

▼ 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([3, 1, 2, 4])
y = np.array(0.5)
b = 0.5

# Pesos iniciais
w = np.array([0.1, -0.2, 0.3, 0.4])

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

nn_output = sigmoid(h)
print(nn_output)

    0.9426758241011313

# TODO: erro Calcular de rede neural
error = y - nn_output
print(error)

    -0.44267582410113127

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

# TODO: Calcule a mudança nos pesos
del_w = learnrate * error_term * x

print(del_w)

    [-0.03588205 -0.01196068 -0.02392137 -0.04784273]

w = w + del_w
```



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

```
0.9199140268993846
```