

Tarea 1.7 Operadores T-norma y S-norma adaptativos

Abarca Romero José Ángel

Lógica Difusa

2TM9

Operadores T-norma:

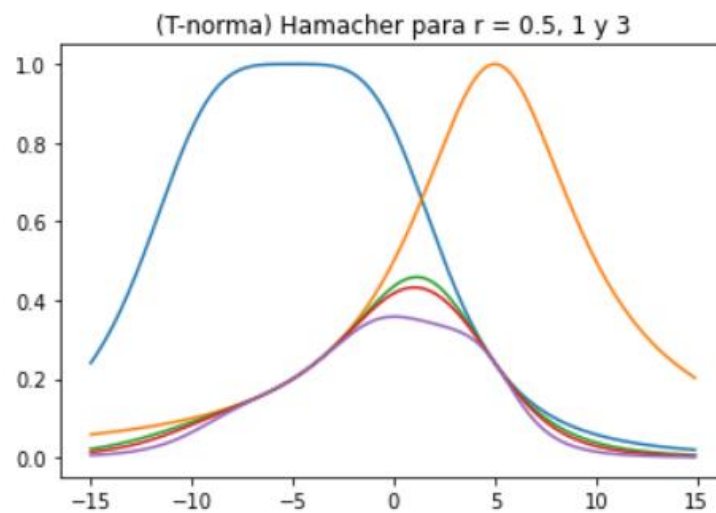


Ilustración 1 Operador de Hamacher con $r > 0$

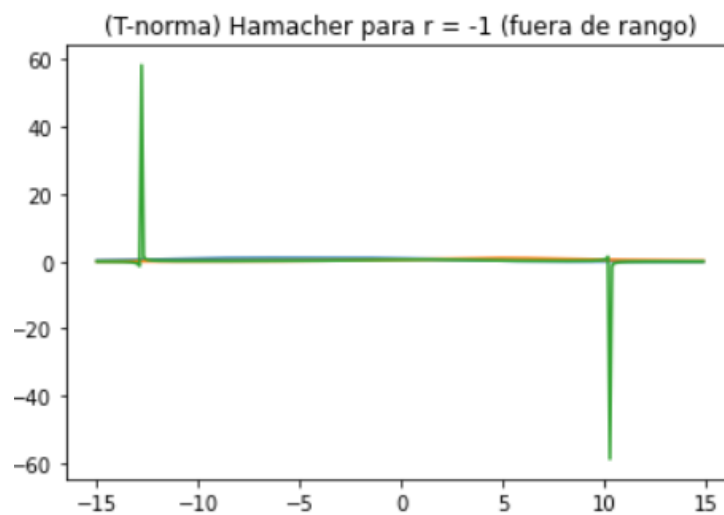


Ilustración 2 Operador de Hamacher para r fuera de rango

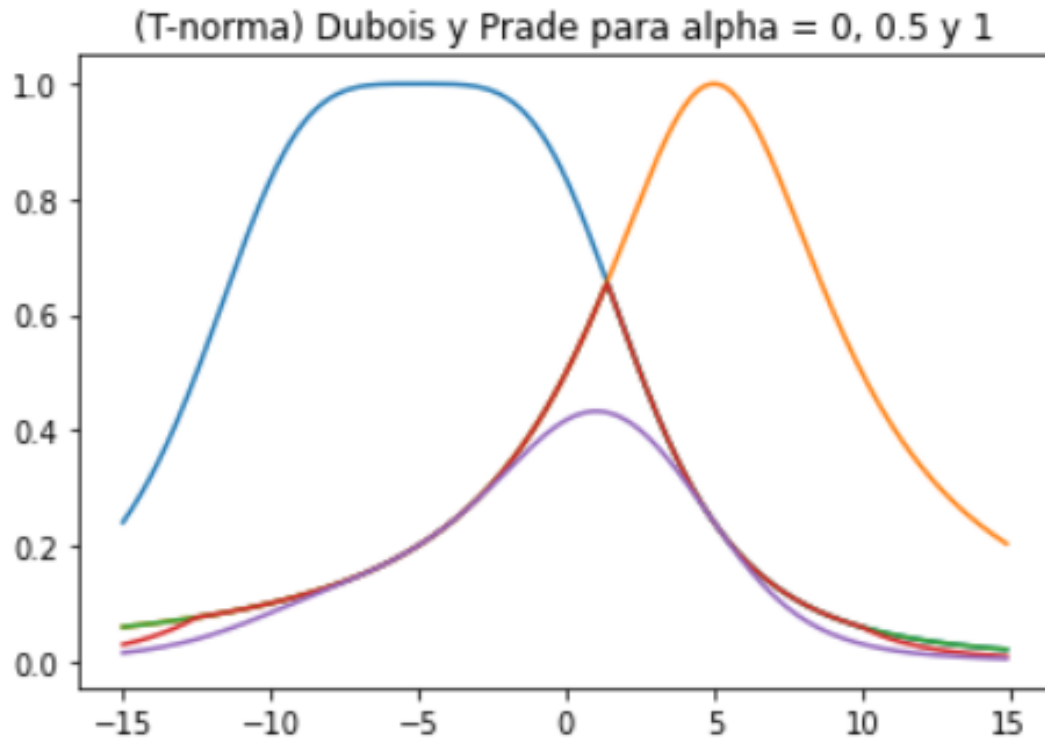


Ilustración 3 Operador de Dubois y Prade con α entre 1 y 0

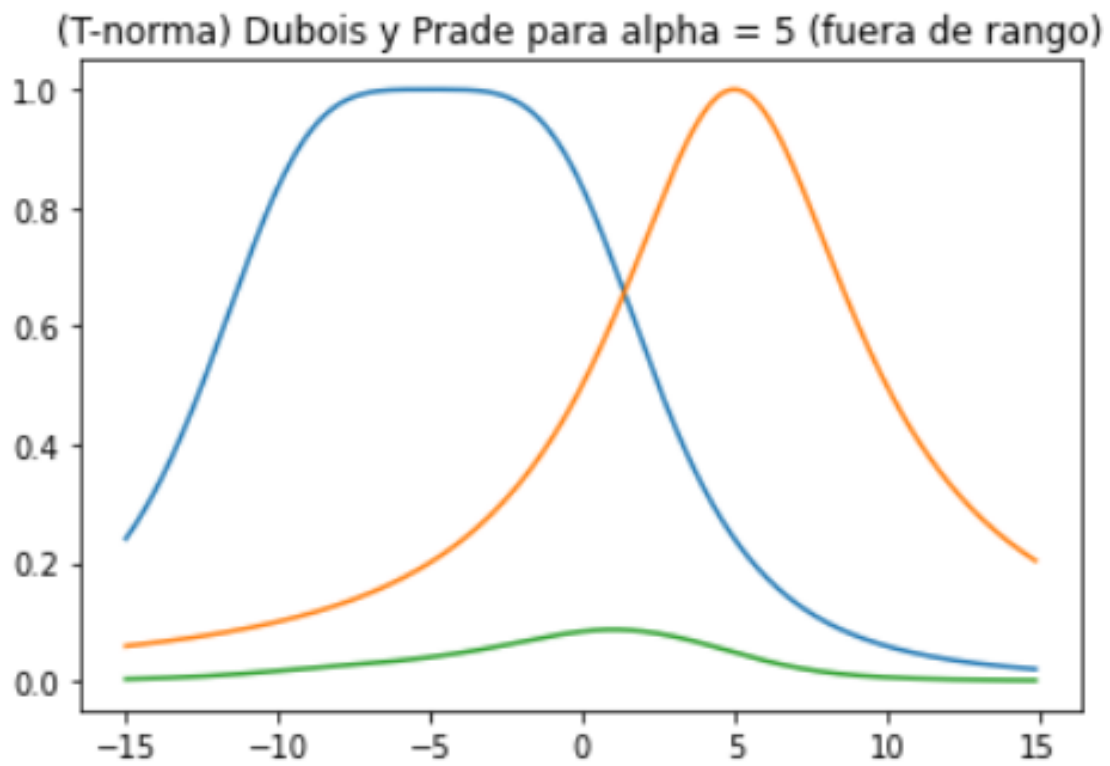


Ilustración 4 Operador de Dubois y Prade con α fuera de rango

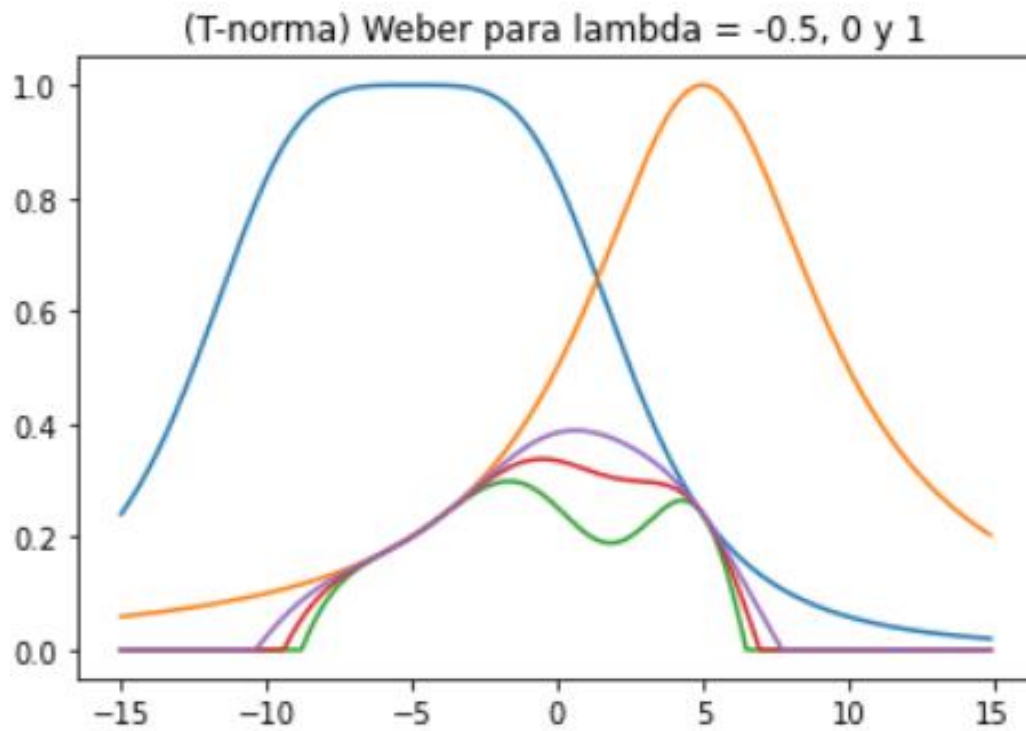


Ilustración 5 Operador de Weber con $\lambda > -1$

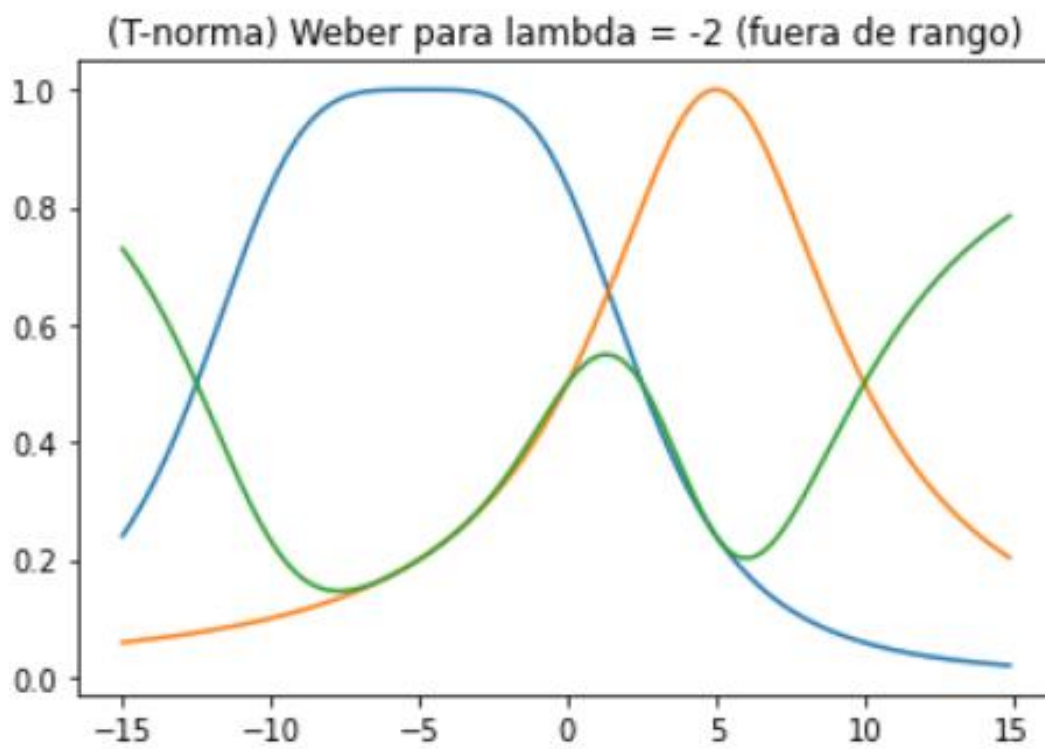


Ilustración 6 Operador de Weber con λ fuera de rango

Código de Python:

```
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(-15, 15, 0.1)
A = np.zeros(len(x))
B = np.zeros(len(x))                                # Rango del universo

for i in range(len(x)):
    iN = i/10-15
    A[i] = 1 / (1 + ((iN+5)/7.5)**4)
    B[i] = 1 / (1 + ((iN-5)/5)**2)

#Conjuntos A y B
plt.figure(1)
plt.title("Conjuntos A y B originales")
plt.plot(x,A,x,B)

#T-norma

#Hamacher
HamT = np.zeros((4,len(x)))
r = [0.5,1,3,-1]
#Dubois y Prade
DPT = np.zeros((4,len(x)))
alfa = [0,0.5,1,5]
#Weber
WebT = np.zeros((4,len(x)))
lamda = [-0.5,0,1.5,-2]

for i in range(len(x)):
    for j in range(4):
        HamT[j][i] = (A[i]*B[i])/(r[j] + (1-r[j])*(A[i]+B[i]-A[i]*B[i]))
        DPT[j][i] = (A[i]*B[i])/(np.maximum(np.maximum(A[i],B[i]),alfa[j]))
        WebT[j][i] =np.maximum(0, (A[i]+B[i]+lamda[j]*A[i]*B[i]-
1)/(1+lamda[j]))

plt.figure(2)
plt.title("(T-norma) Hamacher para r = 0.5, 1 y 3")
plt.plot(x,HamT[0],x,HamT[1],x,HamT[2])
plt.figure(3)
plt.title("(T-norma) Hamacher para r = -1 (fuera de rango)")
plt.plot(x,HamT[3])
```

```

plt.figure(4)
plt.title("(T-norma) Dubois y Prade para alpha = 0, 0.5 y 1")
plt.plot(x,DPT[0],x,DPT[1],x,DPT[2])
plt.figure(5)
plt.title("(T-norma) Dubois y Prade para alpha = 5 (fuera de rango)")
plt.plot(x,DPT[3])

plt.figure(6)
plt.title("(T-norma) Weber para lambda = -0.5, 0 y 1")
plt.plot(x,WebT[0],x,WebT[1],x,WebT[2])
plt.figure(7)
plt.title("(T-norma) Weber para lambda = -2 (fuera de rango)")
plt.plot(x,WebT[3])

```

Operadores S-norma:

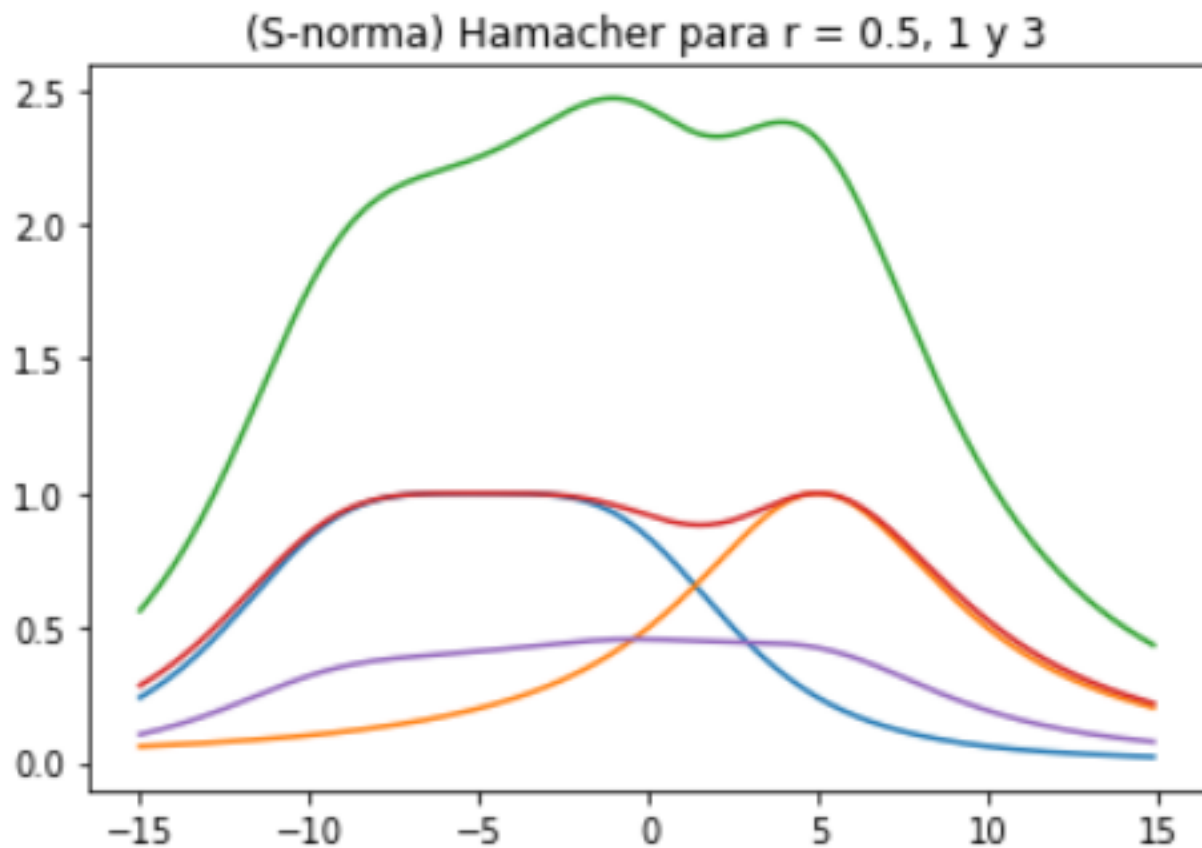


Ilustración 7 Operador de Hamacher con $r > 0$

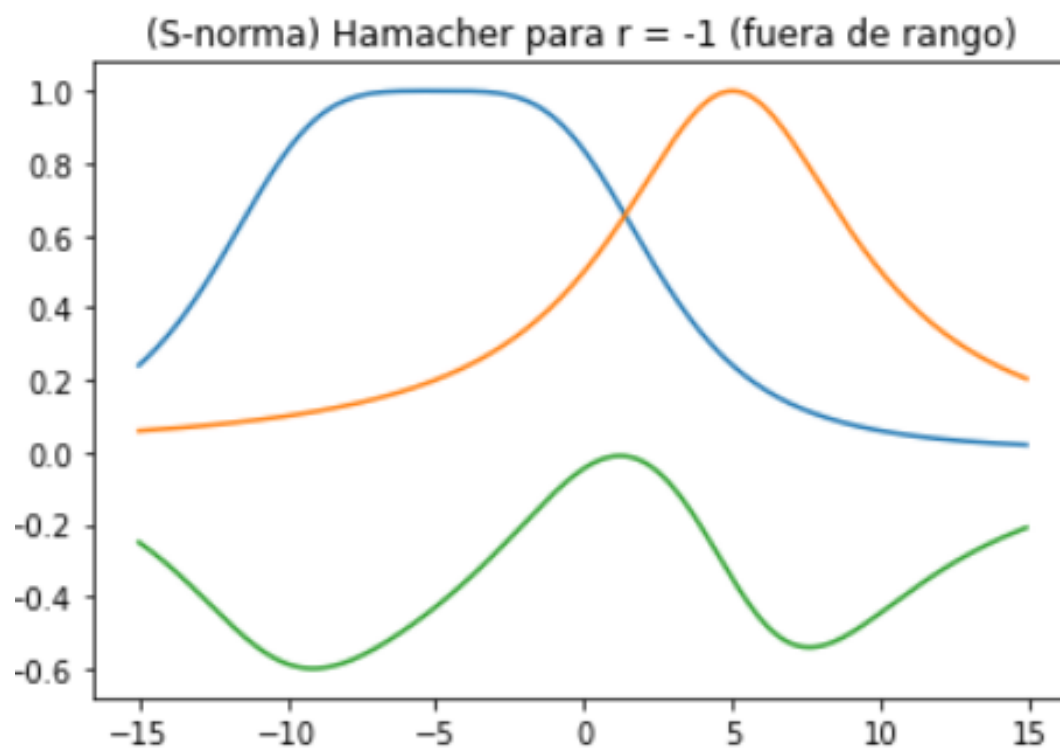


Ilustración 8 Operador de Hamacher con r fuera de rango

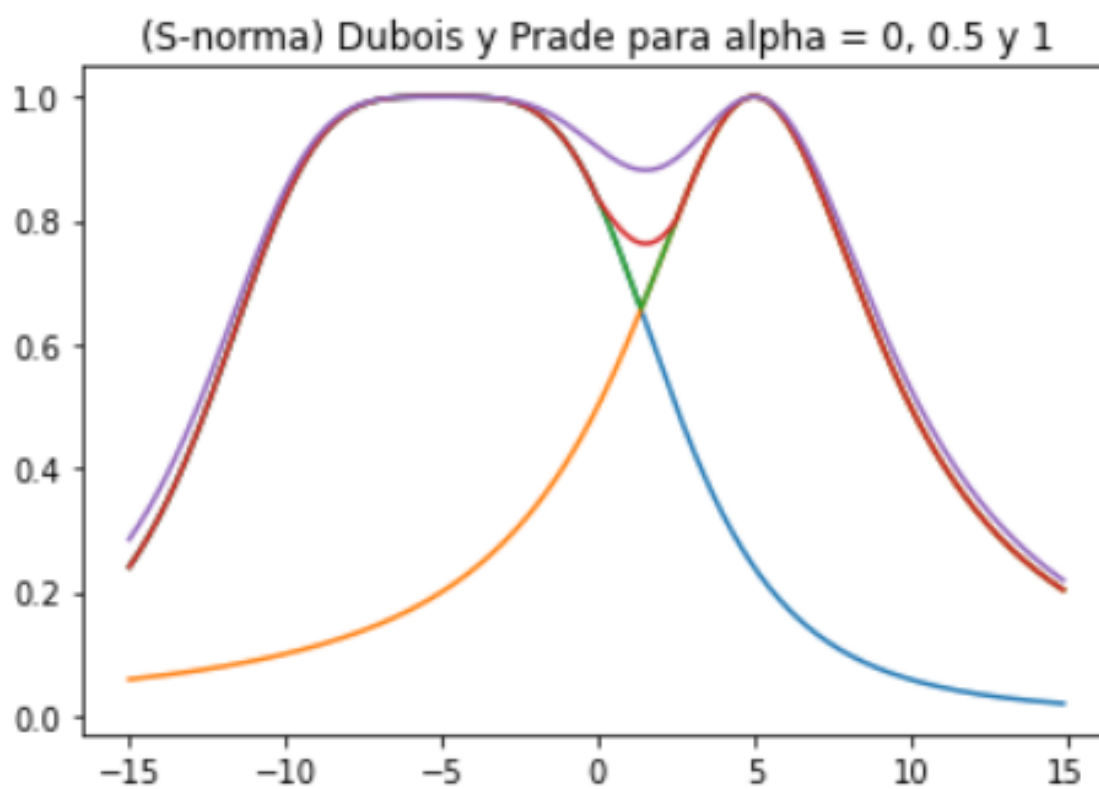


Ilustración 9 Operador de Dubois y Prade con α entre 0 y 1

(S-norma) Dubois y Prade para $\alpha = 5$ (fuera de rango)

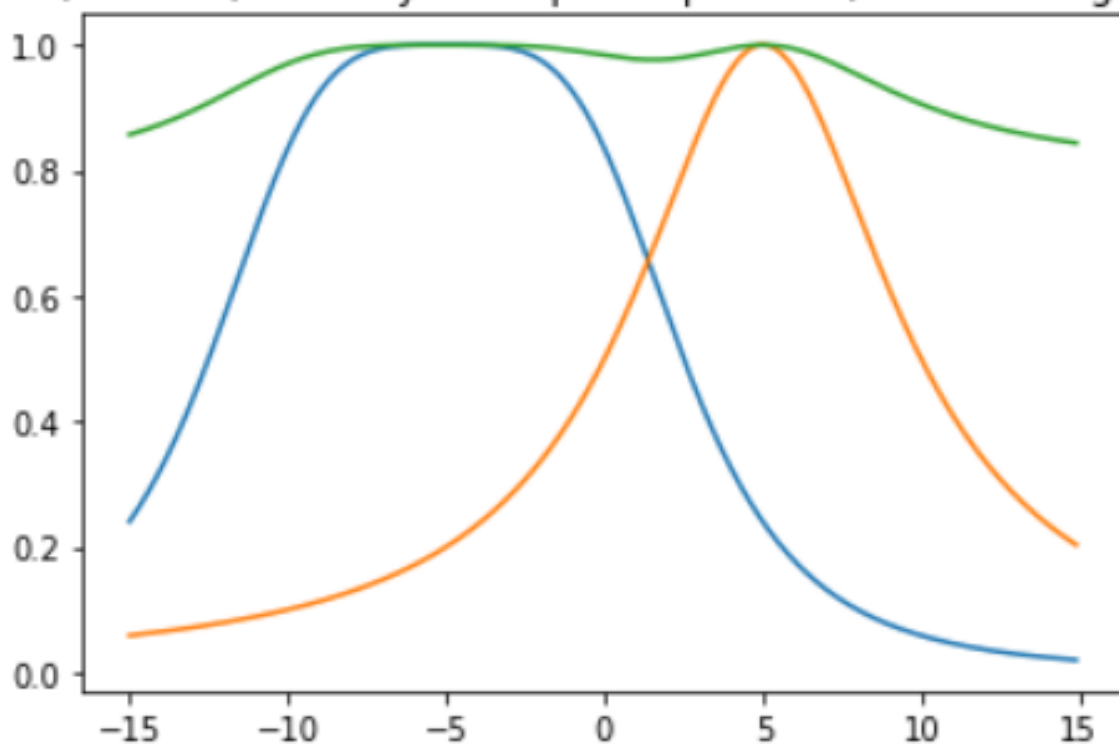


Ilustración 10 Operador de Dubois y Prade con α fuera de rango

(S-norma) Weber para $\lambda = -0.5, 0$ y 1

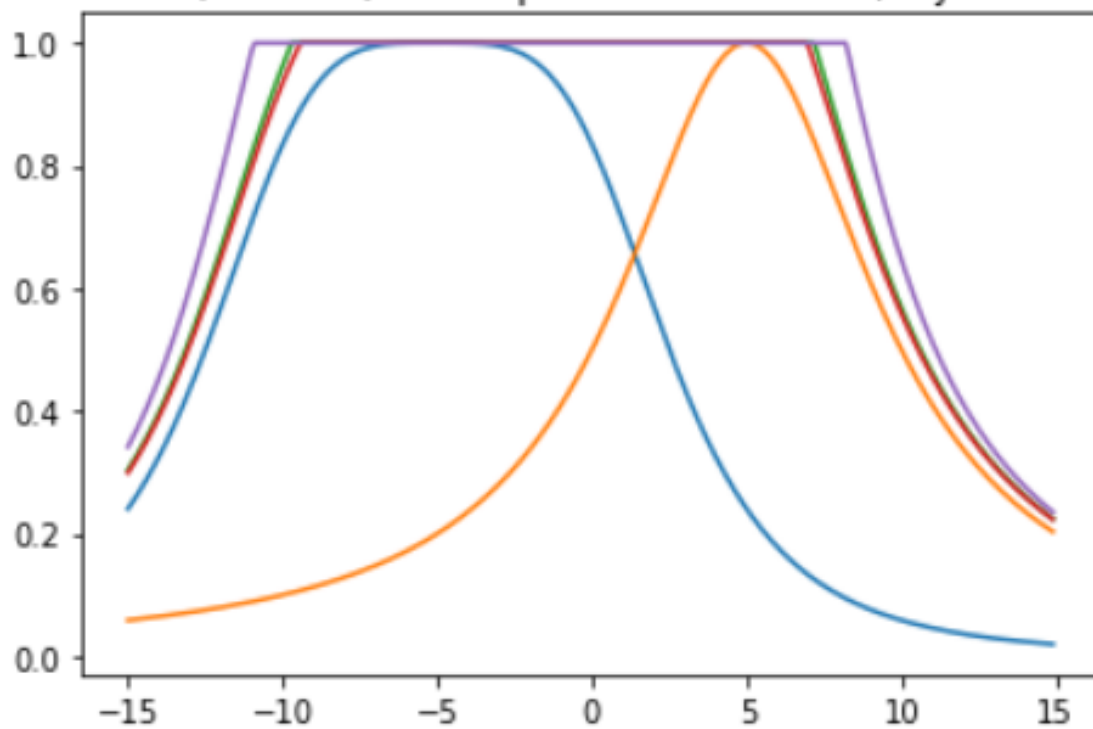


Ilustración 11 Operador de Weber con $\lambda > -1$

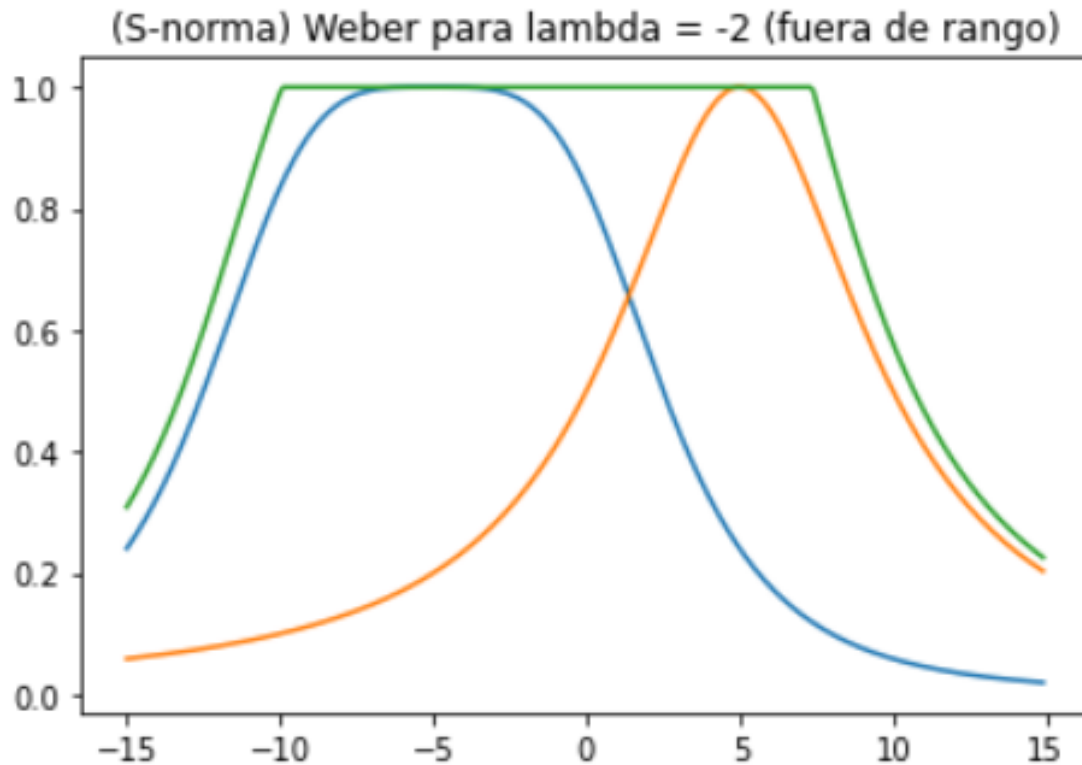


Ilustración 12 Operador de Weber con λ fuera de rango

Código de Python:

```
#S-norma

#Hamacher
HamS = np.zeros((4,len(x)))
#Dubois y Prade
DPS = np.zeros((4,len(x)))
#Weber
WebS = np.zeros((4,len(x)))

for i in range(len(x)):
    for j in range(4):
        HamS[j][i] = (A[i]+B[i]+(r[j]-2)*A[i]*B[i])/(r[j]+(r[j]-1)*A[i]*B[i])
        DPS[j][i] = 1-((1-A[i])*(1-B[i]))/(np.maximum(1-A[i],np.maximum(1-B[i],alfa[j])))
        WebS[j][i] = np.minimum(1,A[i]+B[i]-lamda[j]/(1-lamda[j])*A[i]*B[i])

plt.figure(8)
plt.title("(S-norma) Hamacher para r = 0.5, 1 y 3")
plt.plot(x,HamS[0],x,HamS[1],x,HamS[2])
```



```
plt.figure(9)
plt.title("(S-norma) Hamacher para  $r = -1$  (fuera de rango)")
plt.plot(x, HamS[3])

plt.figure(10)
plt.title("(S-norma) Dubois y Prade para  $\alpha = 0, 0.5$  y  $1$ ")
plt.plot(x, DPS[0], x, DPS[1], x, DPS[2])
plt.figure(11)
plt.title("(S-norma) Dubois y Prade para  $\alpha = 5$  (fuera de rango)")
plt.plot(x, DPS[3])

plt.figure(12)
plt.title("(S-norma) Weber para  $\lambda = -0.5, 0$  y  $1$ ")
plt.plot(x, WebS[0], x, WebS[1], x, WebS[2])
plt.figure(13)
plt.title("(S-norma) Weber para  $\lambda = -2$  (fuera de rango)")
plt.plot(x, WebS[3])
```