|  |
| --- |
| **Department of Computer Sci. &Engineering** |
|  |
| **LAB MANUAL** |

|  |
| --- |
| **BE(CSE)**  **Soft Computing** |

|  |
| --- |
| D:\JNEC-My-PC-D-Drive-20-2-15\DEPT-1-Jul-13-to-30-Jun-14\Manuals\SC-image.jpg |

|  |
| --- |
| **MGM's**  **Jawaharlal Nehru Engineering College, Aurangabad** |

### **FOREWORD**

It is my great pleasure to present this laboratory manual for Final year Engineering students for the subject – “ Soft Computing”.

As a student, many of you may be wondering with some of the questions in your mind regarding the subject and exactly what has been tried is to answer through this manual.

You may be aware that MGM has already been awarded with ISO 9001:2000 certification and it is our endure to technically equip our students taking the advantage of the procedural aspects of ISO 9001:2000 Certification.

Faculty members are also advised that covering these aspects in initial stage itself will greatly relieve them in future as much of the load will be taken care by the enthusiasm energies of the students once they are conceptually clear.

**Dr. S.D. Deshmukh**

**Principal**

### **LABORATORY MANUAL CONTENTS**

This manual is intended for the final year students of Computer Science and Engineering in the subject of Soft Computing. This manual typically contains practical/Lab Sessions related to the subject on the MATLAB/Scilab platform covering various aspects of the subject to enhance understanding.

Students' are advised to thoroughly go through this manual rather than only topics mentioned in the syllabus as practical aspects are the key to understanding and conceptual visualization of theoretical aspects covered in the books.

Good Luck for your Enjoyable Laboratory Sessions.

Dr.Vijaya B. Musande Dr. Madhuri S. Joshi

Head, CSE Dept. Professor -CSE Dept.

**DOs and DON’Ts in Laboratory:**

1. Make entry in the Log Book as soon as you enter the Laboratory.

2. All the students should sit according to their roll numbers starting from their left to right.

3. All the students are supposed to enter the terminal number in the log book.

4. Do not change the terminal on which you are working.

5. All the students are expected to get at least the algorithm of the program/concept

to be implemented.

6. Strictly observe the instructions given by the teacher/Lab Instructor.

**Instruction for Laboratory Teachers::**

1. Submission related to whatever lab work has been completed should be done during the next lab session. The immediate arrangements for printouts related to submission on the day of practical assignments.

2. Students should be taught for taking the printouts under the observation of lab teacher.

3. The promptness of submission should be encouraged by way of marking and evaluation patterns that will benefit the sincere students.

**MGM’s**

****

**Jawaharlal Nehru Engineering College, Aurangabad**

**Department of Computer Science and Engineering**

**Vision of CSE Department**

To develop computer engineers with necessary analytical ability and human values who can creatively design, implement a wide spectrum of computer systems for welfare of the society.

**Mission of the CSE Department:**

1. Preparing graduates to work on multidisciplinary platforms associated with their professional position both independently and in a team environment.
2. Preparing graduates for higher education and research in computer science and engineering enabling them to develop systems for society development.

**Programme Educational Objectives**

**Graduates will be able to**

1. To analyze, design and provide optimal solution for Computer Science & Engineering and multidisciplinary problems.
2. To pursue higher studies and research by applying knowledge of mathematics and fundamentals of computer science.
3. To exhibit professionalism, communication skills and adapt to current trends by engaging in lifelong learning.

**Programme Outcomes (POs):**

**Engineering Graduates will be able to:**

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems anddesign system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms ofthe engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage independent and life-long learning in the broadest context of technological change.

**Class: BE(CSE) Subject: Lab VII- Soft Computing (CSE 473)**

**Experiment No. 1**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Title: To study MATLAB software and its toolboxes.**

**Objectives:**

* **To study MATLAB platform**
* **To know information about Scilab**

**Instructions:**

Study MATLAB menus. Describe it with respect to following points:

1. What is MATLAB?
2. What are its areas of applications? What is meant by a Tool Box?
3. Describe Neural Network Tool Box in brief.
4. What is the structure of a MATLAB program?
5. What is Scilab?

**Refer the files:** 1. introtomatlab-2.ppt 2. MATLAB\_and\_SciLab\_Comparison.ppt

(Sent on class group mail id)

**Conclusion:**

**===================================================**

**Class: BE(CSE) Subject: Lab VII- Soft Computing (CSE 473)**

**Experiment No. 2**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Title: Write a program to implement MP Model.**

**Objectives:**

* **To implement AND logic gate**
* **To implement OR logic gate**

**Instructions:**

1. Study MP Model
2. Study logic gates: AND,OR,NOT,Exclusive-OR
3. Write a program to implement AND gate without using neural network toolbox.
4. Write a program to implement OR gate without using neural network toolbox.

**Program to implement AND gate:**

%AND function using McCulloch-Pitts neuron

clear;

clc;

% Getting weights and threshold value

disp('Enter the weights');

w1=input('Weight w1=');

w2=input('Weight w2=');

disp('Enter threshold value');

theta=input('theta=');

y=[0 0 0 0];

x1=[1 1 0 0];

x2=[1 0 1 0];

z=[1 0 0 0];

con=1;

while con

zin = x1\*w1+x2\*w2;

for i=1:4

if zin(i)>=theta

y(i)=1;

else y(i)=0;

end

end

disp('Output of net=');

disp(y);

if y==z

con=0;

else

disp('Net is not learning Enter another set of weights and threshold value');

w1=input('Weight w1=');

w2=input('Weight w2=');

theta=input('theta=');

end

end

disp('McCulloch Pitts Net for ANDNOT function');

disp('Weights of neuron');

disp(w1);

disp(w2);

disp('Threshold value=');

disp(theta);

**Output :-**

Enter the weights

Weight w1=1

Weight w2=1

Enter threshold value

theta=2

Output of net= 1 0 0 0

McCulloch Pitts Net for AND function

Weights of neuron

1

1

Threshold value=

2

**Implement OR gate.**

**Conclusion:**

**===================================================**

**Class: BE(CSE) Subject: Lab VII- Soft Computing (CSE 473)**

**Experiment No. 3**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Title: Write a program for solving linearly separable problem using Perceptron Model.**

**Objectives:**

* **To implement AND function using Perceptron**

**Instructions:**

1. Study Perceptron Model
2. Write a program to implement AND function
3. Draw the diagram of AND using Perceptron used below

**Program to implement AND function:**

%Perceptron for AND funtion

clear;

clc;

x=[1 1 -1 -1;1 -1 1 -1];

t=[1 -1 -1 -1];

w=[0 0];

b=0;

alpha=input('Enter Learning rate=');

theta=input('Enter Threshold value=');

con=1;

epoch=0;

while con

con=0;

for i=1:4

yin=b+x(1,i)\*w(1)+x(2,i)\*w(2);

if yin>theta

y=1;

end

if yin <=theta & yin>=-theta

y=0;

end

if yin<-theta

y=-1;

end

if y-t(i)

con=1;

for j=1:2

w(j)=w(j)+alpha\*t(i)\*x(j,i);

end

b=b+alpha\*t(i);

end

end

epoch=epoch+1;

end

disp('Perceptron for AND function');

disp(' Final Weight matrix');

disp(w);

disp('Final Bias');

disp(b);

**Output**

Enter Learning rate=1

Enter Threshold value=0.5

Perceptron for AND function

Final Weight matrix

1 1

Final Bias

-1

**Conclusion:**

**===================================================**

**Class: BE(CSE) Subject: Lab VII- Soft Computing (CSE 473)**

**Experiment No. 4**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Title: Write a program for pattern classification using Perceptron Model**

**Objectives:**

* To Perceptron for simple pattern classification
* To learn mapping of patterns and design of neural network for such problem

**Problem statement:**

Classify two dimensional patterns defined by a 3x3 matrix using a simple perceptron network. Assume any two English alphabets as input, target output is defined as 1 for one alphabet and -1 for another. Assume suitable initial weights, bias and threshold and mention them.

Write a program to train the network Output the weight matrix after each epoch. Mention how many epochs are required for learning. The program should be able to test the alphabet by taking input from user.

**Test the network:**

Read a same/noisy data ( 9 value string ) as an input. The program should display whether the input is recognised, not recognised or close to alphabet.

**Instructions:**

1. Draw the diagram of the network used.

**Output:**

Indicate output in three different cases at least.

**Exercise:**

1. Will the above network work, if the input alphabets/ patterns to be recognized / classified are more than two?

How should it be modified to classify four input patterns? Devise the scheme at least.

1. Modify your program to incorporate this change.

**Conclusion:**Write the conclusion by observing the outputs in various cases.

**===================================================**

**Class: BE(CSE) Subject: Lab VII- Soft Computing (CSE 473)**

**Experiment No. 5**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Title: Write a program for XOR function (binary input and output) with momentum factor using back propagation algorithm.**

**Objectives:**

* To understand the concept of linear separability associated with simple XOR problem
* To learn and implementback-propagation learning algorithm

**Training Algorithm:**

The Back-propagation (BP) learning algorithm can be outlined in the following algorithm:

**Step 0:** Initialize weights and learning rate(take some small random values).

**Step 1:** Perform Steps 2-9 when stopping condition is false.

**Step 2:** perform 3-8 for each training pair.

**Feed-forward phase (Phase I):**

**Step 3:** Each input unit receives input signal xi and sends it to the hidden unit (i=1 to n).

**Step 4:** Each hidden unit zj (j=1 to p) sums its weighted input signals to calculate net input:

Zinj=v0j+

Calculate output of the hidden unit by applying its activation functions over zinj (binary or bipolar sigmoidal activation function):

Zj=f(zinj)

and send the output signal from the hidden unit to the input of output layer units.

**Step 5:** For each output unit yk (k=1 to m), calculate the net input:

yink=w0k+

And apply the activation function to compute output signal

yk=f(yink)

**Back-propagation of error (Phase II):**

**Step 6:** Each output unit yk  (k=1 to m) receives a target pattern corresponding to the input training pattern and computes the error correction term:

=()f’()

Calculate derivative of f’(yink). On the basis of the calculated error correction term, update the change in weights and bias:

Also, send k to the hidden layer backwards.

**Step 7:** Each hidden unit (zj, j=1 to p) sums its delta inputs from the output unit:

=

The term gets multiplied with the derivative of f() to calculate the error term:

f’()

The derivative f’() can be calculated depending on whether binary or bipolar sigmoidal function is used. On the basis of the calculated , update the change in weights and bias:

=; =

**Weight and bias updation (Phase III):**

**Step 8:** Each output unit (, k=1 to m) updates the bias and weights:

=+

=+

Each hidden unit (, j=1 to p) updates its bias and weights:

=+

=+

**Step 9:** Check for the stopping condition. The stopping condition may be certain number of epochs reached or when the actual output equals the target output.

Truth Table for XOR function (binary input and output) is given as:

|  |  |  |
| --- | --- | --- |
| X1 | X2 | Y |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Make use of sigmoid function for activation.

**Instructions:**

1. Implement the algorithm:
2. Write the weight matrix and bias.
3. Display the Output for the test cases

**Conclusion:**

**===============**

**Class: BE(CSE) Subject: Lab VII- Soft Computing (CSE 473)**

**Experiment No. 7**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Title: Write a program to store a pattern (1 1 1 0). Test the network using Discrete Hopfield Net by giving the input with mistakes in First and Second position**

**Objectives:**

* To understand the concept of Pattern storage
* To implement **Discrete Hopfield Net**

**Training Algorithm:**

* Describe about Pattern Storage networks
* Write theory for Hopfield Net.

**MATLAB Source Code:**

%Discrete hopfield net

clc;

clear;

x=[1 1 1 0];

tx=[0 0 1 0];

w=(2\*x'-1)\*(2\*x-1);

for i=1:4

w(i,i)=0;

end

con=1;

y=[0 0 1 0];

while con

up=[4 2 1 3];

for i=1:4

yin(up(i))=tx(up(i))+y\*w(1:4,up(i));

if yin(up(i))>0

y(up(i))=1;

end

end

if y==x

disp('convergence has been obtained');

disp('the converged output');

disp(y);

con=0;

end

end

**%output**

**%convergence has been obtained**

**%the converged output**

**% 1 1 1 0**

**Conclusion:** *Write conclusion after executing the program.*

**===============**

**Class: BE(CSE) Subject: Lab VII- Soft Computing (CSE 473)**

**Experiment No. 8**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Title: Program for Pattern storage of 10 digits with Discrete Hopfield Network**

**Objectives:**

* **To study pattern storage function using Discrete Hopfield Network using build in function of MATLAB**
* **To test the network for noisy inputs**

**Theory:** Write theory of Discrete Hopfield Network. Draw the diagram of the network too.

**Algorithm:**

Write the training and the testing algorithm used in the program below.

**Source Code:**

%Program for Pattern storage of 10 digits - 0-9- using Discrete Hopfield

%Network. Testing the network using a noisy input pattern.

close all

clc

clear

load ('D:\MSJ\pat.mat')

figure(1)

pat2=pat;

pat2(pat2>=0)=255;

pat2(pat2<0)=0;

pat2=reshape(pat2, [10 100/10\*size(pat,2)]);

image(pat2)

iterations=10;

character=5;

net=newhop(pat);

d2=pat(:,character);

r=rand(size(d2));

figure(2)

digit(d2)

title(sprintf('Original digit %i',character))

%A bit is 'flipped' with probalility (1-lim)

lim=.7;

d2(r>lim)=-d2(r>lim);

figure(3)

digit(d2)

title(sprintf('Digit %i with noise added',character))

[Y, Pf, Af] = sim(net, {1 iterations}, [ ], {d2});

Y=cell2mat(Y);

figure(4)

digit(Y)

title('All iterations of Hopfield Network')

axis equal

%------End of Program------------

%-----Function------------

function [ ] = digit(pat)

%load pat

% change color

pat2=pat;

pat2(pat2>=0)=255;

pat2(pat2<0)=0;

%pat2(pat2==-1)=255;

pat2=reshape(pat2, [10 100/10\*size(pat,2)]);

image(pat2)

end

%--Input Data----------

% File pat.mat

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 –1

1 1 1 1 1 1 1 1 1 –1

1 1 1 1 1 1 1 1 1 –1

1 1 1 1 1 1 1 1 1 –1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 –1 –1 1 –1 1 1 1 1 1

1 1 1 1 –1 1 1 –1 1 –1

1 1 1 1 –1 –1 1 –1 1 1

1 1 1 1 –1 –1 1 –1 1 1

1 1 1 1 1 –1 1 1 1 1

1 1 1 1 1 –1 1 –1 1 1

1 1 1 1 1 –1 1 –1 1 –1

1 –1 –1 1 1 –1 1 –1 1 1

1 –1 1 1 –1 1 1 1 1 1

1 –1 –1 1 1 1 1 1 1 1

–1 1 1 –1 –1 1 –1 –1 1 –1

1 1 1 –1 1 –1 1 1 1 1

1 1 1 –1 1 1 1 1 1 1

1 1 1 –1 –1 1 1 1 1 1

1 1 1 –1 1 –1 1 –1 1 1

1 1 1 1 1 1 1 1 1 1

1 –1 1 1 1 1 1 1 1 1

1 –1 1 1 1 1 1 1 1 –1

1 –1 –1 1 –1 –1 –1 –1 1 1

1 –1 –1 1 1 1 1 1 1 –1

–1 1 1 1 –1 –1 –1 –1 1 1

1 1 1 1 1 1 1 1 –1 1

1 1 1 1 1 1 1 1 –1 1

1 1 1 1 –1 1 1 1 –1 1

1 1 1 –1 1 –1 1 –1 1 1

1 –1 1 1 1 1 1 1 1 1

1 –1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 –1 1 1 1

–1 –1 –1 1 –1 –1 1 –1 1 –1

–1 –1 –1 1 1 –1 1 1 1 –1

–1 1 1 1 –1 1 –1 –1 –1 1

–1 1 1 1 1 1 1 1 1 1

–1 1 1 1 1 1 1 1 1 1

–1 1 –1 1 1 1 1 1 1 1

–1 –1 –1 –1 –1 –1 1 –1 –1 1

–1 –1 1 1 1 1 1 1 1 1

–1 1 1 1 1 1 –1 1 1 1

–1 1 1 1 1 1 1 1 1 1

–1 –1 –1 1 –1 –1 1 –1 –1 –1

1 –1 –1 1 1 1 1 1 1 1

1 1 1 1 –1 1 –1 1 –1 –1

1 1 1 1 1 1 1 –1 1 1

1 1 1 1 1 1 1 –1 1 1

1 –1 –1 1 1 1 1 –1 1 1

1 –1 –1 –1 1 1 1 1 –1 1

1 1 1 1 –1 –1 –1 –1 1 1

1 1 1 1 –1 –1 1 –1 1 1

1 1 1 1 –1 –1 1 –1 –1 –1

–1 –1 –1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 –1 –1 –1 1 1 1 1 –1 1

1 –1 –1 –1 1 1 –1 1 1 –1

1 –1 –1 –1 1 1 –1 1 1 1

1 –1 1 –1 1 1 –1 1 1 1

1 1 1 –1 1 1 –1 1 –1 1

1 1 –1 –1 1 1 1 1 1 1

1 1 –1 –1 1 1 1 1 –1 –1

1 1 –1 –1 1 1 1 1 1 1

1 –1 1 –1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 –1 1

1 1 1 1 1 1 1 1 –1 –1

1 1 1 1 1 1 1 1 –1 –1

1 1 1 1 1 1 1 1 –1 –1

1 1 1 1 1 1 1 1 –1 –1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

%-------------------------------------------------------------

**Class: BE(CSE) Subject: Lab VII- Soft Computing (CSE 473)**

**Experiment No. 11**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Title:** To studyfuzzy control, principle of Fuzzy Control Design and rule-based Fuzzy Inference System (FIS)

**Answer the following questions:**

1. What are the methods of fuzzyfication?
2. Explain with the block diagram a Fuzzy Inference System (FIS or Expert System).
3. What are the steps to be followed to compute the output from Mamdani FIS?
4. Describe in short the following primary GUI tools available with MATLAB Fuzzy Toolbox.
   1. Fuzzy Inference System (FIS) Editor
   2. Membership Function Editor
   3. Rule Editor
   4. Rule Viewer
   5. Surface Viewer
5. Describe any **one application / device** making use of fuzzy control. How and where the fuzzy control is used in that device?

*{ Students should select different applications. There are plenty available on the web reference.* }

**=====================**