

Computação Inteligente para a Internet das Coisas

Mestrado Engenharia Electrónica

Projeto 2 – Fuzzy Systems

Autores:

João Bernardo Silva (97004) Victor Macedo (1105095) joao.garces.silva@tecnico.ulisboa.pt
 victor.macedo@tecnico.ulisboa.pt

Grupo 4

Conteúdo

1	Introdução	2			
2	Metodologia2.1 Parâmetros de entrada2.2 Rede Fuzzy2.3 Regras2.4 Implementação	4			
3	Resultados	7			
4	Conclusão	8			
\mathbf{A}_{1}	pêndices	ę			
A	Código main	ę			
В	B Código TestMe				

1 Introdução

O crescente avanço e a crescente capacidade computacional dos dispositivos IoT resultaram em volumes sem precedentes de dados. A quantidade de dados gerados por dispositivos no nosso mundo é maior do que nunca. No entanto, grande parte desses dados do IoT não é explorada ou utilizada, uma vez que o envio de todos esses dados gerados pelos dispositivos para um centro de dados centralizado ou para a nuvem causa problemas de largura de banda e latência.

O objetivo deste projeto é através de um sistema Fuzzy, criar um sistema inteligente de tarefas de computação Edge que não comprometa diversos fatores, como latência, congestão e largura de banda.

2 Metodologia

2.1 Parâmetros de entrada

O problema apresentado possui 6 possíveis entradas para o sistema fuzzy [1], porém é importante selecionar quais são realmente importantes, pois o numero de entrada aumenta a quantidade de regras de forma exponencial. Idealmente a escolha dos parâmetros deve ser feita em conjunto com um especialista do assunto, na falta de um deve-se utilizar o bom senso para o mesmo. Os parâmetros possíveis são:

- Memory usage (%)
- Processor load (%) Note: Processor load is not the same as CLP
- Input network throughput
- Output network throughput
- Available output bandwidth
- Latency (mS)

Após analisar as opções acimas decidiu-se não utilizar os parâmetros Input network throughput e Available output bandwidth.

2.2 Rede Fuzzy

Como explicado na secção 2.1 a rede possui 4 parâmetros de entradas, considerando 3 termos linguísticos em cada seriam necessários $5^4 = 625$ regras para o sistema. Com objetivo de evitar a o crescimento exponencial o sistema deve ser dividido em 2 grupos menores, o primeiro chamado load que deve receber Memory usage e Processor load e Network que recebe Output network throughput e Latency.

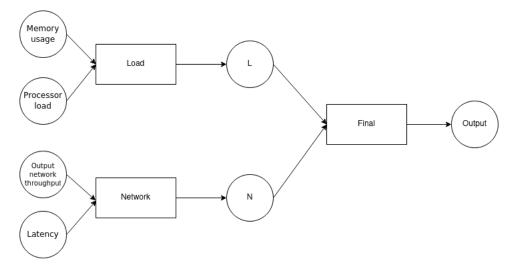


Figura 1: Diagrama da rede

Dessa forma são necessário apenas $5^2+5^2+5^2=75$ regras, uma redução superior a um oitavo no numero de regras.

2.3 Regras

Para as construções das regras do sistema foram utilizados 5 termos linguísticos em cada entrada resultando na seguinte seguem a tabelas 1, 2, 3

				Process		
Sis	stema Load	Very low	Low	Normal	High	Very High
	Very low	Very High	Very High	High	High	High
ıry	Low	Very High	Very High	High	High	Low
Memory	Normal	High	High	Normal	Low	Low
m Me	High	High	High	Low	Low	Very low
	Very High	High	Low	Low	Very low	Very low

Tabela 1: Regras do primeiro sistema Fuzzy

		Output network throughput				
Sist	tema Network	Very low	Low	Normal	High	Very High
	Very low	Low	Low	Low	Very low	Very low
cy	Low	High	Low	Low	Low	Very low
Latency	Normal	High	High	Normal	Low	Low
La	High	Very High	High	High	Low	Low
	Very High	Very High	Very High	High	High	Low

Tabela 2: Regras do segundo sistema Fuzzy

		Network				
Sis	stema Final	Very low	Low	Normal	High	Very High
	Very low	Normal	High	Very High	Very High	Very High
-	Low	Normal	Normal	High	Very High	Very High
Load	Normal	Very low	Normal	Normal	High	Very High
	High	Very low	Low	Low	Low	High
	Very High	Very low	Very low	Very low	Very low	Very low

Tabela 3: Regras do terceiro sistema Fuzzy

Os termos linguisticos foram definidos usando funções triangulares, como apresentado nas figuras 2, 3 e 4

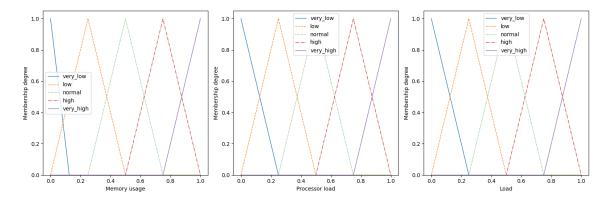


Figura 2: Regras termos linguísticos load

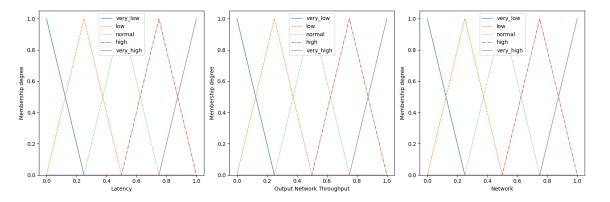


Figura 3: Regras termos linguísticos network

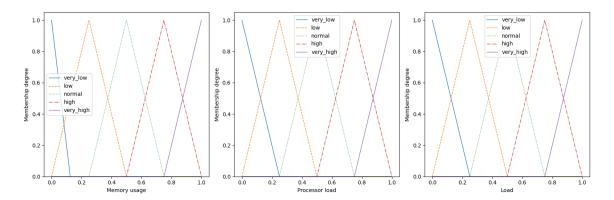


Figura 4: Regras termos linguísticos final

Por fim os sistemas descritos acima resultaram nos gráfico 3D das figuras 5, 6 e 7.

Fuzzy Load

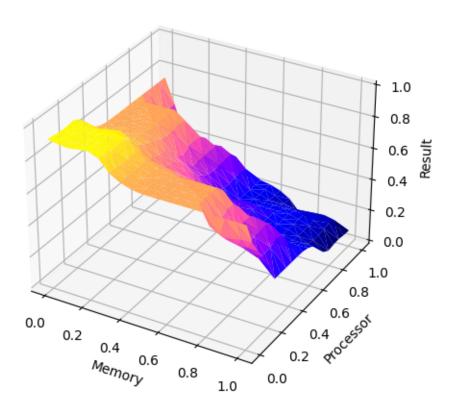


Figura 5: Two Input / Single Output Load

Fuzzy Network

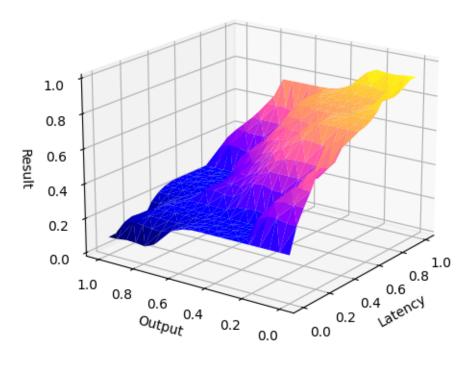


Figura 6: Two Input / Single Output Network

Fuzzy final

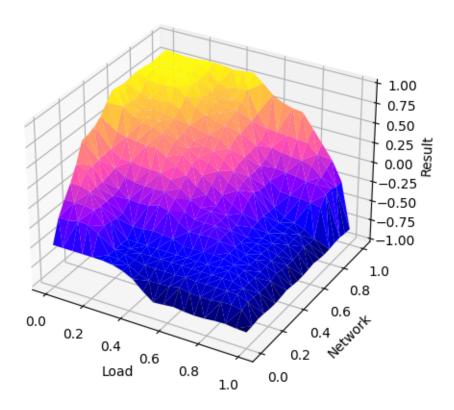


Figura 7: Two Input / Single Output finalç

2.4 Implementação

Para implementar o sistema descrito acima deve ser utilizado a linguagem de programação Python ([3]) utilizando o package simpful ([2]), o qual permite definir os termos linguísticos, regras e o sistema fuzzy de forma simples.

3 Resultados

Para avaliar o desempenho do sistema foi utilizado um dataset exemplo, os resultados obtido podem ser observado na tabela 4

Esperados	Obtidos
0.73	0.028
-0.82	-0.250
0.12	-0.050
-0.85	0.116
0.5	0.000
0.8	0.599
-0.31	-0.413
-0.65	-0.413
0.85	-0.254
0.85	-0.422

Tabela 4: Resultados utilizando o dataset

Durante a avaliação dos resultados foi necessário ajustar os parâmetros, como regras e termos linguísticos, do sistema diversas vezes para obter melhores respostas, dessas forma fica evidente que o auxilio de um profissional da área é capaz de agilizar e melhorar os resultados do sistema.

4 Conclusão

A construção desta rede Fuzzy apresentou desafios significativos, uma vez que os resultados obtidos variam consideravelmente em relação às expectativas estabelecidas. A implementação de tais redes é um processo complexo que requer experiência para alcançar resultados satisfatórios.

Apesar dos valores numéricos obtidos e esperados não serem tão próximos ao considerar o termo linguístico tem-se um resultado satisfatório para a maioria dos casos.

Referências

- [1] Joao Paulo Carvalho. Slides Computação Inteligente para a Internet das Coisas. 2023.
- [2] Cazzaniga P. Kaymak U. Besozzi D. Nobile M.S. Spolaor S., Fuchs C. Simpful: a user-friendly python library for fuzzy logic. *International Journal of Computational Intelligence Systems*.
- [3] Guido Van Rossum and Fred L Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

Apêndices

A Código main

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pylab as plt
5 from simpful import *
6 from sklearn.neural_network import MLPRegressor
 #df = pd.read_csv("/Users/mac/Documents/GitHub/CI4Iot/Projeto_2/
     Project2_SampleData.csv")
11 df = pd.read_csv("Projeto_2/Project2_SampleData.csv")
13 # Rede Fuzzy, input memory usage and processor load, output Load
14 # Create a fuzzy system object
15 FS_L = FuzzySystem()
17 # Define fuzzy sets and linguistic variables
18 S_1 = FuzzySet(function=Triangular_MF(a=0, b=0, c=0.125), term="very_low")
19 S_2 = FuzzySet(function=Triangular_MF(a=0.0, b=0.25, c=0.5), term="low")
20 S_3 = FuzzySet(function=Triangular_MF(a=0.25, b=0.5, c=0.75), term="normal")
21 S_4 = FuzzySet(function=Triangular_MF(a=0.5, b=0.75, c=1), term="high")
S_5 = FuzzySet(function=Triangular_MF(a=0.75, b=1, c=1), term="very_high")
23 S_LV = LinguisticVariable([S_1, S_2, S_3,S_4,S_5], concept="Memory usage",
     universe_of_discourse=[0,1])
FS_L.add_linguistic_variable("Memory",S_LV)
27 F_1 = FuzzySet(function=Triangular_MF(a=0, b=0, c=0.25), term="very_low")
28 F_2 = FuzzySet(function=Triangular_MF(a=0.0, b=0.25, c=0.5), term="low")
29 F_3 = FuzzySet(function=Triangular_MF(a=0.25, b=0.5, c=0.75), term="normal")
30 F_4 = FuzzySet(function=Triangular_MF(a=0.5, b=0.75, c=1), term="high")
31 F_5 = FuzzySet(function=Triangular_MF(a=0.75, b=1, c=1), term="very_high")
32 F_LV = LinguisticVariable([F_1, F_2,F_3,F_4,F_5], concept="Processor load",
     universe_of_discourse=[0,1])
33 FS_L.add_linguistic_variable("Processor", F_LV)
35 # Define output fuzzy sets and linguistic variable
36 T_1 = FuzzySet(function=Triangular_MF(a=0, b=0, c=0.25), term="very_low")
37 T_2 = FuzzySet(function=Triangular_MF(a=0.0, b=0.25, c=0.5), term="low")
38 T_3 = FuzzySet(function=Triangular_MF(a=0.25, b=0.5, c=0.75), term="normal")
T_4 = FuzzySet(function=Triangular_MF(a=0.5, b=0.75, c=1), term="high")
40 T_5 = FuzzySet(function=Triangular_MF(a=0.75, b=1, c=1), term="very_high")
T_LV = LinguisticVariable([T_1, T_2, T_3, T_4, T_5], universe_of_discourse
     =[0,1]
42 FS_L.add_linguistic_variable("Load", T_LV)
#FS_L.produce_figure()
45 # Define fuzzy rules
```

```
46 R_L1 = "IF (Memory IS very_low) AND (Processor IS very_low) THEN (Load IS
     very_high)"
47 R_L2 = "IF (Memory IS very_low) AND (Processor IS low) THEN (Load IS
     very_high)"
48 R_L3 = "IF (Memory IS very_low) AND (Processor IS normal) THEN (Load IS high
     ) "
49 R_L4 = "IF (Memory IS very_low) AND (Processor IS high) THEN (Load IS high)"
50 R_L5 = "IF (Memory IS very_low) AND (Processor IS very_high) THEN (Load IS
     high)"
51 R_L6 = "IF (Memory IS low) AND (Processor IS very_low) THEN (Load IS
     very_high)"
52 R_L7 = "IF (Memory IS low) AND (Processor IS low) THEN (Load IS very_high)"
8 R_L8 = "IF (Memory IS low) AND (Processor IS normal) THEN (Load IS high)"
54 R_L9 = "IF (Memory IS low) AND (Processor IS high) THEN (Load IS high)"
55 R_L10 = "IF (Memory IS low) AND (Processor IS very_high) THEN (Load IS low)"
56 R_L11 = "IF (Memory IS normal) AND (Processor IS very_low) THEN (Load IS
     high)"
57 R_L12 = "IF (Memory IS normal) AND (Processor IS low) THEN (Load IS high)"
58 R_L13 = "IF (Memory IS normal) AND (Processor IS normal) THEN (Load IS
59 R_L14 = "IF (Memory IS normal) AND (Processor IS high) THEN (Load IS low)"
60 R_L15 = "IF (Memory IS normal) AND (Processor IS very_high) THEN (Load IS
     low)"
61 R_L16 = "IF (Memory IS high) AND (Processor IS very_low) THEN (Load IS high)
62 R_L17 = "IF (Memory IS high) AND (Processor IS low) THEN (Load IS high)"
63 R_L18 = "IF (Memory IS high) AND (Processor IS normal) THEN (Load IS low)"
64 R_L19 = "IF (Memory IS high) AND (Processor IS high) THEN (Load IS low)"
65 R_L20 = "IF (Memory IS high) AND (Processor IS very_high) THEN (Load IS
    very_low)"
66 R_L21 = "IF (Memory IS very_high) AND (Processor IS very_low) THEN (Load IS
     high)"
67 R_L22 = "IF (Memory IS very_high) AND (Processor IS low) THEN (Load IS low)"
68 R_L23 = "IF (Memory IS very_high) AND (Processor IS normal) THEN (Load IS
69 R_L24 = "IF (Memory IS very_high) AND (Processor IS high) THEN (Load IS
     very_low)"
70 R_L25 = "IF (Memory IS very_high) AND (Processor IS very_high) THEN (Load IS
      very_low)"
71 FS_L.add_rules([ R_L1, R_L2, R_L3, R_L4, R_L5, R_L6, R_L7, R_L8, R_L9, R_L10
     , R_L11, R_L12, R_L13, R_L14, R_L15, R_L16, R_L17, R_L18, R_L19, R_L20,
     R_L21, R_L22, R_L23, R_L24, R_L25])
 ########### FIM Da REDE FS_L
                                  73
74
75
76 FS_N = FuzzySystem()
77 ## REDE NETWORK
78 L_1 = FuzzySet(function=Triangular_MF(a=0, b=0, c=0.25), term="very_low")
79 L_2 = FuzzySet(function=Triangular_MF(a=0.0, b=0.25, c=0.5), term="low")
80 L_3 = FuzzySet(function=Triangular_MF(a=0.25, b=0.5, c=0.75), term="normal")
81 L_4 = FuzzySet(function=Triangular_MF(a=0.5, b=0.75, c=1), term="high")
82 L_5 = FuzzySet(function=Triangular_MF(a=0.75, b=1, c=1), term="very_high")
83 L_LV = LinguisticVariable([L_1, L_2, L_3,L_4,L_5], concept="Latency",
  universe_of_discourse=[0, 1])
```

```
84 FS_N.add_linguistic_variable("Latency", L_LV)
86 O_1 = FuzzySet(function=Triangular_MF(a=0, b=0, c=0.25), term="very_low")
87 O_2 = FuzzySet(function=Triangular_MF(a=0.0, b=0.25, c=0.5), term="low")
88 O_3 = FuzzySet(function=Triangular_MF(a=0.25, b=0.5, c=0.75), term="normal")
89 O_4 = FuzzySet(function=Triangular_MF(a=0.5, b=0.75, c=1), term="high")
0_5 = FuzzySet(function=Triangular_MF(a=0.75, b=1, c=1), term="very_high")
0_LV = LinguisticVariable([0_1, 0_2, 0_3,0_4,0_5], concept="Output Network
     Throughput", universe_of_discourse=[0, 1])
92 FS_N.add_linguistic_variable("Output", O_LV)
94 # Define output fuzzy sets and linguistic variable
N_1 = FuzzySet(function=Triangular_MF(a=0, b=0, c=0.25), term="very_low")
N_2 = FuzzySet(function=Triangular_MF(a=0.0, b=0.25, c=0.5), term="low")
97 N_3 = FuzzySet(function=Triangular_MF(a=0.25, b=0.5, c=0.75), term="normal")
98 N_4 = FuzzySet(function=Triangular_MF(a=0.5, b=0.75, c=1), term="high")
99 N_5 = FuzzySet(function=Triangular_MF(a=0.75, b=1, c=1), term="very_high")
100 N_LV = LinguisticVariable([N_1, N_2, N_3, N_4, N_5], universe_of_discourse
     =[0, 1]
101 FS_N.add_linguistic_variable("Network", N_LV)
#FS_N.produce_figure()
103 # Define fuzzy rules
104
105 R_N1 = "IF (Output IS very_low) AND (Latency IS very_low) THEN (Network IS
     low)"
106 R_N2 = "IF (Output IS very_low) AND (Latency IS low) THEN (Network IS high)"
107 R_N3 = "IF (Output IS very_low) AND (Latency IS normal) THEN (Network IS
     high)"
108 R_N4 = "IF (Output IS very_low) AND (Latency IS high) THEN (Network IS
     very_high)"
109 R_N5 = "IF (Output IS very_low) AND (Latency IS very_high) THEN (Network IS
     very_high)"
110 R_N6 = "IF (Output IS low) AND (Latency IS very_low) THEN (Network IS low"
111 R_N7 = "IF (Output IS low) AND (Latency IS low) THEN (Network IS low)"
112 R_N8 = "IF (Output IS low) AND (Latency IS normal) THEN (Network IS high)"
R_N9 = "IF (Output IS low) AND (Latency IS high) THEN (Network IS high)"
114 R_N10 = "IF (Output IS low) AND (Latency IS very_high) THEN (Network IS
     very_high)"
115 R_N11 = "IF (Output IS normal) AND (Latency IS very_low) THEN (Network IS
     low)"
116 R_N12 = "IF (Output IS normal) AND (Latency IS low) THEN (Network IS low)"
_{117} R_N13 = "IF (Output IS normal) AND (Latency IS normal) THEN (Network IS
     normal)"
118 R_N14 = "IF (Output IS normal) AND (Latency IS high) THEN (Network IS high)"
119 R_N15 = "IF (Output IS normal) AND (Latency IS very_high) THEN (Network IS
     high)"
120 R_N16 = "IF (Output IS high) AND (Latency IS low) THEN (Network IS low)"
121 R_N17 = "IF (Output IS high) AND (Latency IS very_low) THEN (Network IS
     very_low)"
122 R_N18 = "IF (Output IS high) AND (Latency IS normal) THEN (Network IS low)"
123 R_N19 = "IF (Output IS high) AND (Latency IS high) THEN (Network IS low)"
R_N20 = "IF (Output IS high) AND (Latency IS very_high) THEN (Network IS
     high)"
125 R_N21 = "IF (Output IS very_high) AND (Latency IS low) THEN (Network IS
    very_low)"
```

```
126 R_N22 = "IF (Output IS very_high) AND (Latency IS very_low) THEN (Network IS
      very_low)"
127 R_N23 = "IF (Output IS very_high) AND (Latency IS normal) THEN (Network IS
     low)"
  R_N24 = "IF (Output IS very_high) AND (Latency IS high) THEN (Network IS low
     ) "
129 R_N25 = "IF (Output IS very_high) AND (Latency IS very_high) THEN (Network
     IS low)"
130 FS_N.add_rules([ R_N1, R_N2, R_N3, R_N4, R_N5, R_N6, R_N7, R_N8, R_N9, R_N10
      , R_N11, R_N12, R_N13, R_N14, R_N15, R_N16, R_N17, R_N18, R_N19, R_N20,
     R_N21, R_N22, R_N23, R_N24, R_N25])
  133 #Rede Final
134 FS_F = FuzzySystem()
136 G_1 = FuzzySet(function=Triangular_MF(a=0, b=0, c=0.25), term="very_low")
137 G_2 = FuzzySet(function=Triangular_MF(a=0.0, b=0.25, c=0.5), term="low")
138 G_3 = FuzzySet(function=Triangular_MF(a=0.25, b=0.5, c=0.75), term="normal")
139 G_4 = FuzzySet(function=Triangular_MF(a=0.5, b=0.75, c=1), term="high")
140 G_5 = FuzzySet(function=Triangular_MF(a=0.75, b=1, c=1), term="very_high")
141 G_LV = LinguisticVariable([G_1, G_2, G_3,G_4,G_5], concept="Network",
     universe_of_discourse=[0, 1])
142 FS_F.add_linguistic_variable("Network", G_LV)
143
144 H_1 = FuzzySet(function=Triangular_MF(a=0, b=0, c=0.25), term="very_low")
145 H_2 = FuzzySet(function=Triangular_MF(a=0.0, b=0.25, c=0.5), term="low")
146 H_3 = FuzzySet(function=Triangular_MF(a=0.25, b=0.5, c=0.75), term="normal")
147 H_4 = FuzzySet(function=Triangular_MF(a=0.5, b=0.75, c=1), term="high")
148 H_5 = FuzzySet(function=Triangular_MF(a=0.75, b=1, c=1), term="very_high")
149 H_LV = LinguisticVariable([H_1, H_2, H_3, H_4,H_5], concept="Load",
     universe_of_discourse=[0, 1])
FS_F.add_linguistic_variable("Load", H_LV)
151
# Define output fuzzy sets and linguistic variable
J<sub>153</sub> J<sub>1</sub> = FuzzySet(function=Triangular_MF(a=-1, b=-1, c=-0.5), term="very_low")
J<sub>2</sub> = FuzzySet(function=Triangular_MF(a=-1, b=-0.5, c=0.0), term="low")
J_3 = FuzzySet(function=Triangular_MF(a=-0.5, b=0.0, c=0.5), term="normal")
156 J_4 = FuzzySet(function=Triangular_MF(a=0.0, b=0.5, c=1), term="high")
J<sub>5</sub> J<sub>5</sub> = FuzzySet(function=Triangular_MF(a=0.5, b=1, c=1), term="very_high")
J_LV = LinguisticVariable([J_1, J_2, J_3,J_4,J_5], universe_of_discourse
     =[-1, 1])
159 FS_F.add_linguistic_variable("Result", J_LV)
#FS_L.produce_figure()
  # Define fuzzy rules
161
162
163 R_F1 = "IF (Network IS very_low) AND (Load IS very_low) THEN (Result IS low)
164 R_F2 = "IF (Network IS very_low) AND (Load IS low) THEN (Result IS low)"
165 R_F3 = "IF (Network IS very_low) AND (Load IS normal) THEN (Result IS
     very_low)"
166 R_F4 = "IF (Network IS very_low) AND (Load IS high) THEN (Result IS very_low
     ) "
167 R_F5 = "IF (Network IS very_low) AND (Load IS very_high) THEN (Result IS
```

```
168 R_F6 = "IF (Network IS low) AND (Load IS very_low) THEN (Result IS high"
169 R_F7 = "IF (Network IS low) AND (Load IS low) THEN (Result IS low)"
170 R_F8 = "IF (Network IS low) AND (Load IS normal) THEN (Result IS low)"
171 R_F9 = "IF (Network IS low) AND (Load IS high) THEN (Result IS low)"
172 R_F10 = "IF (Network IS low) AND (Load IS very_high) THEN (Result IS
      very_low)"
_{173} R_F11 = "IF (Network IS normal) AND (Load IS very_low) THEN (Result IS
      very_high)"
174 R_F12 = "IF (Network IS normal) AND (Load IS low) THEN (Result IS high)"
175 R_F13 = "IF (Network IS normal) AND (Load IS normal) THEN (Result IS normal)
176 R_F14 = "IF (Network IS normal) AND (Load IS high) THEN (Result IS low)"
R_F15 = "IF (Network IS normal) AND (Load IS very_high) THEN (Result IS
     very_low)"
178 R_F16 = "IF (Network IS high) AND (Load IS low) THEN (Result IS very_high)"
179 R_F17 = "IF (Network IS high) AND (Load IS very_low) THEN (Result IS
      very_high)"
180 R_F18 = "IF (Network IS high) AND (Load IS normal) THEN (Result IS high)"
181 R_F19 = "IF (Network IS high) AND (Load IS high) THEN (Result IS low)"
182 R_F20 = "IF (Network IS high) AND (Load IS very_high) THEN (Result IS
      very_low)"
R_F21 = "IF (Network IS very_high) AND (Load IS low) THEN (Result IS
      very_high)"
184 R_F22 = "IF (Network IS very_high) AND (Load IS very_low) THEN (Result IS
      very_high)"
185 R_F23 = "IF (Network IS very_high) AND (Load IS normal) THEN (Result IS
     very_high)"
R_F24 = "IF (Network IS very_high) AND (Load IS high) THEN (Result IS high)"
R_F25 = "IF (Network IS very_high) AND (Load IS very_high) THEN (Result IS
     very_low)"
188 FS_F.add_rules([ R_F1, R_F2, R_F3, R_F4, R_F5, R_F6, R_F7, R_F8, R_F9, R_F10
      , R_F11, R_F12, R_F13, R_F14, R_F15, R_F16, R_F17, R_F18, R_F19, R_F20,
      R_F21, R_F22, R_F23, R_F24, R_F25])
  # Set antecedents values
190
  for n_teste in range(10):
      FS_N.set_variable("Latency", df['Latency'][n_teste])
192
      FS_N.set_variable("Output", df['OutBandwidth'][n_teste])
193
      FS_L.set_variable("Memory", df['ProcessorLoad'][n_teste])
195
      FS_L.set_variable("Processor", df['MemoryUsage'][n_teste])
196
197
      FS_F.set_variable("Load", float(FS_L.Mamdani_inference(["Load"])["Load"
     ]))
      FS_F.set_variable("Network", float(FS_N.Mamdani_inference(["Network"])["
199
      Network"]))
      print("Teste n: ", n_teste )
201
      print(FS_F.Mamdani_inference(["Result"]))
202
      print("Resultado esperado: ", df['CLPVariation'][n_teste])
203
205 from mpl_toolkits.mplot3d import Axes3D
      # Plotting surface
207 \text{ xs} = []
208 \text{ ys} = []
```

```
209 zs = []
210 DIVs = 20
  for x in np.linspace(0,1,DIVs):
       for y in np.linspace(0,1,DIVs):
212
           FS_N.set_variable("Latency", x)
           FS_N.set_variable("Output", y)
214
           tip = FS_N.Mamdani_inference()["Network"]
215
           xs.append(x)
216
           ys.append(y)
           zs.append(tip)
218
219 xs = np.array(xs)
220 ys = np.array(ys)
  zs = np.array(zs)
222
223 fig = plt.figure()
224 ax = fig.add_subplot(111, projection='3d')
226 xx, yy = plt.meshgrid(xs,ys)
227
228 ax.plot_trisurf(xs,ys,zs, vmin=0, vmax=1, cmap='gnuplot2')
ax.set_xlabel("Latency")
230 ax.set_ylabel("Output")
231 ax.set_zlabel("Result")
232 ax.set_title("Fuzzy Network", pad=20)
233 ax.set_zlim(0, 1)
234 plt.tight_layout()
235
      # Plotting surface
237 \text{ xs} = []
  vs = []
238
239 zs = []
for x in np.linspace(0,1,DIVs):
       for y in np.linspace(0,1,DIVs):
           FS_L.set_variable("Memory", x)
242
           FS_L.set_variable("Processor", y)
243
           tip = FS_L.Mamdani_inference()["Load"]
           xs.append(x)
245
           ys.append(y)
246
247
           zs.append(tip)
248 xs = np.array(xs)
ys = np.array(ys)
250 zs = np.array(zs)
252 fig = plt.figure()
  ax = fig.add_subplot(111, projection='3d')
253
254
255 xx, yy = plt.meshgrid(xs,ys)
ax.plot_trisurf(xs,ys,zs, vmin=0, vmax=1, cmap='gnuplot2')
ax.set_xlabel("Memory")
259 ax.set_ylabel("Processor")
260 ax.set_zlabel("Result")
ax.set_title("Fuzzy Load", pad=20)
262 ax.set_zlim(0, 1)
263 plt.tight_layout()
```

```
# Plotting surface
266 \text{ xs} = []
267 \text{ vs} = []
  zs = []
  for x in np.linspace(0,1,DIVs):
       for y in np.linspace(0,1,DIVs):
           FS_F.set_variable("Load", x)
271
           FS_F.set_variable("Network", y)
           tip = FS_F.Mamdani_inference()["Result"]
           xs.append(x)
274
           ys.append(y)
           zs.append(tip)
277 xs = np.array(xs)
278 ys = np.array(ys)
z_{79} z_{5} = np.array(z_{5})
281 from mpl_toolkits.mplot3d import Axes3D
282 fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
xx, yy = plt.meshgrid(xs,ys)
286
287 ax.plot_trisurf(xs,ys,zs, vmin=-1, vmax=1, cmap='gnuplot2')
288 ax.set_xlabel("Load")
289 ax.set_ylabel("Network")
290 ax.set_zlabel("Result")
291 ax.set_title("Fuzzy final", pad=20)
292 ax.set_zlim(-1, 1)
  plt.tight_layout()
293
294
296 plt.show()
```

B Código TestMe

```
import pandas as pd
from simpful import *
from sklearn.neural_network import MLPRegressor

d

from sklearn.neural_net
```

```
18 S_LV = LinguisticVariable([S_1, S_2, S_3,S_4,S_5], concept="Memory usage",
     universe_of_discourse=[0,1])
19 FS_L.add_linguistic_variable("Memory",S_LV)
21 F_1 = FuzzySet(function=Triangular_MF(a=0, b=0, c=0.25), term="very_low")
22 F_2 = FuzzySet(function=Triangular_MF(a=0.0, b=0.25, c=0.5), term="low")
F_3 = FuzzySet(function=Triangular_MF(a=0.25, b=0.5, c=0.75), term="normal")
24 F_4 = FuzzySet(function=Triangular_MF(a=0.5, b=0.75, c=1), term="high")
25 F_5 = FuzzySet(function=Triangular_MF(a=0.75, b=1, c=1), term="very_high")
26 F_LV = LinguisticVariable([F_1, F_2,F_3,F_4,F_5], concept="Processor load",
     universe_of_discourse=[0,1])
27 FS_L.add_linguistic_variable("Processor", F_LV)
29 # Define output fuzzy sets and linguistic variable
30 T_1 = FuzzySet(function=Triangular_MF(a=0, b=0, c=0.25), term="very_low")
31 T_2 = FuzzySet(function=Triangular_MF(a=0.0, b=0.25, c=0.5), term="low")
32 T_3 = FuzzySet(function=Triangular_MF(a=0.25, b=0.5, c=0.75), term="normal")
33 T_4 = FuzzySet(function=Triangular_MF(a=0.5, b=0.75, c=1), term="high")
34 T_5 = FuzzySet(function=Triangular_MF(a=0.75, b=1, c=1), term="very_high")
T_LV = LinguisticVariable([T_1, T_2, T_3, T_4, T_5], universe_of_discourse
     =[0,1])
36 FS_L.add_linguistic_variable("Load", T_LV)
37
38 # Define fuzzy rules
39 R_L1 = "IF (Memory IS very_low) AND (Processor IS very_low) THEN (Load IS
     very_high)"
40 R_L2 = "IF (Memory IS very_low) AND (Processor IS low) THEN (Load IS
     very_high)"
 R_L3 = "IF (Memory IS very_low) AND (Processor IS normal) THEN (Load IS high
42 R_L4 = "IF (Memory IS very_low) AND (Processor IS high) THEN (Load IS high)"
43 R_L5 = "IF (Memory IS very_low) AND (Processor IS very_high) THEN (Load IS
44 R_L6 = "IF (Memory IS low) AND (Processor IS very_low) THEN (Load IS
     very_high)"
45 R_L7 = "IF (Memory IS low) AND (Processor IS low) THEN (Load IS very_high)"
_{46} R_L8 = "IF (Memory IS low) AND (Processor IS normal) THEN (Load IS high)"
47 R_L9 = "IF (Memory IS low) AND (Processor IS high) THEN (Load IS high)"
48 R_L10 = "IF (Memory IS low) AND (Processor IS very_high) THEN (Load IS low)"
_{49} R_L11 = "IF (Memory IS normal) AND (Processor IS very_low) THEN (Load IS
     high)"
_{50} R_L12 = "IF (Memory IS normal) AND (Processor IS low) THEN (Load IS high)"
51 R_L13 = "IF (Memory IS normal) AND (Processor IS normal) THEN (Load IS
     normal)"
52 R_L14 = "IF (Memory IS normal) AND (Processor IS high) THEN (Load IS low)"
53 R_L15 = "IF (Memory IS normal) AND (Processor IS very_high) THEN (Load IS
     low)"
54 R_L16 = "IF (Memory IS high) AND (Processor IS very_low) THEN (Load IS high)
55 R_L17 = "IF (Memory IS high) AND (Processor IS low) THEN (Load IS high)"
56 R_L18 = "IF (Memory IS high) AND (Processor IS normal) THEN (Load IS low)"
87 R_L19 = "IF (Memory IS high) AND (Processor IS high) THEN (Load IS low)"
58 R_L20 = "IF (Memory IS high) AND (Processor IS very_high) THEN (Load IS
     very_low)"
59 R_L21 = "IF (Memory IS very_high) AND (Processor IS very_low) THEN (Load IS
```

```
high)"
60 R_L22 = "IF (Memory IS very_high) AND (Processor IS low) THEN (Load IS low)"
61 R_L23 = "IF (Memory IS very_high) AND (Processor IS normal) THEN (Load IS
62 R_L24 = "IF (Memory IS very_high) AND (Processor IS high) THEN (Load IS
     very_low)"
63 R_L25 = "IF (Memory IS very_high) AND (Processor IS very_high) THEN (Load IS
      very_low)"
64 FS_L.add_rules([ R_L1, R_L2, R_L3, R_L4, R_L5, R_L6, R_L7, R_L8, R_L9, R_L10
      , R_L11, R_L12, R_L13, R_L14, R_L15, R_L16, R_L17, R_L18, R_L19, R_L20,
     R_L21, R_L22, R_L23, R_L24, R_L25 ])
  ########### FIM Da REDE FS_L
                                  68
69 FS_N = FuzzySystem()
70 ## REDE NETWORK
71 L_1 = FuzzySet(function=Triangular_MF(a=0, b=0, c=0.25), term="very_low")
72 L_2 = FuzzySet(function=Triangular_MF(a=0.0, b=0.25, c=0.5), term="low")
73 L_3 = FuzzySet(function=Triangular_MF(a=0.25, b=0.5, c=0.75), term="normal")
74 L_4 = FuzzySet(function=Triangular_MF(a=0.5, b=0.75, c=1), term="high")
75 L_5 = FuzzySet(function=Triangular_MF(a=0.75, b=1, c=1), term="very_high")
76 L_LV = LinguisticVariable([L_1, L_2, L_3,L_4,L_5], concept="Latency",
     universe_of_discourse=[0, 1])
77 FS_N.add_linguistic_variable("Latency", L_LV)
79 O_1 = FuzzySet(function=Triangular_MF(a=0, b=0, c=0.25), term="very_low")
80 O_2 = FuzzySet(function=Triangular_MF(a=0.0, b=0.25, c=0.5), term="low")
0_3 = FuzzySet(function=Triangular_MF(a=0.25, b=0.5, c=0.75), term="normal")
0_4 = FuzzySet(function=Triangular_MF(a=0.5, b=0.75, c=1), term="high")
83 O_5 = FuzzySet(function=Triangular_MF(a=0.75, b=1, c=1), term="very_high")
84 O_LV = LinguisticVariable([0_1, 0_2, 0_3,0_4,0_5], concept="Output Network
     Throughput", universe_of_discourse=[0, 1])
85 FS_N.add_linguistic_variable("Output", O_LV)
87 # Define output fuzzy sets and linguistic variable
88 N_1 = FuzzySet(function=Triangular_MF(a=0, b=0, c=0.25), term="very_low")
89 N_2 = FuzzySet(function=Triangular_MF(a=0.0, b=0.25, c=0.5), term="low")
90 N_3 = FuzzySet(function=Triangular_MF(a=0.25, b=0.5, c=0.75), term="normal")
91 N_4 = FuzzySet(function=Triangular_MF(a=0.5, b=0.75, c=1), term="high")
92 N_5 = FuzzySet(function=Triangular_MF(a=0.75, b=1, c=1), term="very_high")
93 N_LV = LinguisticVariable([N_1, N_2, N_3, N_4, N_5], universe_of_discourse
     =[0, 1])
94 FS_N.add_linguistic_variable("Network", N_LV)
  # Define fuzzy rules
95
97 R_N1 = "IF (Output IS very_low) AND (Latency IS very_low) THEN (Network IS
98 R_N2 = "IF (Output IS very_low) AND (Latency IS low) THEN (Network IS high)"
99 R_N3 = "IF (Output IS very_low) AND (Latency IS normal) THEN (Network IS
     high)"
100 R_N4 = "IF (Output IS very_low) AND (Latency IS high) THEN (Network IS
     very_high)"
101 R_N5 = "IF (Output IS very_low) AND (Latency IS very_high) THEN (Network IS
  very_high)"
```

```
102 R_N6 = "IF (Output IS low) AND (Latency IS very_low) THEN (Network IS low"
103 R_N7 = "IF (Output IS low) AND (Latency IS low) THEN (Network IS low)"
104 R_N8 = "IF (Output IS low) AND (Latency IS normal) THEN (Network IS high)"
105 R_N9 = "IF (Output IS low) AND (Latency IS high) THEN (Network IS high)"
106 R_N10 = "IF (Output IS low) AND (Latency IS very_high) THEN (Network IS
     very_high)"
_{107} R_N11 = "IF (Output IS normal) AND (Latency IS very_low) THEN (Network IS
     low)"
108 R_N12 = "IF (Output IS normal) AND (Latency IS low) THEN (Network IS low)"
109 R_N13 = "IF (Output IS normal) AND (Latency IS normal) THEN (Network IS
     normal)"
110 R_N14 = "IF (Output IS normal) AND (Latency IS high) THEN (Network IS high)"
R_N15 = "IF (Output IS normal) AND (Latency IS very_high) THEN (Network IS
     high)"
112 R_N16 = "IF (Output IS high) AND (Latency IS low) THEN (Network IS low)"
113 R_N17 = "IF (Output IS high) AND (Latency IS very_low) THEN (Network IS
     very_low)"
114 R_N18 = "IF (Output IS high) AND (Latency IS normal) THEN (Network IS low)"
115 R_N19 = "IF (Output IS high) AND (Latency IS high) THEN (Network IS low)"
116 R_N20 = "IF (Output IS high) AND (Latency IS very_high) THEN (Network IS
     high)"
117 R_N21 = "IF (Output IS very_high) AND (Latency IS low) THEN (Network IS
     very_low)"
118 R_N22 = "IF (Output IS very_high) AND (Latency IS very_low) THEN (Network IS
      very_low)"
119 R_N23 = "IF (Output IS very_high) AND (Latency IS normal) THEN (Network IS
     low)"
  R_N24 = "IF (Output IS very_high) AND (Latency IS high) THEN (Network IS low
     ) "
121 R_N25 = "IF (Output IS very_high) AND (Latency IS very_high) THEN (Network
     IS low)"
122 FS_N.add_rules([ R_N1, R_N2, R_N3, R_N4, R_N5, R_N6, R_N7, R_N8, R_N9, R_N10
      , R_N11, R_N12, R_N13, R_N14, R_N15, R_N16, R_N17, R_N18, R_N19, R_N20,
     R_N21, R_N22, R_N23, R_N24, R_N25])
125 #Rede Final
126 FS_F = FuzzySystem()
127
128 G_1 = FuzzySet(function=Triangular_MF(a=0, b=0, c=0.25), term="very_low")
129 G_2 = FuzzySet(function=Triangular_MF(a=0.0, b=0.25, c=0.5), term="low")
130 G_3 = FuzzySet(function=Triangular_MF(a=0.25, b=0.5, c=0.75), term="normal")
131 G_4 = FuzzySet(function=Triangular_MF(a=0.5, b=0.75, c=1), term="high")
132 G_5 = FuzzySet(function=Triangular_MF(a=0.75, b=1, c=1), term="very_high")
133 G_LV = LinguisticVariable([G_1, G_2, G_3,G_4,G_5], concept="Network",
     universe_of_discourse=[0, 1])
134 FS_F.add_linguistic_variable("Network", G_LV)
136 H_1 = FuzzySet(function=Triangular_MF(a=0, b=0, c=0.25), term="very_low")
137 H_2 = FuzzySet(function=Triangular_MF(a=0.0, b=0.25, c=0.5), term="low")
H_3 = FuzzySet(function=Triangular_MF(a=0.25, b=0.5, c=0.75), term="normal")
H_4 = FuzzySet(function=Triangular_MF(a=0.5, b=0.75, c=1), term="high")
140 H_5 = FuzzySet(function=Triangular_MF(a=0.75, b=1, c=1), term="very_high")
141 H_LV = LinguisticVariable([H_1, H_2, H_3, H_4,H_5], concept="Load",
     universe_of_discourse=[0, 1])
```

```
142 FS_F.add_linguistic_variable("Load", H_LV)
# Define output fuzzy sets and linguistic variable
145 J_1 = FuzzySet(function=Triangular_MF(a=-1, b=-1, c=-0.5), term="very_low")
J_2 = FuzzySet(function=Triangular_MF(a=-1, b=-0.5, c=0.0), term="low")
J_3 = FuzzySet(function=Triangular_MF(a=-0.5, b=0.0, c=0.5), term="normal")
J_4 = FuzzySet(function=Triangular_MF(a=0.0, b=0.5, c=1), term="high")
_{149} J_5 = FuzzySet(function=Triangular_MF(a=0.5, b=1, c=1), term="very_high")
J_LV = LinguisticVariable([J_1, J_2, J_3,J_4,J_5], universe_of_discourse
     =[-1, 1]
FS_F.add_linguistic_variable("Result", J_LV)
152 # Define fuzzy rules
154 R_F1 = "IF (Network IS very_low) AND (Load IS very_low) THEN (Result IS low)
155 R_F2 = "IF (Network IS very_low) AND (Load IS low) THEN (Result IS low)"
156 R_F3 = "IF (Network IS very_low) AND (Load IS normal) THEN (Result IS
     very_low)"
157 R_F4 = "IF (Network IS very_low) AND (Load IS high) THEN (Result IS very_low
     ) "
158 R_F5 = "IF (Network IS very_low) AND (Load IS very_high) THEN (Result IS
     very_low)"
159 R_F6 = "IF (Network IS low) AND (Load IS very_low) THEN (Result IS high"
_{160} R_F7 = "IF (Network IS low) AND (Load IS low) THEN (Result IS low)"
161 R_F8 = "IF (Network IS low) AND (Load IS normal) THEN (Result IS low)"
162 R_F9 = "IF (Network IS low) AND (Load IS high) THEN (Result IS low)"
163 R_F10 = "IF (Network IS low) AND (Load IS very_high) THEN (Result IS
     very_low)"
_{164} R_F11 = "IF (Network IS normal) AND (Load IS very_low) THEN (Result IS
     very_high)"
165 R_F12 = "IF (Network IS normal) AND (Load IS low) THEN (Result IS high)"
166 R_F13 = "IF (Network IS normal) AND (Load IS normal) THEN (Result IS normal)
R_F14 = "IF (Network IS normal) AND (Load IS high) THEN (Result IS low)"
168 R_F15 = "IF (Network IS normal) AND (Load IS very_high) THEN (Result IS
     very_low)"
169 R_F16 = "IF (Network IS high) AND (Load IS low) THEN (Result IS very_high)"
170 R_F17 = "IF (Network IS high) AND (Load IS very_low) THEN (Result IS
     very_high)"
171 R_F18 = "IF (Network IS high) AND (Load IS normal) THEN (Result IS high)"
172 R_F19 = "IF (Network IS high) AND (Load IS high) THEN (Result IS low)"
R_F20 = "IF (Network IS high) AND (Load IS very_high) THEN (Result IS
     very_low)"
_{174} R_F21 = "IF (Network IS very_high) AND (Load IS low) THEN (Result IS
     very_high)"
175 R_F22 = "IF (Network IS very_high) AND (Load IS very_low) THEN (Result IS
     very_high)"
176 R_F23 = "IF (Network IS very_high) AND (Load IS normal) THEN (Result IS
     very_high)"
177 R_F24 = "IF (Network IS very_high) AND (Load IS high) THEN (Result IS high)"
178 R_F25 = "IF (Network IS very_high) AND (Load IS very_high) THEN (Result IS
     very_low)"
179 FS_F.add_rules([ R_F1, R_F2, R_F3, R_F4, R_F5, R_F6, R_F7, R_F8, R_F9, R_F10
     , R_F11, R_F12, R_F13, R_F14, R_F15, R_F16, R_F17, R_F18, R_F19, R_F20,
    R_F21, R_F22, R_F23, R_F24, R_F25])
```

```
180 # Set antecedents values
181
  for n_teste in range(len(df)):
182
      FS_N.set_variable("Latency", df['Latency'][n_teste])
183
      FS_N.set_variable("Output", df['OutNetThroughput'][n_teste])
185
      FS_L.set_variable("Memory", df['ProcessorLoad'][n_teste])
186
      FS_L.set_variable("Processor", df['MemoryUsage'][n_teste])
187
188
      FS_F.set_variable("Load", float(FS_L.Mamdani_inference(["Load"])["Load"
189
      FS_F.set_variable("Network", float(FS_N.Mamdani_inference(["Network"])["
190
      Network"]))
      df["CLPVariation"][n_teste] = FS_F.Mamdani_inference(["Result"])["
191
      Result"1
      df.to_csv("Projeto_2/Lab10-Proj2_TestS.csv")
192
```