

ML_CLASIFICACION_RN_01

Red Neuronal de una sola neurona (Perceptron)

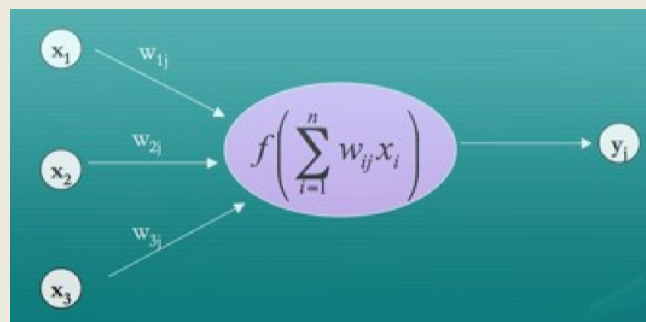
ML

Esta práctica consiste en simular lo que hace una red neuronal formada por una sola neurona (perceptron).

La capa de entrada (inputs o atributos), tiene como datos: x_1 , x_2 y x_3 .

Sus conexiones con la neurona tienen los pesos: w_1 , w_2 y w_3 .

Se evalúa la función de activación ' $h()$ ' considerando la tangente hiperbólica (\tanh).



Como resultado se muestra el valor de salida o output ('y')

SOLUCIÓN

Importar las librerías necesarias para realizar la práctica.

```
# Librerías
import numpy as np
```

Definir el perceptron (neurona simple) como una función.

```
def perceptron(W,X):
    # Multiplicar W*X elemento a elemento los vectores = w1*x1+w2*x2+w3*x3
    s= np.sum(W*X)
    print("La suma w1.x1+w2.x2+w3.x3 es: ",s)
    # Aplicar la función de activación (por ejemplo tanh) al resultado 's'
    pred = np.tanh(s)
    return pred
```

Valores de entradas (inputs) y pesos (w).

```
# Inputs 'x'
x1 = 0.9
x2 = 0.3
x3 = 0.5
X=[x1,x2,x3]
X = np.column_stack((x1,x2,x3))    # en forma matricial
print("Inputs :",X)
```

```
# Pesos 'w'
w1=1
w2=0.4
w3=0.3
W =[w1,w2,w3]
print("Pesos :",W)
```

Evaluar la Red Neuronal.

```
y = perceptron(W,X)
print("El output 'y' de la Red Neuronal es: ",y)

Inputs : [[0.9 0.3 0.5]]
Pesos : [1, 0.4, 0.3]
La suma  $w1.x1+w2.x2+w3.x3$  es: 1.17
El output 'y' de la Red Neuronal es: 0.8242721703413975
```