

▼ Cálculo do gradiente

```
import numpy as np

def sigmoid(x):
    return 1/(1+np.exp(-x))

def sigmoid_prime(x):
    return sigmoid(x) * (1 - sigmoid(x))

learnrate = 0.5
x = np.array([4, 3, 2, 4])
y = np.array(0.5)
b = 0.5

# Pesos iniciais
w = np.array([0.25, -0.45, 0.33, 0.21])

h = np.dot(x, w)+b

nn_output = sigmoid(h)
print(nn_output)

    0.8388910504234147

# TODO: Calcule o erro da Rede Neural
error = y - nn_output
print(error)

    -0.3388910504234147

# TODO: Calcule o termo de erro
error_term = error * sigmoid_prime(h)

# TODO: Calcule a variação do peso
del_w = learnrate * error_term * x

print(del_w)

    [-0.09160419 -0.06870314 -0.04580209 -0.09160419]
```



```
h = np.dot(x, w) + b
```

```
nn_output = sigmoid(h)  
print(nn_output)
```

```
0.650094109073785
```