

A lógica fuzzy é um sistema de lógica multivalorado que permite valores intermediários a serem definidos entre avaliações convencionais como verdadeiro/falso, sim/não, alto/baixo, etc. Ela foi concebida por Lotfi A. Zadeh da Universidade da Califórnia em 1965 como uma forma mais próxima da maneira como os humanos realmente pensam.

Conceitos Fundamentais

- **Conjuntos Fuzzy:**
 - Ao contrário dos conjuntos clássicos, onde um elemento pertence ou não a um conjunto, os conjuntos fuzzy permitem que um elemento tenha um grau de pertinência entre 0 e 1.
 - Esse grau de pertinência representa o quão bem um elemento se encaixa em um conjunto fuzzy.
- **Variáveis Linguísticas:**
 - São variáveis cujos valores são palavras ou frases em linguagem natural, como "quente", "frio", "muito alto", etc.
 - Cada valor linguístico é representado por um conjunto fuzzy.
- **Funções de Pertinência:**
 - Definem o grau de pertinência de um elemento a um conjunto fuzzy.
 - Existem várias formas de funções de pertinência, como triangular, trapezoidal, gaussiana, etc.
- **Regras Fuzzy:**
 - São regras do tipo "SE-ENTÃO" que relacionam variáveis linguísticas de entrada a variáveis linguísticas de saída.
 - Exemplo: "SE a temperatura é quente, ENTÃO a velocidade do ventilador é alta".
- **Inferência Fuzzy:**
 - É o processo de aplicar regras fuzzy às entradas para gerar uma saída fuzzy.
 - Existem vários métodos de inferência fuzzy, como o método de Mamdani e o método de Takagi-Sugeno.
- **Defuzzificação:**

- É o processo de converter uma saída fuzzy em um valor numérico único.
- Existem vários métodos de defuzzificação, como o centro de gravidade e o máximo médio.

Aplicações

A lógica fuzzy é usada em uma ampla variedade de aplicações, incluindo:

- **Sistemas de Controle:**
 - Controladores de temperatura, sistemas de freios antibloqueio (ABS), sistemas de controle de tráfego, etc.
- **Inteligência Artificial:**
 - Sistemas de reconhecimento de padrões, sistemas de diagnóstico médico, sistemas de recomendação, etc.
- **Eletrônica de Consumo:**
 - Máquinas de lavar roupa, câmeras digitais, fornos de micro-ondas, etc.
- **Finanças:**
 - Análise de risco de crédito, previsão de mercado de ações, etc.

Vantagens

- Capacidade de lidar com incerteza e imprecisão.
- Flexibilidade para modelar sistemas complexos.
- Facilidade de implementação e manutenção.
- Capacidade de incorporar conhecimento especializado.

Desvantagens

- Dificuldade em definir funções de pertinência e regras fuzzy.
- Requer conhecimento especializado para projetar sistemas fuzzy.
- Pode ser computacionalmente caro para sistemas complexos.

```
import numpy as np
```

```
import skfuzzy as fuzz
```

```
from skfuzzy import control as ctrl
```

```
# Variáveis de entrada
```

```
temperatura = ctrl.Antecedent(np.arange(0, 41, 1), 'temperatura')
```

```
umidade = ctrl.Antecedent(np.arange(0, 101, 1), 'umidade')
```

```
# Variável de saída
```

```
velocidade_ventilador = ctrl.Consequent(np.arange(0, 101, 1),  
'velocidade_ventilador')
```

```
# Funções de pertinência
```

```
temperatura['frio'] = fuzz.trimf(temperatura.universe, [0, 0, 20])
```

```
temperatura['médio'] = fuzz.trimf(temperatura.universe, [10, 25, 35])
```

```
temperatura['quente'] = fuzz.trimf(temperatura.universe, [30, 40, 40])
```

```
umidade['seca'] = fuzz.trimf(umidade.universe, [0, 0, 50])
```

```
umidade['úmida'] = fuzz.trimf(umidade.universe, [25, 75, 100])
```

```
velocidade_ventilador['baixa'] = fuzz.trimf(velocidade_ventilador.universe, [0,  
0, 50])
```

```
velocidade_ventilador['alta'] = fuzz.trimf(velocidade_ventilador.universe, [50,  
100, 100])
```

```
# Regras fuzzy
```

```
regra1 = ctrl.Rule(temperatura['frio'] | umidade['seca'],  
velocidade_ventilador['baixa'])
```

```
regra2 = ctrl.Rule(temperatura['médio'], velocidade_ventilador['alta'])
```

```
regra3 = ctrl.Rule(temperatura['quente'] | umidade['úmida'],  
velocidade_ventilador['alta'])
```

```
# Sistema de controle
```

```
sistema_controle = ctrl.ControlSystem([regra1, regra2, regra3])  
simulador = ctrl.ControlSystemSimulation(sistema_controle)
```

```
# Entradas
```

```
simulador.input['temperatura'] = 30
```

```
simulador.input['umidade'] = 80
```

```
# Computação
```

```
simulador.compute()
```

```
# Saída
```

```
print(simulador.output['velocidade_ventilador'])
```

Este código demonstra um sistema de controle fuzzy simples para a velocidade de um ventilador com base na temperatura e umidade.