

CS5486 Intelligent Systems - Assignment 3

A 5-neuron MAXNET with lateral connection weight of -0.15, and external input vector of (0.1, 0.3, 0.5, 0.7, 0.9)

```
clear;
input = [0.1, 0.3, 0.5, 0.7, 0.9];
x = input;
[p,n] = size(x);
epsilon = -0.15; %weight of inhibitory lateral connections
w = ones(5,5) * epsilon - (eye(5) * epsilon) + eye(5); %weight matrix
it = 0; %iteration
stop = false; %stopping state for finding the winner
while (stop == false)
    it = it+1;
    for i=1:n %update the activation of each node
        u(i) = x*w(i,:);
        if u(i) > 0 %ReLU
            v(i) = u(i);
        else
            v(i) = 0;
        end
    end
    x = v; %recurrent output, save activation for use in next iteration
    count = 0; %for counting non-zero node
    for i = 1:n
        if v(i) ~= 0 %check if the node has a non-zero activation
            count = count + 1;
            if count > 1 %continue if more than one non-zero node
                stop = false;
                break;
            elseif count == 1 %stop when there's only one non-zero node
                stop = true;
                winner_index = i; %winner
            end
        end
    end
end
end
```

The winner is the fifth node with the maximum input as 0.9 as it is the only node that has a non-zero activation as 0.4541 after six iterations.

A k-Winners-Take-All Model (kWTA) where $k = 1$ with a Single State Variable (y) and the Heaviside Step Activation Function

```
clear;
k = 1;
input = [0.1, 0.3, 0.5, 0.7, 0.9];
u = input;
[p,n] = size(u);
y = 0; %state variable
deltaT = 0.05;
it = 0; %iteration
stop = false; %stopping state
for i = 1:n %initialization
    if u(i) >= y %infinite gain activation function
        x(i) = 1;
    else
        x(i) = 0;
    end
end
```

This study source was downloaded by 100000895614552 from CourseHero.com on 01-18-2025 02:42:19 GMT -06:00

```

end
end
while (stop == false)
    it = it + 1;
    y = y + deltaT * (sum(x) - k);
    for i = 1:n
        if u(i) >= y %infinite gain activation function
            x(i) = 1;
        else
            x(i) = 0;
        end
    end
    count = 0;
    for i = 1:n
        if x(i) ~= 0 %check if the node has a non-zero activation
            count = count + 1;
            if count > 1 %continue if more than one non-zero node
                stop = false;
                break;
            elseif count == 1 %stop when there's only one non-zero node
                stop = true;
                winner_index = i; %winner
            end
        end
    end
end
end
end

```

The kWTA runs for seven iterations and obtains the winner as the fifth node with y equals to 0.7 and output vectors as $[0, 0, 0, 0, 1]$ at the end.

kWTA goes for 1 iteration longer than MAXNET to find the winner.

A 9-neuron discrete-time Hopfield network as an associative memory of 3-by-3 digital images

```

clear;
x1 = [-1,+1,+1, -1,-1,-1, -1,+1,+1,]';
x2 = [-1,-1,-1, -1,+1,-1, -1,+1,-1,]';
W1=x1*x1'-eye(9);
W2=x2*x2'-eye(9);
W=W1+W2;
img1=reshape(x1,[3 3 1 1]); figure(1),subplot(1,5,1),imshow(255*uint8(img1));xlabel(' (a)');
img2=reshape(x2,[3 3 1 1]); figure(1),subplot(1,5,2),imshow(255*uint8(img2));xlabel(' (b)');
y1 = [-1,+1,-1, -1,-1,-1, -1,+1,+1,]';
y2 = [-1,-1,-1, -1,+1,-1, -1,+1,+1,]';
img21=reshape(y1,[3 3 1 1]); figure(2),subplot(1,5,1),imshow(255*uint8(img21));xlabel(' (a)');
img22=reshape(y2,[3 3 1 1]); figure(2),subplot(1,5,2),imshow(255*uint8(img22));xlabel(' (b)');
y11=W*y1;img31=reshape(y11, [3 3 1 1]);
figure(3),subplot(1,5,1),imshow(255*uint8(img31));xlabel(' (a)');
y22=W*y2;img32=reshape(y22, [3 3 1 1]);
figure(3),subplot(1,5,2),imshow(255*uint8(img32));xlabel(' (b)');

```

The two original patterns could be retrieved from the two noisy patterns, which is shown in the following picture as the outputs of the program:



(a)

(b)

Original Patterns



(a)

(b)

Noisy Patterns



(a)

(b)

Retrieved Patterns