Clase 1

x_1	x_2	$y = x_1 \text{ xor } x_2$
0	0	0
0	1	1
1	0	1
1	1	0

$n = 4$ (samples)

$m = 2$ (features)

labels

datos

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

* Modelo: $\hat{y} = x_1 w_1 + x_2 w_2 + 1 b$

parametros a aprender

* Loss function: $J_{\bar{w}} = \frac{1}{4} \sum_{i=1}^{4} (y_i - \hat{y}_i)^2$ MSE

* Optimizador: $\bar{\nabla}_{\bar{w}} (J_{\bar{w}}) = \bar{0}$

$$\bar{w} = [w_1 \quad w_2 \quad b]^T$$

$$\bar{\nabla}(J_{\bar{w}}) = \begin{bmatrix} \partial J / \partial w_1 \\ \partial J / \partial w_2 \\ \partial J / \partial b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

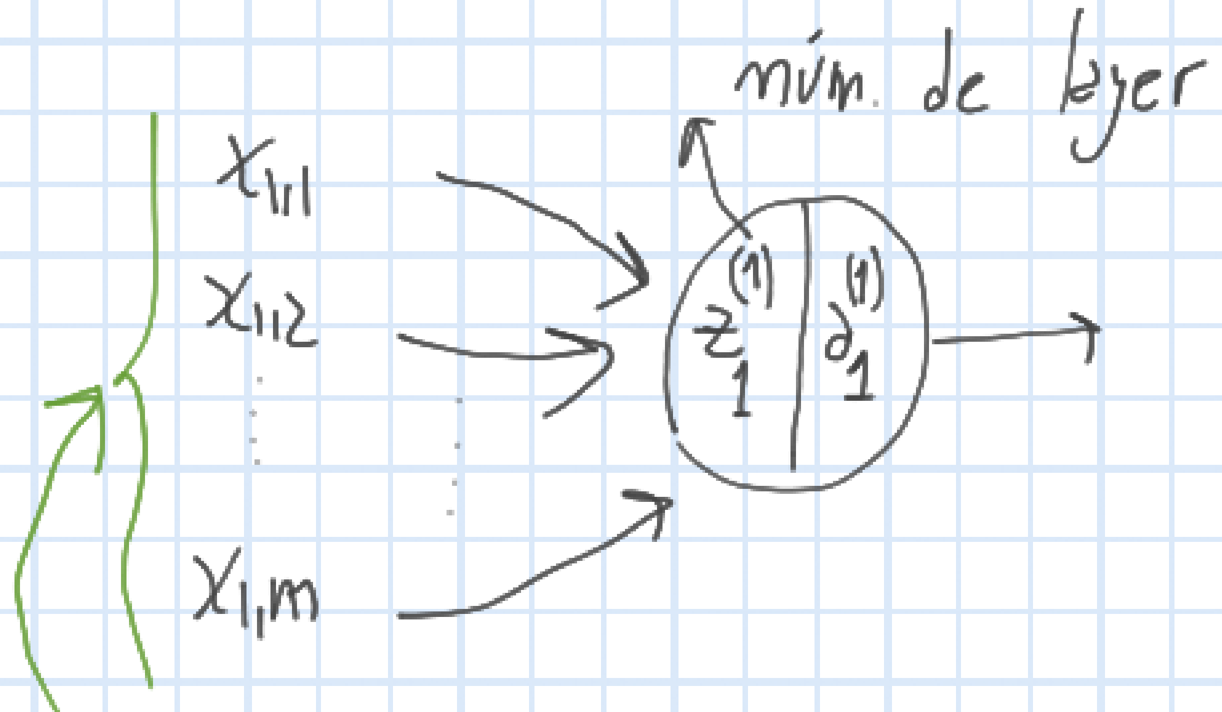
Regresión Lineal

$$\bar{w} = \begin{bmatrix} w_1 \\ w_2 \\ b \end{bmatrix} = (X^T X)^{-1} X^T Y$$

$$\bar{w} = \begin{bmatrix} w_1 \\ w_2 \\ b \end{bmatrix} = [$$

Modelo no lineal

Def de neurona



$$z_1^{(1)} = x_{1,1}\, w_{1,1}^{(1)} + x_{1,2}\, w_{1,2}^{(1)} + \cdots + x_{1,m}\, w_{1,m}^{(1)} + b_1^{(1)} = \bar{W}^{(1)T}\, \bar{x}_1 + b_1^{(1)}$$

$$\underbrace{\bar{w}^T \in \mathbb{R}^{1\times m} \quad \bar{x} \in \mathbb{R}^{m\times 1}}_{\in \mathbb{R}^{1\times 1}}$$

$$a_1^{(1)} = g\left(z_1^{(1)}\right)$$

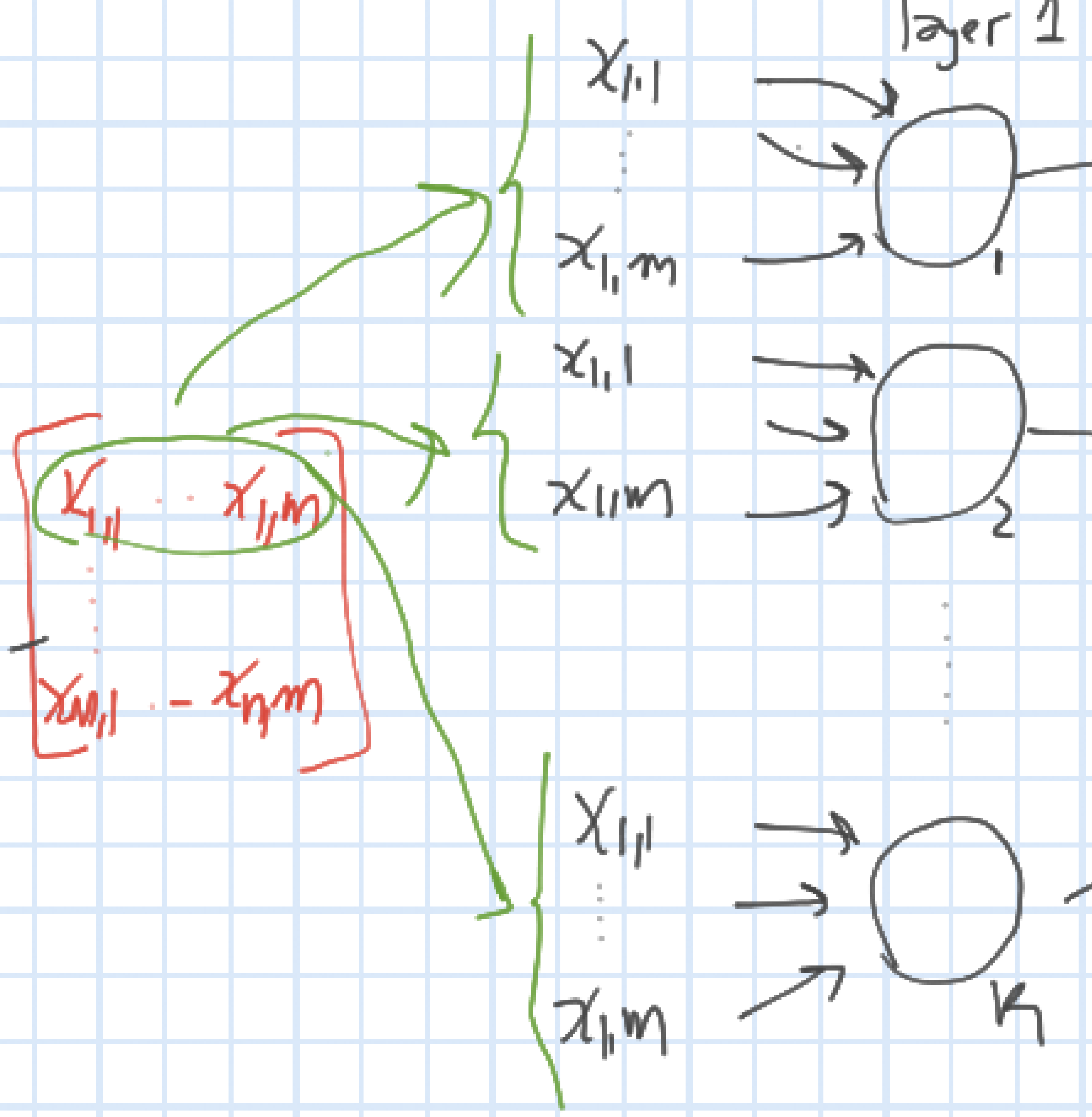
func de activación

$$a_1^{(1)} = \sigma\left(z_1^{(1)}\right) = \frac{1}{1+e^{-z_1^{(1)}}}$$

$$X = \begin{array}{c} m \\ \left[\begin{array}{cccc} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & & & \\ x_{n,1} & \cdots & & x_{n,m} \end{array}\right] \end{array} \Bigg\} n \qquad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$\boxed{\text{Def layers}}$

$$\overbrace{\sigma, \mathrm{Relu}, \tanh}^{\text{func no lineal}}$$



$$a_1^{(1)} = g(z_1^{(1)}) = g\Big(\overbrace{w_{1,1}^{(1)} x_{1,1} + \cdots + w_{1,m}^{(1)} x_{1,m} + b_1^{(1)}}^{\mathbb{R}^{1\times 1}}\Big)$$

$$a_2^{(1)} = g(z_2^{(1)}) = g\left(w_{2,1}^{(1)} x_{1,1} + \cdots + w_{2,m}^{(1)} x_{1,m} + b_2^{(1)}\right)$$

$$\vdots$$

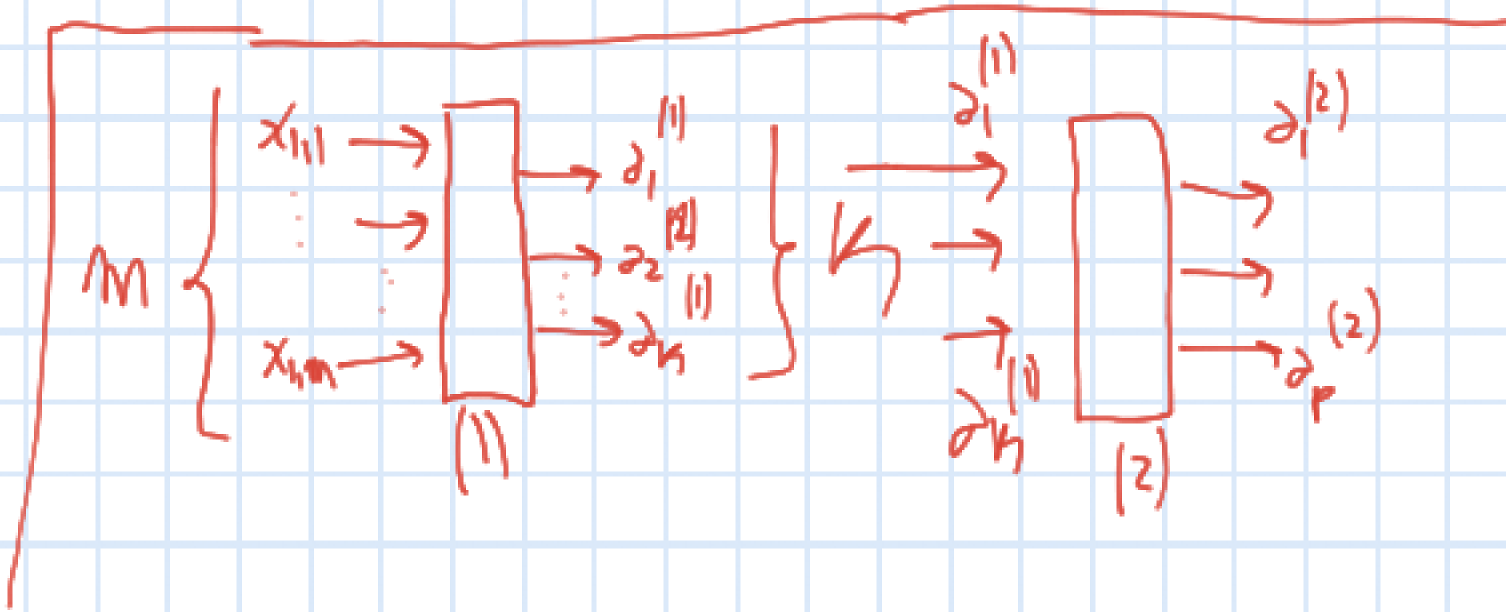
$$a_k^{(1)} = g(z_k^{(1)}) = g\left(w_{k,1}^{(1)} x_{1,1} + \cdots + w_{k,m}^{(1)} x_{1,m} + b_k^{(1)}\right)$$

$$\vec{a}^{(1)} = g^{(1)}\left(W^{(1)} \vec{x}_1 + \vec{b}^{(1)}\right)$$

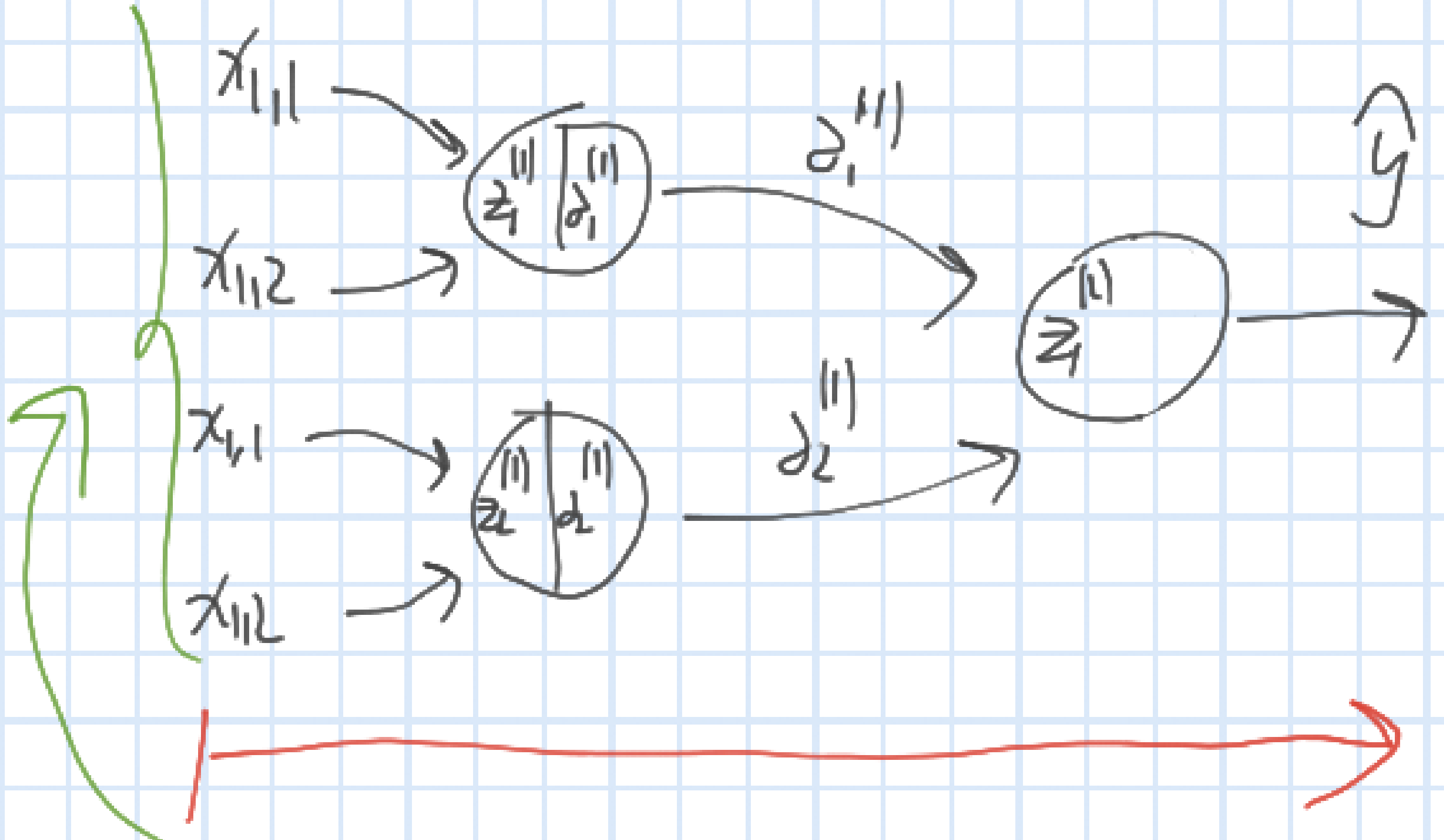
$W^{(1)} \in \mathbb{R}^{k \times m}$

k: dim salida del layer

m: dim entrada del layer



Resolver XOR con modelo no lineal



$$X = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \\ \vdots & \vdots \\ x_{4,1} & x_{4,2} \end{pmatrix}$$

$$Y = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_4 \end{pmatrix}$$

$\boxed{* \text{ Forward}}$

$$z_1^{(1)} = W_{1,1}^{(1)} x_{1,1} + W_{1,2}^{(1)} x_{1,2} + b_1^{(1)}$$

$$a_1^{(1)} = \sigma(z_1^{(1)})$$

$$z_2^{(1)} = W_{2,1}^{(1)} x_{1,1} + W_{2,2}^{(1)} x_{1,2} + b_2^{(1)}$$

$$a_2^{(1)} = \sigma(z_2^{(1)}) = \frac{1}{1+e^{-z_2^{(1)}}}$$

$$z_1^{(2)} = W_{1,1}^{(2)} a_1^{(1)} + W_{1,2}^{(2)} a_2^{(1)} + b_1^{(2)}$$

$$\hat{y}_1 = z_1^{(2)}$$

* Error para cada muestra: $J_1 = \text{error} = (y_i - \hat{y}_i)^2$

¿Cuantos params desconocidos tengo?

9

* Algoritmo de optimización
↳ SGD

n_epochs

mi dataset tiene 4 filas

inicializar params random

for epoch in range(100)
    for sample in range(4)
        (1) Forward
        (2) Backward → $\nabla(J_{w,b})$
        (3) Actualizar los pesos

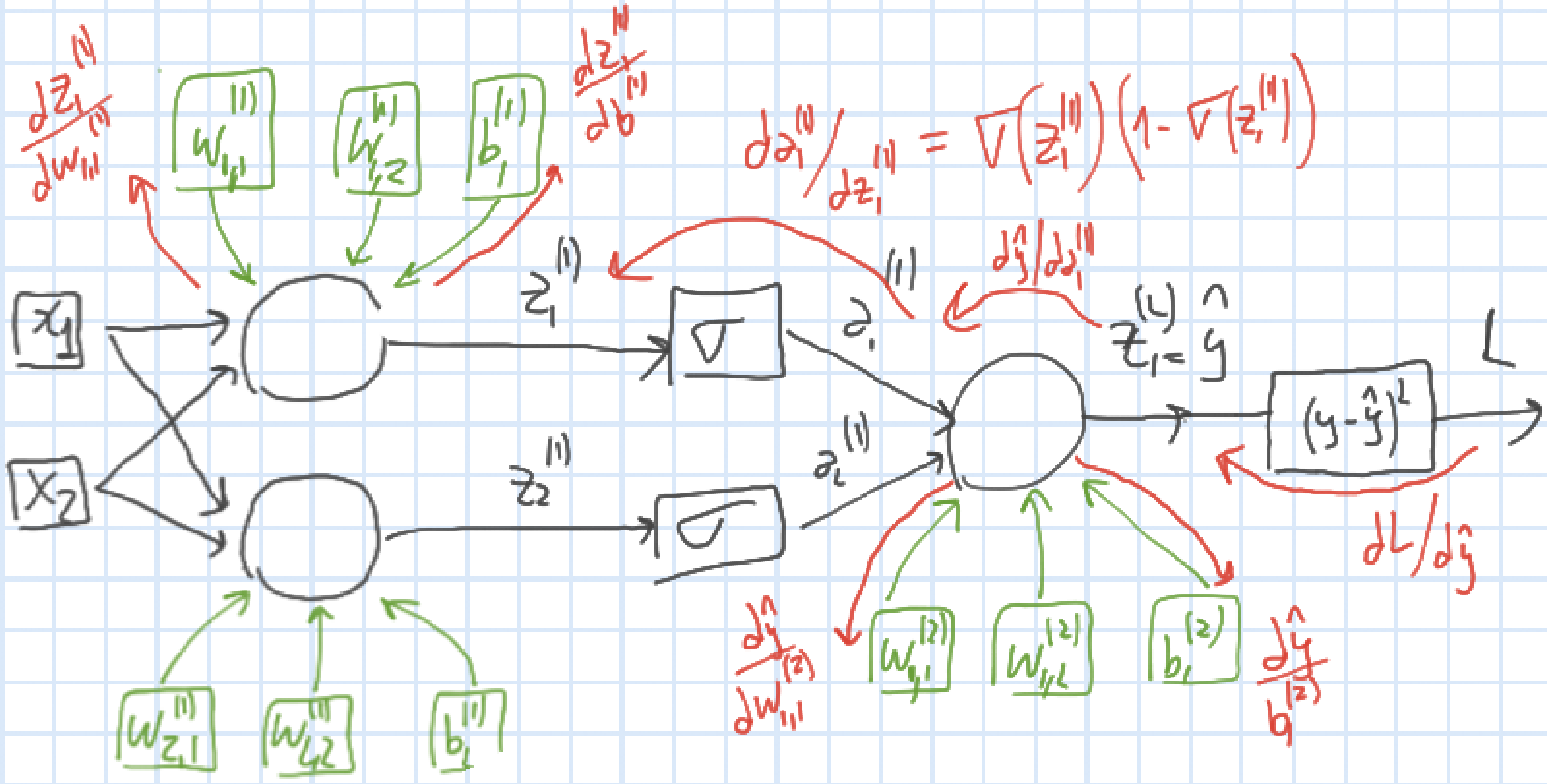
$$w_{1,1}^{[1]} = w_{1,1}^{[1]} - \alpha\, \partial J/\partial w_{1,1}$$
$$w_{1,2}^{[1]} = w_{1,2}^{[1]} - \alpha\, \partial J/\partial w_{1,2}$$
$$b_1^{[1]} = b_1^{[1]} - \alpha\, \partial J/\partial b_1^{[1]}$$

9 actualizaciones

n_epochs = 100
$\alpha \equiv$ learning rate

HP del modelo

Backward



Verde → parámetros del modelo

Rojo → camino del gradiente

* $$\frac{\partial L}{\partial b_1^{(2)}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial b_1^{(2)}}$$
$$= -2(y-\hat{y})\ 1 = -2\ \overbrace{\text{error}}^{(y-\hat{y})}$$

* $$\frac{\partial L}{\partial w_{1,1}^{(2)}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_{1,1}^{(2)}} = -2(y-\hat{y})\ a_1^{(1)}$$

* $$\frac{\partial L}{\partial b_1^{(1)}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_1^{(1)}} \frac{\partial a_1^{(1)}}{\partial z_1^{(1)}} \frac{\partial z_1^{(1)}}{\partial b_1^{(1)}}$$
$$= -2(y-\hat{y})\ w_1^{(2)}\ \sigma(z_1^{(1)})(1-\sigma(z_1^{(1)}))\ 1$$

* Ejercicio

Calcular derivadas restantes e implementar en numpy