Manav Jawrani 19

Aim - To implement Fuzzy Membership Functions in Python.

Theory -

What is fuzzy logic?

Fuzzy logic is a type of logic that allows for degrees of truth, rather than strict binary (true or false) values. It's designed to handle imprecise or uncertain information by assigning degrees of membership to different categories. This allows for more nuanced decision-making in situations where traditional binary logic might be inadequate.

Here's a simple **example**: Consider a thermostat controlling a room's temperature. In binary logic, the temperature is either "hot" or "cold" based on a fixed threshold. However, in reality, the room temperature can be warm but not excessively hot, or cool but not freezing cold. Fuzzy logic helps capture this by allowing terms like "warm," "slightly warm," "slightly cold," and "cold" to have varying degrees of truth.

For instance, if we use fuzzy logic to control the thermostat, the rules might look like this:

1. If the temperature is "very cold," then increase the heater's intensity significantly.
2. If the temperature is "cold," then increase the heater's intensity moderately.
3. If the temperature is "warm," then maintain the current heater intensity.
4. If the temperature is "hot," then decrease the heater's intensity moderately.
5. If the temperature is "very hot," then decrease the heater's intensity significantly.

This allows the thermostat to make more gradual and accurate adjustments based on the degree to which the temperature matches each category. So, if the room is slightly cold, the thermostat would increase the heater's intensity slightly, rather than abruptly switching it from off to full blast, as binary logic might do.

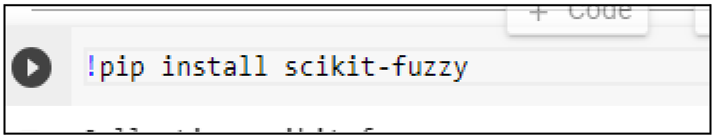
What is a membership function in fuzzy logic?

A membership function determines how much an input belongs to a fuzzy set. It's like assigning a degree of membership, from 0 to 1, to show how well the input fits a specific set's description. For example, if we have a "Tallness" fuzzy set with terms like "Short," "Average," and "Tall," and someone is 175 cm tall, their membership values might be:

Short: 0.2
Average: 0.7
Tall: 0.1
This indicates that their height fits the "Average" category more than the others.

Code -

Installing fuzzy library -



```
!pip install scikit-fuzzy
```

Code 1 - We have created three membership functions

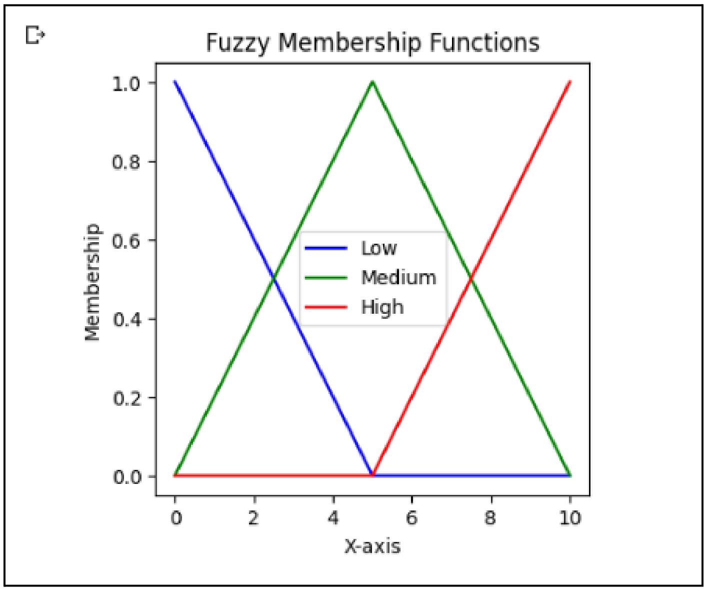
```
import numpy as np
import matplotlib.pyplot as plt
import skfuzzy as fuzz

# Create a range of values for the x-axis
x = np.arange(0, 11, 1)

# Create membership functions
low = fuzz.trimf(x, [0, 0, 5])
medium = fuzz.trimf(x, [0, 5, 10])
high = fuzz.trimf(x, [5, 10, 10])

# Plot the membership functions
plt.figure(figsize=(4, 4))
plt.plot(x, low, 'b', linewidth=1.5, label='Low')
plt.plot(x, medium, 'g', linewidth=1.5, label='Medium')
plt.plot(x, high, 'r', linewidth=1.5, label='High')
plt.title('Fuzzy Membership Functions')
plt.xlabel('X-axis')
plt.ylabel('Membership')
plt.legend()
plt.show()
```

Code 2 - We have created five membership functions

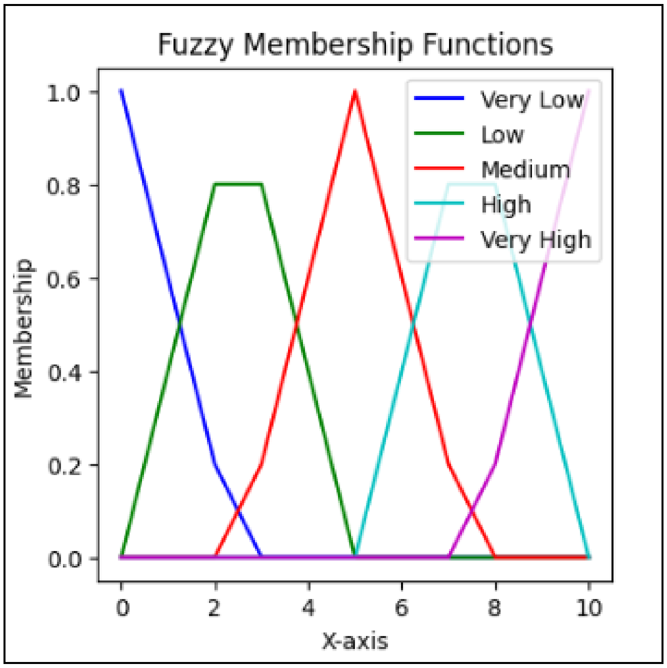
```
import numpy as np
import matplotlib.pyplot as plt
import skfuzzy as fuzz

# Create a range of values for the x-axis
x = np.arange(0, 11, 1)

# Create membership functions
very_low = fuzz.trimf(x, [0, 0, 2.5])
low = fuzz.trimf(x, [0, 2.5, 5])
medium = fuzz.trimf(x, [2.5, 5, 7.5])
high = fuzz.trimf(x, [5, 7.5, 10])
very_high = fuzz.trimf(x, [7.5, 10, 10])

# Plot the membership functions
plt.figure(figsize=(4, 4))
plt.plot(x, very_low, 'b', linewidth=1.5, label='Very Low')
plt.plot(x, low, 'g', linewidth=1.5, label='Low')
plt.plot(x, medium, 'r', linewidth=1.5, label='Medium')
plt.plot(x, high, 'c', linewidth=1.5, label='High')
plt.plot(x, very_high, 'm', linewidth=1.5, label='Very High')
plt.title('Fuzzy Membership Functions')
plt.xlabel('X-axis')
plt.ylabel('Membership')
plt.legend()
plt.show()
```

Output -

Output 1 -



Output 2 -



Conclusion - Thus, we can conclude that we have understood how to create Fuzzy Membership Functions in Python by three and five membership functions in two different codes.