *Establishment of a Scaffold for Orientation Maps in Primary Visual Cortex of Higher Mammals*. The Journal of Neuroscience. Vol 2 (Jan 2, 2008).

[7] Grossberg S, Seitz A. *Laminar Development of Receptive Fields, Maps and Columns in Visual Cortex: The Coordinating Role of the Subplate*. Cerebral Cortex. Vol 13 (Aug 2003).

[8] Grossberg S. *Neural Expectation: Cerebellar and Retinal Analogs of Cells Fired by Learnable or Unlearned Pattern Classes*. Kybernetik. Vol 1 (1972).

[9] Grossberg S. *Neural Pattern Discrimination*. Journal of Theoretical Biology. Vol 27 (1970).

[10] Linsker R. *Perceptual Neural Organization: Some Approaches Based on Network Models and Information Theory*. Annual Review of Neuroscience. Vol 13 (1990).

[11] Linsker R. *Self-Organization in a Perceptual Network*. Computer. Vol 21, No 3 (Mar 1988).

[12] Lytton WW, Stark JM, Yamasaki DS, Sober SJ. *Computer Models of Stroke Recovery: Implications for Neurorehabilitation*. The Neuroscientist. Vol 5 (1999).

[13] MacKay DJC, Miller KD. *Analysis of Linsker's Simulations of Hebbian Rules*. Neural Computation. Vol 2 (1990).

[14] Miikulainen R, Bednar JA, Choe Y, Sirosh J. *Self-Organization, Plasticity, and Low-level Visual Phenomena in a Laterally Connected Map Model of the Primary Visual Cortex*. Psychology of Learning and Motivation. Vol 36 (1997).

[15] Obermayer K, Blasdel GG, Schulten K. *Statistical-Mechanical Analysis of Self-Organization and Pattern Formation during the Development of Visual Maps*. The American Physical Society: Physical Review A. Vol 45, No 10 (May 15, 1992).

[16] Reggia JA, D'Autrechy CL, Sutton GG, Weinrich M. *A Competitive Distribution Theory of Neocortical Dynamics*. Neural Computation. Vol 4 (1992).

[17] Sober SJ, Stark JM, Yamasaki DS, Lytton WW. *Receptive Field Changes After Strokelike Cortical Ablation: A Role for Activation Dynamics*. Journal of Neurophysiology. Vol 78 (1997).

[18] von der Malsburg C. "Self-Organization in the Brain". *The Handbook of Brain Theory and Neural Networks 2nd Edition*. Cambridge, MA: The MIT Press (2002).

[19] von der Malsburg C. *Self-Organization of Orientation Sensitive Cells in the Striate Cortex*. Kybernetik. Vol 14 (1973).

[20] von der Malsburg C. *Self-Organization: The Unfinished Revolution*. Idea-Finding Symposium Frankfurt Institute for Advanced Studies. (2003).

[21] Willshaw DJ, von der Malsburg C. *How Patterned Neural Connections can be Set up by Self-Organization*. Proceedings of the Royal Society of London, Series B, Biological Sciences. Vol 194 (1976).