Name: Vidhi Gulhane **Roll no:** 15 **Class:** D20B

EXPERIMENT 6

Aim:

To implement Fuzzy Membership Functions.

Theory:

Fuzzy Membership Functions are used to map elements of a universe of discourse to membership values ranging between 0 and 1. These functions are essential for fuzzy logic systems that deal with imprecise or uncertain information.

The membership functions used in fuzzy logic systems can take various forms, such as:

1. **Singleton Membership Function:**

   ○ This is the simplest membership function where a single point has full membership (value = 1), and all other points have zero membership.

   ○ **Equation:**

$$\mu(x) = \begin{cases} 1 & \text{if } x = x_0 \\ 0 & \text{if } x \neq x_0 \end{cases}$$

   This function is useful when a specific value is given full membership in a fuzzy set.

2. **Triangular Membership Function:**

   ○ A triangular function is defined by three points: a left endpoint, a peak, and a right endpoint. This is a widely used membership function due to its simplicity.

   ○ **Equation:**

$$\mu(x) = \begin{cases} 0 & \text{if } x \leq a \text{ or } x \geq c \\ \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ \frac{c-x}{c-b} & \text{if } b \leq x \leq c \end{cases}$$

   This function is often used to represent fuzzy sets that gradually increase and then decrease.

3. **Trapezoidal Membership Function:**

   ○ Similar to the triangular function but with a flat top. All values in the top range have full membership (value = 1).

   ○ **Equation:**

$$\mu(x) = \begin{cases} 0 & \text{if } x \leq a \text{ or } x \geq d \\ \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ 1 & \text{if } b \leq x \leq c \\ \frac{d-x}{d-c} & \text{if } c \leq x \leq d \end{cases}$$

   This function is useful for fuzzy sets that have a range of values with full membership.

4. **Gaussian Membership Function:**

   ○ A bell-shaped curve that smoothly transitions from 0 to 1 and back to 0. It is useful when smooth transitions are needed.

   ○ **Equation:**

$$\mu(x) = e^{-\frac{(x-c)^2}{2\sigma^2}}$$

   This function represents sets where elements gradually move from "no membership" to "full membership" and back.

Code:

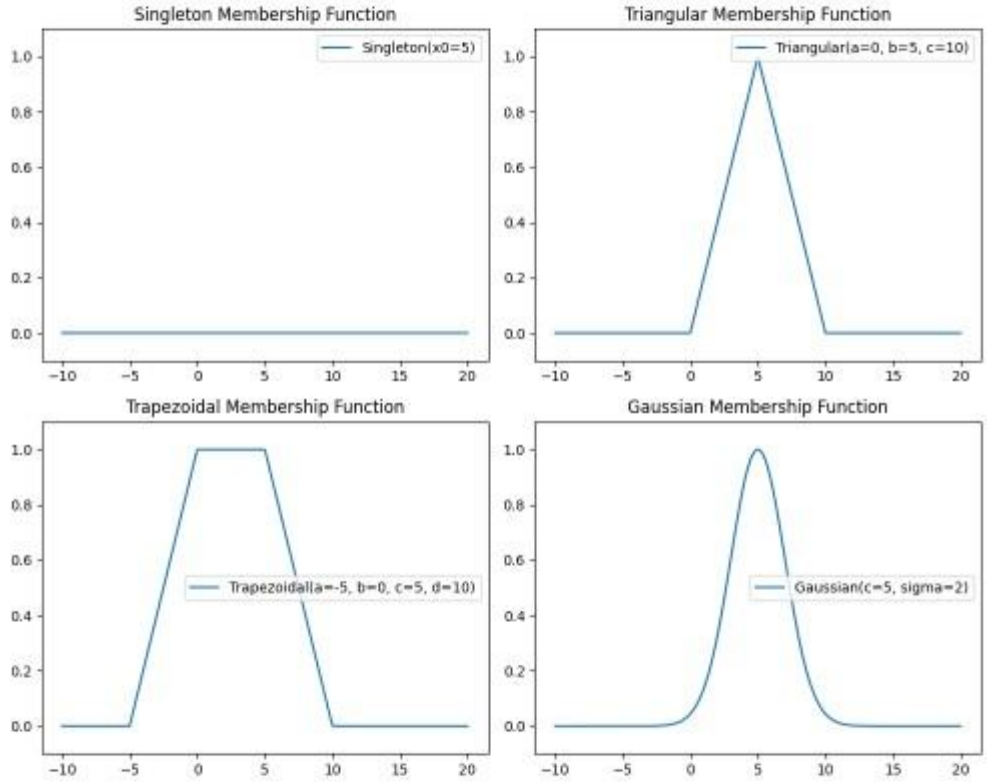
```
import numpy as np
import matplotlib.pyplot as plt

# Singleton Membership Function
def singleton_mf(x, x0):
    return np.where(x == x0, 1, 0)
```

```

# Triangular Membership Function
```

```
def triangular_mf(x, a, b, c):
    return np.maximum(np.minimum((x-a)/(b-a), (c-x)/(c-b)), 0)

# Trapezoidal Membership Function
def trapezoidal_mf(x, a, b, c, d):
    return np.maximum(np.minimum(np.minimum((x-a)/(b-a), 1), (d-x)/(d-c)),
0)

# Gaussian Membership Function def
gaussian_mf(x, c, sigma):
    return np.exp(-((x-c)**2) / (2*sigma**2))

# Plotting the membership functions
x = np.linspace(-10, 20, 500)

plt.figure(figsize=(10, 8))

# Singleton
plt.subplot(2, 2, 1)
plt.plot(x, singleton_mf(x, 5), label="Singleton(x0=5)")
plt.title('Singleton Membership Function')
plt.ylim(-0.1, 1.1)
plt.legend()

# Triangular
plt.subplot(2, 2, 2)
plt.plot(x, triangular_mf(x, 0, 5, 10), label="Triangular(a=0, b=5,
c=10)")
plt.title('Triangular Membership Function')
plt.ylim(-0.1, 1.1) plt.legend()


# Trapezoidal
plt.subplot(2, 2, 3)
plt.plot(x, trapezoidal_mf(x, -5, 0, 5, 10), label="Trapezoidal(a=-5, b=0,
c=5, d=10)")
```

```
plt.title('Trapezoidal Membership Function') plt.ylim(-0.1,
1.1)

plt.legend()


# Gaussian

plt.subplot(2, 2, 4)

plt.plot(x, gaussian_mf(x, 5, 2), label="Gaussian(c=5, sigma=2)")

plt.title('Gaussian Membership Function')

plt.ylim(-0.1, 1.1)

plt.legend()


plt.tight_layout()

plt.show()
```

Output:



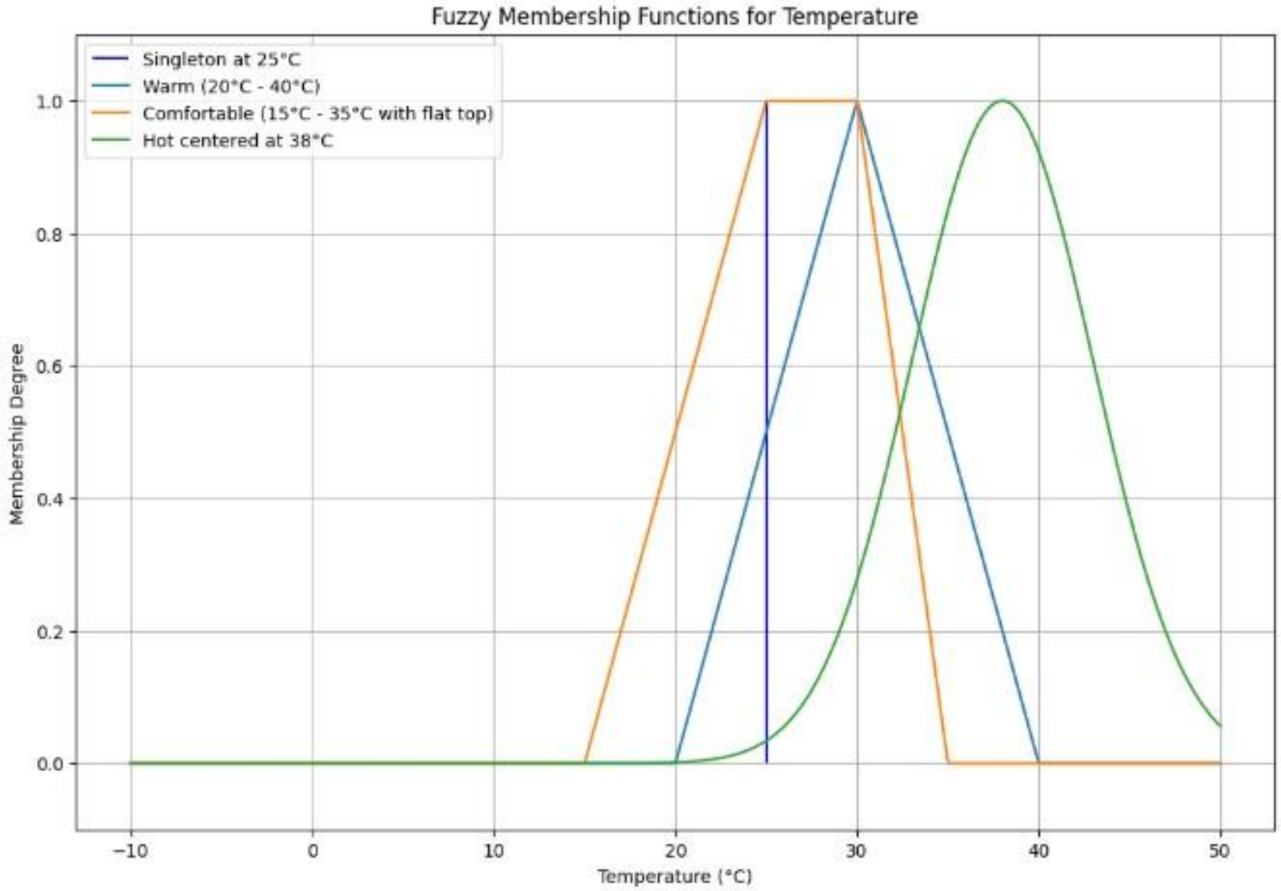
Example: Temperature Classification

Let’s use fuzzy membership functions to classify temperature:

```
# Membership Function Definitions
def singleton_mf(x, x0):
    return np.where(x == x0, 1, 0)
```

```


def triangular_mf(x, a, b, c):
    return np.maximum(np.minimum((x-a)/(b-a), (c-x)/(c-b)), 0)
 def trapezoidal_mf(x, a, b, c,
d):
    return np.maximum(np.minimum(np.minimum((x-a)/(b-a), 1), (d-x)/(d-c)),
0)

def gaussian_mf(x, c, sigma):
    return np.exp(-((x-c)**2) / (2*sigma**2))

# Example: Temperature Classification
x = np.linspace(-10, 50, 500)  # Temperature range from -10°C to 50°C

# Membership Functions for different fuzzy sets
singleton_temp = singleton_mf(x, 25)  # Singleton at 25°C
warm_temp = triangular_mf(x, 20, 30, 40)  # Warm (20°C - 40°C)
comfortable_temp = trapezoidal_mf(x, 15, 25, 30, 35)  # Comfortable (15°C
- 35°C)
hot_temp = gaussian_mf(x, 38, 5)  # Hot centered at 38°C

# Plot the membership functions
plt.figure(figsize=(12, 8))

# Use vlines for Singleton to show it as a vertical line plt.vlines(25,
0, 1, colors='blue', label="Singleton at 25°C",
linestyles='solid')

plt.plot(x, warm_temp, label="Warm (20°C - 40°C)")
plt.plot(x, comfortable_temp, label="Comfortable (15°C - 35°C with flat
top)")
plt.plot(x, hot_temp, label="Hot centered at 38°C")

plt.title('Fuzzy Membership Functions for Temperature')
plt.xlabel('Temperature (°C)')
plt.ylabel('Membership Degree')
plt.ylim(-0.1, 1.1)
plt.legend()
plt.grid(True) plt.show()
```

Output:



Conclusion:

We have successfully implemented and understood different **Fuzzy Membership Functions** (Singleton, Triangular, Trapezoidal, and Gaussian). These functions play a crucial role in fuzzy logic systems by representing vague and uncertain data with membership values. By using these functions, we can model real-world problems where precise categorization is not possible, such as classifying temperatures into "warm," "comfortable," or "hot."