CSE2330   Introduction to Computational Neuroscience

Tutorial/Assignment 2: Model of Limulus Vision

The Horseshoe Crab
Limulus polyphemus

La Limule

2.1   About this tutorial

The objective of this tutorial is to introduce:

- computational building blocks used in modelling the input stage of a simple vision system.

- study a model of the input stage of a simple vision system and its operation.

The vision system is based on that of a famous **limulus** that has been extensively studied due to its simplicity. Read about it in:
`http://www.mbl.edu/animals/Limulus`, and in the 1967 Nobel Prize presentation speech at the end of this prac manual.

Contents

2.2   getting started ...

- Invoke MATLAB and in the desktop window that will be open identify two important panes: Command window and Current Directory.

- Navigate in the Current Directory pane to your favourite working prac directory (folder). I will refer to it as $YFD . Create there the directory prac2 and navigate to it. Type in MATLAB command window pwd to confirm that you are in $YFD/prac2.

- From the pull down menu select New M-file. This will open an Untitled editor window. Save As say, myprac2.m or under any name you like. That will be the name of your MATLAB script file that you will build up in the course of this prac.

  It is easier to type in a command in the editor window, try type in a = 5, and then execute this statement selecting from the pull down menu

  Debug $->$ Save and Run

  or just pressing the F5 function key.

  If you would like to execute only part of your script, insert command return to break program at that point. There are more sophisticated tools to do so, but we do not need them at this point.

- Remember to use a semicolon ; at the end of a statement to suppress printing. Imagine that a 100×100 matrix consists of 10 000 numbers that will be printed if ; is not present at the end of the statement.

2.3   A simple test image

A simple image in MATLAB is an $r \times c$ matrix, each element of the matrix (known as a **pixel**) representing intensity of the pixel. Mathematically such an image can be also represented as a surface in three-dimensional space, the pixel value gives the elevation of the surface.

Our first test image will be a cylinder of the unity hight, so that its image will be just a circular disk. Let us start with the following MATLAB commands that you should type in in the editor window called myprac2.m or something similar:

```
%  myprac2.m
%           29 July 2005
 clear,  close all
 m = 80 ;
 x = 1:m ;
 [X Y] = meshgrid(x,x) ;
 % 2-D cylinder:
 Z = ((X - m/2).^2 + (Y - m/2).^2) < (0.125*m^2) ;
 islogical(Z)   % Z is logical
 figure(1)
 imagesc(Z), axis image
 figure(2)
 surf(double(Z))  % conversion to double and visaalisation
```

This should produce the two following plots showing the top view and the perspective view of the unity high cylinder. We will use this image to test the way the vision system model works.
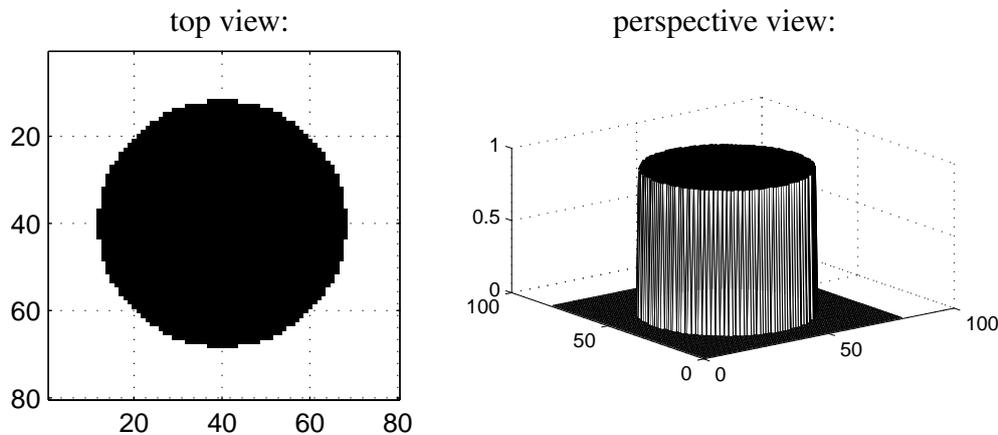
Figure 1: A simple test image

2.4   One-dimensional model of the limulus vision

The central aspect of a simple neural model of the limulus vision is based on the concept of **lateral inhibition**, which is instrumental in image enhancement through sharpening edges in the image. Having said "simple" model, I am fully aware that we have some background knowledge to work out first.

2.4.1   A rectangular signal

To start with let us first extract from the image just one horizontal line, say line 13, and see the profile of intensity along that line. This can be done using the following code that you should append to you script in the editor window and execute:

```
% the intensity profile along the line 13
 u = Z(13,27:56) ;  % row 13, columns 27 to 56
 figure(1)
 plot(u), grid on,
 title('intensity profile along the line 13')
 m = max(size(u))
```

As a result you should see the plots of the intensity profile as in Figure 2. The second plot has been obtained using stem(u) instead of the plot(u) command. Note that for clarity we have extracted only 30 points from the original image line numbered originally from 27 to 56.

intensity profile along the line 13
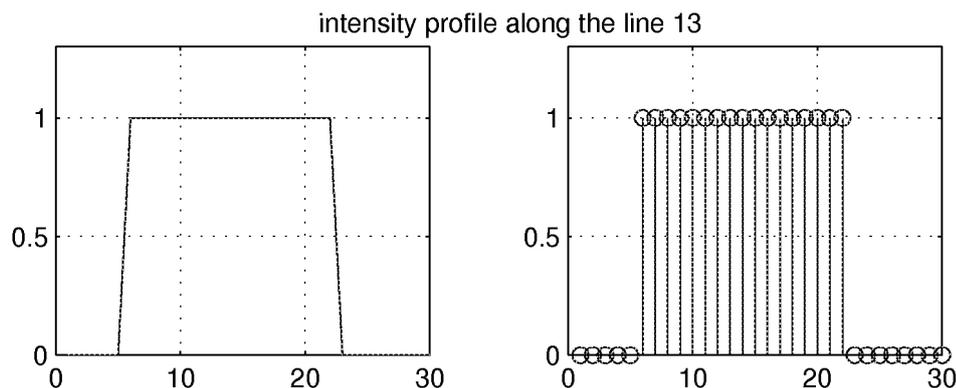
Figure 2: The intensity profile along the horizontal line 13

2.4.2   1-D feedforward model of lateral inhibition

We can imagine that the vector u contains all $m = 30$ **intensity signals** coming from the **visual receptors** to synapses of 30 neurons.

The essence of the operation performs by the limulus on the signals from receptors is the **sharpening the edges** present in the perceived signals.

In order to achieve such an effect the signal u_k from the k-th receptor R is connected to an **excitatory synapse** of the k-th neuron and at the same time ic connected to the **inhibitory synapses** of the neighbouring neurons, $k - 1$ and $k + 1$. Such a sidewise connection is known as the **lateral inhibition**. A one-dimensional feed-forward model of the limulus vision based on lateral inhibition is illustrated in Figure 3.
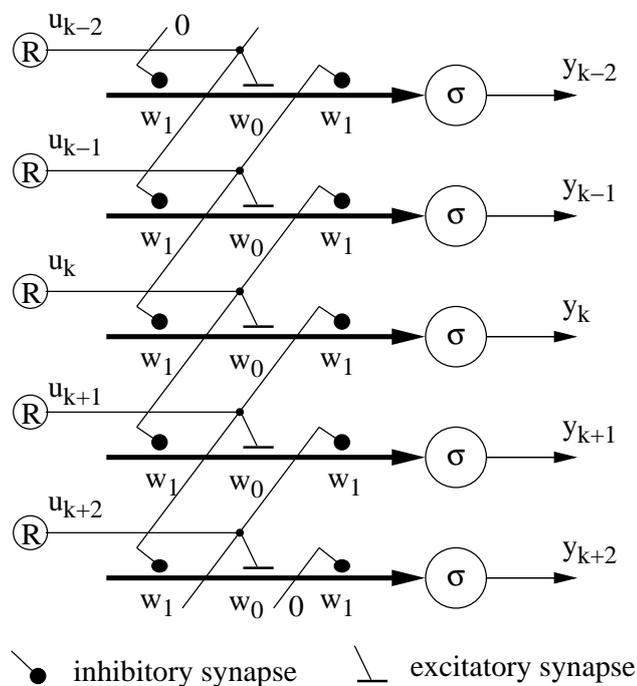
Figure 3: One-dimensional feed-forward model of the limulus vision

In the example you can identify five neurons, the central one with the efferent (output) signal y_k. Along the dendrite of each neuron we have three synapses, the central excitatory synapse with the weight $w_0 > 0$ accepts the afferent signal u_k from the receptor. This signal goes also to two neighbouring neurons inhibiting them. The weights of the inhibiting synapses are negative, $w_1 < 0$ Note that all neurons have synapses with identical weights, $[w_1, \ w_0, \ w_1]$

2.4.3   Maths of lateral inhibition. A Mexican hat mask

The diagram of a 1-D limulus vision system can now be converted into equations and subsequently into a modelling program. First note that, assuming that $\sigma = 1$, a single efferent signal y_k is calculated as a weighted combination of all three synaptic signals, u_{k-1}, u_k, y_{k+1}:

$$y_k = w_1 \cdot u_{k-1} + w_0 \cdot u_k + w_1 \cdot u_{k+1} = [w_1 \ w_0 \ w_1] \cdot \begin{bmatrix} u_{k-1} \\ u_k \\ u_{k+1} \end{bmatrix} \tag{1}$$

In the above equation we recognize the familiar inner product of two vectors, namely, the weight vector $[w_{i-1} \ w_i \ w_{i+1}]$ and the afferent signal vector $[u_{k-1} \ u_k \ u_{k+1}]$.

We can use a matrix notation to calculate all efferent signal in one step. All, meaning 30, might be too much, but we can try three to get the following equation

$$\begin{bmatrix} y_{k-1} \\ y_k \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} w_1 & w_0 & w_1 & 0 & 0 \\ 0 & w_1 & w_0 & w_1 & 0 \\ 0 & 0 & w_1 & w_0 & w_1 \end{bmatrix} \cdot \begin{bmatrix} u_{k-2} \\ u_{k-1} \\ u_k \\ u_{k+1} \\ u_{k+2} \end{bmatrix} \tag{2}$$

Eqn (2) looks horrible, I know, but it still says that every efferent signal, say y_k, is formed as a inner product of the respective row weight vector and the signal vector.

Exercise 2. 1
From eqn (2) write down expressions for y_{k-1} and y_{k+1} similar to that of eqn (1).
□   2 marks

To simplify eqn (1) we introduce the important concept of a **mask, h** that describes the pattern of **lateral exhibition** and excitation:

$$\mathbf{h} = \begin{bmatrix} w_1 & w_0 & w_1 \end{bmatrix}$$

Now we can re-write eqn (2) into a more concise form:

$$\begin{bmatrix} y_{k-1} \\ y_k \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{h} & 0 & 0 \\ 0 & \mathbf{h} & 0 \\ 0 & 0 & \mathbf{h} \end{bmatrix} \cdot \begin{bmatrix} u_{k-2} \\ u_{k-1} \\ u_k \\ u_{k+1} \\ u_{k+2} \end{bmatrix} \tag{3}$$

As you can see, we need five (1+3+1 = 5) signals u on the right hand side of the equation to obtain three signals on the left hand side of the equation.

It is possible to write the above equation in a user-friendly matrix-vector form, which, after small peripheral adjustments, can be expressed in the following way:

$$\mathbf{y} = W \cdot \mathbf{u} \tag{4}$$

where $\mathbf{y}, \mathbf{u}$ are vectors of the efferent signals and the afferent signals from the visual receptors, respectively, and W is the weight matrix built from the mask vector $\mathbf{h}$.

In its simplest form the **mask** might look as follows

$$\mathbf{h} = \left[\begin{array}{ccc} -1 & +3 & -1 \end{array} \right]$$

The specific values have been selected having in mind the lateral symmetry and the **balance of inhibitory and excitatory behaviour**.
The equivalent matrix W for this mask has the following form (eqn (3)):

$$W = \left[\begin{array}{ccccc} -1 & 3 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 3 & -1 \end{array} \right]$$

Let us visualise the above mask and another similar in MATLAB adding the following commands to our script which produce the adjacent plot.

```
h1 = [0   0   0 -1   3 -1   0   0 0] ;
h2 = [0 -2 -1   2   4   2 -1 -2 0] ;
n = floor(max(size(h1))/2)
figure(2)
plot(-n:n, [h1' h2']), grid on
```


Two Mexican hat masks
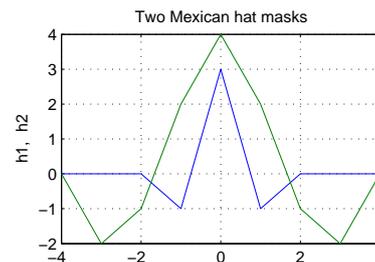
Such a plot with the positive and negative parts representing the lateral inhibition is called a **Mexican hat** mask.

Now, an exciting step is coming (next section) in which we apply a Mexican hat mask to one line of the image presented in Figure 2.

Let us first do some signal calculations related to eqn (4).
Assume now that the afferent signals and the mask are specified as follows

$$\texttt{u = [2 3 1 2]'} \quad \text{and} \quad \texttt{h = [-1 3 -1]}$$

and calculate $y_2 = -1 \cdot u_1 + 3 \cdot u_2 - 1 \cdot u_3 = 6$
Similarly $y_3 = -1 \cdot u_2 + 3 \cdot u_3 - 1 \cdot u_4 = -2$

Exercise 2. 2
For the above u and h **write down** the full form of the eqn (4) so that all individual signals and parameters are visible and **calculate** all efferent signals y_k for $k = 1, \dots, 4$.
□   3 marks

2.4.4 Correlation and convolution

From the above exercise you can see that the output signals are calculated sliding the mask **h** along the vector **u** and forming the required sum of products.

Calculation of such sum of products with a mask **h** sliding along the vector **u** is very tedious. Fortunately this is a typical operation with vectors called **correlation**. For the symmetric masks correlation is identical with **convolution**.

That is the function that we will be using in MATLAB. Such an operation with a mask is also called **filtering**.

Let us continue our MATLAB script in the following way:

```
u = [2 3 1 2]
h = [-1 3 -1]
y = conv(u, h)
```

The result of the convolution of two vectors, `u` and `h` should be `y=[-2 3 6 -2 5 -2]`. Simple and effective! Compare it with the results of your calculations in the last exercise.

Let us now apply a simple three-element mask

$$\mathbf{h} = [h_1 \ h_0 \ h_1] = \begin{bmatrix} -1 & 3 & -1 \end{bmatrix}$$

to a rectangular intensity profile as in Figure 2. Such a mask describes lateral inhibitory connection as in Figure 3. The relevant script is as follows:

```
% the intensity profile along the line 13
u = Z(13,27:56) ;  % row 13, columns 27 to 56
y = conv(u, h) ;   % magic happens here

figure(1)
subplot(2,1,1)
plot([0 u 0])
grid on, axis([0 30 -1 2.5])
title('original rectangular signal along receptors')
ylabel('u')
subplot(2,1,2)
plot(y)
grid on, axis([0 30 -1 2.5])
title('after application of lateral inhibition')
ylabel('y')
xlabel('pixel/receptor #')
```

The script generates plots as in Figure 4. From the plots we can note that the Mexican hat mask, inhibiting the neighbouring neurons, amplifies a change of intensity along the receptors which results in **enhancement of the image edges**.
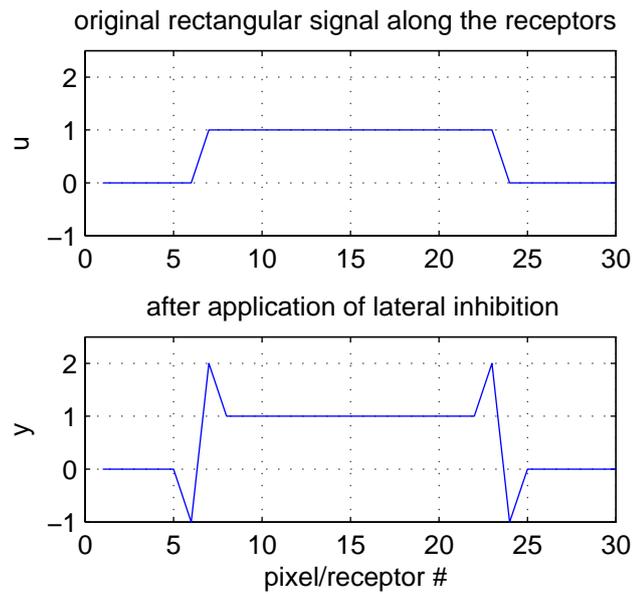
Figure 4: Sharpening the visual signal in a 1-D feedforward lateral inhibition circuit

2.4.5 Designing a good Mexican hat mask

In the processing we might like to make sure that the level of the signal is maintained, that is, if the afferent signal is constant one, then it stays constant one after processing with the mask. We will say that such a circuit/filter has a **unity dc gain**. The acronym 'dc' stands for 'direct-current' and is a shortcut for a constant level signal.

To ensure the unity dc gain we just need a mask in which **sum of its all coefficients is unity**, e.g,

$$y_c = [h_1 \ h_2 \ h_3] \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = h_1 + h_2 + h_3 = 1$$

More formally we can say that for a n-element mask the dc gain should satisfy the following condition:

$$y_c = \sum_{i=1}^{n} h_i = 1 \tag{5}$$

Another aspect that we would like to control in the output signal is the amount of negative/positive **overshoot**. This can be easily calculated if we imagine that the mask has its all negative values matching signal $u = 1$ all other mask's values being still at $u = 0$.

For a simple 3-element symmetric mask

$$h = [h_1 \ h_0 \ h_1], \quad \text{where} \quad h_1 < 0 \quad \text{end} \quad h_0 > 0$$

it just means that the overshoot is equal to the absolute value of h_1

$$y_m = |h_1|$$

and the output signal varies from $-y_m$ to $1 + y_m$.

For a general mask the overshoot y_m is a sum of all negative coefficients. If we number coefficients of the symmetric mask in the following way:

$$h = \begin{bmatrix} h_n & \dots & h_1 & h_{p1} & \dots & h_{pk} & h_1 & \dots & h_n \end{bmatrix}$$

then the overshoot can be calculated as follows:

$$y_m = |\sum_{i=1}^{n} h_i| \tag{6}$$

and the dc gain (constant level) as

$$y_c = \sum_{i=1}^{k} h_{pi} - 2y_m \tag{7}$$

For example the mask

$$h = \begin{bmatrix} -0.1 & -0.4 & 2 & -0.4 & -0.1 \end{bmatrix}$$

has the overshoot equal to

$$y_m = | -0.1 - 0.4| = 0.5$$

and the dc gain/constant level

$$y_c = -0.1 - 0.4 + 2 - 0.4 - 0.1 = 1$$

Exercise 2. 3
Design a 5-element Mexican hat mask with the overshoot equal to $y_m = 0.7$ and the unity dc gain, $y_c = 1$.
Plot the output signal for such a 5-element mask (as in Fig. 4).
Compare the slope of the edge for the 3-element and the 5 element mask.
□ 6 marks

2.4.6 Sigmoidal activation function

One, a bit inconvenient, effect of the processing with the Mexican hat mask is the change of the range of the numbers: they are expanded from the input range range, say $\{0, 1\}$, to the output range $\{-y_m, 1 + y_m\}$ both in the negative and positive directions due to overshoot.
One way of dealing with this problem is to pass signals through the saturating function, $y = \sigma(v)$ (see Figure 3) in order to put the signals back into the range $\{0, 1\}$.
Typically such a saturating function is defined as a **sigmoidal function** as illustrated in Figure 5. In Figure 5 'tanh' is a shifted hyperbolic tangent that can be expressed by means of an exponential function $\exp(v) = e^v$.

Let us see what happens if we add such a saturating function to our previous script in the following way:

$$y = \sigma(v) = \frac{1}{2}(\tanh(0.5v) + 1) = \frac{1}{1 + e^{-v}}$$
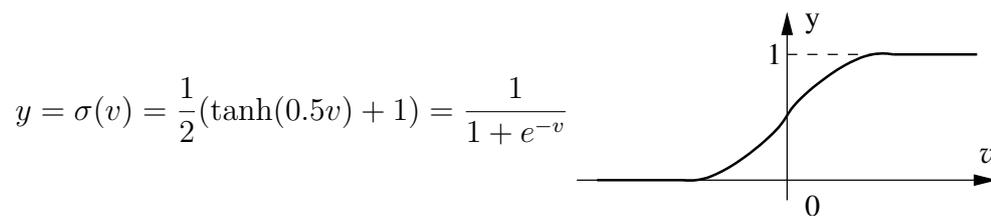
Figure 5: Sigmoidal function

```
MnMx  = [min(y)  max(y)]
y1  = 0.5*(tanh(0.5*y)+1) ;
MnMx1 = [min(y1)  max(y1)]
```

The resulting minimum and maximum values of the efferent signals are:

```
MnMx  = [-1   2]=   [h(1)   1-h(1)]
MnMx1 =  0.2689    0.8808
```

which means that the sigmoidal function squashed the range of numbers so that they are well in the range between 0 and 1, which might be desired.

Exercise 2. 4
Plot the output signals as in Figure 4 after passing them through the sigmoidal function. Use the 5-element mask from the previous exercise.
□   3 marks

2.4.7   From a convolution matrix to a weight matrix

It is a simple method computationally to use the convolution function, but it might be convenient to work with a weight matrix as in eqns (2), (3) and (4). Such a weight matrix will have a size equal to a number of neuronal inputs and outputs.

Let us consider again a simple feedforward network as in Figure 3 where we have $m = 5$ neurons and also $p = 5$ afferent signals. Assuming that $k = 3$ and that the mask $\mathbf{h} = [h_1 \; h_0 \; h_1]$, we can write a matrix equation describing processing of signals by the network similar to eqn (2):

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} h_0 & h_1 & 0 & 0 & 0 \\ h_1 & h_0 & h_1 & 0 & 0 \\ 0 & h_1 & h_0 & h_1 & 0 \\ 0 & 0 & h_1 & h_0 & h_1 \\ 0 & 0 & 0 & h_1 & h_0 \end{bmatrix}}_{W} \cdot \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix}}_{\mathbf{u}} \qquad (8)$$

It is good to know that the weight matrix W in this constant along diagonal form is called the **convolution matrix**. In MATLAB there is a function `convmtx` that can be used to form the weight matrix, W, from the mask $\mathbf{h}$. Note also that the matrix W of eqn (8) is a symmetric matrix (with respect to diagonal). Now we need a bit of a MATLAB practice to be able to swallow all the new bits.

```
u = [0 0 1 1 1]'   % a step function
h = [-1 3 -1]      % a mask as before
W = convmtx(h, 5)  % convolution matrix
W = W(:, 2:end-1)  % weight matrix (the 1st and last columns removed)
y = W*u            % output signals
yc = conv(u, h)    % to compare
```

Exercise 2. 5

Re-write eqn (8) for the 5-element mask $\mathbf{h} = [h_2 \; h_1 \; h_0 \; h_1 \; h_2]$.

Calculate the matrix W using the `convmtx` function for the mask as in exercise 2.3.

□   4 marks

2.4.8 Feedforward and recurrent networks

The **feedforward network** implemented above transforms input (afferent) signals into the output (efferent) signals in one processing step:

$$\mathbf{y} = W \cdot \mathbf{u}$$

In the next model of the limulus vision, we will add the feedback loops connecting the output signals to the input synapses. Networks with feedback are also referred to as **recurrent networks.** The following block diagrams clarify the concepts.

Feedforward network

Recurrent network
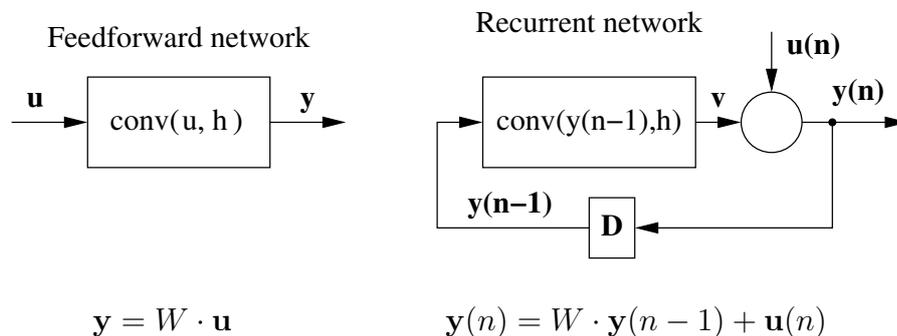
$$\mathbf{y} = W \cdot \mathbf{u} \qquad\qquad \mathbf{y}(n) = W \cdot \mathbf{y}(n-1) + \mathbf{u}(n)$$

The big circle at the output of the convolution block represents a summation $\mathbf{y} = \mathbf{v} + \mathbf{u}$. More precisely, the recurrent processing can be described by the following equation

$$\mathbf{y}(0) = \mathbf{u}(0), \quad \mathbf{y}(n) = W \cdot \mathbf{y}(n-1) + \mathbf{u}(n) = \text{conv}(\mathbf{y}(n-1), \mathbf{h}) + \mathbf{u}(n), \quad \text{for} \quad n = 0, 1, \dots \quad (9)$$

The variable n represents time and is the number of the current processing step. The **unit-delay** block "**D**" in the recurrent network block-diagram performs the delay of output signals by one time step, that is, it forms $\mathbf{y}(n-1)$ form $\mathbf{y}(n)$.

It can be safely assume that the feedback brings additional complexity in processing signals. Nature, however, has no problems in dealing with complexities. What we have to do is just follow nature.
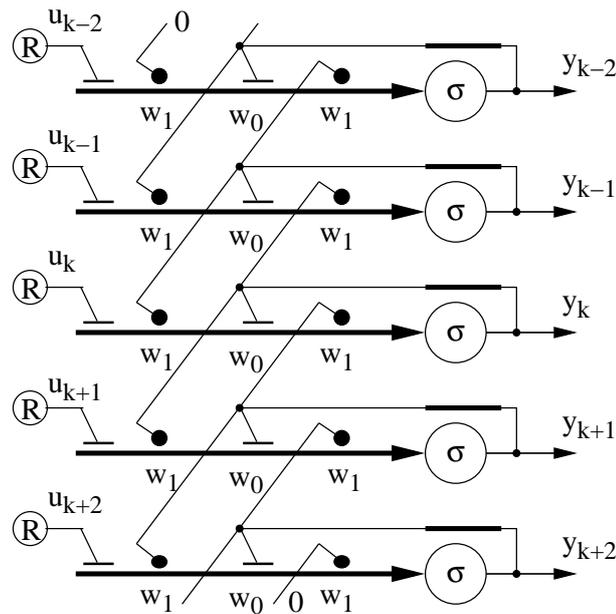
Figure 6: 1-D recurrent model of limulus vision

2.4.9   1-D recurrent model of limulus vision

A simple one-dimensional recurrent model of limulus vision is presented in Figure 6. Comparing with a feedforward network of Figure 3 we note that in the recurrent network the vision receptors are connected only to one synapse, whereas the laterally inhibiting signals are formed from the fed back output signals. This is a more realistic model of the limulus vision.

For the feedworward processing, we were able to design a mask according to eqns (5) and (6) to achieve a required behaviour, namely, to maintain the constant level y_c and to control the overshoot y_m.

Behaviour of recurrent networks as described by eqn (9) is significantly more complex, therefore designing the prescribed behaviour of a recurrent network is also, in general, difficult. In our case, however, with a simple 1-D Mexican hat mask we can again be in a full control of the network behaviour.

Exercise 2. 6
For a 3-element mask as in Figure 6 re-write eqn (9) in a form similar to eqn (8).
□   2 marks

Referring to eqn (9) we note that now, since we are adding the afferent signals **u**, we should modify the mask **h** so that the sum of its coefficients should be zero, that is,

$$h_c = \sum_{i=1}^{n} h_i = 0 \ ; \ \mathbf{h} = [h_1 \ldots h_n] \tag{10}$$

where h_i represents all coefficients of the mask. This will ensure maintaining the **constant level** through recurrent addition of **u**.

To work out **overshoots** is a bit more complicated. They depend on the sum of negative coefficients. One way of dealing with the complexity is to introduce one parameter, g, called **feedback gain** and multiply the Mexican hat mask by g:

$$\mathbf{h} = g \cdot [h_1 \ h_2 \ldots h_p \ldots h_2 \ h_1]$$

Now, we will find out that the amount of overshoot and its shape will depend, first of all, on the value of the feedback gain, g.

The MATLAB script that implements the recurrent network can have the following form:

```
m = 40 ;
u = zeros(m,1) ;
u((1:m/2)+m/4) = ones(m/2,1) ; % rectangular signal
g = 0.2 ;              % a gain parameter
h = g*[-1 2 -1] ;   % a mask
sum(h)                 % must be zero
W = convmtx(h, m); % convolution matrix
W = W(:, 2:end-1); % weight matrix
figure(3)
y = zeros(m, 1) ;   % initial value of the afferent signals
x = (1:m)' ;        % used in plotting
nn = 16 ;           % number of recurrent runs
for n = 1:nn
   y = W*y + u ;    % recurrent network
   plot(x, [u y] ), grid on
   title(sprintf('gain = %1.2f,    recurrent run n = %d', g,n))
   % axis([0 m -1 2])
   pause(2)         % the loop goes every 2 secs
end
```

The final form of the efferent signals is given in the following figure:
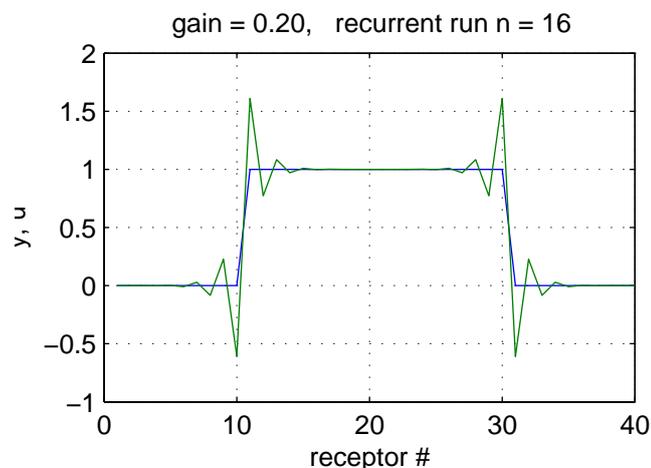
Note that the there are spatial (along the receptors) oscillations around the edges of the signal.

Exercise 2. 7

I would like you to observe how the gain g influences the behaviour of the recurrent network.

- Plot the maximum value of overshoots versus time steps (recurrent runs) $n = 1 \ldots 40$ for the following gains: $g = 0.16, 0.2, 0.24, 0.26$

- Note that there is a critical value of the gain (for a given mask) beyond which the network becomes **unstable**. Estimate this value.

  Unstable in particular means that the overshoots grow without limitations.

☐ 6 marks

2.5 Recurrent 2-D model

We now jump straight into 2-D recurrent networks that are real model of the limulus vision.

2.5.1 2-D structure of the limulus vision

The structure of such a network is a bit more complicated to represent, but the principle is the same as in Figure 6, namely, that the central neuron inhibits neighbouring neurons and is also inhibited by the neighbours. This idea is illustrated in Figure 7.
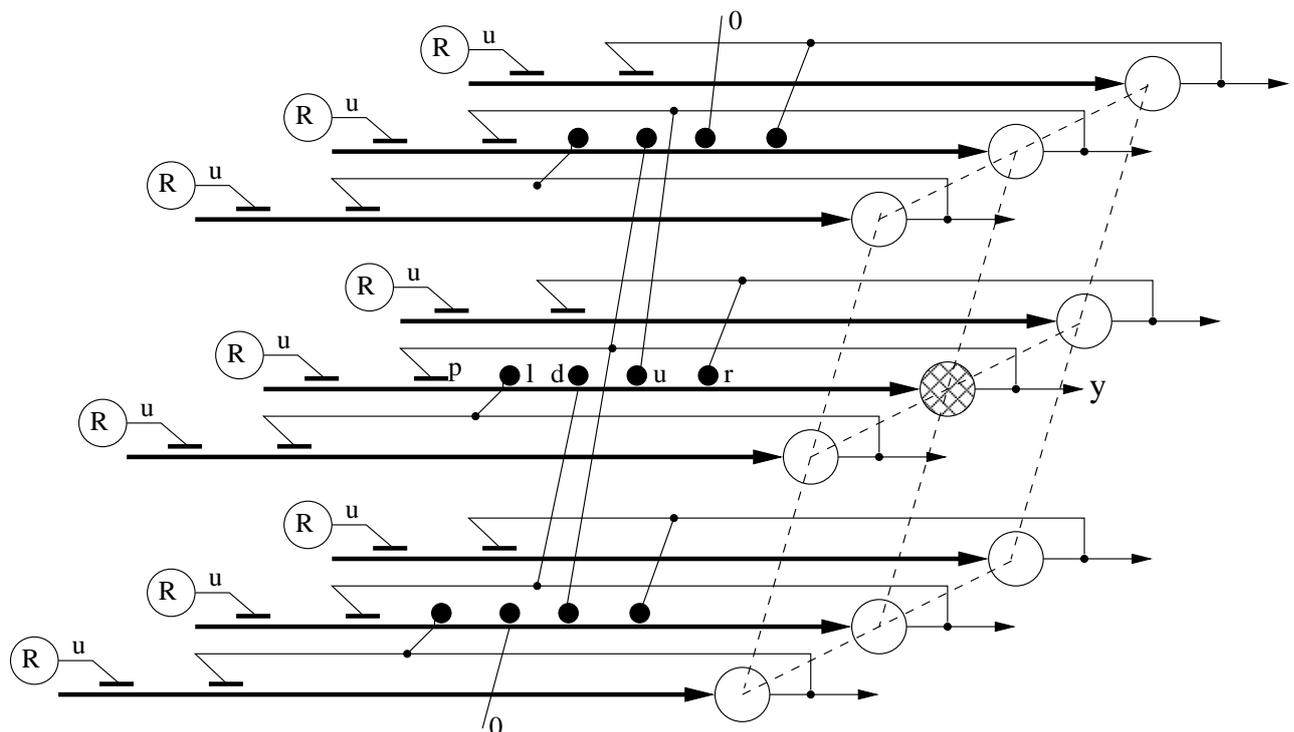
Figure 7: 2-D structure of the limulus vision system

In Figure 7 the central neuron marked y and checkered is inhibited by its four neighbours, from left, right, up and down. Relevant synapses and equivalent elements of the 2-D mask **h** are marked,

l, d, u, r, respectively. The mask element related to the local excitatory feedback signal is marked with p.

Conversely, every neuron, inhibits its four neighbours. For the checkered neuron only connections to neurons straight above and below are shown, for clarity.

The afferent signals from visual receptors are marked u in Figure 7, and they are organized into a matrix U, representing the image to be processed.

2.5.2   2-D masks and convolution

The simplest **2-D mask** describing interconnections between neurons as in Figure 7 can now be represented by the following matrix H:

$$H = g \cdot \begin{bmatrix} 0 & h_u & 0 \\ h_l & h_p & h_r \\ 0 & h_d & 0 \end{bmatrix} , \quad h_u, h_l, h_r, h_d < 0 , \quad h_p > 0 \tag{11}$$

where g is a feedback gain.
Following arguments from the previous section, in order to maintain the constant level of signals during recurrent runs, the the sum of all coefficients of the mask H must be zero:

$$h_u + h_l + h_r + h_d + h_p = 0 \tag{12}$$

Such a mask is applied to each element of processed image in a way similar to a 1-D case. For the simple mask as in eqn (11) an efferent signal y is formed as a linear combination of the mask elements and relevant image pixels (receptor signals):

$$y = h_p \cdot y + h_u \cdot y_u + h_l \cdot y_l + h_r \cdot y_r + h_d \cdot y_d + u$$

However complicated it sounds, the good news is that there exists a **2-D convolution** that takes an image, say U and a matrix mask, H, and creates the output image, say Y sliding the mask over every pixel of the input image and forming the necessary sum of products of mask elements and image pixels. Mathematically we just write:

$$Y = \text{conv2}(U, H) \tag{13}$$

We will use modification of this expression in a recurrent loop in the next section.

2.5.3   2-D recurrent network operations

A 2-D recurrent visual network operates as described in eqn (9) with a suitable change from vectors to matrices:

$$Y(0) = U(0), \quad Y(n) = \text{conv2}(Y(n-1), H) + U(n) , \quad \text{for} \quad n = 0, 1, \dots \tag{14}$$

We are now ready to process the simple test image presented in Figure 1. Let us start with a bit more complicated Mexican hat mask of the following form:

$$H = g \cdot \begin{bmatrix} -1 & -2 & -1 \\ -2 & 12 & -2 \\ -1 & -2 & -1 \end{bmatrix} \tag{15}$$

where gain g will be selected with stability of the processing in mind.

Before we start our next MATLAB exercise it might be a good idea to save the current script and open a new one named, say `myprac2d.m`. It will be still possible to copy and paste from the previous script if we need that.

In the new script we can type in the commands to create a mask as in eqn (15):

```
%  myprac2d.m
 clear, % close all
 g = 1/12
 HH = [-1 -2 -1 ; -2 12 -2 ; -1 -2 -1]
 % visualization of the mask
 H1 = zeros(5, 5) ; % adding a ring of zeros around the mask
 H1(2:4,2:4) = g*HH
 figure(1)
 surf(H1)
```

You should see a rather low resolution 2-D Mexican hat.

With a mask established, we can create a test image. We can copy the relevant commands from the previous script:

```
%  test image
 m = 100 ;
 x = 1:m ;
 [X Y] = meshgrid(x,x) ;
 U = ((X - m/2).^2 + (Y - m/2).^2) < (0.125*m^2) ;
 figure(1)
 imagesc(U), axis image
 grid on, colormap(1-gray(2))
```

This will reproduce the familiar cylinder. Finally, we can run our network with the selected image and mask:

```
 % recurrent 2-D network
 Y = zeros(size(U)) ;  % initial value of the afferent signals
 g = 0.04 ;   % adjustable feedback gain
 H = g*HH ;   % mask adjusted with gain
 nn = 10 ;    % number of recurrent runs
 figure(2)
 for n = 1:nn
   Y = conv2(Y, H, 'same') + U ;   % 2-D recurrent network
   subplot(2,2,1)
   imagesc(Y), axis image   % top view of the cylinder
```

```
    colormap(1-gray(256))
    text(60,-6, sprintf('gain = %1.2f,    recurrent run n = %d', g,n))
    subplot(2,2,2)
    surf(Y), grid on, view(-30, 20)        % perspective view
    axis([0 100 0 100 -1 2])
    subplot(2,2,3)
    plot(Y(50,1:30))        % cross-section along the row 50
    grid on, axis square
    xlabel('receptor # in row 50')
    pause(2)    % the loop goes every 2 secs
  end
```

The result, after 10 iterations, with a given gain is shown in Figure 8.
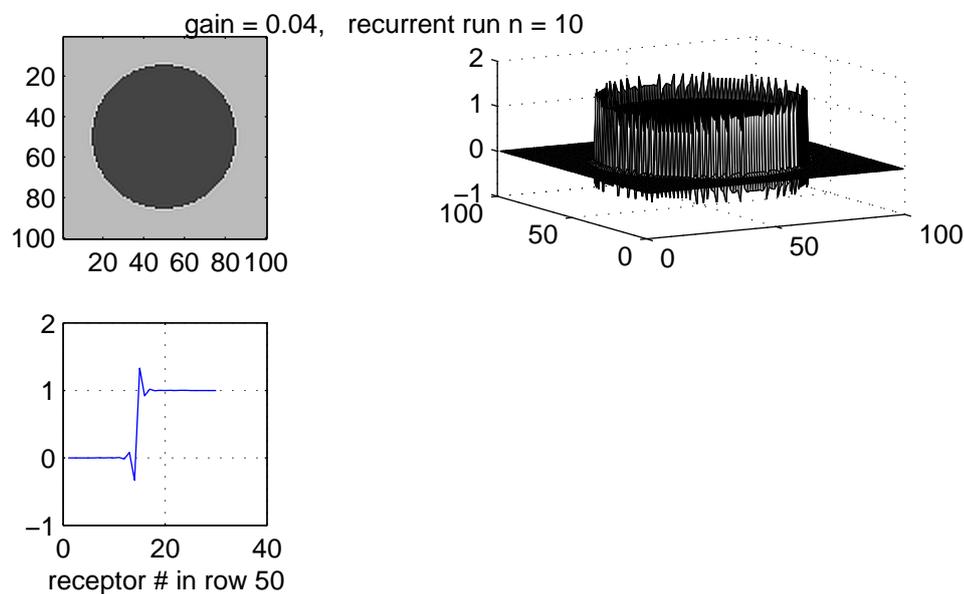
Figure 8: Processing a test image by a recurrent neural network

As in a 1-D case the jump in intensity results in sharpening the edge and creates an overshoot, or oscillations, depending on the value of the feedback gain.

Exercise 2. 8
Design your own 5×5 Mexican hat mask and repeat recurrent runs on the test image. Describe the results.

☐   4 marks

2.6   Images in MATLAB

Finally, we apply our 2-D vision network to processing a real image. For simplicity we use first with **gray scale** (monochrome) image. Such an image in MATLAB is just an $r \times c$ matrix, each element of the matrix (known as a **pixel**) representing intensity of the pixel.

Externally, images are stored in multiplicity of formats, such as **jpg**, **png**, **pdf**, **gif**, **tif** and many others. To begin with, download a test image from
`http://www.csse.monash.edu.au/coursware/cse2330/Pracs/apple.png`
to `$YFD/prac2` directory.

Now type in in the editor window the following commands:

```
clear
fnm = 'apple.png';          % an image file name
UU = imread(fnm, 'png') ;  % getting the image in a MATLAB matrix
figure(2)
image(UU), axis image,  colormap(gray(255))
```

Execute the above script. As a results you should get a gray scale image of an apple. The image is stored in a variable (matrix) `UU`. To get the basic parameters of the image execute the following commands

```
imsz = size(UU)
MnMx = [min(min(UU)) max(max(UU))]
```

Note that the image size is `imsz = [116 115]` and the range of pixel values (image intensities) is `MnMx = [1 245]` (assuming that you work with the same apple I do).

In the pane `Workspace` you should see all four variables that we created by now including their `sizes` and `classes`. Classes are a bit of a computational "bad news", but we need to be aware of the existence of two types: **double** and **uint8**.

The `uint8` type means "unsigned 8-bit integer" that will be used only when we load images in MATLAB. 8-bit integers can take values from 0 to 255.

All other numbers in MATLAB are of type **double**. They can store all values possible that you might ever need. Therefore, the next command we execute is conversion of our image from **uint8** to **double**. This can be done using the command:

```
U = double(UU)/256 ; % remember  ;  here !
```

In addition, by dividing by 256 we have scaled the range of pixel values to be between 0 and 1, which is more convenient for neuronal modelling.

The script processing an apple image looks as follows (remember to specify the mask `HH`):

```
% recurrent 2-D network
  U = double(UU)/256 ;  %  U is between 0 and 1
  Y = zeros(size(U)) ;  % initial value of the afferent signals
  g = 0.05 ;   % adjustable feedback gain
  H = g*HH ;   % mask adjusted with gain
```

```
nn = 10 ;     % number of recurrent runs
for n = 1:nn
   Y = conv2(Y, H, 'same') + U ;    % 2-D recurrent network
   imagesc(Y)
   axis image, colormap(gray(255))
   title(sprintf('gain = %1.2f,   recurrent run  n = %d', g,n))
   pause(2)    % the loop goes every 2 secs
end
```

The resulting apple has visibly sharpened edges and might look as in Figure 9.
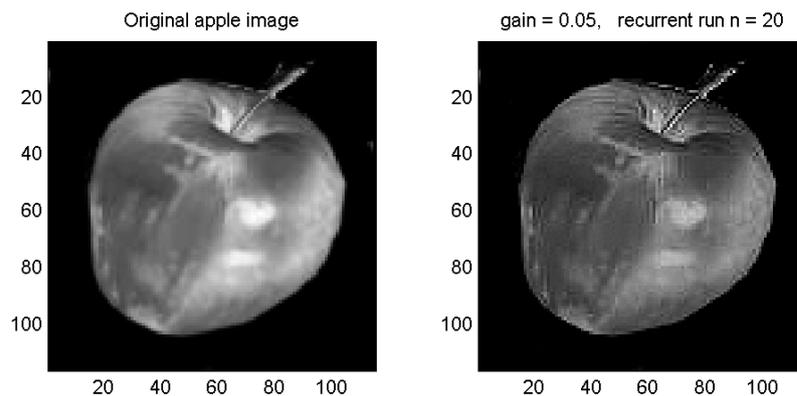
Figure 9: Original apple image and the image after sharpening its edges

Exercise 2. 9
As the final exercise write a MATLAB script to enhance edges in a **colour** image of your choice preferably in the jpg format. Use recurrent network with 5×5, or greater mask.

Note that a colour image in MATLAB is stored as an array of size $r \times c \times 3$, each $r \times c$ matrix storing a single primary colour information, that is, Red, Green, and Blue, respectively.

Use the 2-D convolution separately for each color component.

☐   6 marks

Submission

In **your report/submission** include answers/solutions to all exercises.

Include relevant derivations, equations, scripts, results and plots with suitable comments.

2.7   The Nobel Prize in Physiology or Medicine 1967

Presentation Speech by Professor C.G. Bernhard, Member of the Nobel Committee for Physiology or Medicine of the Royal Caroline Institute

Your Majesty, Royal Highnesses, Ladies and Gentlemen.

Light, shadows and colours do not exist in the world around us. What we perceive visually and call light is the result of the action of a certain portion of the electromagnetic radiation on the sensory cells in the retina of the eye. Our awareness of the play of light in nature, the multiplicity of the forms and the richness of the colours is ultimately dependent on the pattern of this radiation with respect to frequency and intensity. The light is composed of packets of energy, which combine the properties of waves and particles. When these particles – the quanta – strike the retina of the eye they are caught by the specialized sense cells – rods and cones. It is known that one quantum, which represents the least possible amount of light, is sufficient to initiate a reaction in a single rod. The excitation of the sensory cells results in messages directed towards the brain. As there are no direct connections from the eye to the brain the messages must be transmitted through several relays which combine signals from several sensory cells and translate the message into a language which can be understood by the brain. The primary relay is in the retina itself, represented by an intricate nerve net, the structural beauty of which was revealed by the neurohistologist Ramón y Cajal, Nobel laureate of 1906. In this complex structure messages from a great number of sensory cells converge on a far smaller number of optic nerve fibers and this results in a transformation of the pattern of signals.

Picasso has said: "To me painting is a sum of destructions. I paint a motif, then I destroy it. The painting goes through a series of metamorphoses but in the solution of the problem nothing has been lost. The final impression is still there in spite of all revisions". However, it is obvious to everyone that in the finished work a re-evaluation has taken place of the original elements of the motif. In some way this is a description of what happens in the visual system. An image of the outer world is formed on the retina in the same way as it is formed in the film of the camera. The image that falls on the closely packed mosaic of light sensory cells is disintegrated, since different cell types respond to various parts and qualities of the image. The primary data are then brought together in the nerve net in which a considerable processing takes place involving not only addition but also subtraction. This characterization of the message induces an impression in which there is a re-evaluation of the image projected on the retina. Does it mean that we cannot rely on what the eye tells us? No, not in the sense that there is full agreement between the external stimulus pattern and the composition of the impression. But rather in the sense that certain characteristics of the picture with essential biological and psychological significance are emphasized. There is a sharpening of contrast so that forms stand out more clearly, colours are exaggerated and movements accentuated.

We now know the mechanism by which light triggers off the reaction in the sensory cells of the eye thanks to the discoveries by George Wald and his coworkers among whom Ruth Hubbard – now Mrs. Wald – should be mentioned in the first place. The light-sensitive substances in the sensory cells, the visual pigments, consist in principle of two pieces. One, containing vitamin A, the smaller piece or the chromophore, fits like a hooked puzzle piece in the surface profile of the larger protein piece, the opsin. When a light quantum is taken up by the visual pigment the chromophore changes its form: there is an isomerization from II-cis to all-trans. The puzzle piece straightens out and releases itself from its position so that a successive splitting of the visual pigment follows. This

molecular transformation induced by light – the isomerization triggers the subsequent events in the visual system. All later changes – chemical, physiological and psychological – are as Wald says "dark" consequences of this single light reaction. Wald's conclusion that this reaction applies to the whole animal world also emphasizes the broad significance of his discovery.

Our ability to differentiate colours requires that different visual cells respond characteristically to different parts of the spectrum. Theories concerning the physiological basis of colour vision originated with Isaac Newton, Thomas Young and Hermann von Helmholtz. These theories were based on perception experiments. Today it is possible to attack this problem more directly with the aid of electronics which permits interpretation of the language of the nerve cells, thanks to the pioneer work in the 1920's by E.D. Adrian, Nobel laureate 1932. It is a great pleasure to see Lord Adrian here today and in this context I am reminded of his work with Yngve Zotterman which 40 years ago taught us the ABC's of the symbols in the sensory cells' language.

We honour Ragnar Granit for his discovery of elements in the retina possessing differential spectral sensitivities as determined by means of electrophysiological methods. The first work together with Svaetichin appeared in 1939. It was followed by an impressive series of investigations which led to the conclusion that there are different types of cones representing three characteristic spectral sensitivities. This important conclusion of Granit has recently been confirmed by Wald and collaborators as well as by research groups in U.S.A. and Great Britain using other methods. The discovery implies that the signal patterns which the optic nerve transmits to the brain and which result in perception of colours are dependent on the contributions from the three types of cone cells.

Keffer Hartline's elegant analysis of impulse generation in the sensory cells and the code they transmit in response to illumination of different intensity and duration has given us the basic understanding of how they evaluate the light stimulus. His later studies have led to the discovery of fundamental principles according to which the rough data from the sensory cells are re- evaluated. A precise quantitative analysis of the results was made possible by a refined technique and a careful choice of a suitable object – the eye of the horseshoe crab, a large marine spider. This approach to the problem led him to the discovery of the lateral inhibition, which in this eye was shown to be mediated by simple neuronal connections. Already in the 1930's Granit had shown the existence and importance of inhibition in the complex vertebrate retina. After having shown the interconnections of adjacent visual cells Hartline employed his discovery in a most imaginative way in order to obtain a quantitative description how a nerve-net processes the data from the sensory cells by means of inhibition. His discoveries have in a unique manner contributed to our understanding of the physiological mechanism whereby heightened contrast sharpens the visual impressions of form and movement.

Professor Granit, Professor Hartline, Professor Wald. Your discoveries have deepened our insight into the nature of the subtle processes in the eye which form the basis of our ability to perceive light and to distinguish brightness, colour, form and movement. They have also proved to be of paramount importance for the understanding of sensory processes in general.

Professor Granit. About 100 years ago the distinguished physiologist in Uppsala, Frithiof Holmgren, discovered the electrical response of the eye to light. The hopes that he expressed for the future regarding the possibilities of all electrophysiological analysis of the retinal processes and the mechanism of colour vision have been realized by your distinguished discoveries. These show the

importance of inhibition in the integrative action of the retina and the principles for spectral discrimination by retinal elements. Your discoveries have pointed the way in modern physiology of vision and your stimulating research work has contributed to the fruitful development of this field.

Professor Hartline. Your laboratory has been described as a "slightly disorganized but extremely fertile chaos". Your work which – by the same right has been characterized by elegance in design, expertise in manipulation and clarity in exposition has resulted in an exemplary limited number of publications, each of which is a corner-stone in sensory physiology. They have given us the basic knowledge about the impulse coding in the visual receptors and presented discoveries of the most fundamental principles for data processing in neuronal networks which serve sensory functions. In the case of vision they are vital for the understanding of the mechanisms underlying perceptions of brightness, form and movement.

Professor Wald. With a deep biological insight and a great biochemical skill you have successfully identified visual pigments and their precursors. As a byproduct you were able to describe the absorption spectra of the different types of cones serving colour vision. Your most important discovery of the primary molecular reaction to light in the eye represents a dramatic advance in vision since it plays the role of a trigger in the photoreceptors of all living animals.

Gentlemen. It is with great satisfaction that Karolinska Institutet has decided to award you this year's Nobel Prize for physiology or medicine for your discoveries concerning the primary physiological and chemical visual processes. On behalf of the Institute I wish to extend to you our warm congratulations and I ask you to receive the prize from the hands of His Majesty the King.

From Les Prix Nobel 1967.

http://www.nobel.se/medicine/laureates/1967/press.html