Tarea 1.8 Producto Cartesiano Difuso

Abarca Romero José Ángel

Lógica Difusa

2TM9

Conjuntos A y B:

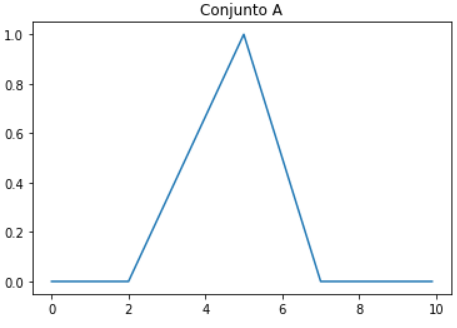


Ilustración Conjunto A definido en el universo X

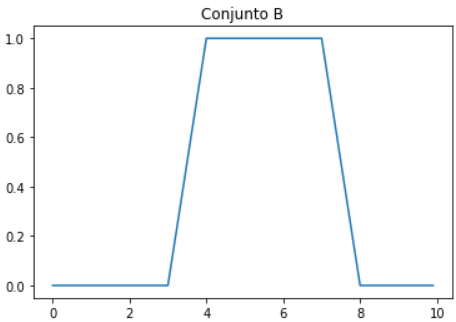


Ilustración Conjunto B definido en el universo Y

Extensión cilíndrica de los conjuntos A y B:

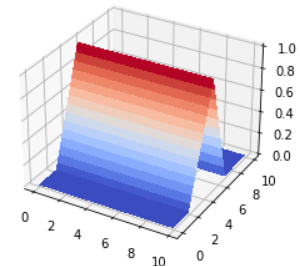


Ilustración Extensión cilíndrica del conjunto A en el espacio XxY

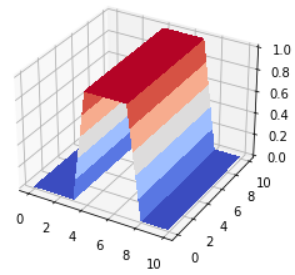


Ilustración Extensión cilíndrica del conjunto B sobre el espacio XxY

Producto cartesiano de las extensiones cilíndricas de los conjuntos A y B:

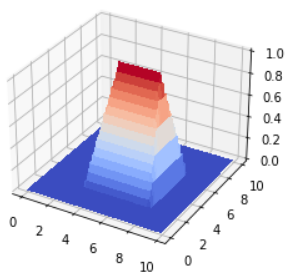


Ilustración Producto cartesiano de C(A) y C(B) en el espacio XxY

Código de Python:

import numpy as np

import matplotlib.pyplot as plt

from matplotlib import cm

x = np.arange(0, 10, 0.1)

y = np.arange(0, 10, 0.1)

X, Y = np.meshgrid(x, y)

A = np.zeros(len(x))

B = np.zeros(len(y))

#Extensiones cilíndricas

CA = np.zeros((len(A),len(B)))

CB = np.zeros((len(A),len(B)))

def funcionTriangular():

  a = 2\*10

  b = 5\*10

  c = 7\*10

  for i in range(len(x)):

    if i <= a:

      A[i] = 0

    elif a < i and i <= b:

      A[i] = (i - a)/(b - a)

    elif b < i and i <= c:

      A[i] = (c - i)/(c - b)

    elif c < i:

      A[i] = 0

def funcionTrapezoidal():

  a = 3\*10

  b = 4\*10

  c = 7\*10

  d = 8\*10

  for i in range(len(x)):

    if i <= a:

      B[i] = 0

    elif a < i and i <= b:

      B[i] = (i - a)/(b - a)

    elif b < i and i <= c:

      B[i] = 1

    elif c < i and i <= d:

      B[i] = (d - i)/(d - c)

    elif d < i:

      B[i] = 0

funcionTriangular()

funcionTrapezoidal()

plt.figure(1)

plt.plot(x,A)

plt.title("Conjunto A")

plt.figure(2)

plt.plot(y,B)

plt.title("Conjunto B")

for i in range(len(A)):

  for j in range(len(B)):

    CA[i,j] = A[i]

    CB[i,j] = B[j]

#Producto cartesiano

CAxCB = np.minimum(CA,CB)

#Extensión cilíndrica de A

fig, ax = plt.subplots(subplot\_kw={"projection": "3d"})

surf = ax.plot\_surface(X, Y, CA, cmap=cm.coolwarm,

                       linewidth=0, antialiased=False)

#Extensión cilíndrica de B

fig2, ax2 = plt.subplots(subplot\_kw={"projection": "3d"})

surf = ax2.plot\_surface(X, Y, CB, cmap=cm.coolwarm,

                       linewidth=0, antialiased=False)

#Producto cartesiano de C(A) y C(B)

fig3, ax3 = plt.subplots(subplot\_kw={"projection": "3d"})

surf = ax3.plot\_surface(X, Y, CAxCB, cmap=cm.coolwarm,

                       linewidth=0, antialiased=False)