Index

Experiment No. 1:
Title: Generate AND, NOT function using McCulloch-Pitts neural net by MATLAB program.

Aim: Write down briefly about the importance/ applicability of McCulloch-Pitts neural net.

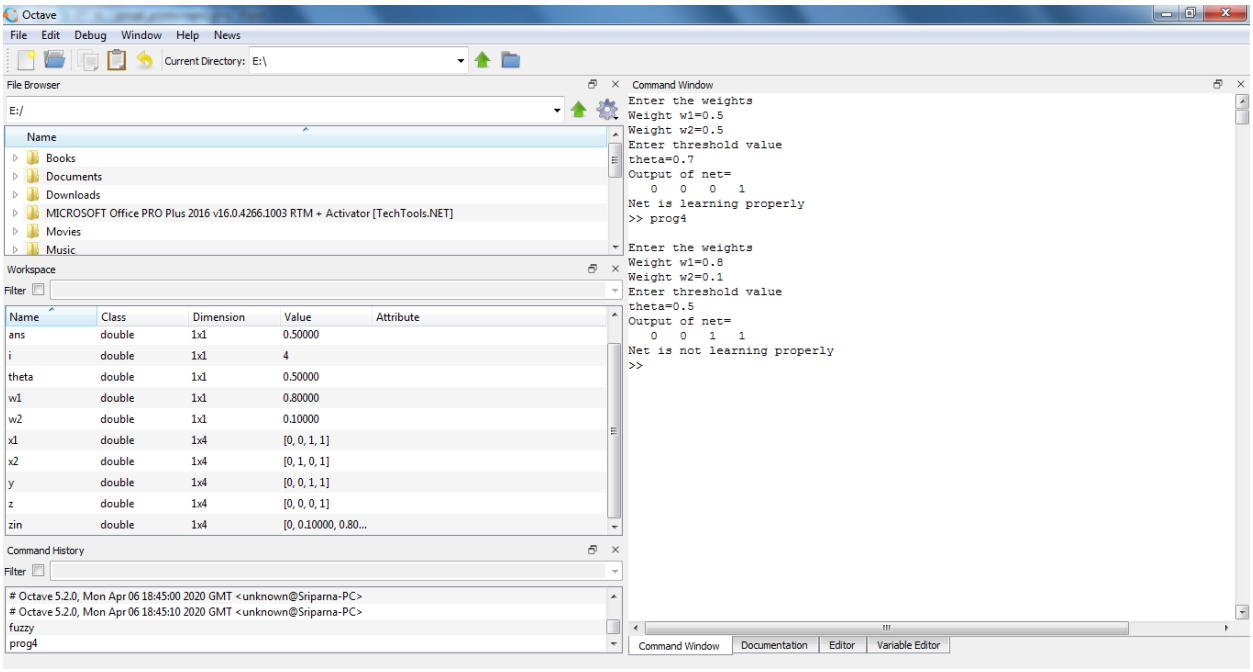
Theory: Write it as taught in the class.

Matlab Code:

```
%AND function using McCulloch-Pitts Neuron
clc
clear all
%close all
%Getting weights & Threshold value
disp('Enter the weights');
w1 = input('Weight w1=');
w2 = input('Weight w2=');
disp('Enter the Threshold value');
theta = input('Theta=');

y = [0 0 0 0];%initialize to avoid garbage value
x1 = [0 0 1 1];%Input1
x2 = [0 1 0 1];%input2
z = [0 0 0 1];%ideal output

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
    disp('Net is Learning properly');
else
    disp('Net is not Learning properly');
end
```

```
%NOT function using McCulloch-Pitts Neuron
clc
clear all
%close all
%Getting weights & Threshold value
disp('Enter the weights');
w1 = input('Weight w1=');
disp('Enter the Threshold value');
theta = input('Theta=');

y = [0 0];%initialize to avoid garbage value
x1 = [0 1];%Input1
z = [1 0];%ideal output

zin = x1*w1;
for i=1:2
    if zin(i)>=theta
        y(i)=0;
    else y(i)=1;
    end
end
disp('Output of net=');
disp(y);
if y==z
    disp('Net is Learning properly');
else
    disp('Net is not Learning properly');
end
```

Results:



Conclusion:

Reference:

https://www.youtube.com/watch?v=6XhSJbfT1pk

Experiment No. 2:
Title: Write a MATLAB program for Perceptron net for an AND function with bipolar inputs and targets.

Aim: Write down briefly about the importance/ applicability of Perceptron net.

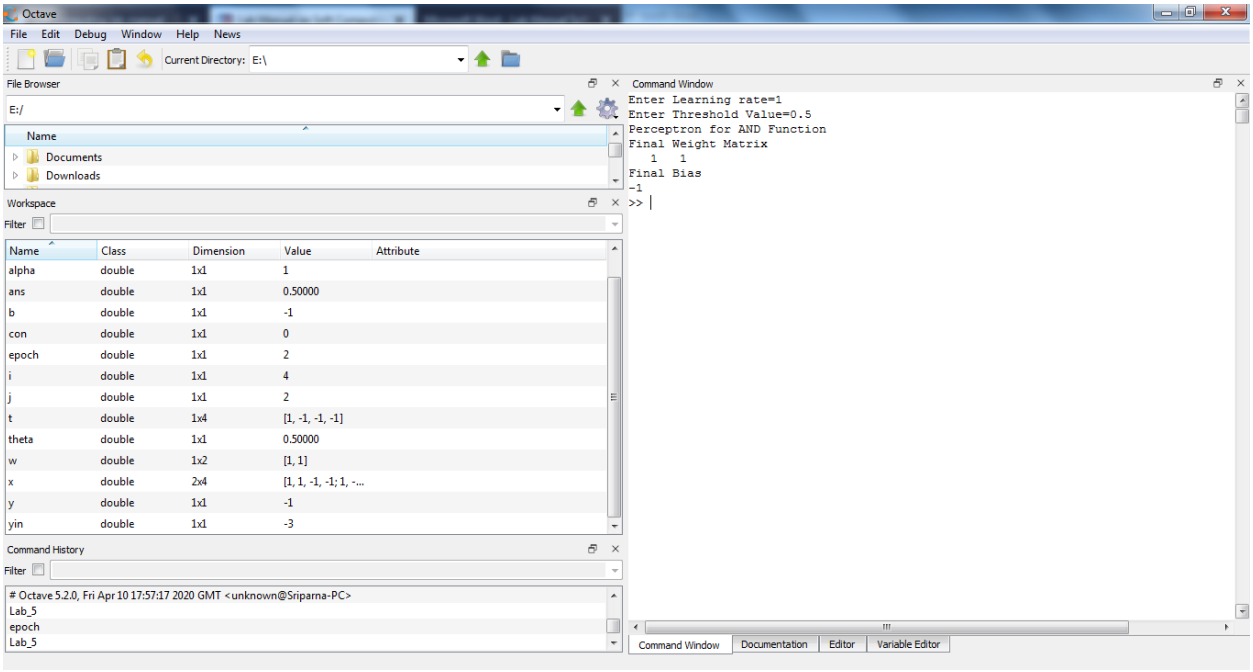
Theory: Write it as taught in the class.

Matlab Code:

```
% 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
  end
  epoch=epoch+1;
```

```
end

disp('Perceptron for AND Function');
disp('Final Weight Matrix');
disp(w);
disp('Final Bias');
disp(b);
```

Results:



Conclusion:

Reference:

https://www.youtube.com/watch?v=VRcixOuG-TU

Experiment No. 3
Title: Write a MATLAB Program on Back propagation neural network.

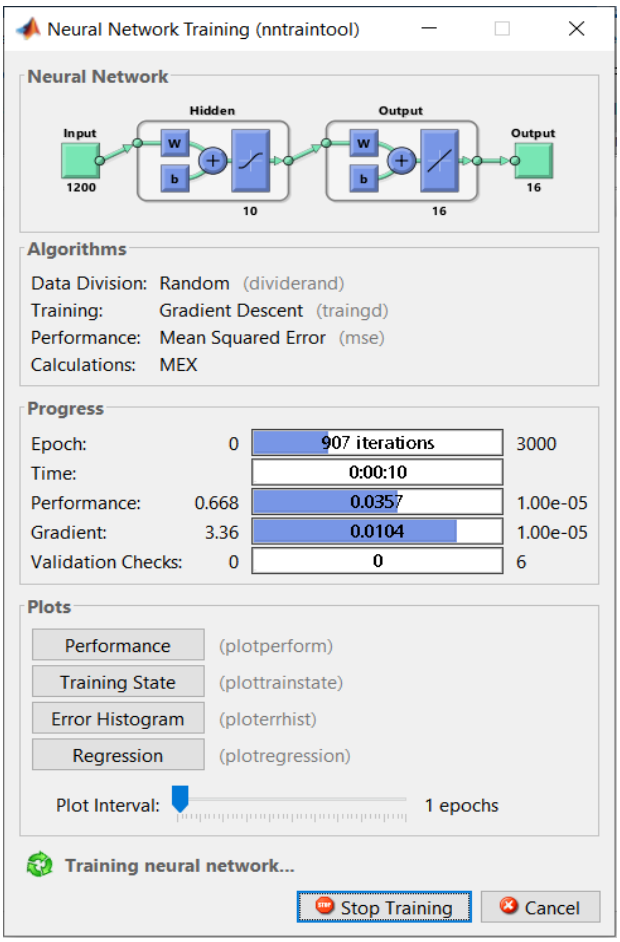
Aim: Write down briefly about the importance/ applicability of back propagation neural network.

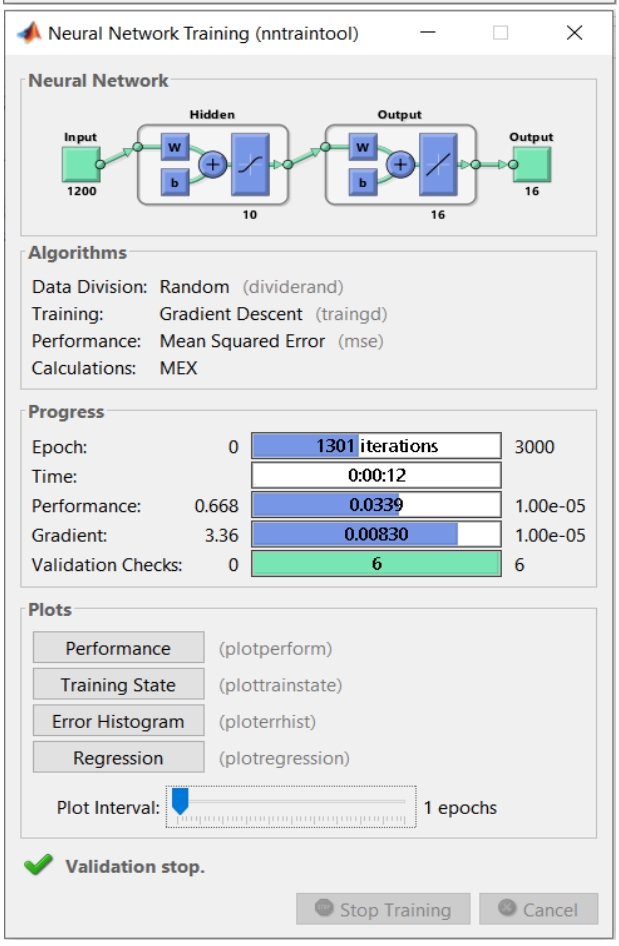
Theory: Write it as taught in the class.

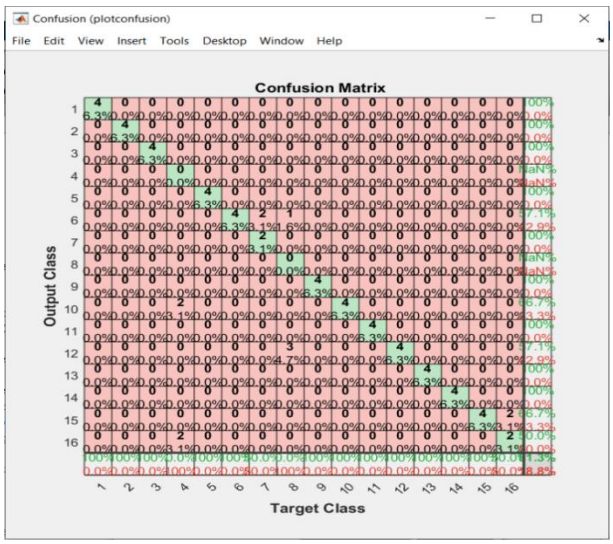
Matlab Code:

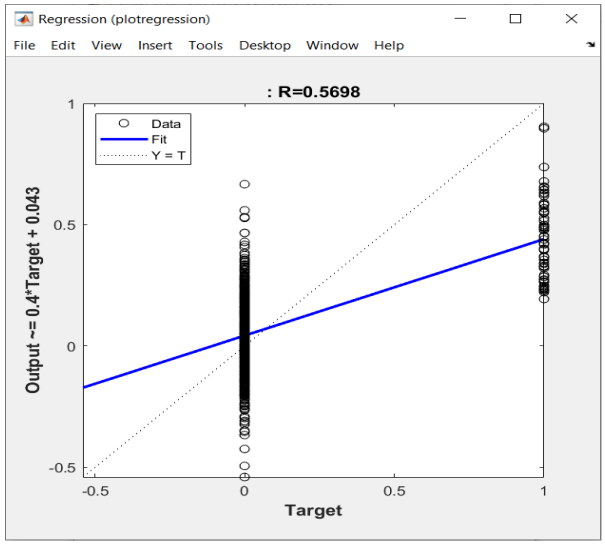
```
clear all;
close all;
clc;

input=xlsread('fv.xlsx');
target=xlsread('target.xlsx');
nntic=tic;
hiddenLayerSize = 10;
net = feedforwardnet(hiddenLayerSize,'traingd');
net.trainParam.lr = 0.05; %its not mandatory to give this value,
automatic value will be taken
net.trainParam.epochs = 3000; %its not mandatory to give this
value, automatic value will be taken
net.trainParam.goal = 1e-5; %its not mandatory to give this
value, automatic value will be taken
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
net=init(net);
[net,tr] = train(net,input,target); %training
output = sim(net,input); %simulation
figure,plotconfusion(target,output)
plotregression(target,output); %regresson plot
error = gsubtract(target,output);
performance = mse(error); %mean square error
figure, plotroc(target,output)
nntime=toc(nntic);

unknown=xlsread('unknown.xlsx');%let it is the unknown feature
value
y = net(unknown);%results obtained for all classes

% initlay is a network initialization function that initializes
each layer i according to
% its own initialization function net.layers{i}.initFcn.
% The weights and biases of each layer i are initialized
according to net.layers{i}.initFcn.
```

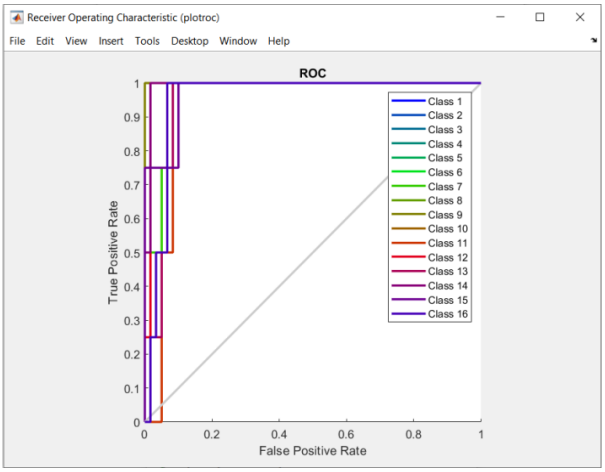
Results:

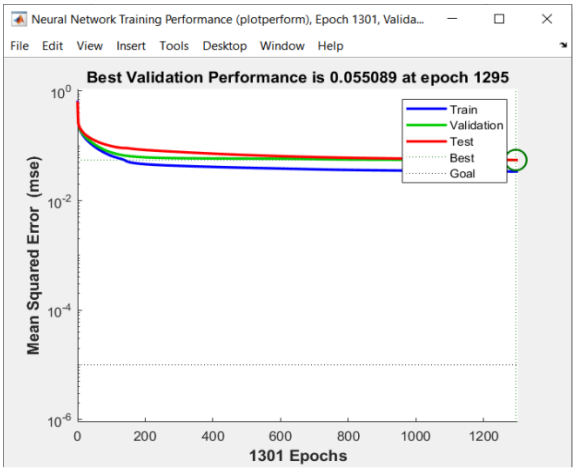

Neural Network Training (nntraintool)
Neural Network
Input
1200
Hidden
W
b
10
Output
W
b
16
Output
16
Algorithms
Data Division: Random (dividerand)
Training: Gradient Descent (traingd)
Performance: Mean Squared Error (mse)
Calculations: MEX
Progress
Epoch: 0 907 iterations 3000
Time: 0:00:10
Performance: 0.668 0.0357 1.00e-05
Gradient: 3.36 0.0104 1.00e-05
Validation Checks: 0 0 6
Plots
Performance (plotperform)
Training State (plottrainstate)
Error Histogram (ploterrhist)
Regression (plotregression)
Plot Interval: 1 epochs
Training neural network...
Stop Training
Cancel

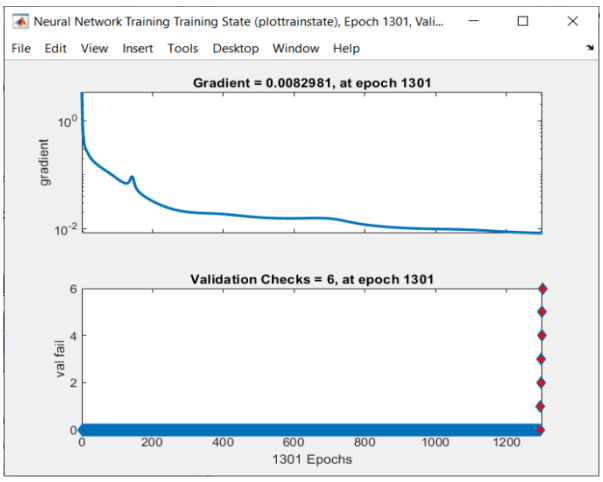

Neural Network Training (nntraintool)
Neural Network
Input
1200
Hidden
W
b
10
Output
W
b
16
Output
16
Algorithms
Data Division: Random (dividerand)
Training: Gradient Descent (traingd)
Performance: Mean Squared Error (mse)
Calculations: MEX
Progress
Epoch: 0 1301 iterations 3000
Time: 0:00:12
Performance: 0.668 0.0339 1.00e-05
Gradient: 3.36 0.00830 1.00e-05
Validation Checks: 0 6 6
Plots
Performance (plotperform)
Training State (plottrainstate)
Error Histogram (ploterrhist)
Regression (plotregression)
Plot Interval: 1 epochs
Validation stop.
Stop Training
Cancel

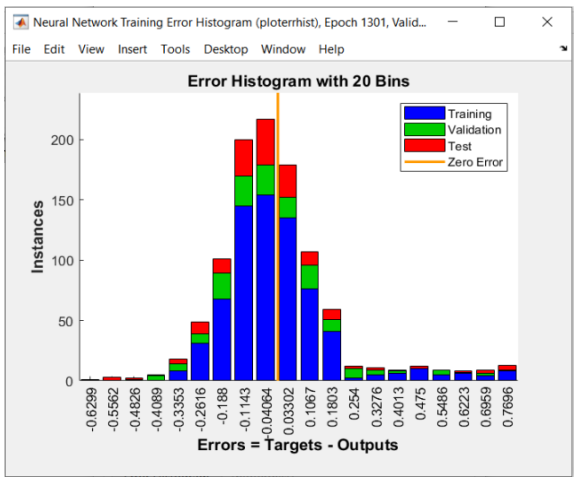

Confusion (plotconfusion)
File Edit View Insert Tools Desktop Window Help
Confusion Matrix
Output Class
Target Class

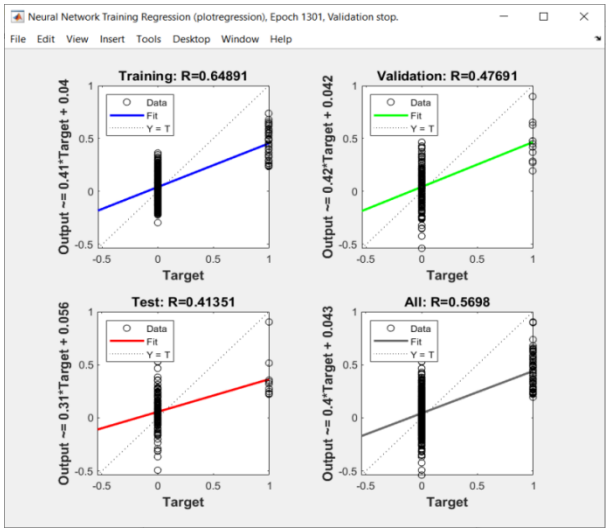

Regression (plotregression)
File Edit View Insert Tools Desktop Window Help
: R=0.5698
Data
Fit
Y = T
Output ~= 0.4*Target + 0.043
Target


Receiver Operating Characteristic (plotroc)
File Edit View Insert Tools Desktop Window Help
ROC
True Positive Rate
False Positive Rate
Class 1
Class 2
Class 3
Class 4
Class 5
Class 6
Class 7
Class 8
Class 9
Class 10
Class 11
Class 12
Class 13
Class 14
Class 15
Class 16


Neural Network Training Performance (plotperform), Epoch 1301, Valida...
File Edit View Insert Tools Desktop Window Help
Best Validation Performance is 0.055089 at epoch 1295
Train
Validation
Test
Best
Goal
Mean Squared Error (mse)
1301 Epochs







Conclusion:

Reference:

https://www.youtube.com/watch?v=PEmSbdC4y_Y&list=PLsEIbHOtypISN0ZXjZ7Uhp0YwCToyrOLM

Experiment No. 4:
Title: Write a program in MATLAB to plot various membership functions.

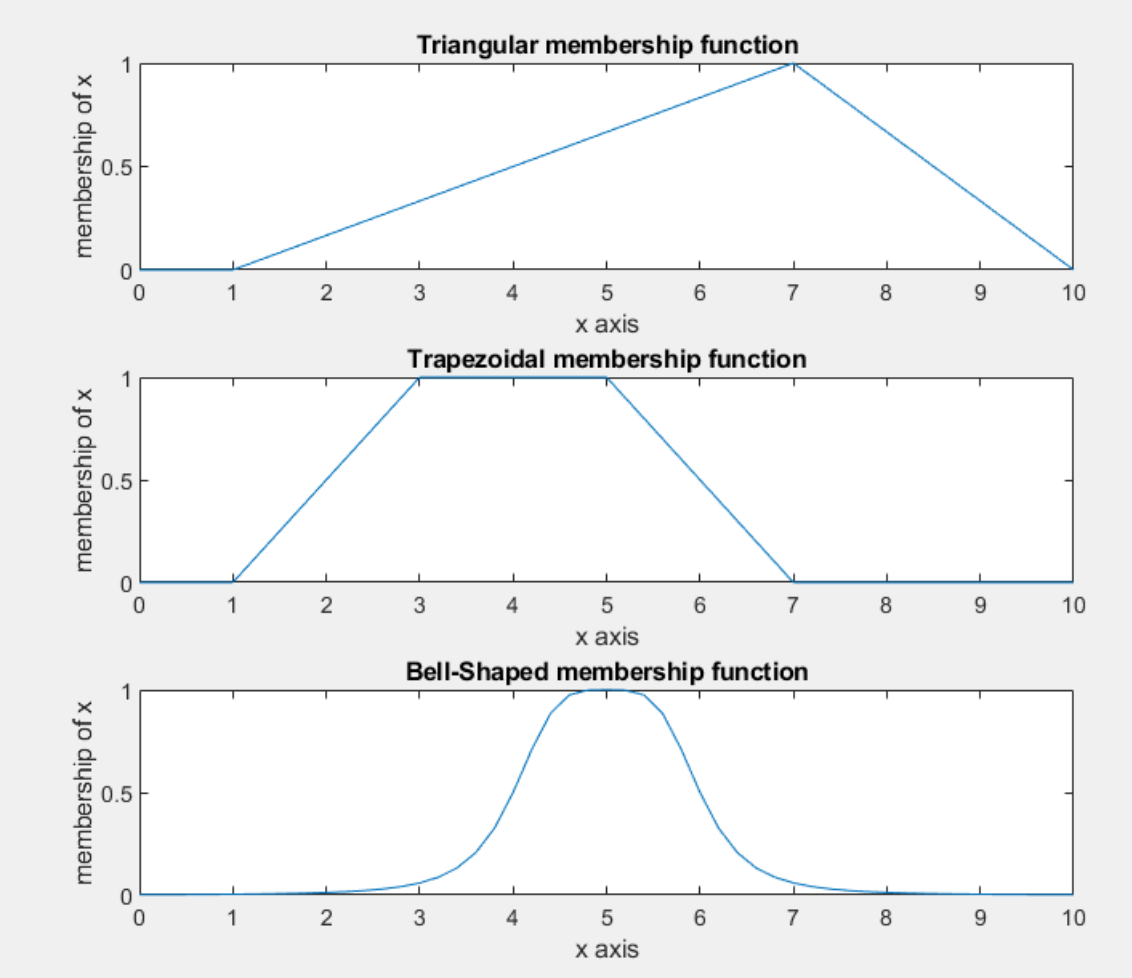
Aim: Write down briefly about importance/ applicability of membership functions.

Theory: You are supposed to explain about different membership functions along with their equations.

Matlab Code:

```
clc
clear all
close all

%Triangular Membership Function
x=(0:1:10)';
y1=trimf(x, [1 7 10]);
subplot(3,1,1)
plot(x,[y1]);
xlabel('x axis')
ylabel('membership of x')
title('Triangular membership function')

%Trapezoidal Membership Function
x=(0:1:10)';
y1=trapmf(x, [1 3 5 7]);
subplot(3,1,2)
plot(x,[y1]);
xlabel('x axis')
ylabel('membership of x')
title('Trapezoidal membership function')

%Bell-Shaped Membership Function
x=(0:0.2:10)';
y1=gbellmf(x, [1 2 5]);
subplot(3,1,3)
plot(x,[y1]);
xlabel('x axis')
ylabel('membership of x')
title('Bell-Shaped membership function')
```

Results:



Conclusion:

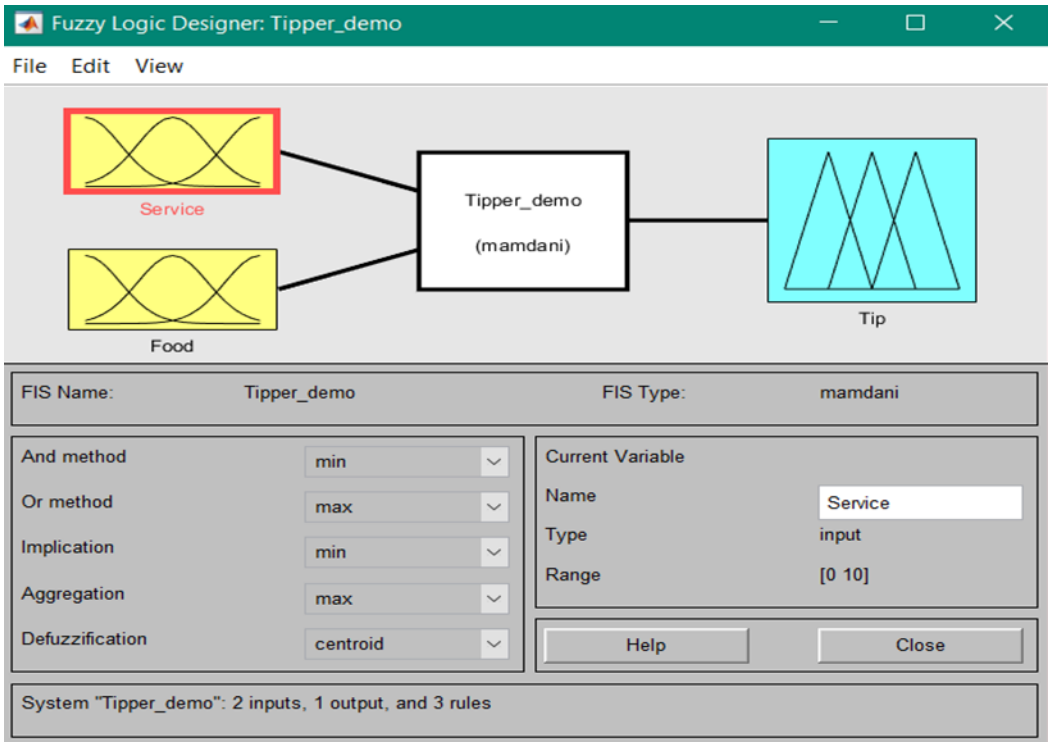
Reference:

https://www.youtube.com/watch?v=whIR88tAANE&list=PLJ5C_6qdAvBFqAYS0P9INAogIMklG8E-9&index=3

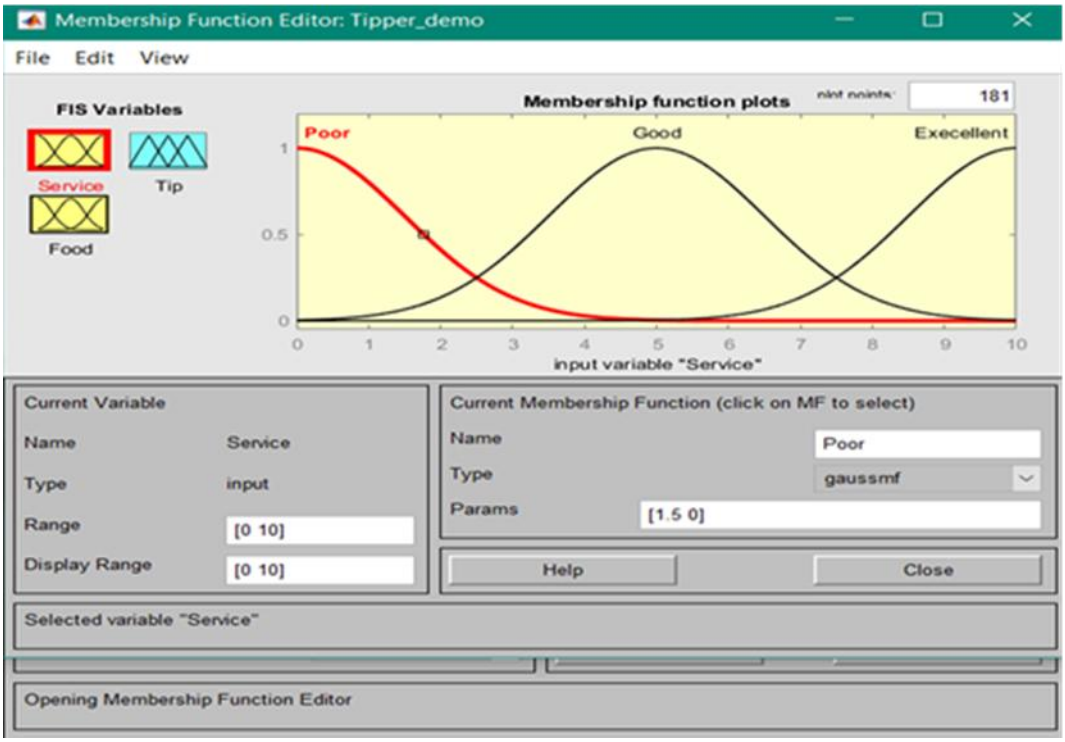
Experiment No. 5:
Title: Use Fuzzy toolbox to model tip value based on service and food quality.

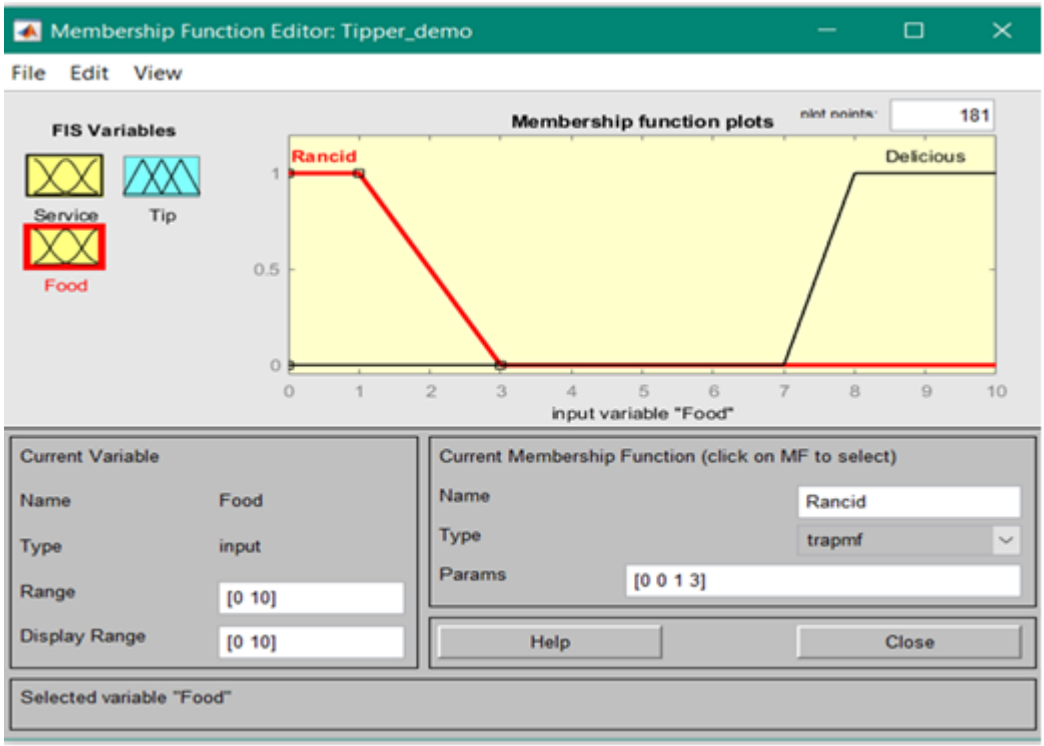
Aim: Write down briefly about the importance/ applicability of fuzzy toolbox.

Procedure and Results: You are required to explain how fuzzy toolbox can be implemented.

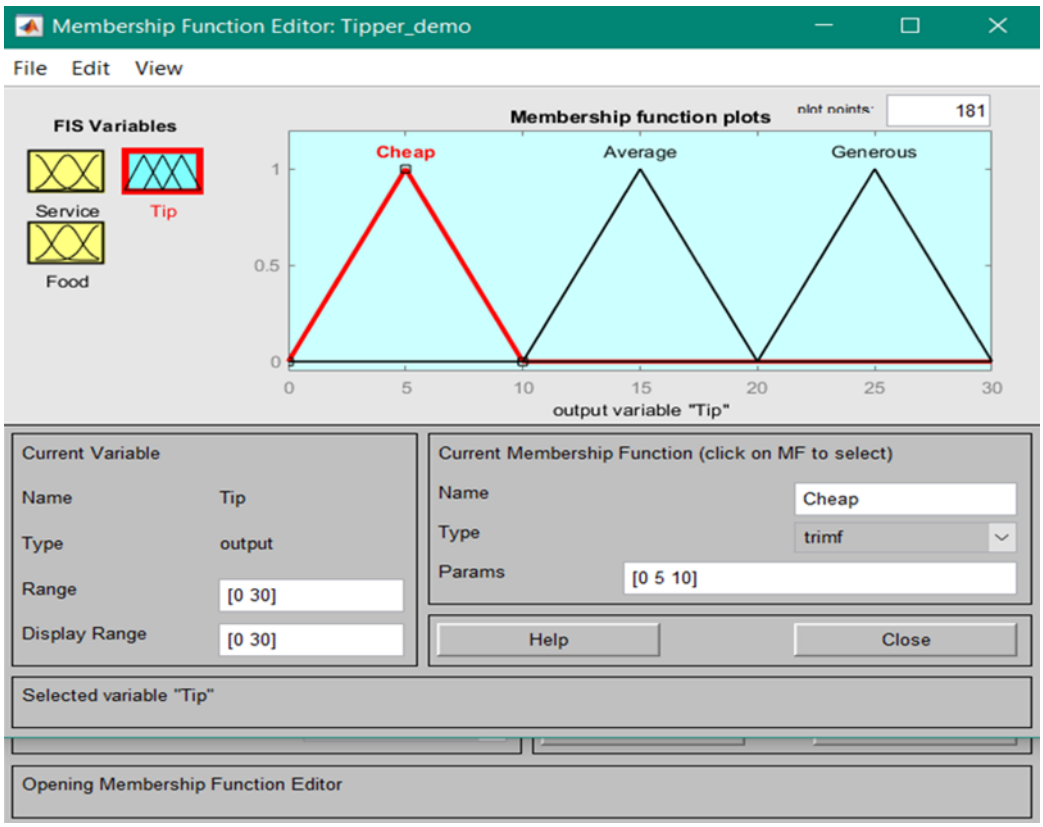


Input linguistic variables: Service (Poor, Good, Excellent) & Food (Rancid, Delicious)



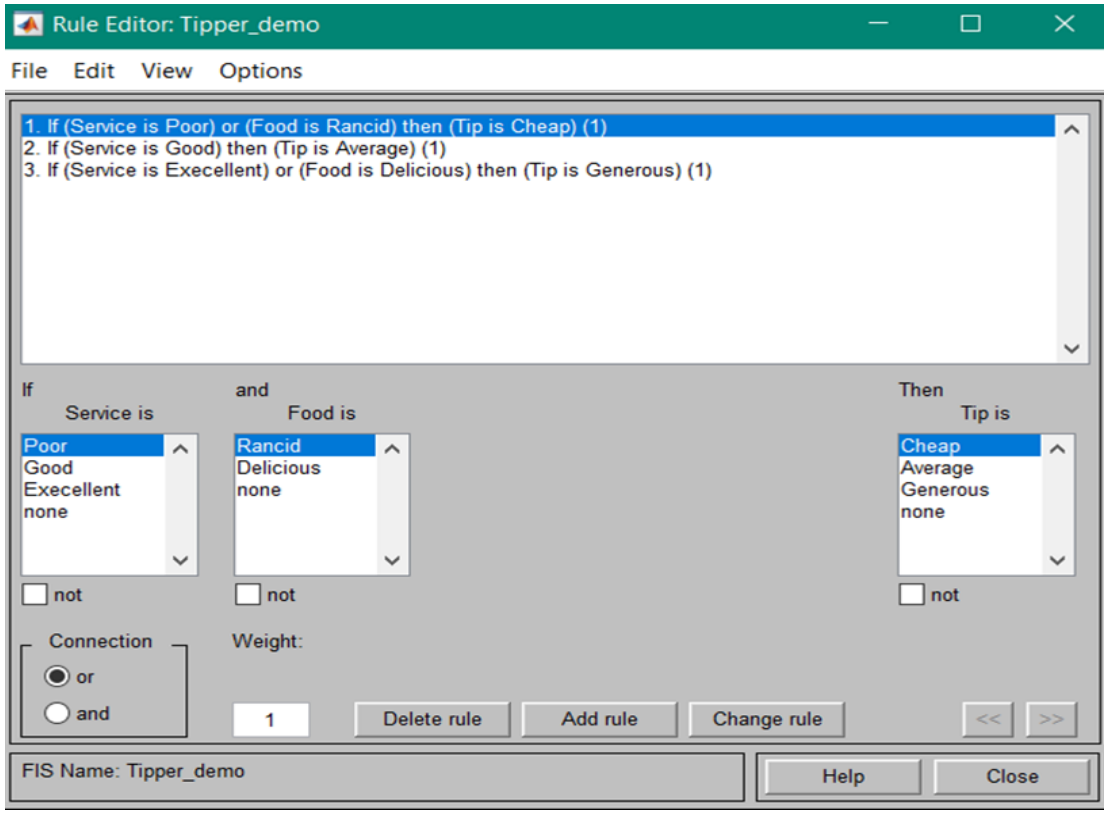


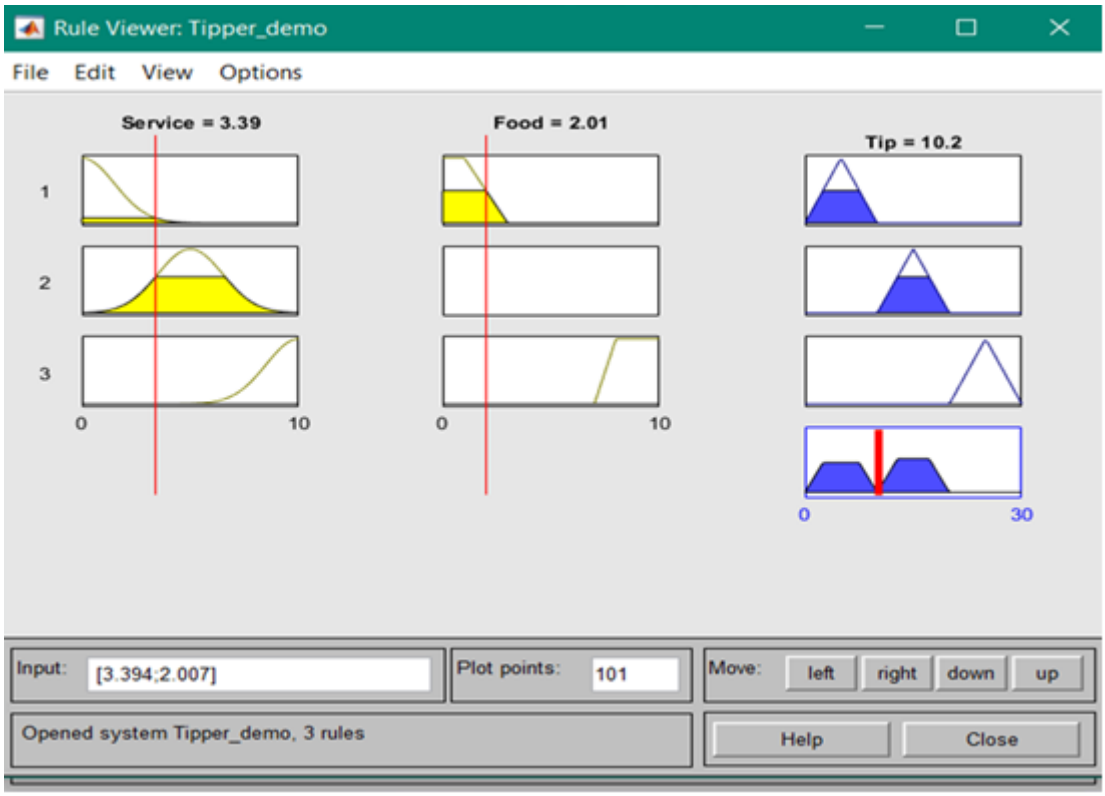
Output linguistic variables: Tip (Cheap, Average, Generous)

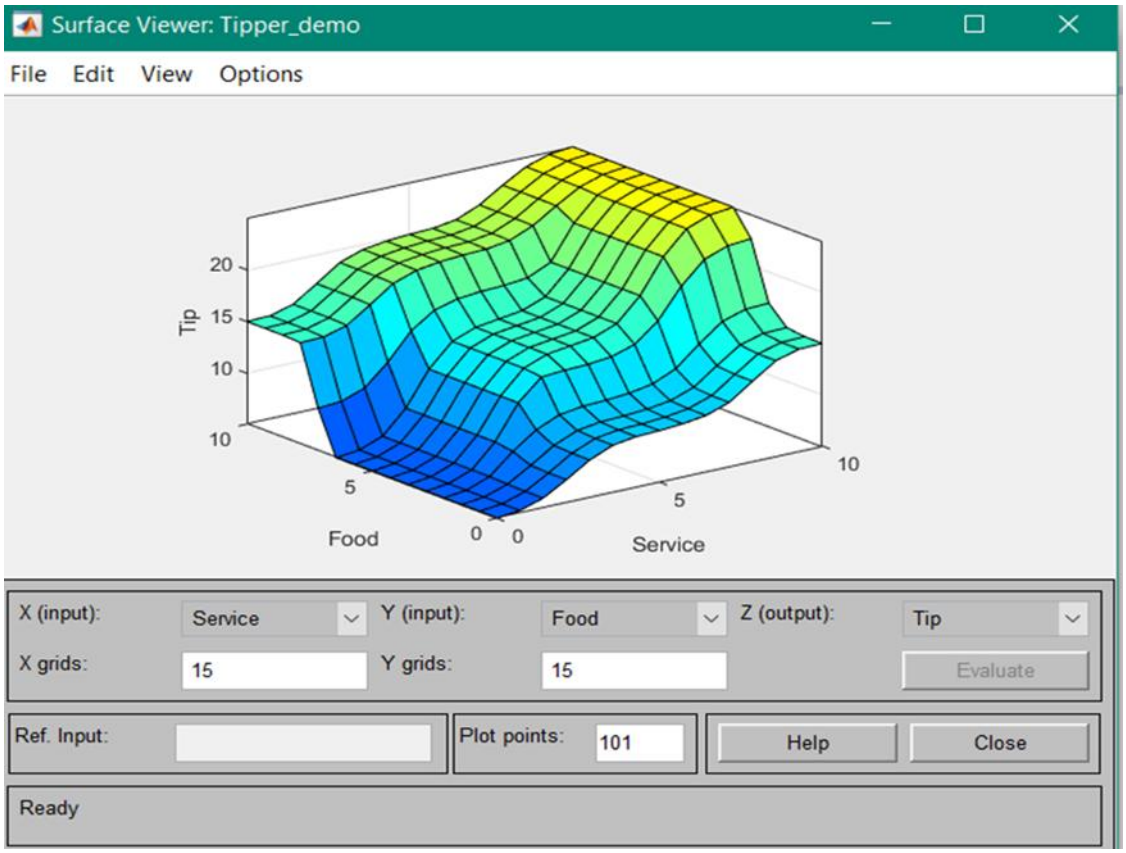


In general a compositional rule for inference involves the following procedure:

- Compute memberships of current inputs in the relevant antecedent fuzzy set of rule.
- If the antecedents are in conjunctive form, the AND operation is replaced by a minimum, if OR then by Maximum and similarly other operations are performed.
- Scale or clip the consequent fuzzy set of the rule by a minimum value found in step 2 since this gives the smallest degree to which the rule must fire.
- Repeat steps 1-3 for each rule in the rule base. Superpose the scaled or clipped consequent fuzzy sets formed by such a superposition. There are numerous variants of the defuzzification.


Rule Editor: Tipper_demo
File Edit View Options
1. If (Service is Poor) or (Food is Rancid) then (Tip is Cheap) (1)
2. If (Service is Good) then (Tip is Average) (1)
3. If (Service is Execellent) or (Food is Delicious) then (Tip is Generous) (1)
If
Service is
Poor
Good
Execellent
none
not
and
Food is
Rancid
Delicious
none
not
Then
Tip is
Cheap
Average
Generous
none
not
Connection
or
and
Weight:
1
Delete rule
Add rule
Change rule
<<
>>
FIS Name: Tipper_demo
Help
Close


Rule Viewer: Tipper_demo
File Edit View Options
Service = 3.39
Food = 2.01
Tip = 10.2
1
2
3
0
10
0
10
0
30
Input:
[3.394;2.007]
Plot points:
101
Move:
left
right
down
up
Opened system Tipper_demo, 3 rules
Help
Close



Conclusion:

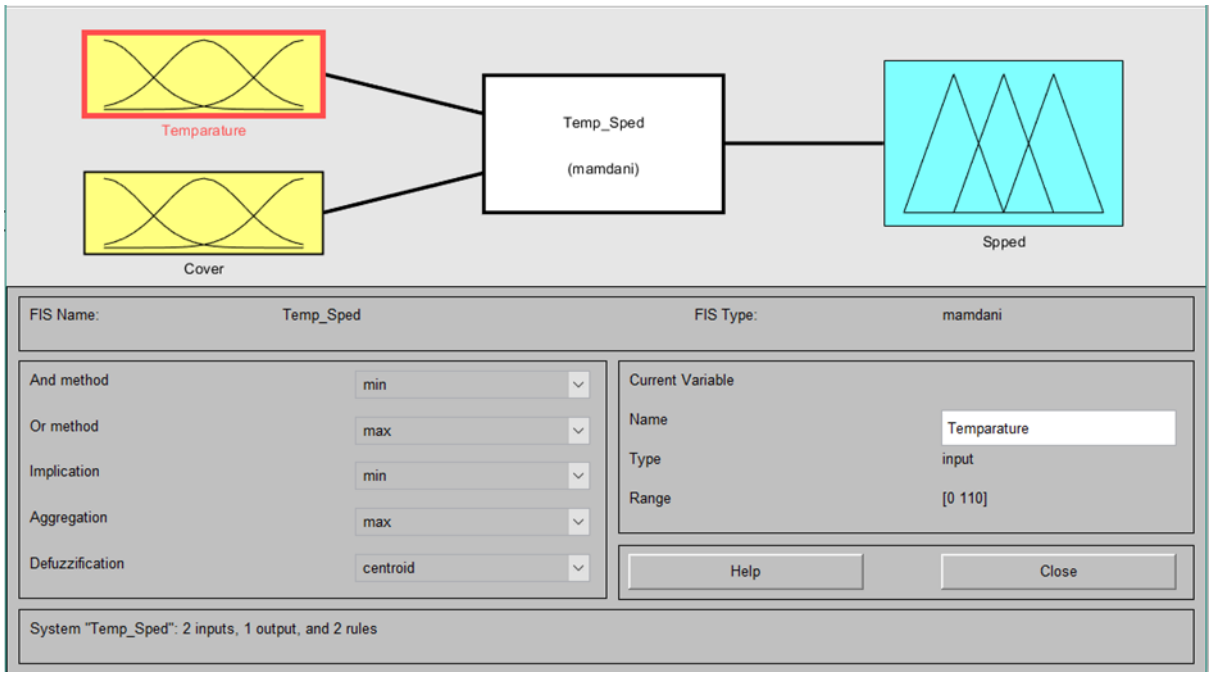
Reference:

- https://www.youtube.com/watch?v=O348HnWPm7A&t=1s
- https://www.youtube.com/watch?v=wBrHEXkTero&t=20s
- https://www.youtube.com/watch?v=LupUhRJo_sU&t=3s

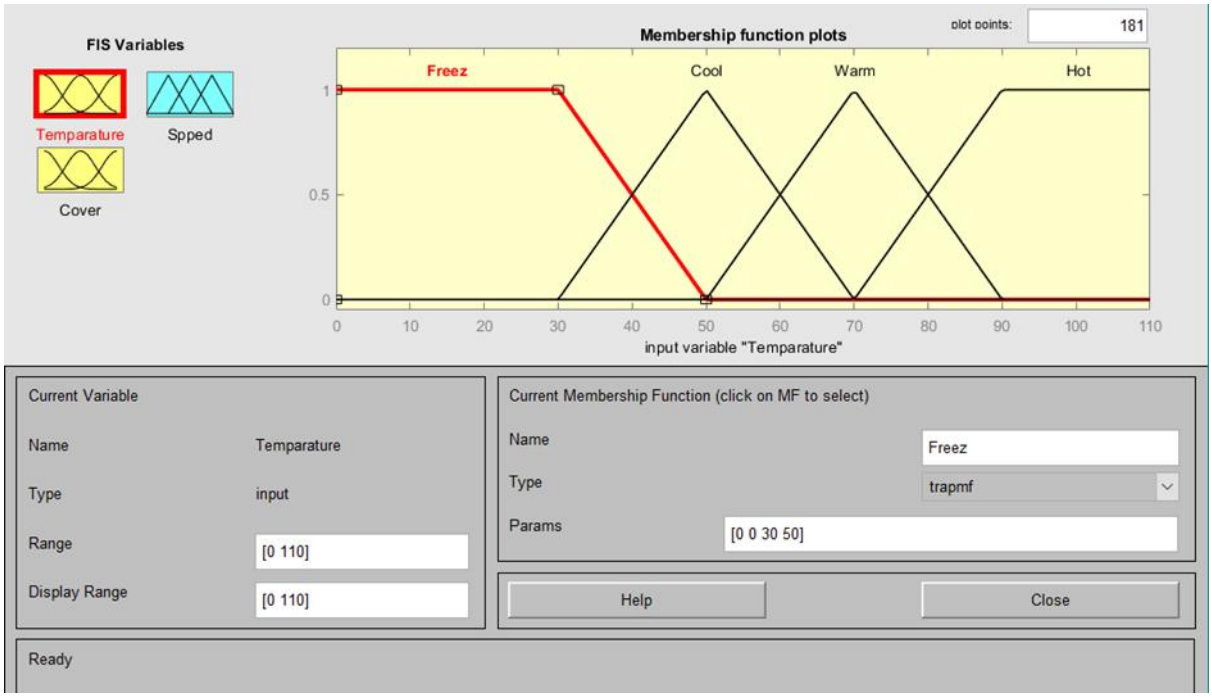
Experiment No. 6:
Title: Implement FIS Editor.

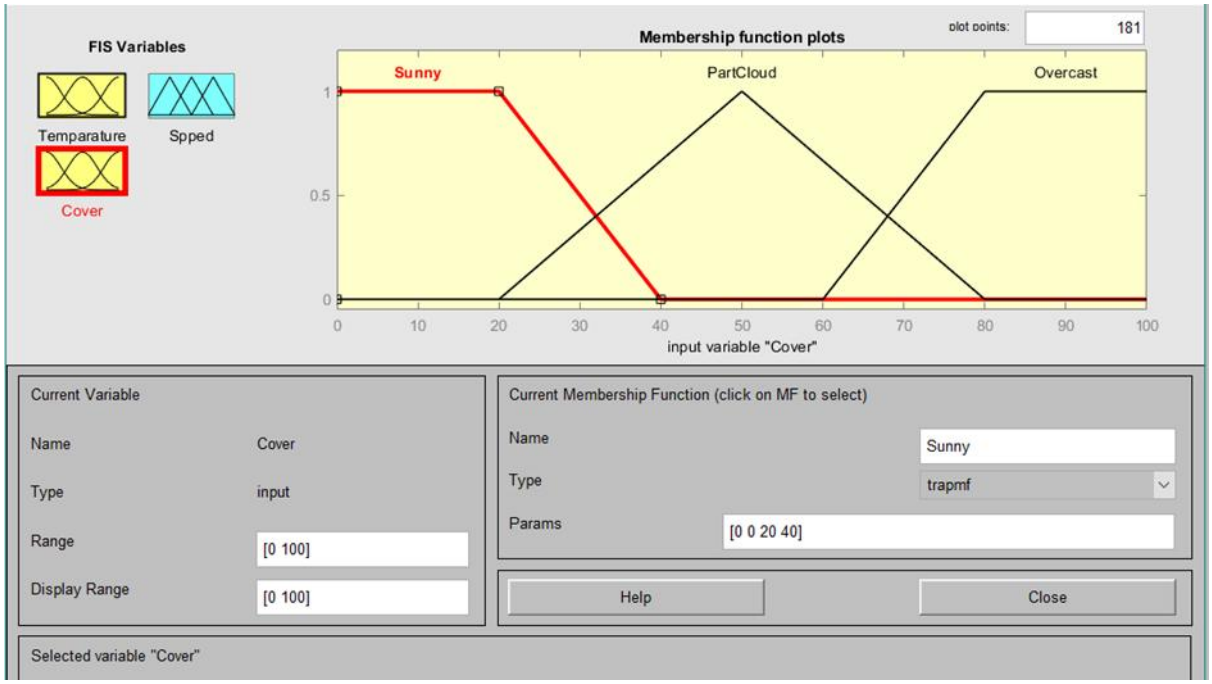
Aim: Write down briefly about the importance/ applicability of fuzzy editor.

Procedure and Results: You are required to explain how fuzzy editor can be implemented.

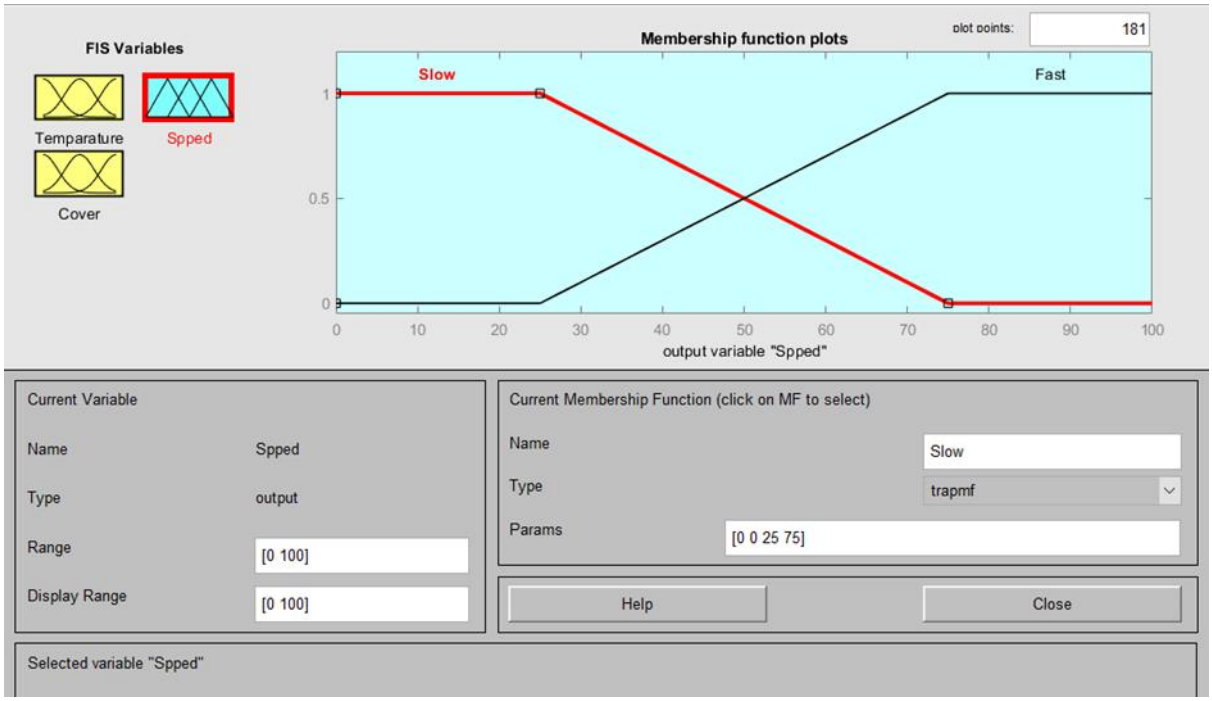


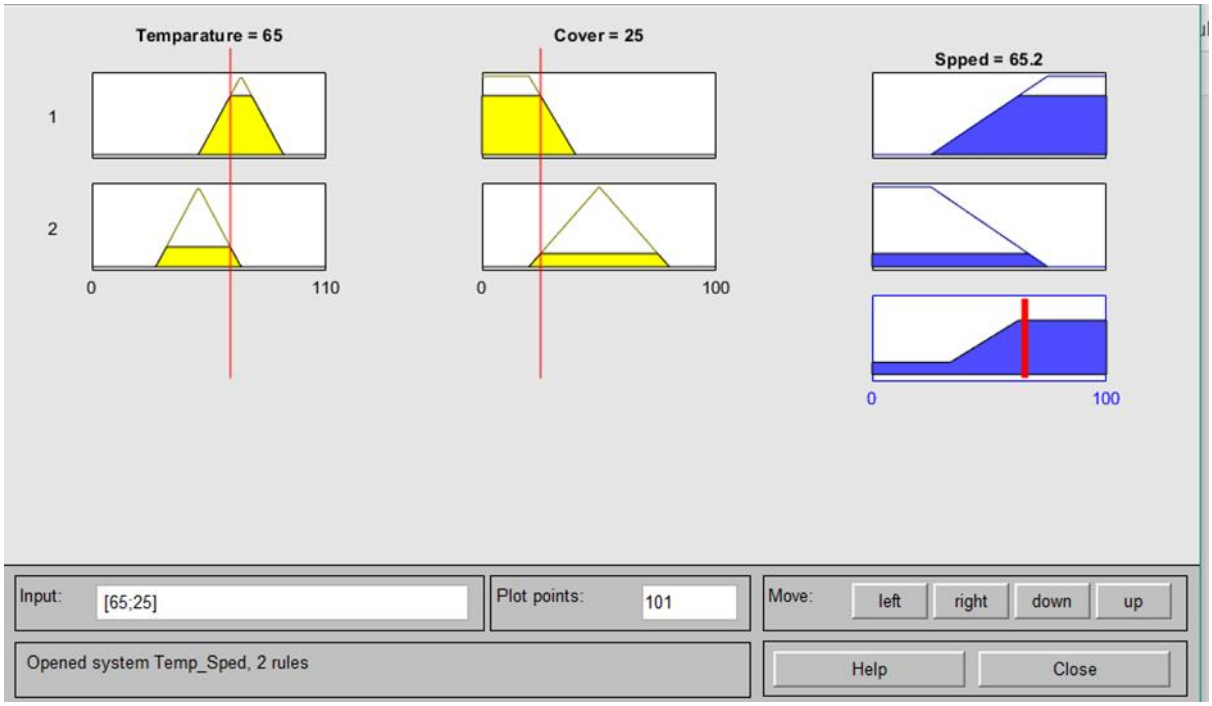
Input linguistic variables: Temperature (Freezing, Cool, Warm, Hot) & Cover (Sunny, Partly, Overcast)

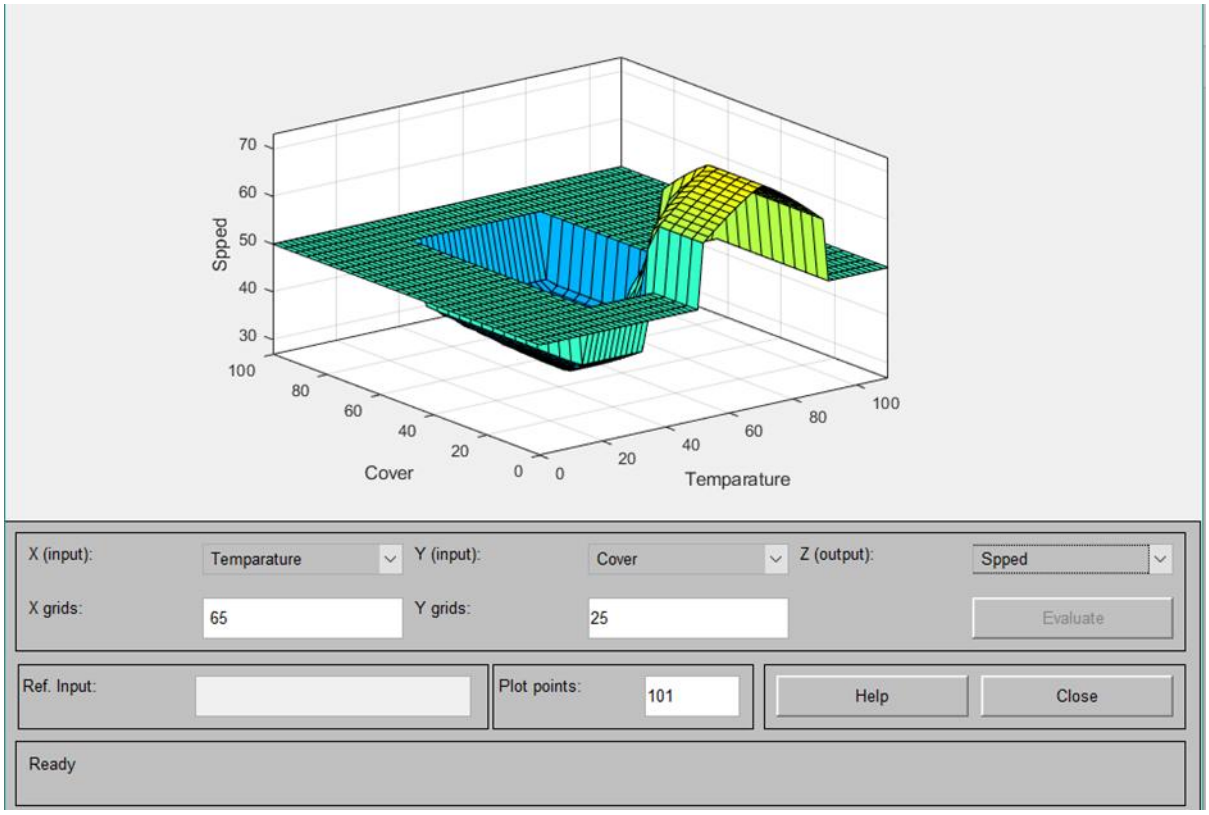




Output linguistic variables: Speed (Slow, Fast)







Conclusion:

Reference:

- https://www.youtube.com/watch?v=uBxWYTdF0UA

Experiment No: 7:
Title: Write a MATLAB Program on Basic Operations of Genetic Algorithm.

Aim: Write down briefly about the importance/ applicability of genetic algorithm.

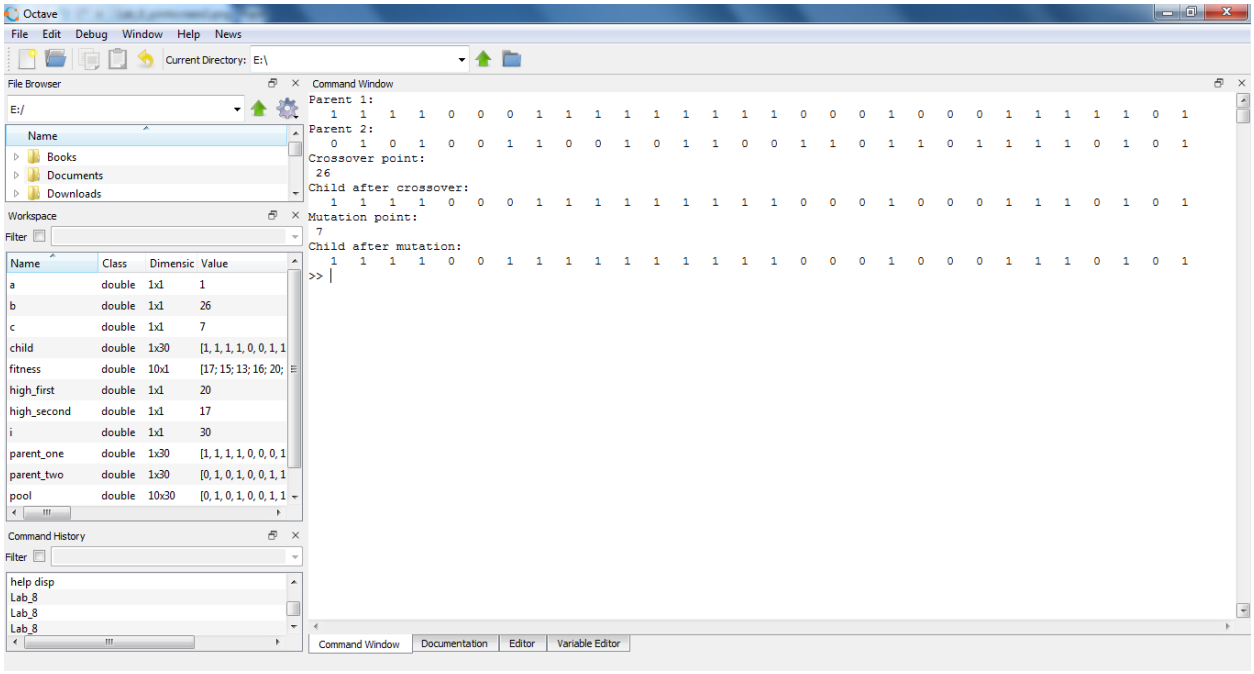
Theory: Write it as taught in the class.

Matlab Code:

```
clc
clear all
close all

%generation of genes randomly
%generate 10 genes each of length 30 using binary encoding
pool = randi([0,1], 10, 30);

%fitness is decided based on summation of values for each gene
fitness = sum(pool,2);

%selection of best fittted genes
high_first = max(fitness);
for i = 1:10
  if fitness(i) == high_first
    a=i;
  end
end
parent_one = pool(a,:);
disp('Parent 1:'), disp (parent_one)

high_second = max(fitness(fitness<max(fitness)));
for i = 1:10
  if fitness(i) == high_second
    a=i;
  end
end
parent_two = pool(a,:);
disp('Parent 2:'), disp (parent_two)

%crossover is done at any random point
b = randi([1 , 30]);
for i = 1:30
  if i <= b
    child(i) = parent_one(i);
  else
    child(i) = parent_two(i);
  end
end
disp('Crossover point:'), disp (b)
disp('Child after crossover:'), disp (child)
```

```
%mutation is done at any random point
c = randi([1 , 30]);
if child(c) == 0
  child(c) = 1;
end
disp('Mutation point:'), disp (c)
disp('Child after mutation:'), disp (child)
```

Results:



Conclusion:

Reference:

https://www.youtube.com/watch?v=mwXckn8up_U&list=PLsEIbHOtypISN0ZXjZ7Uhp0YwCToyrOLM&index=2

Experiment No: 8:
Title: Basic Understanding of Prolog Programming.

Aim: Write down briefly about the importance/ applicability of knowledge representation.

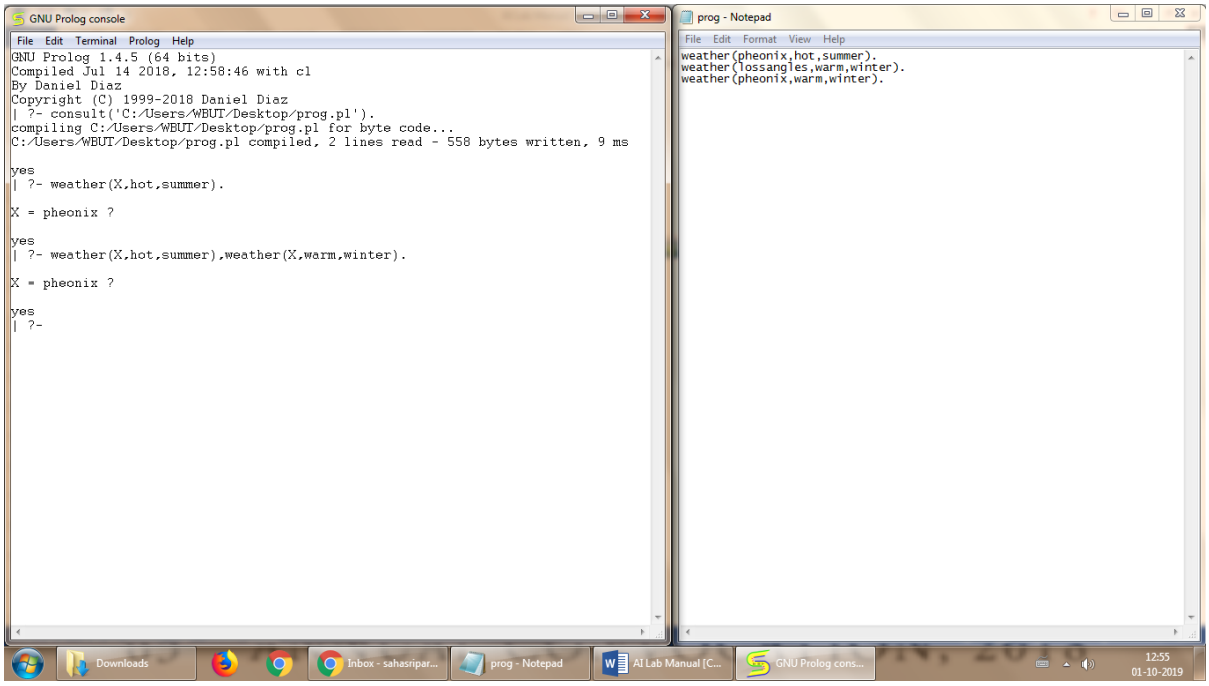
Theory: Write it as taught in the class.

Facts:
Pheonix is hot in summer.
Loss Angels is warm in winter.
Pheonix is warm in winter.

Query:
1. Which city is hot in summer?
2. Which city is hot in summer and warm in winter?

Code:
```
weather(pheonix,hot,summer).
weather(lossangles,warm,winter).
weather(pheonix,warm,winter).
```

Query:
```
weather(X,hot,summer).
weather(X,hot,summer),weather(X,warm,winter).
```



Facts:

a. Ram likes mango.
b. Seema is a girl.
c. Bill likes Cindy.
d. Rose is red.
e. John owns gold.

Code:

```
likes(ram, mango).
likes(bill, cindy).

girl(seema).

red(rose).

owns(john, gold).
```

Query:

```
likes(ram,What).
likes(Who,cindy).
red(What).
owns(Who,What).
```