

# FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**B.E (CSE)** 

**VII - SEMESTER** 

# **08CP706 - SOFT COMPUTING TECHNIQUES LAB**

Name:

Reg. No. :....



# FACULTY OF ENGINEERING AND TECHNOLOGY

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# **BONAFIDE CERTIFICATE**

Staff In-charge

**Internal Examiner** 

Place: Annamalainagar

**External Examiner** 

Date:

# **INDEX**

Ex. No.	DATE	LIST OF EXERCISES	PAGE No.	SIGNATURE
1		Performing Union, Intersection and Complement operations		
2		Implementation of De-Morgan's Law		
3		Plotting various membership functions		
4		Using fuzzy toolbox to model tips value		
5		Implementation of Fuzzy Inference System		
6		Simple fuzzy set operations		
7		Using Hopfield network with no self connection		
8		Generation of ANDNOT function using McCulloch-Pitts neural net		
9		Finding weight matrix and bias of HebbNet to classify two dimensional input patterns		
10		Perceptron net for AND function with bipolar		
		inputs and targets		
11		Finding weight matrix of Hetero-Associative		
		neural net for mapping of vectors		
12		Generation of XOR function using back		
		propagation algorithm		

# Performing Union, Intersection and Complement Operations

#### Ex. No: 1

#### Date:

#### Aim :

To write a Program in MATLAB to perform union, intersection and complement operations of fuzzy set.

#### Algorithm:

- 1. Read the membership values of two fuzzy sets.
- 2. Perform union operation by using max() function.
- 3. Perform intersection operation by using min() function.
- 4. Perform complement operation by subtracting membership value from 1
- 5. Display the result.

#### **Program:**

% Enter the membership value of fuzzy set

u = input ('Enter the membership value of First Fuzzy set');

v = input ('Enter the membership value of Second Fuzzy set');

%performs Union, Intersection and Complement operations

w=max (u, v);

p=min (u, v);

q1=1-u;

q2=1-v;

%Display Output

disp('Union of Two Fuzzy sets');

disp(w);

disp('Intersection of Two Fuzzy sets');

disp(p);

disp('Complement of First Fuzzy set'); disp(q1); disp('Complement of Second Fuzzy set'); disp(q2);

#### Sample Input and Output:

Enter the membership value of First Fuzzy set [0.3 0.4]
Enter the membership value of Second Fuzzy set [0.1 0.7]
Union of Two Fuzzy sets
0.3000. 0.7000
Intersection of Two fuzzy sets
0.1000 0.4000.
Complement of First Fuzzy set
0.7000 0.6000.
Complement of Second Fuzzy set
0.9000 0.3000.

#### **Result:**

Thus, the MATLAB program to perform Union, Intersection and Complement operations of two Fuzzy sets has been executed successfully and the output is verified.

# Implementation of De-Morgan's Law

#### Ex. No: 2

### Date:

## Aim:

To write a Program in MATLAB to implement De-Morgan's law.

## Algorithm:

- 1. Read the membership values of two fuzzy set.
- 2. Perform Union operation by using max() function and take the complement for the fuzzy set.
- 3. Perform Intersection operation by using min() function and take the complement.
- 4. Perform Complement operation for the both fuzzy sets.
- 5. Perform Intersection operation and Union operation for the Complements of fuzzy set.
- 6. Verify the formula and display the result.

#### **Program:**

% Enter the membership values of fuzzy set

u = input ('Enter the membership values of first fuzzy set');

v = input ('Enter the membership values of second fuzzy set');

%To perform operation

w=max (u, v); p=min (u, v);

**r** ----- (--, -

q1=1-u;

q2=1-v;

x1=1-w;

x2=min(q1,q2);

y1=1-p;

y2=max(q1-q2);

%Display Output

```
disp('Union of two fuzzy sets ');
```

```
disp(w);
disp('Intersection of two fuzzy sets ');
disp(p);
disp('Complement of first fuzzy set ');
disp(q1);
disp('Complement of second fuzzy set ');
disp('Complement of second fuzzy set ');
disp(q2);
disp('De-Morgan's Law');
disp('De-Morgan's Law');
disp('LHS');
disp('LHS');
disp(x1);
disp(x2);
disp(x1);
disp(LHS);
disp(LHS);
disp(y1);
disp('RHS');
```

disp(y2);

# Sample Input and output:

Enter the membership values of first fuzzy set [0.3 0.4] Enter the membership values of second fuzzy set [0.2 0.5] Union of two fuzzy sets 0.3000. 0.5000 Intersection of two fuzzy sets 0.3000 0.4000. Complement of first fuzzy set 0.7000 0.6000. Complement of second fuzzy set 0.8000 0.5000.

De –Morgan's	s Law
LHS	
0.7000	0.5000
RHS	
0.7000	0.5000
LHS	
0.8000	0.6000
RHS	
0.8000	0.6000

# **Result:**

Thus, the MATLAB program for implementation of De-Morgan's has been executed successfully and the output is verified.

# **Plotting Various Membership Functions**

#### Ex. No: 3

#### Date:

## Aim:

To write a program in MATLAB to plot triangular, trapezoidal and bell shaped membership functions.

## Algorithm:

- 1. Set the limits of x axis.
- 2. Calculate y using trimf() function with three parameters for triangular membership function.
- 3. Calculate y using trapmf() function with four parameters for trapezoidal membership function.
- 4. Calculate y using gbellmf() function with three parameters for bell shaped membership function.
- 5. Plot x and y values.

### **Program:**

%Triangular membership function

x=(0.0:1.0:10.0)';

y1= trimf(x, [1 3 5]);

subplot(311 )

plot(x,[y1]);

%Trapezoidal membership function

x=(0.0:1.0:10.0)';

y1= trapmf(x, [1 3 5 7]);

subplot(312)

plot(x, [y1] );

%Bell-shaped membership function

x=(0.0:0.2:10.0);

y1=gbellmf (x,[3 5 7]); subplot(313)

plot(x, [y1]);

# **Sample Input and Output:**



# **Result:**

Thus, the MATLAB program for plotting membership functions has been executed successfully and the output is verified.

# Using Fuzzy toolbox to model tips value

#### Ex. No: 4

#### Date:

#### Aim :

To use fuzzy toolbox to model tips value that is given after a dinner based on quality (not good, satisfying, good and delightful) and service (poor, average or good) and the tip value ranges from Rs. 10 to 100.

#### **Procedure:**

#### **INPUTS:**

Quality: { not good, satisfying, good, delightful }

Service :{ poor, average, good }

#### **OUTPUT:**

Tips: Tip\_value ranging from Rs. 10 to 100

#### Use Fuzzy Inference System (FIS) Editor and perform the following

- 1. Go to command window in Matlab and type fuzzy.
- 2. New Fuzzy Logic Designer window will be opened.
- 3. Give Input / Output Variable .
  - a. Go to Edit Window and click Add variable
  - b. As per our requirements create two input variables namely quality and service Quality: { not good, satisfying, good, delightful }
     Service :{ poor, average, good }
  - c. Similarly, one output variable as tip value ranges from 10 to 100.
- 4. The values for Quality and Service variables are selected for their respective ranges.
- 5. Quality:
  - a. Double click the Quality input variable .
  - b. New window will be opened and remove all the Membership Functions.
  - c. Go to Edit and Click Add MFs and select the 4 Parameters for Quality table.

d. Change the following fields as per the table given below .

Inputs : Quality $\rightarrow$ not good, satisfying, good, delightful							
MF1:	MF2:	MF3:	MF4:				
Range : [0 1 10]	Range : [0 1 10]	Range : [0 1 10]	Range : [0 1 10]				
Name : not good	Name : Cool	Name : Warm	Name : Hot				
Type : trapmf	Type : trimf	Type : trimf	Type : trapmf				
Parameter [0 10 30	Parameter [30 50 70]	Parameter [50 70 90]	Parameter [70 90 110 110]				
50]							

- 6. Similarly add the data to service and tips variables.
- 7. Go to Rules: Edit  $\rightarrow$  Rules
- 8. Add the Rules
- 9. Go to view  $\rightarrow$  Rules
- 10. Exit

# Sample Input and Output:

# Input and Output Variable in Edit Window

A MATLAB 7.6.0	R2008a)							_ 0 🛛
File Edit Debug Parallel Desktop Window Help								
: 🗋 🖨   🔏 🐂 🛙	1 7 C   🎝 🗹 🖻	Current Directory: C:\Docum	ents and Settings\Administrat	cor\My Documents\MATLAB 🛛 🔽	) 🛍			
Shortcuts 🛃 How to	Add 💽 What's New							
<b>Current Directory</b>	₩ C R X Wo	irkspace	Command Wind	ow				× 5 🗆 🕂
🖻 🖆 🖪 🔒 •			New to MATLAB	New to MATLAB? Watch this <u>Video</u> , see <u>Demos</u> , or read <u>Getting Started</u> .				×
All Files 🔺	Туре	Size Date Modified	>> fuzzy					
13-13.m	M-file	1 KB 5/9/11 11:32 AM	>>	File Edit View Cuality September 2015	Unit (mam	iled dani)	tip-value	
Command History 😐 🖬 🛪 🗙			×	FIC home:	ind.	EIC Tumo:	momdeni	
-* 5/9/11 11:07 AM*				ris Name: Ontil	ea	FIS Type.	mamdani	
⊟-% 5/9/11 1	1:32 AM%			And method	min	Current Variable		
				Or method	max 🖌	Tranz	Quality	
				Implication	min 💌	Type	input	
				Aggregation	max 👻	Range		
				Defuzzification	centroid 🗸	Help	Close	
				Ready				
				52				
start		-					in the second	OVR
🛃 start	🙏 7 MATLAB	🔹 🍩 Local Disk (C:)	Document1 - Microsof				🛛 🗘 🌾 🔤	🔓 🚺 11:44 AM

# Membership functions for Quality variable



## Membership functions for Service variable



# Membership functions for Tips variable



#### **Created rules**

🕹 Rule Editor: Untitled	
File Edit. View Options	
1. If (Quality is not_good) and (service is poor) then (tip-value is less) (1)     2. If (Quality is not_good) and (service is spoor) then (tip-value is less) (1)     3. If (Quality is adisfield) and (service is good) then (tip-value is medium) (1)     4. If (Quality is satisfield) and (service is good) then (tip-value is tigb) (1)     5. If (Quality is delicibus) and (service is good) then (tip-value is tigb) (1)     5. If (Quality is delicibus) and (service is good) then (tip-value is tigb) (1)     7. If (Quality is delicibus) and (service is good) then (tip-value is medium) (1)     7. If (Quality is delicibus) and (service is average) then (tip-value is medium) (1)     7. If (Quality is delicibus) and (service is average) then (tip-value is medium) (1)     7. If (Quality is delicibus) and (service is poor) then (tip-value is medium) (1)     7. If (Quality is delicibus) and (service is poor) then (tip-value is medium) (1)     7. If (Quality is delicibus) and (service is poor) then (tip-value is medium) (1)     7. If (Quality is delicibus) and (service is poor) then (tip-value is medium) (1)	^
Guality is service is	tip-value is
not good satisfied delicious none	less medium high none
not not	not
Connection Weight:	
and 1 Delete rule Add rul	Change rule
The rule is added	Help Close
🕂 Start 刘 7 MATLAB 🔹 🗢 Local Disk (C:) 📓 Document 1 - Microsof	😰 🌹 🔇 📴 🖏 🏠 🚮 11:45 AM

# Output



#### **Result:**

Thus, the fuzzy tool box is used to model tips value.

# **Implementation of Fuzzy Inference System**

#### Ex. No: 5

Date:

# Aim :

To implement a Fuzzy Inference System (FIS) for which the inputs, output and rules are given as below.

**INPUTS:** Temperature and Cloud Cover

**Temperature:** {Freeze, Cool, Warm and Hot}

Cloud Cover: {Sunny, Partly Cloud and Overcast}

# **OUTPUT:** Speed

**Speed :** {Fast and Slow}

#### **RULES:**

- If cloud cover is Sunny and temperature is warm, then drive Fast Sunny (Cover) and Warm (Temp) -> Fast (Speed)
- If cloud cover is cloudy and temperature is cool, then drive Slow Cloudy (Cover) and Cool (Temp) -> Slow (Speed)

#### Procedure

- 1. Go to command window in Matlab and type fuzzy.
- 2. Now, new Fuzzy Logic Designer window will be opened.
- 3. Input / Output Variable
  - a. Go to Edit Window and click Add variable.
  - b. As per our requirements create two input variables, Temperature and Cloud Cover.
  - c. Create one output variable, Speed.
- 4. Temperature:
  - a. Double click the Temperature input variable in Fuzzy Logic Designer window.
  - b. New window will be opened and remove all the Membership Functions.
  - c. Now, Go to Edit and Click Add MFs and select the 4 Parameters for Temperature Class.
  - d. Change the following fields as mentioned data in the given below table.

Inputs : Temperature $\rightarrow$ Freezing, Cool, Warm and Hot						
MF1:	MF2:	MF3:	MF4:			
Range : [0 110]	Range : [0 110]	Range : [0 110]	Range : [0 110]			
Name : Freezing	Name : Cool	Name : Warm	Name : Hot			
Type : trapmf	Type : trimf	Type : trimf	Type : trapmf			
Parameter [0 0 30 50]	Parameter [30 50 70]	Parameter [50 70 90]	Parameter [70 90 110 110]			

- 5. Similarly, add the data's to the Cloud Cover variables and Speed variables.
- 6. Cloud Cover:

Inputs : Cloud Cover $\rightarrow$ Sunny, Partly Cloud and Overcast						
MF1:	MF2:	MF3:				
Range : [0 100]	Range : [0 100]	Range : [0 100]				
Name : Sunny	Name : Partly Cloud	Name : Overcast				
Type : trapmf	Type : trimf	Type : trapmf				
Parameter [0 0 20 40]	Parameter [20 50 80]	Parameter [60 80 100]				

7. Speed:

Output : Speed $\rightarrow$ Slow and Fast				
MF1:	MF2:			
Range : [0 100]	Range : [0 100]			
Name : Slow	Name : Fast			
Type : trapmf	Type : trapmf			
Parameter [0 0 25 75]	Parameter [25 75 100 100]			
	_			

- 8. Goto Rules: Edit  $\rightarrow$  Rules
- 9. Add the Rules

Rule-1 : Sunny (Cover) and Warm (Temp) -> Fast (Speed)

Rule-2 : Cloudy (Cover) and Cool (Temp) -> Slow (Speed)

- 10. Go to view  $\rightarrow$  Rules
- 11. Exit.

# Sample Input and Output:

Membership functions for Temperature variable

📣 MATLAB R2017a			- 0 ×
HOME PLOTS APPS			Log In
New New Open Compare Data Workspace	New Variable Membership Fu Open Variable File Edit View Clear Worksp 5 FIS Variables	unction Editor: Untitled — Membership function plots <sup>edit</sup> coints:	pport 48
← → T Z . + C + Program Files + MATLAB + R2	2017a 🕨 bir 🔽 📈	freezing cool warm	hot v Q
Current Folder  Current Folder  Name *  Warker, bat  mex.ubit.pm  mex.ubat  mex.ubit.bare  Current folder  Current folder  Current folder  Name *  Current folder  Name *  Current folder  Name *  Current folder  Name *  Current folder  Cur	2 - cload_cover 3 - cload_cover 3 - cload_cover 4	temperature input to 10 to 20 to 20	
win64	14 - Display Range	[0 110] Help #	Close
Details ^	15 - Charles a		
Workspace 💿	17 - Changing parameter	for MF 2 to [30 50 70]	
Name A Value	Command Window fuzzy ft >>		0
IIII+ Ready			
Type here to search	J # C	🗟 🔒 🐋 🥠	名 へ ┥ 🎲 (də) ENG 11:18 🖏

# Membership functions for cloud over variable

📣 MATLAB R2017a					— <b>D</b> ×
HOME PLOTS APPS					🔄 🔁 🕐 Search Documentation 🛛 🔎 Log In
New New Open - Grower Import	👌 New Variabi 🛃 Open Variab	Membership Function Editor: Un File Edit View	itled —	D × pport	
Script • • Data Workspace	Clear Works; ABLE	FIS Variables	Membership function plots <sup>olot coints:</sup>	181 AB	*
🖛 🖚 💽 🔀 📙 🕨 C: 🕨 Program Files 🕨 MATLAB 🕨	R2017a 🕨 bi				<i>۹</i> •
Current Folder	🖲 🗾 Editor	temperature speed			⊙ ×
Name 🖛	exy7				
© worker.bat m mw.mpissec.bat messelb.pm messet.bat mes.pl % mes.pl	1 - 2 - 3 4 - 5 - 6 -	cloud <sub>e</sub> ever 0.5	20 30 40 50 60 70 80 incut variable "cloud over"	90 100	~*
mcc.bat	7 -	Current Variable	Current Membership Function (click on MF to selec	ct)	
i mbuild.bat	8 -		Hama		
- matlab.exe	10 -	Name cloud_cover	sunny		-
Icdata_utf8.xml	11 -	Type input	Type trapmf	~	
icdata.xml	12 -	Panne	Params [0 0 20 40]		
deploytool.bat	13 -	[0 100]			
win64	✓ 14 -	Display Range [0 100]	Help	Close	
Details	16 -				
Workspace	9 17 -	Changing parameter for MF 3 to [60 80	100 136]		
Name - Value					
	Comman	na window			٢
	fuzz fx >>				
IIII+ Ready					
					11:19
U Type here to search	Q	Ri 🦰 🗖 🛄 I	<u>≺</u>		유 <sup>4</sup> ^ ◀ 🎲 🕼 ENG 29-10-2019 🔞

A MATLAR P2017-					- a ×
HOME PLOTS APPS				8.04	Search Documentation
New New Open Compare Data Work	New Variable	Membership Function Editor: Untitles File Edit View	d	D X pport	×
Current Folder     Name *     worker.bat     mex.cubits.pm     mex.cubits.pm	TLAB ► R2017a ► bit            • R2017a ► bit            • Editor            • Editor            • I =            2 =            3            4 =            5 =            6 =	termenture speed cloud_over	20 30 40 50 60 70 80 odbut valable "seeed"	90 100	• p
me.Jat     mes.Jat     mould.bat     matbb.exe     icdata_xsd     icdata.xnl     icdeploytool.bat     wm64 Details	7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 -	Current Variable Name speed Type output Range [0 100] Display Range [0 100]	Current Membership Function (click on MF to select Name         slow           Type         trapmf           Params         [0 o 25 75]           Help	ct) Close	
Workspace	⑦ 16 − 1 17 −	Changing parameter for MF 2 to [25 75 100	100]		
Name A Value	Comman fuzz ft >>	d Window ¥			· · · · · · · · · · · · · · · · · · ·
IIII+ Ready					
Type here to search	Ū.	Hi 🤮 🗮 🏦 숙	1		R <sup>R</sup> ∧ ◀ *⁄⁄⁄ (↓») ENG 11:21 29-10-2019 😽

# Membership functions for speed variable

## **Created rules**

📣 MATLAB R2017a		- 5 X
HOME PLOTS APPS		Log In
Image: New Script         Image: Dec Script	New Variable	Rule Editor: Untitled
< 🔹 🔁 🔀 📙 🕨 C: 🕨 Program Files 🕨 MATLAB	▶ R2017a ▶ bi	٩ -
Current Folder	🕤 📝 Editor	• ×
Name *	exy7	
S worker, bat me, mills, pm messetup, pm messetup, pm messet, bat mes, pl mes, pl mes, bat mes, ba	▲ 1 - 2 - 3 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 13 - 13 - 13 - 13 - 13	If     and     Then       temperature is     speed is       freezing     isunty       ince     isseed is       ince     none       none     none       ince     ince       ince     ince
Workspace	· 17 -	The rule is added Help Close v
Name A Value	Comman fuzz ft >>	s Window 🔊
E O Type here to search	Q	부 😋 📄 🔒 🐋 🥠

# Output

HOME PLOTS APPS	
	🔎 Login
Rule Viewer: Untitled - X	
C Pind Files C Let Dopon Variab File Edit View Options port	
New Den Compare Import Save Script - Data Workspace 20 Clear Workspace 21.4 AB	
FILE VARIABLE speed = 62.3	Ā
	٩ •
	(e) X
workerbat A 1-	_
in my_mpiesec.bat 2 - 2	
mescuspm 3 4-0 110 0 100	
Demonstrate 5 -	_
mex.pi 6-	_
B mccbat 8 - 0 100	
a matabase	
	-
codaxsd 12 -	
S deploytool.bat	
Patale A 15 Input [55,21.36] Plot points: 101 Move left right down up	
Nucknare	
WORSpace 017 - 100 000	~
Name ~ Value Command Window	۲
fuzzy	
<i>E</i> >>	
and the second se	
	11:22
🛨 🕐 Type here to search 🛛 🚽 📮 🔁 💼 👫 🥆 🗳	29-10-2010
	23 10 2013
▲ MATLAB R2017a	- 0 X
MATLAB R2017a HOME R_OTS APPS	- 🗇 X 🔎 Log In
MATLAB R2017a HOME PLOTS AFPS C O Search Documentation PLOTS AFPS C O Search Documentation PLOTS AFPS C O Search Documentation	- 🗇 X
MATLAB R2017a HOME PLOTS ASPS	- D X P Log In
MATLAB R2017a       APPS       Image: Compare Import       Save Import	- 🗇 X 🔎 Log In
MATLAB R2017a       APPS       Image: Compare Import       Save Import       Rule Viewer: Unitited       —       ×       >	- 0 X
MATLAB R2017₂ HOME PLOTS APPS FIG Fies PLE Compare Data State Of Clar Works Sorpt Clar Works Clar Works Clar Works Sorpt Clar Works	- 0 ×
MATLAB R2017a      MATLAB R2017a      MOME     R_COTS     APPS     AP	- 0 ×
MATLAB R2017a      MOXE     PLOTS     APPS     Compare     Deta     Workspace     Compare     Deta     Workspace     Cerror     Compare     Deta     Workspace     Cerror     Ceror     Ceror     Ceror     Ceror	- • ×
MATLAB R2017a      MOME     PLOTS     ASPS     Rule Viewer: Untitled     PLOTS     Save     Open Varias     File     Current Folder     Workspace     Clar     Save     Compare Files     MATLAB P.2017a     T     Current Folder     Mame     Mame     Current Folder     Mame     Mame     Current Folder     Mame     Ma	- 0 ×
MATLAB R2017a       APPS       Search Documentation         HOME       P.OTS       APPS       Rule Views: Untitled       > P.OTS         New New Open       Compare       Open Varias       File       Edit View Options       Point         Script       Script       Script       Open Varias       File       Edit View Options       Point         Name *       Open Varias       File       Editor       Open Varias       File       Cloud_cover = 50       speed = 28.9         Name *       Open Varias       File       Editor       Open Varias       File       Cloud_cover = 50       speed = 28.9         Name *       Open Varias       File       Cover Varias       File       Open Varias       Open Varias <th< td=""><td>→ D × → Login → Q ⊙ ×</td></th<>	→ D × → Login → Q ⊙ ×
MATLAB R2017a       APPS       Search Documentation         HOME       PLOTS       APPS       Rule Viewer: Untitled       -       ×       >       >       >       >       APPS       Search Documentation         New New Open       Compare       mport       Save       Open Variats       File       Edit       View Options       >       port       AB         File       Variate       Clear Works       -       -       ×       port       AB         Current Folder        -       -       -       -       AB       -       -       AB         MartLAB P.2017a       -	- 0 X - Login - 0 X - 0 X - 0 X - 0 X - 0 X
MATLAB R2017a         HOME       RCTS       APPS       Rele Viewer: Untitled         New       New Variab       Rule Viewer: Untitled       -       ×         New       New Variab       Rule Viewer: Untitled       -       ×       pport         New       New       Open       Compare       Dia       Workspace       Clear Works         New New       Program Files       MATLAB + R2017a > bit       Immerature = 55       cloud_cover = 50       speed = 28.9         Current Folder       I       -       2       0       110       0       100         messflippm       -       2       0       110       0       100       100	- 0 X - 0 Login - 0 X - 0 X - 0 X
MATLAB R2017a       NOME       APPS       Read       Processor	- 0 ×
MATLAB R2017s         HOME       PLOTS       APPS         Image: Serie Find Files       Image: Serie Fi	→ Cog In → Log In → P © ×
MATLAB R2017a       PCOTS       APPS       Search Documentation         Image: Compare Import Save Scrept File       Image: Compare Import Save Data Workspace       Image: Comp	→ D × → Login → A ⊙ ×
MATLAB R20172 HOME PCOTS APPS Prof Find Fies Worksham New New Open Compare month New New Open Compare month N	- □ ×
MATLAB P2017₀          HOME       PCOTS       ASPS       Search Documentation         New       New New Open       Compare       Dials       Workspace       Clear Works         New       New       New New Open       Compare       Dials       Workspace       Clear Works         Name       Image: State       Im	- 0 X - 0 Login - 0 X - 0 X - 0 X - 0 X
MATLAB P2017s       HOME     PCTS     APPS       Image: Prod Files     Image: Prod Files     Image: Prod Files       Image: Prod Files     Image: Prod Files     Image: Prod Files       Image: Prod Files     Image: Prod Files     Image: Prod Files       Image: Prod Files     Image: Prod Files     Image: Prod Files       Image: Prod Files     Image: Prod Files     Image: Prod Files       Image: Prod Files     Image: Prod Files     Image: Prod Files       Image: Prod Files     Image: Prod Files     Image: Prod Files       Image: Prod Files     Image: Prod Files     Image: Prod Files       Image: Prod Files     Image: Prod Files     Image: Prod Files       Image: Prod Files     Image: Prod Files     Image: Prod Files       Image: Prod Files     Image: Prod Files     Image: Prod Files       Image: Prod Files     Image: Prod Files     Image: Prod Files       Image: Prod Files     Image: Prod Files     Image: Prod Files       Image: Prod Files     Image: Prod Files     Image: Prod Files       Image: Prod Files     Image: Prod Files     Image: Prod Files       Image: Prod Files     Image: Prod Files     Image: Prod Files       Image: Prod Files     Image: Prod Files     Image: Prod Files       Image: Prod Files     Image: Prod Files       Image	- 0 X - 0 X - 0 Cogin - 0 X - 0 X - 0 X - 0 X
MATLAB R2017       PCTS       APPS       Search Documentation         INME       Image: Save       Open Variantic       Image: Save       Search Documentation         Script       Image: Save       Open Variantic       Image: Save       Closer Works       Image: Save       Search Documentation         Image: Script       Image: Save       Open Variantic       Image: Save       Closer Vorks       Image: Save       Image: Save       Save       Image:	→ D × Log In → D × Cog In → P Log In →
NMTLAB R2017a         HOHE       FLOTS       APPS       Event Variage         New New Open       Compase       Event Variage       Fle       Edit       View Options       port         Storet       Variage       Clear Works       Versepace       Clear Works       Edits       View Options       port         Current Folder       Versepace       Clear Works       Versepace       Clear Works       Versepace       Edits       View Options       port         Now Ker, bott       Immessaturg pm	- C X P LogIn - O X - O X
NMLAB R207x         HOHE       RCOTS       APS         Percent Field Fies       Percent Visited         Percent Field Fies       Percent Field Fies         Percent Field Fies       Percent Field Field Fies         Percent Field Fies       Percent Field	→ D × → Login → A • A • A
NULLAR REDIT         NONE       PLOTS       APPS       Item Varia         New Open       Compare       Did Workspace       Decen Workspace         New Open       Compare       Did Workspace       Did Workspace       Did Workspace       Did Workspace         New Open       Compare       Did Workspace       Did Workspace <th< td=""><td>- 0 × - 0 × -</td></th<>	- 0 × -
NATURE ROTS         AVS         Search Documentation           INDE         FOR Files         Import Variable         File         Edit Viewer Untitled         -         >         port         Port         Search Documentation           Item         Import Files         Import Variable         Import Variable         Import         Import         Import         Import Variable	→ Cogin → Login → Cogin →
MATLAB R2017s           HORE         AVYS         Earth Decumentation           INDE         COTS         AVYS         Earth Decumentation           Service         Dear Workspace         Dear Workspace         Dear Workspace         Dear Workspace           Current None         Dear Workspace         Editor         Provide Later         Provi	→ D × Log In → P
MATLAB R201%         Nove         Adds         Adds         Adds         Adds         Concentration           INDIAL SECURITY         India Mathematication         Indion	→ D × → LogIn → A ⊕ X ⊕ X ⊕ X ⊕ X ⊕ X ⊕ X ⊕ X ⊕ X
NATURE RED/s         PLot APP         APP         Plot APP	
MATLAB RD/3           NOR         POR File         If we have varies in the varies in th	Cogin → Login → Cogin → Co
MATLAB 2017         NOT       APS         Note       For Frides         Note       Dear Writeplace         Note       Note         Note       Dear Writeplace         Note       Note	→ Constant of the second seco
MATLAN 2017          NOTE       APS         INCLUS 2017       INCLUS 2017         NOTE       Comparing the post of the post o	

# **Result:**

Thus a Fuzzy Inference System is implemented for temperature, cloud cover and speed using the given rules.

# **Simple Fuzzy Set Operations**

#### Ex. No: 6 Date:

#### Aim:

To write a MATLAB program to find algebraic sum, algebraic subtraction, algebraic product, bounded sum, bounded subtraction and bounded product of two fuzzy sets.

#### Algorithm:

- 1. Read the values of the two fuzzy sets.
- 2. Perform the algebraic sum operation by,

A + B = (a + b) - (a \* b)

3. Perform the algebraic subtraction operation by,

A - B = (a + b) where b'= 1- b

4. Perform the algebraic product operation by,

A \* B = (a \* b)

5. Perform the bounded sum operation by,

 $A \oplus B = \min [1, (a + b)]$ 

- 6. Perform bounded subtraction operation by,  $A \ominus B = \max[0, (a - b)]$
- 7. Perform bounded product operation by,

 $A \odot B = \max [0, (a + b - 1)]$ 

8. Display the results

#### **Program:**

- a= input('Enter the fuzzy set a')
- b= input('Enter the fuzzy set b')

c = a + b

d= a \* b

as = c - d

e = 1 - bad = a + ef=a-bbs=min(1, c)bd= max (0, f) g = c - 1bp= max (0,g) disp('The algebraic sum') disp(as) disp('The algebraic difference') disp(ad) disp('The algebraic product') disp(d) disp('The bounded sum') disp(bs) disp('The bounded difference') disp (bd) disp('The bounded product') disp(bp)

# **Output:**

Enter fuzzy set a [1 0.5] Enter fuzzy set b [0.4 0.2] The algebraic sum [1.0000 0.6000] The algebraic difference [1 0.9000] The algebraic product [0.4000 0.1000] The bounded sum [1.0000 0.7000] The bounded difference [0.6000 0.3000] The bounded product [0.4000 0]

## **Result:**

Thus, a program to perform simple fuzzy set operations has been executed and successfully verified.

# Using Hopfield network with no self connection

#### Ex. No: 7 Date:

#### Aim:

To write a MATLAB program to store the vector  $(1\ 1\ 1\ 0)$  and to find the weight matrix with no self connection using a discrete hopfield net with mistake in first and second component of vector that is  $(0\ 0\ 1\ 0)$ .

#### Algorithm:

- 1. Make the initial activations of the net equal to given binary pattern  $x = (1 \ 1 \ 1 \ 0)$ .
- 2. Let  $tx = (0 \ 0 \ 1 \ 0)$ .
- 3. Initialize weight matrix using the formula

 $w=(2^{*}x^{-1})^{*}(2^{*}x^{-1})$ 

- 4. Set the diagonal values of weight matrix as 0.
- 5. Enter updation vector up=[4 2 1 3].
- 6. Perform the following for each i

net input, Yin(up(i))=tx(up(i))+y\*w(1:4,up(i))

- 7. Apply activation function to calculate output.
- 8. Test the network for convergence.

#### **Program:**

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
if y==x
```

```
disp('Convergence has been obtained');
disp('The converged output');
disp(y);
con=0;
```

end end

# **Output:**

up=4 2 1 3

Convergence has been obtained

The Converged Output

 $1 \ 1 \ 1 \ 0$ 

#### **Result:**

Thus the MATLAB program for using Hopfield network with no self connection has been successfully executed and the output is verified.

# Generation of ANDNOT function using McCulloch-Pitts neural

net

#### Ex. No: 8 Date:

## Aim:

To write a MATLAB program to generate ANDNOT function using McCulloch-Pitts neural net.

# Algorithm:

- 1. Initialize weights w1,w2 and threshold theta
- 2. Assign input values
  - x1=[0 0 1 1]
  - x2=[0 1 0 1]
- 3. Assign output  $Z = [0 \ 0 \ 1 \ 0]$
- 4. Initialize y = [0 0 0 0]
- 5. Repeat the following for each input
  - i) Zin = x1\*w1+x2\*w2
  - ii) If Zin > theta set y as 1 else 0
- 6. If y is not equal to Z update weights and repeat step 5
- 7. Display weights and threshold value

#### **Program:**

clear;

clc;

disp('Enter the weight');

```
w1=input('weight w1=');
```

w2=input('weight w2=');

disp('Enter threshold value');

theta=input('theta=');

y=[0000];

```
x1=[0 0 1 1];
x2=[0 1 0 1];
Z = [0 \ 0 \ 1 \ 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);

#### Sample Input and Output:

Enter the weights Weight w1=1 Weight w2=1

Enter threshold value Theta=1 Output of net= 0 1 1 1 Net is not learning

Enter another set of weights and threshold value Weight w1=1 Weight w2= -1 Theta=1

Output of net= 0 0 1 0 McCulloch Pitts Net for ANDNOT function Weights of neuron 1 -1 Threshold value=1

#### **Result:**

Thus, a MATLAB program to generate ANDNOT function using McCulloch-Pitts neural net has been successfully executed and the output is verified.

# Finding weight matrix and bias of HebbNet to classify two dimensional input patterns

Ex. No: 9 Date:

#### Aim:

To write a MATLAB program to find the weight matrix and bias of Hebbnet in bipolar to classify two dimensional input patterns with their targets given below.

"' indicates a '+' and	l '.' Indicates a '-
****	****
*	*
****	****
*	*
****	*

#### Algorithm:

- 1. Create a single layer neural network with 25 neuron.
- 2. Set the initial weight and bias to zero.
- 3. Calculate the weights using

w<sub>i</sub>(new)=w<sub>i</sub>(old)+x<sub>i</sub>\*t

and bias using

b(new)=b(old)+t(i)

4. Display the final weight matrix and bias.

#### **Program:**

% Hebb Net to classify 2d input patterns

clear;

clc;

%Input Pattern

```
t=[1 -1];
b=0;
for i=1:2
w=w+x(I,1:25)*t(i);
b=b+t(i);
end
disp(' Weight matrix:');
disp(w);
disp('Final Bias:');
disp(b);
```

# **Output:**

Weight matrix

**Final Bias** 

0

#### **Result:**

Thus a MATLAB program to find the weight matrix and bias to classify two dimensional input patterns in bipolar using Hebb Net have been executed and verified successfully.

# Perceptron net for AND function with bipolar inputs and targets

Ex. No: 10 Date:

#### Aim:

To write a MATLAB program to implement AND function with bipolar input and output using Perceptron net.

#### Algorithm:

- 1. Initialize weight and bias to 0
- 2. Accept learning rate, alpha and threshold, theta
- 3. For each input calculate yin = b+x(1)\*w(1)+x(2)\*w(2)
- 4. Apply activation function
- 5. If calculated output  $\neq$  target output
  - i) update weight and bias
  - ii) Go to step 3
- 6. Display final weight matrix and bias value

#### **Program**:

% Perceptron for AND function 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

```
epoch=epoch+1;
```

end

disp('Perceptron for AND Function');

```
disp('Final Weight Matrix');
```

disp(w);

disp('Final Bias');

disp(b);

# **Sample Input and Output:**

Enter Learning rate = 1

Enter Threshold Value =0.5

Perceptron for AND Function

Final Weight Matrix

1 1

**Final Bias** 

-1

#### **Result:**

Thus, a MATLAB program for Perception net for an AND function with bipolar inputs have been written and verified successfully.

# Finding weight matrix of hetero associative neural net for mapping of vectors

#### Ex. No: 11 Date:

## Aim:

To write a MATLAB program to calculate the weights using Hetero-associative neural net for mapping of vectors.

<b>S1</b>	<b>S2</b>	<b>S3</b>	<b>S4</b>	t1	t2
1	1	0	0	1	0
1	0	1	0	1	0
1	1	1	0	0	1
0	1	1	0	0	1

# Algorithm:

- 1. Enter input and output vector x and t
- 2. Initialize weight matrix.
- 3. Update weight matrix by using the formula

w<sub>i</sub>(new)=w<sub>i</sub>(old)+x<sub>i</sub>\*t

4. Display the calculated weight.

# **Program:**

%Hetero-associative neural net for mapping input vectors to output vectors clear;

clc;

```
x=[1 1 0 0; 1 0 1 0; 1 1 1 0; 0 1 1 0];
t=[1 0; 1 0; 0 1; 0 1];
w=zeros(4,2);
for i=1:4
w=w+x(i,1:4)'*t(i,1:2);
end
disp(' Weight matrix:');
disp(w);
```

# **Output:**

Weight matrix

- 2 1 1 2 1 2
- 0 0

# **Result:**

Thus a MATLAB program to calculate the weight matrix using hetero associative neural net for mapping of vectors has been executed and verified successfully.

# Generation of XOR function using back propagation algorithm

#### Ex. No: 12 Date:

#### Aim:

To write a MATLAB program to train and test the back propagation neural network for the generation of XOR function.

#### Algorithm:

- 1. Enter the input vector  $x=[0\ 0\ 1\ 1\ ;\ 0\ 1\ 0\ 1]$  and target vector  $y=[0\ 1\ 1\ 0]$ .
- 2. Train the network by using the function newff().
- 3. Set the epoch and learning rate value.
- 4. Test the network by using the trained network.
- 5. Display the result.

#### **Program:**

## Function to Training the Back Propagation neural network:

```
function[net]=trainBPN(x,y)
```

```
[n,i]=size(x);
```

[m,o]=size(y);

```
net = newff(minmax(x),[i,10,m],{'tansig','tansig','purelin'},'trainlm')
```

```
net .trainparam.epoch =50;
```

net.trainparam.lr=0.2;

```
net=train(net,x,y);
```

r=sim(net,x);

return;

# Function for testing the Back Propagation neural network:

```
function[v]=testBPN(x,net)
```

v=sim(net,x);

return;

# **Output:**

#### Matlab command for executing the function:

```
>>x=[0\ 0\ 1\ 1\ ;\ 0\ 1\ 0\ 1]
```

```
>>y=[0 1 1 0]
```

## Calling the function to train the network:

>>[net]=trainBPN(x,y)

#### Calling the function to test the network:

```
>>testBPN(x,net)
```

ans=

-0.0000 1.0000 1.0000 -0.0000

#### **Result:**

Thus a MATLAB program to train and test the back propagation neural network for the generation of XOR function has been executed successfully and the output is verified.