Name: Tania Rajabally

Roll No: 43

UID: 2017130047

Batch: C

Course Code: CEL71 (AI and Soft Computing Lab)

Date:13/10/2020

**Experiment No 7**

**Fuzzy Controllers**

**Aim:** To implement Fuzzy Controllers for an Air Conditioner.

**Theory:**

A fuzzy control system is a control system based on fuzzy logic—a mathematical system that analyzes analog input values in terms of logical variables that take on continuous values between 0 and 1, in contrast to classical or digital logic, which operates on discrete values of either 1 or 0 (true or false, respectively).

Fuzzy logic is widely used in machine control. The term "fuzzy" refers to the fact that the logic involved can deal with concepts that cannot be expressed as the "true" or "false" but rather as "partially true". Although alternative approaches such as genetic algorithms and neural networks can perform just as well as fuzzy logic in many cases, fuzzy logic has the advantage that the solution to the problem can be cast in terms that human operators can understand, so that their experience can be used in the design of the controller. This makes it easier to mechanize tasks that are already successfully performed by humans.

A control system is an arrangement of physical components designed to alter another physical system so that this system exhibits certain desired characteristics. Following are some reasons of using Fuzzy Logic in Control Systems −

* While applying traditional control, one needs to know about the model and the objective function formulated in precise terms. This makes it very difficult to apply in many cases.
* By applying fuzzy logic for control we can utilize the human expertise and experience for designing a controller.
* The fuzzy control rules, basically the IF-THEN rules, can be best utilized in designing a controller.

While designing fuzzy control system, the following six basic assumptions should be made −

* The plant is observable and controllable − It must be assumed that the input, output as well as state variables are available for observation and controlling purpose.
* Existence of a knowledge body − It must be assumed that there exist a knowledge body having linguistic rules and a set of input-output data set from which rules can be extracted.
* Existence of solution − It must be assumed that there exists a solution.
* ‘Good enough’ solution is enough − The control engineering must look for ‘good enough’ solution rather than an optimum one.
* Range of precision − Fuzzy logic controller must be designed within an acceptable range of precision.
* Issues regarding stability and optimality − The issues of stability and optimality must be open in designing Fuzzy logic controller rather than addressed explicitly.

**Procedure:**

Followings are the major components of the FLC as shown in the above figure −

* Fuzzifier − The role of fuzzifier is to convert the crisp input values into fuzzy values.
* Fuzzy Knowledge Base − It stores the knowledge about all the input-output fuzzy relationships. It also has the membership function which defines the input variables to the fuzzy rule base and the output variables to the plant under control.
* Fuzzy Rule Base − It stores the knowledge about the operation of the process of domain.
* Inference Engine − It acts as a kernel of any FLC. Basically it simulates human decisions by performing approximate reasoning.
* Defuzzifier − The role of defuzzifier is to convert the fuzzy values into crisp values getting from fuzzy inference engine.

**Code:**

def moisture(y):

temperature ={}

if y<=50 and y>=0:

sm = (50-y)/50

mm = (y)/50

temperature['sm'] = sm

temperature['mm'] = mm

if y>50 and y<=100:

mm = (100-y)/50

hm = (y-50)/50

temperature['mm'] = mm

temperature['hm'] = hm

return temperature

def temperature(x):

temperature ={}

if x<=50 and x>=0:

st = (50-x)/50

mt = (x)/50

temperature['st'] = st

temperature['mt'] = mt

if x>50 and x<=100:

mt = (100-x)/50

ht = (x-50)/50

temperature['mt'] = mt

temperature['ht'] = ht

return temperature

def compressionSpeed(rule,value):

if rule =='sc':

compressionSpeed = 30 - (30 \* value)

return compressionSpeed

if rule == 'mc':

compressionSpeed1 = 30 \* value

compressionSpeed2 = 60 - (30 \* value)

return (compressionSpeed1 + compressionSpeed2) /2

if rule == 'hc':

compressionSpeed = (30\*value) +30

return compressionSpeed

ruleBase= [['sc','mc','hc'],['sc','mc','hc'],['mc','hc','hc']]

Temperature = temperature(20)

Moisture = moisture(30)

print("From Temperature Fuzzy function:",Temp)

print("From Moisture Fuzzy function:",Moisture)

minList =[]

listOfCombinations =[]

for x in Temperature:

for y in Moisture:

if Temp[x] < Moisture[y]:

minList.append(Temperature[x])

listOfCombinations.append([x,y])

else:

minList.append(Moisture[y])

listOfCombinations.append([x,y])

print("List of minimum values:",minList)

indexOfMax =minList.index(max(minList))

maxOfMinList = max(minList)

# print(indexOfMax)

print("MinMax output:",maxOfMinList)

print("Combination of Temp and Moisture giving this value:",listOfCombinations[indexOfMax])

MoistureList = ['sm','mm','hm']

TempList = ['st','mt','ht']

temperatureRule = TempList.index(listOfCombinations[indexOfMax][0])

moistureRule = MoistureList.index(listOfCombinations[indexOfMax][1])

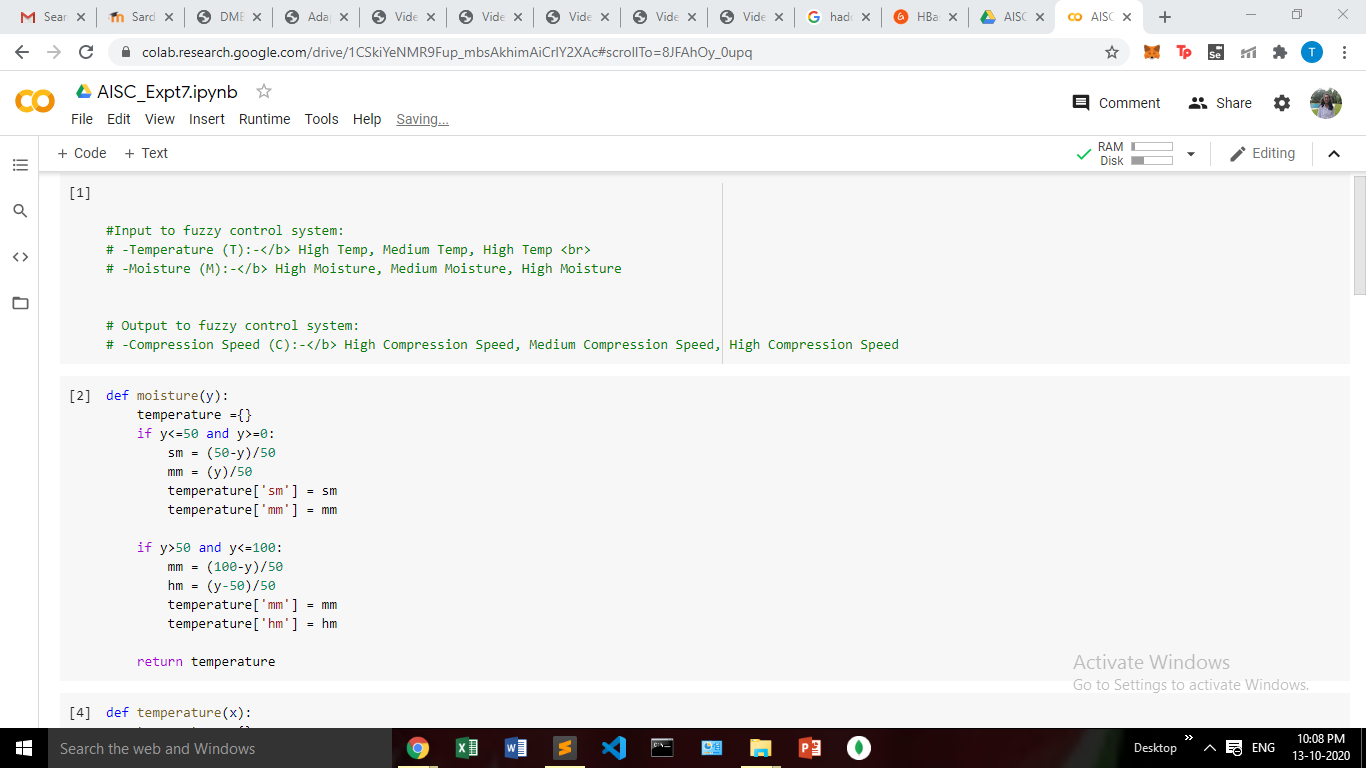
compressionRule = ruleBase[temperatureRule][moistureRule]

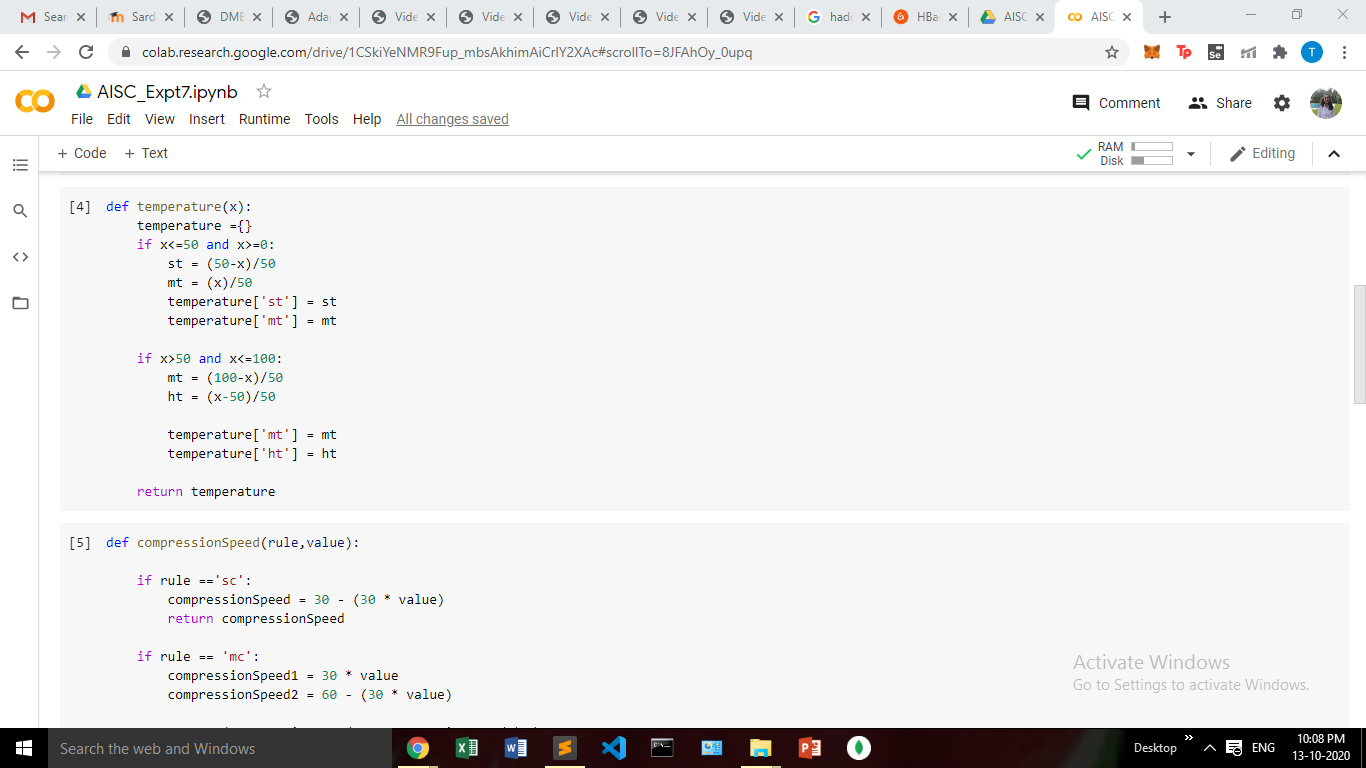
print("Compression rule which follows is: ",compressionRule)

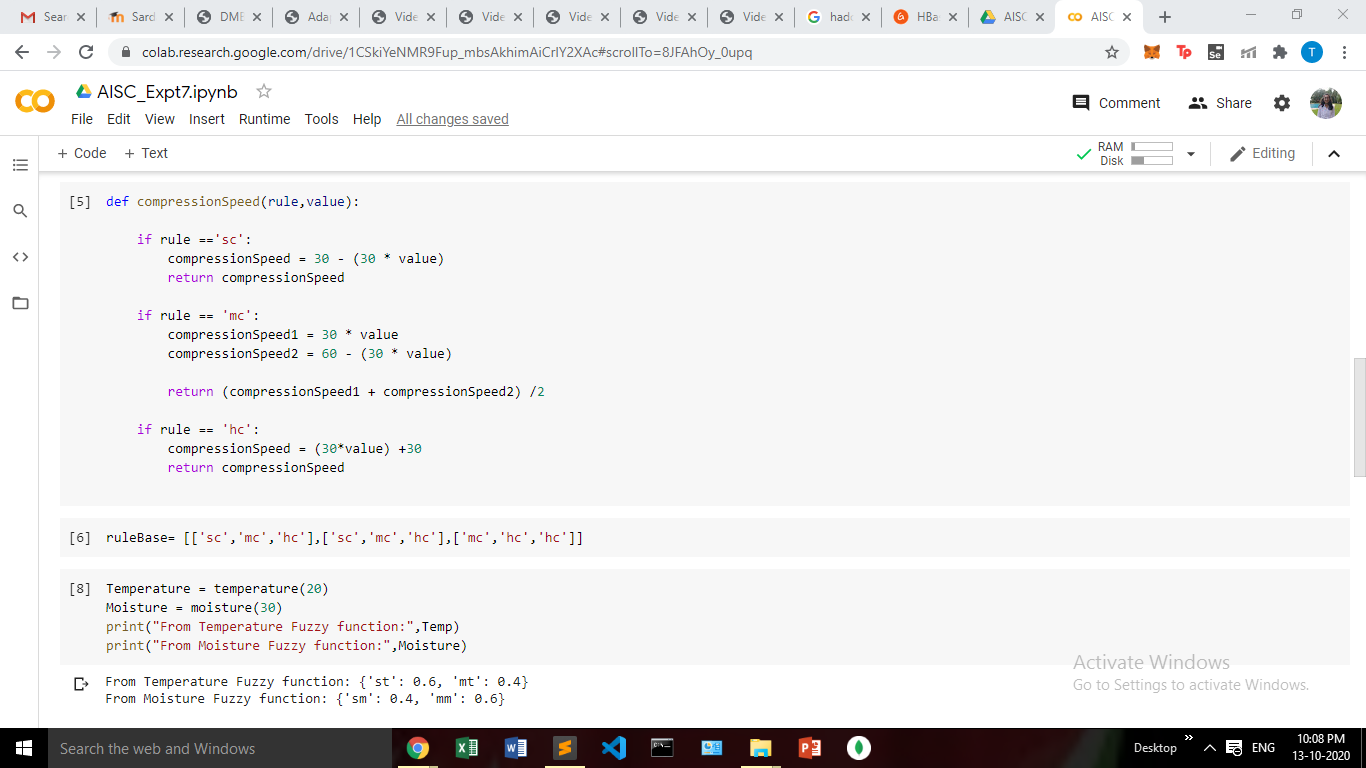
CompressionSpeed = compressionSpeed(compressionRule,maxOfMinList)

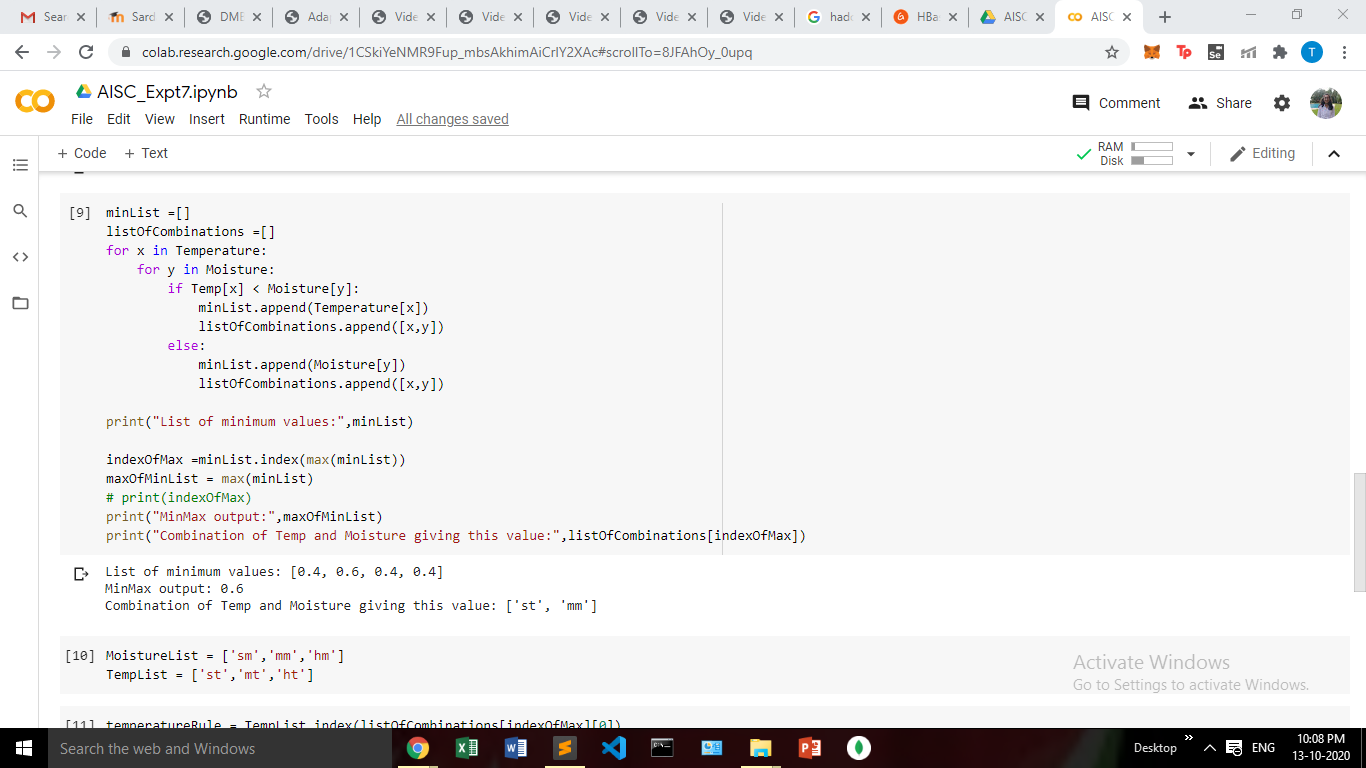
print("Compression Speed at 20% Temp and 30% Humidity=",CompressionSpeed)

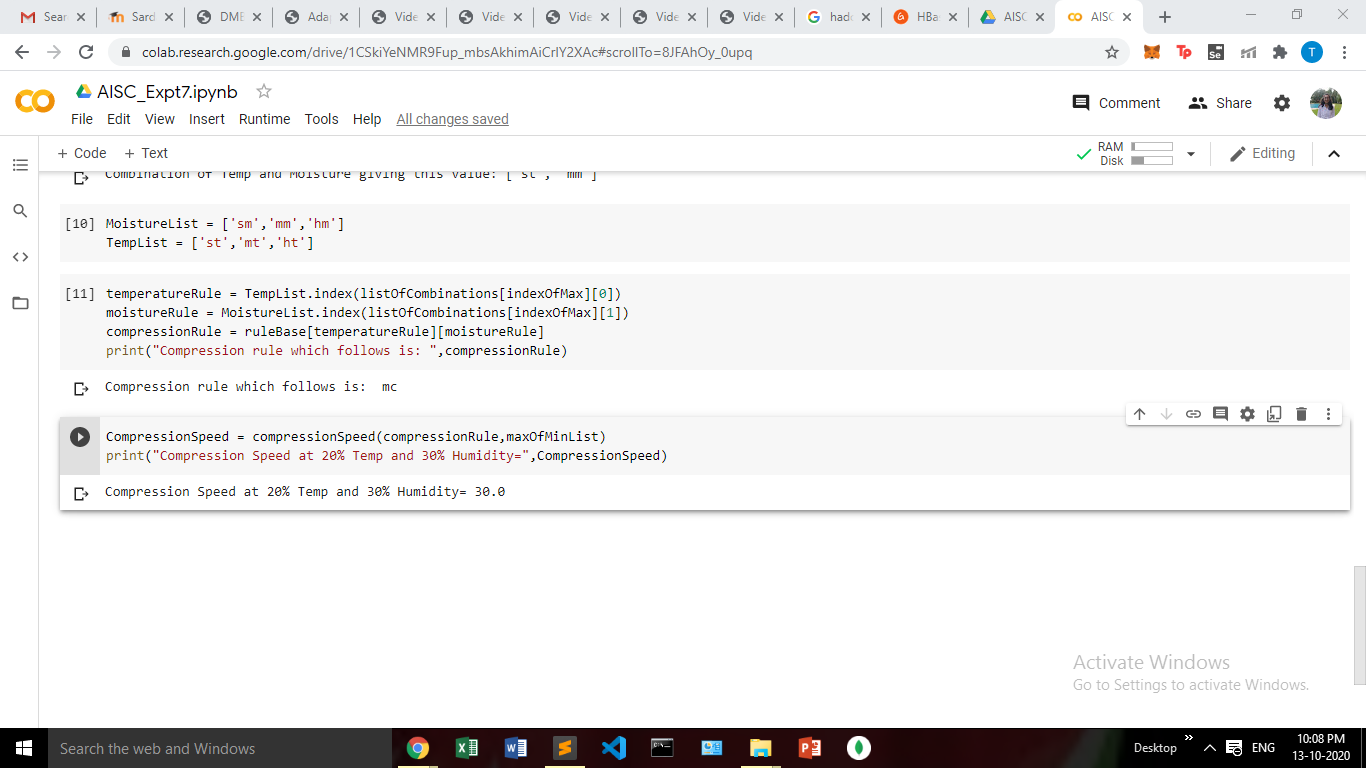
**Output:**

****

****

****

****

****

**Conclusion:**

In this experiment, we have found out the fuzzy rules and made a fuzzy control system for a air conditioner. We then found the compression speed for a certain temperature and humidity.