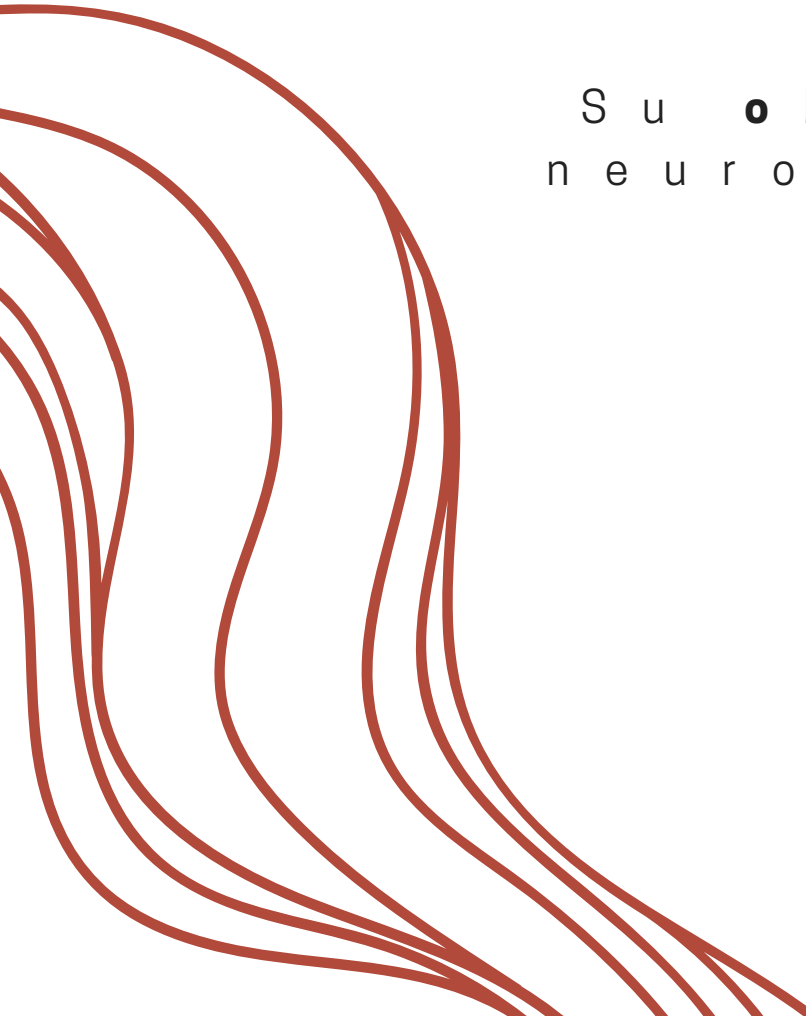




B A C K P R O P A G A T I O N

S u **o b j e t i v o** es **o p t i m i z a r** los **p e s o s** para que la red
n e u r o n a l pueda aprender cómo asignar correctamente
e n t r a d a s a r b i t r a r i a s a s a l i d a s .



Datos iniciales

Pesos de la capa de entrada:

[-80,20,20,-80]

[-80,20,-80,-80]

[-80,-80,-80,20]

[-80,40,-80,20]

[20,20,-80,20]

[30,30,-80,10]

Bias de la capa de entrada:

[-30,-10,-10,-30,-50,-60]

Pesos de la capa de salida:

[30,30,30,30,30,30]

Bias de la capa de salida:

[-10]

Entradas de la red:

[(0,0,0,0),(0,0,0,1),(0,0,1,0),(0,0,1,1),(0,1,0,0),(0,1,0,1),(0,1,1,0),(0,1,1,1),
(1,0,0,0),(1,0,0,1),(1,0,1,0),(1,0,1,1),(1,1,0,0),(1,1,0,1),(1,1,1,0),(1,1,1,1)]

Salidas esperadas:

[[0],[1],[0],[0],[1],[1],[1],[0],[0],[0],[0],[0],[1],[1],[0],[0]]

Pasos:



1. Aplicar las entradas a la red neuronal.
2. Calcular el error entre la salida esperada y la salida obtenida.
3. Propagar el error hacia atrás a través de la red.
4. Calcular los gradientes para cada uno de los pesos y los bias.
5. Ajustar los pesos y los bias en la dirección opuesta al gradiente para reducir el error.
6. Calcular la nueva salida de la red, junto con su error.
7. Repetir los pasos 1 a 6 varias veces hasta que el error sea lo suficientemente pequeño.

Paso 1

Aplicar las entradas a la red neuronal y calcular las salidas.

Se deben calcular las activaciones de las 6 neuronas de la capa oculta.

Esto se consigue mediante el uso de dos fórmulas:

- Aplicar la entrada a la neurona
 - **net** = $w1 * i1 + w2 * i2 + w3 * i3 + w4 * i4 + bias * 1$
- Calcular la activación de la neurona
 - **out** = $1 / (1 + \exp(-net))$
 - -----> $out = \frac{1}{1 + e^{-net}}$

Paso 2

Calcular el error entre la salida esperada y la salida obtenida.

Calcular el error total

• -----> $E_{total} = \sum \frac{1}{2} (output - target)^2$



Paso 3



Propagar el error hacia atrás a través de la red.

El objetivo es actualizar cada uno de los pesos para que hagan que la salida real esté más cerca de la salida objetivo; es decir, reducir el error.

Para actualizar los pesos se usa el **Descenso por gradiente**.

En nuestro caso queremos minimizar la función de error. Para encontrar un mínimo local de una función mediante el descenso de gradiente, se toman pasos proporcionales al negativo del gradiente de la función en el punto actual.

Paso 3

Fórmulas

La fórmula general para la actualización de un peso es:

$$^*W_x = W_x - a \left(\frac{\partial Error}{\partial W_x} \right)$$

Diagram illustrating the weight update formula with annotations:

- Peso antiguo** (Old weight) points to W_x .
- Derivada del error con respecto al peso** (Derivative of error with respect to the weight) points to $\left(\frac{\partial Error}{\partial W_x} \right)$.
- Peso nuevo** (New weight) points to *W_x .
- learning rate** points to a .

Fórmulas

Se evalua la derivada del error con la regla de la cadena.

Ejemplo para un peso "W6" conectado a una capa oculta "h2":

$$\frac{\partial Error}{\partial W_6} = \frac{\partial Error}{\partial \text{output}} * \frac{\partial \text{output}}{\partial W_6} \quad \leftarrow \text{Regla de la cadena}$$

$$\frac{\partial Error}{\partial W_6} = \frac{\partial \frac{1}{2}(\text{output} - \text{target})^2}{\partial \text{output}} * \frac{\partial \text{output}}{\partial W_6}$$

$$\frac{\partial Error}{\partial W_6} = (\text{output} - \text{target}) * (h_2)$$

$$\frac{\partial Error}{\partial W_6} = \Delta h_2 \quad \leftarrow \text{Delta}$$

$\Delta = \text{output} - \text{target}$

Fórmulas

Ejemplo para un peso "W3" conectado a la entrada "E2", a la capa oculta "h2" y dicha capa oculta tiene el peso "W6":

$$\frac{\partial Error}{\partial W_1} = \frac{\partial Error}{\partial \text{output}} * \frac{\partial \text{output}}{\partial h1} * \frac{\partial h1}{\partial W_1} \quad \text{Regla de la cadena}$$

$$\frac{\partial Error}{\partial W_1} = \frac{\partial \frac{1}{2}(\text{output} - \text{target})^2}{\partial \text{output}} * \frac{\partial \text{output}}{\partial h1} * \frac{\partial h1}{\partial W_1}$$

$$\frac{\partial Error}{\partial W_1} = (\text{output} - \text{target}) * (w_5 i_1)$$

$$\frac{\partial Error}{\partial W_1} = \Delta w_5 i_1 \quad \Delta = \text{output} - \text{target} \quad \text{Delta}$$

Ajustar pesos

Capa **entrada**:

$$*w1: w1 - a (i1 \cdot \Delta w25)$$

$$*w2: w2 - a (i2 \cdot \Delta w25)$$

$$*w3: w3 - a (i3 \cdot \Delta w25)$$

$$*w4: w4 - a (i4 \cdot \Delta w25)$$

$$*w5: w5 - a (i1 \cdot \Delta w26)$$

$$*w6: w6 - a (i2 \cdot \Delta w26)$$

$$*w7: w7 - a (i3 \cdot \Delta w26)$$

$$*w8: w8 - a (i4 \cdot \Delta w26)$$

$$*w9: w9 - a (i1 \cdot \Delta w27)$$

$$*w10: w10 - a (i2 \cdot \Delta w27)$$

$$*w11: w11 - a (i3 \cdot \Delta w27)$$

$$*w12: w12 - a (i4 \cdot \Delta w27)$$

$$*w13: w13 - a (i1 \cdot \Delta w28)$$

$$*w14: w14 - a (i2 \cdot \Delta w28)$$

$$*w15: w15 - a (i3 \cdot \Delta w28)$$

$$*w16: w16 - a (i4 \cdot \Delta w28)$$

$$*w17: w17 - a (i1 \cdot \Delta w29)$$

$$*w18: w18 - a (i2 \cdot \Delta w29)$$

$$*w19: w19 - a (i3 \cdot \Delta w29)$$

$$*w20: w20 - a (i4 \cdot \Delta w29)$$

$$*w21: w21 - a (i1 \cdot \Delta w30)$$

$$*w22: w22 - a (i2 \cdot \Delta w30)$$

$$*w23: w23 - a (i3 \cdot \Delta w30)$$

$$*w24: w24 - a (i4 \cdot \Delta w30)$$

Capa **salida**:

$$*w25: w25 - a (h1 \cdot \Delta)$$

$$*w26: w26 - a (h2 \cdot \Delta)$$

$$*w27: w27 - a (h3 \cdot \Delta)$$

$$*w28: w28 - a (h4 \cdot \Delta)$$

$$*w29: w29 - a (h5 \cdot \Delta)$$

$$*w30: w30 - a (h6 \cdot \Delta)$$

w = peso

a = learning rate

h = neurona capa oculta

i = entrada

Δ = delta

Bias

La fórmula general para la actualización de un bias es:

$$^* \mathbf{B}_X = \mathbf{B}_X - a (\Delta)$$

Bias nuevo **Bias antiguo** **learning rate** **Delta**

The diagram illustrates the bias update formula: $^* \mathbf{B}_X = \mathbf{B}_X - a (\Delta)$. It includes four labels with arrows pointing to their respective parts in the formula: 'Bias nuevo' points to the asterisk on \mathbf{B}_X , 'Bias antiguo' points to the \mathbf{B}_X term, 'learning rate' points to the variable a , and 'Delta' points to the Δ term. The variable a is highlighted in yellow.



Ajustar bias

Capa **entrada**:

*bias1: bias1- a (Δ)

*bias2: bias2- a (Δ)

*bias3: bias3- a (Δ)

*bias4: bias4- a (Δ)

*bias5: bias5- a (Δ)

*bias6: bias6- a (Δ)

*bias7: bias7- a (Δ)





Paso 4



Calcular los gradientes para cada uno de los pesos y los bias.

Ahora es momento de calcular los nuevos valores para los pesos y los bias.



Paso 5



Ajustar los pesos y los bias en la dirección opuesta al gradiente para reducir el error.

Una vez calculados los nuevos valores, hay que actualizarlos en el esquema de la red.



Paso 6



Calcular la nueva salida de la red, junto con su error.

Ahora se debe calcular la nueva salida de la red y su error, utilizando la red con los nuevos valores para los pesos y los bias.



Paso 7



Repetir los pasos 1 a 6 varias veces hasta que el error sea lo suficientemente pequeño.