Batch: A1 Roll No.: 16010120015

Experiment No. 7

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

Title: : Fuzzification methods and operations on fuzzy set

Aim : To implement fuzzification methods and perform operations on fuzzy sets

---

Expected Outcome of Experiment:

CO4 : Apply basics of Fuzzy logic and neural networks

Books/ Journals/ Websites referred:

Pre Lab/ Prior Concepts:
Fuzzy logic:

The term **fuzzy** refers to things that are not clear or are vague. In the real world many times we encounter a situation when we cannot determine whether the state is true or false, their fuzzy logic provides very valuable flexibility for reasoning. In this way, we can consider the inaccuracies and uncertainties of any situation.

Following are the characteristics of fuzzy logic:

1. This concept is flexible and we can easily understand and implement it.
2. It is used for helping the minimization of the logics created by the human.
3. It is the best method for finding the solution of those problems which are suitable for approximate or uncertain reasoning.
4. It always offers two values, which denote the two possible solutions for a problem and statement.

5. It allows users to build or create the functions which are non-linear of arbitrary complexity.

In the architecture of the **Fuzzy Logic** system, each component plays an important role. The architecture consists of the different four components which are given below.

1. Rule Base
2. Fuzzification
3. Inference Engine
4. Defuzzification

Definition of membership function (and fuzzy set):

A graph that defines how each point in the input space is mapped to membership value between 0 and 1. Input space is often referred to as the universe of discourse or universal set (u), which contains all the possible elements of concern in each application.

There are largely three types of fuzzifiers:

- Singleton fuzzifier
- Gaussian fuzzifier
- Trapezoidal or triangular fuzzifier

Fuzzy set operations

1. **Union:**

Consider 2 Fuzzy Sets denoted by A and B, then let us consider Y be the Union of them, then for every member of A and B, Y will be:

degree_of_membership(Y)= max(degree_of_membership(A), degree_of_membership(B))

2. **Intersection:**

Consider 2 Fuzzy Sets denoted by A and B, then let us consider Y be the Intersection of them, then for every member of A and B, Y will be:

degree_of_membership(Y)= min(degree_of_membership(A), degree_of_membership(B))

3. Complement:

Consider a Fuzzy Sets denoted by A, then let us consider Y be the Complement of it, then for every member of A, Y will be:

degree_of_membership(Y)= 1 - degree_of_membership(A)

4. Difference:

Consider 2 Fuzzy Sets denoted by A and B, then let us consider Y be the Intersection of them, then for every member of A and B, Y will be:

degree_of_membership(Y)= min(degree_of_membership(A), 1- degree_of_membership(B))

Fuzzy inference system

Fuzzy Inference System is the key unit of a fuzzy logic system having decision making as its primary work. It uses the "IF…THEN" rules along with connectors "OR" or "AND" for drawing essential decision rules.

Characteristics of Fuzzy Inference System

Following are some characteristics of FIS –

- The output from FIS is always a fuzzy set irrespective of its input which can be fuzzy or crisp.
- It is necessary to have fuzzy output when it is used as a controller.
- A defuzzification unit would be there with FIS to convert fuzzy variables into crisp variables.

Implementation Details:

1. Implement the following membership functions and visualize the same
    a. Triangular.
    b. Trapezoidal
    c. Gaussian
    d. Generalized
    e. Sigmoid

Code:

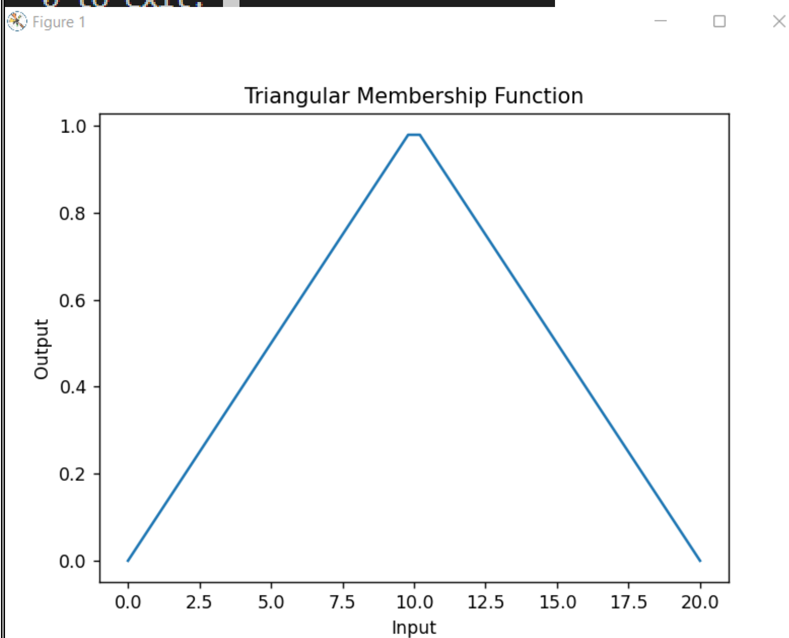
```
import math
import numpy as np
import matplotlib.pyplot as plt
import random

#Triangular Membership Function
def triangular(arr,a,b,c):
  triangular=[]
  for s in range(0,len(arr)):
    if arr[s]<=a:
      triangular.append(0)
    elif arr[s]<=b:
      out = (arr[s]-a)/(b-a)
      triangular.append(out)
    elif arr[s]<=c:
      out = (c-arr[s])/(c-b)
      triangular.append(out)
    else:
      triangular.append(0)
  return triangular

  #Trapezoidal Membership Function
def trapezoidal(arr,a,b,c,d):
  trapezoidal=[]
  for s in range(0,len(arr)):
    if arr[s]<=a:
      trapezoidal.append(0)
    elif arr[s]<=b:
      out = (arr[s]-a)/(b-a)
      trapezoidal.append(out)
    elif arr[s]<=c:
      trapezoidal.append(1)
    elif arr[s]<=d:
      out = (d-arr[s])/(d-c)
      trapezoidal.append(out)
    else:
      trapezoidal.append(0)
  return trapezoidal

  #Gaussian Membership Function
```

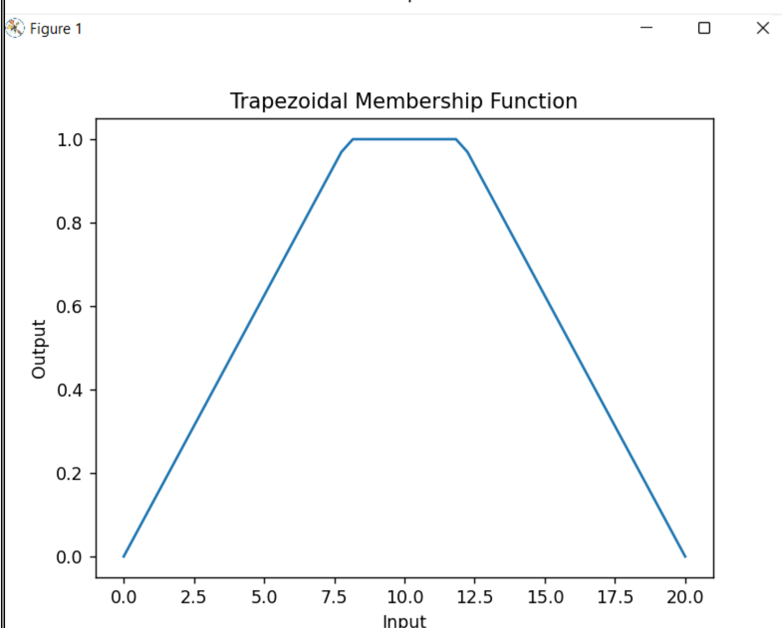
```
def gaussian(arr,c, sigma):
  gaussian = []
  for s in range(0, len(arr)):
    out = math.e**(-0.5*(((arr[s]-c)/sigma)**2))
    gaussian.append(out)
  return gaussian

  #Generalized Membership Function
def generalized(arr, a, b, c):
  generalized = []
  for s in range(0, len(arr)):
    out = 1/(1+(abs((arr[s]-c)/a))**(2*b))
    generalized.append(out)
  return generalized

  #Sigmoidal Membership Function
def sigmoidal(arr, a, c):
  sigmoidal = []
  for s in range(0, len(arr)):
    out = 1/(1+(math.e**(-a/(arr[s]-c))))
    sigmoidal.append(out)
  return sigmoidal

while True:
    choice = int(input("Enter \n1 for Triangular \n2 for Trapezoidal\n3 for Gaussian\n4 for Generalized\n5 for sigmoidal\n 0 to exit: "))

    if choice == 1:
        x = np.linspace(0, 20)
        y = triangular(x, 0, 10, 20)
        plt.plot(x,y)
        plt.title("Triangular Membership Function")
        plt.xlabel("Input")
        plt.ylabel("Output")
        plt.show()

    elif choice == 2:
        x = np.linspace(0, 20)
        y = trapezoidal(x, 0, 8, 12, 20)
        plt.plot(x,y)
        plt.title("Trapezoidal Membership Function")
        plt.xlabel("Input")
```

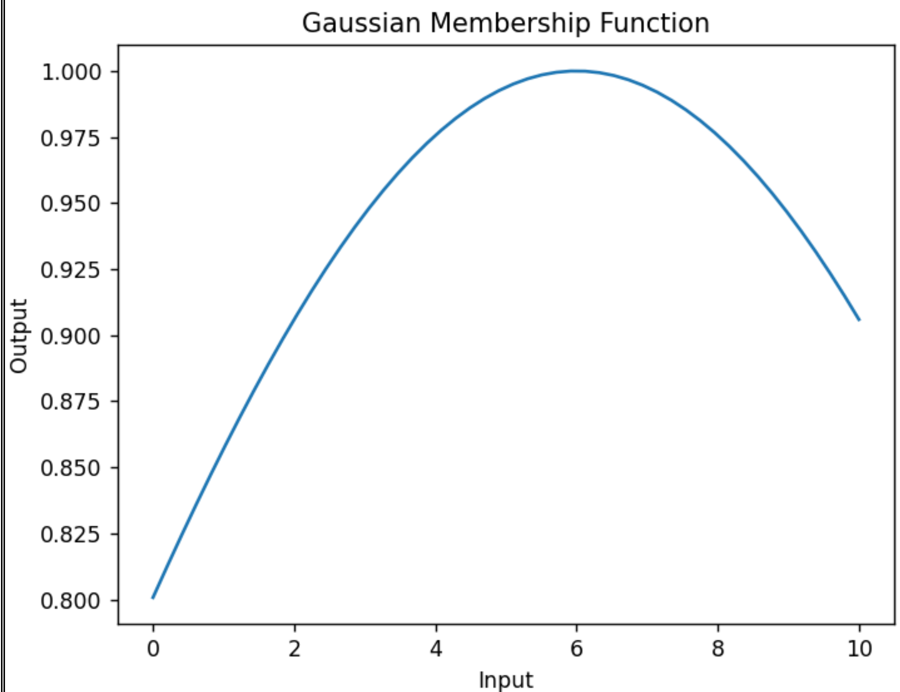
```
        plt.ylabel("Output")
        plt.show()

    elif choice == 3:
        x = np.linspace(0, 10)
        y = gaussian(x, 6,9)
        plt.plot(x,y)
        plt.title("Gaussian Membership Function")
        plt.xlabel("Input")
        plt.ylabel("Output")
        plt.show()

    elif choice == 4:
        x = np.linspace(-5, 5)
        y = generalized(x, 3,5,3)
        plt.plot(x,y)
        plt.title("Generalized Membership Function")
        plt.xlabel("Input")
        plt.ylabel("Output")
        plt.show()

    elif choice == 5:
        x = np.linspace(-5, 5)
        y = sigmoidal(x, 5,2)
        plt.plot(x,y)
        plt.title("Sigmoidal Membership Function")
        plt.xlabel("Input")
        plt.ylabel("Output")
        plt.show()

    elif choice == 0:
        break

    else:
        print("Please enter valid choice")
```

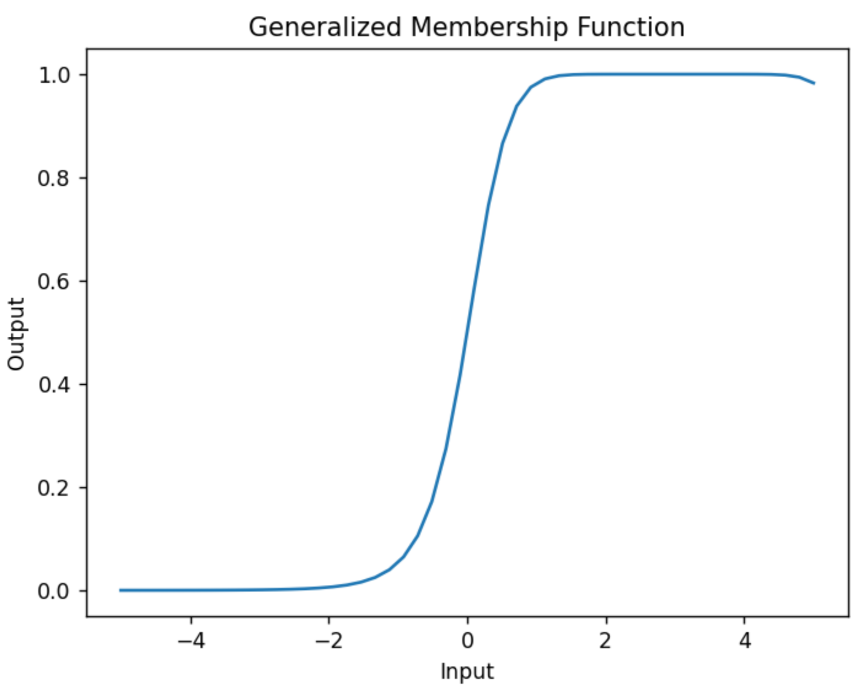
Output:

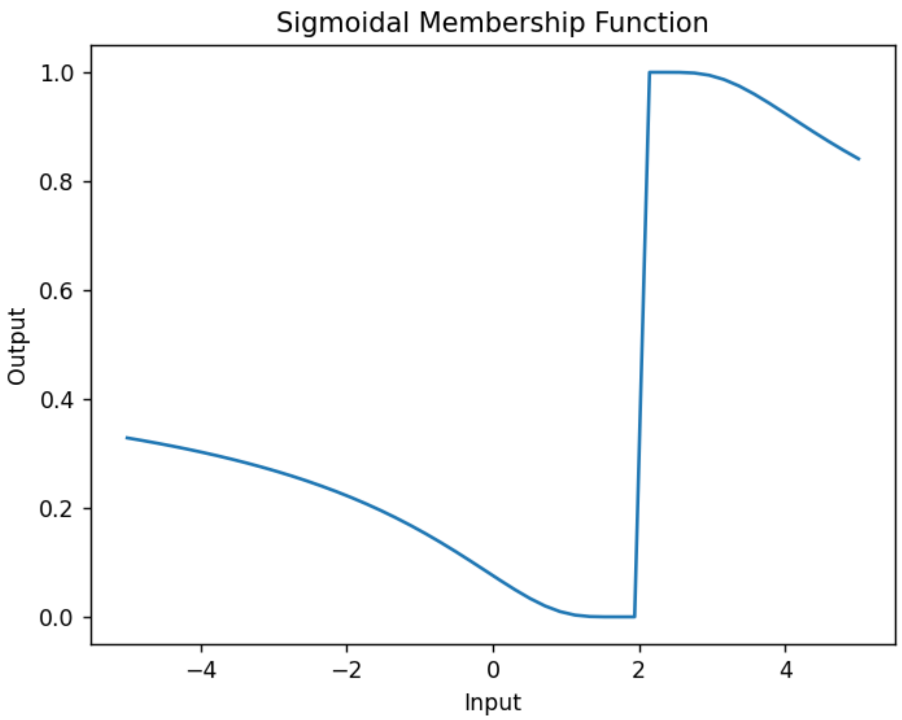
```
PS D:\Semester 5\SC> & "C:/Progr
Enter
1 for Triangular
2 for Trapezoidal
3 for Gaussian
4 for Generalized
5 for sigmoidal
 0 to exit:
```






Gaussian Membership Function
Output
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0.800
0
2
4
6
8
10
Input


Generalized Membership Function
Output
1.0
0.8
0.6
0.4
0.2
0.0
−4
−2
0
2
4
Input



2. Implement the fuzzy operations- Union, intersection, compliment and visualize the same.

```
A = dict()
B = dict()

A = {"a": 0.8, "b": 0.5, "c": 0.3, "d": 2.7}
B = {"a": 0.2, "b": 1.1, "c": 2.7, "d": 1.2}

print("Fuzzy Set A:", A)
print("Fuzzy Set B:", B)

intersect = dict()
complement = dict()
difference = dict()

while True:
    choice = int(input("Enter \n1 for Union \n2 for Intersection\n3 for Complement\n4 for Difference\n0 to exit: "))

    if choice == 1:            #Union
```

```
    for a_key, b_key in zip(A,B):
        a_val = A[a_key]
        b_val = B[b_key]

        if a_val >= b_val:
            union[a_key] = a_val

        else:
            union[b_key] = b_val

    print("Fuzzy Set Union of A and B: ", union)

elif choice == 2:           #Intersection

    for a_key, b_key in zip(A,B):
        a_val = A[a_key]
        b_val = B[b_key]

        if a_val <= b_val:
            intersect[a_key] = a_val

        else:
            intersect[b_key] = b_val

    print("Fuzzy Set Intersection of A and B: ", intersect)

elif choice == 3:          #Complement

    for a_key in A:
        a_val = A[a_key]
        complement[a_key] = round((1 - a_val), 1)

    print("Fuzzy Set Complement of A: ", complement)

elif choice == 4:             #Difference

    for a_key, b_key in zip(A,B):
        a_val = A[a_key]
        b_val = B[b_key]
        b_val = round((1 - B[b_key]), 1)
```

```python
            if a_val <= b_val:
                difference[a_key] = a_val

            else:
                difference[b_key] = b_val

        print("Fuzzy Set Difference of A and B: ", difference)

    elif choice == 0:
        break

    else:
        print("Please enter valid choice")
```

Output:

```
Fuzzy Set A: {'a': 0.3, 'b': 0.8, 'c': 0.6, 'd': 0.7}
Fuzzy Set B: {'a': 0.2, 'b': 1.9, 'c': 2.3, 'd': 1.4}
Enter
1 for Union
2 for Intersection
3 for Complement
4 for Difference
0 to exit: 1
Fuzzy Set Union of A and B:  {'a': 0.3, 'b': 1.9, 'c': 2.3, 'd': 1.4}
Enter
1 for Union
2 for Intersection
3 for Complement
4 for Difference
0 to exit: 2
Fuzzy Set Intersection of A and B:  {'a': 0.2, 'b': 0.8, 'c': 0.6, 'd': 0.7}
Enter
1 for Union
2 for Intersection
3 for Complement
4 for Difference
0 to exit: 3
Fuzzy Set Complement of A:  {'a': 0.7, 'b': 0.2, 'c': 0.4, 'd': 0.3}
Enter
1 for Union
2 for Intersection
3 for Complement
4 for Difference
0 to exit: 4
Fuzzy Set Difference of A and B:  {'a': 0.3, 'b': -0.9, 'c': -1.3, 'd': -0.4}
Enter
1 for Union
2 for Intersection
3 for Complement
4 for Difference
0 to exit: 0
PS D:\Semester 5\SC>
```

Post Lab Descriptive Questions :

1. Implement the fuzzy membership functions to define the fuzzy set for the following course evaluation crisp set
   (Hint: descriptors for fuzzy set can be bad, poor, average, good, very good, excellent)

$EX = \text{Marks} \geq 90$

$A = 80 \leq \text{Marks} < 90$

$B = 70 \leq \text{Marks} < 80$

$C = 60 \leq \text{Marks} < 70$

$D = 50 \leq \text{Marks} < 60$

$P = 35 \leq \text{Marks} < 50$

$F = \text{Marks} < 35$

```python
marks = int(input("Enter marks: "))

def Postlab(x):
    if x>=90:
        return '0'
    elif x>=80:
        return 'A'
    elif x>=70:
        return 'B'
    elif x>=60:
        return 'C'
    elif x>=50:
        return 'D'
    elif x>=35:
        return 'P'
    else:
        return 'F'
print("your grade is ",Postlab(marks))
```

Output:

```
PS D:\Semester 5\SC> & "C:/Program Files/
Enter marks: 45
your grade is  P
PS D:\Semester 5\SC> & "C:/Program Files/
Enter marks: 85
your grade is  A
PS D:\Semester 5\SC> 
```

Date: 29/11/2022 **Signature of faculty in-charge**