# Prova

June 14, 2022


- Aluno: Vítor Gabriel Reis Caitité
- Matrícula: 2021712430


## 1 Implementação das funções de pertinência

[1]:
```
from matplotlib import pyplot as plt
import numpy as np
from math import *
from sklearn.metrics import mean_squared_error
import skfuzzy as fuzz
from skfuzzy import control as ctrl
```

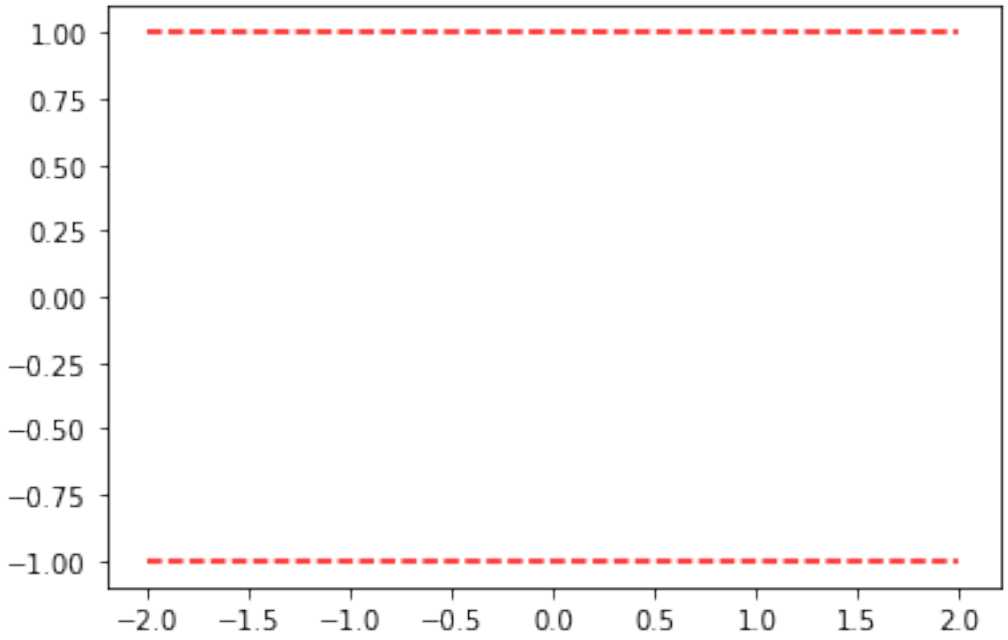
[2]:
```
def trimf(x,a,b,c):
    y = np.zeros(len(x))
    for i in range(len(x)):
        y[i] = max([min([(x[i]-a)/(b-a), (c-x[i])/(c-b)]), 0])
    return y

def gaussmf(x,c,sigma):
    return e**(-1/2 * ((x-c)/sigma)**2)

def trapmf(x,a,b,c,d):
    y = np.zeros(len(x))
    for i in range(len(x)):
        if x[i] > a and x[i] <= b:
            y[i] = (x[i]-a)/(b-a)
        if x[i] > b and x[i] <= c:
            y[i] = 1
        if x[i] > c and x[i] < d:
            y[i] = (-x[i]+d)/(-c+d)
    return y

def gbellmf(x, a, b, c):
    return 1/(1+abs(((x-c)/a)**(2*b)))

def sigmf(x, c, a):
    return 1/(1 + e**(-a*(x-c)))
```

## 1.1 Mecanismo de Inferência de Sugeno (Constante)

### 1.1.1 Definindo y1 e y2

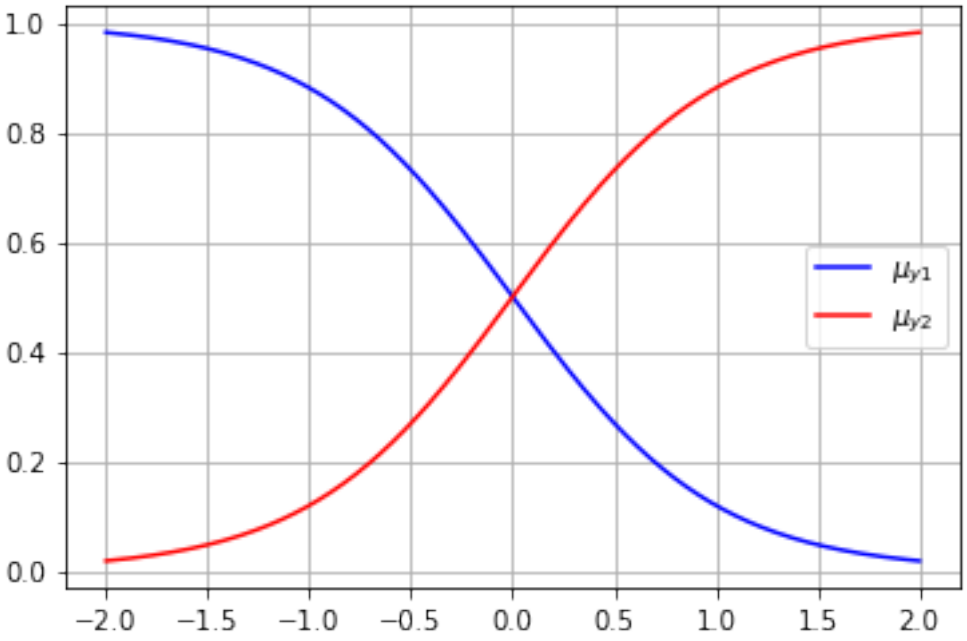
[3]:
```
x = np.linspace(-2,2, 200)
y = np.linspace(-1, 1, 200)
y1 = np.ones(len(x))*-1
y2 = np.ones(len(x))*1
plt.plot(x,y1,'r--', x, y2, 'r--')
```

[3]:
```
[<matplotlib.lines.Line2D at 0x7f8674f50320>,
 <matplotlib.lines.Line2D at 0x7f8674f504e0>]
```



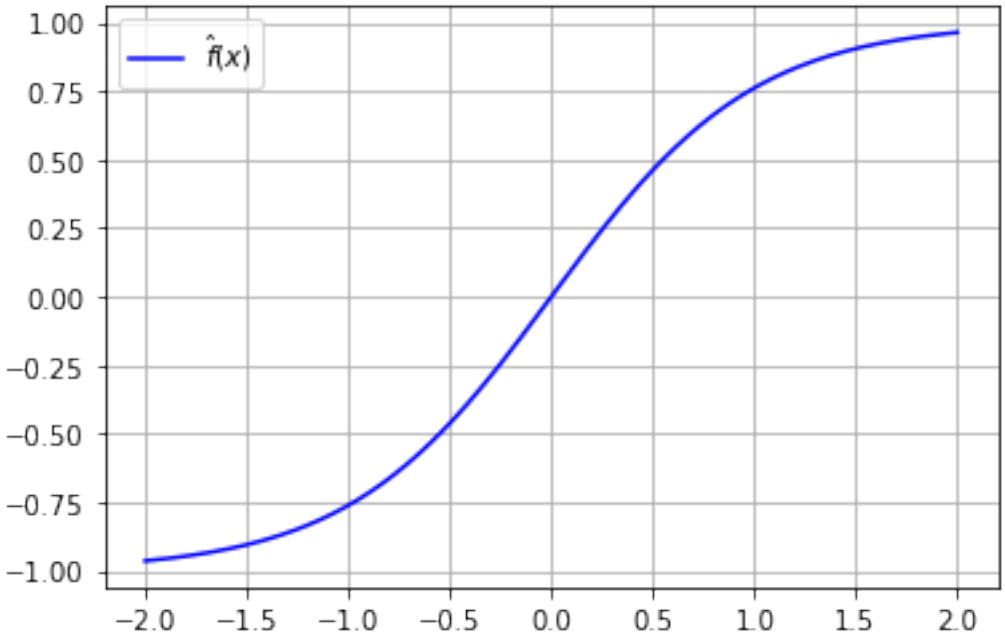

### 1.1.2 Funções de pertinencia

[6]:
```
mu_y1 = sigmf(x, 0, -2)
mu_y2 = sigmf(x, 0, 2)
plt.plot(x,mu_y1,'b-',x,mu_y2,'r-')
legend = plt.legend([r'$\mu_{y1}$', r'$\mu_{y2}$'])
plt.grid()
```




### 1.1.3 Resultado

[7]:
```
div = mu_y1 + mu_y2
div[div==0] = 0.00001
y_hat = np.divide(np.multiply(mu_y1, y1) + np.multiply(mu_y2, y2), div )
plt.plot(x,y_hat,'b-')
legend = plt.legend([r'$\hat{f}(x)$'])
plt.grid()
```


$\hat{f}(x)$
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0