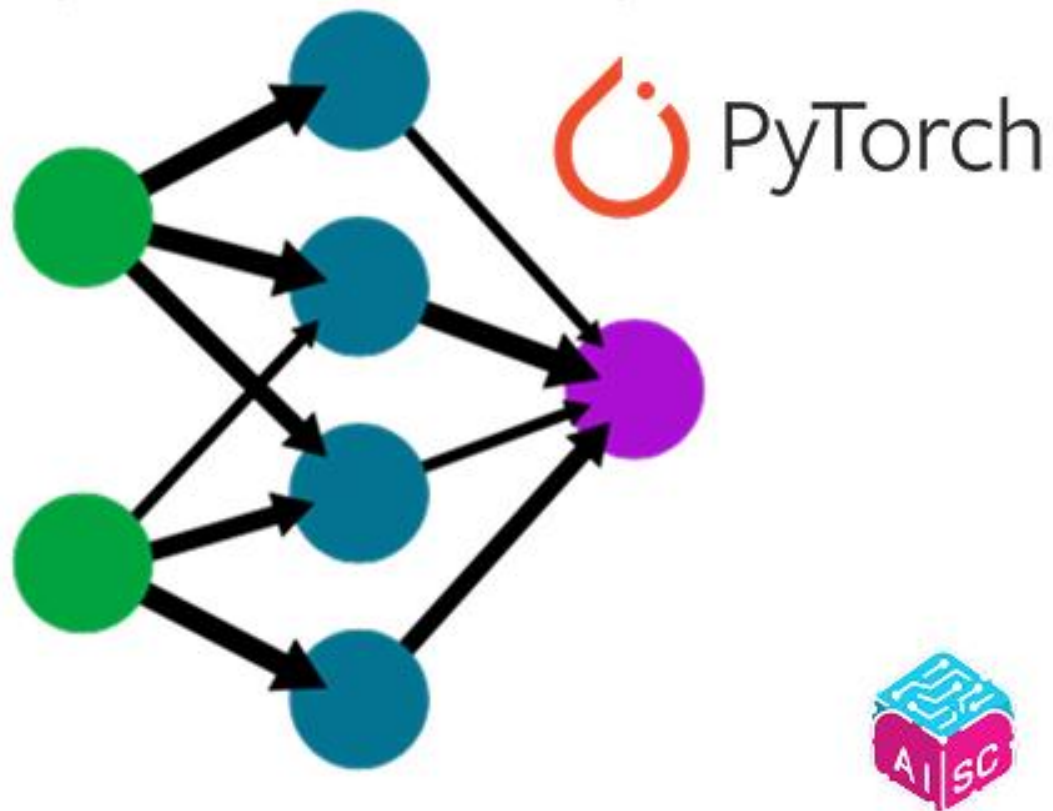


# Introducción a la IA & ML III: Redes Neuronales





**Hugo Centeno Sanz**  
**4to Ingeniería de Datos y Teleco**



IA

IA

IA

IA

IA

IA





**Sreekanth Kumbha** · 3rd+

1w ...

Consultant- Technology Management ,SeMT i...

I can suggest an equation that has the potential to impact the future:

$$E = mc^2 + AI$$

This equation combines Einstein's famous equation  $E=mc^2$ , which relates energy (E) to mass (m) and the speed of light (c), with the addition of AI (Artificial Intelligence). By including AI in the equation, it symbolizes the increasing role of artificial intelligence in shaping and transforming our future. This equation highlights the potential for AI to unlock new forms of energy, enhance scientific discoveries, and revolutionize various fields such as healthcare, transportation, and technology.

Like ·   3 | Reply · 3 replies



**Taosif Ahsan** (He/Him) · 3rd+

4d ...

Physics, MIT PhD| Physics/CS, Princeton...

What

Like ·   5 | Reply



# Introducción y Objetivos

- Conceptos fundamentales
- Optimización único parámetro con backpropagation
- Optimización múltiples parámetros con backpropagation





# Dinámica

<http://www.aiscmadrid.com/events/evento.php?id=14>

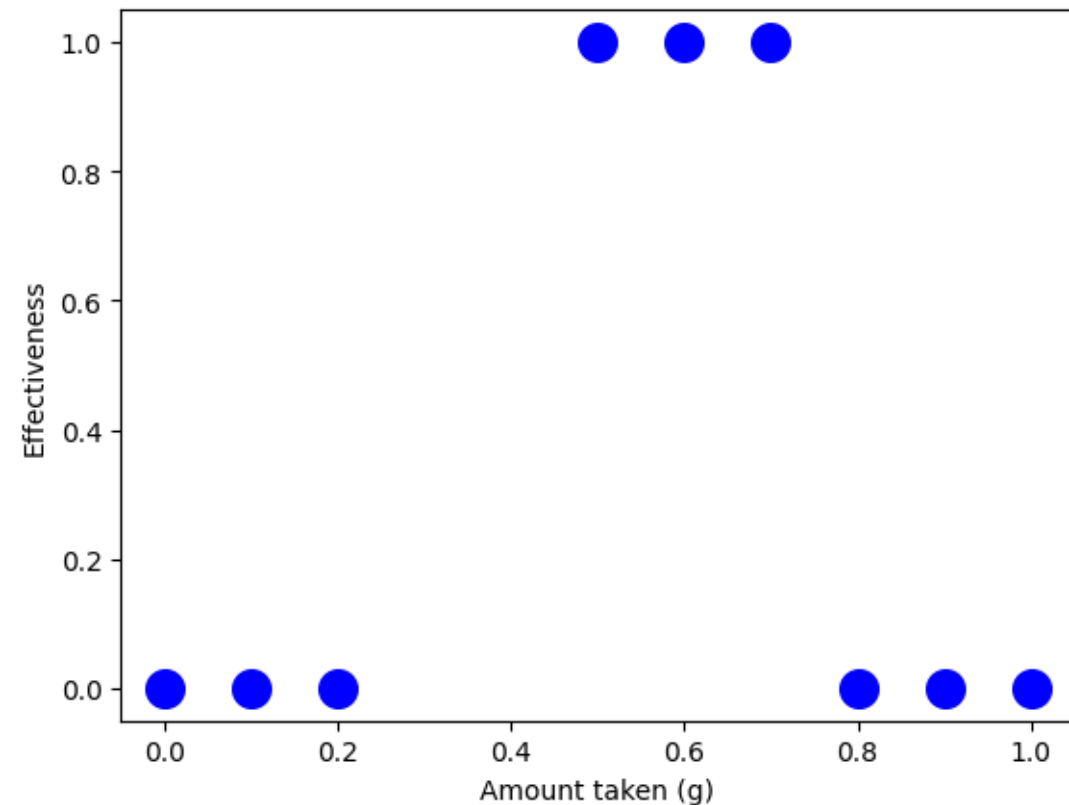


# Conceptos fundamentales

Datos

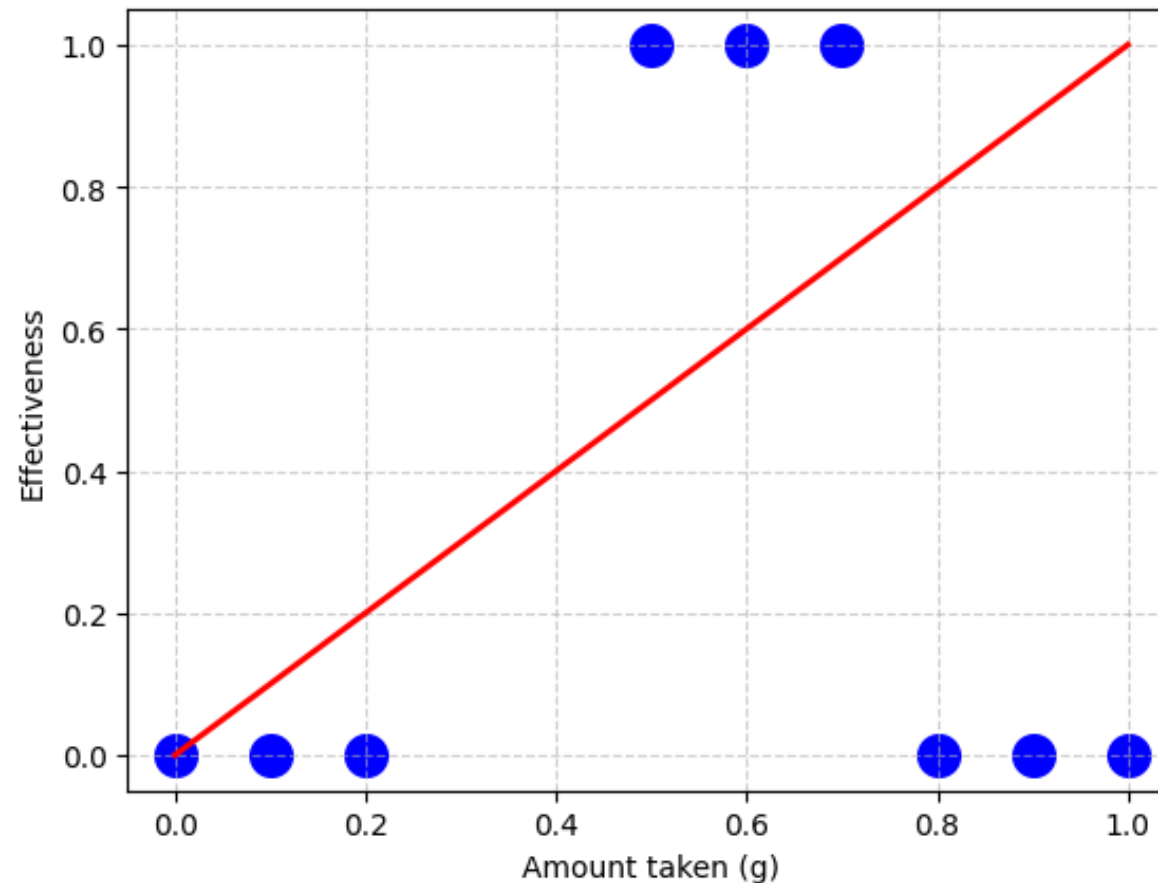
	Amount taken (g)	% Effectiveness
0	0.0	0.0
1	0.1	0.0
2	0.2	0.0
3	0.5	1.0
4	0.6	1.0
5	0.7	1.0
6	0.8	0.0
7	0.9	0.0
8	1.0	0.0

Modelarlos



# Conceptos fundamentales

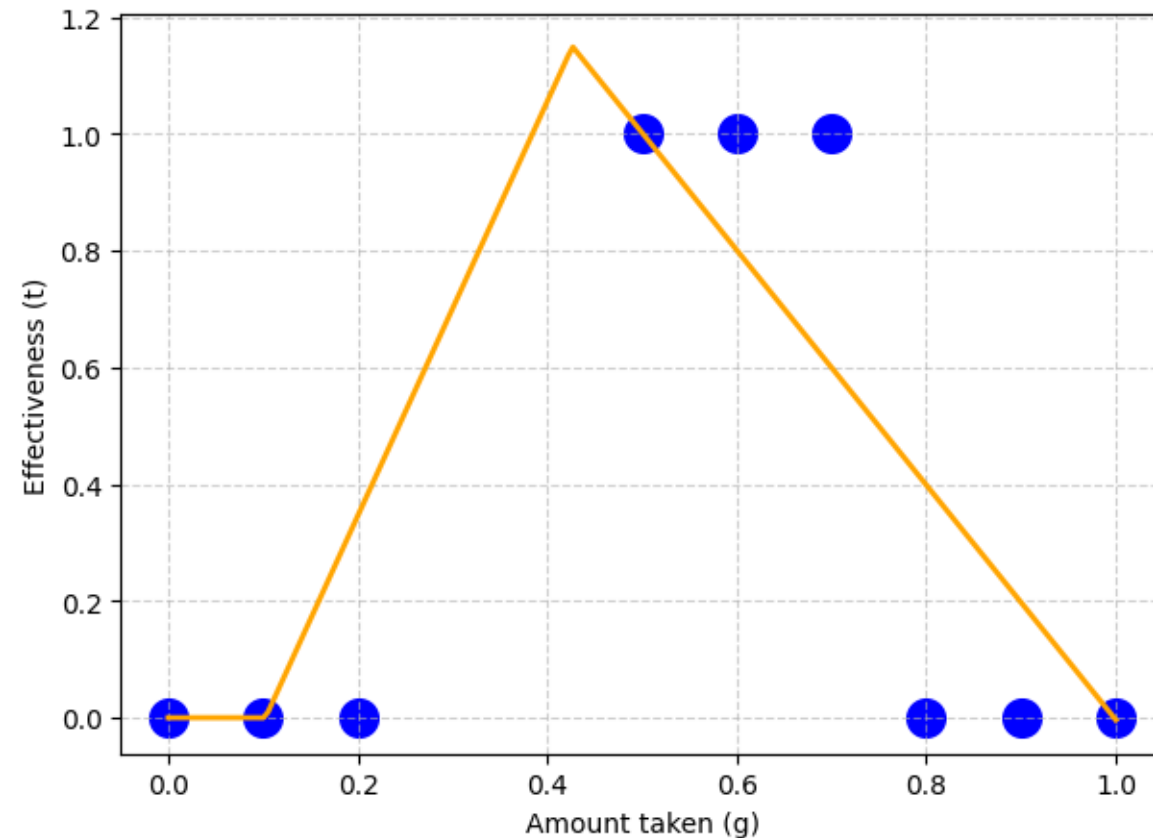
Mala predicción





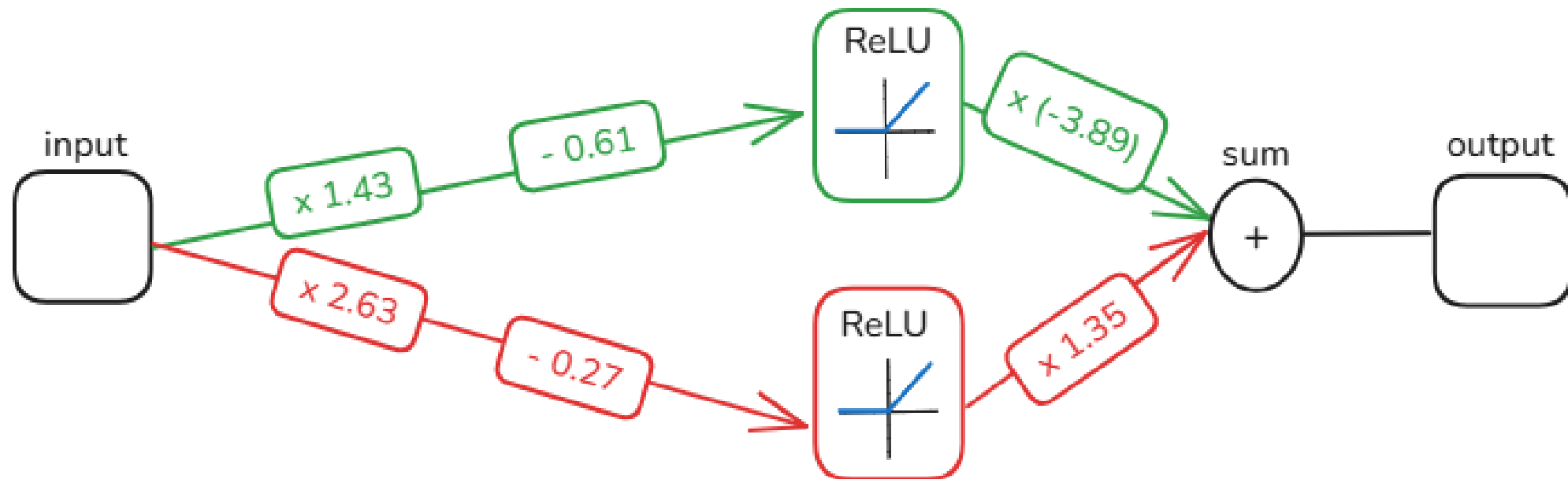
# Conceptos fundamentales

Predicción de una NN (Neural Network)



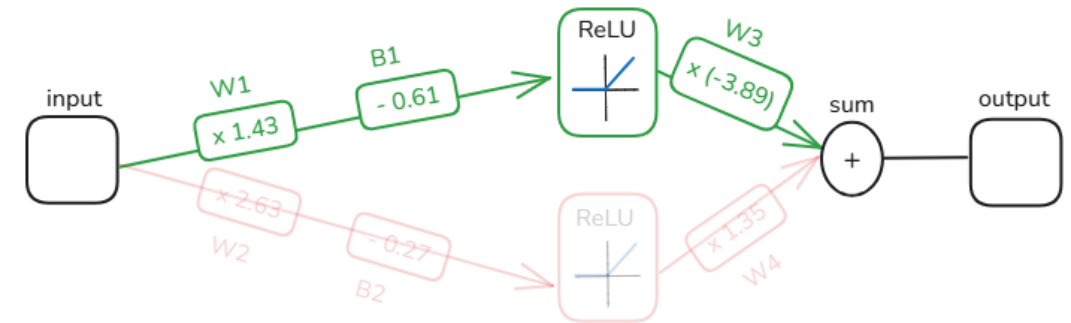
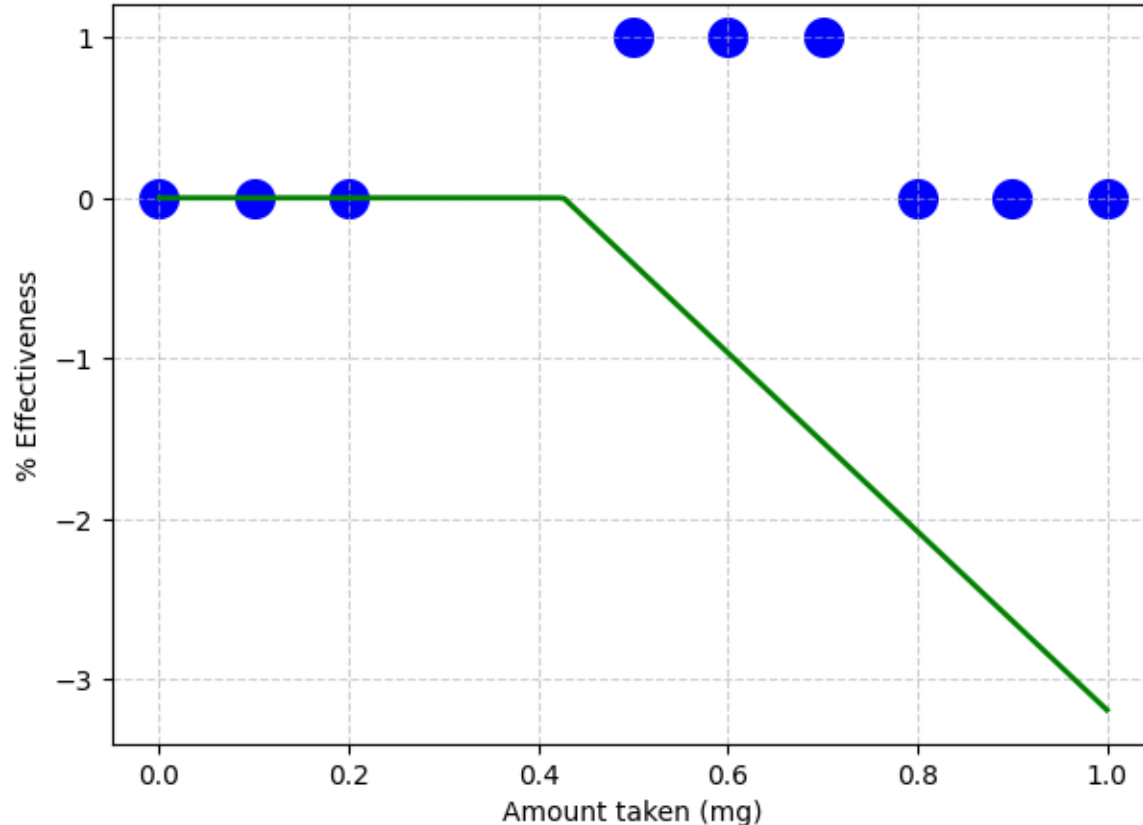
# Conceptos fundamentales

Predicción de una NN (Neural Network)



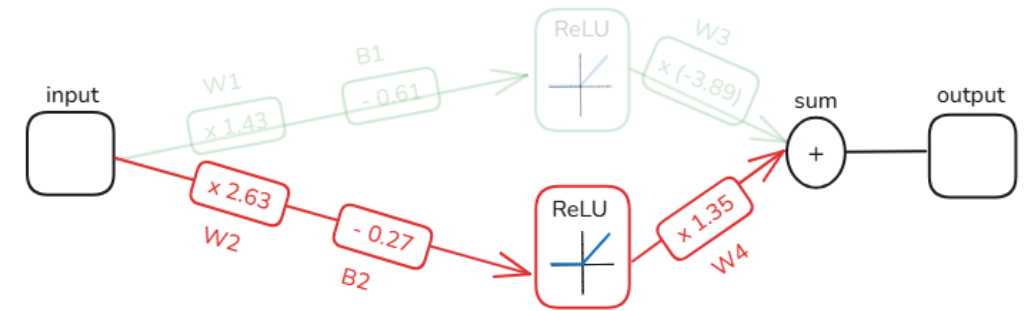
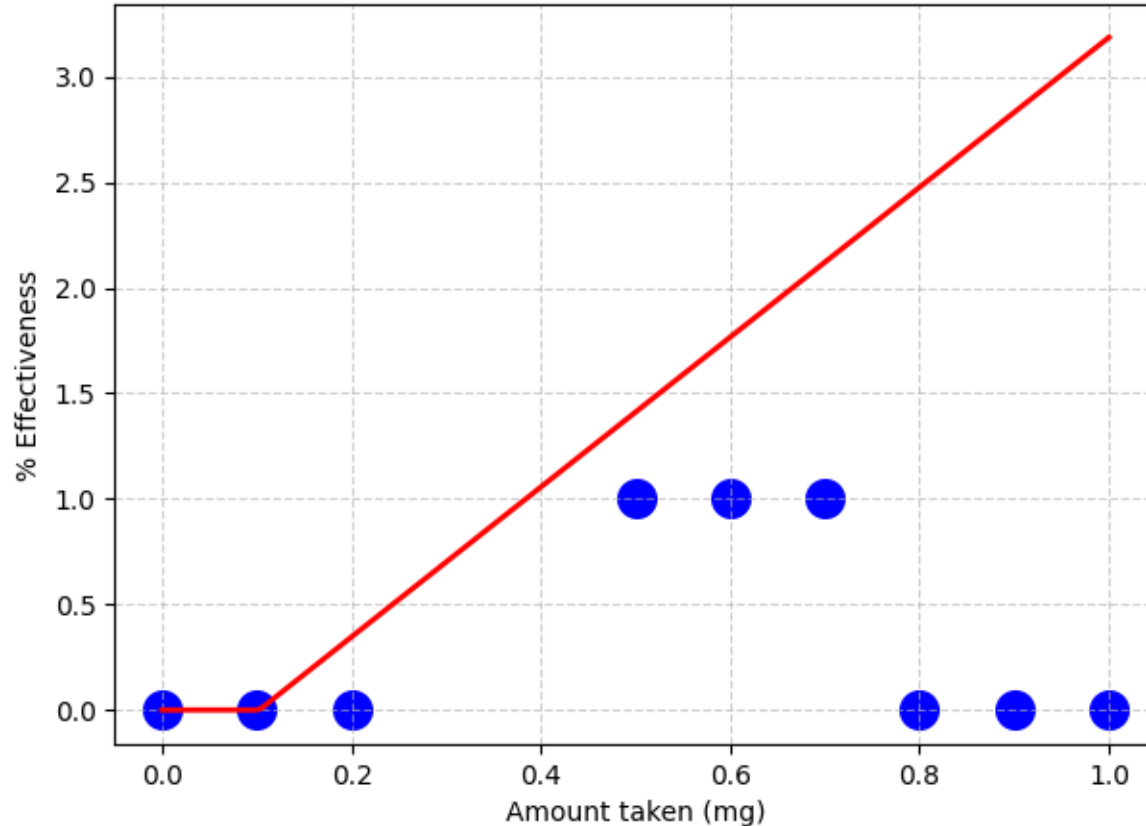
# Conceptos fundamentales

## Predicción de una NN (Neural Network)



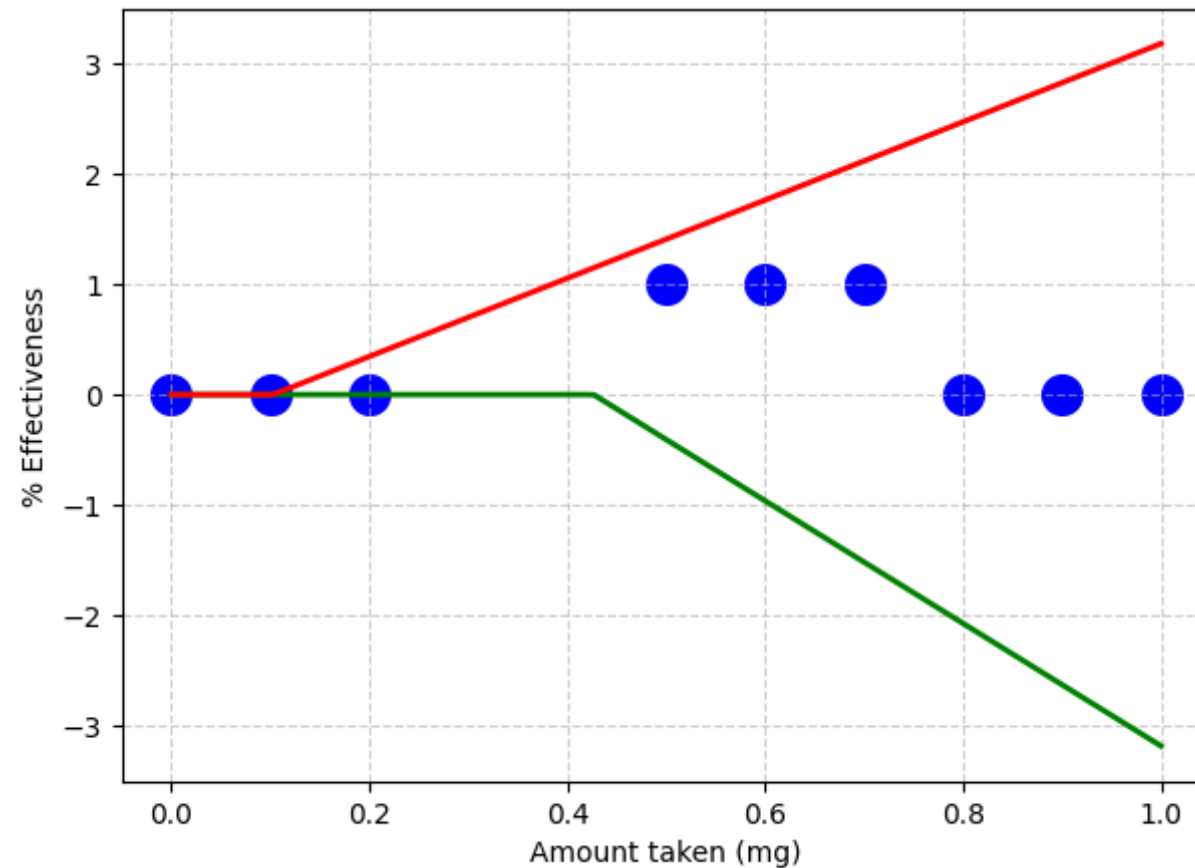
# Conceptos fundamentales

## Predicción de una NN (Neural Network)



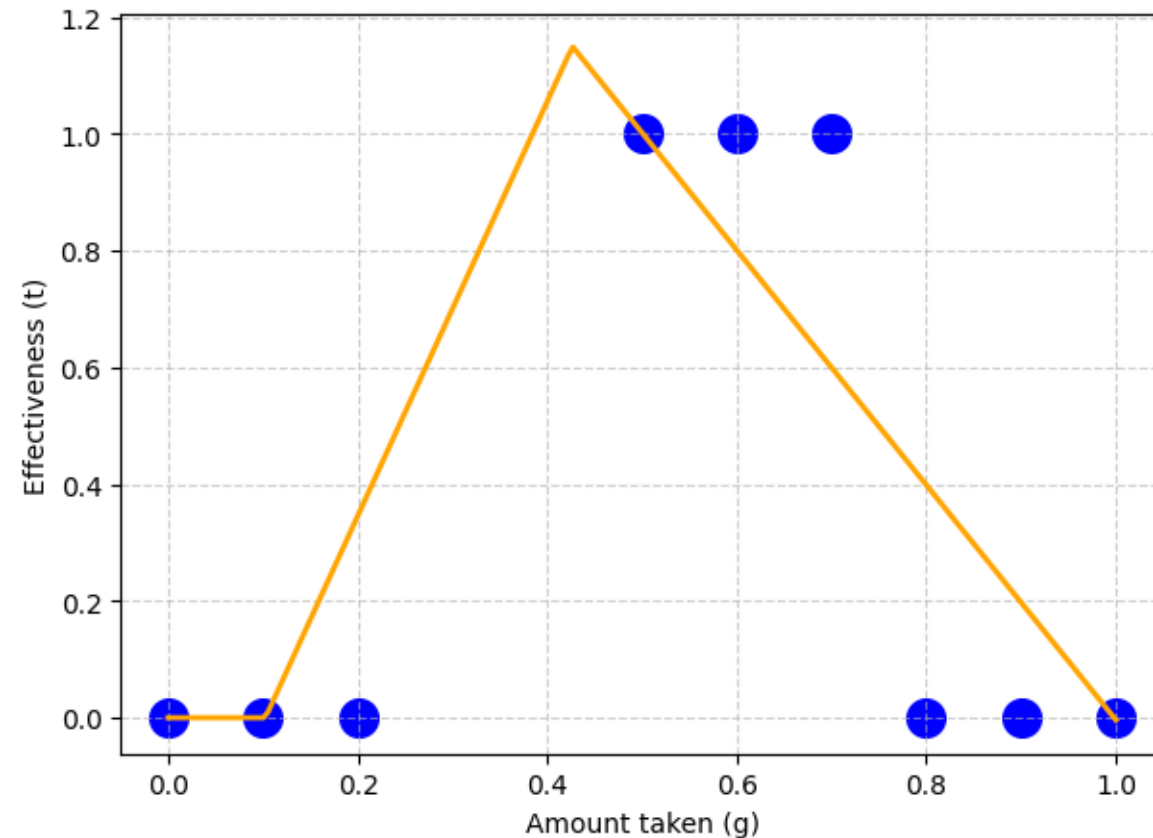
# Conceptos fundamentales

Predicción de una NN (Neural Network)



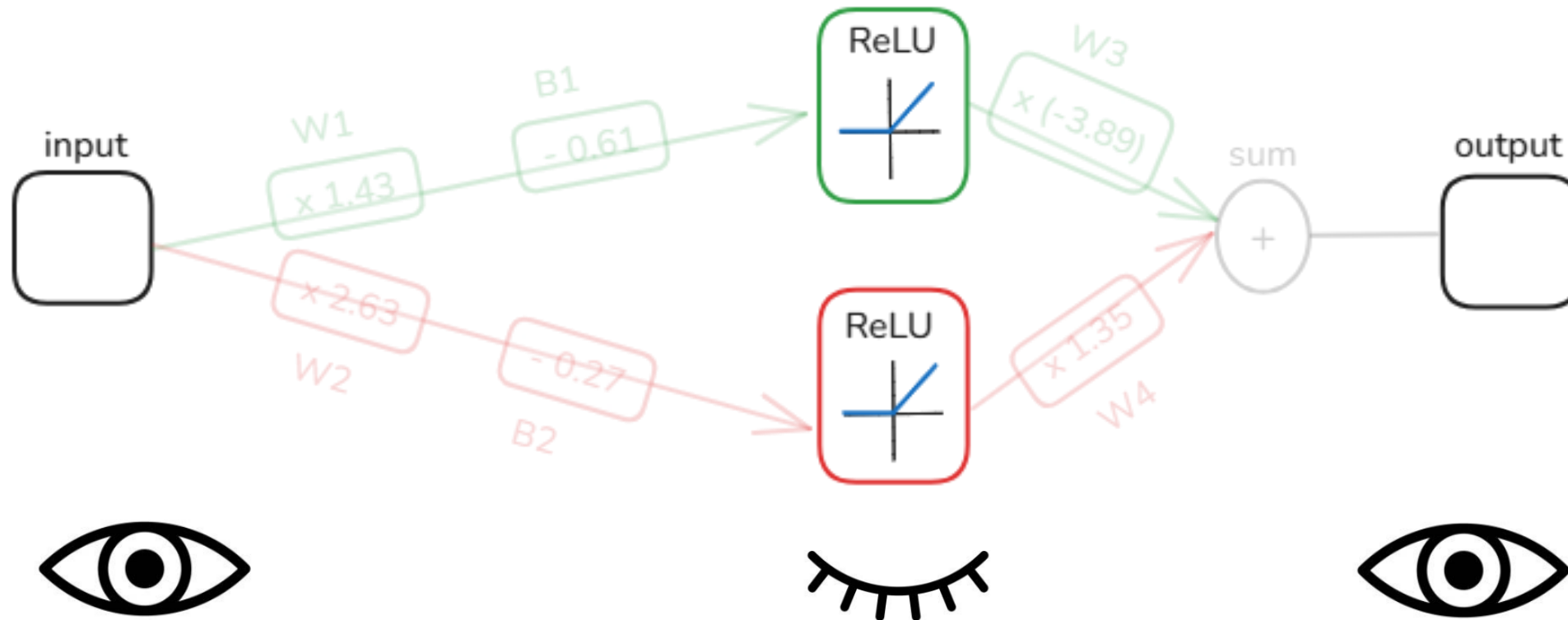
# Conceptos fundamentales

Predicción de una NN (Neural Network)



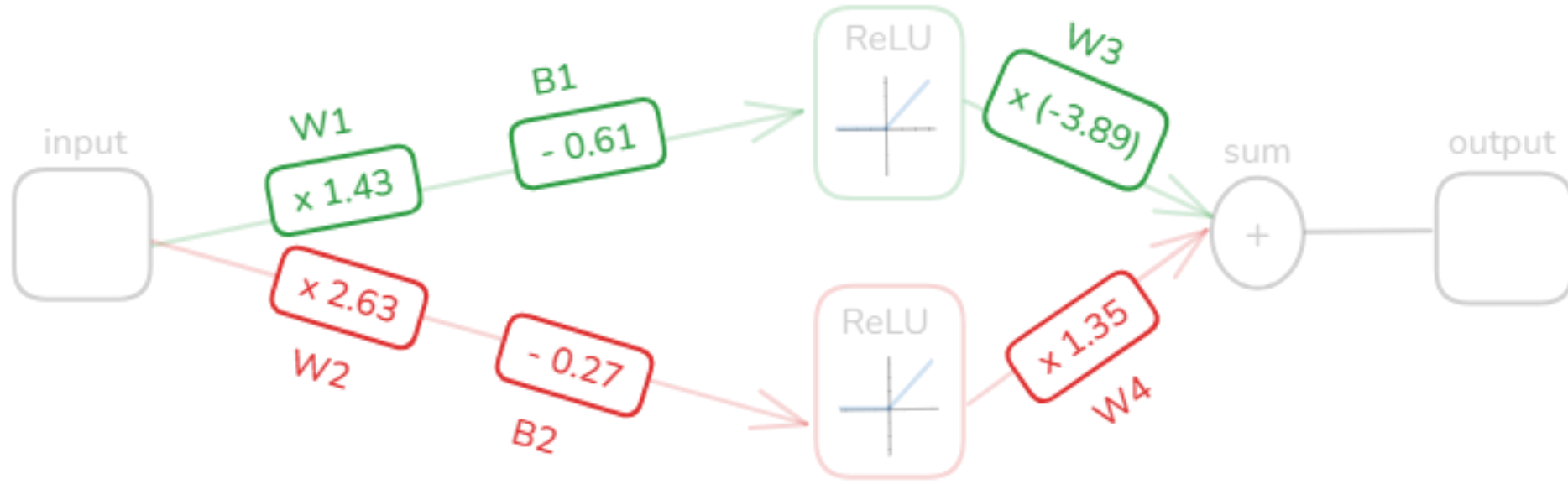
# Conceptos fundamentales

## Nodos



# Conceptos fundamentales

Weights  $W1, W2, W3, W4$ (pesos)  
Biases  $B1, B2$  (sesgos) } Parameters





# Conceptos fundamentales

Entrenar un modelo = Modificar el valor de los parámetros



# Conceptos fundamentales

¿Cuántos parámetros?

“We are releasing two model sizes: gpt-oss-**120b**, which consists of **36 layers**...”[1]

[1] gpt-oss-120b & gpt-oss-20b Model Card. OpenAI. <https://arxiv.org/pdf/2508.10925>



# Conceptos fundamentales

Activation functions (funciones de activación)

**ReLU** (Rectified Linear Unit)

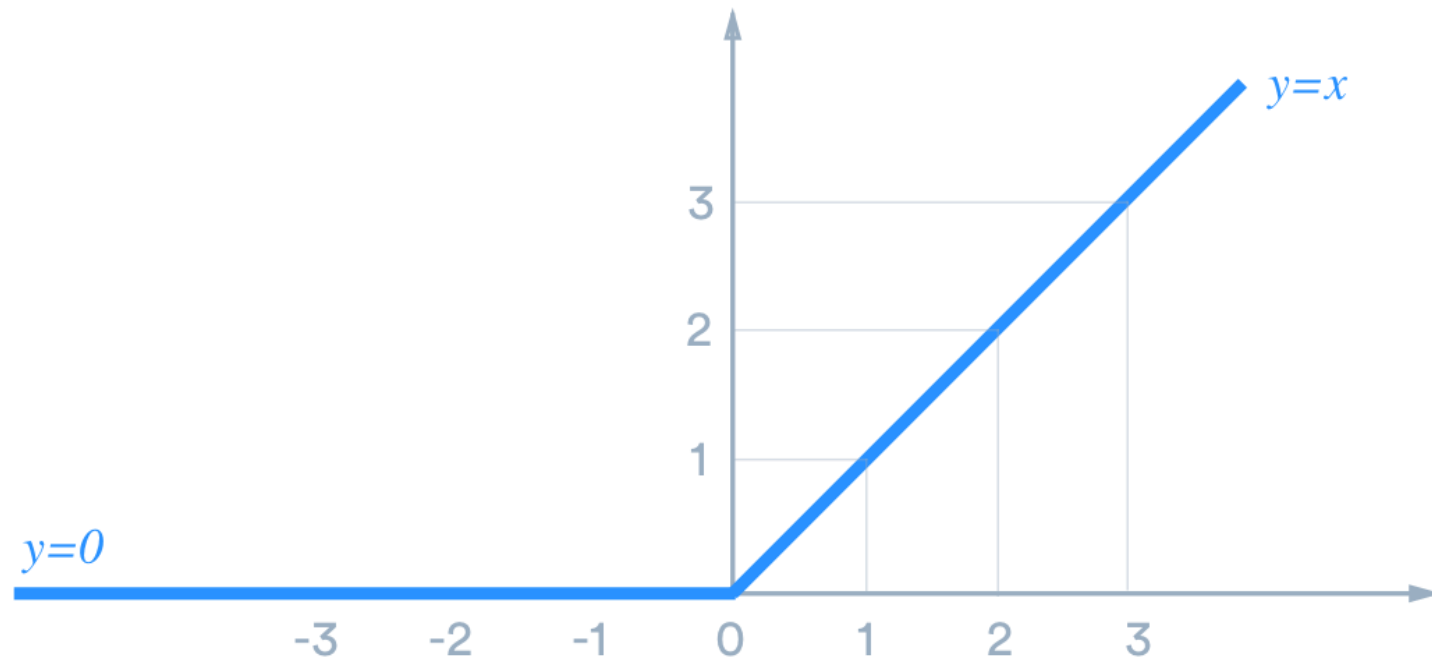
$$f(x) = \max(0, x)$$



# Conceptos fundamentales

Activation functions (funciones de activación)

**ReLU** (Rectified Linear Unit)



# Conceptos fundamentales

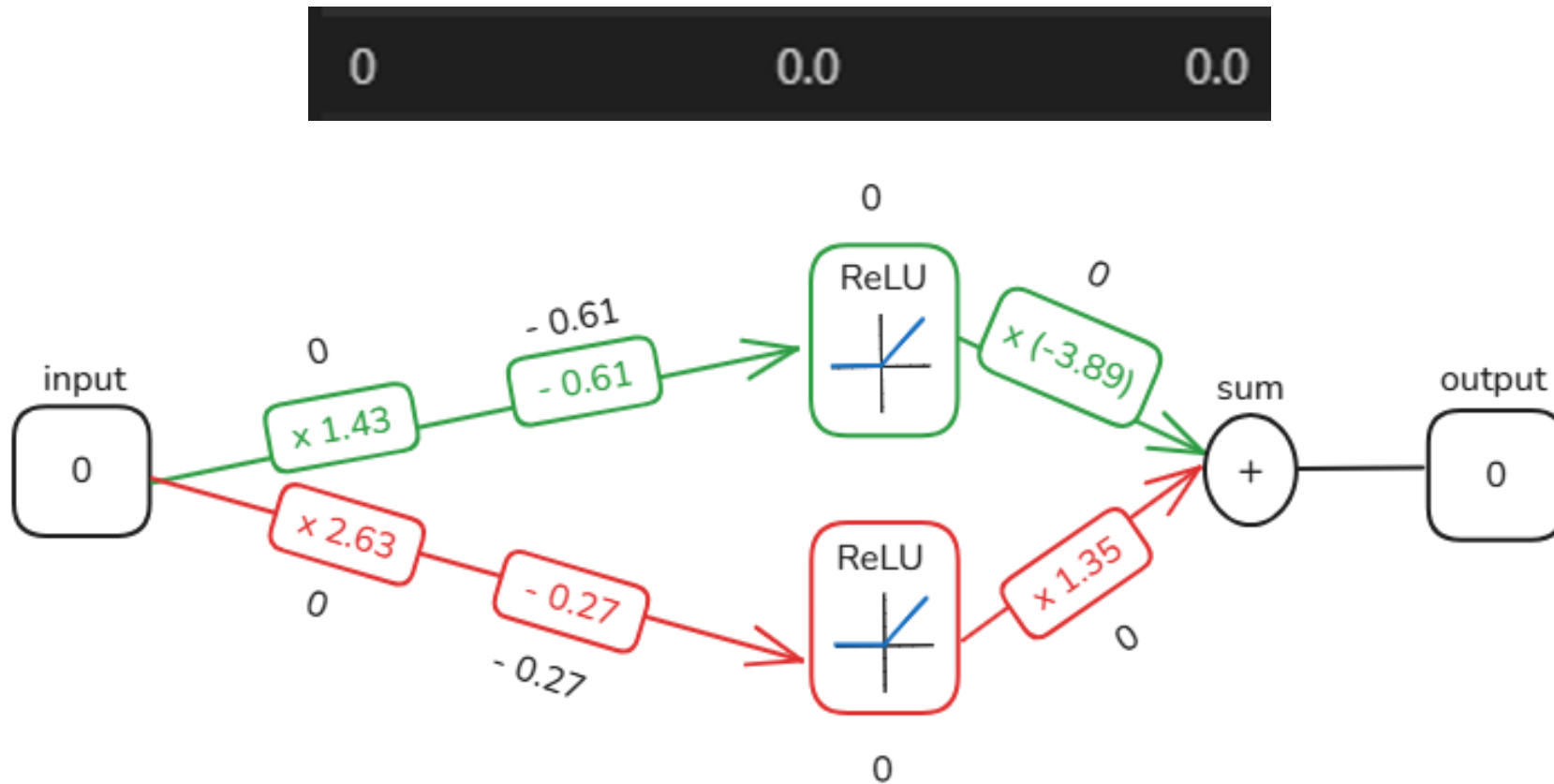
Pass Forward (propagación hacia delante)

	Amount taken (g)	% Effectiveness
0	0.0	0.0
1	0.1	0.0
2	0.2	0.0
3	0.5	1.0
4	0.6	1.0
5	0.7	1.0
6	0.8	0.0
7	0.9	0.0
8	1.0	0.0



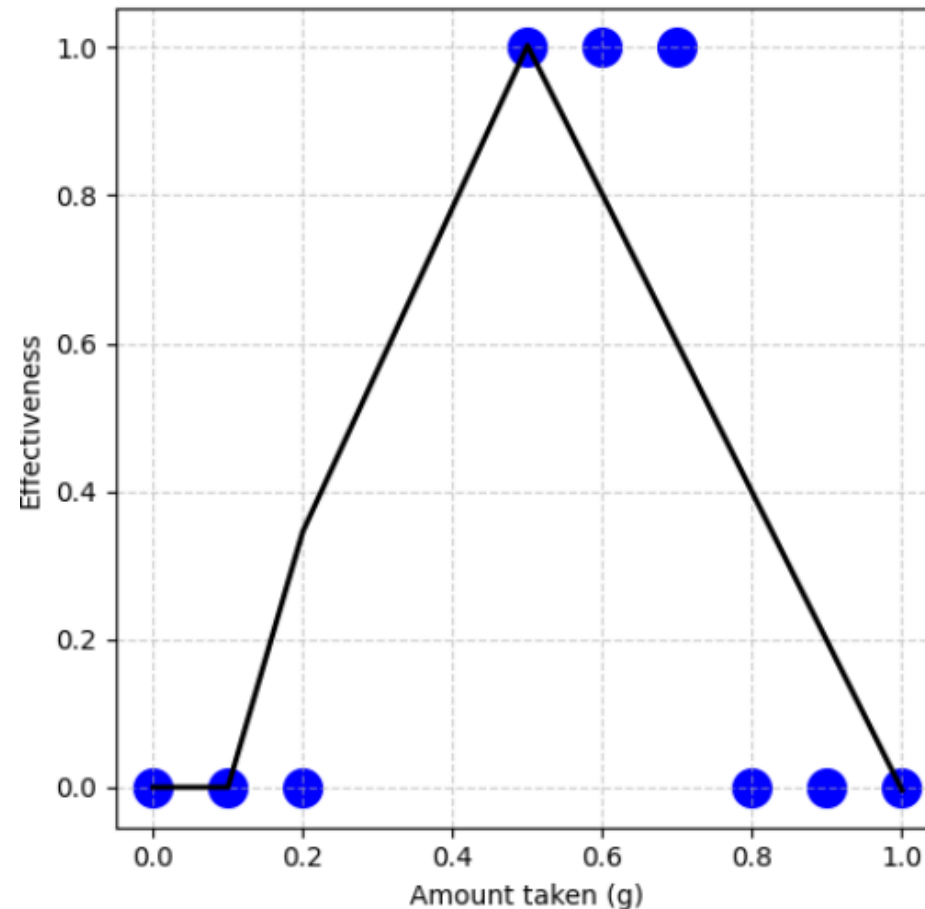
# Conceptos fundamentales

Pass Forward (propagación hacia delante)



# Conceptos fundamentales

Pass Forward (propagación hacia delante)



# Conceptos fundamentales

¡Práctica!

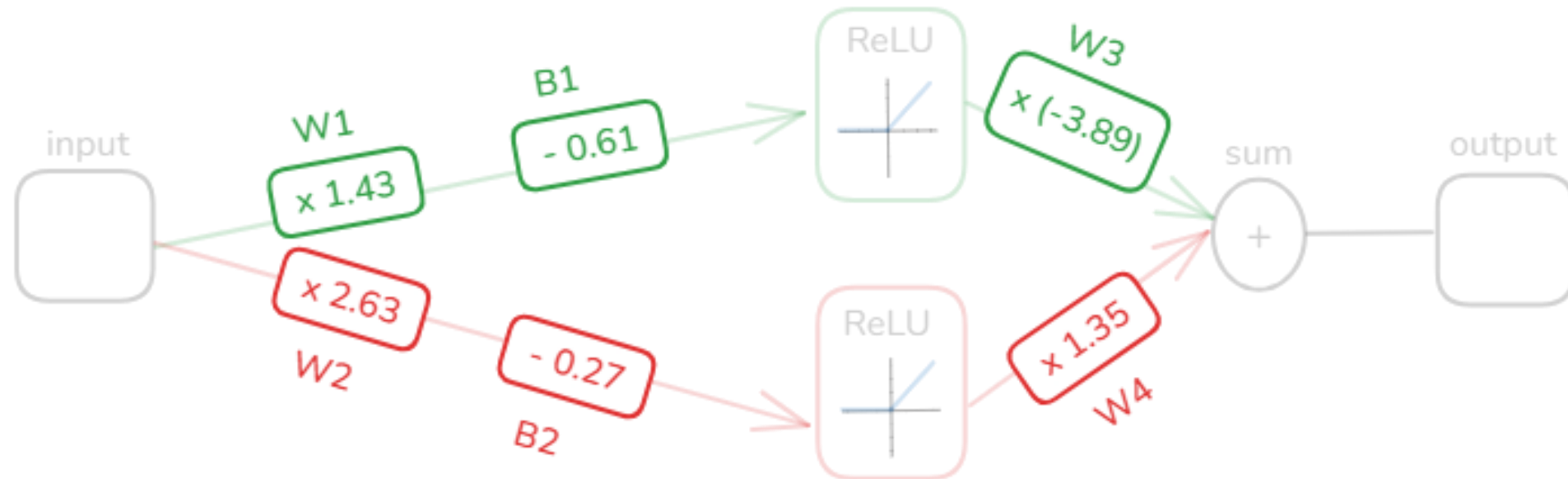




# Optimización único parámetro con backpropagation

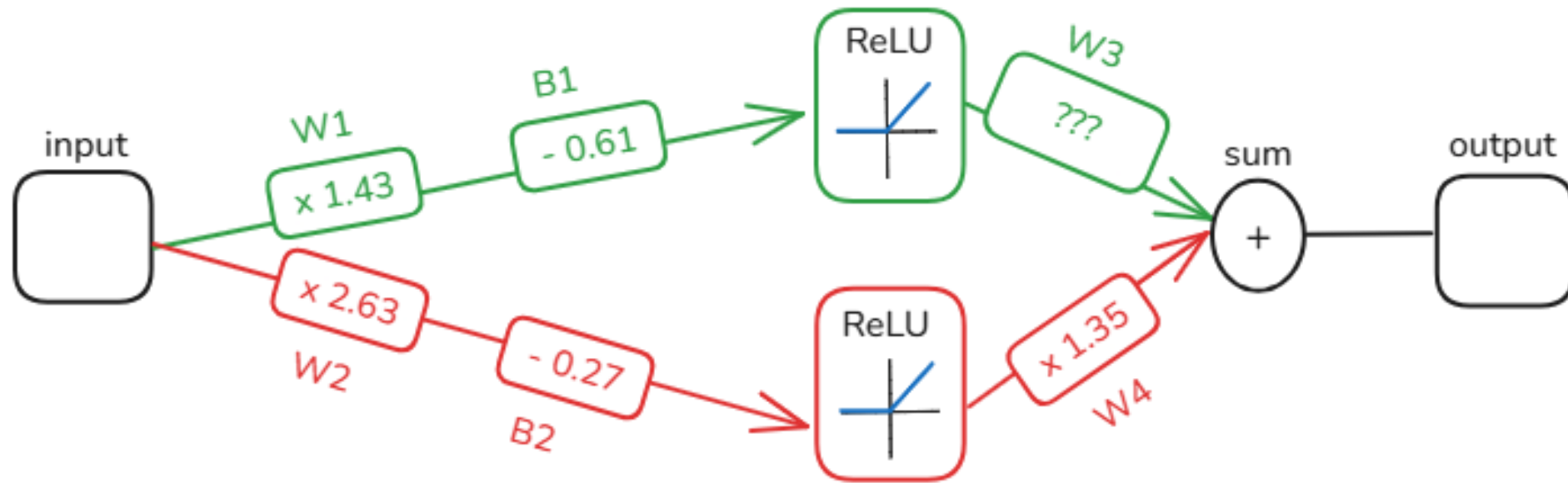
Weights  $W1$ ,  $W2$ ,  $W3$ ,  $W4$

Biases  $B1$ ,  $B2$



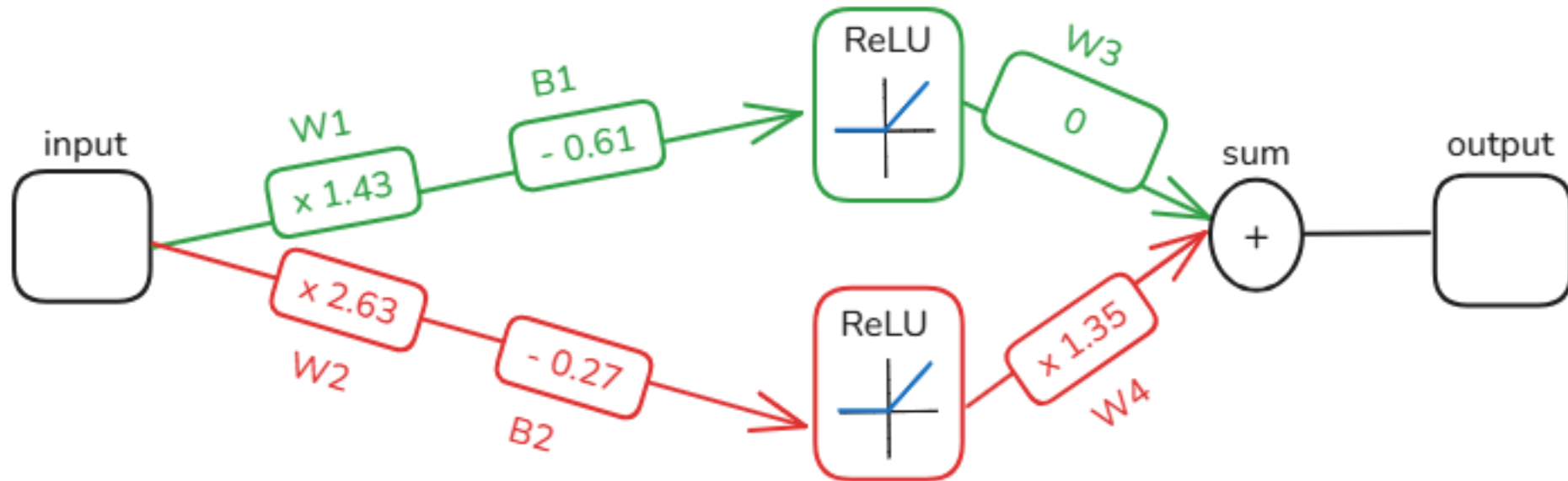
# Optimización único parámetro con backpropagation

W3 única desconocida

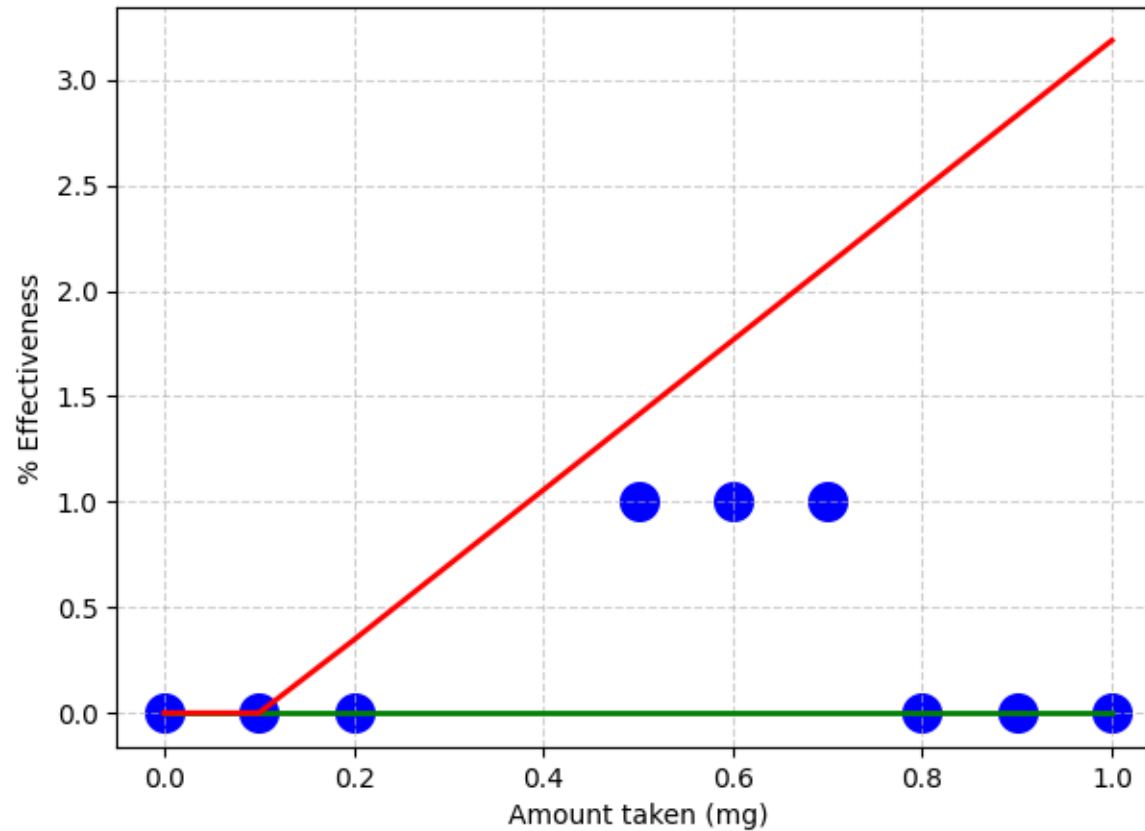


# Optimización único parámetro con backpropagation

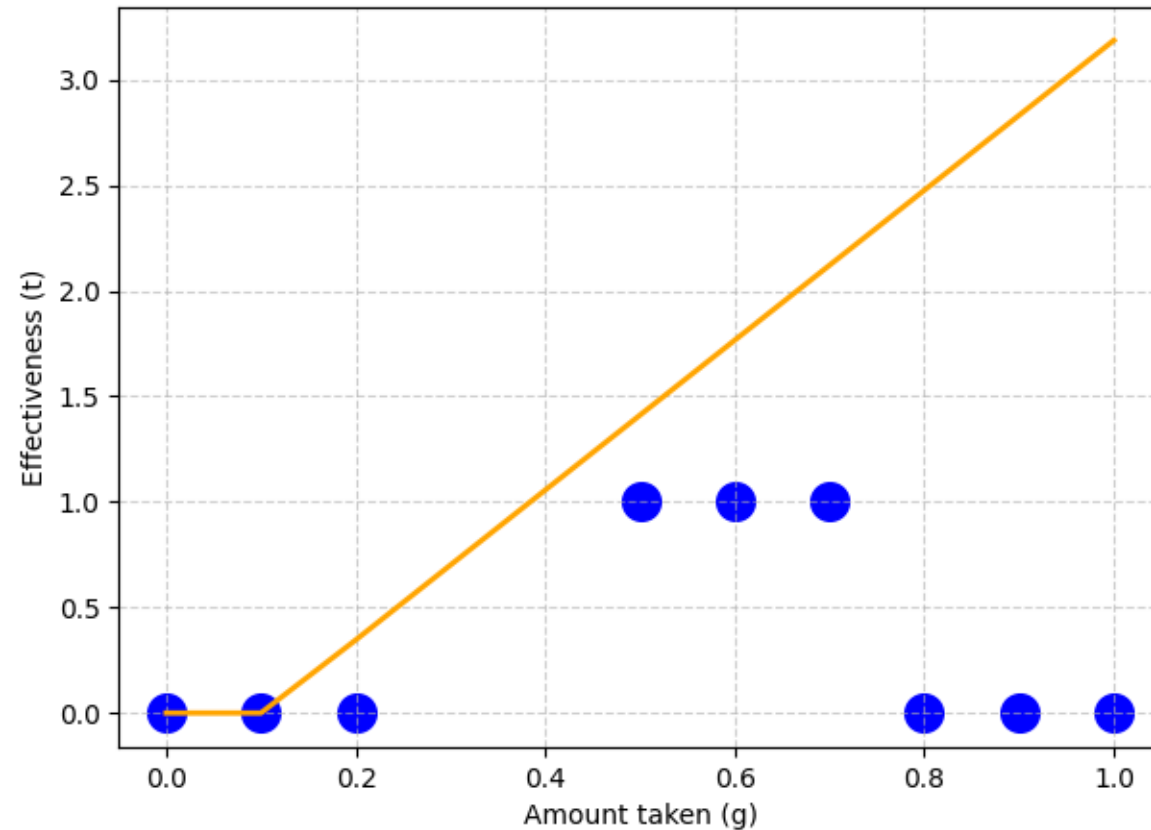
Suponemos  $W3 = 0$



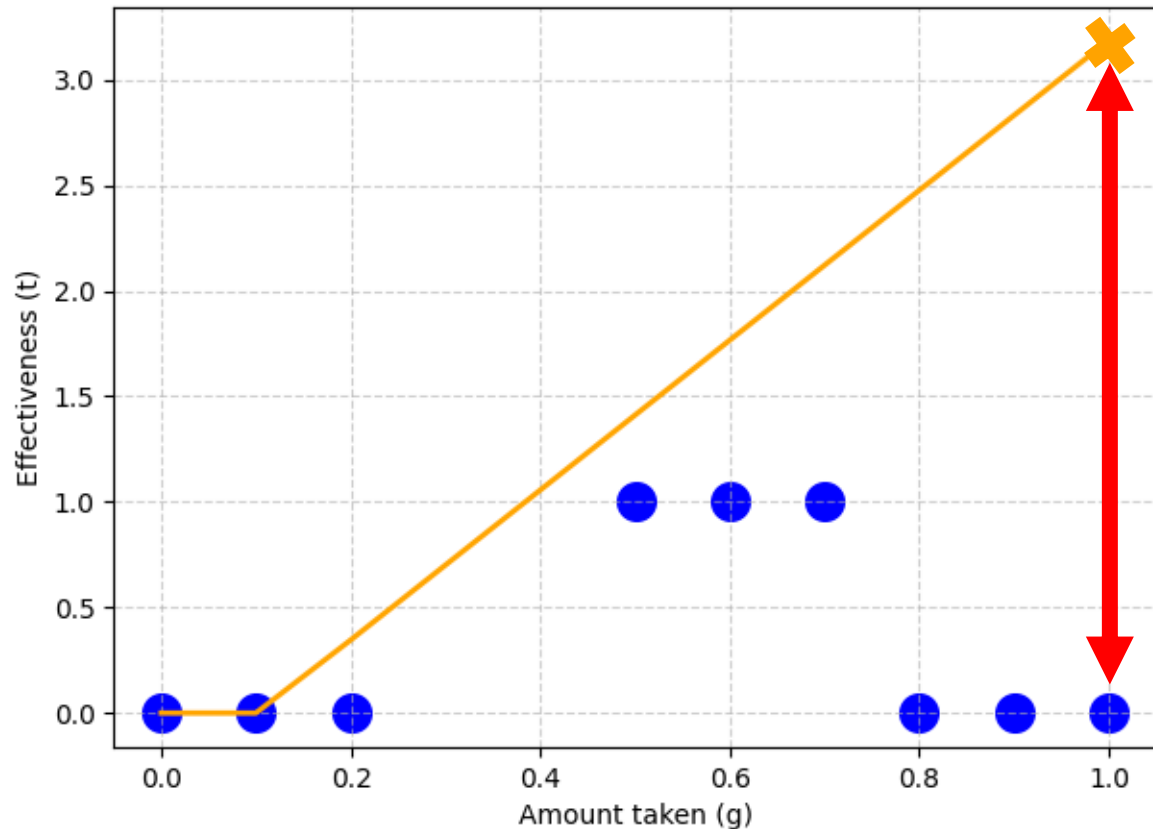
# Optimización único parámetro con backpropagation



# Optimización único parámetro con backpropagation



# Optimización único parámetro con backpropagation

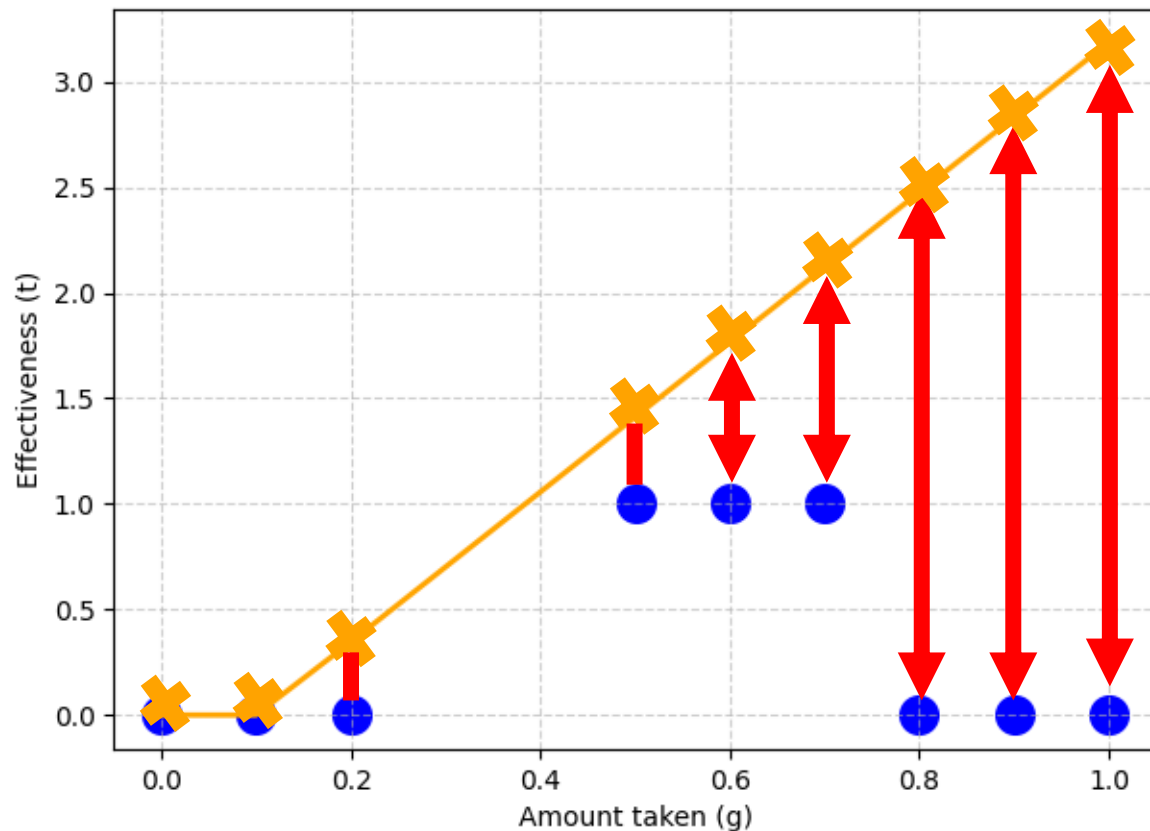


Cuantificar cómo de mala es  
**una única predicción: Residual**

$$\text{Residual} = \text{Observed} - \text{Predicted}$$



# Optimización único parámetro con backpropagation

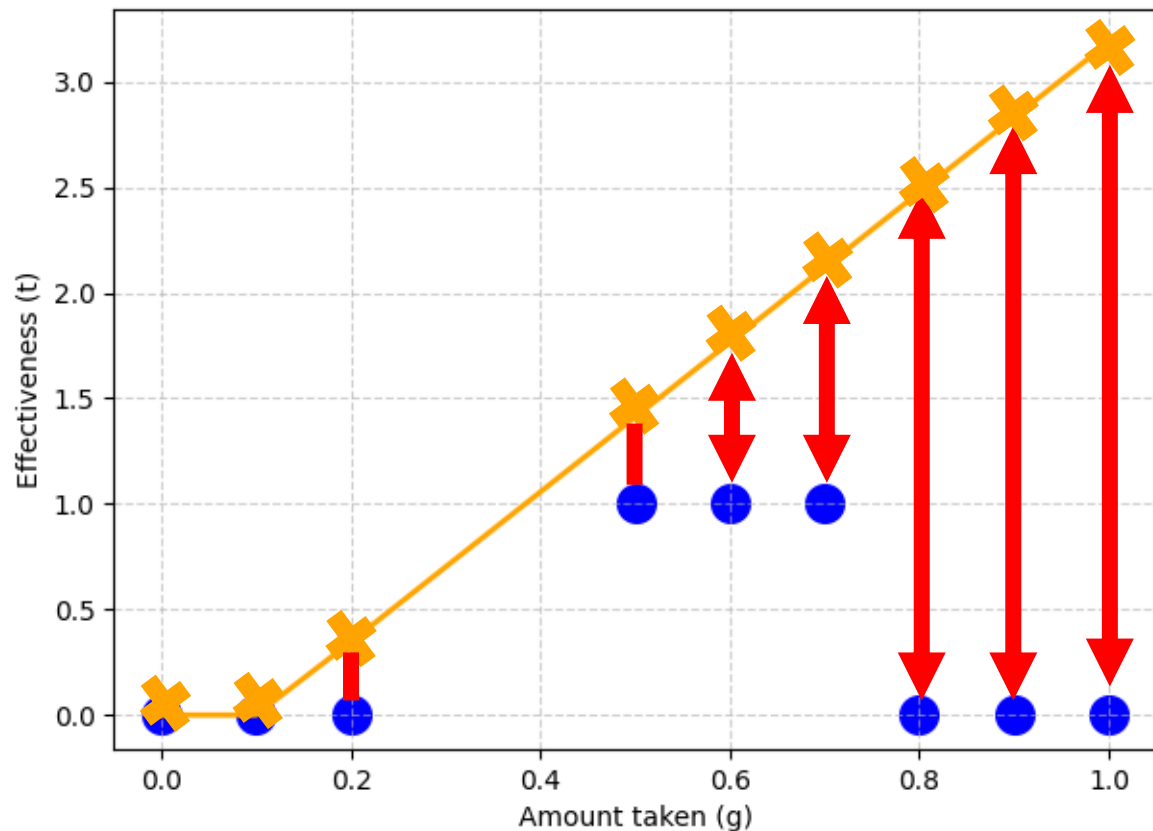


Cuantificar cómo de malo es un modelo (**conjunto de predicciones**): Sum Square of Residuals

$$SSR = \sum_i^n Residual_i^2$$



# Optimización único parámetro con backpropagation



Residuals

```
0  0.00000
1  0.00000
2 -0.34560
3 -0.41075
4 -0.76580
5 -1.12085
6 -2.47590
7 -2.83095
8 -3.18600
```

Residuals^2

```
0  0.000000
1  0.000000
2  0.119439
3  0.168716
4  0.586450
5  1.256305
6  6.130081
7  8.014278
8 10.150596
```





# Optimización único parámetro con backpropagation

If  $w_3 = 0 \rightarrow SSR = \sum_i^n Residual_i^2 = 26.42$

Residuals^2

0	0.000000
1	0.000000
2	0.119439
3	0.168716
4	0.586450
5	1.256305
6	6.130081
7	8.014278
8	10.150596



# Optimización único parámetro con backpropagation

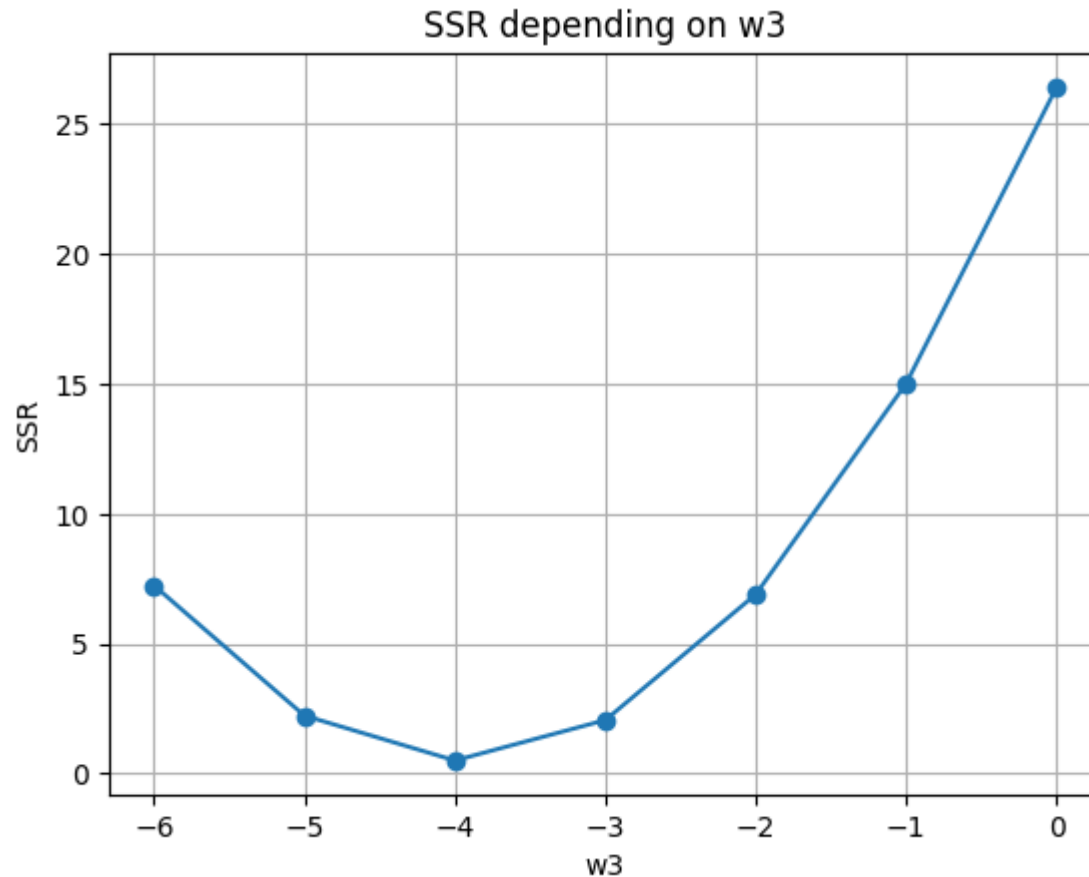
If  $w_3 = 0 \rightarrow SSR = \sum_i^n Residual_i^2 = 26.42$

-6	7.24
-5	2.23
-4	0.5
-3	2.06
-2	6.9
-1	15.02

w3	SSR
-6	7.242445
-5	2.233206
-4	0.506558
-3	2.062499
-2	6.901031
-1	15.022152
0	26.425864



# Optimización único parámetro con backpropagation



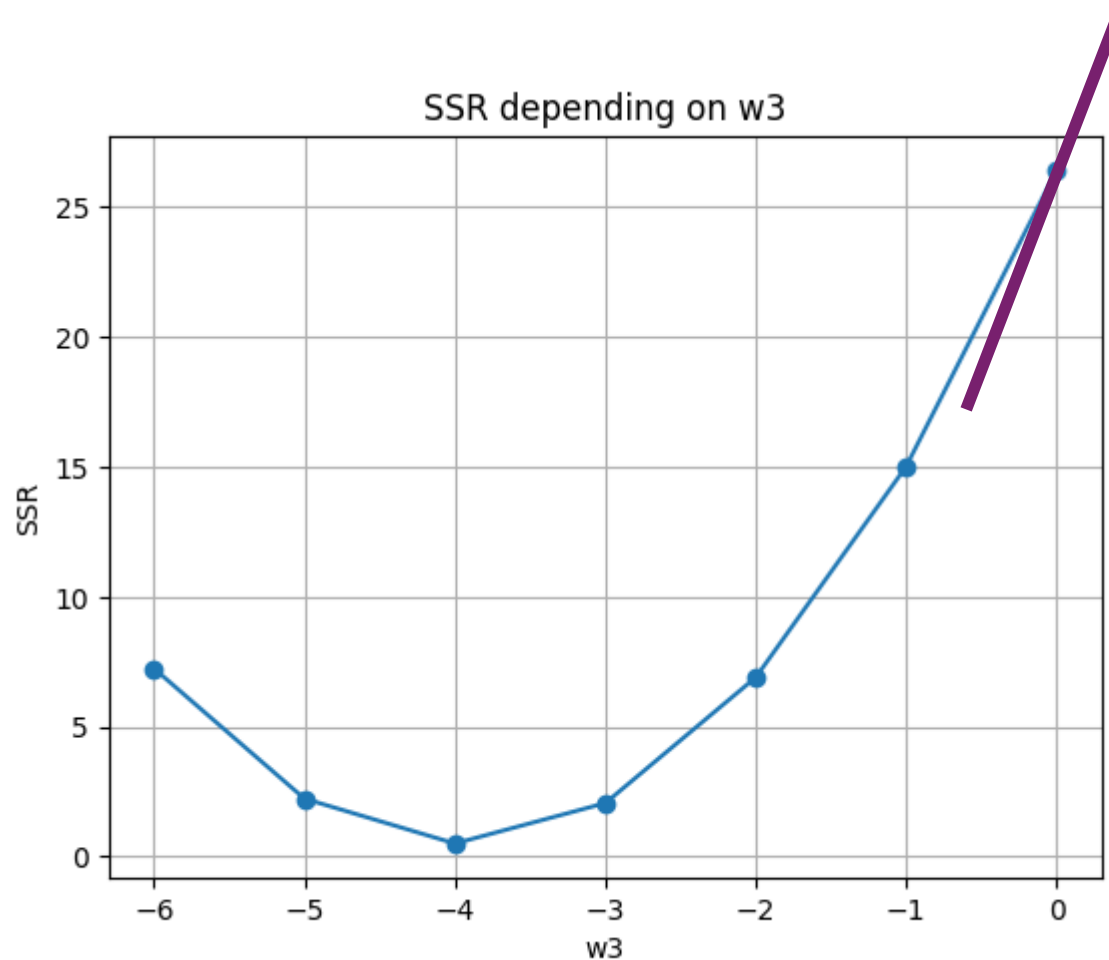
# Optimización único parámetro con backpropagation

Objetivo: optimizar SSR ( $\sum_i^n Residual_i^2$ )

Objetivo: optimizar **Loss Function** (función de pérdida)



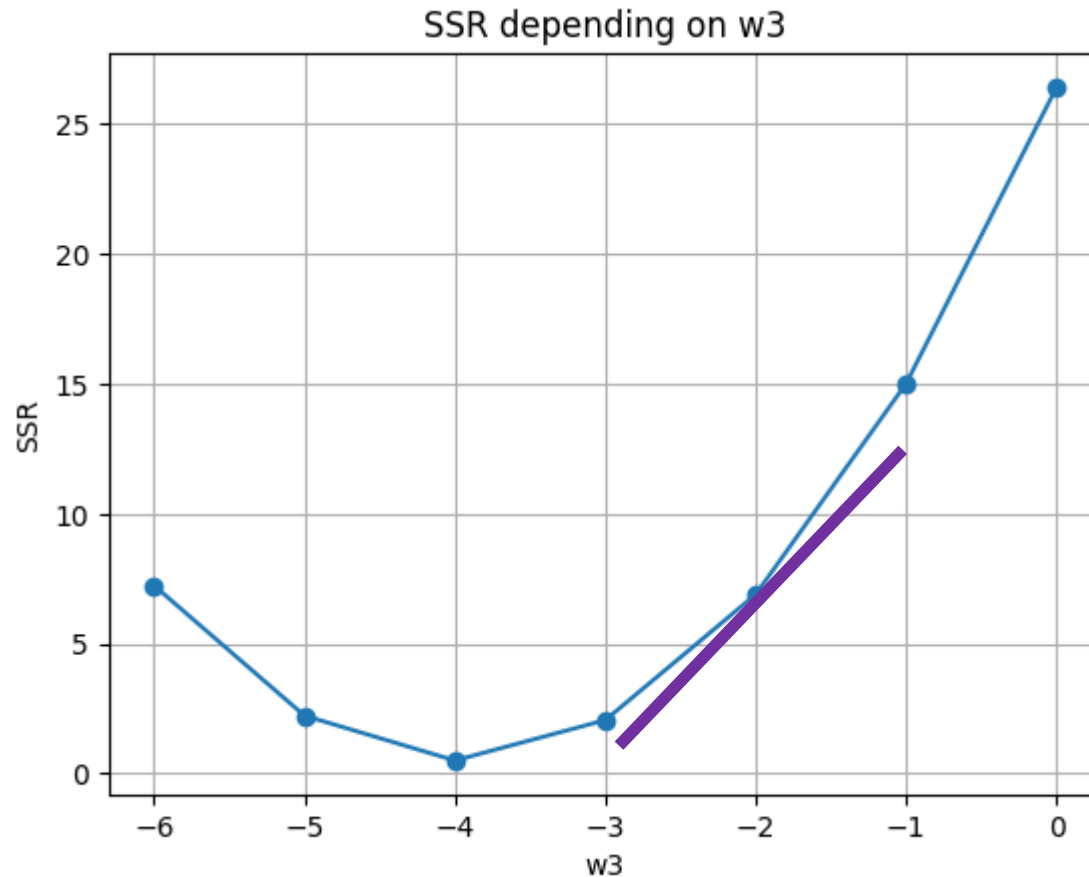
# Optimización único parámetro con backpropagation



$$\frac{\partial SSR}{\partial w_3} > 0$$



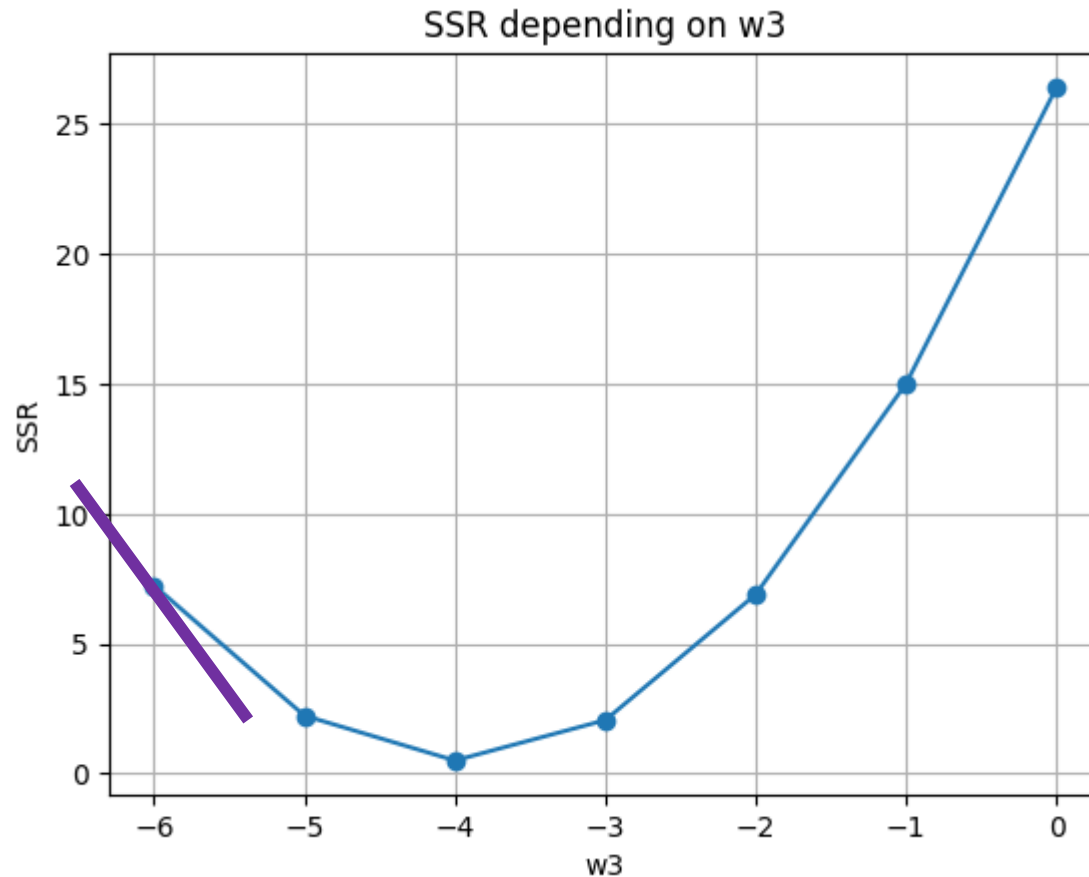
# Optimización único parámetro con backpropagation



$$\frac{\partial SSR}{\partial w3} > 0$$



# Optimización único parámetro con backpropagation



$$\frac{\partial \text{SSR}}{\partial w_3} < 0$$



# Optimización único parámetro con backpropagation

$$\frac{dSSR}{dW3}$$

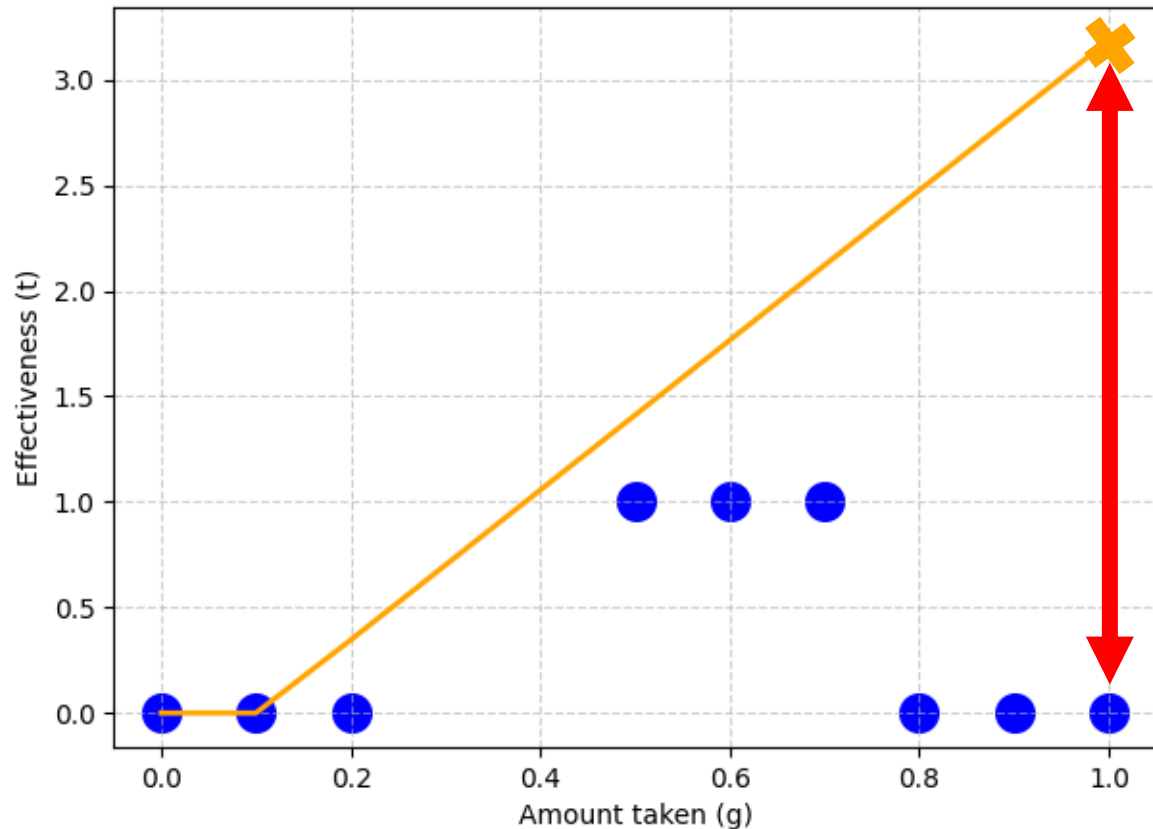
Signo -> en qué dirección moverse

Magnitud -> cómo de lejos estamos





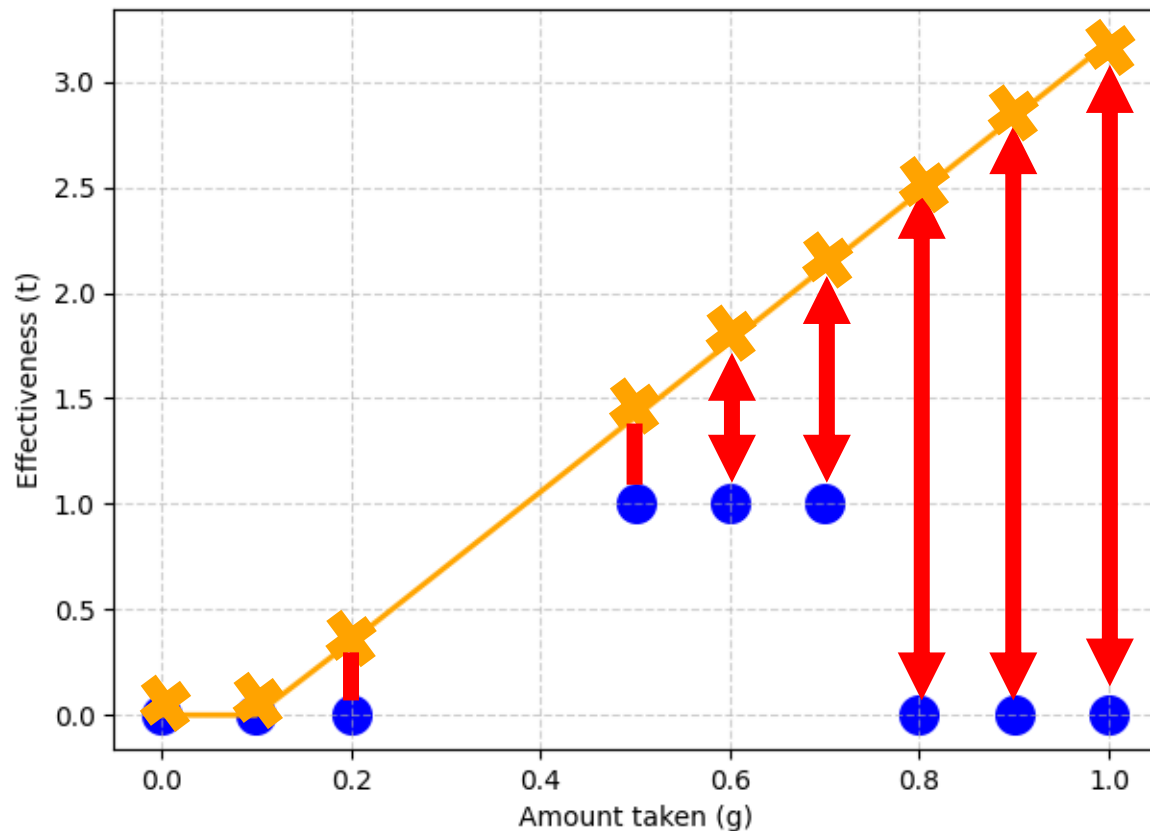
# Optimización único parámetro con backpropagation



$$\text{Residual} = \text{Observed} - \text{Predicted}$$



# Optimización único parámetro con backpropagation



Residual = Observed - Predicted

$$SSR = \sum_{i=0}^n (\text{Residual}_i)^2$$

$$SSR = \sum_{i=0}^n (\text{Observed}_i - \text{Predicted}_i)^2$$



# Optimización único parámetro con backpropagation

$$SSR = \sum_{i=0}^n (\text{Observed}_i - \text{Predicted}_i)^2$$

Observed -> valores tabulados

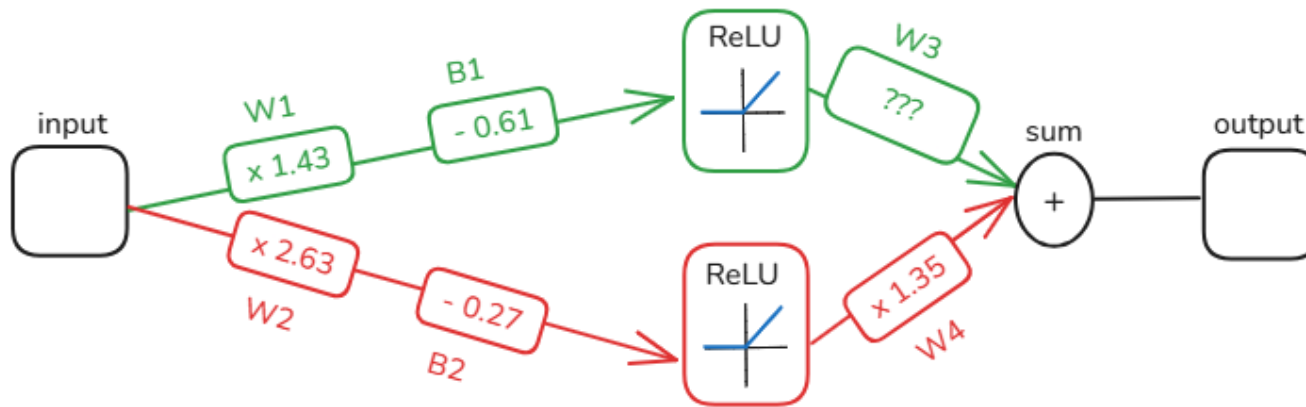
Amount taken (g)
0.0
0.1
0.2
0.5
0.6
0.7
0.8
0.9
1.0



# Optimización único parámetro con backpropagation

$$\text{SSR} = \sum_{i=0}^n (\text{Observed}_i - \text{Predicted}_i)^2$$

$$\text{Predicted} = \text{upper} + \text{lower} = (y \times W3) + \text{lower}$$



# Optimización único parámetro con backpropagation

Chain Rule (Regla de la cadena)[2]

$$\frac{dSSR}{dW3} = \frac{dSSR}{dPredicted} \times \frac{dPredicted}{dW3}$$

[2] More about the Chain Rule: <https://blog.quantinsti.com/understanding-chain-rule/>



# Optimización único parámetro con backpropagation

$$\frac{dSSR}{dW3} = \frac{dSSR}{dPredicted} \times \frac{dPredicted}{dW3}$$

$$\frac{dSSR}{dPredicted} = \frac{\sum_{i=0}^n (\text{Observed}_i - \text{Predicted}_i)^2}{dPredicted} = \sum_{i=0}^n -2(\text{Observed}_i - \text{Predicted}_i)$$



# Optimización único parámetro con backpropagation

$$\frac{dSSR}{dW3} = \frac{dSSR}{dPredicted} \times \frac{dPredicted}{dW3}$$

$$\frac{dPredicted}{dW3} = \frac{(y \times W3) + \text{lower}}{dW3} = y$$



# Optimización único parámetro con backpropagation

$$\frac{dSSR}{dW3} = \frac{dSSR}{dPredicted} \times \frac{dPredicted}{dW3}$$

$$\frac{dSSR}{dW3} = \sum_{i=0}^n -2(\text{Observed}_i - \text{Predicted}_i) * y_i$$





# Optimización único parámetro con backpropagation

$$\frac{dSSR}{dW3} = \sum_{i=0}^n -2(\text{Observed}_i - \text{Predicted}_i) * y_i$$

**Observed**-> valores tabulados  
(Amount taken (g))

**Predicted**-> salida Red Neuronal

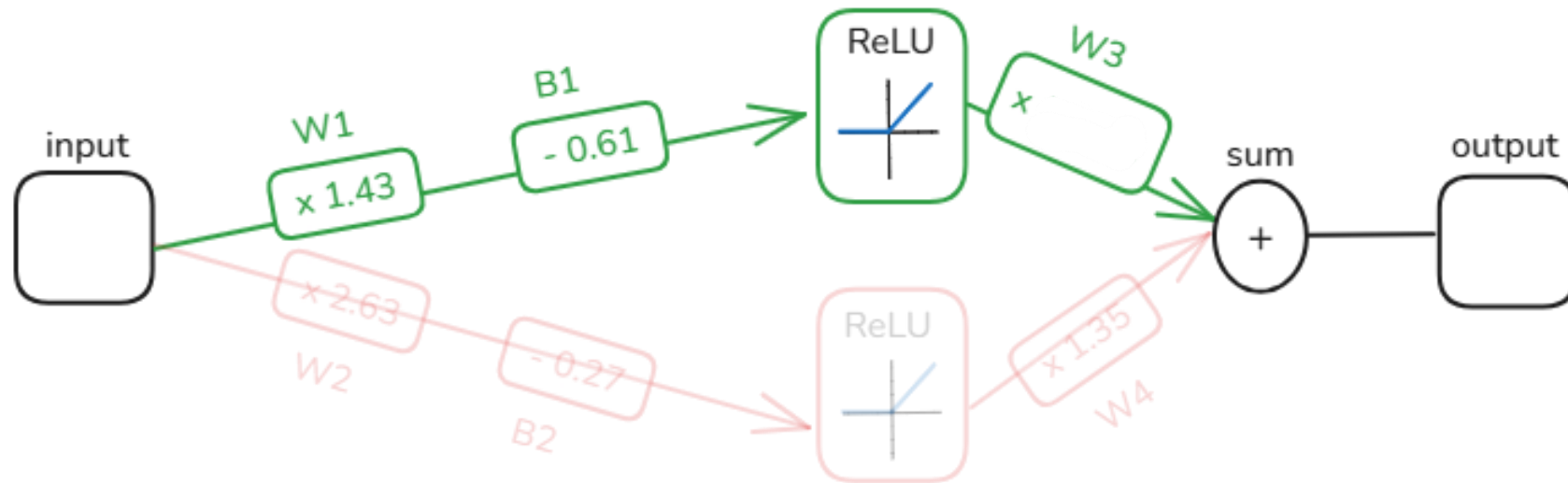
$$y \rightarrow \frac{d\text{Predicted}}{dW3} = \frac{(y \times W3) + \text{lower}}{dW3} = y$$

Amount taken (g)	Predicted	y
0.0	0.00000	0.000
0.1	0.00000	0.000
0.2	0.34560	0.000
0.5	1.41075	0.105
0.6	1.76580	0.248
0.7	2.12085	0.391
0.8	2.47590	0.534
0.9	2.83095	0.677
1.0	3.18600	0.820



# Optimización único parámetro con backpropagation

$$y \rightarrow \frac{d\text{Predicted}}{dW_3} = \frac{(y \times W_3) + \text{lower}}{dW_3} = y$$



# Optimización único parámetro con backpropagation

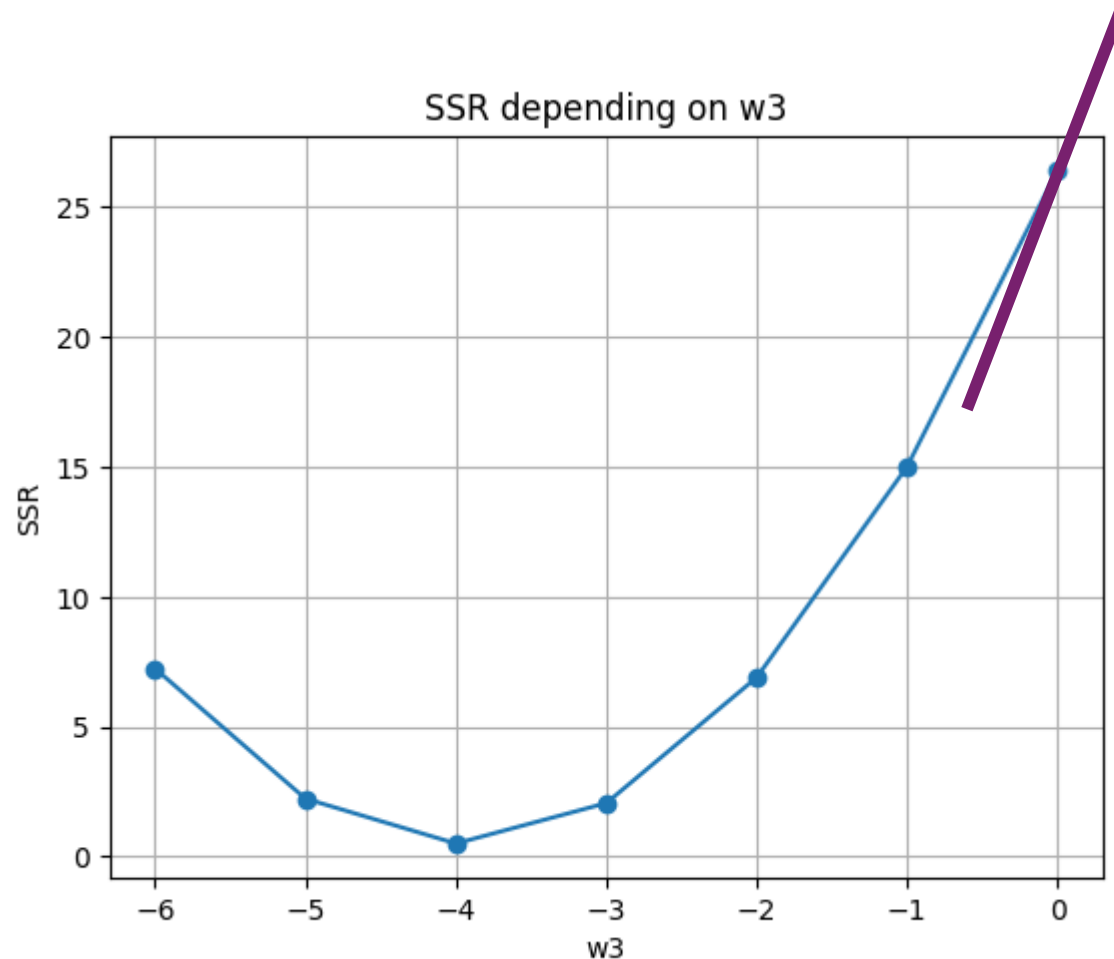
$$\frac{dSSR}{dW3} = \sum_{i=0}^n -2(\text{Observed}_i - \text{Predicted}_i) * y_i$$

Amount taken (g)	Predicted	y	Result
0.0	0.00000	0.000	-0.000000
0.1	0.00000	0.000	-0.000000
0.2	0.34560	0.000	0.000000
0.5	1.41075	0.105	0.191257
0.6	1.76580	0.248	0.578237
0.7	2.12085	0.391	1.111105
0.8	2.47590	0.534	1.789861
0.9	2.83095	0.677	2.614506
1.0	3.18600	0.820	3.585040

Si sumamos todo Result = 9.87



# Optimización único parámetro con backpropagation

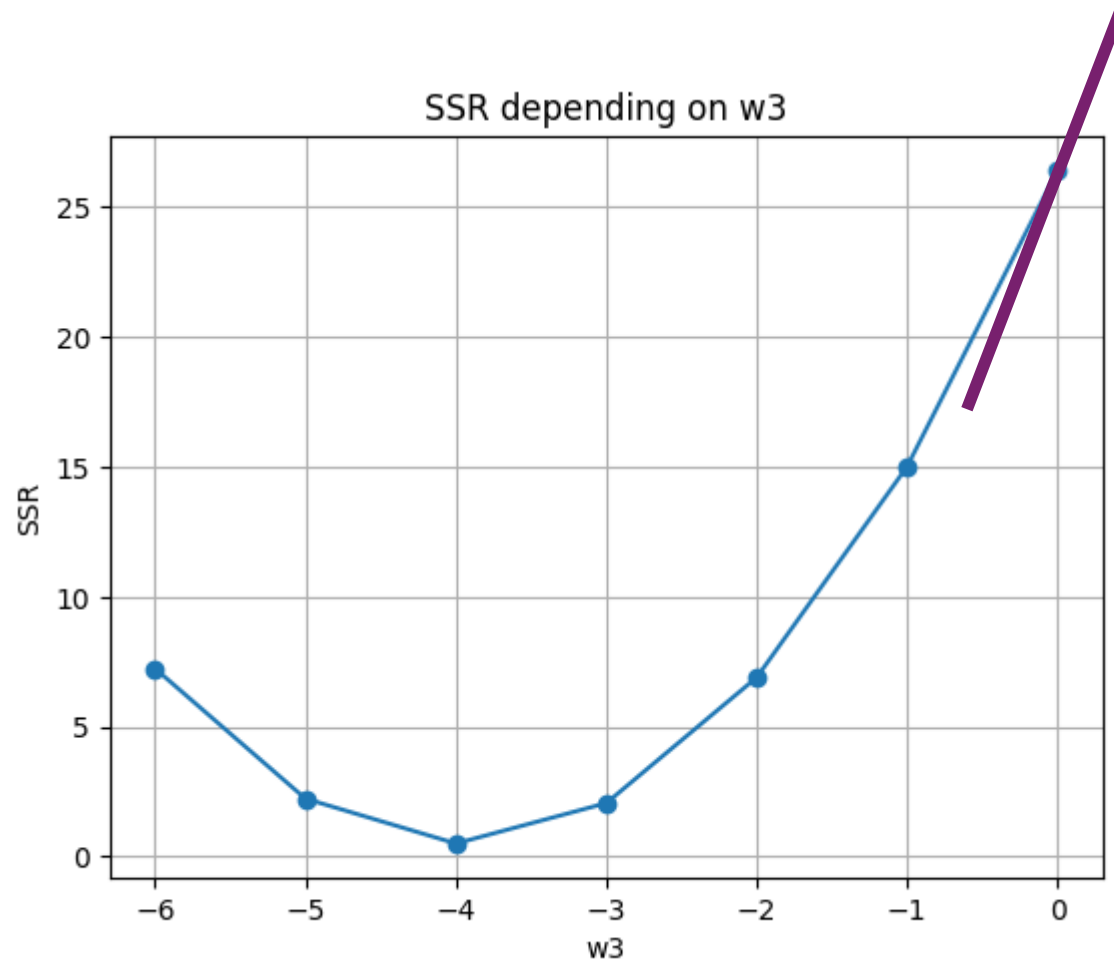


$$\frac{\partial SSR}{\partial w3} > 0$$

Pendiente = **9.87**



# Optimización único parámetro con backpropagation

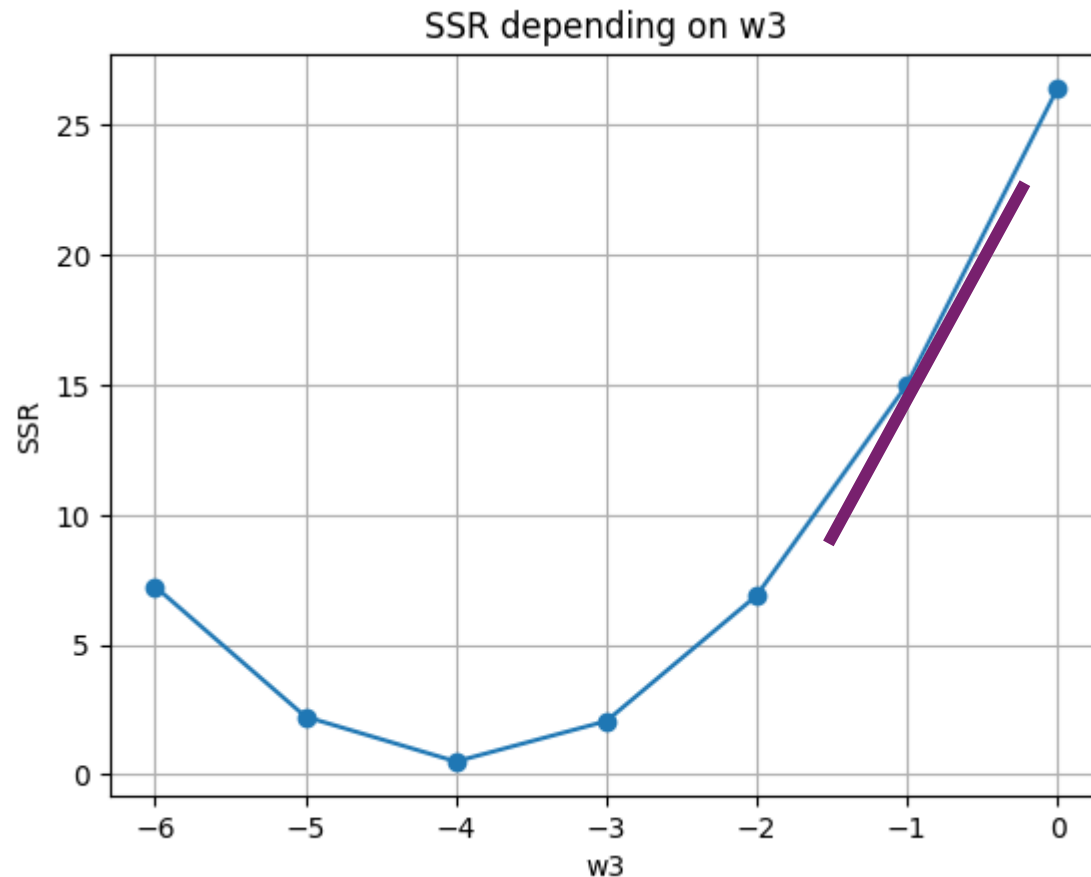


Moverse hacia la izquierda

**¿Cuánto?**



# Optimización único parámetro con backpropagation

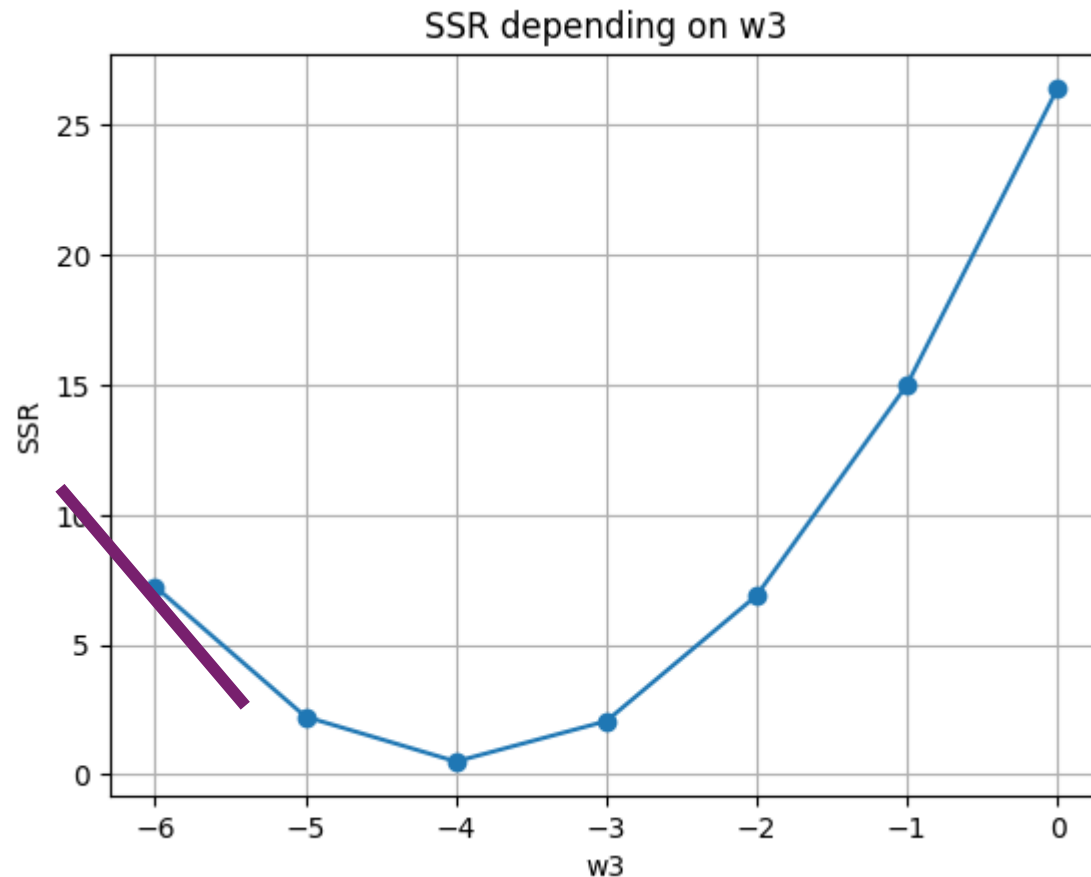


Moverse hacia la izquierda

**¿Cuánto?**



# Optimización único parámetro con backpropagation



Moverse hacia la izquierda

**¿Cuánto?**



# Optimización único parámetro con backpropagation

**Learning Rate** (Ratio de aprendizaje)

**Step Size = Derivative x Learning Rate**





# Optimización único parámetro con backpropagation

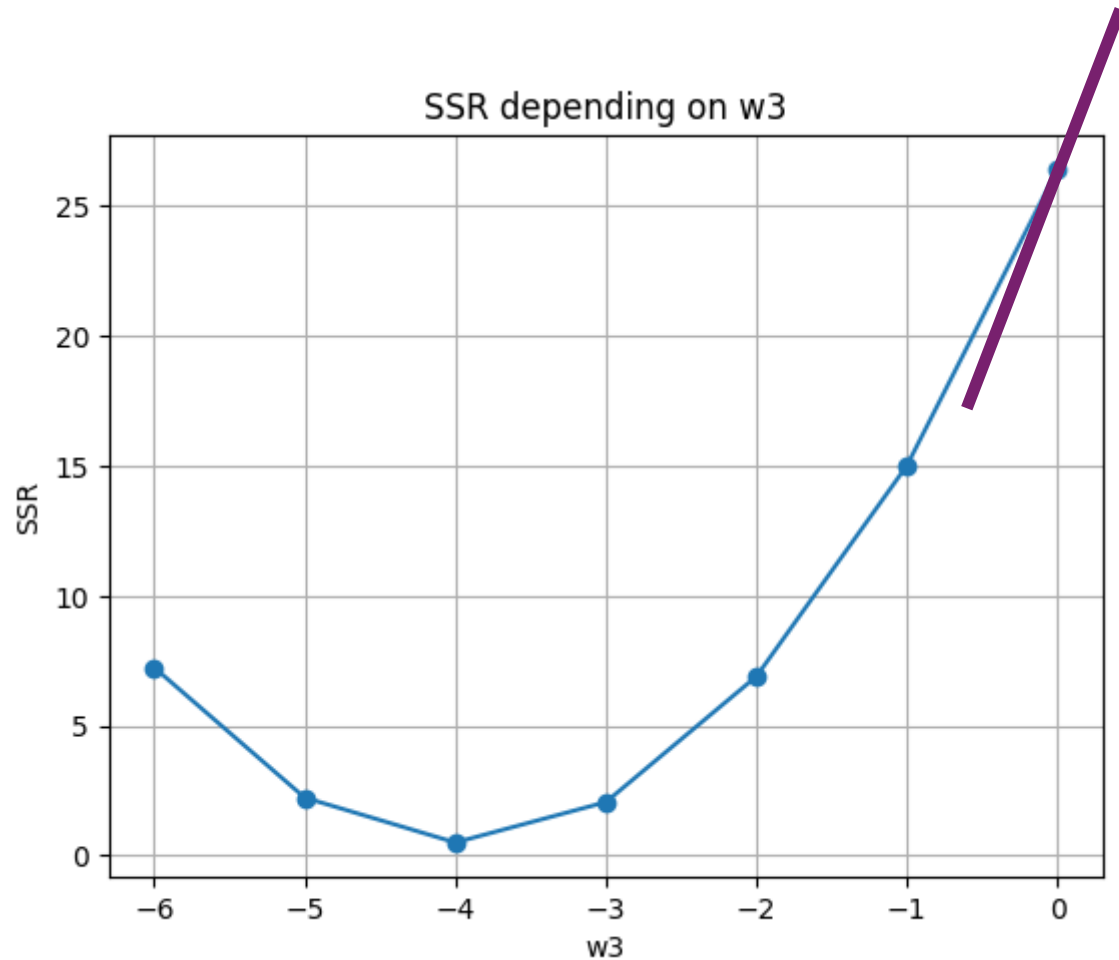
**Learning Rate** (Ratio de aprendizaje)

Step Size = Derivative x Learning Rate

New  $W_3$  = Old  $W_3$  - Step Size



# Optimización único parámetro con backpropagation



Pendiente = 9.87

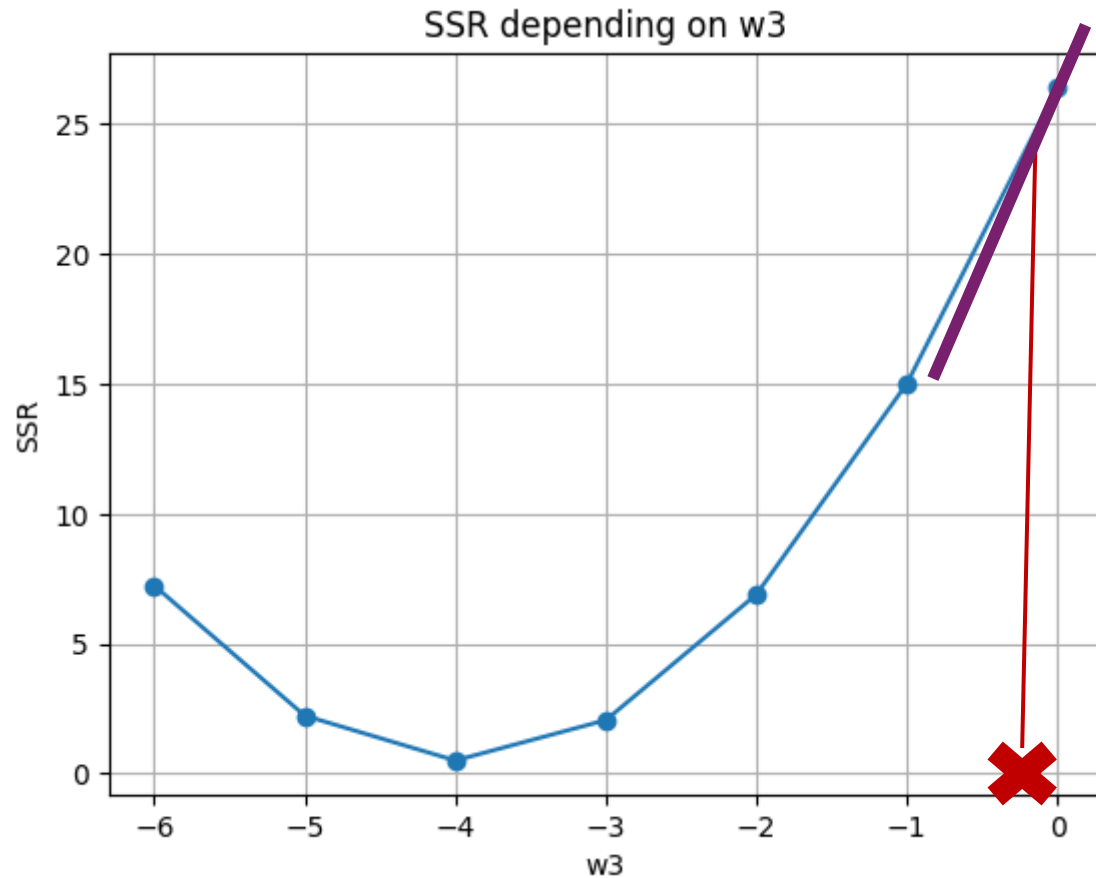
Learning Rate = 0.01

Step Size =  $9.87 \times 0.01 = 0.0987$

New W3 =  $0 - 0.0987 = -0.0987$



# Optimización único parámetro con backpropagation

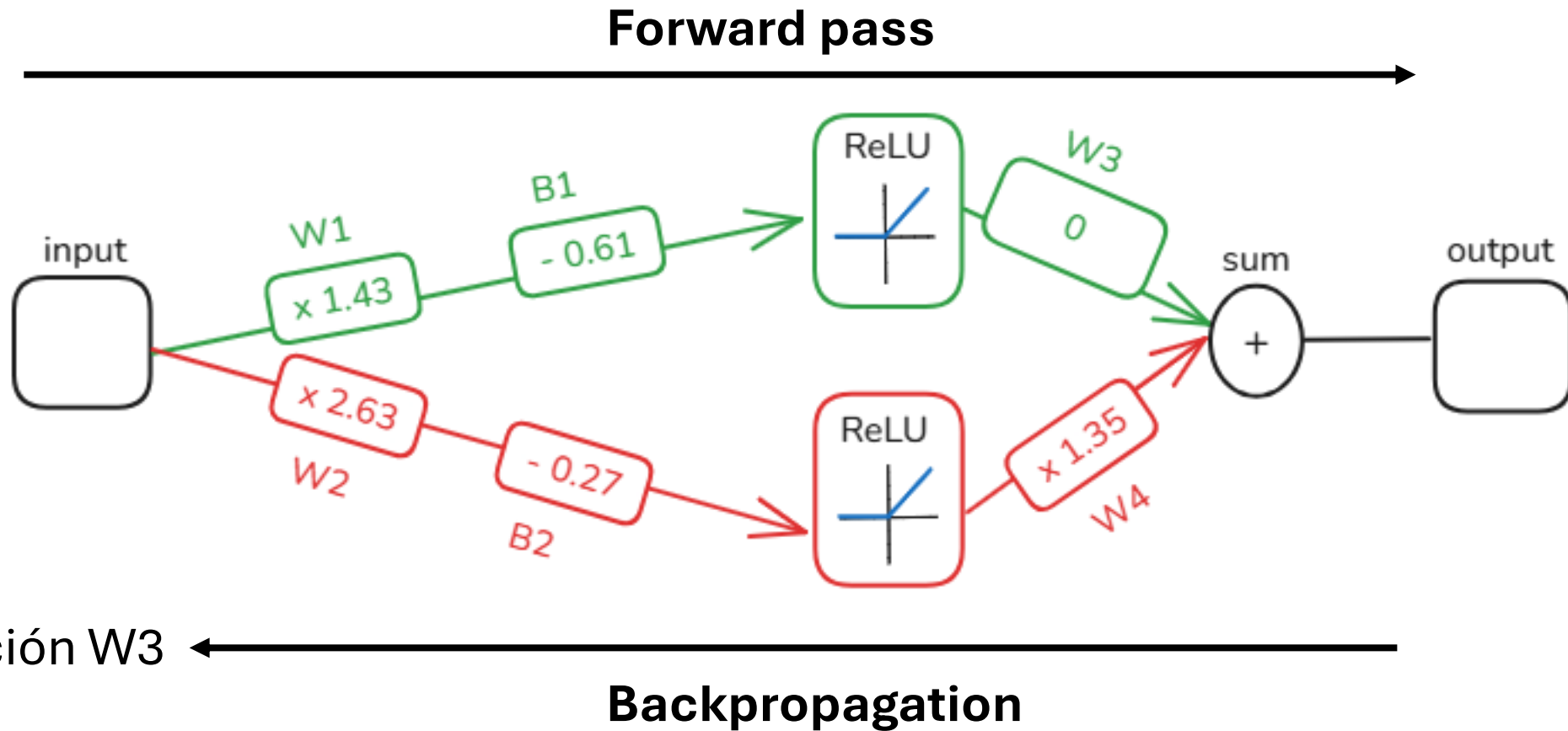


New  $W3 = 0 - 0.0987 = -0.0987$   
 $W3 = \text{New } W3$

**Repetir!**



# Optimización único parámetro con backpropagation

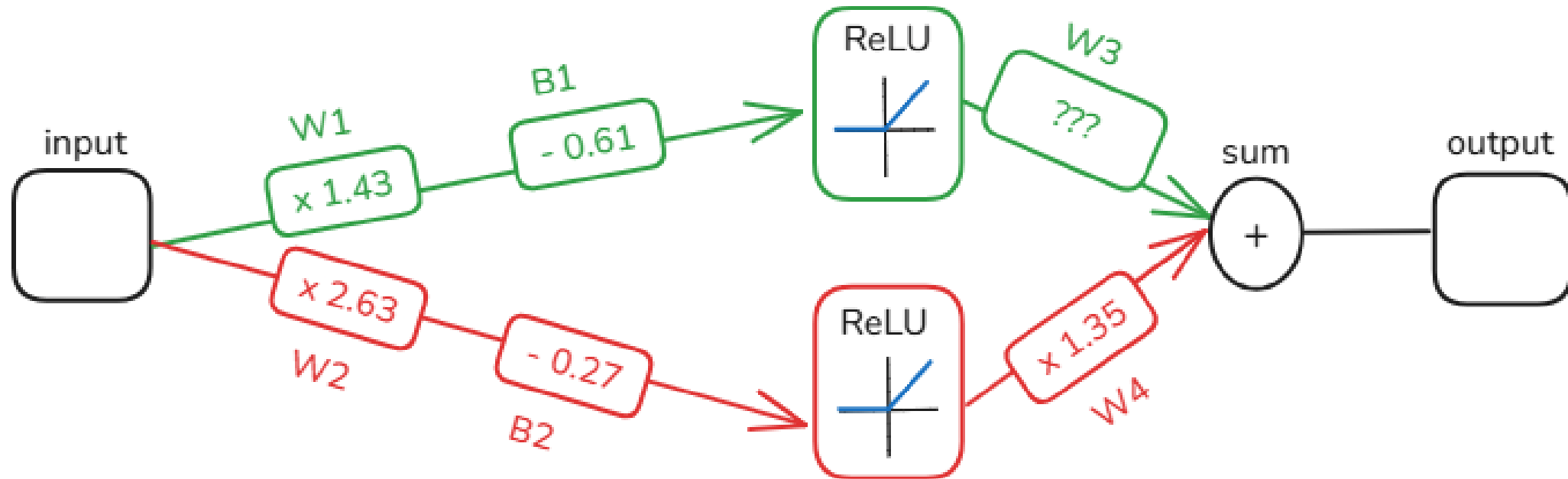


# Optimización único parámetro con backpropagation

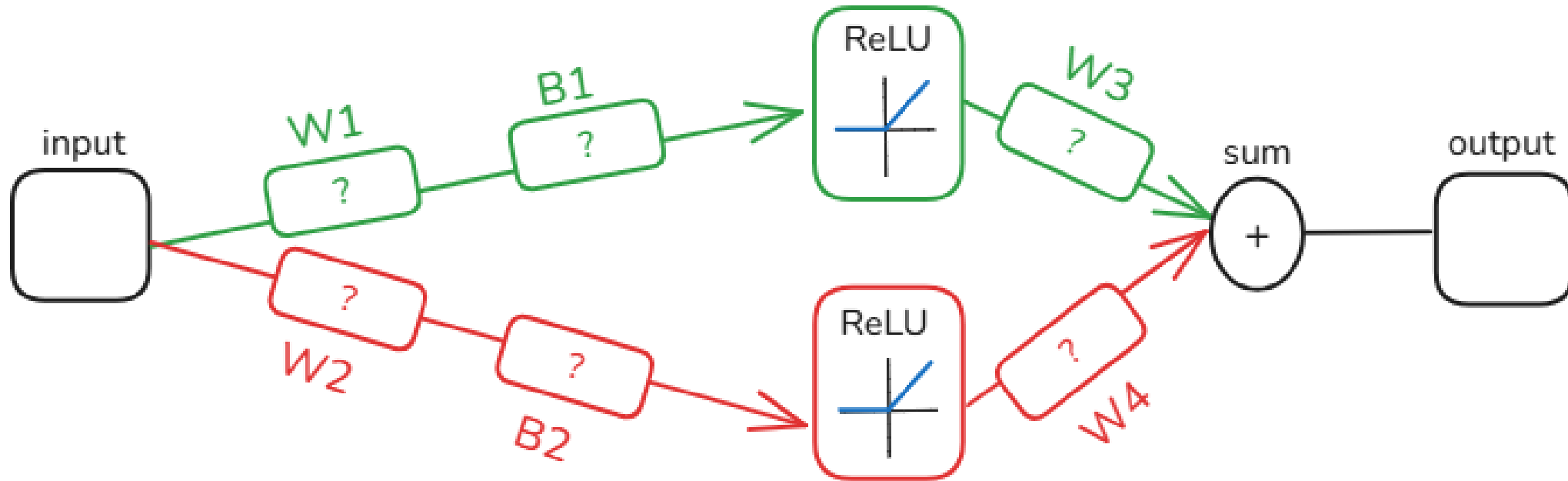
¡Práctica!



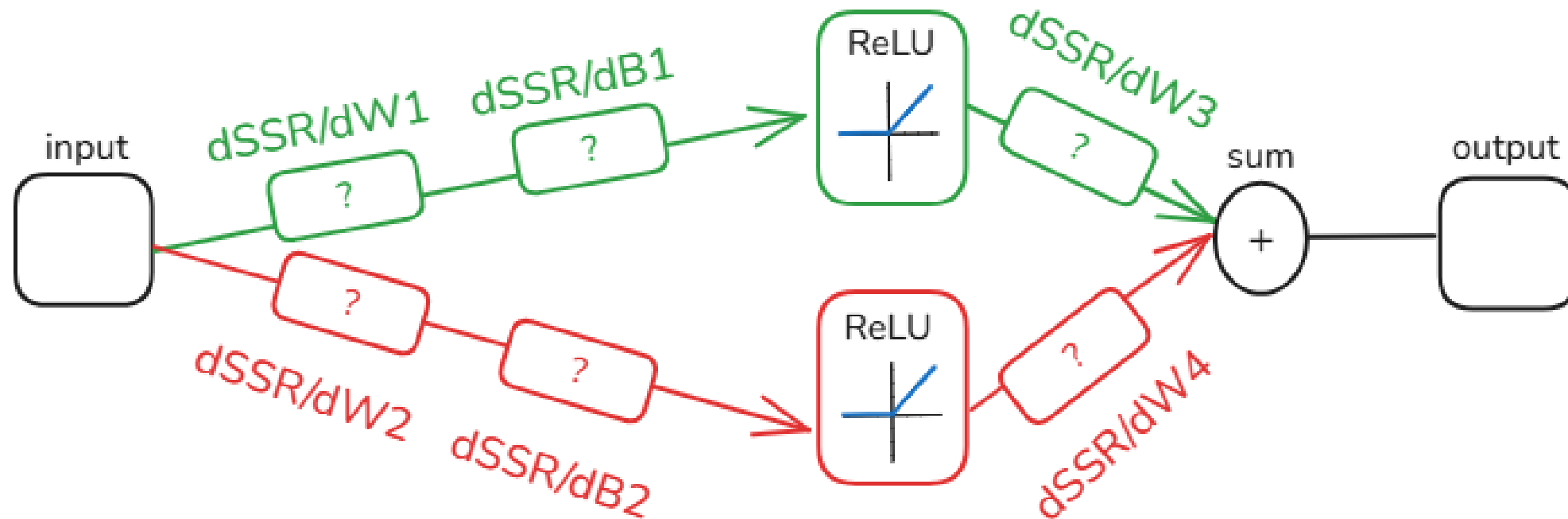
# Optimización múltiples parámetros con backpropagation



# Optimización múltiples parámetros con backpropagation

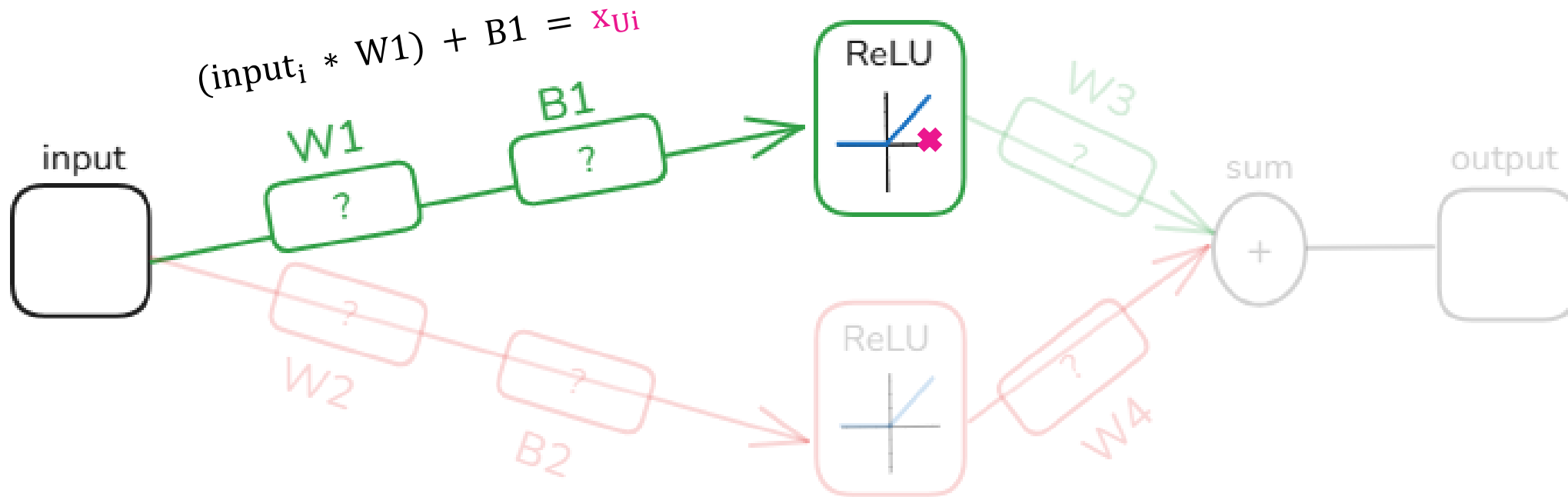


# Optimización múltiples parámetros con backpropagation

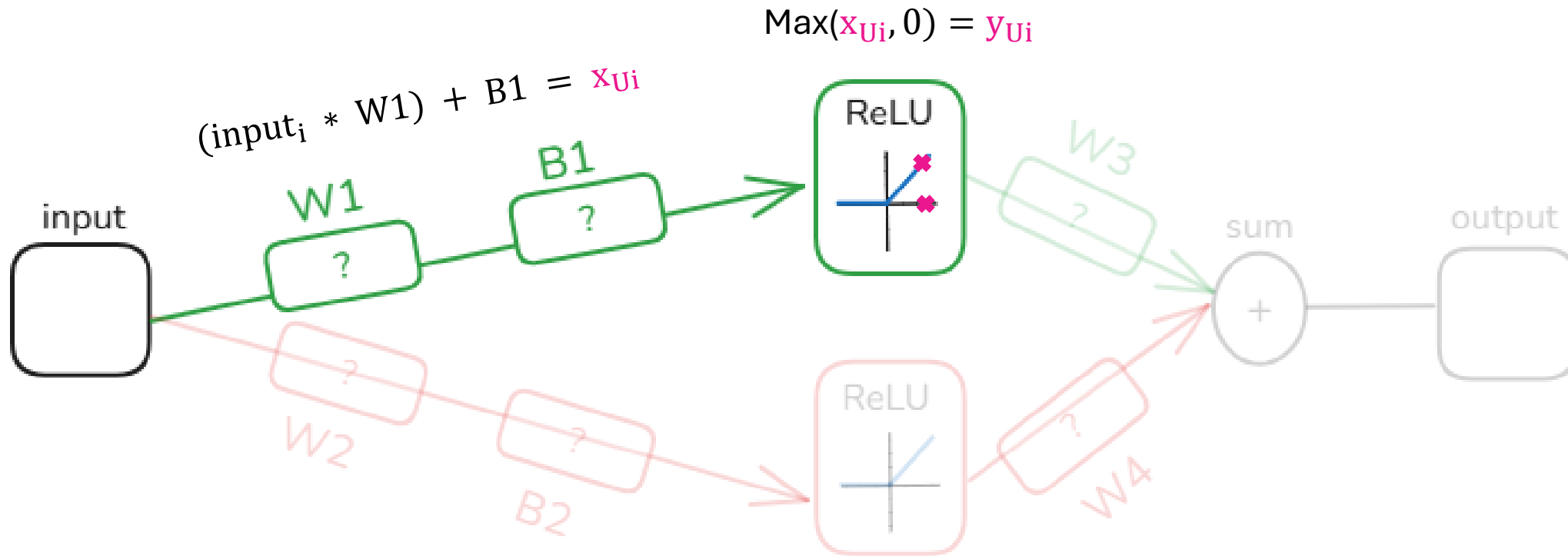




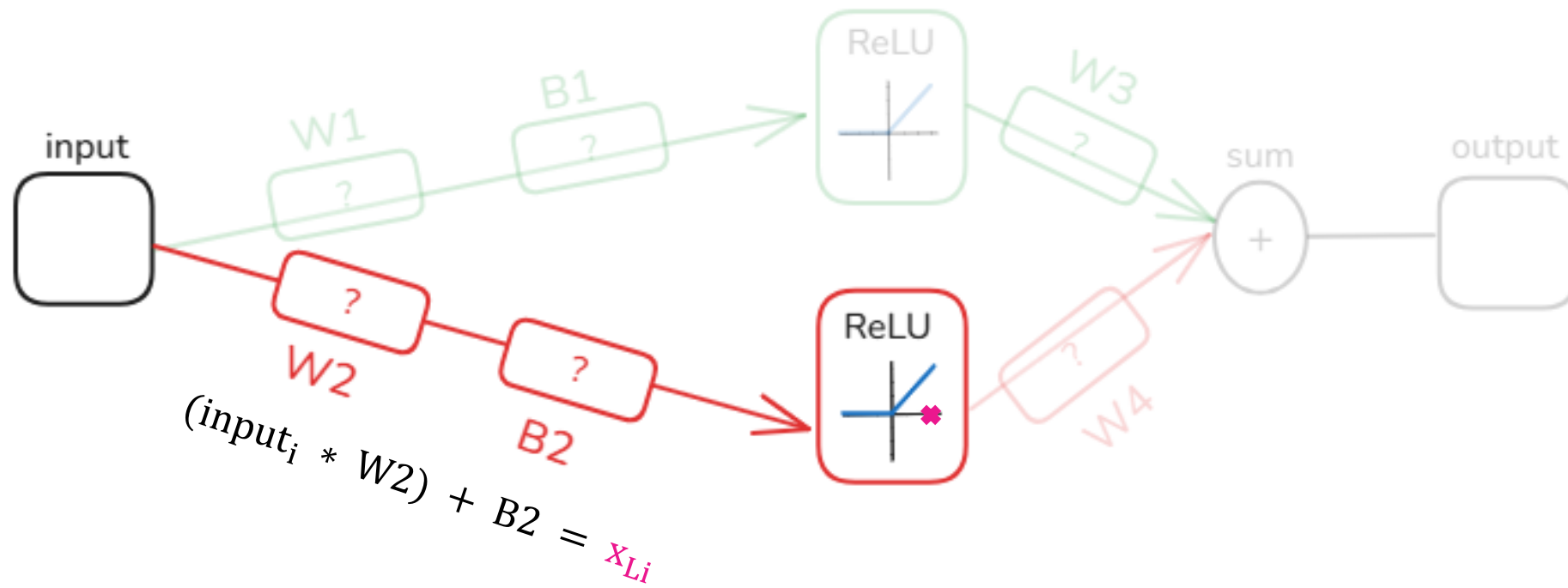
# Optimización múltiples parámetros con backpropagation



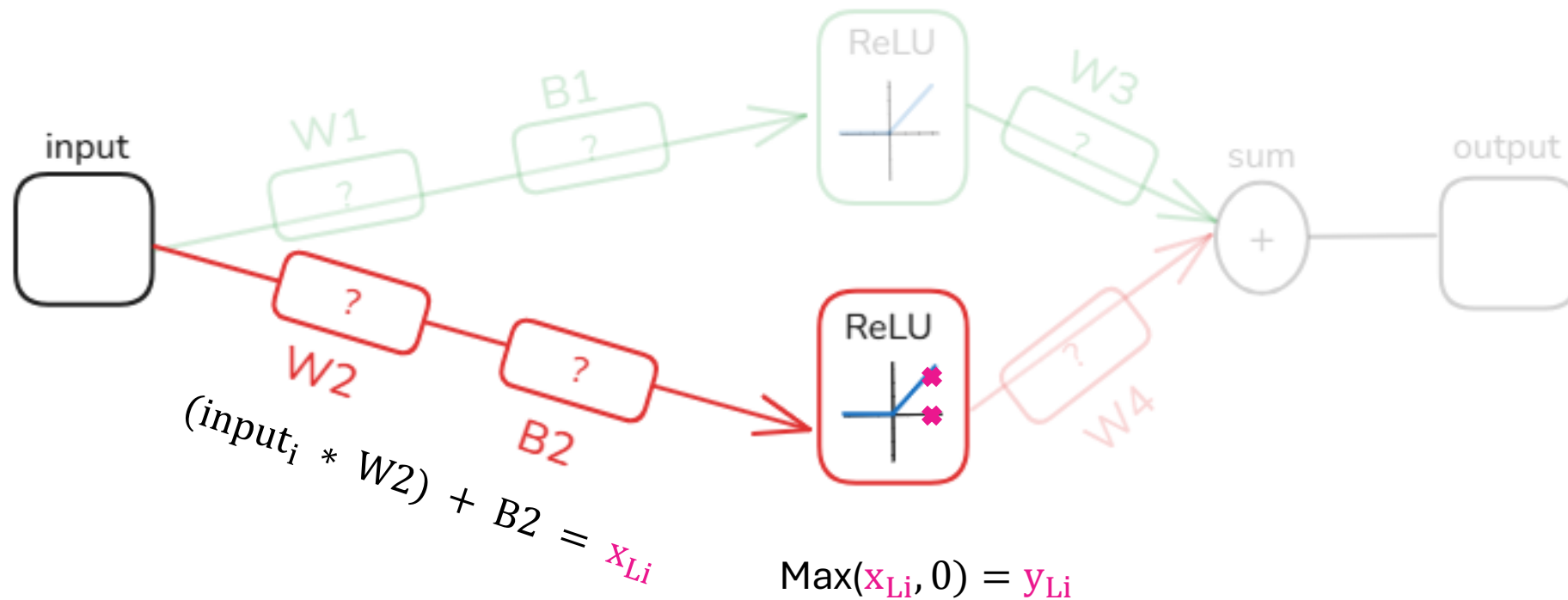
# Optimización múltiples parámetros con backpropagation



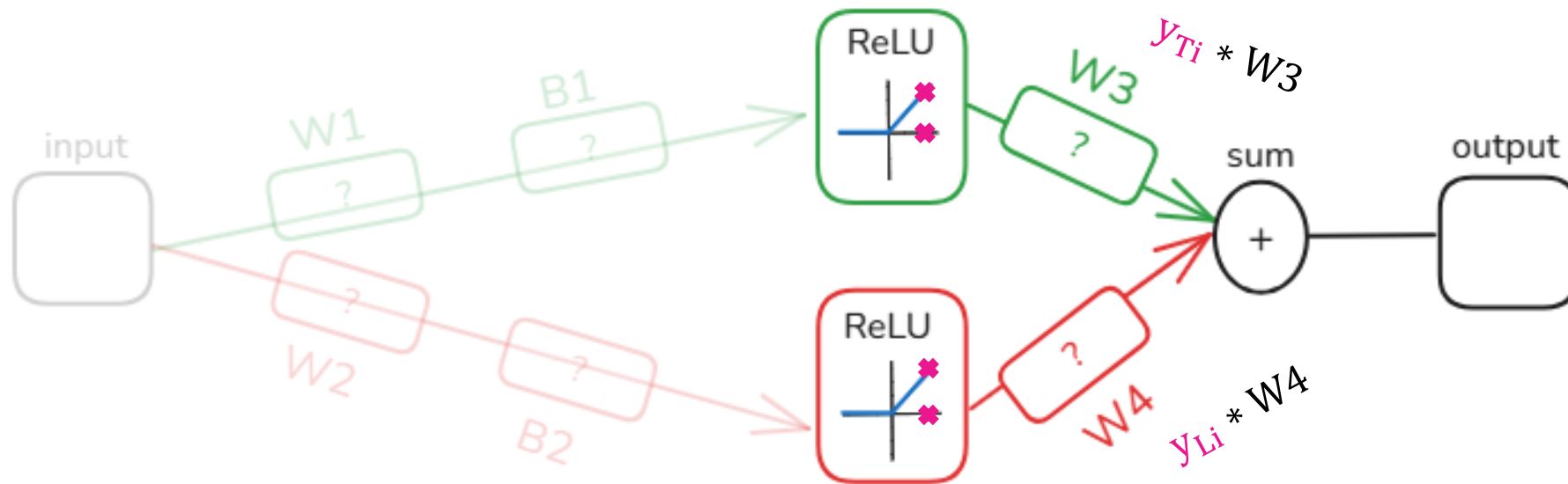
# Optimización múltiples parámetros con backpropagation



# Optimización múltiples parámetros con backpropagation



# Optimización múltiples parámetros con backpropagation



$$\text{Predicted} = (y_{Ti} * W3) + (y_{Li} * W4)$$



# Optimización múltiples parámetros con backpropagation

$$\text{Predicted} = (y_{Ti} * W3) + (y_{Li} * W4)$$

$$\text{SSR} = \sum_{i=0}^n (\text{Observed}_i - \text{Predicted}_i)^2$$



# Optimización múltiples parámetros con backpropagation

$$\frac{dSSR}{dW_4} = \frac{dSSR}{dPredicted} \times \frac{dPredicted}{dW_4}$$



# Optimización múltiples parámetros con backpropagation

$$\frac{dSSR}{dW4} = \frac{dSSR}{dPredicted} \times \frac{dPredicted}{dW4}$$

$$SSR = \sum_{i=0}^n (\text{Observed}_i - \text{Predicted}_i)^2$$

$$\text{Predicted} = (y_{Ti} * W3) + (y_{Li} * W4)$$





# Optimización múltiples parámetros con backpropagation

$$\frac{dSSR}{dW4} = \frac{d \sum_{i=0}^n (\text{Observed}_i - \text{Predicted}_i)^2}{d\text{Predicted}} \times \frac{d[(y_{Ti} * W3) + (y_{Li} * W4)]}{dW4}$$



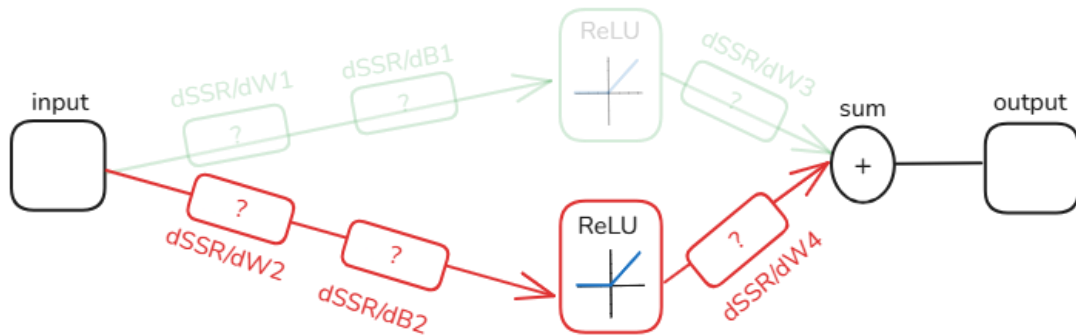
# Optimización múltiples parámetros con backpropagation

$$\frac{dSSR}{dW_4} = \sum_{i=0}^n -2(\text{Observed}_i - \text{Predicted}_i) \times (y_{Li})$$



# Optimización múltiples parámetros con backpropagation

## Lower Branch: derivative expression



$$\frac{dSSR}{dW4} = \frac{dSSR}{dPredicted} \times \frac{dPredicted}{dW4}$$

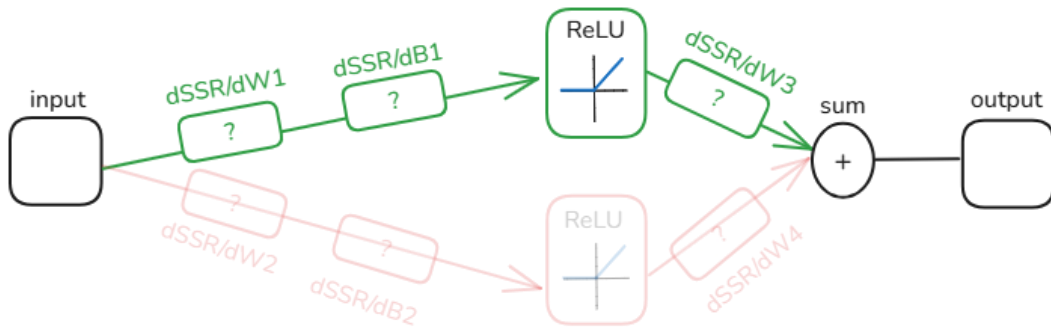
$$\frac{dSSR}{dB2} = \frac{dSSR}{dPredicted} \times \frac{dPredicted}{dy_L} \times \frac{dy_L}{dX_L} \times \frac{dX_L}{dB2}$$

$$\frac{dSSR}{dW2} = \frac{dSSR}{dPredicted} \times \frac{dPredicted}{dy_L} \times \frac{dy_L}{dX_L} \times \frac{dX_L}{dW2}$$



# Optimización múltiples parámetros con backpropagation

**Upper Branch: derivative expression**



$$\frac{dSSR}{dW3} = \frac{dSSR}{dPredicted} \times \frac{dPredicted}{dW3}$$

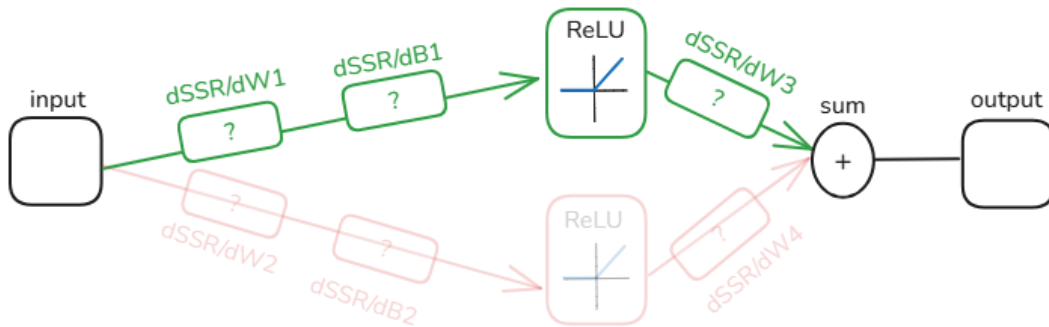
$$\frac{dSSR}{dB1} = \frac{dSSR}{dPredicted} \times \frac{dPredicted}{dy_T} \times \frac{dy_T}{dX_T} \times \frac{dX_T}{dB1}$$

$$\frac{dSSR}{dW1} = \frac{dSSR}{dPredicted} \times \frac{dPredicted}{dy_T} \times \frac{dy_T}{dX_T} \times \frac{dX_T}{dW1}$$



# Optimización múltiples parámetros con backpropagation

**Upper Branch: derivative value**



$$\frac{dSSR}{dW3} = \sum_{i=0}^n -2(\text{Observed}_i - \text{Predicted}_i) \times (y_{Ui})$$

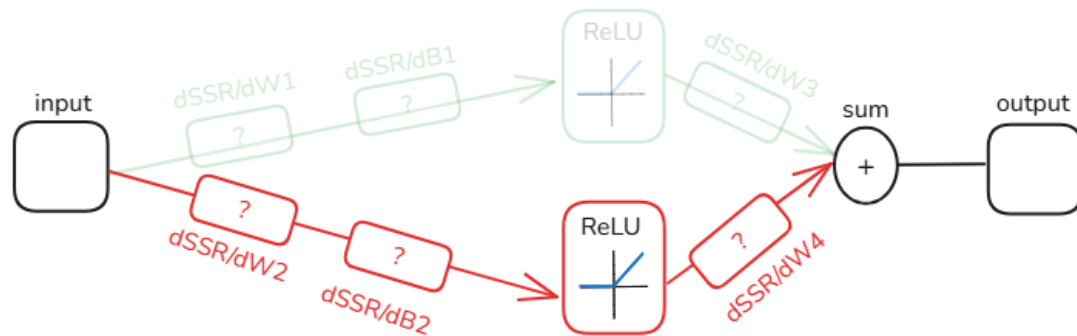
$$\frac{dSSR}{dB1} = \sum_{i=0}^n -2(\text{Observed}_i - \text{Predicted}_i) \times (W3) \times \begin{cases} 0, X_{Ui} < 0 \\ 1, X_{Ui} > 0 \end{cases} \times 1$$

$$\frac{dSSR}{dW1} = \sum_{i=0}^n -2(\text{Observed}_i - \text{Predicted}_i) \times (W3) \times \begin{cases} 0, X_{Ui} < 0 \\ 1, X_{Ui} > 0 \end{cases} \times \text{Dose}_i$$



# Optimización múltiples parámetros con backpropagation

**Lower Branch: derivative value**



$$\frac{dSSR}{dW4} = \sum_{i=0}^n -2(\text{Observed}_i - \text{Predicted}_i) \times (y_{Li})$$

$$\frac{dSSR}{dB2} = \sum_{i=0}^n -2(\text{Observed}_i - \text{Predicted}_i) \times (W4) \times \begin{cases} 0, X_{Li} < 0 \\ 1, X_{Li} > 0 \end{cases} \times 1$$

$$\frac{dSSR}{dW2} = \sum_{i=0}^n -2(\text{Observed}_i - \text{Predicted}_i) \times (W4) \times \begin{cases} 0, X_{Li} < 0 \\ 1, X_{Li} > 0 \end{cases} \times \text{Dose}_i$$



# Optimización múltiples parámetros con backpropagation

¡Práctica!



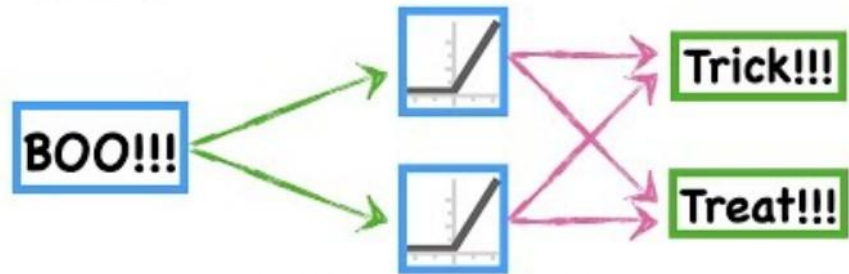
# Q&A



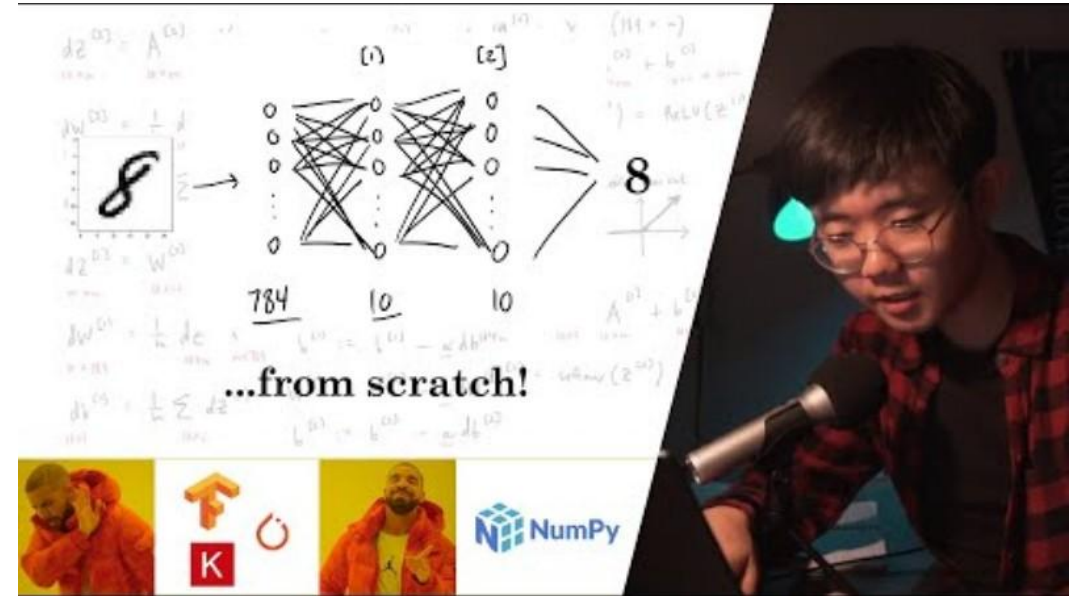


# Recomendaciones

**Happy Halloween!!!**



**(Neural Networks  
are not Scary)**



22 de octubre de 2025

13:30

Aula 4.1.E04, EPS  
Universidad Carlos III



**SERGIO PANIEGO**  
**ML ENGINEER**  
HUGGING FACE

Más información en:  
[www.aiscmadrid.com](http://www.aiscmadrid.com)

AISC Madrid



**ENRIQUE ALCOCER**  
**SOFTWARE DEVELOPER**

**M#SHAKES**

Más información  
[aiscmadrid.com](http://aiscmadrid.com)

**28 de octubre 2025**  
**13:30 – 14:30**  
**Aula 4.1.E04, EPS UC3M**

**@aisc\_madrid**  
**[www.aiscmadrid.com](http://www.aiscmadrid.com)**