

What Is Fuzzy Logic?

Fuzzy Logic is defined as a many-valued logic form which may have truth values of variables in any real number between 0 and 1. It is the handle concept of partial truth. In real life, we may come across a situation where we can't decide whether the statement is true or false. At that time, fuzzy logic offers very valuable flexibility for reasoning.

Fuzzy logic algorithm helps to solve a problem after considering all available data. Then it takes the best possible decision for the given the input. The FL method imitates the way of decision making in a human which consider all the possibilities between digital values T and F.

History of Fuzzy Logic Systems

Although, the concept of fuzzy logic had been studied since the 1920's. The term fuzzy logic was first used with 1965 by Lotfi Zadeh a professor of UC Berkeley in California. He observed that conventional computer logic was not capable of manipulating data representing subjective or unclear human ideas.

Fuzzy algorithm has been applied to various fields, from control theory to AI. It was designed to allow the computer to determine the distinctions among data which is neither true nor false. Something similar to the process of human reasoning. Like Little dark, Some brightness, etc.

Characteristics of Fuzzy Logic

Here, are some important characteristics of fuzzy logic:

- Flexible and easy to implement machine learning technique
- Helps you to mimic the logic of human thought
- Logic may have two values which represent two possible solutions
- Highly suitable method for uncertain or approximate reasoning
- Fuzzy logic views inference as a process of propagating elastic constraints
- Fuzzy logic allows you to build nonlinear functions of arbitrary complexity.
- Fuzzy logic should be built with the complete guidance of experts

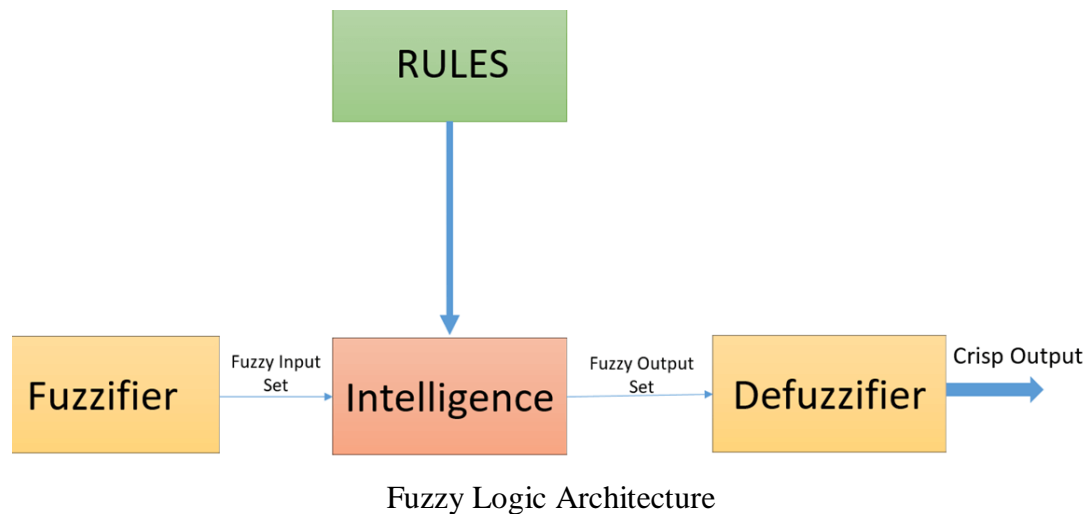
When not to use fuzzy logic

However, fuzzy logic is never a cure for all. Therefore, it is equally important to understand that where we should not use fuzzy logic.

Here, are certain situations when you better not use Fuzzy Logic:

- If you don't find it convenient to map an input space to an output space
- Fuzzy logic should not be used when you can use common sense
- Many controllers can do the fine job without the use of fuzzy logic

Fuzzy Logic Architecture



Fuzzy Logic architecture has four main parts as shown in the diagram:

Rule Base:

It contains all the rules and the if-then conditions offered by the experts to control the decision-making system. The recent update in fuzzy theory provides various methods for the design and tuning of fuzzy controllers. This updates significantly reduce the number of the fuzzy set of rules.

1. Fuzzification:

Fuzzification step helps to convert inputs. It allows you to convert, crisp numbers into fuzzy sets. Crisp inputs measured by sensors and passed into the control system for further processing. Like Room temperature, pressure, etc.

2. Inference Engine:

It helps you to determines the degree of match between fuzzy input and the rules. Based on the % match, it determines which rules need implment according to the given input field. After this, the applied rules are combined to develop the control actions.

3. Defuzzification:

At last the Defuzzification process is performed to convert the fuzzy sets into a crisp value. There are many types of techniques available, so you need to select it which is best suited when it is used with an expert system.

Advantages of Fuzzy Logic System

- The structure of Fuzzy Logic Systems is easy and understandable
- Fuzzy logic is widely used for commercial and practical purposes
- Fuzzy logic in AI helps you to control machines and consumer products
- It may not offer accurate reasoning, but the only acceptable reasoning
- Fuzzy logic in Data Mining helps you to deal with the uncertainty in engineering
- Mostly robust as no precise inputs required

- It can be programmed to in the situation when feedback sensor stops working
- It can easily be modified to improve or alter system performance
- inexpensive sensors can be used which helps you to keep the overall system cost and complexity low
- It provides a most effective solution to complex issues

Disadvantages of Fuzzy Logic Systems

- Fuzzy logic is not always accurate, so The results are perceived based on assumption, so it may not be widely accepted.
- Fuzzy systems don't have the capability of machine learning as-well-as neural network type pattern recognition
- Validation and Verification of a fuzzy knowledge-based system needs extensive testing with hardware
- Setting exact, fuzzy rules and, membership functions is a difficult task
- Some fuzzy time logic is confused with probability theory and the terms

Application:

Smart Washing Machine:

Consider an individual doing the laundry for the first time and is not familiar with the type of fabric or detergent to be used. In order to solve the laundry problem, he would seek input from professionals to save the effort. Consider a solution combining the inputs from detergent maker, fabric maker and professional laundry servicemen into an implementable model that would facilitate the unaware person doing laundry for the first time.

Smart Washing Machine:

FUZZY LOGIC CONTROLLER (FLC) FOR SMART WASHING MACHINE (SWM)

The primary concept in designing a Fuzzy Logic Controller (FLC) lies behind the information gathered from various sources of experience or experts (Laundry-Load Characteristics).

Parameters:

1- Inputs:

- Type of tissue {Silk, wool, cotton}
- Type of dirt { non-greasy, medium greasy, greasy}
- The degree of dirt {small, average, big}

2- Output:

- Washing time { Too short, Short, Medium, Long, Too Long}

Rule Base:

N	IF the tissue is of type:	AND the dirt is:	AND the degree is :	THEN the washing time is:
1	Silk	non-greasy	small	Too Short
2	Silk	non-greasy	average	Short
3	Silk	non-greasy	big	Medium
4	Silk	medium greasy	small	Medium
5	Silk	medium greasy	average	Long
6	Silk	medium greasy	big	Long
7	Silk	greasy	small	Medium
8	Silk	greasy	average	Long
9	Silk	greasy	big	Too Long
10	Wool	non-greasy	small	Short
11	Wool	non-greasy	average	Medium
12	Wool	non-greasy	big	Long
13	Wool	medium greasy	small	Medium
14	Wool	medium greasy	average	Medium
15	Wool	medium greasy	big	Long
16	Wool	greasy	small	Long
17	Wool	greasy	average	Long
18	Wool	greasy	big	Too Long
19	Cotton	non-greasy	small	Short
20	Cotton	non-greasy	average	Medium
21	Cotton	non-greasy	big	Long
22	Cotton	medium greasy	small	Medium
23	Cotton	medium greasy	average	Long
24	Cotton	medium greasy	big	Too Long
25	Cotton	greasy	small	Long
26	Cotton	greasy	average	Long
27	Cotton	greasy	big	Too Long

Program the system using Python:

Creating the Washing Machine Controller Using the "skfuzzy control API"

We can use the skfuzzy control system API to model this. First, we need to import the needed libraries:

```
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl

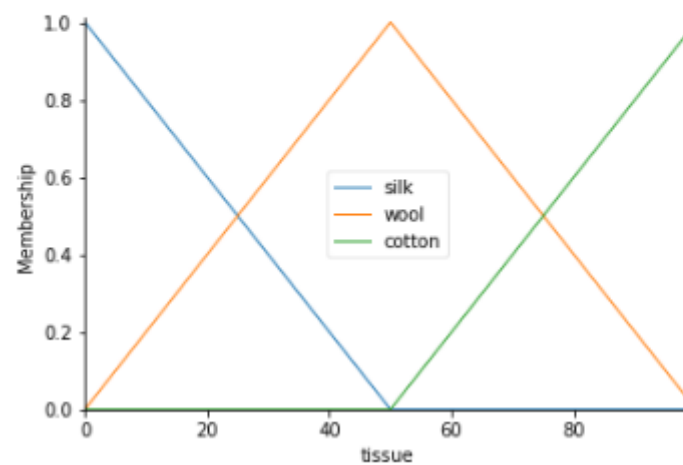
#"scikit-fuzzy"(skfuzzy) is an fuzzy logic Python package that works with numpy arrays.
```

Then, we need to define fuzzy variables:

```
tissue = ctrl.Antecedent(np.arange(0, 100, 1), 'tissue')
dirt = ctrl.Antecedent(np.arange(0, 100, 1), 'dirt')
degree = ctrl.Antecedent(np.arange(0, 100, 1), 'degree')
time = ctrl.Consequent(np.arange(0, 100, 1), 'washing time')
```

```
tissue['silk'] = fuzz.trimf(tissue.universe, [0, 0, 50])
tissue['wool'] = fuzz.trimf(tissue.universe, [0, 50, 100])
tissue['cotton'] = fuzz.trimf(tissue.universe, [50, 100, 100])
```

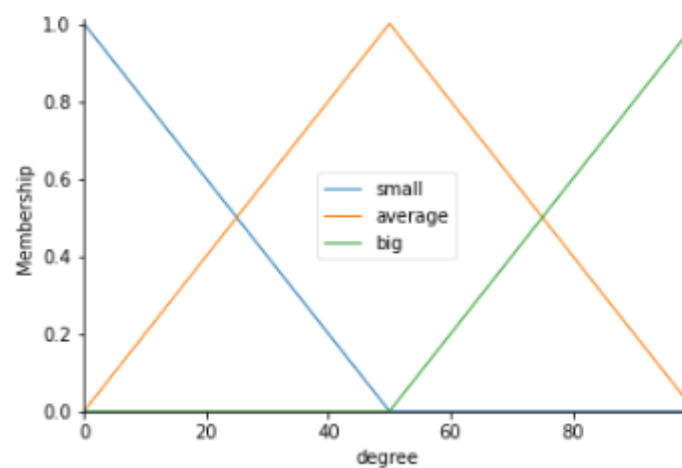
```
tissue.view()
```



Membership of tissue

```
degree['small'] = fuzz.trimf(degree.universe, [0, 0, 50])
degree['average'] = fuzz.trimf(degree.universe, [0, 50, 100])
degree['big'] = fuzz.trimf(degree.universe, [50, 100, 100])
```

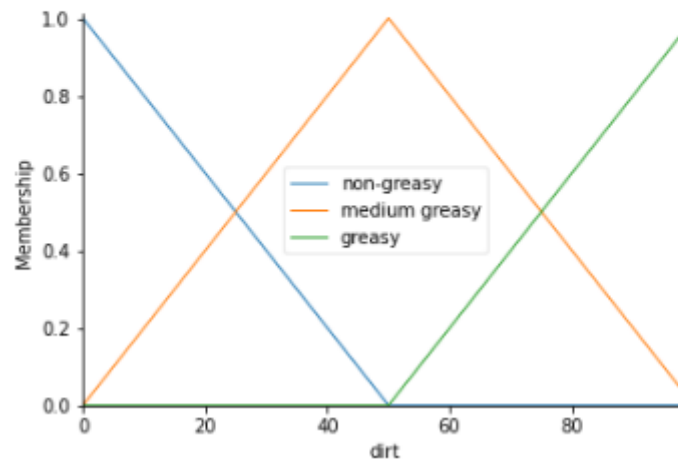
```
degree.view()
```



Membership of degree of dirt

```
dirt['non-greasy'] = fuzz.trimf(dirt.universe, [0, 0, 50])
dirt['medium greasy'] = fuzz.trimf(dirt.universe, [0, 50, 100])
dirt['greasy'] = fuzz.trimf(dirt.universe, [50, 100, 100])
```

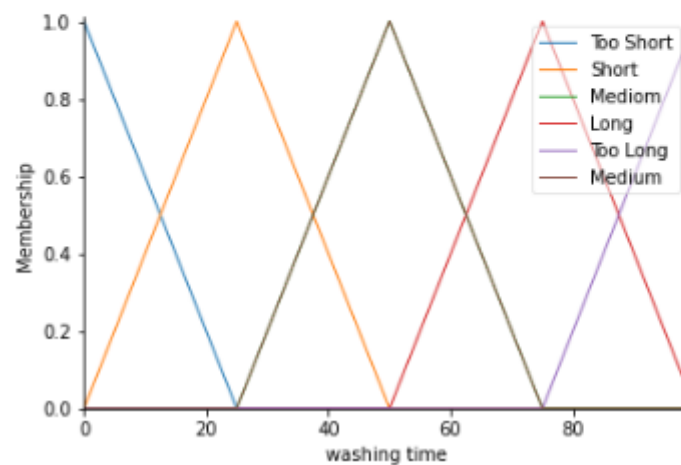
```
dirt.view()
```



Membership of dirt

```
time['Too Short'] = fuzz.trimf(time.universe, [0, 0, 25])
time['Short'] = fuzz.trimf(time.universe, [0, 25, 50])
time['Medium'] = fuzz.trimf(time.universe, [25, 50, 75])
time['Long'] = fuzz.trimf(time.universe, [50, 75, 100])
time['Too Long'] = fuzz.trimf(time.universe, [75, 100, 100])
```

```
time.view()
```



Membership of washing time

Fuzzy rules

Now, to make these triangles useful, we define the *fuzzy relationship* between input and output variables.

Most people would agree on these rules, but the rules are fuzzy. Mapping the imprecise rules into a defined, actionable tip is a challenge. This is the kind of task at which fuzzy logic excels.

```
rule1 = ctrl.Rule(tissue['silk'] & dirt['non-greasy'] & degree['small'],time['Too Short'] )
rule2 = ctrl.Rule(tissue['silk'] & dirt['non-greasy'] & degree['average'],time['Short'] )
rule3 = ctrl.Rule(tissue['silk'] & dirt['non-greasy'] & degree['big'],time['Medium'] )
rule4 = ctrl.Rule(tissue['silk'] & dirt['medium greasy'] & degree['small'],time['Medium'] )
rule5 = ctrl.Rule(tissue['silk'] & dirt['medium greasy'] & degree['average'],time['Long'] )
rule6 = ctrl.Rule(tissue['silk'] & dirt['medium greasy'] & degree['big'],time['Long'] )
rule7 = ctrl.Rule(tissue['silk'] & dirt['greasy'] & degree['small'],time['Medium'] )
rule8 = ctrl.Rule(tissue['silk'] & dirt['greasy'] & degree['average'],time['Long'] )
rule9 = ctrl.Rule(tissue['silk'] & dirt['greasy'] & degree['big'],time['Too Long'] )
rule10 = ctrl.Rule(tissue['wool'] & dirt['non-greasy'] & degree['small'],time['Short'] )
rule11 = ctrl.Rule(tissue['wool'] & dirt['non-greasy'] & degree['average'],time['Medium'] )
rule12 = ctrl.Rule(tissue['wool'] & dirt['non-greasy'] & degree['big'],time['Long'] )
rule13 = ctrl.Rule(tissue['wool'] & dirt['medium greasy'] & degree['small'],time['Medium'] )
rule14 = ctrl.Rule(tissue['wool'] & dirt['medium greasy'] & degree['average'],time['Medium'] )
rule15 = ctrl.Rule(tissue['wool'] & dirt['medium greasy'] & degree['big'],time['Long'] )
rule16 = ctrl.Rule(tissue['wool'] & dirt['greasy'] & degree['small'],time['Long'] )
rule17 = ctrl.Rule(tissue['wool'] & dirt['greasy'] & degree['average'],time['Long'] )
rule18 = ctrl.Rule(tissue['wool'] & dirt['greasy'] & degree['big'],time['Too Long'] )
rule19 = ctrl.Rule(tissue['cotton'] & dirt['non-greasy'] & degree['small'],time['Short'] )
rule20 = ctrl.Rule(tissue['cotton'] & dirt['non-greasy'] & degree['average'],time['Medium'] )
rule21 = ctrl.Rule(tissue['cotton'] & dirt['non-greasy'] & degree['big'],time['Long'] )
rule22 = ctrl.Rule(tissue['cotton'] & dirt['medium greasy'] & degree['small'],time['Medium'] )
rule23 = ctrl.Rule(tissue['cotton'] & dirt['medium greasy'] & degree['average'],time['Long'] )
rule24 = ctrl.Rule(tissue['cotton'] & dirt['medium greasy'] & degree['big'],time['Too Long'] )
rule25 = ctrl.Rule(tissue['cotton'] & dirt['greasy'] & degree['small'],time['Long'] )
rule26 = ctrl.Rule(tissue['cotton'] & dirt['greasy'] & degree['average'],time['Long'] )
rule27 = ctrl.Rule(tissue['cotton'] & dirt['greasy'] & degree['big'],time['Too Long'] )
```

Control System Creation and Simulation

Now that we have our rules defined, we can simply create a control system via:

```
rules = ctrl.ControlSystem([rule1, rule2, rule3, rule4,
                             rule5, rule6, rule7, rule8,
                             rule9, rule10, rule11, rule12,
                             rule13, rule14, rule15, rule16,
                             rule17, rule18, rule19, rule20,
                             rule21, rule22, rule23, rule24,
                             rule25, rule26, rule27])
```

Despite the lengthy ruleset, the new fuzzy control system framework will execute in milliseconds. Next we add these rules to a new `ControlSystem` and define a `ControlSystemSimulation` to run it.

```
f_logic = ctrl.ControlSystemSimulation(rules)
```

We can now simulate our control system by simply specifying random inputs and calling the `compute` method.

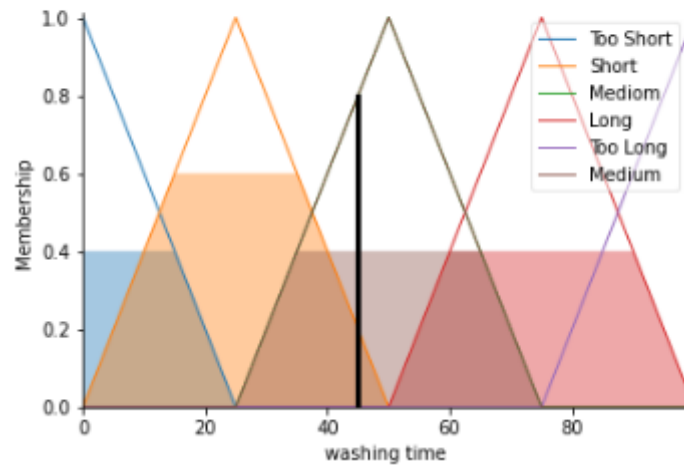
```
f_logic.input['tissue'] = 10
f_logic.input['dirt'] = 20
f_logic.input['degree'] = 30

f_logic.compute()
```

Once computed, we can view the result as well as visualize it.

```
print (f_logic.output['washing time'])
time.view(sim=f_logic)
```

44.897161470451444



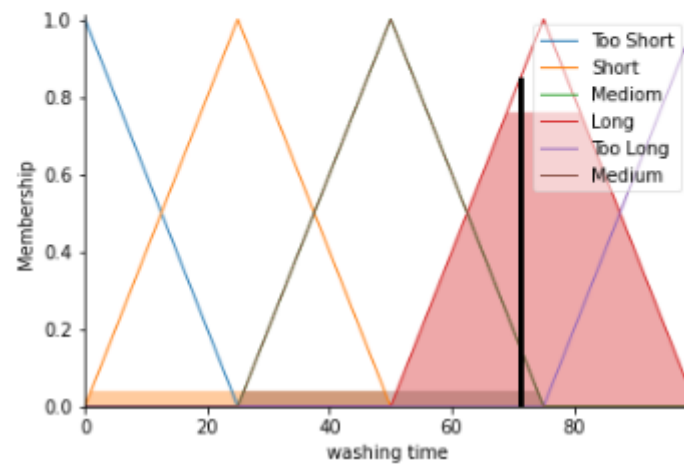
The resulting suggested the washing time is **44.89%**.

Another set of inputs:

```
f_logic.input['tissue'] = 48
f_logic.input['dirt'] = 12
f_logic.input['degree'] = 98
```

```
f_logic.compute()
print (f_logic.output['washing time'])
time.view(sim=f_logic)
```

71.10467240929262



The resulting suggested the washing time is **71.10%**.

The power of fuzzy systems is allowing complicated, intuitive behavior based on a sparse system of rules with minimal overhead. Note our membership function universes were coarse, only defined at the integers, but `fuzz.interp_membership` allowed the effective resolution to increase on demand. This system can respond to arbitrarily small changes in inputs, and the processing burden is minimal.

Conclusion:

The commercial applications of fuzzy has made its mark over the past few decades because of the theoretically infinite range of control over a particular application. In this paper, the precision in wash time will not only economize energy resources (including electricity & water) but also benefit the user to save finances in commercial laundry solutions offered in the market.

The areas of application of fuzzy logic controllers have more dynamic range as compared to the conventional PID controller. However, there is an importance to highlight that one particular fuzzified solution may not be applicable to all users so there is a dire need to standardize the domain of fuzzified solution.