## Atividade Prática II - Parte 2

- Alunos: Giovanni Martins de Sá Júnior(2017001850) e João Vitor Gato de Araújo (2017089090)
- Disciplina: Sistemas Nebulosos
- Professor: Cristiano Leite de Castro
- 1. Ler o Capítulo 4 do livro texto: Jyh-Shing Roger Jang and Chuen-Tsai Sun. 1996. Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence.
- ▼ 2. Seja a função y = cosseno(x), para x definido no intervalo de  $[-\pi/2, 3\pi/2]$ , conforme ilustra a Figura 2. Pede-se:

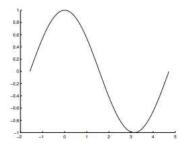


Figure 1.1: Função y = cosseno(x) no intervalo de  $[-\pi/2, 3\pi/2]$ .

a. Empregue o mecanismo o mecanismo de inferência de Sugeno com consequentes de ordem 1 (linear) e obtenha uma expressão analítica para aproximar esta função.

1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -		1 1
Attridade Pratica	T- Parte 2	The state of the s
- Committee of the comm	1000	
Alunos: Cievanni	Moderno de Do Turior (20	17001850)
João lit	Martino di Do Junion (20 on Centre de Araijo (20170)	89095)
	0	
	V A F TI 3-17	· · · · · · · · · · · · · · · · · · · ·
		o macanismo de ingléricia de Dugno
-> Diginição dos	gunção de pertinência	(dita apussão araltica)
pora x mo un	terral de [-1/2, 31/2]	usando gunzas triangulores
· Conjunto 1: X pr	olimo a - "12	
La Função de	Peterincia etiangular:	You the straight was
+	0	week who stands keep
	BLX < QL	Em que:
	-ar, su art x tb1	The thought the
10 (24)	1-01	Q1 = - 1/2 - 1/6 = -491/6
Mr(x)=	1-x , on b1 < x < C1	b1=-1/2
C	7-PT	C1=- T/2+ T/6 = -3/11/6
The transfer of the party of	DI DE X7CL	7 100 1 0 10 10 10 10 10 10 10 10 10 10 1
· (erizunto 2:	x profime at	delta-o A-xaxa
1 51	A Starta	andra descrip
	), se x < Q2	Em gra:
×	-az saaz≼x≤bz	4
	02-02	a2 = T-1/6 = 5T/6
(J. 5()	2-x, sub2 < x < C2	62 = T
	Cz-bz	CZ = T+ T/6 = 7T/6
	), De X7(2)	3 - ot 1 - 18 7 ( + 4 - 1 - ) - 10 - 10 - 10 - 10 - 10 - 10 -
		de date de trata
	Dang!	
		The second second
Landa de la companya del companya de la companya del companya de la companya de l		WE WAT DIE ASTER
The second	A STATE OF THE PERSON NAMED IN	9 TELEPTON = PECHO

	0 - 1/1	3 2	Em	gu		
	0, on x < 0 x- Q3, on Q3 b3- Q3	< x < b 2	- WI	Am		
11301=	h2-93	0 × 6 02	93	= 31/2-11/		
Molal-	03 22 03-Y = h2	< v < c2		z 3T/2		
	C3-X, on b3	C * 1		= 311/2+1/6		
	0,50 ×7/C					
· Apatir	desta lógica é	personal enta	io calcula	n a saida g	1 com Se	or was reading
e-conseque	inter de ordern 1	(Dugme).	Tana you	, considerans	אינט לני ב	MG, LIMBS OIL
odem 1 a	nacidos e codo s	ugra. Assi	um, caleus	lomes es reco	lores de sa	ude plade
regio po	naides e codo r	ide" i " gory	tà.			
					-	
Regna 1:	naquentemente, y	um Ola	á de mó	qualidale,	então i p	eguno
1600	in strengtoniaco	2 Dr. Dunie	co + 91.0	emado + YI	,	1
	10					
B. 010 2:	Di a mondio	' Dom on	tão e a	suite a m	dia	
and and	Leverstrangeron	112 = 02.1	Description +	73		
	C. radionamona !	De be	Total Control	)		
P 3 ·	90 Marela	trologo "	N a com	do idolvier	a entre	a arista a
rugia s	onegentemente,	A WASHING O	11 5 4 0	2.comida + V	2)	OFMINDSELE
0	sustantiment,	73 = 63. DAV	rates 1 d	TOWNS.	)	0
0 /	· · · · · · · · · · · · · · · · · · ·	0 1	-0	1	of ma loss	de water
	do y i stide i	epmontland	as volue	o gur rugu	AN ASSAULT	nes branche
h coda:	V					
	9=0	MT- AT + MJ		3.03		
		(MT+1	WZ T WZ	-47		
-> Essessi	ing sitilars and	ol:				
1	-					
	12 (10) DONATIO	- alicentia	+ Y1) + W2	extenses . Sq).	+ 42) + W3	63. varior + 13 co
(),2	MI. The T. Monday.	1 26 2-1-11				

- 3. Escreva um script em Python para aproximar a função y = seno(x), para x definido no intervalo de  $[0, 2\pi]$ , empregando o mecanismo de inferência de Sugeno com consequentes de ordem 1, yj = pjx + qj, onde j é o índice que representa a regra.
  - Experimento 1: use 3 regras e funções de pertinência do tipo Triangular para "fuzzificação" da variável x.
  - Experimento 2: use 3 regras e funções de pertinência do tipo Gaussiana para "fuzzificação" da variável x.
  - Experimento 3: modifique a quantidade de regras 3, 5, 10 e veja o que acontece com a saída do sistema.

Para todos os experimentos, mostre os gráficos resultados das aproximações e calcule o Erro Quadrático Médio EQM onde yi é a saída real da função e y\_hat é a saída obtida pelo sistema nebuloso.

▼ Experimento 1: use 3 regras e funções de pertinência do tipo Triangular para "fuzzificação da variável x."

```
import numpy as np
import matplotlib.pyplot as plt
# Função de pertinência triangular
def triangular(x, a, b, c):
    return max(0, min((x - a) / (b - a), (c - x) / (c - b)))
# Defina a função de mapeamento
def fuzzy_rule(x, j):
   if j == 0:
       a, b, c = 0, np.pi / 2, np.pi
    elif j == 1:
       a, b, c = np.pi / 2, np.pi, 3 * np.pi / 2
    elif j == 2:
       a, b, c = np.pi, 3 * np.pi / 2, 2 * np.pi
    return triangular(x, a, b, c)
# Função de pertinência do consequente (y = px + q)
def consequent(x, p, q):
   return p * x + q
# Parâmetros das regras (p e q)
p_values = [0.7, 1.0, 0.4]
q_values = [0.0, 0.0, 0.0]
# Intervalo [0, 2*pi]
x = np.linspace(0, 2 * np.pi, 100)
y_real = np.sin(x)
y_approx = np.zeros_like(x)
# Calcula a saída aproximada
for i in range(len(x)):
   outputs = [fuzzy_rule(x[i], j) for j in range(3)]
    y_{approx[i]} = sum([consequent(x[i], p_values[j], q_values[j]) * outputs[j] for j in range(3)]) / sum(outputs)
# Calcula o Erro Quadrático Médio (EQM)
eqm = np.mean((y_real - y_approx) ** 2)
     <ipython-input-4-f3e827b6b5c4>:34: RuntimeWarning: invalid value encountered in double_scalars
       y\_approx[i] = sum([consequent(x[i], p\_values[j], q\_values[j]) * outputs[j] for j in range(3)]) / sum(outputs)
# Plotagem dos gráficos
plt.figure(figsize=(10, 6))
plt.plot(x, y real, label='Função Real (sen(x)', color='blue')
plt.plot(x, y_approx, label='Aproximação Fuzzy', color='red')
plt.xlabel('x')
plt.ylabel('y')
plt.title(f'Aproximação da Função Seno com EQM={eqm:.4f}')
plt.legend()
plt.grid(True)
plt.show()
```

## Aproximação da Função Seno com EQM=nan

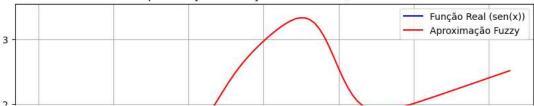


▼ Experimento 2: use 3 regras e funções de pertinência do tipo Gaussiana para "fuzzificação" da variável x.

```
1
import numpy as np
import matplotlib.pyplot as plt
# Função de pertinência Gaussiana
def gaussian(x, mean, sigma):
    return np.exp(-((x - mean) ** 2) / (2 * sigma ** 2))
# Defina a função de mapeamento
def fuzzy_rule(x, j):
    if j == 0:
        mean = np.pi / 2
    elif j == 1:
       mean = np.pi
    elif j == 2:
      mean = 3 * np.pi / 2
    sigma = np.pi / 6 # Ajuste o valor de sigma conforme necessário
   return gaussian(x, mean, sigma)
# Função de pertinência do consequente (y = px + q)
def consequent(x, p, q):
   return p * x + q
# Parâmetros das regras (p e q)
p_values = [0.7, 1.0, 0.4]
q_values = [0.0, 0.0, 0.0]
# Intervalo [0, 2*pi]
x = np.linspace(0, 2 * np.pi, 100)
y_real = np.sin(x)
y_approx = np.zeros_like(x)
# Calcula a saída aproximada
for i in range(len(x)):
   outputs = [fuzzy_rule(x[i], j) for j in range(3)]
    y_approx[i] = sum([consequent(x[i], p_values[j], q_values[j]) * outputs[j] for j in range(3)]) / sum(outputs)
# Calcula o Erro Quadrático Médio (EQM)
eqm = np.mean((y_real - y_approx) ** 2)
# Plotagem dos gráficos
plt.figure(figsize=(10, 6))
plt.plot(x, y_real, label='Função Real (sen(x))', color='blue')
plt.plot(x, y_approx, label='Aproximação Fuzzy', color='red')
plt.xlabel('x')
plt.ylabel('y')
plt.title(f'Aproximação da Função Seno com EQM={eqm:.4f}')
plt.legend()
plt.grid(True)
plt.show()
```

1

## Aproximação da Função Seno com EQM=5.4501



▼ Experimento 3: modifique a quantidade de regras 3, 5, 10 e veja o que acontece com a saída do sistema.

```
import numpy as np
import matplotlib.pyplot as plt
# Função de pertinência Gaussiana
def gaussian(x, mean, sigma):
    return np.exp(-((x - mean) ** 2) / (2 * sigma ** 2))
# Defina a função de mapeamento
def fuzzy_rule(x, j, num_rules):
   mean = (2 * j + 1) * np.pi / (2 * num_rules)
    sigma = np.pi / (2 * num_rules) # Ajuste o valor de sigma conforme necessário
    return gaussian(x, mean, sigma)
# Função de pertinência do consequente (y = px + q)
def consequent(x, p, q):
    return p * x + q
# Parâmetros das regras (p e q)
num_rules = 5  # Altere o número de regras aqui
p_values = np.linspace(0.5, 1.5, num_rules)
q_values = np.zeros(num_rules)
# Intervalo [0, 2*pi]
x = np.linspace(0, 2 * np.pi, 100)
y_real = np.sin(x)
y_{approx} = np.zeros_like(x)
# Calcula a saída aproximada
for i in range(len(x)):
   outputs = [fuzzy_rule(x[i], j, num_rules) for j in range(num_rules)]
   y\_approx[i] = sum([consequent(x[i], p\_values[j], q\_values[j]) * outputs[j] for j in range(num\_rules)]) / sum(outputs)
# Calcula o Erro Ouadrático Médio (EOM)
eqm = np.mean((y_real - y_approx) ** 2)
# Plotagem dos gráficos
plt.figure(figsize=(10, 6))
\verb|plt.plot(x, y_real, label='Função Real (sen(x))', color='blue')|\\
plt.plot(x, y_approx, label='Aproximação Fuzzy', color='red')
plt.xlabel('x')
plt.ylabel('y')
plt.title(f'Aproximação da Função Seno com {num_rules} regras e EQM={eqm:.4f}')
plt.legend()
plt.grid(True)
plt.show()
```

## Aproximação da Função Seno com 5 regras e EQM=32.7189



- **4.** Problema da Gorjeta: Considere as regras de ouro da gorjeta as quais foram construídas segundo a experiência dos clientes ao longo dos anos nos restaurantes americanos:
  - Se o serviço é ruim ou a comida é de má qualidade, então a gorjeta é pequena.
  - Se o serviço é bom, então a gorjeta é média.
  - Se o serviço é excelente ou a comida é deliciosa, então a gorjeta é generosa.

Assuma que uma gorjeta média equivale a 15% do valor da conta, uma gorjeta generosa equivale a 25% e uma gorjeta pequena equivale a 5%. A variável de saída, gorjeta, deve ter a seguinte aparência:

Com base nessas informações, escreva um script em MatLab para projetar um sistema nebuloso que modela o relacionamento entre as variáveis serviço, comida e gorjeta. Use o mecanismo de inferência de Sugeno com consequentes de ordem 1, yj = pjx + qj, onde j é o índice que representa a regra. Mostre o gráfico da aproximação resultante.

```
# Importação de Bibliotecas
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
import matplotlib.pvplot as plt
# Crie as variáveis do problema: serviço, comida e gorjeta
servico = ctrl.Antecedent(np.linspace(0, 10, 101), 'servico')
comida = ctrl.Antecedent(np.linspace(0, 10, 101), 'comida')
gorjeta = ctrl.Consequent(np.linspace(0, 30, 301), 'gorjeta')
# Defina as funções de pertinência para as variáveis de entrada
servico['ruim'] = fuzz.trimf(servico.universe, [0, 0, 5])
servico['bom'] = fuzz.trimf(servico.universe, [0, 5, 10])
servico['excelente'] = fuzz.trimf(servico.universe, [5, 10, 10])
comida['ruim'] = fuzz.trimf(comida.universe, [0, 0, 5])
comida['deliciosa'] = fuzz.trimf(comida.universe, [5, 10, 10])
# Defina as funções de pertinência para a variável de saída (gorjeta)
gorjeta['pequena'] = fuzz.trimf(gorjeta.universe, [0, 0, 15])
gorjeta['media'] = fuzz.trimf(gorjeta.universe, [0, 15, 30])
gorjeta['generosa'] = fuzz.trimf(gorjeta.universe, [15, 30, 30])
# Defina as regras baseadas nas regras de ouro da gorjeta
regra1 = ctrl.Rule(servico['ruim'] | comida['ruim'], gorjeta['pequena'])
regra2 = ctrl.Rule(servico['bom'], gorjeta['media'])
regra3 = ctrl.Rule(servico['excelente'] | comida['deliciosa'], gorjeta['generosa'])
# Crie o sistema de controle
sistema_controle = ctrl.ControlSystem([regra1, regra2, regra3])
# Simule o sistema com entradas específicas
sistema = ctrl.ControlSvstemSimulation(sistema controle)
sistema.input['servico'] = 7.5 # Exemplo: serviço bom
sistema.input['comida'] = 9.0
                              # Exemplo: comida deliciosa
sistema.compute()
# Mostre o gráfico da saída
gorjeta.view(sim=sistema)
plt.show()
\Box
```